Amazon Elastic Compute Cloud

Linux 인스턴스용 사용 설명서

Amazon Elastic Compute Cloud: Linux 인스턴스용 사용 설명서

Copyright © 2020 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 1
관련 서비스 2
Amazon EC2에 액세스 3
Amazon EC2 가격 3
PCI DSS 준수 4
인스턴스 및 AMI 4
인스턴스 5
AMI 6
리전, 가용 영역 및 로컬 영역 6
리전, 가용 영역 및 로컬 영역 개념 7
사용 가능한 리전 9
리전 및 Endpoint 10
리전, 가용 영역 및 로컬 영역 설명 10
리소스에 대한 리전 지정 12
로컬 영역 활성화 14
로컬 영역 비활성화 15
가용 영역 또는 로컬 영역에서 인스턴스 시작 15
다른 가용 영역으로 인스턴스 마이그레이션 15
루트 디바이스 볼륨 16
루트 디바이스 스토리지 개념 16
루트 디바이스 유형에 따른 AMI 선택 18
인스턴스의 루트 디바이스 유형 확인 18
루트 디바이스 볼륨 지속 설정 19
설정 22
AWS에 가입 22
키 페어 생성 22
보안 그룹 생성 24
시작하기 27
개요 27
사전 조건 28
1단계: 인스턴스 시작 28
2단계: 인스턴스에 연결 29
3단계: 인스턴스 정리 29
다음 단계 29
모범 사례 31
자습서 32
LAMP 서버 설치(Amazon Linux 2) 32
1단계: LAMP 서버 준비 32
2단계: LAMP 서버 테스트 36
3단계: 데이터베이스 서버 보안 설정 37
4단계: (선택 사항) phpMyAdmin 설치 38
문제 해결 41
관련 주제 41
LAMP 서버 설치(Amazon Linux AMI) 41
1단계: LAMP 서버 준비 42
2단계: LAMP 서버 테스트 46
3단계: 데이터베이스 서버 보안 설정 47
4단계: (선택 사항) phpMyAdmin 설치 48
문제 해결 51
관련 주제 52
자습서: WordPress 블로그 호스팅 52
사전 조건 53

WordPress 설치 ... 53
다음 단계 ... 59
도움말! 내 퍼블릭 DNS 이름이 변경되어 블로그를 사용할 수 없습니다. ... 59
자습서: Amazon Linux 2에서 SSL/TLS 구성 ... 60
사전 조건 ... 61
1단계: 서버에서 TLS 활성화 ... 61
2단계: CA가 서명한 인증서 가져오기 ... 63
3단계: 보안 구성 테스트 및 강화 ... 68
문제 해결 ... 70
인증서 자동화: Amazon Linux 2에서 Cerbot의 Let's Encrypt ... 71
자습서: Amazon Linux에서 SSL/TLS 구성 ... 74
사전 조건 ... 75
1단계: 서버에서 TLS 활성화 ... 75
2단계: CA가 서명한 인증서 가져오기 ... 77
3단계: 보안 구성 테스트 및 강화 ... 81
문제 해결 ... 83
인증서 자동화: Amazon Linux에서 Cerbot의 Let's Encrypt ... 84
자습서: 애플리케이션의 가용성 향상 ... 87
사전 조건 ... 88
애플리케이션 확장 및 로드 밸런싱 ... 88
로드 밸런서 테스트 ... 90
Amazon 머신 이미지 ... 91
AMI 사용 ... 91
고유 AMI 생성 ... 91
AMI 구입, 공유 및 판매 ... 92
AMI 등록 해제 ... 92
Amazon Linux 2 및 Amazon Linux AMI ... 92
AMI 유형 ... 92
시작 권한 ... 93
루트 디바이스 스토리지 ... 93
가상화 유형 ... 95
Linux AMI 찾기 ... 96
Linux 콘솔을 사용하여 Amazon EC2 AMI 찾기 ... 97
AWS CLI를 사용하여 AMI 찾기 ... 97
시스템 관리자를 사용하여 최신 Amazon Linux AMI 찾기 ... 98
빠른 시작 AMI 검색 ... 98
공유 AMI ... 98
공유 AMI 검색 ... 99
퍼블릭 AMI 설정 ... 101
지정한 AWS 계정과 AMI 공유 ... 102
북마크 사용 ... 103
공유 Linux AMI 지침 ... 104
유료 AMI ... 108
AMI 판매 ... 109
유료 AMI 찾기 ... 109
유료 AMI 구입 ... 110
인스턴스에 대한 제품 코드 가져오기 ... 110
유료 지원 사용 ... 110
유료 및 지원된 AMI에 대한 청구서 ... 111
AWS Marketplace 구독 관리 ... 111
Amazon EBS 지원 Linux AMI 생성 ... 112
Amazon EBS 지원 AMIs 생성 개요 ... 112
인스턴스에서 Linux AMI 생성 ... 113
스냅샷에서 Linux AMI 만들기 ... 114
인스턴스 스토어 기반 Linux AMI 생성 ... 115
인스턴스 스토어 기반 AMI 생성 프로세스 개요 ... 115
사전 조건 ... 116

AMI 도구 설치 116
인스턴스 스토어 지원 인스턴스에서 AMI 생성 119
Amazon EBS 기반 AMI로 변환 126
AMI 도구 참조 129
EBS 지원 AMI를 통한 암호화 사용 145
인스턴스 시작 시나리오 145
이미지 복사 시나리오 148
AMI 복사 150
인스턴스 스토어 지원 AMI 복사 권한 151
리전 간 복사 151
계정 간 복사 152
암호화 및 복사 153
AMI 복사 154
대기 중인 AMI 복사 작업 중지 155
결제 정보 가져오기 155
AMI 결제 정보 필드 156
플랫폼 세부 정보 및 사용 작업 결제 코드 156
플랫폼 세부 정보 및 결제 정보 보기 157
Linux AMI 등록 취소 158
Amazon EBS 기반 AMI 정리 158
인스턴스 스토어 기반 AMI 정리 160
Amazon Linux 161
Amazon Linux 가용성 161
Amazon Linux 인스턴스에 연결 162
Amazon Linux 이미지 식별 162
AWS 명령줄 도구 163
패키지 리포지토리 164
Extras Library(Amazon Linux 2) 166
참조용으로 원본 패키지에 액세스 166
cloud-init 166
Amazon Linux 알림 구독 168
Amazon Linux 2를 온프레미스 가상 머신으로 실행 169
사용자 제공 커널 171
HVM AMIs(GRUB) 171
AMIs 반가상화(PV-GRUB) 172
인스턴스 178
인스턴스 유형 178
사용 가능한 인스턴스 유형 179
하드웨어 사양 181
AMI 가상화 유형 181
Nitro 기반 인스턴스 181
네트워킹 및 스토리지 기능 182
인스턴스 제한 185
범용 인스턴스 185
컴퓨팅 최적화 인스턴스 227
메모리 최적화 인스턴스 232
스토리지 최적화 인스턴스 242
액셀러레이티드 컴퓨팅 인스턴스 248
인스턴스 유형 찾기 261
인스턴스 유형 변경 262
추천 항목 얻기 265
인스턴스 구입 옵션 268
인스턴스 수명 주기 결정 269
온디맨드 인스턴스 270
예약 인스턴스 273
예약된 인스턴스 308
스팟 인스턴스 311

전용 호스트 ... 381
전용 인스턴스 ... 410
온디맨드 용량 예약 ... 415
인스턴스 수명 주기 ... 426
인스턴스 시작 ... 427
인스턴스 중지 및 시작(Amazon EBS 기반 인스턴스에만 해당) ... 427
인스턴스 최대 절전 모드(Amazon EBS 지원 인스턴스에만 해당) ... 428
인스턴스 재부팅 ... 428
인스턴스 만료 ... 429
인스턴스 종료 ... 429
재부팅, 중지, 최대 절전 모드 및 종료의 차이 ... 429
시작 ... 430
연결 ... 485
중지 및 시작 ... 507
최대 절전 모드 ... 510
재부팅 ... 520
만료 ... 520
Terminate ... 522
복구 ... 528
인스턴스 구성 ... 529
일반적인 구성 시나리오 ... 529
소프트웨어 관리 ... 529
사용자 관리 ... 535
프로세서 상태 제어 ... 537
시간 설정 ... 542
CPU 옵션 최적화 ... 546
호스트 이름 변경 ... 561
동적 DNS 설정 ... 564
시작 시 명령 실행 ... 565
인스턴스 메타데이터 및 사용자 데이터 ... 571
Elastic Inference ... 598
인스턴스 식별 ... 598
인스턴스 자격 증명 문서 검사 ... 598
시스템 UUID 검사 ... 598
모니터링 ... 600
자동 및 수동 모니터링 ... 601
자동 모니터링 도구 ... 601
수동 모니터링 도구 ... 602
모니터링 모범 사례 ... 602
인스턴스 상태 모니터링 ... 603
인스턴스 상태 확인 ... 603
예약된 이벤트 ... 607
CloudWatch를 사용해 인스턴스 모니터링하기 ... 616
세부 모니터링 활성화 ... 616
얻을 수 있는 측정치 나열 ... 617
지표에 대한 통계 구하기 ... 628
지표 그래프 ... 636
경보 만들기 ... 636
인스턴스를 중지, 종료, 재부팅 또는 복구하는 경보 만들기 ... 637
CloudWatch 이벤트를 사용한 Amazon EC2 자동화 ... 646
메모리 및 디스크 메트릭 모니터링 ... 646
CloudWatch 에이전트 ... 646
CloudWatch 모니터링 스크립트 ... 647
AWS CloudTrail을 사용하여 API 호출 로깅 ... 654
CloudTrail의 Amazon EC2 및 Amazon EBS 정보 ... 654
Amazon EC2 및 Amazon EBS 로그 파일 항목 이해 ... 655
EC2 Instance Connect를 통해 연결하는 사용자 감사 ... 656

네트워킹 ... 658
인스턴스 IP 주소 지정 ... 658
프라이빗 IPv4 주소 및 내부 DNS 호스트 이름 ... 658
퍼블릭 IPv4 주소 및 외부 DNS 호스트 이름 ... 659
탄력적 IP 주소(IPv4) ... 660
Amazon DNS 서버 ... 660
IPv6 주소 ... 660
인스턴스에 대한 IP 주소 작업 ... 661
다중 IP 주소 ... 665
고유 IP 주소 가져오기 ... 672
요구 사항 ... 673
AWS 계정으로 주소 범위를 가져오도록 준비하기 ... 673
AWS에서 사용할 수 있도록 주소 범위 프로비저닝 ... 675
AWS를 통해 주소 범위 알리기 ... 675
주소 범위 프로비저닝 취소 ... 676
엘라스틱 IP 주소 ... 676
탄력적 IP 주소 기본 사항 ... 676
탄력적 IP 주소 작업 ... 677
이메일 애플리케이션에 역방향 DNS 사용 ... 683
탄력적 IP 주소 제한 ... 683
네트워크 인터페이스 ... 683
네트워크 인터페이스 기본 사항 ... 684
인스턴스 유형별/네트워크 인터페이스당 IP 주소 ... 685
네트워크 인터페이스 시나리오 ... 694
네트워크 인터페이스 구성 모범 사례 ... 696
네트워크 인터페이스 작업 ... 697
요청자 관리 네트워크 인터페이스 ... 705
향상된 네트워킹 ... 706
향상된 네트워킹 유형 ... 707
인스턴스에서 향상된 네트워킹 기능 활성화 ... 707
향상된 네트워킹: ENA ... 707
향상된 네트워킹: intel 82599 VF ... 719
ENA 문제 해결 ... 725
Elastic Fabric Adapter(EFA) ... 731
EFA 기본 사항 ... 731
지원되는 인터페이스 및 라이브러리 ... 733
지원되는 인스턴스 유형 ... 733
지원되는 AMI ... 733
EFA 제한 사항 ... 733
EFA 및 MPI 시작하기 ... 733
EFA 및 NCCL 시작하기 ... 740
EFA로 작업 ... 754
EFA 모니터링 ... 757
배치 그룹 ... 757
클러스터 배치 그룹 ... 758
파티션 배치 그룹 ... 759
분산형 배치 그룹 ... 760
배치 그룹 규칙 및 제한 사항 ... 760
배치 그룹 생성 ... 761
배치 그룹으로 인스턴스 시작 ... 762
배치 그룹의 인스턴스 설명 ... 763
인스턴스에 대한 배치 그룹 변경 ... 764
배치 그룹 삭제 ... 765
네트워크 MTU ... 766
점보 프레임(9001 MTU) ... 767
경로 MTU 검색 ... 767
두 호스트 간 경로 MTU 확인 ... 768

Linux 인스턴스에서 MTU 확인 및 설정 ... 768
문제 해결 ... 769
Virtual Private Cloud ... 769
Amazon VPC 문서 ... 769
EC2-Classic ... 770
지원되는 플랫폼 감지 ... 770
EC2-Classic에서 사용 가능한 인스턴스 유형 ... 771
EC2-Classic과 VPC의 인스턴스 간 차이점 ... 772
EC2-Classic 및 VPC 간 리소스 공유 및 액세스 ... 776
ClassicLink ... 777
EC2-Classic에서 VPC로 마이그레이션 ... 788
보안 ... 799
인프라 보안 ... 799
네트워크 격리 ... 799
물리적 호스트에서 격리 ... 800
네트워크 트래픽 제어 ... 800
인터페이스 VPC 엔드포인트 ... 800
인터페이스 VPC 엔드포인트 생성 ... 801
인터페이스 VPC 엔드포인트 정책 생성 ... 801
복원성 ... 802
데이터 보호 ... 802
저장 데이터 암호화 ... 803
전송 중 데이터 암호화 ... 803
ID 및 액세스 관리 ... 803
인스턴스에 대한 네트워크 액세스 ... 804
Amazon EC2 권한 속성 ... 804
IAM 및 Amazon EC2 ... 804
IAM 정책 ... 806
IAM 역할 ... 850
네트워크 액세스 ... 859
키 페어 ... 861
Amazon EC2를 사용해 키 페어 생성 ... 862
Amazon EC2로 사용자의 퍼블릭 키 가져오기 ... 863
키 페어에 맞는 퍼블릭 키 검색(Linux) ... 865
키 페어에 맞는 퍼블릭 키 검색(Windows) ... 865
인스턴스에서 키 페어의 퍼블릭 키 검색 ... 865
키 페어의 지문 확인 ... 866
키 페어 삭제 ... 866
인스턴스의 키 페어 추가 또는 교체 ... 867
프라이빗 키를 분실했을 때 Linux 인스턴스에 연결하는 방법 ... 868
보안 그룹 ... 872
보안 그룹 규칙 ... 873
기본 보안 그룹 ... 875
사용자 지정 보안 그룹 ... 875
보안 그룹 작업 ... 876
보안 그룹 규칙 참조 ... 880
업데이트 관리 ... 885
규정 준수 확인 ... 885
스토리지 ... 887
Amazon EBS ... 888
Amazon EBS의 기능 ... 889
EBS 볼륨 ... 890
EBS 스냅샷 ... 926
EBS 데이터 서비스 ... 957
EBS 볼륨 및 NVMe ... 981
EBS 최적화 ... 984
EBS 성능 ... 997

EBS CloudWatch 지표 1011
EBS CloudWatch 이벤트 1017
인스턴스 스토어 1027
인스턴스 스토어 수명 1027
인스턴스 스토어 볼륨 1028
인스턴스 스토어 볼륨 추가 1033
SSD 인스턴스 스토어 볼륨 1036
인스턴스 스토리지 스왑 볼륨 1037
디스크 성능 최적화 1039
파일 스토리지 1040
Amazon EFS 1040
Amazon FSx 1044
Amazon S3 1044
Amazon S3 및 Amazon EC2 1045
인스턴스 볼륨 제한 1046
Linux 볼륨 제한 1046
Windows 볼륨 제한 1046
인스턴스 유형 제한 1047
대역폭 및 용량 비교 1047
디바이스 명명 1047
사용 가능한 디바이스 이름 1048
디바이스 이름 고려 사항 1048
블록 디바이스 매핑 1049
블록 디바이스 매핑의 개념 1049
AMI 블록 디바이스 매핑 1052
인스턴스 블록 디바이스 매핑 1054
리소스 및 태그 1059
리소스 위치 1059
리소스 ID 1060
더 긴 ID 작업 1061
긴 ID 설정에 대한 액세스 제어 1064
리소스 목록화 및 필터링 1065
고급 검색 1065
콘솔을 이용하여 리소스 목록화 1066
콘솔을 이용하여 리소스를 필터링 1067
CLI 및 API를 이용하여 목록화 및 필터링 1068
리소스에 태그 지정 1068
태그 기본 사항 1069
리소스에 태그 지정 1070
태그 제한 1072
리소스에 결제용 태그 지정 1073
콘솔을 사용한 태그 작업 1073
CLI 또는 API를 사용한 태그 작업 1076
서비스 제한 1079
현재 제한 조회 1079
제한 증가 요청 1081
포트 25를 사용하여 전송되는 이메일 관련 제한 1081
사용 보고서 1081
문제 해결 1083
시작 문제 해결 1083
인스턴스 제한을 초과함 1083
부족한 인스턴스 용량 1084
인스턴스 즉시 종료 1084
인스턴스에 연결 1085
인스턴스 연결 중 오류 발생: 연결 시간 초과 1086
오류: ... 키를 로드할 수 없습니다. 예상: 모든 프라이빗 키 1087
오류: 서버에서 사용자 키를 인식하지 못함 1088

오류: 호스트 키를 찾을 수 없음. 권한 거부(퍼블릭 키) 또는 인증 실패, 권한 거부 1089
오류: 보호되지 않는 프라이빗 키 파일 1090
오류: 프라이빗 키는 '-----BEGIN RSA PRIVATE KEY-----'로 시작하고 '-----END RSA PRIVATE KEY-----'로 끝나야 합니다. 1090
오류: 서버에서 키 거부 또는 지원되는 인증 방법이 없음 1091
내 브라우저를 사용하여 연결할 수 없음 1091
인스턴스를 ping할 수 없음 1091
오류: 서버에서 예기치 않게 네트워크 연결을 차단함 1092
인스턴스 중지 1092
대체 인스턴스 생성 1092
인스턴스 종료 1093
지연된 인스턴스 종료 1093
종료된 인스턴스가 계속 표시됨 1094
인스턴스 자동 시작 또는 종료 1094
상태 확인 실패 1094
상태 점검 정보 검토 1095
시스템 로그 검색 1095
Linux 기반 인스턴스의 시스템 로그 오류 문제 해결 1096
메모리 부족: 프로세스 중지 1097
ERROR: mmu_update failed(메모리 관리 업데이트 실패) 1097
I/O 오류(블록 디바이스 장애) 1098
I/O ERROR: neither local nor remote disk(분산된 블록 디바이스 손상) 1099
request_module: runaway loop modprobe(이전 Linux 버전에서 레거시 커널 modprobe 반복) 1100
"FATAL: kernel too old" 및 "fsck: No such file or directory while trying to open /dev"(커널과 AMI 불일치) 1100
"FATAL: Could not load /lib/modules" 또는 "BusyBox"(커널 모듈 누락) 1101
ERROR Invalid kernel(EC2 커널이 호환되지 않음) 1102
fsck: No such file or directory while trying to open... (파일 시스템을 찾을 수 없음) 1103
파일 시스템 마운트 관련 일반 오류(마운트 실패) 1105
VFS: Unable to mount root fs on unknown-block(루트 파일 시스템 불일치) 1106
Error: Unable to determine major/minor number of root device... (루트 파일 시스템/디바이스 불일치) 1107
XENBUS: Device with no driver... 1108
... days without being checked, check forced(파일 시스템 검사 필요) 1109
fsck died with exit status... (디바이스 누락) 1109
GRUB 프롬프트(grubdom>) 1110
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) 1112
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) 1113
XENBUS: Timeout connecting to devices(Xenbus 시간 초과) 1114
연결할 수 없는 인스턴스 문제 해결 1115
인스턴스 재부팅 1115
인스턴스 콘솔 출력 1115
연결할 수 없는 인스턴스의 스크린샷 캡처 1116
호스트 컴퓨터 실패 시 인스턴스 복구 1117
잘못된 볼륨에서 부팅 1117
Linux용 EC2Rescue 1119
Linux용 EC2Rescue 설치 1119
(선택 사항) Linux용 EC2Rescue의 서명 확인 1120
Linux용 EC2Rescue로 작업 1122
EC2Rescue 모듈 개발 1124
진단 인터럽트 보내기 1128
지원되는 인스턴스 유형 1128
사전 조건 1129
진단 인터럽트 보내기 1131

문서 기록 1132

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. 또한 Amazon EC2는 요구 사항이나 갑작스러운 인기 증대 등 변동 사항에 따라 신속하게 규모를 확장하거나 축소할 수 있어 서버 트래픽 예측 필요성이 줄어듭니다.

클라우드 컴퓨팅에 대한 자세한 정보는 클라우드 컴퓨팅이란 무엇입니까? 단원을 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 페어를 사용하여 인스턴스 로그인 정보 보호(AWS는 퍼블릭 키를 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
- 탄력적 IP 주소(EIP): 동적 클라우드 컴퓨팅을 위한 고정 IPv4 주소
- 태그: 사용자가 생성하여 Amazon EC2 리소스에 할당할 수 있는 메타데이터
- AWS 클라우드에서는 논리적으로 격리되어 있지만, 원할 때 마다 고객의 네트워크와 간편히 연결할 수 있는 가상 네트워크, Virtual Private Clouds(VPC)

Amazon EC2에 대한 자세한 내용은 Amazon EC2제품 페이지를 참조하십시오.

AWS에서의 웹사이트 실행에 대한 자세한 내용은 웹 호스팅을 참조하십시오.

처음으로 Amazon EC2 사용하기

먼저, Amazon EC2를 사용하도록 설정해야 합니다. 일단 설정을 하시면, Amazon EC2 시작 자습서를 따라 하실 수 있습니다. 기능에 대한 추가 정보가 필요한 경우에는 기술 문서를 참조하십시오.

실행 안내

- Amazon EC2로 설정 (p. 22)
- Amazon EC2 Linux 인스턴스 시작하기 (p. 27)

기본

- 인스턴스 및 AMI (p. 4)

- 리전 및 가용 영역 (p. 6)
- 인스턴스 유형 (p. 178)
- Tags (p. 1068)

네트워킹 및 보안

- Amazon EC2 키 페어 (p. 861)
- 보안 그룹 (p. 872)
- 탄력적 IP 주소 (p. 676)
- Amazon EC2 및 Amazon VPC (p. 769)

스토리지

- Amazon EBS (p. 888)
- 인스턴스 스토어 (p. 1027)

Linux 인스턴스 작업

- AWS 시스템 관리자 사용 설명서의 AWS 시스템 관리자 Run Command
- 자습서: Amazon Linux 2에 LAMP 웹 서버 설치 (p. 32)
- 자습서: Amazon Linux 2에서 SSL/TLS 구성 (p. 60)
- AWS 시작: Linux용 웹 앱 호스팅

AWS 구매에 관련된 질문은 AWS 영업부에 문의하십시오. Amazon EC2에 관련된 기술적인 문의 사항은 Amazon EC2 forum 단원을 참조하십시오.

관련 서비스

Amazon EC2를 사용하여 인스턴스, 볼륨 같은 Amazon EC2 리소스를 직접 프로비저닝할 수 있습니다. 또한 AWS의 다른 서비스를 사용하여 Amazon EC2를 프로비저닝할 수도 있습니다. 자세한 내용은 다음 설명서를 참조하세요.

- Amazon EC2 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 단원을 참조하십시오.

인스턴스와 Amazon EBS 볼륨에 관련된 기본 통계 정보를 모니터링하려면 Amazon CloudWatch를 사용합니다. 자세한 내용은 Amazon CloudWatch 사용 설명서 단원을 참조하십시오.

새로운 Amazon EC2 인스턴스를 시작할 때마다 Lambda 함수를 활성화하는 등의 작업을 자동화하려면 Amazon CloudWatch Events를 사용합니다. 자세한 내용은 Amazon CloudWatch Events 사용 설명서 단원을 참조하십시오.

AWS Management 콘솔, 명령줄 도구 및 기타 서비스에서의 호출을 포함하여 계정의 Amazon EC2 API에 대한 호출을 모니터링하려면 AWS CloudTrail을 사용합니다. 자세한 내용은 AWS CloudTrail User Guide를 참조하십시오.

Amazon Relational Database Service(Amazon RDS)로 데이터베이스 인스턴스를 실행하여, AWS에 의해 관리되는, 관계형 데이터베이스를 사용하실 수 있습니다. EC2 인스턴스로 데이터베이스를 구축하는 것도 가능하지만, Amazon RDS를 선택하면, 직접 소프트웨어 패치 적용, 백업 및 백업 데이터 저장 등 데이터베이스 관리 작업을 할 필요가 없습니다. 자세한 내용은 Amazon Relational Database Service 개발자 안내서 단원을 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 설명서를 참조하십시오.

Amazon EC2에 액세스

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

명령줄 인터페이스를 선호하는 고객의 경우 다음과 같은 옵션이 있습니다.

AWS명령줄 인터페이스(CLI)

다양한 AWS 제품에서 사용되는 명령어를 제공하며 Windows, Mac, Linux를 지원합니다. 시작하려면 AWS Command Line Interface 사용 설명서단원을 참조하십시오. Amazon EC2 명령어에 대한 자세한 내용은 EC2(AWS CLI Command Reference) 단원을 참조하십시오.

Windows PowerShell용 AWS 도구

PowerShell 환경에서 스크립트 작업을 선호하는 다양한 AWS 제품을 관리할 수 있도록, 명령줄 도구를 제공합니다. 시작하려면 Windows PowerShell용 AWS 도구 사용 설명서단원을 참조하십시오. Amazon EC2용 cmdlets에 대한 자세한 내용은 PowerShell용 AWS 도구 Cmdlet Reference 단원을 참조하십시오.

Amazon EC2에서는 쿼리 API를 제공합니다. 이 리퀘스트들은, HTTP나 HTTPS의 메시지 교환 방식인 GET이나 POST이며, 미리 정해진 이름인 "`Action`"을 쿼리 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 개발자를 위해 AWS는, 라이브러리, 샘플 코드, 자습서 및 기타 리소스를 제공합니다. 이 라이브러리는 HTTP/HTTPS 리퀘스트에 암호화된 사인하기, 다시 리퀘스트를 보내기, 오류 응답 처리하기 등의, 작업을 자동화할 수 있는 기본적인 기능을 재공합니다. 자세한 내용은 AWS SDK 및 도구를 참조하십시오.

Amazon EC2 가격

AWS 가입 시 무상으로 Amazon EC2를 시작할 수 있는 AWS 프리 티어를 제공합니다.

예를 들어, Amazon EC2는 다음의 구입 옵션이 있습니다:

온디맨드 인스턴스

장기 약정이나 선결제 금액 없이 초 단위로 사용한 인스턴스에 대한 요금을 지불하는 방식입니다.

Savings Plans

1년 또는 3년 기간 동안 시간당 USD로 일관된 사용량을 약정하여 Amazon EC2 비용을 절감할 수 있습니다.

예약 인스턴스

1년 또는 3년 기간 동안 인스턴스 유형 또는 지역을 포함해 특정 인스턴스 구성을 약정하여 Amazon EC2 비용을 절감할 수 있습니다.

스팟 인스턴스

미사용 EC2 인스턴스를 요청하여 Amazon EC2 비용을 대폭 줄일 수 있습니다.

Amazon EC2에 관련된 전체적인 요금 및 가격 목록은 Amazon EC2 요금을 참조하십시오.

예시로 주어진 프로비저닝 환경에서의 비용은 클라우드 경제 센터에서 계산할 수 있습니다.

청구 요금은 AWS Billing and Cost Management 콘솔의 결제 및 비용 관리 대시보드에서 확인할 수 있습니다. 청구서에는 요금 내역을 자세하게 확인할 수 있는 사용 보고서 링크가 포함됩니다. AWS 계정 결제에 대한 자세한 내용은 AWS 계정 결제 단원을 참조하십시오.

AWS 결제, 계정 및 이벤트에 관련된 질문은 AWS Support에 문의하십시오.

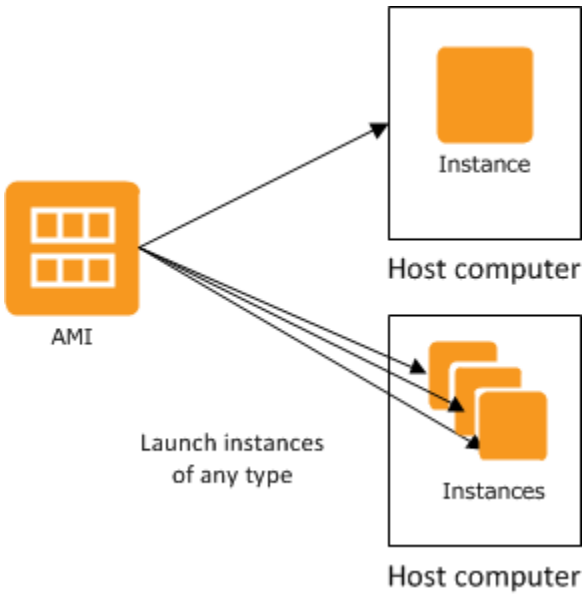
AWS 환경에서의 비용과 보안, 성능 최적화를 돕는 Trusted Advisor의 개요는 AWS Trusted Advisor 단원을 참조하십시오.

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 1을 참조하십시오.

인스턴스 및 AMI

Amazon 머신 이미지(AMI)는 소프트웨어 구성이 기재된 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

인스턴스는 클라우드의 가상 서버입니다. 시작 시 구성은 인스턴스를 시작할 때 지정한 AMI의 사본입니다.

하나의 AMI에서 다양한 인스턴스 유형을 실행할 수 있습니다. 인스턴스 유형에 따라 인스턴스에 사용되는 호스트 컴퓨터의 하드웨어가 기본적으로 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 유형을 참조하십시오.

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

AWS 계정당 동시에 실행할 수 있는 인스턴스 수는 제한됩니다. 해당 제한 및 추가 요청 방법에 대한 자세한 내용은 Amazon EC2의 실행 인스턴스 한도(일반 FAQ의 Amazon EC2) 단원을 참조하십시오.

인스턴스 스토리지

인스턴스의 루트 디바이스에는 인스턴스 부팅에 사용되는 이미지가 포함되어 있습니다. 자세한 내용은 Amazon EC2 루트 디바이스 볼륨 (p. 16) 단원을 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. 1049) 단원을 참조하십시오. 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 데이터를 안전하게 유지하려면 여러 인스턴스에 걸쳐 복제 전략을 사용하거나 영구적 데이터를 Amazon S3 또는 Amazon EBS 볼륨에 저장해야 합니다. 자세한 내용은 스토리지 (p. 887) 단원을 참조하십시오.

보안 모범 사례

- AWS Identity and Access Management(IAM)을 사용하여 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 EC2에 대한 자격 증명 및 액세스 관리 (p. 803) 단원을 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 Amazon EC2 인스턴스에 대한 Linux 보안 그룹 (p. 872) 단원을 참조하십시오.
- 보안 그룹의 규칙을 정기적으로 검토하고 최소 권한 부여라는 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. 105) 단원을 참조하십시오. 안전한 AMI 공유에 대한 자세한 내용은 공유 AMI (p. 98) 단원을 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 중지됨 상태에 있는 동안에는 추가 인스턴스 사용량에 대한 요금이 부과되지 않습니다. 중지됨 상태에서 실행 중 상태로 전환할 때마다 최소 1분의 요금이 부과되며. 인스턴스가 중지된 상태에서 인스턴스 유형을 변경하면, 다음에 인스턴스가 시작된 후 신규 인스턴스 유형에 대한 요금이 부과됩니다. 모든 연결 Amazon EBS 루트 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 만들수도 있으며, 커널, 램 디스크, 인스턴스 유형을 변경할 수 있습니다.

인스턴스 종료

인스턴스가 종료될 때 인스턴스는 일반 종료를 수행합니다. 루트 디바이스 볼륨은 기본적으로 삭제되지만 모든 연결된 Amazon EBS 볼륨은 기본적으로 유지됩니다. 이는 각 볼륨의 `deleteOnTermination` 속성 설정에 따라 결정됩니다. 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. 426) 단원을 참조하십시오.

AMI

Amazon Web Services(AWS)에서는 자주 사용되는 소프트웨어 구성을 포함하는 다양한 Amazon 머신 이미지(AMI) (p. 96)를 공개 게시하고 있습니다. 그뿐 아니라 AWS 개발자 커뮤니티 회원들이 올린 자체 구성 AMI도 게시되어 있습니다. 사용자 지정 AMI는 누구나 생성할 수 있으며 AMI를 생성하면 필요한 기능을 모두 갖춘 새 인스턴스를 쉽고 빠르게 시작할 수 있습니다. 예를 들어 애플리케이션이 웹사이트나 웹 서비스인 경우, 웹 서버와 관련 고정 콘텐츠, 그리고 동적 페이지에 사용할 코드가 포함된 AMI를 정의해 만드실 수 있습니다. 이 AMI에서 인스턴스를 시작하면 웹 서버가 시작되고 애플리케이션에서 바로 요청을 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.

AMI에대한 설명을 보시면, 그 인스턴스의 루트디바이스가 `ebs` 인지 `instance store`인지 알수 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. 93) 단원을 참조하십시오.

리전, 가용 영역 및 로컬 영역

Amazon EC2는 세계 각지의 여러 곳에서 호스팅되고 있습니다. 이 위치는 리전, 가용 영역 및 로컬 영역으로 구성됩니다. 각 리전은 개별 지리 영역입니다. 리전마다 가용 영역이라는 격리된 위치가 여러 개 있습니다. 로컬 영역에서는 최종 사용자에게 가까운 여러 위치에 컴퓨팅, 스토리지 등의 리소스를 배치할 수 있는 기능을 제공합니다. 특별히 선택하지 않는 한 리소스는 여러 리전에 복제되지 않습니다.

Amazon은 최신 기술을 탑재한 고가용성 데이터 센터를 운영하고 있습니다. 드물기는 하지만 동일한 위치에 있는 인스턴스의 가용성에 영향을 미치는 장애가 발생할 수도 있습니다. 장애의 영향을 받는 위치 한 곳에서 모든 인스턴스를 호스팅하면 인스턴스를 전혀 사용하지 못하게 될 수 있습니다.

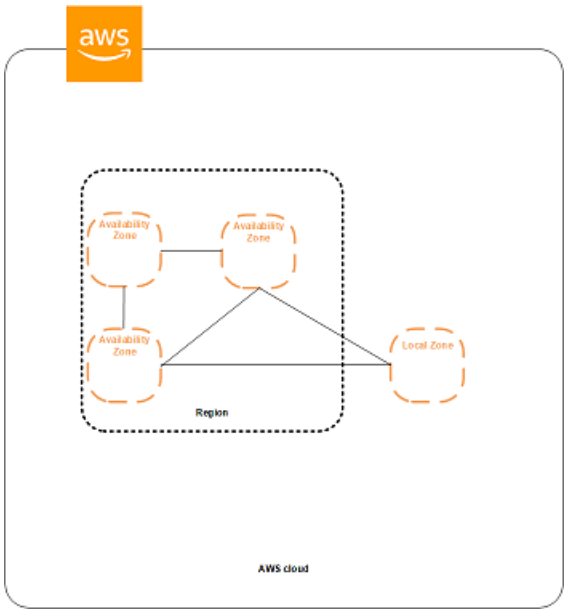
목차

- 리전, 가용 영역 및 로컬 영역 개념 (p. 7)

- 사용 가능한 리전 (p. 9)
- 리전 및 Endpoint (p. 10)
- 리전, 가용 영역 및 로컬 영역 설명 (p. 10)
- 리소스에 대한 리전 지정 (p. 12)
- 로컬 영역 활성화 (p. 14)
- 로컬 영역 비활성화 (p. 15)
- 가용 영역 또는 로컬 영역에서 인스턴스 시작 (p. 15)
- 다른 가용 영역으로 인스턴스 마이그레이션 (p. 15)

리전, 가용 영역 및 로컬 영역 개념

각 리전은 완전히 독립적입니다. 각 가용 영역은 서로 격리되어 있지만, 한 리전 안의 가용 영역은 지연 시간이 짧은 링크를 통해 연결되어 있습니다. 로컬 영역은 선택한 서비스를 최종 사용자에게 더 가까이 배치하는 AWS 인프라 배포입니다. 로컬 영역은 사용자의 리전과 다른 위치에 있는 리전의 확장입니다. AWS 인프라에 높은 대역폭 백본을 제공하고, 기계 학습과 같이 지연 시간에 민감한 애플리케이션에 적합합니다. 다음 다이어그램은 리전, 가용 영역 및 로컬 영역 간의 관계를 보여줍니다.



Amazon EC2 리소스는 전역 리소스이거나 리전, 가용 영역 또는 로컬 영역에 연결된 리소스입니다. 자세한 내용은 리소스 위치 (p. 1059) 단원을 참조하십시오.

목차

- 리전 (p. 7)
- 가용 영역 (p. 8)
- 로컬 영역 (p. 8)
- 네트워크 경계 그룹 (p. 9)

리전

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

리소스를 볼 때 지정한 리전에 연결된 리소스만 표시됩니다. 리전이 서로 격리되어 있고 여러 리전에 리소스가 자동으로 복제되지 않기 때문입니다.

인스턴스를 시작할 때 동일한 리전에 있는 AMI를 선택해야 합니다. AMI가 다른 리전에 있는 경우 해당 AMI를 사용 중인 리전에 복사할 수 있습니다. 자세한 내용은 AMI 복사 (p. 150)을 참조하십시오.

리전 간 데이터 전송 시 비용이 청구됩니다. 자세한 내용은 Amazon EC2 요금 - 데이터 전송을 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. 676)을 참조하십시오.

가용 영역은 리전 코드와 식별 문자의 조합으로 표시됩니다(예: `us-east-1a`). 리소스가 리전의 가용 영역에 걸쳐 배포될 수 있도록 AWS는 각 AWS 계정의 이름에 가용 영역을 독립적으로 매핑합니다. 예를 들어 AWS 계정의 `us-east-1a` 가용 영역은 다른 AWS 계정에 대한 `us-east-1a` 가용 영역과 위치가 동일하지 않을 수 있습니다.

계정에 대해 가용 영역을 조정하려면 가용 영역에 대한 고유하고 일관된 식별자인 AZ ID를 사용해야 합니다. 예를 들어, `use1-az1`은 `us-east-1` 리전의 AZ ID이고, 모든 AWS 계정에서 위치가 동일합니다.

AZ ID를 확인하면 다른 계정의 리소스를 기준으로 한 계정의 리소스 위치를 확인할 수 있습니다. 예를 들어, AZ ID가 `use-az2`인 가용 영역의 서브넷을 다른 계정과 공유하면 이 서브넷은 AZ ID가 `use-az2`인 가용 영역의 계정에서 사용할 수 있습니다. 각 VPC 및 서브넷의 AZ ID가 Amazon VPC 콘솔에 표시됩니다. 자세한 내용은 Amazon VPC 사용 설명서에서 VPC 공유 작업을 참조하십시오.

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 새 계정에 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 리전, 가용 영역 및 로컬 영역 설명 (p. 10) 단원을 참조하십시오.

로컬 영역

로컬 영역은 사용자와 지리적으로 근접한 AWS 리전의 확장입니다. 인스턴스를 시작할 때 로컬 영역에서 서브넷을 선택할 수 있습니다. 로컬 영역은 자체 인터넷 연결 기능을 갖추었으며 AWS Direct Connect를 지원합니다. 따라서 로컬 영역에서 생성된 리소스는 로컬 사용자에게 지연 시간이 매우 짧은 통신을 제공할 수 있습니다. 자세한 내용은 AWS 로컬 영역을 참조하십시오.

리전 코드 뒤에 위치를 나타내는 식별자를 붙여 로컬 영역을 나타냅니다(예: `us-west-2-lax-1a`).

로컬 영역을 사용하려면 먼저 활성화해야 합니다. 자세한 내용은 로컬 영역 활성화 (p. 14) 단원을 참조하십시오. 다음으로, 로컬 영역에서 서브넷을 만듭니다. 마지막으로 로컬 영역 서브넷에서 다음 리소스 중 하나를 실행하여 애플리케이션이 최종 사용자와 더 가까워지도록 합니다.

- Amazon EC2 인스턴스
- Amazon EBS 볼륨

- Amazon FSx 파일 서버
- Application Load Balancer
- 전용 호스트

일부 리전에서는 로컬 영역을 사용할 수 없습니다. 로컬 영역이 지원되는 리전에 대한 자세한 내용은 the section called “사용 가능한 리전” (p. 9) 단원을 참조하십시오.

계정에서 사용할 수 있는 로컬 영역을 나열할 수 있습니다. 자세한 내용은 리전, 가용 영역 및 로컬 영역 설명 (p. 10) 단원을 참조하십시오.

네트워크 경계 그룹

네트워크 경계 그룹은 AWS에서 IP 주소를 알리는 고유한 가용 영역 또는 로컬 영역 집합입니다. 네트워크 경계 그룹에서 다음 리소스를 할당할 수 있습니다.

- Amazon에서 제공하는 탄력적 IPv4 주소
- Amazon에서 제공하는 IPv6 VPC 주소

네트워크 경계 그룹은 주소를 해당 그룹으로 제한합니다. IP 주소를 네트워크 경계 그룹 간에 이동할 수 없습니다.

사용 가능한 리전

계정을 통해 자신이 사용할 수 있는 리전을 결정합니다. 다음 예를 참조하십시오.

- 하나의 AWS 계정은 여러 개의 리전을 제공하므로 사용자는 자신의 요구 사항에 맞는 위치에서 Amazon EC2 인스턴스를 시작할 수 있습니다. 예를 들어 유럽의 고객들과 좀더 가까운 곳에 위치하거나 또는 법적 요구사항을 준수하기 위해 유럽에 소재한 위치에서 인스턴스를 실행할 필요가 있을 수 있습니다.
- AWS GovCloud(미국 서부) 계정은 AWS GovCloud(미국 서부) 리전에만 액세스할 수 있습니다. 자세한 내용은 AWS GovCloud(미국 서부) 리전 단원을 참조하십시오.
- Amazon AWS (중국) 계정은 오직 베이징 및 닝샤 리전에 대한 액세스 권한을 제공합니다. 자세한 내용은 중국의 AWS를 참조하십시오.

다음 표에는 AWS 계정이 제공하는 리전이 나열되어 있습니다. AWS GovCloud(미국 서부) 또는 중국 리전과 같은 AWS 계정의 추가 리전은 설명하거나 액세스할 수 없습니다. 2019년 3월 20일 이후에 도입된 지역을 사용하려면 지역을 사용하도록 설정해야 합니다. 자세한 내용은 AWS General Reference의 AWS 지역 관리를 참조하십시오.

Code	이름	옵트인 상태	로컬 영역
`us-east-2`	미국 동부(오하이오)	불필요	아니요
`us-east-1`	미국 동부(버지니아 북부)	불필요	아니요
`us-west-1`	미국 서부(캘리포니아 북부 지역)	불필요	아니요
`us-west-2`	미국 서부(오레곤)	불필요	예 - `us-west-2-lax-1a` 로컬 영역을 활성화해야 합니다.

Code	이름	옵트인 상태	로컬 영역
ap-east-1	아시아 태평양(홍콩)	필수	아니요
ap-south-1	아시아 태평양(뭄바이)	불필요	아니요
ap-northeast-3	아시아 태평양(오사카-로컬)	불필요	아니요
ap-northeast-2	아시아 태평양(서울)	불필요	아니요
ap-southeast-1	아시아 태평양(싱가포르)	불필요	아니요
ap-southeast-2	아시아 태평양(시드니)	불필요	아니요
ap-northeast-1	아시아 태평양(도쿄)	불필요	아니요
ca-central-1	캐나다(중부)	불필요	아니요
eu-central-1	유럽(프랑크푸르트)	불필요	아니요
eu-west-1	유럽(아일랜드)	불필요	아니요
eu-west-2	유럽(런던)	불필요	아니요
eu-west-3	유럽(파리)	불필요	아니요
eu-north-1	유럽(스톡홀름)	불필요	아니요
me-south-1	중동(바레인)	필수	아니요
sa-east-1	남아메리카(상파울루)	불필요	아니요

자세한 정보는 AWS 글로벌 인프라를 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 리전, 가용 영역 및 로컬 영역 설명 (p. 10) 단원을 참조하십시오.

리전 및 Endpoint

명령줄 인터페이스 또는 API 작업을 사용해서 인스턴스로 작업할 때는 리전 엔드포인트를 지정해야 합니다. Amazon EC2 리전 및 엔드포인트에 대한 자세한 내용은 Amazon Web Services 일반 참조의 리전 및 엔드포인트를 참조하십시오.

AWS GovCloud(미국 서부)의 엔드포인트 및 프로토콜에 대한 자세한 내용은 AWS GovCloud (US) User Guide의 AWS GovCloud(미국 서부) 엔드포인트를 참조하십시오.

리전, 가용 영역 및 로컬 영역 설명

Amazon EC2 콘솔 또는 명령줄 인터페이스를 사용하여 계정에 사용할 수 있는 리전, 가용 영역 및 로컬 영역을 확인할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

콘솔을 사용하여 리전, 가용 영역 및 로컬 영역을 찾으려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 모음에서 리전 선택기의 옵션을 확인합니다.

3. 탐색 창에서 EC2 대시보드를 선택합니다.
4. 가용 영역과 로컬 영역은 서비스 상태, 가용 영역 상태 아래에 나열됩니다.

AWS CLI를 사용하여 리전, 가용 영역 및 로컬 영역을 찾으려면

1. 다음과 같이 describe-regions 명령을 사용하여 계정의 리전을 설명합니다.

```
aws ec2 describe-regions
```

   계정에서 사용 중지된 모든 지역을 포함하여 모든 지역을 설명하려면 다음과 같이 `--all-regions` 옵션을 추가하십시오.

```
aws ec2 describe-regions --all-regions
```

2. 다음과 같이 describe-availability-zones 명령을 사용하여 지정된 리전 내의 가용 영역과 로컬 영역을 설명합니다.

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

3. 다음과 같이 describe-availability-zones 명령을 사용하여 옵트인 상태에 관계없이 가용 영역과 로컬 영역을 설명합니다.

```
aws ec2 describe-availability-zones --all-availability-zones
```

Windows PowerShell용 AWS 도구를 사용하여 리전, 가용 영역 및 로컬 영역을 찾으려면

1. 다음과 같이 Get-EC2Region 명령을 사용하여 계정의 리전을 설명합니다.

```
PS C:\> Get-EC2Region
```

2. 다음과 같이 Get-EC2AvailabilityZone 명령을 사용하여 지정된 리전 내의 가용 영역을 설명합니다.

```
PS C:\> Get-EC2AvailabilityZone -Region region-name
```

리소스에 대한 리전 지정

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

> **Note**
>
> 일부 리전, 가용 영역 및 로컬 영역에서 사용할 수 없는 AWS 리소스도 있습니다. 특정 가용 영역에서 인스턴스를 시작하기 전에 원하는 리전 또는 가용 영역에 필요한 리소스를 생성할 수 있는지 확인합니다.

콘솔을 사용하여 리소스에 대한 리전을 지정하려면

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

명령줄을 사용하여 기본 리전을 지정하려면

원하는 리전 엔드포인트로 환경 변수의 값을 설정할 수 있습니다(예: `https://ec2.us-east-2.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-east-2`.

Amazon EC2에 대한 자세한 내용은 Amazon Elastic Compute Cloud 엔드포인트 단원을 참조하십시오.

로컬 영역 활성화

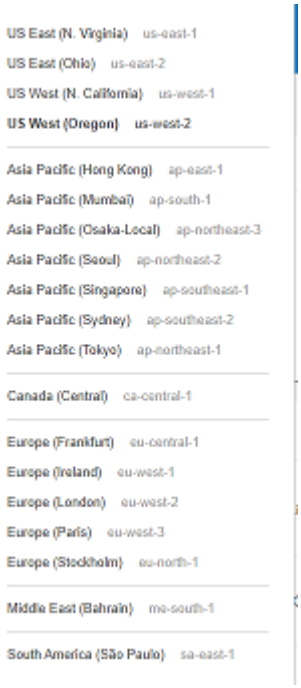
리소스나 서비스에 로컬 영역을 지정하기 전에 해당 영역을 활성화해야 합니다.

AWS Management 콘솔 또는 AWS CLI를 사용하여 로컬 영역를 활성화할 수 있습니다.

> **Note**
>
> 기본적으로 모든 가용 영역이 활성화되며 비활성화할 수 없습니다.
> 일부 리전에서 사용할 수 없는 AWS 리소스도 있습니다. 특정 로컬 영역에서 인스턴스를 시작하기 전에 원하는 리전 또는 로컬 영역에 필요한 리소스를 생성할 수 있는지 확인합니다.

콘솔을 사용하여 로컬 영역을 활성화하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 모음에서 리전 선택기를 사용하여 리전을 선택합니다.



3. 탐색 창에서 EC2 대시보드를 선택합니다.
4. 가용 영역 상태에서 Enable additional Local Zones(추가 로컬 영역 활성화)를 선택합니다.
5. Local Zone Groups(로컬 영역 그룹)에서 원하는 각 로컬 영역을 활성화합니다.
6. 활성화 확인 대화 상자에서 활성화를 입력하고 확인을 선택합니다.

AWS CLI를 사용하여 로컬 영역을 활성화하려면

- modify-availability-zone-group 명령을 사용합니다.

로컬 영역 비활성화

로컬 영역을 비활성화하려면 AWS Support에 문의해야 합니다.

> **Important**
>
> 로컬 영역을 비활성화하기 전에 모든 리소스를 제거하십시오. 로컬 영역에 남아 있는 모든 리소스에 대해 요금이 발생합니다. 리소스를 제거한 후 Disable Zone Group이라는 제목으로 AWS Support에 사례를 생성합니다.

가용 영역 또는 로컬 영역에서 인스턴스 시작

인스턴스를 시작할 때 특정 고객과 가까운 곳에 인스턴스를 배치하거나 법률 또는 기타 요구 사항을 준수할 수 있도록 적절한 리전을 선택합니다. 각각의 개별적인 가용 영역에서 인스턴스를 시작함으로써 단일 위치에서 장애가 발생할 경우 애플리케이션을 보호할 수 있습니다.

로컬 영역에서 인스턴스를 시작하면 지연 시간에 민감한 애플리케이션을 최종 사용자와 가까운 곳에서 실행하고 AWS 인프라의 이점을 누릴 수 있습니다.

인스턴스를 시작할 때 상황에 따라 사용 중인 리전의 가용 영역이나 로컬 영역을 지정할 수 있습니다. 가용 영역이나 로컬 영역을 지정하지 않으면 가용 영역이 자동으로 선택됩니다. 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. 추가 인스턴스를 시작하는 경우 새 인스턴스가 실행 중인 인스턴스와 가까이 있거나 분리되어 있어야 하는 경우에만 가용 영역을 지정하십시오.

다른 가용 영역으로 인스턴스 마이그레이션

필요하다면 한 가용 영역에서 다른 가용 영역으로 인스턴스를 마이그레이션할 수 있습니다. 예를 들어 인스턴스 유형을 수정하려고 하는데 현재 가용 영역에서 새로운 유형의 인스턴스를 시작할 수 없다면 새로운 유형의 인스턴스를 시작할 수 있는 가용 영역으로 인스턴스를 마이그레이션할 수 있습니다.

마이그레이션 프로세스는 다음과 같이 진행됩니다.

- 원래 인스턴스에서 AMI 생성
- 새 가용 영역에서 인스턴스 시작
- 다음 절차에 나오는 대로 새 인스턴스의 구성 업데이트

다른 가용 영역으로의 인스턴스 마이그레이션 방법

1. 인스턴스에서 AMI를 만듭니다. 이 절차는 인스턴스의 운영 체제 및 루트 디바이스 볼륨 유형에 따라 달라집니다. 자세한 내용은 사용자의 운영 체제 및 루트 디바이스 볼륨에 대한 문서를 참조하십시오.
   - Amazon EBS 지원 Linux AMI 생성 (p. 112)
   - 인스턴스 스토어 기반 Linux AMI 생성 (p. 115)
   - Amazon EBS 기반 Windows AMI 생성
2. 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 영역 또는 로컬 영역에서 인스턴스 시작 (p. 15) 단원을 참조하십시오.
4. 원래 인스턴스가 연결된 탄력적 IP 주소를 가지고 있는 경우 이를 새 인스턴스와 연결합니다. 자세한 내용은 탄력적 IP 주소 연결 해제 (p. 681) 단원을 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. 302) 단원을 참조하십시오.
6. (선택 사항) 원래 인스턴스를 종료합니다. 자세한 내용은 인스턴스 종료 (p. 523) 단원을 참조하십시오.

Amazon EC2 루트 디바이스 볼륨

인스턴스를 시작하면 인스턴스 부팅에 사용된 이미지가 루트 디바이스 볼륨에 저장됩니다. 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. 1047) 단원을 참조하십시오.

주제

- 루트 디바이스 스토리지 개념 (p. 16)
- 루트 디바이스 유형에 따른 AMI 선택 (p. 18)
- 인스턴스의 루트 디바이스 유형 확인 (p. 18)
- 루트 디바이스 볼륨 지속 설정 (p. 19)

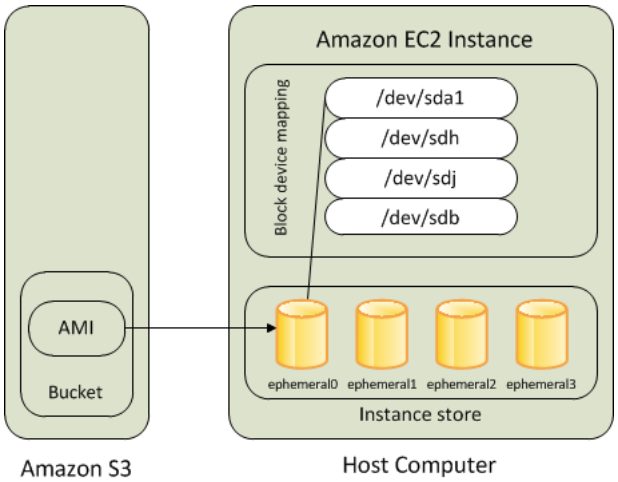
루트 디바이스 스토리지 개념

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. 93) 단원을 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 중지 작업을 지원하지 않음) 장애가 발생하면(예: 기본 드라이브에 문제가 있는 경우) 데이터가 삭제됩니다.

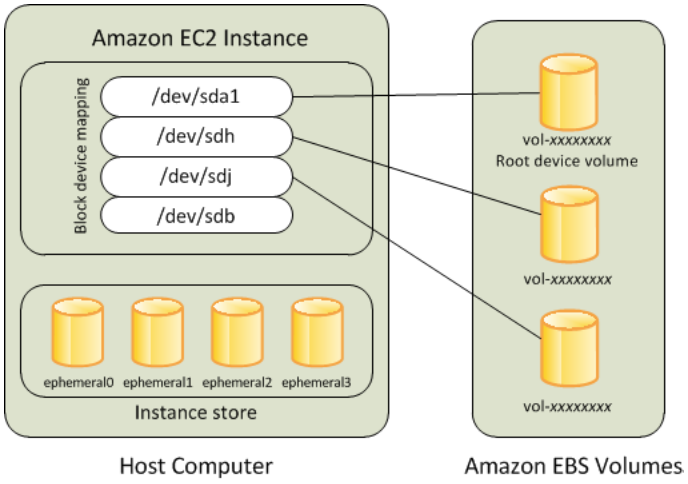


인스턴스 스토어가 지원하는 인스턴스는 종료되거나 장애가 발생할 경우 복원이 불가능합니다. Amazon EC2 인스턴스 스토어가 지원하는 인스턴스를 사용하려는 경우 여러 가용 영역의 인스턴스 스토어로 데이터를 분산하는 것이 좋습니다. 또한 인스턴스 스토어 볼륨의 중요한 데이터를 정기적으로 영구 스토리지로 백업해야 합니다.

자세한 내용은 Amazon EC2 인스턴스 스토어 (p. 1027) 단원을 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 중지한 후 다시 시작해도 연결된 볼륨에 저장된 데이터에 아무런 영향이 없습니다. Amazon EBS 지원 인스턴스가 중지 상태일 때 다양한 인스턴스 및 볼륨 관련 작업을 수행할 수 있습니다. 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. 112) 단원을 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. 890) 단원을 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 AMI를 선택합니다.
3. 필터 목록에서 퍼블릭 이미지 등의 이미지 유형을 선택합니다. 검색 창에서 플랫폼을 선택하고 Amazon Linux 등의 운영 체제를 선택한 후 루트 디바이스 유형을 선택하고 EBS 이미지를 선택합니다.
4. (선택 사항) 결정에 도움이 되는 추가 정보를 얻으려면 열 표시/숨기기 아이콘을 선택하고 표시할 열을 업데이트한 후 닫기를 선택합니다.
5. AMI를 선택하고 AMI ID를 메모해 둡니다.

콘솔을 사용하여 인스턴스 스토어 지원 AMI를 선택하려면 다음을 수행합니다.

1. Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 AMI를 선택합니다.
3. 필터 목록에서 퍼블릭 이미지 등의 이미지 유형을 선택합니다. 검색 창에서 플랫폼을 선택하고 Amazon Linux 등의 운영 체제를 선택한 후 루트 디바이스 유형을 선택하고 인스턴스 스토어를 선택합니다.
4. (선택 사항) 결정에 도움이 되는 추가 정보를 얻으려면 열 표시/숨기기 아이콘을 선택하고 표시할 열을 업데이트한 후 닫기를 선택합니다.
5. AMI를 선택하고 AMI ID를 메모해 둡니다.

명령줄을 사용하여 AMI의 루트 디바이스 볼륨 유형을 확인하려면 다음을 수행합니다.

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

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

인스턴스의 루트 디바이스 유형 확인

콘솔을 사용해 인스턴스의 루트 디바이스 유형을 확인하는 방법

1. Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택하고 인스턴스를 선택합니다.
3. 아래를 참고하여 설명 탭의 루트 디바이스 유형 값을 확인합니다.
   - 값이 `ebs`이면 Amazon EBS 지원 인스턴스입니다.

- 값이 `instance store`이면 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) 단원을 참조하십시오.

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

루트 디바이스 볼륨 지속 설정

기본적으로 Amazon EBS에서 지원하는 AMI의 루트 디바이스 볼륨은 인스턴스 종료 시 삭제됩니다. 인스턴스가 종료된 후에도 볼륨이 지속되도록 기본 동작을 변경할 수 있습니다. 기본 동작을 변경하려면 블록 디바이스 매핑을 사용하여 `DeleteOnTermination` 속성을 `false`로 설정합니다.

주제

- 인스턴스 시작 중 루트 볼륨이 지속되도록 구성 (p. 19)
- 실행 중인 인스턴스에 대해 지속되도록 루트 볼륨 구성 (p. 20)
- 루트 볼륨이 지속되도록 구성되었는지 확인 (p. 20)

인스턴스 시작 중 루트 볼륨이 지속되도록 구성

Amazon EC2 콘솔이나 명령줄 도구를 사용하여 인스턴스를 시작할 때 루트 볼륨이 유지되도록 구성할 수 있습니다.

콘솔을 사용하여 인스턴스를 시작할 때 루트 볼륨이 지속되도록 구성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택한 후 인스턴스 시작을 선택합니다.
3. Amazon 머신 이미지(AMI) 선택 페이지에서 사용할 AMI를 선택하고 선택을 선택합니다.
4. 마법사 안내에 따라 인스턴스 유형 선택 및 인스턴스 세부 정보 구성 설정을 완료합니다.
5. 스토리지 추가 페이지에서 루트 볼륨에 대한 종료 시 삭제의 선택을 해제합니다.
6. 나머지 마법사 페이지를 완료한 후 시작을 선택합니다.

AWS CLI를 사용하여 인스턴스를 시작할 때 루트 볼륨이 지속되도록 구성하려면

run-instances 명령을 사용하고 `DeleteOnTermination` 속성을 `false`로 설정하는 블록 디바이스 매핑을 포함합니다.

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

`mapping.json`에서 다음을 지정합니다.

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

```
]
```

Windows PowerShell용 도구을 사용하여 인스턴스를 시작할 때 루트 볼륨이 지속되도록 구성하려면

New-EC2Instance 명령을 사용하고 `DeleteOnTermination` 속성을 `false`로 설정하는 블록 디바이스 매핑을 포함합니다.

```
C:\> $ebs = New-Object Amazon.EC2.Model.EbsBlockDevice
C:\> $ebs.DeleteOnTermination = $false
C:\> $bdm = New-Object Amazon.EC2.Model.BlockDeviceMapping
C:\> $bdm.DeviceName = "dev/xvda"
C:\> $bdm.Ebs = $ebs
C:\> New-EC2Instance -ImageId ami-0abcdef1234567890 -BlockDeviceMapping $bdm ...other
 parameters...
```

실행 중인 인스턴스에 대해 지속되도록 루트 볼륨 구성

명령줄 도구만 사용하여 실행 중인 인스턴스에 대해 루트 볼륨이 지속되도록 구성할 수 있습니다.

AWS CLI를 사용하여 실행 중인 인스턴스에 대해 루트 볼륨이 지속되도록 구성하려면

modify-instance-attribute 명령을 사용하고 `DeleteOnTermination` 속성을 `false`로 설정하는 블록 디바이스 매핑을 포함합니다.

```
$ aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --block-device-
mappings "[{\"DeviceName\": \"/dev/xvda\",\"Ebs\":{\"DeleteOnTermination\":false}}]"
```

Windows PowerShell용 AWS 도구을 사용하여 실행 중인 인스턴스에 대해 루트 볼륨이 지속되도록 구성하려면

Edit-EC2InstanceAttribute 명령을 사용하고 `DeleteOnTermination` 속성을 `false`로 설정하는 블록 디바이스 매핑을 포함합니다.

```
C:\> $ebs = New-Object Amazon.EC2.Model.EbsInstanceBlockDeviceSpecification
C:\> $ebs.DeleteOnTermination = $false
C:\> $bdm = New-Object Amazon.EC2.Model.InstanceBlockDeviceMappingSpecification
C:\> $bdm.DeviceName = "/dev/xvda"
C:\> $bdm.Ebs = $ebs
C:\> Edit-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -BlockDeviceMapping $bdm
```

루트 볼륨이 지속되도록 구성되었는지 확인

Amazon EC2 콘솔이나 명령줄 도구를 사용하여 루트 볼륨이 지속되도록 구성되어 있는지 확인할 수 있습니다.

Amazon EC2 콘솔을 사용하여 루트 볼륨이 지속되도록 구성되어 있는지 확인하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택한 후 해당 인스턴스를 선택합니다.
3. 설명 탭에서 루트 디바이스 항목을 선택합니다. 종료 시 삭제가 `false`로 설정되어 있으면 볼륨이 지속되도록 구성된 것입니다.

AWS CLI를 사용하여 루트 볼륨이 지속되도록 구성되었는지 확인하려면

describe-instances 명령을 사용하여 `BlockDeviceMappings` 응답 요소의 `DeleteOnTermination` 속성이 `false`로 설정되어 있는지 확인합니다.

```
$ aws ec2 describe-instances --instance-id i-1234567890abcdef0
```

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

Windows PowerShell용 AWS 도구을 사용하여 루트 볼륨이 지속되도록 구성되었는지 확인하려면

Get-EC2Instance를 사용하여 `BlockDeviceMappings` 응답 요소의 `DeleteOnTermination` 속성이 `false`로 설정되어 있는지 확인합니다.

```
C:\> (Get-EC2Instance -InstanceId i-i-1234567890abcdef0).Instances.BlockDeviceMappings.Ebs
```

Amazon EC2로 설정

Amazon Web Services(AWS)에 이미 가입한 경우 Amazon EC2를 즉시 사용할 수 있습니다. Amazon EC2 콘솔을 열고 인스턴스 시작을 선택한 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. 22)
2. 키 페어 생성 (p. 22)
3. 보안 그룹 생성 (p. 24)

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 정보는 AWS 프리 티어 단원을 참조하십시오.

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

AWS 계정을 만들려면 다음을 수행합니다.

1. https://portal.aws.amazon.com/billing/signup을 엽니다.
2. 온라인 지시 사항을 따릅니다.

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

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, 가용 영역 및 로컬 영역 (p. 6) 단원을 참조하십시오.

키 페어를 생성하려면

1. AWS Management 콘솔에 로그인한 다음 https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 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 동부(오하이오) 리전에서 인스턴스를 시작하려면 미국 동부(오하이오) 리전에서 인스턴스에 대한 키 페어를 생성해야 합니다.

3. 탐색 창의 NETWORK & SECURITY에서 Key Pairs를 선택합니다.
4. Create key pair(키 페어 생성)를 선택합니다.
5. 해결 방법:

a. 이름에 새 키 페어에 대한 설명 이름(예: 사용자 이름, 뒤에 `-key-pair` 및 리전 이름)을 입력합니다. 예를 들어, me-key-pair-useast2로 지정할 수 있습니다.
b. 파일 형식에서 프라이빗 키를 저장할 형식을 선택합니다.
   - OpenSSH에서 사용되는 형식으로 프라이빗 키를 저장하려면 pem을 선택합니다.
   - Windows에서 Linux 인스턴스에 연결할 수 있는 도구인 PuTTY와 함께 사용되는 형식으로 프라이빗 키를 저장하려면 ppk를 선택합니다.
c. 생성을 선택합니다.

6. 브라우저에서 프라이빗 키 파일이 자동으로 다운로드됩니다. 기본 파일 이름은 키 페어의 이름으로 지정된 이름이며, 파일 이름 확장명은 `.pem`입니다. 안전한 장소에 프라이빗 키 파일을 저장합니다.

   Important

   이때가 사용자가 프라이빗 키 파일을 저장할 수 있는 유일한 기회입니다. 인스턴스를 시작할 때 키 페어의 이름을 제공하고, 인스턴스에 연결할 때마다 해당 프라이빗 키를 제공해야 합니다.

7. 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
```

   이러한 권한을 설정하지 않으면 이 키 페어를 사용하여 인스턴스에 연결할 수 없습니다. 자세한 내용은 오류: 보호되지 않는 프라이빗 키 파일 (p. 1090) 단원을 참조하십시오.

자세한 내용은 Amazon EC2 키 페어 (p. 861) 단원을 참조하십시오.

보안 그룹 생성

보안 그룹은 연결된 인스턴스에 대한 방화벽 역할을 하여 인스턴스 수준에서 인바운드 트래픽과 아웃바운드 트래픽을 모두 제어합니다. SSH을 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, 가용 영역 및 로컬 영역 (p. 6)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 Check IP 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 단원에서 Amazon VPC 콘솔로 전환한 경우 Amazon EC2 콘솔로 다시 전환하고 이 지침을 사용하거나, Amazon VPC 시작 안내서의 VPC에 대한 보안 그룹 설정에 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 보안 그룹을 선택합니다.
4. 보안 그룹 생성을 선택합니다.
5. 새 보안 그룹의 이름과 설명을 입력합니다. 기억하기 쉬운 이름을 사용합니다(예: 사용자 이름, 뒤에 _SG_ 및 리전 이름). 예를 들어, me_SG_uswest2로 지정할 수 있습니다.

6. VPC 목록에서 리전의 기본 VPC를 선택합니다.
7. 인바운드 탭에서 다음 규칙을 생성한 다음(각 새 규칙에 대해 규칙 추가 선택) 생성을 선택합니다.
   - 유형 목록에서 HTTP를 선택하고 소스가 위치 무관(`0.0.0.0/0`)으로 설정되어 있는지 확인합니다.
   - 유형 목록에서 HTTPS를 선택하고 소스가 위치 무관(`0.0.0.0/0`)으로 설정되어 있는지 확인합니다.
   - 유형 목록에서 SSH, 를 선택합니다. 필드를 로컬 컴퓨터의 퍼블릭 IPv4 주소로 자동으로 채우려면 소스 상자에서 내 IP를 선택하면 됩니다. 또는 사용자 지정을 선택하고 컴퓨터 또는 네트워크의 퍼블릭 IPv4 주소를 CIDR 표기법으로 지정해도 됩니다. 개별 IP 주소를 CIDR 표기법으로 지정하려면 라우팅 접미사 `/32`를 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 Amazon EC2 인스턴스에 대한 Linux 보안 그룹 (p. 872) 단원을 참조하십시오.

Amazon EC2 Linux 인스턴스 시작하기

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

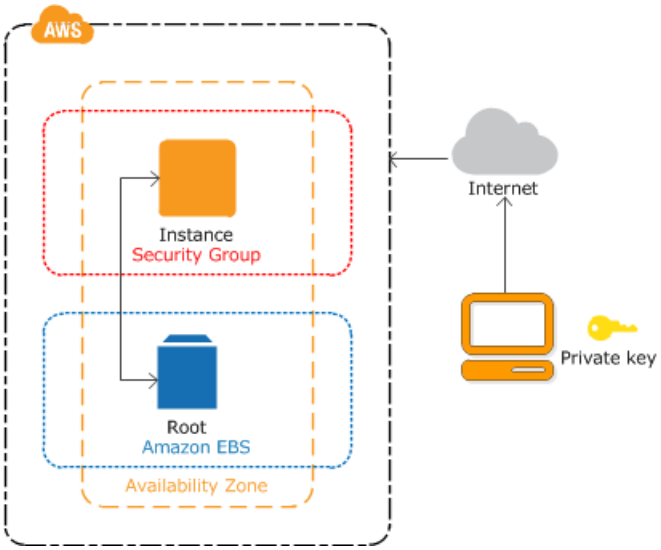
AWS에 가입할 때 AWS 프리 티어를 사용하여 Amazon EC2 사용을 시작할 수 있습니다. 12개월 이전에 AWS 계정을 생성했지만 Amazon EC2에 대한 프리 티어 혜택을 아직 다 사용하지 않은 경우 프리 티어 혜택 안에 포함된 옵션을 선택하는 데 도움이 되는 이 자습서를 무료로 이용할 수 있습니다. 그렇지 않을 경우, 유휴 상태로 유지되더라도 인스턴스를 시작하는 시점부터 인스턴스를 종료할 때까지(이 자습서의 최종 작업) 스탠다드 Amazon EC2 사용료가 발생합니다.

내용

- 개요 (p. 27)
- 사전 조건 (p. 28)
- 1단계: 인스턴스 시작 (p. 28)
- 2단계: 인스턴스에 연결 (p. 29)
- 3단계: 인스턴스 정리 (p. 29)
- 다음 단계 (p. 29)

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. 28)
2. 인스턴스에 연결 (p. 29)
3. 인스턴스 정리 (p. 29)

관련 자습서

- Windows 인스턴스를 시작하려면 이 자습서의 Windows 인스턴스용 Amazon EC2 사용 설명서: Amazon EC2 Windows 인스턴스 시작하기 단원을 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. 22)의 단계를 완료해야 합니다.

1단계: 인스턴스 시작

다음 절차의 설명에 따라 Linux을 사용하여 AWS Management 콘솔 인스턴스를 시작할 수 있습니다. 이 자습서는 첫 번째 인스턴스를 빠르게 시작하도록 돕기 위한 것이므로 가능한 모든 옵션을 다루지는 않습니다. 고급 옵션에 대한 자세한 정보는 인스턴스 시작하기 단원을 참조하십시오.

인스턴스를 시작하려면 다음을 수행합니다.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 콘솔 대시보드에서 인스턴스 시작을 선택합니다.
3. Amazon Machine Image(AMI) 선택 페이지에 인스턴스에 대한 템플릿 역할을 하는 Amazon Machine Image(AMI)라는 기본 구성 목록이 표시됩니다. Amazon Linux 2의 HVM 버전을 선택합니다. 해당되는 AMI는 "프리 티어 사용 가능"으로 표시됩니다.
4. 인스턴스 유형 선택 페이지에서 인스턴스의 하드웨어 구성을 선택할 수 있습니다. 기본적으로 선택된 `t2.micro` 유형을 선택합니다. 이 인스턴스 유형은 프리 티어에 적격입니다.
5. 검토 후 시작(Review and Launch)을 선택하여 마법사가 다른 구성 설정을 완료하게 합니다.
6. 인스턴스 시작 검토 페이지의 보안 그룹 아래에서 마법사가 보안 그룹을 만들고 선택했음을 확인합니다. 이 보안 그룹을 사용하거나, 다음 단계를 이용하여 설정을 시작할 때 만든 보안 그룹을 선택합니다.

   a. 보안 그룹 편집을 선택합니다.
   b. 보안 그룹 구성 페이지에서 Select an existing security group(기존 보안 그룹 선택)이 선택되어 있는지 확인합니다.
   c. 기존 보안 그룹 목록에서 보안 그룹을 선택한 다음 검토 후 시작(Review and Launch)을 선택합니다.
7. 인스턴스 시작 검토 페이지에서 시작을 선택합니다.
8. 키 페어에 대한 메시지가 나타나면 기존 키 페어 선택을 선택한 다음 설치할 때 생성한 키 페어를 선택합니다.

   또는 키 페어를 새로 만들 수 있습니다. 새 키 페어 생성을 선택하고 키 페어 이름을 입력한 다음 키 페어 다운로드를 선택합니다. 이때가 사용자가 프라이빗 키 파일을 저장할 수 있는 유일한 기회이므로 반드시 다운로드하십시오. 프라이빗 키 파일은 안전한 장소에 저장합니다. 인스턴스를 시작할 때 키 페어의 이름을 제공하고, 인스턴스에 연결할 때마다 해당 프라이빗 키를 제공해야 합니다.

Warning

키 페어 없이 계속 옵션을 선택하지 마십시오. 키 페어 없이 인스턴스를 시작하면 인스턴스에 연결할 수 없습니다.

준비되면 승인 확인란을 선택한 다음 인스턴스 시작을 선택합니다.

9. 확인 페이지에서 인스턴스가 실행 중인지 확인할 수 있습니다. 인스턴스 보기를 선택하여 확인 페이지를 닫고 콘솔로 돌아갑니다.
10. 인스턴스 화면에서 시작 상태를 볼 수 있습니다. 인스턴스를 시작하는 데 약간 시간이 걸립니다. 인스턴스를 시작할 때 초기 상태는 `pending`입니다. 인스턴스가 시작된 후에는 상태가 `running`으로 바뀌고 퍼블릭 DNS 이름을 받습니다. (퍼블릭 DNS(IPv4) 열이 숨겨져 있는 경우 페이지 오른쪽 상단 모서리에 있는 열 표시/숨기기(기어 모양 아이콘)를 선택한 다음 퍼블릭 DNS(IPv4)를 선택합니다.)
11. 연결할 수 있도록 인스턴스가 준비될 때까지 몇 분 정도 걸릴 수 있습니다. 인스턴스가 상태 확인을 통과했는지 확인하십시오. 상태 검사 열에서 이 정보를 볼 수 있습니다.

2단계: 인스턴스에 연결

Linux 인스턴스에 연결하는 몇 가지 방법이 있습니다. 자세한 내용은 Linux 인스턴스에 연결 (p. 485) 단원을 참조하십시오.

Important

`.pem` 파일이 있는 키 페어를 사용하여 시작했거나 컴퓨터에서 SSH 액세스를 허용하는 보안 그룹을 사용하여 인스턴스를 시작한 경우가 아니면 인스턴스에 연결할 수 없습니다. 인스턴스에 연결할 수 없는 경우 지원이 필요하면 인스턴스 연결 문제 해결 (p. 1085)을 참조하십시오.

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. 29) 단원을 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 정보는 인스턴스 중단을 참조하십시오.

인스턴스를 종료하려면

1. 탐색 창에서 인스턴스를 선택합니다. 인스턴스 목록에서 인스턴스를 선택합니다.
2. 작업, 인스턴스 상태, 종료를 차례로 선택합니다.
3. 확인 메시지가 나타나면 예, 종료를 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 AWS 시스템 관리자 사용 설명서의 AWS 시스템 관리자 Run Command를 참조하십시오.
- 사용량이 프리 티어 한도를 초과하는 경우 알려 주는 CloudWatch 경보를 구성합니다. 자세한 내용은 AWS Billing and Cost Management 사용 설명서의 청구 경보 생성을 참조하십시오.
- EBS 볼륨을 추가합니다. 자세한 내용은 Amazon EBS 볼륨 생성 (p. 908) 및 Amazon EBS 볼륨을 인스턴스에 연결 (p. 910) 단원을 참조하십시오.
- LAMP 스택을 설치합니다. 자세한 내용은 자습서: Amazon Linux 2에 LAMP 웹 서버 설치 (p. 32) 단원을 참조하십시오.

Amazon EC2 모범 사례

이 모범 사례 목록을 사용하면 Amazon EC2의 이점을 최대한으로 활용할 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 사례를 참조하십시오.
- 보안 그룹에 대한 최소 허용 규칙을 구현합니다. 자세한 내용은 보안 그룹 규칙 (p. 873) 단원을 참조하십시오.
- 인스턴스에서 운영 체제와 애플리케이션을 정기적으로 패치, 업데이트 및 보안합니다. Amazon Linux 2 또는 Amazon Linux AMI 업데이트에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 Linux 인스턴스의 소프트웨어 관리를 참조하십시오.

스토리지

- 루트 디바이스 유형이 데이터 지속성, 백업 및 복구에 미치는 영향을 이해합니다. 자세한 내용은 루트 디바이스 스토리지 (p. 93) 단원을 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. 526) 단원을 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 내결함성을 보장하는 복제 인자를 가진 클러스터가 있어야 합니다.
- EBS 볼륨 및 스냅샷을 암호화합니다. 자세한 내용은 Amazon EBS Encryption (p. 968) 단원을 참조하십시오.

리소스 관리

- 인스턴스 메타데이터 및 사용자 지정 리소스 태그를 사용하여 AWS 리소스를 추적하고 식별합니다. 자세한 내용은 인스턴스 메타데이터 및 사용자 데이터 (p. 571) 및 Amazon EC2리소스에 태그 지정 (p. 1068) 단원을 참조하십시오.
- Amazon EC2에 대한 현재 제한을 조회합니다. 실제로 필요할 시점보다 미리 제한 증가를 요청하도록 계획하십시오. 자세한 내용은 Amazon EC2 서비스 제한 (p. 1079) 단원을 참조하십시오.

백업 및 복구

- Amazon EBS 스냅샷 (p. 926)을 사용하여 EBS 볼륨을 정기적으로 백업하고, 인스턴스에서 Amazon 머신 이미지(AMI) (p. 91)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. 658) 단원을 참조하십시오.
- 이벤트 모니터링 및 응답. 자세한 내용은 Amazon EC2 모니터링 (p. 600) 단원을 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. 683) 단원을 참조하십시오. 자동 솔루션의 경우 Amazon EC2 Auto Scaling을 사용할 수 있습니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서를 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. 동영상은 AWS 설명 비디오 및 실습을 참조하십시오.

자습서

- 자습서: Amazon Linux 2에 LAMP 웹 서버 설치 (p. 32)
- 자습서: Amazon Linux AMI를 사용하여 LAMP 웹 서버 설치 (p. 41)
- 자습서: Amazon Linux를 통한 WordPress 블로그 호스팅 (p. 52)
- 자습서: Amazon Linux 2에서 SSL/TLS 구성 (p. 60)
- 자습서: Amazon Linux에서 SSL/TLS 구성 (p. 74)
- 자습서: Amazon EC2에서 애플리케이션의 가용성 향상 (p. 87)

자습서: Amazon Linux 2에 LAMP 웹 서버 설치

다음 절차를 통해 Amazon Linux 2 인스턴스에서 PHP 및 MariaDB(커뮤니티에서 개발한 MySQL 포크) 지원을 포함하는 Apache 웹 서버를 설치할 수 있습니다. 이 웹 서버를 LAMP 웹 서버 또는 LAMP 스택이라고도 합니다. 이 서버를 사용해서 고정 웹사이트를 호스팅하거나 데이터베이스에서 정보를 읽고 쓰는 동적 PHP 애플리케이션을 배포할 수 있습니다.

> **Important**
>
> Amazon Linux AMI에서 LAMP 웹 서버를 설정하려면 자습서: Amazon Linux AMI를 사용하여 LAMP 웹 서버 설치 (p. 41) 단원을 참조하십시오.
> LAMP 웹 서버를 Ubuntu 또는 Red Hat Enterprise Linux 인스턴스에서 설치하려는 경우는 본 자습서를 이용할 수 없습니다. 기타 배포에 대한 자세한 내용은 해당 설명서를 참조하십시오. Ubuntu의 LAMP 웹 서버에 대한 자세한 내용은 Ubuntu 커뮤니티 문서 ApacheMySQLPHP 항목을 참조하십시오.

옵션: 자동화를 사용하여 이 자습서 완료

다음 작업 대신 AWS 시스템 관리자 자동화를 사용하여 이 자습서를 완료하려면 AWSDocs-InstallALAMPServer-AL2 자동화 문서를 실행합니다.

작업

- 1단계: LAMP 서버 준비 (p. 32)
- 2단계: LAMP 서버 테스트 (p. 36)
- 3단계: 데이터베이스 서버 보안 설정 (p. 37)
- 4단계: (선택 사항) phpMyAdmin 설치 (p. 38)
- 문제 해결 (p. 41)
- 관련 주제 (p. 41)

1단계: LAMP 서버 준비

사전 조건

본 자습서는 Amazon Linux 2를 사용하여 사용자가 인터넷에서 접근할 수 있는 퍼블릭 DNS 이름을 가진 새 인스턴스를 이미 실행한 것으로 가정하고 있습니다. 자세한 내용은 1단계: 인스턴스 시작 (p. 28) 단원을 참조

하십시오. SSH(포트 22), HTTP(포트 80), HTTPS(포트 443) 연결을 허용할 수 있도록 보안 그룹을 구성해야 합니다. 이 사전 요구 사항에 대한 자세한 내용은 Linux 인스턴스의 인바운드 트래픽 권한 부여 (p. 859) 단원을 참조하십시오.

Note

다음 절차는 Amazon Linux 2에서 사용할 수 있는 최신 PHP 버전(PHP 7.2)을 설치합니다. 이 자습서에서 명시한 애플리케이션이 아닌 PHP 애플리케이션을 사용하려는 경우 PHP 7.2와의 호환성을 확인해야 합니다.

LAMP 서버를 준비하려면

1. 인스턴스에 연결합니다 (p. 29).
2. 모든 소프트웨어 패키지가 최신 상태로 업데이트되어 있는지 확인하기 위해, 인스턴스에서 퀵 소프트웨어 업데이트를 실행합니다. 이 업데이트 과정은 몇 분 정도 시간이 소요될 수 있지만, 최신 보안 업데이트와 버그 수정을 위해 수행할 필요가 있습니다.

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

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

3. `lamp-mariadb10.2-php7.2` 및 `php7.2` Amazon Linux Extras 리포지토리를 설치하여 Amazon Linux 2용 LAMP MariaDB 및 PHP 패키지의 최신 버전을 가져옵니다.

   ```
   [ec2-user ~]$ sudo amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
   ```

   Note

   `sudo: amazon-linux-extras: command not found` 오류가 발생하면 인스턴스가 Amazon Linux 2 AMI로 실행되지 않은 것입니다(Amazon Linux AMI를 사용하고 있는 것일 수 있음). 다음 명령을 사용하여 Amazon Linux 버전을 볼 수 있습니다.

   ```
   cat /etc/system-release
   ```

   Amazon Linux AMI에서 LAMP 웹 서버를 설정하려면 자습서: Amazon Linux AMI를 사용하여 LAMP 웹 서버 설치 (p. 41) 단원을 참조하십시오.
4. 이제 인스턴스가 최신 상태이므로 Apache 웹 서버, MariaDB, PHP 소프트웨어 패키지를 설치할 수 있습니다.

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

   ```
   [ec2-user ~]$ sudo yum install -y httpd mariadb-server
   ```

   Note

   다음 명령을 사용하여 이러한 패키지의 현재 버전을 볼 수 있습니다.

   ```
   yum info package_name
   ```

5. Apache 웹 서버를 시작합니다.

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

6. systemctl 명령을 사용하여 Apache 웹 서버가 매번 시스템이 부팅할 때마다 시작되도록 합니다.

```
[ec2-user ~]$ sudo systemctl enable httpd
```

다음 명령을 실행하여 httpd가 실행되고 있는지 확인할 수 있습니다.

```
[ec2-user ~]$ sudo systemctl is-enabled httpd
```

7. 인스턴스에 대해 인바운드 HTTP(포트 80) 연결을 허용하는 보안 규칙이 없는 경우 추가합니다. 기본적으로 초기화 중에 인스턴스에 대해 launch-wizard-*N* 보안 그룹이 설정됩니다. 이 그룹에는 SSH 연결을 허용하는 규칙이 한 개 들어 있습니다.

   a. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
   b. 인스턴스를 선택하고 해당 인스턴스를 선택합니다.
   c. 보안 그룹에서 인바운드 규칙 보기를 선택합니다.
   d. 기본 보안 그룹에는 다음과 같은 규칙 목록이 있습니다.

```
Security Groups associated with i-1234567890abcdef0
Ports     Protocol     Source     launch-wizard-N
22        tcp          0.0.0.0/0          #
```

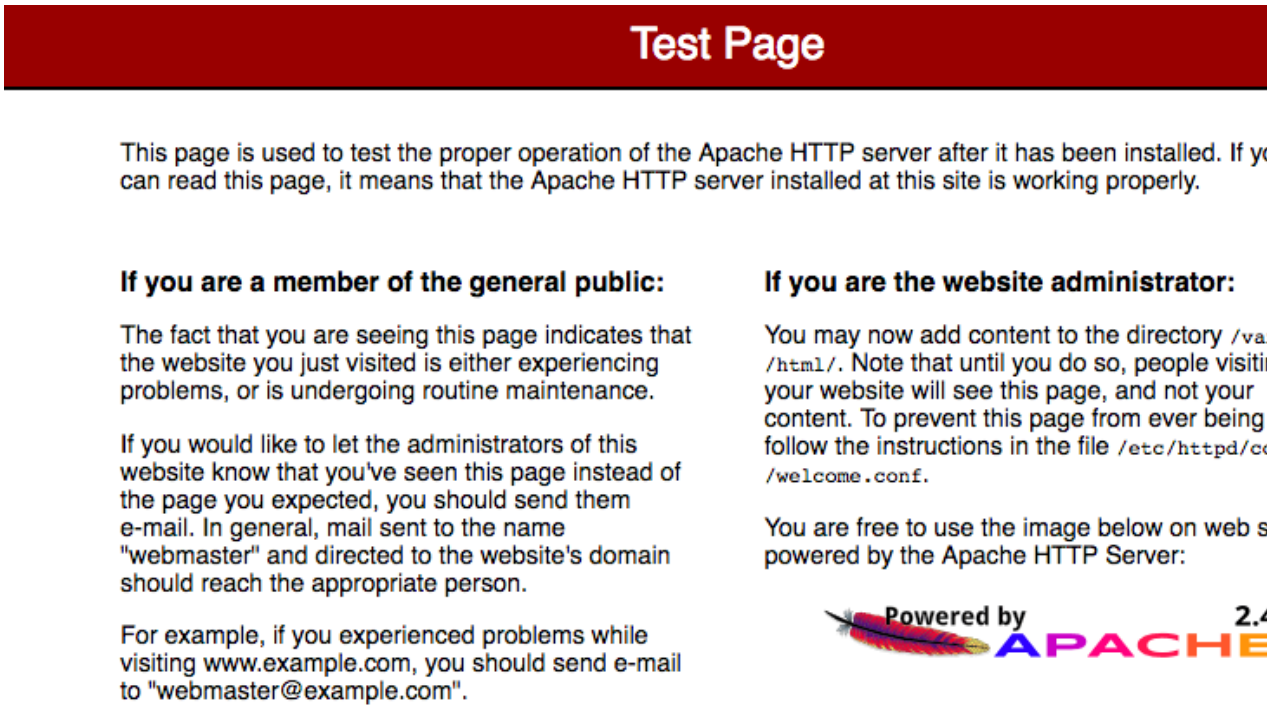
   보안 그룹에 규칙 추가 (p. 877)의 절차를 사용하여 다음 값으로 새로운 인바운드 보안 규칙을 추가합니다.

   - 유형: HTTP
   - 프로토콜: TCP
   - 포트 범위: 80
   - 소스: 사용자 지정

8. 웹 서버를 테스트합니다. 웹 브라우저에서 인스턴스의 공용 DNS 주소(또는 공용 IP 주소)를 입력합니다. `/var/www/html`에 콘텐츠가 없으면 Apache 테스트 페이지가 표시됩니다. Amazon EC2 콘솔을 사용하여 인스턴스의 퍼블릭 DNS를 확인할 수 있습니다. 퍼블릭 DNS 열을 확인합니다. 이 열이 숨겨진 경우 열 표시/숨기기 아이콘(기어 모양 아이콘)을 선택하고 퍼블릭 DNS를 선택합니다.

   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. 877))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.



Apache httpd는 Apache document root라는 디렉터리에 보관된 파일을 처리합니다. Amazon Linux Apache 문서 루트는 `/var/www/html`이며, 기본적으로 루트에서 소유합니다.

`ec2-user` 계정에서 이 디렉터리의 파일을 조작할 수 있게 하려면 디렉터리의 소유권과 권한을 변경해야 합니다. 이 작업을 수행하는 방법에는 여러 가지가 있습니다. 본 자습서에서는 `ec2-user`를 `apache` 그룹에 추가하여 `apache` 그룹에 `/var/www` 디렉터리의 소유권을 부여하고 쓰기 권한을 할당합니다.

파일 권한 설정

1. 사용자(이 경우는 `ec2-user`)를 `apache` 그룹에 추가합니다.

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

2. 로그아웃하고 다시 로그인한 다음, 새 그룹을 선택하고 멤버십을 확인합니다.

   a. 로그아웃합니다(exit 명령을 사용하거나 터미널 창 닫기).

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

   b. `apache` 그룹의 멤버십을 확인하려면 인스턴스에 다시 연결한 후 다음 명령을 실행합니다.

```
[ec2-user ~]$ groups
ec2-user adm wheel apache systemd-journal
```

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

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

4. 그룹 쓰기 권한을 추가하여 나중에 하위 디렉터리에 대한 그룹 ID를 설정하려면 `/var/www`와 그 하위 디렉터리의 디렉터리 권한을 변경합니다.

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

5. 그룹 쓰기 권한을 추가하려면 `/var/www` 및 그 하위 디렉터리의 파일 권한을 반복하여 변경합니다.

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

이제 `ec2-user`와 `apache` 그룹의 향후 멤버는 Apache document root에서 파일 추가, 삭제, 편집을 할 수 있고, 이를 통해 사용자는 정적 웹 사이트 또는 PHP 애플리케이션과 같은 콘텐츠를 추가할 수 있습니다.

웹 서버를 보호하려면(선택 사항)

HTTP 프로토콜을 실행하는 웹 서버는 송신하거나 수신하는 데이터에 대해 아무런 전송 보안 기능도 제공하지 않습니다. 웹 브라우저를 사용하여 HTTP 서버에 연결할 때 방문하는 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 활성화하는 방법에 대한 자세한 내용은 자습서: Amazon Linux 2에서 SSL/TLS 구성 (p. 60) 단원을 참조하십시오.

2단계: LAMP 서버 테스트

서버가 설치되어 실행 중이고 파일 권한이 올바르게 설정되었다면 사용자의 `ec2-user` 계정을 통해 인터넷에서 사용 가능한 `/var/www/html` 디렉터리에서 PHP 파일을 생성할 수 있어야 합니다.

LAMP 서버를 테스트하려면

1. Apache 문서 루트에서 PHP 파일을 생성합니다.

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

   이 명령을 실행하는 동안 "Permission denied" 오류가 발생하면, 로그아웃하고 다시 로그인한 다음, 파일 권한 설정 (p. 35)에서 구성한 적절한 그룹 권한을 선택합니다.

2. 웹 브라우저에서는 방금 생성한 파일의 URL을 입력합니다. 이 URL은 인스턴스의 퍼블릭 DNS 주소에 슬래시(/)와 파일 이름이 추가된 형태입니다. 다음 예를 참조하십시오.

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

   PHP 정보 페이지가 표시되어야 합니다:

PHP Version 7.2.0

System	Linux ip-172-31-22-15.us-west-2.compute.internal 4.9.62-10.57.amzn2.x86_ 2017 x86_64
Build Date	Dec 13 2017 03:34:37
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.d
Additional .ini files parsed	/etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/p /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gettext.ini, /etc/ph /etc/php.d/20-mysqlnd.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/p sqlite3.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30- pdo_sqlite.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS

Note

이 페이지가 보이지 않을 경우 이전 단계에서 `/var/www/html/phpinfo.php` 파일이 제대로 생성되었는지 확인하십시오. 또한 다음 명령을 사용하여 필수 패키지가 모두 설치되었는지도 확인할 수 있습니다.

```
[ec2-user ~]$ sudo yum list installed httpd mariadb-server php-mysqlnd
```

출력에서 필요한 패키지가 하나라도 나열되지 않으면, sudo yum install ***package*** 명령을 사용하여 패키지를 설치합니다. `php7.2` 및 `lamp-mariadb10.2-php7.2` extra가 amazon-linux-extras 명령의 출력에서 활성화되는지도 확인합니다.

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

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

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

3단계: 데이터베이스 서버 보안 설정

MariaDB 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. MariaDB 서버를 사용할 계획이 없더라도 이 절차를 수행하는 것이 좋습니다.

MariaDB 서버의 보안을 유지하려면

1. MariaDB 서버를 시작합니다.

```
[ec2-user ~]$ sudo systemctl start mariadb
```

2. 실행 mysql_secure_installation.

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

   a. 암호를 입력하라는 메시지가 표시되면 루트 계정의 암호를 입력합니다.

      i. 현재 루트 암호를 입력합니다. 기본적으로 root 계정에는 암호가 없습니다. Enter를 누릅니다.

      ii. 암호를 설정하려면 **Y**를 누른 후 안전한 암호를 두 번 입력합니다. 보안 암호 생성에 대한 자세한 내용은 https://identitysafe.norton.com/password-generator/ 단원을 참조하십시오. 이 암호를 안전한 장소에 보관하시기 바랍니다.

      > **Note**
      >
      > MariaDB에 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. **Y**를 눌러서 익명 사용자 계정을 제거합니다.

   c. **Y**를 입력하여 원격 루트 로그인을 비활성화합니다.

   d. **Y**를 눌러서 테스트 데이터베이스를 제거합니다.

   e. **Y**를 눌러서 권한 테이블을 다시 로드하고 변경사항을 저장합니다.

3. (선택 사항) 지금 바로 사용할 계획이 아니라면 MariaDB 서버를 중지합니다. 필요할 때 다시 시작할 수 있습니다.

```
[ec2-user ~]$ sudo systemctl stop mariadb
```

4. (선택 사항) 부팅 시 MariaDB 서버가 시작되도록 하려면 다음 명령을 입력합니다.

```
[ec2-user ~]$ sudo systemctl enable mariadb
```

4단계: (선택 사항) phpMyAdmin 설치

phpMyAdmin은 EC2 인스턴스의 MySQL 데이터베이스를 보고 편집하는 데 사용할 수 있는 웹 기반 데이터베이스 관리 도구입니다. Amazon Linux 인스턴스에서 `phpMyAdmin`을 설치 및 구성하려면 다음 단계를 따르십시오.

> **Important**
>
> Apache에서 SSL/TLS를 활성화하지 않은 경우 `phpMyAdmin`을 사용하여 LAMP 서버에 액세스하지 않는 것이 좋습니다. 이 상태에서 액세스하면 데이터베이스 관리자 암호와 기타 데이터가 인터넷을 통해 안전하지 못한 상태로 전송됩니다. 개발자의 보안 권장 사항을 보려면 phpMyAdmin 설치 보안을 참조하십시오. EC2 인스턴스에서의 웹 서버 보안에 대한 일반적인 정보는 자습서: Amazon Linux 2에서 SSL/TLS 구성 (p. 60) 단원을 참조하십시오.

phpMyAdmin을 설치하려면

1. 필요한 종속 항목을 설치합니다.

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

2. Apache를 다시 시작합니다.

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

3. `php-fpm`을 다시 시작합니다.

```
[ec2-user ~]$ sudo systemctl restart php-fpm
```

4. Apache 문서 루트(`/var/www/html`)로 이동합니다.

```
[ec2-user ~]$ cd /var/www/html
```

5. https://www.phpmyadmin.net/downloads에서 phpMyAdmin 최신 릴리스의 소스 패키지를 선택합니다. 인스턴스로 파일을 직접 다운로드하려면 다음 예제와 같이 링크를 복사한 후 wget 명령에 붙여 넣습니다.

```
[ec2-user html]$ wget https://www.phpmyadmin.net/downloads/phpMyAdmin-latest-all-
languages.tar.gz
```

6. `phpMyAdmin` 폴더를 생성하고 다음 명령을 사용하여 해당 폴더로 패키지의 압축을 풉니다.

```
[ec2-user html]$ mkdir phpMyAdmin && tar -xvzf phpMyAdmin-latest-all-languages.tar.gz -
C phpMyAdmin --strip-components 1
```

7. *`phpMyAdmin-latest-all-languages.tar.gz`* tarball을 삭제합니다.

```
[ec2-user html]$ rm phpMyAdmin-latest-all-languages.tar.gz
```

8. (선택 사항) MySQL 서버가 실행 중이지 않으면 지금 시작합니다.

```
[ec2-user ~]$ sudo systemctl start mariadb
```

9. 웹 브라우저에서 phpMyAdmin 설치의 URL을 입력합니다. 아래의 예와 같이 이 URL은 인스턴스의 퍼블릭 DNS 주소(또는 퍼블릭 IP 주소)에 슬래시(/)와 설치 디렉터리의 이름이 추가된 형태입니다. 다음 예를 참조하십시오.

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

사용자는 phpMyAdmin 로그인 페이지를 볼 수 있어야 합니다:

10. 앞서 만든 `root` 사용자 이름 및 MySQL 루트 암호로 phpMyAdmin 설치에 로그인합니다.

    작동하려면 먼저 설치를 구성해야 합니다. phpMyAdmin을 구성하려면 구성 파일을 직접 만들거나, 설정 콘솔을 사용하거나, 두 방법을 조합하여 사용합니다.

    phpMyAdmin에 대한 자세한 내용은 phpMyAdmin 사용 설명서를 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 ~]$ sudo systemctl is-enabled httpd
  ```

  httpd 프로세스가 실행되지 않는 경우 LAMP 서버를 준비하려면 (p. 33)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. 877))을 참조하십시오.

관련 주제

파일을 인스턴스에 전송하거나 웹 서버에 WordPress 블로그를 설치하는 것에 대한 자세한 다용은 다음 문서를 참고하십시오.

- WinSCP를 사용하여 Linux 인스턴스로 파일 전송 (p. 501)
- SCP를 사용하여 Linux에서 Linux 인스턴스로 파일 전송 (p. 488)
- 자습서: Amazon Linux를 통한 WordPress 블로그 호스팅 (p. 52)

이 자습서에서 사용되는 명령과 소프트웨어에 대한 자세한 내용은 다음 웹 페이지를 확인하십시오.

- Apache 웹 서버: http://httpd.apache.org/
- MariaDB 데이터베이스 서버: https://mariadb.org/
- 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 Amazon Route 53에서 도메인 및 하위 도메인 생성 및 마이그레이션을 참조하십시오.

자습서: Amazon Linux AMI를 사용하여 LAMP 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.

> **Important**
>
> Amazon Linux 2에서 LAMP 웹 서버를 설정하려면 자습서: Amazon Linux 2에 LAMP 웹 서버 설치 (p. 32) 단원을 참조하십시오.
> LAMP 웹 서버를 Ubuntu 또는 Red Hat Enterprise Linux 인스턴스에서 설치하려는 경우는 본 자습서를 이용할 수 없습니다. 기타 배포에 대한 자세한 내용은 해당 설명서를 참조하십시오. Ubuntu의 LAMP 웹 서버에 대한 자세한 내용은 Ubuntu 커뮤니티 문서 ApacheMySQLPHP 항목을 참조하십시오.

옵션: 자동화를 사용하여 이 자습서 완료

다음 작업 대신 AWS 시스템 관리자 자동화를 사용하여 이 자습서를 완료하려면 AWSDocs-InstallALAMPServer-AL 자동화 문서를 실행합니다.

작업

- 1단계: LAMP 서버 준비 (p. 42)
- 2단계: LAMP 서버 테스트 (p. 46)
- 3단계: 데이터베이스 서버 보안 설정 (p. 47)
- 4단계: (선택 사항) phpMyAdmin 설치 (p. 48)
- 문제 해결 (p. 51)
- 관련 주제 (p. 52)

1단계: LAMP 서버 준비

사전 조건

본 자습서는 Amazon Linux AMI를 사용하여 사용자가 인터넷에서 접근할 수 있는 퍼블릭 DNS 이름을 가진 새 인스턴스를 이미 실행한 것으로 가정합니다. 자세한 내용은 1단계: 인스턴스 시작 (p. 28) 단원을 참조하십시오. SSH(포트 22), HTTP(포트 80), HTTPS(포트 443) 연결을 허용할 수 있도록 보안 그룹을 구성해야 합니다. 이 사전 요구 사항에 대한 자세한 내용은 Linux 인스턴스의 인바운드 트래픽 권한 부여 (p. 859) 단원을 참조하십시오.

Amazon Linux AMI에서 LAMP 웹 서버를 설치 및 시작하려면

1. 인스턴스에 연결합니다 (p. 29).
2. 모든 소프트웨어 패키지가 최신 상태로 업데이트되어 있는지 확인하기 위해, 인스턴스에서 퀵 소프트웨어 업데이트를 실행합니다. 이 업데이트 과정은 몇 분 정도 시간이 소요될 수 있지만, 최신 보안 업데이트와 버그 수정을 위해 수행할 필요가 있습니다.

   `-y` 옵션을 사용하면 확인 여부를 묻지 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 애플리케이션이 패키지와 호환되는지 확인하십시오. 문제가 있는 경

우, 다른 환경을 설치해야 할 수 있습니다. 자세한 내용은 서버에서 실행할 애플리케이션 소프트웨어가 설치된 PHP 버전 또는 다른 소프트웨어와 호환되지 않습니다 (p. 51) 단원을 참조하십시오.

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

```
[ec2-user ~]$ sudo yum install -y httpd24 php72 mysql57-server php72-mysqlnd
```

Note

`No package package-name available` 오류가 발생하면 인스턴스가 Amazon Linux AMI로 실행되지 않은 것입니다(Amazon Linux 2를 사용하고 있는 것일 수 있음). 다음 명령을 사용하여 Amazon Linux 버전을 볼 수 있습니다.

```
cat /etc/system-release
```

4. Apache 웹 서버를 시작합니다.

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

5. chkconfig 명령을 사용하여 Apache 웹 서버가 매번 시스템이 부팅할 때마다 시작되도록 합니다.

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

chkconfig 명령을 사용하여 성공적으로 서비스를 활성화했을 경우에는 아무런 확인 메시지가 표시되지 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
```

여기서 httpd는 2, 3, 4, 5의 실행 레벨(사용자가 보려는 부분)에서 on 상태입니다.

6. 인스턴스에 대해 인바운드 HTTP(포트 80) 연결을 허용하는 보안 규칙이 없는 경우 추가합니다. 기본적으로 초기화 중에 인스턴스에 대해 launch-wizard-*N* 보안 그룹이 설정됩니다. 이 그룹에는 SSH 연결을 허용하는 규칙이 한 개 들어 있습니다.

   a. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
   b. 인스턴스를 선택하고 해당 인스턴스를 선택합니다.
   c. 보안 그룹에서 인바운드 규칙 보기를 선택합니다.
   d. 기본 보안 그룹에는 다음과 같은 규칙 목록이 있습니다.

```
Security Groups associated with i-1234567890abcdef0
Ports     Protocol     Source     launch-wizard-N
22        tcp          0.0.0.0/0          #
```

   보안 그룹에 규칙 추가 (p. 877)의 절차를 사용하여 다음 값으로 새로운 인바운드 보안 규칙을 추가합니다.

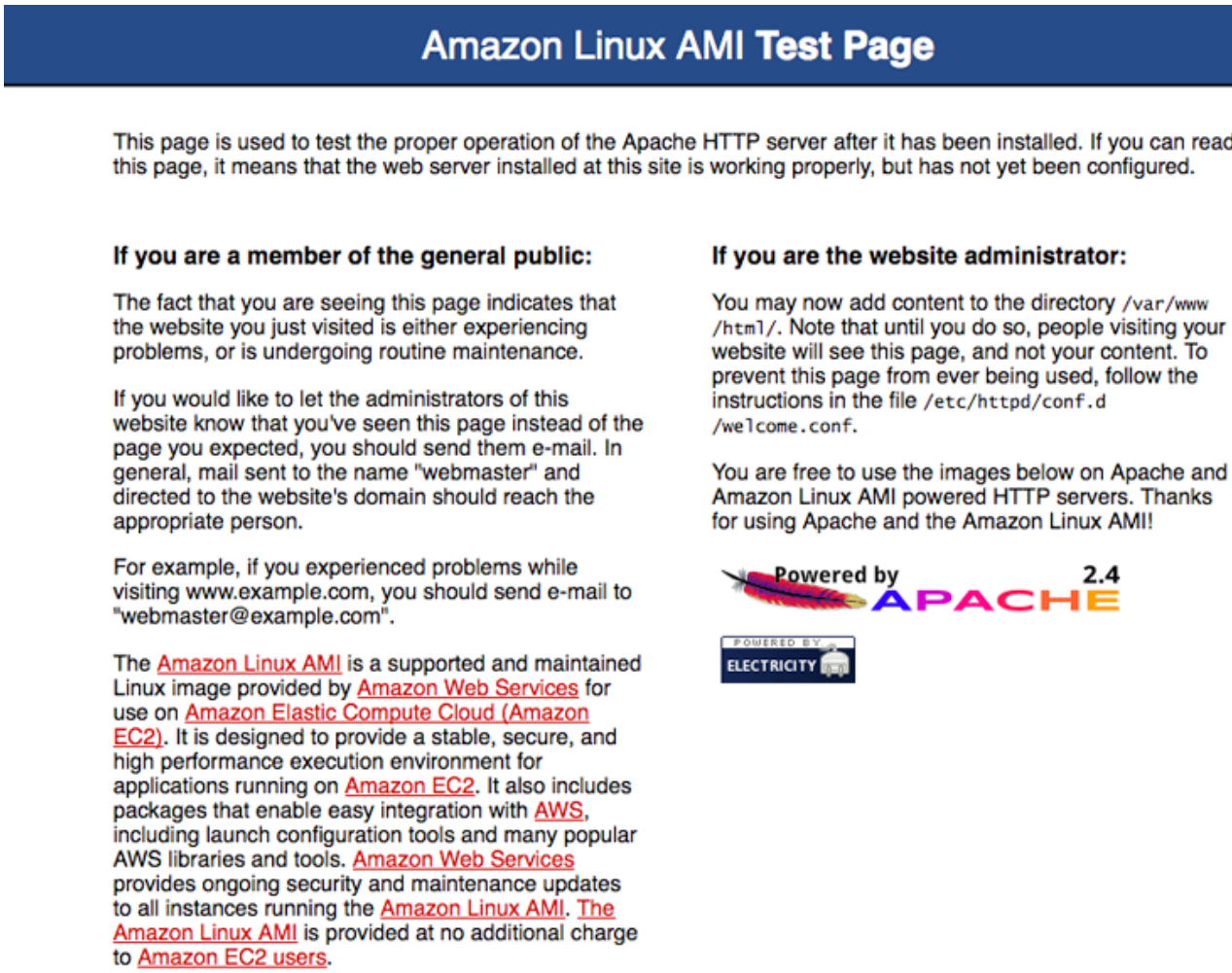
   - 유형: HTTP
   - 프로토콜: TCP
   - 포트 범위: 80
   - 소스: 사용자 지정

7. 웹 서버를 테스트합니다. 웹 브라우저에서 인스턴스의 공용 DNS 주소(또는 공용 IP 주소)를 입력합니다. `/var/www/html`에 콘텐츠가 없으면 Apache 테스트 페이지가 표시됩니다. Amazon EC2 콘솔을 사용하여 인스턴스의 퍼블릭 DNS를 확인할 수 있습니다. 퍼블릭 DNS 열을 확인합니다. 이 열이 숨겨진 경우 열 표시/숨기기 아이콘(기어 모양 아이콘)을 선택하고 퍼블릭 DNS를 선택합니다.

   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. 877))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.



   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 Apache document root라는 디렉터리에 보관된 파일을 처리합니다. Amazon Linux Apache 문서 루트는 `/var/www/html`이며, 기본적으로 루트에서 소유합니다.

```
[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
drwxr-xr-x 2 root root 4096 Aug  7 21:17 noindex
```

`ec2-user` 계정에서 이 디렉터리의 파일을 조작할 수 있게 하려면 디렉터리의 소유권과 권한을 변경해야 합니다. 이 작업을 수행하는 방법에는 여러 가지가 있습니다. 본 자습서에서는 `ec2-user`를 `apache` 그룹에 추가하여 `apache` 그룹에 `/var/www` 디렉터리의 소유권을 부여하고 쓰기 권한을 할당합니다.

파일 권한 설정

1. 사용자(이 경우는 `ec2-user`)를 `apache` 그룹에 추가합니다.

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

2. 로그아웃하고 다시 로그인한 다음, 새 그룹을 선택하고 멤버십을 확인합니다.

   a. 로그아웃합니다(exit 명령을 사용하거나 터미널 창 닫기).

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

   b. `apache` 그룹의 멤버십을 확인하려면 인스턴스에 다시 연결한 후 다음 명령을 실행합니다.

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

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

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

4. 그룹 쓰기 권한을 추가하여 나중에 하위 디렉터리에 대한 그룹 ID를 설정하려면 `/var/www`와 그 하위 디렉터리의 디렉터리 권한을 변경합니다.

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

5. 그룹 쓰기 권한을 추가하려면 `/var/www` 및 그 하위 디렉터리의 파일 권한을 반복하여 변경합니다.

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

이제 `ec2-user`와 `apache` 그룹의 향후 멤버는 Apache document root에서 파일 추가, 삭제, 편집을 할 수 있고, 이를 통해 사용자는 정적 웹 사이트 또는 PHP 애플리케이션과 같은 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 방문하는 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 활성화하는 방법에 대한 자세한 내용은 자습서: Amazon Linux에서 SSL/TLS 구성 (p. 74) 단원을 참조하십시오.

2단계: LAMP 서버 테스트

서버가 설치되어 실행 중이고 파일 권한이 올바르게 설정되었다면 사용자의 `ec2-user` 계정을 통해 인터넷에서 사용 가능한 `/var/www/html` 디렉터리에서 PHP 파일을 생성할 수 있어야 합니다.

LAMP 웹 서버 테스트

1. Apache 문서 루트에서 PHP 파일을 생성합니다.

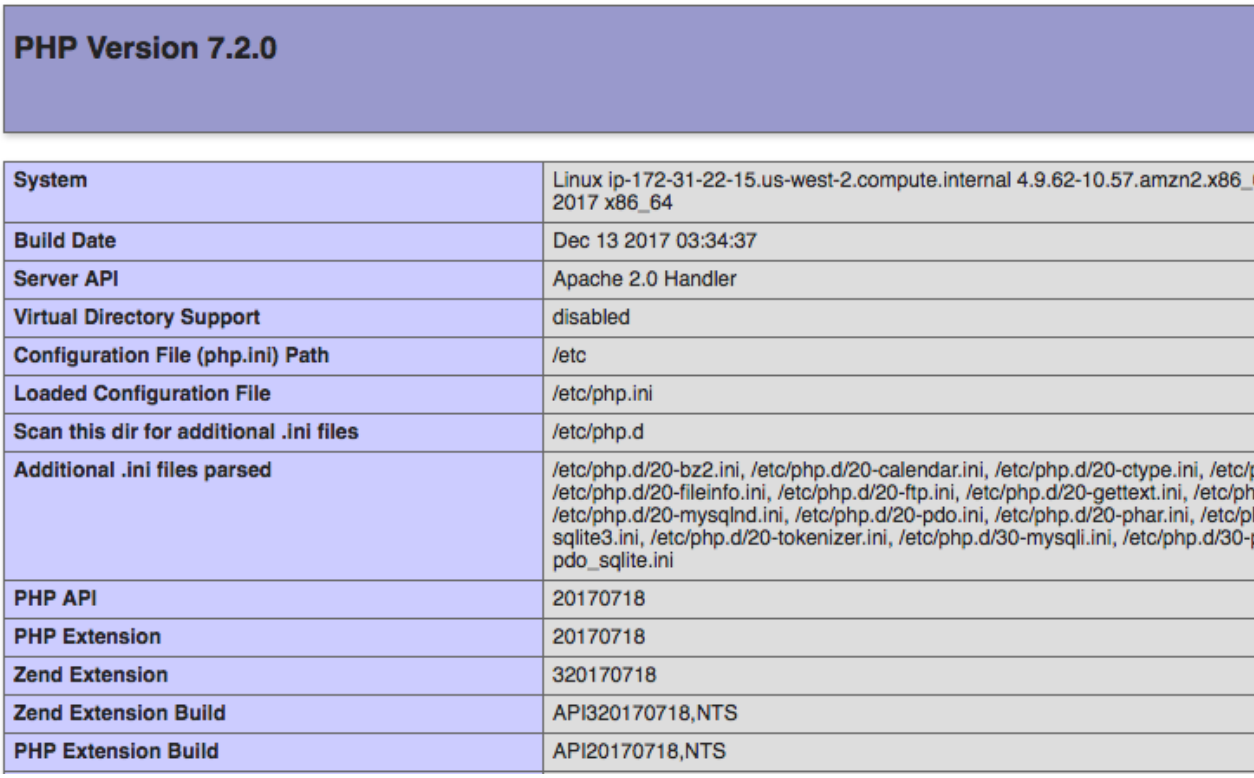
```
[ec2-user ~]$ echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
```

   이 명령을 실행하는 동안 "Permission denied" 오류가 발생하면, 로그아웃하고 다시 로그인한 다음, 1단계: LAMP 서버 준비 (p. 42)에서 구성한 적절한 그룹 권한을 선택합니다.

2. 웹 브라우저에서는 방금 생성한 파일의 URL을 입력합니다. 이 URL은 인스턴스의 퍼블릭 DNS 주소에 슬래시(/)와 파일 이름이 추가된 형태입니다. 다음 예를 참조하십시오.

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

   PHP 정보 페이지가 표시되어야 합니다:



PHP Version 7.2.0

System	Linux ip-172-31-22-15.us-west-2.compute.internal 4.9.62-10.57.amzn2.x86_ 2017 x86_64
Build Date	Dec 13 2017 03:34:37
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.d
Additional .ini files parsed	/etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/ /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gettext.ini, /etc/ph /etc/php.d/20-mysqlnd.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/p sqlite3.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30- pdo_sqlite.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS

   이 페이지가 보이지 않을 경우 이전 단계에서 `/var/www/html/phpinfo.php` 파일이 제대로 생성되었는지 확인하십시오. 또한 다음 명령을 사용하여 필수 패키지가 모두 설치되었는지도 확인할 수 있습니다. 두 번째 열의 패키지 버전은 이 예제 출력과 일치하지 않아도 됩니다.

```
[ec2-user ~]$ sudo yum list installed httpd24 php72 mysql57-server php72-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 사용하여 패키지를 설치합니다.

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

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

3단계: 데이터베이스 서버 보안 설정

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 좋습니다.

데이터베이스 서버를 보안 설정하려면

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. 실행 mysql_secure_installation.

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

   a. 암호를 입력하라는 메시지가 표시되면 루트 계정의 암호를 입력합니다.

      i. 현재 루트 암호를 입력합니다. 기본적으로 root 계정에는 암호가 없습니다. Enter를 누릅니다.

      ii. 암호를 설정하려면 **Y**를 누른 후 안전한 암호를 두 번 입력합니다. 보안 암호 생성에 대한 자세한 내용은 https://identitysafe.norton.com/password-generator/ 단원을 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. **Y**를 눌러서 익명 사용자 계정을 제거합니다.

   c. **Y**를 입력하여 원격 루트 로그인을 비활성화합니다.

   d. **Y**를 눌러서 테스트 데이터베이스를 제거합니다.
   e. **Y**를 눌러서 권한 테이블을 다시 로드하고 변경사항을 저장합니다.
3. (선택 사항) 지금 바로 사용할 계획이 아니라면 MySQL 서버를 중지합니다. 필요할 때 다시 시작할 수 있습니다.

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

4. (선택 사항) 부팅 시 MySQL 서버가 시작되도록 하려면 다음 명령을 입력합니다.

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

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

4단계: (선택 사항) phpMyAdmin 설치

phpMyAdmin을 설치하려면

phpMyAdmin은 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 경우 phpMyAdmin을 사용하여 LAMP 서버에 액세스하지 않는 것이 좋습니다. 이 상태에서 액세스하면 데이터베이스 관리자 암호와 기타 데이터가 인터넷을 통해 안전하지 못한 상태로 전송됩니다. 개발자의 보안 권장 사항을 보려면 phpMyAdmin 설치 보안을 참조하십시오.

> **Note**
>
> Amazon Linux 패키지 관리 시스템은 현재 PHP 7 환경에서 phpMyAdmin의 자동 설치를 지원하지 않습니다. 이 자습서에서는 phpMyAdmin을 직접 설치하는 방법을 설명합니다.

1. SSH를 사용하여 EC2 인스턴스에 로그인합니다.
2. 필요한 종속 항목을 설치합니다.

```
[ec2-user ~]$ sudo yum install php72-mbstring.x86_64 -y
```

3. Apache를 다시 시작합니다.

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

4. Apache 문서 루트(`/var/www/html`)로 이동합니다.

```
[ec2-user ~]$ cd /var/www/html
[ec2-user html]$
```

5. https://www.phpmyadmin.net/downloads에서 phpMyAdmin 최신 릴리스의 소스 패키지를 선택합니다. 인스턴스로 파일을 직접 다운로드하려면 다음 예제와 같이 링크를 복사한 후 wget 명령에 붙여 넣습니다.

```
[ec2-user html]$ wget https://www.phpmyadmin.net/downloads/phpMyAdmin-latest-all-languages.tar.gz
```

6. phpMyAdmin 폴더를 생성하고 다음 명령을 사용하여 이 폴더로 패키지의 압축을 풉니다.

```
[ec2-user html]$ mkdir phpMyAdmin && tar -xvzf phpMyAdmin-latest-all-languages.tar.gz -C phpMyAdmin --strip-components 1
```

7. *phpMyAdmin-latest-all-languages.tar.gz* tarball을 삭제합니다.

```
[ec2-user html]$ rm phpMyAdmin-latest-all-languages.tar.gz
```

8. (선택 사항) MySQL 서버가 실행 중이지 않으면 지금 시작합니다.

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

9. 웹 브라우저에서 phpMyAdmin 설치의 URL을 입력합니다. 아래의 예와 같이 이 URL은 인스턴스의 퍼블릭 DNS 주소(또는 퍼블릭 IP 주소)에 슬래시(/)와 설치 디렉터리의 이름이 추가된 형태입니다. 다음 예를 참조하십시오.

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

사용자는 phpMyAdmin 로그인 페이지를 볼 수 있어야 합니다:

10. 앞서 만든 `root` 사용자 이름 및 MySQL 루트 암호로 phpMyAdmin 설치에 로그인합니다.

    작동하려면 먼저 설치를 구성해야 합니다. phpMyAdmin을 구성하려면 구성 파일을 직접 만들거나, 설정 콘솔을 사용하거나, 두 방법을 조합하여 사용합니다.

    phpMyAdmin에 대한 자세한 내용은 phpMyAdmin 사용 설명서를 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
  ```

  여기서 httpd는 2, 3, 4, 5의 실행 레벨(사용자가 보려는 부분)에서 `on` 상태입니다.

  httpd 프로세스가 실행되지 않는 경우 1단계: LAMP 서버 준비 (p. 42)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. 877))을 참조하십시오.

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

이 자습서는 최신 버전의 Apache HTTP Server, 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 Notes를 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 php72 mysql57-server php72-mysqlnd perl-DBD-
MySQL57
```

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 ~]$ sudo yum remove -y  httpd24 php56 mysql55-server php56-mysqlnd perl-DBD-
MySQL56
```

이제 위에서 설명한 대로 최신 패키지를 설치할 수 있습니다.

관련 주제

파일을 인스턴스에 전송하거나 웹 서버에 WordPress 블로그를 설치하는 것에 대한 자세한 다용은 다음 문서를 참고하십시오.

- WinSCP를 사용하여 Linux 인스턴스로 파일 전송 (p. 501)
- SCP를 사용하여 Linux에서 Linux 인스턴스로 파일 전송 (p. 488)
- 자습서: Amazon Linux를 통한 WordPress 블로그 호스팅 (p. 52)

이 자습서에서 사용되는 명령과 소프트웨어에 대한 자세한 내용은 다음 웹 페이지를 확인하십시오.

- 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 Amazon Route 53에서 도메인 및 하위 도메인 생성 및 마이그레이션을 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(Amazon EC2 Windows 인스턴스에 WordPress 블로그 배포)(출처: Windows 인스턴스용 Amazon EC2 사용 설명서) 단원을 참조하십시오. 데이터베이스가 분리된 고가용성 솔루션이 필요하다면 AWS Elastic Beanstalk 개발자 안내서의 고가용 WordPress 웹 사이트 배포 단원을 참조하십시오.

> **Important**
>
> 이 절차는 Amazon Linux에서 사용하기 위한 것입니다. 기타 배포에 대한 자세한 내용은 해당 설명서를 참조하십시오. 본 자습서에 있는 단계의 상당수가 Ubuntu 인스턴스에서 작동하지 않습니다. Ubuntu 인스턴스에 WordPress를 설치하는 방법은 Ubuntu 설명서에서 WordPress 단원을 참조하십시오.

옵션: 자동화를 사용하여 이 자습서 완료

다음 작업 대신 AWS 시스템 관리자 자동화를 사용하여 이 자습서를 완료하려면 다음 자동화 문서 중 하나를 실행합니다. AWSDocs-HostingAWordPressBlog-AL(Amazon Linux) 또는 AWSDocs-HostingAWordPressBlog-AL2(Amazon Linux 2).

사전 조건

본 자습서는 사용자가 Amazon Linux AMI용 자습서: Amazon Linux AMI를 사용하여 LAMP 웹 서버 설치 (p. 41) 또는 Amazon Linux 2용 자습서: Amazon Linux 2에 LAMP 웹 서버 설치 (p. 32)의 모든 단계를 수행하여 PHP 및 데이터베이스(MySQL 또는 MariaDB) 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. 규칙을 보안 그룹에 추가하는 것에 대한 자세한 내용은 보안 그룹에 규칙 추가 (p. 877) 단원을 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. 676) 단원을 참조하십시오.

블로그에 사용할 도메인 이름이 아직 없을 경우에는 먼저 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 등록을 참조하십시오.

WordPress 설치

인스턴스에 연결한 후 WordPress 설치 패키지를 다운로드합니다.

WordPress 설치 패키지의 다운로드 및 압축해제 방법

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

```
[ec2-user ~]$ wget https://wordpress.org/latest.tar.gz
```

2. 설치 패키지의 압축 및 아카이빙을 해제합니다. 설치 폴더는 `wordpress`라는 폴더로 압축 해제됩니다.

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

WordPress 설치에 대한 데이터베이스 사용자 및 데이터베이스를 생성하려면

WordPress 설치 시 블로그 게시물, 사용자 의견 등의 정보를 데이터베이스에 저장해야 합니다. 다음 절차를 통해 블로그의 데이터베이스와 이 데이터베이스에 대해 정보 읽기 및 저장 권한이 있는 사용자를 생성할 수 있습니다.

1. 데이터베이스 서버를 시작합니다.
   - Amazon Linux 2

     ```
     [ec2-user ~]$ sudo systemctl start mariadb
     ```

   - Amazon Linux AMI

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

2. 데이터베이스 서버에 `root` 사용자로 로그인합니다. 메시지가 표시되면 데이터베이스 `root` 암호를 입력합니다. 이 암호는 사용자의 `root` 시스템 암호와 다를 수 있으며, 데이터베이스 서버를 보안 설정하지 않은 경우 암호가 비어 있을 수도 있습니다.

   데이터베이스 서버를 보안 설정하지 않았다면 반드시 설정하시기 바랍니다. 자세한 내용은 MariaDB 서버의 보안을 유지하려면 (p. 38)(Amazon Linux 2) 또는 데이터베이스 서버를 보안 설정하려면 (p. 47)(Amazon Linux AMI) 단원을 참조하십시오.

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

3. 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.

```
CREATE USER 'wordpress-user'@'localhost' IDENTIFIED BY 'your_strong_password';
```

   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/을(를) 참조하십시오. 기존 암호를 재사용하지 마십시오. 새로 설정한 암호는 안전한 장소에 보관하십시오.

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

   Note

   아래 명령에서 데이터베이스 이름을 앞 뒤로 묶는 기호(`)를 백틱(backtick)이라고 합니다. 백틱(`) 키는 일반적으로 표준 키보드에서 `Tab` 키 위에 있습니다. 백틱이 항상 필요하지는 않지만, 이를 통해 데이터베이스 이름에 하이픈(-) 등 허용되지 않는 문자를 사용할 수 있습니다.

```
CREATE DATABASE `wordpress-db`;
```

5. 데이터베이스에 대한 전체 권한을 이전에 생성한 WordPress 사용자에게 부여합니다.

```
GRANT ALL PRIVILEGES ON `wordpress-db`.* TO "wordpress-user"@"localhost";
```

6. 데이터베이스 권한을 새로고침(flush)해서 모든 변경사항이 적용되도록 합니다.

```
FLUSH PRIVILEGES;
```

7. `mysql` 클라이언트를 종료합니다.

```
exit
```

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 파일에 복사합니다. 이를 통해 새 구성 파일을 생성하고 원본 샘플 파일을 이전 상태 그대로 백업으로 보존할 수 있습니다.

```
[ec2-user ~]$ cp wordpress/wp-config-sample.php wordpress/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 초보자에게는 `nano`가 적합합니다.

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

a. DB_NAME을(를) 정의하는 줄을 찾고 database_name_here을(를) Step 4 (p. 54)의 WordPress 설치에 대한 데이터베이스 사용자 및 데이터베이스를 생성하려면 (p. 53)에서 생성한 데이터베이스 이름으로 변경합니다.

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

b. DB_USER을(를) 정의하는 줄을 찾고 username_here을(를) Step 3 (p. 54)의 WordPress 설치에 대한 데이터베이스 사용자 및 데이터베이스를 생성하려면 (p. 53)에서 생성한 데이터베이스 사용자로 변경합니다.

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

c. DB_PASSWORD을(를) 정의하는 줄을 찾고 password_here을(를) Step 3 (p. 54)의 WordPress 설치에 대한 데이터베이스 사용자 및 데이터베이스를 생성하려면 (p. 53)에서 생성한 보안성이 강력한 암호로 변경합니다.

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

d. Authentication Unique Keys and Salts라는 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/을 방문해서 키 값의 세트를 무작위로 생성하고 이를 wp-config.php 파일로 복사해서 붙여 넣을 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_Keys단원을 참조하시기 바랍니다.

> **Note**
>
> 아래 값은 예시 목적만을 위한 것입니다. 설치 시 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
$eJT@;+(ndLg');
define('LOGGED_IN_KEY',    'ju}qwre3V*+8f_zOWf?{LlGsQ]Ye@2Jh^,8x>)Y |;(^[Iw]Pi
+LG#A4R?7N`YB3');
define('NONCE_KEY',        'P(g62HeZxEes|LnI^i=H,[XwK9I&[2s|:?0N}VJM%?;v2v]v+;
+^9eXUahg@::Cj');
define('AUTH_SALT',        'C$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&+^0P/');
define('NONCE_SALT',       '-97r*V/cgxLmp?Zy4zUU4r99QQ_rGs2LTd%P;|
_e1tS)8_B/,.6[=UK<J_y9?JWG');
```

e. 파일을 저장하고 텍스트 편집기를 종료합니다.

WordPress 파일을 Apache 문서 루트 아래에 설치하려면

- 설치 폴더 압축을 해제하고 MySQL 데이터베이스 및 맞춤형 WordPress 구성 파일을 사용자 설정했으므로, 이제 설치 파일을 웹 서버 문서 루트에 복사하여 설치를 완료하는 설치 스크

립트를 실행할 수 있습니다. 이 파일의 위치는 WordPress 블로그를 웹 서버의 실제 루트(예: *my.public.dns.amazonaws.com*)에서 사용하도록 할지 아니면 루트 아래의 하위 디렉터리나 폴더(예: *my.public.dns.amazonaws.com/blog*)에서 사용하도록 할지에 따라 다릅니다.

- WordPress를 문서 루트에서 실행하려면 WordPress 설치 디렉터리의 파일(디렉터리 자체는 제외)을 다음과 같이 복사합니다.

```
[ec2-user ~]$ cp -r wordpress/* /var/www/html/
```

- WordPress를 문서 루트의 대체 디렉터리에서 실행하려면 먼저 해당 디렉터리를 생성한 후 파일을 그 디렉터리로 복사합니다. 이 예에서는 다음과 같이 WordPress가 `blog` 디렉터리에서 실행됩니다.

```
[ec2-user ~]$ mkdir /var/www/html/blog
[ec2-user ~]$ cp -r wordpress/* /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 루트 아래로 이동한 후에는 WordPress 설치 스크립트가 보호되지 않는 상태이기 때문에 Apache 웹 서버가 실행 중일 때 블로그에 침입자가 액세스할 가능성이 있습니다. Apache 웹 서버를 중지하려면 sudo service httpd stop 명령을 입력합니다. 다음 절차로 즉시 이동하는 경우는 Apache 웹 서버를 중단시킬 필요가 없습니다.

WordPress에서 퍼머링크(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. 자주 사용하는 텍스트 편집기(nano 또는 vim)로 `httpd.conf` 파일을 엽니다. 원하는 텍스트 편집기가 없는 경우 초보자에게는 `nano`가 적합합니다.

```
[ec2-user ~]$ 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 All`로 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.

PHP 그래픽 그리기 라이브러리를 설치하려면

PHP용 GD 라이브러리를 사용하면 이미지를 수정할 수 있습니다. 블로그의 헤더 이미지를 잘라야 하는 경우 다음과 같이 이 라이브러리를 설치합니다.

```
[ec2-user ~]$ sudo yum install php72-gd
```

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. 이미 적용하지 않았다면 다음의 그룹 멤버십 및 권한(LAMP 웹 서버 자습서 (p. 41)에서 자세히 설명)을 적용합니다.

1. `/var/www`의 파일 소유권 및 그 콘텐츠를 `apache` 사용자에게 허용합니다.

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

2. `/var/www` 및 그 콘텐츠의 그룹 소유권을 `apache` 그룹에 허용합니다.

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

3. `/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 {} \;
```

4. `/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 {} \;
```

5. Apache 웹 서버를 재시작해서 새 그룹 및 권한을 가져옵니다.
   - Amazon Linux 2

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

   - Amazon Linux AMI

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

Amazon Linux 2로 WordPress 설치 스크립트 실행 방법

이제 WordPress를 설치할 준비가 되었습니다. 사용하는 명령은 운영 체제에 따라 다릅니다. 이 절차의 명령은 Amazon Linux 2에서 사용하기 위한 것입니다. 이것을 따르는 절차와 Amazon Linux AMI를 함께 사용하십시오.

1. systemctl 명령을 사용하여 시스템이 부팅될 때마다 `httpd` 및 데이터베이스 서비스가 시작되도록 합니다.

```
[ec2-user ~]$ sudo systemctl enable httpd && sudo systemctl enable mariadb
```

2. 데이터베이스 서버가 실행되는지 확인합니다.

```
[ec2-user ~]$ sudo systemctl status mariadb
```

   데이터베이스 서비스가 실행 중이지 않은 경우, 이를 시작합니다.

```
[ec2-user ~]$ sudo systemctl start mariadb
```

3. Apache 웹 서버(`httpd`)가 실행 중인지 확인합니다.

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

   `httpd` 서비스가 실행 중이지 않은 경우, 이를 시작합니다.

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

4. 웹 브라우저에서 WordPress 블로그의 URL을 입력합니다(인스턴스에 대한 퍼블릭 DLS 주소 또는 `blog` 폴더 다음의 주소). 이제 WordPress 설치 스크립트가 나타납니다. WordPress 설치에 필요한 정보를 제공합니다. Install WordPress(WordPress 설치)를 선택해서 설치를 완료합니다. 자세한 내용은 WordPress 웹사이트의 Run the Install Script를 참조하십시오.

Amazon Linux AMI로 WordPress 설치 스크립트 실행 방법

1. chkconfig 명령을 사용하여 시스템이 부팅될 때마다 `httpd` 및 데이터베이스 서비스가 시작되도록 합니다.

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

2. 데이터베이스 서버가 실행되는지 확인합니다.

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

   데이터베이스 서비스가 실행 중이지 않은 경우, 이를 시작합니다.

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

3. Apache 웹 서버(`httpd`)가 실행 중인지 확인합니다.

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

   `httpd` 서비스가 실행 중이지 않은 경우, 이를 시작합니다.

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

4. 웹 브라우저에서 WordPress 블로그의 URL을 입력합니다(인스턴스에 대한 퍼블릭 DLS 주소 또는 `blog` 폴더 다음의 주소). 이제 WordPress 설치 스크립트가 나타납니다. WordPress 설치에 필요한 정보를 제공합니다. Install WordPress(WordPress 설치)를 선택해서 설치를 완료합니다. 자세한 내용은 WordPress 웹사이트의 Run the Install Script를 참조하십시오.

다음 단계

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)을 생성하여 설치 중 오류가 발생하더라도 블로그를 복구할 수 있도록 대비하는 것이 좋습니다. 자세한 내용은 고유 AMI 생성 (p. 91) 단원을 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 Amazon EBS 탄력적 볼륨 (p. 958) 단원을 참조하십시오.
- MySQL 데이터베이스를 Amazon RDS로 이동하여 이 서비스의 간편한 조정 기능을 이용합니다.
- 더 큰 인스턴스 유형으로 마이그레이션합니다. 자세한 내용은 인스턴스 유형 변경 (p. 262) 단원을 참조하십시오.
- 인스턴스를 더 추가합니다. 자세한 내용은 자습서: Amazon EC2에서 애플리케이션의 가용성 향상 (p. 87) 단원을 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_Problems을(를) 참조하십시오. WordPress 블로그 보안 강화에 대한 자세한 내용은 http://codex.wordpress.org/Hardening_WordPress을(를) 참조하십시오. WordPress 블로그 보안 강화에 대한 자세한 내용은 http://codex.wordpress.org/Hardening_WordPress단원을 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_URL을 참조하십시오.

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

wp-cli를 사용하여 WordPress 사이트 URL을 바꾸는 방법

1. SSH를 통해 EC2 인스턴스에 연결합니다.
2. 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 사용하여 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. 다음 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(일반적으로 `/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: Amazon Linux 2에서 SSL/TLS 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 서버 및 Amazon Linux 2의 EC2 인스턴스에 SSL/TLS 지원을 수동으로 추가하는 방법을 설명합니다. 상용 서비스를 제공하려는 경우 여기서 설명하지는 않지만 AWS Certificate Manager가 적합한 옵션입니다.

일반적으로 웹 암호화를 단순히 SSL이라고 부릅니다. 웹 브라우저에서 여전히 SSL을 지원하지만, 후속 프로토콜인 TLS가 공격에 덜 취약합니다. Amazon Linux 2는 기본적으로 모든 SSL 버전에 대한 서버 측 지원을 비활성화합니다. 보안 표준 본문에서는 TLS 1.0이 안전하지 못하다고 간주하며, TLS 1.0과 TLS 1.1은 IETF가 공식적으로 사용 중지 조치를 진행 중입니다. 이 자습서에서는 TLS 1.2 활성화만을 기반으로 하여 지침을 제공합니다. (새 TLS 1.3 프로토콜은 초안 형식으로 존재하지만 아직 Amazon Linux 2에서 지원되지는 않습니다.) 업데이트된 암호화 표준에 대한 자세한 내용은 RFC 7568 및 RFC 8446을 참조하십시오.

이 자습서는 현대 웹 암호화를 단순히 TLS로 언급합니다.

> **Important**
>
> 이 절차는 Amazon Linux 2에서 사용하기 위한 것입니다. 또한 사용자가 새 Amazon EC2 인스턴스로 시작한다고 가정합니다. LAMP 웹 서버를 다른 배포로 인스턴스에서 설치하거나 이전에 사용하던 기존 인스턴스의 재사용하려는 경우에는 본 자습서의 일부 절차를 이용하지 못할 수도 있습니다. Ubuntu의 LAMP 웹 서버에 대한 자세한 내용은 Ubuntu 커뮤니티 문서 ApacheMySQLPHP를 참조하십시오. Red Hat Enterprise Linux에 대한 자세한 내용은 고객 포털 주제 웹 서버를 참조하십시오.

목차

- 사전 조건 (p. 61)
- 1단계: 서버에서 TLS 활성화 (p. 61)
- 2단계: CA가 서명한 인증서 가져오기 (p. 63)
- 3단계: 보안 구성 테스트 및 강화 (p. 68)
- 문제 해결 (p. 70)
- 인증서 자동화: Amazon Linux 2에서 Cerbot의 Let's Encrypt (p. 71)

사전 조건

이 자습서를 시작하기 전에 다음 단계를 완료합니다.

- EBS 지원 Amazon Linux 2 인스턴스를 시작합니다. 자세한 내용은 1단계: 인스턴스 시작 (p. 28) 단원을 참조하십시오.
- 인스턴스가 다음 TCP 포트에서 연결을 허용하도록 보안 그룹을 구성합니다.
  - SSH(포트 22)
  - HTTP(포트 80)
  - HTTPS(포트 443)

  자세한 내용은 Linux 인스턴스의 인바운드 트래픽 권한 부여 (p. 859) 단원을 참조하십시오.
- Apache 웹 서버를 설치합니다. 단계별 지침은 자습서: Amazon Linux 2에 LAMP 웹 서버 설치 (p. 32)를 참조하십시오. httpd 패키지와 그 종속 프로그램만 필요합니다. PHP 및 MariaDB와 관련된 지침은 무시해도 됩니다.
- 웹 사이트를 식별하고 인증하려면 TLS 퍼블릭 키 인프라(PKI)는 도메인 이름 시스템(DNS)을 사용합니다. 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 53을 사용할 수도 있습니다.

1단계: 서버에서 TLS 활성화

이 절차에서는 자체 서명된 디지털 인증서를 사용하여 Amazon Linux 2에 TLS를 설치하는 과정을 보여 줍니다.

> **Note**
>
> 자체 서명된 인증서는 테스트에는 허용되지만 프로덕션에는 허용되지 않습니다. 자체 서명된 인증서를 인터넷에 노출하면 사이트 방문자에게 인사말로 보안 경고가 표시됩니다.

서버에서 TLS를 활성화하려면

1. 인스턴스에 연결 (p. 29)한 다음 Apache가 실행되는지 확인합니다.

```
[ec2-user ~]$ sudo systemctl is-enabled httpd
```

반환된 값이 "enabled"가 아닌 경우 Apache를 시작한 다음 시스템 부팅 시마다 시작하도록 설정합니다.

```
[ec2-user ~]$ sudo systemctl start httpd && sudo systemctl enable httpd
```

2. 모든 소프트웨어 패키지가 최신 상태로 업데이트되어 있는지 확인하기 위해, 인스턴스에서 퀵 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
```

3. 이제 인스턴스가 최신 상태이므로 다음과 같은 Apache module `mod_ssl`을 설치하여 TLS 지원을 추가합니다.

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

이제 인스턴스에는 보안 서버 구성과 테스트를 위한 인증서 생성에 사용할 다음 파일이 포함됩니다.

- `/etc/httpd/conf.d/ssl.conf`

  mod_ssl의 구성 파일입니다. 여기에는 Apache에 암호화 키 및 인증서의 위치, 허용하는 TLS 프로토콜 버전, 허용하는 암호화 암호를 알려주는 명령이 포함되어 있습니다.

- `/etc/pki/tls/certs/make-dummy-cert`

  서버 호스트에 대한 프라이빗 키와 자체 서명된 X.509 인증서 생성용 스크립트입니다. 이 인증서는 Apache가 TLS를 사용하도록 올바르게 설치되었는지 테스트하는 데 유용합니다. 제공하는 자격 증명이 없기 때문에 프로덕션에서 사용되어서는 안됩니다. 프로덕션 환경에서 사용되는 경우 웹 브라우저에서 경고를 트리거합니다.

4. 스크립트를 실행하여 자체 서명된 테스트용 더미 인증서와 키를 생성합니다.

```
[ec2-user ~]$ cd /etc/pki/tls/certs
sudo ./make-dummy-cert localhost.crt
```

그러면 새 파일인 `localhost.crt` 파일이 `/etc/pki/tls/certs/` 디렉터리에 생성됩니다. 지정된 파일의 이름은 `/etc/httpd/conf.d/ssl.conf`의 SSLCertificateFile 명령에 할당된 기본값과 일치합니다.

이 파일에는 자체 서명된 인증서와 인증서의 프라이빗 키가 모두 포함됩니다. Apache에서는 인증서와 키를 PEM 형식으로 요구합니다. 이 형식은 아래의 축약된 예제와 같이 "BEGIN" 및 "END" 라인으로 프레임 처리된 Base64 인코딩 ASCII 문자로 구성됩니다.


```
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQD2KKx/8Zk94m1q
3gQMZF9ZN66Ls19+3tHAgQ5Fpo9KJDhzLjOOCI8u1PTcGmAah5kEitCEc0wzmNeo
BCl0wYR6G0rGaKtK9Dn7CuIjvubtUysVyQoMVPQ97ldeakHWeRMiEJFXg6kZZ0vr
GvwnKoMh3DlK44D9dX7IDua2PlYx5+eroA+1Lqf32ZSaAO0bBIMIYTHigwbHMZoT
...
56tE7THvH7vOEf4/iUOsIrEzaMaJ0mqkmY1A70qQGQKBgBF3H1qNRNHuyMcPODFs
27hDzPDinrquSEvoZIggkDMlh2irTiipJ/GhkvTpoQlv0fK/VXw8vSgeaBuhwJvS
LXU9HvYq0U6O4FgD3nAyB9hI0BE13r1HjUvbjT7moH+RhnNz6eqqdscCS09VtRAo
4QQvAqOa8UheYeoXLdWcHaLP
-----END PRIVATE KEY-----
```

```
-----BEGIN CERTIFICATE-----
MIIEazCCA1OgAwIBAgICWxQwDQYJKoZIhvcNAQELBQAwgbExCzAJBgNVBAYTAi0t
MRIwEAYDVQQIDAlTb21lU3RhdGUxETAPBgNVBAcMCFNvbWVDaXR5MRkwFwYDVQQK
DBBTb21lT3JnYW5pemF0aW9uMR8wHQYDVQQLDBZTb21lT3JnYW5pemF0aW9uYWxV
bml0MRkwFwYDVQQDDBBpcC0xNzItMzEtMjAtMjM2MSQwIgYJKoZIhvcNAQkBFhVy
...
z5rRUE/XzxRLBZOoWZpNWTXJkQ3uFYH6s/sBwtHpKKZMzOvDedREjNKAvk4ws6F0
CuIjvubtUysVyQoMVPQ97ldeakHWeRMiEJFXg6kZZ0vrGvwnKoMh3DlK44D9dlU3
WanXWehT6FiSZvB4sTEXXJN2jdw8g+sHGnZ8zCOsclknYhHrCVD2vnBlZJKSZvak
3ZazhBxtQSukFMOnWPP2a0DMMFGYUHOd0BQE8sBJxg==
-----END CERTIFICATE-----
```

파일 이름과 확장명은 편의상 사용되며 기능에 영향을 미치지 않습니다. 예를 들어 `ssl.conf` 파일에서 관련 명령에 동일한 이름을 사용하는 한, 인증서 이름을 `cert.crt`, `cert.pem` 또는 다른 파일 이름으로 지정할 수 있습니다.

> **Note**
>
> 기본 TLS 파일을 고유의 사용자 지정 파일로 대체하는 경우 파일이 PEM 형식인지 확인하십시오.

5. 자체 서명된 더미 인증서에도 키가 포함되어 있으므로 `/etc/httpd/conf.d/ssl.conf` 파일을 열고 다음 행을 주석으로 처리합니다. 다음 단계를 완료하기 전에 이 행을 주석으로 처리하지 않는 경우 Apache 서비스가 시작되지 않습니다.

```
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
```

6. Apache를 다시 시작합니다.

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

> **Note**
>
> 앞에서 설명한 대로 EC2 인스턴스에서 TCP 포트 443에 액세스할 수 있는지 확인하십시오.

7. 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 호스트 인증서를 사용하여 사이트에 연결하기 때문에 브라우저에 보안 경고가 연속으로 표시될 수 있습니다. 경고를 무시하고 계속 진행합니다.

   Apache 기본 테스트 페이지가 열리면 서버에 TLS가 구성되었다는 것입니다. 브라우저와 서버 사이를 통과하는 모든 데이터가 이제 암호화됩니다.

   > **Note**
   >
   > 사이트 방문자에게 경고 화면이 표시되는 것을 방지하려면 암호화뿐만 아니라 해당 사이트의 소유자라는 것을 공개적으로 인증하는 신뢰할 수 있는 CA 서명 인증서를 가져와야 합니다.

2단계: CA가 서명한 인증서 가져오기

CA가 서명한 인증서를 가져오려면 다음 절차를 사용할 수 있습니다.

- 프라이빗 키에서 인증서 서명 요청(CSR)을 생성합니다.
- Submit the CSR to a certificate authority (CA)
- Obtain a signed host certificate
- Configure Apache to use the certificate

자체 서명된 TLS X.509 호스트 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 서명자로 구성되는 신뢰 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 dmoztools.net에서 확인할 수 있습니다. 일부 CA는 기본 수준 인증서를 무료로 제공합니다. 이러한 CA 중 가장 주목할 만한 것은 Let's Encrypt 프로젝트인데, 이것은 인증서 생성 및 갱신 프로세스의 자동화도 지원합니다. Let's Encrypt as your CA 사용에 대한 자세한 내용은 인증서 자동화: Amazon Linux 2에서 Cerbot의 Let’s Encrypt (p. 71) 단원을 참조하십시오.

호스트 인증서의 기본을 이루는 것은 키입니다. 2019년 현재 정부 및 산업 그룹에서는 2030년까지 문서를 보호하기 위해 마련된 RSA 키에 대해 최소 2048비트의 키(모듈러스) 크기를 사용할 것을 권장합니다. Amazon Linux 2의 OpenSSL에서 생성된 기본 모듈러스 크기는 2048비트이므로 CA가 서명한 인증서에 사용할 수 있습니다. 다음 절차는 사용자 지정된 키(예: 더 큰 모듈러스 또는 다른 암호화 알고리즘 사용)를 원하는 사용자를 위해 제공되는 선택적 단계입니다.

이 CA 서명 호스트 인증서 획득 지침은 등록과 호스팅이 완료된 DNS 도메인을 소유하지 않을 경우 제대로 적용하기 어렵습니다.

CA가 서명한 인증서를 가져오려면

1. 인스턴스에 연결 (p. 29)한 다음 /etc/pki/tls/private/으로 이동합니다. 여기는 TLS에 대한 서버의 프라이빗 키를 저장하는 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 보안 구현의 정도와 유형은 각각 다릅니다.

   - 예 1: 기본 RSA 호스트 키를 만듭니다. 결과 파일인 **custom.key**는 2048비트 RSA 프라이빗 키입니다.

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

   - 예 2: 더 큰 모듈러스로 더 강력한 RSA 키를 만듭니다. 결과 파일인 **custom.key**는 4096비트 RSA 프라이빗 키입니다.

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

   - 예 3: 암호로 보호되는 4096비트 암호화 RSA 키를 생성합니다. 그러면 AES-128 암호화로 암호화된 4096비트 RSA 프라이빗 키인 **custom.key** 파일이 생성됩니다.

     > **Important**
     >
     > 키 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(위의 예에서는 "abcde12345")를 제공해야 합니다.

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

   - 예 4: 비 RSA 암호를 사용하여 키를 생성합니다. RSA 암호화는 두 개의 라지 소수의 결과를 기반으로 하는 공개 키의 크기 때문에 상대적으로 느릴 수 있습니다. 그러나 RSA 암호화 이외의 암호화를 사용하는 TLS의 키를 생성할 수 있습니다. 타원 곡선 수학을 기반으로 하는 키는 동등한 보안 수준을 제공할 때보다 작고 산술적으로 빠릅니다.

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

그 결과는 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 소유권 및 권한은 매우 제한적(소유자=루트, 그룹=루트, 소유자 전용 읽기/쓰기)이어야 합니다. 명령은 다음 예에서와 같습니다.

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

이 명령의 결과는 다음과 같아야 합니다.

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

만족스러운 키를 생성 및 구성한 후 CSR을 생성할 수 있습니다.

3. 원하는 키를 사용하여 CSR을 생성합니다. 다음 예에는 **`custom.key`**가 사용됩니다.

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

OpenSSL은 대화 상자를 열고 아래 표의 정보를 입력하라는 메시지를 표시합니다. 도메인에서 확인된 기본 호스트 인증서의 경우 Common Name을 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 Corporation
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 **`www.example.com`**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 **`*.example.com`**과 같이 와일드카드 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. 사이트 방문자는 이 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.

   Note

   CA는 다양한 목적을 위해 마련된 여러 형식의 파일을 보낼 수 있습니다. 본 자습서에서는 PEM 형식의 인증서 파일만 사용해야 하는데, 이는 보통 `.pem` 또는 `.crt` 파일 확장명으로 표시되지만 항상 그런 것은 아닙니다. 어떤 파일을 사용할지 확실하지 않은 경우 텍스트 편집기로 파일을 열고 다음 라인으로 시작되는 블록 하나 이상이 포함되는 파일을 찾습니다.

