
Amazon Elastic Compute Cloud

Linux 인스턴스용 사용 설명서



Amazon Elastic Compute Cloud: Linux 인스턴스용 사용 설명서

Copyright © 2017 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Amazon EC2란 무엇입니까?	1
Amazon EC2의 기능	1
처음으로 Amazon EC2 사용하기	2
관련 서비스	2
Amazon EC2에 액세스	3
Amazon EC2 가격	4
PCI DSS 준수	4
인스턴스 및 AMI	4
인스턴스	5
AMI	6
리전 및 가용 영역	7
리전 및 가용 영역 개념	7
사용 가능한 리전	8
리전 및 Endpoint	9
사용자의 리전 및 가용 영역 확인	9
리소스에 대한 리전 지정	10
가용 영역에서 인스턴스 실행	10
다른 가용 영역으로 인스턴스 마이그레이션	11
루트 디바이스 볼륨	11
루트 디바이스 스토리지 개념	12
루트 디바이스 유형에 따른 AMI 선택	13
인스턴스의 루트 디바이스 유형 확인	14
루트 디바이스 볼륨이 계속 유지되도록 변경	14
설정	16
AWS에 가입	16
IAM 사용자 생성	16
키 페어 생성	18
Virtual Private Cloud(VPC) 생성	19
보안 그룹 생성	19
시작하기	21
개요	21
사전 조건	22
1단계: 인스턴스 시작	22
2단계: 인스턴스에 연결	23
3단계: 인스턴스 정리	24
다음 단계	24
모범 사례	25
자습서	27
자습서: Amazon LinuxLAMP 웹 서버 설치	27
문제 해결	36
관련 주제	37
자습서: WordPress 블로그 호스팅	37
사전 조건	38
WordPress 설치	38
다음 단계	44
도움말! 내 퍼블릭 DNS 이름이 변경되어 블로그를 사용할 수 없습니다.	45
자습서: SSL/TLS를 사용하여 Amazon Linux에서 Apache 웹 서버 구성	46
사전 조건	47
1단계: 서버에서 SSL/TLS 활성화	47
2단계: CA가 서명한 인증서 가져오기	48
3단계: 보안 구성 테스트 및 강화	52
문제 해결	54
자습서: 애플리케이션의 가용성 향상	54
사전 조건	55

애플리케이션 확장 및 로드 밸런싱	55
로드 밸런서 테스트	57
자습서: 인스턴스 원격 관리	57
사용자 계정에 시스템 관리자 액세스 권한 부여	57
SSM 에이전트 설치(Linux)	58
EC2 콘솔을 사용하여 명령 보내기	59
Windows PowerShell용 AWS 도구를 사용하여 명령 보내기	60
AWS CLI를 사용하여 명령 보내기	60
관련 내용	61
Amazon 머신 이미지	62
AMI 사용	62
고유 AMI 생성	62
AMI 구입, 공유 및 판매	63
AMI 등록 해제	63
Amazon Linux	63
AMI 유형	63
시작 권한	64
루트 디바이스 스토리지	64
가상화 유형	66
Linux AMI 찾기	67
Amazon EC2 콘솔을 사용하여 Linux AMI 찾기	68
AWS CLI를 사용하여 AMI 찾기	68
공유 AMI	69
공유 AMI 검색	69
퍼블릭 AMI 설정	71
지정한 AWS 계정과 AMI 공유	72
북마크 사용	73
공유 Linux AMI 지침	74
유료 AMI	78
AMI 판매	78
유료 AMI 찾기	78
유료 AMI 구입	79
인스턴스에 대한 제품 코드 가져오기	80
유료 지원 사용	80
유료 및 지원된 AMI에 대한 청구서	80
AWS Marketplace 구독 관리	80
Amazon EBS 지원 Linux AMI 생성	81
Amazon EBS 지원 AMI 생성 개요	81
인스턴스에서 Linux AMI 생성	82
스냅샷에서 Linux AMI 만들기	83
인스턴스 스토어 기반 Linux AMI 생성	84
인스턴스 스토어 기반 AMI 생성 프로세스 개요	85
사전 조건	85
AMI 도구 설치	86
인스턴스 스토어 지원 인스턴스에서 AMI 생성	112
Amazon EBS 기반 AMI로 변환	121
암호화된 스냅샷이 있는 AMI	124
암호화된 EBS 스냅샷을 포함하는 AMI 시나리오	124
AMI 복사	126
권한	126
리전 간 AMI 복사	127
교차 계정 AMI 복사	128
암호화 및 AMI 복사	128
AMI 복사	129
대기 중인 AMI 복사 작업 종지	130
AMI 등록 취소	130
Amazon EBS 기반 AMI 정리	131

인스턴스 스토어 기반 AMI 정리	131
Amazon Linux	132
Amazon Linux AMI 찾기	132
Amazon Linux 인스턴스 시작 및 연결	132
Amazon Linux AMI 이미지 식별	133
포함된 AWS 명령줄 도구	133
cloud-init	134
리포지토리 구성	136
패키지 추가	136
참조용으로 원본 패키지에 액세스	137
애플리케이션 개발	137
인스턴스 스토어 액세스	137
제품 수명 주기	137
보안 업데이트	138
지원	138
사용자 제공 커널	139
HVM AMI(GRUB)	139
반가상화 AMI(PV-GRUB)	140
인스턴스	145
인스턴스 유형	146
사용 가능한 인스턴스 유형	146
하드웨어 사양	147
가상화 유형	148
네트워킹 및 스토리지 기능	148
인스턴스 제한	149
T2 인스턴스	149
컴퓨팅 최적화 인스턴스	152
메모리 최적화 인스턴스	155
스토리지 최적화 인스턴스	158
액셀러레이티드 컴퓨팅 인스턴스	162
T1 마이크로 인스턴스	166
인스턴스 크기 조정	169
인스턴스 구입 옵션	173
인스턴스 수명 주기 결정	173
예약 인스턴스	174
예약된 인스턴스	200
스팟 인스턴스	203
전용 호스트	246
전용 인스턴스	257
인스턴스 수명 주기	261
인스턴스 시작	261
인스턴스 중지 및 시작(Amazon EBS 기반 인스턴스에만 해당)	262
인스턴스 재부팅	262
인스턴스 만료	262
인스턴스 종료	263
재부팅, 중지 및 종료의 차이	263
시작하기	264
연결	274
중지 및 시작	285
재부팅	288
만료	289
Terminate	291
복구	295
인스턴스 구성	296
일반적인 구성 시나리오	296
소프트웨어 관리	297
사용자 관리	304

프로세서 상태 제어	306
시간 설정	310
호스트 이름 변경	314
동적 DNS 설정	316
시작 시 명령 실행	317
인스턴스 메타데이터 및 사용자 데이터	321
혼합 컴퓨팅 환경에서 EC2 인스턴스 식별	335
Xen 도메인 UUID 검사	335
인스턴스 자격 증명 문서 검사	335
모니터링	336
자동 및 수동 모니터링	337
자동 모니터링 도구	338
수동 모니터링 도구	339
모니터링 모범 사례	339
인스턴스 상태 모니터링	339
인스턴스 상태 확인	340
예약된 이벤트	344
CloudWatch를 사용해 인스턴스 모니터링하기	347
세부 모니터링 활성화	348
얻을 수 있는 측정치 나열	349
지표에 대한 통계 구하기	354
측정치 그래프	359
경보 만들기	359
인스턴스를 중지, 종료, 재부팅 또는 복구하는 경보 만들기	360
CloudWatch 이벤트를 사용한 자동화	367
메모리 및 디스크 메트릭 모니터링	368
지원되는 시스템	368
패키지 내용	368
사전 조건	369
시작하기	370
mon-put-instance-data.pl	371
mon-get-instance-stats.pl	373
콘솔에서 사용자 지정 측정치 보기	375
문제 해결	375
네트워크 및 보안	376
키 페어	377
Amazon EC2를 사용해 키 페어 만들기	378
Amazon EC2로 사용자의 퍼블릭 키 가져오기	378
키 페어에 맞는 퍼블릭 키 검색(Linux)	380
키 페어에 맞는 퍼블릭 키 검색(Windows)	381
키 페어의 지문 확인	381
키 페어 삭제	381
프라이빗 키를 분실했을 때 Linux 인스턴스에 연결하는 방법	382
보안 그룹	385
EC2-Classic의 보안 그룹	386
EC2-VPC의 보안 그룹	386
보안 그룹 규칙	386
기본 보안 그룹	388
사용자 지정 보안 그룹	389
보안 그룹 작업	389
보안 그룹 규칙 참조	393
Controlling Access	398
인스턴스에 대한 네트워크 액세스	399
Amazon EC2 권한 속성	399
IAM 및 Amazon EC2	399
IAM 정책	401
IAM 역할	456

네트워크 액세스	464
Amazon VPC	466
VPC의 장점	466
EC2-Classic와 EC2-VPC의 차이점	467
EC2-Classic과 EC2-VPC 간 리소스 공유 및 액세스	468
VPC에서만 사용할 수 있는 인스턴스 유형	470
Amazon VPC 문서	470
지원되는 플랫폼	471
ClassicLink	472
EC2-Classic에서 VPC로 마이그레이션	481
인스턴스 IP 주소 지정	490
프라이빗 IPv4 주소 및 내부 DNS 호스트 이름	490
퍼블릭 IPv4 주소 및 외부 DNS 호스트 이름	491
탄력적 IP 주소 (IPv4)	492
Amazon DNS 서버	492
IPv6 주소	492
EC2-Classic과 EC2-VPC의 IP 주소 차이점	493
인스턴스에 대한 IP 주소 작업	493
다중 IP 주소	498
탄력적 IP 주소	505
탄력적 IP 주소 기본 사항	506
EC2-Classic 및 EC2-VPC의 탄력적 IP 주소의 차이점	506
탄력적 IP 주소 작업	508
이메일 애플리케이션에 역방향 DNS 사용	512
탄력적 IP 주소 제한	512
네트워크 인터페이스	512
인스턴스 유형별/네트워크 인터페이스당 IP 주소	514
네트워크 인터페이스 시나리오	517
네트워크 인터페이스 구성 모범 사례	517
ec2-net-utils를 사용하여 네트워크 인터페이스 구성	518
네트워크 인터페이스 작업	519
배치 그룹	527
배치 그룹의 제한 사항	528
배치 그룹으로 인스턴스 시작	529
배치 그룹 삭제	530
네트워크 MTU	530
점보 프레임(9001 MTU)	531
경로 MTU 검색	531
두 호스트 간 경로 MTU 확인	531
Amazon EC2 인스턴스에서 MTU 확인 및 설정	532
문제 해결	533
향상된 네트워킹	533
향상된 네트워킹 유형	533
인스턴스에서 향상된 네트워킹 기능 활성화	534
향상된 네트워킹 사용: Intel 82599 VF	534
향상된 네트워킹 사용: ENA	543
ENA 문제 해결	552
스토리지	559
Amazon EBS	560
Amazon EBS의 기능	561
EBS 볼륨	562
EBS 스냅샷	607
EBS 최적화	614
EBS 암호화	617
EBS 성능	621
EBS CloudWatch 이벤트	637
휘발성 스토리지	642

인스턴스 스토어 수명	643
인스턴스 스토리지 볼륨	643
인스턴스 스토어 볼륨 추가	646
SSD 인스턴스 스토어 볼륨	649
인스턴스 스토리지 스왑 볼륨	651
디스크 성능 최적화	654
Amazon EFS	654
사전 조건	655
1단계: EFS 파일 시스템 만들기	655
2단계: 파일 시스템 마운트	655
3단계: 파일 시스템 테스트	656
4단계: 정리	657
Amazon S3	657
Amazon S3 및 Amazon EC2	658
인스턴스 볼륨 제한	659
Linux 볼륨 제한	659
Windows 볼륨 제한	660
대역폭 및 용량 비교	660
디바이스 명명	660
사용 가능한 디바이스 이름	661
디바이스 이름 고려 사항	661
블록 디바이스 매핑	662
블록 디바이스 매핑의 개념	662
AMI 블록 디바이스 매핑	664
인스턴스 블록 디바이스 매핑	666
퍼블릭 데이터 세트 사용	670
퍼블릭 데이터 세트 개념	670
퍼블릭 데이터 세트 찾기	670
스냅샷에서 퍼블릭 데이터 세트 볼륨 생성	671
퍼블릭 데이터 세트 볼륨 연결 및 마운트	672
리소스 및 태그	673
리소스 위치	673
리소스 ID	674
더 긴 ID 작업	675
긴 ID 설정에 대한 액세스 제어	677
리소스 목록화 및 필터링	678
고급 검색	678
콘솔을 이용하여 리소스 목록화	679
콘솔을 이용하여 리소스를 필터링	679
CLI 및 API를 이용하여 목록화 및 필터링	680
리소스에 태그 지정	681
태그 기본 사항	681
리소스에 태그 지정	681
태그 제한	683
리소스에 결제용 태그 지정	683
콘솔을 사용한 태그 작업	684
CLI 또는 API를 사용한 태그 작업	687
서비스 제한	688
현재 제한 조회	688
제한 증가 요청	689
사용 보고서	689
제공되는 보고서	689
사용 보고서에 대한 설정	689
IAM 사용자에게 Amazon EC2 사용 보고서에 대한 액세스 권한 부여	691
인스턴스 사용량	691
EC2 예약 인스턴스 활용	693
문제 해결	698

인스턴스 시작	698
인스턴스 종료 이유 파악	698
인스턴스에 연결	699
인스턴스 연결 중 오류 발생: 연결 시간 초과	699
오류r: 서버에서 사용자 키를 인식하지 못함	701
오류: 호스트 키를 찾을 수 없음. 권한 거부(퍼블릭 키) 또는 인증 실패, 권한 거부	702
오류: 보호되지 않는 프라이빗 키 파일	703
오류: 서버에서 키 거부 또는 지원되는 인증 방법이 없음	704
Safari 브라우저에서 MindTerm 사용 중 오류 발생	704
Mac OS X RDP 클라이언트 사용 중 오류 발생	705
인스턴스를 ping할 수 없음	705
인스턴스 종지	705
인스턴스 종료	706
지연된 인스턴스 종료	706
종료된 인스턴스가 계속 표시됨	706
인스턴스 자동 시작 또는 종료	706
인스턴스 복구 실패	707
상태 확인 실패	707
초기 단계	708
시스템 로그 검색	708
Linux 기반 인스턴스의 시스템 로그 오류 문제 해결	709
메모리 부족: 프로세스 종지	710
ERROR: mmu_update failed(메모리 관리 업데이트 실패)	710
I/O 오류(블록 디바이스 오류)	711
IO ERROR: neither local nor remote disk(분산된 블록 디바이스 손상)	712
request_module: runaway loop modprobe(이전 Linux 버전에서 레거시 커널 modprobe 반복)	713
"FATAL: kernel too old" 및 "fsck: No such file or directory while trying to open /dev"(커널과 AMI 불일치)	713
"FATAL: Could not load /lib/modules" 또는 "BusyBox"(커널 모듈 누락)	714
ERROR Invalid kernel(EC2 커널이 호환되지 않음)	715
request_module: runaway loop modprobe(이전 Linux 버전에서 레거시 커널 modprobe 반복)	716
fsck: No such file or directory while trying to open...(파일 시스템을 찾을 수 없음)	717
파일 시스템 마운트 관련 일반 오류(마운트 실패)	718
VFS: Unable to mount root fs on unknown-block(루트 파일 시스템 불일치)	720
Error: Unable to determine major/minor number of root device...(루트 파일 시스템/디바이스 불일치)	721
XENBUS: Device with no driver...	722
... days without being checked, check forced(파일 시스템 검사 필요)	723
fsck died with exit status...(디바이스 누락)	723
GRUB 프롬프트(grubdom>)	724
Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring(eth0 인터페이스를 가져오는 중: eth0 디바이스의 MAC 주소가 틀려서 무시합니다). (하드 코딩된 MAC 주소)	726
Unable to load SELinux Policy. Machine is in enforcing mode. Halting now(SELinux 정책을 가져올 수 없습니다. 시스템이 강제 실행 모드입니다. 중단됩니다). (잘못된 SELinux 구성)	727
XENBUS: Timeout connecting to devices(Xenbus 시간 초과)	728
인스턴스 용량	728
Error: InsufficientInstanceCapacity	729
Error: InstanceLimitExceeded	729
콘솔 출력 가져오기 및 인스턴스 재부팅	729
인스턴스 재부팅	729
인스턴스 콘솔 출력	729
연결할 수 없는 인스턴스의 스크린샷 캡처	730
호스트 컴퓨터 실패 시 인스턴스 복구	730
인스턴스가 잘못된 볼륨에서 부팅될 경우	731
문서 기록	733
AWS Glossary	748

Amazon EC2란 무엇입니까?

Amazon Elastic Compute Cloud(Amazon EC2)는 Amazon Web Services(AWS) 클라우드에서 확장식 컴퓨팅을 제공합니다. Amazon EC2를 사용하면 하드웨어에 선투자할 필요가 없어 더 빠르게 애플리케이션을 개발하고 배포할 수 있습니다. Amazon EC2를 통해 원하는 만큼 가상 서버를 구축하고 보안 및 네트워크 구성과 스토리지 관리가 가능합니다. Amazon EC2는 요건이나 갑작스러운 인기 증대 등 변동사항에 따라 확장하거나 축소할 수 있어 트래픽 예측 필요성이 줄어듭니다.

클라우드 컴퓨팅에 대한 자세한 내용은 [클라우드 컴퓨팅이란 무엇입니까? 섹션](#)을 참조하십시오.

Amazon EC2의 기능

Amazon EC2는 다음의 기능을 제공합니다.

- 인스턴스: 가상 컴퓨팅 환경
- Amazon 머신 이미지(AMI): 서버에 필요한 운영체제와 여러 소프트웨어들이 적절히 구성된 상태로 제공되는 템플릿으로 인스턴스를 쉽게 만들 수 있습니다.
- 인스턴스 유형: 인스턴스를 위한 CPU, 메모리, 스토리지, 네트워킹 용량의 여러 가지 구성 제공
- 키 쌍을 사용해 인스턴스 로그인 정보 보호(AWS는 공용키를 저장하고 사용자는 개인 키를 안전한 장소에 보관하는 방식)
- 인스턴스 스토어 볼륨: 임시 데이터를 저장하는 스토리지 볼륨으로 인스턴스 종료 시 삭제됨
- Amazon Elastic Block Store(Amazon EBS), 즉 Amazon EBS 볼륨을 사용해 영구 스토리지 볼륨에 데이터 저장
- 인스턴스와 Amazon EBS 볼륨 등의 리소스를 다른 물리적 장소에서 액세스할 수 있는 리전 및 가용 영역
- 보안 그룹을 사용해 인스턴스에 연결할 수 있는 프로토콜, 포트, 소스 IP 범위를 지정하는 방화벽 기능
- 단력적 IP 주소(EIP): 동적 클라우드 컴퓨팅을 위한 고정 IPv4 주소
- 태그: 사용자가 생성하여 Amazon EC2 리소스에 할당할 수 있는 메타데이터
- AWS 클라우드에스는 논리적으로 격리되어 있지만, 원하실때 마다 고객님의 네트워크와 간편히 연결할 수 있는 가상 네트워크, Virtual Private Clouds(VPC)

Amazon EC2에 대한 자세한 내용은 [Amazon EC2제품 페이지](#)를 참조하십시오.

AWS에서의 웹사이트 실행에 대한 자세한 내용은 [웹사이트 및 웹사이트 호스팅](#)을 참조하십시오.

처음으로 Amazon EC2 사용하기

Amazon EC2 사용에 앞서, 먼저 설정이 필요합니다. 일단 설정을 하시면, Amazon EC2 시작 자습서를 따라 하실 수 있습니다. Amazon EC2 기능에 대한 추가 정보가 필요한 경우에는 기술 문서를 참조하십시오.

실행 안내

- [Amazon EC2로 설정 \(p. 16\)](#)
- [Amazon EC2 Linux 인스턴스 시작하기 \(p. 21\)](#)

기본

- [인스턴스 및 AMI \(p. 4\)](#)
- [리전 및 가용 영역 \(p. 7\)](#)
- [인스턴스 유형 \(p. 146\)](#)
- [태그 \(p. 681\)](#)

네트워크 및 보안

- [Amazon EC2 키 페어 \(p. 377\)](#)
- [보안 그룹 \(p. 385\)](#)
- [탄력적 IP 주소 \(p. 505\)](#)
- [Amazon EC2 및 Amazon VPC \(p. 466\)](#)

스토리지

- [Amazon EBS \(p. 560\)](#)
- [인스턴스 스토어 \(p. 642\)](#)

Linux 인스턴스 작업

- [원격 관리\(Run Command\)](#)
- [자습서: Amazon LinuxLAMP 웹 서버 설치 \(p. 27\)](#)
- [자습서: SSL/TLS를 사용하여 Amazon Linux에서 Apache 웹 서버 구성 \(p. 46\)](#)
- [AWS 시작: Linux용 웹 앱 호스팅](#)

AWS 구매에 관련된 질문은 [AWS 영업부에 문의](#)하십시오. Amazon EC2에 관련된 기술적인 문의 사항은 [Amazon EC2 forum](#) 섹션을 참조하십시오.

관련 서비스

Amazon EC2를 사용하여 인스턴스, 볼륨 같은 Amazon EC2 리소스를 직접 구성 할 수 있습니다. 또한 AWS에서 제공되는 다른 서비스를 사용해 Amazon EC2를 구성할 수 있습니다. 자세한 내용은 다음 문서를 참조하십시오.

- [Auto Scaling 사용 설명서](#)

- [AWS CloudFormation 사용 설명서](#)
- [AWS Elastic Beanstalk 개발자 안내서](#)
- [AWS OpsWorks User Guide](#)

Elastic Load Balancing는 애플리케이션의 인바운드 트래픽을 여러 인스턴스로 자동으로 분산해 줍니다. 자세한 내용은 [Elastic Load Balancing 사용 설명서](#) 섹션을 참조하십시오.

Amazon CloudWatch로 인스턴스와 Amazon EBS 볼륨에 관련된 기본 통계 정보를 모니터할 수 있습니다. 자세한 내용은 [Amazon CloudWatch 사용 설명서](#) 섹션을 참조하십시오.

새로운 Amazon EC2 인스턴스 시작 시 Lambda를 활성화하거나 다른 AWS 서비스에서 이벤트 발생 시 SSM Run Command를 호출하는 등의 작업을 자동화하려면 Amazon CloudWatch Events를 사용하십시오. 자세한 내용은 [Amazon CloudWatch Events 사용 설명서](#) 섹션을 참조하십시오.

AWS CloudTrail로 AWS Management Console, 명령줄 도구 및 기타 서비스로부터 고객님 계정의 Amazon EC2 API로 보내진 콜을 모니터할 수 있습니다. 자세한 내용은 [AWS CloudTrail User Guide](#) 섹션을 참조하십시오.

Amazon Relational Database Service(Amazon RDS)로 데이터베이스 인스턴스를 실행하여, AWS에 의해 관리되는, 관계 데이터베이스를 사용하실 수 있습니다. EC2 인스턴스로 데이터베이스를 구축하는 것도 가능하지만, Amazon RDS를 선택하시면, 고객님이 직접 소프트웨어 패치 적용, 백업 및 백업 데이터 저장 등 데이터베이스 관리 작업을 하실 필요가 없습니다. 자세한 내용은 [Amazon Relational Database Service 개발자 안내서](#) 섹션을 참조하십시오.

로컬 환경에서 가상 머신(VM) 이미지를 AWS로 가져와서 사용 가능한 AMI 또는 인스턴스로 변환하려면 VM Import/Export를 사용하십시오. 자세한 내용은 [VM Import/Export 사용 설명서](#)를 참조하십시오.

Amazon EC2에 액세스

Amazon EC2는 웹 기반 사용자 인터페이스인 Amazon EC2 콘솔을 제공합니다. AWS 계정에 가입한 고객은 AWS Management Console에 로그인한 뒤 콘솔 홈페이지에서 EC2를 선택하여 Amazon EC2에 액세스할 수 있습니다.

명령줄 인터페이스를 선호하는 고객의 경우 다음과 같은 옵션이 있습니다.

AWS 명령줄 인터페이스(CLI)

다양한 AWS 제품에서 사용되는 명령어를 제공하며 Windows, Mac, Linux를 지원합니다. 시작하려면 [AWS Command Line Interface 사용 설명서](#) 섹션을 참조하십시오. Amazon EC2 명령어에 대한 자세한 내용은 [EC2\(AWS Command Line Interface Reference\)](#) 섹션을 참조하십시오.

Windows PowerShell용 AWS 도구

PowerShell 환경에서 스크립트 작업을 선호하시는 고객님께서 다양한 AWS 제품을 관리할 수 있도록, 명령줄 도구를 제공합니다. 시작하려면 [Windows PowerShell용 AWS 도구 사용 설명서](#) 섹션을 참조하십시오. Amazon EC2용 cmdlets에 대한 자세한 내용은 [Windows PowerShell용 AWS 도구 Reference](#) 섹션을 참조하십시오.

Amazon EC2는 Query API를 제공합니다. 이 리퀘스트들은, HTTP나 HTTPS의 메시지 교환 방식인 GET이나 POST이며, 미리 정해진 이름인 "Action"을 퀼리 변수로 사용합니다. Amazon EC2에 관련된 API 작업에 대한 자세한 내용은 [작업\(Amazon EC2 API Reference\)](#)을 참조하십시오.

HTTP나 HTTPS 리퀘스트를 직접 보내는 대신, 각 언어가 제공하는 고유의 API를 사용하여 애플리케이션을 빌드하는 것을 선호하는 개발자 고객님을 위해 AWS는, 라이브러리, 샘플 코드, 자습서 및 기타 리소스를 제공합니다. 이 라이브러리는 HTTP/HTTPS 리퀘스트에 암호화된 사인하기, 다시 리퀘스트를 보내기, 오류 응

답 처리하기 등의, 작업을 자동화할 수 있는 기본적인 기능을 제공합니다. 자세한 내용은 [AWS SDK 및 도구](#) 섹션을 참조하십시오.

Amazon EC2 가격

AWS 가입 시 무상으로 Amazon EC2를 시작할 수 있는 [AWS 프리 티어](#)를 제공합니다.

예를 들어, Amazon EC2는 다음의 구입 옵션이 있습니다:

온 디맨드 인스턴스

장기 약정이나 선결제 없이 시간 단위로 사용한 인스턴스를 기준으로 비용을 지불하는 방식입니다.

예약 인스턴스

저가의 요금을 일시불로 선결제하여 1년 또는 3년 계약 기간 동안 인스턴스를 예약하고 해당 인스턴스를 매우 저렴한 요금으로(시간 단위) 사용하는 방식입니다.

스팟 인스턴스

특정 인스턴스 유형을 실행하기 위해 지불할 의사가 있는 시간당 최고 가격을 지정하는 방식입니다. 스팟 가격은 수요와 공급에 따라 변동되지만 고객이 지정한 최고 가격보다 더 많이 지불하는 상황은 발생하지 않습니다. 스팟 가격이 최대 가격을 초과하는 경우, Amazon EC2에서 스팟 인스턴스를 종료합니다.

Amazon EC2에 관련된 전체적인 요금 및 가격 목록은 [Amazon EC2 요금](#)을 참조하십시오.

예시로 주어진 프로비저닝 환경에서의 비용은 [AWS 경제 센터](#)에서 계산할 수 있습니다.

청구 요금은 [AWS 계정 활동 페이지](#)에서 확인할 수 있습니다. 청구서에는 요금 내역을 자세하게 확인할 수 있는 사용 보고서 링크가 포함됩니다. AWS 계정 결제에 대한 자세한 내용은 [AWS 계정 결제](#) 섹션을 참조하십시오.

AWS 결제 및 계정, 이벤트에 관련된 문의 사항은 [AWS 지원 센터에 문의](#)하십시오.

AWS 환경에서의 비용과 보안, 성능 최적화를 돋는 Trusted Advisor의 개요는 [AWS Trusted Advisor](#) 섹션을 참조하십시오.

PCI DSS 준수

Amazon EC2에서는 전자 상거래 웹사이트 운영자 또는 서비스 공급자에 의한 신용 카드 데이터의 처리, 저장 및 전송을 지원하며, Payment Card Industry(PCI) Data Security Standard(DSS) 준수를 검증 받았습니다. AWS PCI 규정 준수 패키지의 사본을 요청하는 방법 등 PCI DSS에 대해 자세히 알아보려면 [PCI DSS 레벨 1](#)을 참조하십시오.

인스턴스 및 AMI

Amazon 머신 이미지(AMI)은 필요한 소프트웨어가 이미 구성되어 있는 템플릿입니다(예: 운영 체제, 애플리케이션 서버, 애플리케이션). AMI에서 인스턴스를 바로 시작하실 수 있는데, 이 인스턴스는 AMI의 사본으로, 클라우드에서 실행되는 가상 서버입니다. 다음 그림과 같이, 한 AMI로 여러 인스턴스를 실행할 수 있습니다.

중지하거나 종료할 때까지 또는 실패하기 전까지 인스턴스는 계속 실행됩니다. 인스턴스가 실패하면 AMI에서 새로 실행할 수 있습니다.

인스턴스

동일한 AMI에서 다른 유형의 인스턴스를 실행할 수 있습니다. 인스턴스 유형에 따라 인스턴스에 사용되는 호스트 컴퓨터의 하드웨어가 결정됩니다. 각 인스턴스 유형은 서로 다른 컴퓨팅 및 메모리 기능을 제공합니다. 인스턴스에서 실행하려는 애플리케이션 또는 소프트웨어에 필요한 메모리 양과 컴퓨팅 파워를 기준으로 인스턴스 유형을 선택하십시오. Amazon EC2 인스턴스 유형별 하드웨어 사양에 대한 자세한 내용은 [Amazon EC2 인스턴스](#)를 참조하십시오.

일단 인스턴스가 시작되면, 인스턴스는 다른 컴퓨터와 다를 것이 없고, 어느 컴퓨터와 동일한 방식으로 다른 시면 됩니다. 인스턴스의 완벽한 통제가 가능하며, 루트 권한이 필요한 명령어는 sudo를 사용해 실행할 수 있습니다.

AWS 계정당 동시 수행할 수 있는 인스턴스 수는 제한됩니다. 해당 제한 및 추가 요청 방법에 대한 자세한 내용은 [Amazon EC2의 실행 인스턴스 한도](#)(종합 FAQ의 Amazon EC2) 섹션을 참조하십시오.

인스턴스 스토리지

인스턴스의 루트 디바이스에는 인스턴스 부팅에 사용되는 이미지가 포함되어 있습니다. 자세한 내용은 [Amazon EC2 루트 디바이스 볼륨 \(p. 11\)](#) 섹션을 참조하십시오.

인스턴스에는 로컬 스토리지 볼륨이 포함될 수 있는데 이것을 인스턴스 스토어 볼륨이라고 하며, 인스턴스 실행 시 블록 디바이스 매핑으로 구성할 수 있습니다. 자세한 내용은 [블록 디바이스 매핑 \(p. 662\)](#) 섹션을 참조하십시오. 고객님의 인스턴스용 볼륨 추가와 매핑이 완료되면, 마운트하여 사용할 수 있습니다. 인스턴스 오류가 발생하거나 중지 혹은 종료된 경우, 해당 볼륨에 저장된 데이터는 손실되기 때문에 이런 볼륨은 임시 데이터 작성에 사용하는 것이 가장 좋습니다. 중요한 데이터는 여러 인스턴스를 연결하는 복제 방법을 사용하여 데이터를 보호하거나 지속적인 보관이 필요한 데이터를 Amazon S3 또는 Amazon EBS 볼륨에 저장하는 방법이 있습니다. 자세한 내용은 [스토리지 \(p. 559\)](#) 섹션을 참조하십시오.

보안 구현 모범 사례

- AWS Identity and Access Management(IAM)을 사용하여 고객님의 인스턴스를 비롯한 AWS 리소스의 액세스를 제어할 수 있습니다. AWS 계정으로 IAM 사용자와 그룹을 생성하고 사용자나 그룹별로 보안 자격 증명을 할당하고 AWS 서비스 및 리소스에 대한 액세스 권한을 부여할 수 있습니다. 자세한 내용은 [Amazon EC2 리소스에 대한 액세스 제어 \(p. 398\)](#)을 참조하십시오.
- 신뢰할 수 있는 호스트나 네트워크만 인스턴스 포트에 액세스할 수 있도록 제한할 수 있습니다. 예를 들어 22번 포트의 유입 트래픽을 제한하면 SSH 액세스 제한이 가능합니다. 자세한 내용은 [Linux 인스턴스에 대한 Amazon EC2 보안 그룹 \(p. 385\)](#) 섹션을 참조하십시오.
- 보안 그룹의 규칙을 정기적으로 검토하고 최소 권한 부여—라는 개념을 항상 적용하고 필요한 경우 필요한 권한만 허가하십시오. 보안 요구 사항이 다른 각 인스턴트를 처리하기 위해 서로 다른 보안 그룹을 생성할 수도 있습니다. 외부 로그인을 허용하는 접속 보안 그룹을 생성하고 여기에 해당되지 않는 나머지 인스턴스는 외부 로그인을 허용하지 않는 그룹으로 할당하는 것도 생각해 볼 수 있습니다.
- AMI 실행 인스턴스는 비밀번호를 사용한 로그인을 비활성화합니다. 비밀번호는 유출이나 해킹이 가능해 보안 위험이 됩니다. 자세한 내용은 [루트 사용자의 암호 방식 원격 로그인 비활성화 \(p. 75\)](#) 섹션을 참조하십시오. 안전한 AMI 공유에 대한 자세한 내용은 [공유 AMI \(p. 69\)](#) 섹션을 참조하십시오.

인스턴스 중지, 시작 및 종료

인스턴스 중지

인스턴스를 중단하면 정상적인 실행종료 과정이 이루어지고 `stopped` 상태가 됩니다. 인스턴스의 모든 Amazon EBS 볼륨이 연결된 상태로 유지되므로 나중에 언제든지 다시 시작할 수 있습니다.

인스턴스가 중지 상태에 있는 동안에는 추가 인스턴스 시간에 대한 요금이 부과되지 않습니다. 인스턴스를 중지에서 실행으로 상태를 전환할 때마다 "인스턴스 시간"으로 비용이 청구되며, 한시간 내에 여러 번 전환된 경우에도 매 경우마다 인스턴스 시간이 청구됩니다. 인스턴스가 중지된 상태에서 인스턴스 유형을 변경하

면, 다음에 인스턴스가 시작된 후 신규 인스턴스 유형에 대한 요금이 부과됩니다. 모든 연결 Amazon EBS 른트 디바이스 사용을 비롯한 인스턴스 사용에 관련된 비용은 일반 Amazon EBS 요금이 적용됩니다.

인스턴스가 중지 상태인 경우 인스턴스에 Amazon EBS 볼륨을 연결하거나 분리할 수 있습니다. 또한 인스턴스로부터 AMI를 만들 수도 있으며, 커널, 램 디스크, 인스턴스 유형을 변경할 수 있습니다.

인스턴스 종료

인스턴스를 종료하면 일반적인 실행종료 과정이 이루어지고 연결된 Amazon EBS 볼륨은 삭제됩니다(단, 해당 볼륨의 `deleteOnTermination` 속성이 `false`로 설정되어 있는 경우는 제외). 인스턴트 자체도 삭제되므로 나중에 다시 시작할 수 없게 됩니다.

인스턴스 종료를 비활성화하면 실수로 인스턴스를 종료하는 일을 방지할 수 있습니다. 이 경우에는 해당 인스턴스에 관련된 `disableApiTermination` 속성을 `true`로 설정했는지 확인하십시오. Linux의 `shutdown -h` 및 Windows의 `shutdown` 같은 인스턴스 실행종료 동작을 제어하려면 `instanceInitiatedShutdownBehavior` 인스턴스 속성을 `stop`이나 `terminate`로 적절히 설정하십시오. 기본 설정은 인스턴스 실행종료 시 Amazon EBS 볼륨을 루트 디바이스로 사용하는 인스턴스는 `stop` 상태, 인스턴스 스토어를 루트 디바이스로 사용하는 인스턴스는 항상 종료 상태로 변경됩니다.

자세한 내용은 [인스턴스 수명 주기 \(p. 261\)](#) 섹션을 참조하십시오.

AMI

Amazon Web Services(AWS)에서는 자주 사용되는 소프트웨어 구성을 포함하는 다양한 Amazon 머신 이미지(AMI)가 공개 게시하고 있습니다. 그 뿐 아니라 AWS 개발자 커뮤니티 회원들이 올린 자체 구성 AMI도 게시되어 있습니다. 또한 얼마든지 사용자 정의된 AMI를 생성할 수 있어서, 고객님께서 필요하신 기능을 모두 갖춘 새로운 인스턴스를 쉽고 빠르게 시작할 수 있습니다. 예를 들어 고객님의 애플리케이션이 웹사이트나 웹 서비스인 경우, 웹 서버와 관련 고정 콘텐츠, 그리고 동적 페이지에 사용할 코드가 포함된 AMI를 정의해 만드실 수 있습니다. 그래서, 이 AMI에서 인스턴스가 시작이 되면, 고객님의 웹 서버가 자동으로 시작되고 애플리케이션은 바로 Request를 처리할 수 있습니다.

모든 인스턴스는 Amazon EBS 기반(AMI의 인스턴스가 실행되는 루트 디바이스가 Amazon EBS 볼륨인 경우) 또는 인스턴스 스토어 기반(AMI의 인스턴스가 실행되는 루트 디바이스가 Amazon S3에 저장된 템플릿에서 생성된 인스턴스 스토어 볼륨인 경우) 중 하나에 해당됩니다.

AMI에 대한 설명을 보시면, 그 인스턴스의 루트디바이스가 `ebs` 인지 `instance store`인지 알 수 있습니다. 각 AMI 유형별로 수행할 수 있는 작업이나 기능이 달라지기 때문에 이 차이점을 아는 것이 중요합니다. 해당 차이점에 대한 자세한 내용은 [루트 디바이스 스토리지 \(p. 64\)](#) 섹션을 참조하십시오.

리전 및 가용 영역

Amazon EC2는 세계 각지의 여러 곳에서 호스팅되고 있습니다. 이 위치들은 리전과 가용 영역으로 구성됩니다. 각 리전은 개별 지리 영역입니다. 각 리전은 가용 영역이라고 알려진 격리된 위치를 여러 개 가지고 있습니다. Amazon EC2는 여러 위치에 인스턴스, 데이터 등 리소스를 배치할 수 있는 기능을 제공합니다. 리소스는 특별하게 이를 지정하지 않을 경우 복제되지 않습니다.

Amazon은 최신 기술을 탑재한 고가용성 데이터 센터를 운영하고 있습니다. 드물기는 하지만 동일한 위치에 있는 인스턴스의 가용성에 영향을 미치는 장애가 발생할 수도 있습니다. 그런 장애의 영향을 받는 단일한 위치에서 모든 인스턴스를 호스팅하는 경우에는 모든 인스턴스가 사용이 불가능해질 수 있습니다.

목차

- [리전 및 가용 영역 개념 \(p. 7\)](#)
- [사용 가능한 리전 \(p. 8\)](#)
- [리전 및 Endpoint \(p. 9\)](#)
- [사용자의 리전 및 가용 영역 확인 \(p. 9\)](#)
- [리소스에 대한 리전 지정 \(p. 10\)](#)
- [가용 영역에서 인스턴스 실행 \(p. 10\)](#)
- [다른 가용 영역으로 인스턴스 마이그레이션 \(p. 11\)](#)

리전 및 가용 영역 개념

각 리전은 완전히 독립적입니다. 각 가용 영역은 기본적으로 서로 격리되어 있지만, 한 리전의 가용 영역들은 낮은 수준의 지연 시간을 가진 링크를 통해 연결되어 있습니다. 다음 다이어그램은 리전 및 가용 영역 간 관계를 설명합니다.

Amazon EC2 리소스는 글로벌 수준에서 리전 또는 가용 영역에 둑여 있는 상태로 존재합니다. 자세한 내용은 [리소스 위치 \(p. 673\)](#) 섹션을 참조하십시오.

리전

각 Amazon EC2 리전은 다른 Amazon EC2 리전에서 완전히 격리되도록 설계되었습니다. 이를 통해 가장 강력한 내결함성 및 안정성을 달성할 수 있습니다.

리소스를 볼 때 해당 리전에 배치된 리소스만 표시됩니다. 이는 리전이 서로 격리되어 있기 때문이며, AWS는 각 리전 간에 자동으로 리소스를 복제하지 않습니다.

인스턴스를 시작할 때 동일한 리전에 위치하고 있는 AMI를 선택해야 합니다. AMI가 다른 리전에 있는 경우는 해당 AMI를 사용할 리전으로 복사할 수 있습니다. 자세한 내용은 [AMI 복사 \(p. 126\)](#) 섹션을 참조하십시오.

리전 간 통신은 모두 퍼블릭 인터넷을 통해 수행됩니다. 따라서 데이터 보호를 위해 적절한 암호화 방법을 사용해야 합니다. 단, 리전 간 데이터 전송은 비용이 청구됩니다. 자세한 내용은 [Amazon EC2 요금 - 데이터 전송](#)을 참조하십시오.

가용 영역

인스턴스를 실행할 때 사용자가 직접 가용 영역을 선택하거나 AWS가 사용자를 위해 가용 영역을 선택하도록 할 수 있습니다. 복수의 가용 영역에 걸쳐 인스턴스를 배포했을 때 하나의 인스턴스에 장애가 발생한 경우를 대비하여, 다른 가용 영역의 인스턴스가 장애가 발생한 인스턴스 관련 요청을 처리할 수 있도록 애플리케이션을 설계할 수 있습니다.

또한 탄력적 IP 주소를 사용하여 한 가용 영역에서 인스턴스의 장애가 발생한 경우 다른 가용 영역의 인스턴스로 주소를 신속하게 매핑함으로써 인스턴스의 장애를 마스킹할 수 있습니다. 자세한 내용은 [탄력적 IP 주소 \(p. 505\)](#) 섹션을 참조하십시오.

가용 영역은 리전 코드와 식별 문자의 조합으로 표시됩니다. 예: us-east-1a. 리전의 가용 영역에 걸쳐 리소스가 배포될 수 있도록 AWS는 각 계정에 대한 식별자에 가용 영역을 독립적으로 매핑하고 있습니다. 예를 들어 한 계정의 us-east-1a 가용 영역은 다른 계정에 대한 us-east-1a 가용 영역과 동일한 위치에 존재하지 않을 수 있습니다. 계정 간에 걸쳐 가용 영역을 조정할 수 있는 방법은 없습니다.

가용 영역이 시간에 따라 커지면서 가용 영역을 확장할 수 있는 AWS의 역량 부족으로 인해 가용 영역이 제한을 받을 수 있습니다. 이런 문제가 생긴 경우 AWS는 제한을 받는 가용 영역에서 인스턴스를 실행하지 못하도록 합니다(해당 가용 영역에서 이미 인스턴스를 보유하고 있는 경우는 제외). 또 최종적으로는 신규 고객에 대해서는 가용 영역의 목록에서 제한을 받는 가용 영역을 제거하게 될 수도 있습니다. 따라서 어떤 리전에 대해 한 계정에서 사용 가능한 가용 영역의 수는 다른 계정과 다를 수 있습니다.

계정에서 사용 가능한 가용 영역을 표시할 수 있습니다. 자세한 내용은 [사용자의 리전 및 가용 영역 확인 \(p. 9\)](#) 섹션을 참조하십시오.

사용 가능한 리전

계정을 통해 자신이 사용할 수 있는 리전을 결정합니다. 예:

- 하나의 AWS 계정은 복수의 리전을 제공하므로 사용자는 자신의 요구 사항에 맞는 위치에서 Amazon EC2 인스턴스를 시작할 수 있습니다. 예를 들어 유럽의 고객들과 좀더 가까운 곳에 위치하거나 또는 법적 요구 사항을 준수하기 위해 유럽에 소재한 위치에서 인스턴스를 실행할 필요가 있을 수 있습니다.
- AWS GovCloud (US) 계정은 오직 AWS GovCloud (US) 리전에 대한 액세스 권한을 제공합니다. 자세한 내용은 [AWS GovCloud \(US\) Region](#) 섹션을 참조하십시오.
- Amazon AWS (중국) 계정은 오직 중국(베이징) 리전에 대한 액세스 권한을 제공합니다.

다음 표에는 AWS 계정이 제공하는 리전이 나열되어 있습니다. AWS GovCloud (US) 또는 중국(베이징)과 같은 추가 리전은 AWS 계정으로부터 설명 또는 액세스할 수 없습니다.

코드	이름
us-east-1	미국 동부(버지니아 북부)
us-east-2	미국 동부(오하이오)
us-west-1	미국 서부(캘리포니아 북부 지역)
us-west-2	미국 서부(오레곤)
ca-central-1	캐나다(중부)
eu-west-1	EU(아일랜드)
eu-central-1	EU(프랑크푸르트)
eu-west-2	EU(런던)
ap-northeast-1	아시아 태평양(도쿄)
ap-northeast-2	아시아 태평양(서울)
ap-southeast-1	아시아 태평양(싱가포르)
ap-southeast-2	아시아 태평양(시드니)
ap-south-1	아시아 태평양(뭄바이)

코드	이름
sa-east-1	남아메리카(상파울루)

자세한 내용은 [AWS 글로벌 인프라](#)를 참조하십시오.

리전당 가용 영역의 수 및 매핑은 AWS 계정마다 다를 수 있습니다 계정에서 사용 가능한 가용 영역의 목록을 확인하려면 Amazon EC2 콘솔 또는 명령줄 인터페이스를 사용할 수 있습니다. 자세한 내용은 [사용자의 리전 및 가용 영역 확인 \(p. 9\)](#) 섹션을 참조하십시오.

리전 및 Endpoint

명령줄 인터페이스 또는 API 작업을 사용해서 인스턴스로 작업할 경우, 그에 대한 리전 엔드포인트를 지정해야 합니다. Amazon EC2 리전 및 엔드포인트에 대한 자세한 내용은 Amazon Web Services 일반 참조의 [Regions and Endpoints](#)를 참조하십시오.

AWS GovCloud (US)의 엔드포인트 및 프로토콜에 대한 자세한 내용은 AWS GovCloud (US) User Guide의 [AWS GovCloud \(US\) Endpoints](#)를 참조하십시오.

사용자의 리전 및 가용 영역 확인

Amazon EC2 콘솔 또는 명령줄 인터페이스를 사용해 계정에서 어떤 리전과 가용 영역을 사용할 수 있는지 확인할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

콘솔을 사용하여 리전 및 가용 영역을 확인하는 방법

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 모음에서 리전 선택기(region selector)의 옵션을 조회합니다.
3. 대시보드의 [Service Health], [Availability Zone Status] 아래에서 가용 영역을 봅니다.

명령줄을 사용해서 리전 및 가용 영역을 확인하는 방법

1. [AWS CLI] 다음과 같이 `describe-regions` 명령을 사용하여 계정의 리전을 설명합니다.

```
aws ec2 describe-regions
```

2. [AWS CLI] 다음과 같이 `describe-availability-zones` 명령을 사용하여 지정된 리전 내의 가용 영역을 설명합니다.

```
aws ec2 describe-availability-zones --region region-name
```

3. [Windows PowerShell용 AWS 도구] 다음과 같이 `Get-EC2Region` 명령을 사용하여 계정의 리전을 설명합니다.

```
Get-EC2Region
```

4. [Windows PowerShell용 AWS 도구] 다음과 같이 `Get-EC2AvailabilityZone` 명령을 사용하여 지정된 리전 내의 가용 영역을 설명합니다.

```
Get-EC2AvailabilityZone -Region region-name
```

리소스에 대한 리전 지정

Amazon EC2 리소스를 생성할 때마다 리소스에 대한 리전을 지정할 수 있습니다. AWS Management Console 또는 명령줄을 사용해서 리소스에 대한 리전을 지정할 수 있습니다.

Note

일부 AWS 리소스는 모든 리전과 가용 영역에서 사용 가능하지 않을 수 있습니다. 특정 가용 영역에서 인스턴스를 시작하기 전에 원하는 리전 또는 가용 영역에 필요한 리소스를 생성하도록 하십시오.

콘솔을 사용해서 리소스에 대한 리전을 지정하는 방법

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 모음의 리전 선택기를 사용합니다.

명령줄을 사용해서 기본 리전을 지정하는 방법

원하는 리전 엔드포인트로 환경 변수의 값을 설정할 수 있습니다(예: `https://ec2.us-west-1.amazonaws.com`).

- `AWS_DEFAULT_REGION` (AWS CLI)
- `Set-AWSDefaultRegion` (Windows PowerShell용 AWS 도구)

다른 방법으로는 `--region`(AWS CLI) 또는 `-Region`(Windows PowerShell용 AWS 도구) 명령줄 옵션을 개별 명령에 포함해 사용하는 것이 있습니다. 예, `--region us-west-1`.

Amazon EC2에 대한 자세한 내용은 [Amazon Elastic Compute Cloud 엔드포인트](#) 섹션을 참조하십시오.

가용 영역에서 인스턴스 실행

인스턴스를 실행할 때, 특정 고객들과 가까운 곳에 인스턴스를 위치시키거나 법률 또는 기타 요구사항을 준수할 수 있도록 적절한 리전을 선택합니다. 각각의 개별적인 가용 영역에서 인스턴스를 시작함으로써 단일 위치에서 장애가 발생할 경우 애플리케이션을 보호할 수 있습니다.

인스턴스를 실행할 때, 사용할 리전의 가용 영역을 따라 지정할 수 있습니다. 가용 영역을 지정하지 않는 경우 AWS가 사용자를 위해 가용 영역을 선택해줍니다. 초기 인스턴스를 실행할 때는 기본 가용 영역을 그대로 사용하는 것이 좋습니다. 이를 통해 시스템 상태 및 가용 용량에 따라 사용자에게 가장 알맞은 가용 영역을 AWS가 선택할 수 있기 때문입니다. 추가적으로 인스턴스를 실행하는 경우, 추가 할 새 인스턴스를 실행 중인 인스턴스와 가까운 곳에 이를 위치시키거나 실행 중인 인스턴스와 이를 분리시키는 경우에만 가용 영역을 지정합니다.

콘솔을 사용하여 인스턴스에 대한 가용 영역을 지정하는 방법

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 대시보드에서 Launch Instance를 선택합니다.
3. 마법사의 지침대로 진행합니다. [Configure Instance Details](인스턴스 정보 구성) 페이지에서 다음을 수행합니다.
 - [EC2-Classic] 목록에서 가용 영역 옵션 중 하나를 선택하거나 [No Preference]를 선택해서 AWS가 사용자를 위한 최상의 가용 영역을 선택하도록 합니다.
 - [EC2-VPC] 목록에서 서브넷 옵션 중 하나를 선택하거나 [No preference (default subnet in any Availability Zone)]를 선택해서 AWS가 사용자를 위한 최상의 가용 영역을 선택하도록 합니다.

AWS CLI를 사용하여 인스턴스에 대한 가용 영역을 지정하는 방법

이 경우 [run-instances](#) 명령을 다음 옵션 중 하나를 포함해 사용할 수 있습니다.

- [EC2-Classic] --placement
- [EC2-VPC] --subnet-id

Windows PowerShell용 AWS 도구를 사용하여 인스턴스에 대한 가용 영역을 지정하는 방법

이 경우 [New-EC2Instance](#) 명령을 다음 옵션 중 하나를 포함해 사용할 수 있습니다.

- [EC2-Classic] -AvailabilityZone
- [EC2-VPC] -SubnetId

다른 가용 영역으로 인스턴스 마이그레이션

필요하다면 한 가용 영역에서 다른 가용 영역으로 인스턴스를 마이그레이션할 수 있습니다. 예를 들어, 인스턴스의 인스턴스 유형을 변경하려 할 때 AWS가 새 인스턴스 유형의 인스턴스를 현재 가용 영역에서 실행할 수 없는 경우, 해당 인스턴스를 AWS가 원하는 새 인스턴스 유형의 인스턴스를 실행할 수 있는 가용 영역은 마이그레이션할 수 있습니다.

마이그레이션 과정은 원래 인스턴스에서 AMI를 생성, 새 가용 영역에서 인스턴스를 실행, 새 인스턴스의 구성 업데이트 등을 수반합니다. 이 과정은 구체적으로 다음과 같습니다.

다른 가용 영역으로의 인스턴스 마이그레이션 방법

1. 인스턴스에서 AMI를 만듭니다. 이 과정은 인스턴스에 대한 운영 체제 및 루트 디바이스 볼륨의 유형에 따라 달라집니다. 자세한 내용은 사용자의 운영 체제 및 루트 디바이스 볼륨에 대한 문서를 참조하십시오.
 - Amazon EBS 지원 Linux AMI 생성 (p. 81)
 - [인스턴스 스토어 기반 Linux AMI 생성](#) (p. 84)
 - Amazon EBS 기반 Windows AMI 생성
 - [인스턴스 스토어 지원 Windows AMI 생성](#)
2. [EC2-VPC] 인스턴스의 프라이빗 IPv4 주소를 보존해야 할 경우, 현재 가용 영역에서 서브넷을 삭제한 후, 새 가용 영역에 기존 서브넷과 동일한 IPv4 주소 범위를 가지는 서브넷을 생성해야 합니다. 서브넷을 삭제하기 전에는 서브넷의 모든 인스턴스를 종료해야 합니다. 따라서 현재 서브넷의 모든 인스턴스를 새 서브넷으로 이동하려면 서브넷의 모든 인스턴스에서 AMI를 생성해야 합니다.
3. 방금 전 생성한 AMI에서 인스턴스를 실행하고 새 가용 영역 또는 서브넷을 지정합니다. 원래 인스턴스와 동일한 인스턴스 유형을 사용하거나 새로운 인스턴스 유형을 선택할 수 있습니다. 자세한 내용은 [가용 영역에서 인스턴스 실행](#) (p. 10) 섹션을 참조하십시오.
4. 원래 인스턴스가 연결된 탄력적 IP 주소를 가지고 있는 경우 이를 새 인스턴스와 연결합니다. 자세한 내용은 [탄력적 IP 주소의 연결 해제 후 다른 인스턴스와 재연결](#) (p. 509) 섹션을 참조하십시오.
5. 원래 인스턴스가 예약 인스턴스인 경우, 예약에 대한 가용 영역을 변경합니다. 인스턴스 유형도 변경한 경우는 예약에 대한 인스턴스 유형도 변경할 수 있습니다. 자세한 내용은 [변경 요청 제출](#) (p. 195) 섹션을 참조하십시오.
6. (선택 사항) 원래 인스턴스를 종료합니다. 자세한 내용은 [인스턴스 종료](#) (p. 292) 섹션을 참조하십시오.

Amazon EC2 루트 디바이스 볼륨

사용자가 인스턴스를 시작할 때 루트 디바이스 볼륨에는 인스턴스를 부팅하는 데 사용된 이미지가 들어 있습니다. Amazon EC2가 출시되었던 시점에서는 Amazon EC2 인스턴스 스토어가 모든 AMI를 지원했으므로

AMI에서 시작한 인스턴스의 루트 디바이스는 Amazon S3에 저장된 템플릿으로부터 생성된 인스턴스 스토어 볼륨이었습니다. Amazon EBS가 출시된 후에는 Amazon EBS의 지원을 받는 AMI가 도입되었습니다. 따라서 AMI에서 시작한 인스턴스의 루트 디바이스는 Amazon EBS 스냅샷으로부터 생성된 Amazon EBS 볼륨입니다.

사용자는 Amazon EC2 인스턴스 스토어가 지원하는 AMI와 Amazon EBS에서 지원하는 AMI 중에서 선택할 수 있습니다. 시작 속도가 더 빠르고 영구 스토리지를 사용하는 Amazon EBS 지원 AMI를 사용하는 것이 좋습니다.

루트 볼륨에 대해 Amazon EC2에서 사용하는 디바이스 이름에 대한 자세한 내용은 [Linux 인스턴스의 디바이스 명명 \(p. 660\)](#) 섹션을 참조하십시오.

함목

- [루트 디바이스 스토리지 개념 \(p. 12\)](#)
- [루트 디바이스 유형에 따른 AMI 선택 \(p. 13\)](#)
- [인스턴스의 루트 디바이스 유형 확인 \(p. 14\)](#)
- [루트 디바이스 볼륨이 계속 유지되도록 변경 \(p. 14\)](#)

루트 디바이스 스토리지 개념

인스턴스 스토어 지원 AMI 또는 Amazon EBS 지원하는 AMI에서 인스턴스를 시작할 수 있습니다. AMI 설명에는 AMI의 유형이 포함되며, 설명 중간에 루트 디바이스가 `ebs`(Amazon EBS 지원) 또는 `instance store`(인스턴스 스토어 지원)로 언급됩니다. 각 AMI 유형별로 수행할 수 있는 작업이나 기능이 달라지기 때문에 이 차이점을 아는 것이 중요합니다. 해당 차이점에 대한 자세한 내용은 [루트 디바이스 스토리지 \(p. 64\)](#) 섹션을 참조하십시오.

인스턴스 스토어 지원 인스턴스

인스턴스 스토어를 루트 디바이스로 사용하는 인스턴스는 하나 이상의 인스턴스 스토어 볼륨을 자동으로 사용할 수 있으며, 이러한 볼륨 중 하나가 루트 디바이스 볼륨 역할을 합니다. 인스턴스가 시작되면 인스턴스를 부팅하는 데 사용된 이미지가 루트 볼륨으로 복사됩니다. 인스턴스 유형에 따라 다른 인스턴스 스토어 볼륨을 사용할 수도 있습니다.

인스턴스 스토어 볼륨의 모든 데이터는 인스턴스가 실행되는 동안 유지되지만, 인스턴스가 종료되거나(인스턴스 스토어 지원 인스턴스는 [Stop] 작업을 지원하지 않음) 장애가 발생하면(예: 기본 드라이브에 문제가 있는 경우) 데이터가 삭제됩니다.

인스턴스 스토어가 지원하는 인스턴스는 종료되거나 장애가 발생할 경우 복원이 불가능합니다. Amazon EC2 인스턴스 스토어가 지원하는 인스턴스를 사용하려는 경우 여러 가용 영역의 인스턴스 스토어로 데이터를 분산하는 것이 좋습니다. 또한 인스턴스 스토어 볼륨의 중요한 데이터를 정기적으로 영구 스토리지로 백업해야 합니다.

자세한 내용은 [Amazon EC2 인스턴스 스토어 \(p. 642\)](#) 섹션을 참조하십시오.

Amazon EBS 지원 인스턴스

Amazon EBS를 루트 디바이스로 사용하는 인스턴스에는 자동으로 Amazon EBS 볼륨이 연결됩니다. Amazon EBS 지원 인스턴스를 시작하면 사용하는 AMI가 참조하는 각 Amazon EBS 스냅샷에 대한 Amazon EBS 볼륨이 생성됩니다. 인스턴스 유형에 따라 다른 Amazon EBS 볼륨이나 인스턴스 스토어 볼륨을 사용할 수도 있습니다.

Amazon EBS 지원 인스턴스는 종지한 후 다시 시작해도 연결된 볼륨에 저장된 데이터에 아무런 영향이 없습니다. Amazon EBS 지원 인스턴스가 종지 상태일 때 다양한 인스턴스 및 볼륨 관련 작업을 수행할 수 있습니다.

다. 예를 들어 인스턴스의 속성을 수정하거나, 인스턴스의 크기를 변경하거나, 사용하는 커널을 업데이트하거나, 디버깅 등의 목적으로 루트 볼륨을 실행 중인 다른 인스턴스에 연결할 수 있습니다.

Amazon EBS 지원 인스턴스에서 장애가 발생할 경우 다음 방법 중 하나로 세션을 복원할 수 있습니다.

- 중지 후 다시 시작합니다(먼저 이 방법 시도).
- 모든 관련 볼륨의 스냅샷을 자동으로 생성하고 새 AMI를 생성합니다. 자세한 내용은 [Amazon EBS 지원 Linux AMI 생성 \(p. 81\)](#) 섹션을 참조하십시오.
- 다음 단계에 따라 볼륨에 새 인스턴스를 연결합니다.
 1. 루트 볼륨의 스냅샷을 생성합니다.
 2. 스냅샷을 사용하여 새 AMI를 등록합니다.
 3. 새 AMI에서 새 인스턴스를 시작합니다.
 4. 나머지 Amazon EBS 볼륨을 이전 인스턴스에서 분리합니다.
 5. Amazon EBS 볼륨을 새 인스턴스에 다시 연결합니다.

자세한 내용은 [Amazon EBS 볼륨 \(p. 562\)](#) 섹션을 참조하십시오.

루트 디바이스 유형에 따른 AMI 선택

인스턴스를 시작할 때 지정하는 AMI가 인스턴스의 루트 디바이스 볼륨 유형을 결정합니다.

콘솔을 사용하여 Amazon EBS 지원 AMI를 선택하려면 다음을 수행합니다.

1. Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [AMIs]를 선택합니다.
3. 필터 목록에서 [Public images] 등의 이미지 유형을 선택합니다. 검색 창에서 [Platform]을 선택하고 [Amazon Linux] 등의 운영 체제를 선택한 후 [Root Device Type]을 선택하고 [EBS images]를 선택합니다.
4. (선택 사항) 결정에 도움이 되는 추가 정보를 얻으려면 [Show/Hide Columns] 아이콘을 선택하고 표시할 열을 업데이트한 후 [Close]를 선택합니다.
5. AMI를 선택하고 AMI ID를 기록해둡니다.

콘솔을 사용하여 인스턴스 스토어 지원 AMI를 선택하려면 다음을 수행합니다.

1. Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [AMIs]를 선택합니다.
3. 필터 목록에서 [Public images] 등의 이미지 유형을 선택합니다. 검색 창에서 [Platform]을 선택하고 [Amazon Linux] 등의 운영 체제를 선택한 후 [Root Device Type]을 선택하고 [Instance store]를 선택합니다.
4. (선택 사항) 결정에 도움이 되는 추가 정보를 얻으려면 [Show/Hide Columns] 아이콘을 선택하고 표시할 열을 업데이트한 후 [Close]를 선택합니다.
5. AMI를 선택하고 AMI ID를 기록해둡니다.

명령줄을 사용하여 AMI의 루트 디바이스 볼륨 유형을 확인하려면 다음을 수행합니다.

다음 명령 중 하나를 사용할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- `describe-images` (AWS CLI)
- `Get-EC2Image`(Windows PowerShell용 AWS 도구)

인스턴스의 루트 디바이스 유형 확인

콘솔을 사용하여 인스턴스의 루트 디바이스 유형을 확인하려면 다음을 수행합니다.

1. Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Instances]를 선택하고 인스턴스를 선택합니다.
3. 아래를 참고하여 [Description] 탭의 [Root device type] 값을 확인합니다.
 - 값이 ebs이면 Amazon EBS 지원 인스턴스입니다.
 - 값이 instance store이면 인스턴스 스토어 지원 인스턴스입니다.

명령줄을 사용해 인스턴스의 루트 디바이스 유형을 확인하는 방법

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- `describe-instances` (AWS CLI)
- `Get-EC2Instance` (Windows PowerShell용 AWS 도구)

루트 디바이스 볼륨이 계속 유지되도록 변경

기본적으로 Amazon EBS에서 지원하는 AMI의 루트 디바이스 볼륨은 인스턴스 종료 시 삭제됩니다. 기본 동작을 변경하려면 블록 디바이스 매핑을 사용하여 `DeleteOnTermination` 속성을 `false`로 설정합니다.

콘솔을 사용하여 루트 볼륨이 계속 유지되도록 변경

콘솔을 사용하여 인스턴스 시작 시 `DeleteOnTermination` 속성을 변경할 수 있습니다. 실행 중인 인스턴스의 속성을 변경하려면 명령줄을 사용해야 합니다.

콘솔을 사용해 인스턴스의 루트 디바이스 볼륨 유지를 설정하는 방법(인스턴스 시작 시)

1. Amazon EC2 콘솔을 엽니다.
2. Amazon EC2 콘솔 대시보드에서 [Launch Instance]를 선택합니다.
3. [Choose an Amazon Machine Image (AMI)] 페이지에서 사용할 AMI를 선택하고 [Select]를 선택합니다.
4. 마법사 안내에 따라 [Choose an Instance Type] 및 [Configure Instance Details] 설정을 완료합니다.
5. [Add Storage] 페이지에서 루트 볼륨에 대한 [Delete On Termination]의 선택을 해제합니다.
6. 나머지 마법사 페이지를 완료한 다음 [Launch]를 선택합니다.

인스턴스의 세부 정보 창에서 루트 디바이스 볼륨의 세부 정보를 조회하여 설정을 확인할 수 있습니다. [Block devices] 옆의 루트 디바이스 볼륨 항목을 선택합니다. [Delete on termination]의 기본 설정은 `True`입니다. 기본 동작을 변경하면 [Delete on termination]이 `False`가 됩니다.

AWS CLI를 사용하여 인스턴스의 루트 볼륨이 계속 유지되도록 변경

AWS CLI를 사용하여 인스턴스 시작 시 또는 인스턴스 실행 중에 `DeleteOnTermination` 속성을 변경할 수 있습니다.

Example 시작 시

루트 볼륨을 보존하려면 `run-instances` 명령을 사용하여 `DeleteOnTermination` 속성을 `false`로 설정하는 블록 디바이스 매핑을 포함시킵니다.

```
aws ec2 run-instances --block-device-mappings file://mapping.json other parameters...
```

mapping.json에서 다음을 지정합니다.

```
[  
  {  
    "DeviceName": "/dev/sda1",  
    "Ebs": {  
      "DeleteOnTermination": false  
    }  
  }  
]
```

아래와 같이 [describe-instances](#) 명령을 사용하고 명령 출력에서 디바이스의 BlockDeviceMappings 항목을 찾아보면 DeleteOnTermination이 false인 것을 확인할 수 있습니다.

```
...  
  "BlockDeviceMappings": [  
    {  
      "DeviceName": "/dev/sda1",  
      "Ebs": {  
        "Status": "attached",  
        "DeleteOnTermination": false,  
        "VolumeId": "vol-1234567890abcdef0",  
        "AttachTime": "2013-07-19T02:42:39.000Z"  
      }  
    }  
  ...
```

Example 인스턴스 실행 중

루트 볼륨을 보존하려면 [modify-instance-attribute](#) 명령을 사용하여 DeleteOnTermination 속성을 false로 설정하는 블록 디바이스 매핑을 포함시킵니다.

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --block-device-mappings  
file://mapping.json
```

mapping.json에서 다음을 지정합니다.

```
[  
  {  
    "DeviceName": "/dev/sda1",  
    "Ebs": {  
      "DeleteOnTermination": false  
    }  
  }  
]
```

Amazon EC2로 설정

Amazon Web Services(AWS)에 이미 가입한 경우 Amazon EC2를 즉시 사용할 수 있습니다. Amazon EC2 콘솔을 열고 [Launch Instance]를 클릭한 후 시작 마법사의 단계에 따라 첫 번째 인스턴스를 시작합니다.

AWS에 아직 가입하지 않았거나 첫 번째 인스턴스를 시작하는 데 도움이 필요한 경우 다음 작업을 완료하여 Amazon EC2 사용을 준비하십시오.

1. AWS에 가입 (p. 16)
2. IAM 사용자 생성 (p. 16)
3. 키 페어 생성 (p. 18)
4. Virtual Private Cloud(VPC) 생성 (p. 19)
5. 보안 그룹 생성 (p. 19)

AWS에 가입

Amazon Web Services(AWS)에 가입하면 Amazon EC2를 포함해 AWS의 모든 서비스에 AWS 계정이 자동으로 등록됩니다. 사용한 서비스에 대해서만 청구됩니다.

Amazon EC2에서는 사용한 만큼만 지불하면 됩니다. AWS를 처음 사용하는 고객인 경우 Amazon EC2를 무료로 시작할 수 있습니다. 자세한 내용은 [AWS 프리 티어](#) 섹션을 참조하십시오.

이미 AWS 계정이 있다면 다음 작업으로 건너뛰십시오. AWS 계정이 없는 경우에는 아래 단계를 수행하여 계정을 만드십시오.

AWS 계정을 만들려면 다음을 수행합니다.

1. <https://aws.amazon.com/>을 열고 [Create an AWS Account]를 선택합니다.
2. 온라인 지시 사항을 따릅니다.

등록 절차 중 전화를 받고 전화 키패드를 사용하여 PIN을 입력하는 과정이 있습니다.

다음 작업에 필요하므로 AWS 계정 번호를 기록합니다.

IAM 사용자 생성

Amazon EC2 등의 AWS 서비스에 액세스하려면 자격 증명을 제공해야 합니다. 리소스에 대한 액세스 권한이 있는지 여부를 파악해야 하기 때문입니다. 콘솔은 암호를 요구합니다. AWS 계정에 대한 액세스 키를 생

성하면 명령줄 인터페이스 또는 API에 액세스할 수 있습니다. 그러나 AWS 계정에 자격 증명을 사용하여 AWS에 액세스하지 말고, AWS Identity and Access Management(IAM)를 사용하는 것이 좋습니다. IAM 사용자를 생성하여 관리자 권한과 함께 IAM 그룹에 추가하거나, 이 사용자에게 관리자 권한을 부여하십시오. 그러면 IAM 사용자의 특정 URL이나 자격 증명을 사용하여 AWS에 액세스할 수 있습니다.

AWS에 가입했지만 IAM 사용자를 생성하지 않았다면 IAM 콘솔에서 생성할 수 있습니다. 콘솔을 사용하는 데 익숙하지 않은 경우 [Working with the AWS Management Console](#)의 개요를 참조하십시오.

IAM 사용자를 직접 생성하여 Administrators 그룹에 추가하려면

1. <https://console.aws.amazon.com/iam/>에서 IAM 콘솔에 로그인합니다.
2. 탐색 창에서 Users와 Add user를 차례대로 선택합니다.
3. User name으로 Administrator와 같은 사용자 이름을 입력합니다. 이름은 문자, 숫자, 그리고 다음과 같은 기호로 구성될 수 있습니다. 더하기(+), 등호(=), 쉼표(), 마침표(.), 앤(@), 밑줄(_), 하이픈(-). 이름은 대소문자를 구분하지 않으며 최대 64자입니다.
4. AWS Management Console access 옆의 확인란을 선택하고 Custom password를 선택한 다음 텍스트 상자에 새 사용자의 암호를 입력합니다. 선택적으로 Require password reset을 선택하여 다음에 사용자가 로그인할 때 의무적으로 새 암호를 선택하도록 설정할 수 있습니다.
5. Next: Permissions를 선택합니다.
6. Set permissions for user 페이지에서 Add user to group을 선택합니다.
7. Create group을 선택합니다.
8. Create group 대화 상자에 새 그룹의 이름을 입력합니다. 이름은 문자, 숫자, 그리고 다음과 같은 기호로 구성될 수 있습니다. 더하기(+), 등호(=), 쉼표(), 마침표(.), 앤(@), 밑줄(_), 하이픈(-). 이름은 대소문자를 구분하지 않으며 최대 128자입니다.
9. Filter로 Job function을 선택합니다.
10. 정책 목록에서 AdministratorAccess 옆의 확인란을 선택합니다. 그런 다음 Create group을 선택합니다.
11. 그룹 목록으로 돌아가 새로 만든 그룹 옆의 확인란을 선택합니다. 목록에서 그룹을 확인하기 위해 필요한 경우 Refresh를 선택합니다.
12. Next: Review를 선택하여 새 사용자에 추가될 그룹 멤버십의 목록을 확인합니다. 계속 진행할 준비가 되었으면 Create user를 선택합니다.

이제 동일한 절차에 따라 그룹이나 사용자를 추가 생성하여 AWS 계정 리소스에 액세스할 수 있는 권한을 사용자에게 부여할 수 있게 되었습니다. 특정 AWS 리소스에 대한 사용자의 권한을 제한하는 정책을 사용하는 방법을 배우려면 [액세스 관리 및 AWS 리소스를 관리하기 위한 정책의 예](#) 단원으로 이동하십시오.

이 새로운 IAM 사용자로 로그인하려면 먼저 AWS 콘솔에서 로그아웃한 후 다음 URL을 사용합니다. 여기에서 your_aws_account_id는 하이픈을 제외한 AWS 계정 번호를 나타냅니다. 예를 들어, AWS 계정 번호가 1234-5678-9012이면 계정 ID는 123456789012입니다.

`https://your_aws_account_id.signin.aws.amazon.com/console/`

방금 생성한 IAM 사용자 이름(이메일 주소가 아님)과 암호를 입력합니다. 로그인하면 탐색 모음에 "your_user_name @ your_aws_account_id"가 표시됩니다.

로그인 페이지의 URL에 AWS 계정 ID가 포함되지 않게 하려면 계정 별칭을 생성합니다. IAM 콘솔의 탐색 창에서 [Dashboard]를 클릭합니다. 대시보드에서 [Customize]를 클릭하고 회사 이름 등의 별칭을 입력합니다. 계정 별칭 생성 후 로그인할 때는 다음 URL을 사용합니다.

`https://your_account_alias.signin.aws.amazon.com/console/`

본인 계정의 IAM 사용자 로그인 링크를 확인하려면 IAM 콘솔을 열고 대시보드에서 [IAM users sign-in link] 아래를 확인합니다.

IAM에 대한 자세한 내용은 [IAM 및 Amazon EC2 \(p. 399\)](#) 섹션을 참조하십시오.

키 페어 생성

AWS에서는 퍼블릭 키 암호화를 사용하여 인스턴스에 대한 로그인 정보를 보호합니다. Linux 인스턴스에는 암호가 없으므로 인스턴스에 안전하게 로그인하기 위해 키 페어를 사용합니다. 인스턴스를 시작할 때 키 페어의 이름을 지정한 다음 프라이빗 키를 제공하여 SSH를 사용하여 로그인할 때

키 페어를 아직 생성하지 않은 경우 Amazon EC2 콘솔을 사용하여 생성할 수 있습니다. 여러 리전에서 인스턴스를 시작하려면 각 리전에서 키 페어를 생성해야 합니다. 리전에 대한 자세한 내용은 [리전 및 가용 영역 \(p. 7\)](#) 섹션을 참조하십시오.

키 페어를 생성하려면

1. 이전 섹션에서 생성한 URL을 사용하여 AWS에 로그인합니다.
2. AWS 대시보드에서 [EC2]를 선택하여 Amazon EC2 콘솔을 엽니다.
3. 탐색 모음에서 키 페어를 만들 리전을 선택합니다. 현재 위치와 관계없이 사용자가 고를 수 있는 리전을 임의로 선택합니다. 그러나 키 페어는 리전에 고유합니다. 예를 들어, 미국 서부(오레곤) 지역에서 인스턴스를 시작하려면 미국 서부(오레곤) 지역에서 인스턴스에 대한 키 페어를 생성해야 합니다.
4. 탐색 창의 [NETWORK & SECURITY]에서 [Key Pairs]를 클릭합니다.

Tip

탐색 창은 콘솔의 왼쪽에 있습니다. 창이 보이지 않는 경우 창이 최소화되었을 수 있으니 화살표를 클릭해 확대하십시오. [Key Pairs] 링크가 보이려면 아래로 스크롤해야 할 수 있습니다.

5. Create Key Pair를 클릭합니다.
6. [Create Key Pair] 대화 상자의 [Key pair name] 필드에 새 키 페어의 이름을 입력하고 [Create]를 클릭합니다. 기억하기 쉬운 이름을 선택합니다(예: IAM 사용자 이름, -key-pair 및 리전 이름의 조합). 예를 들어, me-key-pair-uswest2로 지정할 수 있습니다.
7. 브라우저에서 프라이빗 키 파일이 자동으로 다운로드됩니다. 기본 파일 이름은 키 페어의 이름으로 지정된 이름이며, 파일 이름 확장명은 .pem입니다. 안전한 장소에 프라이빗 키 파일을 저장합니다.

Important

이때가 사용자가 프라이빗 키 파일을 저장할 수 있는 유일한 기회입니다. 사용자는 인스턴스를 시작할 때 키 페어의 이름을 입력하고, 인스턴스에 연결할 때마다 해당하는 프라이빗 키를 입력해야 합니다.

8. Mac 또는 Linux 컴퓨터에서 SSH 클라이언트를 사용하여 Linux 인스턴스에 연결하려면 사용자만 프라이빗 키 파일을 읽을 수 있도록 다음 명령으로 해당 권한을 설정합니다.

```
$ chmod 400 your_user_name-key-pair-region_name.pem
```

자세한 내용은 [Amazon EC2 키 페어 \(p. 377\)](#) 섹션을 참조하십시오.

키 페어를 사용하여 인스턴스에 연결하려면

Mac 또는 Linux를 실행 중인 컴퓨터에서 Linux 인스턴스에 연결하려면 -i 옵션과 프라이빗 키 경로를 사용하여 SSH 클라이언트에 .pem 파일을 지정합니다. Windows를 실행 중인 컴퓨터에서 Linux 인스턴스에 연결하려면 MindTerm 또는 PuTTY를 사용합니다. PuTTY를 사용하려면 먼저 설치하고 다음 절차에 따라 .pem 파일을 .ppk 파일로 변환해야 합니다.

(선택 사항) PuTTY를 사용하여 Windows에서 Linux 인스턴스에 연결하려면

1. <http://www.chiark.greenend.org.uk/~sgtatham/putty/>에서 PuTTY를 다운로드하여 설치합니다. 전체 제품군을 설치해야 합니다.

2. PuTTYgen을 시작합니다. 예를 들어, [시작] 메뉴에서 [모든 프로그램] > [PuTTY] > [PuTTYgen]을 클릭합니다.
3. [Type of key to generate]에서 [SSH-2 RSA]를 선택합니다.
4. [Load]를 클릭합니다. 기본적으로 PuTTYgen에는 확장명이 .ppk인 파일만 표시됩니다. .pem 파일을 찾으려면 모든 유형의 파일을 표시하는 옵션을 선택합니다.
5. 이전 절차에서 생성한 키 파일을 선택하고 [Open]을 클릭합니다. [OK]를 클릭하여 확인 대화상자를 닫습니다.
6. [Save private key]를 클릭합니다. PuTTYgen에서 암호 없이 키 저장에 대한 경고가 표시됩니다. Yes를 클릭합니다.
7. 키 페어에 사용한 키와 동일한 이름을 지정합니다. PuTTY에서 자동으로 .ppk 파일 확장명을 추가합니다.

Virtual Private Cloud(VPC) 생성

Amazon VPC에서는 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다. 기본 VPC가 있는 경우 이 섹션을 건너뛰고 [보안 그룹 생성 \(p. 19\)](#) 작업으로 이동합니다. 기본 VPC가 있는지 여부를 확인하려면 [Amazon EC2 콘솔에서 지원되는 플랫폼 \(p. 471\)](#) 섹션을 참조하십시오. 그렇지 않으면 아래 단계에 따라 계정에서 기본이 아닌 VPC를 생성할 수 있습니다.

Important

계정에서 특정 리전의 EC2-Classic을 지원하는 경우 해당 리전에 기본 VPC가 없는 것입니다. T2 인스턴스는 VPC로 시작해야 합니다.

기본 VPC가 아닌 VPC를 생성하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 모음에서 VPC를 생성할 리전을 선택합니다. VPC는 리전에 고유하므로 키 페어를 생성한 리전과 동일한 리전을 선택해야 합니다.
3. VPC 대시보드에서 Start VPC Wizard를 클릭합니다.
4. [Step 1: Select a VPC Configuration] 페이지에서 [VPC with a Single Public Subnet]이 선택되어 있는지 확인하고 [Select]를 클릭합니다.
5. [Step 2: VPC with a Single Public Subnet] 페이지의 [VPC name] 필드에 VPC의 이름을 입력합니다. 다른 기본 구성 설정은 그대로 두고 [Create VPC]를 클릭합니다. 확인 페이지에서 [OK]를 클릭합니다.

Amazon VPC에 대한 자세한 내용은 [What is Amazon VPC?](#)(출처: Amazon VPC 사용 설명서) 섹션을 참조하십시오.

보안 그룹 생성

보안 그룹은 연결된 인스턴스에 대한 방화벽 역할을 하여 인스턴스 수준에서 인바운드 트래픽과 아웃바운드 트래픽을 모두 제어합니다. SSH를 사용하여 IP 주소에서 인스턴스에 연결할 수 있게 하는 규칙을 보안 그룹에 추가해야 합니다. 어디서나 인바운드 및 아웃바운드 HTTP/HTTPS 액세스를 허용하는 규칙을 추가할 수도 있습니다.

여러 리전에서 인스턴스를 시작하려면 각 리전에서 보안 그룹을 생성해야 합니다. 리전에 대한 자세한 내용은 [리전 및 가용 영역 \(p. 7\)](#)을 참조하십시오.

사전 조건

로컬 컴퓨터의 퍼블릭 IPv4 주소가 필요합니다. Amazon EC2 콘솔의 보안 그룹 편집기는 퍼블릭 IPv4 주소를 자동으로 검색할 수 있습니다. 또는 인터넷 브라우저에서 "내 IP 주소"와 같은 검색 구문을 사용하거나 <http://checkip.amazonaws.com/> 서비스를 사용할 수도 있습니다. 고정 IP 주소가 없는 방화벽 뒤나 ISP(인터넷 서비스 공급자)를 통해 연결되어 있는 경우 클라이언트 컴퓨터가 사용하는 IP 주소의 범위를 찾아야 합니다.

최소 권한으로 보안 그룹을 생성하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.

Tip

또는 Amazon VPC 콘솔을 사용하여 보안 그룹을 생성할 수 있습니다. 그러나 이 절차의 지침은 Amazon VPC 콘솔과 일치하지 않습니다. 따라서 이전 섹션에서 Amazon VPC 콘솔로 전환한 경우 Amazon EC2 콘솔로 다시 전환하고 이 지침을 사용하거나, Amazon VPC 시작 안내서의 [Set Up a Security Group for Your VPC](#)에 설명된 지침을 사용하십시오.

2. 탐색 모음에서 보안 그룹을 생성할 리전을 선택합니다. 보안 그룹은 리전에 고유하므로 키 페어를 생성한 리전과 동일한 리전을 선택해야 합니다.
3. 탐색 창에서 Security Groups를 클릭합니다.
4. Create Security Group을 클릭합니다.
5. 새 보안 그룹의 이름과 설명을 입력합니다. 기억하기 쉬운 이름을 선택합니다(예: IAM 사용자 이름, _SG_ 및 리전 이름의 조합). 예를 들어, me_SG_uswest2로 지정할 수 있습니다.
6. [VPC] 목록에서 VPC를 선택합니다. 기본 VPC가 있는 경우 별표(*)가 표시되어 있습니다.

Note

계정에서 EC2-Classic을 지원하는 경우 이전 작업에서 생성한 VPC를 선택합니다.

7. [Inbound] 탭에서 다음 규칙을 생성한 다음(각 새 규칙에 대해 [Add Rule] 클릭) [Create]를 클릭합니다.
 - [Type] 목록에서 [HTTP]를 선택하고 [Source]가 [Anywhere](0.0.0.0/0)로 설정되어 있는지 확인합니다.
 - [Type] 목록에서 [HTTPS]를 선택하고 [Source]가 [Anywhere](0.0.0.0/0)로 설정되어 있는지 확인합니다.
 - [Type] 목록에서 [SSH]를 선택합니다. 필드를 로컬 컴퓨터의 퍼블릭 IPv4 주소로 자동으로 채우려면 [Source] 상자에서 [My IP]를 선택하면 됩니다. 또는 [Custom]을 선택하고 컴퓨터 또는 네트워크의 퍼블릭 IPv4 주소를 CIDR 표기법으로 지정해도 됩니다. 개별 IP 주소를 CIDR 표기법으로 지정하려면 라우팅 접미사 /32를 추가합니다(예: 203.0.113.25/32). 회사에서 주소를 범위로 할당하는 경우 전체 범위(예: 203.0.113.0/24)를 지정합니다.

Warning

보안상 테스트를 위해 짧은 시간 동안만 허용하는 경우를 제외하고 모든 IPv4 주소(0.0.0.0/0)에서의 인스턴스에 대한 SSH 액세스를 허용하지 않는 것이 좋습니다.

자세한 내용은 [Linux 인스턴스에 대한 Amazon EC2 보안 그룹 \(p. 385\)](#) 섹션을 참조하십시오.

Amazon EC2 Linux 인스턴스 시작하기

Linux 인스턴스를 시작, 연결 및 사용하여 Amazon Elastic Compute Cloud(Amazon EC2)를 시작합니다. 인스턴스는 AWS 클라우드의 가상 서버입니다. Amazon EC2를 사용하여 인스턴스에서 실행되는 운영 체제와 애플리케이션을 설정하고 구성할 수 있습니다.

AWS 가입 시 무상으로 Amazon EC2를 시작할 수 있는 [AWS 프리 티어](#)를 제공합니다. 12개월 이전에 AWS 계정을 생성했지만 Amazon EC2에 대한 프리 티어 혜택을 아직 다 사용하지 않은 경우 프리 티어 혜택 안에 포함된 옵션을 선택하는 데 도움이 되는 이 자습서를 무료로 이용할 수 있습니다. 그렇지 않을 경우, 유료 상태로 유지되더라도 인스턴스를 시작하는 시점부터 인스턴스를 종료할 때까지(이 자습서의 최종 작업) 스탠다드 Amazon EC2 사용료가 발생합니다.

목차

- [개요 \(p. 21\)](#)
- [사전 조건 \(p. 22\)](#)
- [1단계: 인스턴스 시작 \(p. 22\)](#)
- [2단계: 인스턴스에 연결 \(p. 23\)](#)
- [3단계: 인스턴스 정리 \(p. 24\)](#)
- [다음 단계 \(p. 24\)](#)

개요

이 인스턴스는 Amazon EBS 지원 인스턴스(루트 볼륨이 EBS 볼륨임을 의미)입니다. 인스턴스가 실행되는 가용 영역을 지정하거나 적합한 가용 영역이 Amazon EC2에서 자동으로 선택할 수 있습니다. 인스턴스를 시작할 때 키 페어와 보안 그룹을 지정하여 인스턴스 보안을 설정합니다. 인스턴스에 연결할 때는 인스턴스 시작 시 지정한 키 페어의 프라이빗 키를 지정해야 합니다.

작업

이 자습서를 완료하려면 다음 작업을 수행하십시오.

1. [인스턴스 시작 \(p. 22\)](#)

-
- 2. 인스턴스에 연결 (p. 23)
 - 3. 인스턴스 정리 (p. 24)

관련 자습서

- Windows 인스턴스를 시작하려면 이 자습서의 Windows 인스턴스용 Amazon EC2 사용 설명서: [Getting Started with Amazon EC2 Windows Instances](#) 섹션을 참조하십시오.
- 명령줄을 사용하려는 경우 AWS Command Line Interface 사용 설명서의 [AWS CLI를 통해 Amazon EC2를 사용하는 방법](#) 자습서를 참조하십시오.

사전 조건

시작하기 전에 먼저 [Amazon EC2로 설정 \(p. 16\)](#)의 단계를 완료해야 합니다.

1단계: 인스턴스 시작

다음 절차의 설명에 따라 AWS Management Console을 사용하여 Linux 인스턴스를 시작할 수 있습니다. 이 자습서는 첫 번째 인스턴스를 빠르게 시작하도록 돕기 위한 것이므로 가능한 모든 옵션을 다루지는 않습니다. 어드밴스 옵션에 대한 자세한 내용은 [Launching an Instance](#) 섹션을 참조하십시오.

인스턴스를 시작하려면 다음을 수행합니다.

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 콘솔 대시보드에서 [Launch Instance]를 선택합니다.
3. [Choose an Amazon Machine Image (AMI)] 페이지에 인스턴스에 대한 템플릿 역할을 하는 [Amazon Machine Images (AMIs)]라는 기본 구성 목록이 표시됩니다. Amazon Linux AMI의 HVM 버전을 선택합니다. 이 AMI는 "Free tier eligible"로 표시됩니다.
4. [Choose an Instance Type] 페이지에서 인스턴스의 하드웨어 구성을 선택할 수 있습니다. 기본적으로 선택된 t2.micro 유형을 선택합니다. 이 인스턴스 유형은 프리 티어에 적격입니다.

Note

t2.micro와 같은 [T2 인스턴스](#)는 VPC로 시작되어야 합니다. AWS 계정에서 EC2-Classic을 지원하며 선택한 리전에 VPC가 없는 경우 시작 마법사에서 VPC가 자동으로 생성되므로 다음 단계로 진행할 수 있습니다. 그렇지 않으면 [Review and Launch] 버튼이 비활성화되며 [Next: Configure Instance Details]를 선택하고 지침을 따라 서브넷을 선택해야 합니다.

5. [Review and Launch]를 선택하여 마법사가 다른 구성 설정을 완료하게 합니다.
6. [Review Instance Launch] 페이지의 [Security Groups] 아래에서 마법사가 보안 그룹을 만들고 선택했음을 확인합니다. 이 보안 그룹을 사용하거나, 다음 단계를 이용하여 설정을 시작할 때 만든 보안 그룹을 선택합니다.
 - a. [Edit security groups]를 선택합니다.
 - b. [Configure Security Group] 페이지에서 [Select an existing security group]이 선택되어 있는지 확인합니다.
 - c. 기존 보안 그룹 목록에서 보안 그룹을 선택한 다음 [Review and Launch]를 선택합니다.
7. [Review Instance Launch] 페이지에서 [Launch]를 선택합니다.
8. 키 페어에 대한 메시지가 나타나면 [Choose an existing key pair]를 선택한 다음 설치할 때 생성한 키 페어를 선택합니다.

또는 키 페어를 새로 만들 수 있습니다. [Create a new key pair]를 선택하고 키 페어 이름을 입력한 다음 [Download Key Pair]를 선택합니다. 이때가 사용자가 프라이빗 키 파일을 저장할 수 있는 유일한 기회이

므로 반드시 다운로드하십시오. 프라이빗 키 파일은 안전한 장소에 저장합니다. 인스턴스를 시작할 때 키 페어의 이름을 제공하고, 인스턴스에 연결할 때마다 해당 프라이빗 키를 제공해야 합니다.

Warning

[Proceed without a key pair] 옵션을 선택하지 마십시오. 키 쌍 없이 인스턴스를 시작하면 인스턴스에 연결할 수 없습니다.

준비되면 승인 확인란을 선택한 다음, [Launch Instances]를 선택합니다.

9. 확인 페이지에서 인스턴스가 실행 중인지 확인할 수 있습니다. View Instances를 선택하여 확인 페이지를 닫고 콘솔로 돌아갑니다.
10. [Instances] 화면에서 시작 상태를 볼 수 있습니다. 인스턴스를 시작하는 데 약간 시간이 걸립니다. 인스턴스를 시작할 때 초기 상태는 pending입니다. 인스턴스가 시작된 후에는 상태가 [running]으로 바뀌고 퍼블릭 DNS 이름을 받습니다. ([Public DNS (IPv4)] 열이 숨겨져 있는 경우 페이지 오른쪽 상단 모서리에 있는 [Show/Hide] 아이콘을 선택한 다음 [Public DNS (IPv4)]를 선택합니다.)
11. 연결할 수 있도록 인스턴스가 준비될 때까지 몇 분 정도 걸릴 수 있습니다. 인스턴스가 상태 확인을 통과했는지 확인하십시오. [Status Checks] 열에서 이 정보를 볼 수 있습니다.

2단계: 인스턴스에 연결

Linux 인스턴스에 연결하는 몇 가지 방법이 있습니다. 이 절차에서는 브라우저를 사용해 연결합니다. 그 대신 PuTTY 또는 SSH 클라이언트를 사용하여 연결할 수 있습니다. 또한 이전 단계에 따라 특정 사용자 이름이 있는 Amazon Linux AMI에서 인스턴스를 시작했다고 가정합니다. 다른 Linux 배포판에서는 다른 사용자 이름을 사용할 수 있습니다. 자세한 내용은 [PuTTY를 사용하여 Windows에서 Linux 인스턴스에 연결 \(p. 278\)](#) 또는 [SSH를 사용하여 Linux 인스턴스에 연결 \(p. 274\)](#)을 참조하십시오.

Important

.pem 파일이 있는 키 페어를 사용하여, 그리고 SSH 액세스를 허용하는 보안 그룹을 사용하여 인스턴스를 시작하지 않았다면 인스턴스에 연결할 수 없습니다. 인스턴스에 연결할 수 없는 경우 지원이 필요하면 [인스턴스 연결 문제 해결 \(p. 699\)](#)을 참조하십시오.

웹 브라우저를 사용하여 Linux 인스턴스에 연결하려면

1. 브라우저에 Java가 설치되어 사용할 수 있어야 합니다. 아직 Java를 설치하지 않은 경우 [Java 설치 및 웹 브라우저에서 Java를 사용으로 설정하는 방법은 무엇입니까?](#)에 설명된 단계를 따르십시오.
2. Amazon EC2 콘솔의 탐색 창에서 [Instances]를 선택합니다.
3. 인스턴스를 선택한 다음 [Connect]를 선택합니다.
4. [A Java SSH client directly from my browser (Java required)]를 선택합니다.
5. Amazon EC2에서 인스턴스의 퍼블릭 DNS 이름을 자동으로 검색하여 그 이름으로 Public DNS를 채웁니다. 또한 인스턴스를 시작할 때 지정한 키 페어도 검색합니다. 다음 절차를 완료하고 [Launch SSH Client]를 선택합니다.
 - a. [User name]에 ec2-user를 입력합니다.
 - b. [Private key path]에 키 페어 이름을 포함하는 프라이빗 키(.pem) 파일의 정규화된 경로를 입력합니다.
 - c. (선택 사항) 브라우저 캐시에 프라이빗 키의 위치를 저장하려면 [Store in browser cache]를 선택합니다. 이렇게 하면 Amazon EC2에서는 사용자가 브라우저 캐시를 지울 때까지 이후 브라우저 세션에서 프라이빗 키 위치를 검색할 수 있습니다.
6. 필요할 경우 [Yes]를 선택하여 인증서를 신뢰할 수 있음을 확인하고 [Run]을 선택하여 MindTerm 클라이언트를 실행합니다.
7. MindTerm을 처음 실행하는 경우, 라이선스 계약에 대한 동의 여부, 흄 디렉터리 설정에 대한 확인 여부 및 알려진 호스트 디렉터리 설정에 대한 확인 여부를 묻는 일련의 대화 상자가 표시됩니다. 해당 설정을 확인합니다.

8. 알려진 호스트 세트에 호스트를 추가할지 묻는 대화 상자가 표시됩니다. 로컬 컴퓨터에 호스트 키 정보를 저장하지 않으려면 [No]를 선택합니다.
9. 창이 열리고 인스턴스에 연결됩니다.

Note

이전 단계에서 [No]를 선택한 경우 다음과 같은 메시지가 나타납니다.

Verification of server key disabled in this session.

3단계: 인스턴스 정리

이 자습서용으로 생성한 인스턴스와 볼륨을 완료한 후에는 인스턴스를 종료하여 정리해야 합니다. 정리하기 전에 이 인스턴스로 추가 연습을 수행하려는 경우 [다음 단계 \(p. 24\)](#)를 참조하십시오.

Important

인스턴스를 종료하면 인스턴스가 실제로 삭제되므로 인스턴스를 종료한 후에는 인스턴스에 다시 연결할 수 없습니다.

[AWS 프리 티어](#) 밖에 있는 인스턴스를 시작한 경우 인스턴스 상태가 `shutting down` 또는 `terminated`로 변경되는 즉시 해당 인스턴스에 대한 요금 발생이 중지됩니다. 나중에 사용하기 위해 인스턴스를 보관하지만 요금이 발생하지 않도록 하려면 지금 인스턴스를 중지한 다음 나중에 다시 시작할 수 있습니다. 자세한 내용은 [인스턴스 중단](#)을 참조하십시오.

인스턴스를 종료하려면

1. 탐색 창에서 [`Instances`]를 선택합니다. 인스턴스 목록에서 인스턴스를 선택합니다.
2. [`Actions`]를 선택하고 [`Instance State`]를 선택한 후 [`Terminate`]를 선택합니다.
3. 확인 메시지가 나타나면 [`Yes, Terminate`]를 선택합니다.

Amazon EC2가 인스턴스를 종료합니다. 인스턴스는 종료한 후에도 잠시 동안 콘솔에서 표시된 상태로 유지되며, 그 이후 항목이 삭제됩니다.

다음 단계

인스턴스를 시작한 후 다음 연습을 시도할 수 있습니다.

- Run Command를 사용하여 EC2 인스턴스를 원격으로 관리하는 방법을 알아봅니다. 자세한 내용은 [자습서: Amazon EC2 인스턴스 원격 관리 \(p. 57\)](#) and [시스템 관리자 Remote Management \(Run Command\)](#) 단원을 참조하십시오.
- 사용량이 프리 티어 한도를 초과하는 경우 알려 주는 CloudWatch 경보를 구성합니다. 자세한 내용은 AWS Billing and Cost Management 사용 설명서의 [청구 경보 생성](#)을 참조하십시오.
- EBS 볼륨을 추가합니다. 자세한 내용은 [Amazon EBS 볼륨 생성 \(p. 573\)](#) 및 [Amazon EBS 볼륨을 인스턴스에 연결 \(p. 576\)](#) 섹션을 참조하십시오.
- LAMP 스택을 설치합니다. 자세한 내용은 [자습서: Amazon LinuxLAMP 웹 서버 설치 \(p. 27\)](#) 섹션을 참조하십시오.

Amazon EC2 모범 사례

이 체크리스트를 사용하면 Amazon EC2의 이점을 최대한 활용하여 만족도를 높일 수 있습니다.

보안 및 네트워크

- 자격 증명 연동, IAM 사용자 및 IAM 역할을 사용하여 AWS 리소스 및 API에 대한 액세스를 관리합니다. AWS 액세스 자격 증명의 생성, 배포, 순환 및 취소를 위한 자격 증명 관리 정책과 절차를 설정합니다. 자세한 내용은 IAM 사용 설명서에서 [IAM 모범 사례](#)를 참조하십시오.
- 보안 그룹에 대한 최소 허용 규칙을 구현합니다. 자세한 내용은 [보안 그룹 규칙 \(p. 386\)](#) 섹션을 참조하십시오.
- 인스턴스에서 운영 체제와 애플리케이션을 정기적으로 패치, 업데이트 및 보안합니다. Amazon Linux 업데이트에 대한 자세한 내용은 [Managing Software on Your Linux Instance](#) 섹션을 참조하십시오. Windows 인스턴스 업데이트에 대한 자세한 내용은 Windows 인스턴스용 Amazon EC2 사용 설명서의 [Windows 인스턴스 업데이트](#) 섹션을 참조하십시오.
- EC2-Classic 대신 VPC로 인스턴스를 시작합니다. 2013년 12월 4일 이후에 AWS 계정을 생성한 경우 인스턴스가 자동으로 VPC로 시작됩니다. 이점에 대한 자세한 내용은 [Amazon EC2와 Amazon Virtual Private Cloud \(p. 466\)](#) 섹션을 참조하십시오.

스토리지

- 루트 디바이스 유형이 데이터 지속성, 백업 및 복구에 미치는 영향을 이해합니다. 자세한 내용은 [루트 디바이스 스토리지 \(p. 64\)](#) 섹션을 참조하십시오.
- 운영 체제와 데이터에 대해 별도의 Amazon EBS 볼륨을 사용합니다. 데이터를 포함하는 볼륨이 인스턴스 종료 이후에 지속되는지 확인합니다. 자세한 내용은 [인스턴스 종료 시 Amazon EBS 볼륨 보존 \(p. 294\)](#) 섹션을 참조하십시오.
- 인스턴스에서 임시 데이터를 저장하는 데 사용 가능한 인스턴스 스토어를 사용합니다. 인스턴스를 중지하거나 종료하면 인스턴스 스토어에 저장된 데이터가 삭제됩니다. 인스턴스 스토어를 데이터베이스 스토리지용으로 사용하는 경우 내결합성을 보장하는 복제 인자를 가진 클러스터가 있어야 합니다.

리소스 관리

- 인스턴스 메타데이터 및 사용자 지정 리소스 태그를 사용하여 AWS 리소스를 추적하고 식별합니다. 자세한 내용은 [인스턴스 메타데이터 및 사용자 데이터 \(p. 321\)](#) 및 [Amazon EC2 리소스에 태그 지정 \(p. 681\)](#) 섹션을 참조하십시오.
- Amazon EC2에 대한 현재 제한을 조회합니다. 실제로 필요할 시점보다 미리 제한 증가를 요청하도록 계획하십시오. 자세한 내용은 [Amazon EC2 서비스 제한 \(p. 688\)](#) 섹션을 참조하십시오.

백업 및 복구

- Amazon EBS 스냅샷 ([p. 607](#))을 사용하여 EBS 볼륨을 정기적으로 백업하고, 인스턴스에서 [Amazon 머신 이미지\(AMI\)](#) ([p. 62](#))를 만들어 추후 인스턴스 시작을 위한 템플릿으로 구성을 저장합니다.
- 애플리케이션의 주요 구성 요소를 여러 가용 영역에 배포하고 데이터를 적절히 복제합니다.
- 인스턴스를 다시 시작할 때 IP 주소를 동적으로 지정하도록 애플리케이션을 설계합니다. 자세한 내용은 [Amazon EC2인스턴스 IP 어드레싱](#) ([p. 490](#)) 섹션을 참조하십시오.
- 이벤트 모니터링 및 응답. 자세한 내용은 [Amazon EC2 모니터링](#) ([p. 336](#)) 섹션을 참조하십시오.
- 장애 조치를 처리할 수 있도록 준비해야 합니다. 기본 솔루션의 경우 네트워크 인터페이스 또는 탄력적 IP 주소를 대체 인스턴스에 수동으로 연결할 수 있습니다. 자세한 내용은 [탄력적 네트워크 인터페이스](#) ([p. 512](#))을 참조하십시오. 자동 솔루션의 경우 Auto Scaling을 사용할 수 있습니다. 자세한 내용은 [Auto Scaling 사용 설명서](#) 섹션을 참조하십시오.
- 장애가 발생할 경우에 대비하여 인스턴스 및 Amazon EBS 볼륨의 복구 프로세스를 정기적으로 테스트합니다.

Linux를 실행하는 Amazon EC2 인스턴스에 대한 자습서

다음 자습서에서는 Linux를 실행하는 EC2 인스턴스를 사용하는 일반 작업을 수행하는 방법을 설명합니다.

자습서

- [자습서: Amazon LinuxLAMP 웹 서버 설치 \(p. 27\)](#)
- [자습서: Amazon Linux를 통한 WordPress 블로그 호스팅 \(p. 37\)](#)
- [자습서: SSL/TLS를 사용하여 Amazon Linux에서 Apache 웹 서버 구성 \(p. 46\)](#)
- [자습서: Amazon EC2에서 애플리케이션의 가용성 향상 \(p. 54\)](#)
- [자습서: Amazon EC2 인스턴스 원격 관리 \(p. 57\)](#)

자습서: Amazon LinuxLAMP 웹 서버 설치

다음 절차를 통해 Amazon Linux 인스턴스에서 PHP 및 MySQL 지원을 포함하는 Apache 웹 서버를 설치할 수 있습니다. 이 웹 서버는 LAMP 웹 서버 또는 LAMP 스택이라고 불리기도 합니다. 이 서버를 사용해서 고정 웹사이트를 호스팅하거나 데이터베이스에서 정보를 읽고 쓰는 동적 PHP 애플리케이션을 배포할 수 있습니다.

사전 요구사항

본 자습서는 사용자가 인터넷에서 접근할 수 있는 퍼블릭 DNS 이름을 가진 새 인스턴스를 이미 실행한 것으로 가정하고 있습니다. 자세한 내용은 [1단계: 인스턴스 시작 \(p. 22\)](#) 섹션을 참조하십시오. 보안 그룹이 [SSH\(포트 22\)](#), [HTTP\(포트 80\)](#), [HTTPS\(포트 443\)](#) 연결을 허용하도록 구성되어야 합니다. 이 사전 요구사항에 대한 자세한 내용은 [Amazon EC2로 설정 \(p. 16\)](#)을 참조하십시오.

Important

LAMP 웹 서버를 Ubuntu 인스턴스에서 설치하려는 경우는 본 자습서를 이용할 수 없습니다. 이 절차는 Amazon Linux에서 사용하기 위한 것입니다. 기타 배포에 대한 자세한 내용은 해당 배포의 특정 문서를 참조하십시오. Ubuntu의 LAMP 웹 서버에 대한 자세한 내용은 Ubuntu 커뮤니티 문서 [ApacheMySQLPHP 항목](#)을 참조하십시오.

Amazon Linux에서 LAMP 웹 서버 설치 및 시작

1. [인스턴스에 연결합니다 \(p. 23\).](#)
2. 모든 소프트웨어 패키지가 최신 상태로 업데이트되어 있는지 확인하기 위해, 인스턴스에서 쿼크 소프트웨어 업데이트를 실행합니다. 이 업데이트 과정은 몇 분 정도 시간이 소요될 수 있지만, 최신 보안 업데이트와 버그 수정을 위해 수행할 필요가 있습니다.

Note

-y 옵션을 사용하면 확인 여부를 물지 않고 업데이트를 설치합니다. 설치 전에 업데이트 정보를 확인하려면 이 옵션을 생략합니다.

```
[ec2-user ~]$ sudo yum update -y
```

3. 이제 인스턴스가 최신 상태이므로 Apache 웹 서버, MySQL, PHP 소프트웨어 패키지를 설치할 수 있습니다.

Note

일부 애플리케이션은 다음 권장 소프트웨어 환경과 호환되지 않을 수 있습니다. 이러한 패키지를 설치하기 전에 LAMP 애플리케이션(예: WordPress 또는 phpMyAdmin)이 패키지와 호환되는지 확인하십시오. 문제가 있는 경우, [서버에서 실행할 애플리케이션 소프트웨어가 설치된 PHP 버전 또는 다른 소프트웨어와 호환되지 않습니다 \(p. 36\)](#)의 설명에 따라 다른 환경을 설치해야 할 수 있습니다.

yum install 명령을 사용하여 여러 소프트웨어 패키지와 모든 관련 종속 프로그램을 동시에 설치합니다.

```
[ec2-user ~]$ sudo yum install -y httpd24 php70 mysql56-server php70-mysqld
```

4. Apache 웹 서버를 시작합니다.

```
[ec2-user ~]$ sudo service httpd start
Starting httpd: [ OK ]
```

5. [chkconfig] 명령을 사용해서 Apache 웹 서버가 매번 시스템이 부팅할 때마다 시작되도록 합니다.

```
[ec2-user ~]$ sudo chkconfig httpd on
```

Tip

[chkconfig] 명령을 성공적으로 사용해 서비스를 활성화했을 경우에는 아무런 확인 메시지를 표시하지 않습니다.
다음 명령을 실행하여 httpd가 실행되고 있는지 확인할 수 있습니다.

```
[ec2-user ~]$ chkconfig --list httpd
httpd      0:off    1:off    2:on     3:on     4:on     5:on     6:off
```

여기에서 httpd는 2, 3, 4, 5의 실행 레벨(사용자가 보기 원하는 부분)에서 on 상태입니다.

6. 웹 서버를 테스트합니다. 웹 브라우저에서 인스턴스의 퍼블릭 DNS 주소 또는 퍼블릭 IP 주소를 입력합니다. 이 경우 Apache 테스트 페이지를 볼 수 있어야 합니다. 사용자의 인스턴스에 대한 퍼블릭 DNS를 Amazon EC2 콘솔을 사용해서 얻을 수 있습니다([Public DNS] 열 확인). 이 열이 숨겨진 경우는 [Show/ Hide]를 선택하고 [Public DNS]를 선택합니다.

Tip

Apache 테스트 페이지를 볼 수 없는 경우, 사용 중인 보안 그룹에 HTTP(포트 80) 트래픽을 허용하는 규칙이 있는지 확인하십시오. HTTP 규칙을 보안 그룹에 추가하는 것에 대한 자세한 내용은 다음([보안 그룹에 규칙 추가 \(p. 390\)](#))을 참조하십시오.

Important

Amazon Linux을 사용하지 않는 경우, 이러한 연결을 허용하도록 인스턴스의 방화벽을 구성할 필요가 있습니다. 방화벽 구성 방법에 대한 자세한 내용은 사용자의 특정 배포에 대한 문서를 참조하십시오.

Amazon Linux AMI Test Page

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the web server installed at this site is working properly, but has not yet been configured.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

The [Amazon Linux AMI](#) is a supported and maintained Linux image provided by [Amazon Web Services](#) for use on [Amazon Elastic Compute Cloud \(Amazon EC2\)](#). It is designed to provide a stable, secure, and high performance execution environment for applications running on [Amazon EC2](#). It also includes packages that enable easy integration with [AWS](#), including launch configuration tools and many popular AWS libraries and tools. [Amazon Web Services](#) provides ongoing security and maintenance updates to all instances running the [Amazon Linux AMI](#). The [Amazon Linux AMI](#) is provided at no additional charge to [Amazon EC2 users](#).

Note

테스트 페이지는 `/var/www/html`에 아무 콘텐츠가 없는 경우에만 표시됩니다. 문서 루트에 콘텐츠를 추가하면 이 콘텐츠는 테스트 페이지 대신 인스턴스의 퍼블릭 DNS 주소에 나타납니다.

Apache httpd는 'Acache document root'라는 디렉터리에 보관된 파일을 처리합니다. Amazon Linux Apache document root는 `/var/www/html`이며, 이는 기본적으로 `root`가 소유권을 가지고 있습니다.

```
[ec2-user ~]$ ls -l /var/www
total 16
drwxr-xr-x 2 root root 4096 Jul 12 01:00 cgi-bin
drwxr-xr-x 3 root root 4096 Aug 7 00:02 error
drwxr-xr-x 2 root root 4096 Jan 6 2012 html
drwxr-xr-x 3 root root 4096 Aug 7 00:02 icons
```

`ec2-user`가 해당 디렉터리의 파일을 조작할 수 있도록 하려면 디렉터리의 소유권과 권한을 변경해야 합니다. 이 작업을 수행할 수 있는 방법은 여러 가지가 있습니다. 이 자습서에서는 `www` 그룹을 인스턴스에 추가하고 해당 그룹에 대해 `/var/www` 디렉터리의 소유권을 부여하고 쓰기 권한을 추가하겠습니다. 해당 그룹의 모든 멤버는 웹 서버에 대해서 파일의 추가, 삭제, 수정을 할 수 있습니다.

If you are the website administrator:

You may now add content to the directory `/var/www/html`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the images below on Apache and Amazon Linux AMI powered HTTP servers. Thanks for using Apache and the Amazon Linux AMI!



2.4

파일 권한 설정

- www 그룹을 인스턴스에 추가합니다.

```
[ec2-user ~]$ sudo groupadd www
```

- 사용자(이 경우는 ec2-user)를 www 그룹에 추가합니다.

```
[ec2-user ~]$ sudo usermod -a -G www ec2-user
```

Important

로그아웃했다가 다시 로그인해서 새 그룹을 선택해야 합니다. [exit] 명령을 사용하거나 터미널 창을 닫을 수 있습니다.

- 로그아웃을 하고 다시 로그인한 다음, www 그룹에 대한 멤버십을 확인하십시오.

- 로그아웃을 합니다.

```
[ec2-user ~]$ exit
```

- 인스턴스에 다시 연결한 다음, 다음 명령을 실행해서 www 그룹에 대한 멤버십을 확인하십시오.

```
[ec2-user ~]$ groups  
ec2-user wheel www
```

- /var/www 및 그 콘텐츠의 그룹 소유권을 www 그룹으로 변경합니다.

```
[ec2-user ~]$ sudo chown -R root:www /var/www
```

- /var/www 및 그 하위 디렉터리의 디렉터리 권한을 변경해서 그룹 쓰기 권한을 추가하고 미래 하위 디렉터리에서 그룹 ID를 설정합니다.

```
[ec2-user ~]$ sudo chmod 2775 /var/www  
[ec2-user ~]$ find /var/www -type d -exec sudo chmod 2775 {} \;
```

- /var/www 및 그 하위 디렉터리의 파일 권한을 계속 변경해서 그룹 쓰기 권한을 추가합니다.

```
[ec2-user ~]$ find /var/www -type f -exec sudo chmod 0664 {} \;
```

이제 ec2-user 및 www 그룹의 향후 멤버는 Apache document root에서 파일의 추가, 삭제, 수정을 할 수 있습니다. 이제, 정적 웹사이트 또는 PHP 애플리케이션 등 콘텐츠를 추가할 수 있습니다.

(선택 사항) 웹 서버 보안

HTTP 프로토콜을 실행하는 웹 서버는 송신하거나 수신하는 데이터에 대해 아무런 전송 보안 기능도 제공하지 않습니다. 웹 브라우저를 사용하여 HTTP 서버에 연결할 때 자신이 입력하는 URL, 수신하는 웹 페이지의 내용, 제출하는 HTML 양식의 내용(암호 포함)이 모두 네트워크 경로를 따라 어디서든 엿보려는 사람들에게 보입니다. 웹 서버를 안전하게 보호하기 위한 최선의 방법은 SSL/TLS 암호화로 데이터를 보호하는 HTTPS(HTTP Secure) 지원 기능을 설치하는 것입니다.

서버에서 HTTPS를 사용하는 자세한 방법은 [자습서: SSL/TLS를 사용하여 Amazon Linux에서 Apache 웹 서버 구성](#)을 참조하십시오.

LAMP 웹 서버 테스트

서버가 설치되고 실행되고 있으며 파일 권한이 올바르게 설정된 경우라면, 사용자의 ec2-user 계정을 통해 인터넷에서 사용 가능한 /var/www/html 디렉터리에서 간단한 PHP 파일을 생성할 수 있어야 합니다.

- Apache 문서 루트에서 간단한 PHP 파일을 생성합니다.

```
[ec2-user ~]$ echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
```

Tip

이 명령을 실행하는 동안 "Permission denied" 오류가 발생하면, 로그아웃하고 다시 로그인한 다음, [파일 권한 설정 \(p. 30\)](#)에서 구성한 적절한 그룹 권한을 선택합니다.

- 웹 브라우저에서는 방금 생성한 파일의 URL을 입력합니다. 이 URL은 인스턴스의 퍼블릭 DNS 주소에 슬래시(/)와 파일 이름이 추가된 형태입니다. 예:

```
http://my.public.dns.amazonaws.com/phpinfo.php
```

PHP 정보 페이지가 표시되어야 합니다:

PHP Version 5.6.6

System	Linux ip-172-31-7-35 3.14.35-28.38.amzn1.x86_64 #1 SMP Wed Mar 11 22:50:37 UTC 2015 x86_64
Build Date	Mar 5 2015 23:26:53
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php-5.6.d
Additional .ini files parsed	/etc/php-5.6.d/20-bz2.ini, /etc/php-5.6.d/20-calendar.ini, /etc/php-5.6.d/20-ctype.ini, /etc/php-5.6.d/20-dom.ini, /etc/php-5.6.d/20-exif.ini, /etc/php-5.6.d/20-fileinfo.ini, /etc/php-5.6.d/20-ftp.ini, /etc/php-5.6.d/20-gettext.ini, /etc/php-5.6.d/20-iconv.ini, /etc/php-5.6.d/20-mysqlnd.ini, /etc/php-5.6.d/20-phar.ini, /etc/php-5.6.d/20-posix.ini, /etc/php-5.6.d/20-shmop.ini, /etc/php-5.6.d/20-simplexml.ini, /etc/php-5.6.d/20-sockets.ini, /etc/php-5.6.d/20-sqlite3.ini, /etc/php-5.6.d/20-sysvmsg.ini, /etc/php-5.6.d/20-sysvshm.ini, /etc/php-5.6.d/20-tokenizer.ini, /etc/php-5.6.d/20-xml.ini, /etc/php-5.6.d/20-xmlwriter.ini, /etc/php-5.6.d/20-xsl.ini, /etc/php-5.6.d/20-zip.ini, /etc/php-5.6.d/30-mysqli.ini, /etc/php-5.6.d/30-mysqli.ini, /etc/php-5.6.d/30-pdo_mysql.ini, /etc/php-5.6.d/30-pdo_sqlite.ini, /etc/php-5.6.d/30-wddx.ini, /etc/php-5.6.d/30-xmlreader.ini, /etc/php-5.6.d/40-json.ini, /etc/php-5.6.d/php.ini
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API20131226,NTS
PHP Extension Build	API20131226,NTS

Note

이 페이지가 보이지 않을 경우 이전 단계에서 `/var/www/html/phpinfo.php` 파일이 제대로 생성되었는지 확인하십시오. 또한 다음 명령을 사용해 필요한 패키지가 모두 설치되었는지 확인할 수 있습니다(이 예제 출력에서 두 번째 열의 패키지 버전은 일치할 필요가 없음).

```
[ec2-user ~]$ sudo yum list installed httpd24 php70 mysql56-server php70-mysqlnd
Loaded plugins: priorities, update-motd, upgrade-helper
Installed Packages
httpd24.x86_64                                2.4.25-1.68.amzn1
@amzn-updates
mysql56-server.x86_64                            5.6.35-1.23.amzn1
@amzn-updates
php70.x86_64                                    7.0.14-1.20.amzn1
@amzn-updates
php70-mysqlnd.x86_64                           7.0.14-1.20.amzn1
@amzn-updates
```

출력에서 필요한 패키지가 하나라도 나열되지 않으면, sudo yum install **package** 명령을 사용해서 패키지를 설치합니다.

3. `phpinfo.php` 파일을 삭제합니다. 이 파일은 사용자에게 유용한 정보를 포함하고 있지만 보안상 이유로 인터넷에 공개되어서는 안 됩니다.

```
[ec2-user ~]$ rm /var/www/html/phpinfo.php
```

MySQL 서버 보안 유지

MySQL 서버의 기본 설치는 테스트 및 개발 기능에 유용한 여러 기능을 포함하고 있지만, 이 기능들은 프로덕션 서버에서는 비활성화되거나 제거되어야 합니다. [mysql_secure_installation] 명령을 통해 루트 암호를 설정하고 설치 패키지에서 보안성이 낮은 기능을 제거하는 과정을 수행할 수 있습니다. MySQL 서버를 사용할 계획이 없더라도 이 절차를 수행하는 것이 도움이 될 수 있습니다.

1. MySQL 서버를 시작합니다.

```
[ec2-user ~]$ sudo service mysqld start
Initializing MySQL database:
...
PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
...
Starting mysqld: [ OK ]
```

2. `mysql_secure_installation`을 실행합니다.

```
[ec2-user ~]$ sudo mysql_secure_installation
```

- a. 요청을 받으면 `root` 계정에 대한 암호를 입력합니다.

- i. 현재 `root` 암호를 입력합니다. 기본적으로 `root` 계정은 암호 세트를 가지고 있지 않으므로 Enter를 누릅니다.
- ii. Y를 눌러서 암호 설정 모드로 들어간 다음, 안전한 암호를 두 번 입력합니다. 안전한 암호 생성에 대한 자세한 내용은 <http://www.pctools.com/guides/password/> 섹션을 참조하십시오. 이 암호를 안전한 장소에 보관하시기 바랍니다.

Note

MySQL에 대한 루트 암호를 설정하는 것은 데이터베이스를 보호하는 가장 기초적인 방법일 뿐입니다. 데이터베이스 기반 애플리케이션을 빌드하거나 설치할 때, 일반적으로 그 애플리케이션의 데이터베이스 서비스 사용자를 만들고 데이터베이스 관리 이외의 어떤 목적으로도 루트 계정을 사용하지 못하게 합니다.

- b. Y를 눌러서 익명 사용자 계정을 제거합니다.
 - c. Y를 눌러서 원격 `root` 로그인을 비활성화합니다.
 - d. Y를 눌러서 테스트 데이터베이스를 제거합니다.
 - e. Y를 눌러서 권한 테이블을 다시 로드하고 변경사항을 저장합니다.
3. (옵션) 지금 바로 사용할 계획이 없는 경우라면 MySQL 서버를 중단시킵니다. 필요할 때 서버를 다시 시작할 수 있습니다.

```
[ec2-user ~]$ sudo service mysqld stop
Stopping mysqld: [ OK ]
```

4. (옵션) MySQL 서버가 매번 부팅할 때마다 시작되도록 하려면 다음 명령을 입력합니다.

```
[ec2-user ~]$ sudo chkconfig mysqld on
```

이제 LAMP 웹 서버가 완전히 동작하는 상태가 됩니다. /var/www/html의 Apache document root에 콘텐츠를 추가하면 인스턴스에 대한 퍼블릭 DNS 주소에서 그 콘텐츠를 볼 수 있습니다.

(선택 사항) phpMyAdmin 설치

[phpMyAdmin](#)은 EC2 인스턴스의 MySQL 데이터베이스를 보고 편집하는 데 사용할 수 있는 웹 기반 데이터베이스 관리 도구입니다. Amazon Linux 인스턴스에서 phpMyAdmin을 설치 및 구성하려면 다음 단계를 따르십시오.

Important

Apache에서 SSL/TLS를 활성화하지 않은 한 phpMyAdmin을 사용하여 LAMP 서버에 액세스하지 않는 것이 좋습니다. 그렇지 않으면, 데이터베이스 관리자 암호와 기타 데이터가 인터넷을 통해 안전하지 못한 상태로 전송됩니다. EC2 인스턴스에 안전한 웹 서버를 구성하는 자세한 방법은 [자습서: SSL/TLS를 사용하도록 Amazon Linux에서 Apache 웹 서버 구성](#)을 참조하십시오.

Note

이러한 지침에서는 동일한 기본 PHP 버전이 Amazon Linux 및 Extra Packages for Enterprise Linux(EPEL)에 지정되어 있다고 가정합니다. EPEL 패키지와의 호환성 문제가 발생할 경우, phpMyAdmin을 수동으로 설치하는 것이 좋습니다. 최신 릴리스는 [phpMyAdmin 다운로드 페이지](#)를 참조하십시오. 설치 요구 사항이 Amazon Linux(또는 다른 Linux) 인스턴스 환경과 일치하는지 확인하십시오.

아래의 문제 해결 도움말을 참조하십시오. [서버에서 실행할 애플리케이션 소프트웨어가 설치된 PHP 버전 또는 다른 소프트웨어와 호환되지 않습니다](#) (p. 36)

1. 인스턴스의 Fedora 프로젝트로부터 EPEL(Extra Packages for Enterprise Linux) 리포지토리를 활성화 합니다.

```
[ec2-user ~]$ sudo yum-config-manager --enable epel
```

2. phpMyAdmin 패키지를 설치합니다.

```
[ec2-user ~]$ sudo yum install -y phpMyAdmin
```

Note

메시지가 표시되면 [y]로 답변하여 EPEL 리포지토리용 GPG 키를 가져옵니다.

3. 로컬 시스템으로부터 액세스를 허용하도록 phpMyAdmin 설치를 구성합니다. 기본적으로, phpMyAdmin은 현재 실행되고 있는 서버로부터의 액세스만 허용하지만, Amazon Linux는 웹 브라우저를 포함하지 않으므로 이 구성은 그다지 유용하지 않습니다.
 - a. [whatismyip.com](#)과 같은 서비스를 방문하여 로컬 IP 주소를 확인합니다.
 - b. /etc/httpd/conf.d/phpMyAdmin.conf 파일을 편집하여 다음 명령을 사용해 서버 IP 주소 (127.0.0.1)를 로컬 IP 주소로 대체합니다. 즉, [`your_ip_address`](#)를 이전 단계에서 확인한 로컬 IP 주소로 바꿉니다.

```
[ec2-user ~]$ sudo sed -i -e 's/127.0.0.1/your\_ip\_address/g' /etc/httpd/conf.d/phpMyAdmin.conf
```

4. Apache 웹 서버를 재시작해서 새 구성은 가져옵니다.

```
[ec2-user ~]$ sudo service httpd restart
```

```
Stopping httpd:  
Starting httpd:
```

```
[ OK ]  
[ OK ]
```

- MySQL 서버를 재시작해서 새 구성을 가져옵니다.

```
[ec2-user ~]$ sudo service mysqld restart  
Stopping mysqld:  
Starting mysqld:
```

```
[ OK ]  
[ OK ]
```

- 웹 브라우저에서는 phpMyAdmin 설치의 URL을 입력합니다. 이 URL은 인스턴스의 퍼블릭 DNS 주소에 슬래시(/) 및 phpmyadmin이 추가된 형태입니다. 예:

```
http://my.public.dns.amazonaws.com/phpmyadmin
```

사용자는 phpMyAdmin 로그인 페이지를 볼 수 있어야 합니다:



Note

403 Forbidden 오류가 반환되는 경우, /etc/httpd/conf.d/phpMyAdmin.conf 파일에서 올바른 IP 주소를 설정했는지 확인하십시오. 다음 명령을 사용해 Apache 액세스 로그를 확인하여 Apache 서버가 실제로 어느 IP 주소로부터 요청을 수신하고 있는지 확인할 수 있습니다.

```
[ec2-user ~]$ sudo tail -n 1 /var/log/httpd/access_log | awk '{ print $1 }'
```

205.251.233.48

여기에 반환된 IP 주소를 사용해 Step 3.b (p. 33)를 반복하고 이전에 입력한 잘못된 주소를
여기에 반환된 주소로 바꿉니다(예):

```
[ec2-user ~]$ sudo sed -i -e 's/previous_ip_address/205.251.233.48/g' /etc/httpd/conf.d/phpMyAdmin.conf
```

).

IP 주소를 바꾼 다음 Step 4 (p. 33)를 사용하여 httpd 서비스를 다시 시작합니다.

- 앞서 만든 root 사용자 이름 및 MySQL 루트 암호로 phpMyAdmin 설치에 로그인합니다. phpMyAdmin 사용에 대한 자세한 내용은 [phpMyAdmin 사용 설명서](#)를 참조하십시오.

문제 해결

이 섹션에서는 새 LAMP 서버를 설정하는 동안 발생할 수 있는 일반적인 문제 해결을 위한 제안을 제공합니다.

웹 브라우저를 사용하여 내 서버에 연결할 수 없습니다.

다음을 확인하여 Apache 웹 서버가 실행 중이고 엑세스 가능한지 확인합니다.

- 웹 서버가 실행되고 있습니까? 다음 명령을 실행하여 httpd가 실행되고 있는지 확인할 수 있습니다.

```
[ec2-user ~]$ chkconfig --list httpd
httpd           0:off    1:off    2:on     3:on     4:on     5:on     6:off
```

여기에서 httpd는 2, 3, 4, 5의 실행 레벨(사용자가 보기 원하는 부분)에서 on 상태입니다.

httpd 프로세스가 실행되지 않는 경우 [Amazon Linux에서 LAMP 웹 서버 설치 및 시작 \(p. 27\)](#)에 설명된 단계를 반복합니다.

- 방화벽이 올바르게 구성되었습니까?

Apache 테스트 페이지를 볼 수 없는 경우, 사용 중인 보안 그룹에 HTTP(포트 80) 트래픽을 허용하는 규칙이 있는지 확인하십시오. HTTP 규칙을 보안 그룹에 추가하는 것에 대한 자세한 내용은 다음([보안 그룹에 규칙 추가 \(p. 390\)](#))을 참조하십시오.

서버에서 실행할 애플리케이션 소프트웨어가 설치된 PHP 버전 또는 다른 소프트웨어와 호환되지 않습니다

이 자습서는 최신 버전의 Apache 웹 서버, PHP, MySQL 설치를 권장합니다. 추가 LAMP 애플리케이션을 설치하기 전에 요구 사항을 점검하여 설치된 환경과 호환되는지 확인합니다. 최신 버전의 PHP가 지원되지 않는 경우, 지원되는 이전 구성으로 다운그레이드할 수 있습니다(안전성 보장). 또한 동시에 두 가지 이상의 PHP 버전을 설치하여 최소 노력으로 특정 호환성 문제를 해결할 수 있습니다. 설치되어 있는 여러 개의 PHP 버전 간에 기본 설정을 구성하는 방법에 대한 정보는 [Amazon Linux AMI 2016.09 Release Notes](#)를 참조하십시오.

다운그레이드 방법

테스트를 통과한 이 자습서의 이전 버전에서는 다음 코어 LAMP 패키지를 호출했습니다.

- httpd24
- php56

- mysql55-server
- php56-mysqlnd

이 자습서의 시작 부분에서 권장한 대로 최신 패키지를 이미 설치한 경우, 먼저 이러한 패키지 및 기타 종속 프로그램을 다음과 같이 제거해야 합니다.

```
[ec2-user ~]$ sudo yum remove -y httpd24 php70 mysql56-server php70-mysqlnd perl-DBD-MySQL56
```

그 다음 대체 환경을 설치합니다.

```
[ec2-user ~]$ sudo yum install -y httpd24 php56 mysql55-server php56-mysqlnd
```

권장 환경으로 나중에 업그레이드하려는 경우, 먼저 사용자 지정된 패키지 및 종속 파일을 제거해야 합니다.

```
[ec2-user ~]$ yum remove -y httpd24 php56 mysql55-server php56-mysqlnd perl-DBD-MySQL55
```

이제 자습서의 시작 부분에서 설명한 대로 최신 패키지를 설치할 수 있습니다.

관련 주제

파일을 인스턴스에 전송하거나 웹 서버에 WordPress 블로그를 설치하는 것에 대한 자세한 내용은 다음 주제를 참고하십시오.

- WinSCP를 사용하여 Linux 인스턴스로 파일 전송 (p. 282)
- SCP를 사용하여 Linux에서 Linux 인스턴스로 파일 전송 (p. 276)
- 자습서: Amazon Linux를 통한 WordPress 블로그 호스팅 (p. 37)

이 주제에서 사용되는 명령과 소프트웨어에 대한 자세한 내용은 다음 웹 페이지를 확인해 보십시오.

- Apache 웹 서버: <http://httpd.apache.org/>
- MySQL 데이터베이스 서버: <http://www.mysql.com/>
- PHP 프로그래밍 언어: <http://php.net/>
- chmod 명령: <https://en.wikipedia.org/wiki/Chmod>
- chown 명령: <https://en.wikipedia.org/wiki/Chown>

웹 서버에 대한 도메인 이름을 등록하거나 기존 도메인 이름을 현재 호스트로 이전하는 것에 대한 자세한 내용은 Amazon Route 53 개발자 안내서의 [Creating and Migrating Domains and Subdomains to Amazon Route 53\(Amazon Route 53에서 도메인 및 하위 도메인 생성 및 마이그레이션\)](#)을 참조하십시오.

자습서: Amazon Linux를 통한 WordPress 블로그 호스팅

다음 절차는 Amazon Linux 인스턴스에 대한 WordPress 블로그의 설치, 구성, 보안 설정을 안내합니다. 본 자습서는 기존 호스팅 서비스에서는 일반적이지 않은 WordPress 블로그를 호스팅하는 웹 서버를 사용자가 완전히 제어할 수 있다는 점에서 Amazon EC2 사용에 있어 좋은 입문 기회를 제공합니다.

사용자는 서버에 대한 소프트웨어 패키지를 업데이트하고 보안 패치를 유지관리할 책임이 있습니다. 웹 서버 구성과 직접 상호작용을 요구하지 않는 보다 자동화된 WordPress 설치를 위해, AWS CloudFormation 서비스는 빠른 시작을 지원하는 WordPress 템플릿을 제공합니다. 자세한 내용은 [시작하기](#)(출처: AWS

CloudFormation 사용 설명서)를 참조하십시오. Windows 인스턴스에서 WordPress 블로그를 호스팅하려는 경우, [Deploying a WordPress Blog on Your Amazon EC2 Windows Instance\(Amzon EC2 Windows 인스턴스에 WordPress 블로그 배포\)](#)(출처: Windows 인스턴스용 Amazon EC2 사용 설명서) 섹션을 참조하십시오. 데이터베이스가 분리된 고가용성 솔루션이 필요하다면 AWS Elastic Beanstalk 개발자 안내서에서 [Deploying a High-Availability WordPress Website](#) 단원을 참조하십시오.

Important

이 절차는 Amazon Linux에서 사용하기 위한 것입니다. 기타 배포에 대한 자세한 내용은 해당 배포의 특정 문서를 참조하십시오. 본 자습서에 있는 단계의 상당수가 Ubuntu 인스턴스에서 작동하지 않습니다. Ubuntu 인스턴스에 WordPress를 설치하는 방법은 Ubuntu 설명서에서 [WordPress](#) 섹션을 참조하십시오.

사전 조건

본 자습서는 사용자가 [자습서: Amazon LinuxLAMP 웹 서버 설치 \(p. 27\)](#)의 모든 단계를 수행해서 PHP 및 MySQL 지원을 통해 작동하는 웹 서버로 Amazon Linux 인스턴스를 시작했다고 가정합니다. 또한 본 자습서는 보안 그룹이 HTTP 및 HTTPS 트래픽을 허용하도록 구성하는 단계와 파일 권한이 웹 서버에 맞게 적절하게 설정되어 있는지 확인하는 여러 단계를 포함하고 있습니다. 상기 사전 조건을 수행하지 않은 경우 [자습서: Amazon LinuxLAMP 웹 서버 설치 \(p. 27\)](#) 섹션을 참조해서 이 사전 조건을 충족하고 본 자습서로 다시 돌아와 WordPress 설치를 시작하시기 바랍니다. 규칙을 보안 그룹에 추가하는 것에 대한 자세한 내용은 [보안 그룹에 규칙 추가 \(p. 390\)](#) 섹션을 참조하십시오.

탄력적 IP 주소(EIP)는 WordPress 블로그를 호스팅하는 데 사용 중인 인스턴스와 연결하는 것이 가장 바람직합니다. 인스턴스의 퍼블릭 DNS 주소가 설치 위치를 바꾸거나 위반하는 것을 방지할 수 있기 때문입니다. 자신이 소유하고 있는 도메인 이름을 블로그에 사용하고 싶다면 도메인 이름의 DNS 레코드가 EIP 주소를 가리키도록 업데이트할 수 있습니다(이와 관련하여 도움이 필요하다면 도메인 이름 등록 기관에게 문의하십시오). 실행 중인 인스턴스와 연결되어 있는 EIP 주소는 한 개까지 무료로 사용할 수 있습니다. 자세한 내용은 [탄력적 IP 주소 \(p. 505\)](#) 섹션을 참조하십시오.

블로그에 사용할 도메인 이름이 아직 없을 경우에는 먼저 Amazon Route 53에 도메인 이름을 등록해야 인스턴스의 EIP 주소와 도메인 이름을 서로 연결할 수 있습니다. 자세한 내용은 Amazon Route 53 개발자 안내서에서 [Amazon Route 53을 사용하여 도메인 이름 등록](#)을 참조하십시오.

WordPress 설치

인스턴스에 연결한 후 WordPress 설치 패키지를 다운로드합니다.

WordPress 설치 패키지의 다운로드 및 압축해제 방법

1. wget 명령을 사용해서 최신 WordPress 설치 패키지를 다운로드 합니다. 다음 명령을 사용할 경우 언제나 최신 릴리스를 다운로드합니다.

```
[ec2-user ~]$ wget https://wordpress.org/latest.tar.gz
--2013-08-09 17:19:01--  https://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 66.155.40.249, 66.155.40.250
Connecting to wordpress.org (wordpress.org)|66.155.40.249|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4028740 (3.8M) [application/x-gzip]
Saving to: latest.tar.gz

100%[=====] 4,028,740   20.1MB/s   in 0.2s

2013-08-09 17:19:02 (20.1 MB/s) - latest.tar.gz saved [4028740/4028740]
```

2. 설치 패키지의 압축 및 아카이빙을 해제합니다. 설치 폴더는 wordpress라는 폴더로 압축 해제됩니다.

```
[ec2-user ~]$ tar -xzf latest.tar.gz
[ec2-user ~]$ ls
```

```
latest.tar.gz wordpress
```

WordPress 설치에 대한 MySQL 사용자 및 데이터베이스 생성 방법

WordPress 설치에는 블로그 포스트 항목, 사용자 의견 등 정보를 데이터베이스에 저장할 수 있도록 구성하는 작업을 필요로 합니다. 다음 프로시저를 통해 블로그에 대한 데이터베이스와 데이터베이스에 대한 정보의 읽기/저장 권한을 부여받게 되는 사용자를 생성할 수 있습니다.

- MySQL 서버를 시작합니다.

```
[ec2-user ~]$ sudo service mysqld start
```

- MySQL 서버를 root 사용자로 로그인합니다. 요청받은 경우 MySQL root 암호를 입력합니다. 이 암호는 사용자의 root 시스템 암호와 다를 수 있으며, MySQL 서버를 보안 설정하지 않은 경우 암호가 비어 있을 수도 있습니다.

Important

MySQL 서버를 보안 설정하지 않은 경우, 이를 반드시 수행하시기 바랍니다. 자세한 내용은 [MySQL 서버 보안 유지 \(p. 32\)](#) 섹션을 참조하십시오.

```
[ec2-user ~]$ mysql -u root -p  
Enter password:
```

- MySQL 데이터베이스에 대한 사용자 및 암호를 생성합니다. WordPress 설치는 MySQL 데이터베이스를 통신하기 위해 상기 값을 사용합니다. 고유한 사용자 이름과 암호로 해당 부분을 대체하여 다음 명령을 입력합니다.

```
mysql> CREATE USER 'wordpress-user'@'localhost' IDENTIFIED BY 'your_strong_password';  
Query OK, 0 rows affected (0.00 sec)
```

사용자에 대해 보안이 강력한 암호를 생성하시기 바랍니다. 작은따옴표(')는 각 명령을 구별하는 구분자로 기능하기 때문에, 암호에는 사용하지 마십시오. 안전한 암호 생성에 대한 자세한 내용은 <http://www.pctools.com/guides/password/> 섹션을 참조하십시오. 기존 암호를 재사용하지 마십시오. 새로 설정한 암호는 안전한 장소에 보관해 두십시오.

- 데이터베이스를 생성합니다. 데이터베이스에 이를 설명할 수 있는 유의미한 이름을 붙입니다(예: wordpress-db).

Note

).

아래 명령에서 데이터베이스 이름을 둘러싼 기호(')는 백틱(backtick)이라고 불립니다. 백틱(') 키는 일반적으로 표준 키보드에서 Tab 키 위에 위치합니다. 백틱이 언제나 요구되는 것은 아니지만, 이를 통해 데이터베이스 이름에 하이픈(-) 등 허용되지 않는 문자를 사용할 수 있게 됩니다.

```
mysql> CREATE DATABASE `wordpress-db`;  
Query OK, 1 row affected (0.01 sec)
```

- 데이터베이스에 대한 전체 권한을 이전에 생성한 WordPress 사용자에게 부여합니다.

```
mysql> GRANT ALL PRIVILEGES ON `wordpress-db`.* TO "wordpress-user"@"localhost";  
Query OK, 0 rows affected (0.00 sec)
```

- MySQL 권한을 새로고침(flush)해서 모든 변경사항이 적용되도록 합니다.

```
mysql> FLUSH PRIVILEGES;
```

```
Query OK, 0 rows affected (0.01 sec)
```

7. mysql 클라이언트를 종료합니다.

```
mysql> exit  
Bye
```

wp-config.php 파일 생성 및 편집 방법

WordPress 설치 폴더는 `wp-config-sample.php`라는 샘플 구성 파일을 포함하고 있습니다. 본 절차에서는 이 파일을 복사하고 특정 구성에 맞도록 편집합니다.

1. `wp-config-sample.php` 파일을 `wp-config.php`라는 파일로 이름을 바꿔 복사합니다. 이를 통해 새 구성 파일을 생성하고 원본 샘플 파일을 이전 상태 그대로 백업으로 보존할 수 있습니다.

```
[ec2-user ~]$ cd wordpress/  
[ec2-user wordpress]$ cp wp-config-sample.php wp-config.php
```

2. `wp-config.php` 파일을 원하는 텍스트 편집기(nano, vim 등)로 편집하고 설치에 대한 값을 입력합니다. 원하는 텍스트 편집기가 없는 경우, 초보자는 nano를 사용하는 것이 더욱 편리합니다.

```
[ec2-user wordpress]$ nano wp-config.php
```

- a. `DB_NAME`을(를) 정의하는 줄을 찾고 `database_name_here`을(를) [WordPress 설치에 대한 MySQL 사용자 및 데이터베이스 생성 방법 \(p. 39\)](#)의 Step 4 (p. 39)에서 생성한 데이터베이스 이름으로 변경합니다.

```
define('DB_NAME', 'wordpress-db');
```

- b. `DB_USER`을(를) 정의하는 줄을 찾고 `username_here`을(를) [WordPress 설치에 대한 MySQL 사용자 및 데이터베이스 생성 방법 \(p. 39\)](#)의 Step 3 (p. 39)에서 생성한 데이터베이스 사용자로 변경합니다.

```
define('DB_USER', 'wordpress-user');
```

- c. `DB_PASSWORD`을(를) 정의하는 줄을 찾고 `password_here`을(를) [WordPress 설치에 대한 MySQL 사용자 및 데이터베이스 생성 방법 \(p. 39\)](#)의 Step 3 (p. 39)에서 생성한 보안성이 강력한 암호로 변경합니다.

```
define('DB_PASSWORD', 'your_strong_password');
```

- d. `Authentication Unique Keys and Salts`라는 섹션을 검색합니다. 이 `KEY` 및 `SALT` 값은 WordPress 사용자가 로컬 컴퓨터에 저장하는 브라우저 쿠키에 암호 계층을 제공합니다. 기본적으로 긴 무작위 값을 추가해서 사이트의 보안성을 강화할 수 있습니다. <https://api.wordpress.org/secret-key/1.1/salt/>를 방문해서 키 값의 세트를 무작위로 생성하고 이를 `wp-config.php` 파일로 복사해서 붙여넣기할 수 있습니다. PuTTY 터미널로 텍스트를 붙여넣기 하기 위해, PuTTY 터미널 내부에서 텍스트를 붙여넣기하려는 위치에 커서를 놓고 마우스를 오른쪽 클릭합니다.

보안 키에 대한 자세한 내용은 http://codex.wordpress.org/Editing_wp-config.php#Security_Keys 섹션을 참조하시기 바랍니다.

Note

아래 값은 예시 목적만을 위한 것입니다. 설치할 경우 이 값을 사용하지는 마십시오.

```
define('AUTH_KEY',           '#U$$+[RXN8:b^-L_0(WU_+ c+WFkI~c]o]-bHw+/'
Aj[wTwSiZ<Qb[mghExcRh-']);
define('SECURE_AUTH_KEY',    'zsz._P=l/|y.Lq)XjlkwS1y5NJ76E6EJ.AV0pCKZZB,*~*r ?6OP
$eJ@;+(ndLg');
define('LOGGED_IN_KEY',      'ju}qwre3V*+8f_zOWf?{LlGsQ]Ye@2Jh^,8x>)Y |;(^[Iw]Pi
+LG#A4R?7N`YB3');
define('NONCE_KEY',          'P(g62HeZxEes/LnI^i=H,[XwK9I&[2s|:?ON}VJM%?;v2v)v+;
+^9eXUahg@::Cj');
define('AUTH_SALT',           'C$DpB4Hj[JK:{ql`sRVa{:7yShy(9A@5wg+`JJVb1fk%-_
Bx*M4(qc[Qg%JT!h');
define('SECURE_AUTH_SALT',   'd!uRu#)+q#{f$Z?Z9uFPG.${+S{n~1M&%@~gL>U>NV<zpD-@2-
Es7Q1O-bp28EKv');
define('LOGGED_IN_SALT',     'j{00P*owZf)kVD+FVLn-->. |Y%Ug4#I^*Lvd9QeZ^&XmK/e(76mic
+&W&+^OP/');
define('NONCE_SALT',         '-97r*V/cgxLmp?Zy4zUU4r99QQ_xGs2LTd%P; |
_e1tS)8_B/, .6[=UK<J_y9?JWG');
```

- e. 파일을 저장하고 텍스트 편집기를 종료합니다.

WordPress 설치를 Apache 문서 루트로 이동시키는 방법

설치 폴더 압축을 해제하고 MySQL 데이터베이스 및 맞춤형 WordPress 구성 파일을 사용자 설정했으므로, 이제 설치 파일을 웹 서버 문서 루트에 이동시켜서 설치를 완료하는 설치 스크립트를 실행할 수 있습니다. 이 파일을 이동시킬 위치는 WordPress 블로그를 웹 서버 루트(예: my.public.dns.amazonaws.com) 또는 하위 디렉터리나 폴더(예: my.public.dns.amazonaws.com/blog) 중 어디에서 사용 가능하게 하려는지에 따라 달라집니다.

- 블로그를 사용 가능하게 할 위치를 선택하고 해당 위치에 대해서만 mv 명령을 실행합니다.

Important

아래 명령을 두 차례 실행한 경우, 두 번째 mv 명령에서는 이동하려는 파일이 더 이상 대상 위치에 존재하지 않기 때문에 오류 메시지가 발생합니다.

- 블로그를 my.public.dns.amazonaws.com에서 사용 가능하게 하려면 wordpress 폴더의 파일(폴더는 제외)을 Apache 문서 루트에 이동시킵니다(Amazon Linux 인스턴스의 /var/www/html).

```
[ec2-user wordpress]$ mv * /var/www/html/
```

- 또는, 블로그를 대신 my.public.dns.amazonaws.com/blog에서 사용 가능하게 하려면 Apache 문서 루트 내부에 blog라는 새 폴더를 생성하고 wordpress 폴더의 파일(폴더는 제외)을 새 blog 폴더로 이동시킵니다.

```
[ec2-user wordpress]$ mkdir /var/www/html/blog
[ec2-user wordpress]$ mv * /var/www/html/blog
```

Important

다음 프로시저로 즉시 이동하지 않는 경우는 보안상 문제가 발생할 수 있으므로 Apache 웹 서버(`httpd`)를 중단하십시오. WordPress 설치를 Apache 문서 루트로 이동한 후에는 WordPress 설치 스크립트가 보호되지 않는 상태이기 때문에 Apache 웹 서버가 실행 중일 때 블로그에 침입자가 액세스할 가능성이 있습니다. Apache 웹 서버를 중단시키려면 `sudo service httpd stop` 명령을 입력합니다. 다음 절차로 즉시 이동하는 경우는 Apache 웹 서버를 중단시킬 필요가 없습니다.

WordPress에서 퍼마링크(permalinks)를 사용하는 방법

WordPress가 올바로 작동하려면 Apache .htaccess 파일을 사용해야 하지만 Amazon Linux에서는 기본적으로 이 파일을 사용할 수 없습니다. 따라서 아래 방법에 따라 Apache 문서 루트에서 모든 재정의를 허용해야 합니다.

1. 즐겨 사용하는 텍스트 편집기(nano, vim 등)로 httpd.conf 파일을 엽니다. 원하는 텍스트 편집기가 없는 경우, 초보자는 nano를 사용하는 것이 더욱 편리합니다.

```
[ec2-user wordpress]$ sudo vim /etc/httpd/conf/httpd.conf
```

2. 다음과 같이 시작하는 영역을 찾습니다. <Directory "/var/www/html">

```
<Directory "/var/www/html">
#
# Possible values for the Options directive are "None", "All",
# or any combination of:
#   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
#
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
#
# The Options directive is both complicated and important. Please see
# http://httpd.apache.org/docs/2.4/mod/core.html#options
# for more information.
#
Options Indexes FollowSymLinks

#
# AllowOverride controls what directives may be placed in .htaccess files.
# It can be "All", "None", or any combination of the keywords:
#   Options FileInfo AuthConfig Limit
#
AllowOverride None

#
# Controls who can get stuff from this server.
#
Require all granted
</Directory>
```

3. 위 영역에서 AllowOverride None 라인을 AllowOverride All로 변경합니다.

Note

이 파일에는 AllowOverride 라인이 많기 때문에 <Directory "/var/www/html"> 영역의 라인을 변경할 때는 주의해야 합니다.

```
AllowOverride All
```

4. 파일을 저장하고 텍스트 편집기를 종료합니다.

Apache 웹 서버에 대한 파일 권한 수정 방법

WordPress의 제공 기능 중 일부(예: 관리 화면을 통한 미디어 업로드 등)는 Apache 문서 루트에 대한 쓰기 권한을 필요로 합니다. 웹 서버가 apache 사용자로 실행 중이기 때문에, 해당 사용자를 [LAMP web server tutorial\(LAMP 웹 서버 자습서\) \(p. 27\)](#)에서 생성된 www 그룹에 추가할 필요가 있습니다.

1. apache 사용자를 www 그룹에 추가.

```
[ec2-user wordpress]$ sudo usermod -a -G www apache
```

2. /var/www 및 그 콘텐츠의 파일 소유권을 apache 사용자로 변경합니다.

```
[ec2-user wordpress]$ sudo chown -R apache /var/www
```

3. /var/www 및 그 콘텐츠의 그룹 소유권을 www 그룹으로 변경합니다.

```
[ec2-user wordpress]$ sudo chgrp -R www /var/www
```

4. /var/www 및 그 하위 디렉터리의 디렉터리 권한을 변경해서 그룹 쓰기 권한을 추가하고 미래 하위 디렉터리에서 그룹 ID를 설정합니다.

```
[ec2-user wordpress]$ sudo chmod 2775 /var/www
[ec2-user wordpress]$ find /var/www -type d -exec sudo chmod 2775 {} \;
```

5. /var/www 및 그 하위 디렉터리의 파일 권한을 계속 변경해서 그룹 쓰기 권한을 추가합니다.

```
[ec2-user wordpress]$ find /var/www -type f -exec sudo chmod 0664 {} \;
```

6. Apache 웹 서버를 재시작해서 새 그룹 및 권한을 가져옵니다.

```
[ec2-user wordpress]$ sudo service httpd restart
Stopping httpd:                                     [    OK    ]
Starting httpd:                                     [    OK    ]
```

WordPress 설치 스크립트 실행 방법

1. [chkconfig] 명령을 사용해서 매번 시스템이 부팅할 때마다 httpd 및 mysqld 서비스가 시작되도록 합니다.

```
[ec2-user wordpress]$ sudo chkconfig httpd on
[ec2-user wordpress]$ sudo chkconfig mysqld on
```

2. MySQL 서버(mysqld)가 실행 중인지 확인합니다.

```
[ec2-user wordpress]$ sudo service mysqld status
mysqld (pid  4746) is running...
```

mysqld 서비스가 실행 중이지 않은 경우, 이를 시작합니다.

```
[ec2-user wordpress]$ sudo service mysqld start
Starting mysqld:                                     [    OK    ]
```

3. Apache 웹 서버(httpd)가 실행 중인지 확인합니다.

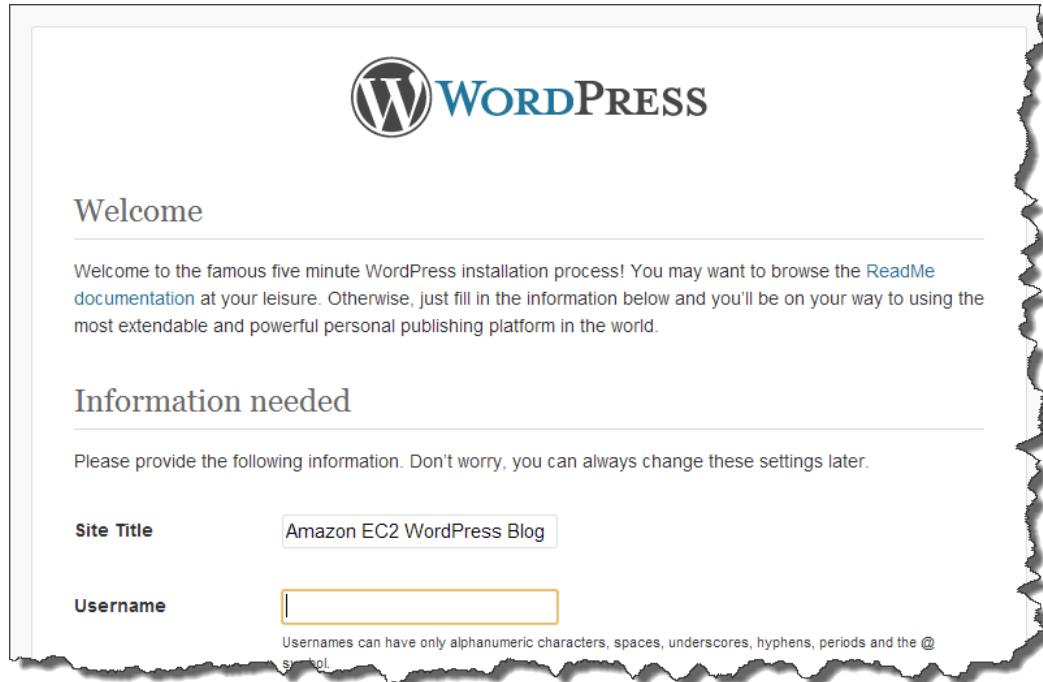
```
[ec2-user wordpress]$ sudo service httpd status
httpd (pid  502) is running...
```

httpd 서비스가 실행 중이지 않은 경우, 이를 시작합니다.

```
[ec2-user wordpress]$ sudo service httpd start
Starting httpd:                                     [    OK    ]
```

4. 웹 브라우저에서 WordPress 블로그의 URL을 입력합니다(인스턴스에 대한 퍼블릭 DLS 주소 또는 blog 폴더 다음의 주소). 이제 WordPress 설치 화면이 나타납니다.

<http://my.public.dns.amazonaws.com>



5. 남은 설치 정보를 WordPress 설치 마법사에 입력합니다.

필드	값
[사이트 제목]	WordPress 사이트의 이름을 입력합니다.
사용자명	WordPress 관리자의 이름을 입력합니다. 보안을 위해 사용자의 기본 사용자 이름(admin)보다 도용하기 더 어려운 고유 이름을 선택해야 합니다.
비밀번호	보안성이 강력한 암호를 입력하고 이를 한 번 더 입력해서 확인합니다. 기존 암호를 재사용하지 마십시오. 새로 설정한 암호는 안전한 장소에 보관하십시오.
[이메일 주소]	알림에 사용할 이메일 주소를 입력합니다.

6. [Install WordPress]를 클릭해서 설치를 완료합니다.

축하합니다. 이제 WordPress 블로그에 로그인해서 항목 기록을 시작할 수 있습니다.

다음 단계

초기 WordPress 블로그를 테스트한 후 구성을 업데이트하십시오.

사용자 지정 도메인 이름 사용

EC2 인스턴스의 EIP 주소와 연결되어 있는 도메인 이름이 있는 경우에는 EC2 퍼블릭 DNS 주소 대신에 해당 이름을 사용하여 블로그를 구성할 수 있습니다. 자세한 내용은 http://codex.wordpress.org/Changing_The_Site_URL을 참조하십시오.

블로그 구성

다른 테마와 플러그인을 사용하여 더욱 풍부한 맞춤형 경험을 독자에게 제공하도록 블로그를 구성할 수도 있습니다. 하지만 설치 프로세스가 역효과를 낳아 전체 블로그를 잃는 경우가 발생할 수도 있습니다. 따라서 테마나 플러그인을 설치하기 전에 인스턴스의 백업 Amazon 머신 이미지(AMI)를 생성하여 설치 중 오류가 발생하더라도 블로그를 복구할 수 있도록 대비하는 것이 좋습니다. 자세한 내용은 [고유 AMI 생성 \(p. 62\)](#) 섹션을 참조하십시오.

용량 증가

운영하는 WordPress 블로그가 유명해지고 그에 따라 보다 많은 컴퓨팅 파워 또는 스토리지가 필요하게 될 경우 다음 단계를 고려하십시오.

- 인스턴스에서 스토리지 공간을 확장합니다. 자세한 내용은 [Linux에서 EBS 볼륨의 크기, IOPS 또는 유형 수정 \(p. 590\)](#) 섹션을 참조하십시오.
- MySQL 데이터베이스를 [Amazon RDS](#)로 이동하여 이 서비스의 자동 확장 기능을 이용합니다.
- 더 큰 인스턴스 유형으로 마이그레이션합니다. 자세한 내용은 [인스턴스 크기 조정 \(p. 169\)](#) 섹션을 참조하십시오.
- 인스턴스를 더 추가합니다. 자세한 내용은 [자습서: Amazon EC2에서 애플리케이션의 가용성 향상 \(p. 54\)](#) 섹션을 참조하십시오.

WordPress에 대해 자세히 알아보기

WordPress에 대한 자세한 내용은 <http://codex.wordpress.org/>에서 WordPress Codex 도움 문서를 참조하십시오. 설치 문제 해결에 대한 자세한 내용은 [http://codex.wordpress.org/Installing_WordPress#Common_Installation_Problems](#) 섹션을 참조하십시오. WordPress 블로그 보안 강화에 대한 자세한 내용은 [http://codex.wordpress.org/Hardening_WordPress](#) 섹션을 참조하십시오. WordPress 블로그 보안 강화에 대한 자세한 내용은 [http://codex.wordpress.org/Hardening_WordPress](#) 섹션을 참조하십시오.

도움말! 내 퍼블릭 DNS 이름이 변경되어 블로그를 사용할 수 없습니다.

WordPress 설치 위치는 EC2 인스턴스의 퍼블릭 DNS 주소를 사용해 자동으로 구성됩니다. 이때 인스턴스를 중단했다가 다시 시작하면 퍼블릭 DNS 주소가 바뀌어(탄력적 IP 주소와 연결되어 있지 않은 경우) 블로그를 더 이상 사용할 수 없게 됩니다. 리소스를 참조해야 할 주소가 더 이상 존재하지 않거나 다른 EC2 인스턴스에 할당되었기 때문입니다. 이 문제를 비롯해 몇 가지 해결책에 대한 자세한 내용은 [http://codex.wordpress.org/Changing_The_Site_URL](#)을 참조하십시오.

이 문제가 WordPress 설치 위치에 발생하더라도 아래 절차에 따라 WordPress의 wp-cli 명령줄 인터페이스를 사용하면 블로그를 복구할 수 있습니다.

wp-cli를 이용해 WordPress 사이트 URL을 바꾸는 방법

- SSH를 통해 EC2 인스턴스에 연결합니다.
- 인스턴스의 이전 사이트 URL과 새로운 사이트 URL을 기록합니다. 이전 사이트 URL은 WordPress 설치 시 EC2 인스턴스의 퍼블릭 DNS 이름일 가능성이 높습니다. 그리고 새로운 사이트 URL은 EC2 인스턴스의 현재 퍼블릭 DNS 이름입니다. 이전 사이트 URL을 잘 모르더라도 아래와 같이 curl 명령을 사용해 찾을 수 있습니다.

```
[ec2-user ~]$ curl localhost | grep wp-content
```

명령을 실행하여 출력되는 화면에서 이전 퍼블릭 DNS 이름의 참조를 확인해야 합니다. 출력 화면은 다음과 같습니다(빨간색의 이전 사이트 URL).

```
<script type='text/javascript' src='http://ec2-52-8-139-223.us-west-1.compute.amazonaws.com/wp-content/themes/twentyfifteen/js/functions.js?ver=20150330'></script>
```

3. 아래와 같은 명령으로 wp-cli를 다운로드합니다.

```
[ec2-user ~]$ curl -O https://raw.githubusercontent.com/wp-cli/builds/gh-pages/phar/wp-cli.phar
```

4. 아래와 같은 명령으로 이전 사이트 URL을 찾아 WordPress 설치 위치로 바꿉니다. EC2 인스턴스의 이전 사이트 URL과 새로운 사이트 URL, 그리고 WordPress 설치 경로(대체로 /var/www/html 또는 /var/www/html/blog)를 치환합니다.

```
[ec2-user ~]$ php wp-cli.phar search-replace 'old_site_url' 'new_site_url' --path=/path/to/wordpress/installation --skip-columns=guid
```

5. 웹 브라우저에서 WordPress 블로그의 새로운 사이트 URL을 입력하여 사이트에 올바로 접속되는지 다시 확인합니다. 그렇지 않으면 http://codex.wordpress.org/Changing_The_Site_URL과 http://codex.wordpress.org/Installing_WordPress#Common_Installation_Problems에서 자세한 내용을 참조하십시오.

자습서: SSL/TLS를 사용하여 Amazon Linux에서 Apache 웹 서버 구성

Secure Sockets Layer/Transport Layer Security(SSL/TLS)는 웹 서버와 웹 클라이언트 간 암호화된 채널을 만들어 전송 중인 데이터가 도청되지 않도록 보호합니다. 이 자습서에서는 Apache 웹 서버를 실행하는 Amazon Linux의 단일 인스턴스에 SSL/TLS 지원을 수동으로 추가하는 방법을 설명합니다. [AWS Certificate Manager](#)는 여기에서 설명하지 않지만 여러 도메인을 관리하거나 상용 등급 서비스를 제공해야 할 경우 특히 적합한 옵션입니다.

Note

일반적으로 웹 암호화를 단순히 SSL이라고 부릅니다. 웹 브라우저에서 여전히 SSL을 지원하지만, 후속 프로토콜인 TLS가 공격에 덜 취약한 것으로 여겨집니다. Amazon Linux는 기본적으로 SSL 버전 2를 사용하지 않지만, 이 자습서에서는 아래에 설명된 대로 SSL 버전 3도 사용하지 않을 것을 권장합니다. 업데이트된 암호화 표준에 대한 자세한 내용은 [RFC7568](#)을 참조하십시오.

Important

이 절차는 Amazon Linux에서 사용하기 위한 것입니다. LAMP 웹 서버를 다른 배포 인스턴스에서 설치하려는 경우는 본 자습서를 이용할 수 없습니다. Ubuntu의 LAMP 웹 서버에 대한 자세한 내용은 Ubuntu 커뮤니티 문서 [ApacheMySQLPHP](#) 섹션을 참조하십시오. Red Hat Enterprise Linux에 대한 자세한 내용은 고객 포털 주제 [웹 서버](#)를 참조하십시오.

항목

- [사전 조건 \(p. 47\)](#)
- [1단계: 서버에서 SSL/TLS 활성화 \(p. 47\)](#)
- [2단계: CA가 서명한 인증서 가져오기 \(p. 48\)](#)
- [3단계: 보안 구성 테스트 및 강화 \(p. 52\)](#)
- [문제 해결 \(p. 54\)](#)

사전 조건

이 자습서를 시작하기 전에 다음 단계를 완료합니다.

- Amazon Linux 인스턴스를 시작합니다. 자세한 내용은 [1단계: 인스턴스 시작 \(p. 22\)](#)를 참조하십시오.
- SSH(포트 22), HTTP(포트 80) 및 HTTPS(포트 443) 연결을 허용하도록 보안 그룹을 구성합니다. 자세한 내용은 [Amazon EC2로 설정 \(p. 16\)](#)를 참조하십시오.
- Apache 웹 서버를 설치합니다. 단계별 지침은 [자습서: Amazon Linux에 LAMP 웹 서버 설치 \(p. 27\)](#)를 참조하십시오. http24 패키지와 그 종속 프로그램만 필요합니다. PHP 및 MySQL과 관련된 지침은 무시해도 됩니다.
- SSL/TLS 퍼블릭 키 인프라(PKI)는 DNS(도메인 이름 시스템)를 사용하여 웹 사이트를 식별하고 인증합니다. EC2 인스턴스를 사용하여 퍼블릭 웹 사이트를 호스팅하려는 경우, 웹 서버의 도메인 이름을 등록하거나 Amazon EC2 호스트로 기존 도메인 이름을 전송해야 합니다. 수많은 타사 도메인 등록 및 DNS 호스팅 서비스를 이에 사용할 수 있습니다. 또는 [Amazon Route 53](#)을 사용할 수도 있습니다.

1단계: 서버에서 SSL/TLS 활성화

이 절차에서는 자체 서명된 디지털 인증서를 사용하여 Amazon Linux에 SSL/TLS를 설치하는 과정을 보여줍니다.

서버에서 SSL/TLS를 활성화하려면

- [인스턴스에 연결 \(p. 23\)](#)한 다음 Apache가 실행되는지 확인합니다.

```
[ec2-user ~]$ sudo service httpd status
```

필요한 경우 Apache를 시작합니다.

```
[ec2-user ~]$ sudo service httpd start
```

- 모든 소프트웨어 패키지가 최신 상태로 업데이트되어 있는지 확인하기 위해, 인스턴스에서 쿠 소프트웨어 업데이트를 실행합니다. 이 업데이트 과정은 몇 분 정도 시간이 소요될 수 있지만, 최신 보안 업데이트와 버그 수정을 위해 수행할 필요가 있습니다.

Note

-y 옵션을 사용하면 확인 여부를 묻지 않고 업데이트를 설치합니다. 설치 전에 업데이트 정보를 확인하려면 이 옵션을 생략합니다.

```
[ec2-user ~]$ sudo yum update -y
```

- 이제 인스턴스가 최신 상태이므로 다음과 같은 Apache module mod_ssl을 설치하여 SSL/TLS 지원을 추가합니다.

```
[ec2-user ~]$ sudo yum install -y mod24_ssl
```

이 자습서에서 나중에 다음과 같은 설치된 세 가지 중요한 파일을 작업합니다.

- /etc/httpd/conf.d/ssl.conf
- mod_ssl의 구성 파일입니다. 이 파일에는 Apache에 암호화 키 및 인증서의 위치, 허용하는 SSL/TLS 프로토콜, 사용하는 암호화 알고리즘을 알려주는 "명령"이 포함되어 있습니다.
- /etc/pki/tls/private/localhost.key

Amazon EC2 호스트의 2048비트 RSA 프라이빗 키로, 자동으로 생성됩니다. 설치하는 동안 OpenSSL은 이 키를 사용하여 자체 서명된 호스트 인증서를 생성하며, 나중에 이를 사용하여 인증 기관(CA)에 제출할 인증서 서명 요청(CSR)을 생성할 수 있습니다.

- /etc/pki/tls/certs/localhost.crt

서버 호스트의 자체 서명된 X.509 인증서로, 자동으로 생성됩니다. 이 인증서는 Apache가 SSL/TLS를 사용하도록 올바르게 설치되었는지 테스트하는 데 유용합니다.

.key 및 .crt 파일은 모두 PEM 형식입니다. 이 형식은 아래의 축약된 인증서 예제와 같이 "BEGIN" 및 "END" 라인으로 프레임 처리된 Base64 인코딩 ASCII 문자로 구성됩니다.

```
-----BEGIN CERTIFICATE-----  
  
MIIEazCCA1OgAwIBAgICWxQwDQYJKoZIhvcNAQELBQAwbExCzAJBgNVBAYTAi0t  
MRIwEAYDVQQIDA1Tb21lU3RhGUxETAPBgNVBACMCFNvbWVDaXR5MRkwFwYDVQQK  
DBBTb21lT3JnYW5pemF0aW9uMR8wHQYDVQQLDBZTb21lT3JnYW5pemF0aW9uYWxv  
bm10MRkwFwYDVQODDBpcC0xNzItMzEtMjAtMjM2MSQwIgYJKoZIhvcNAQkBFhVv  
...  
z5rRUE/XzxRLBZ0oWZpNWTXJkQ3uFYH6s/  
sBwtHpKKZMzOvDedREjNKAvk4ws6F0  
WanXWehT6FiSzvB4sTEXXJN2jdw8g  
+sHGnZ8zCOsclknYhHrCVD2vnBlZJKSzvak  
3ZazhBxtQSukFMOnWPP2a0DMMFGYUHod0BQE8sBJxg==  
-----END CERTIFICATE-----
```

파일 이름 및 확장명은 편의상 사용되며 기능에 영향을 미치지 않습니다. ssl.conf 파일에서 관련 명령에 동일한 이름을 사용하는 한, 인증서 이름을 cert.crt 또는 cert.pem 또는 certificate.pem으로 지정할 수 있습니다.

Note

기본 SSL 파일을 고유의 사용자 지정 파일로 대체하는 경우 파일이 PEM 형식인지 확인하십시오.

4. Apache를 다시 시작합니다.

```
[ec2-user ~]$ sudo service httpd restart
```

5. Apache 웹 서버가 현재 포트 443에 대해 HTTPS(보안 HTTP)를 지원해야 합니다. 접두사가 <https://>인 브라우저 URL 표시줄에 IP 주소 또는 EC2 인스턴스의 정규화된 도메인 이름을 입력하여 이를 테스트합니다. 신뢰할 수 없는 자체 서명된 인증서를 사용하여 사이트에 연결하기 때문에 브라우저에 경고가 연속으로 표시될 수 있습니다.

이러한 경고를 무시하고 계속 진행합니다. Apache 기본 시작 페이지가 열리면 서버에 SSL/TLS가 구성되었다는 것입니다. 브라우저의 URL 표시줄의 잠금 아이콘에서 알 수 있듯이 브라우저와 서버를 통과하는 모든 데이터가 이제 안전하게 암호화됩니다.

사이트 방문자에게 경고 화면이 표시되는 것을 방지하려면 암호화뿐만 아니라 해당 사이트의 소유자라는 것을 공개적으로 인증하는 인증서를 가져와야 합니다.

2단계: CA가 서명한 인증서 가져오기

이 섹션에서는 프라이빗 키에서 인증서 서명 요청(CSR)을 생성하고, 인증 기관(CA)에 CSR을 제출하고, 서명된 인증서를 가져오고, Apache를 구성하여 이를 사용하는 절차를 설명합니다.

자체 서명된 SSL/TLS X.509 인증서는 CA가 서명한 인증서와 암호적으로 동일합니다. 그 차이는 수학적인 것이 아니라 사회적입니다. CA는 신청자에게 인증서를 발급하기 전에 도메인의 소유권을 최소한으로 검사합니다. 각 웹 브라우저에는 이를 하도록 브라우저 공급업체에서 신뢰한 CA 목록이 포함되어 있습니다. X.509 인증서는 프라이빗 서버 키에 해당하는 퍼블릭 키와 퍼블릭 키에 암호화 방식으로 연결된 CA의 서명으로 주로 구성되어 있습니다. 브라우저가 HTTPS를 통해 웹 서버에 연결되면 서버는 브라우저에서 신뢰할 수 있는 CA 목록을 확인하도록 인증서를 제공합니다. 서명자가 목록에 있거나 신뢰할 수 있는 다른 서명자 체인을 통해 서명자에 액세스할 수 있는 경우, 브라우저는 서버와 암호화된 빠른 데이터 채널을 협상하고 페이지를 로드합니다.

요청 확인 절차로 인해 인증서에는 일반적으로 비용이 발생하므로 여러 인증 기관을 알아봐야 합니다. 잘 알려진 CA 목록은 dmoz.org에서 확인할 수 있습니다. StartCom과 같은 일부 CA는 기본 수준("클래스 1") 인증서를 무료로 제공합니다.

인증서의 기본을 이루는 것은 키입니다. 2013년 현재 정부 및 산업 그룹에서는 RSA 키에 대해 최소 2048비트의 키(모듈러스) 크기를 사용할 것을 권장합니다. Amazon Linux의 OpenSSL에서 생성된 기본 모듈러스 크기는 2048비트이므로, 기존의 자동 생성된 키를 CA가 서명한 인증서에 사용할 수 있습니다. 예를 들면, 모듈러스가 더 크거나 다른 암호화 알고리즘을 사용하는 사용자 지정 키를 원하는 경우 아래 절차를 참조합니다.

CA가 서명한 인증서를 가져오려면

1. [인스턴스에 연결](#)(p. 23)한 다음 /etc/pki/tls/private/으로 이동합니다. 이는 SSL/TLS에 대한 서버의 프라이빗 키가 저장된 디렉터리입니다. 기존 호스트 키를 사용하여 CSR을 생성하려면 3단계로 건너뜁니다.
2. (선택 사항) 새 프라이빗 키를 생성합니다. 다음은 몇 가지 키 구성 샘플입니다. 어떤 결과 키도 웹 서버에서 사용할 수 있지만 어떻게(또한 얼마나) 보안을 구현할지는 각각 다릅니다.
 1. 다음은 이를 위한 첫 단계로써 인스턴스의 기본 호스트 키와 유사한 RSA 키를 생성하는 명령입니다.

```
[ec2-user ~]$ sudo openssl genrsa -out custom.key 2048
```

결과 파일인 `custom.key`는 2048비트 RSA 프라이빗 키입니다.

2. 모듈러스가 더 큰 보다 강력한 RSA 키를 생성하려면 다음 명령을 사용합니다.

```
[ec2-user ~]$ sudo openssl genrsa -out custom.key 4096
```

결과 파일인 `custom.key`는 4096비트 RSA 프라이빗 키입니다.

3. 암호로 보호되는 4096비트 암호화 RSA 키를 생성하려면 다음 명령을 사용합니다.

```
[ec2-user ~]$ sudo openssl genrsa -aes128 -passout pass:abcde12345 -out custom.key 4096
```

그러면 AES-128 암호화로 암호화된 4096비트 RSA 프라이빗 키가 생성됩니다.

Important

암호화를 통해 보안을 강화할 수 있지만, 암호화된 키에는 암호가 필요하기 때문에 이를 사용하는 서비스는 자동으로 시작할 수 없습니다. 이 키를 사용할 때마다 SSH 연결을 통해 암호 "abcde12345"를 입력해야 합니다.

4. RSA 암호화는 상대적으로 느린데, 그 보안 체계가 큰 두 개의 소수를 소인수 분해하는 것의 난해성에 기반을 두기 때문입니다. 그러나 RSA 암호화 이외의 암호화를 사용하는 SSL/TLS의 키를 생성할 수 있습니다. 타원 곡선 수학을 기반으로 하는 키는 동등한 보안 수준을 제공할 때 보다 작고 빠릅니다. 다음은 그 예입니다.

```
[ec2-user ~]$ sudo openssl ecparam -name prime256v1 -out custom.key -genkey
```

이 예에서 출력은 OpenSSL에서 지원하는 "명명된 곡선"인 prime256v1을 사용하는 256비트 타원 곡선 프라이빗 키입니다. NIST에 따르면 이 키의 암호화 강도는 2048비트 RSA 키보다 약간 더 높습니다.

Note

모든 CA에서 타원 곡선 기반 키에 대해 RSA 키와 동등한 수준의 지원을 제공하지는 않습니다.

새 프라이빗 키의 권한은 매우 제한적(소유자 루트, 그룹 루트, 소유자 전용 읽기/쓰기)이어야 합니다. 명령은 다음과 같습니다.

```
[ec2-user ~]$ sudo chown root.root custom.key
[ec2-user ~]$ sudo chmod 600 custom.key
[ec2-user ~]$ ls -al custom.key
```

위 명령의 결과는 다음과 같아야 합니다.

```
-rw----- root root custom.key
```

만족스러운 키를 생성 및 구성한 후 CSR을 생성할 수 있습니다.

- 원하는 키를 사용하여 CSR을 생성합니다. 아래 예에서는 `private.key`를 사용합니다.

```
[ec2-user ~]$ sudo openssl req -new -key private.key -out csr.pem
```

OpenSSL은 대화 상자를 열고 아래 표의 정보를 입력하라는 메시지를 표시합니다. 도메인에서 확인된 기본 인증서의 경우 [Common Name]을 제외한 모든 필드는 선택 사항입니다.

이름	설명	예
국가 이름	해당 국가의 두 자리 ISO 약자.	US(=미국)
주 또는 지방 이름	해당 조직이 위치한 주 또는 지방의 이름. 이 이름은 약어로 사용할 수 없음.	워싱턴
시 이름	조직의 위치(예: 도시).	Seattle
조직 이름	해당 조직의 정식 이름. 조직 이름의 약칭을 사용하지 마십시오.	Example Corp
조직 단위 이름	조직에 대한 추가 정보(있는 경우).	부서 예
일반 이름	이 값은 사용자가 브라우저에 입력해야 하는 웹 주소와 정확히 일치해야 합니다. 일반적으로 이는 <code>www.example.com</code> 의 형식으로, 호스트 이름 또는 별칭이 앞에 붙는 도메인 이름을 뜻합니다. 자체 서명된 인증서로 DNS 확인 없이 테스트하는 경우, 일반 이름은 호스트 이름만으로 구성될 수 있습니다. CA는 <code>*.example.com</code> 과 같이 와일드 카드 이름을 허용하는 비싼 인증서도 제공합니다.	www.example.com
이메일 주소	서버 관리자의 이메일 주소.	someone@example.com

마지막으로 OpenSSL은 챌린지 암호(선택 사항)를 입력하라는 메시지를 표시합니다. 이 암호는 해당 CSR 및 사용자와 해당 CA 간의 트랜잭션에만 적용되므로, 암호 및 기타 선택적 필드(선택적 회사 이름)에 대한 해당 CA의 권장 사항을 따릅니다. CSR 챌린지 암호는 서버 작업에 영향을 미치지 않습니다.

결과 파일인 `csr.pem`에는 퍼블릭 키, 퍼블릭 키의 디지털 서명 및 입력한 메타데이터가 포함되어 있습니다.

4. CA에 CSR을 제출합니다. 이는 보통 텍스트 편집기에서 CSR 파일을 열고 웹 양식에 내용을 복사하는 것으로 구성됩니다. 이때 인증서에 추가할 하나 이상의 주체 대체 이름(SAN)을 입력하라는 메시지가 나타날 수 있습니다. `www.example.com`이 일반 이름일 경우, `example.com`은 좋은 SAN이며, 그 반대의 경우도 마찬가지입니다. 위 이름 중 하나를 입력하면 오류 없이 연결됩니다. CA 웹 양식에서 이를 허용하는 경우, SAN 목록에 일반 이름을 포함시킵니다. (일부 CA는 이를 자동으로 포함시킵니다.)

요청이 승인되면 CA에서 서명한 새 호스트 인증서를 받게 됩니다. CA의 신뢰 체인을 완료하는 데 필요한 추가 인증서가 포함된 중간 인증서 파일을 다운로드하라는 안내를 받을 수도 있습니다.

5. `/etc/pki/tls/certs` 디렉터리에서 자체 서명된 이전 호스트 인증서인 `localhost.crt`를 제거하고 해당 디렉터리에 (중간 인증서와 함께) CA가 서명한 새 인증서를 추가합니다.

Note

여러 가지 방법으로 새 인증서를 EC2 인스턴스에 업로드할 수 있지만, 가장 간편하고 유익한 방법은 텍스트 편집기(vi, nano, 메모장 등)를 로컬 컴퓨터와 인스턴스에 모두 열고 두 편집기 간에 파일 콘텐츠를 복사하여 붙이는 것입니다. 이렇게 하면 권한 또는 경로 문제가 있는 경우 즉시 확인할 수 있습니다. 하지만 콘텐츠를 복사하는 동안 라인을 추가하거나 어떤 식으로든 콘텐츠를 변경하지 않도록 주의하십시오.

`/etc/pki/tls/certs` 디렉터리 내에서 파일 소유권, 그룹 및 권한 설정이 매우 제한적인 Amazon Linux 기본 값(소유자 루트, 그룹 루트, 소유자 전용 읽기/쓰기)과 일치하는지 확인합니다. 명령은 다음과 같습니다.

```
[ec2-user certs]$ sudo chown root.root custom.crt
[ec2-user certs]$ sudo chmod 600 custom.crt
[ec2-user certs]$ ls -al custom.crt
```

위 명령의 결과는 다음과 같아야 합니다.

```
-rw----- root root custom.crt
```

중간 인증서 파일에 대한 권한은 덜 엄격합니다(소유자 루트, 그룹 루트, 소유자 쓰기 가능, 모든 사용자 읽기 가능). 이 명령은 다음과 같습니다.

```
[ec2-user certs]$ sudo chown root.root intermediate.crt
[ec2-user certs]$ sudo chmod 644 intermediate.crt
[ec2-user certs]$ ls -al intermediate.crt
```

위 명령의 결과는 다음과 같아야 합니다.

```
-rw-r--r-- root root intermediate.crt
```

6. CA가 인증한 새 인증서의 파일 이름(이 예에서는 `custom.crt`)은 이전 인증서의 것과 다를 수 있습니다. `/etc/httpd/conf.d/ssl.conf`를 편집하고 Apache의 `SSLCertificateFile` 명령을 사용하여 올바른 경로 및 파일 이름을 입력합니다.

```
SSLCertificateFile /etc/pki/tls/certs/custom.crt
```

중간 인증서 파일을 받은 경우(이 예에서는 `intermediate.crt`), Apache의 `SSLCACertificateFile` 명령을 사용하여 경로 및 파일 이름을 입력합니다.

```
SSLCACertificateFile /etc/pki/tls/certs/intermediate.crt
```

7. `/etc/httpd/conf.d/ssl.conf`를 저장하고 Apache를 다시 시작합니다.

```
[ec2-user ~]$ sudo service httpd restart
```

3단계: 보안 구성 테스트 및 강화

SSL/TLS가 작동되고 일반에 공개된 후 이의 실제 보안 수준을 테스트해야 합니다. 보안 설정을 무료로 완벽하게 분석해 주는 [Qualys SSL Labs](#)와 같은 온라인 서비스를 사용하면 이를 손쉽게 수행할 수 있습니다. 그 결과에 따라 수용할 프로토콜, 원하는 암호 및 제외할 암호를 관리하여 기본 보안 구성을 강화할 수 있습니다. 자세한 내용은 [how Qualys formulates its scores](#) 섹션을 참조하십시오.

Important

실제 테스트는 서버 보안에 매우 중요합니다. 구성상의 작은 오류가 심각한 보안 침해 및 데이터 손실로 이어질 수 있습니다. 권장되는 보안 사례는 연구 및 새롭게 생겨나는 위협에 대처하기 위해 끊임없이 변화하므로 보안 감사를 주기적으로 실시하는 것이 서버 관리에 필수적입니다.

[Qualys SSL Labs](#) 사이트에 `www.example.com` 형식으로 서버의 정규화된 도메인 이름을 입력합니다. 약 2분 후 사이트 등급(A - F) 및 확인된 상세 분석 결과를 받게 됩니다. 아래 표에 Amazon Linux의 기본 Apache 구성과 설정이 동일한 도메인에 대한 보고서가 요약되어 있습니다.

종합 등급	C
인증서	100%
프로토콜 지원	90%
키 교환	90%
암호화 수준	90%

이 보고서에 따르면 인증서, 프로토콜 지원, 키 교환 및 암호화 수준이 수용할 만한 등급으로, 구성이 대체로 안전하다는 것을 보여 줍니다. 그러나 이 보고서는 또한 종합 등급을 낮춘 세 가지 취약성을 플래그로 지정하고 있으며, 이러한 취약성은 다음과 같이 해결할 수 있습니다.

- 푸들 취약성: 2014년에 발견된 [푸들 공격](#)은 공격자가 웹 사이트를 가장할 수 있도록 하는 SSL 버전 3의 취약점을 활용합니다. 해결책은 간단합니다. 바로 서버에 대한 SSL 버전 3 지원을 비활성화하는 것입니다. `/etc/httpd/conf.d/ssl.conf` 구성 파일에서 줄의 시작 부분에 "#"을 입력하여 다음을 주석 처리합니다.

```
SSLProtocol all -SSLv2
```

그런 다음, 다음 명령을 추가합니다.

```
SSLProtocol -SSLv2 -SSLv3 +TLSv1 +TLSv1.1 +TLSv1.2
```

이 명령은 SSL 버전 2를 명시적으로 비활성화하는 것 외에도 (보안 감사에서 플래그 지정한) SSL 버전 3 을 비활성화하고 기존의 모든 TLS 버전을 명시적으로 허용합니다. 이제 이 서버는 TLS 이외의 프로토콜을 사용하는 클라이언트와의 암호화된 연결을 허용하지 않습니다. 명령의 상세 내용은 서버의 구성 내용을 사람에게 더욱 명확히 전달합니다.

- RC4 암호 지원: 암호는 암호화 알고리즘의 수학적 핵심입니다. SSL/TLS 데이터 스트림을 암호화하는 데 사용하는 빠른 암호인 RC4에는 몇 가지 **심각한 취약점**이 있는 것으로 알려져 있습니다. 해결책은 ssl.conf에서 RC4 지원을 비활성화하는 것으로, 이는 다음 예의 해결책 중 일부이기도 합니다.
- 순방향 비밀성 지원 누락: **순방향 비밀성**은 프라이빗 키에서 파생된 임시(사용 후 삭제) 세션 키를 사용하여 암호화하는 프로토콜의 기능입니다. 이는 실제 공격자가 웹 서버의 장기 프라이빗 키를 보유하고 있더라도 HTTPS 데이터의 암호를 해독할 수 없다는 것을 뜻합니다. Qualys의 "참조 브라우저" 목록을 구성하는 웹 브라우저는 모두 순방향 비밀성을 지원합니다.

RC4와 순방향 비밀성 문제의 해결책은 Apache의 허용 및 금지 암호 목록을 사용자 지정하고, 취약한 암호에 대해 강력한 암호 기본 설정을 적용하는 것입니다. 여기에는 두 가지 구성 변경이 필요합니다.

/etc/httpd/conf.d/ssl.conf 구성 파일에서 **ssLCipherSuite** 구성을 위한 주석 처리된 예가 포함된 섹션을 찾고, 현재 목록을 주석 처리(유지)한 다음, 다음 명령을 추가합니다.

Note

여기에서는 가독성을 위해 여러 줄로 표시했지만, 전체 명령은 암호 이름 사이에 공백이 없는 한 줄이어야 합니다.

```
SSLCipherSuite ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES256-SHA384:  
ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-SHA256:  
AES128-GCM-SHA256:AES256-GCM-SHA384:AES128-SHA256:AES256-SHA256:AES:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!PSK:aECDH:  
!EDH-DSS-DES-CBC3-SHA:!EDH-RSA-DES-CBC3-SHA:!KRB5-DES-CBC3-SHA
```

이러한 암호는 OpenSSL에서 지원되는 훨씬 더 긴 암호 목록의 하위 집합으로, 다음 기준에 따라 선택 및 정렬되었습니다.

1. 순방향 비밀성 지원
2. 암호화 수준
3. 속도
4. 암호 패밀리 앞의 특정 암호
5. 거부된 암호 앞의 허용된 암호

순위가 높은 암호의 경우 이름에 ECDHE(Elliptic Curve Diffie-Hellman Ephemeral)가 있습니다. ephemeral(사용 후 삭제)은 순방향 비밀성을 나타냅니다. 또한 RC4는 현재 끝 부분에 있는 금지된 암호 중 하나입니다.

내용이 표시되지 않는 기본값 또는 terse 명령 대신 명시적 암호 목록을 사용하는 것이 좋습니다.

Important

여기에 나와 있는 암호 목록은 가능한 여러 목록 중 하나에 불과합니다. 예를 들어, 순방향 비밀성 대신 속도를 기준으로 목록을 최적화 할 수도 있습니다.

이전 클라이언트를 지원해야 할 경우, DES-CBC3-SHA 암호 그룹을 허용할 수 있습니다.

마지막으로 각 OpenSSL 업데이트 시 새 암호가 도입되고 이전 암호의 사용이 중단됩니다. EC2 Amazon Linux 인스턴스를 최신으로 유지하고, OpenSSL의 보안 알림을 잘 확인하며, 기술 정보 신문의 새 보안 취약점에 대한 보고를 잘 살핍니다. 자세한 내용은 Elastic Load Balancing 사용 설명서의 [Predefined SSL Security Policies for Elastic Load Balancing](#) 섹션을 참조하십시오.

"#"을 제거하여 다음 줄의 주석 처리를 해제합니다.

```
#SSLHonorCipherOrder on
```

이 명령은 (이 예에서는) 순방향 비밀성을 지원하는 암호를 포함하여 서버에서 순위가 높은 암호를 선호하도록 합니다. 이 명령이 설정되면 서버는 먼저 강력한 보안 연결 설정을 시도해 본 후 보안이 더 약한 허용된 암호로 대체합니다.

편집한 구성 파일을 저장한 후 Apache를 다시 시작합니다.

Qualys SSL Labs에서 도메인을 다시 테스트하려면 취약성이 해결되고 요약이 다음과 같아야 합니다.

종합 등급	A
인증서	100%
프로토콜 지원	95%
키 교환	90%
암호화 수준	90%

문제 해결

- 암호를 입력하지 않으면 Apache 웹 서버가 시작하지 않습니다.

암호화되고 암호로 보호되는 프라이빗 서버 키를 설치한 경우 이는 예상된 동작입니다.

키에서 암호화 및 암호를 제거할 수 있습니다. 기본 디렉터리에 custom.key라는 암호화된 프라이빗 RSA 키가 있고 이 키의 암호가 abcde12345라고 가정하면, EC2 인스턴스에서 다음 명령을 실행하여 이 키의 암호화되지 않은 버전을 생성합니다.

```
[ec2-user ~]$ cd /etc/pki/tls/private/
[ec2-user private]$ sudo cp custom.key custom.key.bak
[ec2-user private]$ sudo openssl rsa -in custom.key -passin pass:abcde12345 -out
    custom.key.nocrypt
[ec2-user private]$ sudo mv custom.key.nocrypt custom.key
[ec2-user private]$ sudo chown root.root custom.key
[ec2-user private]$ sudo chmod 600 custom.key
[ec2-user private]$ sudo service httpd restart
```

이제 Apache가 암호를 묻지 않고 시작할 것입니다.

자습서: Amazon EC2에서 애플리케이션의 가용성 향상

단일 EC2 인스턴스에서 애플리케이션 또는 웹 사이트 실행을 시작한 후 시간이 지나면서 수요를 충족하기 위해 두 개 이상의 인스턴스가 필요한 지점까지 트래픽이 증가하는 경우를 가정해 봅니다. AMI에서 여러 EC2 인스턴스를 시작한 다음 Elastic Load Balancing을 사용하여 애플리케이션에 대한 수신 트래픽을 EC2 인스턴스 간에 분산할 수 있습니다. 이렇게 하면 애플리케이션의 가용성이 향상됩니다. 인스턴스를 여러 가용 영역에 배치하면 애플리케이션의 내결함성도 향상됩니다. 가용 영역 하나가 중단되면 트래픽이 다른 가용 영역으로 라우팅됩니다.

Auto Scaling을 사용하여 애플리케이션에 대한 실행 인스턴스 수를 항상 최소한으로 유지할 수 있습니다. Auto Scaling은 인스턴스나 애플리케이션이 비정상일 때를 감지하고 자동으로 교체하여 애플리케이션의 가용성을 유지합니다. 또한 Auto Scaling을 사용하면 지정한 기준을 사용하여 필요에 따라 자동으로 Amazon EC2 용량을 확장하거나 축소할 수 있습니다.

이 자습서에서는 Auto Scaling과 Elastic Load Balancing을 사용하여 로드 밸런서 뒤에 지정된 수의 정상 EC2 인스턴스를 유지합니다. 트래픽이 로드 밸런서로 이동한 다음 인스턴스로 라우팅되므로 이러한 인스턴스에는 퍼블릭 IP 주소가 필요 없습니다. 자세한 내용은 [Auto Scaling](#) 및 [Elastic Load Balancing](#) 섹션을 참조하십시오.

목차

- [사전 조건 \(p. 55\)](#)
- [애플리케이션 확장 및 로드 밸런싱 \(p. 55\)](#)
- [로드 밸런서 테스트 \(p. 57\)](#)

사전 조건

이 자습서에서는 다음을 이미 완료했다고 가정합니다.

1. 기본 Virtual Private Cloud(VPC)가 없는 경우 두 개 이상의 가용 영역에서 하나의 퍼블릭 서브넷을 사용하여 VPC를 만듭니다. 자세한 내용은 [Virtual Private Cloud\(VPC\) 생성 \(p. 19\)](#) 섹션을 참조하십시오.
2. VPC에서 인스턴스를 시작합니다.
3. 인스턴스에 연결하여 인스턴스를 사용자 지정합니다. 예를 들어, 소프트웨어와 애플리케이션을 설치하고 데이터를 복사하고 추가 EBS 볼륨을 연결할 수 있습니다. 인스턴스에서 웹 서버를 설정하는 방법에 대한 자세한 내용은 [자습서: Amazon LinuxLAMP 웹 서버 설치 \(p. 27\)](#)를 참조하십시오.
4. 인스턴스에서 애플리케이션을 테스트하여 인스턴스가 올바르게 구성되었는지 확인합니다.
5. 인스턴스에서 사용자 지정 Amazon 머신 이미지(AMI)를 만듭니다. 자세한 내용은 [Amazon EBS 지원 Linux AMI 생성 \(p. 81\)](#) 또는 [인스턴스 스토어 기반 Linux AMI 생성 \(p. 84\)](#)을 참조하십시오.
6. (선택 사항) 더 이상 필요하지 않은 경우 인스턴스를 종료합니다.
7. 필요한 AWS에 대한 액세스를 애플리케이션에 부여하는 IAM 역할을 만듭니다. 자세한 내용은 [IAM 콘솔을 사용하여 IAM 역할을 생성하려면 다음을 수행합니다. \(p. 459\)](#) 섹션을 참조하십시오.

애플리케이션 확장 및 로드 밸런싱

다음 절차를 사용하여 로드 밸런서를 만들고, 인스턴스에 대한 시작 구성을 만든 다음, 두 개 이상의 인스턴스가 포함된 Auto Scaling 그룹을 만들고, 로드 밸런서를 Auto Scaling 그룹과 연결합니다.

애플리케이션을 확장하고 로드 밸런싱하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창의 [LOAD BALANCING] 아래에서 [Load Balancers]를 선택합니다.
3. [Create Load Balancer]를 선택합니다.
4. [Application Load Balancer]를 선택하고 [Continue]를 선택합니다.
5. [Configure Load Balancer] 페이지에서 다음 작업을 수행하십시오.
 - a. [Name]에 로드 밸런서 이름을 입력합니다. 예를 들면 **my-1b**를 입력하십시오.
 - b. [Scheme]의 [internet-facing]은 기본 값으로 유지합니다.
 - c. 포트 80에서 HTTP 트래픽을 수락하는 리스너를 뜻하는 [Listeners]는 기본 값으로 유지합니다.
 - d. [Availability Zones]에서 인스턴스에 사용한 VPC를 선택합니다. 가용 영역을 선택한 후 해당 가용 영역의 퍼블릭 서브넷을 선택합니다. 두 번째 가용 영역에 대해 이 절차를 반복합니다.
 - e. [Next: Configure Security Settings]를 선택합니다.
6. 이 자습서의 경우 안전한 리스너를 사용하지 않고 있습니다. [Next: Configure Security Groups]를 선택합니다.
7. [Configure Security Groups] 페이지에서 다음 작업을 수행하십시오.

- a. Create a new security group을 선택합니다.
 - b. 보안 그룹의 이름과 설명을 입력하거나 기본 이름과 설명을 유지합니다. 이 새 보안 그룹에는 리스너에 구성된 포트로 보내는 트래픽을 허용하는 규칙이 포함되어 있습니다.
 - c. [Next: Configure Routing]을 선택합니다.
8. [Configure Routing] 페이지에서 다음을 수행합니다.
 - a. [Target group]에서는 기본 값인 [New target group]을 유지합니다.
 - b. [Name]에 대상 그룹의 이름을 입력합니다.
 - c. [Protocol]을 HTTP로, [Port]를 80으로 유지합니다.
 - d. [Health checks]의 기본 프로토콜과 경로를 유지합니다.
 - e. [Next: Register Targets]를 선택합니다.
 9. Auto Scaling을 이용해 EC2 인스턴스를 대상 그룹에 추가해야 하므로 [Register Targets] 페이지에서 [Next: Review]를 선택하여 다음 페이지로 계속합니다.
 10. [Review] 페이지에서 [Create]를 선택합니다. 로드 밸런서가 생성된 후 [Close]를 선택합니다.
 11. 탐색 창의 [AUTO SCALING]에서 [Launch Configurations]를 선택합니다.
 - Auto Scaling을 처음 사용하는 경우 시작 페이지가 표시됩니다. [Create Auto Scaling group]을 선택하여 Auto Scaling 그룹 생성 마법사를 시작한 다음 [Create launch configuration]을 선택합니다.
 - 그렇지 않으면 [Create launch configuration]을 선택합니다.
 12. [Choose AMI] 페이지에서 [My AMIs] 탭을 선택한 다음, [사전 조건 \(p. 55\)](#)에서 생성한 AMI를 선택합니다.
 13. [Choose Instance Type] 페이지에서 인스턴스 유형을 선택한 다음 [Next: Configure details]를 선택합니다.
 14. [Configure details(세부 정보 구성)] 페이지에서 다음을 수행합니다.
 - a. [Name]에 시작 구성의 이름을 입력합니다(예: `my-launch-config`).
 - b. [IAM role]에서, [사전 조건 \(p. 55\)](#)에서 만든 IAM 역할을 선택합니다.
 - c. (선택 사항) 스타트업 스크립트를 실행해야 하는 경우 [Advanced Details]를 확장하고 [User data]에 스크립트를 입력합니다.
 - d. [Skip to review]를 선택합니다.
 15. [Review] 페이지에서 [Edit security groups]를 선택합니다. 기존의 보안 그룹을 선택하거나 새로 만들 수 있습니다. 이 보안 그룹은 로드 밸런서의 HTTP 트래픽과 상태 확인을 허용해야 합니다. 인스턴스에 퍼블릭 IP 주소가 있는 경우 인스턴스에 연결해야 한다면 선택적으로 SSH 트래픽을 허용할 수 있습니다. 작업을 마쳤으면 [Review]를 선택합니다.
 16. [Review] 페이지에서 [Create launch configuration]를 선택합니다.
 17. 메시지가 나타나면 기존 키 페어를 선택하거나 새 키 페어를 만들거나 키 페어 없이 진행합니다. 승인 확인란을 선택한 다음 [Create launch configuration]을 선택합니다.
 18. 시작 구성이 생성된 후에는 Auto Scaling 그룹을 만들어야 합니다.
 - Auto Scaling을 처음 사용하고 Auto Scaling 그룹 생성 마법사를 사용 중인 경우 다음 단계로 자동으로 이동합니다.
 - 그렇지 않으면 [Create an Auto Scaling group using this launch configuration]을 선택합니다.
 19. [Configure Auto Scaling group details] 페이지에서 다음 작업을 수행하십시오.
 - a. [Group name]에 Auto Scaling 그룹의 이름을 입력합니다. 예를 들면 `my-asg`를 입력하십시오.
 - b. [Group size]에 인스턴스 수를 입력합니다(예: 2). 각 가용 영역에서 인스턴스 수를 대략적으로 동일하게 유지하는 것이 좋습니다.
 - c. [Network]에서 VPC를 선택하고 [Subnet]에서 2개의 퍼블릭 서브넷을 선택합니다.
 - d. [Advanced Details] 아래에서 [Receive traffic from one or more load balancers]를 선택합니다. [Target Groups]에서 대상 그룹을 선택합니다.
 - e. [Next: Configure scaling policies]를 선택합니다.

20. Auto Scaling에서 그룹을 지정된 크기로 유지할 예정이므로 [Configure scaling policies] 페이지에서 [Review]를 선택합니다. 나중에 이 Auto Scaling 그룹을 수동으로 확장하거나, 일정에 따라 그룹을 확장하도록 구성하거나, 필요에 따라 그룹을 확장하도록 구성할 수 있습니다.
21. [Review] 페이지에서 [Create Auto Scaling group]을 선택합니다.
22. 그룹이 생성된 후 [Close]를 선택합니다.

로드 밸런서 테스트

클라이언트가 로드 밸런서에 요청을 보내면 로드 밸런서는 그 요청을 등록된 인스턴스 중 하나로 라우팅합니다.

로드 밸런서를 테스트하려면

1. 인스턴스가 준비되었는지 확인합니다. [Auto Scaling Groups] 페이지에서 Auto Scaling 그룹을 선택한 다음 [Instances] 탭을 선택합니다. 처음에는 인스턴스가 Pending 상태로 되어 있습니다. 상태가 InService이면, 사용할 준비가 된 것입니다.
2. 인스턴스가 로드 밸런서에 등록되어 있는지 확인합니다. [Target Groups] 페이지에서 대상 그룹을 선택한 다음 [Targets] 탭을 선택합니다. 인스턴스의 상태가 initial이면, 아직 등록 중일 수도 있습니다. 인스턴스의 상태가 healthy이면 사용할 준비가 된 것입니다. 인스턴스가 준비되면 다음과 같이 로드 밸런서를 테스트할 수 있습니다.
3. [Load Balancers] 페이지에서 로드 밸런서를 선택합니다.
4. [Description] 탭에서 DNS 이름을 찾습니다. 이름은 다음과 같은 형식으로 되어 있습니다.

`my-lb-xxxxxxxxxx.us-west-2.elb.amazonaws.com`

5. 웹 브라우저에서 로드 밸런서의 DNS 이름을 주소 표시줄에 붙여넣기하고 Enter 키를 누릅니다. 웹 사이트가 표시됩니다.

자습서: Amazon EC2 인스턴스 원격 관리

이 자습서에서는 로컬 시스템에서 시스템 관리자 Run Command를 사용하여 Amazon EC2 인스턴스를 원격으로 관리하는 방법을 보여 줍니다. 이 자습서에는 Amazon EC2 콘솔, Windows PowerShell용 AWS 도구, 및 AWS Command Line Interface를 사용하여 명령을 실행하는 절차가 나와 있습니다.

Note

Run Command로 온프레미스 환경 또는 다른 클라우드 공급자가 제공하는 환경에서 서버 및 VM(가상 머신)을 관리할 수도 있습니다. 자세한 내용은 [Setting Up 시스템 관리자 in Hybrid Environments](#)를 참조하십시오.

시작하기 전

시스템 관리자에 AWS Identity and Access Management(IAM) 인스턴스 프로파일 역할을 구성해야 합니다. AmazonEC2RoleforSSM 역할을 Amazon EC2 인스턴스에 추가합니다. 이 역할을 사용하면 인스턴스가 Systems Manager API와 통신할 수 있습니다. 역할을 기존 인스턴스에 추가하는 방법에 대한 자세한 내용은 [IAM 역할을 인스턴스에 연결 \(p. 461\)](#) 단원을 참조하십시오.

또한 다음 섹션에서 설명하겠지만 시스템 관리자에 IAM 사용자 계정을 구성해야 합니다.

사용자 계정에 시스템 관리자 액세스 권한 부여

사용자 계정이 SSM API와 통신하도록 구성되어 있어야 합니다. 다음 절차에 따라 SSM API 작업에 대한 완전한 액세스 권한을 부여하는 관리형 AWS Identity and Access Management(IAM) 정책을 사용자 계정에 주어십시오.

사용자 계정에 대한 IAM 정책을 생성하려면

1. <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 [Policies]를 선택합니다. (IAM을 처음 사용하는 경우 [Get Started]를 선택한 다음 [Create Policy]를 선택합니다.)
3. [Filter] 필드에 **AmazonSSMFullAccess**를 입력하고 Enter 키를 누릅니다.
4. [AmazonSSMFullAccess] 옆의 확인란을 선택하고 [Policy Actions]와 [Attach]를 차례로 선택합니다.
5. [Attach Policy] 페이지에서 사용자 계정을 선택한 다음 [Attach Policy]를 선택합니다.

SSM 에이전트 설치(Linux)

SSM 에이전트는 Run Command 요청을 처리하고 요청에서 지정한 인스턴스를 구성합니다. Windows 인스턴스에서는 에이전트가 기본적으로 설치됩니다. 하지만 Linux에서는 에이전트를 수동으로 설치해야 합니다. 다음 절차에서는 RHEL(Red Hat Enterprise Linux)에 에이전트를 설치하는 방법을 설명합니다. Ubuntu, Amazon Linux 또는 CentOS에 설치하는 방법에 대한 자세한 내용은 [Installing SSM Agent On Linux](#) 단원을 참조하십시오.

Red Hat Enterprise Linux에 SSM 에이전트를 설치하는 방법은 다음과 같습니다.

1. RHEL 인스턴스에 연결하고 그 인스턴스에 임시 디렉터리를 만듭니다.

```
mkdir /tmp/ssm
```

2. 다음 명령 중 하나를 사용하여 SSM 설치 관리자를 임시 디렉터리에 다운로드합니다.

64비트

```
curl https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/linux_amd64/amazon-ssm-agent.rpm -o /tmp/ssm/amazon-ssm-agent.rpm
```

32비트

```
curl https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/linux_386/amazon-ssm-agent.rpm -o /tmp/ssm/amazon-ssm-agent.rpm
```

3. SSM 설치 관리자를 실행합니다.

```
sudo yum install -y /tmp/ssm/amazon-ssm-agent.rpm
```

4. 다음 명령 중 하나를 실행하여 SSM 에이전트가 실행 중인지 확인합니다. 이 명령은 "amazon-ssm-agent is running"을 반환해야 합니다.

RHEL 7.x

```
sudo systemctl status amazon-ssm-agent
```

RHEL 6.x

```
sudo status amazon-ssm-agent
```

5. 이전 명령에서 "amazon-ssm-agent is stopped"가 반환되는 경우 다음 명령을 실행합니다.

- a. 서비스를 시작합니다.

RHEL 7.x

```
sudo systemctl start amazon-ssm-agent
```

RHEL 6.x

```
sudo start amazon-ssm-agent
```

- b. 에이전트의 상태를 확인합니다.

RHEL 7.x

```
sudo systemctl status amazon-ssm-agent
```

RHEL 6.x

```
sudo status amazon-ssm-agent
```

EC2 콘솔을 사용하여 명령 보내기

다음 절차에 따라 Amazon EC2 콘솔에서 Run Command를 사용하여 해당 인스턴스에서 실행 중인 모든 서비스를 나열합니다.

콘솔에서 Run Command를 사용하여 명령을 실행하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Run Command를 선택합니다.
3. [Run a command]를 선택합니다.
4. [Command document]에서 Windows 인스턴스라면 [AWS-RunPowerShellScript]를, 그리고 Linux 인스턴스라면 [AWS-RunShellScript]를 선택합니다.
5. Target instances에서 앞서 생성한 인스턴스를 선택합니다. 새로 만든 인스턴스가 표시되지 않으면 현재 그 인스턴스와 같은 리전에 있는지 확인합니다. 그리고 앞부분에서 설명한 대로 IAM 역할과 신뢰 정책을 구성했는지도 확인합니다.
6. [Commands]에서 Windows라면 **Get-Service**를, 그리고 Linux라면 **ps -aux | less**를 입력합니다.
7. (선택 사항) Working Directory에서 명령을 실행할 EC2 인스턴스의 폴더 경로를 지정합니다.
8. (선택 사항) Execution Timeout에서 명령 시간이 초과되어 명령이 실패하기 전까지 EC2Config 서비스 또는 SSM 에이전트에서 명령 실행을 시도할 시간(초)을 지정합니다.
9. Comment에서 명령 목록에서 이 명령을 식별하는 데 도움이 될 정보를 제공하는 것이 좋습니다.
10. Timeout (seconds)에서 인스턴스에 접속할 수 없다고 간주되어 명령 실행이 실패하기 전까지 Run Command가 인스턴스에 접속을 시도하는 시간(초)을 입력합니다.
11. [Run]을 선택하여 명령을 실행합니다. Run Command가 상태 화면에 표시됩니다. [View result]를 선택합니다.
12. 출력을 보려면 명령에 대한 명령 호출을 선택하고, Output 탭을 선택한 다음, View Output을 선택합니다.

Run Command를 사용하여 명령을 실행하는 방법에 대한 자세한 내용은 [Executing Commands Using 시스템 관리자 Run Command](#) 단원을 참조하십시오.

Windows PowerShell용 AWS 도구를 사용하여 명령 보내기

다음 절차에 따라 Windows PowerShell용 AWS 도구에서 Run Command를 사용하여 해당 인스턴스에서 실행 중인 모든 서비스를 나열합니다.

명령을 실행하려면

- 로컬 컴퓨터에서 [Windows PowerShell용 AWS 도구](#) 최신 버전을 다운로드합니다.
- 로컬 컴퓨터에서 Windows PowerShell용 AWS 도구를 열고 다음 명령을 실행하여 자격 증명을 지정합니다.

```
Set-AWSCredentials -AccessKey key -SecretKey key
```

- 다음 명령을 실행하여 PowerShell 세션의 리전을 설정합니다. 이전 절차에서 인스턴스를 만들었던 리전을 지정합니다. 이 예에서는 us-west-2 리전을 사용합니다.

```
Set-DefaultAWSRegion -Region us-west-2
```

- 다음 명령을 실행하여 해당 인스턴스에서 실행 중인 서비스를 검색합니다.

```
Send-SSMCommand -InstanceId 'Instance-ID' -DocumentName AWS-RunPowerShellScript -Comment 'listing services on the instance' -Parameter @{'commands'=@('Get-Service')}
```

이 명령으로 반환되는 명령 ID를 사용하여 결과를 확인할 수 있습니다.

- 다음 명령은 원래 Send-SSMCommand 출력을 반환합니다. 2,500자를 초과하는 출력 부분은 잘립니다. 전체 서비스 목록을 보려면 -OutputS3BucketName *bucket_name* 파라미터를 사용하여 명령에 Amazon S3 버킷을 지정하십시오.

```
Get-SSMCommandInvocation -CommandId Command-ID -Details $true | select -ExpandProperty CommandPlugins
```

Windows PowerShell용 도구에서 Run Command를 사용하여 명령을 실행하는 방법에 대한 자세한 내용은 [시스템 관리자 Run Command Walkthrough Using the Windows PowerShell용 AWS 도구](#) 단원을 참조하십시오.

AWS CLI를 사용하여 명령 보내기

다음 절차에 따라 AWS CLI에서 Run Command를 사용하여 해당 인스턴스에서 실행 중인 모든 서비스를 나열합니다.

명령을 실행하려면

- 로컬 컴퓨터에서 [AWS Command Line Interface\(AWS CLI\)](#) 최신 버전을 다운로드합니다.
- 로컬 컴퓨터에서 AWS CLI를 열고 다음 명령을 실행하여 자격 증명과 리전을 지정합니다.

```
aws configure
```

- 시스템에서 다음을 지정하라는 메시지를 표시합니다.

```
AWS Access Key ID [None]: key
AWS Secret Access Key [None]: key
Default region name [None]: region, for example us-east-1
```

Default output format [None]: ENTER

4. 다음 명령을 실행하여 해당 인스턴스에서 실행 중인 서비스를 검색합니다.

```
aws ssm send-command --document-name "AWS-RunShellScript" --comment "listing services" --instance-ids "Instance-ID" --parameters commands="service --status-all" --region us-west-2 --output text
```

이 명령으로 반환되는 명령 ID를 사용하여 결과를 확인할 수 있습니다.

5. 다음 명령은 원래 Send-SSMCommand 출력을 반환합니다. 2,500자를 초과하는 출력 부분은 잘립니다. 전체 서비스 목록을 보려면 --output-s3-bucket-name *bucket_name* 파라미터를 사용하여 명령에 Amazon S3 버킷을 지정해야 합니다.

```
aws ssm list-command-invocations --command-id "command ID" --details
```

AWS CLI에서 Run Command를 사용하여 명령을 실행하는 방법에 대한 자세한 내용은 [시스템 관리자 Run Command Walkthrough Using the AWS CLI](#) 단원을 참조하십시오.

관련 내용

Run Command 및 시스템 관리자에 대한 자세한 내용은 다음 주제와 참고 자료를 참조하십시오.

- [Amazon EC2 시스템 관리자 사용 설명서](#)
- [Amazon EC2 Systems Manager API Reference](#)
- [시스템 관리자 Windows PowerShell용 AWS 도구 Reference](#)
- [시스템 관리자 AWS Command Line Interface Reference](#)
- [AWS SDK](#)

동영상은 [AWS Instructional Videos and Labs](#)를 참조하십시오.

Amazon 머신 이미지(AMI)

Amazon 머신 이미지(AMI)는 클라우드의 가상 서버인 인스턴스를 시작하는 데 필요한 정보를 제공합니다. 인스턴스를 시작할 때 AMI를 지정해야 하며, AMI에서 필요한 만큼 많은 인스턴스를 시작할 수 있습니다. 또한, 필요한 만큼의 서로 다른 AMI에서 인스턴스를 시작할 수 있습니다.

AMI는 다음을 포함합니다.

- 인스턴스 루트 볼륨 템플릿(예: 운영 체제, 애플리케이션 서버, 애플리케이션)
- AMI를 사용하여 인스턴스를 시작할 수 있는 AWS 계정을 제어하는 시작 권한
- 시작될 때 인스턴스에 연결할 볼륨을 지정하는 블록 디바이스 매핑

AMI 사용

다음 다이어그램은 AMI 수명 주기를 요약하여 설명합니다. AMI를 생성 및 등록한 다음 새 인스턴스를 시작하기 위해 그것을 사용할 수 있습니다. (AMI 소유자가 시작 권한을 부여한 경우 AMI에서 인스턴스를 시작할 수 있습니다.) AMI를 동일 리전 또는 다른 리전으로 복사할 수 있습니다. AMI에서 인스턴스 시작이 완료되면 AMI를 등록할 수 있습니다.

인스턴스의 기준을 충족하는 AMI를 검색할 수 있습니다. AWS에서 제공하는 AMI 또는 커뮤니티에서 제공하는 AMI를 검색할 수 있습니다. 자세한 내용은 [AMI 유형 \(p. 63\)](#) 및 [Linux AMI 찾기 \(p. 67\)](#) 섹션을 참조하십시오.

인스턴스에 연결되면 사용자는 인스턴스를 다른 서버와 동일한 방식으로 사용할 수 있습니다. 인스턴스 시작, 연결 및 사용에 대한 자세한 내용은 [Amazon EC2 인스턴스 \(p. 145\)](#) 섹션을 참조하십시오.

고유 AMI 생성

퍼블릭 AMI에서 시작된 인스턴스를 최적화한 다음 해당 구성은 자체적으로 사용하기 위한 사용자 정의 AMI로 저장할 수 있습니다. AMI에서 시작된 인스턴스는 사용자가 이전에 생성한 모든 사용자 정의를 사용합니다.

인스턴스의 루트 스토리지 디바이스는 어떤 프로세스로 AMI가 생성될 수 있는지를 결정합니다. 인스턴스의 루트 볼륨은 Amazon EBS 볼륨 또는 인스턴스 스토어 볼륨입니다. 자세한 내용은 [Amazon EC2 루트 디바이스 볼륨 \(p. 11\)](#) 섹션을 참조하십시오.

Amazon EBS 지원 AMI를 생성하려면 [Amazon EBS 지원 Linux AMI 생성 \(p. 81\)](#) 섹션을 참조하십시오. 인스턴스 스토어 지원 AMI를 생성하려면 [인스턴스 스토어 기반 Linux AMI 생성 \(p. 84\)](#) 섹션을 참조하십시오.

AMI를 범주화하고 관리하기 위해 사용자는 AMI에 사용자 정의 태그를 할당할 수 있습니다. 자세한 내용은 [Amazon EC2 리소스에 태그 지정 \(p. 681\)](#) 섹션을 참조하십시오.

AMI 구입, 공유 및 판매

AMI를 생성한 후 사용자는 AMI를 프라이빗으로 유지하여 자체적으로 사용하거나 특정 AWS 계정 목록과 공유할 수 있습니다. 또한 사용자 정의 AMI를 퍼블릭으로 설정하여 커뮤니티에서 사용되도록 할 수 있습니다. 간단한 몇 단계만 수행하면 간단한 프로세스를 통해 안전하고 사용이 가능하며 보안이 제공되는 퍼블릭 AMI를 구축할 수 있습니다. AMI 사용 및 공유 방법에 대한 자세한 내용은 [공유 AMI \(p. 69\)](#) 섹션을 참조하십시오.

Red Hat와 같은 업체와 서비스 계약을 맺고 제공되는 AMI 등 AMI를 타사에서 구입하는 것도 가능합니다. 또한, AMI를 생성한 후 다른 Amazon EC2 사용자에게 판매할 수도 있습니다. AMI 구입 및 판매에 대한 자세한 내용은 [유료 AMI \(p. 78\)](#) 섹션을 참조하십시오.

AMI 등록 해제

관련 작업이 완료되면 AMI의 등록을 해제할 수 있습니다. AMI의 등록을 해제한 이후에는 새 인스턴스를 시작하기 위해 해당 AMI를 사용하는 것을 불가능합니다. 자세한 내용은 [AMI 등록 취소 \(p. 130\)](#) 섹션을 참조하십시오.

Amazon Linux

Amazon Linux AMI는 AWS가 제공하는 Linux 이미지로 지원 및 유지됩니다. Amazon Linux이 제공하는 일부 기능은 다음과 같습니다.

- Amazon EC2에서 실행되는 애플리케이션을 위한 안정적이고 안전한 고성능 실행 환경.
- 추가 요금 없이 Amazon EC2 사용자에게 제공됨.
- 다양한 버전의 MySQL, PostgreSQL, Python, Ruby, Tomcat 및 많은 표준 패키지에 대한 리포지토리 액세스
- 정기적으로 제공되는 업데이트에는 최신 구성 요소가 포함되고 이러한 업데이트는 실행 중인 인스턴스에 설치될 수 있도록 yum 레포지토리에서 이용할 수 있습니다.
- AWS CLI, Amazon EC2 API 및 AMI 도구, Python용 Boto 라이브러리 및 Elastic Load Balancing 도구 등과 같이 AWS 서비스와 쉽게 통합할 수 있는 패키지가 포함되어 있습니다.

자세한 내용은 [Amazon Linux \(p. 132\)](#) 섹션을 참조하십시오.

AMI 유형

다음 유형을 기준으로 사용할 AMI를 선택할 수 있습니다.

- 리전([리전 및 가용 영역 \(p. 7\)](#) 참조)

- 운영 체제
- 아키텍처(32비트 또는 64비트)
- [시작 권한 \(p. 64\)](#)
- [루트 디바이스 스토리지 \(p. 64\)](#)

시작 권한

AMI 소유자는 시작 권한을 지정하여 가용성을 결정합니다. 시작 권한은 다음 범주로 분류됩니다.

시작 권한	설명
퍼블릭	소유자는 모든 AWS 계정에 시작 권한을 부여합니다.
명시적	소유자는 특정 AWS 계정에 시작 권한을 부여합니다.
암묵적	소유자는 AMI에 대한 암묵적인 시작 권한을 갖습니다.

Amzon 및 Amazon EC2 커뮤니티는 퍼블릭 AMI에 대한 다양한 선택권을 제공합니다. 자세한 내용은 [공유 AMI \(p. 69\)](#) 섹션을 참조하십시오. 개발자들은 자신의 AMI에 비용을 부과할 수 있습니다. 자세한 내용은 [유료 AMI \(p. 78\)](#) 섹션을 참조하십시오.

루트 디바이스 스토리지

모든 AMI는 Amazon EBS에 의해 지원되는 유형 또는 인스턴스 스토어에 의해 지원되는 유형으로 분류됩니다. 전자는 AMI에서 시작된 인스턴스의 루트 디바이스가 Amazon EBS 스냅샷에서 생성된 Amazon EBS 볼륨이라는 것을 의미합니다. 후자는 AMI에서 시작된 인스턴스의 루트 디바이스가 Amazon S3에 저장된 템플릿에서 생성된 인스턴스 스토어 볼륨이라는 것을 의미합니다. 자세한 내용은 [Amazon EC2 루트 디바이스 볼륨 \(p. 11\)](#) 섹션을 참조하십시오.

이 섹션은 두 AMI 유형의 주요 차이점을 요약합니다. 다음 표를 통해 그러한 차이를 빠르게 확인할 수 있습니다.

특성	Amazon EBS 지원	Amazon 인스턴스 스토어 지원
부팅 시간	일반적으로 1분 이하	일반적으로 5분 이하
크기 제한	16TiB	10GiB
루트 디바이스 볼륨	Amazon EBS 볼륨	인스턴스 스토어 볼륨
데이터 지속성	기본적으로 인스턴스가 종료되면 루트 볼륨이 삭제됩니다.* 기타 Amazon EBS 볼륨의 데이터는 기본적으로 인스턴스 종료 후에도 유지됩니다. 모든 인스턴스 스토어의 데이터는 인스턴스 수명 주기 동안만 유지됩니다.	모든 인스턴스 스토어의 데이터는 인스턴스 수명 주기 동안만 유지됩니다. 기타 Amazon EBS 볼륨의 데이터는 기본적으로 인스턴스 종료 후에도 유지됩니다.
업데이트	인스턴스 유형, 커널 RAM 디스크 및 사용자 데이터는 인스턴스가 종지된 동안에 변경될 수 있습니다.	인스턴스 속성은 인스턴스 수명 주기 동안 고정됩니다.
요금	인스턴스 사용량, Amazon EBS 볼륨 사용량 및 AMI를 Amazon EBS 스냅샷으로 저장하는 것에 대한 비용이 청구됩니다.	인스턴스 사용량 및 Amazon S3에 AMI를 저장하는 것에 대한 비용이 청구됩니다.

특성	Amazon EBS 지원	Amazon 인스턴스 스토어 지원
AMI 생성/번들링	단일 명령/호출을 사용합니다	AMI 도구를 설치 및 사용해야 합니다
중지 상태	인스턴스가 실행 중이 아니면 중지 상태가 될 수 있지만 루트 볼륨은 Amazon EBS에 유지됩니다	중지 상태가 될 수 없습니다. 인스턴스가 실행 중이거나 종료되었습니다

* 기본적으로, Amazon EBS 지원 인스턴스 루트 볼륨에서는 `DeleteOnTermination` 플래그가 `true`로 설정됩니다. 이 플래그를 변경하여 종료 후에도 볼륨을 유지하는 방법에 대한 자세한 내용은 [루트 디바이스 볼륨이 계속 유지되도록 변경 \(p. 14\)](#) 섹션을 참조하십시오.

AMI의 루트 디바이스 유형 결정

콘솔을 이용하여 AMI의 루트 디바이스 유형을 결정하려면

1. Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [AMIs]를 클릭한 다음 AMI를 선택합니다.
3. [Details] 탭에서 [Root Device Type]의 값을 다음과 같이 확인합니다.
 - 값이 `ebs`이면, Amazon EBS 지원 AMI입니다.
 - 값이 `instance store`이면, 인스턴스 스토어 지원 AMI입니다.

명령줄을 이용하여 AMI의 루트 디바이스 유형을 결정하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [describe-images](#) (AWS CLI)
- [Get-EC2Image](#) (Windows PowerShell용 AWS 도구)

중지 상태

Amazon EC2 인스턴스 스토어 지원 인스턴스가 아닌 Amazon EBS 지원 인스턴스는 중지가 가능합니다. 중지를 하면 인스턴스는 실행이 중지됩니다(상태가 `running`에서 `stopping ~ stopped`으로 변함). 중지된 인스턴스는 Amazon EBS에서 유지되어 다시 시작하는 것이 가능합니다. 중지와 종료는 다른 것입니다. 종료된 인스턴스는 다시 시작할 수 없습니다. Amazon EC2 인스턴스 스토어 지원 AMI는 중지될 수 없기 때문에 실행 또는 종료 상태 둘 중 하나가 됩니다. 인스턴스 중지로 인한 영향 및 해결 방법에 대한 자세한 내용은 [인스턴스 중지 및 시작 \(p. 285\)](#) 섹션을 참조하십시오.

기본 데이터 스토리지 및 유지

루트 디바이스에서 인스턴스 스토어 볼륨을 사용하는 인스턴스는 자동으로 인스턴스 스토어를 사용할 수 있습니다(루트 볼륨에 루트 파티션이 포함되고 추가 데이터를 저장 가능). 인스턴스 스토어 볼륨에 있는 모든 데이터는 인스턴스에서 오류가 발생하거나 인스턴스가 종료되면 삭제됩니다(루트 디바이스의 데이터 제외). 1개 이상의 Amazon EBS 볼륨을 연결하여 인스턴스에 영구 스토리지를 추가할 수 있습니다.

루트 디바이스에서 Amazon EBS를 사용하는 인스턴스는 자동으로 Amazon EBS 볼륨에 연결됩니다. 이 볼륨은 다른 볼륨과 마찬가지로 볼륨 목록에 표시됩니다. 기본적으로 인스턴스는 모든 사용 가능한 인스턴스 스토어 볼륨을 사용하지 않습니다. 인스턴스 스토리지 또는 추가 Amazon EBS 볼륨은 블록 디바이스 매핑을 이용하여 추가될 수 있습니다. 자세한 내용은 [블록 디바이스 매핑 \(p. 662\)](#) 섹션을 참조하십시오. 인스턴스가 정지되는 경우 인스턴스 스토어 볼륨에서 어떤 일이 일어나는지에 대한 자세한 내용은 [인스턴스 중지 및 시작 \(p. 285\)](#) 섹션을 참조하십시오.

부팅 시간

Amazon EBS 지원 AMI는 Amazon EC2 인스턴스 스토어 지원 AMI보다 빠르게 시작됩니다. Amazon EC2 인스턴스 스토어 지원 AMI가 시작되는 경우 인스턴스가 사용 가능해지기 전에 모든 요소가 Amazon S3에서 검색되어야 합니다. Amazon EBS 지원 AMI의 경우 인스턴스가 사용 가능해지기 전에 인스턴스 부팅에 필요한 요소만 스냅샷에서 검색되면 됩니다. 그러나 스냅샷에서 나머지 요소를 검색하고 볼륨으로 로드되는 동안 루트 디바이스에서 Amazon EBS 볼륨을 사용하는 인스턴스의 성능은 잠시 느려질 수 있습니다. 인스턴스를 종지한 다음 다시 시작하면 Amazon EBS 볼륨에 상태가 저장되어 빠르게 시작됩니다.

AMI 생성

인스턴스 스토어에서 지원하는 Linux AMI를 생성하려면 Amazon EC2 AMI 도구를 사용하여 인스턴스 자체의 인스턴스에서 AMI를 생성해야 합니다.

Amazon EBS 지원 AMI에서 AMI를 생성하는 것이 훨씬 쉽습니다. `CreateImage` AMI 작업을 통해 Amazon EBS 지원 AMI를 생성하고 등록할 수 있습니다. 또한 AWS Management Console에는 실행 상태의 AMI를 생성하는 버튼이 있습니다. 자세한 내용은 [Amazon EBS 지원 Linux AMI 생성 \(p. 81\)](#) 섹션을 참조하십시오.

요금 부과 방법

인스턴스 스토어 지원 AMI의 경우 AMI 스토리지 및 인스턴스 사용량에 따라 비용이 청구됩니다. Amazon EBS 지원 AMI의 경우 AMI와 인스턴스 사용 요금과 함께 볼륨 스토리지 및 사용량에 대한 비용이 청구됩니다.

Amazon EC2 인스턴스 스토어 지원 AMI의 경우 사용자가 AMI를 사용자 정의하여 새 AMI를 생성할 때마다 모든 요소가 각 AMI의 Amazon S3에 저장됩니다. 그러므로 각 사용자 정의 AMI의 스토리지 크기가 AMI의 전체 크기가 됩니다. Amazon EBS 지원 AMI의 경우 사용자가 AMI를 사용자 정의하여 새 AMI를 생성할 때마다 변경 사항만이 저장됩니다. 그러므로 최초 AMI 이후 사용자 지정한 AMI의 스토리지는 크기가 훨씬 작아 AMI 스토리지 비용이 훨씬 낮아집니다.

Amazon EBS 지원 인스턴스가 정지되면 인스턴스 사용에 대한 비용이 청구되지 않지만 볼륨 스토리지에 대한 비용은 계속해서 발생합니다. 인스턴스를 종지에서 실행으로 상태를 전환할 때마다 전체 인스턴스 시간 비용이 청구되며, 1시간 내에 여러 번 전환된 경우에도 동일하게 청구됩니다. 예를 들어, 인스턴스의 시간당 인스턴스 비용이 \$0.10인 경우 정지하지 않고 1시간 동안 인스턴스를 실행한 경우의 비용은 \$0.10입니다. 해당 시간 동안 인스턴스를 종지한 다음 다시 시작하면 사용 시간 비용으로 \$0.30이 청구됩니다(초기 \$0.10 더하기 재시작 비용으로 2 x \$0.10).

Linux AMI 가상화 유형

Linux Amazon 머신 이미지은 PV(반가상화) 또는 HVM(하드웨어 가상 머신)의 두 가지 유형의 가상화를 사용합니다. PV AMI와 HVM AMI의 주요 차이점은 부팅 방법과 더 나은 성능을 위해 특수 하드웨어 확장(CPU, 네트워크, 스토리지)을 활용할 수 있는지 여부에 있습니다.

최상의 성능을 위해서는 인스턴스를 시작할 때 현재 세대 인스턴스 유형 및 HVM AMI를 사용하는 것이 좋습니다. 현재 세대 인스턴스 유형에 대한 자세한 내용은 [Amazon EC2 인스턴스](#) 정보 페이지를 참조하십시오. 이전 세대 인스턴스 유형을 사용 중인 경우 업그레이드하려면 자세한 내용은 [업그레이드 경로](#)를 참조하십시오.

각 인스턴스 유형에 대해 Amazon Linux 권장 유형에 대한 자세한 내용은 [Amazon Linux AMI 인스턴스 유형](#) 정보 페이지를 참조하십시오.

HVM AMI

HVM AMI는 이미지 루트 볼륨 디바이스의 마스터 부트 레코드를 실행하여 완벽하게 가상화된 하드웨어 및 부트 세트를 함께 제공합니다. 이 가상화 유형은 운영 체제 미설치 하드웨어에서 실행될 때처럼 가상 머신에

서 운영 체제를 수정하지 않고 실행할 수 있습니다. Amazon EC2 호스트 시스템은 게스트에게 제공되는 기본 하드웨어의 일부 또는 모두를 에뮬레이트합니다.

PV 게스트와 달리 HVM 게스트는 하드웨어 확장을 활용하여 호스트 시스템의 기본 하드웨어에 빠르게 액세스할 수 있습니다. Amazon EC2에서 제공되는 CPU 가상화 확장에 대한 자세한 내용은 Intel 웹 사이트의 [Intel Virtualization Technology](#)를 참조하십시오. 향상된 네트워킹 및 GPU 처리를 활용하려면 HVM AMI가 필요합니다. 특수 네트워크 및 GPU 디바이스에 대한 명령을 통과하기 위해 OS는 기본 하드웨어 플랫폼에 액세스할 수 있어야 하고, HVM 가상화는 이 액세스 기능을 제공합니다. 자세한 내용은 [Enhanced Networking \(p. 533\)](#) 및 [Linux 액셀러레이티드 컴퓨팅 인스턴스 \(p. 162\)](#) 섹션을 참조하십시오.

모든 최신 인스턴스 유형은 HVM AMI를 지원합니다. CC2, CR1, HI1 및 HS1 이전 세대 인스턴스 유형은 Linux HVM AMI를 지원합니다.

HVM AMI를 찾으려면 콘솔 또는 [describe-images](#) 명령을 사용하여 AMI의 가상화 유형이 `hvm`으로 설정되어 있는지 확인합니다.

PV AMI

PV AMI는 PV-GRUB라는 특수 부트 로더를 통해 부팅되며, 이는 부팅 주기를 시작한 다음 이미지의 `menu.lst` 파일에 지정된 커널을 체인 로드합니다. 반가상화 게스트는 가상화를 명시적으로 지원하지 않는 호스트 하드웨어에서 실행될 수 있지만, 향상된 네트워킹 또는 GPU 처리와 같은 특수 하드웨어 확장을 활용할 수 없습니다. 이전에는 대부분의 경우 PV 게스트가 HVM 게스트보다 더 나은 성능을 제공했지만, HVM 가상화 기능이 향상되고 HVM AMI용 PV 드라이버가 제공되는 현재는 더 이상 그렇지 않습니다. PV-GRUB 및 Amazon EC2에서의 사용에 대한 자세한 내용은 [사용자 제공 커널 \(p. 139\)](#) 섹션을 참조하십시오.

C3 및 M3 최신 인스턴스 유형은 PV AMI를 지원합니다. C1, HI1, HS1, M1, M2, T1 이전 세대 인스턴스 유형은 PV AMI를 지원합니다.

PV AMI를 찾으려면 콘솔 또는 [describe-images](#) 명령을 사용하여 AMI의 가상화 유형이 `paravirtual`로 설정되어 있는지 확인합니다.

HVM 기반 PV

이전에는 반가상화 게스트는 I/O용 특수 드라이버를 활용하여 네트워크 및 디스크 하드웨어 에뮬레이트 오버헤드를 방지할 수 있지만, HVM 게스트는 이러한 명령을 에뮬레이트된 하드웨어로 변환해야 했기 때문에, 반가상화 게스트가 HVM 게스트보다 스토리지 및 네트워크 운영 성능이 더 뛰어났습니다. 현재는 HVM 게스트용 PV 드라이버가 제공되므로 반가상화된 환경에서 실행하도록 이식할 수 없는 운영 체제(예: Windows)에서도 이러한 PV 드라이버를 통해 스토리지 및 네트워크 I/O 성능이 향상될 수 있습니다. HVM 게스트는 이러한 HVM 기반 PV 드라이버를 사용하여 반가상 게스트와 동일하거나 더 나은 성능을 제공할 수 있습니다.

Linux AMI 찾기

인스턴스를 시작하려면 사용할 AMI를 선택해야 합니다. AMI를 선택할 때 시작할 인스턴스에 대해 다음 요구 사항을 고려하십시오.

- 리전
- 운영 체제
- 아키텍처: 32비트(`i386`) 또는 64비트(`x86_64`)
- 루트 디바이스 유형: Amazon EBS 또는 인스턴스 스토어
- 공급자: Amazon Web Services, Oracle, IBM, Microsoft 또는 커뮤니티

Windows AMI를 찾아야 하는 경우 Windows 인스턴스용 Amazon EC2 사용 설명서의 [Finding a Windows AMI](#) 섹션을 참조하십시오.

목차

- Amazon EC2 콘솔을 사용하여 Linux AMI 찾기 (p. 68)
- AWS CLI를 사용하여 AMI 찾기 (p. 68)

Amazon EC2 콘솔을 사용하여 Linux AMI 찾기

Amazon EC2 콘솔을 사용하여 Linux AMI를 찾을 수 있습니다. [Images] 페이지를 사용하여 모든 사용 가능한 AMI를 검색하거나, 콘솔을 사용하여 인스턴스를 시작할 때 [Quick Launch] 탭에서 일반적으로 사용되는 AMI를 선택할 수 있습니다.

[Images] 페이지를 사용하여 Linux AMI를 찾으려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 모음에서 리전을 선택합니다. 현재 위치와 관계없이 사용자가 고를 수 있는 리전을 임의로 선택합니다. 이 리전에서 인스턴스가 시작됩니다.
3. 탐색 창에서 [AMIs]를 선택합니다.
4. (선택 사항) [Filter] 옵션을 사용하여 원하는 AMI만 표시하도록 표시되는 AMI 목록의 범위를 지정합니다. 예를 들어, AWS에서 제공하는 모든 Linux AMI를 나열하려면 [Public images]를 선택합니다. 검색 창을 선택하고 메뉴에서 [Owner]를 선택한 다음 [Amazon images]를 선택합니다. 검색 창을 다시 선택하고 [Platform]을 선택한 다음 제공된 목록에서 운영 체제를 선택합니다.
5. (선택 사항) [Show/Hide Columns] 아이콘을 선택하여 표시할 이미지 속성(예: 루트 디바이스 유형)을 선택합니다. 또는 목록에서 AMI를 선택하고 [Details] 탭에서 속성을 조회할 수 있습니다.
6. AMI를 선택하기 전에 해당 AMI가 인스턴스 스토어 기반인지, Amazon EBS 기반인지 확인하고 이 차이점에 따른 영향을 잘 알고 있어야 합니다. 자세한 내용은 [루트 디바이스 스토리지 \(p. 64\)](#) 섹션을 참조하십시오.
7. 이 AMI에서 인스턴스를 시작하려면 원하는 인스턴스를 선택한 다음 [Launch]를 선택합니다. 콘솔을 통한 인스턴스 시작에 대한 자세한 내용은 [Launching Your Instance from an AMI \(p. 266\)](#) 섹션을 참조하십시오. 아직 인스턴스를 시작할 준비가 되지 않은 경우 나중에 사용할 수 있도록 AMI ID(ami-xxxxxxxx)를 기록해 듦니다.

인스턴스 시작 시 Linux AMI를 찾으려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 콘솔 대시보드에서 [Launch Instance]를 선택합니다.
3. [Choose an Amazon Machine Image (AMI)] 페이지의 [Quick Start] 탭에 있는 목록에서 일반적으로 사용되는 AMI 중 하나를 선택합니다. 필요한 AMI가 표시되지 않는 경우 [AWS Marketplace] 또는 [Community AMIs] 탭을 선택하여 추가 AMI를 찾습니다.

AWS CLI를 사용하여 AMI 찾기

명령줄 파라미터를 사용하여 원하는 AMI 유형만 나열할 수 있습니다. 예를 들면 다음과 같이 [describe-images](#) 명령을 사용하여 사용자 또는 Amazon에서 소유한 퍼블릭 AMI를 찾을 수 있습니다.

```
$ aws ec2 describe-images --owners self amazon
```

Amazon EBS 기반 AMI만 표시하려면 이전 명령에 다음 필터를 추가합니다.

```
--filters "Name=root-device-type,Values=ebs"
```

요구 사항을 충족하는 AMI를 찾은 다음 해당 ID(ami-xxxxxxxx)를 기록해 드립니다. 이 AMI를 사용하여 인스턴스를 시작할 수 있습니다. 자세한 내용은 AWS Command Line Interface 사용 설명서의 [Launching an Instance Using the AWS CLI](#) 섹션을 참조하십시오.

공유 AMI

공유 AMI는 다른 개발자가 사용할 수 있도록 공유된 개발자 생성 AMI입니다. Amazon EC2를 처음 시작할 때 가장 손쉬운 방법 중 하나는 필요한 구성 요소를 가진 공유 AMI를 선택한 다음 개인 설정을 추가하는 것입니다. 자체 AMI를 생성하여 다른 사람과 공유할 수도 있습니다.

공유 AMI를 사용할 때는 사용자의 주의가 필요합니다. Amazon에서는 다른 Amazon EC2 사용자와 공유된 AMI의 무결성이나 보안성을 보장하지 않습니다. 따라서 공유 AMI를 사용할 때는 데이터 센터에서 외부 코드를 배포하는 경우와 마찬가지로 이런 AMI를 취급하고 그에 따라 적합한 조치를 취해야 합니다. 신뢰할 수 있는 출처의 AMI를 사용하십시오. 공유 AMI에 대한 질문이나 정보는 [AWS 포럼](#)을 이용하시기 바랍니다.

Amazon의 공개 이미지는 별칭을 소유주 이름으로 사용하며 계정 필드에 amazon가 표시됩니다. 따라서 Amazon에서 배포한 AMI를 쉽게 찾을 수 있습니다. 다른 사용자는 AMI 별칭을 사용할 수 없습니다.

AMI 생성에 대한 자세한 내용은 [인스턴스 스토어 지원 Linux AMI 생성](#) 또는 [Amazon EBS 지원 Linux AMI 생성](#)을 참조하십시오. AWS Marketplace에서 애플리케이션을 개발하고 제공하며 관리하는 정보는 [AWS Marketplace User Guide](#) 및 [AWS Marketplace Seller Guide](#)를 참조하십시오.

목차

- [공유 AMI 검색 \(p. 69\)](#)
- [퍼블릭 AMI 설정 \(p. 71\)](#)
- [지정한 AWS 계정과 AMI 공유 \(p. 72\)](#)
- [북마크 사용 \(p. 73\)](#)
- [공유 Linux AMI 지침 \(p. 74\)](#)

공유 AMI 검색

공유 AMI는 Amazon EC2 콘솔 또는 명령줄을 사용해 검색할 수 있습니다.

공유 AMI를 검색하는 방법(콘솔)

콘솔을 사용해 프라이빗 AMI를 검색하는 방법

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [AMIs]를 선택합니다.
3. [Private images]를 첫 필터로 선택합니다. 사용자와 공유된 모든 AMI가 나열됩니다. 검색 결과를 좀 더 세부적으로 보려면 [Search] 창을 선택하여 메뉴에서 제공하는 필터 옵션을 사용하십시오.

콘솔을 사용해 퍼블릭 AMI를 검색하는 방법

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [AMIs]를 선택합니다.
3. [Public images]를 첫 필터로 선택합니다. 검색 결과를 좀 더 세부적으로 보려면 [Search] 창을 선택하여 메뉴에서 제공하는 필터 옵션을 사용하십시오.
4. 필터를 사용하면 원하는 유형의 AMI만 검색할 수 있습니다. 예를 들어 [Owner :]를 선택한 다음 [Amazon images]를 선택하면 Amazon의 퍼블릭 이미지만 표시됩니다.

공유 AMI(명령줄) 검색

명령줄 도구를 사용해 퍼블릭 AMI를 검색하는 방법

[describe-images](#) 명령(AWS CLI)을 사용해 AMI를 나열합니다. 아래 예시와 같이 원하는 유형의 AMI만 나타나도록 목록을 정리할 수 있습니다.

다음 명령은 --executable-users 옵션을 사용하는 모든 퍼블릭 AMI를 나열합니다. 이 목록에는 사용자가 보유 중인 퍼블릭 AMI도 포함됩니다.

```
$ aws ec2 describe-images --executable-users all
```

다음 명령은 사용자가 명시적 시작 권한을 가지고 있는 AMI를 나열합니다. 사용자가 보유한 AMI는 이 목록에서 제외됩니다.

```
$ aws ec2 describe-images --executable-users self
```

다음 명령은 Amazon 소유 AMI를 나열합니다. Amazon의 공개 AMI는 별칭을 소유주 이름으로 사용하며 계정 필드에 amazon가 표시됩니다. 따라서 Amazon에서 배포한 AMI를 쉽게 찾을 수 있습니다. 다른 사용자는 AMI 별칭을 사용할 수 없습니다.

```
$ aws ec2 describe-images --owners amazon
```

다음 명령은 지정된 AWS 계정에서 소유한 AMI를 나열합니다.

```
$ aws ec2 describe-images --owners 123456789012
```

표시된 AMI 수가 너무 많다면 필터를 사용하여 원하는 유형의 AMI만 나타나도록 할 수 있습니다. 예를 들어, 다음 필터를 사용하면 EBS 기반 AMI만 나열됩니다.

```
--filters "Name=root-device-type,Values=ebs"
```

또는 Windows PowerShell용 AWS 도구 명령인 [Get-EC2Image](#)를 사용할 수 있습니다.

공유 AMI 사용

공유 AMI를 사용하기 전에 다음 과정에 따라 다음 과정을 따라 외부 사용자의 인스턴스 액세스를 허용하는 자격 증명 프로그램이나 민감한 정보를 외부로 전송할 수 있는 원격 로그인 설정이 포함된 AMI인지 확인해야 합니다. 시스템 보안성 향상에 대한 정보는 AMI에서 사용하는 Linux 배포 제품용 문서를 참조하십시오.

인스턴스에 대한 액세스가 끊기는 사고를 방지하려면 두 개의 SSH 세션을 시작해 한 세션에서 출처가 확실하지 않은 자격 증명 프로그램을 제거하고 SSH를 사용해 인스턴스에 로그인을 시도해 보고, 문제없음이 확인될 때까지 다른 세션을 오픈된 상태로 유지하는 것을 권장합니다.

1. 허용되지 않은 퍼블릭 SSH 키를 확인하고 비활성화합니다. AMI를 시작할 때는 파일에 포함된 키만 사용해야 합니다. `authorized_keys` 파일을 찾으려면 다음 명령을 사용합니다.

```
$ sudo find / -name "authorized_keys" -print -exec cat {} \;
```

2. 루트 사용자의 암호 방식 인증을 비활성화합니다. `ssh_config` 파일을 열고 `PermitRootLogin` 열을 다음과 같이 변경합니다.

```
PermitRootLogin without-password
```

또는 인스턴스에 루트 사용자로 로그인하여 기능을 비활성화할 수 있습니다.

```
PermitRootLogin No
```

`sshd` 서비스를 재시작합니다.

3. 인스턴스에 로그인할 수 있는 다른 사용자 계정이 있는지를 확인합니다. 이 때 `Supersuer` 권한을 가진 계정에 특히 유의해야 합니다. 알 수 없는 계정은 모두 암호를 제거하거나 잠금 설정합니다.
4. 개방 포트 중 사용하지 않는 포트 및 들어오는 연결을 수신하는 네트워크 서비스가 실행 중이지 않은 포트를 확인합니다.
5. 사전 구성을 통한 원격 로그인을 방지하려면 기존의 구성 파일을 삭제하고 `rsyslog` 서비스를 재시작해야 합니다. 예:

```
$ sudo rm /etc/  
rsyslog.config  
$ sudo service rsyslog restart
```

6. 모든 cron 작업의 유효성을 확인합니다.

보안을 위협하는 것으로 생각되는 퍼블릭 AMI를 발견했다면 AWS 보안 팀에 연락하십시오. 자세한 정보는 [AWS 보안 센터](#) 섹션을 참조하십시오.

퍼블릭 AMI 설정

Amazon EC2에서는 소유한 AMI를 다른 AWS 계정과 공유할 수 있습니다. 모든 AWS 계정에서 공유한 AMI를 시작할 수 있도록 설정하거나(퍼블릭 AMI), 특정 계정에서만 AMI를 시작할 수 있도록 설정할 수 있습니다 ([지정한 AWS 계정과 AMI 공유 \(p. 72\)](#) 참조). 다른 AWS 계정에서 공유 AMI를 사용하면 관련 요금은 공유 AMI를 시작한 해당 계정에만 청구됩니다.

AMI는 리전 리소스입니다. 따라서 AMI를 공유하면 해당 리전에서 사용할 수 있습니다. AMI를 다른 리전에서 사용할 수 있도록 하려면 AMI를 해당 리전에 복사한 후 공유하십시오. 자세한 내용은 [AMI 복사 \(p. 126\)](#) 섹션을 참조하십시오.

AMI를 공유할 때 민감한 데이터의 유출을 방지하려면 [공유 Linux AMI 지침 \(p. 74\)](#)에 설명된 보안상 고려 사항을 확인하여 권장 조치를 따르십시오.

Note

제품 코드가 포함된 AMI는 퍼블릭 설정이 불가능합니다. 이런 AMI는 지정된 AWS 계정하고만 공유 할 수 있습니다.

모든 AWS 계정과 AMI 공유(콘솔)

AMI를 공개한 후 콘솔을 사용하여 동일한 리전에서 인스턴스를 시작하면 [Community AMIs]에서 AMI를 사용할 수 있습니다. AMI 공개 후 AMI가 [Community AMIs]에 표시되는 데 약간의 시간이 걸릴 수 있다는 점에 유의하십시오. AMI를 다시 비공개로 바꾼 후 AMI가 [Community AMIs]에서 제거되는 데도 약간의 시간이 걸릴 수 있습니다.

콘솔을 사용해 퍼블릭 AMI를 공유하는 방법

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [AMIs]를 선택합니다.
3. 목록에서 AMI를 선택한 후 [Actions]에서 [Modify Image Permissions]를 선택합니다.
4. [Public]을 선택한 다음 [Save]를 선택합니다.

모든 AWS 계정과 AMI 공유(명령줄)

각 AMI에는 소유자를 제외하고 해당 AMI를 사용한 인스턴스 시작이 허용된 AWS 계정을 설정할 수 있는 `launchPermission` 속성이 존재합니다. AMI의 `launchPermission` 속성을 변경하여 이 AMI를 퍼블릭 설정(모든 AWS 계정에 시작 권한 허용)하거나 사용자가 지정한 AWS 계정하고만 공유할 수 있습니다.

AMI의 시작 권한을 부여할 계정 ID는 목록에 추가하거나 제거할 수 있습니다. AMI를 퍼블릭 설정하려면 all 그룹을 지정합니다. 퍼블릭 권한과 명시적 시작 권한 모두 설정이 가능합니다.

퍼블릭 AMI 설정

다음과 같이 [modify-image-attribute](#) 명령(AWS CLI)을 실행하고 지정한 AMI의 launchPermission 목록에 all 그룹을 추가합니다.

```
$ aws ec2 modify-image-attribute --image-id ami-12345678 --launch-permission "{\"Add\":[{\"Group\":\"all\"}]}"
```

AMI의 시작 권한을 확인하려면 [describe-image-attribute](#) 명령을 사용합니다.

```
$ aws ec2 describe-image-attribute --image-id ami-12345678 --attribute launchPermission
```

(선택 사항) AMI를 프라이빗 상태로 되돌리려면 시작 권한 목록에서 all 그룹을 삭제합니다. AMI의 소유자는 언제나 시작 권한을 가지며 이 명령에 영향을 받지 않습니다.

```
$ aws ec2 modify-image-attribute --image-id ami-12345678 --launch-permission "{\"Remove\":[{\"Group\":\"all\"}]}"
```

또는 다음 Windows PowerShell용 AWS 도구 [Edit-EC2ImageAttribute](#) 및 [Get-EC2ImageAttribute](#) 명령을 사용할 수 있습니다.

지정한 AWS 계정과 AMI 공유

AMI를 퍼블릭으로 설정하지 않고 지정한 AWS 계정과 공유할 수 있습니다. 이런 작업은 AWS 계정 ID만 있으면 가능합니다.

AMI는 리전 리소스입니다. 따라서 AMI를 공유하면 해당 리전에서 사용할 수 있습니다. AMI를 다른 리전에서 사용할 수 있도록 하려면 AMI를 해당 리전에 복사한 후 공유하십시오. 자세한 내용은 [AMI 복사 \(p. 126\)](#) 섹션을 참조하십시오.

AMI를 공유하는 방법(콘솔)

콘솔을 사용해 명시적 시작 권한을 허용하는 방법

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [AMIs]를 선택합니다.
3. 목록에서 AMI를 선택한 후 [Actions]에서 [Modify Image Permissions]를 선택합니다.
4. [AWS Account Number] 필드에 AMI를 공유할 AWS 계정 번호를 지정하고 [Add Permission]을 선택합니다.

이 AMI를 다수의 사용자와 공유하려면 위 단계를 반복하여 사용자를 추가합니다.

5. 스냅샷 볼륨 권한 생성을 허용하려면 [Add "create volume" permissions to the following associated snapshots when creating permissions.]를 선택합니다.

Note

AMI를 공유하기 위해서 해당 AMI의 레퍼런스인 Amazon EBS 스냅샷을 함께 공유할 필요는 없습니다. AMI만 공유하면 시스템에서 시작에 필요한 Amazon EBS 스냅샷 액세스를 인스턴스에 자동으로 제공합니다.

6. 마치면 [Save]를 선택합니다.

AMI(명령줄) 공유

modify-image-attribute 명령(AWS CLI)을 사용하여 다음 예시와 같이 AMI를 공유할 수 있습니다.

명시적 시작 권한 허용

다음 명령은 지정한 AWS 계정에 특정 AMI의 시작 권한을 허용하는 데 사용됩니다.

```
$ aws ec2 modify-image-attribute --image-id ami-12345678 --launch-permission "{\"Add\": [{\"UserId\":\"123456789012\"}]}"
```

다음 명령은 스냅샷 볼륨 권한 생성을 허용합니다.

```
$ aws ec2 modify-snapshot-attribute --snapshot-id snap-1234567890abcdef0 --attribute createVolumePermission --operation-type add --user-ids 123456789012
```

특정 계정에서 시작 권한을 제거하는 방법

다음 명령은 지정한 AWS 계정에 허용했던 특정 AMI의 시작 권한을 제거하는 데 사용됩니다.

```
$ aws ec2 modify-image-attribute --image-id ami-12345678 --launch-permission "{\"Remove\": [{\"UserId\":\"123456789012\"}]}"
```

다음 명령은 스냅샷 볼륨 권한 생성을 제거합니다.

```
$ aws ec2 modify-snapshot-attribute --snapshot-id snap-1234567890abcdef0 --attribute createVolumePermission --operation-type remove --user-ids 123456789012
```

모든 시작 권한을 제거하는 방법

다음 명령은 특정 AMI의 퍼블릭 및 명시적 시작 권한을 모두 삭제하는 데 사용됩니다. AMI의 소유자는 언제나 시작 권한을 가지며 이 명령에 영향을 받지 않습니다.

```
$ aws ec2 reset-image-attribute --image-id ami-12345678 --attribute launchPermission
```

또는 Windows PowerShell용 AWS 도구 명령인 [Edit-EC2ImageAttribute](#)를 사용할 수 있습니다.

북마크 사용

퍼블릭 AMI를 생성했거나 다른 AWS 사용자와 AMI를 공유했다면 허용된 사용자가 자신의 계정에서 즉시 인스턴스를 시작할 수 있도록 허용하는 북마크를 생성할 수 있습니다. 사용을 위해 AMI 검색에 시간을 할애할 필요 없이 AMI 레퍼런스를 공유하는 간단한 방법입니다.

이 때 AMI는 반드시 퍼블릭 AMI이거나 북마크를 보낼 사용자와 공유된 상태여야 합니다.

AMI 북마크 생성

1. 다음 정보를 참고하여 URL를 입력합니다. 여기에서 <region>은 AMI가 속하는 리전, 그리고 <ami_id>는 AMI의 ID입니다.

```
https://console.aws.amazon.com/ec2/v2/home?  
region=<region>#LaunchInstanceWizard:ami=<ami_id>
```

예를 들어, 다음 URL은 us-east-1 리전의 ami-12345678 AMI에서 인스턴스를 실행합니다.

```
https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard:ami=ami-12345678
```

2. AMI 사용을 원하는 사용자에게 이 링크를 공유합니다.
3. 북마크를 사용하려면 링크를 선택하거나 복사하여 브라우저에 붙여넣기하면 됩니다. AMI 선택이 완료된 상태로 시작 마법사가 열립니다.

공유 Linux AMI 지침

AMI의 안정성을 높이고 공격 대상 영역을 최소화하려면 다음 지침을 사용하십시오.

Note

어떤 보안 지침도 포괄적일 수는 없습니다. 공유 AMI를 구축할 때는 민감한 데이터의 유출 가능성에 특히 유의하고, 충분한 시간을 할애하여 검토하십시오.

항목

- [부팅 시 AMI 도구 업데이트 \(p. 74\)](#)
- [루트 사용자의 암호 방식 원격 로그인 비활성화 \(p. 75\)](#)
- [로컬 루트 액세스 비활성화 \(p. 75\)](#)
- [SSH 호스트 키 페어 삭제 \(p. 75\)](#)
- [퍼블릭 키 자격 증명 프로그램 설치 \(p. 76\)](#)
- [sshd DNS 확인 비활성화\(선택 사항\) \(p. 76\)](#)
- [본인 인증 \(p. 77\)](#)
- [보안 \(p. 77\)](#)

AWS Marketplace용 AMI를 구축하는 경우에는 [Building AMIs for AWS Marketplace](#)의 지침과 정책, 모범 사례를 참조하십시오.

안전한 AMI 공유에 대한 추가 내용은 다음을 참조하십시오.

- [안전한 방식으로 퍼블릭 AMI를 공유하고 사용하는 방법](#)
- [Public AMI Publishing: Hardening and Clean-up Requirements](#)

부팅 시 AMI 도구 업데이트

인스턴스 스토어 지원 AMI의 경우, AMI 부팅 시 Amazon EC2 AMI 생성 도구를 다운로드받고 업그레이드하는 것을 권장합니다. 이를 통해 공유 AMI를 기반으로 한 새 AMI에서 최신 AMI 도구를 사용할 수 있습니다.

Amazon Linux는 다음을 /etc/rc.local에 추가합니다.

```
# Update the Amazon EC2 AMI tools
echo " + Updating EC2 AMI tools"
yum update -y aws-amitools-ec2
echo " + Updated EC2 AMI tools"
```

이 방법을 사용하여 이미지이 다른 소프트웨어를 자동으로 업데이트합니다.

Note

자동 업데이트할 소프트웨어를 결정할 때는 업데이트로 인해 고객이 비용을 부담하게 되는 WAN 트래픽 수준과 업데이트로 AMI의 다른 소프트웨어 오류가 발생할 위험성을 고려해야 합니다.

기타 배포의 경우에는 AMI 도구를 항상 최신으로 유지합니다.

루트 사용자의 암호 방식 원격 로그인 비활성화

퍼블릭 AMI에 대하여 고정 루트 암호를 사용하면 보안 위험을 축소하는 계기가 될 수 있습니다. 고객에게 최초 로그인 시 암호 변경을 알린다 해도 여기에만 의존한다면 어느 정도의 오용 가능성은 여전히 존재합니다.

이런 문제를 해결하려면 루트 사용자의 암호 방식 원격 로그인을 비활성화합니다.

루트 사용자의 암호 방식 원격 로그인 비활성화

1. 텍스트 편집기로 /etc/ssh/sshd_config 파일을 열고 다음 열을 검색합니다.

```
#PermitRootLogin yes
```

2. 해당 열을 다음과 같이 변경합니다.

```
PermitRootLogin without-password
```

이 구성 파일의 저장 위치는 배포에 따라서 혹은 OpenSSH를 실행하지 않는 경우 달라질 수 있습니다.
이 경우에는 관련 문서를 참조하십시오.

로컬 루트 액세스 비활성화

공유 AMI를 사용할 때는 직접 루트 로그인을 비활성화하는 것이 모범 사례입니다. 이렇게 하려면 실행 중인 인스턴스에 로그인하여 다음 명령을 사용합니다.

```
[ec2-user ~]$ sudo passwd -l root
```

Note

이 명령은 sudo 사용에는 영향을 주지 않습니다.

SSH 호스트 키 페어 삭제

퍼블릭 AMI에서 유래된 AMI를 공유할 계획이라면 /etc/ssh에 저장된 현재 SSH 호스트 키 페어를 삭제하십시오. 이 작업은 다른 사용자가 이 AMI를 사용해 인스턴스를 시작할 때 SSH에서 반드시 새로운 고유 SSH 키 페어를 생성하도록 하기 때문에 "중간자 공격" 가능성을 낮추고 보안을 향상시켜 줍니다.

시스템에서 다음의 키 파일을 모두 제거합니다.

- ssh_host_dsa_key
- ssh_host_dsa_key.pub
- ssh_host_key
- ssh_host_key.pub
- ssh_host_rsa_key
- ssh_host_rsa_key.pub
- ssh_host_ecdsa_key
- ssh_host_ecdsa_key.pub
- ssh_host_ed25519_key
- ssh_host_ed25519_key.pub

이런 파일은

```
[ec2-user ~]$ sudo shred -u /etc/ssh/*_key /etc/ssh/*_key.pub
```

명령을 실행하여 안전하게 제거할 수 있습니다.

Warning

shred와 같은 보안 삭제 유틸리티는 스토리지 미디어에서 파일의 사본을 모두 제거하지 못할 수 있습니다. 파일 시스템(Amazon Linux default ext4 포함), 스냅샷, 백업, RAID 및 임시 캐싱을 저널링하여 파일의 숨겨진 사본이 생성될 수 있습니다. 자세한 내용은 [shred 문서](#)를 참조하십시오.

Important

퍼블릭 AMI의 현재 SSH 호스트 키 페어를 제거하지 않은 경우, 기본적인 자체 감사 과정에서 소유자를 비롯해 해당 AMI를 사용해 인스턴스를 실행하는 모든 사용자에게 잠재적인 보안 위험을 알리는 메시지가 표시됩니다. 이 AMI는 단기적인 유예 기간 후 프라이빗 상태로 변경됩니다.

퍼블릭 키 자격 증명 프로그램 설치

암호를 사용한 AMI 로그인을 비활성화했다면 이제 다른 방식으로 사용자가 로그인할 수 있도록 해야 합니다.

Amazon EC2에서는 인스턴스를 시작할 때 사용자가 퍼블릭/프라이빗 키 페어 이름을 설정하는 것을 허용합니다. 유효한 키 페어 이름이 `RunInstances` API 호출(또는 명령줄 API 도구)로 전송되면, 퍼블릭 키(Amazon EC2에서 `CreateKeyPair` 또는 `ImportKeyPair`로의 호출이 이루어진 후에 서버에 저장하는 키 페어의 일부)를 인스턴스 메타데이터에 대한 HTTP 쿼리를 통해 인스턴스에서 사용할 수 있게 됩니다.

SSH를 통해 로그인하려면 AMI에서 반드시 부팅 시 키 값을 회수하고 이 값을 `/root/.ssh/authorized_keys`에 (또는 AMI 상의 다른 사용자 계정의 값) 첨부해야 합니다. 사용자는 루트 암호 없이 키 페어를 사용하여 AMI의 인스턴스를 실행할 수 있습니다.

Amazon Linux 및 Ubuntu를 포함한 대부분의 배포판에서는 지정된 사용자에 대한 퍼블릭 키 자격 증명을 첨가할 때 `cloud-init` 패키지를 사용합니다. 사용하는 배포판에서 `cloud-init`를 지원하지 않는 경우, 시스템 시작 스크립트(예: `/etc/rc.local`)에 다음 코드를 추가하여 `root` 사용자가 시작할 때 지정한 퍼블릭 키를 가져오도록 설정할 수 있습니다.

```
if [ ! -d /root/.ssh ] ; then
    mkdir -p /root/.ssh
    chmod 700 /root/.ssh
fi
# Fetch public key using HTTP
curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key > /tmp/my-key
if [ $? -eq 0 ] ; then
    cat /tmp/my-key >> /root/.ssh/authorized_keys
    chmod 700 /root/.ssh/authorized_keys
    rm /tmp/my-key
fi
```

이 작업은 어떤 사용자 계정에나 적용할 수 있으며, `root`로 제한할 필요가 없습니다.

Note

이 AMI를 기반으로 인스턴스를 다시 번들링하면 시작했을 때 사용했던 키가 포함됩니다. 키가 포함되는 것을 방지하려면 `authorized_keys` 파일의 내용을 지우거나 파일을 삭제하는 방법, 또는 재번들링 시 파일을 포함 제외해야 합니다.

sshd DNS 확인 비활성화(선택 사항)

sshd DNS 확인을 비활성화하면 sshd 보안성은 약간 저하됩니다. 하지만 DNS 확인이 실패했을 때에도 SSH 로그인이 가능하게 해 줍니다. sshd 확인을 비활성화하면 DNS 확인 오류 시 모든 로그인이 금지됩니다.

sshd DNS 확인 비활성화

1. 텍스트 편집기로 `/etc/ssh/sshd_config` 파일을 열고 다음 열을 검색합니다.

```
#UseDNS yes
```

2. 해당 열을 다음과 같이 변경합니다.

```
UseDNS no
```

Note

이 구성 파일의 저장 위치는 배포에 따라서 혹은 OpenSSH를 실행하지 않는 경우 달라질 수 있습니다. 이 경우에는 관련 문서를 참조하십시오.

본인 인증

AMI는 계정 ID로 표시되기 때문에 공유 AMI 제공자를 간편하게 확인할 수 있는 방법은 현재 존재하지 않습니다.

Amazon에서는 [Amazon EC2 forum](#)에 AMI 설명과 AMI ID를 게시하는 것을 권장합니다. 새로운 공유 AMI에 흥미를 가지고 시도하려는 사용자에게 유용한 중심 자료소를 제공하기 때문입니다. 또한 [Amazon Machine Images \(AMIs\)](#) 페이지에 AMI를 게시할 수도 있습니다.

보안

먼저 섹션에서는 안전하고 보안된 공유 AMI를 만들고 사용자 시작 권한을 설정하는 방법에 대해 알아보았습니다. 이 섹션에서는 공유한 AMI를 사용하는 다른 사용자로부터 보안을 유지하는 방법을 안내합니다.

공유하는 AMI에는 민감한 데이터나 소프트웨어를 포함하지 않는 것이 권장됩니다. 공유 AMI를 시작하는 사용자가 이런 AMI를 재번들링하여 본인 소유로 등록할 수 있기 때문입니다. 다음 지침에 따라 그냥 지나치기 쉬운 보안 위험에 대처하십시오.

- `ec2-bundle-vol`에서 `--exclude directory` 옵션을 사용해 번들에 포함하지 않아야 할 보안 정보가 담긴 디렉터리나 하위 디렉터리를 선택하지 않는 방법을 권장합니다. 특히, 이미지를 번들링할 때 모든 사용자 소유 SSH 퍼블릭/프라이빗 키 페어와 SSH `authorized_keys` 파일을 제외하십시오. Amazon 퍼블릭 AMI는 이를 `root` 계정의 경우 `/root/.ssh`에, 일반 사용자 계정의 경우 `/home/user_name/.ssh`에 저장합니다. 자세한 내용은 [ec2-bundle-vol \(p. 91\)](#) 섹션을 참조하십시오.
- 번들링 전에는 항상 셸 기록을 삭제합니다. 동일한 AMI로 하나 이상의 번들을 업로드하려고 시도하면 셸 기록에 보안 액세스 키가 포함됩니다. 다음 명령은 인스턴스 내에서 번들링을 실시하기 전 마지막 단계로 실행해야 합니다.

```
[ec2-user ~]$ shred -u ~/.history
```

Warning

위 경고에서 설명한 `shred`의 제한은 여기에도 적용됩니다.

`bash`는 종료 시점에서 현재 세션의 이력을 디스크에 기록한다는 점을 유의하십시오. `~/.bash_history`를 삭제한 후 인스턴스에서 로그아웃했다가 다시 로그인할 경우 `~/.bash_history`가 다시 생성되고 이전 세션에서 실행한 모든 명령이 포함되어 있는 것을 알 수 있습니다.

`bash` 이외의 다른 프로그램도 디스크에 이력을 기록하므로 불필요한 DOT 파일 및 DOT 디렉터리를 삭제 또는 제외하도록 주의하십시오.

- 실행 중인 인스턴스를 번들링하려면 프라이빗 키와 X.509 인증서가 요구됩니다. 이런 정보와 다른 자격 증명은 번들링에 포함되지 않은 장소(예: 인스턴스 스토어)에 따로 보관하십시오.

유료 AMI

유료 AMI는 개발자에게서 구입할 수 있는 AMI입니다.

Amazon EC2가 AWS Marketplace와 통합되어 개발자가 다른 Amazon EC2 사용자에게 AMI 사용 요금을 청구하거나 인스턴스에 대한 지원을 제공할 수 있습니다.

AWS Marketplace는 EC2 인스턴스를 시작하는 데 사용할 수 있는 AMI를 비롯하여 AWS에서 실행되는 소프트웨어를 구입할 수 있는 온라인 상점입니다. 요구 사항에 맞는 제품을 찾을 수 있도록 AWS Marketplace AMI는 범주(예: Developer Tools)별로 구성됩니다. AWS Marketplace에 대한 자세한 내용은 [AWS Marketplace](#) 사이트를 참조하십시오.

유료 AMI에서 인스턴스를 시작하는 것은 다른 AMI에서 인스턴스를 시작하는 것과 같습니다. 추가 파라미터가 필요하지 않습니다. AMI 소유자가 설정한 요금과 관련 웹 서비스에 대한 스탠다드 사용 요금(예: Amazon EC2에서 m1.small 인스턴스 유형 실행에 대한 시간당 요금)에 따라 인스턴스 요금이 부과됩니다. 추가 세금이 적용될 수도 있습니다. 유료 AMI의 소유자는 특정 인스턴스가 해당 유료 AMI를 사용하여 시작되었는지 여부를 확인할 수 있습니다.

Important

Amazon DevPay는 더 이상 새로운 판매자 또는 제품을 수락하지 않습니다. 이제 AWS Marketplace가 AWS를 통해 소프트웨어와 서비스를 판매하는 단일 통합 전자 상거래 플랫폼입니다. AWS Marketplace에서 소프트웨어를 배포하고 판매하는 방법에 대한 자세한 내용은 [Selling on AWS Marketplace](#) 섹션을 참조하십시오. AWS Marketplace는 Amazon EBS 기반 AMI를 지원합니다.

항목

- [AMI 판매](#) (p. 78)
- [유료 AMI 찾기](#) (p. 78)
- [유료 AMI 구입](#) (p. 79)
- [인스턴스에 대한 제품 코드 가져오기](#) (p. 80)
- [유료 지원 사용](#) (p. 80)
- [유료 및 지원된 AMI에 대한 청구서](#) (p. 80)
- [AWS Marketplace 구독 관리](#) (p. 80)

AMI 판매

AWS Marketplace를 사용하여 AMI를 판매할 수 있습니다. AWS Marketplace는 조직적인 쇼핑 환경을 제공합니다. 또한 AWS Marketplace는 Amazon EBS 기반 AMI, 예약 인스턴스, 스팟 인스턴스 등의 AWS 기능을 기반합니다.

AWS Marketplace에서 AMI를 판매하는 방법에 대한 자세한 내용은 [Selling on AWS Marketplace](#) 섹션을 참조하십시오.

유료 AMI 찾기

구입 가능한 AMI를 찾는 방법은 다양합니다. 예를 들어, [AWS Marketplace](#), Amazon EC2 콘솔 또는 명령줄을 사용할 수 있습니다. 또는 개발자가 유료 AMI에 대한 정보를 제공할 수 있습니다.

콘솔을 사용하여 유료 AMI 찾기

콘솔을 사용하여 유료 AMI를 찾으려면

1. Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [AMIs]를 클릭합니다.

3. 첫 번째 [Filter] 목록에서 [Public images]를 선택합니다. 검색 창을 클릭하고 [Product Code], [Marketplace]를 차례로 선택합니다. 검색 창을 다시 클릭하고 [Platform]을 선택한 다음 목록에서 운영 체제를 선택합니다.

AWS Marketplace를 사용하여 유료 AMI 찾기

AWS Marketplace를 사용하여 유료 AMI를 찾으려면

1. [AWS Marketplace](#)를 엽니다.
2. 검색 상자에 운영 체제의 이름을 입력하고 [Go]를 클릭합니다.
3. 결과 범위를 더 자세히 지정하려면 범주 또는 필터 중 하나를 사용합니다.
4. 각 제품에는 제품 유형(AMI 또는 Software as a Service)으로 레이블로 지정됩니다.

명령줄을 사용하여 유료 AMI 찾기

다음과 같이 `describe-images` 명령(AWS CLI)을 사용하여 유료 AMI를 찾을 수 있습니다.

```
$ aws ec2 describe-images --owners aws-marketplace
```

이 명령은 유료 AMI에 대한 제품 코드를 포함하여 각 AMI를 설명하는 다양한 정보를 반환합니다. `describe-images`의 출력에는 다음과 같은 제품 코드 항목이 포함됩니다.

```
"ProductCodes": [  
    {  
        "ProductCodeId": "product_code",  
        "ProductCodeType": "marketplace"  
    }  
,
```

또는 Windows PowerShell용 AWS 도구 명령인 [Get-EC2Image](#)를 사용할 수 있습니다.

유료 AMI 구입

AMI를 사용하여 인스턴스를 시작하려면 유료 AMI(구입)에 가입해야 합니다.

대개 유료 AMI 소유자가 가격과 해당 AMI를 구입할 수 있는 링크를 비롯하여 AMI에 대한 정보를 제공합니다. 링크를 클릭하면 AWS에 로그인하라는 메시지가 표시되고 그런 다음 AMI를 구입할 수 있습니다.

콘솔을 사용하여 유료 AMI 구입

Amazon EC2 시작 마법사를 사용하여 유료 AMI를 구입할 수 있습니다. 자세한 내용은 [AWS Marketplace 인스턴스 시작 \(p. 271\)](#) 섹션을 참조하십시오.

AWS Marketplace를 사용하여 제품 구독

AWS Marketplace를 사용하려면 AWS 계정이 있어야 합니다. AWS Marketplace 제품에서 인스턴스를 시작하려면 Amazon EC2 서비스를 사용하도록 가입되어 있고, 인스턴스를 시작할 제품을 구독해야 합니다. AWS Marketplace에서 제품을 구독하는 방법은 두 가지입니다.

- AWS Marketplace 웹 사이트: 1-Click 배포 기능을 사용하여 미리 구성된 소프트웨어를 빠르게 시작할 수 있습니다.
- Amazon EC2 시작 마법사: AMI를 검색하고 마법사에서 직접 인스턴스를 시작할 수 있습니다. 자세한 내용은 [AWS Marketplace 인스턴스 시작 \(p. 271\)](#) 섹션을 참조하십시오.

개발자에게서 유료 AMI 구입

유료 AMI의 개발자를 통해 AWS Marketplace에 나열되지 않은 유료 AMI를 구입할 수 있습니다. 개발자는 Amazon을 통해 제품을 구입할 수 있는 링크를 제공합니다. Amazon.com 자격 증명을 사용하여 로그인하고 AMI 구입에 사용할 Amazon.com 계정에 저장된 신용 카드를 선택할 수 있습니다.

인스턴스에 대한 제품 코드 가져오기

인스턴스 메타데이터를 사용하여 인스턴스에 대한 AWS Marketplace 제품 코드를 검색할 수 있습니다. 메타데이터 검색에 대한 자세한 내용은 [인스턴스 메타데이터 및 사용자 데이터 \(p. 321\)](#) 섹션을 참조하십시오.

제품 코드를 검색하려면 다음 쿼리를 사용합니다.

```
$ GET http://169.254.169.254/latest/meta-data/product-codes
```

인스턴스에 제품 코드가 있는 경우 Amazon EC2에서 해당 코드를 반환합니다. 예:

```
774F4FFF8
```

유료 지원 사용

개발자가 Amazon EC2를 사용하여 소프트웨어 또는 파생 AMI를 지원할 수도 있습니다. 개발자는 사용자가 가입하여 사용할 수 있는 지원 제품을 생성할 수 있습니다. 지원 제품에 가입하는 동안 개발자가 제품 코드를 제공합니다. 이 제품 코드를 AMI와 연결해야 합니다. 개발자는 이 제품 코드를 사용하여 인스턴스가 지원 대상인지 확인할 수 있습니다. 또한 제품의 인스턴스를 실행할 때 개발자가 지정한 제품에 대한 조건에 따라 요금이 부과됩니다.

Important

지원 제품을 예약 인스턴스와 함께 사용할 수 없습니다. 항상 지원 제품의 판매자가 지정한 가격을 지불합니다.

제품 코드를 AMI와 연결하려면 다음 명령 중 하나를 사용합니다. 여기에서 ami_id는 AMI의 ID이고 product_code는 제품 코드입니다.

- [modify-image-attribute\(AWS CLI\)](#)

```
$ aws ec2 modify-image-attribute --image-id ami_id --product-codes "product_code"
```

- [Edit-EC2ImageAttribute\(Windows PowerShell용 AWS 도구\)](#)

```
C:\> Edit-EC2ImageAttribute -ImageId ami_id -ProductCode product_code
```

제품 코드 속성을 설정한 후 해당 속성을 변경하거나 제거할 수 없습니다.

유료 및 지원된 AMI에 대한 청구서

매월 말 그 달에 사용한 유료 또는 지원된 AMI에 대해 신용 카드로 청구되는 금액을 이메일로 수신하게 됩니다. 이 청구서는 정기 Amazon EC2 청구서와는 별개입니다. 자세한 내용은 [Paying For AWS Marketplace Products](#) 섹션을 참조하십시오.

AWS Marketplace 구독 관리

AWS Marketplace 웹 사이트에서 구독 정보 확인, 공급업체의 사용 지침 보기, 구독 관리 등을 수행할 수 있습니다.

구독 정보를 확인하려면

1. [AWS Marketplace](#)에 로그인합니다.
2. [Your Account]를 클릭합니다.
3. [Manage Your Software Subscriptions]를 클릭합니다.
4. 현재 구독이 모두 나열됩니다. [Usage Instructions]를 클릭하여 제품 사용에 대한 특정 지침(예: 실행 중인 인스턴스에 연결하기 위한 사용자 이름)을 확인합니다.

AWS Marketplace 구독을 취소하려면

1. 구독에서 실행 중인 모든 인스턴스를 종료해야 합니다.
 - a. Amazon EC2 콘솔을 엽니다.
 - b. 탐색 창에서 Instances를 클릭합니다.
 - c. 인스턴스를 선택하고 [Actions]를 클릭한 다음 [Instance State], [Terminate]를 차례로 선택합니다. 메시지가 나타나면 [Yes, Terminate]를 클릭합니다.
2. [AWS Marketplace](#)에 로그인하고 [Your Account], [Manage Your Software Subscriptions]를 차례로 클릭합니다.
3. [Cancel subscription]을 클릭합니다. 취소를 확인하라는 메시지가 나타납니다.

Note

구독을 취소하면 해당 AMI에서 더 이상 인스턴스를 시작할 수 없습니다. AMI를 다시 사용하려면 AWS Marketplace 웹 사이트 또는 Amazon EC2 콘솔의 시작 마법사를 통해 해당 AMI를 다시 구독해야 합니다.

Amazon EBS 지원 Linux AMI 생성

Amazon EBS 지원 Linux AMI를 생성하려면 기존 Amazon EBS 지원 Linux AMI에서 시작한 인스턴스에서 시작합니다. 이 AMI는 AWS Marketplace에서 구입한 AMI, [AWS Server Migration Service](#) 또는 [VM Import/Export](#)를 사용하여 생성한 AMI 또는 액세스 권한이 있는 기타 AMI 등이 될 수 있습니다. 필요에 맞게 인스턴스를 사용자 지정한 후에는 이러한 사용자 지정을 적용하여 새 인스턴스를 시작하는 데 사용할 수 있는 새 AMI를 생성하여 등록합니다. Amazon EBS 지원 Windows AMI 생성에 대한 자세한 내용은 Windows 인스턴스용 Amazon EC2 사용 설명서의 [Amazon EBS 지원 Windows AMI 생성](#)을 참조하십시오.

아래에 설명된 절차는 암호화된 Amazon EBS 볼륨(루트 볼륨 포함) 및 암호화되지 않은 볼륨에서 지원되는 Amazon EC2 인스턴스에 작동합니다.

인스턴스 스토어 지원 AMI에 대한 AMI 생성 프로세스는 다릅니다. Amazon EBS 지원 인스턴스와 인스턴스 스토어 지원 인스턴스 간의 차이점 및 인스턴스의 루트 디바이스 유형을 확인하는 방법에 대한 자세한 내용은 [루트 디바이스 스토리지 \(p. 64\)](#)를 참조하십시오. 인스턴스 스토어 지원 Linux AMI 생성에 대한 자세한 내용은 [인스턴스 스토어 기반 Linux AMI 생성 \(p. 84\)](#)를 참조하십시오.

Amazon EBS 지원 AMI 생성 개요

우선 만들려는 AMI와 비슷한 AMI에서 인스턴스를 시작합니다. 인스턴스에 연결하여 인스턴스를 사용자 지정할 수 있습니다. 인스턴스가 올바르게 구성되면 AMI를 생성하기 전에 인스턴스를 중단하여 데이터 무결성을 확인한 다음 이미지를 생성합니다. Amazon EBS 기반 AMI를 생성하면 자동으로 등록됩니다.

Amazon EC2는 인스턴스의 모든 기능을 중지하여 생성 프로세스 중 일관된 상태를 유지하기 위해 AMI를 생성하기 전에 인스턴스의 전원을 차단합니다. 인스턴스가 AMI 생성에 적합한 일관된 상태를 유지하는 경우 전원을 차단하지 않고 인스턴스를 재부팅하도록 Amazon EC2를 설정할 수 있습니다. 일부 파일 시스템(예: XFS)에서는 활동을 동결 및 동결 해제하여 인스턴스를 재부팅하지 않고 이미지를 안전하게 생성할 수 있습니다.

AMI 생성 프로세스 중에 Amazon EC2는 인스턴스의 루트 볼륨과 인스턴스에 연결된 다른 EBS 볼륨의 스냅샷을 생성합니다. 인스턴스에 연결된 볼륨이 암호화된 경우 새 AMI는 Amazon EBS 암호화를 지원하는 인스턴스에서만 시작됩니다. 자세한 내용은 [Amazon EBS Encryption \(p. 617\)](#) 섹션을 참조하십시오.

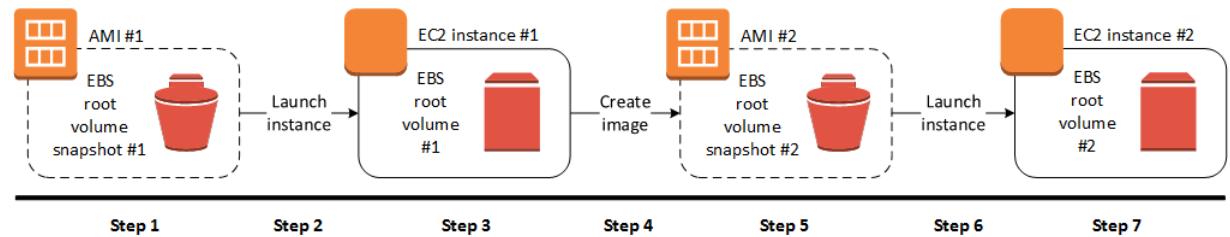
볼륨의 크기에 따라 AMI 생성 프로세스를 완료하는 데 몇 분 정도 걸리지만, 경우에 따라 24시간까지 걸릴 수도 있습니다. AMI를 생성하기 이전에 볼륨의 스냅샷을 만드는 것이 더 효율적일 수 있습니다. 이처럼 AMI를 생성할 때 작은 충분적 스냅샷만 만들어야 프로세스가 더 빠르게 완료됩니다. 스냅샷을 만드는 데 걸리는 전체 시간은 동일하게 유지됩니다. 자세한 내용은 [Amazon EBS 스냅샷 생성 \(p. 608\)](#) 섹션을 참조하십시오.

프로세스가 완료되면 인스턴스의 루트 볼륨에서 새 AMI 및 스냅샷이 생성됩니다. 새 AMI를 사용하여 인스턴스를 시작할 때 스냅샷을 사용하여 루트 볼륨에 대한 새 EBS 볼륨을 생성합니다. AMI와 스냅샷에 대해 사용자가 삭제할 때까지 사용자 계정에 요금이 부과됩니다. 자세한 내용은 [AMI 등록 취소 \(p. 130\)](#) 섹션을 참조하십시오.

루트 디바이스 볼륨 외에도 인스턴스 스토어 볼륨이나 EBS 볼륨을 인스턴스에 추가하는 경우, 새 AMI에 대한 블록 디바이스 매핑과 새 AMI에서 자동으로 시작하는 인스턴스에 대한 블록 디바이스 매핑에 이러한 볼륨에 대한 정보가 포함됩니다. 새 인스턴스에 대한 블록 디바이스 매핑에 지정된 인스턴스 스토어 볼륨은 새 볼륨이므로 AMI를 생성하는 데 사용된 인스턴스에 대한 인스턴스 스토어 볼륨의 데이터가 포함되어 있지 않습니다. EBS 볼륨의 데이터는 유지됩니다. 자세한 내용은 [블록 디바이스 매핑 \(p. 662\)](#) 섹션을 참조하십시오.

인스턴스에서 Linux AMI 생성

AWS Management Console 또는 명령줄을 사용하여 AMI를 생성할 수 있습니다. 다음 다이어그램은 실행 중인 EC2 인스턴스에서 Amazon EBS 지원 AMI를 만드는 프로세스를 요약한 것입니다. 기존 AMI로 시작하여 인스턴스를 시작한 다음 사용자 지정하고 해당 인스턴스에서 새 AMI를 생성합니다. 그런 다음 새 AMI의 인스턴스를 시작합니다. 다음 다이어그램의 단계는 아래 절차의 단계와 일치합니다.



콘솔을 사용하여 인스턴스에서 AMI를 생성하려면

- 새 AMI의 시작점으로 사용할 적절한 EBS 지원 AMI를 선택하고, 시작하기 전에 필요에 따라 구성합니다. 자세한 내용은 [인스턴스 시작하기 \(p. 265\)](#) 섹션을 참조하십시오.
- [Launch]를 선택하여 선택한 EBS 지원 AMI의 인스턴스를 시작합니다. 나머지 기본값을 그대로 두고 마법사를 계속 진행합니다. 자세한 내용은 [인스턴스 시작하기 \(p. 265\)](#) 섹션을 참조하십시오.
- 인스턴스가 실행 중일 때 인스턴스에 연결합니다.

인스턴스에서 다음과 같은 작업을 수행하여 인스턴스를 원하는 대로 사용자 지정할 수 있습니다.

- 소프트웨어 및 애플리케이션 설치
- 데이터 복사
- 임시 파일 삭제, 하드 드라이브 조각 모음, 여유 공간 제로 클리어를 통한 시작 속도 향상
- 추가 Amazon EBS 볼륨 연결

(선택 사항) 인스턴스에 연결한 모든 볼륨의 스냅샷을 생성합니다. 스냅샷 생성에 대한 자세한 내용은 [Amazon EBS 스냅샷 생성 \(p. 608\)](#) 섹션을 참조하십시오.

탐색 창에서 [Instances]를 선택하고 인스턴스를 선택합니다. [Actions], [Image] 및 [Create Image]를 선택합니다.

Tip

이 옵션이 비활성화되어 있다면 Amazon EBS 지원 인스턴스가 아님을 의미합니다.

- [Create Image] 대화 상자에서 다음 필드에 값을 지정한 다음 [Create Image]를 선택합니다.

이름

이미지의 고유한 이름입니다.

설명

(선택 사항) 이미지에 대한 설명이며 최대 255자까지 입력할 수 있습니다.

기본적으로 Amazon EC2는 인스턴스를 종료하고, 연결된 볼륨의 스냅샷을 캡처하고, AMI를 생성하여 등록한 다음 인스턴스를 재부팅합니다. 인스턴스를 종료하지 않으려는 경우 [No reboot]를 선택합니다.

Warning

[No reboot]를 선택하는 경우 생성된 이미지의 파일 시스템 무결성을 보장할 수 없습니다.

루트 볼륨, Amazon EBS 볼륨 및 인스턴스 스토어 볼륨을 다음과 같이 수정할 수 있습니다.

- 루트 볼륨 크기를 변경하려면 [Type 열의 [Root] 볼륨으로 이동한 다음 [Size] 필드에 입력합니다.
- 인스턴스를 시작하는 데 사용되는 AMI의 블록 디바이스 매핑에서 지정된 Amazon EBS 볼륨을 표시하지 않으려면 목록에서 EBS 볼륨으로 이동한 다음 [Delete]를 선택합니다.
- Amazon EBS 볼륨을 추가하려면 [Add New Volume], [Type] 및 [EBS]를 차례로 선택하고 필드를 작성합니다. 그런 다음 새 AMI에서 인스턴스를 시작하면 추가 볼륨이 인스턴스에 자동으로 연결됩니다. 빈 볼륨은 반드시 포맷하고 마운트해야 합니다. 스냅샷 기반 볼륨을 반드시 마운트해야 합니다.
- 인스턴스를 시작하는 데 사용되는 AMI의 블록 디바이스 매핑에서 지정된 인스턴스 스토어 볼륨을 표시하지 않으려면 목록에서 해당 볼륨으로 이동한 다음 [Delete]를 선택합니다.
- 인스턴스 스토어 볼륨을 추가하려면 [Add New Volume], [Type] 및 [Instance Store]를 차례로 선택하고 [Device] 목록에서 디바이스 이름을 선택합니다. 새 AMI에서 인스턴스를 시작하면 추가 볼륨이 자동으로 시작되어 마운트됩니다. 이러한 볼륨에는 AMI를 기반으로 하는 실행 중인 인스턴스에 대한 인스턴스 스토어 볼륨의 데이터가 포함되어 있지 않습니다.

- AMI가 생성되는 동안 탐색 창에서 [AMIs]를 선택하여 상태를 볼 수 있습니다. 초기에 이 상태는 pending입니다. 몇 분 후 상태는 available로 변경됩니다.

(선택 사항) 탐색 창에서 [Snapshots]를 선택하여 새 AMI에 대해 생성된 스냅샷을 봅니다. 이 AMI에서 인스턴스를 시작할 때 이 스냅샷을 사용하여 루트 디바이스 볼륨을 생성합니다.

- 새 AMI에서 인스턴스를 시작합니다. 자세한 내용은 [인스턴스 시작하기 \(p. 265\)](#) 섹션을 참조하십시오.
- 실행 중인 새 인스턴스에는 이전 단계에서 적용한 모든 사용자 지정이 포함되어 있습니다.

명령줄을 사용하여 인스턴스에서 AMI를 생성하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- `create-image`(AWS CLI)
- `New-EC2Image`(Windows PowerShell용 AWS 도구)

스냅샷에서 Linux AMI 만들기

인스턴스의 루트 디바이스 볼륨에 대한 스냅샷이 있는 경우 AWS Management Console 또는 명령줄을 사용하여 이 스냅샷에서 AMI를 생성할 수 있습니다.

Important

Red Hat Enterprise Linux(RHEL) 및 SUSE Linux Enterprise Server(SLES)와 같은 일부 Linux 배포는 AMI와 연결된 Amazon EC2 `billingProduct` 코드를 사용하여 패키지 업데이트의 구독 상태를 확인합니다. EBS 스냅샷에서 AMI를 생성하면 이 결제 코드가 유지되지 않으며, 이러한 AMI에서 시작된 후속 인스턴스는 패키지 업데이트 인프라에 연결할 수 없습니다.
마찬가지로, 스냅샷에서 Windows AMI를 생성할 수 있지만 그러면 AMI에서 인스턴스를 제대로 시작할 수 없습니다.
일반적으로 AWS에서는 스냅샷에서 AMI를 수동으로 생성하는 것을 권장하지 않습니다.
적절한 작동을 위해 AMI 청구 코드를 유지해야 하는 Windows AMI 또는 Linux 운영 체제용 AMI 생성에 대한 자세한 내용은 [인스턴스에서 Linux AMI 생성 \(p. 82\)](#)을 참조하십시오.

콘솔을 사용하여 스냅샷에서 AMI를 생성하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창의 [Elastic Block Store] 아래에서 [Snapshots]를 선택합니다.
3. 스냅샷을 선택하고 [Actions], [Create Image]를 차례로 선택합니다.
4. [Create Image from EBS Snapshot] 대화 상자에서 AMI를 생성하기 위한 필드를 작성한 다음 [Create]를 선택합니다. 상위 인스턴스를 다시 생성하는 경우 상위 인스턴스와 동일한 옵션을 선택합니다.
 - [Architecture]: 32비트의 경우 [i386]을 선택하고 64비트의 경우 [x86_64]를 선택합니다.
 - [Root device name]: 루트 볼륨에 적절한 이름을 입력합니다. 자세한 내용은 [Linux 인스턴스의 디바이스 명명 \(p. 660\)](#) 섹션을 참조하십시오.
 - [Virtualization type]: 이 AMI에서 시작된 인스턴스가 반가상화(PV)를 사용하는지 또는 하드웨어 가상 머신(HVM) 가상화를 사용하는지를 선택합니다. 자세한 내용은 [Linux AMI 가상화 유형 \(p. 66\)](#) 섹션을 참조하십시오.
 - (PV 가상화 유형에만 해당) [Kernel ID] 및 [RAM disk ID]: 목록에서 AKI 및 ARI를 선택합니다. 기본 AKI를 선택하거나 AKI를 선택하지 않으면 이 AMI를 사용하여 인스턴스를 시작할 때마다 AKI를 지정해야 합니다. 또한 기본 AKI가 인스턴스와 호환되지 않는 경우, 상태 확인 작업 시 인스턴스 오류가 발생할 수 있습니다.
 - (선택 사항) [Block Device Mappings]: 볼륨을 추가하거나 AMI에 대한 루트 볼륨의 기본 크기를 확장합니다. 더 큰 볼륨을 사용할 수 있도록 인스턴스의 파일 시스템 크기 조정에 대한 자세한 내용은 [볼륨 크기 조정 후 Linux 파일 시스템 확장 \(p. 594\)](#) 섹션을 참조하십시오.

명령줄을 사용하여 스냅샷에서 AMI를 생성하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- `register-image` (AWS CLI)
- `Register-EC2Image` (Windows PowerShell용 AWS 도구)

인스턴스 스토어 기반 Linux AMI 생성

인스턴스 스토어 기반 Linux AMI를 만들려면 기존 인스턴스 스토어 기반 Linux AMI에서 시작한 인스턴스에서 시작합니다. 필요에 맞게 인스턴스를 사용자 지정한 후에는 볼륨을 번들링하고 이러한 사용자 지정을 적용하여 새 인스턴스를 시작하는 데 사용할 수 있는 새 AMI를 등록합니다.

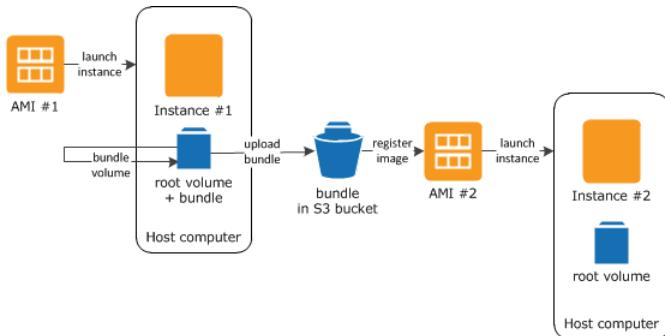
인스턴스 스토어 기반 Windows AMI를 만들어야 하는 경우 Windows 인스턴스용 Amazon EC2 사용 설명서의 [인스턴스 스토어 기반 Windows AMI 생성](#)을 참조하십시오.

AMI 생성 프로세스는 인스턴스 스토어 기반 AMI의 경우와는 다릅니다. Amazon EBS 기반 인스턴스와 인스턴스 스토어 기반 인스턴스의 차이 및 인스턴스의 루트 디바이스 유형을 확인하는 방법에 대한 자세한 내용

은 루트 디바이스 스토리지 (p. 64) 섹션을 참조하십시오. Amazon EBS 기반 Linux AMI를 만들어야 하는 경우 Amazon EBS 지원 Linux AMI 생성 (p. 81) 섹션을 참조하십시오.

인스턴스 스토어 기반 AMI 생성 프로세스 개요

다음 다이어그램은 인스턴스 스토어 기반 인스턴스에서 AMI를 만드는 프로세스를 요약한 것입니다.



우선 만들려는 AMI와 비슷한 AMI에서 인스턴스를 시작합니다. 인스턴스에 연결하여 인스턴스를 사용자 지정할 수 있습니다. 인스턴스가 원하는 대로 설정되었으면 이 인스턴스를 번들링할 수 있습니다. 번들링 프로세스가 완료되는 데 몇 분 정도 걸립니다. 프로세스가 완료된 후에는 이미지 매니페스트 (`image.manifest.xml`)와 루트 볼륨 템플릿을 포함하는 파일(`image.part.xx`)로 구성된 번들이 만들어집니다. 그 다음에는 이 번들을 Amazon S3 버킷으로 업로드하고 AMI를 등록합니다.

새 AMI를 사용하여 인스턴스를 시작하는 경우 Amazon S3으로 업로드한 번들을 사용하여 인스턴스용 루트 볼륨이 생성됩니다. Amazon S3의 번들에 사용된 스토리지 공간에 대해 사용자가 삭제할 때까지 사용자 계정에 요금이 발생합니다. 자세한 내용은 [AMI 등록 취소 \(p. 130\)](#) 섹션을 참조하십시오.

루트 디바이스 볼륨 외에도 인스턴스에 인스턴스 스토어 볼륨을 추가하는 경우, 새 AMI에 대한 블록 디바이스 매핑과 새 AMI에서 시작하는 인스턴스에 대한 블록 디바이스 매핑에 이러한 볼륨에 대한 정보가 포함됩니다. 자세한 내용은 [블록 디바이스 매핑 \(p. 662\)](#) 섹션을 참조하십시오.

사전 조건

AMI를 만들려면 먼저 다음 작업을 완료해야 합니다.

- AMI 도구를 설치합니다. 자세한 내용은 [AMI 도구 설치 \(p. 86\)](#) 섹션을 참조하십시오.
- AWS CLI를 설치합니다. 자세한 내용은 [AWS Command Line Interface 설정 시작하기](#)를 참조하십시오.
- 번들용 Amazon S3 버킷이 있는지 확인합니다. Amazon S3 버킷을 만들려면 Amazon S3 콘솔을 열고 [Create Bucket]을 클릭합니다. 그 밖에 AWS CLI `mb` 명령을 사용할 수도 있습니다.
- AWS 계정 ID가 있어야 합니다. 자세한 내용은 AWS General Reference에서 [AWS Account Identifiers](#) 단원을 참조하십시오.
- 액세스 키 ID와 보안 액세스 키가 있어야 합니다. 자세한 내용은 AWS General Reference에서 [Access Keys](#) 단원을 참조하십시오.
- X.509 인증서와 그에 따른 프라이빗 키가 있어야 합니다.
 - X.509 인증서를 만들어야 할 경우 [서명 인증서 관리 \(p. 108\)](#) 섹션을 참조하십시오. X.509 인증서 및 프라이빗 키는 AMI를 암호화하고 해독하는 데 사용됩니다.
 - [중국(베이징)] `$EC2_AMITOOL_HOME/etc/ec2/amitools/cert-ec2-cn-north-1.pem` 인증서를 사용하십시오.
 - [AWS GovCloud (US)] `$EC2_AMITOOL_HOME/etc/ec2/amitools/cert-ec2-gov.pem` 인증서를 사용하십시오.
- 인스턴스에 연결하여 인스턴스를 사용자 지정합니다. 예를 들어, 소프트웨어 및 애플리케이션을 설치하고, 데이터를 복사하고, 임시 파일을 삭제하고, Linux 구성을 수정할 수 있습니다.

항목

- [AMI 도구 설치 \(p. 86\)](#)
- [인스턴스 스토어 지원 Amazon Linux 인스턴스에서 AMI 생성 \(p. 112\)](#)
- [인스턴스 스토어 지원 Ubuntu 인스턴스에서 AMI 생성 \(p. 116\)](#)
- [인스턴스 스토어 기반 AMI를 Amazon EBS 기반 AMI로 변환 \(p. 121\)](#)

AMI 도구 설치

AMI 도구를 사용하면 인스턴스 스토어 지원 Linux AMI를 생성하고 관리할 수 있습니다. 이 도구를 사용하려면 이를 Linux 인스턴스에 설치해야 합니다. AMI 도구는 RPM으로도 설치 가능하고 RPM을 지원하지 않는 Linux 배포판의 경우 .zip 파일로도 설치 가능합니다. 자세한 내용은 [Amazon EC2 AMI 도구](#) 섹션을 참조하십시오.

Note

AMI 도구는 인스턴스 스토어 지원 Linux 인스턴스에서만 지원됩니다. Amazon EBS 지원 AMI를 생성하려면 [create-image](#) AWS CLI 명령을 대신 사용합니다. 인스턴스 스토어 지원 Windows AMI를 생성하려면 [인스턴스 스토어 지원 Windows AMI 생성](#)을 참조하십시오.

RPM을 사용하여 AMI 도구를 설치하려면

1. yum과 같은 Linux 배포용 패키지 관리자를 사용하여 Ruby를 설치합니다. 예:

```
$ sudo yum install -y ruby
```

2. wget 또는 curl과 같은 도구를 사용하여 RPM 파일을 다운로드합니다. 예:

```
$ sudo wget https://s3.amazonaws.com/ec2-downloads/ec2-ami-tools.noarch.rpm
```

3. 다음 명령을 사용하여 RPM을 설치합니다.

```
$ sudo yum install ec2-ami-tools.noarch.rpm
```

4. 다음 명령을 사용하여 AMI 도구 설치를 확인합니다.

```
$ ec2-ami-tools-version
```

Note

cannot load such file -- ec2/amitools/version (LoadError)과 같은 로드 오류가 발생하면 다음 단계를 수행하여 AMI 도구 설치 위치를 RUBYLIB 경로에 추가합니다.

5. (선택 사항) 이전 단계에서 오류가 발생하면 AMI 도구 설치 위치를 RUBYLIB 경로에 추가합니다.

- a. 다음 명령을 실행하여 추가할 경로를 확인합니다.

```
$ rpm -qil ec2-ami-tools | grep ec2/amitools/version
/usr/lib/ruby/site_ruby/ec2/amitools/version.rb
/usr/lib64/ruby/site_ruby/ec2/amitools/version.rb
```

위의 예제를 보면 이전 로드 경로에서 없다고 표시된 파일이 /usr/lib/ruby/site_ruby 및 /usr/lib64/ruby/site_ruby에 위치하고 있습니다.

- b. 이전 단계의 위치를 RUBYLIB 경로에 추가합니다.

```
$ export RUBYLIB=$RUBYLIB:/usr/lib/ruby/site_ruby:/usr/lib64/ruby/site_ruby
```

- c. 다음 명령을 사용하여 AMI 도구 설치를 확인합니다.

```
$ ec2-ami-tools-version
```

.zip 파일을 사용하여 AMI 도구를 설치하려면

1. apt-get과 같은 Linux 배포용 패키지 관리자를 사용하여 Ruby를 설치하고 압축을 풁니다. 예:

```
$ sudo apt-get update -y && sudo apt-get install -y ruby unzip
```

2. wget 또는 curl과 같은 도구를 사용하여 .zip 파일을 다운로드합니다. 예:

```
$ wget https://s3.amazonaws.com/ec2-downloads/ec2-ami-tools.zip
```

3. 파일의 압축을 적합한 설치 디렉터리(예: /usr/local/ec2)에 풁니다.

```
$ sudo mkdir -p /usr/local/ec2
$ sudo unzip ec2-ami-tools.zip -d /usr/local/ec2
```

.zip 파일에는 ec2-ami-tools-*x.x.x* 폴더가 들어 있습니다. 여기서 *x.x.x*는 도구의 버전 번호입니다(예: ec2-ami-tools-1.5.7).

4. EC2_AMITOOL_HOME 환경 변수를 도구의 설치 디렉터리로 설정합니다. 예:

```
$ export EC2_AMITOOL_HOME=/usr/local/ec2/ec2-ami-tools-x.x.x
```

5. 도구를 PATH 환경 변수에 추가합니다. 예:

```
$ export PATH=$EC2_AMITOOL_HOME/bin:$PATH
```

6. 다음 명령으로 AMI 도구 설치를 확인할 수 있습니다.

```
$ ec2-ami-tools-version
```

AMI 도구 명령

AMI 도구와 함께 다음 명령을 사용하면 인스턴스 스토어 지원 Linux AMI를 생성하고 관리할 수 있습니다. 이 도구를 설치하려면 [AMI 도구 설치 \(p. 86\)](#)을 참조하십시오.

항목

- [ec2-ami-tools-version \(p. 88\)](#)
- [ec2-bundle-image \(p. 88\)](#)
- [ec2-bundle-vol \(p. 91\)](#)
- [ec2-delete-bundle \(p. 96\)](#)
- [ec2-download-bundle \(p. 98\)](#)
- [ec2-migrate-manifest \(p. 101\)](#)
- [ec2-unbundle \(p. 103\)](#)
- [ec2-upload-bundle \(p. 104\)](#)
- [AMI 도구의 일반 옵션 \(p. 107\)](#)

ec2-ami-tools-version

설명

AMI 도구 버전을 설명합니다.

구문

ec2-ami-tools-version

옵션

이 명령에는 파라미터가 없습니다.

결과

버전 정보입니다.

예

이 예제 명령은 사용 중인 AMI 도구의 버전 정보를 표시합니다.

```
$ ec2-ami-tools-version  
1.5.2 20071010
```

ec2-bundle-image

설명

루프백 파일에서 생성한 운영 체제 이미지로부터 인스턴스 스토어 지원 Linux AMI를 생성합니다.

구문

```
ec2-bundle-image -c path -k path -u account -i path [-d path] [--ec2cert path] [-r architecture] [--productcodes code1,code2,...] [-B mapping] [-p prefix]
```

옵션

옵션	설명
-c, --cert path	사용자의 PEM 인코딩된 RSA 퍼블릭 키 인증서 파일입니다. 필수 항목 여부: 예 예: -c cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem
-k, --privatekey path	PEM 인코딩된 RSA 키 파일의 경로입니다. 이 번들이 번들링 되지 않아 안전한 장소에 보관되도록 이 키를 지정해야 합니다. 키를 AWS 계정에 등록할 필요는 없습니다. 필수 항목 여부: 예 예: -k pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem
-u, --user account	사용자의 AWS 계정 ID(대시 없이)입니다. 필수 항목 여부: 예 예: -u 111122223333
-i, --image path	번들링할 이미지에 대한 경로입니다.

옵션	설명
	<p>필수 항목 여부: 예</p> <p>예: <code>-i /var/spool/my-image/version-2/debian.img</code></p>
<code>-d, --destination path</code>	<p>번들을 생성할 디렉터리입니다.</p> <p>기본값: <code>/tmp</code></p> <p>필수 항목 여부: 아니요</p> <p>예: <code>-d /media/ephemeral0</code></p>
<code>--ec2cert path</code>	<p>이미지 매니페스트를 암호화하는 데 사용되는 Amazon EC2 X.509 퍼블릭 키 인증서의 경로입니다.</p> <p><code>us-gov-west-1</code> 및 <code>cn-north-1</code> 리전은 기본값이 아닌 퍼블릭 키 인증서를 사용하며 해당 인증서의 경로를 이 옵션으로 지정해야 합니다. 인증서 경로는 AMI 도구의 설치 방법에 따라 다릅니다. Amazon Linux의 경우 인증서가 <code>/opt/aws/amitools/ec2/etc/ec2/amitools/</code>에 있습니다. AMI 도구 설치 (p. 86)의 RPM 또는 ZIP 파일에서 AMI 도구를 설치했으면 인증서가 <code>\$EC2_AMITOOL_HOME/etc/ec2/amitools/</code>에 있습니다.</p> <p>기본값: 도구에 따라 다름</p> <p>필수 항목 여부: <code>us-gov-west-1</code> 및 <code>cn-north-1</code> 리전의 경우에만.</p> <p>예: <code>--ec2cert \$EC2_AMITOOL_HOME/etc/ec2/amitools/cert-ec2.pem</code></p>
<code>-r, --arch architecture</code>	<p>이미지 아키텍처. 명령줄에 아키텍처를 제공하지 않으면 번들링 시작 시 해당 메시지가 표시됩니다.</p> <p>유효한 값: <code>i386 x86_64</code></p> <p>필수 항목 여부: 아니요</p> <p>예: <code>-r x86_64</code></p>
<code>--productcodes code1,code2,...</code>	<p>등록 시 이미지에 연결할 제품 코드로, 쉼표로 구분됩니다.</p> <p>필수 항목 여부: 아니요</p> <p>예: <code>--productcodes 1234abcd</code></p>

옵션	설명
<code>-B, --block-device-mapping mapping</code>	<p>이 AMI의 인스턴스에 블록 디바이스를 표시할 방법을 정의합니다(인스턴스 유형이 지정된 디바이스를 지원하는 경우).</p> <p>쉼표로 구분된 키-값 페어 목록을 지정합니다. 여기서 각 키는 가상 이름이며 각 값은 해당 디바이스 이름입니다. 가상 이름에는 다음이 포함됩니다.</p> <ul style="list-style-type: none"> • ami-인스턴스에서 보이는 것과 같은 루트 파일 시스템 디바이스 • root-커널에서 보이는 것과 같은 루트 파일 시스템 디바이스 • swap-인스턴스에서 보이는 것과 같은 교체 디바이스 • ephemeralN-N번째 인스턴스 스토어 볼륨 <p>필수 항목 여부: 아니요</p> <p>예: <code>--block-device-mapping ami=sda1,root=/dev/sda1,ephemeral0=sda2,swap=sda3</code></p> <p>예: <code>--block-device-mapping ami=0,root=/dev/dsk/c0d0s0,ephemeral0=1</code></p>
<code>-p, --prefix prefix</code>	<p>번들링된 AMI 파일의 파일 이름 접두사입니다.</p> <p>기본값: 이미지 파일의 이름입니다. 예를 들어, 이미지 경로가 <code>/var/spool/my-image/version-2/debian.img</code>이면, 기본 접두사는 <code>debian.img</code>입니다.</p> <p>필수 항목 여부: 아니요</p> <p>예: <code>-p my-image-is-special</code></p>
<code>--kernel kernel_id</code>	<p>사용되지 않음. register-image를 사용하여 커널을 설정합니다.</p> <p>필수 항목 여부: 아니요</p> <p>예: <code>--kernel aki-ba3adfd3</code></p>
<code>--ramdisk ramdisk_id</code>	<p>사용되지 않음. register-image를 사용하여 RAM 디스크를 설정합니다(필요한 경우).</p> <p>필수 항목 여부: 아니요</p> <p>예: <code>--ramdisk ari-badbad00</code></p>
일반 옵션	<p>대부분의 AMI 도구에 공통적인 옵션에 대한 자세한 내용은 AMI 도구의 일반 옵션 (p. 107)을 참조하십시오.</p>

결과

번들링 프로세스의 단계 및 상태를 설명하는 상태 메시지입니다.

예

이 예제에서는 루프백 파일에서 생성한 운영 체제 이미지로부터 번들링된 AMI를 생성합니다.

```
$ ec2-bundle-image -k pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -c cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -u 111122223333 -i image.img -d bundled/ -r x86_64
Please specify a value for arch [i386]:
Bundling image file...
Splitting bundled/image.gz.crypt...
Created image.part.00
Created image.part.01
Created image.part.02
Created image.part.03
Created image.part.04
Created image.part.05
Created image.part.06
Created image.part.07
Created image.part.08
Created image.part.09
Created image.part.10
Created image.part.11
Created image.part.12
Created image.part.13
Created image.part.14
Generating digests for each part...
Digests generated.
Creating bundle manifest...
ec2-bundle-image complete.
```

ec2-bundle-vol

설명

인스턴스의 루트 디바이스 볼륨의 복사본을 압축, 암호화 및 서명하여 인스턴스 스토어 지원 Linux AMI를 생성합니다.

인스턴스로부터 제품 코드, 커널 설정, RAM 디스크 설정 및 블록 디바이스 매핑을 상속하려는 Amazon EC2 시도입니다.

기본적으로 번들 프로세스에는 중요 정보를 포함할 수 있는 파일이 제외됩니다. *.sw, *.swo, *.swp, *.pem, *.priv, *id_rsa*, *id_dsa* *.gpg, *.jks, */.ssh/authorized_keys, */.bash_history 등을 예로 들 수 있습니다. 이러한 파일을 모든 포함하려면 --no-filter 옵션을 사용합니다. 이러한 파일 중 일부만 포함하려면 --include 옵션을 사용합니다.

자세한 내용은 [인스턴스 스토어 지원 Linux AMI 생성](#)을 참조하십시오.

구문

```
ec2-bundle-vol -c path -k path -u account [-d path] [--ec2cert path] [-r architecture]
[--productcodes code1,code2,...] [-B mapping] [--all] [-e directory1,directory2,...] [-i
file1,file2,...] [--no-filter] [-p prefix] [-s size] [--[no-]inherit] [-v volume] [-P type]
[-S script] [--fstab path] [--generate-fstab] [--grub-config path]
```

옵션

옵션	설명
-c, --cert path	사용자의 PEM 인코딩된 RSA 퍼블릭 키 인증서 파일입니다. 필수 항목 여부: 예 예: -c cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem
-k, --privatekey path	사용자의 PEM 인코딩된 RSA 키 파일의 경로입니다. 필수 항목 여부: 예

옵션	설명
	예: -k pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem
-u, --user account	사용자의 AWS 계정 ID(대시 없이)입니다. 필수 항목 여부: 예 예: -u 111122223333
-d, --destination destination	번들을 생성할 디렉터리입니다. 기본값: /tmp 필수 항목 여부: 아니요 예: -d /var/run/my-bundle
--ec2cert path	이미지 매니페스트를 암호화하는 데 사용되는 Amazon EC2 X.509 퍼블릭 키 인증서의 경로입니다. us-gov-west-1 및 cn-north-1 리전은 기본값이 아닌 퍼블릭 키 인증서를 사용하며 해당 인증서의 경로를 이 옵션으로 지정해야 합니다. 인증서 경로는 AMI 도구의 설치 방법에 따라 다릅니다. Amazon Linux의 경우 인증서가 /opt/aws/amitools/ec2/etc/ec2/amitools/에 있습니다. AMI 도구 설치 (p. 86)의 RPM 또는 ZIP 파일에서 AMI 도구를 설치했으면 인증서가 \$EC2_AMITOOL_HOME/etc/ec2/amitools/에 있습니다. 기본값: 도구에 따라 다름 필수 항목 여부: us-gov-west-1 및 cn-north-1 리전의 경우에만. 예: --ec2cert \$EC2_AMITOOL_HOME/etc/ec2/amitools/cert-ec2.pem
-r, --arch architecture	이미지 아키텍처입니다. 명령줄에 이를 제공하지 않으면 번들링 시작 시 이를 제공하라는 메시지가 표시됩니다. 유효한 값: i386 x86_64 필수 항목 여부: 아니요 예: -r x86_64
--productcodes code1,code2,...	등록 시 이미지에 연결할 제품 코드로, 쉼표로 구분됩니다. 필수 항목 여부: 아니요 예: --productcodes 1234abcd

옵션	설명
<code>-B, --block-device-mapping mapping</code>	<p>이 AMI의 인스턴스에 블록 디바이스를 표시할 방법을 정의합니다(인스턴스 유형이 지정된 디바이스를 지원하는 경우).</p> <p>쉼표로 구분된 키-값 페어 목록을 지정합니다. 여기서 각 키는 가상 이름이며 각 값은 해당 디바이스 이름입니다. 가상 이름에는 다음이 포함됩니다.</p> <ul style="list-style-type: none"> • ami-인스턴스에서 보이는 것과 같은 루트 파일 시스템 디바이스 • root-커널에서 보이는 것과 같은 루트 파일 시스템 디바이스 • swap-인스턴스에서 보이는 것과 같은 교체 디바이스 • ephemeralN-N번째 인스턴스 스토어 볼륨 <p>필수 항목 여부: 아니요</p> <p>예: <code>--block-device-mapping ami=sda1,root=/dev/sda1,ephemeral0=sda2,swap=sda3</code></p> <p>예: <code>--block-device-mapping ami=0,root=/dev/dsk/c0d0s0,ephemeral0=1</code></p>
<code>-a, --all</code>	<p>원격으로 마운트된 파일 시스템의 디렉터리를 포함하여 모든 디렉터리를 번들링합니다.</p> <p>필수 항목 여부: 아니요</p> <p>예: <code>-a</code></p>
<code>-e, --exclude directory1,directory2,...</code>	<p>번들 작업에서 제외할 절대 디렉터리 경로 및 파일 목록입니다. 이 파라미터는 <code>--all</code> 옵션보다 우선합니다. <code>exclude</code>가 지정되면 파라미터와 함께 나열된 디렉터리 및 하위 디렉터리는 볼륨에 번들링되지 않습니다.</p> <p>필수 항목 여부: 아니요</p> <p>예: 볼륨의 마운트 지점이 <code>-v /foo</code>일 때 <code>/foo/bar</code> 및 <code>/foo/baz</code> 디렉터리를 제외하려면, <code>-e /bar,/baz</code>를 지정합니다.</p>
<code>-i, --include file1,file2,...</code>	<p>번들 작업에 포함할 파일 목록입니다. 지정된 파일은 중요한 정보를 포함할 수 있으므로 AMI에서 제외되지 않습니다.</p> <p>필수 항목 여부: 아니요</p> <p>예: 볼륨 마운트 지점이 <code>/mnt/myvol/</code>이고 <code>/mnt/myvol/foo/bar.pem</code> 파일을 포함하려는 경우, <code>-i /foo/bar.pem</code>을 지정합니다.</p>
<code>--no-filter</code>	<p>지정한 경우 지정한 파일이 중요한 정보를 포함할 수 있으므로 AMI에서 파일을 제외하지 않습니다.</p> <p>필수 항목 여부: 아니요</p> <p>예: <code>--no-filter</code></p>

옵션	설명
<code>-p, --prefix prefix</code>	<p>번들링된 AMI 파일의 파일 이름 접두사입니다.</p> <p>기본값: <code>image</code></p> <p>필수 항목 여부: 아니요</p> <p>예: <code>-p my-image-is-special</code></p>
<code>-s, --size size</code>	<p>생성할 이미지 파일의 크기(MB, 1024 * 1024바이트)입니다. 최대 크기는 10240MB입니다.</p> <p>기본값: 10240</p> <p>필수 항목 여부: 아니요</p> <p>예: <code>-s 2048</code></p>
<code>--[no-]inherit</code>	<p>이미지가 인스턴스의 메타데이터를 상속해야 하는지 여부를 나타냅니다(기본값은 상속). <code>--inherit</code>을 활성화했지만 인스턴스 메타데이터에 액세스할 수 없으면 번들링이 실패합니다.</p> <p>필수 항목 여부: 아니요</p> <p>예: <code>--inherit</code></p>
<code>-v, --volume volume</code>	<p>번들을 생성해 올 마운트된 볼륨의 절대 경로입니다.</p> <p>기본값: 루트 디렉터리(/)</p> <p>필수 항목 여부: 아니요</p> <p>예: <code>-v /mnt/my-customized-ami</code></p>
<code>-P, --partition type</code>	<p>디스크 이미지에서 파티션 테이블을 사용해야 하는지 여부를 나타냅니다. 파티션 테이블 유형을 지정하지 않으면, 볼륨의 상위 블록 디바이스에서 사용한 유형이 기본값이 됩니다. 이를 적용할 수 없으면 <code>gpt</code>가 기본값이 됩니다.</p> <p>유효한 값: <code>mbr gpt none</code></p> <p>필수 항목 여부: 아니요</p> <p>예: <code>--partition gpt</code></p>
<code>-S, --script script</code>	<p>번들링 작업 직전에 실행할 사용자 정의 스크립트입니다. 스크립트에서 하나의 인수(볼륨의 마운트 지점)를 예상해야 합니다.</p> <p>필수 항목 여부: 아니요</p>
<code>--fstab path</code>	<p>이미지에 번들링할 <code>fstab</code> 경로입니다. 지정하지 않으면 Amazon EC2가 <code>/etc/fstab</code>을 번들링합니다.</p> <p>필수 항목 여부: 아니요</p> <p>예: <code>--fstab /etc/fstab</code></p>

옵션	설명
--generate-fstab	Amazon EC2-제공 fstab을 사용하여 볼륨을 번들링합니다. 필수 항목 여부: 아니요 예: --generate-fstab
--grub-config	이미지에 번들링할 대체 grub 구성 파일 경로입니다. 기본적으로, ec2-bundle-vol은 /boot/grub/menu.lst 또는 /boot/grub/grub.conf가 복제된 이미지에 있을 것으로 예상합니다. 이 옵션을 사용하면 대체 grub 구성 파일에 대한 경로를 지정할 수 있습니다. 이 경로는 기본값(있는 경우)을 덮어씁니다. 필수 항목 여부: 아니요 예: --grub-config /path/to/grub.conf
--kernel kernel_id	사용되지 않음. register-image 를 사용하여 커널을 설정합니다. 필수 항목 여부: 아니요 예: --kernel aki-ba3adfd3
--ramdisk ramdisk_id	사용되지 않음. register-image 를 사용하여 RAM 디스크를 설정합니다(필요한 경우). 필수 항목 여부: 아니요 예: --ramdisk ari-badbad00
일반 옵션	대부분의 AMI 도구에 공통적인 옵션에 대한 자세한 내용은 AMI 도구의 일반 옵션 (p. 107) 을 참조하십시오.

결과

번들링 단계 및 상태를 설명하는 상태 메시지입니다.

예

이 예제에서는 로컬 시스템의 루트 파일 시스템의 스냅샷을 압축, 암호화 및 서명하여 번들링된 AMI를 생성합니다.

```
$ ec2-bundle-vol -d /mnt -k pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -c cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -u 111122223333 -r x86_64
Copying / into the image file /mnt/image...
Excluding:
  sys
  dev/shm
  proc
  dev/pts
  proc/sys/fs/binfmt_misc
  dev
  media
  mnt
  proc
  sys
  tmp/image
  mnt/img-mnt
1+0 records in
1+0 records out
```

```
mke2fs 1.38 (30-Jun-2005)
warning: 256 blocks unused.

Splitting /mnt/image.gz.crypt...
Created image.part.00
Created image.part.01
Created image.part.02
Created image.part.03
...
Created image.part.22
Created image.part.23
Generating digests for each part...
Digests generated.
Creating bundle manifest...
Bundle Volume complete.
```

ec2-delete-bundle

설명

Amazon S3 스토리지에서 지정된 번들을 삭제합니다. 번들을 삭제한 다음에는 해당 AMI에서 인스턴스를 시작할 수 있습니다.

구문

```
ec2-delete-bundle -b bucket -a access_key_id -s secret_access_key [-t token] [--url url]
[--region region] [--sigv version] [-m path] [-p prefix] [--clear] [--retry] [-y]
```

옵션

옵션	설명
-b, --bucket bucket	번들링된 AMI를 포함하는 Amazon S3 버킷 이름으로, '/'-delimited 경로 접두사가 붙기도 합니다(옵션). 필수 항목 여부: 예 예제: -b myawsbucket/ami-001
-a, --access-key access_key_id	AWS 액세스 키 ID입니다. 이 옵션의 값을 지정하기 전에, AWS 액세스 키 관리 모범 사례 의 지침을 검토하고 따르십시오. 필수 항목 여부: 예 예: -a AKIAIOSFODNN7EXAMPLE
-s, --secret-key secret_access_key	AWS 보안 액세스 키입니다. 이 옵션의 값을 지정하기 전에, AWS 액세스 키 관리 모범 사례 의 지침을 검토하고 따르십시오. 필수 항목 여부: 예 예: -s wJalrXUtnFEMI/K7MDENG/bPxRficyEXAMPLEKEY
-t, --delegation-token token	AWS 요청에 함께 전달되는 위임 토큰입니다. 자세한 내용은 Using Temporary Security Credentials 를 참조하십시오. 필수 항목 여부: 임시 보안 자격 증명을 사용하는 경우에만. 기본값: AWS_DELEGATION_TOKEN 환경 변수 값(설정된 경우). 예: -t AQoDYXdzEJr...<remainder of security token>

옵션	설명
<code>--region region</code>	<p>요청 서명에서 사용하는 리전입니다.</p> <p>기본값: <code>us-east-1</code></p> <p>필수 항목 여부: 조건부</p> <p>조건: 서명 버전 4를 사용하는 경우 필수</p> <p>예: <code>--region eu-west-1</code></p>
<code>--sigv version</code>	<p>요청 서명 시 사용하는 서명 버전입니다.</p> <p>유효한 값: 2 4</p> <p>기본값: 4</p> <p>필수 항목 여부: 아니요</p> <p>예: <code>--sigv 2</code></p>
<code>-m, --manifest path</code>	<p>매니페스트 파일 경로입니다.</p> <p>필수 항목 여부: 조건부</p> <p>조건: <code>--prefix</code> 또는 <code>--manifest</code>을 지정해야 합니다.</p> <p>예: <code>-m /var/spool/my-first-bundle/image.manifest.xml</code></p>
<code>-p, --prefix prefix</code>	<p>번들링된 AMI 파일 이름 접두사입니다. 전체 접두사를 제공합니다. 예를 들어 접두사가 <code>image.img</code>인 경우, <code>-p image.img</code>을 사용해야 합니다(<code>-p image</code> 아님).</p> <p>필수 항목 여부: 조건부</p> <p>조건: <code>--prefix</code> 또는 <code>--manifest</code>을 지정해야 합니다.</p> <p>예: <code>-p image.img</code></p>
<code>--clear</code>	<p>지정된 번들을 삭제한 후 비어 있으면 Amazon S3 버킷을 삭제합니다.</p> <p>필수 항목 여부: 아니요</p> <p>예: <code>--clear</code></p>
<code>--retry</code>	<p>모든 Amazon S3 오류에서 작업당 최대 5회 자동으로 재시도 합니다.</p> <p>필수 항목 여부: 아니요</p> <p>예: <code>--retry</code></p>
<code>-y, --yes</code>	<p>모든 질문 메시지에 대한 답을 <code>yes</code>로 자동으로 가정합니다.</p> <p>필수 항목 여부: 아니요</p> <p>예: <code>-y</code></p>

옵션	설명
일반 옵션	대부분의 AMI 도구에 공통적인 옵션에 대한 자세한 내용은 AMI 도구의 일반 옵션 (p. 107) 을 참조하십시오.

결과

Amazon EC2에서 삭제 프로세스의 단계 및 상황을 나타내는 상태 메시지를 표시합니다.

예

이 예제에서는 Amazon S3에서 번들을 삭제합니다.

```
$ ec2-delete-bundle -b myawsbucket -a your_access_key_id -s your_secret_access_key
Deleting files:
myawsbucket/image.manifest.xml
myawsbucket/image.part.00
myawsbucket/image.part.01
myawsbucket/image.part.02
myawsbucket/image.part.03
myawsbucket/image.part.04
myawsbucket/image.part.05
myawsbucket/image.part.06
Continue? [y/n]
y
Deleted myawsbucket/image.manifest.xml
Deleted myawsbucket/image.part.00
Deleted myawsbucket/image.part.01
Deleted myawsbucket/image.part.02
Deleted myawsbucket/image.part.03
Deleted myawsbucket/image.part.04
Deleted myawsbucket/image.part.05
Deleted myawsbucket/image.part.06
ec2-delete-bundle complete.
```

ec2-download-bundle

설명

지정된 인스턴스 스토어 지원 Linux AMI를 Amazon S3 스토리지에서 다운로드합니다.

구문

```
ec2-download-bundle -b bucket -a access_key_id -s secret_access_key -k path [--url url] [--region region] [--sigv version] [-m file] [-p prefix] [-d directory] [--retry]
```

옵션

옵션	설명
-b, --bucket bucket	번들이 있는 Amazon S3 버킷 이름으로, '/'-delimited 경로 접두사가 붙기도 합니다(옵션). 필수 항목 여부: 예 예: -b myawsbucket/ami-001
-a, --access-key access_key_id	AWS 액세스 키 ID입니다. 이 옵션의 값을 지정하기 전에, AWS 액세스 키 관리 모범 사례 의 지침을 검토하고 따르십시오.

옵션	설명
	<p>필수 항목 여부: 예</p> <p>예: <code>-a AKIAIOSFODNN7EXAMPLE</code></p>
<code>-s, --secret-key secret_access_key</code>	<p>AWS 보안 액세스 키입니다. 이 옵션의 값을 지정하기 전에, AWS 액세스 키 관리 모범 사례의 지침을 검토하고 따르십시오.</p> <p>필수 항목 여부: 예</p> <p>예: <code>-s wJalrXUtnFEMI/K7MDENG/bPxRficyEXAMPLEKEY</code></p>
<code>-k, --privatekey path</code>	<p>매니페스트를 해독하는 데 사용되는 프라이빗 키입니다.</p> <p>필수 항목 여부: 예</p> <p>예: <code>-k pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem</code></p>
<code>--url url</code>	<p>Amazon S3 서비스 URL입니다.</p> <p>기본값: <code>https://s3.amazonaws.com/</code></p> <p>필수 항목 여부: 아니요</p> <p>예: <code>--url https://s3.example.com</code></p>
<code>--region region</code>	<p>요청 서명에서 사용하는 리전입니다.</p> <p>기본값: <code>us-east-1</code></p> <p>필수 항목 여부: 조건부</p> <p>조건: 서명 버전 4를 사용하는 경우 필수</p> <p>예: <code>--region eu-west-1</code></p>
<code>--sigv version</code>	<p>요청 서명 시 사용하는 서명 버전입니다.</p> <p>유효한 값: 2 4</p> <p>기본값: 4</p> <p>필수 항목 여부: 아니요</p> <p>예: <code>--sigv 2</code></p>
<code>-m, --manifest file</code>	<p>매니페스트 파일 이름입니다(경로 제외). 매니페스트(-m) 또는 접두사(-p) 중 하나를 지정하는 것이 좋습니다.</p> <p>필수 항목 여부: 아니요</p> <p>예: <code>-m my-image.manifest.xml</code></p>
<code>-p, --prefix prefix</code>	<p>번들링된 AMI 파일의 파일 이름 접두사입니다.</p> <p>기본값: <code>image</code></p> <p>필수 항목 여부: 아니요</p> <p>예: <code>-p my-image</code></p>

옵션	설명
-d, --directory directory	다운로드된 번들이 저장되는 디렉터리입니다. 존재하는 디렉터리여야 합니다. 기본값: 현재 작업 디렉터리입니다. 필수 항목 여부: 아니요 예: -d /tmp/my-downloaded-bundle
--retry	모든 Amazon S3 오류에서 작업당 최대 5회 자동으로 재시도 합니다. 필수 항목 여부: 아니요 예: --retry
일반 옵션	대부분의 AMI 도구에 공통적인 옵션에 대한 자세한 내용은 AMI 도구의 일반 옵션 (p. 107) 을 참조하십시오.

결과

다운로드 프로세스의 다양한 단계를 나타내는 상태 메시지가 표시됩니다.

예

이 예제에서는 bundled 디렉터리(Linux mkdir 명령)를 생성하고 myawsbucket Amazon S3 버킷에서 번들을 다운로드합니다.

```
$ mkdir bundled
$ ec2-download-bundle -b myawsbucket/bundles/bundle_name -m image.manifest.xml -
a your_access_key_id -s your_secret_access_key -k pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -
d mybundle
Downloading manifest image.manifest.xml from myawsbucket to mybundle/image.manifest.xml ...
Downloading part image.part.00 from myawsbucket/bundles/bundle_name to mybundle/
image.part.00 ...
Downloaded image.part.00 from myawsbucket
Downloading part image.part.01 from myawsbucket/bundles/bundle_name to mybundle/
image.part.01 ...
Downloaded image.part.01 from myawsbucket
Downloading part image.part.02 from myawsbucket/bundles/bundle_name to mybundle/
image.part.02 ...
Downloaded image.part.02 from myawsbucket
Downloading part image.part.03 from myawsbucket/bundles/bundle_name to mybundle/
image.part.03 ...
Downloaded image.part.03 from myawsbucket
Downloading part image.part.04 from myawsbucket/bundles/bundle_name to mybundle/
image.part.04 ...
Downloaded image.part.04 from myawsbucket
Downloading part image.part.05 from myawsbucket/bundles/bundle_name to mybundle/
image.part.05 ...
Downloaded image.part.05 from myawsbucket
Downloading part image.part.06 from myawsbucket/bundles/bundle_name to mybundle/
image.part.06 ...
Downloaded image.part.06 from myawsbucket
```

ec2-migrate-manifest

설명

인스턴스 스토어 지원 Linux AMI(예: 해당 인증서, 커널 및 RAM 디스크)가 다른 리전을 지원하도록 수정합니다.

구문

```
ec2-migrate-manifest -c path -k path -m path {{(-a access_key_id -s secret_access_key --region region) | (--no-mapping)} [--ec2cert ec2_cert_path] [--kernel kernel-id] [--ramdisk ramdisk_id]}
```

옵션

옵션	설명
-c, --cert path	사용자의 PEM 인코딩된 RSA 퍼블릭 키 인증서 파일입니다. 필수 항목 여부: 예 예: -c cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem
-k, --privatekey path	사용자의 PEM 인코딩된 RSA 키 파일의 경로입니다. 필수 항목 여부: 예 예: -k pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem
--manifest path	매니페스트 파일 경로입니다. 필수 항목 여부: 예 예: --manifest my-ami.manifest.xml
-a, --access-key access_key_id	AWS 액세스 키 ID입니다. 이 옵션의 값을 지정하기 전에, AWS 액세스 키 관리 모범 사례 의 지침을 검토하고 따르십시오. 필수 항목 여부: 조건부 조건: 자동 매핑 사용 시 필수입니다. 예: -a AKIAIOSFODNN7EXAMPLE
-s, --secret-key secret_access_key	AWS 보안 액세스 키입니다. 이 옵션의 값을 지정하기 전에, AWS 액세스 키 관리 모범 사례 의 지침을 검토하고 따르십시오. 필수 항목 여부: 조건부 조건: 자동 매핑 사용 시 필수입니다. 예: -s wJalrXUtnFEMI/K7MDENG/bPxRficyEXAMPLEKEY
--region region	매핑 파일에서 조회하는 리전입니다. 조건: 자동 매핑 사용 시 필수입니다. 필수 항목 여부: 조건부 예: --region eu-west-1

옵션	설명
--no-mapping	<p>커널 및 RAM 디스크의 자동 매핑을 비활성화합니다.</p> <p>마이그레이션 동안 Amazon EC2는 매니페스트 파일에 있는 커널 및 RAM 디스크를 대상 리전으로 설계된 커널 및 RAM 디스크로 교체합니다. --no-mapping 파라미터를 지정하지 않으면, <code>ec2-migrate-bundle</code>에서 <code>DescribeRegions</code> 및 <code>DescribeImages</code> 작업을 사용하여 자동화된 매핑을 수행합니다.</p> <p>필수 항목 여부: 조건부</p> <p>조건: -a, -s 및 --region 옵션(자동 매핑에 사용됨)을 제공하지 않는 경우 필수.</p>
--ec2cert path	<p>이미지 매니페스트를 암호화하는 데 사용되는 Amazon EC2 X.509 퍼블릭 키 인증서의 경로입니다.</p> <p>us-gov-west-1 및 cn-north-1 리전은 기본값이 아닌 퍼블릭 키 인증서를 사용하며 해당 인증서의 경로를 이 옵션으로 지정해야 합니다. 인증서 경로는 AMI 도구의 설치 방법에 따라 다릅니다. Amazon Linux의 경우 인증서가 <code>/opt/aws/amitools/ec2/etc/ec2/amitools/</code>에 있습니다. AMI 도구 설치 (p. 86)의 ZIP 파일에서 AMI 도구를 설치했으면 인증서가 <code>\$EC2_AMITOOL_HOME/etc/ec2/amitools/</code>에 있습니다.</p> <p>기본값: 도구에 따라 다름</p> <p>필수 항목 여부: us-gov-west-1 및 cn-north-1 리전의 경우에만.</p> <p>예: <code>--ec2cert \$EC2_AMITOOL_HOME/etc/ec2/amitools/cert-ec2.pem</code></p>
--kernel kernel_id	<p>선택할 커널의 ID입니다.</p> <p>Important</p> <p>커널 및 RAM 디스크 대신 PV-GRUB를 사용하는 것이 좋습니다. 자세한 내용은 PV-GRUB를 참조하십시오.</p> <p>필수 항목 여부: 아니요</p> <p>예: <code>--kernel aki-ba3adfd3</code></p>
--ramdisk ramdisk_id	<p>선택할 RAM 디스크의 ID입니다.</p> <p>Important</p> <p>커널 및 RAM 디스크 대신 PV-GRUB를 사용하는 것이 좋습니다. 자세한 내용은 PV-GRUB를 참조하십시오.</p> <p>필수 항목 여부: 아니요</p> <p>예: <code>--ramdisk ari-badbdb00</code></p>
일반 옵션	대부분의 AMI 도구에 공통적인 옵션에 대한 자세한 내용은 AMI 도구의 일반 옵션 (p. 107) 을 참조하십시오.

결과

번들링 프로세스의 단계 및 상태를 설명하는 상태 메시지입니다.

예

이 예제에서는 `my-ami.manifest.xml` 매니페스트에 지정된 AMI를 복사합니다.

```
$ ec2-migrate-manifest --manifest my-ami.manifest.xml --cert cert-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem --privatekey pk-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem --region eu-west-1

Backing up manifest...
Successfully migrated my-ami.manifest.xml It is now suitable for use in eu-west-1.
```

ec2-unbundle

설명

인스턴스 스토어 기반 Linux AMI에서 번들을 다시 생성합니다.

구문

```
ec2-unbundle -k path -m path [-s source_directory] [-d destination_directory]
```

옵션

옵션	설명
<code>-k, --privatekey <i>path</i></code>	PEM 인코딩된 RSA 키 파일의 경로입니다. 필수 항목 여부: 예 예: <code>-k \$HOME/pk-234242example.pem</code>
<code>-m, --manifest <i>path</i></code>	매니페스트 파일 경로입니다. 필수 항목 여부: 예 예: <code>-m /var/spool/my-first-bundle/Manifest</code>
<code>-s, --source <i>source_directory</i></code>	번들이 포함된 디렉터리입니다. 기본값: 현재 디렉터리입니다. 필수 항목 여부: 아니요 예: <code>-s /tmp/my-bundled-image</code>
<code>-d, --destination <i>destination_directory</i></code>	AMI 번들을 해제해 넣을 디렉터리입니다. 대상 디렉터리가 있어야 합니다. 기본값: 현재 디렉터리입니다. 필수 항목 여부: 아니요 예: <code>-d /tmp/my-image</code>
일반 옵션	대부분의 AMI 도구에 공통적인 옵션에 대한 자세한 내용은 AMI 도구의 일반 옵션 (p. 107) 을 참조하십시오.

예

이 Linux 및 UNIX 예제는 `image.manifest.xml` 파일에 지정된 AMI 번들을 해제합니다.

```
$ mkdir unbundled
$ ec2-unbundle -m mybundle/image.manifest.xml -k pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -s
mybundle -d unbundled
$ ls -l unbundled
total 1025008
-rw-r--r-- 1 root root 1048578048 Aug 25 23:46 image.img
```

결과

번들 해제 프로세스의 다양한 단계를 나타내는 상태 메시지가 표시됩니다.

ec2-upload-bundle

설명

인스턴스 스토어 기반 Linux AMI 번들을 Amazon S3로 업로드하고 업로드된 객체에서 적절한 ACL을 설정합니다. 자세한 내용은 [인스턴스 스토어 지원 Linux AMI 생성](#)을 참조하십시오.

구문

```
ec2-upload-bundle -b bucket -a access_key_id -s secret_access_key [-t token] -m path [--url
url] [--region region] [--sigv version] [--acl acl] [-d directory] [--part part] [--retry]
[--skipmanifest]
```

옵션

옵션	설명
<code>-b, --bucket <i>bucket</i></code>	번들 이름을 저장할 Amazon S3 버킷 이름으로, 선택 항목인 ' <code>/</code> -delimited 경로 접두사가 붙기도 합니다. 버킷이 없을 경우에 생성됩니다(해당 버킷 이름을 사용할 수 있는 경우). 필수 항목 여부: 예 예: <code>-b myawsbucket/bundles/ami-001</code>
<code>-a, --access-key <i>access_key_id</i></code>	사용자의 AWS 액세스 키 ID입니다. 이 옵션의 값을 지정하기 전에, AWS 액세스 키 관리 모범 사례 의 지침을 검토하고 따르십시오. 필수 항목 여부: 예 예: <code>-a AKIAIOSFODNN7EXAMPLE</code>
<code>-s, --secret-key <i>secret_access_key</i></code>	AWS 보안 액세스 키입니다. 이 옵션의 값을 지정하기 전에, AWS 액세스 키 관리 모범 사례 의 지침을 검토하고 따르십시오. 필수 항목 여부: 예 예: <code>-s wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY</code>
<code>-t, --delegation-token <i>token</i></code>	AWS 요청에 함께 전달되는 위임 토큰입니다. 자세한 내용은 Using Temporary Security Credentials 를 참조하십시오. 필수 항목 여부: 임시 보안 자격 증명을 사용하는 경우에만

옵션	설명
	<p>기본값: AWS_DELEGATION_TOKEN 환경 변수 값(설정된 경우). 예: -t AQoDYXdzEJr...<remainder of security token></p>
-m, --manifest path	<p>매니페스트 파일 경로입니다. 매니페스트 파일은 번들링 프로세스 중 생성되며 번들이 포함된 디렉터리에 있습니다.</p> <p>필수 항목 여부: 예</p> <p>예: -m image.manifest.xml</p>
--url url	<p>사용되지 않음. 버킷이 EU 위치(eu-west-1 제외)로 제한되어 있지 않은 경우 대신 --region 옵션을 사용합니다. --location 플래그는 이러한 특정 위치 제한을 적용하는 유일한 방법입니다.</p> <p>Amazon S3 엔드포인트 서비스 URL입니다.</p> <p>기본값: https://s3.amazonaws.com/</p> <p>필수 항목 여부: 아니요</p> <p>예: --url https://s3.example.com</p>
--region region	<p>대상 Amazon S3 버킷의 요청 서명에 사용할 리전입니다.</p> <ul style="list-style-type: none"> 버킷이 없고 리전을 지정하지 않는 경우 이 도구는 (us-east-1에서) 위치 제한 없이 버킷을 만듭니다. 버킷이 없고 리전을 지정하는 경우 이 도구는 지정된 리전에 버킷을 만듭니다. 버킷이 있고 리전을 지정하지 않는 경우 이 도구는 버킷의 위치를 사용합니다. 버킷이 있고 us-east-1을 리전으로 지정하는 경우 이 도구는 오류 메시지 없이 버킷의 실제 위치를 사용하며 기존에 일치하는 파일을 덮어씁니다. 버킷이 있고 버킷의 실제 위치와 일치하지 않는 리전(us-east-1 이외)을 지정하는 경우 도구가 종료되고 오류가 발생합니다. <p>버킷이 EU 위치(eu-west-1 제외)로 제한된 경우 대신 --location 플래그를 사용합니다. --location 플래그는 이러한 특정 위치 제한을 적용하는 유일한 방법입니다.</p> <p>기본값: us-east-1</p> <p>필수 항목 여부: 조건부</p> <p>조건: 서명 버전 4를 사용하는 경우 필수</p> <p>예: --region eu-west-1</p>

옵션	설명
--sigv version	<p>요청 서명 시 사용하는 서명 버전입니다.</p> <p>유효한 값: 2 4</p> <p>기본값: 4</p> <p>필수 항목 여부: 아니요</p> <p>예: --sigv 2</p>
--acl acl	<p>번들링된 이미지의 액세스 제어 목록 정책입니다.</p> <p>유효한 값: public-read aws-exec-read</p> <p>기본값: aws-exec-read</p> <p>필수 항목 여부: 아니요</p> <p>예: --acl public-read</p>
-d, --directory directory	<p>번들링된 AMI 파트가 포함된 디렉터리입니다.</p> <p>기본값: 매니페스트 파일이 포함된 디렉터리(-m 옵션 참조)입니다.</p> <p>필수 항목 여부: 아니요</p> <p>예: -d /var/run/my-bundle</p>
--part part	<p>지정된 파트와 모든 후속 파트의 업로드를 시작합니다.</p> <p>필수 항목 여부: 아니요</p> <p>예: --part 04</p>
--retry	<p>모든 Amazon S3 오류에서 작업당 최대 5회 자동으로 재시도 합니다.</p> <p>필수 항목 여부: 아니요</p> <p>예: --retry</p>
--skipmanifest	<p>매니페스트를 업로드하지 않습니다.</p> <p>필수 항목 여부: 아니요</p> <p>예: --skipmanifest</p>

옵션	설명
--location location	<p>사용되지 않음. 버킷이 EU 위치(eu-west-1 제외)로 제한되어 있지 않은 경우 대신 --region 옵션을 사용합니다. --location 플래그는 이러한 특정 위치 제한을 적용하는 유일한 방법입니다.</p> <p>대상 Amazon S3 버킷의 위치 제한입니다. 버킷이 있고 버킷의 실제 위치와 일치하지 않는 위치를 지정하는 경우 도구가 종료되고 오류가 발생합니다. 버킷이 있고 위치를 지정하지 않는 경우 이 도구는 버킷의 위치를 사용합니다. 버킷이 없고 위치를 지정하는 경우 이 도구는 지정된 리전에 버킷을 만듭니다. 버킷이 없고 위치를 지정하지 않는 경우 이 도구는 (us-east-1에서) 위치 제한 없이 버킷을 만듭니다.</p> <p>기본값: --region이 지정된 경우 위치는 여기에 지정된 리전으로 설정됩니다. --region이 지정되지 않은 경우 위치는 기본적으로 us-east-1로 설정됩니다.</p> <p>필수 항목 여부: 아니요</p> <p>예: --location eu-west-1</p>
일반 옵션	대부분의 AMI 도구에 공통적인 옵션에 대한 자세한 내용은 AMI 도구의 일반 옵션 (p. 107) 을 참조하십시오.

결과

Amazon EC2에서 업로드 프로세스의 단계 및 상황을 나타내는 상태 메시지를 표시합니다.

예

이 예제에서는 `image.manifest.xml` 매니페스트에서 지정한 번들을 업로드합니다.

```
$ ec2-upload-bundle -b myawsbucket/bundles/bundle_name -m image.manifest.xml -a your_access_key_id -s your_secret_access_key
Creating bucket...
Uploading bundled image parts to the S3 bucket myawsbucket ...
Uploaded image.part.00
Uploaded image.part.01
Uploaded image.part.02
Uploaded image.part.03
Uploaded image.part.04
Uploaded image.part.05
Uploaded image.part.06
Uploaded image.part.07
Uploaded image.part.08
Uploaded image.part.09
Uploaded image.part.10
Uploaded image.part.11
Uploaded image.part.12
Uploaded image.part.13
Uploaded image.part.14
Uploading manifest ...
Uploaded manifest.
Bundle upload completed.
```

AMI 도구의 일반 옵션

이 섹션에서 설명하는 대부분의 명령은 다음 표에서 설명하는 선택적 파라미터 집합을 수락합니다.

옵션	설명
--help, -h	도움말 메시지를 표시합니다.
--version	버전 및 저작권 통지를 표시합니다.
--manual	수동 입력 항목을 표시합니다.
--batch	배치 모드에서 실행되며 대화형 메시지를 표시하지 않습니다.
--debug	문제 해결 시 유용할 수 있는 디버깅 정보를 표시합니다.

서명 인증서 관리

이 섹션에서는 X.509 인증서라고도 하는 서명 인증서를 만들고 관리하는 방법에 대해 설명합니다. 이러한 인증서는 특정 AMI 도구 명령에 반드시 필요합니다.

Important

Amazon EC2는 원래 서비스 호출을 위한 SOAP 프로토콜을 지원했으며 SOAP 기반 호출은 요청의 디지털 서명을 위해 서명 인증서를 사용합니다. 하지만 Amazon EC2에서는 더 이상 SOAP가 지원되지 않으며([SOAP 요청 참조](#)) 대신 HTTP 쿼리 요청을 사용해야 합니다. 자세한 내용은 [API 요청 만들기](#)를 참조하십시오.

각 사용자는 자격 증명 교체를 위해 두 개의 인증서를 가질 수 있습니다.

Note

사용자가 직접 인증서를 나열 및 관리할 수 있는 권한을 부여할 수 있습니다. 자세한 내용은 IAM 사용 설명서에서 [사용자가 직접 암호, 액세스 키, 서명 인증서를 관리하도록 허용](#) 섹션을 참조하십시오.

항목

- [사용자 서명 인증서 만들기 \(p. 108\)](#)
- [사용자 서명 인증서 관리 \(p. 111\)](#)

사용자 서명 인증서 만들기

서명 인증서가 필요한 경우 우선 서명서를 가져온 다음 AWS로 업로드해야 합니다. 서명 인증서를 만들기 위한 Amazon EC2 API 작업은 없으므로 OpenSSL과 같은 타사 도구를 사용하여 사용자 서명 인증서를 만들어야 합니다.

Note

AWS Management Console에서 보안 자격 증명 페이지를 사용하여 X.509 인증서를 만드는 경우에도 해당 방법은 AWS 계정 루트 자격 증명에만 적용됩니다. 개별 Amazon EC2 사용자의 콘솔을 사용하여 생성된 인증서는 업로드할 수 없습니다. 대신 다음 섹션에서 설명하는 프로세스를 사용합니다.

서명 인증서를 생성하려면 다음 작업을 수행해야 합니다.

- OpenSSL을 설치 및 구성합니다.
- 프라이빗 키를 만듭니다.
- 프라이빗 키를 사용하여 인증서를 만듭니다.
- 인증서를 AWS로 업로드합니다.

OpenSSL 설치 및 구성

인증서를 생성하고 업로드하려면 SSL 및 TLS 프로토콜을 지원하는 도구가 필요합니다. OpenSSL은 RSA 토큰을 만들고 사용자의 프라이빗 키를 사용하여 서명하는 데 필요한 기본 암호화 기능을 제공하는 오픈 소스 도구입니다. 아직 OpenSSL을 설치하지 않은 경우 다음 지침을 따르십시오.

Linux 또는 UNIX에서 OpenSSL을 설치하려면

1. [OpenSSL: Source, Tarballs](http://www.openssl.org/source/)(<http://www.openssl.org/source/>)로 이동합니다.
2. 최신 소스를 다운로드하여 패키지를 생성합니다.

Windows에서 OpenSSL을 설치하려면

1. [바이너리](https://wiki.openssl.org/index.php/Binaries)(<https://wiki.openssl.org/index.php/Binaries>)로 이동합니다.
2. 적절한 Windows용 OpenSSL 옵션을 선택합니다.

새 페이지에 Windows 다운로드 링크가 표시됩니다.

3. 운영 체제에 아직 설치되지 않았다면 환경에 맞는 [Microsoft Visual C++ 2008 Redistributables] 링크를 선택하고 [Download]를 클릭합니다. [Microsoft Visual C++ 2008 Redistributable Setup Wizard]의 지시를 따릅니다.

Note

운영 체제에 Microsoft Visual C++ 2008 Redistributable 패키지가 설치되었는지 알 수 없는 경우 OpenSSL을 먼저 설치합니다. Microsoft Visual C++ 2008 Redistributable 패키지가 설치되지 않은 경우에는 OpenSSL 설치 관리자에서 오류를 표시합니다. 설치할 OpenSSL 버전에 해당하는 아키텍처(32비트 또는 64비트)를 설치해야 합니다.

4. Microsoft Visual C++ 2008 Redistributable 패키지를 설치한 후에는 환경에 맞는 OpenSSL 바이너리를 선택하고 파일을 로컬 위치에 저장합니다. [OpenSSL Setup Wizard]를 실행합니다.
5. [OpenSSL Setup Wizard] 지시에 따릅니다.

OpenSSL 명령을 사용하려면 OpenSSL이 설치된 위치 정보가 담기도록 운영 체제를 구성해야 합니다.

Linux 또는 Unix에서 OpenSSL을 구성하려면

1. 명령줄에서 `openssl_HOME` 변수를 OpenSSL 설치 위치로 설정합니다.

```
export OpenSSL_HOME=path_to_your_OpenSSL_installation
```

2. OpenSSL 설치가 포함되도록 경로를 설정합니다.

```
export PATH=$PATH:$OpenSSL_HOME/bin
```

Note

`export` 명령을 사용하여 변경한 환경 변수는 현재 세션에만 유효합니다. 쉘 구성 파일을 사용하여 설정하면 환경 변수의 영구 변경이 가능합니다. 자세한 내용은 운영 체제 설명서를 참조하십시오.

Windows에서 OpenSSL을 구성하려면

1. [Command Prompt] 창을 엽니다.
2. `openssl_HOME` 변수를 OpenSSL 설치 위치로 설정합니다.

```
set OpenSSL_HOME=path_to_your_OpenSSL_installation
```

3. OpenSSL_CONF 변수를 OpenSSL 설치에 있는 구성 파일 위치로 설정합니다.

```
set OpenSSL_CONF=path_to_your_OpenSSL_installation\bin\openssl.cfg
```

4. OpenSSL 설치가 포함되도록 경로를 설정합니다.

```
set Path=%Path%;%OpenSSL_HOME%\bin
```

Note

[Command Prompt] 창에서 변경한 Windows 환경 변수는 현재 명령줄 세션에만 유효합니다. 환경 변수를 시스템 속성으로 설정하면 환경 변수의 영구 변경이 가능합니다. 정확한 절차는 사용 중인 Windows 버전에 따라 좌우됩니다. 자세한 내용은 Windows 설명서를 참조하십시오.

Create a Private Key

사용자 서명 인증서를 생성할 때 사용하는 고유 프라이빗 키가 필요합니다.

프라이빗 키를 생성하려면

1. 명령줄에서 다음 구문의 `openssl genrsa` 명령을 사용합니다.

```
openssl genrsa 2048 > private-key.pem
```

`private-key.pem`에 원하는 파일 이름을 입력합니다. 예에서 2048은 2048비트 암호화를 나타냅니다. AWS는 1024비트와 4096비트 암호화도 지원합니다. 2048비트 또는 4096비트 RSA 키를 생성하는 것이 좋습니다.

2. 인증서를 사용하여 Auto Scaling, CloudWatch 또는 Elastic Load Balancing에 대한 CLI 명령을 인증하려는 경우 다음 명령을 사용하여 PKCS8 형식의 인증서를 만듭니다.

```
openssl pkcs8 -topk8 -nocrypt -inform PEM -in private-key.pem -out private-key-in-PKCS8-format.pem
```

사용자 서명 인증서 만들기

이제 사용자 서명 인증서를 만들 수 있습니다.

사용자 서명 인증서를 만들려면

- 다음 구문의 `openssl req` 명령을 사용합니다.

```
openssl req -new -x509 -nodes -sha256 -days 365 -key private-key.pem -outform PEM -out certificate.pem
```

`private-key.pem`의 경우 이전 절차에서 생성한 .pem 파일을 사용합니다. `certificate.pem`의 경우 인증서를 생성하려는 파일의 이름을 사용합니다. 인증서는 .pem 형식이어야 합니다. 이 예제에서는 보안을 위해 해시 알고리즘으로 SHA-256 또는 SHA-512를 사용하는 것이 좋습니다.

이 예제에서 `-days 365` 스위치는 인증서가 365일 적합함을 나타냅니다. 다른 스위치에 대한 자세한 내용을 보려면 명령줄에서 `openssl req -h`를 입력합니다.

OpenSSL에서 다음과 유사한 메시지를 표시합니다.

```
You are about to be asked to enter information that will be incorporated  
into your certificate request.  
What you are about to enter is what is called a Distinguished Name or a DN.  
There are quite a few fields but you can leave some blank.  
For some fields there will be a default value.  
If you enter '.', the field will be left blank.
```

사용자 서명 인증서(서버 인증서가 아님)를 만드는 종이므로 메시지가 표시되었을 때 모든 값을 비워둘 수 있습니다. 이러한 값은 인증 기관(CA)에서 서버 인증서를 인증하는 데 사용됩니다. 하지만 사용자 서명 인증서는 인증된 세션에서 업로드되므로 AWS는 추가 검증을 위해 인증서 정보가 필요하지 않으며 퍼블릭-프라이빗 키 쌍만 필요합니다.

.pem 파일에는 다음의 업로드 절차 중 복사하여 붙여 넣을 수 있는 인증서 값이 포함되어 있습니다.

사용자 서명 인증서 업로드

[upload-signing-certificate](#) AWS CLI 명령을 사용하여 서명 인증서를 업로드할 수 있습니다. 인증서를 업로드 하려는 사용자의 이름과 인증서 값이 있는 .pem 파일의 경로를 지정합니다.

```
aws iam upload-signing-certificate --user-name user-name --certificate-body file://path/to/certificate.pem
```

또는 [UploadSigningCertificate](#) IAM API 작업을 사용합니다.

Note

인증서 규모 때문에 서명 인증서를 업로드할 때는 POST 요청을 사용합니다.

사용자는 2개보다 많은 서명 인증서를 가질 수 없습니다.

사용자 서명 인증서 관리

AWS CLI를 사용해 서명 인증서를 관리할 수 있습니다.

액세스 키와 마찬가지로, 각 인증서의 상태는 Active 또는 Inactive입니다. 기본적으로 인증서를 업로드할 때 상태는 Active입니다. 인증서를 업로드할 때 반환되는 인증서 ID를 보관해둘 수 있습니다. 사용자 인증서의 ID를 나열할 수 있습니다. 인증서는 언제든지 삭제할 수 있습니다.

사용자에게 적용되는 인증서를 나열하려면 [list-signing-certificates](#) AWS CLI 명령을 사용하십시오.

```
aws iam list-signing-certificates --user-name user-name
```

사용자의 서명 인증서를 비활성화하거나 다시 활성화하려면 [update-signing-certificate](#) AWS CLI 명령을 사용하십시오. 다음 명령으로 인증서를 비활성화할 수 있습니다.

```
aws iam update-signing-certificate --certificate-id OFHPLP4ZULTHYPMSYEX704BEXAMPLE --status Inactive --user-name user-name
```

인증서를 삭제하려면 [delete-signing-certificate](#) AWS CLI 명령을 사용하십시오.

```
aws iam delete-signing-certificate --user-name user-name --certificate-id OFHPLP4ZULTHYPMSYEX704BEXAMPLE
```

또는 다음 IAM API 작업을 사용할 수 있습니다.

- [ListSigningCertificates](#)
- [UpdateSigningCertificate](#)
- [DeleteSigningCertificate](#)

인스턴스 스토어 지원 인스턴스에서 AMI 생성

다음은 인스턴스 스토어 기반 인스턴스에서 인스턴스 스토어 기반 AMI를 만드는 절차입니다. 시작하기 전에 먼저 [필수 조건 \(p. 85\)](#)을 읽으십시오.

항목

- [인스턴스 스토어 지원 Amazon Linux 인스턴스에서 AMI 생성 \(p. 112\)](#)
- [인스턴스 스토어 지원 Ubuntu 인스턴스에서 AMI 생성 \(p. 116\)](#)

인스턴스 스토어 지원 Amazon Linux 인스턴스에서 AMI 생성

이 섹션에서는 Amazon Linux 인스턴스에서 AMI를 생성하는 방법을 살펴봅니다. 다음 절차는 다른 Linux 배포를 실행하는 인스턴스에서는 작동하지 않을 수 있습니다. Ubuntu 관련 절차는 [인스턴스 스토어 지원 Ubuntu 인스턴스에서 AMI 생성 \(p. 116\)](#) 섹션을 참조하십시오.

Amazon EC2 AMI 도구 사용을 준비하려면(HVM 인스턴스에만 해당)

1. Amazon EC2 AMI 도구를 올바르게 부팅하려면 GRUB Legacy가 필요합니다. 다음 명령을 사용하여 GRUB을 설치합니다.

```
[ec2-user ~]$ sudo yum install -y grub
```

2. 다음 명령을 사용하여 파티션 관리 패키지를 설치합니다.

```
[ec2-user ~]$ sudo yum install -y gdisk kpartx parted
```

인스턴스 스토어 기반 Linux 인스턴스에서 AMI를 생성하려면

이 절차에서는 [사전 조건 \(p. 85\)](#)의 필수 조건을 충족한다고 가정합니다.

1. 인스턴스에 자격 증명을 업로드합니다. 이러한 자격 증명은 사용자와 Amazon EC2만 사용자의 AMI에 액세스할 수 있음을 보장하는 데 사용됩니다.
 - a. 다음과 같이 인스턴스에서 자격 증명에 대한 임시 디렉터리를 생성합니다.

```
[ec2-user ~]$ mkdir /tmp/cert
```

이렇게 하면 생성된 이미지에서 자격 증명을 제외할 수 있습니다.

- b. [scp \(p. 276\)](#) 등의 보안 복사 도구를 사용하여 컴퓨터의 X.509 인증서와 해당 프라이빗 키를 인스턴스의 /tmp/cert 디렉터리로 복사합니다. 다음 scp 명령의 -i *my-private-key.pem* 옵션은 X.509 프라이빗 키가 아니라 SSH를 사용하여 인스턴스에 연결하는 데 사용되는 프라이빗 키입니다. 예:

```
you@your_computer:~ $ scp -i my-private-key.pem /  
path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem /  
path/to/cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem ec2-  
user@ec2-203-0-113-25.compute-1.amazonaws.com:/tmp/cert/  
pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem 100% 717      0.7KB/s  00:00  
cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem 100% 685      0.7KB/s  00:00
```

또는, 이들은 일반 텍스트 파일이므로 텍스트 편집기에서 인증서와 키를 열고 내용을 /tmp/cert의 새 파일로 복사할 수 있습니다.

2. 인스턴스 안에서 [ec2-bundle-vol \(p. 91\)](#) 명령을 실행하여 Amazon S3로 업로드할 번들을 준비합니다. -e 옵션을 지정하여 자격 증명이 저장되어 있는 디렉터리를 제외해야 합니다. 기본적으로 번들 프로세스에는 중요 정보를 포함할 수 있는 파일이 제외됩니다. *.sw, *.swo, *.swp, *.pem, *.priv, *id_rsa*, *id_dsa* *.gpg, *.jks, */.ssh/authorized_keys, */.bash_history 등을 예로 들 수 있습니다. 이러한 파일을 모든 포함하려면 --no-filter 옵션을 사용합니다. 이러한 파일 중 일부만 포함하려면 --include 옵션을 사용합니다.

Important

기본적으로 AMI 번들링 프로세스에서는 루트 볼륨을 나타내는 /tmp 디렉터리에 압축 및 암호화된 파일 모음이 생성됩니다. /tmp에 사용 가능한 디스크 공간이 충분하지 않아서 번들을 저장할 수 없으면 -d [/path/to/bundle/storage](#) 옵션을 사용하여 번들을 저장할 다른 위치를 지정합니다. 인스턴스 중에는 /mnt 또는 /media/ephemeral0에 사용자가 사용할 수 있는 취약성 스토리지가 마운트된 인스턴스도 있으며, 새 Amazon EBS 볼륨을 [생성 \(p. 573\)](#), [연결 \(p. 576\)](#) 및 [마운트 \(p. 577\)](#)하여 번들을 저장할 수도 있습니다.

- a. ec2-bundle-vol 명령은 root로 실행해야 합니다. 대부분의 명령에 대해 sudo를 사용하여 승격된 권한을 얻을 수 있지만 이 경우 환경 변수를 유지하려면 sudo -E su를 실행해야 합니다.

```
[ec2-user ~]$ sudo -E su
```

이제 bash 프롬프트가 사용자를 루트 사용자로 식별하고 달러 기호가 해시 태그로 바뀌어 현재 위치가 루트 셸임을 표시합니다.

```
[root ec2-user]#
```

- b. AMI 번들을 생성하려면 다음 파라미터를 사용하여 [ec2-bundle-vol \(p. 91\)](#) 명령을 실행합니다:

-c

RSA 인증서 경로 및 파일 이름

-k

RSA 인증서 프라이빗 키 경로 및 파일 이름

--partition

파티션 유형: mbr, gpt 또는 none. HVM 인스턴스의 AMI는 이 항목이 없으면 부팅되지 않습니다.

-r

CPU 아키텍처: i386 또는 x86_64. arch 명령을 실행하여 이 항목을 확인할 수 있습니다.

-u

AWS 사용자 계정 ID

-e

생성된 이미지에서 제외할 디렉터리의 쉼표로 구분된 목록.

전달될 수

기본 디렉터리 /tmp에 번들을 수용할 공간이 충분한 경우 이것은 공간이 충분한 디렉터리의 경로를 제공합니다.

--ec2cert

이 파라미터는 다음 리전에서만 필요합니다. 중국(베이징) and AWS GovCloud (US). 이 리전에서는 각각 퍼블릭 키 인증서를 지정해야 합니다.

이 명령과 관련 옵션에 대한 자세한 내용은 [ec2-bundle-vol \(p. 91\)](#) 단원을 참조하십시오.

다음은 명령 샘플입니다.

```
[root ec2-user]# $EC2_AMITOOL_HOME/bin/ec2-bundle-vol -k /tmp/cert/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -c /tmp/cert/cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -u your_aws_account_id -r x86_64 -e /tmp/cert --partition gpt
```

이미지가 생성되는 데 몇 분 정도 걸릴 수 있습니다. 이 명령이 완료되면 /tmp(또는 기본값이 아닌) 디렉터리에 번들(image.manifest.xml과 여러 image.part.**xx** 파일)이 포함됩니다.

- c. root 셸을 종료합니다.

```
[root ec2-user]# exit
```

3. (선택 사항) AMI의 image.manifest.xml 파일에서 블록 디바이스 매핑을 편집합니다. 인스턴스 스토어 지원 AMI는 AMI가 생성되었으며 이러한 매핑이 image.manifest.xml 파일에 지정된 경우에만 블록 디바이스 매핑에 인스턴스 스토어 볼륨을 지정할 수 있습니다. 자세한 내용은 [블록 디바이스 매핑 \(p. 662\)](#) 섹션을 참조하십시오.

Note

이 단계는 AMI에서 하나 이상의 추가 인스턴스 스토어 볼륨을 추가하려는 경우에만 필요합니다.

- a. image.manifest.xml 파일의 백업을 만듭니다.

```
[ec2-user ~]$ sudo cp /tmp/image.manifest.xml /tmp/image.manifest.xml.bak
```

- b. 읽고 편집하기 쉽도록 image.manifest.xml 파일의 서식을 다시 설정합니다.

```
[ec2-user ~]$ sudo xmllint --format /tmp/image.manifest.xml.bak > sudo /tmp/image.manifest.xml
```

- c. 텍스트 편집기로 image.manifest.xml에서 블록 디바이스 매핑을 편집합니다. 아래 예는 **ephemeral1** 인스턴스 스토어 볼륨의 새 항목을 보여 줍니다.

```
<block_device_mapping>
  <mapping>
    <virtual>ami</virtual>
    <device>sda</device>
  </mapping>
  <mapping>
    <virtual>ephemeral0</virtual>
    <device>sdb</device>
  </mapping>
  <mapping>
    <virtual>ephemeral1</virtual>
    <device>sdc</device>
  </mapping>
  <mapping>
    <virtual>root</virtual>
    <device>/dev/sda1</device>
  </mapping>
```

```
</block_device_mapping>
```

d. `image.manifest.xml` 파일을 저장하고 텍스트 편집기를 종료합니다.

4. 번들을 Amazon S3로 업로드하려면 다음 파라미터를 사용하여 [ec2-upload-bundle \(p. 104\)](#) 명령을 실행합니다.

-b

S3 버킷의 위치: `my-s3-bucket/bundle_folder/bundle_name`. 버킷과 폴더 경로가 없을 경우 이 명령으로 생성된다는 점에 유의하십시오.

-m

`image.manifest.xml` 경로. `## -d /path/to/bundle/storage`[Step 2 \(p. 113\)](#)와 함께 경로를 지정한 경우 이 파라미터를 포함하여 동일 경로를 사용합니다.

-a

AWS 계정 액세스 키 ID

-s

AWS 계정 보안 액세스 키

--region

미국 동부(버지니아 북부) 이외 리전에서 AMI를 등록하려면 `--region` 옵션이 있는 목표 리전과 목표 리전에 이미 존재하는 버킷 경로 또는 목표 리전에 생성할 수 있는 고유 버킷 리전을 모두 지정해야 합니다.

이 명령 및 이와 관련하여 사용 가능한 옵션에 대한 자세한 내용은 [ec2-upload-bundle \(p. 104\)](#) 섹션을 참조하십시오.

다음은 명령 샘플입니다.

```
[ec2-user ~]$ ec2-upload-bundle -b my-s3-bucket/bundle_folder/bundle_name -m /tmp/image.manifest.xml -a your_access_key_id -s your_secret_access_key
```

5. (선택 사항) 번들을 Amazon S3에 업로드한 후에는 다음 `rm` 명령을 사용하여 인스턴스의 `/tmp` 디렉터리에서 번들을 제거할 수 있습니다.

Note

[Step 2 \(p. 113\)](#)에서 `-d /path/to/bundle/storage` 옵션과 함께 경로를 지정한 경우 `/tmp` 대신 아래 동일 경로를 사용합니다.

```
[ec2-user ~]$ sudo rm /tmp/image.manifest.xml /tmp/image.part.* /tmp/image
```

6. AMI를 등록하려면 다음 파라미터를 사용하여 `register-image` AWS CLI 명령을 실행합니다.

--image-location

`my-s3-bucket/bundle_folder/bundle_name/image.manifest.xml`

--name

AMI의 이름

--virtualization-type

가능한 값은 `hvm` 및 `paravirtual`입니다.

--region

[ec2-upload-bundle \(p. 104\)](#) 명령에 리전을 지정한 경우 이 명령에도 해당 리전을 다시 지정하십시오.

이 명령 및 이와 관련하여 사용 가능한 옵션에 대한 자세한 내용은 AWS Command Line Interface Reference의 [register-image](#) 섹션을 참조하십시오.

다음은 명령 샘플입니다.

```
[ec2-user ~]$ aws ec2 register-image --image-location my-s3-
bucket/bundle_folder/bundle_name/image.manifest.xml --name AMI_name --virtualization-
type hvm
```

인스턴스 스토어 지원 Ubuntu 인스턴스에서 AMI 생성

이 섹션에서는 Ubuntu Linux 인스턴스에서 AMI를 생성하는 방법을 살펴봅니다. 다음 절차는 다른 Linux 배포를 실행하는 인스턴스에서는 작동하지 않을 수 있습니다. Amazon Linux 관련 절차는 [인스턴스 스토어 지원 Amazon Linux 인스턴스에서 AMI 생성 \(p. 112\)](#) 섹션을 참조하십시오.

Amazon EC2 AMI 도구 사용을 준비하려면(HVM 인스턴스에만 해당)

Amazon EC2 AMI 도구를 올바르게 부팅하려면 GRUB Legacy가 필요합니다. 하지만 Ubuntu는 GRUB 2를 사용하도록 구성됩니다. 인스턴스에 GRUB Legacy가 사용되는지 확인하고 사용되지 않는 경우 설치하고 구성해야 합니다.

HVM 인스턴스에서도 AMI 도구가 올바르게 작동하려면 파티셔닝 도구를 설치해야 합니다.

1. 인스턴스에 GRUB Legacy(버전 0.9~~x~~ 이하)가 설치되어 있어야 합니다. GRUB Legacy가 존재하는지 확인하고 필요하면 설치합니다.
 - a. GRUB 설치의 버전을 확인합니다.

```
ubuntu:~$ grub-install --version
grub-install (GRUB) 1.99-21ubuntu3.10
```

이 예에서는 GRUB 버전이 0.9~~x~~ 이상이므로 GRUB Legacy를 설치해야 합니다. [Step 1.b \(p. 116\)](#) 항목으로 이동합니다. GRUB Legacy가 이미 존재하는 경우 [Step 2 \(p. 116\)](#)으로 건너뛸 수 있습니다.

- b. 다음 명령을 사용하여 grub 패키지를 설치합니다.

```
ubuntu:~$ sudo apt-get install -y grub
```

다음과 같이 인스턴스에서 GRUB Legacy가 사용되는지 확인합니다.

```
ubuntu:~$ grub --version
grub (GNU GRUB 0.97)
```

2. 배포용 패키지 관리자를 사용하여 다음 파티션 관리 패키지를 설치합니다.

- gdisk(일부 배포에서는 이 gptfdisk 패키지를 대신 호출할 수 있음)
- kpartx
- parted

다음 명령을 사용합니다.

```
ubuntu:~$ sudo apt-get install -y gdisk kpartx parted
```

3. 인스턴스용 커널 파라미터를 확인합니다.

```
ubuntu:~$ cat /proc/cmdline
BOOT_IMAGE=/boot/vmlinuz-3.2.0-54-virtual root=UUID=4f392932-ed93-4f8f-
aee7-72bc5bb6ca9d ro console=ttyS0 xen_emul_unplug=unnecessary
```

커널 및 루트 디바이스 파라미터인 `ro, console=ttyS0` 및 `xen_emul_unplug=unnecessary` 두에 이어지는 옵션을 기록해둡니다. 옵션이 이와 다를 수도 있습니다.

4. `/boot/grub/menu.lst`에서 커널 항목을 확인합니다.

```
ubuntu:~$ grep ^kernel /boot/grub/menu.lst
kernel /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro console=hvc0
kernel /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro single
kernel /boot/memtest86+.bin
```

`console` 파라미터가 `hvc0` 대신 `ttyS0`을 가리키고 있으며 `xen_emul_unplug=unnecessary` 파라미터가 없습니다. 앞에서 말했듯이, 옵션이 이와 다를 수도 있습니다.

5. 주로 사용하는 텍스트 편집기(예: vim 또는 nano)에서 `/boot/grub/menu.lst`를 편집하여 콘솔을 변경하고 앞에서 식별한 파라미터를 부팅 항목에 추가합니다.

```
title      Ubuntu 12.04.3 LTS, kernel 3.2.0-54-virtual
root       (hd0)
kernel     /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs
ro         console=ttyS0 xen_emul_unplug=unnecessary
initrd    /boot/initrd.img-3.2.0-54-virtual

title      Ubuntu 12.04.3 LTS, kernel 3.2.0-54-virtual (recovery mode)
root       (hd0)
kernel     /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro
single    console=ttyS0 xen_emul_unplug=unnecessary
initrd    /boot/initrd.img-3.2.0-54-virtual

title      Ubuntu 12.04.3 LTS, memtest86+
root       (hd0)
kernel     /boot/memtest86+.bin
```

6. 이제 커널 항목에 올바른 파라미터가 들어 있는지 확인합니다.

```
ubuntu:~$ grep ^kernel /boot/grub/menu.lst
kernel /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro console=ttyS0
xen_emul_unplug=unnecessary
kernel /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro single
console=ttyS0 xen_emul_unplug=unnecessary
kernel /boot/memtest86+.bin
```

7. (Ubuntu 14.04 이상에만 해당) Ubuntu 14.04부터는 `/boot/efi`에 마운트된 별도의 EFI 파티션과 GPT 파티션 테이블이 인스턴스 스토어 기반 Ubuntu AMI에 사용됩니다. `ec2-bundle-vol` 명령은 이 부팅 파티션을 번들링할 수 없으므로, 다음 예와 같이 EFI 파티션에 대한 `/etc/fstab` 항목을 주석으로 처리해야 합니다.

```
LABEL=cloudimg-rootfs   /          ext4  defaults        0 0
#LABEL=UEFI            /boot/efi    vfat   defaults        0 0
```

```
/dev/xvdb      /mnt      auto      defaults,nobootwait,comment=cloudconfig 0      2
```

인스턴스 스토어 기반 Linux 인스턴스에서 AMI를 생성하려면

이 절차에서는 [사전 조건 \(p. 85\)](#)의 필수 조건을 충족한다고 가정합니다.

1. 인스턴스에 자격 증명을 업로드합니다. 이러한 자격 증명은 사용자와 Amazon EC2만 사용자의 AMI에 액세스할 수 있음을 보장하는 데 사용됩니다.
 - a. 다음과 같이 인스턴스에서 자격 증명에 대한 임시 디렉터리를 생성합니다.

```
ubuntu:~$ mkdir /tmp/cert
```

이렇게 하면 생성된 이미지에서 자격 증명을 제외할 수 있습니다.

- b. [scp \(p. 276\)](#) 등의 보안 복사 도구를 사용하여 컴퓨터의 X.509 인증서와 프라이빗 키를 인스턴스의 `/tmp/cert` 디렉터리로 복사합니다. 다음 `scp` 명령의 `-i my-private-key.pem` 옵션은 X.509 프라이빗 키가 아니라 SSH를 사용하여 인스턴스에 연결하는 데 사용되는 프라이빗 키입니다. 예:

```
you@your_computer:~ $ scp -i my-private-key.pem /  
path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem /  
path/to/cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem ec2-  
user@ec2-203-0-113-25.compute-1.amazonaws.com:/tmp/cert/  
pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem 100% 717 0.7KB/s 00:00  
cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem 100% 685 0.7KB/s 00:00
```

또는, 이들은 일반 텍스트 파일이므로 텍스트 편집기에서 인증서와 키를 열고 내용을 `/tmp/cert`의 새 파일로 복사할 수 있습니다.

2. 인스턴스 안에서 [ec2-bundle-vol \(p. 91\)](#) 명령을 실행하여 Amazon S3로 업로드할 번들을 준비합니다. `-e` 옵션을 지정하여 자격 증명이 저장되어 있는 디렉터리를 제외해야 합니다. 기본적으로 번들 프로세스에는 중요 정보를 포함할 수 있는 파일이 제외됩니다. `*.sw, *.swo, *.swp, *.pem, *.priv, *id_rsa*, *id_dsa* *.gpg, *.jks, */.ssh/authorized_keys, */.bash_history` 등을 예로 들 수 있습니다. 이러한 파일을 모든 포함하려면 `--no-filter` 옵션을 사용합니다. 이러한 파일 중 일부만 포함하려면 `--include` 옵션을 사용합니다.

Important

기본적으로 AMI 번들링 프로세스에서는 루트 볼륨을 나타내는 `/tmp` 디렉터리에 압축 및 암호화된 파일 모음이 생성됩니다. `/tmp`에 사용 가능한 디스크 공간이 충분하지 않아서 번들을 저장할 수 없으면 `-d /path/to/bundle/storage` 옵션을 사용하여 번들을 저장할 다른 위치를 지정합니다. 인스턴스 중에는 `/mnt` 또는 `/media/ephemeral0`에 사용자가 사용할 수 있는 휘발성 스토리지가 마운트된 인스턴스도 있으며, 새 Amazon EBS 볼륨을 [생성 \(p. 573\)](#), [연결 \(p. 576\)](#) 및 [마운트 \(p. 577\)](#)하여 번들을 저장할 수도 있습니다.

- a. `ec2-bundle-vol` 명령은 `root`로 실행해야 합니다. 대부분의 명령에 대해 `sudo`를 사용하여 승격된 권한을 얻을 수 있지만 이 경우 환경 변수를 유지하려면 `sudo -E su`를 실행해야 합니다.

```
ubuntu:~$ sudo -E su
```

이제 bash 프롬프트가 사용자를 루트 사용자로 식별하고 달러 기호가 해시 태그로 바뀌어 현재 위치가 루트 셸임을 표시합니다.

```
root@ubuntu:~#
```

- b. AMI 번들을 생성하려면 다음 파라미터를 사용하여 [ec2-bundle-vol \(p. 91\)](#) 명령을 실행합니다.

-c

RSA 인증서 경로 및 파일 이름

-k

RSA 인증서 프라이빗 키 경로 및 파일 이름

--partition

파티션 유형: mbr, gpt 또는 none. Ubuntu 14.04 이상 HVM 인스턴스의 경우 부팅 명령을 제대로 번들링하려면 --partition mbr 플래그를 추가합니다. 그렇지 않으면 새로 생성된 AMI가 부팅되지 않습니다.

-r

CPU 아키텍처: i386 또는 x86_64. arch 명령을 실행하여 이 항목을 확인할 수 있습니다.

-u

AWS 사용자 계정 ID

-e

생성된 이미지에서 제외할 디렉터리의 쉼표로 구분된 목록.

전달될 수

기본 디렉터리 /tmp에 번들을 수용할 공간이 충분한 경우 이것은 공간이 충분한 디렉터리의 경로를 제공합니다.

이 명령 및 이와 관련하여 사용 가능한 옵션에 대한 자세한 내용은 [ec2-bundle-vol \(p. 91\)](#) 섹션을 참조하십시오.

다음은 명령 샘플입니다.

```
root@ubuntu:# $EC2_AMITOOL_HOME/bin/ec2-bundle-vol -k /tmp/  
cert/pk-HKZYKTAIG2ECMXY1BH3HXV4ZBEXAMPLE.pem -c /tmp/cert/cert-  
HKZYKTAIG2ECMXY1BH3HXV4ZBEXAMPLE.pem -u your_aws_account_id -r x86_64 -e /tmp/cert  
--partition gpt
```

이미지가 생성되는 데 몇 분 정도 걸릴 수 있습니다. 이 명령이 완료되면 tmp 디렉터리에 번들 (`image.manifest.xml`와 여러 `image.part.xx` 파일)이 포함됩니다.

c. root 셸을 종료합니다.

```
root@ubuntu:# exit
```

3. (선택 사항) AMI의 `image.manifest.xml` 파일에서 블록 디바이스 매핑을 편집합니다. 인스턴스 스토어 지원 AMI는 AMI가 생성되었으며 이러한 매핑이 `image.manifest.xml` 파일에 지정된 경우에만 블록 디바이스 매핑에 인스턴스 스토어 볼륨을 지정할 수 있습니다. 자세한 내용은 [블록 디바이스 매핑 \(p. 662\)](#) 섹션을 참조하십시오.

Note

이 단계는 AMI에서 하나 이상의 추가 인스턴스 스토어 볼륨을 추가하려는 경우에만 필요합니다.

a. `image.manifest.xml` 파일의 백업을 만듭니다.

```
ubuntu:~$ sudo cp /tmp/image.manifest.xml /tmp/image.manifest.xml.bak
```

- b. 읽고 편집하기 쉽도록 `image.manifest.xml` 파일의 서식을 다시 설정합니다.

```
ubuntu:~$ sudo xmllint --format /tmp/image.manifest.xml.bak > /tmp/  
image.manifest.xml
```

- c. 텍스트 편집기로 `image.manifest.xml`에서 블록 디바이스 매핑을 편집합니다. 아래 예는 `ephemeral1` 인스턴스 스토어 볼륨의 새 항목을 보여 줍니다.

```
<block_device_mapping>  
  <mapping>  
    <virtual>ami</virtual>  
    <device>sda</device>  
  </mapping>  
  <mapping>  
    <virtual>ephemeral0</virtual>  
    <device>sdb</device>  
  </mapping>  
  <mapping>  
    <virtual>ephemeral1</virtual>  
    <device>sdc</device>  
  </mapping>  
  <mapping>  
    <virtual>root</virtual>  
    <device>/dev/sda1</device>  
  </mapping>  
</block_device_mapping>
```

- d. `image.manifest.xml` 파일을 저장하고 텍스트 편집기를 종료합니다.

4. 번들을 Amazon S3로 업로드하려면 다음 파라미터를 사용하여 [ec2-upload-bundle \(p. 104\)](#) 명령을 실행합니다.

-b

S3 버킷의 위치: `my-s3-bucket/bundle_folder/bundle_name`. 버킷과 폴더 경로가 없을 경우 이 명령으로 생성된다는 점에 유의하십시오.

-m

`image.manifest.xml` 경로. [Step 2 \(p. 118\)](#)에서 -d `/path/to/bundle/storage`와 함께 경로를 지정한 경우 이 파라미터에 동일 경로를 사용합니다.

-a

AWS 계정 액세스 키 ID

-s

AWS 계정 보안 액세스 키

--region

미국 동부(버지니아 북부) 이외 리전에서 AMI를 등록하려면 --region 옵션이 있는 목표 리전과 목표 리전에 이미 존재하는 버킷 경로 또는 목표 리전에 생성할 수 있는 고유 버킷 리전을 모두 지정해야 합니다.

이 명령 및 이와 관련하여 사용 가능한 옵션에 대한 자세한 내용은 [ec2-upload-bundle \(p. 104\)](#) 섹션을 참조하십시오.

다음은 명령 샘플입니다.

```
ubuntu:~$ ec2-upload-bundle -b my-s3-bucket/bundle_folder/bundle_name -m /tmp/  
image.manifest.xml -a your_access_key_id -s your_secret_access_key
```

- (선택 사항) 번들을 Amazon S3에 업로드한 후에는 다음 rm 명령을 사용하여 인스턴스의 /tmp 디렉터리에서 번들을 제거할 수 있습니다.

Note

Step 2 (p. 118)에서 -d `/path/to/bundle/storage` 옵션과 함께 경로를 지정한 경우 /tmp 대신 아래 동일 경로를 사용합니다.

```
ubuntu:~$ sudo rm /tmp/image.manifest.xml /tmp/image.part.* /tmp/image
```

- AMI를 등록하려면 다음 파라미터를 사용하여 register-image AWS CLI 명령을 실행합니다.

매니페스트 경로

```
my-s3-bucket/bundle_folder/bundle_name/image.manifest.xml  
-n  
AMI의 이름  
--virtualization-type  
가능한 값은 hvm 및 paravirtual입니다.  
--region  
ec2-upload-bundle (p. 104) 명령에 리전을 지정한 경우 이 명령에도 해당 리전을 다시 지정하십시오.
```

이 명령 및 이와 관련하여 사용 가능한 옵션에 대한 자세한 내용은 AWS Command Line Interface Reference의 [register-image](#) 섹션을 참조하십시오.

다음은 명령 샘플입니다.

```
ubuntu:~$ aws ec2 register-image my-s3-bucket/bundle_folder/bundle_name/  
image.manifest.xml --name AMI_name --virtualization-type hvm
```

- (Ubuntu 14.04 이상에만 해당) /etc/fstab에서 EFI 항목의 주석 처리를 제거합니다. 그렇지 않으면 실행 중인 인스턴스를 재부팅할 수 없습니다.

인스턴스 스토어 기반 AMI를 Amazon EBS 기반 AMI로 변환

사용자 소유의 인스턴스 스토어 기반 Linux AMI를 Amazon EBS 기반 Linux AMI로 변환할 수 있습니다.

Important

인스턴스 스토어 기반 Windows AMI는 Amazon EBS 기반 Windows AMI로 변환할 수 없으며 본인 소유가 아닌 AMI도 변환할 수 없습니다.

인스턴스 스토어 기반 AMI를 Amazon EBS 기반 AMI로 변환하려면

- Amazon EBS 기반 AMI에서 Amazon Linux 인스턴스를 시작합니다. 자세한 내용은 [인스턴스 시작하기 \(p. 265\)](#) 섹션을 참조하십시오. Amazon Linux 인스턴스에는 AWS CLI 및 AMI 도구가 미리 설치되어 있습니다.
- 인스턴스 스토어 기반 AMI를 번들링하는 데 사용한 X.509 프라이빗 키를 인스턴스로 업로드합니다. 이 키는 사용자와 Amazon EC2만 사용자의 AMI에 액세스할 수 있음을 보장하는 데 사용됩니다.
 - 다음과 같이 인스턴스에서 X.509 프라이빗 키에 대한 임시 디렉터리를 생성합니다.

```
[ec2-user ~]$ mkdir /tmp/cert
```

- b. [scp \(p. 276\)](#) 등의 보안 복사 도구를 사용하여 컴퓨터의 X.509 프라이빗 키를 인스턴스의 /tmp/cert 디렉터리로 복사합니다. 다음 명령의 *my-private-key* 파라미터는 SSH를 사용하여 인스턴스에 연결하는 데 사용되는 프라이빗 키입니다. 예:

```
you@your_computer:~ $ scp -i my-private-key.pem /path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem ec2-user@ec2-203-0-113-25.compute-1.amazonaws.com:/tmp/cert/
pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem 100% 717      0.7KB/s  00:00
```

3. AWS 액세스 키와 보안 키에 대한 환경 변수를 설정합니다.

```
[ec2-user ~]$ export AWS_ACCESS_KEY_ID=your_access_key_id
[ec2-user ~]$ export AWS_SECRET_ACCESS_KEY=your_secret_access_key
```

4. 새 AMI용 Amazon EBS 볼륨을 준비합니다.

- a. [create-volume](#) 명령을 사용하여 인스턴스와 동일한 가용 영역에 빈 Amazon EBS 볼륨을 생성합니다. 명령 출력의 볼륨 ID를 기록해둡니다.

Important

이 Amazon EBS 볼륨은 크기가 원본 인스턴스 스토어 루트 볼륨보다 크거나 같아야 합니다.

```
[ec2-user ~]$ aws ec2 create-volume --size 10 --region us-west-2 --availability-zone us-west-2b
```

- b. [attach-volume](#) 명령을 사용하여 이 볼륨을 Amazon EBS 기반 인스턴스에 연결합니다.

```
[ec2-user ~]$ aws ec2 attach-volume --volume-id volume_id --instance-id instance_id --device /dev/sdb --region us-west-2
```

5. 번들용 폴더를 생성합니다.

```
[ec2-user ~]$ mkdir /tmp/bundle
```

6. ##### #AMI# ## AMI# ### /tmp/bundle[ec2-download-bundle \(p. 98\)](#)로 다운로드합니다.

```
[ec2-user ~]$ ec2-download-bundle -b my-s3-bucket/bundle_folder/bundle_name -m image.manifest.xml -a $AWS_ACCESS_KEY -s $AWS_SECRET_KEY --privatekey /path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -d /tmp/bundle
```

7. [ec2-unbundle \(p. 103\)](#) 명령을 사용하여 번들에서 이미지 파일을 다시 구성합니다.

- a. 디렉터리를 번들 폴더로 변경합니다.

```
[ec2-user ~]$ cd /tmp/bundle/
```

- b. [ec2-unbundle \(p. 103\)](#) 명령을 실행합니다.

```
[ec2-user bundle]$ ec2-unbundle -m image.manifest.xml --privatekey /path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem
```

8. 번들링되지 않은 이미지의 파일을 새 Amazon EBS 볼륨으로 복사합니다.

```
[ec2-user bundle]$ sudo dd if=/tmp/bundle/image of=/dev/sdb bs=1M
```

- 번들링되지 않은 새 파티션용 볼륨을 검색합니다.

```
[ec2-user bundle]$ sudo partprobe /dev/sdb1
```

- 블록 디바이스를 나열하여 마운트할 디바이스 이름을 찾습니다.

```
[ec2-user bundle]$ lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
/dev/sda    202:0   0   8G  0 disk
##/dev/sda1 202:1   0   8G  0 part /
/dev/sdb    202:80  0  10G  0 disk
##/dev/sdb1 202:81  0  10G  0 part
```

이 예에서는 마운트할 파티션이 /dev/sdb1이지만, 디바이스 이름이 이와 다를 수 있습니다. 볼륨이 파티셔닝되지 않은 경우 마운트할 디바이스는 /dev/sdb(디바이스 파티션 끝 숫자가 없음)와 비슷할 것입니다.

- 새 Amazon EBS 볼륨에 대한 마운트 지점을 생성하고 볼륨을 마운트합니다.

```
[ec2-user bundle]$ sudo mkdir /mnt/ebs
[ec2-user bundle]$ sudo mount /dev/sdb1 /mnt/ebs
```

- 주로 사용하는 텍스트 편집기(예: vim 또는 nano)를 사용하여 EBS 볼륨의 /etc/fstab 파일을 열고 인스턴스 스토어(후발성) 볼륨에 대한 항목을 모두 제거합니다. Amazon EBS 볼륨이 /mnt/ebs에 마운트되므로 fstab 파일은 /mnt/ebs/etc/fstab에 있습니다.

```
[ec2-user bundle]$ sudo nano /mnt/ebs/etc/fstab
#
LABEL=/      /          ext4    defaults,noatime 1 1
tmpfs       /dev/shm    tmpfs   defaults        0 0
devpts      /dev/pts    devpts  gid=5,mode=620 0 0
sysfs       /sys        sysfs   defaults        0 0
proc        /proc       proc    defaults        0 0
/dev/sdb     /media/ephemeral0  auto    defaults,comment=cloudconfig 0
2
```

이 예에서는 마지막 줄을 제거해야 합니다.

- 볼륨 마운트를 해제하고 인스턴스에서 볼륨을 분리합니다.

```
[ec2-user bundle]$ sudo umount /mnt/ebs
[ec2-user bundle]$ aws ec2 detach-volume --volume-id volume_id --region us-west-2
```

- 다음과 같이 새 Amazon EBS 볼륨에서 AMI를 생성합니다.

- 새 Amazon EBS 볼륨의 스냅샷을 생성합니다.

```
[ec2-user bundle]$ aws ec2 create-snapshot --region us-west-2 --description
"your_snapshot_description" --volume-id volume_id
```

- 스냅샷이 완전한지 확인합니다.

```
[ec2-user bundle]$ aws ec2 describe-snapshots --region us-west-2 --snapshot-
id snapshot_id
```

- c. `describe-images` 명령을 사용하여 원래의 AMI에 사용된 프로세스 아키텍처, 가상화 유형 및 커널 이미지(aki)를 식별합니다. 이 단계의 경우 원본 인스턴스 스토어 기반 AMI의 AMI ID가 필요합니다.

```
[ec2-user bundle]$ aws ec2 describe-images --region us-west-2 --image-id ami-id --output text
IMAGES x86_64 amazon/amzn-ami-pv-2013.09.2.x86_64-s3 ami-8ef297be amazon available
public machine aki-fc8f11cc instance-store paravirtual xen
```

이 예에서는 아키텍처가 `x86_64`이고 커널 이미지 ID가 `aki-fc8f11cc`입니다. 다음 단계에서는 이들 값을 사용합니다. 위 명령의 출력에 `ari` ID도 나열되면 이 ID도 기록해둡니다.

- d. 새 Amazon EBS 볼륨의 스냅샷 ID와 이전 단계의 값을 사용하여 새 AMI를 등록합니다. 이전 명령 출력에 `ari` ID가 나열된 경우, `--ramdisk-id ari_id`를 사용하여 이 ID를 다음 명령에 포함합니다.

```
[ec2-user bundle]$ aws ec2 register-image --region us-west-2 --
name your_new_ami_name --block-device-mappings Ebs={SnapshotId=snapshot_id} --
virtualization-type hvm --architecture x86_64 --kernel-id aki-fc8f11cc
```

15. (선택 사항) 새 AMI에서 인스턴스를 시작할 수 있음을 테스트한 후에는 이 절차용으로 생성한 Amazon EBS 볼륨을 삭제할 수 있습니다.

```
$ aws ec2 delete-volume --volume-id volume_id
```

암호화된 스냅샷이 있는 AMI

Amazon EBS 스냅샷의 지원을 받는 AMI에서는 Amazon EBS 암호화를 활용할 수 있습니다. 데이터 볼륨과 루트 볼륨 모두의 스냅샷을 암호화하고 AMI에 연결할 수 있습니다.

암호화된 볼륨이 있는 EC2 인스턴스는 다른 인스턴스와 동일한 방법으로 AMI에서 시작됩니다.

`copyImage` 작업을 사용하면 암호화되지 않은 스냅샷이 있는 AMI에서 암호화된 스냅샷이 있는 AMI를 생성할 수 있습니다. 기본적으로 `CopyImage`에서는 대상 복사본을 생성할 때 원본 스냅샷의 암호화 상태를 유지합니다. 복사 프로세스의 파라미터를 구성하여 대상 스냅샷을 암호화할 수도 있습니다.

기본 AWS Key Management Service 고객 마스터 키(CMK) 또는 지정한 사용자 지정 키를 사용하여 스냅샷을 암호화할 수 있습니다. 어느 경우든 선택한 키에 대한 사용 권한이 있어야 합니다. 암호화된 스냅샷이 있는 AMI를 사용하는 경우 `CopyImage` 작업 중에 다른 암호화 키를 사용하여 다시 암호화할 수 있습니다. `CopyImage`에서는 한 번에 하나의 키만 수락하고 루트 또는 데이터에 상관없이 이미지의 모든 스냅샷을 해당 키로 암호화합니다. 여러 키로 암호화된 스냅샷을 포함하는 AMI를 수동으로 생성할 수 있습니다.

암호화된 스냅샷이 있는 AMI 생성에 대한 지원을 받으려면 Amazon EC2 콘솔, Amazon EC2 API 또는 AWS CLI를 이용합니다.

`CopyImage`의 암호화 파라미터는 AWS KMS를 사용할 수 있는 모든 리전에서 사용 가능합니다.

암호화된 EBS 스냅샷을 포함하는 AMI 시나리오

AMI를 복사하고 동시에 AWS Management Console 또는 명령줄을 사용하여 연결된 EBS 스냅샷을 암호화 할 수 있습니다.

암호화된 데이터 스냅샷이 있는 AMI 복사

이 시나리오에서 EBS 지원 AMI에는 1단계에 표시된 것처럼 암호화되지 않은 루트 스냅샷과 암호화된 데이터 스냅샷이 있습니다. `CopyImage` 작업은 2단계에서 암호화 파라미터를 사용하지 않고 호출됩니다. 따라서

각 스냅샷의 암호화 상태가 유지되므로 3단계의 대상 AMI 또한 암호화되지 않은 루트 스냅샷과 암호화된 데이터 스냅샷에 의해 지원됩니다. 스냅샷에 동일한 데이터가 포함되어 있지만 두 스냅샷은 서로 구분되므로 두 AMI 모두의 스냅샷에 대한 스토리지 비용과 각 AMI에서 시작하는 인스턴스에 대한 비용이 발생합니다.

단순 복사(예: Amazon EC2 콘솔 또는 명령줄을 사용하여 복사)를 수행할 수 있습니다. 자세한 내용은 [AMI 복사 \(p. 126\)](#) 섹션을 참조하십시오.

암호화된 루트 스냅샷에서 지원되는 AMI 복사

이 시나리오에서 Amazon EBS 지원 AMI에는 1단계에 표시된 것처럼 암호화된 루트 스냅샷이 있습니다. `copyImage` 작업은 2단계에서 암호화 파라미터를 사용하지 않고 호출됩니다. 따라서 스냅샷의 암호화 상태가 유지되므로 3단계의 대상 AMI 또한 암호화된 루트 스냅샷에 의해 지원됩니다. 루트 스냅샷에 동일한 시스템 데이터가 포함되어 있지만 두 스냅샷은 서로 구분되므로 두 AMI 모두의 스냅샷에 대한 스토리지 비용과, 각 AMI에서 시작하는 인스턴스에 대한 비용이 발생합니다.

단순 복사(예: Amazon EC2 콘솔 또는 명령줄을 사용하여 복사)를 수행할 수 있습니다. 자세한 내용은 [AMI 복사 \(p. 126\)](#) 섹션을 참조하십시오.

암호화되지 않은 AMI에서 암호화된 루트 스냅샷이 있는 AMI 만들기

이 시나리오에서 Amazon EBS 지원 AMI에는 1단계에 표시된 암호화되지 않은 루트 스냅샷이 있고, 3단계에 표시된 암호화된 루트 스냅샷을 사용하여 AMI가 생성됩니다. 2단계의 `copyImage` 작업은 선택한 CMK를 포함하여 두 암호화 파라미터를 사용하여 호출됩니다. 따라서 루트 스냅샷의 암호화 상태가 변경되므로, 대상 AMI는 원본 스냅샷과 동일한 데이터를 포함하는 루트 스냅샷에 의해 지원되지만 지정된 키를 사용하여 암호화됩니다. 두 AMI 모두의 스냅샷에 대한 스토리지 비용과 각 AMI에서 시작되는 인스턴스에 대한 비용이 발생합니다.

복사 및 암호화 작업(예: Amazon EC2 콘솔 또는 명령줄을 사용하여 작업)을 수행할 수 있습니다. 자세한 내용은 [AMI 복사 \(p. 126\)](#) 섹션을 참조하십시오.

실행 중인 인스턴스에서 암호화된 루트 스냅샷이 있는 AMI 만들기

이 시나리오에서는 실행 중인 EC2 인스턴스에서 AMI를 만듭니다. 1단계의 실행 중인 인스턴스에는 암호화된 루트 볼륨이 있고 3단계에서 만든 AMI에는 원본 볼륨과 동일한 키로 암호화된 루트 스냅샷이 있습니다. `CreateImage` 작업은 암호화의 존재 여부에 상관없이 동일하게 동작합니다.

Amazon EC2 콘솔 또는 명령줄을 사용하여 암호화된 볼륨을 포함하거나 포함하지 않는 실행 중인 Amazon EC2 인스턴스에서 AMI를 만들 수 있습니다. 자세한 내용은 [Amazon EBS 지원 Linux AMI 생성 \(p. 81\)](#) 섹션을 참조하십시오.

각 암호화된 스냅샷에 대해 고유한 CMK를 사용하여 AMI 만들기

이 시나리오는 키 #1로 암호화된 루트 볼륨 스냅샷에서 지원되는 AMI로 시작하여 키 #2 및 키 #3으로 암호화된 2개의 추가 데이터 볼륨 스냅샷이 연결된 AMI로 끝납니다. `copyImage` 작업에서는 여러 암호화 키를 단일 작업에 적용할 수 없습니다. 하지만, 서로 다른 키로 암호화된 여러 볼륨이 연결된 인스턴스에서 AMI를 만들 수 있습니다. 결과 AMI에는 해당 키로 암호화된 스냅샷이 있고 이 새 AMI에서 시작되는 인스턴스에도 해당 키로 암호화된 볼륨이 있습니다.

이 예제 절차 단계는 다음 다이어그램과 일치합니다.

1. 키 #1로 암호화된 볼륨 #1(루트) 스냅샷에서 지원하는 원본 AMI로 시작합니다.

2. 원본 AMI에서 EC2 인스턴스를 시작합니다.
3. 각각 키 #2 및 키 #3으로 암호화된 볼륨 #2(데이터) 및 볼륨 #3(데이터) EBS 볼륨을 만듭니다.
4. 암호화된 데이터 볼륨을 EC2 인스턴스에 연결합니다.
5. 이제 EC2 인스턴스에서 각각 다른 키를 사용하는 암호화된 루트 볼륨 1개와 암호화된 데이터 볼륨 2개가 있습니다.
6. EC2 인스턴스에서 `CreateImage` 작업을 사용합니다.
7. 결과 대상 AMI에는 각각 다른 키를 사용하는 세 EBS 볼륨의 암호화된 스냅샷이 포함되어 있습니다.

Amazon EC2 콘솔 또는 명령줄을 사용하여 이 절차를 수행할 수 있습니다. 자세한 내용은 다음 주제를 참조 하십시오.

- [인스턴스 시작 \(p. 264\)](#)
- [Amazon EBS 지원 Linux AMI 생성 \(p. 81\)](#).
- [Amazon EBS 볼륨 \(p. 562\)](#)
- AWS Key Management Service Developer Guide에서 [AWS 키 관리](#)

AMI 복사

AWS Management Console, AWS 명령줄 도구 또는 SDK, Amazon EC2 API(모두 `copyImage` 작업을 지원 함)를 사용하여 AWS 리전 내부 또는 전체에서 Amazon 머신 이미지(AMI)를 복사할 수 있습니다. Amazon EBS 지원 AMI 및 인스턴스 스토어 지원 AMI를 모두 복사할 수 있습니다. 암호화된 스냅샷 및 암호화된 AMI를 통해 AMI를 복사할 수 있습니다.

원본 AMI를 복사하면 동일하지만 고유의 식별자로 구별되는 대상 AMI가 생성됩니다. Amazon EBS 지원 AMI의 경우, 동일하지만 구분된 대상 스냅샷으로 각 지원 스냅샷이 복사되도록 기본 설정되어 있습니다. (한 가지 예외는 스냅샷을 암호화하도록 선택하는 경우입니다.) 대상 AMI에 영향을 미치지 않고 원본 AMI를 변경하거나 다시 등록할 수 있습니다. 반대의 경우도 마찬가지입니다.

AMI 복사 시 부과되는 요금은 없습니다. 그러나 표준 스토리지 및 데이터 전송 요금은 적용됩니다.

AWS에서는 시작 권한, 사용자 정의 태그 또는 Amazon S3 버킷 권한이 원본 AMI에서 새 AMI로 복사되지 않습니다. 복사 작업이 완료된 후 시작 권한, 사용자 정의 태그 및 Amazon S3 버킷 권한을 새 AMI에 적용할 수 있습니다.

권한

IAM 사용자를 이용하여 인스턴스 스토어 지원 AMI를 복사하는 경우 사용자는 Amazon S3 권한 (`s3:CreateBucket`, `s3:GetBucketAcl`, `s3>ListAllMyBuckets`, `s3:GetObject`, `s3:PutObject` 및 `s3:PutObjectAcl`)을 가지고 있어야 합니다.

다음 예시 정책을 통해 사용자는 지정된 버킷의 AMI 원본을 지정된 리전에 복사할 수 있습니다.

```
{  
    "Version": "2016-12-09",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "s3>ListAllMyBuckets",  
            "Resource": [  
                "arn:aws:s3:::*"]  
        }  
    ]  
}
```

```
        ],
    },
    {
        "Effect": "Allow",
        "Action": "s3:GetObject",
        "Resource": [
            "arn:aws:s3:::ami-source-bucket/*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "s3:CreateBucket",
            "s3:GetBucketAcl",
            "s3:PutObjectAcl",
            "s3:PutObject"
        ],
        "Resource": [
            "arn:aws:s3:::amis-for-123456789012-in-us-east-1*"
        ]
    }
]
```

리전 간 AMI 복사

지리적으로 다른 리전 간에 AMI를 복사하면 다음과 같은 이점이 제공됩니다.

- 일관적인 글로벌 배포: 한 리전에서 다른 리전으로 AMI를 복사하면 동일한 AMI를 기반으로 하는 일관적인 인스턴스를 여러 리전에서 시작할 수 있습니다.
- 확장성: 사용자의 지역에 관계없이 요구 사항에 대응하는 세계적 규모의 애플리케이션을 보다 손쉽게 설계하고 구축할 수 있습니다.
- 성능: 애플리케이션을 분산하여 성능을 높이고 애플리케이션의 핵심 구성 요소를 사용자에게 보다 가까이 들 수 있습니다. 또한 인스턴스 유형이나 여타 AWS 서비스와 같은 리전별 기능을 활용할 수 있습니다.
- 고가용성: 여러 AWS 리전을 포괄하는 애플리케이션을 설계하고 배포하여 가용성을 높일 수 있습니다.

다음 다이어그램은 원본 AMI 및 다른 리전에 복사된 두 개의 AMI 간 관계와 각각에서 시작된 EC2 인스턴스를 보여 줍니다. AMI에서 인스턴스를 시작하는 경우 인스턴스는 AMI가 상주하는 동일한 리전에 상주합니다. 원본 AMI를 변경한 후 대상 리전의 AMI에 변경 내용을 반영하려면 원본 AMI를 대상 리전으로 다시 복사해야 합니다.

먼저 인스턴스 스토어 지원 AMI를 리전에 복사하는 경우 해당 리전에 복사된 AMI에 대한 Amazon S3 버킷이 생성됩니다. 해당 리전에 복사하는 인스턴스 스토어 지원 AMI는 모두 이 버킷에 저장됩니다. 버킷 이름 형식은 **amis-for-account-in-region-hash**를 따릅니다. 예를 들면 다음과 같습니다. **amis-for-123456789012-in-us-west-2-yhjmxvp6**.

필수 조건

AMI를 복사하기 전에 원본 AMI의 내용이 다른 리전에서 실행이 가능하도록 업데이트되었는지 확인해야 합니다. 예를 들어 데이터베이스 연결 문자열 등의 애플리케이션 구성 데이터가 적절한 리소스를 가리키도록 업데이트해야 합니다. 그렇지 않으면 대상 리전의 새 AMI에서 시작된 인스턴스가 여전히 원본 리전의 리소스를 사용하여 성능과 비용에 영향을 줄 수 있습니다.

한도

대상 리전은 동시에 복사할 수 있는 AMI 수가 50개로 제한되며, 이중 단일 원본 리전에서 복사할 수 있는 사본 수도 25개를 넘지 못합니다. 이 제한 수량에 대한 증가를 요청하려면 [Amazon EC2 서비스 제한 \(p. 688\)](#)에 문의하십시오.

교차 계정 AMI 복사

AMI를 다른 AWS 계정과 공유할 수 있습니다. AMI 공유는 AMI 소유권에 영향을 미치지 않습니다. 계정 소유에는 리전의 스토리지에 대한 요금이 부과됩니다. 자세한 내용은 [지정한 AWS 계정과 AMI 공유 \(p. 72\)](#) 섹션을 참조하십시오.

계정과 공유된 AMI를 복사하는 경우 계정에 있는 대상 AMI의 소유자가 됩니다. 원본 AMI 소유자에게는 표준 Amazon EBS 또는 Amazon S3 전송 요금이 청구되고 사용자에게는 대상 리전의 대상 AMI 스토리지에 대한 요금이 부과됩니다.

리소스 권한

사용자가 다른 계정에서 공유한 AMI를 복사하려면 원본 AMI 소유자는 AMI를 연결된 EBS 스냅샷이든 (Amazon EBS 지원 AMI의 경우) 연결된 S3 버킷이든(인스턴스 스토어 지원 AMI의 경우)든 지원하는 스토리지에 대한 읽기 권한을 사용자에게 부여해야 합니다.

제한

- 다른 계정에서 공유한 암호화된 AMI는 복사할 수 없습니다. 대신, 기본 스냅샷 및 암호화 키가 공유된 경우 자신의 키를 사용하여 스냅샷을 재암호화하는 동시에 복사할 수 있습니다. 복사된 스냅샷을 소유하게 되며 이것을 새로운 AMI로 등록할 수 있습니다.
- 다른 계정에서 공유한 암호화된 `billingProduct` 코드와 연결된 AMI는 복사할 수 없습니다. 여기에는 Windows AMI 및 AWS Marketplace의 AMI가 포함됩니다. `billingProduct` 코드로 공유 AMI를 복사하려면, 공유 AMI를 사용하여 계정에서 EC2 인스턴스를 시작한 다음 해당 인스턴스에서 AMI를 생성합니다. 자세한 내용은 [Amazon EBS 지원 Linux AMI 생성 \(p. 81\)](#) 섹션을 참조하십시오.

암호화 및 AMI 복사

AMI 복사 중 암호화는 Amazon EBS 지원 AMI에만 적용됩니다. 인스턴스 스토어 지원 AMI는 스냅샷에 의존하지 않기 때문에 AMI 복사를 사용하여 암호화 상태를 변경할 수 없습니다.

AMI 복사를 사용하여 암호화된 Amazon EBS 스냅샷에서 지원되는 새로운 AMI를 생성할 수 있습니다. AMI를 복사하는 동안 암호화를 허용하면 연결된 Amazon EBS 볼륨(루트 볼륨 포함)에서 찍힌 각 스냅샷은 지정한 키를 사용하여 암호화됩니다. 암호화된 스냅샷이 있는 AMI 사용에 대한 자세한 내용은 [암호화된 스냅샷이 있는 AMI \(p. 124\)](#)를 참조하십시오.

기본적으로 AMI의 지원 스냅샷은 원래 암호화 상태로 복사됩니다. 암호화되지 않은 스냅샷에서 지원되는 AMI를 복사하면 역시 암호화되지 않은 동일한 대상 스냅샷이 생성됩니다. 원본 AMI가 암호화된 스냅샷에서 지원되는 경우 원본을 복사하면 지정된 키로 암호화된 대상 스냅샷이 생성됩니다. 여러 스냅샷에서 지원되는 AMI를 복사하면 각 대상 스냅샷에서 원본 암호화 상태가 유지됩니다. 암호화된 스냅샷이 있는 AMI 복사에 대한 자세한 내용은 [암호화된 스냅샷이 있는 AMI \(p. 124\)](#)를 참조하십시오.

다음 표는 다양한 시나리오에 대한 암호화 지원을 보여 줍니다. 암호화되지 않은 스냅샷을 복사하여 암호화된 스냅샷을 생성할 수 있지만, 암호화된 스냅샷을 복사하여 암호화되지 않은 스냅샷을 생성할 수 없습니다.

시나리오	설명	지원
1	암호화되지 않음-암호화되지 않음	예
2	암호화됨-암호화됨	예
3	암호화되지 않음-암호화됨	예
4	암호화됨-암호화되지 않음	아니요

암호화되지 않은 원본 AMI를 암호화되지 않은 대상 AMI로 복사

이 시나리오에서 암호화되지 않은 단일 지원 스냅샷이 있는 AMI 복사본이 지정된 리전에 생성됩니다 (표시되지 않음). 이 다이어그램은 단일 지원 스냅샷이 있는 AMI를 보여 주지만 사용자는 여러 스냅샷이 있는 AMI도 복사할 수 있습니다. 각 스냅샷의 암호화 상태가 유지됩니다. 따라서 원본 AMI에 암호화되지 않은 스냅샷은 대상 AMI에 암호화되지 않은 스냅샷으로 이어지고, 원본 AMI에 암호화된 스냅샷이 있으면 대상 AMI에 암호화된 스냅샷으로 이어집니다.

암호화된 원본 AMI를 암호화된 대상 AMI로 복사

이 시나리오는 암호화된 스냅샷을 포함하지만 이전 시나리오와 기능적으로 동등합니다. 다른 스냅샷 AMI를 복사하는 동안 암호화를 적용하면 모든 대상 스냅샷은 지정된 키 또는 키가 지정되지 않은 경우 기본 키를 사용하여 암호화됩니다.

암호화되지 않은 원본 AMI를 암호화된 대상 AMI로 복사

이 시나리오에서 AMI 복사는 예를 들어 암호화되지 않은 스냅샷을 암호화하거나 암호화된 스냅샷을 다른 키로 다시 암호화하여 대상 이미지의 암호화 상태를 변경합니다. 복사 중 암호화를 적용하려면 암호화 파라미터인 암호화 플래그와 키를 제공해야 합니다. 이 키를 사용하는 경우에만 대상 스냅샷에서 생성된 볼륨에 액세스할 수 있습니다.

AMI 복사

다음과 같이 AMI를 복사할 수 있습니다.

필수 조건

Amazon EBS 스냅샷에서 지원되는 AMI를 생성하거나 가져옵니다. Amazon EC2 콘솔을 사용하여 AWS가 제공하는 광범위한 AMI를 검색할 수 있다는 점을 유의하십시오. 자세한 내용은 [Amazon EBS 지원 Linux AMI 생성 \(p. 81\)](#) 및 [Linux AMI 찾기 \(p. 67\)](#) 섹션을 참조하십시오.

콘솔을 사용하여 AMI를 복사하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 콘솔 탐색 모음에서 AMI가 들어 있는 리전을 선택합니다. 탐색 창에서 Images, AMIs를 선택하여 리전에서 사용할 수 있는 AMI 목록을 표시합니다.
3. 복사할 AMI를 선택하고 Actions 및 Copy AMI를 선택합니다.
4. AMI Copy 페이지에서 다음 정보를 지정한 다음 Copy AMI를 선택합니다.
 - Destination region: AMI를 복사할 리전.
 - Name: 새 AMI의 이름. AMI에 대한 세부 정보를 표시할 때 운영 체제 정보가 제공되지 않으므로, 이름에 운영 체제 정보를 넣을 수 있습니다.
 - [Description]: 원본과 사본을 구분할 수 있도록 설명에는 기본적으로 원본 AMI에 대한 정보가 포함됩니다. 필요에 따라 이 설명을 수정할 수 있습니다.
 - Encryption: 대상 스냅샷을 암호화하거나 다른 키를 사용하여 다시 암호화하려면 이 필드를 선택합니다.
 - Master Key: 대상 스냅샷을 암호화하기 위해 사용하는 KMS 키.
5. 복사 작업이 시작되었음을 알리는 확인 페이지가 표시되고 새 AMI의 ID가 제공됩니다.

복사 작업의 진행 상황을 즉시 확인하려면 제공된 링크를 따라갑니다. 진행 상황을 나중에 확인하려면 [Done]을 선택한 다음, 준비가 되면 탐색 모음을 사용하여 대상 리전으로 전환하고(해당하는 경우) AMI 목록에서 해당 AMI를 찾습니다.

대상 AMI의 초기 상태는 pending이고 작업이 완료되면 상태가 available이 됩니다.

명령줄을 사용하여 AMI를 복사하려면 다음을 수행합니다.

명령줄을 사용하여 AMI를 복사하려면 원본 리전과 대상 리전을 모두 지정해야 합니다. `--source-region` 파라미터를 사용하여 원본 리전을 지정합니다. 대상 리전의 경우 두 가지 옵션 중 하나를 선택할 수 있습니다.

- `--region` 파라미터를 사용합니다.
- 환경 변수를 설정합니다. 자세한 내용은 [AWS 명령줄 인터페이스 구성](#)을 참조하십시오.

암호화 중 대상 스냅샷을 암호화하는 경우 이러한 추가 파라미터를 지정해야 합니다.

- `부울`, `--encrypted`
- 문자열, `--kms-key-id`, 마스터 암호화 키 ID 제공

다음 명령 중 하나를 사용하여 AMI를 복사할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [copy-image\(AWS CLI\)](#)
- [Copy-EC2Image\(Windows PowerShell용 AWS 도구\)](#)

대기 중인 AMI 복사 작업 중지

다음과 같이 대기 중인 AMI 복사를 중지할 수 있습니다.

콘솔을 사용하여 AMI 복사 작업을 중지하려면 다음을 수행합니다.

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 모음의 리전 선택기에서 대상 리전을 선택합니다.
3. 탐색 창에서 [AMIs]를 선택합니다.
4. 복사를 중지할 AMI를 선택하고 [Actions] 및 [Deregister]를 선택합니다.
5. 확인 메시지가 표시되면 [Continue]를 선택합니다.

명령줄을 사용하여 AMI 복사 작업을 중지하려면 다음을 수행합니다.

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [deregister-image\(AWS CLI\)](#)
- [Unregister-EC2Image\(Windows PowerShell용 AWS 도구\)](#)

AMI 등록 취소

AMI 사용을 마쳤으면 AMI의 등록을 취소할 수 있습니다. AMI의 등록을 취소한 이후에는 새 인스턴스를 시작하기 위해 해당 AMI를 사용하는 것을 불가능합니다.

AMI의 등록을 취소할 경우 AMI에서 이미 시작한 인스턴스에는 영향이 없습니다. 이러한 인스턴스에 대한 사용 비용은 계속 발생합니다. 그러므로 이러한 인스턴스 관련 작업이 완료되면 해당 인스턴스를 종료해야 합니다.

AMI를 정리하는 데 사용할 절차는 Amazon EBS 기반인지, 인스턴스 스토어 기반인지에 따라 달라집니다. 인스턴스 스토어가 지원할 수 있는 Windows AMI는 Windows Server 2003에만 해당됩니다.

목차

- [Amazon EBS 기반 AMI 정리 \(p. 131\)](#)
- [인스턴스 스토어 기반 AMI 정리 \(p. 131\)](#)

Amazon EBS 기반 AMI 정리

Amazon EBS 기반 AMI의 등록을 취소하는 경우 AMI 생성 과정에서 인스턴스의 루트 볼륨에 대해 생성된 스냅샷에는 영향이 없습니다. 이 스냅샷에 대한 스토리지 비용이 계속 발생합니다. 그러므로 스냅샷 관련 작업이 완료되면 해당 스냅샷을 삭제해야 합니다.

다음 다이어그램에서는 Amazon EBS 기반 AMI를 정리하는 프로세스를 보여 줍니다.

Amazon EBS 기반 AMI를 정리하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [AMIs]를 선택합니다. AMI를 선택하고 그 ID를 메모해 놓습니다. — 그러면 다음 단계에서 올바른 스냅샷을 쉽게 찾을 수 있습니다. [Actions]를 선택한 후 [Deregister]를 선택합니다. 확인 메시지가 표시되면 [Continue]를 선택합니다.

이제 AMI 상태는 `unavailable`입니다.

Note

콘솔에서 상태를 `available`에서 `unavailable`로 변경하거나 AMI를 목록에서 완전히 제거하는 데 몇 분 정도 걸릴 수 있습니다. 상태를 새로 고치려면 [Refresh]를 선택합니다.

3. 탐색 창에서 [Snapshots]를 선택한 후 스냅샷을 선택합니다([Description] 열에서 AMI ID를 검색). [Actions]를 선택한 후 [Delete Snapshot]를 선택합니다. 확인 메시지가 나타나면 [Yes, Delete]를 선택합니다.
4. (선택 사항) AMI에서 시작한 인스턴스 관련 작업이 완료되면 해당 인스턴스를 종료합니다. 탐색 창에서 [Instances]를 선택합니다. 인스턴스를 선택하고 [Actions]를 선택한 후 [Instance State]와 [Terminate]를 차례로 선택합니다. 확인 메시지가 나타나면 [Yes, Terminate]를 선택합니다.

인스턴스 스토어 기반 AMI 정리

인스턴스 스토어 기반 AMI의 등록을 취소하는 경우 AMI를 만들 때 Amazon S3으로 업로드한 파일에는 영향이 없습니다. Amazon S3에서 이러한 파일에 대한 사용 비용은 계속 발생합니다. 그러므로 이러한 파일 관련 작업이 완료되면 해당 파일을 삭제해야 합니다.

다음 다이어그램에서는 인스턴스 스토어 기반 AMI를 정리하는 프로세스를 보여 줍니다.

인스턴스 스토어 기반 AMI를 정리하려면

1. 다음과 같이 `deregister-image` 명령을 사용하여 AMI의 등록을 취소합니다.

```
aws ec2 deregister-image --image-id ami_id
```

이제 AMI 상태는 `unavailable`입니다.

2. 다음과 같이 [ec2-delete-bundle \(p. 96\)](#)(AMI 도구)을 사용하여 Amazon S3에서 번들을 삭제합니다.

```
ec2-delete-bundle -b myawsbucket/myami -a your_access_key_id -s your_secret_access_key  
-p image
```

- (선택 사항) AMI에서 시작한 인스턴스 관련 작업이 완료되면 다음과 같이 [terminate-instances](#) 명령을 사용하여 해당 인스턴스를 종료할 수 있습니다.

```
aws ec2 terminate-instances --instance-ids instance_id
```

- (선택 사항) 번들을 업로드했던 Amazon S3 버킷 관련 작업이 완료되면 해당 버킷을 삭제할 수 있습니다. Amazon S3 버킷을 삭제하려면 Amazon S3 콘솔을 열고 해당 버킷을 선택한 후 [Actions], [Delete]를 차례로 선택합니다.

Amazon Linux

Amazon Linux는 Amazon Web Services(AWS)에서 제공하며, Amazon EC2에서 실행 중인 응용 프로그램에 대한 안정된 고성능 보안 실행 환경을 제공하도록 설계되었습니다. 또한 시작 구성 도구와 널리 사용되는 여러 AWS 라이브러리 및 도구 등 AWS와 쉽게 통합하는 데 사용할 수 있도록 하는 패키지가 포함되어 있습니다. AWS에서는 Amazon Linux를 실행하는 모든 인스턴스에 대해 지속적인 보안 및 유지 관리 업데이트를 제공합니다.

Note

Amazon Linux AMI 리포지토리 구조는 Amazon Linux AMI의 한 버전에서 다음 버전으로 롤링할 수 있도록 하는 연속 업데이트 흐름을 제공하도록 구성되었습니다. 기존 인스턴스를 현재 버전으로 잠그려면, [리포지토리 구성 \(p. 136\)](#) 섹션을 참조하십시오.

Amazon Linux 인스턴스를 시작하려면 Amazon Linux AMI를 사용합니다. AWS에서는 추가 요금 없이 Amazon Linux AMI를 Amazon EC2 사용자에게 제공합니다.

항목

- [Amazon Linux AMI 찾기 \(p. 132\)](#)
- [Amazon Linux 인스턴스 시작 및 연결 \(p. 132\)](#)
- [Amazon Linux AMI 이미지 식별 \(p. 133\)](#)
- [포함된 AWS 명령줄 도구 \(p. 133\)](#)
- [cloud-init \(p. 134\)](#)
- [리포지토리 구성 \(p. 136\)](#)
- [패키지 추가 \(p. 136\)](#)
- [참조용으로 원본 패키지에 액세스 \(p. 137\)](#)
- [애플리케이션 개발 \(p. 137\)](#)
- [인스턴스 스토어 액세스 \(p. 137\)](#)
- [제품 수명 주기 \(p. 137\)](#)
- [보안 업데이트 \(p. 138\)](#)
- [지원 \(p. 138\)](#)

Amazon Linux AMI 찾기

최신 Amazon Linux AMI 목록은 [Amazon Linux AMIs](#) 섹션을 참조하십시오.

Amazon Linux 인스턴스 시작 및 연결

원하는 AMI를 찾은 후에는 해당 AMI ID를 기록해둡니다. AMI ID를 사용하여 인스턴스를 시작하고 연결할 수 있습니다.

Amazon Linux에서는 기본적으로 원격 루트 SSH를 허용하지 않습니다. 또한 Brute-Force 암호 공격 방지를 위해 암호 인증을 사용할 수 없습니다. Amazon Linux 인스턴스에 대해 SSH 로그인을 사용하려면 시작 시 인스턴스에 키 페어를 제공해야 합니다. 또한 SSH 액세스를 허용하도록 인스턴스를 시작하는 데 사용되는 보안 그룹을 설정해야 합니다. 기본적으로 SSH를 사용하여 원격으로 로그인할 수 있는 계정은 `ec2-user`뿐입니다. 이 계정에는 `sudo` 권한도 있습니다. 원격 루트 로그인을 사용하려는 경우 키 페어와 부 사용자에 의존하는 방법보다는 덜 안전하다는 점에 유의하십시오.

Amazon Linux 인스턴스 시작 및 사용에 대한 자세한 내용은 [인스턴스 시작 \(p. 264\)](#) 섹션을 참조하십시오. Amazon Linux 인스턴스 연결에 대한 자세한 내용은 [Linux 인스턴스에 연결 \(p. 275\)](#) 섹션을 참조하십시오.

Amazon Linux AMI 이미지 식별

각 이미지에는 AMI를 식별하는 고유한 `/etc/image-id`가 포함되어 있습니다. 이 파일에는 이미지에 대한 정보가 포함되어 있습니다.

다음은 `/etc/image-id` 파일의 예입니다.

```
[ec2-user ~]$ cat /etc/image-id
image_name="amzn-ami-hvm"
image_version="2017.03"
image_arch="x86_64"
image_file="amzn-ami-hvm-2017.03.0.20170401-x86_64.ext4.gpt"
image_stamp="26a3-ed31"
image_date="20170402053945"
recipe_name="amzn ami"
recipe_id="47cfa924-413c-d460-f4f2-2af7-feb6-9e37-7c9f1d2b"
```

`image_name`, `image_version` 및 `image_arch` 항목은 Amazon에서 이미지를 생성하는 데 사용되는 빌드 레시피에서 가져온 것입니다. `image_stamp`은 단순히 이미지 생성 중에 생성된 고유한 임의 16진수 값입니다. `image_date` 항목은 YYYYMMDDhhmmss 형식이며 이미지 생성 시간(UTC)입니다. `recipe_name` 및 `recipe_id`는 Amazon에서 이미지를 생성하는 데 사용된 빌드 레시피의 이름 및 ID를 참조하여 현재 실행 중인 Amazon Linux 버전을 식별합니다. 이 파일은 yum 리포지토리에서 업데이트를 설치할 때 변경되지 않습니다.

Amazon Linux에는 설치된 현재 릴리스를 지정하는 `/etc/system-release` 파일이 포함되어 있습니다. 이 파일은 yum을 통해 업데이트되며 system-release RPM의 일부입니다.

다음은 `/etc/system-release` 파일의 예입니다.

```
[ec2-user ~]$ cat /etc/system-release
Amazon Linux AMI release 2017.03
```

Amazon Linux에는 `/etc/system-release-cpe`에 있는 `/etc/system-release` 파일의 머신 판독 가능한 버전이 포함되어 있으며 MITRE(CPE)의 CPE 사양을 따릅니다.

포함된 AWS 명령줄 도구

다음과 같이 널리 알려진 명령줄 도구가 AWS 통합 및 사용을 위해 Amazon Linux 또는 기본 리포지토리에 포함되었습니다.

- `aws-amitools-ec2`
- `aws-apitools-as`
- `aws-apitools-cfn`
- `aws-apitools-ec2`
- `aws-apitools-elb`

- `aws-apitools-iam`
- `aws-apitools-mon`
- `aws-apitools-rds`
- `aws-cfn-bootstrap`
- `aws-cli`
- `aws-scripts-ses`

Note

Amazon Linux(amzn-ami-minimal-*)의 최소 버전에는 위의 패키지가 포함되지 않습니다. 그러나 기본 yum 리포지토리에 제공되며,

```
[ec2-user ~]$ sudo yum install -y package_name
```

명령을 사용하여 설치할 수 있습니다.

`aws-apitools-*` 명령줄 도구는 모든 Amazon Linux 버전에 포함되어 있지만, `aws-cli` 명령줄 도구는 모든 Amazon Web Services에 걸쳐 스탠다드 환경을 제공하며 궁극적으로 서버별 도구 세트를 대체합니다.

IAM 역할을 사용하여 시작한 인스턴스의 경우, 이러한 도구의 구성을 간소화하기 위해 자격 증명 파일을 설치한 후 `AWS_CREDENTIAL_FILE`, `JAVA_HOME`, `AWS_PATH`, `PATH` 및 제품별 환경 변수를 준비하기 위한 단순 스크립트가 포함되었습니다.

또한 여기에 설명된 바와 같이 API 및 AMI 도구의 여러 버전을 설치할 수 있도록 하기 위해 이러한 도구의 원하는 버전으로 안내하는 심볼 링크를 `/opt/aws`에 추가했습니다.

`/opt/aws/bin`

설치된 각 도구 디렉터리의 `/bin` 디렉터리로 안내하는 심볼 링크

`/opt/aws/{apitools|amitools}`

제품은 `name-version` 형태의 디렉터리에 설치되며 심볼 링크 `##`이 최근 설치된 버전에 연결되어 있습니다.

`/opt/aws/{apitools|amitools}/name/environment.sh`

`/etc/profile.d/aws-apitools-common.sh`에서 `EC2_HOME` 등의 제품별 환경 변수를 설정하는 데 사용됩니다.

`cloud-init`

`cloud-init` 패키징은 Canonical에서 오픈 소스 애플리케이션이며 Amazon EC2 등의 클라우드 컴퓨팅 환경에서 Linux 이미지 부트스트랩을 수행하는 데 사용됩니다. Amazon Linux에는 `cloud-init`의 사용자 지정 버전이 포함되어 있습니다. 부팅 시 인스턴스에 대해 발생할 작업을 지정할 수 있습니다. 인스턴스를 시작할 때 사용자 데이터 필드를 통해 원하는 작업을 `cloud-init`로 전달할 수 있습니다. 다시 말해서 여러 사용 사례에 공통 AMI를 사용하고 이러한 AMI를 시작 시 동적으로 구성할 수 있다는 뜻입니다. 또한 Amazon Linux에서 `cloud-init`를 사용하여 `ec2-user` 계정의 초기 구성도 수행합니다.

`cloud-init`에 대한 자세한 내용은 <http://cloudinit.readthedocs.org/en/latest/> 섹션을 참조하십시오.

Amazon Linux에서는 다음 `cloud-init` 작업(`/etc/sysconfig/cloudinit`에서 구성 가능)을 사용합니다.

- 작업: INIT(항상 실행)
- 기본 로캘 설정

- 호스트 이름 설정
- 사용자 데이터 구문 분석 및 처리
- 작업: CONFIG_SSH
 - 호스트 프라이빗 SSH 키 생성
 - 손쉬운 로그인 및 관리를 위해 사용자의 퍼블릭 SSH 키를 .ssh/authorized_keys에 추가
- 작업: PACKAGE_SETUP
 - yum repo 준비
 - 사용자 데이터에 정의된 패키지 작업 처리
- 작업: RUNCMD
 - 셸 명령 실행
- 작업: RUN_USER_SCRIPTS
 - 사용자 데이터에 있는 사용자 스크립트 실행
- 작업: CONFIG_MOUNTS
 - 휴발성 드라이브 마운트
- 작업: CONFIG_LOCALE
 - 사용자 데이터에 따라 로캘 구성 파일에서 로캘 설정

지원되는 사용자 데이터 형식

cloud-init 패키지는 다양한 형식의 사용자 데이터 처리를 지원합니다.

- Gzip
 - 사용자 데이터가 gzip으로 압축된 경우 cloud-init는 데이터에 대한 압축을 해제하고 데이터를 적절히 처리합니다.
- MIME 멀티파트
 - MIME 멀티파트 파일을 사용하여 두 가지 이상의 데이터 유형을 지정할 수 있습니다. 예를 들어, 사용자 데이터 스크립트와 클라우드 구성 유형을 모두 지정할 수 있습니다. 멀티파트 파일의 각 부분은 지원되는 형식 중 하나일 경우 cloud-init에 의해 처리할 수 있습니다.
- Base64 디코딩
 - 사용자 데이터가 base64-encoded인 경우 cloud-init는 디코딩된 데이터를 지원 유형 중 하나로 인식 할 수 있는지 여부를 판단합니다. 디코딩된 데이터를 인식하는 경우 데이터를 디코딩하여 그에 맞게 처리합니다. 그렇지 않을 경우 base64 데이터를 원상태로 반환합니다.
- 사용자 데이터 스크립트
 - #! 또는 Content-Type: text/x-shellscript로 시작합니다.
 - 이 스크립트는 최초 부팅 주기 중에 /etc/init.d/cloud-init-user-scripts에 의해 실행됩니다. 이 동작은 부팅 프로세스 후반(초기 구성 작업이 수행된 후)에 발생합니다.
- Include 파일
 - #include 또는 Content-Type: text/x-include-url로 시작합니다.
 - 이것은 include 파일의 내용입니다. 이 파일에는 줄당 URL 하나씩, URL 목록이 포함되어 있습니다. 각각의 URL을 읽어오며 해당 내용이 이 동일한 규칙 세트를 통과합니다. URL에서 읽어온 내용은 gzip으로 압축된 형태이거나 MIME-multi-part 또는 일반 텍스트 형태일 수 있습니다.
- 클라우드 구성 데이터
 - #cloud-config 또는 Content-Type: text/cloud-config로 시작합니다.
 - 이것은 클라우드 구성 데이터의 내용입니다. 주석이 포함된 지원되는 구성 형식의 예를 참조하십시오.
- 클라우드 Boothook
 - #cloud-boothook 또는 Content-Type: text/cloud-boothook로 시작합니다.
 - 이것은 bookhook 데이터의 내용입니다. /var/lib/cloud에 있는 파일에 저장되고 나서 즉시 실행됩니다.

- 이것은 맨 처음으로 사용 가능한 "hook"입니다. 한 번만 실행되도록 제공된 메커니즘이 없습니다. boothook는 이 부분을 자체적으로 처리해야 합니다. 환경 변수 `INSTANCE_ID`에 인스턴스 ID가 함께 제공됩니다. 이 변수를 사용하여 booothook 데이터의 인스턴스당 1회 세트를 제공하십시오.

리포지토리 구성

Amazon Linux의 2011.09 릴리스부터는 Amazon Linux AMI가 시간의 스냅샷으로 처리되며, `yum update -y`를 실행할 때 최신 패키지를 제공하는 리포지토리 및 업데이트 구조가 제공됩니다.

리포지토리 구조는 Amazon Linux의 한 버전에서 다음 버전으로 롤링할 수 있도록 하는 연속 업데이트 흐름을 제공하도록 구성되었습니다. 예를 들어, 이전 Amazon Linux AMI 버전(2016.09 이하)의 인스턴스를 시작하고 `yum update -y`를 실행할 경우 최신 패키지가 제공됩니다.

`lock-on-launch` 기능을 사용하도록 설정하여 Amazon Linux에 대한 롤링 업데이트를 사용하지 않도록 설정할 수 있습니다. `lock-on-launch` 기능은 지정된 AMI 릴리스에서만 업데이트를 받도록 새로 시작된 인스턴스를 잡깁니다. 예를 들어, 2016.09 AMI를 시작한 후, 2017.03 AMI로 마이그레이션할 준비될 때까지 2017.03 AMI 이전에 릴리스된 업데이트만 받도록 할 수 있습니다. 새 인스턴스에서 `lock-on-launch`를 사용하려면 Amazon EC2 콘솔이나 `ec2-run-instances` 명령과 함께 `-f` 플래그를 사용하여 `cloud-init`에 전달된 사용자 데이터로 인스턴스를 시작하십시오.

Important

`latest` 버전이 아닌 리포지토리에 AMI를 고정하면, 추가 업데이트가 수신되지 않습니다. Amazon Linux AMI 업데이트를 지속적으로 수신할 수 있는 유일한 방법은 최신 AMI를 사용하거나 이전 AMI를 계속해서 업데이트하여 리포지토리가 `latest`를 가리키게 하는 것입니다.

```
#cloud-config
repo_releasever: 2016.09
```

기존 인스턴스를 현재 AMI 릴리스 버전으로 잠그려면

1. `/etc/yum.conf`를 편집합니다.
2. `releasever=latest`를 주석으로 처리합니다.
3. `yum clean all`을 실행하여 캐시를 지웁니다.

패키지 추가

Amazon Linux는 각 Amazon EC2 리전에 호스팅된 온라인 패키지 리포지토리와 함께 사용하도록 설계되었습니다. 이러한 리포지토리는 Amazon Linux AMI에서 패키지에 대한 지속적인 업데이트는 물론, 수백 개의 추가 일반 오픈 소스 서버 애플리케이션에 액세스할 수 있는 기능도 제공합니다. 리포지토리는 yum 업데이트 도구를 사용하여 액세스되며 모든 리전과 [Amazon Linux AMI 패키지 사이트](#)에서 사용할 수 있습니다. 각 리전에서 리포지토리를 호스팅하면 데이터 전송 요금 없이 데이터를 신속히 배포할 수 있습니다. 패키지는 다음 예와 같이 yum 명령을 실행하여 설치할 수 있습니다.

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

Extra Packages for Enterprise Linux(EPEL) 리포지토리에 대한 액세스가 구성되었지만 기본적으로 사용되지 않습니다. EPEL은 Amazon Linux 리포지토리에 있는 패키지 이외의 타사 패키지를 제공합니다. 타사 패키지는 AWS에서 지원되지 않습니다.

Amazon Linux에 필요한 애플리케이션이 포함되지 않은 경우 단순히 Amazon Linux 인스턴스에 해당 애플리케이션을 직접 설치하기만 하면 됩니다. Amazon Linux에서는 패키지 관리용으로 RPM 및 yum을 사용하며 이것이 새 애플리케이션을 설치하는 가장 간단한 방법일 것입니다. 중앙 Amazon Linux 리포지토리에는 사

용할 수 있는 애플리케이션이 많으므로 항상 이곳에서 애플리케이션을 사용할 수 있는지부터 확인해야 합니다. 이러한 애플리케이션은 Amazon Linux 인스턴스에 쉽게 추가할 수 있습니다.

애플리케이션을 실행 중인 Amazon Linux 인스턴스에 업로드 하려면 `scp` 또는 `sftp`를 사용하고 인스턴스에 로그온하여 애플리케이션을 구성합니다. 또한 기본 제공된 `cloud-init` 패키지의 `PACKAGE_SETUP` 작업을 사용하여 인스턴스 시작 중에 애플리케이션을 업로드할 수도 있습니다. 자세한 내용은 [cloud-init \(p. 134\)](#) 섹션을 참조하십시오.

Important

인스턴스를 Virtual Private Cloud(VPC)에서 실행 중인 경우 `yum` 리포지토리에 연결하려면 VPC에 인터넷 게이트웨이를 연결해야 합니다. 자세한 내용은 Amazon VPC 사용 설명서의 [Internet Gateways](#) 섹션을 참조하십시오.

참조용으로 원본 패키지에 액세스

Amazon Linux에 제공된 도구를 사용하여 참조 목적으로 인스턴스에 설치한 패키지의 원본을 볼 수 있습니다. Amazon Linux에 포함된 모든 패키지와 온라인 패키지 리포지토리에 대해 원본 패키지를 사용할 수 있습니다. 설치할 원본 패키지의 패키지 이름을 간단히 확인하고 `get_reference_source` 명령을 사용하여 실행 중인 인스턴스 내에서 원본을 확인합니다. 예:

```
[ec2-user ~]$ get_reference_source -p bash
```

다음은 응답 예입니다.

```
### ###: bash ## RPM ##### #: bash-4.2.46-20.36.amzn1.x86_64 ## ##### ## RPM:  
bash-4.2.46-20.36.amzn1.src.rpm ### #####? ##### 'yes'# #####.: yes ## RPM ##### #  
#: /usr/src/srpm/debug/bash-4.2.46-20.36.amzn1.src.rpm
```

원본 RPM은 인스턴스의 `/usr/src/srpm/debug` 디렉터리에 있습니다. 여기에서 원본 RPM의 압축을 풀고 스탠다드 RPM 도구를 사용하여 참조용으로 원본 트리를 볼 수 있습니다. 디버깅을 완료한 후에는 패키지를 사용할 수 있습니다.

Important

인스턴스를 Virtual Private Cloud(VPC)에서 실행 중인 경우 `yum` 리포지토리에 연결하려면 VPC에 인터넷 게이트웨이를 연결해야 합니다. 자세한 내용은 Amazon VPC 사용 설명서의 [Internet Gateways](#) 섹션을 참조하십시오.

애플리케이션 개발

전체 Linux 개발 도구 세트가 Amazon Linux용 `yum` 리포지토리에 제공됩니다. Amazon Linux에서 애플리케이션을 개발하려면 `yum`을 사용하여 필요한 개발 도구를 선택하십시오. 또는 CentOS 및 다른 유사 배포에서 개발한 많은 애플리케이션이 Amazon Linux에서 실행됩니다.

인스턴스 스토어 액세스

Amazon 인스턴스 스토어 기반 AMI에만 `/media/ephemeral0`에 인스턴스 스토어 드라이브 `ephemeral0`이 마운트되어 있습니다. 이 점이 `/mnt`에 인스턴스 스토어 드라이브를 마운트하는 다른 여러 이미지와의 차이점입니다.

제품 수명 주기

Amazon Linux AMI는 정기적으로 업데이트되어 보안 및 기능 향상을 제공합니다. Amazon Linux 인스턴스에 대한 데이터 또는 사용자 지정을 유지할 필요가 없는 경우 단순히 Amazon Linux AMI를 사용하여 새 인스

터스를 다시 시작할 수 있습니다. Amazon Linux 인스턴스에 대한 데이터나 사용자 지정을 유지해야 하는 경우 우에는 Amazon Linux yum 리포지토리를 통해 이러한 인스턴스를 유지할 수 있습니다. yum 리포지토리에는 업데이트된 모든 패키지가 포함되어 있습니다. 실행 중인 인스턴스에 이러한 업데이트를 적용하도록 선택할 수 있습니다.

이전 AMI 버전 및 업데이트 패키지는 새 버전이 릴리스되더라도 계속 사용할 수 있습니다. 경우에 따라 AWS Support를 통해 이전 Amazon Linux; 버전에 대한 지원을 필요로 하는 경우 지원 프로세스의 일부로 새 버전으로 이동을 요청드릴 수 있습니다.

보안 업데이트

보안 업데이트는 Amazon Linux AMI yum 리포지토리와 업데이트된 Amazon Linux AMI를 통해 제공됩니다. 보안 알림은 [Amazon Linux AMI 보안 센터](#)에 게시됩니다. AWS 보안 정책에 대한 자세한 내용을 찾아보거나 보안 문제를 보고하려면 [AWS 보안 센터](#)로 이동하십시오.

Amazon Linux AMI는 시작 시 보안 업데이트를 다운로드 및 설치하도록 구성되어 있습니다. 이 기능은 `repo_upgrade`라는 `cloud-init` 설정을 통해 제어됩니다. 다음 `cloud-init` 구성 코드 조각은 인스턴스 초기화에 전달하는 사용자 데이터 텍스트의 설정을 변경하는 방법을 보여 줍니다.

```
#cloud-config
repo_upgrade: security
```

`repo_upgrade` 설정의 사용 가능한 값은 다음과 같습니다.

`security`

Amazon에서 보안 업데이트로 표시하는 대기 중인 업데이트를 적용합니다.

`bugfix`

Amazon에서 버그 수정 사항으로 표시하는 업데이트를 적용합니다. 버그 수정 사항은 대규모 업데이트 세트이며 보안 업데이트와 사소한 기타 버그에 대한 수정 사항을 포함합니다.

`all`

분류와 관계 없이 해당되는 모든 업데이트를 적용합니다.

`none`

시작 시 인스턴스에 어떠한 업데이트도 적용하지 마십시오.

`repo_upgrade`에 대한 기본 설정은 `security`입니다. 즉, 기본적으로 사용자 데이터에 다른 값을 지정하지 않은 경우 Amazon Linux AMI에서는 해당 시점에 설치된 모든 패키지에 대해 시작 시 보안 업그레이드를 수행합니다. Amazon Linux AMI에서는 `/etc/motd` 파일을 사용하여 로그인 시 사용 가능한 업데이트를 나열함으로써 설치된 패키지에 대한 업데이트를 사용자에게 알립니다. 이러한 업데이트를 설치하려면 인스턴스에서 `sudo yum upgrade`를 실행해야 합니다.

Important

인스턴스를 Virtual Private Cloud(VPC)에서 실행 중인 경우 yum 리포지토리에 연결하려면 VPC에 인터넷 게이트웨이를 연결해야 합니다. 자세한 내용은 Amazon VPC 사용 설명서의 [Internet Gateways](#) 섹션을 참조하십시오.

지원

기본 Amazon Linux AMI의 설치 및 사용 지원은 AWS Support 가입에 포함되어 있습니다. 자세한 내용은 [AWS Support](#) 섹션을 참조하십시오.

Amazon Linux에 대한 질문이 있으면 [Amazon EC2 forum](#)에 게시하십시오.

사용자 제공 커널

Amazon EC2 인스턴스에 사용자 지정 커널이 필요할 경우 가장 적합한 AMI를 사용하여 시작한 후 해당 인스턴스에서 사용자 지정 커널을 컴파일하고, `menu.lst` 파일을 열어 새 커널을 지정합니다. 이 프로세스는 AMI에서 사용하는 가상화 유형에 따라 다릅니다. 자세한 내용은 [Linux AMI 가상화 유형 \(p. 66\)](#) 섹션을 참조하십시오.

목차

- [HVM AMI\(GRUB\) \(p. 139\)](#)
- [반가상화 AMI\(PV-GRUB\) \(p. 140\)](#)

HVM AMI(GRUB)

HVM 인스턴스 볼륨은 실제 물리적 디스크인 것처럼 취급됩니다. 부팅 프로세스는 디스크 파티션이 설정되고 부트로더가 있는 베어 메탈(bare metal) 운영 체제의 부팅 프로세스와 비슷하며, 현재 지원되는 모든 Linux 배포를 사용할 수 있습니다. 가장 일반적인 부트로더는 GRUB이며, 다음 섹션에서는 사용자 지정 커널을 사용하도록 GRUB를 구성하는 방법을 설명합니다.

HVM AMI에 대해 GRUB 구성

다음은 HVM AMI에 대한 `menu.lst` 구성 파일의 예입니다. 이 예에서는 Amazon Linux 2017.03(이 AMI에 대한 원래 커널)와, Vanilla Linux 4.7.4(<https://www.kernel.org>)에서 제공되는 Vanilla Linux 커널의 최신 버전) 중 하나를 선택할 수 있습니다. Vanilla 항목은 해당 AMI에 대한 원래 항목에서 복제된 것이며, `kernel` 및 `initrd` 경로는 새 위치로 업데이트됩니다. `default 0` 파라미터는 부트로더가 발견한 첫 번째 항목(이 경우는 Vanilla 항목)을 참조하게 하고, `fallback 1` 파라미터는 첫 항목 부팅에 문제가 있는 경우 부트로더가 두 번째 항목을 참조하게 합니다.

기본적으로 GRUB는 추가적인 부팅 지연을 발생시키지 않기 위해 인스턴스 콘솔에 출력을 전송하지 않습니다. 자세한 내용은 [인스턴스 콘솔 출력 \(p. 729\)](#) 섹션을 참조하십시오. 사용자 지정 커널을 설치할 경우 아래 예제에 나와 있는 것처럼, `hiddenmenu` 행을 삭제하고 `serial` 및 `terminal` 행을 `/boot/grub/menu.lst`에 추가하여 GRUB 출력을 활성화하는 것을 고려해 보십시오.

Important

부팅 프로세스 중 많은 양의 디버그 정보가 출력되지 않도록 하십시오. 직렬 콘솔은 높은 데이터 전송 속도를 지원하지 않습니다.

```
default=0
fallback=1
timeout=5
serial --unit=0 --speed=9600
terminal --dumb --timeout=5 serial console

title Vanilla Linux 4.7.4
root (hd0)
kernel /boot/vmlinuz-4.7.4 root=LABEL=/ console=tty1 console=ttyS0
initrd /boot/initrd.img-4.7.4

title Amazon Linux 2017.03 (4.9.17-8.31.amzn1.x86_64)
root (hd0,0)
kernel /boot/vmlinuz-4.9.17-8.31.amzn1.x86_64 root=LABEL=/ console=tty1 console=ttyS0
initrd /boot/initramfs-4.9.17-8.31.amzn1.x86_64.img
```

`menu.lst` 파일에서 대체 커널을 지정할 필요는 없지만, 새 커널을 테스트할 때 대체 커널을 사용하는 것이 권장됩니다. GRUB은 새 커널에 장애가 발생한 경우 다른 커널로 이를 대체하여 사용할 수 있습니다. 대체 커널이 있을 경우 인스턴스는 새 커널을 발견할 수 없는 경우에도 부팅할 수 있습니다.

새로운 Vanilla Linux 커널 파일이 실패할 경우 아래와 같은 출력이 표시될 수 있습니다.

```
^M Entry 0 will be booted automatically in 3 seconds. ^M Entry 0 will be booted
automatically in 2 seconds. ^M Entry 0 will be booted automatically in 1 seconds.

Error 13: Invalid or unsupported executable format
[ 0.000000] Initializing cgroup subsys cpuset
```

반가상화 AMI(PV-GRUB)

PV(반가상화) 가상화를 사용하는 Amazon 머신 이미지은 PV-GRUB라는 시스템을 부팅 과정 동안 사용합니다. PV-GRUB은 GNU GRUB 0.97의 패치 버전을 실행하는 반가상화 부트로더입니다. 인스턴스를 실행할 때 PV-GRUB은 부팅 과정을 실행하고 이미지의 menu.lst 파일에 지정된 커널을 체인로드합니다.

PV-GRUB는 표준 grub.conf 또는 menu.lst 명령을 이해할 수 있으며 따라서 모든 최신 지원 Linux 배포판과 함께 사용할 수 있습니다. Ubuntu 10.04 LTS, Oracle Enterprise Linux, CentOS 5.x 등 이전 배포판은 특별한 "ec2" 또는 "xen" 커널 패키지를 필요로 하지만, 새 배포판은 필요한 드라이버를 기본 커널 패키지에 포함하고 있습니다.

대부분의 PV(반가상화) AMI는 PV-GRUB AKI를 기본적으로 사용하므로(Amazon EC2 Launch Wizard Start 메뉴에서 제공되는 모든 PV Linux AMI 포함), 사용할 다른 커널이 사용자의 배포판과 호환되는 경우라면 인스턴스에서 해당 커널을 사용하기 위해 별도의 조치를 취할 필요는 없습니다. 인스턴스에서 사용자 지정 커널을 실행하는 최상의 방법은 원하는 것과 가장 근접한 AMI로 실행하고, 인스턴스에서 사용자 지정 커널을 컴파일하고, 반가상화 (PV-GRUB)에 대한 GRUB AMI 구성 (p. 141)에서 표시된 menu.lst 파일을 수정하는 것입니다.

다음 [describe-images](#) 명령을 Amazon EC2 명령줄 도구(검사하려는 커널 이미지 ID로 해당 부분 대체)로 실행하여 AMI에 대한 커널 이미지가 PV-GRUB AKI인지 검사할 수 있습니다.

```
$ aws ec2 describe-images --filters Name=image-id,Values=aki-880531cd
```

Name 필드가 pv-grub으로 시작하는지 확인합니다.

항목

- [PV-GRUB의 제한](#) (p. 140)
- [반가상화 \(PV-GRUB\)에 대한 GRUB AMI 구성](#) (p. 141)
- [Amazon PV-GRUB 커널 이미지 ID](#) (p. 141)
- [PV-GRUB 업데이트](#) (p. 143)

PV-GRUB의 제한

PV-GRUB에는 다음과 같은 제한 사항이 있습니다.

- 64비트 버전의 PV-GRUB을 사용해서 32비트 커널을 실행할 수는 없으며, 32비트 버전의 PV-GRUB을 사용해서 64비트 커널을 실행할 수도 없습니다.
- PV-GRUB AKI를 사용할 때 ARI(Amazon 램디스크 이미지)를 지정할 수 없습니다.
- AWS는 PV-GRUB이 EXT2, EXT3, EXT4, JFS, XFS, ReiserFS 등의 파일 시스템 포맷과 작동함을 테스트하고 검증했습니다. 그 밖의 파일 시스템 포맷은 PV-GRUB에서 작동하지 않을 수 있습니다.
- PV-GRUB은 gzip, bzip2, lzo, xz 압축 포맷을 사용해서 압축된 커널을 부팅시킬 수 있습니다.
- Cluster AMI는 완전한 HVM(하드웨어 가상 머신)을 사용하기 때문에 PV-GRUB을 지원하지 않으며 이를 필요로 하지도 않습니다. PV(반가상화) 인스턴스는 PV-GRUB을 사용해서 부팅하지만, HVM 인스턴스 볼륨은 실제 디스크처럼 취급되며 그 부팅 과정은 파티션 처리된 디스크와 부트로더가 있는 베어 메탈(bare metal) 운영 체제의 부팅 과정과 유사합니다.

- PV-GRUB 버전 1.03 및 그 이하 버전은 GPT 파티셔닝을 지원하지 않으며 MBR 파티셔닝만 지원합니다.
- Amazon EBS 볼륨으로 LVM(Logical Volume Manager)를 사용할 계획인 경우, LVM 외부의 개별적인 부트 파티션을 필요로 합니다. 상기 요건이 갖추어지면 LVM으로 논리적 볼륨을 생성할 수 있게 됩니다.

반가상화 (PV-GRUB)에 대한 GRUB AMI 구성

PV-GRUB 부팅을 하려면 GRUB `menu.lst` 파일이 이미지 내에 존재해야 합니다. 이 파일이 일반적으로 위치하는 곳은 `/boot/grub/menu.lst`입니다.

다음은 PV-GRUB AKI로 AMI를 부팅하는 것에 대한 `menu.lst` 구성 파일의 예입니다. 이 예에서는 Amazon Linux 2017.03(이 AMI에 대한 원래 커널)과, Vanilla Linux 4.7.4(<https://www.kernel.org/>에서 제공되는 Vanilla Linux 커널의 최신 버전) 중 하나를 선택할 수 있습니다. Vanilla 항목은 해당 AMI에 대한 원래 항목에서 복제된 것이며, `kernel` 및 `initrd` 경로는 새 위치로 업데이트됩니다. `default 0` 파라미터는 부트로더가 발견한 첫 번째 항목(이 경우는 Vanilla 항목)을 참조하게 하고, `fallback 1` 파라미터는 첫 항목 부팅에 문제가 있는 경우 부트로더가 두 번째 항목을 참조하게 합니다.

```
default 0
fallback 1
timeout 0
hiddenmenu

title Vanilla Linux 4.7.4
root (hd0)
kernel /boot/vmlinuz-4.7.4 root=LABEL=/ console=hvc0
initrd /boot/initrd.img-4.7.4

title Amazon Linux 2017.03 (4.9.17-8.31.amzn1.x86_64)
root (hd0)
kernel /boot/vmlinuz-4.9.17-8.31.amzn1.x86_64 root=LABEL=/ console=hvc0
initrd /boot/initramfs-4.9.17-8.31.amzn1.x86_64.img
```

`menu.lst` 파일에서 대체 커널을 지정할 필요는 없지만, 새 커널을 테스트할 때 대체 커널을 사용하는 것이 권장됩니다. PV-GRUB은 새 커널에 장애가 발생한 경우 다른 커널로 이를 대체하여 사용할 수 있습니다. 대체 커널이 있을 경우 인스턴스는 새 커널을 발견할 수 없는 경우에도 부팅할 수 있습니다.

PV-GRUB은 `menu.lst`를 찾기 위해 다음 위치를 검사합니다(발견한 경우 그 이하 경로는 검색 안 함).

- `(hd0)/boot/grub`
- `(hd0,0)/boot/grub`
- `(hd0,0)/grub`
- `(hd0,1)/boot/grub`
- `(hd0,1)/grub`
- `(hd0,2)/boot/grub`
- `(hd0,2)/grub`
- `(hd0,0)/boot/grub`
- `(hd0,3)/grub`

PV-GRUB 1.03 이하 버전은 이 목록에서 첫 2개의 위치만 검색합니다.

Amazon PV-GRUB 커널 이미지 ID

PV-GRUB AKI는 모든 Amazon EC2 리전에서 제공됩니다. 32비트 및 64비트의 두 아키텍처 유형에 대한 AKI가 존재합니다. 가장 최신의 AMI는 기본적으로 PV-GRUB AKI를 사용합니다.

모든 PV-GRUB 버전이 모든 인스턴스 유형과 호환되는 것은 아니기 때문에, 언제나 PV-GRUB AKI의 최신 버전을 사용하는 것이 권장됩니다. 다음 [describe-images](#) 명령을 사용하여 현재 리전에 대한 PV-GRUB AKI 목록을 가져옵니다.

```
$ aws ec2 describe-images --owners amazon --filters Name=name,Values=pv-grub-* .gz
```

PV-GRUB은 ap-southeast-2 리전에서만 제공되는 AKI입니다. 사용자가 해당 리전에 복사할 AMI가 해당 리전에서 사용 가능한 PV-GRUB의 버전을 사용하는지를 확인해야 합니다.

다음은 각 리전에 대한 현재 AKI ID입니다. hd0 AKI를 사용해서 새 AMI를 등록할 수 있습니다.

Note

hd00 AKI가 이전에 제공되었던 리전의 경우 이전 버전과의 호환성을 위해 hd00 AKI가 계속해서 제공되고 있습니다.

ap-northeast-1, 아시아 태평양(도쿄)

이미지 ID	이미지 이름
aki-f975a998	pv-grub-hd0_1.05-i386.gz
aki-7077ab11	pv-grub-hd0_1.05-x86_64.gz

ap-southeast-1, 아시아 태평양(싱가포르) 리전

이미지 ID	이미지 이름
aki-17a40074	pv-grub-hd0_1.05-i386.gz
aki-73a50110	pv-grub-hd0_1.05-x86_64.gz

ap-southeast-2, 아시아 태평양(시드니)

이미지 ID	이미지 이름
aki-ba5665d9	pv-grub-hd0_1.05-i386.gz
aki-66506305	pv-grub-hd0_1.05-x86_64.gz

eu-central-1, EU(프랑크푸르트)

이미지 ID	이미지 이름
aki-1419e57b	pv-grub-hd0_1.05-i386.gz
aki-931fe3fc	pv-grub-hd0_1.05-x86_64.gz

eu-west-1, EU(아일랜드)

이미지 ID	이미지 이름
aki-1c9fd86f	pv-grub-hd0_1.05-i386.gz
aki-dc9ed9af	pv-grub-hd0_1.05-x86_64.gz

sa-east-1, 남아메리카(상파울루)

이미지 ID	이미지 이름
aki-7cd34110	pv-grub-hd0_1.05-i386.gz
aki-912fbcf8	pv-grub-hd0_1.05-x86_64.gz

us-east-1, 미국 동부(버지니아 북부)

이미지 ID	이미지 이름
aki-04206613	pv-grub-hd0_1.05-i386.gz
aki-5c21674b	pv-grub-hd0_1.05-x86_64.gz

us-gov-west-1, AWS GovCloud (US)

이미지 ID	이미지 이름
aki-5ee9573f	pv-grub-hd0_1.05-i386.gz
aki-9ee55bff	pv-grub-hd0_1.05-x86_64.gz

us-west-1, 미국 서부(캘리포니아 북부 지역)

이미지 ID	이미지 이름
aki-43cf8123	pv-grub-hd0_1.05-i386.gz
aki-59cc8239	pv-grub-hd0_1.05-x86_64.gz

us-west-2, 미국 서부(오레곤)

이미지 ID	이미지 이름
aki-7a69931a	pv-grub-hd0_1.05-i386.gz
aki-70cb0e10	pv-grub-hd0_1.05-x86_64.gz

PV-GRUB 업데이트

모든 PV-GRUB 버전이 모든 인스턴스 유형과 호환되는 것은 아니기 때문에, 언제나 PV-GRUB AKI의 최신 버전을 사용하는 것이 권장됩니다. 또한 PV-GRUB의 이전 버전이 모든 리전에서 사용 가능한 것은 아니므로, 이전 버전을 사용하는 AMI를 해당 버전을 지원하지 않는 리전으로 복사한 경우는 커널 이미지를 업데이트할 때까지 AMI에서 실행된 인스턴스를 부팅시킬 수 없습니다. 다음 절차를 사용해 PV-GRUB의 인스턴스 버전을 확인하고 필요한 경우 업데이트를 하십시오.

PV-GRUB 버전 확인 방법

1. 인스턴스에 대한 커널 ID를 찾습니다.

```
$ aws ec2 describe-instance-attribute --instance-id instance_id --attribute kernel --region region
```

```
{  
    "InstanceId": "instance_id",  
    "KernelId": "aki-70cb0e10"  
}
```

이 인스턴스에 대한 커널 ID는 aki-70cb0e10입니다.

- 해당 커널 ID의 버전 정보를 확인합니다.

```
$ aws ec2 describe-images --image-ids aki-70cb0e10 --region region  
  
{  
    "Images": [  
        {  
            "VirtualizationType": "paravirtual",  
            "Name": "pv-grub-hd0_1.05-x86_64.gz",  
            ...  
            "Description": "PV-GRUB release 1.05, 64-bit"  
        }  
    ]  
}
```

여기서 커널 이미지는 PV-GRUB 1.05입니다. 사용자의 PV-GRUB 버전이 최신 버전이 아닌 경우 ([Amazon PV-GRUB 커널 이미지 ID \(p. 141\)](#)에서 확인 가능), 다음 절차를 사용해서 이를 업데이트해야 합니다.

PV-GRUB 버전 업데이트 방법

인스턴스가 PV-GRUB의 이전 버전을 사용하는 경우, 이를 최신 버전으로 업데이트해야 합니다.

- [Amazon PV-GRUB 커널 이미지 ID \(p. 141\)](#)에서 리전 및 프로세스 아키텍처에 대한 최신 PV-GRUB AKI를 확인합니다.
- 인스턴스를 중단합니다. 사용하고 있는 커널 이미지를 수정하려면 인스턴스를 중단할 필요가 있습니다.

```
$ aws ec2 stop-instances --instance-ids instance_id --region region
```

- 인스턴스에 대해 사용되는 커널 이미지를 수정합니다.

```
$ aws ec2 modify-instance-attribute --instance-id instance_id --kernel kernel_id --region region
```

- 인스턴스를 재시작합니다.

```
$ aws ec2 start-instances --instance-ids instance_id --region region
```

Amazon EC2 인스턴스

Amazon EC2를 처음 사용하는 경우 시작하려면 다음 섹션을 참조하십시오.

- [Amazon EC2란 무엇입니까? \(p. 1\)](#)
- [Amazon EC2로 설정 \(p. 16\)](#)
- [Amazon EC2 Linux 인스턴스 시작하기 \(p. 21\)](#)
- [인스턴스 수명 주기 \(p. 261\)](#)

프로덕션 환경을 시작하기 전에 다음 질문에 답해야 합니다.

질문: 어떤 인스턴스 유형이 필요에 가장 잘 맞습니까?

Amazon EC2는 애플리케이션을 실행하는 데 필요한 CPU, 메모리, 스토리지 및 네트워킹 용량을 선택할 수 있는 다양한 인스턴스 유형을 제공합니다. 자세한 내용은 [인스턴스 유형 \(p. 146\)](#) 섹션을 참조하십시오.

질문: 어떤 구매 옵션이 필요에 가장 잘 맞습니까?

Amazon EC2는 온디맨드 인스턴스(기본값), 스팟 인스턴스 및 예약 인스턴스를 지원합니다. 자세한 내용은 [인스턴스 구입 옵션 \(p. 173\)](#) 섹션을 참조하십시오.

질문: 어떤 유형의 루트 볼륨이 필요에 가장 잘 맞습니까?

각 인스턴스는 Amazon EBS 또는 인스턴스 스토어 기반입니다. 필요한 루트 볼륨 유형에 따라 AMI를 선택하십시오. 자세한 내용은 [루트 디바이스 스토리지 \(p. 64\)](#) 섹션을 참조하십시오.

질문: 가상 프라이빗 클라우드 사용에 따른 이점이 있습니까?

EC2-Classic 또는 EC2-VPC에서 인스턴스를 시작할 수 있는 경우 어떤 플랫폼이 필요에 맞는지 결정해야 합니다. 자세한 내용은 [지원되는 플랫폼 \(p. 471\)](#) 및 [Amazon EC2와 Amazon Virtual Private Cloud \(p. 466\)](#) 섹션을 참조하십시오.

Q. 하이브리드 환경에서 EC2 인스턴스 및 머신 집합을 원격으로 관리할 수 있습니까?

Amazon Elastic Compute Cloud (Amazon EC2) Run Command를 사용하여 하이브리드 환경의 Amazon EC2 인스턴스, VM(가상 머신)과 서버 또는 다른 클라우드 공급자가 제공하는 VM의 구성

을 안전하게 원격으로 관리할 수 있습니다. 자세한 내용은 [시스템 관리자 Remote Management \(Run Command\)](#) 단원을 참조하십시오.

인스턴스 유형

인스턴스를 시작할 때 지정하는 인스턴스 유형에 따라 인스턴스에 사용되는 호스트 컴퓨터의 하드웨어가 결정됩니다. 각 인스턴스 유형은 서로 다른 컴퓨팅, 메모리, 스토리지 용량을 제공하는데, 이 용량에 따라 서로 다른 인스턴스 패밀리로 분류됩니다. 인스턴스에서 실행하려는 애플리케이션 또는 소프트웨어의 요구 사항에 따라 인스턴스 유형을 선택하십시오.

Amazon EC2에서는 실제로 사용되는 하드웨어에 관계없이 각 인스턴스에 일정하고 예측 가능한 CPU 용량을 제공합니다.

Amazon EC2는 호스트 컴퓨터에 있는 CPU, 메모리 및 인스턴스 스토리지 등의 일부 리소스를 특정 인스턴스에 전용으로 할당합니다. Amazon EC2는 호스트 컴퓨터의 네트워크 및 디스크 하위 시스템과 같은 기타 리소스를 여러 인스턴스와 공유합니다. 호스트 컴퓨터의 각 인스턴스가 이러한 공유 리소스 중 하나를 최대한 많이 사용하려고 할 경우 해당 리소스는 각 인스턴스에 고르게 분배됩니다. 그러나 리소스 사용률이 저조한 경우에는 리소스에 여유가 있는 한 특정 인스턴스가 해당 리소스를 더 많이 소비할 수 있습니다.

각 인스턴스 유형은 공유 리소스의 최소 성능을 더 많이 제공하거나 더 적게 제공합니다. 예를 들어 I/O 성능이 높은 인스턴스 유형에는 더 많은 뷰의 공유 리소스가 할당됩니다. 더 많은 뷰의 공유 리소스가 할당되면 I/O 성능의 변동성도 감소합니다. 대부분의 애플리케이션에 대해서는 중간 수준의 I/O 성능만으로 충분합니다. 그러나 더욱 높거나 일관적인 I/O 성능이 필요한 애플리케이션에 대해서는 I/O 성능이 높은 인스턴스 유형을 사용하는 것이 좋습니다.

목차

- 사용 가능한 인스턴스 유형 (p. 146)
- 하드웨어 사양 (p. 147)
- 가상화 유형 (p. 148)
- 네트워킹 및 스토리지 기능 (p. 148)
- 인스턴스 제한 (p. 149)

사용 가능한 인스턴스 유형

Amazon EC2에서는 다음 표에 나와 있는 인스턴스 유형을 제공합니다.

현재 세대 인스턴스

최상의 성능을 위해서는 새 인스턴스를 시작할 때 현재 세대 인스턴스 유형을 사용하는 것이 좋습니다. 현재 세대 인스턴스 유형에 대한 자세한 내용은 [Amazon EC2 인스턴스](#) 정보 페이지를 참조하십시오.

인스턴스 패밀리	현재 세대 인스턴스 유형
범용	t2.nano t2.micro t2.small t2.medium t2.large t2.xlarge t2.2xlarge m4.large m4.xlarge m4.2xlarge m4.4xlarge m4.10xlarge m4.16xlarge m3.medium m3.large m3.xlarge m3.2xlarge
컴퓨팅 최적화	c4.large c4.xlarge c4.2xlarge c4.4xlarge c4.8xlarge c3.large c3.xlarge c3.2xlarge c3.4xlarge c3.8xlarge

인스턴스 패밀리	현재 세대 인스턴스 유형
메모리 최적화	r3.large r3.xlarge r3.2xlarge r3.4xlarge r3.8xlarge r4.large r4.xlarge r4.2xlarge r4.4xlarge r4.8xlarge r4.16xlarge x1.16xlarge x1.32xlarge
스토리지 최적화	d2.xlarge d2.2xlarge d2.4xlarge d2.8xlarge i2.xlarge i2.2xlarge i2.4xlarge i2.8xlarge i3.large i3.xlarge i3.2xlarge i3.4xlarge i3.8xlarge i3.16xlarge
액셀러레이티드 컴퓨팅	p2.xlarge p2.8xlarge p2.16xlarge g2.2xlarge g2.8xlarge

이전 세대 인스턴스

Amazon Web Services에서는 이전 세대 인스턴스를 기준으로 애플리케이션을 최적화했으며 아직 업그레이드하지 않은 사용자를 위해 이전 세대 인스턴스를 제공합니다. 최상의 성능을 얻으려면 최신 세대 인스턴스를 사용할 것을 권장합니다. 물론 AWS에서는 이전 세대 인스턴스를 계속 지원할 것입니다. 이전 세대 인스턴스를 사용하고 있는 경우 어떠한 인스턴스로 업그레이드하는 것이 적합한지 알아볼 수 있습니다. 자세한 내용은 [이전 세대 인스턴스](#)를 참조하십시오.

인스턴스 패밀리	이전 세대 인스턴스 유형
범용	m1.small m1.medium m1.large m1.xlarge
컴퓨팅 최적화	c1.medium c1.xlarge cc2.8xlarge
메모리 최적화	m2.xlarge m2.2xlarge m2.4xlarge cr1.8xlarge
스토리지 최적화	hi1.4xlarge hs1.8xlarge
액셀러레이티드 컴퓨팅	cg1.4xlarge
マイ크로 인스턴스	t1.micro

하드웨어 사양

Amazon EC2 인스턴스 유형별 하드웨어 사양에 대한 자세한 내용은 [Amazon EC2 인스턴스](#)를 참조하십시오.

요구 사항에 가장 적합한 인스턴스 유형을 확인하려면 인스턴스를 시작한 후 벤치마크 애플리케이션을 직접 사용해 보는 것이 좋습니다. 과금 기준은 인스턴스 시간이므로 여러 인스턴스 유형을 테스트해 본 후에 결정하는 것이 간편하면서도 경제적입니다.

결정을 내린 후에도 이후에 요구 사항이 변화함에 따라 인스턴스 크기를 조정할 수 있습니다. 자세한 내용은 [인스턴스 크기 조정 \(p. 169\)](#) 섹션을 참조하십시오.

Note

Amazon EC2 인스턴스는 인스턴스 유형 제품 페이지에 지정된 대로 64비트 가상 인텔 프로세서에서 실행됩니다. Amazon EC2 인스턴스 유형별 하드웨어 사양에 대한 자세한 내용은 [Amazon EC2 인스턴스](#)를 참조하십시오. 하지만 64비트 CPU에 대한 업계의 이름 지정 방식 때문에 혼란이 빚어질 수 있습니다. 칩 제조업체 Advanced Micro Devices(AMD)는 최초로 인텔 x86 명령 집합 기반의 64비트 아키텍처를 상용화하는 데 성공했습니다. 그 결과, 이 아키텍처는 칩 제조업체와 상관없이 AMD64로 통용됩니다. Windows와 다수의 Linux 배포가 이 관례를 따릅니다. 인스턴스가 인텔 하드웨어에서 실행되고 있음에도 불구하고 Ubuntu나 Windows EC2 인스턴스에 대한 내부 시스템 정보는 CPU 아키텍처를 AMD64로 표시하는 이유가 이 때문입니다.

가상화 유형

각 인스턴스 유형은 반가상화(PV) 또는 하드웨어 가상 머신(HVM) 가상화 유형 중 하나를 또는 두 유형을 모두 지원합니다. 인스턴스의 가상화 유형은 인스턴스를 시작할 때 사용한 AMI에 의해 결정됩니다.

최상의 성능을 위해 HVM AMI를 사용하는 것이 좋습니다. 또한 향상된 네트워킹을 활용하려면 HVM AMI가 필요합니다. HVM 가상화에는 AWS 플랫폼이 제공하는 하드웨어 보조 기술이 사용됩니다. HVM 가상화를 사용하는 경우 게스트 VM은 기본 하드웨어 플랫폼에 있는 것처럼 실행되지만, 성능 향상을 위해 여전히 PV 네트워크 및 스토리지 드라이버가 사용됩니다. 자세한 내용은 [Linux AMI 가상화 유형 \(p. 66\)](#) 섹션을 참조하십시오.

네트워킹 및 스토리지 기능

인스턴스 유형을 선택하면 사용할 수 있는 네트워킹 및 스토리지 기능이 결정됩니다.

네트워킹 기능

- 일부 인스턴스 유형은 EC2-Classic에서 사용할 수 없으므로 VPC에서 시작해야 합니다. VPC에서 인스턴스를 시작하면 EC2 Classic에서 사용할 수 없는 기능을 활용할 수 있습니다. 예를 들어, 향상된 네트워킹 기능을 사용하고, 인스턴스에 여러 프라이빗 IPv4 주소를 할당하고, 인스턴스에 IPv6 주소를 할당하고, 인스턴스에 할당된 보안 그룹을 변경할 수 있습니다. 자세한 내용은 [VPC에서만 사용할 수 있는 인스턴스 유형 \(p. 470\)](#) 섹션을 참조하십시오.
- 인스턴스 유형의 네트워킹 및 대역폭 성능을 극대화하려면 다음을 수행해볼 수 있습니다.
 - 배치 그룹에 대해 지원되는 인스턴스 유형을 실행하여 HPC(고성능 컴퓨팅) 애플리케이션에 맞게 인스턴스를 최적화합니다. 공통 배치 그룹의 인스턴스는 지역 시간이 짧은 고대역폭(10Gbps) 네트워킹이 유용할 수 있습니다. 자세한 내용은 [배치 그룹 \(p. 527\)](#) 섹션을 참조하십시오. 10Gbps 네트워크 속도를 지원하는 인스턴스 유형은 배치 그룹에서 실행할 경우에만 이러한 네트워크 속도가 도움이 됩니다.
 - PPS(Packet Per Second) 성능을 크게 높이고 네트워크 지터 및 지역 시간을 낮추려면 지원되는 최신 인스턴스 유형에 대해 향상된 네트워킹을 활성화합니다. 자세한 내용은 [Linux에서 향상된 네트워킹 \(p. 533\)](#) 섹션을 참조하십시오.
- 지원되는 최대 MTU는 인스턴스 유형에 따라 다릅니다. 모든 Amazon EC2 인스턴스 유형은 표준 이더넷 V2 1500MTU 프레임을 지원합니다. 모든 현재 세대 인스턴스는 9001MTU 또는 점보 프레임을 지원하며, 일부 이전 세대 인스턴스도 이를 지원합니다. 자세한 내용은 [EC2 인스턴스에 대한 네트워크 MTU\(최대 전송 단위\) \(p. 530\)](#) 섹션을 참조하십시오.

스토리지 기능

- 일부 인스턴스 유형은 EBS 볼륨과 인스턴스 스토어 볼륨을 지원하는 반면, EBS 볼륨만 지원하는 인스턴스 유형도 있습니다. 인스턴스 스토어 볼륨을 지원하는 일부 인스턴스 유형은 SSD(Solid State Drive)를 사용하여 매우 높은 랜덤 I/O 성능을 제공합니다. 자세한 내용은 [스토리지 \(p. 559\)](#) 섹션을 참조하십시오.
- 일부 인스턴스 유형을 EBS 최적화 인스턴스로 시작하면 Amazon EBS I/O 전용 용량을 더 많이 확보할 수 있습니다. 일부 인스턴스 유형은 기본적으로 EBS에 최적화되어 있습니다. 자세한 내용은 [Amazon EBS 최적화 인스턴스 \(p. 614\)](#) 섹션을 참조하십시오.

다음 표에서는 현재 세대 인스턴스 유형에서 지원되는 네트워킹 및 스토리지 기능을 요약합니다.

	VPC 전용	EBS 전용	SSD 볼륨	배치 그룹	HVM 전용	향상된 네트워킹	IPv6 지원 (VPC 전용)
C3			예	예		Intel 82599 VF	예

	VPC 전용	EBS 전용	SSD 볼륨	배치 그룹	HVM 전용	향상된 네트워킹	IPv6 지원(VPC 전용)
C4	예	예		예	예	Intel 82599 VF	예
D2				예	예	Intel 82599 VF	예
G2			예	예	예		
I2			예	예	예	Intel 82599 VF	예
I3	예		예 *	예	예	ENA	예
M3			예				
M4	예	예		예	예	m4.16xlarge: ENA 기타 모든 크기: Intel 82599 VF	예
P2	예	예		예	예	ENA	예
R3			예	예	예	Intel 82599 VF	예
R4	예	예		예	예	ENA	예
T2	예	예			예		예
X1	예		예	예	예	ENA	예

* I3 인스턴스의 루트 디바이스 볼륨은 Amazon EBS 볼륨이어야 합니다.

인스턴스 제한

한 리전에서 시작할 수 있는 총 인스턴스 수에는 제한이 있으며, 일부 인스턴스 유형에는 또 다른 제한이 있습니다.

기본 제한에 대한 자세한 내용은 [Amazon EC2에서 실행 가능한 인스턴스 수](#)를 참조하십시오.

이러한 제한을 보거나 현재 제한 증가를 요청하는 방법은 [Amazon EC2 서비스 제한 \(p. 688\)](#) 섹션을 참조하십시오.

T2 인스턴스

T2 인스턴스는 중간 정도의 기본 성능을 발휘하면서 워크로드의 필요에 따라 성능을 크게 높이는 버스트 기능을 제공하도록 설계되었습니다. 이러한 인스턴스는 CPU의 최대 성능을 자주 또는 일관적으로 사용하지 않지만 가끔 순간적인 버스트가 필요한 워크로드에 적합합니다. T2 인스턴스는 웹 서버, 개발자 환경, 소규모 데이터베이스와 같은 범용 워크로드에 매우 적합합니다. T2 인스턴스의 요금에 대한 자세한 내용 및 기타 하드웨어 세부 정보는 [Amazon EC2 인스턴스](#)를 참조하십시오.

계정이 12개월이 아직 안 된 경우 특정 사용 한도 내에서 무료로 t2.micro 인스턴스를 사용할 수 있습니다.
자세한 내용은 [AWS 프리 티어](#) 섹션을 참조하십시오.

목차

- [하드웨어 사양 \(p. 150\)](#)
- [T2 인스턴스 요구 사항 \(p. 150\)](#)
- [CPU 크레딧 \(p. 150\)](#)
- [CPU 크레딧 모니터링 \(p. 152\)](#)

하드웨어 사양

Amazon EC2 인스턴스 유형별 하드웨어 사양에 대한 자세한 내용은 [Amazon EC2 인스턴스](#)를 참조하십시오.

T2 인스턴스 요구 사항

다음은 T2 인스턴스에 대한 기본 요구 사항입니다.

- Virtual Private Cloud(VPC)로 T2 인스턴스를 시작해야 하며 EC2-Classic 플랫폼에서는 지원되지 않습니다. Amazon VPC에서는 AWS 리소스를 사용자가 정의한 가상 네트워크로 시작할 수 있습니다. EC2-Classic의 기존 인스턴스의 인스턴스 유형을 T2 인스턴스 유형으로 변경할 수는 없습니다. EC2-Classic 및 EC2-VPC에 대한 자세한 내용은 [지원되는 플랫폼 \(p. 471\)](#) 섹션을 참조하십시오. VPC 전용 인스턴스를 시작하는 방법에 대한 자세한 내용은 [VPC에서만 사용할 수 있는 인스턴스 유형 \(p. 470\)](#) 섹션을 참조하십시오.
- HVM AMI를 사용해서 T2 인스턴스를 실행해야 합니다. 자세한 내용은 [Linux AMI 가상화 유형 \(p. 66\)](#) 섹션을 참조하십시오.
- EBS 볼륨을 루트 디바이스로 사용하여 T2 인스턴스를 시작해야 합니다. 자세한 내용은 [Amazon EC2 루트 디바이스 볼륨 \(p. 11\)](#) 섹션을 참조하십시오.
- T2 인스턴스는 온디맨드 인스턴스 및 예약 인스턴스로 사용 가능하고 스팟 인스턴스, 예약 인스턴스 또는 전용 인스턴스로는 사용할 수 없습니다. 전용 호스트에서도 지원되지 않습니다. 이러한 옵션에 대한 자세한 내용은 [인스턴스 구입 옵션 \(p. 173\)](#) 섹션을 참조하십시오.
- 한 리전에서 시작할 수 있는 총 인스턴스 수에는 제한이 있으며, 일부 인스턴스 유형에는 또 다른 제한이 있습니다. 기본적으로 최대 20개의 T2 인스턴스를 동시에 실행할 수 있습니다. T2 인스턴스가 더 많이 필요한 경우 [Amazon EC2 인스턴스 요청 양식](#)을 사용하여 요청할 수 있습니다.
- 선택한 T2 인스턴스의 크기가 운영 체제 및 애플리케이션의 최소 메모리 요구 사항을 충족하는지 확인합니다. 그래픽 사용자 인터페이스에서 많은 메모리와 CPU 리소스를 사용하는 운영 체제(예: Windows)에서는 대부분의 경우 인스턴스 크기가 t2.micro 이상이어야 합니다. 시간이 지나면서 워크로드 메모리 및 CPU 요구 사항이 증가함에 따라 보다 큰 규모의 T2 인스턴스 또는 다른 EC2 인스턴스 유형으로 확장할 수 있습니다.

CPU 크레딧

CPU 크레딧은 1분 동안 CPU 코어의 전체 성능을 제공합니다. 기존 Amazon EC2 인스턴스 유형은 고정된 성능을 제공하는 반면 T2 인스턴스는 기본 수준의 CPU 성능을 발휘하면서 기본 수준 이상으로 버스트하는 기능을 제공합니다. 기본 성능과 버스트 기능은 CPU 크레딧에 의해 좌우됩니다.

CPU 크레딧이란?

CPU 크레딧 하나는 1분 동안 100%의 사용률로 실행되는 vCPU 하나에 해당합니다. vCPU, 사용률 및 시간의 여러 가지 조합이 CPU 크레딧 하나에 해당합니다. 예를 들어, vCPU 하나가 2분 동안 50%의 사용률로 실행되거나, vCPU 2개가 2분 동안 25%의 사용률로 실행될 수 있습니다.

CPU 크레딧이 지급되는 방식

각 T2 인스턴스는 상당한 양의 초기 CPU 크레딧 잔고로 시작되며 인스턴스 크기에 따라 지속적으로(밀리초 수준의 시간 정밀도) 특정 비율의 시간당 CPU 크레딧을 지급받습니다. 크레딧이 축적되는지 아니면 소비되는지를 결정하는 산정 프로세스도 밀리초 수준의 시간 정밀도로 수행되므로 CPU 크레딧 과소비를 염려할 필요는 없습니다. 즉, 짧은 CPU 버스트는 약간의 CPU 크레딧만을 소비합니다.

T2 인스턴스가 기본 성능 수준에 허용되는 것보다 적은 CPU 리소스를 사용하는 경우(예: 유휴 상태일 때), CPU 크레딧 미사용분 또는 지급된 크레딧과 소비된 크레딧의 차이가 크레딧 잔고에 최대 24시간 동안 저장되어 버스트에 대비할 CPU 크레딧이 축적됩니다. T2 인스턴스가 기본 성능 수준에 허용되는 것보다 많은 CPU 리소스를 필요로 하면 CPU 크레딧 잔고의 크레딧이 소비되어 최대 100%의 사용률로 버스트됩니다. T2 인스턴스가 보유한 CPU 리소스의 크레딧이 많을수록 추가 성능이 필요할 때 기본 성능 수준을 초과하여 버스트할 수 있는 시간을 늘릴 수 있습니다.

다음 표에서는 시작 시에 지급되는 초기 CPU 크레딧 할당, CPU 크레딧 지급 비율, 전체 코어 성능의 백분율로 나타낸 기본 성능 수준(단일 vCPU 사용) 및 인스턴스가 축적할 수 있는 최대 지급된 CPU 크레딧 잔고를 보여줍니다.

인스턴스 유형	최초 CPU 크레딧*	시간당 지급되는 CPU 크레딧	vCPUs	기본 성능(CPU 사용률)	최대 지급된 CPU 크레딧 잔고***
t2.nano	30	3	1	5%	72
t2.micro	30	6	1	10%	144
t2.small	30	12	1	20%	288
t2.medium	60	24	2	40%(최대 200%)**	576
t2.large	60	36	2	60%(최대 200%)**	864
t2.xlarge	120	54	4	90%(최대 400%)**	1296
t2.2xlarge	240	81	8	135%(최대 800%)**	1944

* 초기 CPU 크레딧과 함께 시작될 수 있는 T2 인스턴스의 수에는 제한이 있습니다. 이 제한은 기본적으로 계정당 24시간 동안 리전별 T2 인스턴스 100회 시작으로 설정됩니다. 이 제한을 높이려는 경우 [Amazon EC2 크레딧 기반 인스턴스 시작 크레딧 양식](#)을 사용하여 고객 지원 제한 증가 요청을 제출할 수 있습니다. 계정에서 24시간 동안 T2 인스턴스를 100회 이상 시작하지 않는 경우 이 제한은 사용자에게 아무 영향도 없습니다.

** t2.medium 및 대형 인스턴스에는 두 개 이상의 vCPU가 있습니다. 표의 기본 성능은 단일 vCPU 사용량의 백분율입니다(여러 vCPU로 성능 분할 가능). 인스턴스의 기본 CPU 사용률을 계산하려면 결합된 vCPU 백분율을 vCPU 개수로 나눕니다. 예를 들어 t2.large의 기본 성능 1 vCPU의 60%입니다. t2.large 인스턴스에 vCPU가 2개 있으므로 기본 성능으로 작동 중인 t2.large 인스턴스의 CPU 사용률은 CloudWatch CPU 측정치에서 30%로 표시됩니다.

*** 이 최대값에는 처음 사용되고 만료되지 않은 초기 CPU 크레딧이 포함되지 않습니다. 예를 들어, 시작된 후 24시간 이상 유휴 상태로 지속된 t2.micro 인스턴스가 최대 174(초기 CPU 크레딧 30 + 지급된 크레딧 144)의 크레딧 잔고에 도달할 수 있습니다. 그러나 인스턴스가 초기 CPU 크레딧 30을 사용한 후 인스턴스를 중지하고 시작하여 새로운 초기 CPU 크레딧 잔고가 발행되지 않는 한, 크레딧 잔고가 144를 초과할 수 없습니다.

초기 크레딧 잔고는 원활한 시작 환경을 제공하기 위한 것입니다. 인스턴스의 최대 지급된 크레딧 잔고는 24시간 동안 지급되는 시간당 CPU 크레딧 수와 일치합니다. 예를 들어 t2.micro 인스턴스는 시간당 6의 CPU 크레딧을 지급받으며 지급되는 최대 144의 CPU 크레딧 잔고를 축적할 수 있습니다.

CPU 크레딧은 시간이 지나면 소멸되나요?

초기 CPU 크레딧은 만료되지 않지만, 인스턴스에서 CPU 크레딧을 사용할 때 처음 사용됩니다. 5분 간격으로 지급된 미사용 크레딧이 지급 24시간 후에 소멸되고, 새로 지급된 크레딧이 추가되기 전에 소멸된 크레딧은 해당 시점에 CPU 크레딧 잔고에서 차감됩니다. 또한 인스턴스의 CPU 크레딧 잔고는 인스턴스 종지 후 시작 시에 유지되지 않습니다. 인스턴스를 종지하면 모든 크레딧 잔고가 소멸되고, 인스턴스를 다시 시작하면 초기 크레딧 잔고가 다시 지급됩니다.

예를 들어 `t2.small` 인스턴스의 CPU 사용률이 1시간 동안 5%인 경우 60분의 5%인 3 CPU 크레딧을 소비한 것입니다. 이 1시간 동안 12 CPU 크레딧이 지급되었으므로 차액인 9 CPU 크레딧이 CPU 크레딧 잔고에 가산됩니다. 이때 잔고에서 소멸 기한인 24시간에 도달한 CPU 크레딧(인스턴스가 24시간 전에 완전히 유휴 상태였을 경우 최대 12크레딧)은 잔고에서 모두 차감됩니다. 소멸된 크레딧이 지급된 크레딧보다 많으면 크레딧 잔고가 감소합니다. 반대로, 소멸된 크레딧이 지급된 크레딧보다 적으면 크레딧 잔고가 증가합니다.

크레딧이 모두 소진되면 어떻게 되나요?

인스턴스에서 CPU 크레딧 잔고를 모두 소진한 경우 성능이 기본 성능 수준으로 유지됩니다. 인스턴스의 크레딧이 부족해지면 인스턴스의 CPU 크레딧 소비, 즉 CPU 성능이 기본 성능 수준까지 15분 간격으로 점차 감소하므로 CPU 크레딧이 고갈되어도 급격한 성능 저하가 체감되지는 않습니다. 인스턴스가 지속적으로 CPU 크레딧 잔고를 모두 소진하는 경우 T2의 크기를 늘리거나 M3, C3 등의 고정 성능 인스턴스 유형을 사용하는 것이 좋습니다.

CPU 크레딧 모니터링

Amazon EC2에 있는 각 T2 인스턴스의 크레딧 잔고를 CloudWatch 콘솔의 인스턴스별 측정치로 확인할 수 있습니다. T2 인스턴스에는 `CPUCreditUsage` 및 `CPUCreditBalance`의 2가지 측정치가 있습니다. `CPUCreditUsage` 측정치는 측정 기간 중에 소비된 CPU 크레딧 수를 나타냅니다. `CPUCreditBalance` 측정치는 T2 인스턴스에 지급된 미사용 CPU 크레딧 수를 나타냅니다. 버스트 중에는 CPU 크레딧 소비 속도가 지급 속도보다 빠르므로 잔고가 고갈됩니다.

다음 표에서는 새롭게 제공되는 CloudWatch 측정치를 설명합니다. CloudWatch에서 이러한 측정치를 사용하는 방법에 대한 자세한 내용은 [인스턴스에 대해 얻을 수 있는 CloudWatch 측정치 나열 \(p. 349\)](#) 섹션을 참조하십시오.

지표	설명
<code>CPUCreditUsage</code>	[T2 인스턴스] 인스턴스가 소비한 CPU 크레딧 수입니다. CPU 크레딧 하나는 1분 동안 100%의 사용률로 실행되는 vCPU 1개 또는 이와 동등한 vCPU, 사용률 및 시간의 조합과 동일합니다(예를 들어 2분 동안 50%의 사용률로 실행되는 vCPU 1개 또는 2분 동안 25%의 사용률로 실행되는 vCPU 2개). CPU 크레딧 측정치는 5분 간격으로만 제공됩니다. 5분 이상의 시간을 지정할 경우 <code>Average</code> 통계 대신 <code>Sum</code> 통계를 사용하십시오. 단위: 수
<code>CPUCreditBalance</code>	[T2 인스턴스] 인스턴스에 대해 기본 CPU 사용률 이상으로 버스트가 가능한 CPU 크레딧 수입니다. 크레딧은 측정 이후에는 크레딧 잔고에 보관되고, 만료 이후에는 크레딧 잔고에서 소멸됩니다. 크레딧은 측정 이후 24시간이 지나면 만료됩니다. CPU 크레딧 측정치는 5분 간격으로만 제공됩니다. 단위: 수

컴퓨팅 최적화 인스턴스

컴퓨팅 최적화 인스턴스는 고성능 프로세서의 이점을 활용하는 컴퓨팅 위주의 애플리케이션에 적합합니다. 컴퓨팅 최적화 인스턴스는 다음 애플리케이션에 매우 적합합니다.

- 일괄 처리 작업
- 미디어 트랜스코딩
- 트래픽이 많은 웹 서버, MMO(Massively Multiplayer Online) 게임 서버, 광고 서비스 엔진
- HPC(고성능 컴퓨팅) 및 기타 컴퓨팅 집약적 애플리케이션

목차

- 하드웨어 사양 (p. 153)
- 컴퓨팅 인스턴스 성능 (p. 153)
- 컴퓨팅 인스턴스 기능 (p. 153)
- 36개의 vCPU 지원 (p. 154)
- 인스턴스 제한 (p. 155)

하드웨어 사양

Amazon EC2 인스턴스 유형별 하드웨어 사양에 대한 자세한 내용은 [Amazon EC2 인스턴스](#)를 참조하십시오.

컴퓨팅 인스턴스 성능

EBS에 최적화된 인스턴스를 사용하면 Amazon EBS I/O와 인스턴스의 다른 네트워크 간의 경합을 제거하여 EBS 볼륨에 대해 일관되게 우수한 성능을 제공할 수 있습니다. C4 인스턴스는 추가 비용 없이 기본적으로 EBS에 최적화되어 있습니다. 낮은 시간당 요금 추가로 C3 인스턴스에 대해 EBS 최적화를 활성화할 수 있습니다. 자세한 내용은 [Amazon EBS 최적화 인스턴스 \(p. 614\)](#) 섹션을 참조하십시오.

향상된 네트워킹 기능을 사용할 수도 있습니다. 향상된 네트워킹을 통해 PPS(Packet Per Second) 성능이 크게 높아지고, 네트워크 지터 및 지연 시간이 낮아집니다. 자세한 내용은 [Linux에서 향상된 네트워킹 \(p. 533\)](#) 섹션을 참조하십시오.

c4.8xlarge 인스턴스 유형은 Linux에서 프로세서 C 상태 및 P 상태를 제어할 수 있는 기능을 제공합니다. C 상태는 유휴 상태일 때 코어가 진입하는 절전 수준을 제어하고, P 상태는 코어의 성능(CPU 주파수)을 제어합니다. 자세한 내용은 [EC2 인스턴스에 대한 프로세서 상태 제어 \(p. 306\)](#) 섹션을 참조하십시오.

컴퓨팅 인스턴스 기능

컴퓨팅 최적화 인스턴스에 대한 기능은 다음과 같이 간략히 설명할 수 있습니다.

	VPC 전용	EBS 전용	SSD 볼륨	배치 그룹	HVM 전용	향상된 네트워킹
C3			예	예		Intel 82599 VF
C4	예	예		예	예	Intel 82599 VF

자세한 내용은 다음 자료를 참조하십시오.

- [VPC에서만 사용할 수 있는 인스턴스 유형 \(p. 470\)](#)
- [Amazon EBS 최적화 인스턴스 \(p. 614\)](#)
- [Amazon EC2 인스턴스 스토어 \(p. 642\)](#)

- 배치 그룹 (p. 527)
- Linux에서 향상된 네트워킹 (p. 533)

36개의 vCPU 지원

c4.8xlarge 인스턴스 유형은 36개의 vCPU를 지원하므로 vCPU가 32개로 제한되는 일부 Linux 운영 체제에서 시작 문제가 발생할 수 있습니다. 따라서 c4.8xlarge 인스턴스를 시작할 때 최신 AMI를 사용하실 것을 적극 권장합니다.

다음 AMI는 36개의 vCPU로 c4.8xlarge 인스턴스 시작을 지원합니다.

- Amazon Linux AMI 2017.03(HVM)
- Ubuntu Server 14.04 LTS(HVM)
- Red Hat Enterprise Linux 7.1(HVM)
- SUSE Linux Enterprise Server 12(HVM)

애플리케이션에 대해 다른 AMI를 사용해야 하지만 c4.8xlarge 인스턴스가 완전히 시작되지 않은 경우(예: Client.InstanceInitiatedShutdown 상태 전환으로 인해 시작 중에 인스턴스 상태가 stopped로 변경된 경우) c4.8xlarge 인스턴스 유형을 사용할 수 있도록 다음 절차에 따라 32개보다 많은 vCPU를 지원하도록 AMI를 수정하십시오.

32개보다 많은 vCPU를 지원하도록 인스턴스를 업데이트하려면

1. AMI를 사용하여 C4 인스턴스를 시작하여 c4.8xlarge 이외의 C4 인스턴스 유형을 선택합니다.
2. 운영 체제 관련 지침에 따라 커널을 최신 버전으로 업데이트합니다. 예를 들어, RHEL 6의 경우 다음 명령을 사용합니다.

```
sudo yum update -y kernel
```

3. 인스턴스를 종지합니다.
4. (선택 사항) 나중에 필요한 추가 c4.8xlarge 인스턴스를 시작하는 데 사용할 수 있는 인스턴스에서 AMI를 생성합니다.
5. 종지된 인스턴스의 인스턴스 유형을 c4.8xlarge로 변경합니다(Actions, Instance Settings, Change Instance Type을 선택한 다음 지침을 따름).
6. 인스턴스를 시작합니다. 인스턴스가 올바로 시작되면 완료된 것입니다. 그래도 인스턴스가 올바로 부팅되지 않으면 다음 단계로 진행하십시오.
7. (선택 사항) 그래도 인스턴스가 올바로 부팅되지 않으면 인스턴스의 커널이 32개를 초과하는 vCPU를 지원하지 않을 수도 있습니다. 하지만 vCPU 수를 제한하면 인스턴스를 부팅할 수 있습니다.
 - a. 종지된 인스턴스의 인스턴스 유형을 c4.8xlarge가 아닌 다른 C4 인스턴스 유형으로 변경합니다 (Actions, Instance Settings, Change Instance Type을 선택한 다음 지침을 따름).
 - b. 운영 체제 관련 지침에 따라 maxcpus=32 옵션을 부팅 커널 파라미터에 추가합니다. 예를 들어, RHEL 6의 경우 /boot/grub/menu.lst 파일을 편집하고 다음 옵션을 가장 최근 활성 kernel 항목에 추가합니다.

```
default=0
timeout=1
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux Server (2.6.32-504.3.3.el6.x86_64)
root (hd0,0)
kernel /boot/vmlinuz-2.6.32-504.3.3.el6.x86_64 maxcpus=32 console=ttyS0 ro
root=UUID=9996863e-b964-47d3-a33b-3920974fdbd9 rd_NO_LUKS KEYBOARDTYPE=pc
```

```
KEYTABLE=us LANG=en_US.UTF-8 xen_blkfront.sda_is_xvda=1 console=ttyS0,115200n8
console=tty0 rd_NO_MD SYSFONT=latarcyrheb-sun16 crashkernel=auto rd_NO_LVM
rd_NO_DM
initrd /boot/initramfs-2.6.32-504.3.3.el6.x86_64.img
```

- c. 인스턴스를 중지합니다.
- d. (선택 사항) 나중에 필요한 추가 c4.8xlarge 인스턴스를 시작하는 데 사용할 수 있는 인스턴스에서 AMI를 생성합니다.
- e. 중지된 인스턴스의 인스턴스 유형을 c4.8xlarge로 변경합니다(Actions, Instance Settings, Change Instance Type을 선택한 다음 지침을 따름).
- f. 인스턴스를 시작합니다.

인스턴스 제한

- 64비트 HVM AMI가 필요한 C4 인스턴스는 최대 244GiB RAM의 고용량 메모리를 보유하며, 이 용량을 활용하기 위해 64비트 운영 체제를 필요로 합니다. HVM AMI는 고용량 메모리 인스턴스 유형의 반가상화(PV) AMI보다 우수한 성능을 제공합니다. 또한 향상된 네트워킹을 활용하려면 HVM AMI를 사용해야 합니다.
- 한 리전에서 시작할 수 있는 총 인스턴스 수에는 제한이 있으며, 일부 인스턴스 유형에는 또 다른 제한이 있습니다. 자세한 내용은 [Amazon EC2에서 실행 가능한 인스턴스 수 섹션을 참조하십시오](#).. 한도 증가를 요청하려면 [Amazon EC2 인스턴스 요청 양식](#)을 사용하십시오.

메모리 최적화 인스턴스

메모리 최적화 인스턴스는 메모리에서 대규모 데이터를 처리하는 워크로드에 대해 빠른 성능을 제공하도록 설계되었습니다.

R4 인스턴스

R4 인스턴스는 다음 애플리케이션에 적합합니다.

- 고성능 관계형(MySQL) 및 NoSQL(MongoDB, Cassandra) 데이터베이스.
- 키-값 유형 데이터의 인 메모리 캐싱을 제공하는 분산된 웹 규모 캐시 저장소(Memcached 및 Redis).
- 비즈니스 인텔리전스를 위해 최적화된 데이터 스토리지 형식과 분석을 사용하는 인 메모리 데이터베이스(예: SAP HANA).
- 대용량 비정형 데이터를 실시간으로 처리하는 애플리케이션(금융 서비스, Hadoop/Spark 클러스터).
- HPC(고성능 컴퓨팅) 및 EDA(전자 설계 자동화) 애플리케이션.

X1 인스턴스

X1 인스턴스는 다음 애플리케이션에 적합합니다.

- SAP HANA와 같은 인 메모리 데이터베이스[Business Suite S/4HANA, Business Suite on HANA(SoH), Business Warehouse on HANA(BW) 및 Data Mart Solutions on HANA에 대한 SAP 인증 지원 포함]. 자세한 내용은 [SAP HANA on the AWS Cloud](#) 섹션을 참조하십시오.
- Apache Spark나 Presto와 같은 빅데이터 처리 엔진.
- 고성능 컴퓨팅(HPC) 애플리케이션.

R3 인스턴스

R3 인스턴스는 다음 애플리케이션에 적합합니다.

- 고성능 관계형(MySQL) 및 NoSQL(MongoDB, Cassandra) 데이터베이스.
- 인 메모리 분석.
- 게임 조립 및 분석.
- 엔터프라이즈 애플리케이션(예: Microsoft SharePoint).

하드웨어 사양

Amazon EC2 인스턴스 유형별 하드웨어 사양에 대한 자세한 내용은 [Amazon EC2 인스턴스](#)를 참조하십시오.

메모리 성능

R4 인스턴스는 최대 488GiB RAM을 지원합니다.

X1 인스턴스에는 Intel 확장형 메모리 버퍼가 포함되어 있어, 300GiB/s의 지속 가능 메모리 읽기 대역폭과 140GiB/s의 지속 가능 메모리 쓰기 대역폭을 제공합니다.

R3 인스턴스는 최대 244GiB RAM을 지원합니다.

메모리 최적화 인스턴스는 고용량 메모리를 보유하며, 이 용량을 활용하기 위해 64비트 HVM AMI가 필요합니다. HVM AMI는 고용량 메모리 인스턴스 유형의 반가상화(PV) AMI보다 우수한 성능을 제공합니다. 자세한 내용은 [Linux AMI 가상화 유형 \(p. 66\)](#) 섹션을 참조하십시오.

컴퓨팅 성능

R4 인스턴스는 최대 64개의 vCPU를 사용할 수 있는 것이 특징이며, 인 메모리 애플리케이션의 성능을 강화하기 위해 고용량 메모리 대역폭과 대용량 L3 캐시가 특징인 E5-2686v4 기반의 AWS 맞춤형 Intel Xeon 프로세서 2개로 작동됩니다.

X1 인스턴스는 최대 128개의 vCPU를 사용할 수 있는 것이 특징이며, 인 메모리 애플리케이션의 성능을 강화하기 위해 고용량 메모리 대역폭과 대용량 L3 캐시가 특징인 Intel Xeon E7-8880 v3 프로세서 4개로 작동됩니다.

메모리 최적화 인스턴스는 최신 Intel AES-NI 기능을 통해 암호화 성능을 끌어올릴 수 있고, Intel TSX(Transactional Synchronization Extensions)를 지원하여 인 메모리 트랜잭션 데이터 처리의 성능을 강화하며, Intel AVX2(Advanced Vector Extensions 2) 프로세서 지원을 지원하여 대부분의 정수 명령을 256비트로 확장합니다.

일부 메모리 최적화 인스턴스는 Linux에서 프로세서 C 상태 및 P 상태를 제어할 수 있는 기능을 제공합니다. C 상태는 유류 상태일 때 코어가 진입하는 절전 수준을 제어하고, P 상태는 코어의 성능(CPU 주파수로 측정)을 제어합니다. 자세한 내용은 [EC2 인스턴스에 대한 프로세서 상태 제어 \(p. 306\)](#) 섹션을 참조하십시오.

네트워크 성능

메모리 최적화 인스턴스의 네트워크 성능을 강화하기 위해 향상된 네트워킹을 지원합니다. 자세한 내용은 [Linux에서 향상된 네트워킹 \(p. 533\)](#) 섹션을 참조하십시오.

R4 인스턴스는 ENA(Elastic Network Adapter)를 이용해 뛰어난 PPS(Packet Per Second) 성능과 낮은 지연 시간을 제공합니다. 대부분의 애플리케이션은 항상 높은 수준의 네트워크 성능을 필요로 하지 않지만, 데이터를 주고 받을 때 넓은 대역폭에 액세스 할 수 있을 경우 유익할 수 있습니다. 작은 크기의 R4 인스턴스는 10Gbps의 최대 처리 속도를 보장합니다. 이러한 인스턴스는 네트워크 I/O 크레딧 메커니즘을 이용해 평균 대역폭 활용도를 기준으로 인스턴스에 네트워크 대역폭을 할당합니다. 이러한 인스턴스의 네트워크 처리 속도가 기준 한도 미만으로 떨어지면 크레딧이 발생하는데, 이 크레딧은 네트워크 데이터를 전송할 때 사용할 수 있습니다. 지속적으로 10Gbps 이상의 대역폭에 액세스해야 하는 워크로드의 경우, `r4.8xlarge` 및

r4.16xlarge 인스턴스 사용을 권장합니다. 그러면 각각 10Gbps 및 20Gbps의 네트워크 대역폭을 활용할 수 있습니다.

인스턴스 기능

메모리 최적화 인스턴스에 대한 기능은 다음과 같이 간략히 설명할 수 있습니다.

	VPC 전용	EBS 전용	SSD 볼륨	배치 그룹	향상된 네트워킹
R3			예	예	Intel 82599 VF
R4	예	예		예	ENI
X1	예		예	예	ENI

자세한 내용은 다음 자료를 참조하십시오.

- [VPC에서만 사용할 수 있는 인스턴스 유형 \(p. 470\)](#)
- [Amazon EBS 최적화 인스턴스 \(p. 614\)](#)
- [Amazon EC2 인스턴스 스토어 \(p. 642\)](#)
- [배치 그룹 \(p. 527\)](#)
- [Linux에서 향상된 네트워킹 \(p. 533\)](#)

개의 vCPU 지원

메모리 최적화 인스턴스는 다수의 vCPU를 지원하므로 vCPU 수가 제한된 운영 체제에서 시작 문제가 발생할 수 있습니다. 따라서 메모리 최적화 인스턴스를 시작할 때 최신 AMI를 사용하실 것을 적극 권장합니다.

다음은 메모리 최적화 인스턴스 시작을 지원하는 AMI입니다.

- Amazon Linux AMI 2016.03(HVM) 이상
- Ubuntu Server 14.04 LTS(HVM)
- Red Hat Enterprise Linux 7.1(HVM)
- SUSE Linux Enterprise Server 12 SP1(HVM)
- Windows Server 2016
- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2 64비트
- Windows Server 2008 SP2 64비트
- Windows Server 2003 R2 64비트

인스턴스 제한

- Windows Server 2008 R2 64비트 AMI를 사용하여 **r4.large** 및 **r4.4xlarge** 인스턴스를 시작할 수 없습니다.
- Windows Server 2008 SP2 64비트 AMI 또는 Windows Server 2003 R2 64비트 AMI를 사용하여 X1 인스턴스를 시작할 수 없습니다. 단 **x1.16xlarge** 인스턴스는 예외입니다.
- 한 리전에서 시작할 수 있는 총 인스턴스 수에는 제한이 있으며, 일부 인스턴스 유형에는 또 다른 제한이 있습니다. 자세한 내용은 [Amazon EC2에서 실행 가능한 인스턴스 수 섹션](#)을 참조하십시오.. 한도 증가를 요청하려면 [Amazon EC2 인스턴스 요청 양식](#)을 사용하십시오.

스토리지 최적화 인스턴스

스토리지 최적화 인스턴스는 로컬 스토리지의 초대형 데이터 세트에 대한 순차적 읽기 및 쓰기 액세스가 많이 필요한 작업에 적합하도록 설계되었습니다. 낮은 지연 시간의 임의의 IOPS(초당 I/O 작업)를 만 단위 수준으로 애플리케이션에 제공할 수 있도록 최적화되어 있습니다.

D2 인스턴스

D2 인스턴스는 다음 애플리케이션에 적합합니다.

- 대량 병렬 처리(MPP) 데이터 웨어하우스
- MapReduce 및 Hadoop 분산 컴퓨팅
- 로그 또는 데이터 처리 애플리케이션

I2 인스턴스

I2 인스턴스는 다음 애플리케이션에 적합합니다.

- NoSQL 데이터베이스
- 클러스터링된 데이터베이스
- OLTP(온라인 트랜잭션 처리) 시스템

I3 인스턴스

I3 인스턴스는 다음 애플리케이션에 적합합니다.

- 빈도가 높은 온라인 트랜잭션 처리(OLTP) 시스템
- 관계형 데이터베이스
- NoSQL 데이터베이스
- 인 메모리 데이터베이스의 캐시(예: Redis)
- 데이터 웨어하우징 애플리케이션
- 지연 시간이 짧은 광고 기술 처리 애플리케이션

목차

- [하드웨어 사양](#) (p. 158)
- [스토리지 성능](#) (p. 159)
- [SSD I/O 성능](#) (p. 159)
- [스토리지 인스턴스 기능](#) (p. 160)
- [개의 vCPU 지원](#) (p. 160)
- [인스턴스 제한](#) (p. 161)

하드웨어 사양

D2 인스턴스의 기본 데이터 스토리지는 HDD 인스턴스 스토어 볼륨입니다. I2 인스턴스의 기본 데이터 스토리지는 SATA SSD 인스턴스 스토어 볼륨입니다. I3 인스턴스의 기본 데이터 스토리지는 NVMe(Non-Volatile Memory Express) SSD 인스턴스 스토어 볼륨입니다.

인스턴스 스토어 볼륨은 인스턴스의 수명 기간 동안만 유지됩니다. 인스턴스가 종료되거나 종료되면 인스턴스 스토어 볼륨의 애플리케이션과 데이터는 삭제됩니다. 따라서 정기적으로 인스턴스 스토어 볼륨에 중요한 데이터를 백업 또는 복제하는 것이 좋습니다. 자세한 내용은 [Amazon EC2 인스턴스 스토어 \(p. 642\)](#) 및 [SSD 인스턴스 스토어 볼륨 \(p. 649\)](#) 섹션을 참조하십시오.

Amazon EC2 인스턴스 유형별 하드웨어 사양에 대한 자세한 내용은 [Amazon EC2 인스턴스](#)를 참조하십시오.

스토리지 성능

Linux의 인스턴스에서 최상의 디스크 처리량 성능을 보장하려면 최신 Amazon Linux AMI 버전을 사용하는 것이 좋습니다.

NVMe 인스턴스 스토어 볼륨이 있는 인스턴스의 경우, 커널 버전이 4.4 이상인 Linux AMI를 사용해야 합니다. 그렇지 않으면 인스턴스가 최대 IOPS 성능을 낼 수 없습니다.

디스크 처리량과 확장성을 크게 향상하는 Xen 블록 링 프로토콜의 확장인 지속적 권한 부여를 지원하는 Linux 커널을 사용하면 D2 인스턴스가 최상의 디스크 성능을 제공합니다. 지속적 권한 부여에 대한 자세한 내용은 Xen Project Blog의 [기사](#)를 참조하십시오.

EBS에 최적화된 인스턴스를 사용하면 Amazon EBS I/O와 인스턴스의 다른 네트워크 간의 경합을 제거하여 EBS 볼륨에 대해 일관되게 우수한 성능을 제공할 수 있습니다. D2 인스턴스는 추가 비용 없이 기본적으로 EBS에 최적화되어 있습니다. 낮은 시간당 요금 추가로 I2 인스턴스에 대해 EBS 최적화를 활성화할 수 있습니다. 자세한 내용은 [Amazon EBS 최적화 인스턴스 \(p. 614\)](#) 섹션을 참조하십시오.

향상된 네트워킹 기능을 사용할 수도 있습니다. 향상된 네트워킹을 통해 PPS(Packet Per Second) 성능이 크게 높아지고, 네트워크 지터 및 지연 시간이 낮아집니다. 자세한 내용은 [Linux에서 향상된 네트워킹 \(p. 533\)](#)을(를) 참조하십시오.

d2.8xlarge 및 i3.16xlarge 인스턴스 유형은 Linux에서 프로세서 C 상태 및 P 상태를 제어할 수 있는 기능을 제공합니다. C 상태는 유휴 상태일 때 코어가 진입하는 절전 수준을 제어하고, P 상태는 코어의 성능(CPU 주파수)을 제어합니다. 자세한 내용은 [EC2 인스턴스에 대한 프로세서 상태 제어 \(p. 306\)](#) 섹션을 참조하십시오.

SSD I/O 성능

커널 버전이 4.4 이상인 Linux AMI를 사용하고 인스턴스에서 사용 가능한 모든 SSD 기반 인스턴스 스토어 볼륨을 활용하는 경우, 다음 표와 같은 IOPS(블록 크기 4,096바이트) 성능을 얻을 수 있습니다(대기열 깊이 포함 상태에서). 그렇지 않으면 더 낮은 IOPS 성능을 얻게 됩니다.

인스턴스 크기	100% 임의 읽기 IOPS	IOPS 첫 쓰기
i2.xlarge	35,000	35,000
i2.2xlarge	75,000	75,000
i2.4xlarge	175,000	155,000
i2.8xlarge	365,000	315,000
i3.large *	100,125	9,375
i3.xlarge *	206,250	18,750
i3.2xlarge	412,500	37,500
i3.4xlarge	825,000	75,000
i3.8xlarge	1.65백만	150,000
i3.16xlarge	3.3백만	300,000

* i3.large 및 i3.xlarge 인스턴스의 경우, 지정된 최대 성능을 얻을 수 있습니다.

인스턴스에 대한 SSD 기반 인스턴스 스토어 볼륨에 데이터가 있는 경우, 달성 가능한 쓰기 IOPS의 수는 감소합니다. 이는 SSD 컨트롤러가 가용 공간을 찾고 기존 데이터를 다시 쓰고 미사용 공간을 삭제하여 다시 쓸 수 있는 공간을 마련하기 위해 추가적인 작업을 해야 하기 때문입니다. 이러한 폐영역 회수 과정은 SSD에 대한 내부 쓰기 작업이 증폭되는 결과를 낳게 되며, 이런 결과는 사용자 쓰기 작업에 대한 SSD 쓰기 작업의 비로 표현됩니다. 이러한 성능 감소는 쓰기 작업이 4096바이트의 배수들 또는 4096바이트 경계에 정렬되지 않은 상태로 수행되는 경우에 더 심해질 수 있습니다. 정렬되지 않은 바이트를 소량으로 쓰기 작업하는 경우, SSD 컨트롤러는 쓰려는 부분의 주변 데이터를 읽고 그 결과도 새 위치에 저장해야 합니다. 이런 패턴으로 인해 쓰기 작업이 크게 증폭되고 지연 시간 증가와 I/O 성능의 급격한 감소를 초래합니다.

SSD 컨트롤러는 여러 전략을 사용해서 쓰기 작업 증폭의 영향을 감쇄할 수 있습니다. 그 중 하나의 전력은 SSD 인스턴스 스토리지에 예약 공간을 마련해서 SSD 컨트롤러가 쓰기 작업에 사용 가능한 공간을 보다 효율적으로 관리할 수 있게 하는 것입니다. 이를 오버-프로비저닝이라고 합니다. 인스턴스에 제공된 SSD 기반 인스턴스 스토어 볼륨은 오버-프로비저닝을 위한 예약 공간을 가지고 있지 않습니다. 쓰기 작업 증폭의 영향 감쇄를 위해 최소한 볼륨의 10%를 파티션 처리되지 않은 상태로 두어서 SSD 컨트롤러가 이를 오버-프로비저닝에 사용할 수 있도록 하는 것이 좋습니다. (`hdparm` 유ти리티를 사용하여 SSD 볼륨을 오버-프로비저닝 할 수 있습니다.) 그러면 사용할 수 있는 스토리지는 줄어들지만, 디스크를 전체 용량에 가깝게 사용하더라도 성능은 향상됩니다.

TRIM을 지원하는 인스턴스 스토어 볼륨의 경우, TRIM 명령을 사용하여 작성한 데이터가 더 이상 필요하지 않을 때 SSD 컨트롤러에 알릴 수 있습니다. 이를 통해 컨트롤러에 더 많은 여유 공간이 제공되어 쓰기 증폭이 줄어들고 성능이 향상될 수 있습니다. 자세한 내용은 [인스턴스 스토어 볼륨 TRIM 지원 \(p. 649\)](#) 섹션을 참조하십시오.

스토리지 인스턴스 기능

스토리지 최적화 인스턴스에 대한 기능은 다음과 같이 간략히 설명할 수 있습니다.

	VPC 전용	SSD 볼륨	배치 그룹	향상된 네트워킹
D2			예	Intel 82599 VF
I2		SATA	예	Intel 82599 VF
I3	예	NVMe	예	ENI

자세한 내용은 다음 자료를 참조하십시오.

- [VPC에서만 사용할 수 있는 인스턴스 유형 \(p. 470\)](#)
- [Amazon EBS 최적화 인스턴스 \(p. 614\)](#)
- [Amazon EC2 인스턴스 스토어 \(p. 642\)](#)
- [배치 그룹 \(p. 527\)](#)
- [Linux에서 향상된 네트워킹 \(p. 533\)](#)

개의 vCPU 지원

`d2.8xlarge` 인스턴스 유형은 36개의 vCPU를 지원하므로 vCPU가 32개로 제한되는 일부 Linux 운영 체제에서 시작 문제가 발생할 수 있습니다. 따라서 `d2.8xlarge` 인스턴스를 시작할 때 최신 AMI를 사용하실 것을 적극 권장합니다.

다음 Linux AMI는 36개의 vCPU로 `d2.8xlarge` 인스턴스 시작을 지원합니다.

- [Amazon Linux AMI 2017.03\(HVM\)](#)
- [Ubuntu Server 14.04 LTS\(HVM\)](#)
- [Red Hat Enterprise Linux 7.1\(HVM\)](#)

- SUSE Linux Enterprise Server 12(HVM)

애플리케이션에 대해 다른 AMI를 사용해야 하지만 d2.8xlarge 인스턴스가 완전히 시작되지 않은 경우(예: Client.InstanceInitiatedShutdown 상태 전환으로 인해 시작 중에 인스턴스 상태가 stopped로 변경된 경우) d2.8xlarge 인스턴스 유형을 사용할 수 있도록 다음 절차에 따라 32개보다 많은 vCPU를 지원하도록 AMI를 수정하십시오.

32개보다 많은 vCPU를 지원하도록 인스턴스를 업데이트하려면

1. AMI를 사용하여 D2 인스턴스를 시작하여 d2.8xlarge 이외의 D2 인스턴스 유형을 선택합니다.
2. 운영 체제 관련 지침에 따라 커널을 최신 버전으로 업데이트합니다. 예를 들어, RHEL 6의 경우 다음 명령을 사용합니다.

```
sudo yum update -y kernel
```

3. 인스턴스를 중지합니다.
4. (선택 사항) 나중에 필요한 추가 d2.8xlarge 인스턴스를 시작하는 데 사용할 수 있는 인스턴스에서 AMI를 생성합니다.
5. 중지된 인스턴스의 인스턴스 유형을 d2.8xlarge로 변경합니다(Actions, Instance Settings, Change Instance Type을 선택한 다음 지침을 따릅니다).
6. 인스턴스를 시작합니다. 인스턴스가 올바로 시작되면 완료된 것입니다. 그래도 인스턴스가 올바로 부팅되지 않으면 다음 단계로 진행하십시오.
7. (선택 사항) 그래도 인스턴스가 올바로 부팅되지 않으면 인스턴스의 커널이 32개를 초과하는 vCPU를 지원하지 않을 수도 있습니다. 하지만 vCPU 수를 제한하면 인스턴스를 부팅할 수 있습니다.
 - a. 중지된 인스턴스의 인스턴스 유형을 d2.8xlarge가 아닌 다른 D2 인스턴스 유형으로 변경합니다(Actions, Instance Settings, Change Instance Type을 선택한 다음 지침을 따릅니다).
 - b. 운영 체제 관련 지침에 따라 maxcpus=32 옵션을 부팅 커널 파라미터에 추가합니다. 예를 들어, RHEL 6의 경우 /boot/grub/menu.lst 파일을 편집하고 다음 옵션을 가장 최근 활성 kernel 항목에 추가합니다.

```
default=0
timeout=1
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux Server (2.6.32-504.3.3.el6.x86_64)
root (hd0,0)
kernel /boot/vmlinuz-2.6.32-504.3.3.el6.x86_64 maxcpus=32 console=ttyS0 ro
root=UUID=9996863e-b964-47d3-a33b-3920974fdbd9 rd_NO_LUKS KEYBOARDTYPE=pc
KEYTABLE=us LANG=en_US.UTF-8 xen_blkfront.sda_is_xvda=1 console=ttyS0,115200n8
console=tty0 rd_NO_MD SYSFONT=latarcyrheb-sun16 crashkernel=auto rd_NO_LVM
rd_NO_DM
initrd /boot/initramfs-2.6.32-504.3.3.el6.x86_64.img
```

- c. 인스턴스를 중지합니다.
- d. (선택 사항) 나중에 필요한 추가 d2.8xlarge 인스턴스를 시작하는 데 사용할 수 있는 인스턴스에서 AMI를 생성합니다.
- e. 중지된 인스턴스의 인스턴스 유형을 d2.8xlarge로 변경합니다(Actions, Instance Settings, Change Instance Type을 선택한 다음 지침을 따릅니다).
- f. 인스턴스를 시작합니다.

인스턴스 제한

- HVM AMI를 사용해서 스토리지 최적화 인스턴스를 실행해야 합니다. 자세한 내용은 [Linux AMI 가상화 유형 \(p. 66\)](#) 섹션을 참조하십시오.

- Amazon EBS 지원 AMI를 사용하여 I3 인스턴스를 실행해야 합니다.
- d2.8xlarge 인스턴스 유형에는 36개의 vCPU가 있으므로, vCPU가 32개로 제한되는 일부 Linux 운영 체제에서 시작 문제가 발생할 수 있습니다. 자세한 내용은 [개의 vCPU 지원 \(p. 160\)](#) 섹션을 참조하십시오.
- 한 리전에서 시작할 수 있는 총 인스턴스 수에는 제한이 있으며, 일부 인스턴스 유형에는 또 다른 제한이 있습니다. 자세한 내용은 [Amazon EC2에서 실행 가능한 인스턴스 수](#) 섹션을 참조하십시오.. 한도 증가를 요청하려면 [Amazon EC2 인스턴스 요청 양식](#)을 사용하십시오.

Linux 액셀러레이티드 컴퓨팅 인스턴스

고성능 병렬 처리 기능이 필요한 경우 NVIDIA GPU에 대한 액세스를 제공하는 액셀러레이티드 컴퓨팅 인스턴스를 활용할 수 있습니다. 액셀러레이티드 컴퓨팅 인스턴스로 CUDA(Compute Unified Device Architecture) 또는 OpenCL(Open Computing Language) 병렬 컴퓨팅 프레임워크를 활용하여 다양한 과학, 공학 및 렌더링 애플리케이션의 속도를 높일 수 있습니다. 게임 스트리밍, 3-D 애플리케이션 스트리밍 등의 그래픽 애플리케이션 및 기타 그래픽 워크로드에 활용할 수도 있습니다.

액셀러레이티드 컴퓨팅 인스턴스는 HVM 기반 인스턴스로 실행됩니다. 하드웨어 가상 머신(HVM) 가상화에는 AWS 플랫폼이 제공하는 하드웨어 보조 기술이 사용됩니다. HVM 가상화를 통해 게스트 VM은 마치 원래 하드웨어 플랫폼에 있는 것처럼 실행되며, 이를 통해 Amazon EC2는 액셀러레이티드 컴퓨팅 인스턴스 각각에 분산되어 있는 하나 이상의 GPU에 대해 전용 액세스를 제공할 수 있습니다.

여러 개의 액셀러레이티드 컴퓨팅 인스턴스를 하나의 배치 그룹으로 클러스터링할 수 있습니다. 배치 그룹은 단일 가용 영역에 속하는 인스턴스 간에 낮은 지연 시간과 고대역폭 연결을 제공합니다. 자세한 내용은 [배치 그룹 \(p. 527\)](#) 섹션을 참조하십시오.

목차

- [액셀러레이티드 컴퓨팅 인스턴스 패밀리 \(p. 162\)](#)
- [하드웨어 사양 \(p. 163\)](#)
- [액셀러레이티드 컴퓨팅 인스턴스 제한 \(p. 163\)](#)
- [액셀러레이티드 컴퓨팅 인스턴스용 AMI \(p. 163\)](#)
- [Amazon Linux에 NVIDIA 드라이버 설치 \(p. 163\)](#)
- [GPU 설정 최적화\(P2 인스턴스만 해당\) \(p. 165\)](#)

Windows 액셀러레이티드 컴퓨팅 인스턴스에 대한 자세한 내용은 [Windows 인스턴스용 Amazon EC2 사용 설명서](#)의 Windows 액셀러레이티드 컴퓨팅 인스턴스를 참조하십시오.

액셀러레이티드 컴퓨팅 인스턴스 패밀리

액셀러레이티드 컴퓨팅 인스턴스 패밀리는 하드웨어 액셀러레이터나 코프로세서를 사용함으로써 부동 소수점 계산 및 트래픽 처리 등과 같은 일부 기능을, CPU에서 실행하는 소프트웨어에서 수행하는 것보다 효율적으로 수행합니다. Amazon EC2에서는 다음과 같은 액셀러레이티드 컴퓨팅 인스턴스 패밀리를 실행할 수 있습니다.

P2 인스턴스

P2 인스턴스는 NVIDIA Tesla K80 GPU를 사용하며, CUDA 또는 OpenCL 프로그래밍 모델을 사용하는 일반 GPU 컴퓨팅에 맞게 설계되었습니다. P2 인스턴스는 고대역 네트워킹, 강력한 단일 정밀도 및 배정밀도 부동 소수점 기능, GPU당 12GiB의 메모리를 제공하므로, 딥 러닝, 그래프 데이터베이스, 고성능 데이터베이스, 전산 유체 역학(CFD), 계산 금융(Computational Finance), 내진 해석, 분자 모델링, 유전체학, 렌더링 및 기타 서버 GPU 컴퓨팅 워크로드에 이상적입니다.

- P2 인스턴스는 ENA(Elastic Network Adapter)를 통해 향상된 네트워킹을 지원합니다. 자세한 내용은 [VPC 의 Linux 인스턴스에서 ENA\(Elastic Network Adapter\)를 사용하여 향상된 네트워킹 활성화 \(p. 543\)](#) 섹션을 참조하십시오.

- P2 인스턴스는 기본적으로 EBS에 최적화되어 있습니다. 자세한 내용은 [Amazon EBS 최적화 인스턴스 \(p. 614\)](#) 섹션을 참조하십시오.
- P2 인스턴스는 NVIDIA GPUDirect 피어 투 피어 전송을 지원합니다. 자세한 내용은 [NVIDIA GPUDirect 섹션을 참조하십시오.](#)
- P2 인스턴스에서 최고의 성능을 달성하기 위해 수행할 수 있는 몇 가지 GPU 설정 최적화가 있습니다. 자세한 내용은 [GPU 설정 최적화\(P2 인스턴스만 해당\) \(p. 165\)](#) 섹션을 참조하십시오.
- p2.16xlarge 인스턴스 유형은 운영 체제에서 프로세서 C 상태 및 P 상태를 제어할 수 있는 기능을 제공합니다. 자세한 내용은 [EC2 인스턴스에 대한 프로세서 상태 제어 \(p. 306\)](#) 섹션을 참조하십시오.

G2 인스턴스

G2 인스턴스는 NVIDIA GRID K520 GPU를 사용하며 DirectX 또는 OpenGL을 사용하는 그래픽 애플리케이션을 위한 경제적이고도 높은 성능의 플랫폼을 제공합니다. 또한 NVIDIA GRID GPU는 NVIDIA의 빠른 캡처 기능을 지원하고 API 연산을 인코딩합니다. 애플리케이션의 예로는 비디오 제작 서비스, 3D 가상화, 스트리밍 그래픽 집약적 애플리케이션 및 기타 서버 측 그래픽 워크로드 등을 들 수 있습니다.

CG1 인스턴스

CG1 인스턴스는 NVIDIA Tesla M2050 GPU를 사용하며, CUDA 또는 OpenCL 프로그래밍 모델을 사용하는 일반 GPU 컴퓨팅에 맞게 설계되었습니다. CG1 인스턴스는 고대역 네트워킹, 배정밀도 부동 소수점 기능, ECC(Error-Correcting Code) 메모리를 제공하므로 고성능 컴퓨팅(HPC) 애플리케이션에 이상적입니다.

하드웨어 사양

Amazon EC2 인스턴스 유형별 하드웨어 사양에 대한 자세한 내용은 [Amazon EC2 인스턴스](#)를 참조하십시오.

액셀러레이티드 컴퓨팅 인스턴스 제한

액셀러레이티드 컴퓨팅 인스턴스는 다음과 같은 제한이 있습니다.

- HVM AMI를 사용해서 I 인스턴스를 실행해야 합니다.
- NVIDIA 드라이버를 설치해야 GPU에 액세스할 수 있습니다.
- 실행할 수 있는 인스턴스의 수에 제한이 있습니다. 자세한 내용은 [Amazon EC2에서 실행 가능한 인스턴스 수\(Amazon EC2 FAQ\)](#) 섹션을 참조하십시오. 이 제한을 높이도록 요청하려면 [Amazon EC2 인스턴스 제한 증가 요청 양식](#)을 사용하십시오.

액셀러레이티드 컴퓨팅 인스턴스용 AMI

쉽게 시작할 수 있도록 하기 위해 NVIDIA에서 액셀러레이티드 컴퓨팅 인스턴스를 위한 AMI를 제공합니다. 이러한 참조 AMI에는 NVIDIA GPU의 기능과 성능을 완벽하게 발휘하도록 하는 NVIDIA 드라이버가 포함되어 있습니다.

NVIDIA 드라이버가 포함된 AMI의 목록은 [AWS Marketplace\(NVIDIA GRID\)](#) 섹션을 참조하십시오.

모든 HVM AMI를 사용하여 액셀러레이티드 컴퓨팅 인스턴스를 시작할 수 있습니다.

Amazon Linux에 NVIDIA 드라이버 설치

액셀러레이티드 컴퓨팅 인스턴스에는 적합한 NVIDIA 드라이버가 있어야 합니다. 인스턴스에서 실행하려는 커널에 맞게 컴파일된 NVIDIA 드라이버를 설치해야 합니다.

Amazon은 AWS Marketplace의 각 공식 커널 업그레이드에 대한 NVIDIA 커널 드라이버의 업데이트된 호환 빌드를 AMI에 제공합니다. Amazon이 제공하는 것과 다른 버전의 NVIDIA 드라이버를 사용하거나 공식 Amazon 빌드가 아닌 커널을 사용하려는 경우 Amazon에서 제공한 NVIDIA 패키지를 시스템에서 제거하여 설치하려는 드라이버 버전과 충돌하지 않도록 해야 합니다.

이 명령을 사용하여 Amazon에서 제공한 NVIDIA 패키지를 제거합니다.

```
[ec2-user ~]$ sudo yum erase nvidia cuda
```

Amazon에서 제공한 CUDA 도구 키트 패키지는 NVIDIA 드라이버에 의존합니다. NVIDIA 패키지를 제거하면 CUDA 도구 키트도 삭제됩니다. NVIDIA 드라이버를 설치한 후 CUDA 도구 키트를 다시 설치해야 합니다.

NVIDIA 드라이버는 <http://www.nvidia.com/Download/Find.aspx>에서 다운로드할 수 있습니다. 인스턴스에 적합한 드라이버를 선택합니다.

P2 인스턴스

제품 유형	Tesla
제품 시리즈	K 시리즈
제품	K-80
운영 체제	Linux 64비트
권장/베타	권장/인증

G2 인스턴스

제품 유형	GRID
제품 시리즈	GRID 시리즈
제품	GRID K520
운영 체제	Linux 64비트
권장/베타	권장/인증

CG1 인스턴스

제품 유형	Tesla
제품 시리즈	M-Class
제품	M2050
운영 체제	Linux 64비트
권장/베타	권장/인증

드라이버 설치 및 구성에 대한 자세한 내용을 보려면 NVIDIA 웹 사이트에서 드라이버 다운로드 페이지의 [ADDITIONAL INFORMATION] 탭을 선택하고 README 링크를 선택하십시오.

NVIDIA 드라이버 직접 설치

Amazon Linux AMI용 드라이버를 설치하려면 다음을 수행합니다.

1. yum update 명령을 실행하여 인스턴스용 패키지의 최신 버전을 가져옵니다.

```
[ec2-user ~]$ sudo yum update -y
```

2. 인스턴스를 재부팅하여 최신 커널 버전을 로드합니다.

```
[ec2-user ~]$ sudo reboot
```

3. 재부팅이 끝난 후 인스턴스에 다시 연결합니다.
4. 현재 실행 중인 커널의 버전에 맞는 gcc 컴파일러와 kernel-devel 패키지가 설치되었는지 확인합니다.

```
[ec2-user ~]$ sudo yum install -y gcc kernel-devel-`uname -r`
```

5. 앞서 확인한 드라이버 패키지를 다운로드합니다. 예를 들어 다음 명령은 P2 인스턴스용 NVIDIA 드라이버의 352.99 버전을 다운로드합니다.

```
[ec2-user ~]$ wget http://us.download.nvidia.com/XFree86/Linux-x86_64/352.99/NVIDIA-Linux-x86_64-352.99.run
```

6. 자동 설치 스크립트를 실행하여 NVIDIA 드라이버를 설치합니다. 예:

```
[ec2-user ~]$ sudo /bin/bash ./NVIDIA-Linux-x86_64-352.99.run
```

7. 인스턴스를 재부팅합니다.

```
[ec2-user ~]$ sudo reboot
```

8. 드라이버가 작동하는지 확인합니다. 다음 명령의 응답에는 설치된 NVIDIA 드라이버 버전 및 GPU 관련 세부 정보가 나열됩니다.

Note

이 명령은 실행 시간이 몇 분 정도 걸릴 수 있습니다.

```
[ec2-user ~]$ nvidia-smi -q | head  
=====NVSMI LOG=====  
  
Timestamp : Thu Aug 25 04:59:03 2016  
Driver Version : 352.99  
  
Attached GPUs : 8  
GPU 0000:00:04.0 : Tesla K80  
Product Name : Tesla  
Product Brand : Tesla
```

9. (P2 인스턴스만 해당) P2 인스턴스를 사용할 경우 다음 섹션에 나와 있는 최적화 단계를 수행하여 GPU의 최고 성능을 활용하십시오.

GPU 설정 최적화(P2 인스턴스만 해당)

P2 인스턴스에서 최고의 성능을 달성하기 위해 수행할 수 있는 몇 가지 GPU 설정 최적화가 있습니다. 기본적으로 NVIDIA 드라이버는 GPU 클록 속도에 변화를 주는 자동 부스트 기능을 사용합니다. 자동 부스트 기능을 비활성화하고 GPU 클록 속도를 최대 주파수로 설정하면 P2 인스턴스의 성능을 최대로 유지할 수 있습니다. 다음 절차는 GPU 설정을 영구적으로 구성하고, 자동 부스트 기능을 비활성화하며, GPU 클록 속도를 최대 주파수로 설정하는 방법을 보여 줍니다.

P2 GPU 설정을 최적화하려면

1. GPU 설정을 영구적으로 구성합니다.

Note

이 명령은 실행 시간이 몇 분 정도 걸릴 수 있습니다.

```
[ec2-user ~]$ sudo nvidia-smi -pm 1
Enabled persistence mode for GPU 0000:00:0F.0.
Enabled persistence mode for GPU 0000:00:10.0.
Enabled persistence mode for GPU 0000:00:11.0.
Enabled persistence mode for GPU 0000:00:12.0.
Enabled persistence mode for GPU 0000:00:13.0.
Enabled persistence mode for GPU 0000:00:14.0.
Enabled persistence mode for GPU 0000:00:15.0.
Enabled persistence mode for GPU 0000:00:16.0.
Enabled persistence mode for GPU 0000:00:17.0.
Enabled persistence mode for GPU 0000:00:18.0.
Enabled persistence mode for GPU 0000:00:19.0.
Enabled persistence mode for GPU 0000:00:1A.0.
Enabled persistence mode for GPU 0000:00:1B.0.
Enabled persistence mode for GPU 0000:00:1C.0.
Enabled persistence mode for GPU 0000:00:1D.0.
Enabled persistence mode for GPU 0000:00:1E.0.
All done.
```

2. 인스턴스에 대해 모든 GPU의 자동 부스트 기능을 비활성화합니다.

```
[ec2-user ~]$ sudo nvidia-smi --auto-boost-default=0
All done.
```

3. 모든 GPU 클록 속도를 최대 주파수로 설정합니다.

```
[ec2-user ~]$ sudo nvidia-smi -ac 2505,875
Applications clocks set to "(MEM 2505, SM 875)" for GPU 0000:00:0F.0
Applications clocks set to "(MEM 2505, SM 875)" for GPU 0000:00:10.0
Applications clocks set to "(MEM 2505, SM 875)" for GPU 0000:00:11.0
Applications clocks set to "(MEM 2505, SM 875)" for GPU 0000:00:12.0
Applications clocks set to "(MEM 2505, SM 875)" for GPU 0000:00:13.0
Applications clocks set to "(MEM 2505, SM 875)" for GPU 0000:00:14.0
Applications clocks set to "(MEM 2505, SM 875)" for GPU 0000:00:15.0
Applications clocks set to "(MEM 2505, SM 875)" for GPU 0000:00:16.0
Applications clocks set to "(MEM 2505, SM 875)" for GPU 0000:00:17.0
Applications clocks set to "(MEM 2505, SM 875)" for GPU 0000:00:18.0
Applications clocks set to "(MEM 2505, SM 875)" for GPU 0000:00:19.0
Applications clocks set to "(MEM 2505, SM 875)" for GPU 0000:00:1A.0
Applications clocks set to "(MEM 2505, SM 875)" for GPU 0000:00:1B.0
Applications clocks set to "(MEM 2505, SM 875)" for GPU 0000:00:1C.0
Applications clocks set to "(MEM 2505, SM 875)" for GPU 0000:00:1D.0
Applications clocks set to "(MEM 2505, SM 875)" for GPU 0000:00:1E.0
All done.
```

T1 마이크로 인스턴스

T1 마이크로 인스턴스(t1.micro)는 소량의 CPU 리소스를 일정하게 제공하며 추가 주기를 사용할 수 있을 때 단기간에 CPU 용량을 확장할 수 있습니다. 처리 속도가 낮아 정기적으로 컴퓨팅 주기를 추가해야 하는 애플리케이션과 웹 사이트에 적합합니다.

Note

t1.micro는 이전 세대 인스턴스이며, 성능 프로필이 훨씬 더 뛰어난 t2.micro로 대체되었습니다. t1.micro 대신 t2.micro 인스턴스 유형을 사용하는 것이 좋습니다. 자세한 내용은 [T2 인스턴스 \(p. 149\)](#) 섹션을 참조하십시오.

t1.micro 인스턴스는 Amazon EBS 기반 인스턴스로만 제공됩니다.

이 문서에서는 `t1.micro` 인스턴스를 적용하는 방법을 알 수 있도록 인스턴스의 작동 방법을 설명합니다. 이 문서는 정확한 동작을 설명하기 위한 것이 아니라 인스턴스의 동작에 대한 가시성을 제공하여 인스턴스의 성능을 이해할 수 있도록 돋기 위한 것입니다.

함목

- [하드웨어 사양 \(p. 167\)](#)
- [최적의 상태로 T1 마이크로 인스턴스 적용 \(p. 167\)](#)
- [스파이크 동안 사용 가능한 CPU 리소스 \(p. 167\)](#)
- [인스턴스에서 할당된 리소스를 사용하는 경우 \(p. 168\)](#)
- [m1.small 인스턴스 유형과 비교 \(p. 168\)](#)
- [마이크로 인스턴스에 대한 AMI 최적화 \(p. 169\)](#)

하드웨어 사양

Amazon EC2 인스턴스 유형별 하드웨어 사양에 대한 자세한 내용은 [Amazon EC2 인스턴스](#)를 참조하십시오.

최적의 상태로 T1 마이크로 인스턴스 적용

`t1.micro` 인스턴스는 다음 그림에 표시된 것과 비슷한 CPU 사용 프로필을 가진 작업을 위해 급증하는 CPU 리소스를 제공합니다.

인스턴스는 두 가지 수준 즉, 정상 이하 배경 수준과 배경 수준보다 훨씬 더 높은 순간 스파이크 수준에서만 CPU를 사용하도록 설계되었습니다. 인스턴스를 최대 2개의 EC2 컴퓨팅 유닛(ECU)으로 작동할 수 있습니다. ECU 한 개당 제공하는 용량은 1.0-1.2 GHz 2007 Opteron 또는 2007 Xeon 프로세서와 동일합니다. 최대 수준과 배경 수준 사이의 비율이 커지도록 설계되었습니다. 애플리케이션에서 분당 수십 개의 요청을 지원하도록 `t1.micro` 인스턴스를 설계했습니다. 그러나 실제 성능은 애플리케이션에서 각 요청에 필요한 CPU 리소스의 양에 따라 상당히 다를 수 있습니다.

사용 중인 애플리케이션의 CPU 사용 프로필은 이전 섹션에 설명된 것과 다를 수 있습니다. 다음 그림에서는 `t1.micro` 인스턴스에 적합하지 않은 애플리케이션에 대한 프로필을 보여 줍니다. 각 요청에 대해 데이터를 고속으로 대량 처리하는 CPU 리소스가 애플리케이션에 필요하므로 `t1.micro` 인스턴스에서 처리할 수 없는 CPU 사용 정체가 발생합니다.

다음 그림에서는 `t1.micro` 인스턴스에 적합하지 않은 다른 프로필을 보여 줍니다. 여기에서 CPU 사용 스파이크는 짧지만 너무 자주 발생되므로 마이크로 인스턴스에서 처리할 수 없습니다.

다음 그림에서는 `t1.micro` 인스턴스에 적합하지 않은 다른 프로필을 보여 줍니다. 여기에서는 스파이크가 너무 자주 발생하지는 않지만 스파이크 간의 배경 수준이 너무 높아서 `t1.micro` 인스턴스에서 처리할 수 없습니다.

`t1.micro` 인스턴스에 적합하지 않은 각각의 이전 작업 사례에서는 다른 인스턴스 유형을 사용하는 것이 좋습니다. 인스턴스 유형에 대한 자세한 내용은 [인스턴스 유형 \(p. 146\)](#) 섹션을 참조하십시오.

스파이크 동안 사용 가능한 CPU 리소스

인스턴스는 컴퓨팅 리소스에 대한 수요 스파이크를 수용하도록 확장될 경우 호스트에서 사용되지 않은 리소스를 사용합니다. 사용 가능한 양은 스파이크가 발생할 때의 경합 정도에 따라 다릅니다. 호스트에 있는 다른 인스턴스의 급증 여부에 상관없이 인스턴스는 CPU 리소스가 고갈된 상태로 유지되지 않습니다.

인스턴스에서 할당된 리소스를 사용하는 경우

애플리케이션에서 지정된 기간 동안 특정 양의 CPU 리소스만 사용해야 합니다. 애플리케이션에서 인스턴스에 할당된 것보다 더 많은 CPU 리소스를 사용할 경우 낮은 CPU 수준에서 작동하도록 인스턴스를 일시적으로 제한합니다. 인스턴스에서 계속해서 할당된 리소스를 모두 사용할 경우 성능이 저하됩니다. 이 경우 CPU 수준을 제한하는 시간이 늘어나므로 다시 확장하기 위해 인스턴스가 할당되는 시간이 길어집니다.

`t1.micro` 인스턴스에 대한 CloudWatch 모니터링을 활성화할 경우 AWS Management Console에서 "Avg CPU Utilization" 그래프를 사용하여 인스턴스에서 모든 할당된 CPU 리소스를 정기적으로 사용하는지 여부를 확인할 수 있습니다. 각 지정된 기간 중에 도달한 최대값을 조사하는 것이 좋습니다. 최대값이 100%인 경우 Auto Scaling을 사용하여 확장하거나(주가 `t1.micro` 인스턴스 및 로드 밸런서 사용) 더 큰 인스턴스 유형으로 전환하는 것이 좋습니다. 자세한 내용은 [Auto Scaling 사용 설명서](#) 섹션을 참조하십시오.

다음 그림에서는 이전 섹션에서 최적화되지 않은 세 프로필과 인스턴스에서 할당된 리소스를 소진하여 CPU 수준을 제한해야 하는 경우를 보여 줍니다. 인스턴스에서 할당된 리소스를 소진한 경우 인스턴스를 낮은 배경 수준으로 제한합니다.

다음 그림에서는 데이터를 고속으로 대량 처리하는 CPU 사용 정체가 긴 상황을 보여 줍니다. CPU가 최대 허용 수준에 도달하여 해당 기간에 대해 인스턴스에 할당된 리소스가 소진될 때까지 해당 상태로 유지됩니다. 그러면 인스턴스를 낮은 배경 수준에서 작동하도록 제한하고, 인스턴스는 해당 수준 이상으로 다시 확장되도록 허용될 때까지 제한된 수준으로 작동합니다. 인스턴스는 할당된 리소스가 소진되어 다시 제한될 때까지(그래프에 표시되지 않음) 이 상태로 유지됩니다.

다음 그림에서는 요청이 너무 자주 발생하는 상황을 보여 줍니다. 인스턴스에서 단 몇 번의 요청만에 할당된 리소스를 모두 사용하므로 인스턴스가 제한됩니다. 제한을 높이면 인스턴스에서 요청을 처리하기 위해 CPU 사용을 최대화하고 그러면 인스턴스가 다시 제한됩니다.

다음 그림에서는 배경 수준이 너무 높은 상황을 보여 줍니다. 인스턴스를 제한 대상이 되는 최대 CPU 수준으로 작동할 필요가 없습니다. 인스턴스가 정상 배경 수준 이상에서 작동하여 지정된 기간에 대해 할당된 리소스가 모두 소진될 경우 인스턴스를 제한합니다. 이전의 경우와 마찬가지로 이 경우에도 인스턴스에서 작업을 계속할 수 없으므로 인스턴스를 다시 제한합니다.

m1.small 인스턴스 유형과 비교

`t1.micro` 인스턴스는 항상 다른 수준의 CPU 리소스를 제공합니다(최대 2 ECU). 이에 비해 `m1.small` 인스턴스 유형은 항상 1 ECU를 제공합니다. 다음 그림에서는 차이점을 보여 줍니다.

다음 그림에서는 이전 섹션에서 설명한 다양한 시나리오를 기준으로 `t1.micro` 인스턴스의 CPU 사용을 `m1.small` 인스턴스와 비교합니다.

첫 번째 그림은 `t1.micro` 인스턴스에 가장 적합한 시나리오(왼쪽 그래프)와 `m1.small` 인스턴스의 경우(오른쪽 그래프)를 보여 줍니다. 이 경우에는 `t1.micro` 인스턴스를 제한할 필요가 없습니다. CPU 수요의 각 스파이크에 대한 `m1.small` 인스턴스의 처리 시간이 `t1.micro` 인스턴스에 비해 더 길입니다.

다음 그림에서는 `t1.micro` 인스턴스에서 할당된 리소스를 모두 사용한, 데이터를 고속으로 대량 처리하는 요청이 있는 시나리오와 `m1.small` 인스턴스의 경우를 보여 줍니다.

다음 그림에서는 `t1.micro` 인스턴스에서 할당된 리소스를 모두 사용하는 잣은 요청과 `m1.small` 인스턴스에서 해당 요청이 어떻게 나타나는지 보여 줍니다.

다음 그림에서는 배경 수준에서 `t1.micro` 인스턴스에 할당된 리소스를 모두 사용하는 상황과 `m1.small` 인스턴스에서 해당 배경 수준이 어떻게 나타나는지 보여 줍니다.

マイクロ 인스턴스에 대한 AMI 최적화

`t1.micro` 인스턴스 유형에 대해 AMI를 최적화할 경우 다음 모범 사례를 따르는 것이 좋습니다.

- 600MB의 RAM에서 실행하도록 AMI 설계
- CPU 시간을 사용하는 반복 프로세스 수 제한(예: 크론 작업, 데몬)

스왑 공간 및 가상 메모리를 사용하여 성능을 최적화할 수 있습니다(예: 루트 파일 시스템과 별도의 파티션에 스왑 공간 설정).

인스턴스 크기 조정

요구 사항이 변함에 따라 인스턴스가 과도하게(인스턴스 유형 크기가 너무 작은 경우) 또는 과소하게(인스턴스 유형 크기가 너무 큰 경우) 활용되고 있는 경우가 생길 수 있습니다. 이 경우는 인스턴스의 크기를 변경할 수 있습니다. 예를 들어 `t2.micro` 인스턴스가 워크로드에 비해 너무 작은 경우는 이를 `m3.medium` 인스턴스로 변경할 수 있습니다.

인스턴스의 루트 디바이스가 EBS 볼륨인 경우, 인스턴스 유형을 변경하여 간단히 인스턴스의 크기를 변경할 수 있습니다. 이를 크기 조정이라고 합니다. 인스턴스의 루트 디바이스가 인스턴스 스토어 볼륨인 경우, 원하는 인스턴스 유형의 새 인스턴스로 애플리케이션을 마이그레이션해야 합니다. 루트 디바이스 볼륨에 대한 자세한 내용은 [루트 디바이스 스토리지 \(p. 64\)](#) 섹션을 참조하십시오.

인스턴스의 크기를 조정할 경우 인스턴스의 구성과 호환되는 인스턴스 유형을 선택해야 합니다. 원하는 인스턴스 유형이 해당 인스턴스 구성과 호환되지 않을 경우, 원하는 인스턴스 유형의 새 인스턴스로 애플리케이션을 마이그레이션해야 합니다.

Important

인스턴스의 크기를 조정할 때 일반적으로 크기를 조정한 인스턴스는 원본 인스턴스를 시작할 때 지정한 것과 동일한 수의 인스턴스 스토어 볼륨을 갖습니다. 인스턴스 스토어 볼륨을 추가하려면 원하는 인스턴스 유형과 인스턴스 스토어 볼륨을 갖는 완전히 새로운 인스턴스로 애플리케이션을 마이그레이션해야 합니다. 이 규칙의 예외는 기본적으로 더 많은 수의 볼륨을 포함하는 스토리지 집약적 인스턴스 유형의 크기를 조정하는 경우입니다. 인스턴스 스토어 볼륨에 대한 자세한 내용은 [Amazon EC2 인스턴스 스토어 \(p. 642\)](#) 섹션을 참조하십시오.

목차

- [인스턴스 크기 조정을 위한 호환성 \(p. 169\)](#)
- [Amazon EBS 지원 인스턴스 크기 조정 \(p. 170\)](#)
- [인스턴스 스토어 지원 인스턴스 마이그레이션 \(p. 171\)](#)
- [새 인스턴스 구성으로 마이그레이션 \(p. 172\)](#)

인스턴스 크기 조정을 위한 호환성

인스턴스의 현재 인스턴스 유형과 새 인스턴스 유형이 호환될 경우에만 다음과 같이 인스턴스 크기를 조정할 수 있습니다.

- 가상화 유형. Linux AMI는 PV(반가상화) 또는 HVM(하드웨어 가상 머신)의 두 가지 유형의 가상화를 사용합니다. PV AMI에서 시작한 인스턴스를 HVM 전용의 인스턴스 유형으로 크기 조정할 수 없습니다. 자세한 내용은 [Linux AMI 가상화 유형 \(p. 66\)](#) 섹션을 참조하십시오. 인스턴스의 가상화 유형을 확인하려면 Amazon EC2 콘솔의 Instances 화면에서 세부 정보 창의 Virtualization 필드를 확인하십시오.
- 네트워크. 일부 인스턴스 유형은 EC2-Classic에서 지원되지 않으며 VPC에서 시작해야 합니다. 따라서 기반이 아닌 VPC가 아닌 한 EC2-Classic의 인스턴스를, VPC에서만 사용할 수 있는 인스턴스 유형으로 크

기 조정할 수 없습니다. 자세한 내용은 [VPC에서만 사용할 수 있는 인스턴스 유형 \(p. 470\)](#) 섹션을 참조하십시오. 인스턴스가 VPC에 있는지 확인하려면 Amazon EC2 콘솔의 Instances 화면에서 세부 정보 창의 VPC ID 값을 확인하십시오.

- 플랫폼. 모든 Amazon EC2 인스턴스 유형은 64비트 AMI를 지원하지만, 다음 인스턴스 유형만이 32비트 AMI를 지원합니다: t2.nano, t2.micro, t2.small, t2.medium, c3.large, t1.micro, m1.small, m1.medium, c1.medium. 32비트 인스턴스의 크기를 조정하는 경우는 상기 인스턴스 유형만 사용 가능합니다. 인스턴스 플랫폼을 확인하려면 Amazon EC2 콘솔에서 Instances 화면으로 이동한 후 Show/Hide Columns, Architecture를 선택합니다.

예를 들어 T2 인스턴스는 EC2-Classic에서 지원되지 않으며 HVM 전용입니다. 따라서 T1 인스턴스는 HVM을 지원하지 않고 PV AMI에서 시작해야 하기 때문에 T1 인스턴스를 T2 인스턴스로 크기 조정할 수 없습니다. T2 인스턴스를 더 큰 인스턴스 유형으로 크기 조정하려는 경우, 기존 세대 인스턴스 유형은 모두 HVM AMI를 지원하므로 M3와 같은 기존 세대 인스턴스 유형을 선택할 수 있습니다. 자세한 내용은 [사용 가능한 인스턴스 유형 \(p. 146\)](#) 섹션을 참조하십시오.

Amazon EBS 지원 인스턴스 크기 조정

인스턴스 유형을 변경하기 전에는 Amazon EBS 지원 인스턴스를 종단해야 합니다. 인스턴스를 중지했다가 시작할 때 다음 사항을 인식하십시오.

- 인스턴스를 새 하드웨어로 이동하지만, 인스턴스 ID는 변경되지 않습니다.
- 인스턴스가 VPC에서 실행 중이고 퍼블릭 IPv4 주소를 가지고 있으면 주소를 해제하고 새 퍼블릭 IPv4 주소를 제공합니다. 인스턴스는 프라이빗 IPv4 주소와 모든 탄력적 IP 주소(EIP), IPv6 주소를 유지합니다.
- 인스턴스가 EC2-Classic에서 실행 중인 경우, AWS는 거기에 새로운 퍼블릭 및 프라이빗 IP 주소를 부여하고 해당 인스턴스와 연결된 모든 탄력적 IP 주소를 분리합니다. 따라서 사용자가 인스턴스에 호스팅하는 애플리케이션을 계속 중단 없이 사용할 수 있도록 보장하기 위해, 인스턴스를 재시작 후 탄력적 IP 주소를 다시 연결할 필요가 있습니다.
- 인스턴스가 Auto Scaling 그룹에 있는 경우, Auto Scaling 서비스는 중단된 인스턴스를 비정상으로 간주해 이를 종료하고 대체 인스턴스를 시작합니다. 이를 방지하기 위해서는 해당 인스턴스의 크기를 조정하는 동안 그 그룹에 대한 Auto Scaling 과정을 일시 중지할 수 있습니다. 자세한 내용은 Auto Scaling 사용 설명서의 [Suspend and Resume Auto Scaling Processes](#)를 참조하십시오.
- 가동 중지는 인스턴스가 종단되었을 때 계획해야 합니다. 인스턴스 종단 및 크기 조정은 몇 분이 걸릴 수 있으며, 인스턴스를 다시 시작하는 시간은 애플리케이션의 시작 스크립트에 따라 달라질 수 있습니다.

자세한 내용은 [인스턴스 중지 및 시작 \(p. 285\)](#) 섹션을 참조하십시오.

다음 절차를 사용해서 AWS Management Console을 통해 Amazon EBS 지원 인스턴스의 크기를 조정합니다.

Amazon EBS 지원 인스턴스의 크기 조정

1. Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Instances]를 선택하고 인스턴스를 선택합니다.
3. [EC2-Classic] 원본 인스턴스에 연결된 탄력적 IP 주소가 있는 경우, 세부 정보 창에 표시된 탄력적 IP 주소 및 인스턴스 ID를 기록합니다.
4. [Actions]를 선택하고 [Instance State]를 선택한 후 [Stop]을 선택합니다.
5. 확인 대화 상자가 나타나면 Yes, Stop을 선택합니다. 인스턴스가 중지하는 데 몇 분 정도 걸릴 수 있습니다.

[EC2-Classic] 인스턴스가 stopped 상태가 되면, 세부 정보 창의 [Elastic IP], [Public DNS (IPv4)], [Private DNS] 및 [Private IPs] 필드가 공백으로 표시됩니다. 이는 기존 값이 더 이상 인스턴스와 연결되어 있지 않음을 나타냅니다.

6. 인스턴스를 선택된 상태에서 [Actions]를 선택하고 [Instance Settings]를 선택한 후 [Change Shutdown Behavior]를 선택합니다. 인스턴스가 stopped 상태가 아닐 경우에는 이 작업을 수행할 수 없습니다.

7. [Change Instance Type] 대화 상자에서 다음과 같이 합니다.
 - a. [Instance Type]에서 원하는 인스턴스 유형을 선택합니다. 원하는 인스턴스 유형이 목록에 없으면 해당 인스턴스의 구성과 호환되지 않는 것입니다. 예를 들어 가상화 유형 때문일 수 있습니다.
 - b. (선택 사항) 선택한 인스턴스 유형이 EBS 최적화를 지원할 경우 [EBS-optimized]를 선택하여 EBS 최적화를 활성화하거나, [EBS-optimized]의 선택을 취소하여 EBS 최적화를 비활성화합니다. 선택한 인스턴스 유형이 기본적으로 EBS 최적화되었을 경우 [EBS-optimized]가 선택되고 이를 선택 취소할 수 없습니다.
 - c. [Apply]를 선택하여 새로운 설정을 승인합니다.
8. 중지된 인스턴스를 다시 시작하려면 인스턴스를 선택하고, [Actions]를 선택한 후 [Instance State]를 선택하고, [Start]를 선택합니다.
9. 확인 대화 상자가 나타나면 [Yes, Start]를 선택합니다. 인스턴스가 `running` 상태가 되는 데 몇 분 정도 걸릴 수 있습니다.
10. [EC2-Classic] 인스턴스가 `running` 상태일 때, 세부 정보 창의 [Public DNS (IPv4)], [Private DNS], [Private IPs] 필드는 인스턴스에 할당된 새 값을 포함하고 있습니다. 인스턴스에 탄력적 IP 주소가 연결되어 있을 경우 다음과 같이 다시 연결해야 합니다.
 - a. 탐색 창에서 [Elastic IPs]를 선택합니다.
 - b. 인스턴스를 중지하기 전에 기록해 둔 탄력적 IP 주소를 선택합니다.
 - c. Actions를 선택한 후 Associate address를 선택합니다.
 - d. [Instance]에서 인스턴스를 중지하기 전에 기록해 둔 인스턴스 ID를 선택한 후 [Associate]를 선택합니다.

인스턴스 스토어 지원 인스턴스 마이그레이션

한 인스턴스 스토어 지원 인스턴스에서, 인스턴스 유형이 다른 인스턴스 스토어 지원 인스턴스로 애플리케이션을 이동할 경우 인스턴스에서 이미지를 작성한 후 이 이미지로부터 해당 인스턴스 유형의 새 인스턴스를 시작하여 애플리케이션을 마이그레이션해야 합니다. 따라서 사용자가 인스턴스에 호스팅하는 애플리케이션을 계속 중단 없이 사용할 수 있도록 보장하기 위해, 원래 인스턴스와 연결된 탄력적 IP 주소를 기록하고 이를 새 인스턴스와 연결해야 합니다. 그런 다음 원래 인스턴스를 종료하면 됩니다.

인스턴스 스토어 지원 인스턴스를 마이그레이션하려면

1. [EC2-Classic] 마이그레이션할 인스턴스에 연결된 탄력적 IP 주소가 있을 경우 나중에 새 인스턴스와 탄력적 IP 주소를 연결할 수 있도록 그 주소를 기록합니다.
2. 영구 스토리지를 유지해야 할 인스턴스 스토어 볼륨에 데이터를 백업합니다. 유지해야 하는 EBS 볼륨에 데이터를 마이그레이션하려면 볼륨의 스냅샷을 생성하거나([Amazon EBS 스냅샷 생성 \(p. 608\)](#) 참조), 나중에 새 인스턴스에 연결할 수 있도록 인스턴스에서 볼륨을 분리합니다([인스턴스에서 Amazon EBS 볼륨 분리 \(p. 588\)](#) 참조).
3. [인스턴스 스토어 기반 Linux AMI 생성 \(p. 84\)](#)의 사전 조건을 충족하고 해당 절차를 수행해서 인스턴스 스토어 지원 인스턴스에서 AMI를 생성합니다. 인스턴스에서 AMI를 생성했으면 이 절차로 다시 돌아옵니다.
4. 탐색 창에서 Amazon EC2 콘솔을 열고 [AMIs]를 선택합니다. 필터 목록에서 [Owned by me]를 선택하고 이전 단계에서 생성한 이미지를 선택합니다. 여기서 AMI Name은 이미지를 등록할 때 지정한 이름, Source는 사용자의 Amazon S3 버킷입니다.

Note

전 단계에서 생성한 AMI가 표시되지 않을 경우 AMI를 생성한 리전을 선택했는지 확인합니다.

5. [Launch]를 선택합니다. 인스턴스에 대한 옵션을 지정할 경우 필요한 새 인스턴스 유형을 선택하도록 합니다. 원하는 인스턴스 유형을 선택할 수 없으면 생성한 AMI의 구성과 호환되지 않는 것입니다. 예를 들어 가상화 유형 때문일 수 있습니다. 원래 인스턴스에서 분리한 EBS 볼륨을 지정할 수도 있습니다.

인스턴스가 `running` 상태가 되기까지 몇 분 정도 걸릴 수 있습니다.

6. [EC2-Classic] 시작한 인스턴스에 연결된 탄력적 IP 주소가 있을 경우, 다음과 같이 이 주소를 새 인스턴스에 연결해야 합니다.
 - a. 탐색 창에서 [Elastic IPs]를 선택합니다.
 - b. 이 절차를 시작할 때 기록해 둔 탄력적 IP 주소를 선택합니다.
 - c. [Actions]를 선택한 후 [Associate Address]를 선택합니다.
 - d. [Instance]에서 새 인스턴스를 선택한 후 [Associate]를 선택합니다.
7. (선택 사항) 시작한 인스턴스가 더 이상 필요하지 않은 경우 이를 종료할 수 있습니다. 인스턴스를 선택하고 새 인스턴스가 아닌 원래 인스턴스를 종료하고 있는지 확인합니다. 예를 들어 이름이나 시작 시간을 확인합니다. [Actions]를 선택하고 [Instance State]를 선택한 후 [Terminate]를 선택합니다.

새 인스턴스 구성으로 마이그레이션

인스턴스의 현재 구성이 새 인스턴스 유형과 호환되지 않을 경우, 인스턴스를 해당 인스턴스 유형으로 크기 조정할 수 없습니다. 대신 새 인스턴스 유형과 호환되는 구성을 가진 새 인스턴스로 애플리케이션을 마이그레이션할 수 있습니다.

PV AMI에서 시작한 인스턴스를 HVM 전용 인스턴스 유형으로 이동하려는 경우 일반적인 절차는 다음과 같습니다.

1. 영구 스토리지를 유지해야 할 인스턴스 스토어 볼륨에 데이터를 백업합니다. 유지해야 하는 EBS 볼륨에 데이터를 마이그레이션하려면 볼륨의 스냅샷을 생성하거나([Amazon EBS 스냅샷 생성 \(p. 608\)](#) 참조), 나중에 새 인스턴스에 연결할 수 있도록 인스턴스에서 볼륨을 분리합니다([인스턴스에서 Amazon EBS 볼륨 분리 \(p. 588\)](#) 참조).
2. 다음을 선택하여 새 인스턴스를 시작합니다.
 - HVM AMI
 - HVM 전용 인스턴스 유형
 - [EC2-VPC] 탄력적 IP 주소를 사용할 경우 원래 인스턴스가 현재 실행 중인 VPC를 선택합니다.
 - 원래 인스턴스에서 분리하여 새 인스턴스에 연결하려는 EBS 볼륨 또는 생성한 스냅샷에 기반한 새로운 EBS 볼륨
 - 새 인스턴스로 동일한 트래픽을 허용하려는 경우 원래 인스턴스와 연결된 보안 그룹을 선택합니다.
3. 애플리케이션과 기타 필요한 소프트웨어를 인스턴스에 설치합니다.
4. 원래 인스턴스의 인스턴스 스토어 볼륨에서 백업한 데이터를 복원합니다.
5. 탄력적 IP 주소를 사용할 경우 다음과 같이 이 주소를 새로 시작한 인스턴스에 지정합니다.
 - a. 탐색 창에서 [Elastic IPs]를 선택합니다.
 - b. 원래 인스턴스와 연결된 탄력적 IP 주소를 선택하고 Actions를 선택한 후 Disassociate address를 선택합니다. 확인 메시지가 나타나면 Disassociate address를 선택합니다.
 - c. 탄력적 IP 주소를 선택한 상태에서 Actions를 선택한 후 Associate address를 선택합니다.
 - d. [Instance]에서 새 인스턴스를 선택한 후 [Associate]를 선택합니다.
6. (선택 사항) 원래 인스턴스가 더 이상 필요하지 않을 경우 이를 종료할 수 있습니다. 인스턴스를 선택하고 새 인스턴스가 아닌 원래 인스턴스를 종료하고 있는지 확인합니다. 예를 들어 이름이나 시작 시간을 확인합니다. [Actions]를 선택하고 [Instance State]를 선택한 후 [Terminate]를 선택합니다.

EC2-Classic의 인스턴스에서 VPC의 인스턴스로 애플리케이션을 마이그레이션하는 내용은 [Linux 내 EC2-Classic 인스턴스에서 VPC 내 Linux 인스턴스로 마이그레이션 \(p. 481\)](#)을 참조하십시오.

인스턴스 구입 옵션

Amazon EC2는 사용자가 요구 사항에 따라 비용을 최적화할 수 있도록 다음과 같은 구입 옵션을 제공합니다.

- 온디맨드 인스턴스 - 시작하는 인스턴스 비용을 시간 단위로 지불합니다.
- [Reserved Instances] - 1년부터 3년까지의 기간 동안 항상 사용할 수 있는 인스턴스를 크게 할인된 가격으로 구입합니다.
- 정기 인스턴스 - 1개월부터 3년까지의 기간 동안 항상 사용할 수 있는 인스턴스를 크게 할인된 가격으로 구입합니다.
- 스팟 인스턴스 - 사용할 수 있는 기간 동안만 실행할 수 있으며 입찰 가격이 스팟 가격보다 높은 미사용 인스턴스를 크게 할인된 가격으로 입찰합니다.
- 전용 호스트 - 인스턴스를 실행을 전담하는 실제 호스트 비용을 지불하며, 기존의 소켓, 코어 또는 VM 소프트웨어별 라이선스를 가져와 비용을 절감합니다.
- [Dedicated instances] - 단일 테넌트 하드웨어에서 실행되는 인스턴스 비용을 시간 단위로 지불합니다.

용량 예약이 필요할 경우 예약 인스턴스 또는 정기 인스턴스를 고려하십시오. 스팟 인스턴스는 애플리케이션이 실행되는 시간을 유연하게 조정할 수 있고 애플리케이션을 중단할 수 있는 경우에 선택할 수 있는 비용 효과적인 방법입니다. 전용 호스트는 규정 준수 요건을 충족하는데 도움이 되고 기존의 서버별 소프트웨어 라이선스를 사용하여 비용을 절감할 수 있습니다. 자세한 내용은 [Amazon EC2 인스턴스 구입 옵션](#)을 참조하십시오.

목차

- [인스턴스 수명 주기 결정 \(p. 173\)](#)
- [예약 인스턴스 \(p. 174\)](#)
- [정기 예약 인스턴스 \(p. 200\)](#)
- [스팟 인스턴스 \(p. 203\)](#)
- [전용 호스트 \(p. 246\)](#)
- [전용 인스턴스 \(p. 257\)](#)

인스턴스 수명 주기 결정

인스턴스의 수명 주기는 인스턴스가 시작될 때부터 종료될 때까지입니다. 선택한 구매 옵션이 인스턴스의 수명 주기에 영향을 미칩니다. 예를 들어 온디맨드 인스턴스는 사용자가 그 인스턴스를 시작하면 실행되고 종료시키면 끝납니다. 스팟 인스턴스는 가용 용량이 있고 입찰 가격이 스팟 가격보다 더 높은 조건 하에서만 실행됩니다. 지정 기간 동안 정기 인스턴스를 시작할 수 있습니다. 예를 들어 Amazon EC2는 인스턴스를 시작하고 나서 기간 종료 3분 전에 종료됩니다.

다음 절차를 사용하여 인스턴스의 수명 주기를 결정합니다.

콘솔을 사용하여 인스턴스 수명 주기를 결정하려면

- <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
- 탐색 창에서 Instances를 선택합니다.
- 인스턴스를 선택합니다.
- [Description] 탭에서 [Tenancy]를 찾습니다. 값이 host인 경우 그 인스턴스는 전용 호스트에서 실행되고 있습니다. 값이 dedicated인 경우 그 인스턴스는 전용 인스턴스입니다.
- [Description] 탭에서 [Lifecycle]을 찾습니다. 값이 spot인 경우 그 인스턴스는 스팟 인스턴스입니다. 값이 scheduled인 경우 그 인스턴스는 정기 인스턴스입니다. 값이 normal인 경우 그 인스턴스는 온디맨드 인스턴스이거나 예약 인스턴스입니다.

6. (선택 사항) 예약 인스턴스를 구매했는데 그것이 적용되고 있는지 확인하고 싶은 경우, Amazon EC2에 대한 사용 보고서를 참고할 수 있습니다. 자세한 내용은 [예약 인스턴스 사용률 보고서 \(p. 693\)](#) 섹션을 참조하십시오.

AWS CLI를 사용하여 인스턴스 수명 주기를 결정하려면

아래와 같이 `describe-instances` 명령을 사용합니다.

```
aws ec2 describe-instances --instance-ids i-1234567890abcdef0
```

인스턴스가 전용 호스트에서 실행되고 있는 경우, 출력에는 다음 정보가 포함됩니다.

```
"Tenancy": "host"
```

인스턴스가 전용 인스턴스인 경우, 출력에는 다음 정보가 포함됩니다.

```
"Tenancy": "dedicated"
```

인스턴스가 스팟 인스턴스인 경우, 출력에는 다음 정보가 포함됩니다.

```
"InstanceLifecycle": "spot"
```

인스턴스가 정기 인스턴스인 경우, 출력에는 다음 정보가 포함됩니다.

```
"InstanceLifecycle": "scheduled"
```

그 외 경우에는 출력에 다음 정보가 포함됩니다.

```
"InstanceLifecycle": "normal"
```

예약 인스턴스

예약 인스턴스는 온디맨드 인스턴스 요금과 비교하여 대폭 할인된 요금을 제공합니다. 예약 인스턴스는 물리적 인스턴스가 아니고 오히려 계정에서 온디맨드 인스턴스를 사용할 때 적용되는 결제 할인에 가깝습니다. 이러한 온디맨드 인스턴스가 결제 할인의 혜택을 받으려면 일정한 속성에 부합되어야 합니다.

특정 가용 영역에서 예약 인스턴스를 구매할 경우 용량을 예약할 수 있습니다. 이후 특정 리전에서 예약 인스턴스를 구매하면서(리전 단위의 예약 인스턴스) 이 용량 예약을 포기할 수도 있습니다. 이러한 리전 단위의 예약 인스턴스로 가용 영역과 인스턴스의 크기를 유연하게 바꿀 수 있습니다. 이러한 가용 영역의 유연성은 단일 리전에 속한 모든 가용 영역의 인스턴스 사용량에 대해 예약 인스턴스의 할인 혜택을 제공합니다. 인스턴스 크기 유연성은 해당 인스턴스 패밀리 내에서 크기에 상관없이 인스턴스 사용량에 대해 예약 인스턴스의 할인 혜택을 제공합니다. 자세한 내용은 [예약 인스턴스 적용 \(p. 177\)](#) 섹션을 참조하십시오.

예약 인스턴스를 구입할 경우 원하는 결제 옵션, 기간 및 제공 클래스를 선택하십시오. 일반적으로 선결제 금액이 높은 예약 인스턴스를 선택할수록 요금 절약 혜택이 커집니다. 결제 옵션에는 선결제 없음, 부분 선결제, 전체 선결제의 세 가지가 있으며, 기간은 1년 또는 3년 중에서 선택할 수 있습니다. 또한 제공 클래스는 전환형과 표준형, 두 가지입니다.

- 선결제 및 부분 선결제 예약 인스턴스는 사용 여부와 상관없이 시간제로 사용 요금이 청구되지 않습니다. 모든 선결제 예약 인스턴스는 시간당 추가 요금이 없습니다.
- 전환형 예약 인스턴스는 해당 기간 동안 인스턴스 유형을 비롯한 새로운 속성의 전환형 예약 인스턴스와 교환할 수 있습니다. 표준 예약 인스턴스는 기간 동안 수정할 수 있지만 인스턴스 유형은 기간 동안 고정됩니다.

타사 판매업체에서 제공하는 예약 인스턴스는 기간이 짧고 요금 할인 폭도 적습니다. 자세한 내용은 [예약 인스턴스 마켓플레이스 \(p. 177\)](#) 섹션을 참조하십시오.

예약 인스턴스는 자동으로 갱신되지 않으므로 만료될 경우 중단 없이 EC2 인스턴스를 계속 사용할 수 있지 만 온디맨드 가격이 부과됩니다. 예약 인스턴스를 새로 구매하더라도 만료된 예약 인스턴스와 동일한 파라미터를 사용할 수 있으며, 혹은 다른 파라미터로 예약 인스턴스를 구매할 수도 있습니다.

Auto Scaling 또는 다른 AWS 서비스를 사용하여 예약 인스턴스의 혜택이 적용되는 온디맨드 인스턴스를 시작할 수 있습니다. 온디맨드 인스턴스 시작에 대한 자세한 내용은 [인스턴스 시작](#) 단원을 참조하십시오. Auto Scaling을 사용하여 인스턴스를 시작하는 방법에 대한 자세한 내용은 [Auto Scaling 사용 설명서](#)를 참조하십시오.

제품 요금 정보에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS 서비스 요금 개요](#)
- [Amazon EC2 온디맨드 인스턴스 요금](#)
- [Amazon EC2 예약 인스턴스 요금](#)

예약 인스턴스 요금 티어에 대한 자세한 내용은 [예약 인스턴스 할인 요금 티어 \(p. 181\)](#) 섹션을 참조하십시오.

항목

- [예약 인스턴스의 유형 \(p. 175\)](#)
- [예약 인스턴스의 작동 방식 \(p. 175\)](#)
- [청구 혜택과 결제 옵션 \(p. 177\)](#)
- [예약 인스턴스 구매 \(p. 183\)](#)
- [예약 인스턴스 마켓플레이스 판매 \(p. 187\)](#)
- [표준 예약 인스턴스 변경 \(p. 193\)](#)
- [전환형 예약 인스턴스 교환 \(p. 198\)](#)
- [변경 요청 문제 해결 \(p. 199\)](#)

예약 인스턴스의 유형

다음과 같은 두 가지 유형의 예약 인스턴스가 있습니다. 표준 예약 인스턴스는 1년 또는 3년 기간으로 구입할 수 있으며, 해당 기간 동안 단일 인스턴스 패밀리, 플랫폼, 범위 및 테넌시에 적용됩니다.

전환형 예약 인스턴스는 3년 기간으로 구입할 수 있으며 이 기간 동안 다른 인스턴스 패밀리, 플랫폼, 테넌시 또는 범위의 전환형 예약 인스턴스와 교환할 수 있습니다.

표준 예약 인스턴스와 전환형 예약 인스턴스는 특정 가용 영역 또는 한 리전의 인스턴스에 적용하도록 구입할 수 있습니다. 특정 가용 영역에 대해 구입한 표준 예약 인스턴스는 리전에 적용하도록 수정할 수 있습니다. 하지만— 이렇게 하면 관련된 용량 예약이 상실됩니다.

전환형 예약 인스턴스는 인스턴스 유형, 플랫폼, 범위, 테넌시 등을 비롯하여 구성이 완전히 다른 전환형 예약 인스턴스와 교환할 수 있습니다. 표준 예약 인스턴스는 이렇게 교환할 수 없습니다. 전환형 예약 인스턴스를 구입한 후에는 범위를 수정할 수 없습니다. 자세한 내용은 [표준 예약 인스턴스 변경 \(p. 193\)](#) 및 [전환형 예약 인스턴스 교환 \(p. 198\)](#) 섹션을 참조하십시오.

예약 인스턴스의 작동 방식

Amazon EC2 예약 인스턴스와 예약 인스턴스 마켓플레이스는 강력하면서도 비용 절감이 가능한 비즈니스 운영 전략으로 활용할 수 있습니다. 그러나 예약 인스턴스나 예약 인스턴스 마켓플레이스를 차질 없이 사용하려면 먼저 구매와 판매에 관련된 요구 조건을 확인해야 합니다. 예약 인스턴스와 예약 인스턴스 마켓플레이스의 특정 요소에 대한 세부 정보와 제한 사항도 알아두어야 합니다. 이러한 제한 사항에는 판매자 등록, 은행 업무, AWS 프리 티어 사용, 취소된 인스턴스 취급 등이 포함될 수 있습니다. 이 항목을 참고하여 예약

인스턴스의 구매와 판매, 그리고 예약 인스턴스 마켓플레이스 상에서의 구매와 판매 활동 시 체크리스트로 활용하십시오.

Note

예약 인스턴스를 구매하고 수정하려면 가용 영역을 설명할 수 있는 권한과 같은 적절한 권한이 IAM 사용자 계정에 있는지 확인하십시오. 자세한 내용은 [AWS CLI 또는 AWS SDK 작업을 위한 예제 정책](#) 및 [Amazon EC2 콘솔 작업을 위한 예제 정책](#)을 참조하십시오.

시작하기

- AWS 계정 — 예약 인스턴스를 구매하려면 AWS 계정이 필요합니다. AWS 계정이 없는 사용자는 [Amazon EC2로 설정 \(p. 16\)](#) 섹션에 나와 있는 지침에 따라 작업을 완료해야 합니다. 이 섹션에서는 Amazon EC2 계정 가입과 자격 증명에 대한 정보를 제공합니다.
- AWS 프리 티어: AWS 프리 티어는 신규 AWS 계정에 제공됩니다. AWS 프리 티어를 사용하여 Amazon EC2 인스턴스를 실행하는 사용자는 예약 인스턴스를 구매할 때 표준 요금 정책에 따라 요금이 부과됩니다. 관련 서비스 및 사용량에 대한 자세한 내용은 [AWS 프리 티어](#) 섹션을 참조하십시오.

예약 인스턴스 구매

- 사용 요금 — 예약 인스턴스를 사용하면 사용 여부와 상관없이 약관 기간에 걸쳐 특정 요금을 지불해야 합니다.
- 구매 시 티어 할인: 예약 인스턴스 요금 티어의 할인 혜택은 AWS를 통한 구매에만 적용됩니다. 외부 판매자의 예약 인스턴스를 구매할 때는 이 할인이 적용되지 않습니다. 자세한 내용은 [예약 인스턴스 할인 요금 티어 \(p. 181\)](#) 섹션을 참조하십시오.
- 구매 취소 — 구매를 확정하기 전에 구매하기로 결정한 예약 인스턴스의 세부 정보를 검토하고 모든 파라미터 설정이 정확한지를 확인하십시오. 예약 인스턴스는 한 번 구매하고 나면 구매를 취소할 수 없습니다 (예약 인스턴스 마켓플레이스의 타사에서 구매한 경우와 AWS에서 구매한 경우 모두 동일). 단, 변경이 필요한 경우 예약 인스턴스를 판매할 수 있습니다. 자세한 내용은 [예약 인스턴스 판매 등록 \(p. 189\)](#) 섹션을 참조하십시오.

예약 인스턴스 판매와 예약 인스턴스 마켓플레이스

- 전환형 예약 인스턴스 - 예약 인스턴스 마켓플레이스에서는 Amazon EC2 표준 예약 인스턴스만 판매할 수 있습니다. 전환형 예약 인스턴스는 판매할 수 없습니다.
- 예약 인스턴스 범위 — 예약 인스턴스 마켓플레이스에서는 가용 영역을 대상으로 한 표준 예약 인스턴스만 판매할 수 있습니다. 리전을 대상으로는 예약 인스턴스를 판매할 수 없습니다.
- 판매자 요구 조건: 예약 인스턴스 마켓플레이스에서 판매자가 되려면 반드시 판매자로 등록해야 합니다. 자세한 내용은 [예약 인스턴스 판매 등록 \(p. 189\)](#) 섹션을 참조하십시오.
- 은행 요건 - AWS에서 예약 인스턴스를 판매할 경우 판매 대금을 지급하기 위해 사용자의 은행 정보가 필요합니다. 이때 미국 소재지가 있는 은행을 선택해야 합니다. 자세한 내용은 [은행 계좌 \(p. 188\)](#) 섹션을 참조하십시오.
- 세금 요구 조건 — 거래 수가 50건 이상이거나 20,000 USD 이상의 표준 예약 인스턴스를 판매할 계획이 있는 판매자는 세금 정산에 필요한 추가 정보를 제공해야 합니다. 자세한 내용은 [세금 정보 \(p. 188\)](#) 섹션을 참조하십시오.
- 최소 판매가 - 예약 인스턴스 마켓플레이스에서 허용되는 최소 판매가는 0.00 USD입니다.
- 표준 예약 인스턴스 판매 가능 시기 - 표준 예약 인스턴스는 AWS에 선결제가 완료된 이후, 그리고 활성 기간(소유 기간)이 30일 이상인 경우에만 판매할 수 있습니다. 또한, 판매 등록하려는 표준 예약 인스턴스의 남은 사용 기간이 한 달 이상이어야 합니다.
- 판매 등록 항목 변경 - 현재 등록된 항목을 예약 인스턴스 마켓플레이스에서 바로 변경하는 것은 불가능합니다. 하지만 판매 등록을 취소하고 새 파라미터를 지정한 다음 다시 등록하는 방식으로 변경하는 것은 가능합니다. 자세한 내용은 [예약 인스턴스의 가격 책정 \(p. 190\)](#) 섹션을 참조하십시오. 판매 등록하기 전에 예약 인스턴스를 수정할 수도 있습니다. 자세한 내용은 [표준 예약 인스턴스 변경 \(p. 193\)](#) 섹션을 참조하십시오.

- 할인 적용된 표준 예약 인스턴스 판매 - Amazon EC2 표준 예약 인스턴스 중 할인 티어로 더 낮은 가격에 구매한 인스턴스는 예약 인스턴스 마켓플레이스에서 판매할 수 없습니다. 자세한 내용은 [예약 인스턴스 마켓플레이스 \(p. 177\)](#) 섹션을 참조하십시오.
- 서비스 수수료 - AWS는 예약 인스턴스 마켓플레이스에서 판매하는 각각의 표준 예약 인스턴스에 대해 총 선결제 금액의 12%를 서비스 수수료로 청구합니다. 선결제 금액은 판매자가 판매 등록한 표준 예약 인스턴스에 책정한 가격입니다.
- 다른 AWS 예약 인스턴스 - 예약 인스턴스 마켓플레이스에서는 Amazon EC2 표준 예약 인스턴스만 판매 할 수 있습니다. 그 외 Amazon RDS와 Amazon ElastiCache 예약 인스턴스와 같은 다른 AWS 예약 인스턴스는 예약 인스턴스 마켓플레이스에서 판매할 수 없습니다.

VPC에서의 예약 인스턴스 사용

VPC로 인스턴스를 실행하여 표준 예약 인스턴스의 혜택을 받을 수 있습니다. 자세한 내용은 [What is Amazon VPC?](#)(출처: Amazon VPC 사용 설명서) 섹션을 참조하십시오.

EC2-Classic 계정이 있다면 예약 인스턴스를 구매해 그 이름에 Amazon VPC를 포함하는 플랫폼을 선택함으로써 기본이 아닌 VPC로 시작된 인스턴스에 적용할 수 있습니다. 자세한 내용은 [Detecting Your Supported Platforms and Whether You Have a Default VPC](#)를 참조하십시오.

EC2 VPC 전용 계정이라면 모든 플랫폼에 기본 서브넷이 존재하므로 사용 가능한 플랫폼 목록에는 그 이름에 Amazon VPC가 포함된 플랫폼이 들어 있지 않습니다. 예약된 용량과 동일한 구성의 인스턴스를 시작하고 그 인스턴스가 기본 또는 기본이 아닌 VPC에서 시작된다면, 용량 예약과 청구 혜택이 해당 인스턴스에 자동으로 적용됩니다. 자세한 내용은 [Amazon VPC 사용 설명서](#)의 기본 VPC 및 서브넷을 참조하십시오.

또한 인스턴스 테넌시를 전용으로 지정하여 호스트 하드웨어 수준에서 물리적으로 격리되는 예약 인스턴스를 선택하여 구매할 수 있습니다. 자세한 내용은 [전용 인스턴스 \(p. 257\)](#) 섹션을 참조하십시오.

예약 인스턴스 마켓플레이스

예약 인스턴스 마켓플레이스은 타사 및 AWS 고객의 사용하지 않은 표준 예약 인스턴스 판매를 지원하는 플랫폼으로, 사용 기간 및 요금 옵션이 다릅니다. 예를 들어, AWS 고객은 인스턴스를 새로운 AWS 리전으로 이동하거나, 새 인스턴스 유형으로 변경한 후에 또는 약정이 끝나기 전에 프로젝트가 종료될 경우 예약 인스턴스를 판매할 수 있습니다.

예약 인스턴스 마켓플레이스를 이용하면 특정 비즈니스 요구 사항에 따라 다양하고 유연하게 선택할 수 있습니다. 원하는 인스턴스 유형, 리전 및 기간에 가장 적합한 예약 인스턴스를 찾아보십시오.

Note

예약 인스턴스 마켓플레이스에서는 Amazon EC2 표준 예약 인스턴스만 판매할 수 있습니다.
Amazon RDS 및 Amazon ElastiCache 예약 인스턴스와 같은 다른 유형은 예약 인스턴스 마켓플레이스에서 판매할 수 없습니다.

청구 혜택과 결제 옵션

모든 예약 인스턴스는 온디맨드 요금과 비교하여 할인된 요금을 제공합니다. 가용 영역에 할당되는 예약 인스턴스는 용량을 예약할 수 있습니다. 이후 특정 리전에서 예약 인스턴스를 구매하면서(리전 단위의 예약 인스턴스) 이 용량 예약을 포기할 수도 있습니다. 리전 단위의 예약 인스턴스로 가용 영역과 인스턴스의 크기를 유연하게 바꿀 수 있습니다. 이러한 유연성은 예약 인스턴스의 할인 혜택을 더욱 쉽게 이용할 수 있는 효과가 있습니다.

예약 인스턴스 적용

예약 인스턴스는 제공 유형(표준 또는 전환형)과 상관없이 동일한 방식으로 사용량에 적용되기 때문에 테넌시, 플랫폼 등 일치하는 명세에 따라 실행 중인 온디맨드 인스턴스에 자동 적용됩니다. 예약 인스턴스를 특정 가용 영역에 할당하면 해당 가용 영역에서 일치하는 인스턴스 사용량에 대해서는 예약 인스턴스의 할인 혜택이 제공됩니다.

리전 단위의 예약 인스턴스는 가용 영역의 유연성을 제공합니다. 이 밖에도 Linux/Unix 플랫폼을 기반으로 기본 테넌시가 포함된 리전 단위의 예약 인스턴스 역시 인스턴스 크기 유연성을 제공합니다. 가용 영역의 유연성은 해당 리전에 속한 모든 가용 영역의 인스턴스 사용량에 대해 예약 인스턴스의 할인 혜택을 제공합니다. 또한 인스턴스 크기 유연성은 해당 인스턴스 유형 내에서 크기에 상관없이 인스턴스 사용량에 대해 예약 인스턴스의 할인 혜택을 제공합니다.

Note

인스턴스 크기 유연성은 기본 테넌시를 포함하여 리전에 할당되는 Linux/Unix 기반 예약 인스턴스에서만 지원됩니다. Windows, Windows with SQL Standard, Windows with SQL Server Enterprise, Windows with SQL Server Web, RHEL 및 SLES 예약 인스턴스에는 인스턴스 크기 유연성이 적용되지 않습니다.

가용 영역 us-east-1a에서 c4.xlarge 기본 테넌시 Linux/Unix 표준 예약 인스턴스 2개를 구매하면 가용 영역 us-east-1a에서 실행하는 c4.xlarge 기본 테넌시 Linux/Unix 인스턴스에 최대 2개까지 예약 인스턴스의 할인 혜택을 적용할 수 있습니다. 단, 실행할 인스턴스의 명세(테넌시, 플랫폼, 가용 영역, 인스턴스 유형, 인스턴스 크기)가 예약 인스턴스의 명세와 일치해야 합니다.

미국 동부(버지니아 북부)에서 c4.xlarge 기본 테넌시 Amazon Linux/Unix 예약 인스턴스 4개를 구매하면 크기에 상관없이 미국 동부(버지니아 북부) 리전에 속한 모든 가용 영역에서 계정의 모든 c4 인스턴스에 예약 인스턴스 할인 혜택이 자동 적용됩니다. 이때는 인스턴스 유형, 테넌시 및 플랫폼만 일치하면 됩니다.

아래 표는 인스턴스 유형 내에서 다른 크기와 그에 따른 정규화 인자를 설명한 것입니다. 인스턴스 크기 유연성의 경우에는 예약 인스턴스의 할인 요금을 정규화된 인스턴스 유형 사용량에 적용하는 데 이 배율이 사용됩니다.

표준 예약 인스턴스를 수정하면 정규화 인자 역시 적용됩니다. 자세한 내용은 [표준 예약 인스턴스 변경 \(p. 193\)](#) 섹션을 참조하십시오.

인스턴스 크기	정규화 인자
nano	0.25
micro	0.5
small	1
medium	2
large	4
xlarge	8
2xlarge	16
4xlarge	32
8xlarge	64
10xlarge	80
32xlarge	256

예약 인스턴스의 적용에 대한 이해

다음은 예약 인스턴스가 적용되는 방식을 설명한 시나리오입니다.

시나리오 1

고객이 계정 A에서 다음과 같은 온디맨드 인스턴스를 실행하고 있습니다.

- 4 x m3.large Linux, 가용 영역 us-east-1a의 기본 테넌시 인스턴스
- 2 x m4.xlarge Amazon Linux, 가용 영역 us-east-1b의 기본 테넌시 인스턴스
- 1 x c4.xlarge Amazon Linux, 가용 영역 us-east-1c의 기본 테넌시 인스턴스

그런 다음 고객은 계정 A에서 다음 예약 인스턴스를 구입합니다.

- 4 x m3.large Linux, 가용 영역 us-east-1a의 기본 테넌시 예약 인스턴스(용량 예약됨)
- 4 x m4.large Amazon Linux, us-east-1의 기본 테넌시 예약 인스턴스
- 1 x c4.large Amazon Linux, us-east-1의 기본 테넌시 예약 인스턴스

예약 인스턴스의 혜택은 다음과 같이 적용됩니다.

- 예약 인스턴스와 인스턴스의 속성(인스턴스 크기, 리전, 플랫폼, 테넌시)이 일치하기 때문에 m3.large 예약 인스턴스 4개의 할인 및 용량 예약이 m3.large 인스턴스 4개에 적용됩니다.
- m4.large 예약 인스턴스는 기본 테넌시가 포함된 Amazon Linux 예약 인스턴스이기 때문에 가용 영역 및 인스턴스 크기 유연성이 제공됩니다.

m4.large는 시간당 정규화 유닛 4개와 같습니다.

고객이 m4.large 예약 인스턴스를 4개 구매하였으며, 이에 따라 시간당 정규화 유닛은 총 16개(4x4)와 같습니다. 현재 계정 A에는 실행 중인 m4.xlarge 인스턴스가 2개이며, 이에 따라 시간당 정규화 유닛은 16개(2x8)와 같습니다. 이 경우 m4.large 예약 인스턴스가 4개이므로 m4.xlarge 인스턴스 2개의 전체 시간 사용량에 결제 혜택이 제공됩니다.

- us-east-1의 c4.large 예약 인스턴스는 기존 테넌시가 포함된 Amazon Linux 예약 인스턴스이므로 가용 영역 및 인스턴스 크기 유연성을 c4.xlarge 인스턴스에 적용합니다. c4.large 인스턴스는 시간당 정규화 유닛 4개와 같고, c4.xlarge는 시간당 정규화 유닛 8개와 같습니다.

이 경우에는 c4.large 예약 인스턴스가 c4.xlarge 사용량에 부분적 혜택을 제공합니다. 이는 c4.large 예약 인스턴스가 사용량이 시간당 정규화 유닛이 4개와 같지만 c4.xlarge 인스턴스는 시간당 정규화 유닛이 8개와 같기 때문입니다. 따라서 c4.large 예약 인스턴스의 결제 할인이 c4.xlarge 사용량의 50%에 적용됩니다. 나머지 c4.xlarge 사용량은 온디맨드 요금이 부과됩니다.

리전 단위의 Linux/Unix 예약 인스턴스는 인스턴스 패밀리 내에서 리전, 테넌시 및 플랫폼이 일치하는 모든 사용량에 적용됩니다. 예약 인스턴스가 구매 계정 내 사용량에 먼저 적용된 후 조직의 다른 계정에서 해당하는 사용량에 적용됩니다. 크기 유연성을 제공하는 예약 인스턴스의 경우에는 예약 인스턴스 패밀리 내의 인스턴스 크기에 대해서는 아무런 우선권도 없습니다. 예약 인스턴스 할인은 AWS 결제 시스템에서 먼저 감지되는 해당 사용량에 적용됩니다. 다음 예제를 참조하십시오.

시나리오 2

고객이 계정 A에서 다음과 같은 온디맨드 인스턴스를 실행하고 있습니다.

- 2 x m4.xlarge Linux, 가용 영역 us-east-1a의 기본 테넌시 인스턴스
- 1 x m4.2xlarge Linux, 가용 영역 us-east-1b의 기본 테넌시 인스턴스
- 2 x c4.xlarge Linux, 가용 영역 us-east-1a의 기본 테넌시 인스턴스
- 1 x c4.2xlarge Linux, 가용 영역 us-east-1b의 기본 테넌시 인스턴스

고객이 연결 계정인 계정 B—에서 다음과 같은 온디맨드 인스턴스를 실행하고 있습니다.

- 2 x m4.xlarge Linux, 가용 영역 us-east-1a의 기본 테넌시 인스턴스

그런 다음 고객은 계정 A에서 다음 예약 인스턴스를 구입합니다.

- 4 x m4.xlarge Linux, us-east-1의 기본 테넌시 예약 인스턴스
- 2 x c4.xlarge Linux, us-east-1의 기본 테넌시 예약 인스턴스

예약 인스턴스의 혜택은 다음과 같이 적용됩니다.

- 4개의 m4.xlarge 예약 인스턴스 할인은 계정 A의 m4.xlarge 인스턴스 2개와 계정 A의 m4.2xlarge 인스턴스에서 사용됩니다. 이 3개의 인스턴스는 모두 속성(인스턴스 패밀리, 리전, 플랫폼, 테넌시)이 일치합니다. 용량 예약은 없습니다.
- 2개의 c4.xlarge 예약 인스턴스 할인은 결제 시스템에서 처음 감지되는 사용량에 따라 c4.xlarge 인스턴스 2개 또는 c4.2xlarge 인스턴스에 적용될 수 있으며, 이때 각 인스턴스는 속성(인스턴스 패밀리, 리전, 플랫폼, 테넌시)이 일치합니다. 특정 인스턴스 크기에 대해서는 아무런 우선권도 없습니다. 용량 예약은 없습니다.

일반적으로 계정에 속한 예약 인스턴스가 해당 계정의 사용량에 먼저 적용됩니다. 하지만 조직 내 다른 계정에 자격을 갖추었지만 아직 사용하지 않은 영역 단위의 예약 인스턴스가 있다면 계정에 속한 리전 단위의 예약 인스턴스에 앞서 이 인스턴스가 계정에 적용됩니다. 이는 예약 인스턴스의 활용도를 극대화하면서 결제 비용을 낮추기 위한 것입니다. 결제의 편의를 위해 조직 내 모든 계정은 하나의 계정으로 취급됩니다. 다음 예제를 참조하십시오.

시나리오 3

고객이 계정 A에서 다음과 같은 인스턴스를 실행하고 있습니다.

- 1 x m4.xlarge Linux, 가용 영역 us-east-1a의 기본 테넌시 인스턴스

고객이 다른 연결 계정 B에서 다음과 같은 인스턴스를 실행하고 있습니다.

- 1 x m4.xlarge Linux, 가용 영역 us-east-1b의 기본 테넌시 인스턴스

그런 다음 고객은 계정 A에서 다음 예약 인스턴스를 구입합니다.

- 1 x m4.xlarge Linux, 가용 영역 us-east-1의 기본 테넌시 예약 인스턴스

또한 계정 C에서 다음과 같은 예약 인스턴스를 구입합니다.

- 1 x m4.xlarge Linux, 가용 영역 us-east-1a의 기본 테넌시 예약 인스턴스

예약 인스턴스의 혜택은 다음과 같이 적용됩니다.

- 계정 C에 속한 m4.xlarge 예약 인스턴스의 할인은 계정 A의 m4.xlarge 사용량에 적용됩니다.
- 계정 A에 속한 m4.xlarge 예약 인스턴스의 할인은 계정 B의 m4.xlarge 사용량에 적용됩니다.
- 계정 A에 속한 예약 인스턴스의 할인이 계정 A의 사용량에 먼저 적용된 경우에는 계정 C에 속한 예약 인스턴스가 미사용 상태로 남게 되고 계정 B의 사용량은 온디マン드 요금으로 부과됩니다.

자세한 내용은 [Reserved Instances in the Billing and Cost Management Report](#) 단원을 참조하십시오.

예약 인스턴스 결제 방식 선택

예약 인스턴스 결제 방식은 세 가지입니다.

- 선결제 없음 - 사용 기간 동안 사용량에 상관없이 매시간마다 할인된 시간당 요금이 청구되며, 선결제가 필요없습니다. 이 옵션은 표준 예약 인스턴스의 경우 1년 예약, 전환형 예약 인스턴스의 경우 3년 예약으로만 사용할 수 있습니다.

Note

선결제 방식이 아닌 예약 인스턴스는 전체 예약 기간 동안 매월 결제하는 계약 조건입니다. 따라서 선결제 방식이 아닌 예약 인스턴스를 구입할 자격이 되려면 해당 계정의 결제 기록에 미납액이 없어야 합니다.

- 부분 선결제 - 비용의 일부를 선결제하고, 사용 기간 내 나머지 시간에 대해서는 사용량에 상관없이 할인된 시간당 요금이 청구됩니다.
- 전체 선결제 — 약관이 시작되는 시점에서 모든 금액을 결제하고 사용 기간 동안 추가 비용 없이 무제한으로 사용할 수 있습니다.

시간 기준 청구에 대한 이해

예약 인스턴스는 선택한 기간 동안 인스턴스 실행 여부와 상관없이 매시간 청구됩니다. 인스턴스 상태 간의 차이점과 이런 상태가 요금 청구 시간에 미치는 영향을 이해하는 것이 중요합니다. 자세한 내용은 [인스턴스 수명 주기 \(p. 261\)](#) 섹션을 참조하십시오.

예약 인스턴스 청구 혜택은 매 시간마다 한 인스턴스 시간에만 적용됩니다. 인스턴스 시간은 인스턴스가 시작될 때 시작되어 60분간 계속되거나 인스턴스 종지 또는 종료 중 어느 것이든 먼저 발생할 때까지 계속됩니다. 시계로 표시되는 시간은 자정부터 다음날 자정까지의 표준 24시간제로 정의되고, 24시간으로 나뉩니다 (예: 1:00:00부터 1:59:59까지가 시계상의 한 시간임).

한 인스턴스가 60분간 계속 실행되었거나 인스턴스가 종지된 다음에 시작된 경우 새로운 인스턴스 시간이 시작됩니다. 인스턴스를 재부팅해도 실행 중인 인스턴스 시간은 재설정되지 않습니다.

예를 들어 어떤 시계상의 시간 중에 한 인스턴스가 종지된 후 다시 시작되어 2시간 동안 더 계속 실행 중인 경우, 첫 번째 인스턴스 시간(다시 시작하기 전)에 대해서는 할인된 예약 인스턴스 요금이 청구됩니다. 다음 인스턴스 시간(다시 시작한 후)에 대해서는 온디맨드 요금이 청구되고, 그 다음 2시간의 인스턴스 시간에 대해서는 할인된 예약 인스턴스 요금이 청구됩니다.

[예약 인스턴스 사용률 보고서 \(p. 693\)](#) 단원에는 실행 중인 온디맨드 인스턴스에 대해 절약된 금액을 보여주는 샘플 보고서가 나와 있습니다. [예약 인스턴스 FAQ](#)에는 정가 계산의 예가 나와 있습니다.

예약 인스턴스 할인 요금 티어

할인 요금 티어의 사용 자격에 해당되는 계정은 적용 시점부터 구매한 예약 인스턴스 중 해당 티어에 속하는 모든 예약 인스턴스의 선결제 금액과 시간당 사용비가 자동으로 할인됩니다. 할인은 한 리전 내 예약 인스턴스의 정가 총합이 500,000 USD 이상인 경우만 해당됩니다.

Note

할인 요금 티어는 현재 전환형 인스턴스 구입에는 적용되지 않습니다.

항목

- [예약 인스턴스 요금 할인 계산 \(p. 181\)](#)
- [요금 티어 통합 결제 \(p. 182\)](#)
- [구매 시 할인 티어 적용 \(p. 182\)](#)
- [현 요금 티어의 제한 사항 \(p. 182\)](#)
- [요금 티어 교차 \(p. 183\)](#)

예약 인스턴스 요금 할인 계산

리전의 모든 예약 인스턴스에 대한 정가를 계산하여 계정에 대한 요금 티어를 확인할 수 있습니다. 각 예약 인스턴스의 정가는 시간당 부과 요금(hourly recurring price)에 남은 약정 시간을 곱한 다음, 구매 시 지불한 [AWS 마케팅 웹 사이트](#) 상의 정가, 즉 할인이 적용되지 않은 예약 인스턴스 선결제 금액(fixed price: 고정 가격)을 더한 가격입니다. 정가는 할인이 적용되지 않은 요금 또는 (공개) 요금을 기준으로 하기 때문에 볼륨 할인을 적용받는 경우나 예약 인스턴스 구매 후 가격이 내려가는 경우 정가에는 영향을 주지 않습니다.