   ```
   - - - - -BEGIN CERTIFICATE - - - - -
   ```

   또한 파일은 다음 라인으로 끝나야 합니다.

   ```
   - - - -END CERTIFICATE - - - - -
   ```

   또한 명령줄의 파일을 다음과 같이 테스트할 수 있습니다.

   ```
   [ec2-user certs]$ openssl x509 -in certificate.crt -text
   ```

   이 줄이 파일에 나타나는지 확인하십시오. `.p7b`, `.p7c`, 또는 유사한 파일 확장명으로 끝나는 파일을 사용하지 않습니다.

5. `/etc/pki/tls/certs` 디렉터리에 CA가 서명한 새 인증서와 모든 중간 인증서를 배치합니다.

   Note

   여러 가지 방법으로 새 인증서를 EC2 인스턴스에 업로드할 수 있지만, 가장 간편하고 유익한 방법은 텍스트 편집기(예: vi, nano, 메모장)를 로컬 컴퓨터와 인스턴스에 모두 열고 두 편집기 간에 파일 콘텐츠를 복사하여 붙이는 것입니다. EC2 인스턴스에서 이러한 작업을 수행할 때 루트 [sudo] 권한이 필요합니다. 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 2 기본값(소유자=루트, 그룹=루트, 소유자 전용 읽기/쓰기)과 일치하는지 확인합니다. 다음 예제는 사용하는 명령을 보여 줍니다.

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

   이 명령의 결과는 다음과 같아야 합니다.

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

중간 인증서 파일에 대한 권한은 덜 엄격합니다(소유자=루트, 그룹=루트, 소유자 쓰기 가능, 그룹 읽기 가능, 모든 사용자 읽기 가능). 다음 예제는 사용하는 명령을 보여 줍니다.

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

이 명령의 결과는 다음과 같아야 합니다.

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

6. `/etc/pki/tls/private/` 디렉터리에서 CSR을 생성할 때 사용한 프라이빗 키를 배치합니다.

   Note

   여러 가지 방법으로 사용자 지정 키를 EC2 인스턴스에 업로드할 수 있지만, 가장 간편하고 유익한 방법은 텍스트 편집기(예: vi, nano, 메모장)를 로컬 컴퓨터와 인스턴스에 모두 열고 두 편집기 간에 파일 콘텐츠를 복사하여 붙이는 것입니다. EC2 인스턴스에서 이러한 작업을 수행할 때 루트 [sudo] 권한이 필요합니다. 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/private` 디렉터리 내에서 파일 소유권, 그룹 및 권한 설정이 매우 제한적인 Amazon Linux 2 기본값(소유자=루트, 그룹=루트, 소유자 전용 읽기/쓰기)과 일치하는지 다음 명령을 사용하여 확인합니다.

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

   이 명령의 결과는 다음과 같아야 합니다.

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

7. 새 인증서 및 키 파일을 반영하기 위해 `/etc/httpd/conf.d/ssl.conf`를 편집합니다.

   a. Apache의 `SSLCertificateFile` 명령에 CA가 서명한 호스트 인증서의 경로와 파일 이름을 입력합니다.

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

   b. 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
```

   Note

   일부 CA는 호스트 인증서와 중간 인증서를 단일 파일로 결합하기 때문에 `SSLCACertificateFile` 명령이 불필요합니다. CA가 제공한 지침을 참조하십시오.

   c. Apache의 `SSLCertificateKeyFile` 명령에 프라이빗 키(이 예에서는 `custom.key`)의 경로와 파일 이름을 입력합니다.

```
SSLCertificateKeyFile /etc/pki/tls/private/custom.key
```

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

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

9. `https://` 접두사가 포함된 브라우저 URL 막대에 도메인 이름을 입력하여 서버를 테스트합니다. 브라우저에서는 테스트 페이지가 오류 생성 없이 HTTPS를 통해 로드되어야 합니다.

3단계: 보안 구성 테스트 및 강화

TLS이 작동되고 일반에 공개된 후 이의 실제 보안 수준을 테스트해야 합니다. 보안 설정을 무료로 완벽하게 분석해 주는 Qualys SSL Labs와 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 단원을 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 Linux 2 및 기본 Certbot 인증서의 기본 Apache 구성과 설정이 동일한 도메인에 대한 보고서가 요약되어 있습니다.

종합 등급	B
Certificate	100%
프로토콜 지원	95%
키 교환	70%
암호화 수준	90%

개요에서 구성이 대체로 문제가 없어 보여도 세부 정보 보고서에서는 몇몇 잠재적 문제를 여기에 심각도 순서로 나열하여 표시합니다.

✗ RC4 암호는 오래된 특정 브라우저에서 사용하도록 지원됩니다. 암호는 암호화 알고리즘의 수학적 핵심입니다. TLS 데이터 스트림을 암호화하는 데 사용하는 빠른 암호인 RC4에는 몇 가지 심각한 취약점이 있는 것으로 알려져 있습니다. 타당한 레거시 브라우저 지원 사유가 없다면 비활성화해야 합니다.

✗ 이전의 TLS 버전이 지원됩니다. 구성에서는 TLS 1.0(이미 사용 중지 상태)과 TLS 1.1(사용 중지 절차 진행 중)을 지원합니다. 2018년부터는 TLS 1.2만 권장됩니다.

✗ 전방향 보안은 부분적으로 지원됩니다. 전방향 보안은 프라이빗 키에서 파생된 임시(사용 후 삭제) 세션 키를 사용하여 암호화하는 알고리즘의 기능입니다. 이는 실제 공격자가 웹 서버의 장기 프라이빗 키를 보유하고 있더라도 HTTPS 데이터의 암호를 해독할 수 없다는 것을 뜻합니다.

TLS 구성을 수정하고 향후에 대비하려면

1. 텍스트 편집기에서 `/etc/httpd/conf.d/ssl.conf` 구성 파일을 열고 다음 줄의 시작 부분에 "#"을 입력하여 해당 줄을 주석으로 처리합니다.

```
#SSLProtocol all -SSLv3
```

2. 다음 명령을 추가합니다.

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

이러한 명령은 SSL 버전 2 및 3과 TLS 버전 1.0 및 1.1을 명시적으로 비활성화합니다. 이제 이 서버는 TLS 1.2 이외의 프로토콜을 사용하는 클라이언트와의 암호화된 연결을 허용하지 않습니다. 명령의 상세 내용은 서버의 구성 내용을 사람에게 더욱 명확히 전달합니다.

Note

이러한 방식으로 TLS 버전 1.0 및 1.1을 비활성화하면 적은 비율의 오래된 웹 브라우저가 사이트에 액세스하지 못하도록 차단합니다.

허용된 암호의 목록을 수정하려면

1. 구성 파일인 `/etc/httpd/conf.d/ssl.conf`에서 **`SSLCipherSuite`** 명령이 포함된 섹션을 찾고 기존의 줄을 줄의 시작에 “#”을 입력하여 주석으로 처리합니다.

```
#SSLCipherSuite HIGH:MEDIUM:!aNULL:!MD5
```

2. 명시적 암호 그룹과 전방향 보안을 우선순위에 두고 부정확한 암호를 방지하는 암호 오더를 지정합니다. 여기에 사용된 `SSLCipherSuite` 명령은 서버에서 실행되는 특정 소프트웨어에 맞게 TLS 구성을 조정하는 Mozilla SSL Configuration Generator의 출력에 기반합니다. (자세한 내용은 Mozilla의 Security/Server Side TLS(보안/서버 측 TLS)를 참조하십시오.) 먼저 다음 명령의 출력을 사용하여 Apache와 OpenSSL의 버전을 확인합니다.

```
[ec2-user ~]$ yum list installed | grep httpd

[ec2-user ~]$ yum list installed | grep openssl
```

예를 들어, 반환된 정보가 Apache 2.4.34 및 OpenSSL 1.0.2인 경우 이를 생성기에 입력합니다. "현대" 호환성 모델을 선택하면 적극적으로 보안을 적용하지만 대부분의 브라우저에서 여전히 작동하는 `SSLCipherSuite` 명령을 생성합니다. 소프트웨어가 최신 구성을 지원하지 않으면 소프트웨어를 업데이트하거나 대신 "중간" 구성을 선택할 수 있습니다.

```
SSLCipherSuite ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-
CHACHA20-POLY1305:
ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:
ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-
AES128-SHA256
```

선택된 암호에는 이름에 Elliptic Curve Diffie-Hellman Ephemeral의 약자인 ECDHE가 포함되어 있습니다. ephemeral은 전방향 보안을 나타냅니다. 부차적 결과로서 해당 암호는 RC4를 지원하지 않습니다.

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

생성된 명령을 `/etc/httpd/conf.d/ssl.conf`에 복사합니다.

Note

여기에서는 가독성을 위해 여러 줄로 표시했지만, 이 명령은 `/etc/httpd/conf.d/ssl.conf`에 복사할 때 암호 이름 사이에 공백 없이 콜론만을 추가하여 한 줄에 입력해야 합니다.

3. 마지막으로 줄 시작 부분에 있는 "#"을 제거하여 다음 줄의 주석 처리를 해제합니다.

```
#SSLHonorCipherOrder on
```

이 명령은 (이 예에서는) 전방향 보안을 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.

이 두 절차를 모두 완료한 다음에는 변경 사항을 `/etc/httpd/conf.d/ssl.conf`에 저장하고 Apache를 재시작합니다.

Qualys SSL Labs에서 도메인을 다시 테스트하면 RC4 취약성과 다른 경고 문제가 해결되고 요약은 다음과 같을 것입니다.

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

> **Important**
>
> OpenSSL을 업데이트할 때마다 새 암호가 사용되고 이전 암호에 대한 지원은 제거됩니다. EC2 Amazon Linux 2 인스턴스를 최신으로 유지하고, OpenSSL의 보안 알림을 잘 확인하며, 기술 정보 신문의 새 보안 취약점에 대한 보고를 잘 살핍니다. 자세한 정보는 Classic Load Balancer 사용 설명서의 Predefined SSL Security Policies for Elastic Load Balancing 단원을 참조하십시오.

문제 해결

- 암호를 입력하지 않으면 Apache 웹 서버가 시작하지 않습니다.

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

  키에서 암호화 및 암호 요구 사항을 제거할 수 있습니다. 기본 디렉터리에 `custom.key`라는 암호화된 프라이빗 RSA 키가 있고 이 키의 암호가 **`abcde12345`**라고 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 systemctl restart httpd
```

  이제 Apache가 암호를 묻지 않고 시작할 것입니다.

- **`sudo yum install -y mod_ssl`**을 실행하면 오류가 발생합니다.

  SSL에 필요한 패키지를 설치하려 할 때 다음과 같은 오류가 표시될 수 있습니다.

```
Error: httpd24-tools conflicts with httpd-tools-2.2.34-1.16.amzn1.x86_64
```

```
Error: httpd24 conflicts with httpd-2.2.34-1.16.amzn1.x86_64
```

이 오류는 일반적으로 EC2 인스턴스에서 Amazon Linux 2가 실행되고 있지 않음을 의미합니다. 이 자습서는 공식 Amazon Linux 2 AMI에서 새로 생성된 인스턴스만 지원합니다.

인증서 자동화: Amazon Linux 2에서 Cerbot의 Let's Encrypt

Let's Encrypt 인증 기관은 전체 인터넷을 암호화하기 위한 EFF(Electronic Frontier Foundation) 작업의 중요한 부분입니다. 그 목표에 따라, Let's Encrypt 호스트 인증서는 사용자의 개입을 최소화하면서 생성, 검증, 설치 및 유지되도록 설계되었습니다. 인증서 관리의 자동화된 측면은 웹 서버에서 실행되는 소프트웨어 에이전트에 의해 수행됩니다. 에이전트를 설치 및 구성한 후 Let's Encrypt와 안전하게 통신하고 Apache 및 키 관리 시스템에서 관리 작업을 수행합니다. 본 자습서에서는 사용자 지정된 암호화 키를 인증서의 기반으로 공급할 수 있거나 에이전트 자체가 기본값에 따라 키를 생성할 수 있기 때문에 무료 Certbot 에이전트를 사용합니다. 또한 Certbot이 Certbot을 자동화하려면 (p. 74)에서 설명한 대로 사용자의 상호 작용 없이 정기적으로 인증서를 갱신하도록 구성할 수 있습니다. 자세한 내용은 Certbot User Guide 및 man page를 참조하십시오.

Certbot은 Amazon Linux 2에서 공식적으로 지원되지 않지만 다운로드가 가능하고 설치하면 제대로 작동합니다. 다음과 같이 백업을 만들어 데이터를 보호하고 불편을 피하는 것이 좋습니다.

- 시작하기 전에 Amazon EBS 루트 볼륨의 스냅샷을 만듭니다. 이를 통해 EC2 인스턴스의 원래 상태를 복원할 수 있습니다. EBS 스냅샷에 대한 자세한 내용은 Amazon EBS 스냅샷 만들기 (p. 929)을 참조하십시오.
- 아래 절차의 경우 Apache의 운영을 제어하는 `httpd.conf` 파일을 편집해야 합니다. Certbot은 이 파일과 기타 구성 파일을 고유하게 자동 변경합니다. 복원이 필요한 경우에 대비해 전체 `/etc/httpd` 디렉터리 백업 복사본을 만드십시오.

설치 준비

Certbot을 설치하기 전에 다음 절차를 완료합니다.

1. EPEL(Extra Packages for Enterprise Linux) 7 리포지토리 패키지를 다운로드합니다. Certbot에 필요한 종속성을 공급하는 데 필요합니다.

   a. 홈 디렉터리(`/home/ec2-user`)로 이동합니다. 다음 명령으로 EPEL을 다운로드합니다.

   ```
   [ec2-user ~]$ sudo wget -r --no-parent -A 'epel-release-*.rpm' http://
   dl.fedoraproject.org/pub/epel/7/x86_64/Packages/e/
   ```

   b. 다음 명령과 같이 리포지토리 패키지를 설치합니다.

   ```
   [ec2-user ~]$ sudo rpm -Uvh dl.fedoraproject.org/pub/epel/7/x86_64/Packages/e/epel-
   release-*.rpm
   ```

   c. 다음 명령과 같이 EPEL을 활성화합니다.

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

   다음 명령을 사용하여 EPEL이 활성화되었는지 확인할 수 있습니다. 그러면 다음과 비슷한 정보가 반환됩니다.

   ```
   [ec2-user ~]$ sudo yum repolist all

   ...
   ```

```
epel/x86_64                                Extra Packages for Enterprise Linux 7 - x86_64
                                 enabled: 12949+175
epel-debuginfo/x86_64                      Extra Packages for Enterprise Linux 7 - x86_64
 - Debug                         enabled:      2890
epel-source/x86_64                         Extra Packages for Enterprise Linux 7 - x86_64
 - Source                        enabled:         0
epel-testing/x86_64                        Extra Packages for Enterprise Linux 7 -
 Testing - x86_64                           enabled:    778+12
epel-testing-debuginfo/x86_64              Extra Packages for Enterprise Linux 7 -
 Testing - x86_64 - Debug                   enabled:       107
epel-testing-source/x86_64                 Extra Packages for Enterprise Linux 7 -
 Testing - x86_64 - Source                  enabled:         0
...
```

2. Apache 구성 파일인 `/etc/httpd/conf/httpd.conf`를 편집합니다. "`Listen 80`" 명령을 찾고 이 명령 뒤에 다음 줄을 추가하여 예제 도메인 이름을 실제 일반 이름 및 주체 대체 이름(SAN)으로 교체합니다.

```
<VirtualHost *:80>
    DocumentRoot "/var/www/html"
    ServerName "example.com"
    ServerAlias "www.example.com"
</VirtualHost>
```

파일을 저장하고 Apache를 다시 시작합니다.

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

Certbot 설치 및 실행

이 절차는 Fedora 및 RHEL 7에 Certbot 설치에 대한 EFF의 문서를 기반으로 합니다. Certbot의 기본 사용 및 결과로 얻은 2048비트 RSA 키에 기반을 둔 인증서에 대해 설명합니다. 사용자 지정 키를 실험하려면 Let's Encrypt에 ECDSA 인증서 사용부터 시작할 수 있습니다.

1. 다음 명령을 사용하여 Certbot 패키지 및 종속성을 설치합니다.

```
[ec2-user ~]$ sudo yum install -y certbot python2-certbot-apache
```

2. Certbot을 실행합니다.

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

3. "Enter email address (used for urgent renewal and security notices)," 프롬프트에서 연락처 주소를 입력하고 Enter 키를 누릅니다.
4. 프롬프트에서 Let's Encrypt 서비스 계약 조건에 동의합니다. 진행하려면 "A"를 입력하고 Enter 키를 누릅니다.

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf. You must
agree in order to register with the ACME server at
https://acme-v02.api.letsencrypt.org/directory
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(A)gree/(C)ancel: A
```

5. EFF 권한 부여에서 "Y" 또는 "N"를 입력하여 메일 발송 목록에 포함시키고 Enter 키를 누릅니다.
6. Certbot은 VirtualHost 블록에 입력한 일반 이름 및 주체 대체 이름(SAN)을 표시합니다.

```
Which names would you like to activate HTTPS for?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1: example.com
2: www.example.com
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Select the appropriate numbers separated by commas and/or spaces, or leave input
blank to select all options shown (Enter 'c' to cancel):
```

입력란을 공백으로 남겨 두고 Enter를 누릅니다.

7. Certbot은 인증서를 생성하고 Apache를 구성하면서 다음 출력을 표시합니다. 그런 다음 HTTPS로의 HTTP 쿼리 리디렉션 메시지가 표시됩니다.

```
Obtaining a new certificate
Performing the following challenges:
http-01 challenge for example.com
http-01 challenge for www.example.com
Waiting for verification...
Cleaning up challenges
Created an SSL vhost at /etc/httpd/conf/httpd-le-ssl.conf
Deploying Certificate for example.com to VirtualHost /etc/httpd/conf/httpd-le-ssl.conf
Enabling site /etc/httpd/conf/httpd-le-ssl.conf by adding Include to root configuration
Deploying Certificate for www.example.com to VirtualHost /etc/httpd/conf/httpd-le-
ssl.conf

Please choose whether or not to redirect HTTP traffic to HTTPS, removing HTTP access.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1: No redirect - Make no further changes to the webserver configuration.
2: Redirect - Make all requests redirect to secure HTTPS access. Choose this for
new sites, or if you're confident your site works on HTTPS. You can undo this
change by editing your web server's configuration.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Select the appropriate number [1-2] then [enter] (press 'c' to cancel):
```

방문자가 암호화되지 않은 HTTP를 통해 서버에 연결하는 것을 허용하려면 "1"을 입력합니다. HTTPS를 통한 암호화된 연결만을 허용하고자 하는 경우 "2"를 입력합니다. Enter 키를 눌러 선택을 제출합니다.

8. Certbot은 Apache의 구성을 완료하고 성공 및 기타 정보를 보고합니다.

```
Congratulations! You have successfully enabled https://example.com and
https://www.example.com

You should test your configuration at:
https://www.ssllabs.com/ssltest/analyze.html?d=example.com
https://www.ssllabs.com/ssltest/analyze.html?d=www.example.com
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

IMPORTANT NOTES:
 - Congratulations! Your certificate and chain have been saved at:
   /etc/letsencrypt/live/certbot.oneeyedman.net/fullchain.pem
   Your key file has been saved at:
   /etc/letsencrypt/live/certbot.oneeyedman.net/privkey.pem
   Your cert will expire on 2019-08-01. To obtain a new or tweaked
   version of this certificate in the future, simply run certbot again
   with the "certonly" option. To non-interactively renew *all* of
   your certificates, run "certbot renew"
 - Your account credentials have been saved in your Certbot
   configuration directory at /etc/letsencrypt. You should make a
   secure backup of this folder now. This configuration directory will
   also contain certificates and private keys obtained by Certbot so
   making regular backups of this folder is ideal.
```

9. 설치를 완료한 후 3단계: 보안 구성 테스트 및 강화 (p. 68)에서 설명한 대로 서버의 보안을 테스트 및 최적화합니다.

자동화된 인증서 갱신 구성

Certbot은 서버 시스템의 표시되지 않고 오류를 방지하는 부분이 되도록 설계되었습니다. 기본적으로 만료 시간이 90일로 짧은 호스트 인증서를 생성합니다. 이 명령을 자동으로 호출하도록 시스템을 구성하지 않은 경우 만료 전에 certbot 명령을 수동으로 다시 실행해야 합니다. 이 절차는 cron 작업을 설정하여 Certbot을 자동화하는 방법을 보여줍니다.

Certbot을 자동화하려면

1. 텍스트 편집기로 `/etc/crontab` 파일을 열고 다음과 유사한 행을 추가합니다.

```
39      1,13    *       *       *       root    certbot renew --no-self-upgrade
```

   작업을 마치면 파일을 저장합니다. 다음은 명령의 각 구성에 대한 설명입니다.

   `39 1,13 * * *`

   명령이 매일 01:39와 13:39에 실행되도록 예약합니다. 선택한 값은 임의이지만 Certbot 개발자는 최소한 매일 두 번 명령을 실행하는 것을 제안합니다. 이를 통해 손상된 것으로 발견된 모든 인증서가 즉시 취소 및 교체됩니다.

   `root`

   명령은 루트 권한으로 실행됩니다.

   `certbot renew --no-self-upgrade`

   실행할 명령입니다. renew 하위 명령을 사용하면 Certbot이 이전에 얻은 모든 인증서가 점검하고 만료 날짜가 다가오고 있는 인증서를 갱신합니다. `--no-self-upgrade` 플래그를 사용하면 Certbot이 사용자의 개입 없이 자체 업그레이드하지 않습니다.

2. cron 데몬을 다시 시작합니다.

```
[ec2-user ~]$ sudo systemctl restart crond
```

자습서: Amazon Linux에서 SSL/TLS 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 Amazon Linux AMI 및 Apache 웹 서버의 EC2 인스턴스에 SSL/TLS 지원을 수동으로 추가하는 방법을 설명합니다. 상용 서비스를 제공하려는 경우 여기서 설명하지는 않지만 AWS Certificate Manager가 적합한 옵션입니다.

일반적으로 웹 암호화를 단순히 SSL이라고 부릅니다. 웹 브라우저에서 여전히 SSL을 지원하지만, 후속 프로토콜인 TLS가 공격에 덜 취약합니다. Amazon Linux AMI는 기본적으로 모든 버전의 SSL을 서버 측에서 비활성화합니다. 보안 표준 본문에서는 TLS 1.0이 안전하지 못하다고 간주하며, TLS 1.0과 TLS 1.1은 IETF가 공식적으로 사용 중지 조치를 진행 중입니다. 이 자습서에서는 TLS 1.2 활성화만을 기반으로 하여 지침을 제공합니다. (새 TLS 1.3 프로토콜은 초안 형식으로 존재하지만 아직 Amazon Linux 2에서 지원되지는 않습니다.) 업데이트된 암호화 표준에 대한 자세한 내용은 RFC 7568 및 RFC 8446을 참조하십시오.

이 자습서는 현대 웹 암호화를 단순히 TLS로 언급합니다.

> **Important**
>
> 이 절차는 Amazon Linux AMI에서 사용하기 위한 것입니다. LAMP 웹 서버를 다른 배포 인스턴스에서 설치하려는 경우는 본 자습서의 일부 절차를 이용할 수 없습니다. Ubuntu의 LAMP 웹 서버

에 대한 자세한 내용은 Ubuntu 커뮤니티 문서 ApacheMySQLPHP 항목을 참조하십시오. Red Hat Enterprise Linux에 대한 자세한 내용은 고객 포털 문서 웹 서버를 참조하십시오.

목차

- 사전 조건 (p. 75)
- 1단계: 서버에서 TLS 활성화 (p. 75)
- 2단계: CA가 서명한 인증서 가져오기 (p. 77)
- 3단계: 보안 구성 테스트 및 강화 (p. 81)
- 문제 해결 (p. 83)
- 인증서 자동화: Amazon Linux에서 Cerbot의 Let's Encrypt (p. 84)

사전 조건

이 자습서를 시작하기 전에 다음 단계를 완료합니다.

- Amazon Linux AMI를 사용하여 EBS 지원 인스턴스를 시작합니다. 자세한 내용은 1단계: 인스턴스 시작 (p. 28) 단원을 참조하십시오.
- 보안 그룹을 인스턴스가 다음 TCP 포트에서 연결을 허용하도록 구성합니다.
  - SSH(포트 22)
  - HTTP(포트 80)
  - HTTPS(포트 443)

  자세한 내용은 Linux 인스턴스의 인바운드 트래픽 권한 부여 (p. 859) 단원을 참조하십시오.
- Apache 웹 서버를 설치합니다. 단계별 지침은 자습서: Amazon Linux에 LAMP 웹 서버 설치 (p. 41)를 참조하십시오. http24 패키지와 그 종속 프로그램만 필요합니다. PHP 및 MySQL과 관련된 지침은 무시해도 됩니다.
- 웹 사이트를 식별하고 인증하려면 TLS 퍼블릭 키 인프라(PKI)는 도메인 이름 시스템(DNS)을 사용합니다. 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 53을 사용할 수도 있습니다.

1단계: 서버에서 TLS 활성화

이 절차에서는 자체 서명된 디지털 인증서를 사용하여 Amazon Linux에 TLS를 설치하는 과정을 보여 줍니다.

> **Note**
>
> 자체 서명된 인증서는 테스트에는 허용되지만 프로덕션에는 허용되지 않습니다. 자체 서명된 인증서를 인터넷에 노출하면 사이트 방문자에게 인사말로 보안 경고가 표시됩니다.

서버에서 TLS를 활성화하려면

1. 인스턴스에 연결 (p. 29)한 다음 Apache가 실행되는지 확인합니다.

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

   필요한 경우 Apache를 시작합니다.

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

2. 모든 소프트웨어 패키지가 최신 상태로 업데이트되어 있는지 확인하기 위해, 인스턴스에서 퀵 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
```

3. 이제 인스턴스가 최신 상태이므로 다음과 같은 Apache module `mod_ssl`을 설치하여 TLS 지원을 추가합니다.

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

   이제 인스턴스에는 보안 서버 구성과 테스트를 위한 인증서 생성에 사용할 다음 파일이 포함됩니다.

   `/etc/httpd/conf.d/ssl.conf`

   mod_ssl의 구성 파일입니다. 여기에는 Apache에 암호화 키 및 인증서의 위치, 허용하는 TLS 프로토콜 버전, 허용하는 암호화 암호를 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, 이 키를 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 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-----
MIIEazCCA1OgAwIBAgICWxQwDQYJKoZIhvcNAQELBQAwgbExCzAJBgNVBAYTAi0t
MRIwEAYDVQQIDAlTb211lU3RhdGUxETAPBgNVBAcMCFNvbWVDaXR5MRkwFwYDVQQK
DBBTb211lT3JnYW5pemF0aW9uMR8wHQYDVQQLDBZTb211lT3JnYW5pemF0aW9uYWxV
bml0MRkwFwYDVQQDDBBpcC0xNzItMzEtMjAtMjM2MSQwIgYJKoZIhvcNAQkBFhVy
...
z5rRUE/XzxRLBZOoWZpNWTXJkQ3uFYH6s/sBwtHpKKZMzOvDedREjNKAvk4ws6F0
WanXWehT6FiSZvB4sTEXXJN2jdw8g+sHGnZ8zCOsclknYhHrCVD2vnBlZJKSZvak
3ZazhBxtQSukFMOnWPP2a0DMMFGYUHOd0BQE8sBJxg==
-----END CERTIFICATE-----
```


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

   Note

   기본 TLS 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 호스트 인증서를 사용하여 사이트에 연결하기 때문에 브라우저에 보안 경고가 연속으로 표시될 수 있습니다.

   경고를 무시하고 계속 진행합니다. Apache 기본 테스트 페이지가 열리면 서버에 TLS가 구성되었다는 것입니다. 브라우저와 서버 사이를 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

CA가 서명한 인증서를 가져오려면 다음 절차를 사용할 수 있습니다.

- 프라이빗 키에서 인증서 서명 요청(CSR)을 생성합니다.
- Submit the CSR to a certificate authority (CA)
- Obtain a signed host certificate
- Configure Apache to use the certificate

자체 서명된 TLS X.509 호스트 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 서명자로 구성되는 신뢰 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 dmoztools.net에서 확인할 수 있습니다. 일부 CA는 기본 수준 인증서를 무료로 제공합니다. 이 중 가장 주목할 만한 것은 Let's Encrypt 프로젝트인데, 이것은 인증서 생성 및 갱신 프로세스의 자동화도 지원합니다. Let's Encrypt as your CA 사용에 대한 자세한 내용은 인증서 자동화: Amazon Linux에서 Cerbot의 Let’s Encrypt (p. 84) 단원을 참조하십시오.

호스트 인증서의 기본을 이루는 것은 키입니다. 2017년 현재 정부 및 산업 그룹에서는 2030년까지 문서를 보호하기 위해 마련된 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.

이 CA 서명 호스트 인증서 획득 지침은 등록과 호스팅이 완료된 DNS 도메인을 소유하지 않을 경우 제대로 적용하기 어렵습니다.

CA가 서명한 인증서를 가져오려면

1. 인스턴스에 연결 (p. 29)한 다음 /etc/pki/tls/private/으로 이동합니다. 이는 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.

   - 예 1: 기본 RSA 호스트 키를 만듭니다. 결과 파일인 **`custom.key`**는 2048비트 RSA 프라이빗 키입니다.

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

- 예 2: 더 큰 모듈러스로 더 강력한 RSA 키를 만듭니다. 결과 파일인 **custom.key**는 4096비트 RSA 프라이빗 키입니다.

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

- 예 3: 암호로 보호되는 4096비트 암호화 RSA 키를 생성합니다. 그러면 AES-128 암호화로 암호화된 4096비트 RSA 프라이빗 키인 **custom.key** 파일이 생성됩니다.

  Important

  키 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(위의 예에서는 "abcde12345")를 제공해야 합니다.

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

- 예 4: 비 RSA 암호를 사용하여 키를 생성합니다. RSA 암호화는 두 개의 라지 소수의 결과를 기반으로 하는 공개 키의 크기 때문에 상대적으로 느릴 수 있습니다. 그러나 RSA 암호화 이외의 암호화를 사용하는 TLS의 키를 생성할 수 있습니다. 타원 곡선 수학을 기반으로 하는 키는 동등한 보안 수준을 제공할 때보다 작고 산술적으로 빠릅니다.

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

  그 결과는 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 소유권 및 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.

3. 원하는 키를 사용하여 CSR을 생성합니다. 아래 예에서는 **custom.key**를 사용합니다.

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

OpenSSL은 대화 상자를 열고 아래 표의 정보를 입력하라는 메시지를 표시합니다. 도메인에서 확인된 기본 호스트 인증서의 경우 Common Name을 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 Corporation
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 **`www.example.com`**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 **`*.example.com`**과 같이 와일드카드 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. 사이트 방문자는 이 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.

   Note

   CA는 다양한 목적을 위해 마련된 여러 형식의 파일을 보낼 수 있습니다. 본 자습서에서는 PEM 형식의 인증서 파일만 사용해야 하는데, 이는 보통 `.pem` 또는 `.crt` 확장명으로 표시되지만 항상 그런 것은 아닙니다. 어떤 파일을 사용할지 확실하지 않은 경우 텍스트 편집기로 파일을 열고 다음으로 시작되는 블록 하나 이상이 포함되는 파일을 찾습니다.

```
- - - - -BEGIN CERTIFICATE - - - - -
```

   또한 파일은 다음으로 끝나야 합니다.

```
- - - -END CERTIFICATE - - - - -
```

   또한 명령줄의 파일을 다음과 같이 테스트할 수 있습니다.

```
[ec2-user certs]$ openssl x509 -in certificate.crt -text
```

이 줄이 파일에 나타나는지 확인하십시오. `.p7b`, `.p7c`, 또는 유사한 파일 확장명으로 끝나는 파일을 사용하지 않습니다.

5. `/etc/pki/tls/certs` 디렉터리에 CA가 서명한 새 인증서와 모든 중간 인증서를 배치합니다.

   Note

   여러 가지 방법으로 사용자 지정 키를 EC2 인스턴스에 업로드할 수 있지만, 가장 간편하고 유익한 방법은 텍스트 편집기(예: vi, nano, 메모장)를 로컬 컴퓨터와 인스턴스에 모두 열고 두 편집기 간에 파일 콘텐츠를 복사하여 붙이는 것입니다. EC2 인스턴스에서 이러한 작업을 수행할 때 루트 [sudo] 권한이 필요합니다. 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 기본값(소유자=루트, 그룹=루트, 소유자 전용 읽기/쓰기)과 일치하는지 다음 명령을 사용하여 확인합니다.

```
[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, 그룹 읽기 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. 사용자 지정 키를 사용하여 CSR 및 결과로 얻은 호스트 인증서를 생성한 경우 `/etc/pki/tls/private/` 디렉터리에서 기존 키를 제거하거나 이름을 바꾼 다음 해당 디렉터리에 새 키를 설치합니다.

   Note

   여러 가지 방법으로 사용자 지정 키를 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. EC2 인스턴스에서 이러한 작업을 수행할 때 루트 [sudo] 권한이 필요합니다. 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/private` 디렉터리 내에서 파일 소유권, 그룹 및 권한 설정이 매우 제한적인 Amazon Linux 기본값(소유자=루트, 그룹=루트, 소유자 전용 읽기/쓰기)과 일치하는지 확인합니다. 명령은 다음과 같습니다.

```
[ec2-user private]$ sudo chown root.root custom.key
```

```
[ec2-user private]$ sudo chmod 600 custom.key
[ec2-user private]$ ls -al custom.key
```

위 명령의 결과는 다음과 같아야 합니다.

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

7. 새 인증서 및 키 파일을 반영하기 위해 `/etc/httpd/conf.d/ssl.conf`를 편집합니다.

   a. Apache의 `SSLCertificateFile` 명령에 CA가 서명한 호스트 인증서의 경로와 파일 이름을 입력합니다.

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

   b. 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
   ```

   > **Note**
   >
   > 일부 CA는 호스트 인증서와 중간 인증서를 단일 파일로 결합하기 때문에 이 명령이 불필요합니다. CA가 제공한 지침을 참조하십시오.

   c. Apache의 `SSLCertificateKeyFile` 명령에 프라이빗 키의 경로와 파일 이름을 입력합니다.

   ```
   SSLCertificateKeyFile /etc/pki/tls/private/custom.key
   ```

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

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

9. `https://` 접두사가 포함된 브라우저 URL 막대에 도메인 이름을 입력하여 서버를 테스트합니다. 브라우저에서는 테스트 페이지가 오류 생성 없이 HTTPS를 통해 로드되어야 합니다.

3단계: 보안 구성 테스트 및 강화

TLS이 작동되고 일반에 공개된 후 이의 실제 보안 수준을 테스트해야 합니다. 보안 설정을 무료로 완벽하게 분석해 주는 Qualys SSL Labs와 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 단원을 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. 개요에서 구성이 대체로 문제가 없어 보여도 세부 정보 보고서에서는 몇몇 잠재적 문제를 표시합니다. 예:

✗ RC4 암호는 오래된 특정 브라우저에서 사용하도록 지원됩니다. 암호는 암호화 알고리즘의 수학적 핵심입니다. TLS 데이터 스트림을 암호화하는 데 사용하는 빠른 암호인 RC4에는 몇 가지 심각한 취약점이 있는 것으로 알려져 있습니다. 타당한 레거시 브라우저 지원 사유가 없다면 비활성화해야 합니다.

✗ 이전의 TLS 버전이 지원됩니다. 구성에서는 TLS 1.0(이미 사용 중지 상태)과 TLS 1.1(사용 중지 절차 진행 중)을 지원합니다. 2018년부터는 TLS 1.2만 권장됩니다.

TLS 구성을 수정하려면 다음을 수행합니다.

1. 텍스트 편집기에서 `/etc/httpd/conf.d/ssl.conf` 구성 파일을 열고 다음 줄의 시작 부분에 "#"을 입력하여 해당 줄을 주석으로 처리합니다.

```
#SSLProtocol all -SSLv3
#SSLProxyProtocol all -SSLv3
```

2. 다음 명령을 추가합니다.

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

이러한 명령은 SSL 버전 2 및 3과 TLS 버전 1.0 및 1.1을 명시적으로 비활성화합니다. 이제 이 서버는 TLS 1.2 이외의 프로토콜을 사용하는 클라이언트와의 암호화된 연결을 허용하지 않습니다. 명령의 상세 내용은 서버의 구성 내용을 사람에게 더욱 명확히 전달합니다.

> **Note**
>
> 이러한 방식으로 TLS 버전 1.0 및 1.1을 비활성화하면 적은 비율의 오래된 웹 브라우저가 사이트에 액세스하지 못하도록 차단합니다.

허용된 암호의 목록을 수정하려면

1. `/etc/httpd/conf.d/ssl.conf` 구성 파일에서 **`SSLCipherSuite`** 및 **`SSLProxyCipherSuite`** 구성을 위한 주석 처리된 예가 포함된 섹션을 찾습니다.

```
#SSLCipherSuite HIGH:MEDIUM:!aNULL:!MD5
#SSLProxyCipherSuite HIGH:MEDIUM:!aNULL:!MD5
```

이러한 항목을 그대로 두고 그 아래에 다음 명령을 추가합니다.

> **Note**
>
> 여기에서는 가독성을 위해 여러 줄로 표시했지만, 이 두 가지 명령 각각은 암호 이름 사이에 공백이 없는 한 줄이어야 합니다.

```
SSLCipherSuite ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-
CHACHA20-POLY1305:
ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-
SHA256:ECDHE-ECDSA-AES256-SHA384:
ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256:AES:!aNULL:!
eNULL:!EXPORT:!DES:
!RC4:!MD5:!PSK:!aECDH:!EDH-DSS-DES-CBC3-SHA:!EDH-RSA-DES-CBC3-SHA:!KRB5-DES-CBC3-SHA

SSLProxyCipherSuite ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-
ECDSA-CHACHA20-POLY1305:
ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-
SHA256:ECDHE-ECDSA-AES256-SHA384:
ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-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.

- 전방향 보안 지원
- 암호화 수준

- Speed
- 암호 패밀리 앞의 특정 암호
- 거부된 암호 앞의 허용된 암호

순위가 높은 암호의 경우 이름에 ECDHE(Elliptic Curve Diffie-Hellman Ephemeral)가 있습니다. ephemeral은 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 최적화할 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 정보는 Classic Load Balancer 사용 설명서의 Predefined SSL Security Policies for Elastic Load Balancing 단원을 참조하십시오.

2. "#"을 제거하여 다음 줄의 주석 처리를 해제합니다.

```
#SSLHonorCipherOrder on
```

이 명령은 (이 예에서는) 전방향 보안을 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.

3. Apache를 다시 시작합니다. Qualys SSL Labs에서 도메인을 다시 테스트하면 RC4 취약성이 해결된 것을 알 수 있습니다.

문제 해결

- 암호를 입력하지 않으면 Apache 웹 서버가 시작하지 않습니다.

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

  키에서 암호화 및 암호 요구 사항을 제거할 수 있습니다. 기본 디렉터리에 `custom.key`라는 암호화된 프라이빗 RSA 키가 있고 이 키의 암호가 **`abcde12345`**라고 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.

인증서 자동화: Amazon Linux에서 Cerbot의 Let's Encrypt

Let's Encrypt 인증 기관은 전체 인터넷을 암호화하기 위한 EFF(Electronic Frontier Foundation) 작업의 중요한 부분입니다. 그 목표에 따라, Let's Encrypt 호스트 인증서는 사용자의 개입을 최소화하면서 생성, 검증, 설치 및 유지되도록 설계되었습니다. 인증서 관리의 자동화된 측면은 웹 서버에서 실행되는 에이전트에 의해 수행됩니다. 에이전트를 설치 및 구성한 후 Let's Encrypt와 안전하게 통신하고 Apache 및 키 관리 시스템에서 관리 작업을 수행합니다. 본 자습서에서는 사용자 지정된 암호화 키를 인증서의 기반으로 공급할 수 있거나 에이전트 자체가 기본값에 따라 키를 생성할 수 있기 때문에 무료 Certbot 에이전트를 사용합니다. 또한 Certbot이 Certbot을 자동화하려면 (p. 74)에서 설명한 대로 사용자의 상호 작용 없이 정기적으로 인증서를 갱신하도록 구성할 수 있습니다. 자세한 내용은 Certbot 사용 설명서 또는 맨 페이지를 참조하십시오.

Certbot은 Amazon Linux AMI에서 공식적으로 지원되지 않지만 다운로드가 가능하고 설치하면 제대로 작동합니다. 다음과 같이 백업을 만들어 데이터를 보호하고 불편을 피하는 것이 좋습니다.

- 시작하기 전에 Amazon EBS 루트 볼륨의 스냅샷을 만듭니다. 이를 통해 EC2 인스턴스의 원래 상태를 복원할 수 있습니다. EBS 스냅샷에 대한 자세한 내용은 Amazon EBS 스냅샷 만들기 (p. 929)을 참조하십시오.
- 아래 절차의 경우 Apache의 운영을 제어하는 `httpd.conf` 파일을 편집해야 합니다. Certbot은 이 파일과 기타 구성 파일을 고유하게 자동 변경합니다. 복원이 필요한 경우에 대비해 전체 `/etc/httpd` 디렉터리 백업 복사본을 만드십시오.

Certbot 설치 및 실행

1. 인스턴스의 Fedora 프로젝트로부터 EPEL(Extra Packages for Enterprise Linux) 리포지토리를 활성화합니다. EPEL의 패키지는 Certbot 설치 스크립트를 실행할 때 종속성으로 필요합니다.

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

2. 다음 명령을 사용하여 EFF에서 EC2 인스턴스에 Certbot 최신 릴리스를 다운로드합니다.

```
[ec2-user ~]$ wget https://dl.eff.org/certbot-auto
```

3. 다운로드한 파일을 실행 파일로 만듭니다.

```
[ec2-user ~]$ chmod a+x certbot-auto
```

4. 루트 권한 및`--debug` 플래그로 파일을 실행합니다.

```
[ec2-user ~]$ sudo ./certbot-auto --debug
```

5. "Is this ok [y/d/N]," 프롬프트에서 "y"를 입력하고 Enter 키를 누릅니다.
6. "Enter email address (used for urgent renewal and security notices)," 프롬프트에서 연락처 주소를 입력하고 Enter 키를 누릅니다.
7. 프롬프트에서 Let's Encrypt 서비스 계약 조건에 동의합니다. 진행하려면 "A"를 입력하고 Enter 키를 누릅니다.

```
-------------------------------------------------------------------------------
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.1.1-August-1-2016.pdf. You must agree
in order to register with the ACME server at
https://acme-v01.api.letsencrypt.org/directory
-------------------------------------------------------------------------------
(A)gree/(C)ancel: A
```

8. EFF 권한 부여를 클릭하면 "Y" 또는 "N"를 입력하여 메일 발송 목록에 포함되고 Enter 키를 누릅니다.
9. 아래에 표시된 프롬프트에서 일반 이름(위에서 설명한 도메인의 이름) 및 주체 대체 이름(SAN)을 입력하고 스페이스 또는 쉼표로 두 개의 이름을 분리합니다. 그런 다음 Enter 키를 누릅니다. 이 예제에서는 이름이 제공되었습니다.

```
No names were found in your configuration files. Please enter in your domain
name(s) (comma and/or space separated)  (Enter 'c' to cancel):example.com
 www.example.com
```

10. 기본 Apache 구성의 Amazon Linux 시스템에서는 아래 예제와 유사한 제공한 첫 번째 이름에 대해 묻는 출력이 표시됩니다. "1"을 입력하고 Enter 키를 누릅니다.

```
Obtaining a new certificate
Performing the following challenges:
tls-sni-01 challenge for example.com
tls-sni-01 challenge for www.example.com

We were unable to find a vhost with a ServerName or Address of example.com.
Which virtual host would you like to choose?
(note: conf files with multiple vhosts are not yet supported)
-------------------------------------------------------------------------------
1: ssl.conf                       |                       | HTTPS | Enabled
-------------------------------------------------------------------------------
Press 1 [enter] to confirm the selection (press 'c' to cancel): 1
```

11. 다음으로 Certbot은 두 번째 이름을 묻습니다. "1"을 입력하고 Enter 키를 누릅니다.

```
We were unable to find a vhost with a ServerName or Address of www.example.com.
Which virtual host would you like to choose?
(note: conf files with multiple vhosts are not yet supported)
-------------------------------------------------------------------------------
1: ssl.conf                       |                       | HTTPS | Enabled
-------------------------------------------------------------------------------
Press 1 [enter] to confirm the selection (press 'c' to cancel): 1
```

이 시점에서 Certbot이 키와 CSR을 생성합니다.

```
Waiting for verification...
Cleaning up challenges
Generating key (2048 bits): /etc/letsencrypt/keys/0000_key-certbot.pem
Creating CSR: /etc/letsencrypt/csr/0000_csr-certbot.pem
```

12. 생성할 Certbot 및 필요한 모든 호스트 인증서를 승인합니다. 각 이름을 묻는 메시지가 표시되면 예제에 나와 있는 것과 같이 "1"을 입력하고 Enter 키를 누릅니다.

```
We were unable to find a vhost with a ServerName or Address of example.com.
Which virtual host would you like to choose?
(note: conf files with multiple vhosts are not yet supported)
-------------------------------------------------------------------------------
1: ssl.conf                       |                       | HTTPS | Enabled
-------------------------------------------------------------------------------
Press 1 [enter] to confirm the selection (press 'c' to cancel): 1
Deploying Certificate for example.com to VirtualHost /etc/httpd/conf.d/ssl.conf

We were unable to find a vhost with a ServerName or Address of www.example.com.
Which virtual host would you like to choose?
(note: conf files with multiple vhosts are not yet supported)
-------------------------------------------------------------------------------
1: ssl.conf                       | example.com        | HTTPS | Enabled
-------------------------------------------------------------------------------
Press 1 [enter] to confirm the selection (press 'c' to cancel): 1
```

```
Deploying Certificate for www.example.com to VirtualHost /etc/httpd/conf.d/ssl.conf
```

13. 웹 서버에 보안성이 낮은 연결을 허용할지를 선택합니다. 예제에 나와 있는 것과 같이 옵션 2를 선택하면 모든 서버 연결이 암호화되거나 거부됩니다.

```
Please choose whether HTTPS access is required or optional.
-------------------------------------------------------------------------------
1: Easy - Allow both HTTP and HTTPS access to these sites
2: Secure - Make all requests redirect to secure HTTPS access
-------------------------------------------------------------------------------
Select the appropriate number [1-2] then [enter] (press 'c' to cancel): 2
```

Certbot은 Apache의 구성을 완료하고 성공 및 기타 정보를 보고합니다.

```
Congratulations! You have successfully enabled https://example.com and
https://www.example.com

You should test your configuration at:
https://www.ssllabs.com/ssltest/analyze.html?d=example.com
https://www.ssllabs.com/ssltest/analyze.html?d=www.example.com
-------------------------------------------------------------------------------

IMPORTANT NOTES:
 - Congratulations! Your certificate and chain have been saved at
   /etc/letsencrypt/live/example.com/fullchain.pem. Your cert will
   expire on 2017-07-19. To obtain a new or tweaked version of this
   certificate in the future, simply run certbot-auto again with the
   "certonly" option. To non-interactively renew *all* of your
   certificates, run "certbot-auto renew"
....
```

14. 설치를 완료한 후 3단계: 보안 구성 테스트 및 강화 (p. 68)에서 설명한 대로 서버의 보안을 테스트 및 최적화합니다.

Certbot은 서버 시스템의 표시되지 않고 오류를 방지하는 부분이 되도록 설계되었습니다. 기본적으로 만료 시간이 90일로 짧은 호스트 인증서를 생성합니다. 이 명령을 자동으로 호출하도록 시스템을 구성하지 않은 경우 certbot 명령을 수동으로 다시 실행해야 합니다. 이 절차는 cron 작업을 설정하여 Certbot을 자동화하는 방법을 보여줍니다.

자동화된 인증서 갱신 구성

1. Certbot을 처음으로 성공적으로 실행한 후 텍스트 편집기에서 `/etc/crontab`을 열고 다음과 유사한 줄을 추가합니다.

```
39      1,13    *       *       *       root    certbot renew --no-self-upgrade
```

작업을 마치면 파일을 저장합니다. 다음은 각 구성에 대한 설명입니다.

`39 1,13 * * *`

명령이 매일 01:39와 13:39에 실행되도록 예약합니다. 선택한 값은 임의이지만 Certbot 개발자는 최소한 매일 두 번 명령을 실행하는 것을 제안합니다. 이를 통해 손상된 것으로 발견된 모든 인증서가 즉시 취소 및 교체됩니다.

`root`

명령은 루트 권한으로 실행됩니다.

```
certbot renew --no-self-upgrade
```

실행할 명령입니다. renew 하위 명령을 사용하면 Certbot이 이전에 얻은 모든 인증서가 점검하고 만료 날짜가 다가오고 있는 인증서를 갱신합니다. `--no-self-upgrade` 플래그를 사용하면 Certbot이 사용자의 개입 없이 자체 업그레이드하지 않습니다.

2. cron 데몬을 다시 시작합니다.

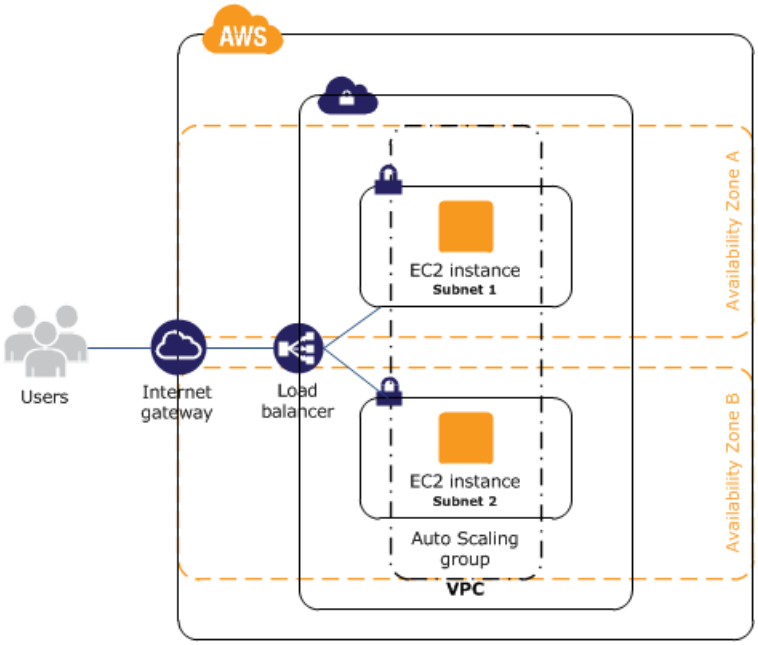
```
[ec2-user ~]$ sudo service crond restart
```

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.

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

이 자습서에서는 Amazon EC2 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 Amazon EC2 Auto Scaling 및 Elastic Load Balancing 단원을 참조하십시오.



내용

- 사전 조건 (p. 88)

- 애플리케이션 확장 및 로드 밸런싱 (p. 88)
- 로드 밸런서 테스트 (p. 90)

사전 조건

이 자습서에서는 다음을 이미 완료했다고 가정합니다.

1. 두 개 이상의 가용 영역에 있는 퍼블릭 서브넷 하나로 가상 사설 클라우드(VPC)를 생성하였습니다.
2. VPC에서 인스턴스를 시작하였습니다.
3. 인스턴스에 연결하여 인스턴스를 사용자 지정하였습니다. 소프트웨어 및 애플리케이션 설치, 데이터 복사, 추가 EBS 볼륨 연결 등을 예로 들 수 있습니다. 인스턴스에서 웹 서버를 설정하는 방법에 대한 자세한 내용은 자습서: Amazon Linux AMI를 사용하여 LAMP 웹 서버 설치 (p. 41) 단원을 참조하십시오.
4. 인스턴스에서 애플리케이션을 테스트하여 인스턴스가 올바르게 구성되었는지 확인하였습니다.
5. 인스턴스에서 사용자 지정 Amazon 머신 이미지(AMI)를 생성하였습니다. 자세한 내용은 Amazon EBS 지원 Linux AMI 생성 (p. 112) 또는 인스턴스 스토어 기반 Linux AMI 생성 (p. 115) 단원을 참조하십시오.
6. (선택 사항) 더 이상 필요하지 않은 인스턴스를 종료하였습니다.
7. 필요한 AWS에 대한 액세스 권한을 애플리케이션에 부여하는 IAM 역할을 생성하였습니다. 자세한 내용은 IAM 콘솔을 사용하여 IAM 역할을 생성하려면 다음을 수행합니다. (p. 853) 단원을 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 로드 밸런싱 아래에서 로드 밸런서를 선택합니다.
3. 로드 밸런서 생성을 선택하십시오.
4. Application Load Balancer에서 생성을 선택합니다.
5. 로드 밸런서 구성 페이지에서 다음 작업을 수행하십시오.
   a. Name에 로드 밸런서 이름을 입력합니다. 예: **`my-lb`**.
   b. 체계의 internet-facing은 기본값으로 유지합니다.
   c. 포트 80에서 HTTP 트래픽을 수락하는 리스너를 뜻하는 리스너는 기본값으로 유지합니다.
   d. 가용 영역에서 인스턴스에 사용한 VPC를 선택합니다. 가용 영역을 선택한 후 해당 가용 영역의 퍼블릭 서브넷을 선택합니다. 두 번째 가용 영역에 대해 이 절차를 반복합니다.
   e. 다음: 보안 설정 구성을 선택합니다.
6. 이 자습서의 경우 안전한 리스너를 사용하지 않고 있습니다. 다음: 보안 그룹 구성을 선택합니다.
7. 보안 그룹 구성 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. 다음: 라우팅 구성을 선택합니다.
8. 라우팅 구성 페이지에서 다음을 수행합니다.
   a. 대상 그룹에서는 기본값인 새 대상 그룹을 유지합니다.
   b. 이름에 대상 그룹의 이름을 입력합니다.

   c. 프로토콜을 HTTP로, 포트를 80으로, 대상 유형을 인스턴스로 유지합니다.
   d. 상태 검사의 기본 프로토콜과 경로를 유지합니다.
   e. Next: Register Targets(다음: 대상 등록)를 선택합니다.
9. Amazon EC2 Auto Scaling을 사용하여 EC2 인스턴스를 대상 그룹에 추가해야 하므로 대상 등록 페이지에서 다음: 검토를 선택하여 다음 페이지로 계속합니다.
10. 검토 페이지에서 생성을 선택합니다. 로드 밸런서가 생성된 후 닫기를 선택합니다.
11. 탐색 창의 AUTO SCALING에서 구성 시작을 선택합니다.
    - Amazon EC2 Auto Scaling를 처음 사용하는 경우 시작 페이지가 표시됩니다. Auto Scaling 그룹 생성을 선택하여 Auto Scaling 그룹 생성 마법사를 시작한 후 시작 구성 생성을 선택합니다.
    - 그렇지 않으면 시작 구성 생성을 선택합니다.
12. AMI 선택 페이지에서 My AMIs(내 AMI) 탭을 선택한 후 사전 조건 (p. 88)에서 생성한 AMI를 선택합니다.
13. 인스턴스 유형 선택 페이지에서 인스턴스 유형을 선택한 후 다음: 세부 정보 구성을 선택합니다.
14. 세부 정보 구성 페이지에서 다음을 수행합니다.
    a. 이름에 시작 구성의 이름을 입력합니다(예 **`my-launch-config`**).
    b. IAM 역할에서, 사전 조건 (p. 88)에서 만든 IAM 역할을 선택합니다.
    c. (선택 사항) 스타트업 스크립트를 실행해야 하는 경우 고급 세부 정보를 확장하고 사용자 데이터에 스크립트를 입력합니다.
    d. Skip to review(검토로 이동)를 선택합니다.
15. 검토 페이지에서 보안 그룹 편집을 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 검토를 선택합니다.
16. 검토 페이지에서 시작 구성 생성을 선택합니다.
17. 메시지가 나타나면 기존 키 페어를 선택하거나 새 키 페어를 만들거나 키 페어 없이 진행합니다. 승인 확인란을 선택한 다음 시작 구성 생성을 선택합니다.
18. 시작 구성이 생성된 후에는 Auto Scaling 그룹을 만들어야 합니다.
    - Amazon EC2 Auto Scaling을 처음 사용하고 Auto Scaling 그룹 생성 마법사를 사용 중인 경우 다음 단계로 자동으로 이동합니다.
    - 그렇지 않으면 이 시작 구성을 사용하여 Auto Scaling 그룹 생성을 선택합니다.
19. Auto Scaling 그룹 세부 정보 구성 페이지에서 다음 작업을 수행하십시오.
    a. 그룹 이름에 Auto Scaling 그룹의 이름을 입력합니다. 예: **`my-asg`**.
    b. 그룹 크기에 인스턴스 수를 입력합니다(예: **`2`**). 각 가용 영역에서 인스턴스 수를 대략적으로 동일하게 유지하는 것이 좋습니다.
    c. 네트워크에서 VPC를 선택하고 서브넷에서 2개의 퍼블릭 서브넷을 선택합니다.
    d. 고급 세부 정보 아래에서 하나 이상의 로드 밸런서에서 트래픽 수신을 선택합니다. 대상 그룹에서 대상 그룹을 선택합니다.
    e. Next: Configure scaling policies(다음: 조정 정책 구성)를 선택합니다.
20. Amazon EC2 Auto Scaling을 사용하여 그룹을 지정된 크기로 유지할 것이므로 Configure scaling policies(조정 정책 구성) 페이지에서 검토를 선택합니다. 나중에 이 Auto Scaling 그룹을 수동으로 확장하거나, 일정에 따라 그룹을 확장하도록 구성하거나, 필요에 따라 그룹을 확장하도록 구성할 수 있습니다.
21. 검토 페이지에서 Auto Scaling 그룹 생성을 선택합니다.
22. 그룹이 생성되면 닫기를 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 그룹 페이지에서 Auto Scaling 그룹을 선택한 후 인스턴스를 선택합니다. 처음에는 인스턴스가 `Pending` 상태로 되어 있습니다. 상태가 `InService`이면, 사용할 준비가 된 것입니다.
2. 인스턴스가 로드 밸런서에 등록되어 있는지 확인합니다. 대상 그룹 페이지에서 대상 그룹을 선택한 후 대상 탭을 선택합니다. 인스턴스의 상태가 `initial`이면, 아직 등록 중일 수도 있습니다. 인스턴스의 상태가 `healthy`이면 사용할 준비가 된 것입니다. 인스턴스가 준비되면 다음과 같이 로드 밸런서를 테스트할 수 있습니다.
3. 로드 밸런서 페이지에서 로드 밸런서를 선택합니다.
4. 설명 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 붙여넣기 하고 Enter 키를 누릅니다. 웹 사이트가 표시됩니다.

Amazon 머신 이미지(AMI)

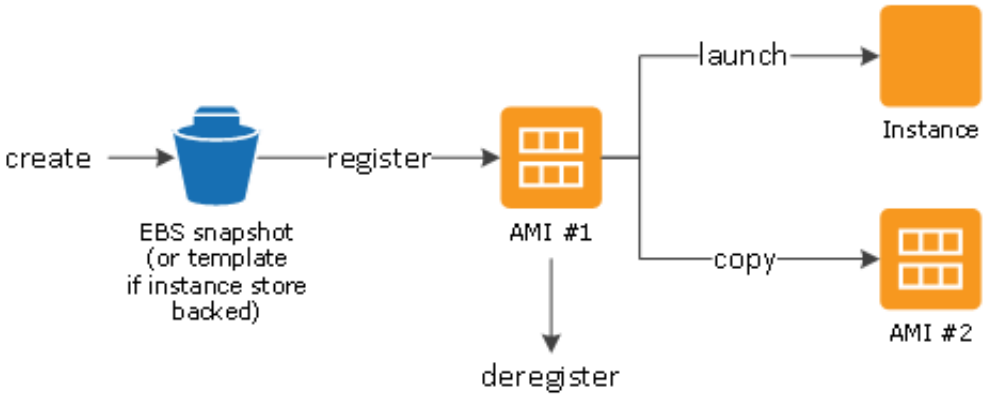
Amazon 머신 이미지(AMI)는 인스턴스를 시작하는 데 필요한 정보를 제공합니다. 인스턴스를 시작할 때 AMI를 지정해야 합니다. 동일한 구성의 인스턴스가 여러 개 필요할 때는 한 AMI에서 여러 인스턴스를 시작할 수 있습니다. 서로 다른 구성의 인스턴스가 필요할 때는 다양한 AMI를 사용하여 인스턴스를 시작하면 됩니다.

AMI는 다음을 포함합니다.

- 1개 이상의 EBS 스냅샷 또는, 인스턴스 저장 지원 AMI의 경우, 인스턴스의 루트 볼륨에 대한 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. 더 이상 필요 없는 AMI는 등록 취소할 수 있습니다.



인스턴스의 기준을 충족하는 AMI를 검색할 수 있습니다. AWS에서 제공하는 AMI 또는 커뮤니티에서 제공하는 AMI를 검색할 수 있습니다. 자세한 내용은 AMI 유형 (p. 92) 및 Linux AMI 찾기 (p. 96) 단원을 참조하십시오.

AMI에서 인스턴스 시작한 이후에 인스턴스를 연결할 수 있습니다. 인스턴스에 연결되면 사용자는 인스턴스를 다른 서버와 동일한 방식으로 사용할 수 있습니다. 인스턴스 시작, 연결 및 사용에 대한 자세한 내용은 Amazon EC2 인스턴스 (p. 178) 단원을 참조하십시오.

고유 AMI 생성

기존 AMI에서 인스턴스를 시작하고 사용자 지정한 다음, 업데이트된 이 구성을 사용자 지정 AMI로 저장할 수 있습니다. 이 새로운 사용자 지정 AMI에서 인스턴스를 시작하면 해당 AMI를 만들 때 지정한 사용자 정의 값을 포함하게 됩니다.

인스턴스의 루트 스토리지 디바이스는 어떤 프로세스로 AMI가 생성될 수 있는지를 결정합니다. 인스턴스의 루트 볼륨은 Amazon EBS 볼륨 또는 인스턴스 스토어 볼륨입니다. 루트 디바이스 볼륨에 대한 자세한 내용은 Amazon EC2 루트 디바이스 볼륨 (p. 16) 단원을 참조하십시오.

- Amazon EBS 지원 AMI를 생성하려면 Amazon EBS 지원 Linux AMI 생성 (p. 112) 단원을 참조하십시오.

- 인스턴스 스토어 지원 AMI를 생성하려면 인스턴스 스토어 기반 Linux AMI 생성 (p. 115) 단원을 참조하십시오.

AMI를 범주화하고 관리하기 위해 사용자는 AMI에 사용자 정의 태그를 할당할 수 있습니다. 자세한 내용은 Amazon EC2리소스에 태그 지정 (p. 1068) 단원을 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. 98) 단원을 참조하십시오.

Red Hat과 같은 조직의 서비스 계약에 따라 제공되는 AMI를 포함하여 타사에서 AMI를 구입할 수 있습니다. 또한, AMI를 생성한 후 다른 Amazon EC2 사용자에게 판매할 수도 있습니다. AMI 구입 및 판매에 대한 자세한 내용은 유료 AMI (p. 108) 단원을 참조하십시오.

AMI 등록 해제

관련 작업이 완료되면 AMI의 등록을 해제할 수 있습니다. 등록 취소한 AMI에서는 새 인스턴스를 시작할 수 없습니다. 그 AMI에서 시작된 기존 인스턴스에는 영향을 주지 않습니다. 자세한 내용은 Linux AMI 등록 취소 (p. 158) 단원을 참조하십시오.

Amazon Linux 2 및 Amazon Linux AMI

Amazon Linux 2 및 Amazon Linux AMI는 AWS가 제공하는 Linux 이미지로 지원 및 유지됩니다. Amazon Linux 2 및 Amazon Linux AMI가 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 리포지토리에서 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. 161) 단원을 참조하십시오.

AMI 유형

다음 유형을 기준으로 사용할 AMI를 선택할 수 있습니다.

- 리전(리전, 가용 영역 및 로컬 영역 (p. 6) 참조)
- 운영 체제
- 아키텍처(32비트 또는 64비트)
- 시작 권한 (p. 93)

- 루트 디바이스 스토리지 (p. 93)

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.

Amazon 및 Amazon EC2 커뮤니티는 퍼블릭 AMI에 대한 다양한 선택권을 제공합니다. 자세한 내용은 공유 AMI (p. 98) 단원을 참조하십시오. 개발자들은 자신의 AMI에 비용을 부과할 수 있습니다. 자세한 내용은 유료 AMI (p. 108) 단원을 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. 16) 단원을 참조하십시오.

다음 표에는 두 가지 유형의 AMI를 사용할 때 주요 차이점이 요약되어 있습니다.

특성	Amazon EBS 지원 AMI	Amazon 인스턴스 스토어 지원 AMI
인스턴스의 부팅 시간	일반적으로 1분 이하	일반적으로 5분 이하
루트 디바이스의 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.
수정	인스턴스 유형, 커널 RAM 디스크 및 사용자 데이터는 인스턴스가 중지된 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.
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 볼륨 지속 설정 (p. 19) 단원을 참조하십시오.

AMI의 루트 디바이스 유형 결정

콘솔을 이용하여 AMI의 루트 디바이스 유형을 결정하려면

1. Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 AMI를 클릭한 후 AMI를 선택합니다.
3. 세부 정보 탭에서 루트 디바이스 유형의 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. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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 인스턴스는 중지할 수 없기 때문에 실행 중 또는 종료됨 상태입니다. 인스턴스 중지로 인한 영향 및 해결 방법에 대한 자세한 내용은 인스턴스 중지 및 시작 (p. 507) 단원을 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). 1개 이상의 Amazon EBS 볼륨을 연결하여 인스턴스에 영구 스토리지를 추가할 수 있습니다. 인스턴스 스토어 볼륨의 모든 데이터는 인스턴스가 장애를 일으키거나 종료되면 삭제됩니다. 자세한 내용은 인스턴스 스토어 수명 (p. 1027) 단원을 참조하십시오.

Amazon EBS를 루트 디바이스로 사용하는 인스턴스에는 자동으로 Amazon EBS 볼륨이 연결됩니다. 이 볼륨은 다른 볼륨과 마찬가지로 볼륨 목록에 표시됩니다. 대부분의 인스턴스 유형에서는 Amazon EBS 지원 인스턴스에 기본적으로 인스턴스 스토어 볼륨이 없습니다. 인스턴스 스토어 볼륨 또는 추가 Amazon EBS 볼륨은 블록 디바이스 매핑을 이용하여 추가될 수 있습니다. 자세한 내용은 블록 디바이스 매핑 (p. 1049) 단원을 참조하십시오.

부팅 시간

Amazon EBS 지원 AMI에서 시작된 인스턴스는 인스턴스 스토어 지원 AMI에서 시작된 인스턴스보다 빠르게 시작됩니다. 인스턴스 스토어 지원 AMI에서 인스턴스를 시작하는 경우 인스턴스가 사용 가능하게 되려면 먼저 Amazon S3에서 모든 부분을 가져와야 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 중지한 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` API 작업을 통해 Amazon EBS 지원 AMI를 생성하고 등록할 수 있습니다. 또한 AWS Management 콘솔에는 실행 상태의 AMI를 생성하는 버튼이 있습니다. 자세한 내용은 Amazon EBS 지원 Linux AMI 생성 (p. 112) 단원을 참조하십시오.

요금 부과 방법

인스턴스 스토어 지원 AMI의 경우 AMI 스토리지 및 인스턴스 사용 및 Amazon S3에 AMI 저장에 대해 요금이 부과됩니다. Amazon EBS 지원 AMI를 사용하는 경우 인스턴스 사용, Amazon EBS 볼륨 스토리지 및 사용, AMI를 Amazon EBS 스냅샷으로 저장에 대해 요금이 부과됩니다.

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 시작하는 즉시 최소 1분의 사용 요금이 부과됩니다. 1분 이후에는 사용한 시간(초)에 대해서만 요금이 부과됩니다. 예를 들어 인스턴스를 20초간 실행한 후 중지했다면 1분에 대한 요금이 부과됩니다. 인스턴스를 3분 40초간 실행한 경우 정확히 3분 40초에 대한 요금이 부과됩니다. 인스턴스를 실행 중 상태로 유지하는 동안에는 인스턴스가 유휴 상태로 남아 있고 인스턴스에 연결하지 않더라도 최소 1분 요금과 함께 초 단위로 요금이 부과됩니다.

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 유형 정보 페이지를 참조하십시오. 이전 세대 인스턴스 유형을 사용 중인 경우 업그레이드하려면 자세한 정보는 업그레이드 경로를 참조하십시오.

HVM AMI

HVM AMIs는 이미지 루트 블록 디바이스의 마스터 부트 레코드를 실행하여 완벽하게 가상화된 하드웨어 및 부트 세트를 함께 제공합니다. 이 가상화 유형은 운영 체제 미설치 하드웨어에서 실행될 때처럼 가상 머신에서 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 Technology를 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 Linux에서 향상된 네트워킹 (p. 706) 및 Linux 액셀러레이티드 컴퓨팅 인스턴스 (p. 248) 단원을 참조하십시오.

모든 인스턴스 유형이 HVM AMI를 지원합니다.

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

PV AMI

PV AMIs는 PV-GRUB라는 특수 부트 로더를 통해 부팅되며, 이 로더는 부팅 주기를 시작한 후 사용자 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 자체 Linux 커널 활성화 (p. 171) 단원을 참조하십시오.

C1, C3, HS1, M1, M3, M2, T1 등과 같은 전 세대 인스턴스 유형은 PV AMI를 지원합니다. 최신 세대 인스턴스 유형은 PV AMI를 지원하지 않습니다.

PV 인스턴스를 지원하는 AWS 리전: 아시아 태평양(도쿄), 아시아 태평양(싱가포르), 아시아 태평양(시드니), 유럽(프랑크푸르트), 유럽(아일랜드), 남아메리카(상파울루), 미국 동부(버지니아 북부), 미국 서부(캘리포니아 북부 지역), 미국 서부(오레곤)

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 체제에서도 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 요구사항을 고려하십시오.

- 리전
- 운영 체제
- 아키텍처: 32비트(`i386`), 64비트(`x86_64`) 또는 64비트 ARM(arm64)
- 루트 디바이스 유형: Amazon EBS 또는 인스턴스 스토어
- 공급자(예: Amazon Web Services)
- 추가 소프트웨어(예: SQL server)

Windows AMI를 찾아야 하는 경우 Windows 인스턴스용 Amazon EC2 사용 설명서의 Windows AMI 찾기 단원을 참조하십시오.

내용

- Linux 콘솔을 사용하여 Amazon EC2 AMI 찾기 (p. 97)
- AWS CLI를 사용하여 AMI 찾기 (p. 97)
- 시스템 관리자를 사용하여 최신 Amazon Linux AMI 찾기 (p. 98)
- 빠른 시작 AMI 검색 (p. 98)

Linux 콘솔을 사용하여 Amazon EC2 AMI 찾기

Linux 콘솔을 사용하여 Amazon EC2 AMI를 찾을 수 있습니다. 이미지 페이지에서 사용 가능한 모든 AMI를 검색하거나, 콘솔에서 이미지를 시작할 때 빠른 시작 탭에서 자주 사용되는 AMI를 선택할 수 있습니다. AMI ID는 리전마다 고유합니다.

Choose AMI(AMI 선택) 페이지에서 Linux AMI를 찾으려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 모음에서 인스턴스를 실행할 리전을 선택합니다. 현재 위치와 관계없이 사용자가 고를 수 있는 리전을 임의로 선택합니다.
3. 콘솔 대시보드에서 인스턴스 시작을 선택합니다.
4. 빠른 시작 탭의 목록에서 자주 사용되는 AMI 중 하나를 선택합니다. 필요한 AMI가 표시되지 않는 경우 AWS Marketplace 또는 커뮤니티 AMI 탭을 선택하여 추가 AMI를 찾습니다.

이미지 페이지를 사용하여 Linux AMI를 찾으려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 모음에서 인스턴스를 실행할 리전을 선택합니다. 현재 위치와 관계없이 사용자가 고를 수 있는 리전을 임의로 선택합니다.
3. 탐색 창에서 AMI를 선택합니다.
4. (선택 사항) 필터 옵션을 사용하여 원하는 AMI만 표시하도록 표시되는 AMI 목록의 범위를 지정합니다. 예를 들어, AWS에서 제공하는 모든 Linux AMI를 나열하려면 퍼블릭 이미지를 선택합니다. 검색 창을 선택하고 메뉴에서 소유자를 선택한 다음 Amazon 이미지를 선택합니다. 검색 창을 다시 선택하고 플랫폼을 선택한 다음 제공된 목록에서 운영 체제를 선택합니다.
5. (선택 사항) 열 표시/숨기기 아이콘을 선택하여 표시할 이미지 속성(예: 루트 디바이스 유형)을 선택합니다. 또는 목록에서 AMI를 선택하고 세부 정보 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. 93) 단원을 참조하십시오.
7. 이 AMI에서 인스턴스를 시작하려면 원하는 인스턴스를 선택한 다음 시작을 선택합니다. 콘솔을 통한 인스턴스 시작에 대한 자세한 내용은 AMI에서 인스턴스 시작 (p. 433) 단원을 참조하십시오. 인스턴스를 시작할 준비가 되지 않은 경우, 나중을 위해 AMI ID를 기록해 둡니다.

AWS CLI를 사용하여 AMI 찾기

Amazon EC2의 AWS CLI 명령을 사용하여 필요에 맞는 Linux AMI만 나열되도록 할 수 있습니다. 필요에 맞는 AMI를 찾았으면 인스턴스를 시작할 때 사용할 수 있도록 ID를 기록해 둡니다. 자세한 내용은 AWS Command Line Interface 사용 설명서의 AWS CLI를 사용한 인스턴스 시작하기 단원을 참조하십시오.

describe-images 명령은 파라미터 필터링을 지원합니다. 예를 들어, Amazon 소유의 퍼블릭 AMI를 표시하려면 `--owners` 파라미터를 사용합니다.

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

Amazon EBS 기반 AMI만 표시하려면 이전 명령에 다음 필터를 추가합니다.

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

> **Important**
>
> describe-images 명령에서 `--owners` 플래그가 누락되면 소유 여부와 관계없이 시작 권한을 보유한 모든 이미지를 반환합니다.

시스템 관리자를 사용하여 최신 Amazon Linux AMI 찾기

AWS 시스템 관리자 Parameter Store에서 최신 Amazon Linux AMI의 ID를 쿼리할 수 있습니다. 자세한 내용은 AWS 시스템 관리자 Parameter Store를 사용하여 최신 Amazon Linux AMI ID 쿼리를 참조하십시오.

빠른 시작 AMI 검색

Amazon EC2 콘솔을 사용하여 인스턴스를 시작하는 경우 Amazon 머신 이미지(AMI) 선택 페이지의 빠른 시작 탭에는 자주 사용되는 AMI 목록이 포함되어 있습니다. 이러한 빠른 시작 AMI 중 하나를 사용하여 인스턴스 시작을 자동화하려면 AMI 현재 버전의 ID를 프로그래밍 방식으로 찾아야 합니다.

빠른 시작 AMI 현재 버전을 찾으려면 AMI 이름이 있는 모든 AMI를 나열한 다음, 생성 날짜가 최근인 AMI를 검색하면 됩니다.

Example 예: 현재 Amazon Linux 2 AMI 검색

```
aws ec2 describe-images --owners amazon --filters 'Name=name,Values=amzn2-
ami-hvm-2.0.????????.?-x86_64-gp2' 'Name=state,Values=available' --query
 'reverse(sort_by(Images, &CreationDate))[:1].ImageId' --output text
```

Example 예: 현재 Amazon Linux AMI 검색

```
aws ec2 describe-images --owners amazon --filters 'Name=name,Values=amzn-
ami-hvm-????.??.?.????????-x86_64-gp2' 'Name=state,Values=available' --query
 'reverse(sort_by(Images, &CreationDate))[:1].ImageId' --output text
```

Example 예: 현재 Ubuntu Server 16.04 LTS AMI 검색

```
aws ec2 describe-images --owners 099720109477 --filters 'Name=name,Values=ubuntu/images/
hvm-ssd/ubuntu-xenial-16.04-amd64-server-????????' 'Name=state,Values=available' --query
 'reverse(sort_by(Images, &CreationDate))[:1].ImageId' --output text
```

Example 예: 현재 Red Hat Enterprise Linux 7.5 AMI 검색

```
aws ec2 describe-images --owners 309956199498 --filters 'Name=name,Values=RHEL-7.5_HVM_GA*'
 'Name=state,Values=available' --query 'reverse(sort_by(Images, &CreationDate))
[:1].ImageId' --output text
```

Example 예: 현재 SUSE Linux Enterprise Server 15 AMI 검색

```
aws ec2 describe-images --owners amazon --filters 'Name=name,Values=suse-sles-15-
v????????-hvm-ssd-x86_64' 'Name=state,Values=available' --query 'reverse(sort_by(Images,
 &CreationDate))[:1].ImageId' --output text
```

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 요소을 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 코드

를 배포하는 경우와 마찬가지로 이런 AMI를 취급하고 그에 따라 적합한 조치를 취해야 합니다. 신뢰할 수 있는 출처의 AMI를 사용하십시오.

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

AMI 생성에 대한 자세한 정보는 인스턴스 스토어 지원 Linux AMI 생성 또는 Amazon EBS 지원 Linux AMI 생성 을 참조하십시오. AWS Marketplace에서 애플리케이션을 빌드하고 제공하며 관리하는 정보는 AWS Marketplace 설명서를 참조하십시오.

목차

- 공유 AMI 검색 (p. 99)
- 퍼블릭 AMI 설정 (p. 101)
- 지정한 AWS 계정과 AMI 공유 (p. 102)
- 북마크 사용 (p. 103)
- 공유 Linux AMI 지침 (p. 104)

공유 AMI 검색

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

> **Note**
>
> AMI는 리전 리소스입니다. 따라서 공유 AMI(퍼블릭 또는 프라이빗)를 검색할 경우에는 공유되고 있는 리전 내에서 검색해야 합니다. AMI를 다른 리전에서 사용할 수 있도록 하려면 AMI를 해당 리전에 복사한 후 공유하십시오. 자세한 내용은 AMI 복사를 참조하십시오.

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 AMI를 선택합니다.
3. 프라이빗 이미지를 첫 필터로 선택합니다. 사용자와 공유된 모든 AMI가 나열됩니다. 검색 결과를 좀 더 세부적으로 보려면 검색 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 AMI를 선택합니다.
3. 퍼블릭 이미지를 첫 필터로 선택합니다. 검색 결과를 좀 더 세부적으로 보려면 검색 창을 선택하여 메뉴에서 제공하는 필터 옵션을 사용하십시오.
4. 필터를 사용하면 원하는 유형의 AMI만 검색할 수 있습니다. 예를 들어 소유자 :]를 선택한 다음 Amazon 이미지를 선택하면 Amazon의 퍼블릭 이미지만 표시됩니다.

공유 AMI를 검색하는 방법(AWS CLI)

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

예: 모든 퍼블릭 AMI 나열

다음 명령은 사용자가 소유한 퍼블릭 AMI를 포함한 모든 퍼블릭 AMI를 나열합니다.

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

예: 명시적 시작 권한으로 AMI 나열

다음 명령은 사용자가 명시적 시작 권한을 가지고 있는 AMI를 나열합니다. 이 목록에는 사용자가 소유한 AMI는 포함되지 않습니다.

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

예: Amazon에서 소유한 AMI 나열

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

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

예: 계정에서 소유한 AMI 나열

다음 명령은 지정된 AWS 계정에서 소유한 AMI를 나열합니다.

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

예: 필터를 사용하여 AMI 범위 지정

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"
```

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 AMI인지를 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. 다음 명령은 `authorized_keys` 파일을 찾습니다.

   ```
   [ec2-user ~]$ sudo find / -name "authorized_keys" -print -exec cat {} \;
   ```

2. 루트 사용자의 암호 방식 인증을 비활성화합니다. `sshd_config` 파일을 열고 `PermitRootLogin` 줄을 다음과 같이 편집합니다.

   ```
   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. 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. 다음 예를 참조하십시오.

```
[ec2-user ~]$ sudo rm /etc/rsyslog.config
[ec2-user ~]$ 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 단원을 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. 102) 참조). 다른 AWS 계정에서 공유 AMI를 사용하면 관련 요금은 공유 AMI를 시작한 해당 계정에만 청구됩니다.

암호화된 볼륨이 있는 AMI는 퍼블릭으로 설정할 수 없습니다.

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. 150) 단원을 참조하십시오.

AMI를 공유할 때 민감한 데이터의 유출을 방지하려면 공유 Linux AMI 지침 (p. 104)에 설명된 보안상 고려 사항을 확인하여 권장 조치를 따르십시오.

> **Note**
>
> AMI에 제품 코드가 있거나 암호화된 볼륨의 스냅샷이 포함된 경우, 이 AMI를 퍼블릭으로 설정할 수 없습니다. AMI는 특정 AWS 계정과만 공유할 수 있습니다.

모든 AWS 계정과 AMI 공유(콘솔)

AMI를 공개한 후 콘솔을 사용하여 동일한 리전에서 인스턴스를 시작하면 커뮤니티 AMI에서 AMI를 사용할 수 있습니다. AMI 공개 후 AMI가 커뮤니티 AMI에 표시되는 데 약간의 시간이 걸릴 수 있다는 점에 유의하십시오. AMI를 다시 비공개로 바꾼 후 AMI가 커뮤니티 AMI에서 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 AMI를 선택합니다.
3. 목록에서 AMI를 선택한 후 작업에서 이미지 권한 수정을 선택합니다.
4. 퍼블릭을 선택한 다음 저장을 선택합니다.

모든 AWS 계정과 AMI 공유(AWS CLI)

각 AMI에는 소유자를 제외하고 해당 AMI를 사용한 인스턴스 시작이 허용된 `launchPermission` 계정을 설정할 수 있는 AWS 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

1. 다음과 같이 modify-image-attribute 명령을 실행하고 지정한 AMI의 `launchPermission` 목록에 `all` 그룹을 추가합니다.

```
aws ec2 modify-image-attribute --image-id ami-0abcdef1234567890 --launch-permission
 "Add=[{Group=all}]"
```

2. AMI의 시작 권한을 확인하려면 describe-image-attribute 명령을 사용합니다.

```
aws ec2 describe-image-attribute --image-id ami-0abcdef1234567890 --attribute
 launchPermission
```

3. (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-0abcdef1234567890 --launch-permission
 "Remove=[{Group=all}]"
```

지정한 AWS 계정과 AMI 공유

AMI를 퍼블릭으로 설정하지 않고 지정한 AWS 계정과 공유할 수 있습니다. 이런 작업은 AWS 계정 ID만 있으면 가능합니다. AMI를 암호화된 볼륨과 공유할 경우 AMI의 암호화에 사용되는 모든 CMK도 공유해야 합니다. 자세한 내용은 Amazon EBS 스냅샷 공유 (p. 938) 단원을 참조하십시오.

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. 150) 단원을 참조하십시오.

AMI에서 공유할 수 있는 AWS 계정 수에는 제한이 없습니다.

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 AMI를 선택합니다.
3. 목록에서 AMI를 선택한 후 작업에서 이미지 권한 수정을 선택합니다.
4. AWS 계정 번호 필드에 AMI를 공유할 AWS 계정 번호를 지정하고 권한 추가를 선택합니다.

   이 AMI를 다수의 사용자와 공유하려면 이 단계를 반복하여 사용자를 추가합니다.
5. 스냅샷 볼륨 권한 생성을 허용하려면 권한 생성 시 연결된 스냅샷에 대해 "볼륨 생성" 권한 추가.(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. 그러나 AMI가 참조하는 스냅샷을 암호화하는 데 사용되는 모든 CMK를 공유해야 합니다. 자세한 내용은 Amazon EBS 스냅샷 공유 (p. 938) 단원을 참조하십시오.
6. 모두 완료했으면 저장을 선택합니다.

7. (선택 사항) AMI를 공유하는 AWS 계정 ID를 보려면 목록에서 AMI를 선택하고 권한 탭을 선택합니다. 공유되는 AMI를 찾으려면 공유 AMI 검색 (p. 99) 단원을 참조하십시오.

AMI를 공유하는 방법(AWS CLI)

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-0abcdef1234567890 --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
```

> **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. 그러나 AMI가 참조하는 스냅샷을 암호화하는 데 사용되는 모든 CMK를 공유해야 합니다. 자세한 내용은 Amazon EBS 스냅샷 공유 (p. 938) 단원을 참조하십시오.

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-0abcdef1234567890 --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-0abcdef1234567890 --attribute launchPermission
```

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.

AMI는 반드시 퍼블릭 AMI이거나 북마크를 보낼 사용자와 공유된 상태여야 합니다.

AMI 북마크 생성

1. 다음 정보를 참고하여 URL을 입력합니다. 여기에서 리전은 AMI가 속하는 리전입니다.

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

예를 들어, 다음 URL은 ami-0abcdef1234567890 리전의 us-east-1 AMI에서 인스턴스를 실행합니다.

```
https://console.aws.amazon.com/ec2/v2/home?region=us-
east-1#LaunchInstanceWizard:ami=ami-0abcdef1234567890
```

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.

주제

- AMI 도구를 사용하기 전에 업데이트 (p. 104)
- 루트 사용자의 암호 방식 원격 로그인 비활성화 (p. 105)
- 로컬 루트 액세스 비활성화 (p. 105)
- SSH 호스트 키 페어 삭제 (p. 105)
- 퍼블릭 키 자격 증명 프로그램 설치 (p. 106)
- sshd DNS 확인 비활성화(선택 사항) (p. 107)
- 본인 인증 (p. 107)
- 보안 (p. 107)

AWS Marketplace용 AMI를 구축하는 경우에는 AWS Marketplace용 AMI 구축의 지침과 정책, 모범 사례를 참조하십시오.

안전한 AMI 공유에 대한 추가 내용은 다음을 참조하십시오.

- 안전한 방식으로 퍼블릭 AMI를 공유하고 사용하는 방법
- 퍼블릭 AMI 게시: 강화 및 정리 요구 사항

AMI 도구를 사용하기 전에 업데이트

인스턴스 스토어 지원 AMI의 경우, 사용하기 전에 Amazon EC2 AMI 생성 도구를 다운로드 및 업그레이드하는 것을 권장합니다. 이를 통해 공유 AMI를 기반으로 한 새 AMI에서 최신 AMI 도구를 사용할 수 있습니다.

Amazon Linux 2의 경우 `aws-amitools-ec2` 패키지를 설치하고 다음 명령과 함께 사용자의 PATH에 AMI 도구를 추가하십시오. Amazon Linux AMI의 경우 `aws-amitools-ec2` 패키지가 기본적으로 설치되어 있습니다.

```
[ec2-user ~]$ sudo yum install -y aws-amitools-ec2 && export PATH=$PATH:/opt/aws/bin > /
etc/profile.d/aws-amitools-ec2.sh && . /etc/profile.d/aws-amitools-ec2.sh
```

다음 명령으로 AMI 도구를 업그레이드합니다.

```
[ec2-user ~]$ sudo yum upgrade -y aws-amitools-ec2
```

기타 배포의 경우에는 AMI 도구를 항상 최신으로 유지합니다.

루트 사용자의 암호 방식 원격 로그인 비활성화

퍼블릭 AMI에 대하여 고정 루트 암호를 사용하면 보안 위험을 촉진하는 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/ssh`에 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 다음 명령을 실행하여 안전하게 제거할 수 있습니다.

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

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** 설명서를 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.

SSH을 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 시작 시 루트 사용자에 대해 지정한 퍼블릭 키를 가져오도록 설정할 수 있습니다.

IMDSv2

```
if [ ! -d /root/.ssh ] ; then
        mkdir -p /root/.ssh
        chmod 700 /root/.ssh
fi
# Fetch public key using HTTP
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-
token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-
data/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
```

IMDSv1

```
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 forum에 AMI 설명과 AMI ID를 게시하는 것을 권장합니다. 새로운 공유 AMI에 흥미를 가지고 시도하려는 사용자에게 유용한 중심 자료소를 제공하기 때문입니다.

보안

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.

- `--exclude` *`directory`*에서 `ec2-bundle-vol` 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 이들 파일을 루트 계정의 경우 `/root/.ssh`에 저장하고 일반 사용자 계정의 경우 `/home/user_name/.ssh/`에 저장합니다. 자세한 내용은 ec2-bundle-vol (p. 132) 단원을 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.
> bash는 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 수 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 AWS Marketplace에서의 판매를 참조하십시오. AWS Marketplace는 Amazon EBS 기반 AMI를 지원합니다.

내용

- AMI 판매 (p. 109)
- 유료 AMI 찾기 (p. 109)
- 유료 AMI 구입 (p. 110)
- 인스턴스에 대한 제품 코드 가져오기 (p. 110)
- 유료 지원 사용 (p. 110)
- 유료 및 지원된 AMI에 대한 청구서 (p. 111)
- AWS Marketplace 구독 관리 (p. 111)

AMI 판매

AWS Marketplace를 사용하여 AMI를 판매할 수 있습니다. AWS Marketplace는 조직적인 쇼핑 환경을 제공합니다. 또한 AWS Marketplace에서는 Amazon EBS 지원 AMI, 예약 인스턴스 및 스팟 인스턴스와 같은 AWS 기능도 지원합니다.

AWS Marketplace에서 AMI를 판매하는 방법에 대한 자세한 내용은 AWS Marketplace에서의 판매 단원을 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. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 AMI를 선택합니다.
3. 퍼블릭 이미지를 첫 필터로 선택합니다.
4. 검색 창에서 소유자, AWS Marketplace를 차례로 선택합니다.
5. 제품 코드를 알고 있는 경우 제품 코드를 선택한 다음 제품 코드를 입력합니다.

AWS Marketplace를 사용하여 유료 AMI 찾기

AWS Marketplace를 사용하여 유료 AMI를 찾으려면

1. Open AWS Marketplace.
2. 검색 상자에 운영 체제의 이름을 입력하고 이동을 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.

AWS CLI를 사용하여 유료 AMI 찾기

다음 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"
    }
],
```

제품 코드를 알고 있는 경우 제품 코드별로 결과를 필터링할 수 있습니다. 이 예제는 지정된 제품 코드가 포함된 최신 AMI를 반환합니다.

```
aws ec2 describe-images --owners aws-marketplace \
--filters "Name=product-code,Values=product_code" \
--query "sort_by(Images, &CreationDate)[-1].[ImageId]"
```

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. 448) 단원을 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. 448) 단원을 참조하십시오.

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. 571) 단원을 참조하십시오.

제품 코드를 검색하려면 다음 명령을 사용합니다.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-
ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-
data/product-codes
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/product-codes
```

인스턴스에 제품 코드가 있는 경우 Amazon EC2에서 해당 코드를 반환합니다.

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)

```
PS 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 Products를 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 Marketplace에 로그인합니다.
2. Marketplace 계정(Your Marketplace Account)을 선택합니다.
3. 소프트웨어 구독 관리(Manage your software subscriptions)를 선택합니다.
4. 현재 구독이 모두 나열됩니다. 사용량 제한을 선택하여 제품 사용에 대한 특정 지침(예: 실행 중인 인스턴스에 연결하기 위한 사용자 이름)을 봅니다.

AWS Marketplace 구독을 취소하려면

1. 구독에서 실행 중인 모든 인스턴스를 종료해야 합니다.
   a. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
   b. 탐색 창에서 인스턴스를 선택합니다.
   c. 인스턴스를 선택하고 작업, 인스턴스 상태, 종료를 선택합니다.
   d. 확인 메시지가 나타나면 예, 종료를 선택합니다.
2. AWS Marketplace에 로그인하고 Marketplace 계정(Your Marketplace Account)을 선택한 후 소프트웨어 구독 관리(Manage your software subscriptions)를 선택합니다.
3. 구독 취소를 선택합니다. 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 받은 AMI, AWS Server Migration Service 또는 VM Import/Export를 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.

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

AMI 생성 프로세스는 인스턴스 스토어 지원 AMIs의 경우와는 다릅니다. Amazon EBS 지원 인스턴스와 인스턴스 스토어 지원 인스턴스 간의 차이점에 대한 자세한 정보와 인스턴스의 루트 디바이스 유형을 확인하는 방법은 루트 디바이스 스토리지 (p. 93) 단원을 참조하십시오. 인스턴스 스토어 지원 Linux AMI 생성에 대한 자세한 내용은 인스턴스 스토어 기반 Linux AMI 생성 (p. 115)을 참조하십시오.

Amazon EBS 지원 Windows AMI 생성에 대한 자세한 내용은 Windows 인스턴스용 Amazon EC2 사용 설명서의 Amazon EBS 지원 Windows AMI 생성을 참조하십시오.

Amazon EBS 지원 AMIs 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. AMI 등록을 해제하고 스냅샷을 삭제할 때까지는 스냅샷에 대한 요금이 부과됩니다. 자세한 내용은 Linux AMI 등록 취소 (p. 158) 단원을 참조하십시오. 인스턴스에 연결된 볼륨이 암호화된 경우 새 AMI는 Amazon EBS 암호화을 지원하는 인스턴스에서만 시작됩니다. 자세한 내용은 Amazon EBS Encryption (p. 968) 단원을 참조하십시오.

볼륨의 크기에 따라 AMI 생성 프로세스를 완료하는 데 몇 분 정도 걸리지만 경우에 따라 24시간까지 걸릴 수도 있습니다. AMI를 생성하기 전에 볼륨의 스냅샷을 생성하는 것이 더 효율적일 수 있습니다. 이처럼 AMI를 생성할 때 작은 증분적 스냅샷만 만들어야 프로세스가 더 빠르게 완료됩니다. 스냅샷을 만드는 데 걸리는 전체 시간은 동일하게 유지됩니다. 자세한 내용은 Amazon EBS 스냅샷 만들기 (p. 929) 단원을 참조하십시오.

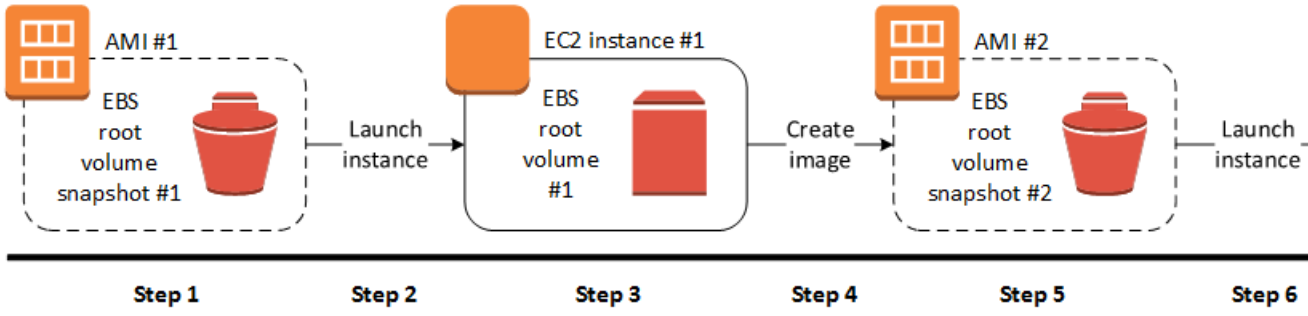
프로세스가 완료되면 인스턴스의 루트 볼륨에서 새 AMI 및 스냅샷이 생성됩니다. 새 AMI를 사용하여 인스턴스를 시작할 때 스냅샷을 사용하여 루트 볼륨에 대한 새 EBS 볼륨을 생성합니다.

루트 디바이스 볼륨 외에도 인스턴스에 인스턴스 스토어 볼륨이나 EBS 볼륨을 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. 1049) 단원을 참조하십시오.

> **Note**
>
> EBS 지원 AMI에서 새 인스턴스를 만들 때는 루트 볼륨과 추가 EBS 저장소를 모두 프로덕션 환경에 배치하기 전에 초기화해야 합니다. 자세한 정보는 Amazon EBS 볼륨 초기화를 참조하십시오.

인스턴스에서 Linux AMI 생성

AWS Management 콘솔 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

1. 새 AMI의 시작점으로 사용할 적절한 EBS 지원 AMI를 선택하고 시작하기 전에 필요에 따라 구성합니다. 자세한 내용은 인스턴스 시작 마법사를 사용하여 인스턴스 시작 (p. 431) 단원을 참조하십시오.
2. 시작을 선택하여 선택한 EBS 지원 AMI의 인스턴스를 시작합니다. 나머지 기본값을 그대로 두고 마법사를 계속 진행합니다. 자세한 내용은 인스턴스 시작 마법사를 사용하여 인스턴스 시작 (p. 431) 단원을 참조하십시오.
3. 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
4. (선택 사항) 인스턴스에 연결한 모든 볼륨의 스냅샷을 생성합니다. 스냅샷 생성에 대한 자세한 내용은 Amazon EBS 스냅샷 만들기 (p. 929) 단원을 참조하십시오.
5. 탐색 창에서 인스턴스를 선택하고 인스턴스를 선택한 후 작업, 이미지, 이미지 생성을 선택합니다.

   Tip

   이 옵션이 비활성화되어 있다면 Amazon EBS 지원 인스턴스가 아님을 의미합니다.
6. 이미지 생성 대화 상자에서 다음 정보를 지정한 후 이미지 생성을 선택합니다.
   - 이미지 이름 – 이미지의 고유한 이름입니다.
   - 이미지 설명 – 이미지에 대한 선택적 설명이며 최대 255자까지 입력할 수 있습니다.
   - 재부팅 안 함 – 기본적으로 이 옵션은 선택되지 않습니다. Amazon EC2는 인스턴스를 종료하고, 연결된 볼륨의 스냅샷을 캡처하고, AMI를 생성하여 등록한 후 인스턴스를 재부팅합니다. 인스턴스가 종료되지 않도록 하려면 재부팅 안 함을 선택합니다.

     Warning

     재부팅 안 함을 선택한 경우 생성된 이미지의 파일 시스템 무결성을 보장할 수 없습니다.
   - 인스턴스 볼륨 – 이 섹션의 필드에서는 루트 볼륨을 수정하고 추가 Amazon EBS 및 인스턴스 스토어 볼륨을 추가할 수 있습니다. 각 필드에 대한 자세한 내용을 보려면 각 필드 옆에 있는 i 아이콘에서 일시 중지하여 필드 도구 설명을 표시합니다. 몇 가지 중요한 사항은 아래에 나열되어 있습니다.
     - 루트 볼륨의 크기를 변경하려면 볼륨 유형 열의 루트로 이동한 후 크기(GiB)에 필요한 값을 입력합니다.

- 종료 시 삭제 여부를 선택할 경우 이 AMI에서 생성된 인스턴스를 종료하면 EBS 볼륨이 삭제됩니다. 종료 시 삭제 여부를 선택 취소할 경우 인스턴스를 종료하면 EBS 볼륨이 삭제되지 않습니다. 자세한 정보는 인스턴스 종료 시 Amazon EBS 볼륨 보존 (p. 526) 단원을 참조하십시오.
- Amazon EBS 볼륨을 추가하려면 새 볼륨 추가를 선택하여 새 행을 추가합니다. 볼륨 유형에서 EBS를 선택하고 행의 필드를 작성합니다. 새 AMI에서 인스턴스를 시작하면 추가 볼륨이 인스턴스에 자동으로 연결됩니다. 빈 볼륨은 반드시 포맷하고 마운트해야 합니다. 스냅샷 기반 볼륨을 반드시 마운트해야 합니다.
- 인스턴스 스토어 볼륨을 추가하려면 AMI에 인스턴스 스토어 볼륨 추가 (p. 1033)를 참조하십시오. 새 AMI에서 인스턴스를 시작하면 추가 볼륨이 자동으로 시작되어 탑재됩니다. 이러한 볼륨에는 AMI를 기반으로 하는 실행 중인 인스턴스에 대한 인스턴스 스토어 볼륨의 데이터가 포함되어 있지 않습니다.

7. 생성 중인 AMI의 상태를 보려면 탐색 창에서 AMI를 선택합니다. 처음에 상태는 `pending`이지만 몇 분 후에 `available`로 변경되어야 합니다.

   (선택 사항) 새 AMI에 대해 생성된 스냅샷을 보려면 스냅샷을 선택합니다. 이 AMI에서 인스턴스를 시작할 때 이 스냅샷을 사용하여 루트 디바이스 볼륨을 생성합니다.
8. 새 AMI에서 인스턴스를 시작합니다. 자세한 내용은 인스턴스 시작 마법사를 사용하여 인스턴스 시작 (p. 431) 단원을 참조하십시오.
9. 실행 중인 새 인스턴스에는 이전 단계에서 적용한 모든 사용자 지정이 포함되어 있습니다.

명령줄을 사용하여 인스턴스에서 AMI를 생성하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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 콘솔 또는 명령줄을 사용하여 이 스냅샷에서 AMI를 생성할 수 있습니다.

콘솔을 사용하여 스냅샷에서 AMI를 생성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창의 Elastic Block Store에서 스냅샷을 선택합니다.
3. 스냅샷을 선택하고 작업, 이미지 생성을 선택합니다.
4. EBS 스냅샷에서 이미지 생성 대화 상자에서 AMI 생성에 필요한 필드 정보를 모두 입력하고 생성을 선택합니다. 상위 인스턴스를 다시 생성하는 경우 상위 인스턴스와 동일한 옵션을 선택합니다.

   - 아키텍처: 32비트의 경우 i386을 선택하고 64비트의 경우 x86_64를 선택합니다.
   - 루트 디바이스 이름: 루트 볼륨에 적절한 이름을 입력합니다. 자세한 내용은 Linux 인스턴스의 디바이스 명명 (p. 1047) 단원을 참조하십시오.
   - 가상화 유형: 이 AMI에서 실행된 인스턴스가 반가상화(PV)를 사용하는지 또는 하드웨어 가상 머신(HVM) 가상화를 사용하는지 선택합니다. 자세한 내용은 Linux AMI 가상화 유형 (p. 95) 단원을 참조하십시오.
   - (PV 가상화 유형에만 해당) 커널 ID 및 RAM 디스크 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) 블록 디바이스 매핑: 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. 965) 단원을 참조하십시오.

명령줄을 사용하여 스냅샷에서 AMI를 생성하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- register-image(AWS CLI)
- Register-EC2Image(Windows PowerShell용 AWS 도구)

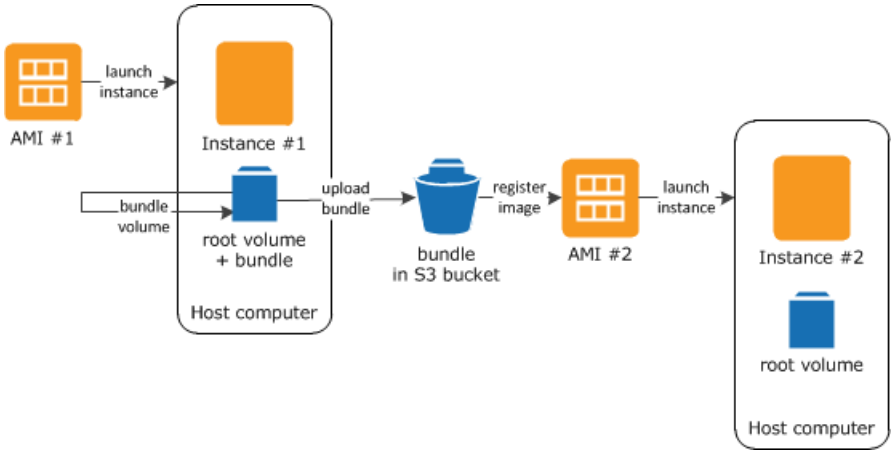
인스턴스 스토어 기반 Linux AMI 생성

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.

이 AMI의 생성 과정은 Amazon EBS 지원 AMI와 다릅니다. Amazon EBS 지원 인스턴스와 인스턴스 스토어 지원 인스턴스 간의 차이점에 대한 자세한 정보와 인스턴스의 루트 디바이스 유형을 확인하는 방법은 루트 디바이스 스토리지 (p. 93) 단원을 참조하십시오. Amazon EBS 기반 Linux AMI를 만들어야 하는 경우 Amazon EBS 지원 Linux AMI 생성 (p. 112) 단원을 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 Linux AMI 등록 취소 (p. 158) 단원을 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. 1049) 단원을 참조하십시오.

사전 조건

AMI를 만들려면 먼저 다음 작업을 완료해야 합니다.

- AMI 도구를 설치합니다. 자세한 내용은 AMI 도구 설치 (p. 116) 단원을 참조하십시오.
- AWS CLI를 설치합니다. 자세한 내용은 AWS Command Line Interface 설정 시작하기를 참조하십시오.
- 번들용 Amazon S3 버킷이 있는지 확인합니다. Amazon S3 버킷을 만들려면 Amazon S3 콘솔을 열고 버킷 생성을 클릭합니다. 또는 AWS CLI mb 명령을 사용할 수도 있습니다.
- AWS 계정 ID가 있어야 합니다. 자세한 내용은 AWS General Reference에서 AWS 계정 식별자 단원을 참조하십시오.
- 액세스 키 ID와 보안 액세스 키가 있어야 합니다. 자세한 내용은 AWS General Reference에서 액세스 키 단원을 참조하십시오.
- X.509 인증서와 그에 따른 프라이빗 키가 있어야 합니다.
  - X.509 인증서를 만들어야 할 경우 서명 인증서 관리 (p. 118) 단원을 참조하십시오. X.509 인증서 및 프라이빗 키는 AMI를 암호화하고 해독하는 데 사용됩니다.
  - [중국(베이징)] `$EC2_AMITOOL_HOME/etc/ec2/amitools/cert-ec2-cn-north-1.pem` 인증서를 사용합니다.
  - [AWS GovCloud (US-West)] `$EC2_AMITOOL_HOME/etc/ec2/amitools/cert-ec2-gov.pem` 인증서를 사용합니다.
- 인스턴스에 연결하여 인스턴스를 사용자 지정합니다. 예를 들어, 소프트웨어 및 애플리케이션을 설치하고, 데이터를 복사하고, 임시 파일을 삭제하고, Linux 구성을 수정할 수 있습니다.

작업

- AMI 도구 설치 (p. 116)
- 인스턴스 스토어 지원 Amazon Linux 인스턴스에서 AMI 생성 (p. 119)
- 인스턴스 스토어 지원 Ubuntu 인스턴스에서 AMI 생성 (p. 122)
- 인스턴스 스토어 기반 AMI를 Amazon EBS 기반 AMI로 변환 (p. 126)

AMI 도구 설치

AMI 도구를 사용하여 인스턴스 스토어 지원 Linux AMIs를 생성하고 관리할 수 있습니다. 도구를 사용하려면 Linux 인스턴스에 이 도구를 설치해야 합니다. AMI 도구는 RPM으로도 설치 가능하고 RPM을 지원하지 않는 Linux 배포판의 경우 .zip 파일로도 설치 가능합니다.

RPM을 사용하여 AMI 도구를 설치하려면

1. yum과 같은 Linux 배포용 패키지 관리자를 사용하여 Ruby를 설치합니다. 다음 예를 참조하십시오.

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

2. wget 또는 curl과 같은 도구를 사용하여 RPM 파일을 다운로드합니다. 다음 예를 참조하십시오.

```
[ec2-user ~]$ wget https://s3.amazonaws.com/ec2-downloads/ec2-ami-tools.noarch.rpm
```

3. 다음 명령을 사용하여 RPM 파일의 서명이 활성화되었는지 확인합니다.

```
[ec2-user ~]$ rpm -K ec2-ami-tools.noarch.rpm
```

   위의 명령은 파일의 SHA1 및 MD5 해시가 `OK.`임을 나타냅니다. 명령에서 해시가 `NOT OK`임을 나타내면 다음 명령을 사용하여 파일의 헤더 SHA1 및 MD5 해시를 표시합니다.

```
[ec2-user ~]$ rpm -Kv ec2-ami-tools.noarch.rpm
```

그런 다음 파일의 헤더 SHA1 및 MD5 해시를 다음 확인된 AMI 도구 해시와 비교하여 파일의 진위 여부를 확인합니다.

- 헤더 SHA1: a1f662d6f25f69871104e6a62187fa4df508f880
- MD5: 9faff05258064e2f7909b66142de6782

파일의 헤더 SHA1 및 MD5 해시가 확인된 AMI 도구 해시와 일치하면 다음 단계를 계속 진행합니다.

4. 다음 명령을 사용하여 RPM을 설치합니다:

```
[ec2-user ~]$ sudo yum install ec2-ami-tools.noarch.rpm
```

5. ec2-ami-tools-version (p. 129) 명령을 사용하여 AMI 도구 설치를 확인합니다.

```
[ec2-user ~]$ 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.

6. (선택 사항) 이전 단계에서 오류가 발생하면 AMI 도구 설치 위치를 `RUBYLIB` 경로에 추가합니다.

   a. 다음 명령을 실행하여 추가할 경로를 확인합니다.

```
[ec2-user ~]$ 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 오류에서 없다고 표시된 파일이 `/usr/lib/ruby/site_ruby` 및 `/usr/lib64/ruby/site_ruby`에 위치하고 있습니다.

   b. 이전 단계의 위치를 `RUBYLIB` 경로에 추가합니다.

```
[ec2-user ~]$ export RUBYLIB=$RUBYLIB:/usr/lib/ruby/site_ruby:/usr/lib64/ruby/
site_ruby
```

   c. ec2-ami-tools-version (p. 129) 명령을 사용하여 AMI 도구 설치를 확인합니다.

```
[ec2-user ~]$ ec2-ami-tools-version
```

.zip 파일을 사용하여 AMI 도구를 설치하려면

1. apt-get과 같은 Linux 배포용 패키지 관리자를 사용하여 Ruby를 설치하고 압축을 풉니다. 다음 예를 참조하십시오.

```
[ec2-user ~]$ sudo apt-get update -y && sudo apt-get install -y ruby unzip
```

2. wget 또는 curl과 같은 도구를 사용하여 .zip 파일을 다운로드합니다. 다음 예를 참조하십시오.

```
[ec2-user ~]$ wget https://s3.amazonaws.com/ec2-downloads/ec2-ami-tools.zip
```

3. 파일의 압축을 적합한 설치 디렉터리(예: `/usr/local/ec2`)에 풉니다.

```
[ec2-user ~]$ 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 있습니다. *x.x.x*는 도구의 버전 번호입니다(예: `ec2-ami-tools-1.5.7`).

4. `EC2_AMITOOL_HOME` 환경 변수를 도구의 설치 디렉터리로 설정합니다. 다음 예를 참조하십시오.

```
[ec2-user ~]$ export EC2_AMITOOL_HOME=/usr/local/ec2/ec2-ami-tools-x.x.x
```

5. 도구를 `PATH` 환경 변수에 추가합니다. 다음 예를 참조하십시오.

```
[ec2-user ~]$ export PATH=$EC2_AMITOOL_HOME/bin:$PATH
```

6. ec2-ami-tools-version (p. 129) 명령을 사용하여 AMI 도구 설치를 확인할 수 있습니다.

```
[ec2-user ~]$ ec2-ami-tools-version
```

서명 인증서 관리

AMI 도구의 특정 명령에는 X.509 인증서라고도 하는 서명 인증서가 필요합니다. 인증서를 생성한 다음 AWS에 업로드해야 합니다. 예를 들어 OpenSSL과 같은 타사 도구를 사용하여 인증서를 생성할 수 있습니다.

서명 인증서를 만들려면

1. OpenSSL을 설치 및 구성합니다.
2. `openssl genrsa` 명령을 사용하여 프라이빗 키를 생성하고 `.pem` 파일에 출력을 저장합니다. 2048 또는 4096비트 RSA 키를 생성하는 것이 좋습니다.

```
openssl genrsa 2048 > private-key.pem
```

3. `openssl req` 명령을 사용하여 인증서를 만듭니다.

```
openssl req -new -x509 -nodes -sha256 -days 365 -key private-key.pem -outform PEM -
out certificate.pem
```

AWS에 인증서를 업로드하려면 upload-signing-certificate 명령을 사용합니다.

```
aws iam upload-signing-certificate --user-name user-name --certificate-body file://path/to/
certificate.pem
```

사용자에게 적용되는 인증서를 나열하려면 list-signing-certificates 명령을 사용하십시오.

```
aws iam list-signing-certificates --user-name user-name
```

사용자의 서명 인증서를 비활성화하거나 다시 활성화하려면 update-signing-certificate 명령을 사용하십시오. 다음 명령으로 인증서를 비활성화할 수 있습니다.

```
aws iam update-signing-certificate --certificate-id OFHPLP4ZULTHYPMSYEX7O4BEXAMPLE --
status Inactive --user-name user-name
```

인증서를 삭제하려면 delete-signing-certificate 명령을 사용하십시오.

```
aws iam delete-signing-certificate --user-name user-name --certificate-
id OFHPLP4ZULTHYPMSYEX7O4BEXAMPLE
```

인스턴스 스토어 지원 인스턴스에서 AMI 생성

다음은 인스턴스 스토어 기반 인스턴스에서 인스턴스 스토어 기반 AMI를 만드는 절차입니다. 시작하기 전에 먼저 사전 조건 (p. 116)을 읽으십시오.

주제

- 인스턴스 스토어 지원 Amazon Linux 인스턴스에서 AMI 생성 (p. 119)
- 인스턴스 스토어 지원 Ubuntu 인스턴스에서 AMI 생성 (p. 122)

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. 122) 단원을 참조하십시오.

AMI 도구 사용을 준비하려면(HVM 인스턴스에만 해당)

1. 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 지원 Amazon Linux 인스턴스에서 AMI를 생성하려면

이 절차에서는 사전 조건 (p. 116)의 사전 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. 488) 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. 다음 예를 참조하십시오.

   ```
   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 내에서 ec2-bundle-vol (p. 132) 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. 값에는 `*.sw`, `*.swo`, `*.swp`, `*.pem`, `*.priv`, `*id_rsa*`, `*id_dsa*` `*.gpg`, `*.jks`, `*/.ssh/authorized_keys` 및 `*/.bash_history`가 포함됩니다. 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 탑재된 인스턴스도 있으며, 새 Amazon EBS 볼륨을 생성 (p. 908), 연결 (p. 910) 및 탑재 (p. 914)하여 번들을 저장할 수도 있습니다.

   a. ec2-bundle-vol 명령을 루트로 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 실행하려면 다음과 같이 ec2-bundle-vol (p. 132) 명령을 실행합니다.

   ```
   [root ec2-user]# ec2-bundle-vol -k /tmp/cert/pk-
   HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -c /tmp/cert/cert-
   HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -u 123456789012 -r x86_64 -e /tmp/cert --
   partition gpt
   ```

   Note

   중국(베이징) 및 AWS GovCloud (US-West) 리전에는 `--ec2cert` 파라미터를 사용하고 사전 조건 (p. 116)에 따라 인증서를 지정하십시오.

   이미지가 생성되는 데 몇 분 정도 걸릴 수 있습니다. 이 명령이 완료되면 `/tmp`(또는 기본값이 아닌) 디렉터리에 번들(`image.manifest.xml`과 여러 `image.part.`*`xx`* 파일)이 포함됩니다.

   c. 루트 셸을 종료합니다.

   ```
   [root ec2-user]# exit
   ```

3. (선택 사항) 인스턴스 스토어 볼륨을 더 추가하려면 AMI의 `image.manifest.xml` 파일에서 블록 디바이스 매핑을 편집합니다. 자세한 내용은 블록 디바이스 매핑 (p. 1049) 단원을 참조하십시오.

   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.xml`에서 블록 디바이스 매핑을 편집합니다. 아래 예는 `ephemeral1` 인스턴스 스토어 볼륨의 새 항목을 보여 줍니다.

Note

제외 파일 목록은 ec2-bundle-vol (p. 132) 단원을 참조하십시오.

```
<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. Amazon S3에 번들을 업로드하려면 다음과 같이 ec2-upload-bundle (p. 142) 명령을 실행합니다.

```
[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
```

Important

미국 동부(버지니아 북부) 이외 리전에서 AMI를 등록하려면 `--region` 옵션이 있는 대상 리전과 대상 리전에 이미 존재하는 버킷 경로 또는 대상 리전에 생성할 수 있는 고유 버킷 경로를 모두 지정해야 합니다.

5. (선택 사항) 번들을 Amazon S3에 업로드한 후에는 다음 rm 명령을 사용하여 인스턴스의 `/tmp` 디렉터리에서 번들을 제거할 수 있습니다.

```
[ec2-user ~]$ sudo rm /tmp/image.manifest.xml /tmp/image.part.* /tmp/image
```

Important

`-d /path/to/bundle/storage`에서 Step 2 (p. 120) 옵션과 함께 경로를 지정한 경우 `/tmp` 대신 해당 경로를 사용합니다.

6. AMI를 등록하려면 다음과 같이 register-image 명령을 사용합니다.

```
[ec2-user ~]$ aws ec2 register-image --image-location my-s3-
bucket/bundle_folder/bundle_name/image.manifest.xml --name AMI_name --virtualization-
type hvm
```

Important

이전에 ec2-upload-bundle (p. 142) 명령에 리전을 지정한 경우 이 명령에도 해당 리전을 다시 지정하십시오.

인스턴스 스토어 지원 Ubuntu 인스턴스에서 AMI 생성

이 단원에서는 인스턴스 스토어 볼륨을 루트 볼륨으로 사용하여 Ubuntu Linux 인스턴스에서 AMI를 생성하는 방법에 대해 설명합니다. 다음 절차는 다른 Linux 배포를 실행하는 인스턴스에서는 작동하지 않을 수 있습니다. Amazon Linux 관련 절차는 인스턴스 스토어 지원 Amazon Linux 인스턴스에서 AMI 생성 (p. 119) 단원을 참조하십시오.

AMI 도구 사용을 준비하려면(HVM 인스턴스에만 해당)

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. 122) 항목으로 이동합니다. GRUB Legacy가 이미 존재하는 경우 Step 2 (p. 122)으로 건너뛸 수 있습니다.

   b. 다음 명령을 사용하여 `grub` 패키지를 설치합니다.

   ```
   ubuntu:~$ sudo apt-get install -y grub
   ```

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` 뒤에 이어지는 옵션을 메모해 둡니다. 옵션이 이와 다를 수도 있습니다.

4. `/boot/grub/menu.lst`의 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 ttyS0을 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 파일을 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/efi에 탑재된 별도의 EFI 파티션과 GPT 파티션 테이블이 인스턴스 스토어 기반 Ubuntu AMI에 사용됩니다. ec2-bundle-vol 명령은 이 부팅 파티션을 번들링할 수 없으므로, 아래 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 Ubuntu 인스턴스에서 AMI를 생성하려면

이 절차에서는 사전 조건 (p. 116)의 사전 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. 488) 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. 다음 예를 참조하십시오.

```
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. 인스턴스에서 ec2-bundle-vol (p. 132) 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. 값에는 `*.sw`, `*.swo`, `*.swp`, `*.pem`, `*.priv`, `*id_rsa*`, `*id_dsa*` `*.gpg`, `*.jks`, `*/.ssh/authorized_keys` 및 `*/.bash_history`가 포함됩니다. 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 탑재된 인스턴스도 있으며, 새 Amazon EBS 볼륨을 생성 (p. 908), 연결 (p. 910) 및 탑재 (p. 914)하여 번들을 저장할 수도 있습니다.

   a. ec2-bundle-vol 명령을 루트로 실행해야 합니다. 대부분의 명령에 대해 sudo를 사용하여 승격된 권한을 얻을 수 있지만 이 경우 환경 변수를 유지하려면 sudo -E su를 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 실행하려면 다음과 같이 ec2-bundle-vol (p. 132) 명령을 실행합니다.

   ```
   root@ubuntu:# 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
   ```

   Important

   Ubuntu 14.04 이상 HVM 인스턴스의 경우 부팅 명령을 제대로 번들링하려면 `--partition mbr` 플래그를 추가합니다. 그렇지 않으면 새로 생성된 AMI가 부팅되지 않습니다.

   이미지가 생성되는 데 몇 분 정도 걸릴 수 있습니다. 이 명령이 완료되면 `tmp` 디렉터리에 번들(`image.manifest.xml`과 여러 `image.part.xx` 파일)이 포함됩니다.

   c. 루트 셸을 종료합니다.

```
root@ubuntu:# exit
```

3. (선택 사항) 인스턴스 스토어 볼륨을 더 추가하려면 AMI의 `image.manifest.xml` 파일에서 블록 디바이스 매핑을 편집합니다. 자세한 내용은 블록 디바이스 매핑 (p. 1049) 단원을 참조하십시오.

   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. Amazon S3에 번들을 업로드하려면 다음과 같이 ec2-upload-bundle (p. 142) 명령을 실행합니다.

   ```
   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
   ```

   Important

   미국 동부(버지니아 북부) 이외 리전에서 AMI를 등록하려면 `--region` 옵션이 있는 대상 리전과 대상 리전에 이미 존재하는 버킷 경로 또는 대상 리전에 생성할 수 있는 고유 버킷 경로를 모두 지정해야 합니다.

5. (선택 사항) 번들을 Amazon S3에 업로드한 후에는 다음 rm 명령을 사용하여 인스턴스의 `/tmp` 디렉터리에서 번들을 제거할 수 있습니다.

   ```
   ubuntu:~$ sudo rm /tmp/image.manifest.xml /tmp/image.part.* /tmp/image
   ```

   Important

   `-d` *`/path/to/bundle/storage`*에서 Step 2 (p. 124) 옵션과 함께 경로를 지정한 경우 `/tmp` 대신 아래 동일 경로를 사용합니다.

6. AMI를 등록하려면 다음과 같이 register-image AWS CLI 명령을 사용합니다.

```
ubuntu:~$ aws ec2 register-image --image-location my-s3-
bucket/bundle_folder/bundle_name/image.manifest.xml --name AMI_name --virtualization-
type hvm
```

> **Important**
>
> 이전에 ec2-upload-bundle (p. 142) 명령에 리전을 지정한 경우 이 명령에도 해당 리전을 다시 지정하십시오.

7. [Ubuntu 14.04 이상에 해당] `/etc/fstab`에서 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

1. Amazon EBS 기반 AMI에서 Amazon Linux 인스턴스를 시작합니다. 자세한 내용은 인스턴스 시작 마법사를 사용하여 인스턴스 시작 (p. 431) 단원을 참조하십시오. Amazon Linux 인스턴스에는 AWS CLI와 AMI 도구가 이미 설치되어 있습니다.
2. 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.

   a. 다음과 같이 인스턴스에서 X.509 프라이빗 키에 대한 임시 디렉터리를 생성합니다.

   ```
   [ec2-user ~]$ mkdir /tmp/cert
   ```

   b. scp (p. 488) 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. 다음 예를 참조하십시오.

   ```
   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 메모해 둡니다.

   > **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. `/tmp/bundle` using the ec2-download-bundle (p. 137) 명령을 사용하여 인스턴스 스토어 기반 AMI 용 번들을 /tmp/bundle로 다운로드합니다.

```
[ec2-user ~]$ ec2-download-bundle -b my-s3-bucket/bundle_folder/bundle_name -m
 image.manifest.xml -a $AWS_ACCESS_KEY_ID -s $AWS_SECRET_ACCESS_KEY --privatekey /path/
to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -d /tmp/bundle
```

7. ec2-unbundle (p. 141) 명령을 사용하여 번들에서 이미지 파일을 다시 구성합니다.

a. 디렉터리를 번들 폴더로 변경합니다.

```
[ec2-user ~]$ cd /tmp/bundle/
```

b. ec2-unbundle (p. 141) 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
```

9. 번들링되지 않은 새 파티션용 볼륨을 검색합니다.

```
[ec2-user bundle]$ sudo partprobe /dev/sdb1
```

10. 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/sdb1`이지만, 디바이스 이름이 이와 다를 수 있습니다. 볼륨이 파티셔닝되지 않은 경우 마운트할 디바이스는 `/dev/sdb`(디바이스 파티션 끝 숫자가 없음)와 비슷할 것입니다.

11. 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
```

12. 주로 사용하는 텍스트 편집기(예: vim 또는 nano)를 사용하여 EBS 볼륨의 `/etc/fstab` 파일을 열고 인스턴스 스토어(휘발성) 볼륨에 대한 항목을 모두 제거합니다. Amazon EBS 볼륨이 `/mnt/ebs`에 마운트되므로 `fstab` 파일은 `/mnt/ebs/etc/fstab`에 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.

13. 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
```

14. 다음과 같이 새 Amazon EBS 볼륨에서 AMI를 생성합니다.

    a. 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
    ```

    b. 스냅샷이 완전한지 확인합니다.

    ```
    [ec2-user bundle]$ aws ec2 describe-snapshots --region us-west-2 --snapshot-
    id snapshot_id
    ```

    c. describe-imagesdescribe-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 메모해 둡니다.

    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 DeviceName=device-
    name,Ebs={SnapshotId=snapshot_id} --virtualization-type paravirtual --architecture
     x86_64 --kernel-id aki-fc8f11cc --root-device-name device-name
    ```

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
```

AMI 도구 참조

AMI 도구 명령을 사용하여 인스턴스 스토어 지원 Linux AMI를 생성하고 관리할 수 있습니다. 도구를 설정하려면 AMI 도구 설치 (p. 116)을 참조하십시오.

액세스 키에 대한 자세한 정보는 AWS 액세스 키 관리 모범 사례를 참조하십시오.

명령

- ec2-ami-tools-version (p. 129)
- ec2-bundle-image (p. 129)
- ec2-bundle-vol (p. 132)
- ec2-delete-bundle (p. 135)
- ec2-download-bundle (p. 137)
- ec2-migrate-manifest (p. 139)
- ec2-unbundle (p. 141)
- ec2-upload-bundle (p. 142)
- AMI 도구의 일반 옵션 (p. 145)

ec2-ami-tools-version

설명

AMI 도구 버전을 설명합니다.

구문

```
ec2-ami-tools-version
```

결과

버전 정보입니다.

예

이 예제 명령은 사용 중인 AMI 도구의 버전 정보를 표시합니다.

```
[ec2-user ~]$ 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

`-c, --cert` path

사용자의 PEM 인코딩된 RSA 퍼블릭 키 인증서 파일입니다.

필수 항목 여부: 예

`-k, --privatekey` path

PEM 인코딩된 RSA 키 파일의 경로입니다. 이 번들이 번들링되지 않아 안전한 장소에 보관되도록 이 키를 지정해야 합니다. 키를 AWS 계정에 등록할 필요는 없습니다.

필수 항목 여부: 예

`-u, --user` account

사용자의 AWS 계정 ID(대시 없이)입니다.

필수 항목 여부: 예

`-i, --image` path

번들링할 이미지에 대한 경로입니다.

필수 항목 여부: 예

`-d, --destination` path

번들을 생성할 디렉터리입니다.

기본값: `/tmp`

필수 항목 여부: 아니요

`--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 경로는 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. 116)의 RPM 또는 ZIP 파일에서 AMI 도구를 설치한 경우 `$EC2_AMITOOL_HOME/etc/ec2/amitools/`에 인증서가 있습니다.

필수: `us-gov-west-1` 및 `cn-north-1` 리전에만 해당됩니다.

`-r, --arch` 보기

이미지 아키텍처. 명령줄에 아키텍처를 제공하지 않으면 번들링 시작 시 해당 메시지가 표시됩니다.

유효한 값: `i386` | `x86_64`

필수 항목 여부: 아니요

`--productcodes` code1,code2,...

등록 시 이미지에 연결할 제품 코드로, 쉼표로 구분됩니다.

필수 항목 여부: 아니요

`-B, --block-device-mapping` 매핑

이 AMI의 인스턴스에 블록 디바이스를 표시할 방법을 정의합니다(인스턴스 유형이 지정된 디바이스를 지원하는 경우).

쉼표로 구분된 키-값 페어 목록을 지정합니다. 여기서 각 키는 가상 이름이며 각 값은 해당 디바이스 이름입니다. 가상 이름에는 다음이 포함됩니다.

- `ami` - 인스턴스별로 표시되는 루트 파일 시스템 디바이스
- `root` - 커널별로 표시되는 루트 파일 시스템 디바이스
- `swap` - 인스턴스별로 표시되는 교체 디바이스
- `ephemeralN` - N번째 인스턴스 스토어 볼륨

필수 항목 여부: 아니요

`-p, --prefix` prefix

번들링된 AMI 파일의 파일 이름 접두사입니다.

기본값: 이미지 파일의 이름입니다. 예를 들어, 이미지 경로가 `/var/spool/my-image/version-2/debian.img`이면, 기본 접두사는 `debian.img`입니다.

필수 항목 여부: 아니요

`--kernel` kernel_id

사용되지 않음. register-image를 사용하여 커널을 설정합니다.

필수 항목 여부: 아니요

`--ramdisk` ramdisk_id

사용되지 않음. register-image를 사용하여 RAM 디스크를 설정합니다(필요한 경우).

필수 항목 여부: 아니요

결과

번들링 프로세스의 단계 및 상태를 설명하는 상태 메시지입니다.

예

이 예제에서는 루프백 파일에서 생성한 운영 체제 이미지로부터 번들링된 AMI를 생성합니다.

```
[ec2-user ~]$ 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. 값에는 `*.sw`, `*.swo`, `*.swp`, `*.pem`, `*.priv`, `*id_rsa*`, `*id_dsa*` `*.gpg`, `*.jks`, `*/.ssh/authorized_keys` 및 `*/.bash_history`가 포함됩니다. 이러한 파일을 모든 포함하려면 `--no-filter` 옵션을 사용합니다. 이러한 파일 중 일부만 포함하려면 `--include` 옵션을 사용합니다.

자세한 내용은 인스턴스 스토어 기반 Linux AMI 생성 (p. 115) 단원을 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

`-c, --cert` path

사용자의 PEM 인코딩된 RSA 퍼블릭 키 인증서 파일입니다.

필수 항목 여부: 예

`-k, --privatekey` path

사용자의 PEM 인코딩된 RSA 키 파일의 경로입니다.

필수 항목 여부: 예

`-u, --user` account

사용자의 AWS 계정 ID(대시 없이)입니다.

필수 항목 여부: 예

`-d, --destination` destination

번들을 생성할 디렉터리입니다.

기본값: `/tmp`

필수 항목 여부: 아니요

`--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 경로는 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. 116)의 RPM 또는 ZIP 파일에서 AMI 도구를 설치한 경우 `$EC2_AMITOOL_HOME/etc/ec2/amitools/`에 인증서가 있습니다.

필수: `us-gov-west-1` 및 `cn-north-1` 리전에만 해당됩니다.

`-r, --arch` 보기

이미지 아키텍처입니다. 명령줄에 이를 제공하지 않으면 번들링 시작 시 이를 제공하라는 메시지가 표시됩니다.

유효한 값: `i386` | `x86_64`

필수 항목 여부: 아니요

`--productcodes` code1,code2,...

등록 시 이미지에 연결할 제품 코드로, 쉼표로 구분됩니다.

필수 항목 여부: 아니요

`-B, --block-device-mapping` 매핑

이 AMI의 인스턴스에 블록 디바이스를 표시할 방법을 정의합니다(인스턴스 유형이 지정된 디바이스를 지원하는 경우).

쉼표로 구분된 키-값 페어 목록을 지정합니다. 여기서 각 키는 가상 이름이며 각 값은 해당 디바이스 이름입니다. 가상 이름에는 다음이 포함됩니다.

- `ami` - 인스턴스별로 표시되는 루트 파일 시스템 디바이스
- `root` - 커널별로 표시되는 루트 파일 시스템 디바이스
- `swap` - 인스턴스별로 표시되는 교체 디바이스
- `ephemeralN` - N번째 인스턴스 스토어 볼륨

필수 항목 여부: 아니요

`-a, --all`

원격으로 마운트된 파일 시스템의 디렉터리를 포함하여 모든 디렉터리를 번들링합니다.

필수 항목 여부: 아니요

`-e, --exclude` directory1,directory2,...

번들 작업에서 제외할 절대 디렉터리 경로 및 파일 목록입니다. 이 파라미터는 `--all` 옵션보다 우선합니다. exclude가 지정되면 파라미터와 함께 나열된 디렉터리 및 하위 디렉터리는 볼륨에 번들링되지 않습니다.

필수 항목 여부: 아니요

`-i, --include` file1,file2,...

번들 작업에 포함할 파일 목록입니다. 지정된 파일은 중요한 정보를 포함할 수 있으므로 AMI에서 제외되지 않습니다.

필수 항목 여부: 아니요

`--no-filter`

지정한 경우 지정한 파일이 중요한 정보를 포함할 수 있으므로 AMI에서 파일을 제외하지 않습니다.

필수 항목 여부: 아니요

`-p, --prefix` prefix

번들링된 AMI 파일의 파일 이름 접두사입니다.

기본값: `image`

필수 항목 여부: 아니요

`-s, --size` size

생성할 이미지 파일의 크기(MB, 1024 * 1024바이트)입니다. 최대 크기는 10240MB입니다.

기본값: 10240

필수 항목 여부: 아니요

`--[no-]inherit`

이미지가 인스턴스의 메타데이터를 상속해야 하는지 여부를 나타냅니다(기본값은 상속). `--inherit`를 활성화했지만 인스턴스 메타데이터에 액세스할 수 없으면 번들링이 실패합니다.

필수 항목 여부: 아니요

`-v, --volume` 볼륨

번들을 생성해 올 마운트된 볼륨의 절대 경로입니다.

기본값: 루트디렉터리입니다(/).

필수 항목 여부: 아니요

`-P, --partition` 유형

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 `gpt`가 기본값이 됩니다.

유효한 값: `mbr` | `gpt` | `none`

필수 항목 여부: 아니요

`-S, --script` script

번들링 작업 직전에 실행할 사용자 정의 스크립트입니다. 스크립트에서 하나의 인수(볼륨의 마운트 지점)를 예상해야 합니다.

필수 항목 여부: 아니요

`--fstab` path

이미지에 번들링할 fstab 경로입니다. 지정하지 않으면 Amazon EC2가 /etc/fstab을 번들링합니다.

필수 항목 여부: 아니요

`--generate-fstab`

Amazon EC2-제공 fstab을 사용하여 볼륨을 번들링합니다.

필수 항목 여부: 아니요

`--grub-config`

이미지에 번들링할 대체 grub 구성 파일 경로입니다. 기본적으로 `ec2-bundle-vol`은 복제된 이미지에 `/boot/grub/menu.lst` 또는 `/boot/grub/grub.conf`가 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

`--kernel` kernel_id

사용되지 않음. register-image를 사용하여 커널을 설정합니다.

필수 항목 여부: 아니요

`--ramdisk`ramdisk_id

사용되지 않음. register-image를 사용하여 RAM 디스크를 설정합니다(필요한 경우).

필수 항목 여부: 아니요

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-user ~]$ 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

`-b, --bucket` bucket

번들링된 AMI를 포함하는 Amazon S3 버킷 이름으로, '/'-delimited 경로 접두사가 붙기도 합니다(옵션).

필수 항목 여부: 예

`-a, --access-key` access_key_id

AWS 액세스 키 ID입니다.

필수 항목 여부: 예

`-s, --secret-key` secret_access_key

AWS 보안 액세스 키입니다.

필수 항목 여부: 예

`-t, --delegation-token` token

AWS 요청에 함께 전달되는 위임 토큰입니다. 자세한 내용은 Using Temporary Security Credentials를 참조하십시오.

필수 항목 여부: 임시 보안 자격 증명을 사용하는 경우에만.

기본값: `AWS_DELEGATION_TOKEN` 환경 변수 값(설정된 경우).

`--region`region

요청 서명에서 사용하는 리전입니다.

기본값: `us-east-1`

필수: 서명 버전 4를 사용하는 경우 필수입니다.

`--sigv`version

요청 서명 시 사용하는 서명 버전입니다.

유효한 값: `2 | 4`

기본값: `4`

필수 항목 여부: 아니요

`-m, --manifest`path

매니페스트 파일 경로입니다.

필수: `--prefix` 또는 `--manifest`를 지정해야 합니다.

`-p, --prefix` prefix

번들링된 AMI 파일 이름 접두사입니다. 전체 접두사를 제공합니다. 예를 들어 접두사가 image.img인 경우, `-p image.img`를 사용해야 합니다(`-p image` 아님).

필수: `--prefix` 또는 `--manifest`를 지정해야 합니다.

`--clear`

지정된 번들을 삭제한 후 비어 있으면 Amazon S3 버킷을 삭제합니다.

필수 항목 여부: 아니요

`--retry`

모든 Amazon S3 오류에서 작업당 최대 5회 자동으로 재시도합니다.

필수 항목 여부: 아니요

`-y, --yes`

모든 질문 메시지에 대한 답은 자동적으로 예라고 가정합니다.

필수 항목 여부: 아니요

결과

Amazon EC2에서 삭제 프로세스의 단계 및 상황을 나타내는 상태 메시지를 표시합니다.

예

이 예제에서는 Amazon S3에서 번들을 삭제합니다.

```
[ec2-user ~]$ ec2-delete-bundle -b aws-s3-bucket1 -a your_access_key_id -
s your_secret_access_key
Deleting files:
aws-s3-bucket1/
image.manifest.xml
aws-s3-bucket1/
image.part.00
aws-s3-bucket1/
image.part.01
aws-s3-bucket1/
image.part.02
aws-s3-bucket1/
image.part.03
aws-s3-bucket1/
image.part.04
aws-s3-bucket1/
image.part.05
aws-s3-bucket1/image.part.06
Continue? [y/n]
y
Deleted aws-s3-bucket1/image.manifest.xml
Deleted aws-s3-bucket1/image.part.00
Deleted aws-s3-bucket1/image.part.01
Deleted aws-s3-bucket1/image.part.02
Deleted aws-s3-bucket1/image.part.03
Deleted aws-s3-bucket1/image.part.04
Deleted aws-s3-bucket1/image.part.05
Deleted aws-s3-bucket1/image.part.06
ec2-delete-bundle complete.
```

ec2-download-bundle

설명

지정된 인스턴스 스토어 지원 Linux AMIs를 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

`-b, --bucket` bucket

번들이 있는 Amazon S3 버킷 이름으로, '/'-delimited 경로 접두사가 붙기도 합니다(옵션).

필수 항목 여부: 예

`-a, --access-key` access_key_id

AWS 액세스 키 ID입니다.

필수 항목 여부: 예

`-s, --secret-key` secret_access_key

AWS 보안 액세스 키입니다.

필수 항목 여부: 예

`-k, --privatekey` path

매니페스트를 해독하는 데 사용되는 프라이빗 키입니다.

필수 항목 여부: 예

`--url` url

Amazon S3 서비스 URL입니다.

기본값: `https://s3.amazonaws.com/`

필수 항목 여부: 아니요

`--region` region

요청 서명에서 사용하는 리전입니다.

기본값: `us-east-1`

필수: 서명 버전 4를 사용하는 경우 필수입니다.

`--sigv` version

요청 서명 시 사용하는 서명 버전입니다.

유효한 값: `2` | `4`

기본값: `4`

필수 항목 여부: 아니요

`-m, --manifest` file

매니페스트 파일 이름입니다(경로 제외). 매니페스트(`-m`) 또는 접두사(`-p`) 중 하나를 지정하는 것이 좋습니다.

필수 항목 여부: 아니요

`-p, --prefix` prefix

번들링된 AMI 파일의 파일 이름 접두사입니다.

기본값: `image`

필수 항목 여부: 아니요

`-d, --directory` directory

다운로드된 번들이 저장되는 디렉터리입니다. 존재하는 디렉터리여야 합니다.

기본값: 현재 작업 디렉터리입니다.

필수 항목 여부: 아니요

`--retry`

모든 Amazon S3 오류에서 작업당 최대 5회 자동으로 재시도합니다.

필수 항목 여부: 아니요

결과

다운로드 프로세스의 다양한 단계를 나타내는 상태 메시지가 표시됩니다.

예

이 예제는 `bundled` 디렉터리(Linux mkdir 명령 사용)를 생성하고 *`aws-s3-bucket1`* Amazon S3 버킷에서 번들을 다운로드합니다.

```
[ec2-user ~]$ mkdir bundled
[ec2-user ~]$ ec2-download-bundle -b aws-s3-bucket1/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 aws-s3-bucket1 to mybundle/
image.manifest.xml ...
Downloading part image.part.00 from aws-s3-bucket1/bundles/bundle_name to mybundle/
image.part.00 ...
Downloaded image.part.00 from aws-s3-bucket1
Downloading part image.part.01 from aws-s3-bucket1/bundles/bundle_name to mybundle/
image.part.01 ...
Downloaded image.part.01 from aws-s3-bucket1
Downloading part image.part.02 from aws-s3-bucket1/bundles/bundle_name to mybundle/
image.part.02 ...
Downloaded image.part.02 from aws-s3-bucket1
Downloading part image.part.03 from aws-s3-bucket1/bundles/bundle_name to mybundle/
image.part.03 ...
Downloaded image.part.03 from aws-s3-bucket1
Downloading part image.part.04 from aws-s3-bucket1/bundles/bundle_name to mybundle/
image.part.04 ...
Downloaded image.part.04 from aws-s3-bucket1
Downloading part image.part.05 from aws-s3-bucket1/bundles/bundle_name to mybundle/
image.part.05 ...
Downloaded image.part.05 from aws-s3-bucket1
Downloading part image.part.06 from aws-s3-bucket1/bundles/bundle_name to mybundle/
image.part.06 ...
Downloaded image.part.06 from aws-s3-bucket1
```

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

`-c, --cert` path

사용자의 PEM 인코딩된 RSA 퍼블릭 키 인증서 파일입니다.

필수 항목 여부: 예

`-k, --privatekey` path

사용자의 PEM 인코딩된 RSA 키 파일의 경로입니다.

필수 항목 여부: 예

`--manifest` path

매니페스트 파일 경로입니다.

필수 항목 여부: 예

`-a, --access-key` access_key_id

AWS 액세스 키 ID입니다.

필수: 자동 매핑 사용 시 필수입니다.

`-s, --secret-key` secret_access_key

AWS 보안 액세스 키입니다.

필수: 자동 매핑 사용 시 필수입니다.

`--region` region

매핑 파일에서 조회하는 리전입니다.

필수: 자동 매핑 사용 시 필수입니다.

`--no-mapping`

커널 및 RAM 디스크의 자동 매핑을 비활성화합니다.

마이그레이션 동안 Amazon EC2는 매니페스트 파일에 있는 커널 및 RAM 디스크를 대상 리전용으로 설계된 커널 및 RAM 디스크로 교체합니다. `--no-mapping` 파라미터를 지정하지 않으면, `ec2-migrate-bundle`에서 `DescribeRegions` 및 `DescribeImages` 작업을 사용하여 자동화된 매핑을 수행합니다.

필수: 자동 매핑에 사용되는 `-a`, `-s` 및 `--region` 옵션을 제공하지 않는 경우 필수입니다.

`--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 경로는 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. 116)의 ZIP 파일에서 AMI 도구를 설치한 경우, `$EC2_AMITOOL_HOME/etc/ec2/amitools/`에 인증서가 있습니다.

필수: `us-gov-west-1` 및 `cn-north-1` 리전에만 해당됩니다.

`--kernel` kernel_id

선택할 커널의 ID입니다.

> **Important**
>
> 커널 및 RAM 디스크 대신 PV-GRUB를 사용하는 것이 좋습니다. 자세한 내용은 자체 Linux 커널 활성화 (p. 171) 단원을 참조하십시오.

필수 항목 여부: 아니요

`--ramdisk` ramdisk_id

선택할 RAM 디스크의 ID입니다.

> **Important**
>
> 커널 및 RAM 디스크 대신 PV-GRUB를 사용하는 것이 좋습니다. 자세한 내용은 자체 Linux 커널 활성화 (p. 171) 단원을 참조하십시오.

필수 항목 여부: 아니요

결과

번들링 프로세스의 단계 및 상태를 설명하는 상태 메시지입니다.

예

이 예제는 `my-ami.manifest.xml` 매니페스트에 지정된 AMI;를 복사합니다.

```
[ec2-user ~]$ 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

`-k, --privatekey` path

PEM 인코딩된 RSA 키 파일의 경로입니다.

필수 항목 여부: 예

`-m, --manifest` path

매니페스트 파일 경로입니다.

필수 항목 여부: 예

`-s, --source` source_directory

번들이 포함된 디렉터리입니다.

기본값: 현재 디렉터리입니다.

필수 항목 여부: 아니요

`-d, --destination` destination_directory

AMI 번들을 해제해 넣을 디렉터리입니다. 대상 디렉터리가 있어야 합니다.

기본값: 현재 디렉터리입니다.

필수 항목 여부: 아니요

예

이 Linux 및 UNIX 예제는 `image.manifest.xml` 파일에 지정된 AMI 번들을 해제합니다.

```
[ec2-user ~]$ 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 기반 Linux AMI 생성 (p. 115) 단원을 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

`-b, --bucket` bucket

번들 이름을 저장할 Amazon S3 버킷 이름으로, 선택 항목인 '/'-delimited 경로 접두사가 붙기도 합니다. 버킷이 없을 경우에는 생성됩니다(해당 버킷 이름을 사용할 수 있는 경우).

필수 항목 여부: 예

`-a, --access-key` access_key_id

사용자의 AWS 액세스 키 ID입니다.

필수 항목 여부: 예

`-s, --secret-key` secret_access_key

AWS 보안 액세스 키입니다.

필수 항목 여부: 예

`-t, --delegation-token` token

AWS 요청에 함께 전달되는 위임 토큰입니다. 자세한 내용은 Using Temporary Security Credentials를 참조하십시오.

필수 항목 여부: 임시 보안 자격 증명을 사용하는 경우에만.

기본값: `AWS_DELEGATION_TOKEN` 환경 변수 값(설정된 경우).

`-m, --manifest` path

매니페스트 파일 경로입니다. 매니페스트 파일은 번들링 프로세스 중 생성되며 번들이 포함된 디렉터리에 있습니다.

필수 항목 여부: 예

`--url` url

사용되지 않음. 버킷이 `--region` 위치(`EU` 제외)로 제한되어 있지 않은 경우 대신 `eu-west-1` 옵션을 사용합니다. `--location` 플래그는 이러한 특정 위치 제한을 적용하는 유일한 방법입니다.

Amazon S3 엔드포인트 서비스 URL입니다.

기본값: `https://s3.amazonaws.com/`

필수 항목 여부: 아니요

`--region` region

대상 S3 버킷의 요청 서명에 사용할 리전입니다.

- 버킷이 없고 리전을 지정하지 않는 경우 이 도구는 (`us-east-1`에서) 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-1`을 리전으로 지정하는 경우 이 도구는 오류 메시지 없이 버킷의 실제 위치를 사용하며 일치하는 기존 파일을 덮어씁니다.
- 버킷이 있고 버킷의 실제 위치와 일치하지 않는 리전(`us-east-1` 이외)을 지정하는 경우 도구가 종료되고 오류가 발생합니다.

버킷이 `EU` 위치(`eu-west-1` 제외)로 제한된 경우 대신 `--location` 플래그를 사용합니다. `--location` 플래그는 이러한 특정 위치 제한을 적용하는 유일한 방법입니다.

기본값: `us-east-1`

필수: 서명 버전 4를 사용하는 경우 필수입니다.

`--sigv` version

요청 서명 시 사용하는 서명 버전입니다.

유효한 값: `2 | 4`

기본값: `4`

필수 항목 여부: 아니요

`--acl` acl

번들링된 이미지의 액세스 제어 목록 정책입니다.

유효한 값: `public-read` | `aws-exec-read`

기본값: `aws-exec-read`

필수 항목 여부: 아니요

`-d, --directory` directory

번들링된 AMI 파트가 포함된 디렉터리입니다.

기본값: 매니페스트 파일이 포함된 디렉터리(`-m` 옵션 참조)입니다.

필수 항목 여부: 아니요

`--part` part

지정된 파트와 모든 후속 파트의 업로드를 시작합니다. 예: `--part 04`.

필수 항목 여부: 아니요

`--retry`

모든 Amazon S3 오류에서 작업당 최대 5회 자동으로 재시도합니다.

필수 항목 여부: 아니요

`--skipmanifest`

매니페스트를 업로드하지 않습니다.

필수 항목 여부: 아니요

`--location` location

사용되지 않음. 버킷이 `--region` 위치(`EU` 제외)로 제한되어 있지 않은 경우 대신 `eu-west-1` 옵션을 사용합니다. `--location` 플래그는 이러한 특정 위치 제한을 적용하는 유일한 방법입니다.

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-1`에서) 위치 제한 없이 버킷을 만듭니다.

기본값: `--region`이 지정된 경우 위치는 여기에 지정된 리전으로 설정됩니다. `--region`이 지정되지 않은 경우 위치는 기본적으로 `us-east-1`로 설정됩니다.

필수 항목 여부: 아니요

결과

Amazon EC2에서 업로드 프로세스의 단계 및 상황을 나타내는 상태 메시지를 표시합니다.

예

이 예제에서는 `image.manifest.xml` 매니페스트에서 지정한 번들을 업로드합니다.

```
[ec2-user ~]$ ec2-upload-bundle -b aws-s3-bucket1/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 aws-s3-bucket1 ...
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

AMI 도구의 대부분은 다음과 같은 선택적 파라미터를 허용합니다.

`--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`

문제를 해결할 때 유용한 정보를 표시합니다.

EBS 지원 AMI를 통한 암호화 사용

Amazon EBS 스냅샷의 지원을 받는 AMI에서는 Amazon EBS 암호화를 활용할 수 있습니다. 데이터 볼륨과 루트 볼륨 모두의 스냅샷을 암호화하고 AMI에 연결할 수 있습니다. 전체 EBS 암호화 지원을 통해 인스턴스를 시작하고 이미지를 복사할 수 있습니다. 이러한 작업을 위한 암호화 파라미터는 AWS KMS를 사용할 수 있는 모든 리전에서 지원됩니다.