```
List value = fixed price + (undiscounted recurring hourly price * hours in term)
```

AWS Management Console을 사용하여 예약 인스턴스의 고정 가격을 확인하는 방법

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 화면 상단 오른쪽의 [Show/Hide]를 선택하여 [Fixed Price] 열을 표시합니다.

명령을 사용하여 예약 인스턴스의 고정 가격을 확인하는 방법

- AWS CLI를 사용할 경우 [describe-reserved-instances](#)를 참조하십시오.
- Windows PowerShell용 AWS 도구를 사용할 경우 [Get-EC2ReservedInstance](#)를 참조하십시오.
- Amazon EC2 API를 사용할 경우 [DescribeReservedInstances](#)를 참조하십시오.

요금 티어 통합 결제

통합 결제 계정은 한 리전 내 회원 계정의 정가를 합산합니다. 통합 결제 계정에 속하는 사용 중인 모든 예약 인스턴스의 정가 총액이 할인 요금 티어의 기준 금액에 도달하면, 통합 결제 계정의 모든 구성원 계정에서 할인을 받을 수 있습니다(해당 통합 결제 계정의 정가가 할인 요금 티어의 기준 금액 이상으로 유지되는 동안 계속 적용). 자세한 내용은 [예약 인스턴스 및 통합 결제 \(p. 183\)](#) 섹션을 참조하십시오.

구매 시 할인 티어 적용

Amazon EC2에서는 예약 인스턴스를 구매할 때 할인 요금 티어에 해당되는 구매에 대해 자동으로 그에 맞는 할인을 적용합니다. 추가 작업 없이 어떤 Amazon EC2 도구에서나 구매가 가능합니다. 자세한 내용은 [예약 인스턴스 마켓플레이스에서 구입 \(p. 186\)](#) 섹션을 참조하십시오.

Note

할인 요금 티어는 예약 인스턴스 구매 내역만을 기준으로 결정되며, 할인은 Amazon EC2 예약 인스턴스 구매 시에만 적용됩니다.

한 리전에서 사용 중인 예약 인스턴스의 정가 총액이 할인 요금 티어 기준에 도달하면 다음에 같은 리전에서 예약 인스턴스를 구매할 때 할인이 적용됩니다. 어떤 리전에서 하나의 예약 인스턴스를 구매했는데 그에 따른 합계가 할인 요금 티어 기준 금액을 초과하는 경우, 기준을 초과한 금액에 대해 할인이 적용됩니다. 구매 과정에서 생성되는 임시 예약 인스턴스 ID에 대한 자세한 내용은 [요금 티어 교차 \(p. 183\)](#) 섹션을 참조하십시오.

정가 총액이 예약 인스턴스 만료 등의 이유로 이용 종이던 할인 요금 티어 기준 이하로 변경되면, 그 다음에 해당 리전에서 예약 인스턴스를 구매할 때는 할인이 적용되지 않습니다. 단, 구매 시 할인 요금 티어 범위에 해당되었던 기존의 예약 인스턴스에 대해서는 계속 할인을 받을 수 있습니다.

예약 인스턴스 구매 상황은 다음 네 가지 중 한 경우입니다.

- 미할인 - 같은 리전에서 구매한 합계가 할인 기준 금액보다 아직 적은 경우입니다.
- 부분 할인 - 같은 리전에서 구매하면서 최하 등급의 할인 티어 기준 금액에 도달한 경우입니다. 미할인이 하나 이상의 예약에 적용되고 할인 요금이 나머지 예약에 적용됩니다.
- 전체 할인 - 한 리전 내의 전체 구매가 동일한 할인 티어에 해당되고 적절히 할인됩니다.
- 이중 할인 - 같은 리전에서 구매하면서 할인 티어 등급이 기준보다 더 높아진 경우입니다. 이 경우 두 가지 요금이 차등 적용됩니다. 합산 가격을 기준으로 하나 또는 그 이상의 예약 인스턴스에는 기준 티어 할인이, 나머지 인스턴스에는 상위 티어 할인이 적용됩니다.

현 요금 티어의 제한 사항

다음 제한 사항은 예약 인스턴스 요금 티어에 현재 적용되는 것입니다.

- 예약 인스턴스 요금 티어와 관련 할인은 Amazon EC2 예약 인스턴스 구매 시에만 적용됩니다.
- SQL Server Standard 사용 Windows 또는 SQL Server Web 사용 Windows의 예약 인스턴스는 예약 인스턴스 요금이 적용되지 않습니다.
- 구매 시 티어 할인이 적용된 예약 인스턴스는 예약 인스턴스 마켓플레이스에서 판매할 수 없습니다. 자세한 내용은 [예약 인스턴스 마켓플레이스 \(p. 177\)](#) 페이지를 참조하십시오.

요금 티어 교차

구매 시점에서 합산 금액이 어떤 할인 요금 티어 기준을 도달하게 되면 함께 구매하는 인스턴스 중 일부는 정상적인 예약 인스턴스 가격이 적용되고 티어 기준을 초과하는 인스턴스는 티어에 따른 할인이 적용됩니다.

함께 구매한 인스턴스에 미활인 티어(정상 가격), 하나 이상의 할인 티어가 차등 적용되므로, 예약 인스턴스 서비스에서는 여러 개의 예약 인스턴스 ID를 생성합니다. ID는 같은 티어의 인스턴스를 둘러 티어당 하나씩 부여됩니다. 따라서, CLI 명령이나 API 작업으로 구매했을 때 부여되는 ID는 새로 구매한 예약 인스턴스의 실제 ID와는 다릅니다.

예약 인스턴스 및 통합 결제

단일 통합 결제 지급인 계정으로 여러 계정의 결제를 함께 관리하는 경우, 이런 계정에서 구입한 예약 인스턴스의 요금 혜택은 구성원 계정 간에 공유됩니다. 모든 하위 계정에서 발생한 시간당 요금 또한 매월 지급인 계정으로 합산됩니다. 이 방식은 일반적으로 직무가 서로 다른 팀이나 그룹이 있는 회사에서 유용하며, 정상적인 예약 인스턴스 규칙에 따라 요금이 계산됩니다. 자세한 내용은 [AWS Billing and Cost Management 사용 설명서](#)의 통합 결제를 참조하십시오.

통합 결제 계정에 대한 예약 인스턴스 요금 티어의 할인 적용에 대한 자세한 내용은 [Amazon EC2 예약 인스턴스](#) 섹션을 참조하십시오.

내역 확인 (인보이스)

계정으로 청구되는 요금과 비용은 AWS Management Console의 [Billing & Cost Management] 페이지에서 확인할 수 있습니다. 페이지는 계정 이름 옆의 화살표를 선택하여 액세스할 수 있습니다.

- [Dashboard] 페이지에는 선결제 금액과 일회성 요금, 기본 요금 등 계정으로 청구되는 모든 요금이 표시됩니다. 청구 내역을 요약 목록 및 세부 목록으로 볼 수 있습니다.
- 예약 인스턴스 마켓플레이스에서 구매한 외부 판매자 예약 인스턴스에 관련된 선결제 금액은 [AWS Marketplace Charges] 섹션에서 확인할 수 있으며, 각 항목 옆에 판매자의 이름이 표시됩니다. 이러한 예약 인스턴스에 대한 모든 기본 요금이나 사용 요금은 [AWS Service Charges] 섹션에 표시됩니다.
- [Detail] 섹션에 가용 영역, 인스턴스 유형, 비용, 인스턴스 개수 등 예약 인스턴스에 대한 정보가 나와 있습니다.

요금은 온라인으로 확인할 수 있으며 요금 정보를 PDF로 다운로드하는 것도 가능합니다.

예약 인스턴스 구매

특정 유형의 예약 인스턴스를 구입하기 위해 검색할 수 있으며, 찾고 있는 인스턴스와 정확히 일치하는 인스턴스를 찾을 때까지 파라미터를 조정할 수 있습니다.

예약 인스턴스를 구입하려는 경우 다음 사항을 반드시 알아 두어야 합니다.

- 사용 요금 - 예약 인스턴스를 사용하면 실제 사용에 상관없이 전체 기간에 걸쳐 특정 요금을 지불해야 합니다.
- 구매 시 티어 할인 - 요금 티어 할인 혜택은 AWS 표준 예약 인스턴스 구입에만 적용됩니다. 이러한 할인은 타사 예약 인스턴스 또는 전환형 예약 인스턴스 구입에는 적용되지 않습니다. 자세한 내용은 [예약 인스턴스 할인 요금 티어 \(p. 181\)](#) 섹션을 참조하십시오.
- 구매 취소 — 구매를 확정한 이후에는 구매를 취소할 수 없습니다. 확정하기 전에 구매하기로 결정한 예약 인스턴스의 세부 정보를 검토하고 모든 파라미터 설정이 정확한지 확인하십시오. 단, 필요 여부가 변경될

경우 요구 사항을 충족한다면 예약 인스턴스를 판매할 수 있습니다. 자세한 내용은 [예약 인스턴스 마켓플레이스 판매 \(p. 187\)](#) 섹션을 참조하십시오.

구매할 예약 인스턴스를 선택하면 선택 내역에 따른 총 비용을 산출한 견적을 받게 됩니다. 내역대로 구매를 진행하기로 결정하면 AWS에서 구매 가격에 제한 가격을 자동으로 설정합니다. 구매하는 예약 인스턴스의 총 가격이 제시된 견적가를 초과하지 않게 됩니다.

여하한 이유로 가격이 오르거나 변경되면 이전 화면으로 돌아가고 구매가 완료되지 않습니다. 구매 당시 선택한 조건과 비슷한데 가격은 더 낮은 상품이 있을 경우 AWS는 더 저렴한 상품을 판매합니다.

AWS Management Console을 사용하여 표준 예약 인스턴스 구입

용량 예약을 지정하거나 지정하지 않고 표준 예약 인스턴스를 구입할 수 있습니다.

AWS Management Console을 사용하여 표준 예약 인스턴스를 구입하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Reserved Instances], [Purchase Reserved Instances]를 선택합니다.
3. [Offering Class]와 [Standard]를 차례대로 선택하여 표준 예약 인스턴스를 표시합니다.
4. 용량 예약을 구입하려면 구입 화면의 상단 오른쪽 모서리 부분에서 [Only show offerings that reserve capacity]를 선택합니다.
5. 필요에 따라 다른 구성을 선택하고 [Search]를 선택합니다.

Note

검색 결과에서 [Seller] 열에는 해당 판매자가 제3자인지 여부가 표시됩니다. 제3자일 경우 [Term] 열에 비 표준 약정이 표시됩니다.

6. 구입할 예약 인스턴스를 선택하고 수량을 입력한 후 [Add to Cart]를 선택합니다.
7. 선택한 예약 인스턴스의 요약을 보려면 [View Cart]를 선택합니다.
8. 주문을 완료하려면 [Purchase]를 선택합니다.

Note

구매 당시 선택한 조건과 비슷한데 가격은 더 낮은 상품이 있을 경우 AWS는 더 저렴한 상품을 판매합니다.

예약을 적용하려면 예약 인스턴스에 대해 지정한 것과 동일한 기준을 지정하여 온디맨드 인스턴스를 시작합니다. AWS에서는 할인된 시간당 요금을 자동으로 적용합니다. 인스턴스를 따로 재시작할 필요가 없습니다.

AWS Management Console을 사용하여 거래 상태를 보려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. [Reserved Instances] 페이지를 선택합니다. 구매 상태가 [State] 열에 표시됩니다. 주문이 완료되면 [State] 값이 payment-pending에서 active로 바뀝니다.

AWS Management Console을 사용하여 전환형 예약 인스턴스 구입

용량 예약을 지정하거나 지정하지 않고 전환형 예약 인스턴스를 구입할 수 있습니다.

AWS Management Console을 사용하여 전환형 예약 인스턴스를 구입하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Reserved Instances]를 선택합니다.
3. [Reserved Instances] 페이지에서 [Purchase Reserved Instances]를 선택합니다.
4. [Offering Class]를 선택하고 [Convertible]을 선택하여 전환형 예약 인스턴스를 표시합니다.

5. 용량 예약을 구입하려면 구입 화면의 상단 오른쪽 모서리 부분에서 [Only show offerings that reserve capacity]를 선택합니다.
6. 필요에 따라 다른 구성을 선택하고 [Search]를 선택합니다.
7. 구입할 전환형 예약 인스턴스를 선택하고 수량을 입력한 후 [Add to Cart]를 선택합니다.
8. 선택한 내역을 보려면 [View Cart]를 선택합니다.
9. 주문을 완료하려면 [Purchase]를 선택합니다.

Note

구매 당시 선택한 조건과 비슷한데 가격은 더 낮은 상품이 있을 경우 AWS는 더 저렴한 상품을 판매 합니다.

지정한 리전에서 일치하는 사양을 가진 해당 온디맨드 인스턴스에 요금 혜택이 자동으로 적용됩니다. AWS에서는 할인된 시간당 요금을 자동으로 적용합니다. 인스턴스를 따로 재시작할 필요가 없습니다.

AWS Management Console을 사용하여 거래 상태를 보려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. [Reserved Instances] 페이지를 선택합니다. 구매 상태가 [State] 열에 표시됩니다. 주문이 완료되면 [State] 값이 `payment-pending`에서 `active`로 바뀝니다.

명령줄 인터페이스나 API를 사용하여 예약 인스턴스 구매

명령줄 인터페이스나 API를 사용하여 예약 인스턴스를 구매하려면

1. AWS CLI를 사용할 경우 [purchase-reserved-instances-offering](#)을 참조하십시오.
2. Windows PowerShell용 AWS 도구를 사용할 경우 [New-EC2ReservedInstance](#)를 참조하십시오.
3. Amazon EC2 API를 사용할 경우 [PurchaseReservedInstancesOffering](#)을 참조하십시오.

명령줄이나 API를 사용하여 거래 상태를 보려면

1. AWS CLI를 사용할 경우 [describe-reserved-instances](#)를 참조하십시오.
2. Windows PowerShell용 AWS 도구를 사용할 경우 [Get-EC2ReservedInstance](#)를 참조하십시오.
3. Amazon EC2 API를 사용할 경우 [DescribeReservedInstances](#)를 참조하십시오.

예약 인스턴스 적용

사양이 일치할 경우 실행 중인 온디맨드 인스턴스에 예약 인스턴스가 자동으로 적용됩니다. AWS Management Console, 명령줄 도구 또는 Amazon EC2 API를 사용하여 다음 작업을 수행할 수 있습니다.

Note

예약 인스턴스를 구매하고 수정하려면 가용 영역을 설명할 수 있는 권한과 같은 적절한 권한이 IAM 사용자 계정에 있는지 확인하십시오. 자세한 내용은 [AWS CLI 또는 AWS SDK 작업을 위한 예제 정책](#) 및 [Amazon EC2 콘솔 작업을 위한 예제 정책](#)을 참조하십시오.

구매 - 예약 용량을 결정합니다. 다음 기준을 지정합니다.

- 플랫폼(예: Linux)

Note

특정 플랫폼(예: Windows, Linux/Unix)에서 예약 인스턴스를 사용하려면 예약 용량을 구매할 때 반드시 해당 플랫폼을 지정해야 합니다. 그리고 해당 용량을 사용하여 인스턴스를 시작할 때 반드

시 구매 시에 지정한 플랫폼과 그 외 규격이 동일하게 설정된 Amazon 머신 이미지(AMI)를 선택해야 합니다.

- 인스턴스 유형(예: m1.small)
- 예약 범위(Region 또는 Availability Zone).
- 용량을 예약할 기간
- 테넌시. 단일 테넌트 하드웨어에서의 인스턴스 실행을 위해 용량을 예약할 수 있습니다(shared가 아닌 dedicated 테넌시). 선택한 테넌시는 예약 인스턴스를 적용 중이거나 적용할 계획인 온디맨드 인스턴스의 테넌시와 일치해야 합니다. 자세한 내용은 [전용 인스턴스 \(p. 257\)](#) 섹션을 참조하십시오.
- 제공 클래스(Standard 또는 Convertible).
- 제공 유형(선결제 없음, 부분 선결제, 전체 선결제).

사용 — 예약 인스턴스를 사용하려면 구입한 예약과 동일한 사양의 온디맨드 인스턴스를 시작합니다. 조건을 충족하는 모든 보유 인스턴스에 요금 혜택과 용량 예약이 자동으로 적용됩니다(이미 다른 예약을 사용하는 인스턴스는 제외).

자세한 내용은 [인스턴스 시작 \(p. 264\)](#) 섹션을 참조하십시오.

예약 인스턴스 상태

예약 인스턴스의 상태는 다음 중 한 가지로 표시됩니다.

- [active] - 예약 인스턴스를 사용할 수 있습니다.
- [payment-pending] - AWS에서 예약 인스턴스에 대한 결제를 처리 중입니다. 상태가 [active]로 변경되면 예약 인스턴스를 사용할 수 있습니다.
- [retired] - 다음과 같은 이유로 예약 인스턴스가 종료되었습니다.
 - AWS 결제가 이루어지지 않았습니다. 예: 신용 카드 거래가 승인되지 않았습니다.
 - 예약 인스턴스의 약정이 끝난 경우입니다.

[Reserved Instance] 페이지의 [State]에 표시되는 상태 정보는 [My Listings] 탭의 [Listing State]에 표시되는 상태 정보와는 다릅니다.

예약 인스턴스 마켓플레이스의 판매자인 경우 [Listing State]에 예약 인스턴스 마켓플레이스에 나열된 예약 상태가 표시됩니다. 자세한 내용은 [예약 인스턴스 판매 상태 \(p. 191\)](#) 섹션을 참조하십시오.

예약 인스턴스 마켓플레이스에서 구입

Note

전환형 예약 인스턴스는 예약 인스턴스 마켓플레이스에서 구입할 수 없습니다.

AWS로부터 Amazon EC2 예약 인스턴스를 구매하거나, 소유하고 있는 예약 인스턴스가 더 이상 필요하지 않은 타사 판매자로부터 구매할 수 있습니다.

구매자의 경우 예약 인스턴스 마켓플레이스를 통해 선택의 폭과 유연성을 높일 수 있습니다. 원하는 인스턴스 유형, 리전 및 기간에 가장 적합한 예약 인스턴스를 찾아볼 수 있습니다.

예약 인스턴스 마켓플레이스에 대한 자세한 내용은 [예약 인스턴스 마켓플레이스 판매 \(p. 187\)](#)를 참조하십시오.

예약 인스턴스 마켓플레이스에서 구매하는 예약 인스턴스와 AWS에서 직접 구매하는 예약 인스턴스 간에는 다음과 같은 몇 가지 차이점이 있습니다.

- 기간 - 제3자로부터 구매하는 예약 인스턴스는 표준 약정 기간보다 남은 기간이 짧습니다. AWS의 표준 약정 기간은 1년 또는 3년입니다.

- 선결제 요금- 제3자 예약 인스턴스는 다양한 선결제 요금으로 판매될 수 있습니다. 사용 요금이나 기본 요금은 AWS에서 예약 인스턴스를 처음에 구매했을 때 설정된 요금과 동일하게 유지됩니다.

귀하에 대한 기본 정보(우편번호 및 국가 정보 등)는 판매자와 공유됩니다.

이 정보는 판매자가 정부에 납부해야 하는 거래세(판매세, 부가가치세 등)을 계산하는 데 필요하며, 지급 내역서 형태로 제공됩니다. 드문 경우지만 판매자가 거래와 관련하여 문의할 수 있도록(세금 관련 질문 등) AWS에서 판매자에게 구매자의 이메일 주소를 제공할 수 있습니다.

또한 AWS에서 구매자에게 제공하는 구매 인보이스에는 판매자의 법인 이름이 표기됩니다. 세금이나 관련 이유로 인해 판매자에 대한 추가 정보가 필요할 경우 [AWS Support](#)로 문의하십시오.

예약 인스턴스 마켓플레이스 판매

Note

전환형 예약 인스턴스는 예약 인스턴스 마켓플레이스에서 볼 수 없습니다.

미사용 예약을 예약 인스턴스 마켓플레이스에서 판매함으로써, 비즈니스 요구 사항이 변했을 때 또는 필요 없는 용량이 있을 때 새로운 구성으로 전환할 수 있습니다.

예약 인스턴스 마켓플레이스에 예약 인스턴스를 등록하자마자 잠재적 구매자들에게 노출되어 판매가 가능합니다. 모든 예약 인스턴스는 남은 약정 기간 및 시간당 요금에 따라 분류됩니다.

구매자의 요청을 처리할 때 AWS는 특정 그룹에서 선결제 금액이 가장 낮은 예약 인스턴스부터 판매합니다. 그런 다음 구매자의 주문이 모두 처리될 때까지 낮은 가격부터 순차적으로 예약 인스턴스를 판매합니다. 그 다음 AWS는 이 거래를 처리하고 해당 예약 인스턴스의 소유권을 구매자에게 이전합니다.

예약 인스턴스가 판매되기 전까지는 판매자에게 소유권이 있습니다. 판매 후에는 용량 예약(예약 인스턴스를 가용 영역을 위해 구입한 경우)과 할인 기본 요금이 구매자에게 양도됩니다. 인스턴스를 계속 사용할 경우 AWS는 해당 예약 인스턴스가 판매된 시점부터 온디맨드 요금을 부과합니다.

다음은 알아 두어야 할 중요한 제한 사항입니다.

- 예약 인스턴스는 30일 후에 판매할 수 있음 - 예약 인스턴스는 소유한지 적어도 30일이 지나야 판매할 수 있습니다. 또한, 판매 등록하려는 예약 인스턴스의 남은 사용 기간이 한 달 이상이어야 합니다.
- 전환형 인스턴스 범위 - 예약 인스턴스 마켓플레이스에서는 용량이 예약된 표준 예약 인스턴스만 판매할 수 있습니다. 리전 혜택이 있는 예약 인스턴스는 판매할 수 없습니다.
- 등록한 상품은 수정할 수 없음 - 예약 인스턴스 마켓플레이스에 등록한 상품은 수정할 수 없습니다. 하지만 판매 등록을 취소하고 새 파라미터를 지정한 다음 다시 등록하는 방식으로 변경하는 것은 가능합니다. 자세한 내용은 [예약 인스턴스 판매 등록 \(p. 189\)](#) 섹션을 참조하십시오. 판매 등록하기 전에 예약 인스턴스를 수정할 수도 있습니다. 자세한 내용은 [표준 예약 인스턴스 변경 \(p. 193\)](#) 섹션을 참조하십시오.
- 할인된 예약 인스턴스는 판매할 수 없음 - 예약 인스턴스 중 할인 티어로 더 낮은 가격에 구매한 인스턴스는 예약 인스턴스 마켓플레이스에서 판매할 수 없습니다. 자세한 내용은 [예약 인스턴스 마켓플레이스 \(p. 177\)](#) 섹션을 참조하십시오.

목차

- [판매자 등록 \(p. 187\)](#)
- [예약 인스턴스 판매 등록 \(p. 189\)](#)
- [판매 기간 \(p. 192\)](#)
- [예약 인스턴스 판매 후 절차 \(p. 192\)](#)

판매자 등록

예약 인스턴스 마켓플레이스에서 인스턴스를 판매하려면 먼저 판매자 등록이 필요합니다. 등록 시 상호명과 은행 정보, 사업자 등록 번호를 제공합니다.

AWS에서 판매자 등록에 필요한 과정을 모두 마치면 등록 확인과 함께 예약 인스턴스 마켓플레이스에서 판매를 시작할 수 있음을 알리는 이메일이 발송됩니다.

항목

- [은행 계좌 \(p. 188\)](#)
- [세금 정보 \(p. 188\)](#)
- [구매자와의 정보 공유 \(p. 189\)](#)
- [판매 대금 정산 \(p. 189\)](#)

은행 계좌

AWS에서 예약 인스턴스의 판매 대금을 지불하기 위해서는 사용자의 은행 정보가 필요합니다. 이때 미국 소재지가 있는 은행을 선택해야 합니다.

지급금을 받을 기본 은행 계좌를 등록하려면

1. [예약 인스턴스 마켓플레이스 판매자 등록 페이지](#)에서 로그인합니다. AWS 계정이 없을 경우 이 페이지에서 계정을 만들 수 있습니다.
2. [Manage Bank Account] 페이지에서 판매 대금을 지급 받을 은행의 다음 정보를 입력합니다.

- 은행 계좌 소유자 이름
- 송금 번호
- 계좌 번호
- 은행 계좌 유형

Note

법인 계좌를 사용할 경우 은행 계좌를 팩스(1-206-765-3424)로 보내라는 메시지가 표시됩니다.

등록되면 이 은행 계좌가 기본 계좌로 설정되고 은행 확인은 보류 상태가 됩니다. 새로운 은행 계좌를 확인하려면 최대 2주 정도 걸리며 이 기간 동안에는 입금을 받을 수 없습니다. 검증된 계좌는 대금 입금이 완료되는 데 보통 2일 정도 걸립니다.

지급금을 받을 기본 은행 계좌를 변경하려면

1. [예약 인스턴스 마켓플레이스 판매자 등록 페이지](#)에서 등록 시 사용한 계정으로 로그인합니다.
2. [Manage Bank Account] 페이지에서 필요에 따라 새로운 은행 계좌를 추가하거나 기본 은행 계좌를 수정합니다.

세금 정보

예약 인스턴스를 판매할 때 판매세나 부가가치세 등 거래세가 발생할 수 있습니다. 거래세의 적용 여부는 회사 내부의 세금, 법무, 회계 부서 등 관련 부서에 문의하여 확인하십시오. 거래에 관련된 세금을 정산하고 관련 부처에 납부할 책임은 사용자에게 있습니다.

판매자 등록 과정에서는 세금 신고서 옵션이 제공됩니다. 다음 조건에 해당되는 경우, 인터뷰 과정을 수행하는 것을 추천합니다.

- AWS에서 Form 1099-K를 작성하려는 경우.
- 계획한 거래 수가 50건 이상인 경우 또는 1년(연년 기준) 내에 20,000 USD 이상의 예약 인스턴스를 판매하려는 경우. 거래당 판매 예약 인스턴스는 하나 이상일 수 있습니다. 등록 시 이 단계를 건너뛰기로 선택한 경우, 이후에 거래 수가 49건에 도달하면 "You have reached the transaction limit for pre-tax. [판매자 등록 포털](#)에서 세금 신고서를 작성해 주십시오."라는 메시지가 표시됩니다. 세금 신고서를 작성하면 계정 한도가 자동으로 증가합니다.

- 미국 내 판매자가 아닌 경우, 이 경우에는 반드시 전자상으로 Form W-8BEN을 작성해야 합니다.

IRS 세금 신고 규정과 Form 1099-K에 대한 자세한 내용은 [IRS 웹 사이트](#) 섹션을 참조하십시오.

세금 신고서 작성 시 입력하는 세금 정보는 미국 법인인지 아니면 미국 외 법인인지에 따라 다릅니다. 세금 신고서를 작성할 때는 다음을 참고하십시오.

- 이 주제를 비롯해 AWS에서 제공하는 정보는 세금과 법률 그 외 분야에 대한 전문 조언이 아닙니다. IRS 세금 신고 규정이 기업에 미칠 수 있는 영향이나 다른 의문점은 세금, 법률, 기타 분야의 전문가에게 상담 하십시오.
- IRS 세금 신고 규정을 가장 효율적으로 준수할 수 있는 방법은 인터뷰에 나오는 모든 질문에 답변하고 요청된 모든 정보를 제공하는 것입니다.
- 답변을 확인하십시오. 오타나 사업자 등록 번호가 잘못 기재되지 않도록 유의해야 합니다. 이에 따라 세금 신고서를 다시 작성해야 할 수 있습니다.

세금 등록 과정을 완료하면 AWS에서 1099-K 양식을 생성합니다. 판매자는 세금 계정이 기준선을 초과한 년도의 다음 해 1월 31일 또는 그 이전에 우편으로 송부된 파일을 받게 됩니다. 예를 들어 세금 계정이 2016년에 한계에 도달하면 2017년에 양식을 받게 됩니다.

구매자와의 정보 공유

예약 인스턴스 마켓플레이스에서 판매할 경우 AWS는 미국 규정에 따라 구매자 명세서에 판매자의 상호명을 기재하여 제공합니다. 그 외에 구매자가 인보이스 또는 다른 세금 관련 이유로 AWS Support에 요청한 경우, AWS에서 구매자가 직접 연락을 취할 수 있도록 판매자의 이메일 주소를 제공할 수 있습니다.

이와 비슷한 이유로 판매자의 지불 내역서에는 구매자의 지역번호(우편번호)와 국가 정보가 제공됩니다. 이 정보는 판매자 측에서 거래에 따라 정부에 납부해야 하는 세금(예: 매출세, 부가가치세)이 발생하는 경우, 이런 세금을 정산하는 데 필요합니다.

AWS에서는 세금에 대해 조언하지 않습니다. 단, 회사의 세금 전문 담당자가 특정 정보를 추가로 요청한 경우에는 [AWS Support](#)에 문의하십시오.

판매 대금 정산

AWS는 구매자가 결제를 완료하자마자 판매된 해당 예약 인스턴스의 소유자로 등록된 계정 이메일 주소로 메시지를 보내 이를 알립니다.

AWS는 ACH(자동 결제) 시스템을 통해 지정된 은행 계좌로 송금합니다. 일반적인 송금 시기는 예약 인스턴스가 판매된 후 1일에서 3일 사이입니다. 지불 상태는 예약 인스턴스 지불 명세서를 조회하여 확인할 수 있습니다. 지불은 매일 한 번 실시됩니다. AWS에서 은행으로부터 계좌를 확인받기 전에는 대금이 지불되지 않으므로 이 점에 유의하십시오. 이 절차는 최대 이주가 소요됩니다.

판매된 예약 인스턴스는 `DescribeReservedInstances` 호출 결과에 계속 표시됩니다.

예약 인스턴스를 판매한 대금은 현금으로 지급되며, 판매자 명의의 은행 계좌로 직접 송금됩니다. AWS는 예약 인스턴스 마켓플레이스에서 판매하는 각 예약 인스턴스에 대해 총 선결제 금액의 12%를 서비스 수수료로 청구합니다.

Note

예약 인스턴스 마켓플레이스에서는 Amazon EC2 예약 인스턴스만 판매할 수 있습니다. Amazon RDS 및 Amazon ElastiCache 예약 인스턴스와 같은 다른 유형은 예약 인스턴스 마켓플레이스에서 판매할 수 없습니다.

예약 인스턴스 판매 등록

등록된 판매자는 하나 이상의 예약 인스턴스를 판매하거나, 모든 인스턴스를 한 번의 판매 등록으로 또는 부분적으로 판매할 수 있습니다. 또한 인스턴스 유형, 플랫폼, 리전, 가용 영역 등을 비롯한 다양한 유형의 인스턴스를 판매 등록할 수 있습니다.

판매 등록을 취소하려는 경우 인스턴스 중 일부가 이미 판매되었다면, 이미 판매된 인스턴스에 대해서는 취소가 적용되지 않습니다. 이 때는 아직 판매되지 않은 인스턴스만 예약 인스턴스 마켓플레이스 판매 목록에서 삭제됩니다.

예약 인스턴스의 가격 책정

판매자는 판매할 예약 인스턴스에 대한 선결제 금액만 책정할 수 있습니다. 선결제 금액은 구매자가 예약 인스턴스를 구매할 때 지불하는 일회성 요금입니다. 판매자는 사용 요금이나 기본 요금을 지정할 수 없습니다. 구매자는 판매자가 처음에 예약을 구매할 때 책정되었던 사용 요금이나 기본 요금과 동일한 요금을 지불해야 합니다.

다음은 알아 두어야 할 중요한 제한 사항입니다.

- 연간 최대 50,000 USD의 예약 인스턴스를 판매할 수 있습니다. 그 이상의 예약 인스턴스를 판매하려는 경우, [Request to Raise Sales Limit on Amazon EC2](#) 양식을 작성해야 합니다.
- 최소 요금은 0 USD입니다. 예약 인스턴스 마켓플레이스에서 허용되는 최소 허용 판매가는 0.00 USD입니다.

판매 등록을 직접 변경할 수는 없습니다. 하지만 판매 등록을 취소하고 새 파라미터를 지정한 다음 다시 등록하는 방식으로 변경하는 것은 가능합니다.

현재 active(활성) 상태가 아닌 항목에 한해 언제든지 판매 등록을 취소할 수 있습니다. 구매자의 검색 결과에 일치하는 항목으로 선정되어 이미 판매 처리 중인 항목은 취소할 수 없습니다. 판매 등록을 취소한 시점에서 이 등록에 속하는 일부 예약 인스턴스가 이미 판매 선정되었다면, 선정된 인스턴스를 제외한 인스턴스만 판매 등록이 취소됩니다.

가격표 설정

기본적으로 예약 인스턴스의 가격은 시간이 지날수록 떨어지므로 AWS는 매달 일정 금액씩 가격이 내려가도록 가격을 설정할 수 있습니다. 하지만 판매자는 예약 판매 시점을 기준으로 선결제 가격을 다르게 설정할 수 있습니다.

예를 들어 사용 기간이 9개월 남은 예약 인스턴스를 판매하는 경우, 9개월이라는 기간이 남아 있는 동안 이 예약 인스턴스를 구매하는 구매자에게 밸류 금액을 설정할 수 있습니다. 남은 기간이 5개월인 시점과 1개월 인 시점에서의 판매 가격을 각각 책정할 수 있습니다.

AWS CLI를 사용하여 예약 인스턴스 판매 등록

AWS CLI를 사용하여 예약 인스턴스 마켓플레이스에서 예약 인스턴스를 판매 등록하려면

1. `aws ec2 describe-reserved-instances` 호출을 실행하여 예약 인스턴스의 목록을 표시합니다.
2. 판매 등록할 예약 인스턴스의 ID를 지정한 다음 `aws ec2 create-reserved-instances-listing`을 호출합니다. 필수적으로 지정해야 하는 파라미터는 다음과 같습니다.
 - Reserved Instance ID(예약 인스턴스 ID)
 - Instance count(인스턴스 개수)
 - MONTH:PRICE(개월:가격)

등록 상품 확인

- `aws ec2 describe-reserved-instances-listings` 명령을 사용하여 등록 상품에 대한 세부 정보를 확인합니다.

판매 등록을 취소 및 변경하려면

- `aws ec2 cancel-reserved-instances-listings` 명령을 사용하여 등록 상품에 대한 세부 정보를 확인합니다.

Amazon EC2 API를 사용하여 예약 인스턴스 판매 등록

Amazon EC2 API를 사용하여 예약 인스턴스 마켓플레이스에서 예약 인스턴스를 판매 등록하려면

1. `DescribeReservedInstances` 호출을 실행하여 예약 인스턴스의 목록을 표시합니다. 예약 인스턴스 마켓플레이스에서 등록한 예약 인스턴스의 ID를 기록해둡니다.
2. `CreateReservedInstancesListing`을 사용하여 판매 등록을 생성합니다.

등록 상품 확인

1. `DescribeReservedInstancesListings`을 호출하면 등록 상품에 대한 세부 정보를 확인할 수 있습니다.

판매 등록 취소

1. 실행 `CancelReservedInstancesListing`.
2. `DescribeReservedInstancesListings`을 호출하여 취소를 확정합니다.

AWS Management Console을 사용하여 예약 인스턴스 판매 등록

AWS Management Console을 사용하여 예약 인스턴스 마켓플레이스에서 예약 인스턴스를 판매 등록하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Reserved Instances]를 선택합니다.
3. 판매 등록할 예약 인스턴스를 선택하고 [Sell Reserved Instances]를 선택합니다.
4. [Configure Your Reserved Instance Listing] 페이지에서 판매할 인스턴스의 수와, 남은 사용 기간에 대한 선결제 금액을 해당 열에 설정합니다. [Months Remaining] 열 옆의 화살표를 선택하여 남은 사용 기간에 따라 예약 가격이 어떻게 변경되는지 확인해 보십시오.
5. 절차에 익숙한 고급 사용자가 따로 가격 책정을 원하는 경우, 개월 수에 따라 각각 다른 금액을 설정할 수 있습니다. 일정 금액씩 하락되는 기본 설정으로 돌아가려면 [Reset]을 선택합니다.
6. 판매 등록 구성을 마쳤으면 [Continue]를 선택합니다.
7. [Confirm Your Reserved Instance Listing] 페이지에 표시된 세부 정보를 확인하고 그대로 진행하려면 [List Reserved Instance]를 선택합니다.

콘솔에서 등록 상품을 보려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Reserved Instances]를 선택합니다.
3. 판매 등록한 예약 인스턴스를 선택하고 [My Listings]를 선택합니다.

예약 인스턴스 판매 상태

[Listing State]에는 판매 등록의 현재 상태가 표시됩니다.

[Listing State]에서 표시되는 정보는 예약 인스턴스 마켓플레이스에 판매 등록된 항목의 상태입니다. 이 상태 정보는 [Reserved Instances] 페이지의 [State] 열에 표시되는 상태 정보와는 다릅니다. 이 [State] 정보는 보유한 예약의 상태입니다.

- active - 구매 가능한 항목입니다.
- cancelled — 판매 등록이 취소되어 예약 인스턴스 마켓플레이스에서 구매할 수 없습니다.
- [closed] - 판매 등록되지 않은 예약 인스턴스입니다. 항목 판매가 완료된 예약 인스턴스의 경우에도 상태가 [closed]로 표시됩니다.

자세한 내용은 [예약 인스턴스 상태 \(p. 186\)](#) 섹션을 참조하십시오.

판매 기간

판매 등록을 마쳤다면 다음은 등록된 상품의 판매 시 진행되는 과정을 알아볼 차례입니다.

등록된 항목의 모든 인스턴스가 판매 완료된 경우, [My Listings] 탭의 [Total instance count]의 값이 [Sold] 항목의 값과 동일합니다. 또한 Available 인스턴스가 더 이상 존재하지 않는 것을 확인할 수 있습니다. Status 항목은 `closed`로 표시됩니다.

항목 중 일부 인스턴스만 판매된 경우, AWS에서 이 등록 항목에서 판매된 예약 인스턴스를 빼고 판매되지 않은 인스턴스와 동일한 개수의 새 인스턴스를 생성합니다. 따라서 판매 등록 ID와 해당 판매 등록은 활성 상태로 유지되지만, 남은 예약 인스턴스 수는 줄어듭니다.

이후 예약 인스턴스가 판매될 때 마다 같은 절차가 반복됩니다. 모든 예약 인스턴스가 판매되면 AWS에서 해당 등록 항목이 `closed`로 변경됩니다.

예를 들어 Reserved Instances listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample 항목으로 5개의 인스턴스를 판매 등록했다고 가정해 보겠습니다.

이때 콘솔의 [Reserved Instance] 페이지를 열었을 때 [My Listings] 탭에 다음 정보가 표시됩니다.

Reserved Instance listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample

- 전체 예약 인스턴스 개수 = 5
- Sold = 0
- Available = 5
- Status = active

구매자가 이 중 2개의 인스턴스를 구매했다면 이제 판매 가능한 인스턴스 수는 이제 3개가 됩니다. AWS에서는 이 부분 판매에 따라 인스턴스 개수가 세 개인 새로운 예약을 생성하며, 이 인스턴스 개수는 아직 판매 중인 인스턴스를 의미합니다.

새롭게 변경된 정보는 [My Listings] 탭에 다음과 같이 나타납니다.

Reserved Instance listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample

- 전체 예약 인스턴스 개수 = 5
- Sold = 2
- Available = 3
- Status = active

판매 등록 항목을 취소할 때 항목의 인스턴스 중 일부가 이미 판매되었다면, 이미 판매된 인스턴스에 대해서는 취소가 적용되지 않습니다. 이 때는 아직 판매되지 않은 인스턴스만 예약 인스턴스 마켓플레이스 판매 목록에서 삭제됩니다.

예약 인스턴스 판매 후 절차

예약 인스턴스가 판매되면 AWS에서 이메일로 이를 알립니다. 어떤 활동이 발생하면 당일에 발생한 모든 활동 내역이 이메일로 발송됩니다. 판매를 등록하거나, 등록 상품을 판매하거나, AWS에서 대금 송금하는 활동 등이 있을 수 있습니다.

콘솔에서 판매 등록된 예약 인스턴스의 상태를 조회하려면 [Reserved Instance], [My Listings]를 선택합니다. [My Listings] 탭에는 [Listing State] 값이 표시됩니다. 또한 사용 기간, 판매 가격, 등록 항목에서 Available(판매 가능), Pending(보류), Sold(판매), Cancelled(취소) 상태의 인스턴스 개수 정보도 제공됩니다. 또한 `ec2-describe-reserved-instances-listings` CLI 명령이나 `DescribeReservedInstancesListings` API 호출을 통해 필터를 사용하여 판매 등록에 대한 정보를 알아볼 수 있습니다.

표준 예약 인스턴스 변경

컴퓨팅 요구사항에 변화가 생긴 경우, 표준 예약 인스턴스를 변경함으로써 요금 혜택에 따른 혜택을 계속 유지할 수 있습니다. 전환형 예약 인스턴스는 교환 프로세스를 사용하여 조정할 수 있습니다. 자세한 내용은 [전환형 예약 인스턴스 교환 \(p. 198\)](#) 섹션을 참조하십시오.

표준 예약 인스턴스의 변경 절차에 대한 자세한 내용은 다음 주제를 참조하십시오.

항목

- [변경 조건 \(p. 193\)](#)
- [예약 내역의 인스턴스 크기 수정 \(p. 194\)](#)
- [변경 요청 제출 \(p. 195\)](#)

표준 예약 인스턴스를 변경해도 약정 기간에는 영향을 미치지 않으며, 종료일도 동일하게 유지됩니다. 변경 비용이 없기 때문에 별도로 청구서나 인보이스를 수신하지 않습니다. 변경은 구매와는 별도의 작업이며, 표준 예약 인스턴스의 사용이나 구매, 판매에 영향을 주지 않습니다. 다음 방법을 사용하여 예약한 인스턴스를 전체적으로 변경하거나 하위 그룹만 선택적으로 변경할 수 있습니다.

- 같은 리전 내 다른 가용 영역으로 변경
- 가용 영역에서 리전으로 예약 범위 변경(반대 방향도 마찬가지)
- EC2-VPC와 EC2-Classic 간에 전환
- 같은 인스턴스 유형 내에서 다른 인스턴스 크기로 변경

가용 영역, 범위 및 네트워크 플랫폼은 모든 플랫폼 유형(Linux 및 Windows)에서 변경을 지원합니다. 인스턴스 유형 변경은 Linux 플랫폼 유형에서만 지원됩니다. 하지만 라이선싱 차이로 인해 RedHat 또는 SUSE Linux 표준 예약 인스턴스의 인스턴스 유형 또는 크기는 변경할 수 없습니다. RedHat 및 SUSE 요금에 대한 자세한 내용은 [Amazon EC2 예약 인스턴스 요금](#)을 참조하십시오.

예약의 가용 영역을 변경할 경우 용량 예약 및 요금 혜택이 새로운 가용 영역의 인스턴스 사용에 자동으로 적용됩니다. 예약 인스턴스의 네트워크 플랫폼을 변경할 경우(예: EC2-Classic에서 EC2-VPC로 변경) 새로운 네트워크의 인스턴스 사용에 용량 예약이 자동으로 적용됩니다.

예약 범위를 가용 영역에서 리전으로 변경하면 가용 영역 유연성과 인스턴스 크기 유연성을 위한 용량 예약은 사라집니다. 이러한 가용 영역의 유연성은 단일 리전에 속한 모든 가용 영역의 인스턴스 사용량에 대해 예약 인스턴스의 할인 혜택을 제공합니다. 인스턴스 크기 유연성은 해당 인스턴스 패밀리 내에서 크기에 상관 없이 인스턴스 사용량에 대해 예약 인스턴스의 할인 혜택을 제공합니다.

Note

인스턴스 크기 유연성은 기본 테넌시를 포함하여 리전에 할당되는 Linux/Unix 기반 예약 인스턴스에서만 지원됩니다. 예약의 요금 혜택은 해당 리전의 모든 해당 인스턴스에 적용됩니다.

변경 후 예약 인스턴스의 요금 혜택은 새로운 파라미터와 일치하는 인스턴스에만 적용됩니다. 새 파라미터와 일치하지 않는 인스턴스는 계정의 다른 예약 내역 할인이 적용되지 않는 한 온디맨드 요금이 부과됩니다. 요금 혜택은 예약 사양이 일치하는 EC2-Classic 인스턴스와 EC2-VPC 인스턴스에 모두 적용됩니다.

변경 조건

Amazon EC2에서는 요청된 구성에 사용할 수 있는 용량이 충분히 남아 있고(해당되는 경우) 다음 조건을 충족하는 경우 변경 요청을 처리합니다.

변경 예약 인스턴스 조건:

- 활성화
- 보류 중인 다른 변경 요청의 대상이 아닌 인스턴스

- 예약 인스턴스 마켓플레이스에 등록되지 않음
- 종료 시간이 동일한 인스턴스(분이나 초가 아닌 시간 기준)

변경 요청 조건:

- 범위, 인스턴스 유형, 인스턴스 크기, 제공 클래스 및 네트워크 플랫폼 속성의 고유 조합
- 활성 예약 상의 인스턴스 공간 크기와 변경 후 규격의 인스턴스 공간 크기가 동일한 요청

제한

- 표준 예약 인스턴스만 수정할 수 있습니다.

예약 인스턴스가 활성 상태가 아니거나 변경이 불가능한 경우, AWS Management Console의 [Modify Reserved Instances] 버튼이 활성화되지 않습니다. 인스턴스 유형 변경을 허용하지 않는 플랫폼에 대해 변경 대상 예약 인스턴스가 하나 이상 존재하는 경우, [Modify Reserved Instances] 페이지에서는 선택한 모든 예약 인스턴스에 대한 인스턴스 유형 변경 옵션이 비활성화됩니다. 자세한 내용은 [예약 내역의 인스턴스 크기 설정 \(p. 194\)](#) 섹션을 참조하십시오.

예약은 제한 없이 원하는 만큼 변경이 가능하지만, 아직 보류 중인 이전 변경 요청에서 선택했던 예약에 대해서는 변경 요청을 제출할 수 없습니다. 또한, 제출 후 보류 상태인 변경은 다시 변경하거나 취소할 수 없습니다. 변경이 성공적으로 처리된 후에는 변경 전 상태로 되돌리기 위해 또 다른 변경 요청을 제출할 수 있습니다. 자세한 내용은 [변경 처리 상태 확인 \(p. 197\)](#) 섹션을 참조하십시오.

예약 인스턴스 마켓플레이스에 판매 등록된 예약 인스턴스를 변경하려면 등록을 취소하고 변경을 요청한 뒤 다시 등록해야 합니다. 추가로, 등록 상품은 구매 전이나 구매 당시에는 변경할 수 없습니다. 자세한 내용은 [예약 인스턴스 마켓플레이스 \(p. 177\)](#) 섹션을 참조하십시오.

예약 내역의 인스턴스 크기 수정

다양한 크기의 인스턴스 유형에 Amazon Linux 등일 시작 인스턴스가 있을 경우 스텠다드 예약 인스턴스의 인스턴스 크기를 조정할 수 있습니다. 인스턴스 크기 수정은 리전, 사용 유형, 테넌시, 플랫폼, 종료 일시 등과 같은 기타 속성이 일치하고 가용 용량이 있는 경우에만 허용된다는 것에 유의하십시오. Windows 예약 인스턴스의 인스턴스 크기를 수정할 수 없습니다.

Note

인스턴스들은 패밀리(스토리지 또는 CPU 용량 기준), 유형(특정 사용 사례에 맞춘 설계), 그리고 크기에 따라 그룹이 나뉩니다. 예를 들어 c4 인스턴스 유형은 컴퓨팅 최적화 인스턴스 패밀리에 속하고 다양한 크기로 사용 가능합니다. c3 인스턴스들은 같은 패밀리에 속하지만, c4 인스턴스들은 하드웨어 사양이 다르기 때문에 c3 인스턴스로 수정할 수 없습니다. 자세한 내용은 [Amazon EC2 인스턴스 유형](#) 섹션을 참조하십시오.

수정 프로세스와 단계에 대한 자세한 내용은 [변경 요청 제출 \(p. 195\)](#) 섹션을 참조하십시오.

다음 인스턴스들은 사용 가능한 다른 크기가 존재하지 않으므로 변경할 수 없습니다.

- t1.micro
- cc1.4xlarge
- cc2.8xlarge
- cg1.8xlarge
- cr1.8xlarge
- hi1.4xlarge
- hs1.8xlarge
- g2.2xlarge

사용 가능한 용량이 충분하고 변경 결과로 원래 예약된 인스턴스 공간 크기가 달라지지 않는다면, 요청이 성공합니다.

인스턴스 공간 크기

각 예약 인스턴스는 인스턴스 공간 크기를 가지며, 이 공간 크기는 인스턴스 유형의 정규화 인자와 예약된 인스턴스 개수에 따라 결정됩니다.

정규화 인자는 인스턴스 유형(예: m1 인스턴스 유형 내 m1.xlarge 인스턴스) 크기를 기준으로 합니다. 동일한 인스턴스 유형 내에서만 의미가 있습니다. 인스턴스 유형은 하나의 유형에서 다른 유형으로 변경할 수 없습니다. Amazon EC2 콘솔에서 정규화 인자는 유닛으로 표시됩니다. 다음 표는 하나의 인스턴스 유형 내에서 적용되는 정규화 인자를 설명합니다.

인스턴스 크기	정규화 인자
nano	0.25
micro	0.5
small	1
medium	2
large	4
xlarge	8
2xlarge	16
4xlarge	32
8xlarge	64
10xlarge	80
16xlarge	128
32xlarge	256

변경 구성의 공간 크기가 원래 구성의 크기와 일치하지 않으면 수정 요청이 처리되지 않습니다.

예약 인스턴스의 인스턴스 공간 크기는 정규화 인자에 인스턴스 수를 곱하여 산출합니다. 예를 들어, 하나의 m1.medium은 정규화 인자 2를 가지므로, m1.medium 인스턴스 4개의 공간은 8유닛입니다.

예약은 예약의 인스턴스 공간 크기가 변동되지 않는 선에서 동일한 인스턴스 유형 내의 다른 인스턴스 크기로 다양하게 할당할 수 있습니다. 예를 들어, 1개의 m1.large(1 x 4) 인스턴스에 대한 예약을 4개의 m1.small(4 x 1) 인스턴스로 나누거나 반대로 기존의 m1.small 인스턴스 4개를 합쳐 1개의 m1.large 인스턴스로 변경할 수 있습니다. 하지만 2개의 m1.small(2 x 1) 인스턴스를 1개의 m1.large(1 x 4) 인스턴스로 변경할 수는 없습니다. 그 이유는 변경했을 때의 인스턴스 공간 크기가 원래 예약에 따른 인스턴스 공간 크기보다 커지기 때문입니다.

자세한 내용은 [Amazon EC2 인스턴스 유형 섹션](#)을 참조하십시오.

변경 요청 제출

AWS에서는 여러 경로로 변경 요청을 확인하고 작업할 수 있는데, AWS Management Console을 사용하거나 Amazon EC2 API를 통해 직접 작업하는 방법, 그리고 명령줄 인터페이스를 사용하는 방법이 있습니다.

항목

- [AWS Management Console \(p. 196\)](#)

- 명령행 인터페이스 (p. 196)
- Amazon EC2 API (p. 196)
- 변경 처리 상태 확인 (p. 197)

AWS Management Console

[Modify Reserved Instances] 페이지의 각 대상 구성 행에는 현재 인스턴스 유형에 해당되는 인스턴스의 개수([Count])와 인스턴스 유형을 기준으로 한 예약의 상대적인 인스턴스 공간 크기([Units])가 표시됩니다. 자세한 내용은 [인스턴스 공간 크기 \(p. 195\)](#) 섹션을 참조하십시오.

변경 가능한 공간보다 적거나 많은 예약 인스턴스를 설정한 경우, 할당 합계(allocated total)가 빨간색으로 표시됩니다. 변경 가능한 모든 예약 인스턴스에 대해 변경 지정하면 합계가 녹색으로 표시되고, [Continue]를 선택할 수 있습니다.

예약 인스턴스를 부분적으로 변경하는 경우, Amazon EC2에서는 원래의 예약 인스턴스를 2개 이상의 새로운 예약 인스턴스 그룹으로 분리합니다. 예를 들어, us-east-1a에서 10개의 동일 시작 인스턴스를 보유하고 있다가 이중 5개의 인스턴스를 us-east-1b로 옮기는 경우, 해당 변경 요청에 따라 5개의 인스턴스가 소속된 us-east-1a(원래의 사용 영역) 그룹과 나머지 5개의 인스턴스가 소속된 us-east-1b 그룹, 이렇게 두 그룹의 동일 시작 인스턴스가 새로 생성됩니다.

AWS Management Console을 사용해 예약 인스턴스를 변경하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. [Reserved Instances] 페이지에서 변경할 예약 인스턴스를 하나 이상 선택하고 [Modify Reserved Instances]를 선택합니다.

Note

수정 표의 첫 항목은 수정되지 않은 원래의 동일 시작 인스턴스입니다. 모든 동일 시작 인스턴스의 속성을 수정하려면 메뉴에서 새로운 사양을 선택합니다. 동일 시작 인스턴스의 일부만 수정하거나 분할하려면 각각의 변경 사항에 대해 새 줄을 추가합니다.

3. 각 추가 속성 변경에 대해 [Add]를 누르고 수정하려는 예약의 수를 [Count]에 입력합니다.
 - 사용 영역을 변경하려면 [Availability Zone] 목록에서 값을 선택합니다.
 - 네트워크 플랫폼을 변경하려면 [Network] 목록에서 값을 선택합니다.
 - 인스턴스 유형을 변경하려면 [Instance Type] 목록에서 값을 선택합니다.
4. 설정한 속성 변경을 삭제하려면 해당 행의 [X]를 선택합니다.

Note

[Modify Reserved Instances] 페이지에 속성 변경 행이 하나만 있는 경우, 이 행은 삭제할 수 없습니다. 여러 예약 인스턴스 속성을 수정하려면 먼저 새로운 사양을 위한 행을 추가한 후 기존 행을 삭제합니다.

5. [Continue]를 선택합니다.
6. 원하는 대로 구성을 지정하고 변경 사항을 확정하려면 [Submit Modifications]를 선택합니다. 변경하지 않으려는 경우 언제든지 [Cancel]을 선택하여 마법사를 종료할 수 있습니다.

명령행 인터페이스

변경 작업은 AWS CLI([modify-reserved-instances](#)), Windows PowerShell용 AWS 도구([Edit-EC2ReservedInstance](#)), Amazon EC2 API([ModifyReservedInstances](#)) 및 AWS SDK for Java를 사용하여 프로그래밍 방식으로 완료할 수 있습니다.

Amazon EC2 API

[ModifyReservedInstances](#) 작업을 사용하여 예약 인스턴스를 변경할 수 있습니다. 자세한 내용은 [Amazon EC2 API 참조](#)를 참조하십시오.

변경 처리 상태 확인

변경 요청의 상태는 변경 대상 예약 인스턴스의 [state]를 확인하여 알 수 있습니다. 확인에 따라 표시되는 요청 상태는 `in-progress`, `fulfilled` 또는 `failed`입니다. 이 정보를 확인하려면 다음 리소스를 사용하십시오.

- AWS Management Console의 [State] 필드
- [DescribeReservedInstancesModifications](#) API 작업
- `describe-reserved-instances-modifications` AWS CLI 명령
- `Get-EC2ReservedInstancesModifications` Windows PowerShell용 AWS 도구 명령

다음 표에는 AWS Management Console에서 볼 수 있는 [State] 값에 대한 설명이 나와 있습니다.

시/도	설명
active (pending modification)	원래 예약 인스턴스에 임시로 적용되는 상태입니다.
retired (pending modification)	새 예약 인스턴스가 생성되는 동안 원래 예약 인스턴스에 임시로 적용되는 상태입니다.
retired	예약 인스턴스가 성공적으로 변경 및 교체되었습니다.
active	변경 요청 성공한 경우 생성된 새 예약 인스턴스의 상태입니다. 또는 변경 요청이 실패한 경우 본래 예약 인스턴스의 상태입니다.

Note

[DescribeReservedInstancesModifications](#) API 작업을 사용하는 경우, 변경 요청 상태는 `processing`, `fulfilled`, 또는 `failed`로 표시됩니다.

변경 요청이 성공한 경우:

- 변경된 예약이 즉시 적용되고 변경 요청 시점을 기준으로 새 인스턴스에 요금 혜택이 적용됩니다. 예를 들어, 예약 변경이 성공적으로 완료된 시간이 오후 9시 15분이라면, 요금 혜택은 오후 9시부터 새 인스턴스에 적용됩니다. (변경된 예약 인스턴스의 `effective date`(시작일)은 [DescribeReservedInstances](#) API 작업이나 `-describe-reserved-instances` 명령(AWS CLI)을 사용하여 확인할 수 있음).
- 본래 예약이 종료됩니다. 이 예약의 종료일은 새로운 예약의 시작일이 되며, 새 예약의 종료일은 본래 예약 인스턴스의 종료일과 동일합니다. 3년 약정 예약 중 16개월 남은 시점에서 변경했다면, 변경된 예약은 16개월 동안 사용이 가능하며 본래 예약의 종료일과 같은 날짜에 사용 기간이 만료됩니다.
- 변경된 예약의 고정 가격은 본래 예약의 고정 가격이 아닌 \$0로 표시됩니다.

Note

변경된 예약의 고정 가격은 계정에 적용되는 할인 요금 티어에는 영향을 주지 않습니다. 할인 요금 티어는 본래 예약의 고정 가격을 기준으로 하기 때문입니다.

변경 요청이 실패한 경우:

- 예약 인스턴스의 규격이 본래 설정대로 유지됩니다.
- 예약 인스턴스에 대한 또 다른 변경 요청이 즉시 가능합니다.

일부 예약 인스턴스를 변경할 수 없는 이유에 대한 자세한 내용은 [변경 조건 \(p. 193\)](#)을 참조하십시오.

전환형 예약 인스턴스 교환

전환형 예약 인스턴스는 인스턴스 패밀리를 비롯하여 구성이 다른 전환형 예약 인스턴스와 교환할 수 있습니다. 교환 횟수에 제한은 없습니다. 단, 해당 전환형 예약 인스턴스가 교환하려는 전환형 예약 인스턴스보다 가격이 높아야 합니다.

전환형 예약 인스턴스 교환 요구 사항

Amazon EC2에서는 다음 조건이 충족될 경우 교환 요청을 처리합니다.

전환형 예약 인스턴스가 다음 조건을 충족해야 합니다.

- 활성 상태
- 보류 중인 다른 교환 요청의 대상이 아닌 인스턴스
- 종료 시간이 동일한 인스턴스(분이나 초가 아닌 시간 기준)

제한:

- 전환형 예약 인스턴스는 AWS에서 현재 제공하는 다른 전환형 예약 인스턴스하고만 교환할 수 있습니다.
- 전환형 예약 인스턴스는 변경할 수 없습니다. 예약 구성을 변경하려면 다른 인스턴스와 교환해야 합니다.
- 전환형 예약 인스턴스는 결제 옵션이 동일하거나 더 비쌀 경우에만 교환할 수 있습니다. 예를 들어 부분 선 결제 전환형 예약 인스턴스는 전체 선결제 전환형 예약 인스턴스와 교환할 수 있습니다. 하지만 선결제가 없는 전환형 인스턴스와 교환할 수는 없습니다.

전환형 예약 인스턴스가 활성 상태가 아니거나 교환이 불가능한 경우, AWS Management Console의 [Exchange Reserved Instances] 버튼이 활성화되지 않습니다.

교환은 횟수에 제한은 없지만, 아직 보류 중인 이전 교환 요청에서 선택했던 예약에 대해서는 교환 요청을 제출할 수 없습니다.

전환형 예약 인스턴스 교환 계산

전환형 예약 인스턴스 교환은 무료입니다. 하지만 트루업(true-up) 비용을 지불해야 할 수 있습니다. 이 비용은 현재 소유했던 전환형 예약 인스턴스와, 교환을 통해 받는 전환형 예약 인스턴스 간의 차이를 비례 할당으로 계산한 선결제 비용입니다.

각 전환형 예약 인스턴스에는 정가가 있습니다. 교환의 결과로 받을 수 있는 동일 시작 인스턴스의 수를 결정하기 위해 이 정가를, 원하는 전환형 예약 인스턴스의 정가와 비교합니다.

정가가 \$35인 전환형 인스턴스 1개를 정가가 \$10인 새 인스턴스 유형과 교환하려는 경우를 예로 들어 보겠습니다.

\$35/\$10 = 3.5

이 경우 현재 가지고 있는 전환형 예약 인스턴스를, \$10의 전환형 예약 인스턴스 3개와 교환할 수 있습니다. 절반의 동일 시작 인스턴스를 구입할 수는 없으므로 전환형 예약 인스턴스를 추가로 구입하여 나머지를 채워야 합니다.

3.5 = 3 whole Convertible Reserved Instances + 1 additional Convertible Reserved Instance.

4번째 전환형 예약 인스턴스는 다른 3개의 전환형 인스턴스와 종료 날짜가 동일합니다. 부분 선결제 또는 전체 선결제 전환형 예약 인스턴스를 교환할 경우 4번째 동일 시작 인스턴스에 대해 트루업 비용을 지불하게

됩니다. 전환형 예약 인스턴스의 나머지 선결제 비용이 \$500이고 대상 동일 시작 인스턴스가 비례 할당 계산 기준으로 \$600일 경우 \$100가 청구됩니다.

`$600 prorated upfront cost of new reservations - $500 remaining upfront cost of original reservations = $100 difference.`

교환 요청 제출

AWS에서는 여러 경로로 교환 요청을 확인하고 작업할 수 있는데, AWS Management Console을 사용하거나 Amazon EC2 API를 통해 직접 작업하는 방법, 그리고 명령줄 인터페이스를 사용하는 방법이 있습니다.

항목

- [AWS Management Console \(p. 199\)](#)
- [명령행 인터페이스 \(p. 199\)](#)
- [Amazon EC2 API \(p. 199\)](#)

AWS Management Console

전환형 예약 인스턴스의 검색 결과에서 새로운 구성 선택할 수 있습니다.

AWS Management Console을 사용하여 전환형 예약 인스턴스를 교환하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. [Reserved Instances]를 선택하여 교환할 전환형 예약 인스턴스를 하나 이상 선택한 다음 [Actions], [Exchange Reserved Instances]를 선택합니다.
3. 새로운 구성 선택합니다. 기본 설정인 [Any]를 사용하거나, 혹은 드롭다운 메뉴에서 원하는 구성 설정할 수도 있습니다.
4. [Find Offering]을 선택합니다.
5. 목록에서 새로운 전환형 예약 인스턴스를 선택한 후 [Exchange]를 선택합니다.

이전 예약 인스턴스는 제거되고, 새로운 예약 인스턴스가 AWS Management Console에 표시됩니다. 이 프로세스는 완료하는데 몇 분 정도 걸릴 수 있습니다.

명령행 인터페이스

AWS CLI를 사용하여 먼저 전환형 예약 인스턴스에 대한 정보를 얻고 나서([get-reserved-instances-exchange-quote](#)) 교환하는 방식으로([accept-reserved-instances-exchange-quote](#)) 전환형 예약 인스턴스를 체계적으로 교환할 수 있습니다.

Amazon EC2 API

전환형 예약 인스턴스에 대한 정보는 [GetReservedInstancesExchangeQuote](#) 작업으로 얻을 수 있습니다. 그런 다음 [AcceptReservedInstancesExchangeQuote](#) 작업을 사용해 교환하면 됩니다. 자세한 내용은 [Amazon EC2 API 참조](#)를 참조하십시오.

변경 요청 문제 해결

요청한 변경 항목이 중복되지 않는 고유한 설정이라면 요청을 처리 중이라는 메시지가 표시됩니다. 이 시점에서는 Amazon EC2에서 변경 요청의 파라미터가 유효함을 확인한 상태입니다. 처리 과정에서 용량이 부족해 변경 요청이 실패할 가능성은 여전히 존재합니다.

일부의 경우, 확인 메시지 대신 완료 실패나 변경 실패 메시지가 표시될 수 있습니다. 메시지에 표시된 정보는 변경 요청을 다시 신청하는 데 참고 기준으로 사용하면 도움이 됩니다.

선택 항목 중 변경할 수 없는 예약 인스턴스가 존재합니다

Amazon EC2에서는 변경할 수 없는 예약 인스턴스를 식별하여 표시합니다. 이 메시지가 표시되었다면 AWS Management Console의 [Reserved Instances] 페이지로 이동해 해당 용량 예약에 대한 세부 정보를 확인하십시오.

변경 요청을 처리하는 동안 오류가 발생했습니다

하나 이상의 예약 인스턴스의 변경을 요청한 후 이 중 어떤 요청도 처리할 수 없을 때 표시되는 메시지입니다. 변경을 시도한 예약의 개수에 따라 다른 버전의 메시지가 표시될 수 있습니다.

Amazon EC2에서 요청을 처리할 수 없는 이유를 표시합니다. 예를 들어 변경하려는 예약 인스턴스의 하위 그룹 중 하나 이상의 그룹에 동일한 변경 항목(가용 영역과 플랫폼)을 설정했을 수 있습니다. 예약의 인스턴스 세부 정보가 일치하는지와 예약의 모든 하위 그룹에 대해 요청한 변경 사항이 서로 겹치지 않는지를 확인한 다음, 변경 요청을 다시 시도해 봅니다.

정기 예약 인스턴스

정기 예약 인스턴스(정기 인스턴스)를 사용하여 1년 동안 지정된 시작 시간과 기간에 따라 매일, 매주 또는 매월 반복적으로 용량 예약을 구입할 수 있습니다. 필요할 때 사용할 수 있도록 용량을 미리 예약합니다. 인스턴스를 사용하지 않더라도 인스턴스가 예약된 시간에 대한 비용을 지불합니다.

정기 인스턴스는 지속적으로 실행되지 않지만, 정기적으로 실행되고 정해진 시간에 완료되는 워크로드에 적합한 옵션입니다. 예를 들어, 업무 시간 중에 실행되는 애플리케이션 또는 주말에 실행되는 일괄 처리에 대해 정기 인스턴스를 사용할 수 있습니다.

지속적으로 용량 예약이 필요한 경우, 예약 인스턴스를 사용하면 요구에 꼭 맞는 동시에 비용을 절감할 수 있습니다. 자세한 내용은 [예약 인스턴스 \(p. 174\)](#) 섹션을 참조하십시오. 인스턴스를 실행하는 시간이 유동적인 경우, 스팟 인스턴스를 사용하면 요구에 꼭 맞는 동시에 비용을 절감할 수 있습니다. 자세한 내용은 [스팟 인스턴스 \(p. 203\)](#) 섹션을 참조하십시오.

목차

- [예약된 인스턴스의 작동 방식 \(p. 200\)](#)
- [예약된 인스턴스 구매 \(p. 201\)](#)
- [예약된 인스턴스 시작 \(p. 201\)](#)
- [예약된 인스턴스 제한 \(p. 202\)](#)

예약된 인스턴스의 작동 방식

Amazon EC2는 정기 인스턴스로 사용하기 위해 각 가용 영역에서 EC2 인스턴스 풀을 무효화합니다. 각 풀은 인스턴스 유형, 운영 체제 및 네트워크의 특정 조합(EC2-Classic 또는 EC2-VPC)을 지원합니다.

시작하려면 사용 가능한 일정을 검색해야 합니다. 여러 풀 또는 단일 풀을 검색할 수 있습니다. 적합한 일정을 찾은 다음 해당 일정을 구매합니다.

인스턴스 유형, 가용 영역, 네트워크 및 플랫폼과 같이 구입한 일정의 속성에 일치하는 시작 구성 사용하여 지정 기간 중에 예약된 인스턴스를 실행해야 합니다. 그러면 Amazon EC2에서는 지정된 시작 사양에 따라 사용자를 대신하여 EC2 인스턴스를 시작합니다. Amazon EC2는 현재 지정 기간이 끝날 때까지 EC2 인스턴스가 종료되도록 함으로써 예약된 다른 정기 인스턴스들에 대한 가용 용량을 확보해야 합니다. 따라서 현재 지정 기간이 끝나기 전에 Amazon EC2에서 EC2 인스턴스를 종료합니다.

정기 인스턴스를 종지하거나 재부팅할 수 없지만, 필요한 경우 수동으로 종료할 수 있습니다. 현재 지정 기간이 종료되기 전에 정기 인스턴스를 종료하는 경우, 몇 분 후에 다시 시작할 수 있습니다. 그렇지 않으면 다음 예약된 시간까지 기다려야 합니다.

다음 그림은 정기 인스턴스의 수명 주기를 보여줍니다.

예약된 인스턴스 구매

정기 인스턴스를 구입하려면 정기 예약 인스턴스 예약 마법사를 사용할 수 있습니다.

Warning

정기 인스턴스를 구입한 이후에는 구입을 취소하거나, 수정하거나, 재판매할 수 없습니다.

정기 인스턴스를 구입하는 방법은 다음과 같습니다. 삭제하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창의 [INSTANCES]에서 [Scheduled Instances]를 선택합니다.
3. [Purchase Scheduled Instances]를 선택합니다.
4. [Find available schedules] 페이지에서 다음을 수행하십시오.
 - a. [Create a schedule]의 [Starting on]에서 시작 날짜를 선택하고, [Recurring]에서 예약 반복(매일, 매주 또는 매월)을 선택하고, [for duration]에서 최소 기간을 선택합니다. 콘솔에서는 정기 인스턴스에 필요한 최소 사용률을 충족하는 최소 기간 값을 지정하는지 확인합니다(연간 1,200시간).
 - b. 인스턴스 세부정보의 플랫폼에서 운영 체제와 네트워크를 선택합니다. 결과 범위를 좁히려면 [Instance type]에서 하나 이상의 인스턴스 유형을 선택하거나 [Availability Zone]에서 하나 이상의 가용 영역을 선택합니다.
 - c. [Find schedules]를 선택합니다.
 - d. [Available schedules]에서 하나 이상의 일정을 선택합니다. 선택하는 각 일정에 대해 인스턴스의 수량을 설정한 다음 [Add to Cart]를 선택합니다.
 - e. 장바구니가 페이지의 아래쪽에 표시됩니다. 장바구니에서 일정 추가 및 제거를 마쳤으면 [Review and purchase]를 선택합니다.
5. 검토 및 구입 페이지에서 선택 항목을 확인하고 필요한 경우 편집합니다. 작업을 마쳤으면 구입을 선택합니다.

AWS CLI를 사용해 정기 인스턴스를 구입하려면

`describe-scheduled-instance-availability` 명령을 사용해 요구 사항을 충족하는 일정 목록을 표시한 다음, `purchase-scheduled-instances` 명령을 사용해 구입을 완료합니다.

예약된 인스턴스 시작

정기 인스턴스를 구입한 후 예약 기간 동안 인스턴스를 시작할 수 있습니다.

콘솔을 사용하여 정기 인스턴스를 시작하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창의 [INSTANCES]에서 [Scheduled Instances]를 선택합니다.
3. 정기 인스턴스를 선택하고 정기 인스턴스 시작을 선택합니다.
4. 구성 페이지에서 정기 인스턴스의 시작 사양을 완료한 다음 검토를 선택합니다.

Important

시작 사양이 구입한 일정의 인스턴스 유형, 가용 영역, 네트워크 및 플랫폼에 일치해야 합니다.

5. [Review] 페이지에서 시작 구성을 확인하고 필요한 경우 수정합니다. 작업을 마쳤으면 [Launch]를 선택합니다.

AWS CLI를 사용하여 정기 인스턴스를 시작하려면

[describe-scheduled-instances](#) 명령을 사용해 정기 인스턴스 목록을 표시한 다음, 예약 기간 동안 [run-scheduled-instances](#) 명령을 사용해 각 정기 인스턴스를 시작합니다.

예약된 인스턴스 제한

정기 인스턴스에는 다음 제한이 적용됩니다.

- C3, C4, M4 및 R3 인스턴스 유형만 지원됩니다.
- 필수 기간은 365일(1년)입니다.
- 최소 필수 사용률은 연간 1,200시간입니다.
- 정기 인스턴스를 최대 3개월 전에 미리 구입할 수 있습니다.

스팟 인스턴스

스팟 인스턴스를 사용하면 미사용 EC2 인스턴스에 입찰하여 Amazon EC2 비용을 상당히 줄일 수 있습니다. 스팟 인스턴스(각 가용 영역에 있는 각 인스턴스 유형)에 대한 시간당 가격은 Amazon EC2에서 설정되며 스팟 인스턴스의 공급과 수요에 따라 변동합니다. 스팟 인스턴스는 입찰이 현재 시장 가격을 초과할 때마다 실행됩니다.

스팟 인스턴스는 애플리케이션이 실행되는 시간을 유연하게 조정할 수 있고 애플리케이션을 중단할 수 있는 경우에 선택할 수 있는 비용 효과적인 방법입니다. 예를 들어, 스팟 인스턴스는 데이터 분석, 배치 작업, 백그라운드 프로세싱 및 선택적 작업에 매우 적합합니다. 자세한 내용은 [Amazon EC2 스팟 인스턴스](#) 섹션을 참조하십시오.

스팟 인스턴스와 온디맨드 인스턴스 간의 주요 차이점은 스팟 인스턴스는 즉시 시작되지 않을 수 있고, 스팟 인스턴스의 시간당 가격은 수요에 따라 다르며, 스팟 인스턴스의 시간당 가격이나 가용성이 변경되면 Amazon EC2가 개별 스팟 인스턴스를 종료할 수 있다는 것입니다. Amazon의 전략은 온디맨드 인스턴스의 코어 그룹을 시작하여 애플리케이션에 대해 보장된 컴퓨팅 리소스를 최소 수준으로 유지하고 기회가 생기면 스팟 인스턴스로 보완하는 것입니다.

또 하나의 전략은 필요한 지속 시간(스팟 블록이라고도 함)을 갖춘 스팟 인스턴스를 시작하는 것인데, 이는 스팟 가격 변동으로 인해 종단되지 않습니다. 자세한 내용은 [스팟 인스턴스의 지속 시간 지정 \(p. 213\)](#) 섹션을 참조하십시오.

개념

스팟 인스턴스를 시작하기 전에 다음 개념을 익혀야 합니다.

- **스팟 인스턴스 풀-동일한 인스턴스 유형, 운영 체제, 가용 영역 및 네트워크 플랫폼을 지닌 일련의 미사용 EC2 인스턴스(EC2-Classic 또는 EC2-VPC).**
- **스팟 가격 - 스팟 인스턴스의 시간당 현재 시중 가격입니다. 이 가격은 마지막 이행된 입찰을 기준으로 Amazon EC2에서 설정됩니다. 스팟 가격 기록도 검색할 수 있습니다.**
- **스팟 인스턴스 요청 또는 스팟 입찰 - 스팟 인스턴스에 대해 시간당 지불하려고 하는 최고 가격(입찰 가격)을 제공합니다. 입찰 가격이 스팟 가격을 초과하면 Amazon EC2가 사용자의 요청을 이행합니다. 스팟 인스턴스 요청이 일회성이거나 영구적입니다. Amazon EC2는 요청과 연결된 스팟 인스턴스가 종료된 후 자동으로 영구 스팟 요청을 다시 제출합니다. 스팟 인스턴스 요청은 스팟 인스턴스에 대해 지속 시간을 지정 할 수 있는 옵션이 있습니다.**
- **스팟 집합 - 사용자가 지정한 조건을 바탕으로 시작되는 스팟 인스턴스 세트입니다. 스팟 집합은 사용자의 요구를 충족시키는 스팟 인스턴스 풀을 선택하고 집합에 대한 목표 용량을 충족시키는 스팟 인스턴스를 시작합니다. 기본적으로 스팟 집합은 집합에서 스팟 인스턴스가 종료된 후 교체 인스턴스를 시작하여 목표 용량을 유지하도록 설정됩니다. 이런 스팟 집합은 일단 인스턴스가 종료된 후에는 유지되지 않는 일회성 요청으로 제출할 수도 있습니다.**
- **스팟 인스턴스 종단 - 스팟 가격이 입찰 가격을 초과하거나 미사용 EC2 인스턴스가 더 이상 없는 경우 Amazon EC2가 스팟 인스턴스를 종료합니다. Amazon EC2는 스팟 인스턴스에 종료 표시를 하고 종료 2분 전에 경고하는 스팟 인스턴스 종료 공지를 제공합니다.**
- **입찰 상태 - 스팟 입찰의 현재 상태에 대한 세부 정보를 제공합니다.**

시작하는 방법

Amazon EC2 사용에 앞서 가장 먼저 설정이 필요합니다. 스팟 인스턴스를 시작하기 전에 온디맨드 인스턴스를 시작해 보는 것도 도움이 될 수 있습니다.

실행 안내

- [Amazon EC2로 설정 \(p. 16\)](#)
- [Amazon EC2 Linux 인스턴스 시작하기 \(p. 21\)](#)

스팟 기본 사항

- [스팟 인스턴스 작업 방식 \(p. 205\)](#)
- [스팟 집합의 작동 방식 \(p. 208\)](#)

스팟 인스턴스 작업

- [종단에 대한 준비 \(p. 243\)](#)
- [스팟 인스턴스 요청 생성 \(p. 215\)](#)
- [입찰 상태 정보 가져오기 \(p. 240\)](#)

스팟 집합 작업

- [스팟 집합 사전 요구사항 \(p. 221\)](#)
- [스팟 집합 요청 생성 \(p. 223\)](#)

관련 서비스

Amazon EC2를 사용하여 스팟 인스턴스를 직접 프로비저닝할 수 있습니다. 또한 AWS의 다른 서비스를 사용하여 스팟 인스턴스를 프로비저닝할 수도 있습니다. 자세한 내용은 다음 문서를 참조하십시오.

Auto Scaling 및 스팟 인스턴스

Auto Scaling에서 스팟 인스턴스를 시작할 수 있도록 입찰 가격으로 시작 구성을 생성할 수 있습니다. 자세한 내용은 Auto Scaling 사용 설명서의 [Launching Spot instances in Your Auto Scaling Group](#)을 참조하십시오.

Amazon EMR 및 스팟 인스턴스

Amazon EMR 클러스터에서 스팟 인스턴스를 실행하는 것이 유용할 수 있는 시나리오가 있습니다. 자세한 내용은 Amazon EMR 개발자 안내서의 [Lower Costs with Spot Instances](#) 섹션을 참조하십시오.

AWS CloudFormation 템플릿

AWS CloudFormation에서는 JSON 형식의 템플릿을 사용하여 AWS 리소스 컬렉션을 생성하고 관리할 수 있습니다. AWS CloudFormation 템플릿에는 스팟 가격이 포함될 수 있습니다. 자세한 내용은 [EC2 Spot Instance Updates - Auto Scaling and CloudFormation Integration](#)을 참조하십시오.

AWS SDK for Java

Java 프로그래밍 언어를 사용하여 스팟 인스턴스를 관리할 수 있습니다. 자세한 내용은 [Tutorial: Amazon EC2 Spot Instances](#) 및 [Tutorial: Advanced Amazon EC2 Spot Request Management](#) 섹션을 참조하십시오.

.NET용 AWS SDK

.NET 프로그래밍 환경을 사용하여 스팟 인스턴스를 관리할 수 있습니다. 자세한 내용은 [Tutorial: Amazon EC2 Spot instances](#)를 참조하십시오.

요금

스팟 인스턴스에 대해 스팟 가격을 지불합니다. 이 가격은 Amazon EC2에서 설정되며 스팟 인스턴스의 수요와 공급에 따라 변동합니다. 입찰 가격이 현재 스팟 가격을 초과하는 경우 Amazon EC2에서 요청이 이행되며 사용자가 인스턴스를 종료하거나 스팟 가격이 입찰 가격보다 높아질 때까지 스팟 인스턴스가 실행됩니다.

입찰 가격이 더 높은지 여부와 상관없이 모든 사람이 해당 기간 동안 동일한 스팟 가격을 지불합니다. 시간당 입찰 가격보다 더 많이 지불하는 경우는 없으며 대개 시간당 더 적은 금액을 지불합니다. 예를 들어, 시간당

0.25 USD를 입찰한 경우 스팟 가격이 시간당 0.20 USD이면 시간당 0.20 USD만 지불합니다. 스팟 가격이 하락하면 인하된 새로운 가격을 지불합니다. 스팟 가격이 상승하면 해당 가격이 입찰 가격보다 적거나 같은 경우 새로운 가격을 지불합니다. 스팟 가격이 입찰 가격보다 상승하면 스팟 인스턴스가 중단됩니다.

각 인스턴스 시간이 시작될 때 스팟 가격을 기준으로 요금이 청구됩니다. 스팟 가격이 입찰 가격을 초과하여 스팟 인스턴스가 인스턴스 시간 중에 중단되는 경우 중단된 스팟 사용 시간에 대해서는 요금이 청구되지 않습니다. 하지만 인스턴스 시간 중간에 스팟 인스턴스를 종료할 경우 시간 요금이 청구됩니다.

미리 정해진 지속 시간을 지닌 스팟 인스턴스는 실행 중에 스팟 인스턴스에 대해 여전히 유효한 시간당 고정 가격을 사용합니다.

가격 보기

리전 및 인스턴스 유형당 현재(5분마다 업데이트됨) 최저 스팟 가격을 보려면 [요금 페이지](#)를 참조하십시오.

지난 3개월 동안의 스팟 가격 기록을 보려면 Amazon EC2 콘솔 또는 [describe-spot-price-history 명령\(AWS CLI\)](#)을 사용하십시오. 자세한 내용은 [스팟 인스턴스 요금 기록 \(p. 212\)](#) 섹션을 참조하십시오.

각 AWS 계정의 코드에 가용 영역을 독립적으로 매핑합니다. 따라서 서로 다른 계정 간에 동일한 가용 영역 코드(예: us-west-2a)에 대한 결과가 다를 수 있습니다.

결제 보기

청구 요금을 검토하려면 [AWS 계정 활동 페이지](#)를 참조하십시오. 청구서에는 요금 내역을 자세하게 확인할 수 있는 사용 보고서 링크가 포함됩니다. 자세한 내용은 [AWS Account Billing](#)을 참조하십시오.

AWS 결제, 계정 및 이벤트에 관련된 질문은 [AWS Support](#)에 문의하십시오.

스팟 인스턴스 작업 방식

스팟 인스턴스를 사용하려면 스팟 인스턴스 요청 또는 스팟 집합 요청을 생성하십시오. 이 요청에는 인스턴스별로 시간당 지불하려는 최고 가격(입찰 가격)과 인스턴스 유형 및 가용 영역과 같은 기타 제약 조건이 포함됩니다. 입찰 가격이 지정된 인스턴스의 현재 스팟 가격보다 높고 지정된 인스턴스가 사용 가능한 상태인 경우 요청이 즉시 이행됩니다. 그렇지 않으면 스팟 가격이 입찰 가격 아래로 하락하거나 지정된 인스턴스가 사용 가능하게 될 때 요청이 이행됩니다. 스팟 인스턴스는 사용자가 인스턴스를 종료하거나 Amazon EC2에서 인스턴스를 종료(스팟 인스턴스 중단이라고 함)해야 할 때까지 실행됩니다.

스팟 인스턴스를 사용할 때는 중단에 대비해야 합니다. Amazon EC2에서는 스팟 가격이 입찰 가격보다 상승하거나 스팟 인스턴스에 대한 수요가 증가하거나 스팟 인스턴스의 공급이 감소할 때 스팟 인스턴스를 중단할 수 있습니다. Amazon EC2는 스팟 인스턴스에 종료 표시를 할 때, 종료 2분 전에 경고하는 스팟 인스턴스 종료 공지를 제공합니다. 스팟 인스턴스에 대한 종료 방지 기능은 활성화할 수 없다는 점에 유의하십시오. 자세한 내용은 [스팟 인스턴스 종단 \(p. 242\)](#) 섹션을 참조하십시오.

Amazon EBS 기반 인스턴스가 스팟 인스턴스인 경우 이 인스턴스를 중지하고 다시 시작할 수 없지만 재부팅하거나 종료할 수 있습니다.

OS 수준에서 스팟 인스턴스를 종료하면 스팟 인스턴스가 종료됩니다. 이 동작은 변경할 수 없습니다.

목차

- [스팟 마켓의 공급과 수요 \(p. 205\)](#)
- [시작 그룹에서 스팟 인스턴스 시작 \(p. 207\)](#)
- [가용 영역 그룹에서 스팟 인스턴스 실행 \(p. 207\)](#)
- [VPC에서 스팟 인스턴스 시작 \(p. 207\)](#)

스팟 마켓의 공급과 수요

AWS는 각 스팟 인스턴스 풀에서 사용 가능한 스팟 인스턴스의 수를 지속적으로 평가하고, 각 풀에 대해 수행된 입찰을 모니터링하며, 사용 가능한 스팟 인스턴스를 최고 입찰자에게 프로비저닝합니다. 풀에 대한 스

팟 가격은 해당 풀에 대해 이행된 입찰 중 최저 가격으로 설정됩니다. 따라서 스팟 가격은 단일 스팟 인스턴스에 대한 스팟 요청을 즉시 이행하려면 그 이상으로 입찰해야 하는 가격입니다.

예를 들어, 스팟 인스턴스 요청을 생성하고 해당 스팟 인스턴스 풀에 판매용 스팟 인스턴스가 다섯 개만 있다 고 가정합니다. 입찰 가격은 현재 스팟 가격인 0.10 USD입니다. 다음 표에서는 내림차순으로 배열된 현재 입찰 가격을 보여 줍니다. 입찰 1-5가 이행됩니다. 마지막으로 이행된 입찰 5는 스팟 가격을 0.10 USD로 설정 합니다. 입찰 6은 이행되지 않습니다. 0.10 USD라는 동일한 입찰 가격을 공유하는 입찰 3-5는 임의 순서로 배열됩니다.

입찰	입찰 가격	현재 스팟 가격	참고
1	1.00 USD	0.10 USD	
2	1.00 USD	0.10 USD	
3	0.10 USD	0.10 USD	
4	0.10 USD	0.10 USD	사용자의 입찰
5	0.10 USD	0.10 USD	스팟 가격을 설정하는 마지막으로 이행된 입찰입니다. 해당 기간 동안 모든 사람이 동일한 스팟 가격을 지불합니다.
---	---		스팟 용량 컨오프
6	0.05 USD		

이제 이 풀의 크기가 3으로 감소했다고 가정하면, 입찰 1-3이 이행됩니다. 마지막으로 이행된 입찰 3은 스팟 가격을 0.10 USD로 설정합니다. 0.10 USD인 입찰 4-5는 이행되지 않습니다. 여기에서 스팟 가격은 변경되지 않았지만 스팟 공급이 감소했기 때문에 사용자의 입찰을 포함하여 입찰 중 두 개는 더 이상 이행되지 않습니다.

입찰	입찰 가격	현재 스팟 가격	참고
1	1.00 USD	0.10 USD	
2	1.00 USD	0.10 USD	
3	0.10 USD	0.10 USD	스팟 가격을 설정하는 마지막으로 이행된 입찰입니다. 해당 기간 동안 모든 사람이 동일한 스팟 가격을 지불합니다.
---	---		스팟 용량 컨오프
4	0.10 USD		사용자의 입찰
5	0.10 USD		
6	0.05 USD		

이 풀의 단일 인스턴스에 대한 스팟 요청을 이행하려면 현재 스팟 가격인 0.10 USD보다 높게 입찰해야 합니다. 0.101 USD를 입찰하면 요청이 이행되고 입찰 3에 대한 스팟 인스턴스가 종단되며 스팟 가격은 0.101 USD가 됩니다. 2.00 USD를 입찰하면 입찰 3에 대한 스팟 인스턴스가 종단되고 스팟 가격은 1.00 USD(입찰 2에 대한 가격)가 됩니다.

얼마나 높게 입찰하든 상관없이 스팟 인스턴스 풀에서 사용 가능한 스팟 인스턴스 수보다 많이 가져올 수 없습니다. 풀 크기가 0으로 떨어지면 풀의 모든 스팟 인스턴스가 중단됩니다.

시작 그룹에서 스팟 인스턴스 시작

스팟 인스턴스 요청에서 시작 그룹을 지정하여 해당 인스턴스를 모두 시작할 수 있는 경우에만 스팟 인스턴스 세트를 시작하도록 Amazon EC2에 알립니다. 또한 스팟 서비스가 시작 그룹에 있는 인스턴스 중 하나를 종료해야 하는 경우(예를 들어, 스팟 가격이 입찰 가격보다 상승하는 경우) 모든 인스턴스를 종료해야 합니다. 그러나 사용자가 시작 그룹에 있는 인스턴스를 하나 이상 종료하는 경우 Amazon EC2는 시작 그룹에 있는 나머지 인스턴스를 종료하지 않습니다.

이 옵션이 유용할 수 있지만 이러한 제약 조건을 추가하면 스팟 인스턴스 요청이 이행될 가능성은 낮아지고 스팟 인스턴스가 종료될 가능성은 높아질 수 있습니다.

이전의 성공적인 요청과 동일한(기존) 시작 그룹을 지정하는 다른 성공적인 스팟 인스턴스 요청을 생성하면 새로운 인스턴스가 시작 그룹에 추가됩니다. 이후 이 시작 그룹의 인스턴스가 종료되면 첫 번째 및 두 번째 요청에서 시작된 인스턴스를 포함하여 시작 그룹의 모든 인스턴스가 종료됩니다.

가용 영역 그룹에서 스팟 인스턴스 실행

스팟 인스턴스 요청에서 가용 영역 그룹을 지정하여 동일한 가용 영역에 있는 스팟 인스턴스 세트를 시작하도록 스팟 서비스에 알립니다. Amazon EC2가 가용 영역 그룹에 있는 모든 인스턴스를 동시에 종료할 필요는 없습니다. Amazon EC2가 가용 영역 그룹의 인스턴스를 하나 종료해야 하는 경우 다른 인스턴스는 실행 중인 상태로 유지됩니다.

이 옵션이 유용할 수 있지만 이러한 제약 조건을 추가하면 스팟 인스턴스 요청이 이행될 가능성은 낮아질 수 있습니다.

가용 영역 그룹을 지정하지만 스팟 인스턴스 요청에서 가용 영역을 지정하지 않는 경우 결과는 EC2-Classic 네트워크, 기본 VPC 또는 기본이 아닌 VPC 중 무엇을 지정했는지에 따라 다릅니다. EC2-Classic 및 EC2-VPC에 대한 자세한 내용은 [지원되는 플랫폼 \(p. 471\)](#) 섹션을 참조하십시오.

EC2-Classic

Amazon EC2는 리전에서 최저 가격의 가용 영역을 찾고 그룹에 대한 최저 입찰이 해당 가용 영역의 현재 스팟 가격보다 높은 경우 해당 가용 영역에서 스팟 인스턴스를 시작합니다. 스팟 가격이 그룹에 대한 최저 입찰보다 낮게 유지되는 한, Amazon EC2는 스팟 인스턴스를 함께 시작할 수 있는 충분한 용량이 될 때까지 대기합니다.

기본 VPC

Amazon EC2는 지정된 서브넷에 대한 가용 영역을 사용하거나, 서브넷을 지정하지 않은 경우 가용 영역과 기본 서브넷을 선택하지만 해당 가용 영역은 최저 가격의 가용 영역이 아닐 수 있습니다. 가용 영역에 대한 기본 서브넷을 삭제한 경우 다른 서브넷을 지정해야 합니다.

기본이 아닌 VPC

Amazon EC2는 지정된 서브넷에 대한 가용 영역을 사용합니다.

VPC에서 스팟 인스턴스 시작

스팟 인스턴스를 사용할 때 EC2-VPC의 기능을 이용하려면 VPC에서 스팟 인스턴스가 시작되도록 스팟 요청에서 지정합니다. 온디맨드 인스턴스에 대해 서브넷을 지정하는 것과 동일한 방법으로 스팟 인스턴스에 대해 서브넷을 지정합니다.

VPC에서 스팟 인스턴스를 시작하는 스팟 인스턴스 요청을 수행하는 프로세스는 다음과 같은 차이점을 제외하고 EC2-Classic에서 스팟 인스턴스를 시작하는 스팟 인스턴스 요청을 수행하는 프로세스와 동일합니다.

- VPC에 있는 스팟 인스턴스의 스팟 가격 기록을 기준으로 입찰해야 합니다.
- [기본 VPC] 낮은 가격의 특정 가용 영역에서 스팟 인스턴스가 시작되도록 하려면 스팟 인스턴스 요청에서 해당 서브넷을 지정해야 합니다. 서브넷을 지정하지 않으면 Amazon EC2에서 서브넷이 자동으로 선택되며, 이 서브넷에 대한 가용 영역에는 최저 스팟 가격이 없을 수 있습니다.

- [기본이 아닌 VPC] 스팟 인스턴스에 대해 서브넷을 지정해야 합니다.

스팟 집합의 작동 방식

스팟 집합은 스팟 인스턴스의 모음입니다. 스팟 집합이 스팟 집합 요청에서 지정한 목표 용량을 충족시키는 데 필요한 스팟 인스턴스의 수 시작을 시도합니다. 또한, 스팟 인스턴스가 스팟 가격 또는 사용 가능한 용량의 변경으로 인해 중단될 경우 스팟 집합은 목표 용량 집합을 유지하려고 시도합니다.

스팟 인스턴스 풀은 동일한 인스턴스 유형, 운영 체제, 가용 영역 및 네트워크 플랫폼을 지닌 일련의 미사용 EC2 인스턴스입니다(EC2-Classic 또는 EC2-VPC). 스팟 집합 요청을 할 때 인스턴스 유형, AMI, 가용 영역 또는 서브넷에 따라 바뀌는 여러 시작 사양을 포함할 수 있습니다. 스팟 집합은 스팟 집합 요청에 포함된 시작 사양을 기준으로 하는 요청과 스팟 집합 요청의 구성을 이행하는 데 사용되는 스팟 인스턴스 풀을 선택합니다. 스팟 인스턴스는 선택한 풀에서 가져옵니다.

목차

- [스팟 집합 할당 전략 \(p. 208\)](#)
- [스팟 가격 재정의 \(p. 209\)](#)
- [스팟 집합 인스턴스 가중치 부여 \(p. 209\)](#)
- [연습: 인스턴스 가중치를 부여한 스팟 집합 사용 \(p. 210\)](#)

스팟 집합 할당 전략

시작 사양으로 표시되는 가용 스팟 인스턴스 풀로부터 스팟 집합 요청을 이행하는 방법은 스팟 집합에 대한 할당 전략에 따라 결정됩니다. 다음은 스팟 집합 요청에서 지정할 수 있는 할당 전략입니다.

`lowestPrice`

스팟 인스턴스는 최저 가격의 풀에서 가져옵니다. 이는 기본 전략입니다.

`diversified`

스팟 인스턴스는 모든 풀에 분산됩니다.

할당 전략 선택

사용 사례를 바탕으로 스팟 집합을 최적화할 수 있습니다.

집합이 작거나 짧은 시간 동안 작동할 경우, 모든 인스턴스가 단일 스팟 인스턴스 풀에 있더라도 스팟 인스턴스가 중단될 확률이 낮습니다. 따라서 `lowestPrice` 전략이 요구를 충족시키는 동시에 최저 가격을 제공할 가능성이 높습니다.

집합이 크거나 긴 시간 동안 작동할 경우, 스팟 인스턴스를 여러 풀로 분산하여 집합의 가용성을 개선할 수 있습니다. 예를 들어 스팟 집합 요청에 풀 10개와 목표 용량으로 인스턴스 100개가 지정되어 있으면, 스팟 집합이 각 풀에서 10개의 스팟 인스턴스를 시작합니다. 한 풀에 대한 스팟 가격이 이 풀에 대한 입찰 가격 이상으로 높아지는 경우, 집합 중 10%만 영향을 받습니다. 이 전략을 사용하면 집합이 시간이 지나면서 어느 한 풀에서 발생하는 스팟 가격의 상승에 덜 민감해집니다.

`diversified` 전략에서는 스팟 집합이 [온디맨드 가격](#)보다 높은 스팟 가격을 지닌 풀로 스팟 인스턴스를 시작하지 않는다는 점에 유의하십시오.

목표 용량 유지

스팟 가격 또는 스팟 인스턴스 풀의 가용 용량 변화로 인해 스팟 인스턴스가 종료된 후에는 스팟 집합이 대체 스팟 인스턴스를 시작합니다. 할당 전략이 `lowestPrice`인 경우, 스팟 집합은 스팟 가격이 현재 가장 낮은 풀에서 대체 인스턴스를 시작합니다. 할당 전략이 `diversified`인 경우, 스팟 집합은 나머지 풀 전체에 걸쳐 대체 스팟 인스턴스를 배포합니다.

스팟 가격 재정의

각 스팟 집합 요청은 글로벌 스팟 가격을 포함해야 합니다. 기본적으로, 스팟 집합에서는 각 시작 사양에 대한 입찰 가격으로 이 가격을 사용합니다.

하나 이상의 시작 사양에서 스팟 가격을 선택적으로 지정할 수 있습니다. 이 입찰 가격은 시작 사양에 특정한 것입니다. 시작 사양에 특정 스팟 가격이 포함되는 경우 스팟 집합은 이 가격을 그 시작 사양에 대한 입찰 가격으로 사용하여 글로벌 스팟 가격을 재정의합니다. 특정 스팟 가격을 포함하지 않는 다른 시작 사양에서 글로벌 스팟 가격을 계속 사용합니다.

스팟 집합 인스턴스 가중치 부여

스팟 인스턴스의 집합을 요청할 때 각 인스턴스 유형이 애플리케이션의 성능에 기여하는 용량 단위를 정의하고, 인스턴스 가중치를 사용하여 적절히 각 스팟 인스턴스 풀에 대한 입찰 가격을 조정할 수 있습니다.

기본적으로, 사용자가 지정하는 스팟 가격은 인스턴스 시간당 입찰 가격을 나타냅니다. 인스턴스 가중치 기능을 사용할 때, 사용자가 지정하는 스팟 가격은 단위 시간당 입찰 가격을 나타냅니다. 단위 시간당 입찰 가격은 인스턴스 유형에 따른 입찰 가격을 인스턴스가 나타내는 단위 수로 나누어 계산합니다. 스팟 집합은 목표 용량을 인스턴스 가중치로 나누어 시작할 스팟 인스턴스의 수를 계산합니다. 결과가 정수가 아닌 경우, 스팟 집합은 결과를 다음 정수로 올림하므로 집합의 크기가 목표 용량을 밀돌지는 않습니다. 시작된 인스턴스의 용량이 요청된 목표 용량을 초과하더라도, 스팟 집합은 시작 사양에서 지정한 어떤 풀이든 선택할 수 있습니다.

다음 표에는 목표 용량이 10인 스팟 집합 요청을 위한 단위당 입찰 가격을 결정하기 위한 계산의 예가 포함되어 있습니다.

인스턴스 유형	인스턴스 가중치	인스턴스 시간당 스팟 가격	단위 시간당 스팟 가격	시작된 인스턴스의 수
r3.xlarge	2	0.05 USD	.025 (0.05를 2로 나눈 값)	5 (10을 2로 나눈 값)
r3.8xlarge	8	0.10 USD	.0125 (0.10를 8로 나눈 값)	2 (10을 8로 나눈 후 올림한 결과)

스팟 집합 인스턴스 가중치를 사용해 다음과 같이 원하는 목표 용량을 이행 시점에 단위당 최저 가격으로 풀에 프로비저닝합니다.

- 스팟 집합에 대한 목표 용량을 인스턴스(기본값) 또는 선택한 단위(예: 가상 CPU 수, 메모리, 스토리지 또는 처리량)로 설정합니다.
- 단위당 입찰 가격을 설정합니다.
- 각 시작 구성을 위해, 목표 용량으로 접근하는 방향으로 인스턴스 유형이 나타내는 단위 수를 의미하는 가중치를 지정합니다.

인스턴스 가중치 부여의 예

다음과 같은 구성의 스팟 집합 요청을 고려합니다.

- 목표 용량은 24
- 인스턴스 유형이 r3.2xlarge이고 가중치가 6인 시작 사양
- 인스턴스 유형이 c3.xlarge이고 가중치가 5인 시작 사양

가중치는 목표 용량에 대하여 인스턴스 유형이 나타내는 단위 수를 의미합니다. 첫 번째 시작 사양에서 단위당 최저 스팟 가격(인스턴스 시간당 `r3.2xlarge`에 대한 스팟 가격을 6으로 나눈 값)을 제공하는 경우, 스팟 집합은 이들 인스턴스 중 4개(24를 6으로 나눈 값)를 시작합니다.

두 번째 시작 사양에서 단위당 최저 스팟 가격(인스턴스 시간당 `c3.xlarge`에 대한 스팟 가격을 5로 나눈 값)을 제공하는 경우, 스팟 집합은 이들 인스턴스 중 5개(24를 5로 나눈 결과를 올림한 값)를 시작합니다.

인스턴스 가중치 부여 및 할당 전략

다음과 같은 구성의 스팟 집합 요청을 고려합니다.

- 목표 용량은 30
- 인스턴스 유형이 `c3.2xlarge`이고 가중치가 8인 시작 사양
- 인스턴스 유형이 `m3.xlarge`이고 가중치가 8인 시작 사양
- 인스턴스 유형이 `r3.xlarge`이고 가중치가 8인 시작 사양

스팟 집합이 4개(30을 8로 나눈 결과를 올림한 값)의 인스턴스를 시작합니다. `lowestPrice` 전략 사용 시, 4개의 인스턴스는 전부 단위당 최저 스팟 가격을 제공하는 풀에서 가져옵니다. `diversified` 전략 사용 시, 스팟 집합은 3개의 풀 각각에서 1개의 인스턴스를 시작하고 3개의 풀 중 어떤 것에 있는 것이든 4번째 인스턴스가 단위당 최저 스팟 가격을 제공합니다.

연습: 인스턴스 가중치를 부여한 스팟 집합 사용

이 연습에서는 Example Corp이라는 가상의 회사를 통해 인스턴스 가중치를 사용한 스팟 집합에 대한 입찰 프로세스를 설명합니다.

목표

제약 회사인 Example Corp은 암 퇴치 효과가 있는 화합물을 검출하는 데 Amazon EC2의 컴퓨팅 파워를 사용하려고 합니다.

계획

Example Corp은 먼저 [스팟 모범 사례](#)를 살펴봅니다. 그런 다음 스팟 집합에 대한 다음 요건을 결정합니다.

인스턴스 유형

Example Corp은 최소 60GB 메모리와 8개의 가상 CPU(vCPU)로 최적의 성능을 자랑하는 컴퓨팅 및 메모리 집약적 애플리케이션을 사용하고 있습니다. 하지만 최저 가격으로 이러한 애플리케이션 리소스를 극대화하는 것이 목표입니다. 그 결과 다음 EC2 인스턴스 유형 중 하나가 이러한 요구에 적합할 것이라는 결정을 내립니다.

인스턴스 유형	메모리(GiB)	vCPUs
<code>r3.2xlarge</code>	61	8
<code>r3.4xlarge</code>	122	16
<code>r3.8xlarge</code>	244	32

단위의 목표 용량

인스턴스 가중치를 부여했을 때 목표 용량은 인스턴스 수(기본값) 또는 코어(vCPU), 메모리(GiB) 및 스토리지(GB)와 같은 요소의 조합과 동일할 수 있습니다. 그래서 Example Corp는 단위 1개당 애플리케이션의 기본 용량(60GB 메모리, vCPU 8개)을 고려하여 기본 용량의 20배면 요구에 부응할 것이라고 결정을 내립니다. 그래서 스팟 집합 요청의 목표 용량을 20으로 설정합니다.

인스턴스 가중치

목표 용량이 결정되자 이제는 인스턴스 가중치를 계산합니다. 각 인스턴스 유형에 대한 인스턴스 가중치를 계산하기 위해, 다음과 같이 목표 용량에 이르기 위해 필요한 각 인스턴스 유형의 단위를 결정합니다.

- r3.2xlarge(61.0GB, 8 vCPU) = 단위 20개 중 1개
- r3.4xlarge(122.0GB, 16 vCPU) = 단위 20개 중 2개
- r3.8xlarge(122.0GB, 32 vCPU) = 단위 20개 중 4개

따라서 Example Corp은 스팟 집합 요청 시 1, 2 및 4의 인스턴스 가중치를 각 시작 구성에 할당합니다.

단위 시간당 입찰 가격

Example Corp은 인스턴스 시간당 [온디맨드 가격](#)으로 입찰 가격을 시작합니다. 그 밖에 최근 스팟 가격을 사용하거나, 둘을 조합할 수도 있습니다. 이때 단위 시간당 입찰 가격을 계산하려면 인스턴스 시간당 최초 입찰 가격을 가중치로 나눕니다. 예:

인스턴스 유형	온디맨드 가격	인스턴스 가중치	단위 시간당 가격
r3.2xLarge	\$0.7	1	\$0.7
r3.4xLarge	\$1.4	2	\$0.7
r3.8xLarge	\$2.8	4	\$0.7

Example Corp은 단위 시간당 글로벌 입찰 가격으로 \$0.7을 입력하여 세 가지 인스턴스 유형을 위해 경쟁할 수도 있습니다. 또한, r3.8xlarge 시작 사양에 단위 시간당 글로벌 입찰 가격으로 \$0.7와 단위 시간당 특별 입찰 가격으로 \$0.9를 입력할 수도 있습니다. 결국 Example Corp은 스팟 집합의 프로비저닝 전략에 따라 낮은 가격으로 입찰하여 비용을 줄일 수도 있고 높은 가격으로 입찰하여 중단 가능성은 낮출 수도 있습니다.

권한 검증

Example Corp은 스팟 집합 요청을 생성하기 전에 우선 필요한 권한을 가진 IAM 역할이 있는지 검증합니다. 자세한 내용은 [스팟 집합 사전 요구사항 \(p. 221\)](#) 섹션을 참조하십시오.

요청 생성

Example Corp은 스팟 집합 요청에 대해 다음 구성으로 config.json 파일을 생성합니다.

```
{  
    "SpotPrice": "0.70",  
    "TargetCapacity": 20,  
    "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",  
    "LaunchSpecifications": [  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "r3.2xlarge",  
            "SubnetId": "subnet-482e4972",  
            "WeightedCapacity": 1  
        },  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "r3.4xlarge",  
            "SubnetId": "subnet-482e4972",  
            "WeightedCapacity": 2  
        },  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "r3.8xlarge",  
            "SubnetId": "subnet-482e4972",  
            "SpotPrice": "0.90",  
            "WeightedCapacity": 4  
        }  
    ]  
}
```

```
}
```

Example Corp이 다음 `request-spot-fleet` 명령을 사용하여 스팟 집합 요청을 생성합니다.

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

자세한 내용은 [스팟 집합 요청 \(p. 220\)](#) 섹션을 참조하십시오.

이행

할당 전략에서는 스팟 인스턴스가 어느 스팟 인스턴스 풀에서 온 것인지 결정합니다.

(기본 전략인) `lowestPrice` 전략 사용 시, 스팟 인스턴스는 이행 시점에 단위당 최저 스팟 가격의 풀에서 온 것입니다. 20단위의 용량을 제공하려면 스팟 집합이 `r3.2xlarge` 인스턴스 20개(20을 1로 나눈 값), `r3.4xlarge` 인스턴스 10개(20을 2로 나눈 값) 또는 `r3.8xlarge` 인스턴스 5개(20을 4로 나눈 값)를 시작합니다.

Example Corp에서 `diversified` 전략을 사용한 경우에는 스팟 인스턴스가 3개의 풀 전부에서 옵니다. 스팟 집합은 총 20개의 단위에 대해 `r3.2xlarge` 인스턴스 6개(6개 단위 제공), `r3.4xlarge` 인스턴스 3개(6개 단위 제공), `r3.8xlarge` 인스턴스 2개(8개 단위 제공)를 시작합니다.

스팟 인스턴스 요금 기록

스팟 가격은 단일 스팟 요청이 이행되도록 보장하려면 그 이상으로 입찰해야 하는 가격을 나타냅니다. 입찰 가격이 스팟 가격보다 높으면 Amazon EC2가 스팟 인스턴스를 시작하고, 스팟 가격이 입찰 가격보다 높게 오르면 Amazon EC2가 스팟 인스턴스를 종료합니다. 스팟 요청이 빨리 이행되도록 현재 스팟 가격보다 높게 입찰할 수 있습니다. 그러나 스팟 인스턴스에 대해 입찰 가격을 지정하기 전에 스팟 가격 기록을 검토하는 것이 좋습니다. 인스턴스 유형, 운영 체제 및 가용 영역을 기준으로 필터링하여 지난 90일 동안의 스팟 가격 기록을 볼 수 있습니다.

스팟 가격 기록을 가이드로 사용하여 과거에 필요에 적합했던 입찰 가격을 선택할 수 있습니다. 예를 들어, 검토한 시간 범위에서 75% 작동 시간을 제공한 입찰 가격을 결정할 수 있습니다. 그러나 기록 추세는 향후 결과를 보장하지 않는다는 점에 주의해야 합니다. 스팟 가격은 실시간 공급 및 수요에 따라 변하며 스팟 가격에서 특정 패턴을 생성한 조건이 향후에는 발생하지 않을 수 있습니다.

콘솔을 사용하여 스팟 가격 기록을 보려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Spot Requests]를 선택합니다.
3. 스팟 인스턴스를 처음 사용하는 경우에는 시작 페이지가 나타나는데, [Get started]를 선택하고 화면 아래로 스크롤한 후 [Cancel]을 선택합니다.
4. [Pricing History]를 선택합니다. 기본적으로 지난 하루 동안 모든 가용 영역의 Linux `t1.micro` 인스턴스에 대한 데이터 그래프가 페이지에 표시됩니다. 마우스를 그래프 위로 이동하면 그래프 아래의 표에 특정 시간의 가격이 표시됩니다.
5. (선택 사항) 특정 가용 영역에 대한 스팟 가격 기록을 검토하려면 목록에서 가용 영역을 선택합니다. 다른 제품, 인스턴스 유형 또는 날짜 범위도 선택할 수 있습니다.

명령줄을 사용하여 스팟 가격 기록을 보려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- `describe-spot-price-history`(AWS CLI)
- `Get-EC2SpotPriceHistory`(Windows PowerShell용 AWS 도구)

스팟 인스턴스 요청

스팟 인스턴스를 사용하려면 인스턴스 수, 인스턴스 유형, 사용 영역 및 인스턴스 시간당 지불하려는 최고 가격(입찰)을 포함한 스팟 인스턴스 요청을 생성합니다. 입찰이 현재 스팟 가격을 초과하면 Amazon EC2에서 요청이 즉시 이행됩니다. 그렇지 않으면 요청이 이행될 수 있을 때까지 또는 사용자가 요청을 취소할 때까지 Amazon EC2가 대기합니다.

다음 그림에서는 스팟 요청이 작동하는 방식을 보여 줍니다. 스팟 인스턴스 종단에 대해 수행되는 작업은 요청 유형(일회 또는 영구)에 따라 다릅니다. 요청이 영구 요청인 경우 스팟 인스턴스가 종료된 후 요청이 다시 열립니다.

목차

- [스팟 인스턴스 요청 상태 \(p. 213\)](#)
- [스팟 인스턴스의 지속 시간 지정 \(p. 213\)](#)
- [스팟 인스턴스의 테넌시 지정 \(p. 214\)](#)
- [스팟 인스턴스 요청 생성 \(p. 215\)](#)
- [실행 중인 스팟 인스턴스 찾기 \(p. 217\)](#)
- [스팟 인스턴스 요청 태그 지정 \(p. 217\)](#)
- [스팟 인스턴스 요청 취소 \(p. 218\)](#)
- [스팟 요청 예시 시작 사양 \(p. 218\)](#)

스팟 인스턴스 요청 상태

스팟 인스턴스 요청은 다음 상태 중 하나일 수 있습니다.

- `open` - 요청이 이행될 때까지 대기 중입니다.
- `active` - 요청이 이행되며 요청에 연결된 스팟 인스턴스가 있습니다.
- `failed` - 요청에 하나 이상의 잘못된 파라미터가 있습니다.
- `closed` - 스팟 인스턴스가 종단되거나 종료되었습니다.
- `cancelled` - 사용자가 요청을 취소했거나 요청이 만료되었습니다.

다음 그림은 요청 상태 간의 전환을 나타냅니다. 전환은 요청 유형(일회 또는 영구)에 따라 다릅니다.

일회 스팟 인스턴스 요청은 Amazon EC2가 스팟 인스턴스를 시작하거나 요청이 만료되거나 사용자가 요청을 취소할 때까지 활성 상태로 유지됩니다. 스팟 가격이 입찰 가격보다 상승하면 스팟 인스턴스가 종료되고 스팟 인스턴스 요청이 닫힙니다.

영구 스팟 인스턴스 요청은 요청이 이행되더라도 요청이 만료되거나 사용자가 요청을 취소할 때까지 활성 상태로 유지됩니다. 예를 들어, 스팟 가격이 0.25 USD일 때 한 인스턴스에 대해 영구 스팟 인스턴스 요청을 생성한 경우 입찰 가격이 0.25 USD보다 높으면 Amazon EC2가 스팟 인스턴스를 시작합니다. 스팟 가격이 입찰 가격보다 상승하면 스팟 인스턴스가 종료됩니다. 하지만 스팟 가격이 입찰 가격 아래로 하락하면 스팟 인스턴스 요청이 다시 열리고 Amazon EC2가 새로운 스팟 인스턴스를 시작합니다.

입찰 상태를 통해 스팟 인스턴스 요청의 상태와 시작된 스팟 인스턴스의 상태를 추적할 수 있습니다. 자세한 내용은 [스팟 입찰 상태 \(p. 238\)](#) 섹션을 참조하십시오.

스팟 인스턴스의 지속 시간 지정

Amazon EC2는 스팟 가격이 변하면 지정된 지속 시간(스팟 블록이라고도 함)을 지닌 스팟 인스턴스를 종료하지 않습니다. 이로 인해 배치성 프로세스, 인코딩 및 렌더링, 모델링 및 분석, 지속적 통합 작업처럼 완료하는데 한정된 시간이 소요되는 작업에 이상적입니다.

1, 2, 3, 4, 5, 또는 6시간의 지속 시간을 지정할 수 있습니다. 지불하는 요금은 지정된 지속 시간에 따라 변합니다. 1시간 또는 6시간의 지속 시간에 대한 현행 요금을 보려면, [스팟 인스턴스 요금](#)을 참조하십시오. 이 요금표를 이용해 2, 3, 4 및 5시간의 지속 시간에 대한 비용을 추산할 수 있습니다. 지속 시간을 지닌 요청이 이행되면, 스팟 인스턴스에 대한 가격이 고정되고 이 가격은 인스턴스가 종료될 때까지 유효합니다. 인스턴스를 실행하는 시간별로, 혹은 부분 시간에 대해 이 가격으로 요금이 청구됩니다. 단 부분 인스턴스 시간도 전체 시간으로 계산됩니다.

스팟 요청에서 지속 시간을 지정하면, 각 스팟 인스턴스에 대한 지속 시간은 인스턴스가 인스턴스 ID를 받자마자 시작됩니다. 스팟 인스턴스는 사용자가 종료할 때까지 또는 지속 시간이 끝날 때까지 실행됩니다. 지속 시간이 끝나는 시점에 Amazon EC2는 스팟 인스턴스에 종료 표시를 하고 종료 2분 전에 경고하는 스팟 인스턴스 종료 공지를 제공합니다.

콘솔을 사용하여 지속 시간이 지정되어 있는 인스턴스를 시작하려면

알맞은 요청 유형을 선택합니다. 자세한 내용은 [스팟 인스턴스 요청 생성 \(p. 215\)](#) 섹션을 참조하십시오.

AWS CLI를 사용하여 지속 시간이 지정되어 있는 인스턴스를 시작하려면

스팟 인스턴스에 대해 지속 시간을 지정하려면 `request-spot-instances` 명령으로 `--block-duration-minutes` 옵션을 포함시키십시오. 예를 들면, 다음 명령은 2시간 동안 실행되는 스팟 인스턴스를 시작하는 스팟 요청을 생성합니다.

```
aws ec2 request-spot-instances --spot-price "0.050" --instance-count 5 --block-duration-minutes 120 --type "one-time" --launch-specification file://specification.json
```

AWS CLI를 이용해 지정된 지속 시간을 지닌 스팟 인스턴스의 비용을 검색합니다.

`describe-spot-instance-requests` 명령을 사용하여 지정된 지속 시간을 지닌 스팟 인스턴스의 고정 비용을 검색합니다. 해당 정보는 `actualBlockHourlyPrice` 필드에 있습니다.

스팟 인스턴스의 테넌시 지정

스팟 인스턴스를 단일 테넌트 하드웨어에서 실행할 수 있습니다. 전용 스팟 인스턴스는 다른 AWS 계정에 속하는 인스턴스로부터 물리적으로 격리됩니다. 자세한 내용은 [전용 인스턴스 \(p. 257\)](#) 및 [Amazon EC2 전용 인스턴스](#) 제품 페이지를 참조하십시오.

전용 스팟 인스턴스를 실행하려면 다음 중 하나를 수행합니다.

- 스팟 인스턴스 요청을 생성할 경우 `dedicated`의 테넌시를 지정합니다. 자세한 내용은 [스팟 인스턴스 요청 생성 \(p. 215\)](#) 섹션을 참조하십시오.
- `dedicated`의 인스턴스 테넌시를 사용하여 VPC에 스팟 인스턴스를 요청합니다. 자세한 내용은 [전용 인스턴스 테넌시의 VPC 생성하기 \(p. 259\)](#) 섹션을 참조하십시오. `dedicated`의 인스턴스 테넌시를 사용하여 VPC에서 스팟 인스턴스를 요청하는 경우 `default`의 테넌시를 사용해서는 스팟 인스턴스를 요청할 수 없습니다.

다음 인스턴스 유형은 전용 스팟 인스턴스를 지원하지 않습니다.

현재 세대

- c3.8xlarge
- c4.8xlarge
- d2.8xlarge
- g2.8xlarge
- i2.8xlarge
- m4.10xlarge
- m4.16xlarge

- p2.16xlarge
- r3.8xlarge
- r4.16xlarge
- x1.32xlarge

이전 세대

- cc2.8xlarge
- cg1.4xlarge
- cr1.8xlarge
- hi1.4xlarge

스팟 인스턴스 요청 생성

스팟 인스턴스를 요청하는 프로세스는 온디맨드 인스턴스를 시작하는 프로세스와 비슷합니다. 요청을 제출한 후에는 입찰 가격을 포함한 스팟 요청 파라미터를 변경할 수 없습니다.

여러 스팟 인스턴스를 한 번에 요청하는 경우 각 요청 상태를 개별적으로 추적할 수 있도록 Amazon EC2에서 개별 스팟 인스턴스 요청이 생성됩니다. 스팟 요청 추적에 대한 자세한 내용은 [스팟 입찰 상태 \(p. 238\)](#) 섹션을 참조하십시오.

사전 조건

시작하기 전에 입찰 가격, 원하는 스팟 인스턴스 수 및 사용할 인스턴스 유형을 결정합니다. 스팟 가격 추세를 검토하려면 [스팟 인스턴스 요금 기록 \(p. 212\)](#) 섹션을 참조하십시오.

콘솔을 사용하여 스팟 인스턴스 요청을 생성하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Spot Requests]를 선택합니다.
3. 스팟 인스턴스를 처음 사용하는 경우 시작 페이지가 표시되면 [Get started]를 선택합니다. 그렇지 않다면, [Request Spot instances]를 선택합니다.
4. [Find instance types] 페이지에서 다음을 수행하십시오.
 - a. [Request type]의 경우 기본값은 스팟 집합을 사용하여 생성되는 일회성 스팟 요청입니다. 자세한 내용은 [스팟 집합 요청 \(p. 220\)](#) 섹션을 참조하십시오. 스팟 블록을 대신 사용하려면 [Reserve for duration]을 선택합니다.
 - b. [Target capacity]에는 요청할 단위 수를 입력합니다. vCPU, 메모리, 스토리지 같이 애플리케이션 위크로드에 중요한 인스턴스 또는 성능 특성을 선택할 수 있습니다.
 - c. [Spot block] [Reserved duration]에는 작업 완료 소요 시간을 선택합니다.
 - d. [AMI]에 대해서는 AWS가 제공하는 기본 Amazon 머신 이미지(AMI) 중 하나를 선택하거나 [Use custom AMI]를 선택하여 자신의 AMI를 지정합니다.
 - e. [Instance type(s)]에는 [Select]를 선택합니다. 필요한 최소한의 하드웨어 사양(vCPU, 메모리, 스토리지)을 지닌 인스턴스 유형을 선택합니다.
 - f. [Spot fleet] [Allocation strategy]에서는 필요에 맞는 전략을 선택합니다. 자세한 내용은 [스팟 집합 할당 전략 \(p. 208\)](#) 섹션을 참조하십시오.
 - g. [Network]의 경우, 계정에 따라 EC2-Classic과 EC2-VPC 플랫폼을 모두 지원하거나 EC2-VPC 플랫폼만 지원합니다. 계정에서 지원하는 플랫폼을 확인하려면 [지원되는 플랫폼 \(p. 471\)](#) 섹션을 참조하십시오.
 - [Existing VPC] VPC를 선택합니다.
 - [New VPC] Amazon VPC 콘솔로 이동하려면 [Create new VPC]를 선택합니다. 마친 후에 마법사로 돌아와 목록을 새로 고칩니다.

- [EC2-Classic] [EC2-Classic]을 선택합니다.
 - h. (선택 사항) [Availability Zones]의 경우, 기본 설정은 AWS가 스팟 인스턴스에 대한 가용 영역을 선택하도록 하는 것입니다. 특정 가용 영역을 선호한다면 다음과 같이 합니다.
 - [EC2-VPC] 하나 이상의 가용 영역을 선택합니다. 가용 영역에 두 개 이상의 서브넷이 있는 경우 [Subnet]에서 알맞은 서브넷을 선택합니다. 서브넷을 추가하려면 [Create new subnet]을 선택하여 Amazon VPC 콘솔로 이동합니다. 마침 후에 마법사로 돌아와 목록을 새로 고칩니다.
 - [EC2-Classic] [Select specific zone/subnet]을 선택한 다음, 하나 이상의 가용 영역을 선택합니다.
 - i. [Spot fleet] [Maximum price]에는 자동 입찰을 사용하거나 입찰 가격을 지정할 수 있습니다. 입찰 가격이 자신이 선택한 인스턴스 유형에 대한 스팟 가격보다 낮으면 스팟 인스턴스가 시작되지 않습니다.
 - j. [Next]를 선택합니다.
5. [Configure (세부 정보 구성)] 페이지에서 다음을 수행합니다.
- a. (선택 사항) 추가 스토리지가 필요한 경우, 인스턴스 유형에 따라 인스턴스 스토어 볼륨이나 EBS 볼륨을 지정할 수 있습니다.
 - b. (선택 사항) 전용 스팟 인스턴스를 실행해야 하는 경우, [Tenancy]에 [Dedicated]를 선택합니다.
 - c. (선택 사항) 인스턴스에 연결해야 하는 경우 [Key pair name]을 사용하여 키 페어를 지정합니다.
 - d. (선택 사항) IAM 역할로 스팟 인스턴스를 시작해야 하는 경우, [IAM instance profile]을 사용하여 역할을 지정합니다.
 - e. (선택 사항) 실행할 시작 스크립트가 있는 경우 [User data]를 사용하여 스크립트를 지정합니다.
 - f. [Security groups]에서 하나 이상의 보안 그룹을 선택합니다.
 - g. [EC2-VPC] VPC에서 인스턴스에 연결할 필요가 있는 경우 [Auto-assign Public IP]를 사용하면 됩니다.
 - h. 기본적으로 요청은 이행되거나 사용자가 취소할 때까지 효력을 유지합니다. 특정 기간 동안에만 유 효한 요청을 생성하려면 [Request valid from] 및 [Request valid to]를 편집합니다.
 - i. [Spot fleet] 기본적으로 요청 만료 시 스팟 인스턴스를 종료합니다. 요청 만료 후에도 계속 실행하려면 [Terminate instances at expiration]을 선택 취소합니다.
 - j. [Review]를 선택합니다.
6. [Review] 페이지에서 시작 구성을 확인합니다. 변경하려면 [Previous]를 선택합니다. AWS CLI용 시작 구성의 사본을 다운로드하려면 [JSON config]를 선택합니다. 준비가 완료되면 [Launch]를 선택합니다.
7. 확인 페이지에서 [OK]를 선택합니다.

[Spot fleet] 요청 유형은 `fleet`입니다. 요청이 이행되면 `instance` 유형의 요청이 추가되며, 그 상태는 `active`이고 상황은 `fulfilled`입니다.

[Spot block] 요청 유형은 `block`이고 초기 상태는 `open`입니다. 요청이 이행되면 상태가 `active`이고 상황은 `fulfilled`입니다.

AWS CLI를 사용하여 스팟 인스턴스 요청을 생성하려면

다음 `request-spot-instances` 명령을 사용하여 일회성 요청을 생성합니다.

```
aws ec2 request-spot-instances --spot-price "0.05" --instance-count 5 --type "one-time" --launch-specification file://specification.json
```

다음 `request-spot-instances` 명령을 사용하여 영구 요청을 생성합니다.

```
aws ec2 request-spot-instances --spot-price "0.05" --instance-count 5 --type "persistent" --launch-specification file://specification.json
```

시작 사양 파일에 대한 예시는 [스팟 요청 예시 시작 사양 \(p. 218\)](#) 섹션을 참조하십시오.

스팟 가격이 입찰 가격보다 낮으면 Amazon EC2가 스팟 인스턴스를 시작합니다. 스팟 인스턴스는 중단될 때 까지 또는 사용자가 직접 종료할 때까지 실행됩니다. 다음 [describe-spot-instance-requests](#) 명령을 사용하여 스팟 인스턴스 요청을 모니터링합니다.

```
aws ec2 describe-spot-instance-requests --spot-instance-request-ids sir-08b93456
```

실행 중인 스팟 인스턴스 찾기

스팟 가격이 입찰 가격보다 낮으면 Amazon EC2가 스팟 인스턴스를 시작합니다. 스팟 인스턴스는 입찰 가격이 스팟 가격보다 더 이상 높지 않거나 사용자가 직접 종료할 때까지 실행됩니다. 입찰 가격이 스팟 가격과 정확히 같은 경우 수요에 따라 스팟 인스턴스가 계속 실행될 수 있습니다.

콘솔을 사용하여 실행 중인 스팟 인스턴스를 찾으려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Spot Requests]를 선택합니다.

스팟 인스턴스 요청과 스팟 집합 요청을 모두 볼 수 있습니다. 스팟 인스턴스 요청이 이행된 경우 [Capacity]는 스팟 인스턴스의 ID입니다. 스팟 집합의 경우, [Capacity]는 요청된 용량 중 충족된 용량을 나타냅니다. 스팟 집합에서 인스턴스의 ID를 보려면 확장 화살표를 선택하거나 집합을 선택한 후 [Instances] 탭을 선택합니다.

3. 또는 탐색 창에서 [Instances]를 선택합니다. 오른쪽 위에서 [Show/Hide] 아이콘을 선택한 다음 [Lifecycle]을 선택합니다. 각 인스턴스에 대해 [Lifecycle]은 `normal`, `spot` 또는 `scheduled`입니다.

AWS CLI를 사용하여 실행 중인 스팟 인스턴스를 찾으려면

스팟 인스턴스를 나열하려면 다음과 같이 `--query` 옵션으로 [describe-spot-instance-requests](#) 명령을 사용하십시오.

```
aws ec2 describe-spot-instance-requests --query SpotInstanceRequests[*].{ID:InstanceId}
```

다음은 예제 출력입니다.

```
[  
  {  
    "ID": "i-1234567890abcdef0"  
  },  
  {  
    "ID": "i-0598c7d356eba48d7"  
  }  
]
```

아니면 다음과 같이 `--filters` 옵션으로 [describe-instances](#) 명령을 사용해 스팟 인스턴스를 나열할 수도 있습니다.

```
aws ec2 describe-instances --filters "Name=instance-lifecycle,Values=spot"
```

스팟 인스턴스 요청 태그 지정

스팟 인스턴스 요청을 쉽게 분류하고 관리할 수 있도록 원하는 메타데이터로 태그를 지정할 수 있습니다. 다른 Amazon EC2 리소스에 태그를 지정하는 것과 동일한 방식으로 스팟 인스턴스 요청에 태그를 지정합니다. 자세한 내용은 [Amazon EC2 리소스에 태그 지정 \(p. 681\)](#) 섹션을 참조하십시오.

요청을 만든 후 요청에 태그를 할당할 수 있습니다.

스팟 인스턴스 요청에 대해 생성하는 태그는 요청에만 적용됩니다. 이러한 태그는 스팟 서비스가 요청을 이행하기 위해 시작하는 스팟 인스턴스에 자동으로 추가되지 않습니다. 스팟 인스턴스가 시작된 후 스팟 인스턴스에 직접 태그를 추가해야 합니다.

AWS CLI를 사용해 스팟 인스턴스 요청 또는 스팟 인스턴스에 태그를 추가하려면

다음 [create-tags](#) 명령을 사용해 리소스에 태그를 지정하십시오.

```
aws ec2 create-tags --resources sir-08b93456 i-1234567890abcdef0 --tags  
Key=purpose,Value=test
```

스팟 인스턴스 요청 취소

이제 필요 없는 스팟 요청을 취소할 수 있습니다. `open` 또는 `active` 상태인 스팟 인스턴스 요청만 취소할 수 있습니다. 요청이 아직 이행되지 않았고 인스턴스가 시작되지 않았을 때 스팟 요청은 `open` 상태입니다. 요청이 이행되었고 결과적으로 스팟 인스턴스가 시작되었을 때 스팟 요청은 `active` 상태입니다. 스팟 요청이 `active`이고 실행 중인 스팟 인스턴스가 연결되어 있을 때 요청을 취소하면 인스턴스가 종료되지 않습니다. 실행 중인 스팟 인스턴스를 수동으로 종료해야 합니다.

스팟 요청이 영구 스팟 요청인 경우 새로운 스팟 인스턴스를 시작할 수 있도록 요청이 `open` 상태로 돌아갑니다. 영구 스팟 요청을 취소하고 스팟 인스턴스를 종료하려면 스팟 요청을 먼저 취소한 다음 스팟 인스턴스를 종료해야 합니다. 그렇지 않으면 스팟 요청이 새로운 인스턴스를 시작할 수 있습니다.

콘솔을 사용하여 스팟 인스턴스 요청을 취소하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Spot Requests]를 선택한 다음 스팟 요청을 선택합니다.
3. [Actions]를 선택한 다음, [Cancel spot request]를 선택합니다.
4. (선택 사항) 연결된 스팟 인스턴스가 완료되면 인스턴스를 종료할 수 있습니다. 탐색 창에서 [Instances]를 선택하고 인스턴스를 선택한 다음 [Actions]를 선택하고 [Instance State]를 선택한 후 [Stop]을 선택합니다.

AWS CLI를 사용해 스팟 인스턴스 요청을 취소하려면

다음 [cancel-spot-instance-requests](#) 명령을 사용하여 지정한 스팟 요청을 취소하십시오.

```
aws ec2 cancel-spot-instance-requests --spot-instance-request-ids sir-08b93456
```

연결된 스팟 인스턴스 작업이 완료되면, 다음과 같이 [terminate-instances](#) 명령을 사용하여 해당 인스턴스를 수동으로 종료할 수 있습니다.

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0 i-0598c7d356eba48d7
```

스팟 요청 예시 시작 사양

다음 예는 [request-spot-fleet](#) 명령과 함께 사용하여 스팟 인스턴스 요청을 생성할 수 있는 시작 구성을 보여줍니다. 자세한 내용은 [스팟 인스턴스 요청 생성 \(p. 215\)](#) 섹션을 참조하십시오.

1. [스팟 인스턴스 시작 \(p. 219\)](#)
2. [지정된 가용 영역에서 스팟 인스턴스 시작 \(p. 219\)](#)
3. [지정된 서브넷에서 스팟 인스턴스 시작 \(p. 219\)](#)
4. [전용 스팟 인스턴스 시작 \(p. 220\)](#)

예 1: 스팟 인스턴스 시작

다음 예는 가용 영역 또는 서브넷을 포함하지 않습니다. Amazon EC2는 사용자를 위한 가용 영역을 선택합니다. 해당 계정에서 EC2-VPC만 지원할 경우 Amazon EC2는 선택된 가용 영역의 기본 서브넷에 있는 인스턴스를 시작합니다. 해당 계정에서 EC2-Classic을 지원할 경우 Amazon EC2는 선택된 가용 영역에서 EC2-Classic의 인스턴스를 시작합니다.

```
{  
    "ImageId": "ami-1a2b3c4d",  
    "KeyName": "my-key-pair",  
    "SecurityGroupIds": [ "sg-1a2b3c4d" ],  
    "InstanceType": "m3.medium",  
    "IamInstanceProfile": {  
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"  
    }  
}
```

ID 또는 이름으로(SecurityGroups 필드를 이용해) EC2-Classic에 보안 그룹을 지정할 수 있다는 것에 유의하십시오. ID에 의해 EC2-VPC에 대한 보안 그룹을 반드시 지정해야 합니다.

예 2: 지정된 가용 영역에서 스팟 인스턴스 시작

다음 예는 가용 영역을 포함합니다. 해당 계정에서 EC2-VPC만 지원할 경우 Amazon EC2는 지정된 가용 영역의 기본 서브넷에 있는 인스턴스를 시작합니다. 해당 계정에서 EC2-Classic을 지원할 경우 Amazon EC2는 지정된 가용 영역에서 EC2-Classic의 인스턴스를 시작합니다.

```
{  
    "ImageId": "ami-1a2b3c4d",  
    "KeyName": "my-key-pair",  
    "SecurityGroupIds": [ "sg-1a2b3c4d" ],  
    "InstanceType": "m3.medium",  
    "Placement": {  
        "AvailabilityZone": "us-west-2a"  
    },  
    "IamInstanceProfile": {  
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"  
    }  
}
```

예 3: 지정된 서브넷에서 스팟 인스턴스 시작

다음 예는 서브넷을 포함합니다. Amazon EC2는 지정된 서브넷에서 인스턴스를 시작합니다. VPC가 기본이 아닌 VPC인 경우, 인스턴스는 기본적으로 퍼블릭 IPv4 주소를 받지 않습니다.

```
{  
    "ImageId": "ami-1a2b3c4d",  
    "SecurityGroupIds": [ "sg-1a2b3c4d" ],  
    "InstanceType": "m3.medium",  
    "SubnetId": "subnet-1a2b3c4d",  
    "IamInstanceProfile": {  
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"  
    }  
}
```

기본이 아닌 VPC에서 인스턴스에 퍼블릭 IPv4 주소를 할당하려면 다음 예시와 같이 AssociatePublicIpAddress 필드를 지정하십시오. 네트워크 인터페이스를 지정할 때는 예 3과 같은 SubnetId 및 SecurityGroupIds 필드를 사용하는 대신 네트워크 인터페이스를 사용해 서브넷 ID 및 보안 그룹 ID를 반드시 포함시켜야 한다는 것에 유의하십시오.

```
{
```

```
"ImageId": "ami-1a2b3c4d",
"KeyName": "my-key-pair",
"InstanceType": "m3.medium",
"NetworkInterfaces": [
    {
        "DeviceIndex": 0,
        "SubnetId": "subnet-1a2b3c4d",
        "Groups": [ "sg-1a2b3c4d" ],
        "AssociatePublicIpAddress": true
    }
],
"IamInstanceProfile": {
    "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
}
```

예 4: 전용 스팟 인스턴스 시작

다음 예제에서는 `dedicated`의 테넌시를 사용하여 스팟 인스턴스를 요청합니다. 전용 스팟 인스턴스가 VPC에서 시작되어야 합니다.

```
{
    "ImageId": "ami-1a2b3c4d",
    "KeyName": "my-key-pair",
    "SecurityGroupIds": [ "sg-1a2b3c4d" ],
    "InstanceType": "c3.8xlarge",
    "SubnetId": "subnet-1a2b3c4d",
    "Placement": {
        "Tenancy": "dedicated"
    }
}
```

스팟 집합 요청

스팟 집합을 사용하려면 대상 용량, 인스턴스에 대한 하나 이상의 시작 사양 및 지불하려는 입찰 가격을 포함한 스팟 집합 요청을 생성합니다. Amazon EC2가 스팟 가격 변경에 따라 스팟 집합의 대상 용량을 유지하도록 시도합니다. 자세한 내용은 [스팟 집합의 작동 방식 \(p. 208\)](#) 섹션을 참조하십시오.

원하는 용량을 위한 일회성 `request` 제출을 위해 스팟 집합을 만들거나 시간의 경과에 따라 스팟 집합에 목표 용량을 `maintain`할 것을 요구할 수 있습니다. 두 가지 유형의 요청 모두 스팟 집합의 할당 전략에 따른 이익을 볼 수 있습니다.

목표 용량을 `request`하면 스팟 집합이 필요한 입찰을 하지만 용량이 감소되는 경우 스팟 인스턴스를 보충하려고 시도하지는 않습니다. 용량을 사용할 수 없는 경우 스팟 집합은 대체 스팟 풀에서 입찰을 제출하지 않습니다.

목표 용량을 `maintain`하려는 경우 스팟 집합은 이 목표 용량을 충족시키기 위해 필요한 입찰을 하며 중단되는 인스턴스를 모두 자동으로 보충합니다. 기본적으로, 스팟 집합은 요청된 목표 용량을 `maintain`하도록 설정됩니다.

일회성 `request`가 일단 제출되고 나면 이 요청의 목표 용량을 수정할 수 없습니다. 목표 용량을 변경하려면 요청을 취소하고 새 요청을 제출합니다.

스팟 집합 요청은 요청이 만료되거나 사용자가 요청을 취소할 때까지 활성 상태로 유지됩니다. 스팟 집합 요청을 취소할 경우에는, 스팟 집합 요청을 취소하면 스팟 집합에 속한 스팟 인스턴스가 종료될지 여부를 지정할 수 있습니다.

각 시작 사양에는 AMI, 인스턴스 유형, 서브넷이나 가용 영역, 하나 이상의 보안 그룹과 같이 Amazon EC2가 인스턴스를 시작하는 데 필요로 하는 정보가 포함됩니다.

목차

- [스팟 집합 요청 상태 \(p. 221\)](#)
- [스팟 집합 사전 요구사항 \(p. 221\)](#)
- [스팟 집합 및 IAM 사용자 \(p. 222\)](#)
- [스팟 집합 상태 확인 \(p. 222\)](#)
- [스팟 집합 요청 계획 \(p. 223\)](#)
- [스팟 집합 요청 생성 \(p. 223\)](#)
- [스팟 집합 모니터링 \(p. 225\)](#)
- [스팟 집합 요청 수정 \(p. 225\)](#)
- [스팟 집합 요청 취소 \(p. 226\)](#)
- [스팟 집합 예제 구성 \(p. 227\)](#)

스팟 집합 요청 상태

스팟 집합 요청은 다음 상태 중 하나일 수 있습니다.

- `submitted` - 스팟 집합 요청이 평가되고 Amazon EC2가 대상 스팟 인스턴스의 수를 시작하도록 준비합니다.
- `active` - 스팟 집합이 검증되었으며, Amazon EC2가 실행 중인 대상 스팟 인스턴스의 수 유지를 시도합니다. 그 요청은 수정 또는 취소될 때까지 계속 이 상태로 유지됩니다.
- `modifying`-스팟 집합 요청이 수정되고 있습니다. 그 요청은 수정이 완전히 처리될 때까지 또는 스팟 집합이 취소될 때까지 계속 이 상태로 유지됩니다. 일회성 `request`는 수정할 수 없으며, 이 상태가 이런 스팟 요청에 적용되지 않습니다.
- `cancelled_running` - 스팟 집합이 취소되고 추가 스팟 인스턴스가 시작되지 않지만, 기존 스팟 인스턴스는 중단 또는 종료될 때까지 계속 실행됩니다. 그 요청은 모든 인스턴스가 중단 또는 종료될 때까지 계속 이 상태로 유지됩니다.
- `cancelled_terminating` - 스팟 집합이 취소되고 해당 스팟 인스턴스가 종료됩니다. 그 요청은 모든 인스턴스가 종료될 때까지 계속 이 상태로 유지됩니다.
- `cancelled` - 스팟 집합이 취소되고 실행 중인 스팟 인스턴스가 없습니다. 스팟 집합 요청은 인스턴스 종료 2일 후에 삭제됩니다.

다음 그림은 요청 상태 간의 전환을 나타냅니다. 스팟 집합 한계를 초과하면 즉시 요청이 취소된다는 것에 유의하십시오.

스팟 집합 사전 요구사항

AWS Management Console을 사용하여 스팟 집합을 만들면 사용자를 대신하여 인스턴스에 대해 입찰하고 인스턴스를 시작하고 종료할 수 있도록 스팟 집합 권한을 부여하는 `aws-ec2-spot-fleet-role`이라는 역할이 생성되어 스팟 집합 요청에 지정됩니다. AWS CLI 또는 API를 사용하여 스팟 집합을 만들면, 이 역할이 존재하는 경우 이 역할을 사용하거나, 다음과 같이 이 용도로 사용할 고유의 역할을 수동으로 만들 수 있습니다.

AmazonEC2SpotFleetRole 정책으로 IAM 역할을 수동으로 만들려면

1. <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 [Roles]를 선택합니다.
3. [Create New Role(새 역할 만들기)]을 선택합니다.
4. [Set Role Name] 페이지에서 역할 이름을 입력한 다음, [Next Step]을 선택합니다.
5. [Select Role Type] 페이지에서 [Amazon EC2 Spot Fleet Role] 옆의 [Select]를 선택합니다.
6. [Attach Policy] 페이지에서 `AmazonEC2SpotFleetRole` 정책을 선택한 후 [Next Step]을 선택합니다.
7. [Review] 페이지에서 [Create Role]을 선택합니다.

스팟 집합 및 IAM 사용자

IAM 사용자가 스팟 집합을 만들거나 관리할 경우 다음과 같이 이들 사용자에게 필요한 권한을 부여해야 합니다.

스팟 집합에 대한 IAM 사용자 권한을 부여하려면

1. <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 [Policies]를 선택한 후 [Create Policy]를 선택합니다.
3. [Create Policy] 페이지에서 [Create Your Own Policy] 옆의 [Select]를 선택합니다.
4. [Review Policy] 페이지에서 정책 이름을 입력한 후 다음 텍스트를 [Policy Document] 영역으로 복사합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:*"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iam:PassRole",  
                "iam>ListRoles",  
                "iam>ListInstanceProfiles"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

ec2:*에서는 IAM 사용자가 모든 Amazon EC2 API 작업을 호출할 수 있습니다. 사용자를 특정 API 작업으로 제한하려면 해당 작업을 대신 지정하십시오.

iam:PassRole은 스팟 집합 요청에서 사용자가 스팟 집합 역할을 지정할 수 있는 작업입니다.
iam>ListRoles 작업에서는 사용자가 기존 역할을 열거할 수 있습니다. iam>ListInstanceProfiles 작업에서는 사용자가 기존 인스턴스 프로파일을 열거할 수 있습니다. Amazon EC2 콘솔은 iam>ListRoles 작업을 사용해 IAM 역할 목록을 채우고, iam>ListInstanceProfiles 작업을 사용해 IAM 인스턴스 프로파일 목록을 채웁니다. 콘솔을 사용하여 사용자가 역할 또는 인스턴스 프로파일을 생성할 수 있도록 설정하려면 iam>CreateRole, iam>CreateInstanceProfile 및 iam>AddRoleToInstanceProfile 작업을 추가해야 합니다.

5. [Create Policy]를 선택합니다.
6. 탐색 창에서 [Users]를 선택한 다음 스팟 집합 요청을 제출할 사용자를 선택합니다.
7. [Permissions] 탭에서 [Add permissions]를 선택합니다.
8. [Attach existing policies directly]를 선택합니다. 위에서 생성한 정책을 선택하고 [Next: Review]와 [Add permissions]를 차례로 선택합니다.

스팟 집합 상태 확인

스팟 집합은 2분마다 집합에 있는 스팟 인스턴스의 상태를 확인합니다. 인스턴스의 상태는 healthy 또는 unhealthy입니다. 스팟 집합은 Amazon EC2에서 제공하는 상태 확인을 사용하여 인스턴스의 상태를 판단합니다. 세 번의 연속 상태 확인에서 인스턴스 상태 또는 시스템 상태가 impaired이면, 해당 인스턴스의 상태

는 `unhealthy`입니다. 그렇지 않으면 상태는 `healthy`입니다. 자세한 내용은 [인스턴스 상태 확인 \(p. 340\)](#) 섹션을 참조하십시오.

비정상 인스턴스를 교체하도록 스팟 집합을 구성할 수 있습니다. 상태 확인 교체를 활성화하면 상태가 `unhealthy`로 보고된 인스턴스가 교체됩니다. 스팟 집합은 비정상 인스턴스가 교체되는 동안 최대 몇 분간 대상 용량보다 적어질 수 있습니다.

요구 사항

- 상태 확인 교체는 대상 용량을 유지하는 스팟 집합에서만 지원되며, 1회용 스팟 집합에서는 지원되지 않습니다.
- 비정상 인스턴스를 만들 경우에만 이를 교체하도록 스팟 집합을 구성할 수 있습니다.
- IAM 사용자는 `ec2:DescribeInstanceStatus` 작업을 호출할 권한이 있는 경우에만 상태 확인 교체를 사용 할 수 있습니다.

스팟 집합 요청 계획

스팟 집합 요청을 생성하려면 그 전에 먼저 [스팟 모범 사례](#)를 살펴보는 것이 좋습니다. 특히 스팟 집합 요청을 계획하여 원하는 인스턴스 유형을 최저 가격으로 프로비저닝하려면 이러한 모범 사례가 필요합니다. 또한, 다음을 수행하는 것이 좋습니다.

- 원하는 목표 용량을 위한 일회성 `request`를 제출하는 스팟 집합을 만들지, 시간의 경과에 따라 목표 용량을 `maintain`할 스팟 집합을 만들지 결정합니다.
- 인스턴스 유형을 결정하고 애플리케이션 요구를 만족합니다.
- 스팟 집합 요청의 목표 용량을 결정합니다. 인스턴스 또는 사용자 지정 단위에서 목표 용량을 설정할 수 있습니다. 자세한 내용은 [스팟 집합 인스턴스 가중치 부여 \(p. 209\)](#) 섹션을 참조하십시오.
- 인스턴스 시간당 입찰 가격을 결정합니다. 더 낮은 가격으로 입찰하면 비용을 추가로 줄일 수 있고, 더 높은 가격으로 입찰하면 중단 가능성을 줄일 수 있습니다.
- 인스턴스 가중치를 사용하는 경우에는 단위당 입찰 가격을 결정합니다. 단위당 입찰 가격을 계산하려면 인스턴스 시간당 입찰 가격을 이 인스턴스가 나타내는 단위 수(또는 가중치)로 나눕니다. (인스턴스 가중치를 사용하지 않는 경우 단위당 기본 입찰 가격은 인스턴스 시간당 입찰 가격입니다.)
- 스팟 집합 요청에 대해 가능한 옵션을 살펴봅니다. 자세한 내용은 AWS Command Line Interface Reference의 `request-spot-fleet` 명령을 참조하십시오. 추가 예제는 다음([스팟 집합 예제 구성 \(p. 227\)](#))을 참조하십시오.

스팟 집합 요청 생성

스팟 집합 요청을 생성할 때는 인스턴스 유형, 스팟 가격과 같은, 시작할 스팟 인스턴스에 대한 정보를 지정해야 합니다.

콘솔을 사용하여 스팟 집합 요청을 생성하려면

- <https://console.aws.amazon.com/ec2spot>에서 스팟 콘솔을 엽니다.
- 스팟을 처음 사용하는 경우 시작 페이지가 표시되면 [Get started]를 선택합니다. 그렇지 않다면, [Request Spot instances]를 선택합니다.
- [Find instance types] 페이지에서 다음을 수행하십시오.
 - [Request type]에는 [Request] 또는 [Request and Maintain]을 선택합니다.
 - [Target capacity]에는 요청할 단위 수를 입력합니다. vCPU, 메모리, 스토리지 같이 애플리케이션 위크로드에 중요한 인스턴스 또는 성능 특성을 선택할 수 있습니다.
 - [AMI]에 대해서는 AWS가 제공하는 기본 AMI(Amazon 머신 이미지) 중 하나를 선택하거나 [Use custom AMI]를 선택하여 사용자 커뮤니티의 AMI, AWS Marketplace의 AMI 또는 자체 AMI를 사용합니다.

- d. [Instance type(s)]에는 [Select]를 선택합니다. 필요한 최소한의 하드웨어 사양(vCPU, 메모리, 스토리지)을 지닌 인스턴스 유형을 선택합니다.
 - e. [Allocation strategy]에서는 필요에 맞는 전략을 선택합니다. 자세한 내용은 [스팟 집합 할당 전략 \(p. 208\)](#) 섹션을 참조하십시오.
 - f. [Network]의 경우, 계정에 따라 EC2-Classic과 EC2-VPC 플랫폼을 모두 지원하거나 EC2-VPC 플랫폼만 지원합니다. 계정에서 지원하는 플랫폼을 확인하려면 [지원되는 플랫폼 \(p. 471\)](#) 섹션을 참조하십시오.
 - [Existing VPC] VPC를 선택합니다.
 - [New VPC] Amazon VPC 콘솔로 이동하려면 [Create new VPC]를 선택합니다. 마친 후에 마법사로 돌아와 목록을 새로 고칩니다.
 - [EC2-Classic] [EC2-Classic]을 선택합니다.
 - g. (선택 사항) [Availability Zones]의 경우, 기본 설정은 AWS가 스팟 인스턴스에 대한 가용 영역을 선택하도록 하는 것입니다. 특정 가용 영역을 선호한다면 다음과 같이 합니다.
 - [EC2-VPC] 하나 이상의 가용 영역을 선택합니다. 가용 영역에 두 개 이상의 서브넷이 있는 경우 [Subnet]에서 알맞은 서브넷을 선택합니다. 서브넷을 추가하려면 [Create new subnet]을 선택하여 Amazon VPC 콘솔로 이동합니다. 마친 후에 마법사로 돌아와 목록을 새로 고칩니다.
 - [EC2-Classic] [Select specific zone/subnet]을 선택한 다음, 하나 이상의 가용 영역을 선택합니다.
 - h. [Maximum price]에는 자동 입찰을 사용하거나 입찰 가격을 지정할 수 있습니다. 입찰 가격이 자신이 선택한 인스턴스 유형에 대한 스팟 가격보다 낮으면 스팟 인스턴스가 시작되지 않습니다.
 - i. [Next]를 선택합니다.
4. [Configure (세부 정보 구성)] 페이지에서 다음을 수행합니다.
 - a. (선택 사항) [Request and Maintain] 스팟 집합에서 비정상 인스턴스를 교체하려면 [Replace unhealthy instances]를 선택합니다.
 - b. (선택 사항) 실행할 시작 스크립트가 있는 경우 [User data]를 사용하여 스크립트를 지정합니다.
 - c. (선택 사항) 인스턴스에 연결해야 하는 경우 [Key pair name]을 사용하여 키 페어를 지정합니다.
 - d. (선택 사항) IAM 역할로 스팟 인스턴스를 시작해야 하는 경우, [IAM instance profile]을 사용하여 역할을 지정합니다.
 - e. [Security groups]에서 하나 이상의 보안 그룹을 선택합니다.
 - f. [EC2-VPC] VPC에서 인스턴스에 연결할 필요가 있는 경우 [Auto-assign IPv4 Public IP]에 대해 [Enable]을 선택합니다.
 - g. 기본적으로 요청은 이행되거나 사용자가 취소할 때까지 효력을 유지합니다. 특정 기간 동안에만 유효한 요청을 생성하려면 [Request valid from] 및 [Request valid to]를 편집합니다.
 - h. (옵션) 기본적으로 요청 만료 시 스팟 인스턴스를 종료합니다. 요청 만료 후에도 계속 실행하려면 [Terminate instances at expiration]을 선택 취소합니다.
 - i. [Review]를 선택합니다.
 5. [Review] 페이지에서 시작 구성을 확인합니다. 변경하려면 [Previous]를 선택합니다. AWS CLI용 시작 구성의 사본을 다운로드하려면 [JSON config]를 선택합니다. 준비가 완료되면 [Launch]를 선택합니다.
 6. 확인 페이지에서 [OK]를 선택합니다. 요청 형식은 `fleet`입니다. 요청이 이행되면 `instance` 유형의 요청이 추가되며, 그 상태는 `active`이고 상황은 `fulfilled`입니다.

AWS CLI를 사용하여 스팟 집합 요청을 생성하려면

다음 `request-spot-fleet` 명령을 사용하여 스팟 집합 요청을 생성합니다.

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

구성 파일에 대한 예시는 [스팟 집합 예제 구성 \(p. 227\)](#) 섹션을 참조하십시오.

다음은 예제 출력입니다.

```
{  
    "SpotFleetRequestId": "sfr-73fb2ce-aa30-494c-8788-1cee4EXAMPLE"  
}
```

스팟 집합 모니터링

스팟 가격이 입찰 가격보다 낮으면 스팟 집합이 스팟 인스턴스를 시작합니다. 스팟 인스턴스는 입찰 가격이 스팟 가격보다 더 이상 높지 않거나 사용자가 직접 종료할 때까지 실행됩니다.

콘솔을 사용하여 스팟 집합을 모니터링하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Spot Requests]를 선택합니다.
3. 스팟 집합 요청을 선택합니다. 구성 세부 정보는 [Description] 탭에서 얻을 수 있습니다.
4. 스팟 집합에 대한 스팟 인스턴스를 나열하려면, [Instances] 탭을 선택합니다.
5. 스팟 집합에 대한 이력을 보려면, [History] 탭을 선택합니다.

AWS CLI를 사용하여 스팟 집합을 모니터링하려면

다음 `describe-spot-fleet-requests` 명령을 사용하여 스팟 집합 요청을 설명합니다.

```
aws ec2 describe-spot-fleet-requests
```

다음 `describe-spot-fleet-instances` 명령을 사용하여 지정한 스팟 집합에 대한 스팟 인스턴스를 설명합니다.

```
aws ec2 describe-spot-fleet-instances --spot-fleet-request-id sfr-73fb2ce-  
aa30-494c-8788-1cee4EXAMPLE
```

다음 `describe-spot-fleet-request-history` 명령을 사용하여 지정한 스팟 집합 요청에 대한 기록을 설명합니다.

```
aws ec2 describe-spot-fleet-request-history --spot-fleet-request-id sfr-73fb2ce-  
aa30-494c-8788-1cee4EXAMPLE --start-time 2015-05-18T00:00:00Z
```

스팟 집합 요청 수정

다음 작업을 완료하기 위해 활성 스팟 집합 요청을 수정할 수 있습니다.

- 목표 용량 증가
- 목표 용량 감소

Note

1회용 스팟 집합 요청은 수정할 수 없습니다.

목표 용량을 증가하면 스팟 집합이 스팟 집합 요청에 대한 할당 전략에 따라 추가적인 스팟 인스턴스를 시작합니다. 할당 전략이 `lowestPrice`인 경우, 스팟 집합은 스팟 집합 요청에 있는 최저 가격의 스팟 인스턴스 풀에서 인스턴스를 시작합니다. 할당 전략이 `diversified`인 경우, 스팟 집합은 스팟 집합 요청에서 풀 전체에 걸쳐 인스턴스를 배포합니다.

목표 용량을 감소하면 스팟 집합이 새 목표 용량을 초과하는 모든 공개 입찰을 취소합니다. 스팟 집합은 집합의 크기가 새 목표 용량에 도달할 때까지 스팟 인스턴스를 종료하도록 요청할 수 있습니다. 할당 전

략이 `lowestPrice`인 경우, 스팟 집합은 단위당 최고 가격을 지닌 인스턴스를 종료합니다. 할당 전략이 `diversified`인 경우, 스팟 집합은 풀 전체에 걸쳐 인스턴스를 종료합니다. 또는 스팟 집합이 집합을 현재 크기로 유지하되, 중단되거나 수동으로 종료하는 스팟 인스턴스는 교체하지 않도록 요청할 수 있습니다.

목표 용량이 감소하여 스팟 집합이 인스턴스를 종료할 때 해당 인스턴스는 스팟 인스턴스 종료 공지를 받는다는 점을 유의하십시오.

C 콘솔을 사용하여 스팟 집합 요청을 수정하려면

1. <https://console.aws.amazon.com/ec2spot/home/fleet>에서 스팟 콘솔을 엽니다.
2. 스팟 집합 요청을 선택합니다.
3. [Actions]를 선택한 다음, [Modify target capacity]를 선택합니다.
4. [Modify target capacity]에서 다음 작업을 수행하십시오.
 - a. 새로운 목표 용량을 입력합니다.
 - b. (선택 사양) 목표 용량을 줄이지만 집합은 현재 크기로 유지하고자 한다면, [Terminate instances] 선택을 취소합니다.
 - c. [Submit]를 선택합니다.

AWS CLI를 사용하여 스팟 집합 요청을 수정하려면

다음 `modify-spot-fleet-request` 명령을 사용하여 지정한 스팟 집합 요청의 목표 용량을 업데이트합니다.

```
aws ec2 modify-spot-fleet-request --spot-fleet-request-id sfr-73fdbd2ce-aa30-494c-8788-1cee4EXAMPLE --target-capacity 20
```

다음과 같이 이전 명령을 사용하여 결과적으로 어떤 스팟 인스턴스도 종료하지 않고 지정한 스팟 집합의 목표 용량을 줄입니다.

```
aws ec2 modify-spot-fleet-request --spot-fleet-request-id sfr-73fdbd2ce-aa30-494c-8788-1cee4EXAMPLE --target-capacity 10 --excess-capacity-termination-policy NoTermination
```

스팟 집합 요청 취소

스팟 집합 사용을 마쳤으면 스팟 집합 요청을 취소할 수 있습니다. 이렇게 하면 스팟 집합과 연결된 스팟 요청이 모두 취소되므로, 스팟 집합에 대해 새로운 스팟 인스턴스가 시작되지 않습니다. 스팟 집합이 스팟 인스턴스를 종료할지 여부를 반드시 지정해야 합니다. 인스턴스를 종료하면 스팟 집합 요청은 `cancelled_terminating` 상태가 됩니다. 인스턴스를 종료하지 않으면, 스팟 집합 요청은 `cancelled_running` 상태가 되고 인스턴스는 중단되거나 사용자가 수동으로 종료하지 않는 한 계속 실행됩니다.

C 콘솔을 사용하여 스팟 집합 요청을 취소하려면

1. <https://console.aws.amazon.com/ec2spot/home/fleet>에서 스팟 콘솔을 엽니다.
2. 스팟 집합 요청을 선택합니다.
3. [Actions]를 선택한 다음, [Cancel spot request]를 선택합니다.
4. [Cancel spot request]에서 스팟 집합을 취소하겠다는 것을 확인합니다. 집합을 현재 크기로 유지하려면 [Terminate instances]를 선택 취소합니다. 준비가 완료되면 [Confirm]를 선택합니다.

AWS CLI를 사용하여 스팟 집합 요청을 취소하려면

다음 `cancel-spot-fleet-requests` 명령을 사용하여 지정한 스팟 집합 요청을 취소하고 인스턴스를 종료합니다.

```
aws ec2 cancel-spot-fleet-requests --spot-fleet-request-ids sfr-73fb2ce-  
aa30-494c-8788-1cee4EXAMPLE --terminate-instances
```

다음은 예제 출력입니다.

```
{  
    "SuccessfulFleetRequests": [  
        {  
            "SpotFleetRequestId": "sfr-73fb2ce-aa30-494c-8788-1cee4EXAMPLE",  
            "CurrentSpotFleetRequestState": "cancelled_terminating",  
            "PreviousSpotFleetRequestState": "active"  
        }  
    ],  
    "UnsuccessfulFleetRequests": []  
}
```

다음과 같이 이전 명령을 수정하여 인스턴스 종료 없이 지정된 스팟 집합 요청을 취소합니다.

```
aws ec2 cancel-spot-fleet-requests --spot-fleet-request-ids sfr-73fb2ce-  
aa30-494c-8788-1cee4EXAMPLE --no-terminate-instances
```

다음은 예제 출력입니다.

```
{  
    "SuccessfulFleetRequests": [  
        {  
            "SpotFleetRequestId": "sfr-73fb2ce-aa30-494c-8788-1cee4EXAMPLE",  
            "CurrentSpotFleetRequestState": "cancelled_running",  
            "PreviousSpotFleetRequestState": "active"  
        }  
    ],  
    "UnsuccessfulFleetRequests": []  
}
```

스팟 집합 예제 구성

다음 예는 [request-spot-fleet](#) 명령과 함께 사용하여 스팟 집합 요청을 생성할 수 있는 시작 구성을 보여 줍니다. 자세한 내용은 [스팟 집합 요청 생성 \(p. 223\)](#) 섹션을 참조하십시오.

1. 최저 가격의 가용 영역 또는 리전에 있는 서브넷을 사용하여 스팟 인스턴스 시작 (p. 227)
2. 최저 가격의 가용 영역 또는 지정된 목록에 있는 서브넷을 사용하여 스팟 인스턴스 시작 (p. 228)
3. 지정된 목록에서 최저 가격의 인스턴스 유형을 사용하여 스팟 인스턴스 시작 (p. 229)
4. 요청에 대한 스팟 가격 재정의 (p. 230)
5. 다각화된 할당 전략을 사용하여 스팟 집합 시작 (p. 231)
6. 인스턴스 가중치를 사용하여 스팟 집합 시작 (p. 232)

예 1: 최저 가격의 가용 영역 또는 리전에 있는 서브넷을 사용하여 스팟 인스턴스 시작

다음 예는 가용 영역이나 서브넷이 없는 단일 시작 사양을 지정합니다. 해당 계정에서 EC2-VPC만 지원할 경우 스팟 집합은 기본 서브넷이 있는 최저 가격의 가용 영역에서 인스턴스를 시작합니다. 해당 계정에서 EC2-Classic을 지원할 경우 스팟 집합은 최저 가격의 가용 영역에서 EC2-Classic의 인스턴스를 시작합니다. 지불하는 가격이 해당 요청에 대해 지정된 스팟 가격을 초과하지는 않습니다.

```
{  
    "SpotPrice": "0.07",  
    "TargetCapacity": 20,
```

```
"IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
"LaunchSpecifications": [
    {
        "ImageId": "ami-1a2b3c4d",
        "KeyName": "my-key-pair",
        "SecurityGroups": [
            {
                "GroupId": "sg-1a2b3c4d"
            }
        ],
        "InstanceType": "m3.medium",
        "IamInstanceProfile": {
            "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
        }
    }
]
```

예 2: 최저 가격의 가용 영역 또는 지정된 목록에 있는 서브넷을 사용하여 스팟 인스턴스 시작

다음 예는 가용 영역이나 서브넷은 다르지만 인스턴스 유형과 AMI는 같은 두 개의 시작 사양을 지정합니다.

가용 영역

해당 계정에서 EC2-VPC만 지원할 경우 스팟 집합이 지정한 최저 가격 가용 영역의 기본 서브넷에서 인스턴스를 시작합니다. 해당 계정에서 EC2-Classic을 지원할 경우 스팟 집합이 지정한 최저 가격의 가용 영역에서 인스턴스를 시작합니다.

```
{
    "SpotPrice": "0.07",
    "TargetCapacity": 20,
    "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "KeyName": "my-key-pair",
            "SecurityGroups": [
                {
                    "GroupId": "sg-1a2b3c4d"
                }
            ],
            "InstanceType": "m3.medium",
            "Placement": {
                "AvailabilityZone": "us-west-2a, us-west-2b"
            },
            "IamInstanceProfile": {
                "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
            }
        }
    ]
}
```

서브넷

기본 서브넷이나 기본이 아닌 서브넷을 지정할 수 있으며, 기본이 아닌 서브넷은 기본 VPC 또는 기본이 아닌 VPC의 서브넷일 수 있습니다. 스팟 서비스는 최저 가격의 가용 영역에 있는 서브넷에서 인스턴스를 시작합니다.

스팟 집합 요청에 동일한 가용 영역의 서로 다른 서브넷을 지정할 수는 없습니다.

```
{
    "SpotPrice": "0.07",
```

```
"TargetCapacity": 20,  
"IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",  
"LaunchSpecifications": [  
    {  
        "ImageId": "ami-1a2b3c4d",  
        "KeyName": "my-key-pair",  
        "SecurityGroups": [  
            {  
                "GroupId": "sg-1a2b3c4d"  
            }  
        ],  
        "InstanceType": "m3.medium",  
        "SubnetId": "subnet-a61dafcf, subnet-65ea5f08",  
        "IamInstanceProfile": {  
            "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"  
        }  
    }  
]
```

인스턴스가 기본 VPC로 시작되는 경우, 인스턴스는 기본적으로 퍼블릭 IPv4 주소를 받습니다. 인스턴스가 기본이 아닌 VPC로 시작되는 경우, 인스턴스는 기본적으로 퍼블릭 IPv4 주소를 받지 않습니다. 시작 사양에서 네트워크 인터페이스를 사용하여 기본이 아닌 VPC에서 시작되는 인스턴스에 퍼블릭 IPv4 주소를 할당하십시오. 네트워크 인터페이스를 지정할 때는 네트워크 인터페이스를 사용해 서브넷 ID 및 보안 그룹 ID를 반드시 포함시켜야 한다는 것에 유의하십시오.

```
...  
{  
    "ImageId": "ami-1a2b3c4d",  
    "KeyName": "my-key-pair",  
    "InstanceType": "m3.medium",  
    "NetworkInterfaces": [  
        {  
            "DeviceIndex": 0,  
            "SubnetId": "subnet-1a2b3c4d",  
            "Groups": [ "sg-1a2b3c4d" ],  
            "AssociatePublicIpAddress": true  
        }  
    ],  
    "IamInstanceProfile": {  
        "Arn": "arn:aws:iam::880185128111:instance-profile/my-iam-role"  
    }  
}  
...
```

예 3: 지정된 목록에서 최저 가격의 인스턴스 유형을 사용하여 스팟 인스턴스 시작

다음 예는 인스턴스 유형은 다르지만 AMI와 가용 영역 또는 서브넷은 같은 두 개의 시작 구성을 지정합니다. 스팟 집합이 최저 가격으로 지정된 인스턴스 유형을 사용하여 인스턴스를 시작합니다.

가용 영역

```
{  
    "SpotPrice": "1.00",  
    "TargetCapacity": 20,  
    "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",  
    "LaunchSpecifications": [  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "SecurityGroups": [  
                {  
                    "GroupId": "sg-1a2b3c4d"  
                }  
            ]  
        }  
    ]  
}
```

```
        },
    ],
    "InstanceType": "cc2.8xlarge",
    "Placement": {
        "AvailabilityZone": "us-west-2b"
    }
},
{
    "ImageId": "ami-1a2b3c4d",
    "SecurityGroups": [
        {
            "GroupId": "sg-1a2b3c4d"
        }
    ],
    "InstanceType": "r3.8xlarge",
    "Placement": {
        "AvailabilityZone": "us-west-2b"
    }
}
]
```

서브넷

```
{
    "SpotPrice": "1.00",
    "TargetCapacity": 20,
    "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "SecurityGroups": [
                {
                    "GroupId": "sg-1a2b3c4d"
                }
            ],
            "InstanceType": "cc2.8xlarge",
            "SubnetId": "subnet-1a2b3c4d"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "SecurityGroups": [
                {
                    "GroupId": "sg-1a2b3c4d"
                }
            ],
            "InstanceType": "r3.8xlarge",
            "SubnetId": "subnet-1a2b3c4d"
        }
    ]
}
```

예 4. 요청에 대한 스팟 가격 재정의

개별 시작 사양에 대한 스팟 가격을 지정할 수 있으므로 입찰 프로세스에 대한 추가적인 통제 능력을 가질 수 있습니다. 다음 예에서는 요청에 대한 스팟 가격을 3가지 시작 사양 중 2가지에 대한 개별 스팟 가격으로 재정의합니다. 해당 요청에 대한 스팟 가격은 개별 스팟 가격을 지정하지 않는 시작 사양에 사용됩니다. 스팟 집합이 최저 가격의 인스턴스 유형을 사용하여 인스턴스를 시작합니다.

가용 영역

```
{
    "SpotPrice": "1.00",
```

```
"TargetCapacity": 30,  
"IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",  
"LaunchSpecifications": [  
    {  
        "ImageId": "ami-1a2b3c4d",  
        "InstanceType": "c3.2xlarge",  
        "Placement": {  
            "AvailabilityZone": "us-west-2b"  
        },  
        "SpotPrice": "0.10"  
    },  
    {  
        "ImageId": "ami-1a2b3c4d",  
        "InstanceType": "c3.4xlarge",  
        "Placement": {  
            "AvailabilityZone": "us-west-2b"  
        },  
        "SpotPrice": "0.20"  
    },  
    {  
        "ImageId": "ami-1a2b3c4d",  
        "InstanceType": "c3.8xlarge",  
        "Placement": {  
            "AvailabilityZone": "us-west-2b"  
        }  
    }  
]
```

서브넷

```
{  
    "SpotPrice": "1.00",  
    "TargetCapacity": 30,  
    "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",  
    "LaunchSpecifications": [  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "c3.2xlarge",  
            "SubnetId": "subnet-1a2b3c4d",  
            "SpotPrice": "0.10"  
        },  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "c3.4xlarge",  
            "SubnetId": "subnet-1a2b3c4d",  
            "SpotPrice": "0.20"  
        },  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "c3.8xlarge",  
            "SubnetId": "subnet-1a2b3c4d"  
        }  
    ]  
}
```

예 5: 다각화된 할당 전략을 사용하여 스팟 집합 시작

다음 예제에서는 diversified 할당 전략을 사용합니다. 시작 사양의 인스턴스 유형은 다르지만 AMI와 가용 영역 또는 서브넷은 같습니다. 스팟 집합이 3개의 시작 사양에 각 유형의 인스턴스가 10개씩 있도록 30개의 인스턴스를 분산합니다. 자세한 내용은 [스팟 집합 할당 전략 \(p. 208\)](#) 섹션을 참조하십시오.

가용 영역

```
{  
    "SpotPrice": "0.70",  
    "TargetCapacity": 30,  
    "AllocationStrategy": "diversified",  
    "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",  
    "LaunchSpecifications": [  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "c4.2xlarge",  
            "Placement": {  
                "AvailabilityZone": "us-west-2b"  
            }  
        },  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "m3.2xlarge",  
            "Placement": {  
                "AvailabilityZone": "us-west-2b"  
            }  
        },  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "r3.2xlarge",  
            "Placement": {  
                "AvailabilityZone": "us-west-2b"  
            }  
        }  
    ]  
}
```

서브넷

```
{  
    "SpotPrice": "0.70",  
    "TargetCapacity": 30,  
    "AllocationStrategy": "diversified",  
    "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",  
    "LaunchSpecifications": [  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "c4.2xlarge",  
            "SubnetId": "subnet-1a2b3c4d"  
        },  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "m3.2xlarge",  
            "SubnetId": "subnet-1a2b3c4d"  
        },  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "r3.2xlarge",  
            "SubnetId": "subnet-1a2b3c4d"  
        }  
    ]  
}
```

예 6: 인스턴스 가중치를 사용하여 스팟 집합 시작

다음 예제에서는 인스턴스 가중치를 사용하는데, 이는 곧 입찰 가격이 인스턴스 시간당이 아니라 단위 시간당 가격이라는 의미입니다. 각 시작 구성마다 다른 인스턴스 유형과 다른 가중치가 나열됩니다. 스팟 집합이 단위 시간당 최저 가격의 인스턴스 유형을 선택합니다. 스팟 집합은 목표 용량을 인스턴스 가중치로 나누어 시작할 스팟 인스턴스의 수를 계산합니다. 결과가 정수가 아닌 경우, 스팟 집합은 결과를 다음 정수로 올림하므로 집합의 크기가 목표 용량을 밀들지는 않습니다.

r3.2xlarge 입찰에 성공하면 스팟이 이들 인스턴스 중 4개를 프로비저닝합니다. (20을 6으로 나누면 총 3.33 개의 인스턴스가 되는데, 이를 올림 처리하여 4개의 인스턴스가 됩니다.)

c3.xlarge 입찰에 성공하면 스팟이 이런 인스턴스 7개를 프로비저닝합니다. (20을 3으로 나누면 총 6.66개의 인스턴스가 되는데, 이를 올림 처리하여 7개의 인스턴스가 됩니다.)

자세한 내용은 [스팟 집합 인스턴스 가중치 부여 \(p. 209\)](#) 섹션을 참조하십시오.

가용 영역

```
{  
    "SpotPrice": "0.70",  
    "TargetCapacity": 20,  
    "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",  
    "LaunchSpecifications": [  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "r3.2xlarge",  
            "Placement": {  
                "AvailabilityZone": "us-west-2b"  
            },  
            "WeightedCapacity": 6  
        },  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "c3.xlarge",  
            "Placement": {  
                "AvailabilityZone": "us-west-2b"  
            },  
            "WeightedCapacity": 3  
        }  
    ]  
}
```

서브넷

```
{  
    "SpotPrice": "0.70",  
    "TargetCapacity": 20,  
    "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",  
    "LaunchSpecifications": [  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "r3.2xlarge",  
            "SubnetId": "subnet-1a2b3c4d",  
            "WeightedCapacity": 6  
        },  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "c3.xlarge",  
            "SubnetId": "subnet-1a2b3c4d",  
            "WeightedCapacity": 3  
        }  
    ]  
}
```

Priority

인스턴스 가중치를 사용하여 가용 영역 또는 서브넷에 우선순위를 지정할 수도 있습니다. 예를 들어 다음 시작 사양은 서브넷과 가중치를 달리 지정한다는 점을 제외하면 거의 똑같습니다. 스팟 집합은 WeightedCapacity에 대한 최고의 값을 가진 사양을 찾아서 그 서브넷에서 가장 가격이 낮은 스팟 인스턴스

풀의 요청을 프로비저닝하려고 합니다. (두 번째 시작 사양에는 가중치가 포함되지 않으므로 기본값은 10입니다.)

```
{  
    "SpotPrice": "0.42",  
    "TargetCapacity": 40,  
    "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",  
    "LaunchSpecifications": [  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "c3.2xlarge",  
            "SubnetId": "subnet-482e4972",  
            "WeightedCapacity": 2  
        },  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "c3.2xlarge",  
            "SubnetId": "subnet-bb3337d"  
        }  
    ]  
}
```

스팟 집합에 대한 CloudWatch 측정치

Amazon EC2는 스팟 집합을 모니터링하는 데 사용할 수 있는 Amazon CloudWatch 측정치를 제공합니다.

Important

정확성을 보장하기 위해, 이 측정치를 사용할 때는 세부 모니터링을 활성화하는 것이 좋습니다. 자세한 내용은 [인스턴스에 대한 세부 모니터링 활성화 또는 비활성화 \(p. 348\)](#) 섹션을 참조하십시오.

Amazon EC2가 제공하는 CloudWatch 측정치에 대한 자세한 내용은 [CloudWatch를 사용해 인스턴스 모니터링하기 \(p. 347\)](#) 섹션을 참조하십시오.

스팟 집합 측정치

AWS/EC2Spot 네임스페이스에는 다음과 같은 측정치가 포함되며, 아울러 집합의 스팟 인스턴스에 대한 CloudWatch 측정치도 들어 있습니다. 자세한 내용은 [인스턴스 측정치 \(p. 349\)](#) 섹션을 참조하십시오.

AWS/EC2Spot 네임스페이스에는 다음 지표가 포함되어 있습니다.

지표	설명
AvailableInstancePoolsCount	스팟 집합 요청에 지정된 스팟 인스턴스 풀. 단위: 수
BidsSubmittedForCapacity	Amazon EC2가 입찰을 제출한 용량. 단위: 수
EligibleInstancePoolCount	Amazon EC2가 입찰을 이행할 수 있는 스팟 집합 요청에 지정된 스팟 인스턴스 풀. Amazon EC2는 입찰 가격이 스팟 가격보다 낮거나 스팟 가격이 온디맨드 인스턴스 가격보다 높은 경우 풀에서 입찰을 이행하지 않습니다. 단위: 수
FulfilledCapacity	Amazon EC2가 달성한 용량. 단위: 수

지표	설명
MaxPercentCapacityAllocation	스팟 집합 요청에 지정된 모든 스팟 인스턴스 폴에 걸친 PercentCapacityAllocation의 최대값. 단위: 백분율
PendingCapacity	TargetCapacity와 FulfilledCapacity의 차이점. 단위: 수
PercentCapacityAllocation	지정된 차원의 스팟 인스턴스 폴에 할당된 용량. 모든 스팟 인스턴스 폴에 기록된 최대값을 얻으려면 MaxPercentCapacityAllocation을 사용하십시오. 단위: 백분율
TargetCapacity	스팟 집합 요청의 목표 용량. 단위: 수
TerminatingCapacity	스팟 인스턴스 간접으로 인해 종료되고 있는 용량. 단위: 수

수치 측정 단위가 Count(수)인 경우, 가장 유용한 통계는 Average(평균)입니다.

스팟 집합 차원

스팟 집합에 대한 데이터를 필터링하기 위해 다음 차원들을 사용할 수 있습니다.

차원	설명
AvailabilityZone	가용 영역별로 데이터를 필터링합니다.
FleetRequestId	스팟 집합 요청별로 데이터를 필터링합니다.
InstanceType	인스턴스 유형별로 데이터를 필터링합니다.

스팟 집합에 대한 CloudWatch 측정치 보기

Amazon CloudWatch 콘솔을 사용해 스팟 집합에 대한 CloudWatch 측정치를 볼 수 있습니다. 이 측정치들은 모니터링 그래프로 표시됩니다. 이 그래프들은 스팟 집합이 활성화되면 데이터 요소를 표시합니다.

측정치는 먼저 네임스페이스별로 그룹화된 다음, 각 네임스페이스 내에서 다양한 차원 조합별로 그룹화됩니다. 예를 들어, 모든 스팟 집합 측정치를 볼 수 있거나, 아니면 스팟 집합 요청 ID, 인스턴스 유형 또는 가용 영역별로 그룹화된 스팟 집합 측정치를 볼 수 있습니다.

스팟 집합 측정치를 보려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창의 [Metrics]에서 [EC2 Spot] 네임스페이스를 선택합니다.
3. (선택 사항) 측정치를 차원을 기준으로 필터링하려면 다음 중 하나를 선택하십시오.
 - [Fleet Request Metrics] – 스팟 집합 요청에 따른 그룹
 - [By Availability Zone] – 스팟 집합 요청 및 가용 영역에 따른 그룹
 - [By Instance Type] – 스팟 집합 요청 및 인스턴스 유형에 따른 그룹

- [By Availability Zone/Instance Type] – 스팟 집합 요청, 가용 영역 및 인스턴스 유형에 따른 그룹
4. 측정치에 대한 데이터를 보려면 측정치 옆의 확인란을 선택합니다.

스팟 집합의 자동 조정

자동 조정은 수요에 따라 스팟 집합의 대상 용량을 자동으로 늘리거나 줄이는 기능입니다. 선택하는 범위 내에서 하나 이상의 조정 정책에 대한 응답으로 스팟 집합이 인스턴스 시작(스케일 아웃) 또는 인스턴스 종료(스케일 인)를 수행할 수 있습니다. 스케일 아웃과 스케일 인에 대해 각각 하나씩 두 개의 정책을 생성하는 것이 좋습니다.

조정 정책은 CloudWatch 경보를 사용하여 조정 프로세스를 트리거합니다. 예를 들어, CPU 사용률이 특정 레벨에 도달하면 확장하려는 경우 Amazon EC2에서 제공하는 `cputUtilization` 측정치를 사용하여 경보를 생성합니다.

조정 정책을 생성할 때 다음 조정 조절 유형 중 하나를 지정해야 합니다.

- [Add] - 지정된 수의 용량 단위 또는 지정된 현재 용량의 퍼센트까지 집합의 대상 용량을 늘립니다.
- [Remove] - 지정된 수의 용량 단위 또는 지정된 현재 용량의 퍼센트까지 집합의 대상 용량을 줄입니다.
- [Set to] - 집합의 대상 용량을 지정된 수의 용량 단위로 설정합니다.

경보가 트리거되면 Auto Scaling 프로세스가 이행된 용량과 조정 정책을 사용하여 새로운 대상 용량을 계산한 후 그에 따라 목표 용량을 업데이트합니다. 예를 들어 목표 용량과 이행된 용량이 10이고 조정 정책이 1을 추가한다고 가정하십시오. 경보가 트리거되면 Auto Scaling 프로세스가 10에 1을 더해 11이 되므로 스팟 집합이 1 인스턴스를 시작합니다.

인스턴스 가중치 부여를 사용 중인 경우 스팟 집합이 필요에 따라 목표 용량을 초과할 수 있고, 이행된 용량이 부등 소수점 숫자일 수 있으나 목표 용량은 정수여야 하므로, 스팟 집합은 결과를 다음 정수로 옮김한다는 사실에 유의하십시오. 경보가 트리거되면 조정 정책의 결과를 확인할 때 이러한 동작을 고려해야 합니다. 예를 들어 목표 용량이 30, 이행된 용량이 30.1이고 조정 정책이 1을 뺀다고 가정하십시오. 경보가 트리거되면 Auto Scaling 프로세스가 30.1에서 1을 빼 29.1을 도출한 후 30으로 옮기므로 조정 작업이 수행되지 않습니다. 다른 예를 들자면, 선택한 인스턴스의 가중치가 2, 4, 8이고 목표 용량이 10이지만 가중치 2인 인스턴스를 사용할 수 없었기 때문에 스팟 집합이 가중치 4와 8인 인스턴스를 프로비저닝하여 이행된 용량이 12가 되었다고 가정하십시오. 조정 정책이 목표 용량을 20% 줄이고 경보가 트리거되면 Auto Scaling 프로세스가 12에서 12*.02를 빼 9.6을 도출한 후 10으로 옮기므로 조정 작업이 수행되지 않습니다.

조정 정책에 대한 휴지 기간 또한 구성할 수 있습니다. 이 기간은 이전 트리거 관련 조정 활동이 향후 조정 이벤트에 영향을 줄 수 있는 경우 조정 활동이 완료된 후의 시간(초)입니다. 확장 정책의 경우, 휴지 기간이 진행되는 동안 휴지하기 시작한 이전 확장 이벤트에 의해 추가된 용량은 다음 확장에 대해 원하는 용량의 일부로 계산됩니다. 지속적이지만 과도하지는 않게 확장하기 위한 목적입니다. 축소 정책의 경우, 휴지 기간은 만료될 때까지 후속 축소 요청을 차단하기 위해 사용됩니다. 보수적으로 축소하여 애플리케이션의 가용성을 보호하기 위한 목적입니다. 그러나 축소 후 휴지 기간 동안 다른 경보가 확장 정책을 트리거하면 Auto Scaling은 확장 가능한 대상을 즉시 확장합니다.

목표 용량이 감소하여 스팟 집합이 인스턴스를 종료할 때 해당 인스턴스는 스팟 인스턴스 종료 공지를 받는다는 점을 유의하십시오.

제한

- 스팟 집합 요청에 `maintain` 요청 유형이 있어야 합니다. 1회 요청 또는 스팟 블록에는 자동 조정이 지원되지 않습니다.

사전 조건

- 어떤 CloudWatch 지표가 애플리케이션에 중요한지 생각하십시오. AWS에서 제공하는 측정치 또는 사용자 지정 측정치를 기반으로 CloudWatch 경보를 생성할 수 있습니다.

- 조정 정책에 사용할 AWS 측정치에 대해 측정치를 제공하는 서비스에서 기본적으로 활성화하지 않는 경우 CloudWatch 측정치 수집을 활성화합니다.
- AWS Management Console을 사용하여 스팟 집합에 대한 자동 조정을 활성화하는 경우 정책에 대한 경보를 설명하고, 집합의 현재 용량을 모니터링하고, 집합의 용량을 수정할 수 있는 권한을 Auto Scaling에 부여하는 `aws-ec2-spot-fleet-autoscale-role` 역할이 생성됩니다. AWS CLI 또는 API를 사용하여 자동 조정을 구성하는 경우 이 역할이 존재하면 이 역할을 사용하거나, 다음과 같이 이 용도로 사용할 고유의 역할을 수동으로 생성할 수 있습니다.
 1. <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
 2. 탐색 창에서 [Roles]를 선택합니다.
 3. [Create New Role(새 역할 만들기)]을 선택합니다.
 4. [Set Role Name] 페이지에서 역할 이름을 입력한 다음, [Next Step]을 선택합니다.
 5. [Select Role Type] 페이지에서 [Amazon EC2] 옆의 [Select]를 선택합니다.
 6. [Attach Policy] 페이지에서 `AmazonEC2SpotFleetAutoscaleRole` 정책을 선택한 후 [Next Step]을 선택합니다.
 7. [Review] 페이지에서 [Create Role]을 선택합니다.
 8. 방금 생성한 역할을 선택합니다.
 9. [Trust Relationships] 탭에서 [Edit Trust Relationship]을 선택합니다.
 10. `ec2.amazonaws.com`을 `application-autoscaling.amazonaws.com`으로 변경한 후 [Update Trust Policy]를 선택합니다.

CloudWatch 경보를 생성하려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Alarms를 선택합니다.
3. [Create Alarm]을 선택합니다.
4. [CloudWatch Metrics by Category]에서 범주를 선택합니다. 예를 들어, [EC2 Spot Metrics], [Fleet Request Metrics]를 선택합니다.
5. 지표를 선택한 후 [Next]를 선택합니다.
6. [Alarm Threshold]에서 경보의 이름과 설명을 입력하고 임계값 및 경보의 기간 수를 설정합니다.
7. (선택 사항) 조정 이벤트에 대한 알림을 받으려면 [Actions]에서 [New list]를 선택하고 이메일 주소를 입력합니다. 또는 지금 알림을 삭제하고 필요한 경우 나중에 추가할 수 있습니다.
8. [Create Alarm]을 선택합니다.

콘솔을 사용하여 스팟 집합에 대한 자동 조정을 구성하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Spot Requests]를 선택합니다.
3. 스팟 집합 요청을 선택한 후 [Auto Scaling] 탭을 선택합니다.
4. 자동 조정이 구성되어 있지 않으면 [Configure]를 선택합니다.
5. [Scale capacity between]을 사용하여 집합에 대한 최소 및 최대 용량을 설정합니다. 자동 조정에서 최소 용량 미만이거나 최대 용량을 초과하는 집합을 조정하지 않습니다.
6. 처음에 [Scaling policies]에 ScaleUp 및 ScaleDown이라는 정책이 포함됩니다. 이러한 정책을 완료하거나 [Remove policy]를 선택하여 삭제할 수 있습니다. 또한 [Add policy]를 선택하여 정책을 추가할 수도 있습니다.
7. 정책을 정의하려면 다음을 수행합니다.
 - a. [Policy name]에서 정책의 이름을 입력합니다.
 - b. [Policy trigger]에서 기존 경보를 선택하거나 [Create new alarm]을 선택하여 Amazon CloudWatch 콘솔을 열고 경보를 생성합니다.

- c. [Modify capacity]에서 조정 조절 유형을 선택하고, 숫자를 선택한 후 단위를 선택합니다.
 - d. (선택 사항) 단계 조정을 수행하려면 [Define steps]를 선택합니다. 기본적으로 추가 정책에 하한값으로 -infinity 값이, 상한값으로 경보 임계치가 적용됩니다. 또한 제거 정책에 하한값으로 경보 임계치 및 상한값으로 +infinity 값이 기본적으로 적용됩니다. 다른 단계를 추가하려면 [Add step]을 선택합니다.
 - e. (선택 사항) 휴지 기간의 기본값을 수정하려면 [Cooldown period]에서 숫자를 선택합니다.
8. [Save]를 선택합니다.

AWS CLI를 사용하여 스팟 집합에 대한 자동 조정을 구성하려면

1. `register-scalable-target` 명령을 사용하여 스팟 집합 요청을 확장 가능 대상으로 등록합니다.
2. `put-scaling-policy` 명령을 사용하여 조정 정책을 생성합니다.
3. `put-metric-alarm` 명령을 사용하여 조정 정책을 트리거할 경보를 생성합니다.

스팟 입찰 상태

스팟 인스턴스 요청을 쉽게 추적하고 스팟 인스턴스 사용을 계획하여 전략적으로 입찰할 수 있도록 Amazon EC2는 입찰 상태를 제공합니다. 예를 들어, 입찰 상태는 스팟 요청이 아직 이행되지 않는 이유를 알려주거나, 스팟 요청을 이행할 수 없는 제약 조건을 나열할 수 있습니다.

프로세스의 각 단계(스팟 요청 수명 주기라고도 함)에서 특정 이벤트에 따라 연속 요청 상태를 결정합니다.

목차

- [스팟 요청의 수명 주기 \(p. 238\)](#)
- [입찰 상태 정보 가져오기 \(p. 240\)](#)
- [스팟 입찰 상태 코드 \(p. 241\)](#)

스팟 요청의 수명 주기

다음 다이어그램에서는 제출부터 종료까지 전체 수명 주기 동안 스팟 요청이 따를 수 있는 경로를 보여 줍니다. 각 단계는 노드로 묘사되며 각 노드의 상태 코드는 스팟 요청 및 스팟 인스턴스의 상태를 설명합니다.

평가 보류

하나 이상의 요청 파라미터가 잘못되지 않은 한(bad-parameters), 스팟 인스턴스 요청을 수행하는 즉시 pending-evaluation 상태로 요청이 전환됩니다.

상태 코드	요청 상태	인스턴스 상태
pending-evaluation	open	해당 사항 없음
bad-parameters	closed	해당 사항 없음

보류

하나 이상의 요청 제약 조건이 적용되지만 아직 충족될 수 없는 경우 또는 용량이 부족한 경우 요청은 제약 조건이 충족될 때까지 대기하는 보류 상태로 전환됩니다. 요청 옵션은 요청이 이행될 가능성에 영향을 미칩니다. 예를 들어, 입찰 가격을 현재 스팟 가격보다 낮게 지정할 경우 스팟 가격이 입찰 가격 아래로 떨어질 때까지 요청은 보류 상태로 유지됩니다. 가용 영역 그룹을 지정할 경우 가용 영역 제약 조건이 충족될 때까지 요청은 보류 상태로 유지됩니다.

상태 코드	요청 상태	인스턴스 상태
capacity-not-available	open	해당 사항 없음
capacity-oversubscribed	open	해당 사항 없음
price-too-low	open	해당 사항 없음
not-scheduled-yet	open	해당 사항 없음
launch-group-constraint	open	해당 사항 없음
az-group-constraint	open	해당 사항 없음
placement-group-constraint	open	해당 사항 없음
constraint-not-fulfillable	open	해당 사항 없음

평가/이행 보류-끝

특정 기간 동안에만 유효한 요청을 생성하는 경우 요청이 이행 보류 단계에 도달하거나 사용자가 요청을 취소하거나 시스템 오류가 발생하기 전에 이 기간이 만료되면 스팟 인스턴스 요청은 `terminal` 상태로 전환될 수 있습니다.

상태 코드	요청 상태	인스턴스 상태
<code>schedule-expired</code>	<code>closed</code>	해당 사항 없음
<code>canceled-before-fulfillment*</code>	<code>cancelled</code>	해당 사항 없음
<code>bad-parameters</code>	<code>failed</code>	해당 사항 없음
<code>system-error</code>	<code>closed</code>	해당 사항 없음

* 사용자가 요청을 취소하는 경우.

이행 보류

지정한 제약 조건(있는 경우)이 충족되고 입찰 가격이 현재 스팟 가격보다 높거나 같은 경우 스팟 요청은 `pending-fulfillment` 상태로 전환됩니다.

이 시점에 Amazon EC2는 요청한 인스턴스를 프로비저닝할 준비를 합니다. 프로세스가 이 시점에 중지될 경우 스팟 인스턴스가 시작되기 전에 사용자가 프로세스를 취소했거나 예상치 않은 시스템 오류가 발생했기 때문일 수 있습니다.

상태 코드	요청 상태	인스턴스 상태
<code>pending-fulfillment</code>	<code>open</code>	해당 사항 없음

이행됨

스팟 인스턴스의 모든 사양이 충족되면 스팟 요청이 이행됩니다. Amazon EC2가 스팟 인스턴스를 시작합니다. 이 작업은 몇 분 정도 걸릴 수 있습니다.

상태 코드	요청 상태	인스턴스 상태
<code>fulfilled</code>	<code>active</code>	<code>pending → running</code>

이행됨-끝

입찰 가격이 스팟 가격보다 높거나 같고 인스턴스 유형에 대한 예비 스팟 용량이 있으며 사용자가 인스턴스를 종료하지 않는 한, 스팟 인스턴스가 계속 실행됩니다. 스팟 가격 또는 가용 용량을 변경하려면 Amazon EC2에서 스팟 인스턴스를 종료해야 하는 경우 스팟 요청이 끝 상태로 전환됩니다. 예를 들어, 입찰 가격이 스팟 가격과 같지만 해당 가격에서 스팟 인스턴스의 수요가 공급을 초과할 때 상태 코드는 `instance-terminated-capacity-oversubscribed`입니다. 사용자가 스팟 요청을 취소하거나 스팟 인스턴스를 종료하는 경우에도 요청이 끝 상태로 전환됩니다.

상태 코드	요청 상태	인스턴스 상태
<code>request-canceled-and-instance-running</code>	<code>cancelled</code>	<code>running</code>
<code>marked-for-termination</code>	<code>closed</code>	<code>running</code>
<code>instance-terminated-by-price</code>	<code>closed(일회)</code> , <code>open(영구)</code>	<code>terminated</code>
<code>instance-terminated-by-user</code>	<code>closed</code> 또는 <code>cancelled</code> *	<code>terminated</code>
<code>instance-terminated-no-capacity</code>	<code>closed(일회)</code> , <code>open(영구)</code>	<code>terminated</code>
<code>instance-terminated-capacity-oversubscribed</code>	<code>closed(일회)</code> , <code>open(영구)</code>	<code>terminated</code>
<code>instance-terminated-launch-group-constraint</code>	<code>closed(일회)</code> , <code>open(영구)</code>	<code>terminated</code>

* 인스턴스를 종료하지만 입찰을 취소하지 않는 경우 요청 상태는 `closed`입니다. 인스턴스를 종료하고 입찰을 취소하는 경우 요청 상태는 `cancelled`입니다. 스팟 요청을 취소하기 전에 스팟 인스턴스를 종료하더라도 Amazon EC2에서 스팟 인스턴스가 종료되었음을 감지하기 전에는 자연이 발생할 수 있습니다. 이 경우 요청 상태는 `closed` 또는 `cancelled`일 수 있습니다.

영구 요청

스팟 인스턴스가 종료될 때(사용자가 종료하거나 Amazon EC2에서 종료) 스팟 요청이 영구 요청인 경우 `pending-evaluation` 상태가 반환되고 제약 조건이 충족되면 Amazon EC2가 새로운 스팟 인스턴스를 시작할 수 있습니다.

입찰 상태 정보 가져오기

AWS Management Console 또는 명령줄 도구를 사용하여 입찰 상태 정보를 가져올 수 있습니다.

콘솔을 사용하여 입찰 상태 정보를 가져오려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Spot Requests]를 선택한 다음 스팟 요청을 선택합니다.
3. [Description] 탭에서 [Status]의 값을 확인합니다.

명령줄을 사용하여 입찰 상태 정보를 가져오려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- `describe-spot-instance-requests`(AWS CLI)

- [Get-EC2SpotInstanceRequest\(Windows PowerShell용 AWS 도구\)](#)

스팟 입찰 상태 코드

스팟 입찰 상태 정보는 입찰 상태 코드, 업데이트 시간 및 상태 메시지로 구성됩니다. 이러한 정보는 스팟 요청 배치를 결정하는 데 도움이 됩니다.

다음 목록은 스팟 입찰 상태 코드입니다.

`az-group-constraint`

Amazon EC2가 동일한 가용 영역에 요청한 모든 인스턴스를 시작할 수 있는 것은 아닙니다.

`bad-parameters`

스팟 요청에 대한 파라미터 하나 이상이 올바르지 않습니다(예를 들어, 지정한 AMI가 존재하지 않음). 입찰 상태 메시지는 어떤 파라미터가 올바르지 않은지를 나타냅니다.

`cancelled-before-fulfillment`

요청이 이행되기 전에 사용자가 스팟 요청을 취소했습니다.

`capacity-not-available`

요청한 인스턴스에 사용 가능한 용량이 부족합니다.

`capacity-oversubscribed`

입찰 가격이 사용자의 입찰 가격과 같거나 그보다 높은 스팟 요청의 수는 이 스팟 인스턴스 풀에서 사용 가능한 용량을 초과합니다.

`constraint-not-fulfillable`

하나 이상의 제약 조건이 올바르지 않기 때문에(예: 가용 영역이 존재하지 않음) 스팟 요청을 이행할 수 없습니다. 입찰 상태 메시지는 어떤 제약 조건이 올바르지 않은지를 나타냅니다.

`fulfilled`

상태 요청이 `active` 상태이며 Amazon EC2가 스팟 인스턴스를 시작하고 있습니다.

`instance-terminated-by-price`

스팟 가격이 입찰 가격보다 상승했습니다. 요청이 영구 입찰인 경우 프로세스가 다시 시작되므로 입찰은 평가 보류 상태입니다.

`instance-terminated-by-user` 또는 `spot-instance-terminated-by-user`

이행되지 않은 스팟 인스턴스를 종료했으므로, 입찰 상태는 `closed`(영구 입찰이 아닌 경우)이고 인스턴스 상태는 `terminated`입니다.

`instance-terminated-capacity-oversubscribed`

입찰 가격이 사용자의 입찰 가격과 같거나 그보다 높은 스팟 요청의 수는 이 스팟 인스턴스 풀에서 사용 가능한 용량을 초과하기 때문에 인스턴스가 종료됩니다. (스팟 가격은 변경되지 않았을 수 있습니다.) 스팟 서비스는 종료할 인스턴스를 임의로 선택합니다.

`instance-terminated-launch-group-constraint`

시작 그룹에 있는 하나 이상의 인스턴스가 종료되었으므로 시작 그룹 제약 조건이 더 이상 충족되지 않습니다.

`instance-terminated-no-capacity`

인스턴스에 사용 가능한 스팟 용량이 부족합니다.

launch-group-constraint

Amazon EC2가 동일한 시간에 요청한 모든 인스턴스를 시작할 수 있는 것은 아닙니다. 시작 그룹에 있는 모든 인스턴스가 함께 시작되고 종료됩니다.

limit-exceeded

EBS 볼륨 또는 전체 볼륨 스토리지 수 제한을 초과했습니다. 이러한 제한값 및 증가 요청 방법에 대한 자세한 내용은 Amazon Web Services 일반 참조에서 [Amazon EBS 제한](#)을 참조하십시오.

marked-for-termination

종료할 스팟 인스턴스가 표시됩니다.

not-scheduled-yet

예정된 날짜까지 스팟 요청이 평가되지 않습니다.

pending-evaluation

스팟 인스턴스 요청을 수행한 후 시스템에서 요청 파라미터를 평가하는 동안 요청이 `pending-evaluation` 상태로 전환됩니다.

pending-fulfillment

Amazon EC2가 스팟 인스턴스를 프로비저닝하려고 하고 있습니다.

placement-group-constraint

이 시점에는 스팟 인스턴스를 배치 그룹에 추가할 수 없기 때문에 스팟 요청을 이행할 수 없지만 스팟 인스턴스를 배치 그룹에 추가할 수 있습니다.

price-too-low

입찰 가격이 스팟 가격보다 낮기 때문에 입찰 요청을 아직 이행할 수 없습니다. 이 경우 인스턴스가 시작되지 않으며 입찰이 `open` 상태로 유지됩니다.

request-cancelled-and-instance-running

스팟 인스턴스가 여전히 실행되는 동안 사용자가 스팟 요청을 취소했습니다. 요청은 `cancelled` 상태지만 인스턴스는 여전히 `running` 상태입니다.

schedule-expired

지정된 날짜 이전에 요청이 이행되지 않았기 때문에 스팟 요청이 만료되었습니다.

system-error

예상치 않은 시스템 오류입니다. 이 문제가 반복되면 고객 지원 센터에 문의하십시오.

스팟 인스턴스 중단

스팟 인스턴스에 대한 수요는 매 순간 상당히 다를 수 있으며 스팟 인스턴스의 가용성도 사용 가능한 미사용 EC2 인스턴스의 양에 따라 상당히 다를 수 있습니다. 또한 얼마나 높게 입찰하든 상관없이 스팟 인스턴스가 여전히 종단될 수 있습니다. 따라서 스팟 인스턴스 종단에 대비하여 애플리케이션을 준비해야 합니다. 종단할 수 없는 애플리케이션에는 스팟 인스턴스를 사용하지 않는 것이 좋습니다.

Amazon EC2가 스팟 인스턴스를 종료할 수 있는 이유는 다음과 같습니다.

- 가격 - 스팟 가격이 입찰 가격보다 큽니다.
- 용량 - 미사용 EC2 인스턴스가 스팟 인스턴스의 수요를 충족하기에 부족한 경우 Amazon EC2는 스팟 인스턴스를 종료하여 최저 입찰 가격으로 이러한 인스턴스를 시작합니다. 입찰 가격이 동일한 여러 스팟 인스턴스가 있는 경우 인스턴스가 종료되는 순서는 임의로 결정됩니다.

- 제약 조건 - 요청에 시작 그룹 또는 가용 영역 그룹과 같은 제약 조건이 포함되는 경우 제약 조건을 더 이상 충족할 수 없으면 이러한 스팟 인스턴스가 그룹으로 종료됩니다.

중단에 대한 준비

스팟 인스턴스를 사용할 때 따라야 할 몇 가지 모범 사례는 다음과 같습니다.

- 합리적인 입찰 가격을 선택합니다. 입찰 가격은 요청이 이행될 수 있을 만큼 높으면서 지불하려는 금액보다 높지 않아야 합니다. 장기간 공급이 부족할 경우 해당 기간 동안 스팟 가격이 최고 입찰 가격을 기준으로 높게 유지될 수 있기 때문에 이 점이 중요합니다. 온디맨드 인스턴스 가격보다 높게 입찰하지 않는 것이 좋습니다.
- 필수 소프트웨어 구성이 포함된 Amazon 머신 이미지(AMI)를 사용하여 요청이 이행되는 즉시 인스턴스를 실행할 준비가 되었는지 확인합니다. 시작 시 사용자 데이터를 사용하여 명령을 실행할 수도 있습니다.
- 스팟 인스턴스가 종료되더라도 영향을 받지 않을 장소에 중요한 데이터를 정기적으로 저장하십시오. 예를 들어, Amazon S3, Amazon EBS 또는 DynamoDB를 사용할 수 있습니다.
- 작업을 작은 부분으로 분리하거나(눈금, 하둡 또는 대기열 기반 아키텍처 사용), 작업을 자주 저장할 수 있도록 검사점을 사용합니다.
- 스팟 인스턴스 종료 공지를 사용하여 스팟 인스턴스의 상태를 모니터링합니다.
- 애플리케이션을 테스트하여 예상치 않은 인스턴스 종료가 정상적으로 처리되는지 확인합니다. 이렇게 하려면 온디맨드 인스턴스를 사용하여 애플리케이션을 실행한 다음 온디맨드 인스턴스를 직접 종료합니다.

스팟 인스턴스 종료 공지

스팟 인스턴스 종단으로부터 보호하는 가장 좋은 방법은 애플리케이션을 내결함성 있게 설계하는 것입니다. 또한 Amazon EC2가 스팟 인스턴스를 종료하기 2분 전에 경고하는 스팟 인스턴스 종료 공지를 이용할 수 있습니다.

이 경고는 인스턴스 메타데이터의 한 항목을 사용하여 스팟 인스턴스의 애플리케이션에 제공됩니다. 예를 들어, 다음 쿼리를 사용하여 인스턴스 메타데이터에서 이 경고를 정기적으로 확인할 수 있습니다(5초마다 확인 권장).

```
$ if curl -s http://169.254.169.254/latest/meta-data/spot/termination-time | grep -q .*T.*Z; then echo terminated; fi
```

인스턴스 메타데이터를 검색하는 다른 방법에 대한 자세한 내용은 [인스턴스 메타데이터 가져오기 \(p. 321\)](#) 섹션을 참조하십시오.

Amazon EC2에서 종료할 스팟 인스턴스가 표시되면 `termination-time` 항목이 나타나며 이 항목을 사용하여 인스턴스가 종료 신호를 받을 적절한 시간(UTC 기준)을 지정합니다. 예:

```
2015-01-05T18:02:00Z
```

Amazon EC2가 인스턴스를 종료할 준비가 되지 않거나 사용자가 스팟 인스턴스를 직접 종료한 경우 `termination-time` 항목이 나타나지 않거나(HTTP 404 오류 수신) 이 항목에 시간 값이 아닌 값이 포함됩니다.

Amazon EC2에서 종료할 스팟 인스턴스가 표시되는 즉시 이 경고를 제공하고자 모든 노력을 다하고 있지만 Amazon EC2에서 경고를 제공하기 전에 스팟 인스턴스가 종료될 수 있습니다. 따라서 스팟 인스턴스 종료 공지를 확인하고 있더라도 예상치 않은 스팟 인스턴스 종료를 처리할 수 있도록 애플리케이션을 준비해야 합니다.

Amazon EC2에서 인스턴스를 종료하지 않으면 스팟 입찰 상태는 `fulfilled`로 설정됩니다. `termination-time`은 과거 시점인 원래 예상 시간과 함께 인스턴스 메타데이터에 남습니다.

스팟 인스턴스 데이터 피드

스팟 인스턴스 요금을 쉽게 이해할 수 있도록 Amazon EC2는 스팟 인스턴스 사용 및 요금을 설명하는 데이터 피드를 제공합니다. 이 데이터 피드는 데이터 피드를 구독할 때 지정하는 Amazon S3 버킷으로 전송됩니다.

일반적으로 데이터 피드 파일은 한 시간에 한 번씩 버킷에 도착하며, 각 사용 시간이 단일 데이터 파일로 설명됩니다. 이 파일은 압축(gzip)된 후 버킷으로 전송됩니다. 파일이 매우 큰 경우 Amazon EC2는 지정된 사용 시간에 대해 여러 개의 파일을 작성할 수 있습니다(예: 압축 전 해당 시간의 파일 콘텐츠가 50MB를 초과하는 경우).

Note

특정 시간 동안 스팟 인스턴스가 없는 경우 해당 시간에 대한 데이터 피드 파일이 수신되지 않습니다.

목차

- [데이터 피드 파일 이름 및 형식 \(p. 244\)](#)
- [Amazon S3 버킷 요구 사항 \(p. 245\)](#)
- [스팟 인스턴스 데이터 피드 구독 \(p. 245\)](#)
- [스팟 인스턴스 데이터 피드 삭제 \(p. 245\)](#)

데이터 피드 파일 이름 및 형식

스팟 인스턴스 데이터 피드 파일 이름은 다음 형식을 사용합니다(UTC 기준 날짜 및 시간).

```
bucket-name.s3.amazonaws.com/{optional prefix}/aws-account-id.YYYY-MM-DD-HH.n.unique-id.gz
```

예를 들어, 버킷 이름이 myawsbucket이고 접두사가 myprefix인 경우 파일 이름은 다음과 같습니다.

```
myawsbucket.s3.amazonaws.com/myprefix/111122223333.2014-03-17-20.001.pwBdGTJG.gz
```

스팟 인스턴스 데이터 피드 파일은 템으로 구분됩니다. 데이터 파일의 각 줄은 1 인스턴스 시간에 해당하며 다음 표에 나열된 필드를 포함합니다.

필드	설명
Timestamp	이 인스턴스 시간에 대해 청구된 가격을 결정하는 데 사용되는 타임스탬프입니다.
UsageType	청구되는 사용 유형 및 인스턴스 유형입니다. m1.small 스팟 인스턴스의 경우 이 필드는 SpotUsage로 설정됩니다. 다른 모든 인스턴스 유형의 경우 이 필드는 SpotUsage:{instance-type}으로 설정됩니다. 예, SpotUsage:c1.medium.
Operation	청구되는 제품입니다. Linux 스팟 인스턴스의 경우 이 필드는 RunInstances로 설정됩니다. Windows 스팟 인스턴스의 경우 이 필드는 RunInstances:0002로 설정됩니다. 스팟 사용은 가용 영역에 따라 그룹화됩니다.
InstanceID	이 인스턴스 시간을 생성한 스팟 인스턴스의 ID입니다.
MyBidID	이 인스턴스 시간을 생성한 스팟 인스턴스 요청의 ID입니다.
MyMaxPrice	이 스팟 인스턴스 요청에 대해 지정된 최고 가격입니다.
MarketPrice	Timestamp 필드에 지정된 시간의 스팟 가격입니다.
Charge	이 인스턴스 시간에 대해 청구된 가격입니다.

필드	설명
Version	이 레코드에 대해 데이터 피드 파일 이름에 포함된 버전입니다.

Amazon S3 버킷 요구 사항

데이터 피드를 구독하면 데이터 피드 파일을 저장하기 위한 Amazon S3 버킷을 지정해야 합니다. 데이터 피드에 대한 Amazon S3 버킷을 선택하기 전에 다음 사항을 고려하십시오.

- 미국 동부(버지니아 북부) 리전(us-east-1 또는 미국 표준 리전이라고도 함)의 버킷을 사용해야 합니다.
- 버킷에 대해 **FULL_CONTROL** 권한이 있어야 합니다.
 - 버킷 소유자인 경우 기본적으로 이 권한이 있습니다. 그렇지 않으면 버킷 소유자가 AWS 계정에 이 권한을 부여해야 합니다.
 - 데이터 피드 구독을 생성할 때 Amazon S3는 지정된 버킷의 ACL을 업데이트하여 AWS 데이터 피드 계정에 읽기 및 쓰기 권한을 허용합니다.
 - 데이터 피드 계정에 대한 권한을 제거해도 데이터 피드가 비활성화되지 않습니다. 해당 권한을 제거할 때 데이터 피드를 비활성화하지 않은 경우 다음에 데이터 피드 계정에서 버킷에 기록해야 할 때 해당 권한을 복원할 수 있습니다.
 - 각 데이터 피드 파일에는 고유의 ACL(버킷용 ACL과는 별도)이 있습니다. 버킷 소유자는 데이터 파일에 대한 **FULL_CONTROL** 권한을 가지고 있습니다. 데이터 피드 계정은 읽기 및 쓰기 권한이 있습니다.
 - 데이터 피드 구독을 삭제해도 Amazon EC2에서 버킷 또는 데이터 파일에 대한 데이터 피드 계정의 읽기 및 쓰기 권한이 제거되지 않습니다. 이러한 권한을 직접 제거해야 합니다.

스팟 인스턴스 데이터 피드 구독

데이터 피드를 구독하려면 다음 [create-spot-datafeed-subscription](#) 명령을 사용합니다.

```
$ aws ec2 create-spot-datafeed-subscription --bucket myawsbucket [--prefix myprefix]
```

다음은 예제 출력입니다.

```
{  
    "SpotDatafeedSubscription": {  
        "OwnerId": "111122223333",  
        "Prefix": "myprefix",  
        "Bucket": "myawsbucket",  
        "State": "Active"  
    }  
}
```

스팟 인스턴스 데이터 피드 삭제

데이터 피드를 삭제하려면 다음 [delete-spot-datafeed-subscription](#) 명령을 사용합니다.

```
$ aws ec2 delete-spot-datafeed-subscription
```

스팟 인스턴스 제한

스팟 인스턴스 요청에는 다음 제한이 적용됩니다.

제한

- [지원되지 않는 인스턴스 유형](#) (p. 246)

- [스팟 요청 제한 \(p. 246\)](#)
- [스팟 입찰 가격 제한 \(p. 246\)](#)
- [스팟 집합 제한 \(p. 246\)](#)
- [지원되지 않는 Amazon EBS 암호화 \(p. 246\)](#)

지원되지 않는 인스턴스 유형

다음 인스턴스 유형은 스팟에 지원되지 않습니다.

- T2
- HS1

일부 스팟 인스턴스 유형은 일부 리전에서만 사용할 수 있습니다. 리전에 지원되는 인스턴스 유형을 보려면 [스팟 인스턴스 요금](#)으로 이동하여 해당 리전을 선택하십시오.

스팟 요청 제한

기본적으로 계정의 스팟 인스턴스는 리전당 20개로 제한됩니다. 스팟 인스턴스를 종료하고 요청을 취소하지 않으면 Amazon EC2가 종료를 감지하고 요청을 닫을 때까지 해당 요청이 한도 계산에 반영됩니다.

스팟 인스턴스 한도는 동적으로 바뀝니다. 새 계정인 경우 한도 20 미만에서 시작하여 시간이 지날수록 늘어날 것입니다. 또한 특정한 스팟 인스턴스 유형에 대한 계정 한도도 있을 수 있습니다. 스팟 인스턴스 요청을 제출한 뒤 `Max spot instance count exceeded` 오류가 발생하는 경우, [AWS Support Center](#)에서 한도 향상 요청서를 제출할 수 있습니다. [Use Case Description]에서 스팟 인스턴스 요청에 대한 한도 향상이 필요하다고 표시합니다.

스팟 입찰 가격 제한

스팟 인스턴스에 대한 입찰 가격 제한은 온디맨드 가격의 열 배입니다. 이 제한은 비용을 통제할 수 있도록 지원하기 위한 것입니다.

스팟 집합 제한

스팟 입찰 가격 제한, 인스턴스 제한 및 볼륨 제한과 같이 스팟 집합에서 시작된 인스턴스에 일반적인 Amazon EC2 제한이 적용됩니다. 또한 다음과 같은 제한이 적용됩니다.

- 리전당 활성 스팟 집합 수: 1,000개
- 집합당 시작 사양 수: 50개
- 시작 사양의 사용자 데이터 크기: 16KB
- 스팟 집합당 목표 용량: 3,000
- 특정 리전 내 모든 스팟 집합의 목표 용량: 5,000
- 스팟 집합 요청은 리전에 적용할 수 없습니다.
- 스팟 집합 요청은 동일한 가용 영역의 서로 다른 서브넷에 적용할 수 없습니다.

지원되지 않는 Amazon EBS 암호화

스팟 인스턴스의 시작 사양에서 암호화된 EBS 볼륨을 지정할 수 있지만 이를 볼륨은 암호화되지 않습니다.

전용 호스트

Amazon EC2 전용 호스트는 고객 전용의 EC2 인스턴스 용량을 갖춘 물리적 서버입니다. 전용 호스트를 통해 Windows Server, Microsoft SQL Server, SUSE, Linux Enterprise Server 등 기존 소켓당, 코어당 또는 VM 당 소프트웨어 라이선스를 사용할 수 있습니다.

목차

- 전용 호스트와 전용 인스턴스 간 차이 (p. 247)
- 요금 및 결제 (p. 247)
- 전용 호스트 제한 및 제약 (p. 248)
- 전용 호스트 구성 (p. 249)
- 전용 호스트 사용 (p. 249)
- 전용 호스트 모니터링 (p. 256)

전용 호스트와 전용 인스턴스 간 차이

전용 호스트와 전용 인스턴스는 모두 사용자 전용 물리적 서버로 Amazon EC2 인스턴스를 시작하는 데 사용할 수 있습니다.

전용 인스턴스와 전용 호스트 상의 인스턴스 사이에 성능이나 보안 상의 차이, 또는 물리적 차이는 없습니다. 하지만 전용 호스트는 물리적 서버 상의 인스턴스 배치에 대한 가시성 및 제어를 추가로 제공합니다.

전용 호스트를 사용하면 호스트 선호도와 인스턴스 자동 배치를 사용하여 호스트의 인스턴스 배치를 제어할 수 있습니다. 전용 인스턴스로는 어떤 호스트에서 인스턴스를 시작하고 실행할지 제어할 수 없습니다. 조직에서 AWS를 사용하고자 하지만 기존 소프트웨어 라이선스에 하드웨어 요구사항이 규정되어 있는 경우, 이 인스턴스를 사용하면 호스트의 하드웨어를 확인하여 이러한 요구사항을 충족할 수 있습니다.

전용 호스트와 전용 인스턴스 간 차이에 대한 자세한 내용은 [Amazon EC2 전용 호스트](#) 섹션을 참조하십시오.

전용 호스트 및 전용 인스턴스 사용에 대한 자세한 내용은 [인스턴스 테넌시 설정 \(p. 253\)](#) 섹션을 참조하십시오.

요금 및 결제

온 디맨드 전용 호스트

계정에 전용 호스트를 할당하면 온디맨드 결제가 자동으로 활성화됩니다.

시간당 온디맨드 요금이 청구됩니다. 요금은 전용 호스트에서 지원하는 인스턴스 유형과 전용 호스트를 실행 중인 리전에 따라 다릅니다. 전용 호스트에서 실행하는 인스턴스 유형의 크기 또는 인스턴스 수는 호스트 비용에 영향을 주지 않습니다.

온디맨드 결제를 종료하려면 먼저 전용 호스트에서 실행 중인 인스턴스를 중지한 다음 해제해야 합니다. 자세한 내용은 [전용 호스트 관리 및 해제 \(p. 254\)](#) 섹션을 참조하십시오.

전용 호스트 예약

전용 호스트 예약은 온디맨드 전용 호스트 실행에 비해 청구 할인이 제공됩니다. 다음과 같은 세 가지 결제 방식을 통해 예약이 가능합니다.

- 선결제 없음 - 선결제가 없는 예약은 사용 기간 동안 전용 호스트 사용에 대해 할인을 제공하고 선결제가 필요없습니다. 사용 기간이 1년인 경우에만 가능합니다.
- 부분 선결제 - 예약의 일부를 선결제하고, 사용 기간 내 나머지 시간에 대해서는 할인 요금이 청구됩니다. 사용 기간이 1년 및 3년인 경우에 가능합니다.
- 전체 선결제 - 최저 실효 가격을 제공합니다. 사용 기간이 1년 및 3년인 경우에 사용 가능하며, 향후 추가 요금 없이 사용 기간 전체 비용을 커버합니다.

계정에 활성화된 전용 호스트가 있어야 예약을 구매할 수 있습니다. 각 예약은 계정에서 1개의 특정 전용 호스트에 해당됩니다. 예약은 인스턴스 크기가 아닌 호스트의 인스턴스 패밀리에 적용됩니다. 인스턴스 크기가

서로 다른 세 가지 전용 호스트(m4.xlarge, m4.medium 및 m4.large)가 있는 경우 단일 m4 예약을 모든 전용 호스트와 연결할 수 있습니다. 예약의 인스턴스 패밀리 및 리전은 연결하고자 하는 전용 호스트의 인스턴스 패밀리 및 리전과 일치해야 합니다.

Note

하나의 예약이 전용 호스트와 연결되면 예약 기간이 끝날 때까지 전용 호스트를 해제할 수 없습니다.

전용 호스트 예약 구매

콘솔 또는 API를 사용하여 전용 호스트 예약을 구입할 수 있습니다.

콘솔을 사용하여 전용 호스트 예약을 구입하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. [Dedicated Hosts] 페이지에서 [Dedicated Host Reservations]를 선택합니다.
3. [Purchase Dedicated Host Reservation]을 선택합니다.
4. [Purchase Dedicated Host Reservation] 화면에서 기본 설정을 사용하여 상품을 검색하거나 상품에 대한 구성을 지정할 수 있습니다.
 - [Host instance family] - 나열되는 옵션은 계정에서 예약에 할당되지 않은 전용 호스트에 해당합니다.
 - [Availability Zone] - 예약에 할당되지 않은 계정 내 전용 호스트의 가용 영역입니다.
 - [Payment Option] - 상품에 대한 결제 방식입니다.
 - [Term] - 예약 기간입니다. 1년 또는 3년 중 하나입니다.
5. [Find offering]을 선택합니다.
6. 상품을 선택합니다.
7. 전용 호스트 예약과 연결할 전용 호스트를 선택합니다.
8. [Review]를 선택합니다.
9. 주문을 검토하고 [Purchase]를 선택하여 트랜잭션을 완료합니다.

전용 호스트 예약 보기

예약과 연결된 전용 호스트, 예약 기간, 선택된 결제 방식, 예약 시작일 및 종료일에 대한 정보를 볼 수 있습니다.

전용 호스트 예약 세부 정보 보기

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. [Dedicated Hosts] 페이지에서 [Dedicated Host Reservations]를 선택합니다.
3. 제공된 목록에서 예약을 선택합니다.
4. 예약에 대한 정보를 보려면 [Details]를 선택합니다.
5. 예약이 연결되어 있는 전용 호스트에 대한 정보를 보려면 [Hosts]를 선택합니다.

전용 호스트 제한 및 제약

전용 호스트를 할당하기 전에 다음 제한 및 제약에 유의하십시오.

- RHEL, SUSE Linux 및 AWS가 제공하거나 AWS Marketplace에서 제공되는 Windows AMI는 전용 호스트와 함께 사용할 수 없습니다.
- Amazon EC2 인스턴스 자동 복구는 지원되지 않습니다.
- 리전별로 인스턴스 패밀리당 최대 2개의 온디맨드 전용 호스트를 할당할 수 있습니다. 한도 향상을 요청할 수 있습니다. [Request to Raise Allocation Limit on Amazon EC2 Dedicated Hosts](#).

- 전용 호스트에서 실행되는 인스턴스는 VPC에서만 시작할 수 있습니다.
- 호스트 제한은 인스턴스 제한과 별개입니다. 전용 호스트에서 실행하는 인스턴스는 인스턴스 제한 계산에 포함되지 않습니다.
- Auto Scaling 그룹은 지원되지 않습니다.
- Amazon RDS 인스턴스는 지원되지 않습니다.
- AWS 프리 티어는 전용 호스트에서 제공되지 않습니다.
- 인스턴스 배치 제어는 전용 호스트에서 인스턴스 시작을 관리하는 것을 말합니다. 배치 그룹은 전용 호스트에서 지원되지 않습니다.

전용 호스트 구성

전용 호스트는 단일 인스턴스 유형 및 크기 용량을 지원하도록 구성됩니다. 전용 호스트에서 시작할 수 있는 인스턴스 수는 해당 전용 호스트가 지원하도록 구성된 인스턴스 유형에 따라 달라집니다. 예를 들어 c3.xlarge 전용 호스트를 할당한 경우 전용 호스트에서 최대 8개의 c3.xlarge 인스턴스를 시작할 수 있습니다. 특정 전용 호스트에서 실행할 수 있는 인스턴스 유형 크기의 수를 확인하려면 [Amazon EC2 전용 호스트 요금](#) 섹션을 참조하십시오.

전용 호스트 사용

전용 호스트를 사용하려면 먼저 계정에서 사용할 호스트를 할당해야 합니다. 그런 다음, 인스턴스에 대해 host 테넌시를 지정하여 호스트에서 인스턴스를 시작합니다. 인스턴스 자동 배치 설정을 사용하면 인스턴스를 특정 호스트에서 시작할지 여부를 제어할 수 있습니다. 인스턴스를 중지했다 다시 시작하는 경우 호스트 선호도 설정이 해당 인스턴스를 동일한 또는 다른 호스트에서 다시 시작할지 여부를 결정합니다. 온디맨드 호스트가 더 이상 필요하지 않을 경우 해당 호스트에서 실행 중인 인스턴스를 중지하고 다른 호스트에서 시작하도록 지시한 후 전용 호스트를 해제합니다.

목차

- [기존 보유 라이선스 사용 \(p. 249\)](#)
- [전용 호스트 할당 \(p. 250\)](#)
- [전용 호스트에서 인스턴스 시작 \(p. 250\)](#)
- [인스턴스 배치와 호스트 선호도의 이해 \(p. 252\)](#)
- [인스턴스 테넌시 수정 \(p. 253\)](#)
- [전용 호스트 관리 및 해제 \(p. 254\)](#)
- [API 및 CLI 명령 개요 \(p. 254\)](#)
- [AWS Config를 사용하여 구성 변경 추적하기 \(p. 255\)](#)

기존 보유 라이선스 사용

전용 호스트에서 사용자의 소프트웨어 라이선스를 사용할 수 있습니다. 다음은 Amazon EC2에서 기존 볼륨 라이선스 머신 이미지를 사용하려면 수행해야 할 일반 단계입니다.

1. 머신 이미지(AMI) 사용을 제어하는 라이선스 조건이 가상 클라우드 환경에서 머신 이미지 사용을 허용하는지 확인합니다. Microsoft 라이선싱에 대한 자세한 내용은 [Amazon Web Services and Microsoft Licensing](#) 섹션을 참조하십시오.
2. 머신 이미지를 Amazon EC2에서 사용 가능하지 확인했으면 VM Import/Export 도구로 생성할 수 있는 ImportImage API 작업을 사용하여 머신 이미지를 가져옵니다. 제약 및 제한에 대한 자세한 내용은 [VM Import/Export 사전 조건](#)을 참조하십시오. ImportImage를 사용하여 VM을 가져오는 방법에 대한 자세한 내용은 [ImportImage를 사용하여 Amazon EC2로 VM 가져오기](#) 섹션을 참조하십시오.
3. AWS에서 사용된 이미지를 추적하는 메커니즘이 필요한 경우 AWS Config 서비스에서 호스트 기록을 사용합니다. AWS Config를 사용하여 전용 호스트에 대한 구성 변경을 기록하고 출력을 라이선스 보고용 데

이터 소스로 사용할 수 있습니다. 자세한 내용은 [AWS Config를 사용하여 구성 변경 추적하기 \(p. 255\)](#) 섹션을 참조하십시오.

4. 머신 이미지를 가져온 후 계정에 활성화된 전용 호스트에서 이 이미지의 인스턴스를 시작할 수 있습니다.
5. 이러한 인스턴스를 실행할 때 운영 체제에 따라 자체 KMS 서버(예: Windows Server 또는 Windows SQL Server)에 대해 해당 인스턴스를 활성화해야 할 수 있습니다. 가져온 Windows AMI는 Amazon Windows KMS 서버에 대해 활성화할 수 없습니다.

전용 호스트 할당

전용 호스트 사용을 시작하려면 계정에 할당해야 합니다. AWS Management Console을 사용하여 API와 직접 상호 작용하거나 명령줄 인터페이스를 사용하여 이러한 작업을 수행할 수 있습니다. 전용 호스트를 할당할 때마다 다음 단계를 수행합니다.

계정에 전용 호스트를 할당하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. [Dedicated Hosts] 페이지에서 [Allocate Dedicated Host]를 선택합니다.
3. 제공된 옵션을 사용하여 호스트를 구성합니다.
 - a. [Instance type] - 전용 호스트에서 사용 가능한 인스턴스 유형.
 - b. [Availability Zone] - 전용 호스트용 가용 영역.
 - c. [Allow instance auto-placement] - 기본 설정은 [Off]입니다. 전용 호스트는 host 테넌시 인스턴스 시작만 수용합니다(가용 용량이 있을 경우). 인스턴스 자동 배치가 [On] 상태인 경우에는 host의 테넌시가 있고 전용 호스트의 구성과 일치하는 모든 인스턴스는 그 호스트에서 시작될 수 있습니다.
 - d. Quantity-이러한 설정으로 할당하려는 호스트의 수입니다.
4. [Allocate host]를 선택합니다.

계정에서 전용 호스트 용량을 즉시 사용할 수 있게 됩니다.

계정에 활성화된 전용 호스트가 없는 상태에서 테넌시가 host인 인스턴스를 시작할 경우 오류가 발생하고 인스턴스 시작에 실패합니다.

전용 호스트에서 인스턴스 시작

전용 호스트를 할당한 후 여기에서 인스턴스를 시작할 수 있습니다. 테넌시가 host인 인스턴스는 특정 전용 호스트에서 시작하거나 Amazon EC2가 자동으로 적절한 전용 호스트를 선택할 수 있습니다(자동 배치). 시작하려는 인스턴스의 인스턴스 유형 구성과 일치하는 가용 용량이 있는 활성 전용 호스트가 사용자 계정에 없으면 테넌시가 host인 인스턴스를 시작할 수 없습니다.

Note

전용 호스트에서 시작되는 인스턴스는 VPC에서만 시작할 수 있습니다. 자세한 내용은 [VPC 소개](#) 섹션을 참조하십시오.

인스턴스를 시작하기 전에 제한 사항에 유의하십시오. 자세한 내용은 [전용 호스트 제한 및 제약 \(p. 248\)](#) 섹션을 참조하십시오.

전용 호스트 페이지를 사용하여 전용 호스트에서 인스턴스 시작

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. [Dedicated Hosts] 페이지에서 호스트를 선택하고 [Actions]를 선택한 후 [Launch Instance(s) onto Host]를 선택합니다.
3. 사용할 AMI를 선택합니다. 자체 AMI를 가져온 경우 왼쪽 사이드바에서 [My AMIs]를 선택하고 해당 AMI를 선택합니다.

4. 전용 호스트에 대해 인스턴스 유형을 선택합니다. 이 유형의 인스턴스만 해당 호스트에서 시작할 수 있습니다.
5. [Configure Instance Details] 페이지에 [Tenancy] 및 [Host] 옵션이 미리 선택되어 있습니다. [Affinity] 설정을 [On] 또는 [Off]로 전환할 수 있습니다.
 - On-중지되면, 인스턴스가 항상 특정 호스트에서 다시 시작합니다.
 - Off-인스턴스가 지정된 전용 호스트에서 시작하지만, 중지될 경우 반드시 그 호스트에서 다시 시작하지는 않습니다.
6. 나머지 단계를 완료하고 [Launch Instances]를 선택합니다.

지정한 전용 호스트에서 인스턴스가 자동으로 시작합니다. 전용 호스트에서 인스턴스를 보려면 [Dedicated Hosts] 페이지로 이동하여 인스턴스를 시작할 때 지정한 전용 호스트를 선택합니다.

인스턴스 페이지를 사용하여 특정한 전용 호스트에서 인스턴스 시작

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. [Instances] 페이지에서 [Launch Instance]를 선택합니다.
3. 목록에서 AMI를 선택합니다. 자체 AMI를 가져온 경우, [My AMIs]를 선택한 후 가져온 이미지를 선택합니다. 일부 AMI는 전용 호스트에서 사용할 수 없습니다.
4. 시작할 인스턴스의 유형을 선택합니다.
5. [Configure Instance Details] 페이지에서 전용 호스트 설정은 다음과 같습니다.
 - Tenancy - Dedicated host - Launch this instance on a Dedicated host. 이 항목을 선택할 수 없다면, 호환되지 않는 AMI나 인스턴스 유형을 선택했는지 확인하십시오.
 - [Host]-호스트를 선택합니다. 전용 호스트를 선택할 수 없는 경우 확인해야 할 사항:
 - 선택한 서브넷이 호스트의 다른 가용 영역에 있는지 확인합니다.
 - 선택한 인스턴스 유형이 전용 호스트가 지원하는 인스턴스 유형과 일치하는지 확인합니다. 일치하는 실행 호스트가 없는 경우 유일한 방법은 [Use auto-placement]를 사용하는 것입니다. 하지만 사용자의 계정에 일치하는 전용 호스트 용량이 없을 경우 인스턴스 실행에 실패합니다.
 - [Affinity] - 이에 대한 기본 설정은 [Off]입니다. 인스턴스가 지정된 전용 호스트에서 시작하지만, 중지될 경우 반드시 그 호스트에서 다시 시작하지는 않습니다.

Note

이러한 설정이 보이지 않으면 [Network] 메뉴에서 VPC를 선택했는지 확인하십시오.

6. 나머지 구성 단계를 완료합니다. [Review and Launch]를 선택합니다.
7. [Launch]를 선택하여 인스턴스를 시작합니다.
8. 기존 키 페어를 선택하거나 새로 생성합니다. [Launch Instances]를 선택합니다.

인스턴스 페이지를 사용하여 원하는 전용 호스트에서 인스턴스 시작

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. [Instances] 페이지에서 [Launch Instance]를 선택합니다.
3. 목록에서 AMI를 선택합니다. 자체 AMI를 가져온 경우, [My AMIs]를 선택한 후 가져온 이미지를 선택합니다. 일부 AMI는 전용 호스트에서 사용할 수 없습니다.
4. 시작할 인스턴스의 유형을 선택합니다.
5. [Configure Instance Details] 페이지에서 전용 호스트 설정은 다음과 같습니다.
 - [Tenancy]-[Dedicated host – Launch this instance on a Dedicated host] 이 항목을 선택할 수 없다면, 호환되지 않는 AMI나 인스턴스 유형을 선택했는지 확인하십시오.
 - [Host]-이 유형으로 시작하려면 설정을 [Use auto-placement]로 유지합니다.

- [Affinity] - 이에 대한 기본 설정은 [Off]입니다. 인스턴스가 계정에서 사용 가능한 모든 전용 호스트에서 시작할 수 있지만, 중지될 경우 반드시 해당 호스트에서 다시 시작하지는 않습니다.

이러한 설정이 보이지 않으면 [Network] 메뉴에서 VPC를 선택했는지 확인하십시오.

6. 나머지 구성 단계를 완료합니다. [Review and Launch]를 선택합니다.
7. [Launch]를 선택하여 인스턴스를 시작합니다.
8. 기존 키 페어를 선택하거나 새로 생성합니다. [Launch Instances]를 선택합니다.

인스턴스 배치와 호스트 선호도의 이해

배치 제어는 인스턴스 수준과 호스트 수준에서 모두 이루어집니다.

목차

- [인스턴스 자동 배치 \(p. 252\)](#)
- [호스트 선호도 \(p. 252\)](#)
- [인스턴스 자동 배치와 호스트 선호도의 설정 \(p. 252\)](#)
- [인스턴스 호스트 선호도 설정 \(p. 253\)](#)

인스턴스 자동 배치

자동 배치를 사용하면 인스턴스를 특정 호스트에서 시작할 것인지, 일치하는 구성의 원하는 호스트에서 시작할 것인지 선택할 수 있습니다. 이에 대한 기본 설정은 [Off]입니다. 즉, 사용자가 할당하는 전용 호스트는 고유한 호스트 ID를 지정하는 `host` 테넌시 인스턴스 시작만 수락합니다. 지정된 호스트 ID 없이 시작한 인스턴스는 인스턴스 자동 배치가 Off로 설정된 호스트에서 시작할 수 없습니다.

호스트 선호도

호스트 선호도는 인스턴스와 전용 호스트 사이의 시작 관계를 설정합니다. 선호도를 `host`로 설정하면 특정 호스트에서 시작한 인스턴스가 중단된 경우 항상 동일한 호스트에서 다시 시작합니다. 대상 지정 및 대상 미지정 시작에 모두 적용됩니다.

선호도가 `default`로 설정된 상태에서 인스턴스를 중지했다 다시 시작할 경우 인스턴스는 어떤 가용 호스트에서도 다시 시작될 수 있지만, 마지막으로 해당 인스턴스를 실행한 전용 호스트에서 다시 시작하려고 합니다(최선의 노력 기준).

선호도를 `host`에서 `default`로 또는 반대로 변경하여 인스턴스와 전용 호스트 간 관계를 수정할 수 있습니다. 자세한 내용은 [인스턴스 테넌시 설정 \(p. 253\)](#) 섹션을 참조하십시오.

인스턴스 자동 배치와 호스트 선호도의 설정

Amazon EC2 콘솔, API 또는 CLI를 사용하여 인스턴스 배치 제어를 관리할 수 있습니다.

인스턴스의 인스턴스 배치 설정을 수정하려면 우선 인스턴스를 중지한 후 인스턴스 배치 설정을 편집합니다.

Note

인스턴스를 중지했다가 다시 시작할 경우 동일한 전용 호스트에서 다시 시작된다는 보장은 없습니다.

인스턴스의 배치 설정을 편집하려면(임의의 가용 호스트)

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. [Instances] 페이지에서 편집할 인스턴스를 선택합니다.
3. [Actions], [Instance State], [Stop]을 차례로 선택합니다.

4. [Actions], [Instance Settings]와 [Modify Instance Placement]를 선택합니다.
5. 인스턴스 테넌시를 [Launch this instance on a Dedicated host]로 변경합니다.
6. [This instance can run on any one of my Hosts]를 선택합니다. 자동 배치가 활성화된 전용 호스트에서 인스턴스가 시작됩니다.
7. [Save]를 선택하여 계속 진행합니다.
8. 인스턴스에 대한 컨텍스트(오른쪽 클릭) 메뉴를 열고 [Instance State], [Start]를 선택합니다.

인스턴스의 배치 설정을 편집하려면(특정한 전용 호스트)

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. [Instances] 페이지에서 편집할 인스턴스를 선택합니다.
3. [Actions], [Instance State], [Stop]을 차례로 선택합니다.
4. [Actions], [Instance Settings]와 [Modify Instance Placement]를 선택합니다.
5. 인스턴스 테넌시를 [Launch this instance on a Dedicated host]로 변경합니다.
6. [This instance can only run on the selected Host]를 선택합니다. 그런 다음 [Target Host]의 값을 선택하고 인스턴스를 임의의 가용 호스트에 배치할지, 특정 호스트에 배치할지 선택합니다.
7. [Save]를 선택하여 계속 진행합니다.
8. 인스턴스에 대한 컨텍스트(오른쪽 클릭) 메뉴를 열고 [Instance State], [Start]를 선택합니다.

인스턴스 호스트 선호도 수정

인스턴스와 호스트 사이의 선호도를 더 이상 원치 않을 경우 인스턴스를 중지하고 선호도를 default로 변경할 수 있습니다. 그러면 인스턴스와 호스트 간 지속성이 제거됩니다. 하지만 인스턴스를 다시 시작하면 동일한 전용 호스트(사용자 계정의 전용 호스트 가용성에 따라 최선의 노력을 기준으로)에서 다시 시작합니다. 그러나 다시 중지되면 동일한 호스트에서 다시 시작하지 않습니다.

인스턴스 테넌시 수정

전용 인스턴스의 테넌시를 dedicated에서 host로 변경할 수 있고, Amazon EC2에서 제공하는 Windows, SUSE 또는 RHEL AMI를 사용하지 않는 경우 그 반대로 변경할 수 있습니다. 이를 위해 전용 인스턴스를 중지해야 합니다. shared 테넌시로 시작된 인스턴스는 host 테넌시로 변경할 수 없습니다.

인스턴스 테넌시를 dedicated에서 host로 수정하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. [Instances]를 선택한 후 수정할 전용 인스턴스를 선택합니다.
3. [Actions], [Instance State], [Stop]을 차례로 선택합니다.
4. 인스턴스에 대한 컨텍스트(오른쪽 클릭) 메뉴를 열고 [Instance Settings], [Modify Instance Placement]를 차례로 선택합니다.
5. [Modify Instance Placement] 페이지에서 다음 절차를 따릅니다.
 - [Tenancy]–[Launch this instance on a Dedicated host]를 선택합니다.
 - [Affinity]–[This instance can run on any one of my Hosts] 또는 [This instance can only run on the selected Host]를 선택합니다.

[This instance can run on any one of my Hosts]를 선택한 경우 인스턴스는 사용자 계정에 있는 호환되는 전용 호스트에서 시작됩니다.

[This instance can only run on the selected Host]를 선택할 경우 [Target Host]에 입력할 값을 선택합니다. 대상 호스트 목록이 표시되지 않는 경우 계정 내에 호환되는 전용 호스트가 없다는 뜻입니다.

6. [Save]를 선택합니다.

7. 인스턴스를 다시 시작하면 Amazon EC2가 사용자 계정 내에서 가용한 전용 호스트에 인스턴스를 배치합니다. 단, 사용자가 시작하는 인스턴스 유형을 호스트가 지원해야 합니다.

전용 호스트 관리 및 해제

콘솔을 사용하거나 API를 통해 직접 작업하거나 명령줄 인터페이스를 사용하여 호스트 상의 개별 인스턴스에 대한 세부 정보를 보고 온디맨드 전용 호스트를 해제할 수 있습니다.

전용 호스트에서 인스턴스의 세부 정보를 확인하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. [Dedicated Hosts] 페이지에서 자세한 정보를 보려는 호스트를 선택합니다.
3. 호스트 정보를 보려면 [Description] 탭을 선택합니다. 호스트에서 실행 중인 인스턴스의 정보를 보려면 [Instances] 탭을 선택합니다.

전용 호스트를 해제하려면

전용 호스트에서 실행되는 모든 인스턴스를 중지해야 해당 호스트를 해제할 수 있습니다. 이 인스턴스들을 계정의 다른 전용 호스트로 마이그레이션하여 계속 사용할 수 있습니다. 자세한 내용은 [인스턴스 자동 배치와 호스트 선호도의 설정 \(p. 252\)](#) 섹션을 참조하십시오. 이 단계들은 온디맨드 전용 호스트에만 적용됩니다.

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. [Dedicated Hosts] 페이지에서 해제할 전용 호스트를 선택합니다.
3. [Actions], [Release Hosts]를 선택합니다.
4. [Release]를 선택하여 선택 항목을 확인합니다.

전용 호스트를 해제한 후에는 동일한 호스트 또는 호스트 ID를 다시 사용할 수 없습니다.

전용 호스트가 해제되면 더 이상 해당 호스트에 대해 온디맨드 결제 요금이 부과되지 않습니다. 전용 호스트 상태가 `released`로 변경되고 이 호스트에서 인스턴스를 시작할 수 없게 됩니다.

최근에 전용 호스트를 해제한 경우, 제한 계산에서 제외될 때까지 시간이 약간 걸릴 수 있습니다. 이 시간 동안 새로운 전용 호스트 할당을 시도할 경우 `LimitExceeded` 오류가 발생할 수 있습니다. 이런 경우 몇 분 후에 새 호스트를 할당해 보십시오.

중지된 인스턴스는 계속 사용할 수 있으며 [Instances] 페이지에 나열됩니다. 또한 `host` 테넌시 설정을 유지합니다.

API 및 CLI 명령 개요

API 또는 명령줄을 사용하여 이 섹션에서 설명하는 작업을 수행할 수 있습니다.

계정에 전용 호스트를 할당하려면

- [allocate-hosts](#)(AWS CLI)
- [AllocateHosts](#)(Amazon EC2 쿼리 API)
- [New-EC2Hosts](#)(Windows PowerShell용 AWS 도구)

전용 호스트를 설명하려면

- [describe-hosts](#)(AWS CLI)
- [DescribeHosts](#)(Amazon EC2 쿼리 API)

- [Get-EC2Hosts](#)(Windows PowerShell용 AWS 도구)

전용 호스트를 변경하려면

- [modify-hosts](#)(AWS CLI)
- [ModifyHosts](#)(Amazon EC2 쿼리 API)
- [Edit-EC2Hosts](#)(Windows PowerShell용 AWS 도구)

인스턴스 자동 배치를 수정하려면

- [modify-instance-placement](#)(AWS CLI)
- [ModifyInstancePlacement](#)(Amazon EC2 쿼리 API)
- [Edit-EC2InstancePlacement](#) (Windows PowerShell용 AWS 도구)

전용 호스트를 해제하려면

- [release-hosts](#)(AWS CLI)
- [ReleaseHosts](#)(Amazon EC2 쿼리 API)
- [Remove-EC2Hosts](#)(Windows PowerShell용 AWS 도구)

AWS Config를 사용하여 구성 변경 추적하기

AWS Config를 사용하여 전용 호스트, 그리고 이 전용 호스트에서 시작, 중지 또는 종료된 인스턴스의 구성 변경 사항을 기록할 수 있습니다. 그런 다음 AWS Config가 캡처한 정보를 라이선스 보고용 데이터 소스로 사용할 수 있습니다.

AWS Config는 전용 호스트 및 인스턴스의 구성 정보를 개별적으로 기록하고 관계를 통해 이 정보를 페어링합니다. 보고 조건은 세 가지가 있습니다.

- AWS Config recording status - [On]으로 설정 시 AWS Config가 전용 호스트 및 전용 인스턴스를 비롯하여 하나 이상의 AWS 리소스 유형을 기록합니다. 라이선스 보고에 필요한 정보를 캡처하려면 다음 필드에서 호스트 및 인스턴스가 기록되는지 확인합니다.
- Host recording status-[Enabled]로 설정 시 전용 호스트 구성 정보가 기록됩니다.
- Instance recording status-[Enabled]로 설정 시 전용 인스턴스 구성 정보가 기록됩니다.

세 조건 중 하나라도 비활성화되면 [Edit Config Recording] 버튼의 아이콘이 빨간색으로 표시됩니다. 이 도구의 이점을 최대한 활용하려면 세 기록 방법을 모두 활성화하십시오. 세 방법이 모두 활성화되면 아이콘이 녹색으로 표시됩니다. 설정을 편집하려면 [Edit Config Recording]을 선택합니다. 그러면 AWS Config 콘솔의 [Set up AWS Config] 페이지로 이동하며, 여기서 AWS Config를 설정하고 호스트, 인스턴스 및 기타 지원되는 리소스 유형에 대한 기록을 시작할 수 있습니다. 자세한 내용은 AWS Config 개발자 안내서의 [Setting up AWS Config using the Console](#) 섹션을 참조하십시오.

Note

AWS Config가 리소스를 발견하여 기록을 시작합니다. 이 과정은 몇 분이 걸릴 수 있습니다.

AWS Config가 호스트 및 인스턴스 구성 변경을 기록하기 시작한 후, 설정 또는 해제한 호스트와 시작, 중지 또는 종료한 인스턴스의 구성 내역을 확인할 수 있습니다. 예를 들어 전용 호스트 구성 내역의 특정 시점에서 호스트에서 몇 개의 인스턴스가 시작되었는지 여부를 호스트의 소켓 및 코어 수와 함께 확인할 수 있습니다. 이러한 인스턴스 각각에 대해 Amazon 머신 이미지(AMI)의 ID를 조회할 수도 있습니다. 이 정보를 이용하여 소켓당 또는 코어당 라이선스된 서버 한정 소프트웨어에 대한 라이선스를 보고할 수 있습니다.

다음 방법 중 하나를 사용하여 구성 내역을 볼 수 있습니다.

- AWS Config 콘솔 사용. 기록된 리소스 각각에 대해 구성 세부 정보의 내역을 제공하는 타임라인 페이지를 볼 수 있습니다. 이 페이지를 보려면 [Dedicated Hosts] 페이지의 [Config Timeline] 열에서 회색 아이콘을 선택합니다. 보다 자세한 내용은 AWS Config 개발자 안내서의 [Viewing Configuration Details in the AWS Config Console](#) 섹션을 참조하십시오.
- AWS CLI 명령 실행. 먼저 `list-discovered-resources` 명령을 사용하여 모든 호스트 및 인스턴스의 목록을 가져올 수 있습니다. 그런 다음 `get-resource-config-history` 명령을 사용하여 특정 기간에 대해 특정 호스트 또는 인스턴스의 구성 세부 정보를 가져올 수 있습니다. 보다 자세한 내용은 AWS Config 개발자 안내서의 [View Configuration Details Using the CLI](#) 섹션을 참조하십시오.
- 애플리케이션에서 AWS Config API 사용. 먼저 `ListDiscoveredResources` 작업을 사용하여 모든 호스트 및 인스턴스의 목록을 가져올 수 있습니다. 그런 다음 `GetResourceConfigHistory` 작업을 사용하여 특정 기간에 대해 특정 호스트 또는 인스턴스의 구성 세부 정보를 가져올 수 있습니다.

예를 들어 AWS Config에서 모든 전용 호스트의 목록을 가져오려면 다음과 같은 CLI 명령을 실행합니다.

```
aws configservice list-discovered-resources --resource-type
    AWS::EC2::Host
```

AWS Config에서 특정 전용 호스트의 구성 내역을 가져오려면 다음과 같은 CLI 명령을 실행합니다.

```
aws configservice get-resource-config-history --resource-type
    AWS::EC2::Instance --resource-id i-36a47fdf
```

AWS Management Console을 사용하여 AWS Config 설정을 관리하는 방법

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. [Dedicated Hosts] 페이지에서 [Edit Config Recording]을 선택합니다.
3. AWS Config 콘솔에서 제공되는 단계를 수행하여 기록을 캡니다. 자세한 내용은 [Setting up AWS Config using the Console](#) 섹션을 참조하십시오.

자세한 내용은 [Viewing Configuration Details in the AWS Config Console](#) 섹션을 참조하십시오.

명령줄 또는 API를 사용하여 AWS Config를 활성화하려면

- AWS CLI를 사용하려면 AWS Config 개발자 안내서의 [Viewing Configuration Details in the AWS Config Console](#) 섹션을 참조하십시오.
- Amazon EC2 API를 사용하려면 [GetResourceConfigHistory](#)를 참조하십시오.

전용 호스트 모니터링

Amazon EC2는 전용 호스트의 상태를 지속적으로 모니터링하고 Amazon EC2 콘솔에서 업데이트가 전달됩니다. 또한 API 또는 CLI를 사용하여 전용 호스트에 대한 정보를 확인할 수도 있습니다.

다음 표에는 콘솔에서 볼 수 있는 [State] 값에 대한 설명이 나와 있습니다.

시/도	설명
available	AWS가 전용 호스트에서 발견한 문제가 없습니다. 예약된 유지 관리 또는 수리가 없습니다. 이 전용 호스트에서 인스턴스를 시작할 수 있습니다.
released	전용 호스트가 해제되었습니다. 더 이상 이 호스트 ID가 사용되지 않습니다. 해제된 호스트는 다시 사용할 수 없습니다.

시/도	설명
under-assessment	AWS가 전용 호스트에 있을 수 있는 문제를 탐색 중입니다. 작업이 필요할 경우 AWS Management Console 또는 이메일을 통해 통보됩니다. 이 상태에서는 전용 호스트에서 인스턴스를 시작할 수 없습니다.
permanent-failure	복구할 수 없는 오류가 감지되었습니다. 인스턴스 및 이메일을 통해 제거 알림이 제공됩니다. 인스턴스는 계속 실행할 수 있습니다. 이 상태의 전용 호스트에서 모든 인스턴스를 중지 또는 종료할 경우 AWS가 해당 호스트를 사용 중지합니다. 이 상태에서는 전용 호스트에서 인스턴스를 시작할 수 없습니다.
released-permanent-failure	AWS가 오류가 발생한 전용 호스트를 영구 해제하여 더 이상 인스턴스가 실행되지 못하도록 합니다. 전용 호스트 ID를 더 이상 사용할 수 없습니다.

전용 인스턴스

전용 인스턴스는 단일 고객에게만 할당된 하드웨어의 Virtual Private Cloud(VPC)에서 실행되는 Amazon EC2 인스턴스입니다. 전용 인스턴스는 호스트 하드웨어 수준에서 다른 AWS 계정에 속하는 인스턴스로부터 물리적으로 격리됩니다. 전용 인스턴스는 전용 인스턴스가 아닌 동일한 AWS 계정의 다른 인스턴스와 하드웨어를 공유할 수 있습니다.

Note

또한 전용 호스트는 고객 전용의 물리적 서버입니다. 전용 호스트를 사용하여 서버에서 인스턴스의 배치 방법을 확인 및 제어할 수 있습니다. 자세한 내용은 [전용 호스트 \(p. 246\)](#) 섹션을 참조하십시오.

항목

- [전용 인스턴스 기본 사항 \(p. 257\)](#)
- [전용 인스턴스 사용 \(p. 259\)](#)
- [API 및 명령 개요 \(p. 260\)](#)

전용 인스턴스 기본 사항

VPC에서 실행하는 각 인스턴스는 테넌시 속성으로 실행됩니다. 이 속성에는 다음과 같은 값이 있습니다.

값	설명
default	인스턴스가 공유된 하드웨어에서 실행됩니다.
dedicated	인스턴스가 단일 테넌트 하드웨어에서 실행됩니다.
host	인스턴스는 구성을 제어할 수 있는 격리된 서버인 전용 호스트에서 실행됩니다.

시작한 이후에는 기본 인스턴스의 테넌시를 변경할 수 없습니다. 시작한 이후에 인스턴스의 테넌시를 [dedicated](#)에서 [host](#)로 변경할 수 있으며 그 반대의 경우도 마찬가지입니다. 자세한 내용은 [인스턴스의 테넌시 변경 \(p. 260\)](#) 섹션을 참조하십시오.

각 인스턴스에는 관련 인스턴스 테넌시 속성이 있습니다. 인스턴스를 생성한 후 VPC의 인스턴스 테넌시를 변경할 수는 없습니다. 이 속성에는 다음과 같은 값이 있습니다.

값	설명
default	인스턴스 시작 중 다른 테넌트를 명시적으로 지정하지 않는 한, VPC로 시작된 인스턴스는 기본적으로 공유 하드웨어에서 실행됩니다.
dedicated	인스턴스 시작 중 host의 테넌트를 명시적으로 지정하지 않는 한, VPC로 시작된 인스턴스는 기본적으로 전용 인스턴스입니다. 인스턴스 시작 중 default의 테넌트를 지정할 수 없습니다.

전용 인스턴스를 생성하려면 다음 작업을 수행할 수 있습니다.

- 인스턴스 테넌시를 dedicated로 설정하여 VPC를 생성합니다. 이 VPC에서 시작된 모든 인스턴스는 전용 인스턴스입니다.
- default로 설정된 인스턴스 테넌시로 VPC를 생성하고 시작할 때 모든 인스턴스에 대해 dedicated 테넌시를 지정합니다.

전용 인스턴스의 제한 사항

일부 AWS 서비스나 그 기능은 인스턴스 테넌시가 dedicated로 설정된 VPC에서 작동하지 않습니다. 서비스 설명서에서 이에 관한 제한 사항이 있는지 확인하십시오.

일부 인스턴스 유형은 인스턴스 테넌시가 dedicated로 설정된 VPC에서 시작할 수 없습니다. 지원되는 인스턴스 유형에 대한 자세한 내용은 [Amazon EC2 전용 인스턴스](#)를 참조하십시오.

Amazon EBS 전용 인스턴스

Amazon EBS 지원 전용 인스턴스를 시작할 경우 EBS 볼륨은 단일 테넌트 하드웨어에서 실행되지 않습니다.

예약 인스턴스 전용 테넌시

전용 인스턴스를 시작하기 위한 용량을 충분히 확보하려면 전용 예약 인스턴스를 구입하면 됩니다. 자세한 내용은 [예약 인스턴스 \(p. 174\)](#) 섹션을 참조하십시오.

전용 예약 인스턴스를 구입하면 대폭 할인된 사용 요금으로 전용 인스턴스를 VPC에서 시작할 수 있는 용량이 제공됩니다. 시간당 요금 인하는 전용 테넌시로 인스턴스를 시작할 경우에만 적용됩니다. 하지만 기본 테넌시 값을 포함하는 예약 인스턴스를 구입하면 dedicated 인스턴스 테넌시로 인스턴스를 시작할 경우 전용 예약 인스턴스를 받지 못합니다.

또한 예약 인스턴스를 구입한 후에는 예약 인스턴스의 테넌시를 변경할 수 없습니다.

전용 인스턴스의 Auto Scaling

Auto Scaling을 사용하여 전용 인스턴스를 시작하는 방법은 Auto Scaling 사용 설명서에서 [Amazon Virtual Private Cloud의 Auto Scaling](#)을 참조하십시오.

전용 스팟 인스턴스

스팟 인스턴스 요청을 생성할 때 dedicated의 테넌시를 지정하여 전용 스팟 인스턴스를 실행할 수 있습니다. 자세한 내용은 [스팟 인스턴스의 테넌시 지정 \(p. 214\)](#) 섹션을 참조하십시오.

전용 인스턴스 요금

전용 인스턴스 요금은 온디맨드 인스턴스 요금과 다릅니다. 자세한 내용은 [Amazon EC2 전용 인스턴스 제품 페이지](#)를 참조하십시오.

전용 인스턴스 사용

전용 인스턴스 테넌시로 VPC를 생성하여 해당 VPC로 시작되는 모든 인스턴스가 전용 인스턴스가 되게 합니다. 또는 시작되는 동안 인스턴스의 테넌시를 지정할 수 있습니다.

항목

- [전용 인스턴스 테넌시의 VPC 생성하기 \(p. 259\)](#)
- [VPC에서 전용 인스턴스 시작 \(p. 259\)](#)
- [테넌시 정보 조회 \(p. 260\)](#)
- [인스턴스의 테넌시 변경 \(p. 260\)](#)

전용 인스턴스 테넌시의 VPC 생성하기

VPC를 생성할 경우 VPC의 인스턴스 테넌시를 지정하는 옵션이 제공됩니다. VPC 마법사를 사용하거나 Amazon VPC 콘솔의 Your VPCs 페이지에서 VPC를 생성할 수 있습니다.

전용 인스턴스 테넌시로 VPC를 생성하려면(VPC 마법사)

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 대시보드에서 [Start VPC Wizard]를 선택합니다.
3. VPC 구성을 선택한 후 [Select]를 선택합니다.
4. 마법사 다음 페이지의 [Hardware tenancy] 목록에서 [Dedicated]를 선택합니다.
5. [Create VPC]를 선택합니다.

전용 인스턴스 테넌시로 VPC를 생성하려면(Create VPC 대화 상자)

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Your VPCs]와 [Create VPC]를 차례로 선택합니다.
3. Tenancy에서 Dedicated를 선택합니다. CIDR 블록을 지정하고 Yes, Create를 선택합니다.

dedicated 인스턴스 테넌시가 있는 VPC에서 인스턴스를 시작하면 인스턴스 테넌시와 상관없이 인스턴스가 자동으로 전용 인스턴스가 됩니다.

VPC에서 전용 인스턴스 시작

이제 Amazon EC2 시작 인스턴스 마법사를 사용하여 전용 인스턴스를 시작할 수 있습니다.

전용 테넌시를 포함하는 인스턴스를 기본 테넌시를 포함하는 VPC로 시작하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. [Launch Instance]를 선택합니다.
3. [Choose an Amazon Machine Image (AMI)] 페이지에서 AMI를 선택한 다음 [Select]를 선택합니다.
4. Choose an Instance Type 페이지에서 인스턴스 유형과 Next: Configure Instance Details를 차례대로 선택합니다.

Note

전용 인스턴스를 지원하는 인스턴스 유형을 선택해야 합니다. 자세한 내용은 [Amazon EC2 전용 인스턴스](#)를 참조하십시오.

5. [Configure Instance Details] 페이지에서 VPC와 서브넷을 선택합니다. [Tenancy] 목록에서 [Dedicated - Run a dedicated instance]와 [Next: Add Storage]를 차례로 선택합니다.
6. 마법사에 표시되는 지침에 따라 계속합니다. [Review Instance Launch] 페이지에서 옵션을 모두 검토했으면 [Launch]를 선택하여 키 페어를 선택하고 전용 인스턴스를 시작합니다.

host 테넌시를 사용하여 인스턴스를 시작하는 자세한 내용은 [전용 호스트에서 인스턴스 시작 \(p. 250\)](#) 섹션을 참조하십시오.

테넌시 정보 조회

VPC의 테넌시 정보를 조회하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Your VPCs를 선택합니다.
3. [Tenancy] 열에서 해당 VPC의 인스턴스 테넌시를 확인합니다.
4. [Tenancy] 열이 표시되지 않는 경우 [Edit Table Columns](기어 모양 아이콘)를 선택하고 [Show/Hide Columns] 대화 상자에서 [Tenancy]를 선택한 다음 [Close]를 선택합니다.

인스턴스의 테넌시 정보를 조회하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. [Tenancy] 열에서 해당 인스턴스의 테넌시를 확인합니다.
4. [Tenancy] 열이 표시되지 않으면 다음 중 하나를 수행합니다.
 - [Edit Table Columns](기어 모양 아이콘)를 선택하고 [Show/Hide Columns] 대화 상자에서 [Tenancy]를 선택한 다음 [Close]를 선택합니다.
 - 인스턴스를 선택합니다. 세부 정보 창의 [Description] 탭에 인스턴스의 정보와 그 테넌시가 표시됩니다.

인스턴스의 테넌시 변경

인스턴스 유형과 플랫폼에 따라 시작된 이후에 중지된 전용 인스턴스의 테넌시를 host로 변경할 수 있습니다. 인스턴스를 다음에 시작하면 계정에 할당된 전용 호스트에서 시작됩니다. 전용 호스트를 할당 및 사용하는 방법과 전용 호스트에서 사용할 수 있는 인스턴스 유형에 대한 자세한 내용은 [전용 호스트 사용 \(p. 249\)](#) 섹션을 참조하십시오. 마찬가지로 시작된 이후에 중지된 전용 호스트 인스턴스의 테넌시를 dedicated로 변경할 수 있습니다. 인스턴스를 다음에 시작하면 Amazon에서 제어하는 단일 테넌트 하드웨어에서 시작됩니다.

인스턴스의 테넌시를 변경하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Instances]를 선택한 다음 인스턴스를 선택합니다.
3. [Actions], [Instance State] 및 [Stop]을 차례로 선택합니다.
4. [Actions]와 [Instance Settings]를 선택한 다음 [Modify Instance Placement]를 선택합니다.
5. [Tenancy] 목록에서 인스턴스를 전용 하드웨어에서 실행할지 전용 호스트에서 실행할지를 선택합니다. [Save]를 선택합니다.

API 및 명령 개요

명령줄 또는 API를 사용하여 이 페이지에서 설명하는 작업을 수행할 수 있습니다.

VPC를 생성할 때 테넌시 옵션 설정

- [create-vpc\(AWS CLI\)](#)
- [New-EC2Vpc\(Windows PowerShell용 AWS 도구\)](#)

VPC에서 시작한 인스턴스에 대해 지원되는 테넌시 옵션 설명

- [describe-vpcs](#)(AWS CLI)
- [Get-EC2Vpc](#)(Windows PowerShell용 AWS 도구)

시작 시 인스턴스의 테넌시 옵션 설정

- [run-instances](#)(AWS CLI)
- [New-EC2Instance](#)(Windows PowerShell용 AWS 도구)

인스턴스의 테넌시 값 설명

- [describe-instances](#) (AWS CLI)
- [Get-EC2Instance](#) (Windows PowerShell용 AWS 도구)

예약 인스턴스의 테넌시 값 설명

- [describe-reserved-instances](#)(AWS CLI)
- [Get-EC2ReservedInstance](#)(Windows PowerShell용 AWS 도구)

예약 인스턴스 제품의 테넌시 값 설명

- [describe-reserved-instances-offerings](#)(AWS CLI)
- [Get-EC2ReservedInstancesOffering](#)(Windows PowerShell용 AWS 도구)

인스턴스의 테넌시 값 수정

- [modify-instance-placement](#)(AWS CLI)
- [Edit-EC2InstancePlacement](#) (Windows PowerShell용 AWS 도구)

인스턴스 수명 주기

Amazon EC2에서 작업하여 시작부터 종료까지 인스턴스를 관리함으로써 인스턴스에 호스팅하는 사이트나 애플리케이션과 관련하여 고객에게 최상의 환경을 제공할 수 있습니다.

다음 그림은 인스턴스 상태 간 전환을 나타냅니다. 인스턴스 스토어 기반 인스턴스를 중지했다가 시작할 수 없습니다. 인스턴스 스토어 기반 인스턴스에 대한 자세한 내용은 [루트 디바이스 스토리지 \(p. 64\)](#) 섹션을 참조하십시오.

인스턴스 시작

인스턴스를 시작하면 인스턴스가 `pending` 상태로 전환됩니다. 시작 시 지정한 인스턴스 유형에 따라 인스턴스에 사용되는 호스트 컴퓨터의 하드웨어가 결정됩니다. 인스턴스는 시작 시 사용자가 지정한 Amazon 머신 이미지(AMI)를 사용하여 부팅됩니다. 인스턴스 사용이 준비되고 나면 인스턴스가 `running` 상태로 전환됩니다. 실행 중인 인스턴스에 연결하여 바로 앞에 있는 컴퓨터를 사용하는 것처럼 인스턴스를 사용할 수 있습니다.

인스턴스가 `running` 상태로 전환되는 즉시 인스턴스 실행이 지속된 각 시간 또는 부분 시간에 대해 비용이 청구됩니다. 인스턴스가 유휴 상태이고 인스턴스에 연결하지 않더라도 마찬가지입니다.

자세한 내용은 [인스턴스 시작 \(p. 264\)](#) 및 [Linux 인스턴스에 연결 \(p. 274\)](#) 섹션을 참조하십시오.

인스턴스 중지 및 시작(Amazon EBS 기반 인스턴스에 만 해당)

인스턴스가 상태 확인을 통과하지 못하거나 애플리케이션이 예상대로 실행되고 있지 않은 경우 또는 인스턴스의 루트 볼륨이 Amazon EBS 볼륨인 경우 인스턴스를 중지했다가 시작하여 문제를 해결해 볼 수 있습니다.

인스턴스를 중지하면 `stopping` 상태로 전환되고 나서 `stopped` 상태로 전환됩니다. 인스턴스를 중지하고 나면 인스턴스에 대해 시간당 사용 요금이나 데이터 전송 요금이 부과되지는 않지만 모든 Amazon EBS 볼륨에 대한 스토리지 요금은 부과됩니다. 인스턴스가 `stopped` 상태인 경우 인스턴스 유형을 비롯하여 인스턴스의 특정 속성을 수정할 수 있습니다.

인스턴스를 시작하면 인스턴스가 `pending` 상태로 전환되며, 대부분의 경우 AWS는 인스턴스를 새 호스트 컴퓨터로 이동합니다. (호스트 컴퓨터에 문제가 없으면 인스턴스는 같은 호스트 컴퓨터에 머물 수 있습니다). 인스턴스를 중지했다가 시작하면 이전 호스트 컴퓨터의 인스턴스 스토어 볼륨에 있는 데이터가 모두 손실됩니다.

인스턴스가 EC2-Classic에서 실행 중인 경우 새 프라이빗 IPv4 주소를 받게 됩니다. 다시 말해서 프라이빗 IPv4 주소와 연결된 탄력적 IP 주소(EIP)가 더 이상 인스턴스와 연결되어 있지 않다는 뜻입니다. 인스턴스가 EC2-VPC에서 실행 중인 경우 프라이빗 IPv4 주소가 유지됩니다. 즉, 새 프라이빗 IPv4 주소 또는 네트워크 인터페이스와 연결된 EIP가 여전히 인스턴스와 연결되어 있다는 의미입니다. 인스턴스에 IPv6 주소가 있는 경우 해당 IPv6 주소를 유지합니다.

`stopped`에서 `running`으로 인스턴스를 전환할 때마다 이러한 전환이 한 시간 내에 여러 번 일어나더라도 전체 인스턴스 시간 요금이 부과됩니다.

자세한 내용은 [인스턴스 중지 및 시작 \(p. 285\)](#) 섹션을 참조하십시오.

인스턴스 재부팅

Amazon EC2 콘솔, 명령줄 도구 및 Amazon EC2 API를 사용하여 인스턴스를 재부팅할 수 있습니다. Amazon EC2를 사용하여 인스턴스에서 운영 체제 재부팅 명령을 실행하는 대신 인스턴스를 재부팅하는 것이 좋습니다.

인스턴스를 재부팅하는 것은 운영 체제를 재부팅하는 것과 같습니다. 인스턴스가 동일 호스트 컴퓨터에서 유지되며 해당 퍼블릭 DNS 이름, 프라이빗 IP 주소 및 인스턴스 스토어 볼륨의 모든 데이터가 그대로 유지됩니다. 일반적으로 재부팅을 완료하는 데 몇 분 정도 소요되지만, 재부팅 소요 시간은 인스턴스 구성에 따라 달라집니다.

인스턴스를 재부팅해도 새 인스턴스 결제 시간이 시작되지는 않습니다.

자세한 내용은 [인스턴스 재부팅 \(p. 288\)](#) 섹션을 참조하십시오.

인스턴스 만료

AWS에서 인스턴스를 호스팅하는 기본 하드웨어의 복구 불가능한 장애가 검색되는 경우 인스턴스가 만료 대상으로 예약됩니다. 예약된 만료 날짜에 도달하면 인스턴스가 AWS에 의해 종지되거나 종료됩니다. 인스턴스 루트 디바이스가 Amazon EBS 볼륨인 경우 인스턴스가 종지되며 언제든지 이 인스턴스를 다시 시작할 수 있습니다. 인스턴스 루트 디바이스가 인스턴스 스토어 볼륨인 경우 인스턴스가 종료되어 다시 사용할 수 없습니다.

자세한 내용은 [인스턴스 만료 \(p. 289\)](#) 섹션을 참조하십시오.

인스턴스 종료

더 이상 인스턴스가 필요하지 않다고 판단되면 인스턴스를 종료할 수 있습니다. 인스턴스 상태가 `shutting-down` 또는 `terminated`로 변경되는 즉시 해당 인스턴스에 대한 요금 부과가 중지됩니다.

종료 방지 기능을 사용하는 경우 콘솔, CLI 또는 API를 사용하여 인스턴스를 종료할 수 없습니다.

인스턴스는 종료한 후에도 잠시 동안 콘솔에 표시되며 그 이후 항목이 자동으로 삭제됩니다. 또한 CLI 및 API를 사용하여 종료된 인스턴스를 설명할 수도 있습니다. 리소스(예: 태그)는 종료된 인스턴스에서 점차 연결 해제되므로 잠시 후 종료된 인스턴스에서 더 이상 보이지 않을 수 있습니다. You can't connect to or recover a terminated instance.

각각의 Amazon EBS 기반 인스턴스는 `InstanceInitiatedShutdownBehavior` 속성을 지원하는데, 이러한 속성은 인스턴스 자체 내에서 종료를 시작할 때 인스턴스가 중지되는지, 종료되는지를 제어합니다(예: Linux에서 `shutdown` 명령 사용). 기본 동작은 인스턴스를 중지하는 것입니다. 인스턴스가 실행 중이거나 종단된 상태에 있을 때 이 속성을 수정할 수 있습니다.

각각의 Amazon EBS 볼륨은 `DeleteOnTermination` 속성을 지원하는데, 이 속성은 연결된 인스턴스를 종료할 때 볼륨이 삭제되는지, 유지되는지를 제어합니다. 기본값은 루트 디바이스 볼륨을 삭제하고 다른 EBS 볼륨을 유지하는 것입니다.

자세한 내용은 [인스턴스 종료 \(p. 291\)](#) 섹션을 참조하십시오.

재부팅, 중지 및 종료의 차이

다음 표에는 인스턴스 재부팅, 중지 및 종료의 주요 차이점이 요약되어 있습니다.

특성	재부팅	중지/시작(Amazon EBS 기반 인스턴스에만 해당)	Terminate
호스트 컴퓨터	인스턴스가 동일 호스트 컴퓨터에서 유지됩니다.	인스턴스가 새 호스트 컴퓨터에서 실행됩니다.	없음
프라이빗 및 퍼블릭 IPv4 주소	이러한 주소는 동일하게 유지됩니다.	EC2-Classic: 인스턴스가 새 프라이빗 및 퍼블릭 IPv4 주소를 가져옵니다. EC2-VPC: 인스턴스가 관련 프라이빗 IPv4 주소를 유지합니다. 중지/시작 중에 변경되지 않는 탄력적 IP 주소(EIP)가 지정되지 않는 한, 인스턴스가 새 퍼블릭 IPv4 주소를 가져옵니다.	없음
탄력적 IP 주소(IPv4)	탄력적 IP가 인스턴스와 연결된 상태로 유지됩니다.	EC2-Classic: 인스턴스로부터 탄력적 IP 연결이 끊깁니다. EC2-VPC: 탄력적 IP가 인스턴스와 연결된 상태로 유지됩니다.	인스턴스로부터 탄력적 IP 연결이 끊깁니다.
IPv6 주소 (EC2-VPC 전용)	주소가 동일하게 유지됩니다.	인스턴스가 관련 IPv6 주소를 유지합니다.	없음
인스턴스 스토어 볼륨	데이터가 유지됩니다.	데이터가 지워집니다.	데이터가 지워집니다.

특성	재부팅	종지/시작(Amazon EBS 기반 인스턴스에만 해당)	Terminate
루트 디바이스 볼륨	볼륨이 유지됩니다.	볼륨이 유지됩니다.	볼륨이 기본적으로 삭제됩니다.
결제	인스턴스 결제 시간이 변경되지 않습니다.	상태가 <code>stopping</code> 으로 변경되는 즉시 인스턴스에 대한 요금 발생이 중지됩니다. 인스턴스가 <code>stopped</code> 에서 <code>running</code> 으로 전환될 때마다 새 인스턴스 결제 시간이 시작됩니다.	상태가 <code>shutting-down</code> 으로 변경되는 즉시 인스턴스에 대한 요금 발생이 중지됩니다.

항상 운영 체제 종료 명령을 실행하면 인스턴스 스토어 기반 인스턴스가 종료됩니다. 운영 체제 종료 명령으로 Amazon EBS 기반 인스턴스를 중지할지, 종료할지를 제어할 수 있습니다. 자세한 내용은 [인스턴스가 개시하는 종료 동작 변경 \(p. 293\)](#) 섹션을 참조하십시오.

인스턴스 시작

인스턴스는 AWS 클라우드의 가상 서버입니다. 인스턴스는 Amazon Machine Image(AMI)에서 시작됩니다. AMI는 운영 체제와 애플리케이션 서버, 그리고 인스턴스 사용을 위한 애플리케이션을 제공합니다.

AWS 가입 시 무상으로 Amazon EC2를 시작할 수 있는 [AWS 프리 티어](#)를 제공합니다. 프리 티어를 통해 12개월 동안 무료로 마이크로 인스턴스를 시작하고 사용할 수 있습니다. 프리 티어 외의 인스턴스를 시작하는 경우에는 인스턴스에 대하여 표준 Amazon EC2 사용 요금이 청구됩니다. 자세한 내용은 [Amazon EC2 요금](#)을 참조하십시오.

다음 방법을 사용하여 인스턴스를 시작할 수 있습니다.

방법	설명서
[Amazon EC2 콘솔] 선택한 AMI 사용	인스턴스 시작하기 (p. 265)
[Amazon EC2 콘솔] 기존 인스턴스를 템플릿으로 사용	기존의 인스턴스를 템플릿으로 새 인스턴스를 시작하는 방법 (p. 270)
[Amazon EC2 콘솔] 생성한 Amazon EBS 스냅샷 사용	백업에서 Linux 인스턴스를 시작하는 방법 (p. 271)
[Amazon EC2 콘솔] AWS Marketplace에서 구매한 AMI 사용	AWS Marketplace 인스턴스 시작 (p. 271)
[AWS CLI] 선택한 AMI 사용	AWS CLI를 통해 Amazon EC2를 사용하는 방법
[Windows PowerShell용 AWS 도구] 선택한 AMI 사용	Amazon EC2(Windows PowerShell용 AWS 도구 사용)

인스턴스 시작한 다음 인스턴스를 연결하여 사용할 수 있습니다. 인스턴스는 `pending` 상태로 시작됩니다. 인스턴스 부팅이 시작되면 인스턴스의 상태가 `running`로 변경됩니다. 인스턴스 연결이 가능해 질 때까지 약간의 시간이 걸릴 수 있습니다. 인스턴스에서 수신하는 퍼블릭 DNS 이름은 사용자가 인터넷 상에서 해당 인스턴스에 접속할 때 사용됩니다. 인스턴스에서 수신하는 프라이빗 DNS 이름은 동일한 Amazon EC2 네트워크 (EC2-Classic 또는 EC2-VPC) 내 다른 인스턴스에서 해당 인스턴스에 접속할 때 사용됩니다. 인스턴스 연결에 대한 자세한 내용은 [Linux 인스턴스에 연결 \(p. 274\)](#)을 참조하십시오.

인스턴스 작업을 완료한 후에는 반드시 인스턴스를 삭제하십시오. 자세한 내용은 [인스턴스 종료 \(p. 291\)](#) 섹션을 참조하십시오.

인스턴스 시작하기

인스턴스를 시작하기 전에 설정을 확인합니다. 자세한 내용은 [Amazon EC2로 설정 \(p. 16\)](#) 섹션을 참조하십시오.

AWS 계정은 생성 시기와 사용 리전에 따라 EC2-Classic과 EC2-VPC 플랫폼을 모두 지원할 수 있습니다. 계정에서 지원하는 플랫폼을 확인하려면 [지원되는 플랫폼 \(p. 471\)](#)을 참조하십시오. 계정에서 EC2-Classic을 지원하는 경우, 다음 중 한 플랫폼에서 인스턴스를 시작할 수 있습니다. 계정에서 EC2-VPC만 지원하는 경우에는 VPC에서만 인스턴스를 시작할 수 있습니다.

Important

시작하는 인스턴스가 [AWS 프리 티어](#)에 해당되지 않는 경우, 유튜 상태를 포함해 인스턴스가 실행된 시간에 대하여 과금이 청구됩니다.

AMI에서 인스턴스 시작

인스턴스를 시작할 때 구성을 선택해야 하며, 이것을 Amazon 머신 이미지(AMI)이라고 합니다. AMI는 새 인스턴스를 생성하는 데 필요한 정보를 담고 있습니다. 예를 들어, AMI에는 웹 서버 역할을 수행하는 데 필요한 소프트웨어가 포함될 수 있습니다(Linux, Apache, 사용자의 웹 사이트 등).

Tip

인스턴스가 빨리 시작되도록 하려면 큰 요청을 여러 개의 작은 배치로 나눕니다. 예를 들어 인스턴스 500개에 대해 시작 요청을 한 개 생성하는 대신, 인스턴스 100개에 대해 한 개씩 총 5개의 시작 요청을 생성합니다.

인스턴스를 시작하려면 다음을 수행합니다.

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 화면 상단의 탐색 모음에는 현재 리전이 표시됩니다. 인스턴스를 사용할 리전을 선택합니다. 일부 Amazon EC2 리소스는 리전 간에 공유될 수 있지만 그렇지 않은 리소스도 있으므로 잘 선택해야 합니다. 요건에 따라 적합한 리전을 선택하십시오. 자세한 내용은 [리소스 위치 \(p. 673\)](#) 섹션을 참조하십시오.
3. Amazon EC2 콘솔 대시보드에서 [Launch Instance]를 선택합니다.
4. [Choose an Amazon Machine Image (AMI)] 페이지에서 다음에 따라 AMI를 선택합니다.
 - a. 왼쪽 창에서 사용할 AMI의 유형을 선택합니다:

빠른 시작

빠른 시작을 도와주는 인기 AMI를 선별하여 보여줍니다. 프리 티어로 이용할 수 있는 AMI만 선택하려면 왼쪽 창에서 [Free tier only]를 선택합니다. (해당되는 AMI는 Free tier eligible로 표시됩니다.)

My AMIs

사용자가 소유한 프라이빗 AMI 또는 공유된 프라이빗 AMI입니다.

AWS Marketplace

AMI를 비롯하여 AWS에서 실행되는 소프트웨어를 구입할 수 있는 온라인 상점입니다. AWS Marketplace에서 인스턴스를 시작하는 방법에 대한 자세한 내용은 [AWS Marketplace 인스턴스 시작 \(p. 271\)](#)을 참조하십시오.

Community AMIs

AWS 커뮤니티 멤버가 다른 사람의 사용을 허용하여 게시한 AMI입니다. 운영 체제에 따라 AMI 목록을 필터링하려면 [Operating system] 아래의 확인란을 선택하십시오. 이 외에도 아키텍처나 루트 디바이스 타입에 따라 필터링할 수 있습니다.

- b. 각 AMI의 지원 [Root device type] 목록을 확인합니다. ebs(Amazon EBS에서 지원 유형) 또는 instance-store(인스턴스 스토어에서 지원) 중 필요한 유형의 AMI를 확인하십시오. 자세한 내용은 [루트 디바이스 스토리지 \(p. 64\)](#) 섹션을 참조하십시오.
 - c. 각 AMI의 지원 [Virtualization type] 목록을 확인합니다. hvm 또는 paravirtual 중 필요한 유형의 AMI를 확인하십시오. 예를 들어 일부 인스턴스 유형은 HVM이 필요합니다. 자세한 내용은 [Linux AMI 가상화 유형 \(p. 66\)](#) 섹션을 참조하십시오.
 - d. 용도에 적합한 AMI를 선택하고 [Select] 버튼을 선택합니다.
5. Choose an Instance Type(인스턴스 유형 선택) 페이지에서 시작할 인스턴스의 하드웨어 구성 및 크기를 선택합니다. 대형 인스턴스는 CPU와 메모리가 더 높습니다. 자세한 내용은 [인스턴스 유형 \(p. 146\)](#) 섹션을 참조하십시오.

[t2.micro] 인스턴스 유형을 선택하면 프리 티어 사용 자격을 유지할 수 있습니다. 자세한 내용은 [T2 인스턴스 \(p. 149\)](#) 섹션을 참조하십시오.

기본 설정에서 마법사는 현 세대의 인스턴스 유형을 표시하고 사용자가 선택한 AMI를 기반으로 하여 첫 번째로 사용 가능한 유형을 선택합니다. 필터 목록에서 [All generations]를 선택하면 이전 세대의 인스턴스 유형을 볼 수 있습니다.

Note

AWS를 처음 사용하는 사용자가 시험 목적으로 빠른 인스턴스 설정을 원하는 경우, 이 단계에서 [Review and Launch]를 선택하면 기본 구성 설정을 적용하여 인스턴스를 시작할 수 있습니다. 그렇지 않은 경우 [Next: Configure Instance Details]를 선택해 인스턴스를 세부 구성할 수 있습니다.

6. [Configure Instance Details] 페이지에서 필요에 맞게 다음 설정을 변경하고(모든 설정 항목을 확장 표시 하려면 Advanced Details 클릭), Next: Add Storage를 선택합니다.
- Number of instances: 시작할 인스턴스의 수를 입력합니다.

Note

애플리케이션을 처리할 인스턴스의 수를 올바르게 유지하는 데 도움을 주기 위해 [Launch into Auto Scaling Group]을 선택해 시작 구성 및 Auto Scaling 그룹을 생성할 수 있습니다. Auto Scaling은 사양에 따라 그룹에서 인스턴스의 수를 조정합니다. 자세한 내용은 [Auto Scaling 사용 설명서](#) 섹션을 참조하십시오.

- [Purchasing option]: 스팟 인스턴스를 시작하려면 [Request Spot instances]를 선택합니다. 자세한 내용은 [스팟 인스턴스 \(p. 203\)](#) 섹션을 참조하십시오.
- 계정에 따라 EC2-Classic과 EC2-VPC 플랫폼을 모두 지원하거나 EC2-VPC만 지원할 수 있습니다. 계정에서 지원하는 플랫폼을 확인하려면 [지원되는 플랫폼 \(p. 471\)](#)을 참조하십시오. EC2-VPC만 지원하는 계정에서는 기본이거나 기본이 아닌 VPC로 인스턴스를 시작할 수 있습니다. 그렇지 않은 경우, EC2-Classic 또는 기본이 아닌 VPC에서 인스턴스를 시작할 수 있습니다.

Note

일부 인스턴스 유형은 VPC로 시작해야 합니다. VPC가 없는 경우에는 마법사를 사용해 계정을 만들 수 있습니다.

EC2-Classic에서 시작:

- Network: [Launch into EC2-Classic]을 선택합니다.
- Availability Zone: 사용할 가용 영역을 선택합니다. [No preference]를 선택하면 AWS에서 임의로 가용 영역을 선택합니다.

VPC에서 시작:

- Network: VPC를 선택하거나 [Create new VPC]를 선택하여 Amazon VPC 콘솔로 이동해 새 VPC를 생성합니다. 마침 후에 마법사로 돌아와 [Refresh]를 선택하면 해당 VPC가 목록에 로딩됩니다.
- Subnet: 인스턴스를 시작할 서브넷을 선택합니다. EC2-VPC만 지원하는 계정의 경우, [No preference]를 선택하면 AWS에서 임의의 가용 영역 내 기본 서브넷을 선택합니다. 새 서브넷을 생

성하려면 Create new subnet을 선택하여 Amazon VPC 콘솔로 이동합니다. 마친 후에 마법사로 돌아와 [Refresh]를 선택하면 해당 서브넷이 목록에 로딩됩니다.

- Auto-assign Public IP: 인스턴스의 퍼블릭 IPv4 주소 수신 여부를 지정합니다. 기본 설정 사용 시, 기본 서브넷을 사용하는 인스턴스는 퍼블릭 IPv4 주소를 수신하고 기본이 아닌 서브넷의 인스턴스는 수신하지 않습니다. [Enable] 또는 [Disable]를 선택하여 서브넷의 기본 설정을 재정의할 수 있습니다. 자세한 내용은 [퍼블릭 IPv4 주소 및 외부 DNS 호스트 이름 \(p. 491\)](#) 섹션을 참조하십시오.
- Auto-assign IPv6 IP: 인스턴스가 서브넷 범위 내에서 IPv6 주소를 수신할지 지정합니다. [Enable] 또는 [Disable]를 선택하여 서브넷의 기본 설정을 재정의합니다. 이 옵션은 IPv6 CIDR 블록에 VPC 와 서브넷을 연결한 경우에만 사용할 수 있습니다. 자세한 내용은 [Amazon VPC 사용 설명서](#)의 VPC 및 서브넷 섹션을 참조하십시오.
- [IAM role]: 인스턴스와 연결할 AWS Identity and Access Management(IAM) 역할을 선택합니다. 자세한 내용은 [Amazon EC2의 IAM 역할 \(p. 456\)](#) 섹션을 참조하십시오.
- Shutdown behavior: 인스턴스 셧다운 시 적용할 인스턴스 상태(중지 또는 종료)를 선택합니다. 자세한 내용은 [인스턴스가 개시하는 종료 동작 변경 \(p. 293\)](#) 섹션을 참조하십시오.
- Enable termination protection: 선택 시 실수로 인스턴스를 종료하는 일을 방지할 수 있습니다. 자세한 내용은 [인스턴스에 대한 종료 방지 기능 활성화 \(p. 292\)](#) 섹션을 참조하십시오.
- Monitoring: 확인란을 선택하면 Amazon CloudWatch 사용한 인스턴스 세부 모니터링 기능이 활성화 됩니다. 이 때 추가 요금이 발생합니다. 자세한 내용은 [CloudWatch를 사용해 인스턴스 모니터링하기 \(p. 347\)](#) 섹션을 참조하십시오.
- EBS-Optimized instance: Amazon EBS 최적화 인스턴스는 최적화된 구성 스택을 사용하며, Amazon EBS I/O를 위한 추가 전용 용량을 제공합니다. 인스턴스 유형이 이 기능을 지원하는 경우, 확인란을 체크하면 기능이 활성화됩니다. 이 때 추가 요금이 발생합니다. 자세한 내용은 [Amazon EBS 최적화 인스턴스 \(p. 614\)](#) 섹션을 참조하십시오.
- [Tenancy]: VPC로 인스턴스를 시작하는 경우 격리된 전용 하드웨어([Dedicated tenancy]) 또는 전용 호스트([Dedicated host])에서 인스턴스를 실행하도록 선택할 수 있습니다. 추가 요금이 적용될 수 있습니다. 자세한 내용은 [전용 인스턴스 \(p. 257\)](#) 및 [전용 호스트 \(p. 246\)](#) 섹션을 참조하십시오.
- [Network interfaces]: 특정 서브넷을 선택한 경우, 인스턴스에 대해 네트워크 인터페이스를 최대 2개 까지 지정할 수 있습니다.
 - [Network Interface]의 경우, AWS에서 새로운 인터페이스가 자동으로 생성되도록 [New network interface]를 선택하거나 사용 가능한 기존 네트워크 인터페이스를 선택합니다.
 - [Primary IP]의 경우, 서브넷 범위에서 프라이빗 IPv4 주소를 입력하거나 AWS에서 프라이빗 IPv4 주소가 자동으로 선택되도록 [Auto-assign]을 그대로 둡니다.
 - [Secondary IP addresses]에서 [Add IP]를 선택하면 선택한 네트워크 인터페이스에 프라이빗 IPv4 주소를 두 개 이상 할당 할 수 있습니다.
 - (IPv6 전용) [IPv6 IPs]에서 [Add IP]를 선택하고 서브넷 범위에서 IPv6 주소를 입력하거나 AWS가 자동으로 선택하도록 [Auto-assign]을 그대로 둡니다.
 - [Add Device]를 선택하여 보조 네트워크 인터페이스를 추가합니다. 보조 네트워크 인터페이스는 인스턴스와 동일한 가용 영역에 있는 경우 VPC의 다른 서브넷에 상주할 수 있습니다.자세한 내용은 [탄력적 네트워크 인터페이스 \(p. 512\)](#) 섹션을 참조하십시오. 네트워크 인터페이스를 두 개 이상 지정하면 인스턴스가 퍼블릭 IPv4 주소를 수신할 수 없습니다. 또한 eth0에 대해 기존 네트워크 인터페이스를 지정하면 [Auto-assign Public IP]를 사용하여 서브넷의 퍼블릭 IPv4 설정을 재정의 할 수 없습니다. 자세한 내용은 [인스턴스 시작 시 퍼블릭 IPv4 주소 배정 \(p. 495\)](#) 섹션을 참조하십시오.
- Kernel ID: (반가상화(PV) AMI만 해당) 특정 커널을 사용하려는 경우가 아니라면 [Use default]를 선택 합니다.
- RAM disk ID: (반가상화(PV) AMI만 해당) 특정 RAM 디스크를 사용하려는 경우가 아니라면 [Use default]를 선택합니다. 커널을 선택해 사용할 때는 해당 커널을 지원하는 드라이버가 설치된 RAM 디스크 지정이 필요할 수 있습니다.
- Placement group: 배치 그룹은 클러스터 인스턴스에 적용되는 논리적 그룹입니다. 기존의 배치 그룹을 선택하거나 새로 만들 수 있습니다. 이 옵션은 배치 그룹을 지원하는 인스턴스 유형을 선택한 경우에만 사용할 수 있습니다. 자세한 내용은 [배치 그룹 \(p. 527\)](#) 섹션을 참조하십시오.

- User data: 시작 과정에서 인스턴스를 구성하거나 구성 스크립트를 실행할 때 사용할 사용자 데이터를 지정할 수 있습니다. 파일을 첨부하려면 As file 옵션을 선택하여 첨부할 파일을 선택하십시오.
7. Add Storage(스토리지 추가) 페이지에서는 AMI를 통해 지정된 볼륨 이외에 인스턴스에 연결할 볼륨들을 지정할 수 있습니다(예: 루트 디바이스 볼륨). 경우에 따라 다음 옵션을 변경하고 설정을 마치면 Next: Add Tags를 선택합니다.
- Type: 인스턴스에 연결할 인스턴스 스토어 또는 Amazon EBS 볼륨을 선택합니다. 목록에 표시되는 볼륨 유형은 선택한 인스턴스 유형에 따라 달라집니다. 자세한 내용은 [Amazon EC2 인스턴스 스토어 \(p. 642\)](#) 및 [Amazon EBS 볼륨 \(p. 562\)](#) 섹션을 참조하십시오.
 - Device: 볼륨에서 사용할 디바이스 이름을 목록에서 선택합니다.
 - Snapshot: 볼륨 복원에 사용할 스냅샷의 이름이나 ID를 입력합니다. 또는 Snapshot 필드에 텍스트를 입력하여 퍼블릭 스냅샷을 검색할 수 있습니다. 스냅샷 정보는 대/소문자를 구분합니다.
 - Size: Amazon EBS를 지원하는 볼륨의 스토리지 크기를 지정할 수 있습니다. 선택한 AMI와 인스턴스가 프리 티어에 해당되는 경우에도 총 스토리지 크기를 30GiB 미만으로 유지해야 프리 티어 한도를 유지할 수 있습니다.

Note

Linux AMI에서 부팅 볼륨 2TiB(2,048GiB) 이상을 사용하려면 GPT 파티션 테이블과 GRUB 2가 필요합니다. 현재 여러 Linux AMI에서 부팅 볼륨을 최대 2,047GiB까지만 지원하는 MBR 파티셔닝 체계를 사용하고 있습니다. 인스턴스가 2TiB 이상의 부팅 볼륨에서 부팅되지 않는 경우 사용 중인 AMI의 부팅 볼륨 크기가 2,047GiB로 제한된 상태일 수 있습니다. 부팅 볼륨이 아닌 볼륨에는 이 Linux 인스턴스에 대한 제한이 적용되지 않습니다.

Note

이 때 루트 볼륨을 비롯해 스냅샷에서 생성된 볼륨 크기를 높이면, 해당 볼륨에 대한 파일 시스템을 확장해야 추가된 공간을 사용할 수 있습니다. 인스턴스 시작 후 파일 시스템 확장에 대한 자세한 내용은 [Linux에서 EBS 볼륨의 크기, IOPS 또는 유형 수정 \(p. 590\)](#)을 참조하십시오.

- Volume Type: Amazon EBS 볼륨은 범용 SSD, 프로비저닝된 IOPS SSD 또는 Magnetic 볼륨 중 선택합니다. 자세한 내용은 [Amazon EBS 볼륨 유형 \(p. 564\)](#) 섹션을 참조하십시오.

Note

Magnetic 부팅 볼륨을 선택한 경우, 마법사를 마칠 때 범용 SSD 볼륨을 해당 인스턴스와 콘솔 시작 시 기본 부팅 볼륨으로 설정하라는 메시지가 나타납니다. (이 설정은 브라우저 세션에서 계속 유지되며, 프로비저닝된 IOPS SSD 부팅 볼륨을 사용하는 AMI에는 적용되지 않습니다.) 범용 SSD 볼륨은 부팅 속도가 훨씬 더 빠르고 대부분의 작업에서 최적화된 볼륨이기 때문에 이 볼륨을 기본으로 설정하는 것을 권장합니다. 자세한 내용은 [Amazon EBS 볼륨 유형 \(p. 564\)](#) 섹션을 참조하십시오.

Note

2012년 이전에 만들어진 일부 AWS 계정은 프로비저닝된 IOPS SSD(`io1`) 볼륨을 지원하지 않는 `us-west-1` 또는 `ap-northeast-1`의 가용 영역에 대한 액세스 권한이 있을 수도 있습니다. 이런 리전 중 하나에 `io1` 볼륨을 만들거나 블록 디바이스 매핑에서 `io1` 볼륨이 있는 인스턴스를 시작할 수 없는 경우, 해당 리전에서 다른 가용 영역을 사용해 보십시오. 가용 영역에 4GiB의 `io1` 볼륨을 만들어 그 영역에서 `io1` 볼륨을 지원하는지 확인할 수 있습니다.