암호화된 EBS 볼륨이 있는 EC2 인스턴스는 다른 인스턴스와 동일한 방법으로 AMIs에서 시작됩니다. 또한 암호화되지 않은 EBS 스냅샷이 지원하는 AMI에서 인스턴스를 시작할 때 시작하는 동안 해당 볼륨의 일부 또는 전체를 암호화할 수 있습니다.

EBS 볼륨처럼 AMI의 스냅샷을 기본 AWS Key Management Service 고객 마스터 키(CMK) 또는 지정한 고객 관리형 키로 암호화할 수 있습니다. 어느 경우든 선택한 키에 대한 사용 권한이 있어야 합니다.

암호화된 스냅샷이 있는 AMI는 AWS 계정 간에 공유할 수 있습니다. 자세한 내용은 공유 AMI 단원을 참조하십시오.

인스턴스 시작 시나리오

Amazon EC2 인스턴스는 AWS Management 콘솔을(를) 사용하거나 Amazon EC2 API 또는 CLI를 직접 사용하여 블록 디바이스 매핑을 통해 파라미터가 입력된 `RunInstances` 작업을 사용하여 AMI에서 시작됩니

다. 블록 디바이스 매핑에 대한 자세한 내용은 블록 디바이스 매핑 단원을 참조하십시오. AWS CLI에서 블록 디바이스 매핑을 제어하는 예제는 시작, 목록 및 EC2 인스턴스 종료 단원을 참조하십시오.

기본적으로 명시적인 암호화 파라미터가 없는 경우 `RunInstances` 작업은 AMI 원본 스냅샷에서 EBS 볼륨을 복원하는 동안 AMI 원본 스냅샷의 기존 암호화 상태를 유지합니다. 암호화 기본 제공이 활성화된 경우, (암호화된 스냅샷에서든 암호화되지 않은 스냅샷에서든) AMI에서 생성된 모든 볼륨이 암호화됩니다. 암호화 기본 제공이 활성화되지 않은 경우, 인스턴스는 AMI의 암호화 상태를 유지합니다.

암호화 파라미터를 입력하여, 인스턴스를 시작하는 동시에 결과 볼륨에 새 암호화 상태를 적용할 수도 있습니다. 결과적으로 다음의 동작이 관찰됩니다.

암호화 파라미터 없이 시작

- 암호화가 기본적으로 활성화되지 않은 경우, 암호화되지 않은 스냅샷이 암호화되지 않은 볼륨으로 복원됩니다. 이런 경우 새로 생성된 모든 볼륨이 암호화됩니다.
- 소유한 암호화된 스냅샷은 동일한 CMK로 암호화된 볼륨으로 복원됩니다.
- 소유하지 않은 암호화된 스냅샷(예: AMI가 자신과 공유됨)은 AWS 계정의 기본 CMK로 암호화된 볼륨으로 복원됩니다.

암호화 파라미터를 입력하여 기본 동작을 재정의할 수 있습니다. 사용 가능한 파라미터는 `Encrypted` 및 `KmsKeyId`입니다. `Encrypted` 파라미터만 설정할 경우 그 결과는 다음과 같습니다.

`Encrypted`이(가) 설정되었지만 **`KmsKeyId`**이(가) 지정되지 않은 경우의 인스턴스 시작 동작

- 암호화되지 않은 스냅샷은 AWS 계정의 기본 CMK로 암호화된 EBS 볼륨으로 복원됩니다.
- 소유한 암호화된 스냅샷은 동일한 CMK로 암호화된 EBS 볼륨으로 복원됩니다. (즉, `Encrypted` 파라미터는 아무런 효과가 없습니다.)
- 소유하지 않은 암호화된 스냅샷(즉, AMI가 자신과 공유됨)은 AWS 계정의 기본 CMK로 암호화된 볼륨으로 복원됩니다. (즉, `Encrypted` 파라미터는 아무런 효과가 없습니다.)

`Encrypted` 및 `KmsKeyId` 파라미터를 모두 설정하면 암호화 작업에 대해 기본이 아닌 CMK를 지정할 수 있습니다. 결과는 다음 동작과 같습니다.

`Encrypted`와(과) **`KmsKeyId`**이(가) 모두 설정된 인스턴스

- 암호화되지 않은 스냅샷은 지정된 CMK로 암호화된 EBS 볼륨으로 복원됩니다.
- 암호화된 스냅샷은 원래의 CMK가 아니라 지정된 CMK로 암호화된 EBS 볼륨으로 복원됩니다.

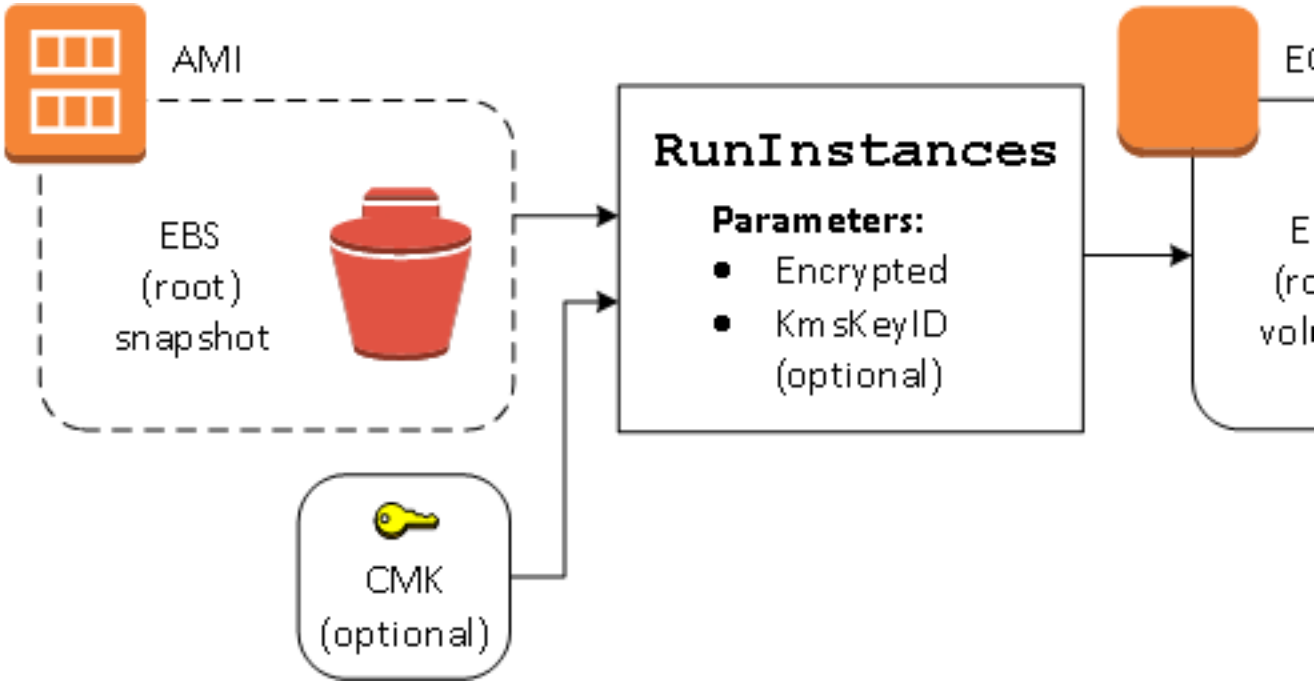
`Encrypted` 파라미터를 설정하지 않고 `KmsKeyId`을(를) 제공하면 오류가 발생합니다.

다음 단원에서는 기본이 아닌 암호화 파라미터를 사용하여 AMI에서 인스턴스를 시작하는 예제를 볼 수 있습니다. 이러한 각각의 시나리오에서 `RunInstances` 작업에 입력된 파라미터에 의해 스냅샷에서 볼륨을 복원하는 동안 암호화 상태의 변경이 유발됩니다.

> **Note**
>
> AMI에서 인스턴스를 시작하는 자세한 콘솔 절차는 인스턴스 시작 단원을 참조하십시오.
> `RunInstances` API 설명서는 RunInstances(인스턴스 실행) 단원을 참조하십시오.
> AWS Command Line Interface의 명령 `run-instances` 설명서는 run-instances(인스턴스 실행) 단원을 참조하십시오.

시작 중에 볼륨 암호화

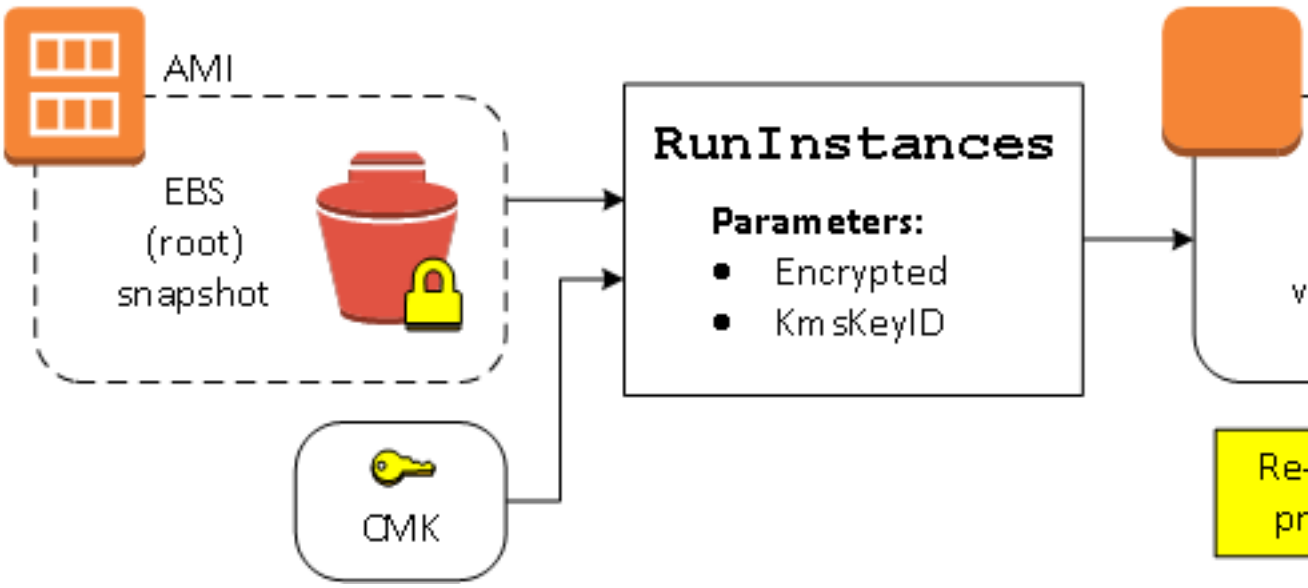
이 예제에서 암호화되지 않은 스냅샷이 지원하는 AMI는 암호화된 EBS 볼륨을 통해 EC2 인스턴스를 시작하는 데 사용됩니다.



`Encrypted` 파라미터만 사용하면 이 인스턴스의 볼륨이 암호화됩니다. `KmsKeyId` 파라미터는 선택 항목입니다. 키 ID를 지정하지 않을 경우 볼륨을 암호화는 데 AWS 계정의 기본 CMK가 사용됩니다. 소유한 다른 CMK로 사본을 암호화하려면 `KmsKeyId` 파라미터를 입력합니다.

시작 중에 볼륨 재암호화

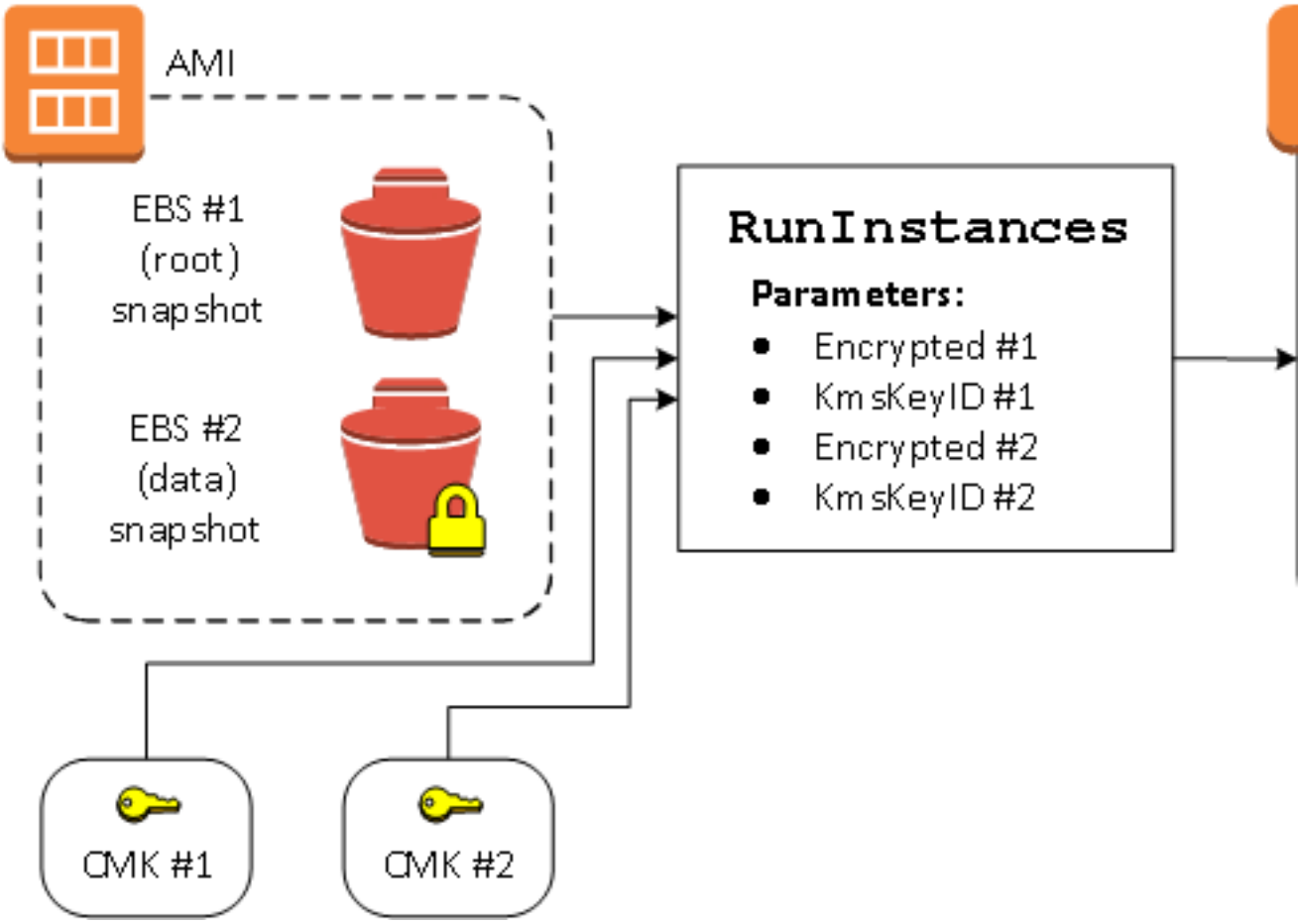
이 예제에서 암호화된 스냅샷이 지원하는 AMI는 새 CMK로 암호화된 EBS 볼륨을 통해 EC2 인스턴스를 시작하는 데 사용됩니다.



AMI를 소유한 상태에서 암호화 파라미터를 입력하지 않을 경우, 결과 인스턴스는 해당 스냅샷과 동일한 키로 암호화된 볼륨을 갖게 됩니다. AMI를 소유하지 않고 공유하며 암호화 파라미터를 입력하지 않을 경우, 볼륨이 기본 CMK로 암호화됩니다. 설명된 대로 암호화 파라미터를 입력할 경우, 볼륨이 지정된 CMK로 암호화됩니다.

시작 중에 여러 볼륨의 암호화 상태 변경

이 더 복잡한 예제에서 여러 스냅샷이 (각기 자체적인 암호화 상태를 통해) 지원하는 AMI는 새로 암호화된 볼륨과 재암호화된 볼륨을 통해 EC2 인스턴스를 시작하는 데 사용됩니다.



이 시나리오에서 `RunInstances` 작업에는 원본 스냅샷 각각에 대한 암호화 파라미터가 입력됩니다. 모든 가능한 암호화 파라미터를 지정하면, 결과 인스턴스는 AMI 소유 여부와 상관없이 동일합니다.

이미지 복사 시나리오

Amazon EC2 AMI은(는) AWS Management 콘솔을(를) 통하거나 직접 Amazon EC2 API 또는 CLI를 통해 `CopyImage` 작업을 사용해 복사됩니다.

기본적으로 명시적인 암호화 파라미터가 없는 경우. `CopyImage` 작업은 복사 중에 AMI 원본 스냅샷의 기존 암호화 상태를 유지합니다. 암호화 파라미터를 입력하여, AMI을(를) 복사하는 동시에 연결된 EBS 스냅샷에 새 암호화 상태를 적용할 수도 있습니다. 결과적으로 다음의 동작이 관찰됩니다.

암호화 파라미터 없이 복사

- 암호화가 기본적으로 활성화되지 않은 경우, 암호화되지 않은 스냅샷이 또 다른 암호화되지 않은 스냅샷으로 복사됩니다. 이런 경우 새로 생성된 모든 스냅샷이 암호화됩니다.
- 소유한 암호화된 스냅샷은 동일한 키로 암호화된 스냅샷으로 복사됩니다.
- 소유하지 않은 암호화된 스냅샷(즉, AMI가 자신과 공유됨)은 AWS 계정의 기본 CMK로 암호화된 스냅샷으로 복사됩니다.

암호화 파라미터를 입력하여 이러한 모든 기본 동작을 재정의할 수 있습니다. 사용 가능한 파라미터는 `Encrypted` 및 `KmsKeyId`입니다. `Encrypted` 파라미터만 설정할 경우 그 결과는 다음과 같습니다.

`Encrypted`이(가) 설정되었지만 **`KmsKeyId`**이(가) 지정되지 않은 경우의 이미지 복사 동작

- 암호화되지 않은 스냅샷은 AWS 계정의 기본 CMK로 암호화된 스냅샷으로 복사됩니다.
- 암호화된 스냅샷은 동일한 CMK로 암호화된 스냅샷으로 복사됩니다. (즉, `Encrypted` 파라미터는 아무런 효과가 없습니다.)
- 소유하지 않은 암호화된 스냅샷(즉, AMI가 자신과 공유됨)은 AWS 계정의 기본 CMK로 암호화된 볼륨으로 복사됩니다. (즉, `Encrypted` 파라미터는 아무런 효과가 없습니다.)

`Encrypted` 및 `KmsKeyId` 파라미터를 모두 설정하면 암호화 작업에 대해 고객 관리형 CMK를 지정할 수 있습니다. 결과는 다음 동작과 같습니다.

`Encrypted`와(과) **`KmsKeyId`**이(가) 모두 설정된 경우의 이미지 복사 동작

- 암호화되지 않은 스냅샷은 지정된 CMK로 암호화된 스냅샷으로 복사됩니다.
- 암호화된 스냅샷은 원래의 CMK가 아니라 지정된 CMK로 암호화된 스냅샷으로 복사됩니다.

`Encrypted` 파라미터를 설정하지 않고 `KmsKeyId`을(를) 제공하면 오류가 발생합니다.

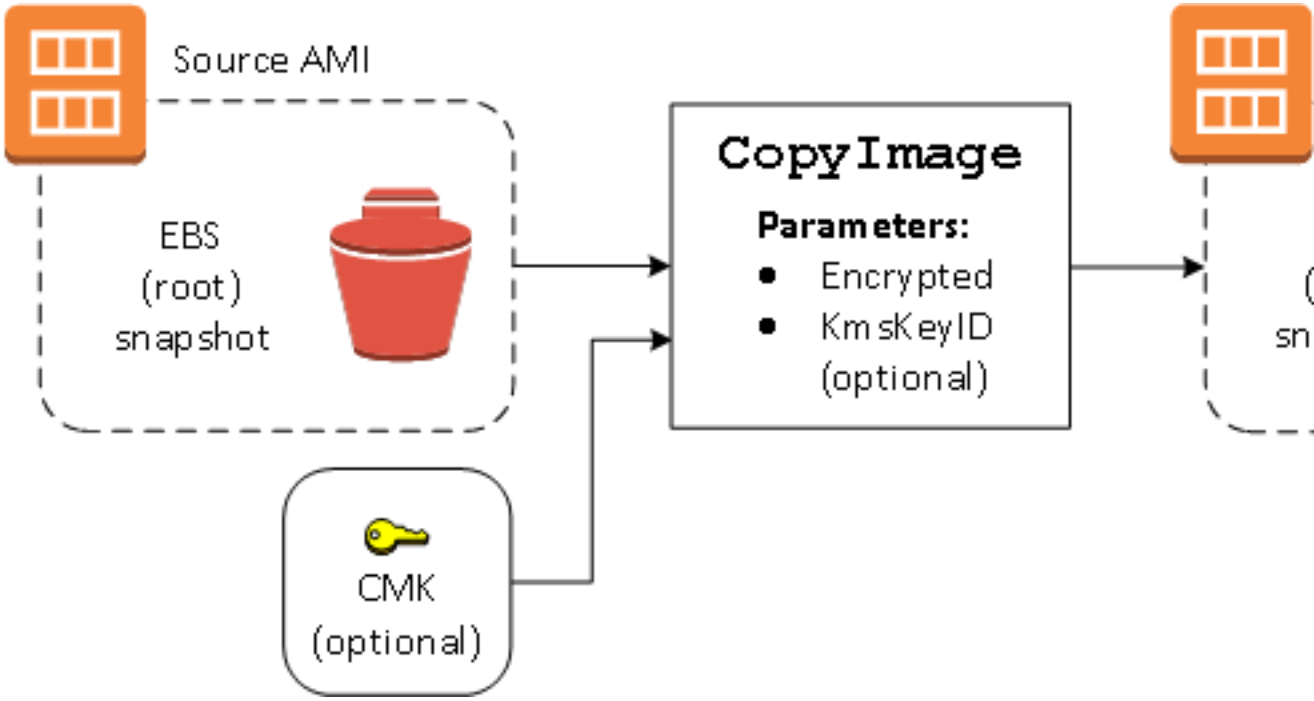
다음 단원에서는 기본이 아닌 암호화 파라미터를 사용하여 AMI를 복사하여 암호화 상태에 변경이 유발되는 예제를 제공합니다.

> **Note**
>
> AMI를 복사하는 자세한 콘솔 절차는 AMI 복사 단원을 참조하십시오.
> `CopyImage` API 설명서는 CopyImage(이미지 복사) 단원을 참조하십시오.
> AWS Command Line Interface의 명령 `copy-image` 설명서는 copy-image(이미지 복사) 단원을 참조하십시오.

복사 중에 암호화되지 않은 이미지 암호화

이 시나리오에서는 암호화되지 않은 루트 스냅샷에서 지원하는 AMI가 암호화된 루트 스냅샷이 있는 AMI로 복사됩니다. `CopyImage` 작업은 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 AMIs 모두의 스냅샷에 대한 스토리지 비용과 각 AMI에서 시작되는 인스턴스에 대한 비용이 발생합니다.

> **Note**
>
> encryption by default(암호화 기본 제공) (p. 971)를 활성화하는 것은 AMI에서 모든 스냅샷에 대해 `Encrypted` 파라미터를 `true`로 설정하는 것과 동일한 효과를 가집니다.



`Encrypted` 파라미터를 설정하면 이 인스턴스에 대한 단일 스냅샷이 암호화됩니다. `KmsKeyId` 파라미터를 지정하지 않으면 스냅샷 복사본을 암호화하는 데 기본 CMK가 사용됩니다.

Note

여러 스냅샷으로 이미지를 복사하고 각 이미지의 암호화 상태를 개별적으로 구성할 수도 있습니다.

AMI 복사

AWS Management 콘솔, AWS Command Line Interface나 SDK 또는 Amazon EC2 API(모두 `CopyImage` 작업을 지원함)를 사용하여 AWS 리전 내부 또는 전체에서 Amazon 머신 이미지(AMI)를 복사할 수 있습니다. 사용자는 Amazon EBS 지원 AMIs와 인스턴스 스토어 지원 AMIs를 모두 복사할 수 있습니다. 암호화된 스냅샷을 사용하여 AMI를 복사하고 복사 과정 중 암호화 상태도 변경할 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. (유일한 예외는 스냅샷의 암호화 또는 재암호화를 선택할 때입니다.) 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. EBS 지원 AMI를 복사하면 추가 EBS 스냅샷의 스토리지에 대한 요금이 발생합니다.

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.

사용자가 직접 가져왔거나 사용자와 공유되었는지 여부와 상관없이 AWS Marketplace에서 가져온 AMI를 복사할 수 없습니다. 대신에 AWS Marketplace AMI를 사용하여 EC2 인스턴스를 시작한 다음, 해당 인스턴스에서 AMI를 생성합니다. 자세한 내용은 Amazon EBS 지원 Linux AMI 생성 (p. 112) 단원을 참조하십시오.

인스턴스 스토어 지원 AMI 복사 권한

IAM 사용자를 사용하여 인스턴스 스토어 지원 AMI를 복사하려면 사용자에게 `s3:CreateBucket`, `s3:GetBucketAcl`, `s3:ListAllMyBuckets`, `s3:GetObject`, `s3:PutObject`, `s3:PutObjectAcl`이라는 Amazon S3 권한이 있어야 합니다.

다음 예시 정책을 통해 사용자는 지정된 버킷의 AMI 원본을 지정된 리전에 복사할 수 있습니다.

```
{
    "Version": "2012-10-17",
    "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*"
            ]
        }
    ]
}
```

AMI 원본 버킷의 Amazon 리소스 이름(ARN)을 찾으려면 https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 열고 탐색 창에서 AMI를 선택하고 소스 열에서 버킷 이름을 찾습니다.

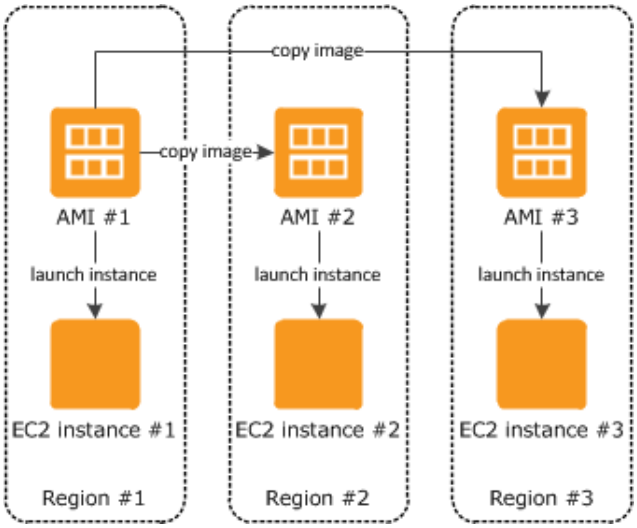
리전 간 복사

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 글로벌 애플리케이션을 보다 손쉽게 설계하고 구축할 수 있습니다.
- 성능: 애플리케이션을 분산하여 성능을 높이고 애플리케이션의 핵심 구성 요소를 사용자에게 보다 가까이 둘 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 AMIs 간 관계와 각각에서 시작된 EC2 인스턴스를 보여 줍니다. AMI에서 인스턴스를 시작하는 경우 인스턴스는 AMI가 상주하는 동일한 리전에 상주합니다.

원본 AMI를 변경한 후 대상 리전의 AMIs에 변경 내용을 반영하려면 원본 AMI를 대상 리전으로 다시 복사해야 합니다.



먼저 인스턴스 스토어 지원 AMI를 리전에 복사하는 경우 해당 리전에 복사된 AMIs에 대한 Amazon S3 버킷이 생성됩니다. 해당 리전에 복사하는 인스턴스 스토어 지원 AMIs는 모두 이 버킷에 저장됩니다. 버킷 이름 형식은 amis-for-*account*-in-*region*-*hash*를 따릅니다. 예: `amis-for-123456789012-in-us-east-2-yhjmxvp6`.

사전 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.

제한

- 대상 리전은 50개의 동시 AMI 복사본으로 제한됩니다.
- 반가상화(PV) AMI를 지원하지 않는 리전으로 PV AMI를 복사할 수 없습니다. 자세한 내용은 Linux AMI 가상화 유형 (p. 95) 단원을 참조하십시오.

계정 간 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. 102) 단원을 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. 공유 AMI에 암호화된 스냅샷이 있는 경우, 소유자는 해당 키를 사용자와도 공유해야 합니다.

암호화 및 복사

다음 표는 다양한 AMI 복사 시나리오에 대한 암호화 지원을 보여 줍니다. 암호화되지 않은 스냅샷을 복사하여 암호화된 스냅샷을 생성할 수 있지만, 암호화된 스냅샷을 복사하여 암호화되지 않은 스냅샷을 생성할 수는 없습니다.

시나리오	설명	지원
1	암호화되지 않음-암호화되지 않음	예
2	암호화됨-암호화됨	예
3	암호화되지 않음-암호화됨	예
4	암호화됨-암호화되지 않음	아니요

> **Note**
>
> `CopyImage` 작업 중 암호화는 Amazon EBS 지원 AMIs에만 적용됩니다. 인스턴스 스토어 지원 AMI에서는 스냅샷을 사용하지 않기 때문에 AMI 사본을 사용하여 암호화 상태를 변경할 수 없습니다.

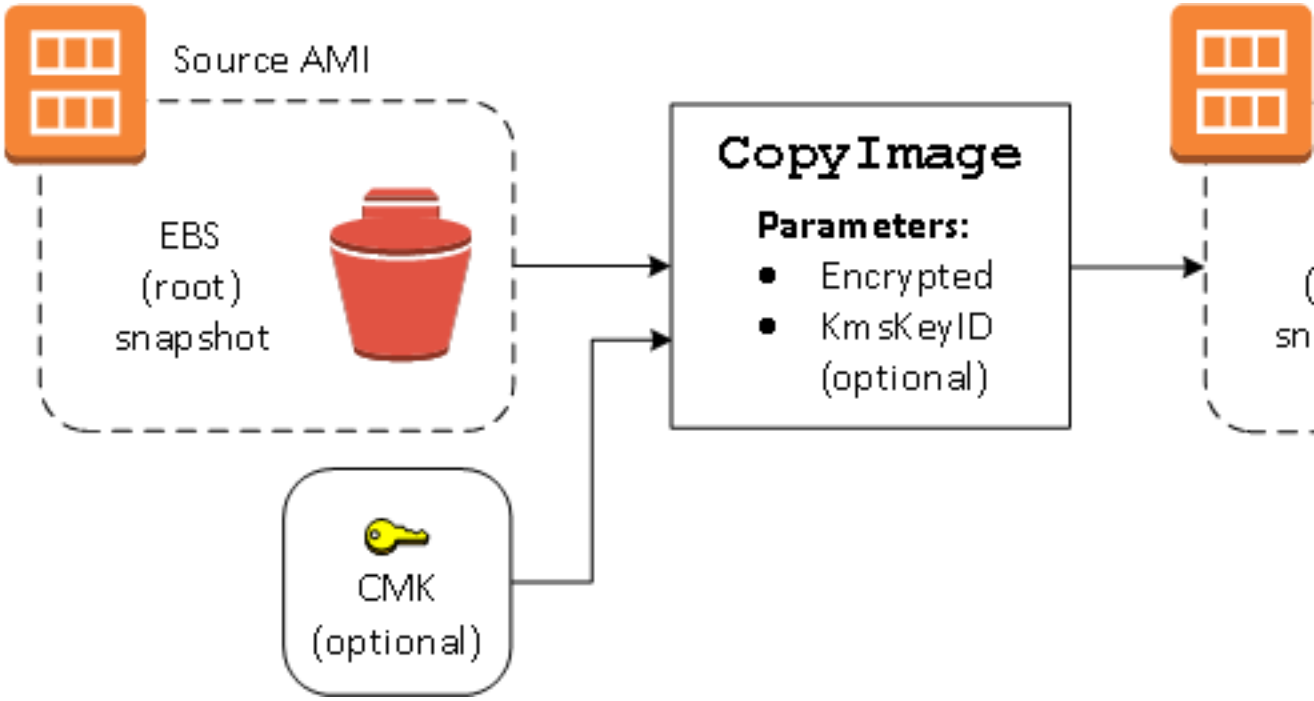
기본적으로(즉, 암호화 파라미터를 지정하지 않은 상태에서) AMI의 스냅샷에 대한 지원은 원래의 암호화 상태와 함께 복사됩니다. 암호화되지 않은 스냅샷에서 지원되는 AMI를 복사하면 역시 암호화되지 않은 동일한 대상 스냅샷이 생성됩니다. 원본 AMI가 암호화된 스냅샷에서 지원되는 경우 원본을 복사하면 동일한 고객 마스터 키(CMK)로 암호화된 동일한 대상 스냅샷이 생성됩니다. 여러 스냅샷에서 지원되는 AMI를 복사하면 각 대상 스냅샷에서 원본 암호화 상태가 유지됩니다.

AMI을(를) 복사하는 동안 암호화 파라미터를 지정하면 해당 지원 스냅샷을 암호화 또는 재암호화할 수 있습니다. 다음 예제에서는 대상 AMI의 암호화 상태를 변경하기 위해 암호화 파라미터를 `CopyImage` 작업으로 공급하는 기본이 아닌 사례를 보여 줍니다.

암호화되지 않은 원본 AMI을(를) 암호화된 대상 AMI(으)로 복사

이 시나리오에서는 암호화되지 않은 루트 스냅샷에서 지원하는 AMI가 암호화된 루트 스냅샷이 있는 AMI로 복사됩니다. `CopyImage` 작업은 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 AMIs 모두의 스냅샷에 대한 스토리지 비용과 각 AMI에서 시작되는 인스턴스에 대한 비용이 발생합니다.

> **Note**
>
> encryption by default(암호화 기본 제공) (p. 971)를 활성화하는 것은 AMI에서 모든 스냅샷에 대해 `Encrypted` 파라미터를 `true`로 설정하는 것과 동일한 효과를 가집니다.



`Encrypted` 파라미터를 설정하면 이 인스턴스에 대한 단일 스냅샷이 암호화됩니다. `KmsKeyId` 파라미터를 지정하지 않으면 스냅샷 복사본을 암호화하는 데 기본 CMK가 사용됩니다.

암호화된 스냅샷을 포함하는 AMIs 복사에 대한 자세한 내용은 EBS 지원 AMI를 통한 암호화 사용 (p. 145) 단원을 참조하십시오.

AMI 복사

다음과 같이 AMI를 복사할 수 있습니다.

사전 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. 112) 및 Linux AMI 찾기 (p. 96) 단원을 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 이미지, AMI를 선택하여 리전에서 사용할 수 있는 AMIs 목록을 표시합니다.
3. 복사할 AMI를 선택하고 작업, AMI 복사를 선택합니다.
4. AMI 복사 대화 상자에서 다음 정보를 지정하고 AMI 복사를 지정합니다.
   - 대상 리전: AMI를 복사할 리전.
   - 이름: 새 AMI의 이름. AMI에 대한 세부 정보를 표시할 때 운영 체제 정보가 제공되지 않으므로, 이름에 운영 체제 정보를 넣을 수 있습니다.
   - 설명: 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. 암호화 기본 제공을 활성화한 경우 암호화 옵션이 설정되어 AMI 콘솔로부터 설정 해제할 수 없습니다.

- 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 완료를 선택한 후, 준비가 되면 탐색 모음을 사용하여 대상 리전으로 전환하고(해당하는 경우) AMI 목록에서 해당 AMI를 찾습니다.

   대상 AMI의 초기 상태는 `pending`이고 작업이 완료되면 상태가 `available`이 됩니다.

AWS CLI를 사용하여 AMI를 복사하려면

copy-image 명령을 사용하여 AMI를 복사할 수 있습니다. 원본 리전과 대상 리전을 모두 지정해야 합니다. `--source-region` 파라미터를 사용하여 원본 리전을 지정합니다. `--region` 파라미터 또는 환경 변수를 사용하여 대상 리전을 지정할 수 있습니다. 자세한 정보는 AWS 명령줄 인터페이스 구성을 참조하십시오.

복사 중에 대상 스냅샷을 암호화하는 경우, `--encrypted` 및 `--kms-key-id` 파라미터를 추가로 지정해야 합니다.

Windows PowerShell용 도구를 사용하여 AMI를 복사하려면

Copy-EC2Image 명령을 사용하여 AMI를 복사할 수 있습니다. 원본 리전과 대상 리전을 모두 지정해야 합니다. `-SourceRegion` 파라미터를 사용하여 원본 리전을 지정합니다. `-Region` 파라미터 또는 `Set-AWSDefaultRegion` 명령을 사용하여 대상 리전을 지정할 수 있습니다. 자세한 정보는 AWS 리전 지정을 참조하십시오.

복사 중에 대상 스냅샷을 암호화하는 경우, `-Encrypted` 및 `-KmsKeyId` 파라미터를 추가로 지정해야 합니다.

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 작업, 등록 해제를 차례대로 선택합니다.
5. 확인 메시지가 표시되면 계속을 선택합니다.

명령줄을 사용하여 AMI 복사 작업을 중지하려면 다음을 수행합니다.

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- deregister-image(AWS CLI)
- Unregister-EC2Image(Windows PowerShell용 AWS 도구)

결제 정보 가져오기

온디맨드 인스턴스 또는 스팟 인스턴스를 시작하거나 예약 인스턴스를 구매하기 전에 Amazon Machine Image(AMI)와 연결된 플랫폼 세부 정보 및 결제 정보를 확인할 수 있습니다. 스팟 인스턴스의 경우 플랫폼

세부 정보를 사용하여 스팟 인스턴스에 대해 AMI가 지원되는지를 확인할 수 있습니다. 예약 인스턴스를 구매할 때 플랫폼에서 AMI의 `PlatformDetails`에 매핑할 올바른 값을 선택합니다. 인스턴스를 시작하거나 예약 인스턴스를 구매하기 전에 결제 정보를 알면 잘못된 AMI에서 인스턴스를 잘못 시작하고 계획되지 않은 비용이 발생할 가능성을 줄일 수 있습니다.

인스턴스 요금에 대한 자세한 내용은 Amazon EC2 요금을 참조하십시오.

목차

- AMI 결제 정보 필드 (p. 156)
- 플랫폼 세부 정보 및 사용 작업 결제 코드 (p. 156)
- 플랫폼 세부 정보 및 결제 정보 보기 (p. 157)

AMI 결제 정보 필드

describe-images 명령이 반환하는 AMI에 대한 정보는 다음과 같습니다.

PlatformDetails

AMI의 결제 코드와 연결된 플랫폼 세부 정보입니다. 예: `Red Hat Enterprise Linux`.

UsageOperation

AMI와 연결된 Amazon EC2 인스턴스 및 결제 코드의 작업입니다. `UsageOperation`은 AWS 가격표 API와 AWS 비용 및 사용 보고서(CUR)의 lineitem/Operation 열에 해당합니다. `UsageOperation` 코드의 목록은 다음 섹션의 플랫폼 세부 정보 및 사용 작업 결제 코드 (p. 156) 단원을 참조하십시오.

플랫폼 세부 정보 및 사용 작업 결제 코드

다음 표에는 AMI에서 describe-images 명령을 실행할 때 반환될 수 있는 `PlatformDetails` 및 `UsageOperation` 값이 나열되어 있습니다.

PlatformDetails	UsageOperation **
Linux/UNIX	RunInstances
Red Hat BYOL Linux	RunInstances:00g0
Red Hat Enterprise Linux	RunInstances:0010
SQL Server Enterprise	RunInstances:0100
SQL Server Standard	RunInstances:0004
SQL Server Web	RunInstances:0200
SUSE Linux	RunInstances:000g
Windows	RunInstances:0002
Windows BYOL	RunInstances:0800
SQL Server Enterprise가 설치된 Windows *	RunInstances:0102
SQL Server Standard가 설치된 Windows *	RunInstances:0006
SQL Server Web이 설치된 Windows *	RunInstances:0202

* AMI에 두 개의 소프트웨어 라이선스가 연결된 경우 `PlatformDetails` 필드에 모두 표시됩니다.

** 스팟 인스턴스를 실행 중인 경우 AWS 비용 및 사용 보고서의 `lineitem/Operation`은 여기 나열된 `UsageOperation` 값과 다를 수 있습니다. 예를 들어, `lineitem/Operation`에 `RunInstances:0010:SV006`이 표시되는 경우 Amazon EC2가 미국 동부(버지니아)의 VPC 영역 #6에서 Red Hat Enterprise Linux 스팟 인스턴스 시간을 실행하고 있는 것입니다.

플랫폼 세부 정보 및 결제 정보 보기

AMI와 연결된 플랫폼 세부 정보 및 결제 정보를 보려면

describe-images 명령을 사용합니다.

```
$ aws ec2 describe-images --image-ids ami-0123456789EXAMPLE
```

다음 예제 출력은 `PlatformDetails` 및 `UsageOperation` 필드를 보여줍니다. 이 예에서 ami-0123456789EXAMPLE 플랫폼은 `Red Hat Enterprise Linux`이며, 사용 작업은 `RunInstances:0010`입니다.

```
{
    "Images": [
        {
            "VirtualizationType": "hvm",
            "Description": "Provided by Red Hat, Inc.",
            "Hypervisor": "xen",
            "EnaSupport": true,
            "SriovNetSupport": "simple",
            "ImageId": "ami-0123456789EXAMPLE",
            "State": "available",
            "BlockDeviceMappings": [
                {
                    "DeviceName": "/dev/sda1",
                    "Ebs": {
                        "SnapshotId": "snap-111222333444aaabb",
                        "DeleteOnTermination": true,
                        "VolumeType": "gp2",
                        "VolumeSize": 10,
                        "Encrypted": false
                    }
                }
            ],
            "Architecture": "x86_64",
            "ImageLocation": "123456789012/RHEL-8.0.0_HVM-20190618-x86_64-1-Hourly2-GP2",
            "RootDeviceType": "ebs",
            "OwnerId": "123456789012",
            "PlatformDetails": "Red Hat Enterprise Linux",
            "UsageOperation": "RunInstances:0010",
            "RootDeviceName": "/dev/sda1",
            "CreationDate": "2019-05-10T13:17:12.000Z",
            "Public": true,
            "ImageType": "machine",
            "Name": "RHEL-8.0.0_HVM-20190618-x86_64-1-Hourly2-GP2"
        }
    ]
}
```

계획되지 않은 비용이 발생하지 않도록 AWS 비용 및 사용 보고서의 결제 정보가 describe-images에서 반환된 결제 정보와 일치하는지 확인할 수 있습니다. ami-0123456789EXAMPLE을 사용하여 인스턴스를 시작한 경우 AWS 비용 및 사용 보고서에서 인스턴스에 대한 결제 정보를 확인할 수 있습니다. 인스턴스 ID를 찾아 `lineitem/Operation` 열에서 해당 값을 확인합니다. 이 예제에서 값은 `RunInstances:0010`이어야 합니다.

인스턴스와 연결된 플랫폼 세부 정보 및 결제 정보를 보려면

인스턴스를 시작한 후에는 인스턴스 메타데이터의 `billingProducts` 필드를 검사하여 결제 정보를 찾을 수 있습니다. 자세한 내용은 인스턴스 자격 증명 문서 및 서명 가져오기 (p. 594) 단원을 참조하십시오. 또는 describe-instances 명령을 사용하여 인스턴스에 대한 AMI ID를 가져온 다음 위의 절차에서 설명한 대로 describe-images 명령을 사용하여 응답의 `PlatformDetails` 및 `UsageOperation` 필드에서 결제 정보를 가져올 수 있습니다.

Linux 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. 자세한 내용은 AMI의 루트 디바이스 유형 결정 (p. 94) 단원을 참조하십시오.

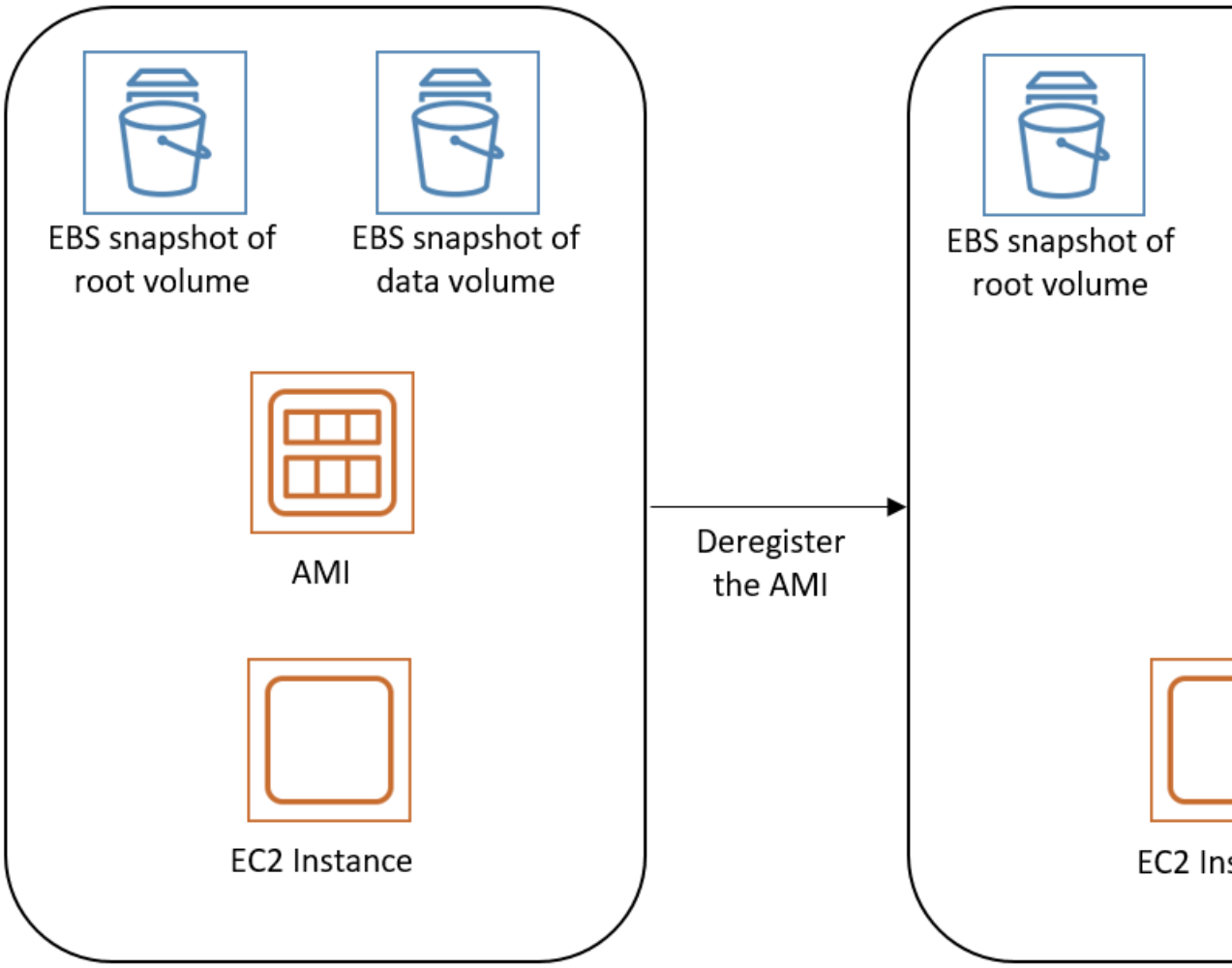
내용

- Amazon EBS 기반 AMI 정리 (p. 158)
- 인스턴스 스토어 기반 AMI 정리 (p. 160)

Amazon EBS 기반 AMI 정리

Amazon EBS 기반 AMI의 등록을 취소하는 경우, AMI 생성 과정에서 인스턴스의 볼륨에 대해 생성된 스냅샷에는 영향을 주지 않습니다. 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.



Your AMI, its snapshots, and an instance launched from the AMI

Amazon EBS 기반 AMI를 정리하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 AMI를 선택합니다. AMI를 선택하고 그 ID를 메모해 놓습니다. 그러면 다음 단계에서 올바른 스냅샷을 쉽게 찾을 수 있습니다. 작업을 선택한 후 등록 취소를 선택합니다. 확인 메시지가 표시되면 계속을 선택합니다.

   Note

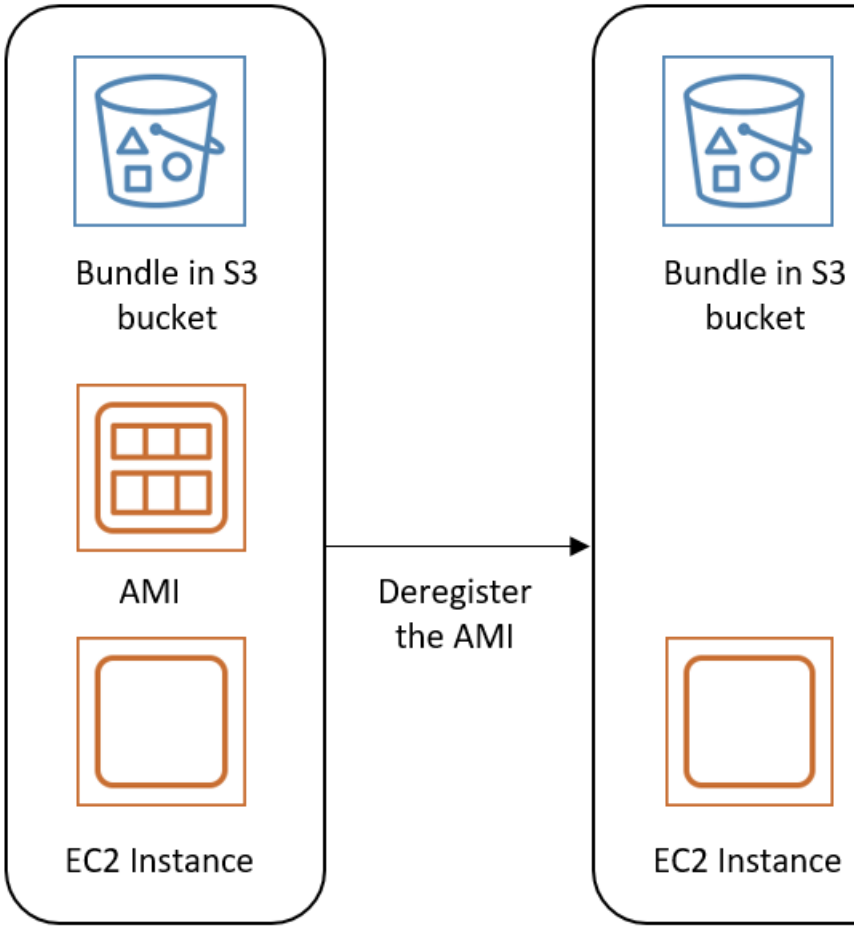
   콘솔이 목록에서 AMI를 제거하는 데 몇 분 정도 걸릴 수 있습니다. 상태를 새로 고치려면 새로 고침을 선택합니다.
3. 탐색 창에서 스냅샷을 선택한 후 스냅샷을 선택합니다(설명 열에서 AMI ID를 검색). 작업을 선택한 후 스냅샷 삭제를 선택합니다. 확인 메시지가 나타나면 예, 삭제를 선택합니다.

4. (선택 사항) AMI에서 시작한 인스턴스 관련 작업이 완료되면 해당 인스턴스를 종료합니다. 탐색 창에서 인스턴스를 선택합니다. 인스턴스를 선택하고 작업을 선택한 후 인스턴스 상태와 종료를 차례로 선택합니다. 확인 메시지가 나타나면 예, 종료를 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.



Your AMI, its bundle in Amazon S3, and an instance launched from the AMI

인스턴스 스토어 기반 AMI를 정리하려면

1. 다음과 같이 deregister-image 명령을 사용하여 AMI의 등록을 취소합니다.

```
aws ec2 deregister-image --image-id ami_id
```

2. 다음과 같이 ec2-delete-bundle (p. 135)(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
```

3. (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
```

4. (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 작업, 삭제를 차례로 선택합니다.

Amazon Linux

Amazon Linux는 Amazon Web Services(AWS)에서 제공하며, Amazon EC2에서 실행 중인 애플리케이션에 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. CentOS 및 다른 유사 배포에서 개발한 많은 애플리케이션이 Amazon Linux에서 실행됩니다.

목차

- Amazon Linux 가용성 (p. 161)
- Amazon Linux 인스턴스에 연결 (p. 162)
- Amazon Linux 이미지 식별 (p. 162)
- AWS 명령줄 도구 (p. 163)
- 패키지 리포지토리 (p. 164)
- Extras Library(Amazon Linux 2) (p. 166)
- 참조용으로 원본 패키지에 액세스 (p. 166)
- cloud-init (p. 166)
- Amazon Linux 알림 구독 (p. 168)
- Amazon Linux 2를 온프레미스 가상 머신으로 실행 (p. 169)

Amazon Linux 가용성

AWS는 Amazon Linux 2 및 Amazon Linux AMI를 제공합니다. 다른 Linux 배포판에서 Amazon Linux 2로 마이그레이션하는 경우 Amazon Linux로 마이그레이션하는 것이 좋습니다.

Amazon Linux AMI의 최신 버전인 2018.03은 2020년 12월 31일에 표준 지원이 종료됩니다. 자세한 내용은 블로그 게시물 Amazon Linux AMI 지원 종료를 참조하십시오. 현재 Amazon Linux AMI를 사용하고 있다면 Amazon Linux 2로 마이그레이션하는 것이 좋습니다. Amazon Linux 2로 마이그레이션하려면 현재 Amazon Linux 2 이미지를 사용하여 인스턴스를 시작하거나 가상 머신을 생성합니다. 애플리케이션과 필수 패키지를 설치합니다. 애플리케이션을 테스트하고 Amazon Linux 2에서 애플리케이션을 실행하는 데 필요한 변경을 수행합니다.

자세한 내용은 Amazon Linux 2 및 Amazon Linux AMI를 참조하십시오. Amazon Linux Docker 컨테이너 이미지는 Docker Hub의 amazonlinux를 참조하십시오.

Amazon Linux 인스턴스에 연결

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-user뿐입니다. 이 계정에는 sudo 권한도 있습니다. 원격 루트 로그인을 사용하는 경우 키 페어와 부 사용자에 의존하는 방법보다는 덜 안전하다는 점에 유의하십시오.

Amazon Linux 이미지 식별

각 이미지에는 식별을 위한 고유한 `/etc/image-id` 파일이 포함되어 있습니다. 이 파일에는 이미지에 대한 다음 정보가 포함되어 있습니다.

- `image_name`, `image_version`, `image_arch` — Amazon에서 이미지 생성에 사용한 빌드 레시피에서 가져온 값입니다.
- `image_stamp` — 이미지 생성 중에 생성된 고유한 임의 16진수 값입니다.
- `image_date` — YYYYMMDDhhmmss 형식이며 이미지 생성 시간(UTC)입니다.
- `recipe_name`, `recipe_id` — Amazon에서 이미지 생성에 사용한 빌드 레시피의 이름과 ID입니다.

Amazon Linux에는 설치된 현재 릴리스를 지정하는 `/etc/system-release` 파일이 포함되어 있습니다. 이 파일은 yum을 사용하여 업데이트되며 `system-release` RPM의 일부입니다.

Amazon Linux에는 `/etc/system-release`의 머신 판독 가능한 버전도 포함되어 있으며 CPE 사양을 따릅니다. `/etc/system-release-cpe` 단원을 참조하십시오.

Amazon Linux 2

현재 버전의 Amazon Linux 2에 대한 `/etc/image-id`의 예제입니다.

```
[ec2-user ~]$ cat /etc/image-id
image_name="amzn2-ami-hvm"
image_version="2"
image_arch="x86_64"
image_file="amzn2-ami-hvm-2.0.20180810-x86_64.xfs.gpt"
image_stamp="8008-2abd"
image_date="20180811020321"
recipe_name="amzn2 ami"
recipe_id="c652686a-2415-9819-65fb-4dee-9792-289d-1e2846bd"
```

현재 버전의 Amazon Linux 2에 대한 `/etc/system-release`의 예제입니다.

```
[ec2-user ~]$ cat /etc/system-release
Amazon Linux 2
```

다음은 Amazon Linux 2에 대한 `/etc/os-release`의 예제입니다.

```
[ec2-user ~]$ cat /etc/os-release
NAME="Amazon Linux"
VERSION="2"
ID="amzn"
ID_LIKE="centos rhel fedora"
VERSION_ID="2"
PRETTY_NAME="Amazon Linux 2"
ANSI_COLOR="0;33"
```

```
CPE_NAME="cpe:2.3:o:amazon:amazon_linux:2"
HOME_URL="https://amazonlinux.com/"
```

Amazon Linux AMI

현재 버전의 Amazon Linux AMI에 대한 `/etc/image-id`의 예제입니다.

```
[ec2-user ~]$ cat /etc/image-id
image_name="amzn-ami-hvm"
image_version="2018.03"
image_arch="x86_64"
image_file="amzn-ami-hvm-2018.03.0.20180811-x86_64.ext4.gpt"
image_stamp="cc81-f2f3"
image_date="20180811012746"
recipe_name="amzn ami"
recipe_id="5b283820-dc60-a7ea-d436-39fa-439f-02ea-5c802dbd"
```

현재 버전의 Amazon Linux AMI에 대한 `/etc/system-release`의 예제입니다.

```
[ec2-user ~]$ cat /etc/system-release
Amazon Linux AMI release 2018.03
```

AWS 명령줄 도구

AWS 통합 및 사용을 위해 다음과 같은 명령줄 도구가 Amazon Linux AMI 또는 Amazon Linux 2의 기본 리포지토리에 포함되었습니다. Amazon Linux AMI의 전체 패키지 목록은 Amazon Linux AMI 2017년 9월 패키지를 참조하십시오.

- aws-amitools-ec2
- aws-apitools-as
- aws-apitools-cfn
- aws-apitools-ec2
- aws-apitools-elb
- aws-apitools-mon
- aws-cfn-bootstrap
- aws-cli

Amazon Linux 2 및 Amazon Linux(`amzn-ami-minimal-*` 및 `amzn2-ami-minimal-*`)의 최소 버전에는 이 패키지가 항상 전부 포함되어 있는 것은 아닙니다. 그러나 다음 명령을 사용하여 기본 리포지토리에서 설치할 수 있습니다.

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

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`가 제품별 환경 변수(예: `EC2_HOME`)를 설정하는 데 사용합니다.

패키지 리포지토리

Amazon Linux 2와 Amazon Linux AMI는 각 Amazon EC2 AWS 리전에서 호스팅되는 온라인 패키지 리포지토리와 함께 사용하도록 설계되었습니다. 이러한 리포지토리는 Amazon Linux 2 및 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 리전에서 사용할 수 있습니다. 각 리전에서 리포지토리를 호스팅하면 데이터 전송 요금 없이 데이터를 신속히 배포할 수 있습니다.

Amazon Linux 2 및 Amazon Linux AMI는 정기적으로 업데이트되어 보안 및 기능 향상을 제공합니다. 인스턴스에 대한 데이터 또는 사용자 지정을 유지할 필요가 없는 경우 단순히 현재 AMI를 사용하여 새 인스턴스를 다시 시작할 수 있습니다. 인스턴스에 대한 데이터나 사용자 지정을 유지해야 하는 경우에는 Amazon Linux 패키지 리포지토리를 통해 이러한 인스턴스를 유지할 수 있습니다. 이러한 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.

> **Important**
>
> 인스턴스는 리포지토리에 액세스하기 위해 인터넷에 액세스할 수 있어야 합니다.

패키지를 설치하려면 다음 명령을 사용합니다.

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

Amazon Linux AMI의 경우, Extra Packages for Enterprise Linux(EPEL) 리포지토리에 대한 액세스가 구성되었지만 기본적으로 사용되지 않습니다. Amazon Linux 2가 EPEL 리포지토리를 사용하도록 구성되지 않았습니다. EPEL은 리포지토리에 있는 패키지 이외의 타사 패키지를 제공합니다. 타사 패키지는 AWS에서 지원되지 않습니다. 다음 명령으로 EPEL 리포지토리를 활성화할 수 있습니다.

- 대상 Amazon Linux 2:

```
[ec2-user ~]$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-
latest-7.noarch.rpm
```

- Amazon Linux AMI의 경우:

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

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 사용할 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 업로드하려면 scp 또는 sftp를 사용하고 인스턴스에 로그온하여 애플리케이션을 구성합니다. 또한 기본 제공된 cloud-init 패키지의 PACKAGE_SETUP 작업을 사

용하여 인스턴스 시작 중에 애플리케이션을 업로드할 수도 있습니다. 자세한 내용은 cloud-init (p. 166) 단원을 참조하십시오.

보안 업데이트

보안 업데이트는 패키지 리포지토리와 업데이트된 AMI를 사용하여 제공됩니다. 보안 알림은 Amazon Linux 보안 센터에 게시됩니다. AWS 보안 정책에 대한 자세한 내용을 찾아보거나 보안 문제를 보고하려면 AWS 보안 센터로 이동하십시오.

Amazon Linux는 시작 시 심각한거나 중요한 보안 업데이트를 다운로드 및 설치하도록 구성되어 있습니다. 시작 후에는 사용 사례를 위해 필수 업데이트를 설치하는 것이 좋습니다. 예를 들면 시작 시 보안 업데이트뿐 아니라 모든 업데이트를 적용하거나, 각 업데이트를 평가하고 시스템에 해당하는 업데이트만 적용할 수도 있습니다. 이는 다음과 같은 cloud-init 설정 `repo_upgrade`를 사용하여 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`에 가능한 값은 다음과 같습니다.

`security`

Amazon에서 보안 업데이트로 표시하는 대기 중인 중요하거나 심각한 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 security입니다. 즉, 기본적으로 사용자 데이터에 다른 값을 지정하지 않은 경우 Amazon Linux에서는 해당 시점에 설치된 모든 패키지에 대해 시작 시 보안 업그레이드를 수행합니다. Amazon Linux에서는 `/etc/motd` 파일을 사용하여 로그인 시 사용 가능한 업데이트를 나열함으로써 설치된 패키지에 대한 업데이트를 사용자에게 알립니다. 이러한 업데이트를 설치하려면 인스턴스에서 sudo yum upgrade를 실행해야 합니다.

리포지토리 구성

Amazon Linux 사용을 통해 AMI가 시간의 스냅샷으로 처리되며, yum update -y를 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(2017.09 이하)의 인스턴스를 시작하고 yum update -y를 실행할 경우 최신 패키지가 제공됩니다.

lock-on-launch 기능을 사용하도록 설정하여 롤링 업데이트를 사용하지 않도록 설정할 수 있습니다. lock-on-launch 기능은 지정된 AMI 릴리스에서만 업데이트를 받도록 인스턴스를 잠급니다. 예를 들어, 2017.09 AMI를 시작한 후, 2018.03 AMI로 마이그레이션할 준비될 때까지 2018.03 AMI 이전에 릴리스된 업데이트만 받도록 할 수 있습니다.

> **Important**
>
> 최신 버전이 아닌 리포지토리에 고정하면 추가 업데이트가 수신되지 않습니다. 업데이트 흐름을 지속적으로 수신하려면 최신 AMI를 사용하거나 리포지토리가 최신 버전을 가리키도록 AMI를 계속 업데이트합니다.

새 인스턴스에서 lock-on-launch를 사용하려면 cloud-init에 전달된 다음 사용자 데이터로 인스턴스를 시작하십시오.

```
#cloud-config
repo_releasever: 2017.09
```

기존 인스턴스를 현재 AMI 버전으로 잠그려면

1. 편집 `/etc/yum.conf`.
2. `releasever=latest`를 주석으로 처리합니다.
3. 캐시를 지우려면 yum clean all을 실행합니다.

Extras Library(Amazon Linux 2)

Amazon Linux 2를 사용하면 Extras Library를 사용하여 인스턴스에 애플리케이션 및 소프트웨어 업데이트를 설치할 수 있습니다. 이러한 소프트웨어 업데이트를 주제라고 합니다. 특정 버전의 주제를 설치하거나 버전 정보를 생략하여 최신 버전을 사용할 수 있습니다.

사용 가능한 주제를 나열하려면 다음 명령을 사용하십시오.

```
[ec2-user ~]$ amazon-linux-extras list
```

주제를 활성화하고 패키지의 최신 버전을 설치하려면 다음 명령을 사용하십시오.

```
[ec2-user ~]$ sudo amazon-linux-extras install topic
```

안정화를 위해 주제를 활성화하고 해당 패키지의 특정 버전을 설치하려면 다음 명령을 사용하십시오.

```
[ec2-user ~]$ sudo amazon-linux-extras install topic=version topic=version
```

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 yumdownloader --source 명령을 사용하여 실행 중인 인스턴스 내에서 원본을 확인합니다. 다음 예를 참조하십시오.

```
[ec2-user ~]$ yumdownloader --source bash
```

원본 RPM의 압축을 풀고 스탠다드 RPM 도구를 사용하여 참조용으로 원본 트리를 볼 수 있습니다. 디버깅을 완료한 후에는 패키지를 사용할 수 있습니다.

cloud-init

cloud-init 패키지는 Canonical에서 오픈 소스 애플리케이션이며 Amazon EC2 등의 클라우드 컴퓨팅 환경에서 Linux 이미지 부트스트랩을 수행하는 데 사용됩니다. Amazon Linux에는 cloud-init의 사용자 지정 버전이 포함되어 있습니다. 부팅 시 인스턴스에 대해 발생할 작업을 지정할 수 있습니다. 인스턴스를 시작할 때 사용자 데이터 필드를 통해 원하는 작업을 cloud-init로 전달할 수 있습니다. 다시 말해서 여러 사용 사례에 공통 AMI를 사용하고 이러한 AMI를 시작 시 동적으로 구성할 수 있다는 뜻입니다. 또한 Amazon Linux에서는 cloud-init를 사용하여 ec2-user 계정의 초기 구성도 수행합니다.

자세한 내용은 cloud-init 설명서를 참조하십시오.

Amazon Linux는 `/etc/cloud/cloud.cfg.d` 및 `/etc/cloud/cloud.cfg`에 있는 cloud-init 작업을 사용합니다. `/etc/cloud/cloud.cfg.d`에서 자체 cloud-init 작업 파일을 만들 수 있습니다. 이 디렉터리의 모든 파일은 cloud-init으로 읽습니다. 어휘 순으로 읽히며, 나중의 파일이 이전 파일의 값을 덮어씁니다.

cloud-init 패키지는 부팅 시 인스턴스에 대해 다음과 같은 일반적인 구성 작업을 수행합니다.

- 기본 로캘 설정.
- 호스트 이름 설정.
- 사용자 데이터 구문 분석 및 처리.
- 호스트 프라이빗 SSH 키 생성.
- 손쉬운 로그인 및 관리를 위해 사용자의 퍼블릭 SSH 키를 `.ssh/authorized_keys`에 추가.
- 패키지 관리를 위한 리포지토리 준비.
- 사용자 데이터에 정의된 패키지 작업 처리.
- 사용자 데이터에 있는 사용자 스크립트 실행.
- 해당하는 경우 인스턴스 스토어 볼륨을 탑재합니다.
  - 기본적으로 `ephemeral0` 인스턴스 스토어 볼륨은 `/media/ephemeral0`이 있고 유효한 파일 시스템이 포함된 경우 여기에 마운트됩니다. 그렇지 않은 경우 마운트되지 않습니다.
  - 기본적으로 인스턴스와 연결된 모든 스왑 볼륨이 탑재됩니다(`m1.small` 및 `c1.medium` 인스턴스 유형만 해당).
  - 다음 cloud-init 명령을 사용하여 기본 인스턴스 스토어 볼륨 탑재를 재정의할 수 있습니다.

    ```
    #cloud-config
    mounts:
    - [ ephemeral0 ]
    ```

    탑재를 더 세부적으로 제어하려면 cloud-init 설명서의 탑재를 참조하십시오.
  - TRIM을 지원하는 인스턴스 스토어 볼륨은 인스턴스를 시작할 때 포맷되지 않으므로, 볼륨을 파티셔닝하고 포맷한 후 탑재해야 합니다. 자세한 내용은 인스턴스 스토어 볼륨 TRIM 지원 (p. 1037) 단원을 참조하십시오. `disk_setup` 모듈을 사용하여 부팅 시 인스턴스 스토어 볼륨을 파티셔닝하고 포맷할 수 있습니다. 자세한 내용은 cloud-init 문서의 Disk Setup 단원을 참조하십시오.

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 인식할 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-shellscript`로 시작합니다.

  - 스크립트는 첫 부팅 사이클 도중 `/etc/init.d/cloud-init-user-scripts`에 의해 실행됩니다. 이 동작은 부팅 프로세스 후반(초기 구성 작업이 수행된 후)에 발생합니다.
- Include 파일
  - `#include` 또는 `Content-Type: text/x-include-url`로 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-config`로 시작합니다.
  - 이 내용은 클라우드 구성 데이터입니다. 지원되는 구성 형식의 주석 포함 예는 해당 예를 참조하십시오.
- Upstart 작업
  - `#upstart-job` 또는 `Content-Type: text/upstart-job`로 시작합니다.
  - 이 콘텐츠는 `/etc/init`에서 파일에 저장되고, upstart는 다른 upstart 작업에 따라 콘텐츠를 소비합니다.
- 클라우드 Boothook
  - `#cloud-boothook` 또는 `Content-Type: text/cloud-boothook`로 시작합니다.
  - 이것은 bookhook 데이터의 내용입니다. `/var/lib/cloud`에 있는 파일에 저장된 후 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 boothook 데이터의 인스턴스당 1회 세트를 제공하십시오.

Amazon Linux 알림 구독

최신 AMI가 배포될 때 알림을 받으려면 Amazon SNS를 사용하여 구독할 수 있습니다.

Amazon Linux 알림을 구독하려면

1. https://console.aws.amazon.com/sns/v3/home에서 Amazon SNS 콘솔을 엽니다.
2. 필요한 경우 탐색 모음에서 리전을 미국 동부(버지니아 북부)로 변경하십시오. 구독하려는 SNS 알림이 생성된 리전을 선택해야 합니다.
3. 탐색 창에서 구독과 구독 생성을 선택합니다.
4. 구독 생성 대화 상자에서 다음과 같이 수행합니다.
   a. [Amazon Linux 2] 주제 ARN의 경우, 다음 Amazon 리소스 이름(ARN)을 복사합니다. **`arn:aws:sns:us-east-1:137112412989:amazon-linux-2-ami-updates`**.
   b. [Amazon Linux] 주제 ARN의 경우, 다음 Amazon 리소스 이름(ARN)을 복사합니다. **`arn:aws:sns:us-east-1:137112412989:amazon-linux-ami-updates`**.
   c. 프로토콜에서 이메일을 선택합니다.
   d. 엔드포인트에는 알림을 받는 데 사용할 수 있는 이메일 주소를 입력합니다.
   e. Create subscription을 선택합니다.
5. "AWS Notification - Subscription Confirmation"이라는 제목의 확인 이메일을 받게 됩니다. 이메일을 열고 구독 확인을 선택하여 구독을 완료합니다.

AMI가 릴리스될 때마다, 해당 주제의 구독자에게 알림이 발송됩니다. 이 알림을 수신하지 않으려면 다음 절차를 수행하여 구독 해제합니다.

Amazon Linux 알림을 구독 해제하려면

1. https://console.aws.amazon.com/sns/v3/home에서 Amazon SNS 콘솔을 엽니다.

2. 필요한 경우 탐색 모음에서 리전을 미국 동부(버지니아 북부)로 변경하십시오. SNS 알림이 생성된 리전을 사용해야 합니다.
3. 탐색 창에서 구독을 선택하고 해당 구독을 선택한 후 작업, 구독 삭제를 선택합니다.
4. 확인 메시지가 나타나면 Delete를 선택합니다.

Amazon Linux 2를 온프레미스 가상 머신으로 실행

온프레미스 개발 및 테스트를 위해 Amazon Linux 2 가상 머신(VM) 이미지를 사용합니다. 이러한 이미지는 다음 가상 플랫폼에서 사용할 수 있습니다.

- VMware
- KVM
- VirtualBox(Oracle VM)
- Microsoft Hyper-V

지원되는 가상화 플랫폼 중 하나에서 Amazon Linux 2 가상 머신 이미지를 사용하려면 다음을 수행하십시오.

- 1단계: `seed.iso` 부팅 이미지 준비 (p. 169)
- 2단계: Amazon Linux 2 VM 이미지 다운로드 (p. 171)
- 3단계: 새 VM 부팅 및 연결 (p. 171)

1단계: `seed.iso` 부팅 이미지 준비

`seed.iso` 부팅 이미지에는 네트워크 구성, 호스트 이름, 사용자 데이터와 같이 새 VM을 부팅하는 데 필요한 초기 구성 정보가 포함되어 있습니다.

> **Note**
>
> `seed.iso` 부팅 이미지에는 VM을 부팅하는 데 필요한 구성 정보만 포함합니다. Amazon Linux 2 운영 체제 파일은 포함하지 않습니다.

`seed.iso` 부팅 이미지를 생성하려면 다음과 같은 구성 파일 두 개가 필요합니다.

- `meta-data` - 이 파일에는 VM에 대한 호스트 이름과 정적 네트워크 설정이 포함됩니다.
- `user-data` - 이 파일에서는 사용자 계정을 구성하고 이 계정의 암호, 키 페어 및 액세스 메커니즘을 지정합니다. 기본적으로 Amazon Linux 2 VM 이미지에서는 `ec2-user` 사용자 계정을 생성합니다. `user-data` 구성 파일을 사용하여 기본 사용자 계정의 암호를 설정합니다.

`seed.iso` 부팅 디스크를 생성하려면

1. `seedconfig`라는 새 폴더를 만들고 이 폴더로 이동합니다.
2. `meta-data` 구성 파일을 생성합니다.
   a. `meta-data`라는 이름의 새로운 파일을 만듭니다.
   b. 원하는 편집기를 사용하여 `meta-data` 파일을 열고 다음을 추가합니다.

```
local-hostname: vm_hostname
# eth0 is the default network interface enabled in the image. You can configure
 static network settings with an entry like the following.
network-interfaces: |
  auto eth0
  iface eth0 inet static
  address 192.168.1.10
```

```
  network 192.168.1.0
  netmask 255.255.255.0
  broadcast 192.168.1.255
  gateway 192.168.1.254
```

*vm_hostname*을 선택한 VM 호스트 이름으로 바꾸고 필요에 따라 네트워크 설정을 구성합니다.

c. `meta-data` 구성 파일을 저장하고 닫습니다.

VM 호스트 이름(`amazonlinux.onprem`)을 지정하고, 기본 네트워크 인터페이스(`eth0`)를 구성하며, 필요한 네트워크 디바이스의 정적 IP 주소를 지정하는 `meta-data` 구성 파일의 예는 샘플 Seed.iso 파일을 참조하십시오.

3. `user-data` 구성 파일을 생성합니다.

   a. `user-data`라는 이름의 새로운 파일을 만듭니다.

   b. 원하는 편집기를 사용하여 `user-data` 파일을 열고 다음을 추가합니다.

```
#cloud-config
#vim:syntax=yaml
users:
# A user by the name `ec2-user` is created in the image by default.
  - default
chpasswd:
  list: |
    ec2-user:plain_text_password
# In the above line, do not add any spaces after 'ec2-user:'.
```

*plain_text_password*를 기본 `ec2-user` 사용자 계정을 대해 선택한 암호로 바꿉니다.

   c. (선택 사항) 기본적으로 cloud-init은 VM이 부팅될 때마다 네트워크 설정을 적용합니다. 부팅될 때마다 cloud-init이 네트워크 설정을 적용하지 않고, 첫 번째 부팅 중에 적용된 네트워크 설정을 유지하려면 다음을 추가합니다.

```
# NOTE: Cloud-init applies network settings on every boot by default. To retain
 network settings from first
boot, add following ‘write_files’ section:
write_files:
  - path: /etc/cloud/cloud.cfg.d/80_disable_network_after_firstboot.cfg
    content: |
      # Disable network configuration after first boot
      network:
        config: disabled
```

   d. `user-data` 구성 파일을 저장하고 닫습니다.

추가 사용자 계정을 생성하고 이 계정의 액세스 메커니즘, 암호 및 키 페어를 지정할 수도 있습니다. 지원되는 명령에 대한 자세한 내용은 Modules를 참조하십시오. 추가 사용자를 세 명 생성하고 기본 `ec2-user` 사용자 계정의 사용자 지정 암호를 지정하는 `user-data` 파일의 예는 샘플 Seed.iso 파일을 참조하십시오.

4. `seed.iso` 및 `meta-data` 구성 파일을 사용하여 `user-data` 부팅 이미지를 생성합니다.

   Linux의 경우 genisoimage와 같은 도구를 사용합니다. `seedconfig` 폴더로 이동하여 다음 명령을 실행합니다.

```
$ genisoimage -output seed.iso -volid cidata -joliet -rock user-data meta-data
```

   macOS의 경우 hdiutil과 같은 도구를 사용합니다. `seedconfig` 폴더에서 한 수준 위로 이동하여 다음 명령을 실행합니다.

```
$ hdiutil makehybrid -o seed.iso -hfs -joliet -iso -default-volume-name cidata
 seedconfig/
```

2단계: Amazon Linux 2 VM 이미지 다운로드

Amazon에서는 지원되는 가상화 플랫폼 각각에 대해 서로 다른 Amazon Linux 2 VM 이미지를 제공합니다. 선택한 플랫폼에 대해 다음과 같이 정확한 VM 이미지를 다운로드합니다.

- VMware
- KVM
- Oracle VirtualBox
- Microsoft Hyper-V

3단계: 새 VM 부팅 및 연결

새 VM을 부팅하여 이에 연결하려면 `seed.iso` 부팅 이미지(1단계에서 생성됨)와 Amazon Linux 2 VM 이미지(2단계에서 다운로드함)가 있어야 합니다. 선택한 VM 플랫폼에 따라 단계가 달라집니다.

최초 부팅 시 `seed.iso` 부팅 이미지를 VM에 연결해야 합니다. `seed.iso`는 최초 부팅 시에만 평가됩니다.

VM이 부팅된 후에 `user-data` 구성 파일에 정의된 사용자 계정 중 하나를 사용해 로그인합니다. VMWare를 제외한 다른 가상화 플랫폼의 경우 처음 로그인한 후에 VM에 연결된 `seed.iso` 부팅 이미지를 연결 해제할 수 있습니다.

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 열러 새 커널을 지정합니다. 이 프로세스는 AMI에서 사용하는 가상화 유형에 따라 다릅니다. 자세한 내용은 Linux AMI 가상화 유형 (p. 95) 단원을 참조하십시오.

내용

- HVM AMIs(GRUB) (p. 171)
- AMIs 반가상화(PV-GRUB) (p. 172)

HVM AMIs(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 AMIs에 대한 GRUB 구성

다음은 HVM AMI에 대한 `menu.lst` 구성 파일의 예입니다. 이 예에서는 Amazon Linux 2018.03(이 AMI에 대한 원래 커널) 커널과 Vanilla Linux 4.16.4(https://www.kernel.org/에서 제공되는 Vanilla Linux 커널의 새로운 버전) 커널 중에서 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. 1115) 단원을 참조하십시오. 사용자 지정 커널을 설치하는 경우 아래 예와 같이 `hiddenmenu` 줄을 삭제하고 `serial` 및 `terminal` 줄을 `/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 4.16.4
root (hd0)
kernel /boot/vmlinuz-4.16.4 root=LABEL=/ console=tty1 console=ttyS0
initrd /boot/initrd.img-4.16.4

title Amazon Linux 2018.03 (4.14.26-46.32.amzn1.x86_64)
root (hd0,0)
kernel /boot/vmlinuz-4.14.26-46.32.amzn1.x86_64 root=LABEL=/ console=tty1 console=ttyS0
initrd /boot/initramfs-4.14.26-46.32.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
```

AMIs 반가상화(PV-GRUB)

PV(반가상화) 가상화를 사용하는 Amazon 머신 이미지는 부팅 과정에서 PV-GRUB라는 시스템을 사용합니다. PV-GRUB는 GNU GRUB 0.97의 패치 버전을 실행하는 반가상화 부트로더입니다. 인스턴스를 실행하면 PV-GRUB가 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 시작하여, 인스턴스에서 사용자 지정 커널을 컴파일하고, 반가상화 (PV-GRUB)에 대한 (p. 173)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-grub`으로 시작하는지 확인합니다.

주제

- PV-GRUB의 제한 사항 (p. 173)
- 반가상화 AMIs에 대해 GRUB 구성 (p. 173)
- Amazon PV-GRUB 커널 이미지 ID (p. 174)
- PV-GRUB 업데이트 (p. 176)

PV-GRUB의 제한 사항

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 AMIs에 대해 GRUB 구성

PV-GRUB을 부팅하려면 이미지 내에 GRUB `menu.lst` 파일이 존재해야 합니다. 이 파일의 가장 일반적인 위치는 `/boot/grub/menu.lst`입니다.

다음은 PV-GRUB AKI로 AMI를 부팅하는 것에 대한 `menu.lst` 구성 파일의 예입니다. 이 예에서는 Amazon Linux 2018.03(이 AMI에 대한 원래 커널) 커널과 Vanilla Linux 4.16.4(https://www.kernel.org/에서 제공되는 Vanilla Linux 커널의 새로운 버전) 커널 중에서 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 4.16.4
root (hd0)
kernel /boot/vmlinuz-4.16.4 root=LABEL=/ console=hvc0
initrd /boot/initrd.img-4.16.4

title Amazon Linux 2018.03 (4.14.26-46.32.amzn1.x86_64)
root (hd0)
kernel /boot/vmlinuz-4.14.26-46.32.amzn1.x86_64 root=LABEL=/ console=hvc0
```

```
initrd /boot/initramfs-4.14.26-46.32.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`
- `(hd0,3)/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, 유럽(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, 유럽(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-912fbcfd	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-West)

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

이미지 ID	이미지 이름
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-70cb0e10`입니다.

2. 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. 174)에서 확인 가능), 다음 절차를 사용하여 이를 업데이트해야 합니다.

PV-GRUB 버전 업데이트 방법

인스턴스가 PV-GRUB의 이전 버전을 사용하는 경우, 이를 최신 버전으로 업데이트해야 합니다.

1. Amazon PV-GRUB 커널 이미지 ID (p. 174)에서 리전 및 프로세스 아키텍처에 대한 최신 PV-GRUB AKI를 확인합니다.
2. 인스턴스를 중단합니다. 사용하고 있는 커널 이미지를 수정하려면 인스턴스를 중단할 필요가 있습니다.

```
aws ec2 stop-instances --instance-ids instance_id --region region
```

3. 인스턴스에 대해 사용되는 커널 이미지를 수정합니다.

```
aws ec2 modify-instance-attribute --instance-id instance_id --kernel kernel_id --
region region
```

4. 인스턴스를 재시작합니다.

```
aws ec2 start-instances --instance-ids instance_id --region region
```

Amazon EC2 인스턴스

Amazon EC2를 처음 사용하는 경우 시작하려면 다음 단원을 참조하십시오.

- Amazon EC2란 무엇입니까? (p. 1)
- Amazon EC2로 설정 (p. 22)
- Amazon EC2 Linux 인스턴스 시작하기 (p. 27)
- 인스턴스 수명 주기 (p. 426)

프로덕션 환경을 시작하기 전에 다음 질문에 답해야 합니다.

Q. 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. 178) 단원을 참조하십시오.

Q. 어떤 구매 옵션이 필요에 가장 잘 맞습니까?

Amazon EC2는 온디맨드 인스턴스(기본값), 스팟 인스턴스 및 예약 인스턴스를 지원합니다. 자세한 내용은 인스턴스 구입 옵션 (p. 268) 단원을 참조하십시오.

Q. 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. 93) 단원을 참조하십시오.

Q. 하이브리드 환경에서 EC2 인스턴스 및 머신 플릿을 원격으로 관리할 수 있습니까?

AWS 시스템 관리자를 통해 하이브리드 환경의 Amazon EC2 인스턴스, 온-프레미스 인스턴스 및 다른 클라우드 공급자가 제공하는 VM(가상 머신)을 포함하는 VM의 구성을 안전하게 원격으로 관리할 수 있습니다. 자세한 내용은 AWS 시스템 관리자 사용 설명서를 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 몫의 공유 리소스가 할당됩니다. 더 많은 몫의 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.

내용

- 사용 가능한 인스턴스 유형 (p. 179)
- 하드웨어 사양 (p. 181)
- AMI 가상화 유형 (p. 181)
- Nitro 기반 인스턴스 (p. 181)
- 네트워킹 및 스토리지 기능 (p. 182)
- 인스턴스 제한 (p. 185)
- 범용 인스턴스 (p. 185)
- 컴퓨팅 최적화 인스턴스 (p. 227)
- 메모리 최적화 인스턴스 (p. 232)
- 스토리지 최적화 인스턴스 (p. 242)
- Linux 액셀러레이티드 컴퓨팅 인스턴스 (p. 248)
- 인스턴스 유형 찾기 (p. 261)
- 인스턴스 유형 변경 (p. 262)
- 인스턴스 유형에 대한 권장 사항 가져오기 (p. 265)

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. 리전이나 가용 영역에서 사용할 수 있는 인스턴스 유형을 결정하려면 describe-instance-type-offerings 명령을 사용합니다. 인스턴스 유형의 특성을 설명하려면 describe-instance-types 명령을 사용합니다.

현재 세대 인스턴스 유형에 대한 자세한 내용은 Amazon EC2 인스턴스 유형 정보 페이지를 참조하십시오.

인스턴스 패밀리	현재 세대 인스턴스 유형
범용	a1.medium | a1.large | a1.xlarge | a1.2xlarge | a1.4xlarge | m4.large | m4.xlarge | m4.2xlarge | m4.4xlarge | m4.10xlarge | m4.16xlarge | m5.large | m5.xlarge | m5.2xlarge | m5.4xlarge | m5.8xlarge | m5.12xlarge | m5.16xlarge | m5.24xlarge | m5.metal | m5a.large | m5a.xlarge | m5a.2xlarge | m5a.4xlarge | m5a.8xlarge | m5a.12xlarge | m5a.16xlarge | m5a.24xlarge | m5ad.large | m5ad.xlarge | m5ad.2xlarge | m5ad.4xlarge | m5ad.8xlarge | m5ad.12xlarge | m5ad.16xlarge | m5ad.24xlarge | m5d.large | m5d.xlarge | m5d.2xlarge | m5d.4xlarge | m5d.8xlarge | m5d.12xlarge | m5d.16xlarge | m5d.24xlarge | m5d.metal | m5dn.large | m5dn.xlarge | m5dn.2xlarge | m5dn.4xlarge | m5dn.8xlarge | m5dn.12xlarge | m5dn.16xlarge | m5dn.24xlarge | m5n.large | m5n.xlarge | m5n.2xlarge | m5n.4xlarge | m5n.8xlarge | m5n.12xlarge | m5n.16xlarge | m5n.24xlarge | t2.nano | t2.micro | t2.small | t2.medium | t2.large | t2.xlarge | t2.2xlarge | t3.nano | t3.micro | t3.small | t3.medium | t3.large |

인스턴스 패밀리	현재 세대 인스턴스 유형
	t3.xlarge | t3.2xlarge | t3a.nano | t3a.micro | t3a.small | t3a.medium | t3a.large | t3a.xlarge | t3a.2xlarge
컴퓨팅 최적화	c4.large | c4.xlarge | c4.2xlarge | c4.4xlarge | c4.8xlarge | c5.large | c5.xlarge | c5.2xlarge | c5.4xlarge | c5.9xlarge | c5.12xlarge | c5.18xlarge | c5.24xlarge | c5.metal | c5d.large | c5d.xlarge | c5d.2xlarge | c5d.4xlarge | c5d.9xlarge | c5d.12xlarge | c5d.18xlarge | c5d.24xlarge | c5d.metal | c5n.large | c5n.xlarge | c5n.2xlarge | c5n.4xlarge | c5n.9xlarge | c5n.18xlarge | c5n.metal
메모리 최적화	r4.large | r4.xlarge | r4.2xlarge | r4.4xlarge | r4.8xlarge | r4.16xlarge | r5.large | r5.xlarge | r5.2xlarge | r5.4xlarge | r5.8xlarge | r5.12xlarge | r5.16xlarge | r5.24xlarge | r5.metal | r5a.large | r5a.xlarge | r5a.2xlarge | r5a.4xlarge | r5a.8xlarge | r5a.12xlarge | r5a.16xlarge | r5a.24xlarge | r5ad.large | r5ad.xlarge | r5ad.2xlarge | r5ad.4xlarge | r5ad.12xlarge | r5ad.24xlarge | r5d.large | r5d.xlarge | r5d.2xlarge | r5d.4xlarge | r5d.8xlarge | r5d.12xlarge | r5d.16xlarge | r5d.24xlarge | r5d.metal | r5dn.large | r5dn.xlarge | r5dn.2xlarge | r5dn.4xlarge | r5dn.8xlarge | r5dn.12xlarge | r5dn.16xlarge | r5dn.24xlarge | r5n.large | r5n.xlarge | r5n.2xlarge | r5n.4xlarge | r5n.8xlarge | r5n.12xlarge | r5n.16xlarge | r5n.24xlarge | u-6tb1.metal | u-9tb1.metal | u-12tb1.metal | u-18tb1.metal | u-24tb1.metal | x1.16xlarge | x1.32xlarge | x1e.xlarge | x1e.2xlarge | x1e.4xlarge | x1e.8xlarge | x1e.16xlarge | x1e.32xlarge | z1d.large | z1d.xlarge | z1d.2xlarge | z1d.3xlarge | z1d.6xlarge | z1d.12xlarge | z1d.metal
스토리지 최적화	d2.xlarge | d2.2xlarge | d2.4xlarge | d2.8xlarge | h1.2xlarge | h1.4xlarge | h1.8xlarge | h1.16xlarge | i3.large | i3.xlarge | i3.2xlarge | i3.4xlarge | i3.8xlarge | i3.16xlarge | i3.metal | i3en.large | i3en.xlarge | i3en.2xlarge | i3en.3xlarge | i3en.6xlarge | i3en.12xlarge | i3en.24xlarge | i3en.metal
액셀러레이티드 컴퓨팅	f1.2xlarge | f1.4xlarge | f1.16xlarge | g3s.xlarge | g3.4xlarge | g3.8xlarge | g3.16xlarge | g4dn.xlarge | g4dn.2xlarge | g4dn.4xlarge | g4dn.8xlarge | g4dn.12xlarge | g4dn.16xlarge | p2.xlarge | p2.8xlarge | p2.16xlarge | p3.2xlarge | p3.8xlarge | p3.16xlarge | p3dn.24xlarge | inf1.xlarge | inf1.2xlarge | inf1.6xlarge | inf1.24xlarge

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 지원합니다. 이전 세대 인스턴스를 사용하고 있는 경우 어떠한 인스턴스로 업그레이드하는 것이 적합한지 알아볼 수 있습니다. 자세한 내용은 이전 세대 인스턴스를 참조하십시오.

인스턴스 패밀리	이전 세대 인스턴스 유형
범용	`m1.small` | `m1.medium` | `m1.large` | `m1.xlarge` | `m3.medium` | `m3.large` | `m3.xlarge` | `m3.2xlarge` | `t1.micro`
컴퓨팅 최적화	`c1.medium` | `c1.xlarge` | `cc2.8xlarge` | `c3.large` | `c3.xlarge` | `c3.2xlarge` | `c3.4xlarge` | `c3.8xlarge`
메모리 최적화	`m2.xlarge` | `m2.2xlarge` | `m2.4xlarge` | `cr1.8xlarge` | `r3.large` | `r3.xlarge` | `r3.2xlarge` | `r3.4xlarge` | `r3.8xlarge`
스토리지 최적화	`hs1.8xlarge` | `i2.xlarge` | `i2.2xlarge` | `i2.4xlarge` | `i2.8xlarge`
액셀러레이티드 컴퓨팅	`g2.2xlarge` | `g2.8xlarge`

하드웨어 사양

Amazon EC2 인스턴스 유형별 하드웨어 사양에 대한 자세한 내용은 Amazon EC2 인스턴스 유형을 참조하십시오.

요구 사항에 가장 적합한 인스턴스 유형을 확인하려면 인스턴스를 시작한 후 벤치마크 애플리케이션을 직접 사용해 보는 것이 좋습니다. 과금 기준은 인스턴스 초이므로 여러 인스턴스 유형을 테스트해 본 후에 결정하는 것이 간편하면서도 경제적입니다.

변경이 필요할 경우 결정을 내린 후에도 인스턴스 크기를 조정할 수 있습니다. 자세한 내용은 인스턴스 유형 변경 (p. 262) 단원을 참조하십시오.

> **Note**
>
> Amazon EC2 인스턴스는 일반적으로 인스턴스 유형 제품 페이지에 지정된 대로 64비트 가상 intel 프로세서에서 실행됩니다. Amazon EC2 인스턴스 유형별 하드웨어 사양에 대한 자세한 내용은 Amazon EC2 인스턴스 유형을 참조하십시오. 하지만 64비트 CPU에 대한 업계의 이름 지정 방식 때문에 혼란이 빚어질 수 있습니다. 칩 제조업체 Advanced Micro Devices(AMD)는 최초로 intel 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 intel 하드웨어에서 실행되고 있음에도 불구하고 Ubuntu나 Windows EC2 인스턴스에 대한 내부 시스템 정보는 CPU 아키텍처를 AMD64로 표시하는 이유가 이 때문입니다.

AMI 가상화 유형

인스턴스의 가상화 유형은 인스턴스를 시작할 때 사용한 AMI에 의해 결정됩니다. 현재 세대의 인스턴스 유형은 HVM(하드웨어 가상 머신)만 지원합니다. 이전 세대의 일부 인스턴스 유형은 반가상화(PV)를 지원하고 일부 AWS 리전이 PV 인스턴스를 지원합니다. 자세한 내용은 Linux AMI 가상화 유형 (p. 95) 단원을 참조하십시오.

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.

Nitro 기반 인스턴스

Nitro 시스템은 우수한 성능과 고가용성, 철저한 보안을 가능하게 만드는 AWS 구축 하드웨어 및 소프트웨어 구성 요소의 모음입니다. 또한 Nitro 시스템은 가상화 오버헤드를 없애고 호스트 하드웨어에 대한 모든 액세스 권한이 필요한 워크로드를 지원하는 베어 메탈 기능을 제공합니다.

Nitro 구성 요소

다음 구성 요소는 Nitro 시스템의 일부입니다.

- Nitro 하이퍼바이저 - 메모리 및 CPU 할당을 관리하고, 대부분의 워크로드를 위한 베어 메탈과 구분이 어려울 정도로 뛰어난 성능을 제공하는 경량 하이퍼바이저입니다.
- Nitro 카드
  - 로컬 NVMe 스토리지 볼륨
  - 네트워킹 하드웨어 지원
  - 관리
  - 모니터링
  - 보안
- 마더보드에 통합된 Nitro 보안 칩

인스턴스 유형

다음은 Nitro 시스템을 기반으로 하는 인스턴스입니다.

- A1, C5, C5d, C5n, G4, I3en, Inf1, M5, M5a, M5ad, M5d, M5dn, M5n, `p3dn.24xlarge`, R5, R5a, R5ad, R5d, R5dn, R5n, T3, T3a 및 z1d
- 베어 메탈: `c5.metal`, `c5d.metal`, `c5n.metal`, `i3.metal`, `i3en.metal`, `m5.metal`, `m5d.metal`, `r5.metal`, `r5d.metal`, `u-6tb1.metal`, `u-9tb1.metal`, `u-12tb1.metal`, `u-18tb1.metal`, `u-24tb1.metal`, and `z1d.metal`

리소스

자세한 내용은 다음 비디오를 참조하십시오.

- AWS re:Invent 2017: Amazon EC2 Nitro 시스템 아키텍처
- AWS re:Invent 2017: Amazon EC2 베어 메탈 인스턴스
- Nitro 프로젝트: 차세대 EC2 인프라

네트워킹 및 스토리지 기능

인스턴스 유형을 선택하면 사용할 수 있는 네트워킹 및 스토리지 기능이 결정됩니다. 인스턴스 유형을 설명하려면 describe-instance-types 명령을 사용합니다.

네트워킹 기능

- IPv6는 모든 현재 세대 인스턴스 유형과 C3, R3 및 I2 이전 세대 인스턴스 유형에서 지원됩니다.
- 인스턴스 유형의 네트워킹 및 대역폭 성능을 극대화하려면 다음을 수행해볼 수 있습니다.
  - 클러스터 배치 그룹에 대해 지원되는 인스턴스 유형을 실행하여 HPC(고성능 컴퓨팅) 애플리케이션에 맞게 인스턴스를 최적화합니다. 공통 클러스터 배치 그룹의 인스턴스는 지연 시간이 짧은 고대역폭 네트워킹이 유용할 수 있습니다. 자세한 내용은 배치 그룹 (p. 757) 단원을 참조하십시오.
  - PPS(Packet Per Second) 성능을 크게 높이고 네트워크 지터 및 지연 시간을 낮추려면 지원되는 최신 인스턴스 유형에 대해 향상된 네트워킹을 활성화합니다. 자세한 내용은 Linux에서 향상된 네트워킹 (p. 706) 단원을 참조하십시오.
- 향상된 네트워킹을 지원하는 현재 세대 인스턴스 유형에는 다음 네트워킹 성능 속성이 있습니다.
  - 동일한 리전 내에서 프라이빗 IPv4 또는 IPv6를 통한 트래픽에서는 단일 흐름 트래픽의 경우 5Gbps, 다중 흐름 트래픽의 경우 최대 25Gbps를 지원할 수 있습니다(인스턴스 유형에 따라 다름).

  - 동일한 리전 내에서 퍼블릭 IP 주소 공간이나 VPC 엔드포인트를 통해 Amazon S3 버킷과 주고받는 트래픽은 사용 가능한 인스턴스 집계 대역폭을 전부 사용할 수 있습니다.
- 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. 766) 단원을 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. 일부 인스턴스 유형은 NVMe 인스턴스 스토어 볼륨을 지원합니다. 일부 인스턴스 유형은 NVMe EBS 볼륨을 지원합니다. 자세한 내용은 Linux 인스턴스의 Amazon EBS 및 NVMe (p. 981) 및 NVMe SSD 볼륨 (p. 1036) 단원을 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. 984) 단원을 참조하십시오.

네트워킹 및 스토리지 기능 요약

다음 표에서는 현재 세대 인스턴스 유형에서 지원되는 네트워킹 및 스토리지 기능을 요약합니다.

	EBS 전용	NVMe EBS	인스턴스 스토어	배치 그룹	향상된 네트워킹
A1	예	예	아니요	예	ENA
C4	예	아니요	아니요	예	intel 82599 VF
C5	예	예	아니요	예	ENA
C5d	아니요	예	NVMe *	예	ENA
C5n	예	예	아니요	예	ENA
D2	아니요	아니요	HDD	예	intel 82599 VF
SharePoint용 F1	아니요	아니요	NVMe *	예	ENA
G3	예	아니요	아니요	예	ENA
G4	아니요	예	NVMe *	예	ENA
HS1	아니요	아니요	HDD *	예	ENA
I3	아니요	아니요	NVMe *	예	ENA
I3en	아니요	예	NVMe *	예	ENA
M4	예	아니요	아니요	예	`m4.16xlarge`: ENA 기타 모든 크기: intel 82599 VF
M5	예	예	아니요	예	ENA

	EBS 전용	NVMe EBS	인스턴스 스토어	배치 그룹	향상된 네트워킹
M5a	예	예	아니요	예	ENA
M5ad	아니요	예	NVMe *	예	ENA
M5d	아니요	예	NVMe *	예	ENA
M5dn	아니요	예	NVMe *	예	ENA
M5n	예	예	아니요	예	ENA
P2	예	아니요	아니요	예	ENA
P3	예	아니요	아니요	예	ENA
P3dn	아니요	예	NVMe *	예	ENA
R4	예	아니요	아니요	예	ENA
R5	예	예	아니요	예	ENA
R5a	예	예	아니요	예	ENA
R5ad	아니요	예	NVMe *	예	ENA
R5d	아니요	예	NVMe *	예	ENA
R5dn	아니요	예	NVMe *	예	ENA
R5n	예	예	아니요	예	ENA
T2	예	아니요	아니요	아니요	아니요
T3	예	예	아니요	아니요	ENA
T3a	예	예	아니요	아니요	ENA
u-xtb1.metal	예	예	아니요	아니요	ENA
X1	아니요	아니요	SSD	예	ENA
X1e	아니요	아니요	SSD *	예	ENA
z1d	아니요	예	NVMe *	예	ENA

* 루트 디바이스 볼륨은 Amazon EBS 볼륨이어야 합니다.

다음 표에서는 이전 세대 인스턴스 유형에서 지원되는 네트워킹 및 스토리지 기능을 요약합니다.

	인스턴스 스토어	배치 그룹	향상된 네트워킹
C3	SSD	예	Intel 82599 VF
G2	SSD	예	아니요
I2	SSD	예	intel 82599 VF
M3	SSD	아니요	아니요

	인스턴스 스토어	배치 그룹	향상된 네트워킹
R3	SSD	예	intel 82599 VF

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 수를 참조하십시오.

이러한 제한을 보거나 현재 제한 증가를 요청하는 방법은 Amazon EC2 서비스 제한 (p. 1079) 단원을 참조하십시오.

범용 인스턴스

컴퓨팅, 메모리 및 네트워킹 리소스의 균형을 유지하는 범용 인스턴스는 다양한 워크로드에 사용할 수 있습니다.

A1 인스턴스

A1 인스턴스는 Arm 에코시스템에서 지원하는 스케일 아웃 워크로드에 매우 적합합니다. 이러한 인스턴스는 다음 애플리케이션에 매우 적합합니다.

- 웹 서버
- 컨테이너화된 마이크로서비스
- 캐싱 집합
- 분산된 데이터 스토어
- Arm 명령 세트가 필요한 애플리케이션

자세한 정보는 Amazon EC2 A1 인스턴스를 참조하십시오.

M5, M5a, M5ad, M5d, M5dn 및 M5n 인스턴스

이러한 인스턴스를 사용하면 클라우드에 배포된 광범위한 애플리케이션을 위해 컴퓨팅, 메모리, 네트워킹 리소스의 균형을 유지해 주는 이상적인 클라우드 인프라를 구축할 수 있습니다. M5 인스턴스는 다음 애플리케이션에 적합합니다.

- 웹 및 애플리케이션 서버
- 소규모 및 중간 규모 데이터베이스
- 게임 서버
- 캐싱 집합
- SAP, Microsoft SharePoint, 클러스터 컴퓨팅 및 기타 엔터프라이즈 애플리케이션을 위한 백엔드 서버 운영

`m5.metal` 및 `m5d.metal` 인스턴스에서는 애플리케이션이 프로세서, 메모리 등 호스트 서버의 물리적 리소스에 직접 액세스할 수 있습니다. 이 인스턴스는 다음과 같은 경우에 적합합니다.

- 가상화된 환경에서 사용할 수 없거나 완전히 지원되지 않는 저수준 하드웨어 기능(예: intel VT)에 액세스해야 하는 워크로드
- 라이선스 또는 지원을 위해 가상화되지 않은 환경이 필요한 애플리케이션

자세한 정보는 Amazon EC2 M5 인스턴스를 참조하십시오.

T2, T3 및 T3a 인스턴스

이러한 인스턴스는 기본 수준의 CPU 성능 외에 버스트 기능이 있어 워크로드에 필요한 만큼 성능을 높일 수 있습니다. 무제한 인스턴스는 필요할 때마다 원하는 기간 동안 높은 CPU 성능을 유지할 수 있습니다. 자세한 내용은 성능 순간 확장 가능 인스턴스 (p. 194) 단원을 참조하십시오. 이러한 인스턴스는 다음 애플리케이션에 매우 적합합니다.

- 웹 사이트 및 웹 애플리케이션:
- 코드 리포지토리
- 개발, 빌드, 테스트 및 스테이징 환경
- 마이크로서비스

자세한 내용은 Amazon EC2 T2 인스턴스 및 Amazon EC2 T3 인스턴스를 참조하십시오.

내용

- 하드웨어 사양 (p. 186)
- 인스턴스 성능 (p. 189)
- 네트워크 성능 (p. 189)
- SSD I/O 성능 (p. 191)
- 인스턴스 기능 (p. 192)
- 출시 정보 (p. 193)
- 성능 순간 확장 가능 인스턴스 (p. 194)

하드웨어 사양

다음은 범용 인스턴스용 하드웨어 사양의 요약 설명입니다.

인스턴스 유형	기본 vCPU	메모리(GiB)
`a1.medium`	1	2
`a1.large`	2	4
`a1.xlarge`	4	8
`a1.2xlarge`	8	16
`a1.4xlarge`	16	32
`m4.large`	2	8
`m4.xlarge`	4	16
`m4.2xlarge`	8	32
`m4.4xlarge`	16	64
`m4.10xlarge`	40	160
`m4.16xlarge`	64	256
`m5.large`	2	8
`m5.xlarge`	4	16
`m5.2xlarge`	8	32

인스턴스 유형	기본 vCPU	메모리(GiB)
m5.4xlarge	16	64
m5.8xlarge	32	128
m5.12xlarge	48	192
m5.16xlarge	64	256
m5.24xlarge	96	384
m5.metal	96	384
m5a.large	2	8
m5a.xlarge	4	16
m5a.2xlarge	8	32
m5a.4xlarge	16	64
m5a.8xlarge	32	128
m5a.12xlarge	48	192
m5a.16xlarge	64	256
m5a.24xlarge	96	384
m5ad.large	2	8
m5ad.xlarge	4	16
m5ad.2xlarge	8	32
m5ad.4xlarge	16	64
m5ad.8xlarge	32	128
m5ad.12xlarge	48	192
m5ad.16xlarge	64	256
m5ad.24xlarge	96	384
m5d.large	2	8
m5d.xlarge	4	16
m5d.2xlarge	8	32
m5d.4xlarge	16	64
m5d.8xlarge	32	128
m5d.12xlarge	48	192
m5d.16xlarge	64	256
m5d.24xlarge	96	384
m5d.metal	96	384

인스턴스 유형	기본 vCPU	메모리(GiB)
m5dn.large	2	8
m5dn.xlarge	4	16
m5dn.2xlarge	8	32
m5dn.4xlarge	16	64
m5dn.8xlarge	32	128
m5dn.12xlarge	48	192
m5dn.16xlarge	64	256
m5dn.24xlarge	96	384
m5n.large	2	8
m5n.xlarge	4	16
m5n.2xlarge	8	32
m5n.4xlarge	16	64
m5n.8xlarge	32	128
m5n.12xlarge	48	192
m5n.16xlarge	64	256
m5n.24xlarge	96	384
t2.nano	1	0.5
t2.micro	1	1
t2.small	1	2
t2.medium	2	4
t2.large	2	8
t2.xlarge	4	16
t2.2xlarge	8	32
t3.nano	2	0.5
t3.micro	2	1
t3.small	2	2
t3.medium	2	4
t3.large	2	8
t3.xlarge	4	16
t3.2xlarge	8	32
t3a.nano	2	0.5

인스턴스 유형	기본 vCPU	메모리(GiB)
t3a.micro	2	1
t3a.small	2	2
t3a.medium	2	4
t3a.large	2	8
t3a.xlarge	4	16
t3a.2xlarge	8	32

Amazon EC2 인스턴스 유형별 하드웨어 사양에 대한 자세한 내용은 Amazon EC2 인스턴스 유형을 참조하십시오.

CPU 옵션 지정에 대한 자세한 내용은 CPU 옵션 최적화 (p. 546) 단원을 참조하십시오.

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. 일부 범용 인스턴스는 추가 비용 없이도 EBS에 최적화할 수 있게 기본 설정되어 있습니다. 자세한 내용은 Amazon EBS 최적화 인스턴스 (p. 984) 단원을 참조하십시오.

일부 범용 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. 537) 단원을 참조하십시오.

네트워크 성능

지원되는 인스턴스 유형에서 향상된 네트워킹 기능을 활성화할 수 있습니다. 향상된 네트워킹을 통해 PPS(Packet Per Second) 성능이 크게 높아지고, 네트워크 지터 및 지연 시간이 낮아집니다. 자세한 내용은 Linux에서 향상된 네트워킹 (p. 706) 단원을 참조하십시오.

향상된 네트워킹을 지원하는 데 ENA(Elastic Network Adapter)를 사용하는 인스턴스 유형은 일관되게 낮은 지연 시간과 함께 높은 초당 패킷 성능을 제공합니다. 대부분의 애플리케이션은 항시 높은 수준의 네트워크 성능을 필요로 하지 않지만, 데이터를 주고 받을 때 넓은 대역폭에 액세스할 수 있을 경우 유익할 수 있습니다. ENA를 사용하고 "최대 10Gbps" 또는 "최대 25Gbps"의 네트워크 성능으로 기록된 인스턴스 크기는 네트워크 I/O 크레딧 메커니즘을 사용하여 평균 대역폭 이용률에 따라 네트워크 대역폭을 인스턴스에 할당합니다. 이러한 인스턴스의 네트워크 대역폭이 기준 한도 미만으로 떨어지면 크레딧이 발생하는데, 이 크레딧은 네트워크 데이터를 전송할 때 사용할 수 있습니다.

다음은 확장 네트워킹을 지원하는 범용 인스턴스용 네트워크 성능의 요약 설명입니다.

인스턴스 유형	네트워크 성능	향상된 네트워킹
t2.nano | t2.micro | t2.small | t2.medium | t2.large | t2.xlarge | t2.2xlarge	최대 1Gbps	
t3.nano | t3.micro | t3.small | t3.medium | t3.large | t3.xlarge | t3.2xlarge | t3a.nano | t3a.micro | t3a.small |	최대 5Gbps	ENA (p. 707)

인스턴스 유형	네트워크 성능	향상된 네트워킹
t3a.medium | t3a.large | t3a.xlarge | t3a.2xlarge		
m4.large	보통	intel 82599 VF (p. 719)
m4.xlarge | m4.2xlarge | m4.4xlarge	높음	Intel 82599 VF (p. 719)
a1.medium | a1.large | a1.xlarge | a1.2xlarge | a1.4xlarge | m5.large | m5.xlarge | m5.2xlarge | m5.4xlarge | m5a.large | m5a.xlarge | m5a.2xlarge | m5a.4xlarge | m5a.8xlarge | m5ad.large | m5ad.xlarge | m5ad.2xlarge | m5ad.4xlarge | m5ad.8xlarge | m5d.large | m5d.xlarge | m5d.2xlarge | m5d.4xlarge	최대 10Gbps	ENA (p. 707)
m4.10xlarge	10Gbps	Intel 82599 VF (p. 719)
m5.8xlarge | m5.12xlarge | m5a.12xlarge | m5ad.12xlarge | m5d.8xlarge | m5d.12xlarge	10Gbps	ENA (p. 707)
m5a.16xlarge | m5ad.16xlarge	12Gbps	ENA (p. 707)
m5.16xlarge | m5a.24xlarge | m5ad.24xlarge | m5d.16xlarge	20Gbps	ENA (p. 707)
m5dn.4xlarge 이하 | m5n.4xlarge 이하	최대 25Gbps	ENA (p. 707)
m4.16xlarge | m5.24xlarge | m5.metal | m5d.24xlarge | m5d.metal | m5dn.8xlarge | m5n.8xlarge	25Gbps	ENA (p. 707)
m5dn.12xlarge | m5n.12xlarge	50Gbps	ENA (p. 707)
m5dn.16xlarge | m5n.16xlarge	75Gbps	ENA (p. 707)
m5dn.24xlarge | m5n.24xlarge	100Gbps	ENA (p. 707)

SSD I/O 성능

커널 버전이 4.4 이상인 Linux AMI를 사용하고 인스턴스에서 사용 가능한 모든 SSD 기반의 인스턴스 스토어 볼륨을 활용하는 경우, 다음 표와 같은 IOPS(블록 크기 4,096바이트) 성능을 얻을 수 있습니다(대기열 깊이 포화 상태에서). 그렇지 않으면 IOPS 성능이 더 낮아집니다.

인스턴스 크기	100% 임의 읽기 IOPS	IOPS 쓰기
`m5ad.large` *	30,000개	15,000
`m5ad.xlarge` *	59,000	29,000
`m5ad.2xlarge` *	117,000	57,000
`m5ad.4xlarge` *	234,000	114,000
`m5ad.8xlarge`	466,666	233,333
`m5ad.12xlarge`	700,000	340,000
`m5ad.16xlarge`	933,333	466,666
`m5ad.24xlarge`	1,400,000	680,000
`m5d.large` *	30,000개	15,000
`m5d.xlarge` *	59,000	29,000
`m5d.2xlarge` *	117,000	57,000
`m5d.4xlarge` *	234,000	114,000
`m5d.8xlarge`	466,666	233,333
`m5d.12xlarge`	700,000	340,000
`m5d.16xlarge`	933,333	466,666
`m5d.24xlarge`	1,400,000	680,000
`m5d.metal`	1,400,000	680,000
`m5dn.large` *	30,000개	15,000
`m5dn.xlarge` *	59,000	29,000
`m5dn.2xlarge` *	117,000	57,000
`m5dn.4xlarge` *	234,000	114,000
`m5dn.8xlarge`	466,666	233,333
`m5dn.12xlarge`	700,000	340,000
`m5dn.16xlarge`	933,333	466,666
`m5dn.24xlarge`	1,400,000	680,000

* 이러한 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. 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 않음을 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. 1037) 단원을 참조하십시오.

인스턴스 기능

다음은 범용 인스턴스를 위한 기능의 요약 설명입니다.

	EBS 전용	NVMe EBS	인스턴스 스토어	배치 그룹
A1	예	예	아니요	예
M4	예	아니요	아니요	예
M5	예	예	아니요	예
M5a	예	예	아니요	예
M5ad	아니요	예	NVMe *	예
M5d	아니요	예	NVMe *	예
M5dn	아니요	예	NVMe *	예
M5n	예	예	아니요	예
T2	예	아니요	아니요	아니요
T3	예	예	아니요	아니요
T3a	예	예	아니요	아니요

* 루트 디바이스 볼륨은 Amazon EBS 볼륨이어야 합니다.

자세한 내용은 다음 단원을 참조하십시오.

- Linux 인스턴스의 Amazon EBS 및 NVMe (p. 981)
- Amazon EC2 인스턴스 스토어 (p. 1027)
- 배치 그룹 (p. 757)

출시 정보

- M5, M5d 및 T3 인스턴스는 1세대(Skylake-SP) 또는 2세대(Cascade Lake)의 3.1 GHz Intel Xeon Platinum 8000 시리즈 프로세서를 사용합니다.
- M5a, M5ad 및 T3a 인스턴스는 2.5GHz AMD EPYC 7000 시리즈 프로세서를 사용하는 것이 특징입니다.
- A1 인스턴스는 64비트 Arm 아키텍처 기반 2.3GHz AWS Graviton 프로세서를 사용하는 것이 특징입니다.
- M4, M5, M5a, M5ad, M5d, `t2.large` 이상, `t3.large` 이상, `t3a.large` 및 그 이상 인스턴스 유형에는 64비트 HVM AMIs가 필요합니다. 이들은 고용량 메모리를 보유하는데, 이 용량을 활용하기 위해서는 64비트 운영 체제가 필요합니다. HVM AMI는 고용량 메모리 인스턴스 유형의 반가상화(PV) AMI보다 우수한 성능을 제공합니다. 또한 확장 네트워킹을 활용하려면 HVM AMI를 사용해야 합니다.
- A1 인스턴스의 요구 사항은 다음과 같습니다.
  - 64비트 Arm 아키텍처용 AMI를 사용합니다.
  - NVMe 드라이버 (p. 981)가 설치되어 있어야 합니다.
  - ENA(Elastic Network Adapter) (p. 707) 드라이버가 설치되어 있어야 합니다.
  - ACPI 테이블을 사용하여 UEFI를 통해 부팅을 지원하고 PCI 디바이스의 ACPI 핫플러그를 지원합니다.

  다음 AMI는 아래 요구 사항을 충족해야 합니다.
  - Amazon Linux 2(64비트 Arm)
  - Ubuntu 16.04 이상(64비트 Arm)
  - Red Hat Enterprise Linux 7.6 이상(64비트 Arm)
  - SUSE Linux Enterprise Server 15 이상(64비트 Arm)
- M5, M5a, M5ad, M5d, M5dn, M5n, T3 및 T3a 인스턴스의 요구 사항은 다음과 같습니다.
  - NVMe 드라이버 (p. 981)가 설치되어 있어야 합니다.
  - ENA(Elastic Network Adapter) (p. 707) 드라이버가 설치되어 있어야 합니다.

  다음 Linux AMI는 이러한 요구 사항을 충족합니다.
  - Amazon Linux 2
  - Amazon Linux AMI 2018.03
  - Ubuntu 14.04(`linux-aws` 커널) 이상
  - Red Hat Enterprise Linux 7.4 이상
  - SUSE Linux Enterprise Server 12 SP2 이상
  - CentOS 7.4.1708 이상
  - FreeBSD 11.1 이상
  - Debian GNU/Linux 9 이상
- A1, M5, M5a, M5ad, M5d, M5dn, M5n, T3 및 T3a 인스턴스는 네트워크 인터페이스, EBS 볼륨 및 NVMe 인스턴스 스토어 볼륨을 포함해 최대 28개의 연결을 지원합니다. 모든 인스턴스에는 최소 한 개의 네트워크 인터페이스 연결이 있습니다. 예를 들어, EBS 전용 인스턴스에 그 밖의 네트워크 인터페이스 연결이 없는 경우, EBS 볼륨 27개를 연결할 수 있습니다.
- 베어 메탈 인스턴스를 시작하면 기본 서버가 부팅되는데, 이때 모든 하드웨어 및 펌웨어 구성 요소를 확인합니다. 즉, 인스턴스가 실행 상태가 되어 네트워크를 통해 사용할 수 있게 될 때까지 20분이 걸릴 수 있습니다.
- 베어 메탈 인스턴스에서 EBS 볼륨 또는 보조 네트워크 인터페이스를 연결 또는 분리하려면 PCIe 기본 핫플러그 지원이 필요합니다. Amazon Linux 2 및 최신 Amazon Linux AMI 버전에서는 PCIe 기본 핫플러그를 지원하지만 이전 버전에서는 지원하지 않습니다. 다음 Linux 커널 구성 옵션을 활성화해야 합니다.