- IOPS: 프로비저닝된 IOPS SSD 볼륨 유형을 선택한 경우, 볼륨에서 지원되는 초당 I/O (IOPS) 수를 입력할 수 있습니다.
- Delete on Termination: Amazon EBS 볼륨에 적용되는 기능으로, 확인란을 선택하면 인스턴스 종료 시 볼륨을 삭제합니다. 자세한 내용은 [인스턴스 종료 시 Amazon EBS 볼륨 보존 \(p. 294\)](#) 섹션을 참조하십시오.
- Encrypted: 확인란을 선택하면 신규 Amazon EBS 볼륨을 암호화합니다. 암호화된 스냅샷에서 복구된 Amazon EBS 볼륨은 자동으로 암호화됩니다. 암호화된 볼륨은 [지원되는 인스턴스 유형 \(p. 619\)](#)에만 연결될 수도 있습니다.

8. [Add Tags] 페이지에서 키와 값의 조합을 제공하여 [태그 \(p. 681\)](#)를 지정합니다. 인스턴스 또는 볼륨 또는 이 둘 모두에 태그를 지정할 수 있습니다. 리소스에 2개 이상의 태그를 추가하려면 [Add another tag]를 선택합니다. 모두 마쳤으면 [Next: Configure Security Group]을 선택합니다.
9. [Configure Security Group] 페이지에서 기존 보안 그룹을 사용하여 인스턴스의 방화벽 규칙을 정의할 수 있습니다. 이 규칙은 인스턴스에 전달되는 수신 네트워크 트래픽을 정의합니다. 다른 모든 트래픽은 무시됩니다. (보안 그룹에 대한 자세한 내용은 [Linux 인스턴스에 대한 Amazon EC2 보안 그룹 \(p. 385\)](#)을 참조하십시오.) 다음 과정에 따라 그룹을 선택하거나 새로 생성하고 [Review and Launch]를 선택합니다.

기존 보안 그룹에서 선택하는 경우:

1. [Select an existing security group]을 선택합니다. 사용자의 보안 그룹이 표시됩니다. (이 때 EC2-Classic에서 시작하는 경우에는 EC2-Classic 보안 그룹이 표시되고, VPC에서 시작하는 경우에는 해당 VPC의 보안 그룹이 표시됩니다.)
2. 목록에서 보안 그룹을 선택합니다.
3. (선택 사항) 기존의 보안 그룹 규칙은 수정할 수 없으며, 대신 [Copy to new]를 선택하여 새 보안 그룹으로 규칙을 복사할 수 있습니다. 다음 절차로 진행하여 설명에 따라 규칙을 추가할 수 있습니다.

새 보안 그룹을 생성하는 경우:

1. Create a new security group을 선택합니다. 마법사에서 launch-wizard-x 보안 그룹을 자동으로 정의합니다.
2. (선택 사항) 생성된 보안 그룹의 이름과 설명을 수정할 수 있습니다.
3. 마법사에서 SSH(22번 포트, Linux) 또는 RDP(3389번 포트, Windows)를 사용한 인스턴스 연결을 허용하는 인바운드 규칙을 자동으로 정의합니다.

Warning

이 규칙은 특정 포트를 사용한 모든 IP 주소(0.0.0.0/0)에서의 인스턴스 액세스를 허용합니다. 예제에서 잠시 사용하는 것은 괜찮지만 프로덕션 환경에서 사용하기에는 안전하지 않습니다. 이 때는 특정 주소나 IP 주소 범위에서만 인스턴스 액세스를 허용하도록 설정해야 합니다.

4. 규칙은 필요에 따라 추가할 수 있습니다. 예를 들어 웹 서버인 인스턴스는 80번 포트(HTTP)와 443 번 포트(HTTPS)를 개방해 인터넷 트래픽을 허용할 수 있습니다.

규칙을 추가하려면 [Add Rule]을 선택한 다음 네트워크 트래픽의 개방 프로토콜을 선택하고 소스를 지정합니다. [Source] 목록에서 [My IP]를 선택하면 마법사에서 사용자 컴퓨터의 퍼블릭 IP 주소가 자동으로 추가됩니다. 하지만 고정 IP 주소 없이 방화벽 뒤에서 또는 ISP를 통해 연결하는 경우에는 클라이언트 컴퓨터가 사용하는 IP 주소의 범위를 찾아야 합니다.

10. [Review Instance Launch] 페이지에서 인스턴스 세부 정보를 확인한 다음, 해당되는 [Edit] 링크를 선택하여 필요한 사항을 변경합니다.

준비가 완료되면 [Launch]를 선택합니다.

11. [Select an existing key pair or create a new key pair] 대화 상자에서 기존 키 쌍을 선택하거나 새 키 쌍을 만들 수 있습니다. 예를 들어, [Choose an existing key pair]를 선택하고 초기 설정에서 생성한 키 페어를 선택합니다.

인스턴스를 시작하려면 승인 확인란을 선택한 후 [Launch Instances]를 선택합니다.

Important

[Proceed without key pair] 옵션을 선택할 경우 사용자가 다른 방법으로 로그인할 수 있도록 구성된 AMI를 선택해야만 인스턴스에 연결할 수 있습니다.

12. (선택 사항) 인스턴스의 상태 확인 정보를 생성할 수 있습니다(추가 비용 적용 가능). (지금 결정하지 않아도 언제든지 나중에 추가할 수 있습니다.) 확인 화면에서 [Create status check alarms]를 선택하여 지원에 따릅니다. 자세한 내용은 [상태 확인 경보 생성 및 수정 \(p. 342\)](#) 섹션을 참조하십시오.

13. 인스턴스 상태가 `running`이 아닌 `terminated`로 변경되는 경우, 정보를 통해 인스턴스가 시작되지 않은 이유를 알 수 있습니다. 자세한 내용은 [인스턴스가 즉시 종료되는 경우 해결 방법 \(p. 698\)](#) 섹션을 참조하십시오.

기존의 인스턴스를 템플릿으로 새 인스턴스를 시작하는 방법

Amazon EC2 콘솔에서는 Launch More Like This 마법사 옵션을 제공하여 현재 인스턴스를 템플릿으로 사용하여 다른 인스턴스를 시작할 수 있습니다. 이 옵션을 사용하면 Amazon EC2 시작 마법사에서 선택한 인스턴스의 세부적인 구성 정보가 자동으로 입력됩니다.

Note

Launch More Like This 마법사 옵션은 선택한 인스턴스를 복제하는 것이 아니라 일부 구성 정보만 복제합니다. 인스턴스의 사본을 만드려면 해당 인스턴스에서 AMI를 생성한 후 AMI에서 추가 인스턴스를 시작하십시오.

선택한 인스턴스에서 시작 마법사로 복제되는 구성 정보:

- AMI ID
- 인스턴스 유형
- 선택 인스턴스가 위치한 가용 영역 또는 VPC, 서브넷
- 퍼블릭 IPv4 주소. 선택한 인스턴스에 현재 할당된 퍼블릭 IPv4 주소가 있다면 이 인스턴스의 기본 퍼블릭 IPv4 주소 설정에 상관 없이 새 인스턴스에서도 퍼블릭 IPv4 주소를 수신합니다. 퍼블릭 IPv4 주소에 대한 자세한 내용은 [퍼블릭 IPv4 주소 및 외부 DNS 호스트 이름 \(p. 491\)](#) 섹션을 참조하십시오.
- 배치 그룹(해당되는 경우)
- 인스턴스에 연결된 IAM 규칙(해당되는 경우)
- 종료 동작 설정(중지 또는 종료)
- 종료 보호 설정(True 또는 False)
- CloudWatch 모니터링(활성화 또는 비활성화)
- Amazon EBS 최적화 설정(True/False 설정)
- 테넌시 설정(VPC에서 시작하는 경우, 공유 또는 전용)
- 커널 ID 및 RAM 디스크 ID(해당되는 경우)
- 사용자 데이터(지정된 경우)
- 인스턴스에 연결된 태그(해당되는 경우)
- 인스턴스에 연결된 보안 그룹

선택한 인스턴스에서 마법사로 복제되지 않고 마법사에서 자체 기본 설정을 적용하는 구성 정보:

- (VPC에만 해당) 네트워크 인터페이스 수(기본값은 기본 네트워크 인터페이스(eth0)인 네트워크 인터페이스 1개)
- Storage: AMI와 인스턴스 유형에 따라 기본 스토리지 구성이 결정됩니다.

현재 인스턴스를 템플릿으로 사용하는 방법

1. [Instances] 페이지에서 사용할 인스턴스를 선택합니다.
2. [Actions]를 선택한 다음 [Launch More Like This]를 선택합니다.
3. [Review Instance Launch] 페이지에서 시작 마법사가 열립니다. 인스턴스 세부 정보를 확인한 다음, 해당되는 [Edit] 링크를 클릭해 필요한 사항을 변경할 수 있습니다.

준비되면 [Launch]를 선택한 다음 키 페어를 선택하고 인스턴스를 시작합니다.

백업에서 Linux 인스턴스를 시작하는 방법

Amazon EBS을 지원하는 Linux 인스턴스는 스냅샷을 생성하여 인스턴스의 루트 디바이스 볼륨을 백업할 수 있습니다. 인스턴스의 루트 디바이스 볼륨 스냅샷이 있으면 해당 인스턴스를 종료하고 나중에 스냅샷에서 새 인스턴스를 시작할 수 있습니다. 인스턴스를 시작할 때 사용된 원래 AMI 없이 동일한 이미지를 사용하여 새로 인스턴스를 시작해야 할 때 유용한 방법입니다.

Important

스냅샷에서 Windows AMI를 생성할 수 있지만 그러면 AMI에서 인스턴스를 제대로 시작할 수 없습니다.

Red Hat Enterprise Linux(RHEL) 및 SUSE Linux Enterprise Server(SLES)와 같은 일부 Linux 배포판은 AMI 와 연관된 결제 제품 코드를 사용하여 패키지 업데이트의 구독 상태를 확인합니다. EBS 스냅샷으로 AMI를 생성하면 이 결제 코드가 유지되지 않으며, AMI 등에서 시작된 이후 인스턴스는 패키지 업데이트 인프라에 연결할 수 없습니다. 결제 제품 코드를 유지하려면 스냅샷이 아닌 인스턴스에서 AMI를 생성하십시오. 자세한 내용은 [Amazon EBS 지원 Linux AMI 생성 \(p. 81\)](#) 또는 [인스턴스 스토어 기반 Linux AMI 생성 \(p. 84\)](#)을 (를) 참조하십시오.

다음 절차에 따라 콘솔을 사용해 인스턴스의 루트 볼륨에서 AMI를 생성합니다. 원하는 경우, [register-image](#)(AWS CLI) 또는 [Register-EC2Image](#)(Windows PowerShell용 AWS 도구) 명령을 대신 사용할 수 있습니다. 스냅샷은 블록 디바이스 매핑을 사용해 지정합니다.

콘솔을 이용하여 루트 볼륨에서 AMI를 생성하는 방법

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Elastic Block Store]과 [Snapshots]를 선택합니다.
3. Create Snapshot을 클릭합니다.
4. [Volumes] 필드에서 루트 볼륨의 이름 또는 ID를 입력한 다음 옵션 목록에서 선택합니다.
5. 방금 만든 스냅샷을 선택한 다음 [Actions]와 [Create Image]를 선택합니다.
6. [Create Image from EBS Snapshot] 대화 상자에서 다음 정보를 입력한 후 [Create]를 선택합니다. 상위 인스턴스를 다시 생성하는 경우 상위 인스턴스와 동일한 옵션을 선택합니다.
 - [Architecture]: 32비트의 경우 [i386]을 선택하고 64비트의 경우 [x86_64]를 선택합니다.
 - [Root device name]: 루트 볼륨에 적절한 이름을 입력합니다. 자세한 내용은 [Linux 인스턴스의 디바이스 명명 \(p. 660\)](#)섹션을 참조하십시오.
 - [Virtualization type]: 이 AMI에서 실행된 인스턴스가 반가상화(PV)를 사용하는지 또는 하드웨어 가상 머신(HVM) 가상화를 사용하는지 선택합니다. 자세한 내용은 [Linux AMI 가상화 유형 \(p. 66\)](#)섹션을 참조하십시오.
 - (PV 가상화 유형에만 해당) [Kernel ID] 및 [RAM disk ID]: 목록에서 AKI 및 ARI를 선택합니다. 기본 AKI를 선택하거나 AKI를 선택하지 않으면 이 AMI를 사용하여 인스턴스를 시작할 때마다 AKI를 지정하라는 메시지가 나타납니다. 또한 기본 AKI가 인스턴스와 호환되지 않는 경우, 상태 확인 작업 시 인스턴스 오류가 발생할 수 있습니다.
 - (선택 사항) [Block Device Mappings]: 볼륨을 추가하거나 AMI에 대한 루트 볼륨의 기본 크기를 확장합니다. 더 큰 볼륨을 사용할 수 있도록 인스턴스의 파일 시스템 크기 조정에 대한 자세한 내용은 [볼륨 크기 조정 후 Linux 파일 시스템 확장 \(p. 594\)](#) 섹션을 참조하십시오.
7. 탐색 창에서 [AMIs]를 선택합니다.
8. 방금 생성한 AMI를 선택하고 [Launch]를 선택합니다. 마법사가 안내하는 대로 인스턴스를 시작합니다. 마법사 단계별 구성에 대한 자세한 정보는 [인스턴스 시작하기 \(p. 265\)](#)을 참조하십시오.

AWS Marketplace 인스턴스 시작

AWS Marketplace 제품을 구독하고 Amazon EC2 시작 마법사를 사용하여 제품의 AMI에서 인스턴스를 시작할 수 있습니다. 유료 AMI에 대한 자세한 내용은 [유료 AMI \(p. 78\)](#) 섹션을 참조하십시오. 시작한 이후에

구독을 취소하려면 먼저 해당 구독에서 실행 중인 모든 인스턴스를 종료해야 합니다. 자세한 내용은 [AWS Marketplace 구독 관리 \(p. 80\)](#) 섹션을 참조하십시오.

시작 마법사를 사용하여 AWS Marketplace에서 인스턴스를 시작하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. Amazon EC2 대시보드에서 [Launch Instance]를 선택합니다.
3. [Choose an Amazon Machine Image (AMI)] 페이지에서 왼쪽에 있는 [AWS Marketplace] 범주를 선택합니다. 범주를 검색하거나 검색 기능을 사용하여 적합한 AMI를 찾습니다. [Select]를 선택하여 제품을 선택합니다.
4. 대화 상자에 선택한 제품에 대한 개요가 표시됩니다. 요금 정보와 공급업체에서 제공한 기타 정보를 조회할 수 있습니다. 준비가 되면 [Continue]를 선택합니다.

Note

AMI를 사용하여 인스턴스를 시작하기 전에는 제품 사용 요금이 부과되지 않습니다. 마법사의 다음 페이지에서 인스턴스 유형을 선택하라는 메시지가 표시되므로 지원되는 각 인스턴스 유형의 요금을 기록해 둡니다. 제품에 추가 세금이 적용될 수도 있습니다.

5. [Choose an Instance Type] 페이지에서 시작할 인스턴스의 하드웨어 구성 및 크기를 선택합니다. 완료되면 [Next: Configure Instance Details]를 선택합니다.
6. 마법사의 다음 페이지에서 인스턴스를 구성하고, 스토리지 및 태그를 추가할 수 있습니다. 구성 가능한 다른 옵션에 대한 자세한 내용은 [인스턴스 시작하기 \(p. 265\)](#) 섹션을 참조하십시오. [Configure Security Group] 페이지가 나타날 때까지 [Next]를 선택합니다.

이 마법사에서는 제품에 대한 공급업체의 사양에 따라 새 보안 그룹을 생성합니다. 보안 그룹은 Linux의 SSH(포트 22) 또는 Windows의 RDP(포트 3389)에 모든 IPv4 주소(0.0.0.0/0) 액세스를 허용하는 규칙을 포함할 수 있습니다. 특정 주소 또는 주소 범위에만 해당 포트를 통한 인스턴스 액세스를 허용하도록 규칙을 조정하는 것이 좋습니다.

준비가 되면 [Review and Launch]를 선택합니다.

7. [Review Instance Launch] 페이지에서 인스턴스를 시작할 AMI에 대한 세부 정보와 마법사에서 설정한 기타 구성 정보를 확인합니다. 준비되면 [Launch]를 선택하여 키 페어를 선택하거나 생성하고 인스턴스를 시작합니다.
8. 구독한 제품에 따라 인스턴스를 시작하는 데 몇 분 또는 그 이상 걸릴 수 있습니다. 인스턴스를 시작하면 먼저 제품을 구독해야 합니다. 신용 카드 정보에 문제가 있는 경우 계정 세부 정보를 업데이트하라는 메시지가 나타납니다. 시작 확인 페이지가 표시되면 [View Instances]를 선택하여 인스턴스 페이지로 이동합니다.

Note

유형 상태를 포함해 인스턴스가 실행 중인 동안 구독 요금이 청구됩니다. 인스턴스를 중지하더라도 스토리지에 대해 요금이 부과될 수 있습니다.

9. 인스턴스가 [running] 상태일 때 인스턴스에 연결할 수 있습니다. 이렇게 하려면 목록에서 인스턴스를 선택하고 [Connect]를 선택합니다. 대화 상자의 지침을 따릅니다. 인스턴스 연결에 대한 자세한 내용은 [Linux 인스턴스에 연결 \(p. 274\)](#) 섹션을 참조하십시오.

Important

인스턴스에 로그인하는 데 특정 사용자 이름을 사용해야 할 수도 있으므로 공급업체의 사용 지침을 주의해서 확인하십시오. 구독 세부 정보 액세스에 대한 자세한 내용은 [AWS Marketplace 구독 관리 \(p. 80\)](#) 섹션을 참조하십시오.

API 및 CLI를 사용하여 AWS Marketplace AMI 인스턴스를 시작하는 방법

API 또는 명령줄 도구를 사용하여 AWS Marketplace 제품에서 인스턴스를 시작하려면 먼저 제품을 구독해야 합니다. 다음 방법을 사용하여 제품의 AMI ID로 인스턴스를 시작할 수 있습니다.

방법	설명서
AWS CLI	run-instances 명령을 사용합니다. 자세한 내용은 Launching an Instance 섹션을 참조하십시오.
Windows PowerShell용 AWS 도구	New-EC2Instance 명령을 사용합니다. 자세한 내용은 Launch an Amazon EC2 Instance Using Windows PowerShell 을 참조하십시오.
Query API	RunInstances 요청을 사용합니다.

Linux 인스턴스에 연결

시작한 Linux 인스턴스에 연결하여 로컬 컴퓨터와 인스턴스 간에 파일을 전송하는 방법을 알아봅니다.

Windows 인스턴스에 연결해야 하는 경우 Windows 인스턴스용 Amazon EC2 사용 설명서의 [Connecting to Your Windows Instance](#) 섹션을 참조하십시오.

사용자 컴퓨터	주제
리눅스	SSH를 사용하여 Linux 인스턴스에 연결 (p. 274)
Windows가 설치된	PuTTY를 사용하여 Windows에서 Linux 인스턴스에 연결 (p. 278)
모두	MindTerm을 사용하여 Linux 인스턴스에 연결 (p. 284)

인스턴스에 연결한 후에는 [자습서: Amazon LinuxLAMP 웹 서버 설치 \(p. 27\)](#) 또는 [자습서: Amazon Linux를 통한 WordPress 블로그 호스팅 \(p. 37\)](#) 등의 자습서 중 하나를 참조하여 실행해 볼 수 있습니다.

SSH를 사용하여 Linux 인스턴스에 연결

인스턴스를 시작한 후 인스턴스에 연결하고 바로 앞에 있는 컴퓨터를 사용하는 것처럼 인스턴스를 사용할 수 있습니다.

Note

인스턴스를 시작한 후, 연결할 수 있도록 인스턴스가 준비될 때까지 몇 분 정도 걸릴 수 있습니다. 인스턴스가 상태 확인을 통과했는지 확인하십시오. [Instances] 페이지의 [Status Checks] 열에서 이 정보를 볼 수 있습니다.

다음 지침에서는 SSH 클라이언트를 사용하여 인스턴스에 연결하는 방법을 설명합니다. 인스턴스에 연결을 시도하는 동안 오류가 발생한 경우 [인스턴스 연결 문제 해결](#)을 참조하십시오.

사전 조건

Linux 인스턴스에 연결하려면 먼저 다음 사전 요구 사항을 완료하십시오.

- **SSH 클라이언트 설치**

Linux 컴퓨터에는 기본적으로 SSH 클라이언트가 포함되어 있을 가능성이 높습니다. 명령줄에 ssh를 입력하여 SSH 클라이언트가 있는지 확인할 수 있습니다. 컴퓨터에서 이 명령이 인식되지 않으면 OpenSSH 프로젝트는 전체 SSH 도구의 무료 구현을 제공합니다. 자세한 내용은 <http://www.openssh.com>을 참조하십시오.

- **AWS CLI 도구 설치**

(선택 사항) 타사의 퍼블릭 AMI를 사용하고 있는 경우 명령줄 도구를 사용하여 지문을 확인할 수 있습니다. AWS CLI 설치에 대한 자세한 내용은 [AWS Command Line Interface 사용 설명서](#)의 설정을 참조하십시오.

- **인스턴스의 ID 보기**

Amazon EC2 콘솔을 사용하여 인스턴스의 ID를 볼 수 있습니다([Instance ID] 열에서). [describe-instances](#)(AWS CLI) 또는 [Get-EC2Instance](#)(Windows PowerShell용 AWS 도구) 명령을 사용할 수도 있습니다.

- **인스턴스의 퍼블릭 DNS 이름 보기**

&EC2; 콘솔을 사용해서 사용자의 인스턴스에 대한 퍼블릭 DNS를 얻을 수 있습니다([Public DNS (IPv4)] 열 확인. 이 열이 숨겨진 경우는 [Show/Hide] 아이콘을 클릭하고 [Public DNS (IPv4)]를 선택). [describe-instances](#)(AWS CLI) 또는 [Get-EC2Instance](#)(Windows PowerShell용 AWS 도구) 명령을 사용할 수도 있습니다.

- **(IPv6 전용) 인스턴스의 IPv6 주소를 얻습니다.**

인스턴스에 IPv6 주소를 할당했다면 퍼블릭 IPv4 주소나 퍼블릭 IPv4 DNS 호스트 이름 대신 IPv6 주소를 사용하여 인스턴스에 연결할 수도 있습니다. 로컬 컴퓨터에 IPv6 주소가 있고 IPv6를 사용하도록 컴퓨터를 구성해야 합니다. Amazon EC2 콘솔을 사용하여 인스턴스의 IPv6 주소를 얻을 수 있습니다([IPv6 IPs] 필드 확인). [describe-instances\(AWS CLI\)](#) 또는 [Get-EC2Instance\(Windows PowerShell용 AWS 도구\)](#) 명령을 사용할 수도 있습니다. IPv6에 대한 자세한 내용은 [IPv6 주소 \(p. 492\)](#) 단원을 참조하십시오.

- 프라이빗 키 찾기

인스턴스를 시작할 때 지정한 키 페어에 대한 .pem 파일의 정규화된 경로가 필요합니다.

- IP 주소에서 인스턴스로의 인바운드 SSH 트래픽 활성화

인스턴스와 연관된 보안 그룹이 IP 주소로부터 들어오는 SSH 트래픽을 허용하는지 확인하십시오. 자세한 내용은 [인스턴스에 네트워크 액세스 권한 부여](#)를 참조하십시오.

Important

기본 보안 그룹은 기본적으로 들어오는 SSH 트래픽을 허용하지 않습니다.

Linux 인스턴스에 연결

SSH 클라이언트를 사용하여 Linux 인스턴스에 연결하려면 다음 프로시저를 사용하십시오. 인스턴스에 연결을 시도하는 동안 오류가 발생한 경우 [인스턴스 연결 문제 해결](#)을 참조하십시오.

SSH를 사용하여 인스턴스에 연결하려면

1. (선택 사항) 로컬 시스템(인스턴스가 아님)에서 다음 명령 중 하나를 사용하여 실행 중인 인스턴스에서 RSA 키 지문을 확인할 수 있습니다. 이 기능은 타사의 퍼블릭 AMI에서 인스턴스를 시작한 경우에 유용합니다. `SSH HOST KEY FINGERPRINTS` 섹션을 찾아서 RSA 지문(예: 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f)을 적고 이것을 인스턴스의 지문과 비교합니다.

- [get-console-output\(AWS CLI\)](#)

```
aws ec2 get-console-output --instance-id instance_id
```

Note

인스턴스가 pending 상태가 아닌 running 상태인지 확인합니다. `SSH HOST KEY FINGERPRINTS` 섹션은 인스턴스를 처음 부팅한 후에만 사용할 수 있습니다.

2. 명령줄 셸에서 디렉터리를 인스턴스를 시작할 때 만든 프라이빗 키 파일의 위치로 변경합니다.
3. chmod 명령을 사용하여 프라이빗 키 파일을 공개적으로 볼 수 없는지 확인합니다. 예를 들어, 프라이빗 키 파일의 이름이 `my-key-pair.pem`인 경우 다음 명령을 사용합니다.

```
chmod 400 /path/my-key-pair.pem
```

4. ssh 명령을 사용하여 인스턴스에 연결합니다. 프라이빗 키(.pem) 파일과 `user_name@public_dns_name`을 지정합니다. Amazon Linux의 경우 사용자 이름은 `ec2-user`입니다. RHEL의 경우 사용자 이름은 `ec2-user` 또는 `root`입니다. Ubuntu의 경우 사용자 이름은 `ubuntu` 또는 `root`입니다. CentOS의 경우 사용자 이름은 `centos`입니다. Fedora의 경우 사용자 이름은 `ec2-user`입니다. SUSE의 경우 사용자 이름은 `ec2-user` 또는 `root`입니다. `ec2-user` 및 `root`를 사용할 수 없는 경우 AMI 공급자에게 문의하십시오.

```
ssh -i /path/my-key-pair.pem ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com
```

다음과 같은 응답이 표시됩니다.

```
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com (10.254.142.33)'  
can't be established.  
RSA key fingerprint is 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f.  
Are you sure you want to continue connecting (yes/no)?
```

5. (IPv6 전용) 또는 IPv6 주소를 이용해 인스턴스에 연결할 수도 있습니다. 프라이빗 키(.pem) 파일 경로 및 적절한 사용자 이름을 사용하여 ssh 명령을 지정합니다. Amazon Linux의 경우 사용자 이름은 ec2-user입니다. RHEL의 경우 사용자 이름은 ec2-user 또는 root입니다. Ubuntu의 경우 사용자 이름은 ubuntu 또는 root입니다. CentOS의 경우 사용자 이름은 centos입니다. Fedora의 경우 사용자 이름은 ec2-user입니다. SUSE의 경우 사용자 이름은 ec2-user 또는 root입니다. ec2-user 및 root를 사용할 수 없는 경우 AMI 공급자에게 문의하십시오.

```
ssh -i /path/my-key-pair.pem ec2-user@2001:db8:1234:1a00:9691:9503:25ad:1761
```

6. (선택 사항) 보안 알림의 지문이 1단계에서 얻은 지문과 일치하는지 확인합니다. 이들 지문이 일치하지 않으면 누군가가 "메시지 가로채기(man-in-the-middle)" 공격을 시도하고 있는 것일 수 있습니다. 이를 지문이 일치하면 다음 단계를 계속 진행합니다.
7. yes를 입력합니다.

다음과 같은 응답이 표시됩니다.

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (RSA)  
to the list of known hosts.
```

SCP를 사용하여 Linux에서 Linux 인스턴스로 파일 전송

로컬 컴퓨터와 Linux 인스턴스 간에 파일을 전송하는 한 가지 방법은 SCP(Secure Copy)를 사용하는 것입니다. 이 섹션에서는 SCP를 사용하여 파일을 전송하는 방법을 설명합니다. 이 절차는 SSH를 사용하여 인스턴스에 연결하는 절차와 매우 비슷합니다.

사전 조건

- SCP 클라이언트 설치

대부분의 Linux, Unix 및 Apple 컴퓨터에는 기본적으로 SCP 클라이언트가 포함되어 있습니다. 그렇지 않은 경우, OpenSSH 프로젝트는 SCP 클라이언트를 포함하는 전체 SSH 도구의 무료 구현을 제공합니다. 자세한 내용은 <http://www.openssh.org> 섹션을 참조하십시오.

- 인스턴스의 ID 보기

Amazon EC2 콘솔을 사용하여 인스턴스의 ID를 볼 수 있습니다([Instance ID] 열에서). [describe-instances](#)(AWS CLI) 또는 [Get-EC2Instance](#)(Windows PowerShell용 AWS 도구) 명령을 사용할 수도 있습니다.

- 인스턴스의 퍼블릭 DNS 이름 보기

&EC2; 콘솔을 사용해서 사용자의 인스턴스에 대한 퍼블릭 DNS를 얻을 수 있습니다([Public DNS (IPv4)] 열 확인. 이 열이 숨겨진 경우는 [Show/Hide] 아이콘을 클릭하고 [Public DNS (IPv4)]를 선택). [describe-instances](#)(AWS CLI) 또는 [Get-EC2Instance](#)(Windows PowerShell용 AWS 도구) 명령을 사용할 수도 있습니다.

- (IPv6 전용) 인스턴스의 IPv6 주소를 얻습니다.

인스턴스에 IPv6 주소를 할당했다면 퍼블릭 IPv4 주소나 퍼블릭 IPv4 DNS 호스트 이름 대신 IPv6 주소를 사용하여 인스턴스에 연결할 수도 있습니다. 로컬 컴퓨터에 IPv6 주소가 있고 IPv6를 사용하도록 컴퓨터를 구성해야 합니다. Amazon EC2 콘솔을 사용하여 인스턴스의 IPv6 주소를 얻을 수 있습니다([IPv6 IPs] 필드 확인). [describe-instances](#)(AWS CLI) 또는 [Get-EC2Instance](#)(Windows PowerShell용 AWS 도구) 명령을 사용할 수도 있습니다. IPv6에 대한 자세한 내용은 [IPv6 주소 \(p. 492\)](#) 단원을 참조하십시오.

- 프라이빗 키 찾기

인스턴스를 시작할 때 지정한 키 페어에 대한 .pem 파일의 정규화된 경로가 필요합니다.

- IP 주소에서 인스턴스로의 인바운드 SSH 트래픽 활성화

인스턴스와 연관된 보안 그룹이 IP 주소로부터 들어오는 SSH 트래픽을 허용하는지 확인하십시오. 자세한 내용은 [인스턴스에 네트워크 액세스 권한 부여](#)를 참조하십시오.

Important

기본 보안 그룹은 기본적으로 들어오는 SSH 트래픽을 허용하지 않습니다.

다음 절차에서는 SCP를 사용하여 파일을 전송하는 과정을 단계별로 안내합니다. 이미 SSH를 사용하여 인스턴스에 연결했으며 지문을 확인한 경우 SCP 명령(4단계)을 포함하는 단계부터 시작할 수 있습니다.

SCP를 사용하여 파일을 전송하려면

1. (선택 사항) 로컬 시스템에서 다음 명령 중 하나를 사용하여 인스턴스에서 RSA 키 지문을 확인할 수 있습니다. 이 기능은 타사의 퍼블릭 AMI에서 인스턴스를 시작한 경우에 유용합니다. **SSH HOST KEY FINGERPRINTS** 섹션을 찾아서 RSA 지문(예: 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f)을 적고 이것을 인스턴스의 지문과 비교합니다.

- [get-console-output\(AWS CLI\)](#)

```
aws ec2 get-console-output --instance-id instance_id
```

Note

SSH HOST KEY FINGERPRINTS 섹션은 인스턴스를 처음 부팅한 후에만 사용할 수 있습니다.

2. 명령 셸에서 디렉터리를 인스턴스를 시작할 때 지정한 프라이빗 키 파일의 위치로 변경합니다.
3. chmod 명령을 사용하여 프라이빗 키 파일을 공개적으로 볼 수 없는지 확인합니다. 예를 들어, 프라이빗 키 파일의 이름이 my-key-pair.pem인 경우 다음 명령을 사용합니다.

```
chmod 400 /path/my-key-pair.pem
```

4. 인스턴스의 퍼블릭 DNS 이름을 사용하여 인스턴스로 파일을 전송합니다. 예를 들어, 프라이빗 키 파일의 이름이 my-key-pair이고 전송할 파일이 SampleFile.txt이며 인스턴스의 퍼블릭 DNS 이름이 ec2-198-51-100-1.compute-1.amazonaws.com인 경우, 다음 명령을 사용하여 파일을 ec2-user 훈 디렉터리로 복사합니다.

```
scp -i /path/my-key-pair.pem /path/SampleFile.txt ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com:~
```

Tip

Amazon Linux의 경우 사용자 이름은 ec2-user입니다. RHEL의 경우 사용자 이름은 ec2-user 또는 root입니다. Ubuntu의 경우 사용자 이름은 ubuntu 또는 root입니다. Centos의 경우 사용자 이름은 centos입니다. Fedora의 경우 사용자 이름은 ec2-user입니다. SUSE의 경우 사용자 이름은 ec2-user 또는 root입니다. ec2-user 및 root를 사용할 수 없는 경우 AMI 공급자에게 문의하십시오.

다음과 같은 응답이 표시됩니다.

```
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com (10.254.142.33)' can't be established.
```

```
RSA key fingerprint is 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f.  
Are you sure you want to continue connecting (yes/no)?
```

- (IPv6 전용) 또는 인스턴스의 IPv6 주소를 이용해 파일을 전송할 수도 있습니다. IPv6 주소는 이스케이프된(\\) 대괄호([])로 둑어야 합니다.

```
scp -i /path/my-key-pair.pem /path/SampleFile.txt ec2-user@  
\[2001:db8:1234:1a00:9691:9503:25ad:1761\]:~
```

- (선택 사항) 보안 알림의 지문이 1단계에서 얻은 지문과 일치하는지 확인합니다. 이들 지문이 일치하지 않으면 누군가가 "메시지 가로채기(man-in-the-middle)" 공격을 시도하고 있는 것일 수 있습니다. 이들 지문이 일치하면 다음 단계를 계속 진행합니다.
- yes**를 입력합니다.

다음과 같은 응답이 표시됩니다.

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (RSA)  
to the list of known hosts.  
Sending file modes: C0644 20 SampleFile.txt  
Sink: C0644 20 SampleFile.txt  
SampleFile.txt 100% 20 0.0KB/s 00:00
```

Note

"bash: scp: command not found" 오류가 표시되는 경우 먼저 Linux 인스턴스에 scp를 설치해야 합니다. 일부 운영 체제의 경우, 이 명령어는 openssh-clients 패키지에 있습니다. Amazon ECS 최적화 AMI 같은 Amazon Linux 변형의 경우에는 다음 명령을 사용하여 scp를 설치하십시오.

```
[ec2-user ~]$ sudo yum install -y openssh-clients
```

- 반대 방향으로(Amazon EC2 인스턴스에서 로컬 컴퓨터로) 파일을 전송하려면 단순히 호스트 파라미터의 순서를 역순으로 지정하면 됩니다. 예를 들어, EC2 인스턴스의 SampleFile.txt 파일을 로컬 컴퓨터의 험 디렉터리에 SampleFile2.txt로 다시 전송하려면 로컬 컴퓨터에서 다음 명령을 사용합니다.

```
scp -i /path/my-key-pair.pem ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com:~/  
SampleFile.txt ~/SampleFile2.txt
```

- (IPv6 전용) 또는 인스턴스의 IPv6 주소를 이용해 반대 방향으로 파일을 전송할 수도 있습니다.

```
scp -i /path/my-key-pair.pem ec2-user@\[2001:db8:1234:1a00:9691:9503:25ad:1761\]:~/  
SampleFile.txt ~/SampleFile2.txt
```

PuTTY를 사용하여 Windows에서 Linux 인스턴스에 연결

인스턴스를 시작한 후 인스턴스에 연결하고 바로 앞에 있는 컴퓨터를 사용하는 것처럼 인스턴스를 사용할 수 있습니다.

Note

인스턴스를 시작한 후, 연결할 수 있도록 인스턴스가 준비될 때까지 몇 분 정도 걸릴 수 있습니다. 인스턴스가 상태 확인을 통과했는지 확인하십시오. [Instances] 페이지의 [Status Checks] 열에서 이 정보를 볼 수 있습니다.

다음 지침에서는 Windows용 무료 SSH 클라이언트인 PuTTY를 사용하여 인스턴스에 연결하는 방법을 설명합니다. 인스턴스에 연결을 시도하는 동안 오류가 발생한 경우 [인스턴스 연결 문제 해결](#)을 참조하십시오.

사전 조건

PuTTY(를) 사용하여 Linux 인스턴스에 연결하려면 먼저 다음 사전 요구 사항을 완료하십시오.

- PuTTY 설치

[PuTTY 다운로드 페이지](#)에서 PuTTY를 다운로드하여 설치합니다. 이미 이전 버전의 PuTTY가 설치되어 있다면 최신 버전을 다운로드하는 것이 좋습니다. 전체 제품군을 설치해야 합니다.

- 인스턴스의 ID 보기

Amazon EC2 콘솔을 사용하여 인스턴스의 ID를 볼 수 있습니다([Instance ID] 열에서). [describe-instances](#)(AWS CLI) 또는 [Get-EC2Instance](#)(Windows PowerShell용 AWS 도구) 명령을 사용할 수도 있습니다.

- 인스턴스의 퍼블릭 DNS 이름 보기

&EC2; 콘솔을 사용해서 사용자의 인스턴스에 대한 퍼블릭 DNS를 얻을 수 있습니다([Public DNS (IPv4)] 열 확인. 이 열이 숨겨진 경우는 [Show/Hide] 아이콘을 클릭하고 [Public DNS (IPv4)]를 선택). [describe-instances](#)(AWS CLI) 또는 [Get-EC2Instance](#)(Windows PowerShell용 AWS 도구) 명령을 사용할 수도 있습니다.

- (IPv6 전용) 인스턴스의 IPv6 주소를 얻습니다.

인스턴스에 IPv6 주소를 할당했다면 퍼블릭 IPv4 주소나 퍼블릭 IPv4 DNS 호스트 이름 대신 IPv6 주소를 사용하여 인스턴스에 연결할 수도 있습니다. 로컬 컴퓨터에 IPv6 주소가 있고 IPv6를 사용하도록 컴퓨터를 구성해야 합니다. Amazon EC2 콘솔을 사용하여 인스턴스의 IPv6 주소를 얻을 수 있습니다([IPv6 IPs] 필드 확인). [describe-instances](#)(AWS CLI) 또는 [Get-EC2Instance](#)(Windows PowerShell용 AWS 도구) 명령을 사용할 수도 있습니다. IPv6에 대한 자세한 내용은 [IPv6 주소 \(p. 492\)](#) 단원을 참조하십시오.

- 프라이빗 키 찾기

인스턴스를 시작할 때 지정한 키 페어에 대한 .pem 파일의 정규화된 경로가 필요합니다.

- IP 주소에서 인스턴스로의 인바운드 SSH 트래픽 활성화

인스턴스와 연관된 보안 그룹이 IP 주소로부터 들어오는 SSH 트래픽을 허용하는지 확인하십시오. 자세한 내용은 [인스턴스에 네트워크 액세스 권한 부여](#)를 참조하십시오.

Important

기본 보안 그룹은 기본적으로 들어오는 SSH 트래픽을 허용하지 않습니다.

PuTTYgen을 사용하여 프라이빗 키 변환

PuTTY에서는 Amazon EC2에서 생성된 프라이빗 키 형식(.pem)을 기본적으로 지원하지 않습니다. PuTTY에는 PuTTYgen이라는 도구가 있는데, 이 도구는 키를 필요한 PuTTY 형식(.ppk)으로 변환할 수 있습니다. PuTTY를 사용하여 인스턴스에 연결하기 전에 프라이빗 키를 이 형식(.ppk)으로 변환해야 합니다.

개인 키를 변환하려면

1. PuTTYgen을 시작합니다(예: [Start] 메뉴에서 [All Programs > PuTTY > PuTTYgen] 선택).
2. Type of key to generate에서 RSA를 선택합니다.

Note

이전 버전의 PuTTYgen을 사용하는 경우 [SSH-2 RSA]를 선택합니다.

3. Load를 선택합니다. 기본적으로 PuTTYgen에는 확장명이 .ppk인 파일만 표시됩니다. .pem 파일을 찾으려면 모든 유형의 파일을 표시하는 옵션을 선택합니다.

4. 인스턴스를 시작할 때 지정한 키 페어에 대한 .pem 파일을 선택한 다음 [Open]을 선택합니다. [OK]를 선택하여 확인 대화 상자를 닫습니다.
5. [Save private key]를 선택하여 PuTTY에서 사용할 수 있는 형식으로 키를 저장합니다. PuTTYgen에서 암호 없이 키 저장에 대한 경고가 표시됩니다. [Yes]를 선택합니다.

Note

프라이빗 키의 암호는 추가 보호 계층이므로 프라이빗 키가 공개되었더라도 이 암호가 없으면 사용할 수 없습니다. 암호 사용 시 단점은 인스턴스에 로그온하거나 인스턴스로 파일을 복사할 때 사용자의 개입이 필요하므로 자동화하기 어렵다는 것입니다.

6. 키 페어에 사용된 키에 대해 동일한 이름을 지정합니다(예: my-key-pair). PuTTY에서 자동으로 .ppk 파일 확장명을 추가합니다.

이제 개인 키가 PuTTY에 사용하기에 올바른 형식으로 되어 있으므로 PuTTY의 SSH 클라이언트를 사용하여 인스턴스에 연결할 수 있습니다.

PuTTY 세션 시작

PuTTY을(를) 사용하여 Linux 인스턴스에 연결하려면 다음 프로시저를 사용하십시오. 프라이빗 키에 대해 생성한 .ppk 파일이 필요합니다. 인스턴스에 연결을 시도하는 동안 오류가 발생한 경우 [인스턴스 연결 문제 해결](#)을 참조하십시오.

PuTTY 세션을 시작하려면

1. (선택 사항) 로컬 시스템에서 다음 명령 중 하나를 사용하여 인스턴스에서 RSA 키 지문을 확인할 수 있습니다. 이 기능은 타사의 퍼블릭 AMI에서 인스턴스를 시작한 경우에 유용합니다. **SSH HOST KEY FINGERPRINTS** 섹션을 찾아서 RSA 지문(예: 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f)을 적고 이것을 인스턴스의 지문과 비교합니다.

- [get-console-output\(AWS CLI\)](#)

```
aws ec2 get-console-output --instance-id instance_id
```

다음은 살펴봐야 할 예제를 나타낸 코드 조각입니다.

```
\r\nec2: -----BEGIN SSH HOST KEY FINGERPRINTS-----\r\n...\r\n\r\nec2: 2048 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f\r\n root@ip-192-0-2-0 (RSA)\r\n...\r\n\r\nec2: -----END SSH HOST KEY FINGERPRINTS-----
```

Note

SSH HOST KEY FINGERPRINTS 섹션은 인스턴스를 처음 부팅한 후에만 사용할 수 있습니다.

2. PuTTY를 시작합니다. 즉, [시작] 메뉴에서 [모든 프로그램] > [PuTTY] > [PuTTY]를 선택합니다.
3. [Category] 창에서 [Session]를 선택하고 다음 필드를 작성합니다.
 - a. [Host Name] 상자에 [*user_name@public_dns_name*](#)을 입력합니다. AMI에 적합한 사용자 이름을 지정해야 합니다. 예:
 - Amazon Linux AMI의 경우 사용자 이름은 `ec2-user`입니다.
 - RHEL AMI의 경우 사용자 이름은 `ec2-user` 또는 `root`입니다.
 - Ubuntu AMI의 경우 사용자 이름은 `ubuntu` 또는 `root`입니다.
 - Centos AMI의 경우 사용자 이름은 `centos`입니다.

- Fedora AMI의 경우 사용자 이름은 `ec2-user`입니다.
 - SUSE의 경우 사용자 이름은 `ec2-user` 또는 `root`입니다.
 - `ec2-user` 및 `root`를 사용할 수 없는 경우 AMI 공급자에게 문의하십시오.
- b. (IPv6 전용) 인스턴스의 IPv6 주소를 이용해 연결하려면 `user_name@ipv6_address`를 입력합니다. AMI에 적합한 사용자 이름을 지정해야 합니다. 예:
- Amazon Linux AMI의 경우 사용자 이름은 `ec2-user`입니다.
 - RHEL AMI의 경우 사용자 이름은 `ec2-user` 또는 `root`입니다.
 - Ubuntu AMI의 경우 사용자 이름은 `ubuntu` 또는 `root`입니다.
 - Centos AMI의 경우 사용자 이름은 `centos`입니다.
 - Fedora AMI의 경우 사용자 이름은 `ec2-user`입니다.
 - SUSE의 경우 사용자 이름은 `ec2-user` 또는 `root`입니다.
 - `ec2-user` 및 `root`를 사용할 수 없는 경우 AMI 공급자에게 문의하십시오.
- c. [Connection type] 아래에서 [SSH]를 선택합니다.
- d. [Port]가 22인지 확인합니다.
4. [Category] 창에서 [Connection], [SSH]를 차례로 확장하고 [Auth]를 선택합니다. 다음 작업을 완료합니다.
- a. [Browse]를 선택합니다.
 - b. 키 페어에 대해 생성한 `.ppk` 파일을 선택한 다음 [Open]을 선택합니다.
 - c. (선택 사항) 이 세션을 나중에 다시 시작하려는 경우 세션 정보를 나중에 사용할 수 있게 저장할 수 있습니다. [Category] 트리에서 [Session]을 선택하고 [Saved Sessions]에 세션 이름을 입력한 다음 [Save]를 선택합니다.
 - d. [Open]을 클릭하여 PuTTY 세션을 선택합니다.
5. 이 인스턴스에 처음 연결한 경우 PuTTY에서 연결하려는 호스트를 신뢰할 수 있는지 묻는 보안 알림 대화 상자가 표시됩니다.
6. (선택 사항) 보안 알림 대화 상자의 지문이 1단계에서 얻은 이전 지문과 일치하는지 확인합니다. 이들 지문이 일치하지 않으면 누군가가 "메시지 가로채기(man-in-the-middle)" 공격을 시도하고 있는 것일 수 있습니다. 이들 지문이 일치하면 다음 단계를 계속 진행합니다.
7. [Yes]를 선택합니다. 창이 열리고 인스턴스에 연결됩니다.

Note

개인 키를 PuTTY 형식으로 변환할 때 암호문을 지정한 경우 인스턴스에 로그인할 때 암호문을 제공해야 합니다.

인스턴스에 연결을 시도하는 동안 오류가 발생한 경우 [인스턴스 연결 문제 해결](#)을 참조하십시오.

PuTTY 보안 사본 클라이언트를 사용하여 Linux 인스턴스로 파일 전송

PuTTY SCP(Secure Copy) 클라이언트는 Windows 컴퓨터와 Linux 인스턴스 간에 파일을 전송하는 데 사용할 수 있는 명령줄 도구입니다. GUI(그래픽 사용자 인터페이스)를 선호하는 경우 WinSCP라는 오픈 소스 GUI 도구를 사용할 수 있습니다. 자세한 내용은 [WinSCP를 사용하여 Linux 인스턴스로 파일 전송 \(p. 282\)](#) 섹션을 참조하십시오.

PSCP를 사용하려면 [PuTTYgen을 사용하여 프라이빗 키 변환 \(p. 279\)](#)에서 생성한 프라이빗 키가 필요합니다. 또한 Linux 인스턴스의 퍼블릭 DNS 주소도 필요합니다.

다음 예에서는 `sample_file.txt` 파일을 Windows 컴퓨터의 C:\ 드라이브에서 Linux 인스턴스의 `/usr/local` 디렉터리로 전송합니다.

```
C:\> pscp -i C:\Keys\my-key-pair.ppk C:\Sample_file.txt user_name@public_dns:/usr/local/  
Sample_file.txt
```

(IPv6 전용) 다음 예에서는 인스턴스의 IPv6 주소를 이용해 `sample_file.txt` 파일을 전송합니다. IPv6 주소는 대괄호([])로 둘어야 합니다.

```
C:\> pscp -i C:\Keys\my-key-pair.ppk C:\Sample_file.txt user_name@[ipv6-address]:/usr/  
local/Sample_file.txt
```

WinSCP를 사용하여 Linux 인스턴스로 파일 전송

WinSCP는 SFTP, SCP, FTP 및 FTPS 프로토콜을 사용하여 원격 컴퓨터로 파일을 업로드하고 전송할 수 있는 Windows용 GUI 기반 파일 관리자입니다. WinSCP를 사용하면 Windows 시스템에서 Linux 인스턴스로 파일을 끌어 놓거나 두 시스템 간에 전체 디렉터리 구조를 동기화할 수 있습니다.

WinSCP를 사용하려면 [PuTTYgen](#)을 사용하여 [프라이빗 키 변환](#) (p. 279)에서 생성한 프라이빗 키가 필요합니다. 또한 Linux 인스턴스의 퍼블릭 DNS 주소도 필요합니다.

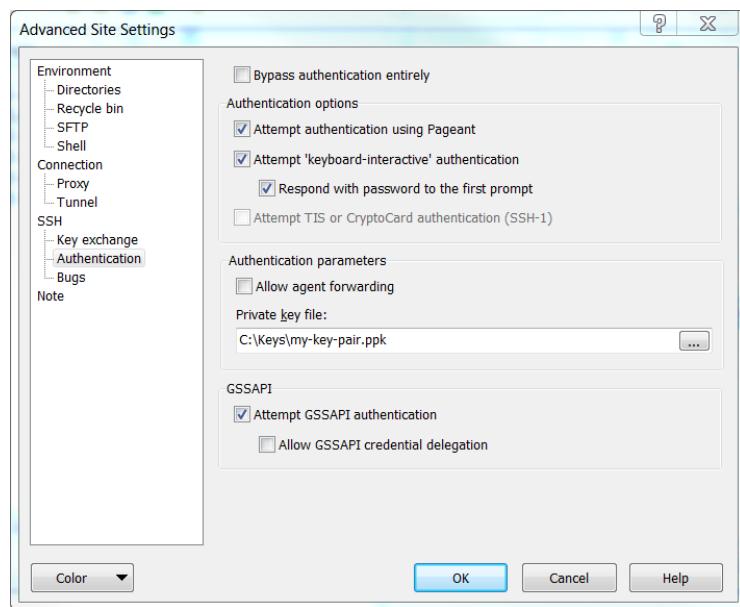
1. <http://winscp.net/eng/download.php>에서 WinSCP를 다운로드하여 설치합니다. 대부분 사용자의 경우 기본 설치 옵션을 그대로 사용해도 좋습니다.
2. WinSCP를 시작합니다.
3. [WinSCP Login] 화면에서 [Host name]에 인스턴스의 퍼블릭 DNS 호스트 이름 또는 퍼블릭 IPv4 주소를 입력합니다.

Note

(IPv6 전용) 인스턴스의 IPv6 주소를 이용해 로그인하려면 인스턴스의 IPv6 주소를 입력합니다.

4. [User name]에는 AMI의 기본 사용자 이름을 입력합니다. Amazon Linux AMI의 경우 사용자 이름은 `ec2-user`입니다. Red Hat AMI의 경우 사용자 이름은 `root`이며, Ubuntu AMI의 경우 사용자 이름은 `ubuntu`입니다.
5. 인스턴스의 프라이빗 키를 지정합니다. [Private key]에서는 프라이빗 키의 경로를 입력하거나 "..." 버튼을 선택하여 파일을 찾아봅니다. 최신 WinSCP 버전의 경우 [Advanced]를 선택하여 어드밴스 사이트 설정을 열고 [SSH]에서 [Authentication]을 선택하여 [Private key file] 설정을 찾습니다.

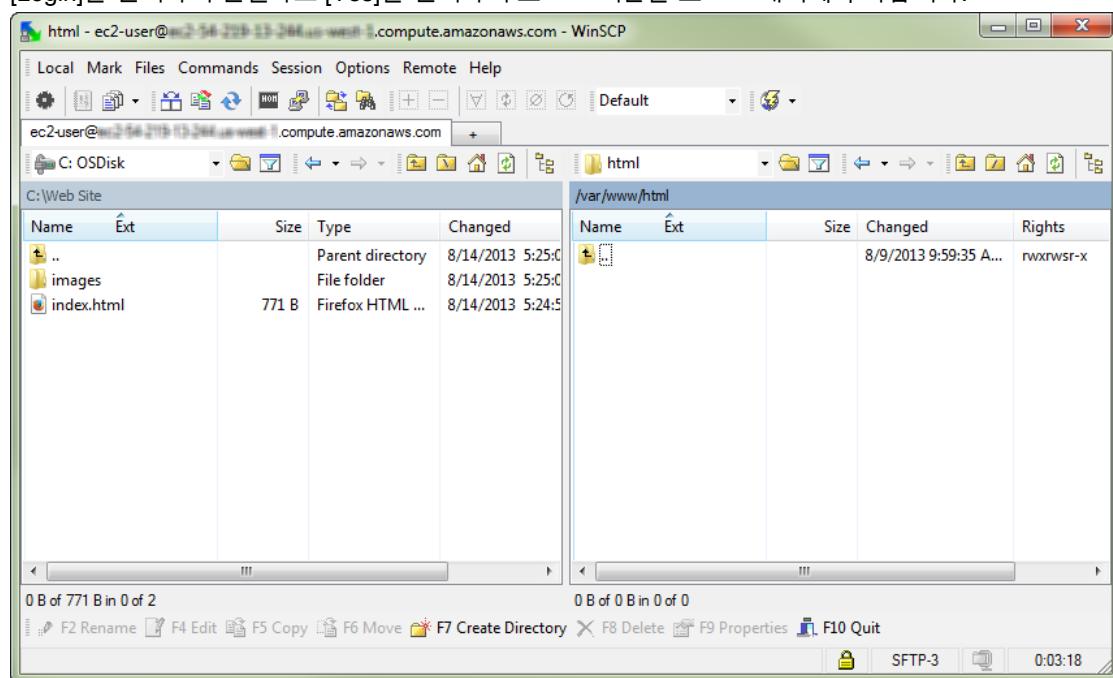
다음은 WinSCP 버전 5.9.4의 스크린샷입니다.



Note

WinSCP에는 PuTTY 프라이빗 키 파일(.ppk)이 필요합니다. PuTTYgen을 사용하여 .pem 보안 키 파일을 .ppk 형식으로 변환할 수 있습니다. 자세한 내용은 [PuTTYgen을 사용하여 프라이빗 키 변환 \(p. 279\)](#) 섹션을 참조하십시오.

6. (선택 사항) 왼쪽 패널에서 [Directories]를 선택하고 파일을 추가할 디렉터리의 경로를 [Remote directory]에 입력합니다. 최신 WinSCP 버전의 경우 [Advanced]를 선택하여 어드밴스 사이트 설정을 연 다음 [Environment]에서 [Directories]를 선택하여 [Remote directory] 설정을 찾습니다.
7. [Login]을 선택하여 연결하고 [Yes]를 선택하여 호스트 지문을 호스트 캐시에 추가합니다.



8. 연결이 설정된 후 연결 창에서 Linux 인스턴스는 오른쪽에 있고 로컬 시스템은 왼쪽에 있습니다. 로컬 시스템에서 원격 파일 시스템으로 파일을 직접 끌어 놓을 수 있습니다. WinSCP에 대한 자세한 내용은 <http://winscp.net/eng/docs/start>의 프로젝트 설명서를 참조하십시오.

Note

"Cannot execute SCP to start transfer" 오류가 표시되는 경우 먼저 Linux 인스턴스에 scp를 설치해야 합니다. 일부 운영 체제의 경우, 이 명령어는 `openssh-clients` 패키지에 있습니다. Amazon ECS 최적화 AMI 같은 Amazon Linux 변형의 경우에는 다음 명령을 사용하여 scp를 설치하십시오.

```
[ec2-user ~]$ sudo yum install -y openssh-clients
```

MindTerm을 사용하여 Linux 인스턴스에 연결

인스턴스를 시작한 후 인스턴스에 연결하고 바로 앞에 있는 컴퓨터를 사용하는 것처럼 인스턴스를 사용할 수 있습니다.

Note

인스턴스를 시작한 후, 연결할 수 있도록 인스턴스가 준비될 때까지 몇 분 정도 걸릴 수 있습니다. 인스턴스가 상태 확인을 통과했는지 확인하십시오. [Instances] 페이지의 [Status Checks] 열에서 이 정보를 볼 수 있습니다.

다음 지침에서는 Amazon EC2 콘솔을 통해 MindTerm을 사용하여 인스턴스에 연결하는 방법을 설명합니다. 인스턴스에 연결을 시도하는 동안 오류가 발생한 경우 [인스턴스 연결 문제 해결](#)을 참조하십시오.

Important

Chrome 브라우저는 NPAPI 플러그인을 지원하지 않으므로 MindTerm 클라이언트를 실행할 수 없습니다. 자세한 정보는 Chromium의 [NPAPI 운영 중단 관련 문서](#)를 참조하십시오. 대신에 Firefox, Safari 또는 Internet Explorer 9 이상 버전을 사용할 수 있습니다.

사전 조건

- Java 설치

Linux 컴퓨터에는 Java가 포함되어 있을 가능성이 높습니다. 그렇지 않은 경우 [웹 브라우저에서 Java를 사용으로 설정하는 방법은 무엇입니까?](#)를 참조하십시오. Windows 또는 Mac 클라이언트에서 관리자 자격 증명을 사용하여 브라우저를 실행해야 합니다. Linux의 경우 `root`로 로그인하지 않으면 추가 단계가 필요 할 수 있습니다.

- 브라우저에서 Java 사용

지침은 https://java.com/en/download/help/enable_browser.xml을 참조하십시오.

- 프라이빗 키 찾기

인스턴스를 시작할 때 지정한 키 페어에 대한 `.pem` 파일의 정규화된 경로가 필요합니다.

- IP 주소에서 인스턴스로의 인바운드 SSH 트래픽 활성화

인스턴스와 연관된 보안 그룹이 IP 주소로부터 들어오는 SSH 트래픽을 허용하는지 확인하십시오. 자세한 내용은 [인스턴스에 네트워크 액세스 권한 부여](#)를 참조하십시오.

Important

기본 보안 그룹은 기본적으로 들어오는 SSH 트래픽을 허용하지 않습니다.

MindTerm 시작

MindTerm과 함께 웹 브라우저를 사용하여 인스턴스에 연결하려면

- Amazon EC2 콘솔의 탐색 창에서 [Instances]를 선택합니다.

2. 인스턴스를 선택한 다음 [Connect]를 선택합니다.
3. [A Java SSH client directly from my browser (Java required)]를 선택합니다.
4. Amazon EC2가 인스턴스의 퍼블릭 DNS 이름을 자동으로 검색하여 그 이름으로 [Public DNS]를 채웁니다. 또한 인스턴스를 시작할 때 지정한 키 페어의 이름도 검색합니다. 다음 절차를 완료하고 [Launch SSH Client]를 선택합니다.
 - a. User name에 인스턴스에 로그인할 사용자 이름을 입력합니다.

Tip

Amazon Linux의 경우 사용자 이름은 `ec2-user`입니다. RHEL의 경우 사용자 이름은 `ec2-user` 또는 `root`입니다. Ubuntu의 경우 사용자 이름은 `ubuntu` 또는 `root`입니다. CentOS의 경우 사용자 이름은 `centos`입니다. Fedora의 경우 사용자 이름은 `ec2-user`입니다. SUSE의 경우 사용자 이름은 `ec2-user` 또는 `root`입니다. `ec2-user` 및 `root`을 사용할 수 없는 경우 AMI 공급자에게 문의하십시오.

- b. [Private key path]에 키 페어 이름을 포함하는 프라이빗 키(`.pem`) 파일의 정규화된 경로를 입력합니다. 예를 들면 다음과 같습니다.
`C:\KeyPairs\my-key-pair.pem`
 - c. (선택 사항) 브라우저 캐시에 프라이빗 키의 위치를 저장하려면 [Store in browser cache]를 선택합니다. 이렇게 하면 Amazon EC2에서는 사용자가 브라우저 캐시를 지울 때까지 이후 브라우저 세션에서 프라이빗 키 위치를 검색할 수 있습니다.
5. 필요할 경우 [Yes]를 선택하여 인증서를 신뢰할 수 있음을 확인하고 [Run]을 선택하여 MindTerm 클라이언트를 실행합니다.
 6. MindTerm을 처음 실행하는 경우, 라이선스 계약에 대한 동의 여부, 험 디렉터리 설정에 대한 확인 여부 및 알려진 호스트 디렉터리 설정에 대한 확인 여부를 묻는 일련의 대화 상자가 표시됩니다. 해당 설정을 확인합니다.
 7. 알려진 호스트 세트에 호스트를 추가할지 묻는 대화 상자가 표시됩니다. 로컬 컴퓨터에 호스트 키 정보를 저장하지 않으려면 [No]를 선택합니다.
 8. 창이 열리고 인스턴스에 연결됩니다.

Note

이전 단계에서 [No]를 선택한 경우 다음과 같은 메시지가 나타납니다.

Verification of server key disabled in this session.

인스턴스 중지 및 시작

인스턴스에서 Amazon EBS 볼륨을 루트 디바이스로 사용하는 경우 해당 인스턴스를 중지했다가 다시 시작할 수 있습니다. 인스턴스 ID는 유지되지만 개요 섹션의 설명처럼 인스턴스는 변경될 수 있습니다.

인스턴스를 중지하면 인스턴스가 종료됩니다. 중지된 인스턴스에 대해 시간당 사용 요금이나 데이터 전송 요금이 부과되지는 않지만 모든 Amazon EBS 볼륨에 대한 스토리지 요금은 부과됩니다. 중지한 인스턴스를 시작할 때마다 전체 인스턴스 시간 요금이 부과되며, 1시간 내에 여러 번 전환한 경우에도 동일하게 부과됩니다.

인스턴스가 중지되어 있는 동안 해당 루트 볼륨을 다른 볼륨과 마찬가지로 처리하고 수정할 수 있습니다. 예를 들어, 파일 시스템 문제를 복구하거나 소프트웨어를 업데이트 할 수 있습니다. 볼륨을 중지된 인스턴스에서 분리하고 실행 중인 인스턴스에 연결하고 변경한 후 실행 중인 인스턴스에서 분리하고 중지된 인스턴스에 다시 연결하면 됩니다. 볼륨을 다시 연결할 때 인스턴스에 대한 블록 디바이스 매핑에 루트 디바이스로 지정된 스토리지 디바이스 이름을 사용해야 합니다.

더 이상 필요 없는 인스턴스는 종료할 수 있습니다. 인스턴스의 상태가 `shutting-down`이나 `terminated`로 변경되는 즉시 해당 인스턴스에 대한 요금 발생이 중지됩니다. 자세한 내용은 [인스턴스 종료 \(p. 291\)](#) 섹션을 참조하십시오.

목차

- [개요 \(p. 286\)](#)
- [인스턴스 중지 및 시작 \(p. 287\)](#)
- [중지된 인스턴스 수정 \(p. 288\)](#)
- [문제 해결 \(p. 288\)](#)

개요

Amazon EBS 기반 인스턴스만 중지할 수 있습니다. 인스턴스의 루트 디바이스 유형을 확인하려면 인스턴스를 설명하고 해당 루트 볼륨의 디바이스 유형이 `ebs`(Amazon EBS 기반 인스턴스)인지 아니면 `instance store`(인스턴스 스토어 기반 인스턴스)인지 점검합니다. 자세한 내용은 [AMI의 루트 디바이스 유형 결정 \(p. 65\)](#) 섹션을 참조하십시오.

실행 중인 인스턴스를 중지하면 다음과 같이 진행됩니다.

- 인스턴스가 일반적인 종료 과정을 수행하고 실행을 중지하며, 인스턴스의 상태가 `stopping`으로 바뀌었다가 `stopped`로 바뀝니다.
- 모든 Amazon EBS 볼륨이 인스턴스에 연결된 상태로 유지되고 해당 데이터도 남습니다.
- 호스트 컴퓨터의 RAM이나 인스턴스 스토어 볼륨에 저장된 모든 데이터가 손실됩니다.
- 대부분의 경우 인스턴스가 시작되면 새로운 기본 호스트 컴퓨터로 마이그레이션됩니다.
- EC2-Classic: 인스턴스를 중지하면 인스턴스에 대한 퍼블릭 및 프라이빗 IPv4 주소가 해제되고, 인스턴스를 다시 시작할 때 새로 할당됩니다.

EC2-VPC: 인스턴스를 중지했다가 다시 시작할 때 프라이빗 IPv4 주소와 모든 IPv6 주소는 유지됩니다. 퍼블릭 IPv4 주소는 해제되고 인스턴스를 다시 시작할 때 새로 할당됩니다.

- EC2-Classic: 인스턴스와 연결된 탄력적 IP 주소의 연결이 해제됩니다. 인스턴스와 연결되지 않은 탄력적 IP 주소에 대한 요금이 부과됩니다. 인스턴스를 다시 시작할 때 탄력적 IP 주소가 인스턴스와 자동으로 연결되지 않으므로 직접 연결해야 합니다.

EC2-VPC: 인스턴스가 연결된 탄력적 IP 주소를 유지합니다. 중지된 인스턴스와 연결된 모든 탄력적 IP 주소에 대한 요금이 부과됩니다.

- Windows 인스턴스를 중지 및 시작할 때 EC2Config 서비스가 연결된 Amazon EBS 볼륨의 드라이브 문자를 변경하는 등 인스턴스에 대한 작업을 수행합니다. 이러한 기본값에 대한 자세한 내용과 기본값을 변경하는 방법은 Windows 인스턴스용 Amazon EC2 사용 설명서의 [Configuring a Windows Instance Using the EC2Config Service](#) 섹션을 참조하십시오.
- 로드 밸런서에 인스턴스를 등록한 경우 인스턴스를 중지했다가 다시 시작한 후 로드 밸런서에서 해당 인스턴스로 트래픽을 라우팅하지 못할 수 있습니다. 인스턴스를 중지한 후 로드 밸런서에서 인스턴스 등록을 취소한 다음 인스턴스를 시작한 후 다시 등록해야 합니다. 자세한 내용은 Classic Load Balancer 가이드에서 [로드 밸런서로 EC2 인스턴스 등록 취소 및 등록](#)을 참조하십시오.
- 인스턴스가 Auto Scaling 그룹에 있는 경우, Auto Scaling 서비스는 중단된 인스턴스를 비정상으로 간주해 이를 종료하고 대체 인스턴스를 시작합니다. 자세한 내용은 Auto Scaling 사용 설명서의 [Health Checks for Auto Scaling Instances](#)를 참조하십시오.
- ClassicLink 인스턴스를 중지하면 연결되었던 VPC와의 연결이 해제됩니다. 인스턴스를 다시 시작한 후 VPC에 다시 연결해야 합니다. ClassicLink에 대한 자세한 내용은 [ClassicLink \(p. 472\)](#) 섹션을 참조하십시오.

자세한 내용은 [재부팅, 중지 및 종료의 차이 \(p. 263\)](#) 섹션을 참조하십시오.

다음은 인스턴스가 중지되었을 때만 수정할 수 있는 인스턴스의 속성입니다.

- 인스턴스 유형
- 사용자 데이터
- 커널
- RAM 디스크

인스턴스가 실행되고 있을 때 이러한 속성을 수정하려고 하면 Amazon EC2에서 `IncorrectInstanceState` 오류를 반환합니다.

인스턴스 중지 및 시작

콘솔이나 명령줄을 사용하여 Amazon EBS 기반 인스턴스를 시작하고 중지할 수 있습니다.

기본적으로 shutdown, halt 또는 poweroff 명령을 사용하여 Amazon EBS 기반 인스턴스에서 종료를 시작하면 인스턴스가 중지됩니다. 인스턴스가 중지되지 않고 종료되도록 이 동작을 변경할 수 있습니다. 자세한 내용은 [인스턴스가 개시하는 종료 동작 변경 \(p. 293\)](#) 섹션을 참조하십시오.

콘솔을 사용하여 Amazon EBS 기반 인스턴스를 중지하고 시작하려면

1. 탐색 창에서 [Instances]를 선택하고 인스턴스를 선택합니다.
2. [EC2-클래식] 인스턴스에 연결된 탄력적 IP 주소가 있는 경우 세부 정보 창에 표시되는 탄력적 IP 주소와 인스턴스 ID를 기록해둡니다.
3. [Actions]를 선택하고 [Instance State]를 선택한 후 [Stop]을 선택합니다. [Stop]이 비활성화되어 있으면 해당 인스턴스가 이미 중지되었거나 루트 디바이스가 인스턴스 스토어 볼륨인 것입니다.

Warning

인스턴스를 중지하면 인스턴스 스토어 볼륨의 데이터가 삭제됩니다. 따라서 인스턴스 스토어 볼륨에 보존하려는 데이터가 있을 경우 영구 스토리지에 백업하십시오.

4. 확인 대화 상자가 나타나면 Yes, Stop을 선택합니다. 인스턴스가 중지하는 데 몇 분 정도 걸릴 수 있습니다.

- [EC2-Classic] 인스턴스가 `stopped` 상태가 되면 세부 정보 창의 [Elastic IP], [Public DNS (IPv4)], [Private DNS] 및 [Private IPs] 필드가 공백으로 표시됩니다. 이는 기존 값이 인스턴스와 더 이상 연결되어 있지 않음을 의미합니다.
5. 인스턴스가 중지되어 있는 동안 특정 인스턴스 속성을 수정할 수 있습니다. 자세한 내용은 [중지된 인스턴스 수정 \(p. 288\)](#) 섹션을 참조하십시오.
 6. 중지된 인스턴스를 다시 시작하려면 인스턴스를 선택하고, [Actions]를 선택한 후 [Instance State]를 선택하고, [Start]를 선택합니다.
 7. 확인 대화 상자가 나타나면 [Yes, Start]를 선택합니다. 인스턴스가 `running` 상태가 되는 데 몇 분 정도 걸릴 수 있습니다.

- [EC2-Classic] 인스턴스가 `running` 상태가 되면 인스턴스에 할당한 새 값이 세부 정보 창의 [Public DNS (IPv4)], [Private DNS] 및 [Private IPs] 필드에 채워집니다.
8. [EC2-Classic] 인스턴스에 탄력적 IP 주소가 연결되어 있는 경우 다음과 같이 다시 연결해야 합니다.
- a. 탐색 창에서 [Elastic IPs]를 선택합니다.
 - b. 인스턴스를 중지하기 전에 기록해둔 탄력적 IP 주소를 선택합니다.
 - c. Actions를 선택한 후 Associate address를 선택합니다.
 - d. 인스턴스를 중지하기 전에 기록해둔 인스턴스 ID를 선택하고 [Associate]를 선택합니다.

명령줄을 사용하여 Amazon EBS 기반 인스턴스를 중지하고 시작하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [stop-instances](#) 및 [start-instances](#)(AWS CLI)
- [Stop-EC2Instance](#) 및 [Start-EC2Instance](#)(Windows PowerShell용 AWS 도구)

중지된 인스턴스 설정

AWS Management Console 또는 명령줄 인터페이스를 사용하여 중지된 인스턴스의 인스턴스 유형, 사용자 데이터 및 EBS 최적화 속성을 변경할 수 있습니다. AWS Management Console을 사용하여 `DeleteOnTermination`, 커널 또는 RAM 디스크 속성을 수정할 수 없습니다.

인스턴스 속성을 수정하려면

- 인스턴스 유형을 변경하려면 [인스턴스 크기 조정 \(p. 169\)](#)을 참조하십시오.
- 인스턴스의 사용자 데이터를 변경하려면 사용자 데이터를 사용하여 [인스턴스 구성 \(p. 324\)](#)를 참조하십시오.
- 인스턴스의 EBS 최적화를 설정 또는 해제하려면 [EBS 최적화 수정 \(p. 617\)](#)을 참조하십시오.
- 인스턴스의 루트 볼륨의 `DeleteOnTermination` 속성을 변경하려면 다음([실행 인스턴스의 블록 디바이스 매핑 업데이트 \(p. 668\)](#))을 참조하십시오.

명령줄을 사용하여 인스턴스 속성을 수정하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [modify-instance-attribute](#) (AWS CLI)
- [Edit-EC2InstanceAttribute](#) (Windows PowerShell용 AWS 도구)

문제 해결

중지한 Amazon EBS 기반 인스턴스가 `stopping` 상태에서 "멈춘" 것으로 나타나는 경우 해당 인스턴스를 강제로 중지할 수 있습니다. 자세한 내용은 [인스턴스 중지 문제 해결 \(p. 705\)](#) 섹션을 참조하십시오.

인스턴스 재부팅

인스턴스 재부팅은 운영 체제 재부팅과 같습니다. 대부분의 경우 인스턴스를 재부팅하는 데는 몇 분 밖에 걸리지 않습니다. 인스턴스를 재부팅하는 경우 동일한 물리적 호스트에 남아 있으므로 퍼블릭 DNS 이름 (IPv4), 프라이빗 IPv4 주소, IPv6 주소(해당되는 경우) 및 인스턴스 스토어 볼륨의 모든 데이터가 유지됩니다.

인스턴스를 재부팅해도 인스턴스를 종지했다가 다시 시작할 때와는 달리 새 인스턴스 청구 시간이 시작되지 않습니다.

재부팅이 필요한 업데이트를 적용해야 하는 경우와 같이 필수 유지 관리를 위해 인스턴스 재부팅을 예약해야 합니다. 사용자의 별도 작업은 필요하지 않습니다. 예약된 시간 내에 재부팅될 때까지 기다리는 것이 좋습니다. 자세한 내용은 [예약된 인스턴스 이벤트 \(p. 344\)](#) 섹션을 참조하십시오.

Amazon EC2를 사용하여 인스턴스에서 운영 체제 재부팅 명령을 실행하는 대신 인스턴스를 재부팅하는 것이 좋습니다. Amazon EC2를 사용하여 인스턴스를 재부팅하는 경우 해당 인스턴스가 4분 이내에 안전하게 종료되면 하드 재부팅을 수행합니다. AWS CloudTrail을 사용하는 경우 Amazon EC2를 사용하여 인스턴스를 재부팅해도 인스턴스가 재부팅되는 시점의 API 레코드가 생성됩니다.

콘솔을 사용하여 인스턴스를 재부팅하려면

1. Amazon EC2 콘솔을 엽니다.

2. 탐색 창에서 [Instances]를 선택합니다.
3. 인스턴스를 선택하고 [Actions]를 선택한 후 [Instance State]를 선택하고 [Reboot]를 선택합니다.
4. 확인 메시지가 표시되면 [Yes, Reboot]를 선택합니다.

명령줄을 사용하여 인스턴스를 재부팅하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [reboot-instances\(AWS CLI\)](#)
- [Restart-EC2Instance\(Windows PowerShell용 AWS 도구\)](#)

인스턴스 만료

AWS에서 인스턴스를 호스팅하는 기본 하드웨어의 복구 불가능한 장애가 검색되는 경우 인스턴스가 만료 대상으로 예약됩니다. 예약된 만료 날짜에 도달하면 인스턴스가 AWS에 의해 중지되거나 종료됩니다. 인스턴스 루트 디바이스가 Amazon EBS 볼륨인 경우 인스턴스가 중지되며 언제든지 이 인스턴스를 다시 시작할 수 있습니다. 중지된 인스턴스를 시작하면 새 하드웨어로 마이그레이션됩니다. 인스턴스 루트 디바이스가 인스턴스 스토어 볼륨인 경우 인스턴스가 종료되어 다시 사용할 수 없습니다.

항목

- [만료 예약된 인스턴스 식별 \(p. 289\)](#)
- [만료 예약된 인스턴스 관련 작업 \(p. 290\)](#)

인스턴스 이벤트 유형에 대한 자세한 내용은 [예약된 인스턴스 이벤트 \(p. 344\)](#) 섹션을 참조하십시오.

만료 예약된 인스턴스 식별

인스턴스에 대한 만료가 예약되어 있는 경우 만료 이벤트가 발생하기 전에 인스턴스 ID와 만료 날짜가 포함된 이메일이 수신됩니다. 이 이메일은 계정과 연결된 주소로 전송되며, 이 주소는 AWS Management Console에 로그인할 때 사용하는 동일한 이메일 주소입니다. 정기적으로 확인하지 않는 이메일 계정을 사용하는 경우 Amazon EC2 콘솔이나 명령줄을 사용하여 인스턴스 중 하나에 대해 만료가 예약되어 있는지 여부를 확인하십시오. 계정의 연락처 정보를 업데이트하려면 [Account Settings](#) 페이지로 이동하십시오.

콘솔을 사용하여 만료 예약된 인스턴스를 식별하려면

1. Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [EC2 Dashboard]를 선택합니다. [Scheduled Events]에서는 리전별로 구성되어 있는 Amazon EC2 인스턴스 및 볼륨과 연결된 이벤트를 확인할 수 있습니다.
3. 예약된 이벤트가 나열되어 있는 인스턴스가 있는 경우 리전 이름 아래에 있는 링크를 선택하여 [Events] 페이지로 이동합니다.
4. [Events] 페이지에는 모든 리소스 및 연결된 이벤트가 나열됩니다. 만료가 예약되어 있는 인스턴스를 보려면 첫 번째 필터 목록에서 [Instance resources]를 선택하고 두 번째 필터 목록에서 [Instance stop or retirement]를 선택합니다.
5. 필터 결과에 인스턴스에 대한 만료가 예약되어 있는 것으로 나타나면 해당 인스턴스를 선택하고 세부 정보 창의 [Start time] 필드에 표시된 날짜와 시간을 기록해둡니다. 이 날짜가 인스턴스 만료 날짜입니다.

명령줄을 사용하여 만료 예약된 인스턴스를 식별하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [describe-instance-status\(AWS CLI\)](#)
- [Get-EC2InstanceState \(Windows PowerShell용 AWS 도구\)](#)

만료 예약된 인스턴스 관련 작업

인스턴스에 대한 만료가 예약되어 있는 경우 몇 가지 작업을 사용할 수 있습니다. 수행하는 작업은 인스턴스 루트 디바이스가 Amazon EBS 볼륨인지, 인스턴스 스토어 볼륨인지에 따라 달라집니다. 인스턴스 루트 디바이스 유형에 대해 잘 모르는 경우 Amazon EC2 콘솔이나 명령줄을 사용하여 확인할 수 있습니다.

인스턴스 루트 디바이스 유형 확인

콘솔을 사용하여 인스턴스 루트 디바이스 유형을 확인하려면

1. 탐색 창에서 [Events]를 선택합니다. 위의 [만료 예약된 인스턴스 식별 \(p. 289\)](#) 절차의 설명에 따라 필터 목록을 사용하여 만료될 인스턴스를 식별합니다.
2. [Resource ID] 열에서 인스턴스 ID를 선택하여 [Instances] 페이지로 이동합니다.
3. 인스턴스를 선택하고 [Description] 탭의 [Root device type] 필드를 찾습니다. 값이 `ebs`인 경우 EBS 기반 인스턴스이고, 값이 `instance-store`인 경우 인스턴스 스토어 기반 인스턴스입니다.

명령줄을 사용하여 인스턴스 루트 디바이스 유형을 확인하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [describe-instances \(AWS CLI\)](#)
- [Get-EC2Instance \(Windows PowerShell용 AWS 도구\)](#)

만료 예약된 인스턴스 관리

아래 나열된 작업 중 하나를 수행하여 만료될 인스턴스의 데이터를 유지할 수 있습니다. 예기치 않은 종단 시간 및 데이터 손실을 방지하려면 인스턴스 만료 날짜 전에 이 작업을 수행해야 합니다.

Warning

인스턴스 스토어 기반 인스턴스의 만료 날짜가 경과되면 해당 인스턴스가 종료되어 인스턴스 또는 인스턴스에 저장된 모든 데이터를 복구할 수 없게 됩니다. 인스턴스의 루트 디바이스와 관계 없이, EBS 기반 인스턴스에 연결되어 있더라도 인스턴스가 만료되면 인스턴스 스토어 볼륨의 데이터는 손실됩니다.

인스턴스 루트 디바이스 유형	작업
EBS	예약된 만료 날짜까지 기다리거나(인스턴스가 종지되는 경우), 만료 날짜 전에 인스턴스를 종지합니다. 언제든지 인스턴스를 시작할 수 있습니다. 인스턴스 종지 및 시작과 인스턴스 종지 시 발생하는 결과(예: 인스턴스와 연결된 퍼블릭, 프라이빗 및 탄력적 IP 주소에 대한 영향)에 대한 자세한 내용은 인스턴스 종지 및 시작 (p. 285) 섹션을 참조하십시오.
EBS	인스턴스에서 EBS 기반 AMI를 생성하고 대체 인스턴스를 시작합니다. 자세한 내용은 Amazon EBS 지원 Linux AMI 생성 (p. 81) 섹션을 참조하십시오.
인스턴스 스토어	AMI 도구를 사용하여 인스턴스에서 인스턴스 스토어 기반 AMI를 생성하고 대체 인스턴스를 시작합니다. 자세한 내용은 인스턴스 스토어 기반 Linux AMI 생성 (p. 84) 섹션을 참조하십시오.

인스턴스 루트 디바이스 유형	작업
인스턴스 스토어	데이터를 EBS 볼륨으로 전송한 후 볼륨의 스냅샷을 생성하고 스냅샷에서 AMI를 생성하여 인스턴스를 EBS 기반 인스턴스로 변환합니다. 새 AMI에서 대체 인스턴스를 시작할 수 있습니다. 자세한 내용은 인스턴스 스토어 기반 AMI를 Amazon EBS 기반 AMI로 변환 (p. 121) 섹션을 참조하십시오.

인스턴스 종료

더 이상 인스턴스가 필요하지 않다고 판단되면 인스턴스를 종료할 수 있습니다. 인스턴스 상태가 `shutting-down` 또는 `terminated`로 변경되는 즉시 해당 인스턴스에 대한 반복적인 요금 부과가 중단됩니다.

인스턴스를 종료하면 인스턴스에 다시 연결하거나 인스턴스를 재시작할 수 없습니다. 하지만 동일한 AMI를 사용해서 추가 인스턴스를 실행할 수 있습니다. 인스턴스를 중지하고 재시작하려는 경우, [인스턴스 종지 및 시작 \(p. 285\)](#) 섹션을 참조하십시오. 자세한 내용은 [재부팅, 중지 및 종료의 차이 \(p. 263\)](#) 섹션을 참조하십시오.

목차

- [인스턴스 종료 \(p. 291\)](#)
- [인스턴스 종료 \(p. 292\)](#)
- [인스턴스에 대한 종료 방지 기능 활성화 \(p. 292\)](#)
- [인스턴스가 개시하는 종료 동작 변경 \(p. 293\)](#)
- [인스턴스 종료 시 Amazon EBS 볼륨 보존 \(p. 294\)](#)
- [문제 해결 \(p. 295\)](#)

인스턴스 종료

인스턴스는 종료한 후에도 잠시 동안 콘솔에 표시되며 그 이후 항목이 자동으로 삭제됩니다. 종료된 인스턴스 항목을 사용자가 직접 삭제할 수는 없습니다. 인스턴스가 종료된 후 태그 및 볼륨과 같은 리소스는 인스턴스에서 점차 연결 해제되므로 잠시 후 종료된 인스턴스에서 더 이상 보이지 않을 수 있습니다.

인스턴스가 종료하면 해당 인스턴스와 관련된 모든 인스턴스 스토어 볼륨의 데이터는 삭제됩니다.

기본적으로 Amazon EBS 루트 디바이스 볼륨은 인스턴스 종료 시 자동으로 삭제됩니다. 하지만 시작 시 연결하는 추가 EBS 볼륨 또는 기존 인스턴스에 연결하는 EBS 볼륨은 인스턴스가 종료된 후에도 기본적으로 유지됩니다. 이런 동작은 해당 볼륨의 `DeleteOnTermination` 속성에 의해 제어되며, 이러한 속성은 사용자가 변경할 수 있습니다. 자세한 내용은 [인스턴스 종료 시 Amazon EBS 볼륨 보존 \(p. 294\)](#) 섹션을 참조하십시오.

AWS Management Console, CLI, API를 사용하는 타인의 실수로 인스턴스가 종료되는 것을 방지할 수 있습니다. 이 기능은 Amazon EC2 인스턴스 스토어 지원 및 Amazon EBS-지원 인스턴스에 대해 제공됩니다. 각 인스턴스는 `DisableApiTermination` 속성을 가지고 있으며 그 기본 값은 `false`로 설정되어 있습니다(해당 인스턴스는 Amazon EC2를 통해 종료할 수 있음). 인스턴스가 실행 중이거나 중단된 상태에 있을 때 이 인스턴스를 변경할 수 있습니다(Amazon EBS 지원 인스턴스의 경우). 자세한 내용은 [인스턴스에 대한 종료 방지 기능 활성화 \(p. 292\)](#) 섹션을 참조하십시오.

시스템 종료에 대한 운영 체제 명령을 사용해서 인스턴스에서 종료를 개시한 경우, 인스턴스가 중단 또는 종료되는 것을 사용자가 제어할 수 있습니다. 자세한 내용은 [인스턴스가 개시하는 종료 동작 변경 \(p. 293\)](#) 섹션을 참조하십시오.

인스턴스 종료에 대한 스크립트를 사용하는 경우, 종료 스크립트가 안정적으로 실행되는 것을 보장할 방법이 없기 때문에 인스턴스가 비정상적으로 종료될 수 있습니다. Amazon EC2가 인스턴스를 안전하게 종료하고 시스템 종료 스크립트를 실행하도록 작동하지만, 하드웨어 장애 등 특정 이벤트가 이런 시스템 종료 스크립트 실행을 방해할 수 있습니다.

인스턴스 종료

AWS Management Console 또는 명령줄을 사용해서 인스턴스를 종료할 수 있습니다.

콘솔을 사용한 인스턴스 종료 방법

1. 인스턴스를 종료하기 전에, Amazon EBS 볼륨이 종료 시 삭제되지 않는지와 인스턴스 스토어 볼륨에서 Amazon EBS 또는 Amazon S3으로 필요한 데이터를 복사했는지를 확인해서 데이터 손실이 일어나지 않도록 합니다.
2. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
3. 탐색 창에서 [Instances]를 선택합니다.
4. 인스턴스를 선택하고, [Actions]를 선택한 다음, [Instance State]를 선택하고, [Terminate]를 선택합니다.
5. 확인 메시지가 나타나면 [Yes, Terminate]를 선택합니다.

명령줄을 사용한 인스턴스 종료 방법

다음 명령 중 하나를 사용할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [terminate-instances \(AWS CLI\)](#)
- [Stop-EC2Instance \(Windows PowerShell용 AWS 도구\)](#)

인스턴스에 대한 종료 방지 기능 활성화

기본 설정상 Amazon EC2, 콘솔, 명령줄, API를 사용해서 인스턴스를 종료할 수 있습니다. Amazon EC2를 사용할 때 인스턴스가 실수로 종료되지 않도록 방지하려면 해당 인스턴스에 대한 종료 방지 기능을 활성화 할 수 있습니다. `DisableApiTermination` 속성은 콘솔, CLI, API를 사용해서 인스턴스가 종료될 수 있는지를 제어합니다. 기본 설정상 인스턴스에 대한 종료 보호 기능은 비활성화되어 있습니다. 인스턴스를 실행할 때 또는 인스턴스가 실행 중이거나 인스턴스가 중지되어 있을 때, 이 속성의 값을 설정할 수 있습니다(Amazon EBS 지원 인스턴스의 경우).

`DisableApiTermination` 속성은 `InstanceInitiatedShutdownBehavior` 속성이 설정된 때에는 시스템 종료에 대한 운영 체제 명령을 사용해서 인스턴스에서 종료를 개시한 경우의 인스턴스 종료를 방지하지는 않습니다. 자세한 내용은 [인스턴스가 개시하는 종료 동작 변경 \(p. 293\)](#) 섹션을 참조하십시오.

제한

스팟 인스턴스의 종료 방지 기능은 활성화할 수 없습니다. — 스팟 가격이 입찰 가격을 넘어서면 스팟 인스턴스가 종료되기 때문입니다. 하지만 스팟 인스턴스 중단을 처리할 수 있도록 애플리케이션을 준비하는 것은 가능합니다. 자세한 내용은 [스팟 인스턴스 중단 \(p. 242\)](#) 섹션을 참조하십시오.

`DisableApiTermination` 속성으로는 Auto Scaling의 인스턴스 종료를 방지할 수 없습니다. Auto Scaling 그룹에 있는 인스턴스의 경우 Amazon EC2 종료 보호 대신 다음의 Auto Scaling 기능을 사용합니다.

- 인스턴스 보호 기능을 사용하면 확장 시 Auto Scaling 그룹에 속한 인스턴스가 종료되지 않습니다. 자세한 내용은 Auto Scaling 사용 설명서의 [인스턴스 보호](#) 섹션을 참조하십시오.
- Auto Scaling의 비정상 인스턴스 종료를 방지하려면 `ReplaceUnhealthy` 프로세스를 일시 중단하십시오. 자세한 내용은 Auto Scaling 사용 설명서의 [Auto Scaling 프로세스 일시 중단 및 재개](#) 섹션을 참조하십시오.
- 종료 정책을 선택하여 Auto Scaling이 어떤 인스턴스를 먼저 종료해야 할지 지정하십시오. 자세한 내용은 Auto Scaling 사용 설명서의 [종료 정책 사용자 지정](#) 섹션을 참조하십시오.

실행 시에 인스턴스에 대한 종료 방지 기능 활성화 방법

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.

2. 대시보드에서 [Launch Instance]를 선택하고 마법사의 지시를 따릅니다.
3. Configure Instance Details페이지에서 [Enable termination protection] 체크박스를 선택합니다.

실행 중인 또는 중단된 인스턴스에 대한 종료 방지 기능 활성화 방법

1. 인스턴스를 선택하고 [Actions], [Instance Settings], [Change Termination Protection]을 차례로 선택합니다.
2. [Yes, Enable]을 선택합니다.

실행 중인 또는 중단된 인스턴스에 대한 종료 방지 기능 비활성화 방법

1. 인스턴스를 선택하고 [Actions]를 선택하고 [Instance Settings]를 선택한 다음에 [Change Termination Protection]을 선택합니다.
2. [Yes, Disable]을 선택합니다.

명령줄을 사용한 종료 방지 기능의 활성화 또는 비활성화 방법

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [modify-instance-attribute](#) (AWS CLI)
- [Edit-EC2InstanceAttribute](#) (Windows PowerShell용 AWS 도구)

인스턴스가 개시하는 종료 동작 변경

기본 설정상 shutdown, halt, poweroff 등 명령을 사용해서 Amazon EBS 지원 인스턴스에서 종료를 개시할 때, 인스턴스는 중단됩니다. 인스턴스에 대한 `InstanceInitiatedShutdownBehavior` 속성을 사용해서 이런 동작을 변경해서 인스턴스가 중단되지 않고 종료되도록 할 수 있습니다. 인스턴스가 실행 중이거나 중단된 상태에 있을 때 이 속성을 업데이트할 수 있습니다.

인스턴스 스토어가 지원하는 인스턴스는 종료할 수 있지만 중지할 수 없음에 유의하십시오.

Amazon EC2 콘솔이나 명령줄을 사용해서 `InstanceInitiatedShutdownBehavior` 속성을 업데이트할 수 있습니다. `InstanceInitiatedShutdownBehavior` 속성은 인스턴스 자체의 운영 체제에서 종료하는 경우에만 적용되며, `StopInstances` API 또는 Amazon EC2 콘솔을 사용하여 인스턴스를 종지하는 경우에는 적용되지 않습니다.

콘솔을 사용한 인스턴스의 종료 동작 변경 방법

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. 인스턴스를 선택하고 [Actions], [Instance Settings], [Change Shutdown Behavior]를 차례로 선택합니다. 현재 동작은 이미 선택된 상태입니다.
4. 동작을 변경하려면 [Shutdown behavior] 목록에서 옵션을 선택하고 [Apply]를 선택합니다.

명령줄을 사용한 인스턴스의 종료 동작 변경 방법

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [modify-instance-attribute](#) (AWS CLI)
- [Edit-EC2InstanceAttribute](#) (Windows PowerShell용 AWS 도구)

인스턴스 종료 시 Amazon EBS 볼륨 보존

인스턴스가 종료되면 Amazon EC2가 연결된 각 Amazon EBS 볼륨의 `DeleteOnTermination` 속성 값을 사용하여 볼륨 유지 또는 삭제 여부를 결정합니다.

기본적으로 인스턴스의 루트 볼륨의 `DeletionOnTermination` 속성은 `true`로 설정됩니다. 따라서 기본값은 인스턴스가 종료될 때 인스턴스의 루트 볼륨을 삭제하는 것입니다.

기본적으로 EBS 볼륨을 인스턴스에 연결하면 그 `DeleteOnTermination` 속성이 `false`로 설정됩니다. 따라서 기본값은 이러한 볼륨을 유지하는 것입니다. 인스턴스가 종료된 후에 유지된 볼륨의 스냅샷을 만들거나 다른 인스턴스에 연결할 수 있습니다.

사용 중인 EBS 볼륨의 `DeleteOnTermination` 속성 값을 확인하려면 인스턴스의 블록 디바이스 매핑을 검색합니다. 자세한 내용은 [인스턴스 블록 디바이스 매핑에서 EBS 볼륨 보기 \(p. 668\)](#) 섹션을 참조하십시오.

인스턴스를 시작할 때 또는 인스턴스 실행 중에 볼륨의 `DeleteOnTermination` 속성 값을 변경할 수 있습니다.

예제

- [콘솔을 사용하여 실행 시 유지할 루트 볼륨 변경 \(p. 294\)](#)
- [명령줄을 사용하여 실행 시 유지할 루트 볼륨 변경 \(p. 294\)](#)
- [명령줄을 사용해서 실행 중인 인스턴스의 루트 볼륨이 유지되도록 변경 \(p. 295\)](#)

콘솔을 사용하여 실행 시 유지할 루트 볼륨 변경

콘솔을 사용하면 인스턴스를 시작할 때 `DeleteOnTermination` 속성을 변경할 수 있습니다. 실행 중인 인스턴스에 대한 속성을 변경하려면 명령줄을 사용해야 합니다.

콘솔을 사용해서 실행 시에 인스턴스의 루트 볼륨이 유지되도록 변경하는 방법

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 콘솔 대시보드에서 [Launch Instance]를 선택합니다.
3. [Choose an Amazon Machine Image (AMI)] 페이지에서 AMI를 선택한 다음 [Select]를 선택합니다.
4. 마법사 안내에 따라 [Choose an Instance Type] 및 [Configure Instance Details] 설정을 완료합니다.
5. [Add Storage] 페이지에서 루트 볼륨에 대한 [Delete On Termination] 확인란 선택을 해제합니다.
6. 나머지 마법사 페이지를 완료한 다음 [Launch]를 선택합니다.

인스턴스의 세부 정보 창에서 루트 디바이스 볼륨의 세부 정보를 조회하여 설정을 확인할 수 있습니다. [Block devices] 옆의 루트 디바이스 볼륨 항목을 클릭합니다. [Delete on termination]의 기본 설정은 `True`입니다. 기본 설정을 변경하면 [Delete on termination]의 설정 값이 `False`가 됩니다.

명령줄을 사용하여 실행 시 유지할 루트 볼륨 변경

EBS 지원 인스턴스를 시작할 때 다음 명령 중 하나를 사용해서 루트 디바이스 볼륨이 유지되도록 변경할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [run-instances\(AWS CLI\)](#)
- [New-EC2Instance\(Windows PowerShell용 AWS 도구\)](#)

예를 들어, 다음 옵션을 `run-instances` 명령에 추가합니다.

```
--block-device-mappings file://mapping.json
```

mapping.json에서 다음을 지정합니다.

```
[  
  {  
    "DeviceName": "/dev/sda1",  
    "Ebs": {  
      "DeleteOnTermination": false,  
      "SnapshotId": "snap-1234567890abcdef0",  
      "VolumeType": "gp2"  
    }  
  }  
]
```

명령줄을 사용해서 실행 중인 인스턴스의 루트 볼륨이 유지되도록 변경

다음 명령 중 하나를 사용하여 실행 중인 EBS 지원 인스턴스의 루트 장치 볼륨이 유지되도록 변경할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [modify-instance-attribute](#) (AWS CLI)
- [Edit-EC2InstanceAttribute](#) (Windows PowerShell용 AWS 도구)

예를 들어, 다음 명령을 사용합니다.

```
$ aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --block-device-mappings file://mapping.json
```

mapping.json에서 다음을 지정합니다.

```
[  
  {  
    "DeviceName": "/dev/sda1",  
    "Ebs": {  
      "DeleteOnTermination": false  
    }  
  }  
]
```

문제 해결

인스턴스가 `shutting-down` 상태에 일반적인 경우보다 장기간 머물러 있는 경우, 해당 인스턴스는 Amazon EC2 서비스 내 자동화된 과정에 의해 클린업(종료)됩니다. 자세한 내용은 [인스턴스 종료 문제 해결 \(p. 706\)](#) 섹션을 참조하십시오.

인스턴스 복구

사용자는 Amazon EC2 인스턴스를 모니터링하고 기본 하드웨어 장애나 복구에 AWS 개입이 필요한 문제로 인해 인스턴스가 손상된 경우 인스턴스를 자동으로 복구하는 Amazon CloudWatch 경보를 만들 수 있습니다. 종료한 인스턴스는 복구할 수 없습니다. 복구된 인스턴스는 인스턴스 ID, 프라이빗 IP 주소, 탄력적 IP 주소 및 모든 인스턴스 메타데이터를 포함하여 원본 인스턴스와 동일합니다. 인스턴스 복구를 위한 Amazon CloudWatch 경보 사용의 자세한 내용은 [인스턴스를 종지, 종료, 재부팅 또는 복구하는 경보 만들기 \(p. 360\)](#) 섹션을 참조하십시오. 인스턴스 복구 실패 문제를 해결하려면 Linux 인스턴스용 Amazon EC2 사용 설명서의 [인스턴스 복구 실패 문제 해결](#)을 참조하십시오.

`statusCheckFailed_System` 경보가 트리거되고 복구 작업이 시작되는 경우 경보를 생성하고 복구 작업을 연결할 때 선택한 Amazon SNS 주제로 통지됩니다. 인스턴스 복구 중에 인스턴스를 재부팅할 때 인스턴스가 마이그레이션되고 모든 인 메모리 데이터가 손실됩니다. 프로세스가 완료되면 해당 경보를 위해 구성해 둔

SNS 주제로 정보가 게시됩니다. 이 SNS 주제에 가입되어 있는 사람은 누구나 복구 시도 상태와 세부 지침이 포함된 이메일 알림을 받게 됩니다. 복구된 인스턴스에서 인스턴스를 재부팅하라는 메시지가 나타납니다.

시스템 상태 확인이 실패하게 되는 문제의 예를 들면 다음과 같습니다.

- 네트워크 연결 끊김
- 시스템 전원 중단
- 물리적 호스트의 소프트웨어 문제
- 물리적 호스트의 하드웨어 문제 네트워크 도달 가능성 개선

또한 복구 작업은 기본 하드웨어의 성능 저하로 인해 AWS가 인스턴스를 중지 또는 만료하도록 예약할 때 트리거될 수도 있습니다. 예약된 이벤트에 대한 자세한 내용은 [예약된 인스턴스 이벤트 \(p. 344\)](#) 섹션을 참조하십시오.

복구 작업은 다음 특성을 지닌 인스턴스에만 지원됩니다.

- C3, C4, M3, M4, R3, R4, T2 또는 X1 인스턴스 유형 사용
- VPC(EC2-Classic 아님)에서 실행
- 공유 테넌시 사용(테넌시 속성이 default로 설정되어 있음)
- EBS 볼륨(인스턴스 스토어 볼륨을 구성하지 않음)만 사용합니다. 자세한 정보는 '[Recover this instance](#)' is disabled 단원을 참조하십시오.

인스턴스에 퍼블릭 IPv4 주소가 있는 경우 복구 후에도 해당 퍼블릭 IPv4 주소를 유지합니다.

Amazon Linux 인스턴스 구성

Amazon Linux 인스턴스를 시작하여 로그인한 후 인스턴스를 변경할 수 있습니다. 특정 애플리케이션의 요구 사항에 맞춰 다양한 방법으로 인스턴스를 구성할 수 있습니다. 다음은 관련 내용을 익히는데 도움이 되는 몇 가지 일반적인 작업입니다.

목차

- [일반적인 구성 시나리오 \(p. 296\)](#)
- [Linux 인스턴스의 소프트웨어 관리 \(p. 297\)](#)
- [Linux 인스턴스의 사용자 계정 관리 \(p. 304\)](#)
- [EC2 인스턴스에 대한 프로세서 상태 제어 \(p. 306\)](#)
- [Linux 인스턴스의 시간 설정 \(p. 310\)](#)
- [Linux 인스턴스의 호스트 이름 변경 \(p. 314\)](#)
- [Your Linux 인스턴스에 동적 DNS 설정 \(p. 316\)](#)
- [시작 시 Linux 인스턴스에서 명령 실행 \(p. 317\)](#)
- [인스턴스 메타데이터 및 사용자 데이터 \(p. 321\)](#)

일반적인 구성 시나리오

Amazon Linux의 기본 배포에는 기본적인 서버 작업에 필요한 여러 가지 소프트웨어 패키지 및 유틸리티가 포함되어 있습니다. 이외에도 다양한 소프트웨어 리포지토리의 더 많은 소프트웨어 패키지를 사용할 수 있고, 훨씬 더 많은 패키지를 소스 코드로 개발할 수 있습니다. 이러한 위치의 소프트웨어를 설치 및 개발하는 방법에 대한 자세한 내용은 [Linux 인스턴스의 소프트웨어 관리 \(p. 297\)](#) 섹션을 참조하십시오.

Amazon Linux 인스턴스는 ec2-user 계정으로 미리 구성되어 제공되지만 수퍼유저 권한이 없는 다른 사용자 계정을 추가할 수도 있습니다. 사용자 계정 추가 및 제거에 대한 자세한 내용은 [Linux 인스턴스의 사용자 계정 관리 \(p. 304\)](#) 섹션을 참조하십시오.

Amazon Linux 인스턴스의 기본 시간 구성에서는 NTP(Network Time Protocol)를 사용하여 인스턴스의 시스템 시간을 설정합니다. 기본 표준 시간대는 UTC입니다. 인스턴스의 표준 시간대를 설정하거나 자체 시간 서버를 사용하는 방법에 대한 자세한 내용은 [Linux 인스턴스의 시간 설정 \(p. 310\)](#) 섹션을 참조하십시오.

도메인 이름이 등록된 자체 네트워크를 보유한 경우 인스턴스의 호스트 이름을 변경하여 해당 도메인에 속한 것으로 표시할 수 있습니다. 호스트 이름 설정은 그대로 두고 시스템 프롬프트를 더욱 의미 있는 이름으로 변경할 수도 있습니다. 자세한 내용은 [Linux 인스턴스의 호스트 이름 변경 \(p. 314\)](#)을 참조하십시오. 인스턴스에서 동적 DNS 서비스 공급자를 사용하도록 구성할 수 있습니다. 자세한 내용은 [Your Linux 인스턴스에 동적 DNS 설정 \(p. 316\)](#) 섹션을 참조하십시오.

Amazon EC2에서 인스턴스를 시작할 때 사용자 데이터를 인스턴스에 전달하여 일반적인 구성 작업을 수행하는 데 사용하도록 할 수 있고, 인스턴스가 시작된 후에 스크립트를 실행할 수도 있습니다. Amazon EC2에 두 가지 유형의 사용자 데이터(`cloud-init` 명령 및 shell 스크립트)를 전달할 수 있습니다. 자세한 내용은 [시작 시 Linux 인스턴스에서 명령 실행 \(p. 317\)](#) 섹션을 참조하십시오.

Linux 인스턴스의 소프트웨어 관리

Amazon Linux의 기본 배포에는 기본적인 서버 작업에 필요한 여러 가지 소프트웨어 패키지 및 유ти리티가 포함되어 있습니다. 이외에도 다양한 소프트웨어 리포지토리의 더 많은 소프트웨어 패키지를 사용할 수 있고, 훨씬 더 많은 패키지를 소스 코드로 개발할 수 있습니다.

목차

- [인스턴스 소프트웨어 업데이트 \(p. 297\)](#)
- [리포지토리 추가 \(p. 300\)](#)
- [소프트웨어 패키지 찾기 \(p. 302\)](#)
- [소프트웨어 패키지 설치 \(p. 303\)](#)
- [소프트웨어 컴파일 준비 \(p. 304\)](#)

소프트웨어를 최신 상태로 유지하는 것이 중요합니다. Linux 배포의 다양한 패키지가 버그 수정, 기능 추가 및 보안 취약점 해결을 위해 자주 업데이트됩니다. 자세한 내용은 [인스턴스 소프트웨어 업데이트 \(p. 297\)](#) 섹션을 참조하십시오.

기본적으로 Amazon Linux 인스턴스는 두 리포지토리(`amzn-main` 및 `amzn-updates`)가 활성화된 상태로 시작됩니다. Amazon Web Services에서 업데이트하는 이러한 리포지토리의 다양한 패키지 이외에도 다른 리포지토리에 포함된 패키지를 설치할 수 있습니다. 자세한 내용은 [리포지토리 추가 \(p. 300\)](#)을 참조하십시오. 활성화된 리포지토리에서 패키지를 찾는 방법은 [소프트웨어 패키지 찾기 \(p. 302\)](#) 섹션을 참조하십시오. Amazon Linux 인스턴스에 소프트웨어를 설치하는 방법은 [소프트웨어 패키지 설치 \(p. 303\)](#) 섹션을 참조하십시오.

리포지토리에 저장된 소프트웨어 패키지만 사용할 수 있는 것은 아닙니다. 일부 소프트웨어의 경우 인스턴스에서 소스 코드를 컴파일해야 합니다. 자세한 내용은 [소프트웨어 컴파일 준비 \(p. 304\)](#) 섹션을 참조하십시오.

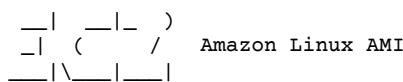
Amazon Linux 인스턴스는 yum 패키지 관리자를 사용하여 소프트웨어를 관리합니다. yum 패키지 관리자는 소프트웨어를 설치, 제거 및 업데이트하고 각 패키지의 모든 종속성을 관리할 수 있습니다. Ubuntu 등의 Debian 기반 Linux 배포에서는 apt-get 명령 및 dpkg 패키지 관리자를 사용하므로 해당 배포에서는 다음 섹션의 yum 예제가 작동하지 않습니다.

인스턴스 소프트웨어 업데이트

소프트웨어를 최신 상태로 유지하는 것이 중요합니다. Linux 배포의 다양한 패키지가 버그 수정, 기능 추가 및 보안 취약점 해결을 위해 자주 업데이트됩니다. 처음으로 Amazon Linux 인스턴스를 시작하여 연결하면 보안을 위해 소프트웨어 패키지를 업데이트하라는 메시지가 표시될 수 있습니다. 이 섹션에서는 전체 시스템 또는 단일 패키지를 업데이트하는 방법을 보여 줍니다.

Important

이 절차는 Amazon Linux에서 사용하기 위한 것입니다. 기타 배포에 대한 자세한 내용은 해당 설명서를 참조하십시오.



```
https://aws.amazon.com/amazon-linux-ami/2013.03-release-notes/  
There are 12 security update(s) out of 25 total update(s) available  
Run "sudo yum update" to apply all updates.  
[ec2-user ~]$
```

Amazon Linux 인스턴스의 모든 패키지를 업데이트하려면 다음을 수행합니다.

1. (선택 사항) shell 창에서 screen 세션을 시작합니다. 경우에 따라 네트워크 장애로 인해 인스턴스에 대한 SSH 연결이 끊어질 수 있습니다. 오래 걸리는 소프트웨어 업데이트 중에 연결이 끊어진 경우 인스턴스가 복구 가능한 흔동 상태로 유지될 수 있습니다. 연결이 끊어진 경우에도 screen 세션을 통해 업데이트가 계속 실행되며, 이후에 아무런 문제 없이 세션에 다시 연결할 수 있습니다.

- a. screen 명령을 실행하여 세션을 시작합니다.

```
[ec2-user ~]$ screen
```

- b. 세션의 연결이 끊어진 경우 인스턴스에 다시 로그인하고 사용 가능한 화면을 나열합니다.

```
[ec2-user ~]$ screen -ls  
There is a screen on:  
 17793.pts-0.ip-12-34-56-78 (Detached)  
 1 Socket in /var/run/screen/S-ec2-user.
```

- c. 이전 명령에서 확인한 프로세스 ID와 screen -r 명령을 사용하여 화면에 다시 연결합니다.

```
[ec2-user ~]$ screen -r 17793
```

- d. screen 사용을 마쳤으면 exit 명령을 사용하여 세션을 닫습니다.

```
[ec2-user ~]$ exit  
[screen is terminating]
```

2. yum update 명령을 실행합니다. --security 플래그를 추가하여 보안 업데이트만 적용할 수도 있습니다.

```
[ec2-user ~]$ sudo yum update  
Loaded plugins: priorities, security, update-motd, upgrade-helper  
amzn-main                                         | 2.1 kB   00:00  
amzn-updates                                      | 2.3 kB   00:00  
Setting up Update Process  
Resolving Dependencies  
--> Running transaction check  
--> Package aws-apitools-ec2.noarch 0:1.6.8.1-1.0.amzn1 will be updated  
--> Package aws-apitools-ec2.noarch 0:1.6.10.0-1.0.amzn1 will be an update  
--> Package gnupg2.x86_64 0:2.0.18-1.16.amzn1 will be updated  
--> Package gnupg2.x86_64 0:2.0.19-8.21.amzn1 will be an update  
--> Package libgcrypt.i686 0:1.4.5-9.10.amzn1 will be updated  
--> Package libgcrypt.x86_64 0:1.4.5-9.10.amzn1 will be updated  
--> Package libgcrypt.i686 0:1.4.5-9.12.amzn1 will be an update  
--> Package libgcrypt.x86_64 0:1.4.5-9.12.amzn1 will be an update  
--> Package openssl.x86_64 1:1.0.1e-4.53.amzn1 will be updated
```

```

---> Package openssl.x86_64 1:1.0.0-1.0.1e-4.54.amzn1 will be an update
---> Package python-boto.noarch 0:2.9.9-1.0.amzn1 will be updated
---> Package python-boto.noarch 0:2.13.3-1.0.amzn1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package          Arch      Version       Repository    Size
=====
Updating:
aws-apitools-ec2   noarch   1.6.10.0-1.0.amzn1   amzn-updates  14 M
gnupg2            x86_64   2.0.19-8.21.amzn1   amzn-updates  2.4 M
libgcrypt          i686     1.4.5-9.12.amzn1   amzn-updates  248 k
libgcrypt          x86_64   1.4.5-9.12.amzn1   amzn-updates  262 k
openssl            x86_64   1:1.0.0-1.0.1e-4.54.amzn1 amzn-updates  1.7 M
python-boto         noarch   2.13.3-1.0.amzn1   amzn-updates  1.6 M

Transaction Summary
=====
Upgrade      6 Package(s)

Total download size: 20 M
Is this ok [y/N]:
```

3. 나열된 패키지를 검토하고 y를 입력한 후 Enter 키를 눌러 업데이트를 수락합니다. 시스템의 모든 패키지를 업데이트하는데 몇 분이 걸릴 수 있습니다. yum 출력은 실행 중인 업데이트의 상태를 보여 줍니다.

```

Downloading Packages:
(1/6): aws-apitools-ec2-1.6.10.0-1.0.amzn1.noarch.rpm | 14 MB  00:00
(2/6): gnupg2-2.0.19-8.21.amzn1.x86_64.rpm           | 2.4 MB  00:00
(3/6): libgcrypt-1.4.5-9.12.amzn1.i686.rpm           | 248 kB  00:00
(4/6): libgcrypt-1.4.5-9.12.amzn1.x86_64.rpm         | 262 kB  00:00
(5/6): openssl-1.0.0-1.0.1e-4.54.amzn1.x86_64.rpm    | 1.7 MB  00:00
(6/6): python-boto-2.13.3-1.0.amzn1.noarch.rpm       | 1.6 MB  00:00
-----
Total                                         28 MB/s | 20 MB  00:00

Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating    : libgcrypt-1.4.5-9.12.amzn1.x86_64          1/12
  Updating    : gnupg2-2.0.19-8.21.amzn1.x86_64           2/12
  Updating    : aws-apitools-ec2-1.6.10.0-1.0.amzn1.noarch 3/12
  Updating    : 1:openssl-1.0.0-1.0.1e-4.54.amzn1.x86_64   4/12
...
Complete!
```

4. (선택 사항) 인스턴스를 재부팅하여 업데이트에서 최신 패키지 및 라이브러리를 사용 중인지를 확인합니다. 커널 업데이트를 로드하려면 재부팅해야 합니다. glibc 라이브러리를 업데이트한 이후에도 항상 재부팅해야 합니다. 서비스를 제어하는 패키지를 업데이트할 경우 서비스를 다시 시작하여 업데이트를 선택하면 되지만, 시스템을 재부팅하면 이전의 모든 패키지 및 라이브러리 업데이트가 완료됩니다.

Amazon Linux 인스턴스의 단일 패키지를 업데이트하려면 다음을 수행합니다.

이 절차를 사용하여 전체 시스템이 아닌 단일 패키지와 해당 종속 패키지를 업데이트할 수 있습니다.

- 업데이트할 패키지의 이름과 함께 yum update 명령을 실행합니다.

```
[ec2-user ~]$ sudo yum update openssl
Loaded plugins: priorities, security, update-motd, upgrade-helper
```

```
amzn-main | 2.1 kB 00:00
amzn-updates | 2.3 kB 00:00
Setting up Update Process
Resolving Dependencies
--> Running transaction check
---> Package openssl.x86_64 1:1.0.1e-4.53.amzn1 will be updated
---> Package openssl.x86_64 1:1.0.1e-4.54.amzn1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package      Arch      Version      Repository      Size
=====
Updating:
openssl      x86_64    1:1.0.1e-4.54.amzn1    amzn-updates   1.7 M

Transaction Summary
=====
Upgrade      1 Package(s)

Total download size: 1.7 M
Is this ok [y/N]:
```

2. 나열된 패키지 정보를 검토하고 y를 입력한 후 Enter 키를 눌러 업데이트를 수락합니다. 해결되어야 하는 패키지 종속성이 있는 경우 들 이상의 패키지가 나열될 수 있습니다. yum 출력은 실행 중인 업데이트의 상태를 보여 줍니다.

```
Downloading Packages:
openssl-1.0.1e-4.54.amzn1.x86_64.rpm | 1.7 MB 00:00
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating : 1:openssl-1.0.1e-4.54.amzn1.x86_64 1/2
  Cleanup  : 1:openssl-1.0.1e-4.53.amzn1.x86_64 2/2
  Verifying : 1:openssl-1.0.1e-4.54.amzn1.x86_64 1/2
  Verifying : 1:openssl-1.0.1e-4.53.amzn1.x86_64 2/2

Updated:
  openssl.x86_64 1:1.0.1e-4.54.amzn1

Complete!
```

3. (선택 사항) 인스턴스를 재부팅하여 업데이트에서 최신 패키지 및 라이브러리를 사용 중인지를 확인합니다. 커널 업데이트를 로드하려면 재부팅해야 합니다. glibc 라이브러리를 업데이트한 이후에도 항상 재부팅해야 합니다. 서비스를 제어하는 패키지를 업데이트할 경우 서비스를 다시 시작하여 업데이트를 선택하면 되지만, 시스템을 재부팅하면 이전의 모든 패키지 및 라이브러리 업데이트가 완료됩니다.

리포지토리 추가

기본적으로 Amazon Linux 인스턴스는 두 리포지토리(amzn-main 및 amzn-updates)가 활성화된 상태로 시작됩니다. Amazon Web Services에서 업데이트하는 이러한 리포지토리의 다양한 패키지 이외에도 다른 리포지토리에 포함된 패키지를 설치할 수 있습니다.

Important

이 절차는 Amazon Linux에서 사용하기 위한 것입니다. 기타 배포에 대한 자세한 내용은 해당 설명서를 참조하십시오.

yum이 아닌 다른 리포지토리의 패키지를 설치하려면 /etc/yum.repos.d 디렉터리의 /etc/yum.conf 파일 또는 자체 **repository.repo** 파일에 리포지토리 정보를 추가해야 합니다. 이 작업을 직접 수행할 수도 있지만, 대부분의 yum 리포지토리는 리포지토리 URL을 통해 자체 **repository.repo** 파일을 제공합니다.

yum 리포지토리가 이미 설치되어 있는지 확인하려면

- 다음 명령을 사용하여 설치되어 있는 yum 리포지토리를 조회합니다.

```
[ec2-user ~]$ yum repolist all
```

명령 결과에 설치된 리포지토리가 출력되고 각 상태가 보고됩니다. 사용 가능한 리포지토리에는 해당 리포지토리에 포함된 패키지 수가 표시됩니다.

repo id	status	repo name
!amzn-main/latest	enabled: 5,612	amzn-main-Base
amzn-main-debuginfo/latest	disabled	amzn-main-debuginfo
amzn-main-source/latest	disabled	amzn-main-source
amzn-nosrc/latest	disabled	amzn-nosrc-Base
amzn-preview/latest	disabled	amzn-preview-Base
amzn-preview-debuginfo/latest	disabled	amzn-preview-debuginfo
amzn-preview-source/latest	disabled	amzn-preview-source
!amzn-updates/latest	enabled: 1,152	amzn-updates-Base
amzn-updates-debuginfo/latest	disabled	amzn-updates-debuginfo
amzn-updates-source/latest	disabled	amzn-updates-source
epel/x86_64	disabled	Extra Packages for Enterprise Linux 6 - x86_64
epel-debuginfo/x86_64	disabled	Extra Packages for Enterprise Linux 6 - x86_64 - Debug
epel-source/x86_64	disabled	Extra Packages for Enterprise Linux 6 - x86_64 - Source
epel-testing/x86_64	disabled	Extra Packages for Enterprise Linux 6 - Testing - x86_64
epel-testing-debuginfo/x86_64	disabled	Extra Packages for Enterprise Linux 6 - Testing - x86_64 - Debug
epel-testing-source/x86_64	disabled	Extra Packages for Enterprise Linux 6 - Testing - x86_64 - Source

/etc/yum.repos.d에 yum 리포지토리를 추가하려면 다음을 수행합니다.

리포지토리를 설치한 후 다음 절차에 따라 리포지토리를 활성화해야 합니다.

- .repo 파일의 위치를 찾습니다. 위치는 추가할 리포지토리에 따라 다를 수 있습니다. 이 예제에서는 .repo 파일이 <https://www.example.com/repository.repo>에 있습니다.
- yum-config-manager 명령으로 리포지토리를 추가합니다.

```
[ec2-user ~]$ sudo yum-config-manager --add-repo https://www.example.com/repository.repo
Loaded plugins: priorities, update-motd, upgrade-helper
adding repo from: https://www.example.com/repository.repo
```

```
grabbing file https://www.example.com/repository.repo to /etc/yum.repos.d/repository.repo
repository.repo | 4.0 kB      00:00
repo saved to /etc/yum.repos.d/repository.repo
```

/etc/yum.repos.d의 yum 리포지토리를 활성화하려면 다음을 수행합니다.

- enable repository 플래그와 함께 yum-config-manager 명령을 사용합니다. 다음 명령은 Fedora 프로젝트의 EPEL(Extra Packages for Enterprise Linux) 리포지토리를 활성화합니다. 이 리포지토리는 기본적으로 Amazon Linux 인스턴스의 /etc/yum.repos.d에 있지만 활성화되지 않은 상태입니다.

```
[ec2-user ~]$ sudo yum-config-manager --enable epel
```

Note

Red Hat, CentOS 등의 다른 배포에서 EPEL 리포지토리를 활성화하는 방법은 <https://fedoraproject.org/wiki/EPEL>의 EPEL 설명서를 참조하십시오.

소프트웨어 패키지 찾기

yum search 명령을 사용하여 구성된 리포지토리에서 사용 가능한 패키지 설명을 검색할 수 있습니다. 이 기능은 설치할 패키지의 이름을 정확히 알지 못할 때 특히 유용합니다. 명령에 검색어를 덧붙이기만 하면 됩니다. 여러 단어를 검색하려는 경우 검색어를 따옴표로 묶습니다.

Important

이 절차는 Amazon Linux에서 사용하기 위한 것입니다. 기타 배포에 대한 자세한 내용은 해당 설명서를 참조하십시오.

여러 단어를 따옴표로 묶은 검색어를 사용하면 검색어와 정확히 일치하는 결과만 반환됩니다. 원하는 패키지가 검색되지 않은 경우 키워드를 하나만 사용하여 검색한 후 결과를 살펴봅니다. 뜻이 같은 키워드를 사용하여 검색 범위를 넓힐 수도 있습니다.

```
[ec2-user ~]$ sudo yum search "find"
Loaded plugins: priorities, security, update-motd, upgrade-helper
=====
N/S Matched: find
=====
findutils.x86_64 : The GNU versions of find utilities (find and xargs)
perl-File-Find-Rule.noarch : Perl module implementing an alternative interface
                             : to File::Find
perl-Module-Find.noarch : Find and use installed modules in a (sub)category
libpuzzle.i686 : Library to quickly find visually similar images (gif, png, jpg)
libpuzzle.x86_64 : Library to quickly find visually similar images (gif, png,
                   : jpg)
mlocate.x86_64 : An utility for finding files by name
```

yum 패키지 관리자는 웹 서버 설치, 소프트웨어 컴파일 도구 작성 등의 특정 작업을 수행하기 위해 여러 패키지를 한 명령으로 설치할 수 있도록 서로 그룹으로 묶기도 합니다. 시스템에 이미 설치된 그룹 및 설치 가능한 그룹을 나열하려면 yum grouplist 명령을 사용합니다.

```
[ec2-user ~]$ sudo yum grouplist
Loaded plugins: priorities, security, update-motd, upgrade-helper
Setting up Group Process
Installed Groups:
  Development Libraries
  Development tools
  Editors
  Legacy UNIX compatibility
```

```
Mail Server
MySQL Database
Network Servers
Networking Tools
PHP Support
Perl Support
System Tools
Web Server
Available Groups:
Console internet tools
DNS Name Server
FTP Server
Java Development
MySQL Database client
NFS file server
Performance Tools
PostgreSQL Database client (version 8)
PostgreSQL Database server (version 8)
Scientific support
TeX support
Technical Writing
Web Servlet Engine
Done
```

yum groupinfo "## ##" 명령을 사용하여 그룹에 포함된 여러 패키지를 확인할 수 있습니다. ## ## 자리에 정 보를 확인할 그룹의 이름을 넣으면 됩니다. 이 명령은 해당 그룹과 함께 설치될 수 있는 필수, 기본 및 선택 패 키지를 모두 나열합니다.

기본 amzn-main 및 amzn-updates 리포지토리에서 필요한 소프트웨어를 찾을 수 없을 경우 리포지토리 를 더 추가할 수 있습니다(예: EPEL(Extra Packages for Enterprise Linux)). 자세한 내용은 [리포지토리 추 가 \(p. 300\)](#)를 참조하십시오.

소프트웨어 패키지 설치

yum 패키지 관리자는 소프트웨어 설치를 위한 탁월한 도구로서 활성화된 모든 리포지토리를 검색하여 다양 한 소프트웨어 패키지를 찾을 뿐 아니라 소프트웨어 설치 과정에서 모든 종속성을 자동으로 처리합니다.

Important

이 절차는 Amazon Linux에서 사용하기 위한 것입니다. 기타 배포에 대한 자세한 내용은 해당 설명 서를 참조하십시오.

리포지토리의 패키지를 설치하려면 yum install ### 명령을 사용합니다. *package* 자리에 설치할 소프트웨어 의 이름을 넣으면 됩니다. 예를 들어 links 텍스트 기반 웹 브라우저를 설치하려면 다음 명령을 입력합니다.

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

패키지 그룹을 설치하려면 yum groupinstall ## ## 명령을 사용합니다. ## ## 자리에 설치할 그룹의 이름을 넣으면 됩니다. 예를 들어 "Performance Tools" 그룹을 설치하려면 다음 명령을 입력합니다.

```
[ec2-user@ip-10-161-113-54 ~]$ sudo yum groupinstall "Performance Tools"
```

기본적으로 yum은 그룹 목록의 필수 및 기본 패키지만 설치합니다. 그룹에 속하는 선택 패키지도 설치하려 면 명령을 실행할 때 선택 패키지를 추가하는 `group_package_types` 구성 파라미터를 설정합니다.

```
[ec2-user ~]$ sudo yum --setopt=group_package_types=mandatory,default,optional groupinstall "Performance Tools"
```

yum install을 사용하여 인터넷에서 다운로드한 RPM 패키지 파일을 설치할 수도 있습니다. 이렇게 하려면 설치 명령에 리포지토리 패키지 이름 대신 RPM 파일의 경로 이름을 덧붙이면 됩니다.

```
[ec2-user ~]$ sudo yum install my-package.rpm
```

소프트웨어 컴파일 준비

인터넷에서 구할 수 있는 방대한 오픈 소스 소프트웨어 중에는 아직 컴파일되지 않은 상태로 패키지 리포지토리에서 다운로드 가능한 것도 있습니다. 또한 이후에 소스 코드로 직접 컴파일해야 하는 소프트웨어 패키지를 검색할 수도 있습니다. 시스템에서 소프트웨어를 컴파일할 수 있으려면 make, gcc, autoconf 등의 몇 가지 개발 도구를 설치해야 합니다.

Important

이 절차는 Amazon Linux에서 사용하기 위한 것입니다. 기타 배포에 대한 자세한 내용은 해당 설명서를 참조하십시오.

소프트웨어 컴파일은 모든 Amazon EC2 인스턴스에 필요한 작업은 아니기 때문에 이러한 도구는 기본적으로 설치되지 않고 "Development Tools"라는 패키지 그룹으로 제공됩니다. yum groupinstall 명령으로 인스턴스에 이 그룹을 손쉽게 추가할 수 있습니다.

```
[ec2-user ~]$ sudo yum groupinstall "Development Tools"
```

<https://github.com/> 및 <http://sourceforge.net/> 등의 웹 사이트에서 소프트웨어 소스 코드 패키지를 tarball이라는 압축된 아카이브 파일로 다운로드할 수 있는 경우가 많습니다. 이러한 tarball의 파일 확장명은 일반적으로 .tar.gz입니다. tar 명령으로 이러한 아카이브의 압축을 풀 수 있습니다.

```
[ec2-user ~]$ tar -xzf software.tar.gz
```

소스 코드 패키지의 압축을 풀고 아카이빙을 해제한 후에는 소스 코드 디렉터리의 README 또는 INSTALL 파일을 참조하여 자세한 소스 코드 컴파일 및 설치 방법을 확인해야 합니다.

Amazon Linux 패키지의 소스 코드를 검색하려면 다음을 수행합니다.

Amazon Web Services에서는 유지 관리되는 패키지의 소스 코드를 제공합니다. get_reference_source 명령으로 설치된 패키지의 소스 코드를 다운로드할 수 있습니다.

- get_reference_source -p ### 명령을 사용하여 ###의 소스 코드를 다운로드합니다. 예를 들어 htop 패키지의 소스 코드를 다운로드하려면 다음 명령을 입력합니다.

```
[ec2-user ~]$ get_reference_source -p htop
Requested package: htop
Found package from local RPM database: htop-1.0.1-2.3.amzn1.x86_64
Corresponding source RPM to found package : htop-1.0.1-2.3.amzn1.src.rpm

Are these parameters correct? Please type 'yes' to continue: yes
Source RPM downloaded to: /usr/src/srpm/debug/htop-1.0.1-2.3.amzn1.src.rpm
```

명령 출력에 소스 RPM의 위치가 나열됩니다. 여기에서는 /usr/src/srpm/debug/htop-1.0.1-2.3.amzn1.src.rpm입니다.

Linux 인스턴스의 사용자 계정 관리

각 Linux 인스턴스 유형은 기본 Linux 시스템 사용자 계정으로 시작됩니다. Amazon Linux의 경우 사용자 이름은 ec2-user입니다. RHEL의 경우 사용자 이름은 ec2-user 또는 root입니다. Ubuntu의 경우 사용자 이름은 ubuntu 또는 root입니다. Centos의 경우 사용자 이름은 centos입니다. Fedora의 경우 사용자 이름은

ec2-user입니다. SUSE의 경우 사용자 이름은 ec2-user 또는 root입니다. ec2-user 및 root를 사용할 수 없는 경우 AMI 공급자에게 문의하십시오.

Note

Linux 시스템 사용자를 AWS Identity and Access Management(IAM) 사용자와 혼동하지 마십시오.
자세한 내용은 IAM 사용 설명서의 [IAM Users and Groups](#)를 참조하십시오.

대부분의 애플리케이션은 기본 사용자 계정만으로 충분하지만, 사용자 계정을 추가하면 각 사용자에게 별도의 파일과 작업 영역을 제공할 수 있습니다. 신규 사용자 계정을 생성하는 방법은 사용이 미숙할 수 있는 여러 사용자에게 ec2-user 계정 액세스를 허용하는 방법보다 보안상 훨씬 안전합니다. 이 계정은 잘못 사용될 경우 시스템에 심각한 손상을 줄 수 있기 때문입니다.

시스템에 사용자를 새로 추가하려면 다음을 수행합니다.

Linux 인스턴스에 사용자를 실제로 추가하려면 두 가지 기본 작업을 수행해야 합니다. 즉, 사용자를 시스템에 추가하고 해당 사용자에게 원격 로그인 수단을 제공해야 합니다.

1. 시스템에 사용자를 새로 추가하려면 adduser 명령과 관련 옵션 및 생성할 사용자의 이름을 입력합니다.

Important

Ubuntu 시스템에 사용자를 추가하는 경우 계정에 암호가 걸리지 않도록 --disabled-password 옵션을 추가해야 합니다.

```
[ec2-user ~]$ sudo adduser newuser
```

이 명령은 시스템에 newuser 계정을 추가하고, /etc/passwd 파일에 항목을 기록하며, newuser 그룹을 생성하고, /home/newuser에 계정의 흄 디렉터리를 생성합니다.

2. 이 계정에 대한 원격 액세스를 제공하려면 newuser 흄 디렉터리에 .ssh 디렉터리를 생성하고 퍼블릭 키를 담은 "authorized_keys"라는 파일을 해당 디렉터리에 생성해야 합니다.
- a. 새로 생성하는 파일이 정확한 소유권을 가질 수 있도록 새 계정으로 전환합니다.

```
[ec2-user ~]$ sudo su - newuser
[newuser ~]$
```

이제 프롬프트에 ec2-user 대신 newuser가 표시되어 shell 세션이 새 계정으로 전환된 것을 알 수 있습니다.

- b. authorized_keys 파일을 넣을 .ssh 디렉터리를 생성합니다.

```
[newuser ~]$ mkdir .ssh
```

- c. .ssh 디렉터리의 파일 권한을 700으로 변경합니다. 이 권한은 파일 소유자만 디렉터리를 읽거나, 쓰거나, 열 수 있다는 의미입니다.

Important

이 단계는 매우 중요합니다. 파일 권한이 정확하지 않으면 SSH를 사용하여 이 계정에 로그인할 수 없습니다.

```
[newuser ~]$ chmod 700 .ssh
```

- d. .ssh 디렉터리에 "authorized_keys"라는 파일을 생성합니다.

```
[newuser ~]$ touch .ssh/authorized_keys
```

- e. authorized_keys 파일의 파일 권한을 600으로 변경합니다. 이 권한은 파일 소유자만 파일을 읽거나 쓸 수 있다는 의미입니다.

Important

이 단계는 매우 중요합니다. 파일 권한이 정확하지 않으면 SSH를 사용하여 이 계정에 로그인할 수 없습니다.

```
[newuser ~]$ chmod 600 .ssh/authorized_keys
```

- f. 선호하는 텍스트 편집기로 `authorized_keys` 파일을 편집하고 키 페어의 퍼블릭 키를 파일에 붙여 넣습니다. 예를 들면 다음과 같습니다.

```
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706V  
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr  
lsLnB1tntcki7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0Fzz  
qaeJAAHco+CY/SWrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221Cb5IMucxXPkX4rWi+z7wB3Rb  
BQoQzd8v7yeb7OzlPnWOyN0qFU0XA246RA8QFYiCNYWI3f05p6KLxEXAMPLE
```

Note

키 페어 생성에 대한 자세한 내용은 [Amazon EC2를 사용해 키 페어 만들기 \(p. 378\)](#)를 참조하십시오. 기존 키 페어에서 퍼블릭 키를 검색하는 방법에 대한 자세한 내용은 [키 페어에 맞는 퍼블릭 키 검색\(Linux\) \(p. 380\)](#)을 참조하십시오.

이제 [Step 2.f \(p. 306\)](#)의 퍼블릭 키와 쌍을 이루는 프라이빗 키를 사용하여 SSH를 통해 인스턴스에서 `newuser` 계정에 로그인할 수 있습니다.

시스템에서 사용자를 제거하려면 다음을 수행합니다.

사용자 계정이 더 이상 필요하지 않은 경우 더 이상 사용할 수 없도록 계정을 제거할 수 있습니다.

- 사용자 계정, 사용자의 홈 디렉터리 및 사용자의 메일 스폴을 삭제하려면 `userdel -r` 명령과 삭제할 사용자 이름을 입력합니다.

```
[ec2-user ~]$ sudo userdel -r olduser
```

Note

사용자의 홈 디렉터리와 메일 스폴을 보존하려면 `-r` 옵션을 생략합니다.

EC2 인스턴스에 대한 프로세서 상태 제어

C 상태는 유휴 상태일 때 코어가 진입하는 절전 수준을 제어합니다. C 상태는 C0(코어가 완전 활성 상태에서 명령을 실행하는 가장 얇은 단계) ~ C6(코어의 전원이 꺼지는 가장 깊은 유휴 단계)의 숫자로 표시됩니다. P 상태는 코어의 성능(CPU 주파수)을 제어합니다. P 상태는 P0(코어가 Intel Turbo Boost Technology를 사용하여 최대 주파수로 증가하는 최고 성능 설정)에서 시작하여 P1(최대 기준 주파수의 P 상태) ~ P15(최저 주파수)의 숫자로 표시됩니다.

다음 인스턴스 유형은 운영 체제에서 프로세서 C 상태 및 P 상태를 제어할 수 있는 기능을 제공합니다.

- c4.8xlarge
- d2.8xlarge
- i3.16xlarge
- m4.10xlarge
- m4.16xlarge
- p2.16xlarge

- r4.8xlarge
- r4.16xlarge
- x1.16xlarge
- x1.32xlarge

프로세서의 성능 일관성을 향상하고 지연 시간을 줄이거나 특정 워크로드에 대해 인스턴스를 조정하기 위해 C 상태 또는 P 상태 설정을 변경할 수 있습니다. 기본 C 상태 및 P 상태는 대부분의 최고 성능을 제공하도록 설정되어 있고 대부분의 워크로드에 적합합니다. 그러나 애플리케이션에서 단일 또는 이중 코어의 높은 주파수에서 지연 시간을 줄이는 것이 비용상 이익이 되거나 Turbo Boost 버스트 주파수에 비해 낮은 주파수에서 일관된 성능을 제공하는 것이 이익이 되는 경우 이러한 인스턴스에서 사용 가능한 C 상태 또는 P 상태 설정을 시험해보는 것을 고려하십시오.

다음 섹션은 다른 프로세서 상태 구성 및 구성에 따른 영향을 확인하는 방법에 대해 설명합니다. 이러한 절차는 Amazon Linux용으로 작성 및 적용되었지만 Linux 커널 3.9 버전 이상의 다른 Linux 배포판에서도 적용될 수 있습니다. Linux 배포판 및 프로세서 상태 제어에 대한 자세한 내용은 시스템별 설명서를 참조하십시오.

Note

이 섹션의 예제에서는 turbostat 유틸리티(Amazon Linux에서 기본 제공됨)를 사용하여 프로세서 주파수 및 C 상태 정보를 표시하고 stress 명령(sudo yum install -y stress를 실행하여 설치 가능)을 사용하여 워크로드가 시뮬레이션됩니다.

목차

- 최고 Turbo Boost 주파수에서 최상의 성능 (p. 307)
- C 상태 심화 제한을 통한 고성능 및 저 지연 시간 (p. 308)
- 변동성이 가장 낮은 기준 성능 (p. 309)

최고 Turbo Boost 주파수에서 최상의 성능

이는 Amazon Linux AMI의 기본 프로세서 상태 제어 구성이고 대부분의 워크로드에 권장됩니다. 이 구성은 변동성이 낮은 최고 성능을 제공합니다. 비활성 코어가 더 깊은 절전 상태로 진입하도록 함으로써 필요한 가용 온도를 제공하여 단일 또는 듀얼 코어 프로세서가 최대 Turbo Boost 성능을 실현할 수 있습니다.

다음 예제는 적극적으로 작업을 수행하는 코어 2개가 있는 c4.8xlarge 인스턴스가 최대 프로세서 Turbo Boost 주파수에 도달한 것을 보여줍니다.

```
[ec2-user ~]$ sudo turbostat stress -c 2 -t 10
stress: info: [30680] dispatching hogs: 2 cpu, 0 io, 0 vm, 0 hdd
stress: info: [30680] successful run completed in 10s
pk cor CPU %c0 GHz TSC SMI %c1 %c3 %c6 %c7 %pc2 %pc3 %pc6 %pc7
Pkg_W RAM_W PKG_% RAM_%
      5.54 3.44 2.90  0  9.18  0.00  85.28  0.00  0.00  0.00  0.00  0.00
94.04 32.70 54.18  0.00
0   0   0   0.12 3.26 2.90  0  3.61  0.00  96.27  0.00  0.00  0.00
48.12 18.88 26.02  0.00
0   0   18   0.12 3.26 2.90  0  3.61
0   1   1   0.12 3.26 2.90  0  4.11  0.00  95.77  0.00
0   1   19   0.13 3.27 2.90  0  4.11
0   2   2   0.13 3.28 2.90  0  4.45  0.00  95.42  0.00
0   2   20   0.11 3.27 2.90  0  4.47
0   3   3   0.05 3.42 2.90  0  99.91  0.00  0.05  0.00
0   3   21   97.84 3.45 2.90  0  2.11
...
1   1   10   0.06 3.33 2.90  0  99.88  0.01  0.06  0.00
1   1   28   97.61 3.44 2.90  0  2.32
...
10.0002556 sec
```

이 예에서는 다른 코어가 c6 절전 상태에 진입하여 전력을 절감하고 작업 코어에 전력과 가용 온도를 제공하기 때문에 vCPU 21 및 28은 최대 Turbo Boost 주파수로 실행될 수 있습니다. vCPUs 3 및 10(각각은 vCPUs 21 및 28과 프로세서 코어를 공유)은 c1 상태에서 명령을 대기합니다.

다음 예에서 18개 코어 모두는 적극적으로 작업을 수행하여 최대 Turbo Boost의 가용 온도가 없지만 3.2GHz의 "전체 코어 Turbo Boost" 속도에서 모두 실행됩니다.

```
[ec2-user ~]$ sudo turbostat stress -c 36 -t 10
stress: info: [30685] dispatching hogs: 36 cpu, 0 io, 0 vm, 0 hdd
stress: info: [30685] successful run completed in 10s
pk cor CPU %c0 GHz TSC SMI %c1 %c3 %c6 %c7 %pc2 %pc3 %pc6 %pc7
Pkg_W RAM_W PKG_% RAM_%
 99.27 3.20 2.90 0 0.26 0.00 0.47 0.00 0.00 0.00 0.00 0.00 0.00
228.59 31.33 199.26 0.00
0 0 0 99.08 3.20 2.90 0 0.27 0.01 0.64 0.00 0.00 0.00 0.00
114.69 18.55 99.32 0.00
0 0 18 98.74 3.20 2.90 0 0.62
0 1 1 99.14 3.20 2.90 0 0.09 0.00 0.76 0.00
0 1 19 98.75 3.20 2.90 0 0.49
0 2 2 99.07 3.20 2.90 0 0.10 0.02 0.81 0.00
0 2 20 98.73 3.20 2.90 0 0.44
0 3 3 99.02 3.20 2.90 0 0.24 0.00 0.74 0.00
0 3 21 99.13 3.20 2.90 0 0.13
0 4 4 99.26 3.20 2.90 0 0.09 0.00 0.65 0.00
0 4 22 98.68 3.20 2.90 0 0.67
0 5 5 99.19 3.20 2.90 0 0.08 0.00 0.73 0.00
0 5 23 98.58 3.20 2.90 0 0.69
0 6 6 99.01 3.20 2.90 0 0.11 0.00 0.89 0.00
0 6 24 98.72 3.20 2.90 0 0.39
...
...
```

C 상태 심화 제한을 통한 고성능 및 저 지연 시간

C 상태는 비활성 상태일 때 코어가 진입하는 절전 수준을 제어합니다. C 상태를 제어하여 시스템의 지연 시간과 성능 조합을 미세 조정할 수 있습니다. 코어가 절전 상태에 진입하기 위해서는 시간이 소요되고 비록 한 코어가 절전 중이면 다른 코어는 더 많은 가용 온도로 더 높은 주파수로 동작할 수 있지만 절전 중인 코어가 다시 정상 상태로 돌아와 작업을 수행하는 데는 시간이 소요됩니다. 예를 들어, 네트워크 패킷 인터럽트를 처리하는 코어가 절전 상태인 경우 인터럽트 상태를 해결하는 것이 지연될 수 있습니다. 그 경우 C 상태가 심화되지 않도록 시스템을 구성하여 프로세서 반응 지연 시간을 줄일 수 있지만 그 대가로 Turbo Boost를 위해 다른 코어에서 사용할 수 있는 가용성이 줄어듭니다.

절전 상태가 심화되지 않도록 설정하는 일반적인 방법에서는 Redis 데이터베이스 애플리케이션이 사용되고 이 경우 최대한 빠른 쿼리 응답 시간이 제공되도록 시스템 메모리에 데이터베이스가 저장됩니다.

Amazon Linux에서 절전 상태 심화를 제한하려면

1. 편집기에서 /boot/grub/grub.conf 파일을 엽니다.

```
[ec2-user ~]$ sudo vim /boot/grub/grub.conf
```

2. 처음 항목의 kernel 라인을 수정하고 intel_idle.max_cstate=1 옵션을 추가하여 c1을 유휴 코어의 최대 유휴 C 상태로 설정합니다.

```
# created by imagebuilder
default=0
timeout=1
hiddenmenu

title Amazon Linux 2014.09 (3.14.26-24.46.amzn1.x86_64)
root (hd0,0)
```

```
kernel /boot/vmlinuz-3.14.26-24.46.amzn1.x86_64 root=LABEL=/ console=ttyS0
intel_idle.max_cstate=1
initrd /boot/initramfs-3.14.26-24.46.amzn1.x86_64.img
```

3. 파일을 저장하고 편집기를 종료합니다.
4. 인스턴스를 재부팅하여 새 커널 옵션을 활성화합니다.

```
[ec2-user ~]$ sudo reboot
```

다음 예제는 "전체 코어 Turbo Boost" 코어 주파수에서 적극적으로 작업을 수행하는 코어 2개가 있는 c4.8xlarge 인스턴스를 보여줍니다.

```
[ec2-user ~]$ sudo turbostat stress -c 2 -t 10
stress: info: [5322] dispatching hogs: 2 cpu, 0 io, 0 vm, 0 hdd
stress: info: [5322] successful run completed in 10s
pk cor CPU %c0 GHz TSC SMI %c1 %c3 %c6 %c7 %pc2 %pc3 %pc6 %pc7
Pkg_W RAM_W PKG_% RAM_%
      5.56 3.20 2.90   0 94.44  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
131.90 31.11 199.47 0.00
0   0   0   0.03 2.08 2.90   0 99.97  0.00  0.00  0.00  0.00  0.00  0.00
67.23 17.11 99.76 0.00
0   0   18  0.01 1.93 2.90   0 99.99
0   1   1   0.02 1.96 2.90   0 99.98  0.00  0.00  0.00
0   1   19  99.70 3.20 2.90   0 0.30
...
1   1   10  0.02 1.97 2.90   0 99.98  0.00  0.00  0.00
1   1   28  99.67 3.20 2.90   0 0.33
1   2   11  0.04 2.63 2.90   0 99.96  0.00  0.00  0.00
1   2   29  0.02 2.11 2.90   0 99.98
...
```

이 예에서 vCPUs 19 및 28 코어는 3.2GHz에서 동작하고 다른 코어는 c1 C 상태에서 명령을 대기합니다. 비록 작업 중인 코어는 최대 Turbo Boost 주파수에 도달할 수 없지만 비활성 코어는 가장 깊은 c6 C 상태에 있을 때보다 훨씬 빠르게 새 요청에 응답할 수 있습니다.

변동성이 가장 낮은 기준 성능

P 상태를 조정하여 프로세서 주파수의 변동성을 줄일 수 있습니다. P 상태는 코어의 성능(CPU 주파수)을 제어합니다. 대부분의 워크로드는 P0에서 더 좋은 성능을 발휘하지만 그 경우 Turbo Boost가 필요합니다. 그러나 Turbo Boost 주파수가 사용되는 경우 발생할 수 있는 성능 버스트보다 일관적인 성능을 갖도록 시스템을 미세 조정하는 것이 필요할 때가 있습니다.

인텔 Advanced Vector Extensions(AVX 또는 AVX2) 워크로드는 낮은 주파수에서 좋은 성능을 보이고 AVX 명령은 더 많은 전력을 사용할 수 있습니다. Turbo Boost를 비활성화하여 낮은 주파수에서 프로세서를 실행하면 사용 전력을 줄이고 스피드를 좀 더 일관성 있게 유지할 수 있습니다. 인스턴스 구성 최적화 및 AVX 워크로드에 대한 자세한 내용은 <http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/performance-xeon-e5-v3-advanced-vector-extensions-paper.pdf> 섹션을 참조하십시오.

이 섹션은 절전 상태가 심화되는 것을 제한하고 Turbo Boost(P1 P 상태 요청)를 비활성화하여 이러한 워크로드 유형에 짧은 지연 시간과 낮은 프로세서 속도 변동성을 제공하는 방법에 대해 설명합니다.

Amazon Linux에서 절전 상태 심화를 제한하고 Turbo Boost를 비활성화하려면

1. 편집기에서 /boot/grub/grub.conf 파일을 엽니다.

```
[ec2-user ~]$ sudo vim /boot/grub/grub.conf
```

2. 처음 항목의 kernel 라인을 수정하고 intel_idle.max_cstate=1 옵션을 추가하여 c1을 유휴 코어의 최대 유휴 C 상태로 설정합니다.

```
# created by imagebuilder
default=0
timeout=1
hiddenmenu

title Amazon Linux 2014.09 (3.14.26-24.46.amzn1.x86_64)
root (hd0,0)
kernel /boot/vmlinuz-3.14.26-24.46.amzn1.x86_64 root=LABEL=/ console=ttyS0
intel_idle.max_cstate=1
initrd /boot/initramfs-3.14.26-24.46.amzn1.x86_64.img
```

3. 파일을 저장하고 편집기를 종료합니다.
4. 인스턴스를 재부팅하여 새 커널 옵션을 활성화합니다.

```
[ec2-user ~]$ sudo reboot
```

5. P1 P 상태가 제공하는 낮은 프로세서 속도 변동성이 필요한 경우 다음 명령을 사용하여 Turbo Boost를 비활성화합니다.

```
[ec2-user ~]$ sudo sh -c "echo 1 > /sys/devices/system/cpu/intel_pstate/no_turbo"
```

6. 워크로드가 종료되면 다음 명령으로 Turbo Boost를 다시 활성화할 수 있습니다.

```
[ec2-user ~]$ sudo sh -c "echo 0 > /sys/devices/system/cpu/intel_pstate/no_turbo"
```

다음 예제는 Turbo Boost 없이 기준 코어 주파수에서 적극적으로 작업을 수행하는 vCPU 2개가 있는 c4.8xlarge 인스턴스를 보여줍니다.

```
[ec2-user ~]$ sudo turbostat stress -c 2 -t 10
stress: info: [5389] dispatching hogs: 2 cpu, 0 io, 0 vm, 0 hdd
stress: info: [5389] successful run completed in 10s
pk cor CPU %c0 GHz TSC SMI %c1 %c3 %c6 %c7 %pc2 %pc3 %pc6 %pc7
Pkg_W RAM_W PKG_% RAM_%
      5.59 2.90 2.90   0 94.41  0.00  0.00  0.00  0.00  0.00  0.00  0.00
128.48 33.54 200.00  0.00
0 0 0 0.04 2.90 2.90   0 99.96  0.00  0.00  0.00  0.00  0.00
65.33 19.02 100.00  0.00
0 0 18 0.04 2.90 2.90   0 99.96
0 1 1 0.05 2.90 2.90   0 99.95  0.00  0.00  0.00
0 1 19 0.04 2.90 2.90   0 99.96
0 2 2 0.04 2.90 2.90   0 99.96  0.00  0.00  0.00
0 2 20 0.04 2.90 2.90   0 99.96
0 3 3 0.05 2.90 2.90   0 99.95  0.00  0.00  0.00
0 3 21 99.95 2.90 2.90   0 0.05
...
1 1 28 99.92 2.90 2.90   0 0.08
1 2 11 0.06 2.90 2.90   0 99.94  0.00  0.00  0.00
1 2 29 0.05 2.90 2.90   0 99.95
```

vCPUs 21 및 28용 코어는 2.9GHz의 기준 프로세서 속도에서 적극적으로 작업을 수행하고 모든 비활성 코어 또한 c1 C 상태에서 기준 속도로 동작하여 명령을 수락할 수 있습니다.

Linux 인스턴스의 시간 설정

많은 서버 작업과 프로세스에서 일관되고 정확한 시간 참조가 중요합니다. 대부분의 시스템 로그에는 문제가 발생한 시간과 이벤트가 발생한 순서를 파악하는 데 사용할 수 있는 타임스탬프가 포함되어 있습니다. AWS

CLI 또는 AWS SDK를 사용하여 인스턴스에서 요청하는 경우 이러한 도구가 사용자를 대신하여 요청에 서명합니다. 인스턴스의 날짜와 시간이 잘못 설정되어 있으면 서명 날짜가 요청 날짜와 일치하지 않아 AWS가 해당 요청을 거부할 수 있습니다. Amazon Linux 인스턴스에는 기본적으로 NTP(Network Time Protocol)가 구성되며, 시스템 시간은 인터넷에 있는 퍼블릭 서버의 로드 밸런싱 풀과 동기화되고 UTC 표준 시간대로 설정됩니다. NTP에 대한 자세한 내용은 <http://www.ntp.org/>를 참조하십시오.

작업

- [표준 시간대 변경 \(p. 311\)](#)
- [NTP 구성 \(p. 312\)](#)

Important

이 절차는 Amazon Linux에서 사용하기 위한 것입니다. 기타 배포에 대한 자세한 내용은 해당 설명서를 참조하십시오.

표준 시간대 변경

기본적으로 Amazon Linux 인스턴스는 UTC(협정 세계시) 표준 시간대로 설정되지만, 인스턴스의 시간을 현지 시간 또는 네트워크의 다른 표준 시간대로 변경해야 할 수도 있습니다.

인스턴스의 표준 시간대를 변경하려면 다음을 수행합니다.

1. 인스턴스에서 사용할 표준 시간대를 식별합니다. `/usr/share/zoneinfo` 디렉터리에는 표준 시간대 데잍 파일이 계층 구조로 들어 있습니다. 해당 위치의 디렉터리 구조를 탐색하여 원하는 표준 시간대의 파일을 찾습니다.

```
[ec2-user ~]$ ls /usr/share/zoneinfo
Africa      Chile     GB          Indian       Mideast    posixrules   US
America    CST6CDT  GB-Eire    Iran         MST        PRC          UTC
Antarctica Cuba      GMT        iso3166.tab MST7MDT   PST8PDT    WET
Arctic     EET       GMT0       Israel      Navajo    right        W-SU
...
...
```

이 위치의 일부 항목(예: America)은 디렉터리이며, 이러한 디렉터리에는 도시별 표준 시간대 파일이 들어 있습니다. 인스턴스에 사용할 도시 또는 해당 표준 시간대에 속하는 도시를 찾습니다. 이 예제에서는 로스앤젤레스의 표준 시간대 파일인 `/usr/share/zoneinfo/America/Los_Angeles`를 사용할 수 있습니다.

2. `/etc/sysconfig/clock` 파일을 새 표준 시간대로 업데이트합니다.
 - a. vim, nano 등의 선호하는 텍스트 편집기로 `/etc/sysconfig/clock` 파일을 엽니다. `/etc/sysconfig/clock`은 root가 소유하므로 sudo와 함께 편집기 명령을 사용해야 합니다.
 - b. ZONE 항목을 찾아서 표준 시간대 파일로 변경합니다. 경로에서 `/usr/share/zoneinfo` 부분은 생략하십시오. 예를 들어 로스앤젤레스 표준 시간대로 변경하려면 ZONE 항목을 다음과 같이 변경합니다.

```
ZONE="America/Los_Angeles"
```

Note

`UTC=true` 항목을 다른 값으로 변경하지 마십시오. 이 항목은 하드웨어 클록에 대한 것으로, 인스턴스에 대해 다른 표준 시간대를 설정할 때 따로 조정할 필요가 없습니다.

- c. 파일을 저장하고 텍스트 편집기를 종료합니다.
3. 인스턴스가 현지 시간 정보를 참조할 때 표준 시간대 파일을 찾을 수 있도록 `/etc/localtime`과 표준 시간대 파일 사이에 심볼 링크를 생성합니다.

```
[ec2-user ~]$ sudo ln -sf /usr/share/zoneinfo/America/Los_Angeles /etc/localtime
```

4. 시스템을 재부팅하여 모든 서비스와 애플리케이션에 새로운 표준 시간대 정보를 적용합니다.

```
[ec2-user ~]$ sudo reboot
```

NTP 구성

Amazon Linux 인스턴스에는 기본적으로 NTP(Network Time Protocol)가 구성되지만, 스탠다드 NTP 구성 을 사용하려면 인스턴스에서 인터넷에 액세스할 수 있어야 합니다. 또한 인스턴스의 보안 그룹 규칙은 포트 123(NTP)에서 아웃바운드 UDP 트래픽을 허용해야 하고, 네트워크 ACL 규칙은 포트 123에서 인바운드와 아웃바운드 UDP 트래픽을 모두 허용해야 합니다. 이 섹션의 절차에서는 기본 NTP 구성이 올바르게 작동하는지 확인하는 방법을 보여 줍니다. 인스턴스에서 인터넷에 액세스할 수 없는 경우 정확한 시간을 유지하려 면 프라이빗 네트워크에서 다른 서버를 쿼리하도록 NTP를 구성해야 합니다.

NTP가 제대로 작동하는지 확인하려면

1. ntpstat 명령을 사용하여 인스턴스에서 NTP 서비스의 상태를 확인합니다.

```
[ec2-user ~]$ ntpstat
```

아래와 비슷하게 출력되는 경우 인스턴스에서 NTP가 제대로 작동하고 있는 것입니다.

```
synchronised to NTP server (12.34.56.78) at stratum 3
      time correct to within 399 ms
      polling server every 64 s
```

출력에 "unsynchronised"가 표시되는 경우 1분 정도 기다렸다가 다시 시도하십시오. 동기화를 처음으로 완료하는 데 1분 정도 걸릴 수 있습니다.

출력에 "Unable to talk to NTP daemon. Is it running?"이 표시되는 경우 NTP 서비스를 시작하고 부팅 시 자동으로 시작하도록 NTP를 활성화해야 할 수 있습니다.

2. (선택 사항) ntpq -p 명령을 사용하여 NTP 서버에 알려진 피어 목록과 상태 요약을 확인할 수 있습니다.

```
[ec2-user ~]$ ntpq -p
      remote          refid      st t when poll reach   delay    offset  jitter
=====+
+lttlemman.deekay  204.9.54.119    2 u   15  128  377    88.649    5.946   6.876
-bittorrent.tomh  91.189.94.4     3 u   133  128  377   182.673    8.001   1.278
*ntp3.junkemailf  216.218.254.202  2 u    68  128  377    29.377    4.726  11.887
+tesla.selinc.co 149.20.64.28    2 u    31  128  377    28.586   -1.215   1.435
```

이 명령의 출력에 활동 내역이 없는 경우 보안 그룹, 네트워크 ACL 또는 NTP 포트에 대한 액세스를 차단하는 방화벽이 있는지 확인합니다.

NTP를 시작하고 활성화하려면

1. 다음 명령을 사용하여 NTP 서비스를 시작합니다.

```
[ec2-user ~]$ sudo service ntpd start
Starting ntpd:                                         [ OK ]
```

2. chkconfig 명령을 사용하여 부팅 시 시작하도록 NTP를 활성화합니다.

```
[ec2-user ~]$ sudo chkconfig ntpd on
```

3. 다음 명령을 사용하여 NTP가 활성화되었는지 확인합니다.

```
[ec2-user ~]$ sudo chkconfig --list ntpd
ntpda          0:off 1:off 2:on 3:on 4:on 5:on 6:off
```

여기에서 ntpd는 실행 수준 2, 3, 4, 5에서 on인 올바른 상태입니다.

NTP 서버를 변경하려면

스탠다드 NTP 서버를 사용하지 않도록 하거나 인터넷에 액세스할 수 없는 인스턴스에 대해 프라이빗 네트워크 내의 자체 NTP 서버를 사용해야 할 수 있습니다.

1. vim, nano 등의 텍스트 편집기에서 /etc/ntp.conf 파일을 업니다. /etc/ntp.conf는 root가 소유하므로 sudo와 함께 편집기 명령을 사용해야 합니다.
2. NTP 구성에 대해 폴링할 서버를 정의하는 server 섹션을 찾습니다.

```
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
server 0.amazon.pool.ntp.org iburst
server 1.amazon.pool.ntp.org iburst
server 2.amazon.pool.ntp.org iburst
server 3.amazon.pool.ntp.org iburst
```

Note

.amazon.pool.ntp.org DNS 레코드를 통해 AWS에서 NTP 트래픽의 로드 밸런스를 유지할 수 있습니다. 그러나 이러한 서버는 pool.ntp.org 프로젝트에서 퍼블릭 NTP 서버이고 AWS에서 소유하거나 관리하지 않습니다. 이러한 서버는 지리적으로 인스턴스에 가깝거나 AWS 네트워크 내에 위치하지 않을 수도 있습니다. 자세한 내용은 <http://www.pool.ntp.org/en/>을 참조하십시오.

3. 서버 정의의 시작 부분에 "#" 문자를 추가하여 사용하지 않을 서버를 주석으로 처리합니다.

```
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
#server 0.amazon.pool.ntp.org iburst
#server 1.amazon.pool.ntp.org iburst
#server 2.amazon.pool.ntp.org iburst
#server 3.amazon.pool.ntp.org iburst
```

4. 시간 동기화를 위해 폴링할 각 서버에 대한 항목을 추가합니다. 이 항목에 대한 DNS 이름 또는 점으로 구분된 네 부분의 숫자로 된 IP 주소(예: 10.0.0.254)를 사용할 수 있습니다.

```
server my-ntp-server.my-domain.com iburst
```

5. 새 서버를 선택하려면 NTP 서비스를 다시 시작합니다.

```
[ec2-user ~]$ sudo service ntpd start
Starting ntpd: [ OK ]
```

6. 새 설정이 적용되고 NTP가 작동하는지 확인합니다.

```
[ec2-user ~]$ ntpstat
synchronised to NTP server (64.246.132.14) at stratum 2
```

time correct to within 99 ms

Linux 인스턴스의 호스트 이름 변경

인스턴스를 시작하면 인스턴스에 호스트 이름(내부 프라이빗 IPv4 주소)이 지정됩니다. 일반적인 Amazon EC2 프라이빗 DNS 이름은 ip-12-34-56-78.us-west-2.compute.internal과 같이 내부 도메인, 서비스(이 경우 compute), 리전 및 프라이빗 IPv4 주소 형태로 구성됩니다. 인스턴스에 로그인하면 shell 프롬프트에 이 호스트 이름의 일부(예: ip-12-34-56-78)가 표시됩니다. 탄력적 IP 주소를 사용하지 않는 경우 Amazon EC2 인스턴스를 종지하고 다시 시작할 때마다 퍼블릭 IPv4 주소, 퍼블릭 DNS 이름, 시스템 호스트 이름 및 shell 프롬프트가 바뀝니다. EC2-Classic에서 시작된 인스턴스도 종지 후 다시 시작하면 새로운 프라이빗 IPv4 주소, 프라이빗 DNS 호스트 이름 및 시스템 호스트 이름을 지정받지만, VPC에서 시작된 인스턴스는 그렇지 않습니다.

Important

이 절차는 Amazon Linux에서 사용하기 위한 것입니다. 기타 배포에 대한 자세한 내용은 해당 설명서를 참조하십시오.

시스템 호스트 이름 변경

인스턴스의 IP 주소에 퍼블릭 DNS 이름을 등록한 경우(예: webserver.mydomain.com) 인스턴스가 자신이 해당 도메인에 속함을 인식하도록 시스템 호스트 이름을 설정할 수 있습니다. 또한 이렇게 하면 AWS에서 지정한 ip-12-34-56-78과 같은 호스트 이름 대신 이 이름의 첫 부분이 shell 프롬프트에 표시됩니다. 퍼블릭 DNS 이름을 등록하지 않은 경우에도 호스트 이름을 변경할 수 있지만 절차가 약간 다릅니다.

시스템 호스트 이름을 퍼블릭 DNS 이름으로 변경하려면 다음을 수행합니다.

이미 퍼블릭 DNS 이름을 등록한 경우 이 절차를 따릅니다.

1. 인스턴스에서 선호하는 텍스트 편집기로 /etc/sysconfig/network 구성 파일을 열고 HOSTNAME 항목을 변경하여 webserver.mydomain.com과 같이 정규화된 도메인 이름을 반영합니다.

HOSTNAME=webserver.mydomain.com

2. 인스턴스를 재부팅하여 새 호스트 이름을 적용합니다.

[ec2-user ~]\$ sudo reboot

또는 Amazon EC2 콘솔을 사용하여 재부팅할 수 있습니다([Instances] 페이지에서 [Actions], [Instance State], [Reboot] 선택).

3. 인스턴스에 로그인하고 호스트 이름이 업데이트되었는지 확인합니다. 프롬프트에 새 호스트 이름이 첫 번째 "."까지 표시되어야 하고, hostname 명령이 정규화된 도메인 이름을 표시해야 합니다.

[ec2-user@webserver ~]\$ hostname
webserver.mydomain.com

퍼블릭 DNS 이름 없이 시스템 호스트 이름을 변경하려면 다음을 수행합니다.

1. 선호하는 텍스트 편집기로 /etc/sysconfig/network 구성 파일을 열고 HOSTNAME 항목을 변경하여 webserver와 같이 원하는 호스트 이름을 반영합니다.

HOSTNAME=webserver.localdomain

2. 선호하는 텍스트 편집기로 /etc/hosts 파일을 열고 127.0.0.1로 시작되는 항목을 아래 예제와 일치하도록 변경합니다. 원하는 호스트 이름을 대신 입력하면 됩니다.

```
127.0.0.1 webserver.localdomain webserver localhost4 localhost4.localdomain4
```

3. 인스턴스를 재부팅하여 새 호스트 이름을 적용합니다.

```
[ec2-user ~]$ sudo reboot
```

또는 Amazon EC2 콘솔을 사용하여 재부팅할 수 있습니다([Instances] 페이지에서 [Actions], [Instance State], [Reboot] 선택).

4. 인스턴스에 로그인하고 호스트 이름이 업데이트되었는지 확인합니다. 프롬프트에 새 호스트 이름이 첫 번째 ":"까지 표시되어야 하고, hostname 명령이 정규화된 도메인 이름을 표시해야 합니다.

```
[ec2-user@webserver ~]$ hostname  
webserver.localdomain
```

호스트 이름에 영향을 주지 않고 shell 프롬프트 변경

인스턴스의 호스트 이름을 수정하지 않으면서 AWS에서 제공한 프라이빗 이름(예: ip-12-34-56-78)보다 더 유용한 시스템 이름(예: webserver)을 표시하려는 경우 shell 프롬프트 구성 파일을 편집하여 호스트 이름 대신 시스템 별칭을 표시할 수 있습니다.

shell 프롬프트를 호스트 별칭으로 변경하려면 다음을 수행합니다.

1. /etc/profile.d에 NICKNAME이라는 환경 변수를 shell 프롬프트로 사용할 값으로 설정하는 파일을 생성합니다. 예를 들어 시스템 별칭을 webserver라고 설정하려면 다음 명령을 실행합니다.

```
[ec2-user ~]$ sudo sh -c 'echo "export NICKNAME=webserver" > /etc/profile.d/prompt.sh'
```

2. vim, nano 등의 선호하는 텍스트 편집기로 /etc/bashrc 파일을 열니다. /etc/bashrc는 root가 소유하므로 sudo와 함께 편집기 명령을 사용해야 합니다.
3. 파일을 편집하여 호스트 이름 대신 별칭을 표시하도록 shell 프롬프트 변수(\$PS1)를 변경합니다. /etc/bashrc에서 shell 프롬프트를 설정하는 다음 줄을 찾습니다. 아래에서는 참조를 위해 위아래 몇 줄을 함께 표시했으며, ["\$PS1"]로 시작되는 줄을 찾으면 됩니다.

```
# Turn on checkwinsize  
shopt -s checkwinsize  
[ "$PS1" = "\s-\v\\\$ " ] && PS1="\u@\h \w]\\$\n"  
# You might want to have e.g. tty in prompt (e.g. more virtual machines)  
# and console windows
```

해당 줄에서 \h(hostname을 나타내는 기호)를 NICKNAME 변수 값으로 변경합니다.

```
# Turn on checkwinsize  
shopt -s checkwinsize  
[ "$PS1" = "\s-\v\\\$ " ] && PS1="\u@\$NICKNAME \w]\\$\n"  
# You might want to have e.g. tty in prompt (e.g. more virtual machines)  
# and console windows
```

4. (선택 사항) shell 창의 제목을 새 별칭으로 설정하려면 다음 단계를 완료합니다.

- a. /etc/sysconfig/bash-prompt-xterm이라는 파일을 생성합니다.

```
[ec2-user ~]$ sudo touch /etc/sysconfig/bash-prompt-xterm
```

- b. 다음 명령으로 파일을 실행 가능하도록 만듭니다.

```
[ec2-user ~]$ sudo chmod +x /etc/sysconfig/bash-prompt-xterm
```

- c. vim, nano 등의 선호하는 텍스트 편집기로 /etc/sysconfig/bash-prompt-xterm 파일을 업니다. /etc/sysconfig/bash-prompt-xterm은 root가 소유 하므로 sudo와 함께 편집기 명령을 사용해야 합니다.
- d. 파일에 다음 줄을 추가합니다.

```
echo -ne "\033]0;${USER}@${NICKNAME}: ${PWD/#$HOME/~}\007"
```

5. 로그아웃하고 다시 로그인하여 새 별칭 값을 적용합니다.

다른 Linux 배포판에서 호스트 이름 변경

위 절차는 Amazon Linux 전용입니다. 다른 Linux 배포판에 대한 자세한 내용은 해당 설명서와 다음 항목을 참조하십시오.

- [How do I assign a static hostname to a private Amazon EC2 instance running RHEL 7 or Centos 7?](#)
- [How do I assign a static hostname to a private Amazon EC2 instance running SuSe Linux?](#)
- [How do I assign a static hostname to a private Amazon EC2 instance running Ubuntu Linux?](#)

Your Linux 인스턴스에 동적 DNS 설정

EC2 인스턴스를 시작하면 인터넷에서 인스턴스에 접속하는데 사용할 수 있는 퍼블릭 IP 주소와 퍼블릭 DNS(도메인 이름 시스템) 이름이 지정됩니다. Amazon Web Services 도메인에는 수없이 많은 호스트가 있으므로 퍼블릭 이름이 상당히 길어야 각 이름의 고유성을 유지할 수 있습니다. 일반적인 Amazon EC2 퍼블릭 DNS 이름은 ec2-12-34-56-78.us-west-2.compute.amazonaws.com과 같이 Amazon Web Services 도메인, 서비스(이 경우 compute), 리전 및 퍼블릭 IP 주소 형태로 구성됩니다.

동적 DNS 서비스는 도메인 영역 내에서 기억하기 쉽고 호스트의 사용 사례에 더욱 적합한 맞춤형 DNS 호스트 이름을 제공하며, 경우에 따라 이러한 서비스를 무료로 이용할 수도 있습니다. Amazon EC2에 동적 DNS 공급자를 사용하고 인스턴스가 시작될 때마다 퍼블릭 DNS 이름에 연결된 IP 주소를 업데이트하도록 인스턴스를 구성할 수 있습니다. 매우 다양한 공급자 중에서 선택할 수 있으며, 적합한 공급자를 선택하는 구체적인 방법 및 이름을 등록하는 방법은 본 안내서의 범위를 벗어납니다.

Important

이 절차는 Amazon Linux에서 사용하기 위한 것입니다. 기타 배포에 대한 자세한 내용은 해당 설명서를 참조하십시오.

Amazon EC2에 동적 DNS를 사용하려면 다음을 수행합니다.

1. 동적 DNS 서비스 공급자의 서비스에 가입하고 퍼블릭 DNS 이름을 등록합니다. 이 절차에서는 [noip.com/free](#)의 무료 서비스를 예제로 사용합니다.
2. 동적 DNS 업데이트 클라이언트를 구성합니다. 동적 DNS 서비스 공급자의 서비스에 가입하고 퍼블릭 DNS 이름을 등록했으면 DNS 이름에 인스턴스의 IP 주소를 가리킵니다. 공급자에 따라([noip.com](#) 포함) 공급자 웹 사이트의 계정 페이지에서 수동으로 입력하거나 소프트웨어 업데이트 클라이언트를 지원합니다. 업데이트 클라이언트가 EC2 인스턴스에서 실행되고 있다면 종료 및 재시작 후와 같이 IP 주소가 바뀔 때마다 동적 DNS 레코드가 업데이트됩니다. 이 예제에서는 [noip.com](#)에서 제공하는 서비스와 연동되는 noip2 클라이언트를 설치합니다.
 - a. EPEL(Extra Packages for Enterprise Linux) 리포지토리를 활성화하여 noip2 클라이언트에 액세스합니다.

Note

Amazon Linux 인스턴스에서는 EPEL 리포지토리에 대한 GPG 키와 리포지토리 정보가 기본적으로 설치되지만, Red Hat 및 CentOS 인스턴스의 경우에는 먼저 `epel-release` 패키지를 설치해야 EPEL 리포지토리를 활성화할 수 있습니다. 자세한 내용을 확인하고 이 패키지의 최신 버전을 다운로드하려면 <https://fedoraproject.org/wiki/EPEL>을 참조하십시오.

```
[ec2-user ~]$ sudo yum-config-manager --enable epel
```

- b. `noip` 패키지를 설치합니다.

```
[ec2-user ~]$ sudo yum install -y noip
```

- c. `noip2` 구성 파일을 생성합니다. 요청에 따라 로그인 및 암호 정보를 입력하고 후속 질문에 답하여 클라이언트를 구성합니다.

```
[ec2-user ~]$ sudo noip2 -C
```

3. `chkconfig` 명령으로 `noip` 서비스를 활성화합니다.

```
[ec2-user ~]$ sudo chkconfig noip on
```

`chkconfig --list` 명령으로 서비스 활성화 여부를 확인할 수 있습니다.

```
[ec2-user ~]$ chkconfig --list noip
noip           0:off 1:off 2:on 3:on 4:on 5:on 6:off
```

여기에서 `noip`는 실행 레벨 2, 3, 4, 5에서 `on`인 올바른 상태입니다. 이제 부팅 시마다 업데이트 클라이언트가 시작되고 퍼블릭 DNS 레코드를 업데이트하여 인스턴스의 IP 주소를 가리킵니다.

4. `noip` 서비스를 시작합니다.

```
[ec2-user ~]$ sudo service noip start
Starting noip2:                                         [ OK ]
```

이 명령은 클라이언트를 시작합니다. 클라이언트는 앞서 생성한 구성 파일(`/etc/no-ip2.conf`)을 읽고 사용자가 선택한 퍼블릭 DNS 이름의 IP 주소를 업데이트합니다.

5. 업데이트 클라이언트가 동적 DNS 이름의 IP 주소를 올바르게 설정했는지 확인합니다. 몇 분 동안 DNS 레코드가 업데이트되기를 기다린 후, 이 절차에서 구성한 퍼블릭 DNS 이름을 사용하여 SSH를 통해 인스턴스에 연결해 봅니다.

시작 시 Linux 인스턴스에서 명령 실행

Amazon EC2에서 인스턴스를 시작할 때 사용자 데이터를 인스턴스에 전달하여 일반적인 구성 작업을 자동으로 수행하는 데 사용하도록 할 수 있고, 인스턴스가 시작된 후에 스크립트를 실행할 수도 있습니다. Amazon EC2에 두 가지 유형의 사용자 데이터(shell 스크립트 및 `cloud-init` 명령)를 전달할 수 있습니다. 시작 마법사에 이 데이터를 일반 텍스트 파일(명령줄 도구를 통해 인스턴스를 시작하는 데 유용) 또는 `base64` 인코딩 텍스트(API 호출용)로 전달할 수도 있습니다.

더욱 복잡한 자동화 시나리오를 원하는 경우 AWS CloudFormation 또는 AWS OpsWorks를 사용할 수 있습니다. 자세한 내용은 [AWS CloudFormation 사용 설명서](#) 및 [AWS OpsWorks User Guide](#) 섹션을 참조하십시오.

시작 시 Windows 인스턴스에서 명령 실행에 대한 정보는 Windows 인스턴스용 Amazon EC2 사용 설명서에서 [사용자 데이터 실행](#) 및 [Windows 인스턴스 구성 관리](#) 섹션을 참조하십시오.

다음 예제에서는 [LAMP 웹 서버 설치 자습서 \(p. 27\)](#)의 명령을 인스턴스 시작 시 실행되는 shell 스크립트 및 `cloud-init` 명령 세트로 변환합니다. 각 예제에서는 사용자 데이터에 따라 다음 작업을 실행합니다.

- 배포 소프트웨어 패키지를 업데이트합니다.
- 필요한 웹 서버, `php` 및 `mysql` 패키지를 설치합니다.
- `httpd` 서비스를 시작하고 `chkconfig`를 통해 활성화합니다.
- `www` 그룹을 추가하고 해당 그룹에 `ec2-user`를 추가합니다.
- 웹 디렉터리 및 해당 디렉터리에 들어 있는 파일에 적절한 소유권과 파일 권한을 설정합니다.
- 간단한 웹 페이지를 생성하여 웹 서버 및 `php` 엔진을 테스트합니다.

Note

기본적으로 사용자 데이터 및 `cloud-init` 명령은 인스턴스 시작 시 최초 부팅 주기에서만 실행됩니다. 그러나 AWS Marketplace 공급업체와 타사 AMI 소유자가 스크립트 실행 방식과 시기를 원하는 대로 사용자 지정했을 수 있습니다.

목차

- [사전 조건 \(p. 318\)](#)
- [사용자 데이터 및 shell 스크립트 \(p. 318\)](#)
- [사용자 데이터 및 cloud-init 명령 \(p. 319\)](#)
- [API 및 CLI 개요 \(p. 321\)](#)

사전 조건

다음 예제에서는 인터넷에서 접속 가능한 퍼블릭 DNS 이름이 인스턴스에 지정되었다고 가정합니다. 자세한 내용은 [1단계: 인스턴스 시작 \(p. 22\)](#)을 참조하십시오. 또한 `ssh`(포트 22), `HTTP`(포트 80), `HTTPS`(포트 443) 연결을 허용하도록 보안 그룹을 구성해야 합니다. 이 사전 요구사항에 대한 자세한 내용은 [Amazon EC2로 설정 \(p. 16\)](#)을 참조하십시오.

또한 이러한 명령은 Amazon Linux에만 사용해야 합니다. 다른 Linux 배포에서는 명령이 작동하지 않을 수 있습니다. 다른 배포에 대한 `cloud-init` 지원 등의 자세한 내용은 해당 설명서를 참조하십시오.

사용자 데이터 및 shell 스크립트

shell 스크립트에 익숙한 경우, 이 방법은 인스턴스 시작 시 명령을 전송하는 가장 쉽고 완벽한 방법입니다. `cloud-init` 출력 로그 파일(`/var/log/cloud-init-output.log`)이 콘솔 출력을 캡처하므로 시작 후 인스턴스가 의도한 대로 동작하지 않더라도 스크립트를 손쉽게 디버깅할 수 있습니다.

Important

사용자 데이터 스크립트 및 `cloud-init` 명령은 인스턴스 시작 시 최초 부팅 주기에서만 실행됩니다.

사용자 데이터 shell 스크립트는 `#!` 문자 및 스크립트를 읽을 인터프리터의 경로(일반적으로 `/bin/bash`)로 시작되어야 합니다. shell 스크립트에 대한 자세한 소개는 [Linux Documentation Project\(tldp.org\)](#)에서 [BASH Programming - Introduction HOW-TO](#) 섹션을 참조하십시오.

사용자 데이터로 입력된 스크립트는 `root` 사용자 권한으로 실행되므로 스크립트에 `sudo` 명령을 사용하지 마십시오. 생성하는 모든 파일의 소유권은 `root`에 있습니다. `root`가 아닌 사용자에게 파일 액세스를 허용하려면 스크립트에서 권한을 적절히 수정해야 합니다. 또한 스크립트는 대화형으로 실행되지 않으므로 사용자의 입력이 필요한 명령(예: `-y` 플래그 없는 `yum update`)은 포함할 수 없습니다.

부팅 시에 이러한 작업을 추가하면 인스턴스 부팅에 걸리는 시간이 그만큼 늘어납니다. 사용자 스크립트가 성공적으로 완료되었는지 테스트하려면 우선 작업이 완료될 수 있도록 몇 분의 여유 시간을 두어야 합니다.

인스턴스에 사용자 데이터로 shell 스크립트를 전달하려면 다음을 수행합니다.

1. [AMI에서 인스턴스 시작 \(p. 265\)](#)의 인스턴스 시작 절차를 따르되, [Step 6 \(p. 266\)](#)까지 진행하고 [User data] 필드에 사용자 데이터 스크립트 텍스트를 붙여넣은 후 시작 절차를 완료합니다. 아래 예제에서는 스크립트가 웹 서버를 생성하고 구성합니다.

```
#!/bin/bash
yum update -y
yum install -y httpd24 php56 mysql55-server php56-mysqld
service httpd start
chkconfig httpd on
groupadd www
usermod -a -G www ec2-user
chown -R root:www /var/www
chmod 2775 /var/www
find /var/www -type d -exec chmod 2775 {} +
find /var/www -type f -exec chmod 0664 {} +
echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
```

2. 인스턴스가 시작되고 스크립트의 명령이 실행되도록 충분한 시간을 허용한 후 스크립트에서 의도된 작업을 완료했는지 확인합니다. 이 예제의 경우 스크립트가 생성한 PHP 테스트 파일의 URL을 웹 브라우저에 입력합니다. 이 URL은 인스턴스의 퍼블릭 DNS 주소에 슬래시(/)와 파일 이름이 추가된 형태입니다.

```
http://my.public.dns.amazonaws.com/phpinfo.php
```

PHP 정보 페이지가 표시되어야 합니다.

Tip

PHP 정보 페이지가 표시되지 않는 경우 사용하고 있는 보안 그룹이 HTTP(포트 80) 트래픽을 허용하는 규칙을 포함하고 있는지 확인하십시오. 보안 그룹에 HTTP 규칙을 추가하는 방법은 [보안 그룹에 규칙 추가 \(p. 390\)](#)을 참조하십시오.

3. (선택 사항) 스크립트가 의도한 작업을 완료하지 못한 경우, 또는 단지 스크립트가 오류 없이 완료되었는지 확인하고자 하는 경우에는 `/var/log/cloud-init-output.log`에서 `cloud-init` 출력 로그 파일을 검토하여 출력에 나타난 오류 메시지를 찾아봅니다.

다음 명령을 사용하여 `cloud-init` 데이터 섹션을 포함하는 Mime 멀티파트 아카이브를 생성하면 자세한 디버깅 정보를 확인할 수 있습니다.

```
output : { all : '| tee -a /var/log/cloud-init-output.log' }
```

이 명령은 스크립트의 명령 출력을 `/var/log/cloud-init-output.log`로 전송합니다. `cloud-init` 데이터 형식 및 Mime 멀티파트 아카이브를 생성하는 방법에 대한 자세한 내용은 [cloud-init 형식](#)을 참조하십시오.

사용자 데이터 및 cloud-init 명령

`cloud-init` 패키지는 새 Amazon Linux 인스턴스가 시작될 때의 특정 측면을 구성합니다. 가장 널리 사용되는 기능은 사용자가 자신의 프라이빗 키로 로그인할 수 있도록 `ec2-user`의 `.ssh/authorized_keys` 파일을 구성하는 것입니다.

`cloud-init` 사용자 명령을 인스턴스 시작 시에 전달하는 방법은 스크립트를 전달하는 방법과 동일하지만 구문은 서로 다릅니다. `cloud-init`에 대한 자세한 내용은 <http://cloudinit.readthedocs.org/en/latest/index.html>을 참조하십시오.

Important

사용자 데이터 스크립트 및 `cloud-init` 명령은 인스턴스 시작 시 최초 부팅 주기에서만 실행됩니다.

`cloud-init`의 Amazon Linux 버전은 기본 패키지가 제공하는 명령 중 일부를 지원하지 않으며 몇 가지 명령의 이름이 바뀌었습니다(예: `apt-upgrade` 대신 `repo_update` 사용).

부팅 시에 이러한 작업을 추가하면 인스턴스 부팅에 걸리는 시간이 그만큼 늘어납니다. 사용자 데이터 명령이 완료되었는지 테스트하려면 우선 작업이 완료될 수 있도록 몇 분의 여유 시간을 두어야 합니다.

인스턴스에 사용자 데이터로 `cloud-init` 명령을 전달하려면 다음을 수행합니다.

1. [AMI에서 인스턴스 시작 \(p. 265\)](#)의 인스턴스 시작 절차를 따르되, [Step 6 \(p. 266\)](#)까지 진행하고 [User data] 필드에 `cloud-init` 명령 텍스트를 붙여넣은 후 시작 절차를 완료합니다. 아래 예제에서는 명령을 통해 웹 서버를 생성하고 구성합니다.

```
#cloud-config
repo_update: true
repo_upgrade: all

packages:
- httpd24
- php56
- mysql55-server
- php56-mysqlnd

runcmd:
- service httpd start
- chkconfig httpd on
- groupadd www
- [ sh, -c, "usermod -a -G www ec2-user" ]
- [ sh, -c, "chown -R root:www /var/www" ]
- chmod 2775 /var/www
- [ find, /var/www, -type, d, -exec, chmod, 2775, {}, + ]
- [ find, /var/www, -type, f, -exec, chmod, 0664, {}, + ]
- [ sh, -c, 'echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php' ]
```

2. 인스턴스가 시작되고 사용자 데이터의 명령이 실행되도록 충분한 시간을 허용한 후 명령에서 의도된 작업을 완료했는지 확인합니다. 이 예제의 경우 명령에서 생성한 PHP 테스트 파일의 URL을 웹 브라우저에 입력합니다. 이 URL은 인스턴스의 퍼블릭 DNS 주소에 슬래시(/)와 파일 이름이 추가된 형태입니다.

```
http://my.public.dns.amazonaws.com/phpinfo.php
```

PHP 정보 페이지가 표시되어야 합니다.

Tip

PHP 정보 페이지가 표시되지 않는 경우 사용하고 있는 보안 그룹이 HTTP(포트 80) 트래픽을 허용하는 규칙을 포함하고 있는지 확인하십시오. 보안 그룹에 HTTP 규칙을 추가하는 방법은 [보안 그룹에 규칙 추가 \(p. 390\)](#)을 참조하십시오.

3. ([선택 사항](#)) 명령에서 의도한 작업을 완료하지 못한 경우, 또는 단지 명령이 오류 없이 완료되었는지 여부를 확인하고자 하는 경우에는 `/var/log/cloud-init-output.log`에서 `cloud-init` 출력 로그 파일을 검토하여 출력에 나타난 오류 메시지를 찾아봅니다. 명령에 다음 줄을 추가하면 자세한 디버깅 정보를 확인할 수 있습니다.

```
output : { all : '| tee -a /var/log/cloud-init-output.log' }
```

이 명령은 `runcmd` 출력을 `/var/log/cloud-init-output.log`로 전송합니다.

API 및 CLI 개요

다음 명령 중 하나를 사용하여 시작 중에 사용자 데이터를 인스턴스로 전달할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 정보는 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- AWS CLI: `run-instances` 명령과 함께 `--user-data` 파라미터를 사용하십시오. `file://` 접두사를 사용하여 파일에서 사용자 데이터를 전달하십시오.
- Windows PowerShell용 AWS 도구: `New-EC2Instance` 명령과 함께 `-UserData` 파라미터를 사용하십시오.
- Amazon EC2 쿼리 API: `RunInstances` 명령과 함께 `UserData` 파라미터를 사용하십시오.

인스턴스 메타데이터 및 사용자 데이터

인스턴스 메타데이터는 실행 중인 인스턴스를 구성 또는 관리하는 데 사용될 수 있는 인스턴스 관련 데이터입니다. 인스턴스 메타데이터는 몇 가지 범주로 분류될 수 있습니다. 자세한 내용은 [인스턴스 메타데이터 카테고리 \(p. 328\)](#) 섹션을 참조하십시오.

또한, EC2 인스턴스에는 인스턴스가 시작되었을 때 생성되는 인스턴스 자격 증명 문서와 같은 동적 데이터가 포함됩니다. 자세한 내용은 [동적 데이터 카테고리 \(p. 332\)](#) 섹션을 참조하십시오.

사용자는 인스턴스 시작 시 제공한 사용자 데이터에도 액세스할 수 있습니다. 예를 들어, 인스턴스를 구성하기 위한 파라미터를 지정하거나 단순 스크립트를 추가하는 것도 가능합니다. 또한, 이 데이터를 사용하여 실행 시점에 제공된 구성 파일로 수정이 가능한 일반 AMI를 작성할 수도 있습니다. 예를 들어, 여러 소규모 비즈니스용으로 웹 서버를 운영하는 경우 모두 동일한 AMI를 사용하고 실행 시점에 사용자 데이터에 지정된 Amazon S3 버킷에서 콘텐츠를 가져올 수 있습니다. 언제라도 새 고객을 추가하려면 해당 고객용 버킷을 생성하고 내용을 추가한 다음 AMI를 시작하기만 하면 됩니다. 1개 이상의 인스턴스를 동시에 시작하는 경우 해당 동일 시작 인스턴스의 모든 인스턴스에서 사용자 데이터를 이용할 수 있습니다.

Important

사용자는 인스턴스 내에서 인스턴스 메타데이터 및 사용자 데이터에만 액세스할 수 있지만 암호화 기법을 통해 데이터가 암호화되지 않습니다. 인스턴스에 액세스하는 모든 사용자가 인스턴스의 메타데이터를 확인할 수 있습니다. 그러므로 민감한 데이터를 보호할 수 있는 적절한 예방 조치(수명이 긴 암호화 키 등)를 취해야 합니다. 비밀번호와 같은 민감한 사용자 데이터를 저장하지 마십시오.

목차

- [인스턴스 메타데이터 가져오기 \(p. 321\)](#)
- [사용자 데이터를 사용하여 인스턴스 구성 \(p. 324\)](#)
- [사용자 데이터 가져오기 \(p. 325\)](#)
- [동적 데이터 가져오기 \(p. 325\)](#)
- [예제: AMI 시작 색인 값 \(p. 325\)](#)
- [인스턴스 메타데이터 카테고리 \(p. 328\)](#)
- [인스턴스 자격 증명 문서 \(p. 332\)](#)

인스턴스 메타데이터 가져오기

실행 중인 인스턴스에서 인스턴스 메타데이터를 사용할 수 있기 때문에 Amazon EC2 콘솔 또는 AWS CLI를 사용할 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.

실행 중인 모든 인스턴스 메타데이터 범주를 살펴보려면 다음 URI를 사용하십시오.

```
http://169.254.169.254/latest/meta-data/
```

인스턴스 메타데이터 및 사용자 데이터를 가져오기 위해 사용되는 HTTP 요청 비용은 청구되지 않습니다.

cURL 등의 도구를 사용할 수 있고 인스턴스가 지원하는 경우에는 GET 명령어를 사용할 수 있습니다. 예:

```
$ curl http://169.254.169.254/latest/meta-data/
```

```
$ GET http://169.254.169.254/latest/meta-data/
```

또한 인스턴스 메타데이터 쿼리 도구를 다운로드하면 전체 URI 또는 카테고리 이름을 입력하지 않아도 인스턴스 메타데이터를 쿼리할 수 있습니다.

<http://aws.amazon.com/code/1825>

모든 인스턴스 메타데이터는 텍스트로 반환됩니다(콘텐츠 유형 `text/plain`). 특정 메타데이터 리소스를 요청하면 적절한 값이 반환되거나 소스를 이용할 수 없는 경우 `404 - Not Found` HTTP 오류 코드가 반환됩니다.

일반 메타데이터 리소스(/로 끝나는 URI)를 요청한 경우 이용 가능한 리소스 목록이 반환되거나 해당 리소스가 없는 경우 `404 - Not Found` HTTP 오류 코드가 반환됩니다. 목록 항목은 개별 라인에 표시되고 줄바꿈(ASCII 10)으로 끝납니다.

인스턴스 메타데이터 가져오기 예제

이 예제를 통해 이용 가능한 인스턴스 메타데이터 버전을 가져올 수 있습니다. 이 버전과 Amazon EC2 API 버전은 서로 상관이 없습니다. 이전 버전의 구조 및 정보를 사용하는 스크립트인 경우 이전 버전을 사용할 수 있습니다.

```
$ 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
latest
```

이 예제는 최고 수준 메타데이터 항목을 가져옵니다. 일부 항목은 VPC 인스턴스에서만 사용할 수 있습니다. 각 항목에 대한 자세한 내용은 [인스턴스 메타데이터 카테고리 \(p. 328\)](#) 섹션을 참조하십시오.

```
$ curl http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
hostname
instance-action
instance-id
instance-type
kernel-id
local-hostname
local-ipv4
mac
network/
```

```
placement/  
public-hostname  
public-ipv4  
public-keys/  
reservation-id  
security-groups  
services/
```

이 예제에서는 이전 예제에서 일부 메타데이터 항목의 값을 획득합니다.

```
$ curl http://169.254.169.254/latest/meta-data/ami-id  
ami-12345678
```

```
$ curl http://169.254.169.254/latest/meta-data/reservation-id  
r-fea54097
```

```
$ curl http://169.254.169.254/latest/meta-data/local-hostname  
ip-10-251-50-12.ec2.internal
```

```
$ curl http://169.254.169.254/latest/meta-data/public-hostname  
ec2-203-0-113-25.compute-1.amazonaws.com
```

이 예제를 통해 이용 가능한 퍼블릭 키 목록을 획득할 수 있습니다.

```
$ curl http://169.254.169.254/latest/meta-data/public-keys/  
0=my-public-key
```

이 예제는 퍼블릭 키 0을 이용할 수 있는 형식을 보여줍니다.

```
$ curl http://169.254.169.254/latest/meta-data/public-keys/0/  
openssh-key
```

이 예제에서는 퍼블릭 키 0(OpenSSH 키 형식)을 획득합니다.

```
$ curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key  
ssh-rsa MIICitCCAfICCQD6m7oRw0uXOjANBgkqhkiG9wOBAQUFADCbIDELMAkGA1UEBhMC  
VVMxCzAJBgNVBAgTAlDbMRawDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6  
b24xFDASBgNVBASTC01BTSBDb25zb2x1MRIwEAYDVQDewlUZXN0Q21sYWMxHzAd  
BgkqhkiG9wOBCQEWEg5vb251QGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN  
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVVMxCzAJBgNVBAgTAlDbMRawDgYD  
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAstC01BTSBDb25z  
b2x1MRIwEAYDVQDewlUZXN0Q21sYWMxHzAdBgkqhkiG9wOBCQEWEg5vb251QGFt  
YXpvbi5jb20wgZ8wDQYJKoZIhvNAQEBBQADgY0AMIGJAoGBAMAk0dn+a4GmWIJ  
21uUSfwfEvYSwTC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T  
rDHudUZg3qX4waLG5M43q7Wgc/MbqITxOUSQv7c7ugFFDzQGBzZswY6786m86gPE  
Ibb3OhjZnzcvcQAArHdlQWIMm2nrAgMBAAEwDQYJKoZIhvNAQEFBQADgYEAtCu4  
nUhVvXyUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb  
FFBjvSfpJ1IJ00zbhNY5f6GuoEdmFJ10ZxBHJnyp378OD8uTs7fLvjkx79LjSTb  
NYiytVbZPQU5Yaxu2jXnimvw3rrszlaEXAMPLE my-public-key
```

이 예제는 EC2-Classic 플랫폼의 NAT 인스턴스에 있는 특정 네트워크 인터페이스(MAC 주소로 표시됨)에서 사용할 수 있는 정보를 보여줍니다.

```
$ curl http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:29:96:8f:6a:2d/  
device-number  
local-hostname  
local-ipv4s
```

```
mac
owner-id
public-hostname
public-ipv4s
```

이 예제에서는 VPC에서 시작된 인스턴스의 서브넷 ID를 획득합니다.

```
$ curl http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:29:96:8f:6a:2d/
subnet-id
subnet-be9b61d7
```

Throttling

쿼리는 인스턴스당 인스턴스 메타데이터 서비스로 스로틀링되고, 한 인스턴스에서 인스턴스 메타데이터 서비스로의 동시 연결 수에도 제한이 있습니다.

인스턴스 메타데이터 서비스를 사용하여 AWS 보안 자격 증명을 가져올 경우, 트랜잭션을 수행할 때마다 또는 많은 스레드나 프로세스로부터 동시에 자격 증명을 쿼리하지 마십시오. 이렇게 하면 스로틀링이 발생할 수 있습니다. 자격 증명 만료일이 다가오기 전까지는 자격 증명을 캐시에 저장하는 것이 좋습니다.

인스턴스 메타데이터 서비스를 액세스할 때 스로틀링이 발생하면 지수 백오프 전략으로 쿼리를 다시 시도하십시오.

사용자 데이터를 사용하여 인스턴스 구성

사용자 데이터를 지정하는 경우 다음에 유의하십시오.

- 사용자 데이터는 불투명 데이터로 취급됨: 제공한 것만을 살펴볼 수 있습니다. 해석 가능성은 인스턴스에 따라 다릅니다.
- 사용자 데이터의 크기는 16KB로 제한됩니다. 이 제한은 베이스64 인코딩 형식이 아닌 원시 형식 데이터에 적용됩니다.
- API로 제출되기 전 사용자 데이터는 베이스64로 인코딩되어야 합니다. AWS CLI 및 Amazon EC2 콘솔은 사용자를 위해 base64 인코딩을 수행합니다. 데이터는 인스턴스에 위치하기 전에 디코딩됩니다. base64 인코딩에 대한 자세한 내용은 <http://tools.ietf.org/html/rfc4648>을 참조하십시오.
- 시작 시에만 사용자 데이터가 실행됩니다. 인스턴스를 중지하고, 사용자 데이터를 수정하고, 인스턴스를 시작하는 경우에는 새 사용자 데이터가 자동으로 실행되지 않습니다.

인스턴스 시작 시 사용자 데이터를 지정하려면

인스턴스를 시작할 때 사용자 데이터를 지정할 수 있습니다. 자세한 내용은 [인스턴스 시작하기 \(p. 265\)](#), [cloud-init \(p. 134\)](#), [시작 시 Linux 인스턴스에서 명령 실행 \(p. 317\)](#) 섹션을 참조하십시오.

실행 중인 인스턴스에 대한 사용자 데이터 수정

기존 인스턴스의 사용자 데이터를 수정할 수 있습니다. 인스턴스가 실행 중인 경우 인스턴스를 먼저 중단해야 합니다. 새 사용자 데이터는 인스턴스를 다시 시작한 뒤에 사용할 수 있습니다.

Amazon EBS 지원 인스턴스에 대한 사용자 데이터를 수정하려면

- <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
- 탐색 창에서 [Instances]를 선택하고 인스턴스를 선택합니다.
- [Actions]를 클릭하고 [Instance State]를 선택한 다음 [Stop]을 선택합니다.

Warning

인스턴스를 중지하면 인스턴스 스토어 볼륨의 데이터가 삭제됩니다. 따라서 인스턴스 스토어 볼륨에 보존하려는 데이터가 있을 경우 영구 스토리지에 백업하십시오.

4. 확인 대화 상자가 나타나면 [Yes, Stop]을 클릭합니다. 인스턴스가 중지하는 데 몇 분 정도 걸릴 수 있습니다.
5. 인스턴스가 선택된 상태에서 [Actions]를 선택하고 [Instance Settings]를 선택한 후 [View/Change User Data]를 선택합니다. 인스턴스가 실행 중일 때는 사용자 데이터를 변경할 수 없지만 볼 수는 있습니다.
6. [View/Change User Data] 대화 상자에서 사용자 데이터를 업데이트하고 [Save]를 선택합니다.

사용자 데이터 가져오기

사용자 데이터를 가져오려면 다음 URI를 사용합니다.

```
http://169.254.169.254/latest/user-data
```

사용자 데이터를 요청하면 데이터 자체(콘텐츠 유형 application/octet-stream)가 반환됩니다.

이를 통해 콤마로 구분되어 반환되는 사용자 데이터의 예제가 표시됩니다.

```
$ curl http://169.254.169.254/latest/user-data
1234, john,reboot,true | 4512,richard, | 173,,,
```

이를 통해 라인으로 구분되어 반환되는 사용자 데이터의 예제가 표시됩니다.

```
$ curl http://169.254.169.254/latest/user-data
[general]
instances: 4

[instance-0]
s3-bucket: <user_name>

[instance-1]
reboot-on-error: yes
```

동적 데이터 가져오기

실행 중인 동적 데이터를 가져오려면 다음 URI를 사용하십시오.

```
http://169.254.169.254/latest/dynamic/
```

이 예제는 고수준 인스턴스 자격 증명 카테고리를 가져오는 방법을 보여줍니다.

```
$ curl http://169.254.169.254/latest/dynamic/instance-identity/
pkcs7
signature
document
```

동적 데이터 및 가져오기 방법의 예제에 대한 자세한 내용은 [인스턴스 자격 증명 문서 \(p. 332\)](#) 섹션을 참조하십시오.

예제: AMI 시작 색인 값

이 예제는 사용자 데이터 및 인스턴스 메타데이터를 사용하여 인스턴스를 구성하는 방법을 보여줍니다.

Alice는 데이터베이스 AMI 인스턴스 4개를 시작하여 그 중 첫 번째 인스턴스는 마스터의 역할을 하고 나머지 3개는 복제본의 역할을하도록 하려고 합니다. 그러한 인스턴스는 시작되었을 때 각 복제품의 복제 전략에 대한 사용자 데이터가 추가될 수 있어야 합니다. Alice는 네 인스턴스 모두에서 이 데이터가 사용될 수 있다는

것을 알고 있기 때문에 각 인스턴스가 적용 가능한 부분을 인식할 수 있도록 하는 방식으로 사용자 데이터를 구축해야 합니다. Alice는 `ami-launch-index` 인스턴스 메타데이터 값은 이를 수행할 수 있고 이 값은 각 인스턴스에서 공유합니다.

Alice가 구성한 사용자 데이터는 다음과 같습니다.

```
replicate-every=1min | replicate-every=5min | replicate-every=10min
```

`replicate-every=1min` 데이터는 최초 복제 구성은 정의하고 `replicate-every=5min`는 두 번째 복제 구성은 하는 식으로 동작합니다. Alice는 서로 다른 인스턴스의 데이터 구분자로 파이프 기호(|)를 사용하는 ASCII 문자열로 이 데이터를 제공하려 합니다.

Alice는 `run-instances` 명령으로 4개의 인스턴스를 시작하고 다음과 같이 사용자 데이터를 지정합니다.

```
aws ec2 run-instances --image-id ami-12345678 --count 4 --instance-type t2.micro --user-data "replicate-every=1min | replicate-every=5min | replicate-every=10min"
```

시작된 이후 모든 인스턴스는 다음과 같은 사용자 데이터 및 공통 메타데이터 사본을 갖습니다.

- AMI id: ami-12345678
- 예약 ID: r-1234567890abcabc0
- 퍼블릭 키: 없음
- 보안 그룹 이름: 기본
- 인스턴스 유형: t2.micro

그러나 각 인스턴스에는 고유한 특정 메타데이터가 있습니다.

인스턴스 1

Metadata	값
instance-id	i-1234567890abcdef0
ami-launch-index	0
public-hostname	ec2-203-0-113-25.compute-1.amazonaws.com
public-ipv4	67.202.51.223
local-hostname	ip-10-251-50-12.ec2.internal
local-ipv4	10.251.50.35

인스턴스 2

Metadata	값
instance-id	i-0598c7d356eba48d7
ami-launch-index	1
public-hostname	ec2-67-202-51-224.compute-1.amazonaws.com
public-ipv4	67.202.51.224
local-hostname	ip-10-251-50-36.ec2.internal

Metadata	값
local-ipv4	10.251.50.36

인스턴스 3

Metadata	값
instance-id	i-0ee992212549ce0e7
ami-launch-index	2
public-hostname	ec2-67-202-51-225.compute-1.amazonaws.com
public-ipv4	67.202.51.225
local-hostname	ip-10-251-50-37.ec2.internal
local-ipv4	10.251.50.37

인스턴스 4

Metadata	값
instance-id	i-1234567890abcdef0
ami-launch-index	3
public-hostname	ec2-67-202-51-226.compute-1.amazonaws.com
public-ipv4	67.202.51.226
local-hostname	ip-10-251-50-38.ec2.internal
local-ipv4	10.251.50.38

Alice는 ami-시작-색인 값을 사용하여 사용자 데이터의 어느 부분이 특정 인스턴스에 적용 가능한지를 결정 할 수 있습니다.

1. Alice는 인스턴스 중 하나에 접속한 다음 해당 인스턴스의 ami-시작-색인을 검색하여 복제본인지 확인합니다.

```
$ curl http://169.254.169.254/latest/meta-data/ami-launch-index
2
```

2. ami-시작-색인을 변수로 저장합니다.

```
$ ami_launch_index=`curl http://169.254.169.254/latest/meta-data/ami-launch-index`
```

3. 사용자 데이터를 변수로 저장합니다.

```
$ user_data=`curl http://169.254.169.254/latest/user-data/`
```

4. 마지막으로 Alice는 cut 명령을 사용하여 해당 인스턴스에 적용 가능한 사용자 데이터 부분을 추출합니다.

```
$ echo $user_data | cut -d"|" -f"$ami_launch_index"
```

replicate-every=5min

인스턴스 메타데이터 카테고리

다음 표는 인스턴스 메타데이터의 카테고리를 목록으로 표시합니다.

테스트	설명	버전 소개
ami-id	인스턴스를 시작하기 위해 사용된 AMI ID.	1.0
ami-launch-index	1개 이상의 인스턴스를 동시에 시작하는 경우 이 값은 인스턴스가 시작된 순서를 나타냅니다. 첫 번째 인스턴스의 값은 0입니다.	1.0
ami-manifest-path	Amazon S3에 위치한 AMI 매니페스트 파일 경로. Amazon EBS 지원 AMI를 사용하여 인스턴스를 시작한 경우 반환되는 결과는 <code>unknown</code> 입니다.	1.0
ancestor-ami-ids	이 AMI를 생성하기 위해 다시 번들링 된 모든 인스턴스의 AMI ID. 이 값은 AMI 매니페스트 파일에 <code>ancestor-ams</code> 키가 있는 경우에만 존재합니다.	2007-10-10
block-device-mapping/ami	루트/부트 파일 시스템을 포함하는 가상 디바이스.	2007-12-15
block-device-mapping/ebs N	존재하는 경우 Amazon EBS와 연결된 가상 디바이스. Amazon EBS 볼륨은 시작 시 존재하는 경우 또는 인스턴스를 마지막으로 시작한 시점에만 메타데이터에서 사용할 수 있습니다. N은 Amazon EBS 볼륨의 색인을 나타냅니다(<code>ebs1</code> 또는 <code>ebs2</code> 등).	2007-12-15
block-device-mapping/ephemeral N	존재하는 경우 사용 후 삭제 디바이스와 연결된 가상 디바이스. N은 사용 후 삭제 볼륨의 색인을 나타냅니다.	2007-12-15
block-device-mapping/root	루트 디바이스와 연결된 가상 디바이스 또는 파티션 또는 루트(/ 또는 C:) 파일 시스템이 해당 인스턴스와 연결된 가상 디바이스 파티션.	2007-12-15
block-device-mapping/swap	<code>swap</code> 와 연결된 가상 디바이스. 항상 존재하는 것은 아님.	2007-12-15
hostname	인스턴스의 프라이빗 IPv4 DNS 호스트 이름. 다중 네트워크 인터페이스가 존재하는 경우 eth0 디바이스를 의미함(디바이스 번호가 0인 디바이스).	1.0
iam/info	인스턴스 시작 시 IAM 역할이 연결되어 있을 경우, 인스턴스의 LastUpdated date, InstanceProfileArn 및 InstanceProfileId 등 마지막으로 인	2012-01-12

테스트	설명	버전 소개
	스턴스 프로파일이 업데이트된 시간 관련 정보를 포함합니다. 그렇지 않을 경우 제공되지 않습니다.	
iam/security-credentials/role-name	인스턴스 시작 시 IAM 역할이 연결되어 있을 경우 role-name 은 역할 이름이고 role-name 에는 이 역할과 연결된 임시 보안 자격 증명이 들어 있습니다 (자세한 내용은 인스턴스 메타데이터에서 보안 자격 증명 검색 (p. 457) 참조). 그렇지 않을 경우 제공되지 않습니다.	2012-01-12
instance-action	번들링을 준비하기 위해 재부팅되어야 함을 인스턴스에 통지합니다. 유효한 값: none shutdown bundle-pending.	2008-09-01
instance-id	이 인스턴스의 ID.	1.0
instance-type	인스턴스 유형. 자세한 내용은 인스턴스 유형 (p. 146) 섹션을 참조하십시오.	2007-08-29
kernel-id	이 인스턴스와 함께 시작한 커널 ID(해당하는 경우).	2008-02-01
local-hostname	인스턴스의 프라이빗 IPv4 DNS 호스트 이름. 다중 네트워크 인터페이스가 존재하는 경우 eth0 디바이스를 의미함(디바이스 번호가 0인 디바이스).	2007-01-19
local-ipv4	인스턴스의 프라이빗 IPv4 주소. 다중 네트워크 인터페이스가 존재하는 경우 eth0 디바이스를 의미함(디바이스 번호가 0인 디바이스).	1.0
mac	인스턴스의 미디어 액세스 제어(MAC) 주소. 다중 네트워크 인터페이스가 존재하는 경우 eth0 디바이스를 의미함(디바이스 번호가 0인 디바이스).	2011-01-01
network/interfaces/macs/mac/device-number	해당 인터페이스와 연결된 고유한 디바이스 번호. 이 디바이스 번호는 디바이스 이름과 부합됩니다. 예를 들어 device-number 2는 eth2 디바이스의 번호입니다. 이 범주는 AWS CLI용 Amazon EC2 API 및 EC2 명령에서 사용하는 DeviceIndex 및 device-index 필드와 부합됩니다.	2011-01-01
network/interfaces/macs/mac/ipv4-associations/public-ip	각 public-ip 주소에 연결되고 해당 인터페이스에 할당된 프라이빗 IPv4 주소.	2011-01-01
network/interfaces/macs/mac/ipv6s	IPv6 주소는 인터페이스와 연결됩니다. VPC에서 시작된 인스턴스인 경우에만 반환됩니다.	2016-06-30

테스트	설명	버전 소개
network/interfaces/macs/mac/ local-hostname	인터페이스의 로컬 호스트 이름.	2011-01-01
network/interfaces/macs/mac/ local-ipv4s	프라이빗 IPv4 주소는 인터페이스와 연결됩니다.	2011-01-01
network/interfaces/macs/mac/mac	인스턴스의 MAC 주소.	2011-01-01
network/interfaces/macs/mac/ owner-id	네트워크 인터페이스 소유자 ID. 다중 인터페이스 환경에서 인터페이스는 Elastic Load Balancing 등 타사 제품이 연결될 수 있습니다. 인터페이스 상의 트래픽은 항상 인터페이스 소유자에게 청구됩니다.	2011-01-01
network/interfaces/macs/mac/ public-hostname	인터넷 인터페이스의 퍼블릭 DNS(IPv4). 인스턴스가 VPC에 위치하는 경우 enableDnsHostnames 속성이 true로 설정된 경우에만 이 카테고리가 반환됩니다. 자세한 내용은 VPC에서 DNS 사용하기 섹션을 참조하십시오.	2011-01-01
network/interfaces/macs/mac/ public-ipv4s	탄력적 IP 주소는 인터페이스와 연결됩니다. 인스턴스에는 다중 IPv4 주소가 있을 수 있습니다.	2011-01-01
network/interfaces/macs/mac/ security-groups	네트워크 인터페이스에 속한 보안 그룹. VPC에서 시작된 인스턴스인 경우에만 반환됩니다.	2011-01-01
network/interfaces/macs/mac/ security-group-ids	네트워크 인터페이스에 속한 보안 그룹의 ID. VPC에서 시작된 인스턴스인 경우에만 반환됩니다. EC2-VPC 플랫폼에서의 보안 그룹에 대한 자세한 내용은 VPC의 보안 그룹 을 참조하십시오.	2011-01-01
network/interfaces/macs/mac/ subnet-id	인터넷 인터페이스가 위치하는 서브넷 ID. VPC에서 시작된 인스턴스인 경우에만 반환됩니다.	2011-01-01
network/interfaces/macs/mac/ subnet-ipv4-cidr-block	인터넷 인터페이스가 위치하는 서브넷의 IPv4 CIDR 블록. VPC에서 시작된 인스턴스인 경우에만 반환됩니다.	2011-01-01
network/interfaces/macs/mac/ subnet-ipv6-cidr-blocks	인터넷 인터페이스가 위치하는 서브넷의 IPv6 CIDR 블록. VPC에서 시작된 인스턴스인 경우에만 반환됩니다.	2016-06-30
network/interfaces/macs/mac/ vpc-id	인터넷 인터페이스가 위치하는 VPC의 ID. VPC에서 시작된 인스턴스인 경우에만 반환됩니다.	2011-01-01
network/interfaces/macs/mac/ vpc-ipv4-cidr-block	인터넷 인터페이스가 위치하는 VPC의 IPv4 CIDR 블록. VPC에서 시작된 인스턴스인 경우에만 반환됩니다.	2011-01-01

테스트	설명	버전 소개
network/interfaces/macs/mac/ vpc-ipv4-cidr-blocks	인터페이스가 위치하는 VPC의 IPv4 CIDR 블록. VPC에서 시작된 인스턴스 인 경우에만 반환됩니다.	2016-06-30
network/interfaces/macs/mac/ vpc-ipv6-cidr-blocks	인터페이스가 위치하는 VPC의 IPv6 CIDR 블록. VPC에서 시작된 인스턴스 인 경우에만 반환됩니다.	2016-06-30
placement/availability-zone	인스턴스가 시작된 가용 영역.	2008-02-01
product-codes	인스턴스에 연결된 제품 코드(해당되는 경우).	2007-03-01
public-hostname	인스턴스의 퍼블릭 DNS. 인스턴스가 VPC에 위치하는 경우 enableDnsHostnames 속성이 true로 설정된 경우에만 이 카테고리가 반환됩니다. 자세한 내용은 VPC에서 DNS 사용하기 섹션을 참조하십시오.	2007-01-19
public-ipv4	퍼블릭 IPv4 주소. 인스턴스와 탄력적 IP 주소가 연결된 경우 반환된 값은 탄력적 IP 주소입니다.	2007-01-19
public-keys/0/openssh-key	퍼블릭 키. 시작 시에 인스턴스가 제공된 경우에만 사용할 수 있습니다.	1.0
ramdisk-id	시작 시에 지정된 RAM의 ID(해당하는 경우).	2007-10-10
reservation-id	예약 ID:	1.0
security-groups	인스턴스에 적용된 보안 그룹의 이름. 시작한 이후 사용자는 VPC에서 실행 중인 인스턴스의 보안 그룹만을 변경할 수 있습니다. 해당 변경은 여기 및 network/interfaces/macs/ <i>mac</i> /security-groups에 반영됩니다.	1.0
services/domain	리전의 AWS 리소스 도메인. 예: for us-east-1 관련 amazonaws.com .	2014-02-25
services/partition	리소스가 있는 파티션. 표준 AWS 리전에서 파티션은 aws입니다. 리소스가 다른 파티션에 있는 경우 파티션은 aws- <i>partitionname</i> 입니다. 예를 들어 중국(베이징) 리전에 있는 리소스의 파티션은 aws-cn입니다.	2015-10-20

테스트	설명	버전 소개
spot/termination-time	스팟 인스턴스의 운영 체제가 종료 신호를 수신하는 UTC 기준 예상 시간. Amazon EC2가 스팟 인스턴스에 종료 표시를 한 경우에만 이 항목이 존재하고 시간 값(예: 2015-01-05T18:02:00Z)이 포함됩니다. 사용자가 스팟 인스턴스를 직접 종료한 경우 종료 시간 항목에 시간이 설정되지 않습니다.	2014-11-05

동적 데이터 카테고리

다음 표는 동적 데이터의 카테고리를 목록으로 표시합니다.

테스트	설명	버전 소개
fws/instance-monitoring	고객이 CloudWatch에서 1분 세부 모니터링을 설정했는지 보여주는 값. 유효한 값: enabled disabled	2009-04-04
instance-identity/document	인스턴스 ID, 프라이빗 IP 주소 등 인스턴스 속성을 포함하는 JSON. 인스턴스 자격 증명 문서 (p. 332) 을 참조하십시오.	2009-04-04
instance-identity/pkcs7	문서의 신뢰성 및 서명 내용을 검증하는 데 사용됩니다. 인스턴스 자격 증명 문서 (p. 332) 을 참조하십시오.	2009-04-04
instance-identity/signature	출처 및 신뢰성을 검증하기 위해 다른 사용자가 사용할 수 있는 데이터. 인스턴스 자격 증명 문서 (p. 332) 을 참조하십시오.	2009-04-04

인스턴스 자격 증명 문서

인스턴스 자격 증명 문서는 인스턴스를 설명하는 JSON 파일입니다. 인스턴스 자격 증명 문서에는 문서에 제공된 정보의 정확도, 오리진 및 신뢰성을 확인하는데 사용할 수 있는 서명 및 PKCS7 서명이 함께 제공됩니다. 예를 들어, 유료 업데이트가 포함된 무료 소프트웨어를 다운로드했을 수 있습니다.

인스턴스 자격 증명 문서는 인스턴스를 시작할 때 생성되고 [인스턴스 메타데이터 \(p. 321\)](#)를 통해 인스턴스에 공개됩니다. 이 문서는 인스턴스의 속성(예: 구독 소프트웨어, 인스턴스 크기, 인스턴스 유형, 운영 체제, AMI 등)이 유효한지 검사합니다.

Important

인스턴스 자격 증명 문서와 서명은 동적인 특성을 지니고 있기 때문에 규칙적으로 인스턴스 자격 증명 문서와 서명을 검색하는 것이 좋습니다.

인스턴스 자격 증명 문서 및 서명 가져오기

인스턴스 자격 증명 문서를 검색하려면 실행 중인 인스턴스에서 아래의 URL을 사용합니다.

```
http://169.254.169.254/latest/dynamic/instance-identity/document

{
    "devpayProductCodes" : null,
    "availabilityZone" : "us-east-1d",
    "privateIp" : "10.158.112.84",
    "version" : "2010-08-31",
    "region" : "us-east-1",
```

Amazon Elastic Compute Cloud

Linux 인스턴스용 사용 설명서

인스턴스 메타데이터 및 사용자 데이터

```
"instanceId" : "i-1234567890abcdef0",
"billingProducts" : null,
"instanceType" : "t1.micro",
"accountId" : "123456789012",
"pendingTime" : "2015-11-19T16:32:11Z",
"imageId" : "ami-5fb8c835",
"kernelId" : "aki-919dcraf8",
"ramdiskId" : null,
"architecture" : "x86_64"
}
```

인스턴스 자격 증명 서명을 검색하려면 실행 중인 인스턴스에서 아래의 URL을 사용합니다.

<http://169.254.169.254/latest/dynamic/instance-identity/signature>
dExampLesjNQhhJan7p0RLpLSr7lJEF4V2DhKGlyoYVBouYRy9njyBCmhEayaGrHTs/AWY+LPx
1vSQuRF5n0gwPNcU06ICT0fNrm5IH7w9ydaexamplejJw8XvWPxbuRkcNOTAA1p4RtCAqm4ms
x2oALjWSxCBExample=

PKCS7 서명을 검색하려면 실행 중인 인스턴스에서 아래의 URL을 사용합니다.

<http://169.254.169.254/latest/dynamic/instance-identity/pkcs7>

MIICiTCCAFICCQD6m7oRw0uX0jgANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTA1dBMRAwDgYDVQQHEwdTzWFD0dgx1M0Q8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC01BTSBDb25zb2x1M1IwEAYDVQQDEwluZXN0Q21sYWMxHzAd
BgkqhkiG9w0BCQEWEg5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxJwJCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTA1dBMRAwDgYD
VQQHEwdTzWFD0dgx1M0Q8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC01BTSBDb25z
b2x1M1IwEAYDVQQDEwluZXN0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEg5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvCNaqEBBQADgY0AMIGJAOGBAMaK0dn+a4GmWIW
21uUSfwfEvySwC2XADZ4n+BLYgV1k60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHuDUsZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFDzQGbzZswY6786m86gpE
Ibb3OhjznzcvQAArHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvCNaqEFBQADgYEAtCu4
nUhvvxYUntneD9+h8Mg9q6q+auNrkyExzyLwax1Aoo7TJHidbtS4J51NmZgXL0Fkb
FFBjvSfpJ1J00zbhNYS5f6GuoEdmFJ10zxBHjJnyp378OD8uTs7flvjx79LjSTb
NYiytVbZPQU5Yaxu2jXnimvw3rrszlaEXAMPLE

예: PKCS7 서명 확인

PKCS7 서명을 사용해 리전의 AWS 퍼블릭 인증서에 대해 유효성을 검증함으로써 인스턴스를 확인할 수 있습니다.

모든 퍼블릭 리전을 위한 AWS 퍼블릭 인증서는 다음과 같습니다.

-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgchkjOOAQDMFwxCzAJBgNVBAYTA1VTMRkw
FwYDVQQIExBXYYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYD
VQOKExdBbWF6b24gV2ViIFNlcnPzY2VzIExEQzAeFwOxMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaFMwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKExdBbWF6b24gV2ViIFN1
cnPzY2VzIExEQzCABCwgEsBgcqhkjOOAQBMIIIBhKBgQCjkvcS2b2b1VQ4yt/5e
ih5006kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6UDlZ1gYipr58Kj3nssSNpI6bx3
VyiIQzK7wlclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmvMegN6P
hviYt5Jh/nY14hh3Pa1HJDskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tqkMVHuAfCvAGKocTgsJem6/5qomzJuKDmbJNu9Qxw3rAoXau8Qe+MBCJ1/U
hhy1KHVpCGl9fueQ2s6IL0Ca0/buyucU1CiYqk40KNHCChFnZbd1e9rpUp7bnF
1Ra2v1ntTMX3caRVdDtbPEWmdxSCSYfDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmp9CM5eovQOGx5h08Wqd+aTebs+k2tn92BBPqeZqpWrA5P/+jrdKml1qx411HW
MXrs3Igib6+hUIB+S8dz8/mm00brp76RoZVCXYab2CZedFut7gc3WUH9+EUAH5mw

```
vSeDCOUMYQR7R9LINYwouHIZiqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw  
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6R0k0k9K  
-----END CERTIFICATE-----
```

AWS GovCloud (US) 리전을 위한 AWS 퍼블릭 인증서는 다음과 같습니다.

```
-----BEGIN CERTIFICATE-----  
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgCqhkJOOAQDMFwxCzAJBgNVBAYTA1VTMRkw  
FwYDVQQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYD  
VQQKExdBbWF6b24gV2ViIFN1cnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJjaFw0z  
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIEExBXYXNoaW5ndG9u  
IFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKExdBbWF6b24gV2ViIFN1  
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIBHwKBgQCjkvcS2bb1VQ4yt/5e  
ih5006k/n1Lz1lr7D8ZwtQP8f0Epp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3  
VyiQzK7wLc1nd/YozqNnmgIyZecN7Eg1K9ITHJLP+x8FtUpT3QbyYXJdmVMegN6P  
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvWhh6+ERYRAoGBAI1j  
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJ1/U  
hhy1KHVpCG19fueQ2s6IL0CaO/buycU1CiYQk40KNHCchfNiZbdlx1E9rpUp7bnF  
1Ra2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf  
MNmP9CM5eoVQOGx5ho8WqD+aTeb+k2tn92BPqeZqpWRa5P/+jrdKml1qx4llHW  
MXrs3IgIb6+hUIB+S8dz8/mmOObpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw  
vSeDCOUMYQR7R9LINYwouHIZiqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw  
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6R0k0k9K  
-----END CERTIFICATE-----
```

AWS GovCloud (US)에 대한 자세한 내용은 [AWS GovCloud \(US\) User Guide](#)를 참조하십시오.

다른 리전에 대해서는 [AWS Support](#)에 문의하여 AWS 퍼블릭 인증서를 얻으십시오.

PKCS7 서명을 확인하려면

1. Amazon Linux 인스턴스에서 다음과 같이 PKCS7 서명을 위한 임시 파일을 만듭니다.

```
PKCS7=$(mktemp)
```

2. 다음과 같이 파일을 -----BEGIN PKCS7----- 헤더로 채운 다음, 인스턴스 메타데이터로부터 PKCS7 서명의 내용, 새 줄과 -----END PKCS7----- 바닥글을 추가합니다.

```
echo "-----BEGIN PKCS7-----" > $PKCS7
```

```
curl -s http://169.254.169.254/latest/dynamic/instance-identity/pkcs7 >> $PKCS7
```

```
echo "" >> $PKCS7
```

```
echo "-----END PKCS7-----" >> $PKCS7
```

3. 인스턴스 자격 증명 문서를 위한 임시 파일을 만들고 다음과 같이 인스턴스 메타데이터로부터 얻은 문서의 내용으로 파일을 채웁니다.

```
DOCUMENT=$(mktemp)
```

```
curl -s http://169.254.169.254/latest/dynamic/instance-identity/document > $DOCUMENT
```

4. 텍스트 편집기를 열어 `AWSPubkey`라는 파일을 만듭니다. 파일에 위 AWS 퍼블릭 인증서의 내용을 복사해 붙여 넣은 다음 저장합니다.
5. OpenSSL 도구를 사용해 다음과 같이 서명을 확인합니다.

```
openssl smime -verify -in $PKCS7 -inform PEM -content $DOCUMENT -certfile AWSpubkey -  
noverify > /dev/null  
Verification successful
```

혼합 컴퓨팅 환경에서 EC2 인스턴스 식별

Azure나 Google Cloud Platform 등 다른 클라우드 인프라에서 컴퓨터 리소스를 실행하는 경우 또는 VMware, Xen 또는 KVM에서 온프레미스 가상화를 사용하는 경우 가상 머신이 EC2 인스턴스인지 판단하는 단순한 메서드를 활용할 수 있습니다. 이 주제에서는 EC2 인스턴스를 식별하는 두 가지 방법을 설명합니다. 하나는 빠르지만 정확하지 않을 수 있으며, 다른 하나는 더 복잡하지만 확실합니다.

Xen 도메인 UUID 검사

이 섹션에서 설명하는 메서드는 Xen 도메인 UUID를 검사하여 Linux 가상 머신이 EC2 인스턴스인지 최적의 판단을 내립니다. 이 방법은 UUID의 시작 8진수에 문자 "ec2" 또는 "EC2"가 있는지 찾습니다.

Note

EC2에 있지 않은 Xen 인스턴스에 이러한 문자가 포함되었을 가능성은 낮습니다.

아래 방법으로 Xen UUID를 찾을 수 있습니다. Windows 인스턴스를 식별하는 방법에 대한 자세한 내용은 http://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/identify_ec2_instances.html을 참조하십시오.

- Linux VM에서 다음 명령을 실행합니다.

```
$ cat /sys/hypervisor/uuid
```

이렇게 하면 UUID가 반환됩니다.

```
ec2e1916-9099-7caf-fd21-012345abcdef
```

이 예에서 접두사로 쓰인 "ec2"는 현재 EC2 인스턴스를 보고 있을 가능성이 높음을 나타냅니다.

- 또는 HVM 인스턴스에서만 유일하게 데스크톱 관리 인터페이스(DMI)에 시스템 일련 번호 및 시스템 UUID(대문자 표기)와 동일한 UUID가 포함됩니다.

```
$ sudo dmidecode --string system-serial-number  
ec2e1916-9099-7caf-fd21-01234example  
$ sudo dmidecode --string system-uuid  
EC2E1916-9099-7CAF-FD21-01234EXAMPLE
```

Note

이전 메서드와 달리 DMI 메서드는 수퍼유저 권한을 요구합니다. 그러나 일부 이전 버전 Linux 커널은 /sys/를 통해 UUID를 표시하지 않을 수 있습니다.

인스턴스 자격 증명 문서 검사

EC2 인스턴스를 식별하는 명확하고 암호화된 방법은 서명이 포함된 인스턴스 자격 증명 문서를 확인하십시오. 이러한 문서는 라우팅할 수 없는 로컬 주소 <http://169.254.169.254/latest/dynamic/instance-identity/>에서 모든 EC2 인스턴스에 대해 제공됩니다. 자세한 내용은 [인스턴스 자격 증명 문서](#)를 참조하십시오.

Amazon EC2 모니터링

모니터링은 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 및 AWS 솔루션의 안정성, 가용성 및 성능을 유지하는 데 있어서 중요한 부분입니다. 발생하는 다중 지점 실패를 보다 쉽게 디버깅할 수 있도록 AWS 솔루션의 모든 부분으로부터 모니터링 데이터를 수집해야 합니다. 그러나 Amazon EC2 모니터링을 시작하려면 먼저 다음을 포함하는 모니터링 계획을 생성해야 합니다.

- 모니터링의 목표
- 모니터링할 리소스
- 이러한 리소스를 모니터링하는 빈도
- 사용할 모니터링 도구
- 모니터링 작업을 수행할 사람
- 문제 발생 시 알려야 할 대상

모니터링 목표를 정의하고 모니터링 계획을 생성했으면, 다음 단계는 환경에서 Amazon EC2 성능의 기준선을 설정하는 것입니다. 다양한 시간과 다양한 부하 조건에서 Amazon EC2 성능을 측정해야 합니다. Amazon EC2를 모니터링할 때 수집한 모니터링 데이터의 기록을 저장해야 합니다. 현재 Amazon EC2 성능을 이 기록 데이터와 비교하면 일반적인 성능 패턴과 성능 이상을 식별하고 이를 해결할 방법을 고안할 수 있습니다. 예를 들어, Amazon EC2 인스턴스에 대해 CPU 사용률, 디스크 I/O 및 네트워크 사용률을 모니터링할 수 있습니다. 설정한 기준 이하로 성능이 떨어지면 인스턴스를 재구성하거나 최적화하여 CPU 사용률을 줄이거나 디스크 I/O를 개선하거나 네트워크 트래픽을 줄일 수 있습니다.

기준선을 설정하려면 최소한 다음 항목을 모니터링해야 합니다.

모니터링할 항목	Amazon EC2 지표	모니터링 스크립트/CloudWatch Logs
CPU 사용률	CPUUtilization (p. 349)	
메모리 사용률		(Linux 인스턴스) Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances (Windows 인스턴스) 성능 카운터를 CloudWatch로, 로그를 CloudWatch Logs로 전송
사용된 메모리		(Linux 인스턴스) Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances

모니터링할 항목	Amazon EC2 지표	모니터링 스크립트/CloudWatch Logs
		(Windows 인스턴스) 성능 카운터를 CloudWatch로, 로그를 CloudWatch Logs로 전송
가용 메모리		(Linux 인스턴스) Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances (Windows 인스턴스) 성능 카운터를 CloudWatch로, 로그를 CloudWatch Logs로 전송
네트워크 사용률	NetworkIn (p. 349) NetworkOut (p. 349)	
디스크 성능	DiskReadOps (p. 349) DiskWriteOps (p. 349)	
디스크 스왑 사용률(Linux 인스턴스에만 해당) 사용된 스왑(Linux 인스턴스에만 해당)		Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances
페이지 파일 사용률(Windows 인스턴스에만 해당) 사용된 페이지 파일(Windows 인스턴스에만 해당) 사용 가능한 페이지 파일(Windows 인스턴스에만 해당)		성능 카운터를 CloudWatch로, 로그를 CloudWatch Logs로 전송
디스크 읽기/쓰기	DiskReadBytes (p. 349) DiskWriteBytes (p. 349)	
디스크 공간 사용률(Linux 인스턴스에만 해당)		Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances
사용된 디스크 공간(Linux 인스턴스에만 해당)		Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances
사용 가능한 디스크 공간(Linux 인스턴스에만 해당)		Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances

자동 및 수동 모니터링

AWS는 Amazon EC2를 모니터링하는 데 사용할 수 있는 다양한 도구를 제공합니다. 이들 도구 중에는 모니터링을 자동으로 수행하도록 구성할 수 있는 도구도 있지만, 수동 작업이 필요한 도구도 있습니다.

항목

- [자동 모니터링 도구 \(p. 338\)](#)
- [수동 모니터링 도구 \(p. 339\)](#)

자동 모니터링 도구

다음과 같은 자동 모니터링 도구를 사용하여 Amazon EC2를 관찰하고 문제 발생 시 보고를 받을 수 있습니다.

• [System Status Checks] - 인스턴스를 사용하는 데 필요한 AWS 시스템을 모니터링하여 올바르게 작동 중인지 확인합니다. 이러한 확인에서는 복구 시 AWS 개입이 필요한 인스턴스 관련 문제를 찾아냅니다. 시스템 상태 확인이 실패하는 경우, AWS에서 문제를 해결할 때까지 기다리거나, 인스턴스를 중지했다가 다시 시작하거나 종료하고 교체하는 등의 방법으로 사용자가 문제를 직접 해결할 수도 있습니다. 시스템 상태 확인이 실패하게 되는 문제의 예를 들면 다음과 같습니다.

- 네트워크 연결 끊김
- 시스템 전원 중단
- 물리적 호스트의 소프트웨어 문제
- 물리적 호스트의 하드웨어 문제 네트워크 도달 가능성 개선

자세한 내용은 [인스턴스 상태 확인 \(p. 340\)](#) 섹션을 참조하십시오.

• [Instance Status Checks] - 개별 인스턴스에 대한 소프트웨어 및 네트워크 구성은 모니터링합니다. 이러한 확인에서는 복구 시 사용자의 개입이 필요한 문제를 찾아냅니다. 인스턴스 상태 확인이 실패할 경우 일반적으로 사용자는 인스턴스를 재부팅하거나 운영 체제를 수정하는 등의 방법으로 문제를 직접 해결해야 합니다. 인스턴스 상태 확인이 실패하게 되는 문제의 예를 들면 다음과 같습니다.

- 시스템 상태 확인 실패
- 네트워크 구성 또는 시작 구성이 잘못됨
- 메모리가 모두 사용됨
- 파일 시스템 손상
- 호환되지 않는 커널

자세한 내용은 [인스턴스 상태 확인 \(p. 340\)](#) 섹션을 참조하십시오.

• [Amazon CloudWatch Alarms] - 지정하는 기간 동안 단일 지표를 관찰하고 특정 기간 동안 지정된 임계값을 기준으로 지표의 값에 따라 하나 이상의 작업을 수행합니다. 이 작업은 Amazon Simple Notification Service(Amazon SNS) 주제나 Auto Scaling 정책으로 전송되는 알림입니다. 경보는 지속적인 상태 변경에 대해서만 작업을 호출합니다. CloudWatch 경보는 특정한 상태에 있으며 이러한 상태가 변경되어야 하며 지정한 수의 기간에 유지되어야 하므로 간단하게 작업을 호출하지 않습니다. 자세한 내용은 [CloudWatch를 사용해 인스턴스 모니터링하기 \(p. 347\)](#) 섹션을 참조하십시오.

• Amazon CloudWatch Events - AWS 서비스를 자동화하여 시스템 이벤트에 자동으로 응답합니다. AWS 서비스 이벤트는 거의 실시간으로 CloudWatch 이벤트에 전송되며, 전송된 이벤트가 사용자가 정의한 규칙과 일치할 경우 실행할 자동 작업을 지정할 수 있습니다. 자세한 내용은 [What is Amazon CloudWatch Events?](#) 단원을 참조하십시오.

• [Amazon CloudWatch Logs] - Amazon EC2 인스턴스, AWS CloudTrail, 또는 기타 소스의 로그 파일을 모니터링, 저장 및 액세스합니다. 자세한 내용은 [What is Amazon CloudWatch Logs?](#) 단원을 참조하십시오.

• [Amazon EC2 Monitoring Scripts] - 인스턴스에서 메모리, 디스크 및 스왑 파일 사용량을 모니터링할 수 있는 Perl 스크립트입니다. 자세한 내용은 [Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances](#)를 참조하십시오.

• [AWS Management Pack for System Center Operations Manager] - Amazon EC2 인스턴스와 인스턴스 내부에서 실행 중인 Microsoft Windows 또는 Linux 운영 체제를 연결합니다. AWS 관리 팩은 Microsoft System Center Operations Manager의 확장 기능입니다. 데이터 센터의 지정된 컴퓨터(감시자 노드)와 Amazon Web Services API를 사용하여 AWS 리소스에 대한 정보를 원격으로 검색하고 수집합니다. 자세한 내용은 [AWS Management Pack for Microsoft System Center](#) 섹션을 참조하십시오.

수동 모니터링 도구

Amazon EC2 모니터링의 또 한 가지 중요한 부분은 모니터링 스크립트, 상태 확인 및 CloudWatch 경보에 포함되지 않는 항목을 수동으로 모니터링해야 한다는 점입니다. Amazon EC2 및 CloudWatch 콘솔 대시보드에서는 Amazon EC2 환경을 한 눈에 파악할 수 있습니다.

- Amazon EC2 대시보드는 다음 정보를 표시합니다.
 - 리전별 서비스 상태 및 예약된 이벤트
 - 인스턴스 상태
 - 상태 확인
 - 경보 상태
 - 인스턴스 지표 세부 정보([Instances]에서 인스턴스를 선택하고 [Monitoring] 탭 선택)
 - 볼륨 지표 정보(탐색 창에서 [Volumes]을 클릭하고 볼륨을 선택한 다음 [Monitoring] 탭 클릭)
- Amazon CloudWatch 대시보드는 다음 정보를 표시합니다.
 - 현재 경보 및 상태
 - 경보 및 리소스 그래프
 - 서비스 상태

또한 CloudWatch를 사용하여 다음 작업을 수행할 수도 있습니다.

- Amazon EC2 모니터링 데이터를 그래프로 작성하여 문제를 해결하고 추세 파악
- 모든 AWS 리소스 지표 검색 및 찾아보기
- 문제에 대해 알려주는 경보 생성 및 편집
- 경보 및 AWS 리소스를 한 눈에 파악할 수 있는 개요 정보 보기

모니터링 모범 사례

다음과 같은 모니터링 모범 사례를 이용하면 Amazon EC2 모니터링 작업을 보다 효과적으로 수행할 수 있습니다.

- 큰 문제로 확대되기 전에 작은 문제를 미리 방지하도록 모니터링 우선 순위를 지정하십시오.
- 발생하는 경우 다중 지점 실패를 보다 쉽게 디버깅할 수 있도록 AWS 솔루션의 모든 부분에서 모니터링 데이터를 수집하는 모니터링 계획을 생성하고 구현하십시오. 모니터링 계획은 최소한 다음 질문 사항에 대한 해답을 제공해야 합니다.
 - 모니터링의 목표
 - 모니터링할 리소스
 - 이러한 리소스를 모니터링하는 빈도
 - 사용할 모니터링 도구
 - 모니터링 작업을 수행할 사람
 - 문제 발생 시 알려야 할 대상
 - 모니터링 작업을 최대한 자동화하십시오.
 - EC2 인스턴스에서 로그 파일을 확인하십시오.

인스턴스 상태 모니터링

인스턴스의 상태 확인과 예약된 이벤트 정보를 확인하면 인스턴스의 상태를 모니터링할 수 있습니다. 상태 확인은 Amazon EC2에서 실시하는 자동 확인 작업을 통해 정보를 제공합니다. 이러한 자동 검사는 인스턴스에 영향을 미치는 특정 문제가 있을 때 이를 감지합니다. 상태 확인 정보는 Amazon CloudWatch에서 제공되는 데이터와 함께 각 인스턴스에 대한 세부적인 운영 정보를 시작적으로 제공합니다.

인스턴스에서 예약된 특정 이벤트의 상태 또한 확인이 가능합니다. 이런 이벤트는 재부팅이나 만료 등 예약 인스턴스에 대해 설정된 예정 활동에 대한 정보를 제공하며, 각 이벤트의 예약된 시작 시간과 종료 시간을 함께 확인할 수 있습니다.

목차

- [인스턴스 상태 확인 \(p. 340\)](#)
- [예약된 인스턴스 이벤트 \(p. 344\)](#)

인스턴스 상태 확인

인스턴스 상태 모니터링 작업은 Amazon EC2에서 인스턴스의 애플리케이션 실행에 지장을 줄 수 있는 문제를 발견했을 때 빠르게 확인할 수 있는 방법입니다. Amazon EC2에서는 실행 중인 모든 EC2 인스턴스에 대하여 자동 검사를 실행하여 하드웨어 및 소프트웨어 문제를 확인합니다. 이러한 상태 확인 결과를 토대로 식별 가능한 특정 문제를 확인할 수 있습니다. 이러한 데이터는 계획했던 각 인스턴스 상태(*pending*, *running*, *stopping* 등)를 비롯해 Amazon CloudWatch가 모니터링하는 사용 지표(CPU 사용량, 네트워크 트래픽, 디스크 입/출력)에 대한 Amazon EC2 정보를 늘려주는 역할을 합니다.

상태 확인은 매분마다 자동으로 실행되며 통과 또는 실패 상태를 반환합니다. 모든 검사 결과가 통과인 경우 인스턴스의 전체 상태는 [OK]로 표시됩니다. 하나 이상의 검사 결과가 실패인 경우에는 인스턴스의 전체 상태가 [*impaired*]로 표시됩니다. 상태 확인은 Amazon EC2에 내장된 기능으로 비활성화하거나 삭제할 수 없습니다. 그러나 상태 확인 결과를 기준으로 표시되는 경보를 새로 추가하거나 삭제하는 것은 가능합니다. 예를 들어 특정 인스턴스의 상태 확인에서 실패 항목이 있을 때 알리는 경보를 생성할 수 있습니다. 자세한 내용은 [상태 확인 경보 생성 및 수정 \(p. 342\)](#) 섹션을 참조하십시오.

목차

- [상태 확인 유형 \(p. 340\)](#)
- [상태 확인 결과 확인 \(p. 341\)](#)
- [상태 확인 보고 \(p. 342\)](#)
- [상태 확인 경보 생성 및 수정 \(p. 342\)](#)

상태 확인 유형

상태 확인은 시스템 상태 확인과 인스턴스 상태 확인, 두 가지로 제공됩니다.

시스템 상태 확인

인스턴스를 사용하는 데 필요한 AWS 시스템을 모니터링하여 올바르게 작동 중인지 확인합니다. 이러한 확인에서는 복구 시 AWS 개입이 필요한 인스턴스 관련 문제를 찾아냅니다. 하지만 시스템 상태 확인이 실패할 경우에는 AWS에서 문제를 해결할 때까지 기다리거나, 혹은 인스턴스를 중지했다가 다시 시작하거나 종료 후 교체하는 등의 방법으로 사용자가 문제를 직접 해결할 수도 있습니다.

다음은 시스템 상태 확인의 실패 원인이 되는 몇 가지 문제의 예입니다.

- 네트워크 연결 끊김
- 시스템 전원 중단
- 물리적 호스트의 소프트웨어 문제
- 물리적 호스트의 하드웨어 문제 네트워크 도달 가능성 개선

인스턴스 상태 확인

개별 인스턴스에 대한 소프트웨어 및 네트워크 구성은 모니터링합니다. 이러한 확인에서는 복구 시 사용자의 개입이 필요한 문제를 찾아냅니다. 인스턴스 상태 확인이 실패할 경우에는 일반적으로 사용자가 인스턴스를 재부팅하거나 인스턴스 구성은 변경하는 등의 방법으로 문제를 직접 해결해야 합니다.

다음은 인스턴스 상태 확인의 실패 원인이 되는 몇 가지 문제의 예입니다.

- 시스템 상태 확인 실패
- 잘못된 네트워킹 또는 스트리밍 구성
- 메모리가 모두 사용됨
- 파일 시스템 손상
- 호환되지 않는 커널

상태 확인 결과 확인

Amazon EC2는 상태 확인 결과를 확인하고 대응할 수 있는 몇 가지 방법이 있습니다.

콘솔을 사용해 상태 확인

AWS Management Console을 사용해 상태 확인 결과를 확인할 수 있습니다.

콘솔을 사용해 상태 확인 결과를 확인하는 방법

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. [Instances] 페이지의 [Status Checks] 열에는 각 인스턴스의 운영 상태가 목록으로 표시됩니다.
4. 특정 인스턴스의 상태를 보려면 인스턴스를 선택하고 [Status Checks] 탭을 선택합니다.
5. 상태 확인 실패로 표시된 인스턴스가 있고 이 인스턴스가 확인 불가 상태로 20분 넘게 유지될 경우, [AWS Support]를 선택하여 지원을 요청합니다. 시스템 또는 인스턴스 상태 확인 실패 문제를 직접 해결 하려면 [상태 확인에 실패한 인스턴스 문제 해결 \(p. 707\)](#) 섹션을 참조하십시오.

명령줄 또는 API를 사용해 상태 확인

[describe-instance-status](#)(AWS CLI) 명령을 사용해 실행 중인 인스턴스의 상태 확인 결과를 확인할 수 있습니다.

모든 인스턴스 상태를 확인하려면 다음 명령을 사용합니다.

```
aws ec2 describe-instance-status
```

impaired로 표시된 인스턴스의 상태를 모두 확인할 수 있는 명령:

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

단일 인스턴스의 상태를 확인하려면 다음 명령을 사용합니다.

```
aws ec2 describe-instance-status --instance-ids i-1234567890abcdef0
```

또는 다음 명령을 사용합니다.

- [Get-EC2InstanceState](#) (Windows PowerShell용 AWS 도구)
- [DescribeInstanceStatus](#)(Amazon EC2 Query API)

상태 확인 실패로 표시된 인스턴스가 있는 경우, [상태 확인에 실패한 인스턴스 문제 해결 \(p. 707\)](#) 섹션을 참조하십시오.

상태 확인 보고

상태가 impaired로 표시되지 않은 인스턴스임에도 불구하고 문제가 발생하는 경우 피드백을 제공할 수 있습니다. 또는 impaired 상태의 인스턴스와 관련해 문제에 대한 추가 정보를 작성하여 AWS로 전달할 수 있습니다.

전달받은 피드백은 다수의 고객이 경험하는 문제를 식별하는 데 사용되며, 개별적인 계정 문제에 대해 따로 응답을 제공하지는 않습니다. 피드백을 제공해도 해당 인스턴스의 상태 확인 결과는 변동되지 않습니다.

콘솔을 사용해 상태 피드백을 보고하는 방법

콘솔을 사용해 인스턴스 상태를 보고하는 방법

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. 인스턴스를 선택합니다.
4. [Status Checks] 탭을 선택한 후 [Submit feedback]을 선택합니다.
5. [Report Instance Status] 양식을 작성한 후 [Submit]을 선택합니다.

명령줄 또는 API를 사용해 상태 피드백 보고

아래와 같이 `report-instance-status`(AWS CLI) 명령을 사용해 impaired 상태의 인스턴스에 대한 피드백을 전송합니다.

```
aws ec2 report-instance-status --instances i-1234567890abcdef0 --status impaired --reason-codes code
```

또는 다음 명령을 사용합니다.

- `Send-EC2InstanceState`(Windows PowerShell용 AWS 도구)
- `ReportInstanceStatus`(Amazon EC2 Query API)

상태 확인 경보 생성 및 수정

인스턴스 상태와 시스템 상태 경보를 추가하면 인스턴스에서 상태 확인 실패가 발생했을 때 알림을 받을 수 있습니다.

콘솔을 사용해 상태 확인 경보 생성

인스턴스 상태나 시스템 상태를 모니터링하기 위해 기존 인스턴스에 대한 상태 확인 경보를 생성할 수 있습니다. 알림 내용을 이메일로 받아보거나 인스턴스/시스템에서 상태 확인 실패가 발생했을 때 해당 인스턴스를 중단하거나 종료, 또는 복구하도록 경보를 설정할 수 있습니다.

상태 확인 경보 생성

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. 인스턴스를 선택합니다.
4. [Status Checks] 탭을 선택한 후 [Create Status Check Alarm]을 선택합니다.
5. [Send a notification to]를 선택합니다. 기존 SNS 주제를 선택하거나, [Create topic]을 클릭하여 새로운 주제를 생성합니다. 새로운 주제를 생성할 때는 [With these recipients]에 자신의 이메일 주소와 추가 수신자의 주소를 품마로 구분하여 입력합니다.
6. (선택 사항) [Take the action]을 선택한 후 원하는 작업을 선택합니다.
7. [Whenever]에서 알고 싶은 상태 확인을 선택합니다.

Note

이전 단계에서 [Recover this instance]를 선택한 경우에는 [Status Check Failed (System)]를 선택합니다.

8. [For at least]에서 원하는 평가 주기의 개수를 설정하고 [consecutive periods]에서 경보가 실행되고 이메일 전송이 이루어지기 전에 적용할 평가 주기의 시간 단위를 설정합니다.
9. (선택 사항) [Name of alarm]에서 경보의 기본 이름을 다른 이름으로 변경합니다.
10. [Create Alarm]을 선택합니다.

Important

수신자 목록에 이메일 주소를 추가했거나 새 주제를 만든 경우 Amazon SNS에서는 각각의 새 주소로 가입 확인 이메일을 보냅니다. 모든 수신자는 각각 이메일에 포함된 링크를 클릭하여 가입 여부를 확인해야 합니다. 경고 알림은 확인된 주소로만 전송됩니다.

필요한 경우 인스턴스 상태 경보를 수정할 수 있습니다.

상태 확인 경보 수정

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. 인스턴스를 선택하고 [Actions]를 선택한 후 [CloudWatch Monitoring]을 선택하고 [Add/Edit Alarms]를 선택합니다.
4. [Alarm Details] 대화 상자에서 경보 이름을 선택합니다.
5. [Edit Alarm] 대화 상자에서 필요한 설정을 변경한 후 [Save]를 선택합니다.

AWS CLI를 사용해 상태 확인 경보 생성

다음은 인스턴스에서 연속으로 2주기 이상 인스턴스 검사 또는 시스템 상태 확인이 중단되면서 경보가 발생하여 SNS 주제인 `arn:aws:sns:us-west-2:111122223333:my-sns-topic`에 대한 알림 메시지를 게시하는 예제입니다. 지표는 `StatusCheckFailed`입니다.

CLI를 사용해 상태 확인 경보를 생성하는 방법

1. 기존의 SNS 주제를 선택하거나 새로운 주제를 생성합니다. 자세한 내용은 AWS Command Line Interface 사용 설명서에서 [Using the AWS CLI with Amazon SNS](#) 섹션을 참조하십시오.
2. 아래와 같이 `list-metrics` 명령을 사용하여 Amazon EC2에 유효한 Amazon CloudWatch 지표를 확인합니다.

```
aws cloudwatch list-metrics --namespace AWS/EC2
```

3. 아래와 같이 `put-metric-alarm` 명령을 사용하여 경보를 생성합니다.

```
aws cloudwatch put-metric-alarm --alarm-name StatusCheckFailed-Alarm-for-i-1234567890abcdef0 --metric-name StatusCheckFailed --namespace AWS/EC2 --statistic Maximum --dimensions Name=InstanceId,Value=i-1234567890abcdef0 --unit Count --period 300 --evaluation-periods 2 --threshold 1 --comparison-operator GreaterThanOrEqualToThreshold --alarm-actions arn:aws:sns:us-west-2:111122223333:my-sns-topic
```

참고

- `--period`는 Amazon CloudWatch 지표가 수집되는 시간 주기(초)입니다. 이 예제에서는 60초와 5분을 곱셈하여 300초를 사용합니다.

- `--evaluation-periods`는 지표 값과 기준 값을 비교하도록 설정된 연속 주기의 개수입니다. 이 예제에서는 2를 사용합니다.
- `--alarm-actions`는 경보 트리거 시 실행할 작업 목록입니다. 각 작업은 Amazon 리소스 이름(ARN)으로 지정되어 있습니다. 이 예제에서는 Amazon SNS를 사용해 이메일을 보낼 수 있도록 경보를 구성합니다.

예약된 인스턴스 이벤트

AWS는 재부팅, 종단/시작 또는 만료 등 여러 가지 인스턴스 이벤트를 예약할 수 있습니다. 이러한 이벤트들은 자주 발생하지 않습니다. 예약된 이벤트로 영향을 받는 인스턴스가 존재하는 경우 AWS가 이벤트가 발생하기 전에 시작일과 종료일 등 해당 이벤트의 세부 정보가 포함된 이메일을 AWS 계정에 연동되어 있는 이메일 주소로 전송합니다. 이벤트 기간을 제어할 수 있는 작업은 이벤트에 따라 다릅니다.

예약된 이벤트에 대한 세부 정보를 알 수 있도록 계정의 연락처 정보를 업데이트하려면 [Account Settings](#) 페이지로 이동합니다.

목차

- [예약된 이벤트 유형 \(p. 344\)](#)
- [예약된 이벤트 확인 \(p. 344\)](#)
- [중지 또는 만료 예약된 인스턴스 관련 작업 \(p. 345\)](#)
- [재부팅 예약된 인스턴스 작업 \(p. 346\)](#)
- [인스턴스의 유지 관리 예약 작업 \(p. 347\)](#)

예약된 이벤트 유형

Amazon EC2는 인스턴스에 예약된 이벤트 유형을 다음과 같이 지원합니다.

- [Instance stop]: 인스턴스가 중지됩니다. 인스턴스를 다시 시작하면 새 호스트 컴퓨터로 마이그레이션됩니다. 이러한 유형은 Amazon EBS가 지원하는 인스턴스에만 적용됩니다.
- [Instance retirement]: 인스턴스가 종지되거나 종료됩니다.
- [Reboot]: 인스턴스가 재부팅되거나(인스턴스 재부팅), 혹은 인스턴스의 호스트 컴퓨터가 재부팅됩니다(시스템 재부팅).
- [System maintenance]: 네트워크 또는 전력 유지 관리로 인스턴스가 일시적인 영향을 받을 수 있습니다.

예약된 이벤트 확인

예약된 이벤트에 대한 알림 메시지를 이메일로 받는 것 외에도 예약된 이벤트를 확인할 수 있는 방법이 있습니다.

콘솔을 사용해 인스턴스에 예약된 이벤트를 확인하는 방법

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Events]를 클릭합니다. 연동되어 있는 이벤트와 함께 모든 리소스가 표시됩니다. 표시 방식은 리소스 유형이나 특정 이벤트 유형으로 필터링할 수 있습니다. 또한 리소스를 선택하여 세부 정보를 확인할 수도 있습니다.
3. 또는 탐색 창에서 [EC2 Dashboard]를 선택합니다. 연동되어 있는 이벤트와 함께 모든 리소스가 [Scheduled Events] 아래 표시됩니다.
4. 이때는 해당 리소스의 이벤트까지 표시됩니다. 예를 들어 탐색 창에서 [Instances]를 선택한 후 인스턴스를 하나 선택합니다. 그러면 이 인스턴스에 연동되어 있는 이벤트까지 하단 창에 표시됩니다.

명령줄 또는 API를 사용해 인스턴스에 예약된 이벤트를 확인하는 방법

다음 AWS CLI 명령을 사용합니다.

```
aws ec2 describe-instance-status --instance-id i-1234567890abcdef0
```

다음은 인스턴스 만료 이벤트를 나타내는 예제 출력 화면입니다.

```
{  
    "InstanceStatuses": [  
        {  
            "InstanceState": {  
                "Status": "ok",  
                "Details": [  
                    {  
                        "Status": "passed",  
                        "Name": "reachability"  
                    }  
                ]  
            },  
            "AvailabilityZone": "us-west-2a",  
            "InstanceId": "i-1234567890abcdef0",  
            "InstanceState": {  
                "Code": 16,  
                "Name": "running"  
            },  
            "SystemStatus": {  
                "Status": "ok",  
                "Details": [  
                    {  
                        "Status": "passed",  
                        "Name": "reachability"  
                    }  
                ]  
            },  
            "Events": [  
                {  
                    "Code": "instance-stop",  
                    "Description": "The instance is running on degraded hardware",  
                    "NotBefore": "2015-05-23T00:00:00.000Z"  
                }  
            ]  
        }  
    ]  
}
```

또는 다음 명령을 사용합니다.

- [Get-EC2InstanceState](#) (Windows PowerShell용 AWS 도구)
- [DescribeInstanceState](#)(Amazon EC2 Query API)

중지 또는 만료 예약된 인스턴스 관련 작업

AWS가 인스턴스의 기본 호스트 컴퓨터에서 복구 불가능한 장애를 감지한 경우에는 인스턴스의 루트 디바이스 유형에 따라 인스턴스 종단 또는 종료를 예약합니다. 루트 디바이스가 EBS 볼륨이면 인스턴스 종단이 예약됩니다. 그렇지 않고 루트 디바이스가 인스턴스 스토어 볼륨이면 인스턴스 종료가 예약됩니다. 자세한 내용은 [인스턴스 만료](#) (p. 289) 섹션을 참조하십시오.

Important

인스턴스가 중지되거나 종료되면 인스턴스 스토어 볼륨에 저장되었던 데이터는 모두 삭제됩니다. 여기에는 루트 디바이스가 EBS 볼륨인 인스턴스에 연결된 인스턴스 스토어 볼륨도 포함됩니다. 따라서 인스턴스 스토어 볼륨에서 나중에 필요한 데이터는 인스턴스가 종단 또는 종료되기 전에 반드시 저장하십시오.

Amazon EBS에서 지원되는 인스턴스 작업

인스턴스가 예약 시간에 종단될 때까지 기다릴 수 있습니다. 혹은 직접 인스턴스를 종단한 후 다시 시작하여 새로운 호스트 컴퓨터로 마이그레이션하는 것도 가능합니다. 인스턴스 종단과 종단 후 인스턴스 구성을 변경하는 방법에 대한 자세한 내용은 [인스턴스 종지 및 시작 \(p. 285\)](#) 섹션을 참조하십시오.

인스턴스 스토어에서 지원되는 인스턴스 작업

인스턴스 종료 예약 시간 이전에 가장 최신 AMI에서 생성된 인스턴스로 대체하고 필요한 모든 정보를 대체 인스턴스로 마이그레이션하는 것이 권장됩니다. 작업 후에는 원본 인스턴스를 종료하거나 예약 시간에 종료될 때까지 기다리면 됩니다.

재부팅 예약된 인스턴스 작업

AWS에 업데이트 설치나 호스트 컴퓨터 유지 관리 등의 작업이 필요할 때는 인스턴스 또는 인스턴스의 기본 호스트 컴퓨터가 재부팅되도록 예약할 수 있습니다. 재부팅을 위해 예약된 기존 인스턴스와는 상관없이, 업데이트가 기본 호스트에 이미 적용되어 있으므로 새 인스턴스를 시작하려고 재부팅할 필요는 없습니다.

즉, 재부팅 이벤트의 인스턴스 재부팅 또는 시스템 재부팅 여부를 지정할 수 있습니다.

콘솔에서 예약된 재부팅 이벤트의 유형을 확인하는 방법

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Events]를 선택합니다.
3. 필터에서 [Instance resources]를 선택한 다음 원하는 인스턴스를 선택합니다.
4. 하단 창에서 [Event type]을 찾습니다. 이때 값은 `system-reboot` 또는 `instance-reboot`입니다.

AWS CLI를 사용하여 예약된 재부팅 이벤트의 유형을 확인하는 방법

아래와 같이 `describe-instance-status` 명령을 사용합니다.

```
aws ec2 describe-instance-status --instance-ids i-1234567890abcdef0
```

인스턴스 재부팅 작업

예약된 유지 관리 기간 내에 인스턴스 재부팅이 실행될 때까지 기다릴 수 있습니다. 그 밖에 편리한 시간에 직접 인스턴스를 재부팅하는 것도 가능합니다. 자세한 내용은 [인스턴스 재부팅 \(p. 288\)](#) 섹션을 참조하십시오.

인스턴스 재부팅이 완료되면 인스턴스에 예정되었던 재부팅 이벤트가 즉시 취소되고 이벤트 정보가 갱신됩니다. 기본 호스트 컴퓨터에서 보류되었던 유지 관리가 완료되면 부팅이 완전히 끝난 이후에 인스턴스를 다시 사용할 수 있습니다.

시스템 재부팅 작업

시스템은 직접 재부팅할 수 없습니다. 예약된 유지 관리 기간에 시스템이 재부팅될 때까지 기다리는 것이 좋습니다. 일반적으로 시스템 재부팅은 몇 분 내에 완료되며, 인스턴스의 IP 주소와 DNS 이름도 그대로 유지됩니다. 또한 로컬 인스턴스 스토어 볼륨에 저장된 데이터도 모두 보존됩니다. 시스템 재부팅이 완료되면 인스턴스에 예약된 이벤트가 삭제되며, 인스턴스 소프트웨어가 예상대로 실행되는지 확인할 수 있습니다.

그 밖에도 인스턴스를 다른 시간에 유지 관리해야 하는 경우에는 EBS 기반 인스턴스를 종료한 후 다시 시작하여 새로운 호스트로 마이그레이션하는 것이 가능합니다. 하지만 이때는 로컬 인스턴스 스토어 볼륨에 저장된 데이터가 손실됩니다. 인스턴스 스토어 기반 인스턴스의 경우에는 가장 최근 AMI에서 새로운 인스턴스를 시작할 수 있습니다.

인스턴스의 유지 관리 예약 작업

AWS에 인스턴스의 기본 호스트 컴퓨터에 대한 유지 관리가 필요한 경우에는 인스턴스의 유지 관리 이벤트가 예약됩니다. 유지 관리 유형은 네트워크 유지 관리와 전력 유지 관리, 두 가지입니다.

네트워크 유지 관리 시에는 예약된 인스턴스의 네트워크 연결이 잠시 동안 끊어집니다. 유지 관리가 완료되면 인스턴스의 네트워크 연결이 평소처럼 복구됩니다.

전력 유지 관리 시에는 예약된 인스턴스가 잠시 동안 오프라인 상태로 전환되었다가 재부팅됩니다. 재부팅 이후에도 인스턴스의 모든 구성 설정은 그대로 유지됩니다.

약 몇 분 후에 인스턴스가 재부팅되면 애플리케이션이 정상적으로 작동하는지 확인하도록 합니다. 이때 인스턴스는 더 이상 예약된 이벤트가 없어야 하거나, 혹은 예약했던 이벤트가 [Completed]로 표시됩니다. 인스턴스 상태 간에는 최대 1시간까지 소요될 수 있습니다. 완료된 유지 관리 이벤트는 Amazon EC2 콘솔 대시 보드에 일주일까지 표시됩니다.

Amazon EBS에서 지원되는 인스턴스 작업

예약 시간에 유지 관리가 실행될 때까지 기다릴 수 있습니다. 혹은 직접 인스턴스를 중단한 후 다시 시작하여 새로운 호스트 컴퓨터로 마이그레이션하는 것도 가능합니다. 인스턴스 중단과 중단 후 인스턴스 구성을 변경하는 방법에 대한 자세한 내용은 [인스턴스 종지 및 시작 \(p. 285\)](#) 섹션을 참조하십시오.

인스턴스 스토어에서 지원되는 인스턴스 작업

예약 시간에 유지 관리가 실행될 때까지 기다릴 수 있습니다. 그 밖에 유지 관리 예약 기간에도 정상적인 작업을 지속해야 할 경우에는 가장 최근 AMI에서 대체 인스턴스를 실행한 다음 예약 기간 이전에 필요한 데이터를 모두 대체 인스턴스로 마이그레이션하고 원본 인스턴스를 종료할 수도 있습니다.

CloudWatch를 사용해 인스턴스 모니터링하기

Amazon EC2에서 원시 데이터를 수집하여 읽기 가능하며 실시간에 가까운 측정치로 처리하는 Amazon CloudWatch를 사용해 인스턴스를 모니터링할 수 있습니다. 이러한 통계는 15개월간 기록되므로 기록 정보를 보고 웹 애플리케이션이나 서비스가 어떻게 실행되고 있는지 전체적으로 더 잘 파악할 수 있습니다.

Amazon EC2는 기본적으로 측정치 데이터를 5분 동안 CloudWatch에 전송합니다. 인스턴스에 대한 측정치 데이터를 CloudWatch에 1분 동안 전송하기 위해 해당 인스턴스에 대한 세부 모니터링을 활성화할 수 있습니다. 자세한 내용은 [인스턴스에 대한 세부 모니터링 활성화 또는 비활성화 \(p. 348\)](#) 섹션을 참조하십시오.

Amazon EC2 콘솔에는 Amazon CloudWatch의 원시 데이터를 기초로 하는 일련의 그래프가 표시됩니다. 필요에 따라 콘솔의 그래프 대신에 Amazon CloudWatch에서 인스턴스 데이터를 얻는 것을 선호할 수도 있습니다.

Amazon CloudWatch에 대한 자세한 내용은 [Amazon CloudWatch 사용 설명서](#) 섹션을 참조하십시오.

목차

- [인스턴스에 대한 세부 모니터링 활성화 또는 비활성화 \(p. 348\)](#)
- [인스턴스에 대해 얻을 수 있는 CloudWatch 측정치 나열 \(p. 349\)](#)
- [지표에 대한 통계 구하기 인스턴스에 대한 \(p. 354\)](#)
- [인스턴스에 대한 그래프 지표 \(p. 359\)](#)
- [인스턴스에 대한 CloudWatch 경보 생성 \(p. 359\)](#)

- [인스턴스를 중지, 종료, 재부팅 또는 복구하는 경보 만들기 \(p. 360\)](#)

인스턴스에 대한 세부 모니터링 활성화 또는 비활성화

기본 설정상 인스턴스는 기본 모니터링 기능이 활성화되어 있습니다. 세부 모니터링 활성화를 선택할 수 있습니다. 세부 모니터링을 활성화하면 Amazon EC2 콘솔에 인스턴스에 대한 1분 모니터링 그래프가 표시됩니다. 다음 표에서는 인스턴스에 대한 기본 및 세부 모니터링을 설명합니다.

유형	설명
기본	자동으로 5분 기간 동안 데이터를 무료로 사용할 수 있습니다.
세부	<p>추가 비용을 지불하면 데이터를 1분 동안 사용할 수 있습니다. 이러한 데이터 수준을 얻으려면 인스턴스에 대해 해당 수준을 사용하도록 설정해야 합니다. 세부 모니터링을 활성화한 인스턴스의 경우 유사한 인스턴스 그룹 간에 집계된 데이터를 얻을 수도 있습니다.</p> <p>요금에 대한 자세한 내용은 Amazon CloudWatch 제품 페이지를 참조하십시오.</p>

세부 모니터링 활성화

인스턴스를 시작할 때 또는 인스턴스가 실행 중이거나 중지된 후에 인스턴스에 대한 세부 모니터링을 활성화 할 수 있습니다.

콘솔을 사용해 기존 인스턴스에 대한 세부 모니터링을 활성화하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. 인스턴스를 선택하고 [Actions], [CloudWatch Monitoring], [Enable Detailed Monitoring]을 차례로 선택합니다.
4. [Enable Detailed Monitoring] 대화 상자에서 [Yes, Enable]을 선택합니다.
5. [Close]를 선택합니다.

콘솔을 사용해 인스턴스 시작 시 세부 모니터링을 활성화하려면

AWS Management Console을 사용해 인스턴스를 시작할 때 [Configure Instance Details] 페이지에서 [Monitoring]에 있는 확인란을 선택합니다.

AWS CLI를 사용해 기존 인스턴스에 대한 세부 모니터링을 활성화하려면

다음 `monitor-instances` 명령을 사용하여 지정된 인스턴스에 대한 세부 모니터링을 활성화합니다.

```
aws ec2 monitor-instances --instance-ids i-1234567890abcdef0
```

AWS CLI로 인스턴스 시작 시 세부 모니터링을 활성화하려면

--monitoring 플래그와 함께 `run-instances` 명령을 사용하여 세부 모니터링을 활성화합니다.

```
aws ec2 run-instances --image-id ami-09092360 --monitoring Enabled=true...
```

세부 모니터링 비활성화

인스턴스를 시작할 때 또는 인스턴스가 실행 종이거나 중지된 후에 인스턴스에 대한 세부 모니터링을 비활성화할 수 있습니다.

콘솔을 사용해 세부 모니터링을 비활성화하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. 인스턴스를 선택하고 [Actions], [CloudWatch Monitoring], [Disable Detailed Monitoring]을 차례로 선택합니다.
4. [Disable Detailed Monitoring] 대화 상자에서 [Yes, Disable]을 선택합니다.
5. [Close]를 선택합니다.

AWS CLI를 사용해 세부 모니터링을 비활성화하려면

다음 `unmonitor-instances` 명령을 사용하여 지정된 인스턴스에 대한 세부 모니터링을 비활성화합니다.

```
aws ec2 unmonitor-instances --instance-ids i-1234567890abcdef0
```

인스턴스에 대해 얻을 수 있는 CloudWatch 측정치 나열

Amazon EC2는 측정치를 Amazon CloudWatch로 전송합니다. AWS Management Console, AWS CLI 또는 API를 사용하여 Amazon EC2가 CloudWatch에 전송하는 측정치를 나열할 수 있습니다. 기본적으로 각 데이터 요소는 인스턴스의 이전 5분간 활동을 다룹니다. 세부 모니터링을 활성화한 경우 각 데이터 요소는 이전 1분간 활동을 다룹니다.

이 측정치에 대한 통계를 얻는 방법에 대한 자세한 내용은 [지표에 대한 통계 구하기 인스턴스에 대한 \(p. 354\)](#) 섹션을 참조하십시오.

인스턴스 측정치

지표	설명
CPUCreditUsage	[T2 인스턴스] 인스턴스가 소비한 CPU 크레딧 수입니다. CPU 크레딧 하나는 1분 동안 100%의 사용률로 실행되는 vCPU 1개 또는 이와 동등한 vCPU, 사용률 및 시간의 조합과 동일합니다(예를 들어 2분 동안 50%의 사용률로 실행되는 vCPU 1개 또는 2분 동안 25%의 사용률로 실행되는 vCPU 2개). CPU 크레딧 측정치는 5분 간격으로만 제공됩니다. 5분 이상의 시간을 지정할 경우 Average 통계 대신 Sum 통계를 사용하십시오. 단위: 수
CPUCreditBalance	[T2 인스턴스] 인스턴스에 대해 기본 CPU 사용률 이상으로 버스트가 가능한 CPU 크레딧 수입니다. 크레딧은 측정 이후에는 크레딧 잔고에 보관되고, 만료 이후에는 크레딧 잔고에서 소멸됩니다. 크레딧은 측정 이후 24시간이 지나면 만료됩니다. CPU 크레딧 측정치는 5분 간격으로만 제공됩니다. 단위: 수

지표	설명
CPUUtilization	<p>인스턴스에서 현재 사용 중인 할당된 EC2 컴퓨팅 유닛(ECU)의 비율(%)입니다. 이 측정치는 선택한 인스턴스에서 애플리케이션을 실행하는데 필요한 처리 능력을 식별합니다.</p> <p>Note</p> <p>인스턴스 유형에 따라, 인스턴스에 전체 프로세스 코어가 할당되지 않았을 때 운영 체제의 도구에서 비율이 CloudWatch 보다 낮게 표시될 수 있습니다.</p> <p>단위: 백분율</p>
DiskReadOps	<p>지정된 시간 내에 인스턴스에 사용할 수 있는 모든 인스턴스 스토어 볼륨에서 읽기 작업 완료.</p> <p>Note</p> <p>기간의 평균 IOPS(초당 I/O 작업 수)를 계산하려면 기간의 총 작업 수를 해당 기간의 초 수로 나누십시오.</p> <p>단위: 수</p>
DiskWriteOps	<p>지정된 시간 내에 인스턴스에 사용할 수 있는 모든 인스턴스 스토어 볼륨에 대한 쓰기 작업 완료.</p> <p>Note</p> <p>기간의 평균 IOPS(초당 I/O 작업 수)를 계산하려면 기간의 총 작업 수를 해당 기간의 초 수로 나누십시오.</p> <p>단위: 수</p>
DiskReadBytes	<p>인스턴스에 사용할 수 있는 모든 인스턴스 스토어 볼륨에서 읽은 바이트 수.</p> <p>이 측정치는 애플리케이션이 인스턴스의 하드 디스크에서 읽는 데이터 볼륨을 결정하는 데 사용됩니다. 이를 사용하여 애플리케이션의 속도를 결정할 수 있습니다.</p> <p>단위: 바이트</p>
DiskWriteBytes	<p>인스턴스에 사용할 수 있는 모든 인스턴스 스토어 볼륨에 쓴 바이트 수.</p> <p>이 측정치는 애플리케이션이 인스턴스의 하드 디스크에 쓰는 데이터 볼륨을 결정하는 데 사용됩니다. 이를 사용하여 애플리케이션의 속도를 결정할 수 있습니다.</p> <p>단위: 바이트</p>
NetworkIn	<p>인스턴스가 모든 네트워크 인터페이스에서 받은 바이트 수입니다. 이 측정치는 단일 인스턴스에서 애플리케이션으로 들어오는 네트워크 트래픽의 볼륨을 식별합니다.</p> <p>단위: 바이트</p>

지표	설명
<code>NetworkOut</code>	<p>인스턴스가 모든 네트워크 인터페이스에서 보낸 바이트 수입니다. 이 측정치는 단일 인스턴스에서 애플리케이션으로 나가는 네트워크 트래픽의 볼륨을 측정합니다.</p> <p>단위: 바이트</p>
<code>NetworkPacketsIn</code>	<p>인스턴스가 모든 네트워크 인터페이스에서 받은 패킷 수입니다. 이 측정치는 단일 인스턴스에서 수신 트래픽의 볼륨을 측정하는 기준으로 측정합니다. 기본 모니터링에서만 이 측정치를 사용할 수 있습니다.</p> <p>단위: 수</p> <p>Statistics: Minimum, Maximum, Average</p>
<code>NetworkPacketsOut</code>	<p>인스턴스가 모든 네트워크 인터페이스에서 보낸 패킷 수입니다. 이 측정치는 단일 인스턴스에서 발신 트래픽의 볼륨을 측정하는 기준으로 측정합니다. 기본 모니터링에서만 이 측정치를 사용할 수 있습니다.</p> <p>단위: 수</p> <p>Statistics: Minimum, Maximum, Average</p>
<code>StatusCheckFailed</code>	<p>상태 확인 중 하나가 실패했는지 여부를 보고하는 StatusCheckFailed_Instance 및 StatusCheckFailed_System의 조합입니다. 이 측정치 값은 0(영) 또는 1(일)입니다. 0은 상태 확인이 통과했음을 나타내고, 1은 상태 확인 실패를 나타냅니다.</p> <p>Note</p> <p>상태 확인 측정치는 1분 간격으로 제공됩니다. 새로 시작된 인스턴스의 경우, 인스턴스에서 초기화 상태를 완료해야 상태 확인 측정치 데이터를 얻을 수 있습니다. 인스턴스가 실행 상태가 되고 몇 분 내에 상태 확인 측정치를 얻을 수 있습니다.</p> <p>단위: 수</p>
<code>StatusCheckFailed_Instance</code>	<p>인스턴스가 Amazon EC2 인스턴스 상태 확인을 통과했는지 여부를 마지막으로 보고합니다. 이 측정치 값은 0(영) 또는 1(일)입니다. 0은 상태 확인이 통과했음을 나타내고, 1은 상태 확인 실패를 나타냅니다.</p> <p>Note</p> <p>상태 확인 측정치는 1분 간격으로 제공됩니다. 새로 시작된 인스턴스의 경우, 인스턴스에서 초기화 상태를 완료해야 상태 확인 측정치 데이터를 얻을 수 있습니다. 인스턴스가 실행 상태가 되고 몇 분 내에 상태 확인 측정치를 얻을 수 있습니다.</p> <p>단위: 수</p>

지표	설명
<code>StatusCheckFailed_System</code>	<p>인스턴스가 마지막으로 EC2 시스템 상태 확인을 통과했는지 여부를 보고합니다. 이 측정치 값은 0(영) 또는 1(일)입니다. 0은 상태 확인이 통과했음을 나타내고, 1은 상태 확인 실패를 나타냅니다.</p> <p style="margin-left: 20px;">Note</p> <p style="margin-left: 20px;">상태 확인 측정치는 1분 간격으로 제공됩니다. 새로 시작된 인스턴스의 경우, 인스턴스에서 초기화 상태를 완료해야 상태 확인 측정치 데이터를 얻을 수 있습니다. 인스턴스가 실행 상태가 되고 몇 분 내에 상태 확인 측정치를 얻을 수 있습니다.</p> <p style="margin-left: 20px;">단위: 수</p>
<code>BurstBalance</code>	<p>처리량에 최적화된 HDD(<code>st1</code>) 및 Cold HDD(<code>sc1</code>) 볼륨에만 사용됩니다. 버스트 버킷에서 사용할 수 있는 잔고에 관한 정보를 제공합니다. 볼륨이 활성 상태일 때만 CloudWatch에 데이터가 보고되고, 볼륨이 연결되지 않은 경우에는 데이터가 보고되지 않습니다.</p> <p style="margin-left: 20px;">단위: 백분율</p>

EBS 볼륨에 제공되는 측정치에 대한 자세한 내용은 [Amazon EBS 지표 \(p. 581\)](#) 섹션을 참조하십시오. 스팟 집합에 제공되는 측정치에 대한 자세한 내용은 [스팟 집합에 대한 CloudWatch 측정치 \(p. 234\)](#) 섹션을 참조하십시오.

Amazon EC2 차원

다음 차원을 사용하여 인스턴스에 대해 반환되는 지표를 구체화할 수 있습니다.

차원	설명
<code>AutoScalingGroupName</code>	이 차원은 사용자가 지정된 용량 그룹의 모든 인스턴스에 대해 요청하는 데이터를 필터링합니다. Auto Scaling 그룹은 Auto Scaling을 사용할 경우 사용자가 정의하는 인스턴스 모음입니다. 이 차원은 인스턴스가 이러한 Auto Scaling 그룹에 있을 때 Amazon EC2 측정치에만 사용할 수 있습니다. 세부 또는 기본 모니터링이 설정된 인스턴스에 사용할 수 있습니다.
<code>ImageId</code>	이 차원은 사용자가 이 Amazon EC2 Amazon 머신 이미지(AMI)를 실행하는 모든 인스턴스에 대해 요청하는 데이터를 필터링합니다. 세부 모니터링이 설정된 인스턴스에 사용할 수 있습니다.
<code>InstanceId</code>	이 차원은 사용자가 식별된 인스턴스에 대해 요청하는 데이터만 필터링합니다. 이는 데이터를 모니터링할 정확한 인스턴스를 정확히 식별하는 데 도움이 됩니다.
<code>InstanceType</code>	이 차원은 사용자가 지정된 이 인스턴스 유형으로 실행되는 모든 인스턴스에 대해 요청하는 데이터를 필터링합니다. 이는 실행 중인 인스턴스 유형별로 데이터를 범주화하는 데 도움이 됩니다. 예를 들어, <code>m1.small</code> 인스턴스와 <code>m1.large</code> 인스턴스의 데이터를 비교하여 애플리케이션에 대해 더 높은 비즈니스 가치를 가진 인스턴스를 결정할 수 있습니다. 세부 모니터링이 설정된 인스턴스에 사용할 수 있습니다.

콘솔을 사용해 측정치 나열하기

측정치는 먼저 네임스페이스별로 그룹화된 다음, 각 네임스페이스 내에서 다양한 차원 조합별로 그룹화됩니다. 예를 들어, Amazon EC2에 의해 제공되는 모든 측정치나 인스턴스 ID, 인스턴스 유형, 이미지(AMI) ID 또는 Auto Scaling 그룹별로 제공되는 측정치를 볼 수 있습니다.

범주별로 사용 가능한 지표를 보려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Metrics]를 선택합니다.
3. EC2 측정치 네임스페이스를 선택합니다.
4. 측정치 차원(예: 인스턴스당 메트릭)을 선택합니다.
5. 측정치를 정렬하려면 열 머리글을 사용합니다. 측정치를 그래프로 표시하려면 측정치 옆에 있는 확인란을 선택합니다. 리소스로 필터링하려면 리소스 ID를 선택한 후 [Add to search]를 선택합니다. 측정치로 필터링하려면 측정치 이름을 선택한 후 [Add to search]를 선택합니다.

AWS CLI를 사용해 측정치 나열하기

`list-metrics` 명령을 사용하여 인스턴스에 대한 CloudWatch 측정치를 나열합니다.

Amazon EC2의 모든 측정치를 표시하려면

다음 예제는 Amazon EC2의 모든 측정치를 표시하는 AWS/EC2 네임스페이스를 지정합니다.

```
aws cloudwatch list-metrics --namespace AWS/EC2
```

다음은 예제 출력입니다.

```
{
    "Metrics": [
        {
            "Namespace": "AWS/EC2",
            "Dimensions": [
                {
                    "Name": "InstanceId",
                    "Value": "i-1234567890abcdef0"
                }
            ],
            "MetricName": "NetworkOut"
        },
        {
            "Namespace": "AWS/EC2",
            "Dimensions": [
                {
                    "Name": "InstanceId",
                    "Value": "i-1234567890abcdef0"
                }
            ],
            "MetricName": "CPUUtilization"
        },
        {
            "Namespace": "AWS/EC2",
            "Dimensions": [
                {
                    "Name": "InstanceId",
                    "Value": "i-1234567890abcdef0"
                }
            ],
            "MetricName": "MemoryUtilization"
        }
    ]
}
```

```
        "Value": "i-1234567890abcdef0"
    },
    ],
    "MetricName": "NetworkIn"
},
...
}
```

인스턴스에 대한 모든 측정치를 표시하려면

다음 예제는 지정한 인스턴스의 결과만 보도록 AWS/EC2 네임스페이스와 InstanceId 차원을 지정합니다.

```
aws cloudwatch list-metrics --namespace AWS/EC2 --dimensions
  Name=InstanceId,Value=i-1234567890abcdef0
```

모든 인스턴스에 대한 측정치를 나열하려면

다음 예제는 지정한 측정치의 결과만 보도록 AWS/EC2 네임스페이스와 측정치 이름을 지정합니다.

```
aws cloudwatch list-metrics --namespace AWS/EC2 --metric-name CPUUtilization
```

지표에 대한 통계 구하기 인스턴스에 대한

인스턴스에 대한 CloudWatch 측정치 통계를 볼 수 있습니다.

목차

- 통계 개요 (p. 354)
- 특정 인스턴스에 대한 통계 얻기 (p. 355)
- 여러 인스턴스의 통계 집계하기 (p. 356)
- Auto Scaling 그룹별 통계 집계 (p. 357)
- AMI의 집계 통계 (p. 358)

통계 개요

통계는 지정한 기간에 걸친 메트릭 데이터 집계입니다. CloudWatch에서는 사용자 지정 데이터를 통해 제공되었거나 AWS의 기타 서비스에서 CloudWatch에 제공한 메트릭 데이터 요소를 기반으로 통계를 제공합니다. 집계는 네임스페이스, 메트릭 이름, 차원 및 데이터 요소 측정 단위를 사용하여 지정한 기간에 대해 수행됩니다. 다음 표에서는 사용 가능한 통계에 대해 설명합니다.

통계	설명
Minimum	지정된 기간 중 관찰된 가장 낮은 값입니다. 이 값을 사용하여 애플리케이션에 대한 낮은 볼륨의 활동을 확인할 수 있습니다.
Maximum	지정된 기간 중 관찰된 가장 높은 값입니다. 이 값을 사용하여 애플리케이션에 대한 높은 볼륨의 활동을 확인할 수 있습니다.
Sum	일치하는 메트릭에 대해 제출된 모든 값이 서로 더해진 값입니다. 이 통계는 메트릭의 총 볼륨을 확인할 때 유용할 수 있습니다.
Average	지정된 기간 중 Sum/SampleCount의 값입니다. 이 통계를 Minimum 및 Maximum과 비교하면 메트릭의 전체 범위와 평균 사용량이 Minimum 및 Maximum에 얼마나 근접했는지 확인할 수 있습니다. 이와 같은 비교를 통해 필요에 따라 리소스를 늘리거나 줄어야 하는 시점을 파악할 수 있습니다.

통계	설명
SampleCount	통계 계산에 사용된 데이터 요소의 수(숫자)입니다.
pNN.NN	지정된 백분위 수의 값. 소수점 두 자리까지 사용하여 백분위 수를 지정할 수 있습니다 (예: p95.45).

특정 인스턴스에 대한 통계 얻기

다음 예제는 AWS Management Console 또는 AWS CLI 명령을 사용하여 특정 EC2 인스턴스의 최대 CPU 사용률을 확인하는 방법을 보여 줍니다.

요구 사항

- 인스턴스의 ID가 필요합니다. 인스턴스 ID는 AWS Management Console이나 [describe-instances](#) 명령을 사용하여 확인할 수 있습니다.
- 기본적으로 기본 모니터링이 사용되지만 세부 모니터링을 사용하도록 설정할 수 있습니다. 자세한 내용은 [인스턴스에 대한 세부 모니터링 활성화 또는 비활성화 \(p. 348\)](#) 섹션을 참조하십시오.

콘솔을 사용해 특정 인스턴스에 대한 CPU 사용률을 표시하려면

- <https://console.aws.amazon.com/cloudwatch>에서 CloudWatch 콘솔을 엽니다.
- 탐색 창에서 [Metrics]를 선택합니다.
- EC2 측정치 네임스페이스를 선택합니다.
- 인스턴스당 측정치 차원을 선택합니다.
- 검색 필드에 **cpuutilization**을 입력하고 Enter를 누릅니다. 특정 인스턴스의 행을 선택합니다. 그러면 해당 인스턴스의 [CPUUtilization] 측정치 그래프가 표시됩니다. 그래프 이름을 지정하려면 연필 아이콘을 선택합니다. 시간 범위를 변경하려면 제공되는 값 중 하나를 선택하거나 [custom]을 선택합니다.
- 측정치에 대한 통계 또는 기간을 변경하려면 [Graphed metrics] 탭을 선택합니다. 열 머리글이나 개별 값을 선택한 후 다른 값을 선택합니다.

AWS CLI를 사용해 특정 인스턴스에 대한 CPU 사용률을 얻으려면

다음 [get-metric-statistics](#) 명령을 사용하여, 지정된 기간 및 시간 간격을 사용하는 지정된 인스턴스의 CPUUtilization 측정치를 확인합니다.

```
aws cloudwatch get-metric-statistics --namespace AWS/ElasticComputeCloud --metric-name CPUUtilization --period 3600 \
--statistics Maximum --dimensions Name=InstanceId,Value=i-1234567890abcdef0 \
--start-time 2016-10-18T23:18:00 --end-time 2016-10-19T23:18:00
```

다음은 예제 출력입니다. 각 값은 단일 EC2 인스턴스에 대한 최대 CPU 사용률을 나타냅니다.

```
{
  "Datapoints": [
    {
      "Timestamp": "2016-10-19T00:18:00Z",
      "Maximum": 0.33000000000000002,
      "Unit": "Percent"
    },
    ...
  ]
}
```

```
{  
    "Timestamp": "2016-10-19T03:18:00Z",  
    "Maximum": 99.67000000000002,  
    "Unit": "Percent"  
},  
{  
    "Timestamp": "2016-10-19T07:18:00Z",  
    "Maximum": 0.3400000000000002,  
    "Unit": "Percent"  
},  
{  
    "Timestamp": "2016-10-19T12:18:00Z",  
    "Maximum": 0.3400000000000002,  
    "Unit": "Percent"  
},  
...  
],  
"Label": "CPUUtilization"  
}
```

여러 인스턴스의 통계 집계하기

세부 모니터링이 활성화된 인스턴스에 대해서만 통계를 집계할 수 있습니다. 기본 모니터링을 사용하는 인스턴스는 집계에 포함되지 않습니다. 또한 Amazon CloudWatch에서는 리전 간 데이터는 집계하지 않습니다. 따라서 측정치는 리전 간에 완전히 개별적입니다. 인스턴스 간에 집계된 통계를 얻으려면 1분 기간의 데이터를 제공하는 세부 모니터링(추가 비용 발생)을 활성화해야 합니다.

이 예제는 세부 모니터링을 사용하여 EC2 인스턴스의 평균 CPU 사용량을 확인하는 방법을 보여 줍니다. 지정된 차원이 없으므로 CloudWatch에서는 AWS/EC2 네임스페이스의 모든 차원에 대한 통계를 반환합니다.

Important

AWS 네임스페이스에서 모든 차원을 검색하는 기능은 Amazon CloudWatch에 게시한 사용자 지정 네임스페이스에 대해서는 작동하지 않습니다. 사용자 지정 네임스페이스를 사용하는 경우 데이터 요소가 포함된 통계를 검색하려면 특정 데이터 요소와 연결된 전체 차원 세트를 지정해야 합니다.

인스턴스 전반에 걸친 평균 CPU 사용률을 표시하려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Metrics]를 선택합니다.
3. [EC2] 네임스페이스를 선택한 후 [Across All Instances]를 선택합니다.
4. [CPUUtilization]을 포함하는 행을 선택합니다. 그러면 모든 EC2 인스턴스에 대한 측정치 그래프가 표시됩니다. 그래프 이름을 지정하려면 연필 아이콘을 선택합니다. 시간 범위를 변경하려면 제공되는 값 중 하나를 선택하거나 [custom]을 선택합니다.
5. 측정치에 대한 통계 또는 기간을 변경하려면 [Graphed metrics] 탭을 선택합니다. 열 머리글이나 개별 값을 선택한 후 다른 값을 선택합니다.

인스턴스 전반에 걸친 평균 CPU 사용률을 얻으려면

다음과 같이 `get-metric-statistics` 명령을 사용하여 인스턴스에 대한 평균 [CPUUtilization] 측정치를 확인합니다.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization \  
--period 3600 --statistics "Average" "SampleCount" \  
--start-time 2016-10-11T23:18:00 --end-time 2016-10-12T23:18:00
```

다음은 예제 출력입니다.

```
{  
    "Datapoints": [  
        {  
            "SampleCount": 238.0,  
            "Timestamp": "2016-10-12T07:18:00Z",  
            "Average": 0.038235294117647062,  
            "Unit": "Percent"  
        },  
        {  
            "SampleCount": 240.0,  
            "Timestamp": "2016-10-12T09:18:00Z",  
            "Average": 0.1667083333333332,  
            "Unit": "Percent"  
        },  
        {  
            "SampleCount": 238.0,  
            "Timestamp": "2016-10-11T23:18:00Z",  
            "Average": 0.041596638655462197,  
            "Unit": "Percent"  
        },  
        ...  
    ],  
    "Label": "CPUUtilization"  
}
```

Auto Scaling 그룹별 통계 집계

EC2 인스턴스에 대한 통계를 하나의 Auto Scaling 그룹에 집계할 수 있습니다. Amazon CloudWatch는 리전 전체의 데이터는 집계할 수 없습니다. 측정치는 리전별로 개별적입니다.

이 예제는 하나의 Auto Scaling 그룹에 대해 디스크에 기록되는 총 바이트 수를 확인하는 방법을 보여 줍니다. 이 값은 지정한 Auto Scaling 그룹의 모든 EC2 인스턴스에 대해 24시간 간격으로 1분 기간에 대해 계산됩니다.

콘솔을 사용하여 한 Auto Scaling 그룹의 인스턴스에 대한 DiskWriteBytes를 보려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Metrics]를 선택합니다.
3. [EC2] 네임스페이스를 선택한 후 [By Auto Scaling Group]을 선택합니다.
4. [DiskWriteBytes] 측정치의 행과 특정 Auto Scaling 그룹을 선택합니다. 그러면 해당 Auto Scaling 그룹의 인스턴스에 대한 측정치 그래프가 표시됩니다. 그래프 이름을 지정하려면 연필 아이콘을 선택합니다. 시간 범위를 변경하려면 제공되는 값 중 하나를 선택하거나 [custom]을 선택합니다.
5. 측정치에 대한 통계 또는 기간을 변경하려면 [Graphed metrics] 탭을 선택합니다. 열 머리글이나 개별 값을 선택한 후 다른 값을 선택합니다.

AWS CLI를 사용하여 한 Auto Scaling 그룹의 인스턴스에 대한 DiskWriteBytes를 보려면

다음과 같이 `get-metric-statistics` 명령을 사용합니다.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name DiskWriteBytes --  
period 360 \  
--statistics "Sum" "SampleCount" --dimensions Name=AutoScalingGroupName,Value=my-asg --  
start-time 2016-10-16T23:18:00 --end-time 2016-10-18T23:18:00
```

다음은 예제 출력입니다.

```
{  
    "Datapoints": [  
        {  
            "SampleCount": 238.0,  
            "Timestamp": "2016-10-12T07:18:00Z",  
            "Average": 0.038235294117647062,  
            "Unit": "Percent"  
        },  
        {  
            "SampleCount": 240.0,  
            "Timestamp": "2016-10-12T09:18:00Z",  
            "Average": 0.1667083333333332,  
            "Unit": "Percent"  
        },  
        {  
            "SampleCount": 238.0,  
            "Timestamp": "2016-10-11T23:18:00Z",  
            "Average": 0.041596638655462197,  
            "Unit": "Percent"  
        },  
        ...  
    ],  
    "Label": "CPUUtilization"  
}
```

```
{  
    "SampleCount": 18.0,  
    "Timestamp": "2016-10-19T21:36:00Z",  
    "Sum": 0.0,  
    "Unit": "Bytes"  
},  
{  
    "SampleCount": 5.0,  
    "Timestamp": "2016-10-19T21:42:00Z",  
    "Sum": 0.0,  
    "Unit": "Bytes"  
}  
,  
]  
,  
"Label": "DiskWriteBytes"  
}
```

AMI의 집계 통계

세부 모니터링이 활성화된 인스턴스에 대해 통계를 집계할 수 있습니다. 기본 모니터링을 사용하는 인스턴스는 포함되지 않습니다. Amazon CloudWatch는 리전 전체의 데이터는 집계할 수 없습니다. 측정치는 리전별로 개별적입니다.

인스턴스 간에 집계된 통계를 얻으려면 1분 기간의 데이터를 제공하는 세부 모니터링(추가 비용 발생)을 활성화해야 합니다. 자세한 내용은 [인스턴스에 대한 세부 모니터링 활성화 또는 비활성화 \(p. 348\)](#) 섹션을 참조하십시오.

이 예제는 특정 Amazon 머신 이미지(AMI)를 사용하는 모든 인스턴스의 평균 CPU 사용률을 확인하는 방법을 보여 줍니다. 평균은 1일 기간의 60초 시간 간격에 대한 평균입니다.

콘솔을 사용하여 AMI의 평균 CPU 사용률을 보려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Metrics]를 선택합니다.
3. [EC2] 네임스페이스를 선택한 후 [By Image (AMI) Id]를 선택합니다.
4. [CPUUtilization] 측정치 행과 특정 AMI를 선택합니다. 그러면 지정한 AMI의 그래프가 표시됩니다. 그래프 이름을 지정하려면 연필 아이콘을 선택합니다. 시간 범위를 변경하려면 제공되는 값 중 하나를 선택하거나 [custom]을 선택합니다.
5. 측정치에 대한 통계 또는 기간을 변경하려면 [Graphed metrics] 탭을 선택합니다. 열 머리글이나 개별 값을 선택한 후 다른 값을 선택합니다.

이미지 ID에 대한 평균 CPU 사용률을 얻으려면

다음과 같이 `get-metric-statistics` 명령을 사용합니다.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization --  
period 3600 \  
--statistics Average --dimensions Name=ImageId,Value=ami-3c47a355 --start-  
time 2016-10-10T00:00:00 --end-time 2016-10-11T00:00:00
```

다음은 예제 출력입니다. 각 값은 지정한 AMI를 실행 중인 EC2 인스턴스의 평균 CPU 사용률을 나타냅니다.