  ```
  CONFIG_HOTPLUG_PCI_PCIE=y
  CONFIG_PCIEASPM=y
  ```

- 베어 메탈 인스턴스에서는 I/O 포트 기반 직렬 디바이스가 아닌 PCI 기반 직렬 디바이스를 사용합니다. 업스트림 Linux 커널 및 최신 Amazon Linux AMI에서는 이 디바이스를 지원합니다. 베어 메탈 인스턴스도 시

스템에서 PCI 기반 직렬 디바이스를 자동으로 사용할 수 있게 해주는 ACPI SPCR 테이블을 제공합니다. 최신 Windows AMI에서는 PCI 기반 직렬 디바이스를 자동으로 사용합니다.

- API 요청을 통해 원활한 종료를 지원하려면 A1, M5, M5a, M5ad, M5d, M5dn, M5n, T3 및 T3a 인스턴스에 system-logind 또는 acpid가 설치되어 있어야 합니다.
- 한 리전에서 시작할 수 있는 총 인스턴스 수에는 제한이 있으며, 일부 인스턴스 유형에는 또 다른 제한이 있습니다. 자세한 정보는 Amazon EC2에서 실행 가능한 인스턴스 수 단원을 참조하십시오. 한도 증가를 요청하려면 Amazon EC2 인스턴스 요청 양식을 사용하십시오.

성능 순간 확장 가능 인스턴스

성능 순간 확장 가능 인스턴스인 T3, T3a, 및 T2 인스턴스는 기본 수준의 CPU 성능과 함께 워크로드에서 필요한 만큼 성능을 높이는 버스트 기능을 제공하도록 설계되었습니다. 성능 순간 확장 가능 인스턴스는 광범위한 범용 애플리케이션에 적합합니다. 해당되는 예로 마이크로서비스, 지연 시간이 짧은 대화형 애플리케이션, 중소 규모 데이터베이스, 가상 데스크톱, 개발, 빌드, 스테이지 환경, 코드 리포지토리, 제품 프로토타입을 들 수 있습니다.

성능 순간 확장 가능 인스턴스는 CPU 사용량에 대해 크레딧을 사용하는 유일한 인스턴스 유형입니다. 인스턴스 요금에 대한 정보와 기타 자세한 하드웨어 정보는 Amazon EC2 요금 및 Amazon EC2 인스턴스 유형을 참조하십시오.

계정이 12개월이 아직 안 된 경우 특정 사용 한도 내에서 무료로 `t2.micro` 인스턴스를 사용할 수 있습니다. 자세한 정보는 AWS 프리 티어 단원을 참조하십시오.

내용

- 성능 순간 확장 가능 인스턴스에 대한 요구 사항 (p. 194)
- 모범 사례 (p. 194)
- 성능 순간 확장 가능 인스턴스에 대한 CPU 크레딧 및 기준 성능 (p. 195)
- 성능 순간 확장 가능 인스턴스의 무제한 모드 (p. 197)
- 버스트 가능한 성능 인스턴스의 표준 모드 (p. 206)
- 성능 순간 확장 가능 인스턴스의 작업 (p. 220)
- CPU 크레딧 모니터링 (p. 224)

성능 순간 확장 가능 인스턴스에 대한 요구 사항

다음은 이러한 인스턴스에 대한 기본 요구 사항입니다.

- 이러한 인스턴스는 온디맨드 인스턴스, 예약 인스턴스, 전용 인스턴스 및 스팟 인스턴스로 사용할 수 있지만 예약 인스턴스로는 사용할 수 없습니다. 또한 전용 호스트에서도 지원되지 않습니다. 자세한 내용은 인스턴스 구입 옵션 (p. 268) 단원을 참조하십시오.
- 선택한 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 워크로드의 메모리 및 CPU 요구 사항이 증가함에 따라 더 큰 규모의 동일한 인스턴스 유형 또는 다른 인스턴스 유형에 맞게 조정할 수 있습니다.
- 추가 요구 사항은 범용 인스턴스 출시 정보 (p. 193)를 참조하십시오.

모범 사례

다음 모범 사례를 따르면 성능 순간 확장 가능 인스턴스의 이점을 최대한 활용할 수 있습니다.

- 권장 AMI 사용 – 필수 드라이버를 제공하는 AMI를 사용합니다. 자세한 내용은 출시 정보 (p. 193) 단원을 참조하십시오.

- 인스턴스 복구 켜기 – EC2 인스턴스를 모니터링하고 어떤 이유로든 손상된 경우 이를 자동으로 복구하는 CloudWatch 경보를 생성합니다. 자세한 내용은 Amazon CloudWatch 경보에 복구 작업 추가 (p. 640) 단원을 참조하십시오.

성능 순간 확장 가능 인스턴스에 대한 CPU 크레딧 및 기준 성능

기존 Amazon EC2 인스턴스 유형은 고정된 성능을 제공하는 반면, 성능 순간 확장 가능 인스턴스는 기본 수준의 CPU 성능을 발휘하면서 기본 수준 이상으로 버스트하는 기능을 제공합니다. 기본 성능과 버스트 기능은 CPU 크레딧에 의해 좌우됩니다.

CPU 크레딧은 1분 동안 100% 사용률로 실행되는 전체 CPU 코어의 성능을 제공합니다. vCPU, 사용률 및 시간의 여러 가지 조합이 CPU 크레딧 하나에 해당할 수도 있습니다. 예를 들어, CPU 크레딧 하나는 2분 동안 50%의 사용률로 실행되는 vCPU 하나 또는 2분 동안 25%의 사용률로 실행되는 vCPU 2개에 해당합니다.

목차

- CPU 크레딧 획득 (p. 195)
- CPU 크레딧 획득률 (p. 196)
- CPU 크레딧 누적 한도 (p. 196)
- 획득한 CPU 크레딧의 수명 (p. 197)
- 기준 성능 (p. 197)

CPU 크레딧 획득

각 성능 순간 확장 가능 인스턴스는 인스턴스 크기에 따라 특정 비율의 시간당 CPU 크레딧을 지속적으로 (밀리초 수준의 시간 정밀도로) 획득합니다. 크레딧이 누적되는지 아니면 소비되는지를 결정하는 산정 프로세스도 밀리초 수준의 시간 정밀도로 수행되므로 CPU 크레딧 과소비를 염려할 필요는 없습니다. 즉, 짧은 CPU 버스트는 약간의 CPU 크레딧만을 소비합니다.

성능 순간 확장 가능 인스턴스에서 기준 성능에 필요한 것보다 더 적은 CPU 리소스를 사용하는 경우(예: 유휴 상태) 사용하지 않은 CPU 크레딧은 CPU 크레딧 밸런스에 누적됩니다. 성능 순간 확장 가능 인스턴스가 기준 성능 수준 이상으로 버스트해야 할 경우 축적된 크레딧을 소모합니다. 성능 순간 확장 가능 인스턴스가 획득한 크레딧이 많을수록 추가 성능이 필요할 때 기준 성능 수준을 초과하여 버스트할 수 있는 시간이 증가합니다.

다음 표에는 성능 순간 확장 가능 인스턴스 유형, 시간당 CPU 크레딧 획득률, 인스턴스가 획득할 수 있는 최대 CPU 크레딧 수, 인스턴스당 vCPU 수, 전체 코어 성능의 백분율로 나타낸 기준 성능 수준(단일 vCPU 사용 시) 등이 나와 있습니다.

인스턴스 유형	시간당 지급되는 CPU 크레딧	누적 가능한 최대 지급된 크레딧*	vCPUs	vCPU당 기준 성능
T2				
`t2.nano`	3	72	1	5%
`t2.micro`	6	144	1	10%
`t2.small`	12	288	1	20%
`t2.medium`	24	576	2	20%**
`t2.large`	36	864	2	30%**
`t2.xlarge`	54	1296	4	22.5%**
`t2.2xlarge`	81.6	1958.4	8	17%**

인스턴스 유형	시간당 지급되는 CPU 크레딧	누적 가능한 최대 지급된 크레딧*	vCPUs	vCPU당 기준 성능
T3				
`t3.nano`	6	144	2	5%**
`t3.micro`	12	288	2	10%**
`t3.small`	24	576	2	20%**
`t3.medium`	24	576	2	20%**
`t3.large`	36	864	2	30%**
`t3.xlarge`	96	2304	4	40%**
`t3.2xlarge`	192	4608	8	40%**
T3a				
`t3a.nano`	6	144	2	5%**
`t3a.micro`	12	288	2	10%**
`t3a.small`	24	576	2	20%**
`t3a.medium`	24	576	2	20%**
`t3a.large`	36	864	2	30%**
`t3a.xlarge`	96	2304	4	40%**
`t3a.2xlarge`	192	4608	8	40%**

* 누적될 수 있는 크레딧은 수는 24시간 동안 획득할 수 있는 크레딧의 수와 동일합니다.

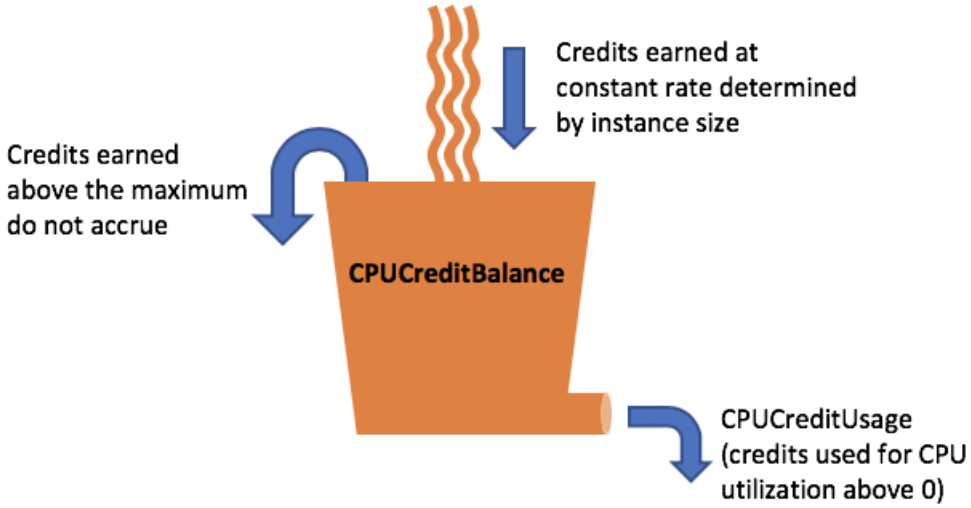
** 테이블의 기준 성능은 vCPU 단위로 계산됩니다. vCPU가 두 개 이상 있는 인스턴스 크기의 경우 인스턴스의 기본 CPU 사용률을 계산하려면 vCPU 백분율에 vCPU 개수를 곱합니다. 예를 들어 `t3.large` 인스턴스에는 vCPU가 두 개 있어 인스턴스에 60%의 기본 CPU 사용률을 제공합니다(vCPU 2개 x vCPU 1개의 30% 기준 성능). CloudWatch에서 CPU 사용률은 vCPU 기준으로 표시됩니다. 따라서 기준 성능으로 작동하는 `t3.large` 인스턴스에 대한 CPU 사용률은 CloudWatch CPU 지표의 30%로 표시됩니다.

CPU 크레딧 획득률

시간당 획득하는 CPU 크레딧의 수는 인스턴스 크기에 의해 결정됩니다. 예를 들어 `t3.nano`는 시간당 6개의 크레딧을 획득하는 반면, `t3.small`은 시간당 24개의 크레딧을 획득합니다. 이전 표에는 모든 인스턴스에 대한 크레딧 획득률이 나와 있습니다.

CPU 크레딧 누적 한도

실행 중인 인스턴스에서 획득한 크레딧은 만료되지 않습니다. 하지만 인스턴스가 누적할 수 있는 획득 크레딧 수에는 한도가 있습니다. 한도는 CPU 크레딧 밸런스 한도에 따라 결정됩니다. 한도에 도달한 후에 새로 획득하는 크레딧은 다음 이미지와 같이 모두 삭제됩니다. 최대 버킷은 CPU 크레딧 밸런스 한도를 나타내고, 스필오버는 한도를 초과하여 새로 획득한 크레딧을 나타냅니다.



CPU 크레딧 밸런스 한도는 각 인스턴스 크기에 따라 다릅니다. 예를 들어 `t3.micro` 인스턴스는 CPU 크레딧 밸런스에서 최대 288의 획득한 CPU 크레딧을 누적할 수 있습니다. 이전 표에는 각 인스턴스에서 누적할 수 있는 최대 획득 크레딧 수가 나와 있습니다.

> **Note**
>
> T2 스탠다드 인스턴스에서도 시작 크레딧을 획득합니다. 시작 크레딧은 CPU 크레딧 밸런스 한도에 포함되지 않습니다. T2 인스턴스가 시작 크레딧을 사용하지 않고 획득 크레딧을 누적하면서 24시간 동안 유휴 상태를 유지한 경우 CPU 크레딧 밸런스는 한도 이상으로 표시됩니다. 자세한 내용은 시작 크레딧 (p. 207) 단원을 참조하십시오.
> T3 및 T3a 인스턴스에서는 시작 크레딧을 획득하지 않습니다. 이러한 인스턴스는 `unlimited`로 시작하도록 기본 설정되어 있으므로 시작 크레딧 없이도 시작하자마자 즉시 버스트할 수 있습니다.

획득한 CPU 크레딧의 수명

실행 중인 인스턴스의 CPU 크레딧은 만료 기간이 없습니다.

T3 및 T3a의 경우 인스턴스가 중단된 후 CPU 크레딧 밸런스가 7일 동안 지속하다가 7일이 지나면 크레딧이 상실됩니다. 7일 이내에 인스턴스를 시작하면 크레딧이 상실되지 않습니다.

T2의 경우 CPU 크레딧 밸런스는 인스턴스 중지와 시작 사이의 기간 동안 지속하지 않습니다. T2 인스턴스를 중지하면 인스턴스는 누적된 크레딧을 모두 상실합니다.

자세한 내용은 CloudWatch 지표 (p. 224) 표에서 `CPUCreditBalance` 항목을 참조하십시오.

기준 성능

인스턴스가 시간당 획득한 크레딧 수는 CPU 사용률의 백분율로 표시가 가능합니다. 이를 기준 성능이라 하며, 기준이라고도 합니다. 예를 들어 vCPU가 2개인 `t3.nano` 인스턴스는 시간당 6개의 크레딧을 획득하므로 vCPU당 기준 성능이 5%(3/60분)입니다. vCPU가 4개인 `t3.xlarge` 인스턴스는 시간당 96개의 크레딧을 획득하므로 vCPU당 기준 성능이 40%(24/60분)입니다.

성능 순간 확장 가능 인스턴스의 무제한 모드

`unlimited`로 구성된 성능 순간 확장 가능 인스턴스는 필요한 경우 언제든지 원하는 기간 동안 높은 CPU 성능을 유지할 수 있습니다. 24시간 동안 또는 인스턴스 수명(더 짧음) 동안 인스턴스의 평균 CPU 사용률이 기준 이하인 경우에 모든 CPU 사용량 급증에 대해 시간당 CPU 인스턴스 요금이 적용됩니다.

대부분의 범용 워크로드에서 `unlimited`로 구성된 인스턴스는 추가 요금 없이 충분한 성능을 제공합니다. 인스턴스 실행에 장기간 높은 CPU 사용률이 필요한 경우, vCPU-시간당 추가 고정 요금으로 인스턴스를 실

행할 수 있습니다. 인스턴스 요금에 대한 자세한 내용은 Amazon EC2 요금 및 Amazon EC2 온디맨드 요금의 무제한 요금 단원을 참조하십시오.

> **Important**
>
> AWS 프리 티어 혜택에 따라 `t2.micro` 인스턴스를 사용하고 이 인스턴스를 `unlimited`로 구성하는 경우 24시간 롤링 기간 동안 평균 사용률이 인스턴스의 기준 (p. 197)을 초과하면 요금이 적용될 수 있습니다.
> T3 인스턴스는 `unlimited`로 시작하도록 기본 설정되어 있습니다. T3 스팟 인스턴스를 `unlimited`로 시작하고 CPU 크레딧 발생에 대한 유휴 시간 없이 즉시 짧은 기간 동안 사용하려는 경우 잉여 크레딧에 대한 요금이 발생합니다. 24시간 기간 동안 평균 CPU 사용량이 기준을 초과하면 잉여 크레딧에 대한 요금도 발생합니다. 더 높은 비용을 지불하지 않으려면 표준 (p. 206) 모드에서 T3 스팟 인스턴스를 시작하는 것이 좋습니다. 자세한 내용은 잉여 크레딧으로 요금 발생 가능 (p. 201) 및 T3 스팟 인스턴스 (p. 381) 단원을 참조하십시오.

목차

- 무제한 모드 개념 (p. 198)
  - 무제한 성능 순간 확장 가능 인스턴스의 작동 방식 (p. 198)
  - 무제한 모드 대 고정 CPU 사용 시기 (p. 199)
  - 잉여 크레딧으로 요금 발생 가능 (p. 201)
  - T2 무제한에는 시작 크레딧이 없음 (p. 202)
  - 무제한 모드 활성화 (p. 202)
  - 무제한과 스탠다드에서 전환 시 크레딧에 발생하는 현상 (p. 202)
  - 크레딧 사용량 모니터링 (p. 202)
- 예: 무제한 모드 (p. 203)
  - 예 1: T3 무제한에서 크레딧을 사용하는 것에 관해 설명 (p. 203)
  - 예 2: T2 무제한에서 크레딧을 사용하는 것에 관해 설명 (p. 204)

무제한 모드 개념

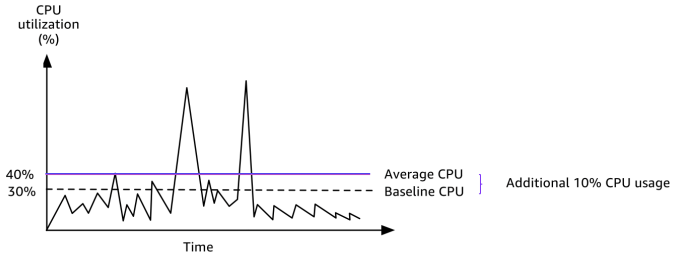
`unlimited` 모드는 성능 순간 확장 가능 인스턴스에 사용할 수 있는 크레딧 구성 옵션입니다. 이 모드는 실행 중인 또는 중지된 인스턴스에 대해 언제든지 활성화 또는 비활성화할 수 있습니다. 성능 순간 확장 가능 인스턴스 패밀리마다 AWS 리전별 계정 수준에서 `unlimited`를 기본 크레딧 옵션으로 설정하여 계정에 속한 새 성능 순간 확장 가능 인스턴스가 모두 기본 크레딧 옵션을 사용하여 시작되게 할 수 있습니다.

> **Note**
>
> T3 및 T3a 인스턴스는 `unlimited`로 시작하도록 기본 설정되어 있습니다. T2 인스턴스는 `standard`로 시작하도록 기본 설정되어 있습니다. AWS 리전별로 계정 수준에서 기본값을 변경할 수 있습니다. 자세한 내용은 계정의 기본 크레딧 사양 설정 (p. 223) 단원을 참조하십시오.

무제한 성능 순간 확장 가능 인스턴스의 작동 방식

`unlimited`로 구성된 성능 순간 확장 가능 인스턴스의 CPU 크레딧 밸런스가 감소하면 잉여 크레딧을 사용하여 기준 성능 이상으로 버스트할 수 있습니다. CPU 사용률이 기준 미만으로 떨어지면 획득한 CPU 크레딧을 사용하여 이전에 소비한 잉여 크레딧을 청산할 수 있습니다. CPU 크레딧을 획득하고 잉여 크레딧을 청산하는 기능을 통해 Amazon EC2은 24시간 동안 인스턴스의 CPU 사용률을 평균 수준으로 유지할 수 있습니다. 24시간 동안의 평균 CPU 사용량이 기준을 초과하는 경우 인스턴스에 추가 사용량에 대해 vCPU 시간당 고정 추가 요금이 청구됩니다.

다음 그래프는 `t3.large`의 CPU 사용량을 보여줍니다. `t3.large`에 대한 기본 CPU 사용률은 30%입니다. 인스턴스가 24시간 동안 평균 30% CPU 사용률로 실행되는 경우 이미 인스턴스 시간당 가격으로 비용이 처리되었으므로 추가 비용이 발생하지 않습니다. 그러나 그래프에 표시된 것처럼 24시간 동안의 평균 40%의 CPU 사용률로 실행되는 경우 이 인스턴스는 추가 10% CPU 사용량에 대해 vCPU 시간당 추가 고정 요금이 청구됩니다.



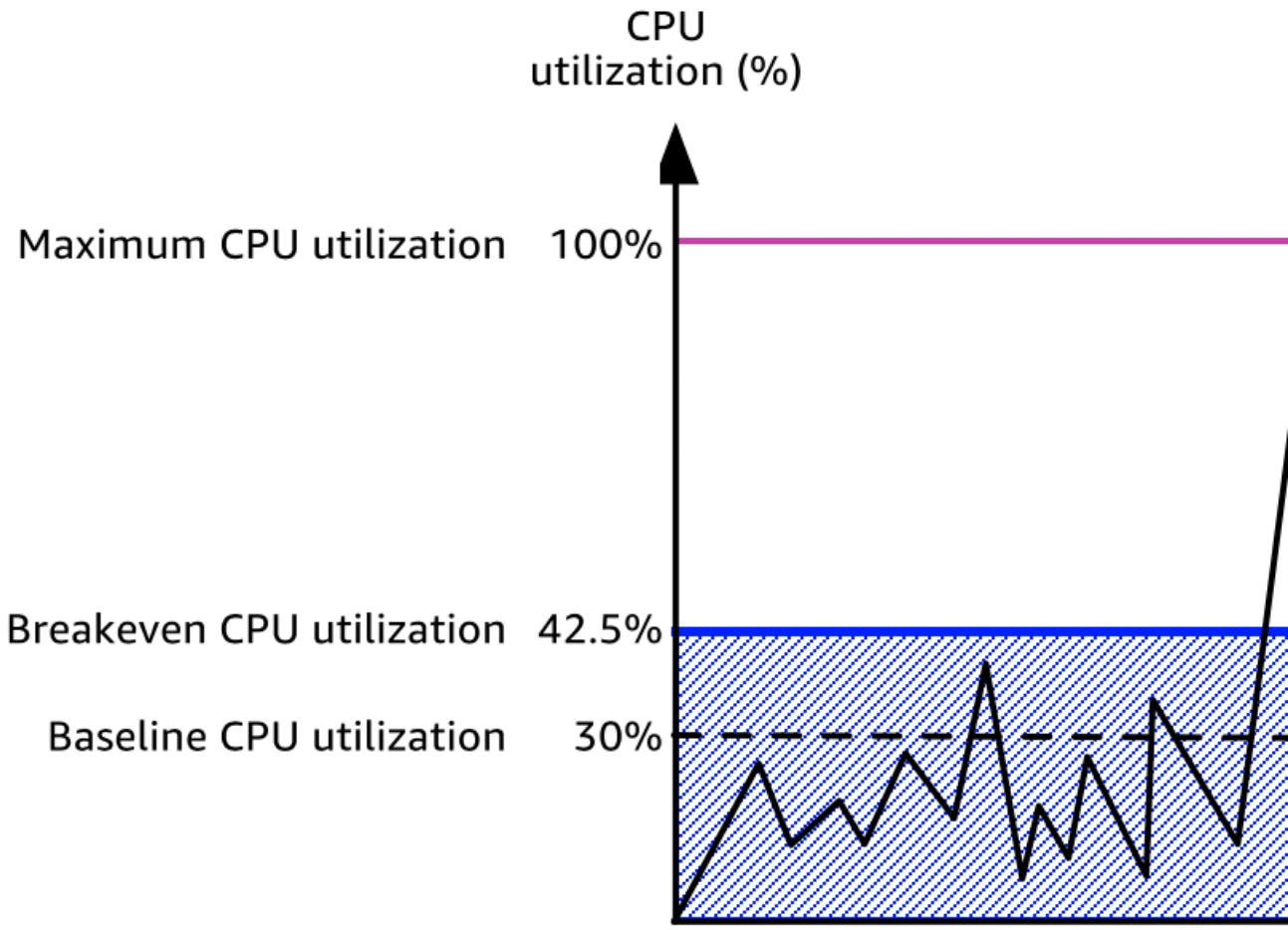
각 인스턴스 유형별 vCPU 당 기준 성능 및 각 인스턴스 유형에서 얻은 크레딧 수에 대한 자세한 내용은 크레딧 표 (p. 195)를 참조하십시오.

무제한 모드 대 고정 CPU 사용 시기

T3와 같은 `unlimited` 모드 또는 M5와 같은 고정 성능 인스턴스에서 버스트 가능한 성능 인스턴스를 사용해야 하는지 결정할 때는 손익분기 CPU 사용량을 결정해야 합니다. 버스트 가능한 성능 인스턴스에 대한 손익분기 CPU 사용량은 버스트 가능한 성능 인스턴스가 고정 성능 인스턴스와 동일한 비용을 부담합니다. 손익분기 CPU 사용량은 다음을 결정하는 데 도움이 됩니다.

- 24시간 동안의 평균 CPU 사용량이 손익분기 CPU 사용량 또는 그 이하인 경우 `unlimited` 모드에서 버스트 가능한 성능 인스턴스를 사용하면 버스트 가능 성능 인스턴스의 저렴한 가격으로 혜택을 누릴 수 있으며 동시에 고정 성능 인스턴스와 동일한 성능을 얻을 수 있습니다.
- 24시간 동안의 평균 CPU 사용량이 손익분기 CPU 사용량보다 많으면 버스트 가능한 성능 인스턴스의 비용은 동등한 크기의 고정 성능 인스턴스보다 증가합니다. T3 인스턴스가 100% CPU에서 연속적으로 버스트하면 동등한 크기의 M5 인스턴스 가격의 약 1.5배를 지불하게 됩니다.

다음 그래프에서는 `t3.large`이 `m5.large`와 동일한 비용의 손익분기 CPU 사용량을 보여줍니다. `t3.large`에 대한 손익분기 CPU 사용량은 42.5%입니다. 평균 CPU 사용량이 42.5%인 경우 `t3.large`을 실행하는 비용은 `m5.large`와 동일하며 평균 CPU 사용량이 42.5%를 초과하면 비용이 더 많이 듭니다. 작업 부하가 42.5% 미만의 평균 CPU 사용량이 필요한 경우 `m5.large`와 동일한 성능을 얻는 동안 `t3.large`의 저렴한 가격으로 혜택을 볼 수 있습니다.



다음 표는 손익분기 CPU 사용량 임계값을 계산하여 `unlimited` 모드 또는 고정 성능 인스턴스에서 버스트 가능한 성능 인스턴스를 사용하는 것이 더 경제적인 시기가 언제인지를 결정할 수 있는 방법을 보여줍니다. 테이블의 열은 A에서 K로 표시됩니다.

인스턴스 유형	vCPUs	T3 가격*/시간	M5 가격*/시간	가격 차이	vCPU(%)당 T3 기준 성능	잉여 크레딧에 대한 vCPU 시간당 요금	vCPU 분당 요금	vCPU 당 사용 가능한 추가 버스트 시간(분)	사용 가능한 추가 CPU%	손익 분기 CPU%
A	B	C	D	E = D - C	F	G	H = G/60	I = E/H	J = (I/60)/B	K = F + J
`t3.large`	2	$0.0835 USD	$0.096 USD	$0.0125 USD	30%	0.05 USD	$0.000833 USD	15	12.5%	42.5%

* 가격은 us-east-1 및 Linux OS를 기준으로 합니다.

이 테이블에서는 다음 정보를 제공합니다.

- A열은 인스턴스 유형인 `t3.large`을 표시합니다.
- B열은 `t3.large`에 대한 vCPU 수를 나타냅니다.
- C열은 시간당 `t3.large`의 가격을 보여줍니다.
- D열은 시간당 `m5.large`의 가격을 보여줍니다.
- D열은 `t3.large`과 `m5.large` 사이의 가격 차이를 보여줍니다.
- F열은 30%인 `t3.large`의 vCPU당 기준 성능을 보여줍니다. 기준선에서 인스턴스의 시간당 비용은 CPU 사용량 비용을 포함합니다.
- G열은 획득된 크레딧이 소진된 후 100% CPU에서 버스트되는 경우 인스턴스에 청구되는 vCPU 시간당 고정 추가 요금을 보여줍니다.
- H열은 획득된 크레딧이 소진된 후 100% CPU에서 버스트되는 경우 인스턴스에 청구되는 vCPU 시간(분)당 고정 추가 요금을 보여줍니다.
- I열은 `t3.large`이 시간당 100% CPU에서 버스트 가능하고 `m5.large`와 같은 시간당 가격을 지불하는 추가 시간(분)을 보여줍니다.
- J열은 `m5.large`로 동일한 가격을 지불하면서 인스턴스가 버스트 가능한 기준선에 대한 추가 CPU 사용량(%)을 보여줍니다.
- K열은 `t3.large`이 `m5.large`보다 많은 비용을 들이지 않고 버스트 가능한 손익분기 CPU 사용량(%)을 보여줍니다. `t3.large` 비용 및 그 어떤 비용도 `m5.large`보다 많습니다.

다음 테이블은 비슷한 크기의 M5 인스턴스 유형과 비교한 T3 인스턴스 유형의 손익분기 CPU 사용량(%)을 보여줍니다.

T3 인스턴스 유형	M5와 비교한 T3에 대한 손익분기 CPU 사용량(%)
`t3.large`	42.5%
`t3.xlarge`	52.5%
`t3.2xlarge`	52.5%

잉여 크레딧으로 요금 발생 가능

인스턴스의 평균 CPU 사용률이 기준 이하인 경우에는 인스턴스로 인해 추가 요금이 발생하지 않습니다. 인스턴스는 24시간 동안 최대 크레딧 수 (p. 195)를 획득하기 때문에(예를 들면 `t3.micro` 인스턴스는 24시간 동안 최대 288개의 크레딧 획득이 가능) 요금을 부과하지 않고 이 최대 값까지 잉여 크레딧을 소비할 수 있습니다.

그러나 CPU 사용률이 기준 이상으로 유지되는 경우 인스턴스는 소비한 잉여 크레딧을 청산하기에 충분한 수준으로 크레딧을 획득할 수 없습니다. 청산된 잉여 크레딧은 vCPU-시간당 추가 고정 요금으로 부과됩니다.

이전에 소비된 잉여 크레딧은 다음이 발생할 때 요금이 부과됩니다.

- 소비한 잉여 크레딧이 인스턴스가 24시간 동안 획득할 수 있는 최대 크레딧 수 (p. 195)를 초과하는 경우. 해당 시간이 끝날 때 최대 값 이상으로 소비한 잉여 크레딧에 요금이 부과됩니다.
- 인스턴스가 중지 또는 종료된 경우.
- 인스턴스가 `unlimited`에서 `standard`로 전환됩니다.

소비한 잉여 크레딧은 CloudWatch 지표 `CPUSurplusCreditBalance`에 의해 추적이 가능합니다. 요금이 부과된 잉여 크레딧은 CloudWatch 지표 `CPUSurplusCreditsCharged`에 의해 추적이 가능합니다. 자세한 내용은 성능 순간 확장 가능 인스턴스에 대한 추가 CloudWatch 측정치 (p. 224) 단원을 참조하십시오.

T2 무제한에는 시작 크레딧이 없음

T2 스탠다드 인스턴스에서는 시작 크레딧 (p. 207)을 획득하지만 T2 무제한 인스턴스에서는 시작 크레딧을 획득하지 않습니다. 24시간 동안 또는 인스턴스 수명(더 짧음) 동안 평균 CPU 사용률이 기준 이하인 경우, T2 무제한 인스턴스는 언제라도 추가 요금 없이 기준 성능 이상으로 버스트가 가능합니다. 따라서 T2 무제한 인스턴스는 시작 크레딧 없이도 시작 즉시 높은 성능을 달성할 수 있습니다.

T2 인스턴스가 `standard`에서 `unlimited`으로 전환된 경우 남은 `CPUCreditBalance`가 전달되기 전에 `CPUCreditBalance`에서 누적된 시작 크레딧이 모두 삭제됩니다.

> **Note**
>
> T3 및 T3a 인스턴스는 시작 크레딧을 획득하지 않습니다.

무제한 모드 활성화

T3 및 T3a 인스턴스는 `unlimited`로 시작하도록 기본 설정되어 있습니다. T2 인스턴스는 기본적으로 `standard`로 시작하지만 시작 시 `unlimited`를 활성화할 수 있습니다.

실행 중이거나 중지된 인스턴스에서 언제든지 `unlimited`에서 `standard`로, `standard`에서 `unlimited`로 전환할 수 있습니다. 자세한 내용은 무제한 또는 스탠다드로 성능 순간 확장 가능 인스턴스 시작 (p. 220) 및 성능 순간 확장 가능 인스턴스의 크레딧 사양 수정 (p. 222) 단원을 참조하십시오.

성능 순간 확장 가능 인스턴스 패밀리마다 AWS 리전별 계정 수준에서 `unlimited`를 기본 크레딧 옵션으로 설정하여 계정에 속한 새 성능 순간 확장 가능 인스턴스가 모두 기본 크레딧 옵션을 사용하여 시작되게 할 수 있습니다. 자세한 내용은 계정의 기본 크레딧 사양 설정 (p. 223) 단원을 참조하십시오.

Amazon EC2 콘솔 또는 AWS CLI를 사용하여 확장 가능한 성능 인스턴스가 `unlimited` 또는 `standard`로 구성되었는지 확인할 수 있습니다. 자세한 내용은 성능 순간 확장 가능 인스턴스의 크레딧 사양 보기 (p. 222) 및 기본 크레딧 사양 보기 (p. 224) 단원을 참조하십시오.

무제한과 스탠다드에서 전환 시 크레딧에 발생하는 현상

`CPUCreditBalance`는 인스턴스에서 누적한 크레딧 수를 추적하는 CloudWatch 측정치입니다. `CPUSurplusCreditBalance`는 인스턴스에서 사용한 잉여 크레딧 수를 추적하는 CloudWatch 측정치입니다.

`unlimited`로 구성된 인스턴스를 `standard`로 변경하면 다음이 발생합니다.

- `CPUCreditBalance` 값은 변경되지 않은 채 전달됩니다.
- `CPUSurplusCreditBalance` 값은 즉시 요금이 부과됩니다.

`standard` 인스턴스가 `unlimited`로 전환될 경우 다음이 발생합니다.

- 누적된 획득 크레딧이 포함된 `CPUCreditBalance` 값이 전달됩니다.
- T2 스탠다드 인스턴스의 경우 `CPUCreditBalance` 값에서 모든 시작 크레딧이 삭제되고, 누적된 획득 크레딧이 포함된 나머지 `CPUCreditBalance` 값이 전달됩니다.

크레딧 사용량 모니터링

인스턴스가 기준 이상의 크레딧을 사용하고 있는지 여부를 확인하기 위해 CloudWatch 측정치를 사용하여 사용량을 추적할 수 있으며 시간별 경보를 설정하여 크레딧 사용량에 대한 알림을 받을 수 있습니다. 자세한 내용은 CPU 크레딧 모니터링 (p. 224) 단원을 참조하십시오.

예: 무제한 모드

다음은 `unlimited`로 구성된 인스턴스에 크레딧 사용을 설명하는 예입니다.

예제

- 예 1: T3 무제한에서 크레딧을 사용하는 것에 관해 설명 (p. 203)
- 예 2: T2 무제한에서 크레딧을 사용하는 것에 관해 설명 (p. 204)

예 1: T3 무제한에서 크레딧을 사용하는 것에 관해 설명

이 예에서는 `unlimited`로 시작된 `t3.nano` 인스턴스의 CPU 사용률과 CPU 성능 유지를 위해 획득 및 잉여 크레딧을 어떻게 사용하고 있는지 보여줍니다.

`t3.nano` 인스턴스는 24시간 동안 144개의 CPU 크레딧을 획득하고, 이를 사용하여 144분의 vCPU 사용 시간을 확보할 수 있습니다. CPU 크레딧 밸런스(CloudWatch 측정치 `CPUCreditBalance`에 의해 표현)가 고갈되면 인스턴스는 아직 획득되지 않은 잉여 CPU— —크레딧을 사용하여 필요한 시간 동안 버스트를 할 수 있습니다. `t3.nano` 인스턴스는 24시간 동안 최대 144개의 크레딧을 획득하기 때문에 즉시 요금을 부과하지 않고 이 최대 값까지 잉여 크레딧을 소비할 수 있습니다. 144개 이상의 CPU 크레딧을 사용하고 있는 경우에는 해당 시간이 끝날 때 그 차이만큼 비용이 부과됩니다.

이 예제는 다음 그래프를 통해 `CPUCreditBalance`가 감소한 이후에도 인스턴스가 잉여 크레딧을 사용하여 어떻게 버스트를 할 수 있는지 보여줍니다. 아래 워크플로는 그래프에서 번호가 매겨진 지점을 참조합니다.

P1 – 그래프의 0시간에서 인스턴스는 `unlimited`로 시작되며 즉시 크레딧을 획득하기 시작합니다. 인스턴스는 시작된 시간부터 유휴 상태로 유지되어 CPU 사용률이 0%이므로 크레딧이 사용되지 않습니다. 사용하지 않은 모든 크레딧은 크레딧 밸런스에 누적됩니다. 처음 24시간 동안 `CPUCreditUsage`는 0이고 `CPUCreditBalance` 값은 최대 144에 이릅니다.

P2 – 향후 12시간 동안 CPU 사용률은 2.5%이며, 이는 5% 기준 아래입니다. 인스턴스는 사용하는 크레딧보다 더 많은 크레딧을 획득하지만, `CPUCreditBalance` 값은 최대 144 크레딧을 초과할 수 없습니다.

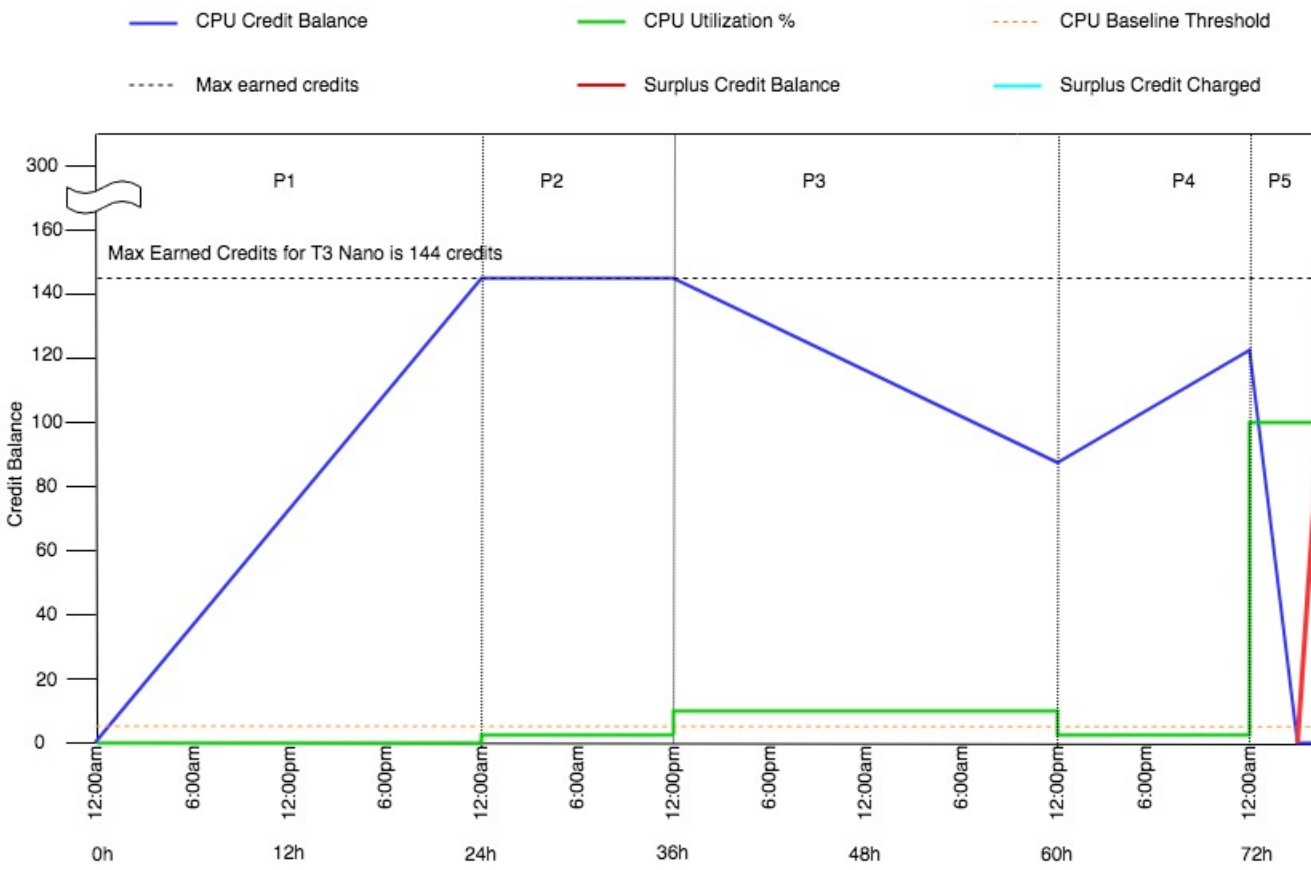
P3 – 향후 24시간 동안 CPU 사용률은 7%(기준보다 높음)이며, 이를 위해서는 57.6 크레딧을 사용해야 합니다. 인스턴스는 획득한 것보다 더 많은 크레딧을 사용하므로 `CPUCreditBalance` 값은 86.4 크레딧으로 감소합니다.

P4 – 향후 12시간 동안 CPU 사용률은 2.5%(기준보다 낮음)로 감소하며, 이를 위해서는 36 크레딧을 사용해야 합니다. 인스턴스에서는 동시에 72 크레딧을 획득할 수 있습니다. 인스턴스는 사용하는 크레딧보다 더 많은 크레딧을 획득하므로 `CPUCreditBalance` 값은 122 크레딧으로 증가합니다.

P5 – 향후 5시간 동안 인스턴스는 100% CPU 사용률로 버스트하고 이 버스트를 지속하기 위해 총 570 크레딧을 사용합니다. 이 기간 중 1시간이 지나면 인스턴스는 122 크레딧의 전체 `CPUCreditBalance`를 소진하고 높은 CPU 성능을 유지하기 위해 잉여 크레딧을 사용하기 시작해 이 기간 동안 총 448 잉여 크레딧(570-122=448)을 사용합니다. `CPUSurplusCreditBalance` 값이 144 CPU 크레딧(`t3.nano` 인스턴스는 24시간 동안 획득할 수 있는 최대 크레딧)에 이르면 이후에 사용된 모든 잉여 크레딧은 획득한 크레딧으로 상쇄되지 않습니다. 이후에 사용된 잉여 크레딧은 304 크레딧(448-144=304)에 해당하며, 이로써 304 크레딧에 대한 시간이 종료될 때 약간의 추가 요금이 발생하게 됩니다.

P6 – 향후 13시간 동안 CPU 사용률은 5%(기준)입니다. 인스턴스는 사용하는 크레딧과 동일한 크레딧을 획득하므로 `CPUSurplusCreditBalance`를 청산할 여력은 없습니다. `CPUSurplusCreditBalance` 값은 144 크레딧을 유지합니다.

P7 – 이 예에서는 최근 24시간 동안 인스턴스가 유휴 상태로, CPU 사용률이 0%입니다. 이 기간 동안 인스턴스는 144 크레딧을 획득하고 이 크레딧은 `CPUSurplusCreditBalance`를 청산하는 데 사용합니다.



예 2: T2 무제한에서 크레딧을 사용하는 것에 관해 설명

이 예에서는 `unlimited`로 시작된 `t2.nano` 인스턴스의 CPU 사용률과 CPU 성능 유지를 위해 획득 및 잉여 크레딧을 어떻게 사용하고 있는지 보여줍니다.

`t2.nano` 인스턴스는 24시간 동안 72개의 CPU 크레딧을 획득하고, 이를 사용하여 72분의 vCPU 사용 시간을 확보할 수 있습니다. CPU 크레딧 밸런스(CloudWatch 측정치 `CPUCreditBalance`에 의해 표현)가 고갈되면 인스턴스는 아직 획득되지 않은 잉여 CPU— —크레딧을 사용하여 필요한 시간 동안 버스트를 할 수 있습니다. `t2.nano` 인스턴스는 24시간 동안 최대 72개의 크레딧을 획득하기 때문에 즉시 요금을 부과하지 않고 이 최대 값까지 잉여 크레딧을 소비할 수 있습니다. 72개 이상의 CPU 크레딧을 사용하고 있는 경우에는 해당 시간이 끝날 때 그 차이만큼 비용이 부과됩니다.

이 예제는 다음 그래프를 통해 `CPUCreditBalance`가 감소한 이후에도 인스턴스가 잉여 크레딧을 사용하여 어떻게 버스트를 할 수 있는지 보여줍니다. 그래프의 타임 라인 시작 지점에서 인스턴스가 24시간 동안 획득할 수 있는 최대 수와 동일한 크레딧 밸런스를 누적했다고 가정할 수 있습니다. 아래 워크플로는 그래프에서 번호가 매겨진 지점을 참조합니다.

1 – 처음 10분 동안 `CPUCreditUsage`가 0이고 `CPUCreditBalance` 값이 최대 72로 유지됩니다.

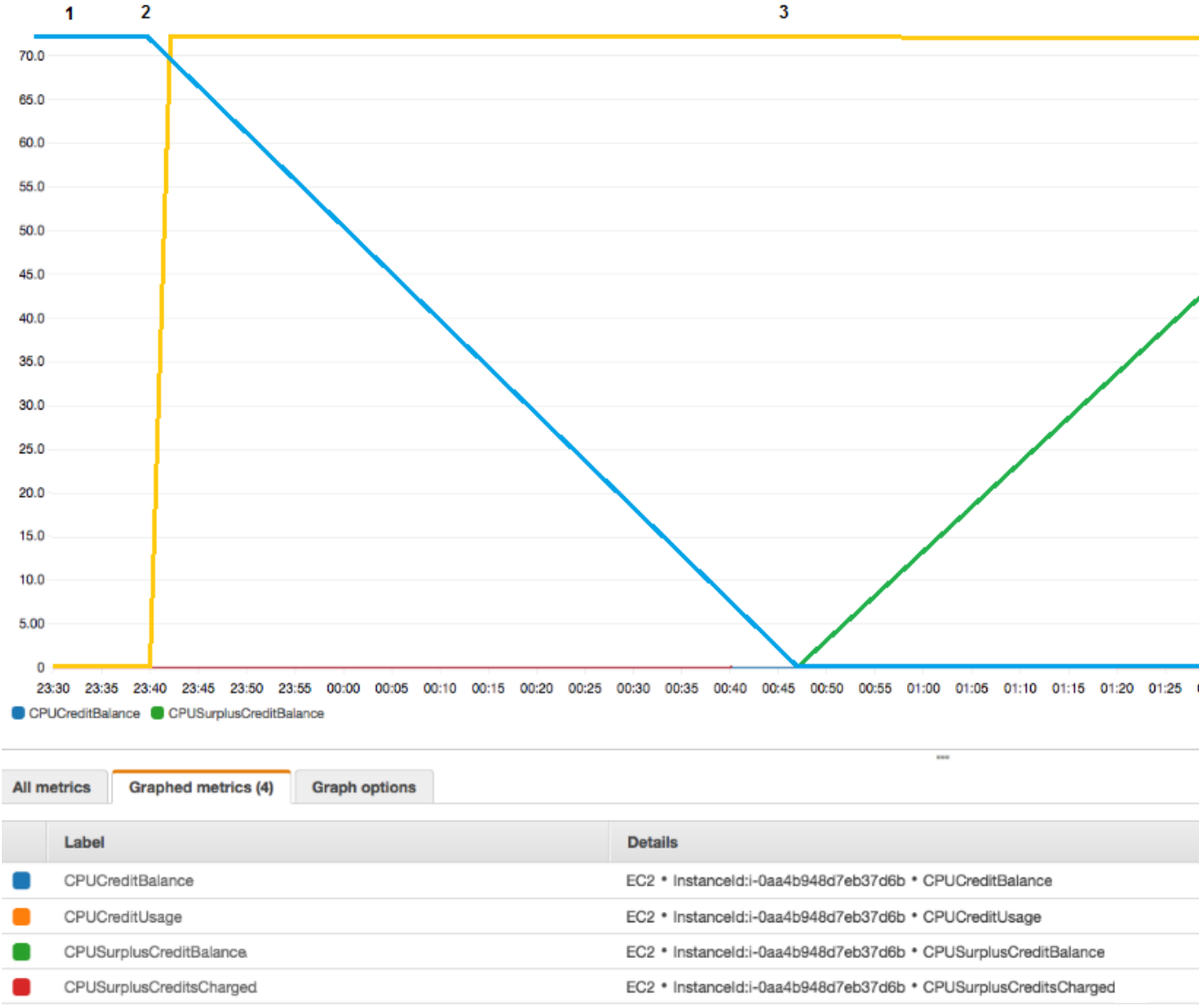
2 – 23:40에 CPU 사용률이 증가하면서 인스턴스가 CPU 크레딧을 사용하고, 이에 따라 `CPUCreditBalance` 값이 줄어듭니다.

3 – 00:47경, 인스턴스에서 전체 `CPUCreditBalance`가 고갈되고 높은 CPU 성능을 유지하기 때문에 잉여 크레딧을 사용하기 시작합니다.

4 – `CPUSurplusCreditBalance` 값이 72 CPU 크레딧에 도달하는 01:55까지 잉여 크레딧이 사용됩니다. 이는 `t2.nano` 인스턴스가 24시간 동안 획득할 수 있는 최대 값과 동일합니다. 이후에 사용된 모든 잉여 크

레딧은 24시간 내에 획득한 크레딧으로 상쇄가 되지 않기 때문에 해당 시간이 끝날 때 약간의 추가 요금이 발생하게 됩니다.

5 – 인스턴스가 02:20경까지 잉여 크레딧을 계속해 사용합니다. 이때 CPU 사용률이 기준 이하로 떨어지면 인스턴스는 시간당 3개씩(5분마다 0.25개) 크레딧을 획득하기 시작합니다. 이는 `CPUSurplusCreditBalance`를 청산하는 데 사용됩니다. `CPUSurplusCreditBalance` 값이 줄어들어 0이 되고 나면 인스턴스는 5분마다 0.25개씩 `CPUCreditBalance` 획득 크레딧을 누적하기 시작합니다.



청구서 계산

vCPU-시간당 잉여 크레딧의 요금은 0.05 USD입니다. 인스턴스는 01:55부터 02:20까지 약 25개의 잉여 크레딧을 소비했으며, 이는 0.42 vCPU-시간에 해당됩니다.

이 인스턴스에 대한 추가 요금은 0.42 vCPU-시간 x 0.05 USD/vCPU-시간 = 0.021 USD(반올림해서 0.02 USD).

여기 이 T2 무제한 인스턴스에 대한 월말 청구서가 나와 있습니다.

Amazon Elastic Compute Cloud running Linux/UNIX		
$0.0058 per On Demand Linux t2.nano Instance Hour	720.000 Hrs	$4.18

Amazon Elastic Compute Cloud T2 CPU Credits		
$0.05 per vCPU-Hour of T2 CPU credits	0.420 vCPU-Hours	$0.02

발생하는 모든 요금을 매시간 공지하는 청구서 알림을 설정하고 필요 시 조치를 취할 수 있습니다.

버스트 가능한 성능 인스턴스의 표준 모드

`standard`로 구성된 성능 순간 확장 가능 인스턴스는 평균 CPU 사용률이 인스턴스의 기준 성능보다 일관되게 낮은 워크로드에 적합합니다. 기준 이상으로 버스트하려면 인스턴스는 CPU 크레딧 밸런스에 누적한 크레딧을 사용합니다. 인스턴스가 획득한 크레딧이 부족해지면 성능이 점차적으로 기준 성능 수준으로 떨어지기 때문에 획득한 CPU 크레딧 밸런스가 고갈되어도 급격한 성능 저하가 발생하지 않습니다. 자세한 내용은 성능 순간 확장 가능 인스턴스에 대한 CPU 크레딧 및 기준 성능 (p. 195) 단원을 참조하십시오.

내용

- 스탠다드 모드 개념 (p. 206)
  - 스탠다드 성능 순간 확장 가능 인스턴스의 작동 방식 (p. 206)
  - 시작 크레딧 (p. 207)
  - 시작 크레딧 한도 (p. 207)
  - 시작 크레딧과 획득 크레딧의 차이 (p. 208)
- 예: 스탠다드 모드 (p. 208)
  - 예 1: T3 스탠다드에서 크레딧을 사용하는 것에 관한 설명 (p. 208)
  - 예 2: T2 스탠다드에서 크레딧을 사용하는 것에 관한 설명 (p. 210)
    - 기간 1: 1 – 24시간 (p. 211)
    - 기간 2: 25 – 36시간 (p. 212)
    - 기간 3: 37 – 61시간 (p. 213)
    - 기간 4: 62 – 72시간 (p. 214)
    - 기간 5: 73 – 75시간 (p. 216)
    - 기간 6: 76 – 90시간 (p. 217)
    - 기간 7: 91 – 96시간 (p. 218)

스탠다드 모드 개념

`standard` 모드는 성능 순간 확장 가능 인스턴스에 사용할 수 있는 구성 옵션입니다. 이 모드는 실행 중인 또는 중지된 인스턴스에 대해 언제든지 활성화 또는 비활성화할 수 있습니다. 성능 순간 확장 가능 인스턴스 패밀리마다 AWS 리전별 계정 수준에서 `standard`를 기본 크레딧 옵션으로 설정하여 계정에 속한 새 성능 순간 확장 가능 인스턴스가 모두 기본 크레딧 옵션을 사용하여 시작되게 할 수 있습니다.

> **Note**
>
> T3 및 T3a 인스턴스는 `unlimited`로 시작하도록 기본 설정되어 있습니다. T2 인스턴스는 `standard`로 시작하도록 기본 설정되어 있습니다. AWS 리전별로 계정 수준에서 기본값을 변경할 수 있습니다. 자세한 내용은 계정의 기본 크레딧 사양 설정 (p. 223) 단원을 참조하십시오.

스탠다드 성능 순간 확장 가능 인스턴스의 작동 방식

`standard`로 구성된 성능 순간 확장 가능 인스턴스는 실행 중 상태인 경우 시간당 특정 비율의 획득 크레딧을 지속적으로 (밀리초 수준의 시간 정밀도로) 획득합니다. T2 스탠다드 인스턴스가 중지되면 발생한 크레딧

이 모두 손실되고 크레딧 밸런스가 0으로 재설정됩니다. 인스턴스가 다시 시작되면 새로운 세트의 시작 크레딧이 지급되고 획득 크레딧이 누적되기 시작합니다. T3 및 T3a 스탠다드의 경우 인스턴스가 중단된 후 CPU 크레딧 밸런스가 7일 동안 지속하다가 7일이 지나면 크레딧이 상실됩니다. 7일 이내에 인스턴스를 시작하면 크레딧이 상실되지 않습니다.

T2 스탠다드 인스턴스에는 두 가지 CPU 크레딧이 지급됩니다. 획득 크레딧 및 시작 크레딧. T2 스탠다드 인스턴스가 실행 중 상태인 경우 지속적으로 시간당 특정 비율의 획득 크레딧을 획득합니다(밀리초 수준의 시간 정밀도). 시작 시에는 아직 뛰어난 시작 환경을 위한 크레딧이 없으므로, 뛰어난 시작 환경을 제공하기 위해 획득 크레딧이 누적되는 동안 먼저 소비할 수 있도록 시작 시에 시작 크레딧이 지급됩니다.

T3 및 T3a 스탠다드 인스턴스는 시작 크레딧을 획득하지 않습니다.

시작 크레딧

T2 스탠다드 인스턴스는 시작 또는 재시작 시 vCPU당 30개의 시작 크레딧이 지급됩니다. 예를 들어 `t2.micro` 인스턴스는 1개의 vCPU에서 30개의 시작 크레딧을 획득하는 반면에 `t2.xlarge` 인스턴스는 4개의 vCPU에서 120개의 시작 크레딧을 획득합니다. 시작 크레딧은 획득 크레딧을 누적하기 전에 인스턴스가 시작 즉시 버스트를 할 수 있도록 허용하는 뛰어난 시작 경험을 제공하도록 설계되었습니다.

시작 크레딧은 획득 크레딧보다 먼저 소비됩니다. 소비되지 않은 시작 크레딧은 CPU 크레딧 밸런스에 누적됩니다. 하지만 CPU 크레딧 밸런스 한도에 포함되지 않습니다. 예를 들어 `t2.micro` 인스턴스는 최대 144의 CPU 크레딧 밸런스 한도를 가지고 있습니다. 시작된 후 24시간 이상 유휴 상태로 지속된 경우 CPU 크레딧 밸런스는 한도 이상인 174(시작 크레딧 30 + 획득 크레딧 144)에 도달합니다. 그러나 인스턴스가 30개의 시작 크레딧을 사용하고 나면 크레딧 밸런스가 144개를 초과할 수 없습니다. 각 인스턴스 크기별 CPU 크레딧 밸런스 한도에 대한 자세한 내용은 크레딧 표 (p. 195)를 참조하십시오.

아래 표에는 시작 시 획득한 초기 CPU 크레딧 할당과 vCPU의 수가 나와 있습니다.

인스턴스 유형	시작 크레딧	vCPUs
`t1.micro`	15	1
`t2.nano`	30	1
`t2.micro`	30	1
`t2.small`	30	1
`t2.medium`	60	2
`t2.large`	60	2
`t2.xlarge`	120	4
`t2.2xlarge`	240	8

시작 크레딧 한도

T2 스탠다드 인스턴스가 시작 크레딧을 획득할 수 있는 횟수에는 제한이 있습니다. 기본 한도는 24시간마다 계정, 리전 및 24시간당 모든 T2 스탠다드 인스턴스에 대해 총 100회 시작입니다. 예를 들어 한 인스턴스가 24시간 이내에 100회 중지 및 시작되는 경우, 24시간 이내에 100개의 인스턴스가 시작되는 경우 또는 기타 조합으로 100회의 시작에 도달한 경우 한도에 도달하게 됩니다. 새 계정에는 사용량에 따라 증가하는 하한이 설정되어 있을 수 있습니다.

> **Tip**
>
> 워크로드가 항상 필요한 성능을 얻도록 하려면 성능 순간 확장 가능 인스턴스의 무제한 모드 (p. 197) 전환 또는 크기가 더 큰 인스턴스 사용을 고려하십시오.

시작 크레딧과 획득 크레딧의 차이

다음 표에는 시작 크레딧과 획득 크레딧의 차이가 나와 있습니다.

	시작 크레딧	획득 크레딧
크레딧 획득률	T2 스탠다드 인스턴스는 시작 또는 재시작 시 vCPU당 30개의 시작 크레딧이 지급됩니다. T2 인스턴스가 `unlimited`에서 `standard`로 전환되는 경우 전환되는 시점에는 이 인스턴스에서 시작 크레딧을 획득하지 않습니다.	각 T2 인스턴스는 인스턴스 크기에 따라 지속적으로 특정 비율의 시간당 CPU 크레딧을 얻습니다(밀리초 수준의 시간 정밀도로). 인스턴스 크기에 따라 지급되는 CPU 크레딧 수에 대한 자세한 내용은 크레딧 표 (p. 195)를 참조하십시오.
크레딧 획득 한도	시작 크레딧 획득 한도는 24시간마다 계정, 리전 및 24시간당 모든 T2 스탠다드 인스턴스에 대해 총 100회 시작입니다. 새 계정에는 사용량에 따라 증가하는 하한이 설정되어 있을 수 있습니다.	T2 인스턴스는 CPU 크레딧 밸런스 한도 이상의 크레딧을 누적할 수 없습니다. CPU 크레딧 밸런스가 한도에 도달한 경우 한도 도달 이후 획득한 모든 크레딧은 삭제됩니다. 시작 크레딧은 한도에 포함되지 않습니다. 각 T2 인스턴스 크기별 CPU 크레딧 밸런스 한도에 대한 자세한 내용은 크레딧 표 (p. 195)를 참조하십시오.
크레딧 사용	시작 크레딧은 획득 크레딧보다 먼저 소비됩니다.	획득 크레딧은 모든 시작 크레딧이 소비된 후에만 소비됩니다.
크레딧 만료	T2 인스턴스가 실행 중인 동안 시작 크레딧은 만료되지 않습니다. T2 스탠다드 인스턴스가 중단되거나 T2 무제한으로 전환될 때 모든 시작 크레딧이 삭제됩니다.	T2 인스턴스가 실행 중일 때는 누적된 획득 크레딧이 만료되지 않습니다. T2 인스턴스가 중지되면 누적된 획득 크레딧이 모두 상실됩니다.

누적된 시작 크레딧 및 획득 크레딧의 수는 CloudWatch 지표 `CPUCreditBalance`를 통해 추적됩니다. 자세한 내용은 CloudWatch 지표 (p. 224) 표에서 `CPUCreditBalance` 항목을 참조하십시오.

예: 스탠다드 모드

다음은 인스턴스가 `standard`로 구성되었을 때의 크레딧 사용을 설명하는 예입니다.

예제

- 예 1: T3 스탠다드에서 크레딧을 사용하는 것에 관한 설명 (p. 208)
- 예 2: T2 스탠다드에서 크레딧을 사용하는 것에 관한 설명 (p. 210)

예 1: T3 스탠다드에서 크레딧을 사용하는 것에 관한 설명

이 예에서는 `standard`로 시작된 `t3.nano` 인스턴스가 획득 크레딧을 획득, 축적, 사용하는 방식을 보여줍니다. 이로써 누적된 획득 크레딧이 크레딧 밸런스에 반영되는 방식을 알 수 있습니다.

> **Note**
>
> `standard`로 구성된 T3 및 T3a 인스턴스는 시작 크레딧을 획득하지 않습니다.

실행 중인 `t3.nano` 인스턴스는 24시간마다 144개 크레딧을 획득합니다. 크레딧 밸런스 한도는 획득 크레딧 144개입니다. 한도에 도달하면 새로 획득한 크레딧이 삭제됩니다. 획득 및 누적될 수 있는 크레딧 수에 대한 자세한 내용은 크레딧 표 (p. 195)를 참조하십시오.

T3 스탠다드 인스턴스를 시작하고 즉시 사용할 수 있습니다. 또는 T3 스탠다드 인스턴스를 시작하고 애플리케이션을 실행하기 전에 며칠 동안 유휴 상태로 둘 수 있습니다. 인스턴스를 사용했는지 아니면 유휴 상태로 두었는지에 따라 크레딧이 사용되는지 또는 누적되는지가 결정됩니다. 인스턴스가 시작된 시간부터 24시간 동안 유휴 상태로 유지된 경우 크레딧 밸런스는 한도에 이릅니다. 여기서 한도는 누적될 수 있는 획득 크레딧의 최대 수입니다.

이 예에서는 시작 시간부터 24시간 동안 유휴 상태로 유지된 인스턴스에 대해 설명하며, 96시간 기간 동안 7단계 기간을 통해 크레딧이 획득, 누적, 사용되고 폐기되는 비율과 각 기간 종료 시 크레딧 밸런스의 값을 보여 줍니다.

아래 워크플로는 그래프에서 번호가 매겨진 지점을 참조합니다.

P1 – 그래프의 0시간에서 인스턴스는 `standard`로 시작되며 즉시 크레딧을 획득하기 시작합니다. 인스턴스는 시작된 시간부터 유휴 상태로 유지되어 CPU 사용률이 0%이므로 크레딧이 사용되지 않습니다. 사용하지 않은 모든 크레딧은 크레딧 밸런스에 누적됩니다. 처음 24시간 동안 `CPUCreditUsage`는 0이고 `CPUCreditBalance` 값은 최대 144에 이릅니다.

P2 – 향후 12시간 동안 CPU 사용률은 2.5%이며, 이는 5% 기준 아래입니다. 인스턴스는 사용하는 크레딧보다 더 많은 크레딧을 획득하지만, `CPUCreditBalance` 값은 최대 144 크레딧을 초과할 수 없습니다. 한도를 초과하여 획득한 모든 크레딧은 삭제됩니다.

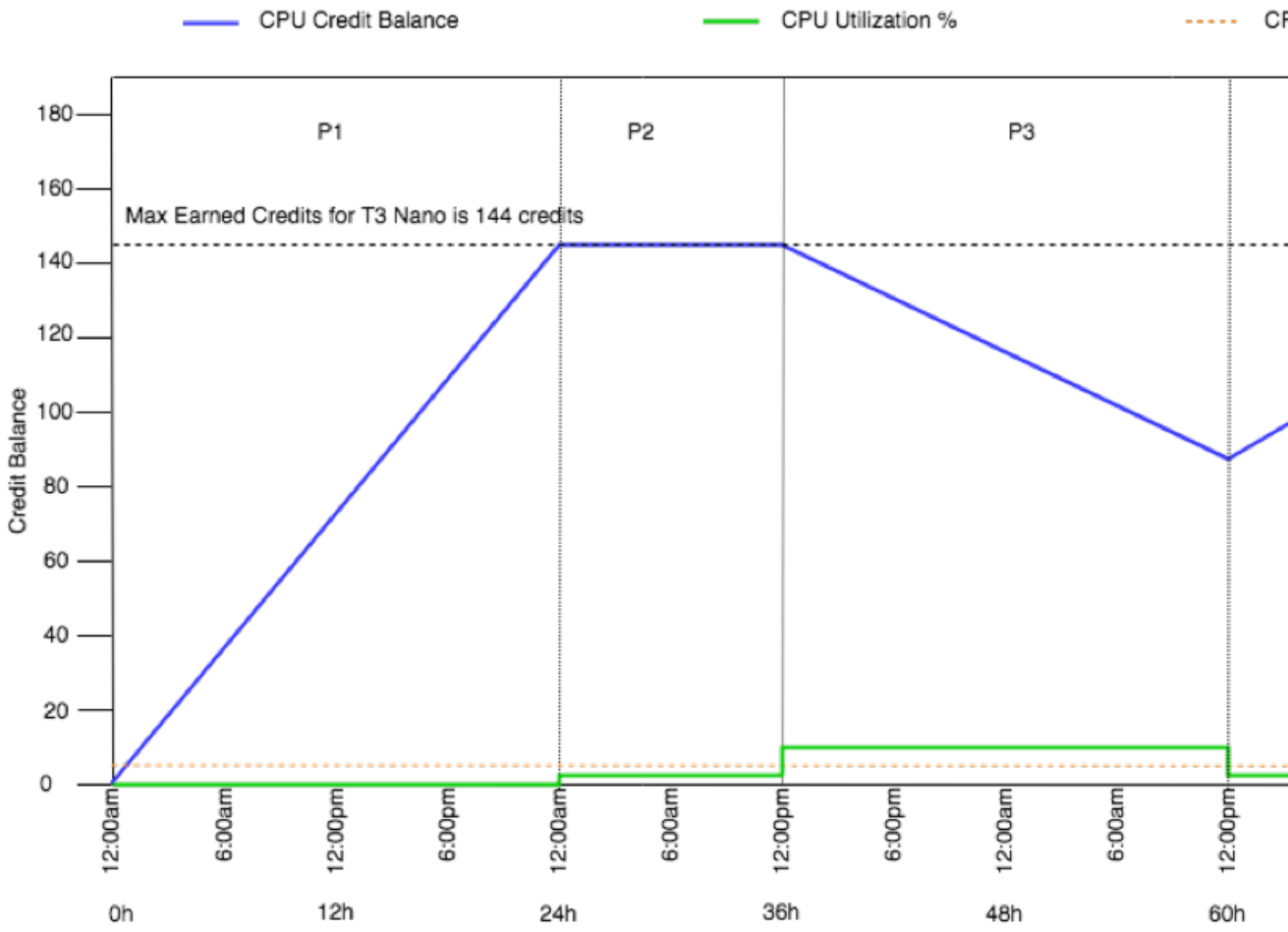
P3 – 향후 24시간 동안 CPU 사용률은 7%(기준보다 높음)이며, 이를 위해서는 57.6 크레딧을 사용해야 합니다. 인스턴스는 획득한 것보다 더 많은 크레딧을 사용하므로 `CPUCreditBalance` 값은 86.4 크레딧으로 감소합니다.

P4 – 향후 12시간 동안 CPU 사용률은 2.5%(기준보다 낮음)로 감소하며, 이를 위해서는 36 크레딧을 사용해야 합니다. 인스턴스에서는 동시에 72 크레딧을 획득할 수 있습니다. 인스턴스는 사용하는 크레딧보다 더 많은 크레딧을 획득하므로 `CPUCreditBalance` 값은 122 크레딧으로 증가합니다.

P5 – 향후 2시간 동안 인스턴스는 100% CPU 사용률로 버스트하고 전체 `CPUCreditBalance`인 122크레딧을 소진합니다. 이 기간이 종료되는 시점에 `CPUCreditBalance`가 0이고, 및 CPU 사용률은 강제로 5%의 기준 성능 수준으로 하락합니다. 기준 수준에서 인스턴스는 사용하는 크레딧과 동일한 크레딧을 획득합니다.

P6 – 향후 14시간 동안 CPU 사용률은 5%(기준)입니다. 인스턴스는 사용하는 크레딧과 동일한 크레딧을 획득합니다. `CPUCreditBalance` 값은 0을 유지합니다.

P7 – 이 예에서는 최근 24시간 동안 인스턴스가 유휴 상태로, CPU 사용률이 0%입니다. 이 기간 동안 인스턴스는 144크레딧을 획득하고 이 크레딧은 `CPUCreditBalance`에 누적됩니다.



예 2: T2 스탠다드에서 크레딧을 사용하는 것에 관한 설명

이 예제는 `standard`로 실행된 `t2.nano` 인스턴스가 어떻게 시작 및 획득 크레딧을 획득하고 축적하고 사용하는지를 보여줍니다. 크레딧 밸런스에 획득 크레딧의 누적뿐 아니라 시작 크레딧의 누적이 어떻게 반영되는지 볼 수 있습니다.

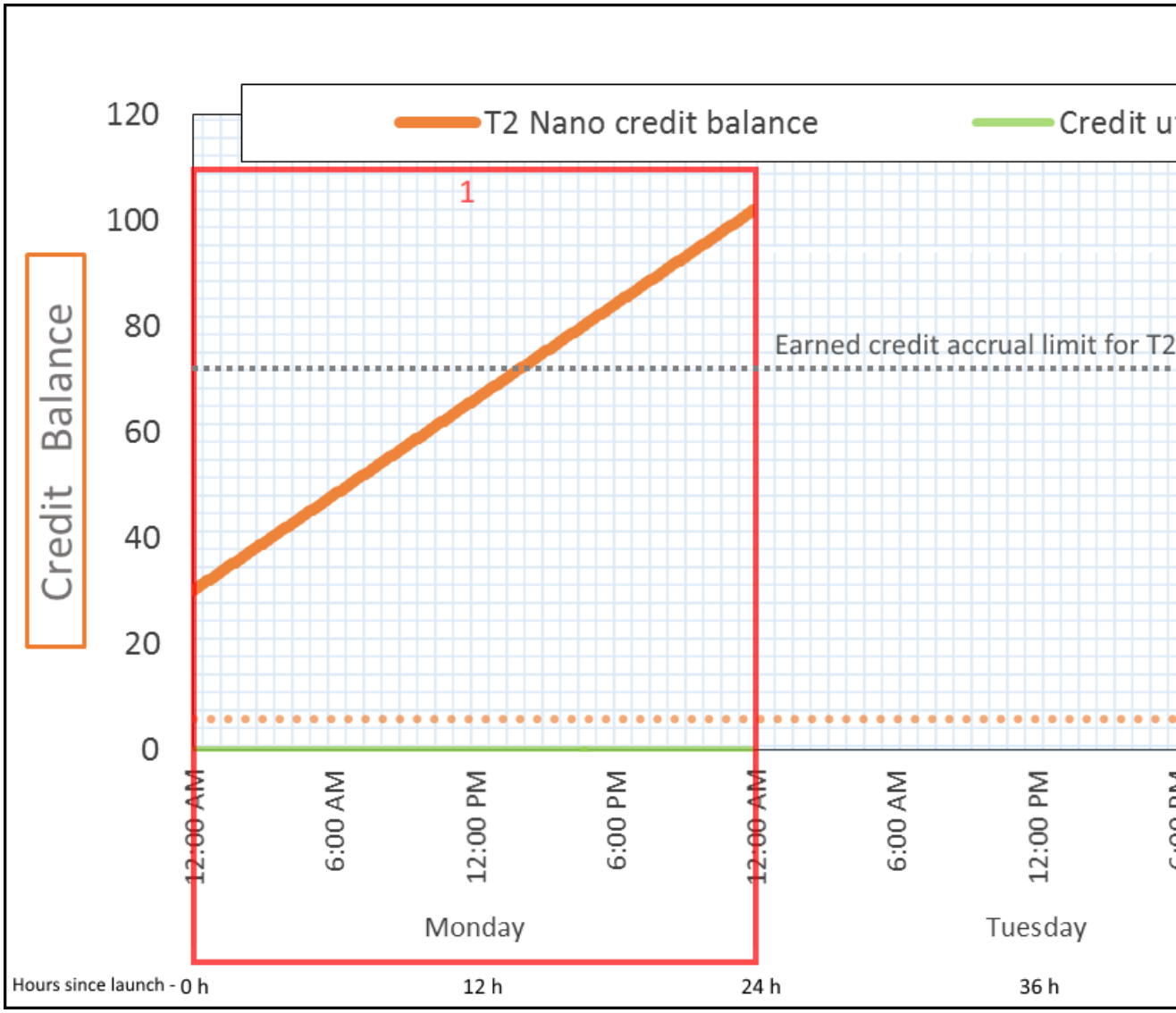
`t2.nano` 인스턴스는 시작 시 30개의 시작 크레딧을 받고 24시간마다 72개의 크레딧을 획득합니다. 크레딧 밸런스 한도는 획득 크레딧 72개이며, 시작 크레딧은 한도에 포함되지 않습니다. 한도에 도달하면 새로 획득한 크레딧이 삭제됩니다. 획득 및 누적될 수 있는 크레딧 수에 대한 자세한 내용은 크레딧 표 (p. 195)를 참조하십시오. 제한에 대한 자세한 내용은 시작 크레딧 한도 (p. 207) 단원을 참조하십시오.

T2 스탠다드 인스턴스를 시작하고 즉시 사용할 수 있습니다. 또는 T2 스탠다드 인스턴스를 시작하고 애플리케이션을 실행하기 전에 며칠 동안 유휴 상태로 둘 수 있습니다. 인스턴스를 사용했는지 아니면 유휴 상태로 두었는지에 따라 크레딧이 사용되는지 또는 누적되는지가 결정됩니다. 인스턴스가 시작 시간부터 24시간 동안 유휴 상태로 유지된 경우, 잔고에 획득 누적 크레딧과 시작 누적 크레딧이 모두 반영되어 잔고가 한도를 초과한 것으로 나타납니다. 하지만 CPU가 사용되면 시작 크레딧이 먼저 사용됩니다. 그 후 한도에는 누적될 수 있는 최대 획득 크레딧이 항상 반영됩니다.

이 예에서는 시작 시간부터 24시간 동안 유휴 상태로 유지된 인스턴스에 대해 설명하며, 96시간 기간 동안 7단계 기간을 통해 크레딧이 획득, 누적, 사용되고 폐기되는 비율과 각 기간 종료 시 크레딧 밸런스의 값을 보여 줍니다.

기간 1: 1 – 24시간

그래프의 0시간에서 T2 인스턴스는 `standard`로 시작되며 30개의 시작 크레딧을 바로 받습니다. 인스턴스가 실행 상태일 때 크레딧을 획득합니다. 인스턴스는 시작된 시간부터 유휴— 상태로 유지되어 CPU 사용률이 0%—이므로 크레딧이 사용되지 않습니다. 사용하지 않은 모든 크레딧은 크레딧 밸런스에 누적됩니다. 시작 후 약 14시간이 되면 크레딧 밸런스가 72(시작 크레딧 30 + 획득 크레딧 42)개가 되고, 이 값은 인스턴스가 24시간 안에 획득할 수 있는 값과 동일합니다. 시작 후 24시간이 경과하면 사용하지 않은 시작 크레딧이 크레딧 밸런스에 누적되기 때문에 크레딧 밸런스가 72개를 초과합니다. 즉 크레딧 밸런스는 102(시작 크레딧 30 + 획득 크레딧 72)입니다.—



크레딧 사용률	24시간당 0 크레딧(CPU 사용률 0%)
크레딧 획득률	24시간당 72 크레딧
크레딧 폐기율	24시간당 0 크레딧
크레딧 밸런스	102 크레딧(시작 크레딧 30 + 획득 크레딧 72)

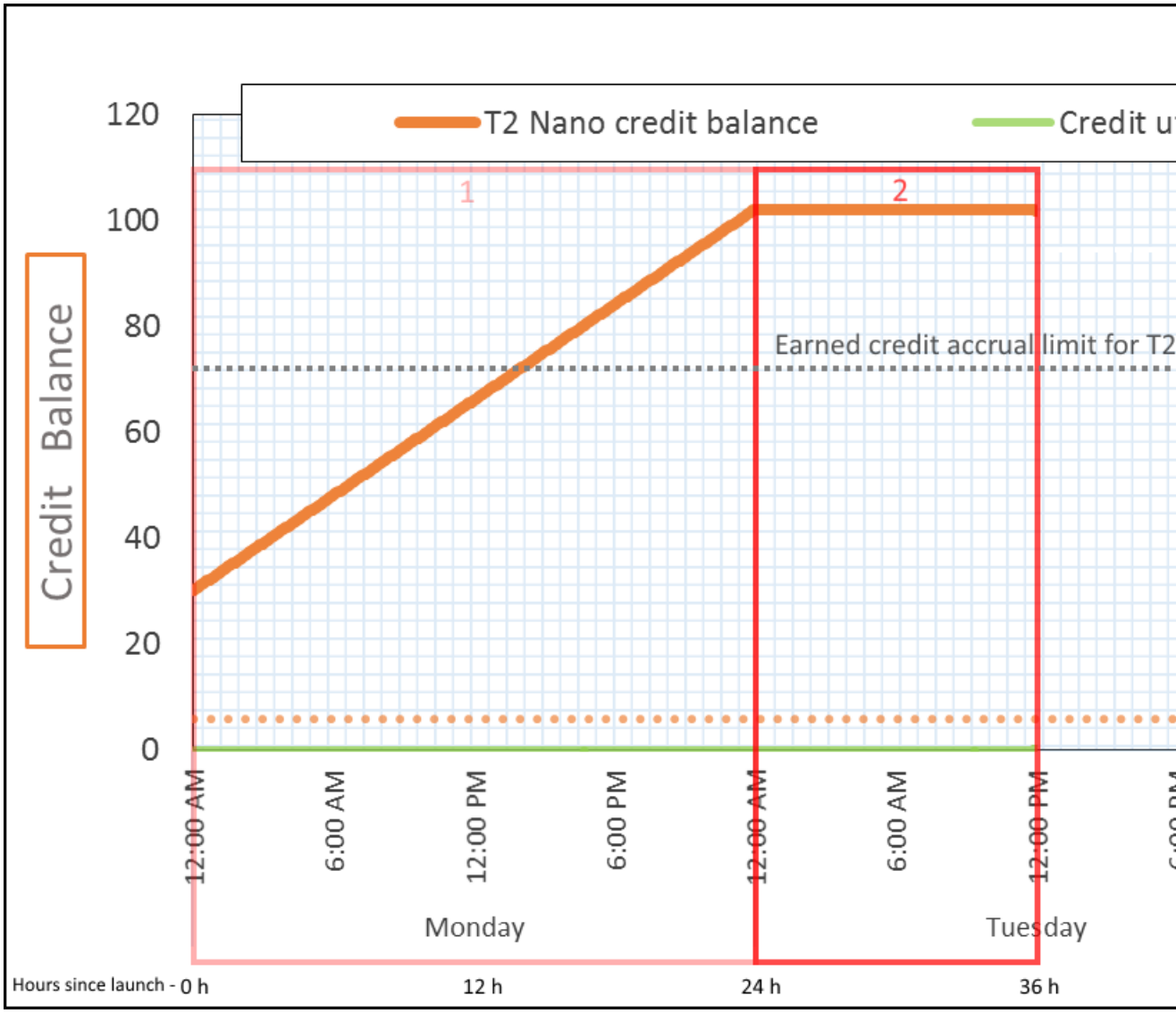
결론

시작 후 CPU를 사용하지 않으면 24시간 후에 적립할 수 있는 크레딧보다 더 많은 크레딧이 인스턴스에 적립됩니다(시작 크레딧 30 + 획득 크레딧 72 = 102 크레딧).

실제 상황에서 EC2 인스턴스는 시작 및 실행 중에 적은 양의 크레딧을 사용하므로 잔고는 이 예에서의 이론적인 최댓값에 도달하지 않습니다.

기간 2: 25 – 36시간

다음 12시간 동안 인스턴스는 계속 유휴 상태이고 크레딧을 획득하지만 크레딧 밸런스는 증가하지 않습니다. 102 크레딧(시작 크레딧 30 + 획득 크레딧 72)에서 더 이상 증가하지 않습니다. 크레딧 밸런스가 한도인 72개 획득 누적 크레딧에 도달한 경우 새로 획득한 크레딧은 버려집니다.



크레딧 사용률	24시간당 0 크레딧(CPU 사용률 0%)
크레딧 획득률	24시간당 72 크레딧(시간당 3 크레딧)
크레딧 폐기율	24시간당 72 크레딧(크레딧 획득률 100%)

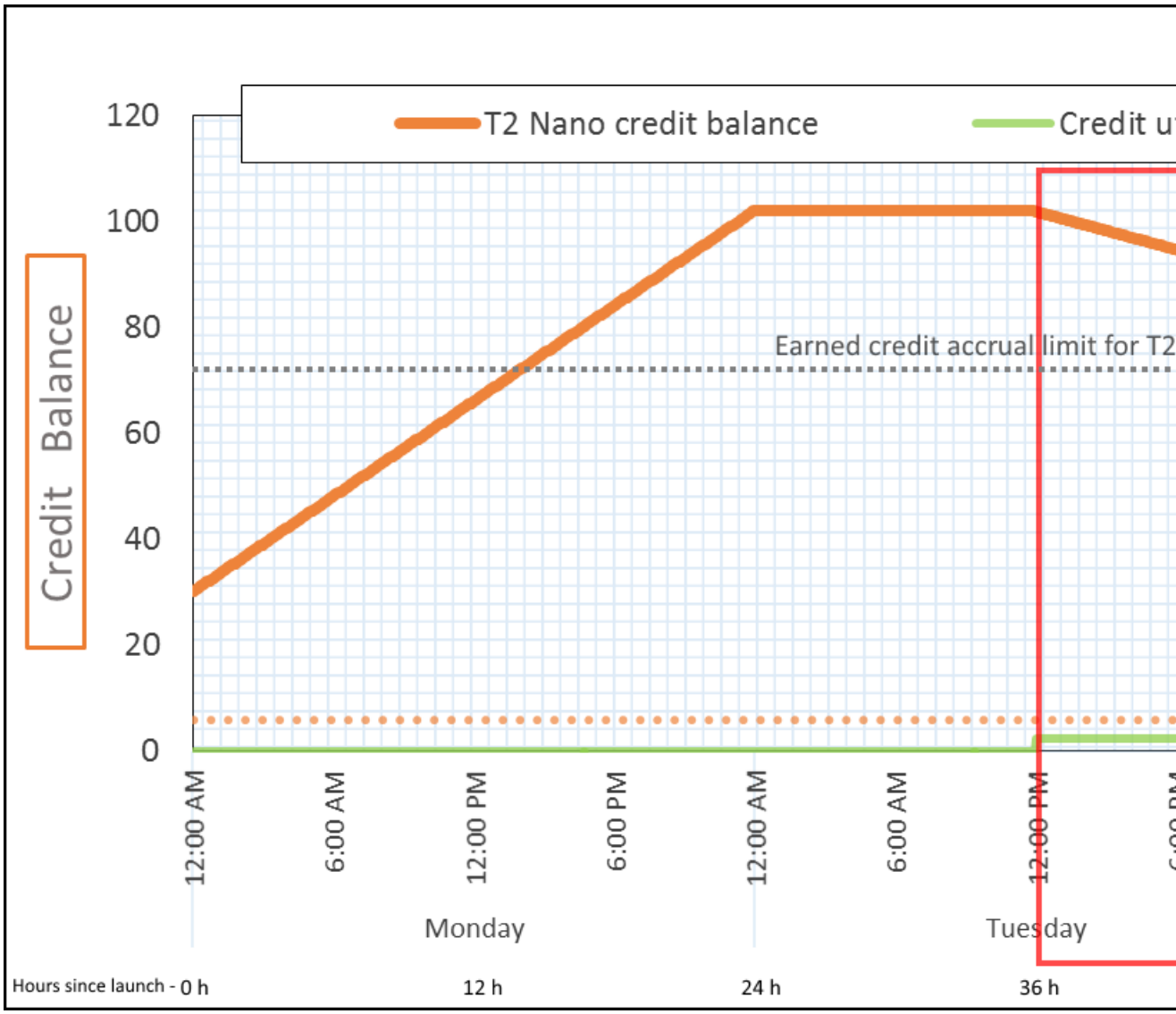
크레딧 밸런스	102크레딧(시작 크레딧 30 + 획득 크레딧 72)— – 잔고 변경 없음

결론

인스턴스는 크레딧을 계속 획득하지만 크레딧 밸런스가 한도에 도달하면 획득 크레딧을 더 이상 누적할 수 없습니다. 한도에 도달한 후 새로 획득한 크레딧은 버려집니다. 시작 크레딧은 크레딧 밸런스 한도에 포함되지 않습니다. 잔고에 시작 누적 크레딧이 포함되면 잔고가 한도를 초과한 것으로 나타납니다.

기간 3: 37 – 61시간

다음 25시간 동안 인스턴스는 2% CPU를 사용하며 이는 30 크레딧이 필요합니다. 동일한 기간에서 75 크레딧을 획득하지만 크레딧 밸런스는 감소합니다. 누적된 시작 크레딧이 처음 사용되고, 크레딧 밸런스가 이미 획득 크레딧 한도 72에 도달함에 따라 새로 획득한 크레딧은 버려지기 때문에 잔고가 감소합니다.



크레딧 사용률	24시간당 28.8 크레딧(시간당 1.2 크레딧, 2% CPU 사용률, 크레딧 획득률 400%) – 25시간 동안 30 크레딧—
크레딧 획득률	24시간당 72 크레딧
크레딧 폐기율	24시간당 72 크레딧(크레딧 획득률 100%)
크레딧 밸런스	72 크레딧(시작 크레딧 30개가 사용되고, 획득 크레딧 72개는 사용하지 않은 상태로 유지됨)

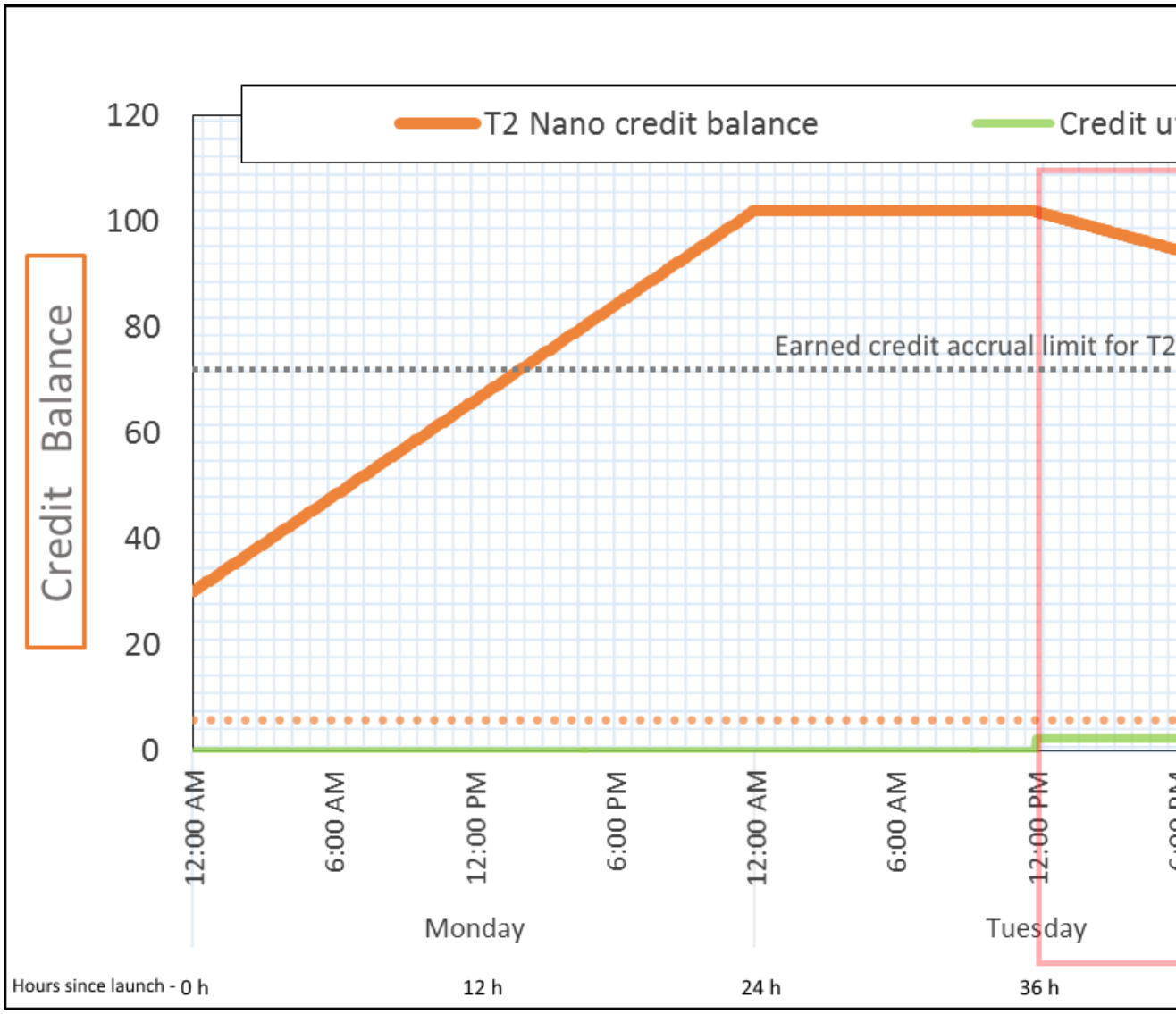
결론

인스턴스는 획득 크레딧을 사용하기 전에 시작 크레딧을 먼저 사용합니다. 시작 크레딧은 크레딧 한도에 포함되지 않습니다. 시작 크레딧이 사용된 후에는 24시간만에 획득할 수 있는 크레딧보다 잔고 더 많아지는 경우가 없습니다. 또한 인스턴스가 실행 중인 동안에는 시작 크레딧을 획득할 수 없습니다.

기간 4: 62 – 72시간

다음 11시간 동안 인스턴스는 2% CPU를 사용하며 이는 13.2 크레딧이 필요합니다. CPU 사용률은 이전 기간과 동일하지만 잔고는 감소하지 않습니다. 72 크레딧으로 유지됩니다.

크레딧 획득률이 크레딧 사용률보다 높기 때문에 잔고가 감소하지 않습니다. 인스턴스는 13.2개 크레딧을 사용하는 동안 33개 크레딧을 획득합니다. 하지만 잔고 한도는 72개이므로 이 한도를 초과하는 획득 크레딧은 버려집니다. 잔고는 72개로 유지되고, 이 값이 기간 2에서 102개 크레딧으로 유지된 것과 다른 이유는 획득 크레딧이 누적되지 않기 때문입니다.



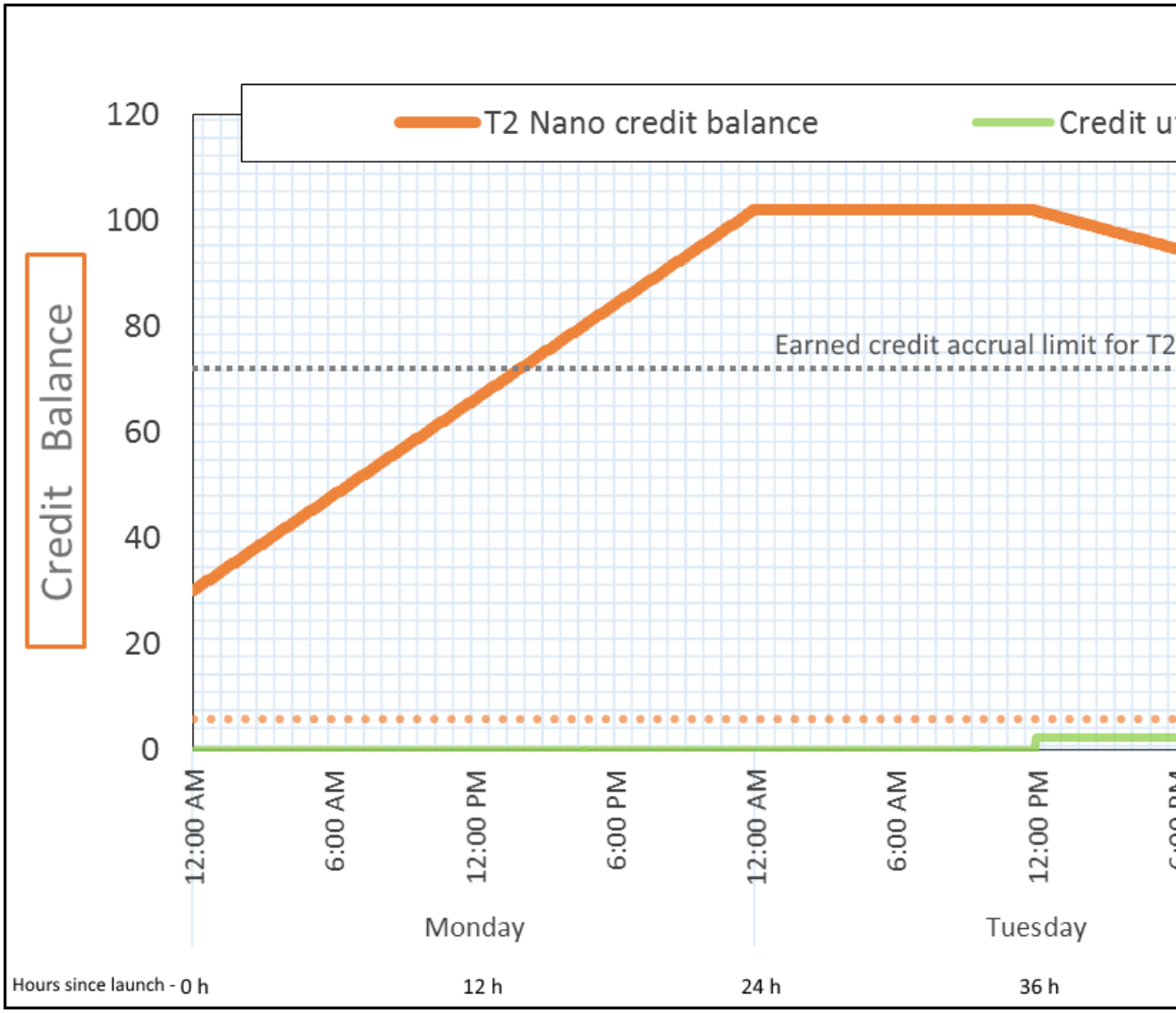
크레딧 사용률	24시간당 28.8 크레딧(시간당 1.2 크레딧, 2% CPU 사용률, 크레딧 획득률 40%) &endash; 11시간 동안 13.2 크레딧—
크레딧 획득률	24시간당 72 크레딧
크레딧 폐기율	24시간당 43.2 크레딧(크레딧 획득률 60%)
크레딧 밸런스	72 크레딧(시작 크레딧 0, 획득 크레딧 72) &endash;— 잔고가 한도에 이름

결론

시작 크레딧이 사용된 후에는 인스턴스가 24시간만에 획득할 수 있는 크레딧 수에 따라 크레딧 밸런스 한도가 결정됩니다. 인스턴스가 사요한 것보다 더 많은 크레딧을 획득한 경우 새로 획득한 크레딧 중 한도를 초과하는 크레딧은 버려집니다.

기간 5: 73 – 75시간

다음 3시간 동안 인스턴스의 CPU 사용률은 20%가 되고 36개의 크레딧을 사용합니다. 인스턴스는 이 3시간 동안 9개의 크레딧을 획득하므로 실제로 크레딧 밸런스는 27개가 감소합니다. 3시간이 지나면 크레딧 밸런스는 45개(획득 누적 크레딧)가 됩니다.



크레딧 사용률	24시간당 288 크레딧(시간당 12 크레딧, 20% CPU 사용률, 크레딧 획득률 400%) &endash; 3시간 동안 36 크레딧—
크레딧 획득률	24시간당 72 크레딧(3시간 동안 9 크레딧)
크레딧 폐기율	24시간당 0 크레딧
크레딧 밸런스	45 크레딧(이전 잔고(72) - 사용한 크레딧(36) +— 획득한 크레딧(9)) &endash; 24시간당 잔고 감소율 216개(사용률 288/24 + 획득률 72/24 = 잔고 감소율 216/24)

결론

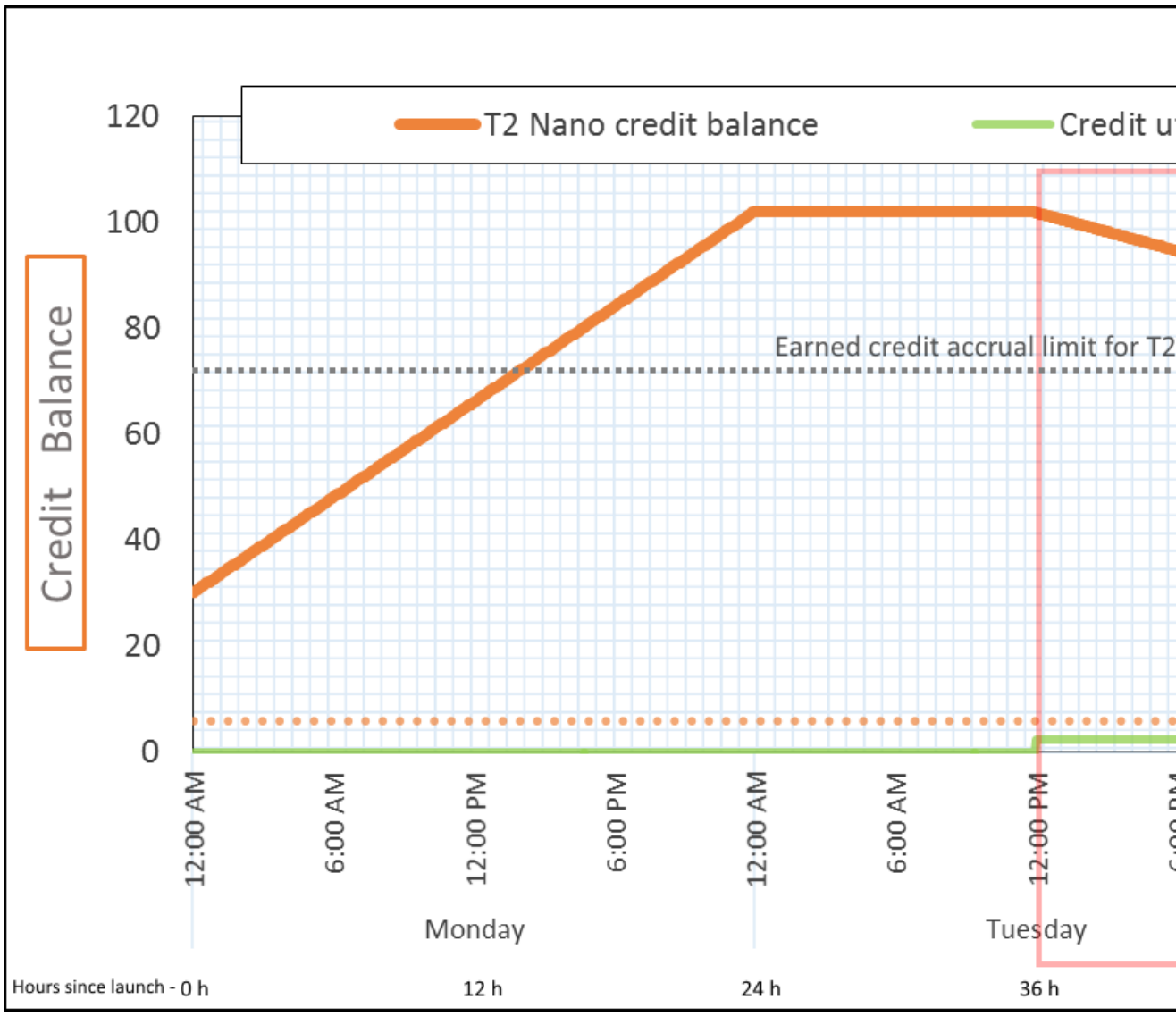
인스턴스가 획득한 것보다 더 많은 크레딧을 사용한 경우 크레딧 밸런스가 감소합니다.

기간 6: 76 – 90시간

다음 15시간 동안 인스턴스는 2% CPU를 사용하며 이는 18 크레딧이 필요합니다. 기간 3 및 4와 동일한 CPU 사용률입니다. 하지만 기간 3에서는 잔고가 감소하고, 기간 4에서는 잔고가 그대로 유지된 반면, 이 기간 동안에는 잔고가 증가합니다.

기간 3에서는 누적된 시작 크레딧이 사용되었고, 크레딧 한도를 초과하는 획득 크레딧은 모두 버려졌기 때문에 크레딧 밸런스가 감소했습니다. 기간 4에서는 인스턴스가 획득한 것보다 더 적은 크레딧을 사용했습니다. 한도를 초과하는 획득한 크레딧은 폐기되고, 잔고는 최대 72 크레딧으로 유지됩니다.

이 기간에는 누적된 시작 크레딧이 없고 잔고에 누적된 획득 크레딧이 한도보다 적습니다. 획득된 크레딧이 버려지지 않습니다. 또한 인스턴스는 사용한 것보다 더 많은 크레딧을 획득하므로 크레딧 밸런스가 증가합니다.



크레딧 사용률	24시간당 28.8 크레딧(시간당 1.2 크레딧, 2% CPU 사용률, 크레딧 획득률 40%) &endash; 15시간 동안 18 크레딧—
크레딧 획득률	24시간당 72 크레딧(15시간 동안 45 크레딧)
크레딧 폐기율	24시간당 0 크레딧
크레딧 밸런스	72 크레딧(24시간당 잔고 증가율 43.2 크레딧 &endash;— 변화율 = 사용률 28.8/24 + 획득률 72/24)

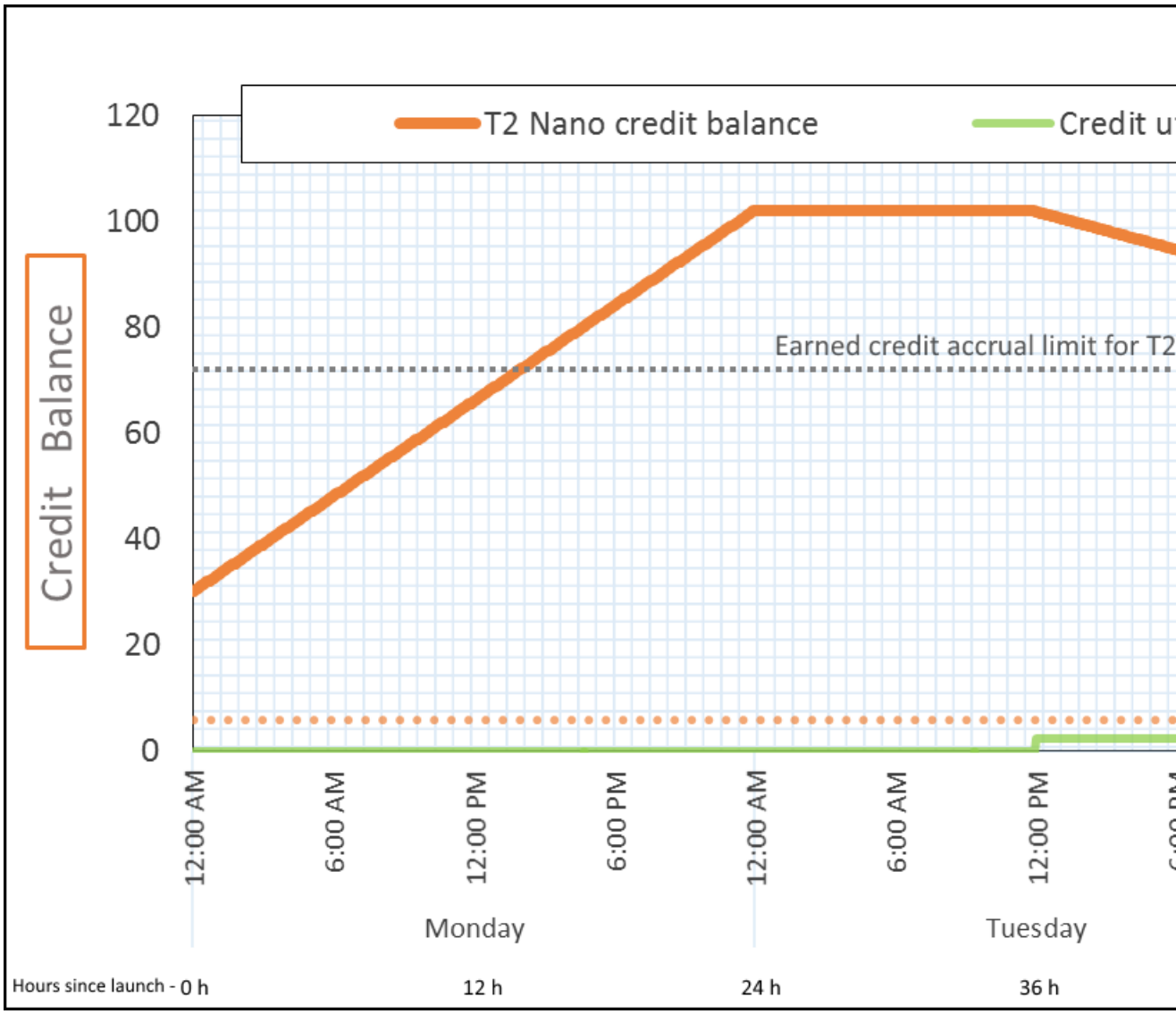
결론

인스턴스가 획득한 것보다 적은 크레딧을 사용한 경우 크레딧 밸런스가 증가합니다.

기간 7: 91 – 96시간

다음 6시간 동안 인스턴스는 유휴 —상태로 유지되어 —CPU 사용률이 0%이므로 크레딧이 사용되지 않습니다. 기간 2의 CPU 사용률과 동일하지만 잔고는 102 크레딧으로 유지되지 않고 인스턴스 크레딧 밸런스 한도인— 72 크레딧으로 유지됩니다.

기간 2에서 크레딧 밸런스에는 누적된 시작 크레딧 30개가 포함됩니다. 기간 3에서는 시작 크레딧이 사용되었습니다. 실행 중 인스턴스는 더 이상 시작 크레딧을 받을 수 없습니다. 크레딧 밸런스 한도에 도달한 후, 획득 크레딧 중 한도를 초과하는 크레딧은 버려집니다.



크레딧 사용률	24시간당 0 크레딧(CPU 사용률 0%)
크레딧 획득률	24시간당 72 크레딧
크레딧 폐기율	24시간당 72 크레딧(크레딧 획득률 100%)
크레딧 밸런스	72 크레딧(시작 크레딧 0, 획득 크레딧 72)

결론

인스턴스는 크레딧을 계속 획득하지만 크레딧 밸런스 한도에 도달하면 획득 크레딧을 더 이상 누적할 수 없습니다. 한도에 도달한 후 새로 획득한 크레딧은 버려집니다. 크레딧 밸런스 한도는 인스턴스가 24시간만에 획득할 수 있는 크레딧 수에 따라 결정됩니다. 크레딧 밸런스 한도에 대한 자세한 내용은 크레딧 표 (p. 195)를 참조하십시오.

성능 순간 확장 가능 인스턴스의 작업

이러한 인스턴스 시작, 모니터링 및 수정 절차는 서로 유사합니다. 주요 차이점은 인스턴스가 시작할 때 기본 적용되는 크레딧 사양입니다. 기본 크레딧 사양을 변경하지 않을 경우 기본값은 다음과 같습니다.

- T3 및 T3a 인스턴스는 `unlimited`로 시작하도록 기본 설정되어 있습니다.
- T2 인스턴스는 `standard`로 시작하도록 기본 설정되어 있습니다.

내용

- 무제한 또는 스탠다드로 성능 순간 확장 가능 인스턴스 시작 (p. 220)
- Auto Scaling 그룹을 사용하여 성능 순간 확장 가능 인스턴스를 무제한으로 시작 (p. 221)
- 성능 순간 확장 가능 인스턴스의 크레딧 사양 보기 (p. 222)
- 성능 순간 확장 가능 인스턴스의 크레딧 사양 수정 (p. 222)
- 계정의 기본 크레딧 사양 설정 (p. 223)
- 기본 크레딧 사양 보기 (p. 224)

무제한 또는 스탠다드로 성능 순간 확장 가능 인스턴스 시작

T3 및 T3a 인스턴스는 `unlimited`로 시작하도록 기본 설정되어 있습니다. T2 인스턴스는 `standard`로 시작하도록 기본 설정되어 있습니다.

이러한 인스턴스에 대한 AMI 및 드라이버 요구 사항에 대한 자세한 내용은 출시 정보 (p. 193) 단원을 참조하십시오.

Amazon EBS 볼륨을 루트 디바이스로 사용하여 인스턴스를 시작해야 합니다. 자세한 내용은 Amazon EC2 루트 디바이스 볼륨 (p. 16) 단원을 참조하십시오.

Amazon EC2 콘솔, AWS SDK, 명령줄 도구 또는 Auto Scaling 그룹을 사용하여 인스턴스를 `unlimited` 또는 `standard`로 시작할 수 있습니다. 자세한 내용은 Auto Scaling 그룹을 사용하여 성능 순간 확장 가능 인스턴스를 무제한으로 시작 (p. 221) 단원을 참조하십시오.

무제한 또는 스탠다드로 성능 순간 확장 가능 인스턴스를 시작하려면(콘솔)

1. 인스턴스 시작 마법사를 사용하여 인스턴스 시작 (p. 431)의 절차를 따르십시오.
2. Choose an Instance Type(인스턴스 유형 선택) 페이지에서 인스턴스 유형을 선택하고 Next: Configure Instance Details(다음: 인스턴스 정보 구성)를 선택합니다.
3. 다음과 같이 크레딧 사양을 선택합니다. T3 및 T3a는 기본적으로 `unlimited`이고, T2는 `standard`입니다.
   a. T3 또는 T3a 인스턴스를 `standard`로 시작하려면 Configure Instance Details(인스턴스 정보 구성) 페이지의 T2/T3 무제한에 대해 활성화를 선택 취소합니다.
   b. T2 인스턴스를 `unlimited`로 시작하려면 인스턴스 세부 정보 구성 페이지의 T2/T3 무제한(T2/T3 Unlimited)에서 활성화를 선택합니다.
4. 마법사에 표시되는 지침에 따라 계속합니다. 인스턴스 시작 검토 페이지에서 옵션 검토를 마쳤으면 시작을 선택합니다. 자세한 내용은 인스턴스 시작 마법사를 사용하여 인스턴스 시작 (p. 431) 단원을 참조하십시오.

무제한 또는 스탠다드로 성능 순간 확장 가능 인스턴스를 시작하려면(AWS CLI)

run-instances 명령을 사용하여 인스턴스를 시작합니다. `--credit-specification CpuCredits=` 파라미터를 사용하여 크레딧 사양을 지정합니다. 유효한 크레딧 사양은 `unlimited` 및 `standard`입니다.

- T3 및 T3a의 경우 `--credit-specification` 파라미터를 포함하지 않으면 인스턴스가 `unlimited`로 시작되도록 기본 설정되어 있습니다.

- T2의 경우 `--credit-specification` 파라미터를 포함하지 않으면 인스턴스가 `standard`로 시작되도록 기본 설정되어 있습니다.

```
aws ec2 run-instances --image-id ami-abc12345 --count 1 --instance-type t3.micro --key-
name MyKeyPair --credit-specification "CpuCredits=unlimited"
```

Auto Scaling 그룹을 사용하여 성능 순간 확장 가능 인스턴스를 무제한으로 시작

성능 확장 가능 인스턴스가 시작되면 CPU 크레딧 없이도 좋은 부트스트래핑 환경을 경험할 수 있습니다. Auto Scaling 그룹을 사용하여 인스턴스를 시작하는 경우 인스턴스를 `unlimited`로 구성합니다. 그러한 경우 인스턴스는 Auto Scaling 그룹에서 자동으로 시작 또는 재시작될 때 잉여 크레딧을 사용합니다. 잉여 크레딧을 사용하면 성능 제한을 막을 수 있습니다.

시작 템플릿 생성

Auto Scaling 그룹에서 인스턴스를 `unlimited`로 시작하는 데 시작 템플릿을 사용해야 합니다. 시작 구성에서는 인스턴스를 `unlimited`로 시작하는 것은 지원하지 않습니다.

인스턴스를 무제한으로 시작하는 시작 템플릿을 생성하려면(콘솔)

1. Auto Scaling 그룹에 대한 시작 템플릿 생성 절차를 수행하십시오.
2. 시작 템플릿 내용의 인스턴스 유형에서 T3, T3a 또는 T2 인스턴스 크기를 선택합니다.
3. Auto Scaling 그룹에서 인스턴스를 `unlimited`로 시작하려면 고급 세부 정보의 T2/T3 무제한에서 활성화를 선택합니다.
4. 시작 템플릿 파라미터 정의를 완료한 경우 시작 템플릿 생성을 선택합니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Auto Scaling 그룹에 대한 시작 템플릿 생성을 참조하십시오.

인스턴스를 무제한으로 시작하는 시작 템플릿을 생성하려면(AWS CLI)

create-launch-template 명령을 사용하고 `unlimited`를 크레딧 사양으로 지정합니다.

- T3 및 T3a의 경우 `CreditSpecification={CpuCredits=unlimited}` 값을 포함하지 않으면 인스턴스가 `unlimited`로 시작되도록 기본 설정되어 있습니다.
- T2의 경우 `CreditSpecification={CpuCredits=unlimited}` 값을 포함하지 않으면 인스턴스가 `standard`로 시작되도록 기본 설정되어 있습니다.

```
aws ec2 create-launch-template --launch-template-name MyLaunchTemplate
 --version-description FirstVersion --launch-template-data
 ImageId=ami-8c1be5f6,InstanceType=t3.medium,CreditSpecification={CpuCredits=unlimited}
```

Auto Scaling 그룹을 시작 템플릿에 연결

Auto Scaling 그룹에 시작 템플릿을 연결하려면 시작 템플릿을 사용하여 Auto Scaling 그룹을 생성하거나 기존 Auto Scaling 그룹에 시작 템플릿을 추가합니다.

시작 템플릿을 사용하여 Auto Scaling 그룹을 생성하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 화면 상단의 탐색 모음에서 시작 템플릿을 만들 때 사용한 리전과 동일한 리전을 선택합니다.
3. 탐색 창에서 Auto Scaling 그룹을 선택하고 Auto Scaling 그룹 생성을 선택합니다.
4. 시작 템플릿을 선택하고 시작 템플릿을 선택한 후 다음 단계를 선택합니다.
5. Auto Scaling 그룹 관련 필드를 작성합니다. 검토 페이지에서 구성 설정 검토를 마쳤으면 Auto Scaling 그룹 생성을 선택합니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 시작 템플릿을 사용한 Auto Scaling 그룹 생성 을 참조하십시오.

시작 템플릿을 사용하여 Auto Scaling 그룹을 생성하려면(AWS CLI)

create-auto-scaling-group AWS CLI 명령을 사용하여 `--launch-template` 파라미터를 지정합니다.

기존 Auto Scaling 그룹에 시작 템플릿을 추가하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 화면 상단의 탐색 모음에서 시작 템플릿을 만들 때 사용한 리전과 동일한 리전을 선택합니다.
3. 탐색 창에서 Auto Scaling 그룹을 선택합니다.
4. Auto Scaling 그룹 목록에서 Auto Scaling 그룹을 선택하고 작업, 편집을 선택합니다.
5. 세부 정보 탭의 시작 템플릿에서 시작 템플릿을 선택한 다음, 저장을 선택합니다.

기존 Auto Scaling 그룹에 시작 템플릿을 추가하려면(AWS CLI)

update-auto-scaling-group AWS CLI 명령을 사용하여 `--launch-template` 파라미터를 지정합니다.

성능 순간 확장 가능 인스턴스의 크레딧 사양 보기

실행 중이거나 중지된 인스턴스의 크레딧 사양(`unlimited` 또는 `standard`)을 확인할 수 있습니다.

순간 확장 가능 인스턴스의 크레딧 사양을 보려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 인스턴스를 선택하고 인스턴스를 선택합니다.
3. 설명을 선택하고 T2/T3 무제한(T2/T3 Unlimited) 필드를 확인합니다.
   - 값이 `Enabled`이면 인스턴스가 `unlimited`로 구성됩니다.
   - 값이 `Disabled`이면 인스턴스가 `standard`로 구성됩니다.

성능 순간 확장 가능 인스턴스의 크레딧 사양을 보려면(AWS CLI)

describe-instance-credit-specifications 명령을 사용합니다. 하나 이상의 인스턴스 ID를 지정하지 않은 경우 크레딧 사양이 `unlimited`인 모든 인스턴스가 반환되고 이전에 `unlimited` 크레딧 사양으로 구성된 인스턴스 또한 반환됩니다. 예를 들어 T3 인스턴스가 `unlimited`로 구성된 가운데 이를 M4 인스턴스로 크기 조정하는 경우 Amazon EC2에서 M4 인스턴스를 반환합니다.

Example

```
aws ec2 describe-instance-credit-specifications --instance-id i-1234567890abcdef0
```

다음은 예제 출력입니다.

```
{
  "InstanceCreditSpecifications": [
    {
        "InstanceId": "i-1234567890abcdef0",
        "CpuCredits": "unlimited"
    }
  ]
}
```

성능 순간 확장 가능 인스턴스의 크레딧 사양 수정

실행 중이거나 중지된 인스턴스의 크레딧 사양을 `unlimited`와 `standard` 간에 언제든지 전환할 수 있습니다.

버스트 가능한 성능 인스턴스의 크레딧 사양을 수정하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 인스턴스를 선택하고 인스턴스를 선택합니다. 몇 가지 인스턴스에 대한 크레딧 사양을 한 번에 수정하려면 해당되는 인스턴스를 모두 선택합니다.
3. 작업, 인스턴스 설정, T2/T3 무제한 변경(Change T2/T3 Unlimited)을 선택합니다.

   Note

   T2/T3 무제한 변경 옵션은 T3, T3a 또는 T2 인스턴스를 선택한 경우에만 활성화됩니다.
4. 크레딧 사양을 `unlimited`로 변경하려면 활성화를 선택합니다. 크레딧 사양을 `standard`로 변경하려면 비활성화를 선택합니다. 인스턴스에 대한 현재 크레딧 사양은 인스턴스 ID 뒤의 괄호에 표시됩니다.

버스트 가능한 성능 인스턴스의 크레딧 사양을 수정하려면(AWS CLI)

modify-instance-credit-specification 명령을 사용합니다. `--instance-credit-specification` 파라미터를 사용하여 인스턴스 및 크레딧 사양을 지정합니다. 유효한 크레딧 사양은 `unlimited` 및 `standard`입니다.

Example

```
aws ec2 modify-instance-credit-specification --region us-east-1 --instance-credit-
specification "InstanceId=i-1234567890abcdef0,CpuCredits=unlimited"
```

다음은 예제 출력입니다.

```
{
  "SuccessfulInstanceCreditSpecifications": [
    {
      "InstanceId": "i- 1234567890abcdef0"
    }
  ],
  "UnsuccessfulInstanceCreditSpecifications": []
}
```

계정의 기본 크레딧 사양 설정

AWS 리전별 계정 수준에서 기본 크레딧 사양을 설정할 수 있습니다. 인스턴스 패밀리별로 기본 크레딧 사양을 지정합니다(T2, T3 또는 T3a).

AWS Management 콘솔에서 인스턴스 시작 마법사를 사용하여 인스턴스를 시작하면 T2/T3 Unlimited(T2/T3 무제한)의 값이 계정 수준 기본 크레딧 사양을 재정의합니다. AWS CLI를 사용하여 인스턴스를 시작하면 계정에 속한 새 성능 순간 확장 가능 인스턴스가 모두 기본 크레딧 옵션을 사용하여 시작됩니다. 실행 중이거나 중지된 기존 인스턴스의 크레딧 사양은 영향을 받지 않습니다.

`modify-default-credit-specification` API는 AWS 리전 수준에서 작동하며 각 가용 영역의 크레딧 옵션을 수정하는 비동기 작업입니다. 5분 이내에 리전의 모든 영역이 업데이트됩니다. 하지만 이 작업 중에 인스턴스가 시작되면 영역이 업데이트될 때까지 인스턴스가 새 크레딧 옵션을 가져오지 못할 수도 있습니다. 업데이트가 발생했는지 확인하려면 `get-default-credit-specification`을 호출하고 기본 크레딧 사양에서 업데이트를 확인하면 됩니다. 자세한 내용은 기본 크레딧 사양 보기 (p. 224) 단원을 참조하십시오.

Note

인스턴스 패밀리의 기본 크레딧 사양은 5분 동안 한 번만 수정할 수 있으며, 24시간 동안 4회까지 수정할 수 있습니다.

계정 수준에서 기본 크레딧 사양을 설정하려면(AWS CLI)

modify-default-credit-specification 명령을 사용하십시오. `--cpu-credits` 파라미터를 사용하여 AWS 리전, 인스턴스 패밀리 및 기본 크레딧 사양을 지정합니다. 유효한 기본 크레딧 사양은 `unlimited` 및 `standard`입니다.

```
aws ec2 modify-default-credit-specification --region us-east-1 --instance-family t2 --cpu-credits unlimited
```

기본 크레딧 사양 보기

AWS 리전별로 계정 수준에서 성능 순간 확장 가능 인스턴스 패밀리의 기본 크레딧 사양을 볼 수 있습니다.

계정 수준에서 기본 크레딧 사양을 보려면(AWS CLI)

get-default-credit-specification 명령을 사용하십시오. AWS 리전과 인스턴스 패밀리를 지정합니다.

```
aws ec2 get-default-credit-specification --region us-east-1 --instance-family t2
```

CPU 크레딧 모니터링

각 인스턴스의 크레딧 밸런스를 CloudWatch 콘솔의 Amazon EC2 인스턴스별 측정치로 확인할 수 있습니다.

주제

- 성능 순간 확장 가능 인스턴스에 대한 추가 CloudWatch 측정치 (p. 224)
- CPU 크레딧 사용량 계산 (p. 225)

성능 순간 확장 가능 인스턴스에 대한 추가 CloudWatch 측정치

T3, T3a 및 T2 인스턴스에는 다음과 같은 추가 CloudWatch 측정치가 있으며 5분마다 업데이트됩니다.

- `CPUCreditUsage` – 측정 기간 중에 소비한 CPU 크레딧 수.
- `CPUCreditBalance` – 한 인스턴스에서 발생한 CPU 크레딧 수입니다. CPU에서 버스트가 발생하고 CPU 크레딧이 획득 속도보다 빠르게 소비될 때 크레딧 밸런스가 고갈됩니다.
- `CPUSurplusCreditBalance` – `CPUCreditBalance` 값이 0일 때 CPU 성능을 유지하기 위해 소비되는 잉여 CPU 크레딧 수.
- `CPUSurplusCreditsCharged` – 24시간 동안 획득할 수 있는 최대 CPU 크레딧 수 (p. 195)를 초과하여 추가 요금을 유발하는 잉여 CPU 크레딧 수.

마지막 두 측정치는 `unlimited`로 구성된 인스턴스에만 적용됩니다.

다음 표에서는 성능 순간 확장 가능 인스턴스에 대한 CloudWatch 측정치를 설명합니다. 자세한 내용은 인스턴스에 대해 얻을 수 있는 CloudWatch 측정치 나열 (p. 617) 단원을 참조하십시오.

지표	설명
`CPUCreditUsage`	CPU 사용률을 위해 인스턴스에서 소비되는 CPU 크레딧의 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 `Sum` 통계 대신 `Average` 통계를 사용하십시오.

지표	설명
	단위: 크레딧 (vCPU-분)
`CPUCreditBalance`	시작 이후 인스턴스가 누적한 획득 CPU 크레딧 수입니다. T2 스탠다드의 경우 `CPUCreditBalance`에 누적된 시작 크레딧 수도 포함됩니다. 크레딧은 획득 이후에 크레딧 밸런스에 누적되고, 소비 시 크레딧 밸런스에서 소멸됩니다. 크레딧 밸런스는 최대 한도(인스턴스 크기에 따라 결정)가 있습니다. 한도에 도달하면 새로 획득한 크레딧이 모두 삭제됩니다. T2 스탠다드의 경우 시작 크레딧은 한도에 포함되지 않습니다. `CPUCreditBalance`의 크레딧은 인스턴스가 기준 CPU 사용률 이상으로 버스터를 하는 데 소비할 수 있습니다. 인스턴스가 실행 중인 동안 `CPUCreditBalance`의 크레딧은 만료되지 않습니다. T3 또는 T3a 인스턴스가 중지되면 `CPUCreditBalance` 값은 7일 동안 지속됩니다. 그 이후에는 누적된 크레딧이 모두 삭제됩니다. T2 인스턴스가 중지되면 `CPUCreditBalance` 값은 지속되지 않고 누적된 크레딧이 모두 삭제됩니다. CPU 크레딧 측정치는 5분 간격으로만 제공됩니다. 단위: 크레딧 (vCPU-분)
`CPUSurplusCreditBalance`	`CPUCreditBalance` 값이 0일 때 `unlimited` 인스턴스에서 소비된 잉여 크레딧의 수 입니다. 획득한 CPU 크레딧에 따라 `CPUSurplusCreditBalance` 값이 청산됩니다. 잉여 크레딧의 수가 인스턴스가 24시간 동안 획득할 수 있는 최대 크레딧 수를 초과한 경우 최대 값 이상으로 소비된 잉여 크레딧은 추가 요금으로 부과됩니다. 단위: 크레딧 (vCPU-분)
`CPUSurplusCreditsCharged`	획득한 CPU 크레딧으로 청산되지 않는 소비 잉여 크레딧의 수로, 추가 요금으로 부과됩니다. 소비된 잉여 크레딧은 다음이 발생할 때 요금이 부과됩니다. • 소비한 잉여 크레딧이 인스턴스가 24시간 동안 획득할 수 있는 최대 크레딧 수를 초과하는 경우. 해당 시간이 끝날 때 최대 값 이상으로 소비한 잉여 크레딧에 요금이 부과됩니다. • 인스턴스가 중지 또는 종료된 경우. • 인스턴스가 `unlimited`에서 `standard`로 전환됩니다. 단위: 크레딧 (vCPU-분)

CPU 크레딧 사용량 계산

인스턴스의 CPU 크레딧 사용량은 앞 표에 설명되어 있는 인스턴스 CloudWatch 측정치를 사용해 계산됩니다.

Amazon EC2는 5분마다 CloudWatch에 지표를 전송합니다. 어떤 시점에서든 지표의 이전 값을 참조하여 5분 전에 전송된 지표의 이전 값을 알 수 있습니다.

스탠다드에서 CPU 크레딧 사용량 계산

- CPU 사용률이 기준 미만이고 소비된 크레딧이 5분 전에 획득한 크레딧보다 적을 때 CPU 크레딧 밸런스가 증가합니다.
- CPU 사용률이 기준 이상이고 소비된 크레딧이 5분 전에 획득한 크레딧보다 많을 때 CPU 크레딧 밸런스가 감소합니다.

수학적으로 다음 수식을 통해 이를 확인할 수 있습니다:

Example

```
CPUCreditBalance = prior CPUCreditBalance + [Credits earned per hour * (5/60) -
 CPUCreditUsage]
```

인스턴스 크기에 따라 인스턴스가 시간당 획득할 수 있는 크레딧 수와 크레딧 밸런스에 누적할 수 있는 획득 크레딧의 수가 결정됩니다. 시간당 획득 크레딧 수, 각 인스턴스 크기에 대한 크레딧 밸런스 한도에 대한 자세한 내용은 크레딧 표 (p. 195)를 참조하십시오.

예

이 예제에서는 `t3.nano` 인스턴스를 사용합니다. 인스턴스의 `CPUCreditBalance` 값을 계산하려면 앞의 수식을 사용하여 다음과 같이 합니다.

- `CPUCreditBalance` – 계산하려는 현재 크레딧 밸런스입니다.
- `prior CPUCreditBalance` – 5분 전의 크레딧 밸런스입니다. 이 예제에서는 인스턴스가 2개의 크레딧을 획득했습니다.
- `Credits earned per hour` – `t3.nano` 인스턴스는 시간당 6개의 크레딧을 획득합니다.
- `5/60` – CloudWatch 지표가 게시되는 5분 간격을 나타냅니다. 시간당 획득한 크레딧에 5/60(5분)을 곱해 인스턴스가 이전 5분 동안 획득한 크레딧 수를 계산합니다. `t3.nano` 인스턴스는 5분마다 0.5개 크레딧을 획득합니다.
- `CPUCreditUsage` – 이전 5분 동안 소비된 인스턴스의 크레딧 수입니다. 이 예제에서는 인스턴스가 이전 5분 동안 크레딧 1개를 소비했습니다.

이러한 값을 사용하여 `CPUCreditBalance` 값을 계산할 수 있습니다.

Example

```
CPUCreditBalance = 2 + [0.5 - 1] = 1.5
```

무제한 인스턴스에 대한 CPU 크레딧 사용량 계산

T3, T3a 또는 T2 인스턴스가 기준 성능 이상으로 버스트해야 할 때는 잉여 크레딧을 소비하기 전에 항상 누적 크레딧을 소비합니다. 획득한 CPU 크레딧 밸런스가 감소하면 필요한 시간만큼 잉여 크레딧을 소비하여 버스트를 할 수 있습니다. 인스턴스의 CPU 사용률이 기준 미만으로 떨어지면 인스턴스가 크레딧을 획득하기 전에 잉여 크레딧이 항상 먼저 청산됩니다.

5분 간격으로 발생하는 활동을 반영하기 위해 다음 수식에서 `Adjusted balance`라는 용어를 사용하고 있습니다. `CPUCreditBalance` 및 `CPUSurplusCreditBalance` CloudWatch 지표의 값에 도달하기 위해 이 값을 사용합니다.

Example

```
Adjusted balance = [prior CPUCreditBalance - prior CPUSurplusCreditBalance] + [Credits
 earned per hour * (5/60) - CPUCreditUsage]
```

0의 값이 Adjusted balance이면 인스턴스는 버스트에 획득한 모든 크레딧을 소비했으며 잉여 크레딧은 소비되지 않았다는 뜻입니다. 그 결과 CPUCreditBalance와 CPUSurplusCreditBalance가 모두 0으로 설정됩니다.

Adjusted balance 값이 양수이면 인스턴스가 크레딧을 획득했고 이전의 잉여 크레딧(존재할 경우)이 청산되었다는 뜻입니다. 그 결과 Adjusted balance 값이 CPUCreditBalance로 지정되고 CPUSurplusCreditBalance가 0으로 설정됩니다. 누적할 수 있는 최대 크레딧 수 (p. 195)는 인스턴스 크기에 따라 결정됩니다.

Example

```
CPUCreditBalance = min [max earned credit balance, Adjusted balance]
CPUSurplusCreditBalance = 0
```

Adjusted balance 값이 음수면 인스턴스가 누적한 모든 크레딧을 소비했고 버스트에 잉여 크레딧이 소비되었다는 뜻입니다. 그 결과 Adjusted balance 값이 CPUSurplusCreditBalance로 지정되고 CPUCreditBalance가 0으로 설정됩니다. 즉 누적할 수 있는 최대 크레딧 수 (p. 195)는 인스턴스 크기에 따라 결정됩니다.

Example

```
CPUSurplusCreditBalance = min [max earned credit balance, -Adjusted balance]
CPUCreditBalance = 0
```

소비된 잉여 크레딧이 인스턴스가 누적할 수 있는 최대 크레딧을 초과하면 이전 수식에서와 같이 잉여 크레딧 밸런스가 최대 값으로 설정됩니다. 나머지 잉여 크레딧은 CPUSurplusCreditsCharged 측정치로 표현되어 요금이 부과됩니다.

Example

```
CPUSurplusCreditsCharged = max [-Adjusted balance - max earned credit balance, 0]
```

마지막으로 인스턴스가 종료하면 CPUSurplusCreditBalance로 추적된 모든 잉여 크레딧에 요금이 부과됩니다. 인스턴스가 unlimited에서 standard로 전환되면 나머지 모든 CPUSurplusCreditBalance에도 요금이 부과됩니다.

컴퓨팅 최적화 인스턴스

컴퓨팅 최적화 인스턴스는 고성능 프로세서의 이점을 활용하는 컴퓨팅 집약적 애플리케이션에 적합합니다. 컴퓨팅 최적화 인스턴스는 다음 애플리케이션에 매우 적합합니다.

- 일괄 처리 작업
- 미디어 트랜스코딩
- 고성능 웹 서버
- 고성능 컴퓨팅(HPC)
- 과학 모델링
- 전용 게임 서버, 광고 서비스 엔진
- 기계 학습 추론 및 기타 컴퓨팅 집약적 애플리케이션

자세한 내용은 Amazon EC2 C5 인스턴스를 참조하십시오.

내용

- 하드웨어 사양 (p. 228)

- 인스턴스 성능 (p. 229)
- 네트워크 성능 (p. 229)
- SSD I/O 성능 (p. 230)
- 인스턴스 기능 (p. 231)
- 출시 정보 (p. 231)

하드웨어 사양

다음은 컴퓨팅 최적화 인스턴스용 하드웨어 사양을 요약한 것입니다.

인스턴스 유형	기본 vCPU	메모리(GiB)
`c4.large`	2	3.75
`c4.xlarge`	4	7.5
`c4.2xlarge`	8	15
`c4.4xlarge`	16	30
`c4.8xlarge`	36	60
`c5.large`	2	4
`c5.xlarge`	4	8
`c5.2xlarge`	8	16
`c5.4xlarge`	16	32
`c5.9xlarge`	36	72
`c5.12xlarge`	48	96
`c5.18xlarge`	72	144
`c5.24large`	96	192
`c5.metal`	96	192
`c5d.large`	2	4
`c5d.xlarge`	4	8
`c5d.2xlarge`	8	16
`c5d.4xlarge`	16	32
`c5d.9xlarge`	36	72
`c5d.12xlarge`	48	96
`c5d.18xlarge`	72	144
`c5d.24large`	96	192
`c5d.metal`	96	192
`c5n.large`	2	5.25

인스턴스 유형	기본 vCPU	메모리(GiB)
`c5n.xlarge`	4	10.5
`c5n.2xlarge`	8	21
`c5n.4xlarge`	16	42
`c5n.9xlarge`	36	96
`c5n.18xlarge`	72	192
`c5n.metal`	72	192

Amazon EC2 인스턴스 유형별 하드웨어 사양에 대한 자세한 내용은 Amazon EC2 인스턴스 유형을 참조하십시오.

CPU 옵션 지정에 대한 자세한 내용은 CPU 옵션 최적화 (p. 546) 단원을 참조하십시오.

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. 일부 컴퓨팅 최적화 인스턴스는 추가 비용 없이도 EBS에 최적화할 수 있게 기본 설정되어 있습니다. 자세한 내용은 Amazon EBS 최적화 인스턴스 (p. 984) 단원을 참조하십시오.

일부 컴퓨팅 최적화 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. 537) 단원을 참조하십시오.

네트워크 성능

지원되는 인스턴스 유형에서 향상된 네트워킹 기능을 활성화할 수 있습니다. 향상된 네트워킹을 통해 PPS(Packet Per Second) 성능이 크게 높아지고, 네트워크 지터 및 지연 시간이 낮아집니다. 자세한 내용은 Linux에서 향상된 네트워킹 (p. 706) 단원을 참조하십시오.

향상된 네트워킹을 지원하는 데 ENA(Elastic Network Adapter)를 사용하는 인스턴스 유형은 일관되게 낮은 지연 시간과 함께 높은 초당 패킷 성능을 제공합니다. 대부분의 애플리케이션은 항시 높은 수준의 네트워크 성능을 필요로 하지 않지만, 데이터를 주고 받을 때 넓은 대역폭에 액세스할 수 있을 경우 유익할 수 있습니다. ENA를 사용하고 "최대 10Gbps" 또는 "최대 25Gbps"의 네트워크 성능으로 기록된 인스턴스 크기는 네트워크 I/O 크레딧 메커니즘을 사용하여 평균 대역폭 이용률에 따라 네트워크 대역폭을 인스턴스에 할당합니다. 이러한 인스턴스의 네트워크 대역폭이 기준 한도 미만으로 떨어지면 크레딧이 발생하는데, 이 크레딧은 네트워크 데이터를 전송할 때 사용할 수 있습니다.

다음은 향상된 네트워킹을 지원하는 컴퓨팅 최적화 인스턴스용 네트워크 성능을 요약한 것입니다.

인스턴스 유형	네트워크 성능	향상된 네트워킹
`c5.4xlarge` 이하 | `c5d.4xlarge` 이하	최대 10Gbps	ENA (p. 707)
`c5.9xlarge` | `c5d.9xlarge`	10Gbps	ENA (p. 707)
`c5.12xlarge` | `c5d.12xlarge`	12Gbps	ENA (p. 707)
`c5n.4xlarge` 이하	최대 25Gbps	ENA (p. 707)

인스턴스 유형	네트워크 성능	향상된 네트워킹
`c5.18xlarge` | `c5.24xlarge` | `c5.metal` | `c5d.18xlarge` | `c5d.24xlarge` | `c5d.metal`	25Gbps	ENA (p. 707)
`c5n.9xlarge`	50Gbps	ENA (p. 707)
`c5n.18xlarge` | `c5n.metal`	100Gbps	ENA (p. 707)
`c4.large`	보통	intel 82599 VF (p. 719)
`c4.xlarge` | `c4.2xlarge` | `c4.4xlarge`	높음	intel 82599 VF (p. 719)
`c4.8xlarge`	10Gbps	intel 82599 VF (p. 719)

SSD I/O 성능

커널 버전이 4.4 이상인 Linux AMI를 사용하고 인스턴스에서 사용 가능한 모든 SSD 기반의 인스턴스 스토어 볼륨을 활용하는 경우, 다음 표와 같은 IOPS(블록 크기 4,096바이트) 성능을 얻을 수 있습니다(대기열 깊이 포화 상태에서). 그렇지 않으면 IOPS 성능이 더 낮아집니다.

인스턴스 크기	100% 임의 읽기 IOPS	IOPS 쓰기
`c5d.large` *	20,000건	9,000
`c5d.xlarge` *	40,000	18,000
`c5d.2xlarge` *	80,000	37,000
`c5d.4xlarge` *	175,000	75,000
`c5d.9xlarge`	350,000	170,000
`c5d.12xlarge`	700,000	340,000
`c5d.18xlarge`	700,000	340,000
`c5d.24xlarge`	1,400,000	680,000
`c5d.metal`	1,400,000	680,000

* 이러한 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 전략은 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. 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 않음을 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. 1037) 단원을 참조하십시오.

인스턴스 기능

컴퓨팅 최적화 인스턴스에 대한 기능은 다음과 같이 간략히 설명할 수 있습니다.

	EBS 전용	NVMe EBS	인스턴스 스토어	배치 그룹
C4	예	아니요	아니요	예
C5	예	예	아니요	예
C5d	아니요	예	NVMe *	예
C5n	예	예	아니요	예

* 루트 디바이스 볼륨은 Amazon EBS 볼륨이어야 합니다.

자세한 내용은 다음 단원을 참조하십시오.

- Linux 인스턴스의 Amazon EBS 및 NVMe (p. 981)
- Amazon EC2 인스턴스 스토어 (p. 1027)
- 배치 그룹 (p. 757)

출시 정보

- C5 및 C5d 인스턴스는 1세대(Skylake-SP) 또는 2세대(Cascade Lake)의 3.1 GHz Intel Xeon Platinum 8000 시리즈 프로세서를 사용합니다.
- C4, C5, C5d 및 C5n 인스턴스에는 64비트 EBS 지원 HVM AMIs가 필요합니다. 이들은 고용량 메모리를 보유하는데, 이 용량을 활용하기 위해서는 64비트 운영 체제가 필요합니다. HVM AMI는 고용량 메모리 인스턴스 유형의 반가상화(PV) AMI보다 우수한 성능을 제공합니다. 또한 확장 네트워킹을 활용하려면 HVM AMI를 사용해야 합니다.
- C5, C5d 및 C5n 인스턴스의 요구 사항은 다음과 같습니다.
  - NVMe 드라이버 (p. 981)가 설치되어 있어야 합니다.
  - ENA(Elastic Network Adapter) (p. 707) 드라이버가 설치되어 있어야 합니다.

  다음 Linux AMI는 이러한 요구 사항을 충족합니다.
  - Amazon Linux 2
  - Amazon Linux AMI 2018.03
  - Ubuntu 14.04(`linux-aws` 커널) 이상
  - Red Hat Enterprise Linux 7.4 이상
  - SUSE Linux Enterprise Server 12 SP2 이상
  - CentOS 7.4.1708 이상
  - FreeBSD 11.1 이상

  - Debian GNU/Linux 9 이상
- C5, C5d 및 C5n 인스턴스는 네트워크 인터페이스, EBS 볼륨 및 NVMe 인스턴스 스토어 볼륨을 포함해 최대 28개의 연결을 지원합니다. 모든 인스턴스에는 최소 한 개의 네트워크 인터페이스 연결이 있습니다.
- 베어 메탈 인스턴스를 시작하면 기본 서버가 부팅되는데, 이때 모든 하드웨어 및 펌웨어 구성 요소를 확인합니다. 즉, 인스턴스가 실행 상태가 되어 네트워크를 통해 사용할 수 있게 될 때까지 20분이 걸릴 수 있습니다.
- 베어 메탈 인스턴스에서 EBS 볼륨 또는 보조 네트워크 인터페이스를 연결 또는 분리하려면 PCIe 기본 핫플러그 지원이 필요합니다. Amazon Linux 2 및 최신 Amazon Linux AMI 버전에서는 PCIe 기본 핫플러그를 지원하지만 이전 버전에서는 지원하지 않습니다. 다음 Linux 커널 구성 옵션을 활성화해야 합니다.

```
CONFIG_HOTPLUG_PCI_PCIE=y
CONFIG_PCIEASPM=y
```

- 베어 메탈 인스턴스에서는 I/O 포트 기반 직렬 디바이스가 아닌 PCI 기반 직렬 디바이스를 사용합니다. 업스트림 Linux 커널 및 최신 Amazon Linux AMI에서는 이 디바이스를 지원합니다. 베어 메탈 인스턴스도 시스템에서 PCI 기반 직렬 디바이스를 자동으로 사용할 수 있게 해주는 ACPI SPCR 테이블을 제공합니다. 최신 Windows AMI에서는 PCI 기반 직렬 디바이스를 자동으로 사용합니다.
- API 요청을 통해 종료가 원활히 이루어지도록 지원하려면 C5, C5d 및 C5n 인스턴스에 acpid가 설치되어 있어야 합니다.
- 한 리전에서 시작할 수 있는 총 인스턴스 수에는 제한이 있으며, 일부 인스턴스 유형에는 또 다른 제한이 있습니다. 자세한 정보는 Amazon EC2에서 실행 가능한 인스턴스 수 단원을 참조하십시오. 한도 증가를 요청하려면 Amazon EC2 인스턴스 요청 양식을 사용하십시오.

메모리 최적화 인스턴스

메모리 최적화 인스턴스는 메모리에서 대규모 데이터를 처리하는 워크로드에 대해 빠른 성능을 제공하도록 설계되었습니다.

R4, R5, R5a, R5ad, R5d, R5dn 및 R5n 인스턴스

이러한 인스턴스는 다음 애플리케이션에 매우 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 intel리전스를 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.

`r5.metal` 및 `r5d.metal` 인스턴스에서는 애플리케이션이 프로세서, 메모리 등 호스트 서버의 물리적 리소스에 직접 액세스할 수 있습니다. 이 인스턴스는 다음과 같은 경우에 적합합니다.

- 가상화된 환경에서 사용할 수 없거나 완전히 지원되지 않는 저수준 하드웨어 기능(예: intel VT)에 액세스해야 하는 워크로드
- 라이선스 또는 지원을 위해 가상화되지 않은 환경이 필요한 애플리케이션

자세한 내용은 Amazon EC2 R5 인스턴스를 참조하십시오.

고용량 메모리 인스턴스

고용량 메모리 인스턴스(`u-6tb1.metal`, `u-9tb1.metal`, `u-12tb1.metal`, `u-18tb1.metal` 및 `u-24tb1.metal`)는 인스턴스당 6TiB, 9TiB, 12TiB, 18TiB 및 24TiB의 메모리를 제공합니다. 이러한 인스턴

스는 SAP HANA의 프로덕션 설치를 포함한 대량의 인 메모리 데이터베이스를 실행하도록 설계되었습니다. 베어 메탈 성능에 호스트 하드웨어에 대한 직접 액세스를 제공합니다.

X1 인스턴스

이러한 인스턴스는 다음 애플리케이션에 매우 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 정보는 AWS 클라우드 기반의 SAP HANA를 참조하십시오.
- Apache Spark나 Presto와 같은 빅데이터 처리 엔진.
- HPC(고성능 컴퓨팅) 애플리케이션.

자세한 내용은 Amazon EC2 X1 인스턴스를 참조하십시오.

X1e 인스턴스

이러한 인스턴스는 다음 애플리케이션에 매우 적합합니다.

- 고성능 데이터베이스.
- SAP HANA와 같은 인 메모리 데이터베이스. 자세한 정보는 AWS 클라우드 기반의 SAP HANA를 참조하십시오.
- 메모리 집약적인 엔터프라이즈 애플리케이션.

자세한 내용은 Amazon EC2 X1e 인스턴스를 참조하십시오.

z1d 인스턴스

이러한 인스턴스는 컴퓨팅 용량과 메모리가 대형이며 다음 애플리케이션에 매우 적합합니다.

- EDA(전자 설계 자동화)
- 관계형 데이터베이스 워크로드

`z1d.metal` 인스턴스에서는 애플리케이션이 프로세서, 메모리 등 호스트 서버의 물리적 리소스에 직접 액세스할 수 있습니다. 이 인스턴스는 다음과 같은 경우에 적합합니다.

- 가상화된 환경에서 사용할 수 없거나 완전히 지원되지 않는 저수준 하드웨어 기능(예: intel VT)에 액세스해야 하는 워크로드
- 라이선스 또는 지원을 위해 가상화되지 않은 환경이 필요한 애플리케이션

자세한 내용은 Amazon EC2 z1d 인스턴스를 참조하십시오.

내용

- 하드웨어 사양 (p. 234)
- 메모리 성능 (p. 236)
- 인스턴스 성능 (p. 236)
- 네트워크 성능 (p. 237)
- SSD I/O 성능 (p. 238)
- 인스턴스 기능 (p. 239)
- 개의 vCPU 지원 (p. 240)

- 출시 정보 (p. 241)

하드웨어 사양

다음은 메모리 최적화 인스턴스용 하드웨어 사양의 요약 설명입니다.

인스턴스 유형	기본 vCPU	메모리(GiB)
r4.large	2	15.25
r4.xlarge	4	30.5
r4.2xlarge	8	61
r4.4xlarge	16	122
r4.8xlarge	32	244
r4.16xlarge	64	488
r5.large	2	16
r5.xlarge	4	32
r5.2xlarge	8	64
r5.4xlarge	16	128
r5.8xlarge	32	256
r5.12xlarge	48	384
r5.16xlarge	64	512
r5.24xlarge	96	768
r5.metal	96	768
r5a.large	2	16
r5a.xlarge	4	32
r5a.2xlarge	8	64
r5a.4xlarge	16	128
r5a.8xlarge	32	256
r5a.12xlarge	48	384
r5a.16xlarge	64	512
r5a.24xlarge	96	768
r5ad.large	2	16
r5ad.xlarge	4	32
r5ad.2xlarge	8	64
r5ad.4xlarge	16	128

인스턴스 유형	기본 vCPU	메모리(GiB)
r5ad.12xlarge	48	384
r5ad.24xlarge	96	768
r5d.large	2	16
r5d.xlarge	4	32
r5d.2xlarge	8	64
r5d.4xlarge	16	128
r5d.8xlarge	32	256
r5d.12xlarge	48	384
r5d.16xlarge	64	512
r5d.24xlarge	96	768
r5d.metal	96	768
r5dn.large	2	16
r5dn.xlarge	4	32
r5dn.2xlarge	8	64
r5dn.4xlarge	16	128
r5dn.8xlarge	32	256
r5dn.12xlarge	48	384
r5dn.16xlarge	64	512
r5dn.24xlarge	96	768
r5n.large	2	16
r5n.xlarge	4	32
r5n.2xlarge	8	64
r5n.4xlarge	16	128
r5n.8xlarge	32	256
r5n.12xlarge	48	384
r5n.16xlarge	64	512
r5n.24xlarge	96	768
u-6tb1.metal	448 *	6,144
u-9tb1.metal	448 *	9,216
u-12tb1.metal	448 *	12,288
u-18tb1.metal	448 *	18,432

인스턴스 유형	기본 vCPU	메모리(GiB)
`u-24tb1.metal`	448 *	24,576
`x1.16xlarge`	64	976
`x1.32xlarge`	128	1,952
`x1e.xlarge`	4	122
`x1e.2xlarge`	8	244
`x1e.4xlarge`	16	488
`x1e.8xlarge`	32	976
`x1e.16xlarge`	64	1,952
`x1e.32xlarge`	128	3,904
`z1d.large`	2	16
`z1d.xlarge`	4	32
`z1d.2xlarge`	8	64
`z1d.3xlarge`	12	96
`z1d.6xlarge`	24	192
`z1d.12xlarge`	48	384
`z1d.metal`	48	384

* 각 논리 프로세서는 224개 코더의 하이퍼스레드입니다.

Amazon EC2 인스턴스 유형별 하드웨어 사양에 대한 자세한 내용은 Amazon EC2 인스턴스 유형을 참조하십시오.

CPU 옵션 지정에 대한 자세한 내용은 CPU 옵션 최적화 (p. 546) 단원을 참조하십시오.

메모리 성능

X1 인스턴스에는 intel확장형 메모리 버퍼가 포함되어 있어, 300GiB/s의 지속 가능 메모리 읽기 대역폭과 140GiB/s의 지속 가능 메모리 쓰기 대역폭을 제공합니다.

메모리 최적화 인스턴스에 사용할 수 있는 RAM 크기에 대한 자세한 내용은 하드웨어 사양 (p. 234) 단원을 참조하십시오.

메모리 최적화 인스턴스는 고용량 메모리를 보유하며, 이 용량을 활용하기 위해 64비트 HVM AMI가 필요합니다. HVM AMI는 메모리 최적화 인스턴스의 반가상화(PV) AMI보다 우수한 성능을 제공합니다. 자세한 내용은 Linux AMI 가상화 유형 (p. 95) 단원을 참조하십시오.

인스턴스 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.

X1e 및 X1 인스턴스는 최대 128개의 vCPU를 사용할 수 있는 것이 특징이며, 인메모리 애플리케이션의 성능을 강화하기 위해 높은 메모리 대역폭과 대용량 L3 캐시를 채용한 intel Xeon E7-8880 v3 프로세서 4개로 작동됩니다.

고용량 메모리 인스턴스(`u-6tb1.metal`, `u-9tb1.metal`, `u-12tb1.metal`)는 미션 크리티컬 엔터프라이즈 워크로드에 최적화된 intel Xeon Platinum 8176M(Skylake) 프로세서로 8소켓 플랫폼에서 작동하는 최초의 인스턴스입니다.

18TB 및 24TB 메모리가 제공되는 대용량 메모리 인스턴스(`u-18tb1.metal`, `u-24tb1.metal`)는 2세대 인텔 제온 확장형 8280L(Cascade Lake) 프로세서가 탑재된 8소켓 플랫폼에서 구동되는 최초의 Amazon 2nd 인스턴스입니다.

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 지침을 지원하여 대부분의 정수 명령을 256비트로 확장합니다.

일부 메모리 최적화 인스턴스는 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 주파수로 측정)을 제어합니다. 자세한 내용은 EC2 인스턴스에 대한 프로세서 상태 제어 (p. 537) 단원을 참조하십시오.

네트워크 성능

지원되는 인스턴스 유형에서 향상된 네트워킹 기능을 활성화할 수 있습니다. 향상된 네트워킹을 통해 PPS(Packet Per Second) 성능이 크게 높아지고, 네트워크 지터 및 지연 시간이 낮아집니다. 자세한 내용은 Linux에서 향상된 네트워킹 (p. 706) 단원을 참조하십시오.

향상된 네트워킹을 지원하는 데 ENA(Elastic Network Adapter)를 사용하는 인스턴스 유형은 일관되게 낮은 지연 시간과 함께 높은 초당 패킷 성능을 제공합니다. 대부분의 애플리케이션은 항시 높은 수준의 네트워크 성능을 필요로 하지 않지만, 데이터를 주고 받을 때 넓은 대역폭에 액세스할 수 있을 경우 유익할 수 있습니다. ENA를 사용하고 "최대 10Gbps" 또는 "최대 25Gbps"의 네트워크 성능으로 기록된 인스턴스 크기는 네트워크 I/O 크레딧 메커니즘을 사용하여 평균 대역폭 이용률에 따라 네트워크 대역폭을 인스턴스에 할당합니다. 이러한 인스턴스의 네트워크 대역폭이 기준 한도 미만으로 떨어지면 크레딧이 발생하는데, 이 크레딧은 네트워크 데이터를 전송할 때 사용할 수 있습니다.

다음은 향상된 네트워킹을 지원하는 메모리 최적화 인스턴스용 네트워크 성능의 요약 설명입니다.

인스턴스 유형	네트워크 성능	향상된 네트워킹
`r4.4xlarge` 이하 | `r5.4xlarge` 이하 | `r5a.8xlarge` 이하 | `r5ad.4xlarge` 이하 | `r5d.4xlarge` 이하 | `x1e.8large` 이하 | `z1d.3xlarge` 이하	최대 10Gbps	ENA (p. 707)
`r4.8xlarge` | `r5.8xlarge` | `r5.12xlarge` | `r5a.12xlarge` | `r5ad.12xlarge` | `r5d.8xlarge` | `r5d.12xlarge` | `x1.16xlarge` | `x1e.16xlarge` | `z1d.6xlarge`	10Gbps	ENA (p. 707)
`r5a.16xlarge` | `r5ad.16xlarge`	12Gbps	ENA (p. 707)
`r5.16xlarge` | `r5a.24xlarge` | `r5ad.24xlarge` | `r5d.16xlarge`	20Gbps	ENA (p. 707)
`r5dn.4xlarge` 이하 | `r5n.4xlarge` 이하	최대 25Gbps	ENA (p. 707)
`r4.16xlarge` | `r5.24xlarge` | `r5.metal` | `r5d.24xlarge` | `r5d.metal` | `r5dn.8xlarge` |	25Gbps	ENA (p. 707)

인스턴스 유형	네트워크 성능	향상된 네트워킹
r5n.8xlarge | u-6tb1.metal | u-9tb1.metal | u-12tb1.metal | x1.32xlarge | x1e.32xlarge | z1d.12xlarge | z1d.metal		
r5dn.12xlarge | r5n.12xlarge	50Gbps	ENA (p. 707)
r5dn.16xlarge | r5n.16xlarge	75Gbps	ENA (p. 707)
r5dn.24xlarge | r5n.24xlarge | u-18tb1.metal | u-24tb1.metal	100Gbps	ENA (p. 707)

SSD I/O 성능

커널 버전이 4.4 이상인 Linux AMI를 사용하고 인스턴스에서 사용 가능한 모든 SSD 기반의 인스턴스 스토어 볼륨을 활용하는 경우, 다음 표와 같은 IOPS(블록 크기 4,096바이트) 성능을 얻을 수 있습니다(대기열 깊이 포화 상태에서). 그렇지 않으면 IOPS 성능이 더 낮아집니다.

인스턴스 크기	100% 임의 읽기 IOPS	IOPS 쓰기
r5ad.large *	30,000개	15,000
r5ad.xlarge *	59,000	29,000
r5ad.2xlarge *	117,000	57,000
r5ad.4xlarge *	234,000	114,000
r5ad.12xlarge	700,000	340,000
r5ad.24xlarge	1,400,000	680,000
r5d.large *	30,000개	15,000
r5d.xlarge *	59,000	29,000
r5d.2xlarge *	117,000	57,000
r5d.4xlarge *	234,000	114,000
r5d.8xlarge	466,666	233,333
r5d.12xlarge	700,000	340,000
r5d.16xlarge	933,333	466,666
r5d.24xlarge	1,400,000	680,000
r5d.metal	1,400,000	680,000
r5dn.large *	30,000개	15,000
r5dn.xlarge *	59,000	29,000
r5dn.2xlarge *	117,000	57,000
r5dn.4xlarge *	234,000	114,000
r5dn.8xlarge	466,666	233,333

인스턴스 크기	100% 임의 읽기 IOPS	IOPS 쓰기
`r5dn.12xlarge`	700,000	340,000
`r5dn.16xlarge`	933,333	466,666
`r5dn.24xlarge`	1,400,000	680,000
`z1d.large` *	30,000개	15,000
`z1d.xlarge` *	59,000	29,000
`z1d.2xlarge` *	117,000	57,000
`z1d.3xlarge` *	175,000	75,000
`z1d.6xlarge`	350,000	170,000
`z1d.12xlarge`	700,000	340,000
`z1d.metal`	700,000	340,000

* 이러한 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. 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 않음을 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. 1037) 단원을 참조하십시오.

인스턴스 기능

메모리 최적화 인스턴스에 대한 기능은 다음과 같이 간략히 설명할 수 있습니다.

	EBS 전용	NVMe EBS	인스턴스 스토어	배치 그룹
R4	예	아니요	아니요	예
R5	예	예	아니요	예
R5a	예	예	아니요	예

	EBS 전용	NVMe EBS	인스턴스 스토어	배치 그룹
R5ad	아니요	예	NVME *	예
R5d	아니요	예	NVME *	예
R5dn	아니요	예	NVME *	예
R5n	예	예	아니요	예
`u-6tb1.metal`	예	예	아니요	아니요
`u-9tb1.metal`	예	예	아니요	아니요
`u-12tb1.metal`	예	예	아니요	아니요
`u-18tb1.metal`	예	예	아니요	아니요
`u-24tb1.metal`	예	예	아니요	아니요
X1	아니요	아니요	SSD	예
X1e	아니요	아니요	SSD *	예
z1d	아니요	예	NVME *	예

* 루트 디바이스 볼륨은 Amazon EBS 볼륨이어야 합니다.

자세한 내용은 다음 단원을 참조하십시오.

- Linux 인스턴스의 Amazon EBS 및 NVMe (p. 981)
- Amazon EC2 인스턴스 스토어 (p. 1027)
- 배치 그룹 (p. 757)

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 2(HVM)
- 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 2019
- Windows Server 2016
- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2 64비트
- Windows Server 2008 SP2 64비트

출시 정보

- R5 및 R5d 인스턴스는 1세대(Skylake-SP) 또는 2세대(Cascade Lake)의 3.1 GHz Intel Xeon Platinum 8000 시리즈 프로세서를 사용합니다.
- R5a 및 R5ad 인스턴스는 2.5GHz AMD EPYC 7000 시리즈 프로세서를 사용하는 것이 특징입니다.
- 다음은 고용량 메모리 R5, R5a, R5ad, R5d, R5dn, R5n 및 z1d 인스턴스에 대한 요구 사항입니다.
  - NVMe 드라이버 (p. 981)가 설치되어 있어야 합니다.
  - ENA(Elastic Network Adapter) (p. 707) 드라이버가 설치되어 있어야 합니다.

  다음 Linux AMI는 이러한 요구 사항을 충족합니다.
  - Amazon Linux 2
  - Amazon Linux AMI 2018.03
  - Ubuntu 14.04(`linux-aws` 커널) 이상
  - Red Hat Enterprise Linux 7.4 이상
  - SUSE Linux Enterprise Server 12 SP2 이상
  - CentOS 7.4.1708 이상
  - FreeBSD 11.1 이상
  - Debian GNU/Linux 9 이상
- R5, R5a, R5ad, R5d, R5dn, R5n 및 z1d 인스턴스는 네트워크 인터페이스, EBS 볼륨 및 NVMe 인스턴스 스토어 볼륨을 포함해 최대 28개의 연결을 지원합니다. 모든 인스턴스에는 최소 한 개의 네트워크 인터페이스 연결이 있습니다. 예를 들어, EBS 전용 인스턴스에 그 밖의 네트워크 인터페이스 연결이 없는 경우, EBS 볼륨 27개를 연결할 수 있습니다.
- `u-6tb1.metal`, `u-9tb1.metal`, `u-12tb1.metal` 인스턴스는 최대 13개의 EBS 볼륨을 지원합니다. `u-18tb1.metal` 및 `u-24tb1.metal`인스턴스는 최대 19개의 EBS 볼륨을 지원합니다.
- 베어 메탈 인스턴스를 시작하면 기본 서버가 부팅되는데, 이때 모든 하드웨어 및 펌웨어 구성 요소를 확인합니다. 즉, 인스턴스가 실행 상태가 되어 네트워크를 통해 사용할 수 있게 될 때까지 20분이 걸릴 수 있습니다.
- 베어 메탈 인스턴스에서 EBS 볼륨 또는 보조 네트워크 인터페이스를 연결 또는 분리하려면 PCIe 기본 핫플러그 지원이 필요합니다. Amazon Linux 2 및 최신 Amazon Linux AMI 버전에서는 PCIe 기본 핫플러그를 지원하지만 이전 버전에서는 지원하지 않습니다. 다음 Linux 커널 구성 옵션을 활성화해야 합니다.

  ```
  CONFIG_HOTPLUG_PCI_PCIE=y
  CONFIG_PCIEASPM=y
  ```
- 베어 메탈 인스턴스에서는 I/O 포트 기반 직렬 디바이스가 아닌 PCI 기반 직렬 디바이스를 사용합니다. 업스트림 Linux 커널 및 최신 Amazon Linux AMI에서는 이 디바이스를 지원합니다. 베어 메탈 인스턴스도 시스템에서 PCI 기반 직렬 디바이스를 자동으로 사용할 수 있게 해주는 ACPI SPCR 테이블을 제공합니다. 최신 Windows AMI에서는 PCI 기반 직렬 디바이스를 자동으로 사용합니다.
- `x1.16xlarge` 인스턴스를 제외하고는 Windows Server 2008 SP2 64비트 AMI를 사용하여 X1 인스턴스를 시작할 수 없습니다.
- Windows Server 2008 SP2 64비트 AMI를 사용하여 X1e 인스턴스를 시작할 수 없습니다.
- Windows Server 2008 R2 64비트 AMI의 구 버전에서는 `r4.large` 및 `r4.4xlarge` 인스턴스를 시작할 수 없습니다. 이 문제가 발생하면 이 AMI의 최신 버전으로 업데이트합니다.
- 한 리전에서 시작할 수 있는 총 인스턴스 수에는 제한이 있으며, 일부 인스턴스 유형에는 또 다른 제한이 있습니다. 자세한 정보는 Amazon EC2에서 실행 가능한 인스턴스 수 단원을 참조하십시오. 한도 증가를 요청하려면 Amazon EC2 인스턴스 요청 양식을 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

H1 인스턴스

H1 인스턴스는 다음 애플리케이션에 적합합니다.

- MapReduce 및 분산형 파일 시스템 같은 데이터 집약적 워크로드
- 직접 연결 인스턴스 스토리지에서 대량 데이터에 순차적으로 액세스해야 하는 애플리케이션
- 대량의 데이터에 대해 고속 액세스가 필요한 애플리케이션

I3 및 I3en 인스턴스

이러한 인스턴스는 다음 애플리케이션에 매우 적합합니다.

- 빈도가 높은 온라인 트랜잭션 처리(OLTP) 시스템
- 관계형 데이터베이스
- NoSQL 데이터베이스
- 인 메모리 데이터베이스의 캐시(예: Redis)
- 데이터 웨어하우징 애플리케이션
- 분산 파일 시스템

베어 메탈 인스턴스에서는 애플리케이션이 프로세서, 메모리 등 호스트 서버의 물리적 리소스에 직접 액세스할 수 있습니다. 이 인스턴스는 다음과 같은 경우에 적합합니다.

- 가상화된 환경에서 사용할 수 없거나 완전히 지원되지 않는 저수준 하드웨어 기능(예: intel VT)에 액세스해야 하는 워크로드
- 라이선스 또는 지원을 위해 가상화되지 않은 환경이 필요한 애플리케이션

자세한 내용은 Amazon EC2 I3 인스턴스를 참조하십시오.

내용

- 하드웨어 사양 (p. 243)
- 인스턴스 성능 (p. 244)
- 네트워크 성능 (p. 244)
- SSD I/O 성능 (p. 245)
- 인스턴스 기능 (p. 246)
- 개의 vCPU 지원 (p. 246)

- 출시 정보 (p. 247)

하드웨어 사양

D2 인스턴스의 기본 데이터 스토리지는 HDD 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 중지되거나 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. 1027) 및 SSD 인스턴스 스토어 볼륨 (p. 1036) 단원을 참조하십시오.

다음은 스토리지 최적화 인스턴스용 하드웨어 사양의 요약 설명입니다.

인스턴스 유형	기본 vCPU	메모리(GiB)
`d2.xlarge`	4	30.5
`d2.2xlarge`	8	61
`d2.4xlarge`	16	122
`d2.8xlarge`	36	244
`h1.2xlarge`	8	32
`h1.4xlarge`	16	64
`h1.8xlarge`	32	128
`h1.16xlarge`	64	256
`i3.large`	2	15.25
`i3.xlarge`	4	30.5
`i3.2xlarge`	8	61
`i3.4xlarge`	16	122
`i3.8xlarge`	32	244
`i3.16xlarge`	64	488
`i3.metal`	72	512
`i3en.large`	2	16
`i3en.xlarge`	4	32
`i3en.2xlarge`	8	64
`i3en.3xlarge`	12	96
`i3en.6xlarge`	24	192
`i3en.12xlarge`	48	384
`i3en.24xlarge`	96	768
`i3en.metal`	96	768

Amazon EC2 인스턴스 유형별 하드웨어 사양에 대한 자세한 내용은 Amazon EC2 인스턴스 유형을 참조하십시오.

CPU 옵션 지정에 대한 자세한 내용은 CPU 옵션 최적화 (p. 546) 단원을 참조하십시오.

인스턴스 성능

Linux의 인스턴스에서 최상의 디스크 처리량 성능을 보장하려면 최신 Amazon Linux 2 또는 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 기사를 참조하십시오.

EBS에 최적화된 인스턴스를 사용하면 Amazon EBS I/O와 인스턴스의 다른 네트워크 간의 경합을 제거하여 EBS 볼륨에 대해 일관되게 우수한 성능을 제공할 수 있습니다. 일부 스토리지 최적화 인스턴스는 추가 비용 없이도 EBS에 최적화할 수 있게 기본 설정되어 있습니다. 자세한 내용은 Amazon EBS 최적화 인스턴스 (p. 984) 단원을 참조하십시오.

일부 스토리지 최적화 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. 537) 단원을 참조하십시오.

네트워크 성능

지원되는 인스턴스 유형에서 향상된 네트워킹 기능을 활성화할 수 있습니다. 향상된 네트워킹을 통해 PPS(Packet Per Second) 성능이 크게 높아지고, 네트워크 지터 및 지연 시간이 낮아집니다. 자세한 내용은 Linux에서 향상된 네트워킹 (p. 706) 단원을 참조하십시오.

향상된 네트워킹을 지원하는 데 ENA(Elastic Network Adapter)를 사용하는 인스턴스 유형은 일관되게 낮은 지연 시간과 함께 높은 초당 패킷 성능을 제공합니다. 대부분의 애플리케이션은 항시 높은 수준의 네트워크 성능을 필요로 하지 않지만, 데이터를 주고 받을 때 넓은 대역폭에 액세스할 수 있을 경우 유익할 수 있습니다. ENA를 사용하고 "최대 10Gbps" 또는 "최대 25Gbps"의 네트워크 성능으로 기록된 인스턴스 크기는 네트워크 I/O 크레딧 메커니즘을 사용하여 평균 대역폭 이용률에 따라 네트워크 대역폭을 인스턴스에 할당합니다. 이러한 인스턴스의 네트워크 대역폭이 기준 한도 미만으로 떨어지면 크레딧이 발생하는데, 이 크레딧은 네트워크 데이터를 전송할 때 사용할 수 있습니다.

다음은 확장 네트워킹을 지원하는 스토리지 최적화 인스턴스용 네트워크 성능의 요약 설명입니다.

인스턴스 유형	네트워크 성능	향상된 네트워킹
`i3.4xlarge` 이하	최대 10Gbps, 네트워크 I/O 크레딧 메커니즘 사용	ENA (p. 707)
`i3.8xlarge` | `h1.8xlarge`	10Gbps	ENA (p. 707)
`i3en.3xlarge` 이하	최대 25Gbps, 네트워크 I/O 크레딧 메커니즘 사용	ENA (p. 707)
`i3.16xlarge` | `i3.metal` | `i3en.6xlarge` | `h1.16xlarge`	25Gbps	ENA (p. 707)
`i3en.12xlarge`	50Gbps	ENA (p. 707)

인스턴스 유형	네트워크 성능	향상된 네트워킹
i3en.24xlarge	100Gbps	ENA (p. 707)
d2.xlarge	보통	intel 82599 VF (p. 719)
d2.2xlarge | d2.4xlarge	높음	intel 82599 VF (p. 719)
d2.8xlarge	10Gbps	intel 82599 VF (p. 719)

SSD I/O 성능

커널 버전이 4.4 이상인 Linux AMI를 사용하고 인스턴스에서 사용 가능한 모든 SSD 기반의 인스턴스 스토어 볼륨을 활용하는 경우, 다음 표와 같은 IOPS(블록 크기 4,096바이트) 성능을 얻을 수 있습니다(대기열 깊이 포화 상태에서). 그렇지 않으면 IOPS 성능이 더 낮아집니다.

인스턴스 크기	100% 임의 읽기 IOPS	IOPS 쓰기
i3.large *	100,125	35,000
i3.xlarge *	206,250	70,000
i3.2xlarge	412,500	180,000
i3.4xlarge	825,000	360,000
i3.8xlarge	1.65백만	720,000
i3.16xlarge	3.3백만	1.4백만
i3.metal	3.3백만	1.4백만
i3en.large *	42,500	32,500
i3en.xlarge *	85,000	65,000
i3en.2xlarge *	170,000	130,000
i3en.3xlarge	250,000	200,000
i3en.6xlarge	500,000	400,000
i3en.12xlarge	1백만 개	800,000
i3en.24xlarge	2백만	160만
i3en.metal	2백만	160만

* 이러한 인스턴스의 경우, 지정된 최대 성능을 얻을 수 있습니다.

SSD 기반 인스턴스 저장 볼륨이 줄어듦에 따라 사용자가 얻을 수 있는 I/O 성능이 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 전략은 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. 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 않음을 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. 1037) 단원을 참조하십시오.

인스턴스 기능

스토리지 최적화 인스턴스에 대한 기능은 다음과 같이 간략히 설명할 수 있습니다.

	EBS 전용	인스턴스 스토어	배치 그룹
D2	아니요	HDD	예
H1	아니요	HDD *	예
I3	아니요	NVMe *	예
I3en	아니요	NVMe *	예

* 루트 디바이스 볼륨은 Amazon EBS 볼륨이어야 합니다.

자세한 내용은 다음 단원을 참조하십시오.

- Linux 인스턴스의 Amazon EBS 및 NVMe (p. 981)
- Amazon EC2 인스턴스 스토어 (p. 1027)
- 배치 그룹 (p. 757)

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 2(HVM)
- Amazon Linux AMI 2018.03(HVM)
- Ubuntu Server 14.04 LTS(HVM) 이상
- 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: `stopped` 상태 전환으로 인해 시작 중에 인스턴스 상태가 `Client.InstanceInitiatedShutdown`로 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 `d2.8xlarge`로 변경합니다(작업, 인스턴스 설정, 인스턴스 유형 변경을 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. 중지된 인스턴스의 인스턴스 유형을 `d2.8xlarge`가 아닌 다른 D2 인스턴스 유형으로 변경합니다(작업, 인스턴스 설정, 인스턴스 유형 변경을 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 최근의 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 `d2.8xlarge`로 변경합니다(작업, 인스턴스 설정, 인스턴스 유형 변경을 선택한 다음 지침을 따름).

   f. 인스턴스를 시작합니다.

출시 정보

- HVM AMI를 사용해서 스토리지 최적화 인스턴스를 실행해야 합니다. 자세한 내용은 Linux AMI 가상화 유형 (p. 95) 단원을 참조하십시오.
- 다음은 I3en 및 `i3.metal` 인스턴스에 대한 요구 사항입니다.
  - NVMe 드라이버 (p. 981)가 설치되어 있어야 합니다.
  - ENA(Elastic Network Adapter) (p. 707) 드라이버가 설치되어 있어야 합니다.

  다음 Linux AMI는 이러한 요구 사항을 충족합니다.

  - Amazon Linux 2

  - Amazon Linux AMI 2018.03
  - Ubuntu 14.04(`linux-aws` 커널) 이상
  - Red Hat Enterprise Linux 7.4 이상
  - SUSE Linux Enterprise Server 12 SP2 이상
  - CentOS 7.4.1708 이상
  - FreeBSD 11.1 이상
  - Debian GNU/Linux 9 이상
- `i3.metal` 인스턴스를 시작하면 기본 서버가 부팅되는데, 이때 모든 하드웨어 및 펌웨어 구성 요소를 확인합니다. 즉, 인스턴스가 실행 상태가 되어 네트워크를 통해 사용할 수 있게 될 때까지 20분이 걸릴 수 있습니다.
- 베어 메탈 인스턴스에서 EBS 볼륨 또는 보조 네트워크 인터페이스를 연결 또는 분리하려면 PCIe 기본 핫플러그 지원이 필요합니다. Amazon Linux 2 및 최신 Amazon Linux AMI 버전에서는 PCIe 기본 핫플러그를 지원하지만 이전 버전에서는 지원하지 않습니다. 다음 Linux 커널 구성 옵션을 활성화해야 합니다.

```
CONFIG_HOTPLUG_PCI_PCIE=y
CONFIG_PCIEASPM=y
```

- 베어 메탈 인스턴스에서는 I/O 포트 기반 직렬 디바이스가 아닌 PCI 기반 직렬 디바이스를 사용합니다. 업스트림 Linux 커널 및 최신 Amazon Linux AMI에서는 이 디바이스를 지원합니다. 베어 메탈 인스턴스도 시스템에서 PCI 기반 직렬 디바이스를 자동으로 사용할 수 있게 해주는 ACPI SPCR 테이블을 제공합니다. 최신 Windows AMI에서는 PCI 기반 직렬 디바이스를 자동으로 사용합니다.
- FreeBSD AMI에서 베어 메탈 인스턴스는 부팅하는 데 거의 한 시간이 걸리고 로컬 NVMe 스토리지에 대한 I/O는 완료되지 않습니다. 차선책으로 `/boot/loader.conf`를 다음 행에 추가한 후 재부팅하십시오.

```
hw.nvme.per_cpu_io_queues="0"
```

- `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. 246) 단원을 참조하십시오.
- 한 리전에서 시작할 수 있는 총 인스턴스 수에는 제한이 있으며, 일부 인스턴스 유형에는 또 다른 제한이 있습니다. 자세한 정보는 Amazon EC2에서 실행 가능한 인스턴스 수 단원을 참조하십시오. 한도 증가를 요청하려면 Amazon EC2 인스턴스 요청 양식을 사용하십시오.

Linux 액셀러레이티드 컴퓨팅 인스턴스

높은 처리 기능이 필요한 경우 GPU(그래픽 처리 장치) 또는 FPGA(필드 프로그래머블 게이트 어레이)와 같은 하드웨어 기반 컴퓨팅 액셀러레이터에 대한 액세스를 제공하는 액셀러레이티드 컴퓨팅 인스턴스를 사용하면 도움이 됩니다. 액셀러레이티드 컴퓨팅 인스턴스는 컴퓨팅 집약적 워크로드에서 더 높은 처리량을 위한 병렬 처리를 활성화합니다.

GPU 기반 인스턴스는 수천 개의 컴퓨팅 코어로 NVIDIA GPU에 대한 액세스를 제공합니다. 액셀러레이티드 컴퓨팅 인스턴스로 CUDA(Compute Unified Device Architecture) 또는 OpenCL(Open Computing Language) 병렬 컴퓨팅 프레임워크를 활용하여 GPU 기반 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.

FPGA 기반 인스턴스는 수백만 개의 병렬 시스템 로직 셀로 대용량 FPGA에 대한 액세스를 제공합니다. FPGA 기반 액셀러레이티드 컴퓨팅 인스턴스를 사용하여 사용자 지정 하드웨어 액셀러레이션 활용을 통해 게놈, 재무 분석, 실시간 비디오 프로세싱, 빅 데이터 분석 및 보안 워크로드와 같은 워크로드를 가속화할 수 있습니다. Verilog 또는 VHDL과 같은 하드웨어 설명 언어를 사용하거나 OpenCL 병렬 컴퓨팅 프레임워크와 같은 상위 수준의 언어를 사용하여 이러한 액셀러레이션을 개발할 수 있습니다. 자체 하드웨어 액셀러레이션 코드를 개발하거나 AWS Marketplace를 통해 하드웨어 액셀러레이션을 구매할 수 있습니다.

> **Important**
>
> FPGA 기반 인스턴스는 Microsoft Windows를 지원하지 않습니다.

가속 컴퓨팅 인스턴스를 하나의 클러스터 배치 그룹으로 클러스터링할 수 있습니다. 클러스터 배치 그룹은 단일 가용 영역 내의 인스턴스 간에 낮은 지연 시간과 높은 대역폭 연결을 제공합니다. 자세한 내용은 배치 그룹 (p. 757) 단원을 참조하십시오.

내용

- 액셀러레이티드 컴퓨팅 인스턴스 패밀리 (p. 249)
- 하드웨어 사양 (p. 250)
- 인스턴스 성능 (p. 251)
- 네트워크 성능 (p. 252)
- 인스턴스 기능 (p. 252)
- 출시 정보 (p. 253)
- Linux 인스턴스에 NVIDIA 드라이버 설치 (p. 253)
- NVIDIA GRID 가상 애플리케이션 활성화 (p. 259)
- GPU 설정 최적화 (p. 260)
- FPGA 개발 시작하기 (p. 260)
- AWS Inferentia 개발 시작하기 (p. 261)

Windows 액셀러레이티드 컴퓨팅 인스턴스에 대한 자세한 내용은 Windows 인스턴스용 Amazon EC2 사용 설명서의 Windows 액셀러레이티드 컴퓨팅 인스턴스를 참조하십시오.

액셀러레이티드 컴퓨팅 인스턴스 패밀리

액셀러레이티드 컴퓨팅 인스턴스 패밀리는 하드웨어 액셀러레이터나 코프로세서를 사용함으로써 부동 소수점 계산 및 그래픽 처리, 또는 데이터 패턴 매칭 등과 같은 일부 기능을, CPU에서 실행하는 소프트웨어에서 수행하는 것보다 더 효율적으로 수행합니다. Amazon EC2에서는 다음과 같은 액셀러레이티드 컴퓨팅 인스턴스 패밀리를 실행할 수 있습니다.

F1 인스턴스

F1 인스턴스는 Xilinx UltraScale+ VU9P FPGA를 사용하고 범용 CPU에는 적합하지 않은 데이터 흐름 또는 고도로 병렬적인 작업과 같은 컴퓨팅 집약적 알고리즘을 가속화하도록 설계되었습니다. F1 인스턴스의 각 FPGA에는 약 250만 개의 로직 요소와 약 6,800개의 DSP(Digital Signal Processing) 엔진이 포함되어 있고 전용 PCIe Gen3 x16 연결로 인스턴스에 연결된 로컬 DDR ECC 보호 메모리 64GiB가 있습니다. F1 인스턴스는 로컬 NVMe SSD 볼륨을 제공합니다.

개발자는 FPGA 개발자 AMI 및 AWS 하드웨어 개발자 키트를 사용하여 F1 인스턴스에 사용할 사용자 지정 하드웨어 액셀러레이션을 생성할 수 있습니다. FPGA 개발자AMI에는 클라우드의 전체 주기 FPGA 개발용 개발 도구가 포함됩니다. 이러한 도구를 사용하여 개발자는 F1 인스턴스의 FPGA에 로드할 수 있는 Amazon FPGA 이미지(AFI)를 생성 및 공유할 수 있습니다.

자세한 내용은 Amazon EC2 F1 인스턴스를 참조하십시오.

P3 인스턴스

P3 인스턴스는 NVIDIA Tesla V100 GPU를 사용하며, CUDA 또는 OpenCL 프로그래밍 모델을 사용하거나 머신 러닝 프레임워크를 통해 범용 GPU 컴퓨팅에 맞게 설계되었습니다. P3 인스턴스는 고대역 네트워킹, 강력한 반정밀도, 단정밀도 및 배정밀도 부동 소수점 기능, GPU당 최대 32GiB의 메모리를 제공하므로, 딥 러닝, 전산 유체 역학(CFD), 계산 금융(Computational Finance), 내진 해석, 분자 모델링, 유전체학, 렌더링 및 기타 서버 측 GPU 컴퓨팅 워크로드에 이상적입니다. Tesla V100 GPU는 그래픽 모드를 지원하지 않습니다. 자세한 내용은 Amazon EC2 P3 인스턴스를 참조하십시오.

P3 인스턴스는 NVIDIA NVLink 피어 투 피어 전송을 지원합니다.

시스템에 대한 토폴로지 정보를 보려면 다음 명령을 실행합니다.