```
{  
    "Datapoints": [  
        {  
            "Timestamp": "2016-10-10T07:00:00Z",  
            "Average": 0.04100000000000009,  
            "Unit": "Percent"  
        },  
    ]
```

```
{  
    "Timestamp": "2016-10-10T14:00:00Z",  
    "Average": 0.079579831932773085,  
    "Unit": "Percent"  
},  
{  
    "Timestamp": "2016-10-10T06:00:00Z",  
    "Average": 0.036000000000000011,  
    "Unit": "Percent"  
},  
...  
],  
"Label": "CPUUtilization"  
}
```

인스턴스에 대한 그래프 지표

인스턴스를 시작한 후 Amazon EC2 콘솔을 열고 [Monitoring] 탭에서 인스턴스에 대한 모니터링 그래프를 볼 수 있습니다. 각 그래프는 사용 가능한 Amazon EC2 측정치 중 하나를 기반으로 합니다.

다음과 같은 그래프를 사용할 수 있습니다.

- Average CPU Utilization (Percent)
- Average Disk Reads (Bytes)
- Average Disk Writes (Bytes)
- Maximum Network In (Bytes)
- Maximum Network Out (Bytes)
- Summary Disk Read Operations (Count)
- Summary Disk Write Operations (Count)
- Summary Status (Any)
- Summary Status Instance (Count)
- Summary Status System (Count)

측정치와 이러한 측정치가 그래프에 제공하는 데이터에 대한 자세한 내용은 [인스턴스에 대해 얻을 수 있는 CloudWatch 측정치 나열 \(p. 349\)](#) 섹션을 참조하십시오.

CloudWatch 콘솔을 사용한 측정치 그래프

CloudWatch 콘솔을 사용하여 Amazon EC2 및 기타 AWS 서비스에서 생성한 측정치 데이터의 그래프를 생성할 수도 있습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서에서 [측정치 그래프](#)를 참조하십시오.

인스턴스에 대한 CloudWatch 경보 생성

인스턴스 중 하나의 CloudWatch 측정치를 모니터링하는 CloudWatch 경보를 만들 수 있습니다. 지표가 지정된 임계값에 도달하면 CloudWatch에서 자동으로 알림을 보냅니다. Amazon EC2 콘솔이나 CloudWatch 콘솔에 제공된 고급 옵션을 이용해 CloudWatch 경보를 만들 수 있습니다.

CloudWatch 콘솔을 이용하여 경보 생성하기

구체적인 예시는 Amazon CloudWatch 사용 설명서의 [Amazon CloudWatch 경보 생성](#)을 참조하십시오.

Amazon EC2 콘솔을 이용하여 경보 생성하기

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. 인스턴스를 선택합니다.

4. [Monitoring] 탭에서 [Create Alarm]을 선택합니다.
5. [Create Alarm] 페이지에서 다음과 같이 실행합니다.
 - a. [create topic]을 선택합니다. [Send a notification to]에 SNS 주제의 이름을 입력합니다. [With these recipients]에 알림을 수신할 하나 이상의 이메일 주소를 입력합니다.
 - b. 정책에 대한 지표와 기준을 지정합니다. 예를 들어 [Whenever](CPU 평균 사용률)를 기본 설정으로 유지할 수 있습니다. [Is]에서 >=을 선택하고 80%를 입력합니다. [For at least]에 1 consecutive period of 5 minutes를 입력합니다.
 - c. [Create Alarm]을 선택합니다.

인스턴스를 중지, 종료, 재부팅 또는 복구하는 경보 만들기

Amazon CloudWatch 경보 작업을 사용하면 인스턴스를 자동으로 중지, 종료, 재부팅 또는 복구하는 경보를 만들 수 있습니다. 인스턴스를 더 이상 실행할 필요가 없을 때 중지 또는 종료 작업을 사용하여 비용을 절약할 수 있습니다. 재부팅 및 복구 작업을 사용하면 시스템 장애가 발생할 경우 인스턴스를 자동으로 재부팅하거나 새로운 하드웨어로 인스턴스를 복구할 수 있습니다.

생성한 모든 경보 작업은 경보 작업 ARN을 사용합니다. 한 세트의 ARN은 해당 계정에 대해 EC2ActionsAccess IAM 역할을 요구하기 때문에 보안을 강화합니다. 이 IAM 역할을 사용하여 중지, 종료 또는 재부팅 작업을 수행할 수 있습니다. 이전에는 IAM 역할을 사용하여 작업을 실행할 수 없었습니다. 이전 경보 작업 ARN을 사용하는 기존 경보는 이 IAM 역할이 필요하지 않지만, ARN을 변경하여 해당 ARN을 사용하는 기존 경보를 편집할 때 역할을 추가하는 것이 좋습니다.

AWS는 EC2ActionsAccess 역할을 통해 사용자를 대신하여 경보 작업을 수행할 수 있습니다. Amazon EC2 또는 Amazon CloudWatch 콘솔을 사용하여 처음으로 경보 작업을 생성하면 AWS에서 이 역할을 자동으로 생성합니다.

인스턴스를 자동으로 중지하거나 종료해야 하는 경우는 매우 다양합니다. 예를 들어 일정 기간 동안 실행한 다음 작업을 완료하는 일괄 급여 처리 작업 또는 과학적 컴퓨팅 작업 전용 인스턴스가 있을 수 있습니다. 이러한 인스턴스를 유류 상태로 유지하여 비용이 발생하도록 하는 대신 중지하거나 종료하면 비용을 절감할 수 있습니다. 경보 작업 중지와 종료 간의 주요 차이는 나중에 다시 실행해야 하는 경우 중지된 인스턴스는 쉽게 다시 시작할 수 있고 동일한 인스턴스 ID 및 루트 볼륨을 유지할 수 있다는 점입니다. 그러나 종료된 인스턴스를 다시 시작할 수는 없습니다. 대신, 새 인스턴스를 시작해야 합니다.

Amazon CloudWatch에서 제공하는 기본 및 세부 모니터링 측정치(AWS/EC2 네임스페이스)를 비롯한 인스턴스 측정치당 Amazon EC2 및 InstanceId 값이 실행 중인 유효한 Amazon EC2 인스턴스를 참조하는 경우 InstanceId 차원을 포함하는 모든 사용자 지정 측정치에 대해 설정된 경보에 중지, 종료, 재부팅 또는 복구 작업을 추가할 수 있습니다.

콘솔 지원

Amazon EC2 콘솔 또는 CloudWatch 콘솔을 사용하여 경보를 만들 수 있습니다. 이 문서의 절차는 Amazon EC2 콘솔을 사용합니다. CloudWatch 콘솔을 사용하는 절차는 Amazon CloudWatch 사용 설명서의 [인스턴스를 중지, 종료, 재부팅 또는 복구하는 경보 생성](#)을 참조하십시오.

권한

AWS Identity and Access Management(IAM) 사용자인 경우 경보를 만들거나 수정하려면 다음과 같은 권한이 있어야 합니다.

- `ec2:DescribeInstanceStatus` 및 `ec2:DescribeInstances` - Amazon EC2 인스턴스 상태 지표에 대한 모든 경보
- `ec2:StopInstances` - 중지 작업을 수반하는 경보

- `ec2:TerminateInstances` - 종료 작업을 수반하는 경보
- `ec2:DescribeInstanceRecoveryAttribute` 및 `ec2:RecoverInstances` - 복구 작업을 수반하는 경보

읽기/쓰기 권한이 Amazon CloudWatch에 대해서는 있지만 Amazon EC2에 대해서는 없는 경우 경보를 만들 수는 있지만 Amazon EC2 인스턴스에 대해 중지 또는 종료 작업이 수행되지 않습니다. 그러나 이후에 연결된 Amazon EC2 API를 사용하도록 권한을 부여 받은 경우 앞서 만든 경보 작업이 수행됩니다. IAM 권한에 대한 자세한 내용은 IAM 사용 설명서의 [권한 및 정책](#)을 참조하십시오.

IAM 역할을 사용하여 경보 작업으로 인스턴스를 중지하거나 종료하거나 재부팅하려면 `EC2ActionsAccess` 역할만 사용할 수 있습니다. 다른 IAM 역할은 지원되지 않습니다. 다른 IAM 역할을 사용할 경우 인스턴스를 중지하거나 종료하거나 재부팅할 수 없습니다. 그러나 경보 상태는 계속 표시되고 Amazon SNS 알림 또는 Auto Scaling 정책 등의 다른 작업을 수행할 수 있습니다.

목차

- [Amazon CloudWatch 경보에 중지 작업 추가 \(p. 361\)](#)
- [Amazon CloudWatch 경보에 종료 작업 추가 \(p. 362\)](#)
- [Amazon CloudWatch 경보에 재부팅 작업 추가 \(p. 362\)](#)
- [Amazon CloudWatch 경보에 복구 작업 추가 \(p. 363\)](#)
- [Amazon CloudWatch 콘솔을 사용하여 트리거된 경보 및 작업 기록 보기 \(p. 364\)](#)
- [Amazon CloudWatch 경보 작업 시나리오 \(p. 365\)](#)

Amazon CloudWatch 경보에 중지 작업 추가

특정 임계값에 도달한 경우 Amazon EC2 인스턴스를 중지하는 경보를 만들 수 있습니다. 예를 들어 개발 또는 테스트 인스턴스를 실행한 후 종료하는 것을 잊을 수 있습니다. 24시간 동안 평균 CPU 사용률이 10% 아래로 떨어지는 경우 즉, 유휴 상태로 더 이상 사용되지 않는 경우 트리거되는 경보를 만들 수 있습니다. 필요에 맞춰 임계값 및 기간을 조정할 수 있습니다. 또한 경보가 트리거되면 이메일을 받을 수 있도록 Amazon Simple Notification Service(Amazon SNS) 알림을 추가할 수 있습니다.

Amazon EBS 볼륨을 루트 디바이스로 사용하는 인스턴스는 중지하거나 종료할 수 있지만, 인스턴스 스토어를 루트 디바이스로 사용하는 인스턴스는 종료만 할 수 있습니다.

Amazon EC2 콘솔을 사용하여 유휴 인스턴스를 중지하는 경보를 만들려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창의 [INSTANCES]에서 [Instances]를 선택합니다.
3. 인스턴스를 선택합니다. [Monitoring] 탭에서 [Create Alarm]을 선택합니다.
4. [Alarm Details for] 대화 상자에서 [Create Alarm]을 선택합니다.
5. 경보가 트리거될 때 이메일을 받으려면 [Create Alarm for] 대화 상자의 [Send a notification to]에서 기존 Amazon SNS 주제를 선택하거나 [Create Topic]을 선택하여 새 주제를 만들습니다.

새 주제를 만들려면 [Send a notification to]에 주제 이름을 입력한 다음 [With these recipients]에 수신자의 이메일 주소를 쉼표로 구분하여 입력합니다. 경보를 만든 후에는 이 주제에 대한 알림을 받으려면 먼저 수락해야 하는 구독 확인 이메일이 전송됩니다.

6. [Take the action]을 선택한 다음 [Stop this instance] 라디오 버튼을 선택합니다.
7. 메시지가 표시되면 [Create IAM role: EC2ActionsAccess]를 선택하여 IAM 역할을 자동으로 생성합니다. 그러면 경보가 트리거될 때 AWS가 사용자를 대신하여 인스턴스를 자동으로 중지할 수 있습니다.
8. [Whenever]에서 사용하려는 통계를 선택한 다음 측정치를 선택합니다. 이 예에서는 Average 및 CPU Utilization을 선택합니다.
9. [Is]에서 측정치 임계값을 정의합니다. 이 예에서는 10%를 입력합니다.
10. [For at least]에서 경보의 샘플링 기간을 선택합니다. 이 예에서는 1시간짜리 연속 기간 24개를 입력합니다.

11. 경보 이름을 변경하려면 [Name this alarm]에 새 이름을 입력합니다.

경보 이름을 입력하지 않으면 Amazon CloudWatch에서는 이름을 자동으로 생성합니다.

Note

경보 구성은 경보를 만들기 전에 요구사항에 따라 조정하거나 나중에 편집할 수 있습니다. 이러한 구성에는 메트릭, 임계값, 기간, 작업 및 알림 설정이 있습니다. 그러나 경보를 만든 후에는 경보 이름은 편집할 수 없습니다.

12. [Create Alarm]을 선택합니다.

Amazon CloudWatch 경보에 종료 작업 추가

인스턴스에 대해 종료 보호가 비활성화되어 있는 경우에 한해서 특정 임계값에 도달한 경우 EC2 인스턴스를 자동으로 종료하는 경보를 만들 수 있습니다. 예를 들어 인스턴스의 작업 완료 후 해당 인스턴스가 다시 필요 없는 경우 인스턴스를 종료하려고 할 수 있습니다. 나중에 인스턴스를 사용하려는 경우에는 종료하지 말고 중지해야 합니다. 인스턴스에 대한 종료 보호 활성화 및 비활성화에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [인스턴스에 대한 종료 보호 활성화](#)를 참조하십시오.

Amazon EC2 콘솔을 사용하여 유휴 인스턴스를 종료하는 경보를 만들려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창의 [INSTANCES]에서 [Instances]를 선택합니다.
3. 인스턴스를 선택합니다. [Monitoring] 탭에서 [Create Alarm]을 선택합니다.
4. [Alarm Details for] 대화 상자에서 [Create Alarm]을 선택합니다.
5. 경보가 트리거될 때 이메일을 받으려면 [Create Alarm for] 대화 상자의 [Send a notification to]에서 기존 Amazon SNS 주제를 선택하거나 [Create Topic]을 선택하여 새 주제를 만들니다.

새 주제를 만들려면 [Send a notification to]에 주제 이름을 입력한 다음 [With these recipients]에 수신자의 이메일 주소를 쉼표로 구분하여 입력합니다. 경보를 만든 후에는 이 주제에 대한 알림을 받으려면 먼저 수락해야 하는 구독 확인 이메일이 전송됩니다.

6. [Take the action]을 선택한 다음 [Terminate this instance]를 선택합니다.
7. 메시지가 표시되면 [Create IAM role: EC2ActionsAccess]를 선택하여 IAM 역할을 자동으로 생성합니다. 그러면 경보가 트리거될 때 AWS가 사용자를 대신하여 인스턴스를 자동으로 중지할 수 있습니다.
8. [Whenever]에서 사용하려는 통계를 선택한 다음 측정치를 선택합니다. 이 예에서는 Average 및 CPU Utilization을 선택합니다.
9. [Is]에서 측정치 임계값을 정의합니다. 이 예에서는 10%를 입력합니다.
10. [For at least]에서 경보의 샘플링 기간을 선택합니다. 이 예에서는 1시간짜리 연속 기간 24개를 입력합니다.
11. 경보 이름을 변경하려면 [Name this alarm]에 새 이름을 입력합니다.

경보 이름을 입력하지 않으면 Amazon CloudWatch에서는 이름을 자동으로 생성합니다.

Note

경보 구성은 경보를 만들기 전에 요구사항에 따라 조정하거나 나중에 편집할 수 있습니다. 이러한 구성에는 메트릭, 임계값, 기간, 작업 및 알림 설정이 있습니다. 그러나 경보를 만든 후에는 경보 이름은 편집할 수 없습니다.

12. [Create Alarm]을 선택합니다.

Amazon CloudWatch 경보에 재부팅 작업 추가

Amazon EC2 인스턴스를 모니터링하고 인스턴스를 자동으로 재부팅하는 Amazon CloudWatch 경보를 만들 수 있습니다. 재부팅 경보 작업은 인스턴스 상태 확인 오류(복구 경보 작업은 시스템 상태 확인 오류에 적합)

에 권장됩니다. 인스턴스 재부팅은 운영 체제 재부팅과 같습니다. 대부분의 경우 인스턴스를 재부팅하는 데는 몇 분 밖에 걸리지 않습니다. 인스턴스를 재부팅하는 경우 동일한 물리적 호스트에 남아 있으므로 퍼블릭 DNS 이름, 프라이빗 IP 주소 및 인스턴스 스토어 볼륨의 모든 데이터가 유지됩니다.

인스턴스를 재부팅해도 인스턴스를 중지했다가 다시 시작할 때와는 달리 새 인스턴스 청구 시간이 시작되지 않습니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서에서 [인스턴스 재부팅](#)을 참조하십시오.

Important

재부팅 및 복구 작업 간의 경합 조건을 방지하려면 Amazon EC2 인스턴스를 재부팅하는 경보를 만들 때 1분에 대해 경보 임계값을 3으로 설정하는 것이 좋습니다.

Amazon EC2 콘솔을 사용하여 인스턴스를 재부팅하는 경보를 만들려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창의 [INSTANCES]에서 [Instances]를 선택합니다.
3. 인스턴스를 선택합니다. [Monitoring] 탭에서 [Create Alarm]을 선택합니다.
4. [Alarm Details for] 대화 상자에서 [Create Alarm]을 선택합니다.
5. 경보가 트리거될 때 이메일을 받으려면 [Create Alarm for] 대화 상자의 [Send a notification to]에서 기존 Amazon SNS 주제를 선택하거나 [Create Topic]을 선택하여 새 주제를 만듭니다.

새 주제를 만들려면 [Send a notification to]에 주제 이름을 입력하고 [With these recipients]에 수신자의 이메일 주소를 쉼표로 구분하여 입력합니다. 경보를 만든 후에는 이 주제에 대한 알림을 받으려면 먼저 수락해야 하는 구독 확인 이메일이 전송됩니다.

6. [Take the action]을 선택한 다음 [Reboot this instance]를 선택합니다.
7. 메시지가 표시되면 [Create IAM role: EC2ActionsAccess]를 선택하여 IAM 역할을 자동으로 생성합니다. 그러면 경보가 트리거될 때 AWS가 사용자를 대신하여 인스턴스를 자동으로 중지할 수 있습니다.
8. [Whenever]에서 Status Check Failed (Instance)를 선택합니다.
9. [For at least]에 2를 입력합니다.
10. [consecutive period(s) of]에서 [1 minute]를 선택합니다.
11. 경보 이름을 변경하려면 [Name of alarm]에 새 이름을 입력합니다.

경보 이름을 입력하지 않으면 Amazon CloudWatch에서는 이름을 자동으로 생성합니다.

12. [Create Alarm]을 선택합니다.

Amazon CloudWatch 경보에 복구 작업 추가

사용자는 Amazon EC2 인스턴스를 모니터링하고 기본 하드웨어 장애나 복구에 AWS 개입이 필요한 문제로 인해 인스턴스가 손상된 경우 인스턴스를 자동으로 복구하는 Amazon CloudWatch 경보를 만들 수 있습니다. 종료한 인스턴스는 복구할 수 없습니다. 복구된 인스턴스는 인스턴스 ID, 프라이빗 IP 주소, 탄력적 IP 주소 및 모든 인스턴스 메타데이터를 포함하여 원본 인스턴스와 동일합니다.

`statusCheckFailed_System` 경보가 트리거되고 복구 작업이 시작되는 경우 경보를 만들고 복구 작업을 연결할 때 선택한 Amazon SNS 주제별로 통지됩니다. 인스턴스 복구 중에 인스턴스를 재부팅할 때 인스턴스가 마이그레이션되고 모든 인 메모리 데이터가 손실됩니다. 프로세스가 완료되면 해당 경보를 위해 구성해둔 SNS 주제로 정보가 게시됩니다. 이 SNS 주제에 가입되어 있는 사람은 누구나 복구 시도 상태와 세부 지침이 포함된 이메일 알림을 받게 됩니다. 복구된 인스턴스에서 인스턴스를 재부팅하라는 메시지가 나타납니다.

시스템 상태 확인이 실패하게 되는 문제의 예를 들면 다음과 같습니다.

- 네트워크 연결 끊김
- 시스템 전원 중단
- 물리적 호스트의 소프트웨어 문제

- 물리적 호스트의 하드웨어 문제 네트워크 도달 가능성 개선

복구 작업은 다음 특성을 지닌 인스턴스에만 지원됩니다.

- C3, C4, M3, M4, R3, R4, T2 또는 X1 인스턴스 유형 사용
- VPC(EC2-Classic 아님)에서 실행
- 공유 테넌시 사용(테넌시 속성이 default로 설정되어 있음)
- EBS 볼륨(인스턴스 스토어 볼륨을 구성하지 않음)만 사용합니다. 자세한 정보는 '['Recover this instance' is disabled](#)' 단원을 참조하십시오.

인스턴스에 퍼블릭 IP 주소가 있는 경우 복구 후에도 해당 퍼블릭 IP 주소를 유지합니다.

Important

재부팅 및 복구 작업 간의 경합 조건을 방지하려면 Amazon EC2 인스턴스를 복구하는 경보를 만들 때 1분에 대해 경보 임계값을 2로 설정하는 것이 좋습니다.

Amazon EC2 콘솔을 사용하여 인스턴스를 복구하는 경보를 만들려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창의 [INSTANCES]에서 [Instances]를 선택합니다.
3. 인스턴스를 선택합니다. [Monitoring] 탭에서 [Create Alarm]을 선택합니다.
4. [Alarm Details for] 대화 상자에서 [Create Alarm]을 선택합니다.
5. 경보가 트리거될 때 이메일을 받으려면 [Create Alarm for] 대화 상자의 [Send a notification to]에서 기존 Amazon SNS 주제를 선택하거나 [Create Topic]을 선택하여 새 주제를 만듭니다.

새 주제를 만들려면 [Send a notification to]에 주제 이름을 입력하고 [With these recipients]에 수신자의 이메일 주소를 쉼표로 구분하여 입력합니다. 경보를 만든 후에는 이 주제에 대한 이메일을 받기 전에 수락해야 하는 구독 확인 이메일이 전송됩니다.

6. [Take the action]을 선택한 다음 [Recover this instance]를 선택합니다.
7. 메시지가 표시되면 [Create IAM role: EC2ActionsAccess]를 선택하여 IAM 역할을 자동으로 생성합니다. 그러면 경보가 트리거될 때 AWS가 사용자를 대신하여 인스턴스를 자동으로 중지할 수 있습니다.
8. [Whenever]에서 Status Check Failed (System)를 선택합니다.
9. [For at least]에 2를 입력합니다.
10. [consecutive period(s) of]에서 [1 minute]를 선택합니다.
11. 경보 이름을 변경하려면 [Name of alarm]에 새 이름을 입력합니다.

경보 이름을 입력하지 않으면 Amazon CloudWatch에서는 이름을 자동으로 생성합니다.

12. [Create Alarm]을 선택합니다.

Amazon CloudWatch 콘솔을 사용하여 트리거된 경보 및 작업 기록 보기

Amazon CloudWatch 콘솔에서 경보 및 작업 기록을 볼 수 있습니다. Amazon CloudWatch에서는 지난 2주 간의 경보 및 작업 기록을 보관합니다.

트리거된 경보 및 작업 기록을 보려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Alarms를 선택합니다.
3. 경보를 선택합니다.

4. [Details] 탭에 최근 상태 변화가 시간 및 지표 값과 함께 표시됩니다.
5. 최신 기록 항목을 보려면 [History] 탭을 선택합니다.

Amazon CloudWatch 경보 작업 시나리오

Amazon EC2 콘솔을 사용하여 특정 조건이 충족되면 Amazon EC2 인스턴스를 중지하거나 종료하는 경보 작업을 만들 수 있습니다. 경보 작업을 설정하는 콘솔 페이지의 다음 화면 캡처에서는 설정에 번호가 표시되어 있습니다. 또한 적절한 작업을 만드는 데 도움이 되도록 시나리오의 설정에도 번호를 표시했습니다.

시나리오 1: 유휴 개발 및 테스트 인스턴스 중지

소프트웨어 개발 및 테스트에 사용된 인스턴스가 한 시간 이상 유휴 상태인 경우 중지하는 경보를 만들습니다.

설정	값
	Stop
	Maximum
	CPUUtilization
	<=
	10%
	60 minutes
	1

시나리오 2: 유휴 인스턴스 중지

인스턴스가 24시간 동안 유휴 상태인 경우 인스턴스를 중지하고 이메일을 보내는 경보를 만들습니다.

설정	값
	Stop and email
	Average
	CPUUtilization
	<=
	5%
	60 minutes
	24

시나리오 3: 트래픽이 비정상적으로 높은 웹 서버에 대해 이메일 보내기

인스턴스가 일일 아웃바운드 네트워크 트래픽인 10GB를 초과하는 경우 이메일을 보내는 경보를 만들습니다.

설정	값
	이메일

설정	값
	Sum
	NetworkOut
	>
	10GB
	1 day
	1

시나리오 4: 트래픽이 비정상적으로 높은 웹 서버 중지

아웃바운드 트래픽이 시간당 1GB를 초과하는 경우 인스턴스를 중지하고 문자 메시지(SMS)를 보내는 경보를 만듭니다.

설정	값
	Stop and send SMS
	Sum
	NetworkOut
	>
	1GB
	1 hour
	1

시나리오 5: 메모리가 부족한 경우 인스턴스 중지

메모리 사용률이 90%에 도달했거나 90%를 초과한 경우 인스턴스를 중지하는 경보를 만듭니다. 그러면 문제 해결을 위해 애플리케이션 로그를 검색할 수 있습니다.

Note

MemoryUtilization 지표는 사용자 지정 지표입니다. MemoryUtilization 지표를 사용하려면 Linux 인스턴스용 Perl 스크립트를 설치해야 합니다. 자세한 내용은 [Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances](#)를 참조하십시오.

설정	값
	Stop
	Maximum
	MemoryUtilization
	>=
	90%
	1 minute

설정	값
	1

시나리오 6: 손상된 인스턴스 중지

5분 간격으로 수행된 연속 3회의 상태 확인에 실패한 인스턴스를 중지하는 경보를 만듭니다.

설정	값
	Stop
	Average
	StatusCheckFailed_System
	>=
	1
	15 minutes
	1

시나리오 7: 일괄 처리 작업이 완료되면 인스턴스 종료

결과 데이터를 더 이상 보내지 않는 경우 일괄 작업을 실행하는 인스턴스를 종료하는 경보를 만듭니다.

설정	값
	Terminate
	Maximum
	NetworkOut
	<=
	100,000 bytes
	5 minutes
	1

CloudWatch 이벤트를 사용한 자동화

Amazon CloudWatch Events는 AWS 서비스를 자동화하여 애플리케이션 가용성 문제나 리소스 변경 같은 시스템 이벤트에 자동으로 응답할 수 있는 기능입니다. AWS 서비스 이벤트는 거의 실시간으로 CloudWatch 이벤트로 전송됩니다. 원하는 이벤트만 표시하도록 간단한 규칙을 정의한 후 규칙과 일치하는 이벤트 발생 시 실행할 자동 작업을 지정할 수 있습니다. 가능한 작업으로는 AWS Lambda 함수 호출, Amazon Kinesis Streams로의 이벤트 릴레이, AWS Step Functions 상태 머신 활성화 등이 있습니다.

다음은 CloudWatch 이벤트를 Amazon EC2에 사용하는 몇 가지 예입니다.

- 새로운 Amazon EC2 인스턴스를 시작할 때마다 Lambda 함수를 활성화합니다.
- Amazon EBS 볼륨을 생성하거나 수정할 때 Amazon SNS 주제를 알립니다.

- 다른 AWS 서비스에서 특정 이벤트 발생 시 Amazon EC2 Run Command를 사용하여 명령을 하나 이상의 Amazon EC2 인스턴스에 전송합니다.

자세한 내용은 [Amazon CloudWatch Events 사용 설명서](#) 섹션을 참조하십시오.

Amazon EC2 Linux 인스턴스의 메모리 및 디스크 메트릭 모니터링

Amazon Elastic Compute Cloud(Amazon EC2) Linux 기반 인스턴스에 대한 Amazon CloudWatch 모니터링 스크립트는 Amazon CloudWatch의 사용자 지정 측정치를 생성하고 사용하는 방법을 보여줍니다. 이 예제는 Linux 인스턴스에 대한 메모리, 스왑 및 디스크 공간 사용률 측정치를 보고하는 완벽하게 작동하는 예로 구성된 Perl 스크립트입니다. [Linux에 대한 Amazon CloudWatch 모니터링 스크립트](#)는 AWS 샘플 코드 라이브러리에서 다운로드할 수 있습니다.

Important

이러한 스크립트는 예일 뿐으로, 있는 그대로 제공되며 지원되지 않습니다.

사용자 지정 지표에 대한 표준 Amazon CloudWatch 사용 요금이 이러한 스크립트 사용에 적용됩니다. 자세한 내용은 [Amazon CloudWatch 요금](#) 페이지를 참조하십시오.

목차

- [지원되는 시스템](#) (p. 368)
- [패키지 내용](#) (p. 368)
- [사전 조건](#) (p. 369)
- [시작하기](#) (p. 370)
- [mon-put-instance-data.pl](#) (p. 371)
- [mon-get-instance-stats.pl](#) (p. 373)
- [콘솔에서 사용자 지정 측정치 보기](#) (p. 375)
- [문제 해결](#) (p. 375)

지원되는 시스템

이러한 모니터링 스크립트는 Linux에서 실행 중인 Amazon EC2 인스턴스에 사용하기 위해 작성된 것입니다. 스크립트는 32비트와 64비트를 모두 포함하여 다음 Amazon 머신 이미지(AMI)를 사용하는 인스턴스에서 테스트되었습니다.

- Amazon Linux 2014.09.2
- Red Hat Enterprise Linux 6.6
- SUSE Linux Enterprise Server 12
- Ubuntu Server 16.04 및 14.04

Windows를 실행 중인 Amazon EC2 인스턴스에서 EC2Config를 사용하여 이 데이터를 CloudWatch Logs로 보내어 메모리 및 디스크 측정치를 모니터링할 수 있습니다. 자세한 내용은 Windows 인스턴스용 Amazon EC2 사용 설명서에서 [Sending Performance Counters to CloudWatch and Logs to CloudWatch Logs](#)를 참조하십시오.

패키지 내용

모니터링 스크립트 패키지는 다음과 같은 파일로 구성됩니다.

- CloudWatchClient.pm – 다른 스크립트에서 Amazon CloudWatch 호출을 간소화하는 공유 Perl 모듈입니다.
- mon-put-instance-data.pl – Amazon EC2 인스턴스에 대한 시스템 메트릭(메모리, 스왑, 디스크 사용률)에 대한 시스템 측정치를 수집하여 Amazon CloudWatch에 전송합니다.
- mon-get-instance-stats.pl – Queries Amazon CloudWatch에 쿼리하여 이 스크립트가 실행된 EC2 인스턴스에 대한 최신 사용률 통계를 표시합니다.
- awscreds.template – 액세스 키 ID 및 보안 액세스 키를 저장하는 AWS 자격 증명의 파일 템플릿입니다.
- LICENSE.txt – Apache 2.0 라이선스가 들어 있는 텍스트 파일입니다.
- NOTICE.txt – 저작권 통지입니다.

사전 조건

일부 Linux 버전에서는 모니터링 스크립트를 실행하려면 먼저 모듈을 추가로 설치해야 합니다.

Amazon Linux AMI

Amazon Linux AMI 2014.03 이상 버전을 사용 중인 경우 Perl 모듈을 추가로 설치해야 합니다.

필요한 패키지를 설치하려면,

1. 인스턴스에 로그온합니다. 자세한 내용은 [Linux 인스턴스에 연결 \(p. 274\)](#) 섹션을 참조하십시오.
2. 명령 프롬프트에서 다음과 같이 패키지를 설치합니다.

```
sudo yum install perl-Switch perl-Datetime perl-Sys-Syslog perl-LWP-Protocol-https
```

Red Hat Enterprise Linux

Perl 모듈을 추가로 설치해야 합니다.

Red Hat Enterprise Linux에 필요한 패키지를 설치하려면,

1. 인스턴스에 로그온합니다. 자세한 내용은 [Linux 인스턴스에 연결 \(p. 274\)](#) 섹션을 참조하십시오.
2. 명령 프롬프트에서 다음과 같이 패키지를 설치합니다.

```
sudo yum install perl-Switch perl-Datetime perl-Sys-Syslog perl-LWP-Protocol-https
perl-Digest-SHA -y
sudo yum install zip unzip
```

SUSE Linux Enterprise Server

Perl 모듈을 추가로 설치해야 합니다.

SUSE에 필요한 패키지를 설치하려면,

1. 인스턴스에 로그온합니다. 자세한 내용은 [Linux 인스턴스에 연결 \(p. 274\)](#) 섹션을 참조하십시오.
2. 명령 프롬프트에서 다음과 같이 패키지를 설치합니다.

```
sudo zypper install perl-Switch perl-Datetime
```

```
sudo zypper install -y "perl(LWP::Protocol::https)"
```

Ubuntu Server

다음과 같이 서버를 구성해야 합니다.

Ubuntu에 필요한 패키지를 설치하려면,

1. 인스턴스에 로그온합니다. 자세한 내용은 [Linux 인스턴스에 연결 \(p. 274\)](#) 섹션을 참조하십시오.
2. 명령 프롬프트에서 다음과 같이 패키지를 설치합니다.

```
sudo apt-get update
sudo apt-get install unzip
sudo apt-get install libwww-perl libdatatime-perl
```

시작하기

다음 단계에서는 EC2 Linux 인스턴스에 대한 CloudWatch 모니터링 스크립트를 다운로드, 압축 해제 및 구성하는 방법을 보여줍니다.

모니터링 스크립트를 다운로드, 설치 및 구성하려면,

1. 명령 프롬프트에서 모니터링 스크립트를 저장할 폴더로 이동하여 다음 명령을 실행하고 스크립트를 다운로드합니다.

```
curl http://aws-cloudwatch.s3.amazonaws.com/downloads/
CloudWatchMonitoringScripts-1.2.1.zip -O
```

2. 다음 명령을 실행하여 다운로드한 모니터링 스크립트를 설치합니다.

```
unzip CloudWatchMonitoringScripts-1.2.1.zip
rm CloudWatchMonitoringScripts-1.2.1.zip
cd aws-scripts-mon
```

3. 다음 옵션 중 하나를 사용하여 스크립트에 CloudWatch 작업 권한이 있는지 확인합니다.
 - AWS Identity and Access Management(IAM) 역할을 인스턴스에 연결하였다면 다음 작업 권한이 부여되는지 확인합니다.
 - cloudwatch:PutMetricData
 - cloudwatch:GetMetricStatistics
 - cloudwatch>ListMetrics
 - ec2:DescribeTags
 - 자격 증명 파일에서 AWS 자격 증명을 지정합니다. 먼저 다음과 같이 모니터링 스크립트가 포함된 `awscreds.template` 파일을 `awscreds.conf`로 복사합니다.

```
cp awscreds.template awscreds.conf
```

다음 내용을 이 파일에 추가합니다.

```
AWSAccessKeyId=my-access-key-id
AWSSecretKey=my-secret-access-key
```

AWS 자격 증명을 확인하는 방법에 대한 자세한 내용은 Amazon Web Services 일반 참조에서 [Understanding and Getting Your Security Credentials](#)를 참조하십시오.

mon-put-instance-data.pl

이 스크립트는 현재 시스템에 대한 메모리, 스왑 및 디스크 공간 사용량 데이터를 수집합니다. 그런 다음 Amazon CloudWatch를 원격으로 호출하여 수집한 데이터를 사용자 지정 측정치로 보고합니다.

옵션

이름	설명
--mem-util	MemoryUtilization 측정치를 수집하여 % 단위로 보고합니다. 이 옵션은 애플리케이션 및 운영 체제에서 할당한 메모리만 보고하고 캐시 및 버퍼 메모리는 제외합니다.
--mem-used	MemoryUsed 측정치를 수집하여 전송합니다. 이때, MB 단위로 보고합니다. 이 옵션은 애플리케이션 및 운영 체제에서 할당한 메모리만 보고하고 캐시 및 버퍼 메모리는 제외합니다.
--mem-avail	MemoryAvailable 측정치를 수집하여 전송합니다. 이때, MB 단위로 보고합니다. 이 옵션은 애플리케이션 및 운영 체제에서 사용할 수 있는 메모리를 보고합니다.
--swap-util	SwapUtilization 측정치를 수집하여 전송합니다. 이때, % 단위로 보고합니다.
--swap-used	SwapUsed 측정치를 수집하여 전송합니다. 이때, MB 단위로 보고합니다.
--disk-path=PATH	보고할 디스크를 선택합니다. PATH는 보고해야 할 파일 시스템의 마운트 지점 또는 마운트 지점에 있는 모든 파일을 지정할 수 있습니다. 디스크를 여러 개 선택하는 경우 각 디스크에 대해 --disk-path=PATH를 지정합니다. / 및 /home에 마운트된 파일 시스템의 디스크를 선택하려면 다음 매개 변수를 사용하십시오. --disk-path=/ --disk-path=/home
--disk-space-util	선택한 디스크의 DiskSpaceUtilization 측정치를 수집하여 전송합니다. 이 측정치는 % 단위로 보고됩니다. 참고로, 이 스크립트로 계산되는 디스크 사용률 측정치는 df -k -l 명령으로 계산한 값과 다릅니다. df -k -l 명령으로 계산한 값이 더 유용하다고 생각하면 스크립트에서 계산 값을 변경할 수 있습니다.
--disk-space-used	선택한 디스크의 DiskSpaceUsed 측정치를 수집하여 전송합니다. 기본적으로 이 측정치는 GB 단위로 보고됩니다. Linux 운영 체제의 예약된 디스크 공간으로 인해 사용된 디스크 공간 및 사용 가능한 디스크 공간이 총 디스크 공간에 정확하게 더해지지 않을 수 있습니다.
--disk-space-avail	선택한 디스크의 DiskSpaceAvailable 측정치를 수집하여 전송합니다. 이 측정치는 GB 단위로 보고됩니다.

이름	설명
	Linux 운영 체제의 예약된 디스크 공간으로 인해 사용된 디스크 공간 및 사용 가능한 디스크 공간이 총 디스크 공간에 정확하게 더해지지 않을 수 있습니다.
--memory-units=UNITS	메모리 사용량을 보고할 단위를 지정합니다. 단위를 지정하지 않으면 메모리는 MB 단위로 보고됩니다. UNITS는 바이트, KB, MB, GB 중 하나입니다.
--disk-space-units=UNITS	디스크 공간 사용량을 보고할 단위를 지정합니다. 단위를 지정하지 않으면 디스크 공간은 GB 단위로 보고됩니다. UNITS는 바이트, KB, MB, GB 중 하나입니다.
--aws-credential-file=PATH	AWS 자격 증명이 들어 있는 파일의 위치를 제공합니다. 이 매개 변수는 --aws-access-key-id 및 --aws-secret-key 매개 변수와 함께 사용할 수 없습니다.
--aws-access-key-id=VALUE	호출자를 식별하는 데 사용할 AWS 액세스 키 ID를 지정합니다. --aws-secret-key 옵션과 함께 사용해야 합니다. 이 옵션은 --aws-credential-file 매개 변수와 함께 사용하지 마십시오.
--aws-secret-key=VALUE	CloudWatch에 대한 요청에 서명하는 데 사용할 AWS 보안 액세스 키를 지정합니다. --aws-access-key-id 옵션과 함께 사용해야 합니다. 이 옵션은 --aws-credential-file 매개 변수와 함께 사용하지 마십시오.
--aws-iam-role=VALUE	AWS 자격 증명을 제공하는 데 사용되는 IAM 역할을 지정합니다. =VALUE 값이 필요합니다. 자격 증명을 지정하지 않으면 EC2 인스턴스와 연결된 기본 IAM 역할이 적용됩니다. IAM 역할은 하나만 사용할 수 있습니다. IAM 역할이 없거나 IAM 역할이 두 개 이상 있는 경우 스크립트에서는 오류를 반환합니다. 이 옵션은 --aws-credential-file, --aws-access-key-id 또는 --aws-secret-key 매개 변수와 함께 사용하지 마십시오.
--aggregated[=only]	인스턴스 유형, AMI ID 및 리전 전체에 대한 집계 측정치를 추가합니다. =only 값은 선택 사항입니다. 이 값을 지정하면 스크립트는 집계된 메트릭만 보고합니다.
--auto-scaling[=only]	Auto Scaling 그룹에 대해 집계된 측정치를 추가합니다. =only 값은 선택 사항입니다. 이 값을 지정하면 스크립트는 Auto Scaling 메트릭만 보고합니다. 스크립트를 사용하여 IAM 계정 또는 역할과 연결된 IAM 정책에는 EC2 작업 DescribeTags 을 호출할 권한이 있어야 합니다.
--verify	측정치를 수집하는 스크립트 실행을 테스트하고, 전체 HTTP 요청을 준비하지만 데이터를 보고하기 위해 CloudWatch를 실제로 호출하지는 않습니다. 이 옵션 역시 자격 증명이 제공되었는지 확인합니다. 자세한 정보 표시 모드에서 실행 중인 경우 이 옵션은 CloudWatch에 전송될 측정치를 출력합니다.
--from-cron	Cron에서 스크립트를 호출하는 경우 이 옵션을 사용합니다. 이 옵션을 사용하면 모든 진단 결과가 표시되지 않지만 사용자 계정의 로컬 시스템 로그에 오류 메시지가 전송됩니다.
--verbose	스크립트가 수행한 작업에 대한 자세한 정보를 표시합니다.
--help	사용 정보를 표시합니다.

이름	설명
--version	스크립트의 버전 번호를 표시합니다.

예제

다음 예제는 사용자가 IAM 역할 또는 `awscreds.conf` 파일을 입력하였다고 가정합니다. 그렇지 않으면 이 명령에서 `--aws-access-key-id` 및 `--aws-secret-key` 파라미터를 사용하여 자격 증명을 입력해야 합니다.

CloudWatch에 데이터를 게시하지 않고 간단한 테스트 실행을 수행하려면

```
./mon-put-instance-data.pl --mem-util --verify --verbose
```

사용 가능한 모든 메모리 측정치를 수집한 다음 CloudWatch에 전송하려면

```
./mon-put-instance-data.pl --mem-util --mem-used --mem-avail
```

CloudWatch에 보고되는 메트릭에 대한 Cron 일정을 설정하려면

1. 다음 명령을 사용하여 `crontab` 편집을 시작합니다.

```
crontab -e
```

2. 다음 명령을 추가하여 5분마다 메모리 및 디스크 공간 사용량을 CloudWatch에 보고합니다.

```
*/5 * * * * ~/aws-scripts-mon/mon-put-instance-data.pl --mem-util --disk-space-util --disk-path=/ --from-cron
```

스크립트 오류가 발생하면 스크립트에서는 시스템 로그에 오류 메시지를 기록합니다.

Auto Scaling 그룹에 대한 집계 측정치를 수집하여 개별 인스턴스 측정치를 보고하지 않고 Amazon CloudWatch에 전송하려면

```
./mon-put-instance-data.pl --mem-util --mem-used --mem-avail --auto-scaling=only
```

인스턴스 유형, AMI ID 및 리전에 대한 집계 측정치를 수집하여 개별 인스턴스 측정치를 보고하지 않고 Amazon CloudWatch에 전송하려면

```
./mon-put-instance-data.pl --mem-util --mem-used --mem-avail --aggregated=only
```

mon-get-instance-stats.pl

이 스크립트는 최근 시간 수를 사용하여 입력한 시간 간격 내에서 메모리, 스왑 및 디스크 공간 메트릭에 대한 통계를 CloudWatch에 쿼리합니다. 이 데이터는 스크립트가 실행된 Amazon EC2 인스턴스에 대해 제공됩니다.

옵션

이름	설명
--recent-hours=N	보고할 최근 시간 수를 지정합니다. 이 옵션은 N으로 표시되는데 여기서 N은 정수입니다.

이름	설명
--aws-credential-file=PATH	AWS 자격 증명이 들어 있는 파일의 위치를 제공합니다.
--aws-access-key-id=VALUE	호출자를 식별하는 데 사용할 AWS 액세스 키 ID를 지정합니다. --aws-secret-key 옵션과 함께 사용해야 합니다. 이 옵션은 --aws-credential-file 옵션과 함께 사용하지 마십시오.
--aws-secret-key=VALUE	CloudWatch에 대한 요청에 서명하는 데 사용할 AWS 보안 액세스 키를 지정합니다. --aws-access-key-id 옵션과 함께 사용해야 합니다. 이 옵션은 --aws-credential-file 옵션과 함께 사용하지 마십시오.
--aws-iam-role=VALUE	AWS 자격 증명을 제공하는 데 사용되는 IAM 역할을 지정합니다. =VALUE 값이 필요합니다. 자격 증명을 지정하지 않으면 EC2 인스턴스와 연결된 기본 IAM 역할이 적용됩니다. IAM 역할은 하나만 사용할 수 있습니다. IAM 역할이 없거나 IAM 역할이 두 개 이상 있는 경우 스크립트에서는 오류를 반환합니다. 이 옵션은 --aws-credential-file, --aws-access-key-id 또는 --aws-secret-key 매개 변수와 함께 사용하지 마십시오.
--verify	측정치를 수집하는 스크립트 실행을 테스트하고, 전체 HTTP 요청을 준비하지만 데이터를 보고하기 위해 CloudWatch를 실제로 호출하지는 않습니다. 이 옵션 역시 자격 증명이 제공되었는지 확인합니다. 자세한 정보 표시 모드에서 실행 중인 경우 이 옵션은 CloudWatch에 전송될 측정치를 출력합니다.
--verbose	스크립트가 수행한 작업에 대한 자세한 정보를 표시합니다.
--help	사용 정보를 표시합니다.
--version	스크립트의 버전 번호를 표시합니다.

예

지난 12시간에 대한 사용률 통계를 얻으려면 다음 명령을 실행합니다.

```
./mon-get-instance-stats.pl --recent-hours=12
```

다음은 응답의 예입니다.

```
Instance metric statistics for the last 12 hours.

CPU Utilization
    Average: 1.06%, Minimum: 0.00%, Maximum: 15.22%

Memory Utilization
    Average: 6.84%, Minimum: 6.82%, Maximum: 6.89%

Swap Utilization
    Average: N/A, Minimum: N/A, Maximum: N/A

Disk Space Utilization on /dev/xvda1 mounted as /
    Average: 9.69%, Minimum: 9.69%, Maximum: 9.69%
```

콘솔에서 사용자 지정 측정치 보기

`mon-put-instance-data.pl` 스크립트를 성공적으로 실행한 경우 Amazon CloudWatch 콘솔에서 사용자 정의 측정치를 확인할 수 있습니다.

사용자 지정 측정치를 보려면

1. 앞에서 설명한 대로 `mon-put-instance-data.pl`을 실행합니다.
2. <https://console.aws.amazon.com/cloudwatch>에서 CloudWatch 콘솔을 엽니다.
3. View Metrics를 선택합니다.
4. 스크립트에 게시된 사용자 정의 측정치가 접두사 `System/Linux`와 함께 Viewing에 표시됩니다.

문제 해결

CloudWatchClient.pm 모듈은 인스턴스 메타데이터를 로컬 캐시에 저장합니다. 모니터링 스크립트를 실행한 인스턴스에서 AMI를 생성하는 경우 캐시 TTL(기본값: 6시간, Auto Scaling 그룹은 24시간) 내에 이 AMI에서 시작한 인스턴스는 원본 인스턴스의 ID를 사용하여 측정치를 내보냅니다. 캐시 TTL 시간이 지난 후에는 스크립트가 최신 데이터를 검색하고, 모니터링 스크립트는 현재 인스턴스의 ID를 사용합니다. 이를 즉시 수정하려면 다음 명령을 사용하여 캐시에 저장된 데이터를 제거하십시오.

```
rm /var/tmp/aws-mon/instance-id
```

네트워크 및 보안

Amazon EC2는 다음과 같은 네트워크 및 보안 기능을 제공합니다.

기능

- [Amazon EC2 키 페어 \(p. 377\)](#)
- [Linux 인스턴스에 대한 Amazon EC2 보안 그룹 \(p. 385\)](#)
- [Amazon EC2 리소스에 대한 액세스 제어 \(p. 398\)](#)
- [Amazon EC2와 Amazon Virtual Private Cloud \(p. 466\)](#)
- [Amazon EC2인스턴스 IP 어드레싱 \(p. 490\)](#)
- [탄력적 IP 주소 \(p. 505\)](#)
- [탄력적 네트워크 인터페이스 \(p. 512\)](#)
- [배치 그룹 \(p. 527\)](#)
- [EC2 인스턴스에 대한 네트워크 MTU\(최대 전송 단위\) \(p. 530\)](#)
- [Linux에서 향상된 네트워킹 \(p. 533\)](#)

명령줄 도구나 API를 이용하여 Amazon EC2에 액세스하는 경우 액세스 키 ID 및 보안 액세스 키가 필요합니다. 자세한 내용은 [How Do I Get Security Credentials?](#)(출처: Amazon Web Services 일반 참조) 섹션을 참조하십시오.

인스턴스는 EC2-Classic 또는 EC2-VPC 플랫폼에서 시작됩니다. EC2-Classic 또는 기본 VPC에서 시작된 인스턴스에는 퍼블릭 IP 주소가 자동으로 배정됩니다. 기본이 아닌 VPC에서 시작된 인스턴스는 실행 시 퍼블릭 IP 주소를 배정받을 수 있습니다. EC2-Classic 및 EC2-VPC에 대한 자세한 내용은 [지원되는 플랫폼 \(p. 471\)](#) 섹션을 참조하십시오.

인스턴스는 사용자가 제어할 수 없는 이유로 인해 오류가 발생하거나 종료될 수 있습니다. 인스턴스에 오류가 발생하여 대체 인스턴스를 시작한 경우 이 인스턴스는 원본 인스턴스와 다른 퍼블릭 IP 주소를 갖습니다. 그러나 고정 IP 주소가 필요한 애플리케이션인 경우 탄력적 IP 주소를 사용할 수 있습니다.

보안 그룹을 사용하여 인스턴스에 액세스할 수 있는 사용자를 관리할 수 있습니다. 보안 그룹은 인스턴스 접속할 수 있는 허용 프로토콜, 포트, 소스 IP 범위를 지정하는 인바운드 네트워크 방화벽과 유사합니다. 여러 보안 그룹을 생성하고 각 그룹마다 다른 규칙을 할당할 수 있습니다. 그럼 다음 각 인스턴스에 보안 그룹을 1

개 이상 배정하고 규칙을 사용하여 인스턴스에 도달할 수 있는 트래픽을 지정할 수 있습니다. 특정 IP 주소나 특정 보안 그룹만 인스턴스에 액세스할 수 있도록 보안 그룹을 구성할 수 있습니다.

Amazon EC2 키 페어

Amazon EC2는 퍼블릭 키 암호화 기법을 사용하여 로그인 정보를 암호화 및 해독합니다. 공개 키 암호화 기법은 공개 키를 사용하여 암호 등의 데이터를 암호화하고, 수신자가 개인 키를 사용하여 해당 데이터를 해독하는 방식입니다. 퍼블릭 키와 프라이빗 키를 키 페어라고 합니다.

인스턴스에 로그인하려면 키 페어를 만들고, 인스턴스를 시작할 때 키 페어의 이름을 지정하고, 인스턴스에 연결할 때 프라이빗 키를 제공해야 합니다. Linux 인스턴스는 암호가 없으므로 키 페어를 사용하여 SSH를 통해 로그인합니다. Windows 인스턴스에서는 키 페어를 사용하여 관리자 암호를 가져오고 RDP를 사용하여 로그인합니다.

키 페어 만들기

Amazon EC2를 사용하여 키 페어를 만들 수 있습니다. 자세한 내용은 [Amazon EC2를 사용해 키 페어 만들기 \(p. 378\)](#) 섹션을 참조하십시오.

또는 타사 도구를 사용하여 Amazon EC2로 퍼블릭 키를 가져올 수도 있습니다. 자세한 내용은 [Amazon EC2로 사용자의 퍼블릭 키 가져오기 \(p. 378\)](#) 섹션을 참조하십시오.

각 키 페어에는 이름이 필요합니다. 이름은 당연히 기억하기 쉬워야 합니다. Amazon EC2에서 퍼블릭 키는 키 이름으로 지정한 이름에 연결됩니다.

퍼블릭 키는 Amazon EC2에 저장되며 프라이빗 키는 사용자가 저장합니다. 프라이빗 키 소유자는 임의로 로그인 정보를 해독할 수 있으므로 보안된 장소에 프라이빗 키를 저장해 두는 것이 중요합니다.

Amazon EC2에서 사용되는 키는 2048비트 SSH-2 RSA 키입니다. 키 페어는 리전당 최대 5천 개까지 보유할 수 있습니다.

인스턴스 시작 및 인스턴스로의 연결

인스턴스를 시작할 때는 인스턴스에 연결 시 사용하고자 하는 키 페어의 이름을 지정해야 합니다. 인스턴스를 시작할 때 기존 키 페어의 이름을 지정하지 않으면 해당 인스턴스에 연결하지 못합니다. 인스턴스에 연결할 때는 인스턴스 시작 시 지정한 키 페어에 해당하는 프라이빗 키를 지정해야 합니다.

Note

Amazon EC2에는 프라이빗 키의 사본이 보관되지 않으므로, 프라이빗 키를 분실하면 이를 복구할 방법이 전혀 없습니다. 인스턴스 스토어 지원 인스턴스에 대한 프라이빗 키를 분실하는 경우 해당 인스턴스에는 액세스할 수 없으므로 이 인스턴스를 종료하고 새 키 페어를 사용하는 다른 인스턴스를 시작해야 합니다. EBS 기반 Linux 인스턴스용 프라이빗 키를 분실하는 경우 인스턴스에 대한 액세스 권한을 다시 얻을 수 있습니다. 자세한 내용은 [프라이빗 키를 분실했을 때 Linux 인스턴스에 연결하는 방법 \(p. 382\)](#) 섹션을 참조하십시오.

여러 사용자를 위한 키 페어

한 인스턴스에 여러 명의 사용자가 액세스해야 할 경우 인스턴스에 사용자 계정을 추가할 수 있습니다. 자세한 내용은 [Linux 인스턴스의 사용자 계정 관리 \(p. 304\)](#) 섹션을 참조하십시오. 각 사용자에 대해 키 페어를 만든 다음 각 키 페어의 퍼블릭 키 정보를 해당 인스턴스의 각 사용자에 대한 `.ssh/authorized_keys` 파일에 추가합니다. 그런 다음 사용자에게 이 프라이빗 키 파일을 배포하면 됩니다. 이렇게 하면 루트 계정에 사용되는 것과 동일한 프라이빗 키 파일을 여러 사용자에게 배포하지 않아도 됩니다.

목차

- [Amazon EC2를 사용해 키 페어 만들기 \(p. 378\)](#)

- Amazon EC2로 사용자의 퍼블릭 키 가져오기 (p. 378)
- 키 페어에 맞는 퍼블릭 키 검색(Linux) (p. 380)
- 키 페어에 맞는 퍼블릭 키 검색(Windows) (p. 381)
- 키 페어의 지문 확인 (p. 381)
- 키 페어 삭제 (p. 381)
- 프라이빗 키를 분실했을 때 Linux 인스턴스에 연결하는 방법 (p. 382)

Amazon EC2를 사용해 키 페어 만들기

키 페어는 Amazon EC2 콘솔이나 명령줄을 사용하여 만들 수 있습니다. 키 페어를 생성한 후 인스턴스를 시작할 때 키 페어를 지정할 수 있습니다. 실행 중인 인스턴스에 키 페어를 추가하여 다른 사용자가 인스턴스에 연결하도록 할 수도 있습니다. 자세한 내용은 [Linux 인스턴스의 사용자 계정 관리 \(p. 304\)](#) 섹션을 참조하십시오.

Amazon EC2 콘솔을 이용한 키 페어 생성

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창의 NETWORK & SECURITY에서 Key Pairs를 선택합니다.

Tip

탐색 창은 Amazon EC2 콘솔의 왼쪽에 있습니다. 창이 보이지 않는 경우 창이 최소화되었을 수 있으니 화살표를 선택해 확대하십시오.

3. [Create Key Pair]를 선택합니다.
4. [Create Key Pair] 대화 상자의 [Key pair name] 필드에 새 키 페어의 이름을 입력하고 [Create]를 선택합니다.
5. 브라우저에서 프라이빗 키 파일이 자동으로 다운로드됩니다. 기본 파일 이름은 키 페어의 이름으로 지정된 이름이며, 파일 이름 확장명은 .pem입니다. 안전한 장소에 프라이빗 키 파일을 저장합니다.

Important

이때가 사용자가 프라이빗 키 파일을 저장할 수 있는 유일한 기회입니다. 사용자는 인스턴스를 시작할 때 키 페어의 이름을 입력하고, 인스턴스에 연결할 때마다 해당하는 프라이빗 키를 입력해야 합니다.

6. Mac이나 Linux 컴퓨터에서 SSH 클라이언트를 사용하여 Linux 인스턴스에 연결하려면 다음 명령을 사용하여 프라이빗 키 파일의 권한을 설정함으로써 사용자에게만 읽기 권한이 생기도록 합니다.

```
$ chmod 400 my-key-pair.pem
```

명령줄을 이용한 키 페어 만들기

다음 명령 중 하나를 사용할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [create-key-pair\(AWS CLI\)](#)
- [New-EC2KeyPair\(Windows PowerShell용 AWS 도구\)](#)

Amazon EC2로 사용자의 퍼블릭 키 가져오기

Amazon EC2를 사용하지 않고 키 페어를 만들었다면 타사 도구를 사용하여 RSA 키 페어를 만든 후 Amazon EC2로 퍼블릭 키를 가져올 수 있습니다. 예를 들면 ssh-keygen(표준 OpenSSH 설치 시 제공되는 도구)을 사용

용하여 키 페어를 만들 수 있습니다. 또는 Java, Ruby, Python 등 각종 프로그래밍 언어에서 제공하는 표준 라이브러리를 사용하여 RSA 키 페어를 만들어도 됩니다.

Amazon EC2에서는 다음의 형식이 허용됩니다.

- OpenSSH 퍼블릭 키 형식(~/.ssh/authorized_keys 형식)
- Base64 인코딩 DER 형식
- [RFC4716](#)에 지정된 SSH 퍼블릭 키 파일 형식

Amazon EC2에서 DSA 키는 허용되지 않습니다. 키 생성기가 RSA 키를 만들도록 설정되어 있는지 확인합니다.

지원되는 길이: 1024, 2048, 4096

타사 도구를 이용한 키 페어 만들기

1. 타사 도구로 원하는 키 페어를 생성합니다.
2. 퍼블릭 키는 로컬 파일에 저장합니다. 예를 들어, ~/.ssh/my-key-pair.pub(Linux) 또는 c:\keys\my-key-pair.pub(Windows)입니다. 이 파일의 파일 이름 확장자는 중요하지 않습니다.
3. 프라이빗 키는 다른 로컬 파일에 저장하되 확장자는 .pem을 사용해야 합니다. 예를 들어, ~/.ssh/my-key-pair.pem(Linux) 또는 c:\keys\my-key-pair.pem(Windows)입니다. 프라이빗 키 파일은 안전한 장소에 저장합니다. 인스턴스를 시작할 때 키 페어의 이름을 제공하고, 인스턴스에 연결할 때마다 해당 프라이빗 키를 제공해야 합니다.

Amazon EC2 콘솔을 이용하여 키 페어를 가져오는 단계는 다음과 같습니다.

퍼블릭 키 가져오기

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창의 NETWORK & SECURITY에서 Key Pairs를 선택합니다.
3. [Import Key Pair]를 선택합니다.
4. [Import Key Pair] 대화 상자에서 [Browse]를 선택하고 이전에 저장한 퍼블릭 키 파일을 선택합니다. [Key pair name] 필드에 키 페어의 이름을 입력하고 [Import]를 선택합니다.

명령줄을 사용하여 퍼블릭 키 가져오기

다음 명령 중 하나를 사용할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [import-key-pair \(AWS CLI\)](#)
- [Import-EC2KeyPair \(Windows PowerShell용 AWS 도구\)](#)

퍼블릭 키 파일을 가져왔다면 Amazon EC2 콘솔을 사용한 키 페어 가져오기가 완료되었음을 다음과 같이 확인할 수 있습니다.

가져온 키 페어 확인

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 모음에서 키 페어를 만든 리전을 선택합니다.
3. 탐색 창의 NETWORK & SECURITY에서 Key Pairs를 선택합니다.
4. 가져온 키 페어가 화면에 표시된 키 페어 목록에 있는지 확인합니다.

명령줄을 사용하여 키 페어를 보려면

다음 명령 중 하나를 사용할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [describe-key-pairs \(AWS CLI\)](#)
- [Get-EC2KeyPair \(Windows PowerShell용 AWS 도구\)](#)

키 페어에 맞는 퍼블릭 키 검색(Linux)

Linux 인스턴스에서 퍼블릭 키 콘텐츠는 `~/.ssh/authorized_keys` 내 항목에 있습니다. 검색은 부팅 시에 처리되므로 암호가 없어도 안전하게 인스턴스에 액세스할 수 있습니다. 편집기에서 이 파일을 열면 키 페어에 대한 퍼블릭 키를 볼 수 있습니다. 다음은 이름이 `my-key-pair`인 키 페어에 대한 예시 항목입니다. 항목은 퍼블릭 키와 이 키 페어의 이름 순서로 구성됩니다. 예:

```
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4xyyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnB1tntckij7FbtJMxLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0Fzz
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUzofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb7Oz1PnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE my-key-pair
```

`ssh-keygen`을 사용하면 키 페어에 대한 퍼블릭 키를 구할 수 있습니다. 프라이빗 키를 다운로드한 컴퓨터에서 다음 명령을 실행합니다.

```
$ ssh-keygen -y
```

키가 들어 있는 파일을 입력하라는 메시지가 나타나면 `.pem` 파일의 경로를 지정하십시오. 예:

```
/path_to_key_pair/my-key-pair.pem
```

이 명령으로 다음과 같이 퍼블릭 키가 반환됩니다.

```
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4xyyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnB1tntckij7FbtJMxLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0Fzz
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUzofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb7Oz1PnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE
```

이 명령이 실패하는 경우 다음 명령을 실행하여 사용자 자신만 볼 수 있도록 키 페어 파일에 대한 권한이 변경되어 있는지 확인해야 합니다.

```
$ chmod 400 my-key-pair.pem
```

또한, 인스턴스를 시작할 때 지정한 퍼블릭 키는 해당 인스턴스 메타데이터를 통해 확인할 수 있습니다. 인스턴스를 시작할 때 지정한 퍼블릭 키를 보려면 인스턴스에서 다음 명령을 사용하면 됩니다.

```
$ GET http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4xyyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnB1tntckij7FbtJMxLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0Fzz
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUzofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb7Oz1PnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE my-key-pair
```

자세한 내용은 [인스턴스 메타데이터 가져오기 \(p. 321\)](#) 섹션을 참조하십시오.

단, 이 페이지의 마지막 섹션에 설명된 바와 같이 인스턴스에 연결할 때 사용하는 키 페어를 변경하는 경우 새 퍼블릭 키가 표시되는 인스턴스 메타데이터가 업데이트되지 않습니다. 이 경우 사용자가 인스턴스 메타데이터에서 인스턴스를 시작하면 지정한 키 페어에 대한 퍼블릭 키를 계속해서 볼 수 있습니다.

키 페어에 맞는 퍼블릭 키 검색(Windows)

Windows의 경우 PuTTYgen을 사용하면 키 페어에 대한 퍼블릭 키를 구할 수 있습니다. PuTTYgen을 시작하고 [Load]를 클릭한 후 .ppk 또는 .pem 파일을 선택합니다. PuTTYgen에 퍼블릭 키가 표시됩니다.

또한, 인스턴스를 시작할 때 지정한 퍼블릭 키는 해당 인스턴스 메타데이터를 통해 확인할 수 있습니다. 인스턴스를 시작할 때 지정한 퍼블릭 키를 보려면 인스턴스에서 다음 명령을 사용하면 됩니다.

```
$ GET http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnBItnckij7fbtxJMXLvvwJryDUilBMTjytwB+QhYXUMOzce5Pjz5/18SeJtjnV3iAoG/cQk+OFzz
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb7Oz1PnWoyNoqFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE my-key-pair
```

자세한 내용은 [인스턴스 메타데이터 가져오기 \(p. 321\)](#) 섹션을 참조하십시오.

키 페어의 지문 확인

Amazon EC2 콘솔에서 [Key Pairs] 페이지를 보면 [Fingerprint] 열에 키 페어에서 생성된 지문이 표시됩니다. AWS에 의한 지문 계산 값은 키 페어가 AWS와 타사 도구 중 어디서 생성되었는지에 따라 달라집니다. AWS를 사용하여 키 페어를 만든 경우 지문은 SHA-1 해시 함수를 통해 산출됩니다. 타사 도구로 키 페어를 만들고 AWS에 퍼블릭 키를 업로드한 경우이거나 기존 AWS에서 만든 프라이빗 키에서 새 퍼블릭 키를 생성하여 AWS에 업로드한 경우, 지문은 MD5 해시 함수를 통해 산출됩니다.

[Key Pairs] 페이지에 표시된 지문을 사용하면 로컬 컴퓨터에 있는 프라이빗 키가 AWS에 저장된 퍼블릭 키와 일치하는지 확인할 수 있습니다.

AWS를 사용하여 키 페어를 만든 경우 OpenSSL 도구를 사용하면 다음과 같이 프라이빗 키 파일로 지문을 생성할 수 있습니다.

```
$ openssl pkcs8 -in path_to_private_key -inform PEM -outform DER -topk8 -nocrypt | openssl sha1 -c
```

타사 도구를 사용하여 키 페어를 만들고 AWS에 퍼블릭 키를 업로드한 경우, OpenSSL 도구를 사용하면 로컬 컴퓨터에 있는 프라이빗 키 파일로 지문을 생성할 수 있습니다.

```
$ openssl rsa -in path_to_private_key -pubout -outform DER | openssl md5 -c
```

이 경우 출력이 콘솔에 표시된 지문과 일치해야 합니다.

키 페어 삭제

키 페어 삭제는 곧 Amazon EC2의 퍼블릭 키 사본만 삭제하는 것을 의미합니다. 키 페어를 삭제하더라도 컴퓨터에 있는 프라이빗 키나 해당 키 페어를 사용하여 이전에 시작한 임의의 인스턴스에 대한 퍼블릭 키에는 영향이 미치지 않습니다. 삭제된 키 페어를 사용하여 새 인스턴스를 시작할 수는 없지만, 프라이빗 키(.pem) 파일을 계속 보유하고 있다면 삭제된 키 페어를 사용하여 시작한 임의의 인스턴스에 계속해서 연결할 수 있습니다.

Note

Auto Scaling 그룹을 사용 중인 경우(예: Elastic Beanstalk 환경에서 사용), 삭제하려는 키 페어가 시작 구성에서 지정되지 않았는지 확인하십시오. 비정상 인스턴스가 발견될 경우 Auto Scaling에서 대체 인스턴스를 시작하지만, 키 페어를 찾을 수 없으면 인스턴스 시작에 실패합니다.

키 페어는 Amazon EC2 콘솔이나 명령줄을 사용하여 삭제할 수 있습니다.

콘솔을 이용한 키 페어 삭제

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창의 NETWORK & SECURITY에서 Key Pairs를 선택합니다.
3. 키 페어를 선택하고 [Delete]를 선택합니다.
4. 메시지가 나타나면 [Yes]를 선택합니다.

명령줄을 이용한 키 페어 삭제

다음 명령 중 하나를 사용할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [delete-key-pair\(AWS CLI\)](#)
- [Remove-EC2KeyPair\(Windows PowerShell용 AWS 도구\)](#)

Note

인스턴스에서 Linux AMI를 생성한 후 해당 AMI를 사용하여 다른 리전 또는 계정에서 새 인스턴스를 시작하는 경우 새 인스턴스에 원본 인스턴스의 키 쌍이 포함됩니다. 이를 통해 원본 인스턴스와 동일한 프라이빗 키 파일을 사용하여 새 인스턴스에 연결할 수 있습니다. 원하는 텍스트 편집기를 사용해 .ssh/authorized_keys 파일에서 항목을 제거하여 인스턴스에서 이 퍼블릭 키를 제거할 수 있습니다. 인스턴스의 사용자 관리 및 특정 키 쌍을 사용하여 원격 액세스 제공에 대한 자세한 내용은 [Linux 인스턴스의 사용자 계정 관리 \(p. 304\)](#) 섹션을 참조하십시오.

프라이빗 키를 분실했을 때 Linux 인스턴스에 연결하는 방법

EBS 기반 인스턴스용 프라이빗 키를 분실하는 경우 인스턴스에 대한 액세스 권한을 다시 얻을 수 있습니다. 인스턴스를 중지하고 루트 볼륨을 분리하여 다른 인스턴스에 데이터 볼륨으로 연결하여 authorized_keys 파일을 수정하고 해당 볼륨을 원본 인스턴스로 복구한 뒤 인스턴스를 다시 시작합니다. 인스턴스 시작, 연결, 중지에 대한 자세한 내용은 [인스턴스 수명 주기 \(p. 261\)](#)에서 확인하십시오.

인스턴스 스토어 지원 인스턴스의 경우 상기의 절차가 적용되지 않습니다. 인스턴스의 루트 디바이스 유형을 확인하려면 Amazon EC2 콘솔을 열고 [Instances]를 선택하여 인스턴스를 선택하고 세부 정보 창에서 [Root device type]의 값을 확인합니다. 이때 값은 ebs 또는 instance store입니다. 루트 디바이스가 인스턴스 스토어 볼륨인 경우 인스턴스에 연결하려면 사용자에게 프라이빗 키가 있어야 합니다.

필수 조건

새 키 페어는 Amazon EC2 콘솔이나 타사 도구를 사용해 만들 수 있습니다. 새 키 페어의 이름을 잊어버린 프라이빗 키와 동일하게 지정하려면 먼저 기존 키 페어를 삭제해야 합니다.

키 페어가 다른 EBS 기반 인스턴스로의 연결

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.

2. 탐색 창에서 [Instances]를 선택한 후 연결할 인스턴스를 선택합니다. (이후의 내용에서는 이를 원본 인스턴스라고 지칭함)
3. 다음 정보는 이 절차를 완료할 때 필요하므로 저장해야 합니다.
 - 원본 인스턴스의 인스턴스 ID, AMI ID 및 가용 영역을 메모합니다.
 - [Root device] 필드에서 루트 볼륨의 디바이스 이름(예: /dev/sda1 또는 /dev/xvda)을 기록합니다 링크를 선택하고 [EBS ID] 필드에 있는 볼륨 ID(vol-xxxxxxxxxxxxxx)를 메모합니다.
 - [EC2-클래식] 원본 인스턴스에 연결된 탄력적 IP 주소가 있다면, 세부 정보 창에서 [Elastic IP] 필드에 표시된 탄력적 IP 주소를 기록해둡니다.
4. [Actions]를 선택하고 [Instance State]를 선택한 후 [Stop]을 선택합니다. [Stop]이 비활성화되어 있으면 해당 인스턴스가 이미 중지되었거나 루트 디바이스가 인스턴스 스토어 볼륨인 것입니다.

Warning

인스턴스를 중지하면 인스턴스 스토어 볼륨의 데이터가 삭제됩니다. 따라서 인스턴스 스토어 볼륨에 보존하려는 데이터가 있을 경우 영구 스토리지에 백업하십시오.

5. [Launch Instance]를 선택한 후 시작 마법사를 사용하여 다음 옵션으로 임시 인스턴스를 시작합니다.
 - [Choose an AMI] 페이지에서, 원본 인스턴스를 시작할 때와 같은 AMI를 선택합니다. 이 AMI가 표시되지 않는 경우 중지된 인스턴스에서 사용 가능한 AMI를 만들 수 있습니다. 자세한 내용은 [Amazon EBS 지원 Linux AMI 생성 \(p. 81\)](#) 을(를) 참조하십시오.
 - [Choose an Instance Type] 페이지에서 마법사에 의해 자동 선택된 기본 인스턴스 유형을 그대로 유지합니다.
 - [Configure Instance Details] 페이지에서 연결하고자 하는 인스턴스와 같은 가용 영역을 지정합니다. VPC에서 인스턴스를 시작하는 경우 가용 영역에서 서브넷을 선택합니다.
 - Add Tags 페이지에서 인스턴스에 Name=Temporary 태그를 추가하여 임시 인스턴스임을 표시합니다.
 - Review 페이지에서 Launch를 선택합니다. 새 키 쌍을 생성하고, 컴퓨터에서 안전한 위치에 다운로드 한 후 [Launch Instances]를 선택합니다.
6. 탐색 창에서 [Volumes]를 선택하고 원본 인스턴스에 대한 루트 디바이스 볼륨을 선택합니다(전 단계에서 기록해 둔 볼륨 ID). [Actions]를 선택한 후 [Detach Volume]을 선택합니다. 볼륨이 available 상태가 될 때까지 기다리십시오. ([Refresh] 아이콘을 클릭해야 할 수도 있습니다.)
7. 해당 볼륨을 선택한 상태에서 [Actions]를 선택한 후 [Attach Volume]을 선택합니다. 임시 인스턴스의 인스턴스 ID를 선택하고 [Device]에서 지정된 디바이스를 메모한 후(예: /dev/sdf) [Yes, Attach]를 선택합니다.

Note

AWS Marketplace AMI에서 원본 인스턴스를 시작했고 볼륨에 AWS Marketplace 코드가 포함되어 있는 경우 볼륨을 연결하기 전에 먼저 임시 인스턴스를 중지해야 합니다.

8. 임시 인스턴스에 연결합니다.
9. 임시 인스턴스에서 인스턴스에 연결한 볼륨을 마운트해야 해당 파일 시스템에 액세스할 수 있습니다. 예를 들어 디바이스 이름이 /dev/sdf인 경우 다음 명령을 사용하면 /mnt/tempvol이라는 볼륨이 마운트됩니다.

Note

디바이스 이름은 인스턴스에서 다르게 표시될 수 있습니다. 예를 들면 /dev/sdf로 마운트된 디바이스가 인스턴스에서는 /dev/xvdf로 표시되기도 합니다. Red Hat 중 일부 버전(CentOS 등 변형 버전 포함)은 후행 문자가 4자씩 늘어나기도 하며, 이 경우 /dev/sdf가 /dev/xvdk로 변경됩니다.

- a. lsblk 명령을 사용하면 볼륨이 파티셔닝됐는지 여부를 확인할 수 있습니다.

```
[ec2-user ~]$ lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda     202:0    0   8G  0 disk
```

```
##xvda1 202:1      0      8G  0 part /
xvdf   202:80     0    101G  0 disk
##xvdf1 202:81     0    101G  0 part
xvdg   202:96     0     30G  0 disk
```

위의 예에서 `/dev/xvda` 및 `/dev/xvdf`는 파티셔닝된 볼륨이고 `/dev/xvdg`은 파티셔닝되지 않은 것 입니다. 볼륨이 파티셔닝되지 않은 경우 이후의 단계에서 원시 디바이스(`/dev/xvdf`) 대신에 파티션(`/dev/xvdf1`)을 마운트해야 합니다.

- b. 임시 디렉터리를 만들어 볼륨을 마운트합니다.

```
[ec2-user ~]$ sudo mkdir /mnt/tempvol
```

- c. 임시 마운트 지점에 볼륨(파티션)을 마운트하되, 이전에 인식된 볼륨 이름이나 디바이스 이름을 사용합니다.

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

10. 임시 인스턴스에서 다음 명령을 사용하면 임시 인스턴스에 연결할 `authorized_keys`에서 얻은 새 퍼블릭 키로 마운트된 볼륨에서 `authorized_keys`를 업데이트 할 수 있습니다(Ubuntu 인스턴스의 경우 `ubuntu`와 같은 명령어에서 다음과 같이 다른 사용자 이름으로 대치해야 할 수도 있음).

```
[ec2-user ~]$ cp .ssh/authorized_keys /mnt/tempvol/home/ec2-user/.ssh/authorized_keys
```

이렇게 복사가 완료됐다면 다음 단계로 넘어갑니다.

(선택 사항) 사용자가 `/mnt/tempvol`에서 파일을 편집할 권한이 없다면, `sudo`를 사용하여 파일을 업데이트한 후 이 파일에 대한 권한을 확인해야 원본 인스턴스에 로그인할 수 있는지 여부를 확실하게 알 수 있습니다. 파일에 대한 권한을 확인하려면 다음 명령을 사용하십시오.

```
[ec2-user ~]$ sudo ls -l /mnt/tempvol/home/ec2-user/.ssh
total 4
-rw----- 1 222 500 398 Sep 13 22:54 authorized_keys
```

이 예시에서 출력을 보면 `222`가 사용자 ID이고 `500`이 그룹 ID입니다. 곧이어 `sudo`를 사용하여 실패한 복사 명령을 다시 실행합니다.

```
[ec2-user ~]$ sudo cp .ssh/authorized_keys /mnt/tempvol/home/ec2-user/.ssh/
authorized_keys
```

권한이 변경됐는지 확인하려면 다음 명령을 다시 실행합니다.

```
[ec2-user ~]$ sudo ls -l /mnt/tempvol/home/ec2-user/.ssh
```

사용자 ID와 그룹 ID가 변경되었다면 다음 명령을 사용하여 해당 항목을 복구합니다.

```
[ec2-user ~]$ sudo chown 222:500 /mnt/tempvol/home/ec2-user/.ssh/authorized_keys
```

11. 임시 인스턴스에서 연결된 볼륨을 마운트 해제해야 이 볼륨을 원본 인스턴스에 다시 연결할 수 있습니다. 예를 들어 다음 명령을 사용하면 `/mnt/tempvol`에서 볼륨을 마운트 해제할 수 있습니다.

```
[ec2-user ~]$ sudo umount /mnt/tempvol
```

12. Amazon EC2 콘솔에서 사용자가 메모해 둔 ID가 표시된 볼륨을 선택하고 [Actions]를 선택한 후 [Detach Volume]을 선택합니다. 볼륨이 `available` 상태가 될 때까지 기다리십시오. ([Refresh] 아이콘을 클릭해야 할 수도 있습니다.)

-
13. 해당 볼륨을 선택한 상태에서 [Actions], [Attach Volume]을 선택합니다. 원본 인스턴스의 인스턴스 ID를 선택하고, 앞서 원래 루트 디바이스 연결을 위해 메모한 디바이스 이름을 지정한 뒤(/dev/sda1 또는 /dev/xvda) [Yes, Attach]를 선택합니다.

Warning

원래 연결 디바이스와 같은 이름을 지정하지 않으면 원본 인스턴스를 시작할 수 없습니다.
Amazon EC2는 루트 디바이스 볼륨이 sda1 또는 /dev/xvda에 있다고 인식합니다.

14. 원본 인스턴스를 선택하고 [Actions]를 선택한 후 [Instance State]를 선택하고 [Start]를 선택합니다. 인스턴스가 running 상태로 진입했다면 새 키 페어에 대한 프라이빗 키 파일을 사용하여 해당 인스턴스에 연결할 수 있습니다.

Note

새 키 페어와 해당 프라이빗 키 파일의 이름이 원래 키 페어의 이름과 다른 경우 인스턴스에 연결할 때 새 프라이빗 키 파일의 이름을 지정해야 합니다.

15. [EC2-Classic] 원본 인스턴스를 중지하기 전에 이 인스턴스에 탄력적 IP 주소가 연결되어 있는 경우 다음에 따라 이 주소를 해당 인스턴스에 다시 연결해야 합니다.
- 탐색 창에서 [Elastic IPs]를 선택합니다.
 - 이 절차를 시작할 때 메모해 둔 탄력적 IP 주소를 선택합니다.
 - Actions를 선택한 후 Associate address를 선택합니다.
 - 원본 인스턴스의 ID를 선택한 후 [Associate]를 선택합니다.
16. (선택 사항) 임시 인스턴스를 더 이상 사용하지 않는 경우 해당 인스턴스는 종료해도 됩니다. 임시 인스턴스를 선택하고 [Actions]를 선택한 후 [Instance State]를 선택하고 [Terminate]를 선택합니다.

Linux 인스턴스에 대한 Amazon EC2 보안 그룹

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 추가합니다. 언제든지 보안 그룹에 대한 규칙을 수정할 수 있습니다. 새 규칙은 보안 그룹과 연결된 모든 인스턴스에 자동으로 적용됩니다. 트래픽이 인스턴스에 도달하도록 허용할지 여부를 결정할 때 인스턴스와 연결된 모든 보안 그룹에서 모든 규칙을 평가합니다.

Windows 인스턴스에 대한 트래픽을 허용해야 하는 경우 Windows 인스턴스용 Amazon EC2 사용 설명서의 [Amazon EC2 Security Groups for Windows Instances](#) 섹션을 참조하십시오.

항목

- [EC2-Classic의 보안 그룹 \(p. 386\)](#)
- [EC2-VPC의 보안 그룹 \(p. 386\)](#)
- [보안 그룹 규칙 \(p. 386\)](#)
- [기본 보안 그룹 \(p. 388\)](#)
- [사용자 지정 보안 그룹 \(p. 389\)](#)
- [보안 그룹 작업 \(p. 389\)](#)
- [보안 그룹 규칙 참조 \(p. 393\)](#)

보안 그룹으로 총족되지 않는 요구 사항이 있는 경우 보안 그룹을 사용하면서 인스턴스에 대한 자체 방화벽을 유지합니다.

계정은 생성 시기에 따라 일부 리전에서 EC2-Classic을 지원할 수 있습니다. 자세한 내용은 [지원되는 플랫폼 \(p. 471\)](#) 섹션을 참조하십시오. EC2-Classic의 보안 그룹은 EC2-VPC의 보안 그룹과는 별개입니다.

EC2-Classic의 보안 그룹

EC2-Classic을 사용하는 경우 EC2-Classic용으로 특별히 생성된 보안 그룹을 사용해야 합니다. EC2-Classic에서 인스턴스를 시작할 경우 인스턴스와 동일한 리전에서 보안 그룹을 지정해야 합니다. EC2-Classic에서 인스턴스를 시작할 경우 VPC용으로 생성된 보안 그룹을 지정할 수 없습니다.

EC2-Classic에서 인스턴스를 시작한 이후에는 해당 보안 그룹을 변경할 수 없습니다. 그러나 보안 그룹에 규칙을 추가하거나 보안 그룹에서 규칙을 제거할 수 있으며, 보안 그룹과 연결된 모든 인스턴스에 해당 변경 내용이 자동으로 적용됩니다.

EC2-Classic에서는 계정별로 각 리전에 최대 500개의 보안 그룹이 있을 수 있습니다. 최대 500개의 보안 그룹에 인스턴스를 연결하고 보안 그룹에 최대 100개의 규칙을 추가할 수 있습니다.

EC2-VPC의 보안 그룹

EC2-VPC를 사용하는 경우 VPC용으로 특별히 생성된 보안 그룹을 사용해야 합니다. VPC에서 인스턴스를 시작할 경우 해당 VPC의 보안 그룹을 지정해야 합니다. VPC에서 인스턴스를 시작할 경우 EC2-Classic용으로 생성된 보안 그룹을 지정할 수 없습니다. EC2-VPC의 보안 그룹에는 EC2-Classic의 보안 그룹에서 지원하지 않는 추가 기능이 있습니다. 자세한 내용은 Amazon VPC 사용 설명서의 [EC2-Classic과 EC2-VPC에 대한 보안 그룹의 차이점](#) 섹션을 참조하십시오.

VPC에서 인스턴스를 시작한 이후에는 해당 보안 그룹을 변경할 수 있습니다. 보안 그룹은 네트워크 인터페이스와 연결됩니다. 인스턴스의 보안 그룹을 변경하면 기본 네트워크 인터페이스(eth0)와 연결된 보안 그룹이 변경됩니다. 자세한 내용은 [Amazon VPC 사용 설명서](#)의 Changing an Instance's Security Groups 섹션을 참조하십시오. 다른 네트워크 인터페이스와 연결된 보안 그룹을 변경할 수도 있습니다. 자세한 내용은 [보안 그룹 변경](#) (p. 523) 섹션을 참조하십시오.

EC2-VPC의 보안 그룹에는 별도의 제한이 있습니다. 자세한 내용은 [Amazon VPC 사용 설명서](#)의 Amazon VPC 제한을 참조하십시오. EC2-Classic의 보안 그룹은 EC2-VPC의 보안 그룹 제한에 포함되지 않습니다.

IPv6에 대해 VPC를 사용할 수 있습니다. 자세한 내용은 Amazon VPC 사용 설명서의 [VPC에서 IP 주소 지정](#)을 참조하십시오. VPC 보안 그룹에 규칙을 추가하여 인바운드 및 아웃바운드 IPv6 트래픽을 사용할 수 있습니다.

보안 그룹 규칙

보안 그룹의 규칙은 보안 그룹과 연결된 인스턴스에 도달할 수 있는 인바운드 트래픽과 인스턴스에서 나갈 수 있는 아웃바운드 트래픽을 제어합니다.

다음은 보안 그룹 규칙의 특징입니다.

- 기본적으로 보안 그룹은 모든 아웃바운드 트래픽을 허용합니다.
- EC2-Classic 보안 그룹에 대한 아웃바운드 규칙을 변경할 수 없습니다.
- 보안 그룹 규칙은 항상 허용적입니다. 따라서 액세스를 거부하는 규칙을 생성할 수 없습니다.
- 보안 그룹은 상태가 저장됩니다. — 사용자가 인스턴스에서 요청을 전송하면 해당 요청의 응답 트래픽은 인바운드 보안 그룹 규칙에 관계없이 인바운드 흐름이 허용됩니다. VPC 보안 그룹의 경우, 허용된 인바운드 트래픽에 대한 응답은 아웃바운드 규칙에 관계없이 아웃바운드 흐름이 허용됩니다. 자세한 내용은 [연결 추적](#) (p. 387) 섹션을 참조하십시오.
- 언제든지 규칙을 추가하고 제거할 수 있습니다. 변경 내용은 잠시 후에 보안 그룹과 연결된 인스턴스에 자동으로 적용됩니다.

Note

일부 규칙 변경 사항이 미치는 효과는 트래픽의 추적 방법에 따라 다를 수 있습니다. 자세한 내용은 [연결 추적](#) (p. 387) 섹션을 참조하십시오.

- 여러 보안 그룹을 인스턴스와 연결할 경우 각 보안 그룹의 규칙이 유효하게 결합된 단일 규칙 세트가 생성됩니다. 이 규칙 세트를 사용하여 액세스를 허용할지 여부를 결정합니다.

Note

인스턴스에 여러 보안 그룹을 배정할 수 있으므로 인스턴스에 수백 개의 규칙이 적용될 수 있습니다. 이로 인해 인스턴스에 액세스할 때 문제가 발생할 수 있습니다. 규칙을 최대한 간략하게 만드는 것이 좋습니다.

각 규칙에 대해 다음을 지정합니다.

- **프로토콜:** 허용할 프로토콜. 가장 일반적인 프로토콜은 6(TCP), 17(UDP) 및 1(ICMP)입니다.
- **포트 범위:** TCP, UDP 또는 사용자 지정 프로토콜의 경우 허용할 포트의 범위. 단일 포트 번호(예: 22) 또는 포트 번호의 범위(예: 7000-8000)를 지정할 수 있습니다.
- **ICMP 유형 및 코드:** ICMP의 경우, ICMP 유형과 코드.
- **원본 또는 대상:** 트래픽에 대한 원본(인바운드 규칙) 또는 대상(아웃바운드 규칙). 다음 옵션 중 하나를 지정합니다.
 - 개별 IPv4 주소. IPv4 주소 다음에 /32 접두사를 사용해야 합니다(예: 203.0.113.1/32).
 - (VPC만 해당) 개별 IPv6 주소. /128 접두사 길이를 사용해야 합니다(예: 2001:db8:1234:1a00::123/128).
 - CIDR 블록 표기법으로 표시된 IPv4 주소의 범위(예: 203.0.113.0/24).
 - (VPC만 해당) CIDR 블록 표기법으로 표시된 IPv6 주소의 범위(예: 2001:db8:1234:1a00::/64).
- 다른 보안 그룹. 이 옵션을 사용하면 지정된 보안 그룹과 연결된 인스턴스가 이 보안 그룹과 연결된 인스턴스에 액세스할 수 있습니다. 이 보안 그룹에 원본 보안 그룹의 규칙이 추가되지는 않습니다. 다음 보안 그룹 중 하나를 지정할 수 있습니다.
 - 현재 보안 그룹
 - EC2-Classic: 동일한 리전의 EC2-Classic에 대한 다른 보안 그룹.
 - EC2-Classic: 동일한 리전의 다른 AWS 계정에 대한 보안 그룹(AWS 계정 ID를 접두사로 추가, 예: 111122223333/sg-edcd9784)
 - EC2-VPC: VPC 피어링 연결에서 동일한 VPC 또는 피어 VPC에 대한 다른 보안 그룹.

보안 그룹을 규칙의 원본 또는 대상으로 지정할 경우 규칙은 보안 그룹과 연결된 모든 인스턴스에 영향을 줍니다. 유입 트래픽은 퍼블릭 IP 주소 또는 탄력적 IP 주소가 아닌 원본 보안 그룹과 연결된 인스턴스의 프라이빗 IP 주소를 기반으로 허용됩니다. IP 주소에 대한 자세한 내용은 [Amazon EC2 인스턴스 IP 어드레싱 \(p. 490\)](#) 섹션을 참조하십시오. 보안 그룹 규칙이 피어 VPC의 보안 그룹을 참조하고 참조된 보안 그룹 또는 VPC 피어링 연결이 삭제된 경우, 규칙은 무효로 표시됩니다. 자세한 내용은 [Amazon VPC Peering Guide](#)의 Working with Stale Security Group Rules 섹션을 참조하십시오.

특정 포트에 대한 규칙이 여러 개 있는 경우 최대 허용 규칙을 적용합니다. 예를 들어, IP 주소 203.0.113.1의 TCP 포트 22(SSH) 액세스를 허용하는 규칙과 모든 사용자의 TCP 포트 22 액세스를 허용하는 규칙이 있는 경우 모든 사용자가 TCP 포트 22에 액세스할 수 있습니다.

연결 추적

보안 그룹은 연결 추적을 사용해 인스턴스가 송수신하는 트래픽에 대한 정보를 추적합니다. 규칙은 트래픽의 연결 상태를 기반으로 적용되어 해당 트래픽을 허용 또는 거부할지 결정합니다. 이를 통해 보안 그룹은 상태가—저장될 수 있습니다. 인바운드 트래픽에 대한 응답은 아웃바운드 보안 그룹 규칙에 관계없이 인스턴스에서 나가도록 허용되며 반대의 경우도 마찬가지입니다. 예를 들어 인바운드 보안 규칙이 ICMP 트래픽을 허용하는 경우 사용자가 자택 컴퓨터에서 인스턴스로 ICMP ping 명령을 시작하면 연결에 대한 정보(포트 정보 포함)가 추적됩니다. ping 명령에 대한 인스턴스의 응답 트래픽은 새로운 요청이 아니라 설정된 연결로 주적되며, 아웃바운드 보안 그룹 규칙이 아웃바운드 ICMP 트래픽을 제한하더라도 인스턴스에서 나가도록 허용됩니다.

모든 트래픽 흐름이 추적되지는 않습니다. 보안 그룹 규칙이 모든 트래픽(0.0.0.0/0)에 대해 TCP 또는 UDP를 허용하고 다른 방향에서 모든 응답 트래픽(0.0.0.0/0)을 허용하는 대응 규칙이 있을 경우, 해당 트래픽 흐름은 추적되지 않습니다. 그러므로 응답 트래픽이 추적 정보가 아니라 응답 트래픽을 허용하는 인바운드 또

는 아웃바운드를 기반으로 흐름이 허용됩니다. 다음 예의 보안 그룹에는 SSH, HTTP 및 ICMP 트래픽에 대한 특정 인바운드 규칙과 모든 아웃 바운드 트래픽을 허용하는 아웃 바운드 규칙이 있습니다.

인바운드 규칙		
프로토콜 유형	포트 번호	소스 IP
TCP	22(SSH)	203.0.113.1/32
TCP	80(HTTP)	0.0.0.0/0
ICMP	모두	0.0.0.0/0

아웃바운드 규칙		
프로토콜 유형	포트 번호	목적지 IP
모두	모두	0.0.0.0/0

제한적인 인바운드 규칙으로 인해 인스턴스 간에 SSH 트래픽이 추적됩니다. 규칙에 관계없이 ICMP 트래픽은 항상 추적됩니다. 인바운드 및 아웃바운드 규칙이 모든 HTTP 트래픽을 허용하므로 인스턴스 간에 HTTP 트래픽이 추적되지 않습니다.

사용자가 흐름을 허용하는 보안 그룹 규칙을 제거할 때 추적되는 기존 트래픽 흐름이 중단되지 않을 수 있습니다. 대신, 사용자 또는 다른 호스트가 중지할 때 적어도 몇 분(설정된 TCP 연결의 경우 최대 5일) 이상 동안 흐름이 중단됩니다. UDP의 경우, 이를 위해 흐름의 원격 측에서 종료 작업이 필요할 수 있습니다. 흐름을 허용하는 규칙이 제거 또는 수정될 경우 추적되지 않는 트래픽 흐름이 즉시 중단됩니다. 예를 들어 인스턴스로 들어오는 모든 인바운드 SSH 트래픽을 허용하는 규칙을 제거할 경우 기존의 인스턴스와의 SSH 연결이 즉시 삭제됩니다.

TCP, UDP 또는 ICMP 이외의 프로토콜에 대해서는 IP 주소와 프로토콜 번호만 추적됩니다. 인스턴스가 다른 호스트(호스트 B)로 트래픽을 보내고 호스트 B가 원래 요청 또는 응답으로부터 600초 이내에 별도의 요청으로 사용자의 인스턴스에 대해 동일한 유형의 트래픽을 시작할 경우 인스턴스는 인바운드 보안 그룹 규칙에 관계없이 해당 트래픽을 수락합니다(응답 트래픽으로 간주되기 때문).

VPC 보안 그룹의 경우, 보안 그룹 규칙을 제거하는 즉시 트래픽이 중단되도록 하거나 모든 인바운드 트래픽이 방화벽 규칙에 따르도록 하려면 서브넷의 네트워크 ACL을 사용할 수 있습니다. 네트워크 ACL은 상태 비저장이므로 자동으로 응답 트래픽을 허용하지 않기 때문입니다. 자세한 내용은 Amazon VPC 사용 설명서의 [네트워크 ACL](#)을 참조하십시오.

기본 보안 그룹

AWS 계정에는 EC2-Classic에 대한 VPC 및 리전별 기본 보안 그룹이 자동으로 생성됩니다. 인스턴스를 시작할 때 보안 그룹을 지정하지 않을 경우 인스턴스는 기본 보안 그룹과 자동으로 연결됩니다.

기본 보안 그룹의 이름은 `default`고 AWS에 의해 ID가 배정됩니다. 각 기본 보안 그룹에 대한 기본 규칙은 다음과 같습니다.

- 기본 보안 그룹과 연결된 다른 인스턴스에서 수신되는 모든 인바운드 트래픽을 허용함(보안 그룹은 인바운드 규칙에서 스스로를 원본 보안 그룹으로 지정함)
- 인스턴스의 모든 아웃바운드 트래픽을 허용합니다.

기본 보안 그룹에 대한 인바운드 규칙을 추가하거나 제거할 수 있습니다. VPC 기본 보안 그룹에 대한 아웃바운드 규칙을 추가하거나 제거할 수 있습니다.

기본 보안 그룹을 삭제할 수 없습니다. EC2-Classic 기본 보안 그룹을 삭제하려고 하면 `Client.InvalidGroup.Reserved: The security group 'default' is reserved`라는 오류가 표시되고,

VPC 기본 보안 그룹을 삭제하려고 하면 `Client.CannotDelete: the specified group: "sg-51530134" name: "default" cannot be deleted by a user`라는 오류가 표시됩니다.

사용자 지정 보안 그룹

인스턴스에서 기본 보안 그룹을 사용하지 않도록 하려면 고유한 보안 그룹을 생성하고 인스턴스를 시작할 때 해당 보안 그룹을 지정합니다. 인스턴스가 수행하는 다양한 역할(예: 웹 서버, 데이터베이스 서버)을 반영하는 여러 보안 그룹을 생성할 수 있습니다.

보안 그룹을 생성할 때 이름과 설명을 제공해야 합니다. 보안 그룹의 이름과 설명은 최대 255자이며 다음과 같은 문자로 제한됩니다.

- EC2-Classic: ASCII 문자
- EC2-VPC: a-z, A-Z, 0-9, 공백, ._-:/()#@[]+=&{}!\$*

다음은 생성하는 보안 그룹의 기본 규칙입니다.

- 인바운드 트래픽을 허용 안 함
- 모든 아웃바운드 트래픽을 허용합니다

보안 그룹을 생성한 후 연결된 인스턴스에 도달할 인바운드 트래픽의 유형을 반영하도록 인바운드 규칙을 변경할 수 있습니다. EC2-VPC에서는 아웃바운드 규칙도 변경할 수 있습니다.

보안 그룹에 추가할 수 있는 규칙의 유형에 대한 자세한 내용은 [보안 그룹 규칙 참조 \(p. 393\)](#) 섹션을 참조하십시오.

보안 그룹 작업

Amazon EC2 콘솔을 사용하여 보안 그룹과 보안 그룹 규칙을 생성하고 보고 업데이트하고 삭제할 수 있습니다.

목차

- [보안 그룹 생성 \(p. 389\)](#)
- [보안 그룹 설명 \(p. 390\)](#)
- [보안 그룹에 규칙 추가 \(p. 390\)](#)
- [보안 그룹에서 규칙 삭제 \(p. 391\)](#)
- [보안 그룹 삭제 \(p. 392\)](#)
- [API 및 명령 개요 \(p. 392\)](#)

보안 그룹 생성

Amazon EC2 콘솔을 이용해 사용자 지정 보안 그룹을 만들 수 있습니다. EC2-VPC의 경우 보안 그룹을 생성할 VPC를 지정해야 합니다.

새 보안 그룹을 생성하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다.
3. [Create Security Group]를 선택합니다.
4. 보안 그룹의 이름과 설명을 지정합니다.
5. (EC2-Classic만 해당) EC2-Classic에서 사용할 보안 그룹을 생성하려면 [No VPC]를 선택하십시오.

(EC2-VPC) [VPC]의 경우 VPC ID를 선택하여 해당 VPC의 보안 그룹을 생성합니다.

6. 규칙을 추가할 수 있습니다. 또는 [Create]를 선택하여 지금 보안 그룹을 생성하고 나중에 규칙을 추가할 수 있습니다. 규칙 추가에 대한 자세한 내용은 [보안 그룹에 규칙 추가 \(p. 390\)](#) 섹션을 참조하십시오.

Amazon EC2 콘솔을 사용하면 기존의 보안 그룹에서 새 보안 그룹으로 규칙을 복사할 수 있습니다.

보안 그룹을 복사하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다.
3. 복사할 보안 그룹을 선택하고 [Actions]와 [Copy to new]를 차례로 선택합니다.
4. [Create Security Group] 대화 상자가 열리고 기존 보안 그룹의 규칙으로 채워집니다. 새 보안 그룹의 이름과 설명을 지정합니다. [VPC] 목록에서 [No VPC]를 선택하여 EC2-Classic의 보안 그룹을 생성하거나, VPC ID를 선택하여 해당 VPC의 보안 그룹을 생성합니다. 완료했으면 [Create]를 선택합니다.

인스턴스를 시작할 때 인스턴스에 보안 그룹을 할당할 수 있습니다. 규칙을 추가하거나 제거하면 해당 보안 그룹을 할당한 모든 인스턴스에 변경 내용이 자동으로 적용됩니다.

EC2-Classic에서 인스턴스를 시작한 후에는 해당 보안 그룹을 변경할 수 없습니다. VPC에서 인스턴스를 시작한 이후에는 해당 보안 그룹을 변경할 수 있습니다. 자세한 내용은 Amazon VPC 사용 설명서의 [Changing an Instance's Security Groups](#) 섹션을 참조하십시오.

보안 그룹 설명

EC2-Classic의 보안 그룹을 설명하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다.
3. 필터 목록에서 [Network Platforms]를 선택한 다음 [EC2-Classic]을 선택합니다.
4. 보안 그룹을 선택합니다. [Description] 탭에는 일반 정보가 표시됩니다. [Inbound] 탭에는 인바운드 규칙이 표시됩니다.

EC2-VPC의 보안 그룹을 설명하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다.
3. 필터 목록에서 [Network Platforms]를 선택한 다음 [EC2-VPC]를 선택합니다.
4. 보안 그룹을 선택합니다. [Description] 탭에 일반 정보가 표시되고, [Inbound] 탭에 인바운드 규칙이 표시되고, [Outbound] 탭에 아웃바운드 규칙이 표시됩니다.

보안 그룹에 규칙 추가

보안 그룹에 규칙을 추가할 경우 보안 그룹과 연결된 인스턴스에 새 규칙이 자동으로 적용됩니다.

특정 유형의 액세스를 위한 보안 그룹 규칙 선택에 대한 자세한 내용은 [보안 그룹 규칙 참조 \(p. 393\)](#) 섹션을 참조하십시오.

보안 그룹에 규칙을 추가하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택하고 보안 그룹을 선택합니다.

3. Inbound 탭에서 [Edit]를 선택합니다.
4. 대화 상자에서 [Add Rule]를 선택하고 다음과 같이 실행합니다.
 - [Type]에는 프로토콜을 선택합니다.
 - 사용자 지정 TCP 또는 UDP 프로토콜을 선택하는 경우 [Port Range]에 포트 범위를 지정합니다.
 - 사용자 지정 ICMP 프로토콜을 선택하는 경우 [Protocol]에서 ICMP 유형 이름을 선택하고, 해당되는 경우 [Port Range]에서 코드 이름을 선택합니다.
 - [Source]의 경우 다음 중 하나를 선택합니다.
 - [Custom]: 제공되는 필드에 IP 주소(CIDR 표기법), CIDR 블록 또는 다른 보안 그룹을 지정해야 합니다.
 - [Anywhere]: 0.0.0.0/0 IPv4 CIDR 블록을 자동으로 추가합니다. 이 옵션으로 지정된 유형의 모든 트래픽이 인스턴스에 도착하도록 할 수 있습니다. 테스트 환경에서 잠시 사용하는 것은 괜찮지만 프로덕션 환경에서는 안전하지 않습니다. 프로덕션에서는 특정 IP 주소나 주소 범위만 인스턴스에 액세스하도록 허용하십시오.

Note

보안 그룹이 IPv6용으로 사용되는 VPC에 있는 경우, [Anywhere] 옵션을 선택하면 IPv4 트래픽에 대해 한 개(0.0.0.0/0), IPv6 트래픽에 대해 한 개(:/:0), 총 2개의 규칙이— 생성됩니다.

- [My IP]: 로컬 컴퓨터의 퍼블릭 IPv4 주소를 자동으로 추가합니다.

추가할 수 있는 규칙의 유형에 대한 자세한 내용은 [보안 그룹 규칙 참조 \(p. 393\)](#) 섹션을 참조하십시오.

5. [Save]를 선택합니다.
6. VPC 보안 그룹의 경우 아웃바운드 규칙도 지정할 수 있습니다. [Outbound] 탭에서 [Edit], [Add Rule]을 선택하고 다음 작업을 수행합니다.
 - [Type]에는 프로토콜을 선택합니다.
 - 사용자 지정 TCP 또는 UDP 프로토콜을 선택하는 경우 [Port Range]에 포트 범위를 지정합니다.
 - 사용자 지정 ICMP 프로토콜을 선택하는 경우 [Protocol]에서 ICMP 유형 이름을 선택하고, 해당되는 경우 [Port Range]에서 코드 이름을 선택합니다.
 - [Destination]의 경우 다음 중 하나를 선택합니다.
 - [Custom]: 제공되는 필드에 IP 주소(CIDR 표기법), CIDR 블록 또는 다른 보안 그룹을 지정해야 합니다.
 - [Anywhere]: 0.0.0.0/0 IPv4 CIDR 블록을 자동으로 추가합니다. 이 옵션을 선택하면 모든 IP 주소로 아웃바운드 트래픽이 전송됩니다.

Note

보안 그룹이 IPv6용으로 사용되는 VPC에 있는 경우, [Anywhere] 옵션을 선택하면 IPv4 트래픽에 대해 한 개(0.0.0.0/0), IPv6 트래픽에 대해 한 개(:/:0), 총 2개의 규칙이— 생성됩니다.

- [My IP]: 로컬 컴퓨터의 IP 주소를 자동으로 추가합니다.

7. [Save]를 선택합니다.

보안 그룹에서 규칙 삭제

보안 그룹에서 규칙을 삭제할 경우 보안 그룹과 연결된 인스턴스에 해당 변경 내용이 자동으로 적용됩니다.

보안 그룹 규칙을 삭제하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다.

3. 보안 그룹을 선택합니다.
4. [Inbound] 탭(인바운드 규칙) 또는 [Outbound] 탭(아웃바운드 규칙)에서 [Edit]를 선택합니다. 삭제할 각 규칙 옆의 [Delete](x 아이콘)를 선택합니다.
5. [Save]를 선택합니다.

보안 그룹 삭제

인스턴스와 연결된 보안 그룹과 기본 보안 그룹은 삭제할 수 없습니다. 같은 VPC에 있는 다른 보안 그룹의 규칙에서 참조하는 보안 그룹도 삭제할 수 없습니다. 자체 규칙 중 하나에서 보안 그룹이 참조하는 경우 보안 그룹을 삭제하려면 해당 규칙을 삭제해야 합니다.

보안 그룹을 삭제하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다.
3. 보안 그룹을 선택한 다음 [Actions], [Delete Security Group]을 선택합니다.
4. [Yes, Delete]를 선택합니다.

API 및 명령 개요

명령줄 또는 API를 사용하여 이 페이지에서 설명하는 작업을 수행할 수 있습니다. 명령줄 인터페이스 및 사용 가능한 API 목록에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

명령줄 도구를 사용할 때 기본 VPC가 아닌 VPC의 보안 그룹을 지정할 경우 보안 그룹 이름이 아닌 보안 그룹 ID를 사용하여 보안 그룹을 식별해야 합니다.

보안 그룹 생성

- [create-security-group](#)(AWS CLI)
- [New-EC2SecurityGroup](#)(Windows PowerShell용 AWS 도구)

보안 그룹에 하나 이상의 수신 규칙 추가

- [authorize-security-group-ingress](#)(AWS CLI)
- [Grant-EC2SecurityGroupIngress](#)(Windows PowerShell용 AWS 도구)

[EC2-VPC] 보안 그룹에 하나 이상의 송신 규칙 추가

- [authorize-security-group-egress](#)(AWS CLI)
- [Grant-EC2SecurityGroupEgress](#)(Windows PowerShell용 AWS 도구)

하나 이상의 보안 그룹 설명

- [describe-security-groups](#)(AWS CLI)
- [Get-EC2SecurityGroup](#)(Windows PowerShell용 AWS 도구)

[EC2-VPC] 인스턴스에 대한 보안 그룹 수정

- [modify-instance-attribute](#) (AWS CLI)
- [Edit-EC2InstanceAttribute](#) (Windows PowerShell용 AWS 도구)

보안 그룹에서 하나 이상의 수신 규칙 제거

- [revoke-security-group-ingress\(AWS CLI\)](#)
- [Revoke-EC2SecurityGroupIngress\(Windows PowerShell용 AWS 도구\)](#)

[EC2-VPC] 보안 그룹에서 하나 이상의 송신 규칙 제거

- [revoke-security-group-egress\(AWS CLI\)](#)
- [Revoke-EC2SecurityGroupEgress\(Windows PowerShell용 AWS 도구\)](#)

보안 그룹 삭제

- [delete-security-group\(AWS CLI\)](#)
- [Remove-EC2SecurityGroup\(Windows PowerShell용 AWS 도구\)](#)

보안 그룹 규칙 참조

보안 그룹을 생성하고 보안 그룹과 연결된 인스턴스의 역할을 반영하는 규칙을 추가합니다. 예를 들어, 웹 서버로 구성된 인스턴스는 인바운드 HTTP 및 HTTPS 액세스를 허용하는 보안 그룹 규칙이 필요하고, 데이터베이스 인스턴스는 MySQL용 포트 3306을 통한 액세스와 같이, 데이터베이스의 유형에 알맞은 액세스를 허용하는 규칙이 필요합니다.

다음은 특정한 종류의 액세스에 대해 보안 그룹에 추가할 수 있는 규칙의 종류를 예로 든 것입니다.

항목

- [웹 서버 \(p. 393\)](#)
- [데이터베이스 서버 \(p. 394\)](#)
- [같은 그룹에 있는 다른 인스턴스에서 액세스 \(p. 395\)](#)
- [로컬 컴퓨터에서 액세스 \(p. 395\)](#)
- [경로 MTU 검색 \(p. 396\)](#)
- [인스턴스 ping \(p. 396\)](#)
- [DNS 서버 \(p. 397\)](#)
- [Amazon EFS 파일 시스템 \(p. 397\)](#)
- [Elastic Load Balancing \(p. 397\)](#)

웹 서버

다음 인바운드 규칙에서는 임의의 IP 주소로부터 HTTP 및 HTTPS 액세스를 허용합니다. VPC가 IPv6용으로 활성화되면 IPv6 주소에서 인바운드 HTTP 및 HTTPS 트래픽을 제어하기 위한 규칙을 추가할 수 있습니다.

프로토콜 유형	프로토콜 번호	포트	소스 IP	참고
TCP	6	80(HTTP)	0.0.0.0/0	임의의 IPv4 주소에서 인바운드 HTTP 액세스를 허용함
TCP	6	443(HTTPS)	0.0.0.0/0	임의의 IPv4 주소에서 인바운드 HTTPS 액세스를 허용함

프로토콜 유형	프로토콜 번호	포트	소스 IP	참고
TCP	6	80(HTTP)	::/0	(VPC만 해당) 임의의 IPv6 주소에서 인바운드 HTTP 액세스를 허용함
TCP	6	443(HTTPS)	::/0	(VPC만 해당) 임의의 IPv6 주소에서 인바운드 HTTPS 액세스를 허용함

데이터베이스 서버

다음의 인바운드 규칙은 인스턴스에서 실행 중인 데이터베이스의 유형에 따라 데이터베이스 액세스를 위해 추가할 수 있는 규칙을 예로 든 것입니다. Amazon RDS 인스턴스에 대한 자세한 내용은 [Amazon Relational Database Service 사용 설명서](#) 섹션을 참조하십시오.

원본 IP의 경우 다음 중 하나를 지정합니다.

- 로컬 네트워크의 특정 IP 주소 또는 IP 주소의 범위
- 데이터베이스에 액세스하는 인스턴스 그룹의 보안 그룹 ID

프로토콜 유형	프로토콜 번호	포트	참고
TCP	6	1433(MS SQL)	Microsoft SQL Server 데이터베이스 액세스를 위한 기본 포트(예: Amazon RDS 인스턴스에서)
TCP	6	3306(MYSQL/Aurora)	MySQL 또는 Aurora 데이터베이스 액세스를 위한 기본 포트(예: Amazon RDS 인스턴스에서)
TCP	6	5439(Redshift)	Amazon Redshift 클러스터 데이터베이스 액세스를 위한 기본 포트.
TCP	6	5432(PostgreSQL)	PostgreSQL 데이터베이스 액세스를 위한 기본 포트(예: Amazon RDS 인스턴스에서)
TCP	6	1521(Oracle)	Oracle 데이터베이스 액세스를 위한 기본 포트(예: Amazon RDS 인스턴스에서)

(VPC만 해당) 예를 들어, 소프트웨어 업데이트를 위해 인터넷 액세스를 허용하지만 다른 모든 종류의 트래픽은 제한하려는 경우, 데이터베이스 서버에서 아웃바운드 트래픽을 선택적으로 제한할 수 있습니다. 먼저 모든 아웃바운드 트래픽을 허용하는 기본 아웃바운드 규칙을 제거해야 합니다.

프로토콜 유형	프로토콜 번호	포트	목적지 IP	참고
TCP	6	80(HTTP)	0.0.0.0/0	임의의 IPv4 주소에 대한 아웃바운드 HTTP 액세스를 허용함
TCP	6	443(HTTPS)	0.0.0.0/0	임의의 IPv4 주소에 대한 아웃바운드 HTTPS 액세스를 허용함
TCP	6	80(HTTP)	::/0	(IPv6 사용 VPC만 해당) 임의의 IPv6 주소에 대한 아웃바운드 HTTP 액세스를 허용함
TCP	6	443(HTTPS)	::/0	(IPv6 사용 VPC만 해당) 임의의 IPv6 주소에 대한 아웃바운드 HTTPS 액세스를 허용함

같은 그룹에 있는 다른 인스턴스에서 액세스

같은 보안 그룹과 연결된 여러 인스턴스가 서로 통신할 수 있게 하려면 이에 대한 규칙을 명시적으로 추가해야 합니다.

다음 표에서는 연결된 인스턴스가 서로 통신할 수 있도록 하기 위한 VPC 보안 그룹의 인바운드 규칙을 설명합니다. 이 규칙에서는 모든 유형의 트래픽을 허용합니다.

프로토콜 유형	프로토콜 번호	포트	소스 IP
-1(모두)	-1(모두)	-1(모두)	보안 그룹의 ID

다음 표에서는 연결된 인스턴스가 서로 통신할 수 있도록 하기 위한 EC2-Classic 보안 그룹의 인바운드 규칙을 설명합니다. 이런 규칙에서는 모든 유형의 트래픽을 허용합니다.

프로토콜 유형	프로토콜 번호	포트	소스 IP
ICMP	1	-1(모두)	보안 그룹의 ID
TCP	6	0 - 65535(모두)	보안 그룹의 ID
UDP	17	0 - 65535(모두)	보안 그룹의 ID

로컬 컴퓨터에서 액세스

인스턴스에 연결하려면 보안 그룹에 SSH 액세스(Linux 인스턴스) 또는 RDP 액세스(Windows 인스턴스)를 허용하는 인바운드 규칙이 있어야 합니다.

프로토콜 유형	프로토콜 번호	포트	소스 IP
TCP	6	22(SSH)	컴퓨터의 퍼블릭 IPv4 주소 또는 로컬 네트워크의 IP 주소 범위. IPv6를 위해 VPC가 활성화되어 있고 인스턴스에 IPv6 주소가 있는 경우 IPv6 주소 또는 범위를 입력할 수 있습니다.
TCP	6	3389(RDP)	컴퓨터의 퍼블릭 IPv4 주소 또는 로컬 네트워크의 IP 주소 범위. IPv6를 위해 VPC가 활성화되어 있고 인스턴스에 IPv6 주소가 있는 경우 IPv6 주소 또는 범위를 입력할 수 있습니다.

경로 MTU 검색

경로 MTU는 발신 호스트와 수신 호스트 간의 경로에서 지원되는 최대 패킷 사이즈입니다. 호스트가 수신 호스트의 MTU 또는 경로를 따라 디바이스의 MTU보다 큰 패킷을 전송하는 경우 수신 호스트가

Destination Unreachable: Fragmentation Needed and Don't Fragment was Set

과 같은 ICMP 메시지를 반환합니다.

인스턴스가 이 메시지를 수신하고 패킷이 삭제되지 않도록 하려면 인바운드 보안 그룹 규칙에 ICMP 규칙을 추가해야 합니다.

프로토콜 유형	프로토콜 번호	ICMP 유형	ICMP 코드	소스 IP
ICMP	1	3(대상에 연결할 수 없음)	4(조각화가 필요하지만 조각화 금지가 설정되었음)	인스턴스와 통신하는 호스트의 IP 주소

인스턴스 ping

ping 명령은 ICMP 트래픽의 한 유형입니다. 인스턴스를 ping하려면 다음 인바운드 ICMP 규칙을 추가해야 합니다.

프로토콜 유형	프로토콜 번호	ICMP 유형	ICMP 코드	소스 IP
ICMP	1	8(에코)	해당 사항 없음	컴퓨터의 퍼블릭 IPv4 주소 또는 로컬 네트워크의 IPv4 주소 범위

ping6 명령을 사용하여 인스턴스에 대한 IPv6 주소를 ping하려면 다음 인바운드 ICMPv6 규칙을 추가해야 합니다.

프로토콜 유형	프로토콜 번호	ICMP 유형	ICMP 코드	소스 IP
ICMPv6	58	128(에코)	0	컴퓨터의 IPv6 주소 또는 로컬 네트워크의 IPv6 주소 범위

DNS 서버

EC2 인스턴스를 DNS 서버로 설정한 경우 TCP 및 UDP 트래픽이 포트 53을 통해 DNS 서버에 연결할 수 있는지 확인해야 합니다.

원본 IP의 경우 다음 중 하나를 지정합니다.

- 네트워크의 특정 IP 주소 또는 IP 주소의 범위
- 네트워크에서 DNS 서버에 액세스할 필요가 있는 인스턴스 그룹의 보안 그룹 ID

프로토콜 유형	프로토콜 번호	포트
TCP	6	53
UDP	17	53

Amazon EFS 파일 시스템

Amazon EC2 인스턴스에서 Amazon EFS 파일 시스템을 사용하려면 Amazon EFS 마운트 대상과 연결되는 보안 그룹이 NFS 프로토콜을 통한 트래픽 전송을 허용해야 합니다.

프로토콜 유형	프로토콜 번호	포트	소스 IP	참고
TCP	6	2049(NFS)	보안 그룹의 ID.	이 보안 그룹과 연결된 리소스(탑재 대상 포함)에서 인바운드 NFS 액세스를 허용합니다.

Amazon EFS 파일 시스템을 Amazon EC2 인스턴스에 마운트하려면 인스턴스에 연결해야 합니다. 따라서 인스턴스와 연결되는 보안 그룹은 로컬 컴퓨터 또는 로컬 네트워크의 인바운드 SSH 트래픽을 허용하는 규칙이 필요합니다.

프로토콜 유형	프로토콜 번호	포트	소스 IP	참고
TCP	6	22(SSH)	로컬 컴퓨터의 IP 주소 범위 또는 네트워크의 IP 주소 범위.	로컬 컴퓨터로부터의 인바운드 SSH 액세스를 허용합니다.

Elastic Load Balancing

로드 밸런서를 사용하고 있는 경우 로드 밸런서에 연결된 보안 그룹은 인스턴스 또는 대상과 통신을 허용하는 규칙을 보유해야 합니다.

인바운드

프로토콜 유형	프로토콜 번호	포트	소스 IP	참고
TCP	6	리스너 포트	인터넷 경계 로드 밸런서의 경우: 0.0.0.0/0(모든 IPv4 주소) 내부 로드 밸런서의 경우: VPC의 IPv4 CIDR 블록	로드 밸런서 리스너 포트의 인바운드 트래픽을 허용합니다.

아웃바운드

프로토콜 유형	프로토콜 번호	포트	목적지 IP	참고
TCP	6	인스턴스 리스너 포트	인스턴스 보안 그룹의 ID	인스턴스 리스너 포트의 인스턴스로 아웃바운드 트래픽을 허용합니다.
TCP	6	상태 확인 포트	인스턴스 보안 그룹의 ID	상태 확인 포트의 인스턴스로 아웃바운드 트래픽을 허용합니다.

인스턴스에 대한 보안 그룹 규칙은 로드 밸런서가 리스너 포트 및 상태 확인 포트에서 인스턴스와 통신할 수 있도록 허용해야 합니다.

인바운드

프로토콜 유형	프로토콜 번호	포트	소스 IP	참고
TCP	6	인스턴스 리스너 포트	로드 밸런서 보안 그룹의 ID	인스턴스 리스너 포트의 로드 밸런서에서 트래픽을 허용합니다.
TCP	6	상태 확인 포트	로드 밸런서 보안 그룹의 ID	상태 확인 포트의 로드 밸런서에서 트래픽을 허용합니다.

자세한 내용은 Classic Load Balancer 가이드의 [Configure Security Groups for Your Classic Load Balancer](#) 및 Application Load Balancer 가이드의 [Security Groups for Your Application Load Balancer](#)를 참조하십시오.

Amazon EC2 리소스에 대한 액세스 제어

보안 자격 증명은 AWS의 서비스에서 사용자를 식별하고 Amazon EC2 리소스와 같은 AWS 리소스의 무제한 사용을 허가하는 데 사용됩니다. Amazon EC2 및 AWS Identity and Access Management(IAM)의 기능을 사용하면 보안 자격 증명을 공유하지 않고도 다른 사용자, 서비스 및 애플리케이션에 Amazon EC2 리소스 사용을 허가할 수 있습니다. IAM을 사용하여 다른 사용자가 AWS 계정의 리소스를 사용하는 방법을 제어하고 보안 그룹을 사용하여 Amazon EC2 인스턴스에 대한 액세스를 제어할 수 있습니다. Amazon EC2 리소스의 전체 사용 또는 제한 사용을 허가할 수 있습니다.

목차

- [인스턴스에 대한 네트워크 액세스 \(p. 399\)](#)
- [Amazon EC2 권한 속성 \(p. 399\)](#)
- [IAM 및 Amazon EC2 \(p. 399\)](#)
- [Amazon EC2에 대한 IAM 정책 \(p. 401\)](#)
- [Amazon EC2의 IAM 역할 \(p. 456\)](#)
- [Linux 인스턴스의 인바운드 트래픽 권한 부여 \(p. 464\)](#)

인스턴스에 대한 네트워크 액세스

보안 그룹은 하나 이상의 인스턴스에 도달하도록 허용되는 트래픽을 제어하는 방화벽 역할을 합니다. 인스턴스를 시작할 때 하나 이상의 보안 그룹을 할당합니다. 각 보안 그룹에는 인스턴스의 트래픽을 제어하는 규칙을 추가합니다. 언제든지 보안 그룹에 대한 규칙을 수정할 수 있습니다. 새 규칙은 보안 그룹이 할당된 모든 인스턴스에 자동으로 적용됩니다.

자세한 내용은 [Linux 인스턴스의 인바운드 트래픽 권한 부여 \(p. 464\)](#) 섹션을 참조하십시오.

Amazon EC2 권한 속성

조직에는 여러 AWS 계정이 있을 수 있습니다. Amazon EC2에서는 Amazon 머신 이미지(AMI) 및 Amazon EBS 스냅샷을 사용할 수 있는 추가 AWS 계정을 지정할 수 있습니다. 이러한 권한은 AWS 계정 수준으로만 적용되며, 지정된 AWS 계정에 속한 특정 사용자의 권한을 제한할 수는 없습니다. 지정한 AWS 계정의 모든 사용자가 AMI 또는 스냅샷을 사용할 수 있습니다.

각 AMI에는 AMI에 액세스할 수 있는 AWS 계정을 제어하는 `LaunchPermission` 속성이 있습니다. 자세한 내용은 [퍼블릭 AMI 설정 \(p. 71\)](#) 섹션을 참조하십시오.

각 Amazon EBS 스냅샷에는 스냅샷을 사용할 수 있는 AWS 계정을 제어하는 `createVolumePermission` 속성이 있습니다. 자세한 내용은 [Amazon EBS 스냅샷 공유 \(p. 612\)](#) 섹션을 참조하십시오.

IAM 및 Amazon EC2

IAM을 사용하여 다음을 수행할 수 있습니다.

- AWS 계정에 속하는 사용자 및 그룹 생성
- AWS 계정 사용자 각각에 고유한 보안 자격 증명 할당
- 작업 수행 시 각 사용자의 AWS 리소스 사용 권한 제어
- 다른 AWS 계정의 사용자와 AWS 리소스 공유
- AWS 계정에 적용할 규칙 생성 및 규칙을 관리할 사용자나 서비스 규정
- 엔터프라이즈의 기존 자격 증명을 사용해 AWS 리소스를 사용하는 작업 권한 허용

IAM과 Amazon EC2 함께 사용하면 조직 내 사용자별로 특정 Amazon EC2 API 작업을 사용하는 작업 수행과 특정 AWS 리소스의 사용 권한을 제어할 수 있습니다.

이 항목에서는 다음과 같은 의문 사항을 해결해 줍니다.

- IAM에서 그룹과 사용자를 생성하려면 어떻게 해야 하나요?
- 정책을 생성하려면 어떻게 해야 하나요?
- Amazon EC2에서 작업을 수행하려면 어떠한 IAM 정책이 필요한가요?
- Amazon EC2에서 작업을 수행할 수 있는 권한을 부여하려면 어떻게 해야 하나요?

- Amazon EC2의 특정 리소스에 대해 작업을 수행할 수 있는 권한을 부여하려면 어떻게 해야 하나요?

IAM 그룹 및 사용자 생성

IAM 그룹을 생성하려면 다음을 수행합니다.

- <https://console.aws.amazon.com/iam/>에서 IAM 콘솔에 로그인합니다.
- 탐색 창에서 [Groups]를 선택한 다음, [Create New Group]을 선택합니다.
- [Group Name] 상자에 그룹 이름을 입력한 다음, [Next Step]을 선택합니다.
- [Attach Policy] 페이지에서 AWS 관리형 정책을 선택합니다. 예를 들어 Amazon EC2의 경우 다음 AWS 관리형 정책 중 하나가 적합할 수 있습니다.
 - PowerUserAccess
 - ReadOnlyAccess
 - AmazonEC2FullAccess
 - AmazonEC2ReadOnlyAccess
- [Next Step]을 선택한 다음, [Create Group]을 선택합니다.

[Group Name] 아래에 새 그룹이 나열됩니다.

IAM 사용자를 생성하고, 그룹에 사용자를 추가하고, 사용자의 암호를 생성하려면 다음을 수행합니다.

- 탐색 창에서 [Users]를 선택한 다음 [Add Users]를 선택합니다.
- 사용자 이름을 입력합니다.
- 이 사용자 세트의 액세스 유형을 선택합니다. [Programmatic access]와 [AWS Management Console access]를 모두 선택합니다.
- Console password type의 경우 다음 중 하나를 선택합니다.
 - Autogenerated password. 각 사용자는 현재 유효한 암호 정책(있는 경우)에 따라 임의로 생성되는 암호를 받습니다. [Final] 페이지에 이르면 암호를 보거나 다운로드할 수 있습니다.
 - [Custom password]. 입력란에 입력하는 암호가 각 사용자에게 할당됩니다.
- Next: Permissions를 선택합니다.
- [Set permissions] 페이지에서 [Add user to group]를 선택합니다. 이전에 만든 그룹을 선택합니다.
- [Next: Review]와 [Create user]를 선택합니다.
- 사용자의 액세스 키(액세스 키 ID와 보안 액세스 키)를 보려면 보고자 하는 각 암호와 보안 액세스 키 옆에 있는 [Show]를 선택합니다. 액세스 키를 저장하려면 [Download .csv]를 선택한 후 안전한 위치에 파일을 저장합니다.

Note

이 단계를 완료한 후에는 보안 액세스 키를 검색할 수 없으며, 키를 분실한 경우 새로 생성해야 합니다.

- [Close]를 선택합니다.
- 각 사용자에게 자격 증명(액세스 키와 암호)을 제공하여 IAM 그룹에 지정한 권한에 따라 서비스를 사용할 수 있도록 허용합니다.

관련 주제

IAM에 대한 자세한 내용은 다음을 참조하십시오.

- Amazon EC2에 대한 IAM 정책 (p. 401)
- Amazon EC2의 IAM 역할 (p. 456)
- AWS Identity and Access Management(IAM)
- IAM 사용 설명서

Amazon EC2에 대한 IAM 정책

기본적으로 IAM 사용자에게는 Amazon EC2 리소스를 생성 또는 수정하거나 Amazon EC2 API를 사용하여 작업을 수행할 권한이 없습니다. Amazon EC2 콘솔이나 CLI를 사용하더라도 마찬가지입니다. IAM 사용자에게 리소스 생성 또는 수정 및 작업 수행을 허용하려면 IAM 사용자에게 필요한 특정 리소스 및 API 작업을 사용할 권한을 부여하는 IAM 정책을 생성하고, 해당 권한을 필요로 하는 IAM 사용자 또는 그룹에게 정책을 연결해야 합니다.

사용자 또는 사용자 그룹에 정책을 연결하면 지정된 리소스에 대해 지정된 작업을 수행할 권한이 허용되거나 거부됩니다. IAM 정책에 대한 자세한 내용은 IAM 사용 설명서에서 [권한 및 정책](#)을 참조하십시오. 사용자 지정 IAM 정책 관리 및 생성에 대한 자세한 내용은 [IAM 정책 관리](#) 섹션을 참조하십시오.

시작하기

IAM 정책은 하나 이상의 Amazon EC2 작업을 사용할 권한을 허용하거나 거부해야 합니다. 또한 작업에 사용할 수 있는 리소스를 지정해야 합니다. 모든 리소스일 수도 있고, 경우에 따라서는 특정 리소스일 수도 있습니다. 또한 정책은 리소스에 적용할 조건을 포함할 수 있습니다.

Amazon EC2에서는 리소스 수준 권한을 부분적으로 지원합니다. 즉, 일부 EC2 API 작업의 경우에는 해당 작업에 사용할 수 있는 리소스를 별도로 지정할 수 없으며 해당 작업에 모든 리소스를 사용할 수 있도록 허용해야 합니다.

작업	주제
정책의 기본적인 구조 이해	정책 구문 (p. 402)
정책의 작업 정의	Amazon EC2 작업 (p. 402)
정책의 특정 리소스 정의	Amazon EC2의 Amazon 리소스 이름 (p. 403)
리소스 사용에 조건 적용	Amazon EC2의 조건 키 (p. 405)
Amazon EC2에서 사용 가능한 리소스 수준 권한 작업	Amazon EC2 API 작업에 지원되는 리소스 수준 권한 (p. 409)
정책 테스트	사용자에게 필요한 권한이 있는지 확인 (p. 408)
CLI 또는 SDK용 예제 정책	AWS CLI 또는 AWS SDK 작업을 위한 예제 정책 (p. 431)
Amazon EC2 콘솔용 예제 정책	Amazon EC2 콘솔 작업을 위한 예제 정책 (p. 449)

정책 구조

다음 항목에서는 IAM 정책의 구조에 대해 설명합니다.

항목

- [정책 구문 \(p. 402\)](#)
- [Amazon EC2 작업 \(p. 402\)](#)

- Amazon EC2의 Amazon 리소스 이름 (p. 403)
- Amazon EC2의 조건 키 (p. 405)
- 사용자에게 필요한 권한이 있는지 확인 (p. 408)

정책 구문

IAM 정책은 하나 이상의 명령문으로 구성된 JSON 문서입니다. 각 명령문의 구조는 다음과 같습니다.

```
{  
    "Statement": [  
        {  
            "Effect": "effect",  
            "Action": "action",  
            "Resource": "arn",  
            "Condition": {  
                "condition": {  
                    "key": "value"  
                }  
            }  
        }  
    ]  
}
```

명령문을 이루는 요소는 다양합니다.

- Effect: effect는 Allow 또는 Deny일 수 있습니다. 기본적으로 IAM 사용자에게는 리소스 및 API 작업을 사용 할 권한이 없으므로 모든 요청이 거부됩니다. 명시적 허용은 기본 설정을 무시합니다. 명시적 거부는 모든 허용을 무시합니다.
- Action: action은 권한을 부여하거나 거부할 특정 API 작업입니다. action을 지정하는 방법에 대한 자세한 내용은 [Amazon EC2 작업 \(p. 402\)](#) 섹션을 참조하십시오.
- [Resource]: 작업의 영향을 받는 리소스입니다. 일부 Amazon EC2 API 작업의 경우 작업이 생성하거나 수정할 수 있는 리소스를 정책에 구체적으로 포함할 수 있습니다. 명령문에서 리소스를 지정하려면 Amazon 리소스 이름(ARN)을 사용해야 합니다. ARN 값을 지정하는 방법에 대한 자세한 내용은 [Amazon EC2의 Amazon 리소스 이름 \(p. 403\)](#) 섹션을 참조하십시오. 어떠한 API 작업이 어떠한 ARN을 지원하는지에 대한 자세한 내용은 [Amazon EC2 API 작업에 지원되는 리소스 수준 권한 \(p. 409\)](#) 섹션을 참조하십시오. API 작업이 ARN을 지원하지 않는 경우 * 와일드카드를 사용하여 모든 리소스가 작업에 영향을 받을 수 있도록 지정합니다.
- Condition: Condition은 선택 사항으로서 정책이 적용되는 시점을 제어하는 데 사용할 수 있습니다. Amazon EC2에 조건을 지정하는 방법에 대한 자세한 내용은 [Amazon EC2의 조건 키 \(p. 405\)](#) 섹션을 참조하십시오.

Amazon EC2용 예제 IAM 정책 명령문에 대한 자세한 내용은 [AWS CLI 또는 AWS SDK 작업을 위한 예제 정책 \(p. 431\)](#) 섹션을 참조하십시오.

Amazon EC2 작업

IAM 정책 명령문에는 IAM을 지원하는 모든 서비스의 모든 API 작업을 지정할 수 있습니다. Amazon EC2의 경우 ec2: 접두사와 함께 API 작업 이름을 사용합니다. 예를 들면 ec2:RunInstances 및 ec2:CreateImage 등입니다.

명령문 하나에 여러 작업을 지정하려면 다음과 같이 쉼표로 구분합니다.

```
"Action": ["ec2:action1", "ec2:action2"]
```

와일드카드를 사용하여 여러 작업을 지정할 수도 있습니다. 예를 들어 다음과 같이 이름이 "Describe"로 시작 되는 모든 작업을 지정할 수 있습니다.

```
"Action": "ec2:Describe*"
```

모든 Amazon EC2 API 작업을 지정하려면 다음과 같이 * 와일드카드를 사용합니다.

```
"Action": "ec2:/*"
```

Amazon EC2 작업의 목록은 Amazon EC2 API Reference에서 [작업](#)을 참조하십시오.

Amazon EC2의 Amazon 리소스 이름

각 IAM 정책 명령문은 ARN을 사용하여 지정한 리소스에 적용됩니다.

Important

현재 일부 API 작업은 개별 ARN을 지원하지 않으며, 이후에 더 많은 API 작업과 Amazon EC2 리소스 ARN이 추가로 지원될 예정입니다. 어떠한 Amazon EC2 API 작업에 어떠한 ARN을 사용할 수 있는지 및 각 ARN에 지원되는 조건 키에 대한 자세한 내용은 [Amazon EC2 API 작업에 지원되는 리소스 수준 권한 \(p. 409\)](#) 섹션을 참조하십시오.

ARN의 일반적인 구문은 다음과 같습니다.

```
arn:aws:[service]:[region]:[account]:resourceType/resourcePath
```

service

서비스(예: ec2)입니다.

region

리소스의 리전(예: us-east-1)입니다.

account

AWS 계정 ID이며 하이픈은 제외합니다(예: 123456789012).

resourceType

리소스의 유형(예: instance)입니다.

resourcePath

리소스를 식별하는 경로입니다. 경로에 * 와일드카드를 사용할 수 있습니다.

예를 들어 명령문에서 다음과 같이 ARN을 사용하여 특정 인스턴스(i-1234567890abcdef0)를 나타낼 수 있습니다.

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0"
```

모든 리소스를 지정해야 하거나 특정 API 작업이 ARN을 지원하지 않는 경우 다음과 같이 Resource 요소에 * 와일드카드를 사용합니다.

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*"
```

모든 리소스를 지정해야 하거나 특정 API 작업이 ARN을 지원하지 않는 경우 다음과 같이 Resource 요소에 * 와일드카드를 사용합니다.

```
"Resource": "*"
```

다음 표에서는 Amazon EC2 API 작업에 사용되는 각 리소스 유형의 ARN을 보여 줍니다.

리소스 유형	ARN
모든 Amazon EC2 리소스	arn:aws:ec2:*
지정한 리전에서 지정한 계정이 소유한 모든 Amazon EC2 리소스	arn:aws:ec2:region:account:*
고객 게이트웨이	arn:aws:ec2:region:account:customer-gateway/cgw-id 여기에서 cgw-id는 cgw-xxxxxxxx입니다.
DHCP 옵션 세트	arn:aws:ec2:region:account:dhcp-options/dhcp-options-id 여기에서 dhcp-options-id는 dopt-xxxxxxxx입니다.
이미지	arn:aws:ec2:region::image/image-id 여기에서 image-id는 AMI, AKI 또는 ARI의 ID이며 account는 사용되지 않습니다.
인스턴스	arn:aws:ec2:region:account:instance/instance-id 여기에서 instance-id는 i-xxxxxxxx 또는 i-xxxxxxxxxxxxxxxxx입니다.
인스턴스 프로필	arn:aws:iam::account:instance-profile/instance-profile-name 여기에서 instance-profile-name은 인스턴스 프로파일의 이름이며 region은 사용되지 않습니다.
인터넷 게이트웨이	arn:aws:ec2:region:account:internet-gateway/igw-id 여기에서 igw-id는 igw-xxxxxxxx입니다.
키 페어	arn:aws:ec2:region:account:key-pair/key-pair-name 여기에서 key-pair-name은 키 페어 이름(예: gsg-keypair)입니다.
네트워크 ACL	arn:aws:ec2:region:account:network-acl-nacl-id 여기에서 nacl-id는 acl-xxxxxxxx입니다.
네트워크 인터페이스	arn:aws:ec2:region:account:network-interface/eni-id 여기에서 eni-id는 eni-xxxxxxxx입니다.
배치 그룹	arn:aws:ec2:region:account:placement-group/placement-group-name 여기에서 placement-group-name은 배치 그룹 이름(예: my-cluster)입니다.
예약 인스턴스	arn:aws:ec2:region:account:reserved-instance/reservation-id 여기서 reservation-id는 xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx입니다.
라우팅 테이블	arn:aws:ec2:region:account:route-table/route-table-id 여기에서 route-table-id는 rtb-xxxxxxxx입니다.
보안 그룹	arn:aws:ec2:region:account:security-group/security-group-id

리소스 유형	ARN
	여기에서 security-group-id는 sg-xxxxxxxx입니다.
스냅샷	arn:aws:ec2:region::snapshot/snapshot-id 여기에서 snapshot-id는 snap-xxxxxxxx 또는 snap-xxxxxxxxxxxxxxxxxx이며 account는 사용되지 않습니다.
스팟 인스턴스 요청	arn:aws:ec2:region:account:spot-instance-request/spot-instance-request-id 여기에서 spot-instance-request-id는 sir-xxxxxxxx입니다.
서브넷	arn:aws:ec2:region:account:subnet/subnet-id 여기에서 subnet-id는 subnet-xxxxxxxx입니다.
볼륨	arn:aws:ec2:region:account:volume/volume-id 여기에서 volume-id는 vol-xxxxxxxx 또는 vol-xxxxxxxxxxxxxxxxxx입니다.
VPC	arn:aws:ec2:region:account:vpc/vpc-id 여기에서 vpc-id는 vpc-xxxxxxxx입니다.
VPC 피어링 연결	arn:aws:ec2:region:account:vpc-peering-connection/vpc-peering-connection-id 여기에서 vpc-peering connection-id는 pcx-xxxxxxxx입니다.
VPN 연결	arn:aws:ec2:region:account:vpn-connection/vpn-connection-id 여기에서 vpn-connection-id는 vpn-xxxxxxxx입니다.
VPN 게이트웨이	arn:aws:ec2:region:account:vpn-gateway/vpn-gateway-id 여기에서 vpn-gateway-id는 vgw-xxxxxxxx입니다.

다양한 Amazon EC2 API 작업에는 여러 리소스가 관여합니다. 예를 들어 `AttachVolume`은 Amazon EBS 볼륨을 인스턴스에 연결하므로 IAM 사용자에게 볼륨 사용 권한과 인스턴스 사용 권한이 있어야 합니다. 명령문 하나에 여러 리소스를 지정하라면 다음과 같이 각 ARN을 쉼표로 구분합니다.

```
"Resource": ["arn1", "arn2"]
```

ARN에 대한 보다 일반적인 내용은 Amazon Web Services 일반 참조에서 [Amazon 리소스 이름\(ARN\)](#) 및 [AWS 서비스 네임스페이스](#) 섹션을 참조하십시오. Amazon EC2 작업에 의해 생성 또는 수정되는 리소스에 대한 자세한 내용 및 IAM 정책 명령문에 사용할 수 있는 ARN에 대한 자세한 내용은 Amazon EC2 API Reference에서 [IAM 사용자에게 Amazon EC2 리소스에 대한 필요 권한 부여](#) 섹션을 참조하십시오.

Amazon EC2의 조건 키

정책 명령문에서 정책이 적용되는 시점을 제어하는 조건을 지정할 수 있습니다. 각 조건에는 하나 이상의 키-값 쌍이 포함됩니다. 조건 키에는 대/소문자가 구분되지 않습니다. AWS 전체 범위 조건 키 및 추가적인 서비스별 조건 키가 정의되어 있습니다.

여러 조건을 지정하거나 조건 하나에 여러 키를 지정하는 경우 논리적 AND 연산을 적용하여 평가합니다. 조건 하나에서 키 하나에 여러 값을 지정하면 논리적 OR 연산자를 적용하여 조건을 평가합니다. 모든 조건이 충족되어야 권한이 부여됩니다.

조건을 지정할 때 자리표시자를 사용할 수도 있습니다. 예를 들어 IAM 사용자 이름을 지정하는 태그가 포함된 리소스를 사용할 IAM 사용자 권한을 부여할 수 있습니다. 자세한 내용은 IAM 사용 설명서에서 [Policy Variables](#)를 참조하십시오.

Important

여러 조건 키들이 하나의 리소스에 딸려 있고, 일부 API 작업은 다수의 리소스를 사용합니다. 조건 키로 정책을 작성하는 경우에는 설명의 `Resource` 요소를 이용해 조건 키가 적용되는 리소스를 지정하십시오. 그렇게 하지 않으면, 조건 키가 해당되지 않는 리소스에 대해서는 조건 검사가 실패하여 정책이 사용자로 하여금 작업을 전혀 수행하지 못하게 막을 수도 있습니다. 리소스를 지정하고 싶지 않거나 다수의 API 작업을 포함하도록 정책의 `Action` 요소를 작성했다면, 반드시 ...`IfExists` 조건 유형을 이용해 조건 키가 그것을 사용하지 않는 리소스에 대해서는 무시되도록 해야 합니다. 자세한 내용은 IAM 사용 설명서의 [...IfExists Conditions](#)를 참조하십시오.

Amazon EC2는 다음의 서비스별 조건 키를 구현합니다.

조건 키	키-값 쌍	평가 유형
<code>ec2:AccepterVpc</code>	" <code>ec2:AccepterVpc</code> ":" <code>vpc-arn</code> " 여기서 <code>vpc-arn</code> 은 VPC 피어링 연결에서 수락자 VPC에 대한 VPC ARN입니다.	ARN, Null
<code>ec2:AvailabilityZone</code>	" <code>ec2:AvailabilityZone</code> ":" <code>az-api-name</code> " 여기에서 <code>az-api-name</code> 은 가용 영역의 이름(예: <code>us-west-2a</code>)입니다. 가용 영역을 나열하려면 -describe-availability-zones 를 사용합니다.	String, Null
<code>ec2>CreateAction</code>	" <code>ec2>CreateAction</code> ":" <code>api-name</code> " 여기서 <code>api-name</code> 은 리소스 생성 작업의 이름입니다(예: <code>RunInstances</code>)。	String, Null
<code>ec2:EbsOptimized</code>	" <code>ec2:EbsOptimized</code> ":" <code>optimized-flag</code> " 여기서 <code>optimized-flag</code> 는 <code>true</code> <code>false</code> 입니다. (인스턴스에 대해)	Boolean, Null
<code>ec2:Encrypted</code>	" <code>ec2:Encrypted</code> ":" <code>encrypted-flag</code> " 여기서 <code>encrypted-flag</code> 는 <code>true</code> <code>false</code> 입니다. (EBS 볼륨에 대해)	Boolean, Null
<code>ec2:ImageType</code>	" <code>ec2:ImageType</code> ":" <code>image-type-api-name</code> " 여기에서 <code>image-type-api-name</code> 은 <code>ami</code> <code>aki</code> <code>ari</code> 입니다.	String, Null
<code>ec2:InstanceProfile</code>	" <code>ec2:InstanceProfile</code> ":" <code>instance-profile-arn</code> " 여기에서 <code>instance-profile-arn</code> 은 인스턴스 프로파일 ARN입니다.	ARN, Null
<code>ec2:InstanceType</code>	" <code>ec2:InstanceType</code> ":" <code>instance-type-api-name</code> " 여기에서 <code>instance-type-api-name</code> 은 인스턴스 유형 이름()입니다.	String, Null
<code>ec2:Owner</code>	" <code>ec2:Owner</code> ":" <code>account-id</code> "	String, Null

조건 키	키-값 쌍	평가 유형
	여기에서, account-id는 amazon aws-marketplace aws-account-id입니다.	
ec2:ParentSnapshot	"ec2:ParentSnapshot":"snapshot-arn" 여기에서 snapshot-arn은 스냅샷 ARN입니다.	ARN, Null
ec2:ParentVolume	"ec2:ParentVolume":"volume-arn" 여기에서 volume-arn은 볼륨 ARN입니다.	ARN, Null
ec2:PlacementGroup	"ec2:PlacementGroup":"placement-group-arn" 여기에서 placement-group-arn은 배치 그룹 ARN입니다.	ARN, Null
ec2:PlacementGroup	"ec2:PlacementGroupStrategy":"placement-group-strategy" 여기에서 placement-group-strategy는 cluster입니다.	String, Null
ec2:ProductCode	"ec2:ProductCode":"product-code" 여기에서 product-code는 제품 코드입니다.	String, Null
ec2:Public	"ec2:Public":"public-flag" 여기서 public-flag는 true false입니다. (AMI에 대해)	Boolean, Null
ec2:Region	"ec2:Region":"region-name" 여기에서 region-name은 리전 이름(예: us-west-2)입니다. 리전을 나열하려면 describe-regions 를 사용합니다. 이 조건 키는 모든 Amazon EC2 작업에 사용할 수 있습니다.	String, Null
ec2:RequesterVpc	"ec2:RequesterVpc":"vpc-arn" 여기서 vpc-arn은 VPC 피어링 연결에서 요청자 VPC에 대한 VPC ARN입니다.	ARN, Null
ec2:ResourceTag/ tag-key	"ec2:ResourceTag/tag-key":"tag-value" 여기에서 tag-key 및 tag-value는 태그-키 페어입니다.	String, Null
ec2:RootDeviceType	"ec2:RootDeviceType":"root-device-type-name" 여기에서 root-device-type-name은 ebs instance-store입니다.	String, Null
ec2:SnapshotTime	"ec2:SnapshotTime":"time" 여기에서 time은 스냅샷 생성 시간(예: 2013-06-01T00:00:00Z)입니다.	Date, Null
ec2:Subnet	"ec2:Subnet":"subnet-arn" 여기에서 subnet-arn은 서브넷 ARN입니다.	ARN, Null
ec2:Tenancy	"ec2:Tenancy":"tenancy-attribute" 여기에서 tenancy-attribute는 default dedicated입니다. host	String, Null

조건 키	키/값 쌍	평가 유형
ec2:VolumeIops	"ec2:VolumeIops":"volume-iops" 여기에서 volume-iops는 초당 입력/출력 작업 수(IOPS)이며 범위는 100 ~ 20,000입니다.	Numeric, Null
ec2:VolumeSize	"ec2:VolumeSize":"volume-size" 여기에서 volume-size는 볼륨 크기(GiB 단위)입니다.	Numeric, Null
ec2:VolumeType	"ec2:VolumeType":"volume-type-name" 여기에서 volume-type-name은 범용 SSD 볼륨의 경우 gp2, 프로비저닝된 IOPS SSD 볼륨의 경우 io1, 처리량에 최적화된 HDD 볼륨의 경우 st1, Cold HDD 볼륨의 경우 sc1 또는 Magnetic 볼륨의 경우 standard입니다.	String, Null
ec2:vpc	"ec2:Vpc":"vpc-arn" 여기에서 vpc-arn은 VPC ARN입니다.	ARN, Null

Amazon EC2는 AWS 차원의 조건 키도 구현합니다([사용 가능한 키](#) 참조). 다음의 AWS 조건 키는 현재 Amazon EC2에 고유합니다.

조건 키	키/값 쌍	평가 유형
aws:RequestTag/tag-key	"aws:Request/tag-key":"tag-value" 여기서 tag-key와 tag-value는 태그 키-값 페어입니다.	String, Null
aws:TagKeys	"aws:TagKeys":"tag-key" 여기에서 tag-key는 태그 키 목록(예: ["A","B"])입니다.	String, Null

어떠한 Amazon EC2 리소스에 작업별로 어떠한 조건 키를 사용할 수 있는지에 대한 자세한 내용은 [Amazon EC2 API 작업에 지원되는 리소스 수준 권한 \(p. 409\)](#) 섹션을 참조하십시오. Amazon EC2용 예제 정책 명령문은 [AWS CLI 또는 AWS SDK 작업을 위한 예제 정책 \(p. 431\)](#) 섹션을 참조하십시오.

사용자에게 필요한 권한이 있는지 확인

IAM 정책을 생성한 후에는 사용자에게 필요한 특정 API 작업 및 리소스를 사용할 권리가 제대로 부여되는지를 확인한 후에 정책을 실무에 적용하는 것이 좋습니다.

우선 테스트용으로 IAM 사용자를 생성하고 앞서 생성한 IAM 정책을 연결하여 사용자를 테스트합니다. 그런 다음 테스트 사용자 자격으로 요청을 수행합니다.

리소스를 생성하거나 수정하는 Amazon EC2 작업을 테스트하는 경우 DryRun 파라미터를 사용하여 요청하거나 --dry-run 옵션과 함께 AWS CLI 명령을 실행해야 합니다. 이렇게 하면 호출 시 권한 부여 확인은 완료되지만 작업은 완료되지 않습니다. 예를 들어 인스턴스를 실제로 종료하지 않고 사용자가 특정 인스턴스를 종료할 수 있는지 여부를 확인할 수 있습니다. 테스트 사용자에게 필요한 권한이 있는 경우 요청 시 DryRunOperation이 반환되고, 그렇지 않은 경우 UnauthorizedOperation이 반환됩니다.

정책이 사용자에게 정상적으로 권한을 부여하지 못하거나 권한을 과도하게 부여하는 경우, 원하는 결과가 나올 때까지 정책을 조정하고 다시 테스트할 수 있습니다.

Important

변경된 정책이 전파되어 효력을 발휘하려면 몇 분이 걸릴 수 있습니다. 따라서 정책을 업데이트한 경우 5분간 기다린 후에 테스트하는 것이 좋습니다.

요청 시 권한 부여 확인에 실패하면 진단 정보가 포함된 인코딩 메시지가 반환됩니다.
`DecodeAuthorizationMessage` 작업을 사용하여 메시지를 디코딩할 수 있습니다. 자세한 내용은 AWS Security Token Service API Reference의 [DecodeAuthorizationMessage](#) 및 AWS Command Line Interface Reference의 [decode-authorization-message](#) 섹션을 참조하십시오.

Amazon EC2 API 작업에 지원되는 리소스 수준 권한

리소스 수준 권한이란 사용자가 작업을 수행할 수 있는 리소스를 지정하는 기능을 말합니다. Amazon EC2에서는 리소스 수준 권한을 부분적으로 지원합니다. 즉, 필요 조건을 지정하거나 사용 가능한 특정 리소스를 지정하여 사용자가 특정 Amazon EC2 작업을 사용할 수 있는지 여부를 제어할 수 있습니다. 예를 들어 사용자에게 인스턴스 시작 권한을 부여하면서 특정 유형 또는 특정 AMI만 사용하도록 제한할 수 있습니다.

다음 표에서는 현재 리소스 수준 권한을 지원하는 Amazon EC2 API 작업 및 각 작업에 지원되는 리소스, 해당 ARN 및 조건 키를 보여 줍니다. ARN을 지정할 때, 예를 들어, 정확한 리소스 ID를 지정할 수 없거나 지정 하길 원치 않는 경우에는 경로에 * 와일드카드를 사용할 수 있습니다. 와일드카드의 용례는 [AWS CLI 또는 AWS SDK 작업을 위한 예제 정책 \(p. 431\)](#) 섹션을 참조하십시오.

Important

이 표에 기재되지 않은 Amazon EC2 API 작업은 리소스 수준 권한을 지원하지 않습니다. Amazon EC2 API 작업이 리소스 수준 권한을 지원하지 않는 경우 사용자에게 작업 사용 권한을 부여할 때 정책 명령문의 리소스 요소를 *로 지정해야 합니다. 문제 해결 예는 [1: 읽기 전용 액세스 \(p. 431\)](#) 단원을 참조하십시오. 현재 리소스 수준 권한을 지원하지 않는 Amazon EC2 API 작업의 목록은 Amazon EC2 API Reference에서 [지원되지 않는 리소스 수준 권한](#)을 참조하십시오.

모든 Amazon EC2 작업은 `ec2:Region` 조건 키를 지원합니다. 문제 해결 예는 [2: 특정 리전으로 액세스 제한 \(p. 432\)](#)을(를) 참조하십시오.

API 작업	리소스	조건 키
AcceptVpcPeeringConnection	VPC 피어링 연결 arn:aws:ec2:region:account:vpc-peering-connection/* arn:aws:ec2:region:account:vpc-peering-connection/vpc-peering-connection-id	ec2:AcceptorVpc ec2:Region ec2:ResourceTag/tag-key ec2:RequesterVpc
	VPC arn:aws:ec2:region:account:vpc/* arn:aws:ec2:region:account:vpc/vpc-id 여기에서 vpc-id는 수락자가 소유한 VPC입니다.	ec2:ResourceTag/tag-key ec2:Region ec2:Tenancy
AssociateIamInstanceProfile	인스턴스 arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/instance-id	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup

Amazon Elastic Compute Cloud
Linux 인스턴스용 사용 설명서
IAM 정책

API 작업	리소스	조건 키
		ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy
AttachClassicLinkVpc	인스턴스 arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ instance-id	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy
	보안 그룹 arn:aws:ec2:region:account:security- group/* arn:aws:ec2:region:account:security- group/security-group-id 여기에서 보안 그룹은 VPC에 대한 보안 그룹입니다.	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
	VPC arn:aws:ec2:region:account:vpc/* arn:aws:ec2:region:account:vpc/vpc-id	ec2:Region ec2:ResourceTag/tag-key ec2:Tenancy
AttachVolume	인스턴스 arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ instance-id	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy

Amazon Elastic Compute Cloud
Linux 인스턴스용 사용 설명서
IAM 정책

API 작업	리소스	조건 키
	볼륨 arn:aws:ec2:region:account:volume/* arn:aws:ec2:region:account:volume/volume-id	ec2:AvailabilityZone ec2:Encrypted ec2:ParentSnapshot ec2:Region ec2:ResourceTag/tag-key ec2:VolumeIops ec2:VolumeSize ec2:VolumeType
AuthorizeSecurityGroupEgress	外出 그룹 arn:aws:ec2:region:account:security-group/* arn:aws:ec2:region:account:security-group/security-group-id	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
AuthorizeSecurityGroupIngress	내부 그룹 arn:aws:ec2:region:account:security-group/* arn:aws:ec2:region:account:security-group/security-group-id	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
CreateTags	DHCP 옵션 세트 arn:aws:ec2:region:account:dhcp-options/* arn:aws:ec2:region:account:dhcp-options/dhcp-options-id	ec2:CreateAction ec2:Region ec2:ResourceTag/tag-key aws:RequestTag/tag-key aws:TagKeys
	이미지 arn:aws:ec2:region::image/* arn:aws:ec2:region::image/image-id	ec2:CreateAction ec2:ImageType ec2:Owner ec2:Public ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType aws:RequestTag/tag-key aws:TagKeys

API 작업	리소스	조건 키
	인스턴스 arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ instance-id	ec2:AvailabilityZone ec2>CreateAction ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy aws:RequestTag/tag-key aws:TagKeys
	인터넷 게이트웨이 arn:aws:ec2:region:account:internet-gateway/* arn:aws:ec2:region:account:internet-gateway/igw-id	ec2>CreateAction ec2:Region ec2:ResourceTag/tag-key aws:RequestTag/tag-key aws:TagKeys
	네트워크 ACL arn:aws:ec2:region:account:network-acl/* arn:aws:ec2:region:account:network-acl/nacl-id	ec2>CreateAction ec2:Region ec2:ResourceTag/tag-key ec2:Vpc aws:RequestTag/tag-key aws:TagKeys
	네트워크 인터페이스 arn:aws:ec2:region:account:network-interface/* arn:aws:ec2:region:account:network-interface/eni-id	ec2:AvailabilityZone ec2>CreateAction ec2:Region ec2:Subnet ec2:ResourceTag/tag-key ec2:Vpc

Amazon Elastic Compute Cloud
Linux 인스턴스용 사용 설명서
IAM 정책

API 작업	리소스	조건 키
		aws:RequestTag/tag-key aws:TagKeys
	예약 인스턴스 arn:aws:ec2:region:account:reserved-instance/* arn:aws:ec2:region:account:reserved-instance/reservation-id	ec2:AvailabilityZone ec2>CreateAction ec2:InstanceType ec2:ReservedInstancesOfferingType ec2:Region ec2:ResourceTag/tag-key ec2:Tenancy
		aws:RequestTag/tag-key aws:TagKeys
	라우팅 테이블 arn:aws:ec2:region:account:route-table/* arn:aws:ec2:region:account:route-table/route-table-id	ec2>CreateAction ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
		aws:RequestTag/tag-key aws:TagKeys
	보안 그룹 arn:aws:ec2:region:account:security-group/* arn:aws:ec2:region:account:security-group/security-group-id	ec2>CreateAction ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
		aws:RequestTag/tag-key aws:TagKeys
	스냅샷 arn:aws:ec2:region::snapshot/* arn:aws:ec2:region::snapshot/snapshot-id	ec2>CreateAction ec2:Owner ec2:ParentVolume ec2:Region ec2:ResourceTag/tag-key ec2:SnapshotTime ec2:VolumeSize

Amazon Elastic Compute Cloud
Linux 인스턴스용 사용 설명서
IAM 정책

API 작업	리소스	조건 키
		aws:RequestTag/tag-key aws:TagKeys
	스팟 인스턴스 요청 arn:aws:ec2:region:account:spot-instance-request/* arn:aws:ec2:region:account:spot-instance-request/spot-instance-request-id	ec2>CreateAction ec2:Region ec2:ResourceTag/tag-key aws:RequestTag/tag-key aws:TagKeys
	서브넷 arn:aws:ec2:region:account:subnet/* arn:aws:ec2:region:account:subnet/subnet-id	ec2:AvailabilityZone ec2>CreateAction ec2:Region ec2:ResourceTag/tag-key ec2:Vpc aws:RequestTag/tag-key aws:TagKeys
	볼륨 arn:aws:ec2:region:account:volume/* arn:aws:ec2:region:account:volume/volume-id	ec2:AvailabilityZone ec2>CreateAction ec2:Encrypted ec2:ParentSnapshot ec2:Region ec2:ResourceTag/tag-key ec2:VolumeLops ec2:VolumeSize ec2:VolumeType aws:RequestTag/tag-key aws:TagKeys
	VPC arn:aws:ec2:region:account:vpc/* arn:aws:ec2:region:account:vpc/vpc-id	ec2>CreateAction ec2:Region ec2:ResourceTag/tag-key ec2:Tenancy

Amazon Elastic Compute Cloud
Linux 인스턴스용 사용 설명서
IAM 정책

API 작업	리소스	조건 키
		aws:RequestTag/tag-key aws:TagKeys
	VPN 연결 arn:aws:ec2:region:account:vpn-connection/* arn:aws:ec2:region:account:vpn-connection/vpn-connection-id	ec2>CreateAction ec2:Region ec2:ResourceTag/tag-key
		aws:RequestTag/tag-key aws:TagKeys
	VPN 게이트웨이 arn:aws:ec2:region:account:vpn-gateway/* arn:aws:ec2:region:account:vpn-gateway/vpn-gateway-id	ec2>CreateAction ec2:Region ec2:ResourceTag/tag-key
		aws:RequestTag/tag-key aws:TagKeys
CreateVolume	볼륨 arn:aws:ec2:region:account:volume/*	ec2:AvailabilityZone ec2:Encrypted ec2:ParentSnapshot ec2:Region ec2:VolumeLops ec2:VolumeSize ec2:VolumeType
		aws:RequestTag/tag-key aws:TagKeys
CreateVpcPeeringConnection	VPC arn:aws:ec2:region:account:vpc/* arn:aws:ec2:region:account:vpc/vpc-id 여기에서 vpc-id는 요청자 VPC입니다.	ec2:ResourceTag/tag-key ec2:Region ec2:Tenancy
	VPC 피어링 연결 arn:aws:ec2:region:account:vpc-peering-connection/*	ec2:AcceptorVpc ec2:Region ec2:RequesterVpc

API 작업	리소스	조건 키
DeleteCustomerGateway	고객 게이트웨이 arn:aws:ec2:region:account:customer-gateway/* arn:aws:ec2:region:account:customer-gateway/cgw-id	ec2:Region ec2:ResourceTag/tag-key
DeleteDhcpOptions	DHCP 옵션 세트 arn:aws:ec2:region:account:dhcp-options/* arn:aws:ec2:region:account:dhcp-options/dhcp-options-id	ec2:Region ec2:ResourceTag/tag-key
DeleteInternetGateway	인터넷 게이트웨이 arn:aws:ec2:region:account:internet-gateway/* arn:aws:ec2:region:account:internet-gateway/igw-id	ec2:Region ec2:ResourceTag/tag-key
DeleteNetworkAcl	네트워크 ACL arn:aws:ec2:region:account:network-acl/* arn:aws:ec2:region:account:network-acl/nacl-id	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
DeleteNetworkAclEntry	네트워크 ACL arn:aws:ec2:region:account:network-acl/* arn:aws:ec2:region:account:network-acl/nacl-id	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
DeleteRoute	라우팅 테이블 arn:aws:ec2:region:account:route-table/* arn:aws:ec2:region:account:route-table/route-table-id	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
DeleteRoute	라우팅 테이블 arn:aws:ec2:region:account:route-table/* arn:aws:ec2:region:account:route-table/route-table-id	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
DeleteSecurityGroup	보안 그룹 arn:aws:ec2:region:account:security-group/security-group-id	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc

Amazon Elastic Compute Cloud
Linux 인스턴스용 사용 설명서
IAM 정책

API 작업	리소스	조건 키
DeleteTags	DHCP 옵션 세트 arn:aws:ec2:region:account:dhcp-options/* arn:aws:ec2:region:account:dhcp-options/dhcp-options-id	ec2:Region ec2:ResourceTag/tag-key aws:RequestTag/tag-key aws:TagKeys
	이미지 arn:aws:ec2:region::image/* arn:aws:ec2:region::image/image-id	ec2:Region ec2:ResourceTag/tag-key aws:RequestTag/tag-key aws:TagKeys
	인스턴스 arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/instance-id	ec2:Region ec2:ResourceTag/tag-key aws:RequestTag/tag-key aws:TagKeys
	인터넷 게이트웨이 arn:aws:ec2:region:account:internet-gateway/* arn:aws:ec2:region:account:internet-gateway/igw-id	ec2:Region ec2:ResourceTag/tag-key aws:RequestTag/tag-key aws:TagKeys
	네트워크 ACL arn:aws:ec2:region:account:network-acl/* arn:aws:ec2:region:account:network-acl-nacl-id	ec2:Region ec2:ResourceTag/tag-key aws:RequestTag/tag-key aws:TagKeys
	네트워크 인터페이스 arn:aws:ec2:region:account:network-interface/* arn:aws:ec2:region:account:network-interface/eni-id	ec2:Region ec2:ResourceTag/tag-key aws:RequestTag/tag-key aws:TagKeys
	예약 인스턴스 arn:aws:ec2:region:account:reserved-instance/* arn:aws:ec2:region:account:reserved-instance/reservation-id	ec2:Region ec2:ResourceTag/tag-key aws:RequestTag/tag-key aws:TagKeys

Amazon Elastic Compute Cloud
Linux 인스턴스용 사용 설명서
IAM 정책

API 작업	리소스	조건 키
라우팅 테이블	arn:aws:ec2:region:account:route-table/*	ec2:Region ec2:ResourceTag/tag-key
	arn:aws:ec2:region:account:route-table/route-table-id	aws:RequestTag/tag-key aws:TagKeys
	arn:aws:ec2:region:account:security-group/*	ec2:Region ec2:ResourceTag/tag-key
	arn:aws:ec2:region:account:security-group/security-group-id	aws:RequestTag/tag-key aws:TagKeys
보안 그룹	arn:aws:ec2:region::snapshot/*	ec2:Region ec2:ResourceTag/tag-key
	arn:aws:ec2:region::snapshot/snapshot-id	aws:RequestTag/tag-key aws:TagKeys
	arn:aws:ec2:region::snapshot/*	ec2:Region ec2:ResourceTag/tag-key
	arn:aws:ec2:region::snapshot/snapshot-id	aws:RequestTag/tag-key aws:TagKeys
스냅샷	arn:aws:ec2:region:account:spot-instance-request/*	ec2:Region ec2:ResourceTag/tag-key
	arn:aws:ec2:region:account:spot-instance-request/spot-instance-request-id	aws:RequestTag/tag-key aws:TagKeys
	arn:aws:ec2:region:account:subnet/*	ec2:Region ec2:ResourceTag/tag-key
	arn:aws:ec2:region:account:subnet/subnet-id	aws:RequestTag/tag-key aws:TagKeys
스팟 인스턴스 요청	arn:aws:ec2:region:account:subnet/*	ec2:Region ec2:ResourceTag/tag-key
	arn:aws:ec2:region:account:subnet/subnet-id	aws:RequestTag/tag-key aws:TagKeys
	arn:aws:ec2:region:account:volume/*	ec2:Region ec2:ResourceTag/tag-key
	arn:aws:ec2:region:account:volume/volume-id	aws:RequestTag/tag-key aws:TagKeys
서브넷	arn:aws:ec2:region:account:vpc/*	ec2:Region ec2:ResourceTag/tag-key
	arn:aws:ec2:region:account:vpc/vpc-id	aws:RequestTag/tag-key aws:TagKeys
	arn:aws:ec2:region:account:vpc/*	ec2:Region ec2:ResourceTag/tag-key
	arn:aws:ec2:region:account:vpc/vpc-id	aws:RequestTag/tag-key aws:TagKeys
볼륨	arn:aws:ec2:region:account:vpc/*	ec2:Region ec2:ResourceTag/tag-key
	arn:aws:ec2:region:account:vpc/vpc-id	aws:RequestTag/tag-key aws:TagKeys
	arn:aws:ec2:region:account:vpc/*	ec2:Region ec2:ResourceTag/tag-key
	arn:aws:ec2:region:account:vpc/vpc-id	aws:RequestTag/tag-key aws:TagKeys
VPC	arn:aws:ec2:region:account:vpc/*	ec2:Region ec2:ResourceTag/tag-key
	arn:aws:ec2:region:account:vpc/vpc-id	aws:RequestTag/tag-key aws:TagKeys
	arn:aws:ec2:region:account:vpc/*	ec2:Region ec2:ResourceTag/tag-key
	arn:aws:ec2:region:account:vpc/vpc-id	aws:RequestTag/tag-key aws:TagKeys

API 작업	리소스	조건 키
	VPN 연결 arn:aws:ec2:region:account:vpn-connection/* arn:aws:ec2:region:account:vpn-connection/vpn-connection-id	ec2:Region ec2:ResourceTag/tag-key aws:RequestTag/tag-key aws:TagKeys
	VPN 게이트웨이 arn:aws:ec2:region:account:vpn-gateway/* arn:aws:ec2:region:account:vpn-gateway/vpn-gateway-id	ec2:Region ec2:ResourceTag/tag-key aws:RequestTag/tag-key aws:TagKeys
DeleteVolume	볼륨 arn:aws:ec2:region:account:volume/* arn:aws:ec2:region:account:volume/volume-id	ec2:AvailabilityZone ec2:Encrypted ec2:ParentSnapshot ec2:Region ec2:ResourceTag/tag-key ec2:Volumelops ec2:VolumeSize ec2:VolumeType
DeleteVpcPeeringConnection	VPC 피어링 연결 arn:aws:ec2:region:account:vpc-peering-connection/* arn:aws:ec2:region:account:vpc-peering-connection/vpc-peering-connection-id	ec2:AcceptorVpc ec2:Region ec2:ResourceTag/tag-key ec2:RequesterVpc
DetachClassicLinkVpc	인스턴스 arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/instance-id	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy

Amazon Elastic Compute Cloud
Linux 인스턴스용 사용 설명서
IAM 정책

API 작업	리소스	조건 키
	VPC arn:aws:ec2:region:account:vpc/* arn:aws:ec2:region:account:vpc/vpc-id	ec2:Region ec2:ResourceTag/tag-key ec2:Tenancy
DetachVolume	인스턴스 arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/instance-id	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy
	볼륨 arn:aws:ec2:region:account:volume/* arn:aws:ec2:region:account:volume/volume-id	ec2:AvailabilityZone ec2:Encrypted ec2:ParentSnapshot ec2:Region ec2:ResourceTag/tag-key ec2:Volumelops ec2:VolumeSize ec2:VolumeType
DisableVpcClassicLink	VPC arn:aws:ec2:region:account:vpc/* arn:aws:ec2:region:account:vpc/vpc-id	ec2:Region ec2:ResourceTag/tag-key ec2:Tenancy

API 작업	리소스	조건 키
DisassociateIamInstanceProfile	인스턴스 arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ instance-id	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy
EnableVpcClassicLink	VPC arn:aws:ec2:region:account:vpc/* arn:aws:ec2:region:account:vpc/vpc-id	ec2:Region ec2:ResourceTag/tag-key ec2:Tenancy
GetConsoleScreenshot	인스턴스 arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ instance-id	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy
RebootInstances	인스턴스 arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ instance-id	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy

Amazon Elastic Compute Cloud
Linux 인스턴스용 사용 설명서
IAM 정책

API 작업	리소스	조건 키
RejectVpcPeeringConnection	VPC 피어링 연결 arn:aws:ec2:region:account:vpc-peering-connection/* arn:aws:ec2:region:account:vpc-peering-connection/vpc-peering-connection-id	ec2:AcceptorVpc ec2:Region ec2:ResourceTag/tag-key ec2:RequesterVpc
ReplaceIamInstanceProfileAssignment	aws:Ec2:Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/instance-id	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy
RevokeSecurityGroupEgress	보안 그룹 arn:aws:ec2:region:account:security-group/* arn:aws:ec2:region:account:security-group/security-group-id	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
RevokeSecurityGroupEgress	보안 그룹 arn:aws:ec2:region:account:security-group/* arn:aws:ec2:region:account:security-group/security-group-id	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
RunInstances	이미지 arn:aws:ec2:region::image/* arn:aws:ec2:region::image/image-id	ec2:ImageType ec2:Owner ec2:Public ec2:Region ec2:RootDeviceType ec2:ResourceTag/tag-key

API 작업	리소스	조건 키
	인스턴스 arn:aws:ec2:region:account:instance/*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy
		aws:RequestTag/tag-key aws:TagKeys
	키 페어 arn:aws:ec2:region:account:key-pair/* arn:aws:ec2:region:account:key-pair/key-pair-name	ec2:Region
	네트워크 인터페이스 arn:aws:ec2:region:account:network-interface/* arn:aws:ec2:region:account:network-interface/eni-id	ec2:AvailabilityZone ec2:Region ec2:Subnet ec2:ResourceTag/tag-key ec2:Vpc
	배치 그룹 arn:aws:ec2:region:account:placement-group/* arn:aws:ec2:region:account:placement-group/placement-group-name	ec2:Region ec2:PlacementGroupStrategy
	보안 그룹 arn:aws:ec2:region:account:security-group/* arn:aws:ec2:region:account:security-group/security-group-id	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc

Amazon Elastic Compute Cloud
Linux 인스턴스용 사용 설명서
IAM 정책

API 작업	리소스	조건 키
	스냅샷 arn:aws:ec2:region::snapshot/* arn:aws:ec2:region::snapshot/snapshot-id	ec2:Owner ec2:ParentVolume ec2:Region ec2:SnapshotTime ec2:ResourceTag/tag-key ec2:VolumeSize
	서브넷 arn:aws:ec2:region:account:subnet/* arn:aws:ec2:region:account:subnet/subnet-id	ec2:AvailabilityZone ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
	볼륨 arn:aws:ec2:region:account:volume/*	ec2:AvailabilityZone ec2:Encrypted ec2:ParentSnapshot ec2:Region ec2:ResourceTag/tag-key ec2:VolumeLops ec2:VolumeSize ec2:VolumeType
		aws:RequestTag/tag-key aws:TagKeys
StartInstances	인스턴스 arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/instance-id	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy

API 작업	리소스	조건 키
StopInstances	인스턴스 arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ instance-id	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy
TerminateInstances	인스턴스 arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ instance-id	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy

RunInstances에 대한 리소스 수준의 권한

[RunInstances](#) API 작업은 하나 이상의 인스턴스를 시작하고 다수의 Amazon EC2 리소스를 생성 및 사용합니다. 이 작업은 AMI를 필요로 하며 인스턴스를 생성하는데, 인스턴스는 보안 그룹과 연결되어야 합니다. VPC로 시작하는 경우 서브넷을 입력 받아 네트워크 인터페이스를 생성합니다. Amazon EBS 지원 AMI에서 시작하면 볼륨이 생성됩니다. `ec2:RunInstances` 작업에 리소스 수준 권한을 사용하는 정책의 `Resource` 요소에서 지정될 수 있도록 사용자가 이들 리소스를 사용할 권한을 보유해야 합니다. `ec2:RunInstances` 작업에 리소스 수준 권한을 사용하지 않으려면 명령문의 `Resource` 요소에 개별 ARN 대신 * 와일드카드를 지정할 수 있습니다.

리소스 수준 권한을 사용하는 경우, 다음 표에 `ec2:RunInstances` 작업을 사용하는 데 필요한 최소 리소스가 설명되어 있습니다.

시작 유형	필요 리소스	조건 키
인스턴스 스토어 기반 AMI를 사용하는 EC2-Classic으로 시작	arn:aws:ec2:region:account:instance/*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile

시작 유형	필요 리소스	조건 키
		ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:RootDeviceType ec2:Tenancy
	arn:aws:ec2:region::image/*(또는 특정 AMI ID)	ec2:ImageType ec2:Owner ec2:Public ec2:Region ec2:RootDeviceType ec2:ResourceTag/tag-key
	arn:aws:ec2:region:account:securitygroup/*(또는 특정 보안 그룹 ID)	ec2:ResourceTag/tag-key ec2:Vpc
Amazon EBS 기반 AMI를 사용하는 EC2-Classic으로 시작	arn:aws:ec2:region:account:instance*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:RootDeviceType ec2:Tenancy
	arn:aws:ec2:region::image/*(또는 특정 AMI ID)	ec2:ImageType ec2:Owner ec2:Public ec2:Region ec2:RootDeviceType ec2:ResourceTag/tag-key

시작 유형	필요 리소스	조건 키
	arn:aws:ec2:region:account:securitygroup/*(또는 특정 보안 그룹 ID)	ec2:ResourceTag/tag-key ec2:Vpc
	arn:aws:ec2:region:account:volume/*	ec2:ParentSnapshot ec2:Region ec2:VolumeLops ec2:VolumeSize ec2:VolumeType
인스턴스 스토어 기반 AMI를 사용하는 VPC로 시작	arn:aws:ec2:region:account:instance/*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:RootDeviceType ec2:Tenancy
	arn:aws:ec2:region::image/*(또는 특정 AMI ID)	ec2:ImageType ec2:Owner ec2:Public ec2:Region ec2:RootDeviceType ec2:ResourceTag/tag-key
	arn:aws:ec2:region:account:securitygroup/*(또는 특정 보안 그룹 ID)	ec2:ResourceTag/tag-key ec2:Vpc

시작 유형	필요 리소스	조건 키
	arn:aws:ec2:region:account:networkinterface/*(또는 특정 네트워크 인터페이스 ID)	ec2:AvailabilityZone ec2:Region ec2:Subnet ec2:ResourceTag/tag-key ec2:Vpc
	arn:aws:ec2:region:account:subnet/*(또는 특정 서브넷 ID)	ec2:AvailabilityZone ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
Amazon EBS 기반 AMI를 사용하는 VPC로 시작	arn:aws:ec2:region:account:instance/*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:RootDeviceType ec2:Tenancy
	arn:aws:ec2:region::image/*(또는 특정 AMI ID)	ec2:ImageType ec2:Owner ec2:Public ec2:Region ec2:RootDeviceType ec2:ResourceTag/tag-key
	arn:aws:ec2:region:account:securitygroup/*(또는 특정 보안 그룹 ID)	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc

시작 유형	필요 리소스	조건 키
	arn:aws:ec2:region:account:networkinterface/*(또는 특정 네트워크 인터페이스 ID)	ec2:AvailabilityZone ec2:Region ec2:Subnet ec2:ResourceTag/tag-key ec2:Vpc
	arn:aws:ec2:region:account:volume/*	ec2:Encrypted ec2:ParentSnapshot ec2:Region ec2:VolumeLops ec2:VolumeSize ec2:VolumeType
	arn:aws:ec2:region:account:subnet/*(또는 특정 서브넷 ID)	ec2:AvailabilityZone ec2:Region ec2:ResourceTag/tag-key ec2:Vpc

정책에서 키 페어 리소스도 지정할 것을 권장합니다. — 인스턴스를 시작하는 데는 필요하지 않지만 키 페어가 없으면 인스턴스에 연결할 수 없습니다. `ec2:RunInstances` 작업에서 리소스 수준 권한을 사용하는 예제는 [5: 인스턴스 시작\(RunInstances\) \(p. 436\)](#) 섹션을 참조하십시오.

Amazon EC2의 리소스 수준 권한에 대한 자세한 내용은 AWS 보안 블로그 게시물 [Demystifying EC2 Resource-Level Permissions](#) 섹션을 참조하십시오.

태그 지정을 위한 리소스 수준 권한

일부 리소스 생성 Amazon EC2 API 작업에서는 리소스를 생성할 때 태그를 지정할 수 있습니다. 자세한 내용은 [리소스에 태그 지정 \(p. 681\)](#) 단원을 참조하십시오.

사용자가 생성 시 리소스에 태그를 지정할 수 있으려면 리소스를 생성하는 작업을 사용할 권한이 있어야 합니다(예: `ec2:RunInstances` 또는 `ec2>CreateVolume`). 리소스 생성 작업에서 태그가 지정되면 Amazon은 `ec2:CreateTags` 작업에서 추가 권한 부여를 수행해 사용자에게 태그를 생성할 권한이 있는지 확인합니다. 따라서 사용자는 `ec2:CreateTags` 작업을 사용할 명시적 권한도 가지고 있어야 합니다.

`ec2:CreateTags` 작업의 경우, `ec2:CreateAction` 조건 키를 사용하여 태그 지정 권한을 리소스 생성 작업으로만 제한할 수 있습니다. 예를 들어 다음 정책은 사용자가 인스턴스를 시작하고 시작 도중 인스턴스와 볼륨에 임의의 태그를 적용하는 것을 허용합니다. 사용자는 기존 리소스에 태그를 지정할 수 없습니다(`ec2:CreateTags` 작업을 직접 호출할 수 없습니다).

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances"
    }
  ]
}
```

```
"Action": [
    "ec2:RunInstances"
],
"Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:CreateTags"
    ],
    "Resource": "arn:aws:ec2:region:account:*/",
    "Condition": {
        "StringEquals": {
            "ec2:CreateAction" : "RunInstances"
        }
    }
}
]
```

마찬가지로 다음 정책은 사용자가 볼륨을 생성하고 볼륨 생성 도중 볼륨에 임의의 태그를 적용하는 것을 허용합니다. 사용자는 기존 리소스에 태그를 지정할 수 없습니다(ec2:CreateTags 작업을 직접 호출할 수 없습니다).

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateVolume"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:region:account:*/",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction" : "CreateVolume"
                }
            }
        }
    ]
}
```

ec2:CreateTags 작업은 리소스 생성 작업 도중 태그가 적용되는 경우에만 평가됩니다. 따라서 리소스를 생성할 권한이 있는 사용자(태그 지정 조건은 없다고 가정)는 요청에서 태그가 지정되지 않은 경우, ec2:CreateTags 작업을 사용할 권리가 필요하지 않습니다. 하지만 사용자가 태그를 사용하여 리소스 생성을 시도하는 경우, 사용자에게 ec2:CreateTags 작업을 사용할 권리가 없다면 요청은 실패합니다.

다음 조건 키를 사용하여 리소스에 적용되는 태그 키와 값을 제어할 수 있습니다.

- **aws:RequestTag**: 특정 태그 키 또는 태그 키 및 값이 요청에 존재해야 함을 표시. 요청에서 다른 태그도 지정할 수 있습니다.
 - 특정한 태그와 키의 조합을 적용하려면(예를 들어 태그 StringEquals=cost-center:를 적용하려면) cc123 조건 연산자와 함께 사용하십시오.

```
"StringEquals": "aws:RequestTag/cost-center": "cc123"
```

- 요청에서 특정 태그 키를 적용하려면(예를 들어 태그 키 `purpose`를 적용하려면) `StringLike` 조건 연산자와 함께 사용하십시오.

```
"StringLike": "aws:RequestTag/purpose": "*"
```

- `aws:TagKeys`: 요청에서 사용되는 태그 키를 적용.

- 요청 시 지정하려면 `ForAllValues` 변경자와 함께 특정 태그 키를 적용하십시오(요청에서 태그가 지정되면 특정 태그 키만 허용되고 다른 태그는 허용되지 않습니다). 예를 들어 태그 키 `environment` 또는 `cost-center`가 허용됩니다.

```
"ForAllValues:StringEquals": { "aws:TagKeys": [ "environment", "cost-center" ] }
```

- 요청에서 지정된 태그 키 중 최소한 1개의 존재를 적용하려면 `ForAnyValue` 변경자와 함께 사용하십시오. 예를 들어 요청에 태그 키 `environment` 또는 `webserver` 중 최소한 1개가 존재해야 합니다.

```
"ForAnyValue:StringEquals": { "aws:TagKeys": [ "environment", "webserver" ] }
```

이들 조건 키는 `ec2:CreateTags` 및 `ec2:DeleteTags` 작업뿐 아니라 태그 지정을 지원하는 리소스 생성 작업에 적용될 수 있습니다.

사용자가 리소스를 생성할 때 강제로 태그를 지정하도록 하려면 리소스 생성 작업에서 `aws:RequestTag` 조건 키 또는 `aws:TagKeys` 조건 키를 `ForAnyValue` 변경자와 함께 사용해야 합니다. 이때 사용자가 리소스 생성 시 태그를 지정하지 않으면 `ec2:CreateTags` 작업이 평가되지 않습니다.

태그 키와 값은 대/소문자를 구분합니다.

다중 값 조건에 대한 자세한 내용은 IAM 사용 설명서의 [다중 키 값을 테스트하는 조건 생성](#) 단원을 참조하십시오. 예제 IAM 정책은 [AWS CLI 또는 AWS SDK 작업을 위한 예제 정책](#) (p. 431) 단원을 참조하십시오.

AWS CLI 또는 AWS SDK 작업을 위한 예제 정책

다음 예제는 IAM 사용자가 갖는 Amazon EC2 관련 권한을 제어하는 데 사용할 수 있는 정책 명령문을 보여 줍니다. 이러한 정책은 AWS CLI 또는 AWS SDK를 통한 요청에 맞게 설계되었습니다. Amazon EC2 콘솔 작업과 관련된 예제 정책은 [Amazon EC2 콘솔 작업을 위한 예제 정책](#) (p. 449) 섹션을 참조하십시오. Amazon VPC별 IAM 정책의 예제는 [Amazon VPC 리소스에 대한 액세스 제어](#) 섹션을 참조하십시오..

목차

- [1: 읽기 전용 액세스](#) (p. 431)
- [2: 특정 리전으로 액세스 제한](#) (p. 432)
- [3: 인스턴스 작업](#) (p. 432)
- [4: 볼륨 작업](#) (p. 434)
- [5: 인스턴스 시작\(RunInstances\)](#) (p. 436)
- [6: ClassicLink 작업](#) (p. 443)
- [7: 예약 인스턴스 사용](#) (p. 445)
- [8: 리소스에 태그 지정](#) (p. 446)
- [9: IAM 역할 작업](#) (p. 448)

1: 읽기 전용 액세스

다음 정책은 이름이 `Describe`로 시작되는 모든 Amazon EC2 API 작업을 사용할 권한을 부여합니다. Resource 요소에 와일드카드가 사용되었으므로 사용자가 이러한 API 작업에 모든 리소스를 지정할 수 있습

니다. API 작업이 리소스 수준 권한을 지원하지 않는 경우에도 * 와일드카드가 필요합니다. 어떠한 Amazon EC2 API 작업에 어떠한 ARN을 사용할 수 있는지에 대한 자세한 내용은 [Amazon EC2 API 작업에 지원되는 리소스 수준 권한 \(p. 409\)](#) 섹션을 참조하십시오.

다른 명령문으로 해당 권한을 부여하지 않는 경우 리소스에 대해 작업을 수행할 권한은 부여되지 않습니다. 해당 API 작업을 사용할 권한은 기본적으로 거부됩니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:Describe*",  
            "Resource": "*"  
        }  
    ]  
}
```

2: 특정 리전으로 액세스 제한

다음 정책은 사용자에게 EU(프랑크푸르트)에서만 Amazon EC2 API 작업을 모두 사용할 수 있는 권한을 부여합니다. 사용자는 다른 리전에서 리소스를 확인, 생성, 수정 또는 삭제할 수 없습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:*",  
            "Resource": "*",  
            "Condition": {  
                "StringEquals": {  
                    "ec2:Region": "eu-central-1"  
                }  
            }  
        }  
    ]  
}
```

3: 인스턴스 작업

항목

- [모든 인스턴스를 설명, 실행, 중지, 시작 및 종료 \(p. 432\)](#)
- [모든 인스턴스를 설명할 수 있지만 특정 인스턴스만 중지, 시작 및 종료 \(p. 433\)](#)

모든 인스턴스를 설명, 실행, 중지, 시작 및 종료

다음 정책은 Action 요소에 지정된 API 작업을 사용할 권한을 부여합니다. Resource 요소에 * 와일드카드가 사용되었으므로 사용자가 이러한 API 작업에 모든 리소스를 지정할 수 있습니다. API 작업이 리소스 수준 권한을 지원하지 않는 경우에도 * 와일드카드가 필요합니다. 어떠한 Amazon EC2 API 작업에 어떠한 ARN을 사용할 수 있는지에 대한 자세한 내용은 [Amazon EC2 API 작업에 지원되는 리소스 수준 권한 \(p. 409\)](#) 섹션을 참조하십시오.

다른 명령문으로 해당 권한을 부여하지 않는 경우 다른 API 작업을 사용할 권한은 부여되지 않습니다. 해당 API 작업을 사용할 권한은 기본적으로 거부됩니다.

```
{  
    "Version": "2012-10-17",
```

```
"Statement": [
    "Effect": "Allow",
    "Action": [
        "ec2:DescribeInstances", "ec2:DescribeImages",
        "ec2:DescribeKeyPairs", "ec2:DescribeSecurityGroups",
        "ec2:DescribeAvailabilityZones",
        "ec2:RunInstances", "ec2:TerminateInstances",
        "ec2:StopInstances", "ec2:StartInstances"
    ],
    "Resource": "*"
]
}
```

모든 인스턴스를 설명할 수 있지만 특정 인스턴스만 중지, 시작 및 종료

다음 정책은 모든 인스턴스를 설명하고, 인스턴스 i-1234567890abcdef0 및 i-0598c7d356eba48d7만 시작 및 종지하고, 리소스 태그가 "purpose=test"인 미국 동부(버지니아 북부) 지역의 인스턴스(us-east-1)만 종료하도록 허용합니다.

첫 번째 명령문의 Resource 요소에 * 와일드카드가 사용되었으므로 사용자가 작업에 모든 리소스를 지정할 수 있습니다. 여기에서는 모든 인스턴스를 나열할 수 있습니다. API 작업(여기에서는 ec2:DescribeInstances)이 리소스 수준 권한을 지원하지 않는 경우에도 * 와일드카드가 필요합니다. 어떠한 Amazon EC2 API 작업에 어떠한 ARN을 사용할 수 있는지에 대한 자세한 내용은 [Amazon EC2 API 작업에 지원되는 리소스 수준 권한 \(p. 409\)](#) 섹션을 참조하십시오.

두 번째 명령문의 StopInstances 및 StartInstances 작업에는 리소스 수준 권한이 사용되었습니다. Resource 요소의 ARN에 의해 특정 인스턴스가 지정되었습니다.

세 번째 명령문은 사용자가 지정된 AWS 계정에 속하며 "purpose=test" 태그를 갖는 미국 동부(버지니아 북부) 지역의 모든 인스턴스(us-east-1)를 종료하도록 허용합니다. Condition 요소는 정책 명령문 적용 시에 평가됩니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeInstances",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:StopInstances",
                "ec2:StartInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0",
                "arn:aws:ec2:us-east-1:123456789012:instance/i-0598c7d356eba48d7"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "ec2:TerminateInstances",
            "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/purpose": "test"
                }
            }
        }
    ]
}
```

```
    ]  
}
```

4: 볼륨 작업

항목

- [볼륨 연결 및 연결 해제 \(p. 434\)](#)
- [볼륨 생성 \(p. 434\)](#)
- [태그를 사용하여 볼륨 생성 \(p. 435\)](#)

볼륨 연결 및 연결 해제

API 작업의 호출자가 여러 리소스를 지정해야 하는 경우 사용자가 필요한 모든 리소스에 액세스하도록 허용하는 정책 명령문을 생성해야 합니다. 이러한 리소스가 하나 이상 포함된 `Condition` 요소를 사용해야 하는 경우 이 예제와 같이 여러 명령문을 생성해야 합니다.

다음 정책은 "volume_user=iam-user-name" 태그가 있는 볼륨을 "department=dev" 태그가 있는 인스턴스에 연결하고 해당 볼륨을 해당 인스턴스에서 분리하도록 허용합니다. IAM 그룹에 이 정책을 연결하면 `aws:username` 정책 변수가 그룹의 각 IAM 사용자에게 자신의 IAM 사용자 이름을 값으로 하는 `volume_user`라는 태그가 있는 인스턴스에 볼륨을 연결하거나 분리할 권한을 부여합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:AttachVolume",  
                "ec2:DetachVolume"  
            ],  
            "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*",  
            "Condition": {  
                "StringEquals": {  
                    "ec2:ResourceTag/department": "dev"  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:AttachVolume",  
                "ec2:DetachVolume"  
            ],  
            "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",  
            "Condition": {  
                "StringEquals": {  
                    "ec2:ResourceTag/volume_user": "${aws:username}"  
                }  
            }  
        }  
    ]  
}
```

볼륨 생성

다음 정책은 사용자가 [CreateVolume](#) API 작업을 사용하는 것을 허용합니다. 사용자는 볼륨이 암호화되고 볼륨 크기가 20GB 미만인 경우에만 볼륨을 생성할 수 있습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:CreateVolume",  
            "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",  
            "Condition": {  
                "StringNotEquals": {  
                    "ec2:VolumeSize": "20"  
                }  
            }  
        }  
    ]  
}
```

```
"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "ec2:CreateVolume"
        ],
        "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",
        "Condition": {
            "NumericLessThan": {
                "ec2:VolumeSize" : "20"
            },
            "Bool": {
                "ec2:Encrypted" : "true"
            }
        }
    }
]
```

태그를 사용하여 볼륨 생성

다음 정책에는 사용자가 태그 costcenter=115 및 stack=prod를 사용하여 생성하는 볼륨에 태그를 지정해야 하는 aws:RequestTag 조건 키가 포함됩니다. aws:TagKeys 조건 키는 ForAllValues 변경자를 사용하여 요청에서 키 costcenter 및 stack만 허용됨을 표시합니다(다른 어떤 태그도 지정할 수 없습니다). 사용자가 이 특정 키들을 전달하지 않거나 태그를 전혀 지정하지 않으면 요청은 실패합니다.

태그를 적용하는 리소스 생성 작업의 경우, 사용자에게 CreateTags 작업을 사용할 권한도 있어야 합니다. 두 번째 문은 ec2:CreateAction 조건 키를 사용하여 사용자가 CreateVolume의 컨텍스트에서만 태그를 생성하도록 허용합니다. 사용자는 기존의 볼륨이나 다른 어떤 리소스에도 태그를 지정할 수 없습니다. 자세한 내용은 [태그 지정을 위한 리소스 수준 권한 \(p. 429\)](#) 단원을 참조하십시오.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateTaggedVolumes",
            "Effect": "Allow",
            "Action": "ec2:CreateVolume",
            "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/costcenter": "115",
                    "aws:RequestTag/stack": "prod"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": ["costcenter", "stack"]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction" : "CreateVolume"
                }
            }
        }
    ]
}
```

다음 정책은 사용자가 태그를 지정하지 않고 볼륨을 생성하는 것을 허용합니다. `CreateTags` 작업은 `CreateVolume` 요청에서 태그가 지정되는 경우에만 평가됩니다. 사용자가 태그를 지정하는 경우, 태그는 `purpose=test`여야 합니다. 다른 어떤 태그도 요청에서 허용되지 않습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:CreateVolume",  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:CreateTags"  
            ],  
            "Resource": "arn:aws:ec2:us-east-1:1234567890:volume/*",  
            "Condition": {  
                "StringEquals": {  
                    "aws:RequestTag/purpose": "test",  
                    "ec2:CreateAction" : "CreateVolume"  
                },  
                "ForAllValues:StringEquals": {  
                    "aws:TagKeys": "purpose"  
                }  
            }  
        }  
    ]  
}
```

5: 인스턴스 시작(RunInstances)

`RunInstances` API 작업은 하나 이상의 인스턴스를 시작합니다. `RunInstances`는 AMI를 입력 받아 인스턴스를 생성하며, 사용자는 요청에서 키 페어와 보안 그룹을 지정할 수 있습니다. EC2-VPC로 시작하는 경우 서브넷을 입력 받아 네트워크 인터페이스를 생성합니다. Amazon EBS 지원 AMI에서 시작하면 볼륨이 생성됩니다. 따라서 사용자에게 해당 Amazon EC2 리소스를 사용할 권한이 있어야 합니다. 또한 호출자는 `RunInstances`에 인스턴스 유형, 서브넷 등의 선택적 파라미터를 전달하여 인스턴스를 구성할 수 있습니다. 사용자가 선택적 파라미터를 반드시 지정하도록 요구하거나 파라미터에 특정 값을 사용할 수 없도록 제한하는 정책 명령문을 생성할 수 있습니다. 이 섹션의 예제에서는 사용자가 시작할 수 있는 인스턴스의 구성을 제어하는 몇 가지 방법을 보여 줍니다.

기본적으로는 사용자에게 결과 인스턴스를 설명, 시작, 중지 또는 종료할 권한이 없습니다. 사용자에게 결과 인스턴스를 관리할 권한을 부여하는 방법 중 하나는 각 인스턴스에 대한 특정 태그를 생성하고 해당 태그를 갖는 인스턴스를 관리하도록 허용하는 명령문을 생성하는 것입니다. 자세한 내용은 [3: 인스턴스 작업 \(p. 432\)](#)를 참조하십시오.

항목

- [AMI \(p. 436\)](#)
- [인스턴스 유형 \(p. 438\)](#)
- [서브넷 \(p. 439\)](#)
- [EBS 볼륨 \(p. 440\)](#)
- [태그 적용 \(p. 441\)](#)

AMI

다음 정책은 지정된 태그("department=dev")가 연결된 AMI만 사용하여 인스턴스를 시작하도록 허용합니다. 첫 번째 명령문의 condition 요소에서 사용자가 이 태그를 갖는 AMI를 지정하도록 요구하므로 다른 AMI

를 사용하여 인스턴스를 시작할 수 없습니다. 또한 정책에서 서브넷 및 네트워크 인터페이스 리소스에 대한 권한을 부여하지 않으므로 서브넷으로 시작할 수도 없습니다. 그러나 EC2-Classic으로 시작할 수는 있습니다. 두 번째 명령문에서는 와일드카드를 사용하여 인스턴스 리소스 생성을 허용하고, 사용자가 키 페어 `project_keypair` 및 보안 그룹 `sg-1a2b3c4d`를 지정하도록 요구합니다. 키 페어 없이도 인스턴스를 시작할 수는 있습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Effect": "Allow",  
         "Action": "ec2:RunInstances",  
         "Resource": [  
             "arn:aws:ec2:region::image/ami-*"  
         ],  
         "Condition": {  
             "StringEquals": {  
                 "ec2:ResourceTag/department": "dev"  
             }  
         }  
     },  
     {  
         "Effect": "Allow",  
         "Action": "ec2:RunInstances",  
         "Resource": [  
             "arn:aws:ec2:region:account:instance/*",  
             "arn:aws:ec2:region:account:volume/*",  
             "arn:aws:ec2:region:account:key-pair/project_keypair",  
             "arn:aws:ec2:region:account:security-group/sg-1a2b3c4d"  
         ]  
     }  
    ]  
}
```

또는 다음 정책으로 사용자가 지정된 AMI(`ami-9e1670f7` 및 `ami-45cf5c3c`)만 사용하여 인스턴스를 시작하도록 허용할 수 있습니다. 다른 명령문에서 해당 권한을 부여하지 않는 경우 다른 AMI를 사용하여 인스턴스를 시작할 수 없으며, 인스턴스를 서브넷으로 시작할 수도 없습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Effect": "Allow",  
         "Action": "ec2:RunInstances",  
         "Resource": [  
             "arn:aws:ec2:region::image/ami-9e1670f7",  
             "arn:aws:ec2:region::image/ami-45cf5c3c",  
             "arn:aws:ec2:region:account:instance/*",  
             "arn:aws:ec2:region:account:volume/*",  
             "arn:aws:ec2:region:account:key-pair/*",  
             "arn:aws:ec2:region:account:security-group/*"  
         ]  
     }  
    ]  
}
```

또는 다음 정책으로 Amazon이 소유한 모든 AMI에서 인스턴스를 시작하도록 허용할 수 있습니다. 첫 번째 명령문의 `Condition` 요소는 `ec2:Owner`가 `amazon`인지 여부를 테스트합니다. 다른 명령문에서 해당 권한을 부여하지 않는 경우 다른 AMI를 사용하여 인스턴스를 시작할 수 없습니다. 인스턴스를 서브넷으로 시작할 수 있습니다.

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

```
"Effect": "Allow",
"Action": "ec2:RunInstances",
"Resource": [
    "arn:aws:ec2:region::image/ami-*"
],
"Condition": {
    "StringEquals": {
        "ec2:Owner": "amazon"
    }
}
},
{
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
        "arn:aws:ec2:region:account:instance/*",
        "arn:aws:ec2:region:account:subnet/*",
        "arn:aws:ec2:region:account:volume/*",
        "arn:aws:ec2:region:account:network-interface/*",
        "arn:aws:ec2:region:account:key-pair/*",
        "arn:aws:ec2:region:account:security-group/*"
    ]
}
]
```

인스턴스 유형

다음 정책은 사용자가 t2.micro 및 t2.small 인스턴스 유형만 사용하여 인스턴스를 시작하도록 허용하므로 비용 통제에 도움이 됩니다. 첫 번째 명령문의 Condition 요소에서 ec2:InstanceType이 t2.micro 또는 t2.small인지 여부를 테스트하므로 더욱 큰 인스턴스는 시작할 수 없습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:region:account:instance/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:InstanceType": ["t2.micro", "t2.small"]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:region::image/ami-*",
                "arn:aws:ec2:region:account:subnet/*",
                "arn:aws:ec2:region:account:network-interface/*",
                "arn:aws:ec2:region:account:volume/*",
                "arn:aws:ec2:region:account:key-pair/*",
                "arn:aws:ec2:region:account:security-group/*"
            ]
        }
    ]
}
```

또는 t2.micro 및 t2.small 인스턴스 유형을 제외한 모든 인스턴스를 시작하기 위한 사용자 권한을 거부하는 정책을 생성할 수 있습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Effect": "Deny",  
         "Action": "ec2:RunInstances",  
         "Resource": [  
             "arn:aws:ec2:region:account:instance/*"  
         ],  
         "Condition": {  
             "StringNotEquals": {  
                 "ec2:InstanceType": ["t2.micro", "t2.small"]  
             }  
         }  
    },  
    {  
        "Effect": "Allow",  
        "Action": "ec2:RunInstances",  
        "Resource": [  
            "arn:aws:ec2:region::image/ami-*",  
            "arn:aws:ec2:region:account:network-interface/*",  
            "arn:aws:ec2:region:account:instance/*",  
            "arn:aws:ec2:region:account:subnet/*",  
            "arn:aws:ec2:region:account:volume/*",  
            "arn:aws:ec2:region:account:key-pair/*",  
            "arn:aws:ec2:region:account:security-group/*"  
        ]  
    }  
]
```

서브넷

다음 정책은 사용자가 지정된 서브넷(subnet-12345678)만 사용하여 인스턴스를 시작하도록 허용합니다. 다른 명령문에서 해당 권한을 부여하지 않는 경우 그룹에서 다른 서브넷으로 인스턴스를 시작할 수 없습니다. EC2-Classic으로 인스턴스를 시작할 수는 있습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Effect": "Allow",  
         "Action": "ec2:RunInstances",  
         "Resource": [  
             "arn:aws:ec2:region:account:subnet/subnet-12345678",  
             "arn:aws:ec2:region:account:network-interface/*",  
             "arn:aws:ec2:region:account:instance/*",  
             "arn:aws:ec2:region:account:volume/*",  
             "arn:aws:ec2:region::image/ami-*",  
             "arn:aws:ec2:region:account:key-pair/*",  
             "arn:aws:ec2:region:account:security-group/*"  
         ]  
    }  
]
```

또는 다른 서브넷으로 인스턴스를 시작할 권한을 거부하는 정책을 생성할 수 있습니다. 명령문에서 subnet-12345678 서브넷이 지정된 경우를 제외하고 네트워크 인터페이스를 생성할 권한을 거부하면 됩니다. 이러한 거부는 다른 서브넷으로 인스턴스를 시작하도록 허용할 목적으로 생성된 다른 정책을 모두 무시합니다. EC2-Classic으로 인스턴스를 시작할 수는 있습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Effect": "Deny",  
         "Action": "ec2:RunInstances",  
         "Resource": [  
             "arn:aws:ec2:region:account:subnet/*"  
         ],  
         "Condition": {  
             "StringNotEquals": {  
                 "ec2:SubnetId": "subnet-12345678"  
             }  
         }  
    }  
]
```

```
"Statement": [
    "Effect": "Deny",
    "Action": "ec2:RunInstances",
    "Resource": [
        "arn:aws:ec2:region:account:network-interface/*"
    ],
    "Condition": {
        "ArnNotEquals": {
            "ec2:Subnet": "arn:aws:ec2:region:account:subnet/subnet-12345678"
        }
    }
},
{
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
        "arn:aws:ec2:region::image/ami-*",
        "arn:aws:ec2:region:account:network-interface/*",
        "arn:aws:ec2:region:account:instance/*",
        "arn:aws:ec2:region:account:subnet/*",
        "arn:aws:ec2:region:account:volume/*",
        "arn:aws:ec2:region:account:key-pair/*",
        "arn:aws:ec2:region:account:security-group/*"
    ]
}
]
```

EBS 볼륨

다음 정책은 인스턴스의 EBS 볼륨이 암호화된 경우에만 사용자가 인스턴스를 시작하는 것을 허용합니다. 사용자는 암호화된 스냅샷을 사용하여 생성된 AMI에서 인스턴스를 시작하여 루트 볼륨이 암호화되도록 해야 합니다. 시작 도중 사용자가 인스턴스에 연결하는 추가적 볼륨도 암호화되어야 합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:***:volume/*"
            ],
            "Condition": {
                "Bool": {
                    "ec2:Encrypted": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:***:image/ami-*",
                "arn:aws:ec2:***:network-interface/*",
                "arn:aws:ec2:***:instance/*",
                "arn:aws:ec2:***:subnet/*",
                "arn:aws:ec2:***:key-pair/*",
                "arn:aws:ec2:***:security-group/*"
            ]
        }
    ]
}
```

태그 적용

다음 정책은 사용자가 인스턴스를 시작하고 생성 중에 인스턴스에 태그를 지정하는 것을 허용합니다. 태그를 적용하는 리소스 생성 작업의 경우, 사용자에게 `CreateTags` 작업을 사용할 권한이 있어야 합니다. 두 번째 문은 `ec2:CreateAction` 조건 키를 사용하여 사용자가 `RunInstances`의 컨텍스트에 한해 인스턴스의 태그만을 생성하는 것을 허용합니다. 사용자는 기존의 리소스에 태그를 지정할 수 없으며, `RunInstances` 요청을 사용하여 볼륨에 태그를 지정할 수 없습니다.

자세한 내용은 [태그 지정을 위한 리소스 수준 권한 \(p. 429\)](#) 단원을 참조하십시오.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:RunInstances"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:CreateTags"  
            ],  
            "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*",  
            "Condition": {  
                "StringEquals": {  
                    "ec2:CreateAction" : "RunInstances"  
                }  
            }  
        }  
    ]  
}
```

다음 정책에는 태그 `environment=production` 및 `purpose=webserver`를 사용하여 `RunInstances`에 의해 생성되는 인스턴스와 볼륨에 사용자가 태그를 지정해야 하는 `aws:RequestTag` 조건 키가 포함됩니다. `aws:TagKeys` 조건 키는 `ForAllValues` 변경자를 사용하여 요청에서 키 `environment` 및 `purpose`만 허용됨을 표시합니다(다른 어떤 태그도 지정할 수 없습니다). 요청에서 태그가 지정되지 않으면 요청이 실패합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:RunInstances"  
            ],  
            "Resource": [  
                "arn:aws:ec2:region::image/*",  
                "arn:aws:ec2:region:account:subnet/*",  
                "arn:aws:ec2:region:account:network-interface/*",  
                "arn:aws:ec2:region:account:security-group/*",  
                "arn:aws:ec2:region:account:key-pair/*"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:RunInstances"  
            ],  
            "Resource": [  
                "arn:aws:ec2:region:account:volume/*",  
                "arn:aws:ec2:region:account:volume/*"  
            ]  
        }  
    ]  
}
```

```
        "arn:aws:ec2:region:account:instance/*"
    ],
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/environment": "production" ,
            "aws:RequestTag/purpose": "webserver"
        },
        "ForAllValues:StringEquals": {
            "aws:TagKeys": ["environment","purpose"]
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:CreateTags"
    ],
    "Resource": "arn:aws:ec2:region:account:/*/*",
    "Condition": {
        "StringEquals": {
            "ec2>CreateAction" : "RunInstances"
        }
    }
}
]
}
```

다음 정책은 aws:TagKeys 조건에서 ForAnyValue 변수자를 사용하여 요청에서 적어도 하나의 태그가 지정되어야 하고 태그에 키 environment 또는 webserver가 포함되어야 함을 표시합니다. 태그는 인스턴스와 볼륨에 모두 적용되어야 합니다. 요청에서 어떤 태그 값도 지정할 수 있습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:region::image/*",
                "arn:aws:ec2:region:account:subnet/*",
                "arn:aws:ec2:region:account:network-interface/*",
                "arn:aws:ec2:region:account:security-group/*",
                "arn:aws:ec2:region:account:key-pair/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:region:account:volume/*",
                "arn:aws:ec2:region:account:instance/*"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:TagKeys": ["environment","webserver"]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:region:account:/*/*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:TagKeys": ["environment","webserver"]
                }
            }
        }
    ]
}
```

```
"Action": [
    "ec2:CreateTags"
],
"Resource": "arn:aws:ec2:region:account:/*/*",
"Condition": {
    "StringEquals": {
        "ec2:CreateAction" : "RunInstances"
    }
}
}
```

다음 정책에서는 요청에서 태그를 지정할 필요가 없지만 지정하는 경우, 태그는 `purpose=test`여야 합니다. 다른 어떤 태그도 허용되지 않습니다. 사용자는 `RunInstances` 요청에서 태그 지정 가능한 어떤 리소스에도 태그를 적용할 수 있습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:region:account:/*/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/purpose": "test",
                    "ec2:CreateAction" : "RunInstances"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "purpose"
                }
            }
        }
    ]
}
```

6. ClassicLink 작업

VPC에서 ClassicLink를 활성화한 후 VPC에 EC2-Classic 인스턴스를 링크할 수 있습니다. 또한 ClassicLink 가능 VPC 및 VPC에 링크된 모든 EC2-Classic 인스턴스를 확인할 수 있습니다. `ec2:EnableVpcClassicLink`, `ec2:DisableVpcClassicLink`, `ec2:AttachClassicLinkVpc` 및 `ec2:DetachClassicLinkVpc` 작업에 대한 리소스 수준 권한을 포함하는 정책을 생성하여 사용자가 해당 작업을 사용할 수 있는지 여부를 제어할 수 있습니다. `ec2:Describe*` 작업에는 리소스 수준 권한이 지원되지 않습니다.

항목

- [ClassicLink 작업 관련 전체 권한 \(p. 444\)](#)
- [VPC에서 ClassicLink 활성화 및 비활성화 \(p. 444\)](#)
- [인스턴스 링크 \(p. 444\)](#)
- [인스턴스 링크 해제 \(p. 445\)](#)

ClassicLink 작업 관련 전체 권한

다음 정책은 ClassicLink 가능 VPC 및 링크된 EC2-Classic 인스턴스를 확인하고, VPC에서 ClassicLink를 활성화 및 비활성화하고, ClassicLink 가능 VPC에서 인스턴스를 링크 및 링크 해제할 권한을 부여합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:DescribeClassicLinkInstances", "ec2:DescribeVpcClassicLink",  
                "ec2:EnableVpcClassicLink", "ec2:DisableVpcClassicLink",  
                "ec2:AttachClassicLinkVpc", "ec2:DetachClassicLinkVpc"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

VPC에서 ClassicLink 활성화 및 비활성화

다음 정책은 'purpose=classiclink' 태그가 있는 VPC에서 ClassicLink를 활성화 및 비활성화하도록 허용합니다. 다른 VPC에서는 ClassicLink를 활성화하거나 비활성화할 수 없습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:*VpcClassicLink",  
            "Resource": "arn:aws:ec2:region:account:vpc/*",  
            "Condition": {  
                "StringEquals": {  
                    "ec2:ResourceTag/purpose": "classiclink"  
                }  
            }  
        }  
    ]  
}
```

인스턴스 링크

다음 정책은 인스턴스가 m3.large 유형일 때만 VPC에 링크할 권한을 부여합니다. 두 번째 명령문에서는 인스턴스를 VPC에 링크하는 데 필요한 VPC 및 보안 그룹 리소스 사용을 허용합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:AttachClassicLinkVpc",  
            "Resource": "arn:aws:ec2:region:account:instance/*",  
            "Condition": {  
                "StringEquals": {  
                    "ec2:InstanceType": "m3.large"  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": "ec2:AttachClassicLinkVpc",  
            "Resource": [  
                "arn:aws:ec2:region:account:vpc/*",  
                "arn:aws:ec2:region:account:securitygroup/*"  
            ]  
        }  
    ]  
}
```

```
        "arn:aws:ec2:region:account:security-group/*"
    ]
}
]
```

다음 정책은 인스턴스를 특정 VPC(vpc-1a2b3c4d)에만 링크하고 VPC의 특정 보안 그룹(sg-1122aabb 및 sg-aabb2233)만 인스턴스에 연결할 권한을 부여합니다. 사용자는 다른 VPC에는 인스턴스를 링크할 수 없고, 요청 시 인스턴스에 연결할 다른 VPC 보안 그룹을 지정할 수 없습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:AttachClassicLinkVpc",
      "Resource": [
        "arn:aws:ec2:region:account:vpc/vpc-1a2b3c4d",
        "arn:aws:ec2:region:account:instance/*",
        "arn:aws:ec2:region:account:security-group/sg-1122aabb",
        "arn:aws:ec2:region:account:security-group/sg-aabb2233"
      ]
    }
  ]
}
```

인스턴스 링크 해제

다음 정책은 인스턴스에 "unlink=true" 태그가 있을 때만 VPC에서 링크된 EC2-Classic 인스턴스의 링크를 해제할 권한을 부여합니다. 두 번째 명령문에서는 VPC에서 인스턴스의 링크를 해제하는 데 필요한 VPC 리소스를 사용할 권한을 부여합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:DetachClassicLinkVpc",
      "Resource": [
        "arn:aws:ec2:region:account:instance/*"
      ],
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/unlink": "true"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:DetachClassicLinkVpc",
      "Resource": [
        "arn:aws:ec2:region:account:vpc/*"
      ]
    }
  ]
}
```

7. 예약 인스턴스 사용

다음 정책에서는 계정의 예약 인스턴스에 대한 보기, 수정, 구매 권한을 사용자에게 부여합니다.

개별 예약 인스턴스에 대해서는 리소스 수준 권한을 설정할 수 없습니다. 이 정책은 사용자들이 계정의 모든 예약 인스턴스에 액세스할 수 있음을 뜻합니다.

Resource 요소에 사용되는 * 와일드카드는 사용자가 그 작업을 통해 모든 리소스를 지정할 수 있음을 나타냅니다. 이 경우 사용자는 계정의 모든 예약 인스턴스를 나열하고 수정할 수 있습니다. 계정 자격 증명을 사용해 예약 인스턴스를 구매할 수도 있습니다. API 작업이 리소스 수준 권한을 지원하지 않는 경우에도 * 와일드카드가 필요합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:DescribeReservedInstances", "ec2:ModifyReservedInstances",  
                "ec2:PurchaseReservedInstancesOffering", "ec2:DescribeAvailabilityZones",  
                "ec2:DescribeReservedInstancesOfferings"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

사용자에게 계정의 예약 인스턴스를 보고 수정할 수 있도록 허용하되 새 예약 인스턴스를 구매할 수는 없도록 하려면

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:DescribeReservedInstances", "ec2:ModifyReservedInstances",  
                "ec2:DescribeAvailabilityZones"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

8. 리소스에 태그 지정

다음 정책은 태그에 키 environment 및 값 production이 포함된 경우에만 사용자가 createTags 작업을 사용하여 인스턴스에 태그를 적용하는 것을 허용합니다. ForAllValues 변경자는 aws:TagKeys 조건 키와 함께 사용되어 요청에서 키 environment만 허용됨을 표시합니다(다른 어떤 태그도 허용되지 않습니다). 사용자는 다른 어떤 리소스 유형에도 태그를 지정할 수 없습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:CreateTags"  
            ],  
            "Resource": "arn:aws:ec2:region:account:instance/*",  
            "Condition": {  
                "StringEquals": {  
                    "ec2:RequestTag/environment": "production"  
                },  
                "ForAllValues:StringEquals": {  
                    "aws:TagKeys": [  
                        "environment"  
                    ]  
                }  
            }  
        }  
    ]  
}
```

```
        }
    }
}
```

다음 정책은 키가 `owner`이고 값이 IAM `username`인 태그를 이미 가진 태그 지정 가능한 리소스에 태그를 지정하는 것을 허용합니다. 또한 사용자는 요청에서 키가 `environment`이고 값이 `test` 또는 `prod`인 태그를 지정해야 합니다. 사용자는 요청에서 추가 태그를 지정할 수 있습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:region:account:/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/environment": ["test", "prod"],
          "ec2:ResourceTag/owner": "${aws:username}"
        }
      }
    }
  ]
}
```

사용자가 리소스의 특정 태그를 삭제하는 것을 허용하는 IAM 정책을 만들 수 있습니다. 예를 들어 요청에서 지정된 태그 키가 `environment` 또는 `cost-center`인 경우, 다음 정책은 사용자가 볼륨의 태그를 삭제하는 것을 허용합니다. 태그에는 어떤 값도 지정할 수 있지만 태그 키는 지정된 키 중 하나와 일치해야 합니다.

Note

리소스를 삭제하면 리소스에 지정되어 있는 모든 태그도 함께 삭제됩니다. 사용자는 태그가 지정된 리소스를 삭제할 때 `ec2:DeleteTags` 작업 권한이 필요하지 않습니다. 삭제 작업을 위한 권한만 있으면 됩니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:DeleteTags",
      "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",
      "Condition": {
        "ForAllValues:StringEquals": {
          "aws:TagKeys": ["environment", "cost-center"]
        }
      }
    }
  ]
}
```

이 정책은 키가 `owner`이고 값이 IAM `username`인 키로 리소스에 태그가 지정된 경우에 한해 어떤 리소스에 서든 `environment=prod` 태그만을 삭제하는 것을 허용합니다. 사용자는 리소스의 다른 어떤 태그도 삭제할 수 없습니다.

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

```

```
"Action": [
    "ec2:DeleteTags"
],
"Resource": "arn:aws:ec2:region:account:/*",
"Condition": {
    "StringEquals": {
        "aws:RequestTag/environment": "prod",
        "ec2:ResourceTag/owner": "${aws:username}"
    },
    "ForAllValues:StringEquals": {
        "aws:TagKeys": ["environment"]
    }
}
]
```

9: IAM 역할 작업

다음 정책을 통해 사용자는 department=test 태그가 있는 인스턴스에 IAM 역할을 연결, 교체 및 분리할 수 있습니다. IAM 역할을 교체하거나 분리하려면 연결 ID가 필요하기 때문에 정책은 사용자에게 ec2:DescribeIamInstanceProfileAssociations 작업을 사용할 수 있는 권한도 부여합니다.

IAM 사용자가 인스턴스에 역할을 전달하기 위해서는 iam:PassRole 작업을 사용할 수 있는 권한이 있어야 합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AssociateIamInstanceProfile",
                "ec2:ReplaceIamInstanceProfileAssociation",
                "ec2:DisassociateIamInstanceProfile"
            ],
            "Resource": "arn:aws:ec2:region:account:instance/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/department": "test"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeIamInstanceProfileAssociations",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "*"
        }
    ]
}
```

다음 정책을 통해 사용자는 모든 인스턴스에 IAM 역할을 연결하거나 교체할 수 있습니다. 사용자는 이름이 TestRole-로 시작하는 IAM 역할만 연결하거나 교체할 수 있습니다. iam:PassRole 작업의 경우, 인스턴스 프로파일이 아닌 IAM 역할의 이름을 지정하십시오(이름이 다른 경우). 자세한 내용은 [인스턴스 프로파일 \(p. 457\)](#) 섹션을 참조하십시오.

```
{
```

```
"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "ec2:AssociateIamInstanceProfile",
            "ec2:ReplaceIamInstanceProfileAssociation"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": "ec2:DescribeIamInstanceProfileAssociations",
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": "iam:PassRole",
        "Resource": "arn:aws:iam::account:role/TestRole-*"
    }
]
```

Amazon EC2 콘솔 작업을 위한 예제 정책

Amazon EC2 콘솔에서 IAM 정책을 사용하여 특정 리소스를 조회하고 관련 작업을 수행할 권한을 부여할 수 있습니다. 이전 섹션의 예제 정책을 사용할 수 있지만 해당 정책은 AWS CLI 또는 AWS SDK를 통한 요청에 맞게 설계되었습니다. 콘솔에서는 추가적인 API 작업을 통해 해당 기능을 구현하므로 이러한 정책이 예상과 다르게 작동할 수 있습니다. 예를 들어 `DescribeVolumes` API 작업만 사용할 권한을 갖는 경우 콘솔에서 볼륨을 조회하려고 하면 오류가 발생합니다. 이 섹션에서는 콘솔의 특정 부분을 사용하도록 허용하는 정책을 보여 줍니다.

항목

- 1: 읽기 전용 액세스 (p. 449)
- 2: EC2 시작 마법사 사용 (p. 450)
- 3: 볼륨 작업 (p. 452)
- 4: 보안 그룹 작업 (p. 453)
- 5: 탄력적 IP 주소 작업 (p. 455)
- 6: 예약 인스턴스 사용 (p. 456)

Note

콘솔에서 작업을 수행하는 데 필요한 API 작업을 파악하려는 경우 AWS CloudTrail 등의 서비스를 사용할 수 있습니다. 자세한 내용은 [AWS CloudTrail User Guide](#)을 참조하십시오. 정책에서 특정 리소스를 생성하거나 수정할 권한을 부여하지 않는 경우 콘솔에 진단 정보가 포함된 인코딩 메시지가 표시됩니다. AWS STS의 `DecodeAuthorizationMessage` API 작업이나 AWS CLI의 `decode-authorization-message` 명령을 사용하여 메시지를 디코딩할 수 있습니다.

Amazon EC2 콘솔용 정책을 생성하는 방법에 대한 자세한 내용은 AWS 보안 블로그 게시물 [Granting Users Permission to Work in the Amazon EC2 Console](#)을 참조하십시오.

1: 읽기 전용 액세스

사용자가 Amazon EC2 콘솔에서 모든 리소스를 조회하도록 허용하려면 다음 예제와 같은 정책을 사용합니다. 1: 읽기 전용 액세스 (p. 431). 다른 명령문에서 해당 권한을 부여하지 않는 경우 이러한 리소스에 대해 작업을 수행하거나 새 리소스를 생성할 수는 없습니다.

- a. 인스턴스, AMI 및 스냅샷 조회

리소스 중 일부에 대한 읽기 전용 액세스를 제공할 수도 있습니다. 이렇게 하려면 `ec2:Describe` API 작업에서 * 와일드카드를 구체적인 리소스별 `ec2:Describe` 작업으로 대체합니다. 다음 정책은 사용자가 Amazon EC2 콘솔에서 모든 인스턴스, AMI 및 스냅샷을 조회하도록 허용합니다. `ec2:DescribeTags` 작업에서는 사용자가 퍼블릭 AMI를 조회할 수 있습니다. 콘솔에 퍼블릭 AMI를 표시하려면 태그 지정 정보가 필요하지만 사용자가 프라이빗 AMI만 조회하도록 하려면 이 작업을 제거할 수도 있습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [{  
        "Effect": "Allow",  
        "Action": [  
            "ec2:DescribeInstances", "ec2:DescribeImages",  
            "ec2:DescribeTags", "ec2:DescribeSnapshots"  
        ],  
        "Resource": "*"  
    }  
}
```

Note

현재 Amazon EC2 `ec2:Describe*` API 작업은 리소스별 권한을 지원하지 않으므로 사용자가 콘솔에서 조회할 수 있는 리소스를 개별적으로 제어할 수는 없습니다. 따라서 위 명령문의 `Resource` 요소에 * 와일드카드가 필요합니다. 어떠한 Amazon EC2 API 작업에 어떠한 ARN을 사용할 수 있는지에 대한 자세한 내용은 [Amazon EC2 API 작업에 지원되는 리소스 수준 권한 \(p. 409\)](#) 섹션을 참조하십시오.

b. 인스턴스 및 CloudWatch 측정치 조회

다음 정책은 사용자로 하여금 [Instances] 페이지의 [Monitoring] 탭에 있는 CloudWatch 경보 및 측정치 뿐만 아니라 Amazon EC2 콘솔의 인스턴스까지도 조회할 수 있도록 허용합니다. Amazon EC2 콘솔은 CloudWatch API를 이용해 경보 및 측정치를 표시하므로, 반드시 사용자에게 `cloudwatch:DescribeAlarms` 및 `cloudwatch:GetMetricStatistics` 작업을 사용할 수 있는 권한을 부여해야 합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [{  
        "Effect": "Allow",  
        "Action": [  
            "ec2:DescribeInstances",  
            "cloudwatch:DescribeAlarms",  
            "cloudwatch:GetMetricStatistics"  
        ],  
        "Resource": "*"  
    }  
}
```

2: EC2 시작 마법사 사용

Amazon EC2 시작 마법사는 인스턴스 구성 및 시작 옵션을 포함하는 일련의 화면으로 구성됩니다. 사용자가 마법사의 옵션을 사용할 수 있도록 정책에 API 작업 사용 권한이 포함되어야 합니다. 해당 작업 사용 권한이 정책에 포함되지 않으면 마법사의 일부 항목이 제대로 로드되지 않고 사용자가 시작을 완료할 수 없습니다.

a. 기본 시작 마법사 액세스

성공적으로 시작을 완료하려면 사용자에게 `ec2:RunInstances` API 작업 및 최소한 다음과 같은 API 작업 사용 권한을 부여해야 합니다.

- `ec2:DescribeImages`: AMI를 조회하고 선택합니다.

- `ec2:DescribeVPCs`: 사용 가능한 네트워크 옵션(EC2-Classic 및 VPC 목록)을 조회합니다. VPC로 시작하는 경우에도 필수 항목입니다.
- `ec2:DescribeSubnets`: VPC로 시작하는 경우 해당 VPC에서 사용 가능한 모든 서브넷을 조회합니다.
- `ec2:DescribeSecurityGroups`: 마법사에서 보안 그룹 페이지를 조회합니다. 사용자는 기존 보안 그룹을 선택할 수 있습니다.
- `ec2:DescribeKeyPairs` 또는 `ec2:CreateKeyPair`: 기존 키 페어를 선택하거나 새로 생성합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:DescribeInstances", "ec2:DescribeImages",  
                "ec2:DescribeKeyPairs", "ec2:DescribeVpcs", "ec2:DescribeSubnets",  
                "ec2:DescribeSecurityGroups"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": "ec2:RunInstances",  
            "Resource": "*"  
        }  
    ]  
}
```

정책에 API 작업을 추가하여 다음과 같이 사용자에게 더 많은 옵션을 제공할 수 있습니다.

- `ec2:DescribeAvailabilityZones`: EC2-Classic으로 시작하는 경우 특정 가용 영역을 조회하고 선택합니다.
- `ec2:DescribeNetworkInterfaces`: VPC로 시작하는 경우 선택한 서브넷의 기존 네트워크 인터페이스를 조회하고 선택합니다.
- `ec2:CreateSecurityGroup`: 새 보안 그룹을 생성합니다. 예를 들어 마법사가 제안하는 `launch-wizard-x` 보안 그룹을 생성합니다. 그러나 이 작업은 보안 그룹을 생성하기만 하며 규칙을 추가하거나 수정하지 않습니다. 인바운드 규칙을 추가하려면 `ec2:AuthorizeSecurityGroupIngress` API 작업 사용 권한이 부여되어야 합니다. VPC 보안 그룹에 아웃바운드 규칙을 추가하려면 `ec2:AuthorizeSecurityGroupEgress` API 작업 사용 권한이 부여되어야 합니다. 기존 규칙을 수정 또는 삭제하려면 관련 `ec2:RevokeSecurityGroup*` API 작업 사용 권한이 부여되어야 합니다.
- `ec2:CreateTags`: `RunInstances`에 의해 생성된 리소스에 태그 지정. 자세한 내용은 [태그 지정을 위한 리소스 수준 권한 \(p. 429\)](#) 섹션을 참조하십시오. 이 작업을 사용할 권한이 없는 사용자가 시작 마법사의 태그 지정 페이지에서 태그를 지정하려 시도하는 경우, 시작은 실패합니다.

Important

`ec2:CreateTags` 작업 사용 권한을 부여할 때는 신중해야 합니다. 이렇게 하면 사용자가 리소스의 태그를 임의로 변경하여 제한 조건을 무력화할 수 있으므로 `ec2:ResourceTag` 조건 키로 다른 리소스의 사용을 제한하지 못하게 됩니다.

현재 Amazon EC2 `Describe*` API 작업은 리소스별 권한을 지원하지 않으므로 사용자가 시작 마법사에서 조회할 수 있는 리소스를 개별적으로 제한할 수는 없습니다. 그러나 `ec2:RunInstances` API 작업에 리소스별 권한을 적용하여 사용자가 인스턴스를 시작하는 데 사용 가능한 리소스를 제한할 수 있습니다. 사용자가 사용 권한이 없는 옵션을 선택하면 시작에 실패합니다.

b. 특정 인스턴스 유형, 서브넷, 리전에 대한 액세스 제한

다음 정책은 Amazon이 소유한 AMI를 사용하여 `m1.small` 인스턴스를 시작하되 특정 서브넷 (`subnet-1a2b3c4d`)으로만 시작하도록 허용합니다. 사용자는 `sa-east-1` 리전에서만 시작할 수 있습니다. 사

용자가 다른 리전을 선택하거나 시작 마법사에서 다른 인스턴스 유형, AMI 또는 서브넷을 선택하면 시작에 실패합니다.

첫 번째 명령문은 위 예제와 같이 사용자가 시작 마법사에서 옵션을 조회할 권한을 부여합니다. 두 번째 명령문은 `ec2:RunInstances` 작업에서 네트워크 인터페이스, 볼륨, 키 페어, 보안 그룹 및 서브넷 리소스를 사용할 권한을 부여합니다. 이 권한은 인스턴스를 VPC로 시작하는 데 필요합니다. `ec2:RunInstances` 작업 사용에 대한 자세한 내용은 [5: 인스턴스 시작\(RunInstances\) \(p. 436\)](#) 섹션을 참조하십시오. 세 번째, 네 번째 명령문은 각각 인스턴스 및 AMI 리소스 사용 권한을 부여하지만 인스턴스가 `m1.small` 인스턴스이고 AMI를 Amazon이 소유한 경우로 한정합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:DescribeInstances", "ec2:DescribeImages",  
                "ec2:DescribeKeyPairs", "ec2:DescribeVpcs", "ec2:DescribeSubnets",  
                "ec2:DescribeSecurityGroups"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": "ec2:RunInstances",  
            "Resource": [  
                "arn:aws:ec2:sa-east-1:111122223333:network-interface/*",  
                "arn:aws:ec2:sa-east-1:111122223333:volume/*",  
                "arn:aws:ec2:sa-east-1:111122223333:key-pair/*",  
                "arn:aws:ec2:sa-east-1:111122223333:security-group/*",  
                "arn:aws:ec2:sa-east-1:111122223333:subnet/subnet-1a2b3c4d"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": "ec2:RunInstances",  
            "Resource": [  
                "arn:aws:ec2:sa-east-1:111122223333:instance/*"  
            ],  
            "Condition": {  
                "StringEquals": {  
                    "ec2:InstanceType": "m1.small"  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": "ec2:RunInstances",  
            "Resource": [  
                "arn:aws:ec2:sa-east-1::image/ami-*"  
            ],  
            "Condition": {  
                "StringEquals": {  
                    "ec2:Owner": "amazon"  
                }  
            }  
        }  
    ]  
}
```

3: 볼륨 작업

다음 정책은 볼륨을 조회 및 생성하고 특정 인스턴스에 볼륨을 연결 및 분리할 권한을 부여합니다.

사용자는 "purpose=test" 태그가 있는 인스턴스에 볼륨을 연결하고 해당 인스턴스에서 볼륨을 분리할 수 있습니다. Amazon EC2 콘솔을 사용하여 볼륨을 연결하려는 경우 사용자에게 ec2:DescribeInstances 작업 사용 권한을 부여하는 것이 좋습니다. 이렇게 하면 [Attach Volume] 대화 상자의 미리 구성된 목록에서 인스턴스를 선택할 수 있습니다. 그러나 이렇게 하면 사용자가 콘솔의 [Instances] 페이지에서 모든 인스턴스를 조회할 수 있으므로 이 작업을 생략할 수 있습니다.

첫 번째 명령문에서 ec2:DescribeVolumeStatus 및 ec2:DescribeAvailabilityZones 작업은 볼륨을 콘솔에 올바르게 표시하는데 필요합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:DescribeVolumes", "ec2:DescribeVolumeStatus",  
                "ec2:DescribeAvailabilityZones", "ec2>CreateVolume",  
                "ec2:DescribeInstances"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:AttachVolume",  
                "ec2:DetachVolume"  
            ],  
            "Resource": "arn:aws:ec2:region:111122223333:instance/*",  
            "Condition": {  
                "StringEquals": {  
                    "ec2:ResourceTag/purpose": "test"  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:AttachVolume",  
                "ec2:DetachVolume"  
            ],  
            "Resource": "arn:aws:ec2:region:111122223333:volume/*"  
        }  
    ]  
}
```

4: 보안 그룹 작업

a. 보안 그룹 조회와 규칙의 추가 및 삭제

다음 정책은 사용자가 Amazon EC2 콘솔에서 보안 그룹을 조회하고 Department=Test 태그가 있는 기존 보안 그룹에서 인바운드 및 아웃바운드 규칙을 추가 및 제거할 권한을 부여합니다.

Note

EC2-Classic 보안 그룹의 아웃바운드 규칙은 수정할 수 없습니다. 보안 그룹에 대한 자세한 내용은 [Linux 인스턴스에 대한 Amazon EC2 보안 그룹 \(p. 385\)](#)을 참조하십시오.

첫 번째 명령문에서 ec2:DescribeTags 작업은 사용자가 콘솔에서 태그를 조회하도록 허용하므로 사용자가 수정 가능한 보안 그룹을 쉽게 식별할 수 있습니다.

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

```
"Effect": "Allow",
"Action": [
    "ec2:DescribeSecurityGroups", "ec2:DescribeTags"
],
"Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:AuthorizeSecurityGroupIngress", "ec2:RevokeSecurityGroupIngress",
        "ec2:AuthorizeSecurityGroupEgress", "ec2:RevokeSecurityGroupEgress"
    ],
    "Resource": [
        "arn:aws:ec2:region:111122223333:security-group/*"
    ],
    "Condition": {
        "StringEquals": {
            "ec2:ResourceTag/Department": "Test"
        }
    }
}
]
```

b. “Create Security Group” 대화 상자에서 작업하기

사용자가 Amazon EC2 콘솔에서 [Create Security Group] 대화 상자를 사용하도록 허용하는 정책을 생성할 수 있습니다. 이 대화 상자를 사용하려면 최소한 다음과 같은 API 작업 사용 권한을 부여해야 합니다.

- `ec2:CreateSecurityGroup`: 새 보안 그룹을 생성합니다.
- `ec2:DescribeVpcs`: [VPC] 목록에서 기존 VPC의 목록을 조회합니다. 이 작업은 EC2-Classic에서 보안 그룹을 만드는 데 필요하지 않습니다.

이 권한이 있으면 사용자가 새 보안 그룹을 생성할 수 있지만 규칙을 추가할 수는 없습니다. [Create Security Group] 대화 상자에서 규칙 관련 작업을 수행하려면 정책에 다음 API 작업을 추가합니다.

- `ec2:AuthorizeSecurityGroupIngress`: 인바운드 규칙을 추가합니다.
- `ec2:AuthorizeSecurityGroupEgress`: VPC 보안 그룹에 아웃바운드 규칙을 추가합니다.
- `ec2:RevokeSecurityGroupIngress`: 기존 인바운드 규칙을 수정하거나 삭제합니다. 이 권한은 사용자가 콘솔에서 [Copy to new] 기능을 사용하도록 허용하려는 경우에 유용합니다. 이 기능은 [Create Security Group] 대화 상자를 열고 선택한 보안 그룹과 같은 규칙을 미리 입력합니다.
- `ec2:RevokeSecurityGroupEgress`: VPC 보안 그룹의 아웃바운드 규칙을 수정하거나 삭제합니다. 이 권한은 모든 아웃바운드 트래픽을 허용하는 기본 아웃바운드 규칙을 사용자가 수정 또는 삭제하도록 허용하는 데 유용합니다.
- `ec2>DeleteSecurityGroup`: 잘못된 규칙을 저장할 수 없도록 합니다. 콘솔에서 먼저 보안 그룹을 만든 후 지정된 규칙을 추가합니다. 규칙이 잘못된 경우 작업이 실패하고 콘솔이 보안 그룹을 삭제하려고 합니다. 사용자는 [Create Security Group] 대화 상자에 남아 있기 때문에 잘못된 규칙을 수정한 후 보안 그룹을 다시 생성해 볼 수 있습니다. 이 API 작업은 필수적이지는 않지만 해당 사용 권한을 부여하지 않으면 사용자가 잘못된 규칙이 포함된 보안 그룹을 생성하려고 할 때 규칙이 없는 보안 그룹이 생성되며, 사용자가 이후에 규칙을 추가해야 합니다.

현재 `ec2:CreateSecurityGroup` API 작업은 리소스 수준 권한을 지원하지 않지만 `ec2:AuthorizeSecurityGroupIngress` 및 `ec2:AuthorizeSecurityGroupEgress` 작업에 리소스 수준 권한을 적용하여 사용자의 규칙 생성 방법을 제어할 수 있습니다.

다음 정책은 [Create Security Group] 대화 상자를 사용하고 특정 VPC(`vpc-1a2b3c4d`)에 연결된 보안 그룹에 인바운드 및 아웃바운드 규칙을 생성할 권한을 부여합니다. 사용자는 EC2-Classic 또는 다른 VPC의 보안 그룹을 생성할 수 있지만 규칙을 추가할 수는 없습니다. 마찬가지로 VPC `vpc-1a2b3c4d`에 연결되지 않은 기존

보안 그룹에는 규칙을 추가할 수는 없습니다. 또한 콘솔에서 모든 보안 그룹을 조회할 권한이 부여됩니다. 따라서 사용자가 인바운드 규칙을 추가할 수 있는 보안 그룹을 쉽게 식별할 수 있습니다. 또한 이 정책은 VPC vpc-1a2b3c4d에 연결된 보안 그룹을 삭제할 권한을 부여합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [{  
        "Effect": "Allow",  
        "Action": [  
            "ec2:DescribeSecurityGroups", "ec2:CreateSecurityGroup", "ec2:DescribeVpcs"  
        ],  
        "Resource": "*"  
    },  
    {  
        "Effect": "Allow",  
        "Action": [  
            "ec2:DeleteSecurityGroup", "ec2:AuthorizeSecurityGroupIngress",  
            "ec2:AuthorizeSecurityGroupEgress"  
        ],  
        "Resource": "arn:aws:ec2:region:111122223333:security-group/*",  
        "Condition": {  
            "ArnEquals": {  
                "ec2:Vpc": "arn:aws:ec2:region:111122223333:vpc/vpc-1a2b3c4d"  
            }  
        }  
    }  
]
```

5: 탄력적 IP 주소 작업

사용자가 Amazon EC2 콘솔에서 탄력적 IP 주소를 볼 수 있도록 하려면 사용자에게 ec2:DescribeAddresses 작업을 사용할 수 있는 권한을 부여해야 합니다.

사용자에게 탄력적 IP 주소 관련 작업을 허용하려면 정책에 다음 작업을 추가합니다.

- **ec2:AllocateAddress:** VPC 또는 EC2-Classic에서 사용할 주소를 할당합니다.
- **ec2:ReleaseAddress:** 탄력적 IP 주소를 해제합니다.
- **ec2:AssociateAddress:** 인스턴스 또는 네트워크 인터페이스에 탄력적 IP 주소를 연결합니다.
- **ec2:DescribeNetworkInterfaces** 및 **ec2:DescribeInstances:** [Associate address] 화면에서 작업합니다. 탄력적 IP 주소를 연결할 수 있는 네트워크 인터페이스나 사용 가능한 인스턴스가 화면에 표시됩니다. EC2-Classic 인스턴스의 경우, 사용자는 **ec2:DescribeInstances**를 사용할 권한만 필요합니다.
- **ec2:DisassociateAddress:** 인스턴스 또는 네트워크 인터페이스에서 탄력적 IP 주소를 분리합니다.

다음 정책을 통해 사용자는 탄력적 IP 주소를 확인하고 인스턴스에 할당, 연결할 수 있습니다. 사용자는 탄력적 IP 주소를 네트워크 인터페이스에 연결하거나 탄력적 IP 주소 연결을 끊거나 릴리스할 수 없습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:DescribeAddresses",  
                "ec2:AllocateAddress",  
                "ec2:DescribeInstances",  
                "ec2:AssociateAddress"  
            ],  
            "Resource": "*"  
        }  
    ]
```

```
        }
    }
```

6: 예약 인스턴스 사용

다음 정책을 IAM 사용자에 연결할 수 있습니다. 그렇게 하면 사용자가 계정의 예약 인스턴스를 보고 수정할 수 있을 뿐만 아니라 AWS Management Console에서 새 예약 인스턴스를 구매할 수 있는 액세스 권한을 갖게 됩니다.

이 정책은 사용자들이 계정에서 온디맨드 인스턴스뿐만 아니라 모든 예약 인스턴스를 볼 수 있도록 허용합니다. 개별 예약 인스턴스에 대해서는 리소스 수준 권한을 설정할 수 없습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeReservedInstances", "ec2:ModifyReservedInstances",
        "ec2:PurchaseReservedInstancesOffering", "ec2:DescribeInstances",
        "ec2:DescribeAvailabilityZones", "ec2:DescribeReservedInstancesOfferings"
      ],
      "Resource": "*"
    }
  ]
}
```

`ec2:DescribeAvailabilityZones` 작업은 Amazon EC2 콘솔이 예약 인스턴스를 구매할 수 있는 가용 영역에 대한 정보를 표시하도록 하는 데 필수적입니다. `ec2:DescribeInstances` 작업은 필수적이지는 않지만 사용자가 계정에서 인스턴스를 보고, 정확한 사양에 맞추기 위해 예약을 구매할 수 있도록 해줍니다.

API 작업을 조정해 사용자 액세스를 제한할 수 있습니다. 예를 들어 `ec2:DescribeInstances`와 `ec2:DescribeAvailabilityZones`를 제거하면 사용자가 읽기 전용 액세스 권한만을 갖게 됩니다.

Amazon EC2의 IAM 역할

애플리케이션은 AWS 자격 증명으로 API 요청에 서명해야 합니다. 따라서 애플리케이션 개발자는 EC2 인스턴스에서 실행되는 인스턴스의 자격 증명을 관리할 전략을 수립해야 합니다. 예를 들어 AWS 자격 증명을 인스턴스에 안전하게 배포하여 다른 사용자로부터 보호하는 한편 해당 인스턴스의 애플리케이션이 자격 증명을 사용하여 요청에 서명하도록 할 수 있습니다. 그러나 각 인스턴스에 자격 증명을 안전하게 배포하기란 쉽지 않으며, 스팟 인스턴스와 같이 AWS에서 자동으로 생성하는 인스턴스 또는 Auto Scaling 그룹의 인스턴스에 대해서는 특히 어렵습니다. 또한 AWS 자격 증명을 교체할 때 각 인스턴스의 자격 증명을 업데이트할 수 있어야 합니다.

애플리케이션이 사용하는 보안 자격 증명을 직접 관리할 필요 없이 인스턴스의 애플리케이션에서 안전하게 API 요청을 전송할 수 있도록 IAM 역할을 설계했습니다. AWS 자격 증명을 생성하고 배포하는 대신 다음과 같이 IAM 역할을 사용하여 API 요청 전송 권한을 위임할 수 있습니다.

1. IAM 역할을 생성합니다.
2. 역할을 수행할 수 있는 계정 또는 AWS 서비스를 정의합니다.
3. 역할을 수행하면서 애플리케이션이 사용할 수 있는 API 작업 및 리소스를 정의합니다.
4. 인스턴스를 시작할 때 역할을 지정하거나, 실행 중이거나 종지된 인스턴스에 역할을 연결합니다.
5. 애플리케이션에서 임시 자격 증명 세트를 검색하여 사용하도록 합니다.

예를 들어 IAM 역할을 사용하여 인스턴스에서 실행되며 Amazon S3의 버킷을 사용해야 하는 애플리케이션에 해당 권한을 부여할 수 있습니다. JSON 형식으로 정책을 생성하여 IAM 역할에 권한을 지정할 수 있습니다.

다. 이 방법은 IAM 사용자를 대상으로 정책을 생성할 때와 비슷합니다. 역할을 변경하면 모든 인스턴스에 변경 내용이 전파됩니다.

단일 인스턴스에 여러 IAM 역할을 연결할 수 없지만 여러 인스턴스에 단일 IAM 역할을 연결할 수 있습니다. IAM 역할 생성 및 사용에 대한 자세한 내용은 IAM 사용 설명서에서 [역할](#)을 참조하십시오.

IAM 정책에 리소스 수준 권한을 적용하여 사용자가 인스턴스에 IAM 역할을 연결, 교체 또는 분리할 수 있는 권한을 제어할 수 있습니다. 자세한 내용은 [Amazon EC2 API 작업에 지원되는 리소스 수준 권한 \(p. 409\)](#) 및 다음 예제: [9: IAM 역할 작업 \(p. 448\)](#)을(를) 참조하십시오.

항목

- [인스턴스 프로파일 \(p. 457\)](#)
- [인스턴스 메타데이터에서 보안 자격 증명 검색 \(p. 457\)](#)
- [IAM 사용자에게 IAM 역할을 인스턴스에 전달할 수 있는 권한 부여 \(p. 458\)](#)
- [IAM 역할 작업 \(p. 458\)](#)

인스턴스 프로파일

Amazon EC2에서는 인스턴스 프로파일을 IAM 역할의 컨테이너로 사용합니다. IAM 콘솔을 사용하여 IAM 역할을 생성하면 인스턴스 프로파일이 자동으로 생성되고 해당 역할과 동일한 이름이 지정됩니다. Amazon EC2 콘솔을 사용하여 IAM 역할로 인스턴스를 시작하거나 인스턴스에 IAM 역할을 연결하는 경우 인스턴스 프로파일 이름 목록을 기반으로 인스턴스를 선택합니다.

AWS CLI, API 또는 AWS SDK를 사용하여 역할을 생성하면 역할과 인스턴스 프로파일이 별개의 작업으로 생성되며 이름은 각각 다를 수 있습니다. AWS CLI, API 또는 AWS SDK를 사용하여 IAM 역할로 인스턴스를 시작하거나 인스턴스에 IAM 역할을 연결하는 경우 인스턴스 프로파일 이름을 지정합니다.

인스턴스 프로파일은 하나의 IAM 역할만 포함할 수 있습니다. 이 한도는 늘릴 수 없습니다.

자세한 내용은 IAM 사용 설명서에서 [인스턴스 프로파일](#)을 참조하십시오.

인스턴스 메타데이터에서 보안 자격 증명 검색

인스턴스의 애플리케이션은 인스턴스 메타데이터 항목 `iam/security-credentials/role-name`에서 역할이 제공하는 보안 자격 증명을 검색합니다. 역할에 연결된 보안 자격 증명을 통해 역할에 정의한 작업 및 리소스에 대한 권한이 애플리케이션에 부여됩니다. 이러한 보안 자격 증명은 임시로 발급되며 자동으로 교체됩니다. 이전 자격 증명이 만료되기 최소 5분 전에 새 자격 증명이 제공됩니다.

Warning

IAM 역할과 함께 인스턴스 메타데이터를 사용하는 서비스를 사용하는 경우 서비스에서 사용자 대신 HTTP 호출을 수행할 때 자격 증명이 노출되지 않도록 주의하십시오. 자격 증명이 노출될 수 있는 서비스 유형은 HTTP 프록시, HTML/CSS 검증 서비스, XML 포함을 지원하는 XML 프로세서 등입니다.

다음 명령은 `s3access`라는 IAM 역할의 보안 자격 증명을 검색합니다.

```
curl http://169.254.169.254/latest/meta-data/iam/security-credentials/s3access
```

다음은 예제 출력입니다.

```
{  
    "Code" : "Success",  
    "LastUpdated" : "2012-04-26T16:39:16Z",  
    "Type" : "AWS-HMAC",  
    "AccessKeyId" : "AKIAIOSFODNN7EXAMPLE",  
    "SecretAccessKey" : "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEkey",  
    "Token" : "FQoGKYB4CgqDfHnLWzZGvVdXyfKuXGt",  
    "Expiration" : "2012-04-26T16:40:16Z",  
    "SessionToken" : "FQoGKYB4CgqDfHnLWzZGvVdXyfKuXGt"}  
Content-Type: application/json
```

```
{  
    "SecretAccessKey" : "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",  
    "Token" : "token",  
    "Expiration" : "2012-04-27T22:39:16Z"  
}
```

인스턴스에서 실행되는 애플리케이션, AWS CLI 및 Windows PowerShell용 도구 명령의 경우, 임시 보안 자격 증명을— 명시적으로 얻지 않아도 됩니다. AWS SDKs, AWS CLI 및 Windows PowerShell용 도구이 EC2 인스턴스 메타데이터 서비스에서 자격 증명을 자동으로 얻어 그것을 사용하기 때문입니다. 임시 보안 자격 증명을 사용하여 인스턴스 외부로 호출하려면(예: IAM 정책 테스트) 액세스 키, 보안 키 및 세션 토큰을 제공해야 합니다. 자세한 내용은 IAM 사용 설명서의 [임시 보안 자격 증명을 사용해 AWS 리소스에 대한 액세스 요청하기](#)를 참조하십시오.

인스턴스 메타데이터에 대한 자세한 내용은 [인스턴스 메타데이터 및 사용자 데이터 \(p. 321\)](#) 섹션을 참조하십시오.

IAM 사용자에게 IAM 역할을 인스턴스에 전달할 수 있는 권한 부여

IAM 사용자가 IAM 역할로 인스턴스를 시작하거나 기존 인스턴스에 IAM 역할을 연결하거나 교체할 수 있도록 하려면 인스턴스에 역할을 전달할 권한을 부여해야 합니다.

다음 IAM 정책은 사용자에게 IAM 역할로 인스턴스(ec2:RunInstances)를 시작하거나 기존 인스턴스(ec2:AssociateIamInstanceProfile 및 ec2:ReplaceIamInstanceProfileAssociation)에 IAM 역할을 연결하거나 교체할 수 있는 권한을 부여합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:RunInstances",  
                "ec2:AssociateIamInstanceProfile",  
                "ec2:ReplaceIamInstanceProfileAssociation"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": "iam:PassRole",  
            "Resource": "*"  
        }  
    ]  
}
```

이 정책은 리소스를 "*"로 지정하여 IAM 사용자에게 모든 역할에 대한 액세스 권한을 부여합니다. 그러나 이 경우 해당 역할(기존 역할 또는 이후에 생성할 역할)로 인스턴스를 시작하는 사용자에게 불필요한 권한까지 과도하게 부여해도 무방할지를 고려해야 합니다.

IAM 역할 작업

IAM 역할을 만들고 시작 도중 또는 후에 인스턴스에 연결할 수 있습니다. 인스턴스에 대한 IAM 역할을 교체하거나 분리할 수도 있습니다.

목차

- [IAM 역할 만들기 \(p. 459\)](#)
- [IAM 역할로 인스턴스 시작 \(p. 460\)](#)
- [IAM 역할을 인스턴스에 연결 \(p. 461\)](#)
- [IAM 역할 분리 \(p. 462\)](#)

- IAM 역할 교체 (p. 463)

IAM 역할 만들기

특정 역할로 인스턴스를 시작하거나 인스턴스에 연결하려면 우선 IAM 역할을 생성해야 합니다.

IAM 콘솔을 사용하여 IAM 역할을 생성하려면 다음을 수행합니다.

1. <https://console.aws.amazon.com/iam/>에서 IAM 콘솔에 로그인합니다.
2. 탐색 창에서 [Roles], [Create New Role]을 선택합니다.
3. [Set Role Name] 페이지에서 역할 이름을 입력하고 [Next Step]을 선택합니다.
4. [Select Role Type] 페이지에서 [Amazon EC2] 옆의 [Select]를 선택합니다.
5. [Attach Policy] 페이지에서 AWS 관리형 정책을 선택합니다. 예를 들어 Amazon EC2의 경우 다음 AWS 관리형 정책 중 하나가 적합할 수 있습니다.
 - PowerUserAccess
 - ReadOnlyAccess
 - AmazonEC2FullAccess
 - AmazonEC2ReadOnlyAccess
6. 역할 정보를 검토하고 필요에 따라 역할을 편집한 후 [Create Role]을 선택합니다.

또는 AWS CLI를 사용하여 IAM 역할을 만들 수 있습니다.

AWS CLI를 사용하여 IAM 역할 및 인스턴스 프로파일 만들기

- 역할이 Amazon S3 버킷을 사용하도록 허용하는 정책을 갖는 IAM 역할을 생성합니다.
 - a. 다음 트러스트 정책을 생성하고 `ec2-role-trust-policy.json`이라는 텍스트 파일로 저장합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": { "Service": "ec2.amazonaws.com"},  
            "Action": "sts:AssumeRole"  
        }  
    ]  
}
```

- b. `s3access` 역할을 만들고 생성한 신뢰 정책을 지정합니다.

```
aws iam create-role --role-name s3access --assume-role-policy-document file://ec2-  
role-trust-policy.json  
{  
    "Role": {  
        "AssumeRolePolicyDocument": {  
            "Version": "2012-10-17",  
            "Statement": [  
                {  
                    "Action": "sts:AssumeRole",  
                    "Effect": "Allow",  
                    "Principal": {  
                        "Service": "ec2.amazonaws.com"  
                    }  
                }  
            ]  
        }  
    }  
}
```

```
        },
        "RoleId": "AROAIIZKPBKS2LEXAMPLE",
        "CreateDate": "2013-12-12T23:46:37.247Z",
        "RoleName": "s3access",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/s3access"
    }
}
```

- c. 액세스 정책을 생성하고 `ec2-role-access-policy.json`이라는 텍스트 파일로 저장합니다. 예를 들어 이 정책은 인스턴스에서 실행되는 애플리케이션에 Amazon S3 관리 권한을 부여합니다.

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

- d. 역할에 액세스 정책을 연결합니다.

```
aws iam put-role-policy --role-name s3access --policy-name S3-Permissions --policy-document file://ec2-role-access-policy.json
```

- e. `s3access-profile`이라는 인스턴스 프로파일을 생성합니다.

```
aws iam create-instance-profile --instance-profile-name s3access-profile
{
    "InstanceProfile": {
        "InstanceProfileId": "AIPAJTLBPJLEGREXAMPLE",
        "Roles": [],
        "CreateDate": "2013-12-12T23:53:34.093Z",
        "InstanceProfileName": "s3access-profile",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:instance-profile/s3access-profile"
    }
}
```

- f. `s3access-profile` 인스턴스 프로파일에 `s3access` 역할을 추가합니다.

```
aws iam add-role-to-instance-profile --instance-profile-name s3access-profile --role-name s3access
```

이러한 명령에 대한 자세한 내용은 AWS Command Line Interface Reference에서 [create-role](#), [put-role-policy](#) 및 [create-instance-profile](#)을 참조하십시오.

또는 다음 Windows PowerShell용 AWS 도구 명령을 사용할 수 있습니다.

- [New-IAMRole](#)
- [Register-IAMRolePolicy](#)
- [New-IMAMInstanceProfile](#)

IAM 역할로 인스턴스 시작

IAM 역할을 생성한 후 인스턴스를 시작하면서 해당 역할을 연결할 수 있습니다.

Important

IAM 역할을 생성한 후 권한이 전파되기까지 몇 초가 걸릴 수 있습니다. 특정 역할로 인스턴스를 처음 시작하려는 시도가 실패할 경우 몇 초간 기다린 후에 다시 시도해 보십시오. 자세한 내용은 IAM 사용 설명서에서 [Troubleshooting Working with Roles](#)를 참조하십시오.

IAM 역할로 인스턴스를 시작하려면 다음을 수행합니다. 삭제하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 대시보드에서 Launch Instance를 선택합니다.
3. AMI와 인스턴스 유형을 선택한 후 [Next: Configure Instance Details]를 선택합니다.
4. [Configure Instance Details] 페이지의 [IAM role]에서 생성한 IAM 역할을 선택합니다.

Note

IAM 역할 생성 시 생성된 인스턴스 프로파일 이름이 [IAM role] 목록에 표시됩니다. 콘솔을 사용하여 IAM 역할을 생성한 경우 인스턴스 프로파일이 자동으로 생성되어 역할과 동일한 이름이 지정된 상태입니다. AWS CLI, API 또는 AWS SDK를 사용하여 IAM 역할을 생성한 경우 인스턴스 프로파일의 이름이 다를 수 있습니다.

5. 기타 세부 정보를 구성하고 지침에 따라 마법사의 나머지 절차를 완료하거나, [Review and Launch]를 선택하여 기본 설정을 수락하고 [Review Instance Launch] 페이지로 바로 이동합니다.
6. 설정을 검토한 다음 [Launch]를 선택하여 키 페어를 선택하고 인스턴스를 시작합니다.
7. 애플리케이션에서 Amazon EC2 API 작업을 사용하는 경우 인스턴스에 제공된 AWS 보안 자격 증명을 검색하고 이를 사용하여 요청에 서명합니다. AWS SDK는 이 작업을 자동으로 수행합니다.

```
curl http://169.254.169.254/latest/meta-data/iam/security-credentials/role_name
```

또는 AWS CLI를 사용하여 시작 중에 인스턴스와 역할을 연결할 수 있습니다. 명령에 인스턴스 프로파일을 지정해야 합니다.

AWS CLI를 사용하여 IAM 역할로 인스턴스를 시작하려면 다음을 수행합니다.

1. [run-instances](#) 명령을 사용하여 인스턴스 프로파일을 사용하는 인스턴스를 시작합니다. 다음 예제에서는 인스턴스 프로파일과 함께 인스턴스를 시작하는 방법을 보여 줍니다.

```
aws ec2 run-instances --image-id ami-11aa22bb --iam-instance-profile Name="s3access-profile" --key-name my-key-pair --security-groups my-security-group --subnet-id subnet-1a2b3c4d
```

또는 [New-EC2Instance](#) Windows PowerShell용 도구 명령을 사용합니다.

2. 애플리케이션에서 Amazon EC2 API 작업을 사용하는 경우 인스턴스에 제공된 AWS 보안 자격 증명을 검색하고 이를 사용하여 요청에 서명합니다. AWS SDK는 이 작업을 자동으로 수행합니다.

```
curl http://169.254.169.254/latest/meta-data/iam/security-credentials/role_name
```

IAM 역할을 인스턴스에 연결

IAM 역할을 만든 후 실행 중이거나 중지된 인스턴스에 연결할 수 있습니다.

콘솔을 사용하여 IAM 역할을 인스턴스에 연결하기

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.

2. 탐색 창에서 Instances를 선택합니다.
3. 인스턴스를 선택하고 나서 [Actions], [Instance Settings], [Attach/Replace IAM role]을 차례대로 선택합니다.
4. 인스턴스에 연결할 IAM 역할을 선택한 후 [Apply]를 선택합니다.

AWS CLI를 사용하여 IAM 역할을 인스턴스에 연결하기

1. 필요한 경우 인스턴스를 설명하여 역할을 연결할 인스턴스의 ID를 가져옵니다.

```
aws ec2 describe-instances
```

2. **associate-iam-instance-profile** 명령을 사용하여 인스턴스 프로파일을 지정하여 인스턴스에 IAM 역할을 연결합니다. 인스턴스 프로파일의 Amazon 리소스 이름(ARN) 또는 이름을 사용할 수 있습니다.

```
aws ec2 associate-iam-instance-profile --instance-id i-1234567890abcdef0 --iam-instance-profile Name="TestRole-1"

{
    "IamInstanceProfileAssociation": {
        "InstanceId": "i-1234567890abcdef0",
        "State": "associating",
        "AssociationId": "iip-assoc-0dbd8529a48294120",
        "IamInstanceProfile": {
            "Id": "AIPAJLNLDX3AMYZNWYYAY",
            "Arn": "arn:aws:iam::123456789012:instance-profile/TestRole-1"
        }
    }
}
```

또는 다음 Windows PowerShell용 도구 명령을 사용합니다.

- [Get-EC2Instance](#)
- [Register-EC2IamInstanceProfile](#)

IAM 역할 분리

실행 중이거나 중지된 인스턴스에서 IAM 역할을 분리할 수 있습니다.

콘솔을 사용하여 인스턴스에서 IAM 역할 분리

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. 인스턴스를 선택하고 나서 [Actions], [Instance Settings], [Attach/Replace IAM role]을 차례대로 선택합니다.
4. [IAM role]에서 [No Role]을 선택합니다. Apply를 선택합니다.
5. 확인 대화 상자에서 [Yes, Detach]를 선택합니다.

AWS CLI를 사용하여 인스턴스에서 IAM 역할 분리

1. 필요한 경우 [describe-iam-instance-profile-associations](#)를 사용하여 IAM 인스턴스 프로파일 연결을 설명하고 분리할 IAM 인스턴스 프로파일의 연결 ID를 가져옵니다.

```
aws ec2 describe-iam-instance-profile-associations
```

```
{  
    "IamInstanceProfileAssociations": [  
        {  
            "InstanceId": "i-088ce778fbfeb4361",  
            "State": "associated",  
            "AssociationId": "iip-assoc-0044d817db6c0a4ba",  
            "IamInstanceProfile": {  
                "Id": "AIPAJEDNCAA64SSD265D6",  
                "Arn": "arn:aws:iam::123456789012:instance-profile/TestRole-2"  
            }  
        }  
    ]  
}
```

2. [disassociate-iam-instance-profile](#) 명령을 사용하여 연결 ID를 사용 중인 IAM 인스턴스 프로파일을 분리합니다.

```
aws ec2 disassociate-iam-instance-profile --association-id iip-assoc-0044d817db6c0a4ba  
  
{  
    "IamInstanceProfileAssociation": {  
        "InstanceId": "i-087711ddaf98f9489",  
        "State": "disassociating",  
        "AssociationId": "iip-assoc-0044d817db6c0a4ba",  
        "IamInstanceProfile": {  
            "Id": "AIPAJEDNCAA64SSD265D6",  
            "Arn": "arn:aws:iam::123456789012:instance-profile/TestRole-2"  
        }  
    }  
}
```

또는 다음 Windows PowerShell용 도구 명령을 사용합니다.

- [Get-EC2IamInstanceProfileAssociation](#)
- [Unregister-EC2IamInstanceProfile](#)

IAM 역할 교체

실행 중인 인스턴스에 대한 IAM 역할을 교체할 수 있습니다. 예를 들어 인스턴스에서 실행 중인 애플리케이션이 수행한 API 작업이 중단되지 않도록 하려는 경우 등 기존 인스턴스를 먼저 분리하지 않고도 인스턴스에 대한 IAM 역할을 변경하고자 할 때 이 작업을 수행할 수 있습니다.

콘솔을 사용하여 인스턴스에 대한 IAM 역할 교체

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. 인스턴스를 선택하고 나서 [Actions], [Instance Settings], [Attach/Replace IAM role]을 차례대로 선택합니다.
4. 인스턴스에 연결할 IAM 역할을 선택한 후 [Apply]를 선택합니다.

AWS CLI를 사용하여 인스턴스에 대한 IAM 역할 교체

1. 필요한 경우 IAM 인스턴스 프로파일 연결을 설명하여 교체할 IAM 인스턴스 프로파일의 연결 ID를 가져옵니다.

```
aws ec2 describe-iam-instance-profile-associations
```

- replace-iam-instance-profile-association 명령을 사용하여 기존 인스턴스 프로파일에 대한 연결 ID와 교체할 인스턴스 프로파일의 ARN 또는 이름을 지정하여 IAM 인스턴스 프로파일을 교체합니다.

```
aws ec2 replace-iam-instance-profile-association --association-id iip-assoc-0044d817db6c0a4ba --iam-instance-profile Name="TestRole-2"
```

```
{  
    "IamInstanceProfileAssociation": {  
        "InstanceId": "i-087711ddaf98f9489",  
        "State": "associating",  
        "AssociationId": "iip-assoc-09654be48e33b91e0",  
        "IamInstanceProfile": {  
            "Id": "AIPAJCJEDKX7QYHWYK7GS",  
            "Arn": "arn:aws:iam::123456789012:instance-profile/TestRole-2"  
        }  
    }  
}
```

또는 다음 Windows PowerShell용 도구 명령을 사용합니다.

- Get-EC2IamInstanceProfileAssociation
- Set-EC2IamInstanceProfileAssociation

Linux 인스턴스의 인바운드 트래픽 권한 부여

보안 그룹을 통해 인스턴스로 들어오는 트래픽 및 인스턴스에 도달할 수 있는 트래픽의 유형을 제어할 수 있습니다. 예를 들어 흔 네트워크에 속하는 컴퓨터만 SSH를 사용하여 인스턴스에 액세스하도록 허용할 수 있습니다. 인스턴스가 웹 서버인 경우 외부 사용자가 웹 서버의 콘텐츠를 탐색할 수 있도록 모든 IP 주소가 HTTP를 통해 인스턴스에 액세스하도록 허용할 수 있습니다.

인스턴스에 대한 네트워크 액세스를 가능하게 하려면 인스턴스의 인바운드 트래픽을 허용해야 합니다. 인바운드 트래픽을 위한 포트를 열려면 인스턴스 시작 시 연결한 보안 그룹에 규칙을 추가합니다.

인스턴스에 연결하려면 컴퓨터의 퍼블릭 IPv4 주소에서 비롯되는 SSH 트래픽을 인증하는 규칙을 설정해야 합니다. 추가 IP 주소 범위에서 비롯되는 SSH 트래픽을 허용하려면 인증할 각 범위에 대해 다른 규칙을 추가합니다.

IPv6용 VPC를 활성화하고 IPv6 주소로 인스턴스를 시작했다면, 퍼블릭 IPv4 주소 대신 IPv6 주소를 사용해 인스턴스에 연결할 수 있습니다. 로컬 컴퓨터에 IPv6 주소가 있고 IPv6를 사용하도록 컴퓨터를 구성해야 합니다.

Windows 인스턴스에 대한 네트워크 액세스를 가능하게 하려면 Windows 인스턴스용 Amazon EC2 사용 설명서에서 [Windows 인스턴스의 인바운드 트래픽 권한 부여](#) 섹션을 참조하십시오.

시작하기 전

인스턴스에 액세스해야 하는 대상(예: 단일 호스트 또는 신뢰할 수 있는 특정 네트워크, 로컬 컴퓨터의 퍼블릭 IPv4 주소)을 결정합니다. Amazon EC2 콘솔의 보안 그룹 편집기는 로컬 컴퓨터의 퍼블릭 IPv4 주소를 자동으로 검색할 수 있습니다. 또는 인터넷 브라우저에서 "내 IP 주소"와 같은 검색 구문을 사용하거나 <http://checkip.amazonaws.com/> 서비스를 사용할 수도 있습니다. 고정 IP 주소 없이 ISP 또는 방화벽을 경유하여 연결하는 경우에는 클라이언트 컴퓨터가 사용하는 IP 주소의 범위를 알아내야 합니다.

Warning

0.0.0.0/0을 사용하면 모든 IPv4 주소를 통해 SSH를 사용하는 인스턴스에 액세스할 수 있습니다.
::/0을 사용하면 모든 IPv6 주소를 사용해 인스턴스에 액세스할 수 있습니다. 테스트 환경에서 잠시

사용하는 것은 괜찮지만 프로덕션 환경에서는 안전하지 않습니다. 프로덕션에서는 특정 IP 주소나 주소 범위만 인스턴스에 액세스하도록 허용하십시오.

보안 그룹에 대한 자세한 내용은 [Linux 인스턴스에 대한 Amazon EC2 보안 그룹 \(p. 385\)](#)을 참조하십시오.

Linux 인스턴스의 인바운드 SSH 트래픽에 대한 규칙 추가

보안 그룹은 연결된 인스턴스에 대한 방화벽 역할을 하여 인스턴스 수준에서 인바운드 트래픽과 아웃바운드 트래픽을 모두 제어합니다. SSH를 사용하여 IP 주소에서 Linux 인스턴스에 연결할 수 있게 하는 규칙을 보안 그룹에 추가해야 합니다.

콘솔을 사용하여 보안 그룹에 IPv4를 통한 인바운드 SSH 트래픽에 대한 규칙을 추가하려면 다음을 수행합니다.

1. Amazon EC2 콘솔의 탐색 창에서 [Instances]를 선택합니다. 인스턴스를 선택하고 [Description] 탭을 확인합니다. 인스턴스에 연결된 보안 그룹이 [Security groups]에 나열됩니다. [view rules]를 선택하여 인스턴스에 적용되고 있는 규칙 목록을 표시합니다.
2. 탐색 창에서 [Security Groups]를 선택합니다. 인스턴스에 연결된 보안 그룹 중 하나를 선택합니다.
3. 세부 정보 창의 [Inbound] 탭에서 [Edit]를 선택합니다. 대화 상자에서 [Add Rule]를 선택하고 [Type] 목록에서 [SSH]를 선택합니다.
4. 필드를 로컬 컴퓨터의 퍼블릭 IPv4 주소로 자동으로 채우려면 [Source] 필드에서 [My IP]를 선택하면 됩니다. 또는 [Custom]을 선택하고 컴퓨터 또는 네트워크의 퍼블릭 IPv4 주소를 CIDR 표기법으로 지정해도 됩니다. 예를 들어, IPv4 주소가 203.0.113.25인 경우 이 단일 IPv4 주소를 CIDR 표기법으로 나열하려면 203.0.113.25/32를 지정합니다. 회사에서 주소를 범위로 할당하는 경우 전체 범위(예: 203.0.113.0/24)를 지정합니다.

IP 주소 확인에 대한 자세한 내용은 [시작하기 전 \(p. 464\)](#) 섹션을 참조하십시오.

5. [Save]를 선택합니다.

(VPC만 해당) IPv6 주소로 인스턴스를 시작했는데 IPv6 주소를 사용해 인스턴스에 연결하려면 SSH를 통한 인바운드 IPv6 트래픽을 허용하는 규칙을 추가해야 합니다.

콘솔을 사용하여 보안 그룹에 IPv6를 통한 인바운드 SSH 트래픽에 대한 규칙을 추가하려면 다음을 수행합니다.

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다. 인스턴스에 대한 보안 그룹을 선택합니다.
3. [Inbound], [Edit], [Add Rule]를 선택합니다.
4. [Type]의 경우 [SSH]를 선택합니다.
5. [Source] 필드에 컴퓨터의 IPv6 주소를 CIDR 표기법으로 지정합니다. 예를 들어, IPv6 주소가 2001:db8:1234:1a00:9691:9503:25ad:1761인 경우 단일 IP 주소를 CIDR 표기법으로 나열하려면 2001:db8:1234:1a00:9691:9503:25ad:1761/128을 지정합니다. 회사에서 주소를 범위로 할당하는 경우 전체 범위(예: 2001:db8:1234:1a00::/64)를 지정합니다.
6. [Save]를 선택합니다.

명령줄을 사용하여 보안 그룹에 규칙을 추가하려면 다음을 수행합니다.

다음 명령 중 하나를 사용할 수 있습니다. 인스턴스 자체가 아닌 로컬 시스템에서 이 명령을 실행해야 합니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [authorize-security-group-ingress\(AWS CLI\)](#)
- [Grant-EC2SecurityGroupIngress\(Windows PowerShell용 AWS 도구\)](#)

인스턴스에 보안 그룹 할당

인스턴스를 시작할 때 인스턴스에 보안 그룹을 할당할 수 있습니다. 규칙을 추가하거나 제거하면 해당 보안 그룹을 할당한 모든 인스턴스에 변경 내용이 자동으로 적용됩니다.

EC2-Classic에서 인스턴스를 시작한 후에는 해당 보안 그룹을 변경할 수 없습니다. VPC에서 인스턴스를 시작한 이후에는 해당 보안 그룹을 변경할 수 있습니다. 자세한 내용은 Amazon VPC 사용 설명서에서 [인스턴스의 보안 그룹 변경](#)을 참조하십시오.

Amazon EC2와 Amazon Virtual Private Cloud

Amazon Virtual Private Cloud(Amazon VPC)를 사용하면 Virtual Private Cloud(VPC)로 알려져 있는 AWS 클라우드에서 논리적으로 독립된 고유 영역에 가상 네트워크를 정의할 수 있습니다. 인스턴스와 같은 AWS 리소스를 VPC로 실행할 수 있습니다. VPC는 고객의 자체 데이터 센터에서 운영하는 기존 네트워크와 매우 유사하지만 AWS의 확장 가능한 인프라를 사용한다는 이점을 제공합니다. 해당 IP 주소 범위를 선택하고, 서브넷을 만든 후 라우팅 테이블, 네트워크 게이트웨이 및 보안 설정을 구성하여 VPC를 구성할 수 있습니다. VPC의 인스턴스를 인터넷에 연결합니다. VPC를 사내 데이터 센터에 연결하여 AWS 클라우드에서 데이터 센터를 확장할 수 있습니다. 각의 서브넷에서 리소스를 보호하기 위해 보안 그룹 및 네트워크 액세스 제어 목록을 포함한 다중 보안 계층을 사용할 수 있습니다. 자세한 내용은 [Amazon VPC 사용 설명서](#) 항목을 참조하십시오.

계정에서 리전별로 EC2-VPC와 EC2-Classic 플랫폼을 모두 지원할 수 있습니다. 2013년 12월 4일 이후에 계정을 생성한 경우에는 EC2-VPC만 지원됩니다. 계정에서 지원하는 플랫폼을 확인하려면 [지원되는 플랫폼 \(p. 471\)](#) 섹션을 참조하십시오. 계정에서 EC2-VPC만 지원하는 경우 기본 VPC가 자동으로 생성됩니다. 기본 VPC는 이미 구성되어 즉시 사용할 수 있는 VPC입니다. 기본 VPC로 인스턴스를 즉시 시작할 수 있습니다. 자세한 내용은 Amazon VPC 사용 설명서의 [기본 VPC 및 서브넷](#)을 참조하십시오. 계정에서 EC2-Classic과 EC2-VPC를 지원하는 경우 두 플랫폼 중 하나로 인스턴스를 시작할 수 있습니다. 계정에서 지원하는 플랫폼과 상관없이 기본이 아닌 VPC를 직접 생성할 수 있으며 필요에 따라 구성할 수 있습니다.

목차

- [VPC의 장점 \(p. 466\)](#)
- [EC2-Classic와 EC2-VPC의 차이점 \(p. 467\)](#)
- [EC2-Classic과 EC2-VPC 간 리소스 공유 및 액세스 \(p. 468\)](#)
- [VPC에서만 사용할 수 있는 인스턴스 유형 \(p. 470\)](#)
- [Amazon VPC 문서 \(p. 470\)](#)
- [지원되는 플랫폼 \(p. 471\)](#)
- [ClassicLink \(p. 472\)](#)
- [Linux 내 EC2-Classic 인스턴스에서 VPC 내 Linux 인스턴스로 마이그레이션 \(p. 481\)](#)

VPC의 장점

EC2-Classic 대신 VPC로 인스턴스를 실행하면 다음의 장점이 있습니다.

- 인스턴스의 시작/중지에 상관 없이 유지되는 고정 프라이빗 IPv4 주소 할당
- 인스턴스에 여러 개의 IPv4 주소 할당이 가능합니다.
- 네트워크 인터페이스를 정의하고, 하나 혹은 그 이상의 네트워크 인터페이스를 VPC 인스턴스에 설치 가능합니다.
- 인스턴스가 실행중이라도 상관없이, 인스턴스의 보안 그룹 멘버십 변경이 가능합니다.
- 인스턴스의 인바운드 트래픽 제어(인그레스 필터링) 뿐만 아니라 아웃바운드 트래픽도 제어(이그레스 필터링) 가능합니다.

- 네트워크 액세스 제어 리스트(ACL)를 통해, 인스턴스에 대한 액세스 보안이 한단계 더 강화되었습니다.
- 단일 테넌트 하드웨어에서 인스턴스 실행
- 인스턴스에 IPv6 주소 할당이 가능합니다.

EC2-Classic와 EC2-VPC의 차이점

다음 표에서는 EC2-Classic, 기본 VPC에서 시작되는 인스턴스, 기본이 아닌 VPC에서 시작되는 인스턴스의 차이점을 요약합니다.

특성	EC2-Classic	기본 VPC	기본이 아닌 VPC
퍼블릭 IPv4 주소(Amazon 퍼블릭 IP 주소 풀에서 제공)	인스턴스에 퍼블릭 IPv4 주소가 할당됩니다.	실행 시 따로 설정하거나 서브넷의 공인 IPv4 주소 속성을 변경하지 않은 경우 기본 서브넷 실행 인스턴스에 퍼블릭 IPv4 주소가 기본으로 할당됩니다.	시작 시 따로 지정하거나 서브넷의 퍼블릭 IPv4 주소 속성을 변경하지 않는 한 퍼블릭 IPv4 주소는 인스턴스에 기본으로 할당되지 않습니다.
프라이빗 IPv4 주소	인스턴스를 시작할 때마다 EC2-Classic 범위 내의 프라이빗 IPv4 주소가 할당됩니다.	인스턴스를 시작할 때마다 기본 VPC 주소 범위 내의 고정 프라이빗 IPv4 주소가 할당됩니다.	인스턴스를 시작할 때마다 VPC 주소 범위 내의 고정 프라이빗 IPv4 주소가 할당됩니다.
다중 프라이빗 IPv4 주소	인스턴스별로 하나의 프라이빗 IP 주소가 할당되며 다른 IP 주소는 지원되지 않습니다.	인스턴스에 다중 프라이빗 IPv4 주소를 할당할 수 있습니다.	인스턴스에 다중 프라이빗 IPv4 주소를 할당할 수 있습니다.
탄력적 IP 주소 (IPv4)	인스턴스를 종지하면 탄력적 IP는 인스턴스에서 연결해제됩니다.	인스턴스를 종지하면 탄력적 IP는 인스턴스와 연결된 상태를 유지합니다.	인스턴스를 종지하면 탄력적 IP는 인스턴스와 연결된 상태를 유지합니다.
DNS 호스트 이름	기본적으로 DNS 호스트 이름을 사용하도록 되어있습니다.	기본적으로 DNS 호스트 이름을 사용하도록 되어있습니다.	기본적으로 DNS 호스트 이름을 사용하지 않도록 되어있습니다.
보안 그룹	보안 그룹에서 다른 AWS 계정에 속한 보안 그룹을 참조할 수 있습니다. 리전당 최대 500개의 보안 그룹을 만드실 수 있습니다.	보안 그룹에서는 사용자의 VPC 내 보안 그룹만 참조할 수 있습니다. VPC당 최대 100개의 보안 그룹을 만드실 수 있습니다.	보안 그룹에서는 사용자의 VPC 내 보안 그룹만 참조할 수 있습니다. VPC당 최대 100개의 보안 그룹을 만드실 수 있습니다.
보안 그룹 연결	인스턴스를 실행할 때 할당할 수 있는 보안 그룹 수에 제한이 없습니다. 실행 중인 인스턴스의 보안 그룹은 변경할 수 없습니다. 할당된 보안 그룹의 규칙을 변경하시거나, 인스턴스를 새로운 것으로 교체하시면 됩니다(인스턴스에서 AMI 생성 -> 해당 AMI의 인스턴스를 원하는 보안 그룹에 연결하여 실행 -> 기존 인스턴스에 할당했던 EIP 주소 연	인스턴스당 최대 5개의 보안 그룹을 할당할 수 있습니다. 인스턴스 실행 시, 그리고 실행 중에도 보안 그룹을 할당할 수 있습니다.	인스턴스당 최대 5개의 보안 그룹을 할당할 수 있습니다. 인스턴스 실행 시, 그리고 실행 중에도 보안 그룹을 할당할 수 있습니다.

특성	EC2-Classic	기본 VPC	기본이 아닌 VPC
	결을 해제하고 해당 주소를 새 인스턴스에 할당 -> 기존 인스턴스 종료).		
보안 그룹 규칙	인바운드 트래픽에만 규칙을 추가할 수 있습니다. 보안 그룹당 최대 100개의 규칙을 추가할 수 있습니다.	인바운드 및 아웃바운드 모두에 규칙을 지정 할 수 있습니다. 보안 그룹당 최대 50개의 규칙을 추가할 수 있습니다.	인바운드 및 아웃바운드 모두에 규칙을 지정 할 수 있습니다. 보안 그룹당 최대 50개의 규칙을 추가할 수 있습니다.
테넌시	인스턴스가 공유된 하드웨어에서 실행됩니다.	공유된 하드웨어나 단일 테넌트 하드웨어에서 인스턴스를 실행할 수 있습니다.	공유된 하드웨어나 단일 테넌트 하드웨어에서 인스턴스를 실행할 수 있습니다.
인터넷 액세스	인스턴스에서 인터넷에 액세스할 수 있습니다. 인스턴스가 퍼블릭 IP 주소를 자동으로 수신하고 AWS 네트워크 엣지를 통해 직접 인터넷에 액세스할 수 있습니다.	기본적으로 인스턴스가 인터넷에 액세스할 수 있습니다. 인스턴스에 퍼블릭 IP 주소가 기본으로 할당됩니다. 인터넷 게이트웨이가 기본 VPC에 연결되고 기본 서브넷에 인터넷 게이트웨이로 연결되는 경로가 있습니다.	기본적으로 인스턴스가 인터넷에 액세스할 수 없습니다. 인스턴스에 퍼블릭 IP 주소가 기본으로 할당되지 않습니다. 생성된 방법에 따라 VPC에 인터넷 게이트웨이가 있을 수 있습니다.
IPv6 주소 지정	IPv6 주소 지정은 지원되지 않습니다. 인스턴스에 IPv6 주소를 할당할 수 없습니다.	IPv6 CIDR 블록을 VPC에 연결하고 IPv6 주소를 VPC의 인스턴스에 할당할 수도 있습니다.	IPv6 CIDR 블록을 VPC에 연결하고 IPv6 주소를 VPC의 인스턴스에 할당할 수도 있습니다.

다음 다이어그램은 각 플랫폼의 인스턴스를 보여 줍니다. 다음을 참조하십시오.

- 인스턴스 1, 2, 3, 4는 EC2-Classic 플랫폼에 있습니다. 1과 2는 한 계정에서 시작되었고 3과 4는 다른 계정에서 시작되었습니다. 이러한 인스턴스들은 서로 통신이 가능할 뿐만 아니라 직접 인터넷에 액세스할 수도 있습니다.
- 인스턴스 5, 6는 EC2-VPC 플랫폼에서 같은 VPC의 다른 서브넷에 있습니다. 이러한 인스턴스는 VPC를 소유하는 계정에서 시작되었으며 이 VPC에서 다른 계정은 인스턴스를 시작할 수 없습니다. 이러한 인스턴스들은 서로 통신이 가능할 뿐만 아니라 인터넷 게이트웨이를 통해 EC2-Classic 인스턴스와 인터넷에 액세스할 수도 있습니다.

EC2-Classic과 EC2-VPC 간 리소스 공유 및 액세스

고객님의 AWS 계정에 부여된 리소스 및 기능 중 일부는 EC2-Classic 및 EC2-VPC 플랫폼간에, 예를 들어 ClassicLink(를) 통해서 공유하거나 액세스가 가능합니다. ClassicLink에 대한 자세한 내용은 [ClassicLink \(p. 472\)](#) 섹션을 참조하십시오.

EC2-Classic 지원 계정을 사용하는 경우 EC2-Classic에서 사용할 리소스를 설정했을 수 있습니다. EC2-Classic에서 VPC로 마이그레이션하기 위해서는 해당 리소스를 VPC에서 다시 만들어야 합니다. EC2-Classic에서 VPC로의 마이그레이션에 대한 자세한 내용은 [Linux 내 EC2-Classic 인스턴스에서 VPC 내 Linux 인스턴스로 마이그레이션 \(p. 481\)](#)을(를) 참조하십시오.

다음은 EC2-Classic와 VPC 간에 공유나 액세스가 가능한 리소스입니다.

Resource	참고
AMI	
번들 작업	
EBS 볼륨	
탄력적 IP 주소(IPv4)	<p>탄력적 IP 주소는 EC2-Classic에서 EC2-VPC로 마이그레이션할 수 있습니다. 처음부터 EC2-VPC에서 사용할 목적으로 할당한 탄력적 IP 주소는 EC2-Classic으로 마이그레이션하지 못합니다. 자세한 내용은 EC2-Classic에서 EC2-VPC로 탄력적 IP 주소의 마이그레이션 (p. 507) 섹션을 참조하십시오.</p>
인스턴스	<p>EC2-Classic 인스턴스는 퍼블릭 IPv4 주소를 할당 받은 VPC의 인스턴스와 통신할 수 있으며, 프라이빗 IPv4 주소를 가지고 있는 인스턴스와의 통신을 원하는 경우 ClassicLink를 사용할 수 있습니다.</p> <p>EC2-Classic에서 VPC로 인스턴스를 마이그레이션 할 수 없습니다. 그러나 EC2-Classic 내 인스턴스에서 VPC 내 인스턴스로는 애플리케이션을 마이그레이션 할 수 있습니다. 자세한 내용은 Linux 내 EC2-Classic 인스턴스에서 VPC 내 Linux 인스턴스로 마이그레이션 (p. 481) 섹션을 참조하십시오.</p>
키 쌍	
로드 밸런서	<p>ClassicLink를 사용하는 경우 VPC 내 링크한 EC2-Classic 인스턴스에 로드 밸런서를 등록할 수 있습니다. 이때 VPC가 인스턴스처럼 동일한 가용 영역에 서브넷을 보유해야 합니다.</p> <p>EC2-Classic에서 VPC로 로드 밸런서를 마이그레이션 할 수 없습니다. VPC 내 EC2-Classic에 인스턴스를 등록할 수 없습니다.</p>
플레이스먼트 그룹	
예약 인스턴스	<p>예약 인스턴스를 실행할 네트워크 플랫폼을 EC2-Classic에서 EC2-VPC로 변경할 수 있습니다. 자세한 내용은 표준 예약 인스턴스 변경 (p. 193) 섹션을 참조하십시오.</p>
보안 그룹	<p>링크한 EC2-Classic 인스턴스는 ClassicLink를(를) 통해서 VPC 보안 그룹을 사용하여 VPC로 그리고 VPC로부터 트래픽을 제어할 수 있습니다. VPC 인스턴스에서 EC2-Classic 보안 그룹을 사용할 수 없습니다.</p> <p>EC2-Classic에서 VPC로 보안 그룹을 마이그레이션 할 수 없습니다. EC2-Classic 내 보안 그룹에서 VPC 내 보안 그룹으로 규칙을 복사할 수 있습니다. 자세한 내용은 보안 그룹 생성 (p. 389)을 참조하십시오.</p>
스냅샷	

다음은 EC2-Classic와 VPC 간에 공유나 이동이 불가능한 리소스입니다.

- 스팟 인스턴스

VPC에서만 사용할 수 있는 인스턴스 유형

다음 인스턴스 유형의 인스턴스는 EC2-Classic에서 지원되지 않으며 VPC에서 시작해야 합니다.

- C4
- I3
- M4
- P2
- R4
- T2
- X1

계정에서 EC2-Classic을(를) 지원하며 기본이 아닌 VPC를 생성하지 않은 경우 다음 중 하나를 수행하여 VPC 전용 인스턴스를 시작할 수 있습니다.

- 기본이 아닌 VPC를 생성하고 요청에 서브넷 ID 또는 네트워크 인터페이스 ID를 지정하여 VPC 전용 인스턴스를 시작하십시오. 기본 VPC가 없고 AWS CLI, Amazon EC2 API 또는 AWS SDK를 사용하여 VPC 전용 인스턴스를 시작하는 경우 기본이 아닌 VPC를 생성해야 합니다. 자세한 내용은 [Virtual Private Cloud\(VPC\) 생성 \(p. 19\)](#)을(를) 참조하십시오.
- Amazon EC2 콘솔 대시보드를 사용하여 VPC 전용 인스턴스를 시작합니다. Amazon EC2 콘솔은 계정에서 기본이 아닌 VPC를 생성하고 첫 가용 영역의 서브넷에서 인스턴스를 시작합니다. 콘솔은 다음 속성이 있는 VPC를 생성합니다.
 - 가용 영역마다 하나의 서브넷에서 퍼블릭 IPv4 주소 속성이 `true`로 설정되므로 인스턴스가 퍼블릭 IPv4 주소를 받습니다. 자세한 내용은 Amazon VPC 사용 설명서의 [VPC에서 IP 주소 지정](#)을 참조하십시오.
 - 인터넷 게이트웨이 및 VPC의 트래픽을 인터넷 게이트웨이로 라우팅하는 기본 라우팅 테이블입니다. 이를 통해 VPC에서 시작하는 인스턴스가 인터넷을 통해 통신할 수 있습니다. 자세한 내용은 [Amazon VPC 사용 설명서](#)의 Internet Gateways 섹션을 참조하십시오.
 - VPC의 기본 보안 그룹 및 각 서브넷과 연결된 기본 네트워크 ACL입니다. 자세한 내용은 Amazon VPC 사용 설명서의 [VPC의 보안](#)을 참조하십시오.

EC2-Classic에 기타 리소스가 있는 경우 EC2-VPC로 리소스를 마이그레이션할 수 있습니다. 자세한 내용은 [Linux 내 EC2-Classic 인스턴스에서 VPC 내 Linux 인스턴스로 마이그레이션 \(p. 481\)](#)을(를) 참조하십시오.

Amazon VPC 문서

Amazon VPC에 대한 자세한 내용은 다음 문서를 참조하십시오.

가이드	설명
Amazon VPC 시작 안내서	Amazon VPC 실습 소개
Amazon VPC 사용 설명서	Amazon VPC 사용에 관련된 자세한 정보를 제공합니다.
Amazon VPC 네트워크 관리자 안내서	네트워크 관리자가 고객의 게이트웨이를 구성하는 방법을 설명합니다.

지원되는 플랫폼

Amazon EC2에서는 다음과 같은 플랫폼을 지원합니다. AWS 계정에서는 리전별로 두 가지 플랫폼 모두 또는 EC2-VPC로만 인스턴스를 시작할 수 있습니다.

플랫폼	도입 시점	설명
EC2-Classic	Amazon EC2 최초 출시	다른 고객과 공유하는 단일 일반 네트워크에서 인스턴스가 실행됩니다.
EC2-VPC	Amazon VPC 최초 출시	AWS 계정에 속하도록 논리적으로 독립된 Virtual Private Cloud(VPC)에서 인스턴스가 실행됩니다.

계정에서 두 가지 플랫폼을 사용할 수 있는지에 대한 자세한 내용은 [가용성\(Amazon VPC 사용 설명서\)](#)을 참조하십시오. EC2-Classic과 EC2-VPC의 차이점에 대한 자세한 내용은 [EC2-Classic와 EC2-VPC의 차이점\(p. 467\)](#) 섹션을 참조하십시오.

Amazon EC2 콘솔에서 지원되는 플랫폼

Amazon EC2 콘솔은 사용자가 선택한 리전에서 인스턴스를 시작할 수 있는 플랫폼 및 해당 리전에 기본 VPC가 있는지 여부를 표시합니다.

탐색 모음에서 사용할 리전이 선택되어 있는지 확인합니다. Amazon EC2 콘솔 대시보드의 [Account Attributes]에서 [Supported Platforms]를 찾습니다. 두 개의 값(EC2와 VPC)이 있을 경우 두 플랫폼 중 하나에서 인스턴스를 시작할 수 있습니다. 하나의 값(VPC)만 있을 경우 EC2-VPC에서만 인스턴스를 시작할 수 있습니다.

EC2-VPC에서만 인스턴스를 시작할 수 있는 경우 기본 VPC가 자동으로 생성됩니다. 사용자가 직접 기본이 아닌 VPC를 생성하여 인스턴스 시작 시 지정한 경우가 아니면 인스턴스를 시작할 때 기본 VPC로 시작됩니다.

EC2-VPC

대시보드에서 [Account Attributes] 아래에 다음과 같은 정보가 표시되어 계정에서 EC2-VPC 플랫폼만 지원하며 ID가 vpc-1a2b3c4d인 기본 VPC가 있음을 알립니다.

계정에서 EC2-VPC만 지원하는 경우 시작 마법사를 사용하여 인스턴스를 시작할 때 [Network] 목록에서 VPC를, [Subnet] 목록에서 서브넷을 선택할 수 있습니다.

EC2-Classic, EC2-VPC

대시보드에서 [Account Attributes] 아래에 다음과 같은 정보가 표시되어 계정에서 EC2-Classic, EC2-VPC 플랫폼을 모두 지원함을 알립니다.

계정에서 EC2-Classic 및 EC2-VPC를 지원하는 경우 시작 마법사를 사용하여 [Network] 목록에서 [Launch into EC2-Classic]을 선택하면 EC2-Classic으로 시작할 수 있습니다. VPC를 시작하려면 [Network] 목록에서 VPC를, [Subnet] 목록에서 서브넷을 선택합니다.

관련 주제

인스턴스를 시작할 수 있는 플랫폼을 확인하는 방법에 대한 자세한 내용은 [지원되는 플랫폼 확인\(Amazon VPC 사용 설명서\)](#)을 참조하십시오.

ClassicLink

ClassicLink를 사용하면 EC2-Classic 인스턴스를 같은 리전 내에 있는 계정의 VPC에 연결할 수 있습니다. 이를 통해 VPC 보안 그룹을 EC2-Classic 인스턴스에 연결하여 EC2-Classic 인스턴스와 VPC의 인스턴스가 프라이빗 IPv4 주소를 사용해 서로 통신하도록 허용할 수 있습니다. ClassicLink를 사용하면 이러한 플랫폼의 인스턴스 간 통신을 위해 퍼블릭 IPv4 주소 또는 탄력적 IP 주소를 사용할 필요가 없습니다. 프라이빗 및 퍼블릭 IPv4 주소에 대한 자세한 내용은 [VPC의 IP 주소 지정](#)을 참조하십시오.

ClassicLink는 EC2-Classic 플랫폼을 지원하는 계정을 갖는 모든 사용자에게 제공되며 모든 EC2-Classic 인스턴스에 사용할 수 있습니다. 계정에서 지원하는 플랫폼을 확인하려면 [지원되는 플랫폼 \(p. 471\)](#)을 참조하십시오. VPC를 사용하는 데 따르는 이점에 대한 자세한 내용은 [Amazon EC2와 Amazon Virtual Private Cloud \(p. 466\)](#) 섹션을 참조하십시오. 리소스를 VPC로 마이그레이션한 방법에 대한 자세한 내용은 [Linux 내 EC2-Classic 인스턴스에서 VPC 내 Linux 인스턴스로 마이그레이션 \(p. 481\)](#) 섹션을 참조하십시오.

ClassicLink 사용에 따르는 추가 요금은 없습니다. 데이터 전송 및 인스턴스 시간 사용량에 대한 표준 요금이 그대로 적용됩니다.

Note

EC2-Classic 인스턴스는 IPv6 통신에는 사용할 수 없습니다. IPv6 CIDR 블록을 VPC와 연결하고 IPv6 주소를 VPC의 리소스에 할당할 수 있지만 ClassicLinked 인스턴스와 VPC의 리소스 간 통신은 IPv4를 통해서만 이루어집니다.

항목

- [ClassicLink 기본 사항 \(p. 472\)](#)
- [ClassicLink의 제한 사항 \(p. 474\)](#)
- [ClassicLink 작업 \(p. 475\)](#)
- [API 및 CLI 개요 \(p. 478\)](#)
- [예: 3티어 웹 애플리케이션의 ClassicLink 보안 그룹 구성 \(p. 480\)](#)

ClassicLink 기본 사항

두 단계를 통해 ClassicLink를 사용하여 EC2-Classic 인스턴스를 VPC에 링크할 수 있습니다. 우선 VPC에서 ClassicLink를 활성화해야 합니다. 기본적으로는 격리 상태를 유지하기 위해 계정의 모든 VPC에서 ClassicLink가 비활성화됩니다. VPC에서 ClassicLink를 활성화하면 계정의 같은 리전에서 실행 중인 모든 EC2-Classic 인스턴스를 해당 VPC에 링크할 수 있습니다. 인스턴스를 링크할 때는 EC2-Classic 인스턴스에 연결할 보안 그룹을 VPC에서 선택합니다. 링크된 인스턴스는 VPC 보안 그룹에서 허용하는 경우 프라이빗 IP 주소를 사용하여 VPC의 인스턴스와 통신할 수 있습니다. EC2-Classic 인스턴스의 프라이빗 IP 주소는 VPC에 링크되어도 그대로 유지됩니다.

Note

인스턴스를 VPC에 링크하는 작업을 인스턴스 연결이라고도 합니다.

링크된 EC2-Classic 인스턴스는 VPC의 인스턴스와 통신할 수 있지만 VPC에 속하지 않습니다. 예를 들어 `DescribeInstances` API 요청 또는 Amazon EC2 콘솔의 [Instances] 화면을 사용하여 인스턴스를 나열하고 VPC로 필터링하면 VPC에 링크된 EC2-Classic 인스턴스는 결과로 반영되지 않습니다. 링크된 EC2-Classic 인스턴스를 확인하는 방법에 대한 자세한 내용은 [ClassicLink 가능 VPC 및 링크된 EC2-Classic 인스턴스 보기 \(p. 477\)](#) 섹션을 참조하십시오.

기본적으로 퍼블릭 DNS 호스트 이름을 사용하여 연결된 EC2-Classic 인스턴스에서 VPC의 인스턴스를 처리하는 경우, 호스트 이름은 해당 인스턴스의 퍼블릭 IP 주소로 확인됩니다. 퍼블릭 DNS 호스트 이름을 사용하여 VPC의 인스턴스에서 연결된 EC2-Classic 인스턴스를 처리하는 경우에도 동일합니다. 퍼블릭 DNS 호스트 이름이 프라이빗 IP 주소가 되도록 하려면 VPC에 대해 ClassicLink DNS 지원을 활성화하면 됩니다. 자세한 내용은 [ClassicLink DNS 지원 활성화 \(p. 477\)](#) 섹션을 참조하십시오.

인스턴스와 VPC 간에 ClassicLink 연결이 더 이상 필요하지 않은 경우 VPC에서 EC2-Classic 인스턴스의 링크를 해제할 수 있습니다. 이렇게 하면 VPC 보안 그룹이 EC2-Classic 인스턴스에서 분리됩니다. 링크된 EC2-Classic 인스턴스를 종지하면 자동으로 VPC와 링크가 해제됩니다. VPC에서 링크된 모든 EC2-Classic 인스턴스의 링크를 해제한 후 VPC에서 ClassicLink를 비활성화할 수 있습니다.

VPC의 다른 AWS 서비스와 함께 ClassicLink 사용

링크된 EC2-Classic 인스턴스는 VPC의 Amazon Redshift, Amazon ElastiCache, Elastic Load Balancing 및 Amazon RDS AWS 서비스에 액세스할 수 있습니다. 그러나 VPC의 인스턴스는 ClassicLink를 사용하여 EC2-Classic 플랫폼이 프로비저닝한 AWS 서비스에 액세스할 수 없습니다.

VPC에서 Elastic Load Balancing을 사용하는 경우 링크한 EC2-Classic 인스턴스에 로드 밸런서를 등록할 수 있습니다. 이때 VPC가 서브넷을 보유한 가용 영역에 인스턴스가 속해야 합니다. 링크된 EC2-Classic 인스턴스를 종료하면 로드 밸런서가 인스턴스의 등록을 해제합니다. VPC에서 로드 밸런서를 사용하는 방법에 대한 자세한 내용은 Elastic Load Balancing 사용 설명서에서 [Amazon VPC의 Elastic Load Balancing](#)을 참조하십시오.

Auto Scaling 사용 시에는 시작될 때 지정한 ClassicLink 가능 VPC에 자동으로 링크되는 인스턴스가 포함된 Auto Scaling 그룹을 생성할 수 있습니다. 자세한 내용은 Auto Scaling 사용 설명서에서 [VPC에 EC2-Classic 인스턴스 링크](#)를 참조하십시오.

VPC에서 Amazon RDS 인스턴스 또는 Amazon Redshift 클러스터를 사용하고 공개적으로 액세스 가능(인터넷에서 액세스 가능)한 경우, 기본적으로 연결된 EC2-Classic 인스턴스에서 이러한 리소스를 처리하는 데 사용하는 엔드포인트가 퍼블릭 IP 주소로 확인됩니다. 이러한 리소스에 공개적으로 액세스할 수 없는 경우에는 엔드포인트가 프라이빗 IP 주소로 확인됩니다. ClassicLink를 사용하여 프라이빗 IP를 통해 공개적으로 액세스 가능한 RDS 인스턴스 또는 Redshift 클러스터를 처리하려면 해당 프라이빗 IP 주소 또는 프라이빗 DNS 호스트 이름을 사용하거나 VPC에 대해 ClassicLink DNS 지원을 활성화해야 합니다.

프라이빗 DNS 호스트 이름 또는 프라이빗 IP 주소를 사용하여 RDS 인스턴스를 처리하는 경우 링크된 EC2-Classic 인스턴스에서 다중 AZ 배포에 장애 조치 지원을 사용할 수 없습니다.

Amazon EC2 콘솔을 사용하여 Amazon Redshift, Amazon ElastiCache 또는 Amazon RDS 리소스의 프라이빗 IP 주소를 확인할 수 있습니다.

VPC에서 AWS 리소스의 프라이빗 IP 주소를 확인하려면 다음을 수행합니다.

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.
3. [Description] 열에서 네트워크 인터페이스의 설명을 확인합니다. Amazon Redshift, Amazon ElastiCache 또는 Amazon RDS에 사용된 네트워크 인터페이스의 설명에는 서비스의 이름이 명시됩니다. 예를 들어 Amazon RDS 인스턴스에 연결된 네트워크 인터페이스의 설명은 `RDSNetworkInterface`입니다.
4. 필요한 네트워크 인터페이스를 선택합니다.
5. 세부 정보 창의 [Primary private IPv4 IP] 필드에서 프라이빗 IP 주소를 확인합니다.

ClassicLink 사용 제어

기본적으로 IAM 사용자에게는 ClassicLink 사용 권한이 없습니다. 사용자에게 VPC의 ClassicLink를 활성화/비활성화하고, ClassicLink 가능 VPC에 인스턴스를 링크하거나 링크를 해제하고, ClassicLink 가능 VPC 및 링크된 EC2-Classic 인스턴스를 조회하는 권한을 부여하는 IAM 정책을 생성할 수 있습니다. Amazon EC2의 IAM 정책에 대한 자세한 내용은 [Amazon EC2에 대한 IAM 정책 \(p. 401\)](#) 섹션을 참조하십시오.

ClassicLink 작업 관련 정책에 대한 자세한 내용은 [6. ClassicLink 작업 \(p. 443\)](#)의 예제를 참조하십시오.

ClassicLink의 보안 그룹

EC2-Classic 인스턴스를 VPC에 링크해도 EC2-Classic 보안 그룹에는 영향이 없습니다. 보안 그룹은 인스턴스가 송수신하는 모든 트래픽을 계속 제어합니다. VPC의 인스턴스가 송수신하는 트래픽은 여기에서 제

외되며, EC2-Classic 인스턴스에 연결한 VPC 보안 그룹에 의해 제어됩니다. 같은 VPC에 링크된 여러 EC2-Classic 인스턴스는 같은 VPC 보안 그룹에 연결되어 있는지 여부에 관계없이 VPC를 통해 서로 통신할 수 없습니다. EC2-Classic 인스턴스 간의 통신은 이러한 인스턴스에 연결된 EC2-Classic 보안 그룹에 의해 제어됩니다. 보안 그룹 구성의 예는 [예: 3tier 웹 애플리케이션의 ClassicLink 보안 그룹 구성 \(p. 480\)](#) 섹션을 참조하십시오.

인스턴스를 VPC에 링크한 후에는 인스턴스에 연결된 VPC 보안 그룹을 변경할 수 없습니다. 인스턴스에 다른 보안 그룹을 연결하려면 우선 인스턴스의 링크를 해제한 후 VPC에 다시 링크하면서 필요한 보안 그룹을 선택해야 합니다.

ClassicLink의 라우팅

VPC에서 ClassicLink를 활성화하면 모든 VPC 라우팅 테이블에 대상이 10.0.0.0/8, 타겟이 local인 고정 라우팅이 추가됩니다. 따라서 VPC의 인스턴스와 VPC에 링크된 EC2-Classic 인스턴스 간에 통신이 가능합니다. ClassicLink 가능 VPC에 사용자 지정 라우팅 테이블을 추가하면 대상이 10.0.0.0/8, 타겟이 local인 고정 라우팅이 자동으로 추가됩니다. VPC에서 ClassicLink를 비활성화하면 이 라우팅이 모든 VPC 라우팅 테이블에서 자동으로 삭제됩니다.

10.0.0.0/16 및 10.1.0.0/16 IP 주소 범위에 속하는 VPC에서 ClassicLink를 활성화하려면 라우팅 테이블에 10.0.0.0/8 IP 주소 범위에 속하는 기존 고정 라우팅이 없어야 합니다. VPC 생성 시 자동으로 추가된 로컬 라우팅은 여기에서 제외됩니다. 마찬가지로 VPC에서 ClassicLink를 활성화한 경우 10.0.0.0/8 IP 주소 범위에 속하는 특정 라우팅을 라우팅 테이블에 추가할 수 없습니다.