```
nvidia-smi topo -m
```

자세한 내용은 NVIDIA NVLink 단원을 참조하십시오.

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 NVIDIA GPUDirect 피어 투 피어 전송을 지원합니다.

시스템에 대한 토폴로지 정보를 보려면 다음 명령을 실행합니다.

```
nvidia-smi topo -m
```

자세한 내용은 NVIDIA GPUDirect 단원을 참조하십시오.

G4 인스턴스

G4 인스턴스는 NVIDIA Tesla GPU를 사용하며 DirectX 또는 OpenGL을 사용하는 그래픽 애플리케이션과 함께 CUDA 또는 기계 학습 프레임워크를 사용하여 범용 GPU 컴퓨팅을 위한 경제적인 고성능 플랫폼을 제공합니다. G4 인스턴스는 INT8 및 INT4 정밀도와 함께 고대역폭 네트워킹, 강력한 반정밀도 및 단정밀도 부동 소수점 기능을 제공합니다. 각 GPU에는 16GiB의 GDDR6 메모리가 있으므로 G4 인스턴스는 기계 학습 추론, 비디오 트랜스코딩, 원격 그래픽 워크스테이션 및 클라우드에서의 게임 스트리밍과 같은 그래픽 애플리케이션에 적합합니다.

G4 인스턴스는 NVIDIA GRID 가상 워크스테이션을 지원합니다. 자세한 내용은 NVIDIA Marketplace 제품군을 참조하십시오.

G3 인스턴스

G3 인스턴스는 NVIDIA Tesla M60 GPU를 사용하며 DirectX 또는 OpenGL을 사용하는 그래픽 애플리케이션을 위한 경제적이고도 높은 성능의 플랫폼을 제공합니다. 또한 G3 인스턴스는 해상도가 최대 4096x2160인 모니터 4대와 NVIDIA GRID 가상 애플리케이션을 지원하는 등 NVIDIA GRID 가상 워크스테이션 기능도 제공합니다. G3 인스턴스는 3D 가상화, 그래픽 집약적인 원격 워크스테이션, 3D 렌더링, 비디오 인코딩, 가상 현실 및 기타 서버 측 그래픽 워크로드 같이 뛰어난 병렬 처리 성능이 필요한 애플리케이션에 적합합니다.

G3 인스턴스는 NVIDIA GRID 가상 워크스테이션 및 NVIDIA GRID 가상 애플리케이션을 지원합니다. 이러한 기능들 중 하나를 활성화하는 방법은 NVIDIA GRID 가상 애플리케이션 활성화 (p. 259) 단원을 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.

하드웨어 사양

다음은 액셀러레이티드 컴퓨팅 인스턴스용 하드웨어 사양의 요약 설명입니다.

인스턴스 유형	기본 vCPU	메모리(GiB)	액셀러레이터
`p2.xlarge`	4	61	1
`p2.8xlarge`	32	488	8

인스턴스 유형	기본 vCPU	메모리(GiB)	액셀러레이터
p2.16xlarge	64	732	16
p3.2xlarge	8	61	1
p3.8xlarge	32	244	4
p3.16xlarge	64	488	8
p3dn.24xlarge	96	768	8
g2.2xlarge	8	15	1
g2.8xlarge	32	60	4
g3s.xlarge	4	30.5	1
g3.4xlarge	16	122	1
g3.8xlarge	32	244	2
g3.16xlarge	64	488	4
g4dn.xlarge	4	16	1
g4dn.2xlarge	8	32	1
g4dn.4xlarge	16	64	1
g4dn.8xlarge	32	128	1
g4dn.12xlarge	48	192	4
g4dn.16xlarge	64	256	1
f1.2xlarge	8	122	1
f1.4xlarge	16	244	2
f1.16xlarge	64	976	8

Amazon EC2 인스턴스 유형별 하드웨어 사양에 대한 자세한 내용은 Amazon EC2 인스턴스 유형을 참조하십시오.

CPU 옵션 지정에 대한 자세한 내용은 CPU 옵션 최적화 (p. 546) 단원을 참조하십시오.

인스턴스 성능

인스턴스에서 최고의 성능을 달성하기 위해 수행할 수 있는 몇 가지 GPU 설정 최적화가 있습니다. 자세한 내용은 GPU 설정 최적화 (p. 260) 단원을 참조하십시오.

EBS에 최적화된 인스턴스를 사용하면 Amazon EBS I/O와 인스턴스의 다른 네트워크 간의 경합을 제거하여 EBS 볼륨에 대해 일관되게 우수한 성능을 제공할 수 있습니다. 일부 액셀러레이티드 컴퓨팅 인스턴스는 추가 비용 없이도 EBS에 최적화할 수 있게 기본 설정되어 있습니다. 자세한 내용은 Amazon EBS 최적화 인스턴스 (p. 984) 단원을 참조하십시오.

일부 액셀러레이티드 컴퓨팅 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. 537) 단원을 참조하십시오.

네트워크 성능

지원되는 인스턴스 유형에서 향상된 네트워킹 기능을 활성화할 수 있습니다. 향상된 네트워킹을 통해 PPS(Packet Per Second) 성능이 크게 높아지고, 네트워크 지터 및 지연 시간이 낮아집니다. 자세한 내용은 Linux에서 향상된 네트워킹 (p. 706) 단원을 참조하십시오.

향상된 네트워킹을 지원하는 데 ENA(Elastic Network Adapter)를 사용하는 인스턴스 유형은 일관되게 낮은 지연 시간과 함께 높은 초당 패킷 성능을 제공합니다. 대부분의 애플리케이션은 항시 높은 수준의 네트워크 성능을 필요로 하지 않지만, 데이터를 주고 받을 때 넓은 대역폭에 액세스할 수 있을 경우 유익할 수 있습니다. ENA를 사용하고 "최대 10Gbps" 또는 "최대 25Gbps"의 네트워크 성능으로 기록된 인스턴스 크기는 네트워크 I/O 크레딧 메커니즘을 사용하여 평균 대역폭 이용률에 따라 네트워크 대역폭을 인스턴스에 할당합니다. 이러한 인스턴스의 네트워크 대역폭이 기준 한도 미만으로 떨어지면 크레딧이 발생하는데, 이 크레딧은 네트워크 데이터를 전송할 때 사용할 수 있습니다.

다음은 향상된 네트워킹을 지원하는 액셀러레이티드 컴퓨팅 인스턴스를 위한 네트워크 성능의 요약 설명입니다.

인스턴스 유형	네트워크 성능	향상된 네트워킹
`f1.2xlarge` | `f1.4xlarge` | `g3.4xlarge` | `p3.2xlarge`	최대 10Gbps	ENA (p. 707)
`g3s.xlarge` | `g3.8xlarge` | `p2.8xlarge` | `p3.8xlarge`	10Gbps	ENA (p. 707)
`g4dn.xlarge` | `g4dn.2xlarge` | `g4dn.4xlarge`	최대 25Gbps	ENA (p. 707)
`f1.16xlarge` | `g3.16xlarge` | `p2.16xlarge` | `p3.16xlarge`	25Gbps	ENA (p. 707)
`g4dn.8xlarge` | `g4dn.12xlarge` | `g4dn.16xlarge`	50Gbps	ENA (p. 707)
`p3dn.24xlarge`	100Gbps	ENA (p. 707)

인스턴스 기능

액셀러레이티드 컴퓨팅 인스턴스를 위한 기능의 요약 설명입니다.

	EBS 전용	NVMe EBS	인스턴스 스토어	배치 그룹
G2	아니요	아니요	SSD	예
G3	예	아니요	아니요	예
G4	아니요	예	NVMe *	예
P2	예	아니요	아니요	예
P3	`p3dn.24xlarge`: 아니요 기타 모든 크기: 예	`p3dn.24xlarge`: 예 기타 모든 크기: 아니요	`p3dn.24xlarge`: NVMe *	예
SharePoint 용 F1	아니요	아니요	NVMe *	예

* 루트 디바이스 볼륨은 Amazon EBS 볼륨이어야 합니다.

자세한 내용은 다음 단원을 참조하십시오.

- Linux 인스턴스의 Amazon EBS 및 NVMe (p. 981)
- Amazon EC2 인스턴스 스토어 (p. 1027)
- 배치 그룹 (p. 757)

출시 정보

- HVM AMI를 사용해서 I 인스턴스를 실행해야 합니다.
- 다음은 G4 인스턴스에 대한 요구 사항입니다.
  - NVMe 드라이버 (p. 981)가 설치되어 있어야 합니다.
  - ENA(Elastic Network Adapter) (p. 707) 드라이버가 설치되어 있어야 합니다.

  다음 Linux AMI는 이러한 요구 사항을 충족합니다.
  - Amazon Linux 2
  - Amazon Linux AMI 2018.03
  - Ubuntu 14.04(`linux-aws` 커널) 이상
  - Red Hat Enterprise Linux 7.4 이상
  - SUSE Linux Enterprise Server 12 SP2 이상
  - CentOS 7.4.1708 이상
  - FreeBSD 11.1 이상
  - Debian GNU/Linux 9 이상
- GPU 기반 인스턴스는 NVIDIA 드라이버를 설치해야 GPU에 액세스할 수 있습니다. 자세한 내용은 Linux 인스턴스에 NVIDIA 드라이버 설치 (p. 253) 단원을 참조하십시오.
- 리전당 100개의 AFI 한도가 있습니다.
- 실행할 수 있는 인스턴스의 수에 제한이 있습니다. 자세한 내용은 Amazon EC2 FAQ에서 Amazon EC2에서 실행 가능한 인스턴스 수를 참조하십시오. 이 제한을 높이도록 요청하려면 Amazon EC2 인스턴스 제한 증가 요청 양식을 사용하십시오.

Linux 인스턴스에 NVIDIA 드라이버 설치

P3 또는 G4 인스턴스와 같이 연결된 GPU가 있는 인스턴스에는 적절한 NVIDIA 드라이버가 설치되어 있어야 합니다. 인스턴스 유형에 따라 퍼블릭 NVIDIA 드라이버를 다운로드하거나, AWS 고객만 사용할 수 있는 Amazon S3에서 드라이버를 다운로드하거나, 드라이버가 미리 설치되어 있는 AMI를 사용할 수 있습니다.

목차

- NVIDIA 드라이버의 유형 (p. 253)
- 인스턴스 유형별로 사용 가능한 드라이버 (p. 254)
- 설치 옵션 (p. 254)
  - 옵션 1: NVIDIA 드라이버가 설치되어 있는 AMI (p. 255)
  - 옵션 2: 퍼블릭 NVIDIA Tesla 드라이버 (p. 255)
  - 옵션 3: GRID 드라이버(G3 및 G4 인스턴스) (p. 255)
  - 옵션 4: NVIDIA 게임 드라이버(G4 인스턴스) (p. 257)

NVIDIA 드라이버의 유형

다음은 GPU 기반 인스턴스에서 사용할 수 있는 주요 NVIDIA 드라이버 유형입니다.

Tesla 드라이버

이러한 드라이버는 주로 컴퓨팅 워크로드를 위한 것입니다. 컴퓨팅 워크로드는 기계 학습을 위한 병렬화된 부동 소수점 계산과 고성능 컴퓨팅 애플리케이션을 위한 고속 푸리에 변환과 같은 컴퓨팅 작업에 GPU를 사용합니다.

GRID 드라이버

이러한 드라이버는 3D 모델 또는 고해상도 비디오와 같은 콘텐츠를 렌더링하는 전문 시각화 애플리케이션에 최적의 성능을 제공하는 것으로 인증됩니다. 두 가지 모드를 지원하도록 GRID 드라이버를 구성할 수 있습니다. Quadro 가상 워크스테이션은 GPU당 4개의 4K 디스플레이에 대한 액세스를 제공합니다. GRID vApp은 RDSH 앱 호스팅 기능을 제공합니다.

게임 드라이버

이러한 드라이버는 게임을 위한 최적화 기능을 포함하고 있으며 성능 향상을 제공하기 위해 자주 업데이트됩니다. 이러한 드라이버는 GPU당 단일 4K 디스플레이를 지원합니다.

NVIDIA 제어판

NVIDIA 제어판은 GRID 및 게임 드라이버에서 지원됩니다. Tesla 드라이버에서는 지원되지 않습니다.

에 지원되는 API

- OpenCL, OpenGL 및 Vulkan
- NVIDIA CUDA 및 관련 라이브러리(예: cuDNN, TensorRT, nvJPEG, cuBLAS)
- 비디오 인코딩용 NVENC 및 비디오 디코딩용 NVDEC

인스턴스 유형별로 사용 가능한 드라이버

다음 표에는 각 GPU 인스턴스 유형에 지원되는 NVIDIA 드라이버가 요약되어 있습니다.

인스턴스 유형	Tesla 드라이버	GRID 드라이버	게임 드라이버
G2	예	아니요	아니요
G3	예	예	아니요
G4	예	예	예
P2	예	아니요	아니요
P3	예	예 †	아니요

† Marketplace AMI만 사용

설치 옵션

다음 옵션 중 하나를 사용하여 GPU 인스턴스에 필요한 NVIDIA 드라이버를 가져옵니다.

옵션

- 옵션 1: NVIDIA 드라이버가 설치되어 있는 AMI (p. 255)
- 옵션 2: 퍼블릭 NVIDIA Tesla 드라이버 (p. 255)
- 옵션 3: GRID 드라이버(G3 및 G4 인스턴스) (p. 255)
- 옵션 4: NVIDIA 게임 드라이버(G4 인스턴스) (p. 257)

옵션 1: NVIDIA 드라이버가 설치되어 있는 AMI

AWS와 NVIDIA는 NVIDIA 드라이버가 설치된 상태로 제공되는 다양한 Amazon 머신 이미지(AMI)를 제공합니다.

- Tesla 드라이버와 함께 제공되는 Marketplace 상품
- GRID 드라이버와 함께 제공되는 Marketplace 상품
- 게임 드라이버와 함께 제공되는 Marketplace 상품

이러한 AMI 중 하나를 사용하여 설치된 드라이버 버전을 업데이트하려면 버전 충돌을 피하기 위해 인스턴스에서 NVIDIA 패키지를 제거해야 합니다. 다음 명령을 사용하여 NVIDIA 패키지를 제거합니다.

```
[ec2-user ~]$ sudo yum erase nvidia cuda
```

CUDA 도구 키트 패키지는 NVIDIA 드라이버에 종속성을 가지고 있습니다. NVIDIA 패키지를 제거하면 CUDA 도구 키트도 삭제됩니다. NVIDIA 드라이버를 설치한 후 CUDA 도구 키트를 다시 설치해야 합니다.

옵션 2: 퍼블릭 NVIDIA Tesla 드라이버

NVIDIA 드라이버를 다운로드하려면

Linux 인스턴스에 로그인하고 http://www.nvidia.com/Download/Find.aspx에서 인스턴스 유형에 적합한 64비트 NVIDIA 드라이버를 다운로드할 수 있습니다.

인스턴스	제품 시리즈	제품
G2	K 시리즈	K520
G3	M-Class	M60
G4 †	T 시리즈	T4
P2	K 시리즈	K80
P3	V 시리즈	V100

† G4 인스턴스에는 드라이버 버전 418.87 이상이 필요합니다.

Linux에서 NVIDIA 드라이버를 설치하려면

드라이버 설치 및 구성에 대한 자세한 내용은 NVIDIA 드라이버 설치 빠른 시작 안내서를 참조하십시오.

옵션 3: GRID 드라이버(G3 및 G4 인스턴스)

이러한 다운로드는 AWS 고객만 사용할 수 있습니다. 드라이버를 다운로드하면 NVIDIA Tesla T4 또는 NVIDIA Tesla M60 하드웨어와 함께 사용하기 위해 AMIs를 개발하는 용도로만 다운로드한 소프트웨어를 사용한다는 것에 동의하는 것입니다. 소프트웨어를 설치하면 NVIDIA GRID 클라우드 최종 사용자 라이선스 계약의 약관이 적용됩니다.

Linux 인스턴스에서 NVIDIA GRID 드라이버를 설치하려면

1. Linux 인스턴스에 연결합니다. gcc 및 make를 설치합니다(아직 설치되지 않은 경우).
2. 패키지 캐시를 업데이트하고 인스턴스에 대해 필요한 패키지 업데이트를 가져옵니다.
   - Amazon Linux, CentOS, Red Hat Enterprise Linux:

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

- Ubuntu 및 Debian:

```
[ec2-user ~]$ sudo apt-get update -y
```

3. (Ubuntu 16.04 이상, `linux-aws` 패키지 포함) `linux-aws` 패키지를 업그레이드하여 최신 버전을 받습니다.

```
[ec2-user ~]$ sudo apt-get upgrade -y linux-aws
```

4. 인스턴스를 재부팅하여 최신 커널 버전을 로드합니다.

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

5. 재부팅이 끝난 후 인스턴스에 다시 연결합니다.
6. 현재 실행 중인 커널의 버전에 맞는 gcc 컴파일러와 커널 헤더 패키지를 설치합니다.

   - Amazon Linux, CentOS, Red Hat Enterprise Linux:

```
[ec2-user ~]$ sudo yum install -y gcc kernel-devel-$(uname -r)
```

   - Ubuntu 및 Debian:

```
[ec2-user ~]$ sudo apt-get install -y gcc make linux-headers-$(uname -r)
```

7. [CentOS, Red Hat Enterprise Linux, Ubuntu, Debian] NVIDIA 그래픽 카드용 `nouveau` 오픈 소스 드라이버를 비활성화합니다.

   a. `nouveau`를 `/etc/modprobe.d/blacklist.conf` 블랙리스트 파일에 추가합니다. 다음 코드 블록을 복사하여 터미널에 붙여 넣습니다.

```
[ec2-user ~]$ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
blacklist vga16fb
blacklist nouveau
blacklist rivafb
blacklist nvidiafb
blacklist rivatv
EOF
```

   b. `/etc/default/grub` 파일을 편집하고 다음 줄을 추가합니다.

```
GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
```

   c. Grub 구성을 다시 빌드합니다.

      - CentOS 및 Red Hat Enterprise Linux:

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

      - Ubuntu 및 Debian:

```
[ec2-user ~]$ sudo update-grub
```

8. 다음 명령을 사용하여 GRID 드라이버 설치 유틸리티를 다운로드합니다.

   - G3 인스턴스의 경우:

```
[ec2-user ~]$ aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/latest/ .
```

- G4 인스턴스의 경우:

```
[ec2-user ~]$ aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/g4/latest/ .
```

여러 버전의 GRID 드라이버가 이 버킷에 저장되어 있습니다. 다음 명령을 사용하여 사용 가능한 모든 버전을 볼 수 있습니다.

```
[ec2-user ~]$ aws s3 ls --recursive s3://ec2-linux-nvidia-drivers/
```

9. 다음 명령을 사용하여 드라이버 설치 유틸리티를 실행할 수 있는 권한을 추가합니다.

```
[ec2-user ~]$ chmod +x NVIDIA-Linux-x86_64*.run
```

10. 다음과 같이 자체 설치 스크립트를 실행하여 다운로드한 GRID 드라이버를 설치합니다. 다음 예를 참조하십시오.

```
[ec2-user ~]$ sudo /bin/sh ./NVIDIA-Linux-x86_64*.run
```

메시지가 표시되면 라이선스 계약에 동의하고 필요에 따라 설치 옵션을 지정합니다. 기본 옵션을 사용해도 됩니다.

11. 인스턴스를 재부팅합니다.

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

12. 드라이버가 작동하는지 확인합니다. 다음 명령의 응답에는 설치된 NVIDIA 드라이버 버전과 GPU에 대한 세부 정보가 나열됩니다.

```
[ec2-user ~]$ nvidia-smi -q | head
```

13. (선택 사항) 최대 4K 해상도의 디스플레이 4개를 활용하는 데 도움이 되도록 고성능 디스플레이 프로토콜인 NICE DCV를 설정합니다.
14. (선택 사항) NVIDIA Quadro 가상 워크스테이션 모드는 기본적으로 활성화되어 있습니다. RDSH 애플리케이션 호스팅 기능을 위해 GRID 가상 애플리케이션을 활성화하려면 NVIDIA GRID 가상 애플리케이션 활성화 (p. 259)의 GRID 가상 애플리케이션 활성화 단계를 완료하십시오.
15. (선택 사항) NVIDIA Quadro 가상 워크스테이션 모드는 기본적으로 활성화되어 있습니다. NVIDIA GRID 가상 애플리케이션을 활성화하려면 NVIDIA GRID 가상 애플리케이션 활성화 (p. 259)에서 GRID vApps 정품 인증 단계를 완료하십시오.

옵션 4: NVIDIA 게임 드라이버(G4 인스턴스)

이러한 드라이버는 AWS 고객만 사용할 수 있습니다. 드라이버를 다운로드하면 NVIDIA Tesla T4 하드웨어와 함께 사용하기 위해 AMIs를 개발하는 용도로만 다운로드한 소프트웨어를 사용한다는 것에 동의하는 것입니다. 소프트웨어를 설치하면 NVIDIA GRID 클라우드 최종 사용자 라이선스 계약의 약관이 적용됩니다.

Linux 인스턴스에서 NVIDIA 게임 드라이버를 설치하려면

1. Linux 인스턴스에 연결합니다. gcc 및 make를 설치합니다(아직 설치되지 않은 경우).
2. 패키지 캐시를 업데이트하고 인스턴스에 대해 필요한 패키지 업데이트를 가져옵니다.

- Amazon Linux, CentOS, Red Hat Enterprise Linux:

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

- Ubuntu 및 Debian:

```
[ec2-user ~]$ sudo apt-get update -y
```

3. (Ubuntu 16.04 이상, `linux-aws` 패키지 포함) `linux-aws` 패키지를 업그레이드하여 최신 버전을 받습니다.

```
[ec2-user ~]$ sudo apt-get upgrade -y linux-aws
```

4. 인스턴스를 재부팅하여 최신 커널 버전을 로드합니다.

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

5. 재부팅이 끝난 후 인스턴스에 다시 연결합니다.
6. 현재 실행 중인 커널의 버전에 맞는 gcc 컴파일러와 커널 헤더 패키지를 설치합니다.

   - Amazon Linux, CentOS, Red Hat Enterprise Linux:

   ```
   [ec2-user ~]$ sudo yum install -y gcc kernel-devel-$(uname -r)
   ```

   - Ubuntu 및 Debian:

   ```
   [ec2-user ~]$ sudo apt-get install -y gcc make linux-headers-$(uname -r)
   ```

7. [CentOS, Red Hat Enterprise Linux, Ubuntu, Debian] NVIDIA 그래픽 카드용 `nouveau` 오픈 소스 드라이버를 비활성화합니다.

   a. `nouveau`를 `/etc/modprobe.d/blacklist.conf` 블랙리스트 파일에 추가합니다. 다음 코드 블록을 복사하여 터미널에 붙여 넣습니다.

   ```
   [ec2-user ~]$ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
   blacklist vga16fb
   blacklist nouveau
   blacklist rivafb
   blacklist nvidiafb
   blacklist rivatv
   EOF
   ```

   b. `/etc/default/grub` 파일을 편집하고 다음 줄을 추가합니다.

   ```
   GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
   ```

   c. Grub 구성을 다시 빌드합니다.

   - CentOS 및 Red Hat Enterprise Linux:

   ```
   [ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
   ```

   - Ubuntu 및 Debian:

   ```
   [ec2-user ~]$ sudo update-grub
   ```

8. 다음 명령을 사용하여 게임 드라이버 설치 유틸리티를 다운로드합니다.

```
[ec2-user ~]$ aws s3 cp --recursive s3://nvidia-gaming/linux/latest/ .
```

여러 버전의 게임 드라이버가 이 버킷에 저장되어 있습니다. 다음 명령을 사용하여 사용 가능한 모든 버전을 볼 수 있습니다.

```
[ec2-user ~]$ aws s3 ls --recursive s3://nvidia-gaming/linux/
```

9. 다음 명령을 사용하여 드라이버 설치 유틸리티를 실행할 수 있는 권한을 추가합니다.

```
[ec2-user ~]$ chmod +x NVIDIA-Linux-x86_64*.run
```

10. 다음 명령을 사용하여 설치 프로그램을 실행합니다.

```
[ec2-user ~]$ sudo ./NVIDIA-Linux-x86_64*.run
```

메시지가 표시되면 라이선스 계약에 동의하고 필요에 따라 설치 옵션을 지정합니다. 기본 옵션을 사용해도 됩니다.

11. 다음 명령을 사용하여 필수 구성 파일을 생성합니다.

```
[ec2-user ~]$ cat << EOF | sudo tee -a /etc/nvidia/gridd.conf
vGamingMarketplace=2
EOF
```

12. 인증 파일을 다운로드하고 이름을 바꾸려면 다음 명령을 사용하십시오.

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://s3.amazonaws.com/nvidia-gaming/GridSwCert-Linux.cert"
```

13. 인스턴스를 재부팅합니다.

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

14. (선택 사항) 최대 4K 해상도의 단일 디스플레이를 활용하는 데 도움이 되도록 고성능 디스플레이 프로토콜인 NICE DCV를 설정합니다.
15. (선택 사항) 최대 4K 해상도의 디스플레이 4개를 활용하는 데 도움이 되도록 고성능 디스플레이 프로토콜인 NICE DCV를 설정합니다.

NVIDIA GRID 가상 애플리케이션 활성화

G3 및 G4 인스턴스에서 GRID 가상 애플리케이션을 활성화하려면(NVIDIA GRID 가상 워크스테이션은 기본적으로 활성화되어 있음)`/etc/nvidia/gridd.conf` 파일에서 드라이버에 맞는 제품 유형을 정의해야 합니다.

Linux 인스턴스에서 GRID 가상 애플리케이션을 활성화하려면

1. 제공된 템플릿 파일에서 `/etc/nvidia/gridd.conf` 파일을 생성합니다.

```
[ec2-user ~]$ sudo cp /etc/nvidia/gridd.conf.template /etc/nvidia/gridd.conf
```

2. 즐겨찾는 텍스트 편집기에서 `/etc/nvidia/gridd.conf` 파일을 엽니다.
3. `FeatureType` 줄을 찾은 다음 0와 동일하게 설정합니다. 그런 다음 `IgnoreSP=TRUE`로 라인을 추가합니다.

```
FeatureType=0
IgnoreSP=TRUE
```

4. 파일을 저장하고 종료합니다.
5. 인스턴스를 재부팅하여 새 구성을 적용합니다.

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

GPU 설정 최적화

G3, G4, P2, P3, P3dn 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 GPU 인스턴스의 성능을 최대로 유지할 수 있습니다. 다음 절차는 GPU 설정을 영구적으로 구성하고, 자동 부스트 기능을 비활성화하며, GPU 클록 속도를 최대 주파수로 설정하는 방법을 보여줍니다.

GPU 설정을 최적화하려면

1. GPU 설정을 영구적으로 구성합니다. 이 명령은 실행하는 데 몇 분이 소요될 수 있습니다.

```
[ec2-user ~]$ sudo nvidia-persistenced
```

2. 인스턴스에 대해 모든 GPU의 자동 부스트 기능을 비활성화합니다.

```
[ec2-user ~]$ sudo nvidia-smi --auto-boost-default=0
```

   Note

   P3, P3dn 및 G4 인스턴스에서 GPU는 자동 부스트를 지원하지 않습니다.

3. 모든 GPU 클록 속도를 최대 주파수로 설정합니다. 다음 명령에 지정된 메모리와 그래픽 클록 속도를 사용합니다.

   Note

   일부 버전의 NVIDIA 드라이버는 애플리케이션 클록 속도를 설정할 수 없으며 `"Setting applications clocks is not supported for GPU …"` 오류가 발생합니다. 이 오류는 무시해도 됩니다.

   - G3 인스턴스:

```
[ec2-user ~]$ sudo nvidia-smi -ac 2505,1177
```

   - G4 인스턴스:

```
[ec2-user ~]$ sudo nvidia-smi -ac 5001,1590
```

   - P2 인스턴스:

```
[ec2-user ~]$ sudo nvidia-smi -ac 2505,875
```

   - P3 및 P3dn 인스턴스::

```
[ec2-user ~]$ sudo nvidia-smi -ac 877,1530
```

FPGA 개발 시작하기

FPGA 개발자 AMI는 AFI를 개발, 테스트 및 빌드할 수 있는 도구를 제공합니다. 시스템 메모리가 최소 32GB인 모든 EC2 인스턴스(예: C5, M4, R4 인스턴스)에 FPGA 개발자 AMI를 사용할 수 있습니다.

자세한 내용은 AWS FPGA 하드웨어 개발 키트 설명서를 참조하십시오.

AWS Inferentia 개발 시작하기

AWS Deep Learning AMI에서는 AWS Inferentia를 사용하여 기계 학습 애플리케이션을 개발, 테스트 및 빌드할 수 있는 도구를 제공합니다. 모든 Amazon EC2 Inf1 인스턴스에서 AWS Inferentia 개발을 위해 Deep Learning AMI를 사용할 수 있습니다.

자세한 내용은 AWS Neuron 설명서를 참조하십시오.

인스턴스 유형 찾기

인스턴스를 시작하려면 먼저 사용할 인스턴스 유형을 선택해야 합니다. 선택한 인스턴스 유형은 시작할 인스턴스의 요구 사항에 따라 다를 수 있습니다. 예를 들면 다음과 같은 요구 사항을 고려해야 할 수 있습니다.

- Region
- 아키텍처: 32비트(i386), 64비트(x86_64) 또는 64비트 ARM(arm64)
- 컴퓨팅
- Memory
- 스토리지
- 네트워크 성능

Amazon EC2 콘솔을 사용하여 인스턴스 유형 찾기

Amazon EC2 콘솔을 사용하여 인스턴스 유형을 찾을 수 있습니다. 인스턴스 유형 페이지를 사용하여 사용 가능한 모든 인스턴스 유형을 검색할 수 있습니다.

1. Amazon EC2 콘솔을 엽니다.
2. 탐색 모음에서 인스턴스를 실행할 리전을 선택합니다. 현재 위치와 관계없이 사용자가 고를 수 있는 리전을 임의로 선택합니다.
3. 탐색 창에서 인스턴스 유형을 선택합니다.
4. (선택 사항) 기본 설정을 아이콘을 선택하여 표시할 인스턴스 유형 속성을 선택합니다(예: 온디맨드 Linux 요금). 또는 목록에서 인스턴스 유형을 선택하고 세부 정보 탭에서 모든 속성을 볼 수 있습니다.
5. (선택 사항) 필터 옵션을 사용하여 관심 있는 인스턴스 유형만 보이도록 표시되는 인스턴스 유형의 목록 범위를 지정합니다. 예를 들어, vCPU가 8개를 초과하고 최대 절전 모드를 지원하는 모든 인스턴스 유형을 나열할 수 있습니다.
6. (선택 사항) 여러 인스턴스 유형을 선택하여 세부 정보 창에 있는 모든 속성을 항목별로 비교해 봅니다.
7. (선택 사항) 추가 검토를 위해 인스턴스 유형 목록을 저장하려면 목록 다운로드(CSV)를 선택합니다. 쉼표로 구분된 값(.csv) 형식 파일이 다운로드됩니다. 이 파일에는 설정한 필터(있는 경우)와 일치하고 테이블에 표시된 모든 속성과 일치하는 모든 인스턴스 유형이 포함됩니다.

AWS CLI를 사용하여 인스턴스 유형 찾기

Amazon EC2에 대한 AWS CLI 명령을 사용하여 필요에 맞는 인스턴스 유형만 나열할 수 있습니다. 필요에 맞는 인스턴스 유형을 찾은 후 인스턴스를 시작할 때 사용할 수 있도록 ID를 기록해 둡니다. 자세한 내용은 AWS Command Line Interface 사용 설명서의 AWS CLI를 사용하여 인스턴스 시작을 참조하십시오.

describe-instance-types 명령은 필터링 파라미터를 지원합니다. 예를 들어, 다음 필터를 사용하여 vCPU 48개가 있는 인스턴스 유형만 표시할 수 있습니다.

```
aws ec2 describe-instance-types --filters "Name=vcpu-info.default-vcpus,Values=48"
```

describe-instance-type-offerings 명령은 필터링 파라미터를 지원합니다. 예를 들어, `--location-type` 파라미터를 사용하여 가용 영역 제공을 표시할 수 있습니다.

```
aws ec2 describe-instance-type-offerings --location-type "availability-zone"
```

다음 필터를 이전 명령에 추가하여 `us-east-1a` 가용 영역에서 제공되는 인스턴스 유형만 표시할 수 있습니다.

```
--filters "Name=location,Values=us-east-1a"
```

인스턴스 유형 변경

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 워크로드에 적합한 다른 인스턴스 유형으로 변경할 수 있습니다.

또는 IPv6 지원과 같은 일부 기능의 장점을 활용하기 위해 이전 세대 인스턴스 유형에서 현재 세대 인스턴스 유형으로 마이그레이션할 수도 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, 필요한 인스턴스 유형의 새 인스턴스로 애플리케이션을 마이그레이션해야 합니다. 루트 디바이스 볼륨에 대한 자세한 내용은 루트 디바이스 스토리지 (p. 93) 단원을 참조하십시오.

인스턴스의 크기를 조정할 경우 인스턴스의 구성과 호환되는 인스턴스 유형을 선택해야 합니다. 필요한 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. (기본적으로 제공되는) NVMe 인스턴스 스토어 볼륨을 지원하는 인스턴스 유형의 경우, 크기 조정된 인스턴스는 AMI에 따라 추가 인스턴스 스토어 볼륨을 가질 수도 있습니다. 그렇지 않다면 애플리케이션을 새 인스턴스 유형이 있는 인스턴스로 직접 마이그레이션해, 새 인스턴스 실행 시 필요한 인스턴스 스토어 볼륨 수를 지정하면 됩니다.

내용

- 인스턴스 크기 조정을 위한 호환성 (p. 262)
- Amazon EBS– 지원 인스턴스 크기 조정 (p. 263)
- 인스턴스 스토어 지원 인스턴스 마이그레이션 (p. 264)
- 새 인스턴스 구성으로 마이그레이션 (p. 265)

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 반가상화(PV) 또는 하드웨어 가상 머신(HVM)의 두 가지 가상화를 사용합니다. PV AMI에서 시작한 인스턴스를 HVM 전용의 인스턴스 유형으로 크기 조정할 수 없습니다. 자세한 내용은 Linux AMI 가상화 유형 (p. 95) 단원을 참조하십시오. 인스턴스의 가상화 유형을 확인하려면 Amazon EC2 콘솔의 인스턴스 화면에서 세부 정보 창의 가상화 필드를 확인하십시오.
- 아키텍처: AMI는 프로세서의 아키텍처에 고유하기 때문에 프로세서 아키텍처가 현재 인스턴스 유형과 동일한 인스턴스를 선택해야 합니다. 다음 예를 참조하십시오.

  - A1 인스턴스는 Arm 아키텍처 기반 프로세서를 지원하는 유일한 인스턴스입니다. Arm 아키텍처 기반 프로세서를 사용하는 인스턴스 유형의 크기를 조정하는 경우 Arm 아키텍처 기반 프로세서를 지원하는 인스턴스 유형으로 제한됩니다.
  - 다음 인스턴스 유형은 32비트 AMIs를 지원하는 유일한 인스턴스 유형입니다. `t2.nano`, `t2.micro`, `t2.small`, `t2.medium`, `c3.large`, `t1.micro`, `m1.small`, `m1.medium` 및 `c1.medium` 32비트 인스턴스의 크기를 조정하는 경우는 상기 인스턴스 유형만 사용 가능합니다.
- 네트워크: 최신 인스턴스 유형은 VPC에서 시작해야 합니다. 따라서 기본이 아닌 VPC가 아닌 한 EC2-Classic의 인스턴스를, VPC에서만 사용할 수 있는 인스턴스 유형으로 크기 조정할 수 없습니다. 인스턴스가 VPC에 있는지 확인하려면 Amazon EC2 콘솔의 인스턴스 화면에서 세부 정보 창의 VPC ID 값을 확인하십시오. 자세한 내용은 Linux 내 EC2-Classic 인스턴스에서 VPC 내 Linux 인스턴스로 마이그레이션 (p. 788) 단원을 참조하십시오.
- 향상된 네트워킹: 향상된 네트워킹 (p. 706)을 지원하는 인스턴스 유형을 사용하려면 필요한 드라이버가 설치되어 있어야 합니다. 예를 들어 A1, C5, C5d, C5n, F1, G3, G4, H1, I3, I3en, Inf1, `m4.16xlarge`, M5, M5a, M5ad, M5d, M5dn, M5n, P2, P3, R4, R5, R5a, R5ad, R5d, R5dn, R5n, T3, T3a, `u-6tb1.metal`, `u-9tb1.metal`, `u-12tb1.metal`, `u-18tb1.metal`, `u-24tb1.metal`, X1, X1e, and z1d 인스턴스 유형에는 ENA(Elastic Network Adapter) 드라이버가 설치된 EBS 지원 AMI가 필요합니다. 향상된 네트워킹을 지원하는 인스턴스에 맞게 기존 인스턴스의 크기를 조정하려는 경우, 먼저 해당 인스턴스에 ENA 드라이버 (p. 707) 또는 ixgbevf 드라이버 (p. 719)를 적절히 설치해야 합니다.
- NVMe: EBS 볼륨이 Nitro 기반 인스턴스 (p. 181)에 NVMe 블록 디바이스로 표시됩니다. NVMe를 지원하지 않는 인스턴스 유형에서 NVMe를 지원하는 인스턴스 유형으로 인스턴스의 크기를 조정하는 경우 먼저, 인스턴스에 NVMe 드라이버 (p. 981)를 설치해야 합니다. 또한 블록 디바이스 매핑에서 지정한 디바이스에 대한 디바이스 이름은 NVMe 디바이스 이름(`/dev/nvme[0-26]n1`)을 이용하여 변경됩니다. 따라서 `/etc/fstab`을 사용하여 부팅 시 파일 시스템을 탑재하려면 디바이스 이름 대신 UUID/레이블을 사용해야 합니다.
- AMI: 향상된 네트워킹 및 NVMe를 지원하는 인스턴스 유형별로 필요한 AMI에 대한 정보는 다음 설명서의 릴리스 정보를 참조하십시오.
  - 범용 인스턴스 (p. 185)
  - 컴퓨팅 최적화 인스턴스 (p. 227)
  - 메모리 최적화 인스턴스 (p. 232)
  - 스토리지 최적화 인스턴스 (p. 242)

Amazon EBS– 지원 인스턴스 크기 조정

인스턴스 유형을 변경하기 전에는 Amazon EBS– 지원 인스턴스를 중단해야 합니다. 인스턴스를 중지했다가 시작할 때 다음 사항을 인식하십시오.

- 인스턴스를 새 하드웨어로 이동하지만, 인스턴스 ID는 변경되지 않습니다.
- 인스턴스에 퍼블릭 IPv4 주소가 있는 경우 주소를 해제하고 새 퍼블릭 IPv4 주소를 제공합니다. 인스턴스는 프라이빗 IPv4 주소와 모든 탄력적 IP 주소(EIP), IPv6 주소를 유지합니다.
- 인스턴스가 Auto Scaling 그룹에 있는 경우, Amazon EC2 Auto Scaling 서비스는 중단된 인스턴스를 비정상으로 간주해 이를 종료하고 대체 인스턴스를 시작합니다. 이를 방지하기 위해서는 해당 인스턴스의 크기를 조정하는 동안 그 그룹에 대한 조정 프로세스를 일시 중지할 수 있습니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 조정 프로세스 일시 중단 및 재개 단원을 참조하십시오.
- 인스턴스가 클러스터 배치 그룹 (p. 758)에 있으며 인스턴스 유형을 변경한 후 인스턴스 시작에 실패한 경우 다음 작업을 시도합니다. 클러스터 배치 그룹의 모든 인스턴스를 중지하고 영향을 받은 인스턴스의 인스턴스 유형을 변경한 다음 클러스터 배치 그룹의 모든 인스턴스를 다시 시작합니다.
- 가동 중지는 인스턴스가 중단되었을 때 계획해야 합니다. 인스턴스 중단 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. 507) 단원을 참조하십시오.

다음 절차를 사용해서 AWS Management 콘솔을 통해 Amazon EBS– 지원 인스턴스의 크기를 조정합니다.

Amazon EBS– 지원 인스턴스의 크기 조정

1. (선택 사항) 새로운 인스턴스 유형이 기존 인스턴스에 드라이버가 설치되어 있어야 하는 유형인 경우, 먼저 인스턴스에 연결하여 해당 드라이버를 설치해야 합니다. 자세한 내용은 인스턴스 크기 조정을 위한 호환성 (p. 262) 단원을 참조하십시오.
2. Amazon EC2 콘솔을 엽니다.
3. 탐색 창에서 인스턴스를 선택합니다.
4. 인스턴스를 선택하고 작업, 인스턴스 상태, 중지를 선택합니다.
5. 확인 대화 상자가 나타나면 Yes, Stop을 선택합니다. 인스턴스가 중지하는 데 몇 분 정도 걸릴 수 있습니다.
6. 인스턴스를 선택한 상태에서 Actions, Instance Settings, Change Instance Type를 선택합니다. 인스턴스가 `stopped` 상태가 아닐 경우에는 이 작업을 수행할 수 없습니다.
7. 인스턴스 유형 변경 대화 상자에서 다음을 수행합니다.
   a. 인스턴스 유형에서 원하는 인스턴스 유형을 선택합니다. 원하는 인스턴스 유형이 목록에 없으면 해당 인스턴스의 구성과 호환되지 않는 것입니다. 예를 들어 가상화 유형 때문일 수 있습니다. 자세한 내용은 인스턴스 크기 조정을 위한 호환성 (p. 262) 단원을 참조하십시오.
   b. (선택 사항) 선택한 인스턴스 유형이 EBS– 최적화를 지원할 경우 EBS 최적을 선택하여 EBS– 최적화를 활성화하거나, EBS 최적의 선택을 취소하여 EBS– 최적화를 비활성화합니다. 선택한 인스턴스 유형이 기본적으로 EBS에 최적화되었을 경우 EBS 최적이 선택되고 이를 선택 취소할 수 없습니다.
   c. 적용을 선택하여 새로운 설정을 승인합니다.
8. 중지된 인스턴스를 다시 시작하려면 인스턴스를 선택하고 Actions, Instance State, Start를 선택합니다.
9. 확인 대화 상자가 나타나면 예, 시작을 선택합니다. 인스턴스가 `running` 상태가 되는 데 몇 분 정도 걸릴 수 있습니다.
10. (문제 해결) 인스턴스가 부팅되지 않는 경우 새 인스턴스 유형에 대한 요구 사항 중 하나가 충족되지 않았을 가능성이 있습니다. 자세한 내용은 인스턴스 유형을 변경한 후 Linux 인스턴스가 부팅되지 않는 이유는 무엇입니까?를 참조하십시오.

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. 영구 스토리지를 유지해야 할 인스턴스 스토어 볼륨에 데이터를 백업합니다. 유지해야 하는 EBS 볼륨에 데이터를 마이그레이션하려면 볼륨의 스냅샷을 생성하거나(Amazon EBS 스냅샷 만들기 (p. 929) 참조), 나중에 새 인스턴스에 연결할 수 있도록 인스턴스에서 볼륨을 분리합니다(인스턴스에서 Amazon EBS 볼륨 분리 (p. 924) 참조).
2. 인스턴스 스토어 기반 Linux AMI 생성 (p. 115)의 사전 조건을 충족하고 해당 절차를 수행해서 인스턴스 스토어 지원 인스턴스에서 AMI를 생성합니다. 인스턴스에서 AMI를 생성했으면 이 절차로 다시 돌아옵니다.
3. 탐색 창에서 Amazon EC2 콘솔을 열고 AMI를 선택합니다. 필터 목록에서 내 소유를 선택하고 이전 단계에서 생성한 이미지를 선택합니다. 여기서 AMI 이름은 이미지를 등록할 때 지정한 이름, 소스는 사용자의 Amazon S3 버킷입니다.

   Note

   이전 단계에서 생성한 AMI가 표시되지 않을 경우 AMI를 생성한 리전을 선택했는지 확인합니다.

4. 시작을 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 되는 데 몇 분 정도 걸릴 수 있습니다.
5. (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. 작업, 인스턴스 상태, 종료를 차례로 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.

애플리케이션을 호환되는 인스턴스로 마이그레이션하려면

1. 영구 스토리지를 유지해야 할 인스턴스 스토어 볼륨에 데이터를 백업합니다. 유지해야 하는 EBS 볼륨에 데이터를 마이그레이션하려면 볼륨의 스냅샷을 생성하거나(Amazon EBS 스냅샷 만들기 (p. 929) 참조), 나중에 새 인스턴스에 연결할 수 있도록 인스턴스에서 볼륨을 분리합니다(인스턴스에서 Amazon EBS 볼륨 분리 (p. 924) 참조).
2. 다음을 선택하여 새 인스턴스를 시작합니다.
   - HVM AMI
   - HVM 전용 인스턴스 유형
   - 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 탄력적 IP를 선택합니다.
   b. 원래 인스턴스와 연결된 탄력적 IP 주소를 선택하고 작업, 주소 연결 해제를 차례로 선택합니다. 확인 메시지가 나타나면 주소 연결 해제를 선택합니다.
   c. 탄력적 IP 주소를 선택한 상태에서 작업, 주소 연결을 차례로 선택합니다.
   d. 인스턴스에서 새 인스턴스를 선택한 후 연결을 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. 작업, 인스턴스 상태, 종료를 차례로 선택합니다.

인스턴스 유형에 대한 권장 사항 가져오기

AWS Compute Optimizer는 성능 향상, 비용 절감 또는 두 가지 모두를 위한 Amazon EC2 인스턴스 권장 사항을 제공합니다. 이러한 권장 사항을 사용하여 새 인스턴스 유형으로 이동할지 여부를 결정할 수 있습니다.

권장 사항을 만들기 위해 Compute Optimizer는 기존 인스턴스 사양과 사용률 지표를 분석합니다. 그런 다음 컴파일된 데이터를 사용하여 기존 워크로드를 처리하는 데 가장 적합한 Amazon EC2 인스턴스 유형을 권장합니다. 권장 사항은 시간당 인스턴스 요금과 함께 반환됩니다.

이 주제에서는 Amazon EC2 콘솔을 통해 권장 사항을 보는 방법에 대해 간략하게 설명합니다. 자세한 내용은 AWS Compute Optimizer 사용 설명서 단원을 참조하십시오.

> **Note**
>
> Compute Optimizer에서 권장 사항을 받으려면 먼저 Compute Optimizer를 옵트인해야 합니다. 자세한 내용은 AWS Compute Optimizer 사용 설명서의 AWS Compute Optimizer 시작하기를 참조하십시오.

목차

- 제한 사항 (p. 266)
- 결과 (p. 266)
- 권장 사항 보기 (p. 266)
- 권장 사항 평가를 위한 고려 사항 (p. 267)

제한 사항

Compute Optimizer는 현재 M, C, R, T 및 X 인스턴스 유형에 대한 권장 사항을 생성합니다. 다른 인스턴스 유형은 Compute Optimizer의 대상이 아닙니다. 다른 인스턴스 유형을 사용하는 경우, Compute Optimizer 권장 사항 보기에 나타나지 않습니다. 이 인스턴스 유형 및 다른 인스턴스 유형에 대한 자세한 내용은 인스턴스 유형 (p. 178) 단원을 참조하십시오.

결과

Compute Optimizer는 EC2 인스턴스에 대한 결과를 다음과 같이 분류합니다.

- Under-provisioned(프로비저닝 부족) – 적어도 하나의 인스턴스 사양(CPU, 메모리, 네트워크 등)이 워크로드의 성능 요구 사항을 충족하지 않을 때 EC2 인스턴스가 부족하게 프로비저닝되었다고 봅니다. 프로비저닝 부족 EC2 인스턴스는 애플리케이션 성능 저하를 불러올 수 있습니다.
- Over-provisioned(프로비저닝 과다) – 워크로드 성능 요구 사항을 충족하면서 적어도 하나의 인스턴스 사양(CPU, 메모리, 네트워크 등)을 줄일 수 있고, 부족하게 프로비저닝된 사양이 없을 때 EC2 인스턴스가 과하게 프로비저닝되었다고 봅니다. 프로비저닝 과다 EC2 인스턴스는 불필요한 인프라 비용을 유발할 수 있습니다.
- Optimized(최적화) – CPU, 메모리, 네트워크 등 모든 인스턴스 사양이 워크로드의 성능 요구 사항을 충족하고 인스턴스가 과다 프로비저닝되지 않았을 때 EC2 인스턴스가 최적화된 것으로 봅니다. EC2 인스턴스가 최적화되면 최적의 성능과 인프라 비용으로 워크로드가 실행됩니다. 최적화된 인스턴스의 경우 Compute Optimizer가 차세대 인스턴스 유형을 권장하기도 합니다.
- 없음 – 이 인스턴스의 권장 사항이 없습니다. Compute Optimizer를 옵트인한 지 12시간이 지나지 않았거나, 인스턴스가 실행된 지 30시간이 지나지 않았거나, Compute Optimizer에서 지원하지 않는 인스턴스 유형인 경우 이 문제가 생길 수 있습니다. 자세한 내용은 이전 섹션의 제한 사항 (p. 266) 단원을 참조하십시오.

권장 사항 보기

Compute Optimizer를 옵트인하면 EC2 콘솔에서 Compute Optimizer가 EC2 인스턴스에 대해 생성한 결과를 볼 수 있습니다. 그런 다음 Compute Optimizer 콘솔에 액세스하여 권장 사항을 볼 수 있습니다. 최근에 옵트인한 경우 검색 결과가 EC2 콘솔에 최대 12시간 동안 반영되지 않을 수 있습니다.

EC2 콘솔을 통해 EC2 인스턴스의 권장 사항을 보려면

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

3. 인스턴스를 선택하고 설명 탭에서 Finding(결과) 필드를 살펴봅니다. 세부 정보 보기를 선택합니다.

   Compute Optimizer에서 인스턴스가 열리고 여기서 현재 인스턴스로 레이블이 지정됩니다. 서로 다른 인스턴스 유형 권장 사항이 Option 1(옵션 1), Option 2(옵션 2), Option 3(옵션 3)이라는 레이블로 3개까지 제공됩니다. 창 아래쪽에 현재 인스턴스의 최근 CloudWatch 지표 데이터인 CPU 사용률, 메모리 사용률, 네트워크 입력, 네트워크 출력이 표시됩니다.

4. (선택 사항) Compute Optimizer 콘솔에서 설정( ) 아이콘을 선택하여 테이블에 표시된 열을 변경하거나 현재 및 권장 인스턴스 유형의 여러 가지 구매 옵션에 대해 공개된 요금 정보를 볼 수 있습니다.

   > **Note**
   >
   > 예약 인스턴스를 구매한 경우 온디맨드 인스턴스 요금이 예약 인스턴스로 청구될 수 있습니다. 현재 인스턴스 유형을 변경하기 전에 먼저 예약 인스턴스 사용률 및 적용 범위에 미치는 영향을 평가합니다.

권장 사항 중 하나를 사용할지 여부를 결정합니다. 성능 향상, 비용 절감 또는 이 두 가지를 조합하여 최적화할 것인지 결정합니다. 자세한 내용은 AWS Compute Optimizer 사용 설명서의 리소스 권장 사항 보기를 참조하십시오.

Compute Optimizer 콘솔을 통해 모든 리전의 모든 EC2 인스턴스에 대한 권장 사항을 보려면

1. https://console.aws.amazon.com/compute-optimizer/에서 Compute Optimizer 콘솔을 엽니다.
2. View recommendations for all EC2 instances(모든 EC2 인스턴스에 대한 권장 사항 보기)를 선택합니다.
3. 권장 사항 페이지에서 다음 작업을 수행할 수 있습니다.

   a. 하나 이상의 AWS 리전으로 권장 사항을 필터링하려면 Filter by one or more Regions(하나 이상의 리전별 필터링) 텍스트 상자에 리전 이름을 입력하거나, 표시되는 드롭다운 목록에서 리전을 하나 이상 선택합니다.

   b. 다른 계정의 리소스에 대한 권장 사항을 보려면 계정을 선택한 다음 다른 계정 ID를 선택합니다.

      이 옵션은 조직의 마스터 계정에 로그인하고 조직 내의 모든 구성원 계정을 옵트인한 경우에만 사용할 수 있습니다.

   c. 선택한 필터를 지우려면 Clear filters(필터 지우기)를 선택합니다.

   d. 현재 및 권장 인스턴스 유형에 대해 표시되는 구매 옵션을 변경하려면 설정( ) 아이콘을 선택한 다음 On-Demand Instances(온디맨드 인스턴스), Reserved Instances, standard 1-year no upfront(예약 인스턴스, 표준 1년 선결제 없음) 또는 Reserved Instances, standard 3-year no upfront(예약 인스턴스, 표준 3년 선결제 없음)를 선택합니다.

   e. 추가 권장 사항 및 사용률 지표 비교와 같은 세부 정보를 보려면 원하는 인스턴스 옆에 나열된 결과(Under-provisioned(프로비저닝 부족), Over-provisioned(프로비저닝 과다) 또는 Optimized(최적화))를 선택합니다. 자세한 내용은 AWS Compute Optimizer 사용 설명서의 리소스 세부 정보 보기를 참조하십시오.

권장 사항 평가를 위한 고려 사항

인스턴스 유형을 변경하기 전에 다음 사항을 고려하십시오.

- 권장 사항은 사용량을 예측하지 않습니다. 권장 사항은 최근 14일 기간 동안의 사용량을 기준으로 합니다. 향후 리소스 요구 사항을 충족할 것으로 예상되는 인스턴스 유형을 선택해야 합니다.
- 그래프로 표시된 지표를 집중적으로 살펴보고 실제 사용량이 인스턴스 용량보다 낮은지 확인합니다. 또한 CloudWatch에서 지표 데이터(평균, 피크, 백분위수)를 보고 EC2 인스턴스 권장 사항을 추가로 평가할 수 있습니다. 예를 들어 CPU 백분율 지표가 하루 동안 어떻게 변화하고 수용해야 하는 피크가 있는지 확인합니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 사용 가능한 지표 보기를 참조하십시오.

- Compute Optimizer는 T3, T3a, T2 인스턴스 등 성능 버스트가 가능한 인스턴스에 대한 권장 사항을 제공할 수 있습니다. 기준 이상으로 주기적으로 버스트하는 경우 새 인스턴스 유형의 vCPU에 따라 계속 버스트할 수 있어야 합니다. 자세한 정보는 성능 순간 확장 가능 인스턴스에 대한 CPU 크레딧 및 기준 성능 (p. 195) 단원을 참조하십시오.
- 예약 인스턴스를 구매한 경우 온디맨드 인스턴스 요금이 예약 인스턴스로 청구될 수 있습니다. 현재 인스턴스 유형을 변경하기 전에 먼저 예약 인스턴스 사용률 및 적용 범위에 미치는 영향을 평가합니다.
- 가능한 경우 최신 세대 인스턴스로의 변환을 고려합니다.
- 다른 인스턴스 패밀리로 마이그레이션할 때 현재 인스턴스 유형과 새 인스턴스 유형이 가상화, 아키텍처 또는 네트워크 유형 측면에서 호환되어야 합니다. 자세한 내용은 인스턴스 크기 조정을 위한 호환성 (p. 262) 단원을 참조하십시오.
- 마지막으로 각 권장 사항에 대해 제공되는 성능 위험 등급을 고려합니다. 성능 위험은 권장 인스턴스 유형이 워크로드의 성능 요구 사항을 충족하는지 여부를 검증하기 위해 얼마나 많은 노력을 기울여야 하는지를 나타냅니다. 또한 변경 전후에 엄격한 로드 및 성능 테스트를 수행하는 것이 좋습니다.

EC2 인스턴스 크기를 조정할 때 고려할 다른 내용도 있습니다. 자세한 내용은 인스턴스 유형 변경 (p. 262) 단원을 참조하십시오.

추가 리소스

- 인스턴스 유형 (p. 178)
- AWS Compute Optimizer 사용 설명서

인스턴스 구입 옵션

Amazon EC2는 사용자가 요구 사항에 따라 비용을 최적화할 수 있도록 다음과 같은 구입 옵션을 제공합니다.

- 온디맨드 인스턴스 - 시작하는 인스턴스에 대한 비용을 초 단위로 지불합니다.
- Savings Plans – 1년 또는 3년 기간 동안 시간당 USD로 일관된 사용량을 약정하여 Amazon EC2 비용을 절감할 수 있습니다.
- 예약 인스턴스 – 1년 또는 3년 기간 동안 인스턴스 유형 또는 지역을 포함해 일관된 인스턴스 구성을 약정하여 Amazon EC2 비용을 절감할 수 있습니다.
- 예약된 인스턴스 - 1년 동안 항상 사용할 수 있는 인스턴스를 지정된 되풀이 일정으로 구입합니다.
- 스팟 인스턴스 – 미사용 EC2 인스턴스를 요청하여 Amazon EC2 비용을 대폭 줄일 수 있습니다.
- 전용 호스트 - 인스턴스 실행을 전담하는 실제 호스트 비용을 지불하며, 기존의 소켓, 코어 또는 VM 소프트웨어별 라이선스를 가져와 비용을 절감합니다.
- 전용 인스턴스 - 단일 테넌트 하드웨어에서 실행되는 인스턴스 비용을 시간 단위로 지불합니다.
- 용량 예약 – 원하는 기간 동안 특정 가용 영역의 EC2 인스턴스에 대해 용량을 예약합니다.

용량을 예약해야 할 경우 특정 가용 영역에 대해 예약 인스턴스 또는 용량 예약을 구입하거나 정기 인스턴스를 구입하십시오. 스팟 인스턴스는 애플리케이션이 실행할 때와 중단될 수 있는 때를 유연하게 조정할 수 있는 경우 선택 가능한 비용 효율적인 방법입니다. 전용 호스트 또는 전용 인스턴스를 사용하면 기존 서버별 소프트웨어 라이선스를 사용하여 규정 준수 요구 사항을 해결하고 비용을 줄일 수 있습니다. 자세한 내용은 Amazon EC2 요금을 참조하십시오.

Savings Plans에 대한 자세한 내용은 AWS Savings Plans 사용 설명서를 참조하십시오.

목차

- 인스턴스 수명 주기 결정 (p. 269)
- 온디맨드 인스턴스 (p. 270)
- 예약 인스턴스 (p. 273)

- 정기 예약 인스턴스 (p. 308)
- 스팟 인스턴스 (p. 311)
- 전용 호스트 (p. 381)
- 전용 인스턴스 (p. 410)
- 온디맨드 용량 예약 (p. 415)

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 종료하면 끝납니다. 는 용량이 가용 상태이고 최고 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

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다.
4. 설명 탭에서 테넌시를 찾습니다. 값이 `host`인 경우 그 인스턴스는 전용 호스트에서 실행 중인 것입니다. 값이 `dedicated`인 경우 인스턴스는 전용 인스턴스입니다.
5. 설명 탭에서 수명 주기를 찾습니다. 값이 `spot`인 경우 인스턴스는 스팟 인스턴스입니다. 값이 `scheduled`인 경우 그 인스턴스는 정기 인스턴스입니다. 값이 `normal`인 경우 인스턴스는 온디맨드 인스턴스 또는 예약 인스턴스입니다.
6. (선택 사항) 예약 인스턴스를 구매했는데 적용되었는지 확인하고 싶은 경우 Amazon EC2에 대한 사용 보고서를 참고할 수 있습니다. 자세한 내용은 Amazon EC2 사용 보고서 (p. 1081) 단원을 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 실행 중인 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 `InstanceLifecycle`이 포함되지 않습니다.

온디맨드 인스턴스

온디맨드 인스턴스는 온디맨드 방식으로 사용하는 인스턴스입니다. 수명 주기를 완전하게 제어할 수 있습니다. 즉 시작, 중지, 수면, 사용 시작 또는 종료 시기를 결정할 수 있습니다.

온디맨드 인스턴스를 구매할 때 장기 약정은 필요 없습니다. 온디맨드 인스턴스가 `running` 상태인 시간(초)에 대해서만 지불하면 됩니다. 실행 중인 온디맨드 인스턴스에 대한 초당 비용은 고정 요금이며, 온디맨드 요금 페이지에서 확인할 수 있습니다.

중단할 수 없는 불규칙한 단기 워크로드가 있는 애플리케이션의 경우 온디맨드 인스턴스를 사용하는 것이 좋습니다.

온디맨드 인스턴스를 통해 비용을 대폭 절감하려면 AWS Savings Plans, 스팟 인스턴스 (p. 311) 또는 예약 인스턴스 (p. 273)를 사용합니다.

목차

- 온디맨드 인스턴스 작업 (p. 270)
- 온디맨드 인스턴스 제한 (p. 270)
  - 필요한 vCPU 수 계산 (p. 271)
  - 제한 증가 요청 (p. 273)
  - 온디맨드 인스턴스 제한 및 사용량 모니터링 (p. 273)
- AWS 서비스 요금 쿼리 (p. 273)

온디맨드 인스턴스 작업

다음과 같은 방식으로 온디맨드 인스턴스 작업을 수행할 수 있습니다.

- 인스턴스 시작 (p. 430)
- Linux 인스턴스에 연결 (p. 485)
- 인스턴스 중지 및 시작 (p. 507)
- Linux 인스턴스를 최대 절전 모드로 실행합니다. (p. 510)
- 인스턴스 재부팅 (p. 520)
- 인스턴스 만료 (p. 520)
- 인스턴스 종료 (p. 522)
- 인스턴스 복구 (p. 528)
- Amazon Linux 인스턴스 구성 (p. 529)
- EC2 Linux 인스턴스 식별 (p. 598)

Amazon EC2를 처음 사용한다면 먼저 처음으로 Amazon EC2 사용하기 (p. 1) 단원을 참조하십시오.

온디맨드 인스턴스 제한

리전별로 AWS 계정당 온디맨드 인스턴스 실행 수에는 제한이 있습니다. 온디맨드 인스턴스 제한은 인스턴스 유형과 상관없이 실행 중인 온디맨드 인스턴스가 사용하는 가상 중앙 처리 장치(vCPU) 수를 기준으로 관리됩니다.

다음 표와 같은 다섯 가지 온디맨드 인스턴스 제한이 있습니다. 각 제한은 하나 이상의 인스턴스 패밀리에 vCPU 제한을 지정합니다. 다른 인스턴스 패밀리에 대한 자세한 내용은 Amazon EC2 인스턴스 유형 단원을 참조하십시오.

온디맨드 인스턴스 제한 이름	기본 vCPU 제한
온디맨드 표준(A, C, D, H, I, M, R, T, Z) 인스턴스 실행	vCPU 1152개
온디맨드 F 인스턴스 실행	vCPU 128개
온디맨드 G 인스턴스 실행	vCPU 128개
온디맨드 Inf 인스턴스 실행	vCPU 128개
온디맨드 P 인스턴스 실행	vCPU 128개
온디맨드 X 인스턴스 실행	vCPU 128개

> **Note**
>
> 새 AWS 계정의 경우 여기에 설명된 것보다 더 낮은 수로 제한하여 시작할 수 있습니다.

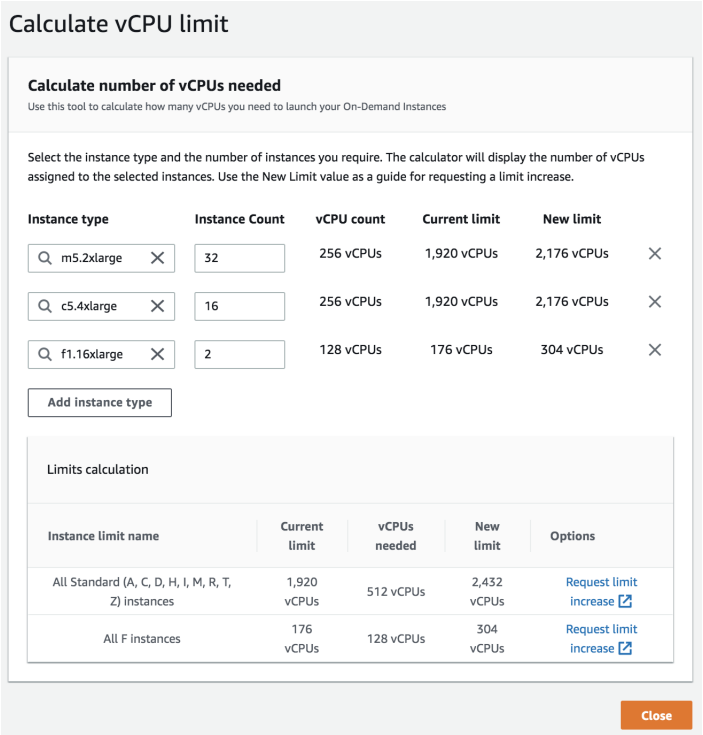
vCPU 제한을 사용하면 변화하는 애플리케이션 요구 사항을 충족하는 모든 인스턴스 유형 조합을 시작하는 데 필요한 vCPU 수에 따라 제한을 사용할 수 있습니다. 예를 들어 표준 인스턴스 제한이 vCPU 256개인 경우 32개의 `m5.2xlarge` 인스턴스(32x8 vCPU), 16개의 `c5.4xlarge` 인스턴스(16x16 vCPU) 또는 표준 인스턴스 유형 및 크기를 총 256개의 vCPU로 맞춘 조합을 시작할 수 있습니다. 자세한 내용은 EC2 온디맨드 인스턴스 제한 단원을 참조하십시오.

필요한 vCPU 수 계산

vCPU 제한 계산기를 사용하여 애플리케이션 요구에 맞게 필요한 vCPU 수를 결정할 수 있습니다.

계산기를 사용할 때는 다음 사항에 주의하십시오. 계산기는 현재 제한에 도달했다고 가정합니다. 인스턴스 수에 입력하는 값은 현재 제한에서 허용되는 인스턴스에 추가하여 시작해야 하는 인스턴스 수입니다. 계산기는 현재 제한을 인스턴스 수에 추가하여 새로운 제한에 도달합니다.

다음 스크린샷은 vCPU 제한 계산기를 보여줍니다.



다음 컨트롤과 정보를 확인하고 사용할 수 있습니다.

- 인스턴스 유형 – vCPU 제한 계산기에 추가하는 인스턴스 유형입니다.
- 인스턴스 수 – 선택한 유형에 필요한 인스턴스 수입니다.
- vCPU 수 – 인스턴스 수에 해당하는 vCPU 수입니다.
- 현재 제한 – 인스턴스 유형이 속한 제한 유형의 현재 제한입니다. 제한은 제한 유형이 동일한 모든 인스턴스 유형에 적용됩니다. 예를 들어 위의 스크린샷에서 `m5.2xlarge` 및 `c5.4xlarge`에 대한 현재 제한은 vCPU 1,920개이며, 모든 표준 인스턴스 제한에 속한 모든 인스턴스 유형에 대한 제한이 됩니다.
- 새 제한 – vCPU 수와 현재 제한을 추가하여 계산되는 새로운 제한(vCPU 수)입니다.
- X – 행을 제거하려면 X를 선택합니다.
- Add instance type(인스턴스 유형 추가) – 계산기에 다른 인스턴스 유형을 추가하려면 Add instance type(인스턴스 유형 추가)을 선택합니다.
- Limits calculation(제한 계산) – 현재 제한, 필요한 vCPU 수 및 제한 유형에 대한 새 제한을 표시합니다.
  - Instance limit name(인스턴스 제한 이름) – 선택한 인스턴스 유형에 대한 제한 유형입니다.
  - 현재 제한 – 제한 유형에 대한 현재 제한입니다.
  - 필요한 vCPU 수 – 인스턴스 수에서 지정한 인스턴스 수에 해당하는 vCPU 수입니다. 모든 표준 인스턴스 제한에 필요한 vCPU 수는 이 제한 유형의 모든 인스턴스 유형에 대한 vCPU count(vCPU 수) 값을 더해 계산됩니다.
  - 새 제한 – 새로운 제한은 현재 제한과 필요한 vCPU 수를 추가하여 계산됩니다.
  - 옵션 – 해당하는 제한 유형에 대한 제한 증가를 요청하려면 사용량 제한 증가 요청을 선택합니다.

필수 vCPU 수를 계산하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 모음에서 리전을 선택합니다.
3. 왼쪽 탐색기에서 제한을 선택합니다.
4. Calculate vCPU limit(vCPU 제한 계산)을 선택합니다.

5. Add instance type(인스턴스 유형 추가)을 선택하고, 필수 인스턴스 유형을 선택하고, 필수 인스턴스 수를 지정합니다. 인스턴스 유형을 더 추가하려면 Add instance type(인스턴스 유형 추가)를 다시 선택합니다.
6. 필수 새 제한에 대한 Limits calculation(제한 계산)을 확인합니다.
7. 계산기를 사용하여 종료한 경우 닫기를 선택합니다.

제한 증가 요청

Limits page(제한 페이지) 또는 Amazon EC2 콘솔의 vCPU 제한 계산기에서 각 온디맨드 인스턴스 제한 유형에 대한 제한 증가를 요청할 수 있습니다. 사용 사례를 활용하여 AWS 지원 센터의 제한 증가 양식에서 필수 필드를 입력하면 됩니다. Primary Instance Type(기본 인스턴스 유형)에서 vCPU 제한 계산기의 Instance limit name(인스턴스 제한 이름)에 상응하는 제한 유형을 선택합니다. 새 제한 값에는 vCPU 제한 계산기의 New limit(새 제한) 열에 표시되는 값을 사용합니다. 제한 증가 요청에 대한 자세한 내용은 Amazon EC2 서비스 제한 (p. 1079) 단원을 참조하십시오.

온디맨드 인스턴스 제한 및 사용량 모니터링

Amazon EC2 콘솔의 제한 페이지, 서비스 할당량 콘솔의 Amazon EC2 서비스 할당 페이지, AWS Trusted Advisor 콘솔의 서비스 제한 페이지에서 온디맨드 인스턴스 제한을 확인하고 관리할 수 있습니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 Amazon EC2 서비스 제한 (p. 1079), 서비스 할당량 사용 설명서의 서비스 할당량 확인, AWS Trusted Advisor를 참조하십시오.

Amazon CloudWatch 지표를 통해 제한에 대한 EC2 사용량을 확인할 수 있습니다. 제한 도달에 대해 경고를 받도록 경보를 구성할 수도 있습니다. 자세한 내용은 서비스 할당량 사용 설명서의 Amazon CloudWatch 경보 사용을 참조하십시오.

AWS 서비스 요금 쿼리

Price List Service API 또는 AWS Price List API를 사용하여 온디맨드 인스턴스 요금을 쿼리할 수 있습니다. 자세한 내용은 AWS Billing and Cost Management 사용 설명서의 AWS Price List API 사용을 참조하십시오.

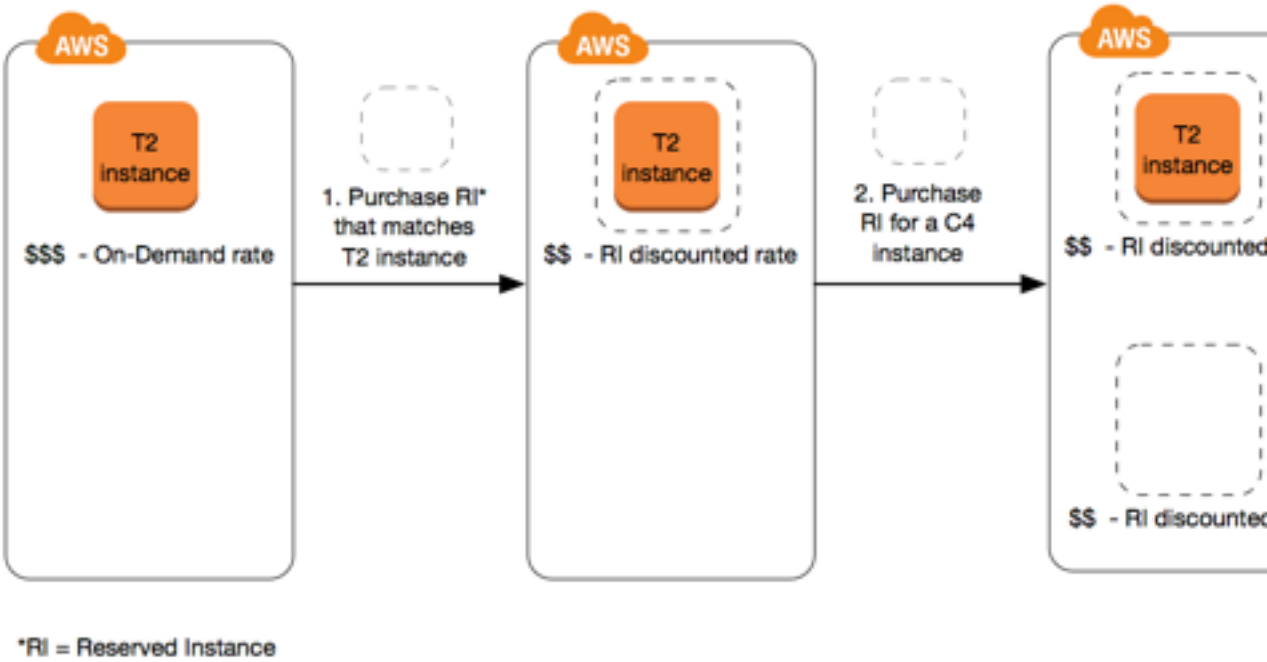
예약 인스턴스

예약 인스턴스를 이용하면 온디맨드 인스턴스 요금에 비해 Amazon EC2 비용을 상당히 절감할 수 있습니다. 예약 인스턴스는 물리적 인스턴스가 아니며 계정에서 온디맨드 인스턴스를 사용할 때 적용되는 결제 할인에 가깝습니다. 이러한 온디맨드 인스턴스의 경우 결제 할인 혜택을 받으려면 인스턴스 유형 및 지역과 같은 특정 속성에 부합해야 합니다.

또한 Savings Plans는 온디맨드 인스턴스 요금에 비해 상당한 Amazon EC2 비용 절감을 제공합니다. Savings Plans를 이용해 시간당 USD로 측정되는 일관된 사용량을 약정할 수 있습니다. Savings Plans는 특정 인스턴스 구성을 약정하는 것이 아니라 사용자의 요구 사항에 가장 적합하고 지속적으로 비용을 절약해 주는 인스턴스 구성을 사용할 수 있는 유연성을 제공합니다. 자세한 내용은 AWS Savings Plans 사용 설명서를 참조하십시오.

예약 인스턴스 개요

다음 다이어그램에는 예약 인스턴스 구입 및 사용에 대한 기본 개요가 나와 있습니다.



이 시나리오에서, 현재 온디맨드 요금으로 지불하고 있는 온디맨드 인스턴스(T2)가 사용자의 계정에서 실행되고 있습니다. 사용자는 실행되고 있는 인스턴스의 속성과 일치하는 예약 인스턴스를 구입하며, 결제 혜택이 즉시 적용됩니다. 그런 다음 C4 인스턴스에 대한 예약 인스턴스를 구입합니다. 이 예약 인스턴스의 속성과 일치하는 인스턴스가 계정에서 실행되고 있지 않습니다;. 최종 단계에서 사용자는 C4 예약 인스턴스의 속성과 일치하는 인스턴스를 시작하며, 결제 혜택이 즉시 적용됩니다.

예약 인스턴스 요금을 결정하는 주요 변수

예약 인스턴스 요금은 다음과 같은 주요 변수에 의해 결정됩니다.

인스턴스 속성

예약 인스턴스에는 요금을 결정하는 4개의 인스턴스 속성이 있습니다. 이러한 속성은 계정의 실행 중인 인스턴스에 예약 인스턴스가 적용되는 방식도 결정합니다.

- 인스턴스 유형: 예를 들어, `m4.large`입니다. 이는 인스턴스 패밀리(`m4`) 및 인스턴스 크기(`large`)로 구성됩니다.
- 리전: 예약 인스턴스를 구매한 리전입니다.
- 테넌시: 인스턴스가 공유된 하드웨어(기본)에서 실행되는지 단일 테넌트(전용) 하드웨어에서 실행되는지 여부입니다. 자세한 내용은 전용 인스턴스 (p. 410) 단원을 참조하십시오.
- 플랫폼: Windows 또는 Linux/Unix와 같은 운영 체제입니다. 자세한 내용은 플랫폼 선택 (p. 286) 단원을 참조하십시오.

예약 인스턴스는 자동으로 갱신되지 않으므로 만료될 경우 중단 없이 EC2 인스턴스를 계속 사용할 수 있지만 온디맨드 요금이 부과됩니다. 위 예에서 T2 및 C4 인스턴스에 적용되는 예약 인스턴스가 만료되면 사용자가 인스턴스를 종료하거나 인스턴스 속성과 일치하는 새 예약 인스턴스를 구입할 때까지 온디맨드 요금 결제로 돌아갑니다.

약정

1년 약정 또는 더 큰 할인을 제공하는 3년 약정으로 예약 인스턴스를 구입할 수 있습니다.

- 1년: 1년은 31536000초(365일)로 정의됩니다.
- 3년: 3년은 94608000초(1095일)로 정의됩니다.

결제 옵션

예약 인스턴스에 사용할 수 있는 결제 옵션은 다음과 같습니다.

- 전체 선결제: 기간이 시작되는 시점에서 모든 금액을 결제하고 사용 기간 동안 기타 비용이나 추가 시간당 요금 없이 무제한으로 사용할 수 있습니다.
- 부분 선결제: 비용 중 일부를 먼저 결제해야 하며, 결제하지 않은 시간에 대해서는 예약 인스턴스가 사용되는지 여부와 상관없이 할인된 시간당 요금이 청구됩니다.
- 선결제 없음: 예약 인스턴스가 사용되는지 여부와 상관없이 사용 기간 동안 매시간마다 할인된 시간당 요금이 청구됩니다. 선결제 금액이 필요하지 않습니다.

  Note

  선결제가 없는 예약 인스턴스는 전체 예약 기간 동안 매월 결제해야 하는 계약 조건입니다. 따라서 선결제가 없는 예약 인스턴스를 구입할 수 있으려면 결제 기록에 미납액이 없어야 합니다.

일반적으로 예약 인스턴스에 대한 선결제 금액이 높을수록 요금 절약 혜택이 커집니다. 또한 예약 인스턴스 마켓플레이스에서 타사 판매업체가 제공하는 저렴하고 기간이 짧은 예약 인스턴스를 찾을 수도 있습니다. 자세한 내용은 예약 인스턴스 마켓플레이스 (p. 291) 단원을 참조하십시오.

제공 클래스

컴퓨팅 요건이 변경되면 제공 클래스에 따라 예약 인스턴스를 수정 또는 교환할 수 있습니다.

- 표준: 가장 큰 할인 혜택을 제공하지만 수정만 가능합니다.
- 컨버터블: 표준 예약 인스턴스보다 낮은 할인 혜택을 제공하지만 다른 인스턴스 속성을 포함하는 다른 컨버터블 예약 인스턴스와 교환 가능합니다. 컨버터블 예약 인스턴스는 수정도 가능합니다.

자세한 내용은 예약 인스턴스 유형(제공 클래스) (p. 276)을 참조하십시오.

예약 인스턴스를 구입한 이후에는 구입을 취소할 수 없습니다. 그러나 변경이 필요한 경우 예약 인스턴스를 수정 (p. 297), 교환 (p. 304) 또는 판매 (p. 291)할 수 있습니다.

요금에 대한 자세한 내용은 Amazon EC2 예약 인스턴스 요금을 참조하십시오.

예약 인스턴스 제한

조직에서 보유할 수 있는 예약 인스턴스 수는 제한되어 있습니다. 각 리전에 대해 20개의 리전 (p. 277) 예약 인스턴스와, 각 가용 영역에 대해 달마다 추가로 20개의 영역 (p. 277) 예약 인스턴스를 구입할 수 있습니다.

예를 들어 가용 영역이 3개인 리전에서 한도는 월당 예약 인스턴스 80개입니다. 즉, 리전에 대해 리전 예약 인스턴스 20개와, 각 가용 영역 3개에 대해 영역 예약 인스턴스 20개(20x3=60)입니다.

리전 예약 인스턴스는 실행 중인 온디맨드 인스턴스에 할인을 적용합니다. 기본 온디맨드 인스턴스 제한은 20개입니다. 리전 예약 인스턴스 구매로 실행 중인 온디맨드 인스턴스 제한을 초과할 수는 없습니다. 예를 들어 이미 20개의 온디맨드 인스턴스를 실행 중이고 20개의 리전 예약 인스턴스를 구매한 경우 20개의 리전 예약 인스턴스는 20개의 실행 중인 온디맨드 인스턴스에 할인을 적용합니다. 리전 예약 인스턴스를 추가로 구매한 경우 온디맨드 인스턴스 제한에 도달했기 때문에 더 많은 인스턴스를 시작할 수는 없습니다.

영역별 예약 인스턴스를 구입하기 전에 온디맨드 인스턴스 제한이 소유하려는 리전별 예약 인스턴스의 수와 일치하는지 아니면 초과하는지 확인합니다. 필요한 경우 추가 리전별 예약 인스턴스를 구입하기 전에 온디맨드 인스턴스 제한 증가를 요청해야 합니다.

예약 인스턴스 특정 가용 영역에 대해 구매한 영역 예약 인스턴스는 할인은 물론 용량 예약을 제공합니다. 영역 예약 인스턴스 구매를 통해 실행 중인 온디맨드 인스턴스 제한을 초과할 수 있습니다. 예를 들어 이미 20개의 온디맨드 인스턴스를 실행 중이고 20개의 영역 예약 인스턴스를 구매한 경우 영역 예약 인스턴스의 사양과 일치하는 20개의 온디맨드 인스턴스를 추가로 시작할 수 있어, 총 40개의 인스턴스를 실행할 수 있습니다.

Amazon EC2 콘솔은 한도 정보를 제공합니다. 자세한 내용은 현재 제한 조회 (p. 1079)을 참조하십시오.

리전 및 영역 예약 인스턴스(범위)

예약 인스턴스를 구입할 때 예약 인스턴스의 범위를 결정합니다. 범위는 리전 또는 영역입니다.

- 리전: 리전에 대해 예약 인스턴스를 구입하는 경우 이를 리전 예약 인스턴스라고 합니다.
- 영역: 특정 가용 영역에 대해 예약 인스턴스를 구입하는 경우 이를 영역 예약 인스턴스라고 합니다.

리전 및 영역 예약 인스턴스의 차이점

다음 표에서는 리전 예약 인스턴스와 영역 예약 인스턴스의 주요 차이점 중 일부를 요약하여 설명합니다.

	리전 예약 인스턴스	영역 예약 인스턴스
가용 영역 유연성	지정된 리전에 있는 모든 가용 영역의 인스턴스 사용량에 예약 인스턴스 할인이 적용됩니다.	가용 영역 유연성 없음 — 지정된 가용 영역의 인스턴스 사용량에만 예약 인스턴스 할인이 적용됩니다.
용량 예약	용량 예약 없음 — 리전 예약 인스턴스에서 용량 예약을 제공하지 않습니다.	영역 예약 인스턴스에서는 지정된 가용 영역의 용량 예약을 제공합니다.
인스턴스 크기 유연성	크기에 상관없이 인스턴스 패밀리 내 인스턴스 사용량에 예약 인스턴스 할인이 적용됩니다. 기본 테넌시가 있는 Amazon Linux/Unix 예약 인스턴스에 대해서만 지원됩니다. 자세한 내용은 정규화 인자에 의해 결정되는 인스턴스 크기 유연성 (p. 277)를 참조하십시오.	인스턴스 크기 유연성 없음 — 지정된 인스턴스 유형 및 크기의 인스턴스 사용량에만 예약 인스턴스 할인이 적용됩니다.

자세한 정보와 지침은 예약 인스턴스 적용 방식 (p. 277) 단원을 참조하십시오.

예약 인스턴스 유형(제공 클래스)

예약 인스턴스를 구입할 경우 표준 또는 전환형 제공 클래스 중에서 선택할 수 있습니다. 예약 인스턴스는 해당 기간 동안 단일 인스턴스 유형, 플랫폼, 범위 및 테넌시에 적용됩니다. 컴퓨팅 요건이 변경되면 제공 클래스에 따라 예약 인스턴스를 수정 또는 교환할 수 있습니다. 제공 클래스에도 추가 제한이 있을 수 있습니다.

표준 및 전환형 제공 클래스 간의 차이점은 다음과 같습니다.

표준 예약 인스턴스	전환형 예약 인스턴스
해당 기간 동안 인스턴스 크기와 같은 일부 속성을 수정할 수 있지만 인스턴스 패밀리는 수정할 수 없습니다. 표준 예약 인스턴스는 교환할 수 없으며 수	기간 동안 인스턴스 패밀리, 인스턴스 유형, 플랫폼, 범위 또는 테넌시를 비롯한 새 속성이 있는 다른 전환형 예약 인스턴스와 교환할 수 있습니다. 자세한 내용은 전환형 예약 인스턴스 교환 (p. 304) 단원

표준 예약 인스턴스	전환형 예약 인스턴스
정만 가능합니다. 자세한 내용은 예약 인스턴스 수정 (p. 297) 단원을 참조하십시오.	을 참조하십시오. 전환형 예약 인스턴스의 일부 속성을 수정할 수도 있습니다. 자세한 내용은 예약 인스턴스 수정 (p. 297) 단원을 참조하십시오.
예약 인스턴스 마켓플레이스에서 판매할 수 있습니다.	예약 인스턴스 마켓플레이스에서 판매할 수 없습니다.

표준 및 전환형 예약 인스턴스는 특정 가용 영역의 인스턴스(영역 예약 인스턴스) 또는 리전의 인스턴스(리전 예약 인스턴스)에 적용하기 위해 구입할 수 있습니다. 자세한 정보와 지침은 예약 인스턴스 적용 방식 (p. 277) 단원을 참조하십시오.

지정된 시작 시간과 기간에 따라 매일, 매주 또는 매월 반복적으로 용량 예약을 구매사고 싶다면 정기 예약 인스턴스가 필요 사항을 충족할 수 있습니다. 자세한 내용은 정기 예약 인스턴스 (p. 308) 단원을 참조하십시오.

예약 인스턴스 적용 방식

예약 인스턴스를 구입했으며 예약 인스턴스의 사양과 일치하는 인스턴스가 이미 실행 중인 경우 결제 혜택이 즉시 적용됩니다. 인스턴스를 따로 재시작할 필요가 없습니다. 실행 중인 적용 대상 인스턴스가 없는 경우, 인스턴스를 시작하고 예약 인스턴스에 대해 지정한 동일한 조건과 일치하는지 확인합니다;. 자세한 내용은 예약 인스턴스 사용 (p. 291) 단원을 참조하십시오.

예약 인스턴스는 제공 유형(표준 또는 전환형)과 상관없이 동일한 방식으로 사용량에 적용되므로 속성이 일치하는 실행 중인 온디맨드 인스턴스에 자동으로 적용됩니다.

영역 예약 인스턴스 적용 방식

예약 인스턴스를 특정 가용 영역에 할당하면 해당 가용 영역에서 일치하는 인스턴스 사용량에 대해서는 예약 인스턴스 할인 혜택이 제공됩니다. 예를 들어 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 속성(테넌시, 플랫폼, 가용 영역, 인스턴스 유형 및 인스턴스 크기)이 예약 인스턴스의 속성과 일치해야 합니다.

리전 단위 예약 인스턴스 적용 방식

리전 예약 인스턴스는 리전용으로 구입되며 가용 영역 유연성을 제공합니다. 해당 리전에 있는 모든 가용 영역의 인스턴스 사용량에 예약 인스턴스 할인이 적용됩니다.

리전 예약 인스턴스도 크기에 상관없이 인스턴스 패밀리 내 인스턴스 사용량에 예약 인스턴스 할인이 적용되는 인스턴스 크기 유연성을 제공합니다.

인스턴스 크기 유연성에 대한 제한

다음 예약 인스턴스에는 인스턴스 크기 유연성이 적용되지 않습니다.

- 특정 가용 영역(영역 예약 인스턴스)용으로 구입한 예약 인스턴스
- 전용 테넌시를 포함하는 예약 인스턴스
- Windows Server, Windows Server with SQL Standard, Windows Server with SQL Server Enterprise, Windows Server with SQL Server Web, RHEL 및 SLES용 예약 인스턴스
- G4 인스턴스는 예약 인스턴스 단원을 참조하십시오.

정규화 인자에 의해 결정되는 인스턴스 크기 유연성

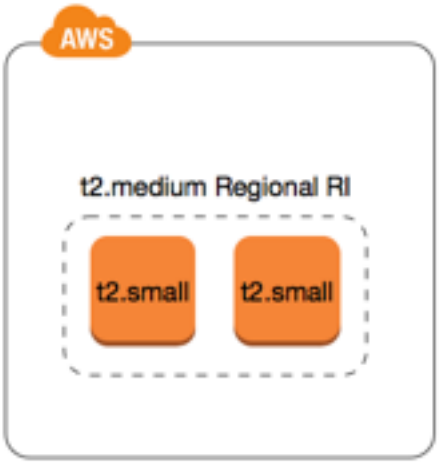
인스턴스 크기 유연성은 인스턴스 크기의 정규화 인자에 의해 결정됩니다. 리전의 모든 가용 영역에서 예약의 인스턴스 크기에 따라 모든 할인 또는 일부 할인이 동일한 인스턴스 패밀리의 실행 중인 인스턴스에 적용됩니다. 속성 중 인스턴스 패밀리, 테넌시 및 플랫폼만 일치하면 됩니다.

인스턴스 크기 유연성은 정규화 인자에 따라 인스턴스 패밀리 내 가장 작은 인스턴스 크기에서 가장 큰 인스턴스 크기에 이르기까지 두루 적용됩니다.

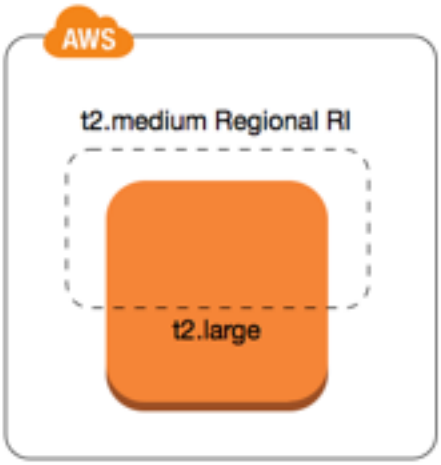
다음 표는 인스턴스 패밀리 내 서로 다른 크기 및 그에 따른 시간당 정규화 인자를 나열한 것입니다. 이 배율은 예약 인스턴스의 할인 요금을 정규화된 인스턴스 패밀리 사용량에 적용하는 데 사용됩니다.

인스턴스 크기	정규화 인자
nano	0.25
micro	0.5
small	1
medium	2
large	4
xlarge	8
2xlarge	16
3xlarge	24
4xlarge	32
6xlarge	48
8xlarge	64
9xlarge	72
10xlarge	80
12xlarge	96
16xlarge	128
18xlarge	144
24xlarge	192
32xlarge	256

예를 들어, `t2.medium` 인스턴스의 정규화 인자는 2입니다. 미국 동부(버지니아 북부)에서 `t2.medium` 기본 테넌시 Amazon Linux/Unix 예약 인스턴스를 구입하고, 해당 리전의 계정에 `t2.small` 인스턴스 2개가 실행 중인 경우 결제 혜택이 두 인스턴스에 전체적으로 적용됩니다.



또는 미국 동부(버지니아 북부) 리전의 계정에 실행 중인 `t2.large` 인스턴스 1개가 있는 경우 결제 혜택은 인스턴스 사용량의 50%에 적용됩니다.



예약 인스턴스를 수정하면 정규화 인자 역시 적용됩니다. 자세한 내용은 예약 인스턴스 수정 (p. 297) 단원을 참조하십시오.

베어 메탈 인스턴스에 대한 정규화 인자

인스턴스 크기 유연성은 인스턴스 패밀리 내 베어 메탈 인스턴스에도 적용됩니다. 베어 메탈 인스턴스에서 공유 테넌시를 포함하는 리전 Amazon Linux/Unix 예약 인스턴스가 있는 경우 동일한 인스턴스 패밀리 내에서 예약 인스턴스 절감 혜택을 얻을 수 있습니다. 반대의 경우도 마찬가지입니다. 베어 메탈 인스턴스와 동일한 패밀리의 인스턴스에서 공유 테넌시를 포함하는 리전 Amazon Linux/Unix 예약 인스턴스가 있는 경우 베어 메탈 인스턴스에서 예약 인스턴스 절감 혜택을 얻을 수 있습니다.

베어 메탈 인스턴스의 크기는 동일한 인스턴스 패밀리 내의 가장 큰 인스턴스의 크기와 동일합니다. 예를 들어 `i3.metal` 및 `i3.16xlarge`의 크기가 서로 동일하므로 해당 정규화 인자도 동일합니다.

> **Note**
>
> `.metal` 인스턴스 크기에는 단일 정규화 인자가 없습니다. 이 크기는 인스턴스 패밀리에 따라 달라집니다.

베어 메탈 인스턴스 크기	정규화 인자
`c5.metal`	192
`i3.metal`	128
`r5.metal`	192
`r5d.metal`	192
`z1d.metal`	96
`m5.metal`	192
`m5d.metal`	192

예를 들어 `i3.metal` 인스턴스의 정규화 인자는 128입니다. 미국 동부(버지니아 북부)에서 `i3.metal` 기본 테넌시 Amazon Linux/Unix 예약 인스턴스를 구입하면 다음과 같은 요금 혜택이 적용될 수 있습니다.

- 리전에서 사용자의 계정에 실행 중인 `i3.16xlarge`가 하나인 경우 `i3.16xlarge` 인스턴스(`i3.16xlarge` 정규화 인자 = 128)에 모든 요금 혜택이 적용됩니다.
- 또는 리전에서 사용자의 계정에 실행 중인 `i3.8xlarge` 인스턴스가 두 개인 경우 두 `i3.8xlarge` 인스턴스(`i3.8xlarge` 정규화 인자 = 64)에 모든 요금 혜택이 적용됩니다.
- 또는 리전에서 사용자의 계정에 실행 중인 `i3.4xlarge` 인스턴스가 4개인 경우 모든 4개의 `i3.4xlarge` 인스턴스(`i3.4xlarge` 정규화 인자 = 32)에 모든 요금 혜택이 적용됩니다.

반대의 경우도 마찬가지입니다. 예를 들어 미국 동부(버지니아 북부)에서 두 개의 `i3.8xlarge` 기본 테넌시 Amazon Linux/Unix 예약 인스턴스를 구입하고 리전에서 한 개의 `i3.metal` 인스턴스를 실행 중인 경우 `i3.metal` 인스턴스에 모든 요금 혜택이 적용됩니다.

예약 인스턴스 적용의 예

다음 시나리오에서 예약 인스턴스가 적용되는 방식을 알 수 있습니다.

Example 시나리오 1: 단일 계정의 예약 인스턴스

계정 A에서 다음 온디맨드 인스턴스를 실행 중입니다.

- 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

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-1 리전의 기본 테넌시 예약 인스턴스
- 1 x `c4.large` Amazon Linux, us-east-1 리전의 기본 테넌시 예약 인스턴스

예약 인스턴스의 혜택은 다음과 같이 적용됩니다.

- 인스턴스 간에 속성(인스턴스 크기, 리전, 플랫폼, 테넌시)이 서로 일치하므로 `m3.large` 영역 예약 인스턴스 4개의 할인 및 용량 예약이 `m3.large` 인스턴스 4개에 적용됩니다.
- `m4.large` 리전 예약 인스턴스에서는 기본 테넌시가 포함된 리전 단위의 Amazon Linux 예약 인스턴스이므로 가용 영역 및 인스턴스 크기 유연성을 제공합니다.

`m4.large`는 시간당 정규화 유닛 4개와 같습니다.

`m4.large` 리전 예약 인스턴스를 4개 구입하였으며, 이에 따라 시간당 정규화 유닛은 총 16개(4x4)입니다. 현재 계정 A에는 실행 중인 `m4.xlarge` 인스턴스가 2개이며, 이에 따라 시간당 정규화 유닛은 16개(2x8)와 같습니다. 이 경우 `m4.large` 리전 예약 인스턴스 4개에서는 `m4.xlarge` 인스턴스 2개의 전체 시간 사용량에 결제 혜택을 제공합니다.

- us-east-1의 `c4.large` 리전 예약 인스턴스는 기존 테넌시가 포함된 지역 Amazon Linux 예약 인스턴스이므로 가용 영역 및 인스턴스 크기 유연성을 `c4.xlarge` 인스턴스에 적용합니다. `c4.large` 인스턴스는 시간당 정규화 유닛 4개와 같고, `c4.xlarge`는 시간당 정규화 유닛 8개와 같습니다.

  이 경우에는 `c4.large` 리전 예약 인스턴스가 `c4.xlarge` 사용량에 부분적 혜택을 제공합니다. 이는 `c4.large` 예약 인스턴스가 사용량의 시간당 정규화 유닛이 4개와 같지만 `c4.xlarge` 인스턴스는 시간당 정규화 유닛이 8개가 필요하기 때문입니다. 따라서 `c4.large` 예약 인스턴스의 결제 할인이 `c4.xlarge` 사용량의 50%에 적용됩니다. 나머지 `c4.xlarge` 사용량은 온디맨드 요금이 부과됩니다.

Example 시나리오 2: 연결된 계정의 리전 단위 예약 인스턴스

예약 인스턴스가 구입 계정 내 사용량에 먼저 적용된 후 조직의 다른 계정에서 해당하는 사용량에 적용됩니다. 자세한 내용은 예약 인스턴스 및 통합 결제 (p. 284) 단원을 참조하십시오. 인스턴스 크기 유연성을 제공하는 리전 단위 예약 인스턴스의 경우 혜택은 인스턴스 패밀리 내 가장 작은 인스턴스 크기에서 가장 큰 인스턴스 크기에 이르기까지 두루 적용됩니다.

계정 A(구입 계정)에서 다음과 같은 온디맨드 인스턴스를 실행 중입니다.

- 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

다른 고객이 연결 계정인 계정 B에서 다음과 같은 온디맨드 인스턴스를 실행 중입니다.

- 2 x `m4.xlarge` Linux, 가용 영역 us-east-1a의 기본 테넌시 인스턴스

계정 A에서 다음과 같은 리전 단위 예약 인스턴스를 구입합니다.

- 4 x `m4.xlarge` Linux, us-east-1 리전의 기본 테넌시 예약 인스턴스
- 2 x `c4.xlarge` Linux, us-east-1 리전의 기본 테넌시 예약 인스턴스

지역 예약 인스턴스의 혜택은 다음과 같이 적용됩니다.

- 4개의 `m4.xlarge` 예약 인스턴스 할인은 계정 A(구입 계정)의 두 개의 `m4.xlarge` 인스턴스 및 하나의 `m4.2xlarge` 인스턴스에서 사용됩니다. 세 개의 인스턴스 모두 속성(인스턴스 패밀리, 리전, 패밀리, 테넌시)과 일치합니다. 계정 B(연결된 계정)에 예약 인스턴스와도 일치하는 두 개의 `m4.xlarge`가 있어도 먼저 구입 계정(계정 A)의 인스턴스에 할인이 적용됩니다. 예약 인스턴스가 리전 예약 인스턴스이므로 용량 예약이 없습니다.
- 2개의 `c4.xlarge` 예약 인스턴스 할인이 2개의 `c4.xlarge` 인스턴스에 적용되는데, 이는 `c4.2xlarge` 인스턴스보다 인스턴스 크기가 작기 때문입니다. 예약 인스턴스가 리전 예약 인스턴스이므로 용량 예약이 없습니다.

Example 시나리오 3: 연결된 계정의 영역 예약 인스턴스

일반적으로 계정에 속한 예약 인스턴스가 해당 계정의 사용량에 먼저 적용됩니다. 하지만 조직 내 다른 계정에 특정 가용 영역에 대해 자격을 갖추었지만 사용하지 않은 예약 인스턴스(영역 예약 인스턴스)가 있다면 계

정에 속한 리전 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.

계정 A(구입 계정)에서 다음과 같은 온디맨드 인스턴스를 실행하고 있습니다.

- 1 x `m4.xlarge` Linux, 가용 영역 us-east-1a의 기본 테넌시 인스턴스

고객이 연결된 계정 B에서 다음과 같은 온디맨드 인스턴스를 실행하고 있습니다.

- 1 x `m4.xlarge` Linux, 가용 영역 us-east-1b의 기본 테넌시 인스턴스

계정 A에서 다음과 같은 리전 단위 예약 인스턴스를 구입합니다.

- 1 x `m4.xlarge` Linux, us-east-1 리전의 기본 테넌시 예약 인스턴스

고객은 또한 연결된 계정 C에서 다음과 같은 영역 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` 영역 예약 인스턴스의 할인은 계정 A의 `m4.xlarge` 사용량에 적용됩니다.
- 계정 A에 속한 `m4.xlarge` 리전 예약 인스턴스의 할인은 계정 B의 `m4.xlarge` 사용량에 적용됩니다.
- 계정 A에 속한 리전 예약 인스턴스가 계정 A의 사용량에 먼저 적용된 경우에는 계정 C에 속한 리전 예약 인스턴스가 미사용 상태로 남게 되고 계정 B의 사용량은 온디맨드 요금으로 부과됩니다.

자세한 내용은 Billing and Cost Management 보고서의 예약 인스턴스 단원을 참조하십시오.

요금 부과 방법

모든 예약 인스턴스에서는 온디맨드 요금에 비해 할인된 요금을 제공합니다. 예약 인스턴스를 사용하면 실제 사용에 상관없이 전체 기간에 대해 요금을 지불합니다. 예약 인스턴스에 대해 지정한 결제 옵션 (p. 275)에 따라 예약 인스턴스에 대해 선결제, 부분 선결제 또는 월별 결제를 선택할 수 있습니다.

예약 인스턴스가 만료되면 EC2 인스턴스 사용량에 대해 온디맨드 요금이 청구됩니다. 최대 3년 전에 예약 인스턴스 구매를 대기할 수 있습니다. 이를 통해 중단 없는 보장을 받을 수 있습니다. 자세한 내용은 구매 대기 (p. 287) 단원을 참조하십시오.

AWS 프리 티어는 신규 AWS 계정에 제공됩니다. AWS 프리 티어를 사용하여 Amazon EC2 인스턴스를 실행하며 예약 인스턴스를 구입하는 경우 표준 요금 정책에 따라 요금이 부과됩니다. 자세한 내용은 AWS 프리 티어를 참조하십시오.

내용

- 사용량 결제 (p. 283)
- 청구서 보기 (p. 284)
- 예약 인스턴스 및 통합 결제 (p. 284)
- 예약 인스턴스 할인 요금 티어 (p. 284)

사용량 결제

예약 인스턴스는 선택한 기간 동안 인스턴스 실행 여부와 상관없이 매 시간 청구됩니다. 각 시간은 표준 24시간 시계의 정각(0분 0초)에 시작합니다. 예를 들어 1:00:00부터 1:59:59까지가 1시간입니다. 인스턴스 상태에 대한 자세한 내용은 인스턴스 수명 주기 (p. 426) 단원을 참조하십시오.

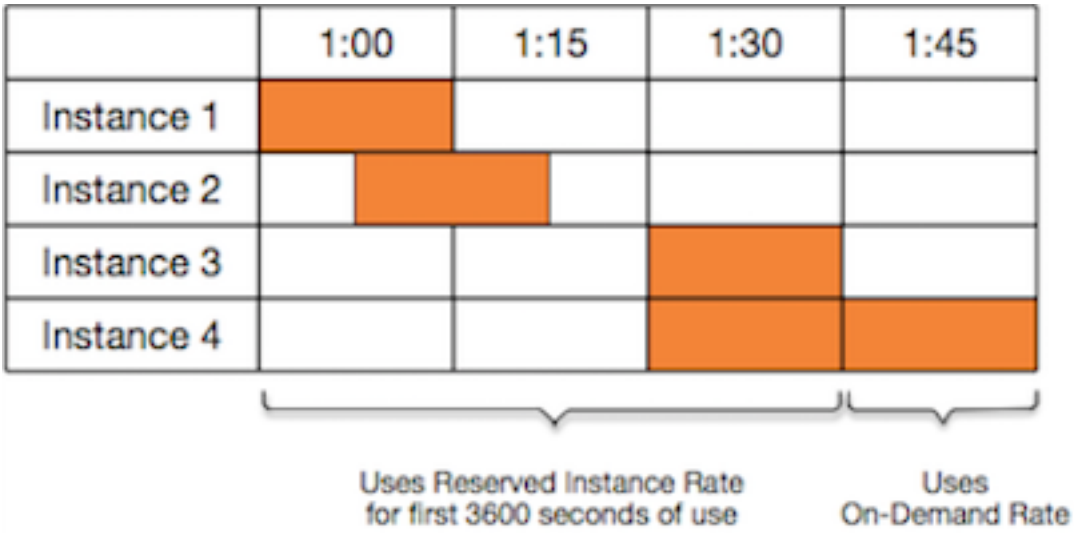
예약 인스턴스 결제 혜택이 초 단위로 실행 중 인스턴스에 적용됩니다. 초당 요금은 오픈 소스 Linux 배포판(예: Amazon Linux 및 Ubuntu)을 사용하는 경우 사용할 수 있습니다. 시간당 청구는 Red Hat Enterprise Linux 및 SUSE Linux Enterprise Server와 같은 상용 Linux 배포판에 사용됩니다.

예약 인스턴스 결제 혜택은 매 시간마다 최대 3,600초(1시간)의 인스턴스 사용량에 적용될 수 있습니다. 여러 인스턴스를 동시에 실행할 수 있지만, 예약 인스턴스 할인 혜택은 매 시간마다 총 3,600초에 대해서만 받을 수 있고 매 시간 3,600초를 초과하는 인스턴스 사용량에는 온디맨드 요금이 청구됩니다.

예를 들어 `m4.xlarge` 예약 인스턴스 1개를 구매하고 `m4.xlarge` 인스턴스 4개를 동시에 1시간 동안 실행한 경우 인스턴스 1개에 대해 예약 인스턴스 사용량 1시간의 요금이 부과되고 나머지 인스턴스 3개에 대해 온디맨드 사용량 3시간의 요금이 부과됩니다.

하지만 `m4.xlarge` 예약 인스턴스 1개를 구매하고 `m4.xlarge` 인스턴스 4개를 동일한 시간 내에 15분씩 실행하여 총 인스턴스 실행 시간이 1시간인 경우 예약 인스턴스 사용량은 1시간이 되고 온디맨드 사용량은 0시간이 됩니다.

여러 개의 적용 대상 인스턴스가 동시에 실행 중인 경우 예약 인스턴스 결제 혜택이 모든 인스턴스에 대해 동시에 매 시간마다 최대 3,600초까지 적용되고, 그 이후에는 온디맨드 요금이 적용됩니다.



Billing and Cost Management 콘솔의 Cost Explorer를 사용하면 실행 중인 온디맨드 인스턴스에 대해 절약된 금액을 분석할 수 있습니다. 예약 인스턴스 FAQ에는 정가 계산의 예가 나와 있습니다.

AWS 계정을 닫은 경우 리소스에 대한 온디맨드 결제가 중지됩니다. 그러나 계정에 예약 인스턴스가 있는 경우 해당 인스턴스가 만료될 때까지 이에 대한 청구서를 계속 받게 됩니다.

청구서 보기

계정으로 청구되는 요금과 비용은 AWS Billing and Cost Management 콘솔에서 확인할 수 있습니다.

- 대시보드에는 계정에 대한 소비 요약이 표시됩니다.
- 청구서 페이지의 세부 정보에서 Elastic Compute Cloud 섹션과 리전을 확장하여 예약 인스턴스에 대한 결제 정보를 가져옵니다.

요금을 온라인으로 확인하거나 CSV 파일을 다운로드할 수 있습니다.

AWS 비용 및 사용 보고서를 통해 예약 인스턴스 사용률을 추적할 수도 있습니다. 자세한 내용은 AWS Billing and Cost Management 사용 설명서의 비용 및 사용 보고서에서 예약 인스턴스를 참조하십시오.

예약 인스턴스 및 통합 결제

구입 계정이 단일 통합 결제 지급인 계정으로 과금되는 일련의 계정 중 하나인 경우, 예약 인스턴스 요금 혜택이 공유됩니다. 모든 멤버 계정에서 발생한 인스턴스 사용량은 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 Organizations 사용 설명서의 통합 결제 및 AWS Organizations 단원을 참조하십시오.

예약 인스턴스를 구매한 계정을 닫으면 예약 인스턴스가 만료되거나 닫힌 계정이 영구적으로 삭제될 때까지 지급인 계정에 예약 인스턴스 요금이 계속 청구됩니다. 닫힌 계정은 90일 후에 영구적으로 삭제됩니다. 삭제 후에는, 해당 멤버 계정에 대한 예약 인스턴스 결제 할인의 혜택이 중지됩니다. 계정 닫기에 대한 자세한 내용은 AWS Organizations 사용 설명서의 AWS 계정 닫기를 참조하십시오.

예약 인스턴스 할인 요금 티어

할인 요금 티어의 사용 자격에 해당되는 계정은 적용 시점부터 구입한 예약 인스턴스 중 해당 티어에 속하는 예약 인스턴스의 선결제 금액과 인스턴스 사용비가 자동으로 할인됩니다. 할인은 해당 리전 내 예약 인스턴스의 정가 총합이 500,000 USD 이상인 경우만 해당됩니다.

다음 규칙이 적용됩니다.

- 요금 티어 및 이와 관련된 할인은 Amazon EC2 스탠다드 예약 인스턴스 구입 시에만 적용됩니다.
- SQL Server Standard, SQL Server Web 및 SQL Server Enterprise 포함 Windows에는 예약 인스턴스 요금 티어가 적용되지 않습니다.
- SQL Server Standard, SQL Server Web 및 SQL Server Enterprise 포함 Linux에는 예약 인스턴스 요금 티어가 적용되지 않습니다.
- 요금 티어의 할인 혜택은 AWS를 통한 구매에만 적용됩니다. 타사 예약 인스턴스를 구입할 때는 이 할인 혜택이 적용되지 않습니다.
- 할인 요금 티어는 현재 전환형 예약 인스턴스 구입에는 적용되지 않습니다.

주제

- 예약 인스턴스 요금 할인 계산 (p. 285)
- 구매 시 할인 티어 적용 (p. 285)
- 요금 티어 교차 (p. 286)
- 요금 티어 통합 결제 (p. 286)

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 약정 기간의 총 시간을 곱한 다음, 구입 시 지불한 예약 인스턴스 요금 페이지 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 구입 후 가격이 내려가는 경우 정가에는 영향을 주지 않습니다.

```
List value = fixed price + (undiscounted recurring hourly price * hours in term)
```

예를 들어 1년 부분 선결제 `t2.small` 예약 인스턴스의 경우 선결제 가격이 60.00 USD이고 시간당 요금이 0.007 USD라고 가정해 봅니다. 이렇게 하면 정가는 121.32 USD입니다.

```
121.32 = 60.00 + (0.007 * 8760)
```

Amazon EC2 콘솔을 사용하여 예약 인스턴스의 고정 가격을 확인하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 예약 인스턴스를 선택합니다.
3. 화면 상단 오른쪽의 열 표시/숨기기(기어 모양 아이콘)를 선택하여 선결제 가격 열을 표시합니다.

명령줄을 사용하여 예약 인스턴스의 고정 가격을 확인하려면

- describe-reserved-instances(AWS CLI)
- Get-EC2ReservedInstance(Windows PowerShell용 AWS 도구)
- DescribeReservedInstances(Amazon EC2 API)

구매 시 할인 티어 적용

사용자가 예약 인스턴스를 구입하면 Amazon EC2에서는 할인 요금 티어에 해당되는 구입에 대해 자동으로 그에 맞는 할인을 적용합니다. 추가 작업 없이 어떤 Amazon EC2 도구에서나 예약 인스턴스 구매가 가능합니다. 자세한 내용은 예약 인스턴스 구입 (p. 286) 단원을 참조하십시오.

한 리전에서 사용 중인 예약 인스턴스의 정가 총액이 할인 요금 티어 기준에 도달하면 다음에 같은 리전에서 예약 인스턴스를 구입할 때 할인이 적용됩니다. 어떤 리전에서 예약 인스턴스를 하나 구입했는데 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. 286) 단원을 참조하십시오.

정가 총액이 예약 인스턴스 만료 등의 이유로 이용 중이던 할인 요금 티어 기준 이하로 변경되면 그 다음에 해당 리전에서 예약 인스턴스를 구입할 때는 할인이 적용되지 않습니다. 단, 구입 시 할인 요금 티어 범위에 해당되었던 기존의 예약 인스턴스에 대해서는 계속 할인을 받을 수 있습니다.

예약 인스턴스 구입 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 기존보다 더 높아진 경우입니다. 이 경우 두 가지 요금이 차등 적용됩니다. 합산 가격을 기준으로 하나 또는 그 이상의 예약 인스턴스에는 기존 티어 할인이, 나머지 인스턴스에는 상위 티어 할인이 적용됩니다.

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 미할인 티어(정상 가격), 하나 이상의 할인 티어가 차등 적용되므로, 예약 인스턴스 서비스에서는 여러 개의 예약 인스턴스 ID를 생성합니다. ID는 같은 티어의 인스턴스를 묶어 티어당 하나씩 부여됩니다. 따라서 CLI 명령이나 API 작업으로 구입했을 때 부여되는 ID는 새로 구입한 예약 인스턴스의 실제 ID와는 다릅니다.

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 구입한 예약 인스턴스에 대해 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. 284) 단원을 참조하십시오.

예약 인스턴스 구입

예약 인스턴스를 구입하려면 AWS 및 타사 판매업자로부터 예약 인스턴스 상품을 검색하고, 찾고 있는 인스턴스와 정확히 일치하는 인스턴스를 찾을 때까지 검색 파라미터를 조정합니다.

구입할 예약 인스턴스를 검색하면 반환된 상품의 비용에 대한 견적을 받게 됩니다. 구입을 진행하면 AWS에서 구입 가격에 제한 가격을 자동으로 설정합니다. 그러면 구입하는 예약 인스턴스의 총 가격이 제시된 견적가를 초과하지 않게 됩니다.

여하한 이유로 가격이 오르거나 변경되면 구입이 완료되지 않습니다. 구매 당시 선택한 조건과 비슷한데 가격은 더 낮은 상품이 있을 경우 AWS는 더 저렴한 상품을 판매합니다.

구입을 확정하기 전에 구매하기로 결정한 예약 인스턴스의 세부 정보를 검토하고 모든 파라미터가 정확한지 확인하십시오. 예약 인스턴스는 한 번 구매하고 나면 구매를 취소할 수 없습니다(예약 인스턴스 마켓플레이스의 타사에서 구매한 경우와 AWS에서 구매한 경우 모두 동일).

> **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 정책을 참조하십시오.

작업

- 플랫폼 선택 (p. 286)
- 구매 대기 (p. 287)
- 표준 예약 인스턴스 구입 (p. 287)
- 전환형 예약 인스턴스 구입 (p. 289)
- 예약 인스턴스 보기 (p. 290)
- 대기 중인 구매 취소 (p. 290)
- 예약 인스턴스 갱신 (p. 291)
- 예약 인스턴스 사용 (p. 291)

플랫폼 선택

Amazon EC2는 예약 인스턴스에 다음과 같은 Linux 플랫폼을 지원합니다.

- Linux/UNIX
- SQL Server Standard가 설치된 Linux
- SQL Server Web이 설치된 Linux

- SQL Server Enterprise가 설치된 Linux
- SUSE Linux
- Red Hat Enterprise Linux

예약 인스턴스를 구입할 경우 해당 인스턴스의 운영 체제를 나타내는 플랫폼용 서비스를 선택해야 합니다.

- SUSE Linux 및 RHEL 배포의 경우, 해당 플랫폼(예: SUSE Linux 또는 Red Hat Enterprise Linux 플랫폼)용 서비스를 선택해야 합니다.
- 그 외 모든 Linux 배포에 대해서는(Ubuntu 포함) Linux/UNIX 플랫폼용 서비스를 선택합니다.
- 기존 RHEL 구독을 가져오는 경우 Red Hat Enterprise Linux 플랫폼용 서비스가 아닌 Linux/UNIX 플랫폼용 서비스를 선택해야 합니다.

Windows에 지원되는 플랫폼에 대한 자세한 내용은 Windows 인스턴스용 Amazon EC2 사용 설명서의 플랫폼 선택을 참조하십시오.

결제 제품 코드로 AMI에서 시작된 온디맨드 인스턴스에 적용하기 위해 예약 인스턴스를 구매하는 경우 일치하는 청구 제품 코드가 예약 인스턴스에 있는지 확인합니다. 일치하는 청구 제품 코드 없이 예약 인스턴스를 구매하는 경우 예약 인스턴스는 온디맨드 인스턴스에 적용되지 않습니다. AMI 결제 코드를 얻는 방법에 대한 자세한 내용은 결제 정보 가져오기 (p. 155) 단원을 참조하십시오.

구매 대기

기본적으로 예약 인스턴스를 구매하면 즉시 실행됩니다. 또는 향후 날짜와 시간으로 구매를 대기시킬 수 있습니다. 예를 들어, 기존 예약 인스턴스 만료 즈음에 구매를 대기시킬 수 있습니다. 이를 통해 중단 없는 보장을 받을 수 있습니다.

다른 판매자로부터의 지역 예약 인스턴스(영역 예약 인스턴스나 예약 인스턴스 아님) 구매를 대기할 수 있습니다. 최대 3년 전에 구매를 대기할 수 있습니다. 예약된 날짜 및 시간에 기본 결제 방법을 사용하여 구매가 실행됩니다. 결제가 완료되면 결제 혜택이 적용됩니다.

대기 중인 구매 내역을 Amazon EC2 콘솔에서 볼 수 있습니다. 대기 중인 구매의 상태는 대기 중입니다. 예약된 시간 전에 언제든지 대기 중인 구매를 취소할 수 있습니다. 세부 정보는 대기 중인 구매 취소 (p. 290) 단원을 참조하십시오.

표준 예약 인스턴스 구입

특정 가용 영역에서 표준 예약 인스턴스를 구입하고 용량을 예약할 수 있습니다. 또는 용량 예약을 포기하고 리전 단위의 표준 예약 인스턴스를 구입할 수 있습니다;.

콘솔을 사용하여 표준 예약 인스턴스를 구매하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 예약 인스턴스를 선택한 다음 예약 인스턴스 구입을 선택합니다.
3. 제공 클래스에서 표준을 선택하여 표준 예약 인스턴스를 표시합니다.
4. 용량 예약을 구입하려면 구입 화면의 상단 오른쪽 모서리 부분에서 용량이 예약된 제공만 표시를 선택합니다. 리전 단위의 예약 인스턴스를 구입하려면 상자를 선택하지 않은 채로 둡니다.
5. 필요에 따라 다른 구성을 선택하고 검색을 선택합니다.

   예약 인스턴스 마켓플레이스에서 표준 예약 인스턴스를 구입하려면 검색 결과의 판매자 열에서 타사를 찾습니다. 기간 열에 비 표준 약정이 표시됩니다.
6. 구입할 예약 인스턴스를 선택하고 수량을 입력한 후 장바구니에 추가를 선택합니다.
7. 선택한 예약 인스턴스의 요약을 보려면 장바구니 보기를 선택합니다.
8. Order On(주문 시각)이 Now(지금)이면 즉시 구매가 완료됩니다. 구매를 대기시키려면 지금을 선택하고 날짜를 선택하십시오. 장바구니에서 적합한 각 상품에 대해 다른 날짜를 선택할 수 있습니다. 구매는 브라우저 시간대에서 선택한 날짜에 00:00까지 대기합니다.

9. 주문을 완료하려면 주문을 선택합니다.

   주문 당시 선택한 조건과 비슷하지만 가격이 더 낮은 상품이 있는 경우 AWS는 더 저렴한 상품을 판매합니다.

10. 주문 상태가 상태 열에 나열됩니다. 주문이 완료되면 상태 값이 `payment-pending`에서 `active`로 바뀝니다. 예약 인스턴스가 `active`인 경우 사용할 준비가 됩니다.

    Note

    상태가 `retired`로 바뀌면 AWS에서 결제를 받을 수 없습니다.

AWS CLI를 사용하여 표준 예약 인스턴스를 구매하려면

1. describe-reserved-instances-offerings 명령을 사용하여 사용 가능한 예약 인스턴스를 찾습니다. `--offering-class` 파라미터에 대해 `standard`를 지정하여 표준 예약 인스턴스만 반환합니다. 예를 들어 1년 동안만 `Linux/UNIX`에 대해 기본 테넌시가 포함된 리전 단위의 `t2.large` 예약 인스턴스를 구입하려 할 경우와 같이 결과를 좁히기 위해 추가 파라미터를 적용할 수 있습니다.

```
aws ec2 describe-reserved-instances-offerings --instance-type t2.large --offering-class standard --product-description "Linux/UNIX" --instance-tenancy default --filters Name=duration,Values=31536000 Name=scope,Values=Region
```

   예약 인스턴스 마켓플레이스에서만 예약 인스턴스를 찾으려면 `marketplace` 필터를 사용하고 기간이 1–년 또는 3년 기간보다 짧을 수 있으므로 요청에 기간을 지정하지 않습니다.

```
aws ec2 describe-reserved-instances-offerings --instance-type t2.large --offering-class standard --product-description "Linux/UNIX" --instance-tenancy default --filters Name=marketplace,Values=true
```

   요구 사항에 맞는 예약 인스턴스를 찾은 경우 상품 ID를 기록합니다. 예:

```
"ReservedInstancesOfferingId": "bec624df-a8cc-4aad-a72f-4f8abc34caf2"
```

2. purchase-reserved-instances-offering 명령을 사용하여 예약 인스턴스를 구매합니다. 이전 단계에서 얻은 예약 인스턴스 상품 ID를 지정하고 예약을 위한 인스턴스 수를 지정해야 합니다.

```
aws ec2 purchase-reserved-instances-offering --reserved-instances-offering-id bec624df-a8cc-4aad-a72f-4f8abc34caf2 --instance-count 1
```

   기본적으로 구매는 즉시 완료됩니다. 또는 구매를 대기시키려면 다음 매개 변수를 이전 호출에 추가하십시오.

```
--purchase-time "2020-12-01T00:00:00Z"
```

3. describe-reserved-instances 명령을 사용하여 예약 인스턴스의 상태를 가져옵니다.

```
aws ec2 describe-reserved-instances
```

또는 다음 Windows PowerShell용 AWS 도구 명령을 사용합니다.

- Get-EC2ReservedInstancesOffering
- New-EC2ReservedInstance
- Get-EC2ReservedInstance

구매를 완료했으며 예약 인스턴스의 사양과 일치하는 인스턴스가 이미 실행 중인 경우 결제 혜택이 즉시 적용됩니다. 인스턴스를 따로 재시작할 필요가 없습니다. 실행 중인 적합 인스턴스가 없는 경우, 인스턴스를 시작하고 예약 인스턴스에 대해 지정한 동일한 조건과 일치하는지 확인합니다. 자세한 내용은 예약 인스턴스 사용 (p. 291) 단원을 참조하십시오.

예를 들어 실행 중인 인스턴스에 예약 인스턴스를 적용하는 방법은 예약 인스턴스 적용 방식 (p. 277) 단원을 참조하십시오.

전환형 예약 인스턴스 구입

특정 가용 영역에서 전환형 예약 인스턴스를 구입하고 용량을 예약할 수 있습니다. 또는 용량 예약을 포기하고 리전 단위의 전환형 예약 인스턴스를 구입할 수 있습니다;.

콘솔을 사용하여 전환형 예약 인스턴스를 구입하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 예약 인스턴스를 선택한 다음 예약 인스턴스 구입을 선택합니다.
3. 제공 클래스에 대해 컨버터블을 선택하여 전환형 예약 인스턴스를 표시합니다.
4. 용량 예약을 구입하려면 구입 화면의 상단 오른쪽 모서리 부분에서 용량이 예약된 제공만 표시를 선택합니다. 리전 단위의 예약 인스턴스를 구입하려면 상자를 선택하지 않은 채로 둡니다.
5. 필요에 따라 다른 구성을 선택하고 검색을 선택합니다.
6. 구입할 전환형 예약 인스턴스를 선택하고 수량을 입력한 후 장바구니에 추가를 선택합니다.
7. 선택한 내역을 보려면 장바구니 보기를 선택합니다.
8. Order On(주문 시각)이 Now(지금)이면 즉시 구매가 완료됩니다. 구매를 대기시키려면 지금을 선택하고 날짜를 선택하십시오. 장바구니에서 적합한 각 상품에 대해 다른 날짜를 선택할 수 있습니다. 구매는 브라우저 시간대에서 선택한 날짜에 00:00까지 대기합니다.
9. 주문을 완료하려면 주문을 선택합니다.

   주문 당시 선택한 조건과 비슷하지만 가격이 더 낮은 상품이 있는 경우 AWS는 더 저렴한 상품을 판매합니다.
10. 주문 상태가 상태 열에 나열됩니다. 주문이 완료되면 상태 값이 `payment-pending`에서 `active`로 바뀝니다. 예약 인스턴스가 `active`인 경우 사용할 준비가 됩니다.

    Note

    상태가 `retired`로 바뀌면 AWS에서 결제를 받을 수 없습니다.

AWS CLI를 사용하여 전환형 예약 인스턴스를 구입하려면

1. describe-reserved-instances-offerings 명령을 사용하여 사용 가능한 예약 인스턴스를 찾습니다. `--offering-class` 파라미터에 대해 `convertible`를 지정하여 전환형 예약 인스턴스만 반환합니다. 예를 들어 `Linux/UNIX`에 대해 기본 테넌시가 포함된 리전 단위의 `t2.large` 예약 인스턴스를 구입하려 할 경우와 같이 결과를 좁히기 위해 추가 파라미터를 적용할 수 있습니다.

   ```
   aws ec2 describe-reserved-instances-offerings --instance-type t2.large --offering-class convertible --product-description "Linux/UNIX" --instance-tenancy default --filters Name=scope,Values=Region
   ```

   요구 사항에 맞는 예약 인스턴스를 찾은 경우 상품 ID를 기록합니다. 예:

   ```
   "ReservedInstancesOfferingId": "bec624df-a8cc-4aad-a72f-4f8abc34caf2"
   ```
2. purchase-reserved-instances-offering 명령을 사용하여 예약 인스턴스를 구매합니다. 이전 단계에서 얻은 예약 인스턴스 상품 ID를 지정하고 예약을 위한 인스턴스 수를 지정해야 합니다.

```
aws ec2 purchase-reserved-instances-offering --reserved-instances-offering-id bec624df-a8cc-4aad-a72f-4f8abc34caf2 --instance-count 1
```

기본적으로 구매는 즉시 완료됩니다. 또는 구매를 대기시키려면 다음 매개 변수를 이전 호출에 추가하십시오.

```
--purchase-time "2020-12-01T00:00:00Z"
```

3. describe-reserved-instances 명령을 사용하여 예약 인스턴스의 상태를 가져옵니다.

```
aws ec2 describe-reserved-instances
```

또는 다음 Windows PowerShell용 AWS 도구 명령을 사용합니다.

- Get-EC2ReservedInstancesOffering
- New-EC2ReservedInstance
- Get-EC2ReservedInstance

예약 인스턴스의 사양과 일치하는 인스턴스가 이미 실행 중인 경우 결제 혜택이 즉시 적용됩니다. 인스턴스를 따로 재시작할 필요가 없습니다. 실행 중인 적합 인스턴스가 없는 경우, 인스턴스를 시작하고 예약 인스턴스에 대해 지정한 동일한 조건과 일치하는지 확인합니다. 자세한 내용은 예약 인스턴스 사용 (p. 291) 단원을 참조하십시오.

예를 들어 실행 중인 인스턴스에 예약 인스턴스를 적용하는 방법은 예약 인스턴스 적용 방식 (p. 277) 단원을 참조하십시오.

예약 인스턴스 보기

Amazon EC2 콘솔 또는 명령줄 도구를 사용하여 구입한 예약 인스턴스를 볼 수 있습니다.

콘솔에서 예약 인스턴스를 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 예약 인스턴스를 선택합니다.
3. 활성 및 수명종료 예약 인스턴스가 나열됩니다. 상태 열에 상태가 표시됩니다.
4. 예약 인스턴스 마켓플레이스의 판매자인 경우 예약 인스턴스 마켓플레이스 (p. 291)에 나열된 예약 상태가 내 항목 탭에 표시됩니다. 자세한 내용은 예약 인스턴스 항목 상태 (p. 295) 단원을 참조하십시오.

명령줄을 사용하여 예약 인스턴스를 보려면

- describe-reserved-instances(AWS CLI)
- Get-EC2ReservedInstance(Windows PowerShell용 도구)

대기 중인 구매 취소

최대 3년 전에 구매를 대기할 수 있습니다. 예약된 시간 전에 언제든지 대기 중인 구매를 취소할 수 있습니다.

대기 중인 구매 취소

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

3. 하나 이상의 예약 인스턴스를 선택합니다.
4. 조치, 대기 중인 예약 인스턴스 삭제를 선택하십시오.
5. 확인 메시지가 나타나면 예, 삭제합니다를 선택합니다.

명령줄을 사용하여 대기 중인 구매를 취소하려면

- delete-queued-reserved-instances(AWS CLI)
- Remove-EC2QueuedReservedInstance(Windows PowerShell용 도구)

예약 인스턴스 갱신

만료되기 전에 예약 인스턴스를 갱신할 수 있습니다. 예약 인스턴스 대기열을 갱신하면 현재 예약 인스턴스가 만료될 때까지 동일한 구성으로 예약 인스턴스를 구매할 수 있습니다.

대기 중인 구매를 사용하여 예약 인스턴스를 갱신하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 예약 인스턴스를 선택합니다.
3. 하나 이상의 예약 인스턴스를 선택합니다.
4. 작업, Renew Reserved Instances(예약 인스턴스 갱신)를 선택합니다.
5. 주문을 완료하려면 주문을 선택합니다.

예약 인스턴스 사용

사양이 일치할 경우 실행 중인 온디맨드 인스턴스에 예약 인스턴스가 자동으로 적용됩니다. 예약 인스턴스의 사양과 일치하는 온디맨드 인스턴스가 실행되고 있지 않은 경우 필수 사양이 포함된 인스턴스를 시작할 때까지 예약 인스턴스가 사용되지 않습니다.

예약 인스턴스의 결제 혜택을 활용하기 위해 인스턴스를 시작할 경우 시작 시 다음 정보를 지정해야 합니다.

- 플랫폼: 예약 인스턴스의 플랫폼(제품 설명)과 일치하는 Amazon 머신 이미지(AMI)를 선택해야 합니다;. 예를 들어, `Linux/UNIX`를 지정한 경우 Amazon Linux AMI 또는 Ubuntu AMI에서 인스턴스를 시작할 수 있습니다.
- 인스턴스 유형: `t2.large`와 같이 예약 인스턴스와 동일한 인스턴스 유형을 지정합니다.
- 가용 영역: 특정 가용 영역에 대해 예약 인스턴스를 구입한 경우 동일한 가용 영역으로 해당 인스턴스를 시작해야 합니다. 리전 단위의 예약 인스턴스를 구입한 경우 모든 가용 영역으로 인스턴스를 시작할 수 있습니다.
- 테넌시: 인스턴스의 테넌시는 예약 인스턴스의 테넌시와 일치해야 합니다(예: `dedicated` 또는 `shared`). 자세한 내용은 전용 인스턴스 (p. 410) 단원을 참조하십시오.

자세한 내용은 인스턴스 시작 마법사를 사용하여 인스턴스 시작 (p. 431) 단원을 참조하십시오. 예를 들어 실행 중인 인스턴스에 예약 인스턴스를 적용하는 방법은 예약 인스턴스 적용 방식 (p. 277) 단원을 참조하십시오.

Amazon EC2 Auto Scaling 또는 다른 AWS 서비스를 사용하여 예약 인스턴스의 혜택이 적용되는 온디맨드 인스턴스를 시작할 수 있습니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서를 참조하십시오.

예약 인스턴스 마켓플레이스

예약 인스턴스 마켓플레이스은 타사 및 AWS 고객의 미사용 표준 예약 인스턴스 판매를 지원하는 플랫폼으로, 사용 기간 및 요금 옵션이 다릅니다. 예를 들어 인스턴스를 새로운 AWS 리전으로 이동할 경우, 새 인스턴

스 유형으로 변경한 후, 약정이 만료되기 전에 프로젝트가 종료될 경우, 비즈니스에서 변경이 필요한 경우 또는 불필요한 용량이 있는 경우 예약 인스턴스를 판매할 수 있습니다.

예약 인스턴스 마켓플레이스에서 미사용 예약 인스턴스를 판매하려 할 경우 특정 자격 기준을 충족해야 합니다.

내용

- 예약 인스턴스 마켓플레이스에서 판매 (p. 292)
- 예약 인스턴스 Marketplace에서 구입 (p. 297)

예약 인스턴스 마켓플레이스에서 판매

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. 그런 다음 AWS는 이 거래를 처리하고 해당 예약 인스턴스의 소유권을 구매자에게 이전합니다.

예약 인스턴스가 판매되기 전까지는 판매자에게 소유권이 있습니다. 판매 후에는 용량 예약과 할인 기본 요금이 구매자에게 양도됩니다. 인스턴스를 계속 사용할 경우 AWS는 해당 예약 인스턴스가 판매된 시점부터 온디맨드 요금을 적용합니다.

목차

- 규제 및 제한 (p. 292)
- 판매자 등록 (p. 293)
- 지급금 은행 계좌 (p. 293)
- 세금 정보 (p. 294)
- 예약 인스턴스 요금 (p. 294)
- 예약 인스턴스 나열 (p. 295)
- 예약 인스턴스 항목 상태 (p. 295)
- 판매 기간 (p. 296)
- 예약 인스턴스 판매 후 절차 (p. 296)
- 판매 대금 정산 (p. 296)
- 구매자와의 정보 공유 (p. 297)

규제 및 제한

예약 인스턴스 마켓플레이스에서 사용하지 않은 예약을 판매하기 전에 판매자로 등록해야 합니다. 자세한 내용은 판매자 등록 (p. 293) 단원을 참조하십시오.

예약 인스턴스 판매 시 다음과 같은 제한 및 제약이 적용됩니다.

- 예약 인스턴스 마켓플레이스에서 Amazon EC2 표준 예약 인스턴스만 판매할 수 있습니다. Amazon EC2 전환형 예약 인스턴스는 판매할 수 없습니다. Amazon RDS 및 Amazon ElastiCache와 같은 기타 AWS의 예약 인스턴스는 판매할 수 없습니다.
- 표준 예약 인스턴스의 남은 사용 기간이 한 달 이상이어야 합니다.
- 기본적으로 비활성화된 리전(아시아 태평양(홍콩) 및 중동(바레인))에서는 표준 예약 인스턴스를 판매할 수 없습니다.
- 예약 인스턴스 마켓플레이스에서 허용되는 최소 허용 판매가는 0.00 USD입니다.
- 예약 인스턴스 마켓플레이스에서는 선결제 없음, 부분 선결제, 혹은 전체 선결제 예약 인스턴스를 판매할 수 있습니다. 예약 인스턴스에 선결제가 있으면 AWS에 선결제가 완료된 이후, 그리고 예약 활성 기간(소유 기간)이 30일 이상인 경우에만 판매할 수 있습니다.

- 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 인스턴스 요금 (p. 294) 단원을 참조하십시오. 판매 등록하기 전에 예약 인스턴스를 수정할 수도 있습니다. 자세한 내용은 예약 인스턴스 수정 (p. 297) 단원을 참조하십시오.
- AWS는 예약 인스턴스 마켓플레이스에서 판매하는 각 표준 예약 인스턴스에 대해 총 선결제 금액의 12%를 서비스 수수료로 청구합니다. 선결제 금액은 판매자가 판매 등록한 표준 예약 인스턴스에 책정한 가격입니다;.

판매자 등록

> **Note**
>
> AWS 계정 루트 사용자만 계정을 판매자로 등록할 수 있습니다.

예약 인스턴스 마켓플레이스에서 판매하려면 먼저 판매자로 등록해야 합니다. 등록 과정에서 다음 정보를 제공해야 합니다.

- 은행 정보 - AWS에서 예약 인스턴스를 판매할 경우 판매 대금을 지급하기 위해 사용자의 은행 정보가 필요합니다. 이때 미국 소재지가 있는 은행을 선택해야 합니다. 자세한 내용은 지급금 은행 계좌 (p. 293) 단원을 참조하십시오.
- 세금 정보 - 모든 판매자는 세금 신고 의무를 결정하기 위해서 세금 신고서를 작성해야 합니다. 자세한 내용은 세금 정보 (p. 294) 단원을 참조하십시오.

AWS에서 판매자 등록에 필요한 과정을 모두 마치면 등록 확인과 함께 예약 인스턴스 마켓플레이스에서 판매를 시작할 수 있음을 알리는 이메일이 발송됩니다.

지급금 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 페이지를 열고 AWS 자격 증명을 사용하여 로그인합니다.
2. 은행 계좌 관리(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 완료되는데 보통 2일 정도 걸립니다.

지급금을 받을 기본 은행 계좌를 변경하려면

1. 예약 인스턴스 마켓플레이스 판매자 등록 페이지에서 등록 시 사용한 계정으로 로그인합니다.

2. 은행 계좌 관리(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 사용자게게 있습니다.

판매자 등록 과정에서는 판매자 등록 포털에서 세금 신고서를 작성해야 합니다. 인터뷰어가 세금 정보를 받아서 세금 신고 의무를 결정하기 위한 IRS form W-9, W-8BEN, 혹은 W-8BEN-E를 추가합니다.

세금 신고서 작성 시 입력하는 세금 정보는 개인인지 아니면 기업인지 혹은 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 상담하십시오.
- 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 신고 및 IRS 신고 기준을 바탕으로 Amazon은 Form 1099-K를 생성합니다. Amazon은 세금 계정이 기준선을 초과한 년도의 다음 해 1월 31일 또는 그 이전에 Form 1099-K 복사본을 우편으로 보냅니다. 예를 들어 세금 계정이 2018년에 한계에 도달하면 Form 1099-K는 2019년 1월 31일 또는 그 이전에 우편으로 보내집니다.

IRS 세금 신고 규정과 Form 1099-K에 대한 자세한 내용은 IRS 웹 사이트 단원을 참조하십시오.

예약 인스턴스 요금

판매자는 판매할 예약 인스턴스에 대한 선결제 금액만 책정할 수 있습니다. 선결제 금액은 구매자가 예약 인스턴스를 구매할 때 지불하는 일회성 요금입니다;.

다음은 알아 두어야 할 중요한 제한 사항입니다.

- 연간 최대 50,000 USD의 예약 인스턴스를 판매할 수 있습니다. 그 이상 판매하려면 Amazon EC2 예약 인스턴스 판매 한도 상향 요청 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.

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 받을 금액을 설정할 수 있습니다. 남은 기간이 5개월인 시점과 1개월인 시점에서의 판매 가격을 각각 책정할 수 있습니다.

예약 인스턴스 나열

등록된 판매자는 예약 인스턴스를 한 개 이상 판매하기로 선택할 수 있습니다. 한 번의 판매 등록으로 모두 판매하거나 부분적으로 판매하기로 선택할 수 있습니다. 뿐만 아니라 인스턴스 유형, 플랫폼 및 범위의 구성으로 예약 인스턴스를 판매 등록할 수 있습니다.

콘솔에서 제안 가격이 결정됩니다. 콘솔은 예약 인스턴스와 일치하는 제품을 점검하고 해당 제품을 최저 가격과 일치시킵니다. 그렇지 않으면 남은 시간 동안 예약 인스턴스 비용을 기반으로 제안 가격을 계산합니다. 계산된 값이 $1.01보다 낮은 경우 제안 가격은 $1.01입니다.

판매 등록을 취소할 때 인스턴스 중 일부가 이미 판매되었다면, 이미 판매된 인스턴스에 대해서는 취소가 적용되지 않습니다. 아직 판매되지 않은 인스턴스만 예약 인스턴스 마켓플레이스 판매 목록에서 삭제됩니다.

AWS Management 콘솔을 사용하여 예약 인스턴스 마켓플레이스에서 예약 인스턴스를 판매 등록하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 예약 인스턴스를 선택합니다.
3. 판매 등록할 예약 인스턴스를 선택하고 예약 인스턴스 판매를 선택합니다.
4. 예약 예약 인스턴스 구성 페이지에서 판매할 인스턴스의 수와, 남은 사용 기간에 대한 선결제 금액을 해당 열에 설정합니다. 남은 개월 수 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 재설정을 선택합니다.
6. 판매 등록 구성을 마쳤으면 계속을 선택합니다.
7. 예약 예약 인스턴스 확인 페이지에 표시된 세부 정보를 확인하고 그대로 진행하려면 예약 인스턴스 리스팅를 선택합니다.

콘솔에서 등록 상품을 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 예약 인스턴스를 선택합니다.
3. 판매 등록한 예약 인스턴스를 선택하고 내 항목을 선택합니다.

AWS CLI를 사용하여 예약 인스턴스 마켓플레이스에서 예약 인스턴스를 관리하려면

1. describe-reserved-instances 명령을 사용하여 예약 인스턴스 목록을 가져옵니다.
2. 판매 등록할 예약 인스턴스의 ID를 기록하고 create-reserved-instances-listing을 호출합니다. 예약 인스턴스의 ID, 인스턴스의 수 및 가격표를 지정해야 합니다.
3. 판매 등록을 보려면 describe-reserved-instances-listings 명령을 사용합니다.
4. 판매 등록을 취소하려면 cancel-reserved-instances-listings 명령을 사용합니다.

예약 인스턴스 항목 상태

다음과 같이 예약 인스턴스 페이지의 내 항목 탭에 있는 항목 상태에 판매 등록의 현재 상태가 표시됩니다.

항목 상태에서 표시되는 정보는 예약 인스턴스 마켓플레이스에 판매 등록된 항목의 상태입니다. 이 상태 정보는 예약 인스턴스 페이지의 상태 열에 표시되는 상태 정보와는 다릅니다. 이 상태 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.

판매 기간

등록된 항목의 모든 인스턴스가 판매 완료된 경우, 내 항목 탭의 전체 인스턴스 수(Total instance count)의 값이 품절 항목의 값과 동일합니다. 또한 사용 가능 인스턴스가 더 이상 존재하지 않는 것을 확인할 수 있습니다. 상태 항목은 `closed`로 표시됩니다.

항목 중 일부 인스턴스만 판매된 경우 AWS에서 이 등록 항목에서 판매된 예약 인스턴스를 빼고 판매되지 않은 예약 인스턴스와 동일한 개수의 새 예약 인스턴스를 생성합니다. 따라서 판매 등록 ID와 해당 판매 등록은 활성 상태로 유지되지만, 남은 예약 인스턴스 수는 줄어듭니다.

이후 이 등록 항목에서 예약 인스턴스가 판매될 때마다 이같은 절차가 반복됩니다. 등록 항목에 있는 모든 예약 인스턴스가 판매되면 AWS에서 해당 등록 항목을 `closed`로 표시합니다.

예를 들어 예약 인스턴스 listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample 항목으로 5개의 인스턴스를 판매 등록했다고 가정해 보겠습니다.

이때 콘솔의 예약 인스턴스 페이지를 열었을 때 내 항목 탭에 다음 정보가 표시됩니다.

예약 인스턴스 listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample

- 전체 예약 인스턴스 개수 = 5
- Sold = 0
- Available = 5
- Status = active

구매자가 예약 중 2개를 구입한 경우 이제 판매 가능한 예약의 수는 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 내 항목 탭에 다음과 같이 나타납니다.

예약 인스턴스 listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample

- 전체 예약 인스턴스 개수 = 5
- Sold = 2
- Available = 3
- Status = active

판매 등록을 취소할 때 인스턴스 중 일부가 이미 판매되었다면, 이미 판매된 인스턴스에 대해서는 취소가 적용되지 않습니다. 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 있습니다.

콘솔에서 판매 등록된 예약 인스턴스 목록의 상태를 조회하려면 예약 인스턴스, 내 항목을 선택합니다. 내 항목 탭에는 항목 상태 값이 표시됩니다. 또한 사용 기간, 판매 가격, 등록 항목에서 Available(판매 가능), Pending(보류), Sold(판매), Cancelled(취소) 상태의 인스턴스 개수 정보도 제공됩니다. 또한 describe-reserved-instances-listings 명령을 통해 필터를 사용하여 판매 등록에 대한 정보를 알아볼 수도 있습니다.

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. 지불은 매일 한 번 실시됩니다. 대금이 지급된 후 지급금 보고서가

포함된 이메일이 전송됩니다. AWS에서 은행으로부터 계좌를 확인받기 전에는 대금이 지불되지 않으므로 이 점에 유의하십시오. 이 절차는 최대 2주가 소요됩니다.

사용자가 판매한 예약 인스턴스는 사용자가 예약 인스턴스를 설명할 때 계속 표시됩니다.

예약 인스턴스를 판매한 대금은 현금으로 지급되며 판매자 명의의 은행 계좌로 직접 송금됩니다. AWS는 예약 인스턴스 마켓플레이스에서 판매하는 각 표준 예약 인스턴스에 대해 총 선결제 금액의 12%를 서비스 수수료로 청구합니다.

구매자와의 정보 공유

예약 인스턴스 마켓플레이스에서 판매할 경우 AWS는 미국 규정에 따라 구매자 명세서에 판매자의 상호명을 기재하여 제공합니다. 또한 구매자가 인보이스 또는 다른 세금 관련 이유에 대해 문의하기 위해 AWS Support에 요청한 경우, AWS에서 구매자가 직접 연락을 취할 수 있도록 판매자의 이메일 주소를 제공해야 할 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 Support에 문의하십시오.

예약 인스턴스 Marketplace에서 구입

예약 인스턴스 마켓플레이스에서 더 이상 필요하지 않은 예약 인스턴스를 소유한 타사 판매업자로부터 예약 인스턴스를 구입할 수 있습니다. Amazon EC2 콘솔 또는 명령줄 도구를 사용하여 이 작업을 수행할 수 있습니다. 이 프로세스는 AWS에서 예약 인스턴스를 구입하는 것과 비슷합니다. 자세한 내용은 예약 인스턴스 구입 (p. 286) 단원을 참조하십시오.

예약 인스턴스 마켓플레이스에서 구입하는 예약 인스턴스와 AWS에서 직접 구입하는 예약 인스턴스 간에는 다음과 같은 몇 가지 차이점이 있습니다.

- 기간 - 타사에서 구입하는 예약 인스턴스는 남은 기간이 표준 약정 기간보다 짧습니다. AWS의 표준 약정 기간은 1년 또는 3년입니다.
- 선결제 요금 - 타사 예약 인스턴스는 다양한 선결제 요금으로 판매될 수 있습니다. 사용 요금이나 기본 요금은 AWS에서 예약 인스턴스를 처음에 구입했을 때 설정된 요금과 동일하게 유지됩니다.
- 예약 인스턴스의 유형 - 예약 인스턴스 마켓플레이스에서 Amazon EC2 표준 예약 인스턴스만 구입할 수 있습니다. 전환형 예약 인스턴스, Amazon RDS 및 Amazon ElastiCache 예약 인스턴스는 예약 인스턴스 마켓플레이스에서 구입할 수 없습니다.

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 Support로 문의하십시오.

예약 인스턴스 수정

변화가 생긴 경우 표준 또는 전환형 예약 인스턴스를 변경함으로써 요금 혜택에 따른 이점을 계속 유지할 수 있습니다. 가용 영역, 인스턴스 크기(동일한 인스턴스 패밀리 내) 및 예약 인스턴스의 범위와 같은 속성을 수정할 수 있습니다.

> **Note**
>
> 전환형 예약 인스턴스를 구성이 다른 전환형 예약 인스턴스와 교환할 수도 있습니다. 자세한 내용은 전환형 예약 인스턴스 교환 (p. 304) 단원을 참조하십시오.

예약 인스턴스의 전부 또는 하위 집합을 수정할 수 있습니다. 원래 예약 인스턴스를 둘 이상의 새 예약 인스턴스로 분리할 수 있습니다. 예를 들어, `us-east-1a`에서 10개의 예약을 보유하고 있으며 5개의 인스턴스를 `us-east-1b`로 옮기는 경우, 수정 요청에 따라 `us-east-1a`의 인스턴스 5개에 대한 예약 하나와 `us-east-1b`의 인스턴스 5개에 대한 다른 예약 하나 등 새로운 예약 두 개가 생성됩니다.

둘 이상의 예약 인스턴스를 단일 예약 인스턴스로 병합할 수도 있습니다. 예를 들어 인스턴스 하나에 대해 각각 `t2.small` 예약 인스턴스 4개가 있는 경우 이를 병합하여 `t2.large` 예약 인스턴스 하나를 생성할 수 있습니다. 자세한 내용은 인스턴스 크기 수정을 위한 지원 (p. 299) 단원을 참조하십시오.