Important

VPC의 CIDR 블록이 공개적으로 라우팅 가능한 IP 주소 범위인 경우, VPC에 EC2-Classic 인스턴스를 링크하기 전에 보안 문제를 고려해야 합니다. 예를 들어 링크된 EC2-Classic 인스턴스가 VPC의 IP 주소 범위에 속하는 소스 IP 주소로부터 DoS(Denial of Service) 요청 포화 공격을 받는 경우 응답 트래픽이 VPC로 전송됩니다. [RFC 1918](#) 규격에 따라 프라이빗 IP 주소 범위를 사용하여 VPC를 생성하는 것이 좋습니다.

라우팅 테이블 및 VPC의 라우팅에 대한 자세한 내용은 Amazon VPC 사용 설명서에서 [라우팅 테이블](#)을 참조하십시오.

ClassicLink에 대한 VPC 피어링 연결 활성화

두 VPC 간에 VPC 피어링 연결이 있고 ClassicLink를 통해 이 두 VPC 중 하나 또는 둘 다에 연결된 하나 이상의 EC2-Classic 인스턴스가 있는 경우, EC2-Classic 인스턴스와 VPC 피어링 연결의 다른 쪽에 있는 VPC의 인스턴스 간 통신이 활성화되도록 VPC 피어링 연결을 확장할 수 있습니다. 이렇게 하면 EC2-Classic 인스턴스와 VPC의 인스턴스가 프라이빗 IP 주소를 사용하여 통신할 수 있습니다. 이를 위해 로컬 VPC가 피어 VPC의 연결된 EC2-Classic 인스턴스와 통신하도록 하거나, 로컬 연결된 EC2-Classic 인스턴스가 피어 VPC의 인스턴스와 통신하도록 할 수 있습니다.

로컬 VPC가 피어 VPC의 연결된 EC2-Classic 인스턴스와 통신하도록 하면, 목적지가 10.0.0.0/8이고 대상이 local인 라우팅 테이블에 정적 경로가 자동으로 추가됩니다.

자세한 내용과 예시는 Amazon VPC Peering Guide의 [ClassicLink로 구성](#) 섹션을 참조하십시오.

ClassicLink의 제한 사항

ClassicLink 기능을 사용하려면 다음과 같은 제한 사항을 숙지해야 합니다.

- 한 번에 하나의 VPC에만 EC2-Classic 인스턴스를 링크할 수 있습니다.
- 링크된 EC2-Classic 인스턴스를 종지하면 자동으로 VPC에서 링크가 해제되고 VPC 보안 그룹이 인스턴스에 더 이상 연결되지 않습니다. 인스턴스를 다시 시작한 후 VPC에 다시 연결할 수 있습니다.
- 다른 리전이나 다른 AWS 계정에 속하는 VPC에는 EC2-Classic 인스턴스를 링크할 수 없습니다.

- 전용 하드웨어 테넌시로 구성된 VPC에서는 ClassicLink를 활성화할 수 없습니다. 전용 테넌시 VPC에서 ClassicLink를 활성화하려는 경우 AWS Support에 문의하십시오.

Important

EC2-Classic 인스턴스는 공유된 하드웨어에서 실행됩니다. 규정 준수, 보안 강화 등의 요건에 따라 VPC의 테넌시를 dedicated로 설정한 경우 VPC에 EC2-Classic 인스턴스를 링크하면 공유된 테넌시 리소스가 프라이빗 IP 주소를 사용하여 격리된 리소스를 직접 참조할 수 있으므로 이러한 요건에 위배될 수 있습니다. 전용 VPC에서 ClassicLink를 활성화하려는 경우 상세한 이유를 기재하여 AWS Support에 요청하십시오.

- EC2-Classic 프라이빗 IP 주소 범위 10/8과 충돌하는 라우팅이 있는 VPC에서는 ClassicLink를 활성화 할 수 없습니다. 라우팅 테이블에 이미 로컬 라우팅이 있는 10.0.0.0/16 및 10.1.0.0/16 IP 주소 범위의 VPC는 여기에 포함되지 않습니다. 자세한 내용은 [ClassicLink의 라우팅 \(p. 474\)](#) 섹션을 참조하십시오.
- 링크된 EC2-Classic 인스턴스에는 VPC 탄력적 IP 주소를 연결할 수 없습니다.
- 실행 중인 스팟 인스턴스를 VPC에 링크할 수 있습니다. 스팟 인스턴스 요청에서 요청 이행 시 인스턴스를 VPC에 링크해야 함을 명시하려면 Amazon EC2 콘솔에서 시작 마법사를 사용해야 합니다.
- ClassicLink는 VPC 외부의 전이 관계를 지원하지 않습니다. 연결된 EC2-Classic 인스턴스는 VPC에 연결된 VPN 연결, VPC 앤드포인트 또는 인터넷 게이트웨이에 액세스할 수 없습니다. 마찬가지로 VPN 연결의 반대편에 있는 리소스 또는 인터넷 게이트웨이는 연결된 EC2-Classic 인스턴스에 액세스할 수 없습니다.
- ClassicLink를 사용하여 VPC 인스턴스를 다른 VPC 또는 EC2-Classic 리소스에 링크할 수 없습니다. VPC 간에 프라이빗 연결을 설정하려면 VPC 피어링 연결을 사용합니다. 자세한 내용은 [Amazon VPC Peering Guide](#) 항목을 참조하십시오.
- VPC 내에서 172.16.0.23/32 IP 주소로 실행되는 DNS 서버가 있는 경우 EC2-Classic 인스턴스를 172.16.0.0/16 범위의 VPC에 링크하면 링크된 EC2-Classic 인스턴스가 VPC DNS 서버에 액세스할 수 없습니다. 이 문제를 해결하려면 DNS 서버를 VPC 내에서 다른 IP 주소로 실행합니다.

ClassicLink 작업

Amazon EC2 및 Amazon VPC 콘솔을 사용하여 ClassicLink 관련 작업을 수행할 수 있습니다. VPC에서 ClassicLink를 활성화 또는 비활성화하고 VPC에 EC2-Classic 인스턴스를 링크하거나 링크를 해제할 수 있습니다.

Note

ClassicLink 기능은 EC2-Classic을 지원하는 계정 및 리전의 콘솔에만 표시됩니다.

항목

- [VPC에서 ClassicLink 활성화 \(p. 475\)](#)
- [VPC에 인스턴스 링크 \(p. 476\)](#)
- [ClassicLink가 활성화된 VPC 생성 \(p. 476\)](#)
- [EC2-Classic 인스턴스 시작 시 VPC에 링크 \(p. 476\)](#)
- [ClassicLink 가능 VPC 및 링크된 EC2-Classic 인스턴스 보기 \(p. 477\)](#)
- [ClassicLink DNS 지원 활성화 \(p. 477\)](#)
- [ClassicLink DNS 지원 비활성화 \(p. 478\)](#)
- [VPC에서 EC2-Classic 인스턴스 연결 해제 \(p. 478\)](#)
- [VPC에 대해 ClassicLink 비활성화 \(p. 478\)](#)

VPC에서 ClassicLink 활성화

VPC에 EC2-Classic 인스턴스를 링크하려면 우선 VPC에서 ClassicLink를 활성화해야 합니다. VPC에 EC2-Classic 프라이빗 IP 주소 범위와 충돌하는 라우팅이 있는 경우 VPC에서 ClassicLink를 활성화할 수 없습니다. 자세한 내용은 [ClassicLink의 라우팅 \(p. 474\)](#) 섹션을 참조하십시오.

VPC에서 ClassicLink를 활성화하려면 다음을 수행합니다.

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Your VPCs를 선택합니다.
3. VPC를 선택한 다음 [Actions], [Enable ClassicLink]를 선택합니다.
4. 확인 대화 상자에서 [Yes, Enable]을 선택합니다.

VPC에 인스턴스 링크

VPC에서 ClassicLink를 활성화한 후 VPC에 EC2-Classic 인스턴스를 링크할 수 있습니다.

Note

실행 중인 EC2-Classic 인스턴스만 VPC에 링크할 수 있습니다. stopped 상태인 인스턴스는 링크할 수 없습니다.

VPC에 인스턴스를 링크하려면 다음을 수행합니다.

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. 실행 중인 EC2-Classic 인스턴스를 선택한 다음, [Actions], [ClassicLink], [Link to VPC]를 선택합니다. 둘 이상의 인스턴스를 선택하여 동일한 VPC에 링크할 수 있습니다.
4. 표시되는 대화 상자의 목록에서 VPC를 선택합니다. ClassicLink가 활성화된 VPC만 표시됩니다.
5. VPC의 보안 그룹을 하나 이상 선택하여 인스턴스와 연결합니다. 완료되면 [Link to VPC]를 선택합니다.

ClassicLink가 활성화된 VPC 생성

Amazon VPC 콘솔에서 VPC 마법사를 사용하면 새 VPC를 생성할 때 ClassicLink를 즉시 활성화할 수 있습니다.

ClassicLink가 활성화된 VPC를 생성하려면 다음을 수행합니다.

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. Amazon VPC 대시보드에서 [Start VPC Wizard]를 선택합니다.
3. VPC 구성 옵션 중 하나를 선택하고 [Select]를 선택합니다.
4. 마법사 다음 페이지에서 [Enable ClassicLink]에 [Yes]를 선택합니다. 마법사의 나머지 단계를 완료하여 VPC를 생성합니다. VPC 마법사 사용에 대한 자세한 내용은 Amazon VPC 사용 설명서에서 [Amazon VPC용 시나리오](#) 섹션을 참조하십시오.

EC2-Classic 인스턴스 시작 시 VPC에 링크

Amazon EC2 콘솔에서 시작 마법사를 사용하면 EC2-Classic 인스턴스를 시작할 때 ClassicLink 가능 VPC에 즉시 링크할 수 있습니다.

인스턴스 시작 시 VPC에 링크하려면 다음을 수행합니다.

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. Amazon EC2 대시보드에서 [Launch Instance]를 선택합니다.
3. AMI를 선택하고 인스턴스 유형을 선택합니다. [Configure Instance Details] 페이지의 [Network] 목록에서 [Launch into EC2-Classic]을 선택합니다.

Note

T2 인스턴스 유형 등의 일부 인스턴스 유형은 VPC로만 시작할 수 있습니다. EC2-Classic으로 시작할 수 있는 인스턴스 유형을 선택해야 합니다.

4. [Link to VPC (ClassicLink)] 섹션의 [Link to VPC]에서 VPC를 선택합니다. ClassicLink 가능 VPC만 표시됩니다. VPC에서 인스턴스에 연결할 보안 그룹을 선택합니다. 페이지의 다른 구성 옵션을 완료한 후 마법사의 나머지 단계를 완료하여 인스턴스를 시작합니다. 시작 마법사 사용에 대한 자세한 내용은 [AMI에서 인스턴스 시작 \(p. 265\)](#) 섹션을 참조하십시오.

ClassicLink 가능 VPC 및 링크된 EC2-Classic 인스턴스 보기

Amazon VPC 콘솔에서 모든 ClassicLink 가능 VPC를, Amazon EC2 콘솔에서 링크된 EC2-Classic 인스턴스를 확인할 수 있습니다.

ClassicLink 가능 VPC를 확인하려면 다음을 수행합니다.

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Your VPCs를 선택합니다.
3. VPC를 선택하고 [Summary] 탭에서 ClassicLink 필드를 찾습니다. 값이 [Enabled]이면 VPC에서 ClassicLink가 활성화된 것입니다.
4. 또는 ClassicLink 열을 찾고 각 VPC에 표시된 값([Enabled] 또는 [Disabled])을 확인합니다. 해당 열이 보이지 않는 경우 [Edit Table Columns](기어 모양 아이콘)를 선택하고 [ClassicLink] 속성을 선택한 다음 [Close]를 선택합니다.

링크된 EC2-Classic 인스턴스를 확인하려면 다음을 수행합니다.

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. EC2-Classic 인스턴스를 선택하고 [Description] 탭에서 ClassicLink 필드를 찾습니다. 인스턴스가 VPC에 링크된 경우 필드에 인스턴스가 링크된 VPC의 ID가 표시됩니다. 인스턴스가 VPC에 링크되지 않은 경우 필드에 [Unlinked]가 표시됩니다.
4. 또는 인스턴스를 필터링하여 특정 VPC 또는 보안 그룹에 링크된 EC2-Classic 인스턴스만 표시할 수도 있습니다. 검색 창에서 **ClassicLink**를 입력하고 관련 ClassicLink 리소스 속성을 선택한 후 보안 그룹 ID 또는 VPC ID를 선택합니다.

ClassicLink DNS 지원 활성화

연결된 EC2-Classic 인스턴스와 VPC의 인스턴스 사이에서 처리되는 DNS 호스트 이름이 퍼블릭 IP 주소가 아니라 프라이빗 IP 주소로 확인되도록 VPC에 대해 ClassicLink DNS 지원을 활성화할 수 있습니다. 이 기능을 사용하려면 DNS 호스트 이름과 DNS 확인에 대해 VPC가 활성화되어 있어야 합니다.

Note

VPC에 대해 ClassicLink DNS 지원을 활성화하면, 연결된 EC2-Classic 인스턴스는 VPC에 연결된 어떤 프라이빗 호스팅 영역에도 액세스할 수 있습니다. 자세한 내용은 Amazon Route 53 개발자 안내서에서 [프라이빗 호스팅 영역 작업](#) 섹션을 참조하십시오.

ClassicLink DNS 지원을 활성화하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Your VPCs를 선택합니다.
3. VPC를 선택한 다음 [Actions], [Edit ClassicLink DNS Support]를 선택합니다.

- [Yes]를 선택하여 ClassicLink DNS 지원을 활성화한 다음, [Save]를 선택합니다.

ClassicLink DNS 지원 비활성화

연결된 EC2-Classic 인스턴스와 VPC의 인스턴스 사이에서 처리되는 DNS 호스트 이름이 프라이빗 IP 주소가 아니라 퍼블릭 IP 주소로 확인되도록 VPC에 대해 ClassicLink DNS 지원을 비활성화할 수 있습니다.

ClassicLink DNS 지원을 비활성화하려면

- <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
- 탐색 창에서 Your VPCs를 선택합니다.
- VPC를 선택한 다음 [Actions], [Edit ClassicLink DNS Support]를 선택합니다.
- [No]를 선택하여 ClassicLink DNS 지원을 비활성화한 다음, [Save]를 선택합니다.

VPC에서 EC2-Classic 인스턴스 연결 해제

EC2-Classic 인스턴스와 VPC 간에 ClassicLink 연결이 더 이상 필요하지 않은 경우 VPC에서 인스턴스의 링크를 해제할 수 있습니다. 인스턴스의 링크를 해제하면 VPC 보안 그룹이 인스턴스에서 분리됩니다.

Note

인스턴스를 중지하면 자동으로 VPC와 링크가 해제됩니다.

VPC에서 인스턴스의 링크를 해제하려면 다음을 수행합니다.

- <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
- 탐색 창에서 [Instances]를 선택하고 인스턴스를 선택합니다.
- [Actions] 목록에서 [ClassicLink], [Unlink Instance]를 선택합니다. 둘 이상의 인스턴스를 선택하여 동일한 VPC에서 링크를 해제할 수 있습니다.
- 확인 대화 상자에서 [Yes]를 선택합니다.

VPC에 대해 ClassicLink 비활성화

EC2-Classic 인스턴스와 VPC 간에 연결이 더 이상 필요하지 않은 경우 VPC에서 ClassicLink를 비활성화할 수 있습니다. 우선 VPC에 링크된 모든 EC2-Classic 인스턴스의 링크를 해제해야 합니다.

VPC에서 ClassicLink를 비활성화하려면 다음을 수행합니다.

- <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
- 탐색 창에서 Your VPCs를 선택합니다.
- VPC를 선택한 다음, [Actions], [Disable ClassicLink]을 선택합니다.
- 확인 대화 상자에서 [Yes, Disable]을 선택합니다.

API 및 CLI 개요

명령줄 또는 Query API를 사용하여 이 페이지에서 설명하는 작업을 수행할 수 있습니다. 명령줄 인터페이스 및 사용 가능한 API 작업 목록에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

VPC에서 ClassicLink 활성화

- enable-vpc-classic-link(AWS CLI)

- [Enable-EC2VpcClassicLink](#)(Windows PowerShell용 AWS 도구)
- [EnableVpcClassicLink](#)(Amazon EC2 Query API)

EC2-Classic 인스턴스에 VPC 링크(연결)

- [attach-classic-link-vpc](#)(AWS CLI)
- [Add-EC2ClassicLinkVpc](#)(Windows PowerShell용 AWS 도구)
- [AttachClassicLinkVpc](#)(Amazon EC2 Query API)

VPC에서 EC2-Classic 인스턴스 링크 해제(분리)

- [detach-classic-link-vpc](#)(AWS CLI)
- [Dismount-EC2ClassicLinkVpc](#)(Windows PowerShell용 AWS 도구)
- [DetachClassicLinkVpc](#)(Amazon EC2 Query API)

VPC에서 ClassicLink 비활성화

- [disable-vpc-classic-link](#)(AWS CLI)
- [Disable-EC2VpcClassicLink](#)(Windows PowerShell용 AWS 도구)
- [DisableVpcClassicLink](#)(Amazon EC2 Query API)

VPC의 ClassicLink 상태 설명

- [describe-vpc-classic-link](#)(AWS CLI)
- [Get-EC2VpcClassicLink](#)(Windows PowerShell용 AWS 도구)
- [DescribeVpcClassicLink](#)(Amazon EC2 Query API)

링크된 EC2-Classic 인스턴스 설명

- [describe-classic-link-instances](#)(AWS CLI)
- [Get-EC2ClassicLinkInstance](#)(Windows PowerShell용 AWS 도구)
- [DescribeClassicLinkInstances](#)(Amazon EC2 Query API)

ClassicLink에 대한 VPC 피어링 연결 활성화

- [modify-vpc-peering-connection-options](#)(AWS CLI)
- [Edit-EC2VpcPeeringConnectionOption](#)(Windows PowerShell용 AWS 도구)
- [ModifyVpcPeeringConnectionOptions](#)(Amazon EC2 Query API)

ClassicLink DNS 지원에 대해 VPC 활성화

- [enable-vpc-classic-link-dns-support](#)(AWS CLI)
- [Enable-EC2VpcClassicLinkDnsSupport](#)(Windows PowerShell용 AWS 도구)
- [EnableVpcClassicLinkDnsSupport](#)(Amazon EC2 쿼리 API)

ClassicLink DNS 지원에 대해 VPC 비활성화

- [disable-vpc-classic-link-dns-support](#)(AWS CLI)

- [Disable-EC2VpcClassicLinkDnsSupport](#) (Windows PowerShell용 AWS 도구)
- [DisableVpcClassicLinkDnsSupport](#) (Amazon EC2 쿼리 API)

VPC에 대한 ClassicLink DNS 지원 설명

- [describe-vpc-classic-link-dns-support](#) (AWS CLI)
- [Get-EC2VpcClassicLinkDnsSupport](#) (Windows PowerShell용 AWS 도구)
- [DescribeVpcClassicLinkDnsSupport](#) (Amazon EC2 쿼리 API)

예: 3트리어 웹 애플리케이션의 ClassicLink 보안 그룹 구성

이 예에서 애플리케이션은 세 가지 인스턴스, 즉 퍼블릭 웹 서버, 애플리케이션 서버, 그리고 데이터베이스 서버로 구성됩니다. 웹 서버는 인터넷의 HTTPS 트래픽을 수신한 후 TCP 포트 6001을 통해 애플리케이션 서버와 통신합니다. 그런 다음 애플리케이션 서버는 TCP 포트 6004를 통해 데이터베이스 서버와 통신합니다. 현재 사용자는 전체 애플리케이션을 계정 내 VPC로 마이그레이션하는 중입니다. 애플리케이션 서버와 데이터베이스 서버는 이미 VPC로 마이그레이션하였습니다. 웹 서버는 아직 EC2-Classic에 있고 ClassicLink를 통해 VPC로 링크된 상태입니다.

사용자는 이 세 가지 인스턴스에서만 트래픽을 주고받을 수 있도록 보안 그룹을 구성하려고 합니다. 보안 그룹은 웹 서버용 2개([sg-1a1a1a1a](#), [sg-2b2b2b2b](#)), 애플리케이션 서버용 1개([sg-3c3c3c3c](#)), 그리고 데이터베이스 서버용 1개([sg-4d4d4d4d](#))까지 총 4개입니다.

다음은 인스턴스 아키텍처와 보안 그룹 구성을 나타낸 다이어그램입니다.

웹 서버용 보안 그룹([sg-1a1a1a1a](#), [sg-2b2b2b2b](#))

보안 그룹 하나는 EC2-Classic에, 그리고 나머지 하나는 VPC에 있습니다. 그리고 ClassicLink를 통해 인스턴스를 VPC로 링크했을 때 VPC 보안 그룹을 웹 서버 인스턴스와 연동시켰습니다. 이제 웹 서버에서 애플리케이션 서버로 보내지는 아웃바운드 트래픽을 VPC 보안 그룹에서 제어할 수 있습니다.

다음은 EC2-Classic 보안 그룹([sg-1a1a1a1a](#))에 적용되는 보안 그룹 규칙입니다.

인바운드			
소스	Type	포트 범위	설명
0.0.0.0/0	HTTPS	443	인터넷 트래픽의 웹 서버 전송을 허용합니다.

다음은 VPC 보안 그룹([sg-2b2b2b2b](#))에 적용되는 보안 그룹 규칙입니다.

아웃바운드			
목적지	Type	포트 범위	설명
sg-3c3c3c3c	TCP	6001	웹 서버에서 VPC의 애플리케이션 서버(또는 sg-3c3c3c3c 와 연동된 다른 인스턴스)로 전송되는 아웃바운드 트래픽을 허용합니다.

애플리케이션 서버용 보안 그룹([sg-3c3c3c3c](#))

다음은 애플리케이션 서버와 연동되어 있는 VPC 보안 그룹의 보안 그룹 규칙입니다.

인바운드			
------	--	--	--

소스	Type	포트 범위	설명
sg-2b2b2b2b	TCP	6001	웹 서버(또는 sg-2b2b2b2b와 연동되어 있는 기타 인스턴스)에서 애플리케이션 서버로 특정 유형의 트래픽을 전송할 수 있도록 허용합니다.
아웃바운드			
목적지	Type	포트 범위	설명
sg-4d4d4d4d	TCP	6004	애플리케이션 서버에서 데이터베이스 서버(또는 sg-4d4d4d4d와 연동되어 있는 기타 인스턴스)로 전송되는 아웃바운드 트래픽을 허용합니다.

데이터베이스 서버용 보안 그룹([sg-4d4d4d4d](#))

다음은 데이터베이스 서버와 연동되어 있는 VPC 보안 그룹의 보안 그룹 규칙입니다.

인바운드			
소스	Type	포트 범위	설명
sg-3c3c3c3c	TCP	6004	애플리케이션 서버(또는 sg-3c3c3c3c와 연동되어 있는 기타 인스턴스)에서 데이터베이스 서버로 특정 유형의 트래픽을 전송할 수 있도록 허용합니다.

Linux 내 EC2-Classic 인스턴스에서 VPC 내 Linux 인스턴스로 마이그레이션

AWS 계정은 생성 시기와 사용 리전에 따라 EC2-Classic과 EC2-VPC를 모두 지원할 수 있습니다. 자세한 내용과 계정에서 지원하는 플랫폼은 [지원되는 플랫폼 \(p. 471\)](#) 섹션을 참조하십시오. VPC 사용에 따른 이점 및 EC2-Classic과 EC2-VPC의 차이에 대한 자세한 내용은 [Amazon EC2와 Amazon Virtual Private Cloud \(p. 466\)](#) 섹션을 참조하십시오.

AWS 계정에서 리소스를 생성하고 사용합니다. 향상된 네트워킹 및 특정 인스턴스 유형과 같은 일부 리소스와 기능은 VPC에서만 사용할 수 있습니다. 일부 리소스는 EC2-Classic 및 VPC간에 공유될 수 있지만 그렇지 않은 리소스도 있습니다. 자세한 내용은 [EC2-Classic과 EC2-VPC 간 리소스 공유 및 액세스 \(p. 468\)](#) 섹션을 참조하십시오.

EC2-Classic 지원 계정을 사용하는 경우 EC2-Classic에서 사용할 리소스를 설정했을 수 있습니다. EC2-Classic에서 VPC로 마이그레이션하기 위해서는 해당 리소스를 VPC에서 다시 만들어야 합니다.

VPC로 마이그레이션 하는 방법에는 두 가지가 있습니다. 전체 마이그레이션을 수행하거나, 시간을 두고 증분식 마이그레이션을 수행할 수 있습니다. EC2-Classic에 있는 애플리케이션의 크기와 복잡성에 따라 적합한 방법을 선택합니다. 예를 들어, 애플리케이션이 고정 웹 사이트를 실행하는 한두 개의 인스턴스로 구성되고 짧은 기간의 가동 중지를 허용할 수 있는 경우 전체 마이그레이션을 수행할 수 있습니다. 프로세스를 중단할 수 없는 다중 티어 애플리케이션이 있는 경우 ClassicLink를 사용하여 증분식 마이그레이션을 수행할 수 있습니다. 이렇게 하면 애플리케이션이 VPC에서 완전히 실행될 때까지 한 번에 구성 요소 하나씩 기능을 전송할 수 있습니다.

Windows 인스턴스를 마이그레이션해야 하는 경우 Windows 인스턴스용 Amazon EC2 사용 설명서의 [Migrating a Windows Instance from EC2-Classic to a VPC](#) 섹션을 참조하십시오.

목차

- [VPC로 전체 마이그레이션 \(p. 482\)](#)

- [ClassicLink를 사용하여 VPC로 중분식 마이그레이션 \(p. 486\)](#)

VPC로 전체 마이그레이션

다음 작업을 완료하여 EC2-Classic에서 VPC로 애플리케이션을 전체 마이그레이션합니다.

작업

- [1 단계: VPC 생성 \(p. 482\)](#)
- [2단계: 보안 그룹 구성 \(p. 482\)](#)
- [3단계: EC2-Classic 인스턴스에서 AMI 생성 \(p. 483\)](#)
- [4단계: VPC로 인스턴스 시작 \(p. 484\)](#)
- [예제: 간단한 웹 애플리케이션 마이그레이션 \(p. 485\)](#)

1 단계: VPC 생성

VPC 사용을 시작하려면 계정에 VPC가 있는지 확인하십시오. 다음 방법 중 하나를 사용하여 VPC를 생성할 수 있습니다.

- 새로운 EC2-VPC 전용 AWS 계정을 사용합니다. EC2-VPC 전용 계정은 즉시 사용할 수 있는 각 리전의 기본 VPC와 함께 제공됩니다. 다르게 지정하지 않는 한, 시작하는 인스턴스는 기본적으로 이 VPC로 시작됩니다. 기본 VPC에 대한 자세한 내용은 [Your Default VPC and Subnets](#) 섹션을 참조하십시오. VPC를 직접 설정하지 않으려는 경우 또는 VPC 구성에 대해 특정한 요구 사항이 필요 없는 경우 이 옵션을 선택합니다.
- 기존 AWS 계정에서 Amazon VPC 콘솔을 열고 VPC 마법사를 사용하여 새로운 VPC를 생성합니다. 자세한 내용은 [Scenarios for Amazon VPC](#) 섹션을 참조하십시오. 마법사에서 사용 가능한 구성 설정 중 하나를 사용하여 기존 EC2-Classic 계정에서 VPC를 빨리 설정하려는 경우 이 옵션을 선택합니다. 인스턴스를 시작할 때마다 이 VPC를 지정합니다.
- 기존 AWS 계정에서 Amazon VPC 콘솔을 열고 요구 사항에 따라 VPC의 구성 요소를 설정합니다. 자세한 내용은 [Your VPC and Subnets](#)을 참조하십시오. 특정 서브넷 수와 같이 VPC에 대한 특정 요구 사항이 있는 경우 이 옵션을 사용합니다. 인스턴스를 시작할 때마다 이 VPC를 지정합니다.

2단계: 보안 그룹 구성

EC2-Classic과 VPC 간에 동일한 보안 그룹을 사용할 수 없습니다. 그러나 VPC의 인스턴스가 EC2-Classic 인스턴스와 동일한 보안 그룹 규칙을 갖도록 하려는 경우 Amazon EC2 콘솔을 사용하여 기존 EC2-Classic 보안 그룹 규칙을 새 VPC 보안 그룹에 복사할 수 있습니다.

Important

동일한 리전의 동일한 AWS 계정에서만 보안 그룹 규칙을 새 보안 그룹에 복사할 수 있습니다. 새로운 AWS 계정을 생성한 경우에는 이 방법을 사용하여 기존 보안 그룹 규칙을 새 계정에 복사할 수 없습니다. 새 보안 그룹을 생성하고 규칙을 직접 추가해야 합니다. 새 보안 그룹 생성에 대한 자세한 내용은 [Linux 인스턴스에 대한 Amazon EC2 보안 그룹 \(p. 385\)](#) 섹션을 참조하십시오.

새 보안 그룹에 보안 그룹 규칙을 복사하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다.
3. EC2-Classic 인스턴스와 연동되어 있는 보안 그룹을 선택한 다음 [Actions]와 [Copy to new]를 차례대로 선택합니다.
4. [Create Security Group] 대화 상자에서 새 보안 그룹의 이름과 설명을 지정합니다. [VPC] 목록에서 해당 VPC를 선택합니다.
5. [Inbound] 탭이 EC2-Classic 보안 그룹의 규칙으로 채워집니다. 필요에 따라 규칙을 수정할 수 있습니다. [Outbound] 탭에는 모든 아웃바운드 트래픽을 허용하는 규칙이 자동으로 생성되어 있습니다. 보안 그룹

규칙 수정에 대한 자세한 내용은 [Linux 인스턴스에 대한 Amazon EC2 보안 그룹 \(p. 385\)](#) 섹션을 참조하십시오.

Note

EC2-Classic 보안 그룹에서 다른 보안 그룹을 참조하는 규칙을 정의한 경우 VPC 보안 그룹에서는 동일한 규칙을 사용할 수 없습니다. 동일한 VPC의 보안 그룹을 참조하도록 규칙을 수정하십시오.

6. [Create]를 선택합니다.

3단계: EC2-Classic 인스턴스에서 AMI 생성

AMI는 인스턴스를 시작하기 위한 템플릿입니다. 기존 EC2-Classic 인스턴스를 기반으로 고유의 AMI를 생성한 다음 해당 AMI를 사용하여 인스턴스를 VPC로 시작할 수 있습니다.

AMI를 생성하기 위해 사용하는 방법은 인스턴스의 루트 디바이스 유형과 인스턴스가 실행되는 운영 체제 플랫폼에 따라 다릅니다. 인스턴스의 루트 디바이스 유형을 알아내려면 [Instances] 페이지로 이동하고 인스턴스를 선택한 다음 [Description] 탭의 [Root device type] 필드에서 정보를 봅니다. 값이 ebs인 경우 EBS 기반 인스턴스이고, 값이 instance-store인 경우 인스턴스 스토어 기반 인스턴스입니다. [describe-instances](#) AWS CLI 명령을 사용하여 루트 디바이스 유형을 알아낼 수도 있습니다.

다음 표에서는 인스턴스의 루트 디바이스 유형과 소프트웨어 플랫폼을 기반으로 AMI를 생성하는 옵션을 제공합니다.

Important

PV 및 HVM 가상화를 모두 지원하는 인스턴스 유형도 있지만, 둘 중 하나만 지원하는 유형도 있습니다. AMI를 사용하여 현재 인스턴스 유형과 다른 인스턴스 유형을 시작하려는 경우 인스턴스 유형이 AMI에서 제공하는 가상화 유형을 지원하는지 확인하십시오. AMI에서 PV 가상화를 지원하는 경우 HVM 가상화를 지원하는 인스턴스 유형을 사용하려면 기본 HVM AMI에 소프트웨어를 다시 설치해야 할 수 있습니다. PV 및 HVM 가상화에 대한 자세한 내용은 [Linux AMI 가상화 유형 \(p. 66\)](#) 섹션을 참조하십시오.

인스턴스 루트 디바이스 유형	작업
EBS	인스턴스에서 EBS 기반 AMI를 생성합니다. 자세한 내용은 Amazon EBS 지원 Linux AMI 생성 (p. 81) 섹션을 참조하십시오.
인스턴스 스토어	AMI 도구를 사용하여 인스턴스에서 인스턴스 스토어 기반 AMI를 생성합니다. 자세한 내용은 인스턴스 스토어 기반 Linux AMI 생성 (p. 84) 섹션을 참조하십시오.
인스턴스 스토어	인스턴스 데이터를 EBS 볼륨에 전송한 후 볼륨의 스냅샷을 만들고 스냅샷에서 AMI를 생성합니다. 자세한 내용은 인스턴스 스토어 기반 AMI를 Amazon EBS 기반 AMI로 변환 (p. 121) 섹션을 참조하십시오. <p>Note</p> <p>이 방법은 인스턴스 스토어 기반 인스턴스를 EBS 기반 인스턴스로 변환합니다.</p>

(선택 사항) Amazon EBS 볼륨에 데이터 저장

Amazon EBS 볼륨을 생성하고 이 볼륨을 사용하여 물리적 하드 드라이브를 사용할 때와 같이 데이터를 백업하고 인스턴스에 저장할 수 있습니다. 동일한 가용 영역의 모든 인스턴스에서 Amazon EBS 볼륨을 연결하고 분리할 수 있습니다. EC2-Classic의 인스턴스에서 볼륨을 분리하고, 동일한 가용 영역의 VPC로 시작하는 새 인스턴스에 연결할 수 있습니다.

Amazon EBS 볼륨에 대한 자세한 내용은 다음 섹션을 참조하십시오.

- [Amazon EBS 볼륨 \(p. 562\)](#)
- [Amazon EBS 볼륨 생성 \(p. 573\)](#)
- [Amazon EBS 볼륨을 인스턴스에 연결 \(p. 576\)](#)

Amazon EBS 볼륨의 데이터를 백업하려면 볼륨의 정기적 스냅샷을 만듭니다. 필요한 경우 스냅샷에서 Amazon EBS 볼륨을 복원할 수 있습니다. Amazon EBS 스냅샷에 대한 자세한 내용은 다음 섹션을 참조하십시오.

- [Amazon EBS 스냅샷 \(p. 607\)](#)
- [Amazon EBS 스냅샷 생성 \(p. 608\)](#)
- [스냅샷에서 Amazon EBS 볼륨 복구 \(p. 574\)](#)

4단계: VPC로 인스턴스 시작

AMI를 생성한 후 VPC로 인스턴스를 시작할 수 있습니다. 인스턴스는 기존 EC2-Classic 인스턴스와 동일한 데이터 및 구성을 사용합니다.

기존 계정에서 생성한 VPC로 인스턴스를 시작하거나, 새로운 VPC 전용 AWS 계정으로 인스턴스를 시작할 수 있습니다.

기존 EC2-Classic 계정 사용

Amazon EC2 시작 마법사를 사용하여 VPC로 인스턴스를 시작할 수 있습니다.

VPC로 인스턴스를 시작하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 대시보드에서 [Launch Instance]를 선택합니다.
3. [Choose an Amazon Machine Image] 페이지에서 [My AMIs] 범주를 선택하고 생성한 AMI를 선택합니다.
4. [Choose an Instance Type] 페이지에서 인스턴스 유형과 [Next: Configure Instance Details]를 차례대로 선택합니다.
5. [Configure Instance Details] 페이지의 [Network] 목록에서 VPC를 선택합니다. [Subnet] 목록에서 필요한 서브넷을 선택합니다. 기타 필요한 세부 정보를 구성한 다음 [Configure Security Group] 페이지에 도달할 때까지 마법사의 다음 페이지로 이동합니다.
6. [Select an existing group]을 선택하고 이전에 생성한 보안 그룹을 선택합니다. [Review and Launch]를 선택합니다.
7. 인스턴스 정보를 검토한 다음 [Launch]를 선택하여 키 페어를 지정하고 인스턴스를 시작합니다.

마법사의 각 단계에서 구성할 수 있는 파라미터에 대한 자세한 내용은 [인스턴스 시작하기 \(p. 265\)](#) 섹션을 참조하십시오.

새로운 VPC 전용 계정 사용

새로운 AWS 계정에서 인스턴스를 시작하려면 먼저 생성한 AMI를 새 계정과 공유해야 합니다. 그런 다음 Amazon EC2 시작 마법사를 사용하여 기본 VPC로 인스턴스를 시작할 수 있습니다.

AMI를 새 AWS 계정과 공유하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.

2. AMI를 생성한 계정으로 전환합니다.
3. 탐색 창에서 [AMIs]를 선택합니다.
4. [Filter] 목록에서 [Owned by me]가 선택되어 있는지 확인한 다음 AMI를 선택합니다.
5. [Permissions] 탭에서 [Edit]를 선택합니다. 새 AWS 계정의 계정 번호를 입력하고 [Add Permission]과 [Save]를 차례대로 선택합니다.

기본 VPC로 인스턴스를 시작하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 새 AWS 계정으로 전환합니다.
3. 탐색 창에서 [AMIs]를 선택합니다.
4. [Filter] 목록에서 [Private images]를 선택합니다. EC2-Classic 계정에서 공유한 AMI와 [Launch]를 차례대로 선택합니다.
5. [Choose an Instance Type] 페이지에서 인스턴스 유형과 [Next: Configure Instance Details]를 차례대로 선택합니다.
6. [Configure Instance Details] 페이지의 [Network] 목록에서 기본 VPC를 선택해야 합니다. 기타 필요한 세부 정보를 구성한 다음 [Configure Security Group] 페이지에 도달할 때까지 마법사의 다음 페이지로 이동합니다.
7. [Select an existing group]을 선택하고 이전에 생성한 보안 그룹을 선택합니다. [Review and Launch]를 선택합니다.
8. 인스턴스 정보를 검토한 다음 [Launch]를 선택하여 키 페어를 지정하고 인스턴스를 시작합니다.

마법사의 각 단계에서 구성할 수 있는 파라미터에 대한 자세한 내용은 [인스턴스 시작하기 \(p. 265\)](#) 섹션을 참조하십시오.

예제: 간단한 웹 애플리케이션 마이그레이션

이 예제에서는 AWS를 사용하여 원예 웹 사이트를 호스팅합니다. 웹 사이트를 관리하기 위해 EC2-Classic에서 세 개의 인스턴스를 실행하고 있습니다. 인스턴스 A와 B는 퍼블릭 웹 애플리케이션을 호스팅하며 엘라스틱 로드 밸런서를 사용하여 이 두 인스턴스 간 트래픽의 로드 밸런스를 유지합니다. 탄력적 IP 주소를 인스턴스 A와 B에 배정하여 해당 인스턴스에 구성 및 관리 작업을 위한 고정 IP 주소를 만들었습니다. 인스턴스 C에는 웹 사이트를 위한 MySQL 데이터베이스가 저장되어 있습니다. 도메인 이름 www.garden.example.com을 등록하고 Amazon Route 53를 사용하여 로드 밸런서의 DNS 이름과 연결된 별칭 레코드 세트를 포함하는 호스팅 영역을 생성했습니다.

VPC로 마이그레이션하는 첫 단계는 어떤 종류의 VPC 아키텍처가 필요에 맞는지 결정하는 것입니다. 이 경우 웹 서버용 퍼블릭 서브넷 하나와 데이터베이스 서버용 프라이빗 서브넷 하나를 결정했습니다. 웹 사이트가 커지면 더 많은 웹 서버와 데이터베이스 서버를 서브넷에 추가할 수 있습니다. 기본적으로 프라이빗 서브넷의 인스턴스는 인터넷에 액세스할 수 없지만, 퍼블릭 서브넷의 NAT(Network Address Translation) 디바이스를 통해 인터넷 액세스를 활성화할 수 있습니다. 인터넷에서 데이터베이스 서버에 대한 정기 업데이트 및 패치를 지원하도록 NAT 디바이스를 설정해야 할 수 있습니다. 탄력적 IP 주소를 EC2-VPC로 마이그레이션하고 퍼블릭 서브넷에서 Elastic Load Balancer를 생성하여 웹 서버 간 트래픽의 로드 밸런스를 유지합니다.

VPC로 웹 애플리케이션을 마이그레이션하려면 다음 단계를 따릅니다.

- VPC 생성: 이 경우 Amazon VPC 콘솔의 VPC 마법사를 사용하여 VPC와 서브넷을 생성할 수 있습니다. 두 번째 마법사 구성은 프라이빗 서브넷 하나와 퍼블릭 서브넷 하나가 있는 VPC를 생성하고, 퍼블릭 서브넷에서 NAT 디바이스를 시작하고 구성합니다. 자세한 내용은 Amazon VPC 사용 설명서의 [Scenario 2: VPC with Public and Private Subnets](#) 섹션을 참조하십시오.
- 인스턴스에서 AMI 생성: 웹 서버 중 하나에서 AMI를 생성하고 데이터베이스 서버에서 두 번째 AMI를 생성합니다. 자세한 내용은 [3단계: EC2-Classic 인스턴스에서 AMI 생성 \(p. 483\)](#) 섹션을 참조하십시오.

- 보안 그룹 구성: EC2-Classic 환경에서는 웹 서버에 대한 보안 그룹이 하나 있고 데이터베이스 서버에 대한 다른 보안 그룹이 하나 있습니다. Amazon EC2 콘솔을 사용하여 각 보안 그룹에서 VPC의 새 보안 그룹으로 규칙을 복사할 수 있습니다. 자세한 내용은 [2단계: 보안 그룹 구성 \(p. 482\)](#) 섹션을 참조하십시오.

Tip

다른 보안 그룹에서 참조되는 보안 그룹을 먼저 생성하십시오.

- 새로운 VPC로 인스턴스 시작: 퍼블릭 서브넷으로 대체 웹 서버를 시작하고, 프라이빗 서브넷으로 대체 데이터베이스 서버를 시작합니다. 자세한 내용은 [4단계: VPC로 인스턴스 시작 \(p. 484\)](#) 섹션을 참조하십시오.
- NAT 디바이스 구성: NAT 인스턴스를 사용 중인 경우 NAT 인스턴스에 대해 프라이빗 서브넷에서 오는 HTTP 및 HTTPS 트래픽을 허용하는 보안 그룹을 만들어야 합니다. 자세한 내용은 [NAT Instances](#) 섹션을 참조하십시오. NAT 게이트웨이를 사용 중인 경우 프라이빗 서브넷에서 오는 트래픽이 자동으로 허용됩니다.
- 데이터베이스 구성: EC2-Classic의 데이터베이스 서버에서 AMI를 생성한 경우 해당 인스턴스에 저장된 모든 구성 정보가 AMI로 복사됩니다. 새로운 데이터베이스 서버에 연결하고 구성 세부 정보를 업데이트해야 할 수 있습니다. 예를 들어, EC2-Classic에서 웹 서버에 전체 읽기, 쓰기 및 수정 권한을 부여하도록 데이터베이스를 구성한 경우 새로운 VPC 웹 서버에 동일한 권한을 대신 부여하도록 구성 파일을 업데이트해야 합니다.
- 웹 서버 구성: 웹 서버는 EC2-Classic 인스턴스와 동일한 구성 설정을 사용합니다. 예를 들어, EC2-Classic에서 데이터베이스를 사용하도록 웹 서버를 구성한 경우 새로운 데이터베이스 인스턴스를 가리키도록 웹 서버의 구성 설정을 업데이트합니다.

Note

시작 시 다르게 지정하지 않는 한, 기본이 아닌 서브넷으로 시작한 인스턴스에는 퍼블릭 IP 주소가 기본적으로 배정되지 않습니다. 새 데이터베이스 서버에는 퍼블릭 IP 주소가 없을 수 있습니다. 이 경우 새로운 데이터베이스 서버의 프라이빗 DNS 이름을 사용하도록 웹 서버의 구성 파일을 업데이트할 수 있습니다. 동일한 VPC에 있는 인스턴스는 프라이빗 IP 주소를 통해 서로 통신할 수 있습니다.

- 탄력적 IP 주소 마이그레이션: 탄력적 IP 주소를 EC2-Classic의 웹 서버에서 해제한 후 EC2-VPC로 마이그레이션합니다. 마이그레이션이 완료되면 탄력적 IP 주소를 VPC의 새로운 웹 서버와 연동시킵니다. 자세한 내용은 [EC2-Classic에서 EC2-VPC로 탄력적 IP 주소의 마이그레이션 \(p. 507\)](#) 섹션을 참조하십시오.
- 새로운 로드 밸런서 생성: 계속 Elastic Load Balancing을 사용하여 인스턴스에 대한 트래픽의 로드 밸런스를 유지하려면 VPC에서 로드 밸런서를 구성할 수 있는 다양한 방법을 이해해야 합니다. 자세한 내용은 [Elastic Load Balancing in Amazon VPC](#) 섹션을 참조하십시오.
- DNS 레코드 업데이트: 퍼블릭 서브넷에서 로드 밸런서를 설정한 후에는 `www.garden.example.com` 도메인이 새로운 로드 밸런서를 가리키는지 확인해야 합니다. 이렇게 하려면 Amazon Route 53에서 DNS 레코드를 업데이트하고 별칭 레코드 세트를 업데이트해야 합니다. Amazon Route 53 사용에 대한 자세한 내용은 [Amazon Route 53 시작하기](#) 섹션을 참조하십시오.
- EC2-Classic 리소스 종료: 웹 애플리케이션이 VPC 아키텍처 내에서 작동하고 있는지 확인한 후 EC2-Classic 리소스를 종료하여 해당 요금이 발생하지 않도록 할 수 있습니다. EC2-Classic 인스턴스를 종료하고 EC2-Classic 탄력적 IP 주소를 릴리스하십시오.

ClassicLink를 사용하여 VPC로 증분식 마이그레이션

ClassicLink 기능을 사용하면 VPC로 증분식 마이그레이션 작업을 더 쉽게 관리할 수 있습니다. ClassicLink에서는 새 VPC 리소스가 프라이빗 IPv4 주소를 사용하여 EC2-Classic 인스턴스와 통신할 수 있도록 EC2-Classic 인스턴스를 동일 리전의 계정에 있는 VPC에 연결할 수 있습니다. 그런 다음 한 번에 한 단계씩 기능을 VPC로 마이그레이션할 수 있습니다. 이 주제에서는 EC2-Classic에서 VPC로 증분식 마이그레이션을 관리하기 위한 몇 가지 기본 단계를 제공하고 .

ClassicLink에 대한 자세한 내용은 [ClassicLink \(p. 472\)](#) 섹션을 참조하십시오.

항목

- 1단계: 마이그레이션 시퀀스 준비 (p. 487)
- 2 단계: VPC 생성 (p. 487)
- 3단계: ClassicLink에 대해 VPC 활성화 (p. 487)
- 4단계: EC2-Classic 인스턴스에서 AMI 생성 (p. 487)
- 5단계: VPC로 인스턴스 시작 (p. 488)
- 6단계: VPC에 EC2-Classic 인스턴스 연결 (p. 489)
- 7단계: VPC 마이그레이션 완료 (p. 489)

1단계: 마이그레이션 시퀀스 준비

ClassicLink를 효과적으로 사용하려면 먼저 VPC로 마이그레이션해야 하는 애플리케이션 구성 요소를 식별한 다음 해당 기능을 마이그레이션하는 순서를 확인해야 합니다.

예를 들어, 프레젠테이션 웹 서버, 백 엔드 데이터베이스 서버 및 거래용 인증 로직을 이용하는 애플리케이션이 있는 경우 인증 로직으로 마이그레이션 프로세스를 시작한 다음 데이터베이스 서버를 마이그레이션하고 마지막으로 웹 서버를 마이그레이션할 수 있습니다.

2 단계: VPC 생성

VPC 사용을 시작하려면 계정에 VPC가 있는지 확인하십시오. 다음 방법 중 하나를 사용하여 VPC를 생성할 수 있습니다.

- 기존 AWS 계정에서 Amazon VPC 콘솔을 열고 VPC 마법사를 사용하여 새로운 VPC를 생성합니다. 자세한 내용은 [Scenarios for Amazon VPC](#) 섹션을 참조하십시오. 마법사에서 사용 가능한 구성 설정 중 하나를 사용하여 기존 EC2-Classic 계정에서 VPC를 빨리 설정하려는 경우 이 옵션을 선택합니다. 인스턴스를 시작할 때마다 이 VPC를 지정합니다.
- 기존 AWS 계정에서 Amazon VPC 콘솔을 열고 요구 사항에 따라 VPC의 구성 요소를 설정합니다. 자세한 내용은 [Your VPC and Subnets](#)을 참조하십시오. 특정 서브넷 수와 같이 VPC에 대한 특정 요구 사항이 있는 경우 이 옵션을 사용합니다. 인스턴스를 시작할 때마다 이 VPC를 지정합니다.

3단계: ClassicLink에 대해 VPC 활성화

VPC를 생성한 후 ClassicLink에 대해 활성화할 수 있습니다. ClassicLink에 대한 자세한 내용은 [ClassicLink \(p. 472\)](#) 섹션을 참조하십시오.

ClassicLink에 대해 VPC를 활성화하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Your VPCs]를 선택합니다.
3. VPC를 선택한 다음 [Actions] 목록에서 [Enable ClassicLink]을 선택합니다.
4. 확인 대화 상자에서 [Yes, Enable]을 선택합니다.

4단계: EC2-Classic 인스턴스에서 AMI 생성

AMI는 인스턴스를 시작하기 위한 템플릿입니다. 기존 EC2-Classic 인스턴스를 기반으로 고유의 AMI를 생성한 다음 해당 AMI를 사용하여 인스턴스를 VPC로 시작할 수 있습니다.

AMI를 생성하기 위해 사용하는 방법은 인스턴스의 루트 디바이스 유형과 인스턴스가 실행되는 운영 체제 플랫폼에 따라 다릅니다. 인스턴스의 루트 디바이스 유형을 알아내려면 [Instances] 페이지로 이동하고 인스턴스를 선택한 다음 [Description] 탭의 [Root device type] 필드에서 정보를 봅니다. 값이 ebs인 경우 EBS 기반 인스턴스이고, 값이 instance-store인 경우 인스턴스 스토어 기반 인스턴스입니다. [describe-instances](#) AWS CLI 명령을 사용하여 루트 디바이스 유형을 알아낼 수도 있습니다.

다음 표에서는 인스턴스의 루트 디바이스 유형과 소프트웨어 플랫폼을 기반으로 AMI를 생성하는 옵션을 제공합니다.

Important

PV 및 HVM 가상화를 모두 지원하는 인스턴스 유형도 있지만, 그 중 하나만 지원하는 유형도 있습니다. AMI를 사용하여 현재 인스턴스 유형과 다른 인스턴스 유형을 시작하려는 경우 인스턴스 유형이 AMI에서 제공하는 가상화 유형을 지원하는지 확인하십시오. AMI에서 PV 가상화를 지원하는 경우 HVM 가상화를 지원하는 인스턴스 유형을 사용하려면 기본 HVM AMI에 소프트웨어를 다시 설치해야 할 수 있습니다. PV 및 HVM 가상화에 대한 자세한 내용은 [Linux AMI 가상화 유형 \(p. 66\)](#) 섹션을 참조하십시오.

인스턴스 루트 디바이스 유형	작업
EBS	인스턴스에서 EBS 기반 AMI를 생성합니다. 자세한 내용은 Amazon EBS 지원 Linux AMI 생성 (p. 81) 섹션을 참조하십시오.
인스턴스 스토어	AMI 도구를 사용하여 인스턴스에서 인스턴스 스토어 기반 AMI를 생성합니다. 자세한 내용은 인스턴스 스토어 기반 Linux AMI 생성 (p. 84) 섹션을 참조하십시오.
인스턴스 스토어	인스턴스 데이터를 EBS 볼륨에 전송한 후 볼륨의 스냅샷을 만들고 스냅샷에서 AMI를 생성합니다. 자세한 내용은 인스턴스 스토어 기반 AMI를 Amazon EBS 기반 AMI로 변환 (p. 121) 섹션을 참조하십시오. Note 이 방법은 인스턴스 스토어 기반 인스턴스를 EBS 기반 인스턴스로 변환합니다.

(선택 사항) Amazon EBS 볼륨에 데이터 저장

Amazon EBS 볼륨을 생성하고 이 볼륨을 사용하여 물리적 하드 드라이브를 사용할 때와 같이 데이터를 백업하고 인스턴스에 저장할 수 있습니다. 동일한 가용 영역의 모든 인스턴스에서 Amazon EBS 볼륨을 연결하고 분리할 수 있습니다. EC2-Classic의 인스턴스에서 볼륨을 분리하고, 동일한 가용 영역의 VPC로 시작하는 새 인스턴스에 연결할 수 있습니다.

Amazon EBS 볼륨에 대한 자세한 내용은 다음 섹션을 참조하십시오.

- [Amazon EBS 볼륨 \(p. 562\)](#)
- [Amazon EBS 볼륨 생성 \(p. 573\)](#)
- [Amazon EBS 볼륨을 인스턴스에 연결 \(p. 576\)](#)

Amazon EBS 볼륨의 데이터를 백업하려면 볼륨의 정기적 스냅샷을 만듭니다. 필요한 경우 스냅샷에서 Amazon EBS 볼륨을 복원할 수 있습니다. Amazon EBS 스냅샷에 대한 자세한 내용은 다음 섹션을 참조하십시오.

- [Amazon EBS 스냅샷 \(p. 607\)](#)
- [Amazon EBS 스냅샷 생성 \(p. 608\)](#)
- [스냅샷에서 Amazon EBS 볼륨 복구 \(p. 574\)](#)

5단계: VPC로 인스턴스 시작

マイグレーション 프로세스의 다음 단계는 기능 전송을 시작할 수 있도록 VPC로 인스턴스를 시작하는 것입니다. 이전 단계에서 생성한 AMI를 사용하여 VPC로 인스턴스를 시작할 수 있습니다. 인스턴스는 기존 EC2-Classic 인스턴스와 동일한 데이터 및 구성을 사용합니다.

사용자 지정 AMI를 사용하여 VPC로 인스턴스를 시작하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 대시보드에서 [Launch Instance]를 선택합니다.
3. [Choose an Amazon Machine Image] 페이지에서 [My AMIs] 범주를 선택하고 생성한 AMI를 선택합니다.
4. [Choose an Instance Type] 페이지에서 인스턴스 유형과 [Next: Configure Instance Details]를 차례대로 선택합니다.
5. [Configure Instance Details] 페이지의 [Network] 목록에서 VPC를 선택합니다. [Subnet] 목록에서 필요한 서브넷을 선택합니다. 기타 필요한 세부 정보를 구성한 다음 [Configure Security Group] 페이지에 도달할 때까지 마법사의 다음 페이지로 이동합니다.
6. [Select an existing group]을 선택하고 이전에 생성한 보안 그룹을 선택합니다. [Review and Launch]를 선택합니다.
7. 인스턴스 정보를 검토한 다음 [Launch]를 선택하여 키 페어를 지정하고 인스턴스를 시작합니다.

마법사의 각 단계에서 구성할 수 있는 파라미터에 대한 자세한 내용은 [인스턴스 시작하기 \(p. 265\)](#) 섹션을 참조하십시오.

인스턴스를 시작한 후 인스턴스가 `running` 상태이면 인스턴스에 연결하고 필요에 따라 구성할 수 있습니다.

6단계: VPC에 EC2-Classic 인스턴스 연결

인스턴스를 구성하고 애플리케이션 기능을 VPC에서 사용할 수 있게 만든 후에는 ClassicLink를 사용하여 새로운 VPC 인스턴스와 EC2-Classic 인스턴스 간에 프라이빗 IP 통신을 활성화할 수 있습니다.

VPC에 인스턴스를 연결하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Instances]를 선택합니다.
3. EC2-Classic 인스턴스를 선택한 다음 [Actions], [ClassicLink] 및 [Link to VPC]를 차례대로 선택합니다.

Note

인스턴스가 `running` 상태인지 확인합니다.

4. 대화 상자에서 ClassicLink 지원 VPC를 선택합니다(CclassicLink에 대해 활성화된 VPC만 표시됨).
5. VPC의 보안 그룹을 하나 이상 선택하여 인스턴스와 연결합니다. 완료되면 [Link to VPC]를 선택합니다.

7단계: VPC 마이그레이션 완료

애플리케이션의 크기와 마이그레이션해야 할 기능에 따라 4~6단계를 반복하여 애플리케이션의 모든 구성 요소를 EC2-Classic에서 VPC로 이동합니다.

EC2-Classic 및 VPC 인스턴스 간에 내부 통신을 활성화한 경우 EC2-Classic 플랫폼의 서비스 대신 VPC에 있는 마이그레이션된 서비스를 가리키도록 애플리케이션을 업데이트해야 합니다. 이 작업을 위한 정확한 단계는 애플리케이션 설계에 따라 다릅니다. 일반적으로 이 작업에는 EC2-Classic 인스턴스 대신 VPC 인스턴스의 IP 주소를 가리키도록 대상 IP 주소를 업데이트하는 단계가 포함됩니다. 현재 EC2-Classic 플랫폼에서 사용 중인 탄력적 IP 주소를 EC2-VPC 플랫폼으로 마이그레이션할 수 있습니다. 자세한 내용은 [EC2-Classic에서 EC2-VPC로 탄력적 IP 주소의 마이그레이션 \(p. 507\)](#) 섹션을 참조하십시오.

이 단계를 완료하고 애플리케이션이 VPC에서 작동하는지 테스트한 후 EC2-Classic 인스턴스를 종료하고 VPC에 대해 ClassicLink를 비활성화할 수 있습니다. 또한 비용이 발생하지 않도록 더 이상 필요하지 않은 EC2-Classic 리소스를 정리할 수 있습니다. 예를 들어, 탄력적 IP 주소를 릴리스하고 EC2-Classic 인스턴스와 연결된 볼륨을 삭제할 수 있습니다.

Amazon EC2인스턴스 IP 어드레싱

인스턴스에는 IP 주소 및 IPv4 DNS 호스트 이름이 함께 제공됩니다. IP 주소 및 DNS 호스트 이름은 EC2-Classic 플랫폼 또는 가상 프라이빗 클라우드(VPC) 중 어디에서 인스턴스가 시작되었는지에 따라 다릅니다. EC2-Classic 및 EC2-VPC 플랫폼에 대한 자세한 내용은 [지원되는 플랫폼 \(p. 471\)](#) 섹션을 참조하십시오.

Amazon EC2와 Amazon VPC는 IPv4 및 IPv6 주소 지정 프로토콜을 모두 지원합니다. Amazon EC2와 Amazon VPC는 IPv4 주소 지정 프로토콜을 사용하도록 기본 설정되어 있으며 이 동작은 비활성화할 수 없습니다. VPC를 생성할 때 VPC에 IPv4 CIDR 블록(프라이빗 IPv4 주소)을 지정해야 합니다. IPv6 CIDR 블록을 VPC와 서브넷에 할당하고 그 블록에 속한 IPv6 주소를 서브넷의 인스턴스에 할당할 수도 있습니다. IPv6 주소는 인터넷으로 접속할 수 있습니다. IPv6에 대한 자세한 내용은 Amazon VPC 사용 설명서의 [VPC에서 IP 주소 지정](#)을 참조하십시오.

IPv6는 EC2-Classic 플랫폼에 지원되지 않습니다.

목차

- [프라이빗 IPv4 주소 및 내부 DNS 호스트 이름 \(p. 490\)](#)
- [퍼블릭 IPv4 주소 및 외부 DNS 호스트 이름 \(p. 491\)](#)
- [탄력적 IP 주소 \(IPv4\) \(p. 492\)](#)
- [Amazon DNS 서버 \(p. 492\)](#)
- [IPv6 주소 \(p. 492\)](#)
- [EC2-Classic과 EC2-VPC의 IP 주소 차이점 \(p. 493\)](#)
- [인스턴스에 대한 IP 주소 작업 \(p. 493\)](#)
- [다중 IP 주소 \(p. 498\)](#)

프라이빗 IPv4 주소 및 내부 DNS 호스트 이름

프라이빗 IPv4 주소는 인터넷을 통해 연결할 수 없는 IP 주소입니다. 프라이빗 IPv4 주소는 동일 네트워크에서 인스턴스 간의 통신을 위해 사용될 수 있습니다(EC2-Classic 또는 VPC). 프라이빗 IPv4 주소의 표준 및 사양에 대한 자세한 내용은 [RFC 1918](#) 섹션을 참조하십시오.

Note

RFC 1918에 지정된 프라이빗 IPv4 주소 범위에 속하지 않는 공개적으로 라우팅 가능한 CIDR 블록을 사용하여 VPC를 생성할 수 있습니다. 하지만 이 설명서에서 프라이빗 IPv4 주소(또는 프라이빗 IP 주소)는 VPC의 IPv4 CIDR 범위 내에 있는 IP 주소를 말합니다.

인스턴스를 시작할 때 DHCP를 사용하면 인스턴스에 프라이빗 IPv4 주소가 할당됩니다. 또한, 각 인스턴스에는 인스턴스의 프라이빗 IPv4 주소를 확인하는 내부 DNS 호스트 이름이 할당됩니다. 예: `ip-10-251-50-12.ec2.internal` 내부 DNS 호스트 이름은 동일 네트워크에서 인스턴스 간의 통신을 위해 사용될 수 있지만 인스턴스가 위치한 네트워크 외부의 DNS 호스트 이름은 확인할 수 없습니다.

VPC에서 시작한 인스턴스에는 서브넷 IPv4 주소 범위 내의 기본 프라이빗 IP 주소가 할당됩니다. 자세한 내용은 Amazon VPC 사용 설명서의 [서브넷 크기 조정](#)을 참조하십시오. 인스턴스 시작 시 사용자가 기본 프라이빗 IP 주소를 지정하지 않으면 사용자 서브넷 IPv4 범위 내의 IP 주소가 할당됩니다. VPC의 각 인스턴스는 기본 프라이빗 IPv4 주소가 할당된 기본 네트워크 인터페이스(eth0)를 갖습니다. 또한, 사용자는 보조 프라이빗 IPv4 주소라는 추가 프라이빗 IPv4 주소를 지정할 수 있습니다. 기본 프라이빗 IP 주소와 달리, 보조 프라이빗 IP 주소는 한 인스턴스에서 다른 인스턴스로 재할당될 수 있습니다. 자세한 내용은 [다중 IP 주소 \(p. 498\)](#) 섹션을 참조하십시오.

EC2-Classic에서 인스턴스가 시작된 경우 인스턴스가 종지 또는 종료되면 Amazon은 프라이빗 IPv4 주소를 해제합니다. 종지된 인스턴스를 다시 시작하면 새 프라이빗 IPv4 주소가 할당됩니다.

VPC에서 시작된 인스턴스의 경우 인스턴스가 중지 및 재시작될 때 프라이빗 IPv4 주소는 네트워크 인터페이스와 계속해서 연동되고 인스턴스가 종료되면 연동이 해제됩니다.

EC2-Classic에서 사용자 지정 방화벽 구성은 생성하는 경우 Amazon DNS 서버의 주소에서 —한시적인 범위의 대상 포트와 함께— 포트 53(DNS)으로부터의 인바운드 트래픽을 허용하는 규칙을 방화벽에 생성해야 합니다. 그렇지 않으면 인스턴스에서 내부 DNS 확인이 실패합니다. 방화벽에서 DNS 쿼리 응답이 자동 허용되지 않는 경우 Amazon DNS 서버의 IP 주소에서 전송되는 트래픽을 허용하도록 설정해야 합니다. Amazon DNS 서버의 IP 주소를 획득하려면 사용자 인스턴스에서 다음 명령을 사용합니다.

- Linux

```
grep nameserver /etc/resolv.conf
```

퍼블릭 IPv4 주소 및 외부 DNS 호스트 이름

퍼블릭 IP 주소는 인터넷을 통해 연결할 수 있는 IPv4 주소입니다. 퍼블릭 주소는 인스턴스와 인터넷의 상호 통신을 위해 사용될 수 있습니다.

또한, 퍼블릭 IP 주소가 할당된 각 인스턴스에는 외부 DNS 호스트이름이 할당됩니다. 예: ec2-203-0-113-25.compute-1.amazonaws.com. Amazon은 외부 DNS 호스트 이름을 인스턴스 네트워크 외부 인스턴스의 퍼블릭 IP 주소로 변환하고 인스턴스 네트워크 내부인 경우 프라이빗 IPv4 주소로 변환합니다. 퍼블릭 IP 주소는 네트워크 주소 변환(NAT)을 통해 기본 프라이빗 IP 주소로 매핑됩니다. NAT에 대한 자세한 내용은 [RFC 1631: IP 네트워크 주소 변환기\(NAT\)](#) 섹션을 참조하십시오.

EC2-Classic에서 인스턴스를 시작하면 자동으로 EC2-Classic 퍼블릭 IPv4 주소 풀에서 퍼블릭 IP 주소 하나가 인스턴스로 할당됩니다. 사용자는 이 동작을 수정할 수 없습니다. VPC로 인스턴스를 시작하는 경우 서브넷은 이 서브넷으로 시작되는 인스턴스가 EC2-VPC 퍼블릭 IPv4 주소 풀로부터 퍼블릭 IP 주소를 부여받는지 여부를 결정하는 속성을 갖습니다. 기본적으로, 기본 VPC로 시작되는 인스턴스에는 퍼블릭 IP 주소를 할당하지만 기본이 아닌 서브넷으로 시작되는 인스턴스에는 퍼블릭 IP 주소를 할당하지 않습니다.

사용자는 다음을 수행하여 VPC 인스턴스에 퍼블릭 IP 주소가 할당되는지를 제어할 수 있습니다.

- 서브넷의 퍼블릭 IP 주소 지정 속성 수정. 자세한 내용은 Amazon VPC 사용 설명서의 [서브넷의 퍼블릭 IPv4 주소 지정 속성 수정](#)을 참조하십시오.
- 시작 시 퍼블릭 IP 주소 지정 기능을 활성화 또는 비활성화(서브넷의 퍼블릭 IP 주소 지정 속성 재정의). 자세한 내용은 [인스턴스 시작 시 퍼블릭 IPv4 주소 배정 \(p. 495\)](#) 섹션을 참조하십시오.

퍼블릭 IP 주소는 Amazon의 퍼블릭 IPv4 주소 풀에서 사용자 인스턴스로 지정되고 AWS 계정과는 관련이 없습니다. 인스턴스와 퍼블릭 IP 주소의 연결이 해제되면 해당 퍼블릭 IP 주소는 퍼블릭 IPv4 주소 풀로 해제되지만 사용자가 해당 주소를 다시 사용할 수 없습니다.

사용자는 인스턴스에서 퍼블릭 IP 주소를 수동으로 연결 또는 해제할 수 없습니다. 대신, 대부분의 경우 Amazon이 사용자 인스턴스에서 퍼블릭 IP 주소를 해제하거나 해당 IP를 새 인스턴스에 할당합니다.

- 인스턴스가 중지 또는 종료되면 인스턴스의 퍼블릭 IP 주소는 해제됩니다. 중지된 인스턴스가 다시 시작되면 새 퍼블릭 IP 주소가 할당됩니다.
- 인스턴스와 탄력적 IP 주소가 연결되거나 VPC 인스턴스의 기본 네트워크 인터페이스(eth0)와 탄력적 IP 주소가 연결되면 인스턴스의 퍼블릭 IP 주소가 해제됩니다. 사용자가 인스턴스에서 탄력적 IP 주소의 연결을 해제하면 새 퍼블릭 IP 주소가 할당됩니다.
- VPC 인스턴스의 퍼블릭 IP 주소가 해제되고 인스턴스에 1개 이상의 네트워크 인터페이스가 연결된 경우 새 퍼블릭 IP 주소가 할당되지 않습니다.

필요에 따라 인스턴스 간에 연결할 수 있는 영구 퍼블릭 IP 주소가 필요한 경우 탄력적 IP 주소를 대신하여 사용합니다. 예를 들어 동적 DNS를 사용하여 새 인스턴스의 퍼블릭 IP 주소에 기존 DNS 이름을 연결하는 경우

IP 주소가 인터넷을 통해 전해지는 데 24시간까지 걸릴 수 있습니다. 따라서 종료된 인스턴스가 요청을 계속 받는 동안 새 인스턴스가 트래픽을 받지 못할 수 있습니다. 이 문제를 해결하려면 탄력적 IP 주소를 사용합니다. 사용자는 고유 탄력적 IP 주소를 할당하고 인스턴스와 연결할 수 있습니다. 자세한 내용은 [탄력적 IP 주소 \(p. 505\)](#) 섹션을 참조하십시오.

인스턴스가 VPC 상태이고 탄력적 IP 주소가 할당된 경우 IPv4 DNA 호스트 이름이 활성화되어 있으면 인스턴스에 DNS 호스트 이름이 할당됩니다. 자세한 내용은 Amazon VPC 사용 설명서의 [VPC에서 DNS 사용](#)을 참조하십시오.

Note

인스턴스가 동일 리전에 존재하는지의 여부에 따라 퍼블릭 NAT IP 주소를 통해 다른 인스턴스에 액세스하는 인스턴스에는 리전별 또는 인터넷 데이터 전송 비용이 청구됩니다.

탄력적 IP 주소 (IPv4)

탄력적 IP 주소는 사용자가 계정에 연결할 수 있는 퍼블릭 IPv4 주소입니다. 사용자는 필요에 따라 인스턴스 간에 연결할 수 있고 해제되기 전까지는 사용자 계정에 할당됩니다. 탄력적 IP 주소 및 사용 방법에 대한 자세한 내용은 [탄력적 IP 주소 \(p. 505\)](#) 섹션을 참조하십시오.

IPv6에 대한 탄력적 IP 주소는 지원하지 않습니다.

Amazon DNS 서버

Amazon은 Amazon이 제공한 IPv4 DNS 호스트 이름을 IPv4 주소로 변환하는 DNS 서버를 제공합니다. EC2-Classic의 경우 Amazon DNS 서버는 172.16.0.23에 위치합니다. EC2-VPC의 경우 Amazon DNS 서버는 사용자 VPC 네트워크 범위 +2의 범위에 위치합니다. 자세한 내용은 Amazon VPC 사용 설명서의 [Amazon DNS 서버](#) 섹션을 참조하십시오.

IPv6 주소

IPv6 CIDR 블록과 VPC를 연결하고 IPv6 CIDR 블록과 서브넷을 연결할 수도 있습니다. VPC에 대한 IPv6 CIDR 블록은 Amazon의 IPv6 주소 폴에서 자동으로 할당되므로 범위를 직접 선택할 수 없습니다. 자세한 내용은 Amazon VPC 사용 설명서에서 다음 주제를 참조하십시오.

- IPv6의 경우, VPC 및 서브넷 크기 조정
- IPv6 CIDR 블록을 VPC와 연결
- IPv6 CIDR 블록을 서브넷에 연결

IPv6 주소는 전역적으로 고유하므로 인터넷으로 접속할 수 있습니다. IPv6 CIDR 블록이 VPC와 서브넷에 연결되어 있고 다음 중 하나가 true이면 VPC의 인스턴스는 IPv6 주소를 받습니다.

- 서브넷은 시작 중인 인스턴스에 IPv6 주소를 자동으로 할당하도록 구성됩니다. 자세한 내용은 [서브넷의 IPv6 주소 지정 속성 설정](#)을 참조하십시오.
- 시작하는 동안 인스턴스에 IPv6 주소를 할당합니다.
- 시작 후 인스턴스의 기본 네트워크 인터페이스에 IPv6 주소를 할당합니다.
- 동일 서브넷에서 네트워크 인터페이스에 IPv6 주소를 할당하고 시작을 완료한 후에 인스턴스에 네트워크 인터페이스를 연결합니다.

시작하는 과정에서 인스턴스가 IPv6 주소를 받는 경우, 해당 주소는 인스턴스의 기본 네트워크 인터페이스 (eth0)와 연결됩니다. 네트워크 인터페이스에서 IPv6 주소 연결을 해제할 수 있습니다. 인스턴스 대해서는 IPv6 DNS 호스트 이름을 지원하지 않습니다.

인스턴스를 중지하고 시작할 때에는 IPv6 주소가 지속되다가 인스턴스를 종료하면 해제됩니다. IPv6 주소는 다른 네트워크 인터페이스—에 할당되는 동안에는 재할당할 수 없으므로, 먼저 할당을 해제해야 합니다.

인스턴스에 연결된 네트워크 인터페이스에 IPv6 주소를 할당함으로써 인스턴스에 추가 IPv6 주소를 할당할 수 있습니다. 네트워크 인터페이스에 할당할 수 있는 IPv6 주소의 개수, 그리고 인스턴스에 연결할 수 있는 네트워크 인터페이스의 개수는 인스턴스 유형에 따라 달라집니다. 자세한 내용은 [인스턴스 유형별/네트워크 인터페이스당 IP 주소 \(p. 514\)](#) 섹션을 참조하십시오.

EC2-Classic과 EC2-VPC의 IP 주소 차이점

다음 표는 EC2-Classic에서 시작된 인스턴스, 기본 서브넷에서 시작된 인스턴스 및 기본이 아닌 서브넷에서 시작된 인스턴스 사이의 IP 주소 차이점을 요약하여 설명합니다.

특성	EC2-Classic	기본 서브넷	기본이 아닌 서브넷
퍼블릭 IP 주소(Amazon 퍼블릭 IPv4 주소 풀에서 제공)	인스턴스에 퍼블릭 IP 주소가 할당됩니다.	시작 시 따로 지정하거나 서브넷의 퍼블릭 IP 주소 속성을 변경하지 않는 한 퍼블릭 IP 주소는 인스턴스에 기본으로 할당됩니다.	시작 시 따로 지정하거나 서브넷의 퍼블릭 IP 주소 속성을 변경하지 않는 한 퍼블릭 IP 주소는 인스턴스에 기본으로 할당되지 않습니다.
프라이빗 IPv4 주소	인스턴스를 시작할 때마다 EC2-Classic 범위 내의 프라이빗 IP 주소가 할당됩니다.	기본 서브넷 IPv4 주소 범위 내의 고정 프라이빗 IP 주소가 인스턴스에 할당됩니다.	서브넷 IPv4 주소 범위 내의 고정 프라이빗 IP 주소가 인스턴스에 할당됩니다.
다중 IPv4 주소	인스턴스별로 하나의 프라이빗 IP 주소가 할당되며 다중 IP 주소는 지원되지 않습니다.	인스턴스에 다중 프라이빗 IP 주소를 할당할 수 있습니다.	사용자는 인스턴스에 다중 프라이빗 IP 주소를 할당할 수 있습니다.
네트워크 인터페이스	IP 주소는 인스턴스와 연결됩니다. 네트워크 인터페이스는 지원되지 않습니다.	IP 주소는 네트워크 인터페이스와 연결됩니다. 각 인스턴스는 1개 이상의 네트워크 인터페이스를 갖습니다.	IP 주소는 네트워크 인터페이스와 연결됩니다. 각 인스턴스는 1개 이상의 네트워크 인터페이스를 갖습니다.
탄력적 IP 주소 (IPv4)	인스턴스를 중지하면 탄력적 IP 주소는 인스턴스에서 연결되지 않은 상태로 유지됩니다.	인스턴스를 중지하면 탄력적 IP 주소는 인스턴스와 연결된 상태를 유지합니다.	인스턴스를 중지하면 엘라스틱 IP 주소는 인스턴스와 연결된 상태를 유지합니다.
DNS 호스트 이름 (IPv4)	기본적으로 DNS 호스트 이름을 사용하도록 되어있습니다.	DNS 호스트 이름은 기본적으로 활성화되어 있습니다.	사용자가 Amazon VPC 콘솔에서 VPC 마법사를 통해 VPC를 생성한 경우를 제외하고 DNS 호스트이름은 기본적으로 비활성화됩니다.
IPv6 주소	지원하지 않음. 인스턴스는 IPv6 주소를 수신할 수 없습니다.	IPv6 CIDR 블록을 VPC와 서브넷에 연결한 후 시작 중에 IPv6 주소를 지정했거나 서브넷의 IPv6 주소 지정 속성을 수정하지 않은 한, 인스턴스는 기본적으로 IPv6 주소를 수신하지 않습니다.	IPv6 CIDR 블록을 VPC와 서브넷에 연결한 후 시작 중에 IPv6 주소를 지정했거나 서브넷의 IPv6 주소 지정 속성을 수정하지 않은 한, 인스턴스는 기본적으로 IPv6 주소를 수신하지 않습니다.

인스턴스에 대한 IP 주소 작업

인스턴스에 할당된 IP 주소를 확인하고, 시작 중에 퍼블릭 IPv4 주소를 인스턴스에 할당하며, 시작 중에 IPv6 주소를 인스턴스에 할당할 수 있습니다.

목차

- 퍼블릭, 프라이빗, 탄력적 IP 주소 결정 (p. 494)
- IPv6 주소 결정 (p. 495)
- 인스턴스 시작 시 퍼블릭 IPv4 주소 배정 (p. 495)
- 인스턴스에 IPv6 주소 할당 (p. 496)
- 인스턴스에 할당된 IPv6 주소 해제 (p. 497)

퍼블릭, 프라이빗, 탄력적 IP 주소 결정

사용자는 Amazon EC2 콘솔을 사용하여 인스턴스의 프라이빗 IPv4 주소, 퍼블릭 IPv4 주소 및 탄력적 IP 주소를 결정할 수 있습니다. 또한, 사용자는 인스턴스 메타데이터를 사용하여 인스턴스 내에서 인스턴스의 퍼블릭 IPv4 및 프라이빗 IPv4 주소를 결정할 수 있습니다. 자세한 내용은 [인스턴스 메타데이터 및 사용자 데이터 \(p. 321\)](#) 섹션을 참조하십시오.

콘솔을 이용하여 인스턴스의 프라이빗 IPv4 주소를 결정하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. 인스턴스를 선택합니다. 세부 정보 창의 [Private IPs] 필드에서 프라이빗 IPv4 주소를 획득하고 [Private DNS] 필드에서 내부 DNS 호스트 이름을 획득합니다.
4. (VPC에만 해당) 인스턴스에 연결된 네트워크 인터페이스에 1개 이상의 보조 프라이빗 IPv4 주소가 할당된 경우 해당 IP 주소는 [Secondary private IPs] 필드에서 획득할 수 있습니다.
5. (VPC에만 해당) 아니면, 탐색 창에서 [Network Interfaces]를 선택한 후 인스턴스에 연결된 네트워크 인터페이스를 선택합니다.
6. [Primary private IPv4 IP] 필드에서 기본 프라이빗 IP 주소를 획득하고 [Private DNS (IPv4)] 필드에서 내부 DNS 호스트 이름을 획득합니다.
7. 네트워크 인터페이스에 보조 프라이빗 IP 주소가 할당된 경우 해당 IP 주소는 [Secondary private IPv4 IPs]에서 획득할 수 있습니다.

콘솔을 이용하여 인스턴스의 퍼블릭 IPv4 주소를 결정하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. 인스턴스를 선택합니다. 세부 정보 창의 [IPv4 Public IPs] 필드에서 퍼블릭 IP 주소를 획득하고 [Public DNS (IPv4)] 필드에서 외부 DNS 호스트 이름을 획득합니다.
4. 인스턴스와 탄력적 IP 주소가 연결된 경우 [Elastic IPs] 필드에서 탄력적 IP 주소를 획득할 수 있습니다.

Note

인스턴스에 탄력적 IP 주소를 연결한 경우 [IPv4 Public IP] 필드에 탄력적 IP 주소도 표시됩니다.

5. (VPC에만 해당) 아니면, 탐색 창에서 [Network Interfaces]를 선택한 후 인스턴스에 연결된 네트워크 인터페이스를 선택합니다.
6. [IPv4 Public IP] 필드에서 퍼블릭 IP 주소를 획득합니다. 별표시(*)는 기본 프라이빗 IPv4 주소와 매핑된 퍼블릭 IPv4 주소 또는 탄력적 IP 주소를 나타냅니다.

Note

퍼블릭 IPv4 주소는 콘솔에서 네트워크 인터페이스의 속성으로 표시되지만 NAT를 통해 기본 프라이빗 IPv4 주소와 매핑됩니다. 그러므로, 예를 들어 ifconfig(Linux) 또는 ipconfig(Windows)를 통해 인스턴스 네트워크 카드의 속성을 확인하는 경우 퍼블릭 IPv4 주소는 표시되지 않습니다. 인스턴스의 퍼블릭 IPv4 주소를 인스턴스 내에서 결정하려면 인스턴스 메타데이터를 사용할 수 있습니다.

인스턴스 메타데이터를 이용하여 인스턴스의 IPv4 주소를 결정하려면

1. 인스턴스에 연결합니다.
2. 다음 명령을 사용하여 프라이빗 IP 주소에 액세스합니다.
 - Linux

```
$ curl http://169.254.169.254/latest/meta-data/local-ipv4
```

- Windows

```
$ wget http://169.254.169.254/latest/meta-data/local-ipv4
```

3. 다음 명령을 사용하여 퍼블릭 IP 주소에 액세스합니다.

- Linux

```
$ curl http://169.254.169.254/latest/meta-data/public-ipv4
```

- Windows

```
$ wget http://169.254.169.254/latest/meta-data/public-ipv4
```

인스턴스와 탄력적 IP 주소가 연결된 경우 반환된 값은 탄력적 IP 주소입니다.

IPv6 주소 결정

(VPC 전용) Amazon EC2 콘솔을 사용하여 인스턴스의 IPv6 주소를 결정할 수 있습니다.

콘솔을 이용하여 인스턴스의 IPv6 주소를 결정하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. 인스턴스를 선택합니다. 세부 정보 창의 [IPv6 IPs] 필드에서 IPv6 주소를 가져옵니다.

인스턴스 메타데이터를 이용하여 인스턴스의 IPv6 주소를 결정하려면

1. 인스턴스에 연결합니다.
2. 다음 명령을 사용하여 IPv6 주소를 확인합니다(<http://169.254.169.254/latest/meta-data/network/interfaces/macs/>에서 MAC 주소를 가져올 수 있음).
 - Linux

```
$ curl http://169.254.169.254/latest/meta-data/network/interfaces/macs/mac-address/ipv6s
```

- Windows

```
$ wget http://169.254.169.254/latest/meta-data/network/interfaces/macs/mac-address/ipv6s
```

인스턴스 시작 시 퍼블릭 IPv4 주소 배정

EC2-Classic에서 인스턴스를 시작하는 경우 기본적으로 퍼블릭 IPv4 주소가 할당됩니다. 사용자는 이 동작을 수정할 수 없습니다.

VPC의 경우 모든 서브넷은 퍼블릭 IP 주소가 할당되는 서브넷으로 인스턴스가 시작되는지를 결정하는 속성을 갖습니다. 기본적으로 기본이 아닌 서브넷의 이 속성은 `false`로 설정되고 기본 서브넷의 속성 값은 `true`입니다. 인스턴스를 시작할 때 퍼블릭 IPv4 주소 지정 기능을 사용하여 인스턴스에 퍼블릭 IPv4 주소가 할당되는지 여부를 제어할 수도 있습니다. 서브넷의 IP 주소 지정 속성의 기본 동작을 재정의할 수 있습니다. 퍼블릭 IPv4 주소는 Amazon의 퍼블릭 IPv4 주소 풀에서 할당되고 디바이스 색인이 eth0인 네트워크 인터페이스에 할당됩니다. 이 기능은 인스턴스 시작 시점의 특정 조건에 따라 달라집니다.

Important

사용자는 인스턴스가 시작된 이후에는 퍼블릭 IP 주소를 수동으로 해제할 수 없습니다. 대신, 특정 조건에 자동으로 해제되고 그 이후에 사용자는 해당 주소를 다시 사용할 수 없습니다. 자세한 내용은 [퍼블릭 IPv4 주소 및 외부 DNS 호스트 이름 \(p. 491\)](#)을 참조하십시오. 연결 또는 해제할 수 있는 영구 퍼블릭 IP 주소가 필요한 경우 시작 후에 인스턴스에 탄력적 IP 주소를 할당합니다. 자세한 내용은 [탄력적 IP 주소 \(p. 505\)](#) 섹션을 참조하십시오.

인스턴스 시작 시 퍼블릭 IP 주소 지정 기능에 액세스하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. [Launch Instance]를 선택합니다.
3. AMI와 인스턴스 유형을 선택한 후 [Next: Configure Instance Details]를 선택합니다.
4. [Configure Instance Details] 페이지의 [Network]에서 VPC를 선택합니다. [Auto-assign Public IP] 목록이 표시됩니다. [Enable] 또는 [Disable]를 선택하여 서브넷의 기본 설정을 재정의합니다.

Important

네트워크 인터페이스를 두 개 이상 지정하면 퍼블릭 IP 주소를 자동 할당할 수 없습니다. 또한 eth0에 대해 기존 네트워크 인터페이스를 지정하면 퍼블릭 IP 자동 할당 기능을 사용하여 서브넷 설정을 재정의할 수 없습니다.

5. 마법사의 다음 페이지에서 제공되는 단계를 따라 인스턴스 설정을 완료합니다. 마법사 구성 옵션에 대한 자세한 내용은 [인스턴스 시작하기 \(p. 265\)](#) 섹션을 참조하십시오. 마지막 [Review Instance Launch] 페이지에서는 설정을 검토한 후 [Launch]를 선택하여 키 쌍을 선택하고 인스턴스를 시작합니다.
6. [Instances] 페이지에서 새 인스턴스를 선택한 다음 세부 정보 창의 [IPv4 Public IP] 필드에서 퍼블릭 IP 주소를 확인합니다.

퍼블릭 IP 주소 지정 기능은 시작 동안에만 사용 가능합니다. 그러나 시작 도중에 퍼블릭 IP 주소가 인스턴스에 할당되는지의 여부와는 관계없이 시작 후에는 인스턴스와 탄력적 IP 주소를 연결할 수 있습니다. 자세한 내용은 [탄력적 IP 주소 \(p. 505\)](#)을 참조하십시오. 또한, 사용자는 서브넷의 퍼블릭 IPv4 주소 지정 동작을 변경할 수 있습니다. 자세한 내용은 [서브넷의 퍼블릭 IPv4 주소 지정 속성 수정](#)을 참조하십시오.

명령줄을 사용한 퍼블릭 IP 주소 지정 기능의 활성화 또는 비활성화 방법

- 다음 명령 중 하나를 사용할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.
 - `run-instances` 명령에서 `--associate-public-ip-address` 또는 `--no-associate-public-ip-address` 옵션을 사용합니다. (AWS CLI)
 - `New-EC2Instance` 명령과 함께 `-AssociatePublicIp` 파라미터를 사용합니다. (Windows PowerShell 용 AWS 도구)

인스턴스에 IPv6 주소 할당

VPC와 서브넷에 연결된 IPv6 CIDR 블록이 있는 경우, 시작 중 또는 시작 후 인스턴스에 IPv6 주소를 할당할 수 있습니다. IPv6 주소는 서브넷의 IPv6 주소 범위에서 할당되고 디바이스 색인이 eth0인 네트워크 인터페이스에 할당됩니다.

시작하는 과정에서 인스턴스에 IPv6 주소를 할당하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. AMI와 인스턴스 유형을 선택하고 [Next: Configure Instance Details]를 선택합니다.

Note

IPv6 주소를 지원하는 인스턴스 유형을 선택해야 합니다. 자세한 내용은 [인스턴스 유형 \(p. 146\)](#) 섹션을 참조하십시오.

3. [Configure Instance Details] 페이지의 [Network]에서 VPC를 선택하고 [Subnet]에서 서브넷을 선택합니다. [Auto-assign IPv6 IP]에 대해 [Enable]을 선택합니다.
4. 마법사의 나머지 단계를 수행하여 인스턴스를 시작합니다.

또는 시작을 완료한 후 인스턴스에 IPv6 주소를 할당할 수 있습니다.

시작 후 인스턴스에 IPv6 주소를 할당하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. 인스턴스를 선택하고 [Actions], [Manage IP Addresses]를 선택합니다.
4. [IPv6 Addresses]에서 [Assign new IP]를 선택합니다. 서브넷 범위에 속한 IPv6 주소를 지정하거나, Amazon이 IPv6 주소를 자동으로 선택하도록 [Auto-assign] 값을 그대로 둡니다.
5. [Save]를 선택합니다.

Note

Amazon Linux 2016.09.0 이상 버전 또는 Windows Server 2008 R2 이상 버전을 사용하여 인스턴스를 시작한 경우, 인스턴스는 IPv6에 맞게 구성되어 있으므로 IPv6 주소가 인스턴스에서 인식되는지 추가적으로 확인할 필요가 없습니다. 이전 AMI에서 인스턴스를 시작한 경우 인스턴스를 수동으로 구성해야 할 수 있습니다. 자세한 내용은 Amazon VPC 사용 설명서의 [인스턴스에서 IPv6 구성하기](#)를 참조하십시오.

명령줄을 사용하여 IPv6 주소를 할당하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- `run-instances` 명령에서 `--ipv6-addresses` 옵션을 사용합니다. (AWS CLI)
- `New-EC2Instance` 명령에서 `-NetworkInterface`에 대한 `Ipv6Addresses` 속성을 사용합니다. (Windows PowerShell용 AWS 도구)
- `assign-ipv6-addresses`(AWS CLI)
- `Register-EC2Ipv6AddressList`(Windows PowerShell용 AWS 도구)

인스턴스에 할당된 IPv6 주소 해제

Amazon EC2 콘솔을 사용하여 인스턴스에서 IPv6 주소 할당을 해제할 수 있습니다.

인스턴스에서 IPv6 주소 할당 해제

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

3. 인스턴스를 선택하고 [Actions], [Manage IP Addresses]를 선택합니다.
4. [IPv6 Addresses]에서 할당을 해제할 IPv6 주소에 대해 [Unassign]를 선택합니다.
5. [Yes, Update]를 선택합니다.

명령줄을 사용하여 IPv6 주소 할당을 해제하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [unassign-ipv6-addresses\(AWS CLI\)](#)
- [Unregister-EC2Ipv6AddressList \(Windows PowerShell용 AWS 도구\)](#).

다중 IP 주소

EC2-VPC에서 사용자는 인스턴스에 다중 프라이빗 IPv4 및 IPv6 주소를 지정할 수 있습니다. 인스턴스에 지정할 수 있는 네트워크 인터페이스 및 프라이빗 IPv4 및 IPv6 주소의 수는 인스턴스 유형에 의해 결정됩니다. 자세한 내용은 [인스턴스 유형별/네트워크 인터페이스당 IP 주소 \(p. 514\)](#) 섹션을 참조하십시오.

다음을 수행하여 VPC 인스턴스에 다중 IP 주소를 할당할 수 있습니다.

- 단일 서버에서 다중 SSL 인증서를 사용하거나 특정 IP 주소에 각 인증서를 연결하여 단일 서버에 다중 웹 사이트 호스팅.
- 각 네트워크 인터페이스에 다중 IP 주소가 있는 네트워크 어플라이언스(방화벽 또는 로드 밸런서 등) 운영.
- 대기 중인 인스턴스에 보조 IP 주소를 할당하여 인스턴스에서 오류가 발생한 경우 대기 인스턴스로 내부 트래픽 리디렉션.

목차

- [다중 IP 주소 동작 방법 \(p. 498\)](#)
- [다중 IPv4 주소 작업 \(p. 499\)](#)
- [다중 IPv6 주소 작업 \(p. 502\)](#)

다중 IP 주소 동작 방법

다음 목록은 다중 IP 주소를 갖는 네트워크 인터페이스의 동작 방법을 설명합니다.

- 사용자는 모든 네트워크 인터페이스에 보조 프라이빗 IPv4 주소를 할당할 수 있습니다. 네트워크 인터페이스는 인스턴스에서 연결 및 분리될 수 있습니다.
- 연결된 IPv6 CIDR 블록이 있는 서브넷의 네트워크 인터페이스에 다중 IPv6 주소를 할당할 수 있습니다.
- 네트워크 인터페이스의 서브넷 IPv4 CIDR 블록 범위 내에서 보조 IPv4를 선택해야 합니다.
- 네트워크 인터페이스의 서브넷 IPv6 CIDR 블록 범위 내에서 IPv6 주소를 선택해야 합니다.
- 보안 그룹은 IP 주소가 아닌 네트워크 인터페이스에 적용됩니다. 그러므로 IP 주소는 지정된 네트워크 인터페이스의 보안 그룹에 종속됩니다.
- 다중 IP 주소는 실행 중 또는 중지된 인스턴스에 연결된 네트워크 인터페이스에 할당되거나 할당되지 않을 수 있습니다.
- 네트워크 인터페이스에 할당된 보조 프라이빗 IPv4 주소는 사용자가 명시적으로 허용한 경우 다른 네트워크 인터페이스로 재할당될 수 있습니다.
- IPv6 주소는 다른 네트워크 인터페이스에 재할당될 수 없습니다. 우선 기준 네트워크 인터페이스에서 IPv6 주소의 할당을 해제해야 합니다.

- 명령줄 도구 또는 API를 이용하여 네트워크 인터페이스에 IP 주소를 여러 개 할당하는 경우 IP 주소 중 하나를 할당할 수 없으면 전체 작업이 실패하게 됩니다.
- 인스턴스에서 분리되거나 다른 인스턴스에 연결되어도 기본 프라이빗 IPv4 주소, 보조 프라이빗 IPv4 주소, 탄력적 IP 주소 및 IPv6 주소는 네트워크 인터페이스에 연결 상태를 유지합니다.
- 기본 네트워크 인터페이스는 인스턴스에서 이동할 수 없지만 기본 네트워크 인터페이스의 보조 프라이빗 IPv4 주소는 다른 네트워크 인터페이스로 재할당이 가능합니다.
- 사용자는 추가 네트워크 인터페이스를 한 인스턴스에서 다른 인스턴스로 이동시킬 수 있습니다.

다음 목록은 다중 IP 주소를 갖는 탄력적 IP 주소의 동작 방법을 설명합니다(IPv4만 해당).

- 각 프라이빗 IPv4 주소는 단일 탄력적 IP 주소로 연결될 수 있고 그 반대도 가능합니다.
- 보조 프라이빗 IPv4 주소가 다른 인터페이스로 재할당된 경우 보조 프라이빗 IPv4 주소와 탄력적 IP 주소는 연결 상태를 유지합니다.
- 보조 프라이빗 IPv4 주소가 인터페이스에서 할당이 해제된 경우 연결된 탄력적 IP 주소는 보조 프라이빗 IPv4 주소에서 자동으로 할당이 해제됩니다.

다중 IPv4 주소 작업

보조 프라이빗 IPv4 주소를 인스턴스에 할당하고 탄력적 IPv4 주소와 보조 프라이빗 IPv4 주소를 연결하며, 보조 프라이빗 IPv4 주소의 할당을 해제할 수 있습니다.

목차

- [보조 프라이빗 IPv4 주소 할당](#) (p. 499)
- [인스턴스에 운영 체제를 구성하여 보조 프라이빗 IPv4 주소 인식](#) (p. 501)
- [탄력적 IP 주소와 보조 프라이빗 IPv4 주소 연결](#) (p. 501)
- [보조 프라이빗 IPv4 주소 확인](#) (p. 501)
- [보조 프라이빗 IPv4 주소 할당 해제](#) (p. 502)

보조 프라이빗 IPv4 주소 할당

사용자는 인스턴스 시작 시 또는 인스턴스가 실행된 다음 인스턴스의 네트워크 인터페이스에 보조 프라이빗 IPv4 주소를 할당할 수 있습니다. 이 섹션에는 다음 절차가 포함됩니다.

- [EC2-VPC에서 인스턴스를 시작할 때 보조 프라이빗 IPv4 주소를 할당하려면](#) (p. 499)
- [명령줄을 이용하여 시작 중에 보조 IPv4 주소를 할당하려면](#) (p. 500)
- [네트워크 인터페이스에 보조 프라이빗 IPv4 주소를 할당하려면](#) (p. 500)
- [명령줄을 이용하여 기존 인스턴스에 보조 프라이빗 IPv4 주소를 할당하려면](#) (p. 501)

EC2-VPC에서 인스턴스를 시작할 때 보조 프라이빗 IPv4 주소를 할당하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. [Launch Instance]를 선택합니다.
3. AMI를 선택하고 인스턴스 유형을 선택한 후 [Next: Configure Instance Details]를 선택합니다.
4. [Configure Instance Details] 페이지의 [Network]에서 VPC를 선택하고 [Subnet]에서 서브넷을 선택합니다.
5. [Network Interfaces] 섹션에서 다음을 수행하고 [Next: Add Storage]를 선택합니다.
 - 다른 네트워크 인터페이스를 추가하려면 [Add Device]를 선택합니다. 콘솔을 사용하여 인스턴스 시작 시 네트워크 인터페이스를 최대 두 개 지정할 수 있습니다. 인스턴스를 시작한 후 탐색 창에서

[Network Interfaces]를 선택하여 네트워크 인터페이스를 추가합니다. 연결 가능한 총 네트워크 인터페이스의 수는 인스턴스 유형에 따라 다릅니다. 자세한 내용은 [인스턴스 유형별/네트워크 인터페이스당 IP 주소 \(p. 514\)](#) 섹션을 참조하십시오.

Important

두 번째 네트워크 인터페이스를 추가하면 시스템에서 더 이상 퍼블릭 IPv4 주소를 자동 할당할 수 없습니다. 기본 네트워크 인터페이스(eth0)에 탄력적 IP 주소를 할당하지 않는 이상 IPv4를 통해 인스턴스에 연결할 수 없습니다. 시작 마법사를 완료한 후에는 탄력적 IP 주소를 할당할 수 있습니다. 자세한 내용은 [탄력적 IP 주소 작업 \(p. 508\)](#) 섹션을 참조하십시오.

- 각 네트워크 인터페이스에 대해 [Secondary IP addresses]에서 [Add IP]를 선택한 후 서브넷 범위 내의 프라이빗 IP 주소를 입력하거나, 기본 설정인 Auto-assign을 수락하여 Amazon의 주소 선택을 허용합니다.
- 6. 다음 Add Storage 페이지에서 사용자는 볼륨을 지정하여 AMI에 의해 지정된 볼륨 옆에 인스턴스(루트 디바이스 볼륨 등)를 연결한 다음 Next: Add Tags를 선택합니다.
- 7. Add Tags 페이지에서 인스턴스에 태그(예: 사용자에게 친숙한 이름)를 지정한 후 Next: Configure Security Group을 선택합니다.
- 8. [Configure Security Group] 페이지에서 기존 보안 그룹을 선택하거나 새 보안 그룹을 생성합니다. [Review and Launch]를 선택합니다.
- 9. [Review Instance Launch] 페이지에서 설정을 검토한 후 [Launch]를 선택하여 키 페어를 선택하고 인스턴스를 시작합니다. Amazon EC2를 처음 사용하며 아직 키 페어를 생성하지 않은 경우 키 페어를 생성하라는 메시지가 마법사에 표시됩니다.

Important

네트워크 인터페이스에 보조 프라이빗 IP 주소를 추가한 이후에는 인스턴스에 연결하고 인스턴스 자체에 보조 프라이빗 IP 주소를 구성해야 합니다. 자세한 내용은 [인스턴스에 운영 체제를 구성하여 보조 프라이빗 IPv4 주소 인식 \(p. 501\)](#) 섹션을 참조하십시오.

명령줄을 이용하여 시작 중에 보조 IPv4 주소를 할당하려면

- 다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.
 - `run-instances` 명령에서 `--secondary-private-ip-addresses` 옵션(AWS CLI)
 - `-NetworkInterface`를 정의하고 `New-EC2Instance` 명령(Windows PowerShell용 AWS 도구)과 함께 `PrivateIpAddresses` 파라미터를 지정합니다.

네트워크 인터페이스에 보조 프라이빗 IPv4 주소를 할당하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Network Interfaces]를 선택한 후 인스턴스에 연결된 네트워크 인터페이스를 선택합니다.
3. [Actions], [Manage IP Addresses]를 선택합니다.
4. [IPv4 Addresses]에서 [Assign new IP]를 선택합니다.
5. 인스턴스의 서브넷 범위 내에 있는 특정 IPv4 주소를 입력합니다. 또는 필드를 공란으로 남기면 Amazon에서 IP 주소를 자동으로 선택합니다.
6. (선택 사항) [Allow reassignment]를 선택하면 다른 네트워크 인터페이스가 이미 할당된 경우 보조 프라이빗 IP 주소가 재할당됩니다.
7. [Yes, Update]를 선택합니다.

대안으로, 사용자는 인스턴스에 보조 프라이빗 IPv4 주소를 할당할 수 있습니다. 탐색 창에서 [Instances]를 선택하고 인스턴스를 선택한 후 [Actions], [Networking], [Manage IP Addresses]를 차례로 선택합니다. 위의

단계와 마찬가지로 동일한 정보를 구성할 수 있습니다. IP 주소는 인스턴스에 대한 기본 네트워크 인터페이스(eth0)에 할당됩니다.

명령줄을 이용하여 기존 인스턴스에 보조 프라이빗 IPv4 주소를 할당하려면

- 다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.
 - `assign-private-ip-addresses` (AWS CLI)
 - `Register-EC2PrivateIpAddress` (Windows PowerShell용 AWS 도구)

인스턴스에 운영 체제를 구성하여 보조 프라이빗 IPv4 주소 인식

인스턴스에 보조 프라이빗 IPv4 주소를 할당한 이후에는 인스턴스에 운영 체제를 구성하여 보조 프라이빗 IP 주소가 인식되어야 합니다.

- Amazon Linux를 사용하는 경우 `ec2-net-utils` 패키지로 이 단계를 수행할 수 있습니다. `ec2-net-utils`는 인스턴스 실행 중에 사용자가 연결한 추가 네트워크 인터페이스를 구성하고 DHCP 임대가 갱신되는 동안 보조 IPv4 주소를 새로 고침하며 관련이 있는 라우팅 규칙을 업데이트합니다. `sudo service network restart` 명령을 사용하여 인터페이스 목록을 즉시 새로 고친 다음, `ip addr li`를 사용하여 최신 목록을 볼 수 있습니다. 네트워크 구성을 수동으로 설정해야 하는 경우 `ec2-net-utils` 패키지를 삭제하면 됩니다. 자세한 내용은 [ec2-net-utils를 사용하여 네트워크 인터페이스 구성 \(p. 518\)](#) 섹션을 참조하십시오.
- 다른 Linux 배포판을 사용하는 경우 해당 Linux 배포판에서 제공된 문서를 참조하십시오. 추가 네트워크 인터페이스 및 보조 IPv4 주소 구성 정보를 검색합니다. 동일 네트워크에 있는 인스턴스에 인터페이스가 1개 이상 있는 경우 라우팅 규칙을 사용하여 비대칭 라우팅으로 동작하는 것과 관련된 정보를 검색합니다.

Windows 인스턴스에 대한 자세한 내용은 Windows 인스턴스용 Amazon EC2 사용 설명서의 [VPC에서 Windows 인스턴스에 보조 프라이빗 IP 주소 구성](#)을 참조합니다.

탄력적 IP 주소와 보조 프라이빗 IPv4 주소 연결

EC2-VPC에서 탄력적 IP 주소와 보조 프라이빗 IPv4 주소를 연결하려면

- <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
- 탐색 창에서 [Elastic IPs]를 선택합니다.
- Actions를 선택한 후 Associate address를 선택합니다.
- Network interface에서 네트워크 인터페이스를 선택한 다음 Private IP 목록에서 보조 IP 주소를 선택합니다.
- [Associate]를 선택합니다.

명령줄을 이용하여 탄력적 IP 주소와 보조 프라이빗 IPv4 주소를 연결하려면

- 다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.
 - `associate-address`(AWS CLI)
 - `Register-EC2Address`(Windows PowerShell용 AWS 도구)

보조 프라이빗 IPv4 주소 확인

EC2-VPC에서 네트워크 인터페이스에 할당된 프라이빗 IPv4 주소를 확인하려면

- <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.

2. 탐색 창에서 Network Interfaces를 선택합니다.
3. 확인할 프라이빗 IP 주소를 갖는 네트워크 인터페이스를 선택합니다.
4. 세부 정보 창의 [Details] 탭에서 네트워크 인터페이스에 할당된 기본 프라이빗 IPv4 주소 및 모든 보조 프라이빗 IPv4 주소의 [Primary private IPv4 IP] 및 [Secondary private IPv4 IPs] 필드에 체크 표시합니다.

인스턴스에 할당된 프라이빗 IPv4 주소를 확인하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. 확인할 프라이빗 IPv4 주소를 갖는 인스턴스를 선택합니다.
4. 세부 정보 창의 [Description] 탭에서 네트워크 인터페이스를 통해 할당된 기본 프라이빗 IPv4 주소 및 모든 보조 프라이빗 IPv4 주소의 [Private IPs] 및 [Secondary private IPs] 필드에 체크 표시합니다.

보조 프라이빗 IPv4 주소 할당 해제

보조 프라이빗 IPv4 주소가 더 이상 필요하지 않은 경우 인스턴스 또는 네트워크 인터페이스에서 해당 주소를 할당 해제할 수 있습니다. 보조 프라이빗 IPv4 주소가 네트워크 인터페이스에서 할당이 해제된 경우 탄력적 IP 주소(존재하는 경우)도 또한 연결이 해제됩니다.

인스턴스에서 보조 프라이빗 IPv4 주소의 할당을 해제하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. 인스턴스를 선택하고 [Actions], [Networking], [Manage IP Addresses]를 선택합니다.
4. [IPv4 Addresses]에서 할당을 해제할 IPv4 주소에 대해 [Unassign]을 선택합니다.
5. [Yes, Update]를 선택합니다.

네트워크 인터페이스에서 보조 프라이빗 IPv4 주소의 할당을 해제하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.
3. 네트워크 인터페이스를 선택하고 [Actions], [Manage IP Addresses]를 선택합니다.
4. [IPv4 Addresses]에서 할당을 해제할 IPv4 주소에 대해 [Unassign]을 선택합니다.
5. [Yes, Update]를 선택합니다.

명령줄을 이용하여 보조 프라이빗 IPv4 주소의 할당을 해제하려면

- 다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.
 - [unassign-private-ip-addresses \(AWS CLI\)](#)
 - [Unregister-EC2PrivateIpAddress \(Windows PowerShell용 AWS 도구\)](#)

다중 IPv6 주소 작업

다중 IPv6 주소를 인스턴스에 할당하고, 인스턴스에 할당된 IPv6 주소를 확인하며, 인스턴스에서 IPv6 주소 할당을 해제할 수 있습니다.

목차

- [다중 IPv6 주소 할당 \(p. 503\)](#)
- [IPv6 주소 확인 \(p. 504\)](#)
- [IPv6 주소 할당 해제 \(p. 505\)](#)

다중 IPv6 주소 할당

시작 중 또는 시작 후 인스턴스에 하나 이상의 IPv6 주소를 할당할 수 있습니다. 인스턴스에 IPv6 주소를 할당하려면 인스턴스를 시작하는 VPC와 서브넷에 연결된 IPv6 CIDR 블록이 있어야 합니다. 자세한 내용은 [Amazon VPC 사용 설명서](#)의 VPCs and Subnets 섹션을 참조하십시오.

시작 중에 다중 IPv6 주소 할당

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 대시보드에서 [Launch Instance]를 선택합니다.
3. AMI를 선택하고 인스턴스 유형을 선택한 후 [Next: Configure Instance Details]를 선택합니다. IPv6를 지원하는 인스턴스 유형을 선택해야 합니다. 자세한 내용은 [인스턴스 유형 \(p. 146\)](#) 섹션을 참조하십시오.
4. [Configure Instance Details] 페이지의 [Network] 목록에서 VPC를 선택한 다음 [Subnet] 목록에서 서브넷을 선택합니다.
5. [Network Interfaces] 섹션에서 다음을 수행하고 [Next: Add Storage]를 선택합니다.
 - 기본 네트워크 인터페이스(eth0)에 단일 IPv6 주소를 할당하려면 [IPv6 IPs]에서 [Add IP]를 선택합니다. 보조 IPv6 주소를 추가하려면 [Add IP]를 다시 선택합니다. 서브넷 범위에 속한 IPv6 주소를 직접 입력하거나, Amazon이 서브넷에 속한 IPv6 주소를 자동으로 선택하도록 기본 [Auto-assign] 값을 그대로 둘 수 있습니다.
 - 다른 네트워크 인터페이스를 추가하려면 [Add Device]를 선택하고, 하나 이상의 IPv6 주소를 네트워크 인터페이스에 추가하려면 단계를 반복합니다. 콘솔을 사용하여 인스턴스 시작 시 네트워크 인터페이스를 최대 두 개 지정할 수 있습니다. 인스턴스를 시작한 후 탐색 창에서 [Network Interfaces]를 선택하여 네트워크 인터페이스를 추가합니다. 연결 가능한 총 네트워크 인터페이스의 수는 인스턴스 유형에 따라 다릅니다. 자세한 내용은 [인스턴스 유형별/네트워크 인터페이스당 IP 주소 \(p. 514\)](#) 섹션을 참조하십시오.
6. 마법사의 다음 단계를 수행하여 볼륨을 연결하고 인스턴스에 태그를 지정합니다.
7. [Configure Security Group] 페이지에서 기존 보안 그룹을 선택하거나 새 보안 그룹을 생성합니다. IPv6를 통해 인스턴스에 연결할 수 있으려면 보안 그룹에 IPv6 주소로부터 액세스하도록 허용하는 규칙이 있어야 합니다. 자세한 내용은 [보안 그룹 규칙 참조 \(p. 393\)](#) 섹션을 참조하십시오. [Review and Launch]를 선택합니다.
8. [Review Instance Launch] 페이지에서 설정을 검토한 후 [Launch]를 선택하여 키 페어를 선택하고 인스턴스를 시작합니다. Amazon EC2를 처음 사용하며 아직 키 페어를 생성하지 않은 경우 키 페어를 생성하라는 메시지가 마법사에 표시됩니다.

[Instances] 화면 Amazon EC2 콘솔을 사용하여 기존 인스턴스에 다중 IPv6 주소를 할당할 수 있습니다. 그러면 인스턴스의 기본 네트워크 인터페이스(eth0)에 IPv6 주소가 할당됩니다. 인스턴스에 특정 IPv6 주소를 할당하려면 IPv6 주소에 이미 다른 인스턴스나 네트워크 인터페이스가 할당되어 있어서는 안 됩니다.

기존 인스턴스에 다중 IPv6 주소를 할당하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. 인스턴스를 선택하고 [Actions], [Manage IP Addresses]를 선택합니다.
4. [IPv6 Addresses]에서 추가할 IPv6 주소에 대해 [Assign new IP]를 선택합니다. 서브넷 범위에 속한 IPv6 주소를 지정하거나, Amazon이 IPv6 주소를 자동으로 선택하도록 [Auto-assign] 값을 그대로 둡니다.
5. [Yes, Update]를 선택합니다.

또는 기존 네트워크 인터페이스에 다중 IPv6 주소를 할당할 수도 있습니다. 네트워크 인터페이스는 연결된 IPv6 CIDR 블록이 있는 서브넷에서 생성되어야 합니다. 네트워크 인터페이스에 특정 IPv6 주소를 할당하려면 IPv6 주소에 이미 다른 네트워크 인터페이스가 할당되어 있어서는 안 됩니다.

네트워크 인터페이스에 다중 IPv6 주소를 할당하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.
3. 네트워크 인터페이스를 선택하고 [Actions], [Manage IP Addresses]를 선택합니다.
4. [IPv6 Addresses]에서 추가할 IPv6 주소에 대해 [Assign new IP]를 선택합니다. 서브넷 범위에 속한 IPv6 주소를 지정하거나, Amazon이 IPv6 주소를 자동으로 선택하도록 [Auto-assign] 값을 그대로 둡니다.
5. [Yes, Update]를 선택합니다.

CLI 개요

다음 명령 중 하나를 사용할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- 시작 중에 IPv6 주소 할당:
 - `run-instances` 명령과 함께 `--ipv6-addresses` 또는 `--ipv6-address-count` 옵션을 사용합니다. (AWS CLI)
 - `-NetworkInterface`를 정의하고 `New-EC2Instance` 명령과 함께 `Ipv6Addresses` 또는 `Ipv6AddressCount` 파라미터를 지정합니다. (Windows PowerShell용 AWS 도구).
- 네트워크 인터페이스에 IPv6 주소 할당:
 - `assign-ipv6-addresses`(AWS CLI)
 - `Register-EC2Ipv6AddressList`(Windows PowerShell용 AWS 도구)

IPv6 주소 확인

인스턴스 또는 네트워크 인터페이스에 대한 IPv6 주소를 확인할 수 있습니다.

인스턴스에 할당된 IPv6 주소를 확인하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. 인스턴스를 선택합니다. 세부 정보 창에서 [IPv6 IPs] 필드를 검토합니다.

네트워크 인터페이스에 할당된 IPv6 주소를 확인하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.
3. 네트워크 인터페이스를 선택합니다. 세부 정보 창에서 [IPv6 IPs] 필드를 검토합니다.

CLI 개요

다음 명령 중 하나를 사용할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- 인스턴스에 대한 IPv6 주소 확인:
 - `describe-instances` (AWS CLI)

- [Get-EC2Instance](#) (Windows PowerShell용 AWS 도구).
- 네트워크 인터페이스에 대한 IPv6 주소 확인:
 - [describe-network-interfaces](#)(AWS CLI)
 - [Get-EC2NetworkInterface](#)(Windows PowerShell용 AWS 도구)

IPv6 주소 할당 해제

인스턴스의 기본 네트워크 인터페이스에서 IPv6 주소 할당을 해제하거나 네트워크 인터페이스에서 IPv6 주소 할당을 해제할 수 있습니다.

인스턴스에서 IPv6 주소 할당 해제

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. 인스턴스를 선택하고 [Actions], [Manage IP Addresses]를 선택합니다.
4. [IPv6 Addresses]에서 할당을 해제할 IPv6 주소에 대해 [Unassign]를 선택합니다.
5. [Yes, Update]를 선택합니다.

네트워크 인터페이스에 할당된 IPv6 주소를 해제하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.
3. 네트워크 인터페이스를 선택하고 [Actions], [Manage IP Addresses]를 선택합니다.
4. [IPv6 Addresses]에서 할당을 해제할 IPv6 주소에 대해 [Unassign]를 선택합니다.
5. [Save]를 선택합니다.

CLI 개요

다음 명령 중 하나를 사용할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [unassign-ipv6-addresses](#)(AWS CLI)
- [Unregister-EC2Ipv6AddressList](#) (Windows PowerShell용 AWS 도구).

탄력적 IP 주소

탄력적 IP 주소는 동적 클라우드 컴퓨팅을 위해 고안된 고정 IPv4 주소입니다. 탄력적 IP 주소는 AWS 계정과 연결됩니다. 탄력적 IP 주소를 사용하면 주소를 계정의 다른 인스턴스에 신속하게 다시 매핑하여 인스턴스나 소프트웨어의 오류를 마스킹할 수 있습니다.

탄력적 IP 주소는 인터넷에서 연결 가능한 퍼블릭 IPv4 주소입니다. 인스턴스에 퍼블릭 IP 주소가 없는 경우 탄력적 IPv4 주소를 인스턴스와 연결하여 인터넷과 통신을 활성화하고 예를 들어 로컬 컴퓨터에서 인스턴스에 연결할 수 있습니다.

현재는 IPv6에 대한 탄력적 IP 주소를 지원하지 않습니다.

항목

- [탄력적 IP 주소 기본 사항 \(p. 506\)](#)
- [EC2-Classic 및 EC2-VPC의 탄력적 IP 주소의 차이점 \(p. 506\)](#)

- [탄력적 IP 주소 작업 \(p. 508\)](#)
- [이메일 애플리케이션에 역방향 DNS 사용 \(p. 512\)](#)
- [탄력적 IP 주소 제한 \(p. 512\)](#)

탄력적 IP 주소 기본 사항

탄력적 IP 주소의 기본 특성은 다음과 같습니다.

- 탄력적 IP 주소를 사용하려면 먼저 계정에 주소를 할당한 후 인스턴스 또는 네트워크 인터페이스와 연결합니다.
- 탄력적 IP 주소를 인스턴스 또는 기본 네트워크 인터페이스와 연결하면, 인스턴스의 퍼블릭 IPv4 주소(있는 경우)는 Amazon의 퍼블릭 IPv4 주소 폴로 다시 릴리스됩니다. 퍼블릭 IPv4 주소는 재사용할 수 없습니다. 자세한 내용은 [퍼블릭 IPv4 주소 및 외부 DNS 호스트 이름 \(p. 491\)](#) 섹션을 참조하십시오.
- 탄력적 IP 주소는 리소스에서 연결 해제했다가 다른 리소스와 다시 연결할 수 있습니다.
- 연결 해제한 탄력적 IP 주소는 명시적으로 릴리스할 때까지 계정에 할당되어 있습니다.
- 탄력적 IP 주소의 효율적인 사용을 위해 탄력적 IP 주소가 실행 중인 인스턴스와 연결되어 있지 않거나 중지된 인스턴스 또는 연결되지 않은 네트워크 인터페이스와 연결된 경우 소액의 시간당 요금이 부과됩니다. 인스턴스가 실행 중인 동안에는 이와 연결된 탄력적 IP 주소 하나에 대해서는 요금이 부과되지 않지만 해당 인스턴스와 연결된 추가 탄력적 IP 주소에 대해서는 요금이 부과됩니다. 자세한 내용은 [Amazon EC2 요금](#)을 참조하십시오.
- 탄력적 IP 주소는 특정 리전에서만 사용됩니다.
- 탄력적 IP 주소를 이전에 퍼블릭 IPv4 주소가 있었던 인스턴스와 연결하면 인스턴스의 퍼블릭 DNS 호스트 이름이 탄력적 IP 주소에 맞게 변경됩니다.
- Amazon은 퍼블릭 DNS 호스트 이름을 인스턴스 네트워크 외부에서는 인스턴스의 퍼블릭 IPv4 주소 또는 탄력적 IP 주소로 변환하고, 인스턴스 네트워크 내부에서는 인스턴스의 프라이빗 IPv4 주소로 변환합니다.

계정이 EC2-Classic을 지원하는 경우, EC2-Classic과 EC2-VPC에서 탄력적 IP 주소의 사용 및 동작이 다를 수 있습니다. 자세한 내용은 [EC2-Classic 및 EC2-VPC의 탄력적 IP 주소의 차이점 \(p. 506\)](#) 섹션을 참조하십시오.

EC2-Classic 및 EC2-VPC의 탄력적 IP 주소의 차이점

사용자 계정이 EC2-Classic을 지원하는 경우 EC2-Classic 플랫폼에 사용할 수 있는 탄력적 IP 주소 폴과 EC2-VPC 플랫폼에 사용할 수 있는 탄력적 IP 주소 폴이 하나씩 있습니다. VPC에 사용하도록 할당한 탄력적 IP 주소를 EC2-Classic의 인스턴스와 연결할 수 없으며 그 반대의 경우도 마찬가지입니다. 하지만 EC2-Classic 플랫폼에서 사용할 목적으로 할당한 탄력적 IP 주소는 EC2-VPC 플랫폼으로 마이그레이션할 수 있습니다. 탄력적 IP 주소는 다른 리전으로 마이그레이션할 수 없습니다. EC2-Classic 및 EC2-VPC에 대한 자세한 내용은 [지원되는 플랫폼 \(p. 471\)](#) 섹션을 참조하십시오.

EC2-Classic(기본 VPC)의 인스턴스나 시작 시 eth0 네트워크 인터페이스에 퍼블릭 IPv4를 할당한 기본이 아닌 VPC의 인스턴스와 탄력적 IP 주소를 연결하면 인스턴스의 현재 퍼블릭 IPv4 주소가 퍼블릭 IP 주소 폴로 다시 릴리스됩니다. 인스턴스에서 탄력적 IP 주소의 연결을 끊으면 인스턴스에 새 퍼블릭 IPv4 주소가 몇 분 안에 자동으로 할당됩니다. 그러나 VPC의 인스턴스에 두 번째 네트워크 인터페이스를 연결한 경우 인스턴스에 새 퍼블릭 IPv4 주소가 자동으로 배정되지 않습니다. 퍼블릭 IPv4 주소에 대한 자세한 내용은 [퍼블릭 IPv4 주소 및 외부 DNS 호스트 이름 \(p. 491\)](#) 섹션을 참조하십시오.

VPC의 인스턴스에 탄력적 IP 주소를 사용하는 것에 대한 자세한 내용은 Amazon VPC 사용 설명서의 [탄력적 IP 주소](#)를 참조하십시오.

다음 표에는 EC2-Classic의 탄력적 IP 주소와 EC2-VPC의 탄력적 IP 주소 간의 차이점이 나와 있습니다. 프라이빗 IP 주소와 퍼블릭 IP 주소의 차이에 대한 자세한 내용은 [EC2-Classic과 EC2-VPC의 IP 주소 차이점 \(p. 493\)](#) 섹션을 참조하십시오.

특성	EC2-Classic	EC2-VPC
탄력적 IP 주소 할당	탄력적 IP 주소를 할당하면 EC2-Classic에서 사용됩니다. 하지만 탄력적 IP 주소를 EC2-VPC 플랫폼으로 마이그레이션할 수 있습니다. 자세한 내용은 EC2-Classic에서 EC2-VPC로 탄력적 IP 주소의 마이그레이션 (p. 507) 섹션을 참조하십시오.	탄력적 IP 주소를 할당할 경우 VPC에서만 사용됩니다.
탄력적 IP 주소 연결	탄력적 IP 주소를 인스턴스와 연결할 수 있습니다.	탄력적 IP 주소는 네트워크 인터페이스의 속성입니다. 인스턴스에 연결된 네트워크 인터페이스를 업데이트하여 엘라스틱 IP 주소를 인스턴스와 연결할 수 있습니다. 자세한 내용은 탄력적 네트워크 인터페이스 (p. 512) 섹션을 참조하십시오.
탄력적 IP 주소 재연결	다른 인스턴스와 이미 연결된 탄력적 IP 주소를 연결하려고 하면 주소가 자동으로 새 인스턴스와 연결됩니다.	계정에서 EC2-VPC만 지원할 경우 다른 인스턴스와 이미 연결된 탄력적 IP 주소를 연결하려고 하면 주소가 자동으로 새 인스턴스와 연결됩니다. EC2-Classic 계정에서 VPC를 사용할 경우 다른 인스턴스와 이미 연결된 탄력적 IP 주소를 연결하려고 시도하면, 재연결이 허용될 경우에만 성공합니다.
탄력적 IP 주소를 기존 탄력적 IP 주소가 할당된 대상과 연결	기존 탄력적 IP 주소는 인스턴스에서 연결 해제되지만 계정에 할당되어 있는 상태는 유지됩니다.	계정이 EC2-VPC만 지원하는 경우에는 기존 탄력적 IP 주소가 인스턴스에서 연결 해제되지만 계정에 할당되어 있는 상태는 유지됩니다. EC2-Classic 계정에서 VPC를 사용하려면 탄력적 IP 주소를 기존 탄력적 IP 주소가 할당된 네트워크 인터페이스 또는 인스턴스와 연결할 수 없습니다.
인스턴스 종지	인스턴스를 종지하면 해당 탄력적 IP 주소와의 연결이 끊어져서 인스턴스를 다시 시작할 때 탄력적 IP 주소를 재연결해야 합니다.	인스턴스를 종지할 경우 인스턴스의 탄력적 IP 주소는 연결된 상태로 유지됩니다.
다중 IP 주소 할당	인스턴스가 하나의 프라이빗 IPv4 주소와 해당 탄력적 IP 주소만 지원합니다.	인스턴스가 여러 개의 IPv4 주소를 지원하며 각각 해당 탄력적 IP 주소를 가질 수 있습니다. 자세한 내용은 다중 IP 주소 (p. 498) 섹션을 참조하십시오.

EC2-Classic에서 EC2-VPC로 탄력적 IP 주소의 마이그레이션

계정이 EC2-Classic을 지원하는 경우에는 EC2-Classic 플랫폼에서 사용할 목적으로 할당한 탄력적 IP 주소를 동일 리전 내 EC2-VPC 플랫폼으로 마이그레이션할 수 있습니다. EC2-Classic에서 VPC로 리소스를 마이그레이션할 수 있는 이유도 바로 여기에 있습니다. 예를 들어, VPC에서 새로운 웹 서버를 실행한 다음 EC2-Classic의 웹 서버에서 사용한 탄력적 IP 주소를 새로운 VPC 웹 서버에도 똑같이 사용할 수 있습니다.

탄력적 IP 주소를 EC2-VPC로 마이그레이션한 후에는 EC2-Classic 플랫폼에서 다시 사용하지 못합니다. 하지만 혹시라도 필요할 경우 EC2-Classic으로 복구할 수는 있습니다. 탄력적 IP 주소를 EC2-Classic으로 복구한 후 EC2-VPC에서 다시 사용하려면 한 번 더 마이그레이션해야 합니다. 탄력적 IP 주소는 EC2-Classic에서 EC2-VPC로만 마이그레이션할 수 있기 때문입니다. 처음부터 EC2-VPC에서 사용할 목적으로 할당한 탄력적 IP 주소는 EC2-Classic으로 마이그레이션하지 못합니다.

탄력적 IP 주소를 마이그레이션하려면 연결되어 있는 인스턴스가 없어야 합니다. 탄력적 IP 주소를 인스턴스에서 연결 해제하기 위한 자세한 내용은 [탄력적 IP 주소의 연결 해제 후 다른 인스턴스와 재연결 \(p. 509\)](#) 섹션을 참조하십시오.

EC2-Classic 탄력적 IP 주소는 계정에 가질 수 있는 만큼 마이그레이션할 수 있습니다. 하지만 탄력적 IP 주소를 EC2-VPC로 마이그레이션할 때는 EC2-VPC의 탄력적 IP 주소 최대 수에서 차감됩니다. 따라서 최대 수를 초과하는 경우에는 탄력적 IP 주소를 마이그레이션할 수 없습니다. 마찬가지로 탄력적 IP 주소를 EC2-Classic로 복구할 때도 EC2-Classic의 탄력적 IP 주소의 최대 수에서 차감됩니다. 자세한 내용은 [탄력적 IP 주소 제한 \(p. 512\)](#) 섹션을 참조하십시오.

계정에 할당되고 24시간이 지나지 않은 엘라스틱 IP 주소는 마이그레이션할 수 없습니다.

자세한 내용은 [탄력적 IP 주소 이동 \(p. 510\)](#) 섹션을 참조하십시오.

탄력적 IP 주소 작업

다음 섹션에서는 탄력적 IP 주소를 이용한 작업 방법에 대해 살펴보겠습니다.

항목

- [탄력적 IP 주소 할당 \(p. 508\)](#)
- [탄력적 IP 주소 설명 \(p. 509\)](#)
- [실행 중인 인스턴스와 탄력적 IP 주소 연결 \(p. 509\)](#)
- [탄력적 IP 주소의 연결 해제 후 다른 인스턴스와 재연결 \(p. 509\)](#)
- [탄력적 IP 주소 이동 \(p. 510\)](#)
- [탄력적 IP 주소 해제 \(p. 511\)](#)

탄력적 IP 주소 할당

Amazon EC2 콘솔이나 명령줄을 사용하여 탄력적 IP 주소를 할당할 수 있습니다. 계정이 EC2-Classic을 지원할 경우 EC2-Classic 또는 EC2-VPC에서 사용할 주소를 할당할 수 있습니다.

콘솔을 사용하여 EC2-VPC에서 사용할 탄력적 IP 주소를 할당하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Elastic IPs]를 선택합니다.
3. Allocate new address를 선택합니다.
4. (EC2-Classic 계정) VPC를 선택한 다음 Allocate를 선택합니다. 확인 화면을 닫습니다.
5. (VPC 전용 계정) Allocate를 선택하고 확인 화면을 닫습니다.

콘솔을 사용하여 EC2-Classic에서 사용할 탄력적 IP 주소를 할당하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Elastic IPs]를 선택합니다.
3. Allocate new address를 선택합니다.
4. Classic을 선택한 후 Allocate를 선택합니다. 확인 화면을 닫습니다.

명령줄을 사용하여 탄력적 IP 주소를 할당하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [allocate-address \(AWS CLI\)](#)

- [New-EC2Address](#)(Windows PowerShell용 AWS 도구)

탄력적 IP 주소 설명

Amazon EC2이나 명령줄을 사용하여 탄력적 IP 주소를 설명할 수 있습니다.

콘솔을 사용하여 탄력적 IP 주소를 설명하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Elastic IPs]를 선택합니다.
3. [Resource Attribute] 목록에서 필터를 선택하여 검색을 시작합니다. 단일 검색에 여러 필터를 사용할 수 있습니다.

명령줄을 사용하여 탄력적 IP 주소를 설명하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [describe-addresses](#)(AWS CLI)
- [Get-EC2Address](#)(Windows PowerShell용 AWS 도구)

실행 중인 인스턴스와 탄력적 IP 주소 연결

Amazon EC2 콘솔이나 명령줄을 사용하여 인스턴스에 탄력적 IP 주소를 연결할 수 있습니다.

(VPC만 해당) 인스턴스와 탄력적 IP 주소를 연결하여 인터넷과 통신을 활성화하는 경우 인스턴스가 퍼블릭 서브넷에 위치해야 합니다. 자세한 내용은 [Amazon VPC 사용 설명서](#)의 Internet Gateways 섹션을 참조하십시오.

콘솔을 사용하여 인스턴스와 탄력적 IP 주소를 연결하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Elastic IPs]를 선택합니다.
3. 탄력적 IP 주소를 선택하고 Actions를 선택한 후 Associate address를 선택합니다.
4. Instance에서 인스턴스를 선택한 후 Associate를 선택합니다.

명령줄을 사용하여 탄력적 IP 주소를 연결하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [associate-address](#)(AWS CLI)
- [Register-EC2Address](#)(Windows PowerShell용 AWS 도구)

탄력적 IP 주소의 연결 해제 후 다른 인스턴스와 재연결

탄력적 IP 주소는 연결 해제 후 Amazon EC2 콘솔 또는 명령줄을 사용해 다시 연결할 수 있습니다.

콘솔을 사용하여 탄력적 IP 주소의 연결을 해제한 후 다시 연결하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.

2. 탐색 창에서 [Elastic IPs]를 선택합니다.
3. 탄력적 IP 주소를 선택하고 Actions를 선택한 후 Disassociate address를 선택합니다.
4. Disassociate address를 선택합니다.
5. 이전 단계에서 연결을 해제한 주소를 선택합니다. Actions에서 Associate address를 선택합니다.
6. Instance에서 새 인스턴스를 선택한 후 Associate를 선택합니다.

명령줄을 사용하여 탄력적 IP 주소 연결을 끊으려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [disassociate-address](#)(AWS CLI)
- [Unregister-EC2Address](#)(Windows PowerShell용 AWS 도구)

명령줄을 사용하여 탄력적 IP 주소를 연결하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [associate-address](#)(AWS CLI)
- [Register-EC2Address](#)(Windows PowerShell용 AWS 도구)

탄력적 IP 주소 이동

Amazon EC2 콘솔 또는 Amazon VPC 콘솔을 사용하여 EC2-Classic의 탄력적 IP 주소를 EC2-VPC로 이동할 수 있습니다. 이 옵션은 계정이 EC2-Classic을 지원하는 경우에만 사용 가능합니다.

Amazon EC2 콘솔을 사용하여 탄력적 IP 주소를 EC2-VPC로 이동하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Elastic IPs]를 선택합니다.
3. 탄력적 IP 주소를 선택하고 Actions, Move to VPC scope를 선택합니다.
4. 확인 대화 상자가 나타나면 Move Elastic IP를 선택합니다.

Amazon EC2 콘솔 또는 Amazon VPC 콘솔을 사용하여 탄력적 IP 주소를 EC2-Classic으로 복구할 수 있습니다.

Amazon EC2 콘솔을 사용하여 탄력적 IP 주소를 EC2-Classic으로 복구하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Elastic IPs]를 선택합니다.
3. 탄력적 IP 주소를 선택하고 Actions, Restore to EC2 scope를 선택합니다.
4. 확인 대화 상자가 나타나면 Restore를 선택합니다.

탄력적 IP 주소를 마이그레이션하거나 복구하는 데 명령을 사용한 경우에는 마이그레이션 프로세스가 몇 분 걸릴 수 있습니다. 이때는 [describe-moving-addresses](#) 명령을 사용하면 탄력적 IP 주소가 아직 마이그레이션 중인지, 혹은 마이그레이션이 완료되었는지 확인할 수 있습니다.

탄력적 IP 주소를 EC2-VPC로 이동한 후 Allocation ID 필드의 Elastic IPs 페이지에서 할당 ID를 확인할 수 있습니다.

탄력적 IP 주소를 이동할 때 5분이 지나도 완료되지 않으면 <https://aws.amazon.com/premiumsupport/>에 문의하십시오.

Amazon EC2 Query API 또는 AWS CLI를 사용하여 탄력적 IP 주소를 이동하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [move-address-to-vpc\(AWS CLI\)](#)
- [MoveAddressToVpc\(Amazon EC2 Query API\)](#)
- [Move-EC2AddressToVpc\(Windows PowerShell용 AWS 도구\)](#)

Amazon EC2 Query API 또는 AWS CLI를 사용하여 탄력적 IP 주소를 EC2-Classic으로 복구하는 방법

다음 명령 중 하나를 사용할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [restore-address-to-classic\(AWS CLI\)](#)
- [RestoreAddressToClassic\(Amazon EC2 Query API\)](#)
- [Restore-EC2AddressToClassic\(Windows PowerShell용 AWS 도구\)](#)

Amazon EC2 Query API 또는 AWS CLI를 사용하여 마이그레이션 주소의 상태를 나타내는 방법

다음 명령 중 하나를 사용할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [describe-moving-addresses\(AWS CLI\)](#)
- [DescribeMovingAddresses\(Amazon EC2 Query API\)](#)
- [Get-EC2Address\(Windows PowerShell용 AWS 도구\)](#)

EC2-VPC에서 마이그레이션된 탄력적 IP 주소의 할당 ID를 검색하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [describe-addresses\(AWS CLI\)](#)
- [DescribeAddresses\(Amazon EC2 Query API\)](#)
- [Get-EC2Address\(Windows PowerShell용 AWS 도구\)](#)

탄력적 IP 주소 해제

더 이상 필요 없는 탄력적 IP 주소는 연결을 해제하는 것이 좋습니다(이 주소는 인스턴스와 연결할 수 없음). EC2-Classic에서 사용하기 위해 할당한 탄력적 IP 주소에 대한 요금은 발생하지만 인스턴스와 연결된 주소에 대해서는 요금이 발생하지 않습니다.

Amazon EC2 콘솔이나 명령줄을 사용하여 탄력적 IP 주소를 릴리스할 수 있습니다.

콘솔을 사용하여 탄력적 IP 주소를 릴리스하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.

2. 탐색 창에서 [Elastic IPs]를 선택합니다.
3. 탄력적 IP 주소를 선택하고 Actions를 선택한 후 Release addresses를 선택합니다. 메시지가 나타나면 Release를 선택합니다.

명령줄을 사용하여 탄력적 IP 주소를 릴리스하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [release-address\(AWS CLI\)](#)
- [Remove-EC2Address\(Windows PowerShell용 AWS 도구\)](#)

이메일 애플리케이션에 역방향 DNS 사용

인스턴스에서 타사에 이메일을 보내려는 경우 탄력적 IP 주소를 하나 이상 프로비저닝하고 당사로 제공하는 것이 좋습니다. AWS는 ISP 및 인터넷 스팸 방지 기관과의 공동 작업을 통해 이러한 주소에서 보내는 이메일에 스팸으로 플래그가 지정될 가능성을 줄입니다.

또한 이메일을 보내는 데 사용되는 탄력적 IP 주소에 고정 역방향 DNS 레코드를 배정하면 스팸 방지 기관에서 이메일에 스팸으로 플래그를 지정하는 것을 방지할 수 있습니다. 탄력적 IP 주소를 가리키는 정방향 DNS 레코드(레코드 유형 A)가 있어야 역방향 DNS 레코드를 만들 수 있습니다.

역방향 DNS 레코드가 탄력적 IP 주소와 연결되어 있는 경우 탄력적 IP 주소는 사용자 계정에 고정됩니다. 따라서 계정에서 탄력적 IP 주소를 해제하려면 해당 레코드를 제거해야 합니다.

이메일 전송 제한을 제거하거나 탄력적 IP 주소와 역방향 DNS 레코드를 제공하려면 [이메일 전송 제한 제거 요청](#) 페이지로 이동하십시오.

탄력적 IP 주소 제한

퍼블릭(IPv4) 인터넷 주소는 흔치 않은 퍼블릭 리소스이기 때문에 기본적으로 모든 AWS 계정은 리전당 5개의 탄력적 IP 주소로 제한됩니다. 인스턴스 장애 시 주소를 다른 인스턴스로 다시 매핑하는 기능이 필요할 때는 탄력적 IP 주소를 주로 사용하고, 다른 모든 노드 간 통신에는 DNS 호스트 이름을 사용하는 것이 좋습니다.

사용 중인 아키텍처에서 추가 탄력적 IP 주소를 보증하는 경우 [Amazon EC2 탄력적 IP 주소 요청 양식](#)을 작성하십시오. 추가 주소가 필요한 이유를 파악하기 위한 사용 사례 설명을 요청 받으실 수 있습니다.

탄력적 네트워크 인터페이스

탄력적 네트워크 인터페이스(이 문서에서는 네트워크 인터페이스로 표시)는 VPC의 인스턴스에 연결할 수 있는 가상 네트워크 인터페이스입니다. 네트워크 인터페이스는 VPC에서 실행되는 인스턴스에서만 사용할 수 있습니다.

네트워크 인터페이스에는 다음 속성이 포함될 수 있습니다.

- 기본 프라이빗 IPv4 주소
- 한 개 이상의 보조 프라이빗 IPv4 주소
- 프라이빗 IPv4 주소당 한 개의 탄력적 IP 주소(IPv4)
- 한 개의 퍼블릭 IPv4 주소

- 한 개 이상의 IPv6 주소
- 하나 이상의 보안 그룹
- MAC 주소
- 원본/대상 확인 플래그
- 설명

네트워크 인터페이스를 만들고, 인스턴스에 연결하고, 인스턴스에서 분리한 후 다른 인스턴스에 연결할 수 있습니다. 인스턴스에 연결하거나 분리한 후 다른 인스턴스에 다시 연결하면 네트워크 인터페이스의 속성이 해당 네트워크 인터페이스를 따릅니다. 네트워크 인터페이스를 인스턴스 간에 이동하면 네트워크 트래픽이 새 인스턴스로 리디렉션됩니다.

VPC의 모든 인스턴스는 기본 네트워크 인터페이스(eth0)라는 기본 네트워크 인터페이스를 갖습니다. 주 네트워크 인터페이스는 인스턴스에서 분리할 수 없습니다. 추가 네트워크 인터페이스를 만들고 연결할 수 있습니다. 사용 가능한 최대 네트워크 인터페이스 수는 인스턴스 유형에 따라 다릅니다. 자세한 내용은 [인스턴스 유형별/네트워크 인터페이스당 IP 주소 \(p. 514\)](#) 섹션을 참조하십시오.

네트워크 인터페이스용 프라이빗 IPv4 주소

인스턴스의 기본 네트워크 인터페이스는 VPC의 IPv4 주소 범위에 속하는 기본 프라이빗 IPv4 주소에 할당됩니다. 사용자는 네트워크 인터페이스에 추가 프라이빗 IPv4 주소를 할당할 수 있습니다.

네트워크 인터페이스용 퍼블릭 IPv4 주소

VPC에서 모든 서브넷은 해당 서브넷에서 생성된(따라서 인스턴스가 그 서브넷으로 시작된) 네트워크 인터페이스가 퍼블릭 IPv4 주소에 할당될 것인지 결정하는 수정 가능한 속성을 갖습니다. 자세한 내용은 Amazon VPC 사용 설명서의 [서브넷에 대한 IP 주소 지정 동작](#)을 참조하십시오. 퍼블릭 IPv4 주소는 Amazon의 퍼블릭 IPv4 주소풀에서 할당됩니다. 인스턴스를 시작하면 생성된 기본 네트워크 인터페이스(eth0)에 IP 주소가 할당됩니다.

네트워크 인터페이스를 생성할 때 네트워크 인터페이스는 서브넷에서 퍼블릭 IPv4 주소 지정 속성을 상속합니다. 이후에 서브넷의 퍼블릭 IPv4 주소 지정 속성을 수정하면 네트워크 인터페이스는 처음 생성될 때 적용된 설정을 그대로 유지합니다. 인스턴스를 시작하고 eth0에 대해 기존 네트워크 인터페이스를 지정하는 경우 퍼블릭 IPv4 주소 지정 속성은 네트워크 인터페이스에 따라 결정됩니다.

자세한 내용은 [퍼블릭 IPv4 주소 및 외부 DNS 호스트 이름 \(p. 491\)](#) 섹션을 참조하십시오.

네트워크 인터페이스용 IPv6 주소

IPv6 CIDR 블록을 VPC 및 서브넷에 연결하고 서브넷 범위에 속하는 하나 이상의 IPv6 주소를 네트워크 인터페이스에 할당할 수 있습니다.

모든 서브넷은 해당 서브넷에서 생성된(따라서 인스턴스가 그 서브넷으로 시작된) 네트워크 인터페이스가 서브넷 범위에 속하는 IPv6 주소에 자동으로 할당될 것인지 결정하는 수정 가능한 속성을 갖습니다. 자세한 내용은 Amazon VPC 사용 설명서의 [서브넷에 대한 IP 주소 지정 동작](#)을 참조하십시오. 인스턴스를 시작하면 생성된 기본 네트워크 인터페이스(eth0)에 IPv6 주소가 할당됩니다.

자세한 내용은 [IPv6 주소 \(p. 492\)](#) 섹션을 참조하십시오.

목차

- [인스턴스 유형별/네트워크 인터페이스당 IP 주소 \(p. 514\)](#)
- [네트워크 인터페이스 시나리오 \(p. 517\)](#)
- [네트워크 인터페이스 구성 모범 사례 \(p. 517\)](#)
- [ec2-net-utils를 사용하여 네트워크 인터페이스 구성 \(p. 518\)](#)
- [네트워크 인터페이스 작업 \(p. 519\)](#)

인스턴스 유형별/네트워크 인터페이스당 IP 주소

다음 표에는 인스턴스 유형별 최대 네트워크 인터페이스 수와 네트워크 인터페이스당 최대 프라이빗 IPv4 주소 및 IPv6 주소 수가 나열되어 있습니다. IPv6 주소 제한은 네트워크 인터페이스당 프라이빗 IPv4 주소 제한과 별개입니다. 모든 인스턴스 유형이 IPv6 주소 지정을 지원하는 것은 아닙니다. 네트워크 인터페이스, 여러 프라이빗 IPv4 주소, IPv6 주소는 VPC에서 실행 중인 인스턴스에만 사용할 수 있습니다. 자세한 내용은 [다중 IP 주소 \(p. 498\)](#) 섹션을 참조하십시오. VPC에서 IPv6에 대한 자세한 내용은 Amazon VPC 사용 설명서의 [VPC에서 IP 주소 지정](#) 섹션을 참조하십시오.

인스턴스 유형	최대 네트워크 인터페이스	인터넷페이스당 IPv4 주소 수	인터넷페이스당 IPv6 주소 수
c1.medium	2	6	IPv6는 지원되지 않습니다
c1.xlarge	4	15	IPv6는 지원되지 않습니다
c3.large	3	10	10
c3.xlarge	4	15	15
c3.2xlarge	4	15	15
c3.4xlarge	8	30	30
c3.8xlarge	8	30	30
c4.large	3	10	10
c4.xlarge	4	15	15
c4.2xlarge	4	15	15
c4.4xlarge	8	30	30
c4.8xlarge	8	30	30
cc2.8xlarge	8	30	IPv6는 지원되지 않습니다
cg1.4xlarge	8	30	IPv6는 지원되지 않습니다
cr1.8xlarge	8	30	IPv6는 지원되지 않습니다
d2.xlarge	4	15	15
d2.2xlarge	4	15	15
d2.4xlarge	8	30	30
d2.8xlarge	8	30	30
g2.2xlarge	4	15	IPv6는 지원되지 않습니다
g2.8xlarge	8	30	IPv6는 지원되지 않습니다

Amazon Elastic Compute Cloud
 Linux 인스턴스용 사용 설명서
 인스턴스 유형별/네트워크 인터페이스당 IP 주소

인스턴스 유형	최대 네트워크 인터페이스	인터페이스당 IPv4 주소 수	인터페이스당 IPv6 주소 수
hi1.4xlarge	8	30	IPv6는 지원되지 않습니다
hs1.8xlarge	8	30	IPv6는 지원되지 않습니다
i2.xlarge	4	15	15
i2.2xlarge	4	15	15
i2.4xlarge	8	30	30
i2.8xlarge	8	30	30
i3.large	3	10	10
i3.xlarge	4	15	15
i3.2xlarge	4	15	15
i3.4xlarge	8	30	30
i3.8xlarge	8	30	30
i316xlarge	15	50	50
m1.small	2	4	IPv6는 지원되지 않습니다
m1.medium	2	6	IPv6는 지원되지 않습니다
m1.large	3	10	IPv6는 지원되지 않습니다
m1.xlarge	4	15	IPv6는 지원되지 않습니다
m2.xlarge	4	15	IPv6는 지원되지 않습니다
m2.2xlarge	4	30	IPv6는 지원되지 않습니다
m2.4xlarge	8	30	IPv6는 지원되지 않습니다
m3.medium	2	6	IPv6는 지원되지 않습니다
m3.large	3	10	IPv6는 지원되지 않습니다
m3.xlarge	4	15	IPv6는 지원되지 않습니다
m3.2xlarge	4	30	IPv6는 지원되지 않습니다

Amazon Elastic Compute Cloud
 Linux 인스턴스용 사용 설명서
 인스턴스 유형별/네트워크 인터페이스당 IP 주소

인스턴스 유형	최대 네트워크 인터페이스	인터넷 인터페이스당 IPv4 주소 수	인터넷 인터페이스당 IPv6 주소 수
m4.large	2	10	10
m4.xlarge	4	15	15
m4.2xlarge	4	15	15
m4.4xlarge	8	30	30
m4.10xlarge	8	30	30
m4.16xlarge	8	30	30
p2.xlarge	4	15	15
p2.8xlarge	8	30	30
p2.16xlarge	8	30	30
r3.large	3	10	10
r3.xlarge	4	15	15
r3.2xlarge	4	15	15
r3.4xlarge	8	30	30
r3.8xlarge	8	30	30
r4.large	3	10	10
r4.xlarge	4	15	15
r4.2xlarge	4	15	15
r4.4xlarge	8	30	30
r4.8xlarge	8	30	30
r4.16xlarge	15	50	50
t1.micro	2	2	IPv6는 지원되지 않습니다
t2.nano	2	2	2
t2.micro	2	2	2
t2.small	2	4	4
t2.medium	3	6	6
t2.large	3	12	12
t2.xlarge	3	15	15
t2.2xlarge	3	15	15
x1.16xlarge	8	30	30
x1.32xlarge	8	30	30

네트워크 인터페이스 시나리오

다음을 수행하려는 경우 여러 네트워크 인터페이스를 하나의 인스턴스에 연결하면 유용합니다.

- 관리 네트워크 생성
- VPC에서 네트워크 및 보안 어플라이언스 사용
- 별도의 서브넷에 워크로드/역할이 있는 이중 훔 인스턴스 생성
- 저예산 고가용성 솔루션 생성

관리 네트워크 생성

네트워크 인터페이스를 사용하여 관리 네트워크를 생성할 수 있습니다. 이 시나리오에서는 인스턴스의 부 네트워크 인터페이스가 퍼블릭 트래픽을 처리하고 주 네트워크 인터페이스가 백엔드 관리 트래픽을 처리합니다. 또한 주 네트워크 인터페이스는 VPC에서 액세스 제어가 더욱 제한적인 별도의 서브넷에 연결됩니다. 로드 밸런서를 지원하거나 지원하지 않을 수 있는 퍼블릭 인터페이스에 인터넷에서 서버에 액세스할 수 있도록 허용하는(예: 로드 밸런서 또는 0.0.0.0/0의 TCP 포트 80 및 443) 보안 그룹이 연결되는 반면, 프라이빗 인터페이스에는 VPC, 인터넷 또는 가상 프라이빗 게이트웨이 내 프라이빗 서브넷 안에 속하는 허용 IP 주소 범위에서만 SSH 액세스를 허용하는 보안 그룹이 연결됩니다.

장애 조치 기능을 유지하려면 네트워크 인터페이스에서 유입 트래픽에 대해 보조 프라이빗 IPv4 사용을 고려해 보십시오. 인스턴스 장애 발생 시 인터페이스 및/또는 보조 프라이빗 IPv4 주소를 스탠바이 인스턴스로 이동할 수 있습니다.

VPC에서 네트워크 및 보안 어플라이언스 사용

로드 밸런서, 네트워크 주소 변환(NAT) 서버 및 프록시 서버와 같은 일부 네트워크 및 보안 어플라이언스는 여러 네트워크 인터페이스로 구성하는 것이 좋습니다. 이러한 유형의 애플리케이션을 실행하는 부 네트워크 인터페이스를 VPC에서 생성 및 연결한 후 이 추가 인터페이스를 고유의 퍼블릭 및 프라이빗 IP 주소, 보안 그룹 및 원본/대상 확인으로 구성할 수 있습니다.

작업/역할이 개별 서브넷에 지정된 이중 훔 인스턴스 생성

애플리케이션 서버가 있는 중간 티어 네트워크에 연결되는 각각의 웹 서버에 네트워크 인터페이스를 배치할 수 있습니다. 애플리케이션 서버를 데이터베이스 서버가 있는 백엔드 네트워크(서브넷)에 이중 훔 상태로 연결할 수 있습니다. 이중 훔 인스턴스를 통한 라우팅 네트워크 패킷 대신 각 이중 훔 인스턴스가 프런트 엔드에서 요청을 수신 및 처리하고, 백엔드에 대한 연결을 초기화한 다음 백엔드 네트워크의 서버에 요청을 보냅니다.

저예산 고가용성 솔루션 생성

특정 기능을 제공하는 인스턴스 중 하나에 장애가 발생할 경우 서비스를 신속하게 복구하기 위해 관련 네트워크 인터페이스를 동일한 역할로 미리 구성된 대체 또는 핫 스탠바이 인스턴스에 연결할 수 있습니다. 예를 들어, 데이터베이스 인스턴스 또는 NAT 인스턴스와 같은 중요한 서비스에 대한 기본 또는 보조 네트워크 인터페이스로 네트워크 인터페이스를 사용할 수 있습니다. 인스턴스가 작동하지 않는 경우 사용자 또는 사용자를 대신하는 실행 중인 코드는 네트워크 인터페이스를 핫 스탠바이 인스턴스에 연결할 수 있습니다. 인터페이스에서 프라이빗 IP 주소, 탄력적 IP 주소 및 MAC 주소를 관리하므로 네트워크 인터페이스를 대체 인스턴스에 연결하자마자 네트워크 트래픽이 스탠바이 인스턴스로 전달되기 시작합니다. 인스턴스에 장애가 발생한 시간과 네트워크 인터페이스가 대기 인스턴스에 연결되는 시간 사이에 잠시 연결이 끊기지만 VPC 라우팅 테이블 또는 DNS 서버에 대해 어떠한 변경도 수행할 필요가 없습니다.

네트워크 인터페이스 구성 모범 사례

- 실행 중 상태(핫 연결), 중지 상태(웜 연결) 또는 시작 중 상태(콜드 연결)의 인터페이스에 네트워크 인터페이스를 연결할 수 있습니다.

- 인스턴스가 실행 중이거나 중지된 경우 부(ethN) 네트워크 인터페이스를 분리할 수 있습니다. 그러나 주(eth0) 인터페이스는 분리할 수 없습니다.
- 한 서브넷의 네트워크 인터페이스를 동일 VPC에 있는 다른 서브넷의 인스턴스에 연결할 수 있지만, 네트워크 인터페이스와 인스턴스가 둘 다 동일 가용 영역 안에 상주해야 합니다.
- CLI 또는 API에서 인스턴스를 시작할 때 주(eth0) 및 추가 네트워크 인터페이스 모두에 대해 인스턴스에 연결할 네트워크 인터페이스를 지정할 수 있습니다.
- 여러 네트워크 인터페이스를 포함하는 Amazon Linux 또는 Windows Server 인스턴스를 시작하면 인스턴스의 운영 체제에서 인터페이스, 프라이빗 IPv4 주소 및 라우팅 테이블이 자동으로 구성됩니다.
- 추가 네트워크 인터페이스의 웜 연결 또는 핫 연결을 수행하려면 수동으로 두 번째 인터페이스를 가동하고 프라이빗 IPv4 주소를 구성하며 라우팅 테이블을 그에 맞게 수정해야 합니다. Amazon Linux 또는 Windows Server를 실행하는 인스턴스는 웜 또는 핫 연결을 자동으로 인식하여 자체적으로 구성됩니다.
- 인스턴스에 다른 네트워크 인터페이스를 연결(예: NIC 팀 구성)하는 방법으로 이중 홈 인스턴스로 송/수신되는 네트워크 대역폭을 높이거나 두 배로 늘릴 수 없습니다.
- 동일한 서브넷에서 2개 이상의 네트워크 인터페이스를 인스턴스에 연결하면 비대칭 라우팅과 같은 네트워킹 문제가 발생할 수 있습니다. 가능한 한 기본 네트워크 인터페이스에서 보조 프라이빗 IPv4 주소를 대신 사용하십시오. 자세한 내용은 [보조 프라이빗 IPv4 주소 할당 \(p. 499\)](#) 섹션을 참조하십시오.

ec2-net-utils를 사용하여 네트워크 인터페이스 구성

Amazon Linux AMI에는 AWS에 의해 설치된 ec2-net-utils라는 추가 스크립트가 포함되어 있을 수 있습니다. 이러한 스크립트는 네트워크 인터페이스의 구성을 선택적으로 구성합니다. 이 스크립트는 Amazon Linux 전용입니다.

아직 설치되어 있지 않으면 다음 명령을 사용하여 Amazon Linux에 패키지를 설치합니다. 설치되어 있고 추가 업데이트가 가능한 경우에는 업데이트합니다.

```
$ yum install ec2-net-utils
```

다음 구성 요소는 ec2-net-utils의 일부입니다.

udev 규칙(/etc/udev/rules.d)

실행 중인 인스턴스에 연결, 분리 또는 다시 연결될 때 네트워크 인터페이스를 식별하며 핫플러그 스크립트(53-ec2-network-interfaces.rules)가 실행되도록 합니다. MAC 주소를 디바이스 이름(70-persistent-net.rules를 생성하는 75-persistent-net-generator.rules)에 매핑합니다.

핫플러그 스크립트

DHCP(/etc/sysconfig/network-scripts/ifcfg-ethN)와 함께 사용하기에 적합한 인터페이스 구성 파일을 생성합니다. 또한 라우팅 구성 파일(/etc/sysconfig/network-scripts/route-ethN)도 생성합니다.

DHCP 스크립트

네트워크 인터페이스에서 새 DHCP 임대를 수신할 때마다 이 스크립트는 인스턴스 메타데이터에 엘라스틱 IP 주소를 쿼리합니다. 각 엘라스틱 IP 주소마다 라우팅 정책 데이터베이스에 규칙을 추가하여 해당 주소의 아웃바운드 트래픽에 올바른 네트워크 인터페이스가 사용되도록 합니다. 또한 각 프라이빗 IP 주소를 네트워크 인터페이스에 부 주소로 추가합니다.

ec2ifup ethN

스탠다드 ifup의 기능을 확장합니다. 이 스크립트는 구성 파일 ifcfg-ethN 및 route-ethN을 다시 쓰 후 ifup를 실행합니다.

ec2ifdown ethN

스탠다드 ifdown의 기능을 확장합니다. 이 스크립트는 라우팅 정책 데이터베이스에서 네트워크 인터페이스 관련 규칙을 모두 제거한 후 ifdown을 실행합니다.

ec2ifscan

구성되지 않은 네트워크 인터페이스가 있는지 확인하고 이러한 인터페이스를 구성합니다.

이 스크립트는 ec2-net-utils의 초기 릴리스에서는 사용할 수 없습니다.

ec2-net-utils에서 생성된 구성 파일을 나열하려면 다음 명령을 사용합니다.

```
$ ls -l /etc/sysconfig/network-scripts/*-eth?
```

인스턴스별 자동화를 사용하지 않으려는 경우 EC2SYNC=yes를 ifcfg-ethN 파일에 추가할 수 있습니다. 예를 들어, 다음 명령을 사용하여 eth1 인터페이스에 대한 자동화를 사용하지 않도록 설정합니다.

```
$ sed -i -e 's/^EC2SYNC=yes/EC2SYNC=no/' /etc/sysconfig/network-scripts/ifcfg-eth1
```

자동화를 완전히 사용하지 않으려면 다음 명령을 사용하여 패키지를 제거할 수 있습니다.

```
$ yum remove ec2-net-utils
```

네트워크 인터페이스 작업

Amazon EC2 콘솔을 사용하여 네트워크 인터페이스 작업을 수행할 수 있습니다.

목차

- [네트워크 인터페이스 생성 \(p. 519\)](#)
- [네트워크 인터페이스 삭제 \(p. 520\)](#)
- [네트워크 인터페이스에 대한 세부 정보 보기 \(p. 520\)](#)
- [IP 트래픽 모니터링 \(p. 521\)](#)
- [인스턴스 시작 시 네트워크 인터페이스 연결 \(p. 521\)](#)
- [중지되었거나 실행 중인 인스턴스에 네트워크 인터페이스 연결 \(p. 522\)](#)
- [인스턴스에서 네트워크 인터페이스 분리 \(p. 523\)](#)
 - [보안 그룹 변경 \(p. 523\)](#)
 - [원본/대상 확인 변경 \(p. 524\)](#)
 - [탄력적 IP 주소\(IPv4\) 연결 \(p. 524\)](#)
 - [탄력적 IP 주소\(IPv4\) 연결 해제 \(p. 525\)](#)
 - [IPv6 주소 할당 \(p. 525\)](#)
 - [IPv6 주소 할당 해제 \(p. 526\)](#)
 - [종료 동작 변경 \(p. 526\)](#)
 - [설명 추가 또는 편집 \(p. 526\)](#)
 - [태그 추가 또는 편집 \(p. 527\)](#)

네트워크 인터페이스 생성

Amazon EC2 콘솔 또는 명령줄을 사용하여 네트워크 인터페이스를 생성할 수 있습니다.

콘솔을 사용하여 네트워크 인터페이스를 생성하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.

2. 탐색 창에서 [Network Interfaces]를 선택합니다.
3. [Create Network Interface]를 선택합니다.
4. [Description]에 설명 이름을 입력합니다.
5. [Subnet]에서 서브넷을 선택합니다. 네트워크 인터페이스는 일단 생성되고 나면 다른 서브넷으로 옮길 수 없으며 동일 가용 영역의 인스턴스에만 네트워크 인터페이스를 연결할 수 있습니다.
6. [Private IP](또는 [IPv4 Private IP])에 기본 프라이빗 IPv4 주소를 입력합니다. IPv4 주소를 지정하지 않는 경우 선택한 서브넷 내에서 사용 가능한 프라이빗 IPv4 주소가 선택됩니다.
7. (IPv6 전용) 연결된 IPv6 CIDR 블록이 있는 서브넷을 선택한 경우, 옵션으로 [IPv6 IP] 필드에서 IPv6 주소를 지정할 수 있습니다.
8. [Security groups]에서 하나 이상의 보안 그룹을 선택합니다.
9. [Yes, Create]를 선택합니다.

명령줄을 사용하여 네트워크 인터페이스를 생성하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [create-network-interface](#)(AWS CLI)
- [New-EC2NetworkInterface](#)(Windows PowerShell용 AWS 도구)

네트워크 인터페이스 삭제

삭제하기 전에 먼저 인스턴스에서 네트워크 인터페이스를 분리해야 합니다. 네트워크 인터페이스를 삭제하면 인터페이스와 연결된 모든 속성이 해제되고 다른 인스턴스에서 사용할 수 있도록 프라이빗 IP 주소나 탄력적 IP 주소가 해제됩니다.

Amazon EC2 콘솔 또는 명령줄을 사용하여 네트워크 인터페이스를 삭제할 수 있습니다.

콘솔을 사용하여 네트워크 인터페이스를 삭제하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.
3. 네트워크 인터페이스를 선택하고 [Delete]를 선택합니다.
4. [Delete Network Interface] 대화 상자에서 [Yes, Delete]를 선택합니다.

명령줄을 사용하여 네트워크 인터페이스를 삭제하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [delete-network-interface](#) (AWS CLI)
- [Remove-EC2NetworkInterface](#)(Windows PowerShell용 AWS 도구)

네트워크 인터페이스에 대한 세부 정보 보기

Amazon EC2 콘솔 또는 명령줄을 사용하여 네트워크 인터페이스를 설명할 수 있습니다.

콘솔을 사용하여 네트워크 인터페이스를 설명하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.

3. 네트워크 인터페이스를 선택합니다.
4. [Details] 탭에서 세부 정보를 확인합니다.

명령줄을 사용하여 네트워크 인터페이스를 설명하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [describe-network-interfaces](#)(AWS CLI)
- [Get-EC2NetworkInterface](#)(Windows PowerShell용 AWS 도구)

명령줄을 사용하여 네트워크 인터페이스 속성을 설명하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [describe-network-interface-attribute](#)(AWS CLI)
- [Get-EC2NetworkInterfaceAttribute](#)(Windows PowerShell용 AWS 도구)

IP 트래픽 모니터링

네트워크 인터페이스에서 VPC 흐름 로그를 활성화하여 네트워크 인터페이스로 주고받는 IP 트래픽에 대한 정보를 캡처합니다. 흐름 로그를 생성하고 난 다음 Amazon CloudWatch Logs의 데이터를 확인하고 가져올 수 있습니다.

자세한 내용은 Amazon VPC 사용 설명서에서 [VPC 흐름 로그](#)를 참조하십시오.

인스턴스 시작 시 네트워크 인터페이스 연결

인스턴스를 시작할 때 기존 네트워크 인터페이스를 지정하거나 네트워크 인터페이스를 추가적으로 연결할 수 있습니다. 이 작업은 Amazon EC2 콘솔 또는 명령줄을 사용하여 수행할 수 있습니다.

Note

인스턴스에 네트워크 인터페이스를 연결할 때 오류가 발생하는 경우 이로 인해 인스턴스가 시작되지 않습니다.

콘솔을 사용하여 인스턴스 시작 시 네트워크 인터페이스를 연결하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. [Launch Instance]를 선택합니다.
3. AMI와 인스턴스 유형을 선택하고 [Next: Configure Instance Details]를 선택합니다.
4. [Configure Instance Details] 페이지의 [Network]에서 VPC를, [Subnet]에서 서브넷을 선택합니다.
5. 콘솔을 사용하면 [Network Interfaces] 섹션에서 인스턴스 시작 시 네트워크 인터페이스(신규, 기존 또는 결합)를 두 개까지 지정할 수 있습니다. 새 인터페이스에 대해 하나의 기본 IPv4 주소와 하나 이상의 보조 IPv4 주소를 입력할 수도 있습니다.

시작한 후에는 추가로 다른 네트워크 인터페이스를 인스턴스에 추가할 수 있습니다. 연결 가능한 총 네트워크 인터페이스의 수는 인스턴스 유형에 따라 다릅니다. 자세한 내용은 [인스턴스 유형별/네트워크 인터페이스당 IP 주소 \(p. 514\)](#) 섹션을 참조하십시오.

Note

네트워크 인터페이스를 두 개 이상 지정하면 퍼블릭 IPv4 주소를 인스턴스에 자동 할당할 수 없습니다.

6. (IPv6 전용) 연결된 IPv6 CIDR 블록이 있는 서브넷으로 인스턴스를 시작하는 경우, 연결한 모든 네트워크 인터페이스에 IPv6 주소를 지정할 수 있습니다. [IPv6 IPs]에서 [Add IP]를 선택합니다. 보조 IPv6 주소를 추가하려면 [Add IP]를 다시 선택합니다. 서브넷 범위에 속한 IPv6 주소를 직접 입력하거나, Amazon이 서브넷에 속한 IPv6 주소를 자동으로 선택하도록 기본 [Auto-assign] 값을 그대로 둘 수 있습니다.
7. [Next: Add Storage]를 선택합니다.
8. Add Storage 페이지에서 사용자는 볼륨을 지정하여 AMI에 의해 지정된 볼륨 옆에 인스턴스(루트 디바이스 볼륨 등)를 연결한 다음 Next: Add Tags를 선택합니다.
9. Add Tags 페이지에서 인스턴스에 태그(예: 사용자에게 친숙한 이름)를 지정한 후 Next: Configure Security Group을 선택합니다.
10. [Configure Security Group] 페이지에서 보안 그룹을 선택하거나 새 보안 그룹을 만들 수 있습니다. [Review and Launch]를 선택합니다.

Note

5단계에서 기존 네트워크 인터페이스를 지정한 경우, 이 단계에서 어떤 옵션을 선택하든 상관 없이 인스턴스는 그 네트워크 인터페이스에 대한 보안 그룹과 연결됩니다.

11. [Review Instance Launch] 페이지에 주 및 추가 네트워크 인터페이스에 대한 세부 정보가 표시됩니다. 설정을 검토한 다음 [Launch]를 선택하여 키 페어를 선택하고 인스턴스를 시작합니다. Amazon EC2를 처음으로 사용하는 것이고 키 쌍을 생성하지 않은 경우 새로운 키 쌍을 생성하는 메시지가 마법사에 표시됩니다.

명령줄 사용하여 인스턴스 시작 시 네트워크 인터페이스를 연결하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [run-instances\(AWS CLI\)](#)
- [New-EC2Instance\(Windows PowerShell용 AWS 도구\)](#)

중지되었거나 실행 중인 인스턴스에 네트워크 인터페이스 연결

Amazon EC2 콘솔의 [Instances] 또는 [Network Interfaces] 페이지를 사용하거나 명령줄 인터페이스를 사용하여 VPC에서 중지되었거나 실행 중인 인스턴스 중 하나에 네트워크 인터페이스를 연결할 수 있습니다.

Note

VPC 인스턴스의 퍼블릭 IP 주소가 해제되는 경우 인스턴스에 두 개 이상의 네트워크 인터페이스가 연결되어 있으면 새 퍼블릭 IPv4 주소를 받을 수 없습니다. 퍼블릭 IPv4 주소의 동작에 대한 자세한 내용은 [퍼블릭 IPv4 주소 및 외부 DNS 호스트 이름 \(p. 491\)](#) 섹션을 참조하십시오.

Instances 페이지를 사용하여 인스턴스에 네트워크 인터페이스를 연결하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. [Actions], [Networking], [Attach Network Interface]를 선택합니다.
4. [Attach Network Interface] 대화 상자에서 네트워크 인터페이스를 선택한 다음 [Attach]를 선택합니다.

Network Interfaces 페이지를 사용하여 네트워크 인터페이스를 인스턴스에 연결하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.
3. 네트워크 인터페이스를 선택하고 [Attach]를 선택합니다.
4. [Attach Network Interface] 대화 상자에서 인스턴스를 선택한 다음 [Attach]를 선택합니다.

명령줄 사용하여 인스턴스에 네트워크 인터페이스를 연결하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [attach-network-interface](#)(AWS CLI)
- [Add-EC2NetworkInterface](#)(Windows PowerShell용 AWS 도구)

인스턴스에서 네트워크 인터페이스 분리

Amazon EC2 콘솔의 [Instances] 또는 [Network Interfaces] 페이지를 사용하거나 명령줄 인터페이스를 사용하여 언제라도 보조 네트워크 인터페이스를 분리할 수 있습니다.

Instances 페이지를 사용하여 인스턴스에서 네트워크 인터페이스를 분리하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. [Actions], [Networking], [Detach Network Interface]를 선택합니다.
4. [Detach Network Interface] 대화 상자에서 네트워크 인터페이스를 선택하고 [Detach]를 선택합니다.

[Network Interfaces] 페이지를 사용하여 인스턴스에서 네트워크 인터페이스를 분리하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.
3. 네트워크 인터페이스를 선택하고 [Detach]를 선택합니다.
4. [Detach Network Interface] 대화 상자에서 [Yes, Detach]를 선택합니다. 네트워크 인터페이스가 인스턴스에서 분리되지 않으면 [Force detachment]를 선택하고 다시 시도합니다.

명령줄을 사용하여 네트워크 인터페이스를 분리하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [detach-network-interface](#)(AWS CLI)
- [Dismount-EC2NetworkInterface](#)(Windows PowerShell용 AWS 도구)

보안 그룹 변경

네트워크 인터페이스와 연결된 보안 그룹을 변경할 수 있습니다. 보안 그룹을 생성할 때는 인터페이스의 서브넷과 동일한 VPC를 지정해야 합니다.

Amazon EC2 콘솔 또는 명령줄을 사용하여 네트워크 인터페이스에 대한 보안 그룹을 변경할 수 있습니다.

Note

다른 서비스에서 소유하는 인터페이스에 대한 보안 그룹 멤버십(예: Elastic Load Balancing)을 변경하려면 해당 서비스에 대한 콘솔이나 명령줄 인터페이스를 사용하십시오.

콘솔을 사용하여 네트워크 인터페이스의 보안 그룹을 변경하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.
3. 네트워크 인터페이스를 선택하고 [Actions], [Change Security Groups]를 선택합니다.

4. [Change Security Groups] 대화 상자에서 사용할 보안 그룹을 선택하고 [Save]를 선택합니다.

명령줄을 사용하여 네트워크 인터페이스의 보안 그룹을 변경하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [modify-network-interface-attribute](#)(AWS CLI)
- [Edit-EC2NetworkInterfaceAttribute](#)(Windows PowerShell용 AWS 도구)

원본/대상 확인 변경

Source/Destination Check 속성은 인스턴스에서 원본/대상 확인이 활성화/비활성화되었는지를 제어합니다. 이 속성을 비활성화하면 인스턴스에서 대상이 특별히 해당 인스턴스로 지정되지 않은 네트워크 트래픽을 처리할 수 있습니다. 예를 들어, 네트워크 주소 변환, 라우팅, 방화벽 등의 서비스를 실행 중인 인스턴스는 이 값을 `disabled`로 설정해야 합니다. 기본 값은 `enabled`입니다.

Amazon EC2 콘솔 또는 명령줄을 사용하여 원본/대상 확인을 변경할 수 있습니다.

콘솔을 사용하여 네트워크 인터페이스에 대한 원본/대상 확인을 변경하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.
3. 네트워크 인터페이스를 선택하고 [Actions], [Change Source/Dest Check]를 선택합니다.
4. 대화 상자에서 [Enabled](활성화된 경우) 또는 [Disabled](비활성화된 경우)를 선택하고 [Save]를 선택합니다.

명령줄을 사용하여 네트워크 인터페이스에 대한 원본/대상 확인을 변경하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [modify-network-interface-attribute](#)(AWS CLI)
- [Edit-EC2NetworkInterfaceAttribute](#)(Windows PowerShell용 AWS 도구)

탄력적 IP 주소(IPv4) 연결

탄력적 IP 주소(IPv4)가 있는 경우 이 주소를 네트워크 인터페이스에 대한 프라이빗 IPv4 주소 중 하나와 연결할 수 있습니다. 한 탄력적 IP 주소를 각 프라이빗 IPv4 주소와 연결할 수 있습니다.

Amazon EC2 콘솔이나 명령줄을 사용하여 탄력적 IP 주소를 연결할 수 있습니다.

콘솔을 사용하여 탄력적 IP 주소를 연결하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.
3. 네트워크 인터페이스를 선택하고 [Actions], [Associate Address]를 선택합니다.
4. [Associate Elastic IP Address] 대화 상자의 [Address] 목록에서 탄력적 IP 주소를 선택합니다.
5. [Associate to private IP address]에서 탄력적 IP 주소와 연결할 프라이빗 IPv4 주소를 선택합니다.
6. [Allow reassociation]를 선택하여 탄력적 IP 주소가 현재 다른 인스턴스나 네트워크 인터페이스와 연결되어 있는 경우 지정된 네트워크 인터페이스와 연결될 수 있도록 한 다음 [Associate Address]를 선택합니다.

명령줄을 사용하여 탄력적 IP 주소를 연결하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [associate-address\(AWS CLI\)](#)
- [Register-EC2Address\(Windows PowerShell용 AWS 도구\)](#)

탄력적 IP 주소(IPv4) 연결 해제

네트워크 인터페이스에 연결된 탄력적 IP 주소(IPv4)가 있는 경우 해당 주소를 분리하고 다른 네트워크 인터페이스와 연결하거나 해제하여 주소 풀로 반환합니다. 네트워크 인터페이스는 특정 서브넷에서 고유하므로 네트워크 인터페이스를 사용하여 다른 서브넷이나 VPC의 인스턴스와 엘라스틱 IP 주소를 연결하는 방법은 이 방법뿐입니다.

Amazon EC2 콘솔이나 명령줄을 사용하여 탄력적 IP 주소를 분리할 수 있습니다.

콘솔을 사용하여 탄력적 IP 주소 연결을 끊으려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.
3. 네트워크 인터페이스를 선택하고 [Actions], [Disassociate Address]를 선택합니다.
4. [Disassociate IP Address] 대화 상자에서 [Yes, Disassociate]를 선택합니다.

명령줄을 사용하여 탄력적 IP 주소 연결을 끊으려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [disassociate-address\(AWS CLI\)](#)
- [Unregister-EC2Address\(Windows PowerShell용 AWS 도구\)](#)

IPv6 주소 할당

사용자는 네트워크 인터페이스에 하나 이상의 IPv6 주소를 할당할 수 있습니다. 네트워크 인터페이스는 연결된 IPv6 CIDR 블록이 있는 서브넷에 속해야 합니다. 네트워크 인터페이스에 특정 IPv6 주소를 할당하려면 IPv6 주소에 이미 다른 네트워크 인터페이스가 할당되어 있어서는 안 됩니다.

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Network Interfaces]를 선택한 후 네트워크 인터페이스를 선택합니다.
3. [Actions], [Manage IP Addresses]를 선택합니다.
4. [IPv6 Addresses]에서 [Assign new IP]를 선택합니다. 서브넷 범위에 속한 IPv6 주소를 직접 입력하거나, Amazon이 자동으로 선택하도록 기본 [Auto-assign] 값을 그대로 둡니다.
5. [Yes, Update]를 선택합니다.

명령줄을 사용하여 네트워크 인터페이스에 IPv6 주소를 할당하려면

- 다음 명령 중 하나를 사용할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.
 - [assign-ipv6-addresses\(AWS CLI\)](#)

- [Register-EC2Ipv6AddressList](#)(Windows PowerShell용 AWS 도구)

IPv6 주소 할당 해제

Amazon EC2 콘솔을 사용하여 네트워크 인터페이스에서 IPv6 주소 할당을 해제할 수 있습니다.

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Network Interfaces]를 선택한 후 네트워크 인터페이스를 선택합니다.
3. [Actions], [Manage IP Addresses]를 선택합니다.
4. [IPv6 Addresses]에서 제거할 IPv6 주소에 대해 [Unassign]을 선택합니다.
5. [Yes, Update]를 선택합니다.

명령줄을 사용하여 네트워크 인터페이스에서 IPv6 주소 할당을 해제하려면

- 다음 명령 중 하나를 사용할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.
 - [unassign-ipv6-addresses](#)(AWS CLI)
 - [Unregister-EC2Ipv6AddressList](#)(Windows PowerShell용 AWS 도구)

종료 동작 변경

연결된 인스턴스를 삭제할 때 자동으로 삭제되도록 인스턴스에 연결된 네트워크 인터페이스에 대한 종료 동작을 설정할 수 있습니다.

Note

기본적으로 콘솔을 사용하여 자동으로 생성되고 인스턴스에 연결된 네트워크 인터페이스는 인스턴스가 종료될 때 종료되도록 설정되어 있습니다. 그러나 명령줄 인터페이스를 사용하여 생성한 네트워크 인터페이스는 인스턴스가 종료될 때 종료되도록 설정되어 있지 않습니다.

Amazon EC2 콘솔 또는 명령줄을 사용하여 네트워크 인터페이스에 대한 종료 동작을 변경할 수 있습니다.

콘솔을 사용하여 네트워크 인터페이스에 대한 종료 동작을 변경하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.
3. 네트워크 인터페이스를 선택하고 [Actions], [Change Termination Behavior]를 선택합니다.
4. 인스턴스를 종료할 때 네트워크 인터페이스를 삭제하려면 [Change Termination Behavior] 대화 상자에서 [Delete on termination] 확인란을 선택합니다.

명령줄을 사용하여 네트워크 인터페이스에 대한 종료 동작을 변경하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [modify-network-interface-attribute](#)(AWS CLI)
- [Edit-EC2NetworkInterfaceAttribute](#)(Windows PowerShell용 AWS 도구)

설명 추가 또는 편집

Amazon EC2 콘솔 또는 명령줄을 사용하여 네트워크 인터페이스에 대한 설명을 변경할 수 있습니다.

콘솔을 사용하여 네트워크 인터페이스에 대한 설명을 변경하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.
3. 네트워크 인터페이스를 선택하고 [Actions], [Change Description]을 선택합니다.
4. [Change Description] 대화 상자에서 네트워크 인터페이스에 대한 설명을 입력하고 [Save]를 선택합니다.

명령줄을 사용하여 네트워크 인터페이스에 대한 설명을 변경하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [modify-network-interface-attribute](#)(AWS CLI)
- [Edit-EC2NetworkInterfaceAttribute](#)(Windows PowerShell용 AWS 도구)

태그 추가 또는 편집

태그는 네트워크 인터페이스에 추가할 수 있는 메타데이터입니다. 태그는 개인적인 정보이므로 해당 계정에만 표시됩니다. 각 태그는 키와 값(선택 사항)으로 구성됩니다. 태그에 대한 자세한 내용은 [Amazon EC2 리소스에 태그 지정 \(p. 681\)](#) 섹션을 참조하십시오.

Amazon EC2 콘솔 또는 명령줄을 사용하여 리소스에 태그를 지정할 수 있습니다.

콘솔을 사용하여 네트워크 인터페이스에 대한 태그를 추가하거나 편집하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.
3. 네트워크 인터페이스를 선택합니다.
4. 세부 정보 창에서 [Tags], [Add/Edit Tags]를 선택합니다.
5. [Add/Edit Tags] 대화 상자에서 생성할 각 태그에 대해 [Create Tag]를 선택하고 키와 값(선택 사항)을 입력합니다. 완료되면 [Save]를 선택합니다.

명령줄을 사용하여 네트워크 인터페이스에 대한 태그를 추가 또는 편집하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [create-tags](#)(AWS CLI)
- [New-EC2Tag](#)(Windows PowerShell용 AWS 도구)

배치 그룹

배치 그룹은 단일 가용 영역 내에 있는 인스턴스의 논리적 그룹입니다. 배치 그룹은 짧은 네트워크 지연 시간, 높은 네트워크 처리량 또는 둘 다의 이점을 활용할 수 있는 애플리케이션에 권장됩니다. 배치 그룹에 가장 짧은 지연 시간과 가장 높은 초당 패킷 네트워크 성능을 제공하려면 향상된 네트워킹을 지원하는 인스턴스 유형을 선택하십시오. 자세한 내용은 [향상된 네트워킹 \(p. 533\)](#)을 참조하십시오.

먼저 배치 그룹을 생성한 다음 여러 개의 인스턴스를 배치 그룹으로 시작합니다. 단일 시작 요청의 배치 그룹에서 인스턴스를 필요한 수만큼 시작하고 배치 그룹의 모든 인스턴스에 대해 동일한 인스턴스 유형을 사용하

는 것이 좋습니다. 나중에 배치 그룹에 인스턴스를 더 추가하거나 배치 그룹에서 두 가지 이상의 인스턴스 유형을 시작하려고 하면 용량 부족 오류가 발생할 가능성이 커집니다.

배치 그룹 생성은 무료입니다.

배치 그룹의 인스턴스를 중지한 후 다시 시작하면 인스턴스가 계속 배치 그룹에서 실행됩니다. 그러나 인스턴스에 대해 용량이 부족한 경우 시작에 실패합니다.

이미 인스턴스를 실행한 배치 그룹의 인스턴스를 시작할 때 용량 오류가 발생하는 경우, 배치 그룹의 모든 인스턴스를 중지하고 시작한 후 다시 실행해 보십시오. 인스턴스를 다시 시작하면 요청한 모든 인스턴스를 수용할 용량이 있는 하드웨어로 인스턴스가 마이그레이션될 수 있습니다.

목차

- [배치 그룹의 제한 사항 \(p. 528\)](#)
- [배치 그룹으로 인스턴스 시작 \(p. 529\)](#)
- [배치 그룹 삭제 \(p. 530\)](#)

배치 그룹의 제한 사항

배치 그룹에는 다음과 같은 제한 사항이 있습니다.

- 배치 그룹은 여러 가용 영역을 포괄할 수 없습니다.
- 배치 그룹에 지정하는 이름은 AWS 계정 내에서 고유해야 합니다.
- 인스턴스를 배치 그룹으로 시작할 때는 다음과 같은 인스턴스 유형만 사용할 수 있습니다.
 - 범용: m4.large | m4.xlarge | m4.2xlarge | m4.4xlarge | m4.10xlarge | m4.16xlarge
 - 컴퓨팅 최적화: c4.large | c4.xlarge | c4.2xlarge | c4.4xlarge | c4.8xlarge | c3.large | c3.xlarge | c3.2xlarge | c3.4xlarge | c3.8xlarge | cc2.8xlarge
 - 메모리 최적화: cr1.8xlarge | r3.large | r3.xlarge | r3.2xlarge | r3.4xlarge | r3.8xlarge | r4.large | r4.xlarge | r4.2xlarge | r4.4xlarge | r4.8xlarge | r4.16xlarge | x1.16xlarge | x1.32xlarge
 - 스토리지 최적화: d2.xlarge | d2.2xlarge | d2.4xlarge | d2.8xlarge | h1.4xlarge | hs1.8xlarge | i2.xlarge | i2.2xlarge | i2.4xlarge | i2.8xlarge | i3.large | i3.xlarge | i3.2xlarge | i3.4xlarge | i3.8xlarge | i3.16xlarge
 - 액셀러레이티드 컴퓨팅: cg1.4xlarge | g2.2xlarge | g2.8xlarge | p2.xlarge | p2.8xlarge | p2.16xlarge
- 한 배치 그룹에 있는 두 인스턴스 간의 최대 네트워크 처리 속도는 두 인스턴스의 속도가 느리면 저하됩니다. 많은 양을 처리해야 하는 애플리케이션의 경우 10Gbps 또는 20Gbps 네트워크 연결을 지원하는 인스턴스 유형을 선택하십시오. 인스턴스 유형 네트워크 성능에 대한 자세한 내용은 [Amazon EC2 인스턴스 유형 표](#)를 참조하십시오.
- 여러 개의 인스턴스 유형을 하나의 배치 그룹으로 시작할 수는 있지만 이렇게 하면 시작에 성공하는 데 필요한 용량이 원활하게 제공될 가능성이 낮아집니다. 배치 그룹의 모든 인스턴스에 동일한 인스턴스 유형을 사용하는 것이 좋습니다.
- 여러 배치 그룹을 병합할 수는 없습니다. 대신 배치 그룹 중 하나의 인스턴스를 종료한 후 다른 배치 그룹으로 다시 시작해야 합니다.
- 배치 그룹은 피어링된 여러 VPC를 포괄할 수 있지만, 피어링된 VPC의 인스턴스 간에는 양방향 대역폭이 제공되지 않습니다. VPC 피어링 연결에 대한 자세한 내용은 [Amazon VPC Peering Guide](#)를 참조하십시오.
- 기존 인스턴스를 배치 그룹으로 이동할 수는 없습니다. 기존 인스턴스에서 AMI를 생성한 후 AMI를 통해 새 인스턴스를 배치 그룹으로 시작할 수는 있습니다.
- 예약 인스턴스로 가용 영역에서 EC2 인스턴스의 용량을 예약합니다. 용량 예약은 배치 그룹에서 동일한 가용 영역에 할당된 인스턴스별로 사용할 수 있습니다. 하지만 배치 그룹에 대한 용량을 명시적으로 예약할 수는 없습니다.

- 네트워크 트래픽을 배치 그룹 내로 유지하려면 배치 그룹의 구성원들이 서로를 프라이빗 IPv4 주소 또는 IPv6 주소(해당하는 경우)로 참조해야 합니다. 구성원 간에 퍼블릭 IPv4 주소를 사용하는 경우 처리량이 5Gbps 이하로 떨어집니다.
- 배치 그룹 밖에 있는 리소스와의 네트워크 트래픽은 5Gbps로 제한됩니다.

배치 그룹으로 인스턴스 시작

배치 그룹으로 시작할 인스턴스에 대한 전용 AMI를 생성하는 것이 좋습니다.

콘솔을 사용하여 인스턴스를 배치 그룹으로 시작하려면

- <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
- 인스턴스에 대한 AMI를 생성합니다.
 - Amazon EC2 대시보드에서 [Launch Instance]를 선택합니다. 마법사를 완료한 후 [Launch]를 선택합니다.
 - 인스턴스에 연결합니다. 자세한 내용은 [Linux 인스턴스에 연결 \(p. 274\)](#) 섹션을 참조하십시오.
 - 인스턴스에 소프트웨어와 애플리케이션을 설치하거나, 데이터를 복사하거나, Amazon EBS 볼륨을 추가로 연결합니다.
 - (선택 사항) 인스턴스 유형이 향상된 네트워킹을 지원하는 경우 [Linux에서 향상된 네트워킹 \(p. 533\)](#)의 절차에 따라 해당 기능이 활성화되었는지 확인합니다.
 - 탐색 창에서 [Instances]를 선택하고 인스턴스를 선택한 다음, [Actions], [Image], [Create Image]를 선택합니다. [Create Image] 대화 상자에 요청된 정보를 입력한 다음 [Create Image]를 선택합니다.
 - (선택 사항) 해당 인스턴스를 더 이상 사용하지 않는 경우 종료해도 됩니다.
- 배치 그룹을 생성합니다.
 - 탐색 창에서 [Placement Groups]를 선택합니다.
 - [Create Placement Group]을 선택합니다.
 - [Create Placement Group] 대화 상자에서 사용하는 AWS 계정에 고유한 배치 그룹 이름을 입력하고 [Create]를 선택합니다.

배치 그룹의 상태가 `available`이면 인스턴스를 배치 그룹으로 시작할 수 있습니다.

- 인스턴스를 배치 그룹으로 시작합니다.
 - 탐색 창에서 [Instances]를 선택합니다.
 - [Launch Instance]를 선택합니다. 마법사의 안내에 따라 다음 작업을 주의하여 수행하십시오.
 - [Choose an Amazon Machine Image (AMI)] 페이지에서 [My AMIs] 탭을 선택하고 생성한 AMI를 선택합니다.
 - [Choose an Instance Type] 페이지에서 배치 그룹으로 실행할 인스턴스 유형을 선택합니다.
 - 나중에 배치 그룹에 인스턴스를 추가하지 못할 수 있으므로 [Configure Instance Details] 페이지에서 이 배치 그룹에 필요한 인스턴스의 총 수를 입력합니다.
 - [Configure Instance Details] 페이지의 [Placement group]에서 해당 배치 그룹을 선택합니다. 이 페이지에 [Placement group] 목록이 표시되지 않으면 배치 그룹으로 실행할 수 있는 인스턴스 유형을 선택했는지 확인합니다. 배치 그룹으로 실행할 수 없는 인스턴스 유형을 선택한 경우 이 옵션을 사용할 수 없습니다.

명령줄을 사용하여 인스턴스를 배치 그룹으로 시작하려면

- 다음 명령 중 하나를 사용하여 인스턴스에 대한 AMI를 만듭니다.
 - [create-image\(AWS CLI\)](#)

- [New-EC2Image](#)(Windows PowerShell용 AWS 도구)
2. 다음 명령 중 하나를 사용하여 배치 그룹을 만듭니다.
- [create-placement-group](#)(AWS CLI)
 - [New-EC2PlacementGroup](#)(Windows PowerShell용 AWS 도구)
3. 다음 옵션 중 하나를 사용하여 인스턴스를 배치 그룹으로 실행합니다.
- `--placement with run-instances` (AWS CLI)
 - `-PlacementGroup with New-EC2Instance` (Windows PowerShell용 AWS 도구)

배치 그룹 삭제

대체해야 하거나 더 이상 필요하지 않은 배치 그룹을 삭제할 수 있습니다. 배치 그룹을 삭제하려면 우선 배치 그룹으로 시작한 모든 인스턴스를 종료해야 합니다.

콘솔을 사용하여 배치 그룹을 삭제하려면 다음을 수행합니다.

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. 배치 그룹의 모든 인스턴스를 선택하여 종료합니다. 세부 정보 창에서 [Placement Group]의 값을 확인하면 인스턴스를 종료하기 전에 인스턴스가 해당 배치 그룹에 속하는지 확인할 수 있습니다.
4. 탐색 창에서 [Placement Groups]를 선택합니다.
5. 배치 그룹을 선택하고 [Delete Placement Group]을 선택합니다.
6. 확인 메시지가 나타나면 [Yes, Delete]를 선택합니다.

명령줄을 사용하여 배치 그룹을 삭제하려면 다음을 수행합니다.

다음 명령 세트 중 하나를 사용할 수 있습니다. 해당 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [terminate-instances](#) 및 [delete-placement-group](#)(AWS CLI)
- [Stop-EC2Instance](#) 및 [Remove-EC2PlacementGroup](#)(Windows PowerShell용 AWS 도구)

EC2 인스턴스에 대한 네트워크 MTU(최대 전송 단위)

네트워크 연결의 MTU(최대 전송 단위)는 연결을 통해 전달할 수 있는 허용되는 최대 크기의 패킷 크기(바이트)입니다. 연결의 MTU가 클수록 하나의 패킷으로 전달할 수 있는 데이터의 양이 늘어납니다. 이더넷 패킷은 프레임 또는 전송 중인 실제 데이터와 이를 둘러싼 네트워크 오버헤드 정보로 구성됩니다.

이더넷 프레임은 여러 가지 형식으로 제공될 수 있으며, 가장 일반적인 형식은 표준 이더넷 v2 프레임 형식입니다. 대부분의 인터넷에서 지원되는 최대 이더넷 패킷 크기인 1500MTU를 지원합니다. 인스턴스의 지원되는 최대 MTU는 인스턴스 유형에 따라 다릅니다. 모든 Amazon EC2 인스턴스 유형은 1500MTU를 지원하며, 현재 다수의 인스턴스 크기가 9001MTU 또는 점보 프레임을 지원합니다.

목차

- [점보 프레임\(9001 MTU\) \(p. 531\)](#)
- [경로 MTU 검색 \(p. 531\)](#)
- [두 호스트 간 경로 MTU 확인 \(p. 531\)](#)
- [Amazon EC2 인스턴스에서 MTU 확인 및 설정 \(p. 532\)](#)

- 문제 해결 (p. 533)

점보 프레임(9001 MTU)

점보 프레임에서는 패킷당 페이로드 크기를 늘려 1500바이트 이상의 데이터가 허용됩니다. 그 결과, 패킷 오버헤드에 해당하지 않는 패킷의 비율이 늘어납니다. 같은 양의 사용 가능한 데이터를 보내더라도 더 적은 수의 패킷만 있으면 됩니다. 그러나 지정된 AWS 리전(EC2-Classic), 단일 VPC 또는 VPC 피어링 연결 외부에서는 1500MTU의 최대 경로를 경험할 수 있습니다. VPN 연결 및 인터넷 게이트웨이를 통해 전송되는 트래픽은 1500 MTU로 제한됩니다. 패킷이 1500바이트인 경우, 단편화되거나 IP 헤더에 `Don't Fragment` 플래그가 설정된 경우 삭제됩니다.

인터넷 트래픽이나 VPC를 벗어나는 트래픽에 점보 프레임을 사용할 때는 주의가 필요합니다. 중간 시스템에서 패킷이 단편화되면서 트래픽이 느려지기 때문입니다. VPC 내에서 점보 프레임을 사용하고 VPC 외부의 느린 트래픽에는 사용하지 않으려면 라우팅을 기준으로 MTU 크기를 구성하거나, MTU 크기와 라우팅을 달리하여 다수의 탄력적 네트워크 인터페이스를 사용할 수도 있습니다.

배치 그룹 내부에 배치된 인스턴스의 경우, 점보 프레임이 가능한 많은 네트워크 처리량을 달성하는 데 도움을 주므로 이런 경우에 권장됩니다. 자세한 내용은 [배치 그룹 \(p. 527\)](#) 섹션을 참조하십시오.

다음 인스턴스는 점보 프레임을 지원합니다.

- 최적화된 컴퓨팅: C3, C4, CC2
- 범용: M3, M4, T2
- 액셀러레이티드 컴퓨팅: CG1, G2, P2
- 메모리 최적화: CR1, R3, R4, X1
- 최적화된 스토리지: D2, HI1, HS1, I2, I3

경로 MTU 검색

경로 MTU 검색을 사용하여 두 디바이스 간의 경로 MTU를 확인할 수 있습니다. 경로 MTU는 발신 호스트와 수신 호스트 간의 경로에서 지원되는 최대 패킷 사이즈입니다. 호스트가 수신 호스트의 MTU 또는 경로를 따라 디바이스의 MTU보다 큰 패킷을 전송하는 경우 수신 호스트 또는 디바이스가 `Destination Unreachable: Fragmentation Needed and Don't Fragment was Set`(유형 3, 코드 4)과 같은 ICMP 메시지를 반환합니다. 이는 패킷을 전송할 수 있을 때까지 MTU를 조정하도록 원본 호스트에 지시합니다.

기본적으로 보안 그룹은 인바운드 ICMP 트래픽을 허용하지 않습니다. 인스턴스가 이 메시지를 수신하고 패킷이 삭제되지 않도록 하려면 인스턴스의 인바운드 보안 그룹 규칙에 Destination Unreachable 프로토콜이 있는 사용자 지정 ICMP 규칙을 추가해야 합니다. 자세한 내용은 Amazon EC2 보안 그룹 주제에서 [보안 그룹에 규칙 추가 \(p. 390\)](#) 및 [API 및 명령 개요 \(p. 392\)](#)을 참조하십시오.

Important

경로 MTU 검색을 허용하도록 인스턴스의 보안 그룹을 수정해도 일부 라우터에서 점보 프레임이 삭제되지 않는다고 보장되지는 않습니다. VPC의 인터넷 게이트웨이는 패킷을 최대 1,500바이트까지만 전송합니다. 인터넷 트래픽에는 1500MTU 패킷이 권장됩니다.

두 호스트 간 경로 MTU 확인

tracepath 명령(Linux 배포에서 기본적으로 사용할 수 있는 `iputils` 패키지의 일부로, <http://www.elifulkerson.com/projects/mturoute.php>에서 다운로드하고 설치할 수 있는 Amazon Linux).

tracepath를 사용하여 경로 MTU를 확인하려면

- 다음 명령을 사용해 Amazon EC2 인스턴스와 다른 호스트 간에 경로 MTU를 확인합니다. DNS 이름 또는 IP 주소를 대상으로 사용할 수 있습니다. 이 예제에서는 EC2 인스턴스와 `amazon.com` 사이의 경로 MTU를 확인합니다.

```
[ec2-user ~]$ tracepath amazon.com
1?: [LOCALHOST]          pmtu 9001
1:  ip-172-31-16-1.us-west-1.compute.internal (172.31.16.1)    0.187ms pmtu 1500
1:  no reply
2:  no reply
3:  no reply
4:  100.64.16.241 (100.64.16.241)                                0.574ms
5:  72.21.222.221 (72.21.222.221)                                84.447ms asymm 21
6:  205.251.229.97 (205.251.229.97)                            79.970ms asymm 19
7:  72.21.222.194 (72.21.222.194)                                96.546ms asymm 16
8:  72.21.222.239 (72.21.222.239)                            79.244ms asymm 15
9:  205.251.225.73 (205.251.225.73)                            91.867ms asymm 16
...
31:  no reply
      Too many hops: pmtu 1500
      Resume: pmtu 1500
```

이 예제에서 경로 MTU는 1500입니다.

Note

다른 Amazon EC2 인스턴스에 `tracepath`을 사용할 경우, 인스턴스의 보안 그룹 규칙에서 인바운드 UDP 트래픽을 허용하는지 확인해야 할 수 있습니다.

Amazon EC2 인스턴스에서 MTU 확인 및 설정

일부 AMI는 점보 프레임을 지원하는 인스턴스에서 점보 프레임을 사용하도록 구성되어 있는 반면, 표준 프레임 크기를 사용하도록 구성된 경우도 있습니다. VPC 내의 네트워크 트래픽에 점보 프레임을 사용하거나, 인터넷 트래픽에 표준 프레임을 사용할 수 있습니다. 어떤 사용 사례든 인스턴스가 예상대로 동작하는지 확인하는 것이 좋습니다. 이 섹션의 절차를 사용하여 네트워크 인터페이스의 MTU 설정을 확인하고 필요한 경우 수정할 수 있습니다.

Linux 인스턴스에서 MTU 설정을 확인하려면

- 인스턴스가 Linux 운영 체제에서 작동하는 경우 `ip` 명령을 사용하여 MTU 값을 검토할 수 있습니다. 다음 명령을 실행하여 현재 MTU 값을 결정합니다.

```
[ec2-user ~]$ ip link show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP mode
    DEFAULT group default qlen 1000
        link/ether 02:90:c0:b7:9e:d1 brd ff:ff:ff:ff:ff:ff
```

위의 예제에서 출력의 `mtu 9001`은 이 인스턴스가 점보 프레임을 사용함을 나타냅니다.

Linux 인스턴스에서 MTU 값을 설정하려면

- 인스턴스가 Linux 운영 체제에서 작동하는 경우 `ip` 명령을 사용하여 MTU 값을 설정할 수 있습니다. 다음 명령을 실행하여 원하는 MTU 값을 설정합니다. 이 프로시저는 MTU를 1500으로 설정하지만 9001에서 동일합니다.

```
[ec2-user ~]$ sudo ip link set dev eth0 mtu 1500
```

- (선택 사항) 재부팅 후에 네트워크 MTU 설정을 유지하려면 운영 체제 유형을 기반으로 다음 구성 파일을 수정하십시오. 이 프로시저는 Amazon Linux 및 Ubuntu에 해당하며, 다른 배포의 경우 특정 설명서를 참조하십시오.
 - Amazon Linux의 경우 `/etc/dhcp/dhclient-eth0.conf` 파일에 다음 줄을 추가합니다.

```
interface "eth0" {
    supersede interface-mtu 1500;
}
```

- Ubuntu의 경우 /etc/network/interfaces.d/eth0.cfg에 다음 줄을 추가합니다.

```
post-up /sbin/ifconfig eth0 mtu 1500
```

- (선택 사항) 인스턴스를 재부팅하고 MTU 설정이 올바른지 확인합니다.

문제 해결

점보 프레임을 사용할 때 EC2 인스턴스와 Amazon Redshift 클러스터 사이에 연결 문제가 발생할 경우 Amazon Redshift Cluster Management Guide의 [쿼리가 반응이 없는 것으로 나타남](#)을 참조하십시오.

Linux에서 향상된 네트워킹

향상된 네트워킹에서는 [지원되는 인스턴스 유형](#) (p. 533)에서 단일 루트 I/O 가상화(SR-IOV)를 사용하여 고성능 네트워킹 기능을 제공합니다. SR-IOV는 기존 가상 네트워크 인터페이스에 비해 높은 I/O 성능 및 낮은 CPU 사용률을 제공하는 디바이스 가상화 방법입니다. 향상된 네트워킹을 통해 대역폭과 PPS(Packet Per Second) 성능이 높아지고, 인스턴스 간 지연 시간이 지속적으로 낮아집니다. 향상된 네트워킹 사용에 따른 추가 요금은 없습니다.

목차

- [향상된 네트워킹 유형](#) (p. 533)
- [인스턴스에서 향상된 네트워킹 기능 활성화](#) (p. 534)
- [VPC의 Linux 인스턴스에서 Intel 82599 VF 인터페이스를 사용하여 향상된 네트워킹 활성화](#) (p. 534)
- [VPC의 Linux 인스턴스에서 ENA\(Elastic Network Adapter\)를 사용하여 향상된 네트워킹 활성화](#) (p. 543)
- [ENA\(Elastic Network Adapter\) 문제 해결](#) (p. 552)

향상된 네트워킹 유형

인스턴스 유형에 따라 다음 중 한 가지 메커니즘을 사용하여 향상된 네트워킹을 활성화할 수 있습니다.

Intel 82599 Virtual Function(VF) 인터페이스

Intel 82599 Virtual Function 인터페이스는 지원되는 인스턴스 유형에 대해 최대 10Gbps의 네트워크 속도를 지원합니다.

C3, C4, D2, I2, R3 및 M4(m4.16xlarge 제외) 인스턴스에서는 향상된 네트워킹을 위해 Intel 82599 VF 인터페이스를 사용합니다. 어떤 인스턴스 유형이 10Gbps 네트워크 속도를 지원하는지 알아보려면 [인스턴스 유형 매트릭스](#)를 참조하십시오.

ENA(Elastic Network Adapter)

탄력적 네트워크 어댑터(ENA)는 지원되는 인스턴스 유형에 대해 최대 20Gbps의 네트워크 속도를 지원합니다.

I3, P2, R4, X1, m4.16xlarge 인스턴스는 향상된 네트워킹을 위해 Elastic Network Adapter를 사용합니다. 어떤 인스턴스 유형이 20Gbps 네트워크 속도를 지원하는지 알아보려면 [인스턴스 유형 매트릭스](#)를 참조하십시오.

인스턴스에서 향상된 네트워킹 기능 활성화

인스턴스 유형에서 향상된 네트워킹을 위해 Intel 82599 VF 인터페이스를 지원하는 경우 [VPC의 Linux 인스턴스에서 Intel 82599 VF 인터페이스를 사용하여 향상된 네트워킹 활성화 \(p. 534\)](#)의 절차를 따르십시오.

인스턴스 유형에서 향상된 네트워킹을 위해 ENA를 지원하는 경우 [VPC의 Linux 인스턴스에서 ENA\(Elastic Network Adapter\)를 사용하여 향상된 네트워킹 활성화 \(p. 543\)](#)의 절차를 따르십시오.

VPC의 Linux 인스턴스에서 Intel 82599 VF 인터페이스를 사용하여 향상된 네트워킹 활성화

Amazon EC2는 Intel `ixgbevf` 드라이버를 사용하는 Intel 82599 VF 인터페이스를 통해 C3, C4, D2, I2, R3 및 M4(`m4.16xlarge` 제외) 인스턴스에 향상된 네트워킹 기능을 제공합니다.

Intel 82599 VF 인터페이스를 사용하여 향상된 네트워킹을 준비하려면 인스턴스를 다음과 같이 설정하십시오.

- 2.6.32 버전 이상의 Linux 커널을 사용하는 HVM AMI에서 인스턴스를 시작합니다. 최신 Amazon Linux HVM AMI에는 향상된 네트워킹에 요구되는 모듈이 설치되어 있으며 필요한 속성 세트를 갖추고 있습니다. 따라서 Amazon Linux HVM AMI를 사용하여 Amazon EBS 및 향상된 네트워크를 지원하는 인스턴스를 시작하면, 인스턴스 설정에서 향상된 네트워크 기능이 처음부터 활성화되어 있습니다.
- VPC에서 인스턴스를 시작합니다. (EC2-Classic을 사용하는 경우 인스턴스에서 향상된 네트워킹을 사용할 수 없습니다.)
- [AWS CLI](#) 또는 [Windows PowerShell용 AWS 도구](#)를 자신이 선택한 컴퓨터에 설치하고 구성합니다(로컬 데스크톱/노트북 권장). 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오. Amazon EC2 콘솔에서는 향상된 네트워킹을 관리할 수 없습니다.
- 인스턴스에 보존해야 할 중요한 데이터가 있는 경우 인스턴스에서 AMI를 만들어 데이터를 백업해야 합니다. 커널 및 커널 모듈 업데이트 외에도 `sriovNetSupport` 속성을 활성화하면 호환되지 않는 인스턴스나 운영 체제에 접속할 수 없게 됩니다. 최신 백업을 확보하면 이런 경우에도 데이터를 보존할 수 있습니다.

목차

- [Intel 82599 VF 인터페이스를 통해 향상된 네트워킹이 활성화되는지 여부 확인 \(p. 534\)](#)
- [Amazon Linux에서 Intel 82599 VF 인터페이스를 사용하여 향상된 네트워킹 활성화 \(p. 537\)](#)
- [Ubuntu에서 Intel 82599 VF 인터페이스를 사용하여 향상된 네트워킹 활성화 \(p. 538\)](#)
- [다른 Linux 배포판에서 Intel 82599 VF 인터페이스를 사용하여 향상된 네트워킹 활성화 \(p. 541\)](#)
- [연결 문제 해결 \(p. 543\)](#)

Intel 82599 VF 인터페이스를 통해 향상된 네트워킹이 활성화되는지 여부 확인

Intel 82599 VF 인터페이스를 사용하여 향상된 네트워킹 기능이 이미 활성화되었는지를 알아보려면, 인스턴스에 `ixgbevf` 모듈 가 설치되어 있는지 그리고 `sriovNetSupport` 속성이 설정되어 있는지를 확인해야 합니다. 인스턴스에서 두 조건을 충족하는 경우 `ethtool -i ethn` 명령을 사용했을 때 해당 모듈이 네트워크 인터페이스에서 사용 중이라고 표시됩니다.

커널 모듈(`ixgbevf`)

`ixgbevf` 모듈이 설치되어 있는지 그리고 향상된 네트워킹 기능과 호환되는지를 확인하려면 다음 절차에 따라 `modinfo` 명령을 사용합니다.

```
[ec2-user ~]$ modinfo ixgbevf
filename:      /lib/modules/3.10.48-55.140.amzn1.x86_64/kernel/drivers/amazon/ixgbevf/
ixgbevf.ko
version:       2.14.2
license:        GPL
description:   Intel(R) 82599 Virtual Function Driver
author:         Intel Corporation, <linux.nics@intel.com>
srcversion:    50CBF6F36B99FE70E56C95A
alias:          pci:v00008086d00001515sv*sd*bc*sc*i*
alias:          pci:v00008086d000010EDsv*sd*bc*sc*i*
depends:
intreee:        Y
vermagic:      3.10.48-55.140.amzn1.x86_64 SMP mod_unload modversions
parm:          InterruptThrottleRate:Maximum interrupts per second, per vector,
(956-488281, 0=off, 1=dynamic), default 1 (array of int)
```

위의 Amazon Linux의 경우, ixgbevf 모듈이 이미 설치되었고 최소 버전 요구(2.14.2)을 충족하는 것을 알 수 있습니다.

```
ubuntu:~$ modinfo ixgbevf
filename:      /lib/modules/3.13.0-29-generic/kernel/drivers/net/ethernet/intel/ixgbevf/
ixgbevf.ko
version:       2.11.3-k
license:        GPL
description:   Intel(R) 82599 Virtual Function Driver
author:         Intel Corporation, <linux.nics@intel.com>
srcversion:    0816EA811025C8062A9C269
alias:          pci:v00008086d00001515sv*sd*bc*sc*i*
alias:          pci:v00008086d000010EDsv*sd*bc*sc*i*
depends:
intreee:        Y
vermagic:      3.13.0-29-generic SMP mod_unload modversions
signer:         Magrathea: Glacier signing key
sig_key:        66:02:CB:36:F1:31:3B:EA:01:C4:BD:A9:65:67:CF:A7:23:C9:70:D8
sig_hashalgo:   sha512
parm:          debug:Debug level (0=none,...,16=all) (int)
```

위에 언급된 Ubuntu 인스턴스에서는, 모듈 자체는 설치되어 있지만 권장 버전인 2.14.2에 포함된 최신 버그 수정이 적용되지 않은 2.11.3-k 버전을 사용하고 있습니다. 이 경우 ixgbevf 모듈이 작동은 되지만, 성능을 최대화하려면 인스턴스에 새 버전을 설치하고 로드해야 합니다.

인스턴스 속성(sriovNetSupport)

다음 명령 중 하나를 사용하여 인스턴스에 향상된 네트워킹 sriovNetSupport 속성 세트가 있는지 확인할 수 있습니다.

- [describe-instance-attribute](#) (AWS CLI)

```
aws ec2 describe-instance-attribute --instance-id instance_id --attribute sriovNetSupport
```

- [Get-EC2InstanceAttribute](#) (Windows PowerShell용 AWS 도구)

```
Get-EC2InstanceAttribute -InstanceId instance-id -Attribute sriovNetSupport
```

속성이 설정되지 않은 경우 SrioVNetSupport가 빈 값으로 표시되고, 그렇지 않으면 다음과 같이 설정됩니다.

```
"SriovNetSupport": {
```

```
        "Value": "simple"  
},
```

이미지 속성(sriovNetSupport)

다음 명령 중 하나를 사용하여 AMI에 향상된 네트워킹 sriovNetSupport 속성 세트가 있는지 확인할 수 있습니다.

- [describe-image-attribute](#) (AWS CLI)

```
aws ec2 describe-image-attribute --image-id ami_id --attribute sriovNetSupport
```

이 명령은 사용자가 소유한 이미지에만 적용됩니다. 계정에서 소유한 이미지가 아닌 경우에는 AuthFailure 오류가 표시됩니다.

- [Get-EC2ImageAttribute](#) (Windows PowerShell용 AWS 도구)

```
Get-EC2ImageAttribute -ImageId ami_id -Attribute sriovNetSupport
```

속성이 설정되지 않은 경우 SrioVNetSupport가 빈 값으로 표시되고, 그렇지 않으면 다음과 같이 설정됩니다.

```
"SriovNetSupport": {  
    "Value": "simple"  
},
```

네트워크 인터페이스 드라이버

다음 명령과 확인하고자 하는 인터페이스 이름을 사용하여 해당 인터페이스에서 모듈이 사용되고 있는지를 확인할 수 있습니다. 단일 인터페이스를 사용하는 경우(기본 설정), eth0으로 표시됩니다.

```
[ec2-user ~]$ ethtool -i eth0  
driver: vif  
version:  
firmware-version:  
bus-info: vif-0  
supports-statistics: yes  
supports-test: no  
supports-eeprom-access: no  
supports-register-dump: no  
supports-priv-flags: no
```

위의 경우, 표시된 드라이버가 vif이므로 ixgbevf 모듈이 로드되지 않은 것입니다.

```
[ec2-user ~]$ ethtool -i eth0  
driver: ixgbevf  
version: 2.14.2  
firmware-version: N/A  
bus-info: 0000:00:03.0  
supports-statistics: yes  
supports-test: yes  
supports-eeprom-access: no  
supports-register-dump: yes  
supports-priv-flags: no
```

이 경우, ixgbevf 모듈이 이미 설치되었고 최소 버전 요구를 충족하는 것을 알 수 있습니다. 이 인스턴스는 향상된 네트워킹이 올바르게 구성된 상태입니다.

Amazon Linux에서 Intel 82599 VF 인터페이스를 사용하여 향상된 네트워킹 활성화

최신 Amazon Linux HVM AMI에는 향상된 네트워킹에 요구되는 `ixgbevf` 모듈이 설치되어 있으며 필요한 `sriovNetSupport` 속성 세트를 갖추고 있습니다. 따라서 Amazon Linux HVM AMI를 사용하여 C3, C4, R3 또는 M4(m4.16xlarge 제외) 인스턴스를 시작하는 경우 인스턴스에 대해 향상된 네트워크 기능이 이미 활성화되어 있습니다. 자세한 내용은 [Intel 82599 VF 인터페이스를 통해 향상된 네트워킹이 활성화되는지 여부 확인 \(p. 534\)](#) 섹션을 참조하십시오.

지난 세대의 Amazon Linux AMI를 사용하여 인스턴스를 시작했고 향상된 네트워크 기능이 활성화되어 있지 않은 경우에는 다음 절차에 따라 향상된 네트워크를 활성화할 수 있습니다.

향상된 네트워킹 기능 사용(EBS 기반 인스턴스)

1. 인스턴스 연결 후.
2. 인스턴스 상에서 다음 명령을 사용하여 `ixgbevf`를 포함한 최신 커널과 커널 모듈로 인스턴스를 업데이트합니다.

```
[ec2-user ~]$ sudo yum update
```

3. 로컬 컴퓨터를 사용하는 경우, Amazon EC2 콘솔을 사용하거나 다음 명령 중 하나를 사용하여 인스턴스를 재부팅하십시오. [reboot-instances](#) (AWS CLI), [Restart-EC2Instance](#) (Windows PowerShell용 AWS 도구).
4. 인스턴스에 다시 연결하고 [Intel 82599 VF 인터페이스를 통해 향상된 네트워킹이 활성화되는지 여부 확인 \(p. 534\)](#)에서 `modinfo ixgbevf` 명령을 사용하여 `ixgbevf` 모듈이 설치되어 있고 최소 권장 버전 요건을 충족하는지를 확인합니다.
5. 로컬 컴퓨터를 사용하는 경우, Amazon EC2 콘솔을 사용하거나 다음 명령 중 하나를 사용하여 인스턴스를 중지하십시오. [stop-instances](#) (AWS CLI), [Stop-EC2Instance](#) (Windows PowerShell용 AWS 도구). 인스턴스를 AWS OpsWorks에서 관리할 경우 AWS OpsWorks 콘솔에서 인스턴스를 중지해야 인스턴스 상태가 동기화됩니다.

Important

인스턴스 스토어 지원 인스턴스를 사용할 때는 인스턴스를 중지할 수 없습니다. 이 경우, [향상된 네트워킹 기능 사용\(인스턴스 스토어 지원 인스턴스\) \(p. 538\)](#) 단계로 넘어갑니다.

6. 사용자의 로컬 컴퓨터에서 다음 명령 중 하나를 사용하여 향상된 네트워크 속성을 활성화합니다.

Warning

향상된 네트워킹 속성을 활성화한 다음에는 다시 비활성화할 수 없습니다.

Warning

향상된 네트워킹 기능은 HVM 인스턴스에서만 지원됩니다. PV 인스턴스에서 향상된 네트워킹 기능을 활성화하면 인스턴스 접속이 불가능해질 수 있습니다. 올바른 모듈과 모듈 버전을 사용하지 않고 속성을 설정하는 경우에도 인스턴스 접속이 불가능해질 수 있습니다.

- [modify-instance-attribute](#) (AWS CLI)

```
aws ec2 modify-instance-attribute --instance-id instance_id --sriov-net-support simple
```

- [Edit-EC2InstanceAttribute](#) (Windows PowerShell용 AWS 도구)

```
Edit-EC2InstanceAttribute -InstanceId instance_id -SriovNetSupport "simple"
```

7. (선택 사항) Amazon EBS 지원 Linux AMI 생성 (p. 81) 의 설명에 따라 인스턴스에서 AMI를 생성합니다. 생성된 AMI는 인스턴스의 향상된 네트워크 속성을 상속합니다. 따라서 이 AMI를 사용하여 기본적으로 향상된 네트워킹 기능이 활성화된 상태로 다른 인스턴스를 시작할 수 있습니다.
8. 로컬 컴퓨터를 사용하는 경우, Amazon EC2 콘솔을 사용하거나 다음 명령 중 하나를 사용하여 인스턴스를 시작하십시오. `start-instances`(AWS CLI), `Start-EC2Instance`(Windows PowerShell용 AWS 도구). 인스턴스를 AWS OpsWorks에서 관리할 경우 AWS OpsWorks 콘솔에서 인스턴스를 시작해야 인스턴스 상태가 동기화됩니다.
9. 인스턴스에 다시 연결하고 Intel 82599 VF 인터페이스를 통해 향상된 네트워킹이 활성화되는지 여부 확인 (p. 534)에서 `ethtool -i ethn` 명령을 사용하여 `ixgbevf` 모듈이 설치되어 있고 최소 권장 버전 요구를 충족하는지를 확인합니다.

향상된 네트워킹 기능 사용(인스턴스 스토어 지원 인스턴스)

인스턴스 지원 인스턴스를 사용하는 경우, 전 과정의 내용 중 Step 1 (p. 537)에서 Step 4 (p. 537)까지 수 행한 다음 인스턴스 스토어 기반 Linux AMI 생성 (p. 84)의 설명에 따라 새 AMI를 생성합니다. AMI를 등록할 때 향상된 네트워킹 속성을 활성화해야 합니다.

- `register-image`(AWS CLI)

```
aws ec2 register-image --srivnet-support simple ...
```

- `Register-EC2Image` (Windows PowerShell용 AWS 도구)

```
Register-EC2Image -SriovNetSupport "simple" ...
```

Ubuntu에서 Intel 82599 VF 인터페이스를 사용하여 향상된 네트워킹 활성화

다음 절차는 Ubuntu 인스턴스에서 Intel 82599 VF 인터페이스를 사용하여 향상된 네트워킹을 활성화하는 일반적인 방법입니다.

Ubuntu에서 향상된 네트워킹 기능 사용(EBS 기반 인스턴스)

1. 인스턴스에 연결합니다.
2. 패키지 캐시와 패키지를 업데이트합니다.

```
ubuntu:~$ sudo apt-get update && sudo apt-get upgrade -y
```

Important

업데이트 과정에서 `grub` 설치 메시지가 표시되는 경우, `/dev/xvda`를 사용하여 `grub`을 설치하고, `/boot/grub/menu.lst`의 현재 버전을 유지하도록 선택합니다.

3. 커널이 업데이트될 때마다 `ixgbevf` 모듈이 다시 빌드되도록 dkms 패키지를 설치합니다.

```
ubuntu:~$ sudo apt-get install -y dkms
```

4. Sourceforge(<http://sourceforge.net/projects/e1000/files/ixgbevf%20stable/>)에서 버전 2.16.4의 `ixgbevf` 모듈에 대한 소스를 인스턴스에 다운로드합니다.

최소 버전 요구 사항(2.14.2)을 비롯하여 `ixgbevf`의 이전 버전들은 Ubuntu 일부 버전에서는 제대로 빌드되지 않습니다. `ixgbevf`의 2.16.4 버전은 Ubuntu 인스턴스에 사용되어야 합니다.

```
ubuntu:~$ wget "sourceforge.net/projects/e1000/files/ixgbevf_stable/2.16.4/ixgbevf-2.16.4.tar.gz"
```

5. ixgbevf 패키지의 압축 및 아카이빙을 해제합니다.

```
ubuntu:~$ tar -xzf ixgbevf-2.16.4.tar.gz
```

6. ixgbevf 패키지를 /usr/src/ 디렉터리로 이동하여 커널이 업데이트될 때마다 dkms에서 파일을 찾아 빌드할 수 있도록 합니다.

```
ubuntu:~$ sudo mv ixgbevf-2.16.4 /usr/src/
```

7. 다음 값을 사용하여 dkms 구성 파일을 생성합니다. 버전 값으로 현재 사용 중인 ixgbevf 버전을 입력합니다.

- a. 파일을 생성합니다.

```
ubuntu:~$ sudo touch /usr/src/ixgbevf-2.16.4/dkms.conf
```

- b. 파일을 수정하고 다음 값을 추가합니다.

```
ubuntu:~$ sudo vim /usr/src/ixgbevf-2.16.4/dkms.conf
PACKAGE_NAME="ixgbevf"
PACKAGE_VERSION="2.16.4"
CLEAN="cd src/; make clean"
MAKE="cd src/; make BUILD_KERNEL=${kernelver}"
BUILT_MODULE_LOCATION[0]="src/"
BUILT_MODULE_NAME[0]="ixgbevf"
DEST_MODULE_LOCATION[0]="/updates"
DEST_MODULE_NAME[0]="ixgbevf"
AUTOINSTALL="yes"
```

8. dkms를 사용하여 인스턴스에 ixgbevf 모듈을 추가 및 빌드하고 설치합니다.

- a. 모듈을 dkms에 추가합니다.

```
ubuntu:~$ sudo dkms add -m ixgbevf -v 2.16.4
```

- b. dkms로 모듈을 구축합니다.

```
ubuntu:~$ sudo dkms build -m ixgbevf -v 2.16.4
```

- c. dkms로 모듈을 설치합니다.

```
ubuntu:~$ sudo dkms install -m ixgbevf -v 2.16.4
```

9. 부팅 시 올바른 모듈이 로드되도록 initramfs를 다시 빌드합니다.

```
ubuntu:~$ sudo update-initramfs -c -k all
```

10. Intel 82599 VF 인터페이스를 통해 향상된 네트워킹이 활성화되는지 여부 확인 (p. 534)에서 modinfo ixgbevf 명령을 사용하여 ixgbevf 모듈이 설치되어 있고 최소 권장 버전 요구를 충족하는지를 확인합니다.

```
ubuntu:~$ modinfo ixgbevf
filename:      /lib/modules/3.13.0-74-generic/updates/dkms/ixgbevf.ko
version:       2.16.4
license:       GPL
```

```
description: Intel(R) 10 Gigabit Virtual Function Network Driver
author: Intel Corporation, <linux.nics@intel.com>
srcversion: 759A432E3151C8F9F6EA882
alias: pci:v00008086d00001515sv*sd*bc*sc*i*
alias: pci:v00008086d000010EDsv*sd*bc*sc*i*
depends:
vermagic: 3.13.0-74-generic SMP mod_unload modversions
parm: InterruptThrottleRate:Maximum interrupts per second, per vector,
(956-488281, 0=off, 1=dynamic), default 1 (array of int)
```

- 로컬 컴퓨터를 사용하는 경우, Amazon EC2 콘솔을 사용하거나 다음 명령 중 하나를 사용하여 인스턴스를 중지하십시오. [stop-instances](#) (AWS CLI), [Stop-EC2Instance](#) (Windows PowerShell용 AWS 도구). 인스턴스를 AWS OpsWorks에서 관리할 경우 AWS OpsWorks 콘솔에서 인스턴스를 중지해야 인스턴스 상태가 동기화됩니다.

Important

인스턴스 스토어 지원 인스턴스를 사용할 때는 인스턴스를 중지할 수 없습니다. 이 경우, [Ubuntu에서 향상된 네트워킹 기능 사용\(인스턴스 스토어 지원 인스턴스\)](#) (p. 540) 단계로 넘어갑니다.

- 사용자의 로컬 컴퓨터에서 다음 명령 중 하나를 사용하여 향상된 네트워크 sriovNetSupport 속성을 활성화합니다. 주의: 이 속성을 활성화한 다음에는 다시 비활성화할 수 없습니다.

Warning

향상된 네트워킹 기능은 HVM 인스턴스에서만 지원됩니다. PV 인스턴스에서 향상된 네트워킹 기능을 활성화하면 인스턴스 접속이 불가능해질 수 있습니다. 올바른 모듈과 모듈 버전을 사용하지 않고 속성을 설정하는 경우에도 인스턴스 접속이 불가능해질 수 있습니다.

- [modify-instance-attribute](#) (AWS CLI)

```
aws ec2 modify-instance-attribute --instance-id instance_id --sriov-net-support simple
```

- [Edit-EC2InstanceAttribute](#) (Windows PowerShell용 AWS 도구)

```
Edit-EC2InstanceAttribute -InstanceId instance_id -SriovNetSupport "simple"
```

- (선택 사항) [Amazon EBS 지원 Linux AMI 생성](#) (p. 81) 의 설명에 따라 인스턴스에서 AMI를 생성합니다. 생성된 AMI는 인스턴스의 향상된 네트워킹 sriovNetSupport 속성을 상속합니다. 따라서 이 AMI를 사용하여 기본적으로 향상된 네트워킹 기능이 활성화된 상태로 다른 인스턴스를 시작할 수 있습니다.
- 로컬 컴퓨터를 사용하는 경우, Amazon EC2 콘솔을 사용하거나 다음 명령 중 하나를 사용하여 인스턴스를 시작하십시오. [start-instances](#) (AWS CLI), [Start-EC2Instance](#) (Windows PowerShell용 AWS 도구). 인스턴스를 AWS OpsWorks에서 관리할 경우 AWS OpsWorks 콘솔에서 인스턴스를 시작해야 인스턴스 상태가 동기화됩니다.
- (선택 사항) 인스턴스에 연결하여 모듈의 설치 여부를 확인합니다.

[Ubuntu에서 향상된 네트워킹 기능 사용\(인스턴스 스토어 지원 인스턴스\)](#)

인스턴스 지원 인스턴스를 사용하는 경우, 전 과정의 내용 중 [Step 1](#) (p. 538)에서 [Step 10](#) (p. 539)까지 수행한 다음 [인스턴스 스토어 기반 Linux AMI 생성](#) (p. 84)의 설명에 따라 새 AMI를 생성합니다. AMI를 등록할 때 향상된 네트워킹 속성을 활성화해야 합니다.

Warning

향상된 네트워킹 기능은 HVM 인스턴스에서만 지원됩니다. PV 인스턴스에서 향상된 네트워킹 기능을 활성화하면 인스턴스 접속이 불가능해질 수 있습니다. 올바른 모듈과 모듈 버전을 사용하지 않고 속성을 설정하는 경우에도 인스턴스 접속이 불가능해질 수 있습니다.

- [register-image](#) (AWS CLI)

```
aws ec2 register-image --srivnet-support simple ...
```

- Register-EC2Image (Windows PowerShell용 AWS 도구)

```
Register-EC2Image -SriovNetSupport "simple" ...
```

다른 Linux 배포판에서 Intel 82599 VF 인터페이스를 사용하여 향상된 네트워킹 활성화

다음 절차는 Amazon Linux 또는 Ubuntu를 제외한 다른 Linux 배포판에서 Intel 82599 VF 인터페이스를 사용하여 향상된 네트워킹을 활성화하는 일반적인 방법입니다. 명령 구문과 파일 위치, 패키지 및 도구 지원을 비롯한 자세한 내용은 사용 Linux 배포판의 전용 문서를 참조하십시오.

Linux에서 향상된 네트워킹 기능 사용(EBS 기반 인스턴스)

1. 인스턴스에 연결합니다.
2. Sourceforge(<http://sourceforge.net/projects/e1000/files/ixgbevf%20stable/>)에서 버전 2.14.2의 ixgbevf 모듈에 대한 소스를 인스턴스에 다운로드합니다. 이는 향상된 네트워킹을 사용하기 위해 권장되는 최소 버전입니다.

최소 요구 버전인 2.14.2을 포함하여 ixgbevf의 이전 버전은 특정 버전의 Ubuntu를 포함하는 일부 Linux 배포판에서 제대로 빌드되지 않습니다. 빌드 오류가 생기면 2.16.4(영향을 받는 Ubuntu 버전의 빌드 문제 수정)와 같은 새로운 버전을 시도해볼 수 있습니다.

3. 인스턴스에서 ixgbevf 모듈을 컴파일하고 설치합니다.

사용 중인 배포판에서 dkms를 지원하는 경우, 시스템 커널이 업데이트될 때마다 ixgbevf 모듈이 재컴파일되도록 dkms를 구성하는 것이 좋습니다. 사용 중인 배포판에서 dkms를 기본 지원하지 않는 경우 EPEL 리포지토리(<https://fedoraproject.org/wiki/EPEL>)에서 Red Hat Enterprise Linux 버전용을 검색하거나 <http://linux.dell.com/dkms/>에서 소프트웨어를 다운로드할 수 있습니다. Ubuntu에서 향상된 네트워킹 기능 사용(EBS 기반 인스턴스) (p. 538) 내용 중 Step 6 (p. 539) ~ Step 8 (p. 539) 과정을 참고하여 dkms 구성하십시오.

Warning

현재 사용 중인 커널을 기준으로 ixgbevf 모듈을 컴파일한 후 새 커널에 맞게 다시 빌드하지 않거나 커널 업그레이드를 진행하면, 시스템에서 재부팅할 때 배포 버전의 ixgbevf 모듈로 돌아갈 수 있으며, 이때 배포 버전과 향상된 네트워킹이 호환되지 않으면 시스템에 접속하지 못하는 결과가 발생할 수 있습니다.

4. sudo depmod 명령을 실행하여 모듈 의존관계를 업데이트합니다.
 5. 인스턴스에서 initramfs를 업데이트하여 부팅 시 새 모듈이 로드되도록 합니다.
 6. 시스템이 예측 가능한 네트워크 인터페이스 이름을 기본으로 사용하는지 확인합니다. 사용하는 systemd 또는 udev 버전이 197 이상인 시스템에서는 이더넷 디바이스의 이름 변경이 가능해 단일 네트워크 인터페이스가 아닌 경우에도 eth0 이름이 할당될 수 있습니다. 이에 따라 인스턴스 연결에 문제가 발생할 수 있습니다. 자세한 내용과 다른 구성 옵션을 보려면 freedesktop.org 웹 사이트에서 예측 가능한 네트워크 인터페이스 이름을 참조하십시오.
- a. RPM 기반 시스템에서는 다음 명령을 사용하여 systemd 또는 udev 버전을 확인할 수 있습니다.

```
[ec2-user ~]$ rpm -qa | grep -e '^systemd-[0-9]+\+|\^udev-[0-9]+\+'  
systemd-208-11.el7_0.2.x86_64
```

위의 Red Hat 7 예제에서, systemd 버전은 208이므로, 해당 네트워크 인터페이스 이름을 비활성해야 합니다.

- b. net.ifnames=0/etc/default/grubGRUB_CMDLINE_LINUX의 ## 옵션을 추가하여 예측 가능한 네트워크 인터페이스 이름을 비활성화합니다.

```
[ec2-user ~]$ sudo sed -i '/^GRUB_CMDLINE_LINUX/s/"$/ \ net.ifnames=0"/' /etc/default/grub
```

- c. GRUB 구성 파일을 재구축합니다.

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

7. 로컬 컴퓨터를 사용하는 경우, Amazon EC2 콘솔을 사용하거나 다음 명령 중 하나를 사용하여 인스턴스를 중지하십시오. [stop-instances](#) (AWS CLI), [Stop-EC2Instance](#) (Windows PowerShell용 AWS 도구). 인스턴스를 AWS OpsWorks에서 관리할 경우 AWS OpsWorks 콘솔에서 인스턴스를 중지해야 인스턴스 상태가 동기화됩니다.

Important

인스턴스 스토어 지원 인스턴스를 사용할 때는 인스턴스를 중지할 수 없습니다. 이 경우, [향상된 네트워킹 기능 사용\(인스턴스 스토어 지원 인스턴스\)](#) (p. 542) 단계로 넘어갑니다.

8. 사용자의 로컬 컴퓨터에서 다음 명령 중 하나를 사용하여 향상된 네트워크 속성을 활성화합니다. 주의: 향상된 네트워킹 속성을 활성화한 다음에는 다시 비활성화할 수 없습니다.

Warning

향상된 네트워킹 기능은 HVM 인스턴스에서만 지원됩니다. PV 인스턴스에서 향상된 네트워킹 기능을 활성화하면 인스턴스 접속이 불가능해질 수 있습니다. 올바른 모듈과 모듈 버전을 사용하지 않고 속성을 설정하는 경우에도 인스턴스 접속이 불가능해질 수 있습니다.

- [modify-instance-attribute](#) (AWS CLI)

```
aws ec2 modify-instance-attribute --instance-id instance_id --srivnet-support simple
```

- [Edit-EC2InstanceAttribute](#) (Windows PowerShell용 AWS 도구)

```
Edit-EC2InstanceAttribute -InstanceId instance_id -SriovNetSupport "simple"
```

9. ([선택 사항](#)) [Amazon EBS 지원 Linux AMI 생성](#) (p. 81) 의 설명에 따라 인스턴스에서 AMI를 생성합니다. 생성된 AMI는 인스턴스의 향상된 네트워크 속성을 상속합니다. 따라서 이 AMI를 사용하여 기본적으로 향상된 네트워킹 기능이 활성화된 상태로 다른 인스턴스를 시작할 수 있습니다.

Important

인스턴스 운영 체제에 /etc/udev/rules.d/70-persistent-net.rules 파일이 포함되어 있는 경우 AMI를 생성하기 전에 먼저 삭제해야 합니다. 이 파일에 원본 인스턴스의 이더넷 어댑터에 대한 MAC 주소가 포함되어 있습니다. 이 파일로 다른 인스턴스가 부팅되면 운영 체제에서 디바이스를 찾을 수 없으며 eth0이 실패하여 부팅 문제가 발생할 수 있습니다. 이 파일은 다음 부팅 주기에 생성되고 AMI에서 시작된 모든 인스턴스가 자체 버전의 파일을 생성합니다.

10. 로컬 컴퓨터를 사용하는 경우, Amazon EC2 콘솔을 사용하거나 다음 명령 중 하나를 사용하여 인스턴스를 시작하십시오. [start-instances](#) (AWS CLI), [Start-EC2Instance](#) (Windows PowerShell용 AWS 도구). 인스턴스를 AWS OpsWorks에서 관리할 경우 AWS OpsWorks 콘솔에서 인스턴스를 시작해야 인스턴스 상태가 동기화됩니다.

11. ([선택 사항](#)) 인스턴스에 연결하여 모듈의 설치 여부를 확인합니다.

[향상된 네트워킹 기능 사용\(인스턴스 스토어 지원 인스턴스\)](#)

인스턴스 지원 인스턴스를 사용하는 경우, 전 과정의 내용 중 [Step 1 \(p. 541\)](#)에서 [Step 5 \(p. 541\)](#)까지 수 행한 다음 [인스턴스 스토어 기반 Linux AMI 생성 \(p. 84\)](#)의 설명에 따라 새 AMI를 생성합니다. AMI를 등록할 때 향상된 네트워킹 속성을 활성화해야 합니다.

Warning

향상된 네트워킹 기능은 HVM 인스턴스에서만 지원됩니다. PV 인스턴스에서 향상된 네트워킹 기능을 활성화하면 인스턴스 접속이 불가능해질 수 있습니다. 올바른 모듈과 모듈 버전을 사용하지 않고 속성을 설정하는 경우에도 인스턴스 접속이 불가능해질 수 있습니다.

- [register-image\(AWS CLI\)](#)

```
aws ec2 register-image --srivnet-support simple ...
```

- [Register-EC2Image \(Windows PowerShell용 AWS 도구\)](#)

```
Register-EC2Image -SriovNetSupport "simple" ...
```

연결 문제 해결

향상된 네트워킹 기능을 활성화하는 도중 연결이 끊기는 경우, 커널이 ixgbevf 모듈과 호환되지 않아 발생한 문제일 수 있습니다. 사용 중인 Linux 배포판과 함께 제공되는 ixgbevf 모듈 버전을 인스턴스에 설치하여 사용해 보십시오.

PV 또는 AMI 인스턴스에서 향상된 네트워킹 기능을 활성화하면 인스턴스 접속이 불가능해질 수 있습니다.

VPC의 Linux 인스턴스에서 ENA(Elastic Network Adapter)를 사용하여 향상된 네트워킹 활성화

ENA 네트워크 어댑터를 사용하여 향상된 네트워킹을 준비하려면 인스턴스를 다음과 같이 설정하십시오.

- 3.2 버전 이상의 Linux 커널을 사용하는 HVM AMI에서 인스턴스를 시작합니다. 최신 Amazon Linux HVM AMI에는 향상된 네트워킹에 요구되는 모듈이 설치되어 있으며 필요한 속성 세트를 갖추고 있습니다. 따라서 Amazon Linux HVM AMI를 사용하여 Amazon EBS 및 향상된 네트워크를 지원하는 인스턴스를 시작하면, 인스턴스 설정에서 ENA 향상된 네트워크 기능이 처음부터 활성화되어 있습니다.
- VPC에서 인스턴스를 시작합니다. (EC2-Classic을 사용하는 경우 인스턴스에서 향상된 네트워킹을 사용 할 수 없습니다.)
- [AWS CLI](#) 또는 [Windows PowerShell용 AWS 도구](#)를 자신이 선택한 컴퓨터에 설치하고 구성합니다(로컬 디스크톱/노트북 권장). 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오. Amazon EC2 콘솔에서는 향상된 네트워킹을 관리할 수 없습니다.
- 인스턴스에 보존해야 할 중요한 데이터가 있는 경우 인스턴스에서 AMI를 만들어 데이터를 백업해야 합니다. 커널 및 커널 모듈 업데이트 외에도 enaSupport 속성을 활성화하면 호환되지 않는 인스턴스나 운영 체제에 접속할 수 없게 됩니다. 최신 백업을 확보하면 이런 경우에도 데이터를 보존할 수 있습니다.

목차

- [ENA를 통한 향상된 네트워킹 기능 활성화 여부 확인 \(p. 544\)](#)
- [Amazon Linux에서 ENA를 사용하여 향상된 네트워킹 기능 활성화 \(p. 545\)](#)
- [Ubuntu에서 ENA를 사용하여 향상된 네트워킹 활성화 \(p. 547\)](#)
- [다른 Linux 배포판에서 ENA를 사용하여 향상된 네트워킹 기능 활성화 \(p. 550\)](#)
- [문제 해결 \(p. 552\)](#)

ENA를 통한 향상된 네트워킹 기능 활성화 여부 확인

ENA를 통한 향상된 네트워킹 기능이 활성화되었는지를 확인하려면 인스턴스에 ena 모듈 가 설치되었는지 그리고 enaSupport 속성이 설정되었는지를 확인해야 합니다. 인스턴스에서 두 조건을 충족하는 경우 ethtool -i eth_n 명령을 사용했을 때 해당 모듈이 네트워크 인터페이스에서 사용 중이라고 표시됩니다.

커널 모듈(ena)

ena 모듈이 설치되어 있는지 확인하려면 다음과 같이 modinfo 명령을 사용합니다.

```
[ec2-user ~]$ modinfo ena
filename:      /lib/modules/4.4.11-23.53.amzn1.x86_64/kernel/drivers/amazon/net/ena/ena.ko
version:       0.6.6
license:        GPL
description:   Elastic Network Adapter (ENA)
author:        Amazon.com, Inc. or its affiliates
srcversion:    3141E47566402C79D6B8284
alias:         pci:v00001D0Fd0000EC21sv*sd*bc*sc*i*
alias:         pci:v00001D0Fd0000EC20sv*sd*bc*sc*i*
alias:         pci:v00001D0Fd00001EC2sv*sd*bc*sc*i*
alias:         pci:v00001D0Fd00000EC2sv*sd*bc*sc*i*
depends:
intree:        Y
vermagic:     4.4.11-23.53.amzn1.x86_64 SMP mod_unload modversions
parm:          debug:Debug level (0=none,...,16=all) (int)
parm:          push_mode:Descriptor / header push mode (0=automatic,1=disable,3=enable)
            0 - Automatically choose according to device capability (default)
            1 - Don't push anything to device memory
            3 - Push descriptors and header buffer to device memory (int)
parm:          enable_wd:Enable keepalive watchdog (0=disable,1=enable,default=1) (int)
parm:          enable_missing_tx_detection:Enable missing Tx completions. (default=1)
            (int)
parm:          numa_node_override_array:Numa node override map
            (array of int)
parm:          numa_node_override:Enable/Disable numa node override (0=disable)
            (int)
```

위의 Amazon Linux 예제의 경우, ena 모듈이 설치되어 있습니다.

```
ubuntu:~$ modinfo ena
ERROR: modinfo: could not find module ena
```

위 Ubuntu 인스턴스에서는 모듈이 설치되어 있지 않으므로 먼저 모듈을 설치해야 합니다. 자세한 내용은 [Ubuntu에서 ENA를 사용하여 향상된 네트워킹 활성화 \(p. 547\)](#) 섹션을 참조하십시오.

인스턴스 속성(enaSupport)

다음 명령 중 하나를 사용하여 인스턴스에 향상된 네트워킹 enaSupport 속성 세트가 있는지 확인할 수 있습니다. 속성이 설정되었으면 true가 반환됩니다.

- [describe-instances \(AWS CLI\)](#)

```
aws ec2 describe-instances --instance-id instance_id --query
'Reservations[].[Instances[]].EnaSupport'
```

- [Get-EC2Instance \(Windows PowerShell용 도구\)](#)

```
(Get-EC2Instance -InstanceId instance_id).Instances.EnaSupport
```

이미지 속성(enaSupport)

다음 명령 중 하나를 사용하여 AMI에 향상된 네트워킹 enaSupport 속성이 설정되어 있는지 확인할 수 있습니다. 속성이 설정되었으면 true가 반환됩니다.

- [describe-images](#) (AWS CLI)

```
aws ec2 describe-images --image-id ami_id --query 'Images[ ].EnaSupport'
```

- [Get-EC2Image](#) (Windows PowerShell용 도구)

```
(Get-EC2Image -ImageId ami_id).EnaSupport
```

네트워크 인터페이스 드라이버

다음 명령을 사용하여 해당 인터페이스에서 ena 모듈이 사용되고 있는지를 확인할 수 있습니다. 인터페이스 이름을 확인하려는 인터페이스 이름으로 대체하십시오. 단일 인터페이스를 사용하는 경우(기본 설정), eth0으로 표시됩니다.

```
[ec2-user ~]$ ethtool -i eth0
driver: vif
version:
firmware-version:
bus-info: vif-0
supports-statistics: yes
supports-test: no
supports-eeprom-access: no
supports-register-dump: no
supports-priv-flags: no
```

위의 경우, 표시된 드라이버가 vif이므로 ena 모듈이 로드되지 않은 것입니다.

```
[ec2-user ~]$ ethtool -i eth0
driver: ena
version: 0.6.6
firmware-version:
bus-info: 0000:00:03.0
supports-statistics: yes
supports-test: no
supports-eeprom-access: no
supports-register-dump: no
supports-priv-flags: no
```

이 경우, ena 모듈이 이미 설치되었고 최소 버전 요구를 충족하는 것을 알 수 있습니다. 이 인스턴스는 향상된 네트워킹이 올바르게 구성된 상태입니다.

Amazon Linux에서 ENA를 사용하여 향상된 네트워킹 기능 활성화

최신 Amazon Linux HVM AMI에는 ENA를 사용하는 향상된 네트워킹에 요구되는 모듈이 설치되어 있으며 필요한 enaSupport 속성 세트를 갖추고 있습니다. 따라서 지원되는 인스턴스 유형에서 최신 Amazon Linux HVM AMI를 사용하여 인스턴스를 시작하면, ENA를 사용하는 향상된 네트워크 기능이 이미 해당 인스턴스에서 활성화된 상태입니다. 자세한 내용은 [ENA를 통한 향상된 네트워킹 기능 활성화 여부 확인 \(p. 544\)](#) 섹션을 참조하십시오.

지난 세대의 Amazon Linux AMI를 사용하여 인스턴스를 시작했고 향상된 네트워크 기능이 활성화되어 있지 않은 경우에는 다음 절차에 따라 향상된 네트워크를 활성화할 수 있습니다.

ENA를 통한 향상된 네트워킹 기능을 활성화하려면(EBS 기반 인스턴스)

1. 인스턴스에 연결합니다.
2. 인스턴스에서 다음 명령을 사용하여 기존 인스턴스를 ena를 비롯한 최신 커널과 커널 모듈로 업데이트 합니다.

```
[ec2-user ~]$ sudo yum update
```

3. 로컬 컴퓨터를 사용하는 경우, Amazon EC2 콘솔을 사용하거나 다음 명령 중 하나를 사용하여 인스턴스를 재부팅하십시오. [reboot-instances](#) (AWS CLI), [Restart-EC2Instance](#)(Windows PowerShell용 AWS 도구).
4. 인스턴스에 다시 연결한 후 [ENA를 통한 향상된 네트워킹 기능 활성화 여부 확인 \(p. 544\)](#)에서 modinfo ena 명령을 사용하여 ena 모듈이 설치되어 있고 최소 권장 버전 요구를 충족하는지를 확인합니다.
5. 로컬 컴퓨터를 사용하는 경우, Amazon EC2 콘솔을 사용하거나 다음 명령 중 하나를 사용하여 인스턴스를 중지하십시오. [stop-instances](#) (AWS CLI), [Stop-EC2Instance](#)(Windows PowerShell용 AWS 도구). 인스턴스를 AWS OpsWorks에서 관리할 경우 AWS OpsWorks 콘솔에서 인스턴스를 중지해야 인스턴스 상태가 동기화됩니다.

Important

인스턴스 스토어 지원 인스턴스를 사용할 때는 인스턴스를 중지할 수 없습니다. 이 경우, [ENA를 사용하여 향상된 네트워킹을 활성화하려면\(인스턴스 스토어 지원 인스턴스\) \(p. 546\)](#) 단계로 넘어갑니다.

6. 사용자의 로컬 컴퓨터에서 다음 명령 중 하나를 사용하여 향상된 네트워크 속성을 활성화합니다.

Warning

향상된 네트워킹 기능은 HVM 인스턴스에서만 지원됩니다. PV 인스턴스에서 향상된 네트워킹 기능을 활성화하면 인스턴스 접속이 불가능해질 수 있습니다. 올바른 모듈과 모듈 버전을 사용하지 않고 속성을 설정하는 경우에도 인스턴스 접속이 불가능해질 수 있습니다.

- [modify-instance-attribute](#) (AWS CLI)

```
aws ec2 modify-instance-attribute --instance-id instance_id --ena-support
```

- [Edit-EC2InstanceAttribute](#) (Windows PowerShell용 도구)

```
Edit-EC2InstanceAttribute -InstanceId instance_id -EnaSupport $true
```

7. (선택 사항) [Amazon EBS 지원 Linux AMI 생성 \(p. 81\)](#)의 설명에 따라 인스턴스에서 AMI를 생성합니다. 생성된 AMI는 인스턴스의 향상된 네트워킹 enaSupport 속성을 상속합니다. 따라서 이 AMI를 사용하여 ENA를 사용하는 향상된 네트워킹 기능이 기본적으로 활성화된 상태로 다른 인스턴스를 시작할 수 있습니다.
8. 로컬 컴퓨터를 사용하는 경우, Amazon EC2 콘솔을 사용하거나 다음 명령 중 하나를 사용하여 인스턴스를 시작하십시오. [start-instances](#)(AWS CLI), [Start-EC2Instance](#)(Windows PowerShell용 AWS 도구). 인스턴스를 AWS OpsWorks에서 관리할 경우 AWS OpsWorks 콘솔에서 인스턴스를 시작해야 인스턴스 상태가 동기화됩니다.
9. 인스턴스에 다시 연결한 후 [ENA를 통한 향상된 네트워킹 기능 활성화 여부 확인 \(p. 544\)](#)에서 ethtool -i ethn 명령을 사용하여 ena 모듈이 설치되어 있고 최소 권장 버전 요구를 충족하는지를 확인합니다.

ENA를 사용하여 향상된 네트워킹을 활성화한 이후에 인스턴스에 연결할 수 없는 경우 [ENA\(Elastic Network Adapter\) 문제 해결 \(p. 552\)](#) 섹션을 참조하십시오.

ENA를 사용하여 향상된 네트워킹을 활성화하려면(인스턴스 스토어 지원 인스턴스)

인스턴스 스토어 지원 인스턴스인 경우, 이전 절차의 [Step 1 \(p. 546\)](#)부터 [Step 4 \(p. 546\)](#)까지 실행한 다음 [인스턴스 스토어 지원 Linux AMI 생성](#)에 설명된 대로 새 AMI를 만듭니다. AMI를 등록할 때 향상된 네트워킹 enaSupport 속성을 활성화해야 합니다.

- [register-image\(AWS CLI\)](#)

```
aws ec2 register-image --ena-support ...
```

- [Register-EC2Image \(Windows PowerShell용 AWS 도구\)](#)

```
Register-EC2Image -EnaSupport $true ...
```

Ubuntu에서 ENA를 사용하여 향상된 네트워킹 활성화

다음 절차는 Ubuntu 인스턴스에서 ENA를 사용하여 향상된 네트워킹을 활성화하는 일반적인 방법입니다.

Ubuntu에서 ENA를 사용하여 향상된 네트워킹 기능을 활성화하려면(EBS 지원 인스턴스)

1. 인스턴스에 연결합니다.
2. 패키지 캐시와 패키지를 업데이트합니다.

```
ubuntu:~$ sudo apt-get update && sudo apt-get upgrade -y
```

Important

업데이트 과정에서 grub 설치 메시지가 표시되는 경우, /dev/xvda를 사용하여 grub를 설치하고, /boot/grub/menu.lst의 현재 버전을 유지하도록 선택합니다.

3. 커널 모듈을 컴파일하도록 build-essential 패키지를 설치하고 커널을 업데이트할 때마다 ena 모듈이 다시 빌드되도록 dkms 패키지를 설치합니다.

```
ubuntu:~$ sudo apt-get install -y build-essential dkms
```

4. <https://github.com/amzn/amzn-drivers>의 GitHub로부터 해당 인스턴스의 ena 모듈에 대한 소스 코드를 복제합니다.

```
ubuntu:~$ git clone https://github.com/amzn/amzn-drivers
```

5. amzn-drivers 패키지를 /usr/src/ 디렉터리로 이동하여 커널이 업데이트될 때마다 dkms에서 파일을 찾아 빌드할 수 있도록 합니다. 디렉터리 이름에 소스 코드의 버전 번호(릴리스 정보에서 현재 버전 번호 확인 가능)를 추가합니다. 예를 들어 1.0.0 버전은 아래 예시와 같이 표시됩니다.

```
ubuntu:~$ sudo mv amzn-drivers /usr/src/amzn-drivers-1.0.0
```

6. 다음 값을 사용하여 dkms 구성 파일을 생성합니다. 버전 값은 현재 사용 중인 ena 버전을 입력하십시오.
 - a. 파일을 생성합니다.

```
ubuntu:~$ sudo touch /usr/src/amzn-drivers-1.0.0/dkms.conf
```

- b. 파일을 수정하고 다음 값을 추가합니다.

```
ubuntu:~$ sudo vim /usr/src/amzn-drivers-1.0.0/dkms.conf
PACKAGE_NAME="ena"
PACKAGE_VERSION="1.0.0"
CLEAN="make -C kernel/linux/ena clean"
```

```
MAKE="make -C kernel/linux/ena/ BUILD_KERNEL=$(kernelver)"  
BUILT_MODULE_NAME[0]="ena"  
BUILT_MODULE_LOCATION="kernel/linux/ena"  
DEST_MODULE_LOCATION[0]="/updates"  
DEST_MODULE_NAME[0]="ena"  
AUTOINSTALL="yes"
```

7. dkms를 사용하여 인스턴스에 ena 모듈을 추가 및 빌드하고 설치합니다.

- a. 모듈을 dkms에 추가합니다.

```
ubuntu:~$ sudo dkms add -m amzn-drivers -v 1.0.0
```

- b. dkms를 사용하여 모듈을 빌드합니다.

```
ubuntu:~$ sudo dkms build -m amzn-drivers -v 1.0.0
```

- c. dkms를 사용하여 모듈을 설치합니다.

```
ubuntu:~$ sudo dkms install -m amzn-drivers -v 1.0.0
```

8. 부팅 시 올바른 모듈이 로드되도록 initramfs를 다시 빌드합니다.

```
ubuntu:~$ sudo update-initramfs -c -k all
```

9. ENA를 통한 향상된 네트워킹 기능 활성화 여부 확인 (p. 544)의 modinfo ena 명령을 사용하여 ena 모듈이 설치되어 있는지 확인합니다.

```
ubuntu:~$ modinfo ena  
filename:      /lib/modules/3.13.0-74-generic/updates/dkms/ena.ko  
version:       1.0.0  
license:       GPL  
description:   Elastic Network Adapter (ENA)  
author:        Amazon.com, Inc. or its affiliates  
srcversion:    9693C876C54CA64AE48F0CA  
alias:         pci:v00001D0Fd0000EC21sv*sd*bc*sc*i*  
alias:         pci:v00001D0Fd0000EC20sv*sd*bc*sc*i*  
alias:         pci:v00001D0Fd00001EC2sv*sd*bc*sc*i*  
alias:         pci:v00001D0Fd00000EC2sv*sd*bc*sc*i*  
depends:  
vermagic:     3.13.0-74-generic SMP mod_unload modversions  
parm:          debug:Debug level (0=none,...,16=all) (int)  
parm:          push_mode:Descriptor / header push mode  
(0=automatic,1=disable,3=enable)  
          0 - Automatically choose according to device capability (default)  
          1 - Don't push anything to device memory  
          3 - Push descriptors and header buffer to device memory (int)  
parm:          enable_wd:Enable keepalive watchdog (0=disable,1=enable,default=1)  
(int)  
parm:          enable_missing_tx_detection:Enable missing Tx completions. (default=1)  
(int)  
parm:          numa_node_override_array:Numa node override map  
(array of int)  
parm:          numa_node_override:Enable/Disable numa node override (0=disable)  
(int)
```

10. 로컬 컴퓨터를 사용하는 경우, Amazon EC2 콘솔을 사용하거나 다음 명령 중 하나를 사용하여 인스턴스를 중지하십시오. [stop-instances](#) (AWS CLI), [Stop-EC2Instance](#) (Windows PowerShell용 AWS 도구). 인스턴스를 AWS OpsWorks에서 관리할 경우 AWS OpsWorks 콘솔에서 인스턴스를 중지해야 인스턴스 상태가 동기화됩니다.

Important

인스턴스 스토어 지원 인스턴스를 사용할 때는 인스턴스를 중지할 수 없습니다. 이 경우, [Ubuntu에서 ENA를 사용하여 향상된 네트워킹 기능 활성화\(인스턴스 스토어 지원 인스턴스\) \(p. 549\)](#) 단계로 넘어갑니다.

11. 로컬 컴퓨터에서 다음 명령을 사용하여 향상된 네트워킹 속성을 활성화합니다.

Warning

향상된 네트워킹 기능은 HVM 인스턴스에서만 지원됩니다. PV 인스턴스에서 향상된 네트워킹 기능을 활성화하면 인스턴스 접속이 불가능해질 수 있습니다. 올바른 모듈과 모듈 버전을 사용하지 않고 속성을 설정하는 경우에도 인스턴스 접속이 불가능해질 수 있습니다.

- [modify-instance-attribute \(AWS CLI\)](#)

```
aws ec2 modify-instance-attribute --instance-id instance_id --ena-support
```

- [Edit-EC2InstanceAttribute \(Windows PowerShell용 도구\)](#)

```
Edit-EC2InstanceAttribute -InstanceId instance_id -EnaSupport $true
```

12. (선택 사항) [Amazon EBS 지원 Linux AMI 생성 \(p. 81\)](#)의 설명에 따라 인스턴스에서 AMI를 생성합니다. 생성된 AMI는 인스턴스의 향상된 네트워크 속성을 상속합니다. 따라서 이 AMI를 사용하여 기본적으로 향상된 네트워킹 기능이 활성화된 상태로 다른 인스턴스를 시작할 수 있습니다.
13. 로컬 컴퓨터를 사용하는 경우, Amazon EC2 콘솔을 사용하거나 다음 명령 중 하나를 사용하여 인스턴스를 시작하십시오. [start-instances\(AWS CLI\)](#), [Start-EC2Instance\(Windows PowerShell용 AWS 도구\)](#). 인스턴스를 AWS OpsWorks에서 관리할 경우 AWS OpsWorks 콘솔에서 인스턴스를 시작해야 인스턴스 상태가 동기화됩니다.
14. (선택 사항) 인스턴스에 연결하여 모듈의 설치 여부를 확인합니다.

ENA를 사용하여 향상된 네트워킹을 활성화한 이후에 인스턴스에 연결할 수 없는 경우 [ENA\(Elastic Network Adapter\) 문제 해결 \(p. 552\)](#) 섹션을 참조하십시오.

Ubuntu에서 ENA를 사용하여 향상된 네트워킹 기능 활성화(인스턴스 스토어 지원 인스턴스)

인스턴스 지원 인스턴스를 사용하는 경우, 전 과정의 내용 중 [Step 1 \(p. 547\)](#)에서 [Step 9 \(p. 548\)](#)까지 수 행한 다음 [인스턴스 스토어 기반 Linux AMI 생성 \(p. 84\)](#)의 설명에 따라 새 AMI를 생성합니다. AMI를 등록할 때 향상된 네트워킹 enaSupport 속성을 활성화해야 합니다.

Warning

향상된 네트워킹 기능은 HVM 인스턴스에서만 지원됩니다. PV 인스턴스에서 향상된 네트워킹 기능을 활성화하면 인스턴스 접속이 불가능해질 수 있습니다. 올바른 모듈과 모듈 버전을 사용하지 않고 속성을 설정하는 경우에도 인스턴스 접속이 불가능해질 수 있습니다.

- [register-image\(AWS CLI\)](#)

```
aws ec2 register-image --ena-support ...
```

- [Register-EC2Image \(Windows PowerShell용 AWS 도구\)](#)

```
Register-EC2Image -EnaSupport $true ...
```

다른 Linux 배포판에서 ENA를 사용하여 향상된 네트워킹 기능 활성화

다음 절차는 Amazon Linux 또는 Ubuntu를 제외한 다른 Linux 배포판에서 ENA를 사용하여 향상된 네트워킹을 활성화하는 일반적인 방법입니다. 명령 구문과 파일 위치, 패키지 및 도구 지원을 비롯한 자세한 내용은 사용 Linux 배포판의 전용 문서를 참조하십시오.

Linux에서 ENA를 사용하여 향상된 네트워킹 기능을 활성화하려면(EBS 지원 인스턴스)

1. 인스턴스에 연결합니다.
2. <https://github.com/amzn/amzn-drivers>의 GitHub로부터 해당 인스턴스의 ena 모듈에 대한 소스 코드를 복제합니다.

```
ubuntu:~$ git clone https://github.com/amzn/amzn-drivers
```

3. 인스턴스에서 ena 모듈을 컴파일하고 설치합니다.

사용 중인 배포판에서 dkms를 지원하는 경우, 시스템 커널이 업데이트될 때마다 ena 모듈이 재컴파일되도록 dkms를 구성하는 것이 좋습니다. 사용 중인 배포판에서 dkms를 기본 지원하지 않는 경우 EPEL 리포지토리(<https://fedoraproject.org/wiki/EPEL>)에서 Red Hat Enterprise Linux 버전용을 검색하거나 <http://linux.dell.com/dkms/>에서 소프트웨어를 다운로드할 수 있습니다. [Ubuntu에서 ENA를 사용하여 향상된 네트워킹 기능을 활성화하려면\(EBS 지원 인스턴스\) \(p. 547\)](#) 내용 중 Step 5 (p. 547) ~ Step 7 (p. 548) 과정을 참고하여 dkms 구성하십시오.

4. sudo depmod 명령을 실행하여 모듈 의존관계를 업데이트합니다.
5. 인스턴스에서 initramfs를 업데이트하여 부팅 시 새 모듈이 로드되도록 합니다.
6. 시스템이 예측 가능한 네트워크 인터페이스 이름을 기본으로 사용하는지 확인합니다. 사용하는 systemd 또는 udev 버전이 197 이상인 시스템에서는 이더넷 디바이스의 이름 변경이 가능해 단일 네트워크 인터페이스가 아닌 경우에도 eth0 이름이 할당될 수 있습니다. 이에 따라 인스턴스 연결에 문제가 발생할 수 있습니다. 자세한 내용과 다른 구성 옵션을 보려면 freedesktop.org 웹 사이트에서 [예측 가능한 네트워크 인터페이스 이름](#)을 참조하십시오.

- a. RPM 기반 시스템에서는 다음 명령을 사용하여 systemd 또는 udev 버전을 확인할 수 있습니다.

```
[ec2-user ~]$ rpm -qa | grep -e '^systemd-[0-9]+\+|udev-[0-9]+\+'  
systemd-208-11.el7_0.2.x86_64
```

위의 Red Hat 7 예제에서, systemd 버전은 208이므로, 해당 네트워크 인터페이스 이름을 비활성해야 합니다.

- b. net.ifnames=0/etc/default/grubGRUB_CMDLINE_LINUX의 ## 옵션을 추가하여 예측 가능한 네트워크 인터페이스 이름을 비활성화합니다.

```
[ec2-user ~]$ sudo sed -i '/^GRUB\_CMDLINE\_LINUX/s/"$/ net\.ifnames\=0"/' /etc/default/grub
```

- c. GRUB 구성 파일을 재구축합니다.

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

7. 로컬 컴퓨터를 사용하는 경우, Amazon EC2 콘솔을 사용하거나 다음 명령 중 하나를 사용하여 인스턴스를 중지하십시오. [stop-instances](#) (AWS CLI), [Stop-EC2Instance](#)(Windows PowerShell용 AWS 도구). 인스턴스를 AWS OpsWorks에서 관리할 경우 AWS OpsWorks 콘솔에서 인스턴스를 중지해야 인스턴스 상태가 동기화됩니다.

Important

인스턴스 스토어 지원 인스턴스를 사용할 때는 인스턴스를 중지할 수 없습니다. 이 경우, [ENA를 사용하여 향상된 네트워킹을 활성화하려면\(인스턴스 스토어 지원 인스턴스\) \(p. 551\)](#) 단계로 넘어갑니다.

8. 사용자의 로컬 컴퓨터에서 다음 명령 중 하나를 사용하여 향상된 네트워크 enaSupport 속성을 활성화합니다.

Warning

향상된 네트워킹 기능은 HVM 인스턴스에서만 지원됩니다. PV 인스턴스에서 향상된 네트워킹 기능을 활성화하면 인스턴스 접속이 불가능해질 수 있습니다. 올바른 모듈과 모듈 버전을 사용하지 않고 속성을 설정하는 경우에도 인스턴스 접속이 불가능해질 수 있습니다.

- [modify-instance-attribute \(AWS CLI\)](#)

```
aws ec2 modify-instance-attribute --instance-id instance_id --ena-support
```

- [Edit-EC2InstanceAttribute \(Windows PowerShell용 도구\)](#)

```
Edit-EC2InstanceAttribute -InstanceId instance_id -EnaSupport $true
```

9. (선택 사항) [Amazon EBS 지원 Linux AMI 생성 \(p. 81\)](#)의 설명에 따라 인스턴스에서 AMI를 생성합니다. 생성된 AMI는 인스턴스의 향상된 네트워크 enaSupport 속성을 상속합니다. 따라서 이 AMI를 사용하여 기본적으로 향상된 네트워킹 기능이 활성화된 상태로 다른 인스턴스를 시작할 수 있습니다.

Important

인스턴스 운영 체제에 /etc/udev/rules.d/70-persistent-net.rules 파일이 포함되어 있는 경우 AMI를 생성하기 전에 먼저 삭제해야 합니다. 이 파일에 원본 인스턴스의 이더넷 어댑터에 대한 MAC 주소가 포함되어 있습니다. 이 파일로 다른 인스턴스가 부팅되면 운영 체제에서 디바이스를 찾을 수 없으며 eth0이 실패하여 부팅 문제가 발생할 수 있습니다. 이 파일은 다음 부팅 주기에 생성되고 AMI에서 시작된 모든 인스턴스가 자체 버전의 파일을 생성합니다.

10. 로컬 컴퓨터를 사용하는 경우, Amazon EC2 콘솔을 사용하거나 다음 명령 중 하나를 사용하여 인스턴스를 시작하십시오. [start-instances\(AWS CLI\)](#), [Start-EC2Instance\(Windows PowerShell용 AWS 도구\)](#). 인스턴스를 AWS OpsWorks에서 관리할 경우 AWS OpsWorks 콘솔에서 인스턴스를 시작해야 인스턴스 상태가 동기화됩니다.
11. (선택 사항) 인스턴스에 연결하여 모듈의 설치 여부를 확인합니다.

ENA를 사용하여 향상된 네트워킹을 활성화한 이후에 인스턴스에 연결할 수 없는 경우 [ENA\(Elastic Network Adapter\) 문제 해결 \(p. 552\)](#) 섹션을 참조하십시오.

ENA를 사용하여 향상된 네트워킹을 활성화하려면(인스턴스 스토어 지원 인스턴스)

인스턴스 지원 인스턴스를 사용하는 경우, 전 과정의 내용 중 [Step 1 \(p. 550\)](#)에서 [Step 5 \(p. 550\)](#)까지 수 행한 다음 [인스턴스 스토어 기반 Linux AMI 생성 \(p. 84\)](#)의 설명에 따라 새 AMI를 생성합니다. AMI를 등록할 때 향상된 네트워킹 enaSupport 속성을 활성화해야 합니다.

Warning

향상된 네트워킹 기능은 HVM 인스턴스에서만 지원됩니다. PV 인스턴스에서 향상된 네트워킹 기능을 활성화하면 인스턴스 접속이 불가능해질 수 있습니다. 올바른 모듈과 모듈 버전을 사용하지 않고 속성을 설정하는 경우에도 인스턴스 접속이 불가능해질 수 있습니다.

- [register-image\(AWS CLI\)](#)

```
aws ec2 register-image --ena-support ...
```

- Register-EC2Image (Windows PowerShell용 AWS 도구)

```
Register-EC2Image -EnaSupport ...
```

문제 해결

ENA 어댑터 문제 해결에 대한 자세한 내용은 [ENA\(Elastic Network Adapter\) 문제 해결 \(p. 552\)](#) 섹션을 참조하십시오.

ENA(Elastic Network Adapter) 문제 해결

ENA(Elastic Network Adapter)는 운영 체제의 상태를 향상하고 예기치 못한 하드웨어 동작이나 오류로 인한 장기적 중단 가능성을 줄이도록 설계되었습니다. ENA 아키텍처는 디바이스 또는 드라이버 장애가 시스템에 영향을 주지 않도록 최대한 보호합니다. 이 주제에서는 ENA에 대한 문제 해결 정보를 제공합니다.

인스턴스에 연결할 수 없는 경우 [연결 문제 해결 \(p. 552\)](#) 섹션에서 시작하십시오.

인스턴스에 연결할 수 있는 경우 이 주제의 이후 섹션에서 다루는 장애 탐지 및 복구 메커니즘을 사용하여 진단 정보를 수집할 수 있습니다.

목차

- [연결 문제 해결 \(p. 552\)](#)
- [연결 유지 메커니즘 \(p. 553\)](#)
- [레지스터 읽기 시간 초과 \(p. 554\)](#)
- [통계 \(p. 554\)](#)
- [syslog의 드라이버 오류 로그 \(p. 557\)](#)

연결 문제 해결

향상된 네트워킹 기능을 활성화하는 도중 연결이 해제된 경우, 인스턴스의 현재 실행 중인 커널이 ena 모듈과 호환되지 않아 발생한 문제일 수 있습니다. 이 문제는 dkms가 없거나 dkms.conf 파일이 잘못 구성된 특정 커널 버전용 모듈을 설치한 이후에 인스턴스 커널이 업데이트된 경우에 발생할 수 있습니다. 부팅 시 로드되는 인스턴스 커널에서 ena 모듈을 올바르게 설치하지 않는 경우 인스턴스에서 네트워크 어댑터를 인식하지 못하여 인스턴스에 접속할 수 없습니다.

PV 또는 AMI 인스턴스에서 향상된 네트워킹 기능을 활성화하면 인스턴스 접속이 불가능해질 수도 있습니다.

ENA를 사용하여 향상된 네트워킹을 활성화한 이후에 인스턴스에 접속할 수 없는 경우 인스턴스에 대한 enaSupport 속성을 비활성화할 수 있습니다. 그러면 스톡 네트워크 어댑터로 대체하여 사용됩니다.

ENA를 사용하여 향상된 네트워킹을 비활성화하려면(EBS 기반 인스턴스)

1. 로컬 컴퓨터를 사용하는 경우, Amazon EC2 콘솔을 사용하거나 다음 명령 중 하나를 사용하여 인스턴스를 중지하십시오. [stop-instances](#) (AWS CLI), [Stop-EC2Instance](#)(Windows PowerShell용 AWS 도구). 인스턴스를 AWS OpsWorks에서 관리할 경우 AWS OpsWorks 콘솔에서 인스턴스를 중지해야 인스턴스 상태가 동기화됩니다.

Important

인스턴스 스토어 지원 인스턴스를 사용할 때는 인스턴스를 중지할 수 없습니다. 이 경우, [ENA를 사용하여 향상된 네트워킹을 비활성화하려면\(인스턴스 스토어 지원 인스턴스\) \(p. 553\)](#) 단계로 넘어갑니다.

2. 로컬 컴퓨터에서 다음 명령을 사용하여 향상된 네트워크 속성을 비활성화합니다.

- [modify-instance-attribute \(AWS CLI\)](#)

```
$ aws ec2 modify-instance-attribute --instance-id instance_id --no-ena-support
```

3. 로컬 컴퓨터를 사용하는 경우, Amazon EC2 콘솔을 사용하거나 다음 명령 중 하나를 사용하여 인스턴스를 시작하십시오. [start-instances\(AWS CLI\)](#), [Start-EC2Instance\(Windows PowerShell용 AWS 도구\)](#). 인스턴스를 AWS OpsWorks에서 관리할 경우 AWS OpsWorks 콘솔에서 인스턴스를 시작해야 인스턴스 상태가 동기화됩니다.
4. (선택 사항) 인스턴스에 연결한 후 [VPC의 Linux 인스턴스에서 ENA\(Elastic Network Adapter\)를 사용하여 향상된 네트워킹 활성화 \(p. 543\)](#)의 단계에 따라 현재 커널 버전으로 ena 모듈을 다시 설치해 보십시오.

ENA를 사용하여 향상된 네트워킹을 비활성화하려면(인스턴스 스토어 지원 인스턴스)

인스턴스 스토어 지원 인스턴스를 사용 중인 경우 [인스턴스 스토어 기반 Linux AMI 생성 \(p. 84\)](#)에 설명된 대로 새 AMI를 만듭니다. AMI를 등록할 때 향상된 네트워킹 enaSupport 속성을 비활성화해야 합니다.

- [register-image\(AWS CLI\)](#)

```
$ aws ec2 register-image --no-ena-support ...
```

- [Register-EC2Image \(Windows PowerShell용 AWS 도구\)](#)

```
C:\> Register-EC2Image -EnaSupport $false ...
```

연결 유지 메커니즘

ENA 디바이스는 고정된 속도(일반적으로 1초당 한 번)로 연결 유지 이벤트를 게시합니다. ENA 드라이버는 감시 메커니즘을 구현하여 매번 연결 유지 메시지가 있는지를 확인합니다. 메시지가 있으면 감시를 다시 강화하고, 그렇지 않으면 드라이버에서 디바이스에 오류가 발생한 것으로 간주하고 다음을 수행합니다.

- 현재 통계를 syslog에 덤프
- ENA 디바이스 초기화
- ENA 드라이버 상태 초기화

위 초기화 절차로 인해 잠시 동안 일부 트래픽 손실이 발생할 수 있지만(TCP 연결을 통해 복구 가능) 사용자에게는 영향을 주지 않아야 합니다.

ENA 디바이스는 연결 유지 알림을 전송하지 않아 디바이스 초기화 절차를 간접적으로 요청할 수도 있습니다(예: 복구할 수 없는 구성으로 로드한 이후에 ENA 디바이스의 상태를 알 수 없는 경우).

다음은 초기화 절차에 대한 예시입니다.

```
[18509.800135] ena 0000:00:07.0 eth1: Keep alive watchdog timeout. // The watchdog process initiates a reset
[18509.815244] ena 0000:00:07.0 eth1: Trigger reset is on
[18509.825589] ena 0000:00:07.0 eth1: tx_timeout: 0 // The driver logs the current statistics
[18509.834253] ena 0000:00:07.0 eth1: io_suspend: 0
[18509.842674] ena 0000:00:07.0 eth1: io_resume: 0
[18509.850275] ena 0000:00:07.0 eth1: wd_expired: 1
[18509.857855] ena 0000:00:07.0 eth1: interface_up: 1
[18509.865415] ena 0000:00:07.0 eth1: interface_down: 0
[18509.873468] ena 0000:00:07.0 eth1: admin_q_pause: 0
```

```
[18509.881075] ena 0000:00:07.0 eth1: queue_0_tx_cnt: 0
[18509.888629] ena 0000:00:07.0 eth1: queue_0_tx_bytes: 0
[18509.895286] ena 0000:00:07.0 eth1: queue_0_tx_queue_stop: 0
.....
.....
[18511.280972] ena 0000:00:07.0 eth1: free uncompleted tx skb qid 3 idx 0x7 // At the end
of the down process, the driver discards incomplete packets.
[18511.420112] [ENA_COM: ena_com_validate_version] ena device version: 0.10 //The driver
begins its up process
[18511.420119] [ENA_COM: ena_com_validate_version] ena controller version: 0.0.1
implementation version 1
[18511.420127] [ENA_COM: ena_com_admin_init] ena_defs : Version:[b9692e8] Build date [Wed
Apr 6 09:54:21 IDT 2016]
[18512.252108] ena 0000:00:07.0: Device watchdog is Enabled
[18512.674877] ena 0000:00:07.0: irq 46 for MSI/MSI-X
[18512.674933] ena 0000:00:07.0: irq 47 for MSI/MSI-X
[18512.674990] ena 0000:00:07.0: irq 48 for MSI/MSI-X
[18512.675037] ena 0000:00:07.0: irq 49 for MSI/MSI-X
[18512.675085] ena 0000:00:07.0: irq 50 for MSI/MSI-X
[18512.675141] ena 0000:00:07.0: irq 51 for MSI/MSI-X
[18512.675188] ena 0000:00:07.0: irq 52 for MSI/MSI-X
[18512.675233] ena 0000:00:07.0: irq 53 for MSI/MSI-X
[18512.675279] ena 0000:00:07.0: irq 54 for MSI/MSI-X
[18512.772641] [ENA_COM: ena_com_set_hash_function] Feature 10 isn't supported
[18512.772647] [ENA_COM: ena_com_set_hash_ctrl] Feature 18 isn't supported
[18512.775945] ena 0000:00:07.0: Device reset completed successfully // The reset process
is complete
```

레지스터 읽기 시간 초과

ENA 아키텍처는 MMIO(Memory Mapped I/O) 읽기 작업의 제한된 사용을 제안합니다. ENA 디바이스 드라이버는 초기화 절차 중에만 MMIO 레지스터에 액세스합니다.

dmesg 출력으로 제공되는 드라이버 로그에 읽기 작업 실패가 표시되는 경우 호환되지 않거나 잘못 컴파일된 드라이버, 사용 중인 하드웨어 디바이스 또는 하드웨어 장애가 원인일 수 있습니다.

읽기 작업 실패를 나타내는 자주 끊기는 로그 항목을 문제로 간주해서는 안 됩니다. 이 경우 드라이버에서는 읽기 작업을 다시 시도합니다. 읽기 실패가 포함된 로그 항목이 잇따라 나타날 경우 드라이버 또는 하드웨어 문제를 나타냅니다.

다음은 시간 초과로 인한 읽기 작업 실패를 나타내는 드라이버 로그 항목의 예시입니다.

```
[ 47.113698] [ENA_COM: ena_com_reg_bar_read32] reading reg failed for timeout. expected:
req id[1] offset[88] actual: req id[57006] offset[0]
[ 47.333715] [ENA_COM: ena_com_reg_bar_read32] reading reg failed for timeout. expected:
req id[2] offset[8] actual: req id[57007] offset[0]
[ 47.346221] [ENA_COM: ena_com_dev_reset] Reg read32 timeout occurred
```

통계

네트워크 성능이 저하되거나 지연 시간 문제가 발생할 경우 디바이스 통계를 불러온 후 확인해야 합니다. 아래와 같이 ethtool을 사용하여 이러한 통계를 가져올 수 있습니다.

```
[ec2-user ~]$ ethtool -S ethN
NIC statistics:
    tx_timeout: 0
    io_suspend: 0
    io_resume: 0
    wd_expired: 0
    interface_up: 1
    interface_down: 0
```

```
admin_q_pause: 0
queue_0_tx_cnt: 4329
queue_0_tx_bytes: 1075749
queue_0_tx_queue_stop: 0
...
```

다음은 명령 출력 파라미터입니다.

`tx_timeout: N`

Netdev 감시가 활성화된 횟수입니다.

`io_suspend: N`

지원되지 않습니다. 이 값은 항상 0이어야 합니다.

`io_resume: N`

지원되지 않습니다. 이 값은 항상 0이어야 합니다.

`wd_expired: N`

드라이버가 이전 3초 동안 연결 유지 이벤트를 수신하지 못한 횟수입니다.

`interface_up: N`

ENA 인터페이스가 표시된 횟수입니다.

`interface_down: N`

ENA 인터페이스가 중단된 횟수입니다.

`admin_q_pause: N`

관리 대기열이 불안정한 상태입니다. 이 값은 항상 0이어야 합니다.

`queue_N_tx_cnt: N`

대기열 `N`에 대해 전송된 패킷 수입니다.

`queue_N_tx_bytes: N`

대기열 `N`에 대해 전송된 바이트 수입니다.

`queue_N_tx_queue_stop: N`

대기열 `N`이 꽉 차서 중지된 횟수입니다.

`queue_N_tx_queue_wakeup: N`

대기열 `N`이 중지되었다가 재개된 횟수입니다.

`queue_N_tx_dma_mapping_err: N`

직접 메모리 액세스 오류 수입니다. 이 값이 0이 아니면 시스템 리소스가 부족한 것입니다.

`queue_N_tx_napi_comp: N`

napi 핸들러가 대기열 `N`에 대해 napi_complete을 호출한 횟수입니다.

`queue_N_tx_poll: N`

napi 핸들러가 대기열 `N`에 대해 예약된 횟수입니다.

`queue_N_tx_doorbells: N`

대기열 `N`에 대한 전송 초인종 수입니다.

`queue_N_tx_linearize: N`

대기열 `N`에 대해 SKB 선형화가 시도된 횟수입니다.

queue_<N>_tx_linearize_failed: <N>

대기열 <N>에 대해 SKB 선형화가 실패한 횟수입니다.

queue_<N>_tx_prepare_ctxt_err: <N>

대기열 <N>에 대해 ena_com_prepare_tx가 실패한 횟수입니다. 이 값은 항상 0이어야 합니다. 그렇지 않은 경우 드라이버 로그를 참조하십시오.

queue_<N>_tx_missing_tx_comp: <N>

대기열 <N>에 대해 완료되지 않은 상태로 남은 패킷 수입니다. 이 값은 항상 0이어야 합니다.

queue_<N>_tx_bad_req_id: <N>

대기열 <N>에 대해 잘못된 req_id입니다. 유효한 req_id는 0, -queue_size, -1입니다.

queue_<N>_rx_cnt: <N>

대기열 <N>에 대해 수신된 패킷 수입니다.

queue_<N>_rx_bytes: <N>

대기열 <N>에 대해 수신된 바이트 수입니다.

queue_<N>_rx_refill_partial: <N>

드라이버가 rx 대기열의 빈 부분을 <N> 대기열에 대한 버퍼로 리필하는 데 실패한 횟수입니다. 이 값이 0이 아니면 메모리 리소스가 부족한 것입니다.

queue_<N>_rx_bad_csum: <N>

rx 대기열에 <N> 대기열에 대한 잘못된 체크섬이 포함된 횟수입니다(rx 체크섬 오프로드가 지원되는 경우에만 해당).

queue_<N>_rx_page_alloc_fail: <N>

대기열 <N>에 대해 페이지 할당이 실패한 횟수입니다. 이 값이 0이 아니면 메모리 리소스가 부족한 것입니다.

queue_<N>_rx_skb_alloc_fail: <N>

대기열 <N>에 대해 SKB 할당이 실패한 횟수입니다. 이 값이 0이 아니면 시스템 리소스가 부족한 것입니다.

queue_<N>_rx_dma_mapping_err: <N>

직접 메모리 액세스 오류 수입니다. 이 값이 0이 아니면 시스템 리소스가 부족한 것입니다.

queue_<N>_rx_bad_desc_num: <N>

패킷당 버퍼가 너무 많습니다. 이 값이 0이 아니면 매우 작은 버퍼가 사용되는 것입니다.

queue_<N>_rx_small_copy_len_pkt: <N>

최적화: 패킷이 sysfs에 의해 설정되는 이 임계값보다 작은 경우 패킷이 스택에 직접 복사되어 새 페이지가 할당되는 것을 방지합니다.

ena_admin_q_aborted_cmd: <N>

중단된 관리 명령 수입니다. 이러한 상황은 일반적으로 자동 복구 절차 중에 발생합니다.

ena_admin_q_submitted_cmd: <N>

관리 대기열 초인종 수입니다.

ena_admin_q_completed_cmd: <N>

관리 대기열 완료 횟수입니다.

ena_admin_q_out_of_space: <N>

드라이버가 새 관리 명령을 시도했지만 대기열이 꽉 찬 횟수입니다.

ena_admin_q_no_completion: N

드라이버가 명령에 대한 관리 완료를 가져오지 못한 횟수입니다.

syslog의 드라이버 오류 로그

ENA 드라이버는 시스템 부팅 중에 syslog에 메시지를 기록합니다. 문제가 발생한 경우 이 로그를 조사하여 오류를 확인할 수 있습니다. 다음은 시스템 부팅 중에 ENA 드라이버가 syslog에 기록한 정보와 선택 메시지에 대한 일부 주석의 예시입니다.

```
Jun  3 22:37:46 ip-172-31-2-186 kernel: [ 478.416939] [ENA_COM: ena_com_validate_version]
ena device version: 0.10
Jun  3 22:37:46 ip-172-31-2-186 kernel: [ 478.420915] [ENA_COM: ena_com_validate_version]
ena controller version: 0.0.1 implementation version 1
Jun  3 22:37:46 ip-172-31-2-186 kernel: [ 479.256831] ena 0000:00:03.0: Device watchdog is
Enabled
Jun  3 22:37:46 ip-172-31-2-186 kernel: [ 479.672947] ena 0000:00:03.0: creating 8 io
queues. queue size: 1024
Jun  3 22:37:46 ip-172-31-2-186 kernel: [ 479.680885] [ENA_COM:
ena_com_init_interrupt_moderation] Feature 20 isn't supported // Interrupt moderation is
not supported by the device
Jun  3 22:37:46 ip-172-31-2-186 kernel: [ 479.691609] [ENA_COM: ena_com_get_feature_ex]
Feature 10 isn't supported // RSS HASH function configuration is not supported by the
device
Jun  3 22:37:46 ip-172-31-2-186 kernel: [ 479.694583] [ENA_COM: ena_com_get_feature_ex]
Feature 18 isn't supported // RSS HASH input source configuration is not supported by the
device
Jun  3 22:37:46 ip-172-31-2-186 kernel: [ 479.697433] [ENA_COM:
ena_com_set_host_attributes] Set host attribute isn't supported
Jun  3 22:37:46 ip-172-31-2-186 kernel: [ 479.701064] ena 0000:00:03.0 (unnamed
net_device) (uninitialized): Cannot set host attributes
Jun  3 22:37:46 ip-172-31-2-186 kernel: [ 479.704917] ena 0000:00:03.0: Elastic Network
Adapter (ENA) found at mem f3000000, mac addr 02:8a:3c:le:13:b5 Queues 8
Jun  3 22:37:46 ip-172-31-2-186 kernel: [ 480.805037] EXT4-fs (xvda1): re-mounted. Opts:
(null)
Jun  3 22:37:46 ip-172-31-2-186 kernel: [ 481.025842] NET: Registered protocol family 10
```

무시할 수 있는 오류는 무엇입니까?

시스템 오류 로그에 표시되는 다음 경고는 ENA에 대해 무시해도 됩니다.

Set host attribute isn't supported

호스트 속성은 이 디바이스에 대해 지원되지 않습니다.

failed to alloc buffer for rx queue

복구할 수 있는 오류이며 오류가 발생된 시점에 메모리 부족 문제가 발생했을 수 있습니다.

Feature X isn't supported

참조된 함수는 ENA에서 지원되지 않습니다. 가능한 x 값은 다음과 같습니다.

- 10: RSS 해시 함수 구성은 이 디바이스에 대해 지원되지 않습니다.
- 12: RSS 간접 테이블 구성은 이 디바이스에 대해 지원되지 않습니다.
- 18: RSS 해시 입력 구성은 이 디바이스에 대해 지원되지 않습니다.
- 20: 인터럽트 조절은 이 디바이스에 대해 지원되지 않습니다.

Failed to config AENQ

ENA가 AENQ 구성을 지원하지 않습니다.

Trying to set unsupported AENQ events

이 오류는 ENA에서 지원되지 않는 AENQ 이벤트 그룹을 설정하려고 시도했음을 나타냅니다.

스토리지

Amazon EC2는 고객의 상황에 맞춰 유연하고 비용대비 효율적이며 사용이 쉬운 데이터 스토리지 옵션을 제공합니다. 각 옵션은 성능과 내구성이 조합되어 고유하게 구성됩니다. 이러한 스토리지 옵션은 독립적으로 또는 요구 사항에 맞춰 조합하여 사용할 수 있습니다.

이 섹션을 확인한 후, Amazon EC2가 제공하는 데이터 스토리지 옵션을 활용하여 사용자의 특정 요구 사항을 충족시킬 수 있는 방법에 대해 확실하게 이해할 수 있습니다. 제공되는 스토리지 옵션:

- [Amazon Elastic Block Store\(Amazon EBS\) \(p. 560\)](#)
- [Amazon EC2 인스턴스 스토어 \(p. 642\)](#)
- [Amazon EFS\(Amazon Elastic File System\) \(p. 654\)](#)
- [Amazon Simple Storage Service\(Amazon S3\) \(p. 657\)](#)

다음 그림은 이러한 스토리지 유형 간의 관계를 보여줍니다.

Amazon EBS

Amazon EBS는 실행 중인 인스턴스에 연결할 수 있는 내구성이 뛰어난 블록 수준 스토리지 볼륨을 제공합니다. 세분화된 업데이트를 자주 수행해야 하는 데이터의 경우 Amazon EBS를 기본 스토리지 디바이스로 사용할 수 있습니다. 예를 들어, Amazon EBS는 인스턴스에서 데이터베이스를 실행할 때 권장되는 스토리지 옵션입니다.

EBS 볼륨은 단일 인스턴스에 연결할 수 있고 형식이 지정되지 않은 외부 블록 원시 디바이스와 같은 방식으로 동작합니다. 볼륨은 인스턴스의 실행 수명과 독립적으로 유지됩니다. 일단 EBS 볼륨이 인스턴스에 연결되면, 다른 물리적 하드 드라이브처럼 사용할 수 있습니다. 이전 그림에서 설명된 것과 같이 여러 볼륨을 단일 인스턴스에 연결할 수 있습니다. 한 인스턴스에서 EBS 볼륨을 분리한 다음 다른 인스턴스에 연결하는 것도 가능합니다. 인스턴스에 연결된 볼륨의 구성을 동적으로 변경할 수 있습니다. 또한 Amazon EBS 암호화 기능을 사용하여 EBS 볼륨을 암호화된 볼륨으로 생성할 수도 있습니다. 자세한 내용은 [Amazon EBS Encryption \(p. 617\)](#) 섹션을 참조하십시오.

EBS 볼륨의 스냅샷을 생성하여 Amazon S3에 저장하면 데이터의 백업 사본을 유지할 수 있습니다. 스냅샷에서 새 EBS 볼륨을 만든 후 다른 인스턴스에 연결할 수 있습니다. 자세한 내용은 [Amazon Elastic Block Store\(Amazon EBS\) \(p. 560\)](#) 섹션을 참조하십시오.

Amazon EC2 인스턴스 스토어

여러 인스턴스는 호스트 컴퓨터에 물리적으로 연결된 디스크의 스토리지에 액세스할 수 있습니다. 이러한 디스크 스토리지를 인스턴스 저장소라고 합니다. 인스턴스 스토어는 인스턴스에 볼록 수준의 임시 스토리지를 제공합니다. 인스턴스 스토어에 저장된 데이터는 연관 인스턴스의 수명 기간 동안에만 유지되고, 해당 인스턴스를 종지하거나 종료하면 인스턴스 스토어 볼륨의 데이터가 손실됩니다. 자세한 내용은 [Amazon EC2 인스턴스 스토어 \(p. 642\)](#) 섹션을 참조하십시오.

Amazon EFS 파일 시스템

Amazon EFS는 Amazon EC2에서 사용할 수 있는 확장 가능한 파일 스토리지를 제공합니다. EFS 파일 시스템을 만든 후 파일 시스템을 마운트하도록 인스턴스를 구성할 수 있습니다. 하나의 EFS 파일 시스템을 여러 인스턴스에서 실행하는 워크로드 및 애플리케이션에 대한 공통 데이터 소스로 사용할 수 있습니다. 자세한 내용은 [Amazon EFS\(Amazon Elastic File System\) \(p. 654\)](#) 섹션을 참조하십시오.

Amazon S3

Amazon S3를 활용하면 저렴하지만 신뢰성이 있는 데이터 스토리지 인프라에 액세스할 수 있습니다. S3은 언제든지 Amazon EC2 내 또는 웹의 어디서나 원하는 데이터의 양을 저장하고 가져올 수 있게 해주어 웹 규모의 컴퓨팅 작업을 쉽게 수행할 수 있도록 설계되었습니다. 예를 들어, Amazon S3를 활용하면 데이터 및 애플리케이션의 백업 사본을 저장할 수 있습니다. Amazon EC2는 Amazon S3를 사용하여 EBS 스냅샷과 인스턴스 스토어 지원 AMI를 저장합니다. 자세한 내용은 [Amazon Simple Storage Service\(Amazon S3\) \(p. 657\)](#) 섹션을 참조하십시오.

스토리지 추가

AMI에서 인스턴스를 실행할 때마다 해당 인스턴스에 대한 루트 스토리지 디바이스가 생성됩니다. 루트 스토리지 디바이스에는 인스턴스를 부팅하기 위해 필요한 모든 정보가 포함됩니다. 볼록 디바이스 매핑을 사용하면 AMI를 생성하거나 인스턴스를 실행할 때 루트 디바이스 볼륨과 스토리지 볼륨을 지정할 수 있습니다. 자세한 내용은 [볼록 디바이스 매핑 \(p. 662\)](#) 섹션을 참조하십시오.

실행 중인 인스턴스에 EBS 볼륨을 연결할 수도 있습니다. 자세한 내용은 [Amazon EBS 볼륨을 인스턴스에 연결 \(p. 576\)](#) 섹션을 참조하십시오.

Amazon Elastic Block Store(Amazon EBS)

Amazon Elastic Block Store(Amazon EBS)는 EC2 인스턴스에 사용할 수 있는 볼록 수준 스토리지 볼륨을 제공합니다. EBS 볼륨은 동일한 가용 영역에서 실행 중인 인스턴스에 연결할 수 있는 가용성이 높고 안정적인 스토리지 볼륨입니다. EC2 인스턴스에 연결된 EBS 볼륨은 스토리지 볼륨으로 표시되며, 인스턴스 수명에 관계없이 지속됩니다. Amazon EBS에서는 사용한 만큼만 지불하면 됩니다. Amazon EBS 요금에 대한 자세한 내용은 [Amazon Elastic Block Store 페이지](#)의 비용 예측 섹션을 참조하십시오.

데이터에 빠르게 액세스하고 장기적으로 지속해야 하는 경우 Amazon EBS를 사용하는 것이 좋습니다. EBS 볼륨은 세분화된 업데이트가 필요하고 형식이 지정되지 않은 볼록 수준의 원시 스토리지에 액세스해야 하는 파일 시스템, 데이터베이스 또는 애플리케이션의 기본 스토리지로 사용하기에 특히 적합합니다. Amazon EBS는 임의 읽기 및 쓰기에 의존하는 데이터베이스 스타일의 애플리케이션과 장시간의 지속적인 읽기 및 쓰기를 수행하여 처리량이 큰 애플리케이션에 모두 적합합니다.

단순 데이터 암호화의 경우 EBS 볼륨을 암호화된 볼륨으로 시작할 수 있습니다. Amazon EBS 암호화는 EBS 볼륨에 대해 키 관리 인프라를 사용자가 직접 구축, 관리 및 보호할 필요가 없는 간편한 암호화 솔루션을 제공합니다. 암호화된 EBS 볼륨을 생성하여 지원되는 인스턴스 유형에 연결하면 볼륨에 저장된 데이터, 디스크 I/O 및 볼륨에서 생성된 스냅샷이 모두 암호화됩니다. 암호화는 EC2 인스턴스를 호스팅하는 서버에서 수행되므로 EC2 인스턴스에서 EBS 스토리지로 전송되는 데이터가 암호화됩니다. 자세한 내용은 [Amazon EBS Encryption \(p. 617\)](#) 섹션을 참조하십시오.

Amazon EBS 암호화는 AWS Key Management Service(AWS KMS) 마스터 키를 사용하여 암호화된 볼륨을 생성하고 암호화된 볼륨에서 모든 스냅샷을 생성합니다. 리전에서 암호화된 EBS 볼륨을 처음 생성할 때 기본 마스터 키가 자동으로 생성됩니다. AWS Key Management Service를 사용하여 별도로 생성된 CMK(고객

마스터 키)를 선택하는 경우를 제외하고 이 키가 Amazon EBS 암호화에 사용됩니다. CMK를 직접 생성하면 액세스 제어를 생성, 교체, 비활성화, 정의하고 데이터를 보호하는데 사용된 암호화 키를 감사하는 등 보다 폭넓은 작업이 가능합니다. 자세한 내용은 [AWS Key Management Service Developer Guide](#) 섹션을 참조하십시오.

AWS 계정에 지정된 제한 내에서 동일한 인스턴스에 여러 볼륨을 연결할 수 있습니다. 계정당 사용할 수 있는 EBS 볼륨 수와 사용 가능한 총 스토리지는 제한됩니다. 이러한 제한에 대한 자세한 내용과 제한 증가를 요청하는 방법은 [Amazon EBS 인스턴스 제한 증가 요청](#)을 참조하십시오.

목차

- [Amazon EBS의 기능 \(p. 561\)](#)
- [Amazon EBS 볼륨 \(p. 562\)](#)
- [Amazon EBS 스냅샷 \(p. 607\)](#)
- [Amazon EBS 최적화 인스턴스 \(p. 614\)](#)
- [Amazon EBS Encryption \(p. 617\)](#)
- [Linux 인스턴스의 Amazon EBS 볼륨 성능 \(p. 621\)](#)
- [Amazon EBS용 Amazon CloudWatch Events \(p. 637\)](#)

Amazon EBS의 기능

- 최대 16TiB 크기의 EBS 범용 SSD(gp2), 프로비저닝된 IOPS SSD(io1), 처리량에 최적화된 HDD(st1) 및 Cold HDD(sc1) 볼륨을 만들 수 있습니다. 이러한 볼륨을 Amazon EC2 인스턴스에 디바이스로 마운트할 수 있습니다. 동일한 인스턴스에 여러 볼륨을 마운트할 수 있지만, 각 볼륨을 한 번에 하나의 인스턴스에만 연결할 수 있습니다. 인스턴스에 연결된 볼륨의 구성성을 동적으로 변경할 수 있습니다. 자세한 내용은 [Amazon EBS 볼륨 생성 \(p. 573\)](#) 섹션을 참조하십시오.
- 범용 SSD(gp2) 볼륨에서는 3 IOPS/GiB를 기본 성능으로 제공하며, 시간을 연장할 경우 최대 3,000 IOPS 까지 버스트할 수 있습니다. Gp2 볼륨은 부트 볼륨, 중소 규모 데이터베이스, 개발 및 테스트 환경 등의 광범위한 사용 사례에 적합합니다. Gp2 볼륨은 최대 10,000 IOPS 및 160MB/s의 처리량을 지원합니다. 자세한 내용은 [범용 SSD\(gp2\) 볼륨 \(p. 566\)](#) 섹션을 참조하십시오.
- 프로비저닝된 IOPS SSD(io1) 볼륨을 사용하면 특정 수준의 I/O 성능을 프로비저닝할 수 있습니다. Io1 볼륨은 최대 20,000 IOPS 및 320MB/s의 처리량을 지원합니다. 따라서 예측 가능한 방식으로 EC2 인스턴스 당 수만 IOPS까지 확장할 수 있습니다. 자세한 내용은 [프로비저닝된 IOPS SSD\(io1\) 볼륨 \(p. 567\)](#) 섹션을 참조하십시오.
- 처리량에 최적화된 HDD(st1) 볼륨은 IOPS가 아닌 처리량으로 성능을 정의하는 저비용 마그네틱 스토리지를 제공합니다. 최대 처리량이 500MiB/s인 이 볼륨 유형은 Amazon EMR, ETL, 데이터 웨어하우스, 로그 처리 같은 대용량 순차 워크로드에 적합합니다. 자세한 내용은 [처리량에 최적화된 HDD\(st1\) 볼륨 \(p. 568\)](#) 섹션을 참조하십시오.
- Cold HDD(sc1) 볼륨은 IOPS가 아닌 처리량으로 성능을 정의하는 저비용 마그네틱 스토리지를 제공합니다. 최대 처리량이 250MiB/s인 sc1은 대용량 순차 콜드 데이터 워크로드에 적합합니다. 데이터에 자주 액세스할 필요가 없고 비용을 절약해야 한다면 저렴한 블록 스토리지로 sc1이 적합합니다. 자세한 내용은 [Cold HDD\(sc1\) 볼륨 \(p. 569\)](#) 섹션을 참조하십시오.
- EBS 볼륨은 형식이 지정되지 않은 원시 블록 디바이스처럼 동작합니다. 이러한 볼륨 위에 파일 시스템을 생성하거나 하드 드라이브와 같은 블록 디바이스를 사용하는 것처럼 볼륨을 사용할 수 있습니다. 파일 시스템 생성 및 볼륨 마운트에 대한 자세한 내용은 [Amazon EBS 볼륨을 사용할 수 있도록 만들기 \(p. 577\)](#) 섹션을 참조하십시오.
- 암호화된 EBS 볼륨을 사용하여 규제/감사 데이터 및 애플리케이션에 대한 다양한 유형 데이터 암호화 요구 사항을 충족할 수 있습니다. 자세한 내용은 [Amazon EBS Encryption \(p. 617\)](#) 섹션을 참조하십시오.
- Amazon S3까지 지속되는 EBS 볼륨의 지정 시간 스냅샷을 생성할 수 있습니다. 이러한 스냅샷은 데이터를 장기적으로 안전하게 보호하며 새로운 EBS 볼륨의 시작점으로 사용할 수도 있습니다. 또한, 스냅샷을 사용하여 원하는 수만큼 볼륨을 인스턴스화할 수 있습니다. 이러한 스냅샷을 AWS 리전에서 복사할 수 있습니다. 자세한 내용은 [Amazon EBS 스냅샷 \(p. 607\)](#) 섹션을 참조하십시오.