수정 후 예약 인스턴스의 혜택은 새로운 파라미터와 일치하는 인스턴스에만 적용됩니다. 예를 들어, 예약의 가용 영역을 변경할 경우 용량 예약 및 요금 혜택이 새로운 가용 영역의 인스턴스 사용에 자동으로 적용됩니다. 새 파라미터와 일치하지 않는 인스턴스는 계정의 다른 예약 내역 할인이 적용되지 않는 한 온디맨드 요금이 부과됩니다.

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 유효 날짜는 describe-reserved-instances 명령을 사용하여 확인할 수 있습니다.
- 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.
- 변경된 예약의 고정 가격은 계정에 적용되는 할인 요금 티어에는 영향을 주지 않습니다. 할인 요금 티어는 본래 예약의 고정 가격을 기준으로 하기 때문입니다.

수정 요청이 실패할 경우 예약 인스턴스는 원래의 구성을 유지하며 다른 수정 요청이 즉시 가능합니다.

수정 비용이 없기 때문에 새로운 청구서나 인보이스를 수신하지 않습니다.

원하는 만큼 예약을 수정할 수 있지만 제출한 후에는 보류 중인 수정 요청을 변경하거나 취소할 수 없습니다. 수정이 성공적으로 처리된 후에는 필요한 경우 변경 전 상태로 되돌리기 위해 또 다른 변경 요청을 제출할 수 있습니다.

목차

- 수정 요건 및 제한 사항 (p. 298)
- 인스턴스 크기 수정을 위한 지원 (p. 299)
- 변경 요청 제출 (p. 302)
- 변경 요청 문제 해결 (p. 303)

수정 요건 및 제한 사항

이러한 속성을 다음과 같이 수정할 수 있습니다.

수정 가능한 속성	지원되는 플랫폼	제한 사항
같은 리전 내에서 가용 영역 변경	Linux 및 Windows	-
가용 영역에서 리전으로 범위 변경(반대 방향도 마찬가지)	Linux 및 Windows	범위를 가용 영역에서 리전으로 변경할 경우 용량 예약 혜택을 받을 수 없습니다. 범위를 리전에서 가용 영역으로 변경하면 가용 영역 유연성과 인

수정 가능한 속성	지원되는 플랫폼	제한 사항
		스턴스 크기 유연성이 사라집니다(있는 경우). 자세한 내용은 예약 인스턴스 적용 방식 (p. 277)을 참조하십시오.
동일한 인스턴스 패밀리 내에서 인스턴스 크기를 변경합니다.	Linux/UNIX 전용 SQL Server Standard가 설치된 Linux, SQL Server Web이 설치된 Linux, SQL Server Enterprise가 설치된 Linux, Red Hat Enterprise Linux, SUSE Linux, Windows, SQL Standard가 설치된 Windows, SQL Server Enterprise가 설치된 Windows, SQL Server Web이 설치된 Windows를 비롯한 다른 플랫폼에서는 예약 인스턴스에 인스턴스 크기 유연성이 제공되지 않습니다.	예약은 기본 테넌시를 사용해야 합니다. 사용 가능한 다른 크기가 없으므로 일부 인스턴스 패밀리는 지원되지 않습니다. 자세한 내용은 의 인스턴스 크기 수정을 위한 지원 (p. 299) 단원을 참조하십시오.
EC2-Classic에서 Amazon VPC로 네트워크 변경(그 반대도 가능)	Linux 및 Windows	네트워크 플랫폼은 AWS 계정에서 사용할 수 있어야 합니다. 2013년 12월 4일 이후 AWS 계정을 생성한 경우에는 EC2-Classic을 지원하지 않습니다.

요구 사항

Amazon EC2에서는 대상 구성에 사용할 수 있는 용량이 충분히 남아 있고(해당되는 경우) 다음 조건을 충족하는 경우 수정 요청을 처리합니다.

- 구입 전이나 구입 당시에는 예약 인스턴스를 수정할 수 없습니다.
- 예약 인스턴스는 활성 상태여야 합니다.
- 보류 중인 수정 요청이 있을 수 없습니다.
- 예약 인스턴스가 예약 인스턴스 마켓플레이스에 등록되어 있지 않습니다.
- 활성 예약과 대상 구성의 인스턴스 공간 크기가 일치해야 합니다. 자세한 내용은 인스턴스 크기 수정을 위한 지원 (p. 299)을 참조하십시오.
- 입력 예약 인스턴스는 유형 혼합 없이 모두 표준 예약 인스턴스이거나 모두 전환형 예약 인스턴스입니다.
- 입력 예약 인스턴스가 표준 예약 인스턴스인 경우 동일한 시간 내에 만료되어야 합니다.
- 예약 인스턴스는 G4 인스턴스가 아닙니다.

인스턴스 크기 수정을 위한 지원

플랫폼이 Linux/UNIX이고 인스턴스 패밀리의 크기가 여러 개인 경우 예약 인스턴스의 인스턴스 크기를 수정할 수 있습니다.

> **Note**
>
> 인스턴스들은 패밀리(스토리지 또는 CPU 용량 기준), 유형(특정 사용 사례에 맞춘 설계), 그리고 크기에 따라 그룹이 나뉩니다. 예를 들어 `c4` 인스턴스 패밀리는 컴퓨팅 최적화 패밀리에 속하고 다양한 크기로 사용 가능합니다. `c3` 인스턴스들은 같은 패밀리에 속하지만, `c4` 인스턴스들은 하드웨어 사양이 다르기 때문에 `c3` 인스턴스로 수정할 수 없습니다. 인스턴스 유형에 대한 자세한 내용은 Amazon EC2 인스턴스 유형을 참조하십시오.

인스턴스 패밀리 각각에 한 가지 크기만 사용할 수 있으므로 다음 인스턴스 유형에 대해서는 예약 인스턴스의 인스턴스 크기를 수정할 수 없습니다.

- `cc2.8xlarge`
- `cr1.8xlarge`
- `hs1.8xlarge`
- `t1.micro`

각 예약 인스턴스에는 인스턴스 공간 크기가 있으며, 이 공간 크기는 인스턴스 유형의 정규화 인자와 예약된 인스턴스 개수에 따라 결정됩니다. 예약 인스턴스를 수정하면 대상 구성의 공간이 원래 구성의 공간과 일치해야 합니다. 그렇지 않으면 수정 요청이 처리되지 않습니다.

정규화 인자는 인스턴스 패밀리 내 인스턴스 크기를 기준으로 합니다(예: `m1.xlarge` 인스턴스 패밀리 내 `m1` 인스턴스). 이는 동일한 인스턴스 패밀리 내에서만 의미가 있습니다. 한 패밀리의 인스턴스 유형을 다른 패밀리의 인스턴스 유형으로 수정할 수 없습니다. Amazon EC2 콘솔에서 정규화 인자는 유닛으로 측정됩니다. 다음 표는 동일한 인스턴스 패밀리 내에서의 정규화 인자를 설명합니다.

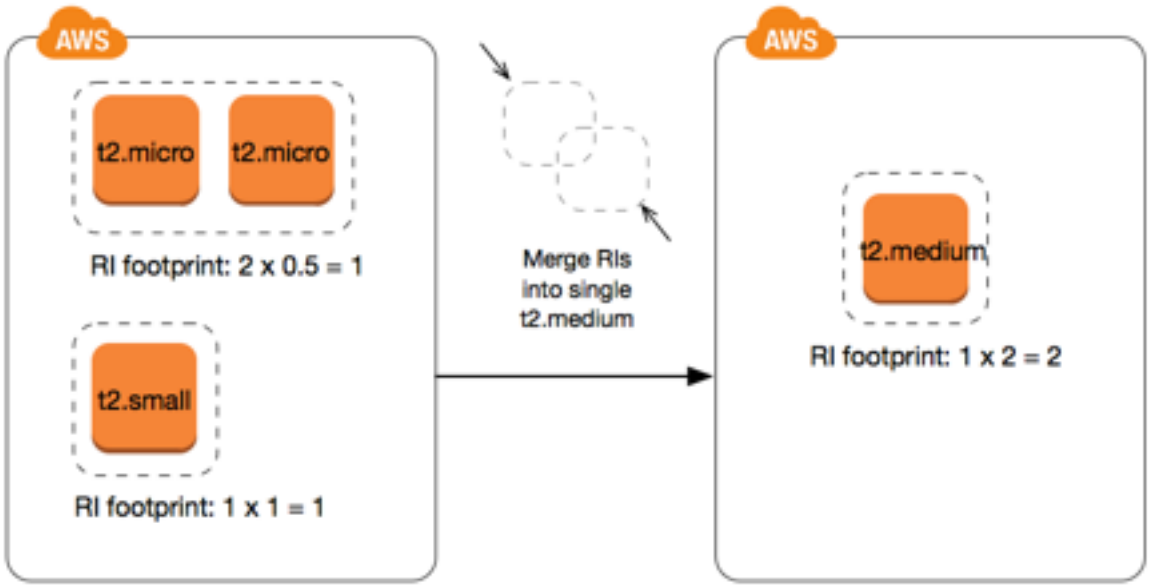
인스턴스 크기	정규화 인자
nano	0.25
micro	0.5
small	1
medium	2
large	4
xlarge	8
2xlarge	16
4xlarge	32
8xlarge	64
9xlarge	72
10xlarge	80
12xlarge	96
16xlarge	128
18xlarge	144
24xlarge	192
32xlarge	256

예약 인스턴스의 인스턴스 공간 크기는 정규화 인자에 인스턴스 수를 곱하여 산출합니다. 예를 들어 하나의 `t2.medium`은 정규화 인자 2를 가지므로, `t2.medium` 인스턴스 4개의 공간은 8유닛입니다.

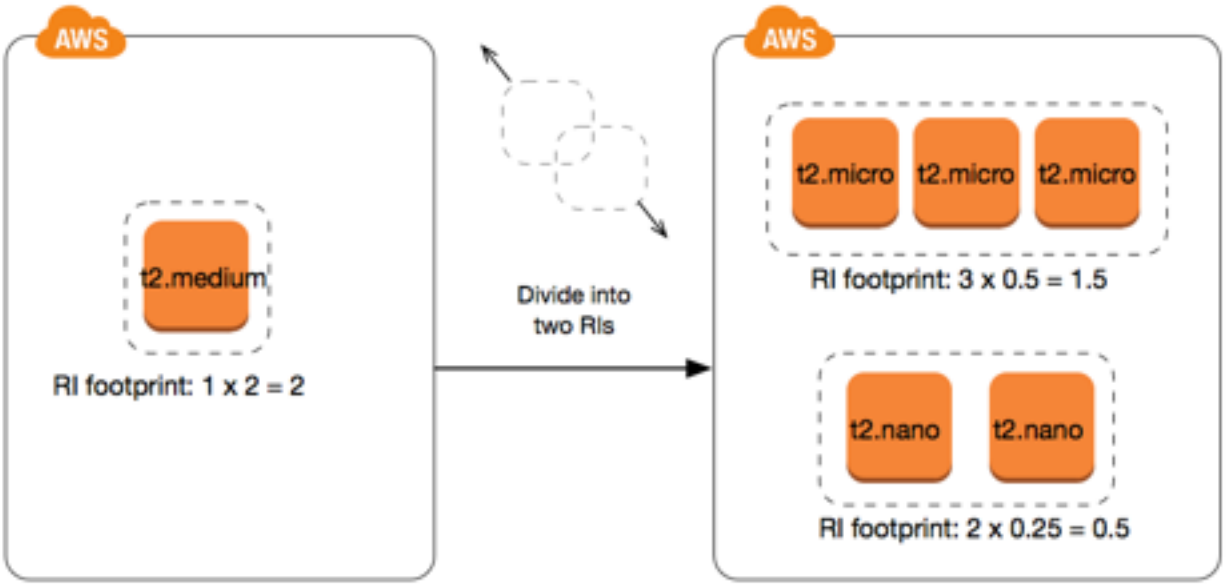
예약은 예약의 인스턴스 공간 크기가 변동되지 않는 선에서 동일한 인스턴스 패밀리(예: T2 인스턴스 패밀리) 내의 다른 인스턴스 크기로 다양하게 할당할 수 있습니다. 예를 들어, 1개의 `t2.large`(1 x 4) 인스턴스에 대한 예약을 4개의 `t2.small`(4 x 1) 인스턴스로 나누거나 반대로 기존의 `t2.small` 인스턴스 4개를 합쳐 1개의 `t2.large` 인스턴스로 변경할 수 있습니다. 하지만 2개의 `t2.small`(2 x 1) 인스턴스를 1개의

`t2.large`(1 x 4) 인스턴스로 변경할 수는 없습니다. 그 이유는 변경했을 때의 인스턴스 공간 크기가 원래 예약에 따른 인스턴스 공간 크기보다 커지기 때문입니다.

다음 예에는 `t2.micro` 인스턴스 2개가 포함된 예약(1 공간 제공)과 `t2.small` 인스턴스 1개가 포함된 예약(1 공간 제공)이 있습니다. 이러한 두 예약을 `t2.medium` 인스턴스 1개가 포함된 단일 예약으로 병합합니다. 이때 원래 예약 2개가 결합된 인스턴스 공간 크기는 수정된 예약의 공간과 같습니다.



둘 이상의 예약으로 나누도록 예약을 수정할 수도 있습니다. 다음 예에는 `t2.medium` 인스턴스가 포함된 예약이 있습니다. 이 예약을 `t2.nano` 인스턴스 2개가 포함된 예약과 `t2.micro` 인스턴스 3개가 포함된 예약으로 나눕니다.



베어 메탈 인스턴스에 대한 정규화 인자

`.metal` 예약 인스턴스를 동일한 패밀리 내 다른 크기로 수정하고 마찬가지로 동일한 패밀리 내 다른 크기의 예약 인스턴스를 `.metal` 예약 인스턴스로 수정할 수 있습니다. 베어 메탈 인스턴스의 크기는 동일한 인스턴

스 패밀리 내의 가장 큰 인스턴스의 크기와 동일합니다. 예를 들어 `i3.metal` 및 `i3.16xlarge`의 크기가 서로 동일하므로 해당 정규화 인자도 동일합니다.

> **Note**
>
> `.metal` 인스턴스 크기에는 단일 정규화 인자가 없습니다. 이 크기는 인스턴스 패밀리에 따라 달라집니다.

베어 메탈 인스턴스 크기	정규화 인자
`c5.metal`	192
`i3.metal`	128
`r5.metal`	192
`r5d.metal`	192
`z1d.metal`	96
`m5.metal`	192
`m5d.metal`	192

예를 들어 `i3.metal` 인스턴스의 정규화 인자는 128입니다. `i3.metal` 기본 테넌시 Amazon Linux/Unix 예약 인스턴스를 구입하면 다음과 같이 예약을 나눌 수 있습니다.

- `i3.16xlarge`의 크기와 `i3.metal` 인스턴스의 크기가 동일하므로 해당 정규화 인자가 128(128/1)입니다. 한 개의 `i3.metal` 인스턴스에 대한 예약이 한 개의 `i3.16xlarge` 인스턴스로 수정될 수 있습니다.
- `i3.8xlarge`의 크기가 `i3.metal` 인스턴스 크기의 반이므로 해당 정규화 인자가 64(128/2)입니다. 한 개의 `i3.metal` 인스턴스에 대한 예약이 두 개의 `i3.8xlarge` 인스턴스로 나뉠 수 있습니다.
- `i3.4xlarge`의 크기가 `i3.metal` 인스턴스 크기의 1/4이므로 해당 정규화 인자가 32(128/4)입니다. 한 개의 `i3.metal` 인스턴스에 대한 예약이 4개의 `i3.4xlarge` 인스턴스로 나뉠 수 있습니다.

변경 요청 제출

예약 인스턴스를 수정하기 전에 해당하는 제한 사항 (p. 298)을 읽어야 합니다. 인스턴스 크기를 수정하기 전에 수정할 예약의 총 인스턴스 공간 크기 (p. 299)를 계산하고, 이 값이 대상 구성의 총 인스턴스 공간 크기와 일치하는지 확인하십시오.

AWS Management 콘솔을 사용하여 예약 인스턴스를 수정하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 예약 인스턴스 페이지에서 수정할 예약 인스턴스를 하나 이상 선택하고 작업, 예약 인스턴스 수정을 선택합니다.

   > **Note**
   >
   > 예약 인스턴스가 활성 상태가 아니거나 수정이 불가능한 경우 예약 인스턴스 수정이 비활성화됩니다.

3. 수정 표의 첫 항목은 선택한 예약 인스턴스의 속성을 표시하고 그 아래에는 하나 이상의 대상 구성을 표시합니다. 단위 열에는 총 인스턴스 공간 크기가 표시됩니다. 추가할 새 구성 각각에 대해 추가를 선택합니다. 각 구성에 대해 필요에 따라 속성을 수정한 다음 계속을 선택합니다.

   - 범위: 구성이 가용 영역에 적용되는지 아니면 전체 리전에 적용되는지 선택합니다.
   - 가용 영역: 필요한 가용 영역을 선택합니다. 리전 단위의 예약 인스턴스에는 적용되지 않습니다.

- 인스턴스 유형: 필요한 인스턴스 유형을 선택합니다. 결합된 구성이 원래 구성의 인스턴스 공간 크기와 같아야 합니다.
- 개수: 인스턴스 수를 지정합니다. 예약 인스턴스를 여러 구성으로 분할하려면 개수를 줄이고 추가를 선택한 후 추가 구성의 개수를 지정합니다. 예를 들어 개수가 10인 단일 구성이 있으면 개수를 6으로 변경하고 개수가 4인 구성을 추가할 수 있습니다. 이 프로세스에서는 새 예약 인스턴스가 활성화되면 원래의 예약 인스턴스를 중지합니다.

4. 원하는 대로 구성을 지정하고 변경 사항을 확정하려면 수정 사항 제출을 선택합니다.
5. 예약 인스턴스 화면의 상태 열을 확인하여 수정 요청의 상태를 알 수 있습니다. 가능한 상태 표시는 다음과 같습니다.
   - active(수정 보류 중) - 기존 예약 인스턴스의 전환 상태
   - retired(수정 보류 중) - 새 예약 인스턴스가 생성되는 동안 기존 예약 인스턴스의 전환 상태
   - retired - 예약 인스턴스가 수정되어 교체되었습니다.
   - active - 다음 중 하나입니다.
     - 수정 요청이 성공한 경우 생성된 새 예약 인스턴스의 상태입니다.
     - 수정 요청이 실패한 후 원래 예약 인스턴스의 상태입니다.

명령줄을 사용하여 예약 인스턴스를 수정하는 방법

1. 예약 인스턴스를 수정하려면 다음 명령 중 하나를 사용할 수 있습니다.
   - modify-reserved-instances(AWS CLI)
   - Edit-EC2ReservedInstance(Windows PowerShell용 AWS 도구)
2. 수정 요청 상태(`processing`, `fulfilled` 또는 `failed`)를 확인하려면 다음 명령 중 하나를 사용하십시오.
   - describe-reserved-instances-modifications(AWS CLI)
   - Get-EC2ReservedInstancesModification(Windows PowerShell용 AWS 도구)

변경 요청 문제 해결

요청한 변경 항목이 중복되지 않는 고유한 설정이라면 요청을 처리 중이라는 메시지가 표시됩니다. 이 시점에서는 Amazon EC2에서 변경 요청의 파라미터가 유효함을 확인만 한 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. 요청을 제출하기 전에 해당하는 제한 사항 (p. 298)을 읽어 보십시오.

선택한 예약 인스턴스 모두를 변경할 수 있도록 처리할 수 있는 것은 아닙니다.

Amazon EC2에서는 변경할 수 없는 예약 인스턴스를 식별하여 표시합니다. 이 메시지가 표시되었다면 Amazon EC2 콘솔의 예약 인스턴스 페이지로 이동하여 예약 인스턴스에 대한 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 수정하려는 예약 인스턴스의 하위 집합 중 하나 이상에 대해 동일한 대상 구성(가용 영역과 플랫폼 조합)을 지정했을 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.

전환형 예약 인스턴스 교환

한 개 이상의 전환형 예약 인스턴스를 인스턴스 패밀리와 운영 체제, 테넌트를 비롯하여 구성이 다른 전환형 예약 인스턴스와 교환할 수 있습니다. 교환 횟수에 제한은 없습니다. 단, 해당 전환형 예약 인스턴스가 교환하려는 전환형 예약 인스턴스보다 가격이 높거나 같아야 합니다.

전환형 예약 인스턴스를 교환하면 현재 예약에 대한 인스턴스의 수가 대상 전환형 예약 인스턴스의 구성 값보다 크거나 같은 여러 인스턴스로 교환됩니다. Amazon EC2는 교환한 결과로 받을 수 있는 예약 인스턴스의 수를 계산합니다.

내용

- 전환형 예약 인스턴스 교환 요건 (p. 304)
- 전환형 예약 인스턴스 교환 계산 (p. 305)
- 전환형 예약 인스턴스 병합 (p. 305)
- 전환형 예약 인스턴스의 일부분 교환 (p. 306)
- 교환 요청 제출 (p. 307)

전환형 예약 인스턴스 교환 요건

Amazon EC2에서는 다음 조건이 충족될 경우 교환 요청을 처리합니다. 전환형 예약 인스턴스가 다음 조건을 충족해야 합니다.

- 활성 상태
- 이전 교환 요청이 보류 중이지 않음

다음 규칙이 적용됩니다.

- 전환형 예약 인스턴스는 AWS에서 현재 제공하는 다른 전환형 예약 인스턴스하고만 교환할 수 있습니다.
- 전환형 예약 인스턴스는 예약 기간 동안 고정된 특정 리전과 연결됩니다. 전환형 예약 인스턴스를 다른 리전의 전환형 예약 인스턴스와 교환할 수 없습니다.
- 한 번에 전환형 예약 인스턴스 하나만 한 개 이상의 전환형 예약 인스턴스로 교환할 수 있습니다.
- 전환형 예약 인스턴스의 일부분을 교환하려면 둘 이상의 예약으로 수정한 다음 예약의 한 개 이상을 새 전환형 예약 인스턴스로 교환하면 됩니다. 자세한 내용은 전환형 예약 인스턴스의 일부분 교환 (p. 306) 단원을 참조하십시오. 예약 인스턴스 변경에 대한 자세한 내용은 예약 인스턴스 수정 (p. 297) 단원을 참조하십시오.
- 전체 선결제 전환형 예약 인스턴스를 부분 선결제 전환형 예약 인스턴스로 교환할 수 있으며 그 반대로도 교환할 수 있습니다.

  Note

  교환에 필요한 총 선결제 금액(트루업 요금)이 $0.00 미만인 경우 AWS는 트루업 요금이 $0.00 이상이 되도록 전환형 예약 인스턴스에 인스턴스 수량을 자동으로 제공합니다.

  Note

  새로운 전환형 예약 인스턴스의 총 금액(선결제 금액 + 시간당 요금 * 남은 시간 수)이 교환한 전환형 예약 인스턴스의 총 금액보다 낮은 경우 AWS는 총 금액이 교환한 전환형 예약 인스턴스의 금액과 같거나 그 이상이 되도록 전환형 예약 인스턴스에 인스턴스 수량을 자동으로 제공합니다.
- 요금 혜택을 더 받기 위해 선결제 없음 전환형 예약 인스턴스를 전체 선결제 또는 부분 선결제 전환형 예약 인스턴스와 교환할 수 있습니다.
- 전체 선결제 또는 부분 선결제 전환형 예약 인스턴스를 선결제 없음 전환형 예약 인스턴스와 교환할 수는 없습니다.

- 새로운 전환형 예약 인스턴스의 시간당 가격이 교환한 전환형 예약 인스턴스의 시간당 가격과 같거나 그 이상인 경우에만 선결제 없음 전환형 예약 인스턴스를 다른 선결제 없음 전환형 예약 인스턴스와 교환할 수 있습니다.

  Note

  새로운 전환형 예약 인스턴스의 총 금액(시간당 요금 * 남은 시간 수)이 교환한 전환형 예약 인스턴스의 총 금액보다 낮은 경우 AWS는 총 금액이 교환한 전환형 예약 인스턴스의 금액과 같거나 그 이상이 되도록 전환형 예약 인스턴스에 인스턴스 수량을 자동으로 제공합니다.
- 만료 날짜가 서로 다른 여러 전환형 예약 인스턴스를 교환하는 경우 새 전환형 예약 인스턴스의 만료 날짜는 더 나중에 오는 날짜입니다.
- 단일 전환형 예약 인스턴스를 교환하는 경우 새 전환형 예약 인스턴스와 기간이 동일해야 합니다(1년 또는 3년). 기간 길이가 다른 여러 전환형 예약 인스턴스를 병합하는 경우 새 전환형 예약 인스턴스의 기간은 3년입니다. 자세한 내용은 전환형 예약 인스턴스 병합 (p. 305) 단원을 참조하십시오.
- 전환형 예약 인스턴스 교환 후 원래 예약은 종료됩니다. 원래 예약의 종료일은 새로운 예약의 시작일이 되며, 새 예약의 종료일은 원래 전환형 예약 인스턴스의 종료일과 동일합니다. 예를 들어, 3년 약정 예약 중 16개월 남은 시점에서 변경했다면, 변경된 예약은 16개월 동안 사용이 가능하며 본래 예약의 종료일과 같은 날짜에 사용 기간이 만료됩니다.

전환형 예약 인스턴스 교환 계산

전환형 예약 인스턴스 교환은 무료입니다. 하지만 트루업(true-up) 비용을 지불해야 할 수 있습니다. 이 비용은 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 인스턴스 예약의 수를 결정하기 위해 이 정가를, 원하는 전환형 예약 인스턴스의 정가와 비교합니다.

정가가 $35인 전환형 예약 인스턴스 1개를 정가가 $10인 새 인스턴스 유형과 교환하려는 경우를 예로 들어 보겠습니다.

```
$35/$10 = 3.5
```

전환형 예약 인스턴스를 10 USD 전환형 예약 인스턴스 세 개와 교환할 수는 없습니다. 절반의 동일 시작 인스턴스를 구입할 수는 없으므로 전환형 예약 인스턴스를 추가로 구입하여 나머지를 채워야 합니다.

```
3.5 = 3 whole ### ## #### + 1 additional ### ## ####.
```

4번째 전환형 예약 인스턴스는 다른 3개와 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 500 USD이고 대상 동일 시작 인스턴스가 비례 할당 계산 기준으로 600 USD일 경우 100 USD가 청구됩니다.

```
$600 prorated upfront cost of new reservations - $500 remaining upfront cost of original
 reservations = $100 difference.
```

전환형 예약 인스턴스 병합

둘 이상의 전환형 예약 인스턴스를 병합하는 경우 새 전환형 예약 인스턴스의 기간은 원래 전환형 예약 인스턴스와 동일하거나 원래 전환형 예약 인스턴스보다 길어야 합니다. 새 전환형 예약 인스턴스의 만료 날짜는 더 나중에 오는 만료 날짜입니다.

예를 들어 계정에 다음과 같은 전환형 예약 인스턴스가 있다고 가정합시다.

예약 인스턴스 ID	기간	만료 날짜
aaaa1111	1년	2018-12-31
bbbb2222	1년	2018-07-31
cccc3333	3년	2018-06-30
dddd4444	3년	2019-12-31

- `aaaa1111`과 `bbbb2222`를 병합하여 1년 전환형 예약 인스턴스로 변경할 수 있습니다. 3년 전환형 예약 인스턴스로는 변경할 수 없습니다. 새 전환형 예약 인스턴스의 만료 날짜는 2018-12-31입니다.
- `bbbb2222`과 `cccc3333`를 병합하여 3년 전환형 예약 인스턴스로 변경할 수 있습니다. 1년 전환형 예약 인스턴스로는 변경할 수 없습니다. 새 전환형 예약 인스턴스의 만료 날짜는 2018-07-31입니다.
- `cccc3333`과 `dddd4444`를 병합하여 3년 전환형 예약 인스턴스로 변경할 수 있습니다. 1년 전환형 예약 인스턴스로는 변경할 수 없습니다. 새 전환형 예약 인스턴스의 만료 날짜는 2019-12-31입니다.

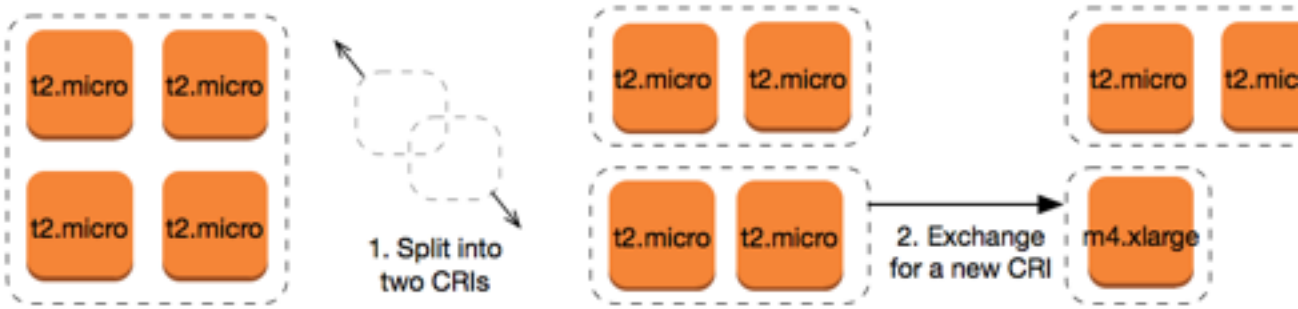
전환형 예약 인스턴스의 일부분 교환

전환형 예약 인스턴스를 작은 예약으로 분리하는 수정 과정을 이용한 다음 새 예약의 한 개 이상을 새 전환형 예약 인스턴스로 교환하면 됩니다. 다음은 그 방법을 설명하는 예제입니다.

Example 예: 인스턴스가 여럿인 전환형 예약 인스턴스

이 예에는 예약에 네 인스턴스가 있는 `t2.micro` 전환형 예약 인스턴스가 있습니다. 두 `t2.micro` 인스턴스를 `m4.xlarge` 인스턴스 한 개로 교환하려면,

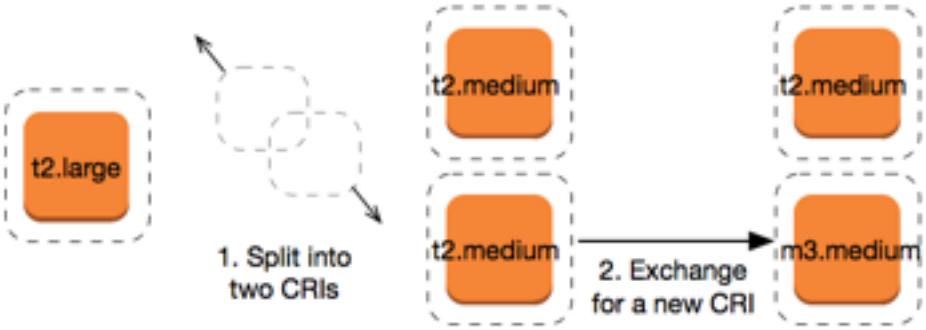
1. `t2.micro` 전환형 예약 인스턴스를 각각 두 인스턴스를 가진 두 `t2.micro` 전환형 예약 인스턴스로 분할하여 수정합니다.
2. 새 `t2.micro` 전환형 예약 인스턴스 중 하나를 `m4.xlarge` 전환형 예약 인스턴스 하나와 교환합니다.



Example 예: 인스턴스가 하나인 전환형 예약 인스턴스

이 예에는 `t2.large` 전환형 예약 인스턴스가 한 개 있습니다. 작은 `t2.medium` 인스턴스 한 개와 `m3.medium` 인스턴스 한 개로 바꾸려면,

1. `t2.large` 전환형 예약 인스턴스를 각각 두 인스턴스를 가진 두 `t2.medium` 전환형 예약 인스턴스로 분할하여 수정합니다. `t2.large` 인스턴스 하나의 인스턴스 공간 크기는 `t2.medium` 인스턴스 두 개의 인스턴스 공간 크기와 동일합니다.
2. 새 `t2.medium` 전환형 예약 인스턴스 중 하나를 `m3.medium` 전환형 예약 인스턴스 하나와 교환합니다.



자세한 내용은 인스턴스 크기 수정을 위한 지원 (p. 299) 및 교환 요청 제출 (p. 307) 단원을 참조하십시오.

교환 요청 제출

Amazon EC2 콘솔 또는 명령줄 도구를 사용하여 전환형 예약 인스턴스를 교환할 수 있습니다.

콘솔을 사용한 전환형 예약 인스턴스 교환

전환형 예약 인스턴스 상품을 검색하고 제공된 선택지에서 새로운 구성을 선택할 수 있습니다.

Amazon EC2 콘솔을 사용해 전환형 예약 인스턴스를 교환하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 예약 인스턴스를 선택하여 교환할 전환형 예약 인스턴스를 선택한 후 작업, 예약 인스턴스 교환을 선택합니다.
3. 드롭다운 메뉴를 사용하여 원하는 구성의 속성을 선택하고 제품 찾기를 선택합니다.
4. 새 전환형 예약 인스턴스를 선택합니다. 인스턴스 수 열에는 교환에 따라 받은 예약 인스턴스의 수가 표시됩니다. 요구 사항에 맞는 전환형 예약 인스턴스를 선택한 경우 교환을 선택합니다.

교환된 예약 인스턴스는 만료되고 새로운 Amazon EC2가 예약 인스턴스 콘솔에 표시됩니다. 이 프로세스는 완료하는 데 몇 분 정도 걸릴 수 있습니다.

명령줄 인터페이스를 사용한 전환형 예약 인스턴스 교환

전환형 예약 인스턴스를 교환하려면 먼저 요구 사항에 맞는 대상 전환형 예약 인스턴스를 찾습니다.

- describe-reserved-instances-offerings(AWS CLI)
- Get-EC2ReservedInstancesOffering(Windows PowerShell용 도구)

교환에서 가져오는 예약 인스턴스의 수 및 교환에 대한 트루업 비용이 포함된 교환 견적서를 가져옵니다.

- get-reserved-instances-exchange-quote(AWS CLI)
- GetEC2-ReservedInstancesExchangeQuote(Windows PowerShell용 도구)

마지막으로 교환을 수행합니다.

- accept-reserved-instances-exchange-quote(AWS CLI)
- Confirm-EC2ReservedInstancesExchangeQuote(Windows PowerShell용 도구)

정기 예약 인스턴스

정기 예약 인스턴스(예약된 인스턴스)를 사용하여 1년 동안 지정된 시작 시간 및 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 필요한 용량을 충족함과 동시에 비용을 절감할 수 있습니다. 자세한 내용은 예약 인스턴스 (p. 273) 단원을 참조하십시오. 인스턴스를 실행하는 시간이 유동적인 경우 스팟 인스턴스를 사용하면 필요한 용량을 충족함과 동시에 비용을 절감할 수 있습니다. 자세한 내용은 스팟 인스턴스 (p. 311) 단원을 참조하십시오.

내용

- 예약된 인스턴스의 작동 방식 (p. 308)
- 정기 인스턴스에 대한 서비스 연결 역할 (p. 309)
- 예약된 인스턴스 구매 (p. 309)
- 예약된 인스턴스 시작 (p. 310)
- 예약된 인스턴스 제한 (p. 311)

예약된 인스턴스의 작동 방식

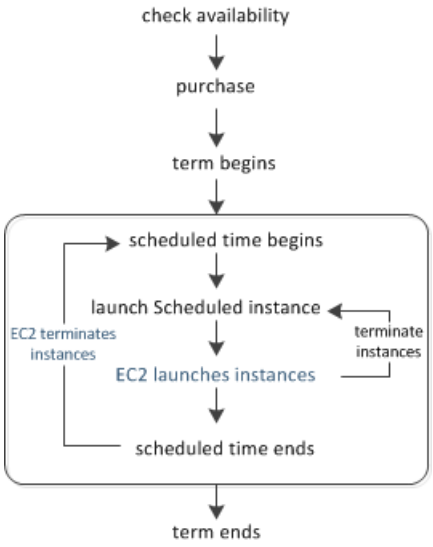
Amazon EC2는 정기 인스턴스로 사용하기 위해 각 가용 영역에서 EC2 인스턴스 풀을 무효화합니다. 각 풀은 인스턴스 유형, 운영 체제 및 네트워크의 특정 조합을 지원합니다.

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 구성을 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 중지하거나 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.



정기 인스턴스에 대한 서비스 연결 역할

정기 인스턴스를 구입하면 Amazon EC2에서 서비스 연결 역할을 만듭니다. 서비스 연결 역할은 Amazon EC2에서 다른 AWS 서비스를 자동으로 호출하기 위해 필요한 모든 권한을 포함합니다. 자세한 내용은IAM 사용 설명서의 서비스 연결 역할 사용 을 참조하십시오.

Amazon EC2는 AWSServiceRoleForEC2ScheduledInstances라는 이름의 서비스 연결 역할을 사용하여 다음 작업을 완료합니다.

- `ec2:TerminateInstances` - 일정 완료 후 정기 인스턴스 종료
- `ec2:CreateTags` - 정기 인스턴스에 시스템 태그 추가

Amazon EC2가 이 서비스 연결 역할을 지원하기 시작한 2017년 10월 이전에 정기 인스턴스를 구입한 경우, Amazon EC2에서 사용자 AWS 계정에 AWSServiceRoleForEC2ScheduledInstances 역할을 이미 생성했습니다. 자세한 내용은 IAM 사용 설명서의 내 계정에 표시되는 새 역할 을 참조하십시오.

정기 인스턴스가 더 이상 필요 없는 경우에는 AWSServiceRoleForEC2ScheduledInstances 역할을 삭제할 것을 권합니다. 계정에서 이 역할을 삭제한 후 정기 인스턴스를 구입하면 Amazon EC2에서 다시 해당 역할을 만듭니다.

예약된 인스턴스 구매

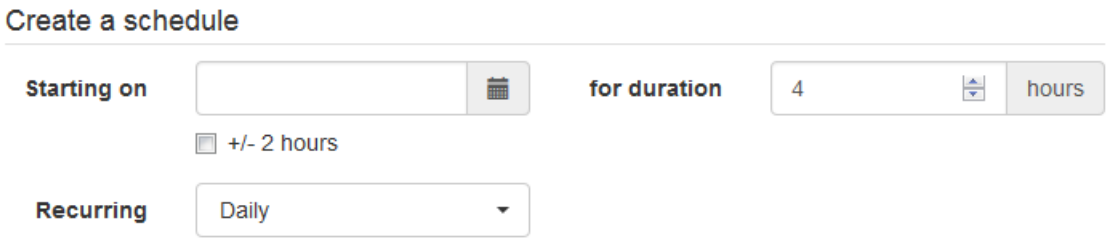
정기 인스턴스를 구입하려면 정기 예약 인스턴스 예약 마법사를 사용할 수 있습니다.

> **Warning**
>
> 정기 인스턴스를 구입한 이후에는 구입을 취소하거나, 수정하거나, 재판매할 수 없습니다.

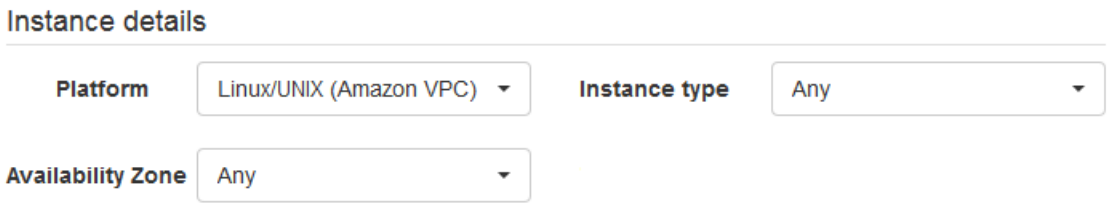
정기 인스턴스를 구입하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창의 인스턴스에서 정기 인스턴스를 선택합니다. 현재 선택한 리전에서 정기 인스턴스를 지원하지 않는 경우 페이지를 사용할 수 없습니다. 자세히 알아보기 (p. 311)
3. 정기 인스턴스 구입(Purchase Scheduled Instances)을 선택합니다.
4. 사용 가능한 일정(Find available schedules) 페이지에서 다음을 수행하십시오.

a. 일정 생성(Create a schedule)의 시작(Starting on)에서 시작 날짜를 선택하고, 반복(Recurring)에서 예약 반복(매일, 매주 또는 매월)을 선택하고, 기간(for duration)에서 최소 기간을 선택합니다. 콘솔에서는 정기 인스턴스에 필요한 최소 사용률을 충족하는 최소 기간 값을 지정하는지 확인합니다(연간 1,200시간).



b. 인스턴스 세부 정보의 플랫폼에서 운영 체제와 네트워크를 선택합니다. 결과 범위를 좁히려면 인스턴스 유형에서 하나 이상의 인스턴스 유형을 선택하거나 가용 영역에서 하나 이상의 가용 영역을 선택합니다.



c. 일정 찾기(Find schedules)를 선택합니다.

d. 사용 가능한 일정(Available schedules)에서 하나 이상의 일정을 선택합니다. 선택하는 각 일정에 대해 인스턴스의 수량을 설정한 다음 장바구니에 추가를 선택합니다.

e. 장바구니가 페이지의 아래쪽에 표시됩니다. 장바구니에서 일정 추가 및 제거를 마쳤으면 검토 및 구입(Review and purchase)을 선택합니다.

5. 검토 및 구입(Review and purchase) 페이지에서 선택 항목을 확인하고 필요한 경우 편집합니다. 작업을 마쳤으면 구입을 선택합니다.

정기 인스턴스를 구입하려면(AWS CLI)

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.

정기 인스턴스를 시작하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창의 인스턴스에서 정기 인스턴스를 선택합니다. 현재 선택한 리전에서 정기 인스턴스를 지원하지 않는 경우 페이지를 사용할 수 없습니다. 자세히 알아보기 (p. 311)
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. 검토 페이지에서 시작 구성을 확인하고 필요한 경우 수정합니다. 작업을 마쳤으면 시작을 선택합니다.

정기 인스턴스를 시작하려면(AWS CLI)

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.
- 이러한 인스턴스는 현재 다음 리전에서 이용 가능합니다. 미국 동부(버지니아 북부), 미국 서부(오리건) 및 유럽(아일랜드).

스팟 인스턴스

스팟 인스턴스는 온디맨드 가격보다 저렴한 비용으로 사용할 수 있는 미사용 EC2 인스턴스입니다. 스팟 인스턴스는 큰 할인율로 미사용 EC2 인스턴스를 요청할 수 있게 해주므로 사용자는 Amazon EC2 비용을 대폭 낮출 수 있습니다. 스팟 인스턴스의 시간당 가격을 스팟 가격이라고 합니다. 각 가용 영역 내 인스턴스 유형별 스팟 가격은 Amazon EC2에서 설정하며, 스팟 인스턴스의 장기적 공급 및 수요에 따라 점진적으로 조정됩니다. 스팟 인스턴스는 용량이 가용 상태이고 요청에 대한 시간당 최고 가격이 스팟 가격보다 더 높을 때마다 실행됩니다.

스팟 인스턴스는 애플리케이션이 실행되는 시간을 유연하게 조정할 수 있고 애플리케이션을 중단할 수 있는 경우에 선택하는 비용 효율적인 방법입니다. 예를 들어 스팟 인스턴스는 데이터 분석, 배치 작업, 백그라운드 프로세싱 및 선택적 작업에 적합합니다. 자세한 내용은 Amazon EC2 스팟 인스턴스 단원을 참조하십시오.

주제

- 개념 (p. 311)
- 시작하는 방법 (p. 312)
- 관련 서비스 (p. 313)
- 요금 및 비용 절감 (p. 314)

개념

스팟 인스턴스를 시작하기 전에 다음 개념을 익혀야 합니다.

- 스팟 인스턴스 풀 – 동일한 인스턴스 유형(예: `m5.large`), 운영 체제, 가용 영역 및 네트워크 플랫폼을 가지는 미사용 EC2 인스턴스의 세트입니다.
- 스팟 가격 – 스팟 인스턴스의 시간당 현재 가격입니다.
- 스팟 인스턴스 요청 – 스팟 인스턴스에 대해 지불하고자 하는 시간당 최고 가격을 제공합니다. 최고 가격을 지정하지 않는 경우, 기본 최고 가격은 온디맨드 가격입니다. 요청에 대한 시간당 최고 가격이 스팟 가격을 초과할 경우, Amazon EC2는 용량이 가용 상태가 되면 요청을 이행합니다. 스팟 인스턴스 요청은 일회성이거나 영구적입니다. Amazon EC2는 요청과 연결된 스팟 인스턴스가 종료된 후 자동으로 영구 스팟 요청을 다시 제출합니다. 스팟 인스턴스 요청은 스팟 인스턴스에 대해 지속 시간을 선택적으로 지정할 수 있는 옵션이 있습니다.
- 스팟 집합 – 사용자가 지정한 기준을 바탕으로 시작되는 스팟 인스턴스 세트입니다. 스팟 집합에서는 사용자의 요구를 충족하는 스팟 인스턴스 풀을 선택하고 플릿에 대한 목표 용량을 충족하는 스팟 인스턴스를 시작합니다. 기본적으로 스팟 집합은 플릿에서 스팟 인스턴스가 종료된 후 교체 인스턴스를 시작하여 목표 용량을 유지하도록 설정되어 있습니다. 스팟 집합을 인스턴스가 종료된 후에는 유지되지 않는 일회성 요청으로 제출할 수도 있습니다. 스팟 집합 요청에 온디맨드 인스턴스 요청을 포함할 수 있습니다.

- 스팟 인스턴스 중단 – 스팟 가격이 요청에 대한 최고 가격을 초과하거나 사용할 수 있는 용량이 더 이상 없는 경우 Amazon EC2는 스팟 인스턴스를 종료 또는 중지시키거나 최대 절전 모드로 전환합니다. Amazon EC2는 스팟 인스턴스 중단 공지를 통해 중지 2분 전에 이를 인스턴스에 경고합니다.

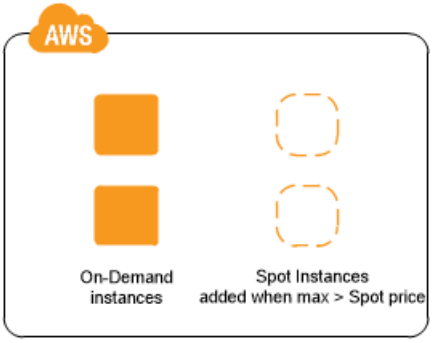
스팟 인스턴스와 온디맨드 인스턴스의 주요 차이점

다음 표에는 스팟 인스턴스와 온디맨드 인스턴스의 주요 차이점이 나열되어 있습니다.

	스팟 인스턴스	온디맨드 인스턴스
시작 시간	스팟 요청이 활성 상태이고 용량이 가용 상태인 경우 즉시 시작할 수 있습니다.	수동 시작을 요청했고 용량이 가용 상태인 경우에만 즉시 시작할 수 있습니다.
가용 용량	용량이 가용 상태가 아닌 경우 용량이 가용 상태가 될 때까지 스팟 요청이 계속해서 자동으로 시작 요청을 합니다.	시작 요청을 할 때 용량이 가용 상태가 아닌 경우 용량 부족 오류(ICE)가 발생합니다.
시간당 가격	스팟 인스턴스의 시간당 가격은 온디맨드 기준 가격입니다.	온디맨드 인스턴스의 시간당 가격은 고정된 가격입니다.
인스턴스 중단	Amazon EBS 지원 스팟 인스턴스를 중지하고 시작할 수 있습니다. 또한 용량을 더 이상 사용할 수 없거나 스팟 가격이 최고 가격을 초과하거나 스팟 인스턴스에 대한 수요가 증가하는 경우 Amazon EC2 스팟 서비스에서 개별 스팟 인스턴스을 중단 (p. 372)할 수 있습니다.	온디맨드 인스턴스가 중단(중지, 최대 절전 또는 종료)되는 시간을 결정합니다.

스팟 인스턴스 사용 전략

애플리케이션에 대해 보장된 컴퓨팅 리소스를 최소 수준으로 유지하기 위한 한 가지 전략은 온디맨드 인스턴스의 코어 그룹을 시작하고 기회가 생기면 스팟 인스턴스로 이를 보완하는 것입니다.



또 다른 전략은 지정된 지속 시간(스팟 블록이라고도 함)을 갖춘 스팟 인스턴스를 시작하는 것인데, 이것은 선택한 지속 시간 동안에는 중단되지 않으며 계속 실행됩니다. 드문 경우 Amazon EC2 용량 때문에 스팟 블록이 중단될 수 있습니다. 이 경우에는 인스턴스를 종료하기 2분 전에 경고를 보내며, 종료된 인스턴스를 사용한 경우에도 해당 인스턴스에 대해 요금이 부과되지 않습니다. 자세한 내용은 스팟 인스턴스의 지속 시간 정의 (p. 326) 단원을 참조하십시오.

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. 22)
- Amazon EC2 Linux 인스턴스 시작하기 (p. 27)

스팟 기본 사항

- 스팟 인스턴스의 작동 방식 (p. 314)
- 스팟 집합 작동 방식 (p. 316)

스팟 인스턴스 작업

- 중단에 대한 준비 (p. 375)
- 스팟 인스턴스 요청 생성 (p. 329)
- 요청 상태 정보 가져오기 (p. 370)

스팟 집합 작업

- 스팟 집합 사전 조건 (p. 336)
- 스팟 집합 요청 생성 (p. 339)

관련 서비스

Amazon EC2를 사용하여 스팟 인스턴스를 직접 프로비저닝할 수 있습니다. 또한 AWS의 다른 서비스를 사용하여 스팟 인스턴스를 프로비저닝할 수 있습니다. 자세한 내용은 다음 설명서를 참조하십시오.

Amazon EC2 Auto Scaling 및 스팟 인스턴스

Amazon EC2 Auto Scaling에서 스팟 인스턴스를 시작할 수 있도록 지불하고자 하는 최고 가격으로 시작 템플릿이나 구성을 생성할 수 있습니다. 자세한 내용은 Auto Scaling 그룹에서 스팟 인스턴스 시작과 Amazon EC2 Auto Scaling 사용 설명서의 여러 인스턴스 유형 및 구매 옵션 사용을 참조하십시오.

Amazon EMR 및 스팟 인스턴스

Amazon EMR 클러스터에서 스팟 인스턴스를 실행하는 것이 유용할 수 있는 시나리오가 있습니다. 자세한 내용은 Amazon EMR 관리 안내서의 스팟 인스턴스 및 스팟 인스턴스를 언제 사용해야 합니까? 단원을 참조하십시오.

AWS CloudFormation 템플릿

AWS CloudFormation에서는 JSON 형식의 템플릿을 사용하여 AWS 리소스 컬렉션을 생성하고 관리할 수 있습니다. AWS CloudFormation 템플릿은 지불하고자 하는 최고 가격을 포함할 수 있습니다. 자세한 내용은 EC2 스팟 인스턴스 Updates - Auto Scaling and CloudFormation Integration을 참조하십시오.

AWS SDK for Java

Java 프로그래밍 언어를 사용하여 스팟 인스턴스를 관리할 수 있습니다. 자세한 내용은 자습서: Amazon EC2 스팟 인스턴스 및 자습서: 고급 Amazon EC2 스팟 요청 관리를 참조하십시오.

.NET용 AWS SDK

.NET 프로그래밍 환경을 사용하여 스팟 인스턴스를 관리할 수 있습니다. 자세한 내용은 자습서: Amazon EC2 스팟 인스턴스 단원을 참조하십시오.

요금 및 비용 절감

스팟 인스턴스에 대해 스팟 가격을 지불합니다. 이 가격은 Amazon EC2에서 설정되며 스팟 인스턴스의 장기적 수요 및 공급에 따라 점진적으로 조정됩니다. 요청에 대한 최고 가격이 현재 스팟 가격을 초과하는 경우 Amazon EC2는 용량이 가용 상태가 될 때 요청을 이행합니다. 스팟 인스턴스는 사용자가 직접 종료를 하거나 용량이 더 이상 가용 상태가 아니거나 스팟 가격이 최고 가격을 초과하거나 Amazon EC2 Auto Scaling 그룹이 축소 도중 종료할 때까지 실행됩니다.

미리 정의된 지속 시간이 있는 스팟 인스턴스는 실행 중에 스팟 인스턴스에 대해 여전히 유효한 시간당 고정 가격을 사용합니다.

사용자 또는 Amazon EC2가 실행 중인 스팟 인스턴스를 중단하는 경우, 사용되는 운영 체제와 누가 스팟 인스턴스를 중단했는지에 따라 사용된 시간(초) 또는 전체 시간에 대한 요금이 부과되거나 요금이 무료일 수 있습니다. 자세한 내용은 중단된 스팟 인스턴스에 대한 청구 (p. 378) 단원을 참조하십시오.

가격 보기

AWS 지역 및 인스턴스 유형당 현재(5분마다 업데이트됨) 최저 스팟 가격을 보려면 스팟 인스턴스 요금 페이지를 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. 322) 단원을 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.

비용 절감액 보기

단일 스팟 집합 또는 모든 스팟 인스턴스에 대해 스팟 인스턴스 사용에서 얻는 절감을 볼 수 있습니다. 지난 1시간 또는 지난 3일 동안 실현된 절감액을 볼 수 있으며, vCPU 시간당 평균 비용 및 메모리(GiB) 단위 시간당 평균 비용을 볼 수 있습니다. 절감액은 추정치이며 사용량에 대한 청구 조정이 제외되어 있기 때문에 실제 절감액과 다를 수 있습니다. 비용 절감액 보기에 대한 자세한 내용은 스팟 인스턴스 구입으로 절감되는 비용 (p. 323) 단원을 참조하십시오.

결제 보기

청구 요금을 검토하려면 AWS 계정 활동 페이지를 참조하십시오. 청구서에는 요금 내역을 자세하게 확인할 수 있는 사용 보고서 링크가 포함됩니다. 자세한 내용은 AWS Account Billing을 참조하십시오.

AWS 결제, 계정 및 이벤트에 관련된 질문은 AWS Support에 문의하십시오.

스팟 인스턴스의 작동 방식

스팟 인스턴스를 사용하려면 스팟 인스턴스 요청 또는 스팟 집합 요청을 생성하십시오. 이 요청에는 인스턴스별로 시간당 지불하려는 최고 가격(기본은 온디맨드 가격)과 인스턴스 유형 및 가용 영역과 같은 기타 제약 조건이 포함될 수 있습니다. 지정된 인스턴스에서 최고 가격이 현재 스팟 가격을 초과하고 용량이 가용 상태인 경우 요청이 즉시 이행됩니다. 그렇지 않은 경우 최고 가격이 스팟 가격을 초과하고 용량이 가용 상태가 될 때마다 요청이 이행됩니다. 스팟 인스턴스는 사용자가 인스턴스를 중지하거나 종료할 때까지 또는 Amazon EC2가 인스턴스를 중단(스팟 인스턴스 중단이라고 함)해야 할 때까지 실행됩니다.

스팟 인스턴스를 사용할 때는 중단에 대비해야 합니다. Amazon EC2는 스팟 가격이 최고 가격을 초과하거나 스팟 인스턴스에 대한 수요가 증가하거나 스팟 인스턴스의 공급이 감소할 때 스팟 인스턴스를 중단할 수 있습니다. Amazon EC2는 스팟 인스턴스를 중단할 때 스팟 인스턴스 중단 공지를 보내서 Amazon EC2가 중지하기 2분 전에 이를 인스턴스에 경고합니다. 스팟 인스턴스에 대한 종료 방지 기능은 활성화할 수 없습니다. 자세한 내용은 스팟 인스턴스 중단 (p. 372) 단원을 참조하십시오.

Amazon EBS 지원 인스턴스를 중지, 시작, 재부팅 또는 종료할 수 있습니다. 스팟 서비스는 인스턴스를 중단할 때 스팟 인스턴스를 중지, 종료 또는 최대 절전 모드로 설정할 수 있습니다.

목차

- 시작 그룹에서 스팟 인스턴스 시작 (p. 315)
- 가용 영역 그룹에서 스팟 인스턴스 시작 (p. 315)
- VPC에서 스팟 인스턴스 시작 (p. 315)

시작 그룹에서 스팟 인스턴스 시작

스팟 인스턴스 요청에서 시작 그룹을 지정하여 해당 인스턴스를 모두 시작할 수 있는 경우에만 스팟 인스턴스 집합을 시작하도록 Amazon EC2에 알립니다. 또한 스팟 서비스가 시작 그룹에 있는 인스턴스 중 하나를 종료해야 하는 경우(예를 들어, 스팟 가격이 최고 가격을 초과하는 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. 예를 들어, 시작 그룹에 다중 가용 영역의 인스턴스가 포함되어 있습니다. 이러한 가용 영역 중 하나에서 용량이 감소되어 더는 사용할 수 없는 상태인 경우 Amazon EC2에서는 이 시작 그룹에 대해 모든 인스턴스를 종료합니다.

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 시작

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 중단할 필요는 없습니다. Amazon EC2가 가용 영역 그룹의 인스턴스를 하나 중단해야 하는 경우 다른 인스턴스는 실행 중인 상태로 유지됩니다.

이 옵션이 유용할 수 있지만 이러한 제약 조건을 추가하면 스팟 인스턴스 요청이 이행될 가능성이 낮아질 수 있습니다.

가용 영역 그룹을 지정하지만 스팟 인스턴스 요청에서 가용 영역을 지정하지 않는 경우 결과는 무엇을 지정했는지에 따라 다릅니다.

기본 VPC

Amazon EC2는 지정된 서브넷에 대한 가용 영역을 사용합니다. 서브넷을 지정하지 않으면 가용 영역 및 해당 가용 영역의 기본 서브넷이 자동으로 선택되지만 최저 요금 영역은 선택되지 않을 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 인스턴스에 대해 서브넷을 지정하는 것과 동일한 방법으로 온디맨드 인스턴스에 대해 서브넷을 지정합니다.

- VPC에서는 기본 최고 가격(온디맨드 가격)을 사용하거나 스팟 인스턴스의 스팟 가격 기록을 토대로 최고 가격을 정해야 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 집합 작동 방식

스팟 집합은 스팟 인스턴스 및 온디맨드 인스턴스의 모음 또는 플릿입니다.

스팟 집합은 사용자가 스팟 집합 요청에서 지정한 목표 용량을 충족하는 데 필요한 수만큼 스팟 인스턴스 및 온디맨드 인스턴스를 시작하려고 합니다. 스팟 인스턴스에 대한 요청은 요청에 지정된 최대 가격이 현재 스팟 가격을 초과하고 사용 가능한 용량이 있으면 수행됩니다. 스팟 인스턴스가 중단될 경우 스팟 집합은 대상 용량 플릿을 유지하려고 시도합니다.

또한 플릿에 대해 지불할 시간당 최대 금액을 설정할 수 있으며, 스팟 집합은 최대 금액에 도달할 때까지 인스턴스를 실행합니다. 지불하려는 최대 금액에 도달하면 플릿은 목표 용량을 충족하지 않은 경우에도 인스턴스 실행을 중지합니다.

스팟 인스턴스 풀은 동일한 인스턴스 유형(예: `m5.large`), 운영 체제, 가용 영역 및 네트워크 플랫폼을 가지는 미사용 EC2 인스턴스의 세트입니다. 스팟 집합 요청을 할 때 인스턴스 유형, AMI, 가용 영역 또는 서브넷에 따라 바뀌는 여러 시작 사양을 포함할 수 있습니다. 스팟 집합은 요청에 포함된 시작 사양과 스팟 집합 요청의 구성을 기반으로 스팟 집합 요청을 이행하는 데 사용되는 스팟 인스턴스 풀을 선택합니다. 스팟 인스턴스는 선택한 풀에서 가져옵니다.

내용

- 스팟 집합의 온디맨드 (p. 316)
- 스팟 인스턴스를 위한 할당 전략 (p. 316)
- 스팟 가격 재정의 (p. 318)
- 지출 제어 (p. 318)
- 스팟 집합 인스턴스 가중치 부여 (p. 319)
- 연습: 인스턴스 가중치를 부여한 스팟 집합 사용 (p. 320)

스팟 집합의 온디맨드

항상 인스턴스 용량을 사용할 수 있도록 스팟 집합 요청에 온디맨드 용량에 대한 요청을 포함할 수 있습니다. 스팟 집합 요청에 원하는 목표 용량 및 해당 용량의 몇 %가 온디맨드 용량이어야 하는지를 지정합니다. 잔고는 스팟 용량으로 구성됩니다. 스팟 용량은 가용 Amazon EC2 용량이 있고 용량이 가용 상태일 경우 시작됩니다. 예를 들어 스팟 집합 요청에 목표 용량을 10으로 지정하고 온디맨드 용량을 8로 지정하면 Amazon EC2는 8개의 용량 단위를 온디맨드로 시작하고 2개(10-8=2)의 용량 단위를 스팟으로 시작합니다.

온디맨드 용량에 대한 인스턴스 유형 우선순위 지정

스팟 집합가 온디맨드 용량을 채우려고 시도하는 경우 기본적으로 최저 가격의 인스턴스 유형을 먼저 시작합니다. `OnDemandAllocationStrategy`가 `prioritized`로 설정된 경우 스팟 집합가, 우선 순위를 통해, 온디맨드 용량을 채우기 위해 먼저 사용할 인스턴스 유형을 결정합니다. 시작 템플릿 재정의에 우선 순위를 할당하고 우선 순위가 가장 높은 것을 먼저 시작합니다.

예를 들어 서로 다른 인스턴스 유형인 `c3.large`, `c4.large`, `c5.large`를 각각 지닌 3개의 시작 템플릿 재정의를 구성했다고 가정해 보겠습니다. `c5.large`에 대한 온디맨드 가격은 `c4.large`에 대한 온디맨드 가격보다 낮습니다. `c3.large`가 가장 낮습니다. 우선 순위를 사용해 순서를 결정하지 않는 경우 플릿이 `c3.large`로 시작하여 온디맨드 용량을 채운 후 `c5.large`를 사용합니다. 종종 `c4.large`에 대한 미사용 예약 인스턴스가 있게 되므로 `c4.large`, `c3.large`, `c5.large`의 순서이도록 시작 템플릿 재정의 우선 순위를 설정할 수 있습니다.

스팟 인스턴스를 위한 할당 전략

스팟 집합의 스팟 인스턴스 할당 전략에 따라 시작 사양으로 표시되는 가능한 스팟 인스턴스 풀에서 스팟 집합에 대한 요청을 이행하는 방법이 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 두루 분산됩니다.

`capacityOptimized`

스팟 인스턴스는 시작하는 인스턴스의 수에 대한 용량이 최적화된 풀에서 가져옵니다.

`InstancePoolsToUseCount`

스팟 인스턴스는 지정한 스팟 풀 수에 걸쳐 배포됩니다. 이 파라미터는 `lowestPrice`와 함께 사용하는 경우에만 유효합니다.

목표 용량 유지

스팟 가격 또는 스팟 인스턴스 풀의 가용 용량 변화로 인해 스팟 인스턴스가 종료된 후에는 `maintain` 유형의 스팟 집합가 대체 스팟 인스턴스를 시작합니다. 할당 전략이 `lowestPrice`인 경우, 플릿은 현재 스팟 가격이 가장 낮은 풀에서 대체 인스턴스를 시작합니다. 할당 전략이 `diversified`인 경우 플릿은 나머지 풀에 대체 스팟 인스턴스를 배포합니다. 할당 전략으로써 `InstancePoolsToUseCount`와 함께 `lowestPrice`를 사용하는 경우 플릿이 최저 가격의 스팟 풀을 선택하여 지정한 스팟 풀 수에 걸쳐 스팟 인스턴스를 시작합니다.

비용 최적화를 위한 스팟 집합 구성

스팟 인스턴스 사용 비용을 최적화하려면 스팟 집합이 현재 스팟 가격을 기반으로 인스턴스 유형과 가용 영역의 가장 저렴한 조합을 자동으로 배포하도록 `lowestPrice` 할당 전략을 지정합니다.

스팟 집합은 항상 퍼블릭 온디맨드 가격을 기반으로 최저 비용의 인스턴스 유형을 온디맨드 인스턴스 목표 용량에 대해 선택하며, 스팟 인스턴스에 대해서는 계속해서 할당 전략(`lowestPrice`, `capacityOptimized` 또는 `diversified`)을 따릅니다.

비용 최적화 및 다각화를 위한 스팟 집합 구성

저렴함과 다각화를 모두 충족하는 스팟 인스턴스 플릿을 생성하려면 `InstancePoolsToUseCount`과 함께 `lowestPrice` 할당 전략을 사용하십시오. 스팟 집합가 지정된 스팟 풀 수에 걸쳐 현재 스팟 가격을 기반으로 인스턴스 유형과 가용 영역의 가장 저렴한 조합을 자동으로 배포합니다. 이 조합을 통해 가장 비싼 스팟 인스턴스를 피할 수 있습니다.

용량 최적화를 위한 스팟 집합 구성

스팟 인스턴스에서 요금은 시간이 지나면서 수요 및 공급의 장기 추세에 따라 서서히 변화하지만 용량은 실시간으로 변동합니다. `capacityOptimized` 전략은 실시간 용량 데이터를 기준으로 가장 가용성이 높은 풀을 예측하여 자동으로 스팟 인스턴스를 가장 가용성이 높은 풀로 시작합니다. 이 전략은 빅 데이터 및 분석, 이미지 및 미디어 렌더링, 기계 학습, 고성능 컴퓨팅과 같이 작업 재시작 및 체크포인트와 관련된 중단으로 인한 비용이 더 높을 수 있는 워크로드에 유용합니다. `capacityOptimized` 전략은 중단을 줄일 수 있는 가능성을 제공함으로써 전체 워크로드 비용을 낮출 수 있습니다.

적합한 할당 전략 선택

사용 사례를 바탕으로 스팟 집합을 최적화할 수 있습니다.

플릿이 작거나 짧은 시간 동안 실행될 경우 모든 인스턴스가 단일 스팟 인스턴스 풀에 있더라도 스팟 인스턴스가 중단될 확률은 낮습니다. 따라서 `lowestPrice` 전략이 요구를 충족시키는 동시에 최저 가격을 제공할 가능성이 높습니다.

플릿이 크거나 장시간 실행될 경우 스팟 인스턴스를 여러 풀로 분산하여 플릿의 가용성을 높일 수 있습니다. 예를 들어 스팟 집합 요청이 풀 10개와 인스턴스 100개의 목표 용량을 지정하면 플릿이 각 풀에서 스팟 인스턴스 10개를 시작합니다. 풀에서 스팟 가격이 최고 가격을 초과하는 경우, 플릿 중 10%만 영향을 받습니다. 이 전략을 사용하면 플릿이 시간이 지나면서 어느 한 풀에서 발생하는 스팟 가격의 상승에 덜 민감해집니다.

`diversified` 전략 사용 시 스팟 집합은 온디맨드 가격보다 높거나 이 가격과 동일한 스팟 가격의 풀로 스팟 인스턴스를 시작하지 않습니다.

저렴하고 다각화된 플릿을 생성하려면 `lowestPrice`과 함께 `InstancePoolsToUseCount` 전략을 사용하십시오. 스팟 인스턴스를 할당할 스팟 풀 수로써 낮은 수 또는 높은 수를 사용할 수 있습니다. 예를 들어 일괄 처리를 실행하는 경우 대기열이 비용 절감 효과를 극대화하는 동시에 컴퓨팅 파워를 항상 확보할 수 있도록 스팟 풀 수를 낮게(예: `InstancePoolsToUseCount=2`) 지정하는 것이 좋습니다. 웹 서비스를 실행하는 경우 스팟 인스턴스 풀을 일시적으로 사용할 수 없게 되었을 때 그 충격을 최소화할 수 있도록 스팟 풀 수를 낮게(예: `InstancePoolsToUseCount=10`) 지정하는 것이 좋습니다.

플릿이 작업 재시작 및 검사와 연관된 중단 비용이 높을 수 있는 워크로드를 실행하는 경우 `capacityOptimized` 전략을 사용하십시오. 이 전략은 중단을 줄일 수 있는 가능성을 제공함으로써 전체 워크로드 비용을 낮출 수 있습니다.

스팟 가격 재정의

각 스팟 집합 요청은 글로벌 최고 가격을 포함하거나 기본 가격(온디맨드 가격)을 사용할 수 있습니다. 스팟 집합은 각각의 시작 사양에 대해 기본 최고 가격으로 이 가격을 사용합니다.

하나 이상의 시작 사양에서 최고 가격을 선택적으로 지정할 수 있습니다. 이 가격은 시작 사양에 특정한 것입니다. 시작 사양에 특정 가격이 포함되는 경우 스팟 집합은 글로벌 최고 가격 대신 이 최고 가격을 사용합니다. 특정 최고 가격을 포함하지 않는 다른 시작 사양은 글로벌 최고 가격을 계속해서 사용합니다.

지출 제어

스팟 집합은 대상 용량 또는 지불할 최대 금액에 도달하면 인스턴스 실행을 중지합니다. 플릿에 대해 시간당 지불하는 금액을 관리하려면 스팟 인스턴스의 경우 `SpotMaxTotalPrice`, 온디맨드 인스턴스의 경우 `OnDemandMaxTotalPrice`를 지정할 수 있습니다. 최대 총 가격에 도달하면 스팟 집합은 목표 용량을 충족하지 않은 경우에도 인스턴스 실행을 중지합니다.

다음 예제와 같이 이 작업을 두 가지 시나리오로 수행할 수 있습니다. 첫 번째 시나리오에서 스팟 집합은 대상 용량을 충족했을 때 인스턴스 실행을 중지합니다. 두 번째 시나리오에서 스팟 집합은 지불할 최대 금액에 도달하면 인스턴스 실행을 중지합니다.

예: 대상 용량에 도달할 때 인스턴스 실행 중지

다음과 같은 `m4.large` 온디맨드 인스턴스 요청 시:

- 온디맨드 가격: 시간당 0.10 USD
- `OnDemandTargetCapacity`: 10
- `OnDemandMaxTotalPrice`: 1.50 USD

스팟 집합은 최대 1.00 USD(10개 인스턴스 x 0.10 USD)가 `OnDemandMaxTotalPrice` 1.50 USD를 초과하지 않기 때문에 10개의 온디맨드 인스턴스를 시작합니다.

예: 최대 총 가격에 도달할 때 인스턴스 실행 중지

다음과 같은 `m4.large` 온디맨드 인스턴스 요청 시:

- 온디맨드 가격: 시간당 0.10 USD
- `OnDemandTargetCapacity`: 10
- `OnDemandMaxTotalPrice`: 0.80 USD

스팟 집합이 온디맨드 대상 용량(온디맨드 인스턴스 10개)을 시작하면 시간당 총 비용은 1.00 USD입니다. `OnDemandMaxTotalPrice`에 대해 지정된 금액(0.80 USD) 보다 높습니다. 지불할 금액보다 더 많은 지출을 방지하기 위해 스팟 집합은 8개의 온디맨드 인스턴스(온디맨드 대상 용량 미만)만 실행합니다. 더 많이 실행하면 `OnDemandMaxTotalPrice`를 초과할 수 있기 때문입니다.

스팟 집합 인스턴스 가중치 부여

스팟 인스턴스의 플릿을 요청할 때 각 인스턴스 유형이 애플리케이션의 성능에 기여하는 용량 단위를 정의하고 인스턴스 가중치를 사용하여 적절히 각 스팟 인스턴스 풀에 대한 최고 가격을 조정할 수 있습니다.

기본적으로, 사용자가 지정하는 가격은 인스턴스 시간당 가격입니다. 인스턴스 가중치 기능을 사용할 때, 사용자가 지정하는 가격은 단위 시간당 가격입니다. 단위 시간당 가격은 인스턴스 유형에 따른 가격을 인스턴스가 나타내는 단위 수로 나누어 계산합니다. 스팟 집합은 목표 용량을 인스턴스 가중치로 나누어 시작할 스팟 인스턴스 수를 계산합니다. 결과가 정수가 아닌 경우 스팟 집합은 결과를 다음 정수로 올림하므로 플릿의 크기가 목표 용량을 밑돌지는 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 요청에서 단위당 가격을 결정하기 위한 계산 예제를 제공합니다.

인스턴스 유형	인스턴스 가중치	인스턴스 시간당 가격	단위 시간당 가격	시작된 인스턴스의 수
`r3.xlarge`	2	0.05 USD	.025 (0.05를 2로 나눈 값)	5 (10을 2로 나눈 값)

인스턴스 유형	인스턴스 가중치	인스턴스 시간당 가격	단위 시간당 가격	시작된 인스턴스의 수
`r3.8xlarge`	8	0.10 USD	.0125 (0.10을 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 풀에서 프로비저닝합니다.

1. 스팟 집합에 대한 목표 용량을 인스턴스(기본값) 또는 선택한 단위(예: 가상 CPU 수, 메모리, 스토리지 또는 처리량)로 설정합니다.
2. 단위당 가격을 설정합니다.
3. 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 고려하십시오.

- 목표 용량은 24
- 인스턴스 유형이 `r3.2xlarge`이고 가중치가 6인 시작 사양
- 인스턴스 유형이 `c3.xlarge`이고 가중치가 5인 시작 사양

가중치는 목표 용량에 대하여 인스턴스 유형이 나타내는 단위 수를 의미합니다. 첫 번째 시작 사양에서 단위당 최저 가격(인스턴스 시간당 `r3.2xlarge`에 대한 가격을 6으로 나눈 값)을 제공하는 경우, 스팟 플릿은 이들 인스턴스 중 4개(24를 6으로 나눈 값)를 시작합니다.

두 번째 시작 사양에서 단위당 최저 가격(인스턴스 시간당 `c3.xlarge`에 대한 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 고려하십시오.

- 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 4개의 인스턴스(30을 8로 나눈 결과를 올림한 값)를 시작합니다. `lowestPrice` 전략 사용 시, 4개의 인스턴스는 전부 단위당 최저 가격을 제공하는 풀에서 가져옵니다. `diversified` 전략 사용 시 스팟 집합은 3개의 풀 각각에서 1개의 인스턴스를 시작하고 어떤 풀에 있는 것이든 4번째 인스턴스가 단위당 최저 가격을 제공합니다.

연습: 인스턴스 가중치를 부여한 스팟 집합 사용

이 연습에서는 Example Corp이라는 가상의 회사를 통해 인스턴스 가중치를 사용하여 스팟 집합을 요청하는 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 사례를 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 요건에 적합할 것이라는 결정을 내립니다.

인스턴스 유형	메모리(GiB)	vCPUs
r3.2xlarge	61	8
r3.4xlarge	122	16
r3.8xlarge	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 가격

Example Corp은 인스턴스 시간당 온디맨드 가격을 시작 가격으로 사용합니다. 그 밖에 최근 스팟 가격을 사용하거나, 둘을 조합할 수도 있습니다. 단위 시간당 가격을 계산하려면 인스턴스 시간당 시작 가격을 가중치로 나눕니다. 다음 예를 참조하십시오.

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 가격으로 0.7 USD를 사용하기 때문에 세 가지 인스턴스 유형 모두에서 경쟁력이 있습니다. 또한 `r3.8xlarge` 시작 사양에서 단위 시간당 글로벌 가격으로 0.7 USD를, 단위 시간당 특정 가격으로 0.9 USD를 사용할 수도 있습니다.

권한 검증

Example Corp은 스팟 집합 요청을 생성하기 전에 필요한 권한을 가진 IAM 역할이 있는지 검증합니다. 자세한 내용은 스팟 집합 사전 조건 (p. 336) 단원을 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/aws-ec2-spot-fleet-tagging-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. 334) 단원을 참조하십시오.

이행

할당 전략에서는 스팟 인스턴스가 어느 스팟 인스턴스 풀에서 온 것인지 확인합니다.

`lowestPrice` 전략(기본 전략) 사용 시, 스팟 인스턴스는 이행 시점에 단위당 최저 가격의 풀에서 온 것입니다. 20단위의 용량을 제공하기 위해 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.

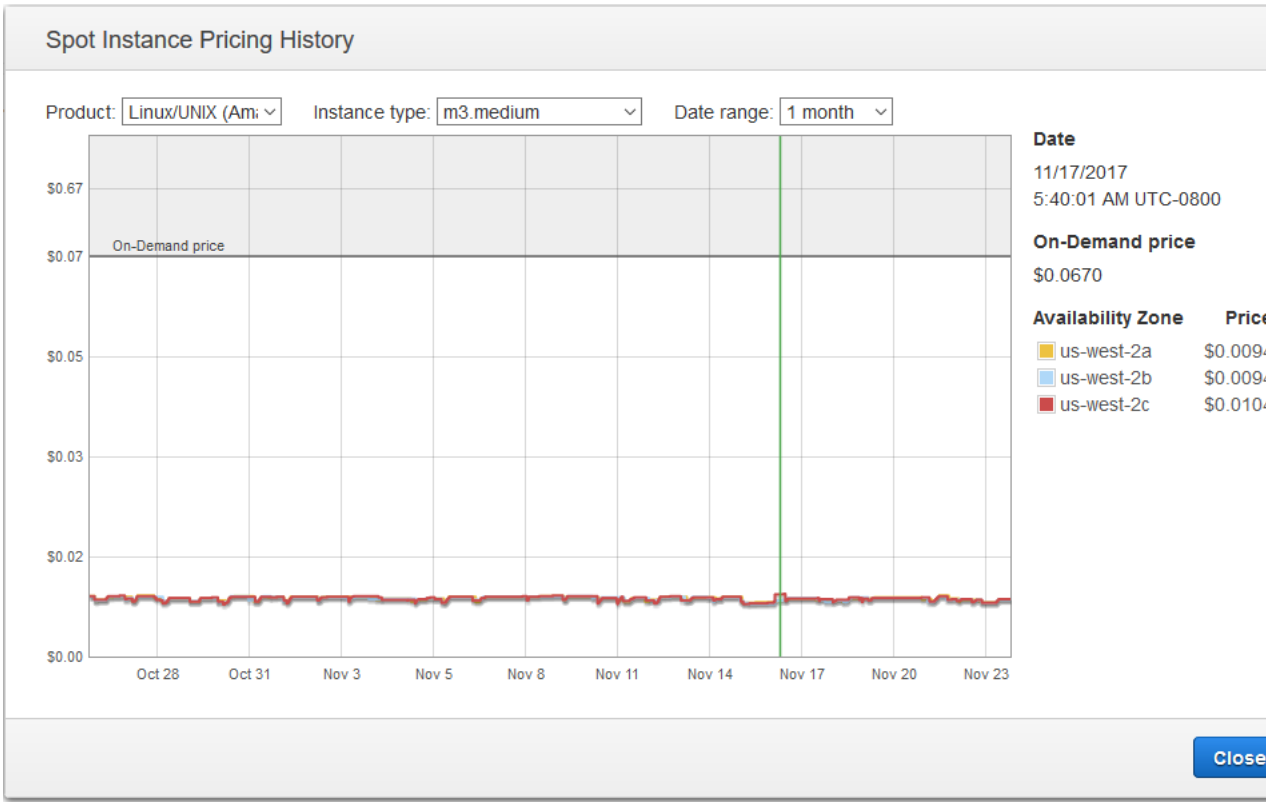
Example Corp에서 `diversified` 전략을 사용한 경우에는 스팟 인스턴스가 3개의 풀 전부에서 옵니다. 스팟 집합은 총 20개의 단위에 대해 `r3.2xlarge` 인스턴스 6개(6개 단위 제공), `r3.4xlarge` 인스턴스 3개(6개 단위 제공), `r3.8xlarge` 인스턴스 2개(8개 단위 제공)를 시작합니다.

스팟 인스턴스 요금 기록

스팟 인스턴스를 요청할 때는 기본 최고 가격(온디맨드 가격)을 사용하는 것이 좋습니다. 최고 가격을 지정하고 싶은 경우에는 설정에 앞서 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 보려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. 스팟 인스턴스를 처음 사용하는 경우 시작 페이지가 표시됩니다. 시작하기를 선택하고 화면 아래로 스크롤한 후 취소를 선택합니다.
4. 요금 기록을 선택합니다.
5. 가격 내역을 조회할 운영 체제(제품), 인스턴스 유형, 및 날짜 범위를 선택합니다. 마우스를 그래프 위에 놓으면 선택한 날짜 범위에서 특정 시간의 가격이 표시됩니다.



6. (선택 사항) 특정 가용 영역에 대한 스팟 가격 기록을 검토하려면 목록에서 영역을 선택합니다. 다른 제품, 인스턴스 유형 또는 날짜 범위도 선택할 수 있습니다.

명령줄을 사용하여 스팟 가격 기록을 보려면

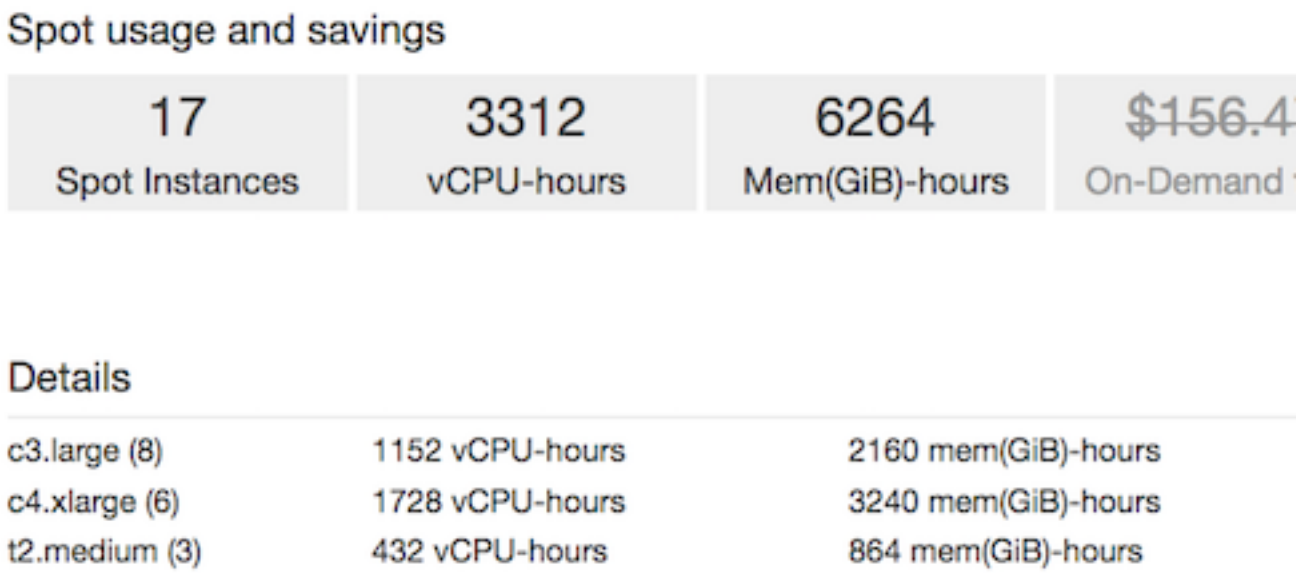
다음 명령 중 하나를 사용할 수 있습니다. 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- describe-spot-price-history(AWS CLI)
- Get-EC2SpotPriceHistory(Windows PowerShell용 AWS 도구)

스팟 인스턴스 구입으로 절감되는 비용

플릿별 수준에서 스팟 인스턴스에 대해 또는 실행 중인 모든 스팟 인스턴스에 대해 사용량 및 절감액 정보를 확인할 수 있습니다. 플릿별 수준에서 확인할 수 있는 사용량 및 절감액 정보에는 해당 플릿에서 시작 및 종료한 모든 인스턴스가 포함됩니다. 지난 1시간 또는 지난 3일에 대해 이러한 정보를 확인할 수 있습니다.

스팟 요청 페이지의 다음 스크린샷에는 스팟 집합에 대한 스팟 사용 및 비용 절감 정보가 표시됩니다.



다음 사용 및 비용 절감 정보를 볼 수 있습니다.

- 스팟 인스턴스 – 스팟 집합에서 시작 및 종료한 스팟 인스턴스 수. 비용 절감 요약을 볼 때 이 숫자는 실행 중인 모든 스팟 인스턴스를 나타냅니다.
- vCPU-hours(vCPU-시간) – 선택한 기간 중 모든 스팟 인스턴스에서 사용된 vCPU 시간 수
- Mem(GiB)-hours(메모리(GiB)-시간) – 선택한 기간 중 모든 스팟 인스턴스에서 사용된 GiB 시간 수.
- On-Demand total(온디맨드 합계) – 인스턴스를 온디맨드 인스턴스로 시작한 경우 선택한 기간 중 결제한 총 금액
- Spot total(스팟 합계) – 선택한 기간 중 결제한 총 금액.
- Savings(절감) – 온디맨드 가격을 결제하지 않아 절감한 비율.
- Average cost per vCPU-hour(vCPU-시간당 평균 비용) – 선택한 기간 중 모든 스팟 인스턴스에서 vCPU 사용의 시간당 평균 비용으로, 다음과 같이 계산합니다. vCPU-시간당 평균 비용 = 스팟 합계 / vCPU-시간.
- Average cost per mem(GiB)-hour(메모리(GiB)-시간당 평균 비용) – 선택한 기간 중 모든 스팟 인스턴스에서 GiB 사용의 시간당 평균 비용으로, 다음과 같이 계산합니다. 메모리(GiB)-시간당 평균 비용 = 스팟 합계 / 메모리(GiB)-시간.

- 세부 정보 표 – 스팟 집합를 구성하는 여러 인스턴스 유형. 인스턴스 유형당 인스턴스 수는 괄호 안에 표시되어 있습니다. 비용 절감 요약을 볼 때 이러한 숫자는 실행 중인 모든 스팟 인스턴스를 나타냅니다.

절감 정보는 Amazon EC2 콘솔에서만 확인할 수 있습니다.

스팟 집합에 대한 비용 절감 정보를 보려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. 스팟 집합 요청을 선택하고 절감을 선택합니다.
4. 기본적으로 이 페이지에는 지난 3일에 대한 사용 및 절감 정보가 표시됩니다. last hour(지난 시간) 또는 last three days(지난 3일)를 선택합니다. 시작한지 한 시간이 되지 않는 스팟 집합의 경우 페이지에는 한 시간에 대한 절감 추정치가 표시됩니다.

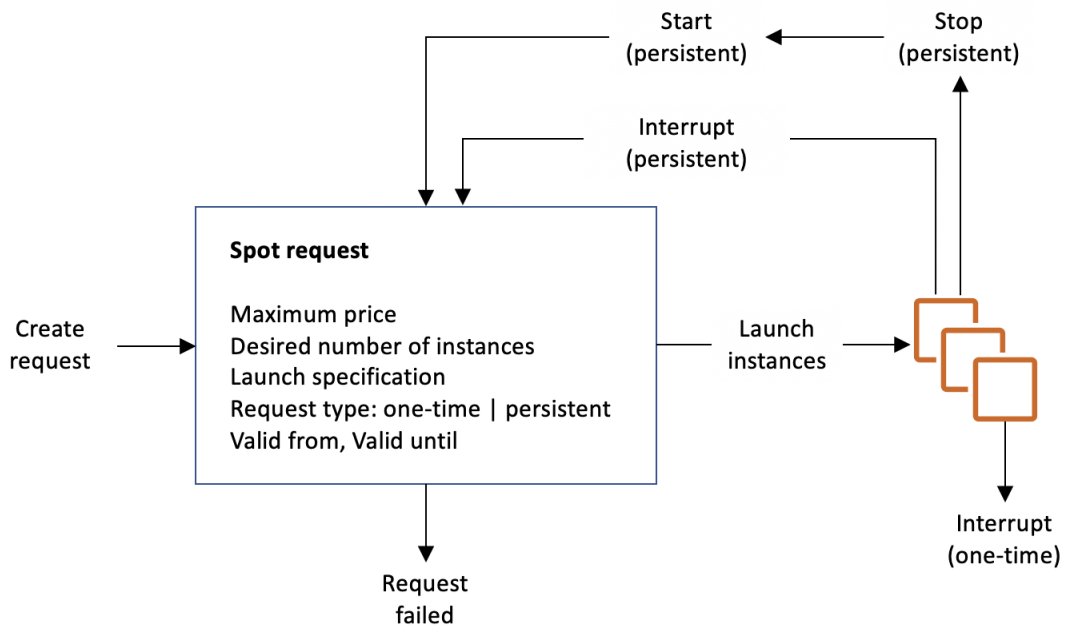
실행 중인 모든 스팟 인스턴스에 대한 비용 절감 정보를 보려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. Savings Summary(비용 절감 요약)를 선택합니다.

스팟 인스턴스 요청

스팟 인스턴스를 사용하려면 원하는 인스턴스 수, 인스턴스 유형, 가용 영역 및 인스턴스 시간당 지불하고자 하는 최고 가격이 포함된 스팟 인스턴스 요청을 생성합니다. 최고 가격이 현재 스팟 가격을 초과하는 경우 Amazon EC2는 용량이 가용 상태가 되는 즉시 요청을 이행합니다. 그렇지 않으면 요청이 이행될 수 있을 때까지 또는 사용자가 요청을 취소할 때까지 Amazon EC2가 대기합니다.

다음 그림에서는 스팟 요청이 작동하는 방식을 보여 줍니다. 요청 유형(일회성 또는 영구)에 따라 Amazon EC2가 스팟 인스턴스를 중단할 때 또는 사용자가 스팟 인스턴스를 중지하는 경우 요청이 다시 열리는지 여부가 결정됩니다. 요청이 영구적인 경우 스팟 인스턴스가 중단된 후 요청이 다시 열립니다. 요청이 영구적이고 사용자가 스팟 인스턴스를 중지하는 경우 스팟 인스턴스를 시작한 후에만 요청이 열립니다.



목차

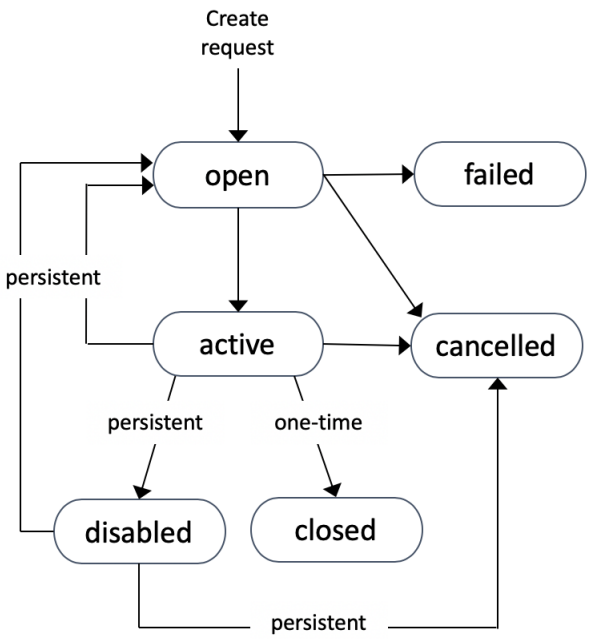
- 스팟 인스턴스 요청 상태 (p. 325)
- 스팟 인스턴스의 지속 시간 정의 (p. 326)
- 스팟 인스턴스의 테넌시 지정 (p. 327)
- 스팟 인스턴스 요청에 대한 서비스 연결 역할 (p. 328)
- 스팟 인스턴스 요청 생성 (p. 329)
- 실행 중인 스팟 인스턴스 찾기 (p. 330)
- 스팟 인스턴스 요청 태그 지정 (p. 330)
- 스팟 인스턴스 요청 취소 (p. 331)
- 스팟 인스턴스 중지 (p. 331)
- 스팟 인스턴스 시작 (p. 332)
- 스팟 인스턴스를 종료하는 중 (p. 332)
- 스팟 요청 예시 시작 사양 (p. 333)

스팟 인스턴스 요청 상태

스팟 인스턴스 요청은 다음 상태 중 하나일 수 있습니다.

- `open` – 요청이 이행될 때까지 대기 중입니다.
- `active` – 요청이 이행되며 연결된 스팟 인스턴스가 있습니다.
- `failed` – 요청에 하나 이상의 잘못된 파라미터가 있습니다.
- `closed` – 스팟 인스턴스가 중단되거나 종료되었습니다.
- `disabled` – 사용자가 스팟 인스턴스를 중지했습니다.
- `cancelled` – 사용자가 요청을 취소했거나 요청이 만료되었습니다.

다음 그림은 요청 상태 간의 전환을 나타냅니다. 전환은 요청 유형(일회 또는 영구)에 따라 다릅니다.



일회성 스팟 인스턴스 요청은 Amazon EC2가 스팟 인스턴스를 시작하거나, 요청이 만료되거나, 사용자가 요청을 취소할 때까지 활성 상태로 유지됩니다. 스팟 가격이 최고 가격을 초과하거나 용량이 가용 상태가 아니면 스팟 인스턴스가 종료되고 스팟 인스턴스 요청이 종결됩니다.

영구적 스팟 인스턴스 요청은 요청이 이행되더라도 요청이 만료되거나 사용자가 요청을 취소할 때까지 활성 상태로 유지됩니다. 스팟 가격이 최고 가격을 초과하거나 용량이 가용 상태가 아니면 스팟 인스턴스가 중단됩니다. 인스턴스가 중단된 후에 최고 가격이 스팟 가격을 초과하거나 용량이 다시 가용 상태가 되면 스팟 인스턴스가 중지된 경우에는 시작되거나, 최대 절전 모드인 경우에는 다시 시작됩니다. 용량을 사용할 수 있고 최고 가격이 현재 스팟 가격을 초과하는 경우 스팟 인스턴스를 중지하고 다시 시작할 수 있습니다. 스팟 인스턴스가 종료되는 경우(스팟 인스턴스가 중지 상태 또는 실행 중 상태인지 여부와 상관없이) 스팟 인스턴스 요청이 다시 열리고 Amazon EC2가 새 스팟 인스턴스를 시작합니다. 자세한 내용은 스팟 인스턴스 중지 (p. 331), 스팟 인스턴스 시작 (p. 332), 스팟 인스턴스를 종료하는 중 (p. 332) 단원을 참조하십시오.

이 상태를 통해 스팟 인스턴스 요청의 상태뿐 아니라 시작된 스팟 인스턴스의 상태도 추적할 수 있습니다. 자세한 내용은 스팟 요청 상태 (p. 366) 단원을 참조하십시오.

스팟 인스턴스의 지속 시간 정의

지속 시간(스팟 블록이라고도 함)이 정의된 스팟 인스턴스는 선택한 지속 시간 동안 중단되지 않고 계속 실행되도록 설계되었습니다. 이로 인해 배치성 프로세스, 인코딩 및 렌더링, 모델링 및 분석, 지속적 통합 작업처럼 완료하는 데 한정된 시간이 소요되는 작업에 이상적입니다.

1, 2, 3, 4, 5 또는 6시간의 지속 시간을 사용할 수 있습니다. 지불하는 요금은 지정된 지속 시간에 따라 변합니다. 1시간 또는 6시간의 지속 시간에 대한 현행 요금을 보려면 스팟 인스턴스 요금을 참조하십시오. 이 요금표를 이용해 2, 3, 4 및 5시간의 지속 시간에 대한 비용을 추산할 수 있습니다. 지속 시간으로 요청이 이행되면 스팟 인스턴스에 대한 가격이 고정되고 이 가격은 인스턴스가 종료될 때까지 유효합니다. 인스턴스를 실행하는 시간별로, 혹은 부분 시간에 대해 이 가격으로 요금이 청구됩니다. 부분 인스턴스 시간은 초 단위로 요금이 부과됩니다.

스팟 요청에서 지속 시간을 정의하면 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 종료를 표시하고 스팟 인스턴스 중단 공지를 통해 중지 2분 전에 이를 인스턴스에 경고합니다. 드문 경우 Amazon EC2 용량 때문에 스팟 블록이 중단될 수 있습

니다. 이 경우에는 인스턴스를 종료하기 2분 전에 경고를 보내며, 종료된 인스턴스를 사용한 경우에도 해당 인스턴스에 대해 요금이 부과되지 않습니다.

정의된 지속 시간으로 스팟 인스턴스를 시작하려면(콘솔)

자세한 내용은 스팟 집합 요청 생성 (p. 339) 단원을 참조하십시오.

정의된 지속 시간으로 스팟 인스턴스를 시작하려면(AWS CLI)

스팟 인스턴스에 대해 지속 시간을 지정하려면 request-spot-instances 명령으로 `--block-duration-minutes` 옵션을 포함시키십시오. 예를 들어 다음 명령을 통해 2시간 동안 실행되는 스팟 인스턴스를 시작하는 스팟 요청이 생성됩니다.

```
aws ec2 request-spot-instances --instance-count 5 --block-duration-minutes 120 --type "one-time" --launch-specification file://specification.json
```

지정된 지속 시간으로 스팟 인스턴스 비용을 검색하려면(AWS CLI)

describe-spot-instance-requests 명령을 사용하여 지정된 지속 시간으로 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. 410) 및 Amazon EC2 전용 인스턴스 제품 페이지를 참조하십시오.

전용 스팟 인스턴스를 실행하려면 다음 중 하나를 수행하십시오.

- 스팟 인스턴스 요청을 생성할 경우 `dedicated`의 테넌시를 지정합니다. 자세한 내용은 스팟 인스턴스 요청 생성 (p. 329) 단원을 참조하십시오.
- `dedicated`의 인스턴스 테넌시로 VPC에 스팟 인스턴스를 요청합니다. 자세한 내용은 전용 인스턴스 테넌시의 VPC 생성하기 (p. 412) 단원을 참조하십시오. VPC에서 `default`의 인스턴스 테넌시로 스팟 인스턴스를 요청한 경우 `dedicated`의 테넌시로 요청할 수 없습니다.

다음 인스턴스 유형에서는 전용 스팟 인스턴스를 지원합니다.

현재 세대

- `c4.8xlarge`
- `d2.8xlarge`
- `i3.16xlarge`
- `m4.10xlarge`
- `m4.16xlarge`
- `p2.16xlarge`
- `r4.16xlarge`
- `x1.32xlarge`

이전 세대

- `c3.8xlarge`
- `cc2.8xlarge`
- `cr1.8xlarge`

- `g2.8xlarge`
- `i2.8xlarge`
- `r3.8xlarge`

스팟 인스턴스 요청에 대한 서비스 연결 역할

Amazon EC2는 다른 AWS 서비스를 자동으로 호출하는 데 필요한 권한에 서비스 연결 역할을 사용합니다. 서비스 연결 역할은 AWS 서비스에 직접 연결된 고유한 유형의 IAM 역할입니다. 연결된 서비스만 서비스 연결 역할을 담당할 수 있으므로 서비스 연결 역할은 AWS 서비스로 권한을 위임하는 안전한 방법을 제공합니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 사용을 참조하십시오.

Amazon EC2는 AWSServiceRoleForEC2Spot이라는 이름의 서비스 연결 역할을 사용하여 사용자 대신 스팟 인스턴스를 시작하고 관리합니다.

AWSServiceRoleForEC2Spot에서 부여된 권한

Amazon EC2는 AWSServiceRoleForEC2Spot을 사용하여 다음 작업을 완료합니다.

- `ec2:DescribeInstances` – 스팟 인스턴스 설명
- `ec2:StopInstances` – 스팟 인스턴스 중지
- `ec2:StartInstances` – 스팟 인스턴스 시작

서비스 연결 역할 생성

대부분의 경우 서비스 연결 역할을 수동으로 생성할 필요가 없습니다. Amazon EC2는 처음으로 콘솔을 사용하여 스팟 인스턴스을 요청할 때 AWSServiceRoleForEC2Spot 서비스 연결 역할을 생성합니다.

Amazon EC2가 이 서비스 연결 역할을 지원하기 시작한 2017년 10월 이전에 활성 스팟 인스턴스를 요청한 경우 Amazon EC2에서 사용자 AWS 계정에 AWSServiceRoleForEC2Spot 역할을 이미 생성했습니다. 자세한 내용은 IAM 사용 설명서의 내 계정에 표시되는 새 역할을 참조하십시오.

AWS CLI 또는 API를 사용하여 스팟 인스턴스을 요청하려면 먼저 이 역할이 있어야 합니다. 역할을 생성하려면 다음과 같이 IAM 콘솔을 사용하십시오.

AWSServiceRoleForEC2Spot 서비스 연결 역할을 수동으로 생성하려면

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 역할을 선택합니다.
3. 역할 생성을 선택합니다.
4. 신뢰할 수 있는 유형의 엔터티 선택 페이지의 EC2에서 EC2 - 스팟 인스턴스를 선택한 후 다음: 권한을 선택합니다.
5. 다음 페이지에서 다음:검토(Next:Review)를 선택합니다.
6. 검토 페이지에서 역할 만들기를 선택합니다.

스팟 인스턴스이 더 이상 필요 없으면 AWSServiceRoleForEC2Spot 역할을 삭제하는 것이 좋습니다. 계정에서 이 역할이 삭제된 후 스팟 인스턴스를 요청하면 Amazon EC2에서 다시 해당 역할을 생성합니다.

암호화된 AMI 및 EBS 스냅샷과 함께 사용할 CMK에 대한 액세스 권한 부여

스팟 인스턴스 요청에서 암호화된 AMI (p. 145) 또는 암호화된 Amazon EBS 스냅샷 (p. 968)을 지정하고 암호화용 고객 관리형 고객 마스터 키(CMK)를 사용하는 경우 AWSServiceRoleForEC2Spot 역할에 CMK를 사용할 수 있는 권한을 부여해야 Amazon EC2가 사용자를 대신하여 스팟 인스턴스를 시작할 수 있습니다. 이렇게 하려면 다음 절차에 표시된 바와 같이 CMK에 권한을 추가해야 합니다.

권한을 제공할 때 권한 부여는 키 정책을 대체합니다. 자세한 내용은 AWS Key Management Service Developer Guide의 권한 부여 사용 및 AWS KMS에서 키 정책 사용 단원을 참조하십시오.

CMK를 사용할 수 있도록 AWSServiceRoleForEC2Spot 역할 권한을 부여하려면

- create-grant 명령을 사용하여 CMK에 권한을 추가하고 허용된 작업을 수행할 수 있는 권한이 부여된 보안 주체(AWSServiceRoleForEC2Spot 서비스 연결 역할)를 지정합니다. CMK는 CMK의 `key-id` 파라미터 및 ARN에 의해 지정됩니다. 보안 주체는 AWSServiceRoleForEC2Spot 서비스 연결 역할의 `grantee-principal` 파라미터 및 ARN에 의해 지정됩니다.

  다음은 읽기 쉬운 형식으로 지정된 예제입니다.

```
aws kms create-grant \
--region us-east-1 \
--key-id arn:aws:kms:us-east-1:444455556666:key/1234abcd-12ab-34cd-56ef-1234567890ab \
--grantee-principal arn:aws:iam::111122223333:role/AWSServiceRoleForEC2Spot \
--operations "Decrypt" "Encrypt" "GenerateDataKey" "GenerateDataKeyWithoutPlaintext"
 "CreateGrant" "DescribeKey" "ReEncryptFrom" "ReEncryptTo"
```

스팟 인스턴스 요청 생성

스팟 인스턴스를 요청하는 프로세스는 온디맨드 인스턴스를 시작하는 프로세스와 유사합니다. 요청을 제출한 후에는 최고 가격을 포함한 스팟 인스턴스 요청 파라미터를 변경할 수 없습니다.

여러 스팟 인스턴스를 한 번에 요청하는 경우 각 요청 상태를 개별적으로 추적할 수 있도록 Amazon EC2에서 개별 스팟 인스턴스 요청을 생성합니다. 스팟 인스턴스 요청 추적에 대한 자세한 내용은 스팟 요청 상태 (p. 366) 단원을 참조하십시오.

사전 조건

시작하기 전에 최고 가격, 원하는 스팟 인스턴스 수, 사용할 인스턴스 유형을 결정합니다. 스팟 가격 추세를 검토하려면 스팟 인스턴스 요금 기록 (p. 322) 단원을 참조하십시오.

스팟 인스턴스 요청을 생성하려면(콘솔)

자세한 내용은 스팟 집합 요청 생성 (p. 339) 단원을 참조하십시오.

스팟 인스턴스 요청을 생성하려면(AWS CLI)

다음 request-spot-instances 명령을 사용하여 일회성 요청을 생성합니다.

```
aws ec2 request-spot-instances --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 --instance-count 5 --type "persistent" --launch-
specification file://specification.json
```

이러한 명령과 함께 사용할 시작 사양 파일에 대한 예시는 스팟 요청 예시 시작 사양 (p. 333) 단원를 참조하십시오. 콘솔에서 시작 사양 파일을 다운로드하는 경우 대신 request-spot-fleet 명령을 사용해야 합니다(콘솔은 스팟 집합을 사용하여 스팟 요청을 지정함).

Amazon EC2는 최고 가격이 스팟 가격을 초과하고 용량이 가용 상태일 때 스팟 인스턴스를 시작합니다. 스팟 인스턴스는 중단되거나 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

Amazon EC2는 최고 가격이 스팟 가격을 초과하고 용량이 가용 상태일 때 스팟 인스턴스를 시작합니다. 스팟 인스턴스는 중단되거나 사용자가 직접 종료할 때까지 실행됩니다. 최고 가격이 스팟 가격과 정확히 같은 경우에는 수요에 따라 스팟 인스턴스가 계속 실행됩니다.

실행 중인 스팟 인스턴스를 찾으려면(콘솔)

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

   스팟 인스턴스 요청과 스팟 집합 요청을 모두 볼 수 있습니다. 스팟 인스턴스 요청이 이행된 경우 용량은 스팟 인스턴스의 ID입니다. 스팟 집합의 경우 용량은 요청된 용량 중 충족된 용량을 나타냅니다. 스팟 집합에서 인스턴스의 ID를 보려면 확장 화살표를 선택하거나 플릿을 선택한 후 인스턴스를 선택합니다.

   > **Note**
   >
   > 스팟 인스턴스 요청에 태그가 즉시 지정되지 않고 일정 기간 동안 SFR(스팟 집합 요청)과 별도로 나타날 수 있습니다.

3. 또는 탐색 창에서 인스턴스를 선택합니다. 오른쪽 위에서 표시/숨기기 아이콘을 선택한 다음 수명 주기를 선택합니다. 각 인스턴스에 대해 수명 주기는 `normal`, `spot` 또는 `scheduled`입니다.

실행 중인 스팟 인스턴스를 찾으려면(AWS CLI)

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"
    }
]
```

또는 다음과 같이 `--filters` 옵션으로 describe-instance 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. 자세한 내용은 Amazon EC2리소스에 태그 지정 (p. 1068) 단원을 참조하십시오.

요청을 만든 후 스팟 인스턴스 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

스팟 인스턴스 요청이 더 이상 필요 없는 경우 해당 요청을 취소할 수 있습니다. `open`, `active`, 또는 `disabled`인 스팟 인스턴스 요청만 취소할 수 있습니다.

- 요청이 아직 이행되지 않았고 인스턴스가 시작되지 않았으면 스팟 인스턴스 요청은 `open`입니다.
- 요청이 이행되었고 그 결과 스팟 인스턴스가 시작되었으면 스팟 인스턴스 요청은 `active`입니다.
- 스팟 인스턴스을 중지하면 스팟 인스턴스 요청은 `disabled`입니다.

스팟 인스턴스 요청이 `active`이고 연결된 실행 중인 스팟 인스턴스가 있거나, 스팟 인스턴스 요청이 `disabled`이고 연결된 중지된 스팟 인스턴스가 있는 경우, 요청을 취소해도 인스턴스가 종료되지 않습니다. 스팟 인스턴스 종료에 대한 자세한 내용은 스팟 인스턴스를 종료하는 중 (p. 332) 단원을 참조하십시오.

스팟 인스턴스 요청을 취소하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택한 다음 스팟 요청을 선택합니다.
3. 작업을 클릭하고 Cancel spot request(스팟 요청 취소)를 선택합니다.
4. (선택 사항) 연결된 스팟 인스턴스에 대한 작업을 완료했으면 종료할 수 있습니다. 탐색 창에서 인스턴스를 선택하고 인스턴스를 선택한 다음 작업, Instance State(인스턴스 상태), 중지를 차례로 선택합니다.

스팟 인스턴스 요청을 취소하려면(AWS CLI)

- 다음 cancel-spot-instance-requests 명령을 사용하여 지정한 스팟 요청을 취소하십시오.

```
aws ec2 cancel-spot-instance-requests --spot-instance-request-ids sir-08b93456
```

스팟 인스턴스 중지

지금 스팟 인스턴스가 필요하지 않지만 나중에 Amazon EBS 볼륨에 영구적으로 있는 데이터를 잃어버리지 않고 인스턴스를 다시 시작하려면 인스턴스를 중지할 수 있습니다. 스팟 인스턴스를 중지하는 단계는 온디맨드 인스턴스를 중지하는 단계와 비슷합니다. 스팟 인스턴스가 `persistent` 스팟 인스턴스 요청에서 시작된 경우에만 스팟 인스턴스를 중지할 수 있습니다.

> **Note**
>
> 스팟 인스턴스가 중지되었을 때 일부 인스턴스 속성을 수정할 수 있지만 인스턴스 유형은 수정할 수 없습니다.
> 중지된 스팟 인스턴스에 대한 사용 요금이나 데이터 전송 요금이 부과되지 않지만, 모든 Amazon EBS 볼륨에 대한 스토리지 요금이 부과됩니다.

제한 사항

- 플릿 또는 시작 그룹, 가용 영역 그룹, 스팟 블록의 일부인 경우 스팟 인스턴스를 중지할 수 없습니다.

스팟 인스턴스를 중지하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택하고 스팟 인스턴스를 선택합니다.
3. [Actions], [Instance State], [Stop]을 차례로 선택합니다.

스팟 인스턴스를 중지하려면(AWS CLI)

- 다음 stop-instances 명령을 사용하여 하나 이상의 스팟 인스턴스를 수동으로 중지합니다.

```
aws ec2 stop-instances --instance-ids i-1234567890abcdef0
```

스팟 인스턴스 시작

이전에 중지한 스팟 인스턴스를 시작할 수 있습니다. 스팟 인스턴스를 시작하는 단계는 온디맨드 인스턴스를 시작하는 단계와 비슷합니다.

사전 조건

다음과 같은 경우에만 스팟 인스턴스를 시작할 수 있습니다.

- 스팟 인스턴스를 수동으로 중지했습니다.
- 스팟 인스턴스가 EBS 지원 인스턴스입니다.
- 스팟 인스턴스 용량을 사용할 수 있습니다.
- 스팟 가격이 최고 가격보다 낮습니다.

제한 사항

- 플릿 또는 시작 그룹, 가용 영역 그룹, 스팟 블록의 일부인 경우 스팟 인스턴스를 시작할 수 없습니다.

스팟 인스턴스를 시작하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택하고 스팟 인스턴스를 선택합니다.
3. 작업, 인스턴스 상태, 시작을 차례로 선택합니다.

스팟 인스턴스를 시작하려면(AWS CLI)

- 다음 start-instance 명령을 사용하여 하나 이상의 스팟 인스턴스를 수동으로 시작합니다.

```
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

스팟 인스턴스를 종료하는 중

스팟 인스턴스 요청이 `active`이고 연결된 실행 중인 스팟 인스턴스가 있거나, 스팟 인스턴스 요청이 `disabled`이고 연결된 중지된 스팟 인스턴스가 있는 경우, 요청을 취소해도 인스턴스가 종료되지 않습니다. 실행 중인 스팟 인스턴스를 수동으로 종료해야 합니다. 영구 스팟 요청에서 시작된 실행 중이거나 중지된 스팟 인스턴스를 종료하면 새 스팟 인스턴스를 시작할 수 있도록 스팟 요청이 `open` 상태로 돌아갑니다. 영구 스팟 요청을 취소하고 스팟 인스턴스를 종료하려면 먼저 스팟 요청을 취소한 다음 스팟 인스턴스를 종료해야 합니다. 그렇지 않으면 영구 스팟 요청이 새로운 인스턴스를 시작할 수 있습니다. 스팟 인스턴스 요청 취소에 대한 자세한 내용은 스팟 인스턴스 요청 취소 (p. 331) 단원을 참조하십시오.

스팟 인스턴스 를 수동으로 종료하려면(AWS CLI)

- 다음 terminate-instances 명령을 사용하여 스팟 인스턴스를 수동으로 종료합니다.

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0 i-0598c7d356eba48d7
```

스팟 요청 예시 시작 사양

다음 예에서는 request-spot-fleet 명령과 함께 사용하여 스팟 인스턴스 요청을 생성할 수 있는 시작 구성을 보여줍니다. 자세한 내용은 스팟 인스턴스 요청 생성 (p. 329) 단원을 참조하십시오.

1. 스팟 인스턴스 시작 (p. 333)
2. 지정된 가용 영역에서 스팟 인스턴스 시작 (p. 333)
3. 지정된 서브넷에서 스팟 인스턴스 시작 (p. 333)
4. 전용 스팟 인스턴스 시작 (p. 334)

예 1: 스팟 인스턴스 시작

다음 예는 가용 영역 또는 서브넷을 포함하지 않습니다. Amazon EC2는 사용자를 위한 가용 영역을 선택합니다. Amazon EC2는 선택된 가용 영역의 기본 서브넷에 있는 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"
  }
}
```

예 2: 지정된 가용 영역에서 스팟 인스턴스 시작

다음 예는 가용 영역을 포함합니다. Amazon EC2는 지정된 가용 영역의 기본 서브넷에 있는 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 합니다.

```
{
  "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 예에서는 `dedicated`의 테넌시를 사용하여 스팟 인스턴스를 요청합니다. 전용 스팟 인스턴스는 VPC에서 시작해야 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 목표 용량, 옵션 온디맨드 부분, 인스턴스에 대한 하나 이상의 시작 사양, 지불하고자 하는 최고 가격을 포함한 스팟 집합 요청을 생성합니다. Amazon EC2가 스팟 가격 변경에 따라 스팟 집합의 목표 용량을 유지하도록 시도합니다. 자세한 내용은 스팟 집합 작동 방식 (p. 316) 단원을 참조하십시오.

스팟 집합 요청에는 두 가지 유형인 `request`와 `maintain`이 있습니다. 스팟 집합을 생성하여 원하는 용량을 위한 일회성 요청을 제출하거나, 일정 시간 동안 목표 용량을 유지하도록 요구할 수 있습니다. 두 가지 유형의 요청 모두 스팟 집합의 할당 전략에 따른 이익을 볼 수 있습니다.

일회성 요청을 생성하는 경우 스팟 집합에서는 필요한 요청을 수행하되, 용량이 줄어도 스팟 인스턴스를 보충하려고 하지는 않습니다. 용량이 가용 상태가 아닌 경우 스팟 집합은 대체 스팟 풀에서 요청을 제출하지 않습니다.

목표 용량 유지를 위해 스팟 집합은 이 목표 용량을 충족하도록 요청을 수행하고 중단된 모든 인스턴스를 자동으로 보충합니다.

일회성 요청이 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 종료되는지 여부를 지정할 수 있습니다.

각 시작 사양에는 AMI, 인스턴스 유형, 서브넷 또는 가용 영역, 하나 이상의 보안 그룹과 같이 Amazon EC2가 인스턴스를 시작하는 데 필요로 하는 정보가 포함됩니다.

내용

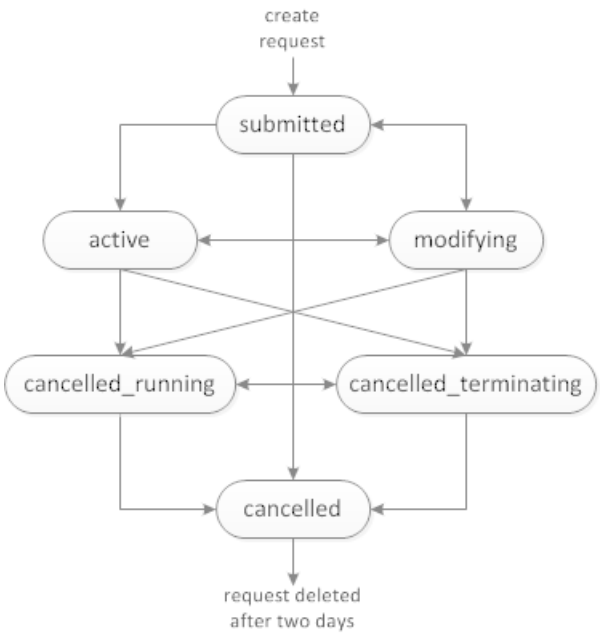
- 스팟 집합 요청 상태 (p. 335)
- 스팟 집합 사전 조건 (p. 336)
- 스팟 집합와 IAM 사용자 (p. 336)
- 스팟 집합 상태 확인 (p. 337)
- 스팟 집합 요청 계획 (p. 338)
- 스팟 집합 요청에 대한 서비스 연결 역할 (p. 338)
- 스팟 집합 요청 생성 (p. 339)
- 스팟 집합 태그 지정 (p. 343)
- 스팟 집합 모니터링 (p. 348)
- 스팟 집합 요청 수정 (p. 349)
- 스팟 집합 요청 취소 (p. 350)
- 스팟 집합 구성의 예 (p. 351)

스팟 집합 요청 상태

스팟 집합 요청은 다음 상태 중 하나일 수 있습니다.

- `submitted` – 스팟 집합 요청을 평가 중이며 Amazon EC2가 목표 개수만큼 인스턴스를 시작하기 위해 준비 중입니다.
- `active` – 스팟 집합 요청이 확인되었으며 Amazon EC2에서는 실행 중인 스팟 인스턴스를 목표 개수만큼 유지하려 하고 있습니다. 그 요청은 수정 또는 취소될 때까지 계속 이 상태로 유지됩니다.
- `modifying` – 스팟 집합 요청을 수정하고 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 인스턴스를 시작하지 않습니다. 중단되거나 종료될 때까지 기존 스팟 인스턴스가 계속 실행됩니다. 그 요청은 모든 인스턴스가 중단 또는 종료될 때까지 계속 이 상태로 유지됩니다.
- `cancelled_terminating` – 스팟 집합이 취소되고 해당 스팟 인스턴스가 종료 중입니다. 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 취소됩니다.



스팟 집합 사전 조건

Amazon EC2 콘솔을 사용하여 스팟 집합을 생성하면 사용자를 대신하여 인스턴스를 요청, 시작, 종료하고, 인스턴스에 태그를 지정할 수 있도록 스팟 집합 권한을 부여하는 `aws-ec2-spot-fleet-tagging-role`이라는 역할이 생성됩니다. 스팟 집합 요청을 생성할 때 이 역할이 선택됩니다. 그 대신 AWS CLI 또는 API를 사용할 경우 이 역할이 존재하는지 확인해야 합니다. 다음과 같이 스팟 인스턴스 요청 마법사(마법사의 두 번째 페이지로 넘어갈 때 역할이 생성됨)를 사용하거나 IAM 콘솔을 사용할 수 있습니다.

> **Important**
>
> 플릿에서 인스턴스에 태그를 지정하고 목표 용량을 유지하기로 선택하면(스팟 집합 요청은 `maintain` 유형) IAM 사용자와 `IamFleetRole`의 권한 차이로 인해 플릿에서 인스턴스의 태그 지정 동작이 일치하지 않을 수 있습니다. `IamFleetRole`에 `CreateTags` 권한이 없으면 플릿에서 시작한 일부 인스턴스에 태그가 지정되지 않을 수 있습니다. 이러한 불일치를 수정하는 동안 플릿에서 시작한 모든 인스턴스에 태그가 지정되도록 하려면 `IamFleetRole`에 `aws-ec2-spot-fleet-tagging-role` 역할을 사용하는 것이 좋습니다. 또는 기존 역할을 사용하려면 `AmazonEC2SpotFleetTaggingRole` AWS 관리형 정책을 기존 역할에 연결합니다. 그렇지 않으면 기존 정책에 `CreateTags` 권한을 수동으로 추가해야 합니다.

스팟 집합에 대한 IAM 역할 생성 방법

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 역할을 선택합니다.
3. 신뢰할 수 있는 유형의 엔터티 선택 페이지에서 AWS 서비스, EC2, EC2 - 스팟 플릿 태그 지정, 다음: 권한을 차례로 선택합니다.
4. 연결된 권한 정책(Attached permissions policy) 페이지에서 다음: 검토(Next: Review)를 선택합니다.
5. 검토 페이지에서 역할의 이름(예: **`aws-ec2-spot-fleet-tagging-role`**)을 입력하고 역할 생성을 선택합니다.

스팟 집합와 IAM 사용자

스팟 집합을 생성하거나 관리하는 IAM 사용자에게는 다음과 같이 필요한 권한을 부여해야 합니다.

스팟 집합의 IAM 사용자 권한을 부여하려면

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 정책을 선택한 후 정책 생성을 선택합니다.
3. 정책 생성 페이지에서 JSON을 선택한 다음, 텍스트를 다음과 같이 바꾸고 정책 검토를 선택합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
              "iam:ListRoles",
              "iam:PassRole",
              "iam:ListInstanceProfiles"
            ],
            "Resource": "*"
        }
    ]
}
```

`ec2:*`는 모든 Amazon EC2 API 작업을 호출할 수 있는 IAM 사용자 권한을 부여합니다. 사용자를 특정 Amazon EC2 API 작업으로 제한하려면 해당 작업을 대신 지정하십시오.

IAM 사용자에는 기존 IAM 역할을 열거하는 `iam:ListRoles` 작업, 스팟 집합 역할을 지정하는 `iam:PassRole` 작업 및 기존 인스턴스 프로파일을 열거하는 `iam:ListInstanceProfiles` 작업을 호출할 수 있는 권한이 있어야 합니다.

(선택 사항) IAM 사용자가 IAM 콘솔을 사용하여 역할 또는 인스턴스 프로파일을 생성할 수 있도록 하려면 정책에 다음 작업도 추가해야 합니다.

- `iam:AddRoleToInstanceProfile`
- `iam:AttachRolePolicy`
- `iam:CreateInstanceProfile`
- `iam:CreateRole`
- `iam:GetRole`
- `iam:ListPolicies`

4. 정책 검토 페이지에 정책 이름과 설명을 입력한 다음 정책 생성을 선택합니다.
5. 탐색 창에서 사용자를 선택하고 사용자를 선택합니다.
6. 권한을 선택하고, Add permissions(권한 추가)를 선택합니다.
7. Attach existing policies directly(기존 정책 직접 연결)를 선택합니다. 앞에서 만든 정책을 선택하고 다음: 검토(Next: Review)를 선택합니다.
8. 권한 추가를 선택합니다.

스팟 집합 상태 확인

스팟 집합은 2분마다 플릿에 있는 스팟 인스턴스의 상태를 확인합니다. 인스턴스의 상태는 `healthy` 또는 `unhealthy`입니다. 스팟 집합은 Amazon EC2에서 제공하는 상태 확인을 사용하여 인스턴스의 상태를 판단합니다. 세 번의 연속 상태 확인에서 인스턴스 상태 또는 시스템 상태가 `impaired`이면, 해당 인스턴

스의 상태는 `unhealthy`입니다. 그렇지 않으면 상태는 `healthy`입니다. 자세한 내용은 인스턴스 상태 확인 (p. 603) 단원을 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 목표 용량보다 적어질 수 있습니다.

요구 사항

- 상태 확인 교체는 목표 용량을 유지하는 스팟 집합에서만 지원되고 1회용 스팟 집합에서는 지원되지 않습니다.
- 비정상 인스턴스를 생성할 경우에만 이를 교체하도록 스팟 집합을 구성할 수 있습니다.
- IAM 사용자는 `ec2:DescribeInstanceStatus` 작업을 호출할 권한이 있는 경우에만 상태 확인 교체를 사용할 수 있습니다.

스팟 집합 요청 계획

스팟 집합 요청을 생성하려면 그 전에 먼저 스팟 모범 사례를 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 요청을 제출하는 스팟 집합을 생성할지, 아니면 시간 경과에 따라 목표 용량을 유지하는 스팟 플릿을 생성할지 여부를 결정합니다.
- 인스턴스 유형을 결정하고 애플리케이션 요건을 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. 319) 단원을 참조하십시오.
- 스팟 집합 목표 용량 중에서 온디맨드 용량이어야 하는 부분을 결정합니다. 온디맨드 용량으로 0을 지정할 수 있습니다.
- 인스턴스 가중치를 사용하는 경우에는 단위당 가격을 결정합니다. 단위당 가격을 계산하려면 인스턴스 시간당 가격을 이 인스턴스가 나타내는 단위 수(또는 가중치)로 나눕니다 인스턴스 가중치를 사용하지 않는 경우 단위당 기본 가격은 인스턴스 시간당 가격입니다.
- 스팟 집합 요청에 대해 가능한 옵션을 살펴봅니다. 자세한 내용은 AWS CLI Command Reference의 request-spot-fleet 명령을 참조하십시오. 추가 예제는 다음(스팟 집합 구성의 예 (p. 351))을 참조하십시오.

스팟 집합 요청에 대한 서비스 연결 역할

Amazon EC2는 다른 AWS 서비스를 자동으로 호출하는 데 필요한 권한에 서비스 연결 역할을 사용합니다. 서비스 연결 역할은 AWS 서비스에 직접 연결된 고유한 유형의 IAM 역할입니다. 연결된 서비스만 서비스 연결 역할을 담당할 수 있으므로 서비스 연결 역할은 AWS 서비스로 권한을 위임하는 안전한 방법을 제공합니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 사용을 참조하십시오.

Amazon EC2는 AWSServiceRoleForEC2Spot이라는 이름의 서비스 연결 역할을 사용하여 사용자 대신 스팟 인스턴스를 시작하고 관리합니다.

AWSServiceRoleForEC2SpotFleet에서 부여된 권한

Amazon EC2는 AWSServiceRoleForEC2SpotFleet을 사용하여 다음 작업을 완료합니다.

- `ec2:RequestSpotInstances` - 스팟 인스턴스 요청
- `ec2:TerminateInstances` - 스팟 인스턴스 종료
- `ec2:DescribeImages` - 스팟 인스턴스용 Amazon 머신 이미지(AMI) 설명
- `ec2:DescribeInstanceStatus` - 스팟 인스턴스의 상태 설명
- `ec2:DescribeSubnets` - 스팟 인스턴스용 서브넷 설명
- `ec2:CreateTags` - 스팟 인스턴스에 시스템 태그 추가

서비스 연결 역할 생성

대부분의 경우 서비스 연결 역할을 수동으로 생성할 필요가 없습니다. Amazon EC2는 처음으로 콘솔을 사용하여 스팟 집합을 생성할 때 AWSServiceRoleForEC2SpotFleet 서비스 연결 역할을 생성합니다.

Amazon EC2가 이 서비스 연결 역할을 지원하기 시작한 2017년 10월 이전에 활성 스팟 집합를 요청한 경우 Amazon EC2에서 사용자 AWS 계정에 AWSServiceRoleForEC2SpotFleet 역할을 이미 생성했습니다. 자세한 내용은 IAM 사용 설명서의 내 계정에 표시되는 새 역할을 참조하십시오.

AWS CLI 또는 API를 사용하여 스팟 집합을 만들려면 먼저 이 역할이 있어야 합니다. 역할을 생성하려면 다음과 같이 IAM 콘솔을 사용하십시오.

AWSServiceRoleForEC2SpotFleet 서비스 연결 역할을 수동으로 생성하려면

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 역할을 선택합니다.
3. 역할 생성을 선택합니다.
4. 신뢰할 수 있는 유형의 엔터티 선택 페이지의 EC2에서 EC2 - 스팟 플릿을 선택한 후 다음: 권한을 선택합니다.
5. 다음 페이지에서 다음:검토(Next:Review)를 선택합니다.
6. 검토 페이지에서 역할 만들기를 선택합니다.

스팟 집합이 더 이상 필요 없는 경우에는 AWSServiceRoleForEC2SpotFleet 역할을 삭제할 것을 권합니다. 계정에서 이 역할이 삭제된 후 스팟 집합를 요청하면 Amazon EC2에서 다시 해당 역할을 생성합니다.

암호화된 AMI 및 EBS 스냅샷과 함께 사용할 CMK에 대한 액세스 권한 부여

스팟 집합 요청에서 암호화된 AMI (p. 145) 또는 암호화된 Amazon EBS 스냅샷 (p. 968)을 지정하고 암호화용 고객 관리형 고객 마스터 키(CMK)를 사용하는 경우 AWSServiceRoleForEC2SpotFleet 역할에 CMK를 사용할 수 있는 권한을 부여해야 Amazon EC2가 사용자를 대신하여 스팟 인스턴스를 시작할 수 있습니다. 이렇게 하려면 다음 절차에 표시된 바와 같이 CMK에 권한을 추가해야 합니다.

권한을 제공할 때 권한 부여는 키 정책을 대체합니다. 자세한 내용은 AWS Key Management Service Developer Guide의 권한 부여 사용 및 AWS KMS에서 키 정책 사용 단원을 참조하십시오.

CMK를 사용할 수 있도록 AWSServiceRoleForEC2SpotFleet 역할 권한을 부여하려면

- create-grant 명령을 사용하여 CMK에 권한을 추가하고 허용된 작업을 수행할 수 있는 권한이 부여된 보안 주체(AWSServiceRoleForEC2SpotFleet 서비스 연결 역할)를 지정합니다. CMK는 CMK의 `key-id` 파라미터 및 ARN에 의해 지정됩니다. 보안 주체는 AWSServiceRoleForEC2SpotFleet 서비스 연결 역할의 `grantee-principal` 파라미터 및 ARN에 의해 지정됩니다.

  다음은 읽기 쉬운 형식으로 지정된 예제입니다.

```
aws kms create-grant \
    --region us-east-1 \
    --key-id arn:aws:kms:us-
east-1:444455556666:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/AWSServiceRoleForEC2SpotFleet \
    --operations "Decrypt" "Encrypt" "GenerateDataKey"
 "GenerateDataKeyWithoutPlaintext" "CreateGrant" "DescribeKey" "ReEncryptFrom"
 "ReEncryptTo"
```

스팟 집합 요청 생성

AWS Management 콘솔을 사용하면 필요한 애플리케이션 또는 작업과 최소 계산 사양을 선택해 스팟 집합 요청을 빠르게 생성할 수 있습니다. Amazon EC2에서는 필요에 가장 잘 맞는 플릿을 구성하고 스팟 모범 사

례를 따릅니다. 자세한 내용은 스팟 집합 요청을 빠르게 생성하려면(콘솔) (p. 340) 단원을 참조하십시오. 그렇지 않으면 어떠한 기본 설정이든 수정할 수 있습니다. 자세한 내용은 정의된 파라미터를 사용하여 스팟 집합 요청 생성(콘솔) (p. 340) 단원을 참조하십시오.

스팟 집합 요청을 빠르게 생성하려면(콘솔)

다음 단계에 따라 빠르게 스팟 집합 요청을 생성합니다.

권장되는 설정을 사용하여 스팟 집합 요청을 생성하려면(콘솔)

1. https://console.aws.amazon.com/ec2spot에서 스팟 콘솔을 엽니다.
2. 스팟을 처음 사용하는 경우 시작 페이지가 표시되면 시작하기를 선택합니다. 그렇지 않다면 스팟 인스턴스 요청을 선택합니다.
3. Tell us your application or task need(필요한 애플리케이션 또는 작업 알리기)에서 Flexible workloads(유연한 워크로드), Load balancing workloads(워크로드 로드 밸런싱), 빅 데이터 워크로드 또는 Defined duration workloads(정의된 기간 워크로드)를 선택합니다.
4. 인스턴스 구성에서 Minimum compute unit(최소 컴퓨팅 단위)에 대해 애플리케이션 또는 작업에 필요한 최소 하드웨어 사양(vCPU, 메모리, 및 스토리지)으로 as specs(사양으로) 또는 as an instance type(인스턴스 유형으로)을 선택합니다.
   - as specs(사양으로)에 대해 필요한 vCPU 수와 메모리 양을 지정합니다.
   - as an instance type(인스턴스 유형으로)에 대해 기본 인스턴스 유형을 수락하거나 Change instance type(인스턴스 유형 변경)을 선택하여 다른 인스턴스 유형을 선택합니다.
5. Tell us how much capacity you need(필요한 용량 알리기)에서 Total target capacity(총 목표 용량)에 대해 목표 용량으로 요청할 단위 수를 지정합니다. 인스턴스 또는 vCPU를 선택할 수 있습니다.
6. 선택한 애플리케이션 또는 작업을 바탕으로 권장되는 Fleet request settings(플릿 요청 설정)를 살펴보고 시작을 선택합니다.

정의된 파라미터를 사용하여 스팟 집합 요청 생성(콘솔)

정의하는 파라미터를 사용하여 스팟 집합을 생성할 수 있습니다.

정의된 파라미터를 사용하여 스팟 집합 요청을 생성하려면(콘솔)

1. https://console.aws.amazon.com/ec2spot에서 스팟 콘솔을 엽니다.
2. 스팟을 처음 사용하는 경우 시작 페이지가 표시되면 시작하기를 선택합니다. 그렇지 않다면 스팟 인스턴스 요청을 선택합니다.
3. Tell us your application or task need(필요한 애플리케이션 또는 작업 알리기)에서 Flexible workloads(유연한 워크로드), Load balancing workloads(워크로드 로드 밸런싱), 빅 데이터 워크로드 또는 Defined duration workloads(정의된 기간 워크로드)를 선택합니다.
4. 인스턴스 구성에 대해 다음을 수행합니다.
   a. (선택 사항) 시작 템플릿에서 시작 템플릿을 선택합니다. 시작 템플릿이 Amazon 머신 이미지(AMI)를 지정해야 합니다. 사용자가 시작 템플릿을 지정할 경우 스팟 집합을 사용하여 AMI를 재정의할 수 없기 때문입니다.

      Important

      Optional On-Demand portion(옵션 온디맨드 부분)을 지정할 경우 시작 템플릿을 선택해야 합니다.

   b. AMI에 대해서는 AWS가 제공하는 기본 AMI 중 하나를 선택하거나 AMI 검색을 선택하여 사용자 커뮤니티의 AMI, AWS Marketplace의 AMI 또는 자체 AMI를 사용합니다.
   c. Minimum compute unit(최소 컴퓨팅 단위)에 대해 애플리케이션 또는 작업에 필요한 최소 하드웨어 사양(vCPU, 메모리, 및 스토리지)으로 as specs(사양으로) 또는 as an instance type(인스턴스 유형으로)을 선택합니다.

- as specs(사양으로)에 대해 필요한 vCPU 수와 메모리 양을 지정합니다.
- as an instance type(인스턴스 유형으로)에 대해 기본 인스턴스 유형을 수락하거나 Change instance type(인스턴스 유형 변경)을 선택하여 다른 인스턴스 유형을 선택합니다.

d. (선택 사항) 네트워크에서 기존 VPC를 선택하거나 새로 생성합니다.

   [기존 VPC] VPC를 선택합니다.

   [새 VPC] Amazon VPC 콘솔로 이동하려면 새 VPC 생성을 선택합니다. 마친 후에 마법사로 돌아와 목록을 새로 고칩니다.

e. (선택 사항) 가용 영역의 경우 AWS에서 스팟 인스턴스에 대한 가용 영역을 자동으로 선택하도록 하거나 가용 영역을 하나 이상 지정합니다.

   가용 영역에 두 개 이상의 서브넷이 있는 경우 서브넷에서 알맞은 서브넷을 선택합니다. 서브넷을 추가하려면 새 서브넷 생성을 선택하여 Amazon VPC 콘솔로 이동합니다. 마친 후에 마법사로 돌아와 목록을 새로 고칩니다.

f. (선택 사항) 키 페어 이름에서 기존 키 페어를 선택하거나 새로 생성합니다.

   [기존 키 페어] 키 페어를 선택합니다.

   [새 키 페어] 새 키 페어 생성을 선택해 Amazon VPC 콘솔로 이동합니다. 마친 후에 마법사로 돌아와 목록을 새로 고칩니다.

5. (선택 사항) 추가 구성에서 다음을 수행합니다.

   a. (선택 사항) 스토리지를 추가하려면 인스턴스 유형에 따라 추가 인스턴스 스토어 볼륨이나 Amazon EBS 볼륨을 지정합니다.

   b. (선택 사항) Amazon EBS 최적화를 활성화하려면 EBS-optimized(EBS 최적화)에서 EBS 최적 인스턴스 시작을 선택합니다.

   c. (선택 사항) 인스턴스에 대해 임시 블록 수준 스토리지를 추가하려면 인스턴스 스토어에 대해 시작 시 연결을 선택합니다.

   d. (선택 사항) 기본적으로 인스턴스에 대해 기본 모니터링 기능이 활성화됩니다. 세부 모니터링을 활성화하려면 모니터링에 대해 CloudWatch 세부 모니터링 활성화를 선택합니다.

   e. (선택 사항) 비정상 인스턴스를 대체하려면 상태 검사(Health check)에서 비정상 인스턴스 교체(Replace unhealthy instances)를 선택합니다. 이 옵션을 활성화하려면 먼저 대상 용량 유지를 선택해야 합니다.

   f. (선택 사항) 전용 스팟 인스턴스를 실행하려면 테넌시에 대해 전용 - 전용 인스턴스로 실행을 선택합니다.

   g. (선택 사항) 보안 그룹에 대해 하나 이상의 보안 그룹을 선택하거나 보안 그룹을 생성합니다.

      [기존 보안 그룹] 하나 이상의 보안 그룹을 선택합니다.

      [새 보안 그룹] 새 보안 그룹 생성을 선택하여 Amazon VPC 콘솔로 이동합니다. 마친 후에 마법사로 돌아와 목록을 새로 고칩니다.

   h. (선택 사항) 인터넷에서 인스턴스에 연결할 수 있도록 하려면 IPv4 퍼블릭 IP 자동 할당에 대해 활성화를 선택합니다.

   i. (선택 사항) IAM 역할로 스팟 인스턴스를 시작하려면 IAM 인스턴스 프로파일에서 역할을 선택합니다.

   j. (선택 사항) 시작 스크립트를 실행하려면 해당 스크립트를 사용자 데이터에 복사합니다.

   k. (선택 사항) 태그를 추가하려면 새로운 태그 추가를 선택하고 해당 태그에 대한 키와 값을 입력합니다. 각 태그에 대해 반복합니다.

      각 태그에서 인스턴스와 스팟 집합 요청에 같은 태그를 사용하여 태그를 지정하려면 Instance tags(인스턴스 태그)와 Fleet tags(플릿 태그)가 모두 선택되어 있는지 확인합니다. 플릿에서 시작한 인스턴스에만 태그를 지정하려면 Fleet tags(플릿 태그) 선택을 취소합니다. 스팟 집합 요청에만 태그를 지정하려면 인스턴스 태그의 선택을 취소합니다.

6. Tell us how much capacity you need(필요한 용량 알리기)에 대해 다음을 수행합니다.

   a. Total target capacity(총 목표 용량)에 목표 용량으로 요청할 단위 수를 지정합니다. 인스턴스 또는 vCPU를 선택할 수 있습니다. 나중에 용량을 추가할 수 있도록 목표 용량을 0으로 지정하려면 Maintain target capacity(목표 용량 유지)를 선택합니다.

   b. (선택 사항) Optional On-Demand portion(옵션 온디맨드 부분)에 요청할 온디맨드 단위 수를 지정합니다. 이 값은 Total target capacity(총 목표 용량)보다 작아야 합니다. Amazon EC2가 차이를 계산하고 스팟 단위에 요청할 차이를 할당합니다.

      Important

      옵션 온디맨드 부분을 지정하려면 먼저, 시작 템플릿을 선택해야 합니다.

   c. (선택 사항) 스팟 서비스는 중단된 스팟 인스턴스를 종료하도록 기본 설정되어 있습니다. 목표 용량을 유지하려면 Maintain target capacity(목표 용량 유지)를 선택합니다. 그런 다음 스팟 서비스가 중단된 스팟 인스턴스를 종료, 중지 또는 최대 절전 모드로 전환하도록 지정할 수 있습니다. 이를 위해 인터럽트 방식에서 해당 옵션을 선택합니다.

7. Fleet request settings(플릿 요청 설정)에서 다음을 수행합니다.

   a. 선택한 애플리케이션 또는 작업을 바탕으로 플릿 요청 및 플릿 할당 전략을 살펴봅니다. 인스턴스 유형 또는 할당 전략을 변경하려면 Apply recommendations(권장 사항 적용)를 선택 취소합니다.

   b. (선택 사항) 인스턴스 유형을 제거하려면 Fleet request(플릿 요청)에 대해 제거를 선택합니다. 인스턴스 유형을 추가하려면 인스턴스 유형 선택을 선택합니다.

   c. (선택 사항) Fleet allocation strategy(플릿 할당 전략)으로 필요에 맞는 전략을 선택합니다. 자세한 내용은 스팟 인스턴스를 위한 할당 전략 (p. 316) 단원을 참조하십시오.

8. (선택 사항) Additional request details(추가 요청 세부 정보)에서 다음을 수행합니다.

   a. 추가 요청 세부 정보를 검토합니다. 변경하려면 Apply defaults(기본값 적용)의 선택을 취소합니다.

   b. (선택 사항) IAM fleet role(IAM 플릿 역할)에서 기본 역할을 사용하거나 다른 역할을 선택할 수 있습니다. 역할을 변경한 후 기본 역할을 사용하려면 기본 역할 사용을 선택합니다.

   c. (선택 사항) 최고 가격에서는 기본 최고 가격(온디맨드 가격)을 사용하거나 지불하고자 하는 최고 가격을 지정합니다. 최고 가격이 선택한 인스턴스 유형에 대한 스팟 가격보다 낮으면 스팟 인스턴스가 시작되지 않습니다.

   d. (선택 사항) 특정 기간 동안만 유효한 요청을 생성하려면 요청 유효 시작 시간 및 요청 유효 종료 시간(Request valid until)을 편집합니다.

   e. (선택 사항) 기본적으로 요청 만료 시 스팟 인스턴스를 종료합니다. 요청 만료 후에도 계속 실행하려면 Terminate the instances when the request expires(요청 만료 시 인스턴스 종료)를 선택 취소합니다.

   f. (선택 사항) 로드 밸런서에 스팟 인스턴스를 등록하려면 하나 이상의 로드 밸런서에서 트래픽 수신을 선택하고 하나 이상의 Classic Load Balancer나 대상 그룹을 선택합니다.

9. (선택 사항) AWS CLI용 시작 구성의 복사본을 다운로드하려면 JSON 구성을 선택합니다.

10. 시작을 선택합니다.

    스팟 집합 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 구성의 예 (p. 351) 단원을 참조하십시오.

다음은 예제 출력입니다.

```
{
    "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
}
```

스팟 집합 태그 지정

스팟 집합 요청을 쉽게 분류하고 관리할 수 있도록 사용자 지정 메타데이터로 이 요청에 태그를 지정할 수 있습니다. 스팟 집합 요청을 만들 때 또는 만든 후 요청에 태그를 지정할 수 있습니다. Amazon EC2 콘솔이나 명령줄 도구를 사용하여 태그를 지정할 수 있습니다.

스팟 집합 요청에 태그를 지정하면 스팟 집합에서 시작한 인스턴스에 자동으로 태그가 지정되지 않습니다. 스팟 집합에서 시작한 인스턴스에 명시적으로 태그를 지정해야 합니다. 스팟 집합 요청에만, 플릿에서 시작한 인스턴스에만 또는 둘 다에 태그를 지정하도록 선택할 수 있습니다.

태그 작동 방식에 대한 자세한 내용은 Amazon EC2리소스에 태그 지정 (p. 1068) 단원을 참조하십시오.

사전 조건

IAM 사용자에게 리소스에 태그를 지정할 수 있는 권한을 부여합니다. 자세한 내용은 예제: 리소스에 태그 지정 (p. 836) 단원을 참조하십시오.

IAM 사용자에게 리소스에 태그를 지정할 수 있는 권한을 부여하려면

다음을 포함하는 IAM 정책을 만듭니다.

- `ec2:CreateTags` 작업. IAM 사용자에게 태그 생성 권한이 부여됩니다.
- `ec2:RequestSpotFleet` 작업. IAM 사용자에게 스팟 집합 요청 생성 권한이 부여됩니다.
- `Resource`의 경우 `"*"`를 지정해야 합니다. 이를 통해 사용자는 모든 리소스 유형에 태그를 지정할 수 있습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "TagSpotFleetRequest",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags",
                "ec2:RequestSpotFleet"
            ],
            "Resource": "*"
}
```

> **Important**
>
> `spot-fleet-request` 리소스에 대한 리소스 수준 권한은 현재 지원되지 않습니다. `spot-fleet-request`를 리소스로 지정하면 플릿에 태그를 지정하려고 할 때 승인되지 않은 예외가 발생합니다. 다음 예에서는 정책을 설정하지 않는 방법을 보여 줍니다.

```
{
    "Effect": "Allow",
    "Action": [
        "ec2:CreateTags",
        "ec2:RequestSpotFleet"
    ],
```

```
    "Resource": "arn:aws:ec2:us-east-1:111122223333:spot-fleet-request/*"
}
```

콘솔을 사용하여 새 스팟 집합 요청에 태그를 지정하려면

1. 정의된 파라미터를 사용하여 스팟 집합 요청 생성(콘솔) (p. 340)의 절차를 따르십시오.
2. 태그를 추가하려면 추가 구성을 확장하고 새로운 태그 추가를 선택한 다음 태그의 키와 값을 입력합니다. 각 태그에 대해 반복합니다.

   각 태그에 대해 동일한 태그로 스팟 집합 요청과 인스턴스에 태그를 지정할 수 있습니다. 인스턴스와 요청에 모두 태그를 지정하려면 Instance tags(인스턴스 태그)와 Fleet tags(플릿 태그)가 모두 선택되어 있는지 확인합니다. 스팟 집합 요청에만 태그를 지정하려면 인스턴스 태그의 선택을 취소합니다. 플릿에서 시작한 인스턴스에만 태그를 지정하려면 Fleet tags(플릿 태그) 선택을 취소합니다.
3. 필수 필드를 입력하여 스팟 집합 요청을 생성한 다음 시작을 선택합니다. 자세한 내용은 정의된 파라미터를 사용하여 스팟 집합 요청 생성(콘솔) (p. 340) 단원을 참조하십시오.

AWS CLI를 사용하여 새 스팟 집합 요청에 태그를 지정하려면

스팟 집합 요청을 생성할 때 태그를 지정하려면 다음과 같이 스팟 집합 요청 구성을 구성합니다.

- `SpotFleetRequestConfig`에서 스팟 집합 요청에 대한 태그를 지정합니다.
- `ResourceType`에 대해 `spot-fleet-request`를 지정합니다. 다른 값을 지정하면 플릿 요청이 실패합니다.
- `Tags`에 대해 키-값 페어를 지정합니다. 둘 이상의 키-값 페어를 지정할 수 있습니다.

다음 예에서 스팟 집합 요청에는 Key=Environment 및 Value=Production, Key=Cost-Center 및 Value=123 두 개의 태그가 지정됩니다.

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "lowestPrice",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::111122223333:role/aws-ec2-spot-fleet-tagging-role",
        "LaunchSpecifications": [
            {
                "ImageId": "ami-0123456789EXAMPLE",
                "InstanceType": "c4.large"
            }
        ],
        "SpotPrice": "5",
        "TargetCapacity": 2,
        "TerminateInstancesWithExpiration": true,
        "Type": "maintain",
        "ReplaceUnhealthyInstances": true,
        "InstanceInterruptionBehavior": "terminate",
        "InstancePoolsToUseCount": 1,
        "TagSpecifications": [
            {
                "ResourceType": "spot-fleet-request",
                "Tags": [
                    {
                        "Key": "Environment",
                        "Value":"Production"
                    },
                    {
                        "Key": "Cost-Center",
                        "Value":"123"
                    }
                ]
```

```
            }
        ]
    }
}
```

새 스팟 집합 요청 및 AWS CLI를 사용하여 시작하는 인스턴스에 태그를 지정하려면

스팟 집합 요청을 생성할 때 태그를 지정하고 플릿에 의해 시작할 때 인스턴스에 태그를 지정하려면 스팟 집합 요청 구성을 다음과 같이 구성합니다.

스팟 집합 요청 태그:

- `SpotFleetRequestConfig`에서 스팟 집합 요청에 대한 태그를 지정합니다.
- `ResourceType`에 대해 `spot-fleet-request`를 지정합니다. 다른 값을 지정하면 플릿 요청이 실패합니다.
- `Tags`에 대해 키-값 페어를 지정합니다. 둘 이상의 키-값 페어를 지정할 수 있습니다.

인스턴스 태그:

- `LaunchSpecifications`의 인스턴스에 대한 태그를 지정합니다.
- `ResourceType`에 대해 `instance`를 지정합니다. 다른 값을 지정하면 플릿 요청이 실패합니다.
- `Tags`에 대해 키-값 페어를 지정합니다. 둘 이상의 키-값 페어를 지정할 수 있습니다.

  또는 스팟 집합 요청에서 참조되는 시작 템플릿 (p. 438)에서 인스턴스에 대한 태그를 지정할 수 있습니다.

다음 예에서 스팟 집합 요청에는 Key=Environment 및 Value=Production, Key=Cost-Center 및 Value=123 두 개의 태그가 지정됩니다. 플릿에서 시작한 인스턴스에는 스팟 집합 요청의 태그 중 하나와 동일한 Key=Cost-Center 및 Value=123 태그 하나만 지정됩니다.

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "lowestPrice",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::111122223333:role/aws-ec2-spot-fleet-tagging-role",
        "LaunchSpecifications": [
            {
                "ImageId": "ami-0123456789EXAMPLE",
                "InstanceType": "c4.large",
                "TagSpecifications": [
                    {
                        "ResourceType": "instance",
                        "Tags": [
                            {
                                "Key": "Cost-Center",
                                "Value": "123"
                            }
                        ]
                    }
                ]
            }
        ],
        "SpotPrice": "5",
        "TargetCapacity": 2,
        "TerminateInstancesWithExpiration": true,
        "Type": "maintain",
        "ReplaceUnhealthyInstances": true,
        "InstanceInterruptionBehavior": "terminate",
        "InstancePoolsToUseCount": 1,
```

```
        "TagSpecifications": [
            {
                "ResourceType": "spot-fleet-request",
                "Tags": [
                    {
                        "Key": "Environment",
                        "Value":"Production"
                    },
                    {
                        "Key": "Cost-Center",
                        "Value":"123"
                    }
                ]
            }
        ]
    }
}
```

AWS CLI를 사용하여 스팟 집합에서 시작한 인스턴스에 태그를 지정하려면

플릿에서 시작될 때 인스턴스에 태그를 지정하려면 스팟 집합 요청에서 참조되는 시작 템플릿 (p. 438)에서 태그를 지정하거나 다음과 같이 스팟 집합 요청 구성에서 태그를 지정할 수 있습니다.

- `LaunchSpecifications`의 인스턴스에 대한 태그를 지정합니다.
- `ResourceType`에 대해 `instance`를 지정합니다. 다른 값을 지정하면 플릿 요청이 실패합니다.
- `Tags`에 대해 키-값 페어를 지정합니다. 둘 이상의 키-값 페어를 지정할 수 있습니다.

다음 예에서는 플릿에 의해 시작되는 인스턴스에 Key=Cost-Center 및 Value=123 태그가 지정되어 있습니다.

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "lowestPrice",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::111122223333:role/aws-ec2-spot-fleet-tagging-role",
        "LaunchSpecifications": [
            {
                "ImageId": "ami-0123456789EXAMPLE",
                "InstanceType": "c4.large",
                "TagSpecifications": [
                    {
                        "ResourceType": "instance",
                        "Tags": [
                            {
                                "Key": "Cost-Center",
                                "Value": "123"
                            }
                        ]
                    }
                ]
            }
        ],
        "SpotPrice": "5",
        "TargetCapacity": 2,
        "TerminateInstancesWithExpiration": true,
        "Type": "maintain",
        "ReplaceUnhealthyInstances": true,
        "InstanceInterruptionBehavior": "terminate",
        "InstancePoolsToUseCount": 1
    }
}
```

AWS CLI를 사용하여 기존 스팟 집합 요청에 태그를 지정하려면

create-tags 명령을 사용해 기존 리소스에 태그를 지정할 수 있습니다. 다음 예에서, 기존 스팟 집합 요청은 Key=purpose 및 Value=test로 태그가 지정되어 있습니다.

```
aws ec2 create-tags \
    --resources sfr-11112222-3333-4444-5555-66666EXAMPLE \
    --tags Key=purpose,Value=test
```

콘솔을 사용하여 기존 스팟 집합 요청에 태그를 지정하려면

스팟 집합 요청을 생성한 후 콘솔을 사용하여 플릿 요청에 태그를 추가할 수 있습니다.

1. https://console.aws.amazon.com/ec2spot에서 스팟 콘솔을 엽니다.
2. 스팟 집합 요청을 선택합니다.
3. 태그 탭을 선택하고 태그 생성을 선택합니다.

콘솔을 사용하여 스팟 집합 요청 태그를 보려면

1. https://console.aws.amazon.com/ec2spot에서 스팟 콘솔을 엽니다.
2. 스팟 집합 요청을 선택하고 태그 탭을 선택합니다.

스팟 집합 요청 태그를 설명하려면

describe-tags 명령을 사용하여 지정된 리소스에 대한 태그를 표시합니다. 다음 예제에서는 지정된 스팟 집합 요청에 대한 태그를 설명합니다.

```
aws ec2 describe-tags \
    --filters "Name=resource-id,Values=sfr-11112222-3333-4444-5555-66666EXAMPLE"
```

```
{
    "Tags": [
        {
            "Key": "Environment",
            "ResourceId": "sfr-11112222-3333-4444-5555-66666EXAMPLE",
            "ResourceType": "spot-fleet-request",
            "Value": "Production"
        },
        {
            "Key": "Another key",
            "ResourceId": "sfr-11112222-3333-4444-5555-66666EXAMPLE",
            "ResourceType": "spot-fleet-request",
            "Value": "Another value"
        }
    ]
}
```

스팟 집합 요청을 설명하여 스팟 집합 요청의 태그를 볼 수도 있습니다.

describe-spot-fleet-requests 명령을 사용하여 지정된 스팟 집합 요청의 구성을 볼 수 있습니다. 여기에는 플릿 요청에 대해 지정된 태그가 모두 포함되어 있습니다.

```
aws ec2 describe-spot-fleet-requests \
    --spot-fleet-request-ids sfr-11112222-3333-4444-5555-66666EXAMPLE
```

```
{
```

```
    "SpotFleetRequestConfigs": [
        {
            "ActivityStatus": "fulfilled",
            "CreateTime": "2020-02-13T02:49:19.709Z",
            "SpotFleetRequestConfig": {
                "AllocationStrategy": "capacityOptimized",
                "OnDemandAllocationStrategy": "lowestPrice",
                "ExcessCapacityTerminationPolicy": "Default",
                "FulfilledCapacity": 2.0,
                "OnDemandFulfilledCapacity": 0.0,
                "IamFleetRole": "arn:aws:iam::111122223333:role/aws-ec2-spot-fleet-tagging-
role",
                "LaunchSpecifications": [
                    {
                        "ImageId": "ami-0123456789EXAMPLE",
                        "InstanceType": "c4.large"
                    }
                ],
                "TargetCapacity": 2,
                "OnDemandTargetCapacity": 0,
                "Type": "maintain",
                "ReplaceUnhealthyInstances": false,
                "InstanceInterruptionBehavior": "terminate"
            },
            "SpotFleetRequestId": "sfr-11112222-3333-4444-5555-66666EXAMPLE",
            "SpotFleetRequestState": "active",
            "Tags": [
                {
                    "Key": "Environment",
                    "Value": "Production"
                },
                {
                    "Key": "Another key",
                    "Value": "Another value"
                }
            ]
        }
    ]
}
```

스팟 집합 모니터링

스팟 집합는 최고 가격이 스팟 가격을 초과하고 용량이 가용 상태일 때 스팟 인스턴스를 시작합니다. 스팟 인스턴스는 중단되거나 사용자가 직접 종료할 때까지 실행됩니다.

스팟 집합을 모니터링하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. 스팟 집합 요청을 선택합니다. 구성 세부 정보를 보려면 설명을 선택합니다.
4. 스팟 집합에 대한 스팟 인스턴스를 나열하려면 인스턴스를 선택합니다.
5. 스팟 집합에 대한 기록을 보려면 기록을 선택합니다.

스팟 집합을 모니터링하려면(AWS CLI)

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-73fbd2ce-
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-73fbd2ce-
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 및 온디맨드 부분을 늘립니다.
- 목표 용량 및 온디맨드 부분을 줄입니다.

> **Note**
>
> 일회용 스팟 집합 요청은 수정할 수 없습니다. 스팟 집합 요청을 만들 때 대상 용량 유지를 선택한 경우에만 스팟 집합 요청을 수정할 수 있습니다.

목표 용량을 늘리면 스팟 집합에서 추가 스팟 인스턴스를 시작합니다. 온디맨드 부분을 늘리면 스팟 집합에서 추가 온디맨드 인스턴스를 시작합니다.

목표 용량을 늘리면 스팟 집합이 스팟 집합 요청에 대한 할당 전략에 따라 추가 스팟 인스턴스를 시작합니다. 할당 전략이 `lowestPrice`이면 스팟 집합이 스팟 집합 요청에 있는 최저 가격의 스팟 인스턴스 풀에서 인스턴스를 시작합니다. 할당 전략이 `diversified`인 경우 스팟 집합은 스팟 집합 요청의 풀 전체에 인스턴스를 배포합니다.

목표 용량을 줄이면 스팟 집합이 새 목표 용량을 초과하는 모든 열린 요청을 취소합니다. 플릿의 크기가 새 목표 용량에 도달할 때까지 스팟 집합이 스팟 인스턴스를 종료하도록 요청할 수 있습니다. 할당 전략이 `lowestPrice`인 경우 스팟 집합은 단위당 최고 가격이 있는 인스턴스를 종료합니다. 할당 전략이 `diversified`인 경우 스팟 집합은 풀 전체의 인스턴스를 종료합니다. 또는 스팟 집합이 플릿을 현재 크기로 유지하도록 요청할 수 있습니다. 그러나 중단되거나 수동으로 종료한 스팟 인스턴스는 교체하지 마십시오.

목표 용량이 줄어 스팟 집합이 인스턴스를 종료하면 해당 인스턴스는 스팟 인스턴스 중단 공지를 받습니다.

스팟 집합 요청을 수정하려면(콘솔)

1. https://console.aws.amazon.com/ec2spot/home/fleet에서 스팟 콘솔을 엽니다.
2. 스팟 집합 요청을 선택합니다.
3. 작업을 선택한 다음, Modify target capacity(목표 용량 수정)을 선택합니다.
4. 목표 용량 수정(Modify target capacity)에서 다음 작업을 수행하십시오.
   a. 새 목표 용량 및 온디맨드 부분을 입력합니다.
   b. (선택 사양) 목표 용량을 줄이지만 집합은 현재 크기로 유지하고자 한다면, 인스턴스 종료 선택을 취소합니다.
   c. 제출을 선택합니다.

AWS CLI를 사용하여 스팟 집합 요청을 수정하려면

다음 modify-spot-fleet-request 명령을 사용하여 지정된 스팟 집합 요청의 목표 용량을 업데이트합니다.

```
aws ec2 modify-spot-fleet-request --spot-fleet-request-id sfr-73fbd2ce-
aa30-494c-8788-1cee4EXAMPLE --target-capacity 20
```

다음과 같이 이전 명령을 수정하여 결과적으로 어떤 스팟 인스턴스도 종료하지 않고 지정된 스팟 집합의 목표 용량을 줄일 수 있습니다.

```
aws ec2 modify-spot-fleet-request --spot-fleet-request-id sfr-73fbd2ce-
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 시작되는 새 스팟 인스턴스가 없습니다. 스팟 집합이 스팟 인스턴스를 종료할지 여부를 지정해야 합니다. 인스턴스를 종료하면 스팟 집합 요청은 `cancelled_terminating` 상태가 됩니다. 인스턴스를 종료하지 않으면, 스팟 플릿 요청은 `cancelled_running` 상태가 되고 인스턴스는 중단되거나 사용자가 수동으로 종료하지 않는 한 계속 실행됩니다.

스팟 집합 요청을 취소하려면(콘솔)

1. https://console.aws.amazon.com/ec2spot/home/fleet에서 스팟 콘솔을 엽니다.
2. 스팟 집합 요청을 선택합니다.
3. 작업을 클릭하고 Cancel spot request(스팟 요청 취소)를 선택합니다.
4. 스팟 요청 취소(Cancel spot request)에서 스팟 집합을 취소하겠다는 것을 확인합니다. 집합을 현재 크기로 유지하려면 인스턴스 종료를 선택 취소합니다. 준비가 완료되면 확인을 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-73fbd2ce-
aa30-494c-8788-1cee4EXAMPLE --terminate-instances
```

다음은 예제 출력입니다.

```
{
    "SuccessfulFleetRequests": [
        {
            "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
            "CurrentSpotFleetRequestState": "cancelled_terminating",
            "PreviousSpotFleetRequestState": "active"
        }
    ],
    "UnsuccessfulFleetRequests": []
}
```

다음과 같이 이전 명령을 수정하여 인스턴스를 종료하지 않고 지정된 스팟 집합 요청을 취소할 수 있습니다.

```
aws ec2 cancel-spot-fleet-requests --spot-fleet-request-ids sfr-73fbd2ce-
aa30-494c-8788-1cee4EXAMPLE --no-terminate-instances
```

다음은 예제 출력입니다.

```
{
    "SuccessfulFleetRequests": [
        {
            "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
            "CurrentSpotFleetRequestState": "cancelled_running",
            "PreviousSpotFleetRequestState": "active"
```

```
        }
    ],
    "UnsuccessfulFleetRequests": []
}
```

스팟 집합 구성의 예

다음 예에서는 request-spot-fleet 명령과 함께 사용하여 스팟 집합 요청을 생성할 수 있는 시작 구성을 보여 줍니다. 자세한 내용은 스팟 집합 요청 생성 (p. 339) 단원을 참조하십시오.

> **Note**
>
> 스팟 집합의 경우 시작 사양에 네트워크 인터페이스 ID를 지정할 수 없습니다. 시작 사양에서 `NetworkInterfaceID` 파라미터를 생략하십시오.

1. 리전에서 최저 가격의 가용 영역 또는 서브넷을 사용하여 스팟 인스턴스 시작 (p. 351)
2. 지정된 목록에서 최저 가격의 가용 영역 또는 서브넷을 사용하여 스팟 인스턴스 시작 (p. 351)
3. 지정된 목록에서 최저 가격의 인스턴스 유형을 사용하여 스팟 인스턴스 시작 (p. 353)
4. 요청에 대한 가격 재정의 (p. 354)
5. 다각화된 할당 전략을 사용하여 스팟 집합 시작 (p. 355)
6. 인스턴스 가중치를 사용하여 스팟 집합 시작 (p. 357)
7. 온디맨드 용량으로 스팟 집합 시작 (p. 358)

예 1: 리전에서 최저 가격의 가용 영역 또는 서브넷을 사용하여 스팟 인스턴스 시작

다음 예는 가용 영역이나 서브넷이 없는 단일 시작 사양을 지정합니다. 스팟 집합은 기본 서브넷을 보유한 최저 가격의 가용 영역에 있는 인스턴스를 시작합니다. 지불하는 가격이 온디맨드 가격을 초과하지 않습니다.

```
{
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-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: 지정된 목록에서 최저 가격의 가용 영역 또는 서브넷을 사용하여 스팟 인스턴스 시작

다음 예는 가용 영역이나 서브넷은 다르지만 인스턴스 유형과 AMI는 같은 두 개의 시작 사양을 지정합니다.

가용 영역

스팟 집합은 지정한 최저 가격의 가용 영역에 있는 기본 서브넷에서 인스턴스를 시작합니다.

```
{
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-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 요청에서 동일한 가용 영역의 서로 다른 서브넷을 지정할 수 없습니다.

```
{
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-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 합니다.

```
  ...
      {
          "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

```
{
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-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

```
{
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-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 가격 재정의

기본 최고 가격인 온디맨드 가격을 사용하는 것이 좋습니다. 원할 경우 플릿 요청에 대한 최고 가격과 개별 시작 사양에 대한 최고 가격을 지정할 수 있습니다.

다음 예제에서는 플릿 요청에 대한 최고 가격과 세 가지 시작 사양 중 두 개에 대한 최고 가격을 지정합니다. 플릿 요청 대한 최고 가격은 최고 가격을 지정하지 않은 모든 시작 사양에 사용됩니다. 스팟 집합이 최저 가격으로 인스턴스 유형을 사용하여 인스턴스를 시작합니다.

가용 영역

```
{
  "SpotPrice": "1.00",
  "TargetCapacity": 30,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-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/aws-ec2-spot-fleet-tagging-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 인스턴스를 위한 할당 전략 (p. 316) 단원을 참조하십시오.

가용 영역

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 30,
  "AllocationStrategy": "diversified",
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-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/aws-ec2-spot-fleet-tagging-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"
        }
    ]
}
```

특정 가용 영역 중단 시 EC2 용량이 스팟 요청을 수행할 확률을 늘리는 가장 좋은 방법은, 영역 간에 분산하는 것입니다. 이 시나리오에서는 사용할 수 있는 각 가용 영역을 시작 사양에 포함합니다. 그리고 매번 같은 서브넷을 사용하는 대신 (각각 다른 영역에 매핑되는) 3개의 고유 서브넷을 사용합니다.

가용 영역

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 30,
  "AllocationStrategy": "diversified",
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "c4.2xlarge",
          "Placement": {
              "AvailabilityZone": "us-west-2a"
          }
      },
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "m3.2xlarge",
          "Placement": {
              "AvailabilityZone": "us-west-2b"
          }
      },
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "r3.2xlarge",
          "Placement": {
              "AvailabilityZone": "us-west-2c"
          }
      }
  ]
}
```

서브넷

```
{
    "SpotPrice": "0.70",
```

```
    "TargetCapacity": 30,
    "AllocationStrategy": "diversified",
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c4.2xlarge",
            "SubnetId": "subnet-1a2b3c4d"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "m3.2xlarge",
            "SubnetId": "subnet-2a2b3c4d"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "r3.2xlarge",
            "SubnetId": "subnet-3a2b3c4d"
        }
    ]
}
```

예 6: 인스턴스 가중치를 사용하여 스팟 집합 시작

다음 예제에서는 인스턴스 가중치를 사용하는데, 이는 곧 가격이 인스턴스 시간당이 아니라 단위 시간당 가격이라는 의미입니다. 각 시작 구성마다 다른 인스턴스 유형과 다른 가중치가 나열됩니다. 스팟 집합은 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 플릿의 크기가 목표 용량을 밑돌지는 않습니다.

`r3.2xlarge` 요청이 성공하면 스팟이 이들 인스턴스 중 4개를 프로비저닝합니다. 20을 6으로 나누면 총 3.33개의 인스턴스가 되는데, 이를 올림 처리하여 4개의 인스턴스가 됩니다.

`c3.xlarge` 요청이 성공하면 스팟이 이들 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. 319) 단원을 참조하십시오.

가용 영역

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-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/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "r3.2xlarge",
          "SubnetId": "subnet-1a2b3c4d",
          "WeightedCapacity": 6
      },
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "c3.xlarge",
          "SubnetId": "subnet-1a2b3c4d",
          "WeightedCapacity": 3
      }
  ]
}
```

예 7: 온디맨드 용량으로 스팟 집합 시작

항상 인스턴스 용량을 사용할 수 있도록 스팟 집합 요청에 온디맨드 용량에 대한 요청을 포함할 수 있습니다. 용량이 있는 경우 온디맨드 요청이 항상 이행됩니다. 용량이 있고 가용 상태일 경우 목표 용량의 잔고는 스팟으로 이행됩니다.

다음 예에서는 원하는 목표 용량을 10으로 지정합니다. 이 중 5는 온디맨드 용량이어야 합니다. 스팟 용량은 지정하지 않습니다. 스팟 용량은 목표 용량에서 온디맨드 용량을 뺀 나머지 용량으로 간주됩니다. Amazon EC2는 사용 가능한 Amazon EC2 용량이 있는 경우 5개의 용량 단위를 온디맨드로 시작하고 5개(10-5=5)의 용량 단위를 스팟으로 시작합니다.

자세한 내용은 스팟 집합의 온디맨드 (p. 316) 단원을 참조하십시오.

```
{
  "IamFleetRole": "arn:aws:iam::781603563322:role/aws-ec2-spot-fleet-tagging-role",
  "AllocationStrategy": "lowestPrice",
  "TargetCapacity": 10,
  "SpotPrice": null,
  "ValidFrom": "2018-04-04T15:58:13Z",
  "ValidUntil": "2019-04-04T15:58:13Z",
  "TerminateInstancesWithExpiration": true,
  "LaunchSpecifications": [],
  "Type": "maintain",
  "OnDemandTargetCapacity": 5,
  "LaunchTemplateConfigs": [
    {
      "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0dbb04d4a6cca5ad1",
        "Version": "2"
      },
      "Overrides": [
        {
          "InstanceType": "t2.medium",
          "WeightedCapacity": 1,
          "SubnetId": "subnet-d0dc51fb"
        }
      ]
    }
  ]
}
```

CloudWatch에 대한 스팟 집합 지표

Amazon EC2는 스팟 집합을 모니터링하는 데 사용할 수 있는 Amazon CloudWatch 지표를 제공합니다.

> **Important**
>
> 정확성을 보장하기 위해, 이 측정치를 사용할 때는 세부 모니터링을 활성화하는 것이 좋습니다. 자세한 내용은 인스턴스에 대한 세부 모니터링 활성화 또는 비활성화 (p. 616) 단원을 참조하십시오.

CloudWatch가 제공하는 Amazon EC2 측정치에 대한 자세한 내용은 CloudWatch를 사용해 인스턴스 모니터링하기 (p. 616) 단원을 참조하십시오.

스팟 집합 지표

`AWS/EC2Spot` 네임스페이스에는 다음과 같은 측정치와 플릿의 스팟 인스턴스에 대한 CloudWatch 지표가 포함되어 있습니다. 자세한 내용은 인스턴스 지표 (p. 618) 단원을 참조하십시오.

`AWS/EC2Spot` 네임스페이스에는 다음 지표가 포함되어 있습니다.

지표	설명
`AvailableInstancePoolsCount`	스팟 집합 요청에 지정된 스팟 인스턴스 풀. 단위: 개수
`BidsSubmittedForCapacity`	Amazon EC2가 스팟 집합 요청을 제출한 용량. 단위: 개수
`EligibleInstancePoolCount`	Amazon EC2에서 요청을 이행할 수 있는 경우 스팟 집합 요청에 지정된 스팟 인스턴스 풀입니다. Amazon EC2는 스팟 인스턴스에 대해 지불할 의향이 있는 최대 가격이 스팟 가격보다 낮거나 스팟 가격이 온디맨드 인스턴스의 가격보다 높은 경우 풀에서 요청을 이행하지 않습니다. 단위: 개수
`FulfilledCapacity`	Amazon EC2가 달성한 용량. 단위: 개수
`MaxPercentCapacityAllocation`	스팟 집합 요청에 지정된 모든 스팟 집합 풀에 걸친 `PercentCapacityAllocation`의 최대값. 단위: 백분율
`PendingCapacity`	`TargetCapacity`와 `FulfilledCapacity`의 차이점. 단위: 개수
`PercentCapacityAllocation`	지정된 차원의 스팟 인스턴스 풀에 할당된 용량. 모든 스팟 인스턴스 풀에 기록된 최대값을 얻으려면 `MaxPercentCapacityAllocation`을 사용하십시오. 단위: 백분율
`TargetCapacity`	스팟 집합 요청의 목표 용량. 단위: 개수
`TerminatingCapacity`	프로비저닝된 용량이 목표 용량보다 커서 종료되는 용량입니다.

지표	설명
	단위: 개수

수치 측정 단위가 `Count`(수)인 경우, 가장 유용한 통계는 `Average`(평균)입니다.

스팟 집합 차원

스팟 집합에 대한 데이터를 필터링하려면 다음 차원을 사용하십시오.

차원	설명
`AvailabilityZone`	가용 영역별로 데이터를 필터링합니다.
`FleetRequestId`	스팟 집합 요청별로 데이터를 필터링합니다.
`InstanceType`	인스턴스 유형별로 데이터를 필터링합니다.

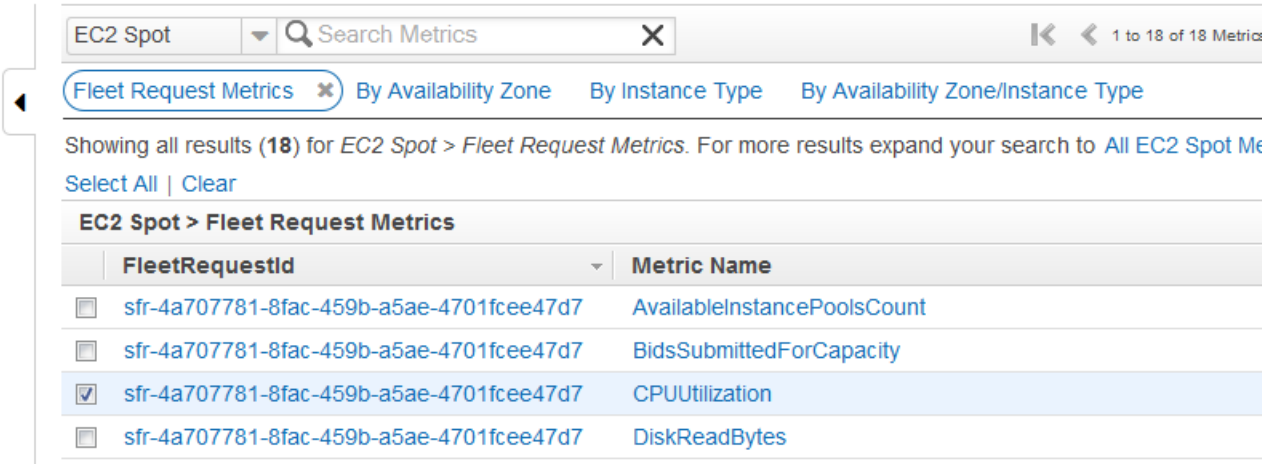
스팟 집합에 대한 CloudWatch 지표 보기

Amazon CloudWatch 콘솔을 사용하여 스팟 집합에 대한 CloudWatch 지표를 볼 수 있습니다. 이 측정치들은 모니터링 그래프로 표시됩니다. 이 그래프는 스팟 집합이 활성화되면 데이터 포인트를 표시합니다.

측정치는 먼저 네임스페이스별로 그룹화된 다음, 각 네임스페이스 내에서 다양한 차원 조합별로 그룹화됩니다. 예를 들어, 모든 스팟 집합 지표를 볼 수 있거나 스팟 집합 요청 ID, 인스턴스 유형 또는 가용 영역별로 그룹화된 스팟 집합 지표를 볼 수 있습니다.

스팟 집합 측정치를 보려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창의 지표에서 EC2 스팟 네임스페이스를 선택합니다.
3. (선택 사항) 측정치를 차원을 기준으로 필터링하려면 다음 중 하나를 선택하십시오.
   - 집합 요청 지표 - 스팟 집합 요청별 그룹.
   - 가용 영역별 - 스팟 집합 요청 및 가용 영역별 그룹.
   - 인스턴스 유형별 - 스팟 집합 요청 및 인스턴스 유형별 그룹.
   - 가용 영역/인스턴스 유형별 - 스팟 집합 요청, 가용 영역 및 인스턴스 유형별 그룹.
4. 측정치에 대한 데이터를 보려면 측정치 옆의 확인란을 선택합니다.



스팟 집합의 자동 조정

자동 조정은 수요에 따라 스팟 집합의 목표 용량을 자동으로 늘리거나 줄이는 기능입니다. 선택 범위 내에서 하나 이상의 조정 정책에 대한 응답으로 스팟 집합이 인스턴스 시작(스케일 아웃) 또는 인스턴스 종료(스케일 인)를 수행할 수 있습니다.

스팟 집합은 다음과 같은 자동 조정을 지원합니다.

- 대상 추적 조정 (p. 362) – 특정 측정치의 목표 값을 기준으로 플릿의 현재 용량을 늘리거나 줄입니다. 이 작업은 온도 조절기가 집안 온도를 유지하는 방식과 비슷합니다. — 사용자가 온도만 선택하면 나머지는 온도 조절기가 알아서 합니다.
- 단계 조정 (p. 363) – 일련의 조정 조절값(즉, 경보 위반의 크기에 따라 달라지는 단계 조절값)에 따라 플릿의 현재 용량을 늘리거나 줄입니다.
- 예약된 조정 (p. 365) – 날짜 및 시간을 기준으로 플릿의 현재 용량을 늘리거나 줄입니다.

인스턴스 가중치 (p. 319)를 사용하는 경우 스팟 집합에서 필요에 따라 목표 용량을 초과할 수 있다는 점에 주의하십시오. 이행된 용량이 부동 소수점 숫자일 수 있으나 목표 용량은 정수여야 하므로 스팟 집합은 결과를 다음 정수로 올립니다. 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 자동 조정 프로세스가 30.1에서 1을 빼 29.1을 도출한 후 30으로 올리므로 조정 작업이 수행되지 않습니다. 또 다른 예로, 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 가정합니다. 조정 정책이 목표 용량을 20% 줄이고 경보가 트리거되면 자동 조정 프로세스가 12에서 12*0.2를 빼 9.6을 도출한 후 10으로 올리므로 조정 작업이 수행되지 않습니다.

스팟 집합를 위해 생성한 조정 정책은 휴지 기간을 지원합니다. 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 자동 조정은 확장 가능한 대상을 즉시 확장합니다.

인스턴스 측정치를 1분 주기로 조정하는 것이 좋습니다. 이렇게 하면 사용량 변동에 따른 응답 속도를 더욱 높일 수 있기 때문입니다. 주기를 5분으로 하면 응답 시간이 느려질 뿐만 아니라 오랜 시간이 지난 측정치 데이터를 기준으로 조정하게 됩니다. 인스턴스에 대한 측정치 데이터를 CloudWatch에 1분 동안 전송하려면 특히 세부 모니터링을 활성화해야 합니다. 자세한 내용은 인스턴스에 대한 세부 모니터링 활성화 또는 비활성화 (p. 616) 및 정의된 파라미터를 사용하여 스팟 집합 요청 생성(콘솔) (p. 340) 단원을 참조하십시오.

스팟 집합 조정 구성에 대한 자세한 내용은 다음 리소스를 참조하십시오.

- AWS CLI Command Reference의 application-autoscaling 단원
- Application Auto Scaling API 참조
- Application Auto Scaling 사용 설명서

스팟 집합 Auto Scaling에 필요한 IAM 권한

Amazon EC2, Amazon CloudWatch, Application Auto Scaling API의 조합으로 스팟 집합에 대한 자동 조정이 가능합니다. Amazon EC2로 스팟 집합 요청이 생성되고, CloudWatch로 경보가 생성되고, Application Auto Scaling로 스케일링 정책이 생성됩니다.

스팟 집합에 대한 IAM권한 (p. 336) 및 Amazon EC2외에도 플릿 조정 설정에 액세스하는 IAM 사용자는 동적 조정을 지원하는 서비스에 대한 적절한 권한이 있어야 합니다. IAM 사용자는 다음 예제 정책에 표시된 조치를 사용할 수 있는 권한이 있어야 합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:*",
                "ec2:DescribeSpotFleetRequests",
                "ec2:ModifySpotFleetRequest",
                "cloudwatch:DeleteAlarms",
                "cloudwatch:DescribeAlarmHistory",
                "cloudwatch:DescribeAlarms",
                "cloudwatch:DescribeAlarmsForMetric",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics",
                "cloudwatch:PutMetricAlarm",
                "cloudwatch:DisableAlarmActions",
                "cloudwatch:EnableAlarmActions",
                "iam:CreateServiceLinkedRole",
                "sns:CreateTopic",
                "sns:Subscribe",
                "sns:Get*",
                "sns:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

Application Auto Scaling API 호출에 대한 보다 세분화된 권한을 허용하는 고유한 IAM 정책을 만들 수도 있습니다. 자세한 정보는 Application Auto Scaling 사용 설명서의 인증 및 액세스 제어를 참조하십시오.

Application Auto Scaling 서비스에는 스팟 집합 및 CloudWatch 경보를 설명할 수 있는 권한과 사용자를 대신하여 스팟 집합 대상 용량을 수정할 수 있는 권한이 필요합니다. 스팟 집합에 자동 조정을 활성화하면 `AWSServiceRoleForApplicationAutoScaling_EC2SpotFleetRequest`라는 서비스 연결 역할이 생성됩니다. 이 서비스 연결 역할은 정책에 대한 경보를 설명하고, 플릿의 현재 용량을 모니터링하고, 플릿의 용량을 수정할 수 있는Application Auto Scaling 권한을 부여합니다. Application Auto Scaling에 대한 관리되는 스팟 집합의 원래 역할은 `aws-ec2-spot-fleet-autoscale-role`이지만 더 이상 필요하지 않습니다. 서비스 연결 역할은 Application Auto Scaling의 기본 역할입니다. 자세한 내용은 Application Auto Scaling 사용 설명서의 서비스 연결 역할을 참조하십시오.

대상 추적 정책을 사용하여 스팟 집합 조정

대상 추적 조정 정책에 따라 측정치를 선택하고 대상 값을 설정합니다. 스팟 집합은 조정 정책을 트리거하는 CloudWatch 경보를 생성 및 관리하면서 측정치와 목표 값을 기준으로 조정 조절값을 계산합니다. 조정 정책은 필요에 따라 용량을 추가하거나 제거하여 측정치를 지정한 목표 값으로, 혹은 목표 값에 가깝게 유지합니다. 대상 추적 조정 정책은 측정치를 목표 값에 가깝게 유지하는 것 외에도 로드 패턴의 변화로 인한 측정치 변동에 따라 반응하여 플릿의 용량이 갑작스럽게 바뀌는 것을 최소화합니다.

각각 다른 측정치를 사용한다는 전제 하에 스팟 집합에 대해 다수의 대상 추적 조정 정책을 생성할 수 있습니다. 스팟 플릿은 최대 플릿 용량을 제공하는 정책에 따라 조정됩니다. 따라서 다양한 시나리오를 포괄하고 애플리케이션 워크로드를 처리하기에 충분한 용량을 항상 확보할 수 있습니다.

애플리케이션 가용성을 보장하기 위해 스팟 플릿은 측정치에 비례하여 가능한 신속하게 확장되지만, 축소는 점진적으로 이루어집니다.

목표 용량이 줄어 스팟 집합이 인스턴스를 종료하면 해당 인스턴스는 스팟 인스턴스 중단 공지를 받습니다.

스팟 집합이 대상 추적 조정 정책에서 관리하는 CloudWatch 경보는 편집하거나 삭제하지 마십시오. 대상 추적 조정 정책을 삭제하면 스팟 집합에서 경보가 자동으로 삭제됩니다.

제한

- 스팟 집합 요청에 `maintain` 요청 유형이 있어야 합니다. 1회 요청 또는 스팟 블록에는 자동 조정이 지원되지 않습니다.

대상 추적 조정 정책을 구성하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. 스팟 집합 요청을 선택한 후 Auto Scaling을 선택합니다.
4. 자동 조정이 구성되어 있지 않으면 구성을 선택합니다.
5. 다음 사이로 용량 조정(Scale capacity between)을 사용하여 플릿에 대한 최소 및 최대 용량을 설정합니다. 자동 조정에서 최소 용량 미만이거나 최대 용량을 초과하는 플릿을 조정하지 않습니다.
6. 정책 이름에서 정책의 이름을 입력합니다.
7. 대상 지표를 선택합니다.
8. 측정치에 대한 대상 값을 입력합니다.
9. (선택 사항) 기본 휴지 기간을 수정하려면 휴지 기간(Cooldown period)을 설정합니다.
10. (선택 사항) 현재 구성에 따라 축소 정책 생성을 생략하려면 축소 비활성화(Disable scale-in)을 선택합니다. 다른 구성을 사용하여 축소 정책을 생성할 수 있습니다.
11. 저장을 선택합니다.

AWS CLI를 사용하여 대상 추적 정책을 구성하려면

1. register-scalable-target 명령을 사용하여 스팟 집합 요청을 확장 가능 대상으로 등록합니다.
2. put-scaling-policy 명령을 사용하여 조정 정책을 생성합니다.

단계 조정 정책을 사용하여 스팟 집합 조정

단계 조정 정책을 사용하여 조정 프로세스를 트리거하도록 CloudWatch 경보를 지정합니다. 예를 들어, CPU 사용률이 특정 레벨에 도달하면 확장하려는 경우 Amazon EC2에서 제공하는 `CPUUtilization` 측정치를 사용하여 경보를 생성합니다.

단계 조정 정책을 생성할 때 다음과 같은 조정 조절 유형 중 하나를 지정해야 합니다.

- 추가 – 지정된 수의 용량 유닛 또는 지정된 현재 용량의 퍼센트까지 플릿의 대상 용량을 늘립니다.
- 제거 – 지정된 수의 용량 단위 또는 지정된 현재 용량의 퍼센트까지 플릿의 대상 용량을 줄입니다.
- 설정 – 플릿의 대상 용량을 지정된 수의 용량 단위로 설정합니다.

경보가 트리거되면 자동 조정 프로세스가 이행된 용량과 조정 정책을 사용하여 새로운 목표 용량을 계산한 후 그에 따라 목표 용량을 업데이트합니다. 예를 들어 목표 용량과 이행된 용량이 10이고 조정 정책이 1을 추가한다고 가정하십시오. 경보가 트리거되면 자동 조정 프로세스가 10에 1을 더해 11이 되므로 스팟 집합이 1 인스턴스를 시작합니다.

목표 용량이 줄어 스팟 집합이 인스턴스를 종료하면 해당 인스턴스는 스팟 인스턴스 중단 공지를 받습니다.

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.

CloudWatch 경보를 생성하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Alarms를 선택합니다.
3. 경보 생성을 선택합니다.
4. [Specify metric and conditions] 페이지에서 Select metric을 선택합니다.
5. EC2 Spot, Fleet Request Metrics을 선택하고, CPUUtilization과 같은 메트릭을 선택하고 Select metric을 선택합니다.

   선택한 지표에 대한 그래프와 기타 정보가 표시된 Specify metric and conditions(지표 및 조건 지정) 페이지가 나타납니다.
6. 기간에 대해 예를 들어 1분과 같은 경보에 대한 평가 기간을 선택합니다. 경보를 평가할 때 각 기간이 하나의 데이터 포인트로 집계됩니다.

   Note

   기간이 짧을수록 경보가 더 민감해집니다.
7. 조건에서 임계 조건을 정의하여 경보를 정의합니다. 예를 들어, 지표 값이 80% 이상일 때마다 경보를 트리거하는 임계값을 정의할 수 있습니다.
8. 추가 구성에서 경고할 데이터포인트에 대해 알람을 트리거하는 ALARM 상태에 있어야 하는 데이터포인트(평가 기간)의 수를 지정합니다. 예를 들어 1개의 평가 기간 또는 3개의 평가 기간 중 2개입니다. 그러면 다수의 연속 기간이 위반되면 ALARM 상태가 되는 경보가 생성됩니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 경보 평가를 참조하십시오.
9. Missing data treatment(누락된 데이터 처리)에서 옵션 중 하나를 선택합니다(또는 기본값인 Treat missing data as missing(누락된 데이터를 누락으로 처리)를 그대로 사용). 자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 경보가 누락된 데이터를 처리하는 방법 구성 단원을 참조하십시오.
10. [Next]를 선택합니다.
11. (선택 사항) 조정 이벤트 알림을 수신하려면 알림에 대해 알림을 받는 Amazon SNS 주제를 선택하거나 작성할 수 있습니다. 또는 지금 알림을 삭제하고 필요에 따라 나중에 추가할 수 있습니다.
12. [Next]를 선택합니다.
13. 설명 추가에서 경보의 이름과 설명을 입력하고 다음을 선택하십시오.
14. 경보 생성을 선택합니다.

스팟 집합에 대한 단계 조정 정책을 구성하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. 스팟 집합 요청을 선택한 후 Auto Scaling을 선택합니다.
4. 자동 조정이 구성되어 있지 않으면 구성을 선택합니다.
5. 다음 사이로 용량 조정(Scale capacity between)을 사용하여 플릿에 대한 최소 및 최대 용량을 설정합니다. 자동 조정에서 최소 용량 미만이거나 최대 용량을 초과하는 플릿을 조정하지 않습니다.
6. 처음에 조정 정책(Scaling policies)에 ScaleUp 및 ScaleDown이라는 정책이 포함됩니다. 이러한 정책을 완료하거나 정책 제거(Remove policy)를 선택하여 삭제할 수 있습니다. 또한 정책 추가를 선택하여 정책을 추가할 수도 있습니다.

7. 정책을 정의하려면 다음을 수행합니다.
   a. 정책 이름에서 정책의 이름을 입력합니다.
   b. 정책 트리거(Policy trigger)에서 기존 경보를 선택하거나 Create new alarm(새 경보 생성)을 선택하여 Amazon CloudWatch 콘솔을 열고 경보를 생성합니다.
   c. 용량 수정(Modify capacity)에서 조정 조절 유형을 선택하고, 숫자를 선택한 후 단위를 선택합니다.
   d. (선택 사항) 단계 조정을 수행하려면 단계 정의(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 단계 추가(Add step)를 선택합니다.
   e. (선택 사항) 휴지 기간의 기본값을 수정하려면 휴지 기간(Cooldown period)에서 숫자를 선택합니다.
8. 저장을 선택합니다.

AWS CLI를 사용하여 스팟 집합에 대한 단계 조정 정책을 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 트리거하는 경보를 생성합니다.

예약된 조정을 사용하여 스팟 집합 조정

일정을 기반으로 조정을 수행하면 수요에 따른 로드 변경에 맞게 애플리케이션을 조정할 수 있습니다. 예약된 조정을 사용하려면 스팟 집합이 특정 시간에 조정 작업을 수행하도록 하는 예약된 작업을 생성할 수 있습니다. 예약된 작업을 생성할 때 스팟 집합, 규모 조정 활동이 발생해야 할 시점, 최소 용량 및 최대 용량을 지정할 수 있습니다. 규모를 한 번만 조정하거나 반복되는 일정으로 조정하도록 예약된 작업을 생성할 수 있습니다.

제한

- 스팟 집합 요청에 `maintain` 요청 유형이 있어야 합니다. 1회 요청 또는 스팟 블록에는 자동 조정이 지원되지 않습니다.

1회성 예약된 작업을 만들려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. 스팟 집합 요청을 선택한 후 예약 조정을 선택합니다.
4. 예약 작업 만들기(Create Scheduled Action)를 선택합니다.
5. 이름에서 예약된 작업의 이름을 지정합니다.
6. 최소 용량, 최대 용량 또는 두 가지 모두 입력합니다.
7. 반복에서 1회(Once)를 선택합니다.
8. (선택 사항) 시작 시간, 종료 시간 또는 두 가지 모두에 대해 날짜와 시간을 선택합니다.
9. 제출을 선택합니다.

반복되는 일정으로 조정하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. 스팟 집합 요청을 선택한 후 예약 조정을 선택합니다.

4. 반복에서 사전 정의된 일정(예: 매일(Every day)) 중 하나를 선택하거나 사용자 지정을 선택하고 Cron 표현식을 입력합니다. 예약된 조정에서 지원하는 Cron 표현식에 대한 자세한 내용은 Amazon CloudWatch Events 사용 설명서의 Cron 표현식 단원을 참조하십시오.
5. (선택 사항) 시작 시간, 종료 시간 또는 두 가지 모두에 대해 날짜와 시간을 선택합니다.
6. 제출을 선택합니다.

예약된 작업을 편집하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. 스팟 집합 요청을 선택한 후 예약 조정을 선택합니다.
4. 예약된 작업을 선택한 다음, 작업, 편집을 선택합니다.
5. 필요한 변경을 수행하고 제출을 선택합니다.

예약된 작업을 삭제하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. 스팟 집합 요청을 선택한 후 예약 조정을 선택합니다.
4. 예약된 작업을 선택한 다음, 작업, 삭제를 선택합니다.
5. 확인 메시지가 나타나면 Delete를 선택합니다.

AWS CLI를 사용하여 예약된 조정을 관리하려면

다음 명령을 사용합니다.

- put-scheduled-action
- describe-scheduled-actions
- delete-scheduled-action

스팟 요청 상태

스팟 인스턴스 요청을 추적하고 스팟 인스턴스 사용 계획을 세우는 데 도움이 되도록 Amazon EC2에서 제공하는 요청 상태를 사용합니다. 예를 들어, 요청 상태는 스팟 요청이 아직 이행되지 않는 이유를 알려주거나, 스팟 요청을 이행할 수 없는 제약 조건을 나열할 수 있습니다.

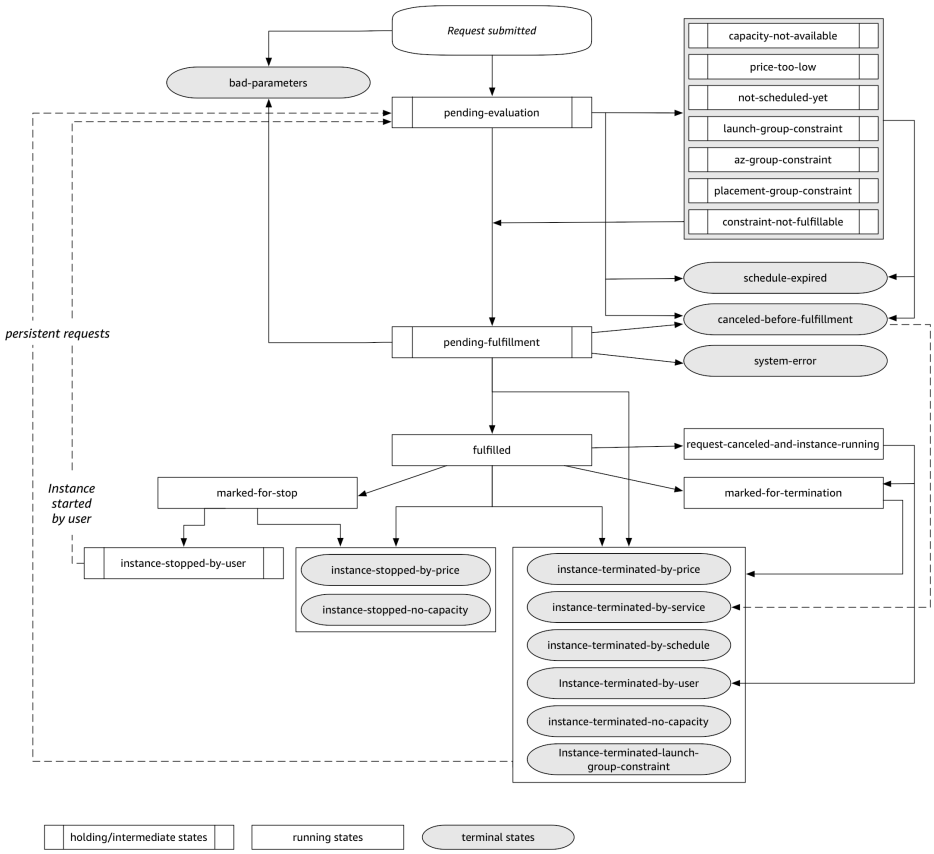
프로세스의 각 단계(스팟 요청 수명 주기라고도 함)에서 특정 이벤트에 따라 연속 요청 상태가 결정됩니다.

목차

- 스팟 요청의 수명 주기 (p. 366)
- 요청 상태 정보 가져오기 (p. 370)
- 스팟 요청 상태 코드 (p. 370)

스팟 요청의 수명 주기

다음 다이어그램에서는 제출부터 종료까지 전체 수명 주기 동안 스팟 요청이 따를 수 있는 경로를 보여 줍니다. 각 단계는 노드로 표시되며 각 노드의 상태 코드는 스팟 요청 및 스팟 인스턴스의 상태를 설명합니다.



평가 보류

하나 이상의 요청 파라미터가 유효하지 않은(`bad-parameters`) 경우를 제외하고, 스팟 인스턴스 요청을 생성하는 즉시 요청은 `pending-evaluation` 상태로 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, 최고 가격을 현재 스팟 가격보다 낮게 지정할 경우 스팟 가격이 최고 가격 아래로 떨어질 때까지 요청은 보류 상태로 유지됩니다. 가용 영역 그룹을 지정할 경우 가용 영역 제약 조건이 충족될 때까지 요청은 보류 상태로 유지됩니다.

특정 가용 영역 중단 시, 다른 가용 영역에서의 스팟 인스턴스 요청에 사용할 수 있는 예비 EC2 용량이 영향 받을 수 있습니다.

상태 코드	요청 상태	인스턴스 상태
`capacity-not-available`	`open`	해당 사항 없음
`price-too-low`	`open`	해당 사항 없음

상태 코드	요청 상태	인스턴스 상태
`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 도달하기 전에 이 기간이 만료되면 스팟 인스턴스 요청은 `terminal` 상태로 전환될 수 있습니다. 요청을 취소하거나 시스템 오류가 발생하는 경우에도 이와 같이 될 수 있습니다.

상태 코드	요청 상태	인스턴스 상태
`schedule-expired`	`cancelled`	해당 사항 없음
`canceled-before-fulfillment*`	`cancelled`	해당 사항 없음
`bad-parameters`	`failed`	해당 사항 없음
`system-error`	`closed`	해당 사항 없음

* 사용자가 요청을 취소하는 경우.

이행 보류

지정한 제약 조건(있는 경우)이 충족되고 최고 가격이 현재 스팟 가격보다 높거나 같은 경우 스팟 요청은 `pending-fulfillment` 상태가 됩니다.

이 시점에 Amazon EC2는 요청한 인스턴스를 프로비저닝할 준비를 합니다. 프로세스가 이 시점에 중지되는 경우 스팟 인스턴스가 시작되기 전에 사용자가 프로세스를 취소했기 때문일 수 있습니다. 예기치 않은 시스템 오류가 원인일 수도 있습니다.

상태 코드	요청 상태	인스턴스 상태
`pending-fulfillment`	`open`	해당 사항 없음

이행됨

스팟 인스턴스의 모든 사양이 충족되면 스팟 요청이 이행됩니다. Amazon EC2에서 스팟 인스턴스를 시작합니다. 이 작업은 몇 분 정도 걸릴 수 있습니다. 중단된 스팟 인스턴스가 최대 절전 모드로 전환되거나 중지되는 경우 요청을 다시 이행할 수 있거나 요청이 취소될 때까지 인스턴스는 이 상태를 유지합니다.

상태 코드	요청 상태	인스턴스 상태
`fulfilled`	`active`	`pending` → `running`

상태 코드	요청 상태	인스턴스 상태
fulfilled	active	stopped → running

스팟 인스턴스를 중지하면 스팟 인스턴스를 다시 시작할 수 있거나 요청을 취소할 때까지 스팟 인스턴스는 marked-for-stop 또는 instance-stopped-by-user 상태로 전환됩니다.

상태 코드	요청 상태	인스턴스 상태
marked-for-stop	active	stopping
instance-stopped-by-user*	disabled 또는 cancelled**	stopped

* 인스턴스를 중지하거나 인스턴스에서 종료 명령을 실행하면 스팟 인스턴스는 instance-stopped-by-user 상태로 전환됩니다. 인스턴스를 중지한 후 다시 시작할 수 있습니다. 다시 시작하면 스팟 인스턴스 요청은 pending-evaluation 상태로 돌아온 다음, 제약 조건이 충족되면 Amazon EC2가 새스팟 인스턴스를 시작합니다.

** 스팟 인스턴스를 중지하지만 요청을 취소하지 않는 경우 스팟 요청 상태는 disabled입니다. 스팟 인스턴스가 중지되고 요청이 만료되는 경우 요청 상태는 cancelled입니다.

이행됨-끝

최고 가격이 스팟 가격보다 높거나 같고 인스턴스 유형에 대한 가용 용량이 있으며 사용자가 인스턴스를 종료하지 않는 한 스팟 인스턴스는 계속 실행됩니다. 스팟 가격 또는 가용 용량을 변경하기 위해 Amazon EC2에서 스팟 인스턴스를 종료해야 하는 경우 스팟 요청이 종료 상태로 전환됩니다. 사용자가 스팟 요청을 취소하거나 스팟 인스턴스를 종료하는 경우에도 요청이 종료 상태로 전환됩니다.

상태 코드	요청 상태	인스턴스 상태
request-canceled-and-instance-running	cancelled	running
marked-for-stop	active	running
marked-for-termination	closed	running
instance-stopped-by-price	disabled	stopped
instance-stopped-by-user	disabled	stopped
instance-stopped-no-capacity	disabled	stopped
instance-terminated-by-price	closed(일회), open(영구)	terminated
instance-terminated-by-schedule	closed	terminated
instance-terminated-by-service	cancelled	terminated
instance-terminated-by-user	closed 또는 cancelled *	terminated

상태 코드	요청 상태	인스턴스 상태
`instance-terminated-no-capacity`	`closed`(일회), `open`(영구)	`terminated`
`instance-terminated-launch-group-constraint`	`closed`(일회), `open`(영구)	`terminated`

* 인스턴스를 종료하되 입찰을 취소하지 않는 경우 요청 상태는 `closed`입니다. 인스턴스를 종료하고 요청을 취소하는 경우 요청 상태는 `cancelled`입니다. 스팟 요청을 취소하기 전에 스팟 인스턴스를 종료해도 Amazon EC2에서 스팟 인스턴스가 종료되었음을 감지하기 전까지 지연이 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` 상태로 복귀한 후 제약 조건이 충족되면 Amazon EC2에서 새로운 스팟 인스턴스를 시작할 수 있습니다.

요청 상태 정보 가져오기

AWS Management 콘솔 또는 명령줄 도구를 사용하여 요청 상태 정보를 가져올 수 있습니다.

요청 상태 정보를 가져오려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택한 다음 스팟 요청을 선택합니다.
3. 상태를 확인하려면 설명, 상태를 선택합니다.

명령줄을 사용하여 요청 상태 정보를 가져오려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- describe-spot-instance-requests(AWS CLI)
- Get-EC2SpotInstanceRequest(Windows PowerShell용 AWS 도구)

스팟 요청 상태 코드

스팟 요청 상태 정보는 상태 코드, 업데이트 시간 및 상태 메시지로 구성됩니다. 이러한 정보를 하나로 모으면 스팟 요청 배치를 결정하는 데 도움이 됩니다.

다음은 스팟 요청 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). 상태 메시지는 어떤 파라미터가 유효하지 않은지를 나타냅니다.

`canceled-before-fulfillment`

요청이 이행되기 전에 사용자가 스팟 요청을 취소했습니다.

`capacity-not-available`

요청한 인스턴스에 사용 가능한 용량이 부족합니다.

`constraint-not-fulfillable`

하나 이상의 제약 조건이 올바르지 않기 때문에(예: 가용 영역이 존재하지 않음) 스팟 요청을 이행할 수 없습니다. 상태 메시지는 어떤 제약 조건이 유효하지 않은지를 나타냅니다.

`fulfilled`

상태 요청이 `active` 상태이며 Amazon EC2가 스팟 인스턴스를 시작하고 있습니다.

`instance-stopped-by-price`

스팟 가격이 최고 가격을 초과하여 인스턴스가 중지됩니다.

`instance-stopped-by-user`

사용자가 인스턴스를 중지했거나 인스턴스에서 종료 명령을 실행했기 때문에 인스턴스가 중지되었습니다.

`instance-stopped-no-capacity`

인스턴스에 사용 가능한 스팟 용량이 부족하여 인스턴스가 중지됩니다.

`instance-terminated-by-price`

스팟 가격이 최고 가격을 초과하여 인스턴스가 종료됩니다. 요청이 영구적일 경우 프로세스가 다시 시작되므로 요청은 평가 보류 상태입니다.

`instance-terminated-by-schedule`

스팟 인스턴스가 예약된 유효 기간의 만료로 종료되었습니다.

`instance-terminated-by-service`

인스턴스가 중지된 상태에서 종료되었습니다.

`instance-terminated-by-user` 또는 `spot-instance-terminated-by-user`

이행된 스팟 인스턴스를 종료했으므로 요청 상태는 `closed`(영구 요청이 아닌 경우)이고 인스턴스 상태는 `terminated`입니다.

`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 부족하여 인스턴스가 종료됩니다.

`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 제한을 참조하십시오.

`marked-for-stop`

스팟 인스턴스가 중지할 대상으로 표시되어 있습니다.

`marked-for-termination`

스팟 인스턴스가 종료할 대상으로 표시되어 있습니다.

`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 EC2에서 스팟 인스턴스를 프로비저닝하려고 합니다.

`placement-group-constraint`

이 시점에는 스팟 인스턴스를 배치 그룹에 추가할 수 없으므로 아직 스팟 요청을 이행할 수 없습니다.

`price-too-low`

최고 가격이 스팟 가격보다 낮기 때문에 요청을 이행할 수 없습니다. 이 경우 인스턴스가 시작되지 않으며 요청이 `open` 상태로 유지됩니다.

`request-canceled-and-instance-running`

스팟 인스턴스가 아직 실행되고 있는 중에 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 AWS Support에 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 달라질 수 있습니다. 스팟 인스턴스가 중단될 가능성은 항상 있습니다. 따라서 스팟 인스턴스 중단에 대비하여 애플리케이션을 준비해야 합니다.

EC2 집합 또는 스팟 집합에 지정된 온디맨드 인스턴스는 중단할 수 없습니다.

목차

- 서비스 중단의 이유 (p. 372)
- 중단 동작 (p. 373)
- 중단에 대한 준비 (p. 375)
- 인스턴스 최대 절전 모드 준비 (p. 375)
- 스팟 인스턴스 중단 공지 (p. 376)
- 중단된 스팟 인스턴스에 대한 청구 (p. 378)

서비스 중단의 이유

Amazon EC2에서 스팟 인스턴스를 중단시킬 수 있는 이유는 다음과 같습니다.

- 가격 – 스팟 가격이 최고 가격보다 큽니다.
- 용량 – 미사용 EC2 인스턴스가 스팟 인스턴스 수요를 충족할 만큼 충분하지 않으면 Amazon EC2에서 스팟 인스턴스를 중단합니다. 인스턴스가 중단되는 순서는 Amazon EC2에서 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.

중단 동작

스팟 인스턴스가 중단되면 Amazon EC2가 최대 절전 모드로 전환할 것인지, 중지시킬 것인지 또는 종료시킬 것인지 지정할 수 있습니다. 사용자의 요구에 적합한 중단 동작을 선택할 수 있습니다. 기본값은 중단된 스팟 인스턴스를 종료하는 것입니다. 중단 동작을 변경하려면 스팟 요청을 생성할 때 콘솔에서 인터럽트 방식을 선택하거나, 시작 구성 또는 시작 템플릿에서 `InstanceInterruptionBehavior`를 지정합니다. 스팟 요청을 생성할 때 콘솔에서 중단 동작을 변경하려면 목표 용량 유지를 선택합니다. 이 옵션을 선택하면 인터럽트 방식이 나타나며 스팟 인스턴스가 중단될 때 스팟 서비스가 스팟 인스턴스를 종료는지, 중지하는지 또는 최대 절전 모드로 전환하는지를 지정할 수 있습니다.

중단된 스팟 인스턴스 중지

다음 요구 사항이 충족되는 경우 Amazon EC2가 중단된 스팟 인스턴스를 중지하도록 이 동작을 변경할 수 있습니다.

요구 사항

- 스팟 인스턴스 요청의 유형은 `persistent`이어야 합니다. 스팟 인스턴스 요청에서 시작 그룹을 지정할 수 없습니다.
- EC2 집합 또는 스팟 집합 요청의 유형은 `maintain`이어야 합니다.
- 루트 볼륨은 인스턴스 스토어 볼륨이 아니라 EBS 볼륨이어야 합니다.

스팟 인스턴스가 Spot 서비스에 의해 중지된 후에는 Spot 서비스만 스팟 인스턴스를 다시 시작할 수 있으며 동일한 시작 사양을 사용해야 합니다.

`persistent` 스팟 인스턴스 요청에 의해 시작된 스팟 인스턴스의 경우 스팟 서비스는 용량이 동일한 가용 영역에서 사용 가능한 경우 중지된 인스턴스와 동일한 인스턴스 유형에 대해 중지된 인스턴스를 다시 시작합니다.

EC2 집합 또는 스팟 집합의 인스턴스가 중지되고 플릿이 `maintain` 유형인 경우 스팟 서비스는 대체 인스턴스를 시작하여 대상 용량을 유지합니다. Spot 서비스는 지정된 할당 전략(`lowestPrice`, `diversified` 또는 `InstancePoolsToUseCount`)을 기반으로 최상의 풀을 찾습니다. 이전에 중지된 인스턴스로 풀의 우선 순위를 지정하지 않습니다. 나중에 할당 전략으로 이전에 중지된 인스턴스가 풀에 포함되면 스팟 서비스는 중지된 인스턴스를 다시 시작하여 대상 용량을 충족합니다.

예를 들어 `lowestPrice` 할당 전략을 사용하여 스팟 집합을 고려하십시오. 초기 시작 시, `c3.large` 풀은 시작 사양의 `lowestPrice` 기준을 충족합니다. 나중에 `c3.large` 인스턴스가 중단되면 스팟 서비스는 인스턴스를 중지하고 `lowestPrice` 전략에 맞는 다른 풀에서 용량을 보충합니다. 이번에는 풀이 `c4.large` 풀로, 스팟 서비스는 `c4.large` 인스턴스를 시작하여 대상 용량을 충족합니다. 마찬가지로 스팟 집합은 다음에 `c5.large` 풀로 이동할 수 있습니다. 이러한 각각의 전환에서 스팟 서비스는 이전에 인스턴스가 중지된 풀에 우선 순위를 지정하지 않고 지정된 할당 전략에 따라 우선 순위를 지정합니다. `lowestPrice` 전략은 이전에 인스턴스가 중지된 풀로 되돌아갈 수 있습니다. 예를 들어 인스턴스가 `c5.large` 풀에서 중단되고 `lowestPrice` 전략이 다시 `c3.large` 또는 `c4.large` 풀로 연결되면 이전에 중지 된 인스턴스가 다시 시작되어 대상 용량을 충족합니다.

스팟 인스턴스가 중지되었을 때 일부 인스턴스 속성을 수정할 수 있지만 인스턴스 유형은 수정할 수 없습니다. EBS 볼륨을 분리하거나 삭제한 경우 스팟 인스턴스를 시작해도 연결되지 않습니다. 루트 볼륨을 분리했는데 스팟 서비스가 해당 스팟 인스턴스를 시작하려고 시도하면 인스턴스 시작은 실패하고 스팟 서비스는 중지된 인스턴스를 종료합니다.

중지된 상태의 스팟 인스턴스를 종료할 수 있습니다. 스팟 요청이나 EC2 집합 또는 스팟 집합을 취소하면 스팟 서비스에서는 중지된 상태의 연결된 스팟 인스턴스를 모두 종료합니다.

스팟 인스턴스가 중지 상태인 동안에는 유지 중인 EBS 볼륨에 대한 요금만 부과됩니다. EC2 집합 및 스팟 집합을 사용하는 경우 중지된 인스턴스가 많으면 해당 계정의 EBS 볼륨 수 제한을 초과할 수 있습니다.

중단된 스팟 인스턴스를 최대 절전 모드로 전환

다음 요구 사항이 충족되는 경우 Amazon EC2가 중단된 스팟 인스턴스를 최대 절전 모드로 전환하도록 이 동작을 변경할 수 있습니다.

요구 사항

- 스팟 인스턴스 요청의 유형은 `persistent`이어야 합니다. 스팟 인스턴스 요청에서 시작 그룹을 지정할 수 없습니다.
- EC2 집합 또는 스팟 집합 요청의 유형은 `maintain`이어야 합니다.
- 루트 볼륨은 인스턴스 스토어 볼륨이 아니라 EBS 볼륨이어야 하고 최대 절전 모드에서 인스턴스 메모리(RAM)를 저장하기에 충분한 크기여야 합니다.
- 메모리 용량이 100GB 미만인 C3, C4, C5, M4, M5, R3, R4 인스턴스가 지원됩니다.
- Amazon Linux 2, Amazon Linux AMI, 4.4.0-1041 이상의 AWS-tuned Ubuntu 커널(linux-aws)이 포함된 Ubuntu, Windows Server 2008 R2 이상 등의 운영 체제가 지원됩니다.
- 지원되는 운영 체제에서 최대 절전 에이전트를 설치하거나 다음과 같이 이미 에이전트를 포함하고 있는 AMI 중 하나를 사용하십시오.
  - Amazon Linux 2
  - Amazon Linux AMI 2017.09.1 이상
  - Ubuntu Xenial 16.04 20171121 이상
  - Windows Server 2008 R2 AMI 2017.11.19 이상
  - Windows Server 2012 또는 Windows Server 2012 R2 AMI 2017.11.19 이상
  - Windows Server 2016 R2 AMI 2017.11.19 이상
  - Windows Server 2019
- 에이전트를 시작합니다. 인스턴스 시작 시 에이전트를 시작하도록 사용자 데이터를 사용하는 것이 좋습니다. 또는 에이전트를 수동으로 시작할 수도 있습니다.

권장 사항

- 최대 절전 모드에서 인스턴스 메모리가 루트 볼륨에 저장되므로 암호화된 Amazon EBS 볼륨을 루트 볼륨으로 사용해야 합니다. 이렇게 해야 데이터가 볼륨에 저장되어 있고 인스턴스와 볼륨 간을 이동 중일 때 메모리(RAM)의 콘텐츠가 암호화됩니다. 루트 볼륨이 암호화된 Amazon EBS 볼륨인지 확인하려면 다음 세 가지 옵션 중 하나를 사용합니다.
  - EBS '단일 단계' 암호화: 단일 실행 인스턴스 API 호출의 경우 암호화되지 않은 AMI에서 암호화된 EBS 지원 EC2 인스턴스를 시작할 수 있습니다. 자세한 내용은 EBS 지원 AMI를 통한 암호화 사용 (p. 145) 단원을 참조하십시오.
  - 기본적인 EBS 암호화: 기본적으로 EBS 암호화를 활성화하여 AWS 계정에서 생성된 모든 새 EBS 볼륨이 암호화되도록 할 수 있습니다. 자세한 내용은 암호화 기본 제공 (p. 971) 단원을 참조하십시오.
  - 암호화된 AMI: 암호화된 AMI를 사용하여 인스턴스를 시작하는 방식으로 EBS 암호화를 활성화할 수 있습니다. AMI에 암호화된 루트 스냅샷이 없을 경우, 이를 새로운 AMI 및 요청 암호화에 복사할 수 있습니다. 자세한 내용은 복사 중에 암호화되지 않은 이미지 암호화 (p. 149) 및 AMI 복사 (p. 154) 단원을 참조하십시오.

스팟 인스턴스가 스팟 서비스에 의해 최대 절전 모드로 전환되어도 루트 볼륨에서 EBS 볼륨과 인스턴스 메모리(RAM)이 유지됩니다. 인스턴스의 프라이빗 IP 주소 역시 유지됩니다. 인스턴스 스토리지 볼륨과 다른 탄력적 IP 주소가 아닌 퍼블릭 IP 주소는 유지되지 않습니다. 인스턴스가 최대 절전 모드에 있는 동안에는

EBS 볼륨에 대한 요금만 부과됩니다. EC2 집합 및 스팟 집합을 사용하는 경우 최대 절전 모드로 전환된 인스턴스가 많으면 해당 계정의 EBS 볼륨 수 제한을 초과할 수 있습니다.

인스턴스가 스팟 서비스에서 신호를 수신할 때 에이전트는 운영 체제에 최대 절전 모드로 전환하라는 메시지를 보냅니다. 에이전트가 설치되어 있지 않거나 기반 운영 체제가 최대 절전 모드를 지원하지 않거나 인스턴스 메모리를 저장하기 충분한 볼륨 공간이 없는 경우에는 최대 절전 모드로의 전환이 실패하고 대신에 스팟 서비스가 인스턴스를 중지합니다.

스팟 서비스가 스팟 인스턴스를 최대 절전 모드로 전환하면 스팟 인스턴스가 중단되기 2분 전에 중지 공지를 수신합니다. 최대 절전 모드가 즉시 시작됩니다. 인스턴스가 최대 절전 모드를 활성화하는 동안에는 인스턴스 상태 확인이 실패할 수 있습니다. 최대 절전 모드 활성화가 완료되면 인스턴스의 상태는 `stopped`가 됩니다.

스팟 서비스에 의해 최대 절전 모드로 전환된 스팟 인스턴스는 해당 스팟 서비스에 의해서만 재개될 수 있습니다. 지정된 최고 가격보다 낮은 스팟 가격에서 용량이 가용 상태가 되면 스팟 서비스가 인스턴스를 재개합니다.

자세한 내용은 인스턴스 최대 절전 모드 준비 (p. 375) 단원을 참조하십시오.

온디맨드 인스턴스 최대 절전 모드 전환에 대한 자세한 내용은 Linux 인스턴스를 최대 절전 모드로 실행합니다. (p. 510) 단원을 참조하십시오.

중단에 대한 준비

스팟 인스턴스 사용 시 따라야 할 몇 가지 모범 사례는 다음과 같습니다.

- 기본 최고 가격인 온디맨드 가격을 사용하십시오.
- 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 종료되어도 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 중단 공지를 사용하여 스팟 인스턴스의 상태를 모니터링합니다.
- 이 경고를 즉시 제공하기 위해 모든 노력을 기울이고 있지만 경고를 보내기 전에 스팟 인스턴스가 종료될 수도 있습니다. 따라서 중단 공지를 테스트할 때도 애플리케이션을 테스트하여 예상치 않은 인스턴스 종료가 정상적으로 처리되는지 확인하십시오. 이렇게 하려면 온디맨드 인스턴스를 사용하여 애플리케이션을 실행한 다음 온디맨드 인스턴스를 직접 종료합니다.

인스턴스 최대 절전 모드 준비

이미 에이전트를 포함하고 있는 AMI를 사용하지 않는 한, 인스턴스에 최대 절전 모드 에이전트를 설치해야 합니다. 에이전트가 AMI에 포함되었거나 직접 설치했는지 여부와 관계없이 인스턴스 시작 시 에이전트를 실행해야 합니다.

다음 절차를 따르면 Linux 인스턴스를 준비하는 데 도움이 됩니다. Windows 인스턴스를 준비하는 방법은 Windows 인스턴스용 Amazon EC2 사용 설명서의 인스턴스 최대 절전 모드 준비를 참조하십시오.

Amazon Linux 인스턴스를 준비하려면

1. 커널이 최대 절전 모드를 지원하고 필요할 경우 커널을 업데이트하는지 확인합니다.
2. AMI에 에이전트가 포함되어 있지 않은 경우에는 다음 명령을 사용하여 에이전트를 설치합니다.

```
sudo yum update; sudo yum install hibagent
```

3. 사용자 데이터에 다음을 추가합니다.

```
#!/bin/bash
/usr/bin/enable-ec2-spot-hibernation
```

Ubuntu 인스턴스를 준비하려면

1. AMI에 에이전트가 포함되어 있지 않은 경우에는 다음 명령을 사용하여 에이전트를 설치합니다. 최대 절전 모드 에이전트는 Ubuntu 16.04 이상에서만 사용할 수 있습니다.

```
sudo apt-get install hibagent
```

2. 사용자 데이터에 다음을 추가합니다.

```
#!/bin/bash
/usr/bin/enable-ec2-spot-hibernation
```

스팟 인스턴스 중단 공지

스팟 인스턴스 중단을 방지하는 가장 좋은 방법은 내결함성을 유지하도록 애플리케이션을 설계하는 것입니다. 또한 Amazon EC2가 스팟 인스턴스를 중단하거나 종료하기 2분 전에 경고해 주는 스팟 인스턴스 중단 공지를 이용할 수 있습니다. 5초마다 이러한 경고를 확인하는 것이 좋습니다.

이 경고를 CloudWatch 이벤트 및 스팟 인스턴스의 인스턴스 메타데이터 (p. 571) 항목으로 사용할 수 있습니다.

최대 절전을 중지 행동으로 지정할 경우 중지 공지를 수신하지만, 최대 절전 과정은 즉시 시작되므로 2분 경고를 받지 않습니다.

EC2 스팟 인스턴스 Interruption Notice

Amazon EC2 스팟 인스턴스를 중단하려고 하면 실제 중단하기 2분 전에 이벤트를 내보냅니다. 이 이벤트는 Amazon CloudWatch Events를 통해 감지될 수 있습니다. 자세한 내용은 Amazon CloudWatch Events 사용 설명서 단원을 참조하십시오.

다음은 스팟 인스턴스 중단에 대한 이벤트의 예제입니다. 가능한 `instance-action` 값은 `hibernate`, `stop` 및 `terminate`입니다.

```
{
    "version": "0",
    "id": "12345678-1234-1234-1234-123456789012",
    "detail-type": "EC2 Spot Instance Interruption Warning",
    "source": "aws.ec2",
    "account": "123456789012",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-2",
    "resources": ["arn:aws:ec2:us-east-2:123456789012:instance/i-1234567890abcdef0"],
    "detail": {
        "instance-id": "i-1234567890abcdef0",
        "instance-action": "action"
    }
}
```

instance-action

스팟 인스턴스가 스팟 서비스에서 중지하거나 종료하도록 표시된 경우 인스턴스 메타데이터 (p. 571)에 `instance-action` 항목이 있습니다. 표시하지 않은 경우에는 이 항목이 없습니다. 다음과 같이 `instance-action`을 검색할 수 있습니다.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/spot/instance-action
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/spot/instance-action
```

`instance-action` 항목은 해당 작업과 작업이 이루어지는 대략적 시간(UTC 기준)을 지정합니다.

다음 예제에서는 이 인스턴스가 중지될 시간을 알려 줍니다.

```
{"action": "stop", "time": "2017-09-18T08:22:00Z"}
```

다음 예제에서는 이 인스턴스가 종료될 시간을 알려 줍니다.

```
{"action": "terminate", "time": "2017-09-18T08:22:00Z"}
```

Amazon EC2가 인스턴스를 중지 또는 종료할 준비가 되지 않거나 사용자가 인스턴스를 직접 종료한 경우 `instance-action` 항목이 나타나지 않고 HTTP 404 오류를 수신하게 됩니다.

termination-time

이 항목은 이전 버전과의 호환성을 위해 보존되며, 그 대신 `instance-action`을 사용해야 합니다.

스팟 서비스에서 종료할 스팟 인스턴스를 표시한 경우 인스턴스 메타데이터에 `termination-time` 항목이 있습니다. 표시하지 않은 경우에는 이 항목이 없습니다. 다음과 같이 `termination-time`을 검색할 수 있습니다.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"`
[ec2-user ~]$ if curl -H "X-aws-ec2-metadata-token: $TOKEN" -s http://169.254.169.254/latest/meta-data/spot/termination-time | grep -q .*T.*Z; then echo terminated; fi
```

IMDSv1

```
[ec2-user ~]$ if curl -s http://169.254.169.254/latest/meta-data/spot/termination-time
 | grep -q .*T.*Z; then echo terminated; fi
```

`termination-time` 항목은 인스턴스가 종료 신호를 받게 될 대략의 시간(UTC 기준)을 지정합니다. 다음 예를 참조하십시오.

```
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 인스턴스를 종료하지 않으면 요청 상태가 `fulfilled`로 설정됩니다. `termination-time` 값은 과거 시점인 원래 예상 시간과 함께 인스턴스 메타데이터에 남습니다.

중단된 스팟 인스턴스에 대한 청구

스팟 인스턴스(스팟 블록에 있지 않음)가 중단되면 다음과 같이 요금이 부과됩니다.

스팟 인스턴스를 중단하는 사람	운영 체제	첫 번째 시간에 중단됨	첫 번째 시간 후 어떤 시간에든 중단됨
사용자가 스팟 인스턴스를 중지하거나 종료하는 경우	Linux(RHEL 및 SUSE 제외)	사용된 시간(초)에 대해 부과	사용된 시간(초)에 대해 부과
	Windows, RHEL, SUSE	일부 시간만 사용한 경우에도 전체 시간에 대해 부과	사용한 전체 시간에 대해 부과되며, 중단된 일부 시간에 대해 전체 시간 요금 부과
Amazon EC2가 스팟 인스턴스를 중단하는 경우	Linux(RHEL 및 SUSE 제외)	무료	사용된 시간(초)에 대해 부과
	Windows, RHEL, SUSE	무료	사용한 전체 시간에 대해 부과되지만, 중단된 일부 시간에 대한 요금은 무료

스팟 블록에 있는 스팟 인스턴스가 중단되면 다음과 같이 요금이 부과됩니다.

스팟 인스턴스를 중단하는 사람	운영 체제	첫 번째 시간에 중단됨	첫 번째 시간 후 어떤 시간에든 중단됨
사용자가 스팟 인스턴스를 중지하거나 종료하는 경우	Linux(RHEL 및 SUSE 제외)	사용된 시간(초)에 대해 부과	사용된 시간(초)에 대해 부과
	Windows, RHEL, SUSE	일부 시간만 사용한 경우에도 전체 시간에 대해 부과	사용한 전체 시간에 대해 부과되며, 중단된 일부 시간에 대해 전체 시간 요금 부과
Amazon EC2가 스팟 인스턴스를 중단하는 경우	Linux(RHEL 및 SUSE 제외)	무료	무료
	Windows, RHEL, SUSE	무료	무료

스팟 인스턴스 데이터 피드

스팟 인스턴스 요금을 쉽게 이해할 수 있도록 Amazon EC2에서는 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.

내용

- 데이터 피드 파일 이름 및 형식 (p. 379)
- Amazon S3 버킷 요구 사항 (p. 379)
- 스팟 인스턴스 데이터 피드 구독 (p. 380)
- 스팟 인스턴스 데이터 피드 삭제 (p. 380)

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 *aws-s3-bucket1*이고 접두사가myprefix인 경우 파일 이름은 다음과 같습니다.

```
aws-s3-bucket1.s3.amazonaws.com/myprefix/111122223333.2014-03-17-20.001.pwBdGTJG.gz
```

스팟 인스턴스 데이터 피드 파일은 탭으로 구분됩니다. 데이터 파일의 각 줄은 1 인스턴스 시간에 해당하며 다음 표에 나열된 필드를 포함합니다.

필드	설명
`Timestamp`	이 인스턴스 사용량에 대해 청구된 가격을 결정하는 데 사용되는 타임스탬프입니다.
`UsageType`	청구되는 사용 유형 및 인스턴스 유형입니다. `m1.small` 스팟 인스턴스의 경우 이 필드는 `SpotUsage`로 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 `RunInstances`로 설정됩니다. Windows 스팟 인스턴스의 경우 이 필드는 `RunInstances:0002`로 설정됩니다. 스팟 사용은 가용 영역에 따라 그룹화됩니다.
`InstanceID`	이 인스턴스 사용량을 생성한 스팟 인스턴스의 ID입니다.
`MyBidID`	이 인스턴스 사용량을 생성한 스팟 인스턴스 요청의 ID입니다.
`MyMaxPrice`	이 스팟 인스턴스 요청에 대해 지정된 최고 가격입니다.
`MarketPrice`	`Timestamp` 필드에 지정된 시간의 스팟 가격입니다.
`Charge`	이 인스턴스 사용량에 대해 청구된 가격입니다.
`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.

- `FULL_CONTROL` 및 `s3:GetBucketAcl` 작업에 대한 권한을 포함하여 버킷에 대한 `s3:PutBucketAcl` 권한이 있어야 합니다.

  버킷 소유자인 경우 기본적으로 이 권한이 있습니다. 그렇지 않으면 버킷 소유자가 AWS 계정에 이 권한을 부여해야 합니다.
- 데이터 피드를 구독할 때 이러한 권한으로 버킷 ACL을 업데이트하여 AWS 데이터 피드 계정에 `FULL_CONTROL` 권한을 부여합니다. AWS 데이터 피드 계정은 버킷에 데이터 피드 파일을 씁니다. 필요한 권한이 계정에 없을 경우 데이터 피드 파일을 버킷에 쓸 수 없습니다.

Note

ACL을 업데이트하고 AWS 데이터 피드 계정에 대한 권한을 제거할 경우 데이터 피드 파일을 버킷에 쓸 수 없습니다. 데이터 피드 파일을 수신하려면 데이터 피드를 다시 구독해야 합니다.

- 각 데이터 피드 파일에는 고유의 ACL(버킷용 ACL과는 별도)이 있습니다. 버킷 소유자는 데이터 파일에 대한 `FULL_CONTROL` 권한을 가지고 있습니다. AWS 데이터 피드 계정은 읽기 및 쓰기 권한이 있습니다.
- 데이터 피드 구독을 삭제해도 Amazon EC2에서 버킷 또는 데이터 파일에 대한 AWS 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 aws-s3-bucket1 [--prefix myprefix]
```

다음은 예제 출력입니다.

```
{
    "SpotDatafeedSubscription": {
        "OwnerId": "111122223333",
        "Prefix": "myprefix",
        "Bucket": "aws-s3-bucket1",
        "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

- 스팟 요청 제한 (p. 380)
- 스팟 집합 제한 (p. 381)
- T3 스팟 인스턴스 (p. 381)
- T2 스팟 인스턴스 (p. 381)

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 제한 계산에 반영됩니다.

스팟 인스턴스 제한은 동적으로 바뀝니다. 새 계정인 경우 한도가 20개 미만일 수 있지만 시간이 지날수록 늘어날 수 있습니다. 또한 특정 스팟 인스턴스 유형에 대한 계정 제한도 있을 수 있습니다. 스팟 인스턴스 요청을 제출하고 `Max spot instance count exceeded` 오류가 발생하면 AWS 지원 센터 사례 생성 서식을 작성하여 스팟 인스턴스 한도 증가를 요청할 수 있습니다. Limit type(한도 유형)에서 EC2 Spot

Instances(EC2 스팟 인스턴스)를 선택합니다. 자세한 내용은 Amazon EC2 서비스 제한 (p. 1079) 단원을 참조하십시오.

스팟 집합 제한

스팟 요청 가격 제한, 인스턴스 제한 및 볼륨 제한과 같이 스팟 집합 또는 EC2 집합에서 시작된 인스턴스에 일반적인 Amazon EC2 제한이 적용됩니다. 또한 다음과 같은 제한이 적용됩니다.

- 리전당 활성 스팟 집합 및 EC2 집합의 수: 1,000개*
- 플릿당 시작 사양 수: 300개*
- 시작 템플릿과 플릿당 재정의 수: 300개*
- 시작 사양의 사용자 데이터 크기: 16KB*
- 스팟 집합 또는 EC2 집합당 대상 용량: 10,000
- 리전 내 모든 스팟 집합 및 EC2 집합의 대상 용량: 100,000
- 스팟 집합 요청 또는 EC2 집합 요청으로 리전을 확장할 수 없습니다.
- 스팟 집합 요청 또는 EC2 집합 요청으로 동일한 가용 영역의 서로 다른 서브넷을 확장할 수 없습니다.

대상 용량의 기본 한도 이상이 필요한 경우, AWS 지원 센터 사례 생성 서식을 작성하여 한도 증가를 요청하십시오. 제한 유형에서 EC2 Fleet(EC2 플릿)을 선택하고 리전을 선택한 다음 Target Fleet Capacity per Fleet (in units)(플릿당 대상 플릿 용량(유닛)) 또는 Target Fleet Capacity per Region (in units)(리전당 대상 플릿 용량(유닛))을 선택하거나 둘 다 선택합니다.

* 이 숫자는 하드 제한입니다. 이러한 제한에 대한 한도 상승을 요청할 수 없습니다.

T3 스팟 인스턴스

T3 스팟 인스턴스를 CPU 크레딧 발생에 대한 유휴 시간 없이 즉시 짧은 기간 동안 사용할 계획인 경우 T3 스팟 인스턴스를 `standard` (p. 206) 모드로 시작하여 높은 비용 지불을 방지하는 것이 좋습니다.

T3 스팟 인스턴스를 `unlimited` (p. 197) 모드를 시작하고 CPU를 즉시 버스트하는 경우 버스팅에 대한 잉여 크레딧을 소모하게 됩니다. 인스턴스를 짧은 기간 동안 사용하는 경우 인스턴스에서 잉여 크레딧을 지불할 정도의 CPU 크레딧이 발생할 시간이 없습니다. 인스턴스를 종료할 때 잉여 크레딧에 대한 요금이 청구됩니다.

T3 스팟 인스턴스가 `Unlimited` 모드인 경우는 버스팅에 대한 CPU 크레딧이 발생할 정도로 인스턴스 실행이 긴 경우에만 적합합니다. 그렇지 않은 경우 잉여 크레딧에 대한 비용을 지불하면 T3 스팟 인스턴스가 M5 또는 C5 인스턴스에 비해 비싸집니다. 자세한 내용은 무제한 모드 대 고정 CPU 사용 시기 (p. 199) 단원을 참조하십시오.

T2 스팟 인스턴스

시작 크레딧은 효율적인 컴퓨팅 리소스를 제공하여 인스턴스를 구성함으로써 T2 인스턴스에 대한 생산적인 최초 시작 환경을 제공하는 것을 목적으로 합니다. 새 시작 크레딧에 액세스하기 위한 T2 인스턴스의 반복된 시작은 허용되지 않습니다. 지속적인 CPU가 필요한 경우 (일정 기간 동안 유휴 상태로 둠으로써) 크레딧을 얻고 T2 무제한 (p. 197)을 사용하거나 전용 CPU(예: `c4.large`)를 포함한 인스턴스 유형을 사용할 수 있습니다.

전용 호스트

Amazon EC2 전용 호스트는 완전히 고객 전용인 EC2 인스턴스 용량을 갖춘 물리적 서버입니다. 전용 호스트에서는 Windows Server, Microsoft SQL Server, SUSE 및 Linux Enterprise Server를 포함한 기존 소켓당, 코어당 또는 VM당 소프트웨어 라이선스를 사용할 수 있습니다.

목차

- 전용 호스트과 전용 인스턴스의 차이점 (p. 382)

- 기존 보유 라이선스 사용 (p. 382)
- 전용 호스트 인스턴스 용량 (p. 383)
- 전용 호스트 제한 사항 (p. 383)
- 요금 및 결제 (p. 383)
- 전용 호스트 작업 (p. 384)
- 공유된 전용 호스트 작업 (p. 401)
- 호스트 복구 (p. 404)
- 구성 변경 추적 (p. 408)

전용 호스트과 전용 인스턴스의 차이점

전용 호스트와 전용 인스턴스는 모두 사용자 전용 물리적 서버로 Amazon EC2 인스턴스를 시작하는 데 사용할 수 있습니다.

전용 호스트의 인스턴스와 전용 인스턴스는 성능이나 보안상의 차이나 물리적 차이는 없습니다. 그러나 둘 사이에는 약간의 차이가 있습니다. 다음 표에는 전용 호스트와 전용 인스턴스의 주요 차이점 중 일부를 요약하여 설명합니다.

	전용 호스트	전용 인스턴스
결제	호스트 단위 결제	인스턴스 단위 결제
소켓, 코어 및 호스트 ID 표시 여부	소켓 및 물리 코어 수 표시 여부 제공	표시 여부 없음
호스트 및 인스턴스 선호도	시간에 따라 지속적으로 동일한 물리 서버에 인스턴스 배포 허용	지원되지 않음
대상 지정 인스턴스 배치	물리 서버 내 인스턴스 배치 방법에 대한 추가 가시성 및 제어 제공	지원되지 않음
자동 인스턴스 복구	지원 자세한 내용은 호스트 복구 (p. 404) 단원을 참조하십시오.	지원
Bring Your Own License(BYOL)	지원	지원되지 않음

기존 보유 라이선스 사용

전용 호스트를 통해 기존 소켓당, 코어당 또는 VM당 소프트웨어 라이선스를 사용할 수 있습니다. 기존 보유 라이선스를 사용하는 경우 자체 라이선스 관리에 대한 책임은 고객에게 있습니다. 하지만 Amazon EC2에는 인스턴스 선호도 및 대상 지정 배치와 같은 라이선스 규정 준수를 유지하는 데 도움이 되는 기능이 있습니다.

다음은 Amazon EC2에서 기존 볼륨 라이선스 머신 이미지를 사용하려면 수행해야 할 일반 단계입니다.

1. 머신 이미지 사용을 제어하는 라이선스 조건이 가상 클라우드 환경에서 사용을 허용하는지 확인합니다.
2. 머신 이미지를 Amazon EC2 내에서 사용할 수 있음을 확인한 이후 VM Import/Export를 사용하여 가져옵니다. 머신 이미지를 가져오는 방법에 대한 자세한 내용은 VM Import/Export 사용 설명서를 참조하십시오.
3. 머신 이미지를 가져온 후 이 머신 이미지에서 계정에 있는 활성 전용 호스트로 인스턴스를 시작할 수 있습니다.
4. 이러한 인스턴스를 실행할 때 운영 체제에 따라 자체 KMS 서버.

Note

AWS에서 이미지가 어떻게 사용되는지 추적하려면 AWS Config에서 호스트 기록을 활성화합니다. AWS Config를 사용하여 전용 호스트에 대한 구성 변경을 기록하고 출력을 라이선스 보고용 데이터 소스로 사용할 수 있습니다. 자세한 내용은 구성 변경 추적 (p. 408) 단원을 참조하십시오.

전용 호스트 인스턴스 용량

AWS Nitro 시스템으로 구동되는 전용 호스트는 동일한 인스턴트 패밀리 내에서 여러 인스턴스 유형을 지원할 수 있습니다. 예를 들어, `r5` 전용 호스트를 할당하면 소켓 2개와 물리적 코어 48개가 있는 호스트를 사용할 수 있으며 이 호스트에서 `r5.2xlarge` 및 `r5.4xlarge`와 같은 다양한 인스턴스 유형을 실행할 수 있습니다. 호스트와 연결된 코어 용량의 한도 내에서 인스턴스를 몇 개든지 실행할 수 있습니다. 예를 들어, 아래 표는 전용 호스트에서 실행할 수 있는 다양한 인스턴스 유형 조합을 보여 줍니다.

인스턴스 패밀리	인스턴스 유형 조합 예	
`R5`	• 예 1: 4 x `r5.4xlarge` + 4 x `r5.2xlarge` • 예 2: 1 x `r5.12xlarge` + 1 x `r5.4xlarge` + 1 x `r5.2xlarge` + 5 x `r5.xlarge` + 2 x `r5.large`	
`C5`	• 예 1: 1 x `c5.9xlarge` + 2 x `c5.4xlarge` + 1 x `c5.xlarge` • 예 2: 4 x `c5.4xlarge` + 1 x `c5.xlarge` + 2 x `c5.large`	
`M5`	• 예 1: 4 x `m5.4xlarge` + 4 x `m5.2xlarge` • 예 2: 1 x `m5.12xlarge` + 1 x `m5.4xlarge` + 1 x `m5.2xlarge` + 5 x `m5.xlarge` + 2 x `m5.large`	

AWS Nitro 시스템으로 구동되지 않는 인스턴스 패밀리의 경우 특정 인스턴스 유형에 대해서만 전용 호스트를 구성할 수 있습니다. 전용 호스트에서 지원되는 모든 인스턴스 패밀리 및 인스턴스 유형 구성의 목록은 Amazon EC2 전용 호스트 요금을 참조하십시오.

전용 호스트 제한 사항

전용 호스트를 할당하기 전에 다음 제한 및 제약에 유의하십시오.

- RHEL, SUSE Linux 및 Windows AMI(AWS 또는 AWS Marketplace에서 제공)는 전용 호스트와 함께 사용할 수 없습니다.
- 리전당 인스턴스 패밀리당 최대 2개의 온디맨드 전용 호스트를 할당할 수 있습니다. Amazon EC2 전용 호스트 할당 한도 상향 요청에서 한도 상향을 요청할 수 있습니다.
- 전용 호스트에서 실행되는 인스턴스는 VPC에서만 시작할 수 있습니다.
- 호스트 리소스 그룹을 지정하는 시작 템플릿을 사용하는 경우 Auto Scaling 그룹이 지원됩니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Auto Scaling 그룹을 위한 시작 템플릿 만들기를 참조하십시오.
- 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.

요금 및 결제

전용 호스트의 요금은 결제 옵션에 따라 다릅니다.

결제 옵션

- 온디맨드 전용 호스트 (p. 384)
- 전용 호스트 예약 (p. 384)
- Savings Plans (p. 384)

온디맨드 전용 호스트

계정에 전용 호스트를 할당하면 온디맨드 결제가 자동으로 활성화됩니다.

전용 호스트에 대한 온디맨드 요금은 인스턴스 패밀리 및 리전에 따라 다릅니다. 시작하기로 선택한 인스턴스의 수 또는 크기에 관계없이 전용 호스트에 대해 시간당 요금이 청구됩니다. 다시 말해 실행하기로 선택한 개별 인스턴스가 아닌 전체 전용 호스트에 대해 요금이 청구됩니다. 온디맨드 요금에 대한 자세한 내용은 Amazon EC2 전용 호스트 온디맨드 요금 단원을 참조하십시오.

언제든 온디맨드 전용 호스트를 해제하여 이에 대한 요금 청구를 중단시킬 수 있습니다. 전용 호스트 해제에 대한 자세한 내용은 전용 호스트 해제 (p. 397) 단원을 참조하십시오.

전용 호스트 예약

전용 호스트 예약는 실행 중인 온디맨드 전용 호스트와 비교해 청구 할인을 제공합니다. 다음과 같은 세 가지 결제 방식을 통해 예약이 가능합니다.

- 선결제 없음 - 선결제가 없는 예약은 사용 기간 동안 전용 호스트 사용에 대해 할인을 제공하고 선결제가 필요하지 않습니다. 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 활성 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`, `m4.large`)가 있는 경우 단일 `m4` 예약을 이 모든 전용 호스트와 연결할 수 있습니다. 예약의 인스턴스 패밀리 및 리전은 연결하고자 하는 전용 호스트의 인스턴스 패밀리 및 리전과 일치해야 합니다.

하나의 예약이 전용 호스트와 연결되면 예약 기간이 끝날 때까지 전용 호스트를 해제할 수 없습니다.

예약 요금에 대한 자세한 정보는 Amazon EC2 전용 호스트 요금 단원을 참조하십시오.

Savings Plans

Savings Plans는 온디맨드 인스턴스를 통해 상당한 비용 절감을 제공하는 유연한 요금 모델입니다. Savings Plans에서는 1년 또는 3년 기간 동안 시간당 USD로 일관적인 사용량을 약정합니다. Savings Plans는 특정 전용 호스트를 약정하는 것이 아니라 사용자의 요구 사항에 가장 적합하고 지속적으로 비용을 절약해 주는 전용 호스트를 사용할 수 있는 유연성을 제공합니다. 자세한 내용은 AWS Savings Plans 사용 설명서를 참조하십시오.

전용 호스트 작업

전용 호스트를 사용하려면 먼저 계정에서 사용할 호스트를 할당해야 합니다. 그런 다음, 인스턴스에 대해 호스트 테넌시를 지정하여 호스트에서 인스턴스를 시작합니다. 인스턴스를 시작할 특정 호스트를 선택해야 합니다. 또는 자동 배치가 활성화되었고 인스턴스 유형이 일치하는 모든 호스트에서 시작하도록 허용할 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 호스트를 해제합니다.

전용 호스트는 AWS License Manager와도 통합됩니다. License Manager를 사용하면 단일 개체로 관리되는 전용 호스트 모음인 호스트 리소스 그룹을 만들 수 있습니다. 호스트 리소스 그룹을 만들 때 auto-allocate, auto-release와 같은 호스트 관리 기본 설정을 전용 호스트에 지정합니다. 그러면 수동으로 호스트를 할당하고 관리할 필요 없이 전용 호스트에서 인스턴스를 시작할 수 있습니다. 자세한 내용은 AWS License Manager 사용 설명서의 호스트 리소스 그룹을 참조하십시오.

목차

- 전용 호스트 할당 (p. 385)
- 전용 호스트에서 인스턴스 시작 (p. 387)
- 호스트 리소스 그룹에서 인스턴스 시작 (p. 389)
- 자동 배치 및 선호도의 이해 (p. 390)
- 전용 호스트 자동 배치 수정 (p. 390)
- 지원되는 인스턴스 유형 수정 (p. 391)
- 인스턴스 테넌시 및 선호도 수정 (p. 393)
- 전용 호스트 보기 (p. 394)
- 전용 호스트 태그 지정 (p. 395)
- 전용 호스트 모니터링 (p. 396)
- 전용 호스트 해제 (p. 397)
- 전용 호스트 예약 구입 (p. 398)
- 전용 호스트 예약 보기 (p. 400)
- 전용 호스트 예약 태그 지정 (p. 400)

전용 호스트 할당

전용 호스트 사용을 시작하려면 Amazon EC2 콘솔 또는 명령줄 도구를 사용하여 계정에서 전용 호스트를 할당해야 합니다.

전용 호스트를 할당한 후에는 전용 호스트 용량을 계정에서 즉시 사용할 수 있으며 전용 호스트에서 인스턴스를 시작할 수 있습니다.

다음 방법을 사용하여 전용 호스트를 할당할 수 있습니다.

새로운 콘솔

전용 호스트를 할당하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 전용 호스트를 선택한 후 전용 호스트 할당을 선택합니다.
3. 인스턴스 패밀리에서 전용 호스트에 대한 인스턴스 패밀리를 선택합니다.
4. 전용 호스트가 선택한 인스턴스 패밀리 내의 여러 인스턴스 유형을 지원하는지 또는 특정 인스턴스 유형만 지원하는지를 지정합니다. 다음 중 하나를 수행하십시오.
   - 선택한 인스턴스 패밀리 내의 여러 인스턴스 유형을 지원하도록 전용 호스트를 구성하려면 Support multiple instance types(여러 인스턴스 유형 지원)에서 활성화를 선택합니다. 이 항목을 활성화하면 동일한 인스턴스 패밀리의 다양한 인스턴스 유형을 전용 호스트에서 시작할 수 있습니다. 예를 들어 `m5` 인스턴스 패밀리를 선택하고 이 옵션을 선택하는 경우 `m5.xlarge` 및 `m5.4xlarge` 인스턴스를 전용 호스트에서 시작할 수 있습니다. 다음 인스턴스 패밀리는 여러 인스턴스 유형을 지원하도록 구성할 수 있습니다. A1, C5, C5n, M5, M5n, R5, and R5n
   - 선택한 인스턴스 패밀리 내의 특정 인스턴스 유형을 지원하도록 전용 호스트를 구성하려면 여러 인스턴스 유형 지원의 선택을 취소한 다음 인스턴스 유형에서 지원할 인스턴스 유형을 선택합니

다. 이렇게 하면 전용 호스트에서 단일 인스턴스 유형을 시작할 수 있습니다. 예를 들어 이 옵션을 선택하고 `m5.4xlarge`를 지원되는 인스턴스 유형으로 선택하는 경우 `m5.4xlarge` 인스턴스만 전용 호스트에서 시작할 수 있습니다.

5. 가용 영역에서 전용 호스트를 할당할 가용 영역을 선택합니다.
6. 전용 호스트가 인스턴스 유형과 일치하는 대상이 지정되지 않은 인스턴스 시작을 허용하게 하려면 인스턴스 자동 배치에서 활성화를 선택합니다. 자동 배치에 대한 자세한 정보는 자동 배치 및 선호도의 이해 (p. 390) 단원을 참조하십시오.
7. 전용 호스트의 호스트 복구를 사용하려면 Host recovery(호스트 복구)에서 활성화를 선택합니다. 자세한 내용은 호스트 복구 (p. 404) 단원을 참조하십시오.
8. 수량에 할당할 전용 호스트 수를 입력합니다.
9. (선택 사항) 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
10. [Allocate]를 선택합니다.

기존 콘솔

전용 호스트를 할당하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 전용 호스트, 전용 호스트 할당을 선택합니다.
3. 인스턴스 패밀리에서 전용 호스트에 대한 인스턴스 패밀리를 선택합니다.
4. 전용 호스트가 선택한 인스턴스 패밀리 내의 여러 인스턴스 유형을 지원하는지 또는 특정 인스턴스 유형만 지원하는지를 지정합니다. 다음 중 하나를 수행하십시오.
   - 선택한 인스턴스 패밀리 내의 여러 인스턴스 유형을 지원하도록 전용 호스트를 구성하려면 여러 인스턴스 유형 지원을 선택합니다. 이 항목을 활성화하면 동일한 인스턴스 패밀리의 다양한 인스턴스 유형을 전용 호스트에서 시작할 수 있습니다. 예를 들어 `m5` 인스턴스 패밀리를 선택하고 이 옵션을 선택하는 경우 `m5.xlarge` 및 `m5.4xlarge` 인스턴스를 전용 호스트에서 시작할 수 있습니다. 다음 인스턴스 패밀리는 여러 인스턴스 유형을 지원하도록 구성할 수 있습니다. A1, C5, C5n, M5, M5n, R5, and R5n
   - 선택한 인스턴스 패밀리 내의 특정 인스턴스 유형을 지원하도록 전용 호스트를 구성하려면 여러 인스턴스 유형 지원의 선택을 취소한 다음 인스턴스 유형에서 지원할 인스턴스 유형을 선택합니다. 이 항목을 활성화하면 전용 호스트에서 단일 인스턴스 유형을 시작할 수 있습니다. 예를 들어 이 옵션을 선택하고 `m5.4xlarge`를 지원되는 인스턴스 유형으로 선택하는 경우 `m5.4xlarge` 인스턴스만 전용 호스트에서 시작할 수 있습니다.
5. 가용 영역에서 전용 호스트를 할당할 가용 영역을 선택합니다.
6. 전용 호스트가 인스턴스 유형과 일치하는 대상이 지정되지 않은 인스턴스 시작을 허용하게 하려면 인스턴스 자동 배치에서 활성화를 선택합니다. 자동 배치에 대한 자세한 정보는 자동 배치 및 선호도의 이해 (p. 390) 단원을 참조하십시오.
7. 전용 호스트의 호스트 복구를 사용하려면 Host recovery(호스트 복구)에서 Enable(활성화)을 선택합니다. 자세한 내용은 호스트 복구 (p. 404) 단원을 참조하십시오.
8. 수량에 할당할 전용 호스트 수를 입력합니다.
9. (선택 사항) 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
10. 호스트 할당을 선택합니다.

AWS CLI

전용 호스트를 할당하려면

allocate-hosts AWS CLI 명령을 사용합니다. 다음 명령은 `us-east-1a` 가용 영역에서 `m5` 인스턴스 패밀리의 여러 인스턴스 유형을 지원하는 전용 호스트를 할당합니다. 또한 호스트에서 호스트 복구가 활성화되고 자동 배치가 비활성화됩니다.

```
aws ec2 allocate-hosts --instance-family "m5" --availability-zone "us-east-1a" --auto-
placement "off" --host-recovery "on" --quantity 1
```

다음 명령은 eu-west-1a 가용 영역에서 대상 지정되지 않은 m4.large 인스턴스를 지원하는 전용 호스트를 할당하고 호스트 복구를 활성화하며 키 purpose 및 값 production과 함께 태그를 적용합니다.

```
aws ec2 allocate-hosts --instance-type "m4.large" --availability-zone "eu-west-1a"
 --auto-placement "on" --host-recovery "on" --quantity 1 --tag-specifications
 'ResourceType=dedicated-host,Tags=[{Key=purpose,Value=production}]'
```

PowerShell

전용 호스트를 할당하려면

New-EC2Host Windows PowerShell용 AWS 도구 명령을 사용합니다. 다음 명령은 us-east-1a 가용 영역에서 m5 인스턴스 패밀리의 여러 인스턴스 유형을 지원하는 전용 호스트를 할당합니다. 또한 호스트에서 호스트 복구가 활성화되고 자동 배치가 비활성화됩니다.

```
PS C:\> New-EC2Host -InstanceFamily m5 -AvailabilityZone us-east-1a -AutoPlacement Off
 -HostRecovery On -Quantity 1
```

다음 명령은 eu-west-1a 가용 영역에서 대상 지정되지 않은 m4.large 인스턴스를 지원하는 전용 호스트를 할당하고 호스트 복구를 활성화하며 키 purpose 및 값 production과 함께 태그를 적용합니다.

생성 시 전용 호스트를 태그 지정하는 데 사용된 TagSpecification 파라미터를 사용하려면 태그 지정될 리소스 유형, 태그 키 및 태그 값을 지정하는 객체가 필요합니다. 다음 명령은 필요 객체를 생성합니다.

```
PS C:\> $tag = @{ Key="purpose"; Value="production" }
PS C:\> $tagspec = new-object Amazon.EC2.Model.TagSpecification
PS C:\> $tagspec.ResourceType = "dedicated-host"
PS C:\> $tagspec.Tags.Add($tag)
```

다음 명령은 전용 호스트를 할당하고 $tagspec 객체에 지정된 태그를 적용합니다.

```
PS C:\> New-EC2Host -InstanceType m4.large -AvailabilityZone eu-west-1a -
AutoPlacement On -HostRecovery On -Quantity 1 -TagSpecification $tagspec
```

전용 호스트에서 인스턴스 시작

전용 호스트를 할당한 후 여기에서 인스턴스를 시작할 수 있습니다. 시작하려는 인스턴스 유형에 사용 가능한 충분한 용량을 갖춘 활성 전용 호스트가 없는 경우 host 테넌시의 인스턴스를 시작할 수 없습니다.

> **Note**
>
> 전용 호스트에서 시작되는 인스턴스는 VPC에서만 시작할 수 있습니다. 자세한 정보는 VPC 소개 단원을 참조하십시오.

인스턴스를 시작하기 전에 제한 사항에 유의하십시오. 자세한 내용은 전용 호스트 제한 사항 (p. 383) 단원을 참조하십시오.

다음 방법을 사용하여 전용 호스트에서 인스턴스를 시작할 수 있습니다.

Console

전용 호스트 페이지의 특정 전용 호스트에서 인스턴스를 시작하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 전용 호스트를 선택합니다.
3. 전용 호스트 페이지에서 호스트를 선택하고 작업, 호스트에서 인스턴스 시작을 선택합니다.
4. 목록에서 AMI를 선택합니다. Windows, SUSE 및 Amazon EC2에서 제공하는 RHEL AMI는 전용 호스트와 함께 사용할 수 없습니다.
5. 인스턴스 유형 선택 페이지에서 시작할 인스턴스 유형을 선택하고 다음: 인스턴스 세부 정보 구성을 선택합니다.

   전용 호스트가 단일 인스턴스 유형만 지원하는 경우 지원되는 인스턴스 유형이 기본적으로 선택되고 변경할 수 없습니다.

   전용 호스트가 여러 인스턴스 유형을 지원하는 경우 전용 호스트의 가용 인스턴스 용량을 기반으로 지원되는 인스턴스 패밀리 내에서 인스턴스 유형을 선택해야 합니다. 크기가 큰 인스턴스부터 시작한 다음 필요에 따라 더 작은 인스턴스로 나머지 인스턴스 용량을 채우는 것이 좋습니다.
6. 인스턴스 세부 정보 구성 페이지에서 필요에 따라 인스턴스 설정을 구성하고 선호도에서 다음 옵션 중 하나를 선택합니다.

   - 비활성 - 인스턴스가 지정된 호스트에서 시작하지만, 중지될 경우 반드시 동일한 전용 호스트에서 다시 시작하지는 않습니다.
   - 호스트 - 중지된 경우 인스턴스가 항상 특정 호스트에서 다시 시작합니다.

   선호도에 대한 자세한 내용은 자동 배치 및 선호도의 이해 (p. 390) 단원을 참조하십시오.

   테넌시 및 호스트 옵션은 선택한 호스트에 따라 사전 구성됩니다.
7. 검토 및 시작을 선택합니다.
8. 인스턴스 시작 검토 페이지에서 시작을 선택합니다.
9. 메시지가 표시될 때 기존 키 페어를 선택하거나 새 키 페어를 생성한 후 인스턴스 시작을 선택합니다.

시작 인스턴스 마법사를 사용하여 전용 호스트에서 인스턴스를 시작하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스, 인스턴스 시작을 선택합니다.
3. 목록에서 AMI를 선택합니다. Windows, SUSE 및 Amazon EC2에서 제공하는 RHEL AMI는 전용 호스트와 함께 사용할 수 없습니다.
4. 시작할 인스턴스 유형을 선택하고 Next: Configure Instance Details(다음: 인스턴스 세부 정보 구성)을 클릭합니다.
5. Configure Instance Details(인스턴스 세부 정보 구성) 페이지에서 필요에 따라 인스턴스 설정을 구성한 후 다음과 같은 전용 호스트 설정을 구성합니다.

   - 테넌시 - 전용 호스트 - 전용 호스트에서 이 인스턴스 시작을 선택합니다.
   - 호스트 - 자동 배치 사용을 선택하여 자동 배치가 활성화된 전용 호스트에서 인스턴스를 시작하거나, 목록에서 특정 전용 호스트를 선택합니다. 이 목록에는 선택한 인스턴스 유형을 지원하는 전용 호스트만 표시됩니다.
   - 선호도 - 다음 옵션 중 하나를 선택합니다.
     - 비활성 - 인스턴스가 지정된 호스트에서 시작하지만, 중지될 경우 반드시 그 호스트에서 다시 시작하지는 않습니다.
     - 호스트 - 중지된 경우 인스턴스가 항상 지정된 호스트에서 다시 시작합니다.

자세한 내용은 자동 배치 및 선호도의 이해 (p. 390) 단원을 참조하십시오.

이러한 설정이 보이지 않으면 네트워크 메뉴에서 VPC를 선택했는지 확인하십시오.

6. 검토 및 시작을 선택합니다.
7. 인스턴스 시작 검토 페이지에서 시작을 선택합니다.
8. 메시지가 표시될 때 기존 키 페어를 선택하거나 새 키 페어를 생성한 후 인스턴스 시작을 선택합니다.

AWS CLI

전용 호스트에서 인스턴스를 시작하려면

run-instances AWS CLI 명령을 사용하고 `Placement` 요청 파라미터 내에서 인스턴스 선호도, 테넌시 및 호스트를 지정합니다.

PowerShell

전용 호스트에서 인스턴스를 시작하려면

New-EC2Instance Windows PowerShell용 AWS 도구 명령을 사용하고 `Placement` 요청 파라미터 내에서 인스턴스 선호도, 테넌시 및 호스트를 지정합니다.

호스트 리소스 그룹에서 인스턴스 시작

가용 인스턴스 용량이 있는 전용 호스트가 포함된 호스트 리소스 그룹에서 인스턴스를 시작하면 Amazon EC2가 그 호스트에서 인스턴스를 시작합니다. 가용 인스턴스 용량이 있는 호스트가 호스트 리소스 그룹에 없으면 Amazon EC2가 호스트 리소스 그룹에 새 호스트를 자동으로 할당하고 그 호스트에서 인스턴스를 시작합니다. 자세한 내용은 AWS License Manager 사용 설명서의 호스트 리소스 그룹을 참조하십시오.

다음 방법을 사용하여 호스트 리소스 그룹에서 인스턴스를 시작할 수 있습니다.

Console

호스트 리소스 그룹에서 인스턴스를 시작하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스, 인스턴스 시작을 선택합니다.
3. 목록에서 AMI를 선택합니다. Windows, SUSE 및 Amazon EC2에서 제공하는 RHEL AMI는 전용 호스트와 함께 사용할 수 없습니다.
4. 시작할 인스턴스 유형을 선택하고 Next: Configure Instance Details(다음: 인스턴스 세부 정보 구성)을 클릭합니다.
5. 인스턴스 세부 정보 구성 페이지에서 필요에 따라 인스턴스 설정을 구성한 후 다음을 수행합니다.

   a. 테넌시에서 전용 호스트를 설정합니다.
   b. Host resource group(호스트 리소스 그룹)에서 Launch instance into a host resource group(호스트 리소스 그룹에서 인스턴스 시작)을 선택합니다.
   c. Host resource group name(호스트 리소스 그룹 이름)에서 인스턴스를 시작할 호스트 리소스 그룹을 선택합니다.

   호스트 ID를 선택하여 특정 호스트를 대상으로 지정할 수 없으며, 호스트 리소스 그룹에서 인스턴스를 시작할 때 인스턴스 선호도를 활성화할 수 없습니다.
6. 검토 및 시작을 선택합니다.

7. 인스턴스 시작 검토 페이지에서 시작을 선택합니다.
8. 메시지가 표시될 때 기존 키 페어를 선택하거나 새 키 페어를 생성한 후 인스턴스 시작을 선택합니다.

AWS CLI

호스트 리소스 그룹에서 인스턴스를 시작하려면

run-instances AWS CLI 명령을 사용하고 `Placement` 요청 파라미터 내에서 테넌시 옵션은 생략하고 호스트 리소스 그룹 ARN을 지정합니다.

PowerShell

호스트 리소스 그룹에서 인스턴스를 시작하려면

New-EC2Instance Windows PowerShell용 AWS 도구 명령을 사용하고 `Placement` 요청 파라미터 내에서 테넌시 옵션은 생략하고 호스트 리소스 그룹 ARN을 지정합니다.

자동 배치 및 선호도의 이해

전용 호스트에 대한 배치 제어는 인스턴스 수준과 호스트 수준에서 모두 수행됩니다.

자동 배치

자동 배치는 호스트 수준에서 구성됩니다. 자동 배치에서는 시작하는 인스턴스가 특정 호스트에서 시작되는지 또는 일치하는 구성이 있는 모든 가용 호스트에서 시작되는지를 선택할 수 있습니다.

전용 호스트의 자동 배치가 비활성화된 경우에는 고유한 호스트 ID를 지정한 호스트 테넌시 인스턴스 시작만 허용합니다. 이것이 새로운 전용 호스트에 대한 기본 설정입니다.

전용 호스트의 자동 배치가 활성화된 경우에는, 인스턴스 유형 구성이 일치하며 대상 지정되지 않은 인스턴스 시작을 허용합니다.

인스턴스를 시작할 때 테넌시를 구성해야 합니다. 특정 `HostId`를 입력하지 않고 전용 호스트에서 인스턴스를 시작하면 자동 배치가 활성화되고 인스턴스 유형이 일치하는 모든 전용 호스트에서 시작할 수 있습니다.

호스트 선호도

호스트 선호도는 인스턴스 수준에서 구성되어야 합니다. 인스턴스와 전용 호스트 간에 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 `Off`로 설정된 상태에서 인스턴스를 중지했다 다시 시작하는 경우 모든 사용 가능한 호스트에서 다시 시작할 수 있습니다. 하지만 인스턴스는 마지막으로 실행되던 전용 호스트에서 다시 시작하려고 시도합니다(최대한 노력).

전용 호스트 자동 배치 수정

AWS 계정에 할당한 후 다음 방법 중 하나를 사용하여 전용 호스트의 자동 배치 설정을 수정할 수 있습니다.

새로운 콘솔

전용 호스트의 자동 배치를 수정하려면

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

2. 탐색 창에서 전용 호스트를 선택합니다.
3. 호스트를 선택하고 작업, Modify host(호스트 수정)를 선택합니다.
4. Instance auto-placement(인스턴스 자동 배치)에서 활성화를 선택하여 자동 배치를 활성화하거나 활성화를 선택 취소하여 자동 배치를 비활성화합니다. 자세한 내용은 자동 배치 및 선호도의 이해 (p. 390) 단원을 참조하십시오.
5. 저장을 선택합니다.

기존 콘솔

전용 호스트의 자동 배치를 수정하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 전용 호스트를 선택합니다.
3. 전용 호스트 페이지에서 호스트를 선택하고 작업을 선택한 후 자동 배치 수정을 선택합니다.
4. 자동 배치 수정 창의 인스턴스 자동 배치 허용에서 예를 선택하면 자동 배치가 활성화되고, 아니요를 선택하면 자동 배치가 비활성화됩니다. 자세한 내용은 자동 배치 및 선호도의 이해 (p. 390) 단원을 참조하십시오.
5. 저장을 선택합니다.

AWS CLI

전용 호스트의 자동 배치를 수정하려면

modify-hosts AWS CLI 명령을 사용합니다. 다음 예는 지정된 전용 호스트에 대한 자동 배치를 활성화합니다.

```
aws ec2 modify-hosts --auto-placement on --host-ids h-012a3456b7890cdef
```

PowerShell

전용 호스트의 자동 배치를 수정하려면

Edit-EC2Host Windows PowerShell용 AWS 도구 명령을 사용합니다. 다음 예는 지정된 전용 호스트에 대한 자동 배치를 활성화합니다.

```
PS C:\> Edit-EC2Host --AutoPlacement 1 --HostId h-012a3456b7890cdef
```

지원되는 인스턴스 유형 수정

전용 호스트를 수정하여 지원되는 인스턴스 유형을 변경할 수 있습니다. 현재 단일 인스턴스 유형을 지원하는 경우 인스턴스 패밀리 내의 여러 인스턴스 유형을 지원하도록 수정할 수 있습니다. 마찬가지로, 현재 여러 인스턴스 유형을 지원하는 경우 특정 인스턴스 유형만 지원하도록 수정할 수 있습니다.

여러 인스턴스 유형을 지원하도록 전용 호스트를 수정하려면 먼저 호스트에서 실행 중인 모든 인스턴스를 중지해야 합니다. 수정을 완료하려면 10분 정도 걸립니다. 수정이 진행 중인 동안 전용 호스트는 `pending` 상태로 전환됩니다. `pending` 상태에 있는 동안에는 전용 호스트에서 중지된 인스턴스를 시작하거나 새 인스턴스를 시작할 수 없습니다. 다음 인스턴스 패밀리는 여러 인스턴스 유형을 지원하도록 수정할 수 있습니다. A1, C5, C5n, M5, M5n, R5, and R5n

여러 인스턴스 유형을 지원하는 전용 호스트를 특정 인스턴스 유형만 지원하도록 수정하려면 호스트에 실행 중인 인스턴스가 없거나 실행 중인 인스턴스가 호스트에서 지원하려는 인스턴스 유형이어야 합니다. 예를 들어, `m5` 인스턴스 패밀리 내의 여러 인스턴스 유형을 지원하는 호스트를 `m5.large` 인스턴스만 지원하도록 수정하려면 전용 호스트에 실행 중인 인스턴스가 없거나 `m5.large` 인스턴스만 실행 중이어야 합니다.

다음 방법 중 하나를 사용하여 지원되는 인스턴스 유형을 수정할 수 있습니다.

새로운 콘솔

전용 호스트에 지원되는 인스턴스 유형을 수정하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 전용 호스트를 선택합니다.
3. 수정할 전용 호스트를 선택하고 작업, Modify host(호스트 수정)을 선택합니다.
4. 전용 호스트의 현재 구성에 따라 다음 중 하나를 수행합니다.
   - 전용 호스트가 현재 특정 인스턴스 유형을 지원하는 경우 Support multiple instance types(여러 인스턴스 유형 지원)이 활성화되지 않으며 인스턴스 유형에 지원되는 인스턴스 유형이 나열됩니다. 현재 인스턴스 패밀리 내의 여러 유형을 지원하도록 호스트를 수정하려면 Support multiple instance types(여러 인스턴스 유형 지원)에서 활성화를 선택합니다.

     여러 인스턴스 유형을 지원하도록 호스트를 수정하려면 먼저 호스트에서 실행 중인 모든 인스턴스를 중지해야 합니다.
   - 전용 호스트가 현재 인스턴스 패밀리의 여러 인스턴스 유형을 지원하는 경우 Support multiple instance types(여러 인스턴스 유형 지원)에서 활성화가 선택됩니다. 특정 인스턴스 유형을 지원하도록 호스트를 수정하려면 Support multiple instance types(여러 인스턴스 유형 지원)에서 활성화를 선택 취소한 다음 인스턴스 유형에서 지원할 특정 인스턴스 유형을 선택합니다.

     전용 호스트에서 지원되는 인스턴스 패밀리는 변경할 수 없습니다.
5. 저장을 선택합니다.

기존 콘솔

전용 호스트에 지원되는 인스턴스 유형을 수정하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 전용 호스트를 선택합니다.
3. 수정할 전용 호스트를 선택하고 작업, 지원되는 인스턴스 유형 수정을 선택합니다.
4. 전용 호스트의 현재 구성에 따라 다음 중 하나를 수행합니다.
   - 전용 호스트가 현재 특정 인스턴스 유형을 지원하는 경우 여러 인스턴스 유형 지원에서 아니요가 선택됩니다. 현재 인스턴스 패밀리 내의 여러 유형을 지원하도록 호스트를 수정하려면 여러 인스턴스 유형 지원에서 예를 선택합니다.

     여러 인스턴스 유형을 지원하도록 호스트를 수정하려면 먼저 호스트에서 실행 중인 모든 인스턴스를 중지해야 합니다.
   - 전용 호스트가 현재 인스턴스 패밀리의 여러 인스턴스 유형을 지원하는 경우 여러 인스턴스 유형 지원에서 예가 선택되고 인스턴스 패밀리에 지원되는 인스턴스 패밀리가 표시됩니다. 특정 인스턴스 유형을 지원하도록 호스트를 수정하려면 여러 인스턴스 유형 지원에서 아니요를 선택한 다음 인스턴스 유형에서 지원할 특정 인스턴스 유형을 선택합니다.

     전용 호스트에서 지원되는 인스턴스 패밀리는 변경할 수 없습니다.
5. Save를 선택합니다.

AWS CLI

전용 호스트에 지원되는 인스턴스 유형을 수정하려면

modify-hosts AWS CLI 명령을 사용합니다.

다음 명령은 `m5` 인스턴스 패밀리 내의 여러 인스턴스 유형을 지원하도록 전용 호스트를 수정합니다.

```
aws ec2 modify-hosts --instance-family m5 --host-ids h-012a3456b7890cdef
```

다음 명령은 `m5.xlarge` 인스턴스만 지원하도록 전용 호스트를 수정합니다.

```
aws ec2 modify-hosts --instance-type m5.xlarge --instance-family --host-
ids h-012a3456b7890cdef
```

PowerShell

전용 호스트에 지원되는 인스턴스 유형을 수정하려면

Edit-EC2Host Windows PowerShell용 AWS 도구 명령을 사용합니다.

다음 명령은 `m5` 인스턴스 패밀리 내의 여러 인스턴스 유형을 지원하도록 전용 호스트를 수정합니다.

```
PS C:\> Edit-EC2Host --InstanceFamily m5 --HostId h-012a3456b7890cdef
```

다음 명령은 `m5.xlarge` 인스턴스만 지원하도록 전용 호스트를 수정합니다.

```
PS C:\> Edit-EC2Host --InstanceType m5.xlarge --HostId h-012a3456b7890cdef
```

인스턴스 테넌시 및 선호도 수정

시작한 이후에는 인스턴스의 테넌시를 `dedicated`에서 `host`로, 또는 `host`에서 `dedicated`로 변경할 수 있습니다. 또한 인스턴스와 호스트 간의 선호도를 수정할 수 있습니다. 인스턴스 테넌시 또는 선호도를 수정하려면 인스턴스가 `stopped` 상태여야 합니다.

다음 방법을 사용하여 인스턴스의 테넌시와 선호도를 수정할 수 있습니다.

Console

인스턴스 테넌시 또는 선호도를 수정하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 인스턴스를 선택한 후 수정할 인스턴스를 선택합니다.
3. 작업, 인스턴스 상태, 중지를 차례로 선택합니다.
4. 인스턴스에 대한 컨텍스트(오른쪽 클릭) 메뉴를 열고 인스턴스 설정, 인스턴스 배치 수정을 차례로 선택합니다.
5. 인스턴스 배치 수정 페이지에서 다음을 구성합니다.
   - 테넌시 - 다음 중 하나를 선택합니다.
     - 전용 하드웨어 인스턴스를 실행 - 인스턴스를 전용 인스턴스로 시작합니다. 자세한 내용은 전용 인스턴스 (p. 410) 단원을 참조하십시오.
     - 전용 호스트에서 인스턴스 시작 - 구성 가능한 선호도가 있는 전용 호스트에서 인스턴스를 시작합니다.
   - 선호도—다음 중 하나를 선택합니다.
     - 내 호스트 중 하나에서 이 인스턴스를 시작할 수 있음—인스턴스 유형을 지원하는 계정의 모든 가용한 전용 호스트에서 인스턴스를 시작합니다.
     - 선택한 호스트에서만 이 인스턴스를 실행할 수 있음—대상 호스트로 선택된 전용 호스트에서만 인스턴스를 실행할 수 있습니다.
   - 대상 호스트—인스턴스를 실행해야 하는 전용 호스트를 선택합니다. 대상 호스트 목록이 표시되지 않는 경우 계정에 사용 가능하며 호환되는 전용 호스트가 없을 수 있습니다.

자세한 내용은 자동 배치 및 선호도의 이해 (p. 390) 단원을 참조하십시오.

6. Save를 선택합니다.

AWS CLI

인스턴스 테넌시 또는 선호도를 수정하려면

modify-instance-placement AWS CLI 명령을 사용합니다. 다음 예는 지정된 인스턴스의 선호도를 `default`에서 `host`로 변경하고 인스턴스가 선호도를 갖는 전용 호스트를 지정합니다.

```
aws ec2 modify-instance-placement --instance-id i-1234567890abcdef0 --affinity host --
host-id h-012a3456b7890cdef
```

PowerShell

인스턴스 테넌시 또는 선호도를 수정하려면

Edit-EC2InstancePlacement Windows PowerShell용 AWS 도구 명령을 사용합니다. 다음 예는 지정된 인스턴스의 선호도를 `default`에서 `host`로 변경하고 인스턴스가 선호도를 갖는 전용 호스트를 지정합니다.

```
PS C:\> Edit-EC2InstancePlacement -InstanceId i-1234567890abcdef0 -Affinity host -
HostId h-012a3456b7890cdef
```

전용 호스트 보기

다음 방법을 사용하여 전용 호스트 및 개별 인스턴스의 세부 정보를 볼 수 있습니다.

새로운 콘솔

전용 호스트의 세부 정보를 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 전용 호스트를 선택합니다.
3. 전용 호스트페이지에서 호스트를 선택합니다.
4. 호스트 정보를 보려면 세부 정보를 선택합니다.

   사용 가능한 vCPU는 새 인스턴스를 시작하기 위해 전용 호스트에서 사용할 수 있는 vCPU를 나타냅니다. 예를 들어, `c5` 인스턴스 패밀리 내의 여러 인스턴스 유형을 지원하고 실행 중인 인스턴스가 없는 전용 호스트에는 사용 가능한 vCPU가 72개 있습니다. 따라서 사용 가능한 vCPU 72개를 활용하여 다양한 조합의 인스턴스 유형을 전용 호스트에서 시작할 수 있습니다.

   호스트에서 실행 중인 인스턴스의 정보를 보려면 실행 중인 인스턴스를 선택합니다.

기존 콘솔

전용 호스트의 세부 정보를 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 전용 호스트를 선택합니다.
3. 전용 호스트페이지에서 호스트를 선택합니다.
4. 호스트 정보를 보려면 설명을 선택합니다. 사용 가능한 vCPU는 새 인스턴스를 시작하기 위해 전용 호스트에서 사용할 수 있는 vCPU를 나타냅니다. 예를 들어, `c5` 인스턴스 패밀리 내의 여러 인스턴

스 유형을 지원하고 실행 중인 인스턴스가 없는 전용 호스트에는 사용 가능한 vCPU가 72개 있습니다. 따라서 사용 가능한 vCPU 72개를 활용하여 다양한 조합의 인스턴스 유형을 전용 호스트에서 시작할 수 있습니다.

호스트에서 실행 중인 인스턴스의 정보를 보려면 인스턴스를 선택합니다.

AWS CLI

전용 호스트의 용량을 보려면

describe-hosts AWS CLI 명령을 사용합니다.

다음 예제에서는 describe-hosts(AWS CLI) 명령을 사용하여 `c5` 인스턴스 패밀리 내의 여러 인스턴스 유형을 지원하는 전용 호스트에 사용 가능한 인스턴스 용량을 봅니다. 전용 호스트에는 이미 실행 중인 `c5.4xlarge` 인스턴스 2개와 `c5.2xlarge` 인스턴스 4개가 있습니다.

```
$  aws ec2 describe-hosts --host-id h-012a3456b7890cdef
```

```
"AvailableInstanceCapacity": [
    { "AvailableCapacity": 2,
      "InstanceType": "c5.xlarge",
      "TotalCapacity": 18 },
    { "AvailableCapacity": 4,
      "InstanceType": "c5.large",
      "TotalCapacity": 36 }
  ],
"AvailableVCpus": 8
```

PowerShell

전용 호스트의 인스턴스 용량을 보려면

Get-EC2Host Windows PowerShell용 AWS 도구 명령을 사용합니다.

```
PS C:\> Get-EC2Host -HostId h-012a3456b7890cdef
```

전용 호스트 태그 지정

기존 전용 호스트에 사용자 지정 태그를 할당하여 용도, 소유자, 환경 등 다양한 방식으로 주소를 분류할 수 있습니다. 그러면 할당한 사용자 지정 태그를 기반으로 특정 전용 호스트를 빠르게 찾을 수 있습니다. 전용 호스트 태그는 비용 할당 추적에만 사용할 수 있습니다.

또한 생성 시 전용 호스트 볼륨에 태그를 적용할 수도 있습니다. 자세한 내용은 전용 호스트 할당 (p. 385) 단원을 참조하십시오.

다음 방법을 사용하여 전용 호스트에 태그를 지정할 수 있습니다.

새로운 콘솔

전용 호스트에 태그를 지정하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 전용 호스트를 선택합니다.
3. 태그 지정할 전용 호스트를 선택하고 작업, 태그 관리를 선택합니다.
4. 태그 관리 화면에서 태그 추가를 선택한 후 태그에 키와 값을 지정합니다.
5. (선택 사항) 태그 추가를 선택하여 전용 호스트에 태그를 추가합니다.
6. [Save changes]를 선택합니다.

기존 콘솔

전용 호스트에 태그를 지정하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 전용 호스트를 선택합니다.
3. 태그 지정할 전용 호스트를 선택하고 태그를 선택합니다.
4. 태그 추가/편집을 선택합니다.
5. 태그 추가/편집 대화 상자에서 태그 생성을 선택하고 태그에 대한 키와 값을 지정합니다.
6. (선택 사항) 추가 태그를 전용 호스트에 추가하려면 태그 생성을 선택합니다.
7. 저장을 선택합니다.

AWS CLI

전용 호스트에 태그를 지정하려면

create-tags AWS CLI 명령을 사용합니다.

다음 명령은 지정된 전용 호스트에 `Owner=TeamA` 태그를 지정합니다.

```
aws ec2 create-tags --resources h-abc12345678909876 --tags Key=Owner,Value=TeamA
```

PowerShell

전용 호스트에 태그를 지정하려면

New-EC2Tag Windows PowerShell용 AWS 도구 명령을 사용합니다.

`New-EC2Tag` 명령에는 전용 호스트 태그에 사용할 키-값 페어를 지정하는 `Tag` 객체가 필요합니다. 다음 명령은 키와 값이 각각 `Tag` , `$tag`인 `Owner`라는 이름의 `TeamA` 객체를 만듭니다.

```
PS C:\> $tag = New-Object Amazon.EC2.Model.Tag
PS C:\> $tag.Key = "Owner"
PS C:\> $tag.Value = "TeamA"
```

다음 명령은 지정된 전용 호스트에 `$tag` 객체를 지정합니다.

```
PS C:\> New-EC2Tag -Resource h-abc12345678909876 -Tag $tag
```

전용 호스트 모니터링

Amazon EC2는 전용 호스트의 상태를 지속적으로 모니터링합니다. 업데이트는 Amazon EC2 콘솔에서 전달됩니다. 다음 방법을 사용하여 전용 호스트에 대한 정보를 볼 수 있습니다.

Console

전용 호스트의 상태를 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 전용 호스트를 선택합니다.
3. 목록에서 전용 호스트를 찾아서 상태 열의 값을 검토합니다.

AWS CLI

전용 호스트의 상태를 보려면

describe-hosts AWS CLI 명령을 사용하고 `hostSet` 응답 요소의 `state` 속성을 검토합니다.

```
aws ec2 describe-hosts --host-id h-012a3456b7890cdef
```

PowerShell

전용 호스트의 상태를 보려면

Get-EC2Host Windows PowerShell용 AWS 도구 명령을 사용하고 `hostSet` 응답 요소의 `state` 속성을 검토합니다.

```
PS C:\> Get-EC2Host -HostId h-012a3456b7890cdef
```

다음 표는 가능한 전용 호스트 상태를 설명합니다.

상태	설명
`available`	AWS가 전용 호스트에 대한 문제를 감지하지 못했습니다. 유지 관리 또는 수리가 예정되어 있지 않습니다. 이 전용 호스트에서 인스턴스를 시작할 수 있습니다.
`released`	전용 호스트가 해제되었습니다. 더 이상 이 호스트 ID가 사용되지 않습니다. 해제된 호스트는 다시 사용할 수 없습니다.
`under-assessment`	AWS가 전용 호스트에 있을 수 있는 문제를 탐색 중입니다. 작업이 필요할 경우 AWS Management 콘솔 또는 이메일을 통해 통보됩니다. 이 상태에서는 전용 호스트에서 인스턴스를 시작할 수 없습니다.
`pending`	새 인스턴스를 시작하기 위해 전용 호스트를 사용할 수 없습니다. 이 호스트는 여러 인스턴스 유형을 지원하도록 수정 (p. 391)되고 있거나, 호스트 복구 (p. 404)가 진행 중입니다.
`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. AWS는 이 상태에 있는 인스턴스는 다시 시작하지 않습니다. 이 상태에서는 전용 호스트에서 인스턴스를 시작할 수 없습니다.
`released-permanent-failure`	AWS에서는 오류가 발생한 전용 호스트를 영구 해제하여 더 이상 인스턴스가 실행되지 못하도록 합니다. 전용 호스트 ID를 더 이상 사용할 수 없습니다.

전용 호스트 해제

전용 호스트에서 실행되는 모든 인스턴스를 중지해야 해당 호스트를 해제할 수 있습니다. 이 인스턴스들을 계정의 다른 전용 호스트로 마이그레이션하여 계속 사용할 수 있습니다. 이 단계들은 온디맨드 전용 호스트에만 적용됩니다.

다음 방법을 사용하여 전용 호스트를 해제할 수 있습니다.

새로운 콘솔

전용 호스트를 해제하려면

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

3. 전용 호스트 페이지에서 해제할 전용 호스트를 선택합니다.
4. 작업, Release host(호스트 릴리스)를 선택합니다.
5. 릴리스를 선택하여 확인합니다.

기존 콘솔

전용 호스트를 해제하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 전용 호스트를 선택합니다.
3. 전용 호스트 페이지에서 해제할 전용 호스트를 선택합니다.
4. 작업, 호스트 릴리스를 선택합니다.
5. 릴리스를 선택하여 확인합니다.

AWS CLI

전용 호스트를 해제하려면

release-hosts AWS CLI 명령을 사용합니다.

```
aws ec2 release-hosts --host-ids h-012a3456b7890cdef
```

PowerShell

전용 호스트를 해제하려면

Remove-EC2Hosts Windows PowerShell용 AWS 도구 명령을 사용합니다.

```
PS C:\> Remove-EC2Hosts -HostId h-012a3456b7890cdef
```

전용 호스트를 해제한 후에는 동일한 호스트 또는 호스트 ID를 다시 사용할 수 없으며, 더 이상 해당 호스트에 대해 온디맨드 결제 요금이 부과되지 않습니다. 전용 호스트의 상태가 `released`로 변경되고 이 호스트에서 인스턴스를 시작할 수 없게 됩니다.

Note

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 인스턴스 페이지에 나열됩니다. 또한 `host` 테넌시 설정을 유지합니다.

전용 호스트 예약 구입

다음과 같은 방법으로 예약을 구매할 수 있습니다.

Console

예약을 구매하려면

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

2. 전용 호스트, 전용 호스트 예약, 전용 호스트 예약 구매를 선택합니다.
3. 전용 호스트 예약 구매 화면에서 기본 설정을 사용하여 이용 가능한 상품을 검색하거나 다음에 대한 사용자 지정 값을 지정할 수 있습니다.
   - 호스트 인스턴스 패밀리—나열되는 옵션은 계정에서 예약에 할당되지 않은 전용 호스트에 해당합니다.
   - 가용 영역—계정에서 예약에 할당되지 않은 전용 호스트의 가용 영역입니다.
   - 결제 옵션—상품에 대한 결제 방식입니다.
   - 기간—1년 또는 3년이 될 수 있는 예약 기간입니다.
4. 상품 찾기를 선택하고 요건에 맞는 상품을 선택합니다.
5. 예약에 연결할 전용 호스트를 선택하고 검토를 선택합니다.
6. 주문을 검토한 후 주문을 선택합니다.

AWS CLI

예약을 구매하려면

1. describe-host-reservation-offerings AWS CLI 명령을 사용하여 요구 사항에 맞는 사용 가능한 상품 목록을 나열합니다. 다음 예는 `m4` 인스턴스 패밀리의 인스턴스를 지원하고 사용 기간이 1년인 상품 목록을 나열합니다.

   Note

   기간은 초 단위로 지정됩니다. 1년 기간에는 31,536,000초가 포함되고, 3년 기간에는 94,608,000초가 포함됩니다.

```
aws ec2 describe-host-reservation-offerings --filter Name=instance-family,Values=m4
 --max-duration 31536000
```

   이 명령은 조건과 일치하는 상품 목록을 반환합니다. 구입할 상품의 `offeringId`를 기록하십시오.
2. purchase-host-reservation AWS CLI 명령을 사용하여 상품을 구입하고 이전 단계에서 기록한 `offeringId`를 입력합니다. 다음 예는 특정 예약을 구입하고 이미 AWS 계정에 할당된 특정 전용 호스트와 연결합니다.

```
aws ec2 purchase-host-reservation --offering-id hro-03f707bf363b6b324 --host-id-
set h-013abcd2a00cbd123
```

PowerShell

예약을 구매하려면

1. Get-EC2HostReservationOffering Windows PowerShell용 AWS 도구 명령을 사용하여 요구 사항에 맞는 사용 가능한 상품 목록을 나열합니다. 다음 예는 `m4` 인스턴스 패밀리의 인스턴스를 지원하고 사용 기간이 1년인 상품 목록을 나열합니다.

   Note

   기간은 초 단위로 지정됩니다. 1년 기간에는 31,536,000초가 포함되고, 3년 기간에는 94,608,000초가 포함됩니다.

```
PS C:\> $filter = @{Name="instance-family"; Value="m4"}
```

```
PS C:\> Get-EC2HostReservationOffering -filter $filter -MaxDuration 31536000
```

이 명령은 조건과 일치하는 상품 목록을 반환합니다. 구입할 상품의 `offeringId`를 기록하십시오.

2. New-EC2HostReservation Windows PowerShell용 AWS 도구 명령을 사용하여 상품을 구입하고 이전 단계에서 기록한 `offeringId`를 입력합니다. 다음 예는 특정 예약을 구입하고 이미 AWS 계정에 할당된 특정 전용 호스트와 연결합니다.

```
PS C:\> New-EC2HostReservation -OfferingId hro-03f707bf363b6b324 -
HostIdSet h-013abcd2a00cbd123
```

전용 호스트 예약 보기

다음과 같은 예약과 관련된 전용 호스트 정보를 볼 수 있습니다.

- 예약 기간
- 결제 옵션
- 시작 및 종료 날짜

다음과 같은 방법으로 전용 호스트 예약에 대한 세부 정보를 볼 수 있습니다.

Console

전용 호스트 예약의 세부 정보를 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 전용 호스트를 선택합니다.
3. 전용 호스트 페이지에서 전용 호스트 예약을 선택하고 제공된 목록에서 해당 예약을 선택합니다.
4. 예약에 대한 정보를 보려면 세부 정보를 선택합니다.
5. 예약이 연결되어 있는 전용 호스트에 대한 정보를 보려면 호스트를 선택합니다.

AWS CLI

전용 호스트 예약의 세부 정보를 보려면

describe-host-reservations AWS CLI 명령을 사용합니다.

```
aws ec2 describe-host-reservations
```

PowerShell

전용 호스트 예약의 세부 정보를 보려면

Get-EC2HostReservation Windows PowerShell용 AWS 도구 명령을 사용합니다.

```
PS C:\> Get-EC2HostReservation
```

전용 호스트 예약 태그 지정

전용 호스트 예약에 사용자 지정 태그를 할당하여 예약을 용도, 소유자, 환경 등 다양한 방식으로 분류할 수 있습니다. 그러면 할당한 사용자 지정 태그를 기반으로 특정 전용 호스트 예약를 빠르게 찾을 수 있습니다.

명령줄 도구만 사용하여 전용 호스트 예약에 태그를 지정할 수 있습니다.

AWS CLI

전용 호스트 예약에 태그를 지정하려면

create-tags AWS CLI 명령을 사용합니다.

```
aws ec2 create-tags --resources hr-1234563a4ffc669ae --tags Key=Owner,Value=TeamA
```

PowerShell

전용 호스트 예약에 태그를 지정하려면

New-EC2Tag Windows PowerShell용 AWS 도구 명령을 사용합니다.

`New-EC2Tag` 명령에는 전용 호스트 예약 태그에 사용할 키-값 페어를 지정하는 `Tag` 파라미터가 필요합니다. 다음 명령은 `Tag` 파라미터를 생성합니다.

```
PS C:\> $tag = New-Object Amazon.EC2.Model.Tag
PS C:\> $tag.Key = "Owner"
PS C:\> $tag.Value = "TeamA"
```

```
PS C:\> New-EC2Tag -Resource hr-1234563a4ffc669ae -Tag $tag
```

공유된 전용 호스트 작업

전용 호스트 공유를 통해 전용 호스트 소유자는 전용 호스트를 다른 AWS 계정과 공유하거나 AWS 조직 내에서 공유할 수 있습니다. 이를 통해 전용 호스트를 중앙에서 생성 및 관리하고 전용 호스트를 여러 AWS 계정 또는 AWS 조직 내에서 공유할 수 있습니다.

이 모델에서 전용 호스트를 소유한 AWS 계정(소유자)이 다른 AWS 계정(소비자)과 전용 호스트를 공유합니다. 소비자는 자신의 계정에서 할당한 전용 호스트에서 인스턴스를 시작할 때와 동일한 방식으로 공유된 전용 호스트에서 인스턴스를 시작할 수 있습니다. 소유자는 전용 호스트 및 거기서 시작한 인스턴스를 관리할 책임이 있습니다. 소유자는 소비자가 공유된 전용 호스트에서 시작하는 인스턴스를 수정할 수 없습니다. 소비자는 공유된 전용 호스트에서 시작한 인스턴스를 관리할 책임이 있습니다. 소비자는 다른 소비자나 전용 호스트 소유자가 소유한 인스턴스를 보거나 수정할 수 없으며 공유된 전용 호스트를 수정할 수 없습니다.

전용 호스트 소유자는 다음 상대와 전용 호스트를 공유할 수 있습니다.

- AWS 조직 내부 또는 외부의 특정 AWS 계정
- AWS 조직 내부의 조직 단위
- 전체 AWS 조직

목차

- 전용 호스트 공유를 위한 사전 조건 (p. 402)
- 관련 서비스 (p. 402)
- 여러 가용 영역 공유 (p. 402)
- 전용 호스트 공유 (p. 402)
- 공유된 전용 호스트의 공유 해제 (p. 403)
- 공유된 전용 호스트 식별 (p. 403)
- 공유된 전용 호스트에서 실행되는 인스턴스 보기 (p. 403)
- 공유된 전용 호스트 권한 (p. 404)

- 결제 및 측정 (p. 404)
- 전용 호스트 제한 (p. 404)
- 호스트 복구 및 전용 호스트 공유 (p. 404)

전용 호스트 공유를 위한 사전 조건

- 전용 호스트를 공유하려면 전용 호스트를 AWS 계정에 소유하고 있어야 합니다. 자신과 공유된 전용 호스트는 공유할 수 없습니다.
- AWS 조직 또는 AWS 조직의 조직 단위와 전용 호스트를 공유하려면 AWS Organizations로 공유를 활성화해야 합니다. 자세한 내용은 AWS RAM 사용 설명서의 AWS Organizations을 사용하여 공유 사용 설정을 참조하십시오.

관련 서비스

AWS 리소스 액세스 관리자

전용 호스트 공유는 AWS 리소스 액세스 관리자(AWS RAM)와 통합됩니다. AWS RAM은 어떤 AWS 계정이나 AWS Organizations를 통해 AWS 리소스를 공유하도록 해주는 서비스입니다. AWS RAM을 사용하여 리소스 공유로 생성한 사용자 소유 리소스를 공유할 수 있습니다. 리소스 공유는 공유할 리소스와 공유 대상 소비자를 지정합니다. 소비자는 개인 AWS 계정 또는 조직 단위 또는 AWS Organizations의 전체 조직일 수 있습니다.

AWS RAM에 대한 자세한 내용은 AWS RAM 사용 설명서를 참조하십시오.

여러 가용 영역 공유

리전의 가용 영역에 걸쳐 리소스가 배포될 수 있도록 AWS는 각 계정의 이름에 가용 영역을 독립적으로 매핑합니다. 이로 인해 계정 전체에서 가용 영역 이름의 차이가 발생할 수 있습니다. 예를 들어 AWS AWS 계정의 `us-east-1a` 가용 영역은 다른 AWS 계정에 대한 `us-east-1a` 가용 영역과 위치가 동일하지 않을 수 있습니다.

계정과 관련된 전용 호스트의 위치를 확인하려면 가용 영역 ID(AZ ID)를 사용해야 합니다. 가용 영역 ID는 모든 AWS 계정에서 가용 영역의 고유하고 일관된 식별자입니다. 예를 들어 `use1-az1`은 `us-east-1` 리전의 가용 영역 ID이고 모든 AWS 계정에서 동일한 위치입니다.

계정에 속한 가용 영역의 가용 영역 ID를 보려면

1. https://console.aws.amazon.com/ram에서 AWS RAM 콘솔을 엽니다.
2. 현재 리전의 가용 영역 ID는 화면 오른쪽에 있는 Your AZ ID(AZ ID) 패널에 표시됩니다.

전용 호스트 공유

소유자가 전용 호스트를 공유하면 소비자가 호스트에서 인스턴스를 시작할 수 있게 됩니다. 소비자는 가용 용량이 허용하는 만큼 많은 인스턴스를 공유된 호스트에서 시작할 수 있습니다.

자동 배치가 활성화된 상태로 전용 호스트를 공유하는 경우 의도하지 않은 전용 호스트 사용으로 이어질 수 있으므로 다음 내용에 유의하십시오.

- 소비자가 전용 호스트 테넌시로 인스턴스를 시작하고 계정에서 소유한 전용 호스트에 용량이 없으면 인스턴스는 공유된 전용 호스트에서 자동으로 시작됩니다.

전용 호스트를 공유하려면 리소스 공유에 추가해야 합니다. AWS RAM 리소스 공유는 여러 AWS 계정에서 리소스를 공유할 수 있게 해주는 리소스입니다. 리소스 공유는 공유할 리소스와 공유 대상 소비자를 지정합니다. 전용 호스트를 기존 리소스에 추가하거나 새 리소스 공유에 추가할 수 있습니다.

AWS Organizations의 조직에 속해 있고 조직 내의 공유가 활성화되어 있으면 조직의 소비자는 공유된 전용 호스트에 대한 액세스 권한이 자동으로 부여됩니다. 그렇지 않으면 소비자는 리소스 공유에 가입하라는 초대장을 받고 초대를 수락한 후 공유된 전용 호스트의 액세스 권한을 받습니다.

> Note
>
> 전용 호스트를 공유한 후 몇 분이 지나면 소비자가 그에 대한 액세스 권한을 갖게 됩니다.

AWS RAM 콘솔이나 AWS CLI를 사용하여 자신이 소유한 전용 호스트를 공유할 수 있습니다.

AWS RAM 콘솔을 사용하여 소유하고 있는 전용 호스트를 공유하려면

AWS RAM 사용 설명서에서 리소스 공유 만들기를 참조하십시오.

AWS CLI를 사용하여 소유하고 있는 전용 호스트를 공유하려면

create-resource-share 명령을 사용합니다.

공유된 전용 호스트의 공유 해제

전용 호스트 소유자는 언제든지 공유된 전용 호스트를 공유 해제할 수 있습니다. 공유된 전용 호스트를 공유 해제할 때 다음 규칙이 적용됩니다.

- 전용 호스트를 공유한 소비자는 더 이상 새 인스턴스를 이 호스트에서 시작할 수 없습니다.
- 공유를 해제할 때 전용 호스트에서 실행되던 소비자 소유의 인스턴스는 계속 실행되지만 만료되도록 예약됩니다. 소비자는 인스턴스 만료 알림을 받고 2주가 지나면 알림에 대한 조치가 취해집니다. 하지만 전용 호스트가 만료 통지 기간 안에 소비자와 다시 공유되면 인스턴스 만료가 취소됩니다.

소유하고 있는 공유된 전용 호스트를 공유 해제하려면 리소스 공유에서 제거해야 합니다. AWS RAM 콘솔이나 AWS CLI를 사용하여 이 작업을 할 수 있습니다.

AWS RAM 콘솔을 사용하여 소유하고 있는 공유 전용 호스트를 공유 해제하려면

AWS RAM 사용 설명서에서 리소스 공유 업데이트를 참조하십시오.

AWS CLI를 사용하여 소유하고 있는 공유 전용 호스트를 공유 해제하려면

disassociate-resource-share 명령을 사용합니다.

공유된 전용 호스트 식별

소유자와 소비자는 Amazon EC2 콘솔 및 AWS CLI를 사용하여 공유된 전용 호스트를 식별할 수 있습니다.

Amazon EC2 콘솔을 사용하여 공유된 전용 호스트를 식별하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 전용 호스트를 선택합니다. 소유하고 있는 전용 호스트 및 자신과 공유된 전용 호스트가 화면에 나열됩니다. 소유자 열에 전용 호스트 소유자의 AWS 계정 ID가 표시됩니다.

AWS CLI를 사용하여 공유된 전용 호스트를 식별하려면

describe-hosts 명령을 사용합니다. 이 명령은 소유하고 있는 전용 호스트 및 자신과 공유된 전용 호스트를 반환합니다.

공유된 전용 호스트에서 실행되는 인스턴스 보기

소유자와 소비자는 Amazon EC2 콘솔과 AWS CLI를 사용하여 언제든지 공유된 전용 호스트에서 실행되는 인스턴스를 볼 수 있습니다.

Amazon EC2 콘솔을 사용하여 공유된 전용 호스트에서 실행되는 인스턴스를 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 전용 호스트를 선택합니다.
3. 인스턴스를 볼 전용 호스트를 선택하고 인스턴스를 선택합니다. 호스트에서 실행 중인 인스턴스가 탭에 나열됩니다. 소유자는 소비자가 시작한 인스턴스를 포함하여 호스트에서 실행되는 인스턴스를 모두 볼 수 있습니다. 소비자는 자신이 호스트에서 시작하여 실행 중인 인스턴스만 볼 수 있습니다. 소유자 열에 인스턴스를 시작한 계정의 AWS 계정 ID가 표시됩니다.

AWS CLI를 사용하여 공유된 전용 호스트에서 실행되는 인스턴스를 보려면

describe-hosts 명령을 사용합니다. 이 명령은 각 전용 호스트에서 실행되는 인스턴스를 반환합니다. 소유자는 호스트에서 실행되는 인스턴스를 모두 볼 수 있습니다. 소비자는 공유된 호스트에서 자신이 시작하여 실행 중인 인스턴스만 볼 수 있습니다. `InstanceOwnerId`에 인스턴스 소유자의 AWS 계정 ID가 표시됩니다.

공유된 전용 호스트 권한

소유자에 대한 사용 권한

소유자는 공유된 전용 호스트 및 거기서 시작한 인스턴스를 관리할 책임이 있습니다. 소유자는 소비자가 시작한 인스턴스를 포함하여 공유된 전용 호스트에서 실행되는 인스턴스를 모두 볼 수 있습니다. 하지만 소유자는 소비자가 시작하여 실행 중인 인스턴스에 대해 아무 작업도 할 수 없습니다.

소비자를 위한 권한

소비자는 공유된 전용 호스트에서 시작한 인스턴스를 관리할 책임이 있습니다. 소비자는 어떤 식으로든 공유된 전용 호스트를 수정할 수 없으며 다른 소비자 또는 전용 호스트 소유자가 시작한 인스턴스를 보거나 수정할 수 없습니다.

결제 및 측정

전용 호스트 공유에 대한 추가 비용은 없습니다.

공유하는 전용 호스트 비용이 소유자에게 청구됩니다. 공유된 전용 호스트에서 소비자가 시작한 인스턴스 비용을 소비자에게 청구하지 않습니다.

전용 호스트 예약은 공유된 전용 호스트의 결제 할인을 계속 제공합니다. 전용 호스트 소유자만 자신이 소유한 공유 전용 호스트의 전용 호스트 예약을 구매할 수 있습니다.

전용 호스트 제한

공유된 전용 호스트 수는 소유자의 전용 호스트 제한에만 포함됩니다. 소비자의 전용 호스트 제한은 자신과 공유된 전용 호스트의 영향을 받지 않습니다. 마찬가지로 공유된 전용 호스트에서 소비자가 시작하는 인스턴스도 그 인스턴스 제한에 포함되지 않습니다.

호스트 복구 및 전용 호스트 공유

호스트 복구는 전용 호스트 소유자 및 이 호스트가 공유된 소비자가 시작한 인스턴스를 복구합니다. 대체 전용 호스트는 소유자 계정에 할당되며, 원래의 전용 호스트와 같은 리소스 공유에 추가되고 같은 소비자와 공유됩니다.

자세한 내용은 호스트 복구 (p. 404) 단원을 참조하십시오.

호스트 복구

호스트 복구는 전용 호스트에서 오류가 감지되면 새 대체 호스트로 인스턴스를 자동으로 다시 시작합니다. 호스트 복구는 수동 개입의 필요성을 줄이고 예기치 않은 전용 호스트 장애가 발생하는 경우 운영 부담을 줄입니다.

또한 내장된 AWS License Manager 통합 기능은 호스트 복구가 발생하는 경우 라이선스 추적 및 관리를 자동화합니다.

> **Note**
>
> AWS License Manager 통합은 AWS License Manager가 제공되는 지역에서만 지원됩니다.

목차

- 호스트 복구 기본 사항 (p. 405)
- 호스트 복구 구성 (p. 406)
- 호스트 복구 상태 (p. 407)
- 지원되는 인스턴스 유형 (p. 407)
- 지원되지 않는 인스턴스 수동 복구 (p. 408)
- 관련 서비스 (p. 408)
- 요금 (p. 408)

호스트 복구 기본 사항

호스트 복구는 호스트 수준 상태 확인을 사용하여 전용 호스트 가용성을 평가하고 기본 시스템 오류를 감지합니다. 시스템 상태 확인이 실패할 수 있는 문제의 예를 들면 다음과 같습니다.

- 네트워크 연결 끊김
- 시스템 전원 중단
- 물리적 호스트의 하드웨어 또는 소프트웨어 문제

전용 호스트에서 시스템 장애가 감지되면 호스트 복구가 시작되고 Amazon EC2가 자동으로 전용 호스트 대체를 할당합니다. 전용 호스트 대체는 새로운 호스트 ID를 받지만 다음과 같이 원본 전용 호스트와 동일한 속성을 유지합니다.

- 가용 영역
- 인스턴스 유형
- 태그
- 자동 배치 설정

대체 전용 호스트가 할당된 후 인스턴스는 대체 전용 호스트로 복구됩니다. 복구 된 인스턴스는 다음과 같이 원본 인스턴스와 동일한 속성을 유지합니다.

- 인스턴스 ID
- 프라이빗 IP 주소
- 탄력적 IP 주소
- EBS 볼륨 장치
- 모든 인스턴스 메타데이터

인스턴스가 손상된 전용 호스트와 호스트 선호도 관계를 갖는 경우 복구된 인스턴스는 대체 전용 호스트와 호스트 선호도를 수립합니다.

모든 인스턴스가 대체 전용 호스트로 복구되면 손상된 전용 호스트가 해제되고 대체 전용 호스트를 사용할 수 있게 됩니다.

호스트 복구가 시작되면 AWS 계정 소유자에게 이메일 및 AWS Personal Health Dashboard 이벤트로 알림이 전송됩니다. 호스트 복구가 성공적으로 완료되면 두 번째 알림이 전송됩니다.

중지된 인스턴스는 대체 전용 호스트로 복구되지 않습니다. 손상된 전용 호스트를 대상으로 하는 중지된 인스턴스를 시작하려고 하면 인스턴스가 시작되지 않습니다. 중지된 인스턴스를 수정하여 다른 전용 호스트를 대상으로 하거나 일치하는 구성 및 자동 배치가 활성화된 사용 가능한 전용 호스트에서 시작하는 것이 좋습니다.

인스턴스 스토리지가 포함된 인스턴스는 대체 전용 호스트로 복구되지 않습니다. 해결 수단으로, 손상된 전용 호스트는 만료 표시되고 호스트 복구가 완료된 후 만료 알림이 전송됩니다. 손상된 전용 호스트의 나머지 인스턴스를 수동으로 복구하려면 지정된 기간 내에 만료 알림에 설명된 해결 단계를 수행합니다.

AWS License Manager를 사용하여 라이선스를 추적하는 경우 AWS License Manager는 라이선스 구성 제한에 따라 대체 전용 호스트에 대한 새 라이선스를 할당합니다. 라이선스 구성에 호스트 복구의 결과로 위반되는 하드 제한이 있는 경우 복구 프로세스가 허용되지 않으며 Amazon SNS 알림을 통해 호스트 복구 실패 알림이 전송됩니다. 라이선스 구성에 호스트 복구의 결과로 위반되는 소프트 제한이 있는 경우 복구가 계속 허용되며 Amazon SNS 알림을 통해 제한 위반 알림이 전송됩니다. 자세한 내용은 AWS License Manager 사용 설명서의 라이선스 구성 사용 단원을 참조하십시오.

호스트 복구 구성

전용 호스트 할당 시 또는 Amazon EC2 콘솔 또는 AWS Command Line Interface(CLI)를 사용하여 할당한 후 호스트 복구를 구성할 수 있습니다.

목차

- 호스트 복구 활성화 (p. 406)
- 호스트 복구 비활성화 (p. 406)
- 호스트 복구 구성 보기 (p. 407)

호스트 복구 활성화

전용 호스트 할당 시 또는 할당 후 호스트 복구를 활성화할 수 있습니다.

전용 호스트 할당 시 호스트 복구를 활성화하는 방법에 대한 자세한 내용은 전용 호스트 할당 (p. 385) 단원을 참조하십시오.

할당 후 호스트 복구를 활성화하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 전용 호스트를 선택합니다.
3. 호스트 복구를 활성화할 전용 호스트를 선택한 다음 작업, Modify Host Recovery(호스트 복구 수정)를 선택합니다.
4. Host recovery(호스트 복구)에서 활성화를 선택한 다음 저장을 선택합니다.

할당 후 호스트 복구를 활성화하려면(AWS CLI)

modify-hosts 명령을 사용하여 `host-recovery` 파라미터를 지정합니다.