- EBS 볼륨은 특정 가용 영역에서 생성한 후 동일한 가용 영역에 있는 아무 인스턴스에나 연결할 수 있습니다. 가용 영역 외부에 볼륨을 제공하기 위해 스냅샷을 생성하고 해당 리전 어디서나 새 볼륨으로 복원할 수 있습니다. 스냅샷을 다른 리전에 복사한 다음 새 볼륨에 복원하면 지리적 확장, 데이터 센터 마이그레이션 및 재해 복구를 위해 여러 AWS 리전을 쉽게 활용할 수 있습니다. 자세한 내용은 [Amazon EBS 스냅샷 생성 \(p. 608\)](#), [스냅샷에서 Amazon EBS 볼륨 복구 \(p. 574\)](#), [Amazon EBS 스냅샷 복사 \(p. 610\)](#) 섹션을 참조하십시오.
- 공개 데이터 세트 스냅샷의 대용량 리포지토리를 EBS 볼륨에 복원하고 AWS 클라우드 기반 애플리케이션에 완벽하게 통합할 수 있습니다. 자세한 내용은 [퍼블릭 데이터 세트 사용 \(p. 670\)](#) 섹션을 참조하십시오.
- 대역폭, 처리량, 지연 시간, 평균 대기열 길이 등의 성능 지표가 AWS Management Console을 통해 제공됩니다. Amazon CloudWatch에 의해 제공되는 이러한 지표를 통해 볼륨의 성능을 모니터링하면 필요 없는 리소스를 구입하지 않고도 애플리케이션에 충분한 성능을 제공할 수 있습니다. 자세한 내용은 [Linux 인스턴스의 Amazon EBS 볼륨 성능 \(p. 621\)](#) 섹션을 참조하십시오.

Amazon EBS 볼륨

Amazon EBS 볼륨은 내구성이 있는 블록 수준 스토리지 디바이스를 제공하여 단일 EC2 인스턴스를 연결하는 것이 가능합니다. 인스턴스의 시스템 드라이브나 데이터베이스 애플리케이션 또는 지속적인 디스크 검사를 수행하면서 처리량이 큰 애플리케이션용 스토리지 등 자주 업데이트해야 하는 데이터의 기본 스토리지로 EBS 볼륨을 사용할 수 있습니다. EBS 볼륨은 EC2 인스턴스의 실행 주기와는 독립적으로 유지됩니다. 볼륨이 인스턴스에 연결되면, 다른 물리적 하드 드라이브처럼 사용할 수 있습니다. EBS 볼륨은 유연합니다. 볼륨을 동적으로 확장하고, 프로비저닝된 IOPS 용량을 수정하고, 라이브 프로덕션 볼륨의 볼륨 유형을 변경할 수 있습니다. Amazon EBS는 범용 SSD(gp2), 프로비저닝된 IOPS SSD(io1), 처리량에 최적화된 HDD(st1), Cold HDD(sc1) 및 Magnetic(standard) 볼륨 유형을 제공합니다. 이 두 유형은 성능 특성과 가격이 다르므로 애플리케이션의 필요에 맞게 스토리지 성능과 비용을 조정할 수 있습니다. 자세한 내용은 [Amazon EBS 볼륨 유형 \(p. 564\)](#) 섹션을 참조하십시오.

목차

- [EBS 볼륨 사용의 이점 \(p. 562\)](#)
- [Amazon EBS 볼륨 유형 \(p. 564\)](#)
- [Amazon EBS 볼륨 생성 \(p. 573\)](#)
- [스냅샷에서 Amazon EBS 볼륨 복구 \(p. 574\)](#)
- [Amazon EBS 볼륨을 인스턴스에 연결 \(p. 576\)](#)
- [Amazon EBS 볼륨을 사용할 수 있도록 만들기 \(p. 577\)](#)
- [볼륨 정보 보기 \(p. 579\)](#)
- [볼륨 상태 모니터링 \(p. 580\)](#)
- [인스턴스에서 Amazon EBS 볼륨 분리 \(p. 588\)](#)
- [Amazon EBS 볼륨 삭제 \(p. 589\)](#)
- [Linux에서 EBS 볼륨의 크기, IOPS 또는 유형 수정 \(p. 590\)](#)
- [Linux 파티션 확장 \(p. 598\)](#)

EBS 볼륨 사용의 이점

EBS 볼륨은 인스턴스 스토어 볼륨과 차별화된 몇 가지 이점을 제공합니다.

- 데이터 가용성

가용 영역에서 Amazon EBS 볼륨을 생성하면 단일 하드웨어 구성 요소의 장애로 인한 데이터 손실을 방지하기 위해 해당 영역 내에서 자동으로 복제됩니다. 볼륨을 생성한 후 동일한 가용 영역에 있는 모든 EC2 인스턴스에 연결할 수 있습니다. 볼륨을 연결한 후에 인스턴스는 하드 드라이브 또는 기타 물리 드라이브와 같은 원시 블록 디바이스처럼 보입니다. 이 시점에 인스턴스는 로컬 드라이브와 동일한 방식으로 볼륨

과 상호 작용할 수 있으므로, 인스턴스가 ext3와 같은 파일 시스템으로 EBS 볼륨을 포맷한 다음 애플리케이션을 설치할 수 있습니다.

동일한 가용 영역에서 EBS 볼륨을 한 번에 한 개의 인스턴스에만 연결할 수 있습니다. 그러나 다중 볼륨은 단일 인스턴스에 연결될 수 있습니다. 사용자가 명명한 디바이스에 다중 볼륨이 연결된 경우 사용자는 I/O 및 처리 성능을 향상하기 위해 전체 볼륨에서 데이터를 스트라이프할 수 있습니다.

사용자는 추가 비용 없이 EBS 볼륨에 대한 데이터를 모니터링할 수 있습니다(이 경우 EBS 기반 인스턴스의 루트 디바이스 볼륨에 대한 데이터가 포함됨). 자세한 내용은 [CloudWatch로 볼륨 모니터링 \(p. 580\)](#) 섹션을 참조하십시오.

- 데이터 지속성

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

기본적으로, 실행 중인 인스턴스에 연결된 EBS 볼륨은 인스턴스가 종료될 때 데이터를 그대로 유지한 상태로 인스턴스로부터 자동으로 분리됩니다. 그러면 해당 볼륨은 새 인스턴스로 재연결되어 빠른 복구가 가능합니다. EBS 기반 인스턴스를 사용하는 경우 연결된 볼륨에 저장된 데이터에 영향을 주지 않고 해당 인스턴스를 중지하고 다시 시작할 수 있습니다. 해당 볼륨은 정지-시작 주기 동안 연결 상태를 유지합니다. 이를 통해 사용자는 필요할 때 처리 및 스토리지 리소스만을 사용하여 볼륨에서 데이터를 무기한으로 처리 및 저장할 수 있습니다. 데이터는 볼륨이 완전히 삭제될 때까지 볼륨에서 유지됩니다. 삭제된 EBS 볼륨에서 사용된 물리 볼록 스토리지는 다른 계정에 할당되기 전까지 0으로 덮어쓰기됩니다. 민감한 데이터를 사용하는 경우 데이터를 직접 암호화하거나 Amazon EBS 암호화으로 보호되는 볼륨에 데이터를 저장해야 합니다. 자세한 내용은 [Amazon EBS Encryption \(p. 617\)](#) 섹션을 참조하십시오.

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

- 데이터 암호화

단순 데이터 암호화의 경우 Amazon EBS 암호화 기능으로 암호화된 EBS 볼륨을 생성할 수 있습니다. 모든 EBS 볼륨 유형은 암호화를 지원합니다. 암호화된 EBS 볼륨을 사용하여 규제/감사 데이터 및 애플리케이션에 대한 다양한 유형 데이터 암호화 요구 사항을 충족할 수 있습니다. Amazon EBS 암호화는 256비트 고급 암호화 표준 알고리즘(AES-256) 및 Amazon이 관리하는 키 인프라를 사용합니다. 암호화는 EC2 인스턴스를 호스트하는 서버에서 수행되므로 EC2 인스턴스에서 Amazon EBS 스토리지로 전송되는 데이터가 암호화됩니다. 자세한 내용은 [Amazon EBS Encryption \(p. 617\)](#) 섹션을 참조하십시오.

Amazon EBS 암호화는 AWS Key Management Service(AWS KMS) 마스터 키를 사용하여 암호화된 볼륨을 생성하고 암호화된 볼륨에서 모든 스냅샷을 생성합니다. 리전에서 암호화된 EBS 볼륨을 처음 생성할 때 기본 마스터 키가 자동으로 생성됩니다. AWS KMS를 사용하여 별도로 생성된 CMK(고객 마스터 키)를 선택하는 경우를 제외하고 이 키가 Amazon EBS 암호화에 사용됩니다. CMK를 직접 생성하면 액세스 제어를 생성, 교체, 비활성화, 정의하고 데이터를 보호하는 데 사용된 암호화 키를 감사하는 등 보다 폭넓은 작업이 가능합니다. 자세한 내용은 [AWS Key Management Service Developer Guide](#) 섹션을 참조하십시오.

- 스냅샷

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

스냅샷에서 새로운 볼륨을 생성하는 경우 새로 생성된 스냅샷이 생성될 시점의 원본 볼륨 사본과 정확히 일치합니다. 암호화된 스냅샷에서 복구된 EBS 볼륨은 자동으로 암호화됩니다. 다양한 가용 영역을 지정하는 옵션이 있습니다. 이러한 기능을 사용하여 이 영역에 복제 볼륨을 생성할 수 있습니다. 스냅샷은 특정 AWS 계정과 공유하거나 공개 상태가 될 수 있습니다. 스냅샷을 생성하는 경우 볼륨의 총 크기에 따라

Amazon S3에서 비용이 발생하게 됩니다. 볼륨 스냅샷을 연속으로 생성하면 볼륨의 원본 크기와 대비하여 추가된 데이터에 해당하는 비용만이 청구됩니다.

;스냅샷은 마지막 스냅샷 이후 변경된 볼륨의 블록만 저장되는 증분식 백업입니다. 100GiB 데이터를 가진 볼륨이 있지만 마지막 스냅샷 이후 5GiB만이 변경된 경우 변경된 5GiB만이 Amazon S3에 작성됩니다. 스냅샷은 증분식으로 저장되지만 스냅샷 삭제 프로세스는 볼륨을 복구하기 위해 가장 최근의 스냅샷만을 유지할 수 있도록 설계됩니다.

볼륨 및 스냅샷을 쉽게 범주화하고 관리할 수 있도록 사용자는 원하는 메타데이터로 볼륨 및 스냅샷에 태그를 사용할 수 있습니다. 자세한 내용은 [Amazon EC2 리소스에 태그 지정 \(p. 681\)](#) 섹션을 참조하십시오.

- **유연성**

EBS 볼륨은 프로덕션 중에 라이브 구성 변경을 지원합니다. 서비스 종단 없이 볼륨 유형, 볼륨 크기, IOPS 용량을 수정할 수 있습니다.

Amazon EBS 볼륨 유형

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

- **SSD 지원 볼륨:** 작은 I/O 크기의 읽기/쓰기 작업을 자주 처리하는 트랜잭션 워크로드에 최적화되어 있으며, 기준 성능 속성은 IOPS
- **HDD 지원 볼륨:** 대용량 스트리밍 워크로드에 최적화되어 있으며, IOPS보다는 처리량(MiB/s로 측정)이 더 정확한 성능 측정 기준

다음 표는 각 볼륨 유형에 대한 사용 사례 및 성능 특성을 설명합니다.

	SSD(Solid-State Drive)		HDD(Hard disk Drive)	
볼륨 유형	범용 SSD(gp2)*	프로비저닝된 IOPS SSD(io1)	처리량에 최적화된 HDD(st1)	Cold HDD(sc1)
설명	다양한 트랜잭션 워크로드에 사용할 수 있으며 가격 대비 성능이 우수한 범용 SSD 볼륨	미션 크리티컬 애플리케이션에 적합한 고성능 SSD 볼륨	자주 액세스하는 처리량 집약적 워크로드에 적합한 저비용 HDD 볼륨	자주 액세스하지 않는 워크로드에 적합한 최저 비용 HDD 볼륨
사용 사례	<ul style="list-style-type: none">• 대부분의 워크로드에 추천• 시스템 부트 볼륨• 가상 데스크톱• 지연 시간이 짧은 대화형 앱• 개발 및 테스트 환경	<ul style="list-style-type: none">• 지속적인 IOPS 성능이나 10,000 IOPS 또는 160MiB/s 이상의 볼륨당 처리량을 필요로 하는 중요한 비즈니스 애플리케이션• 라지 데이터베이스 워크로드. 예:<ul style="list-style-type: none">• MongoDB• Cassandra• Microsoft SQL Server• MySQL• PostgreSQL	<ul style="list-style-type: none">• 저비용으로 일관되고 높은 처리량을 요구하는 스트리밍 워크로드• 빅 데이터• 데이터 웨어하우스• 로그 처리• 부트 볼륨이 될 수 없음	<ul style="list-style-type: none">• 자주 액세스하지 않는 대용량 데이터를 위한 처리량 중심의 스토리지• 스토리지 비용이 최대한 낮아야 하는 시나리오• 부트 볼륨이 될 수 없음

	SSD(Solid-State Drive)		HDD(Hard disk Drive)	
	• Oracle			
API 이름	gp2	io1	st1	sc1
볼륨 크기	1GiB - 16TiB	4GiB - 16TiB	500GiB - 16TiB	500 GiB - 16 TiB
최대 IOPS**/볼륨	10,000개	20,000건	500	250
최대 처리량/볼륨†	160MiB/s	320MiB/s	500MiB/s	250MiB/s
최대 IOPS/인스턴스	65,000	65,000	65,000	65,000
최대 처리량/인스턴스	1,250MiB/s	1,250MiB/s	1,250MiB/s	1,250MiB/s
기준 성능 속성	IOPS	IOPS	MiB/s	MiB/s

*기본 볼륨 유형

16KiB I/O 크기 기준 **gp2/io1, 1MiB I/O 크기 기준 st1/sc1

† 이 처리량을 달성하려면 r3.8xlarge, x1.32xlarge와 같이 이 처리량을 지원하는 인스턴스가 있어야 합니다.

다음 표에서는 이전 세대 EBS 볼륨 유형을 설명합니다. 이전 세대 볼륨보다 우수한 성능 또는 성능 일관성이 필요한 경우, 범용 SSD(gp2) 또는 기타 현재 볼륨 유형 사용을 고려할 것을 권장합니다. 자세한 내용은 [이전 세대 볼륨](#)을 참조하십시오.

이전 세대 볼륨	
볼륨 유형	EBS Magnetic
설명	이전 세대 HDD
사용 사례	데이터에 자주 액세스하지 않는 워크로드
API 이름	standard
볼륨 크기	1GiB - 1TiB
최대 IOPS/볼륨	40~200
최대 처리량/볼륨	40-90MiB/s
최대 IOPS/인스턴스	48,000
최대 처리량/인스턴스	1,250MiB/s
기준 성능 속성	IOPS

Note

Linux AMI에서 부팅 볼륨 2TiB(2,048GiB) 이상을 사용하려면 GPT 파티션 테이블과 GRUB 2가 필요합니다. 현재 여러 Linux AMI에서 부팅 볼륨을 최대 2,047GiB까지만 지원하는 MBR 파티셔닝 체계를 사용하고 있습니다. 인스턴스가 2TiB 이상의 부팅 볼륨에서 부팅되지 않는 경우 사용 중인 AMI의 부팅 볼륨 크기가 2,047GiB로 제한된 상태일 수 있습니다. 부팅 볼륨이 아닌 볼륨에는 이 Linux 인스턴스에 대한 제한이 적용되지 않습니다.

인스턴스 구성, I/O 특성 및 워크로드 요구량 등 여러 가지 요인이 EBS 볼륨의 성능에 영향을 미칠 수 있습니다. EBS 볼륨을 최대한 이용하는 방법에 대한 자세한 내용은 [Linux 인스턴스의 Amazon EBS 볼륨 성능 \(p. 621\)](#)을 참조하십시오.

이러한 볼륨 유형의 자세한 가격 정보는 [Amazon EBS 가격 책정](#)을 참조하십시오.

범용 SSD(gp2) 볼륨

범용 SSD(gp2) 볼륨은 광범위한 작업에서 이상적으로 사용될 수 있는 비용 효과적인 스토리지를 제공합니다. 이러한 볼륨은 시간을 연장할 경우 3,000 IOPS의 버스트 기능까지 지원되어 지연 시간이 한 자릿수 밀리초에 불과합니다. 최소 100 IOPS(33.33GiB 이하)와 최대 10,000 IOPS(3,334GiB 이상) 사이에서, 기준 성능은 볼륨 크기의 GiB당 3 IOPS로 일정하게 확장됩니다. gp2 볼륨 크기는 1GiB~16TiB입니다.

I/O 크레딧 및 버스트 성능

gp2 볼륨의 성능은 볼륨 크기에 따라 정해지고 볼륨의 기준 성능 수준 및 I/O 크레딧이 얼마나 빨리 누적되는지를 결정합니다. 볼륨이 클수록 기준 성능 수준이 크고 I/O 크레딧이 빨리 누적됩니다. I/O 크레딧이란 기준 성능 이상이 필요한 경우 대규모 I/O를 버스트하도록 gp2 볼륨이 사용할 수 있는 가용 대역폭입니다. 볼륨에 I/O 크레딧이 많을수록 더 오랜 기간 동안 볼륨이 기준 성능 수준 이상을 버스트할 수 있고 더 큰 성능이 필요할 때 더 좋은 성능을 발휘할 수 있습니다. 다음 다이어그램은 gp2의 버스트 버킷 동작을 보여줍니다.

각 볼륨의 초기 I/O 크레딧 잔고는 540만 I/O 크레딧이고 이것은 30분 동안 3,000 IOPS의 최대 버스트 성능을 유지할 수 있는 수준입니다. 이러한 초기 크레딧 밸런스는 부트 볼륨에 빠른 초기 부팅 주기를 제공하고 기타 애플리케이션에 좋은 부트스트래핑 환경을 제공하도록 설계되었습니다. 볼륨은 볼륨 크기의 GiB 당 3 IOPS의 기준 성능 비율로 I/O 크레딧을 획득합니다. 예를 들어, 100 GiB gp2 볼륨은 300 IOPS의 기준 성능을 갖습니다.

볼륨에 기준 성능 I/O 수준 이상이 필요한 경우 크레딧 잔고에서 I/O 크레딧을 사용하여 최대 3,000 IOPS까지 필요한 성능 수준을 버스트할 수 있습니다. 1,000GiB 이상의 볼륨은 최대 버스트 성능 이상의 기준 성능을 갖고 I/O 크레딧 잔고는 차감되지 않습니다. 볼륨이 초당 획득한 I/O 크레딧 이하를 사용하는 경우 미사용 I/O 크레딧은 I/O 크레딧 잔고에 가산됩니다. 볼륨의 최대 I/O 크레딧은 초기 크레딧 잔고(540만 I/O 크레딧)와 동일합니다.

다음 표는 여러 볼륨 크기 및 관련 볼륨 기준 성능(I/O 크레딧 누적 비율), 최대 3,000 IOPS에서의 버스트 기간(전체 크레딧 잔고에서 시작 시) 및 볼륨이 빈 크레딧 잔고를 다시 채우는 데 걸리는 초 단위 시간을 보여줍니다.

볼륨 크기(GiB)	기준 성능(IOPS)	최대 버스트 기간 @ 3,000 IOPS(초)	빈 크레딧 잔고를 채우는데 소요되는 시간(초)
1	100	1862	54,000
100	300	2,000건	18,000
214(최대 처리량에 대한 최소 크기)	642	2,290	8,412
250	750	2,400	7,200
500	1,500	3,600	3,600
750	2,250	7,200	2,400
1,000	3,000	해당 사항 없음*	해당 사항 없음*
3,334(최대 IPOS에 대한 최소 크기)	10,000개	해당 사항 없음*	해당 사항 없음*

볼륨 크기(GiB)	기준 성능(IOPS)	최대 버스트 기간 @ 3,000 IOPS(초)	빈 크레딧 잔고를 채우는데 소요되는 시간(초)
16,384(16TiB, 최대 볼륨 크기)	10,000개	해당 사항 없음*	해당 사항 없음*

* 버스트와 I/O는 버스트 성능이 기준 성능을 초과하는 1,000GiB 이하의 볼륨만 관계가 있습니다.

볼륨의 버스트 구간은 볼륨의 크기, 필요한 버스트 IPOS 및 버스트가 시작되는 크레딧 잔고에 의해 결정됩니다. 방법은 다음 수식과 같습니다.

$$\text{Burst duration} = \frac{\text{(Credit balance)}}{(\text{Burst IOPS}) - 3(\text{Volume size in GiB})}$$

I/O 크레딧 잔고가 0이 되면 어떻게 되나요?

gp2 볼륨에서 I/O 크레딧 잔고 전부가 소진되면 볼륨의 최대 IPOS 성능이 기준 IPOS 성능 수준(볼륨이 크레딧을 획득하는 속도)으로 유지되고 볼륨의 최대 처리량은 최대 I/O 크기를 곱한 기준 IPOS로 줄어듭니다. 처리량은 160MiB/s를 초과할 수 없습니다. I/O 요구가 기준 성능 수준 이하로 떨어지고 미사용 크레딧이 I/O 크레딧 잔고에 추가되면, 볼륨의 최대 IPOS 성능이 다시 기준 수준을 초과합니다. 예를 들어, 크레딧 잔고가 0인 100GiB gp2 볼륨은 300 IPOS의 기준 성능 및 75MiB/s의 처리 한도를 갖습니다(초당 300I/O 작업 * I/O 작업당 256KiB = 75MiB/s). 볼륨이 커지면 기준 성능도 높아지고 크레딧 잔고가 더 빨리 보충됩니다. IPOS 측정 방법에 대한 자세한 내용은 [I/O 특성](#)을 참조하십시오.

볼륨 성능이 자주 기준 수준 한도에 도달하는 경우(빈 I/O 크레딧 잔고로 인해) 더 큰 gp2 볼륨(기준 성능 수준이 향상)을 사용하거나 IOPS 성능을 10,000 IOPS 이상으로 유지해야 하는 워크로드인 경우 io1 볼륨으로 전환할 것으로 고려해야 합니다.

CloudWatch 측정치 및 경보를 사용하여 버스트 버킷 잔고를 모니터링하는 방법은 [gp2, st1, sc1 볼륨에 대한 버스트 버킷 잔고 모니터링 \(p. 573\)](#) 섹션을 참조하십시오.

처리량 성능

gp2 볼륨의 처리량 제한은 170GiB 이하 볼륨의 경우 128MiB/s이고, 170GiB가 넘는 볼륨의 경우 160MiB/s입니다.

프로비저닝된 IOPS SSD(`io1`) 볼륨

프로비저닝된 IOPS SSD (`io1`) 볼륨은 스토리지 성능과 일관성에 민감한 I/O 집약적 워크로드, 특히 데이터 베이스 워크로드 요구 사항을 충족하도록 설계되었습니다. `io1` 볼륨에서는 버킷과 크레딧 모델을 사용해 성능을 계산하는 대신, 볼륨을 생성할 때 일정한 IOPS 속도를 지정할 수 있으며, Amazon EBS는 프로비저닝된 IOPS 성능의 10% 이내에서 임의의 1년 기간 중 99.9%까지 전송합니다.

`io1` 볼륨의 크기는 4GiB에서 16TiB 사이가 될 수 있고 볼륨당 100 ~ 최대 20,000 IOPS가 프로비저닝될 수 있습니다. 요청된 볼륨 크기(단위: GiB)에 대한 프로비저닝된 IOPS의 비율은 최대 50:1입니다. 예를 들어 100GiB 볼륨에서는 최대 5,000IOPS까지 프로비저닝할 수 있습니다. 크기가 400GiB 이상인 볼륨은 최대 20,000IOPS까지 프로비저닝할 수 있습니다.

`io1` 볼륨의 처리 한계는 프로비저닝된 각 IPOS에 대해 256KiB이고 최대 값은 320MiB/s(1,280 IOPS에서)입니다.

I/O당 지연 시간 환경은 프로비저닝된 IPOS 및 워크로드 패턴에 따라 다릅니다. 최상의 I/O당 지연 시간 환경을 위해서는 2:1 이상의 GiB 대 IOPS 비율로 프로비저닝하는 것이 좋습니다. 예를 들어 2,000IOPS 볼륨은 1,000GiB보다 작아야 합니다.

Note

2012년 이전에 만들어진 일부 AWS 계정은 프로비저닝된 IOPS SSD(`io1`) 볼륨을 지원하지 않는 `us-west-1` 또는 `ap-northeast-1`의 가용 영역에 대한 액세스 권한이 있을 수도 있습니다. 이런 리전 중 하나에 `io1` 볼륨을 만들거나 블록 디바이스 매핑에서 `io1` 볼륨이 있는 인스턴스를 시작할 수 없는 경우, 해당 리전에서 다른 가용 영역을 사용해 보십시오. 가용 영역에 4GiB의 `io1` 볼륨을 만들어 그 영역에서 `io1` 볼륨을 지원하는지 확인할 수 있습니다.

처리량에 최적화된 HDD(`st1`) 볼륨

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

Note

이 볼륨 유형은 대용량 순차 I/O와 관련된 워크로드에 최적화되어 있으며, 소량의 랜덤 I/O 워크로드 를 처리하는 고객에게는 `gp2` 사용을 권장합니다. 자세한 내용은 [HDD 기반 소량 읽기/쓰기의 비효율성 \(p. 573\)](#) 섹션을 참조하십시오.

처리량 크레딧 및 버스트 성능

`gp2`처럼 `st1` 역시 성능 측정에 버스트 버킷 모델을 사용합니다. 볼륨 크기에 따라 볼륨의 기준 처리량, 즉 볼륨이 처리량 크레딧을 누적하는 속도가 결정됩니다. 볼륨 크기는 볼륨의 버스트 처리량, 즉 사용 가능한 크레딧을 소비할 수 있는 속도도 결정합니다. 볼륨이 클수록 기본 및 버스트 처리량이 높습니다. 볼륨에 크레딧이 많을수록 버스트 수준에서 더 오랫동안 I/O를 구동할 수 있습니다.

다음 다이어그램은 `st1`의 버스트 버킷 동작을 보여줍니다.

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

$$(\text{Volume size}) \times (\text{Credit accumulation rate per TiB}) = \text{Throughput}$$

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

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

0.5TiB~16TiB 범위의 볼륨 크기를 기준으로 기준 처리량은 20MiB/s~500MiB/s(한도)이며, 12.5TiB에서 한도에 도달하는 이유는

$$\frac{40 \text{ MiB/s}}{12.5 \text{ TiB}} = 320 \text{ MiB/s}$$

버스트 처리량은 125MiB/s~500MiB/s(상한)이며, 2TiB에서 한도에 도달하는 이유는

$$\frac{250 \text{ MiB/s}}{2 \text{ TiB}} = 125 \text{ MiB/s}$$

다음 표는 `st1`의 기준 및 버스트 처리량 값 전체를 보여줍니다.

볼륨 크기(TiB)	ST1 기준 처리량(MiB/s)	ST1 버스트 처리량(MiB/s)
0.5	20	125
1	40	250

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

다음 다이어그램은 표의 값을 도식화한 것입니다.

Note

처리량에 최적화된 HDD (`st1`) 볼륨의 스냅샷을 생성하는 경우, 스냅샷이 진행되는 동안 성능이 볼륨의 기준 값까지 떨어질 수 있습니다.

CloudWatch 측정치 및 경보를 사용하여 버스트 버킷 잔고를 모니터링하는 방법은 [gp2, st1, sc1 볼륨에 대한 버스트 버킷 잔고 모니터링 \(p. 573\)](#) 섹션을 참조하십시오.

Cold HDD(`sc1`) 볼륨

Cold HDD(`sc1`) 볼륨은 IOPS가 아닌 처리량으로 성능을 정의하는 저비용 마그네트ic 스토리지를 제공합니다. 처리량 제한이 `st1`보다 낮은 `sc1`은 대용량 순차 콜드 데이터 워크로드에 적합합니다. 데이터에 자주 액세스 할 필요가 없고 비용을 절약해야 한다면 `sc1`이 저렴한 볼록 스토리지로 적합합니다. 부팅 가능한 `sc1` 볼륨은 지원되지 않습니다.

Note

이 볼륨 유형은 대용량 순차 I/O와 관련된 워크로드에 최적화되어 있으며, 소량의 랜덤 I/O 워크로드를 처리하는 고객에게는 `gp2` 사용을 권장합니다. 자세한 내용은 [HDD 기반 소량 읽기/쓰기의 비효율성 \(p. 573\)](#) 섹션을 참조하십시오.

처리량 크레딧 및 버스트 성능

`gp2`처럼 `sc1` 역시 성능 측정에 버스트 버킷 모델을 사용합니다. 볼륨 크기에 따라 볼륨의 기준 처리량, 즉 볼륨이 처리량 크레딧을 누적하는 속도가 결정됩니다. 볼륨 크기는 볼륨의 버스트 처리량, 즉 사용 가능한 크레

딜을 소비할 수 있는 속도도 결정합니다. 볼륨이 클수록 기본 및 버스트 처리량이 높습니다. 볼륨에 크레딧이 많을수록 버스트 수준에서 더 오랫동안 I/O를 구동할 수 있습니다.

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

$$(\text{Volume size}) \times (\text{Credit accumulation rate per TiB}) = \text{Throughput}$$

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

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

0.5TiB~16TiB 범위의 볼륨 크기를 기준으로 기준 처리량은 6MiB/s~192MiB/s(최대)이며, 16TiB에서 한도에 도달하는 이유는

$$12 \text{ MiB/s} \\ 16 \text{ TiB} \times \frac{12 \text{ MiB/s}}{1 \text{ TiB}} = 192 \text{ MiB/s}$$

버스트 처리량은 40MiB/s~250MiB/s(한도)이며, 3.125TiB에서 한도에 도달하는 이유는

$$80 \text{ MiB/s} \\ 3.125 \text{ TiB} \times \frac{80 \text{ MiB/s}}{1 \text{ TiB}} = 250 \text{ MiB/s}$$

다음 표는 sc1의 기준 및 버스트 처리량 값 전체를 보여줍니다.

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

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

다음 다이어그램은 표의 값을 도식화한 것입니다.

Note

Cold HDD (`sc1`) 볼륨의 스냅샷을 생성하는 경우, 스냅샷이 진행되는 동안 성능이 볼륨의 기준 값까지 떨어질 수 있습니다.

CloudWatch 측정치 및 경보를 사용하여 버스트 버킷 잔고를 모니터링하는 방법은 [gp2, st1, sc1 볼륨에 대한 버스트 버킷 잔고 모니터링 \(p. 573\)](#) 섹션을 참조하십시오.

Magnetic(standard)

Magnetic 볼륨은 마그네틱 드라이브로 구성되어 있으며, 데이터 액세스가 드문 워크로드, 작은 볼륨 크기에 맞는 저비용 스토리지가 중요한 시나리오에 적합합니다. Magnetic 볼륨의 평균 IOPS는 약 100 정도이며, 버스팅 시 몇백 수준으로 증가합니다. 크기는 1GiB에서 1TiB까지입니다.

Note

Magnetic은 이전 세대 볼륨입니다. 새로운 애플리케이션에는 새로운 볼륨 유형 중에서 선택해서 사용하는 것이 좋습니다. 자세한 내용은 [이전 세대 볼륨](#)을 참조하십시오.

CloudWatch 측정치 및 경보를 사용하여 버스트 버킷 잔고를 모니터링하는 방법은 [gp2, st1, sc1 볼륨에 대한 버스트 버킷 잔고 모니터링 \(p. 573\)](#) 섹션을 참조하십시오.

HDD 볼륨 사용 시 성능 고려사항

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

처리량에 최적화된 HDD 대 Cold HDD 비교

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

`st1` 및 `sc1`은 모두 99%의 기간 동안 90%의 버스트 처리량에 성능 일관성을 제공하도록 설계되었습니다. 매 시간 총 처리량 목표 99%를 달성하기 위해, 준수하지 않는 기간은 대략적으로 균등하게 분산됩니다.

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

일반적으로 스캔 시간은 이 수식으로 표현됩니다.

Volume size	= Scan time
-----	Throughput

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

Amazon Elastic Compute Cloud
Linux 인스턴스용 사용 설명서
EBS 볼륨

-----	-----	= 10,486 s = 2.91 hours (optimal)
500 MiB/s	0.00047684 TiB/s	

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

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

-----	-----	= 20972 s = 5.83 hours (optimal)
5 TiB	0.000238418 TiB/s	

-----	-----	= 5.83 hours
5.83 hours	+	= 6.54 hours (minimum expected)
(0.90)(0.99)		

볼륨 크기(TiB)	버스팅 시 ST1 스캔 시간(단위: 시간)*	버스팅 시 SC1 스캔 시간(단위: 시간)*
1	1.17	3.64
2	1.17	3.64
3	1.75	3.64
4	2.33	4.66
5	2.91	5.83
6	3.50	6.99
7	4.08	8.16
8	4.66	9.32
9	5.24	10.49
10	5.83	11.65
11	6.41	12.82
12	6.99	13.98
13	7.57	15.15
14	8.16	16.31
15	8.74	17.48
16	9.32	18.64

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

따라서 빠르게 스캔을 완료해야 하거나(최대 500MiB/s) 하루 안에 여러 건의 전체 볼륨 스캔이 필요한 처리량 중심의 워크로드를 가지고 있는 경우 st1을 사용하십시오. 비용을 최적화해야 하고 데이터에 그다지 자주 액세스하지 않으며 250MiB/s 이상의 스캔 성능이 필요하지 않다면 sc1을 사용하십시오.

HDD 기반 소량 읽기/쓰기의 비효율성

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

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

인스턴스당 처리량에 대한 제한

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

- 볼륨의 처리량 제한
- 인스턴스의 처리량 제한

모든 Amazon EBS 볼륨에서와 같이, 네트워크 병목 현상을 피하려면 적절한 EBS 최적화 EC2 인스턴스를 선택하는 것이 좋습니다. 자세한 내용은 [Amazon EBS 최적화 인스턴스](#)를 참조하십시오.

gp2, st1, sc1 볼륨에 대한 버스트 버킷 잡고 모니터링

gp2, st1, sc1 볼륨에 대해 Amazon CloudWatch에서 제공하는 EBS BurstBalance 측정치를 사용하여 버스트 버킷 수준을 모니터링할 수 있습니다. 이 측정치는 버스트 버킷에 남아 있는 I/O 크레딧(gp2의 경우)의 비율 또는 처리량 크레딧의 비율(st1 및 sc1의 경우)을 보여 줍니다. BurstBalance 측정치 및 I/O와 관련된 기타 측정치에 대한 자세한 내용은 [I/O 특성 및 모니터링](#)을 참조하십시오. CloudWatch에서는 BurstBalance 값이 특정 수준 밑으로 떨어질 경우 이를 알리도록 경보를 설정할 수도 있습니다. CloudWatch 경보 사용에 대한 자세한 내용은 [Amazon CloudWatch 경보 만들기](#)를 참조하십시오.

Amazon EBS 볼륨 생성

동일한 가용 영역 내의 모든 EC2 인스턴스에 연결할 수 있는 Amazon EBS 볼륨을 생성할 수 있습니다. 암호화된 EBS 볼륨을 생성하도록 선택할 수 있지만 암호화된 볼륨은 일부 인스턴스 유형에만 연결될 수 있습니다. 자세한 내용은 [지원되는 인스턴스 유형 \(p. 619\)](#) 섹션을 참조하십시오. IAM 정책을 사용하여 새 볼륨에서 암호화를 적용할 수 있습니다. 자세한 내용은 [4: 볼륨 작업 \(p. 434\)](#) 및 [5: 인스턴스 시작 \(RunInstances\) \(p. 436\)](#)의 예제 IAM 정책을 참조하십시오.

또한 인스턴스 시작 시 블록 디바이스 매핑을 지정하여 EBS 볼륨을 생성하고 연결할 수 있습니다. 자세한 내용은 [인스턴스 시작하기 \(p. 265\)](#) 및 [블록 디바이스 매핑 \(p. 662\)](#) 섹션을 참조하십시오. 이전에 생성된 스냅샷에서 볼륨을 복구할 수 있습니다. 자세한 내용은 [스냅샷에서 Amazon EBS 볼륨 복구 \(p. 574\)](#) 섹션을 참조하십시오.

생성 시 EBS 볼륨에 태그를 적용할 수 있습니다. 태그 지정을 통해 Amazon EC2 리소스 인벤토리 추적을 단순화할 수 있습니다. 생성 시 태그 지정을 IAM 정책과 조합하여 새 볼륨에서 태그 지정을 적용할 수 있습니다. 자세한 내용은 [리소스 태그 지정](#) 단원을 참조하십시오.

고성능 스토리지 시나리오용으로 볼륨을 생성하는 경우 프로비저닝된 IOPS SSD(i_o1) 볼륨을 생성한 다음 EBS에 최적화된 인스턴스 또는 10Gb 네트워크에 연결된 인스턴스 등 애플리케이션을 지원하기에 충분한 대역폭이 있는 인스턴스에 연결합니다. 처리량에 최적화된 HDD(st1) 및 Cold HDD(sc1) 볼륨에도 같은 원칙이 적용됩니다. 자세한 내용은 [Amazon EC2 인스턴스 구성 \(p. 623\)](#) 섹션을 참조하십시오.

새 EBS 볼륨은 사용 가능하지만 초기화(이전에는 사전 워밍이라고 함)가 필요하지 않은 시점에 최고 성능을 발휘합니다. 하지만 스냅샷에서 복원된 볼륨의 스토리지 블록은 초기화(Amazon S3에서 가져와 볼륨에 기록) 후에만 액세스할 수 있습니다. 이 예비 작업은 시간이 걸리며, 각 블록을 처음 액세스할 때 I/O 작업의 지연 시간을 상당히 증가시킬 수 있습니다. 대부분 애플리케이션의 경우 볼륨 수명 주기 동안 이 비용을 분할 상환할 수 있습니다. 데이터에 한 번 액세스한 후에는 성능이 복원됩니다. 자세한 내용은 [Amazon EBS 볼륨 초기화 \(p. 628\)](#) 섹션을 참조하십시오.

콘솔을 사용하여 EBS 볼륨을 생성하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.

2. 탐색 모음에서 볼륨을 생성할 리전을 선택합니다. 일부 Amazon EC2 리소스는 리전 간에 공유될 수 있지만 그렇지 않은 리소스도 있으므로 잘 선택해야 합니다. 자세한 내용은 [리소스 위치 \(p. 673\)](#) 섹션을 참조하십시오.
3. 탐색 창의 [ELASTIC BLOCK STORE] 아래에서 [Volumes]를 선택합니다.
4. 위쪽 창에서 [Create Volume]을 선택합니다.
5. Create Volume 대화상자의 Volume Type에서 범용 SSD (GP2), 프로비저닝된 IOPS SSD (IO1), 처리량에 최적화된 HDD (ST1), Cold HDD (SC1) 또는 Magnetic을 선택합니다. 자세한 내용은 [Amazon EBS 볼륨 유형 \(p. 564\)](#) 섹션을 참조하십시오.

Note

2012년 이전에 만들어진 일부 AWS 계정은 프로비저닝된 IOPS SSD(io1) 볼륨을 지원하지 않는 us-west-1 또는 ap-northeast-1의 가용 영역에 대한 액세스 권한이 있을 수도 있습니다. 이런 리전 중 하나에 io1 볼륨을 만들거나 블록 디바이스 매핑에서 io1 볼륨이 있는 인스턴스를 시작할 수 없는 경우, 해당 리전에서 다른 가용 영역을 사용해 보십시오. 가용 영역에 4GiB의 io1 볼륨을 만들어 그 영역에서 io1 볼륨을 지원하는지 확인할 수 있습니다.

6. [Size]에 볼륨 크기를 GiB 단위로 입력합니다.
7. io1 볼륨의 경우 [IOPS] 필드에서 볼륨이 지원하는 최대 초당 입출력 작업수를 입력합니다.
8. [Availability Zone]에서 볼륨을 생성할 가용 영역을 선택합니다.
9. (선택 사항) 암호화된 볼륨을 생성하려면 [Encrypted] 상자를 선택한 다음 볼륨 암호화 시 사용할 마스터 키를 선택합니다. 계정의 기본 마스터 키를 선택하거나 AWS Key Management Service을 사용하여 이전에 생성한 고객 마스터 키(CMK)를 선택할 수 있습니다. 사용 가능한 키는 [Master Key] 메뉴에 표시됩니다. 아니면 액세스한 모든 키의 전체 ARN을 붙여 넣을 수 있습니다. 자세한 내용은 [AWS Key Management Service Developer Guide](#) 섹션을 참조하십시오.

Note

암호화된 볼륨은 일부 인스턴스 유형에만 연결될 수 있습니다. 자세한 내용은 [지원되는 인스턴스 유형 \(p. 619\)](#) 섹션을 참조하십시오.

10. [Yes, Create]를 선택합니다.

명령줄을 사용하여 EBS 볼륨을 생성하려면

다음 명령 중 하나를 사용할 수 있습니다. 이러한 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [create-volume \(AWS CLI\)](#)
- [New-EC2Volume \(Windows PowerShell용 AWS 도구\)](#)

스냅샷에서 Amazon EBS 볼륨 복구

Amazon S3에 저장된 스냅샷의 데이터로 Amazon EBS 볼륨을 복구할 수 있습니다. 사용자는 볼륨 복구에 사용할 스냅샷의 ID를 알아야 하고 해당 스냅샷에 대한 액세스 권한이 있어야 합니다. 스냅샷에 대한 자세한 내용은 [Amazon EBS 스냅샷 \(p. 607\)](#) 섹션을 참조하십시오.

기존 EBS 스냅샷을 이용해 생성한 새 볼륨은 백그라운드에 느리게 로드됩니다. 이는 스냅샷에서 볼륨을 생성한 후 Amazon S3에서 EBS 볼륨으로 모든 데이터가 전송될 때까지 기다리지 않아도 연결된 인스턴스에서 볼륨과 모든 데이터에 액세스할 수 있음을 의미합니다. 인스턴스가 아직 로드되지 않은 데이터에 액세스하는 경우, 볼륨은 요청한 데이터를 Amazon S3에서 즉시 다운로드한 후, 백그라운드에서 볼륨의 나머지 데이터 로드를 진행합니다.

암호화된 스냅샷에서 복구된 EBS 볼륨은 자동으로 암호화됩니다. 암호화된 볼륨은 일부 인스턴스 유형에만 연결될 수 있습니다. 자세한 내용은 [지원되는 인스턴스 유형 \(p. 619\)](#) 섹션을 참조하십시오.

보안 제약 조건 때문에, 자신이 소유하지 않은 암호화된 공유 스냅샷에서 EBS 볼륨을 직접 복원할 수는 없습니다. 먼저 자신이 소유할 스냅샷의 복사본을 만들어야 합니다. 그러면 그 복사본으로부터 볼륨을 복원할 수 있습니다. 자세한 내용은 [Amazon EBS 암호화](#)를 참조하십시오.

새 EBS 볼륨은 사용 가능하지만 초기화(이전에는 사전 워밍이라고 함)가 필요하지 않은 시점에 최고 성능을 발휘합니다. 하지만 스냅샷에서 복원된 볼륨의 스토리지 블록은 초기화(Amazon S3에서 가져와 볼륨에 기록) 후에만 액세스할 수 있습니다. 이 예비 작업은 시간이 걸리며, 각 블록을 처음 액세스할 때 I/O 작업의 지연 시간을 상당히 증가시킬 수 있습니다. 데이터에 한 번 액세스한 후에는 성능이 복원됩니다.

대부분의 애플리케이션은 볼륨 수명 주기 동안 초기화 비용을 분할 상환할 수 있습니다. 복구된 볼륨이 항상 프로덕션의 피크 용량에서 작동하도록 해야 할 경우, dd 또는 fio를 사용해 전체 볼륨에 대한 즉각적인 초기화를 강제로 실행할 수 있습니다. 자세한 내용은 [Amazon EBS 볼륨 초기화 \(p. 628\)](#) 섹션을 참조하십시오.

콘솔을 사용하여 스냅샷에서 EBS 볼륨을 복원하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 모음에서 스냅샷이 있는 리전을 선택합니다. 일부 Amazon EC2 리소스는 리전 간에 공유될 수 있지만 그렇지 않은 리소스도 있으므로 잘 선택해야 합니다. 자세한 내용은 [리소스 위치 \(p. 673\)](#) 섹션을 참조하십시오. 다른 리전에 있는 볼륨으로 스냅샷으로 복구해야 하는 경우 새 리전으로 스냅샷을 복구한 다음 해당 리전에서 볼륨을 복구하면 됩니다. 자세한 내용은 [Amazon EBS 스냅샷 복사 \(p. 610\)](#) 섹션을 참조하십시오.
3. 탐색 창에서 [Volumes], [Create Volume]을 차례로 선택합니다.
4. [Create Volume] 대화상자의 [Volume Type]에서 범용 SSD, 프로비저닝된 IOPS SSD 또는 Magnetic을 선택합니다. 자세한 내용은 [Amazon EBS 볼륨 유형 \(p. 564\)](#) 섹션을 참조하십시오.

Note

2012년 이전에 만들어진 일부 AWS 계정은 프로비저닝된 IOPS SSD(`io1`) 볼륨을 지원하지 않는 `us-west-1` 또는 `ap-northeast-1`의 가용 영역에 대한 액세스 권한이 있을 수도 있습니다. 이런 리전 중 하나에 `io1` 볼륨을 만들거나 블록 디바이스 매핑에서 `io1` 볼륨이 있는 인스턴스를 시작할 수 없는 경우, 해당 리전에서 다른 가용 영역을 사용해 보십시오. 가용 영역에 4GiB의 `io1` 볼륨을 만들어 그 영역에서 `io1` 볼륨을 지원하는지 확인할 수 있습니다.

5. [Snapshot]에서 볼륨을 복원할 스냅샷의 ID 또는 설명을 입력한 다음 제안 옵션 목록에서 선택합니다.

Note

암호화된 스냅샷에서 복구된 볼륨은 Amazon EBS 암호화를 지원하는 인스턴스에만 연결될 수 있습니다. 자세한 내용은 [지원되는 인스턴스 유형 \(p. 619\)](#) 섹션을 참조하십시오.

6. [Size]에서 볼륨을 GiB 단위로 입력하거나 스냅샷의 기본 크기가 적절한지 확인합니다.

볼륨 크기와 스냅샷 ID를 모두 지정한 경우 크기는 스냅샷 크기 이상이어야 합니다. 볼륨 유형과 스냅샷 ID를 선택하면 볼륨의 최소 및 최대 크기를 [Size] 목록 옆에서 확인할 수 있습니다. 스냅샷의 AWS Marketplace 제품 코드는 전부 해당 볼륨으로 전파됩니다.

7. `io1` 볼륨의 경우 [IOPS] 필드에 볼륨이 지원하는 최대 초당 입출력 작업 수를 입력합니다.
8. [Availability Zone] 목록에서 볼륨을 생성할 가용 영역을 선택합니다. EBS 볼륨은 동일한 가용 영역 내의 EC2 인스턴스에만 연결될 수 있습니다.
9. [Yes, Create]를 선택합니다.

Important

스냅샷이 기본 볼륨보다 큰 볼륨으로 스냅샷으로 복구된 경우 볼륨의 파일 시스템을 확장하여 추가 공간을 활용할 수 있어야 합니다. 자세한 내용은 [Linux에서 EBS 볼륨의 크기, IOPS 또는 유형 수정 \(p. 590\)](#) 섹션을 참조하십시오.

스냅샷에서 볼륨을 복원한 후에는 이 볼륨을 인스턴스에 연결하여 사용할 수 있습니다. 자세한 내용은 [Amazon EBS 볼륨을 인스턴스에 연결 \(p. 576\)](#) 섹션을 참조하십시오.

명령줄을 사용하여 EBS 볼륨을 복원하려면

다음 명령 중 하나를 사용할 수 있습니다. 이러한 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [create-volume](#) (AWS CLI)
- [New-EC2Volume](#) (Windows PowerShell용 AWS 도구)

Amazon EBS 볼륨을 인스턴스에 연결

EBS 볼륨을 해당 볼륨과 동일한 가용 영역에 있는 인스턴스 중 하나에 연결할 수 있습니다.

사전 조건

- 사용할 디바이스 이름을 확인합니다. 자세한 내용은 [Linux 인스턴스의 디바이스 명명 \(p. 660\)](#) 섹션을 참조하십시오.
- 인스턴스에 연결할 수 있는 볼륨 수를 확인합니다. 자세한 내용은 [인스턴스 볼륨 제한 \(p. 659\)](#) 섹션을 참조하십시오.
- 볼륨이 암호화된 경우에는 Amazon EBS 암호화를 지원하는 인스턴스에만 연결할 수 있습니다. 자세한 내용은 [지원되는 인스턴스 유형 \(p. 619\)](#) 섹션을 참조하십시오.
- 볼륨에 AWS 마켓플레이스 제품 코드가 있는 경우:
 - 해당 볼륨은 중지된 인스턴스에만 연결될 수 있습니다.
 - 이 경우 볼륨에 있는 AWS 마켓플레이스 코드를 수신해야 합니다.
 - 인스턴스 구성(인스턴스 유형, 운영 체제)에서 해당 AWS 마켓플레이스 코드를 지원해야 합니다. 예를 들어, Windows 인스턴스의 볼륨을 Linux 인스턴스로 연결할 수 없습니다.
- AWS 제품 코드는 볼륨에서 인스턴스로 복사됩니다.

콘솔을 사용하여 EBS 볼륨을 인스턴스에 연결하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Volumes]를 선택합니다.
3. 볼륨을 선택한 후 [Actions], [Attach Volume]을 선택합니다.
4. [Attach Volume] 대화 상자의 [Instance]에 볼륨을 연결할 인스턴스의 ID 또는 이름을 입력하고 제안 옵션 목록에서 선택합니다(볼륨과 동일한 가용 영역에 있는 인스턴스만 표시됨).
5. 제안된 디바이스 이름을 사용하거나 지원되는 다른 디바이스 이름을 입력할 수 있습니다.

Important

인스턴스의 블록 디바이스 드라이버는 볼륨이 마운트될 때 실제 볼륨 이름을 할당하고 할당된 이름은 Amazon EC2 권장 이름과 다를 수 있습니다.

6. [Attach]를 선택합니다.
7. 인스턴스에 연결하고 볼륨을 사용 가능하도록 만듭니다. 자세한 내용은 [Amazon EBS 볼륨을 사용할 수 있도록 만들기 \(p. 577\)](#) 섹션을 참조하십시오.

명령줄을 사용하여 EBS 볼륨을 인스턴스에 연결하려면

다음 명령 중 하나를 사용할 수 있습니다. 이러한 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [attach-volume](#) (AWS CLI)
- [New-EC2Volume](#) (Windows PowerShell용 AWS 도구)

Amazon EBS 볼륨을 사용할 수 있도록 만들기

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

다른 볼륨을 생성할 때 기준으로 사용하거나 백업을 목적으로 EBS 볼륨의 스냅샷을 생성할 수 있습니다. 자세한 내용은 [Amazon EBS 스냅샷 \(p. 607\)](#) 섹션을 참조하십시오.

Linux에서 볼륨을 사용할 수 있도록 만들기

다음 절차를 통해 볼륨을 사용 가능하도록 만들 수 있습니다. Windows 인스턴스에서의 볼륨에 대한 지침은 [Windows 인스턴스용 Amazon EC2 사용 설명서의 Windows에서 볼륨을 사용 가능하게 만들기](#) 섹션을 참조하십시오.

Linux에서 EBS 볼륨을 사용 가능하게 만들려면

1. SSH를 사용하여 인스턴스에 연결합니다. 자세한 내용은 [2단계: 인스턴스에 연결 \(p. 23\)](#) 섹션을 참조하십시오.
2. 커널의 블록 디바이스 드라이버에 따라, 디바이스가 사용자가 지정한 것과는 다른 이름으로 연결될 수 있습니다. 예를 들어 `/dev/sah`라는 디바이스 이름을 지정할 경우 디바이스 이름이 커널에 의해 `/dev/xvdh` 또는 `/dev/hdh`로 바뀔 수 있습니다. 대부분의 경우, 뒤에 오는 문자는 동일하게 유지됩니다. Red Hat Enterprise Linux의 일부 버전과 CentOS와 같은 Red Hat Enterprise Linux의 변형 버전에서는 뒤에 오는 문자가 변경될 수도 있습니다(즉, `/dev/sda`가 `/dev/xvde`로 바뀔 수 있음). 이런 경우, 각 디바이스 이름에서 뒤에 오는 문자는 같은 수로 들어납니다. 예를 들어 `/dev/sdb`는 `/dev/xvdf`가 되고 `/dev/sdc`는 `/dev/xvdg`가 되는 식입니다. Amazon Linux AMI는 시작할 때 지정한 이름으로 심볼 링크를 만들어 이름이 바뀐 디바이스 경로를 가리키지만, 다른 AMI는 다르게 동작할 수도 있습니다.

`lsblk` 명령을 사용하면 사용 가능한 디스크 디바이스 및 마운트 포인트(해당하는 경우)가 표시되어 사용 가능한 올바른 디바이스 이름을 결정하는 데 도움을 받을 수 있습니다.

```
[ec2-user ~]$ lsblk
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvdf  202:80   0 100G  0 disk
xvda1 202:1    0   8G  0 disk /
```

`lsblk` 명령의 출력에서는 전체 디바이스 경로 중 맨 앞에 `/dev/`가 생략됩니다. 이 예제에서 `/dev/xvda1`는 루트 디바이스(MOUNTPOINT는 Linux 파일 시스템 구조의 루트에 해당하는 `/`로 표시됨)로 마운트되고 `/dev/xvdf`는 연결되었지만 아직 마운트된 상태는 아닙니다.

3. 볼륨에서 파일 시스템을 생성해야 하는지의 여부를 결정합니다. 새 볼륨은 원시 블록 디바이스이므로 마운트하고 사용하기 전 해당 볼륨에서 파일 시스템을 생성해야 합니다. 스냅샷에서 복구된 볼륨에는 이미 파일 시스템이 있을 수 있습니다. 기존 파일 시스템 위에 새 파일 시스템을 생성하면 해당 동작으로 데이터가 덮어쓰기됩니다. `sudo file -s device` 명령을 사용하면 파일 시스템 유형 등의 특수 정보를 확인할 수 있습니다.

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

이전 명령의 결과 디바이스에 대한 `data`만 표시되면 해당 디바이스에 파일 시스템이 없는 것으로 파일 시스템을 생성해야 합니다. 그 경우 [Step 4 \(p. 578\)](#)(를) 진행할 수 있습니다. 파일 시스템이 있는 디바이스에서 이 명령을 실행하면 결과가 다르게 표시됩니다.

```
[ec2-user ~]$ sudo file -s /dev/xvda1
/dev/xvda1: Linux rev 1.0 ext4 filesystem data, UUID=1701d228-e1bd-4094-a14c-8c64d6819362 (needs journal recovery) (extents) (large files) (huge files)
```

이전 예제에서 디바이스에는 Linux rev 1.0 ext4 filesystem data이(가) 있으므로 이 볼륨에서는 파일 시스템을 생성할 필요가 없습니다(출력에서 파일 시스템 데이터가 표시되면 Step 4 (p. 578)(으)로 건너뛸 수 있습니다.)

- (조건부) 다음 명령을 사용하면 볼륨에 ext4 파일 시스템을 생성할 수 있습니다. 디바이스 이름을 바꿔 넣습니다(`device_name`에 `/dev/xvdf` 등과 같이). 애플리케이션 요구 사항 또는 운영 체제 제한에 따라 ext3 또는 XFS 등 다른 파일 시스템 유형을 선택할 수 있습니다.

Warning

이 단계에서는 비어 있는 볼륨이 마운트된 경우를 가정합니다. 이미 데이터가 있는 볼륨이 마운트된 경우(예를 들어, 스냅샷에서 복구된 볼륨) 볼륨을 마운트하기 전 mkfs 명령을 사용하지 마십시오(대신 다음 단계로 건너뛰기). 아니면 볼륨을 포맷하여 기존 데이터를 삭제합니다.

```
[ec2-user ~]$ sudo mkfs -t ext4 device_name
```

- 다음 명령을 사용하여 볼륨에서 사용할 마운트 포인트 디렉터리를 생성합니다. 마운트 포인트는 파일 시스템 트리에 볼륨이 위치하고 볼륨을 마운트한 후 파일을 읽고 쓰는 위치입니다. `mount_point`의 위치를 `/data`와 같이 변경합니다.

```
[ec2-user ~]$ sudo mkdir mount_point
```

- 다음 명령을 사용하여 방금 생성한 위치에 볼륨을 마운트합니다.

```
[ec2-user ~]$ sudo mount device_name mount_point
```

- (선택 사항) 시스템을 재부팅할 때마다 이 EBS 볼륨을 마운트하려면 디바이스에 대한 항목을 `/etc/fstab` 파일에 추가합니다.

- 수정 도중 실수로 이 파일이 손상되거나 삭제되는 경우에 대비하여 `/etc/fstab` 파일의 백업을 생성합니다.

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

- nano 또는 vim과 같이 텍스트 편집기를 사용하여 `/etc/fstab` 파일을 엽니다.

Note

파일을 root로 열거나 sudo 명령을 사용하여 열어야 합니다.

- 해당 볼륨의 파일 끝에 다음 형식으로 새 줄을 추가합니다.

```
device_name mount_point file_system_type fs_mntops fs_freq fs_passno
```

이 줄의 마지막 세 필드는 파일 시스템 마운트 옵션, 파일 시스템의 덤프 빈도 및 부팅 시 파일 시스템 확인 순서입니다. 어떤 값을 입력해야 하는지 모르는 경우 다음 예제에 제공된 값을 사용하십시오(`defaults, nofail 0 2`). `/etc/fstab` 항목에 대한 자세한 내용은 fstab 매뉴얼 페이지(명령줄에 `man fstab` 입력)를 참조하십시오.

시스템의 현재 디바이스 이름(`/dev/sda1`, `/dev/xvda1` 등)을 `/etc/fstab`에 사용할 수 있지만, 디바이스의 128비트 UUID(Universally Unique Identifier)를 사용할 것을 권장합니다. 시스템 선언 블록 디바이스 이름은 상황에 따라 변경될 수 있지만, UUID는 볼륨 파티션에 할당되어, 포맷되어 파티션 수명 동안 영구적으로 유지됩니다. UUID를 사용하면 하드웨어 재구성 후 `/etc/fstab`의 블록 디바이스 매핑으로 인해 시스템을 부팅할 수 있게 되는 경우가 줄어듭니다.

디바이스의 UUID를 확인하려면 먼저 사용 가능한 디바이스를 조회합니다.

```
[ec2-user ~]$ df
```

이렇게 하면 다음과 같은 목록이 생성됩니다.

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/xvda1	8123812	1876888	6146676	24%	/
devtmpfs	500712	56	500656	1%	/dev
tmpfs	509724	0	509724	0%	/dev/shm

그런 다음 이 예제를 계속하여 두 명령 중 하나의 출력을 통해 /dev/xvda1의 UUID를 확인합니다.

- **sudo file -s /dev/xvda1**
- **ls -al /dev/disk/by-uuid/**

예를 들어, /dev/xvda1에서 UUID de9a1cccd-a2dd-44f1-8be8-0123456abcdef를 확인한 경우, /etc/fstab에 다음을 추가하여 마운트 지점 /data에 ext4 파일 시스템을 마운트합니다.

UUID=de9a1cccd-a2dd-44f1-8be8-0123456abcdef	/data	ext4	defaults,nofail
0	2		

Note

이 볼륨을 연결하지 않고 인스턴스를 부팅하려면(예: 이 볼륨이 서로 다른 인스턴스 사이를 이동할 수 있도록) 볼륨 마운트 시 오류가 있어도 인스턴스가 부팅되도록 하는 nofail 마운트 옵션을 추가해야 합니다. 16.04 이전의 Ubuntu 버전을 포함하는 Debian 계열 시스템에서는 nobootwait 탑재 옵션도 추가해야 합니다.

- /etc/fstab에 새 항목을 추가한 다음에는 해당 항목이 작동하는지 확인해야 합니다. sudo mount -a 명령을 실행하여 /etc/fstab에서 모든 파일 시스템을 마운트합니다.

[ec2-user ~]\$ sudo mount -a

이전 명령에서 오류가 발생하지 않으면 /etc/fstab 파일이 정상이고 다음 부팅 시 파일 시스템이 자동으로 마운트됩니다. 명령에서 오류가 발생하면 오류를 검토한 다음 /etc/fstab를 수정합니다.

Warning

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

- (선택 사항) /etc/fstab 오류 수정 방법을 모르는 경우 다음 명령으로 항상 백업 /etc/fstab 파일을 복원할 수 있습니다.

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

- 새 볼륨 마운트의 파일 권한을 검토하여 사용자 및 애플리케이션이 볼륨에 기록할 수 있는지 확인합니다. 파일 권한에 대한 자세한 내용은 Linux Documentation Project에서 [File security](#) 섹션을 참조하십시오.

볼륨 정보 보기

AWS Management Console에서 선택한 리전의 Amazon EBS 볼륨에 대한 설명 정보를 확인할 수 있습니다. 또한, 크기, 볼륨, 볼륨 암호화 여부, 볼륨 암호화에 마스터 키 사용 여부 및 볼륨이 연결된 특정 인스턴스 등 단일 볼륨에 대한 자세한 설명도 제공됩니다.

콘솔을 사용하여 EBS 볼륨에 대한 정보 확인

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Volumes]를 선택합니다.
3. 볼륨에 대한 자세한 정보를 확인하려면 선택합니다. 세부 정보 창에서 볼륨에 대해 제공된 정보를 검사할 수 있습니다.

어떤 EBS(또는 다른) 볼륨이 Amazon EC2 인스턴스에 연결되는지 알아보기

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. 인스턴스에 대한 자세한 정보를 확인하려면 선택합니다.
4. 세부 정보 창에서 루트 및 블록 디바이스에 대해 제공된 정보를 검사할 수 있습니다.

명령줄을 사용하여 EBS 볼륨에 대한 정보를 확인하려면

다음 명령 중 하나를 사용하여 볼륨 속성을 볼 수 있습니다. 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [describe-volumes\(AWS CLI\)](#)
- [Get-EC2Volume \(Windows PowerShell용 AWS 도구\)](#)

볼륨 상태 모니터링

Amazon Web Services(AWS)에서는 Amazon Elastic Block Store(Amazon EBS) 볼륨을 모니터링하는 데 사용할 수 있는 데이터(예: Amazon CloudWatch 측정치 및 볼륨 상태 검사)를 자동으로 제공합니다.

목차

- [CloudWatch로 볼륨 모니터링 \(p. 580\)](#)
- [상태 확인으로 볼륨 모니터링 \(p. 583\)](#)
- [볼륨 이벤트 모니터링 \(p. 584\)](#)
- [손상된 볼륨 작업 \(p. 585\)](#)
- [AutoEnableIO 볼륨 속성 작업 \(p. 587\)](#)

CloudWatch로 볼륨 모니터링

CloudWatch 측정치는 볼륨의 작동 동작을 살펴보고, 분석하고, 경보를 설정하는 데 사용할 수 있는 통계 데이터입니다.

다음 표에서는 Amazon EBS 볼륨에 사용할 수 있는 모니터링 데이터 유형을 설명합니다.

유형	설명
기본	자동으로 5분 기간 동안 데이터를 무료로 사용할 수 있습니다. 여기에는 EBS 기반 인스턴스의 루트 디바이스 볼륨에 대한 데이터가 포함됩니다.
세부	프로비저닝된 IOPS SSD(<code>io1</code>) 볼륨이 1분 지표를 CloudWatch에 자동으로 보냅니다.

CloudWatch에서 데이터를 가져올 때 반환되는 데이터의 세부 수준을 지정하는 `Period` 요청 파라미터를 포함할 수 있습니다. 이 파라미터는 데이터를 수집할 때 사용하는 기간(5분 기간)과 같습니다. 반환되는 데이터가 유효하도록 요청의 기간을 수집 기간보다 길거나 같게 지정하는 것이 좋습니다.

CloudWatch API 또는 Amazon EC2 콘솔을 사용하여 데이터를 가져올 수 있습니다. 이 콘솔은 CloudWatch API에서 원시 데이터를 가져오고 데이터를 기반으로 일련의 그래프를 표시합니다. 필요에 따라 API의 데이터나 콘솔의 그래프를 사용할 수 있습니다.

Amazon EBS 지표

Amazon Elastic Block Store(Amazon EBS)에서는 여러 메트릭에 대한 데이터 요소를 CloudWatch에 전송합니다. Amazon EBS 범용 SSD(gp2), 처리량에 최적화된 HDD(st1), 콜드 HDD(sc1) 및 Magnetic(표준) 볼륨은 자동으로 5개 측정치를 CloudWatch로 전송합니다. 프로비저닝 IOPS SSD(io1) 볼륨은 1분 측정치를 CloudWatch에 자동으로 전송합니다. Amazon EBS를 모니터링하는 방법에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [볼륨 상태 모니터링](#)을 참조하십시오.

AWS/EBS 네임스페이스에는 다음 지표가 포함되어 있습니다.

지표	설명
VolumeReadBytes	지정된 기간의 I/O 작업에 대한 정보를 제공합니다. <code>sum</code> 통계는 해당 기간 동안 전송된 총 바이트 수를 보고합니다. <code>Average</code> 통계는 기간 동안 각 I/O 작업의 평균 크기를 보고합니다. <code>SampleCount</code> 통계는 해당 기간 동안 총 I/O 작업 수를 보고합니다. <code>Minimum</code> 및 <code>Maximum</code> 통계는 이 측정치와 관계가 없습니다. 볼륨이 활성 상태일 때만 Amazon CloudWatch에 데이터가 보고되고, 볼륨이 유휴 상태일 때는 Amazon CloudWatch에 데이터가 보고되지 않습니다.
VolumeWriteBytes	단위: 바이트
VolumeReadOps	지정된 기간의 총 I/O 작업 수입니다.
VolumeWriteOps	Note 기간의 평균 IOPS(초당 I/O 작업 수)를 계산하려면 기간의 총 작업 수를 해당 기간의 초 수로 나누십시오. 단위: 수
VolumeTotalReadTime	지정된 기간 동안 완료된 모든 작업에서 사용한 총 시간(초)입니다. 여러 요청이 동시에 제출된 경우 이 총계가 기간 길이보다 클 수 있습니다. 예를 들어, 5분(300초) 동안 700개의 작업이 완료되고 작업당 1초가 걸린 경우 값은 700초입니다.
VolumeTotalWriteTime	단위: 초
VolumeIdleTime	지정된 기간 동안 읽기 또는 쓰기 작업이 제출되지 않은 총 시간(초)입니다. 단위: 초
VolumeQueueLength	지정된 기간 동안 완료 대기 중인 읽기 및 쓰기 작업 요청 수입니다. 단위: 수
VolumeThroughputPercent	프로비저닝된 IOPS SSD 볼륨에만 사용됩니다. Amazon EBS 볼륨에 대해 프로비저닝된 총 IOPS(초당 I/O 작업 수) 중 전송된 IOPS의 비율(%)입니다. 프로비저닝된 IOPS SSD 볼륨은 프로비저닝된 IOPS의 10% 내를 전송하여 지정된 연도의 시간 중 99.9%의 성능을 보입니다. Note 쓰기 중 1분 동안 보류 중인 다른 I/O 요청이 없으면 측정치 값이 100%가 됩니다. 또한 피크 사용 중 볼륨 스냅샷 만들기, EBS 최적화 인스턴스가 아닌 인스턴스에서 볼륨 실행, 볼륨 데이터에 처음 액세스 등의 사용자 작업으로 인해 볼륨의 I/O 성능이 일시적으로 저하될 수 있습니다.

지표	설명
VolumeConsumedReadWriteOps	단위: 백분율 포로 비저닝된 IOPS SSD 볼륨에만 사용됩니다. 지정된 시간 동안 소비한 총 읽기 및 쓰기 작업량(256,000 용량 단위로 정규화됨)입니다. 256,000보다 작은 I/O 작업은 각각 1개의 소비 IOPS로 계산되고, 256,000보다 큰 I/O 작업은 256,000 용량 단위로 계산됩니다. 예를 들어, 1,024,000 I/O는 소비 IOPS 4개로 계산됩니다.
BurstBalance	단위: 수 범용 SSD(gp2), 처리량에 최적화된 HDD(st1), Cold HDD(sc1) 볼륨에만 사용됩니다. 버스트 버킷에 남아 있는 I/O 크레딧(gp2의 경우)의 비율 또는 처리량 크레딧의 비율(st1 및 sc1의 경우) 정보를 보여 줍니다. 볼륨이 활성 상태일 때만 CloudWatch에 데이터가 보고되고, 볼륨이 연결되지 않은 경우에는 데이터가 보고되지 않습니다.
	단위: 백분율

Amazon EBS 메트릭의 차원

Amazon EBS에서 CloudWatch에 전송하는 유일한 차원은 볼륨 ID입니다. 즉, 사용 가능한 모든 통계가 볼륨 ID별로 필터링됩니다.

Amazon EC2 콘솔의 그래프

볼륨을 생성한 후 Amazon EC2 콘솔로 가서 볼륨의 모니터링 그래프를 볼 수 있습니다. 콘솔의 [Volumes] 페이지에서 볼륨을 선택하고 [Monitoring]을 선택합니다. 다음 표에는 표시되는 그래프가 나열되어 있습니다. 오른쪽 열에는 CloudWatch API의 원시 데이터 측정치로 각 그래프가 생성되는 방법이 설명되어 있습니다. 모든 그래프의 기간은 5분입니다.

그래프	원시 지표를 사용하여 설명
Read Bandwidth (KiB/s)	합계(VolumeReadBytes)/기간/1024
Write Bandwidth (KiB/s)	합계(VolumeWriteBytes)/기간/1024
Read Throughput (Ops/s)	합계(VolumeReadOps)/기간
Write Throughput (Ops/s)	합계(VolumeWriteOps)/기간
Avg Queue Length (ops)	평균(VolumeQueueLength)
% Time Spent Idle	합계(VolumedleTime)/기간 * 100
Avg Read Size (KiB/op)	평균(VolumeReadBytes)/1024
Avg Write Size (KiB/op)	평균(VolumeWriteBytes)/1024
Avg Read Latency (ms/op)	평균(VolumeTotalReadTime) * 1000
Avg Write Latency (ms/op)	평균(VolumeTotalWriteTime) * 1000

평균 지연 시간 그래프 및 평균 크기 그래프의 경우 기간 중 완료된 총 작업(그래프에 해당하는 읽기 또는 쓰기) 수를 기준으로 평균을 계산합니다.

상태 확인으로 볼륨 모니터링

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

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

Amazon EBS에서 볼륨의 데이터가 잠재적으로 일치하지 않는 것으로 확인하면 데이터 손상을 방지하기 위해 기본적으로 연결된 EC2 인스턴스에서 볼륨으로의 I/O가 비활성화됩니다. I/O가 비활성화되면 다음 볼륨 상태 확인에 실패하고 볼륨 상태는 `impaired`가 됩니다. 또한 I/O가 비활성화되었으며 볼륨에 대한 I/O를 활성화하여 볼륨의 손상된 상태를 해결할 수 있다고 알려주는 이벤트가 표시됩니다. I/O를 활성화한 다음 인스턴스에서 볼륨을 계속 사용할지 아니면 `fsck(Linux)` 또는 `chkdsk(Windows)`와 같은 명령을 사용하여 일관성 확인을 실행한 다음 볼륨을 사용할지 여부를 결정할 수 있습니다.

Note

볼륨 상태는 볼륨 상태 검사 결과를 기준으로 한 것으로, 볼륨 상태를 직접 반영하는 것은 아닙니다. 따라서 볼륨 상태가 `error` 상태의 볼륨을 나타내는 것은 아닙니다(예: 볼륨이 I/O를 허용할 수 없을 때).

특정 볼륨의 일관성은 문제가 되지 않고, 볼륨이 손상된 경우 볼륨을 즉시 사용할 수 있게 하려면 I/O를 자동으로 활성화하도록 볼륨을 구성하여 기본 동작을 무시할 수 있습니다. `AutoEnableIO` 볼륨 속성을 활성화하면 볼륨 상태 확인을 계속 통과합니다. 또한 볼륨이 잠재적으로 일치하지 않는 것으로 확인되었지만 I/O가 자동으로 활성화되었다고 알려주는 이벤트가 표시됩니다. 그러면 볼륨의 일관성을 확인하거나 나중에 볼륨을 교체할 수 있습니다.

I/O 성능 상태 확인은 실제 볼륨 성능을 볼륨의 예상 성능과 비교하고 볼륨 성능이 예상보다 낮은 경우 알림을 표시합니다. 이 상태 검사는 인스턴스에 연결된 `io1` 볼륨에만 사용할 수 있고 범용 SSD(`gp2`), 처리량에 최적화된 HDD(`st1`), Cold HDD(`sc1`) 또는 Magnetic(`standard`) 볼륨에는 사용할 수 없습니다. I/O 성능 상태 확인은 1분마다 수행되고 CloudWatch에서 이 데이터를 5분마다 수집하므로 `io1` 볼륨을 인스턴스에 연결한 후 이 확인에서 I/O 성능 상태를 보고하는 데 최대 5분 정도 걸릴 수 있습니다.

Important

스냅샷에서 복원한 `io1` 볼륨을 초기화할 경우 볼륨의 성능이 예상 수준보다 50퍼센트 이하로 떨어질 수 있으며, 이로 인해 볼륨에서 [I/O Performance] 상태 확인에 대해 `warning` 상태를 표시할 수 있습니다. 이는 원래 그런 것인므로 초기화 중에는 `io1` 볼륨에 대한 `warning` 상태를 무시해도 됩니다. 자세한 내용은 [Amazon EBS 볼륨 초기화 \(p. 628\)](#) 섹션을 참조하십시오.

다음 표에는 Amazon EBS 볼륨에 대한 상태가 나와 있습니다.

볼륨 상태	I/O 활성화 상태	I/O 성능 상태(프로비저닝된 IOPS 볼륨에만 사용 가능)
<code>ok</code>	활성화됨(I/O 활성화 또는 I/O 자동 활성화)	정상(볼륨 성능이 예상대로임)
<code>warning</code>	활성화됨(I/O 활성화 또는 I/O 자동 활성화)	성능 저하(볼륨 성능이 예상보다 낮음) 심각한 성능 저하(볼륨 성능이 예상보다 훨씬 낮음)
<code>impaired</code>	활성화됨(I/O 활성화 또는 I/O 자동 활성화)	중단됨(볼륨 성능이 저하됨)

볼륨 상태	I/O 활성화 상태	I/O 성능 상태(프로비저닝된 IOPS 볼륨에만 사용 가능)
	비활성화됨(볼륨이 오프라인이고 복구 보류 중이거나 사용자가 I/O 를 활성화하기를 기다리는 중)	사용할 수 없음(I/O가 비활성화되어 I/O 성능을 확인할 수 없음)
insufficient-data	활성화됨(I/O 활성화 또는 I/O 자동 활성화) 데이터 부족	데이터 부족

상태 확인을 보면서 작업하려면 Amazon EC2 콘솔, API 또는 명령줄 인터페이스를 사용합니다.

콘솔에서 상태 확인을 보려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Volumes]를 선택합니다.
3. [EBS Volumes] 페이지에서 각 볼륨의 작업 상태가 나열된 [Volume Status] 열을 사용합니다.
4. 개별 볼륨의 상태를 보려면 볼륨을 선택하고 [Status Checks]를 선택합니다.
5. 상태 확인에 실패한 볼륨이 있는 경우(*impaired* 상태) [손상된 볼륨 작업 \(p. 585\)](#) 섹션을 참조하십시오.

또는 [Events] 창을 사용하여 인스턴스와 볼륨에 대한 모든 이벤트를 단일 창에서 볼 수 있습니다. 자세한 내용은 [볼륨 이벤트 모니터링 \(p. 584\)](#) 섹션을 참조하십시오.

명령줄로 볼륨 상태 정보를 보려면

다음 명령 중 하나를 사용하여 Amazon EBS 볼륨의 상태를 볼 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [describe-volume-status\(AWS CLI\)](#)
- [Get-EC2VolumeStatus\(Windows PowerShell용 AWS 도구\)](#)

볼륨 이벤트 모니터링

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

데이터가 잠재적으로 일치하지 않는 볼륨에서 I/O를 자동으로 활성화하려면 `AutoEnableIO` 볼륨 속성의 설정을 변경합니다. 이 속성 변경에 대한 자세한 내용은 [손상된 볼륨 작업 \(p. 585\)](#) 섹션을 참조하십시오.

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

볼륨 상태 이벤트는 다음 설명 중 하나를 포함합니다.

Awaiting Action: Enable IO

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

IO Enabled

이 볼륨에 대해 I/O 작업이 명시적으로 활성화되었습니다.

IO Auto-Enabled

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

보통

`io1` 볼륨 전용입니다. 볼륨 성능이 예상대로입니다.

성능 저하

`io1` 볼륨 전용입니다. 볼륨 성능이 예상보다 낮습니다.

Severely Degraded

`io1` 볼륨 전용입니다. 볼륨 성능이 예상보다 훨씬 낮습니다.

Stalled

`io1` 볼륨 전용입니다. 볼륨 성능이 저하되었습니다.

Amazon EC2 콘솔, API 또는 명령줄 인터페이스를 사용하여 볼륨에 대한 이벤트를 볼 수 있습니다.

콘솔에서 볼륨에 대한 이벤트를 보려면

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

I/O가 비활성화된 볼륨이 있는 경우 [손상된 볼륨 작업 \(p. 585\)](#) 섹션을 참조하십시오. I/O 성능이 정상보다 낮은 볼륨이 있는 경우 수행한 작업(예: 피크 사용 동안 볼륨 스냅샷 생성, 필요한 I/O 대역폭을 지원할 수 없는 인스턴스에서 볼륨 실행, 볼륨의 데이터에 처음 액세스 등)으로 인한 일시적인 현상일 수 있습니다.

명령줄로 볼륨에 대한 이벤트를 보려면

다음 명령 중 하나를 사용하여 Amazon EBS 볼륨에 대한 이벤트 정보를 볼 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [describe-volume-status\(AWS CLI\)](#)
- [Get-EC2VolumeStatus\(Windows PowerShell용 AWS 도구\)](#)

손상된 볼륨 작업

이 섹션에서는 볼륨의 데이터가 잠재적으로 일치하지 않아서 볼륨이 손상된 경우 사용할 수 있는 옵션을 설명합니다.

옵션

- [옵션 1: 인스턴스에 연결된 볼륨에 대한 일관성 확인 수행 \(p. 585\)](#)
- [옵션 2: 다른 인스턴스를 사용하여 볼륨에 대한 일관성 확인 수행 \(p. 586\)](#)
- [옵션 3: 볼륨이 더 이상 필요하지 않은 경우 볼륨 삭제 \(p. 587\)](#)

옵션 1: 인스턴스에 연결된 볼륨에 대한 일관성 확인 수행

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

연결된 볼륨에 대해 일관성 확인을 수행하려면

1. 모든 애플리케이션의 볼륨 사용을 중지합니다.
2. 볼륨에서 I/O를 활성화합니다.
 - a. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
 - b. 탐색 창에서 [Volumes]를 선택합니다.
 - c. I/O 작업을 활성화할 볼륨을 선택합니다.
 - d. 세부 정보 창에서 [Enable Volume IO]를 선택합니다.
3. 볼륨의 데이터를 확인합니다.
 - a. fsck(Linux) 또는 chkdsk(Windows) 명령을 실행합니다.
 - b. (선택 사항) 애플리케이션 또는 시스템 로그에 관련 오류 메시지가 있는지 검토합니다.
 - c. 볼륨 손상 상태가 20분 이상 지속된 경우 고객 지원 센터에 문의할 수 있습니다. [Troubleshoot]을 선택하고 [Troubleshoot Status Checks] 대화 상자에서 [Contact Support]를 선택하여 지원 사례를 제출합니다.

명령줄로 볼륨에 대한 I/O를 활성화하려면

다음 명령 중 하나를 사용하여 Amazon EBS 볼륨에 대한 이벤트 정보를 볼 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [enable-volume-io\(AWS CLI\)](#)
- [Enable-EC2VolumeIO\(Windows PowerShell용 AWS 도구\)](#)

옵션 2: 다른 인스턴스를 사용하여 볼륨에 대한 일관성 확인 수행

다음 절차에 따라 프로덕션 환경 외부의 볼륨을 확인합니다.

Important

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

격리 중인 볼륨에 대한 일관성 확인을 수행하려면

1. 모든 애플리케이션의 볼륨 사용을 중지합니다.
2. 인스턴스에서 볼륨을 분리합니다.
 - a. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
 - b. 탐색 창에서 [Volumes]를 선택합니다.
 - c. 분리할 볼륨을 선택합니다.
 - d. [Actions], [Force Detach Volume]을 선택합니다. 확인 메시지가 나타납니다.
3. 볼륨에서 I/O를 활성화합니다.
 - a. 탐색 창에서 [Volumes]를 선택합니다.
 - b. 이전 단계에서 분리한 볼륨을 선택합니다.
 - c. 세부 정보 창에서 [Enable Volume IO]를 선택합니다.
4. d. [Enable Volume IO] 대화 상자에서 [Yes, Enable]을 선택합니다.
4. 볼륨을 다른 인스턴스에 연결합니다. 자세한 내용은 [인스턴스 시작 \(p. 264\)](#) 및 [Amazon EBS 볼륨을 인스턴스에 연결 \(p. 576\)](#) 섹션을 참조하십시오.

5. 볼륨의 데이터를 확인합니다.

- a. fsck(Linux) 또는 chkdsk(Windows) 명령을 실행합니다.
- b. (선택 사항) 애플리케이션 또는 시스템 로그에 관련 오류 메시지가 있는지 검토합니다.
- c. 볼륨 손상 상태가 20분 이상 지속된 경우 고객 지원 센터에 문의할 수 있습니다. [Troubleshoot]을 선택하고 문제 해결 대화 상자에서 [Contact Support]를 선택하여 지원 사례를 제출합니다.

명령줄로 볼륨에 대한 I/O를 활성화하려면

다음 명령 중 하나를 사용하여 Amazon EBS 볼륨에 대한 이벤트 정보를 볼 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [enable-volume-io\(AWS CLI\)](#)
- [Enable-EC2VolumeIO\(Windows PowerShell용 AWS 도구\)](#)

옵션 3: 볼륨이 더 이상 필요하지 않은 경우 볼륨 삭제

환경에서 볼륨을 제거하려면 볼륨을 삭제하면 됩니다. 볼륨 삭제에 대한 자세한 내용은 [Amazon EBS 볼륨 삭제 \(p. 589\)](#) 섹션을 참조하십시오.

볼륨의 데이터를 백업하는 최근 스냅샷이 있는 경우 해당 스냅샷에서 새 볼륨을 생성할 수 있습니다. 스냅샷에서 볼륨 생성에 대한 자세한 내용은 [스냅샷에서 Amazon EBS 볼륨 복구 \(p. 574\)](#) 섹션을 참조하십시오.

AutoEnableIO 볼륨 속성 작업

Amazon EBS에서 볼륨의 데이터가 잠재적으로 일치하지 않는 것으로 확인하면 기본적으로 연결된 EC2 인스턴스에서 볼륨으로의 I/O가 비활성화됩니다. 그러면 볼륨 상태 확인에 실패하고 실패의 원인을 나타내는 볼륨 상태 이벤트가 생성됩니다. 특정 볼륨의 일관성은 문제가 되지 않고, 볼륨이 impaired 상태인 경우 볼륨을 즉시 사용할 수 있게 하려면 I/O를 자동으로 활성화하도록 볼륨을 구성하여 기본 동작을 무시할 수 있습니다. AutoEnableIO 볼륨 속성을 활성화하면 볼륨과 인스턴스 사이의 I/O가 자동으로 다시 활성화되고 볼륨 상태 확인을 통과합니다. 또한 볼륨이 잠재적으로 일치하지 않는 상태인 것으로 결정되었지만 I/O가 자동으로 활성화되었다고 알려주는 이벤트가 표시됩니다. 이 이벤트가 발생하면 볼륨의 일관성을 확인하고 필요한 경우 볼륨을 교체해야 합니다. 자세한 내용은 [볼륨 이벤트 모니터링 \(p. 584\)](#) 섹션을 참조하십시오.

이 섹션에서는 Amazon EC2 콘솔, 명령줄 인터페이스 또는 API를 사용하여 볼륨의 AutoEnableIO 속성을 보고 수정하는 방법을 설명합니다.

콘솔에서 볼륨의 AutoEnableIO 속성을 보려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Volumes]를 선택합니다.
3. 볼륨을 선택합니다.
4. 아래쪽 창에서 [Status Checks]를 선택합니다.
5. [Status Checks] 탭에서 [Auto-Enable IO]는 볼륨의 현재 설정(Enabled 또는 Disabled)을 표시합니다.

콘솔에서 볼륨의 AutoEnableIO 속성을 수정하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Volumes]를 선택합니다.
3. 볼륨을 선택합니다.
4. [Volumes] 페이지 위쪽에서 [Actions]를 선택합니다.
5. [Change Auto-Enable IO Setting]을 선택합니다.

6. [Change Auto-Enable IO Setting] 대화 상자에서 [Auto-Enable Volume IO] 옵션을 선택하여 손상된 볼륨에 대한 I/O를 자동으로 활성화합니다. 이 기능을 비활성화하려면 옵션 선택을 취소합니다.
7. [Save]를 선택합니다.

또는 이전 절차의 4~6단계를 수행하지 않고 [Status Checks], [Edit]를 클릭합니다.

명령줄로 볼륨의 AutoEnableIO 속성을 보거나 수정하려면

다음 명령 중 하나를 사용하여 Amazon EBS 볼륨의 AutoEnableIO 속성을 볼 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [describe-volume-attribute](#)(AWS CLI)
- [Get-EC2VolumeAttribute](#)(Windows PowerShell용 AWS 도구)

볼륨의 AutoEnableIO 속성을 수정하려면 다음 명령 중 하나를 사용합니다.

- [modify-volume-attribute](#)(AWS CLI)
- [Edit-EC2VolumeAttribute](#)(Windows PowerShell용 AWS 도구)

인스턴스에서 Amazon EBS 볼륨 분리

인스턴스에서 Amazon EBS 볼륨을 분리하거나 인스턴스를 종료하는 것이 가능합니다. 그러나 인스턴스가 실행 중인 경우 인스턴스에서 먼저 해당 볼륨의 마운트를 해제해야 합니다.;

EBS 볼륨이 인스턴스의 루트 디바이스인 경우에는 볼륨을 분리하기 전에 인스턴스를 중지해야 합니다.

AWS Marketplace 제품 코드가 있는 볼륨을 인스턴스에서 분리하면 제품 코드는 더 이상 인스턴스와 연결되어 있지 않습니다.

Important

볼륨을 해제한 이후에도 스토리지 용량이 AWS 프리 티어 한도를 초과할 경우 볼륨 스토리지에 대해 비용이 계속해서 청구됩니다. 추가 비용이 청구되지 않도록 하려면 볼륨을 삭제해야 합니다. 자세한 내용은 [Amazon EBS 볼륨 삭제 \(p. 589\)](#) 섹션을 참조하십시오.

이 예제에서는 볼륨의 마운트를 해제한 다음 인스턴스에서 명시적으로 분리합니다. 인스턴스를 종료하거나 볼륨을 다른 인스턴스에 연결하려고 할 때 이 예제가 유용하게 사용될 수 있습니다. 볼륨이 인스턴스에 더 이상 연결되어 있지 않은지를 확인하려면 [볼륨 정보 보기 \(p. 579\)](#) 섹션을 참조하십시오.

분리한 볼륨을 다시 연결하는 것이 가능하지만(마운트를 해제하지 않고) 마운트 포인트가 동일하지 않을 수 있어 분리된 상태에서 볼륨에 쓰기가 수행된 경우 볼륨 데이터의 동기화가 일치하지 않을 수 있습니다.

콘솔을 이용하여 EBS 볼륨을 분리하려면

1. 다음 명령어를 사용하여 /dev/sdh 디바이스의 마운트를 해제합니다.

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

2. <https://console.aws.amazon.com/ec2>에서 Amazon EC2 콘솔을 엽니다.
3. 탐색 창에서 [Volumes]를 선택합니다.
4. 볼륨을 선택한 후 [Actions], [Detach Volume]을 선택합니다.

5. 확인 대화 상자에서 [Yes, Detach]를 선택합니다.

명령줄을 사용하여 인스턴스에서 EBS 볼륨을 분리하려면

다음 명령 중 하나를 사용할 수 있습니다. 이러한 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [detach-volume](#) (AWS CLI)
- [Dismount-EC2Volume](#) (Windows PowerShell용 AWS 도구)

문제 해결

이 섹션에서는 볼륨을 해제할 때 발생할 수 있는 일반적인 문제와 해결 방법에 대해 설명합니다.

Note

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

- Amazon EC2 콘솔을 통해 볼륨을 분리하는 동안 문제가 발생할 경우 [describe-volumes](#) CLI 명령을 사용하여 문제를 진단하는 것이 좋습니다. 자세한 내용은 [describe-volumes](#) 섹션을 참조하십시오.
- 볼륨이 [detaching](#) 상태를 유지하는 경우 [Force Detach]를 선택하여 강제 분리할 수 있습니다. 이 옵션은 오류가 발생한 인스턴스에서 볼륨 분리 또는 삭제할 목적으로 볼륨을 분리하는 경우에만 최후의 수단으로 사용하십시오. 인스턴스는 파일 시스템 캐시 또는 파일 시스템 메타데이터를 플러시하지 않습니다. 이 옵션을 사용하는 경우 파일 시스템 확인 및 복구 절차를 수행해야 합니다.
- 몇 분 동안 강제 볼륨 분리를 수차례 시도하였지만 [detaching](#) 상태가 계속해서 유지되는 경우 [Amazon EC2 forum](#)에 도움을 요청하십시오. 해결 방법을 신속히 찾아내려면 볼륨 ID를 기재하고 어떤 단계를 수행했는지에 대해 설명하십시오.
- 아직 마운트되어 있는 볼륨을 분리하려는 경우 분리 시도 중에 볼륨이 [busy](#) 상태로 고착될 수 있습니다. 다음의 [describe-volumes](#) 출력 화면은 이 조건을 보여주는 예입니다.

```
[ec2-user ~]$ aws ec2 describe-volumes --region us-west-2 --volume-ids vol-1234abcd
{
    "Volumes": [
        {
            "AvailabilityZone": "us-west-2b",
            "Attachments": [
                {
                    "AttachTime": "2016-07-21T23:44:52.000Z",
                    "InstanceId": "i-fedc9876",
                    "VolumeId": "vol-1234abcd",
                    "State": "busy",
                    "DeleteOnTermination": false,
                    "Device": "/dev/sdf"
                }
            ],
            ...
        }
    ]
}
```

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

Amazon EBS 볼륨 삭제

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

볼륨을 삭제하려면 볼륨이 available 상태(인스턴스에 연결되지 않음)여야 합니다.

콘솔을 사용하여 EBS 볼륨을 삭제하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Volumes]를 선택합니다.
3. 볼륨을 선택한 후 [Actions], [Delete Volume]을 선택합니다.
4. 확인 대화 상자에서 [Yes, Delete]를 선택합니다.

명령줄을 사용하여 EBS 볼륨을 삭제하려면

다음 명령 중 하나를 사용할 수 있습니다. 이러한 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [delete-volume \(AWS CLI\)](#)
- [Remove-EC2Volume \(Windows PowerShell용 AWS 도구\)](#)

Linux에서 EBS 볼륨의 크기, IOPS 또는 유형 설정

Amazon EBS 볼륨이 현재 세대의 EC2 인스턴스 유형에 연결되면 연결을 해제하지 않고도 크기를 늘리고 볼륨 유형을 변경하거나 (io1 볼륨의 경우) IOPS 성능을 조정할 수 있습니다. 이러한 변경 사항을 분리된 볼륨에도 적용할 수 있습니다. 현재 세대 인스턴스 유형에 대한 자세한 내용은 [현재 세대 인스턴스](#)를 참조하십시오.

이전 세대 인스턴스 유형을 사용하거나 볼륨을 수정하려는 중에 오류가 발생한 경우, [부록: EBS 볼륨을 수정하기 위한 인스턴스 시작 및 중지 \(p. 597\)](#)의 절차를 따르십시오.

일반적으로 볼륨 수정 단계는 다음과 같습니다.

1. 수정 명령을 실행합니다. 자세한 내용은 [콘솔에서 EBS 볼륨 수정 \(p. 590\)](#) 및 [명령줄에서 EBS 볼륨 수정 \(p. 591\)](#)을(를) 참조하십시오.
2. 수정 진행률을 모니터링합니다. 자세한 내용은 [볼륨 수정 진행률 모니터링 \(p. 592\)](#) 섹션을 참조하십시오.
3. 볼륨 크기가 수정된 경우 볼륨의 파일 시스템을 확장하여 스토리지 용량 증가를 활용합니다. 자세한 내용은 [볼륨 크기 조정 후 Linux 파일 시스템 확장 \(p. 594\)](#)을(를) 참조하십시오.

또한 [Amazon CloudWatch Events](#)와 [AWS CloudFormation](#)을 사용하여 볼륨 수정에 연결된 작업을 자동화할 수 있습니다.

볼륨 구성 수정은 무료입니다. 수정이 시작된 후 새 볼륨 구성 가격으로 요금이 청구됩니다. 자세한 내용은 [Amazon EC2 요금](#) 페이지의 Amazon Elastic Block Store 섹션을 참조하십시오.

자세한 내용은 [EBS 볼륨 수정을 위한 고려 사항 \(p. 596\)](#) 섹션을 참조하십시오.

Important

중요한 데이터가 저장된 볼륨을 변경하려면 먼저 변경 내용을 룰백해야 할 경우를 대비하여 볼륨 스냅샷을 생성하는 것이 바람직합니다. EBS 스냅샷에 대한 자세한 내용은 [Creating an Amazon EBS Snapshot](#) 단원을 참조하십시오.

콘솔에서 EBS 볼륨 수정

다음 절차는 Amazon EC2 콘솔에서 사용 가능한 볼륨 수정을 적용하는 방법을 보여줍니다.

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. [Volumes]를 선택하고 수정할 볼륨을 선택한 후 [Actions], [Modify Volume]을 선택합니다.
3. [Modify Volume] 창에 볼륨 ID와 유형, 크기, IOPS를 포함한 볼륨의 현재 구성이 표시됩니다. 한 번의 작업으로 이 모든 설정을 변경할 수 있습니다. 다음과 같이 새로운 구성 값을 설정합니다.
 - 유형을 수정하려면 [Volume Type]의 값을 선택합니다.
 - 크기를 수정하려면 [Size]에 허용된 정수 값을 입력합니다.
 - 볼륨 유형으로 [Provisioned IOPS (IO1)]를 선택한 경우 [IOPS]에 허용된 정수 값을 입력합니다.
4. 적용할 수정 사항을 모두 지정한 후 [Modify], [Yes]를 차례로 선택합니다.

Note

볼륨 크기를 수정해도 새 스토리지 용량을 활용하기 위해 볼륨의 파일 시스템도 확장할 때까지는 실질적인 효과가 없습니다. 자세한 내용은 [볼륨 크기 조정 후 Linux 파일 시스템 확장 \(p. 594\)](#) 섹션을 참조하십시오.

명령줄에서 EBS 볼륨 수정

다음 예제는 AWS CLI를 사용하여 명령줄에서 EBS 볼륨을 어떻게 수정할 수 있는지 보여줍니다. 기본 구성에 따라 리전과 가용 영역 등의 정보를 지정해야 할 수 있습니다. 수정 중인 소스 볼륨의 ID가 필요하고 작업을 수행하기 위한 적절한 권한이 있어야 합니다. io1 볼륨이 수정 대상인 경우 프로비저닝된 IOPS의 수준을 지정해야 합니다. 단일 명령으로 여러 건의 수정 작업(용량, IOPS 또는 유형 변경)을 수행할 수 있습니다.

예를 들어 EBS 볼륨은 다음과 같이 구성됩니다.

- 볼륨 ID: vol-1111111111111111
- 볼륨 크기: 100GiB
- 볼륨 유형: gp2

볼륨 구성을 다음으로 변경할 수 있습니다.

- 볼륨 크기: 200GiB
- 볼륨 유형: io1
- 프로비저닝 수준: 10,000 IOPS

다음 명령으로 위 수정 사항을 적용합니다.

```
aws ec2 modify-volume --region us-east-1 --volume-id vol-1111111111111111 --size 200 --volume-type io1 --iops 10000
```

이 명령으로 다음과 비슷한 출력이 생성됩니다.

```
{  
    "VolumeModification": {  
        "TargetSize": 200,  
        "TargetVolumeType": "io1",  
        "ModificationState": "modifying",  
        "VolumeId": "vol-1111111111111111",  
        "TargetIops": 10000,  
        "StartTime": "2017-01-19T22:21:02.959Z",  
        "Progress": 0,  
        "OriginalVolumeType": "gp2",  
        "OriginalIops": 300,  
        "CurrentVolumeType": "io1",  
        "CurrentIops": 10000  
    }  
}
```

```
        "OriginalSize": 100
    }
}
```

Note

볼륨 크기를 수정해도 새 스토리지 용량을 활용하기 위해 볼륨의 파일 시스템도 확장할 때까지는 실질적인 효과가 없습니다. 자세한 내용은 [볼륨 크기 조정 후 Linux 파일 시스템 확장 \(p. 594\)](#) 섹션을 참조하십시오.

볼륨 수정 진행률 모니터링

수정 중인 EBS 볼륨이 상태 시퀀스를 통과합니다. 콘솔, CLI, API 또는 SDK에서 `ModifyVolume` 명령을 실행한 후 볼륨이 첫 번째 `Modifying` 상태를 입력한 후 `Optimizing` 상태에 이어 `Complete` 상태를 입력합니다. 그러면 볼륨을 더 수정할 준비가 완료됩니다. 드물지만 일시적 AWS 결함으로 인해 `Failed` 상태가 될 수 있습니다. 이 경우 다시 수정을 시도하십시오.

크기를 변경하면 볼륨이 `Optimizing` 상태가 된 후 완료되어 적용되는 데 대개 몇 초가 걸립니다.

성능(IOPS) 변경이 완료되는 데 몇 분에서 몇 시간이 걸릴 수 있으며, 시간은 현재 수행 중인 구성 변경에 따라 달라집니다.

새 구성이 적용되는 데 24시간이 걸릴 수 있습니다. 일반적으로 충분히 사용된 1TiB 볼륨이 새 성능 구성으로 마이그레이션하는 데 약 6시간이 걸립니다.

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

AWS Management Console을 검사하거나 AWS EC2 API/CLI로 볼륨의 상태를 큐리하거나 Amazon CloudWatch Events로 전송된 측정치에 액세스하여 수정 진행률을 모니터링할 수 있습니다. 다음 절차는 이러한 접근 방식을 보여줍니다.

콘솔에서 수정 진행률 모니터링하기

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. [Volumes]를 선택하고 검사할 볼륨을 선택합니다. 볼륨의 상태가 [State] 열에 표시됩니다. 아래 예에서 수정 상태는 [completed]입니다. 이 상태 정보는 세부 정보 창의 [State] 필드에도 표시됩니다.
3. 아래 그림과 같이 [State] 필드 옆의 정보 아이콘을 열어 최근 수정 작업에 대한 모든 전/후 정보를 표시합니다.

명령줄에서 수정 진행률 모니터링하기

- [describe-volumes-modifications \(p. 591\)](#)를 사용하여 수정 진행률을 확인합니다. 이 예에서는 위의 `vol-1111111111111111` 볼륨 및 다른 볼륨, `vol-2222222222222222`가 호출됩니다.

```
aws ec2 describe-volumes-modifications --region us-east-1 --volume-id vol-1111111111111111 vol-2222222222222222
```

이 명령으로 다음과 비슷한 출력이 생성됩니다.

```
{
  "VolumesModifications": [
    {
      "TargetSize": 200,
      "TargetVolumeType": "io1",
```

```
"ModificationState": "modifying",
"VolumeId": "vol-1111111111111111",
"TargetIops": 10000,
"StartTime": "2017-01-19T22:21:02.959Z",
"Progress": 0,
"OriginalVolumeType": "gp2",
"OriginalIops": 300,
"OriginalSize": 100
},
{
    "TargetSize": 2000,
    "TargetVolumeType": "sc1",
    "ModificationState": "modifying",
    "VolumeId": "vol-2222222222222222",
    "StartTime": "2017-01-19T22:23:22.158Z",
    "Progress": 0,
    "OriginalVolumeType": "gp2",
    "OriginalIops": 300,
    "OriginalSize": 1000
}
]
```

`describe-volumes-modifications` 명령은 하나 이상의 `VolumesModification` 객체를 반환합니다. 이 예에서 두 번째는 위에 표시된 원본 `modify-volume` 명령 출력과 거의 동일합니다. 하지만 추가 수정 사항은 적용되지 않았습니다.

다음 예에서는 수정 상태가 `optimizing` 또는 `completed`인 리전에서 모든 볼륨에 대해 쿼리한 후 결과를 필터링하고 형식을 지정하여 2017년 2월 1일 이후 시작된 수정만 표시합니다.

```
aws ec2 describe-volumes-modifications --filters Name=modification-state,Values="optimizing","completed" --region us-east-1 --query "VolumesModifications[?StartTime>='2017-02-01'].{ID:VolumeId,STATE:ModificationState}"
```

이 경우 쿼리는 두 개의 볼륨에 대한 정보를 반환합니다.

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

CloudWatch 이벤트로 수정 진행을 모니터링하기

CloudWatch 이벤트를 이용하면 볼륨 수정 이벤트에서 텍스트 메시지를 보내거나 Lambda 함수를 실행하도록 알림 규칙을 생성할 수 있습니다.

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. [Events], [Create rule]을 선택합니다.
3. [Build event pattern to match events by service]에 대해 [Custom event pattern]을 선택합니다.
4. [Build custom event pattern]에 대해 다음 코드로 콘텐츠를 바꿉니다.

```
{
```

```
"source": [  
    "aws.ec2"  
,  
    "detail-type": [  
        "EBS Volume Notification"  
,  
        "detail": {  
            "event": [  
                "modifyVolume"  
            ]  
        }  
    }  
}
```

완료되면 [Save]를 선택합니다.

일반적인 이벤트 출력은 다음과 같습니다.

```
Body:  
{  
    "version": "0",  
    "id": "1ea2ace2-7790-46ed-99ab-d07a8bd68685",  
    "detail-type": "EBS Volume Notification",  
    "source": "aws.ec2",  
    "account": "065441870323",  
    "time": "2017-01-12T21:09:07Z",  
    "region": "us-east-1",  
    "resources": [  
        "arn:aws:ec2:us-east-1:065441870323:volume/vol-03a55cf56513fa1b6"  
    ],  
    "detail": {  
        "result": "optimizing",  
        "cause": "",  
        "event": "modifyVolume",  
        "request-id": "auto-58c08bad-d90b-11e6-a309-b51ed35473f8"  
    }  
}
```

규칙을 사용하여 Amazon SNS로 알림 메시지를 생성하거나 일치하는 이벤트에 대한 응답으로 AWS Lambda 함수를 호출할 수 있습니다.

볼륨 크기 조정 후 Linux 파일 시스템 확장

파일 시스템 관련 명령을 사용하여 더 큰 새 볼륨에 맞게 파일 시스템의 크기를 조정합니다. 이 명령은 확장 할 볼륨이 루트 볼륨인 경우에도 작동합니다. ext2, ext3 및 ext4 파일 시스템의 경우 이 명령은 resize2fs입니다. XFS 파일 시스템의 경우 이 명령은 xfs_growfs입니다. 다른 파일 시스템에 대한 확장 지침은 해당 파일 시스템의 설명서를 참조하십시오.

사용 중인 파일 시스템을 모르는 경우 file -s 명령을 사용하여 디바이스에 대한 파일 시스템 데이터를 나열합니다. 다음 예는 Linux ext4 파일 시스템과 SGI XFS 파일 시스템을 보여 줍니다.

```
[ec2-user ~]$ sudo file -s /dev/xvd*
/dev/xvda1: Linux rev 1.0 ext4 filesystem data ...
/dev/xvdf:  SGI XFS filesystem data ...
```

Note

확장하려는 볼륨이 파티셔닝된 경우 파일 시스템의 크기를 조정하려면 파티션의 크기를 늘려야 합니다. 이때 추가 파티션을 할당할 수도 있습니다. 자세한 내용은 [Linux 파티션 확장 \(p. 598\)](#) 섹션을 참조하십시오.

볼륨이 `optimizing` 상태가 되자마자 파일 시스템 크기 조정을 시작할 수 있습니다.

Important

중요한 데이터가 저장된 파일 시스템을 확장하려면 먼저 변경 내용을 룰백해야 할 경우를 대비하여 파일 시스템이 저장된 볼륨 스냅샷을 생성하는 것이 바람직합니다. EBS 스냅샷에 대한 자세한 내용은 [Creating an Amazon EBS Snapshot](#) 단원을 참조하십시오.

볼륨 파티션의 크기를 조정해야 하는지 확인하려면

- `lsblk` 명령을 사용하여 인스턴스에 연결된 블록 디바이스를 나열합니다. 아래 예는 `/dev/xvda`, `/dev/xvdb` 및 `/dev/xvdf` 볼륨을 보여 줍니다.

```
[ec2-user ~]$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda    202:0    0  30G  0 disk
##xvda1 202:1    0  30G  0 part /
xvdb    202:16   0  30G  0 disk /mnt
xvdf    202:80   0  35G  0 disk
##xvdf1 202:81   0   8G  0 part
```

루트 볼륨인 `/dev/xvda1`은 `/dev/xvda`의 파티션입니다. 두 볼륨의 크기는 모두 30GiB입니다. 이 경우 파티션이 디바이스의 전체 공간을 차지하므로 크기를 조정할 필요가 없습니다.

`/dev/xvdb` 볼륨은 파티션되지 않았으므로 크기를 조정할 필요가 없습니다.

그러나 `/dev/xvdf1`은 35GiB 디바이스의 8GiB 파티션이고 볼륨에 다른 파티션이 없습니다. 이 경우 볼륨의 나머지 공간을 사용하려면 파티션의 크기를 조정해야 합니다. 자세한 내용은 [Linux 파티션 확장 \(p. 598\)](#)을 참조하십시오. 파티션의 크기를 조정한 후 다음 절차에 따라 파티션의 전체 공간을 차지하도록 파일 시스템을 확장할 수 있습니다.

Linux 파일 시스템을 확장하려면

1. SSH 클라이언트를 사용하여 Linux 인스턴스에 로그인합니다. Linux 인스턴스 연결에 대한 자세한 내용은 [SSH를 사용하여 Linux 인스턴스에 연결 \(p. 274\)](#) 섹션을 참조하십시오.
2. `df -h` 명령을 사용하여 기존 파일 시스템 디스크 공간 사용을 보고합니다. 이 예에서 `/dev/xvda1` 디바이스는 이미 70GiB로 확장했지만 운영 체제는 여전히 원본 7.9GiB `ext4` 파일 시스템만 볼 수 있습니다. 마찬가지로 `/dev/xvdf` 디바이스는 100GiB로 확장되었으나 운영 체제는 여전히 1.0GiB XFS 파일 시스템만 볼 수 있습니다.

```
[ec2-user ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1       8.0G  943M  6.9G  12% /
tmpfs           1.9G     0  1.9G  0% /dev/shm
/dev/xvdf       1014M   33M  982M  4% /mnt
```

3. 파일 시스템 관련 명령을 사용하여 새 볼륨 용량에 맞게 파일 시스템의 크기를 조정합니다.

Linux `ext2`, `ext3` 또는 `ext4` 파일 시스템의 경우 다음 명령을 사용합니다. 여기에서 확장할 디바이스 이름을 대체합니다.

```
[ec2-user ~]$ sudo resize2fs /dev/xvda1
resize2fs 1.42.3 (14-May-2012)
Filesystem at /dev/xvda1 is mounted on /; on-line resizing required
old_desc_blocks = 1, new_desc_blocks = 5
Performing an on-line resize of /dev/xvda1 to 18350080 (4k) blocks.
The filesystem on /dev/xvda1 is now 18350080 blocks long.
```

XFS 파일 시스템의 경우, 먼저 XFS 사용자 공간 도구를 설치합니다.

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

그리고 다음 명령을 사용합니다. 여기에서 파일 시스템의 마운트 지점을 대체합니다. XFS 파일 시스템을 마운트해야 크기를 조정할 수 있습니다.

```
[ec2-user ~]$ sudo xfs_growfs -d /mnt
meta-data=/dev/xvdf          isize=256    agcount=4, agsize=65536 blks
                           =           sectsz=512  attr=2
data              =           bsize=4096   blocks=262144, imaxpct=25
                           =           sunit=0    swidth=0 blks
naming            =version 2  bsize=4096   ascii-ci=0
log               =internal   bsize=4096   blocks=2560, version=2
                           =           sectsz=512  sunit=0 blks, lazy-count=1
realtime         =none        extsz=4096   blocks=0, rtextents=0
data blocks changed from 262144 to 26214400
```

Note

`xfsctl failed: Cannot allocate memory` 오류가 표시되는 경우 Linux 커널을 업데이트해야 할 수 있습니다. 자세한 내용은 해당 운영 체제 설명서를 참조하십시오.
`The filesystem is already nnnnnnn blocks long. Nothing to do!` 오류를 받는 경우, [Linux 파티션 확장 \(p. 598\)](#)을 참조하십시오.

4. `df -h` 명령을 사용하여 기존 파일 시스템 디스크 공간 사용을 보고합니다. 이제 `ext4` 파일 시스템은 전체 70GiB로, XFS 파일 시스템은 100GiB로 표시되어야 합니다:

```
# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1       70G  951M   69G   2% /
tmpfs           1.9G     0  1.9G   0% /dev/shm
/dev/xvdf       100G   45M  100G   1% /mnt
```

Tip

볼륨의 확장된 사용 가능한 공간이 시스템에 표시되지 않는 경우 [Amazon EBS 볼륨 초기화](#)에 설명된 대로 볼륨을 다시 초기화해 보십시오.

EBS 볼륨 수정을 위한 고려 사항

볼륨 수정에 영향을 주는 다음과 같은 제한 사항과 특수 사례에 유의하십시오.

- 경우에 따라 수정을 진행하기 위해 볼륨을 분리하거나 인스턴스를 종단해야 합니다. EBS 볼륨에 수정을 적용하는 동안 오류 메시지가 표시되거나 이전 세대 인스턴스 유형에 연결된 EBS 볼륨을 수정하는 경우 다음 중 한 가지 조치를 취하십시오.
 - 루트가 아닌 볼륨의 경우, 인스턴스에서 볼륨을 분리하고 수정 사항을 적용한 다음 볼륨을 다시 연결합니다. 자세한 내용은 [인스턴스에서 Amazon EBS 볼륨 분리](#) 및 [인스턴스에 Amazon EBS 볼륨 연결](#)을 참조하십시오.
 - 루트(부트) 볼륨의 경우, 인스턴스를 종단하고 수정 사항을 적용한 다음 인스턴스를 다시 시작합니다. 자세한 내용은 [부록: EBS 볼륨을 수정하기 위한 인스턴스 시작 및 종지 \(p. 597\)](#) 섹션을 참조하십시오.
- 이 주제에서 설명하는 볼륨 수정 방법에서는 이전 세대 Magnetic 볼륨 유형이 지원되지 않습니다. 하지만 Magnetic 볼륨의 스냅샷을 만들고 다르게 구성된 EBS 볼륨으로 복원할 수 있습니다.
- EBS 볼륨 크기 축소는 지원되지 않습니다. 그러나 비슷한 볼륨을 만든 후 `rsync` 같은 애플리케이션 수준 도구를 사용하여 그 볼륨으로 데이터를 마이그레이션할 수 있습니다.
- 볼륨을 수정한 후 동일한 볼륨에 추가 수정 사항을 적용하려면 최소한 6시간 기다려야 합니다.

- Linux AMI에서 부팅 볼륨 2TiB(2,048GiB) 이상을 사용하려면 GPT 파티션 테이블과 GRUB 2가 필요합니다. 현재 여러 Linux AMI에서 부팅 볼륨을 최대 2,047GiB까지만 지원하는 MBR 파티셔닝 체계를 사용하고 있습니다. 인스턴스가 2TiB 이상의 부팅 볼륨에서 부팅되지 않는 경우 사용 중인 AMI의 부팅 볼륨 크기가 2,047GiB로 제한된 상태일 수 있습니다. 부팅 볼륨이 아닌 볼륨에는 이 Linux 인스턴스에 대한 제한이 적용되지 않습니다.
- 2016년 11월 1일 이전에 현재 세대 인스턴스에 연결된 볼륨은 다음 중 한 가지 조치를 취하여 이 주제에 설명된 수정 지원을 초기화해야 합니다.
 - 인스턴스를 종지한 후 다시 시작합니다.

Warning

인스턴스를 종지하면 인스턴스 스토어 볼륨의 데이터가 삭제됩니다. 따라서 인스턴스 스토어 볼륨에 보존하려는 데이터가 있을 경우 영구 스토리지에 백업하십시오.

- 볼륨을 분리한 후 다시 연결합니다.
한 번만 수행하면 됩니다.

볼륨이 언제 만들어졌는지 파악하려면 Amazon EC2 콘솔에서 볼륨 세부 정보 페이지를 탐색하여 [Created] 필드를 확인합니다. 볼륨 생성 시간보다 나중일 수 있는 가장 최근 연결 시간을 표시하려면 AWS CLI를 사용합니다. 다음 명령은 컷오프 날짜 이전에 가장 최근 연결된 볼륨에 대해 쿼리를 실행합니다.

```
aws ec2 describe-volumes --region us-east-1 --query "Volumes[?Attachments[?AttachTime<='2016-11-01']]".{ID:VolumeId}" --output text
```

출력은 유의해야 하는 볼륨 ID의 텍스트 목록입니다.

```
vol-0EXAMPLE  
vol-5EXAMPLE  
vol-4EXAMPLE  
vol-bEXAMPLE  
vol-0db1c57561EXAMPLE  
vol-06f90d0c16EXAMPLE
```

- 현재 세대 m3.medium 인스턴스는 볼륨 수정을 완전히 지원합니다. 그러나 일부 m3.large, m3.xlarge 및 m3.2xlarge 인스턴스는 모든 볼륨 수정 기능을 지원하지 않을 수 있습니다. 오류가 발생한 경우, [부록: EBS 볼륨을 수정하기 위한 인스턴스 시작 및 종지 \(p. 597\)](#)의 이전 세대 인스턴스 유형 관련 절차를 따르십시오.

부록: EBS 볼륨을 수정하기 위한 인스턴스 시작 및 종지

이전 세대 Amazon EC2 인스턴스를 사용하는 동안 루트(부트) 볼륨을 수정해야 하는 경우 인스턴스를 종지하고 수정 사항을 적용한 후 인스턴스를 다시 시작해야 합니다. 여기에 설명된 절차는 모든 인스턴스 유형의 EBS 볼륨을 수정하는 데 사용할 수 있습니다.

인스턴스를 종지했다가 시작할 때 다음 사항을 인식하십시오.

- 인스턴스가 VPC에서 실행 중이고 퍼블릭 IPv4 주소를 가지고 있으면 주소를 해제하고 새 퍼블릭 IPv4 주소를 제공합니다. 인스턴스는 프라이빗 IPv4 주소와 모든 탄력적 IP 주소(EIP)를 유지합니다.
- 인스턴스가 EC2-Classic에서 실행 중인 경우, AWS는 거기에 새로운 퍼블릭 및 프라이빗 IPv4 주소를 부여하고 해당 인스턴스와 연결된 모든 탄력적 IP 주소를 분리합니다. 인스턴스를 다시 시작한 후에는 모든 탄력적 IP 주소를 다시 연결해야 합니다.
- 인스턴스가 Auto Scaling 그룹에 있는 경우, Auto Scaling은 종단된 인스턴스를 비정상으로 간주해 이를 종료하고 대체 인스턴스를 시작합니다. 이를 방지하기 위해서는 그 그룹에 대한 Auto Scaling 과정을 일시적으로 중지할 수 있습니다. 자세한 내용은 Auto Scaling 사용 설명서의 [Suspend and Resume Auto Scaling Processes](#) 섹션을 참조하십시오.

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Instances]를 선택한 후 확장할 볼륨이 있는 인스턴스를 선택합니다.
3. [Shutdown Behavior]가 [Terminate]가 아닌 [Stop]으로 설정되어 있는지 확인합니다.
 - a. 인스턴스를 선택합니다.
 - b. 컨텍스트 메뉴(오른쪽 클릭)에서 [Instance Settings], [Change Shutdown Behavior]를 선택합니다.
 - c. [Shutdown behavior]가 [Terminate]로 설정되어 있는 경우 [Stop], [Apply]를 선택합니다.
- [Shutdown behavior]가 이미 [Stop]으로 설정되어 있는 경우 [Cancel]을 선택합니다.
4. 인스턴스를 중지합니다. 자세한 내용은 [인스턴스 중지 및 시작 \(p. 287\)](#) 섹션을 참조하십시오.

Warning

인스턴스를 중지하면 인스턴스 스토어 볼륨의 데이터가 삭제됩니다. 따라서 인스턴스 스토어 볼륨에 보존하려는 데이터가 있을 경우 영구 스토리지에 백업하십시오.

5. [콘솔에서 EBS 볼륨 수정 \(p. 590\)](#) 또는 [명령줄에서 EBS 볼륨 수정 \(p. 591\)](#)에 설명된 대로 EBS 볼륨을 수정합니다.
6. 인스턴스를 다시 시작합니다.
 - a. 탐색 창에서 [Instances]를 선택한 후 다시 시작할 인스턴스를 선택합니다.
 - b. 컨텍스트 메뉴(오른쪽 클릭)에서 [Instance State], [Start]를 선택합니다.
 - c. [Start Instances] 대화 상자에서 [Yes, Start]를 선택합니다. 인스턴스가 시작되지 않고 확장하려는 볼륨이 루트 볼륨인 경우 원래 볼륨과 동일한 디바이스 이름을 사용하여 확장된 볼륨을 연결했는지 확인합니다(예: /dev/sda1).

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

```
[ec2-user ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1       7.9G  943M  6.9G  12% /
tmpfs            1.9G     0  1.9G   0% /dev/shm
```

새로 확장된 볼륨이 크기에 반영되지 않을 경우 인스턴스에서 새 공간을 사용할 수 있도록 디바이스의 파일 시스템을 확장해야 합니다. 자세한 내용은 [볼륨 크기 조정 후 Linux 파일 시스템 확장 \(p. 594\)](#) 섹션을 참조하십시오.

Linux 파티션 확장

스냅샷에서 복원된 일부 Amazon EC2 루트 볼륨 및 볼륨은 파일 시스템과 데이터를 실제로 보유한 파티션을 포함하고 있습니다. 볼륨을 하나의 컨테이너로 생각한다면, 파티션은 볼륨 안의 또 다른 컨테이너라고 볼 수 있으며 데이터는 그 파티션에 존재하게 됩니다. 볼륨 크기를 늘려도 파티션 크기는 늘어나지 않습니다. 따라서 볼륨의 크기가 커진 경우 그 이점을 활용하려면 파티션의 크기를 늘릴 필요가 있습니다.

Note

스냅샷에서 복원된 모든 볼륨이 파티션 처리되지는 않으며, 파티션 처리는 사용자의 볼륨에만 적용됩니다. 모든 공간을 사용 가능하게 하려면 볼륨의 파일 시스템의 크기를 조정하기만 하면 됩니다. 볼륨에 크기 조절을 필요로 하는 파티션이 있는지 확실하지 않은 경우는 [볼륨 파티션의 크기를 조정해야 하는지 확인하려면 \(p. 595\)](#) 섹션을 참조해서 이를 해결하십시오.

크기를 조정하려는 파티션이 루트 파티션이 아닌 경우, 파티션의 마운트 해제를 하고 인스턴스 자체에서 파티션의 크기를 조정할 수 있습니다. 크기를 조정하려는 파티션이 인스턴스에 대한 루트 파티션인 경우, 실행 중인 인스턴스의 루트 파티션의 마운트 해제를 할 수 없기 때문에 절차는 보다 복잡해집니다. 이 경우 다른 인스턴스(보조 인스턴스라고 지칭)에서 다음 절차를 수행해야 합니다.

Important

다음 절차는 Amazon Linux에서 기록되고 테스트됩니다. 다른 도구 세트 및 도구 버전을 가진 기타 배포 버전들은 다르게 동작할 수 있습니다.

항목

- [확장을 위한 Linux 루트 파티션 준비 \(p. 599\)](#)
- [parted 명령을 사용한 Linux 파티션 확장 \(p. 599\)](#)
- [gdisk 명령을 사용한 Linux 파티션 확장 \(p. 603\)](#)
- [확장한 파티션을 원래 인스턴스로 반환 \(p. 607\)](#)

확장을 위한 Linux 루트 파티션 준비

인스턴스의 루트 파티션을 확장하기 위해서는 여러 절차를 수행해야 합니다. 확장할 파티션이 루트 파티션이 아닌 경우는 이 절차가 필요하지 않습니다.

확장을 위한 Linux 루트 파티션 준비 방법

1. 기본 인스턴스가 실행 중인 경우 이를 중단시킵니다. 실행 중인 인스턴스에서는 이 절차의 나머지 단계를 수행할 수 없습니다. 자세한 내용은 [인스턴스 중지 및 시작 \(p. 285\)](#) 섹션을 참조하십시오.
2. 볼륨의 무결성을 확인합니다. 스냅샷에서 파일 시스템 손상이 포착되는 경우, 복원된 루트 볼륨을 부팅할 수 없다는 뜻일 수 있습니다.
3. 사용자 볼륨의 스냅샷을 생성합니다. 이후 진행 절차에서는 데이터가 손상되거나 손실될 가능성이 높습니다. 따라서 원본의 스냅샷을 가지고 있으면 오류가 발생한 경우 언제나 다시 시작하고 데이터를 안전하게 유지할 수 있습니다. 자세한 내용은 [Amazon EBS 스냅샷 생성 \(p. 608\)](#) 섹션을 참조하십시오.
4. 볼륨이 연결된 디바이스 이름을 기록합니다. 인스턴스의 세부 정보 창의 Root device(루트 디바이스) 필드에서 이 정보를 확인할 수 있습니다. 그 이름 값은 `/dev/sda1` 또는 `/dev/xvda`인 경우가 많습니다.
5. 기본 인스턴스에서 볼륨을 분리합니다. 자세한 내용은 [인스턴스에서 Amazon EBS 볼륨 분리 \(p. 588\)](#) 섹션을 참조하십시오.
6. 볼륨을 동일 가용 영역에 위치하고 있는 다른 인스턴스(보조 인스턴스)에 연결합니다. 자세한 내용은 [Amazon EBS 볼륨을 인스턴스에 연결 \(p. 576\)](#)을 참조하십시오. EBS 볼륨이 암호화된 경우, Amazon EBS 암호화를 지원하는 보조 인스턴스를 사용해야 합니다. 다른 방식으로는 이 절차에 대해 `t2.micro` 인스턴스를 사용할 수 있습니다. 자세한 내용은 [지원되는 인스턴스 유형 \(p. 619\)](#)을 참조하십시오. 보조 인스턴스를 가지고 있지 않은 경우는 보조 인스턴스를 실행할 필요가 있습니다. 자세한 내용은 [인스턴스 시작하기 \(p. 265\)](#) 섹션을 참조하십시오.

Important

보조 인스턴스는 볼륨을 연결할 때 실행 중이어야 하며, 복수의 루트 볼륨을 연결하는 중에 보조 인스턴스를 재부팅해서는 안 됩니다. 복수의 루트 볼륨을 가진 인스턴스를 부팅할 경우, 인스턴스가 잘못된 볼륨으로 부팅될 수 있습니다.

7. SSH를 가진 보조 인스턴스에 로그인합니다. 자세한 내용은 [Linux 인스턴스에 연결 \(p. 274\)](#)을 참조하십시오. 다음 절차로 진행합니다.

parted 명령을 사용한 Linux 파티션 확장

parted 유틸리티는 대부분 Linux 배포에서 제공되는 파티션 편집 도구입니다. 이를 통해 MBR 파티션 테이블 및 GPT 파티션 테이블을 모두 생성하고 편집할 수 있습니다. parted 유틸리티의 일부 버전(버전 2.1 이상)은 GPT 파티션 테이블을 제한적으로만 지원하며 부팅 문제를 일으킬 수 있으며, 해당 버전의 parted 유틸리티를 사용해서 부트 볼륨을 변경할 경우 부팅 문제가 발생할 수 있습니다. parted 유틸리티의 버전은 `parted --version` 명령을 통해 확인할 수 있습니다.

GPT 파티션 처리된 장치에 존재하는 파티션을 확장하는 경우는 상기 유틸리티 대신에 gdisk 유틸리티를 사용해야 합니다. 볼륨이 사용하는 디스크 라벨 유형이 확실하지 않은 경우는 `sudo fdisk -l` 명령을 사용해서 이

를 확인할 수 있습니다. 자세한 내용은 [gdisk 명령을 사용한 Linux 파티션 확장 방법 \(p. 603\)](#) 섹션을 참조하십시오.

parted 명령을 사용한 Linux 파티션 확장 방법

확장할 파티션이 루트 파티션인 경우는 [확장을 위한 Linux 루트 파티션 준비 방법 \(p. 599\)](#)의 절차를 먼저 수행합니다.

1. 확장할 파티션을 포함하는 장치를 확인합니다. lsblk 명령을 사용해서 인스턴스에 연결된 모든 디바이스 및 파티션을 표시합니다.

```
[ec2-user ~]$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvdf    202:80   0 100G  0 disk
##xvdf1 202:81   0     8G  0 part /mnt
xvda1   202:1    0   30G  0 disk /

```

이 예에서 xvdf 디바이스는 100GB의 사용 가능한 스토리지를 가지고 있으며 8GB 파티션을 포함하고 있습니다.

2. 마운트된 경우 파티션의 마운트를 해제합니다. lsblk 명령에서 확인한 MOUNTPOINT의 값을 통해 umount 명령을 실행합니다. 이 예에서 파티션에 대한 MOUNTPOINT 값은 /mnt입니다.

```
[ec2-user ~]$ sudo umount /mnt
```

3. 사용자 볼륨의 스냅샷을 생성합니다(이전 절차에서 스냅샷을 생성하지 않은 경우). 이후 진행 절차에서 데이터가 손상되거나 손실될 가능성이 높습니다. 따라서 원본의 스냅샷을 가지고 있으면 오류가 발생한 경우 언제나 다시 시작하고 데이터를 안전하게 유지할 수 있습니다. 자세한 내용은 [Amazon EBS 스냅샷 생성 \(p. 608\)](#) 섹션을 참조하십시오.
4. 디바이스에서 parted 명령을 실행합니다(디바이스의 파티션에서는 실행하지 않음). 이 경우 /dev/ 접두사를 lsblk 명령이 출력하는 이름에 추가하십시오.

```
[ec2-user ~]$ sudo parted /dev/xvdf
GNU Parted 2.1
Using /dev/xvdf
Welcome to GNU Parted! Type 'help' to view a list of commands.
```

5. 섹터에 대한 parted 측정 단위를 변경합니다.

```
(parted) unit s
```

6. print 명령을 실행해서 디바이스의 파티션을 표시합니다. 특정 파티션 테이블 유형의 경우, 보다 큰 볼륨 크기에 맞도록 파티션 테이블을 고칠 것을 요청받을 수도 있습니다. 기존 파티션 테이블을 고치는 것에 대한 모든 질문에 [Ignore] 답변을 합니다. 새 테이블은 이후에 생성하게 됩니다.

```
(parted) print
```

- a. 다음 메시지를 받은 경우, [Ignore]를 입력해서 백업 GPT 위치가 변하지 않도록 합니다.

```
Error: The backup GPT table is not at the end of the disk, as it should be. This
      might mean that another operating
      system believes the disk is smaller. Fix, by moving the backup to the end (and
      removing the old backup)?
Fix/Ignore/Cancel? Ignore
```

- b. 다음 메시지를 받은 경우, [Ignore]를 다시 입력해서 드라이브의 공간을 동일하게 유지합니다.

```
Warning: Not all of the space available to /dev/xvdf appears to be used, you can
fix the GPT to use all of the
space (an extra 46137344 blocks) or continue with the current setting?
Fix/Ignore? Ignore
```

7. 디스크의 총 크기, 파티션 테이블 유형, 파티션 수, 파티션의 시작점, boot 등 플래그에 대한 출력을 검사합니다. gpt 파티션 테이블의 경우, 파티션의 이름을 기록합니다. msdos 파티션 테이블의 경우, Type 필드(primary 또는 extended)를 기록합니다. 이 값들은 나중 단계에서 사용됩니다.

다음은 gpt 파티션 테이블 예입니다.

```
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdf: 209715200s
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start   End     Size      File system  Name           Flags
128      2048s  4095s   2048s          BIOS Boot Partition bios_grub
1       4096s 16777182s 16773087s  ext4          Linux
```

다음은 msdos 파티션 테이블 예입니다.

```
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdd: 104857600s
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number  Start   End     Size      Type      File system  Flags
1      2048s 35649535s 35647488s primary  ext3
```

8. 이전 단계에서 숫자 1을 사용하는 파티션에 대한 파티션 항목을 삭제합니다.

```
(parted) rm 1
```

9. 볼륨의 마지막까지 확장되는 새 파티션을 생성합니다.

(gpt 파티션 테이블 예) 상기 파티션 1의 시작점과 이름을 기록합니다. gpt 예의 경우, 시작점은 4096s, 이름은 Linux입니다. 파티션 1의 시작점, 이름, 100%을 포함하여 mkpart 명령을 실행해서 모든 가용 공간을 사용하도록 설정합니다.

```
(parted) mkpart Linux 4096s 100%
```

(msdos 파티션 테이블 예) 상기 파티션 1의 시작점과 파티션 유형을 기록합니다. msdos 예의 경우, 시작점은 2048s, 파티션 유형은 primary입니다. 기본 파티션 유형, 파티션 1의 시작점, 100%를 포함하여 mkpart 명령을 실행해서 모든 가용 공간을 사용하도록 설정합니다.

```
(parted) mkpart primary 2048s 100%
```

10. print 명령을 다시 실행해서 파티션을 확인합니다.

(gpt 파티션 테이블 예)

```
(parted) print
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdf: 209715200s
Sector size (logical/physical): 512B/512B
Partition Table: gpt
```

Number	Start	End	Size	File system	Name	Flags
					BIOS Boot Partition	bios_grub
1	4096s	209713151s	209709056s	ext4		Linux

(msdos 파티션 테이블 예)

```
(parted) print
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdf: 104857600s
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number Start End Size Type File system Flags
1 2048s 104857599s 104855552s primary ext3
```

- 이전에 존재했던 플래그가 확장했던 파티션에 대해서도 존재하는지를 확인합니다. 일부 경우 boot 플래그가 손실될 수 있습니다. 확장할 때 플래그가 누락된 경우, 사용자의 파티션 번호 및 플래그 이름으로 해당 부분을 대체하여 다음 명령을 사용해서 플래그를 추가합니다. 예를 들어 다음 명령은 boot 플래그를 파티션 1에 추가합니다.

```
(parted) set 1 boot on
```

print 명령을 다시 실행해서 변경사항을 확인할 수 있습니다.

- quit 명령을 실행해서 parted 유ти리티를 종료합니다.

```
(parted) quit
```

Note

파티션을 제거하고 새로 파티션을 추가했기 때문에 parted 유ти리티는 /etc/fstab를 업데이트 할 필요가 있다고 경고할 수도 있습니다. 이는 파티션 번호가 바뀐 경우만 필요한 작업입니다.

- 파일 시스템을 검사해서 오류가 없는지 확인합니다(파일 시스템 확장 전에 필요한 작업). 이전의 print 명령에서 확인한 파일 시스템 유형을 확인합니다. 파일 시스템 유형에 따라 아래 명령 중 하나를 선택합니다. 여기에 없는 파일 시스템의 경우, 해당 파일 시스템의 문서를 참조해서 올바른 검사(check) 명령을 확인합니다.

(ext3 또는 ext4 파일 시스템)

```
[ec2-user ~]$ sudo e2fsck -f /dev/xvdf1
e2fsck 1.42.3 (14-May-2012)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/: 31568/524288 files (0.4% non-contiguous), 266685/2096635 blocks
```

(xfs 파일 시스템)

```
[ec2-user ~]$ sudo xfs_repair /dev/xvdf1
Phase 1 - find and verify superblock...
Phase 2 - using internal log
    - zero log...
    - scan filesystem freespace and inode maps...
    - found root inode chunk
Phase 3 - for each AG...
    - scan and clear agi unlinked lists...
    - process known inodes and perform inode discovery...
```

```
- agno = 0
- agno = 1
- agno = 2
- agno = 3
- process newly discovered inodes...
Phase 4 - check for duplicate blocks...
- setting up duplicate extent list...
- check for inodes claiming duplicate blocks...
- agno = 0
- agno = 1
- agno = 2
- agno = 3
Phase 5 - rebuild AG headers and trees...
- reset superblock...
Phase 6 - check inode connectivity...
- resetting contents of realtime bitmap and summary inodes
- traversing filesystem ...
- traversal finished ...
- moving disconnected inodes to lost+found ...
Phase 7 - verify and correct link counts...
done
```

14. 확장된 파티션이 현재 인스턴스에 속하는지 또는 다른 인스턴스의 루트 파티션인지에 따라 다음 단계는 달라집니다.

- 이 파티션이 현재 인스턴스에 속하는 경우, 파티션을 Step 2 (p. 600)에서 식별된 MOUNTPOINT에 다시 마운트 합니다.

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

파티션을 마운트한 후에 볼륨 크기 조정 후 Linux 파일 시스템 확장 (p. 594)의 절차를 수행하여 파일 시스템을 확장해서 새로운 공간을 사용할 수 있도록 설정합니다.

- 이 볼륨이 다른 인스턴스의 루트 파티션인 경우, 확장한 파티션을 원래 인스턴스로 반환 (p. 607) 내 절차로 진행합니다.

gdisk 명령을 사용한 Linux 파티션 확장

gdisk 유ти리티(종종 GPT fdisk라고 불림)는 파티션 테이블 생성 및 편집에 대한 텍스트 기반의 메뉴 위주 방식의 툴이며, 일부 배포 버전에서는 parted 유ти리티보다 GPT 파티션 테이블을 보다 잘 지원합니다. 많은 공통 Linux 배포 버전(Amazon Linux 및 Ubuntu)은 gdisk를 기본적으로 제공합니다. 사용자의 배포 버전이 gdisk 명령을 제공하지 않는 경우, Obtaining GPT fdisk를 참조해서 이를 구하는 방법을 알아볼 수 있습니다. 많은 경우 gdisk 명령을 별다른 조치 없이 사용할 수 있기 때문에 보조 인스턴스로 사용할 Amazon Linux 인스턴스를 실행하는 것이 상당히 용이한 편입니다.

gdisk 명령을 사용한 Linux 파티션 확장 방법

확장할 파티션이 루트 파티션인 경우는 확장을 위한 Linux 루트 파티션 준비 방법 (p. 599)의 절차를 먼저 수행합니다.

- 확장할 파티션을 포함하는 장치를 확인합니다. lsblk 명령을 사용해서 인스턴스에 연결된 모든 디바이스 및 파티션을 표시합니다.

```
[ec2-user ~]$ lsblk
NAME      MAJ:MIN RM  SIZE RO MOUNTPOINT
xvdf      202:80   0 100G  0
##xvdf1   202:81   0  9.9G  0  /mnt
xvda1    202:1    0   30G  0  /
```

이 예에서 xvdf 디바이스는 100GB 가용 스토리지와 9.9GB 파티션을 가지고 있습니다.

2. 마운트된 경우 파티션의 마운트를 해제합니다. lsblk 명령에서 확인한 MOUNTPOINT의 값을 통해 umount 명령을 실행합니다. 이 예에서 파티션에 대한 MOUNTPOINT 값은 /mnt입니다.

```
[ec2-user ~]$ sudo umount /mnt
```

3. 사용자 볼륨의 스냅샷을 생성합니다(이전 절차에서 스냅샷을 생성하지 않은 경우). 이후 진행 절차에서는 데이터가 손상되거나 손실될 가능성이 높습니다. 따라서 원본의 스냅샷을 가지고 있으면 오류가 발생한 경우 언제나 다시 시작하고 데이터를 안전하게 유지할 수 있습니다. 자세한 내용은 [Amazon EBS 스냅샷 생성 \(p. 608\)](#) 섹션을 참조하십시오.
4. 디바이스에서 gdisk 명령을 실행합니다(디바이스의 파티션에서는 실행하지 않음). 이 경우 /dev/ 접두사를 lsblk 명령이 출력하는 이름에 추가하십시오.

```
[ec2-user ~]$ sudo gdisk /dev/xvdf
Gdisk /dev/xvdf
GPT fdisk (gdisk) version 0.8.10

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

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

5. p 명령을 실행해서 디바이스의 파티션 테이블을 표시합니다.
6. 디스크 식별자, 파티션 번호, 시작 섹터, 파티션 코드, 파티션 이름에 대한 출력을 검사합니다. 볼륨이 복수 파티션을 가진 경우 각각을 확인합니다.

```
Command (? for help): p
Disk /dev/xvdf: 209715200 sectors, 100.0 GiB
Logical sector size: 512 bytes
Disk identifier (GUID): 947F4655-F3BF-4A1F-8203-A7B30C2A4425
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 20705246
Partitions will be aligned on 2048-sector boundaries
Total free space is 2108 sectors (1.0 MiB)

Number  Start (sector)    End (sector)  Size            Code  Name
   1          2048        20705152   9.9 GiB         EF00  lxroot
```

위 예에서 디스크 식별자는 947F4655-F3BF-4A1F-8203-A7B30C2A4425, 파티션 번호는 1, 시작 섹터는 2048, 코드는 EF00, 이름은 lxroot입니다.

7. 기존 파티션 테이블은 이전에 크기가 작은 볼륨에 대해 생성한 것이기 때문에 크기가 큰 볼륨에 대해서는 새 파티션 테이블을 생성할 필요가 있습니다. o 명령을 실행해서 새로운 빈 파티션 테이블을 생성합니다.

```
Command (? for help): o
This option deletes all partitions and creates a new protective MBR.
Proceed? (Y/N): Y
```

8. n 명령을 사용해서 디바이스의 각 파티션에 대한 새 파티션 항목을 생성합니다.
 - 볼륨이 하나의 파티션만 있는 경우, 매 요청 시마다 기존에 기록한 값을 입력합니다. 마지막 섹터 값에는 전체 볼륨 크기로 확장할 수 있도록 기본 값을 사용합니다.

```
Command (? for help): n
Partition number (1-128, default 1): 1
First sector (34-209715166, default = 2048) or {+-}size{KMGTP}: 2048
Last sector (2048-209715166, default = 209715166) or {+-}size{KMGTP}: 209715166
```

```
Current type is 'Linux filesystem'  
Hex code or GUID (L to show codes, Enter = 8300): EF00  
Changed type of partition to 'EFI System'
```

- 볼륨이 둘 이상의 파티션을 가진 경우, BIOS 부트 파티션과 주요 데이터 파티션이 존재할 가능성이 높습니다. 이전에 기록한 값을 사용해서 BIOS 부트 파티션에 대한 새 파티션 항목을 생성합니다. 이전에 기록한 값을 사용하되 마지막 섹터 값은 전체 볼륨 크기로 확장할 수 있도록 기본 값을 사용해서, 주요 데이터 파티션에 대한 새 파티션 항목을 또 하나 생성합니다.

```
Command (? for help): n  
Partition number (1-128, default 1): 1  
First sector (34-209715166, default = 2048) or {+-}size{KMGTP}: 2048  
Last sector (2048-209715166, default = 209715166) or {+-}size{KMGTP}: 4095  
Current type is 'Linux filesystem'  
Hex code or GUID (L to show codes, Enter = 8300): EF02  
Changed type of partition to 'BIOS boot partition'  
  
Command (? for help): n  
Partition number (2-128, default 2): 2  
First sector (34-209715166, default = 4096) or {+-}size{KMGTP}: 4096  
Last sector (4096-209715166, default = 209715166) or {+-}size{KMGTP}: 209715166  
Current type is 'Linux filesystem'  
Hex code or GUID (L to show codes, Enter = 8300): 0700  
Changed type of partition to 'Microsoft basic data'
```

- c 명령을 사용해서 각 파티션의 이름을 이전 파티션의 이름으로 변경합니다. 파티션에 이름이 없는 경우는 Enter만 누릅니다.

```
Command (? for help): c  
Using 1  
Enter name: lxroot
```

- x 명령을 사용해서 전문가 명령 메뉴를 엽니다.
- g 명령을 사용해서 디스크 식별자를 원래 값으로 변경합니다.

```
Expert command (? for help): g  
Enter the disk's unique GUID ('R' to randomize): 947F4655-F3BF-4A1F-8203-A7B30C2A4425  
The new disk GUID is 947F4655-F3BF-4A1F-8203-A7B30C2A4425
```

- w 명령을 사용해서 변경사항을 디바이스에 저장하고 유ти리티를 종료합니다.

```
Expert command (? for help): w  
  
Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING  
PARTITIONS!!  
  
Do you want to proceed? (Y/N): Y  
OK; writing new GUID partition table (GPT) to /dev/xvdf.  
The operation has completed successfully.
```

- 파일 시스템을 검사해서 오류가 없는지 확인합니다(파일 시스템 확장 전에 필요한 작업).
 - 확장했던 파티션으로 해당 부분을 대체하여 다음 명령을 사용해서 파일 시스템 유형을 찾습니다(볼륨이 복수 파티션을 가진 경우 `/dev/xvdf2`일 수 있음).

```
[ec2-user ~]$ sudo file -sL /dev/xvdf1
```

- 파일 시스템 유형에 따라 아래 명령 중 하나를 선택합니다. 여기에 없는 파일 시스템의 경우, 해당 파일 시스템의 문서를 참조해서 올바른 검사(check) 명령을 확인합니다.

(ext3 또는 ext4 파일 시스템)

```
[ec2-user ~]$ sudo e2fsck -f /dev/xvdf1
e2fsck 1.42.3 (14-May-2012)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/: 31568/524288 files (0.4% non-contiguous), 266685/2096635 blocks
```

(xfs 파일 시스템)

Note

XFS 파일 시스템을 사용하려면 `xfsprogs` 패키지를 설치할 필요가 있습니다. 다음 명령을 사용해서 XFS 지원을 Amazon Linux 인스턴스에 추가합니다.

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

```
[ec2-user ~]$ sudo xfs_repair /dev/xvdf1
Phase 1 - find and verify superblock...
Phase 2 - using internal log
    - zero log...
    - scan filesystem freespace and inode maps...
    - found root inode chunk
Phase 3 - for each AG...
    - scan and clear agi unlinked lists...
    - process known inodes and perform inode discovery...
    - agno = 0
    - agno = 1
    - agno = 2
    - agno = 3
    - process newly discovered inodes...
Phase 4 - check for duplicate blocks...
    - setting up duplicate extent list...
    - check for inodes claiming duplicate blocks...
    - agno = 0
    - agno = 1
    - agno = 2
    - agno = 3
Phase 5 - rebuild AG headers and trees...
    - reset superblock...
Phase 6 - check inode connectivity...
    - resetting contents of realtime bitmap and summary inodes
    - traversing filesystem ...
    - traversal finished ...
    - moving disconnected inodes to lost+found ...
Phase 7 - verify and correct link counts...
done
```

14. 확장된 파티션이 현재 인스턴스에 속하는지 또는 다른 인스턴스의 루트 파티션인지에 따라 다음 단계는 달라집니다.

- 이 파티션이 현재 인스턴스에 속하는 경우, 파티션을 Step 2 (p. 604)에서 식별된 MOUNTPOINT에 다시 마운트 합니다.

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

파티션을 마운트한 후에 볼륨 크기 조정 후 Linux 파일 시스템 확장 (p. 594)의 절차를 수행하여 파일 시스템을 확장해서 새로운 가용 공간을 사용할 수 있도록 설정합니다.

- 이 볼륨이 다른 인스턴스의 루트 파티션인 경우, [확장한 파티션을 원래 인스턴스로 반환 \(p. 607\)](#) 내 절차로 진행합니다.

[확장한 파티션을 원래 인스턴스로 반환](#)

다른 인스턴스에서 루트 파티션을 확장한 경우는 이 절차를 수행해서 볼륨을 원래 인스턴스로 반환합니다.

확장한 루트 파티션을 원래 인스턴스로 반환하는 방법

1. 확장한 파티션을 보조 인스턴스에서 분리합니다. 자세한 내용은 [인스턴스에서 Amazon EBS 볼륨 분리 \(p. 588\)](#) 섹션을 참조하십시오.
2. [준비 절차 \(p. 599\)](#)의 [Step 4 \(p. 599\)](#)에서 식별한 디바이스 이름을 사용해서 볼륨을 기본 인스턴스에 다시 연결합니다. 자세한 내용은 [Amazon EBS 볼륨을 인스턴스에 연결 \(p. 576\)](#) 섹션을 참조하십시오.
3. 기본 인스턴스를 시작합니다. 자세한 내용은 [인스턴스 중지 및 시작 \(p. 285\)](#) 섹션을 참조하십시오.
4. (옵션) 파티션 확장만을 위해 보조 인스턴스를 실행한 경우, 인스턴스를 종료해서 추후 발생하는 변경을 방지할 수 있습니다. 자세한 내용은 [인스턴스 종료 \(p. 291\)](#) 섹션을 참조하십시오.
5. [볼륨 크기 조정 후 Linux 파일 시스템 확장 \(p. 594\)](#)의 절차를 수행하여 기본 인스턴스에 연결하고 파일 시스템을 확장해서 새로운 가용 공간을 사용할 수 있도록 설정합니다.

파일 시스템 확장을 완료한 후에는 사용할 수 있는 인스턴스에서 AMI를 생성하여 원하는 파티션 크기로 새 인스턴스를 시작할 수 있습니다. 자세한 내용은 [Amazon 머신 이미지\(AMI\) \(p. 62\)](#) 섹션을 참조하십시오.

Amazon EBS 스냅샷

지정 시간 스냅샷을 만들어 Amazon S3에 EBS 볼륨의 데이터를 백업할 수 있습니다. 스냅샷은 충분식 백업이어서 마지막 스냅샷 이후 변경된 디바이스의 블록만이 저장됩니다. 그러면 스냅샷을 만드는데 필요한 시간이 최소화되어 스토리지 비용이 절약됩니다. 스냅샷을 삭제하면 해당 스냅샷에 고유한 데이터만 제거됩니다. 활성 스냅샷에는 새 EBS 볼륨에 데이터(스냅샷 저장 시점)를 복원하기 위해 필요한 모든 정보가 들어 있습니다.

목차

- [스냅샷 개요 \(p. 607\)](#)
- [Amazon EBS 스냅샷 생성 \(p. 608\)](#)
- [Amazon EBS 스냅샷 삭제 \(p. 609\)](#)
- [Amazon EBS 스냅샷 복사 \(p. 610\)](#)
- [Amazon EBS 스냅샷 정보 보기 \(p. 612\)](#)
- [Amazon EBS 스냅샷 공유 \(p. 612\)](#)

스냅샷 개요

EBS 볼륨을 생성할 때 기존 스냅샷을 기반으로 생성할 수 있습니다. 이러한 경우 새 볼륨은 해당 스냅샷을 생성하는 데 사용된 원본 볼륨과 정확히 일치합니다. 기존 스냅샷에서 볼륨을 생성하면 백그라운드에 느리게 로드되어 사용자가 즉시 해당 볼륨을 사용할 수 있습니다. 아직 로드되지 않은 데이터에 액세스하는 경우, 볼륨은 요청한 데이터를 Amazon S3에서 즉시 다운로드한 후 백그라운드에서 볼륨의 나머지 데이터 로드를 진행합니다. 자세한 내용은 [Amazon EBS 스냅샷 생성 \(p. 608\)](#) 섹션을 참조하십시오.

스냅샷의 액세스 권한을 수정하여 여러 AWS 계정 간에 스냅샷을 공유할 수 있습니다. 자체 스냅샷뿐 아니라 공유된 스냅샷의 복사본을 만들 수 있습니다. 자세한 내용은 [Amazon EBS 스냅샷 공유 \(p. 612\)](#) 섹션을 참조하십시오.

EBS 스냅샷은 EBS 암호화를 폭넓게 지원합니다.

- 암호화된 볼륨의 스냅샷은 자동으로 암호화됩니다.
- 암호화된 스냅샷에서 만든 볼륨은 자동으로 암호화됩니다.
- 자신이 소유한 암호화되지 않은 스냅샷을 복사할 때 복사 프로세스 중에 해당 스냅샷을 암호화할 수 있습니다.
- 자신이 소유한 암호화된 스냅샷을 복사할 때는 복사 프로세스 중에 다른 키로 해당 스냅샷을 다시 암호화 할 수 있습니다.

자세한 내용은 [Amazon EBS 암호화](#) 섹션을 참조하십시오.

스냅샷은 생성된 리전으로 제한됩니다. EBS 볼륨의 스냅샷을 생성한 후에는 해당 스냅샷을 사용하여 동일한 리전에 새로운 볼륨을 생성할 수 있습니다. 자세한 내용은 [스냅샷에서 Amazon EBS 볼륨 복구 \(p. 574\)](#) 섹션을 참조하십시오. 또한 리전 전반에 스냅샷을 복사하면 지리적 확장, 데이터 센터 마이그레이션 및 재해 복구를 위해 여러 리전을 쉽게 활용할 수 있습니다. 스냅샷의 복사 및 액세스는 스냅샷이 completed 상태인 경우 가능합니다. 자세한 내용은 [Amazon EBS 스냅샷 복사 \(p. 610\)](#) 섹션을 참조하십시오.

Amazon EBS 스냅샷 생성

EBS 볼륨에 데이터를 쓴 이후 새 볼륨 또는 데이터 백업의 기준으로 사용할 볼륨의 스냅샷을 주기적으로 생성할 수 있습니다. 볼륨의 스냅샷이 주기적으로 생성되는 경우 스냅샷은 종분식이어서 마지막 스냅샷 이후 변경된 디바이스의 블록만이 새 스냅샷에 저장됩니다. 스냅샷은 종분식으로 저장되지만 스냅샷 삭제 프로세스는 볼륨을 복구하기 위해 가장 최근의 스냅샷만을 유지할 수 있도록 설계됩니다.

스냅샷은 비동기적으로 생성됩니다. 특정 시점 스냅샷은 즉시 생성되지만 스냅샷이 완료될 때까지, 즉 수정된 블록이 Amazon S3로 모두 이동할 때까지 스냅샷 상태는 pending입니다. 따라서 크기가 큰 최초의 스냅샷이나 변경된 블록이 많은 후속 스냅샷의 경우 몇 시간씩 시간이 걸릴 수 있습니다. 완료하는 동안 진행 중인 스냅샷은 볼륨에 대한 지속적인 읽기 및 쓰기의 영향을 받지 않습니다.

Important

볼륨의 이전 스냅샷이 pending 상태일 때에도 볼륨의 스냅샷을 생성할 수는 있지만 볼륨의 pending 스냅샷을 여러 개 생성하면 스냅샷이 완료될 때까지 볼륨 성능이 저하될 수 있습니다.

단일 gp2, io1, 또는 Magnetic 볼륨에 대해 5개의 pending 스냅샷, 단일 st1 또는 sc1 볼륨에 대해 1 개의 pending 스냅샷으로 제한됩니다. 동일 볼륨에 대해 여러 개의 동일 스냅샷을 생성하려 할 때 ConcurrentSnapshotLimitExceeded 오류가 표시되면 한 개 이상의 pending 스냅샷이 완료될 때까지 기다린 후 해당 볼륨의 다른 스냅샷을 생성하십시오.

암호화된 볼륨으로 생성한 스냅샷은 자동으로 암호화됩니다. 암호화된 스냅샷으로 생성한 볼륨도 자동으로 암호화됩니다. 암호화된 볼륨 및 모든 연관 스냅샷의 데이터는 사용되지 않을 때와 사용될 때 모두 보호됩니다. 자세한 내용은 [Amazon EBS 암호화](#) 섹션을 참조하십시오.

기본적으로 사용자는 자기 소유의 스냅샷에서만 볼륨을 생성할 수 있습니다. 하지만 암호화되지 않은 스냅샷은 특정 AWS 계정과 공유하거나 공개 상태로 지정하여 전체 AWS 커뮤니티에서 공유할 수 있습니다. 자세한 내용은 [Amazon EBS 스냅샷 공유 \(p. 612\)](#) 섹션을 참조하십시오.

암호화된 스냅샷은 특정 AWS 계정과만 공유할 수 있습니다. 다른 계정의 사용자가 암호화된 공유 스냅샷을 사용할 수 있도록 하려면 해당 스냅샷을 암호화할 때 사용했던 CMK 키도 공유해야 합니다. 암호화된 스냅샷에 대한 액세스 권한이 있는 사용자는 개인적으로 자체 복사본을 만들고 이를 사용해 볼륨을 복원해야 합니다. 암호화된 공유 스냅샷의 복사본을 다른 키로 다시 암호화할 수도 있습니다. 자세한 내용은 [Amazon EBS 스냅샷 공유 \(p. 612\)](#) 섹션을 참조하십시오.

AWS Marketplace 제품 코드를 가진 볼륨에서 스냅샷을 만들면 해당 제품 코드가 스냅샷으로 전파됩니다.

연결되어 사용 중인 볼륨의 스냅샷을 만들 수 있습니다. 하지만 스냅샷은 snapshot 명령을 실행할 때 Amazon EBS 볼륨에 기록된 데이터만 캡처합니다. 이때 애플리케이션이나 운영 체제에 의해 캐시된 데이터가 제외될 수 있습니다. 스냅샷을 만들기에 충분한 시간 동안 볼륨에 대한 파일 쓰기 작업을 일시 중지할 수 있는 경우 스냅샷이 완전해야 합니다. 하지만 볼륨에 대한 모든 파일 쓰기를 일시 중지할 수는 없는 경우에는 인스턴스 내에서 볼륨을 분리하고 snapshot 명령을 실행한 다음, 볼륨을 다시 마운트하여 일관되고 완전한

스냅샷이 되도록 해야 합니다. 스냅샷 상태가 pending인 상태에서 볼륨을 다시 마운트하고 사용할 수 있습니다.

루트 디바이스 역할을 하는 Amazon EBS 볼륨의 스냅샷을 만들려면 인스턴스를 중지한 후 스냅샷을 만들어야 합니다.

Linux에서 볼륨을 분리하려면 다음 명령을 사용합니다.

```
umount -d device_name
```

여기서 `device_name`은 디바이스 이름입니다(예: `/dev/sdh`).

스냅샷을 생성한 후에는 나중에 쉽게 관리할 수 있도록 스냅샷에 태그를 지정할 수 있습니다. 예를 들어, 스냅샷이 생성된 원래 볼륨 또는 원래 볼륨을 인스턴스에 연결하는 데 사용된 디바이스 이름을 설명하는 태그를 추가할 수 있습니다. 자세한 내용은 [Amazon EC2 리소스에 태그 지정 \(p. 681\)](#) 섹션을 참조하십시오.

콘솔을 이용하여 스냅샷을 생성하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Snapshots]를 선택합니다.
3. Create Snapshot을 클릭합니다.
4. [Create Snapshot] 대화 상자에서 스냅샷을 생성할 볼륨과 [Create]를 차례대로 선택합니다.

명령줄을 이용하여 스냅샷을 생성하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- `create-snapshot` (AWS CLI)
- `New-EC2Snapshot` (Windows PowerShell용 AWS 도구)

Amazon EBS 스냅샷 삭제

스냅샷을 삭제하면 해당 스냅샷과 연관이 있는 데이터만 제거됩니다. 볼륨의 이전 스냅샷을 삭제해도 해당 볼륨의 이후 스냅샷으로 볼륨을 복원하는 기능에는 영향을 주지 않습니다.

볼륨의 스냅샷이 주기적으로 생성되는 경우 스냅샷은 중분식이어서 마지막 스냅샷 이후 변경된 디바이스의 블록만이 새 스냅샷에 저장됩니다. 스냅샷은 중분식으로 저장되지만 스냅샷 삭제 프로세스는 볼륨을 복구하기 위해 가장 최근의 스냅샷만을 유지할 수 있도록 설계됩니다.

등록된 AMI에서 사용된 EBS 볼륨의 루트 디바이스에 대한 스냅샷을 삭제할 수 없음에 유의하십시오. 스냅샷을 삭제하기 전 우선 AMI를 등록해야 합니다. 자세한 내용은 [AMI 등록 취소 \(p. 130\)](#) 섹션을 참조하십시오.

콘솔을 이용하여 스냅샷을 삭제하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Snapshots]를 선택합니다.
3. 스냅샷을 선택한 다음 [Actions] 목록에서 [Delete]를 선택합니다.
4. [Yes, Delete]를 선택합니다.

명령줄을 이용하여 스냅샷을 삭제하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [delete-snapshot](#) (AWS CLI)
- [Remove-EC2Snapshot](#) (Windows PowerShell용 AWS 도구)

Amazon EBS 스냅샷 복사

Amazon EBS를 사용하면 볼륨의 특정 시점 스냅샷을 생성할 수 있으며 이 스냅샷은 Amazon Simple Storage Service(Amazon S3)에 저장됩니다. 스냅샷을 생성하여 Amazon S3에 복사까지 마쳤으면(스냅샷 상태 completed) 다른 AWS 리전으로, 혹은 동일한 리전 내에서 복사할 수 있습니다. Amazon S3 서버 쪽 암호화(256비트 AES)는 복사 과정에서 전송 중인 스냅샷 데이터를 보호합니다. 스냅샷 복사본은 원본 스냅샷의 ID와 다른 스냅샷 ID를 받습니다.

Note

Amazon Relational Database Service (Amazon RDS) 스냅샷을 복사하려면 Amazon Relational Database Service 사용 설명서의 [DB 스냅샷 복사](#) 섹션을 참조하십시오.

다음 방법으로 스냅샷 사본을 사용할 수 있습니다.

- **지리적 확장:** 새 리전에서 애플리케이션 시작.
- **マイグレーション:** 새 리전으로 애플리케이션을 마이그레이션하여 가용성을 향상하고 비용을 최소화.
- **재해 복구:** 서로 다른 지리적 위치에 있는 데이터와 로그를 정기적인 시간 간격으로 백업. 재난 복구의 경우 보조 리전에 저장된 특정 시점 백업을 사용하여 애플리케이션을 복구할 수 있습니다. 이를 통해 데이터 손실 및 복구 시간이 최소화됩니다.
- **암호화:** 이전에 암호화되지 않은 스냅샷을 암호화하고 스냅샷 암호화 시 사용한 키를 변경하거나, 자신과 공유된 암호화된 스냅샷의 경우에는 자기 소유의 복사본을 따로 만들어 그 복사본으로부터 볼륨을 복원합니다.
- **데이터 보존 및 감사 요구 사항:** 한 AWS 계정에서 다른 AWS 계정으로 암호화된 EBS 스냅샷을 복사하여 데이터 로그나 감사 또는 데이터 보존을 위한 다른 파일을 보존합니다. 다른 계정을 사용하면 실수로 스냅샷을 삭제하는 것을 방지하고 기본 AWS 계정에 문제가 생길 경우 보호하는데 도움이 됩니다.

Note

CopySnapshot 작업을 통해 생성된 스냅샷에는 어떠한 용도에도 사용되지 않는 임의 볼륨 ID가 있습니다.

사용자 정의 태그는 원본 스냅샷에서 새로운 스냅샷으로 복사되지 않습니다. 하지만 복사 작업이 완료되면 사용자 정의 태그를 새로운 스냅샷에 적용할 수는 있습니다. 자세한 내용은 [Amazon EC2 리소스에 태그 지정 \(p. 681\)](#) 섹션을 참조하십시오.

단일 목적지에서 계정 당 최대 5개의 스냅샷 사본 요청이 진행될 수 있습니다. 공유 스냅샷 및 사용자가 생성한 스냅샷 등 completed 상태인 액세스 가능 스냅샷을 복사할 수 있습니다. 또한, AWS 마켓플레이스, VM 가져오기/내보내기 및 AWS Storage Gateway 스냅샷을 복사할 수 있지만 목적지 리전에서 해당 스냅샷이 지원되는지 확인해야 합니다.

다른 리전에 대한 볼륨의 최초 스냅샷은 항상 전체 사본입니다. 그 이후의 각 스냅샷은 종분식이어서(복사하는 데 시간이 적게 걸림) 동일한 대상 위치로의 마지막 스냅샷 사본 이후 변경된 볼륨만이 전달됩니다. 종분식 스냅샷은 이전에 완료된 소스 볼륨 스냅샷을 대상 리전에서 이미 사용할 수 있는 리전 페어에서만 지원되며, 암호화된 스냅샷의 경우 기본 EBS CMK로 제한됩니다. 예를 들어 미국 동부(버지니아 북부) 리전에서 미국 서부(오레곤) 리전으로 암호화되지 않은 스냅샷을 복사할 경우 첫 번째 볼륨 스냅샷 복사는 전체 복사이고, 이후부터 같은 리전 사이에서 전송되는 동일한 볼륨 스냅샷 복사는 종분식입니다.

Note

단일 계정 및 리전 내에서의 스냅샷 복사는 다음 조건이 적용되는 한 실제로 데이터가 복사되지 않기 때문에 아무런 비용도 발생하지 않습니다.

- 스냅샷 복사본의 암호화 상태가 복사 작업 중 변경되지 않을 경우.

- 암호화된 스냅샷일 때, 원본 스냅샷과 복사본이 모두 기본 EBS CMK로 암호화된 경우.

다른 계정에서도 스냅샷을 복사할 수 있도록 하고 싶다면 해당 계정에 액세스할 수 있도록 스냅샷 권한을 변경하거나 모든 AWS 계정이 복사할 수 있도록 스냅샷을 퍼블릭으로 설정해야 합니다. 자세한 내용은 [Amazon EBS 스냅샷 공유 \(p. 612\)](#) 섹션을 참조하십시오.

리전 및 계정 간의 스냅샷 복사와 관련된 요금 정보는 [Amazon EBS 요금](#)을 참조하십시오.

암호화된 스냅샷

스냅샷을 복사할 때는 복사본을 암호화하거나(원본 스냅샷이 암호화되지 않은 경우), 혹은 원본과 다른 CMK(고객 마스터 키)를 지정할 수 있습니다. 그러면 복사된 스냅샷은 새로운 CMK를 사용합니다. 하지만 복사 작업 중 스냅샷의 암호화 상태를 변경하거나 기본이 아닌 EBS CMK를 사용하면 증분식이 아닌 전체 복사본이 생성되어 많은 양의 데이터가 전송되고 스토리지 요금이 많이 발생할 수 있습니다.

다른 계정에서 암호화된 스냅샷을 복사하려면 스냅샷 사용 권한을 비롯해 원본 스냅샷의 암호화에 사용된 고객 마스터 키(CMK) 사용 권한도 필요합니다. 자세한 내용은 [Amazon EBS 스냅샷 공유 \(p. 612\)](#) 섹션을 참조하십시오.

Note

자신과 공유된 암호화된 스냅샷을 복사할 때, 복사 프로세스 중에 자신이 관리하는 다른 키를 이용한 스냅샷 재암호화를 고려해야 합니다. 이를 통해 원래 키가 손상되거나 소유자가 어떤 이유로든 키를 취소하는 바람에 자신이 만든 볼륨에 액세스할 수 없게 되는 상황을 방지할 수 있습니다.

Amazon EC2 콘솔을 이용하여 스냅샷을 복사하려면

- <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
- 탐색 창에서 [Snapshots]를 선택합니다.
- 복사할 스냅샷을 선택한 다음 [Actions] 목록에서 [Copy]를 선택합니다.
- [Copy Snapshot] 대화 상자에서 필요한 경우 다음을 업데이트합니다.
 - [Destination region]: 스냅샷 사본을 작성할 리전 선택.
 - [Description]: 기본적으로, 설명에는 소스 스냅샷에 대한 정보가 포함되어 사용자는 원본과 사본을 구분할 수 있습니다. 필요한 경우 이 설명을 수정할 수 있습니다.
 - [Encryption]: 원본 스냅샷이 암호화되지 않은 경우에는 사본을 암호화할 수 있습니다. 암호화된 스냅샷은 해독할 수 없습니다.
 - Master Key: 이 스냅샷을 암호화하는 데 사용할 고객 마스터 키(CMK). 마스터 키는 자신의 계정에서 선택하거나 다른 계정의 키 ARN을 입력하거나 붙여 넣을 수 있습니다. 새로운 마스터 암호화 키는 IAM 콘솔에서 생성할 수 있습니다.
- [Copy]를 선택합니다.
- [Copy Snapshot] 확인 대화상자에서 [Snapshots]를 선택해 지정된 리전의 [Snapshots] 페이지로 이동하거나 [Close]를 선택합니다.

차후에 복사 프로세스 진행률을 확인하려면 목적지 리전으로 전환한 다음 [Snapshots] 페이지를 새로고침합니다. 복사 진행률이 페이지 상단에 표시됩니다.

오류를 확인하려면

암호화 키 사용 권한 없이 암호화된 스냅샷을 복사하려고 하면 작업이 자동으로 실패하게 됩니다. 페이지를 새로 고칠 때까지는 콘솔에 오류 상태가 표시되지 않습니다. 또한, 명령줄에서 스냅샷의 상태를 확인할 수 있습니다. 예:

```
$ aws ec2 describe-snapshots --snapshot-id snap-0123abcd
```

키 권한 부족으로 복사에 실패한 경우에는 다음 메시지가 표시됩니다.

"StateMessage": "Given key ID is not accessible"

Note

암호화된 스냅샷을 복사할 때는 기본 CMK에 대한 권한을 설정해야 합니다. 이러한 권한을 명시적으로 거부하면 복사에 실패하게 됩니다.

명령줄을 이용하여 스냅샷을 복사하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [copy-snapshot](#) (AWS CLI)
- [Copy-EC2Snapshot](#) (Windows PowerShell용 AWS 도구)

Amazon EBS 스냅샷 정보 보기

스냅샷에 대한 세부 정보를 볼 수 있습니다.

콘솔을 사용하여 스냅샷 정보를 보려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Snapshots]를 선택합니다.
3. 목록을 줄이려면 [Filter] 목록에서 옵션을 선택합니다. 예를 들어, 자신이 소유한 스냅샷만을 확인하려면 [Owned By Me]를 선택합니다. 고급 검색 옵션을 이용하여 스냅샷을 추가로 필터링할 수 있습니다. 검색 창을 선택하여 사용 가능한 필터를 확인합니다.
4. 스냅샷에 대한 자세한 정보를 확인하려면 선택합니다.

명령줄을 사용하여 스냅샷 정보를 보려면

다음 명령 중 하나를 사용할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [describe-snapshots](#) (AWS CLI)
- [Get-EC2Snapshot](#) (Windows PowerShell용 AWS 도구)

Amazon EBS 스냅샷 공유

암호화되지 않은 스냅샷은 스냅샷의 권한을 수정하여 동료 또는 AWS 커뮤니티의 다른 사용자와 공유할 수 있습니다. 허가된 사용자는 암호화되지 않은 공유 스냅샷을 기본으로 사용하여 자신의 EBS 볼륨을 빠르게 생성할 수 있습니다. 필요한 경우, 암호화되지 않은 스냅샷을 모든 AWS 사용자에게 공개할 수 있습니다.

암호화된 스냅샷을 퍼블릭 상태로 만들 수는 없지만 특정 AWS 계정과 공유할 수 있습니다. 다른 계정의 사용자가 이 스냅샷을 사용할 수 있도록 하려면 해당 스냅샷을 암호화할 때 사용한 사용자 지정 CMK 키도 공유해야 합니다. 사용자 지정 키가 생성될 때 또는 그 이후에 키에 교차 계정 권한이 적용될 수도 있습니다. 액세스 권한이 있는 사용자는 원본 스냅샷에는 아무런 영향도 주지 않고서 스냅샷을 복사하고 스냅샷을 바탕으로 자신의 EBS 볼륨을 만들 수 있습니다.

Important

스냅샷을 공유(다른 AWS 계정과 공유하거나 퍼블릭으로 설정하여 모든 사용자와 공유)하는 경우 다른 사용자는 스냅샷에 있는 모든 데이터에 액세스할 수 있습니다. 그러므로 스냅샷의 모든 데이터를 공유하려는 사용자하고만 스냅샷을 공유하십시오.

스냅샷 공유에 다음과 같은 여러 기술 및 정책 제한 사항이 적용됩니다.

- 스냅샷은 생성된 리전으로 제한됩니다. 다른 리전에서 스냅샷을 공유하려면 스냅샷을 해당 리전으로 복사해야 합니다. 스냅샷 복사에 대한 자세한 내용은 [Amazon EBS 스냅샷 복사 \(p. 610\)](#) 섹션을 참조하십시오.
- 스냅샷에서 더 긴 리소스 ID 형식을 사용하는 경우 더 긴 ID를 지원하는 다른 계정과 공유해야 합니다. 자세한 내용은 [리소스 ID](#)를 참조하십시오.
- AWS는 기본 CMK로 암호화된 스냅샷의 공유를 금지합니다. 그 대신, 사용자 지정 CMK로 공유하려는 스냅샷을 암호화해야 합니다. 키 생성에 대한 자세한 내용은 [키 생성](#) 섹션을 참조하십시오.
- 암호화된 스냅샷에 액세스하게 될 공유 CMK의 사용자들에게는 [DescribeKey](#) 및 [ReEncrypt](#) 권한이 부여되어야 합니다. CMK 키 관리 및 공유에 대한 자세한 내용은 [고객 마스터 키에 대한 액세스 제어](#)를 참조하십시오.
- 암호화된 공유 스냅샷에 대한 액세스 권한이 있고 이 스냅샷으로부터 볼륨을 복원하려는 경우 스냅샷의 개인 복사본을 만든 후 그 복사본을 사용하여 볼륨을 복원해야 합니다. 복사 프로세스 중에 자신이 관리하는 다른 키로 스냅샷을 다시 암호화하는 게 좋습니다. 이를 통해 원래 키가 손상되거나 소유자가 어떤 이유로든 키를 취소하는 경우 볼륨에 대한 액세스 권한을 보호할 수 있습니다.

콘솔을 이용하여 스냅샷 권한을 수정하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Snapshots]를 선택합니다.
3. 스냅샷을 선택한 다음 [Actions] 목록에서 [Modify Permissions]를 선택합니다.
4. 스냅샷을 퍼블릭으로 설정할지 아니면 특정 AWS 계정과 공유할 것인지를 선택합니다.
 - 스냅샷을 퍼블릭으로 설정하려면 [Public]를 선택합니다.
암호화된 스냅샷 또는 AWS Marketplace 제품 코드가 있는 스냅샷에는 이 옵션을 사용할 수 없습니다.
 - 특정 AWS 계정만 스냅샷에 액세스할 수 있도록 하려면 [Private]을 선택한 다음 [AWS Account Number] 필드에 AWS 계정(하이픈을 사용하지 않음)의 ID를 입력하고 [Add Permission]을 선택합니다. 모든 필요한 AWS 계정에 추가될 때까지 이 과정을 반복합니다.

Important

스냅샷이 암호화된 경우에는 다음 조건을 충족하는지 확인해야 합니다.

- 스냅샷은 기본 CMK가 아니라 사용자 지정 CMK로 암호화됩니다. 기본 CMK로 암호화된 스냅샷의 권한을 변경하려고 하면 콘솔에 오류 메시지가 표시될 것입니다.
 - 스냅샷에 대한 액세스 권한을 가진 계정들과 사용자 지정 CMK를 공유하고 있습니다.
5. [Save]를 선택합니다.

명령줄을 이용하여 스냅샷 권한을 확인 및 수정하려면

`createVolumePermission` 스냅샷의 속성을 확인하려면 다음 명령 중 하나를 사용합니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [describe-snapshot-attribute](#) (AWS CLI)
- [Get-EC2SnapshotAttribute](#) (Windows PowerShell용 AWS 도구)

`createVolumePermission` 스냅샷의 속성을 수정하려면 다음 명령 중 하나를 사용합니다.

- [modify-snapshot-attribute](#) (AWS CLI)
- [Edit-EC2SnapshotAttribute](#) (Windows PowerShell용 AWS 도구)

Amazon EBS 최적화 인스턴스

Amazon EBS 최적화 인스턴스는 최적화된 구성 스택을 사용하며 Amazon EBS(EBS) I/O를 위한 추가 전용 용량을 제공합니다. 이러한 최적화를 통해 인스턴스에서 Amazon EBS I/O와 기타 트래픽 간의 경합이 최소화되어 EBS 볼륨의 성능이 극대화됩니다.

EBS 최적화 인스턴스는 Amazon EBS에 전용 대역폭을 제공하며, 사용하는 인스턴스 유형에 따라 500Mbps와 12,000Mbps 사이의 범위에서 선택할 수 있습니다. EBS 최적화 인스턴스에 연결된 범용 SSD(gp2) 볼륨은 연중 99%의 시간 동안 기본 성능 및 버스트 성능을 10% 이내의 오차로 제공하도록 설계되며, 프로비저닝된 IOPS SSD(io1) 볼륨은 연중 99.9%의 시간 동안 프로비저닝 성능을 10% 이내의 오차로 제공하도록 설계됩니다. 처리량에 최적화된 HDD(st1) 및 Cold HDD(sc1)는 모두 연중 99%의 기간 동안 90%의 버스트 처리량에 성능 일관성을 보장합니다. 매 시간 총 처리량 목표 99%를 달성하기 위해, 준수하지 않는 기간은 대략적으로 균등하게 분산됩니다. 자세한 내용은 [Amazon EBS 볼륨 유형 \(p. 564\)](#) 섹션을 참조하십시오.

기본적으로 EBS 최적화되지 않은 인스턴스에 EBS 최적화를 사용하도록 설정할 경우 전용 용량을 위해 소정의 시간당 추가 요금이 청구됩니다. 요금 정보는 Amazon EC2 요금 페이지에서 [EBS 최적화 인스턴스](#) 섹션을 참조하십시오.

목차

- [EBS 최적화를 지원하는 인스턴스 유형 \(p. 614\)](#)
- [시작 시 EBS 최적화 활성화 \(p. 616\)](#)
- [실행 중인 인스턴스에 대해 EBS 최적화 수정 \(p. 617\)](#)

EBS 최적화를 지원하는 인스턴스 유형

다음 표는 EBS 최적화를 지원할 인스턴스 유형, Amazon EBS에 대한 전용 대역폭, 16KiB I/O 크기를 사용할 경우 인스턴스에서 지원할 수 있는 IOPS 최대량을 보여주며, 해당 연결에서 달성을 할 수 있는 일반적인 최대 집계 처리량(MB/s)을 스트리밍 읽기 워크로드 및 128KiB I/O 크기로 보여줍니다. 애플리케이션에 필요한 것 보다 많은 전용 EBS 처리량을 제공하는 EBS에 최적화된 인스턴스를 선택해야 합니다. 그렇게 하지 않으면 Amazon EBS와 Amazon EC2 간의 연결이 성능 병목 현상으로 변할 수 있습니다.

일부 인스턴스 유형은 기본적으로 EBS 최적화되어 있습니다. 기본적으로 EBS 최적화되는 인스턴스의 경우 EBS 최적화를 활성화할 필요가 없으며, CLI나 API를 사용하여 EBS 최적화를 비활성화해도 적용되지 않습니다. 인스턴스를 시작할 때 EBS 최적화를 지원하는 다른 인스턴스 유형에 대해 EBS 최적화를 활성화하거나, 인스턴스를 실행한 후에 EBS 최적화를 활성화할 수 있습니다.

인스턴스 유형	기본적으로 EBS 최적화됨	최대 대역폭 (Mbps)*	예상 처리량(Mb/s)**	최대 IOPS(16KB I/O 크기)**
c1.xlarge		1,000	125	8,000
c3.xlarge		500	62.5	4,000
c3.2xlarge		1,000	125	8,000
c3.4xlarge		2,000건	250	16,000
c4.large	예	500	62.5	4,000
c4.xlarge	예	750	93.75	6,000
c4.2xlarge	예	1,000	125	8,000
c4.4xlarge	예	2,000건	250	16,000
c4.8xlarge	예	4,000	500	32,000
d2.xlarge	예	750	93.75	6,000

Amazon Elastic Compute Cloud
Linux 인스턴스용 사용 설명서
EBS 최적화

인스턴스 유형	기본적으로 EBS 최적화됨	최대 대역폭 (Mbps)*	예상 처리량(Mb/s)**	최대 IOPS(16KB I/O 크기)**
d2.2xlarge	예	1,000	125	8,000
d2.4xlarge	예	2,000건	250	16,000
d2.8xlarge	예	4,000	500	32,000
g2.2xlarge		1,000	125	8,000
i2.xlarge		500	62.5	4,000
i2.2xlarge		1,000	125	8,000
i2.4xlarge		2,000건	250	16,000
i3.large	예	425	50	3000
i3.xlarge	예	850	100	6000
i3.2xlarge	예	1,700	200	12,000
i3.4xlarge	예	3,500	400	16,000
i3.8xlarge	예	7,000	850	32,500
i3.16xlarge	예	14,000	1,750	65,000
m1.large		500	62.5	4,000
m1.xlarge		1,000	125	8,000
m2.2xlarge		500	62.5	4,000
m2.4xlarge		1,000	125	8,000
m3.xlarge		500	62.5	4,000
m3.2xlarge		1,000	125	8,000
m4.large	예	450	56.25	3,600
m4.xlarge	예	750	93.75	6,000
m4.2xlarge	예	1,000	125	8,000
m4.4xlarge	예	2,000건	250	16,000
m4.10xlarge	예	4,000	500	32,000
m4.16xlarge	예	10,000개	1,250	65,000
p2.xlarge	예	750	93.75	6,000
p2.8xlarge	예	5,000	625	32,500
p2.16xlarge	예	10,000개	1,250	65,000
r3.xlarge		500	62.5	4,000
r3.2xlarge		1,000	125	8,000

인스턴스 유형	기본적으로 EBS 최적화됨	최대 대역폭 (Mbps)*	예상 처리량(Mb/s)**	최대 IOPS(16KB I/O 크기)**
r3.4xlarge		2,000건	250	16,000
r4.large	예	400	50	3,000
r4.xlarge	예	800	100	6,000
r4.2xlarge	예	1600	200	12,000
r4.4xlarge	예	3000	375	16,000
r4.8xlarge	예	6000	750	32,000
r4.16xlarge	예	12,000	1,500	65,000
x1.16xlarge	예	5,000	625	32,500
x1.32xlarge	예	10,000개	1,250	65,000

* 이 수준의 성능을 일관되게 유지하려면 이러한 인스턴스 유형을 EBS에 최적화된 상태로 시작해야 합니다.

** 이 값은 100% 읽기 전용 작업을 기준으로 반올림한 근사치이며, 기준 구성은 보조할 목적으로 제공됩니다. EBS에 최적화된 연결은 전이중 모드로서, 두 통신 경로를 모두 사용하는 경우 50/50 읽기/쓰기 작업으로 균형을 이루며 처리 속도와 IOPS가 향상됩니다. 네트워크, 파일 시스템 및 Amazon EBS encryption 오버헤드로 인해 사용 가능한 최대 처리량과 IOPS가 줄어드는 경우도 있습니다.

i2.8xlarge 및 r3.8xlarge와 같이 10기가비트 네트워크 인터페이스를 사용하는 일부 인터페이스는 EBS 최적화에 영향을 미치지 않으므로, 사용 가능한 전용 EBS 대역폭이 없어 여기에 나열되지 않습니다. 이러한 인스턴스에서 네트워크 트래픽과 Amazon EBS 트래픽이 동일한 10기가비트 네트워크 인터페이스에서 공유됩니다. c4.8xlarge 및 d2.8xlarge와 같은 일부 다른 10기가비트 네트워크 인스턴스는 네트워크 트래픽에만 사용되는 10기가비트 인터페이스 외에도 전용 EBS 대역폭을 제공합니다.

시작 시 EBS 최적화 활성화

인스턴스의 EBS 최적화 속성을 설정하여 인스턴스에 대해 EBS 최적화를 활성화할 수 있습니다.

콘솔을 사용하여 인스턴스 시작 시 EBS 최적화를 활성화하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. [Launch Instance]를 클릭합니다. 1단계: Choose an Amazon Machine Image(AMI)에서 AMI를 선택합니다.
3. [Step 2: Choose an Instance Type]에서 EBS 최적화를 지원하는 인스턴스 유형을 선택합니다.
4. 3단계: Configure Instance Details에서 필요한 필드 정보를 모두 입력하고 Launch as EBS-optimized instance를 선택합니다. 이전 단계에서 선택한 인스턴스 유형이 EBS 최적화를 지원하지 않을 경우 이 옵션이 제공되지 않습니다. 기본적으로 선택한 인스턴스 유형이 EBS 최적화를 지원할 경우에는 이 옵션이 선택되며 선택을 취소할 수 없습니다.
5. 지시에 따라 마법사를 완료하고 인스턴스를 시작합니다.

명령줄을 사용하여 인스턴스 시작 시 EBS 최적화를 활성화하려면

해당 명령과 함께 다음 옵션 중 하나를 사용할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- `--ebs-optimized` with `run-instances` (AWS CLI)
- `-EbsOptimized` with `New-EC2Instance` (Windows PowerShell용 AWS 도구)

실행 중인 인스턴스에 대해 EBS 최적화 설정

실행 중인 인스턴스에 대해 EBS 최적화 인스턴스 속성을 수정하여 EBS 최적화를 활성화하거나 비활성화할 수 있습니다.

콘솔을 사용하여 실행 중인 인스턴스에 대해 EBS 최적화를 활성화하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Instances]를 클릭하고 해당 인스턴스를 선택합니다.
3. [Actions]를 클릭하고 [Instance State]를 선택한 다음 [Stop]을 클릭합니다.

Warning

인스턴스를 중지하면 인스턴스 스토어 볼륨의 데이터가 삭제됩니다. 따라서 인스턴스 스토어 볼륨에 보존하려는 데이터가 있을 경우 스토리지에 백업하십시오.

4. 확인 대화 상자가 나타나면 [Yes, Stop]을 클릭합니다. 인스턴스가 중지하는 데 몇 분 정도 걸릴 수 있습니다.
5. 인스턴스를 선택된 상태에서 [Actions]를 클릭하고 [Instance Settings]를 선택한 다음에 [Change Shutdown Behavior]를 클릭합니다.
6. [Change Instance Type] 대화 상자에서 다음 중 하나를 수행합니다.
 - 해당 인스턴스의 인스턴스 유형이 기본적으로 EBS 최적화되었을 경우 [EBS-optimized]가 선택되고 이를 선택 취소할 수 없습니다. 해당 인스턴스에 대해 EBS 최적화가 이미 활성화되었으므로 [Cancel]을 클릭합니다.
 - 해당 인스턴스의 인스턴스 유형이 EBS 최적화를 지원할 경우 [EBS-optimized]를 선택한 후 [Apply]를 클릭합니다.
 - 해당 인스턴스의 인스턴스 유형이 EBS 최적화를 지원하지 않을 경우 [EBS-optimized]의 선택이 취소되고 이를 선택할 수 없습니다. EBS 최적화를 지원하는 [Instance Type]에서 인스턴스 유형을 선택하고 [EBS-optimized]를 선택한 후 [Apply]를 클릭합니다.
7. Actions를 클릭하고 Instance State를 선택한 다음 Start를 클릭합니다.

명령줄을 사용하여 실행 중인 인스턴스에 대해 EBS 최적화를 활성화하려면

해당 명령과 함께 다음 옵션 중 하나를 사용할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- `--ebs-optimized` with [modify-instance-attribute](#) (AWS CLI)
- `-EbsOptimized` with [Edit-EC2InstanceAttribute](#) (Windows PowerShell용 AWS 도구)

Amazon EBS Encryption

Amazon EBS 암호화는 EBS 볼륨에 대해 키 관리 인프라를 사용자가 직접 구축, 유지 및 보호할 필요가 없는 간편한 암호화 솔루션을 제공합니다. 암호화된 EBS 볼륨을 만들고 지원되는 인스턴스 유형에 이 볼륨을 연결하면 다음 유형의 데이터가 암호화됩니다.

- 볼륨 내부에 있는 데이터
- 볼륨과 인스턴스 사이에서 이동하는 모든 데이터
- 볼륨에서 생성된 모든 스냅샷

암호화는 EC2 인스턴스를 호스팅하는 서버에서 수행되므로 EC2 인스턴스에서 EBS 스토리지로 전송되는 데이터가 암호화됩니다.

Amazon EBS 암호화에서는 암호화된 볼륨을 생성하거나 이런 볼륨에서 스냅샷을 생성할 때 AWS Key Management Service(AWS KMS) 고객 마스터 키(CMK)를 사용합니다. 한 리전에서 암호화된 볼륨을 처음 생성할 때 사용자의 기본 CMK가 자동으로 생성됩니다. AWS KMS(를) 사용하여 별도로 생성한 CMK를 선택하는 경우를 제외하고 이 키가 Amazon EBS 암호화에 사용됩니다. CMK를 직접 생성하면 액세스 제어를 정의하기 위해 키를 생성, 교체, 비활성화하고 데이터를 보호하는 데 사용된 암호화 키를 감사하는 등 보다 폭넓은 작업이 가능합니다. 자세한 내용은 [AWS Key Management Service Developer Guide](#) 섹션을 참조하십시오.

이 기능은 모든 EBS 볼륨 유형(범용 SSD [gp2], 프로비저닝된 IOPS SSD [io1], 처리량에 최적화된 HDD [st1], Cold HDD [sc1], Magnetic [standard])에서 지원되고, 암호화된 볼륨과 암호화되지 않은 볼륨의 IOPS 기대 성능은 동일하며 지연 시간에 대한 영향은 미미합니다. 암호화된 볼륨에 액세스하는 방법은 암호화되지 않은 볼륨에 액세스하는 경우와 동일합니다. 암호화 및 암호 해독은 중단 없이 처리되므로 사용자, EC2 인스턴스 또는 애플리케이션에서 별도로 조치할 부분은 없습니다.

암호화된 볼륨으로 생성한 스냅샷은 자동으로 암호화됩니다. 암호화된 스냅샷으로 생성한 볼륨도 자동으로 암호화됩니다. 암호화된 볼륨의 퍼블릭 스냅샷은 지원되지 않지만, 다음 단계를 수행하면 암호화된 스냅샷을 특정 계정과 공유할 수 있습니다.

1. 볼륨을 암호화하려면 기본 CMK가 아니라 사용자 지정 CMK를 사용합니다.
2. 사용자 지정 CMK에 대해 특정 계정 액세스 권한을 제공합니다.
3. 스냅샷을 생성합니다.
4. 스냅샷에 대한 특정 계정 액세스 권한을 제공합니다.

자세한 내용은 [Amazon EBS 스냅샷 공유](#)를 참조하십시오.

Amazon EBS 암호화 기능은 특정 인스턴스 유형에만 사용할 수 있습니다. 지원되는 인스턴스 유형에는 암호화된 볼륨과 암호화되지 않은 볼륨을 모두 연결할 수 있습니다. 자세한 내용은 [지원되는 인스턴스 유형 \(p. 619\)](#) 섹션을 참조하십시오.

목차

- [암호화 키 관리 \(p. 618\)](#)
- [지원되는 인스턴스 유형 \(p. 619\)](#)
- [데이터의 암호화 상태 변경 \(p. 619\)](#)
- [Amazon EBS 암호화 및 CloudWatch 이벤트 \(p. 621\)](#)

암호화 키 관리

Amazon EBS 암호화에서 자동으로 키를 관리해 줍니다. 새로 생성되는 각 볼륨은 고유한 256비트 키로 암호화되고, 해당 볼륨의 모든 스냅샷 및 이러한 스냅샷으로 이후에 생성하는 모든 볼륨도 해당 키를 공유합니다. 이러한 키는 강력한 논리적, 물리적 보안 제어를 구현하여 무단 액세스를 방지하는 AWS 고유의 키 관리 인프라를 통해 보호됩니다. 데이터 및 이와 연결된 키는 산업 표준 AES-256 알고리즘을 사용하여 암호화됩니다.

기존 스냅샷이나 암호화된 볼륨과 연동되어 있는 CMK는 변경할 수 없습니다. 하지만 스냅샷 복사 작업(암호화되지 않은 스냅샷 사본을 암호화하는 작업 포함) 중에 다른 CMK와 연동 시킬 수는 있습니다. 그러면 복사된 스냅샷은 새로운 CMK를 사용하게 됩니다.

Amazon의 전체 키 관리 인프라에는 FIPS(Federal Information Processing Standards) 140-2의 승인을 획득한 암호화 알고리즘이 사용되며 NIST(National Institute of Standards and Technology) 800-57의 권장 지침을 준수합니다.

각 AWS 계정의 고유 마스터 키는 데이터와 완전히 분리된 별도의 시스템에 저장됩니다. 이 시스템에는 강력한 물리적, 논리적 보안 통제 수단이 적용됩니다. 암호화된 각 볼륨과 후속 스냅샷은 고유한 볼륨 암호화 키로 암호화되고, 이 키는 리전별 보안 마스터 키로 암호화됩니다. 볼륨 암호화 키는 EC2 인스턴스를 호스팅하는 서버의 메모리 상에서 사용되며 디스크에 일반 텍스트로 저장되지 않습니다.

지원되는 인스턴스 유형

아래 표에서는 Amazon EBS 암호화 기능을 사용할 수 있는 인스턴스 유형을 보여 줍니다. 이러한 인스턴스 유형은 인텔 AES-NI(AES New Instructions) 명령 세트를 활용하여 더욱 빠르고 간편하게 데이터를 보호합니다. 이러한 인스턴스 유형에는 암호화된 볼륨과 암호화되지 않은 볼륨을 동시에 연결할 수 있습니다.

인스턴스 패밀리	Amazon EBS 암호화 지원 인스턴스 유형
범용	m3.medium m3.large m3.xlarge m3.2xlarge m4.large m4.xlarge m4.2xlarge m4.4xlarge m4.10xlarge m4.16xlarge t2.nano t2.micro t2.small t2.medium t2.large t2.xlarge t2.2xlarge
컴퓨팅 최적화	c4.large c4.xlarge c4.2xlarge c4.4xlarge c4.8xlarge c3.large c3.xlarge c3.2xlarge c3.4xlarge c3.8xlarge
메모리 최적화	cr1.8xlarge r3.large r3.xlarge r3.2xlarge r3.4xlarge r3.8xlarge r4.large r4.xlarge r4.2xlarge r4.4xlarge r4.8xlarge r4.16xlarge x1.16xlarge x1.32xlarge
스토리지 최적화	d2.xlarge d2.2xlarge d2.4xlarge d2.8xlarge i2.xlarge i2.2xlarge i2.4xlarge i2.8xlarge i3.large i3.xlarge i3.2xlarge i3.4xlarge i3.8xlarge i3.16xlarge
액셀러레이티드 컴퓨팅	g2.2xlarge g2.8xlarge p2.xlarge p2.8xlarge p2.16xlarge

이러한 인스턴스 유형에 대한 자세한 내용은 [인스턴스 유형 세부 정보](#) 섹션을 참조하십시오.

데이터의 암호화 상태 변경

기존의 암호화되지 않은 볼륨을 암호화하거나 암호화된 볼륨에서 암호화를 제거하는 직접적인 방법은 없습니다. 그러나 암호화된 볼륨과 암호화되지 않은 볼륨 간에 데이터를 마이그레이션할 수는 있습니다. 스냅샷 복사 중에 새 암호화 상태를 적용할 수도 있습니다.

- 암호화되지 않은 볼륨의 암호화되지 않은 스냅샷을 복사하는 동안 복사본을 암호화할 수 있습니다. 이 암호화된 복사본에서 복원된 볼륨 역시 암호화됩니다.
- 암호화된 볼륨의 암호화된 스냅샷을 복사하는 동안 다른 CMK를 사용하여 복사본을 다시 암호화할 수 있습니다. 새로 적용된 CMK를 사용하여 암호화된 복사본에서 복원된 볼륨에만 액세스할 수 있습니다.

암호화된 볼륨과 암호화되지 않은 볼륨 간 데이터 마이그레이션

암호화된 볼륨과 암호화되지 않은 볼륨에 모두 액세스할 수 있는 경우, 둘 사이에서 자유롭게 데이터를 전송할 수 있습니다. EC2는 암호화 또는 복호화 작업을 투명하게 수행합니다.

암호화된 볼륨과 암호화되지 않은 볼륨 간에 데이터를 마이그레이션하려면 다음을 수행합니다.

- [Amazon EBS 볼륨 생성](#)(p. 573)의 절차에 따라 대상 볼륨(필요에 따라 암호화 또는 비암호화)을 생성합니다.
- 마이그레이션할 데이터를 호스팅하는 인스턴스에 대상 볼륨을 연결합니다. 자세한 내용은 [Amazon EBS 볼륨을 인스턴스에 연결](#)(p. 576) 섹션을 참조하십시오.
- [Amazon EBS 볼륨을 사용할 수 있도록 만들기](#)(p. 577)의 절차에 따라 대상 볼륨을 사용 가능하도록 만듭니다. Linux 인스턴스의 경우 /mnt/destination에 마운트 지점을 생성하고 해당 위치에 대상 볼륨을 마운트할 수 있습니다.
- 소스 디렉터리에서 대상 볼륨으로 데이터를 복사합니다. 이를 위해 대량 복사 유ти리티를 사용하는 것이 가장 편리할 수 있습니다.

Linux

다음과 같이 rsync 명령을 사용하여 소스에서 대상 볼륨으로 데이터를 복사합니다. 이 예제에서 소스 데이터는 /mnt/source에 있고 대상 볼륨은 /mnt/destination에 마운트되어 있습니다.

```
[ec2-user ~]$ sudo rsync -avh --progress /mnt/source/ /mnt/destination/
```

Windows

명령 프롬프트에서 robocopy 명령을 사용하여 원본 볼륨에서 대상 볼륨으로 데이터를 복사합니다. 이 예제에서 소스 데이터는 D:\에 있고 대상 볼륨은 E:\에 마운트되어 있습니다.

```
PS C:\Users\Administrator> robocopy D:\ E:\ /e /copyall /eta
```

스냅샷 복사 중 암호화 적용

스냅샷을 복사하는 동안 암호화를 적용할 수 있기 때문에, 데이터를 암호화하는 다른 경로는 다음 절차와 같습니다.

스냅샷 복사를 통해 볼륨의 데이터를 암호화하려면

- 암호화하지 않은 EBS 볼륨의 스냅샷을 생성합니다. 이 스냅샷도 암호화되지 않습니다.
- 암호화 파라미터를 적용하여 스냅샷을 복사합니다. 대상 스냅샷이 암호화됩니다.
- 암호화한 스냅샷을 마찬가지로 암호화한 새 볼륨으로 복원합니다.

자세한 내용은 [Amazon EBS 스냅샷 복사](#)를 참조하십시오.

새 CMK로 스냅샷 재암호화

복사 중에 스냅샷을 암호화하는 기능을 사용하여 자신이 소유하고 있는 이미 암호화된 스냅샷을 다시 암호화 할 수도 있습니다. 이 작업에서는 자신이 제공하는 새 CMK를 사용하여 스냅샷의 일반 텍스트가 암호화됩니다. 새 CMK를 사용하여 결과 복사본에서 복원된 볼륨에만 액세스할 수 있습니다.

관련된 시나리오에서 자신과 공유된 스냅샷을 다시 암호화하도록 선택할 수도 있습니다. 암호화된 공유 스냅샷에서 볼륨을 복원하려면, 먼저 그 볼륨의 자체 복사본을 만들어야 합니다. 기본적으로, 이 복사본은 스냅샷의 소유자가 공유하는 키로 암호화됩니다. 하지만 복사 프로세스 중에 자신이 관리하는 다른 키로 스냅샷을 다시 암호화하는 게 좋습니다. 이를 통해 원래 키가 손상되거나 소유자가 어떤 이유로든 키를 취소하는 경우 볼륨에 대한 액세스 권한을 보호할 수 있습니다.

다음 절차에서는 자신이 소유한 스냅샷을 다시 암호화하는 방법을 설명합니다.

콘솔을 이용하여 스냅샷을 다시 암호화하려면

- 사용자 지정 CMK를 생성합니다. 자세한 내용은 [AWS Key Management Service Developer Guide](#)를 참조하십시오.
- (이 예제의 경우) 기본 CMK로 암호화된 EBS 볼륨을 만듭니다.
- 암호화된 EBS 볼륨의 스냅샷을 생성합니다. 이 스냅샷은 기본 CMK로도 암호화됩니다.
- [Snapshots] 페이지에서 [Actions]와 [Copy]를 차례로 선택합니다.
- [Copy Snapshot] 창에서 [Master Key] 필드에 사용자 지정 CMK에 대한 완전한 ARN을 (`arn:aws:kms:us-east-1:012345678910:key/abcd1234-a123-456a-a12b-a123b4cd56ef` 형식으로) 입력하거나 메뉴에서 선택합니다. [Copy]를 클릭합니다.

스냅샷의 결과 복사본과 이 복사본에서 복원되는 모든 볼륨은 사용자 지정 CMK로 암호화됩니다.

다음 절차에서는 암호화된 공유 스냅샷을 복사할 때 이 스냅샷을 다시 암호화하는 방법을 설명합니다. 이렇게 하려면 암호화된 공유 스냅샷과 그 스냅샷을 암호화한 CMK에 대한 액세스 권한이 모두 필요합니다.

콘솔을 이용하여 공유 스냅샷을 복사하고 다시 암호화하려면

1. [Snapshots] 페이지에서 암호화된 공유 스냅샷을 선택하고 [Actions]와 [Copy]를 차례로 선택합니다.
2. [Copy Snapshot] 창에서 [Master Key] 필드에 자신이 소유한 CMK에 대한 완전한 ARN을 (`arn:aws:kms:us-east-1:012345678910:key/abcd1234-a123-456a-a12b-a123b4cd56ef` 형식으로) 입력하거나 메뉴에서 선택합니다. [Copy]를 클릭합니다.

스냅샷의 결과 복사본과 이 복사본에서 복원되는 모든 볼륨은 자신이 제공한 CMK로 암호화됩니다. 원본 공유 스냅샷, 암호화 상태 또는 공유 CMK에 대한 변경 사항은 복사본에 아무런 영향도 미치지 않을 것입니다.

자세한 내용은 [Amazon EBS 스냅샷 복사](#)를 참조하십시오.

Amazon EBS 암호화 및 CloudWatch 이벤트

EBS는 특정 암호화 관련 시나리오에 Amazon CloudWatch Events를 지원합니다. 자세한 내용은 [Amazon EBS용 Amazon CloudWatch Events](#) 섹션을 참조하십시오.

Linux 인스턴스의 Amazon EBS 볼륨 성능

I/O 특성, 인스턴스와 볼륨의 구성 등 여러 가지 요인이 Amazon EBS 볼륨에 영향을 끼칠 수 있습니다. Amazon EBS 및 Amazon EC2 제품 세부정보 페이지의 지침을 따르는 고객은 대체로 처음부터 우수한 성능을 달성할 수 있습니다. 그러나 경우에 따라 플랫폼에서 피크 성능을 얻기 위해서는 약간의 튜닝이 필요합니다. 이 주제에서는 일반적인 모범 사례 및 특정한 사용 사례에만 적용되는 성능 튜닝에 대해 설명합니다. 벤치마킹 외에도 실제 워크로드의 정보에 따라 성능을 튜닝하여 최적의 구성을 결정하는 것이 좋습니다. EBS 볼륨 작업의 기초를 배운 후에는 필요한 I/O 성능과 그러한 요건에 맞게 Amazon EBS 성능을 향상하기 위한 옵션을 살펴보는 것이 좋습니다.

목차

- [Amazon EBS 성능 팁 \(p. 621\)](#)
- [Amazon EC2 인스턴스 구성 \(p. 623\)](#)
- [I/O 특성 및 모니터링 \(p. 626\)](#)
- [Amazon EBS 볼륨 초기화 \(p. 628\)](#)
- [Linux의 RAID 구성 \(p. 629\)](#)
- [EBS 볼륨 벤치마크 \(p. 633\)](#)

Amazon EBS 성능 팁

이러한 팁은 다양한 사용자 시나리오에서 최적의 EBS 볼륨 성능을 달성하는 방법에 대한 모범 사례를 보여줍니다.

EBS 최적화 인스턴스 사용

EBS 최적화 처리량을 지원하지 않는 인스턴스에서는 네트워크 트래픽이 사용자의 인스턴스와 EBS 볼륨 간 트래픽과 경합할 수 있습니다. EBS 최적화 인스턴스에서는 이 두 유형의 트래픽이 분리되어 있습니다. 일부 EBS 최적화 인스턴스 구성은 추가 요금을 요구하지만(예: C3, R3, M3), 일부는 추가 요금 없이 항상 EBS에 최적화됩니다(예: M4, C4, D2). 자세한 내용은 [Amazon EC2 인스턴스 구성 \(p. 623\)](#) 섹션을 참조하십시오.

성능 계산 방법 이해

EBS 볼륨의 성능을 측정할 때는 관련된 측정 단위와 성능 계산 방법을 이해해야 합니다. 자세한 내용은 [I/O 특성 및 모니터링 \(p. 626\)](#) 섹션을 참조하십시오.

워크로드 이해

EBS 볼륨의 최대 성능, I/O 작업의 크기와 횟수, 각 작업을 완료하는 데 걸리는 시간은 서로 관련이 있습니다. 이러한 각 요소(성능, I/O 및 지연 시간)는 서로에게 영향을 미치며 애플리케이션마다 다른 요소에 더 민감합니다. 자세한 내용은 [EBS 볼륨 벤치마크 \(p. 633\)](#) 섹션을 참조하십시오.

스냅샷에서 볼륨을 초기화하는 경우 성능 저하에 유의

스냅샷에서 복원된 새 EBS 볼륨의 각 데이터 블록에 처음 액세스할 때 지연 시간이 상당히 증가합니다. 볼륨을 프로덕션 환경에 투입하기 전 각 블록에 액세스하면 이러한 성능 저하를 막을 수 있습니다. 이 프로세스를 초기화(이전에는 사전 워밍이라고 함)라고 합니다. 자세한 내용은 [Amazon EBS 볼륨 초기화 \(p. 628\)](#) 섹션을 참조하십시오.

HDD 성능을 저하시킬 수 있는 요인

처리량에 최적화된 HDD(st1) 또는 Cold HDD(sc1) 볼륨의 스냅샷을 생성하는 경우, 스냅샷이 진행되는 동안 성능이 볼륨의 기준 값까지 떨어질 수 있습니다. 이 동작은 이러한 볼륨 유형에만 해당합니다. 성능을 제한 할 수 있는 다른 요소로는 인스턴스가 지원할 수 있는 수준 이상의 처리량을 구동하는 경우, 스냅샷에서 복원 한 볼륨을 초기화하는 동안의 성능 저하, 볼륨에 소량의 랜덤 I/O가 과도하게 많은 경우 등을 들 수 있습니다. HDD 볼륨의 처리량 계산에 관한 자세한 내용은 [Amazon EBS 볼륨 유형](#)을 참조하십시오.

애플리케이션이 충분한 I/O 요청을 보내지 않는 경우에도 성능이 영향을 받을 수 있습니다. 볼륨의 대기열 길이와 I/O 크기를 보면 확인할 수 있습니다. 대기열 길이는 애플리케이션에서 볼륨으로 보내는 I/O 요청 중 대기 중인 요청의 수입니다. 일관성을 극대화하기 위해 HDD 지원 볼륨은 1MiB 순차 I/O를 수행하는 동안 4 이상의 대기열 길이(반올림)를 유지해야 합니다. 일정한 볼륨 성능 보장에 관한 자세한 내용은 [I/O 특성 및 모니터링 \(p. 626\)](#)를 참조하십시오.

st1 및 sc1에서 처리량이 높은 읽기 중심 워크로드의 미리 읽기 향상

일부 워크로드는 읽기 중심이며 운영 체제 페이지 캐시를 통해(예: 파일 시스템에서) 블록 디바이스에 액세스 합니다. 이 경우 최대 처리량을 획득하려면 미리 읽기 설정을 1MiB로 구성하는 것이 좋습니다. 이것은 HDD 볼륨에만 적용되는 블록 디바이스별 설정입니다. 다음 예시는 사용자가 Amazon Linux 인스턴스를 실행한다고 가정합니다.

블록 디바이스의 현재 미리읽기 값을 검사하려면 다음 명령을 사용합니다.

```
[ec2-user ~]$ sudo blockdev --report /dev/<device>
```

블록 디바이스 정보는 다음 형식으로 반환됩니다.

RO	RA	SSZ	BSZ	StartSec	Size	Device
rw	256	512	4096	4096	8587820544	/dev/<device>

보기의 디바이스는 256바이트(기본값)의 미리 읽기 값을 보고합니다. 이 값에 섹터 크기(512바이트)를 곱하면 미리읽기 버퍼의 크기를 구할 수 있습니다(이 경우에는 128KiB). 버퍼 값을 1MiB로 설정하려면 다음 명령을 사용합니다.

```
[ec2-user ~]$ sudo blockdev --setra 2048 /dev/<device>
```

첫 번째 명령을 다시 실행해서 현재 미리읽기 설정에 2,048이 표시되는지 확인합니다.

워크로드가 대용량 순차 I/O로 구성된 경우에만 이 설정을 사용합니다. 대부분이 소량 랜덤 I/O로 구성된 경우 이 설정은 실제로 성능을 저하시킵니다. 일반적으로 워크로드의 대부분의 소용량 또는 랜덤 I/O로 구성된 경우 st1이나 sc1보다는 범용 SSD(gp2) 볼륨 사용을 고려해야 합니다.

최신 Linux 커널 사용

간접 서술자를 지원하는 최신 Linux 커널을 사용합니다. 현재 세대 EC2 인스턴스뿐만 아니라 Linux 커널 3.11 이상 버전도 모두 이 지원을 제공합니다. 평균 I/O 크기가 44KiB 정도인 경우에는 간접 서술자가 지원되지 않는 인스턴스 또는 커널을 사용 중일 수 있습니다. Amazon CloudWatch 측정치에서 평균 I/O 크기를 도출하는 방법에 관한 내용은 [I/O 특성 및 모니터링 \(p. 626\)](#) 섹션을 참조하십시오.

Linux 커널 4.2 이상 버전에서 `st1` 또는 `sc1` 볼륨의 처리량을 극대화하려면, `xen_blkfront.max` 파라미터를 256으로 설정하는 것이 좋습니다. 이 파라미터는 OS 부팅 명령줄에 설정될 수 있습니다. 예를 들어 다음과 같이 Amazon Linux AMI에서 `/boot/grub/menu.lst`의 GRUB 구성에 있는 커널 라인의 끝에 이것을 추가할 수 있습니다.

```
kernel /boot/vmlinuz-4.4.5-15.26.amzn1.x86_64 root=LABEL=/ console=ttyS0
xen_blkfront.max=256
```

이 설정을 적용하려면 인스턴스를 재부팅합니다.

자세한 내용은 [GRUB 구성](#)을 참조하십시오. 다른 Linux 배포판, 특히 GRUB 부트로더를 사용하지 않는 경우에는 다른 방식으로 커널 파라미터를 조정해야 할 수 있습니다.

EBS I/O 특성에 관한 자세한 내용은 이 주제를 다룬 [Amazon EBS: Designing for Performance re:Invent 발표](#)를 참조하십시오.

RAID 0을 사용하여 인스턴스 리소스 활용도 극대화

일부 인스턴스 유형은 단일 EBS 볼륨에 대해 프로비저닝할 수 있는 것보다 많은 I/O 처리량을 구동할 수 있습니다. 여러 `gp2`, `io1`, `st1` 또는 `sc1` 볼륨을 RAID 0 구성으로 함께 조인하여 이 인스턴스에 사용 가능한 대역폭을 사용할 수 있습니다. 자세한 내용은 [Linux의 RAID 구성 \(p. 629\)](#) 섹션을 참조하십시오.

Amazon CloudWatch로 성능 추적

Amazon Web Services는 Amazon CloudWatch를 통해 보고 분석할 수 있는 Amazon EBS에 대한 성능 지표와 볼륨의 상태를 모니터링하는 데 사용할 수 있는 상태 검사를 제공합니다. 자세한 내용은 [볼륨 상태 모니터링 \(p. 580\)](#) 섹션을 참조하십시오.

Amazon EC2 인스턴스 구성

애플리케이션에 맞게 EBS 볼륨을 계획하고 구성할 때는 볼륨을 연결할 인스턴스의 구성을 고려해야 합니다. EBS 볼륨의 성능을 최대한 활용하려면 EBS에 최적화된 인스턴스 또는 10Gb 네트워크 연결이 있는 인스턴스와 같이 볼륨을 지원할 수 있는 충분한 대역폭을 갖춘 인스턴스에 볼륨을 연결해야 합니다. 이것은 RAID 구성에서 여러 볼륨을 함께 스트라이프할 때 특히 중요합니다.

EBS에 최적화된 인스턴스 또는 10Gb 네트워크 인스턴스 사용

프로덕션 데이터베이스 또는 비즈니스 애플리케이션과 같이 가변성을 최소화하고 Amazon EBS 트래픽 전용 Amazon EC2를 사용해야 하는 성능에 민감한 작업은 EBS에 최적화된 인스턴스나 10Gb 네트워크 연결이 있는 인스턴스에 연결되는 볼륨을 사용해야 합니다. 이 기준에 맞지 않는 EC2 인스턴스는 네트워크 리소스에 대한 보증을 제공하지 않습니다. EC2 인스턴스와 EBS 볼륨 간에 지속적이고 안정적인 네트워크 대역폭을 보장하는 유일한 방법은 EC2 인스턴스를 EBS에 최적화된 인스턴스로 시작하거나 10Gb 네트워크 연결이 있는 인스턴스 유형을 선택하는 것입니다. 10Gb 네트워크 연결이 있는 인스턴스 유형을 확인하려면 [Instance Type Details](#) 섹션을 참조하십시오. EBS 최적화 인스턴스 구성에 대한 자세한 내용은 [Amazon EBS 최적화 인스턴스](#)를 참조하십시오.

충분한 대역폭이 있는 EC2 인스턴스 선택

EBS에 최적화된 인스턴스를 시작하면 EC2 인스턴스와 EBS 볼륨 간에 전용 연결이 제공됩니다. 그러나 특히 여러 볼륨이 RAID 구성으로 스트라이프된 경우 특정 인스턴스 유형에 사용 가능한 대역폭을 초과하는 EBS 볼륨도 프로비저닝할 수 있습니다. 다음 표에서는 EBS에 최적화된 인스턴스로 시작할 수 있는 인스턴

스 유형, EBS에 최적화된 인스턴스로 시작할 수 있는 인스턴스 유형 전용 처리량, Amazon EBS 전용 대역폭, 16KB I/O 크기를 사용할 경우 인스턴스가 지원할 수 있는 IOPS 최대량, 해당 연결에서 사용할 수 있는 대략적인 I/O 대역폭(MB/s)을 보여 줍니다. 애플리케이션에 필요한 것보다 많은 전용 EBS 처리량을 제공하는 EBS에 최적화된 인스턴스를 선택해야 합니다. 그렇지 않으면 Amazon EBS와 Amazon EC2 연결로 인해 성능 병목 현상이 발생합니다.

Note

아래 표와 다음 예에서는 설명을 목적으로만 16KB를 I/O 크기로 사용합니다. 애플리케이션 I/O 크기는 다를 수 있습니다(Amazon EBS는 초당 각 I/O 작업(256KiB 이하)을 하나의 IOPS로 측정함). IOPS 및 I/O 크기와 볼륨 처리량 제한 간 관계에 대한 자세한 내용은 [I/O 특성 및 모니터링 \(p. 626\)](#) 섹션을 참조하십시오.

인스턴스 유형	기본적으로 EBS 최적화됨	최대 대역폭 (Mbps)*	예상 처리량(Mb/s)**	최대 IOPS(16KB I/O 크기)**
c1.xlarge		1,000	125	8,000
c3.xlarge		500	62.5	4,000
c3.2xlarge		1,000	125	8,000
c3.4xlarge		2,000건	250	16,000
c4.large	예	500	62.5	4,000
c4.xlarge	예	750	93.75	6,000
c4.2xlarge	예	1,000	125	8,000
c4.4xlarge	예	2,000건	250	16,000
c4.8xlarge	예	4,000	500	32,000
d2.xlarge	예	750	93.75	6,000
d2.2xlarge	예	1,000	125	8,000
d2.4xlarge	예	2,000건	250	16,000
d2.8xlarge	예	4,000	500	32,000
g2.2xlarge		1,000	125	8,000
i2.xlarge		500	62.5	4,000
i2.2xlarge		1,000	125	8,000
i2.4xlarge		2,000건	250	16,000
i3.large	예	425	50	3000
i3.xlarge	예	850	100	6000
i3.2xlarge	예	1,700	200	12,000
i3.4xlarge	예	3,500	400	16,000
i3.8xlarge	예	7,000	850	32,500
i3.16xlarge	예	14,000	1,750	65,000
m1.large		500	62.5	4,000

Amazon Elastic Compute Cloud
Linux 인스턴스용 사용 설명서
EBS 성능

인스턴스 유형	기본적으로 EBS 최적화됨	최대 대역폭 (Mbps)*	예상 처리량(Mb/s)**	최대 IOPS(16KB I/O 크기)**
m1.xlarge		1,000	125	8,000
m2.2xlarge		500	62.5	4,000
m2.4xlarge		1,000	125	8,000
m3.xlarge		500	62.5	4,000
m3.2xlarge		1,000	125	8,000
m4.large	예	450	56.25	3,600
m4.xlarge	예	750	93.75	6,000
m4.2xlarge	예	1,000	125	8,000
m4.4xlarge	예	2,000건	250	16,000
m4.10xlarge	예	4,000	500	32,000
m4.16xlarge	예	10,000개	1,250	65,000
p2.xlarge	예	750	93.75	6,000
p2.8xlarge	예	5,000	625	32,500
p2.16xlarge	예	10,000개	1,250	65,000
r3.xlarge		500	62.5	4,000
r3.2xlarge		1,000	125	8,000
r3.4xlarge		2,000건	250	16,000
r4.large	예	400	50	3,000
r4.xlarge	예	800	100	6,000
r4.2xlarge	예	1600	200	12,000
r4.4xlarge	예	3000	375	16,000
r4.8xlarge	예	6000	750	32,000
r4.16xlarge	예	12,000	1,500	65,000
x1.16xlarge	예	5,000	625	32,500
x1.32xlarge	예	10,000개	1,250	65,000

* 이 수준의 성능을 일관되게 유지하려면 이러한 인스턴스 유형을 EBS에 최적화된 상태로 시작해야 합니다.

** 이 값은 100% 읽기 전용 작업을 기준으로 반올림한 근사치이며, 기준 구성을 보조할 목적으로 제공됩니다. EBS에 최적화된 연결은 전이중 모드로서, 두 통신 경로를 모두 사용하는 경우 50/50 읽기/쓰기 작업으로 균형을 이루며 처리 속도와 IOPS가 향상됩니다. 네트워크, 파일 시스템 및 Amazon EBS encryption 오버헤드로 인해 가용한 최대 처리량과 IOPS가 줄어드는 경우도 있습니다.

i2.8xlarge, c3.8xlarge 및 r3.8xlarge와 같이 10기가비트 네트워크 인터페이스를 사용하는 일부 인터페이스는 EBS 최적화에 영향을 미치지 않으므로, 사용 가능한 전용 EBS 대역폭이 없어 여기에 나열되지 않습니다.

니다. 그러나 애플리케이션이 Amazon EBS와 경합하는 다른 네트워크 트래픽을 보내지 않는다면 Amazon EBS로 가는 트래픽에 할당된 모든 대역폭을 사용할 수 있습니다. `c4.8xlarge` 및 `d2.8xlarge`와 같은 일부 다른 10기가비트 네트워크 인스턴스는 네트워크 트래픽에만 사용되는 10기가비트 인터페이스 외에도 전용 Amazon EBS 대역폭을 제공합니다.

`m1.large` 인스턴스의 최대 16KB IOPS 값은 4,000이지만 이 인스턴스 유형이 EBS에 최적화된 인스턴스로 시작되지 않는 한 해당 값은 절대적인 최상의 사례 시나리오일 뿐이며 보장되지 않습니다. 일관되게 4,000 16KB IOPS를 달성하려면 이 인스턴스를 EBS에 최적화된 인스턴스로 시작해야 합니다. 그러나 4,000 IOPS `io1` 볼륨이 EBS에 최적화된 `m1.large` 인스턴스에 연결된 경우 Amazon EC2와 Amazon EBS의 연결 대역폭 제한으로 인해 이 볼륨은 사용 가능한 최대 집계 처리량인 320MB/s를 제공할 수 없습니다. 이 경우 `c4.8xlarge` 인스턴스 유형과 같이 최소한 320MB/s의 처리량을 지원하는 EBS에 최적화된 EC2 인스턴스를 사용해야 합니다.

범용 SSD(`gp2`) 유형의 볼륨에는 볼륨당 128MB/s~160MB/s의 처리량 제한(볼륨 크기에 따라 다름)이 있으며, 이 제한은 1,000Mbps EBS에 최적화된 연결에 적합합니다. 1,000Mbps 이상의 Amazon EBS 처리량을 제공하는 인스턴스 유형은 사용 가능한 처리량을 이용하기 위해 두 개 이상의 `gp2` 볼륨을 사용할 수 있습니다. 프로비저닝된 IOPS SSD(`io1`) 유형의 볼륨은 프로비저닝된 각 IPOS에 대해 256KB의 처리 한계를 갖고 최대 값은 320MB/s(1,280 IOPS에서)입니다. 자세한 내용은 [Amazon EBS 볼륨 유형 \(p. 564\)](#) 섹션을 참조하십시오.

10Gb 네트워크 연결이 있는 인스턴스 유형은 암호화되지 않은 Amazon EBS 볼륨에 대해 최대 800MB/s의 처리량과 48,000 16K IOPS를 지원하며 암호화된 Amazon EBS 볼륨에 대해 최대 25,000 16K IOPS를 지원합니다. EBS 볼륨에 대한 최대 `io1` 값이 `io1` 볼륨의 경우 20,000 및 `gp2` 볼륨의 경우 10,000이므로 여러 EBS 볼륨을 동시에 사용하여 이 인스턴스 유형에 사용 가능한 I/O 성능 수준에 도달할 수 있습니다. 10Gb 네트워크 연결이 있는 인스턴스 유형에 대한 자세한 내용은 [인스턴스 유형 세부 정보](#) 섹션을 참조하십시오.

Amazon EBS `gp2` 및 `io1` 볼륨의 성능상 이점을 최대한 활용하려면 사용 가능한 경우 EBS에 최적화된 인스턴스를 사용해야 합니다. 자세한 내용은 [Amazon EBS 최적화 인스턴스 \(p. 614\)](#) 섹션을 참조하십시오.

I/O 특성 및 모니터링

지정된 볼륨 구성에서 특정 I/O 특성은 EBS 볼륨의 성능 동작을 구동합니다. SSD 지원 볼륨-범용 SSD(`gp2`) 및 프로비저닝된 IOPS SSD(`io1`)-I/O 작업이 랜덤이든 순차든 상관 없이 일관된 성능을 제공합니다. HDD 지원 볼륨-처리량에 최적화된 HDD(`st1`) 및 Cold HDD(`sc1`)-I/O 작업이 대용량 순차인 경우에만 최적의 성능을 제공합니다. SSD 및 HDD 볼륨이 애플리케이션에서 어떻게 작동하는지 이해하려면 볼륨 수요와 가용 IOPS 수량, I/O 작업을 완료하는 데 소요되는 시간, 볼륨의 처리량 제한 사이의 관계를 아는 것이 중요합니다.

IOPS

IOPS는 초당 입력/출력 작업을 나타내는 측정 단위입니다. 작업은 KiB 단위로 측정되며, 기본 드라이브 기술에 따라 볼륨 유형이 단일 I/O로 계산하는 최대 데이터 용량이 결정됩니다. I/O 크기의 최대 한도는 SSD 볼륨이 256KiB, HDD 볼륨이 1,024KiB이며 이렇게 차이가 나는 이유는 SSD 볼륨이 HDD 볼륨보다 소용량 또는 랜덤 I/O를 훨씬 더 효과적으로 처리하기 때문입니다.

소용량 I/O 작업이 물리적으로 연속되는 경우, Amazon EBS가 최대 크기까지 단일 I/O로 병합을 시도합니다. 예를 들어, SSD 볼륨의 경우 1건의 1,024KiB I/O 작업은 4건($1,024 \div 256 = 4$)의 작업으로 계산되지만 각각 32KiB인 8건의 연속하는 I/O 작업은 1건($8 \times 32 = 256$)의 작업으로 계산됩니다. 하지만 각각 32KiB인 8건의 랜덤 I/O 작업은 8건의 작업으로 계산됩니다. 각각 32KiB보다 작은 I/O 작업은 1건의 작업으로 계산됩니다.

마찬가지로, HDD 지원 볼륨의 경우 1,024KiB I/O 작업 1건과 128KiB 순차 작업 8건은 모두 각각 1건의 작업으로 계산됩니다. 하지만 랜덤 128KiB I/O 작업 8건은 8건의 작업으로 계산됩니다.

그러므로 3,000 IOPS를 지원하는 SSD 지원 볼륨을 생성하여(`io1` 볼륨을 3,000 IOPS에서 프로비저닝하거나 `gp2` 볼륨의 크기를 1,000GiB에서 조정하는 방법으로), 충분한 대역폭을 제공할 수 있는 EBS 최적화 인스턴스에 연결할 경우, 초당 최대 3,000 I/O 데이터 전송이 가능하며 처리량은 I/O 크기에 따라 결정됩니다.

볼륨 대기열 길이 및 지연 시간

볼륨 대기열 길이는 장치에 대해 보류 중인 I/O 요청 수입니다. 지연 시간은 I/O 작업의 실제 종단 간 클라이언트 시간입니다. 다시 말해 EBS로 I/O를 전송한 후 EBS로부터 I/O 읽기 또는 쓰기가 완료되었다는 승인을

발기까지 소요된 시간입니다. 대기열 길이를 I/O 크기 및 지연 시간에 따라 정확히 보정하여, 게스트 운영 체제나 EBS로 연결되는 네트워크 링크에 병목 현상이 발생하지 않도록 해야 합니다.

최적의 대기열 길이는 워크로드마다 다른데, IOPS 및 지연 시간에 대한 특정 애플리케이션의 민감도에 따라 결정됩니다. 워크로드가 EBS 볼륨에 대해 사용 가능한 성능을 전부 사용할 만큼 충분한 I/O 요청을 제공하지 않는 경우, 프로비저닝된 처리량이나 IOPS를 볼륨이 제공하지 못할 수 있습니다.

트랜잭션 집약적인 애플리케이션은 I/O 지연 시간 증가에 민감하며, SSD 지원 `io1` 및 `gp2` 볼륨에 적합합니다. 대기열 길이를 줄이고 볼륨에서 사용할 수 있는 IOPS 개수를 늘리면 높은 IOPS를 유지하는 동시에 지연 시간을 단축할 수 있습니다. 볼륨이 수용할 수 있는 수준보다 높은 IOPS를 계속 구동하면 I/O 지연 시간이 길어질 수 있습니다.

처리량 집약적인 애플리케이션은 I/O 지연 시간 증가에 덜 민감하며, HDD 지원 `st1` 및 `sc1` 볼륨에 적합합니다. 대용량 순차 I/O를 수행할 때 대기열 길이를 길게 유지하면 HDD 지원 볼륨에서 높은 처리량을 유지할 수 있습니다.

I/O 크기 및 볼륨 처리량 제한이 없음

SSD 지원 볼륨의 경우, I/O 크기가 매우 크면 볼륨 처리량 제한에 도달하기 때문에 프로비저닝한 것보다 IOPS가 적을 수 있습니다. 예를 들어, 버스트 크레딧을 사용할 수 있는 1,000GiB 미만의 `gp2` 볼륨에는 3,000 IOPS 제한과 160MiB/s의 볼륨 처리량 제한이 있습니다. 256KiB I/O 크기를 사용하는 경우, 볼륨은 640 IOPS에서 처리량 제한에 도달합니다($640 \times 256\text{KiB} = 160\text{MiB}$). I/O 크기가 작다면(예: 16KiB), 처리량이 160MiB/s에 훨씬 못 미치기 때문에 동일한 볼륨이 3,000 IOPS를 유지할 수 있습니다. (이 예제는 볼륨의 I/O가 인스턴스의 처리량 제한에 도달하지 않는다고 가정합니다.) 각 EBS 볼륨 유형의 처리량 제한에 대한 자세한 내용은 [Amazon EBS 볼륨 유형 \(p. 564\)](#) 섹션을 참조하십시오.

소용량 I/O 작업의 경우, 인스턴스 내에서 측정했을 때 프로비저닝된 IOPS 값보다 큰 값을 관찰할 수 있습니다. 인스턴스 운영 체제가 소용량 I/O 작업을 Amazon EBS로 전달하기 전 대용량 작업에 병합할 때 이런 결과가 발생합니다.

워크로드가 HDD 지원 `st1` 및 `sc1` 볼륨에 대해 순차 I/O를 사용하는 경우, 인스턴스 내에서 측정했을 때 예상보다 높은 IOPS를 관찰할 수 있습니다. 인스턴스 운영 체제가 순차 I/O를 병합하고 1,024KiB 크기 단위로 계산되는 경우에 이런 결과가 발생합니다. 워크로드가 소용량 또는 랜덤 I/O를 사용하는 경우 예상보다 적은 처리량을 관찰할 수 있습니다. 이는 각각의 비순차적인 랜덤 I/O를 총 IOPS 계산에 적용하기 때문이며, 이로 인해 예상보다 일찍 볼륨의 IOPS 제한에 도달할 수 있습니다.

EBS 볼륨 유형이 무엇이든 현재 구성에서 기대한 IOPS 또는 처리량을 달성하지 못할 경우에는 EC2 인스턴스 대역폭이 제한 요소가 아닌지 확인하십시오. 최적의 성능을 위해 항상 현재 세대 EBS 최적화 인스턴스(또는 10Gb/s 네트워크 연결을 포함한 인스턴스)를 사용해야 합니다. 자세한 내용은 [Amazon EC2 인스턴스 구성 \(p. 623\)](#) 섹션을 참조하십시오. EBS 볼륨에 충분한 I/O를 구동하고 있지 않은 경우에도 IOPS가 예상과 다를 수 있습니다.

CloudWatch로 I/O 특성 모니터링

각 볼륨의 [CloudWatch 측정치](#)로 이러한 I/O 특성을 모니터링할 수 있습니다. 고려해야 할 중요 측정치로는 다음 항목이 포함됩니다.

- `BurstBalance`
- `VolumeReadBytes`
- `VolumeWriteBytes`
- `VolumeReadOps`
- `VolumeWriteOps`
- `VolumeQueueLength`

`BurstBalance`은 `gp2`, `st1`, `sc1` 볼륨에 대한 버스트 버킷 잔고를 남은 잔고에 대한 비율로 표시합니다. 버스트 버킷이 모두 사용되면 볼륨 I/O 크레딧(`gp2` 볼륨) 또는 볼륨 처리량 크레딧(`st1` 및 `sc1` 볼륨)이 기준 수준으로 스로틀링됩니다. `BurstBalance` 값을 확인하여 이런 이유로 볼륨이 조절되는지 판단합니다.

HDD 지원 `st1` 및 `sc1` 볼륨은 1,024KiB 최대 I/O 크기를 활용하는 워크로드에서 가장 잘 작동하도록 설계되었습니다. 볼륨의 평균 I/O 크기를 구하려면 `VolumeWriteBytes` 를 `VolumeWriteOps` 값으로 나눕니다. 읽기 작업에도 같은 계산 방법이 적용됩니다. 평균 I/O 크기는 64KiB 미만이며, `st1` 또는 `sc1` 볼륨으로 보내는 I/O 작업의 크기가 큰 경우 성능을 개선해야 합니다.

Note

평균 I/O 크기가 44KiB이거나 그에 가까운 경우에는 간접 서술자가 지원되지 않는 인스턴스 또는 커널을 사용 중일 수 있습니다. 현재 세대 인스턴스뿐만 아니라 Linux 커널 3.8 이상 버전도 모두 이 지원을 제공합니다.

I/O 지연 시간이 필요한 것보다 긴 경우 `VolumeQueueLength`을 확인하여 애플리케이션이 프로비저닝한 것보다 많은 IOPS를 구동하려고 하고 있지 않은지 확인하십시오. 볼륨이 제공할 수 있는 것보다 많은 수의 IOPS를 요구하는 애플리케이션인 경우, 기준 성능 수준이 높은 대용량 `gp2` 볼륨 또는 프로비저닝된 IOPS가 더 많은 `io1` 볼륨을 사용하여 지연 시간을 줄이는 것을 고려해야 합니다.

Amazon EBS I/O 특성에 관한 자세한 내용은 이 주제를 다룬 [Amazon EBS: Designing for Performance re:Invent 발표](#)를 참조하십시오.

Amazon EBS 볼륨 초기화

새 EBS 볼륨은 사용 가능하지만 초기화(이전에는 사전 위밍이라고 함)가 필요하지 않은 시점에 최고 성능을 발휘합니다. 하지만 스냅샷에서 복원된 볼륨의 스토리지 블록은 초기화(Amazon S3에서 가져와 볼륨에 기록) 후에만 액세스할 수 있습니다. 이 예비 작업은 시간이 걸리며, 각 블록을 처음 액세스할 때 I/O 작업의 지연 시간을 상당히 증가시킬 수 있습니다. 대부분 애플리케이션의 경우 볼륨 수명 주기 동안 이 비용을 분할 상환할 수 있습니다. 데이터에 한 번 액세스한 후에는 성능이 복원됩니다.

사용하기 전에 볼륨의 모든 블록에서 읽기 작업을 완료하여 프로덕션 환경에서 이 성능 충돌을 방지할 수 있습니다. 이 프로세스를 초기화라고 합니다. 스냅샷에서 생성된 새로운 볼륨의 경우 볼륨을 사용하기 전에 데이터가 있는 모든 블록을 읽어야 합니다.

Important

스냅샷에서 복원한 `io1` 볼륨을 초기화할 경우 볼륨의 성능이 예상 수준보다 50퍼센트 이하로 떨어질 수 있으며, 이로 인해 볼륨에서 [I/O Performance] 상태 확인에 대해 warning 상태를 표시할 수 있습니다. 이는 원래 그런 것임으로 초기화 중에는 `io1` 볼륨에 대한 warning 상태를 무시해도 됩니다. 자세한 내용은 [상태 확인으로 볼륨 모니터링 \(p. 583\)](#) 섹션을 참조하십시오.

Linux에서 Amazon EBS 볼륨 초기화

새 EBS 볼륨은 사용 가능하지만 초기화(이전에는 사전 위밍이라고 함)가 필요하지 않은 시점에 최고 성능을 발휘합니다. 스냅샷에서 복원된 볼륨의 경우, `dd` 또는 `fio` 유ти리티를 사용하여 볼륨의 모든 블록에서 읽습니다. 볼륨의 기존 데이터는 모두 보존됩니다.

Linux의 스냅샷에서 복원된 볼륨을 초기화하려면

1. 새로 복원된 볼륨을 Linux 인스턴스에 연결합니다.
2. `lsblk` 명령을 사용하여 인스턴스의 블록 디바이스를 나열합니다.

```
[ec2-user ~]$ lsblk
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvdf  202:80   0   30G  0 disk
xvda1 202:1    0   8G  0 disk /
```

여기에서 새로운 볼륨인 `/dev/xvdf`가 연결되지만 마운트되지 않음을 확인할 수 있습니다. `MOUNTPOINT` 열 아래에 나열된 경로가 없기 때문입니다.

3. dd 또는 fio 유틸리티를 사용하여 디바이스의 모든 블록을 읽습니다. dd 명령은 Linux 시스템에 기본으로 설치되지만, fio는 다중 스레드 읽기를 허용하기 때문에 훨씬 더 빠릅니다.

Note

이 단계는 EC2 인스턴스 대역폭, 볼륨에 대해 프로비저닝된 IOPS 및 볼륨 크기에 따라 몇 분에서 몇 시간까지 걸릴 수 있습니다.

- dd 사용: if(입력 파일) 파라미터는 초기화할 드라이브로 설정해야 합니다. of(파일 출력) 파라미터를 Linux null 가상 디바이스인 /dev/null로 설정해야 합니다. bs 파라미터는 읽기 작업의 블록 크기를 설정합니다. 최적의 성능을 얻으려면 이 값을 1MB로 설정해야 합니다.

```
[ec2-user ~]$ sudo dd if=/dev/xvdf of=/dev/null bs=1M
```

- fio 사용: 시스템에 fio가 설치되어 있는 경우, 아래 명령을 복사 및 붙여넣기하여 볼륨을 초기화할 수 있습니다. --filename(입력 파일) 파라미터는 초기화할 드라이브로 설정해야 합니다.

Note

Amazon Linux에 fio를 설치하려면 sudo yum install -y fio 명령을 사용합니다.
Ubuntu에 fio를 설치하려면 sudo apt-get install -y fio 명령을 사용합니다.

```
[ec2-user ~]$ sudo fio --filename=/dev/xvdf --rw=read --bs=128k --iodepth=32 --  
ioengine=libaio --direct=1 --name=volume-initialize
```

작업이 끝나면 읽기 작업에 대한 보고서가 나타납니다. 이제 볼륨을 사용할 준비가 되었습니다. 자세한 내용은 [Amazon EBS 볼륨을 사용할 수 있도록 만들기 \(p. 577\)](#)를 참조하십시오.

Linux의 RAID 구성

Amazon EBS를 사용하면 기존 운영 체제 미설치 서버에서 사용 가능한 스탠다드 RAID 구성을 사용할 수 있습니다. 단, 해당 RAID 구성이 인스턴스에 대한 운영 체제에서 지원되어야 합니다. 이는 모든 RAID가 소프트웨어 수준에서 실행되기 때문입니다. 단일 볼륨을 사용할 때보다 더 나은 I/O 성능을 얻기 위해 RAID 0은 여러 볼륨을 함께 스트라이프할 수 있고, 온인스턴스 중복을 위해 RAID 1은 두 개의 볼륨을 함께 미러링할 수 있습니다.

Amazon EBS 볼륨 데이터는 단일 구성 요소의 고장으로 인한 데이터 손실을 방지하기 위해 가용 영역의 여러 서버에 복제됩니다. 이 복제 기능으로 인해 Amazon EBS 볼륨이 일반 상용 디스크 드라이브보다 10배 더 안정적입니다. 자세한 내용은 Amazon EBS 제품 정보 페이지의 [Amazon EBS Availability and Durability](#) 섹션을 참조하십시오.

Note

RAID 볼륨에서는 부팅하지 않아야 합니다. Grub은 일반적으로 RAID 어레이의 디바이스 하나에만 설치되며, 미러링된 디바이스 중 하나에 장애가 발생할 경우 운영 체제를 부팅하지 못할 수 있습니다.

Windows 인스턴스에서 RAID 어레이를 생성해야 하는 경우 Windows 인스턴스용 Amazon EC2 사용 설명서의 [RAID Configuration on Windows](#) 섹션을 참조하십시오.

목차

- [RAID 구성 옵션 \(p. 629\)](#)
- [Linux에서 RAID 어레이 생성 \(p. 630\)](#)

RAID 구성 옵션

다음 표에서는 일반 RAID 0 옵션과 RAID 1 옵션을 비교합니다.

구성	--set-visible-to-all-users	장점	단점
RAID 0	I/O 성능이 내결합성보다 더 중요한 경우(예: 데이터 복제가 이미 별도로 설정되어 있는 사용량이 많은 데이터베이스)	I/O가 스트라이프의 볼륨에 분산됩니다. 볼륨을 추가하면 처리량도 그에 따라 바로 추가됩니다.	스트라이프의 성능이 세트에서 가장 성능이 떨어지는 볼륨으로 제한됩니다. 단일 볼륨이 손실되면 어레이의 데이터가 완전히 손실됩니다.
RAID 1	내결합성이 I/O 성능보다 더 중요한 경우(예: 중요 애플리케이션)	데이터 내구성 면에서 더 안전합니다.	쓰기 성능이 향상되지 않습니다. 또한 데이터를 동시에 여러 볼륨에 쓰기 때문에 비 RAID 구성에 비해 Amazon EC2와 Amazon EBS 사이에 더 큰 대역폭이 필요합니다.

Important

RAID 5 및 RAID 6는 이 RAID 모드의 패리티 쓰기 작업에서 볼륨에 사용 가능한 IOPS의 일부를 사용하기 때문에 Amazon EBS에 권장되지 않습니다. RAID 어레이의 구성에 따라 이러한 RAID 모드에서는 RAID 0 구성보다 20-30% 더 적은 가용 IOPS를 제공합니다. 이러한 RAID 모드는 비용 증가의 한 요인이기도 합니다. 볼륨 크기와 속도가 동일할 경우 2 볼륨 RAID 0 어레이가 두 배 더 비싼 4 볼륨 RAID 6 어레이보다 더 우수한 성능을 제공합니다.

RAID 0 어레이를 생성하면 단일 Amazon EBS 볼륨에서 프로비저닝할 때보다 파일 시스템의 성능이 더 향상됩니다. RAID 1 어레이는 중복성 강화를 위해 데이터를 "미러링"합니다. 이 절차를 수행하기 전에 RAID 어레이의 크기와 프로비저닝할 IOPS 수를 결정해야 합니다.

RAID 0 어레이의 결과 크기는 어레이 내 볼륨의 크기 합계이고, 대역폭은 어레이 내 볼륨의 가용 대역폭 합계입니다. RAID 1 어레이의 결과 크기 및 대역폭은 어레이 내 볼륨의 크기 및 대역폭과 같습니다. 예를 들어, 4,000의 프로비저닝된 IOPS가 있는 두 500GiB Amazon EBS 볼륨은 각각 가용 대역폭이 8,000 IOPS이고 처리량이 640MB/s인 1000GiB RAID 0 어레이를 생성하거나, 가용 대역폭이 4,000 IOPS이고 처리량이 320MB/s인 500GiB RAID 1 어레이를 생성합니다.

이 문서는 기본 RAID 설정의 예를 제공합니다. RAID 구성, 성능 및 복구에 대한 자세한 내용은 https://raid.wiki.kernel.org/index.php/Linux_Raid에서 Linux RAID Wiki 섹션을 참조하십시오.

Linux에서 RAID 어레이 생성

다음 절차에 따라 RAID 어레이를 생성합니다. Windows 인스턴스에 대한 지침은 Windows 인스턴스용 Amazon EC2 사용 설명서의 [Windows에서 RAID 어레이 생성](#)을 참조하십시오.

Linux에서 RAID 어레이를 생성하려면

1. 어레이에 대한 Amazon EBS 볼륨을 생성합니다. 자세한 내용은 [Amazon EBS 볼륨 생성 \(p. 573\)](#) 섹션을 참조하십시오.

Important

어레이에 대해 크기 및 IOPS 성능 값이 동일한 볼륨을 생성합니다. EC2 인스턴스의 가용 대역폭을 초과하는 어레이를 생성하지 마십시오. 자세한 내용은 [Amazon EC2 인스턴스 구성 \(p. 623\)](#) 섹션을 참조하십시오.

2. 어레이를 호스팅할 인스턴스에 Amazon EBS 볼륨을 연결합니다. 자세한 내용은 [Amazon EBS 볼륨을 인스턴스에 연결 \(p. 576\)](#) 섹션을 참조하십시오.
3. mdadm 명령을 사용하여 새로 연결된 Amazon EBS 볼륨에서 로직 RAID 디바이스를 생성합니다. `number_of_volumes`에 대한 어레이의 볼륨 수와 `device_name`에 대한 어레이에 있는 각 볼륨의 디바이스 이름(예: /dev/xvdf)을 대체합니다. 어레이의 고유 이름으로 `MY_RAID`를 대체할 수도 있습니다.

Note

lsblk 명령으로 인스턴스에 디바이스를 나열하여 디바이스 이름을 찾을 수 있습니다.
(RAID 0에만 해당) RAID 0 어레이를 생성하려면 다음 명령을 실행합니다(어레이를 스트라이프하려면 --level=0 옵션 사용).

```
[ec2-user ~]$ sudo mdadm --create --verbose /dev/md0 --level=0 --name=MY_RAID --raid-devices=number_of_volumes device_name1 device_name2
```

(RAID 1에만 해당) RAID 1 어레이를 생성하려면 다음 명령을 실행합니다(어레이를 미러링하려면 --level=1 옵션 사용).

```
[ec2-user ~]$ sudo mdadm --create --verbose /dev/md0 --level=1 --name=MY_RAID --raid-devices=number_of_volumes device_name1 device_name2
```

4. RAID 어레이가 초기화되고 동기화될 때까지 기다립니다. 다음 명령을 사용하여 이 작업의 진행을 추적할 수 있습니다.

```
[ec2-user ~]$ sudo cat /proc/mdstat
```

이렇게 하면 다음과 같은 출력이 생성됩니다.

```
Personalities : [raid1]
md0 : active raid1 xvdf[1] xvdf[0]
      20955008 blocks super 1.2 [2/2] [UU]
      [= =====>.....]  resync = 46.8% (9826112/20955008) finish=2.9min
speed=63016K/sec
```

일반적으로 다음 명령을 사용하여 RAID 어레이에 대한 자세한 정보를 표시할 수 있습니다.

```
[ec2-user ~]$ sudo mdadm --detail /dev/md0
```

이렇게 하면 다음과 같은 정보가 생성됩니다.

```
/dev/md0:
      Version : 1.2
      Creation Time : Mon Jun 27 11:31:28 2016
      Raid Level : raid1
      Array Size : 20955008 (19.98 GiB 21.46 GB)
      Used Dev Size : 20955008 (19.98 GiB 21.46 GB)
      Raid Devices : 2
      Total Devices : 2
      Persistence : Superblock is persistent

      Update Time : Mon Jun 27 11:37:02 2016
      State : clean
      ...
      ...

      Number  Major  Minor  RaidDevice State
          0      202       80          0    active sync   /dev/sdf
          1      202       96          1    active sync   /dev/sdg
```

5. RAID 어레이에서 파일 시스템을 생성하고 이후 해당 파일 시스템에 마운트할 때 사용할 레이블을 지정합니다. 예를 들어, **MY_RAID** 레이블로 ext4 파일 시스템을 생성하려면

```
[ec2-user ~]$ sudo mkfs.ext4 -L MY_RAID /dev/md0
```

명령을 실행합니다. 애플리케이션이의 요구 사항이나 운영 체제의 제한 사항에 따라 ext3 또는 XFS 같은 다른 파일 시스템 유형을 사용할 수 있습니다. 해당 파일 시스템 생성 명령은 파일 시스템 설명서를 참조하십시오.

6. RAID 어레이에 대한 마운트 지점을 생성합니다.

```
[ec2-user ~]$ sudo mkdir -p /mnt/raid
```

7. 마지막으로 생성한 마운트 지점에 RAID 디바이스를 마운트합니다.

```
[ec2-user ~]$ sudo mount LABEL=MY_RAID /mnt/raid
```

이제 RAID 디바이스를 사용할 준비가 되었습니다.

8. (선택 사항) 시스템을 재부팅할 때마다 이 Amazon EBS 볼륨을 마운트하려면 디바이스에 대한 항목을 /etc/fstab 파일에 추가합니다.
 - a. 수정 도중 실수로 이 파일이 손상되거나 삭제되는 경우에 대비하여 /etc/fstab 파일의 백업을 생성합니다.

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

- b. 텍스트 편집기를 사용하여 nano 또는 vim 등의 명령으로 /etc/fstab 파일을 업니다.
- c. "UUID="로 시작하는 줄을 주석 처리하고, 파일 끝에 다음 형식으로 RAID 볼륨을 위한 줄을 새로 추가합니다.

```
device_label    mount_point    file_system_type    fs_mntops    fs_freq    fs_passno
```

이 줄의 마지막 세 필드는 파일 시스템 마운트 옵션, 파일 시스템의 덤프 빈도 및 부팅 시 파일 시스템 확인 순서입니다. 어떤 값을 입력해야 하는지 모르는 경우 아래 예제에 제공된 값을 사용하십시오(defaults,nofail 0 2). /etc/fstab 항목에 대한 자세한 내용은 fstab 매뉴얼 페이지(명령줄에 man fstab 입력)를 참조하십시오. 예를 들어, MY_RAID 레이블이 있는 디바이스에 /mnt/raid 마운트 지점에서 ext4 파일 시스템을 마운트하려면 /etc/fstab에 다음 항목을 추가합니다.

Note

이 볼륨을 연결하지 않고 인스턴스를 부팅하려면(예: 이 볼륨이 서로 다른 인스턴스 사이를 이동할 수 있도록) 볼륨 마운트 시 오류가 있어도 인스턴스가 부팅되도록 하는 nofail 마운트 옵션을 추가해야 합니다. Ubuntu와 같은 Debian 계열 시스템에서는 nobootwait 마운트 옵션도 추가해야 합니다.

LABEL=MY_RAID	/mnt/raid	ext4	defaults,nofail	0	2
---------------	-----------	------	-----------------	---	---

- d. /etc/fstab에 새 항목을 추가한 다음에는 해당 항목이 작동하는지 확인해야 합니다. sudo mount -a 명령을 실행하여 /etc/fstab에서 모든 파일 시스템을 마운트합니다.

```
[ec2-user ~]$ sudo mount -a
```

이전 명령에서 오류가 발생하지 않으면 /etc/fstab 파일이 정상이고 다음 부팅 시 파일 시스템이 자동으로 마운트됩니다. 명령에서 오류가 발생하면 오류를 검토한 다음 /etc/fstab를 수정합니다.

Warning

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

- e. (선택 사항) /etc/fstab 오류 수정 방법을 모르는 경우 다음 명령으로 항상 백업 /etc/fstab 파일을 복원할 수 있습니다.

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

EBS 볼륨 벤치마크

이 섹션에서는 I/O 워크로드를 시뮬레이트하여 Amazon EBS 볼륨의 성능을 테스트하는 방법을 보여줍니다. 프로세스는 다음과 같습니다.

1. EBS에 최적화된 인스턴스 시작.
2. 새 EBS 볼륨을 생성합니다.
3. EBS에 최적화된 인스턴스에 볼륨 추가.
4. 블록 디바이스를 구성하고 마운트합니다.
5. I/O 성능 벤치마크를 위한 도구 설치.
6. 볼륨의 I/O 성능 벤치마크.
7. 요금이 계속 발생하지 않도록 볼륨 삭제 및 인스턴스 종료.

Important

이 항목에 설명되어 있는 일부 절차를 수행할 경우 자신이 벤치마크하는 EBS 볼륨에 있는 기존 데이터가 소멸되는 결과를 낳게 됩니다. 벤치마킹 절차는 프로덕션 볼륨이 아니라 테스트 목적으로 특별히 생성된 볼륨에 적용하기 위한 것입니다.

인스턴스 설정

EBS 볼륨에서 최적의 성능을 얻으려면 EBS에 최적화된 인스턴스를 사용하는 것이 좋습니다. EBS에 최적화된 인스턴스는 인스턴스와 함께 Amazon EC2와 Amazon EBS 사이의 전용 처리량을 제공합니다. EBS에 최적화된 인스턴스는 Amazon EC2와 Amazon EBS 간에 전용 대역폭을 제공하며, 인스턴스 유형에 따라 500~12,000Mbps 범위에서 선택할 수 있습니다.

EBS에 최적화된 인스턴스를 생성하려면 Amazon EC2 콘솔을 사용하여 인스턴스를 시작할 때 [Launch as an EBS-Optimized instance]를 선택하거나 명령줄을 사용할 때 --ebs-optimized를 지정합니다. 이 옵션을 지원하는 현재 세대 인스턴스를 시작해야 합니다. 이 주제에 있는 예제 테스트의 경우 c3.4xlarge 인스턴스를 시작하는 것이 좋습니다. 자세한 내용은 [Amazon EBS 최적화 인스턴스 \(p. 614\)](#) 섹션을 참조하십시오.

프로비저닝된 IOPS SSD(io1) 볼륨 설정

io1 볼륨을 생성하려면, Amazon EC2 콘솔을 사용하여 볼륨 생성 시 [프로비저닝된 IOPS SSD]를 선택하거나 명령줄에서 --type io1 --iops n을 지정합니다. 여기서 n은 100~20,000 범위의 정수입니다. EBS 볼륨 생성에 대한 자세한 내용은 [Amazon EBS 볼륨 생성 \(p. 573\)](#) 섹션을 참조하십시오. 인스턴스에 이러한 볼륨 연결에 대한 자세한 내용은 [Amazon EBS 볼륨을 인스턴스에 연결 \(p. 576\)](#) 섹션을 참조하십시오.

예제 테스트의 경우 6개의 볼륨이 있는 RAID 어레이를 생성하는 것이 좋습니다. 이 어레이에는 높은 수준의 성능을 제공합니다. 볼륨 수가 아닌 프로비저닝된 기가바이트와 io1 볼륨에 대해 프로비저닝된 IOPS 수를 기준으로 요금이 부과되므로, 여러 개의 작은 볼륨을 생성하고 볼륨을 사용하여 스트라이프 세트를 생성하는 데 드는 추가 비용은 없습니다. Oracle Orion을 사용하여 볼륨을 벤치마크하는 경우 Oracle ASM과 동일한 방법으로 스트라이프를 시뮬레이트할 수 있으므로 Orion을 사용하여 스트라이프를 수행하는 것이 좋습니다. 다른 벤치마크 도구를 사용하는 경우 볼륨을 직접 스트라이프해야 합니다.

Amazon Linux에서 6개 볼륨 스트라이프 세트를 생성하려면 다음과 같은 명령을 사용합니다.

```
[ec2-user ~]$ sudo mdadm --create /dev/md0 --level=0 --chunk=64 --raid-devices=6 /dev/sdf /dev/sdg /dev/sdh /dev/sdi /dev/sdj /dev/sdk
```

이 예제의 경우 파일 시스템은 XFS입니다. 각자 요구 사항에 맞는 파일 시스템을 사용합니다. 다음 명령을 사용하여 XFS 파일 시스템 지원을 설치합니다.

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

그 다음에는 이 명령들을 사용하여 다음과 같이 XFS 파일 시스템을 생성 및 마운트하고 그 시스템에 대한 소유권을 할당합니다.

```
[ec2-user ~]$ sudo mkdir -p /mnt/p_iops_volo && sudo mkfs.xfs /dev/md0
[ec2-user ~]$ sudo mount -t xfs /dev/md0 /mnt/p_iops_volo
[ec2-user ~]$ sudo chown ec2-user:ec2-user /mnt/p_iops_volo/
```

처리량에 최적화된 HDD(st1) 또는 Cold HDD(sc1) 볼륨 설정

st1 볼륨을 생성하려면 Amazon EC2 콘솔을 사용하여 볼륨을 생성할 때 [처리량에 최적화된 HDD]를 선택하거나 명령줄을 사용할 때 `--type st1`을 지정합니다. sc1 볼륨을 생성하려면 Amazon EC2 콘솔을 사용하여 볼륨을 생성할 때 Cold HDD를 선택하거나 명령줄을 사용할 때 `--type sc1`을 지정합니다. EBS 볼륨 생성에 대한 자세한 내용은 [Amazon EBS 볼륨 생성 \(p. 573\)](#) 섹션을 참조하십시오. 인스턴스에 이러한 볼륨 연결에 대한 자세한 내용은 [Amazon EBS 볼륨을 인스턴스에 연결 \(p. 576\)](#) 섹션을 참조하십시오.

AWS는 AWS CloudFormation에 사용할 JSON 템플릿을 제공하여 이 설정 절차를 간소화합니다. [template](#)에 엑세스하고 이를 JSON 파일로 저장합니다. AWS CloudFormation에서는 사용자의 고유 SSH 키를 구성하고, 손쉽게 성능 테스트 환경을 설정하여 st1 볼륨을 평가할 수 있습니다. 템플릿은 현재 세대 인스턴스와 2TiB st1 볼륨을 생성하고, /dev/xvdf에서 볼륨을 인스턴스에 연결합니다.

템플릿을 사용하여 HDD 볼륨을 생성하려면

1. <https://console.aws.amazon.com/cloudformation/>에서 AWS CloudFormation 콘솔을 엽니다.
2. [Create Stack]을 선택합니다.
3. [Upload a Template to Amazon S3]를 선택하고 이전에 얻은 JSON 템플릿을 선택합니다.
4. 스택에 "ebs-perf-testing" 같은 이름을 붙이고 인스턴스 유형(기본은 r3.8xlarge)과 SSH 키를 선택합니다.
5. [Next]를 두 번 선택한 다음, [Create Stack]을 선택합니다.
6. 새로운 스택의 상태가 [CREATE_IN_PROGRESS]에서 [COMPLETE]로 바뀐 후에 [Outputs]를 선택하여 새 인스턴스의 퍼블릭 DNS 항목을 얻습니다. 그러면 2TiB st1 볼륨이 연결됩니다.
7. 이전 단계의 DNS 항목에서 얻은 호스트 이름을 통해 SSH를 사용하여 `ec2-user`라는 사용자로 새로운 스택에 연결합니다.
8. [벤치마크 도구 설치 \(p. 634\)](#) 항목으로 이동합니다.

벤치마크 도구 설치

다음 표에는 EBS 볼륨의 성능을 벤치마크하기 위해 사용할 수 있는 도구 중 일부가 나열되어 있습니다.

도구	설명
fio	I/O 성능을 벤치마크합니다. fio는 libaio-devel에 대해 종속성이 있습니다. Amazon Linux에 fio를 설치하려면 다음 명령을 실행합니다. [ec2-user ~]\$ sudo yum install -y fio

도구	설명
	Ubuntu에 fio를 설치하려면 다음 명령을 실행합니다. \$ sudo apt-get install -y fio
Oracle Orion 보정 도구	Oracle 데이터베이스와 함께 사용할 스토리지 시스템의 I/O 성능을 보정합니다.

이러한 벤치마크 도구는 다양한 테스트 파라미터를 지원합니다. 볼륨이 지원하는 작업에 근접하는 명령을 사용해야 합니다. 아래 제공된 명령은 사용자가 시작하는 데 도움이 되는 예시입니다.

볼륨 대기열 길이 선택

워크로드와 볼륨 유형에 따라 최적의 볼륨 대기열 길이를 선택합니다

SSD 지원 볼륨에서 대기열 길이

SSD 지원 볼륨의 워크로드에 대한 최적의 대기열 길이를 확인하려면 사용 가능한 모든 500 IOPS에 대해 대기열 길이를 1로 지정하는 것이 좋습니다(gp2 볼륨의 경우 기준 및 io1 볼륨의 경우 프로비저닝된 양). 그러면 애플리케이션 성능을 모니터링하고 애플리케이션 요구 사항을 기준으로 해당 값을 조정할 수 있습니다.

대기열 길이를 길게 하면 프로비저닝된 IOPS, 처리량 또는 최적 시스템 대기열 길이 값(현재 32로 설정)을 얻을 때까지 유용합니다. 예를 들어 프로비저닝된 IOPS가 1,000인 볼륨은 대기열 길이 2를 목표로 해야 합니다. 이 값을 높이거나 낮추면서 튜닝을 시도하여 애플리케이션에 가장 적합한 설정을 찾아야 합니다.

HDD 지원 볼륨에서 대기열 길이

HDD 지원 볼륨에서 워크로드에 가장 적합한 대기열 길이를 알아내려면 1MiB 순차 I/O를 수행하는 동시에 최소 4 이상의 대기열 길이를 목표로 하는 것이 좋습니다. 그러면 애플리케이션 성능을 모니터링하고 애플리케이션 요구 사항을 기준으로 해당 값을 조정할 수 있습니다. 예를 들어 버스트 처리량은 500MB/s, IOPS는 500인 2TiB st1 볼륨의 경우 1,024KiB, 512KiB 또는 256KiB 순차 I/O를 수행하는 동시에 각각 4, 8 또는 16 대기열 길이를 목표로 해야 합니다. 이 값을 높이거나 낮추면서 튜닝을 시도하여 애플리케이션에 가장 적합한 설정을 찾아야 합니다.

벤치마킹 수행

다음 절차에서는 다양한 EBS 볼륨 유형에 대한 벤치마킹 명령을 설명합니다.

연결된 EBS 볼륨이 있는 EBS에 최적화된 인스턴스에서 다음 명령을 실행합니다. 스냅샷에서 EBS 볼륨을 복원한 경우, 반드시 벤치마킹 전에 초기화합니다. 자세한 내용은 [Amazon EBS 볼륨 초기화 \(p. 628\)](#) 섹션을 참조하십시오.

볼륨 테스트를 마치면 정리에 도움이 되는 [Amazon EBS 볼륨 삭제 \(p. 589\)](#) 및 [인스턴스 종료 \(p. 291\)](#) 주제를 참조하십시오.

io1 볼륨 벤치마킹

생성한 스트라이프 세트에서 fio를 실행합니다.

다음 명령은 16KB 임의 쓰기 작업을 수행합니다.

```
[ec2-user ~]$ sudo fio --directory=/mnt/p_iops_volo \
--name fio_test_file --direct=1 --rw=randwrite --bs=16k --size=1G \
--numjobs=16 --time_based --runtime=180 --group_reporting --norandommap
```

다음 명령은 16KB 임의 읽기 작업을 수행합니다.

```
[ec2-user ~]$ sudo fio --directory=/mnt/p_iops_volo \
--name fio_test_file --direct=1 --rw=randread --bs=16k --size=1G \
--numjobs=16 --time_based --runtime=180 --group_reporting --norandommap
```

결과를 해석하는 방법에 대한 자세한 내용은 [Inspecting disk IO performance with fio](#) 자습서를 참조하십시오.

st1 및 sc1 볼륨 벤치마킹

st1 또는 sc1 볼륨에서 fio를 실행합니다.

Note

이러한 테스트를 실행하기 전, [st1 및 sc1에서 처리량이 높은 읽기 중심 워크로드의 미리 읽기 향상 \(p. 622\)](#)에 설명된 대로 인스턴스에 버퍼 I/O를 설정합니다.

다음 명령은 연결된 st1 블록 디바이스(예: /dev/xvdf)에 대해 1MiB 순차 읽기 작업을 수행합니다.

```
[ec2-user ~]$ sudo fio --filename=/dev/<device> --direct=1 --rw=read \
--randrepeat=0 --ioengine=libaio --bs=1024k --iodepth=8 --time_based=1 --runtime=180 \
--name=fio_direct_read_test
```

다음 명령은 연결된 st1 블록 디바이스에 대해 1MiB 순차 쓰기 작업을 수행합니다.

```
[ec2-user ~]$ sudo fio --filename=/dev/<device> --direct=1 --rw=write \
--randrepeat=0 --ioengine=libaio --bs=1024k --iodepth=8 --time_based=1 --runtime=180 \
--name=fio_direct_write_test
```

일부 워크로드는 블록 디바이스의 다양한 부분에 순차 읽기와 순차 쓰기를 혼합하여 수행합니다. 이러한 워크로드를 벤치마크하려면 읽기와 쓰기에 별도의 fio 작업을 동시에 사용하고, 각 작업에 대해 서로 다른 블록 디바이스 위치를 목표로 하기 위해 fio offset_increment 옵션을 사용하는 것이 좋습니다.

이 워크로드 실행은 순차 쓰기나 순차 읽기 워크로드보다 다소 복잡합니다. 텍스트 편집기를 사용하여 다음을 포함한 fio 작업 파일(이 예에서는 fio_rw_mix.cfg)을 만듭니다.

```
[global]
clocksource=clock_gettime
randrepeat=0
runtime=180
offset_increment=100g

[sequential-write]
bs=1M
ioengine=libaio
direct=1
iodepth=8
filename=/dev/<device>
do_verify=0
rw=write
rwmixread=0
rwmixwrite=100

[sequential-read]
bs=1M
ioengine=libaio
direct=1
iodepth=8
filename=/dev/<device>
do_verify=0
rw=read
rwmixread=100
```

```
rwmixwrite=0
```

그런 다음, 다음 명령을 실행합니다.

```
[ec2-user ~]$ sudo fio fio_rw_mix.cfg
```

결과를 해석하는 방법에 대한 자세한 내용은 [Inspecting disk I/O performance with fio](#) 자습서를 참조하십시오.

순차 읽기나 쓰기 작업을 사용하는 경우라 하더라도 직접 I/O에 대한 다수의 fio 작업은 `st1` 및 `sc1` 볼륨에 기대했던 처리량보다 낮은 수준을 나타낼 수 있습니다. 하나의 직접 I/O 작업을 사용하고 `iodepth` 파라미터를 사용해 동시 I/O 작업의 개수를 제어하는 것이 좋습니다.

Amazon EBS용 Amazon CloudWatch Events

Amazon EBS는 Amazon CloudWatch Events를 기반으로 다양한 스냅샷과 암호화 상태 변경에 대한 알림을 보냅니다. CloudWatch 이벤트에서는 스냅샷이나 암호화 키 상태 변경에 대한 응답으로 프로그래밍 방식의 작업을 트리거하는 규칙을 수립할 수 있습니다. 예를 들어 스냅샷이 생성되면 AWS Lambda 함수를 트리거하여 완료된 스냅샷을 다른 계정과 공유하거나 재해 복구를 위해 다른 리전으로 복사할 수 있습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 [Using Events](#) 섹션을 참조하십시오.

이벤트 정의와 예시

이 섹션에서는 지원되는 Amazon EBS 이벤트를 정의하고, 특정 시나리오에 대한 이벤트 출력의 예를 제공합니다. CloudWatch의 이벤트는 JSON 객체로 표현됩니다. 이벤트 객체의 형식과 내용에 대한 자세한 내용은 Amazon CloudWatch Events 사용 설명서의 [이벤트 및 이벤트 패턴](#)을 참조하십시오.

EBS 이벤트에 고유한 필드는 아래 표시된 JSON 객체의 "세부 정보" 섹션에 포함되어 있습니다. "이벤트" 필드에는 이벤트 이름이 포함됩니다. "결과" 필드에는 이벤트를 트리거한 작업의 완료 상태가 포함됩니다.

스냅샷 생성(`createSnapshot`)

스냅샷 생성 작업이 완료되면 `createSnapshot` 이벤트가 AWS 계정으로 전송됩니다. 이 이벤트에서 `succeeded` 또는 `failed` 결과가 발생할 수 있습니다.

이벤트 데이터

아래 목록에 성공적인 `createSnapshot` 이벤트에 대해 EBS가 발생시키는 JSON 객체의 예를 열거했습니다. `source` 필드에 소스 볼륨의 ARN이 포함됩니다. `StartTime` 및 `EndTime` 필드는 스냅샷 생성이 시작된 시점과 완료된 시점을 나타냅니다.

```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-012345678901",
    "detail-type": "EBS Snapshot Notification",
    "source": "aws.ec2",
    "account": "012345678901",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2::us-west-2:snapshot/snap-01234567"
    ],
    "detail": {
        "event": "createSnapshot",
        "result": "succeeded",
        "cause": "",
        "request-id": "",
        "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567",
        "source": "arn:aws:ec2::us-west-2:volume/vol-01234567",
    }
}
```

```
        "StartTime": "yyyy-mm-ddThh:mm:ssZ",
        "EndTime": "yyyy-mm-ddThh:mm:ssZ"  }
    }
```

스냅샷 복사(copySnapshot)

스냅샷 복사 작업이 완료되면 copySnapshot 이벤트가 AWS 계정으로 전송됩니다. 이 이벤트에서 succeeded 또는 failed 결과가 발생할 수 있습니다.

이벤트 데이터

아래 목록에 copySnapshot 이벤트 실패 이후 EBS가 발생시키는 JSON 객체의 예를 열거했습니다. 실패의 원인은 잘못된 소스 스냅샷 ID였습니다. snapshot_id의 값은 실패한 스냅샷의 ARN입니다. source의 값은 소스 스냅샷의 ARN입니다. StartTime 및 EndTime은 스냅샷 복사 작업이 시작된 시점과 종료된 시점을 나타냅니다.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2::us-west-2:snapshot/snap-01234567"
  ],
  "detail": {
    "event": "copySnapshot",
    "result": "failed",
    "cause": "Source snapshot ID is not valid",
    "request-id": "",
    "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567",
    "source": "arn:aws:ec2::eu-west-1:snapshot/snap-76543210",
    "StartTime": "yyyy-mm-ddThh:mm:ssZ",
    "EndTime": "yyyy-mm-ddThh:mm:ssZ"
  }
}
```

스냅샷 공유(shareSnapshot)

다른 계정이 스냅샷을 공유하면 shareSnapshot 이벤트가 AWS 계정으로 전송됩니다. 결과는 항상 succeeded입니다.

이벤트 데이터

아래 목록에 shareSnapshot 이벤트 완료 이후 EBS가 발생시키는 JSON 객체의 예를 열거했습니다. source의 값은 스냅샷을 공유한 사용자의 AWS 계정 번호입니다. StartTime 및 EndTime은 스냅샷 공유 작업이 시작된 시점과 종료된 시점을 나타냅니다. shareSnapshot 이벤트는 프라이빗 스냅샷이 다른 사용자와 공유될 때만 발생합니다. 퍼블릭 스냅샷 공유는 이벤트를 트리거하지 않습니다.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2::us-west-2:snapshot/snap-01234567"
  ],
}
```

```
"detail": {  
    "event": "shareSnapshot",  
    "result": "succeeded",  
    "cause": "",  
    "request-id": "",  
    "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567",  
    "source": "012345678901",  
    "StartTime": "yyyy-mm-ddThh:mm:ssZ",  
    "EndTime": "yyyy-mm-ddThh:mm:ssZ"  
}  
}
```

볼륨 연결 또는 다시 연결에 유효하지 않은 암호화 키(`attachVolume`, `reattachVolume`)

잘못된 KMS 키로 인해 인스턴스에 볼륨을 연결하거나 다시 연결하지 못하면 AWS 계정으로 `attachVolume` 이벤트가 전송됩니다.

Note

KMS 키를 사용해 EBS 볼륨을 암호화할 수 있습니다. 볼륨을 암호화하는 데 사용되는 키가 무효화되면, 이후 해당 키가 볼륨 생성, 연결 또는 재연결에 사용될 경우 EBS가 이벤트를 발생시킵니다.

이벤트 데이터

아래 목록에 `attachVolume` 이벤트 실패 이후 EBS가 발생시키는 JSON 객체의 예를 열거했습니다. 실패의 원인은 보류 중인 KMS 키의 삭제였습니다.

Note

AWS가 정기 서버 유지 관리 후 볼륨에 다시 연결을 시도할 수 있습니다.

```
{  
    "version": "0",  
    "id": "01234567-0123-0123-0123-0123456789ab",  
    "detail-type": "EBS Volume Notification",  
    "source": "aws.ec2",  
    "account": "012345678901",  
    "time": "yyyy-mm-ddThh:mm:ssZ",  
    "region": "us-east-1",  
    "resources": [  
        "arn:aws:ec2:us-east-1:0123456789ab:volume/vol-01234567",  
        "arn:aws:kms:us-east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab"  
    ],  
    "detail": {  
        "event": "attachVolume",  
        "result": "failed",  
        "cause": "arn:aws:kms:us-east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab  
is pending deletion.",  
        "request-id": ""  
    }  
}
```

아래 목록에 `reattachVolume` 이벤트 실패 이후 EBS가 발생시키는 JSON 객체의 예를 열거했습니다. 실패의 원인은 보류 중인 KMS 키의 삭제였습니다.

```
{  
    "version": "0",  
    "id": "01234567-0123-0123-0123-0123456789ab",  
    "detail-type": "EBS Volume Notification",  
    "source": "aws.ec2",  
    "account": "012345678901",  
    "time": "yyyy-mm-ddThh:mm:ssZ",  
    "region": "us-east-1",  
}
```

```
"resources": [  
    "arn:aws:ec2:us-east-1:0123456789ab:volume/vol-01234567",  
    "arn:aws:kms:us-east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab"  
],  
"detail": {  
    "event": "reattachVolume",  
    "result": "failed",  
    "cause": "arn:aws:kms:us-east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab  
is pending deletion.",  
    "request-id": ""  
}
```

볼륨 생성에 유효하지 않은 암호화 키(`createVolume`)

잘못된 KMS 키로 인해 볼륨 생성에 실패하면 `createVolume` 이벤트가 AWS 계정으로 전송됩니다.

Note

KMS 키를 사용해 EBS 볼륨을 암호화할 수 있습니다. 볼륨을 암호화하는 데 사용되는 키가 무효화되면, 이후 해당 키가 볼륨 생성, 연결 또는 재연결에 사용될 경우 EBS가 이벤트를 발생시킵니다.

이벤트 데이터

아래 목록에 `createVolume` 이벤트 실패 이후 EBS가 발생시키는 JSON 객체의 예를 열거했습니다. 실패의 원인은 비활성화된 KMS 키였습니다.

```
{  
    "version": "0",  
    "id": "01234567-0123-0123-0123-0123456789ab",  
    "detail-type": "EBS Volume Notification",  
    "source": "aws.ec2",  
    "account": "012345678901",  
    "time": "yyyy-mm-ddThh:mm:ssZ",  
    "region": "sa-east-1",  
    "resources": [  
        "arn:aws:ec2:sa-east-1:0123456789ab:volume/vol-01234567",  
    ],  
    "detail": {  
        "event": "createVolume",  
        "result": "failed",  
        "cause": "arn:aws:kms:sa-east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab  
is disabled.",  
        "request-id": "01234567-0123-0123-0123-0123456789ab",  
    }  
}
```

아래에 `createVolume` 이벤트 실패 이후 EBS가 발생시키는 JSON 객체의 예를 제시했습니다. 실패의 원인은 보류 중인 KMS 키 가져오기였습니다.

```
{  
    "version": "0",  
    "id": "01234567-0123-0123-0123-0123456789ab",  
    "detail-type": "EBS Volume Notification",  
    "source": "aws.ec2",  
    "account": "012345678901",  
    "time": "yyyy-mm-ddThh:mm:ssZ",  
    "region": "sa-east-1",  
    "resources": [  
        "arn:aws:ec2:sa-east-1:0123456789ab:volume/vol-01234567",  
    ],  
    "detail": {  
        "event": "createVolume",  
    }  
}
```

```
"result": "failed",
"cause": "arn:aws:kms:sa-east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab
is pending import.",
"request-id": "01234567-0123-0123-0123-0123456789ab",
}
```

Amazon Lambda를 이용한 CloudWatch 이벤트 처리

Amazon EBS와 CloudWatch 이벤트를 사용하여 데이터 백업 워크플로우를 자동화할 수 있습니다. 이를 위해 IAM 정책, 이벤트를 처리할 AWS Lambda 함수, 수신 이벤트와 일치하는 Amazon CloudWatch Events 규칙을 생성하고 Lambda 함수로 라우팅해야 합니다.

다음 절차에서는 재해 복구를 위해 `createSnapshot` 이벤트를 사용하여 완료된 스냅샷을 다른 리전으로 자동으로 복사합니다.

완료된 스냅샷을 다른 리전으로 복사하려면

1. 다음 예에 표시된 것과 같은 IAM 정책을 생성하여 `CopySnapshot` 작업을 실행하고 CloudWatch 이벤트 로그에 쓸 수 있는 권한을 제공합니다. CloudWatch 이벤트를 처리할 IAM 사용자에게 정책을 할당합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs>CreateLogGroup",
        "logs>CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CopySnapshot"
      ],
      "Resource": "*"
    }
  ]
}
```

2. CloudWatch 콘솔에서 사용할 수 있는 Lambda 함수를 정의합니다. 아래 Node.js로 작성된 샘플 Lambda 함수는 Amazon EBS가 일치하는 `createSnapshot` 이벤트를 발생시킬 때 CloudWatch에 의해 호출됩니다(스냅샷이 완료되었음을 의미). 호출되면 함수가 us-east-2에서 us-east-1로 스냅샷을 복사합니다.

```
// Sample Lambda function to copy an EBS snapshot to a different region

var AWS = require('aws-sdk');
var ec2 = new AWS.EC2();

// define variables
var destinationRegion = 'us-east-1';
var sourceRegion = 'us-east-2';
console.log ('Loading function');

//main function
exports.handler = (event, context, callback) => {
```

```
// Get the EBS snapshot ID from the CloudWatch event details
var snapshotArn = event.detail.snapshot_id.split('/');
const snapshotId = snapshotArn[1];
const description = `Snapshot copy from ${snapshotId} in ${sourceRegion}.`;
console.log ("snapshotId:", snapshotId);

// Load EC2 class and update the configuration to use destination region to
initiate the snapshot.
AWS.config.update({region: destinationRegion});
var ec2 = new AWS.EC2();

// Prepare variables for ec2.modifySnapshotAttribute call
const copySnapshotParams = {
    Description: description,
    DestinationRegion: destinationRegion,
    SourceRegion: sourceRegion,
    SourceSnapshotId: snapshotId
};

// Execute the copy snapshot and log any errors
ec2.copySnapshot(copySnapshotParams, (err, data) => {
    if (err) {
        const errorMessage = `Error copying snapshot ${snapshotId} to region
${destinationRegion}.`;
        console.log(errorMessage);
        console.log(err);
        callback(errorMessage);
    } else {
        const successMessage = `Successfully started copy of snapshot ${snapshotId}
to region ${destinationRegion}.`;
        console.log(successMessage);
        console.log(data);
        callback(null, successMessage);
    }
},);
});
```

Lambda 함수를 CloudWatch 콘솔에서 사용하도록 보장하려면 CloudWatch 이벤트가 발생하는 리전에서 생성합니다. 자세한 내용은 [AWS Lambda 개발자 가이드](#)를 참조하십시오.

3. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
4. [Events], [Create rule], [Select event source], [Amazon EBS Snapshots]을 선택합니다.
5. [Specific Event(s)]에서 [createSnapshot]을, [Specific Result(s)]에서 [succeeded]를 선택합니다.
6. [Rule target]에서 이전에 생성한 샘플 함수를 찾아 선택합니다.
7. [Target], [Add Target]을 선택합니다.
8. [Lambda function]에서 이전에 생성한 Lambda 함수를 선택하고 [Configure details]를 선택합니다.
9. [Configure rule details] 페이지에서 [Name] 및 [Description]에 값을 입력합니다. [State] 확인란을 선택하여 함수를 활성화합니다([Enabled]로 설정).
10. [Create rule]을 선택합니다.

이제 [Rules] 탭에 규칙이 표시됩니다. 표시된 예에서 구성된 이벤트는 다음에 스냅샷을 복사할 때 EBS가 발생시킵니다.

Amazon EC2 인스턴스 스토어

인스턴스 스토어는 인스턴스에 블록 수준의 임시 스토리지를 제공합니다. 스토리지는 호스트 컴퓨터에 물리적으로 연결된 디스크에 위치합니다. 인스턴스 스토어는 버퍼, 캐시, 스크래치 데이터 및 기타 임시 콘텐츠와

같이 자주 변경되는 정보의 임시 스토리지나 로드가 분산된 웹 서버 풀과 같은 여러 인스턴스상에서 복제되는 데이터에 가장 적합합니다.

하나 이상의 인스턴스 스토어 볼륨으로 구성된 인스턴스 스토어는 블록 디바이스로 표시됩니다. 인스턴스 스토어의 크기는 물론 사용 가능한 디바이스의 수는 인스턴스 유형에 따라 다릅니다. 어떤 인스턴스 스토어가 특정 인스턴스의 전용인 경우 호스트 컴퓨터의 인스턴스들이 디스크 하위 시스템을 공유합니다.

인스턴스 스토어 볼륨의 가상 디바이스는 `ephemeral[0-23]`입니다. 인스턴스 스토어 볼륨 1개를 지원하는 인스턴스 유형은 `ephemeral0`을 갖습니다. 인스턴스 스토어 볼륨 2개를 지원하는 인스턴스 유형은 `ephemeral0` 및 `ephemeral1` 등을 갖습니다.

NVMe 인스턴스 스토어 볼륨의 가상 디바이스는 `/dev/nvme[0-7]n1`입니다. NVMe 인스턴스 스토어 볼륨 1개를 지원하는 인스턴스 유형에는 `/dev/nvme0n1`이 있습니다. NVMe 인스턴스 스토어 볼륨 2개를 지원하는 인스턴스 유형에는 `/dev/nvme0n1`, `/dev/nvme1n1` 등이 있습니다.

목차

- [인스턴스 스토어 수명 \(p. 643\)](#)
- [인스턴스 스토리지 볼륨 \(p. 643\)](#)
- [EC2 인스턴스에 인스턴스 스토어 볼륨을 추가할 수 있습니다. \(p. 646\)](#)
- [SSD 인스턴스 스토어 볼륨 \(p. 649\)](#)
- [인스턴스 스토리지 스왑 볼륨 \(p. 651\)](#)
- [인스턴스 스토어 볼륨의 디스크 성능 최적화 \(p. 654\)](#)

인스턴스 스토어 수명

실행 시에만 인스턴스에 대한 인스턴스 스토어 볼륨을 지정할 수 있습니다. 하나의 인스턴스에서 인스턴스 스토어 볼륨을 분리하고 다른 인스턴스에 연결할 수 없습니다.

인스턴스 스토리지의 데이터는 관련 인스턴스의 수명 기간 동안만 지속됩니다. 인스턴스가 재부팅(의도적 또는 의도적이지 않게)되면 인스턴스 스토어의 데이터는 유지됩니다. 그러나 다음 상황에서는 인스턴스 스토어의 데이터가 손실됩니다.

- 기본 디스크 드라이브 오류
- 인스턴스 중지
- 인스턴스 종료

그러므로 중요한 장기 데이터의 경우 인스턴스 스토어에 의존하지 마십시오. 오히려 Amazon S3, Amazon EBS Amazon EFS 등 내구성이 뛰어난 데이터 스토리지를 사용하는 것이 좋습니다.

인스턴스를 중지하거나 종료하면 인스턴스 스토어의 모든 스토리지 블록이 리셋됩니다. 따라서 다른 인스턴스의 인스턴스 스토어를 통해 데이터를 액세스할 수 없습니다.

인스턴스에서 AMI를 생성한 경우 해당 인스턴스 스토어 볼륨의 데이터는 보존되지 않고 이 AMI를 실행한 인스턴스용 인스턴스 스토어 볼륨에 존재하지 않습니다.

인스턴스 스토리지 볼륨

인스턴스 유형은 사용 가능한 인스턴스 스토어의 크기와 인스턴스 스토어 볼륨에서 사용되는 하드웨어 유형을 결정합니다. 인스턴스 스토어 볼륨은 인스턴스의 시간당 비용의 일부로 포함됩니다. 인스턴스를 시작할 때 사용할 인스턴스 스토어 볼륨을 지정한 다음(기본적으로 사용할 수 있는 NVMe 인스턴스 스토어 볼륨은 제외), 사용하기에 앞서 이를 포맷하고 탑재해야 합니다. 인스턴스를 시작한 이후에는 사용 가능한 인스턴스

스토어 볼륨을 연결할 수 없습니다. 자세한 내용은 [EC2 인스턴스에 인스턴스 스토어 볼륨을 추가할 수 있습니다. \(p. 646\)](#) 섹션을 참조하십시오.

일부 인스턴스 유형은 NVMe 또는 SATA 기반 SSD(Solid State Drive)를 사용하여 매우 높은 랜덤 I/O 성능을 제공합니다. 이것은 지연시간이 매우 짧은 스토리지가 필요하지만 인스턴스가 종료될 경우에는 지속할 데이터가 필요가 없는 경우, 또는 내결함성 아키텍처를 활용할 수 있는 경우에 적합한 옵션입니다. 자세한 내용은 [SSD 인스턴스 스토어 볼륨 \(p. 649\)](#) 섹션을 참조하십시오.

다음 표에는 지원되는 각 인스턴스 유형에서 사용 가능한 인스턴스 스토어 볼륨의 수량, 크기, 유형 및 성능 최적화가 나와 있습니다. EBS 전용 유형을 포함한 전체 인스턴스 유형 목록을 보려면 [Amazon EC2 인스턴스 유형](#)을 참조하십시오.

인스턴스 유형	인스턴스 스토리지 볼륨	Type	초기화 필요*	TRIM 지원**
c1.medium	1 x 350GB†	HDD	✓	
c1.xlarge	4 x 420GB(1,680GB)	HDD	✓	
c3.large	2 x 16GB(32GB)	SSD	✓	
c3.xlarge	2 x 40GB(80GB)	SSD	✓	
c3.2xlarge	2 x 80GB(160GB)	SSD	✓	
c3.4xlarge	2 x 160GB(320GB)	SSD	✓	
c3.8xlarge	2 x 320GB(640GB)	SSD	✓	
cc2.8xlarge	4 x 840GB(3,360GB)	HDD	✓	
cg1.4xlarge	2 x 840GB(1,680GB)	HDD	✓	
cr1.8xlarge	2 x 120GB(240GB)	SSD	✓	
d2.xlarge	3 x 2,000GB(6TB)	HDD		
d2.2xlarge	6 x 2,000GB(12TB)	HDD		
d2.4xlarge	12 x 2,000GB(24TB)	HDD		
d2.8xlarge	24 x 2,000GB(48TB)	HDD		
g2.2xlarge	1 x 60GB	SSD	✓	
g2.8xlarge	2 x 120GB(240GB)	SSD	✓	
hi1.4xlarge	2 x 1,024GB(2,048GB)	SSD		
hs1.8xlarge	24 x 2,000GB(48TB)	HDD	✓	
i2.xlarge	1 x 800GB	SSD		✓

Amazon Elastic Compute Cloud
Linux 인스턴스용 사용 설명서
인스턴스 스토리지 볼륨

인스턴스 유형	인스턴스 스토리지 볼륨	Type	초기화 필요*	TRIM 지원**
i2.2xlarge	2 x 800GB(1,600GB)	SSD		✓
i2.4xlarge	4 x 800GB(3,200GB)	SSD		✓
i2.8xlarge	8 x 800GB(6,400GB)	SSD		✓
i3.large	1 x 475GB	NVMe SSD		✓
i3.xlarge	1 x 950GB	NVMe SSD		✓
i3.2xlarge	1 x 1,900GB	NVMe SSD		✓
i3.4xlarge	2 x 1,900GB(3.8TB)	NVMe SSD		✓
i3.8xlarge	4 x 1,900GB(7.6TB)	NVMe SSD		✓
i3.16xlarge	8 x 1,900GB(15.2TB)	NVMe SSD		✓
m1.small	1 x 160GB†	HDD	✓	
m1.medium	1 x 410GB	HDD	✓	
m1.large	2 x 420GB(840GB)	HDD	✓	
m1.xlarge	4 x 420GB(1,680GB)	HDD	✓	
m2.xlarge	1 x 420GB	HDD	✓	
m2.2xlarge	1 x 850GB	HDD	✓	
m2.4xlarge	2 x 840GB(1,680GB)	HDD	✓	
m3.medium	1 x 4GB	SSD	✓	
m3.large	1 x 32GB	SSD	✓	
m3.xlarge	2 x 40GB(80GB)	SSD	✓	
m3.2xlarge	2 x 80GB(160GB)	SSD	✓	
r3.large	1 x 32GB	SSD		✓
r3.xlarge	1 x 80GB	SSD		✓
r3.2xlarge	1 x 160GB	SSD		✓
r3.4xlarge	1 x 320GB	SSD		✓
r3.8xlarge	2 x 320GB(640GB)	SSD		✓
x1.16xlarge	1 x 1,920GB	SSD		

인스턴스 유형	인스턴스 스토리지 볼륨	Type	초기화 필요*	TRIM 지원**
x1.32xlarge	2 x 1,920GB(3,840GB)	SSD		

* 특정 인스턴스에 연결된 볼륨은 초기화되지 않을 경우 최초 쓰기 페널티를 받게 됩니다. 자세한 내용은 [인스턴스 스토어 볼륨의 디스크 성능 최적화 \(p. 654\)](#) 섹션을 참조하십시오.

** SSD 기반 인스턴스 스토리지 볼륨(TRIM 명령 지원)은 어떤 파일 시스템으로도 사전 포맷되지 않습니다. 그러나 인스턴스를 시작한 후 볼륨을 원하는 파일 시스템으로 포맷할 수 있습니다. 자세한 내용은 [인스턴스 스토어 볼륨 TRIM 지원 \(p. 649\)](#) 섹션을 참조하십시오.

† 또한, c1.medium 및 m1.small 인스턴스 유형에는 900MB 인스턴스 스토어 스왑 볼륨(부팅 시점에 자동 활성화되지 않음)이 포함됩니다. 자세한 내용은 [인스턴스 스토리지 스왑 볼륨 \(p. 651\)](#) 섹션을 참조하십시오.

EC2 인스턴스에 인스턴스 스토어 볼륨을 추가할 수 있습니다.

블록 디바이스 매핑을 사용하여 인스턴스에 대한 EBS 볼륨 및 인스턴스 스토어 볼륨을 지정합니다. 블록 디바이스 매핑의 각 항목은 디바이스 이름 및 매핑된 볼륨을 포함합니다. 기본 블록 디바이스 매핑은 사용하는 AMI에 의해 지정됩니다. 또는 시작 시 인스턴스에 대한 블록 디바이스 매핑을 지정할 수 있습니다. 인스턴스 유형에서 지원되는 모든 NVMe 인스턴스 스토어 볼륨이 인스턴스 시작 시 자동으로 추가됩니다. 따라서 AMI 또는 인스턴스에 대한 블록 디바이스 매핑에 추가할 필요가 없습니다. 자세한 내용은 [블록 디바이스 매핑 \(p. 662\)](#) 섹션을 참조하십시오.

블록 디바이스 매핑은 항상 인스턴스에 대한 루트 볼륨을 지정합니다. 루트 볼륨은 Amazon EBS 볼륨 또는 인스턴스 스토어 볼륨 중 하나입니다. 자세한 내용은 [루트 디바이스 스토리지 \(p. 64\)](#) 섹션을 참조하십시오. 루트 볼륨은 자동으로 마운트됩니다. 루트 볼륨에 대한 인스턴스 스토어 볼륨이 있는 인스턴스의 경우, 볼륨의 크기는 AMI에 따라 다르지만 최대 크기는 10GB입니다.

블록 디바이스 매핑을 사용하면 인스턴스를 실행할 때 인스턴스에 연결할 추가 EBS 볼륨을 지정하거나 인스턴스가 실행된 후에 추가 EBS 볼륨을 연결할 수 있습니다. 자세한 내용은 [Amazon EBS 볼륨 \(p. 562\)](#) 섹션을 참조하십시오.

인스턴스 실행 시에만 인스턴스에 대한 인스턴스 스토어 볼륨을 지정할 수 있습니다. 인스턴스를 실행한 이후에는 인스턴스 스토어 볼륨을 연결할 수 없습니다.

이러한 볼륨의 개수 및 크기는 인스턴스 유형에 따라 다른 인스턴스에서 사용 가능한 인스턴스 스토어 볼륨을 초과하지 않아야 합니다. 일부 인스턴스 유형은 인스턴스 스토어 볼륨을 지원하지 않습니다. 인스턴스 유형별 인스턴스 스토어 볼륨 지원에 대한 자세한 내용은 [인스턴스 스토리지 볼륨 \(p. 643\)](#) 섹션을 참조하십시오. 인스턴스에 대해 선택한 인스턴스 유형이 인스턴스 스토어 볼륨을 지원하는 경우 인스턴스를 실행할 때 인스턴스에 대한 블록 디바이스 매핑에 추가해야 합니다. 인스턴스를 실행한 후에는 인스턴스에 대한 인스턴스 스토어 볼륨이 사용하기에 앞서 포맷되고 마운트되었는지 확인해야 합니다. 인스턴스 스토어 지원 인스턴스의 루트 볼륨은 기본적으로 마운트됩니다.

목차

- [AMI에 인스턴스 스토어 볼륨 추가 \(p. 647\)](#)
- [인스턴스에 인스턴스 스토어 볼륨 추가 \(p. 647\)](#)
- [인스턴스 스토어 볼륨을 인스턴스에서 사용 가능하도록 만들기 \(p. 648\)](#)

AMI에 인스턴스 스토어 볼륨 추가

인스턴스 스토어 볼륨을 포함하는 블록 디바이스 매핑으로 AMI를 생성할 수 있습니다. AMI에 인스턴스 스토어 볼륨을 추가한 이후에는 AMI에서 실행된 모든 인스턴스에는 이러한 인스턴스 스토어 볼륨이 추가됩니다. 인스턴스를 실행할 때 AMI 블록 디바이스 매핑에서 지정된 볼륨을 생략하고 새 볼륨을 추가할 수 있습니다.

Important

M3 인스턴스의 경우, AMI가 아니라 인스턴스의 블록 디바이스 매핑을 사용하여 인스턴스 스토어 볼륨을 지정합니다. Amazon EC2가 AMI의 블록 디바이스 매핑에서만 지정된 인스턴스 스토어 볼륨을 무시할 수 있습니다.

콘솔을 사용하여 Amazon EBS 지원 AMI에 인스턴스 스토어 볼륨을 추가하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Instances]를 선택하고 인스턴스를 선택합니다.
3. [Actions], [Image], [Create Image]를 차례로 선택합니다.
4. [Create Image] 대화 상자에서 이미지의 이름 및 설명을 추가합니다.
5. 추가할 각 인스턴스 스토어 볼륨에서 [Add New Volume]을 선택하고 [Type]에서 인스턴스 스토어 볼륨을 선택한 다음 [Device]에서 디바이스 이름을 선택합니다. 자세한 내용은 [Linux 인스턴스의 디바이스 명명 \(p. 660\)](#) 섹션을 참조하십시오. 사용할 수 있는 인스턴스 스토어 볼륨의 개수는 인스턴스 유형에 따라 다릅니다. NVMe 인스턴스 스토어 볼륨이 있는 인스턴스의 경우, 이러한 볼륨의 디바이스 매핑은 운영 체제가 볼륨을 열거하는 순서에 따라 다릅니다.
6. Create Image를 선택합니다.

명령줄을 사용하여 AMI에 인스턴스 스토어 볼륨을 추가하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- `create-image` 또는 `register-image` (AWS CLI)
- `New-EC2Image` 및 `Register-EC2Image`(Windows PowerShell용 AWS 도구)

인스턴스에 인스턴스 스토어 볼륨 추가

인스턴스를 실행할 때 기본 블록 디바이스 매핑은 지정된 AMI에 의해 제공됩니다. 추가 인스턴스 스토어 볼륨이 필요할 경우 실행할 때 인스턴스에 추가해야 합니다. AMI 블록 디바이스 매핑에서 지정된 디바이스는 생략할 수도 있습니다.

Important

M3 인스턴스의 경우, 인스턴스의 블록 디바이스 매핑에서 지정하지 않더라도 인스턴스 스토어 볼륨을 받을 수 있습니다.

Important

HS1 인스턴스의 경우, 사용자가 AMI의 블록 디바이스 매핑에서 몇 개의 인스턴스 스토어 볼륨을 지정하더라도 AMI으로부터 시작된 인스턴스의 블록 디바이스 매핑이 지원되는 최대 개수의 인스턴스 스토어 볼륨을 자동으로 포함합니다. 인스턴스를 시작하기 전에 해당 인스턴스의 블록 디바이스 매핑에서 원치 않는 인스턴스 스토어 볼륨을 명시적으로 제거해야 합니다.

콘솔을 사용하여 인스턴스에 대한 블록 디바이스 매핑을 업데이트하려면

1. Amazon EC2 콘솔을 엽니다.

2. 대시보드에서 [Launch Instance]를 선택합니다.
3. [Step 1: Choose an Amazon Machine Image (AMI)]에서 사용할 AMI를 선택하고 [Select]를 선택합니다.
4. 마법사를 따라 [Step 1: Choose an Amazon Machine Image (AMI)], [Step 2: Choose an Instance Type] 및 Step 3: Configure Instance Details를 완료합니다.
5. [Step 4: Add Storage]에서 필요에 따라 기존 항목을 수정합니다. 추가할 각 인스턴스 스토어 볼륨에서 [Add New Volume]을 클릭한 다음 [Type] 목록에서 인스턴스 스토어 볼륨을 선택하고 [Device]에서 디바이스 이름을 선택합니다. 사용할 수 있는 인스턴스 스토어 볼륨의 개수는 인스턴스 유형에 따라 다릅니다.
6. 마법사를 완료해 인스턴스를 시작합니다.

명령줄을 사용하여 인스턴스에 대한 블록 디바이스 매핑을 업데이트하려면

해당 명령과 함께 다음 옵션 명령 중 하나를 사용할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- `--block-device-mappings` with [run-instances](#) (AWS CLI)
- `-BlockDeviceMapping` with [New-EC2Instance](#) (Windows PowerShell용 AWS 도구)

인스턴스 스토어 볼륨을 인스턴스에서 사용 가능하도록 만들기

인스턴스를 시작한 후에 인스턴스 스토어 볼륨은 인스턴스에서 사용 가능하지만 마운트 이후에 액세스할 수 있습니다. Linux 인스턴스의 경우 인스턴스 유형에 따라 어느 인스턴스 스토어 볼륨이 마운트되고 어느 것을 마운트할 수 있는지가 결정됩니다. Windows 인스턴스의 경우 EC2Config 서비스가 인스턴스에서 대한 인스턴스 스토리지 볼륨을 마운트합니다. 인스턴스의 블록 디바이스 드라이버는 볼륨이 마운트될 때 실제 볼륨 이름을 할당하고 할당된 이름은 Amazon EC2 권장 이름과 다를 수 있습니다.

여러 인스턴스 스토리지 볼륨은 ext3 파일 시스템으로 사전 포맷됩니다. SSD 기반 인스턴스 스토리지 볼륨 (TRIM 지원)은 어떤 파일 시스템으로도 사전 포맷되지 않습니다. 그러나 인스턴스를 시작한 후 볼륨을 원하는 파일 시스템으로 포맷할 수 있습니다. 자세한 내용은 [인스턴스 스토어 볼륨 TRIM 지원 \(p. 649\)](#) 섹션을 참조하십시오. Windows 인스턴스의 경우 EC2Config 서비스가 NTFS 파일 시스템으로 인스턴스 스토리지 볼륨을 다시 포맷합니다.

인스턴스에서 인스턴스 스토어 디바이스를 사용할 수 있는지의 여부는 인스턴스 메타데이터를 사용하여 확인할 수 있습니다. 자세한 내용은 [인스턴스 스토어 볼륨용 인스턴스 블록 디바이스 매핑 보기 \(p. 669\)](#) 섹션을 참조하십시오.

Windows 인스턴스의 경우 Windows 디스크 관리를 사용하여 인스턴스 스토어 볼륨을 볼 수도 있습니다. 자세한 내용은 [Windows 디스크 관리를 이용하여 디스크 나열](#)을 참조하십시오.

Linux 인스턴스의 경우 다음 절차에서 설명한 대로 인스턴스 스토어 볼륨을 보고 마운트할 수 있습니다.

Linux에서 인스턴스 스토어 볼륨을 사용 가능하게 만들려면

1. SSH 클라이언트를 사용하여 인스턴스에 연결합니다.
2. `df -h` 명령을 사용하여 포맷되고 마운트된 볼륨을 봅니다. `lsblk`을 사용하여 시작 시에 매핑되지 않았지만 포맷되고 마운트된 볼륨을 봅니다.
3. 매핑된 인스턴스 스토어 볼륨만 포맷하고 마운트하려면 다음을 수행합니다.
 - a. `mkfs` 명령을 사용하여 디바이스에서 파일 시스템을 생성합니다.
 - b. `mkdir` 명령을 사용하여 디바이스를 마운트할 디렉터리를 생성합니다.
 - c. `mount` 명령을 사용하여 새로 생성한 디렉터리에 디바이스를 마운트합니다.

SSD 인스턴스 스토어 볼륨

C3, G2, HI1, I2, I3, M3, R3, X1 인스턴스는 SSD(Solid State Drive)를 사용하여 매우 높은 랜덤 I/O 성능을 제공하는 인스턴스 스토어 볼륨을 지원합니다. 인스턴스 유형별 인스턴스 스토어 볼륨 지원에 대한 자세한 내용은 [인스턴스 스토리지 볼륨 \(p. 643\)](#) 섹션을 참조하십시오.

Linux의 SSD 인스턴스 스토어 볼륨에 최상의 IOPS 성능을 보장하려면 [Amazon Linux AMI](#)의 최신 버전을 사용하거나 커널 버전이 3.8 이상인 기타 Linux AMI를 사용하는 것이 좋습니다. 커널 버전이 3.8 이상인 Linux AMI를 사용하지 않는 경우 사용자의 인스턴스는 해당 인스턴스 유형에 제공되는 최대 IOPS 성능을 달성할 수 없습니다.

다른 인스턴스 스토어 볼륨과 마찬가지로 인스턴스를 시작할 때 인스턴스에 대해 SSD 인스턴스 스토어 볼륨을 매핑해야 하며, SSD 인스턴스 볼륨의 데이터는 연결된 인스턴스의 수명만큼만 지속됩니다. 자세한 내용은 [EC2 인스턴스에 인스턴스 스토어 볼륨을 추가할 수 있습니다. \(p. 646\)](#) 섹션을 참조하십시오.

NVMe SSD 볼륨

I3 인스턴스는 NVMe(Non-Volatile Memory Express) SSD 인스턴스 스토어 볼륨을 제공합니다. NVMe 볼륨에 액세스하려면 NVMe를 지원하는 운영 체제를 사용해야 합니다. 다음은 최소 운영 체제 요구 사항입니다.

- 현재 Amazon Linux AMI
- Ubuntu 버전 16.10 또는 버전 16.04 LTS. 버전 14.04에는 권장하지 않는 이전 버전의 NVMe가 있습니다.
- CentOS 버전 7
- SUSE Linux Enterprise Server 버전 12 또는 SP3가 있는 버전 11
- Windows Server 2016, Windows Server 2012 R2 또는 Windows Server 2008 R2. Windows Server 2012와 Windows Server 2008은 지원되지 않습니다.

인스턴스에 연결한 후 `lspci` 명령을 사용하여 NVMe 디바이스를 나열할 수 있습니다. 다음은 NVMe 디바이스 4개를 지원하는 i3.8xlarge 인스턴스의 예제 출력입니다.

```
[ec2-user ~]$ lspci
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371SB PIIX3 IDE [Natoma/Triton II]
00:01.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 01)
00:02.0 VGA compatible controller: Cirrus Logic GD 5446
00:03.0 Ethernet controller: Device 1d0f:ec20
00:17.0 Non-Volatile memory controller: Device 1d0f:cd01
00:18.0 Non-Volatile memory controller: Device 1d0f:cd01
00:19.0 Non-Volatile memory controller: Device 1d0f:cd01
00:1a.0 Non-Volatile memory controller: Device 1d0f:cd01
00:1f.0 Unassigned class [ff80]: XenSource, Inc. Xen Platform Device (rev 01)
```

지원되는 운영 체제를 사용하지만 NVMe 디바이스가 보이지 않는 경우, NVMe 모듈이 다음 `lsmod` 명령을 사용하여 로드되었는지 확인하십시오.

```
[ec2-user ~]$ lsmod | grep nvme
nvme               48813   0
```

NVMe 볼륨은 NVMe 1.0a 사양을 준수합니다. NVMe 볼륨에 NVMe 명령을 사용할 수 있습니다. Amazon Linux AMI에서는 `yum install` 명령을 사용하여 리포지토리에서 `nvme-cli` 패키지를 설치할 수 있습니다. 지원되는 다른 Linux 버전에서는 `nvme-cli` 패키지가 이미지에 제공되지 않은 경우 다운로드할 수 있습니다.

인스턴스 스토어 볼륨 TRIM 지원

I2, I3, R3 인스턴스는 TRIM을 통해 SSD 볼륨을 지원합니다.

TRIM을 지원하는 인스턴스 스토어 볼륨을 사용할 경우 TRIM 명령을 사용하여 작성한 데이터가 더 이상 필요하지 않음을 SSD 컨트롤러에 통지할 수 있습니다. 이를 통해 컨트롤러에 더 많은 여유 공간이 제공되므로 쓰기 작업 증폭을 줄이고 성능을 향상 시킬 수 있습니다. TRIM 명령에 대한 자세한 내용은 인스턴스에 대한 운영 체제 관련 문서를 참조하십시오.

TM을 지원하는 인스턴스 스토어 볼륨은 인스턴스에 할당되기 전 완전히 트리밍(trimming)됩니다. 이러한 볼륨은 인스턴스가 실행될 때 파일 시스템으로 포맷되지 않으므로, 마운트 후 사용하기 전 사용자가 해당 볼륨을 포맷해야 합니다. 이러한 볼륨에 빠르게 액세스하려면 포맷 시 TRIM 작업을 건너뛰는 파일 시스템별 옵션을 지정해야 합니다. Linux에서는 마운트 명령에 `discard` 옵션을 추가하거나 TRIM을 지원하는 디바이스에 `/etc/fstab` 파일 입력을 추가해야 이 기능을 효과적으로 이용할 수 있습니다. Windows에서는 `fsutil behavior set DisableDeleteNotify 1` 명령을 사용합니다.

Linux에서 인스턴스 스토어를 TRIM 지원을 사용할 수 있도록 만들려면

1. 인스턴스를 실행할 때 인스턴스 스토어 볼륨을 매핑합니다. 자세한 내용은 [EC2 인스턴스에 인스턴스 스토어 볼륨을 추가할 수 있습니다. \(p. 646\)](#) 섹션을 참조하십시오.
2. 인스턴스에서 `lsblk` 명령을 사용해 사용 가능한 디바이스 목록을 나열하거나 [인스턴스 메타데이터를 사용해 인스턴스 스토어 볼륨을 표시합니다 \(p. 669\)](#).
3. 다음 명령을 사용해 운영 체제와 디바이스가 TRIM을 지원하는지 확인합니다(`xvdb`를 디바이스 이름으로 변경).

```
[ec2-user ~]$ sudo cat /sys/block/xvdb/queue/discard_max_bytes
322122547200
```

이 명령을 실행한 결과 0보다 큰 값이 반환되면 운영 체제와 디바이스가 TRIM을 지원하는 것입니다.

4. 볼륨을 원하는 파일 시스템으로 포맷합니다.
 - (EXT4) `ext4` 파일 시스템으로 볼륨을 포맷하려면 다음 명령을 사용합니다(`xvdc`를 디바이스 이름으로 변경).

```
[ec2-user ~]$ sudo mkfs.ext4 -E nodiscard /dev/xvdc
```

- (XFS) `xfs` 파일 시스템으로 볼륨을 포맷하려면 다음 명령을 사용합니다(`xvdb`를 디바이스 이름으로 변경).

```
[ec2-user ~]$ sudo mkfs.xfs -K /dev/xvdb
```

Note

이 명령을 실행하려면 운영 체제에 XFS 파일 시스템 support를 설치해야 합니다. Amazon Linux의 경우 `sudo yum install -y xfsprogs` 명령을 사용합니다.

5. `discard` 옵션을 사용하여 디바이스를 마운트합니다. 볼륨의 디바이스 이름을 지정해야 합니다. 기존 디렉터리를 선택하거나 `mkdir` 명령으로 새 디렉터리를 생성할 수 있습니다.

```
[ec2-user ~]$ sudo mount -o discard /dev/xvdb /mnt/my-data
```

6. (선택 사항) 부팅 시점에 디바이스를 마운트하려면 `## ## /etc/fstab` 파일의 항목을 추가 또는 수정할 수 있습니다.

```
/dev/xvdb    /mnt/xvdb    xfs    defaults,nofail,discard    0    2
/dev/xvdc    /mnt/xvdc    ext4   defaults,nofail,discard    0    2
```

Important

`/etc/fstab` 파일을 편집한 후 `sudo mount -a` 명령을 실행하여 오류가 없는지 확인합니다. 파일에 오류가 있는 경우 시스템이 정상적으로 부팅되지 않거나 아예 부팅되지 않을 수 있습니다.

HI1 SSD 스토리지

HI1 인스턴스에서 SSD 스토리지 사용:

- 메인 데이터 소스는 SSD 스토리지를 사용하는 인스턴스 스토어입니다.
- 읽기 성능은 일관적이지만 쓰기 성능은 변동될 수 있습니다.
- 쓰기 증폭(write amplification)이 발생할 수 있습니다.
- TRIM 명령은 현재 지원되지 않습니다.

인스턴스 스토어와 SSD 스토리지

hi1.4xlarge 인스턴스는 Amazon EBS 기반의 루트 디바이스를 사용합니다. 그러나 메인 데이터 스토리지는 인스턴스 스토어의 SSD 볼륨으로 제공됩니다. 이러한 인스턴스 스토어 볼륨은 다른 인스턴스 스토어 볼륨과 마찬가지로 인스턴스의 수명 동안만 유지됩니다. hi1.4xlarge 인스턴스의 루트 디바이스가 Amazon EBS 기반이므로 인스턴스를 시작하거나 중지하는 것은 가능합니다. 인스턴스를 중단하면 애플리케이션은 유지되지만 인스턴스 스토어의 프로덕션 데이터는 유지되지 않습니다. 인스턴스 스토어 볼륨에 대한 자세한 내용은 [Amazon EC2 인스턴스 스토어 \(p. 642\)](#) 섹션을 참조하십시오.

쓰기 성능 변동

쓰기 성능은 애플리케이션에서 논리 블록 어드레스(LBA) 공간을 어떻게 활용하는지에 따라 달라집니다. 애플리케이션에서 전체 LBA 공간을 사용한다면 쓰기 속도가 90% 가량 저하될 수 있습니다. 애플리케이션에 대해 벤치마크를 수행하고 대기열 길이(queue length: 한 볼륨에서 대기 중인 I/O 요청 수)와 I/O 크기를 모니터링합니다.

쓰기 증폭

쓰기 증폭(write amplification)은 플래시 메모리와 SSD에서 발생하는 이상 현상으로, 실제로 기록된 물리적 데이터의 양이 원래 기록하고자 했던 논리적 양의 배수에 달하는 경우입니다. 플래시 메모리는 특성 상 데이터를 재작성하기 전에 삭제가 필요한데, 이 작업을 수행하는 과정에서 대상 사용자 데이터와 메타 데이터가 한 번 이상 이동(또는 재작성)됩니다. 이러한 배수 효과 때문에 SSD를 사용하는 동안 필요한 쓰기 횟수가 늘어나고 결국 안정적인 작동 기간이 단축되는 것입니다. hi1.4xlarge 인스턴스는 쓰기 증폭을 최소화하기 위한 프로비저닝 모델로 설계되었습니다.

랜덤 쓰기는 연속 쓰기보다 쓰기 증폭에 훨씬 더 심각한 영향을 미칩니다. 쓰기 증폭을 우려하는 사용자라면 테비바이트보다 적은 스토리지 용량을 애플리케이션에 할당하는 것이 좋습니다(오버 프로비저닝).

TRIM 명령

TRIM 명령은 운영 시스템에서 기존 저장 데이터 중 더 이상 사용하지 않는 것으로 간주되는 데이터 블록이 있음을 SSD에 알립니다. TRIM 기능은 쓰기 증폭의 영향을 제한합니다.

HI1 인스턴스는 TRIM을 지원하지 않습니다. TRIM을 지원하는 인스턴스에 대한 자세한 내용은 [인스턴스 스토어 볼륨 TRIM 지원 \(p. 649\)](#) 항목을 참조하십시오.

인스턴스 스토리지 스왑 볼륨

Linux에서 스왑 공간은 물리적으로 할당된 것보다 더 큰 메모리가 시스템에 필요할 때 사용될 수 있습니다. 스왑 공간이 활성화되면 Linux 시스템은 물리 메모리에서 자주 사용되지 않는 메모리 페이지를 스왑 공간(기존 파일 시스템의 스왑 파일 또는 전용 파티션)으로 스왑하고 고속 액세스가 필요한 메모리 페이지용으로 해당 공간을 해제합니다.

Note

메모리 페이지용으로 스왑 공간을 사용하는 것은 RAM을 사용하는 것보다 빠르거나 효율적이지 않습니다. 워크로드가 메모리를 스왑 공간으로 정기적으로 페이징하는 경우 큰 RAM 용량을 갖는

대형 인스턴스 유형으로 마이그레이션할 것을 고려해야 합니다. 자세한 내용은 [인스턴스 크기 조정 \(p. 169\)](#) 섹션을 참조하십시오.

c1.medium 및 m1.small 인스턴스 유형의 물리적 메모리 공간은 제한적이고 실행 시 Linux AMI용 가상 메모리로 사용할 수 있는 스왑 공간으로 900MiB가 제공됩니다. 비록 Linux 커널은 이 스왑 공간을 루트 디바이스의 한 파티션으로 인식하지만 그것은 루트 디바이스 유형과 상관없이 실제로 별도의 인스턴스 스토어 볼륨입니다.

Amazon Linux AMI는 이 스왑 공간을 자동으로 활성화 및 사용하지만 사용자의 AMI에서 이 스왑 공간을 인식 및 사용하기 위해서는 추가적인 몇 단계가 필요합니다. 인스턴스에서 스왑 공간이 사용되는지를 확인하려면 swapon -s 명령을 사용합니다.

```
[ec2-user ~]$ swapon -s
Filename          Type      Size   Used   Priority
/dev/xvda3        partition 917500  0       -1
```

위 인스턴스에서는 900MiB의 스왑 볼륨이 연결 및 활성화되었습니다. 이 명령을 수행했는데 스왑 볼륨이 표시되지 않는 경우 디바이스에서 스왑 공간을 활성화해야 합니다. lsblk 명령을 사용하여 가용 디스크를 확인합니다.

```
[ec2-user ~]$ lsblk
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda1 202:1    0    8G  0 disk /
xvda3 202:3    0  896M 0 disk
```

여기에서 인스턴스는 스왑 볼륨 xvda3를 사용할 수 있지만 해당 볼륨은 활성화되지 않은 상태입니다 (MOUNTPOINT 필드가 공란임). swapon 명령을 사용하면 스왑 볼륨을 활성화할 수 있습니다.

Note

lsblk를 사용하여 디바이스 이름 앞에 /dev/를 추가해야 합니다. 사용자 디바이스는 sda3, sde3 또는 xvde3 등으로 다르게 명명할 수 있습니다. 아래 명령에서 시스템의 디바이스 이름을 사용합니다.

```
[ec2-user ~]$ sudo swapon /dev/xvda3
```

이제 lsblk 및 swapon -s 출력에 스왑 공간이 표시되어야 합니다.

```
[ec2-user ~]$ lsblk
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda1 202:1    0    8G  0 disk /
xvda3 202:3    0  896M 0 disk [SWAP]
[ec2-user@ip-12-34-56-78 ~]$ swapon -s
Filename          Type      Size   Used   Priority
/dev/xvda3        partition 917500  0       -1
```

또한, /etc/fstab 파일을 편집하여 부팅 시마다 이 스왑 공간이 자동 활성화되도록 설정해야 합니다.

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

/etc/fstab 파일에 다음 명령을 추가합니다(시스템의 스왑 디바이스 이름 사용):

```
/dev/xvda3      none     swap     sw  0      0
```

인스턴스 스토어 볼륨을 스왑 공간으로 사용하려면

모든 인스턴스 스토어 볼륨은 스왑 공간으로 사용될 수 있습니다. 예를 들어, m3.medium 인스턴스 유형은 스왑 공간으로 적당한 4GB SSD 인스턴스 스토어 볼륨이 포함됩니다. 사용자의 인스턴스 스토어 볼륨이 훨씬

큰(예: 350GB) 경우 해당 볼륨을 4-8GB의 작은 스왑 파티션으로 나누고 나머지는 데이터 볼륨으로 사용할 수 있습니다.

Note

이 절차는 인스턴스 스토리지를 지원하는 인스턴스 유형에만 적용됩니다. 지원되는 인스턴스 유형의 목록은 [인스턴스 스토리지 볼륨 \(p. 643\)](#) 섹션을 참조하십시오.

1. 인스턴스에 연결된 블록 디바이스 목록을 확인하여 인스턴스 스토어 볼륨에 사용할 디바이스 이름을 엽니다.

```
[ec2-user ~]$ lsblk -p
NAME      MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
/dev/xvdb  202:16   0  4G  0 disk /media/ephemeral0
/dev/xvda1 202:1    0  8G  0 disk /
```

이 예제에서 인스턴스 스토어 볼륨은 /dev/xvdb입니다. Amazon Linux 인스턴스이기 때문에 인스턴스 스토어 볼륨은 포맷된 후 /media/ephemeral0에 마운트됩니다. 모든 Linux 운영 체제에서 이러한 과정이 자동으로 수행되는 것은 아닙니다.

2. (선택 사항) 인스턴스 스토어 볼륨이 마운트되면(MOUNTPOINT lsblk 명령의 출력에서 로 목록 표시) 다음 명령으로 마운트를 해제해야 합니다.

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

3. mkswap 명령으로 디바이스에 Linux 스왑 영역을 설정합니다.

```
[ec2-user ~]$ sudo mkswap /dev/xvdb
mkswap: /dev/xvdb: warning: wiping old ext3 signature.
Setting up swap space version 1, size = 4188668 KiB
no label, UUID=b4f63d28-67ed-46f0-b5e5-6928319e620b
```

4. 새 스왑 공간을 활성화합니다.

```
[ec2-user ~]$ sudo swapon /dev/xvdb
```

5. 새 스왑 공간이 사용 중인지 확인합니다.

```
[ec2-user ~]$ swapon -s
Filename      Type  Size Used Priority
/dev/xvdb      partition 4188668 0 -1
```

6. /etc/fstab 파일을 편집하여 부팅 시마다 이 스왑 공간이 자동 활성화되도록 설정합니다.

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

/etc/fstab 파일에 /dev/xvdb(또는 /dev/sdb) 항목이 있는 경우 아래 라인과 일치하도록 변경합니다. 이 디바이스에 대한 항목이 없는 경우 /etc/fstab 파일에 다음 라인을 추가합니다(시스템의 스왑 디바이스 이름 사용):

```
/dev/xvdb      none      swap      sw  0      0
```

Important

인스턴스가 종단되면 인스턴스 스토어 볼륨 데이터가 손실됩니다. 여기에는 Step 3 (p. 653)에서 생성한 인스턴스 스토어 스왑 공간 포맷도 포함됩니다. 따라서 인스턴스 스토어 스왑 공간을 사용하도록 구성한 인스턴스를 종단했다가 다시 시작할 경우에는 새로운 인스턴스 스토어 볼륨에서 Step 1 (p. 653)부터 Step 5 (p. 653)까지 반복해야 합니다.

인스턴스 스토어 볼륨의 디스크 성능 최적화

Amazon EC2가 디스크를 가상화하는 방식으로 인해 대부분의 인스턴스 스토어 볼륨 내 위치에서 첫 번째 쓰기는 이후의 쓰기보다 느립니다. 대부분 애플리케이션의 경우 인스턴스 수명 주기 동안 이 비용을 나누어 내는 것이 가능합니다. 그러나 높은 디스크 성능이 필요하다면 모든 드라이브 위치에 한 번 쓰기를 수행하여 드라이브를 초기화한 후 프로덕션에 사용하는 것이 좋습니다.

Note

직접 연결 SSD(Solid State Drive) 및 TRIM 지원을 사용하는 일부 인스턴스 유형은 초기화 없이 실행 시점에 최고 성능을 제공합니다. 각 인스턴스 유형의 인스턴스 스토어에 대한 자세한 내용은 [인스턴스 스토리지 볼륨 \(p. 643\)](#) 섹션을 참조하십시오.

지연 시간 또는 처리량에 대한 높은 유연성이 필요한 경우 Amazon EBS 사용을 권장합니다.

인스턴스 스토어 볼륨을 초기화하려면 초기화할 스토어에 따라 다음 dd 명령을 사용합니다(예: /dev/sdb 또는 /dev/name[0-7]n1).

Note

이 명령을 수행하기 전 드라이브 마운트를 해제해야 합니다.
초기화에는 시간이 오래 소요될 수 있습니다(엑스트라 라지 인스턴스의 경우 약 8시간).

인스턴스 스토어 볼륨을 초기화하려면 m1.large, m1.xlarge, c1.xlarge, m2.xlarge, m2.2xlarge 및 m2.4xlarge 인스턴스 유형에서 다음 명령을 사용합니다.

```
dd if=/dev/zero of=/dev/sdb bs=1M
dd if=/dev/zero of=/dev/sdc bs=1M
dd if=/dev/zero of=/dev/sdd bs=1M
dd if=/dev/zero of=/dev/sde bs=1M
```

전체 인스턴스 스토어 볼륨에서 동시에 초기화를 수행하려면 다음 명령을 사용합니다.

```
dd if=/dev/zero bs=1M|tee /dev/sdb|tee /dev/sdc|tee /dev/sde > /dev/sdd
```

RAID에 드라이브를 구성하면 전체 드라이브 위치에 쓰기가 되어 초기화를 수행할 수 있습니다. 소프트웨어 기반 RAID를 구성하는 경우 최소 재구성 속도를 변경해야 합니다.

```
echo $((30*1024)) > /proc/sys/dev/raid/speed_limit_min
```

Amazon EFS(Amazon Elastic File System)

Amazon EFS는 Amazon EC2에서 사용할 수 있는 확장 가능한 파일 스토리지를 제공합니다. EFS 파일 시스템을 만든 후 파일 시스템을 마운트하도록 인스턴스를 구성할 수 있습니다. 하나의 EFS 파일 시스템을 여러 인스턴스에서 실행하는 워크로드 및 애플리케이션에 대한 공통 데이터 소스로 사용할 수 있습니다. 자세한 내용은 [Amazon Elastic File System 제품 페이지](#)를 참조하십시오.

이 자습서에서는 하나의 EFS 파일 시스템과, 이 파일 시스템을 사용하여 데이터를 공유할 수 있는 두 개의 Linux 인스턴스를 만듭니다.

Important

Amazon EFS는 Windows 인스턴스에서 지원되지 않습니다.

작업

- [사전 조건 \(p. 655\)](#)
- [1단계: EFS 파일 시스템 만들기 \(p. 655\)](#)
- [2단계: 파일 시스템 마운트 \(p. 655\)](#)
- [3단계: 파일 시스템 테스트 \(p. 656\)](#)
- [4단계: 정리 \(p. 657\)](#)

사전 조건

- 보안 그룹(예: efs-sg)을 만든 후 다음 규칙을 추가합니다.
 - 컴퓨터로부터의 인바운드 SSH 연결 허용(소스는 네트워크의 CIDR 블록)
 - 그룹과 연결된 EC2 인스턴스로부터 인바운드 NFS 연결 허용(소스는 해당 보안 그룹)
- 키 페어를 생성합니다. 인스턴스를 구성할 경우 키 페어를 지정해야 하며, 키 페어를 지정하지 않으면 연결 할 수 없습니다. 자세한 내용은 [키 페어 생성 \(p. 18\)](#) 섹션을 참조하십시오.

1단계: EFS 파일 시스템 만들기

Amazon EFS를 사용하면 여러 인스턴스에서 마운트하고 동시에 액세스할 수 있는 파일 시스템을 만들 수 있습니다. 자세한 내용은 Amazon Elastic File System 사용 설명서에서 [Amazon EFS용 리소스 만들기](#)를 참조하십시오.

파일 시스템을 만들려면

1. <https://console.aws.amazon.com/efs/>에서 Amazon Elastic File System 콘솔을 엽니다.
2. Create file system을 선택합니다.
3. [Configure file system access] 페이지에서 다음을 수행합니다.
 - a. [VPC]에서 인스턴스에 사용할 VPC를 선택합니다.
 - b. [Create mount targets]에서 가용 영역을 모두 선택합니다.
 - c. 각 가용 영역에 대해 Security group이 [사전 조건 \(p. 655\)](#)에서 만든 보안 그룹인지 확인합니다.
 - d. [Next Step(다음 단계)]을 클릭합니다.
4. [Configure optional settings] 페이지에서 다음을 수행합니다.
 - a. 키=이름 태그의 경우 [Value]에 파일 시스템 이름을 입력합니다.
 - b. [Choose performance mode]에서 기본 옵션인 [General Purpose]를 그대로 사용합니다.
 - c. Next Step(다음 단계)을 클릭합니다.
5. [Review and create] 페이지에서 [Create File System]을 선택합니다.
6. 파일 시스템이 생성되면 파일 시스템 ID를 나중에 참조할 수 있도록 기록해둡니다.

2단계: 파일 시스템 마운트

다음 절차를 수행하여 t2.micro 인스턴스 두 개를 실행합니다. 사용자 데이터 스크립트는 인스턴스 실행 시 파일 시스템을 두 인스턴스 모두에 마운트하고, /etc/fstab를 업데이트하여 파일 시스템이 인스턴스가 재부팅 된 후 다시 마운트되도록 합니다. T2 인스턴스는 반드시 서브넷에서 시작되어야 합니다. 기본 VPC 또는 기본이 아닌 VPC를 사용할 수 있습니다.

Note

이미 실행 중인 인스턴스에 마운트하는 등, 다른 방법으로 볼륨을 마운트할 수 있습니다. 자세한 내용은 Amazon Elastic File System 사용 설명서에서 [파일 시스템 마운트](#)를 참조하십시오.

두 개의 인스턴스를 실행하여 하나의 EFS 파일 시스템을 마운트하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. [Launch Instance]를 선택합니다.
3. [Choose an Amazon Machine Image] 페이지에서 Amazon Linux AMI와 HVM 가상화 유형을 선택합니다.
4. [Choose an Instance Type] 페이지에서 기본 인스턴스 유형인 t2.micro를 그대로 사용하고 [Next: Configure Instance Details]를 선택합니다.
5. [Configure Instance Details] 페이지에서 다음을 수행합니다.
 - a. [Number of instances]에 2를 입력합니다.
 - b. [기본 VPC] 기본 VPC가 있는 경우 [Network]의 기본값이 됩니다. 기본 VPC와 [Subnet]의 기본값을 그대로 유지하여 Amazon EC2에서 인스턴스에 대해 선택한 가용 영역의 기본 서브넷을 사용합니다.
[기본이 아닌 VPC] [Network]에서 해당 VPC를 선택하고 [Subnet]에서 퍼블릭 서브넷을 선택합니다.
 - c. [기본이 아닌 VPC] [Auto-assign Public IP]에서 [Enable]을 선택합니다. 그렇지 않으면 인스턴스가 퍼블릭 IP 주소 또는 퍼블릭 DNS 이름을 받지 못합니다.
 - d. [Advanced Details]에서 다음 스크립트를 [User data]에 붙여 넣습니다. [EFS_ID]를 해당 파일 시스템 ID로 업데이트하고, [EFS_REGION]을 해당 파일 시스템의 리전 코드로 업데이트합니다. 필요할 경우 [EFS_MOUNT_DIR]을, 마운트한 파일 시스템의 디렉터리로 업데이트할 수 있습니다.

```
#!/bin/bash
yum update -y
yum install -y nfs-utils
EFS_ID=fs-xxxxxxxxx
EFS_REGION=region-code
EFS_MOUNT_DIR=/mnt/efs
mkdir -p ${EFS_MOUNT_DIR}
chown ec2-user:ec2-user ${EFS_MOUNT_DIR}
echo $(curl -s http://169.254.169.254/latest/meta-data/placement/availability-zone).${EFS_ID}.efs.${EFS_REGION}.amazonaws.com:/ ${EFS_MOUNT_DIR} nfs4
    nfsvers=4.1,rsize=1048576,wszie=1048576,hard,timeo=600,retrans=2 >> /etc/fstab
mount -a
```

- e. 마법사의 6단계로 진행합니다.
6. [Configure Security Group] 페이지에서 [Select an existing security group]를 선택하고 보안 그룹을 선택한 후 [Review and Launch]를 선택합니다.
7. [Review Instance Launch] 페이지에서 [Launch]를 선택합니다.
8. [Select an existing key pair or create a new key pair] 대화 상자에서 [Choose an existing key pair]를 선택하고 키 페어를 선택합니다. 승인 확인란을 선택하고 [Launch Instances]를 선택합니다.
9. 탐색 창에서 [Instances]를 선택하여 인스턴스의 상태를 확인합니다. 처음에 인스턴스 상태는 pending입니다. 이 상태가 running으로 변경되면 인스턴스를 사용할 수 있습니다.

3단계: 파일 시스템 테스트

인스턴스에 연결하여 지정한 디렉터리(예: /mnt/efs)에 해당 파일 시스템이 마운트되었는지 확인할 수 있습니다.

파일 시스템이 마운트되었는지 확인하려면

1. 인스턴스에 연결합니다. 자세한 내용은 [Linux 인스턴스에 연결 \(p. 274\)](#) 섹션을 참조하십시오.
2. 각 인스턴스의 터미널 창에서 df -T 명령을 실행하여 EFS 파일 시스템이 마운트되었는지 확인합니다.

```
$ df -T
Filesystem      Type            1K-blocks   Used   Available Use% Mounted on
/dev/xvda1      ext4           8123812   1949800    6073764  25% /
devtmpfs        devtmpfs       4078468     56    4078412  1% /dev
tmpfs          tmpfs           4089312     0    4089312  0% /dev/shm
efs-dns         nfs4          9007199254740992    0    9007199254740992  0% /mnt/efs
```

예제 출력에 나와 있는 파일 시스템 이름 **efs-dns**의 형식은 다음과 같습니다.

```
availability-zone.filesystem-id.efs.region.amazonaws.com:/
```

3. (선택 사항) 한 인스턴스의 파일 시스템에서 파일을 하나 생성한 후 다른 인스턴스에서 해당 파일이 보이는지 확인합니다.

- a. 첫 인스턴스에서 다음 명령을 실행하여 파일을 생성합니다.

```
$ sudo touch /mnt/efs/test-file.txt
```

- b. 둘째 인스턴스에서 다음 명령을 실행하여 파일을 봅니다.

```
$ ls /mnt/efs
test-file.txt
```

4단계: 정리

이 자습서를 마치면 인스턴스를 종료하고 파일 시스템을 삭제해도 됩니다.

인스턴스를 종료하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. 종료할 인스턴스를 선택합니다.
4. [Actions], [Instance State], [Terminate]를 차례로 선택합니다.
5. 확인 메시지가 나타나면 [Yes, Terminate]를 선택합니다.

파일 시스템을 삭제하려면

1. <https://console.aws.amazon.com/efs/>에서 Amazon Elastic File System 콘솔을 엽니다.
2. 삭제할 파일 시스템을 선택합니다.
3. [Actions], [Delete file system]을 차례로 선택합니다.
4. 확인 메시지가 표시되면 파일 시스템 ID를 입력하고 [Delete File System]을 선택합니다.

Amazon Simple Storage Service(Amazon S3)

Amazon S3는 인터넷 데이터용 리포지토리입니다. Amazon S3는 안정적이고 빠르며 비용이 저렴한 데이터 스토리지 인프라에 대한 액세스를 제공합니다. S3은 언제든지 Amazon EC2 내 또는 웹의 어디서나 데이터를 원하는 크기만큼 저장하고 가져올 수 있게 지원함으로써 웹 규모의 컴퓨팅 작업을 쉽게 수행할 수 있도록 설계되었습니다. Amazon S3은 데이터 객체를 여러 시설에 걸쳐 다수 장치에 중복으로 저장하고, 많은 개별 클라이언트 또는 애플리케이션 스레드가 이런 데이터 객체에 대해 연속적 읽기 및 쓰기 액세스를 할 수 있도록 설계되었습니다.

록 지원합니다. Amazon S3에 저장된 중복 데이터를 사용해서 인스턴스 또는 애플리케이션 장애로부터 빠르게 복구할 수 있습니다.

Amazon EC2는 AMI(Amazon Machine Image) 저장을 위해 Amazon S3을 사용합니다. AMI를 사용해서 EC2 인스턴스를 시작합니다. 인스턴스 장애의 경우, 저장된 AMI를 사용해서 즉시 다른 인스턴스를 실행할 수 있으며 이를 통해 빠른 복구 및 비즈니스 지속성을 달성할 수 있습니다.

또한 Amazon EC2는 Amazon S3을 사용해서 데이터 볼륨의 스냅샷(백업 사본)을 저장합니다. 애플리케이션 또는 시스템 장애가 발생한 경우 스냅샷을 사용해서 빠르고 안정적으로 데이터를 복원할 수 있습니다. 또한 스냅샷을 하나의 기준으로 사용하여 다수의 새 데이터 볼륨 생성, 기존 데이터 볼륨의 크기 확장, 다수 가용 영역 간 데이터 볼륨 이동 등을 수행할 수 있으며, 이를 통해 사용할 수 있는 데이터를 높은 수준으로 확장할 수 있습니다. 데이터 볼륨 및 스냅샷 사용에 대한 자세한 내용은 [Amazon Elastic Block Store \(p. 560\)](#) 섹션을 참조하십시오.

객체는 Amazon S3에 저장되는 기본 개체입니다. Amazon S3에 저장된 모든 객체는 버킷에 저장됩니다. 버킷은 Amazon S3 네임스페이스를 최상위 수준에서 구성하며 해당 스토리지를 담당하는 계정을 식별합니다. Amazon S3 버킷은 인터넷 도메인 이름과 유사합니다. 버킷에 저장된 객체는 고유의 키 값을 가지고 있으며 HTTP URL 주소를 사용해서 검색할 수 있습니다. 예를 들어 `/photos/mygarden.jpg` 키 값을 가진 객체가 `myawsbucket` 버킷에 저장되어 있는 경우, `http://myawsbucket.s3.amazonaws.com/photos/mygarden.jpg` URL을 사용해서 주소를 지정해 액세스할 수 있습니다.

Amazon S3에 대한 자세한 내용은 [Amazon S3제품 페이지](#)를 참조하십시오.

Amazon S3 및 Amazon EC2

스토리지에 있어 Amazon S3의 이점을 고려하여 이 서비스를 사용해서 EC2 인스턴스에 사용할 파일 및 데이터 세트를 저장하는 경우가 있을 수 있습니다. Amazon S3 및 인스턴스 간에 데이터를 주고 받는 방법은 여러 가지가 있습니다. 아래 설명한 예뿐만 아니라 여러 사람들이 작성한 다양한 도구가 있으며, 이를 사용해서 컴퓨터 또는 인스턴스에서 Amazon S3의 데이터에 액세스할 수 있습니다. 흔하게 사용하는 방법들 중 일부는 AWS 포럼에서 논의되고 있습니다.

권한을 부여받은 경우, 다음 방법 중 하나를 사용해서 Amazon S3 및 인스턴스로 또는 인스턴스로부터 파일을 복사할 수 있습니다.

GET 또는 wget

wget 유ти리티는 Amazon S3에서 퍼블릭 객체를 다운로드할 수 있도록 허용하는 HTTP 및 FTP 클라이언트입니다. 이는 Amazon Linux 및 대부분의 기타 배포판에서 기본적으로 설치되어 있으며, Windows에서 다운로드할 수 있습니다. Amazon S3 객체를 다운로드하려면 다운로드할 객체의 URL로 해당 부분을 대체하여 다음 명령을 사용합니다.

```
wget https://s3.amazonaws.com/my_bucket/my_folder/my_file.ext
```

이 방법은 요청한 객체가 퍼블릭일 것을 요합니다. 객체가 퍼블릭이 아닌 경우는 `ERROR 403: Forbidden` 메시지를 받게 됩니다. 이 오류 메시지를 받은 경우는 Amazon S3 콘솔을 열고 객체의 권한을 퍼블릭으로 변경합니다. 자세한 내용은 [Amazon Simple Storage Service 개발자 가이드](#) 섹션을 참조하십시오.

AWS 명령줄 인터페이스

AWS CLI(AWS 명령줄 인터페이스)는 AWS 서비스를 관리하는 통합 도구입니다. 도구 하나만 다운로드하여 구성하면 여러 AWS 서비스를 명령행에서 관리하고 스크립트를 통해 자동화할 수 있습니다. AWS CLI를 통해 사용자는 인증을 하고 Amazon S3에서 제한되는 항목을 다운로드하고 다른 항목을 업로드할 수도 있게 됩니다. 상기 도구의 설치 및 구성 등에 대한 자세한 내용 [AWS 명령줄 인터페이스 세부 정보 페이지](#) 섹션을 참조하십시오.

`aws s3 cp` 명령은 Unix `cp` 명령과 유사합니다(명령 문법: `aws s3 cp ## ##`). Amazon S3에서 인스턴스로 파일을 복사하거나, 인스턴스에서 Amazon S3로 파일을 복사하거나, 하나의 Amazon S3 위치에서 다른 위치로 파일을 복사할 수도 있습니다.

다음 명령을 사용해서 Amazon S3에서 인스턴스로 객체를 복사합니다.

```
$ aws s3 cp s3://my_bucket/my_folder/my_file.ext my_copied_file.ext
```

다음 명령을 사용해서 인스턴스에서 Amazon S3로 객체를 복사합니다.

```
$ aws s3 cp my_copied_file.ext s3://my_bucket/my_folder/my_file.ext
```

다음 명령을 사용해서 하나의 Amazon S3 위치에서 다른 위치로 객체를 복사합니다.

```
$ aws s3 cp s3://my_bucket/my_folder/my_file.ext s3://my_bucket/my_folder/my_file2.ext
```

aws s3 sync 명령은 전체 Amazon S3 버킷을 로컬 디렉터리 위치에 동기화할 수 있습니다. 이는 데이터 세트를 다운로드하고 로컬 사본을 원격 세트에 따라 최신으로 유지하는 데 도움이 될 수 있습니다. 명령 문법은 다음과 같습니다. aws s3 sync ## ##. Amazon S3 버킷에서 적절한 권한을 보유한 경우, 작업이 완료되면 소스와 대상의 위치를 바꿔 입력해 명령을 실행해서 로컬 디렉터리를 클라우드로 푸시할 수 있습니다.

다음 명령을 사용해서 전체 Amazon S3 버킷을 사용자의 로컬 디렉터리로 다운로드할 수 있습니다.

```
$ aws s3 sync s3://remote_S3_bucket local_directory
```

Windows PowerShell용 AWS 도구

Windows 인스턴스는 Amazon S3 콘솔을 액세스하기 위해 사용할 수 있는 그래픽 브라우저로 인한 이점이 있습니다. 그러나 스크립팅의 경우 Windows 사용자는 [Windows PowerShell용 AWS 도구](#)를 사용해서도 Amazon S3와 객체를 주고 받을 수 있습니다.

다음 명령을 사용해서 Amazon S3 객체를 Windows 인스턴스로 복사합니다.

```
PS C:\> Copy-S3Object -BucketName my_bucket -Key my_folder/my_file.ext -  
LocalFile my_copied_file.ext
```

Amazon S3 API

개발자라면 API를 사용해서 Amazon S3의 데이터에 액세스할 수 있습니다. 자세한 내용은 [Amazon Simple Storage Service 개발자 가이드](#) 섹션을 참조하십시오. 이런 API 및 그 예들을 사용해서 애플리케이션 개발을 지원하고 이를 boto Python 인터페이스 등 다른 API 및 SDK와 통합할 수 있습니다.

인스턴스 볼륨 제한

인스턴스에서 보유할 수 있는 최대 볼륨 수는 운영 체제에 따라 다릅니다. 인스턴스에 추가할 볼륨의 수를 고려할 때 I/O 대역폭 증가 또는 스토리지 용량 증가의 필요성 여부를 고려해야 합니다.

목차

- [Linux 볼륨 제한 \(p. 659\)](#)
- [Windows 볼륨 제한 \(p. 660\)](#)
- [대역폭 및 용량 비교 \(p. 660\)](#)

Linux 볼륨 제한

볼륨을 40개 이상 연결하면 부팅 오류가 발생할 수 있습니다. 이 개수에는 루트 볼륨과 함께 연결된 인스턴스 스토어 볼륨과 EBS 볼륨이 모두 포함됩니다. 볼륨이 여러 개인 인스턴스에서 부팅 문제가 발생하는 경우 인

스턴스를 중지한 후 부팅 과정에서 필요하지 않은 볼륨을 분리하고 인스턴스가 실행되면 해당 볼륨을 다시 연결합니다.

Important

Linux 인스턴스에 볼륨을 40개 이상 연결할 수 있도록 최상의 노력이 제공되지만 보장되지는 않습니다.

Windows 볼륨 제한

다음 표는 사용된 드라이버를 기반으로 Windows 인스턴스에 대한 볼륨 제한을 보여 줍니다. 이러한 개수에는 루트 볼륨과 함께 연결된 인스턴스 스토어 볼륨과 EBS 볼륨이 모두 포함됩니다.

Important

Windows 인스턴스에 다음 개수 이상의 볼륨을 연결할 수 있도록 최상의 노력이 제공되지만 보장되지는 않습니다.

드라이버	볼륨 제한
AWS PV	26
Citrix PV	26
Red Hat PV	17

성능 문제가 발생할 수 있으므로 AWS PV 또는 Citrix PV 드라이버를 사용하는 Windows 인스턴스에 볼륨을 26개 이상 연결하지 않는 것이 좋습니다.

인스턴스에서 사용하는 PV 드라이버를 확인하거나 Red Hat에서 Citrix PV 드라이버로 Windows 인스턴스를 업그레이드하려면 [Windows 인스턴스에서 PV 드라이버 업그레이드](#)를 참조하십시오.

디바이스 이름이 볼륨과 연결된 방식에 대한 자세한 내용은 Windows 인스턴스용 Amazon EC2 사용 설명서의 [Windows EC2 인스턴스의 볼륨에 디스크 매핑](#)을 참조하십시오.

대역폭 및 용량 비교

대역폭을 일관성 있고 예측 가능하게 사용하기 위해서는 EBS에 최적화된 인스턴스 또는 10Gb 네트워크에 연결된 인스턴스 및 범용 SSD 또는 프로비저닝된 IOPS SSD 볼륨을 사용해야 합니다. [Amazon EC2 인스턴스 구성](#) (p. 623)의 지침을 준수하여 볼륨에 프로비저닝된 IOPS와 인스턴스가 최대 성능에서 사용 가능한 대역폭을 일치시킵니다. RAID 구성의 경우 볼륨이 8개 이상인 어레이에는 I/O 오버헤드가 증가하여 성능이 저하되는 결과가 여러 관리자에 의해 관찰되었습니다. 따라서 개별 애플리케이션의 성능을 테스트한 다음 필요에 따라 조정하십시오.

Linux 인스턴스의 디바이스 명명

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

목차

- [사용 가능한 디바이스 이름](#) (p. 661)
- [디바이스 이름 고려 사항](#) (p. 661)

Windows 인스턴스의 디바이스 이름에 대한 자세한 내용은 Windows 인스턴스용 Amazon EC2 사용 설명서의 [Device Naming on Windows Instances](#) 섹션을 참조하십시오.

사용 가능한 디바이스 이름

다음 표에 Linux 인스턴스의 사용 가능한 디바이스 이름이 나와 있습니다. 인스턴스에 연결할 수 있는 볼륨의 수는 운영 체제에 따라 결정됩니다. 자세한 내용은 [인스턴스 볼륨 제한 \(p. 659\)](#) 섹션을 참조하십시오.

가상화 유형	응시 가능	루트 전용	EBS 볼륨 추천	인스턴스 스토어 볼륨 용	NVMe 인스턴스 스토어 볼륨용
반가상화(PV)	/dev/sd[a-z] /dev/sd[a-z][1-15] /dev/hd[a-z] /dev/hd[a-z][1-15]	/dev/sda1	/dev/sd[f-p] /dev/sd[f-p][1-6]	/dev/sd[b-e] /dev/sd[b-y](hs1.8xlarge)	해당 사항 없음
HVM	/dev/sd[a-z] /dev/xvd[b-c][a-z]	AMI에 따라 다른 /dev/sda1 또는 /dev/xvda	/dev/sd[f-p]	/dev/sd[b-e] /dev/sd[b-y](d2.8xlarge) /dev/sd[b-y](hs1.8xlarge) /dev/sd[b-i](i2.8xlarge)	/dev/nvme[0-7]n1 *

* NVMe 볼륨은 자동으로 열거되고 디바이스 이름이 할당됩니다. 블록 디바이스 매핑에 NVMe 볼륨을 지정할 필요가 없습니다.

다음 AWS CLI 명령을 사용하여 특정 AMI에 대한 루트 디바이스 이름을 확인할 수 있습니다.

```
aws ec2 describe-images --image-ids image_id --query 'Images[].RootDeviceName'
```

인스턴스 스토어 볼륨에 대한 자세한 내용은 [Amazon EC2 인스턴스 스토어 \(p. 642\)](#) 섹션을 참조하십시오. 루트 디바이스 스토리지에 대한 자세한 내용은 [Amazon EC2 루트 디바이스 볼륨 \(p. 11\)](#) 섹션을 참조하십시오.

디바이스 이름 고려 사항

디바이스 이름을 선택할 때는 다음 사항에 주의하십시오.

- 인스턴스 스토어 볼륨을 연결할 때 사용된 디바이스 이름을 사용하여 EBS 볼륨을 연결할 수 있지만, 이러한 경우 예기치 않은 동작이 발생할 수 있으므로 수행하지 않는 것이 좋습니다.
- 커널의 블록 디바이스 드라이버에 따라, 디바이스가 사용자가 지정한 것과는 다른 이름으로 연결될 수 있습니다. 예를 들어 /dev/sdh라는 디바이스 이름을 지정할 경우 디바이스 이름이 커널에 의해 /dev/xvdh 또는 /dev/hdh로 바뀔 수 있습니다. 대부분의 경우, 뒤에 오는 문자는 동일하게 유지됩니다. Red Hat

Enterprise Linux의 일부 버전과 CentOS와 같은 Red Hat Enterprise Linux의 변형 버전에서는 뒤에 오는 문자가 변경될 수도 있습니다(즉, /dev/sda가 /dev/xvde로 바뀔 수 있음). 이런 경우, 각 디바이스 이름에서 뒤에 오는 문자는 같은 수로 늘어납니다. 예를 들어 /dev/sdb는 /dev/xvdf가 되고 /dev/sdc는 /dev/xvdg가 되는 식입니다. Amazon Linux AMI는 시작할 때 지정한 이름으로 심볼 링크를 만들어 이름이 바뀐 디바이스 경로를 가리키지만, 다른 AMI는 다르게 동작할 수도 있습니다.

- 인스턴스의 NVMe 인스턴스 스토어 볼륨의 수는 인스턴스의 크기에 따라 다릅니다. 디바이스 이름은 /dev/nvme0n1, /dev/nvme1n1 등입니다.
- Linux 인스턴스에서는 반가상화(PV) 및 하드웨어 가상 머신(HVM)과 같은 두 가지 유형의 가상화를 사용할 수 있습니다. 인스턴스의 가상화 유형은 인스턴스를 시작할 때 사용된 AMI에 의해 결정됩니다. 인스턴스 유형에 따라 PV와 HVM을 모두 지원하거나, HVM 또는 PV만 지원합니다. 인스턴스의 가상화 유형에 따라 권장되고 사용 가능한 디바이스 이름이 다르기 때문에 AMI의 가상화 유형에 주의해야 합니다. 자세한 내용은 [Linux AMI 가상화 유형 \(p. 66\)](#) 섹션을 참조하십시오.
- 뒤에 숫자가 있거나 있지 않고 디바이스 이름의 문자가 동일한 볼륨은 연결할 수 없습니다. 예를 들어, 볼륨을 /dev/sdc로 연결한 다음 다른 볼륨을 /dev/sdc1에 연결하면 인스턴스에서는 /dev/sdc만을 볼 수 있습니다. 디바이스 이름 끝에 숫자를 사용하려면 기본 문자가 동일한 모든 디바이스 이름의 끝에 숫자를 사용해야 합니다(/dev/sdc1, /dev/sdc2, /dev/sdc3 등).
- 하드웨어 가상 머신(HVM) AMI는 디바이스 이름 마지막에 숫자 사용을 지원하지 않습니다.
- 일부 사용자 정의 커널은 /dev/sd[f-p] 또는 /dev/sd[f-p][1-6]의 사용을 제한할 수 있습니다. /dev/sd[q-z] 또는 /dev/sd[q-z][1-6]을 사용할 수 없는 경우 /dev/sd[f-p] 또는 /dev/sd[f-p][1-6]로 바꿔 시도하십시오.

블록 디바이스 매핑

실행된 각 인스턴스에는 연관된 루트 디바이스 볼륨(Amazon EBS 볼륨 또는 인스턴스 스토어 볼륨)이 있습니다. 블록 디바이스 매핑을 사용하면 실행될 때 인스턴스에 연결할 추가 EBS 볼륨 또는 인스턴스 스토어 볼륨을 지정할 수 있습니다. 또한, 실행 중인 인스턴스에 EBS 볼륨을 추가로 연결할 수도 있습니다. [Amazon EBS 볼륨을 인스턴스에 연결 \(p. 576\)](#)을 참조하십시오. 그러나 블록 디바이스 매핑을 사용하여 인스턴스가 실행되었을 때 인스턴스 스토어 볼륨을 연결하는 방식으로만 인스턴스에 인스턴스 스토어 볼륨을 연결할 수 있습니다.

루트 디바이스 볼륨에 대한 자세한 내용은, [루트 디바이스 볼륨이 계속 유지되도록 변경 \(p. 14\)](#) 섹션을 참조하십시오.

목차

- [블록 디바이스 매핑의 개념 \(p. 662\)](#)
- [AMI 블록 디바이스 매핑 \(p. 664\)](#)
- [인스턴스 블록 디바이스 매핑 \(p. 666\)](#)

블록 디바이스 매핑의 개념

블록 디바이스는 바이트 또는 비트(블록) 단위로 순차적으로 데이터를 이동시키는 스토리지 디바이스입니다. 이러한 디바이스는 임의 액세스를 지원하고 일반적으로 버퍼 I/O를 사용합니다. 그러한 예로는 하드 디스크, CD-ROM 드라이브, 플래시 드라이브 등이 있습니다. 블록 디바이스는 컴퓨터에 물리적으로 장착될 수 있고 그렇지 않은 경우 컴퓨터에 물리적으로 장착된 것처럼 임의 액세스가 가능합니다. Amazon EC2가 지원하는 두 가지 블록 디바이스 유형:

- 인스턴스 스토어 볼륨(기본 하드웨어가 인스턴스의 호스트 컴퓨터에 물리적으로 장착된 가상 디바이스)
- EBS 볼륨(원격 스토리지 디바이스)

블록 디바이스 매핑은 인스턴스에 연결할 블록 디바이스(인스턴스 볼륨 및 EBS 볼륨)를 정의합니다. AMI 생성 시 블록 디바이스 매핑을 지정하면 AMI에서 실행되는 모든 인스턴스가 해당 매핑을 사용할 수 있습니다.

아니면, 인스턴스 생성 시 블록 디바이스 매핑을 지정하여 이 매핑이 인스턴스가 실행된 AMI에서 지정된 매핑을 재정의하도록 할 수 있습니다. 인스턴스 유형에서 지원되는 모든 NVMe 인스턴스 스토어 볼륨이 인스턴스 시작 시 자동으로 추가됩니다. 따라서 AMI 또는 인스턴스에 대한 블록 디바이스 매핑에 추가할 필요가 없습니다.

목차

- [블록 디바이스 매핑 항목 \(p. 663\)](#)
- [블록 디바이스 매핑 인스턴스 스토어 경고 \(p. 663\)](#)
- [블록 디바이스 매핑 예제 \(p. 664\)](#)
- [운영 체제에서 디바이스 사용 방법 \(p. 664\)](#)

블록 디바이스 매핑 항목

블록 디바이스 매핑을 생성할 때 인스턴스에 연결할 각 블록 디바이스에 다음 정보를 지정합니다.

- Amazon EC2 내에서 사용되는 디바이스 이름 인스턴스의 블록 디바이스 드라이버는 볼륨이 마운트될 때 실제 볼륨 이름을 할당하고 할당된 이름은 Amazon EC2 권장 이름과 다를 수 있습니다. 자세한 내용은 [Linux 인스턴스의 디바이스 명명 \(p. 660\)](#) 섹션을 참조하십시오.
- [인스턴스 스토어 볼륨] 가상 디바이스: `ephemeral[0-23]` 그러나 이러한 볼륨의 개수 및 크기는 인스턴스 유형에 따라 다른 인스턴스에서 사용 가능한 인스턴스 스토어 볼륨을 초과하지 않아야 합니다.
- [NVMe 인스턴스 스토어 볼륨] 이러한 볼륨은 `/dev/nvme[0-7]n1`로 자동 매핑됩니다. 블록 디바이스 매핑에 인스턴스 유형에서 지원하는 NVMe 볼륨을 지정할 필요가 없습니다.
- [EBS 볼륨] 블록 디바이스를 생성하기 위해 사용하는 스냅샷 ID(`snap-xxxxxxxx`) 볼륨 크기를 지정하는 경우 이 값은 선택 사항입니다.
- [EBS; 볼륨] GiB 단위의 볼륨 크기 지정된 크기는 지정된 스냅샷 크기 이상이어야 합니다.
- [EBS; 볼륨] 인스턴스 종료 시 볼륨 삭제 여부(`true` 또는 `false`) 기본값은 루트 디바이스 볼륨은 `true`이고 연결된 볼륨은 `false`입니다 AMI를 생성하면 그 블록 디바이스 매핑이 인스턴스에서 이 설정을 내려 받습니다. 인스턴스를 시작하면 AMI에서 이 설정을 내려 받습니다.
- [EBS 볼륨] 볼륨 유형은 범용 SSD 볼륨의 경우 `gp2`, 프로비저닝된 IOPS SSD 볼륨의 경우 `io1`, 처리량에 최적화된 HDD 볼륨의 경우 `st1`, Cold HDD 볼륨의 경우 `sc1` 또는 Magnetic 볼륨의 경우 `standard`일 수 있습니다. 기본 값은 Amazon EC2 콘솔에서 `gp2`이고 AWS SDK 및 AWS CLI에서 `standard`입니다.
- [EBS 볼륨] 볼륨이 지원하는 초당 입력/출력 작업 수(IPOS) (`gp2`, `st1`, `sc1` 또는 `standard` 볼륨에서는 사용되지 않음.)

블록 디바이스 매핑 인스턴스 스토어 경고

블록 디바이스 매핑에 인스턴스 스토어 볼륨이 있는 AMI에서 인스턴스를 시작하는 경우 고려해야 할 여러 가지 경고가 있습니다.

- 일부 인스턴스 유형은 다른 인스턴스보다 인스턴스 스토어 볼륨이 더 크고 일부 인스턴스 유형의 경우 인스턴스 스토어 볼륨이 없을 수도 있습니다. 인스턴스 볼륨이 1개의 인스턴스 스토어 볼륨을 지원하는데 AMI에 2개의 인스턴스 스토어 볼륨이 있는 경우 인스턴스는 1개의 인스턴스 스토어 볼륨으로 실행됩니다.
- 인스턴스 스토어 볼륨은 실행 시에만 매핑될 수 있습니다. 인스턴스 스토어 볼륨이 없는 인스턴스 (`t2.micro` 등)는 중지할 수 없으므로 해당 인스턴스를 인스턴스 스토어 볼륨을 지원하는 유형으로 변경한 다음 인스턴스 스토어 볼륨이 있는 인스턴스를 다시 시작합니다. 그러나 인스턴스에서 AMI를 생성하고 인스턴스 스토어 볼륨을 지원하는 인스턴스 유형에서 실행한 다음 그러한 인스턴스 스토어 볼륨을 인스턴스로 매핑하는 것은 가능합니다.
- 인스턴스 스토어 볼륨이 있는 매핑된 인스턴스를 실행한 다음 인스턴스를 중지하고 인스턴스 스토어 볼륨의 개수가 적은 인스턴스 유형으로 변경한 후 다시 시작한 경우 인스턴스 메타데이터에는 처음 실행된 인스턴스 스토어 볼륨 매핑이 계속해서 표시됩니다. 그러나 그러한 인스턴스에서는 해당 인스턴스 유형에서 지원되는 최대 인스턴스 스토어 볼륨 갯수만 사용할 수 있습니다.

Note

인스턴스가 중지되면 인스턴스 스토어 볼륨의 모든 데이터가 손실됩니다.

- 실행 시의 인스턴스 스토어 용량에 따라 실행 시 지정되지 않는 경우 M3 인스턴스는 실행되는 AMI 인스턴스 스토어 블록 디바이스 매핑을 무시할 수 있습니다. 실행하려는 AMI에 AMI 매핑 인스턴스 스토어 볼륨이 있는 경우 실행 시 인스턴스 스토어 블록 디바이스 매핑을 지정해야 인스턴스가 실행될 때 인스턴스 스토어 볼륨을 사용할 수 있습니다.

블록 디바이스 매핑 예제

이 그림은 EBS 기반 인스턴스의 블록 디바이스 매핑 예제를 보여줍니다. 이 그림에서는 `/dev/sdb`가 `ephemeral0`에 매핑되고 EBS 볼륨 2개는 각각 `/dev/sdh` 및 `/dev/sdj`에 매핑됩니다. 또한 여기에서 루트 디바이스 볼륨인 EBS 볼륨은 `/dev/sda1`입니다.

이 예제 블록 디바이스 매핑에서는 이 주제와 관련된 예제 명령어 및 API가 사용되었습니다. [AMI용 블록 디바이스 매핑 지정 \(p. 664\)](#) 및 [인스턴스 실행 시 블록 디바이스 매핑 업데이트 \(p. 667\)](#)에서 블록 디바이스 매핑을 생성하는 API와 예제 명령을 확인할 수 있습니다.

운영 체제에서 디바이스 사용 방법

`/dev/sdh` 및 `xvdh` 등의 디바이스 이름은 Amazon EC2에서 블록 디바이스를 나타내는 이름으로 사용됩니다. Amazon EC2에서 블록 디바이스 매핑은 EC2 인스턴스를 연결하는 블록 디바이스를 지정하는 데 사용됩니다. 블록 디바이스가 인스턴스에 연결되면 운영 체제에 마운트되어야 사용자가 해당 스토리지 디바이스에 액세스할 수 있습니다. 블록 디바이스가 인스턴스에서 분리되면 운영 체제에서 마운트가 해제되고 사용자는 더 이상 해당 스토리지 디바이스에 액세스할 수 없습니다.

Linux 인스턴스의 경우 블록 디바이스 매핑에 지정된 디바이스 이름은 인스턴스가 처음 부팅될 때 해당하는 블록 디바이스로 매핑됩니다. 인스턴스 유형에 따라 어느 인스턴스 스토어 볼륨이 포맷되고 기본 마운트될지가 결정됩니다. 사용자는 인스턴스 유형에 따라 사용 가능한 인스턴스 스토어 볼륨을 초과하지 않는 범위 내에서 실행 시 인스턴스 스토어 볼륨을 추가로 마운트할 수 있습니다. 자세한 내용은 [Amazon EC2 인스턴스 스토어 \(p. 642\)](#) 섹션을 참조하십시오. 인스턴스용 블록 디바이스 드라이버에 따라 볼륨 포맷 및 마운트 시 어느 디바이스가 사용될지가 결정됩니다. 자세한 내용은 [Amazon EBS 볼륨을 인스턴스에 연결 \(p. 576\)](#) 섹션을 참조하십시오.

AMI 블록 디바이스 매핑

각 AMI에는 AMI에서 시작될 때 인스턴스로 연결될 블록 디바이스를 지정하는 블록 디바이스 매핑이 있습니다. Amazon은 루트 디바이스가 포함된 AMI만을 지원합니다. AMI에 추가 블록 디바이스를 추가하려면 고유 AMI를 생성해야 합니다.

목차

- [AMI용 블록 디바이스 매핑 지정 \(p. 664\)](#)
- [AMI 블록 디바이스 매핑에서 EBS 볼륨 보기 \(p. 666\)](#)

AMI용 블록 디바이스 매핑 지정

두 가지 방법으로 AMI를 생성할 때 루트 디바이스 볼륨과 볼륨을 지정할 수 있습니다. 인스턴스에서 AMI를 생성하기 전 실행 중인 인스턴스에 볼륨을 이미 연결한 경우 AMI용 블록 디바이스 매핑에는 동일한 해당 볼륨이 포함됩니다. EBS 볼륨에서 기존 데이터는 새 스냅샷에 저장되고 블록 디바이스 매핑에 새로운 이 스냅샷이 지정됩니다. 인스턴스 스토어 볼륨의 경우 데이터는 보존되지 않습니다.

EBS 기반 AMI의 경우 블록 디바이스 매핑을 사용하여 EBS 볼륨 및 인스턴스 스토어 볼륨을 추가할 수 있습니다. 인스턴스 스토어 지원 AMI의 경우 이미지를 등록할 때 이미지 매니페스트 파일에서 블록 디바이스 매핑 항목을 수정하여 인스턴스 스토어 볼륨만 추가할 수 있습니다.

Note

M3 인스턴스의 경우 실행 시 인스턴스에 대한 블록 디바이스 매핑에 인스턴스 스토어 볼륨을 반드시 지정해야 합니다. M3 인스턴스 실행 시 인스턴스 스토어 볼륨이 인스턴스 블록 디바이스 매핑으로 지정되지 않으면 AMI용 블록 디바이스 매핑에 지정된 인스턴스 스토어 볼륨이 무시될 수 있습니다.

콘솔을 사용하여 AMI에 볼륨을 추가하려면

1. Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Instances]를 선택합니다.
3. 인스턴스를 선택하고 [Actions], [Image], [Create Image]를 선택합니다.
4. [Create Image] 대화 상자에서 [Add New Volume]을 선택합니다.
5. [Type] 목록에서 볼륨 유형을 선택하고 [Device] 목록에서 디바이스 이름을 선택합니다. EBS 볼륨의 경우 스냅샷, 볼륨 크기와 볼륨 유형을 선택적으로 지정할 수 있습니다.
6. Create Image를 선택합니다.

명령줄을 사용하여 AMI에 볼륨을 추가하려면

`create-image` AWS CLI 명령을 사용하여 EBS 지원 AMI에 블록 디바이스 매핑을 지정합니다. `register-image` AWS CLI 명령을 사용하여 인스턴스 스토어 지원 AMI에 블록 디바이스 매핑을 지정합니다.

다음 파라미터를 사용하여 블록 디바이스 매핑을 지정합니다.

```
--block-device-mappings [mapping, ...]
```

인스턴스 스토어 볼륨을 추가하려면 다음 매핑을 사용합니다.

```
{  
    "DeviceName": "/dev/sdf",  
    "VirtualName": "ephemeral0"  
}
```

비어 있는 100 GiB Magnetic 볼륨을 추가하려면 다음 매핑을 사용합니다.

```
{  
    "DeviceName": "/dev/sdg",  
    "Ebs": {  
        "VolumeSize": 100  
    }  
}
```

스냅샷 기반 EBS 볼륨을 추가하려면 다음 매핑을 사용합니다.

```
{  
    "DeviceName": "/dev/sdh",  
    "Ebs": {  
        "SnapshotId": "snap-xxxxxxxx"  
    }  
}
```

디바이스에 대한 매핑을 생략하려면 다음 매핑을 사용합니다.

```
{  
    "DeviceName": "/dev/sdj",  
    "NoDevice": ""  
}
```

}

또는 다음 명령(Windows PowerShell용 AWS 도구)과 함께 `-BlockDeviceMapping` 파라미터를 사용할 수 있습니다.

- [New-EC2Image](#)
- [Register-EC2Image](#)

AMI 블록 디바이스 매핑에서 EBS 볼륨 보기

AMI의 블록 디바이스 매핑에서 EBS 볼륨을 쉽게 확인할 수 있습니다.

콘솔을 사용하여 AMI용 EBS 볼륨을 확인하려면

1. Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [AMIs]를 선택합니다.
3. [Filter] 목록에서 [EBS images]를 선택하여 EBS 지원 AMI 목록을 가져옵니다.
4. 원하는 AMI를 선택한 다음 [Details] 탭을 확인합니다. 루트 디바이스에서 최소한으로 사용 가능한 정보는 다음과 같습니다.
 - 루트 디바이스 유형 (ebs)
 - 루트 디바이스 이름(예: /dev/sda1)
 - 블록 디바이스(예: /dev/sda1=snap-1234567890abcdef0:8:true)

AMI가 블록 디바이스 매핑을 사용하여 추가 EBS 볼륨으로 생성된 경우 Block Devices 필드에 해당 추가 볼륨에 대한 매핑도 표시됩니다. (이 화면에는 인스턴스 스토어 볼륨이 표시되지 않는다는 것에 유의하십시오.)

명령줄을 사용하여 AMI의 EBS 볼륨을 보려면

[describe-images](#)(AWS CLI) 명령 또는 [Get-EC2Image](#)(Windows PowerShell용 AWS 도구) 명령을 사용하여 AMI용 블록 디바이스 매핑에 EBS 볼륨을 표시합니다.

인스턴스 블록 디바이스 매핑

기본적으로, 사용자가 실행한 인스턴스에는 인스턴스가 실행된 AMI의 블록 디바이스 매핑에 지정된 모든 스토리지 디바이스가 포함됩니다. 인스턴스 실행 시 해당 인스턴스에 대한 블록 디바이스 매핑을 변경하면 해당 업데이트는 AMI의 블록 디바이스 매핑을 덮어쓰거나 병합됩니다. 단,

제한

- 루트 볼륨에서는 다음 항목만 수정할 수 있습니다. 볼륨 크기, 볼륨 유형 및 [Delete on Termination] 플래그.
- EBS 볼륨을 수정할 때 크기는 줄일 수 없습니다. 그러므로 AMI의 블록 디바이스 매핑에서 지정된 스냅샷과 크기가 같거나 큰 스냅샷을 지정해야 합니다.

목차

- [인스턴스 실행 시 블록 디바이스 매핑 업데이트 \(p. 667\)](#)
- [실행 인스턴스의 블록 디바이스 매핑 업데이트 \(p. 668\)](#)
- [인스턴스 블록 디바이스 매핑에서 EBS 볼륨 보기 \(p. 668\)](#)
- [인스턴스 스토어 볼륨용 인스턴스 블록 디바이스 매핑 보기 \(p. 669\)](#)

인스턴스 실행 시 블록 디바이스 매핑 업데이트

실행 시 인스턴스에 EBS 볼륨 및 인스턴스 스토어 볼륨을 추가할 수 있습니다. 인스턴스의 블록 디바이스 매핑을 업데이트해도 인스턴스가 실행된 AMI의 블록 디바이스 매핑이 영구적으로 변경되는 것은 아님에 주의하십시오.

콘솔을 사용하여 인스턴스에 볼륨을 추가하려면

1. Amazon EC2 콘솔을 엽니다.
2. 대시보드에서 [Launch Instance]를 선택합니다.
3. [Choose an Amazon Machine Image (AMI)] 페이지에서 사용할 AMI를 선택하고 [Select]를 선택합니다.
4. 마법사 안내에 따라 [Choose an Instance Type] 및 [Configure Instance Details] 설정을 완료합니다.
5. Add Storage 페이지에서 루트 볼륨, EBS 볼륨 및 인스턴스 스토어 볼륨을 다음과 같이 수정할 수 있습니다.
 - 루트 볼륨 크기를 변경하려면 [Type] 열 아래에 있는 [Root] 볼륨으로 이동한 다음 [Size] 필드를 변경합니다.
 - 인스턴스를 실행하는 데 사용된 AMI의 블록 디바이스 매핑에서 지정된 EBS 볼륨을 표시하지 않으려면 해당 볼륨을 찾아 Delete 아이콘을 클릭합니다.
 - EBS 볼륨을 추가하려면 [Add New Volume]을 선택하고 [Type] 목록에서 [EBS]를 선택한 다음 필드 (Device, Snapshot 등)를 작성합니다.
 - 인스턴스가 실행된 AMI의 블록 디바이스 매핑에서 지정된 인스턴스 스토어 볼륨을 표시하지 않으려면 해당 볼륨으로 이동한 다음 [Delete] 아이콘을 선택합니다.
 - 인스턴스 스토어 볼륨을 추가하려면, [Add New Volume]을 선택하고, [Type] 목록에서 [Instance Store]를 선택한 다음 [Device]에서 디바이스 이름을 선택합니다.
6. 나머지 마법사 페이지를 완료한 다음 [Launch]를 선택합니다.

명령줄을 사용하여 인스턴스에 볼륨을 추가하려면

`run-instances` AWS CLI 명령을 사용하여 인스턴스에 블록 디바이스 매핑을 지정합니다.

다음 파라미터를 사용하여 블록 디바이스 매핑을 지정합니다.

```
--block-device-mappings [mapping, ...]
```

예를 들어, EBS 기반 AMI가 다음 블록 디바이스 매핑을 지정한다고 가정하면,

- `/dev/sdb=ephemeral0`
- `/dev/sdh=snap-1234567890abcdef0`
- `/dev/sdj=:100`

이 AMI에서 실행된 인스턴스에 `/dev/sdj`가 연결되지 않게 하려면 다음 매핑을 사용합니다.

```
{  
    "DeviceName": "/dev/sdj",  
    "NoDevice": ""  
}
```

`/dev/sdh`의 크기를 300GiB로 늘리려면, 다음 매핑을 지정합니다. 디바이스 이름을 지정하면 볼륨을 충분히 확인할 수 있기 때문에 `/dev/sdh`에 스냅샷 ID를 지정할 필요가 없음에 유의하십시오.

```
{  
    "DeviceName": "/dev/sdh",  
    "NoDevice": ""  
}
```

```
    "Ebs": {  
        "VolumeSize": 300  
    }  
}
```

추가 인스턴스 스토어 볼륨 /dev/sdc를 연결하려면 다음 매핑을 지정합니다. 다중 인스턴스 스토어 볼륨을 지원하지 않는 인스턴스 유형의 경우 이 매핑은 영향을 미치지 않습니다.

```
{  
    "DeviceName": "/dev/sdc",  
    "VirtualName": "ephemeral1"  
}
```

또는 [New-EC2Instance](#) 명령(Windows PowerShell용 AWS 도구)과 함께 `-BlockDeviceMapping` 파라미터를 사용할 수 있습니다.

실행 인스턴스의 블록 디바이스 매핑 업데이트

다음 [modify-instance-attribute](#) AWS CLI 명령을 사용하여 실행 인스턴스의 블록 디바이스 매핑을 업데이트 할 수 있습니다. 이 속성을 변경하기 전에 인스턴스를 종지할 필요는 없습니다.

```
$ aws ec2 modify-instance-attribute --instance-id i-1a2b3c4d --block-device-mappings  
file://mapping.json
```

예를 들어, 인스턴스 종료 시 루트 볼륨을 유지하려면 mapping.json에서 다음을 지정합니다.

```
[  
    {  
        "DeviceName": "/dev/sda1",  
        "Ebs": {  
            "DeleteOnTermination": false  
        }  
    }  
]
```

또는 [Edit-EC2InstanceAttribute](#) 명령(Windows PowerShell용 AWS 도구)과 함께 `-BlockDeviceMapping` 파라미터를 사용할 수 있습니다.

인스턴스 블록 디바이스 매핑에서 EBS 볼륨 보기

인스턴스에 매핑된 EBS 볼륨을 쉽게 확인할 수 있습니다.

Note

2009-10-31 API 릴리스 이전에 실행된 인스턴스의 경우 AWS는 블록 디바이스 매핑을 표시할 수 없습니다. 반드시 해당 볼륨을 분리 후 다시 연결해야 블록 디바이스 매핑이 표시될 수 있습니다.

콘솔을 사용하여 인스턴스의 EBS 볼륨을 보려면

1. Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Instances]를 선택합니다.
3. 검색 창에 [Root Device Type]을 입력한 다음 [EBS]를 선택합니다. 이렇게 하면 EBS 기반 인스턴스 목록이 표시됩니다.
4. 원하는 인스턴스를 선택한 다음 [Description] 탭에 표시되는 세부 정보를 확인합니다. 루트 디바이스에서 최소한으로 사용 가능한 정보는 다음과 같습니다.
 - 루트 디바이스 유형 (ebs)

- 루트 디바이스(예: /dev/sda1)
- Block devices(예: /dev/sda1, /dev/sdh 및 /dev/sdj)

인스턴스가 블록 디바이스 매핑을 사용하여 추가 EBS 볼륨으로 실행된 경우 [Block devices] 필드에 루트 디바이스와 함께 해당 추가 볼륨도 표시됩니다. (이 대화 상자에는 인스턴스 스토어 볼륨이 표시되지 않는다는 것에 유의하십시오.)

5. 블록 디바이스와 관련한 추가 정보를 표시하려면 [Block devices] 옆의 내용을 선택합니다. 그러면 블록 디바이스와 관련한 다음 정보가 표시됩니다.

- [EBS ID](vol-xxxxxxxx)
- 루트 디바이스 유형 (ebs)
- 연결 시간 (yyyy-mmThh:mm:ss.ssTZD)
- 블록 디바이스 상태 (attaching, attached, detaching, detached)
- 종료 시 삭제 여부 (Yes, No)

명령줄을 사용하여 인스턴스의 EBS 볼륨을 보려면

[describe-instances](#)(AWS CLI) 명령 또는 [Get-EC2Instance](#)(Windows PowerShell용 AWS 도구) 명령을 사용하여 인스턴스용 블록 디바이스 매핑에 EBS 볼륨을 표시합니다.

인스턴스 스토어 볼륨용 인스턴스 블록 디바이스 매핑 보기

인스턴스에 대한 블록 디바이스 매핑을 볼 때 인스턴스 스토어 볼륨이 아닌 EBS 볼륨만 확인할 수 있습니다. 인스턴스 메타데이터를 사용하여 전체 블록 디바이스 매핑에 대해 쿼리할 수 있습니다. 전체 인스턴스 메타데이터를 요청하기 위한 기본 URI는 <http://169.254.169.254/latest>/입니다.

우선, 실행 중인 인스턴스에 연결합니다.

실행 중인 인스턴스에서 이 쿼리를 사용하여 블록 디바이스 매핑을 가져옵니다.

```
$ curl http://169.254.169.254/latest/meta-data/block-device-mapping/
```

인스턴스에 대한 블록 디바이스 이름이 응답으로 제공됩니다. 예를 들어, 인스턴스 스토어 지원 m1.small 인스턴스에 대한 결과는 다음과 같습니다.

```
ami
ephemeral0
root
swap
```

인스턴스에서 보이는 것과 같이 AMI 디바이스가 루트 디바이스입니다. 인스턴스 스토어 볼륨의 이름은 ephemeral[0-23]입니다. 교체 디바이스는 페이지 파일입니다. 또한, EBS 볼륨을 매핑한 경우 ebs1, ebs2 등으로 표시됩니다.

블록 디바이스 매핑 내 개별 블록 디바이스에 대한 세부 정보를 확인하려면 여기에서와 같이 이전 쿼리에 이름을 추가합니다.

```
$ curl http://169.254.169.254/latest/meta-data/block-device-mapping/ephemeral0
```

자세한 내용은 [인스턴스 메타데이터 및 사용자 데이터 \(p. 321\)](#) 섹션을 참조하십시오.

퍼블릭 데이터 세트 사용

Amazon Web Services에서는 AWS 클라우드 기반 애플리케이션에 완벽하게 통합될 수 있는 퍼블릭 데이터 세트 리포지토리를 제공합니다. Amazon에서는 데이터 세트를 커뮤니티에 무료로 저장하고, 모든 AWS 서비스와 마찬가지로 애플리케이션에 사용된 컴퓨팅 및 스토리지에 대해서만 지불합니다.

목차

- [퍼블릭 데이터 세트 개념 \(p. 670\)](#)
- [퍼블릭 데이터 세트 찾기 \(p. 670\)](#)
- [스냅샷에서 퍼블릭 데이터 세트 볼륨 생성 \(p. 671\)](#)
- [퍼블릭 데이터 세트 볼륨 연결 및 마운트 \(p. 672\)](#)

퍼블릭 데이터 세트 개념

이전에는 인간 게놈 및 미국 인구조사 데이터 매핑 같은 대규모 데이터 세트를 검색하고, 다운로드하고, 가공하고, 분석하려면 짧아도 몇 시간, 길면 며칠까지 걸렸습니다. 이제 모든 사용자가 EC2 인스턴스에서 이러한 데이터 세트에 액세스하고 몇 분 이내에 데이터에 대한 컴퓨팅을 시작할 수 있습니다. 전체 AWS 에코시스템을 활용해 다른 AWS 사용자와 쉽게 협업할 수도 있습니다. 예를 들어, 도구와 애플리케이션으로 서버 이미지를 제작하거나 이미 구축되어 있는 서버 이미지를 사용하여 데이터 세트를 분석할 수 있습니다. AWS는 이렇게 중요하고 유용한 데이터를 Amazon EC2와 같은 비용 효율적인 서비스로 호스팅함으로써 다양한 분야의 연구자들과 기업들이 더 빠르게 혁신을 이룰 수 있도록 하는 도구를 제공합니다.

자세한 내용은 [Public Data Sets on AWS Page](#)를 참조하십시오.

사용 가능한 퍼블릭 데이터 세트

현재 다음과 같은 범주의 퍼블릭 데이터 세트를 사용할 수 있습니다.

- 생물학 - 인간 게놈 프로젝트, GenBank 및 기타 콘텐츠를 포함합니다.
- 화학 - 다양한 버전의 PubChem 및 기타 콘텐츠를 포함합니다.
- 경제학 - 인구 조사 데이터, 노동 통계, 교통 통계 및 기타 콘텐츠를 포함합니다.
- 백과사전 - 다양한 출처의 Wikipedia 콘텐츠와 기타 콘텐츠를 포함합니다.

퍼블릭 데이터 세트 찾기

퍼블릭 데이터 세트를 사용하려면 먼저 데이터 세트를 찾은 다음 해당 데이터 세트를 호스팅할 형식을 결정해야 합니다. 데이터 세트는 Amazon EBS 스냅샷 또는 Amazon S3 버킷의 두 가지 형식으로 사용할 수 있습니다.

퍼블릭 데이터 세트를 찾고 해당 형식을 결정하려면

1. [Public Data Sets Page](#)로 이동하여 모든 사용 가능한 퍼블릭 데이터 세트 목록을 확인합니다. 이 페이지에 검색 문구를 입력하여 사용 가능한 퍼블릭 데이터 세트 목록을 쿼리할 수도 있습니다.
2. 데이터 세트 이름을 클릭하여 해당 정보 페이지를 표시합니다.
3. 데이터 세트 정보 페이지에서 스냅샷 ID 목록을 찾아 Amazon EBS 형식의 데이터 세트 또는 Amazon S3 URL을 식별합니다.

스냅샷 형식의 데이터 세트는 EC2 인스턴스에 연결할 새 EBS 볼륨을 생성하는 데 사용됩니다. 자세한 내용은 [스냅샷에서 퍼블릭 데이터 세트 볼륨 생성 \(p. 671\)](#) 섹션을 참조하십시오.

Amazon S3 형식의 데이터 세트의 경우 AWS SDK 또는 HTTP 쿼리 API를 사용하여 정보에 액세스하거나 AWS CLI를 사용하여 인스턴스 간에 데이터를 복사하거나 동기화할 수 있습니다. 자세한 내용은 [Amazon S3 및 Amazon EC2 \(p. 658\)](#) 섹션을 참조하십시오.

Amazon EMR을 사용하여 퍼블릭 데이터 세트를 분석 및 작업할 수도 있습니다. 자세한 내용은 [What is Amazon EMR?](#)을 참조하십시오.

스냅샷에서 퍼블릭 데이터 세트 볼륨 생성

스냅샷 형식의 공개 데이터 세트를 사용하려면 공개 데이터 세트의 스냅샷 ID를 지정하여 새 볼륨을 생성합니다. 다음과 같이 AWS Management Console을 사용하여 새 볼륨을 생성할 수 있습니다. 원하는 경우 [create-volume](#) AWS CLI 명령을 대신 사용할 수 있습니다.

스냅샷에서 퍼블릭 데이터 세트 볼륨을 생성하려면

1. Amazon EC2 콘솔을 엽니다.
2. 탐색 모음에서 데이터 세트 스냅샷이 있는 리전을 선택합니다.

Important

스냅샷 ID는 단일 리전으로 제한되므로 다른 리전에 있는 스냅샷에서 볼륨을 생성할 수 없습니다. 또한 동일한 가용 영역에 있는 인스턴스에만 EBS 볼륨을 연결할 수 있습니다. 자세한 내용은 [리소스 위치 \(p. 673\)](#) 섹션을 참조하십시오.

이 볼륨을 다른 리전에 생성해야 하는 경우 스냅샷을 필요한 리전에 복사한 다음 해당 리전의 볼륨에 스냅샷을 복원하면 됩니다. 자세한 내용은 [Amazon EBS 스냅샷 복사 \(p. 610\)](#) 섹션을 참조하십시오.

3. 탐색 창에서 [Volumes]를 클릭합니다.
4. 위쪽 창에서 [Create Volume]을 클릭합니다.
5. [Create Volume] 대화 상자의 [Type] 목록에서 범용 SSD, 프로비저닝된 IOPS SSD 또는 Magnetic을 선택합니다. 자세한 내용은 [Amazon EBS 볼륨 유형 \(p. 564\)](#) 섹션을 참조하십시오.
6. [Snapshot] 필드에 데이터 세트에 대한 스냅샷의 ID 또는 설명을 입력합니다. 제안 옵션 목록에서 스냅샷을 선택합니다.

Note

원하는 스냅샷 ID가 표시되지 않는 경우 Amazon EC2 콘솔에 다른 리전이 선택되어 있을 수 있습니다. [퍼블릭 데이터 세트 찾기 \(p. 670\)](#)에서 식별된 데이터 세트의 세부 정보 페이지에 리전이 지정되어 있지 않은 경우 데이터 세트가 미국 동부(버지니아 북부) 리전에 포함되어 있을 가능성이 높습니다.`us-east-1`

7. [Size] 필드에 볼륨의 크기(GiB 또는 TiB)를 입력하거나 스냅샷의 기본 크기가 적절한지 확인합니다.

Note

볼륨 크기와 스냅샷 ID를 모두 지정한 경우 크기는 스냅샷 크기 이상이어야 합니다. 볼륨 유형과 스냅샷 ID를 선택하면 볼륨의 최소 및 최대 크기를 [Size] 목록 옆에서 확인할 수 있습니다.

8. 프로비저닝된 IOPS SSD 볼륨의 경우 [IOPS] 필드에 볼륨이 지원하는 최대 초당 입출력 작업 수를 입력합니다.
9. [Availability Zone] 목록에서 인스턴스를 시작할 가용 영역을 선택합니다.

Important

동일한 가용 영역의 인스턴스에만 EBS 볼륨을 연결할 수 있습니다.

10. Yes, Create를 클릭합니다.

Important

[Step 7 \(p. 671\)](#)에서 크기를 지정하여 스냅샷의 기본 크기보다 큰 볼륨을 생성한 경우 볼륨의 파일 시스템을 확장하여 추가 공간을 활용할 수 있어야 합니다. 자세한 내용은 [Linux에서 EBS 볼륨의 크기, IOPS 또는 유형 수정 \(p. 590\)](#) 섹션을 참조하십시오.

퍼블릭 데이터 세트 볼륨 연결 및 마운트

새 데이터 세트 볼륨을 생성한 후 데이터에 액세스하려면 볼륨을 EC2 인스턴스에 연결해야 합니다. 또한 이 인스턴스가 새 볼륨과 동일한 가용 영역에 있어야 합니다. 자세한 내용은 [Amazon EBS 볼륨을 인스턴스에 연결 \(p. 576\)](#) 섹션을 참조하십시오.

볼륨을 인스턴스에 연결한 후 인스턴스에서 볼륨을 마운트해야 합니다. 자세한 내용은 [Amazon EBS 볼륨을 사용할 수 있도록 만들기 \(p. 577\)](#) 섹션을 참조하십시오.

리소스 및 태그

Amazon EC2는 사용자가 생성하여 사용할 수 있는 서로 다른 리소스를 제공합니다. 이러한 리소스에는 이미지, 인스턴스, 볼륨 및 스냅샷 등이 있습니다. 리소스를 생성하면 리소스에 고유 리소스 ID가 할당됩니다.

일부 리소스에는 사용자가 정의하는 값으로 태그를 붙일 수 있어 쉽게 정리하고 식별할 수 있습니다.

다음 주제에서는 리소스와 태그에 대한 설명과 이를 이용한 작업 방법에 대해 살펴보겠습니다.

항목

- [리소스 위치 \(p. 673\)](#)
- [리소스 ID \(p. 674\)](#)
- [리소스 목록화 및 필터링 \(p. 678\)](#)
- [Amazon EC2 리소스에 태그 지정 \(p. 681\)](#)
- [Amazon EC2 서비스 제한 \(p. 688\)](#)
- [Amazon EC2 사용 보고서 \(p. 689\)](#)

리소스 위치

다음 표는 Amazon EC2 리소스가 글로벌, 리전, 가용 영역 중 무엇에 해당하는지를 설명하고 있습니다.

Resource	유형	설명
AWS 계정	전 세계	모든 리전에 동일한 AWS 계정을 사용할 수 있습니다.
키 페어	글로벌 또는 리전	Amazon EC2를 사용하여 생성한 키 페어 이를 생성한 리전에서만 사용할 수 있습니다. 모든 리전에서 사용할 수 있는 RSA 키 페어를 생성하고 업로드할 수 있습니다. 자세한 내용은 Amazon EC2 키 페어 (p. 377) 을 참조하십시오.

Resource	유형	설명
Amazon EC2 리소스 식별자	리전	AMI ID, 인스턴스 ID, EBS 볼륨 ID, EBS 스냅샷 ID 등 각 리소스 식별자는 해당 리전에 둑여 있으며, 리소스를 생성한 리전에서만 사용할 수 있습니다.
사용자가 공급한 리소스 이름	리전	보안 그룹 이름, 키 페어 이름 등 각 리소스 이름은 해당 리전에 둑여 있으며, 리소스를 생성한 리전에서만 사용할 수 있습니다. 여러 리전에서 동일한 이름을 가진 리소스를 생성할 수는 있지만, 이 경우에도 각 리소스들이 서로 관계를 가지게 되는 것은 아닙니다.
AMI	리전	AMI는 Amazon S3 내에서 파일이 위치하고 있는 리전에 둑여 있습니다. 한 리전의 AMI를 다른 리전으로 복사할 수 있습니다. 자세한 내용은 AMI 복사 (p. 126) 섹션을 참조하십시오.
엘라스틱 IP 주소	리전	탄력적 IP 주소는 리전에 둑여 있으며 동일한 리전의 인스턴스에만 연결할 수 있습니다.
보안 그룹	리전	보안 그룹은 리전에 둑여 있으며 동일한 리전의 인스턴스에만 배정할 수 있습니다. 보안 그룹 규칙을 사용해서 인스턴스가 그 리전 바깥의 인스턴스와 통신하게 할 수는 없습니다. 다른 리전의 인스턴스에서 나오는 트래픽은 WAN 대역폭으로 간주됩니다.
EBS 스냅샷	리전	EBS 스냅샷은 리전에 둑여 있으며 동일한 리전에서 볼륨을 생성하는 데만 사용할 수 있습니다. 한 리전의 스냅샷을 다른 리전으로 복사할 수 있습니다. 자세한 내용은 Amazon EBS 스냅샷 복사 (p. 610) 섹션을 참조하십시오.
EBS 볼륨	가용 영역	Amazon EBS 볼륨은 가용 영역에 둑여 있으며 동일한 가용 영역의 인스턴스에만 연결될 수 있습니다.
인스턴스	가용 영역	인스턴스는 이를 실행한 가용 영역에 둑여 있습니다. 그러나 인스턴스 ID는 그 리전에 둑여 있습니다.

리소스 ID

리소스가 생성되면 각 리소스마다 고유 리소스 ID가 할당됩니다. 리소스 ID를 사용하여 Amazon EC2 콘솔에서 리소스를 확인할 수 있습니다. 명령줄 도구 또는 Amazon EC2 API를 사용하여 Amazon EC2로 작업할 경우 특정 명령의 리소스 ID가 필요합니다. 예를 들어, `stop-instances` AWS CLI 명령을 사용하여 인스턴스를 중지할 경우 명령에 인스턴스 ID를 지정해야 합니다.

리소스 ID 길이

리소스 ID는 리소스 식별자(예: 스냅샷은 snap) 다음에 하이픈과 고유한 문자와 숫자 조합이 오는 형식을 갖습니다. 2016년 1월부터 일부 Amazon EC2 및 Amazon EBS 리소스 유형에 더 긴 ID를 도입하고 있습니다. 영문자 문자 조합 길이는 8자 형식이었는데, 새 ID는 17자 형식입니다(예: 인스턴스 ID i-1234567890abcdef0).

지원되는 리소스 유형은 옵트 인 기간을 갖게 되는데 이 기간에 더 긴 ID 형식을 사용할 수 있습니다. 한 리소스 유형에 더 긴 ID를 사용하도록 설정한 후에는 생성하는 모든 새 리소스에 더 긴 ID가 생성됩니다. 단, 더 긴 ID 형식을 사용하지 않도록 설정할 수도 있습니다. 생성 후에는 리소스 ID가 변경되지 않으므로 ID가 더 짧은 기존 리소스에 영향이 가지 않습니다. 이와 마찬가지로 리소스 유형에 더 긴 ID를 사용하지 않도록 설정하면 더 긴 ID로 생성한 리소스에 영향이 가지 않습니다.

지원되는 모든 리소스 유형은 기한이 있는데, 이후에는 이 유형의 모든 새 리소스는 기본적으로 더 긴 ID 형식을 갖게 되어 더 긴 ID 형식을 취소하지 못합니다. 더 긴 ID의 설정 또는 설정 해제는 IAM 사용자 및 IAM 역할로 가능합니다. 기본적으로 IAM 사용자 또는 역할은 루트 사용자와 동일한 설정으로 지정됩니다.

계정을 생성한 시점에 따라, 지원되는 리소스 유형에 기본적으로 더 긴 ID를 사용할 수 있습니다. 하지만 각 리소스 유형에 대해 설정된 기한까지는 더 긴 ID 사용을 옵트아웃할 수 있습니다. 자세한 내용은 [Amazon EC2 FAQ](#)의 더 긴 EC2 및 EBS 리소스 ID를 참조하십시오.

더 긴 ID로 생성되는 리소스는 관련 리소스 유형을 볼 권한이 있는 모든 IAM 사용자와 IAM 역할에게 개별 설정과 상관없이 표시됩니다.

항목

- [더 긴 ID 작업 \(p. 675\)](#)
- [긴 ID 설정에 대한 액세스 제어 \(p. 677\)](#)

더 긴 ID 작업

자기 자신이나 다른 IAM 사용자, IAM 역할 또는 계정의 루트 사용자에 대한 더 긴 ID 설정을 보고 수정할 수 있습니다.

항목

- [더 긴 ID 설정 보기 및 수정 \(p. 675\)](#)
- [사용자 또는 역할의 더 긴 ID 설정 보기 및 수정 \(p. 677\)](#)

더 긴 ID 설정 보기 및 수정

Amazon EC2 콘솔 또는 AWS CLI를 사용하여 긴 ID를 지원하는 리소스 유형을 보고, 더 긴 ID 형식을 스스로 활성화하거나 비활성화할 수 있습니다. 이 섹션의 절차는 콘솔에 로그인되어 있거나 요청을 만드는 IAM 사용자 또는 IAM 역할에 적용되며, 전체 AWS 계정에는 적용되는 것은 아닙니다.

콘솔을 사용하여 더 긴 ID 설정을 확인 및 수정하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 화면 상단의 탐색 모음에는 현재 리전이 표시됩니다. 더 긴 ID 설정을 보거나 변경할 리전을 선택합니다. 리전 간 설정은 공유되지 않습니다.
3. 대시보드의 계정 속성에서 리소스 ID 길이 관리를 선택하십시오. 더 긴 ID를 지원하는 리소스 유형이 열거됩니다. 각 리소스 유형에 더 긴 ID를 사용하도록 자동으로 전환되는 날짜가 기한 열에 표시됩니다.
4. 지원되는 리소스 유형에 더 긴 ID 형식을 사용하려면 더 긴 ID 사용 열의 확인란을 선택합니다. 더 긴 ID 형식을 비활성화하려면 확인란 선택을 취소합니다.

Important

루트 사용자로 로그인했을 경우에는 IAM 사용자 또는 역할이 로그인하여 스스로 이러한 설정을 명시적으로 재정의하지 않는 한 이러한 설정이 전체 계정에 적용됩니다. 더 긴 ID로 생성되는 리소스는 관련 리소스 유형을 볼 권한이 있는 모든 IAM 사용자에게 개별 설정과 상관없이 표시됩니다.

AWS CLI를 사용하여 더 긴 ID 설정을 확인 및 수정하려면

지원되는 모든 리소스의 더 긴 ID 설정을 보려면 [describe-id-format](#) AWS CLI 명령을 사용하십시오.

```
aws ec2 describe-id-format
```

```
{  
    "Statuses": [  
        {  
            "Deadline": "2016-11-01T13:00:00.000Z",  
            "UseLongIds": false,  
            "Resource": "instance"  
        },  
        {  
            "Deadline": "2016-11-01T13:00:00.000Z",  
            "UseLongIds": true,  
            "Resource": "reservation"  
        },  
        {  
            "Deadline": "2016-11-01T13:00:00.000Z",  
            "UseLongIds": false,  
            "Resource": "volume"  
        },  
        {  
            "Deadline": "2016-11-01T13:00:00.000Z",  
            "UseLongIds": false,  
            "Resource": "snapshot"  
        }  
    ]  
}
```

이 결과는 요청을 생성하는 IAM 사용자, IAM 역할 또는 루트 사용자에게 적용되며, 전체 AWS 계정에 적용되는 것은 아닙니다. 위의 결과는 instance, reservation, volume 및 snapshot 리소스 유형의 경우, 더 긴 ID의 사용 또는 사용 해제가 가능하다는 것을 나타냅니다. reservation 리소스는 이미 활성화되어 있습니다. Deadline 필드는 해당 리소스에 더 긴 ID를 사용하도록 자동 전환되는 날짜(UTC)를 나타냅니다. 기한이 아직 없을 경우 이 값이 반환되지 않습니다.

지정한 리소스에 더 긴 ID를 사용하려면 [modify-id-format](#) AWS CLI 명령을 사용합니다.

```
aws ec2 modify-id-format --resource resource-type --use-long-ids
```

지정한 리소스에 더 긴 ID를 사용하지 않으려면 [modify-id-format](#) AWS CLI 명령을 사용합니다.

```
aws ec2 modify-id-format --resource resource-type --no-use-long-ids
```

루트 사용자로서 이런 작업을 사용 중인 경우에는 IAM 사용자 또는 역할이 스스로 이러한 설정을 명시적으로 재정의하지 않는 한 이러한 설정이 전체 계정에 적용됩니다. 이런 명령은 각 리전별로만 사용됩니다. 다른 리전의 설정을 수정하려면 명령에 `--region` 파라미터를 사용하십시오.

Note

Amazon EC2 API의 2015-10-01 버전에서 IAM 역할 자격 증명을 사용하여 `describe-id-format` 또는 `modify-id-format`을 호출하는 경우 그 결과는 특정 IAM 역할이 아니라 전체 AWS 계정에 적용됩니다. Amazon EC2 API의 현재 버전에서는 결과가 IAM 역할에만 적용됩니다.

또는 다음 명령을 사용할 수 있습니다.

ID 형식을 설명하려면

- [DescribeIdFormat](#) (Amazon EC2 API)
- [Get-EC2IdFormat](#) (Windows PowerShell용 AWS 도구)

ID 형식을 수정하려면

- [ModifyIdFormat](#) (Amazon EC2 API)

- [Edit-EC2IdFormat](#) (Windows PowerShell용 AWS 도구)

사용자 또는 역할의 더 긴 ID 설정 보기 및 수정

[describe-identity-id-format](#) 및 [modify-identity-id-format](#) AWS CLI 명령을 사용하여 지원되는 리소스 유형을 보고 계정의 특정 IAM 사용자, IAM 역할 또는 루트 사용자에 대해 더 긴 ID 설정을 활성화할 수 있습니다. 이런 명령을 사용하려면 IAM 사용자, IAM 역할 또는 루트 계정 사용자의 ARN을 요청에 지정해야 합니다. 예를 들어 계정 123456789012에서 역할 'EC2Role'의 ARN은 `arn:aws:iam::123456789012:role/EC2Role`입니다. 자세한 내용은 IAM 사용 설명서의 [보안 주체](#)를 참조하십시오.

특정 IAM 사용자 또는 IAM 역할에 대해 지원되는 모든 리소스의 더 긴 ID 설정을 보려면 다음 AWS CLI 명령을 사용하십시오.

```
aws ec2 describe-identity-id-format --principal-arn arn-of-iam-principal
```

특정 IAM 사용자 또는 IAM 역할에 대한 리소스 유형의 더 긴 ID 설정을 사용하려면 다음 AWS CLI 명령을 사용하십시오.

```
aws ec2 modify-identity-id-format --principal-arn arn-of-iam-principal --resource resource-type --use-long-ids
```

이런 명령은 요청에 지정된 ARN에 적용되고 해당 요청을 만든 IAM 사용자, IAM 역할 또는 루트 사용자에게는 적용되지 않습니다.

다음 AWS CLI 명령을 사용하여 모든 IAM 사용자, IAM 역할 및 계정의 루트 사용자의 더 긴 ID 설정을 활성화할 수 있습니다.

```
aws ec2 modify-identity-id-format --principal-arn all --resource resource-type --use-long-ids
```

또는 다음 명령을 사용할 수 있습니다.

ID 형식을 설명하려면

- [DescribeIdentityIdFormat](#)(Amazon EC2 API)
- [Get-EC2IdentityIdFormat](#)(Windows PowerShell용 AWS 도구)

ID 형식을 수정하려면

- [ModifyIdentityIdFormat](#)(Amazon EC2 API)
- [Edit-EC2IdentityIdFormat](#)(Windows PowerShell용 AWS 도구)

긴 ID 설정에 대한 액세스 제어

IAM 사용자와 역할은 관련된 IAM 정책을 통해 명시적으로 권한을 부여받지 않는 한, 기본적으로 `ec2:DescribeIdFormat`, `ec2:DescribeIdentityIdFormat`, `ec2:ModifyIdFormat` 및 `ec2:ModifyIdentityIdFormat` 작업을 사용할 수 있는 권한이 없습니다. 예를 들어 IAM 역할은 정책 명령문에서 "Action": "ec2:*" 요소를 통해 모든 Amazon EC2 작업을 사용할 수 있는 권한을 가질 수 있습니다.

IAM 사용자와 역할이 그들 스스로 또는 내 계정에 있는 다른 사용자와 역할에 대해 더 긴 리소스 ID 설정을 보거나 수정하지 못하게 하려면, IAM 정책에 다음 명령문을 포함시키십시오.

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Deny",
    "Action": [
      "ec2:ModifyIdFormat",
      "ec2:DescribeIdFormat",
      "ec2:ModifyIdentityIdFormat",
      "ec2:DescribeIdentityIdFormat"
    ],
    "Resource": "*"
  }
]
```

ec2:DescribeIdFormat, ec2:DescribeIdentityIdFormat, ec2:ModifyIdFormat 및 ec2:ModifyIdentityIdFormat 작업에 대한 리소스 수준 권한은 지원되지 않습니다.

리소스 목록화 및 필터링

사용자는 Amazon EC2 콘솔을 이용하여 리소스의 유형 목록을 획득할 수 있습니다. 사용자는 해당 명령 또는 API 작업을 이용하여 리소스의 각 유형 목록을 획득할 수 있습니다. 리소스가 많은 경우 사용자는 결과를 필터링하여 특정 기준에 부합하는 리소스만을 포함시킬 수 있습니다.

항목

- [고급 검색 \(p. 678\)](#)
- [콘솔을 이용하여 리소스 목록화 \(p. 679\)](#)
- [콘솔을 이용하여 리소스를 필터링 \(p. 679\)](#)
- [CLI 및 API를 이용하여 목록화 및 필터링 \(p. 680\)](#)

고급 검색

고급 검색을 사용하면 필터 조합을 이용한 검색으로 정밀한 결과를 얻을 수 있습니다. 필터링은 키워드, 사용자 지정 태그 키 및 사전 정의된 리소스 속성으로 수행될 수 있습니다.

사용 가능한 검색 유형:

- **키워드로 검색**

키워드로 검색하려면 검색하려는 키워드를 검색 상자에 입력 또는 붙여넣기한 다음 Enter를 누릅니다. 예를 들어, 특정 인스턴스를 검색하려면 인스턴스 ID를 입력합니다.

- **필드로 검색**

또한, 리소스와 관련이 있는 필드, 태그 및 속성으로 검색하는 것도 가능합니다. 예를 들어, 중지 상태인 모든 인스턴스를 검색하려면

1. 검색 상자에 **Instance State**를 입력합니다. 입력하기 시작하면 추천 필드 목록이 표시됩니다.
2. 목록에서 **[Instance State]**를 선택합니다.
3. 추천 값 목록에서 **[Stopped]**를 선택합니다.
4. 목록을 미세 조정하려면 검색 상자를 클릭하여 추가 검색 옵션을 선택합니다.

- **고급 검색**

사용자는 여러 필터를 추가하여 어드밴스 쿼리를 생성할 수 있습니다. 예를 들어, 태그별 검색을 하여 프로덕션 스택에서 실행 중인 Flying Mountain 프로젝트 인스턴스를 확인한 다음 속성별 검색을 하면 모든 t2.micro 인스턴스 또는 모든 us-west-2a 인스턴스 또는 두 인스턴스 모두가 표시됩니다.

- 역검색

사용자는 지정된 값에 일치하지 않는 리소스를 검색할 수 있습니다. 예를 들어, 종료되지 않은 모든 인스턴스를 목록화하려면 Instance State 필드로 검색한 다음 느낌표(!)와 함께 종료된 값을 접두어로 사용합니다.

- 부분 검색

필드별로 검색하는 경우 부분 문자열을 입력하여 해당 필드에 있는 문자열에 포함된 모든 리소스를 검색할 수 있습니다. 예를 들어, Instance Type로 검색한 다음 t2를 입력하면 모든 t2.micro, t2.small 또는 t2.medium 인스턴스를 검색할 수 있습니다.

- 정규식

필드의 값이 특정 패턴에 맞아야 하는 경우 정규식을 유용하게 활용할 수 있습니다. 예를 들어, 이름 태그 별로 검색한 다음 ^s.*를 입력하면 's'로 시작하는 이름 태그를 지닌 모든 인스턴스가 표시됩니다. 정규식 검색은 대소문자를 구별하지 않습니다.

검색 결과를 확인한 이후에는 차후에 편리하게 참조할 수 있도록 URL을 즐겨찾기에 등록할 수 있습니다. 인스턴스가 수 천개 있는 경우 필터링하고 즐겨찾기를 등록하면 검색을 반복할 필요가 없어 시간을 크게 줄일 수 있습니다.

검색 필터 결합

일반적으로, 키 필드가 동일한 다중 필터(예를 들어, tag:Name, search, Instance State)는 자동으로 OR로 조인됩니다. AND로 조인되면 대부분의 필터가 논리에 맞지 않기 때문에 이 설정은 의도적인 것입니다. 예를 들어, Instance State=running AND Instance State=stopped로 검색하면 검색 결과가 제공되지 않을 것입니다. 많은 경우에 서로 다른 키 필드에서 보완적인 검색 용어를 사용함으로써 검색 결과를 조정할 수 있고 이 경우 AND 규칙이 자동으로 대체 적용됩니다. tag: Name:=All values and tag:Instance State=running으로 검색을 수행하면 두 기준 모두를 만족하는 검색 결과가 제공됩니다. 결과를 미세 조정하기 위해서는 검색 결과가 만족스러울 때까지 필터를 하나씩 제거하면 됩니다.

콘솔을 이용하여 리소스 목록화

사용자는 콘솔을 이용하여 자주 사용하는 Amazon EC2 리소스의 유형 목록을 확인할 수 있습니다. 추가 리소스를 확인하려면 명령줄 인터페이스 또는 API 작업을 사용합니다.

콘솔을 이용하여 EC2 리소스를 목록화하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [AMIs] 또는 [Instances] 등 리소스에 해당하는 옵션을 선택합니다.
3. 이 페이지는 사용 가능한 모든 리소스를 표시합니다.

콘솔을 이용하여 리소스를 필터링

사용자는 Amazon EC2 콘솔을 이용하여 자주 사용하는 리소스 유형을 필터링 및 정렬할 수 있습니다. 예를 들어, 인스턴스 페이지의 검색 창을 사용하여 태그, 속성 또는 키워드별로 인스턴스를 정렬할 수 있습니다.

또한, 사용자는 각 페이지에서 필드를 검색하여 특정 속성 또는 값이 있는 리소스를 검색할 수 있습니다. 정규식을 사용하여 부분 또는 다중 문자열을 검색하는 것도 가능합니다. 예를 들어, MySG 보안 그룹을 사용하는 모든 인스턴스를 검색하여 검색 필드에 MySG를 입력합니다. 그러면 MySG2 및 MySG3 등 문자열의 일부로 MySG가 있는 모든 값이 검색 결과에 표시됩니다. 결과를 MySG만으로 제한하려면 검색 필드에 \bMySG\b를 입력합니다. 유형이 m1.small 또는 m1.large인 모든 인스턴스를 목록화하려면 검색 필드에 m1.small|m1.large를 입력합니다.

상태가 `available`인 `us-east-1b` 가용 영역에서 볼륨을 목록화하려면

1. 탐색 창에서 [Volumes]를 선택합니다.
2. 검색 상자를 클릭하고 메뉴에서 [Attachment Status]를 선택한 다음 [Detached]를 선택합니다. (분리된 볼륨은 동일 가용 영역에 있는 인스턴스에 연결하는 데 사용될 수 있습니다.)
3. 검색 상자를 다시 클릭한 다음 [State]를 선택하고 [Available]을 선택합니다.
4. 검색 상자를 다시 클릭한 다음 [Availability Zone]을 선택하고 `us-east-1b`를 선택합니다.
5. 이 기준에 부합하는 모든 볼륨이 표시됩니다.

퍼블릭 64비트 Amazon EBS 지원 Linux AMI를 목록화하려면

1. 탐색 창에서 [AMIs]를 선택합니다.
2. [Filter] 창에서 [Public images], [EBS images]를 선택한 다음 [Filter] 목록에서 [your Linux distribution][]를 선택합니다.
3. 검색 필드에 `x86_64`를 입력합니다.
4. 이 기준에 부합하는 모든 AMI가 표시됩니다.

CLI 및 API를 이용하여 목록화 및 필터링

각 리소스 유형에는 사용자가 해당 유형의 리소스를 목록화하기 위해 사용할 수 있는 해당 CLI 명령 또는 API 요청이 있습니다. 예를 들어, 사용자는 `ec2-describe-images` 또는 `DescribeImages`를 이용하여 Amazon 머신 이미지(AMI)를 목록화할 수 있습니다. 응답에는 모든 리소스에 대한 정보가 포함됩니다.

결과 리소스 목록은 길이기 길 수 있기 때문에 사용자는 결과를 필터링하여 특정 기준에 부합하는 리소스만을 포함시킬 수 있습니다. 사용자는 다중 필터 값을 지정할 수 있고 다중 필터를 지정하는 것도 가능합니다. 예를 들어, 유형이 `m1.small` 또는 `m1.large`인 모든 인스턴스 및 인스턴스가 종료될 때 삭제되도록 설정된 연결된 EBS 볼륨을 목록화할 수 있습니다. 인스턴스가 결과에 포함되려면 모든 필터 기준에 부합해야 합니다.

사용자는 또한 필터 값과 함께 와일드카드를 사용할 수 있습니다. 별표(*)는 0개 이상의 문자에 해당하고 물음표(?)는 정확히 1문자에 해당합니다. 예를 들어, 필터 값으로 `*database*`를 사용하면 설명에 `database`가 포함된 모든 EBS 스냅샷이 표시됩니다. 필터 값으로 `database`를 지정한 경우 설명이 `database`와 일치하는 스냅샷만이 반환됩니다. 필터 값은 대소문자를 구분합니다. Amazon은 정확한 문자열 매치 또는 하위문자열 매치(와일드카드 포함)를 지원합니다. 리소스 결과 목록이 긴 경우에는 정확한 문자열 필터를 사용하면 응답 반환 속도가 더 빨라집니다.

검색에는 와이드카드 문자의 리터럴 값이 포함될 수 있고 문자 앞에 백슬래시를 사용하면 벗어날 수 있습니다. 예를 들어, `*amazon\?\\` 값은 리터럴 문자열 `*amazon?\`을 검색합니다.

Amazon EC2 리소스별로 지원되는 필터 목록은 해당 문서를 참조하십시오.

- AWS CLI의 경우 [AWS Command Line Interface Reference](#)의 관련 `describe` 명령을 참조하십시오.
- Windows PowerShell의 경우 [Windows PowerShell용 AWS 도구 Reference](#)의 관련 `Get` 명령을 참조하십시오.
- Query API의 경우 [Amazon EC2 API Reference](#)의 관련 `Describe API` 작업을 참조하십시오.

Note

설명 작업의 응답에는 리소스에 적용된 모든 태그에 대한 정보가 포함됩니다. 하지만 다중 리소스를 설명할 때는 최종 결과가 일치하여 모든 태그가 반환되지 않을 수도 있습니다. 따라서 리소스 태그를 확인하려면 단일 리소스를 설명하십시오.

Amazon EC2 리소스에 태그 지정

원한다면 고유의 메타데이터를 태그의 형태로 각 리소스에 배정하여 인스턴스, 이미지 및 기타 Amazon EC2 리소스를 쉽게 관리할 수 있습니다. 이 주제에서는 태그를 설명하고 태그를 생성하는 방법을 보여 줍니다.

목차

- [태그 기본 사항 \(p. 681\)](#)
- [리소스에 태그 지정 \(p. 681\)](#)
- [태그 제한 \(p. 683\)](#)
- [리소스에 결제용 태그 지정 \(p. 683\)](#)
- [콘솔을 사용한 태그 작업 \(p. 684\)](#)
- [CLI 또는 API를 사용한 태그 작업 \(p. 687\)](#)

태그 기본 사항

태그를 사용하면 용도, 소유자 또는 환경을 기준으로 하는 등 AWS 리소스를 다양한 방식으로 분류할 수 있습니다. 할당한 태그에 따라 특정 리소스를 빠르게 식별할 수 있기 때문에 이 기능은 동일 유형의 리소스가 많을 때 유용합니다. 각 태그는 사용자가 정의하는 키와 선택적 값으로 구성됩니다. 예를 들어, 계정의 Amazon EC2 인스턴스에 대해 각 인스턴스의 소유자나 스택 수준을 추적하는 데 도움이 되는 태그 세트를 정의할 수 있습니다. 각 리소스 유형에 대한 요건을 충족하는 태그 키 세트를 고안하는 것이 좋습니다. 일관된 태그 키 세트를 사용하면 리소스를 보다 쉽게 관리할 수 있습니다. 추가하는 태그에 따라 리소스를 검색하고 필터링할 수 있습니다.

다음 다이어그램은 태그 지정 방식을 설명합니다. 이 예에서는 `owner`와 `stack`이라는 두 태그를 각 인스턴스에 배정했습니다. 또한 각 태그에는 연결된 값이 있습니다.

태그는 Amazon EC2에는 의미가 없으며 엄격하게 문자열로 해석됩니다. 또한 태그는 리소스에 자동으로 배정되지 않습니다. 태그 키와 값을 편집할 수 있으며 언제든지 리소스에서 태그를 제거할 수 있습니다. 태그의 값을 빈 문자열로 설정할 수 있지만 태그의 값을 Null로 설정할 수는 없습니다. 해당 리소스에 대해 키가 기존 태그와 동일한 태그를 추가하는 경우 새 값이 이전 값을 덮어씁니다. 리소스를 삭제하면 리소스 태그도 삭제됩니다.

AWS Management Console, AWS CLI, Amazon EC2 API를 사용하여 태그 관련 작업을 수행할 수 있습니다.

AWS Identity and Access Management(IAM)을 사용하는 경우 AWS 계정에서 태그를 생성, 편집 또는 삭제할 수 있는 권한이 있는 사용자를 제어할 수 있습니다. 자세한 내용은 [Amazon EC2 리소스에 대한 액세스 제어 \(p. 398\)](#) 단원을 참조하십시오.

리소스에 태그 지정

계정에 이미 존재하는 대부분의 Amazon EC2 리소스에 태그를 지정할 수 있습니다. 아래의 [표 \(p. 682\)](#)에 태그 지정을 지원하는 리소스가 나와 있습니다.

Amazon EC2 콘솔을 사용하는 경우, 관련 리소스 화면에서 [Tags] 탭을 사용하여 리소스에 태그를 적용하거나 [Tags] 화면을 사용할 수 있습니다. 일부 리소스 화면에서는 리소스를 생성할 때 리소스의 태그를 지정할 수 있습니다. 대부분의 경우, 콘솔은 리소스 생성 직후(리소스 생성 중이 아니라) 태그를 적용합니다.

Amazon EC2 API, AWS CLI 또는 AWS SDK를 사용하는 경우, `createTags` EC2 API 작업을 사용하여 기존 리소스에 태그를 적용할 수 있습니다. 또한 일부 리소스 생성 작업에서는 리소스 생성 시 리소스의 태그를 지정할 수 있습니다. 리소스 생성 도중 태그를 적용할 수 없는 경우, 리소스 생성 프로세스가 끌백됩니다. 이는 태그를 사용하여 리소스가 생성되거나 아예 리소스가 생성되지 않도록 하고 언제든 태그 지정되지 않은 리소스가 남지 않도록 하는 데 도움이 됩니다. 생성 시 리소스에 태그를 지정하면 리소스 생성 후 사용자 지정 태그 지정 스크립트를 실행할 필요가 없습니다.

다음 표는 태그 지정할 수 있는 Amazon EC2와 생성 시 태그 지정할 수 있는 리소스를 설명합니다.

Amazon EC2 리소스 태그 지정 지원

Resource	태그 지원	생성 시 태그 지원(EC2 API, AWS CLI, AWS SDK)
AMI	예	아니요
번들 작업	아니요	아니요
고객 게이트웨이	예	아니요
전용 호스트	아니요	아니요
DHCP 옵션	예	아니요
EBS 스냅샷	예	아니요
EBS 볼륨	예	예
외부 전용 인터넷 게이트웨이	아니요	아니요
탄력적 IP 주소	아니요	아니요
인스턴스	예	예
인스턴스 스토어 볼륨	해당 사항 없음	해당 사항 없음
인터넷 게이트웨이	예	아니요
키 페어	아니요	아니요
NAT 게이트웨이	아니요	아니요
네트워크 ACL	예	아니요
네트워크 인터페이스	예	아니요
배치 그룹	아니요	아니요
예약 인스턴스	예	아니요
예약 인스턴스 목록	아니요	아니요
라우팅 테이블	예	아니요
스팟 인스턴스 요청	예	아니요
보안 group-EC2-Classic	예	아니요
보안 group-VPC	예	아니요
서브넷	예	아니요
가상 프라이빗 게이트웨이	예	아니요
VPC	예	아니요
VPC 엔드포인트	아니요	아니요
VPC 흐름 로그	아니요	아니요
VPC 피어링 연결	예	아니요

Resource	태그 지원	생성 시 태그 지원(EC2 API, AWS CLI, AWS SDK)
VPN 연결	예	아니요

생성 시 인스턴스나 볼륨에 태그를 지정하기 위해 Amazon EC2 콘솔의 Amazon EC2 Launch Instances 마법사, [RunInstances](#) Amazon EC2 API 또는 [CreateVolume](#) Amazon EC2 API를 사용할 수 있습니다. Amazon EC2 콘솔의 [Volumes] 화면은 생성 시 태그 지정을 지원하지 않습니다.

IAM 정책에서 [CreateVolume](#) 및 [RunInstances](#) Amazon EC2 API 작업에 태그 기반 리소스 수준 권한을 적용하여 생성 시 리소스에 태그를 지정할 수 있는 사용자와 그룹에 대한 세분화된 제어를 구현할 수 있습니다. 리소스를 생성하면 태그가 즉시 적용되기 때문에 생성— 단계부터 리소스를 적절하게 보호할 수 있습니다. 따라서 태그를 기반으로 리소스 사용을 제어하는 리소스 권한이 즉시 발효됩니다. 이에 따라 더욱 정확한 리소스 추적 및 보고가 가능합니다. 새 리소스에서 태그 지정 사용을 적용하고 리소스에서 어떤 태그 키와 값이 설정되는지 제어할 수 있습니다.

IAM 정책에서 [CreateTags](#) 및 [DeleteTags](#) Amazon EC2 API 작업에 리소스 수준 권한을 적용하여 기존 리소스에서 어떤 태그 키와 값이 설정되는지 제어할 수도 있습니다. 자세한 내용은 [Amazon EC2 API 작업에 지원되는 리소스 수준 권한 \(p. 409\)](#) 및 [AWS CLI 또는 AWS SDK 작업을 위한 예제 정책 \(p. 431\)](#)을(를) 참조하십시오.

결제를 위한 리소스 태그 지정에 대한 자세한 내용은 AWS Billing and Cost Management 사용 설명서의 [비용 할당 태그 사용](#) 단원을 참조하십시오.

태그 제한

태그에 적용되는 기본 제한은 다음과 같습니다.

- 리소스당 최대 태그 수 - 50개
- 최대 키 길이 - UTF-8의 유니코드 문자 127자
- 최대 값 길이 - 유니코드 문자 255자(UTF-8)
- 태그 키와 값은 대/소문자를 구분합니다.
- 태그 이름이나 값에서 aws: 접두사는 사용하지 마십시오. 이 단어는 AWS용으로 예약되어 있습니다. 이 접두사가 지정된 태그 이름이나 값은 편집하거나 삭제할 수 없습니다. 이 접두사가 지정된 태그는 리소스당 태그 수 제한에 포함되지 않습니다.
- 태깅 스키마를 여러 서비스와 리소스에서 사용하게 될 경우 다른 서비스 또한 허용되는 문자에 대한 제한이 있을 수 있음을 유의하십시오. 일반적으로 허용되는 문자는 UTF-8로 표현할 수 있는 문자, 공백 및 숫자와 특수 문자 + - = . _ : / @입니다.

태그에만 기초하여 리소스를 종료, 중지 또는 삭제할 수 없습니다. 리소스 식별자를 지정해야 합니다. 예를 들어 [DeleteMe](#)라는 태그 키로 태그를 지정한 스냅샷을 삭제하려면 해당 스냅샷의 리소스 식별자(예: [DeleteSnapshots](#))를 지정하여 snap-1234567890abcdef0 작업을 사용해야 합니다.

퍼블릭 또는 공유 리소스에 태그를 지정할 수 있지만, 배정하는 태그는 AWS 계정에만 사용할 수 있으며 해당 리소스를 공유하는 다른 계정에는 사용할 수 없습니다.

모든 리소스에 태그를 지정할 수는 없습니다. 자세한 내용은 [Amazon EC2 리소스 태그 지정 지원 \(p. 682\)](#) 단원을 참조하십시오.

리소스에 결제용 태그 지정

태그를 사용하여 비용 구조를 반영하도록 AWS 청구서를 구성할 수 있습니다. 이렇게 하려면 가입하여 태그 키 값이 포함된 AWS 계정 청구서를 가져옵니다. 태그를 사용한 비용 할당 보고서 설정에 대한 자세한 내용

은 AWS 계정 결제 정보의 [월간 비용 할당 보고서](#) 단원을 참조하십시오. 결합된 리소스의 비용을 확인하려면 태그 키 값을 동일한 리소스에 따라 결제 정보를 구성할 수 있습니다. 예를 들어, 특정 애플리케이션 이름으로 여러 리소스에 태그를 지정한 다음 결제 정보를 구성하여 여러 서비스에 걸친 해당 애플리케이션의 총 비용을 볼 수 있습니다. 자세한 내용은 AWS Billing and Cost Management 사용 설명서의 [비용 할당 태그 사용](#)을 참조하십시오.

Note

방금 보고서를 활성화한 경우, 24시간 후에 이번 달의 데이터를 볼 수 있습니다.

콘솔을 사용한 태그 작업

Amazon EC2 콘솔을 사용하여 동일 리전의 모든 Amazon EC2 리소스에서 사용 중인 태그를 볼 수 있습니다. 리소스와 리소스 유형별로 태그를 볼 수 있으며, 지정된 태그와 연결되어 있는 각 리소스 유형의 항목 수를 볼 수 있습니다. 또한 Amazon EC2 콘솔을 사용하여 한 번에 하나 이상의 리소스에서 태그를 적용하거나 제거할 수도 있습니다.

리소스 나열 시 필터 사용에 대한 자세한 내용은 [리소스 목록화 및 필터링 \(p. 678\)](#) 섹션을 참조하십시오.

태그를 중앙에서 통합 생성 및 관리할 수 있는 AWS Management Console의 Tag Editor는 사용하기 쉽고 최상의 결과를 발휘합니다. 자세한 내용은 Getting Started with the AWS Management Console의 [Tag Editor 작업](#) 단원을 참조하십시오.

목차

- [태그 표시 \(p. 684\)](#)
- [개별 리소스에 대한 태그 추가 및 삭제 \(p. 685\)](#)
- [리소스 그룹에 태그 추가 및 삭제 \(p. 685\)](#)
- [인스턴스 시작 시 태그 추가 \(p. 686\)](#)
- [태그를 기준으로 리소스 목록 필터링 \(p. 686\)](#)

태그 표시

Amazon EC2 콘솔에는 두 가지 방법으로 태그를 표시할 수 있습니다. 개별 리소스에 대한 태그를 표시하거나 모든 리소스에 대한 태그를 표시할 수 있습니다.

개별 리소스에 대한 태그를 표시하려면

Amazon EC2 콘솔에서 리소스 관련 페이지를 선택하면 이러한 리소스의 목록이 표시됩니다. 예를 들어, 탐색 창에서 [Instances]를 선택하는 경우 콘솔에 Amazon EC2 인스턴스 목록이 표시됩니다. 이러한 목록 중 하나(예: 인스턴스)에서 리소스를 선택하는 경우 해당 리소스에서 태그를 지원하면 관련 태그를 보고 관리할 수 있습니다. 대부분의 리소스 페이지에서는 세부 정보 창의 [Tags] 탭에서 태그를 볼 수 있습니다.

키가 동일한 태그의 값을 모두 표시하는 열을 하나 리소스 목록에 추가할 수 있습니다. 이 열을 사용하면 태그별로 리소스 목록을 정렬하고 필터링할 수 있습니다. 리소스에 새 열을 추가하여 태그를 표시하는 방법에는 두 가지가 있습니다.

- [Tags] 탭에서 [Show Column]을 선택합니다. 새 열이 콘솔에 추가됩니다.
- [Show/Hide Columns] 기어 모양 아이콘을 선택하고 [Show/Hide Columns] 대화 상자의 [Your Tag Keys]에서 태그 키를 선택합니다.

모든 리소스에 대한 태그를 표시하려면

Amazon EC2 콘솔의 탐색 창에서 [Tags]를 선택하여 모든 리소스의 태그를 표시할 수 있습니다. 다음 이미지는 리소스 유형별로 사용 중인 모든 태그를 나열하는 [Tags] 창을 보여 줍니다.

개별 리소스에 대한 태그 추가 및 삭제

리소스 페이지에서 개별 리소스에 대한 태그를 직접 관리할 수 있습니다.

개별 리소스에 태그를 추가하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 모음에서 요구에 맞는 리전을 선택합니다. 일부 Amazon EC2 리소스는 리전 간에 공유될 수 있지만 그렇지 않은 리소스도 있으므로 잘 선택해야 합니다. 자세한 내용은 [리소스 위치 \(p. 673\)](#) 섹션을 참조하십시오.
3. 탐색 창에서 리소스 유형(예: [Instances])을 선택합니다.
4. 리소스 목록에서 리소스를 선택합니다.
5. 세부 정보 창에서 [Tags] 탭을 선택합니다.
6. [Add/Edit Tags] 버튼을 선택합니다.
7. [Add/Edit Tags] 대화 상자에서 각 태그에 대한 키와 값을 지정하고 [Save]를 선택합니다.

개별 리소스에서 태그를 삭제하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 모음에서 요구에 맞는 리전을 선택합니다. 일부 Amazon EC2 리소스는 리전 간에 공유될 수 있지만 그렇지 않은 리소스도 있으므로 잘 선택해야 합니다. 자세한 내용은 [리소스 위치 \(p. 673\)](#) 섹션을 참조하십시오.
3. 탐색 창에서 리소스 유형(예: [Instances])을 선택합니다.
4. 리소스 목록에서 리소스를 선택합니다.
5. 세부 정보 창에서 [Tags] 탭을 선택합니다.
6. [Add/Edit Tags]를 선택하고 태그의 [Delete] 아이콘을 선택한 다음 [Save]를 선택합니다.

리소스 그룹에 태그 추가 및 삭제

리소스 그룹에 태그를 추가하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 모음에서 요구에 맞는 리전을 선택합니다. 일부 Amazon EC2 리소스는 리전 간에 공유될 수 있지만 그렇지 않은 리소스도 있으므로 잘 선택해야 합니다. 자세한 내용은 [리소스 위치 \(p. 673\)](#) 섹션을 참조하십시오.
3. 탐색 창에서 [Tags]를 선택합니다.
4. 콘텐츠 창 맨 위에서 [Manage Tags]를 선택합니다.
5. [Filter]에서 태그를 추가할 리소스 유형(예: 인스턴스)을 선택합니다.
6. 리소스 목록에서 태그를 추가할 각 리소스 옆의 확인란을 선택합니다.
7. [Add Tag]에서 [Key] 및 [Value]에 태그 키와 값을 입력한 다음 [Add Tag]를 선택합니다.

Note

태그 키가 기존 태그와 동일한 새 태그를 추가하는 경우 새 태그가 기존 태그를 덮어씁니다.

리소스 그룹에서 태그를 제거하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.

2. 탐색 모음에서 요구에 맞는 리전을 선택합니다. 일부 Amazon EC2 리소스는 리전 간에 공유될 수 있지만 그렇지 않은 리소스도 있으므로 잘 선택해야 합니다. 자세한 내용은 [리소스 위치 \(p. 673\)](#) 섹션을 참조하십시오.
3. 탐색 창에서 [Tags]와 [Manage Tags]를 선택합니다.
4. 사용 중인 태그를 보려면 [Show/Hide Columns] 기어 모양 아이콘을 선택하고, [Show/Hide Columns] 대화 상자에서 조회할 태그 키를 선택한 다음 [Close]를 클릭합니다.
5. [Filter]에서 태그를 제거할 리소스 유형(예: 인스턴스)을 선택합니다.
6. 리소스 목록에서 태그를 제거할 각 리소스 옆에 있는 확인란을 선택합니다.
7. [Remove Tag]의 [Key] 상자에 태그의 이름을 입력한 다음 [Remove Tag]를 선택합니다.

인스턴스 시작 시 태그 추가

[Launch Wizard]를 사용하여 태그를 추가하려면

1. 탐색 모음에서 인스턴스를 시작할 리전을 선택합니다. 일부 Amazon EC2 리소스는 리전 간에 공유될 수 있지만 그렇지 않은 리소스도 있으므로 잘 선택해야 합니다. 필요에 따라 적합한 리전을 선택하십시오. 자세한 내용은 [리소스 위치 \(p. 673\)](#) 섹션을 참조하십시오.
2. [Launch Instance]를 선택합니다.
3. [Choose an Amazon Machine Image (AMI)] 페이지에서는 Amazon 머신 이미지(AMI)라 불리는 일련의 기본 구성들을 목록으로 표시합니다. 사용할 AMI를 선택하고 [Select]를 선택합니다. AMI 선택에 대한 자세한 내용은 [Linux AMI 찾기 \(p. 67\)](#) 섹션을 참조하십시오.
4. [Configure Instance Details] 페이지에서 필요에 따라 인스턴스 설정을 구성하고 [Next: Add Storage]를 선택합니다.
5. [Add Storage] 페이지에서 인스턴스에 대한 추가 스토리지 볼륨을 지정할 수 있습니다. 모두 마쳤으면 [Next: Add Tags]를 선택합니다.
6. [Add Tags] 페이지에서 인스턴스나 볼륨 또는 이 둘의 태그를 지정합니다. Add Tags를 선택하여 리소스에 한 개 이상의 태그를 인스턴스에 추가할 수 있습니다. 모두 마쳤으면 [Next: Configure Security Group]를 선택합니다.
7. [Configure Security Group] 페이지에서 소유하는 기존 보안 그룹 중 하나를 선택하거나 마법사를 통해 새 보안 그룹을 생성합니다. 작업을 마치면 [Review and Launch]를 선택합니다.
8. 설정을 검토합니다. 선택한 항목에 만족하면 [Launch]를 선택합니다. 기존 키 페어를 선택하거나 새 키 페어를 생성하고, 승인 확인란을 선택하고 [Launch Instances]를 선택합니다.

태그를 기준으로 리소스 목록 필터링

하나 이상의 태그 키와 태그 값에 따라 리소스 목록을 필터링할 수 있습니다.

태그를 기준으로 리소스 목록을 필터링하려면

1. 다음과 같이 태그 열을 표시합니다.
 - a. 리소스를 선택합니다.
 - b. [Details] 창에서 [Tags]를 선택합니다.
 - c. 목록에서 태그를 찾고 [Show Column]을 선택합니다.
2. 태그 열의 오른쪽 위에 있는 필터 아이콘을 선택하여 필터 목록을 표시합니다.
3. 태그 값을 선택하고 [Apply Filter]를 선택하여 결과 목록을 필터링합니다.

Note

필터에 대한 자세한 내용은 [리소스 목록화 및 필터링 \(p. 678\)](#) 단원을 참조하십시오.

CLI 또는 API를 사용한 태그 작업

다음을 사용하여 리소스에 대한 태그를 추가, 업데이트, 조회 및 삭제할 수 있습니다. 해당 설명서의 예제를 참조하십시오.

작업	AWS CLI	Windows PowerShell용 AWS 도구	API 작업
하나 이상의 태그를 추가하거나 덮어 씁니다.	create-tags	New-EC2Tag	CreateTags
하나 이상의 태그를 삭제합니다.	delete-tags	Remove-EC2Tag	DeleteTags
하나 이상의 태그에 대해 설명합니다.	describe-tags	Get-EC2Tag	DescribeTags

또한 태그에 따라 리소스 목록을 필터링할 수도 있습니다. 다음 예제는 [describe-instances](#) 명령으로 태그를 사용하여 인스턴스를 필터링하는 방법을 보여 줍니다.

Important

태그가 리소스에 즉시 적용되기 때문에 IAM 정책에서 사용하는 태그 기반 리소스 권한이 즉시 발효 됩니다. 하지만 다중 리소스를 설명할 때는 최종 결과가 일치하여 모든 태그가 반영되지 않을 수도 있습니다. 따라서 리소스 태그를 확인하려면 단일 리소스를 설명하십시오.

예제 1: 지정한 태그 키를 갖는 인스턴스에 대한 설명 제공

다음 명령은 태그 값과 상관없이 Stack 태그를 가진 인스턴스에 대한 설명을 제공합니다.

```
aws ec2 describe-instances --filters Name=tag-key,Values=Stack
```

예제 2: 지정한 태그 키를 가진 인스턴스에 대한 설명 제공

다음 명령은 Stack=production 태그를 가진 인스턴스에 대한 설명을 제공합니다.

```
aws ec2 describe-instances --filters Name=tag:Stack,Values=production
```

예제 3: 지정한 태그 값을 가진 인스턴스에 대한 설명 제공

다음 명령은 태그 키와 상관없이 값이 production인 태그를 가진 인스턴스에 대한 설명을 제공합니다.

```
aws ec2 describe-instances --filters Name=tag-value,Values=production
```

일부 리소스 생성 작업에서는 리소스를 생성할 때 태그를 지정할 수 있습니다. 다음 작업은 생성 시 태그 지정을 지원합니다.

작업	AWS CLI	Windows PowerShell용 AWS 도구	API 작업
하나 이상의 인스턴스를 시작합니다.	run-instances(인스턴스 실행)	New-EC2Instance(새 EC2 인스턴스)	RunInstances
EBS 볼륨을 생성합니다.	create-volume	New-EC2Volume	CreateVolume

다음 예제는 리소스를 생성할 때 태그를 적용하는 방법을 보여 줍니다.

예제 4: 인스턴스를 시작하고 인스턴스와 볼륨에 태그 적용

다음 명령은 인스턴스를 시작하고 키가 `webserver`이고 값이 `production`인 태그를 인스턴스에 적용합니다. 이 명령은 또 생성되는 EBS 볼륨(이 경우에는 루트 볼륨)에 키가 `cost-center`이고 값이 `cc123`인 태그를 적용합니다.

```
aws ec2 run-instances --image-id ami-abc12345 --count 1 --instance-type t2.micro --key-name MyKeyPair --subnet-id subnet-6e7f829e --tag-specifications 'ResourceType=instance,Tags=[{Key=webserver,Value=production}]' 'ResourceType=volume,Tags=[{Key=cost-center,Value=cc123}]'
```

시작 중에 인스턴스와 볼륨에 동일한 태그 키와 값을 적용할 수 있습니다. 다음 명령은 인스턴스를 시작하고 키가 `cost-center`이고 값이 `cc123`인 태그를 인스턴스와 생성되는 일체의 EBS 볼륨에 적용합니다.

```
aws ec2 run-instances --image-id ami-abc12345 --count 1 --instance-type t2.micro --key-name MyKeyPair --subnet-id subnet-6e7f829e --tag-specifications 'ResourceType=instance,Tags=[{Key=cost-center,Value=cc123}]' 'ResourceType=volume,Tags=[{Key=cost-center,Value=cc123}]'
```

예제 5: 볼륨 생성 및 태그 적용

다음 명령은 볼륨을 생성하고 2개의 태그(`purpose = production` 및 `cost-center = cc123`)를 적용합니다.

```
aws ec2 create-volume --availability-zone us-east-1a --volume-type gp2 --size 80 --tag-specifications 'ResourceType=volume,Tags=[{Key=purpose,Value=production},{Key=cost-center,Value=cc123}]'
```

Amazon EC2 서비스 제한

Amazon EC2는 사용 가능한 여러 가지 리소스를 제공합니다. 이러한 리소스로는 이미지, 인스턴스, 볼륨 및 스냅샷이 있습니다. AWS 계정을 생성하면 이러한 리소스에 대한 기본 제한이 리전별로 설정됩니다. 예를 들어 한 리전에서 시작할 수 있는 인스턴스 수에는 제한이 있습니다. 즉, 미국 서부(오레곤) 지역에서 인스턴스를 시작할 때 해당 요청에 의해 해당 리전의 현재 인스턴스 제한이 초과되어서는 안 됩니다.

Amazon EC2 콘솔은 Amazon EC2 및 Amazon VPC 콘솔이 관리하는 리소스의 제한 정보를 제공합니다. 사용자는 이러한 제한을 높이도록 요청할 수 있습니다. 제공되는 제한 정보를 사용하여 AWS 인프라를 관리하십시오. 제한 증가는 실제로 필요해질 시점보다 미리 요청하도록 계획하십시오.

기타 서비스의 제한에 대한 자세한 내용은 [AWS 서비스 제한](#)(Amazon Web Services 일반 참조)을 참조하십시오.

현재 제한 조회

Amazon EC2 콘솔의 [EC2 Service Limits] 페이지에서 Amazon EC2 및 Amazon VPC에서 리전별로 제공하는 리소스의 현재 제한을 조회할 수 있습니다.

현재 제한을 조회하려면 다음을 수행합니다.

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 모음에서 리전을 선택합니다.
3. 탐색 창에서 [Limits]를 선택합니다.

- 목록에서 리소스를 찾습니다. [Current Limit] 열에는 해당 리소스에 대한 계정의 현재 최대값이 표시됩니다.

제한 증가 요청

Amazon EC2 콘솔의 [Limits] 페이지에서 Amazon EC2 또는 Amazon VPC에서 리전별로 제공하는 리소스에 대한 제한 증가를 요청할 수 있습니다.

제한 증가를 요청하려면 다음을 수행합니다.

- <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
- 탐색 모음에서 리전을 선택합니다.
- 탐색 창에서 [Limits]를 선택합니다.
- 목록에서 리소스를 찾습니다. [Request limit increase]를 선택합니다.
- 제한 증가 양식에서 필수 필드를 기재합니다. 응답은 사용자가 지정한 연락 수단을 통해 제공됩니다.

Amazon EC2 사용 보고서

Amazon EC2가 제공하는 사용 보고서를 통해 인스턴스의 사용을 상세히 분석할 수 있습니다. 사용자 보고서의 데이터는 매일 여러 번 업데이트됩니다. AWS 계정, 리전, 가용 영역, 운영 체제, 인스턴스 유형, 구입 옵션, 테넌시, 태그 등으로 보고서를 필터링할 수 있습니다.

계정에 대한 사용 및 비용 데이터를 얻기 위해서는, 계정 자격 증명을 가지고 있어야 하며 해당 계정에 대한 리소스 및 태그를 포함하는 상세 결제 보고서 기능을 활성화해야 합니다. 통합 결제를 사용하고 있는 경우 지급인 계정 및 그와 관련된 모든 계정에 대한 데이터를 확인하려면 지급인 계정에 로그인해야 합니다. 통합 결제에 대한 자세한 내용은 [통합 결제로 여러 계정의 요금 지불](#)을 참조하십시오.

항목

- [제공되는 보고서 \(p. 689\)](#)
- [사용 보고서에 대한 설정 \(p. 689\)](#)
- [IAM 사용자에게 Amazon EC2 사용 보고서에 대한 액세스 권한 부여 \(p. 691\)](#)
- [인스턴스 사용 보고서 \(p. 691\)](#)
- [예약 인스턴스 사용률 보고서 \(p. 693\)](#)

제공되는 보고서

다음 보고서를 생성할 수 있습니다.

- [인스턴스 사용 보고서 \(p. 691\)](#). 이 보고서는 온디맨드 인스턴스, 스팟 인스턴스, 예약 인스턴스의 사용을 다룹니다.
- [예약 인스턴스 사용률 보고서 \(p. 693\)](#). 이 보고서는 용량 예약의 사용을 다룹니다.

보고서에 액세스하려면 AWS Management Console을 엽니다. 탐색 창에서 [Reports]를 선택한 다음 보고 싶은 보고서를 선택합니다.

사용 보고서에 대한 설정

시작하기 전에 다음 절차와 같이 리소스 및 태그를 포함하는 상세 결제 보고서를 활성화하십시오. 이 절차를 완료한 후에 AWC는 인스턴스에 대한 사용 데이터 수집을 시작합니다. 이미 상세 결제 보고서를 활성화한 경우는 그 활성화 이후부터 AWC가 수집한 사용 데이터에 액세스할 수 있습니다.

Important

이 절차를 완료하려면 AWS 계정 자격 증명을 사용해서 로그인해야 합니다. IAM 사용자 계정을 사용해서 로그인한 경우는 이 절차를 완료할 수 없습니다.

상세 결제 보고서 활성화 방법

1. 기존 Amazon S3 버킷을 선택해서 사용 데이터를 수신합니다. 버킷에는 사용자의 요금 데이터가 포함되어 있으므로 버킷에 대한 액세스를 안전하게 관리하십시오. 이 파일을 계속 보관할 필요는 없으며, 필요하지 않은 경우 이 파일을 즉시 삭제할 수도 있습니다. 버킷이 없는 경우 이를 다음과 같이 생성해야 합니다.
 - a. Amazon S3 콘솔을 엽니다.
 - b. [Create Bucket]을 선택합니다.
 - c. [Create a Bucket] 대화 상자에서 버킷 이름을 입력하고(예: username-ec2-usage-data) 리전을 선택한 후 [Create]를 선택합니다. 버킷 이름 관련 요구사항에 대한 자세한 내용은 [버킷 생성](#)(출처: Amazon Simple Storage Service 콘솔 사용 설명서)을 참조하십시오.
2. <https://console.aws.amazon.com/billing/home?#>에서 Billing and Cost Management 콘솔을 엽니다.
3. 탐색 창에서 [Preferences]를 선택합니다.
4. [Receive Billing Reports]를 선택합니다.
5. [Save to S3 Bucket]에서 Amazon S3 버킷의 이름을 지정합니다.
6. [Receive Billing Reports]에서 [sample policy]를 선택합니다. 상기 샘플 정책(sample policy)을 복사합니다. 샘플 정책은 사용자가 지정한 버킷 이름을 사용합니다.
7. Amazon S3 버킷에 사용자 데이터를 게시할 수 있는 AWS 권한을 부여합니다.
 - a. 다른 브라우저 탭에서 Amazon S3 콘솔을 엽니다. 버킷을 선택하고 [Properties]를 선택한 후 [Permissions]를 펼칩니다. [Permissions] 섹션에서 [Add bucket policy]를 선택합니다. 샘플 정책을 텍스트 영역에 붙여넣기하고 [Save]를 선택합니다. [Permissions] 섹션에서 [Save]를 선택합니다.
 - b. 샘플 정책이 있는 브라우저 탭으로 돌아와 [Verify]를 선택합니다.
8. [Report] 아래에서 [Detailed billing report with resources and tags]를 클릭합니다.
9. [Save preferences]를 선택합니다.

Note

보고서에서 사용자의 데이터를 볼 수 있을 때까지는 하루 정도가 걸릴 수 있습니다.

태그를 사용하여 인스턴스를 범주화합니다. 인스턴스에 태그를 추가한 후에는 이 태그에 대한 보고 기능을 활성화해야 합니다.

태그를 사용해서 사용 보고 기능을 활성화하는 방법

1. 인스턴스에 태그를 추가합니다. 최상의 결과를 얻기 위해 보고에 사용할 각 태그를 인스턴스 각각에 추가하도록 하십시오. 인스턴스에 태그를 추가하는 방법에 대한 자세한 내용은 [Amazon EC2 리소스에 태그 지정](#)(p. 681)을 참조하십시오.
2. <https://console.aws.amazon.com/billing/home?#>에서 Billing and Cost Management 콘솔을 엽니다.
3. 탐색 창에서 [Preferences]를 선택합니다.
4. [Report]에서 [Manage report tags]를 선택합니다.
5. 상기 페이지는 생성한 태그의 목록을 표시합니다. 인스턴스 사용 데이터의 필터링 또는 그룹화에 사용할 태그를 선택하고 [Save]를 클릭합니다. AWC는 인스턴스 사용 보고서에서 선택하지 않은 태그를 자동으로 제외합니다.

Note

AWC는 현재 월에 대한 데이터에만 변경사항을 적용합니다. 구성 변경이 유효하게 적용될 때까지 하루 정도가 걸릴 수 있습니다.

IAM 사용자에게 Amazon EC2 사용 보고서에 대한 액세스 권한 부여

기본적으로 IAM 사용자는 Amazon EC2 사용 보고서에 액세스할 수 없습니다. 따라서 이 보고서를 액세스할 수 있는 IAM 사용자 권한을 부여하는 IAM 정책을 생성할 필요가 있습니다.

다음 정책은 사용자가 두 가지 Amazon EC2 사용 보고서를 볼 수 있도록 허용합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2-reports:*",  
            "Resource": "*"  
        }  
    ]  
}
```

다음 정책은 사용자가 인스턴스 사용 보고서를 볼 수 있도록 허용합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2-reports:ViewInstanceUsageReport",  
            "Resource": "*"  
        }  
    ]  
}
```

다음 정책은 사용자가 예약 인스턴스 사용률 보고서를 볼 수 있도록 허용합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2-reports:ViewReservedInstanceUtilizationReport",  
            "Resource": "*"  
        }  
    ]  
}
```

자세한 내용은 IAM 사용 설명서에서 [권한 및 정책](#)을 참조하십시오.

인스턴스 사용 보고서

인스턴스 사용 보고서를 사용해서 인스턴스 사용 및 비용 추세를 볼 수 있습니다. 또한 인스턴스 사용이나 비용 중 하나로 사용 데이터를 볼 수 있으며, 사용 데이터를 시간별, 일별, 월별로 보도록 선택할 수 있습니다. 리전이나 가용 영역, 인스턴스 유형, AWS 계정, 플랫폼, 테넌시, 구매 옵션, 태그별로 보고서를 필터링하거나 그룹화할 수 있습니다. 보고서를 구성한 뒤 나중에 쉽게 다시 돌아갈 수 있도록 북마크를 지정할 수 있습니다.

인스턴스 사용 보고서를 생성함으로써 사용자가 답변을 할 수 있는 질문 중 일부 예는 다음과 같습니다.

- 각 인스턴스 유형의 인스턴스에 얼마나 비용을 지출하고 있습니까?
- 특정 부서가 인스턴스를 사용하고 있는 시간은 어느 정도입니까?

- 가용 영역에 걸쳐 사용자의 인스턴스 사용이 어떻게 분포되어 있습니까?
- AWS 계정에 걸쳐 사용자의 인스턴스 사용이 어떻게 분포되어 있습니까?

항목

- [보고서 포맷 \(p. 692\)](#)
- [인스턴스 사용 조회 \(p. 692\)](#)
- [사용자 지정 보고서 북마크 \(p. 693\)](#)
- [사용 데이터 내보내기 \(p. 693\)](#)

보고서 포맷

AWC는 사용자가 요청하는 사용 데이터를 그래프와 테이블로 표시합니다.

예를 들어 다음 그래프는 인스턴스 유형별로 비용을 표시합니다. 그래프에 대한 키는 어떤 색상이 어떤 인스턴스 유형을 가리키는지를 나타냅니다. 막대의 세그먼트에 대한 상세 정보를 확인하려면 그 부분에 마우스를 갖다 놓습니다.

이에 대응하는 테이블은 각 인스턴스 유형에 대한 열 1개를 표시합니다. 열 머리에는 그래프의 인스턴스 유형과 동일한 색상을 가지는 색상 밴드가 포함되어 있습니다.

인스턴스 사용 조회

다음 절차는 AWC가 제공하는 기능 중 일부를 사용해서 사용 보고서를 생성하는 방법을 설명합니다.

시작하기 전에 먼저 설정을 해야 합니다. 자세한 내용은 [사용 보고서에 대한 설정 \(p. 689\)](#) 섹션을 참조하십시오.

인스턴스 유형별로 인스턴스를 필터링 및 그룹화하는 방법

1. Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Reports]를 선택한 후 [EC2 Instance Usage Report]를 선택합니다.
3. [Unit]에 대한 옵션을 선택합니다. 인스턴스가 실행 중인 시간(시 단위)을 확인하기 위해 [Instance Hours]를 클릭합니다. 인스턴스 사용의 비용을 확인하려면 [Cost]를 선택합니다.
4. [Granularity] 및 [Time range]에 대한 옵션을 선택합니다.
 - 시간 범위에서 각 시간에 대해 요약된 데이터를 보려면 Hourly 세부 수준을 확인합니다. 시간별 데이터를 확인할 때 최대 2일의 시간 범위까지 선택할 수 있습니다.
 - 시간 범위에서 각 일에 대해 요약된 데이터를 보려면 Daily 세부 수준을 확인합니다. 일별 데이터를 확인할 때 최대 2개월의 시간 범위까지 선택할 수 있습니다.
 - 시간 범위에서 각 월에 대해 요약된 데이터를 보려면 Monthly 세부 수준을 확인합니다.
5. [Filter] 목록에서 [Instance Type]를 선택합니다. [Group by] 목록에서 [Instance Type]를 선택합니다.
6. 필터 영역에서 1개 이상의 인스턴스 유형을 선택한 후 [Update Report]를 선택합니다. 지정한 필터가 [Applied Filters] 아래에 나타납니다.

페이지 상단의 [Reports] 또는 [EC2 Management Console]을 선택하여 Amazon EC2 콘솔로 돌아올 수 있습니다.

태그에 기반한 인스턴스 사용 그룹화 방법

1. 인스턴스 사용 보고서 페이지를 엽니다.

2. [Unit]에 대한 옵션을 선택합니다. 인스턴스가 실행 중인 시간(시 단위)을 확인하기 위해 [Instance Hours]를 클릭합니다. 인스턴스 사용의 비용을 확인하려면 [Cost]를 선택합니다.
3. [Granularity] 및 [Time range]에 대한 옵션을 선택합니다.
 - 시간 범위에서 각 시간에 대해 요약된 데이터를 보려면 Hourly 세부 수준을 확인합니다. 시간별 데이터를 확인할 때 최대 2일의 시간 범위까지 선택할 수 있습니다.
 - 시간 범위에서 각 일에 대해 요약된 데이터를 보려면 Daily 세부 수준을 확인합니다. 일별 데이터를 확인할 때 최대 2개월의 시간 범위까지 선택할 수 있습니다.
 - 시간 범위에서 각 월에 대해 요약된 데이터를 보려면 Monthly 세부 수준을 확인합니다.
4. [Group by] 목록에서 [Tag]를 선택합니다.
5. [Key Name] 상자를 선택하고 목록에서 이름을 선택한 후 [Update Report]를 선택합니다. 이 목록에 항목이 없는 경우는 태그별 사용 보고 기능을 활성화해야 합니다. 자세한 내용은 [태그를 사용해서 사용 보고 기능을 활성화하는 방법 \(p. 690\)](#) 섹션을 참조하십시오.

사용자 지정 보고서 북마크

사용자 지정 보고서에 북마크를 설정하여 보고서를 다시 생성할 수 있습니다.

사용자 지정 보고서에 북마크를 설정하려면

1. 보고서에 대한 옵션과 필터를 선택합니다. 각 선택 내용에 따라 콘솔 URL에 파라미터가 추가됩니다. `granularity=Hourly` 및 `Filters=filter_list`를 예로 들 수 있습니다.
2. 브라우저를 사용하여 콘솔 URL을 북마크로 추가합니다.
3. 이렇게 만든 북마크를 사용하여 이후에 동일한 보고서를 생성할 수 있습니다.

사용 데이터 내보내기

데이터를 내보내서 다른 보고서에 보고서 그래프 또는 테이블을 포함할 수 있습니다.

사용 현황 데이터를 내보내려면

1. 보고서에 대한 옵션과 필터를 선택합니다.
2. 테이블의 사용 현황 데이터를 .csv 파일로 내보내려면 [Download]를 선택하고 [CSV Only]를 선택합니다.
3. 테이블의 그래픽 사용 현황 데이터를 .png 파일로 내보내려면 [Download]를 선택하고 [Graph Only]를 선택합니다.

예약 인스턴스 사용률 보고서

예약 인스턴스 사용률 보고서는 소유하고 있는 Amazon EC2 예약 인스턴스의 각 그룹 또는 버킷의 시간 경과에 따른 사용률을 보여줍니다. 각 버킷에는 리전, 인스턴스 유형, 계정, 플랫폼, 테넌시, 제공 유형의 고유한 조합이 있습니다. 보고서에서 다루는 시간 범위를 몇 주, 몇 개월, 1년 또는 3년까지 사용자 지정할 수 있습니다. 제공되는 데이터는 해당 계정에 대해 상세 결제 보고서를 활성화한 시기가 언제이냐에 따라 달라집니다 (참조: [사용 보고서에 대한 설정 \(p. 689\)](#)). 예약 인스턴스 사용률 보고서는 버킷에서 인스턴스 사용에 대한 예약 인스턴스 가격을 온디マン드 가격과 비교하고, 보고서가 다루는 시간 범위 동안의 비용 절감을 표시합니다.

계정에 대한 사용 및 비용 데이터를 얻기 위해서는, 계정 자격 증명을 가지고 있어야 하며 해당 계정에 대한 리소스 및 태그를 포함하는 상세 결제 보고서 기능을 활성화해야 합니다. 통합 결제를 사용하고 있으며 지금 계정에 로그인한 경우는 지급 계정 및 그와 관련된 모든 계정에 대한 데이터를 확인할 수 있습니다. 통합 결

제를 사용하고 있으며 그와 연결된 계정 중 하나에 로그인한 경우는 그 연결된 계정에 대한 데이터만 확인할 수 있습니다. 통합 결제에 대한 자세한 내용은 [통합 결제로 여러 계정의 요금 지불](#)을 참조하십시오.

Note

예약 인스턴스 버킷은 청구서가 계산되는 방식과 동일하게 EC2-VPC 및 EC2-Classic 네트워크 플랫폼 유형에 걸쳐 예약 인스턴스에 대한 데이터를 집계합니다. 또한 버킷의 예약 인스턴스는 서로 다른 선불 비용 및 시간별 비용을 가질 수 있습니다.

예약 인스턴스 사용률 보고서를 생성함으로써 사용자가 답변을 할 수 있는 질문 중 일부에는 다음과 같습니다.

- 예약 인스턴스를 얼마나 잘 활용하고 있습니까?
- 예약 인스턴스를 통해 비용을 절약하고 있습니까?

예약 인스턴스에 대한 자세한 내용은 [예약 인스턴스 \(p. 174\)](#)을 참조하십시오.

시작하기 전에 먼저 설정을 해야 합니다. 자세한 내용은 [사용 보고서에 대한 설정 \(p. 689\)](#) 섹션을 참조하십시오.

항목

- [보고서에 대해 알아보기 \(p. 694\)](#)
- [예약 인스턴스 사용률 조회 \(p. 695\)](#)
- [사용자 지정 보고서 북마크 \(p. 695\)](#)
- [사용 데이터 내보내기 \(p. 696\)](#)
- [옵션 참조 설명 \(p. 696\)](#)

보고서에 대해 알아보기

예약 인스턴스 사용률 보고서는 요청한 사용률 데이터를 그래프 및 테이블 형식으로 표시합니다.

보고서에 액세스하려면 AWS Management Console을 엽니다. 탐색 창에서 [Reports]를 선택한 후 [EC2 Reserved Instance Usage Report]를 선택합니다.

본 보고서는 버킷별로 주어진 기간 동안 예약 인스턴스 사용 데이터를 집계합니다. 본 보고서에서 테이블의 각 행은 버킷을 나타내며 다음 측정치를 제공합니다.

- Count - 본 보고서가 다루는 기간 동안 동시에 소유된 예약 인스턴스의 최대 개수
- Usage Cost(사용 비용) - 예약 인스턴스 버킷에 의해 처리되는 인스턴스 사용에 적용되는 총 예약 인스턴스 사용 비용
- Total Cost(총 비용) - 예약 인스턴스 버킷과 관련된 사용 기간 동안에 대한 사용 비용 및 분할 상환 선불 비용.

Note

사용자가 예약 인스턴스 마켓플레이스에서 판매한 예약 인스턴스가 버킷에 포함되어 있고 보고서가 다루는 기간 중 어느 시점에 해당 예약 인스턴스가 활성화된 경우, 버킷의 총 비용이 늘어나 절약되는 비용이 과소평가될 수 있습니다.

- Savings(비용 절약) - 보고서가 다루는 기간에 대한 사용에 있어 온디맨드 가격을 기준으로 드는 비용과 예약 인스턴스를 사용한 실제 비용(Total Cost) 간의 차이.
- Average Utilization(평균 사용률)-보고서가 다루는 기간 동안 예약 인스턴스 버킷에 대한 평균 시간별 사용률.
- Maximum Utilization(최대 사용률)-보고서가 다루는 기간 동안의 어떤 시간에 있어 가장 높은 사용률.

테이블의 각 행(예약 인스턴스 버킷)에 대하여, 이에 대응하는 그래프는 본 보고서에 대해 선택한 Time range 등안 선택한 Show 측정치에 기반한 데이터를 표시합니다. 그래프의 각 점은 한 시점에서의 측정치를 표시합니다. 보고서 옵션에 대한 자세한 내용은 [옵션 참조 설명 \(p. 696\)](#)을 참조하십시오.

테이블의 각 선택한 행의 모서리에 있는 색상 밴드는 그레프의 보고서 라인에 대응합니다. 행의 시작에 있는 체크박스를 선택하여 그레프에서 행을 표시할 수 있습니다.

기본적으로 예약 인스턴스 사용률 보고서는 모든 예약 인스턴스 버킷에 대해 지난 14일 동안의 데이터를 반환합니다. 그레프는 테이블의 첫 5개 버킷에 대한 평균 사용률을 표시합니다. 7일, 주, 월, 년 단위의 기간 동안 각기 다른 사용률(평균 사용률, 최대 사용률), 비용(총 비용, 사용 비용) 데이터 등을 표시하도록 그레프를 사용자 지정할 수 있습니다.

보고서 사용자 지정

예약 인스턴스 사용률 보고서는 Time range, Filter 옵션을 사용해서 사용자 지정할 수 있습니다.

Time range(시간 범위)는 일반적인 상대적 시간 범위를 나타내며 그 범위는 Last 7 Days(지난 7일)-Last 3 Years(지난 3년)입니다. 필요에 가장 잘 맞는 시간 범위를 선택하고 [Update Report]를 선택해서 변경사항을 적용합니다. 목록에 없는 시간 범위를 적용하려면 [Custom]을 선택하고 본 보고서가 다루게 되는 기간의 시작일과 종료일을 입력합니다.

[Filter]를 통해 예약 인스턴스 특성(리전, 인스턴스 유형, 계정, 플랫폼, 테넌시, 제공 유형) 중 하나 이상을 사용해서 예약 인스턴스 사용률 보고서를 필터링할 수 있습니다. 예를 들어 리전별, 특정 가용 영역별 또는 둘 다로 필터링을 할 수 있습니다. 리전별로 필터링하려면 [Regions]를 선택한 다음 보고서에 포함할 리전 및 가용 영역을 선택하고 [Update Report]를 선택합니다.

보고서는 필터가 적용되지 않은 경우 모든 결과를 반환합니다.

보고서 옵션에 대한 자세한 내용은 [옵션 참조 설명 \(p. 696\)](#)을 참조하십시오.

예약 인스턴스 사용률 조회

본 섹션에서는 그레프 및 테이블이 나타내는 예약 인스턴스 사용률의 요소를 자세히 설명합니다. 본 설명을 위해 테스트 데이터에 기반한 다음 보고서를 사용하게 됩니다.

예약 인스턴스 사용률 보고서는 지난 3년 동안 예약 인스턴스의 평균 사용률을 표시합니다. 본 보고서는 계정의 예약 인스턴스에 대한 다음 정보 및 그 사용 현황이 어떤지를 표시합니다.

- 평균 사용률

테이블 내 일부 예약 인스턴스만 잘 사용되었습니다. 평균과 다른 것은 각각 50%, 100% 활용되었던 네 개의 t2.micro 예약 인스턴스(행 2 및 3)였습니다.

- 최대 사용률

3년의 보고 기간 동안 모든 t2.micro 예약 인스턴스가 충분히 활용되었습니다. 나머지 예약 인스턴스는 사용률이 저조하여 만족스러운 절감 효과를 나타내지 않았습니다.

- 절감

본 보고서는 이 테스트 계정에 있어 온디맨드 인스턴스 대신 예약 인스턴스를 사용했을 때 미국 동부(버지니아 북부)의 t2.micro 인스턴스 네 개에 대한 비용 절감만 달성할 수 있음을 나타내고 있습니다. 나머지 예약 인스턴스는 적절한 할인 혜택을 제공하지 않았습니다.

사용자 지정 보고서 북마크

사용자 지정 보고서에 북마크를 설정하여 보고서를 다시 생성할 수 있습니다.

사용자 지정 보고서에 북마크를 설정하려면

1. 보고서에 대한 옵션과 필터를 선택합니다. 각 선택 내용에 따라 콘솔 URL에 파라미터가 추가됩니다. `granularity=Hourly` 및 `Filters=filter_list`를 예로 들 수 있습니다.
2. 브라우저를 사용하여 콘솔 URL을 북마크로 추가합니다.
3. 이렇게 만든 북마크를 사용하여 이후에 동일한 보고서를 생성할 수 있습니다.

사용 데이터 내보내기

데이터를 내보내서 다른 보고서에 보고서 그래프 또는 테이블을 포함할 수 있습니다.

사용 현황 데이터를 내보내려면

1. 보고서에 대한 옵션과 필터를 선택합니다.
2. 테이블의 사용 현황 데이터를 `.csv` 파일로 내보내려면 [Download]를 선택하고 [CSV Only]를 선택합니다.
3. 테이블의 그래픽 사용 현황 데이터를 `.png` 파일로 내보내려면 [Download]를 선택하고 [Graph Only]를 선택합니다.

옵션 참조 설명

[Show] 옵션을 사용해서 보고서 그래프에서 측정치가 표시되도록 지정합니다.

- 평균 사용률

선택한 시간 범위 동안 매 시간에 대한 평균 사용률을 표시합니다. 여기서 한 시간에 대한 버킷의 사용률은 그 시간 동안 사용된 인스턴스 시간의 수를 그 시간 동안 소유했던 예약 인스턴스의 총 수로 나눈 값입니다.

- 최대 사용률

선택한 시간 범위 동안 어떤 시간에 있어 가장 높은 사용률을 표시합니다. 여기서 한 시간에 대한 버킷의 사용률은 그 시간 동안 사용된 인스턴스 시간의 수를 그 시간 동안 소유했던 예약 인스턴스의 총수로 나눈 값입니다.

- 총 비용

보고서가 다루는 기간 동안에 대해, 사용 비용에 버킷의 예약 인스턴스의 선불 비용에 대한 분할 상환 부분을 더한 비용을 표시합니다.

- 사용 비용

예약 인스턴스의 선택한 버킷에 대해 시간별 요금에 따른 총 비용을 표시합니다.

[Time range]를 사용해서 보고서가 다루게 되는 기간을 지정합니다.

Note

모든 시간은 협정 세계시(UTC)로 지정됩니다.

- Last 7 Days(지난 7일)

현재 및 지난 6일 역일 동안 발생한 사용 데이터를 표시합니다. 일별, 월별 세부 수준과 함께 사용할 수 있습니다.

- Last 14 Days(지난 14일)

현재 및 지난 13일 역일 동안 발생한 사용 데이터를 표시합니다. 일별, 월별 세부 수준과 함께 사용할 수 있습니다.

- This Month

현재 역월 동안 발생한 사용 데이터를 표시합니다. 일별, 월별 세부 수준과 함께 사용할 수 있습니다.

- Last 3 Months(지난 12월)

현재 및 지난 2개월 역월 동안 발생한 사용 데이터를 표시합니다. 일별, 월별 세부 수준과 함께 사용할 수 있습니다.

- Last 6 Months(지난 6월)

현재 및 지난 5개월 역월 동안 발생한 사용 데이터를 표시합니다. 월별 세부 수준과 함께 사용할 수 있습니다.

- Last 12 Months(지난 12월)

현재 및 지난 11개월 역월 동안 발생한 사용 데이터를 표시합니다. 월별 세부 수준과 함께 사용할 수 있습니다.

- This Year

현재 역년 동안 발생한 사용 데이터를 표시합니다. 월별 세부 수준과 함께 사용할 수 있습니다.

- Last 3 Years(지난 3년)

현재 및 지난 2년 역년 동안 발생한 사용 데이터를 표시합니다. 월별 세부 수준과 함께 사용할 수 있습니다.

- Custom

mm/dd/yyyy 형식으로 지정하는 입력된 Start 및 End 날짜 시간 범위에 대한 데이터를 표시합니다. 시간별, 일별, 월별 세부 수준과 함께 사용할 수 있지만, 시간별 데이터는 2일, 일별 데이터는 2월, 월별 데이터는 3년의 최대 시간 범위를 지정해야 합니다.

[Filter]를 사용해서 보고서에 표시되는 데이터의 범위를 지정합니다.

- Regions(리전)
- 인스턴스 유형
- Accounts(계정)
- Platforms(플랫폼)
- Tenancy(테넌시)
- Offering Type(제공 유형)

인스턴스 문제 해결

다음 문서는 인스턴스 관련 문제를 해결하는 데 도움이 됩니다.

목차

- [인스턴스가 즉시 종료되는 경우 해결 방법 \(p. 698\)](#)
- [인스턴스 연결 문제 해결 \(p. 699\)](#)
- [인스턴스 중지 문제 해결 \(p. 705\)](#)
- [인스턴스 종료 문제 해결 \(p. 706\)](#)
- [인스턴스 복구 실패 문제 해결 \(p. 707\)](#)
- [상태 확인에 실패한 인스턴스 문제 해결 \(p. 707\)](#)
- [인스턴스 용량 문제 해결 \(p. 728\)](#)
- [콘솔 출력 가져오기 및 인스턴스 재부팅 \(p. 729\)](#)
- [인스턴스가 잘못된 볼륨에서 부팅될 경우 \(p. 731\)](#)

Windows 인스턴스와 관련하여 추가적인 도움이 필요하면 Windows 인스턴스용 Amazon EC2 사용 설명서의 [Windows 인스턴스 문제 해결](#)을 참조하십시오.

[Amazon EC2 forum](#)에서 답을 검색하고 질문을 올릴 수도 있습니다.

인스턴스가 즉시 종료되는 경우 해결 방법

인스턴스가 시작된 후에는 인스턴스의 상태를 확인하여 pending 상태가 running 상태(terminated 상태 아님)로 전환되는지 확인하는 것이 좋습니다.

인스턴스가 즉시 종료되는 이유에는 다음과 같이 몇 가지가 있습니다.

- EBS 볼륨 제한에 도달했습니다. 볼륨 제한에 대한 정보를 보거나 볼륨 제한 증가 요청을 제출하려면 [Amazon EBS 볼륨 제한 증가 요청](#)을 참조하십시오.
- EBS 스냅샷이 손상되었습니다.
- 인스턴스를 시작하는 데 사용한 인스턴스 스토어 지원 AMI에 필수 부분(image.part.xx 파일)이 누락되었습니다.

인스턴스 종료 이유 파악

Amazon EC2 콘솔, CLI 또는 API를 사용하여 인스턴스가 종료된 이유를 확인할 수 있습니다.

콘솔을 사용하여 인스턴스 종료 이유 확인하기

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Instances]를 선택하고 인스턴스를 선택합니다.
3. [Description] 탭의 [State transition reason] 레이블 옆에서 이유를 찾습니다. 인스턴스가 실행 중일 경우에는 일반적으로 이유가 나열되지 않습니다. 인스턴스를 명시적으로 중지하거나 종료한 경우에는 이유가 `User initiated shutdown`입니다.

명령줄을 사용하여 인스턴스가 종료된 이유를 파악하려면 다음을 수행합니다.

1. 아래와 같이 `describe-instances` 명령을 사용합니다.

```
aws ec2 describe-instances --instance-id instance_id
```

2. 표시되는 JSON 이유에서 `StateReason` 요소를 찾습니다. 예를 들면 다음과 같습니다.

```
"StateReason": {  
    "Message": "Client.UserInitiatedShutdown: User initiated shutdown",  
    "Code": "Client.UserInitiatedShutdown"  
},
```

이 예에서는 실행 중인 인스턴스를 중지하거나 종료한 후 표시될 이유 코드를 보여 줍니다. 인스턴스를 즉시 종료한 경우 인스턴스의 종료 이유를 설명하는 `code` 및 `message` 요소가 표시됩니다(예: `VolumeLimitExceeded`)。

인스턴스 연결 문제 해결

다음은 인스턴스에 연결하는 중에 발생할 수 있는 문제와 오류 메시지입니다.

목차

- [인스턴스 연결 중 오류 발생: 연결 시간 초과](#) (p. 699)
- [오류r: 서버에서 사용자 키를 인식하지 못함](#) (p. 701)
- [오류: 호스트 키를 찾을 수 없음. 권한 거부\(퍼블릭 키\) 또는 인증 실패, 권한 거부](#) (p. 702)
- [오류: 보호되지 않는 프라이빗 키 파일](#) (p. 703)
- [오류: 서버에서 키 거부 또는 지원되는 인증 방법이 없음](#) (p. 704)
- [Safari 브라우저에서 MindTerm 사용 중 오류 발생](#) (p. 704)
- [Mac OS X RDP 클라이언트 사용 중 오류 발생](#) (p. 705)
- [인스턴스를 ping할 수 없음](#) (p. 705)

Windows 인스턴스와 관련하여 추가적인 도움이 필요하면 Windows 인스턴스용 Amazon EC2 사용 설명서의 [Windows 인스턴스 문제 해결](#)을 참조하십시오.

[Amazon EC2 forum](#)에서 답을 검색하고 질문을 올릴 수도 있습니다.

인스턴스 연결 중 오류 발생: 연결 시간 초과

인스턴스에 연결하려 할 때 `Network error: Connection timed out` 또는 `Error connecting to [instance], reason: -> Connection timed out: connect`라는 오류 메시지가 표시되면 다음과 같이 하십시오.

- 보안 그룹 규칙을 확인합니다. 퍼블릭 IPv4 주소의 인바운드 트래픽을 적절한 포트로 허용하는 보안 그룹 규칙이 필요합니다.
 1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
 2. 탐색 창에서 [Instances]를 선택한 다음 인스턴스를 선택합니다.
 3. [Description] 탭에서 [Security groups] 옆에 있는 [view rules]를 선택하여 적용되는 규칙 목록을 표시합니다.
 4. Linux 인스턴스: 해당 컴퓨터에서 포트 22(SSH)로의 트래픽을 허용하는 규칙이 있는지 확인합니다.

Windows 인스턴스: 해당 컴퓨터에서 포트 3389(RDP)로의 트래픽을 허용하는 규칙이 있는지 확인합니다.

보안 그룹에 단일 IP 주소의 인바운드 트래픽을 허용하는 규칙이 있으며, 컴퓨터가 회사 네트워크에 있거나 ISP(인터넷 서비스 공급자)를 통해 연결하는 경우, 이 주소는 고정되지 않을 수 있습니다. 대신 클라이언트 컴퓨터에서 사용하는 IP 주소 범위를 지정합니다. 보안 그룹에 이전 단계에서 설명한 인바운드 트래픽을 허용하는 규칙이 없을 경우 보안 그룹에 추가합니다. 자세한 내용은 [인스턴스에 네트워크 액세스 권한 부여 \(p. 464\)](#) 섹션을 참조하십시오.

- [EC2-VPC] 서브넷의 라우팅 테이블을 확인합니다. VPC 외부로 지정된 모든 트래픽을 VPC의 인터넷 게이트웨이로 보내는 경로가 필요합니다.
 1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
 2. 탐색 창에서 [Instances]를 선택한 다음 인스턴스를 선택합니다.
 3. [Description] 탭에서 [VPC ID]와 [Subnet ID]의 값을 기록합니다.
 4. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
 5. 탐색 창에서 [Internet Gateways]를 선택합니다. VPC에 인터넷 게이트웨이가 연결되어 있는지 확인합니다. 또는 [Create Internet Gateway]를 선택하여 인터넷 게이트웨이를 만듭니다. 인터넷 게이트웨이를 선택한 후 [Attach to VPC]를 선택하고 지침에 따라 VPC에 연결합니다.
 6. 탐색 창에서 [Subnets]를 선택한 후 해당 서브넷을 선택합니다.
 7. [Route Table] 탭에서 대상 위치로 0.0.0.0/0 경로가 있으며, VPC의 대상으로 해당 인터넷 게이트웨이가 있는지 확인합니다. 그렇지 않을 경우 라우팅 테이블의 ID(rtb-xxxxxx)를 선택하여 라우팅 테이블의 [Routes] 탭으로 이동한 후 [Edit], [Add another route]를 차례로 선택하고 [Destination]에 0.0.0.0/0을 입력한 후, [Target]에서 인터넷 게이트웨이를 선택하고 [Save]를 선택합니다.

IPv6 주소를 이용해 인스턴스에 연결하는 경우 인터넷 게이트웨이를 가리키는 모든 IPv6 트래픽(:/:0)에 대한 경로가 있는지 확인합니다. 그렇지 않으면 대상 위치로 ::/0 경로를, 대상으로 인터넷 게이트웨이를 추가합니다.

- [EC2-VPC] 네트워크 ACL(액세스 제어 목록)을 검사하여 서브넷 유무를 확인하십시오. 네트워크 ACL은 로컬 IP 주소에서 적절한 포트를 통해 수신되는 인바운드와 아웃바운드 트래픽을 모두 허용해야 합니다. 기본 네트워크 ACL은 인바운드와 아웃바운드 트래픽을 모두 허용합니다.
 1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
 2. 탐색 창에서 Your VPCs를 선택합니다.
 3. [Summary] 탭에서 [Network ACL]을 찾아 ID(acl-xxxxxx)를 선택하고 ACL을 선택합니다.
 4. [Inbound Rules] 탭에서 규칙이 해당 컴퓨터로부터의 트래픽을 허용하는지 확인합니다. 허용하지 않을 경우 해당 컴퓨터로부터의 트래픽을 차단하는 규칙을 삭제하거나 수정합니다.
 5. [Outbound Rules] 탭에서 규칙이 해당 컴퓨터로의 트래픽을 허용하는지 확인합니다. 허용하지 않을 경우 해당 컴퓨터로의 트래픽을 차단하는 규칙을 삭제하거나 수정합니다.- 컴퓨터가 회사 네트워크에 있을 경우 네트워크 관리자에게 내부 방화벽이 해당 컴퓨터의 포트 22(Linux 인스턴스) 또는 포트 3389(Windows 인스턴스)로부터의 트래픽을 허용하는지 여부를 문의합니다.
- 인스턴스에 퍼블릭 IPv4 주소가 있는지 확인합니다. 퍼블릭 IP 주소가 없을 경우 인스턴스와 탄력적 IP 주소를 연결할 수 있습니다. 자세한 내용은 [탄력적 IP 주소 \(p. 505\)](#)를 참조하십시오.

- 인스턴스에서 CPU 부하를 확인합니다. 서버 과부하가 발생했을 수 있습니다. AWS는 Amazon CloudWatch 메트릭 및 인스턴스 상태 등과 같은 데이터를 자동으로 제공하므로, 이러한 정보를 사용하여 인스턴스에 대한 CPU 부하를 확인하고 필요할 경우 부하 처리 방법을 조정할 수 있습니다. 자세한 내용은 [CloudWatch를 사용해 인스턴스 모니터링하기 \(p. 347\)](#)을 참조하십시오.
- 부하가 가볍적이면 [Auto Scaling](#) 및 [Elastic Load Balancing](#)을 사용하여 인스턴스를 자동으로 확장하거나 축소할 수 있습니다.
- 부하가 꾸준히 증가하는 경우 더 큰 인스턴스 유형으로 전환할 수 있습니다. 자세한 내용은 [인스턴스 크기 조정 \(p. 169\)](#)을 참조하십시오.

IPv6 주소를 사용해 인스턴스에 연결하려면 다음을 확인합니다.

- 서브넷은 IPv6 트래픽(:/:0)을 인터넷 게이트웨이로 이어주는 경로가 있는 라우팅 테이블과 연결되어야 합니다.
- 보안 그룹 규칙은 로컬 IPv6 주소의 인바운드 트래픽을 적절한 포트(Linux의 경우 22, Windows의 경우 3389)로 허용해야 합니다.
- 네트워크 ACL 규칙은 인바운드 및 아웃바운드 IPv6 트래픽을 허용해야 합니다.
- 이전 AMI에서 인스턴스를 시작한 경우, DHCPv6에 맞게 구성되지 않을 수 있습니다(IPv6 주소가 네트워크 인터페이스에서 자동 인식되지 않습니다). 자세한 내용은 Amazon VPC 사용 설명서의 [인스턴스에서 IPv6 구성하기](#)를 참조하십시오.
- 로컬 컴퓨터에 IPv6 주소가 있고 IPv6를 사용하도록 컴퓨터를 구성해야 합니다.

오류r: 서버에서 사용자 키를 인식하지 못함

SSH를 사용하여 인스턴스에 연결하는 경우

- 연결 시 ssh -vvv를 사용하여 자세한 디버깅 정보를 확인합니다.

```
ssh -vvv -i [your key name].pem ec2-user@[public DNS address of your instance].compute-1.amazonaws.com
```

다음 샘플 출력은 서버에서 인식하지 못하는 키를 사용하여 인스턴스에 연결하려 했는지를 확인하는 방법을 보여 줍니다.

```
open/ANT/myusername/.ssh/known_hosts).
debug2: bits set: 504/1024
debug1: ssh_rsa_verify: signature correct
debug2: kex_derive_keys
debug2: set_newkeys: mode 1
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug2: set_newkeys: mode 0
debug1: SSH2_MSG_NEWKEYS received
debug1: Roaming not allowed by server
debug1: SSH2_MSG_SERVICE_REQUEST sent
debug2: service_accept: ssh-userauth
debug1: SSH2_MSG_SERVICE_ACCEPT received
debug2: key: bogus.pem ((nil))
debug1: Authentications that can continue: publickey
debug3: start over, passed a different list publickey
debug3: preferred gssapi-keyex,gssapi-with-mic,publickey,keyboard-interactive,password
debug3: authmethod_lookup publickey
debug3: remaining preferred: keyboard-interactive,password
debug3: authmethod_is_enabled publickey
debug1: Next authentication method: publickey
debug1: Trying private key: bogus.pem
debug1: read PEM private key done: type RSA
```

Amazon Elastic Compute Cloud
Linux 인스턴스용 사용 설명서
오류: 호스트 키를 찾을 수 없음. 권한 거부(퍼블릭 키) 또는 인증 실패, 권한 거부

```
debug3: sign_and_send_pubkey: RSA 9c:4c:bc:0c:d0:5c:c7:92:6c:8e:9b:16:e4:43:d8:b2
debug2: we sent a publickey packet, wait for reply
debug1: Authentications that can continue: publickey
debug2: we did not send a packet, disable method
debug1: No more authentication methods to try.
Permission denied (publickey).
```

SSH(MindTerm)를 사용하여 인스턴스에 연결하는 경우

- Java를 사용하도록 설정하지 않으면 서버에서 사용자 키를 인식하지 못합니다. Java를 사용하도록 설정하면 Java 설명서에서 웹 브라우저에서 Java를 사용으로 설정하는 방법은 무엇입니까? 섹션을 참조하십시오.

PuTTY를 사용하여 인스턴스에 연결하는 경우

- 프라이빗 키(.pem) 파일이 PuTTY(.ppk)에서 인식하는 형식으로 변환되었는지 확인합니다. 프라이빗 키 변환에 대한 자세한 내용은 PuTTY를 사용하여 Windows에서 Linux 인스턴스에 연결 (p. 278) 섹션을 참조하십시오.

Note

PuTTYgen에서 프라이빗 키 파일을 불러온 후 Generate가 아니라 Save Private Key를 선택합니다.

- AMI에 적합한 사용자 이름을 사용하여 연결하고 있는지 확인합니다. PuTTY Configuration 창에서 Host name 상자에 사용자 이름을 입력합니다.
 - Amazon Linux AMI의 경우 사용자 이름은 `ec2-user`입니다.
 - RHEL AMI의 경우 사용자 이름은 `ec2-user` 또는 `root`입니다.
 - Ubuntu AMI의 경우 사용자 이름은 `ubuntu` 또는 `root`입니다.
 - Centos AMI의 경우 사용자 이름은 `centos`입니다.
 - Fedora AMI의 경우 사용자 이름은 `ec2-user`입니다.
 - SUSE의 경우 사용자 이름은 `ec2-user` 또는 `root`입니다.
 - `ec2-user` 및 `root`을 사용할 수 없는 경우 AMI 공급자에게 문의하십시오.
- 해당 포트로의 인바운드 트래픽을 허용할 인바운드 보안 그룹 규칙이 있는지 확인합니다. 자세한 내용은 인스턴스에 네트워크 액세스 권한 부여 (p. 464) 섹션을 참조하십시오.

오류: 호스트 키를 찾을 수 없음. 권한 거부(퍼블릭 키) 또는 인증 실패, 권한 거부

SSH를 사용하여 인스턴스에 연결할 때 Host key not found in [directory], Permission denied (publickey) 또는 Authentication failed, permission denied 오류 중 하나가 발생하는 경우 AMI에 적합한 사용자 이름을 사용하여 연결하고 있으며 인스턴스에 대한 올바른 프라이빗 키(.pem) 파일을 지정했는지 확인합니다. MindTerm 클라이언트의 경우 Connect To Your Instance 창에서 User name 상자에 사용자 이름을 입력합니다.

적합한 사용자 이름은 다음과 같습니다.

- Amazon Linux AMI의 경우 사용자 이름은 `ec2-user`입니다.
- RHEL AMI의 경우 사용자 이름은 `ec2-user` 또는 `root`입니다.
- Ubuntu AMI의 경우 사용자 이름은 `ubuntu` 또는 `root`입니다.
- Centos AMI의 경우 사용자 이름은 `centos`입니다.
- Fedora AMI의 경우 사용자 이름은 `ec2-user`입니다.

- SUSE의 경우 사용자 이름은 `ec2-user` 또는 `root`입니다.
 - `ec2-user` 및 `root`을 사용할 수 없는 경우 AMI 풀급자에게 문의하십시오.

인스턴스를 시작할 때 선택한 키 페어에 해당하는 프라이빗 키를 사용하고 있는지 확인합니다.

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
 2. 인스턴스를 선택합니다. [Description] 탭에서 [Key pair name]의 값을 확인합니다.
 3. 인스턴스를 시작할 때 키 페어를 지정하지 않은 경우 인스턴스를 종료하고 새 인스턴스를 시작하여 키 페어를 지정할 수 있습니다. 이 인스턴스가 사용하던 인스턴스이지만 해당 키 페어의 .pem 파일이 없을 경우 키 페어를 새 것으로 바꿀 수 있습니다. 자세한 내용은 [프라이빗 키를 분실했을 때 Linux 인스턴스에 연결하는 방법](#) (p. 382) 섹션을 참조하십시오.

고유한 키 페어를 만든 경우 키 생성기가 RSA 키를 만들도록 설정되어 있는지 확인합니다. DSA 키는 허용되지 않습니다.

Permission denied (publickey) 오류가 반환되고 위의 어느 것도 적용되지 않는 경우(예를 들어, 전에는 연결할 수 있었지만), 인스턴스의 험 디렉토리에서의 권한이 변경되었을 수 있습니다. /home/ec2-user/.ssh/authorized_keys에 대한 권한은 소유자로만 제한되어야 합니다.

인스턴스에서 권한을 확인하려면

1. 인스턴스를 중지하고 루트 볼륨을 분리합니다. 자세한 내용은 [인스턴스 중지 및 시작 \(p. 285\)](#) 및 [인스턴스에서 Amazon EBS 볼륨 분리 \(p. 588\)](#) 섹션을 참조하십시오.
 2. 현재의 인스턴스와 동일한 가용 영역에서 임시 인스턴스를 시작하고(현재의 인스턴스에 사용한 것과 비슷하거나 동일한 AMI 사용) 루트 볼륨을 임시 인스턴스에 연결합니다. 자세한 내용은 [Amazon EBS 볼륨을 인스턴스에 연결 \(p. 576\)](#) 섹션을 참조하십시오.
 3. 임시 인스턴스에 연결하고 마운트 지점을 생성한 후 연결한 볼륨을 마운트합니다. 자세한 내용은 [Amazon EBS 볼륨을 사용할 수 있도록 만들기 \(p. 577\)](#) 섹션을 참조하십시오.
 4. 임시 인스턴스에서 연결된 볼륨의 /home/ec2-user/ 디렉토리의 권한을 확인합니다. 필요하다면 다음과 같이 권한을 조정합니다.

```
chmod 600 mount point/home/ec2-user/.ssh/authorized_keys
```

```
chmod 700 mount point/home/ec2-user/.ssh
```

```
chmod 700 mount point/home/ec2-user
```

- 볼륨을 마운트 해제하고 임시 인스턴스에서 분리한 다음 원본 인스턴스에 다시 연결합니다. 루트 볼륨에 올바른 이름을 지정해야 합니다(예: /dev/xvda).
 - 인스턴스를 시작합니다. 더 이상 필요하지 않은 경우, 임시 인스턴스를 종료할 수 있습니다.

오류: 보호되지 않는 프라이빗 키 파일

다른 사용자의 읽기 및 쓰기 작업으로부터 프라이빗 키 파일을 보호해야 합니다. 프라이빗 키를 본인 이외의 다른 사람이 읽거나 쓸 수 있는 경우 SSH는 키를 무시하고 다음과 같은 경고 메시지를 표시합니다.

```
@@@@@WARNING: UNPROTECTED PRIVATE KEY FILE!@@@@@  
Permissions 0777 for '.ssh/my_private_key.pem' are too open.  
It is required that your private key files are NOT accessible by others.  
This private key will be ignored.
```

```
bad permissions: ignore key: .ssh/my_private_key.pem  
Permission denied (publickey).
```

인스턴스에 로그인할 때 이와 비슷한 메시지가 표시되면 오류 메시지의 첫 줄을 살펴보고 인스턴스에 올바른 퍼블릭 키를 사용하고 있는지 확인합니다. 위의 예에서는 프라이빗 키 `.ssh/my_private_key.pem`와 파일 권한 `0777`이 사용되어 모든 사람에게 이 파일에 대한 읽기 또는 쓰기가 허용됩니다. 이 권한 수준은 전혀 보호되지 않는 수준이므로 SSH에서는 이 키를 무시합니다. 오류를 해결하려면 프라이빗 키 파일 경로를 대체하는 다음 명령을 실행합니다.

```
chmod 0400 .ssh/my_private_key.pem
```

오류: 서버에서 키 거부 또는 지원되는 인증 방법이 없음

PuTTY를 사용하여 인스턴스에 연결하려 할 때 `Error: Server refused our key` 또는 `Error: No supported authentication methods available` 오류 중 하나가 발생하면 AMI에 적합한 사용자 이름을 사용하여 연결하고 있는지 확인합니다. PuTTY Configuration 창에서 User name 상자에 사용자 이름을 입력합니다.

적합한 사용자 이름은 다음과 같습니다.

- Amazon Linux AMI의 경우 사용자 이름은 `ec2-user`입니다.
- RHEL AMI의 경우 사용자 이름은 `ec2-user` 또는 `root`입니다.
- Ubuntu AMI의 경우 사용자 이름은 `ubuntu` 또는 `root`입니다.
- Centos AMI의 경우 사용자 이름은 `centos`입니다.
- Fedora AMI의 경우 사용자 이름은 `ec2-user`입니다.
- SUSE의 경우 사용자 이름은 `ec2-user` 또는 `root`입니다.
- `ec2-user` 및 `root`을 사용할 수 없는 경우 AMI 공급자에게 문의하십시오.

또한 프라이빗 키(`.pem`) 파일을 PuTTY(`.ppk`)에서 인식되는 형식으로 올바르게 변환했는지도 확인해야 합니다. 프라이빗 키 변환에 대한 자세한 내용은 [PuTTY를 사용하여 Windows에서 Linux 인스턴스에 연결 \(p. 278\)](#) 섹션을 참조하십시오.

Safari 브라우저에서 MindTerm 사용 중 오류 발생

Safari 웹 브라우저를 사용 중일 때 MindTerm을 사용하여 인스턴스에 연결하려는 경우 다음 오류가 발생할 수 있습니다.

```
Error connecting to your_instance_ip, reason:  
-> Key exchange failed: Host authentication failed
```

브라우저의 보안 설정을 업데이트하여 AWS Management Console에서 비안전 모드에서 Java 플러그인을 실행하는 것을 허용해야 합니다.

Java 플러그인을 비안전 모드에서 실행할 수 있도록 설정하기

1. Safari에서 Amazon EC2 콘솔을 열어 놓은 채 [Safari], [기본 설정], [보안]을 차례로 선택합니다.
2. [Plug-in Settings]를 선택합니다(Safari가 구 버전일 경우 [Manage Website Settings]).
3. 왼쪽에서 [Java] 플러그인을 선택한 후 [현재 열려 있는 웹 사이트] 목록에서 AWS Management Console URL을 찾습니다. 관련 목록에서 비안전 모드에서 실행을 선택합니다.
4. 메시지가 표시되면 경고 대화 상자에서 [신뢰]를 선택합니다. [완료]를 선택하여 브라우저로 돌아갑니다.

Mac OS X RDP 클라이언트 사용 중 오류 발생

Microsoft 웹 사이트의 원격 데스크톱 연결을 사용하여 Windows Server 2012 R2 인스턴스에 연결하려는 경우 다음 오류가 발생할 수 있습니다.

Remote Desktop Connection cannot verify the identity of the computer that you want to connect to.

Apple iTunes 스토어에서 Microsoft Remote Desktop 앱을 다운로드한 후 이 앱을 사용하여 인스턴스에 연결합니다.

인스턴스를 ping할 수 없음

ping 명령은 일종의 ICMP 트래픽입니다. 따라서 인스턴스를 ping할 수 없는 경우, 인바운드 보안 그룹 규칙에서 모든 소스, 즉 컴퓨터 또는 명령을 실행하는 인스턴스에서 오는 Echo Request 메시지에 대한 ICMP 트래픽을 허용하는지 확인합니다. 인스턴스에서 ping 명령을 실행할 수 없는 경우, 아웃바운드 보안 그룹 규칙에서 모든 대상, 즉 ping을 시도 중인 호스트에 보내는 Echo Request 메시지에 대한 ICMP 트래픽을 허용하는지 확인합니다.

인스턴스 중지 문제 해결

Amazon EBS 인스턴스를 중지한 후 이 인스턴스가 `stopping` 상태로 멈춰 있는 것 같아 보일 경우 기본 호스트 컴퓨터에 문제가 있을 수 있습니다.

우선, 인스턴스를 다시 중지해 봅니다. `stop-instances`(AWS CLI) 명령을 사용할 경우 `--force` 옵션을 사용해야 합니다.

인스턴스를 강제로 중지할 수 없으면 인스턴스에서 AMI를 만들어서 대체 인스턴스를 시작할 수 있습니다.

인스턴스가 `running` 상태에 있지 않은 인스턴스 시간에 대해서는 요금이 부과되지 않습니다.

대체 인스턴스 만들기

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Instances]를 선택하고 인스턴스를 선택합니다.
3. [Actions], [Image], [Create Image]를 차례로 선택합니다.
4. [Create Image] 대화 상자에서 다음 필드를 채운 다음 [Create Image]를 선택합니다.
 - a. AMI에 대한 이름 및 설명을 지정합니다.
 - b. [No reboot]를 선택합니다.
5. AMI에서 인스턴스를 시작하고 인스턴스가 작동하는지 확인합니다.
6. 멈춰 있는 인스턴스를 선택하고 [Actions], [Instance State], [Terminate]를 차례로 선택합니다. 또한 인스턴스가 종료 중 상태로 멈추는 경우 Amazon EC2에서 몇 시간 내에 해당 인스턴스를 자동으로 종료합니다.

이전 절차에 설명된 대로 AMI를 만들 수 없으면 다음과 같이 대체 인스턴스를 설정할 수 있습니다.

대체 인스턴스 만들기(앞의 절차가 실패할 경우)

1. 인스턴스를 선택하고 Description 탭을 연 다음 Block devices 목록을 확인합니다. 각 볼륨을 선택하고 볼륨 ID를 적습니다. 어느 볼륨이 루트 볼륨인지 적어두어야 합니다.

2. 탐색 창에서 [Volumes]를 선택합니다. 인스턴스에 해당하는 각 볼륨을 선택하고 [Actions], [Create Snapshot]을 차례로 선택합니다.
3. 탐색 창에서 [Snapshots]를 선택합니다. 방금 만든 스냅샷을 선택한 후 [Actions], [Create Volume]을 선택합니다.
4. 멈춰 있는 인스턴스와 동일한 유형의 인스턴스를 시작합니다(Amazon Linux, Windows 등). 루트 볼륨의 볼륨 ID와 디바이스 이름을 적어둡니다.
5. 탐색 창에서 [Instances]를 선택하고 방금 시작한 인스턴스를 선택한 후, [Actions], [Instance State]를 차례로 선택하고 [Stop]을 선택합니다.
6. 탐색 창에서 [Volumes]를 선택하고 중지된 인스턴스의 루트 볼륨을 선택한 후, [Actions], [Detach Volume]을 선택합니다.
7. 멈춰 있는 인스턴스에서 만든 루트 볼륨을 선택하고 [Actions], [Attach Volume]을 선택한 후, 이 볼륨을 새 인스턴스에 루트 볼륨으로 연결합니다(기록해 놓은 디바이스 이름 사용). 루트 이외의 다른 추가 볼륨을 인스턴스에 연결합니다.
8. 탐색 창에서 [Instances]를 선택하고 대체 인스턴스를 선택합니다. [Actions], [Instance State], [Start]를 차례로 선택합니다. 인스턴스가 작동 중인지 확인합니다.
9. 멈춰 있는 인스턴스를 선택하고 [Actions], [Instance State], [Terminate]를 차례로 선택합니다. 또한 인스턴스가 종료 중 상태로 멈춰 있는 경우 Amazon EC2에서 몇 시간 내에 해당 인스턴스를 자동으로 종료합니다.

이러한 절차를 완료할 수 없으면 [Amazon EC2 forum](#)에서 도움을 요청하는 글을 게시할 수 있습니다. 해결 방법을 신속히 찾아내려면 인스턴스 ID를 포함하고 자신이 이미 수행했던 단계에 대해 설명하십시오.

인스턴스 종료 문제 해결

인스턴스가 `running` 상태에 있지 않은 인스턴스 시간에 대해서는 요금이 부과되지 않습니다. 다시 말해서, 인스턴스를 종료할 때 인스턴스의 상태가 `shutting-down`으로 변경되는 즉시 해당 인스턴스에 대한 요금 발생이 중지되는 것입니다.

지연된 인스턴스 종료

인스턴스가 몇 분 이상 `shutting-down` 상태로 유지되는 경우 인스턴스에 의해 실행 중인 종료 스크립트로 인한 지연이 발생했을 수 있습니다.

또 한 가지 예상 원인은 기본 호스트 컴퓨터 관련 문제입니다. 인스턴스가 몇 시간 동안 `shutting-down` 상태로 유지되는 경우 Amazon EC2는 해당 인스턴스를 멈춰 있는 인스턴스로 간주하여 강제로 종료합니다.

인스턴스가 종료 중 상태로 멈춰 있는 것처럼 보이며 이 상태로 몇 시간 이상이 경과된 경우 [Amazon EC2 forum](#)에 도움을 요청하는 글을 게시하십시오. 해결 방법을 신속히 찾아내려면 인스턴스 ID를 포함하고 자신이 이미 수행했던 단계에 대해 설명하십시오.

종료된 인스턴스가 계속 표시됨

인스턴스를 종료한 후에도 인스턴스는 잠깐 동안 콘솔에서 표시된 후 삭제됩니다. 상태는 `terminated`로 표시됩니다. 몇 시간이 지난 후에도 해당 항목이 삭제되지 않으면 Support에 문의하십시오.

인스턴스 자동 시작 또는 종료

모든 인스턴스를 종료하는 경우 사용자를 대신하여 새 인스턴스가 시작됩니다. 인스턴스 하나를 종료하는 경우 인스턴스 중 하나가 종료됩니다. 인스턴스를 종료하면, AWS가 인스턴스를 종료하고 새 인스턴스를 시작

하는 것을 볼 수 있습니다. 일반적으로 이러한 동작은 사용자가 Auto Scaling 또는 Elastic Beanstalk를 사용하여 정의한 기준에 따라 컴퓨팅 리소스의 규모를 자동으로 확장/축소해 왔음을 의미합니다.

자세한 내용은 [Auto Scaling 사용 설명서](#) 또는 [AWS Elastic Beanstalk 개발자 안내서](#) 섹션을 참조하십시오.

인스턴스 복구 실패 문제 해결

다음 문제로 인해 인스턴스의 자동 복구가 실패할 수 있습니다.

- 대체 하드웨어의 일시적인 용량 부족
- 인스턴스에 인스턴스 스토어 스토리지가 연결되었으나, 자동 인스턴스 복구용으로 지원되지 않는 구성입니다.
- 진행 중인 서비스 상태 대시보드 이벤트가 있어서 복구 프로세스가 성공적으로 실행되지 못했습니다. 최신 서비스 가용성 정보는 <http://status.aws.amazon.com>을 참조하십시오.
- 인스턴스 복구 시도가 하루 최대 허용 횟수인 3회에 도달했습니다.

자동 복구 프로세스는 매일 최대 3회의 개별 실패에 대해서만 인스턴스 복구를 시도합니다. 인스턴스 시스템 상태 확인 실패가 계속되는 경우 인스턴스를 수동으로 시작 및 중지하는 것이 좋습니다. 자세한 내용은 [인스턴스 중지 및 시작 \(p. 285\)](#)을 참조하십시오.

자동 복구가 실패하고 원래 시스템 상태 확인 실패의 근본 원인이 하드웨어 성능 저하로 확인되는 경우 이후에 인스턴스가 사용 중지될 수 있습니다.

상태 확인에 실패한 인스턴스 문제 해결

항목

- 초기 단계 (p. 708)
- 시스템 로그 검색 (p. 708)
- Linux 기반 인스턴스의 시스템 로그 오류 문제 해결 (p. 709)
- 메모리 부족: 프로세스 종지 (p. 710)
- ERROR: mmu_update failed(메모리 관리 업데이트 실패) (p. 710)
- I/O 오류(블록 디바이스 오류) (p. 711)
- IO ERROR: neither local nor remote disk(분산된 블록 디바이스 손상) (p. 712)
- request_module: runaway loop modprobe(이전 Linux 버전에서 레거시 커널 modprobe 반복) (p. 713)
- "FATAL: kernel too old" 및 "fsck: No such file or directory while trying to open /dev"(커널과 AMI 불일치) (p. 713)
- "FATAL: Could not load /lib/modules" 또는 "BusyBox"(커널 모듈 누락) (p. 714)
- ERROR Invalid kernel(EC2 커널이 호환되지 않음) (p. 715)
- request_module: runaway loop modprobe(이전 Linux 버전에서 레거시 커널 modprobe 반복) (p. 716)
- fsck: No such file or directory while trying to open...(파일 시스템을 찾을 수 없음) (p. 717)
- 파일 시스템 마운트 관련 일반 오류(마운트 실패) (p. 718)
- VFS: Unable to mount root fs on unknown-block(루트 파일 시스템 불일치) (p. 720)
- Error: Unable to determine major/minor number of root device...(루트 파일 시스템/디바이스 불일치) (p. 721)
- XENBUS: Device with no driver... (p. 722)

- ... days without being checked, check forced(파일 시스템 검사 필요) (p. 723)
- fsck died with exit status...(디바이스 누락) (p. 723)
- GRUB 프롬프트(grubdom>) (p. 724)
- Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring(eth0 인터페이스를 가져오는 중: eth0 디바이스의 MAC 주소가 틀려서 무시합니다). (하드 코딩된 MAC 주소) (p. 726)
- Unable to load SELinux Policy. Machine is in enforcing mode. Halting now(SELinux 정책을 가져올 수 없습니다. 시스템이 강제 실행 모드입니다. 중단됩니다). (잘못된 SELinux 구성) (p. 727)
- XENBUS: Timeout connecting to devices(Xenbus 시간 초과) (p. 728)

초기 단계

인스턴스에 대한 상태 확인이 실패하는 경우 먼저 애플리케이션에 문제가 있는지 여부를 확인합니다. 예상한 대로 인스턴스에서 애플리케이션이 실행되지 않음을 확인한 경우 다음 단계를 수행하십시오.

Amazon EC2 콘솔을 사용하여 손상된 인스턴스를 찾아내려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Instances]를 선택한 다음 인스턴스를 선택합니다.
3. 세부 정보 창에서 [Status Checks] 탭을 클릭하여 모든 [System Status Checks] 및 [Instance Status Checks]에 대한 개별 결과를 확인합니다.

시스템 상태 확인이 실패한 경우 다음 옵션 중 하나를 시도할 수 있습니다.

- 인스턴스 복구 경보를 만듭니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 [인스턴스를 중지, 종료 또는 복구하는 경보 만들기](#) 섹션을 참조하십시오.
- Amazon EBS 지원 AMI를 사용하는 인스턴스의 경우, 인스턴스를 중지했다가 다시 시작합니다.
- 인스턴스 스토어 스토리지 AMI를 사용하는 인스턴스의 경우 해당 인스턴스를 종료한 후 대체 인스턴스를 시작합니다.
- Amazon EC2에서 문제를 해결할 때까지 기다립니다.
- 문제를 [Amazon EC2 forum](#)에 게시합니다.
- 시스템 로그를 검색하여 오류가 있는지 검토합니다.

시스템 로그 검색

인스턴스 상태 검사가 실패할 경우 인스턴스를 재부팅하여 시스템 로그를 검색할 수 있습니다. 이 로그를 확인하여 문제 해결에 도움이 될 수 있는 오류를 밝혀 낼 수 있습니다. 재부팅하면 로그에서 필요 없는 정보가 지워집니다.

인스턴스를 재부팅하고 시스템 로그를 검색하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Instances]를 선택하고 인스턴스를 선택합니다.
3. [Actions], [Instance State], [Reboot]를 차례로 선택합니다. 인스턴스가 재부팅되는 데 몇 분 정도 걸릴 수 있습니다.
4. 문제가 계속되는지 확인합니다. 경우에 따라 재부팅으로 문제가 해결될 수도 있습니다.
5. 인스턴스가 `running` 상태일 경우 [Actions], [Instance Settings], [Get System Log]를 차례로 선택합니다.
6. 화면에 표시되는 로그를 검토한 후 아래에 나와 있는 시스템 오류 구문 목록을 참조하여 문제를 해결합니다.

7. 경험한 상황이 검사 결과와 다른 경우 또는 검사에서 검색되지 않는 인스턴스 관련 문제가 있는 경우, [Status Checks] 탭의 [Submit feedback]을 선택하면 검색 테스트를 개선하는 데 도움이 될 수 있습니다.
8. 문제가 해결되지 않으면 해당 문제를 [Amazon EC2 forum](#)에 게시할 수 있습니다.

Linux 기반 인스턴스의 시스템 로그 오류 문제 해결

Linux 기반 인스턴스가 인스턴스 액세스 검사와 같은 인스턴스 상태 확인에 실패한 경우, 위의 단계에 따라 시스템 로그를 가져왔는지 확인합니다. 다음 목록에는 일반적인 시스템 로그 오류와 각 오류에 대한 문제를 해결하기 위해 수행할 수 있는 권장 조치가 나와 있습니다.

메모리 오류

- 메모리 부족: 프로세스 중지 (p. 710)
- ERROR: mmu_update failed(메모리 관리 업데이트 실패) (p. 710)

디바이스 오류

- I/O 오류(블록 디바이스 오류) (p. 711)
- IO ERROR: neither local nor remote disk(분산된 블록 디바이스 손상) (p. 712)

커널 오류

- request_module: runaway loop modprobe(이전 Linux 버전에서 레거시 커널 modprobe 반복) (p. 713)
- "FATAL: kernel too old" 및 "fsck: No such file or directory while trying to open /dev"(커널과 AMI 불일치) (p. 713)
- "FATAL: Could not load /lib/modules" 또는 "BusyBox"(커널 모듈 누락) (p. 714)
- ERROR Invalid kernel(EC2 커널이 호환되지 않음) (p. 715)

파일 시스템 오류

- request_module: runaway loop modprobe(이전 Linux 버전에서 레거시 커널 modprobe 반복) (p. 716)
- fsck: No such file or directory while trying to open...(파일 시스템을 찾을 수 없음) (p. 717)
- 파일 시스템 마운트 관련 일반 오류(마운트 실패) (p. 718)
- VFS: Unable to mount root fs on unknown-block(루트 파일 시스템 불일치) (p. 720)
- Error: Unable to determine major/minor number of root device...(루트 파일 시스템/디바이스 불일치) (p. 721)
- XENBUS: Device with no driver... (p. 722)
- ... days without being checked, check forced(파일 시스템 검사 필요) (p. 723)
- fsck died with exit status...(디바이스 누락) (p. 723)

운영 체제 오류

- GRUB 프롬프트(grubdom>) (p. 724)
- Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring(eth0 인터페이스를 가져오는 중: eth0 디바이스의 MAC 주소가 틀려서 무시합니다). (하드 코딩된 MAC 주소) (p. 726)
- Unable to load SELinux Policy. Machine is in enforcing mode. Halting now(SELinux 정책을 가져올 수 없습니다. 시스템이 강제 실행 모드입니다. 중단됩니다). (잘못된 SELinux 구성) (p. 727)
- XENBUS: Timeout connecting to devices(Xenbus 시간 초과) (p. 728)

메모리 부족: 프로세스 중지

메모리 부족 오류는 아래 표시된 것과 비슷한 시스템 로그 항목으로 표시됩니다.

```
[115879.769795] Out of memory: kill process 20273 (httpd) score 1285879
or a child
[115879.769795] Killed process 1917 (php-cgi) vsz:467184kB, anon-
rss:101196kB, file-rss:204kB
```

예상 원인

메모리가 모두 사용되었습니다.

권장 조치

이 인스턴스 유형의 경우	수행할 작업
Amazon EBS 지원	<p>다음 중 하나를 수행하십시오.</p> <ul style="list-style-type: none">인스턴스를 중지하고 다른 인스턴스 유형을 사용하도록 인스턴스를 수정한 다음, 인스턴스를 다시 시작합니다. 예를 들면 더 크거나 메모리 최적화된 인스턴스 유형을 사용합니다.인스턴스를 재부팅하여 손상되지 않은 상태로 복원합니다. 인스턴스 유형을 변경하지 않는 한 이 문제가 다시 발생할 것입니다.
인스턴스 스토어 지원	<p>다음 중 하나를 수행하십시오.</p> <ul style="list-style-type: none">인스턴스를 종료하고 다른 인스턴스 유형을 지정한 새 인스턴스를 시작합니다. 예를 들면 더 크거나 메모리 최적화된 인스턴스 유형을 사용합니다.인스턴스를 재부팅하여 손상되지 않은 상태로 복원합니다. 인스턴스 유형을 변경하지 않는 한 이 문제가 다시 발생할 것입니다.

ERROR: mmu_update failed(메모리 관리 업데이트 실패)

메모리 관리 업데이트 실패는 다음과 비슷한 시스템 로그 항목으로 표시됩니다.

```
...
Press `ESC' to enter the menu... 0      [H[J  Booting 'Amazon Linux 2011.09
(2.6.35.14-95.38.amzn1.i686)'

root (hd0)
Filesystem type is ext2fs, using whole disk
kernel /boot/vmlinuz-2.6.35.14-95.38.amzn1.i686 root=LABEL=/ console=hvc0 LANG=
en_US.UTF-8 KEYTABLE=us
```

```
initrd /boot/initramfs-2.6.35.14-95.38.amzn1.1686.img
ERROR: mmu_update failed with rc=-22
```

예상 원인

Amazon Linux 관련 문제

권장 조치

문제를 [개발자 포럼](#)에 게시하거나, [AWS Support](#)에 문의하십시오.

I/O 오류(블록 디바이스 오류)

입/출력 오류는 다음 예와 비슷한 시스템 로그 항목으로 표시됩니다.

```
[9943662.053217] end_request: I/O error, dev sde, sector 52428288
[9943664.191262] end_request: I/O error, dev sde, sector 52428168
[9943664.191285] Buffer I/O error on device md0, logical block 209713024
[9943664.191297] Buffer I/O error on device md0, logical block 209713025
[9943664.191304] Buffer I/O error on device md0, logical block 209713026
[9943664.191310] Buffer I/O error on device md0, logical block 209713027
[9943664.191317] Buffer I/O error on device md0, logical block 209713028
[9943664.191324] Buffer I/O error on device md0, logical block 209713029
[9943664.191332] Buffer I/O error on device md0, logical block 209713030
[9943664.191339] Buffer I/O error on device md0, logical block 209713031
[9943664.191581] end_request: I/O error, dev sde, sector 52428280
[9943664.191590] Buffer I/O error on device md0, logical block 209713136
[9943664.191597] Buffer I/O error on device md0, logical block 209713137
[9943664.191767] end_request: I/O error, dev sde, sector 52428288
[9943664.191970] end_request: I/O error, dev sde, sector 52428288
[9943664.192143] end_request: I/O error, dev sde, sector 52428288
[9943664.192949] end_request: I/O error, dev sde, sector 52428288
[9943664.193112] end_request: I/O error, dev sde, sector 52428288
[9943664.193266] end_request: I/O error, dev sde, sector 52428288
...
```

예상 원인

인스턴스 유형	예상 원인
Amazon EBS 지원	실패한 Amazon EBS 볼륨
인스턴스 스토어 지원	물리적 드라이브 실패

권장 조치

이 인스턴스 유형의 경우	수행할 작업
Amazon EBS 지원	다음 절차를 수행하십시오. 1. 인스턴스를 중지합니다. 2. 볼륨을 분리합니다. 3. 볼륨 복구를 시도합니다.

이 인스턴스 유형의 경우	수행할 작업
	<p style="text-align: center;">Note</p> <p>Amazon EBS 볼륨의 스냅샷을 정기적으로 생성하는 것이 좋습니다. 그러면 오류로 인한 데이터 손실의 위험을 크게 줄일 수 있습니다.</p> <ul style="list-style-type: none"> 4. 볼륨을 인스턴스에 다시 연결합니다. 5. 볼륨을 분리합니다.
인스턴스 스토어 지원	<p style="text-align: center;">Note</p> <p>데이터를 복구할 수 없습니다. 백업에서 복구합니다.</p> <p style="text-align: center;">Note</p> <p>백업용으로 Amazon S3 또는 Amazon EBS를 사용하는 것이 좋습니다. 인스턴스 스토어 볼륨이 하나의 호스트 및 하나의 디스크 오류와 연결됩니다.</p>

IO ERROR: neither local nor remote disk(분산된 블록 디바이스 손상)

디바이스에 대한 입/출력 오류는 다음 예와 비슷한 시스템 로그 항목으로 표시됩니다.

```
...
block drbd1: Local IO failed in request_timer_fn. Detaching...
Aborting journal on device drbd1-8.
block drbd1: IO ERROR: neither local nor remote disk
Buffer I/O error on device drbd1, logical block 557056
lost page write due to I/O error on drbd1
JBD2: I/O error detected when updating journal superblock for drbd1-8.
```

예상 원인

인스턴스 유형	예상 원인
Amazon EBS 지원	실패한 Amazon EBS 볼륨
인스턴스 스토어 지원	물리적 드라이브 실패

권장 조치

인스턴스를 종료하고 새 인스턴스를 시작합니다.

Amazon EBS 지원 인스턴스의 경우 해당 인스턴스로부터 이미지를 만들어서 최근 스냅샷에서 데이터를 복구할 수 있습니다. 스냅샷 이후에 추가된 데이터는 복구할 수 없습니다.

request_module: runaway loop modprobe(이전 Linux 버전에서 레거시 커널 modprobe 반복)

이 상태는 아래 표시된 것과 비슷한 시스템 로그로 표시됩니다. 불안정하거나 이전 Linux 커널(예: 2.6.16-xenU)을 사용하면 시작 시 무한 반복 상태가 발생합니다.

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007

BIOS-provided physical RAM map:

Xen: 0000000000000000 - 0000000026700000 (usable)

OMB HIGHMEM available.
...

request_module: runaway loop modprobe binfmt-464c
```

권장 조치

이 인스턴스 유형의 경우	수행할 작업
Amazon EBS 지원	다음 옵션 중 하나를 사용하여 GRUB 기반이든 정적이든, 최신 커널을 사용합니다: 옵션 1: 인스턴스를 종료하고 -kernel 및 -ramdisk 매개 변수가 지정된 새 인스턴스를 시작합니다. 옵션 2: <ol style="list-style-type: none">1. 인스턴스를 중지합니다.2. 최신 커널이 사용되도록 kernel 및 ramdisk 특성을 설정합니다.3. 인스턴스를 시작합니다.
인스턴스 스토어 지원	인스턴스를 종료하고 -kernel 및 -ramdisk 매개 변수가 지정된 새 인스턴스를 시작합니다.

"FATAL: kernel too old" 및 "fsck: No such file or directory while trying to open /dev"(커널과 AMI 불일치)

이 상태는 아래 표시된 것과 비슷한 시스템 로그로 표시됩니다.

Amazon Elastic Compute Cloud
Linux 인스턴스용 사용 설명서
"FATAL: Could not load /lib/modules"
또는 "BusyBox"(커널 모듈 누락)

```
Linux version 2.6.16.33-xenU (root@dom0-0-50-45-1-a4-ee.z-2.aes0.internal)
(gcc version 4.1.1 20070105 (Red Hat 4.1.1-52)) #2 SMP Wed Aug 15 17:27:36 SAST 2007
...
FATAL: kernel too old
Kernel panic - not syncing: Attempted to kill init!
```

예상 원인

kernel과 userland가 호환되지 않습니다.

권장 조치

이 인스턴스 유형의 경우	수행할 작업
Amazon EBS 지원	<p>다음 절차를 수행하십시오.</p> <ol style="list-style-type: none"> 인스턴스를 중지합니다. 최신 커널이 사용되도록 구성을 수정합니다. 인스턴스를 시작합니다.
인스턴스 스토어 지원	<p>다음 절차를 수행하십시오.</p> <ol style="list-style-type: none"> 최신 커널을 사용하는 AMI를 만듭니다. 인스턴스를 종료합니다. 만든 AMI에서 새 인스턴스를 시작합니다.

"FATAL: Could not load /lib/modules" 또는 "BusyBox"(커널 모듈 누락)

이 상태는 아래 표시된 것과 비슷한 시스템 로그로 표시됩니다.

```
[    0.370415] Freeing unused kernel memory: 1716k freed
Loading, please wait...
WARNING: Couldn't open directory /lib/modules/2.6.34-4-virtual: No such file or directory
FATAL: Could not open /lib/modules/2.6.34-4-virtual/modules.dep.temp for writing: No such
      file or directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or directory
Couldn't get a file descriptor referring to the console
Begin: Loading essential drivers... ...
Begin: Running /scripts/init-premount ...
Done.
Begin: Running /scripts/local-top ...
Done.
Begin: Waiting for root file system... ...
Done.
Gave up waiting for root device.  Common problems:
 - Boot args (cat /proc/cmdline)
   - Check rootdelay= (did the system wait long enough?)
   - Check root= (did the system wait for the right device?)
 - Missing modules (cat /proc/modules; ls /dev)
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or directory
```

```
ALERT! /dev/sda1 does not exist. Dropping to a shell!  
  
BusyBox v1.13.3 (Ubuntu 1:1.13.3-1ubuntu5) built-in shell (ash)  
Enter 'help' for a list of built-in commands.  
(initramfs)
```

예상 원인

이 문제는 다음 상태 중 하나 이상으로 인해 발생할 수 있습니다.

- ramdisk가 없습니다.
- ramdisk에 올바른 모듈이 없습니다.
- Amazon EBS 루트 볼륨이 /dev/sda1에 올바르게 연결되지 않았습니다.

권장 조치

이 인스턴스 유형의 경우	수행할 작업
Amazon EBS 지원	<p>다음 절차를 수행하십시오.</p> <ol style="list-style-type: none">1. Amazon EBS 볼륨에 맞게 수정된 ramdisk를 선택합니다.2. 인스턴스를 중지합니다.3. 볼륨을 분리하고 복구합니다.4. 볼륨을 인스턴스에 연결합니다.5. 인스턴스를 시작합니다.6. 수정된 ramdisk를 사용하도록 AMI를 변경합니다.
인스턴스 스토어 지원	<p>다음 절차를 수행하십시오.</p> <ol style="list-style-type: none">1. 인스턴스를 종료하고 올바른 ramdisk를 사용하여 새 인스턴스를 시작합니다.2. 올바른 ramdisk를 사용하여 새 AMI를 만듭니다.

ERROR Invalid kernel(EC2 커널이 호환되지 않음)

이 상태는 아래 표시된 것과 비슷한 시스템 로그로 표시됩니다.

```
...  
root (hd0)  
  
Filesystem type is ext2fs, using whole disk  
  
kernel /vmlinuz root=/dev/sda1 ro  
  
initrd /initrd.img  
  
ERROR Invalid kernel: elf_xen_note_check: ERROR: Will only load images  
built for the generic loader or Linux images  
xc_dom_parse_image returned -1  
  
Error 9: Unknown boot failure
```

```
Booting 'Fallback'

root (hd0)

Filesystem type is ext2fs, using whole disk

kernel /vmlinuz.old root=/dev/sda1 ro

Error 15: File not found
```

예상 원인

이 문제는 다음 상태 중 하나 또는 두 가지 모두로 인해 발생할 수 있습니다.

- 제공된 커널이 GRUB에서 지원되지 않습니다.
- 대체 커널이 존재하지 않습니다.

권장 조치

이 인스턴스 유형의 경우	수행할 작업
Amazon EBS 지원	다음 절차를 수행하십시오. <ol style="list-style-type: none">인스턴스를 중지합니다.작동 중인 커널로 대체합니다.대체 커널을 설치합니다.올바른 커널로 변경하여 AMI를 수정합니다.
인스턴스 스토어 지원	다음 절차를 수행하십시오. <ol style="list-style-type: none">인스턴스를 종료하고 올바른 커널을 사용하여 새 인스턴스를 시작합니다.올바른 커널을 사용하여 AMI를 만듭니다.(옵션) AWS Support에 데이터 복구 기술 지원을 요청합니다.

request_module: runaway loop modprobe(이전 Linux 버전에서 레거시 커널 modprobe 반복)

이 상태는 아래 표시된 것과 비슷한 시스템 로그로 표시됩니다. 불안정하거나 이전 Linux 커널(예: 2.6.16-xenU)을 사용하면 시작 시 무한 반복 상태가 발생합니다.

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007

BIOS-provided physical RAM map:

Xen: 0000000000000000 - 0000000026700000 (usable)

OMB HIGHMEM available.
...

request_module: runaway loop modprobe binfmt-464c
```

Amazon Elastic Compute Cloud
Linux 인스턴스용 사용 설명서
fsck: No such file or directory while trying
to open...(파일 시스템을 찾을 수 없음)

```
request_module: runaway loop modprobe binfmt-464c
```

권장 조치

이 인스턴스 유형의 경우	수행할 작업
Amazon EBS 지원	<p>다음 옵션 중 하나를 사용하여 GRUB 기반이든 정적이든, 최신 커널을 사용합니다:</p> <p>옵션 1: 인스턴스를 종료하고 <code>-kernel</code> 및 <code>-ramdisk</code> 매개 변수가 지정된 새 인스턴스를 시작합니다.</p> <p>옵션 2:</p> <ol style="list-style-type: none"> 1. 인스턴스를 중지합니다. 2. 최신 커널이 사용되도록 <code>kernel</code> 및 <code>ramdisk</code> 특성을 설정합니다. 3. 인스턴스를 시작합니다.
인스턴스 스토어 지원	인스턴스를 종료하고 <code>-kernel</code> 및 <code>-ramdisk</code> 매개 변수가 지정된 새 인스턴스를 시작합니다.

fsck: No such file or directory while trying to open...(파일 시스템을 찾을 수 없음)

이 상태는 아래 표시된 것과 비슷한 시스템 로그로 표시됩니다.

```
Welcome to Fedora
Press 'I' to enter interactive startup.
Setting clock : Wed Oct 26 05:52:05 EDT 2011 [ OK ]
Starting udev: [ OK ]
Setting hostname localhost: [ OK ]

No devices found
Setting up Logical Volume Management: File descriptor 7 left open
  No volume groups found
[ OK ]

Checking filesystems
Checking all file systems.
[/sbin/fsck.ext3 (1) -- /] fsck.ext3 -a /dev/sda1
/dev/sda1: clean, 82081/1310720 files, 2141116/2621440 blocks
[/sbin/fsck.ext3 (1) -- /mnt/dbbackups] fsck.ext3 -a /dev/sdh
fsck.ext3: No such file or directory while trying to open /dev/sdh

/dev/sdh:
The superblock could not be read or does not describe a correct ext2
filesystem. If the device is valid and it really contains an ext2
```

```
filesystem (and not swap or ufs or something else), then the superblock  
is corrupt, and you might try running e2fsck with an alternate superblock:  
e2fsck -b 8193 <device>
```

[FAILED]

```
*** An error occurred during the file system check.  
*** Dropping you to a shell; the system will reboot  
*** when you leave the shell.  
Give root password for maintenance  
(or type Control-D to continue):
```

예상 원인

- ramdisk 파일 시스템 정의 /etc/fstab에 버그가 있습니다.
- /etc/fstab에서 파일 시스템 정의가 잘못 구성되었습니다.
- 드라이브 누락/실패

권장 조치

이 인스턴스 유형의 경우	수행할 작업
Amazon EBS 지원	<p>다음 절차를 수행하십시오.</p> <ol style="list-style-type: none">인스턴스를 중지하고 루트 볼륨을 분리한 다음, 볼륨의 /etc/fstab를 복구/수정하고 볼륨을 인스턴스에 연결한 다음, 인스턴스를 시작합니다.수정된 /etc/fstab가 포함되도록 ramdisk를 수정합니다(해당되는 경우).최신 ramdisk를 사용하도록 AMI를 수정합니다. <p>fstab의 여섯 번째 필드는 마운트 가용성 요구 사항을 정의합니다. 즉, 값이 0이 아니면 해당 볼륨에서 fsck가 성공적으로 수행되어야 함을 의미합니다. 일반적으로 Amazon EC2에서는 대화형 콘솔 프롬프트가 지원되지 않아 오류가 발생하므로 Amazon EC2에서는 이 필드 사용 시 문제가 발생할 수 있습니다. 이 기능을 사용할 때는 각별히 주의해야 하며 Linux 맨 페이지에서 fstab에 대한 설명을 참조하십시오.</p>
인스턴스 스토어 지원	<p>다음 절차를 수행하십시오.</p> <ol style="list-style-type: none">인스턴스를 종료하고 새 인스턴스를 시작합니다.잘못된 Amazon EBS 볼륨을 모두 분리하고 인스턴스를 재부팅합니다.(옵션) AWS Support에 데이터 복구 기술 지원을 요청합니다.

파일 시스템 마운트 관련 일반 오류(마운트 실패)

이 상태는 아래 표시된 것과 비슷한 시스템 로그로 표시됩니다.

```
Loading xenblk.ko module
xen-vbd: registered block device major 8

Loading ehci-hcd.ko module
Loading ohci-hcd.ko module
Loading uhci-hcd.ko module
USB Universal Host Controller Interface driver v3.0

Loading mbcache.ko module
Loading jbd.ko module
Loading ext3.ko module
Creating root device.
Mounting root filesystem.
kjournald starting. Commit interval 5 seconds

EXT3-fs: mounted filesystem with ordered data mode.

Setting up other filesystems.
Setting up new root fs
no fstab.sys, mounting internal defaults
Switching to new root and running init.
unmounting old /dev
unmounting old /proc
unmounting old /sys
mountall:/proc: unable to mount: Device or resource busy
mountall:/proc/self/mountinfo: No such file or directory
mountall: root filesystem isn't mounted
init: mountall main process (221) terminated with status 1

General error mounting filesystems.
A maintenance shell will now be started.
CONTROL-D will terminate this shell and re-try.
Press enter for maintenance
(or type Control-D to continue):
```

예상 원인

인스턴스 유형	예상 원인
Amazon EBS 지원	<ul style="list-style-type: none">Amazon EBS 볼륨 분리 또는 실패.파일 시스템 손상.ramdisk와 AMI 조합의 불일치(예: Debian ramdisk와 SUSE AMI).
인스턴스 스토어 지원	<ul style="list-style-type: none">드라이브 실패.파일 시스템 손상.ramdisk와 조합의 불일치(예: Debian ramdisk와 SUSE AMI).

권장 조치

이 인스턴스 유형의 경우	수행할 작업
Amazon EBS 지원	<p>다음 절차를 수행하십시오.</p> <ol style="list-style-type: none">인스턴스를 종지합니다.루트 볼륨을 분리합니다.

Amazon Elastic Compute Cloud
 Linux 인스턴스용 사용 설명서
VFS: Unable to mount root fs on unknown-block(루트 파일 시스템 불일치)

이 인스턴스 유형의 경우	수행할 작업
	<ol style="list-style-type: none"> 3. 루트 볼륨을 작동 중인 것으로 알려진 인스턴스에 연결합니다. 4. 파일 시스템 검사(fsck -a /dev/...)를 실행합니다. 5. 오류를 모두 수정합니다. 6. 작동 중인 것으로 알려진 인스턴스에서 볼륨을 분리합니다. 7. 중지된 인스턴스에 볼륨을 연결합니다. 8. 인스턴스를 시작합니다. 9. 인스턴스 상태를 다시 확인합니다.
인스턴스 스토어 지원	<p>다음 중 하나를 시도하십시오.</p> <ul style="list-style-type: none"> • 새 인스턴스를 시작합니다. • (옵션) AWS Support에 데이터 복구 기술 지원을 요청합니다.

VFS: Unable to mount root fs on unknown-block(루트 파일 시스템 불일치)

이 상태는 아래 표시된 것과 비슷한 시스템 로그로 표시됩니다.

```

Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007
...
Kernel command line: root=/dev/sda1 ro 4
...
Registering block device major 8
...
Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(8,1)

```

예상 원인

인스턴스 유형	예상 원인
Amazon EBS 지원	<ul style="list-style-type: none"> • 디바이스가 올바르게 연결되지 않았습니다. • 루트 디바이스가 올바른 디바이스 지점에서 연결되지 않았습니다. • 필요한 형식의 파일 시스템이 아닙니다. • 레거시 커널(예: 2.6.16-XenU)이 사용되었습니다.
인스턴스 스토어 지원	하드웨어 디바이스 실패.

권장 조치

이 인스턴스 유형의 경우	수행할 작업
Amazon EBS 지원	<p>다음 중 하나를 수행하십시오.</p> <ul style="list-style-type: none"> • 인스턴스를 중지했다가 다시 시작합니다.

Amazon Elastic Compute Cloud
 Linux 인스턴스용 사용 설명서
 Error: Unable to determine major/minor number of
 root device...(루트 파일 시스템/디바이스 불일치)

이 인스턴스 유형의 경우	수행할 작업
	<ul style="list-style-type: none"> 올바른 디바이스 지점(예: /dev/sda 대신에 /dev/sda1)에서 연결되도록 루트 볼륨을 수정합니다. 종지하고 현대식 커널을 사용하도록 수정합니다.
인스턴스 스토어 지원	인스턴스를 종료하고 현대식 커널을 사용하여 새 인스턴스를 시작합니다.

Error: Unable to determine major/minor number of root device...(루트 파일 시스템/디바이스 불일치)

이 상태는 아래 표시된 것과 비슷한 시스템 로그로 표시됩니다.

```

...
XENBUS: Device with no driver: device/vif/0
XENBUS: Device with no driver: device/vbd/2048
drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
Initializing network drop monitor service
Freeing unused kernel memory: 508k freed
:: Starting udevd...
done.
:: Running Hook [udev]
:: Triggering uevents...<30>udevd[65]: starting version 173
done.
Waiting 10 seconds for device /dev/xvda1 ...
Root device '/dev/xvda1' doesn't exist. Attempting to create it.
ERROR: Unable to determine major/minor number of root device '/dev/xvda1'.
You are being dropped to a recovery shell
    Type 'exit' to try and continue booting
sh: can't access tty; job control turned off
[ramfs /]#

```

예상 원인

- 가상 블록 디바이스 드라이버가 없거나 잘못 구성되었습니다.
- 디바이스 열거형이 충돌합니다(sda와 xvda 또는 sda1 대신 sda).
- 잘못된 DomU 커널을 선택했습니다.

권장 조치

이 인스턴스 유형의 경우	수행할 작업
Amazon EBS 지원	<p>다음 절차를 수행하십시오.</p> <ol style="list-style-type: none"> 인스턴스를 종지합니다. 볼륨을 분리합니다. 디바이스 매핑 문제를 해결합니다. 인스턴스를 시작합니다. 디바이스 매핑 문제를 해결하도록 AMI를 수정합니다.
인스턴스 스토어 지원	다음 절차를 수행하십시오.

이 인스턴스 유형의 경우	수행할 작업
	<ol style="list-style-type: none">적절히 수정(블록 디바이스를 올바르게 매핑)하여 새 AMI를 만듭니다.인스턴스를 종료하고 만든 AMI에서 새 인스턴스를 시작합니다.

XENBUS: Device with no driver...

이 상태는 아래 표시된 것과 비슷한 시스템 로그로 표시됩니다.

```
XENBUS: Device with no driver: device/vbd/2048
drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
Initializing network drop monitor service
Freeing unused kernel memory: 508k freed
:: Starting udevd...
done.
:: Running Hook [udev]
:: Triggering uevents...<30>udevd[65]: starting version 173
done.
Waiting 10 seconds for device /dev/xvda1 ...
Root device '/dev/xvda1' doesn't exist. Attempting to create it.
ERROR: Unable to determine major/minor number of root device '/dev/xvda1'.
You are being dropped to a recovery shell
    Type 'exit' to try and continue booting
sh: can't access tty; job control turned off
[ramfs/]#
```

예상 원인

- 가상 블록 디바이스 드라이버가 없거나 잘못 구성되었습니다.
- 디바이스 열거형이 충돌합니다(sda와 xvda).
- 잘못된 DomU 커널을 선택했습니다.

권장 조치

이 인스턴스 유형의 경우	수행할 작업
Amazon EBS 지원	<p>다음 절차를 수행하십시오.</p> <ol style="list-style-type: none">인스턴스를 중지합니다.볼륨을 분리합니다.디바이스 매핑 문제를 해결합니다.인스턴스를 시작합니다.디바이스 매핑 문제를 해결하도록 AMI를 수정합니다.
인스턴스 스토어 지원	<p>다음 절차를 수행하십시오.</p> <ol style="list-style-type: none">적절히 수정(블록 디바이스를 올바르게 매핑)하여 새 AMI를 만듭니다.인스턴스를 종료하고 만든 AMI를 사용하여 새 인스턴스를 시작합니다.

... days without being checked, check forced(파일 시스템 검사 필요)

이 상태는 아래 표시된 것과 비슷한 시스템 로그로 표시됩니다.

```
...
Checking filesystems
Checking all file systems.
[/sbin/fsck.ext3 (1) -- /] fsck.ext3 -a /dev/sda1
/dev/sda1 has gone 361 days without being checked, check forced
```

예상 원인

파일 시스템 검사 시간이 경과되었습니다. 파일 시스템 검사가 강제 실행 중입니다

권장 조치

- 파일 시스템 검사가 완료될 때까지 기다립니다. 파일 시스템 검사는 루트 파일 시스템의 크기에 따라 오래 걸릴 수도 있습니다.
- tune2fs 또는 파일 시스템에 적합한 도구를 사용하여 파일 시스템 검사(fsck) 적용을 제거하도록 파일 시스템을 수정합니다.

fsck died with exit status...(디바이스 누락)

이 상태는 아래 표시된 것과 비슷한 시스템 로그로 표시됩니다.

```
Cleaning up ifupdown....
Loading kernel modules...done.
...
Activating lvm and md swap...done.
Checking file systems...fsck from util-linux-ng 2.16.2
/sbin/fsck.xfs: /dev/sdh does not exist
fsck died with exit status 8
[31mfailed (code 8).[39;49m
```

예상 원인

- Ramdisk에서 누락된 드라이브를 찾고 있습니다.
- 파일 시스템 일관성 검사가 강제 실행되었습니다.
- 드라이브 실패 또는 분리

권장 조치

이 인스턴스 유형의 경우	수행할 작업
Amazon EBS 지원	<p>다음 중 하나 이상을 시도하여 문제를 해결하십시오.</p> <ul style="list-style-type: none">인스턴스를 중지하고 볼륨을 기존의 실행 중인 인스턴스에 연결합니다.

이 인스턴스 유형의 경우	수행할 작업
	<ul style="list-style-type: none"> 일관성 검사를 수동으로 실행합니다. 관련 유ти리티를 포함하도록 ramdisk를 수정합니다. 일관성 요구 사항을 제거하도록 파일 시스템 튜닝 매개 변수를 수정합니다(권장되지 않음).
인스턴스 스토어 지원	<p>다음 중 하나 이상을 시도하여 문제를 해결하십시오.</p> <ul style="list-style-type: none"> 올바른 도구로 ramdisk 번들을 다시 구성합니다. 일관성 요구 사항을 제거하도록 파일 시스템 튜닝 매개 변수를 수정합니다(권장되지 않음). 인스턴스를 종료하고 새 인스턴스를 시작합니다. (옵션) AWS Support에 데이터 복구 기술 지원을 요청합니다.

GRUB 프롬프트(grubdom>)

이 상태는 아래 표시된 것과 비슷한 시스템 로그로 표시됩니다.

```
GNU GRUB  version 0.97  (629760K lower / 0K upper memory)

[ Minimal BASH-like line editing is supported.  For
the first word, TAB lists possible command
completions.  Anywhere else TAB lists the possible
completions of a device/filename. ]
```

grubdom>

예상 원인

인스턴스 유형	예상 원인
Amazon EBS 지원	<ul style="list-style-type: none"> GRUB 구성 파일이 없습니다. 잘못된 GRUB 이미지가 사용되었습니다. 다른 위치에 있는 GRUB 구성 파일이 필요합니다. GRUB 구성 파일을 저장하는 데 지원되지 않는 파일 시스템이 사용되었습니다(예: 루트 파일 시스템을 이전 GRUB 버전에서 지원되지 않는 유형으로 변환).
인스턴스 스토어 지원	<ul style="list-style-type: none"> GRUB 구성 파일이 없습니다. 잘못된 GRUB 이미지가 사용되었습니다. 다른 위치에 있는 GRUB 구성 파일이 필요합니다. GRUB 구성 파일을 저장하는 데 지원되지 않는 파일 시스템이 사용되었습니다(예: 루트 파일 시스템을 이전 GRUB 버전에서 지원되지 않는 유형으로 변환).

권장 조치

이 인스턴스 유형의 경우	수행할 작업
Amazon EBS 지원	<p>옵션 1: AMI를 수정하고 인스턴스를 다시 시작합니다.</p> <ol style="list-style-type: none">표준 위치(/boot/grub/menu.lst)에서 GRUB 구성 파일을 만들도록 원본 AMI를 수정합니다.GRUB 버전에서 기본 파일 시스템 유형을 지원하는지 확인하고 필요할 경우 GRUB을 업그레이드 합니다.적합한 GRUB 이미지(hd0-첫 번째 드라이브 또는 hd00 – 첫 번째 드라이브, 첫 번째 파티션)를 선택 합니다.인스턴스를 종료하고 만든 AMI를 사용하여 새 인스턴스를 시작합니다. <p>옵션 2: 기존 인스턴스 수정합니다.</p> <ol style="list-style-type: none">인스턴스를 종지합니다.루트 파일 시스템을 분리합니다.루트 파일 시스템을 작동하는 것으로 알려진 인스턴스에 연결합니다.파일 시스템을 마운트합니다.GRUB 구성 파일을 만듭니다.GRUB 버전에서 기본 파일 시스템 유형을 지원하는지 확인하고 필요할 경우 GRUB을 업그레이드 합니다.파일 시스템을 분리합니다.원래 인스턴스에 연결합니다.적합한 GRUB 이미지(첫 번째 디스크 또는 첫 번째 디스크의 첫 번째 파티션)를 사용하도록 커널 속성을 수정합니다.인스턴스를 시작합니다.
인스턴스 스토어 지원	<p>옵션 1: AMI를 수정하고 인스턴스를 다시 시작합니다.</p> <ol style="list-style-type: none">표준 위치(/boot/grub/menu.lst)에서 GRUB 구성 파일을 사용하여 새 AMI를 만듭니다.적합한 GRUB 이미지(hd0-첫 번째 드라이브 또는 hd00 – 첫 번째 드라이브, 첫 번째 파티션)를 선택 합니다.GRUB 버전에서 기본 파일 시스템 유형을 지원하는지 확인하고 필요할 경우 GRUB을 업그레이드 합니다.인스턴스를 종료하고 만든 AMI를 사용하여 새 인스턴스를 시작합니다. <p>옵션 2: 인스턴스를 종료하고 올바른 커널을 지정하여 새 인스턴스를 시작합니다.</p>

이 인스턴스 유형의 경우	수행할 작업
	<p style="text-align: center;">Note</p> <p style="text-align: center;">기존 인스턴스에서 데이터를 복구하려면 AWS Support에 문의하십시오.</p>

Bringing up interface eth0: Device eth0 has different
 MAC address than expected, ignoring(eth0 인터페이스
 를 가져오는 중: eth0 디바이스의 MAC 주소가 틀려서
 무시합니다). (하드 코딩된 MAC 주소)

이 상태는 아래 표시된 것과 비슷한 시스템 로그로 표시됩니다.

```

...
Bringing up loopback interface: [ OK ]
Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring.
[FAILED]
Starting auditd: [ OK ]

```

예상 원인

AMI 구성에 하드 코딩된 인터페이스 MAC이 있습니다

권장 조치

이 인스턴스 유형의 경우	수행할 작업
Amazon EBS 지원	<p>다음 중 하나를 수행하십시오.</p> <ul style="list-style-type: none"> • 하드 코딩이 제거되도록 AMI를 수정하고 인스턴스를 다시 시작합니다. • 하드 코딩된 MAC 주소가 제거되도록 인스턴스를 수정합니다. <p>또는</p> <p>다음 절차를 수행하십시오.</p> <ol style="list-style-type: none"> 1. 인스턴스를 중지합니다. 2. 루트 볼륨을 분리합니다. 3. 볼륨을 다른 인스턴스에 연결하고 하드 코딩된 MAC 주소가 제거되도록 볼륨을 수정합니다. 4. 볼륨을 원래 인스턴스에 연결합니다. 5. 인스턴스를 시작합니다.
인스턴스 스토어 지원	다음 중 하나를 수행하십시오.

Amazon Elastic Compute Cloud
 Linux 인스턴스용 사용 설명서
 Unable to load SELinux Policy. Machine is in
 enforcing mode. Halting now(SELinux 정책)

이 인스턴스 유형의 경우	수행할 작업
	<ul style="list-style-type: none"> 하드 코팅된 MAC 주소가 제거되도록 인스턴스를 수정합니다. 인스턴스를 종료하고 새 인스턴스를 시작합니다.

Unable to load SELinux Policy. Machine is in enforcing mode. Halting now(SELinux 정책을 가져올 수 없습니다. 시스템이 강제 실행 모드입니다. 중단됩니다). (잘못된 SELinux 구성)

이 상태는 아래 표시된 것과 비슷한 시스템 로그로 표시됩니다.

```
audit(1313445102.626:2): enforcing=1 old_enforcing=0 auid=4294967295
Unable to load SELinux Policy. Machine is in enforcing mode. Halting now.
Kernel panic - not syncing: Attempted to kill init!
```

예상 원인

SELinux가 오류 상태에서 활성화되었습니다.

- 제공된 커널이 GRUB에서 지원되지 않습니다.
- 대체 커널이 존재하지 않습니다.

권장 조치

이 인스턴스 유형의 경우	수행할 작업
Amazon EBS 지원	<p>다음 절차를 수행하십시오.</p> <ol style="list-style-type: none"> 실패한 인스턴스를 종지합니다. 실패한 인스턴스의 루트 볼륨을 분리합니다. 루트 볼륨을 실행 중인 다른 Linux 인스턴스(나중에 복구 인스턴스로 불립니다)에 연결합니다. 복구 인스턴스에 연결하여 실패한 인스턴스의 루트 볼륨을 마운트합니다. 마운트된 루트 볼륨에서 SELinux를 비활성화합니다. 이 과정은 Linux 배포판에 따라 차이가 있습니다. 자세한 내용은 운영 체제별 설명서를 참조하십시오. <p>Note</p> <p>일부 시스템에서는 <code>/mount_point/etc/sysconfig/selinux</code> 파일에서 <code>SELINUX=disabled</code>를 설정함으로써 SELinux를 비활성화합니다. 여기서 <code>mount_point</code>는 복구 인스턴스에서 볼륨을 마운트한 위치입니다.</p>

이 인스턴스 유형의 경우	수행할 작업
	<ol style="list-style-type: none">6. 복구 인스턴스에서 루트 볼륨을 마운트 해제하고 분리한 다음 원본 인스턴스에 다시 연결합니다.7. 인스턴스를 시작합니다.
인스턴스 스토어 지원	<p>다음 절차를 수행하십시오.</p> <ol style="list-style-type: none">1. 인스턴스를 종료하고 새 인스턴스를 시작합니다.2. (옵션) AWS Support에 데이터 복구 기술 지원을 요청합니다.

XENBUS: Timeout connecting to devices(Xenbus 시간 초과)

이 상태는 아래 표시된 것과 비슷한 시스템 로그로 표시됩니다.

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007
...
XENBUS: Timeout connecting to devices!
...
Kernel panic - not syncing: No init found. Try passing init= option to kernel.
```

예상 원인

- 블록 디바이스가 인스턴스에 연결되지 않았습니다
- 이 인스턴스에 매우 오래된 DomU 커널이 사용되고 있습니다

권장 조치

이 인스턴스 유형의 경우	수행할 작업
Amazon EBS 지원	<p>다음 중 하나를 수행하십시오.</p> <ul style="list-style-type: none">• 현대식 커널을 사용하도록 AMI 및 인스턴스를 수정하고 인스턴스를 다시 시작합니다.• 인스턴스를 재부팅합니다.
인스턴스 스토어 지원	<p>다음 중 하나를 수행하십시오.</p> <ul style="list-style-type: none">• 인스턴스를 종료합니다.• 현대식 커널을 사용하도록 AMI를 수정하고 이 AMI를 사용하여 새 인스턴스를 시작합니다.

인스턴스 용량 문제 해결

다음 오류는 인스턴스 용량과 관련이 있습니다.

Error: InsufficientInstanceCapacity

인스턴스를 시작하거나 중지된 인스턴스를 시작하려 할 때 `InsufficientInstanceCapacity` 오류가 발생하면 현재 AWS에 요청에 대한 서비스를 제공할 수 있을 만큼 가용 용량이 충분하지 않은 것입니다. 다음을 수 행해 보십시오.

- 몇 분 정도 기다린 후 다시 요청을 제출합니다. 용량은 자주 변할 수 있습니다.
- 인스턴스 수가 줄어든 새 요청을 제출하십시오. 예를 들어 단일 요청을 통해 인스턴스 15개를 시작하는 경우 인스턴스 5개에 대해 요청 3개 또는 인스턴스 1개 대신 요청 15개를 시도합니다.
- 인스턴스를 시작하고 있는 경우 가용 영역을 지정하지 않고 새 요청을 제출하십시오.
- 인스턴스를 시작하고 있는 경우 이후의 단계에서 크기를 조정할 수 있는 다른 인스턴스 유형을 사용하여 새 요청을 제출하십시오. 자세한 내용은 [인스턴스 크기 조정 \(p. 169\)](#) 섹션을 참조하십시오.
- 예약 인스턴스를 구매하십시오. 예약 인스턴스는 장기 용량 예약입니다. 자세한 내용은 [Amazon EC2 예약 인스턴스](#)를 참조하십시오.

Error: InstanceLimitExceeded

인스턴스를 시작하려 할 때 `InstanceLimitExceeded` 오류가 발생하면 동시 실행 인스턴스 제한에 도달한 것입니다. 새 AWS 계정의 경우 기본 제한은 20입니다. 추가 실행 인스턴스가 필요할 경우 [Amazon EC2 인스턴스 제한 증가 요청](#)에서 양식을 작성하십시오.

콘솔 출력 가져오기 및 인스턴스 재부팅

콘솔 출력은 문제 진단을 위한 유용한 도구입니다. 인스턴스가 종료되거나 SSH 데몬을 시작하기 전에 연결 할 수 없게 되는 서비스 구성 문제 또는 커널 문제를 해결하는 데 특히 유용합니다.

이와 마찬가지로, 연결할 수 없게 된 인스턴스를 재부팅하는 기능도 문제 해결과 일반 인스턴스 관리용으로 유용합니다.

EC2 인스턴스에는 콘솔 출력을 볼 수 있는 물리적 모니터가 없습니다. 또한 전원 공급, 재부팅 또는 종료 기 능을 제공하는 물리적 컨트롤러도 없습니다. 그 대신, Amazon EC2 API 및 명령줄 인터페이스(CLI)를 통해 이 러한 작업을 수행합니다.

인스턴스 재부팅

Reset 버튼을 눌러서 컴퓨터를 재설정할 수 있게 되는 즉시, Amazon EC2 콘솔, CLI 또는 API를 사용하여 EC2 인스턴스를 재설정할 수 있습니다. 자세한 내용은 [인스턴스 재부팅 \(p. 288\)](#) 섹션을 참조하십시오.

Warning

Windows 인스턴스의 경우, 이 작업으로 하드 재부팅이 수행되며, 이로 인해 데이터가 손상될 수 있 습니다.

인스턴스 콘솔 출력

Linux/Unix 인스턴스의 경우 컴퓨터에 연결된 물리적 컴퓨터에 일반적으로 표시되는 정확한 콘솔 출력이 인 스턴스 콘솔 출력에 표시됩니다. 이 출력은 버퍼링됩니다. 인스턴스에서 생성되고 나서 인스턴스 소유자가 검색할 수 있는 스토리지에 게시되기 때문입니다.

Windows 인스턴스의 경우 인스턴스 콘솔 출력에 마지막 세 개의 시스템 이벤트 로그 오류가 표시됩니다.

게시된 출력은 지속적으로 업데이트되지 않습니다. 최대값일 것 같을 때만 업데이트됩니다. 여기에는 인스턴 스가 부팅되고 나서 얼마 후, 재부팅 후 그리고 인스턴스 종료 시점이 포함됩니다.

Note

게시된 출력의 최근 64 KB만 저장되며, 마지막 게시 후 1시간 이상의 분량이 제공되는 셈입니다.

인스턴스 소유자만 콘솔 출력에 액세스할 수 있습니다. 콘솔이나 명령줄을 사용하여 인스턴스 관련 콘솔 출력을 검색할 수 있습니다.

콘솔을 사용하여 콘솔 출력 가져오기

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 [Instances]를 선택하고 인스턴스를 선택합니다.
3. [Actions], [Instance Settings], [Get System Log]를 차례로 선택합니다.

명령줄을 사용하여 콘솔 출력 가져오기

다음 명령 중 하나를 사용할 수 있습니다. 이러한 명령줄 인터페이스에 대한 자세한 내용은 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [get-console-output\(AWS CLI\)](#)
- [Get-EC2ConsoleOutput\(Windows PowerShell용 AWS 도구\)](#)

일반 시스템 로그 오류에 대한 자세한 내용은 [Linux 기반 인스턴스의 시스템 로그 오류 문제 해결 \(p. 709\)](#)를 참조하십시오.

연결할 수 없는 인스턴스의 스크린샷 캡처

SSH 또는 RDP를 통해 인스턴스에 연결할 수 없는 경우 인스턴스의 스크린샷을 캡처하여 이미지로 볼 수 있습니다. 즉, 인스턴스의 상태에 관한 가시성이 제공되므로 더 빠르게 문제를 해결할 수 있습니다.

이 스크린샷을 위한 데이터 전송 비용은 따로 들지 않습니다. 이미지는 100KB보다 크지 않은 크기의 JPG 형식으로 생성됩니다.

인스턴스 콘솔에 액세스하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 [Instances]를 선택합니다.
3. 캡처할 인스턴스를 선택합니다.
4. [Actions], [Instance Settings]를 차례로 선택합니다.
5. [Get Instance Screenshot]을 선택합니다.

다운로드할 이미지를 마우스 오른쪽 버튼으로 클릭하고 저장합니다.

명령줄을 이용하여 스크린을 캡처하려면

다음 명령 중 하나를 사용할 수 있습니다. 반환되는 내용은 base64-encoded입니다. 다음의 명령줄 인터페이스에 대한 자세한 정보는 [Amazon EC2에 액세스 \(p. 3\)](#) 섹션을 참조하십시오.

- [get-console-screenshot\(AWS CLI\)](#)
- [GetConsoleScreenshot\(Amazon EC2 Query API\)](#)

호스트 컴퓨터 실패 시 인스턴스 복구

기본 호스트 컴퓨터 하드웨어와 관련하여 복구할 수 없는 문제가 발생한 경우 AWS는 인스턴스 종지 이벤트를 예약할 수 있습니다. 이러한 이벤트가 발생하기 전에 이메일을 통해 알림이 전달됩니다.

실패한 호스트 컴퓨터에서 실행 중인 Amazon EBS 지원 인스턴스를 복구하려면

1. 인스턴스 스토어 볼륨의 중요 데이터를 Amazon EBS 또는 Amazon S3으로 백업합니다.
2. 인스턴스를 종지합니다.
3. 인스턴스를 시작합니다.
4. 중요 데이터를 복원합니다.
5. [EC2-Classic] 인스턴스에 Elastic IP 주소가 연결되어 있는 경우 이 주소를 해당 인스턴스에 다시 연결해야 합니다.

자세한 내용은 [인스턴스 종지 및 시작 \(p. 285\)](#)을 참조하십시오.

실패한 호스트 컴퓨터에서 실행 중인 인스턴스 스토어 지원 인스턴스를 복구하려면

1. 인스턴스에서 AMI를 만듭니다.
2. 이미지를 Amazon S3으로 업로드합니다.
3. 중요 데이터를 Amazon EBS 또는 Amazon S3으로 백업합니다.
4. 인스턴스를 종료합니다.
5. AMI에서 새 인스턴스를 시작합니다.
6. 중요 데이터를 새 인스턴스로 모두 복원합니다.
7. [EC2-Classic] 원래 인스턴스에 Elastic IP 주소가 연결되어 있는 경우 이 주소를 새 인스턴스에 다시 연결해야 합니다.

자세한 내용은 [인스턴스 스토어 기반 Linux AMI 생성 \(p. 84\)](#) 섹션을 참조하십시오.

인스턴스가 잘못된 볼륨에서 부팅될 경우

/dev/xvda 또는 /dev/sda에 연결된 볼륨이 아닌 다른 볼륨이 인스턴스의 루트 볼륨이 되는 경우가 있을 수 있습니다. 이 상황은 다른 인스턴스의 루트 볼륨이나 루트 볼륨의 스냅샷에서 생성된 볼륨을 기준 루트 볼륨의 인스턴스에 연결한 경우에 발생할 수 있습니다.

이 문제는 Linux의 첫 번째 ramdisk의 작동 방식 때문에 야기됩니다. 보통 /etc/fstab에서 /로 정의된 볼륨을 선택하게 되는데, Amazon Linux 등의 일부 배포에서는 이를 볼륨 파티션에 연결된 레이블로 확인합니다. 특히 /etc/fstab의 내용이 다음과 같을 수 있습니다.

```
LABEL=/ / ext4 defaults,noatime 1 1
tmpfs /dev/shm tmpfs defaults 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0
```

이때 두 볼륨의 레이블을 확인하면 두 볼륨 모두 / 레이블을 포함하는 것을 볼 수 있습니다.

```
[ec2-user ~]$ sudo e2label /dev/xvda1
/
[ec2-user ~]$ sudo e2label /dev/xvdf1
/
```

이 예에서, 처음에 부팅 대상으로 의도했던 /dev/xvda1 볼륨 대신에 /dev/xvdf1이 ramdisk 실행 후 인스턴스가 부팅되는 루트 디바이스가 될 수 있습니다. 이 문제를 간단히 해결하려면 동일한 e2label 명령을 사용하여 부팅 볼륨이 되게 하지 않으려는 볼륨의 레이블을 변경하면 됩니다.

Note

경우에 따라 /etc/fstab에 UUID를 지정하여 이 문제를 해결할 수 있지만, 두 볼륨이 모두 동일한 스냅샷에서 생성된 경우 또는 두 번째 볼륨이 주 볼륨의 스냅샷에서 생성된 경우에는 두 볼륨이 UUID를 공유합니다.

```
[ec2-user ~]$ sudo blkid  
/dev/xvda1: LABEL="/" UUID=73947a77-ddbe-4dc7-bd8f-3fe0bc840778 TYPE="ext4"  
PARTLABEL="Linux" PARTUUID=d55925ee-72c8-41e7-b514-7084e28f7334  
/dev/xvdf1: LABEL="old/" UUID=73947a77-ddbe-4dc7-bd8f-3fe0bc840778 TYPE="ext4"  
PARTLABEL="Linux" PARTUUID=d55925ee-72c8-41e7-b514-7084e28f7334
```

연결된 볼륨의 레이블을 변경하려면

1. e2label 명령을 사용하여 볼륨의 레이블을 /가 아닌 다른 레이블로 변경합니다.

```
[ec2-user ~]$ sudo e2label /dev/xvdf1 old/
```

2. 볼륨에 새 레이블이 지정되었는지 확인합니다.

```
[ec2-user ~]$ sudo e2label /dev/xvdf1  
old/
```

이렇게 변경한 후 인스턴스를 재부팅하면 인스턴스 부팅 시 첫 번째 ramdisk에서 올바른 볼륨을 선택하게 할 수 있습니다.

Important

볼륨에 연결한 새 레이블을 분리한 후 다른 인스턴스에 연결하여 루트 볼륨으로 사용하려면 위의 절차를 다시 수행하고 볼륨 레이블을 다시 원래 값으로 돌려야 합니다. 이렇게 하지 않으면 ramdisk가 / 레이블을 가진 볼륨을 찾을 수 없기 때문에 다른 인스턴스가 부팅되지 않습니다.

문서 기록

다음 표에서는 Amazon EC2 설명서에 대한 중요 추가 사항을 설명합니다. 사용자로부터 받은 의견을 수렴하기 위해 설명서가 자주 업데이트됩니다.

[Current API version: 2016-11-15.]

기능	API 버전	설명	릴리스 날짜
생성 시 태그 리소스	2016-11-15	생성 단계에서 인스턴스와 볼륨에 태그를 적용할 수 있습니다. 자세한 내용은 리소스에 태그 지정 (p. 681) 섹션을 참조하십시오. 또한 태그 기반 리소스 권한을 사용하여 적용되는 태그를 제어할 수도 있습니다. 자세한 내용은 태그 지정을 위한 리소스 수준 권한 (p. 429) 단원을 참조하십시오.	2017년 3월 28일
I3 인스턴스	2016-11-15	I3 인스턴스는 차세대 스토리지 최적화 인스턴스를 나타냅니다. 자세한 내용은 스토리지 최적화 인스턴스 (p. 158) 섹션을 참조하십시오.	2017년 2월 23일
연결된 EBS 볼륨을 수정 합니다.	2016-11-15	대부분의 EC2 인스턴스에 연결된 대다수 EBS의 경우, 볼륨을 분리하거나 인스턴스를 중지하지 않고도 볼륨 크기, 유형, IOPS를 수정할 수 있습니다. 자세한 내용은 Linux에서 EBS 볼륨의 크기, IOPS 또는 유형 수정 (p. 590) 섹션을 참조하십시오.	2017년 2월 13일
IAM 역할 연결	2016-11-15	기존 인스턴스에 대한 IAM 역할을 연결, 분리하거나 교체할 수 있습니다. 자세한 내용은 Amazon EC2의 IAM 역할 (p. 456) 섹션을 참조하십시오.	2017년 2월 9일
전용 스팟 인스턴스	2016-11-15	Virtual Private Cloud(VPC)의 단일 테넌트 하드웨어에서 스팟 인스턴스를 실행할 수 있습니다. 자세한 내용은 스팟 인스턴스의 테넌시 지정 (p. 214) 섹션을 참조하십시오.	2017년 1월 19일
IPv6 지원	2016-11-15	IPv6 CIDR를 VPC와 서브넷에 연결하고 IPv6 주소를 VPC의 인스턴스에 할당할 수 있습니다. 자세한 내용은 Amazon EC2인스턴스 IP 어드레싱 (p. 490) 섹션을 참조하십시오.	2016년 12월 1일
R4 인스턴스	2016-09-15	R4 인스턴스는 차세대 메모리 최적화 인스턴스를 나타냅니다. R4 인스턴스는 비즈니스 인텔리전스	2016년 11월 30일

기능	API 버전	설명	릴리스 날짜
		(BI), 데이터 마이닝 및 분석, 인 메모리 데이터베이스, 분산형 웹 규모 인 메모리 캐싱, 비정형 빅 데이터의 애플리케이션 성능 실시간 처리 등 메모리 집약적이고 지역 시간에 민감한 워크로드에 매우 적합합니다. 자세한 내용은 메모리 최적화 인스턴스 (p. 155) 섹션을 참조하십시오.	
새로운 t2.xlarge 및 t2.2xlarge 인스턴스 유형	2016-09-15	T2 인스턴스는 중간 정도의 기본 성능을 발휘하면서 작업의 필요에 따라 성능을 크게 높이는 버스트 기능을 제공하도록 설계되었습니다. 이러한 인스턴스는 경제적인 가격으로 제한된 시간 동안 빠른 응답 성과 뛰어난 성능이 필요한 애플리케이션용입니다. 자세한 내용은 T2 인스턴스 (p. 149) 섹션을 참조하십시오.	2016년 11월 30일
P2 인스턴스	2016-09-15	P2 인스턴스는 NVIDIA Tesla K80 GPU를 사용하며, CUDA 또는 OpenCL 프로그래밍 모델을 사용하는 일반 GPU 컴퓨팅에 맞게 설계되었습니다. 자세한 내용은 Linux 액셀러레이티드 컴퓨팅 인스턴스 (p. 162) 섹션을 참조하십시오.	2016년 9월 29일
m4.16xlarge 인스턴스	2016-04-01	64개의 vCPU와 256GiB RAM을 장착한 m4.16xlarge 인스턴스를 도입하여 일반 M4 제품군의 범위를 확장합니다.	2016년 9월 6일
스팟 집합의 자동 조정		이제 스팟 집합에 대한 조정 정책을 설정할 수 있습니다. 자세한 내용은 스팟 집합의 자동 조정 (p. 236) 섹션을 참조하십시오.	2016년 9월 1일
ENI(Elastic Network Adapter)	2016-04-01	이제 ENI를 사용하여 네트워킹 수준을 높일 수 있습니다. 자세한 내용은 향상된 네트워킹 유형 (p. 533) 섹션을 참조하십시오.	2016년 6월 28일
더 긴 ID 보기 및 수정을 위한 지원 기능 향상	2016-04-01	이제 다른 IAM 사용자, IAM 역할 또는 루트 사용자에 대한 더 긴 ID 설정을 보고 수정할 수 있습니다. 자세한 내용은 리소스 ID (p. 674) 섹션을 참조하십시오.	2016년 6월 23일
AWS 계정 간 암호화된 Amazon EBS 스냅샷 복사	2016-04-01	이제 AWS 계정 간에 암호화된 EBS 스냅샷을 복사할 수 있습니다. 자세한 내용은 Amazon EBS 스냅샷 복사 (p. 610) 섹션을 참조하십시오.	2016년 6월 21일
인스턴스 콘솔의 스크린샷 캡처	2015-10-01	이제 접속할 수 없는 인스턴스를 디버깅할 때 추가 정보를 얻을 수 있습니다. 자세한 내용은 연결할 수 없는 인스턴스의 스크린샷 캡처 (p. 730) 섹션을 참조하십시오.	2016년 24월 5일
X1 인스턴스	2015-10-01	인 메모리 데이터베이스, 빅 데이터 처리 엔진 및 HPC(고성능 컴퓨팅) 애플리케이션을 실행하기 위해 설계된 메모리 최적화 인스턴스입니다. 자세한 내용은 메모리 최적화 인스턴스 (p. 155) 섹션을 참조하십시오.	2016년 18월 5일

기능	API 버전	설명	릴리스 날짜
두 가지 새로운 EBS 볼륨 유형	2015-10-01	이제 처리량에 최적화된 HDD(st1) 및 콜드 HDD(sc1) 볼륨을 생성할 수 있습니다. 자세한 내용은 Amazon EBS 볼륨 유형 (p. 564) 섹션을 참조하십시오.	2016년 4 월 19일
Amazon EC2에 대한 새로운 NetworkPacketsIn 및 NetworkPacketsOut 측정치를 추가함.		Amazon EC2에 대한 새로운 NetworkPacketsIn 및 NetworkPacketsOut 측정치를 추가함. 자세한 내용은 인스턴스 측정치 (p. 349) 섹션을 참조하십시오.	2016년 3 월 23일
스팟 집합에 대한 CloudWatch 측정치		이제 스팟 집합에 대한 CloudWatch 측정치를 얻을 수 있습니다. 자세한 내용은 스팟 집합에 대한 CloudWatch 측정치 (p. 234) 섹션을 참조하십시오.	2016년 3 월 21일
예약된 인스턴스	2015-10-01	정기 예약 인스턴스(정기 인스턴스)를 사용하여 지정된 시작 시간과 기간에 따라 매일, 매주 또는 매월 반복적으로 용량 예약을 구입할 수 있습니다. 자세한 내용은 정기 예약 인스턴스 (p. 200) 섹션을 참조하십시오.	2016년 1 월 13일
더 긴 리소스 ID	2015-10-01	일부 Amazon EC2 및 Amazon EBS 리소스 유형에 좀더 긴 ID를 도입하고 있습니다. 옵트인 기간 동안 지원되는 리소스 유형에 더 긴 ID 형식을 사용할 수 있습니다. 자세한 내용은 리소스 ID (p. 674) 섹션을 참조하십시오.	2016년 1 월 13일
ClassicLink DNS 지원	2015-10-01	연결된 EC2-Classic 인스턴스와 VPC의 인스턴스 사이에서 처리되는 DNS 호스트 이름이 퍼블릭 IP 주소가 아니라 프라이빗 IP 주소로 확인되도록 VPC에 대해 ClassicLink DNS 지원을 활성화할 수 있습니다. 자세한 내용은 ClassicLink DNS 지원 활성화 (p. 477) 섹션을 참조하십시오.	2016년 1 월 11일
새 t2.nano 인스턴스 유형	2015-10-01	T2 인스턴스는 중간 정도의 기본 성능을 발휘하면서 작업의 필요에 따라 성능을 크게 높이는 버스트 기능을 제공하도록 설계되었습니다. 이러한 인스턴스는 경제적인 가격으로 제한된 시간 동안 빠른 응답 성과 뛰어난 성능이 필요한 애플리케이션용입니다. 자세한 내용은 T2 인스턴스 (p. 149) 섹션을 참조하십시오.	2015년 12 월 15일
전용 호스트	2015-10-01	Amazon EC2 전용 호스트는 고객 전용의 인스턴스 용량을 갖춘 물리적 서버입니다. 자세한 내용은 전용 호스트 (p. 246) 섹션을 참조하십시오.	2015년 11 월 23일
스팟 인스턴스 지속 시간	2015-10-01	이제 스팟 인스턴스의 지속 시간을 지정할 수 있습니다. 자세한 내용은 스팟 인스턴스의 지속 시간 지정 (p. 213) 섹션을 참조하십시오.	2015년 10 월 6일
스팟 집합 수정 요청	2015-10-01	이제 스팟 집합 요청의 목표 용량을 수정할 수 있습니다. 자세한 내용은 스팟 집합 요청 수정 (p. 225) 섹션을 참조하십시오.	2015년 9 월 29일

기능	API 버전	설명	릴리스 날짜
스팟 집합 다각화 할당 전략	2015-04-15	단일 스팟 집합 요청을 사용하여 여러 스팟 풀에서 스팟 인스턴스를 할당할 수 있습니다. 자세한 내용은 스팟 집합 할당 전략 (p. 208) 섹션을 참조하십시오.	2015년 9 월 15일
스팟 집합 인스턴스 가중치 부여	2015-04-15	각 인스턴스 유형이 애플리케이션의 성능에 기여하는 용량 단위를 정의하고, 그에 따라 적절히 각 스팟 풀에 대한 입찰 가격을 조정할 수 있습니다. 자세한 내용은 스팟 집합 인스턴스 가중치 부여 (p. 209) 섹션을 참조하십시오.	2015년 8 월 31일
새로운 재부팅 경보 작업과 경보 작업에 사용할 새로운 IAM 역할		재부팅 경보 작업과 경보 작업에 사용할 새로운 IAM 역할이 추가되었습니다. 자세한 내용은 인스턴스를 중지, 종료, 재부팅 또는 복구하는 경보 만들기 (p. 360) 섹션을 참조하십시오.	2015년 7 월 23일
새 t2.large 인스턴스 유형		T2 인스턴스는 중간 정도의 기본 성능을 발휘하면서 작업의 필요에 따라 성능을 크게 높이는 버스트 기능을 제공하도록 설계되었습니다. 이러한 인스턴스는 경제적인 가격으로 제한된 시간 동안 빠른 응답 성과 뛰어난 성능이 필요한 애플리케이션용입니다. 자세한 내용은 T2 인스턴스 (p. 149) 섹션을 참조하십시오.	2015년 6 월 16일
M4 인스턴스		컴퓨팅, 메모리 및 네트워크 리소스의 균형을 제공하는 차세대 범용 인스턴스입니다. M4 인스턴스는 AVX2가 포함된 사용자 지정 2.4GHz 인텔® Xeon® E5 2676v3(Haswell) 프로세서에 의해 구동됩니다.	2015년 6 월 11일
스팟 집합	2015-04-15	별도의 스팟 인스턴스 요청을 관리하는 대신 스팟 인스턴스의 모음 또는 집합을 관리할 수 있습니다. 자세한 내용은 스팟 집합의 작동 방식 (p. 208) 섹션을 참조하십시오.	2015년 5 월 18일
탄력적 IP 주소의 EC2-Classic 마이그레이션	2015-04-15	EC2-Classic 플랫폼에서 사용할 목적으로 할당한 탄력적 IP 주소는 EC2-VPC 플랫폼으로 마이그레이션할 수 있습니다. 자세한 내용은 EC2-Classic에서 EC2-VPC로 탄력적 IP 주소의 마이그레이션 (p. 507) 섹션을 참조하십시오.	2015년 5 월 15일
디스크가 여러 개 있는 VM을 AMI로 가져오기	2015-03-01	VM Import 프로세스는 이제 디스크가 여러 개 있는 VM을 AMI로 가져오기를 지원합니다. 자세한 내용은 VM Import/Export 사용 설명서에서 VM Import/Export를 사용하여 VM을 이미지로 가져오기 를 참조하십시오.	2015년 4 월 23일
새 g2.8xlarge 인스턴스 유형		새 g2.8xlarge 인스턴스는 4개의 고성능 NVIDIA GPU를 기반으로 제공되므로 대규모 렌더링, 트랜스코딩, 기계 학습 및 대량 병렬 처리 성능을 필요로 하는 기타 서버 측 워크로드를 비롯한 GPU 컴퓨팅 워크로드에 매우 적합합니다.	2015년 4 월 7일

기능	API 버전	설명	릴리스 날짜
D2 인스턴스		<p>직접 연결 인스턴스 스토리지에서 대량 데이터에 순차적으로 액세스해야 하는 애플리케이션에 최적화된 차세대 Amazon EC2 집약적 스토리지 인스턴스입니다. D2 인스턴스는 집약적 스토리지 제품군에 최적의 가격/성능을 제공하도록 설계되었습니다.</p> <p>2.4GHz 인텔® E5 2676v3(Haswell) 프로세서로 구동되는 D2 인스턴스는 보강된 컴퓨팅 능력, 증가된 메모리 및 향상된 네트워킹을 제공함으로써 HS1 인스턴스를 능가합니다. 또한 D2 인스턴스는 6TB, 12TB, 24TB 및 48TB 스토리지 옵션을 통해 네 가지 인스턴스 크기로 사용할 수 있습니다.</p> <p>자세한 내용은 스토리지 최적화 인스턴스 (p. 158) 섹션을 참조하십시오.</p>	2015년 3 월 24일
EC2 인스턴스용 자동 복구		<p>사용자는 Amazon EC2 인스턴스를 모니터링하고 기본 하드웨어 장애나 복구에 AWS 개입이 필요한 문제로 인해 인스턴스가 손상된 경우 인스턴스를 자동으로 복구하는 Amazon CloudWatch 경보를 만들 수 있습니다. 복구된 인스턴스는 인스턴스 ID, IP 주소 및 모든 인스턴스 메타데이터를 포함하여 원본 인스턴스와 동일합니다.</p> <p>자세한 내용은 인스턴스 복구 (p. 295) 섹션을 참조하십시오.</p>	2015년 1 월 12일
C4 인스턴스		<p>경제적인 가격으로 매우 우수한 성능을 제공하는 컴퓨팅에 최적화된 차세대 인스턴스입니다. C4 인스턴스는 사용자 지정 2.9GHz 인텔® E5-2666 v3(Haswell) 프로세서를 기반으로 합니다. 터보 부스트 기능이 추가된 C4 인스턴스의 프로세서 클럭 속도는 싱글/듀얼 코어의 터보 기능이 있는 3.5Ghz 만큼 빠릅니다. C3 컴퓨팅에 최적화된 인스턴스의 기능이 확장된 C4 인스턴스는 EC2 인스턴스 중에서 최고의 프로세서 성능을 고객에게 제공합니다. 이러한 인스턴스는 트래픽 양이 높은 웹 애플리케이션, 광고 서비스, 배치성 프로세스, 비디오 인코딩, 분산 분석, 고에너지 물리학, 계놈 분석 및 계산 유체 역학에 매우 적합합니다.</p> <p>자세한 내용은 컴퓨팅 최적화 인스턴스 (p. 152) 섹션을 참조하십시오.</p>	2015년 1 월 11일
ClassicLink	2014-10-01	<p>ClassicLink를 사용하면 EC2-Classic 인스턴스를 계정의 VPC에 연결할 수 있습니다. VPC 보안 그룹을 EC2-Classic 인스턴스에 연결할 수 있으므로 EC2-Classic 인스턴스와 VPC의 인스턴스가 프라이빗 IP 주소를 사용하여 서로 통신할 수 있습니다. 자세한 내용은 ClassicLink (p. 472) 섹션을 참조하십시오.</p>	2015년 1 월 7일

기능	API 버전	설명	릴리스 날짜
스팟 인스턴스 종료 공지		<p>스팟 인스턴스 종단으로부터 보호하는 가장 좋은 방법은 애플리케이션을 내결함성 있게 설계하는 것입니다. 또한 Amazon EC2가 스팟 인스턴스를 종료하기 2분 전에 경고하는 스팟 인스턴스 종료 공지를 이용할 수 있습니다.</p> <p>자세한 내용은 스팟 인스턴스 종료 공지 (p. 243) 섹션을 참조하십시오.</p>	2015년 1 월 5일
DescribeVolumes 페이지 매김 지원	2014-09-01	<p>이제 <code>DescribeVolumes</code> API 호출에서는 <code>MaxResults</code> 및 <code>NextToken</code> 파라미터를 사용한 결과의 페이지 매김을 지원합니다. 자세한 내용은 Amazon EC2 API Reference의 DescribeVolumes 섹션을 참조하십시오.</p>	2014년 10 월 23일
T2 인스턴스	2014-06-15	<p>T2 인스턴스는 중간 정도의 기본 성능을 발휘하면서 작업의 필요에 따라 성능을 크게 높이는 버스트 기능을 제공하도록 설계되었습니다. 이러한 인스턴스는 경제적인 가격으로 제한된 시간 동안 빠른 응답 성과 뛰어난 성능이 필요한 애플리케이션용입니다. 자세한 내용은 T2 인스턴스 (p. 149) 섹션을 참조하십시오.</p>	2014년 6 월 30일
새 [EC2 Service Limits] 페이지		<p>Amazon EC2 콘솔의 [EC2 Service Limits] 페이지에서는 리전별로 Amazon EC2 및 Amazon VPC에서 제공하는 리소스의 현재 제한을 볼 수 있습니다.</p>	2014년 6 월 19일
Amazon EBS 범용 SSD 볼륨	2014-05-01	<p>범용 SSD 볼륨은 광범위한 작업에서 이상적으로 사용될 수 있는 비용 효과적인 스토리지를 제공합니다. 이러한 볼륨은 시간을 연장할 경우 3 IOPS의 버스트 기능까지 지원되어 지연 시간이 한 자릿수 밀리초에 불과하며 3,000 IOPS/GiB를 기본 성능으로 제공합니다. 범용 SSD 볼륨 크기는 1GiB~1TiB입니다. 자세한 내용은 범용 SSD(gp2) 볼륨 (p. 566) 섹션을 참조하십시오.</p>	2014년 6 월 16일
Amazon EBS 암호화	2014-05-01	<p>Amazon EBS 암호화에서는 EBS 데이터 볼륨 및 스냅샷에 대한 완벽한 암호를 제공하므로 보안 키 관리 인프라를 구축하고 유지 관리할 필요가 없습니다. EBS 암호화는 Amazon 관리 키를 사용하여 데이터를 암호화하여 상주 데이터에 대한 보안을 활성화합니다. EC2 인스턴스를 호스트하는 서버에서 암호화가 이루어지기 때문에 EC2 인스턴스와 EBS 스토리지 간 이동하는 데이터도 암호화됩니다. 자세한 내용은 Amazon EBS Encryption (p. 617) 섹션을 참조하십시오.</p>	2014년 5 월 21일

기능	API 버전	설명	릴리스 날짜
R3 인스턴스	2014-02-01	<p>경제적인 가격의 GiB RAM이 장착된 고성능 차세대 메모리 최적화 인스턴스. 이러한 인스턴스는 R3 인스턴스의 어드밴스 네트워킹 기능, 우수한 컴퓨팅 성능 및 vCPU당 높은 메모리를 활용할 수 있는 관계형 및 NoSQL 데이터베이스, 메모리 내 분석 솔루션, 공학 계산 및 기타 메모리 집약형 애플리케이션에 매우 적합합니다.</p> <p>Amazon EC2 인스턴스 유형별 하드웨어 사양에 대한 자세한 내용은 Amazon EC2 인스턴스를 참조하십시오.</p>	2014년 4 월 9일
새 Amazon Linux AMI 릴리스		Amazon Linux AMI 2014.03이 릴리스되었습니다.	2014년 3 월 27일
Amazon EC2 사용 보고서		Amazon EC2 사용 보고서는 EC2 사용에 대한 비용 및 사용량 데이터를 보여 주는 보고서 세트입니다. 자세한 내용은 Amazon EC2 사용 보고서 (p. 689) 섹션을 참조하십시오.	2014년 1 월 28일
추가 M3 인스턴스	2013-10-15	이제 M3 인스턴스 크기 <code>m3.medium</code> 및 <code>m3.large</code> 가 지원됩니다. Amazon EC2 인스턴스 유형별 하드웨어 사양에 대한 자세한 내용은 Amazon EC2 인스턴스 를 참조하십시오.	2014년 1 월 20일
I2 인스턴스	2013-10-15	이러한 인스턴스는 매우 높은 IOPS를 제공하며 연속 SSD 쓰기 성능 향상을 위해 Linux 인스턴스에서 TRIM을 지원합니다. 또한 I2 인스턴스는 인스턴스 간 지연 시간과 네트워크 지터를 낮추고 PPS(Packet Per Second) 성능을 크게 높이는 향상된 네트워킹 기능을 지원합니다. 자세한 내용은 스토리지 최적화 인스턴스 (p. 158) 섹션을 참조하십시오.	2013년 12 월 19일
M3 인스턴스 업데이트	2013-10-15	M3 인스턴스 크기 <code>m3.xlarge</code> 및 <code>m3.2xlarge</code> 에서는 이제 SSD 볼륨이 장착된 인스턴스 스토어를 지원합니다.	2013년 12 월 19일
Linux 가상 머신 가져오기	2013-10-15	VM Import 프로세스에서 이제 Linux 인스턴스 가져오기를 지원합니다. 자세한 내용은 VM Import/Export 사용 설명서 를 참조하십시오.	2013년 12 월 16일
RunInstances에 대한 리소스 수준의 권한	2013-10-15	이제 AWS Identity and Access Management에서 정책을 생성하여 Amazon EC2 RunInstances API 작업에 대한 리소스 수준의 권한을 제어할 수 있습니다. 자세한 내용과 정책 예는 Amazon EC2 리소스에 대한 액세스 제어 (p. 398) 섹션을 참조하십시오.	2013년 11 월 20일

기능	API 버전	설명	릴리스 날짜
C3 인스턴스	2013-10-15	<p>경제적인 가격으로 매우 우수한 성능을 제공하는 컴퓨팅에 최적화된 인스턴스입니다. 또한 C3 인스턴스는 인스턴스 간 지연 시간과 네트워크 지터를 낮추고 PPS(Packet Per Second) 성능을 크게 높여주는 향상된 네트워킹 기능을 지원합니다. 이러한 인스턴스는 트래픽 양이 높은 웹 애플리케이션, 광고 서비스, 배치성 프로세스, 비디오 인코딩, 분산 분석, 고에너지 물리학, 게임 분석 및 계산 유체 역학에 매우 적합합니다.</p> <p>Amazon EC2 인스턴스 유형별 하드웨어 사양에 대한 자세한 내용은 Amazon EC2 인스턴스를 참조하십시오.</p>	2013년 11월 14일
AWS Marketplace에서 인스턴스 시작		<p>이제 Amazon EC2 시작 마법사를 사용하여 AWS Marketplace에서 인스턴스를 시작할 수 있습니다. 자세한 내용은 AWS Marketplace 인스턴스 시작 (p. 271) 섹션을 참조하십시오.</p>	2013년 11월 11일
G2 인스턴스	2013-10-01	<p>이러한 인스턴스는 엄청난 병렬 처리 능력을 필요로 하는 비디오 제작 서비스, 3D 가상화, 스트리밍 그래픽 집약형 애플리케이션 및 기타 서버 쪽 작업에 매우 적합합니다. 자세한 내용은 Linux 액셀러레이티드 컴퓨팅 인스턴스 (p. 162) 섹션을 참조하십시오.</p>	2013년 11월 4일
새로운 시작 마법사		<p>재설계된 새로운 EC2 시작 마법사가 제공됩니다. 자세한 내용은 인스턴스 시작하기 (p. 265) 섹션을 참조하십시오.</p>	2013년 10월 10일
Amazon EC2 예약 인스턴스의 인스턴스 유형 수정	2013-10-01	<p>이제 동일 패밀리(예: M1, M2, M3, C1) 내에서 Linux 예약 인스턴스의 인스턴스 유형을 수정할 수 있습니다. 자세한 내용은 표준 예약 인스턴스 변경 (p. 193) 섹션을 참조하십시오.</p>	2013년 10월 9일
새 Amazon Linux AMI 릴리스		<p>Amazon Linux AMI 2013.09가 릴리스되었습니다.</p>	2013년 9월 30일
Amazon EC2 예약 인스턴스 수정	2013-08-15	<p>이제 리전에서 예약 인스턴스를 수정할 수 있습니다. 자세한 내용은 표준 예약 인스턴스 변경 (p. 193) 섹션을 참조하십시오.</p>	2013년 9월 11일
퍼블릭 IP 주소 배정	2013-07-15	<p>이제는 VPC에서 인스턴스를 시작할 때 퍼블릭 IP 주소를 배정할 수 있습니다. 자세한 내용은 인스턴스 시작 시 퍼블릭 IPv4 주소 배정 (p. 495) 섹션을 참조하십시오.</p>	2013년 8월 20일
리소스 수준의 권한 부여	2013-06-15	<p>Amazon EC2에서는 새로운 Amazon 리소스 이름(ARN)과 조건 키를 지원합니다. 자세한 내용은 Amazon EC2에 대한 IAM 정책 (p. 401) 섹션을 참조하십시오.</p>	2013년 7월 8일
증분형 스냅샷 사본	2013-02-01	<p>이제 증분형 스냅샷 사본을 사용할 수 있습니다. 자세한 내용은 Amazon EBS 스냅샷 복사 (p. 610) 섹션을 참조하십시오.</p>	2013년 6월 11일

기능	API 버전	설명	릴리스 날짜
새 [Tags] 페이지		Amazon EC2 콘솔에 새 [Tags] 페이지가 있습니다. 자세한 내용은 Amazon EC2 리소스에 태그 지정 (p. 681) 섹션을 참조하십시오.	2013년 4 월 4일
새 Amazon Linux AMI 릴리스		Amazon Linux AMI 2013.03이 릴리스되었습니다.	2013년 3 월 27일
추가 EBS에 최적화된 인스턴스 유형	2013-02-01	<p>이제 인스턴스 유형 <code>c1.xlarge</code>, <code>m2.2xlarge</code>, <code>m3.xlarge</code> 및 <code>m3.2xlarge</code>를 EBS에 최적화된 인스턴스로 시작할 수 있습니다.</p> <p>자세한 내용은 Amazon EBS 최적화 인스턴스 (p. 614) 섹션을 참조하십시오.</p>	2013년 3 월 19일
리전 간 AMI 복사	2013-02-01	<p>리전 간에 AMI를 복사하여 다른 이상의 AWS 리전에서 일관된 인스턴스를 빠르고 쉽게 시작할 수 있습니다.</p> <p>자세한 내용은 AMI 복사 (p. 126) 섹션을 참조하십시오.</p>	2013년 3 월 11일
인스턴스를 기본 VPC로 시작	2013-02-01	<p>AWS 계정은 인스턴스를 EC2-Classic 또는 EC2-VPC 플랫폼으로 시작하거나, EC2-VPC 플랫폼으로만 시작하거나, 리전별로 시작할 수도 있습니다. EC2-VPC로만 인스턴스를 시작할 수 있는 경우 사용자를 위한 기본 VPC가 생성됩니다. 사용자가 기본이 아닌 VPC를 직접 생성하여 인스턴스 시작 시 지정한 경우가 아니면 인스턴스 시작 시 해당 인스턴스가 기본 VPC로 시작됩니다.</p> <p>자세한 내용은 지원되는 플랫폼 (p. 471) 섹션을 참조하십시오.</p>	2013년 3 월 11일
고용량 메모리 클러스터 (cr1.8xlarge) 인스턴스 유형	2012-12-01	대용량 메모리가 고성능 CPU 및 네트워크 성능과 결합되었습니다. 이러한 인스턴스는 메모리 내 분석, 그래프 분석 및 공학 컴퓨팅 애플리케이션에 매우 적합합니다.	2013년 1 월 21일
고용량 스토리지 (hs1.8xlarge) 인스턴스 유형	2012-12-01	고용량 스토리지 인스턴스는 인스턴스당 매우 높은 스토리지 밀도와 높은 순차 읽기/쓰기 성능을 제공합니다. 이러한 인스턴스는 데이터 웨어하우스, 하둡/MapReduce 및 병렬 파일 시스템에 매우 적합합니다.	2012년 12 월 20일
EBS 스냅샷 복사	2012-12-01	스냅샷 사본으로 데이터 백업, 새 Amazon EBS 볼륨 또는 Amazon 머신 이미지(AMI)를 생성할 수 있습니다. 자세한 내용은 Amazon EBS 스냅샷 복사 (p. 610) 섹션을 참조하십시오.	2012년 12 월 17일
프로비저닝된 IOPS SSD 볼륨에 대한 EBS 지표 및 상태 확인 업데이트	2012-10-01	EBS 지표를 업데이트하여 프로비저닝된 IOPS SSD 볼륨용 새 지표 두 개를 포함했습니다. 자세한 내용은 CloudWatch로 볼륨 모니터링 (p. 580) 을 참조하십시오. 또한 프로비저닝된 IOPS SSD 볼륨에 대한 새로운 상태 확인도 추가했습니다. 자세한 내용은 상태 확인으로 볼륨 모니터링 (p. 583) 섹션을 참조하십시오.	2012년 11 월 20일

기능	API 버전	설명	릴리스 날짜
Linux 커널		AKI ID를 업데이트하고, 배포 커널을 재구성했으며, PVOps 섹션을 업데이트했습니다.	2012년 11 월 13일
M3 인스턴스	2012-10-01	새로운 M3 extra-large 및 M3 double-extra-large 인스턴스 유형을 추가했습니다. Amazon EC2 인스턴스 유형별 하드웨어 사양에 대한 자세한 내용은 Amazon EC2 인스턴스 를 참조하십시오.	2012년 10 월 31일
스팟 인스턴스 요청 상태	2012-10-01	스팟 인스턴스 요청 상태를 사용하면 스팟 요청의 상태를 쉽게 확인할 수 있습니다.	2012년 10 월 14일
새 Amazon Linux AMI 릴리스		Amazon Linux AMI 2012.09가 릴리스되었습니다.	2012년 10 월 11일
Amazon EC2 예약 인스턴스 마켓플레이스	2012-08-15	예약 인스턴스 마켓플레이스는 더 이상 필요하지 않은 Amazon EC2 예약 인스턴스를 보유한 판매자와 추가 용량을 원하는 구매자를 연결합니다. 예약 인스턴스 마켓플레이스를 통해 구매 및 판매되는 예약 인스턴스는 다른 예약 인스턴스와 동일하게 작동합니다. 단, 이러한 인스턴스는 스탠다드 약정보다 사용 기간이 짧을 수 있으며 다른 가격으로 판매할 수 있습니다.	2012년 9 월 11일
Amazon EBS용 프로비저닝된 IOPS SSD	2012-07-20	프로비저닝된 IOPS SSD 볼륨은 일관적이고 빠른 응답 시간을 이용하는 데이터베이스 애플리케이션처럼 I/O 집약적 작업을 위한 예측 가능하고 우수한 성능을 제공합니다. 자세한 내용은 Amazon EBS 볼륨 유형 (p. 564) 섹션을 참조하십시오.	2012년 7 월 31일
Amazon EC2용 고성능 I/O 인스턴스	2012-06-15	고성능 I/O 인스턴스는 SSD 기반 로컬 인스턴스 스토리지를 사용하여 지연 시간을 줄이고 디스크 I/O 성능을 높입니다.	2012년 7 월 18일
Amazon EC2 인스턴스의 IAM 역할	2012-06-01	Amazon EC2의 IAM 역할은 다음을 제공합니다.	2012년 6 월 11일
		<ul style="list-style-type: none"> • Amazon EC2 인스턴스에서 실행 중인 애플리케이션의 AWS 액세스 키 • Amazon EC2 인스턴스에서 AWS 액세스 키 자동 순환 • Amazon EC2 인스턴스에서 실행 중이며 AWS 서비스에 요청하는 애플리케이션에 대한 세분화된 사용 권한 	

기능	API 버전	설명	릴리스 날짜
더 쉽게 시작하고 종단 가능성을 처리할 수 있게 하는 스팟 인스턴스 기능		<p>이제 다음과 같이 스팟 인스턴스를 관리할 수 있습니다.</p> <ul style="list-style-type: none"> Auto Scaling 시작 구성을 사용하여 스팟 인스턴스에 입찰하고 스팟 인스턴스 입찰 계획을 설정합니다. 자세한 내용은 Auto Scaling 사용 설명서의 Launching Spot Instances in Your Auto Scaling Group을 참조하십시오. 인스턴스가 시작되거나 종료될 때 알림을 받습니다. AWS CloudFormation 템플릿을 사용하여 AWS 리소스를 포함하는 스택에서 스팟 인스턴스를 시작합니다. 	2012년 6 월 7일
EC2 인스턴스 내보내기와 Amazon EC2 상태 확인을 위한 타임스탬프	2012-05-01	<p>원래 EC2로 가져왔던 Windows Server 인스턴스에 대한 내보내기 지원을 추가했습니다.</p> <p>상태 확인이 실패한 날짜 및 시간을 나타내는 인스턴스 상태 및 시스템 상태의 타임스탬프에 대한 지원을 추가했습니다.</p>	2012년 5 월 25일
EC2 인스턴스 내보내기와, Amazon VPC에 대한 인스턴스 및 시스템 상태 확인 시 타임스탬프	2012-05-01	<p>Citrix Xen, Microsoft Hyper-V 및 VMware vSphere로 EC2 인스턴스 내보내기 지원을 추가했습니다.</p> <p>인스턴스 및 시스템 상태 확인 시 타임스탬프 지원을 추가했습니다.</p>	2012년 5 월 25일
클러스터 컴퓨팅 에이트 엑스트라 라지 인스턴스	2012-04-01	VPC에서 cc2.8xlarge 인스턴스에 대한 지원을 추가했습니다.	2012년 4 월 26일
AWS Marketplace AMI	2012-04-01	AWS Marketplace AMI 지원을 추가했습니다.	2012년 4 월 19일
새 Linux AMI 릴리스		Amazon Linux AMI 2012.03이 릴리스되었습니다.	2012년 3 월 28일
새 AKI 버전		AWS GovCloud (US) 리전용 AKI 버전 1.03 및 AKI 를 릴리스했습니다.	2012년 3 월 28일
미디엄 인스턴스, 모든 AMI의 64비트 및 Java 기반 SSH 클라이언트에 대한 지원	2011-12-15	새로운 인스턴스 유형에 대한 지원과 64비트 정보를 추가했습니다. Java 기반 SSH 클라이언트를 사용하여 Linux 인스턴스에 연결하기 위한 절차를 추가했습니다.	2012년 3 월 7일
예약된 인스턴스 요금 계층	2011-12-15	예약 인스턴스 요금 계층에 기본 제공되는 할인 요금을 활용하는 방법을 설명하는 새로운 섹션을 추가했습니다.	2012년 3 월 5일
Amazon Virtual Private Cloud의 EC2 인스턴스 용 ENI	2011-12-01	VPC의 EC2 인스턴스용 ENI(탄력적 네트워크 인터페이스)에 대한 새로운 섹션을 추가했습니다. 자세한 내용은 탄력적 네트워크 인터페이스 (p. 512) 섹션을 참조하십시오.	2011년 12 월 21일

기능	API 버전	설명	릴리스 날짜
새 GRU 리전 및 AKI		SA-East-1 리전용 새 AKI 릴리스에 대한 정보를 추가했습니다. 이번 릴리스에서는 AKI 버전 1.01이 더 이상 사용되지 않습니다. AKI 버전 1.02가 이전 버전과 계속 호환됩니다.	2011년 12 월 14일
Amazon EC2 예약 인스턴스를 위한 새로운 제공 유형	2011-11-01	예상되는 인스턴스 사용을 처리하는 다양한 예약 인스턴스 상품 중에서 선택할 수 있습니다.	2011년 12 월 1일
Amazon EC2 인스턴스 상태	2011-11-01	인스턴스에 영향을 줄 수 있는 AWS에서 계획한 예약 이벤트를 포함하여 인스턴스의 상태에 대한 추가 세부 정보를 볼 수 있습니다. 이러한 운영 활동에는 보안 패치나 소프트웨어 업데이트를 적용하는 데 필요한 인스턴스 재부팅이나 하드웨어 문제가 있는 경우에 필요한 인스턴스 중지가 포함됩니다. 자세한 내용은 인스턴스 상태 모니터링 (p. 339) 섹션을 참조하십시오.	2011년 11 월 16일
Amazon EC2 클러스터 컴퓨팅 인스턴스 유형		Amazon EC2에 클러스터 컴퓨팅 에이트 엑스트라지(cc2.8xlarge) 지원을 추가했습니다.	2011년 11 월 14일
새 PDX 리전 및 AKI		새 US-West 2 리전용 새 AKI 릴리스에 대한 정보를 추가했습니다.	2011년 11 월 8일
Amazon VPC의 스팟 인스턴스	2011-07-15	Amazon VPC의 스팟 인스턴스 지원에 대한 정보를 추가했습니다. 이 업데이트로 Virtual Private Cloud(VPC)에서 스팟 인스턴스를 시작할 수 있습니다. 스팟 인스턴스의 사용자는 VPC에서 스팟 인스턴스를 시작하면 Amazon VPC의 이점을 누릴 수 있습니다.	2011년 10 월 11일
새 Linux AMI 릴리스		Amazon Linux AMI 2011.09 릴리스에 대한 정보를 추가했습니다. 이 업데이트에서는 Amazon Linux AMI에서 베타 태그를 제거하고 특정 버전에 대해 리포지토리를 잠그는 기능을 지원하며 보안 업데이트를 포함하여 설치 패키지에 대한 사용 가능한 업데이트가 있을 때 알림을 제공합니다.	2011년 9 월 26일
CLI 도구 사용자를 위한 간소화된 VM Import 프로세스	2011-07-15	이제 <code>ImportInstance</code> 및 <code>ImportVolume</code> 기능의 향상으로 VM Import 프로세스가 간소화되어 가져오기 작업을 생성한 후 이미지가 Amazon EC2로 업로드됩니다. 또한 <code>ResumeImport</code> 가 도입되면서 사용자가 완료되지 않은 업로드를 작업이 중지된 시점부터 다시 시작할 수 있습니다.	2011년 9 월 15일
VHD 파일 형식으로 가져오기 지원		이제 VM Import에서 가상 머신 이미지 파일을 VHD 형식으로 가져올 수 있습니다. VHD 파일 형식은 Citrix Xen 및 Microsoft Hyper-V 가상화 플랫폼과 호환됩니다. 이번 릴리스에 포함된 VM Import 기능에서는 이제 RAW, VHD 및 VMDK(VMware ESX 호환) 이미지 형식을 지원합니다. 자세한 내용은 VM Import/Export 사용 설명서 를 참조하십시오.	2011년 8 월 24일

기능	API 버전	설명	릴리스 날짜
VMware vCenter용 Amazon EC2 VM Import 커넥터 업데이트		VMware vCenter 가상 어플라이언스용 Amazon EC2 VM Import 커넥터 버전 1.1(커넥터)에 대한 정보를 추가했습니다. 이 업데이트에는 인터넷 액세스에 대한 프록시 지원, 오류 처리 개선, 작업 진행률 표시줄 정확도 향상 및 여러 버그 수정 사항이 포함되어 있습니다.	2011년 6 월 27일
Linux AMI에서 사용자 제공 커널 실행 지원		1.01에서 1.02로의 AKI 버전 변경에 대한 정보를 추가했습니다. 이 버전에서는 PVGRUB를 업데이트하여 t1.micro Linux 인스턴스와 연결된 시작 오류를 해결했습니다. 자세한 내용은 사용자 제공 커널 (p. 139) 섹션을 참조하십시오.	2011년 6 월 20일
스팟 인스턴스 가용 영역 요금 변경	2011-05-15	스팟 인스턴스 가용 영역 요금 변경에 대한 정보를 추가했습니다. 이 릴리스에서는 스팟 인스턴스 요청과 스팟 가격 기록을 쿼리할 때 반환되는 정보의 일부로서 새 가용 영역 요금 옵션을 추가했습니다. 이러한 추가를 통해 스팟 인스턴스를 특정 가용 영역으로 시작하는 데 필요한 가격을 보다 쉽게 확인할 수 있습니다.	2011년 5 월 26일
AWS Identity and Access Management		AWS Identity and Access Management(IAM)에 대한 정보를 추가했습니다. 사용자는 IAM을 통해 일반적으로 Amazon EC2 리소스와 함께 사용할 수 있는 Amazon EC2 작업을 지정할 수 있습니다. 자세한 내용은 Amazon EC2 리소스에 대한 액세스 제어 (p. 398) 섹션을 참조하십시오.	2011년 4 월 26일
Linux AMI에서 사용자 제공 커널 실행 지원		Linux AMI에서 PVGRUB Amazon Kernel Image(AKI)를 사용한 사용자 제공 커널 실행 지원에 대한 정보를 추가했습니다. 자세한 내용은 사용자 제공 커널 (p. 139) 섹션을 참조하십시오.	2011년 4 월 26일
전용 인스턴스		Amazon Virtual Private Cloud(Amazon VPC) 내에서 시작되는 전용 인스턴스는 호스트 하드웨어 수준에서 물리적으로 구분되어 있는 인스턴스입니다. 전용 인스턴스에서는 탄력적인 온디맨드 프로비저닝을 포함한 다양한 혜택과 함께 Amazon VPC와 AWS 클라우드를 활용하고 사용하는 서비스에 대해서만 요금을 지불할 수 있으며, 하드웨어 수준에서 Amazon EC2 컴퓨팅 인스턴스를 구분할 수 있습니다. 자세한 내용은 전용 인스턴스 (p. 257) 섹션을 참조하십시오.	2011년 3 월 27일
AWS Management Console에 예약 인스턴스 업데이트		AWS Management Console 업데이트로 사용자는 더욱 더 쉽게 추가 예약 인스턴스를 보고 전용 예약 인스턴스를 비롯한 추가 예약 인스턴스를 구매할 수 있습니다. 자세한 내용은 예약 인스턴스 (p. 174) 섹션을 참조하십시오.	2011년 3 월 27일

기능	API 버전	설명	릴리스 날짜
새 Amazon Linux 참조 AMI		새 Amazon Linux 참조 AMI가 CentOS 참조 AMI를 대체합니다. CentOS 5.4 AMI에서 클러스터 인스턴스의 클럭 드리프트 해결이라는 섹션을 포함하여 CentOS 참조 AMI에 대한 정보를 삭제했습니다. 자세한 내용은 액셀러레이티드 컴퓨팅 인스턴스용 AMI (p. 163) 섹션을 참조하십시오.	2011년 3 월 15일
메타데이터 정보	2011-01-01	2011년 1월 1일 릴리스의 변경 내용을 반영하여 메타데이터에 대한 정보를 추가했습니다. 자세한 내용은 인스턴스 메타데이터 및 사용자 데이터 (p. 321) 및 인스턴스 메타데이터 카테고리 (p. 328) 섹션을 참조하십시오.	2011년 3 월 11일
VMware vCenter용 Amazon EC2 VM Import 커넥터		VMware vCenter 가상 어플라이언스용 Amazon EC2 VM Import 커넥터(커넥터)에 대한 정보를 추가했습니다. 이 커넥터는 VMware vSphere Client가 통합된 VMware vCenter용 플러그 인으로 VMware 가상 머신을 Amazon EC2로 가져오는 데 사용할 수 있는 그래픽 사용자 인터페이스를 제공합니다.	2011년 3 월 3일
강제 볼륨 분리		이제 AWS Management Console을 사용하여 Amazon EBS 볼륨을 인스턴스에서 강제 분리할 수 있습니다. 자세한 내용은 인스턴스에서 Amazon EBS 볼륨 분리 (p. 588) 섹션을 참조하십시오.	2011년 2 월 23일
인스턴스 종료 방지		이제 AWS Management Console을 사용하여 인스턴스가 종료되는 것을 방지할 수 있습니다. 자세한 내용은 인스턴스에 대한 종료 방지 기능 활성화 (p. 292) 섹션을 참조하십시오.	2011년 2 월 23일
CentOS 5.4 AMI에서 클러스터 인스턴스의 클럭 드리프트 해결		Amazon의 CentOS 5.4 AMI에서 실행 중인 클러스터 인스턴스의 클럭 드리프트 문제를 해결하는 방법에 대한 정보를 추가했습니다.	2011년 1 월 25일
VM Import	2010-11-15	VM Import에 대한 정보를 추가했습니다. VM Import 기능을 사용하면 가상 머신이나 볼륨을 Amazon EC2로 가져올 수 있습니다. 자세한 내용은 VM Import/Export 사용 설명서 를 참조하십시오.	2010년 12 월 15일
인스턴스 기본 모니터링	2010-08-31	EC2 인스턴스 기본 모니터링에 대한 정보를 추가했습니다.	2010년 12 월 12일
클러스터 GPU 인스턴스	2010-08-31	Amazon EC2는 고성능 컴퓨팅(HPC) 애플리케이션을 위한 클러스터 GPU 인스턴스(cg1.4xlarge)를 제공합니다. Amazon EC2 인스턴스 유형별 하드웨어 사양에 대한 자세한 내용은 Amazon EC2 인스턴스 를 참조하십시오.	2010년 11 월 14일
필터와 태그	2010-08-31	리소스 목록, 필터링 및 태그에 대한 정보를 추가했습니다. 자세한 내용은 리소스 목록화 및 필터링 (p. 678) 및 Amazon EC2 리소스에 태그 지정 (p. 681) 섹션을 참조하십시오.	2010년 9 월 19일

기능	API 버전	설명	릴리스 날짜
멱등성 인스턴스 시작	2010-08-31	인스턴스 실행 시 멱등성 유지에 대한 정보를 추가했습니다. 자세한 내용은 Amazon EC2 API Reference의 Ensuring Idempotency 섹션을 참조하십시오.	2010년 9 월 19일
마이크로 인스턴스	2010-06-15	Amazon EC2는 특정 애플리케이션 유형을 위한 t1.micro 인스턴스 유형을 제공합니다. 자세한 내용은 T1 마이크로 인스턴스 (p. 166) 섹션을 참조하십시오.	2010년 9 월 8일
Amazon EC2용 AWS Identity and Access Management		Amazon EC2가 이제 AWS Identity and Access Management(IAM)와 통합되었습니다. 자세한 내용은 Amazon EC2 리소스에 대한 액세스 제어 (p. 398) 섹션을 참조하십시오.	2010년 9 월 2일
클러스터 인스턴스	2010-06-15	Amazon EC2는 고성능 컴퓨팅(HPC) 애플리케이션을 위한 클러스터 컴퓨팅 인스턴스를 제공합니다. Amazon EC2 인스턴스 유형별 하드웨어 사양에 대한 자세한 내용은 Amazon EC2 인스턴스 를 참조하십시오.	2010년 7 월 12일
Amazon VPC IP 주소 지정	2010-06-15	Amazon VPC 사용자는 이제 IP 주소를 지정하여 VPC에서 시작된 인스턴스를 배정할 수 있습니다.	2010년 7 월 12일
Amazon EBS 볼륨용 Amazon CloudWatch 모니터링		이제 Amazon EBS 볼륨에 대해 Amazon CloudWatch 모니터링을 자동으로 사용할 수 있습니다. 자세한 내용은 CloudWatch로 볼륨 모니터링 (p. 580) 섹션을 참조하십시오.	2010년 6 월 14일
고용량 메모리 엑스트라 라지 인스턴스	2009-11-30	이제 Amazon EC2에서는 고용량 메모리 엑스트라 라지 (m2.xlarge) 인스턴스 유형을 지원합니다. Amazon EC2 인스턴스 유형별 하드웨어 사양에 대한 자세한 내용은 Amazon EC2 인스턴스 를 참조하십시오.	2010년 2 월 22일

AWS Glossary

For the latest AWS terminology, see the [AWS Glossary](#) in the AWS General Reference.