```
$ aws ec2 modify-hosts --host-recovery on --host-ids h-012a3456b7890cdef
```

호스트 복구 비활성화

전용 호스트가 할당된 후 언제든 호스트 복구를 비활성화할 수 있습니다.

할당 후 호스트 복구를 비활성화하려면(콘솔)

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

2. 탐색 창에서 전용 호스트를 선택합니다.
3. 호스트 복구를 비활성화할 전용 호스트를 선택한 다음 작업, Modify Host Recovery(호스트 복구 수정)를 선택합니다.
4. Host recovery(호스트 복구)에서 비활성화를 선택한 다음 저장을 선택합니다.

할당 후 호스트 복구를 비활성화하려면(AWS CLI)

modify-hosts 명령을 사용하여 `host-recovery` 파라미터를 지정합니다.

```
$ aws ec2 modify-hosts --host-recovery off --host-ids h-012a3456b7890cdef
```

호스트 복구 구성 보기

언제든지 전용 호스트에 대한 호스트 복구 구성을 볼 수 있습니다.

전용 호스트에 대한 호스트 복구 구성을 보려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 전용 호스트를 선택합니다.
3. 전용 호스트를 선택하고 설명 탭에서 Host Recovery(호스트 복구) 필드를 검토합니다.

전용 호스트에 대한 호스트 복구 구성을 보려면(AWS CLI)

describe-hosts 명령을 사용합니다.

```
$ aws ec2 describe-hosts --host-ids h-012a3456b7890cdef
```

`HostRecovery` 응답 요소는 호스트 복구가 활성화 또는 비활성화됐는지 여부를 나타냅니다.

호스트 복구 상태

전용 호스트 장애가 감지되면 손상된 전용 호스트는 `under-assessment` 상태가 되고 모든 인스턴스는 `impaired` 상태가 됩니다. `under-assessment` 상태에 있는 동안에는 손상된 전용 호스트에서 인스턴스를 시작할 수 없습니다.

대체 전용 호스트가 할당된 후에는 `pending` 상태가 됩니다. 호스트 복구 프로세스가 완료될 때까지 이 상태로 유지됩니다. `pending` 상태에 있는 동안에는 대체 전용 호스트에서 인스턴스를 시작할 수 없습니다. 대체 전용 호스트에서 복구된 인스턴스는 복구 프로세스 중에 `impaired` 상태로 유지됩니다.

호스트 복구가 완료되면 대체 전용 호스트가 `available` 상태가 되고 복구된 인스턴스는 `running` 상태로 돌아갑니다. `available` 상태가 된 후 대체 전용 호스트로 인스턴스를 시작할 수 있습니다. 손상된 원본 전용 호스트는 영구적으로 해제되고 `released-permanent-failure` 상태가 됩니다.

손상된 전용 호스트에 인스턴스 스토어 지원 볼륨을 가진 인스턴스 등 호스트 복구를 지원하지 않는 인스턴스가 있는 경우 전용 호스트가 해제되지 않습니다. 대신, 만료 표시되고 `permanent-failure` 상태가 됩니다.

지원되는 인스턴스 유형

호스트 복구는 A1, C3, C4, C5, C5n, Inf1, M3, M4, M5, M5n, P3, R3, R4, R5, R5n, X1, X1e, u-6tb1, u-9tb1, u-12tb1, u-18tb1 및 u-24tb1의 인스턴스 패밀리에 대해 지원됩니다.

지원되지 않는 인스턴스를 복구하려면 지원되지 않는 인스턴스 수동 복구 (p. 408) 단원을 참조하십시오.

지원되지 않는 인스턴스 수동 복구

호스트 복구는 인스턴스 스토어 볼륨을 사용하는 인스턴스 복구를 지원하지 않습니다. 자동으로 복구할 수 없는 인스턴스를 수동으로 복구하려면 아래 지침을 따릅니다.

> **Warning**
>
> 인스턴스가 중지되거나 종료되면 인스턴스 스토어 볼륨의 데이터는 삭제됩니다. 여기에는 루트 디바이스가 EBS 볼륨인 인스턴스에 연결된 인스턴스 스토어 볼륨도 포함됩니다. 인스턴스 스토어 볼륨에서 데이터를 유지하려면 인스턴스가 중지되거나 종료되기 전에 영구 스토리지에 백업하십시오.

EBS 지원 인스턴스 수동 복구

자동으로 복구할 수 없는 EBS 지원 인스턴스의 경우 인스턴스를 수동으로 중지했다가 시작하여 새 전용 호스트로 복구하는 것이 좋습니다. 인스턴스 중단과 중단 후 인스턴스 구성에 발생하는 변경 사항에 대한 자세한 내용은 인스턴스 중지 및 시작 (p. 507) 단원을 참조하십시오.

인스턴스 스토어 지원 인스턴스 수동 복구

예를 들어 자동으로 복구할 수 없는 스토어 백업 인스턴스의 경우 다음작업을 수행하는 것이 좋습니다.

1. 최신 전용 호스트의 새로운 AMI에서 대체 인스턴스를 시작합니다.
2. 필요한 모든 데이터를 대체 인스턴스로 마이그레이션합니다.
3. 손상된 전용 호스트의 원본 인스턴스를 종료합니다.

관련 서비스

전용 호스트는 다음과 같은 AWS 서비스와 통합됩니다.

- AWS License Manager—Amazon EC2 전용 호스트에서 라이선스를 추적합니다(AWS License Manager를 사용할 수 있는 리전에서만 지원됨). 자세한 내용은 AWS License Manager 사용 설명서를 참조하십시오.

요금

호스트 복구 사용에 대한 추가 요금은 없지만 일반적으로 전용 호스트 요금이 적용됩니다. 자세한 내용은 Amazon EC2 전용 호스트 요금을 참조하십시오.

호스트 복구가 시작된 직후 손상된 전용 호스트에 대해서는 더 이상 요금이 청구되지 않습니다. 대체 전용 호스트에 대한 요금 청구는 `available` 상태가 된 후에 시작됩니다.

손상된 전용 호스트가 온디맨드 요금을 사용하여 청구된 경우 대체 전용 호스트도 온디맨드 요금을 사용하여 청구됩니다. 손상된 전용 호스트에 활성 전용 호스트 예약이 있는 경우, 대체 전용 호스트로 전송됩니다.

구성 변경 추적

AWS Config를 사용하여 전용 호스트의 구성 변경 내용과 이 전용 호스트에서 시작, 중지 또는 종료된 인스턴스의 구성 변경 내용을 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 레코딩 상태—활성이면 AWS Config에서 하나 이상의 AWS 리소스 유형을 녹화 중입니다. 이러한 리소스 유형에는 전용 호스트 및 전용 인스턴스가 포함될 수 있습니다. 라이선스 보고에 필요한 정보를 캡처하려면 다음 필드에서 호스트 및 인스턴스가 기록되는지 확인합니다.

- 호스트 레코딩 상태 - 활성으로 설정 시 전용 호스트 구성 정보가 기록됩니다.
- 인스턴스 레코딩 상태 - 활성이면 전용 인스턴스 구성 정보가 레코딩됩니다.

세 조건 중 하나라도 비활성화되면 Config 레코딩 편집 버튼의 아이콘이 빨간색으로 표시됩니다. 이 도구의 이점을 최대한 활용하려면 세 기록 방법을 모두 활성화하십시오. 세 방법이 모두 활성화되면 아이콘이 녹색으로 표시됩니다. 설정을 편집하려면 Config 레코딩 편집을 선택합니다. 그러면 AWS Config 콘솔의 AWS Config 설정 페이지로 이동하며, 여기서 AWS Config를 설정하고 호스트, 인스턴스 및 기타 지원되는 리소스 유형에 대한 기록을 시작할 수 있습니다. 자세한 내용은 AWS Config 개발자 안내서의 콘솔을 사용하여 AWS Config 설정 단원을 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 전용 호스트 페이지의 Config 타임라인 열에서 회색 아이콘을 선택합니다. 보다 자세한 내용은 AWS Config 개발자 안내서의 AWS Config 콘솔에서 구성 세부 사항 보기를 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 CLI를 사용하여 구성 세부 정보 보기 단원을 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-1234567890abcdef0
```

콘솔을 사용하여 AWS Config 설정을 관리하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 전용 호스트 페이지에서 Config 레코딩 편집을 선택합니다.
3. AWS Config 콘솔에서 제공되는 단계를 수행하여 기록을 켭니다. 자세한 내용은 Setting up AWS Config using the Console 단원을 참조하십시오.

자세한 내용은 AWS Config 콘솔에서 구성 세부 정보 보기 단원을 참조하십시오.

명령줄 또는 API를 사용하여 AWS Config를 활성화하려면

- AWS CLI를 사용하는 방법은 AWS Config 개발자 안내서의 구성 세부 정보 보기(AWS CLI)를 참조하십시오.
- Amazon EC2 API를 사용하려면 GetResourceConfigHistory를 참조하십시오.

전용 인스턴스

전용 인스턴스는 단일 고객 전용 하드웨어에의 가상 가설 클라우드(VPC)에서 실행되는 Amazon EC2 인스턴스입니다. 서로 다른 AWS 계정에 속한 전용 인스턴스는 하드웨어에서 물리적으로 구분됩니다. 또한 단일 지급인 계정에 연결된 AWS 계정에 속하는 는 하드웨어 수준에서 물리적으로 격리됩니다. 하지만 전용 인스턴스는 전용 인스턴스가 아닌 동일한 AWS 계정의 다른 인스턴스와 하드웨어를 공유할 수 있습니다.

> **Note**
>
> 전용 호스트 역시 고객 전용의 물리적 서버입니다. 를 사용하여 서버에서 인스턴스의 배치 방법을 확인 및 제어할 수 있습니다. 자세한 내용은 전용 호스트 (p. 381) 단원을 참조하십시오.

전용 인스턴스 기본 사항

VPC에서 실행하는 각 인스턴스는 테넌시 속성으로 실행됩니다. 이 속성에는 다음과 같은 값이 있습니다.

테넌시 값	설명
`default`	인스턴스가 공유된 하드웨어에서 실행됩니다.
`dedicated`	인스턴스가 단일 테넌트 하드웨어에서 실행됩니다.
`host`	인스턴스는 구성을 제어할 수 있는 격리된 서버인 전용 호스트에서 실행됩니다.

인스턴스를 시작한 이후에는 테넌시를 변경하는 데 몇 가지 제한이 있습니다.

- 시작한 이후에는 인스턴스의 테넌시를 `default`에서 `dedicated` 또는 `host`로 변경할 수 없습니다.
- 시작한 이후에는 인스턴스의 테넌시를 `dedicated` 또는 `host`에서 `default`로 변경할 수 없습니다.

시작한 이후에는 인스턴스의 테넌시를 `dedicated`에서 `host`로, 또는 `host`에서 `dedicated`로 변경할 수 있습니다. 자세한 내용은 인스턴스의 테넌시 변경 (p. 414) 단원을 참조하십시오.

각 인스턴스에는 관련 인스턴스 테넌시 속성이 있습니다. 이 속성에는 다음과 같은 값이 있습니다.

테넌시 값	설명
`default`	인스턴스 시작 중 다른 테넌트를 명시적으로 지정하지 않는 한, VPC로 시작된 인스턴스는 기본적으로 공유 하드웨어에서 실행됩니다.
`dedicated`	인스턴스 시작 중 `host`의 테넌시를 명시적으로 지정하지 않는 한, VPC로 시작된 인스턴스는 기본적으로 전용 인스턴스입니다. 인스턴스 시작 중 `default`의 테넌트를 지정할 수 없습니다.

VPC의 인스턴스 테넌시는 생성한 후에 `dedicated`에서 `default`로 변경할 수 있습니다. VPC의 인스턴스 테넌시는 `dedicated`로 변경할 수 없습니다.

전용 인스턴스를 생성하려면 다음 작업을 수행합니다.

- `dedicated`로 설정된 인스턴스 테넌시로 VPC를 생성합니다. 이 VPC로 시작된 모든 인스턴스는 전용 인스턴스입니다.
- `default`로 설정된 인스턴스 테넌시로 VPC를 생성하고 시작할 때 모든 인스턴스에 대해 `dedicated` 테넌시를 지정합니다.

전용 인스턴스 제한 사항

일부 AWS 서비스나 그 기능은 인스턴스 테넌시가 `dedicated`로 설정된 VPC에서 작동하지 않습니다. 서비스 설명서에서 이에 관한 제한 사항이 있는지 확인하십시오.

일부 인스턴스 유형은 인스턴스 테넌시가 `dedicated`로 설정된 VPC에서 시작할 수 없습니다. 지원되는 인스턴스 유형에 대한 자세한 내용은 Amazon EC2 전용 인스턴스를 참조하십시오.

Amazon EBS다음으로 바꿉니다.전용 인스턴스

Amazon EBS 지원 전용 인스턴스를 시작할 경우 EBS 볼륨은 단일 테넌트 하드웨어에서 실행되지 않습니다.

전용 테넌시를 포함하는 예약 인스턴스

전용 인스턴스를 시작하기 위한 용량을 충분히 확보하려면 예약 인스턴스를 구입하면 됩니다. 자세한 내용은 예약 인스턴스 (p. 273)를 참조하십시오.

전용 예약 인스턴스를 구입하면 대폭 할인된 사용 요금으로 전용 인스턴스를 VPC에서 시작할 수 있는 용량이 제공됩니다. 사용 요금 인하는 전용 테넌시 인스턴스를 시작할 경우에만 적용됩니다. 기본 테넌시가 있는 예약 인스턴스를 구입하면 `default` 테넌시가 있는 실행 중인 인스턴스에만 적용되고 `dedicated` 테넌시가 있는 실행 중인 인스턴스에는 적용되지 않습니다.

또한 예약 인스턴스를 구입한 후에는 수정 프로세스를 사용하여 테넌시를 변경할 수 없습니다. 그러나 전환형 예약 인스턴스를 테넌시가 다른 새 전환형 예약 인스턴스와 교환할 수 있습니다.

전용 인스턴스 자동 조정

Amazon EC2 Auto Scaling을 사용하여 전용 인스턴스를 시작할 수 있습니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 VPC에서 Auto Scaling 인스턴스 시작을 참조하십시오.

전용 인스턴스 자동 복구

기본 하드웨어 장애나 복구하는 데 AWS 개입이 필요한 문제로 인해 전용 인스턴스가 손상된 경우 이에 대해 자동 복구를 구성할 수 있습니다. 자세한 내용은 인스턴스 복구 (p. 528) 단원을 참조하십시오.

전용 스팟 인스턴스

스팟 인스턴스 요청을 생성할 때 `dedicated`의 테넌시를 지정하여 전용 스팟 인스턴스를 실행할 수 있습니다. 자세한 내용은 스팟 인스턴스의 테넌시 지정 (p. 327) 단원을 참조하십시오.

전용 인스턴스 가격

전용 인스턴스 요금은 온디맨드 인스턴스 요금과 다릅니다. 자세한 내용은 Amazon EC2 전용 인스턴스 제품 페이지를 참조하십시오.

전용 인스턴스를 사용한 버스트 가능한 성능 인스턴스

the section called “성능 순간 확장 가능 인스턴스” (p. 194)를 사용하여 전용 테넌시 하드웨어에서 실행하는 이점을 활용할 수 있습니다. T3 전용 인스턴스는 기본적으로 무제한 모드로 시작되며, 기본 수준의 CPU 성능 외에 버스트 기능이 있어 워크로드에 필요한 만큼 성능을 높일 수 있습니다. T3 기본 성능과 버스트 기능은 CPU 크레딧에 의해 좌우됩니다. T3 인스턴스 유형의 버스트 가능한 특성상, 최상의 성능을 위해 T3 인스턴스가 전용 하드웨어의 CPU 리소스를 어떻게 사용하는지 모니터링하는 것이 좋습니다. T3 전용 인스턴

스는 임의의 CPU 동작을 나타내지만 평균 CPU 사용량이 기본 사용량 이하인 다양한 워크로드를 사용하는 고객을 대상으로 합니다. 자세한 내용은 the section called “CPU 크레딧 및 기준 성능” (p. 195) 단원을 참조하십시오.

Amazon EC2 에는 성능 변동성을 식별하고 수정할 수 있는 시스템이 있습니다. 그러나 상관 관계가 있는 CPU 사용 패턴을 가진 T3 전용 인스턴스를 여러 개 실행하면 단기적인 변동성이 발생할 수 있습니다. 이처럼 까다롭거나 상관 관계가 있는 워크로드의 경우 T3 전용 인스턴스 대신 M5 또는 M5a 전용 인스턴스를 사용하는 것이 좋습니다.

전용 인스턴스 작업

`dedicated` 인스턴스 테넌시로 VPC를 생성하여 해당 VPC로 시작되는 모든 인스턴스가 전용 인스턴스가 되게 합니다. 또는 시작되는 동안 인스턴스의 테넌시를 지정할 수 있습니다.

주제

- 전용 인스턴스 테넌시의 VPC 생성하기 (p. 412)
- VPC에서 전용 인스턴스 시작 (p. 412)
- 테넌시 정보 조회 (p. 413)
- 인스턴스의 테넌시 변경 (p. 414)
- VPC의 테넌시 변경 (p. 414)

전용 인스턴스 테넌시의 VPC 생성하기

VPC를 생성할 경우 VPC의 인스턴스 테넌시를 지정하는 옵션이 제공됩니다. Amazon VPC 콘솔을 사용하는 경우 VPC 마법사 또는 VPC 페이지를 사용하여 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(VPC 마법사 시작)를 선택합니다.
3. VPC 구성을 선택한 후 선택을 선택합니다.
4. 마법사 다음 페이지의 하드웨어 테넌시 목록에서 전용을 선택합니다.
5. VPC 만들기를 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 VPC와 VPC 만들기를 차례로 선택합니다.
3. 테넌시에서 전용을 선택합니다. CIDR 블록을 지정하고 예, 생성을 선택합니다.

명령줄을 사용하여 VPC를 생성할 때 테넌시 옵션을 설정하려면

- create-vpc(AWS CLI)
- New-EC2Vpc(Windows PowerShell용 AWS 도구)

`dedicated` 인스턴스 테넌시가 있는 VPC에서 인스턴스를 시작하면 인스턴스 테넌시와 상관없이 인스턴스가 자동으로 전용 인스턴스가 됩니다.

VPC에서 전용 인스턴스 시작

Amazon EC2 인스턴스 시작 마법사를 사용하여 전용 인스턴스를 시작할 수 있습니다.

콘솔을 사용하여 기본 테넌시 VPC에서 전용 인스턴스를 시작하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 인스턴스 시작을 선택합니다.
3. Amazon 머신 이미지(AMI) 선택 페이지에서 AMI를 선택한 후 선택을 선택합니다.
4. 인스턴스 유형 선택 페이지에서 인스턴스 유형을 선택하고 Next: Configure Instance Details(다음: 인스턴스 세부 정보 구성)를 차례대로 선택합니다.

   Note

   전용 인스턴스를 지원하는 인스턴스 유형을 선택해야 합니다. 자세한 내용은 Amazon EC2 전용 인스턴스를 참조하십시오.
5. Configure Instance Details(인스턴스 세부 정보 구성) 페이지에서 VPC와 서브넷을 선택합니다. 테넌시 목록에서 Dedicated - Run a dedicated instance(전용 - 전용 인스턴스 실행)와 Next: Add Storage(다음: 스토리지 추가)를 차례로 선택합니다.
6. 마법사에 표시되는 지침에 따라 계속합니다. 인스턴스 시작 검토 페이지에서 옵션을 모두 검토했으면 시작을 선택하여 키 페어를 선택하고 전용 인스턴스를 시작합니다.

`host` 테넌시를 사용하여 인스턴스를 시작하는 자세한 내용은 전용 호스트에서 인스턴스 시작 (p. 387) 단원을 참조하십시오.

명령줄을 사용하여 시작 중에 인스턴스의 테넌시 옵션을 설정하려면

- run-instances(AWS CLI)
- New-EC2Instance(Windows PowerShell용 AWS 도구)

테넌시 정보 조회

콘솔을 사용하여 VPC의 테넌시 정보를 조회하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 VPC를 선택합니다.
3. 테넌시 열에서 해당 VPC의 인스턴스 테넌시를 확인합니다.
4. 테넌시 열이 표시되지 않는 경우 테이블 열 편집(기어 모양 아이콘)을 선택하고 열 표시/숨기기 대화 상자에서 테넌시를 선택한 후 닫기를 선택합니다.

콘솔을 사용하여 인스턴스의 테넌시 정보를 조회하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 테넌시 열에서 해당 인스턴스의 테넌시를 확인합니다.
4. 테넌시 열이 표시되지 않으면 다음 중 하나를 수행하십시오.
   - 열 표시/숨기기](기어 모양 아이콘)를 선택하고 열 표시/숨기기 대화 상자에서 테넌시를 선택한 후 닫기를 선택합니다.
   - 인스턴스를 선택합니다. 세부 정보 창의 설명 탭에 인스턴스의 정보와 그 테넌시가 표시됩니다.

명령줄을 사용하여 VPC의 테넌시를 나타내려면

- describe-vpcs(AWS CLI)
- Get-EC2Vpc(Windows PowerShell용 AWS 도구)

명령줄을 사용하여 인스턴스의 테넌시를 나타내려면

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

명령줄을 사용하여 예약 인스턴스의 테넌시 값을 나타내려면

- describe-reserved-instances(AWS CLI)
- Get-EC2ReservedInstance(Windows PowerShell용 AWS 도구)

명령줄을 사용하여 예약 인스턴스 상품의 테넌시 값을 나타내려면

- describe-reserved-instances-offerings(AWS CLI)
- Get-EC2ReservedInstancesOffering(Windows PowerShell용 AWS 도구)

인스턴스의 테넌시 변경

인스턴스 유형과 플랫폼에 따라 시작된 이후에 중지된 전용 인스턴스의 테넌시를 `host`로 변경할 수 있습니다. 다음에 인스턴스를 시작하면 계정에 할당된 에서 시작됩니다. 전용 호스트를 할당 및 사용하는 방법과 전용 호스트에서 사용할 수 있는 인스턴스 유형에 대한 자세한 내용은 전용 호스트 작업 (p. 384) 단원을 참조하십시오. 마찬가지로 시작된 이후에 중지된 전용 호스트 인스턴스의 테넌시를 `dedicated`로 변경할 수 있습니다. 인스턴스를 다음에 시작하면 Amazon에서 제어하는 단일 테넌트 하드웨어에서 시작됩니다.

콘솔을 사용하여 인스턴스의 테넌시를 변경하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택하고 인스턴스를 선택합니다.
3. 작업, 인스턴스 상태, 중지를 차례로 선택합니다.
4. 작업, 인스턴스 설정, 인스턴스 배치 수정을 차례대로 선택합니다.
5. 테넌시 목록에서 인스턴스를 전용 하드웨어에서 실행할지 전용 호스트에서 실행할지를 선택합니다. 저장을 선택합니다.

명령줄을 사용하여 인스턴스의 테넌시 값을 수정하려면

- modify-instance-placement(AWS CLI)
- Edit-EC2InstancePlacement (Windows PowerShell용 AWS 도구)

VPC의 테넌시 변경

VPC의 인스턴스 테넌시 속성은 `dedicated`에서 `default`로 변경할 수 있습니다. VPC의 인스턴스 테넌시를 수정해도 VPC에 있는 기존 인스턴스의 테넌시에는 영향을 미치지 않습니다. 다음에 VPC에서 인스턴스를 시작할 때 시작 시 다르게 지정하지 않는 한 `default` 테넌시가 유지됩니다.

VPC의 인스턴스 테넌시 속성은 `dedicated`로 변경할 수 없습니다.

VPC의 인스턴스 테넌시 속성은 AWS CLI, AWS SDK 또는 Amazon EC2 API만 사용하여 수정할 수 있습니다.

AWS CLI를 사용하여 VPC의 인스턴스 테넌시 속성을 수정하려면

- modify-vpc-tenancy 명령을 사용하여 VPC의 ID와 인스턴스 테넌시 값을 지정합니다. 지원되는 유일한 값은 `default`입니다.

```
aws ec2 modify-vpc-tenancy --vpc-id vpc-1a2b3c4d --instance-tenancy default
```

온디맨드 용량 예약

온디맨드 용량 예약을 사용하면 특정 가용 영역의 Amazon EC2 인스턴스에 대해 원하는 기간만큼 용량을 예약할 수 있습니다. 따라서 Savings Plans 또는 지역 예약 인스턴스에서 제공하는 결제 할인과는 별도로 용량 예약을 생성 및 관리할 수 있습니다. 용량 예약를 생성하면 필요할 때 언제든지 필요한 만큼 EC2 용량에 액세스할 수 있으며 유지할 수 있습니다. 1년 또는 3년 기간의 약정에 가입하지 않고도 언제든지 용량 예약를 생성할 수 있으며 용량은 즉시 이용할 수 있습니다. 이 예약이 더 이상 필요하지 않은 경우 용량 예약을 취소하면 이에 대해서는 요금이 더 이상 발생하지 않습니다.

용량 예약을 생성할 때 다음을 지정합니다.

- 용량을 예약할 가용 영역입니다.
- 용량을 예약할 인스턴스 수입니다.
- 인스턴스 유형, 테넌시, platform/OS 등을 포함한 인스턴스 속성

용량 예약는 해당 속성과 일치하는 인스턴스에서만 사용할 수 있습니다. 기본적으로 속성과 일치하는 인스턴스를 실행하면 용량 예약이 자동으로 사용됩니다. 용량 예약의 속성과 일치하는 인스턴스를 실행하고 있지 않으면 속성과 일치하는 인스턴스를 시작할 때까지 사용되지 않은 상태로 유지됩니다.

또한 Savings Plans 및 지역 예약 인스턴스를 용량 예약와 함께 사용하여 결제 할인 혜택을 받을 수 있습니다. 용량 예약의 속성이 Savings Plan 및 지역 예약 인스턴스의 속성과 일치하면 AWS에서 할인을 자동으로 적용합니다. 자세한 내용은 결제 할인 (p. 417) 단원을 참조하십시오.

목차

- 용량 예약, 예약 인스턴스 및 Savings Plans 간의 차이점 (p. 415)
- 용량 예약 제한 (p. 416)
- 용량 예약 제한 및 제약 (p. 416)
- 용량 예약 요금 및 결제 (p. 416)
- 용량 예약로 작업 (p. 418)
- 공유된 용량 예약에 대한 작업 (p. 422)

용량 예약, 예약 인스턴스 및 Savings Plans 간의 차이점

다음 표에 용량 예약, 예약 인스턴스 및 Savings Plans 간의 주요 차이점이 설명되어 있습니다.

	용량 예약	영역 예약 인스턴스	리전 예약 인스턴스	Savings Plans
기간	약정이 필요하지 않습니다. 필요할 때마다 생성하거나 취소할 수 있습니다.	고정 1년 또는 3년 약정 필요		
용량 혜택	특정 가용 영역에서 예약된 용량입니다.		가용 영역에서 용량을 예약하지 않습니다.	
결제 할인	결제 할인이 제공되지 않습니다. 용량 예약으로 시작된 인스턴스는 스탠다드 온디맨드 요율로 요금이 부과됩	결제 할인 제공		

	용량 예약	영역 예약 인스턴스	리전 예약 인스턴스	Savings Plans
	니다. 한편, Savings Plans 또는 지역 예약 인스턴스를 용량 예약와 함께 사용하면 결제 할인 혜택을 받을 수 있습니다. 영역 예약 인스턴스는 용량 예약에 적용되지 않습니다..			
인스턴스 제한	리전당 온디맨드 인스턴스 제한으로 제한됩니다.	가용 영역당 20으로 제한됩니다. 제한 증가를 요청할 수 있습니다.	리전당 20으로 제한됩니다. 제한 증가를 요청할 수 있습니다.	제한이 없습니다.

자세한 내용은 다음 자료를 참조하십시오.

- 예약 인스턴스 (p. 273)
- AWS Savings Plans 사용 설명서

용량 예약 제한

용량 예약이 가능한 인스턴스 수는 계정의 온디맨드 인스턴스 제한을 기반으로 합니다. 이 제한이 허용하는 인스턴스 수에서 이미 실행 중인 인스턴스 수를 차감한 인스턴스 수에 대해 용량을 예약할 수 있습니다.

용량 예약 제한 및 제약

용량 예약을 생성하기 전에 다음 제한 및 제약에 유의하십시오.

- 활성 및 미사용 용량 예약 수는 온디맨드 인스턴스 제한에 포함됩니다.
- 용량 예약은 한 AWS 계정에서 다른 AWS 계정으로 이전할 수 없습니다.
- 영역 예약 인스턴스 결제 할인은 용량 예약에 적용되지 않습니다.
- 배치 그룹에서는 용량 예약을 생성할 수 없습니다.
- 용량 예약는 전용 호스트와 사용할 수 없습니다.
- 용량 예약는 Bring Your Own License(BYOL)와 함께 사용할 수 없습니다.

용량 예약 요금 및 결제

용량 예약의 요금은 결제 옵션에 따라 다릅니다.

요금

용량 예약이 활성 상태인 경우 인스턴스 실행 여부에 따라 상응하는 온디맨드 요율로 요금이 부과됩니다. 예약을 사용하지 않는 경우 EC2 청구서에 사용되지 않은 예약으로 표시됩니다. 예약의 특성과 일치하는 인스턴스를 실행하는 경우 인스턴스 요금만 지불하고 예약 요금은 지불하지 않습니다. 선불금 또는 추가 요금은 없습니다.

예를 들어, 20개 `m4.large` Linux 인스턴스에 대해 용량 예약을 생성하고 동일한 가용 영역에서 15개 `m4.large` Linux 인스턴스를 실행할 경우 예약의 15개 활성 인스턴스와 사용되지 않은 5개 인스턴스에 대해 요금이 부과됩니다.

Savings Plans 및 지역 예약 인스턴스에 대한 결제 할인이 용량 예약에 적용됩니다. 자세한 내용은 결제 할인 (p. 417) 단원을 참조하십시오.

Amazon EC2 요금에 대한 자세한 정보는 Amazon EC2 요금을 참조하십시오.

결제

용량 예약은 초당 요금으로 청구됩니다. 다시 말해서 사용 시간이 한 시간 미만이라도 요금이 부과됩니다. 예를 들어 예약이 24시간 15분 동안 계정에서 예약이 활성 상태로 유지되는 경우 24.25시간이 예약 시간으로 청구됩니다.

다음 예는 용량 예약 요금 청구 방식을 보여줍니다. 용량 예약이 `m4.large` Linux 인스턴스 하나에 대해 생성되었으며, 사용 시간당 $0.10의 온디맨드 요율이 적용됩니다. 이 예에서는 용량 예약이 이 계정에 대해 다섯 시간 동안 활성 상태입니다. 처음 한 시간 동안 용량 예약이 사용되지 않았으므로, 사용되지 않은 한 시간에 대해서는 `m4.large` 인스턴스 유형의 스탠다드 온디맨드 요율로 요금이 청구됩니다. 2~5시간 동안은 `m4.large` 인스턴스에서 용량 예약이 사용됩니다. 이 시간 동안은 용량 예약에 대해 요금이 청구되지 않으며, 대신에 용량 예약을 사용하는 `m4.large` 인스턴스에 대해 이 계정에 요금이 청구됩니다. 여섯 번째 시간 동안은 용량 예약이 취소되었으므로 `m4.large` 인스턴스가 예약 용량 외부에서 일반적으로 실행됩니다. 해당 시간 동안은 `m4.large` 인스턴스 유형의 온디맨드 요율로 요금이 청구됩니다.

Hour	1	2	3	
Unused Capacity Reservation	$0.10	$0.00	$0.00	$
On-demand Instance Usage	$0.00	$0.10	$0.10	$
Hourly cost	$0.10	$0.10	$0.10	$

결제 할인

Savings Plans 및 지역 예약 인스턴스에 대한 결제 할인이 용량 예약에 적용됩니다. AWS는 일치하는 속성이 있는 용량 예약에 이러한 할인을 자동으로 적용합니다. 용량 예약이 인스턴스에 의해 사용되는 경우 해당 인스턴스에 할인이 적용됩니다. 할인은 사용되지 않은 용량 예약를 적용하기 전에 인스턴스 사용량에 우선적으로 적용됩니다.

영역 예약 인스턴스에 대한 결제 할인은 용량 예약에 적용되지 않습니다.

자세한 내용은 다음 자료를 참조하십시오.

- 예약 인스턴스 (p. 273)
- AWS Savings Plans 사용 설명서

청구서 보기

계정으로 청구되는 요금과 비용은 AWS Billing and Cost Management 콘솔에서 검토할 수 있습니다.

- 대시보드에는 계정에 대한 소비 요약이 표시됩니다.
- 청구서 페이지의 세부 정보에서 Elastic Compute Cloud 섹션과 리전을 확장하여 용량 예약에 대한 결제 정보를 가져옵니다.

요금을 온라인으로 확인하거나 CSV 파일을 다운로드할 수 있습니다. 자세한 내용은 AWS Billing and Cost Management 사용 설명서의 용량 예약 항목을 참조하십시오.

용량 예약로 작업

용량 예약 사용을 시작하려면 필요 가용 영역에서 용량 예약을 생성합니다. 그런 다음, 인스턴스를 예약 용량으로 시작하거나 용량 사용률을 실시간으로 확인할 수 있으며, 필요 시 용량을 늘리거나 줄일 수 있습니다.

기본적으로 용량 예약은 일치하는 속성(인스턴스 유형, 플랫폼 및 가용 영역)이 있는 새 인스턴스 및 실행 중인 인스턴스와 자동으로 맞춰집니다. 즉, 일치하는 속성을 가진 인스턴스는 용량 예약에서 자동으로 실행됩니다. 하지만 용량 예약을 특정 워크로드에 지정할 수도 있습니다. 이렇게 하면 예약 용량으로 실행할 수 있는 인스턴스를 명시적으로 제어할 수 있습니다.

예약이 종료되는 방법을 지정할 수 있습니다. 용량 예약을 수동으로 취소하거나, 지정된 시간에 자동으로 예약을 종료하도록 선택할 수 있습니다. 종료 시간을 지정하면 지정된 시간부터 1시간 내에 용량 예약이 취소됩니다. 예를 들어 2019년 5월 31일 13:30:55를 지정하는 경우 용량 예약은 2019년 5월 31일 13:30:55 ~ 14:30:55에 종료됩니다. 예약이 종료된 후에는 더 이상 인스턴스를 용량 예약으로 지정할 수 없습니다. 예약 용량으로 실행 중인 인스턴스가 중단되지 않은 상태로 계속 실행됩니다. 용량 예약으로 지정된 인스턴스를 중지하는 경우 용량 예약 지정 기본 설정을 제거하거나 다른 용량 예약으로 지정되도록 구성할 때까지 해당 인스턴스를 다시 시작할 수 없습니다.

목차

- 용량 예약 생성 (p. 418)
- 인스턴스를 기존 용량 예약으로 시작 (p. 419)
- 용량 예약 수정 (p. 420)
- 인스턴스의 용량 예약 설정 수정 (p. 420)
- 용량 예약 보기 (p. 421)
- 용량 예약 취소 (p. 421)

용량 예약 생성

용량 예약을 생성하고 나면 이 용량을 즉시 사용할 수 있습니다. 이 용량은 용량 예약이 활성 상태인 동안은 예약 상태로 유지되며 언제든지 인스턴스를 이 용량으로 시작할 수 있습니다. 용량 예약이 열려 있으면 일치하는 속성이 있는 새 인스턴스 및 기존 인스턴스는 용량 예약의 용량으로 자동 실행됩니다. 용량 예약이 `targeted`이면 인스턴스를 예약된 용량으로 실행하도록 지정해야 합니다.

다음 중 하나에 해당하는 경우 용량 예약 생성 요청이 실패할 수 있습니다.

- Amazon EC2에 용량이 충분하지 않아서 요청을 이행할 수 없습니다. 나중에 다시 시도하거나, 다른 가용 영역을 사용하거나, 용량을 줄여서 시도하십시오. 애플리케이션이 인스턴스 유형 및 크기 면에서 가변적인 경우 다른 인스턴스 속성으로 생성해 봅니다.
- 요청한 수량이 선택한 인스턴스 패밀리에 대한 온디맨드 인스턴스 제한을 초과합니다. 인스턴스 패밀리에 대한 온디맨드 인스턴스 제한을 늘리고 다시 시도하십시오. 자세한 내용은 온디맨드 인스턴스 제한 (p. 270) 단원을 참조하십시오.

콘솔을 사용하여 용량 예약을 생성하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 용량 예약를 선택한 후 용량 예약 생성을 선택합니다.
3. 용량 예약 생성 페이지의 인스턴스 세부 정보 섹션에서 다음 설정을 구성합니다. 시작하는 인스턴스 유형, 플랫폼, 인스턴스의 가용 영역은 여기서 지정한 인스턴스 유형, 플랫폼, 가용 영역과 일치해야 합니다. 그렇지 않으면 용량 예약이 적용되지 않습니다. 예를 들어 열려 있는 용량 예약이 일치하지 않으면 명시적으로 용량 예약을 대상으로 하는 인스턴스 시작이 실패합니다.

   a. 인스턴스 유형 - 예약된 용량으로 시작할 인스턴스 유형입니다.

b. EBS 최적 인스턴스 시작 - EBS 최적 인스턴스 용량을 예약할지 여부를 지정합니다. 이 옵션은 특정 인스턴스 유형에 대해 기본적으로 선택됩니다. EBS 최적화 인스턴스에 대한 자세한 내용은 Amazon Elastic Block Store (p. 888) 단원을 참조하십시오.

c. 시작 시 인스턴스 스토어 연결 - 인스턴스를 용량 예약로 시작된 인스턴스에서 임시 블록 수준 저장소를 사용할지 여부를 지정합니다. 인스턴스 스토어 볼륨의 데이터는 관련 인스턴스의 수명 기간 동안만 지속됩니다.

d. 플랫폼 - 사용자 인스턴스에 사용할 운영 체제입니다.

e. 가용 영역 - 용량을 예약할 가용 영역입니다.

f. 테넌시 - 공유 하드웨어(기본)를 실행할지 전용 인스턴스를 실행할지 지정합니다.

g. 수량—용량을 예약할 인스턴스 수입니다. 선택한 인스턴스 유형에 대해 남은 온디맨드 인스턴스 제한을 초과하는 수량을 지정하는 경우 이 요청이 거부됩니다.

4. 예약 세부 정보 섹션에서 다음 설정을 구성합니다.

   a. 예약 종료 - 다음 옵션 중 하나를 선택합니다.

      - 수동 - 명시적으로 취소할 때까지 용량을 예약합니다.
      - 지정 시간—지정된 날짜 및 시간에 용량 예약을 자동으로 취소합니다.

   b. 인스턴스 자격 - 다음 옵션 중 하나를 선택합니다.

      - open - (기본값) 용량 예약이 일치하는 속성(인스턴스 유형, 플랫폼 및 가용 영역)이 있는 인스턴스를 검색합니다. 일치하는 속성이 있는 인스턴스를 시작할 경우 예약 용량으로 자동 배치됩니다.
      - targeted—용량 예약에서, 일치하는 속성(인스턴스 유형, 플랫폼 및 가용 영역)이 있고 예약을 명시적으로 지정하는 인스턴스만 허용합니다.

5. 예약 요청을 선택합니다.

AWS CLI를 사용하여 용량 예약을 생성하는 방법

create-capacity-reservation 명령을 사용합니다.

```
aws ec2 create-capacity-reservation --instance-type instance_type --instance-
platform platform_type --availability-zone az --instance-count quantity
```

인스턴스를 기존 용량 예약으로 시작

인스턴스를 기존 용량 예약(일치하는 속성(인스턴스 유형, 플랫폼 및 가용 영역) 및 충분한 용량이 있는 경우)으로 시작할 수 있습니다. 인스턴스를 용량 예약으로 시작하면 시작된 인스턴스 수만큼 가용 용량이 감소됩니다. 예를 들어 인스턴스 세 개를 시작할 경우 용량 예약의 가용 용량이 3만큼 감소됩니다.

콘솔을 사용하여 기존 용량 예약으로 인스턴스를 시작하는 방법

1. 대시보드 또는 인스턴스에서 인스턴스 시작을 선택하여 인스턴스 시작 마법사를 엽니다.
2. Amazon 머신 이미지(AMI)와 인스턴스 유형을 선택합니다.
3. 인스턴스 세부 정보 구성을 설정합니다. 용량 예약에서 다음 옵션 중 하나를 선택합니다.

   - 열기 - 일치하는 속성과, 선택한 인스턴스 수에 맞는 용량이 있는 용량 예약으로 인스턴스를 시작합니다. 충분한 용량이 있는 적절한 용량 예약이 없는 경우 인스턴스는 온디맨드 용량을 사용합니다.
   - <용량 예약> — 이 특정 용량 예약로 인스턴스를 시작합니다. 용량 예약의 용량이 선택한 인스턴스 수에 맞게 충분하지 않으면 인스턴스가 시작되지 않습니다.
   - 없음 - 인스턴스가 용량 예약으로 시작되지 않도록 합니다.

4. 나머지 단계를 완료하여 인스턴스를 시작합니다.

AWS CLI를 사용하여 기존 용량 예약으로 인스턴스를 시작하는 방법

run-instances 명령을 사용하여 `--capacity-reservation-specification` 파라미터를 지정합니다.

다음 예제에서는 일치하는 속성과 가용 용량이 있는 열려 있는 용량 예약으로 `t2.micro` 인스턴스가 시작됩니다.

```
aws ec2 run-instances --image-id ami-abc12345 --count 1 --instance-type t2.micro --
key-name MyKeyPair --availability-zone us-east-1b --capacity-reservation-specification
 CapacityReservationPreference=open
```

다음 예제에서는 `t2.micro` 인스턴스가 `targeted` 용량 예약로 시작됩니다.

```
aws ec2 run-instances --image-id ami-abc12345 --count 1 --instance-type t2.micro --
key-name MyKeyPair --availability-zone us-east-1b --capacity-reservation-specification
 CapacityReservationTarget=[{CapacityReservationId=cr-a1234567}]
```

용량 예약 수정

생성한 후 활성 용량 예약의 속성을 변경할 수 있습니다. 만료되었거나 명시적으로 취소한 후에는 용량 예약을 수정할 수 없습니다.

용량 예약을 수정하는 경우, 수량 늘리기/줄이기 또는 해제 방식 변경만 가능합니다. 용량 예약의 인스턴스 유형, EBS 최적화, 인스턴스 스토어 설정, 플랫폼, 가용 영역 또는 인스턴스 자격은 변경할 수 없습니다. 이러한 속성을 수정해야 하는 경우에는 예약을 취소하고 나서 필요한 속성으로 예약을 다시 생성하는 것이 좋습니다.

선택한 인스턴스 유형에 대해 남은 온디맨드 인스턴스 제한을 초과하는 새 수량을 지정하는 경우 업데이트가 실패합니다.

콘솔을 사용하여 용량 예약을 수정하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 용량 예약을 선택한 후, 수정할 용량 예약을 선택하고 나서 편집을 선택합니다.
3. 필요에 따라 수량 또는 예약 종료 옵션을 수정하고 변경 사항 저장을 선택합니다.

AWS CLI를 사용하여 용량 예약을 수정하는 방법

modify-capacity-reservations 명령을 사용합니다.

```
aws ec2 modify-capacity-reservation --capacity-reservation-id reservation_id --instance-
count quantity --end-date-type limited|unlimited --end-date expiration_date
```

인스턴스의 용량 예약 설정 수정

중지된 인스턴트에 대한 다음 용량 예약 설정은 다음과 같이 언제든지 수정할 수 있습니다.

- 일치하는 속성(인스턴스 유형, 플랫폼 및 가용 영역) 및 가용 용량이 있는 용량 예약에서 시작합니다.
- 특정 용량 예약에서 인스턴스를 시작합니다.
- 인스턴스가 용량 예약에서 시작되지 않도록 합니다.

콘솔을 사용하여 인스턴스의 용량 예약 설정을 수정하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 인스턴스를 선택한 후 수정할 인스턴스를 선택합니다. 인스턴스가 아직 중지되지 않은 경우 해당 인스턴스를 중지합니다.

3. 작업, 용량 예약 설정 수정을 선택합니다.
4. 용량 예약에서 다음 옵션 중 하나를 선택합니다.
   - 열기 - 일치하는 속성(인스턴스 유형, 플랫폼 및 가용 영역) 및 가용 용량이 있고 열려 있는 용량 예약에서 인스턴스를 시작합니다. 일치하는 용량 예약이 없는 경우 인스턴스는 온디맨드 용량을 사용합니다.
   - <용량 예약> — 이 특정 용량 예약에서 인스턴스를 실행합니다. 인스턴스 속성(인스턴스 유형, 플랫폼 및 가용 영역)이 용량 예약의 속성과 일치하지 않는 경우 또는 선택한 용량 예약의 용량이 충분하지 않은 경우, 인스턴스가 시작되지 않습니다.
   - 없음 - 인스턴스가 용량 예약에서 실행되지 않게 합니다.

AWS CLI를 사용하여 인스턴스의 용량 예약 설정을 수정하는 방법

modify-instance-capacity-reservation-attributes 명령을 사용합니다.

```
aws ec2 modify-instance-capacity-reservation-attributes --instance-id instance_id --
capacity-reservation-specification 'CapacityReservationPreference=none|open'
```

용량 예약 보기

용량 예약에는 다음과 같은 잠재적인 상태가 있습니다.

- `active` - 용량을 사용할 수 있습니다.
- `expired` - 용량 예약이 예약 요청 시 지정한 날짜 및 시간에 자동으로 만료됩니다. 예약 용량을 더 이상 사용할 수 없습니다.
- `cancelled`—용량 예약이 수동으로 취소되었습니다. 예약 용량을 더 이상 사용할 수 없습니다.
- `pending`- 용량 예약 요청에 성공했지만 용량 프로비저닝이 여전히 대기 중입니다.
- `failed`- 용량 예약 요청에 실패했습니다. 잘못된 요청 파라미터, 용량 제약 조건 또는 인스턴스 제한 제약 조건이 요청 실패의 원인일 수 있습니다. 60분 동안 실패한 요청을 볼 수 있습니다.

콘솔을 사용하여 용량 예약을 보는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 용량 예약을 선택하고 확인할 용량 예약을 선택합니다.
3. 이 예약에 대하여 시작된 인스턴스 보기를 선택합니다.

AWS CLI를 사용하여 용량 예약을 보는 방법

describe-capacity-reservations 명령을 사용합니다.

```
aws ec2 describe-capacity-reservations
```

용량 예약 취소

예약된 용량이 더 이상 필요하지 않을 경우 언제라도 용량 예약을 취소할 수 있습니다. 용량 예약을 취소할 경우 해당 용량이 즉시 해지되고 더 이상 사용 용량으로 예약되지 않습니다.

인스턴스를 실행 중인 용량 예약과 빈 용량 예약을 취소할 수 있습니다. 인스턴스를 실행 중인 용량 예약을 취소할 경우 인스턴스가 일반적으로 표준 온디맨드 인스턴스 요율이나 할인된 요율(일치하는 Savings Plan 또는 지역 예약 인스턴스가 있는 경우)로 용량 예약 외부에서 계속 실행됩니다.

용량 예약을 취소한 후에는 해당 용량 예약으로 지정된 인스턴스를 더 이상 시작할 수 없습니다. 다른 용량 예약으로 지정되거나, 일치하는 속성 및 충분한 용량이 있는 '열린' 용량 예약으로 시작하거나, 용량 예약으로

시작하지 않도록 이러한 인스턴스를 수정합니다. 자세한 내용은 인스턴스의 용량 예약 설정 수정 (p. 420) 단원을 참조하십시오.

콘솔을 사용하여 용량 예약을 취소하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 용량 예약을 선택하고 취소할 용량 예약을 선택합니다.
3. 예약 취소, 예약 취소를 선택합니다.

AWS CLI를 사용하여 용량 예약을 취소하는 방법

cancel-capacity-reservation 명령을 사용합니다.

```
aws ec2 cancel-capacity-reservation --capacity-reservation-id reservation_id
```

공유된 용량 예약에 대한 작업

용량 예약 공유를 통해 용량 예약 소유자는 예약된 용량을 다른 AWS 계정이나 AWS 조직 내에서 공유할 수 있습니다. 이를 통해 용량 예약를 중앙 집중식으로 생성 및 관리하고 예약된 용량을 여러 AWS 계정 또는 AWS 조직 내에서 공유할 수 있습니다.

이 모델에서 용량 예약(소유자)를 소유한 AWS 계정은 다른 AWS 계정(소비자)과 이를 공유합니다. 소비자는 자신의 계정에서 자신이 소유한 용량 예약에서 인스턴스를 시작하는 동일한 방식으로 인스턴스를 용량 예약에서 시작할 수 있습니다. 용량 예약 소유자는 용량 예약 및 해당 위치에서 시작한 인스턴스를 관리해야 합니다. 소유자는 공유한 용량 예약로 소비자가 시작한 인스턴스를 수정할 수 없습니다. 소비자는 공유된 용량 예약 위치에서 시작한 인스턴스를 관리해야 합니다. 소비자는 다른 소비자 또는 용량 예약 소유자가 소유한 인스턴스를 보거나 수정할 수 없습니다.

용량 예약 소유자는 다음과 용량 예약을 공유할 수 있습니다.

- AWS 조직 내부 또는 외부의 특정 AWS 계정
- AWS 조직 내부의 조직 단위
- 전체 AWS 조직

목차

- 용량 예약 공유를 위한 전제 조건 (p. 422)
- 관련 서비스 (p. 423)
- 여러 가용 영역 공유 (p. 423)
- 용량 예약 공유 (p. 423)
- 공유된 용량 예약의 공유 해제 (p. 424)
- 공유 용량 예약 확인 (p. 424)
- 공유 용량 예약 사용량 보기 (p. 425)
- 공유 용량 예약 권한 (p. 425)
- 결제 및 측정 (p. 425)
- 인스턴스 제한 (p. 425)

용량 예약 공유를 위한 전제 조건

- 용량 예약을 공유하려면 AWS 계정에서 소유하고 있어야 합니다. 나와 공유된 용량 예약을 공유할 수 없습니다.

- 공유 테넌시 인스턴스에 대해서만 용량 예약을 공유할 수 있습니다. 전용 테넌시 인스턴스에 대해 용량 예약을 공유할 수 없습니다.
- 결제 내역이 제한된 새 AWS 계정 또는 AWS 계정에서는 용량 예약 공유를 사용할 수 없습니다. 유자격 마스터(지급인) 계정에 링크되어 있거나 AWS 조직을 통해 연결된 새 계정은 이 제한에서 제외됩니다.
- AWS 조직 또는 AWS 조직의 조직 단위와 용량 예약을 공유하려면 AWS Organizations을 사용하여 공유를 사용하도록 설정해야 합니다. 자세한 내용은 AWS RAM 사용 설명서의 AWS Organizations을 사용하여 공유 사용 설정을 참조하십시오.

관련 서비스

용량 예약 공유는 AWS 리소스 액세스 관리자(AWS RAM)와 통합됩니다. AWS RAM은 어떤 AWS 계정이나 AWS Organizations을 통해 나의 AWS 리소스를 공유할 수 있게 해주는 서비스입니다. AWS RAM을 사용하여 리소스 공유로 생성한 사용자 소유 리소스를 공유할 수 있습니다. 리소스 공유는 공유할 리소스와 공유 대상 소비자를 지정합니다. 소비자는 개인 AWS 계정 또는 조직 단위 또는 AWS Organizations의 전체 조직일 수 있습니다.

AWS RAM에 대한 자세한 내용은 AWS RAM 사용 설명서를 참조하십시오.

여러 가용 영역 공유

리전의 가용 영역에 걸쳐 리소스가 배포될 수 있도록 AWS는 각 계정의 이름에 가용 영역을 독립적으로 매핑합니다. 이로 인해 계정 전체에서 가용 영역 이름의 차이가 발생할 수 있습니다. 예를 들어 AWS AWS 계정의 `us-east-1a` 가용 영역은 다른 AWS 계정에 대한 `us-east-1a` 가용 영역과 위치가 동일하지 않을 수 있습니다.

계정과 관련된 용량 예약 상대의 위치를 확인하려면 가용 영역 ID(AZ ID)를 사용해야 합니다. AZ ID는 모든 AWS 계정의 가용 영역에 대한 고유하고 일관된 식별자입니다. 예를 들어, `use1-az1`은 `us-east-1` 리전의 AZ ID이고, 모든 AWS 계정에서 위치가 동일합니다.

계정의 가용 영역에 대한 AZ ID 보는 방법

1. https://console.aws.amazon.com/ram에서 AWS RAM 콘솔을 엽니다.
2. 현재 지역의 AZ ID는 화면의 오른쪽에 있는 사용자 AZ ID 패널에 표시됩니다.

용량 예약 공유

자신이 소유한 용량 예약을 다른 AWS 계정과 공유하면 예약된 용량으로 인스턴스를 시작할 수 있습니다. 열린 용량 예약을 공유하는 경우 의도하지 않은 용량 예약 사용으로 이어질 수 있으므로 다음 사항을 명심하십시오.

- 소비자가 용량 예약의 속성과 일치하는 인스턴스를 실행하고 `CapacityReservationPreference` 매개변수를 `open`으로 설정하고 예약된 용량에서 아직 실행하지 않는 경우, 공유된 용량 예약을 자동으로 사용합니다.
- 소비자가 일치하는 속성(인스턴스 유형, 플랫폼 및 가용 영역)이 있는 인스턴스를 실행하고 `CapacityReservationPreference` 매개 변수를 `open`으로 설정하면 자동으로 공유된 용량 예약로 실행됩니다.

용량 예약을 공유하려면 리소스 공유에 추가해야 합니다. AWS RAM 리소스 공유는 여러 AWS 계정에서 리소스를 공유할 수 있게 해주는 리소스입니다. 리소스 공유는 공유할 리소스와 공유 대상 소비자를 지정합니다. Amazon EC2 콘솔을 사용하여 용량 예약을 공유하면 기존 리소스 공유에 추가합니다. 새 리소스 공유에 용량 예약을 추가하려면 AWS RAM 콘솔을 사용하여 리소스 공유를 만들어야 합니다.

AWS Organizations의 조직에 속해 있고 조직 내의 공유가 활성화되어 있으면 조직의 소비자는 공유된 용량 예약에 대한 액세스 권한이 자동으로 부여됩니다. 그렇지 않으면 소비자는 리소스 공유에 가입하라는 초대장을 받고 초대를 수락한 후 공유 용량 예약에 대한 액세스 권한이 부여됩니다.

Amazon EC2 콘솔, AWS RAM 콘솔 또는 AWS CLI를 사용하여 소유하고 있는 용량 예약을 공유할 수 있습니다.

Amazon EC2 콘솔을 사용하여 소유하고 있는 용량 예약을 공유하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 용량 예약를 선택합니다.
3. 공유할 용량 예약을 선택하고 작업, 공유 예약을 선택하십시오.
4. 용량 예약을 추가할 리소스 공유를 선택하고 용량 예약 공유를 선택하십시오.

   소비자가 공유 용량 예약에 액세스하려면 몇 분이 걸릴 수 있습니다.

AWS RAM 콘솔을 사용하여 소유하고 있는 용량 예약을 공유하려면

AWS RAM 사용 설명서에서 리소스 공유 만들기를 참조하십시오.

AWS CLI를 사용하여 소유하고 있는 용량 예약을 공유하려면

create-resource-share 명령을 사용합니다.

공유된 용량 예약의 공유 해제

용량 예약 소유자는 언제든지 공유 용량 예약을 공유 해제할 수 있습니다. 공유 용량 예약을 공유 해제하면 다음 규칙이 적용됩니다.

- 공유가 해제될 때 공유 용량으로 실행 중인 소비자 소유의 인스턴스는 예약된 용량 밖에서 정상적으로 계속 실행되며 용량은 Amazon EC2 용량 가용성에 따라 용량 예약로 복원됩니다.
- 용량 예약이 공유된 사용자는 더 이상 예약된 용량에 새로운 인스턴스를 시작할 수 없습니다.

소유하고 있는 공유 용량 예약을 공유 해제하려면 리소스 공유에서 제거해야 합니다. 이를 위해 Amazon EC2 콘솔, AWS RAM 콘솔 또는 AWS CLI를 사용할 수 있습니다.

Amazon EC2 콘솔을 사용하여 소유하고 있는 공유 용량 예약을 공유 해제하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 용량 예약를 선택합니다.
3. 공유를 해제할 용량 예약을 선택하고 공유 탭을 선택하십시오.
4. 공유 탭에는 용량 예약이 추가된 리소스 공유가 나열됩니다. 용량 예약을 제거할 리소스 공유를 선택하고 리소스 공유에서 제거를 선택하십시오.

AWS RAM 콘솔을 사용하여 소유하고 있는 공유 용량 예약을 공유 해제하려면

AWS RAM 사용 설명서에서 리소스 공유 업데이트를 참조하십시오.

AWS CLI를 사용하여 소유하고 있는 공유 용량 예약을 공유 해제하려면

disassociate-resource-share 명령을 사용합니다.

공유 용량 예약 확인

소유자와 소비자는 Amazon EC2 콘솔 및 AWS CLI를 사용하여 용량 예약 공유를 식별할 수 있으며

Amazon EC2 콘솔을 사용하여 용량 예약 공유를 식별하려면

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

2. 탐색 창에서 용량 예약를 선택합니다. 화면에는 내가 소유하고 있는 용량 예약와 나와 공유된 용량 예약가 나열됩니다. 소유자 열에는 용량 예약 소유자의 AWS 계정 ID가 표시됩니다. AWS 계정 ID 옆의 `(me)`는 사용자가 소유자임을 나타냅니다.

AWS CLI를 사용하여 용량 예약 공유를 식별하려면

describe-capacity-reservations 명령을 사용합니다. 이 명령은 내가 소유한 용량 예약와 나와 공유된 용량 예약를 반환합니다. `OwnerId`는 용량 예약 소유자의 AWS 계정 ID를 보여줍니다.

공유 용량 예약 사용량 보기

공유 용량 예약의 소유자는 Amazon EC2 콘솔과 AWS CLI를 사용하여 언제든지 사용량을 볼 수 있습니다.

Amazon EC2 콘솔을 사용하여 용량 예약 사용량을 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 용량 예약를 선택합니다.
3. 사용량을 볼 수 있는 용량 예약 항목을 선택하고 사용량 탭을 선택하십시오.

   AWS 계정 ID 열에는 현재 용량 예약을 사용하는 소비자의 계정 ID가 표시됩니다. 시작된 인스턴스 열에는 현재 각 사용자가 예약된 용량으로 실행 중인 인스턴스 수가 표시됩니다.

AWS CLI를 사용하여 용량 예약 사용량을 보려면

get-capacity-reservation-usage 명령을 사용하십시오. `AccountId`는 용량 예약을 사용하는 계정의 계정 ID를 보여줍니다. `UsedInstanceCount`는 현재 예약된 용량에서 실행 중인 인스턴스의 수를 보여줍니다.

공유 용량 예약 권한

소유자에 대한 사용 권한

소유자는 공유 용량 예약을 관리하고 취소하는 일을 담당합니다. 소유자는 다른 계정이 소유한 공유 용량 예약에서 실행 중인 인스턴스를 수정할 수 없습니다. 소유자는 용량 예약 공유로 실행되는 인스턴스를 관리해야 합니다.

소비자를 위한 권한

소비자는 공유 용량 예약을 실행하는 인스턴스를 관리해야 합니다. 소비자는 어떤 식으로든 공유 용량 예약을 수정할 수 없으며 다른 소비자 또는 용량 예약 소유자가 소유한 인스턴스를 보거나 수정할 수 없습니다.

결제 및 측정

용량 예약 공유에 대한 추가 비용은 없습니다.

용량 예약 소유자에게는 용량 예약 내부에서 실행하는 인스턴스 및 사용되지 않은 예약 용량에 대해 요금이 청구됩니다. 소비자는 용량 예약 공유 내에서 실행되는 인스턴스에 대해 요금이 청구됩니다.

인스턴스 제한

모든 용량 예약 사용량은 용량 예약 소유자의 온디맨드 인스턴스 한도에 포함됩니다. 여기에는 다음이 포함됩니다.

- 미사용 예약 용량
- 용량 예약 소유자가 소유한 인스턴스 별 사용량

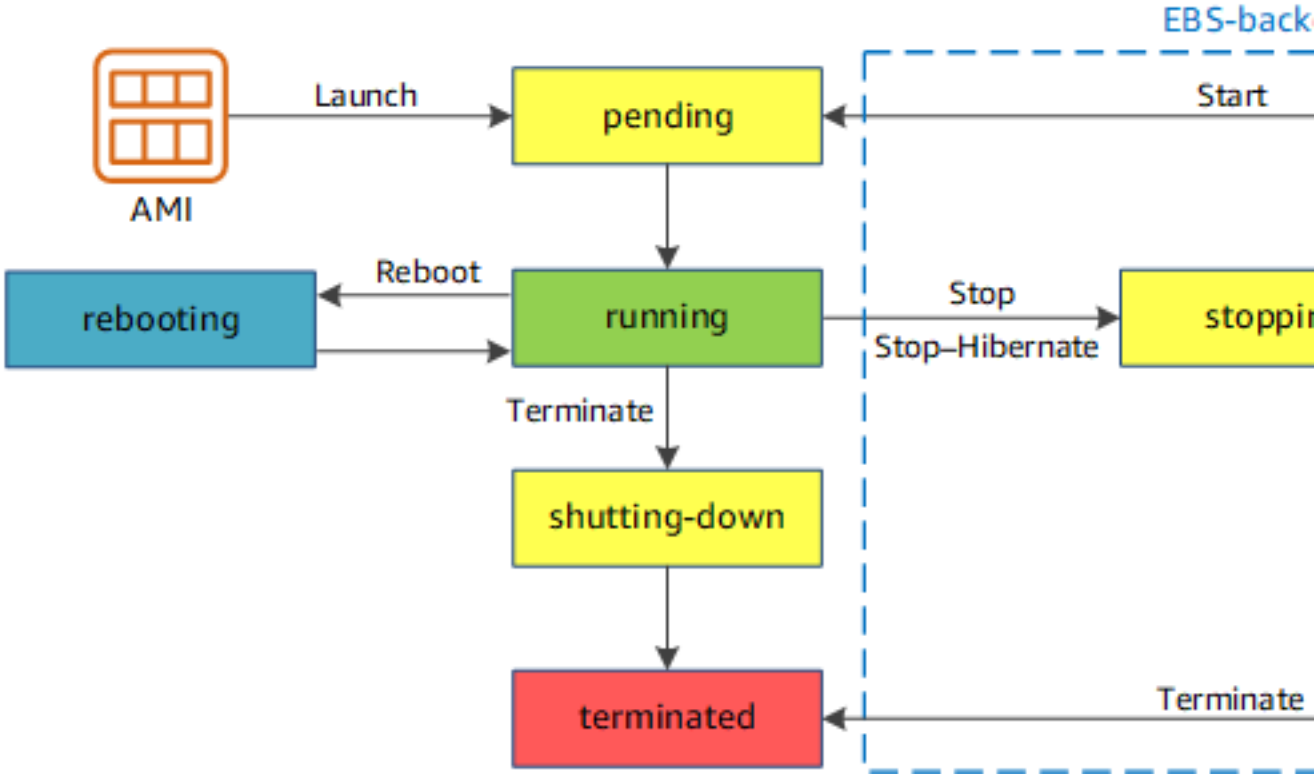
- 소비자가 소유한 인스턴스 별 사용량

소비자가 공유하는 용량으로 시작된 인스턴스는 용량 예약 소유자의 온디맨드 인스턴스 한도에 포함됩니다. 소비자의 인스턴스 제한은 자신의 온디맨드 인스턴스 제한과 액세스할 수 있는 공유 용량 예약에서 사용 가능한 용량의 합계입니다.

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. 93) 단원을 참조하십시오.



다음 표는 각 인스턴스 상태에 대한 간략한 정보와 청구 여부를 설명합니다.

> **Note**
>
> 이 표는 인스턴스 사용에 관련된 청구만 보여줍니다. Amazon EBS 볼륨 및 탄력적 IP 주소와 같은 일부 AWS 리소스는 인스턴스의 상태와 상관없이 요금을 부과합니다. 자세한 내용은 AWS Billing and Cost Management 사용 설명서에서 예기치 않은 비용 방지 단원을 참조하십시오.

인스턴스 상태	설명	인스턴스 사용 요금
`pending`	인스턴스는 `running` 상태로 될 준비를 하고 있습니다.	미청구

인스턴스 상태	설명	인스턴스 사용 요금
	인스턴스를 처음 시작하거나 `stopped` 상태의 인스턴스를 다시 시작하면 `pending` 상태가 됩니다.	
`running`	인스턴스를 실행하고 사용할 준비가 되었습니다.	청구
`stopping`	인스턴스가 중지 또는 중지-최대 절전 모드로 전환할 준비를 하고 있습니다.	중지 준비 중인 경우 미청구 최대 절전 모드로 전환 준비 중인 경우 청구
`stopped`	인스턴스가 종료되고 사용이 불가합니다. 언제든지 인스턴스를 다시 시작할 수 있습니다.	미청구
`shutting down`	인스턴스가 종료할 준비를 하고 있습니다.	미청구
`terminated`	인스턴스가 영구적으로 삭제되었으며 시작할 수 없습니다.	미청구 Note 종료된 인스턴스에 적용되는 예약 인스턴스는 결제 옵션에 따라 기간이 종료될 때까지 요금이 청구됩니다. 자세한 내용은 예약 인스턴스 (p. 273) 단원을 참조하십시오.

Note

인스턴스를 재부팅해도 `running` 상태로 남아있기 때문에 새 인스턴스 청구 기간이 시작되지 않습니다.

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 초(최소 1분)에 대해 비용이 청구됩니다. 인스턴스가 유휴 상태이고 인스턴스에 연결하지 않더라도 마찬가지입니다.

자세한 내용은 인스턴스 시작 (p. 430) 및 Linux 인스턴스에 연결 (p. 485) 단원을 참조하십시오.

인스턴스 중지 및 시작(Amazon EBS 기반 인스턴스에만 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 사용 요금이나 데이터 전송 요금이 부과되지는 않지만 모든 Amazon EBS 볼륨에 대한 스토리지 요금은 부과됩니다. 인스턴스가 stopped상태인 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 계속 있을 수 있습니다. 인스턴스를 중지했다가 시작하면 이전 호스트 컴퓨터의 인스턴스 스토어 볼륨에 있는 데이터가 모두 손실됩니다.

인스턴스에서 프라이빗 IPv4 주소가 유지됩니다. 즉, 프라이빗 IPv4 주소 또는 네트워크 인터페이스와 연결된 탄력적 IP 주소가 여전히 인스턴스와 연결되어 있다는 의미입니다. 인스턴스에 IPv6 주소가 있는 경우 해당 IPv6 주소를 유지합니다.

stopped에서 running으로 인스턴스를 전환할 때마다 인스턴스 실행 시간에 대해 초 단위로 요금을 청구하며, 인스턴스를 시작할 때마다 최소 1분의 요금이 부과됩니다.

자세한 내용은 인스턴스 중지 및 시작 (p. 507) 단원을 참조하십시오.

인스턴스 최대 절전 모드(Amazon EBS 지원 인스턴스에만 해당)

인스턴스를 최대 절전 모드로 전환하면 운영 체제에 최대 절전 모드(suspend-to-disk)를 수행하도록 알립니다. 그러면 인스턴스 메모리(RAM)의 콘텐츠를 Amazon EBS 루트 볼륨에 저장합니다. 인스턴스의 Amazon EBS 루트 볼륨과 연결된 모든 Amazon EBS 데이터 볼륨을 유지합니다. 인스턴스를 시작하면 Amazon EBS 루트 볼륨이 이전 상태로 복원되고, RAM 콘텐츠가 다시 로드됩니다. 이전에 연결된 데이터 볼륨이 다시 연결되고, 인스턴스는 해당 인스턴스 ID를 유지합니다.

인스턴스를 최대 절전 모드로 전환하면 stopping 상태로 전환되고 나서 stopped 상태로 전환됩니다. 최대 절전 모드로 전환하지 않고 인스턴스를 중지 (p. 427)한 경우와 달리 최대 절전 모드 인스턴스가 stopped 상태이면 해당 인스턴스에 대해서는 사용 요금을 청구할 수 없지만 stopping 상태일 때 비용이 청구됩니다. 데이터 전송에 대해 사용 요금이 부과되지는 않지만 RAM 데이터에 대한 스토리지를 포함해 모든 Amazon EBS 볼륨에 대한 스토리지 요금은 부과됩니다.

최대 절전 모드의 인스턴스를 시작하면 인스턴스가 pending 상태로 전환되며, 대부분의 경우 AWS는 그 인스턴스를 새 호스트 컴퓨터로 이동합니다. 호스트 컴퓨터에 문제가 없으면 인스턴스는 같은 호스트 컴퓨터에 계속 있을 수 있습니다.

인스턴스에서 프라이빗 IPv4 주소가 유지됩니다. 즉, 프라이빗 IPv4 주소 또는 네트워크 인터페이스와 연결된 탄력적 IP 주소가 여전히 인스턴스와 연결되어 있다는 의미입니다. 인스턴스에 IPv6 주소가 있는 경우 해당 IPv6 주소를 유지합니다.

자세한 내용은 Linux 인스턴스를 최대 절전 모드로 실행합니다. (p. 510) 단원을 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.

인스턴스 재부팅은 운영 체제 재부팅과 같습니다. 인스턴스가 동일한 호스트 컴퓨터에 남아 있고, 퍼블릭 DNS 이름, 프라이빗 IP 주소 및 인스턴스 스토어 볼륨의 모든 데이터가 유지됩니다. 일반적으로 재부팅을 완료하는 데 몇 분 정도 소요되지만, 재부팅 소요 시간은 인스턴스 구성에 따라 달라집니다.

인스턴스를 재부팅하면 인스턴스 청구 이 새로 시작되지 않으며, 최소 1분 요금 부과 없이 초 단위 결제가 계속됩니다.

자세한 내용은 인스턴스 재부팅 (p. 520) 단원을 참조하십시오.

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. 인스턴스 루트 디바이스가 인스턴스 스토어 볼륨인 경우 인스턴스가 종료되어 다시 사용할 수 없습니다.

자세한 내용은 인스턴스 만료 (p. 520) 단원을 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 중단된 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. 522) 단원을 참조하십시오.

재부팅, 중지, 최대 절전 모드 및 종료의 차이

다음 표에는 인스턴스 재부팅, 중지, 최대 절전 모드 및 종료의 주요 차이점이 요약되어 있습니다.

특성	재부팅	중지/시작(Amazon EBS 기반 인스턴스에만 해당)	최대 절전 모드 (Amazon EBS 지원 인스턴스에만 해당)	Terminate
호스트 컴퓨터	인스턴스가 동일 호스트 컴퓨터에서 유지됩니다.	대부분의 경우 인스턴스를 새 호스트 컴퓨터로 이동합니다. 호스트 컴퓨터에 문제가 없으면 인스턴스는 같은 호스트 컴퓨터에 계속 있을 수 있습니다.	대부분의 경우 인스턴스를 새 호스트 컴퓨터로 이동합니다. 호스트 컴퓨터에 문제가 없으면 인스턴스는 같은 호스트 컴퓨터에 계속 있을 수 있습니다.	없음
프라이빗 및 퍼블릭 IPv4 주소	이러한 주소는 동일하게 유지됩니다.	인스턴스가 관련 프라이빗 IPv4 주소를 유지합니다. 중지/시작 중에 변경되지 않는 탄력적 IP 주소가 지정되지 않는 한, 인스턴스가 새 퍼블릭 IPv4 주소를 가져옵니다.	인스턴스가 관련 프라이빗 IPv4 주소를 유지합니다. 중지/시작 중에 변경되지 않는 탄력적 IP 주소가 지정되지 않는 한, 인스턴스가 새 퍼블릭 IPv4 주소를 가져옵니다.	없음

특성	재부팅	중지/시작(Amazon EBS 기반 인스턴스에만 해당)	최대 절전 모드 (Amazon EBS 지원 인스턴스에만 해당)	Terminate
탄력적 IP 주소(IPv4)	탄력적 IP 주소가 인스턴스와 연결된 상태로 유지됩니다.	탄력적 IP 주소가 인스턴스와 연결된 상태로 유지됩니다.	탄력적 IP 주소가 인스턴스와 연결된 상태로 유지됩니다.	인스턴스로부터 탄력적 IP 주소 연결이 끊깁니다.
IPv6 주소	주소가 동일하게 유지됩니다.	인스턴스가 관련 IPv6 주소를 유지합니다.	인스턴스가 관련 IPv6 주소를 유지합니다.	없음
인스턴스 스토어 볼륨	데이터가 유지됩니다.	데이터가 지워집니다.	데이터가 지워집니다.	데이터가 지워집니다.
루트 디바이스 볼륨	볼륨이 유지됩니다.	볼륨이 유지됩니다.	볼륨이 유지됩니다.	볼륨이 기본적으로 삭제됩니다.
RAM(메모리의 콘텐츠)	RAM이 지워집니다.	RAM이 지워집니다.	RAM은 루트 볼륨의 파일에 저장됩니다.	RAM이 지워집니다.
결제	인스턴스 결제 시간이 변경되지 않습니다.	상태가 `stopping`으로 변경되는 즉시 인스턴스에 대한 요금 발생이 중지됩니다. 인스턴스 상태가 `stopped`에서 `running`으로 전환될 때마다 새로운 인스턴스 결제 이 시작되며, 인스턴스를 시작할 때마다 최소 1분의 요금이 부과됩니다.	인스턴스가 `stopping` 상태이면 비용이 발생하지만 `stopped` 상태일 때는 비용이 발생하지 않습니다. 인스턴스 상태가 `stopped`에서 `running`으로 전환될 때마다 새로운 인스턴스 결제 이 시작되며, 인스턴스를 시작할 때마다 최소 1분의 요금이 부과됩니다.	상태가 `shutting-down`으로 변경되는 즉시 인스턴스에 대한 요금 발생이 중지됩니다.

운영 체제 종료 명령을 실행하면 항상 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. 525) 단원을 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 티어를 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 요금을 참조하십시오.

다음 방법을 사용하여 인스턴스를 시작할 수 있습니다.

방법	설명서
[Amazon EC2 콘솔] 시작 인스턴스 마법사를 사용하여 시작 파라미터를 지정	인스턴스 시작 마법사를 사용하여 인스턴스 시작 (p. 431)
[Amazon EC2 콘솔] 시작 템플릿을 생성하고 이 시작 템플릿에서 인스턴스를 시작	시작 템플릿에서 인스턴스 시작 (p. 436)

방법	설명서
[Amazon EC2 콘솔] 기존 인스턴스를 기본 템플릿으로 사용	기존 인스턴스의 파라미터를 사용하여 인스턴스 시작 (p. 446)
[Amazon EC2 콘솔] 생성한 Amazon EBS 스냅샷 사용	백업에서 Linux 인스턴스 시작 (p. 447)
[Amazon EC2 콘솔] AWS Marketplace에서 구매한 AMI 사용	AWS Marketplace 인스턴스 시작 (p. 448)
[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)
[AWS CLI] EC2 플릿을 사용하여 여러 EC2 인스턴스 유형, 가용 용역 및 온디맨드 인스턴스, 예약 인스턴스 및 스팟 인스턴스 구매 모델에 걸쳐 용량을 프로비저닝합니다.	EC2 플릿 시작 (p. 449)

인스턴스를 시작할 때 다음 리소스 중 하나에 연결된 서브넷에서 인스턴스를 시작할 수 있습니다.

- 가용 영역 - 이 옵션이 기본값입니다.
- 로컬 영역 - 로컬 영역에서 인스턴스를 시작하려면 기능을 옵트인해야 합니다. 자세한 내용은 로컬 영역 옵트인을 참조하십시오.
- Outpost - Outpost에서 인스턴스를 시작하려면 Outpost를 만들어야 합니다. Outpost를 만드는 방법에 대한 자세한 내용은 AWS Outposts 사용 설명서의 AWS Outposts 시작하기를 참조하십시오.

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 네트워크(EC2-Classic 또는 EC2-VPC) 내 다른 인스턴스에서 해당 인스턴스에 접속할 때 사용됩니다. 인스턴스 연결에 대한 자세한 내용은 Linux 인스턴스에 연결 (p. 485) 단원을 참조하십시오.

인스턴스 작업을 완료한 후에는 반드시 인스턴스를 삭제하십시오. 자세한 내용은 인스턴스 종료 (p. 522) 단원을 참조하십시오.

인스턴스 시작 마법사를 사용하여 인스턴스 시작

인스턴스 시작 마법사를 사용하여 인스턴스를 시작할 수 있습니다. 인스턴스 시작 마법사는 인스턴스를 시작하는 데 필요한 모든 시작 파라미터를 지정합니다. 인스턴스 시작 마법사에서 기본값을 제공하는 경우 기본값을 그대로 사용하거나 고유한 값을 지정할 수 있습니다. 인스턴스를 시작하려면 최소한 AMI와 키 페어를 선택해야 합니다.

인스턴스를 시작하기 전에 설정을 확인합니다. 자세한 내용은 Amazon EC2로 설정 (p. 22) 단원을 참조하십시오.

> **Important**
>
> 시작하는 인스턴스가 AWS 프리 티어에 해당되지 않는 경우, 유휴 상태를 포함해 인스턴스가 실행된 시간에 대하여 과금이 청구됩니다.

인스턴스를 시작하는 단계는 다음과 같습니다.

- 인스턴스 시작 개시 (p. 432)
- 1단계: Amazon 머신 이미지(AMI) 선택 (p. 432)

- 2단계: 인스턴스 유형 선택 (p. 433)
- 3단계: 인스턴스 세부 정보 구성 (p. 433)
- 4단계: 스토리지 추가 (p. 435)
- 5단계: 태그 추가 (p. 435)
- 6단계: 보안 그룹 구성 (p. 435)
- 7단계: 인스턴스 시작 검토 및 키 페어 선택 (p. 436)

인스턴스 시작 개시

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 화면 상단의 탐색 모음에는 현재 리전이 표시됩니다(예: 미국 동부(오하이오)). 요구에 맞는 인스턴스의 리전을 선택합니다. 일부 Amazon EC2 리소스는 리전 간에 공유될 수 있지만 그렇지 않은 리소스도 있으므로 잘 선택해야 합니다. 자세한 내용은 리소스 위치 (p. 1059) 단원을 참조하십시오.
3. Amazon EC2 콘솔 대시보드에서 인스턴스 시작을 선택합니다.

1단계: Amazon 머신 이미지(AMI) 선택

Amazon Machine Image(AMI) 선택 페이지에서 다음에 따라 AMI를 선택합니다.

1. 왼쪽 창에서 사용할 AMI의 유형을 선택합니다:

   빠른 시작

   빠른 시작을 도와주는 인기 AMI를 선별하여 보여줍니다. 프리 티어로 이용할 수 있는 AMI만 선택하려면 왼쪽 창에서 프리 티어만을 선택합니다. 이러한 AMI는 프리 티어 사용 가능으로 표시됩니다.

   My AMIs

   사용자가 소유한 프라이빗 AMI 또는 공유된 프라이빗 AMI입니다. 사용자가 공유되는 AMI를 보려면 왼쪽 창에서 나와 공유됨을 선택합니다.

   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. 448)을 참조하십시오.

   Community AMIs

   AWS 커뮤니티 멤버가 다른 사람의 사용을 허용하여 게시한 AMI입니다. 운영 체제에 따라 AMI 목록을 필터링하려면 운영 체제 아래의 확인란을 선택하십시오. 이 외에도 아키텍처나 루트 디바이스 타입에 따라 필터링할 수 있습니다.
2. 각 AMI의 지원 루트 디바이스 유형 목록을 확인합니다. `ebs`(Amazon EBS에서 지원 유형) 또는 `instance-store`(인스턴스 스토어에서 지원) 중 필요한 유형의 AMI를 확인하십시오. 자세한 내용은 루트 디바이스 스토리지 (p. 93) 단원을 참조하십시오.
3. 각 AMI의 지원 가상화 유형 목록을 확인합니다. `hvm` 또는 `paravirtual` 중 필요한 유형의 AMI를 확인하십시오. 예를 들어 일부 인스턴스 유형은 HVM이 필요합니다. 자세한 내용은 Linux AMI 가상화 유형 (p. 95) 단원을 참조하십시오.
4. 용도에 적합한 AMI를 선택하고 선택 버튼을 선택합니다.

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).

2단계: 인스턴스 유형 선택

인스턴스 유형 선택 페이지에서 시작할 인스턴스의 하드웨어 구성 및 크기를 선택합니다. 대형 인스턴스는 CPU와 메모리가 더 높습니다. 자세한 내용은 인스턴스 유형 (p. 178) 단원을 참조하십시오.

t2.micro 인스턴스 유형을 선택하면 프리 티어 사용 자격을 유지할 수 있습니다. 자세한 내용은 성능 순간 확장 가능 인스턴스 (p. 194) 단원을 참조하십시오.

기본 설정에서 마법사는 현 세대의 인스턴스 유형을 표시하고 사용자가 선택한 AMI를 기반으로 하여 첫 번째로 사용 가능한 유형을 선택합니다. 필터 목록에서 모든 세대를 선택하면 이전 세대의 인스턴스 유형을 볼 수 있습니다.

> **Note**
>
> 테스트 목적으로 인스턴스를 빠르게 설정하려는 경우, 검토 및 시작(Review and Launch)을 선택하여 기본 구성 설정을 적용하고 인스턴스를 시작합니다. 그렇지 않은 경우 다음: 인스턴스 세부 정보 구성(Next: Configure Instance Details)을 선택해 인스턴스를 세부 구성할 수 있습니다.

3단계: 인스턴스 세부 정보 구성

인스턴스 세부 정보 구성(Configure Instance Details) 페이지에서 필요에 맞게 다음 설정을 변경하고(모든 설정 항목을 확장 표시하려면 고급 세부 정보 클릭), 다음: 스토리지 추가(Next: Add Storage)를 선택합니다.

- 인스턴스 개수: 시작할 인스턴스의 수를 입력합니다.

  > **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.
- (선택 사항) 애플리케이션 수요를 처리할 인스턴스의 수를 올바르게 유지하는 데 도움을 주기 위해 Auto Scaling 그룹 시작을 선택해 시작 구성 및 Auto Scaling 그룹을 생성할 수 있습니다. Auto Scaling은 사양에 따라 그룹에서 인스턴스의 수를 조정합니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서를 참조하십시오.
- 구입 옵션: 스팟 인스턴스를 시작하려면 스팟 인스턴스 요청을 선택합니다. 이렇게 하여 이 페이지에 선택 사항을 추가하거나 제거합니다. 최대 가격을 설정하고, 필요에 따라 요청 유형과 중단 동작 및 요청 유효성을 업데이트합니다. 자세한 내용은 스팟 인스턴스 요청 생성 (p. 329) 단원을 참조하십시오.
- 네트워크 VPC를 선택하거나 새 VPC 생성을 선택하여 Amazon VPC 콘솔로 이동해 새 VPC를 생성합니다. 마친 후에 마법사로 돌아와 새로 고침을 선택하면 해당 VPC가 목록에 로딩됩니다.
- 서브넷: 가용 영역, 로컬 영역 또는 Outpost와 연결된 서브넷에서 인스턴스를 시작할 수 있습니다.

  가용 영역에서 인스턴스를 시작하려면 인스턴스를 시작할 서브넷을 선택하십시오. 기본 설정 없음을 선택하여 AWS에서 임의의 가용 영역 내 기본 서브넷을 선택할 수 있습니다. 새 서브넷을 생성하려면 새 서브넷 생성을 선택하여 Amazon VPC 콘솔로 이동합니다. 마친 후에 마법사로 돌아와 새로 고침을 선택하면 해당 서브넷이 목록에 로딩됩니다.

  로컬 영역에서 인스턴스를 시작하려면 로컬 영역에서 만든 서브넷을 선택하십시오.

  Outpost에서 인스턴스를 시작하려면 Outpost와 연결된 VPC의 서브넷을 선택하십시오.
- 퍼블릭 IP 자동 할당: 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. 활성화 또는 비활성화를 선택하여 서브넷의 기본 설정을 재정의할 수 있습니다. 자세한 내용은 퍼블릭 IPv4 주소 및 외부 DNS 호스트 이름 (p. 659) 단원을 참조하십시오.
- 자동 할당 IPv6 IP: 인스턴스가 서브넷 범위 내에서 IPv6 주소를 수신할지 지정합니다. 활성화 또는 비활성화를 선택하여 서브넷의 기본 설정을 재정의합니다. 이 옵션은 IPv6 CIDR 블록에 VPC와 서브넷을 연결한 경우에만 사용할 수 있습니다. 자세한 내용은 Amazon VPC 사용 설명서의 VPC 및 서브넷을 참조하십시오.

- 용량 예약: 인스턴스를 공유 용량 또는 기존 용량 예약으로 시작할지 여부를 지정합니다. 자세한 내용은 인스턴스를 기존 용량 예약으로 시작 (p. 419) 단원을 참조하십시오.
- IAM 역할: 인스턴스와 연결할 AWS Identity and Access Management(IAM) 역할을 선택합니다. 자세한 내용은 IAM의 Amazon EC2 역할 (p. 850) 단원을 참조하십시오.
- CPU options(CPU 옵션): Specify CPU options(CPU 옵션 지정)를 선택해 시작 중 vCPU의 수를 사용자 지정할 수 있습니다. CPU 코어 수와 코어당 스레드 수를 설정합니다. 자세한 내용은 CPU 옵션 최적화 (p. 546) 단원을 참조하십시오.
- 종료 동작: 인스턴스 종료 시 적용할 인스턴스 상태(중지 또는 종료)를 선택합니다. 자세한 정보는 인스턴스가 개시하는 종료 동작 변경 (p. 525) 단원을 참조하십시오.
- 중지 - 최대 절전 모드 동작: 최대 절전 모드를 활성화하려면 이 확인란을 선택하십시오. 이 옵션은 인스턴스가 최대 절전 모드 필수 조건을 충족하는 경우에만 사용할 수 있습니다. 자세한 내용은 Linux 인스턴스를 최대 절전 모드로 실행합니다. (p. 510) 단원을 참조하십시오.
- 종료 방지 기능 활성화: 실수로 인스턴스를 종료하는 일을 방지하려면 이 확인란을 선택합니다. 자세한 내용은 인스턴스에 대한 종료 방지 기능 활성화 (p. 524) 단원을 참조하십시오.
- 모니터링: 이 확인란을 선택하면 Amazon CloudWatch 사용을 통한 인스턴스 세부 모니터링 기능이 활성화됩니다. 추가 요금이 발생합니다. 자세한 내용은 CloudWatch를 사용해 인스턴스 모니터링하기 (p. 616) 단원을 참조하십시오.
- EBS 최적화 인스턴스: Amazon EBS 최적화 인스턴스는 최적화된 구성 스택을 사용하며 Amazon EBS I/O를 위한 추가 전용 용량을 제공합니다. 인스턴스 유형이 이 기능을 지원하는 경우 이 확인란을 선택하여 활성화합니다. 추가 요금이 발생합니다. 자세한 내용은 Amazon EBS 최적화 인스턴스 (p. 984) 단원을 참조하십시오.
- 테넌시: VPC로 인스턴스를 시작하는 경우 격리된 전용 하드웨어(전용) 또는 전용 호스트(전용 호스트)에서 인스턴스를 실행하도록 선택할 수 있습니다. 추가 요금이 적용될 수 있습니다. 자세한 내용은 전용 인스턴스 (p. 410) 및 전용 호스트 (p. 381) 단원을 참조하십시오.
- T2/T3 무제한: 애플리케이션이 필요한 시간만큼 기준 이상으로 버스트를 할 수 있도록 하려면 이 확인란을 선택합니다. 추가 요금이 적용될 수 있습니다. 자세한 내용은 성능 순간 확장 가능 인스턴스 (p. 194) 단원을 참조하십시오.
- 네트워크 인터페이스: 특정 서브넷을 선택한 경우, 인스턴스에 대해 네트워크 인터페이스를 최대 2개까지 지정할 수 있습니다.
  - 네트워크 인터페이스의 경우, AWS에서 새로운 인터페이스가 자동으로 생성되도록 새 네트워크 인터페이스를 선택하거나 사용 가능한 기존 네트워크 인터페이스를 선택합니다.
  - 기본 IP의 경우, 서브넷 범위에서 프라이빗 IPv4 주소를 입력하거나 AWS에서 프라이빗 IPv4 주소가 자동으로 선택되도록 자동 할당을 그대로 둡니다.
  - 보조 IP 주소에서 IP 추가를 선택하면 선택한 네트워크 인터페이스에 프라이빗 IPv4 주소를 두 개 이상 할당할 수 있습니다.
  - (IPv6 전용) IPv6 IP에서 IP 추가를 선택하고 서브넷 범위에서 IPv6 주소를 입력하거나 AWS가 자동으로 선택하도록 자동 할당을 그대로 둡니다.
  - 디바이스 추가를 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. 683) 단원을 참조하십시오. 네트워크 인터페이스를 두 개 이상 지정하면 인스턴스가 퍼블릭 IPv4 주소를 수신할 수 없습니다. 또한 eth0에 대해 기존 네트워크 인터페이스를 지정하면 퍼블릭 IP 자동 할당을 사용하여 서브넷의 퍼블릭 IPv4 설정을 재정의할 수 없습니다. 자세한 내용은 인스턴스 시작 시 퍼블릭 IPv4 주소 배정 (p. 663) 단원을 참조하십시오.
- 커널 ID: (반가상화(PV) AMIs만 해당) 특정 커널을 사용하려는 경우가 아니라면 기본값 사용을 선택합니다.
- RAM 디스크 ID: (반가상화(PV) AMIs만 해당) 특정 RAM 디스크를 사용하려는 경우가 아니라면 기본값 사용을 선택합니다. 커널을 선택해 사용할 때는 해당 커널을 지원하는 드라이버가 설치된 RAM 디스크 지정이 필요할 수 있습니다.
- 배치 그룹: 배치 그룹은 인스턴스의 배치 전략을 결정합니다. 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. 757) 단원을 참조하십시오.

- 사용자 데이터: 시작 과정에서 인스턴스를 구성하거나 구성 스크립트를 실행할 때 사용할 사용자 데이터를 지정할 수 있습니다. 파일을 첨부하려면 파일 옵션을 선택하여 첨부할 파일을 선택하십시오.

4단계: 스토리지 추가

선택한 AMI에는 루트 디바이스 볼륨을 포함한 하나 이상의 스토리지 볼륨이 있습니다. 스토리지 추가 페이지에서 새 볼륨 추가를 선택하여 인스턴스에 연결할 추가 볼륨을 지정할 수 있습니다. 각 볼륨을 다음과 같이 구성한 후, 다음: 태그 추가를 선택합니다.

- 유형: 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. 1027) 및 Amazon EBS 볼륨 (p. 890) 단원을 참조하십시오.
- 디바이스: 볼륨에서 사용할 디바이스 이름을 목록에서 선택합니다.
- 스냅샷: 볼륨 복원에 사용할 스냅샷의 이름이나 ID를 입력합니다. 또는 스냅샷 필드에 텍스트를 입력하여 사용 가능한 공유 및 퍼블릭 스냅샷을 검색할 수 있습니다. 스냅샷 정보는 대/소문자를 구분합니다.
- 크기: EBS 볼륨의 경우, 스토리지 크기를 지정할 수 있습니다. 선택한 AMI와 인스턴스가 프리 티어에 해당되는 경우에도 프리 티어 한도를 유지하려면 총 스토리지 크기를 30GiB 미만으로 유지해야 합니다. 자세한 내용은 EBS 볼륨의 크기 및 구성에 대한 제약 (p. 905) 단원을 참조하십시오.
- 볼륨 유형: EBS 볼륨에 대한 볼륨 유형을 선택합니다. 자세한 내용은 Amazon EBS 볼륨 유형 (p. 892) 단원을 참조하십시오.
- IOPS: 프로비저닝된 IOPS SSD 볼륨 유형을 선택한 경우, 볼륨에서 지원되는 초당 I/O(IOPS) 수를 입력할 수 있습니다.
- 종료 시 삭제 여부: Amazon EBS 볼륨에 적용되는 기능으로, 확인란을 선택하면 인스턴스 종료 시 볼륨을 삭제합니다. 자세한 정보는 인스턴스 종료 시 Amazon EBS 볼륨 보존 (p. 526) 단원을 참조하십시오.
- 암호화: 인스턴스 유형이 EBS 암호화를 지원하는 경우, 볼륨의 암호화 상태를 지정할 수 있습니다. 이 리전에서 기본적으로 암호화를 활성화한 경우 사용자에 대해 기본 CMK가 선택됩니다. 다른 키를 선택하거나 암호화를 비활성화할 수 있습니다. 자세한 내용은 Amazon EBS Encryption (p. 968) 단원을 참조하십시오.

5단계: 태그 추가

태그 추가 페이지에서 키와 값의 조합을 제공하여 태그 (p. 1068)를 지정합니다. 인스턴스 또는 볼륨 또는 이 둘 모두에 태그를 지정할 수 있습니다. 스팟 인스턴스의 경우 스팟 인스턴스 요청에만 태그를 지정할 수 있습니다. 리소스에 2개 이상의 태그를 추가하려면 다른 태그 추가를 선택합니다. 모두 마쳤으면 다음: 보안 그룹 구성(Next: Configure Security Group)을 선택합니다.

6단계: 보안 그룹 구성

보안 그룹 구성 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 Amazon EC2 인스턴스에 대한 Linux 보안 그룹 (p. 872)을 참조하십시오.) 다음 과정에 따라 그룹을 선택하거나 새로 생성하고 검토 후 시작(Review and Launch)을 선택합니다.

- 기존 보안 그룹을 선택하려면 Select an existing security group(기존 보안 그룹 선택)을 선택하고 원하는 보안 그룹을 선택합니다. 기존의 보안 그룹 규칙은 수정할 수 없으며, 대신 새로 복사를 선택하여 새 보안 그룹으로 규칙을 복사할 수 있습니다. 다음 단계의 설명에 따라 규칙을 추가할 수 있습니다.
- 새 보안 그룹을 만들려면 새 보안 그룹 생성을 선택합니다. 마법사에서 자동으로 launch-wizard-x 보안 그룹을 정의하고, SSH(22번 포트)를 통한 인스턴스 연결을 허용하는 인바운드 규칙을 생성합니다.
- 규칙은 필요에 따라 추가할 수 있습니다. 예를 들어 웹 서버인 인스턴스는 80번 포트(HTTP)와 443번 포트(HTTPS)를 개방해 인터넷 트래픽을 허용할 수 있습니다.

  규칙을 추가하려면 규칙 추가를 선택한 다음 네트워크 트래픽의 개방 프로토콜을 선택하고 소스를 지정합니다. 소스 목록에서 내 IP를 선택하면 마법사에서 사용자 컴퓨터의 퍼블릭 IP 주소가 자동으로 추가됩니

다. 하지만 고정 IP 주소 없이 방화벽 뒤에서 또는 ISP를 통해 연결하는 경우에는 클라이언트 컴퓨터가 사용하는 IP 주소의 범위를 찾아야 합니다.

> **Warning**
>
> 모든 IP 주소(`0.0.0.0/0`)가 SSH 또는 RDP를 통해 인스턴스에 액세스할 수 있도록 허용하는 규칙은 이 짧은 예제에서만 사용하고, 프로덕션 환경에서는 위험하니 사용하지 마십시오. 특정 주소나 IP 주소 범위에서만 인스턴스 액세스를 허용하도록 설정해야 합니다.

7단계: 인스턴스 시작 검토 및 키 페어 선택

인스턴스 시작 검토 페이지에서 인스턴스 세부 정보를 확인한 다음, 해당되는 편집 링크를 선택하여 필요한 사항을 변경합니다.

준비가 완료되면 시작을 선택합니다.

기존 키 페어 선택 또는 새 키 페어 생성(Select an existing key pair or create a new key pair) 대화 상자에서 기존 키 페어를 선택하거나 새 키 페어를 만들 수 있습니다. 예를 들어, 기존 키 페어 선택를 선택하고 초기 설정에서 생성한 키 페어를 선택합니다.

인스턴스를 시작하려면 승인 확인란을 선택한 후 인스턴스 시작을 선택합니다.

> **Important**
>
> 키 페어 없이 진행(Proceed without key pair) 옵션을 선택할 경우 사용자가 다른 방법으로 로그인할 수 있도록 구성된 AMI를 선택해야만 인스턴스에 연결할 수 있습니다.

(선택 사항) 인스턴스의 상태 확인 경보를 생성할 수 있습니다(추가 비용 적용 가능). (지금 결정하지 않아도 언제든지 나중에 추가할 수 있습니다.) 확인 화면에서 상태 검사 경보 생성을 선택하여 지침에 따릅니다. 자세한 내용은 상태 확인 경보 생성 및 수정 (p. 606) 단원을 참조하십시오.

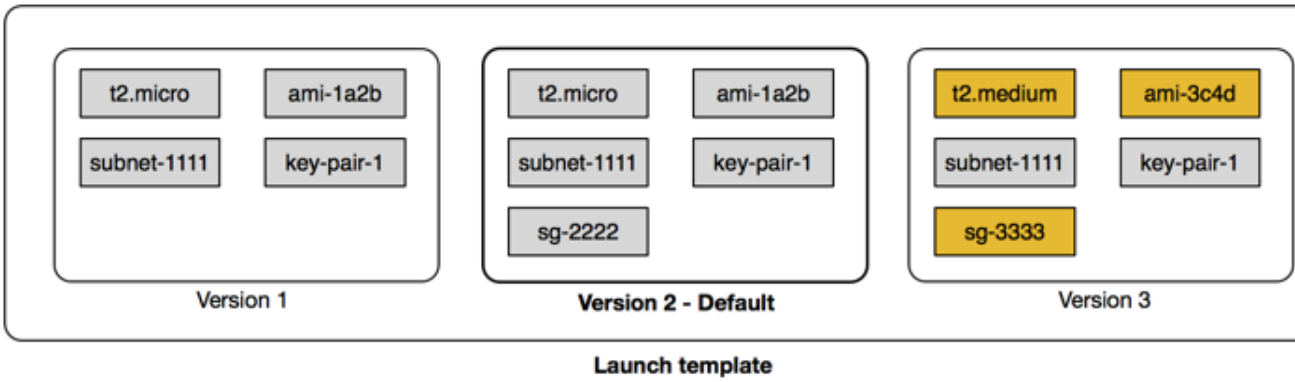
인스턴스가 시작하지 않거나 상태가 `terminated`이 아닌 `running` 로 변경되는 경우, 인스턴스 시작 문제 해결 (p. 1083)단원을 참고하십시오.

시작 템플릿에서 인스턴스 시작

인스턴스 시작에 필요한 구성 정보가 포함된 시작 템플릿을 생성할 수 있습니다. 시작 템플릿은 인스턴스를 시작할 때마다 이를 지정할 필요가 없도록 시작 파라미터를 저장할 수 있게 해줍니다. 예를 들어 시작 템플릿에는 AMI ID, 인스턴스 유형, 인스턴스 시작에 일반적으로 사용되는 네트워크 설정이 포함될 수 있습니다. Amazon EC2 콘솔, AWS SDK 또는 명령줄 도구를 사용하여 인스턴스를 시작할 때 사용할 시작 템플릿을 지정할 수 있습니다.

각 시작 템플릿에서 번호가 지정된 시작 템플릿 버전을 하나 이상 생성할 수 있습니다. 버전마다 시작 파라미터가 다를 수 있습니다. 시작 템플릿에서 인스턴스를 시작할 때 어떤 시작 템플릿 버전이든 사용할 수 있습니다. 버전을 지정하지 않으면 기본 버전이 사용됩니다. 어떤 시작 템플릿 버전이든 기본 버전으로 설정이 가능하며, 기본적으로 시작 템플릿의 최초 버전이 설정되어 있습니다.

다음 그림은 세 가지 버전으로 시작 템플릿을 보여줍니다. 첫 번째 버전은 인스턴스를 시작하는 데 사용할 인스턴스 유형, AMI ID, 서브넷 및 키 페어를 지정합니다. 두 번째 버전은 첫 번째 버전을 토대로 하되, 인스턴스의 보안 그룹도 지정합니다. 세 번째 버전은 일부 파라미터에서 서로 다른 값을 사용합니다. 버전 2가 기본 버전으로 설정되어 있습니다. 이 시작 템플릿에서 인스턴스를 시작한 경우 다른 버전이 지정되어 있지 않으면 버전 2의 시작 파라미터가 사용됩니다.



내용

- 스택 시작 제한 (p. 437)
- 시작 템플릿을 사용하여 시작 파라미터 제어 (p. 437)
- 시작 템플릿 사용 제어 (p. 438)
- 시작 템플릿 생성 (p. 438)
- 시작 템플릿 버전 관리 (p. 443)
- 시작 템플릿에서 인스턴스 시작 (p. 444)
- Amazon EC2 Auto Scaling에서 시작 템플릿 사용 (p. 445)
- EC2 집합에서 시작 템플릿 사용 (p. 445)
- 스팟 집합에서 시작 템플릿 사용 (p. 445)
- 시작 템플릿 삭제 (p. 446)

스택 시작 제한

다음 규칙은 시작 템플릿과 시작 템플릿 버전에 적용됩니다.

- 리전당 5,000개의 시작 템플릿과 시작 템플릿당 10,000개의 버전으로 생성이 제한됩니다.
- 시작 템플릿 파라미터는 선택 사항입니다. 그러나 인스턴스 시작 요청에 필요한 모든 파라미터가 포함되도록 해야 합니다. 예를 들어 시작 템플릿에 AMI ID가 포함되어 있지 않으면 인스턴스를 시작할 때 시작 템플릿과 AMI ID를 모두 지정해야 합니다.
- 시작 템플릿을 생성할 때 시작 템플릿 파라미터가 확인되지 않습니다. 해당 파라미터에 대해 올바른 값을 지정하고 지원되는 파라미터 조합을 사용하는지 확인합니다. 예를 들어 배치 그룹에서 인스턴스를 시작하려면 지원되는 인스턴스 유형을 지정해야 합니다.
- 시작 템플릿에 태그를 지정할 수 있지만, 시작 템플릿 버전에는 태그 지정이 불가능합니다.
- 시작 템플릿 버전은 생성한 순서대로 번호가 지정됩니다. 시작 템플릿 버전을 생성할 때 버전 번호를 자체적으로 지정할 수 없습니다.

시작 템플릿을 사용하여 시작 파라미터 제어

시작 인스턴스에는 인스턴스 시작에 필요한 전체 또는 일부 파라미터가 포함될 수 있습니다. 시작 템플릿을 사용해 인스턴스를 실행할 때 시작 템플릿에 지정된 파라미터를 재정의할 수 있습니다. 또는 시작 템플릿에 없는 추가 파라미터를 지정할 수 있습니다.

> **Note**
>
> 시작 작업 동안 시작 템플릿 파라미터를 제거할 수 없습니다(예를 들어 파라미터에 대해 null 값을 지정할 수 없음). 파라미터를 제거하려면 파라미터 없이 새로운 버전의 시작 템플릿을 생성하고 이 버전을 사용하여 인스턴스를 시작합니다.

인스턴스를 시작하려면 IAM 사용자는 `ec2:RunInstances` 작업을 사용할 권한이 있어야 합니다. 또한 해당 인스턴스로 생성하거나 해당 인스턴스와 연관된 리소스를 생성 또는 사용할 권한이 있어야 합니다. `ec2:RunInstances` 작업에 대한 리소스 수준 권한을 사용하여 사용자가 지정할 수 있는 시작 파라미터를 제어할 수 있습니다. 또는 사용자에게 시작 템플릿을 사용하여 인스턴스를 시작할 권한을 부여할 수 있습니다. 이렇게 하면 IAM 정책이 아닌 시작 템플릿에서 시작 파라미터를 관리하고 시작 템플릿을 사용하여 인스턴스 시작을 위한 권한 부여 방법으로 시작 템플릿을 사용할 수 있습니다. 예를 들어 사용자가 시작 템플릿을 사용하여 인스턴스를 시작만 할 수 있고 특정한 시작 템플릿만 사용하도록 지정할 수 있습니다. 또한 사용자가 시작 템플릿에서 재정의할 수 있는 시작 파라미터를 제어할 수도 있습니다. 예제 정책은 시작 템플릿 (p. 833) 단원을 참조하십시오.

시작 템플릿 사용 제어

기본적으로 IAM 사용자에게는 시작 템플릿 사용 권한이 없습니다. 사용자에게 시작 템플릿과 시작 템플릿 버전을 생성, 수정, 설명 및 삭제할 수 있는 권한을 부여하는 IAM 사용자 정책을 생성할 수 있습니다. 일부 시작 템플릿 작업에 리소스 수준 권한을 적용하여 이러한 작업에서 특정 리소스를 사용할 수 있는 권한을 제어할 수도 있습니다. 자세한 내용은 예제: 시작 템플릿 사용 (p. 840) 정책 예제를 참조하십시오.

`ec2:CreateLaunchTemplate` 및 `ec2:CreateLaunchTemplateVersion` 작업을 사용할 수 있는 권한을 부여할 때는 신중해야 합니다. 리소스 수준 권한을 사용하여 사용자가 시작 템플릿에서 지정할 수 있는 리소스를 제어할 수 없습니다. 인스턴스를 시작하는 데 사용되는 리소스를 제한하려면 시작 템플릿과 시작 템플릿 버전을 생성할 수 있는 권한을 해당 관리자에게만 부여해야 합니다.

시작 템플릿 생성

정의한 파라미터를 사용하여 새로운 시작 템플릿을 생성하거나 기존 시작 템플릿 또는 인스턴스를 새 시작 템플릿 생성을 위한 기준으로 사용합니다.

정의한 파라미터를 사용하여 새 시작 템플릿을 생성하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Launch Templates]와 [Create launch template]를 차례로 선택합니다.
3. Launch template name에 대해 실행 템플릿의 설명이 포함된 이름을 입력하십시오. 작성 시 실행 템플릿에 태그를 지정하려면 Show Tags, Add Tag를 선택한 다음 태그 키 및 값 페어를 입력하십시오.
4. 템플릿 버전 설명에서 시작 템플릿 버전에 대한 간단한 설명을 입력합니다.
5. 시작 템플릿 내용에 다음 정보를 입력합니다.
   - AMI ID: 인스턴스를 시작할 AMI를 지정합니다. 사용 가능한 모든 AMI를 검색하려면 AMI 검색을 선택합니다. 일반적으로 사용되는 AMI를 선택하려면 빠른 시작을 선택합니다. 또는 AWS Marketplace나 커뮤니티 AMI를 선택합니다. 소유하고 있는 AMI를 사용하거나 적절한 AMI를 검색 (p. 96)할 수 있습니다.
   - 인스턴스 유형: 인스턴스 유형이 지정한 AMI와 호환되는지 확인합니다. 자세한 내용은 인스턴스 유형 (p. 178) 단원을 참조하십시오.
   - 키 페어 이름: 인스턴스에 대한 키 페어입니다. 자세한 내용은 Amazon EC2 키 페어 (p. 861) 단원을 참조하십시오.
   - 네트워크 유형: 해당하는 경우 VPC 또는 EC2-Classic으로 인스턴스를 시작할 것인지 여부입니다. VPC를 선택한 경우에는 네트워크 인터페이스 섹션에서 서브넷을 지정합니다. Classic을 선택한 경우에는 EC2-Classic에서 지정된 인스턴스 유형이 지원되는지 확인하고 인스턴스에 대한 가용 영역을 지정합니다.
   - 보안 그룹: 인스턴스에 연결할 하나 이상의 보안 그룹입니다. 자세한 내용은 Amazon EC2 인스턴스에 대한 Linux 보안 그룹 (p. 872) 단원을 참조하십시오.
6. 네트워크 인터페이스에서 인스턴스에 대한 네트워크 인터페이스 (p. 683)를 최대 두 개까지 지정할 수 있습니다.
   - 디바이스: 네트워크 인터페이스의 디바이스 번호입니다(예: 기본 네트워크 인터페이스의 경우 `eth0`). 이 필드를 비워두면 AWS가 기본 네트워크 인터페이스를 생성합니다.

- 네트워크 인터페이스: 네트워크 인터페이스의 ID입니다. AWS가 새로운 네트워크 인터페이스를 생성하도록 하려면 비워둡니다.
- 설명: (선택 사항) 새로운 네트워크 인터페이스의 설명입니다.
- 서브넷: 새로운 네트워크 인터페이스를 생성할 서브넷입니다. 기본 네트워크 인터페이스(`eth0`)에서 이는 인스턴스가 시작되는 서브넷입니다. `eth0`에서 기존 네트워크 인터페이스를 입력한 경우에는 네트워크 인터페이스가 위치하는 서브넷에서 인스턴스가 시작됩니다.
- 자동 할당 퍼블릭 IP: 디바이스 인덱스가 `eth0`인 네트워크 인터페이스에 퍼블릭 IP 주소를 자동으로 할당할 것인지 여부입니다. 이 설정은 새로운 단일 네트워크 인터페이스에서만 활성화될 수 있습니다.
- 기본 IP: 서브넷 범위 중 프라이빗 IPv4 주소입니다. AWS가 자동으로 프라이빗 IPv4 주소를 선택하도록 하려면 비워 둡니다.
- 보조 IP: 서브넷 범위 중 보조 프라이빗 IPv4 주소입니다. AWS가 자동으로 선택하도록 하려면 비워 둡니다.
- (IPv6에만 해당) IPv6 IPs: 서브넷 범위 중 IPv6 주소입니다.
- 보안 그룹 ID: 네트워크 인터페이스를 연결할 VPC의 보안 그룹 ID입니다.
- 종료 시 삭제: 인스턴스가 삭제될 때 네트워크 인터페이스도 삭제되도록 할 것인지 여부입니다.
- Elastic Fabric Adapter(EFA): 네트워크 인터페이스가 Elastic Fabric Adapter(EFA)임을 나타냅니다. 자세한 내용은 Elastic Fabric Adapter(EFA) 단원을 참조하십시오.

7. 스토리지(볼륨)에서 AMI에서 지정된 볼륨 외에 인스턴스에 연결할 볼륨을 지정합니다.

   - 볼륨 유형: 인스턴스에 연결할 인스턴스 스토어나 Amazon EBS 볼륨입니다. 볼륨 유형은 선택한 인스턴스 유형에 따라 달라집니다. 자세한 내용은 Amazon EC2 인스턴스 스토어 (p. 1027) 및 Amazon EBS 볼륨 (p. 890) 단원을 참조하십시오.
   - 디바이스 이름: 볼륨의 디바이스 이름입니다.
   - 스냅샷: 볼륨이 생성되는 스냅샷의 ID입니다.
   - 크기: Amazon EBS볼륨의 경우 스토리지 크기입니다.
   - 볼륨 유형: Amazon EBS 볼륨에 대한 볼륨 유형을 선택합니다. 자세한 내용은 Amazon EBS 볼륨 유형 (p. 892) 단원을 참조하십시오.
   - IOPS: 프로비저닝된 IOPS SSD 볼륨 유형의 경우 해당 볼륨이 지원할 수 있는 초당 입력/출력 작업 수(IOPS)입니다.
   - 종료 시 삭제: Amazon EBS 볼륨의 경우 연결된 인스턴스가 종료되면 볼륨을 삭제할지 여부입니다. 자세한 정보는 인스턴스 종료 시 Amazon EBS 볼륨 보존 (p. 526) 단원을 참조하십시오.
   - 암호화: 인스턴스 유형이 EBS 암호화를 지원하는 경우, 볼륨의 암호화를 활성화할 수 있습니다. 이 리전에서 기본적으로 암호화를 활성화한 경우, 사용자에 대해 암호화가 활성화됩니다. 자세한 내용은 Amazon EBS Encryption (p. 968) 단원을 참조하십시오.
   - 키: EBS 암호화에 사용할 CMK입니다. AWS Key Management Service을 사용하여 생성한 고객 마스터 키(CMK)의 ARN을 지정할 수 있습니다. CMK를 지정하는 경우에도 암호화를 사용하여 암호화를 활성화해야 합니다.

8. 태그에서 키와 값의 조합을 제공하여 태그 (p. 1068)를 지정합니다. 인스턴스 또는 볼륨 또는 이 둘 모두에 태그를 지정할 수 있습니다.

9. 고급 세부 정보에서 필드를 볼 수 있도록 섹션을 확장하고 인스턴스를 위한 추가 파라미터를 지정합니다.

   - 구입 옵션: 구입 모델입니다. 온디맨드 가격으로 제한된 스팟 가격에서 스팟 인스턴스을 요청하려면 스팟 인스턴스 요청을 선택하고 스팟 인스턴스 설정을 변경하려면 스팟 파라미터 사용자 지정을 선택합니다. 스팟 인스턴스를 요청하지 않으면 EC2는 기본적으로 온디맨드 인스턴스를 시작합니다. 자세한 내용은 스팟 인스턴스 (p. 311) 단원을 참조하십시오.
   - IAM 인스턴스 프로파일: 인스턴스에 연결할 AWS Identity and Access Management(IAM) 인스턴스 프로파일입니다. 자세한 내용은 IAM의 Amazon EC2 역할 (p. 850) 단원을 참조하십시오.
   - 종료 방식: 종료 시 인스턴스를 중지 또는 종료해야 할지 여부입니다. 자세한 내용은 인스턴스가 개시하는 종료 동작 변경 (p. 525) 단원을 참조하십시오.

- Stop - Hibernate behavior(중지 - 최대 절전 모드 동작): 인스턴스에 대해 최대 절전 모드가 활성화되어 있는지 여부입니다. 이 필드는 최대 절전 모드 사전 조건을 충족하는 인스턴스에만 유효합니다. 자세한 내용은 Linux 인스턴스를 최대 절전 모드로 실행합니다. (p. 510) 단원을 참조하십시오.
- 종료 방지: 실수로 인한 종료를 방지할 것인지 여부입니다. 자세한 내용은 인스턴스에 대한 종료 방지 기능 활성화 (p. 524) 단원을 참조하십시오.
- 모니터링: Amazon CloudWatch를 사용하여 인스턴스에 대한 세부 모니터링 기능을 활성화할 것인지 여부입니다. 추가 요금이 발생합니다. 자세한 내용은 CloudWatch를 사용해 인스턴스 모니터링하기 (p. 616) 단원을 참조하십시오.
- T2/T3 무제한: 애플리케이션이 필요한 시간만큼 기준 이상으로 버스트를 할 수 있는지 여부입니다. 이 필드는 T2 및 T3 인스턴스에만 유효합니다. 추가 요금이 적용될 수 있습니다. 자세한 내용은 성능 순간 확장 가능 인스턴스 (p. 194) 단원을 참조하십시오.
- 배치 그룹 이름: 인스턴스를 시작할 배치 그룹을 지정합니다. 하나의 배치 그룹에서 모든 인스턴스 유형을 시작할 수 있는 것은 아닙니다. 자세한 내용은 배치 그룹 (p. 757) 단원을 참조하십시오.
- EBS 최적화 인스턴스: Amazon EBS I/O에 대한 추가, 전용 용량을 제공합니다. 모든 인스턴스 유형이 이 기능을 지원하는 것은 아니며, 추가 요금이 적용됩니다. 자세한 내용은 Amazon EBS 최적화 인스턴스 (p. 984) 단원을 참조하십시오.
- 테넌시: 인스턴스를 공유 하드웨어(공유), 격리된 전용 하드웨어(전용) 또는 전용 호스트(전용 호스트)에서 실행할지 선택합니다. 전용 호스트에서 인스턴스를 시작하도록 선택하면 인스턴스를 호스트 리소스 그룹에서 시작할지 여부를 지정하거나 특정 전용 호스트를 대상으로 지정할 수 있습니다. 추가 요금이 적용될 수 있습니다. 자세한 내용은 전용 인스턴스 (p. 410) 및 전용 호스트 (p. 381) 단원을 참조하십시오.
- RAM 디스크 ID: 인스턴스를 위한 RAM 디스크입니다. 커널을 지정해 사용할 때는 해당 커널을 지원하는 드라이버가 설치된 RAM 디스크 지정이 필요할 수 있습니다. 반가상화(PV) AMIs에서만 유효합니다.
- 커널 ID: 인스턴스를 위한 커널입니다. 반가상화(PV) AMIs에서만 유효합니다.
- 사용자 데이터: 시작 과정에서 인스턴스를 구성하거나 구성 스크립트를 실행할 때 사용할 사용자 데이터를 지정할 수 있습니다. 자세한 내용은 다음을 참조하십시오. 시작 시 Linux 인스턴스에서 명령 실행 (p. 565).

10. 시작 템플릿 생성을 선택합니다.

기존 시작 템플릿에서 시작 템플릿을 생성하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 시작 템플릿을 선택합니다.
3. 시작 템플릿 생성을 선택합니다. 시작 템플릿의 이름, 설명, 태그를 제공합니다.
4. 소스 템플릿에서 새로운 시작 템플릿의 토대가 되는 시작 템플릿을 선택합니다.
5. 소스 템플릿 버전에서 새로운 시작 템플릿의 토대가 되는 시작 템플릿 버전을 선택합니다.
6. 필요에 따라 시작 파라미터를 조정하고 시작 템플릿 생성을 선택합니다.

인스턴스에서 시작 템플릿을 생성하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 작업 및 Create template from instance(인스턴스에서 템플릿 만들기)를 선택합니다.
4. 이름, 설명 및 태그를 입력하고 필요에 따라 시작 파라미터를 조정합니다.

   Note

   인스턴스에서 시작 템플릿을 생성할 때 인스턴스의 네트워크 인터페이스 ID 및 IP 주소는 이 템플릿에 포함되지 않습니다.

5. Create Template From Instance(인스턴스에서 템플릿 만들기)를 선택합니다.

시작 템플릿을 생성하려면(AWS CLI)

- create-launch-template(AWS CLI) 명령을 사용합니다. 아래 예제에서는 다음을 지정하는 시작 템플릿을 생성합니다.
  - 시작 템플릿의 이름입니다(`purpose=production`)
  - 시작할 인스턴스 유형(`r4.4xlarge`) 및 AMI(`ami-8c1be5f6`)
  - 총 8개의 vCPU에 대한 코어 수(`4`) 및 코어당 스레드 수(`2`)(코어 4개 x 스레드 2개)
  - 인스턴스를 시작할 서브넷입니다(`subnet-7b16de0c`).

  이 템플릿은 인스턴스에 퍼블릭 IP 주소와 IPv6 주소를 할당하고 인스턴스를 위한 태그를 생성합니다(`Name=webserver`).

```
aws ec2 create-launch-template --launch-template-name TemplateForWebServer
 --version-description WebVersion1 --tag-specifications 'ResourceType=launch-
template,Tags=[{Key=purpose,Value=production}]' --launch-template-data file://template-
data.json
```

  다음은 `template-data.json` 파일의 예입니다.

```
{
    "NetworkInterfaces": [{
        "AssociatePublicIpAddress": true,
        "DeviceIndex": 0,
        "Ipv6AddressCount": 1,
        "SubnetId": "subnet-7b16de0c"
    }],
    "ImageId": "ami-8c1be5f6",
    "InstanceType": "r4.4xlarge",
    "TagSpecifications": [{
        "ResourceType": "instance",
        "Tags": [{
            "Key":"Name",
            "Value":"webserver"
        }]
    }],
    "CpuOptions": {
        "CoreCount":4,
        "ThreadsPerCore":2
    }
}
```

  다음은 예제 출력입니다.

```
{
    "LaunchTemplate": {
        "LatestVersionNumber": 1,
        "LaunchTemplateId": "lt-01238c059e3466abc",
        "LaunchTemplateName": "TemplateForWebServer",
        "DefaultVersionNumber": 1,
        "CreatedBy": "arn:aws:iam::123456789012:root",
        "CreateTime": "2017-11-27T09:13:24.000Z"
    }
}
```

시작 템플릿에 필요한 인스턴스 데이터를 얻으려면(AWS CLI)

- get-launch-template-data(AWS CLI) 명령을 사용하여 인스턴스 ID를 지정합니다. 출력을 새로운 시작 템플릿이나 시작 템플릿 버전을 생성하기 위한 기본 템플릿으로 사용할 수 있습니다. 기본적으로 출력에는 시작 템플릿 데이터에서 지정할 수 없는 최상위 `LaunchTemplateData` 객체가 포함되어 있습니다. 이 객체를 제외하려면 `--query` 옵션을 사용합니다.

```
aws ec2 get-launch-template-data --instance-id i-0123d646e8048babc --query
 "LaunchTemplateData"
```

다음은 예제 출력입니다.

```
    {
        "Monitoring": {},
        "ImageId": "ami-8c1be5f6",
        "BlockDeviceMappings": [
            {
                "DeviceName": "/dev/xvda",
                "Ebs": {
                    "DeleteOnTermination": true
                }
            }
        ],
        "EbsOptimized": false,
        "Placement": {
            "Tenancy": "default",
            "GroupName": "",
            "AvailabilityZone": "us-east-1a"
        },
        "InstanceType": "t2.micro",
        "NetworkInterfaces": [
            {
                "Description": "",
                "NetworkInterfaceId": "eni-35306abc",
                "PrivateIpAddresses": [
                    {
                        "Primary": true,
                        "PrivateIpAddress": "10.0.0.72"
                    }
                ],
                "SubnetId": "subnet-7b16de0c",
                "Groups": [
                    "sg-7c227019"
                ],
                "Ipv6Addresses": [
                    {
                        "Ipv6Address": "2001:db8:1234:1a00::123"
                    }
                ],
                "PrivateIpAddress": "10.0.0.72"
            }
        ]
    }
```

예를 들면 파일에 직접 출력을 기록할 수 있습니다.

```
aws ec2 get-launch-template-data --instance-id i-0123d646e8048babc --query
 "LaunchTemplateData" >> instance-data.json
```

시작 템플릿 버전 관리

특정 시작 템플릿에 대한 시작 템플릿 버전을 생성하고 기본 버전을 설정하며 더 이상 필요하지 않은 버전을 삭제할 수 있습니다.

작업

- 시작 템플릿 버전 생성 (p. 443)
- 기본 시작 템플릿 버전 설정 (p. 443)
- 시작 템플릿 버전 삭제 (p. 444)

시작 템플릿 버전 생성

시작 템플릿 버전을 생성할 때 새로운 시작 파라미터를 지정하거나 기존 버전을 새 버전을 위한 기본 템플릿으로 사용할 수 있습니다. 시작 파라미터에 대한 자세한 내용은 시작 템플릿 생성 (p. 438) 단원을 참조하십시오.

시작 템플릿 버전을 생성하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 시작 템플릿을 선택합니다.
3. 시작 템플릿 생성을 선택합니다.
4. 어떤 작업을 하시겠습니까?에서 새 템플릿 버전 생성을 선택합니다.
5. 시작 템플릿 이름 목록에서 기존 시작 템플릿의 이름을 선택합니다.
6. 템플릿 버전 설명에서 시작 템플릿 버전에 대한 설명을 입력합니다.
7. (선택 사항) 해당하는 시작 템플릿의 버전 또는 새 시작 템플릿 버전의 토대로 사용할 다른 시작 템플릿의 버전을 선택합니다. 새 시작 템플릿 버전은 이 시작 템플릿 버전으로부터 시작 파라미터를 상속합니다.
8. 필요에 따라 시작 파라미터를 수정하고 시작 템플릿 생성을 선택합니다.

시작 템플릿 버전을 생성하려면(AWS CLI)

- create-launch-template-version(AWS CLI) 명령을 사용합니다. 새 버전의 토대가 될 소스 버전을 지정할 수 있습니다. 새 버전은 이 버전에서 시작 파라미터를 상속하며, `--launch-template-data`를 사용하여 파라미터를 재정의할 수 있습니다. 아래 예제에서는 시작 템플릿 버전 1을 토대로 새 버전을 생성하고 다른 AMI ID를 지정합니다.

```
aws ec2 create-launch-template-version --launch-template-id lt-0abcd290751193123 --
version-description WebVersion2 --source-version 1 --launch-template-data "ImageId=ami-
c998b6b2"
```

기본 시작 템플릿 버전 설정

시작 템플릿의 기본 버전을 설정할 수 있습니다. 시작 템플릿에서 인스턴스를 시작하고 버전을 지정하지 않으면 기본 버전의 파라미터를 사용하여 인스턴스가 시작됩니다.

기본 시작 템플릿 버전을 설정하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 시작 템플릿을 선택합니다.
3. 시작 템플릿을 선택하고 작업, 기본 버전 설정을 선택합니다.
4. 기본 버전에서 버전 번호를 선택하고 기본 버전으로 설정하십시오를 선택합니다.

기본 시작 템플릿 버전을 설정하려면(AWS CLI)

- modify-launch-template(AWS CLI) 명령을 사용하여 기본으로 설정하고자 하는 버전을 지정합니다.

```
aws ec2 modify-launch-template --launch-template-id lt-0abcd290751193123 --default-
version 2
```

시작 템플릿 버전 삭제

시작 템플릿 버전이 더 이상 필요하지 않으면 이를 삭제할 수 있습니다. 삭제한 후에는 버전 번호를 바꿀 수 없습니다. 시작 템플릿의 기본 버전은 삭제가 불가능하며, 기본으로 다른 버전을 먼저 할당해야 합니다.

시작 템플릿 버전을 삭제하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 시작 템플릿을 선택합니다.
3. 시작 템플릿을 선택하고 작업, 템플릿 버전 삭제를 선택합니다.
4. 삭제할 버전을 선택하고 시작 템플릿 버전 삭제를 선택합니다.

시작 템플릿 버전을 삭제하려면(AWS CLI)

- delete-launch-template-versions(AWS CLI) 명령을 사용하여 삭제할 버전 번호를 지정합니다.

```
aws ec2 delete-launch-template-versions --launch-template-id lt-0abcd290751193123 --
versions 1
```

시작 템플릿에서 인스턴스 시작

시작 템플릿에 포함된 파라미터를 사용하여 인스턴스를 시작할 수 있습니다. 인스턴스를 시작하기 전에 시작 파라미터를 재정의 또는 추가하는 옵션이 제공됩니다.

시작 템플릿을 사용해 시작되는 인스턴스에는 `aws:ec2launchtemplate:id` 및 `aws:ec2launchtemplate:version` 등 두 개의 키를 통해 두 개의 태그가 자동 할당됩니다. 이러한 태그는 제거 또는 편집이 불가능합니다.

시작 템플릿에서 인스턴스를 시작하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 시작 템플릿을 선택합니다.
3. 시작 템플릿을 선택하고 작업, 템플릿에서 인스턴스 시작을 선택합니다.
4. 사용할 시작 템플릿 버전을 선택합니다.
5. (선택 사항) 인스턴스 세부 정보 섹션에서 파라미터를 변경 및 추가하여 시작 템플릿 파라미터를 재정의하거나 추가할 수 있습니다.
6. 템플릿으로 인스턴스 시작을 선택합니다.

시작 템플릿에서 인스턴스를 시작하려면(AWS CLI)

- run-instances AWS CLI 명령을 사용하여 `--launch-template` 파라미터를 지정합니다. 선택에 따라 사용할 시작 템플릿 버전을 지정합니다. 버전을 지정하지 않으면 기본 버전이 사용됩니다.

```
aws ec2 run-instances --launch-template LaunchTemplateId=lt-0abcd290751193123,Version=1
```

- 시작 템플릿 파라미터를 재정의하려면 run-instances 명령에서 파라미터를 지정합니다. 아래 예제는 시작 템플릿(존재하는 경우)에 지정된 인스턴스 유형을 재정의합니다.

```
aws ec2 run-instances --launch-template LaunchTemplateId=lt-0abcd290751193123 --instance-type t2.small
```

- 복합 구조의 일부인 중첩 파라미터를 지정하면 시작 템플릿에 지정된 복합 구조를 비롯하여 지정된 추가 중첩 파라미터를 사용하여 인스턴스가 시작됩니다.

  아래 예제에서는 `Owner=TeamA` 태그를 비롯해 시작 템플릿에 지정된 기타 태그를 통해 인스턴스가 시작됩니다. 시작 템플릿이 `Owner` 키와 함께 기존 태그를 가지고 있는 경우, 이 값이 `TeamA`로 바뀝니다.

```
aws ec2 run-instances --launch-template LaunchTemplateId=lt-0abcd290751193123 --tag-specifications "ResourceType=instance,Tags=[{Key=Owner,Value=TeamA}]"
```

  아래 예제에서는 디바이스 이름 `/dev/xvdb`를 비롯해 시작 템플릿에 지정된 기타 블록 디바이스 매핑을 통해 볼륨에서 인스턴스가 시작됩니다. 시작 템플릿이 `/dev/xvdb`에 정의된 기존 볼륨을 가지고 있는 경우, 이 값이 지정된 값으로 바뀝니다.

```
aws ec2 run-instances --launch-template LaunchTemplateId=lt-0abcd290751193123 --block-device-mappings "DeviceName=/dev/xvdb,Ebs={VolumeSize=20,VolumeType=gp2}"
```

인스턴스가 시작하지 않거나 상태가 `terminated`이 아닌 `running` 로 변경되는 경우, 인스턴스 시작 문제 해결 (p. 1083)단원을 참고하십시오.

Amazon EC2 Auto Scaling에서 시작 템플릿 사용

Auto Scaling 그룹을 생성하고 그룹에 사용할 시작 템플릿을 지정할 수 있습니다. Amazon EC2 Auto Scaling은 Auto Scaling 그룹에서 인스턴스를 시작할 때 연결된 시작 템플릿에 정의된 시작 파라미터를 사용합니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 시작 템플릿을 사용한 Auto Scaling 그룹 생성을 참조하십시오.

시작 템플릿을 사용하여 Amazon EC2 Auto Scaling 그룹을 생성 또는 업데이트하려면(AWS CLI)

- create-auto-scaling-group 또는 update-auto-scaling-group AWS CLI 명령을 사용하여 `--launch-template` 파라미터를 지정합니다.

EC2 집합에서 시작 템플릿 사용

인스턴스 구성에서 EC2 집합 요청을 생성하고 시작 템플릿을 지정할 수 있습니다. Amazon EC2는 EC2 집합 요청을 이행할 때 연결된 시작 템플릿에 정의된 시작 파라미터를 사용합니다. 시작 템플릿에 지정된 일부 파라미터는 재정의가 가능합니다.

자세한 내용은 EC2 집합 생성 (p. 468) 단원을 참조하십시오.

시작 템플릿을 사용해 EC2 집합을 생성하려면(AWS CLI)

- create-fleet AWS CLI 명령을 사용합니다. `--launch-template-configs` 파라미터를 사용하여 시작 템플릿과 시작 템플릿에 대한 모든 재구성을 지정합니다.

스팟 집합에서 시작 템플릿 사용

인스턴스 구성에서 스팟 집합 요청을 생성하고 시작 템플릿을 지정할 수 있습니다. Amazon EC2는 스팟 집합 요청을 이행할 때 연결된 시작 템플릿에 정의된 시작 파라미터를 사용합니다. 시작 템플릿에 지정된 일부 파라미터는 재정의가 가능합니다.

자세한 내용은 스팟 집합 요청 (p. 334) 단원을 참조하십시오.

시작 템플릿을 사용해 스팟 집합 요청을 생성하려면(AWS CLI)

- request-spot-fleet AWS CLI 명령을 사용합니다. `LaunchTemplateConfigs` 파라미터를 사용하여 시작 템플릿과 시작 템플릿에 대한 모든 재구성을 지정합니다.

시작 템플릿 삭제

시작 템플릿이 더 이상 필요하지 않으면 이를 삭제할 수 있습니다. 시작 템플릿을 삭제하면 모든 버전이 삭제됩니다.

시작 템플릿을 삭제하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 시작 템플릿을 선택합니다.
3. 시작 템플릿을 선택하고 작업, 템플릿 삭제를 선택합니다.
4. 시작 템플릿 삭제를 선택합니다.

시작 템플릿을 삭제하려면(AWS CLI)

- delete-launch-template(AWS CLI) 명령을 사용하여 시작 템플릿을 지정합니다.

```
aws ec2 delete-launch-template --launch-template-id lt-01238c059e3466abc
```

기존 인스턴스의 파라미터를 사용하여 인스턴스 시작

Amazon EC2 콘솔은 기존 인스턴스를 기반으로 시작 마법사 옵션을 제공하여 현재 인스턴스를 기본 템플릿으로 사용하여 다른 인스턴스를 시작할 수 있도록 합니다. 이 옵션을 사용하면 Amazon EC2 시작 마법사에서 선택한 인스턴스의 세부적인 구성 정보가 자동으로 입력됩니다.

> **Note**
>
> 기존 인스턴스를 기반으로 시작 마법사 옵션은 선택한 인스턴스를 복제하는 것이 아니라 일부 구성 정보만 복제합니다. 인스턴스의 사본을 만드려면 해당 인스턴스에서 AMI를 생성한 후 AMI에서 추가 인스턴스를 시작하십시오.
> 또는 인스턴스의 시작 파라미터를 저장하도록 시작 템플릿 (p. 436)을 생성합니다.

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. 659) 단원을 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:

- 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. 인스턴스 페이지에서 사용할 인스턴스를 선택합니다.
2. 작업을 선택한 다음 기존 인스턴스를 기반으로 시작을 선택합니다.
3. 인스턴스 시작 검토 페이지에서 시작 마법사가 열립니다. 인스턴스 세부 정보를 확인한 다음, 해당되는 편집 링크를 클릭해 필요한 사항을 변경할 수 있습니다.

   준비되면 시작을 선택한 다음 키 페어를 선택하고 인스턴스를 시작합니다.
4. 인스턴스가 시작하지 않거나 상태가 `terminated`이 아닌 `running` 로 변경되는 경우, 인스턴스 시작 문제 해결 (p. 1083)단원을 참고하십시오.

백업에서 Linux 인스턴스 시작

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.

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 Store와 스냅샷을 선택합니다.
3. 스냅샷 생성을 클릭합니다.
4. 볼륨 필드에서 루트 볼륨의 이름 또는 ID를 입력한 다음 옵션 목록에서 선택합니다.
5. 방금 만든 스냅샷을 선택한 다음 작업과 이미지 생성을 선택합니다.
6. EBS 스냅샷에서 이미지 생성 대화 상자에서 다음 정보를 입력한 후 생성을 선택합니다. 상위 인스턴스를 다시 생성하는 경우 상위 인스턴스와 동일한 옵션을 선택합니다.

   - 아키텍처: 32비트의 경우 i386을 선택하고 64비트의 경우 x86_64를 선택합니다.
   - 루트 디바이스 이름: 루트 볼륨에 적절한 이름을 입력합니다. 자세한 내용은 Linux 인스턴스의 디바이스 명명 (p. 1047) 단원을 참조하십시오.
   - 가상화 유형: 이 AMI에서 실행된 인스턴스가 반가상화(PV)를 사용하는지 또는 하드웨어 가상 머신(HVM) 가상화를 사용하는지 선택합니다. 자세한 내용은 Linux AMI 가상화 유형 (p. 95) 단원을 참조하십시오.
   - (PV 가상화 유형에만 해당) 커널 ID 및 RAM 디스크 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) 블록 디바이스 매핑: 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. 965) 단원을 참조하십시오.

7. 탐색 창에서 AMI를 선택합니다.
8. 방금 생성한 AMI를 선택하고 시작을 선택합니다. 마법사가 안내하는 대로 인스턴스를 시작합니다. 마법사 단계별 구성에 대한 자세한 내용은 인스턴스 시작 마법사를 사용하여 인스턴스 시작 (p. 431)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. 108) 단원을 참조하십시오. 시작한 이후에 구독을 취소하려면 먼저 해당 구독에서 실행 중인 모든 인스턴스를 종료해야 합니다. 자세한 내용은 AWS Marketplace 구독 관리 (p. 111) 단원을 참조하십시오.

시작 마법사를 사용하여 AWS Marketplace에서 인스턴스를 시작하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. Amazon EC2 대시보드에서 인스턴스 시작을 선택합니다.
3. Amazon 머신 이미지(AMI) 선택 페이지에서 왼쪽에 있는 AWS Marketplace 범주를 선택합니다. 범주를 검색하거나 검색 기능을 사용하여 적합한 AMI를 찾습니다. 선택을 선택하여 제품을 선택합니다.
4. 대화 상자에 선택한 제품에 대한 개요가 표시됩니다. 요금 정보와 공급업체에서 제공한 기타 정보를 조회할 수 있습니다. 준비가 되면 계속을 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. 인스턴스 유형 선택 페이지에서 시작할 인스턴스의 하드웨어 구성 및 크기를 선택합니다. 완료되면 Next: Configure Instance Details(다음: 인스턴스 세부 정보 구성)을 선택합니다.
6. 마법사의 다음 페이지에서 인스턴스를 구성하고, 스토리지 및 태그를 추가할 수 있습니다. 구성 가능한 다른 옵션에 대한 자세한 내용은 인스턴스 시작 마법사를 사용하여 인스턴스 시작 (p. 431) 단원을 참조하십시오. 보안 그룹 구성 페이지가 나타날 때까지 다음을 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(검토 후 시작)를 선택합니다.
7. Review Instance Launch(인스턴스 시작 검토) 페이지에서 인스턴스를 시작할 AMI에 대한 세부 정보와 마법사에서 설정한 기타 구성 정보를 확인합니다. 준비되면 시작을 선택하여 키 페어를 선택하거나 생성하고 인스턴스를 시작합니다.
8. 구독한 제품에 따라 인스턴스를 시작하는 데 몇 분 또는 그 이상 걸릴 수 있습니다. 인스턴스를 시작하려면 먼저 제품을 구독해야 합니다. 신용 카드 정보에 문제가 있는 경우 계정 세부 정보를 업데이트하라는 메시지가 나타납니다. 시작 확인 페이지가 표시되면 인스턴스 보기를 선택하여 인스턴스 페이지로 이동합니다.

   Note

   유휴 상태를 포함해 인스턴스가 실행 중인 동안 구독 요금이 청구됩니다. 인스턴스를 중지하더라도 스토리지에 대해 요금이 부과될 수 있습니다.

9. 인스턴스가 실행 중 상태일 때 인스턴스에 연결할 수 있습니다. 이렇게 하려면 목록에서 인스턴스를 선택하고 연결을 선택합니다. 대화 상자의 지침을 따릅니다. 인스턴스 연결에 대한 자세한 내용은 Linux 인스턴스에 연결 (p. 485) 단원을 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. 111) 단원을 참조하십시오.

10. 인스턴스가 시작하지 않거나 상태가 `terminated`이 아닌 `running` 로 변경되는 경우, 인스턴스 시작 문제 해결 (p. 1083)단원을 참고하십시오.

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 정보는 인스턴스 시작 단원을 참조하십시오.
Windows PowerShell용 AWS 도구	New-EC2Instance 명령을 사용합니다. 자세한 내용은 Windows PowerShell을 사용하여 Amazon EC2 시작을 참조하십시오.
Query API	RunInstances 요청을 사용합니다.

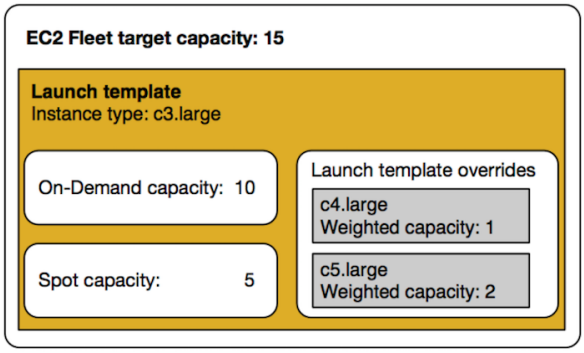
EC2 플릿 시작

EC2 집합에는 인스턴스의 플릿 또는 그룹을 시작하기 위한 구성 정보가 있습니다. 한 번의 API 호출에서 플릿은 온디맨드 인스턴스, 예약 인스턴스 및 스팟 인스턴스 구매 옵션을 함께 사용하여 여러 가용 영역에서 여러 인스턴스 유형을 시작할 수 있습니다. EC2 집합을 사용하여 다음 작업을 수행할 수 있습니다.

- 별도의 온디맨드 및 스팟 용량 목표 및 시간당 지불하려는 최대 금액 정의
- 애플리케이션에 가장 적합하게 작동하는 인스턴스 유형 지정
- 각 구매 옵션 내에서 Amazon EC2가 플릿 용량을 배포해야 하는 방법 지정

또한 플릿에 대해 지불할 시간당 최대 금액을 설정할 수 있으며, EC2 집합은 최대 금액에 도달할 때까지 인스턴스를 실행합니다. 지불하려는 최대 금액에 도달하면 플릿은 목표 용량을 충족하지 않은 경우에도 인스턴스 실행을 중지합니다.

EC2 집합은 요청에 지정된 목표 용량을 충족하는 데 필요한 수만큼 인스턴스를 시작하려고 시도합니다. 시간당 총 최대 요금을 지정하면 지불하려는 최대 금액에 도달할 때까지 용량을 충족합니다. 스팟 인스턴스가 중단될 경우 플릿은 대상 스팟 용량을 유지하려고 시도할 수 있습니다. 자세한 정보는 스팟 인스턴스의 작동 방식 (p. 314) 단원을 참조하십시오.



EC2 집합마다 인스턴스 유형 수를 무제한 지정할 수 있습니다. 온디맨드 및 스팟 구매 옵션을 둘 다 사용하여 이 인스턴스 유형을 프로비저닝할 수 있습니다. 여러 가용 영역을 지정하고, 인스턴스마다 서로 다른 최대 스팟 가격을 지정하며, 플릿마다 추가 스팟 옵션을 선택할 수도 있습니다. Amazon EC2에서는 지정된 옵션을 사용하여 플릿이 시작될 때 용량을 프로비저닝합니다.

플릿이 실행되는 동안 Amazon EC2에서 가격 증가나 인스턴스 오류로 인해 스팟 인스턴스를 회수할 경우 EC2 집합이 해당 인스턴스를 사용자가 지정하는 인스턴스 유형으로 대체하려고 시도할 수 있습니다. 따라서 스팟 가격이 급증하는 동안 용량을 더 쉽게 다시 획득할 수 있습니다. 플릿마다 유연하고 탄력적인 리소싱 전략을 개발할 수 있습니다. 예를 들어 특정한 플릿에서 기본 용량은 저렴한 스팟 용량(사용 가능한 경우)으로 보충된 온디맨드일 수 있습니다.

예약 인스턴스가 있고 플릿에 온디맨드 인스턴스를 지정하면 EC2 집합에서는 예약 인스턴스를 사용합니다. 예를 들어 플릿에서 온디맨드 인스턴스를 `c4.large`로 지정하고 예약 인스턴스의 `c4.large`가 있으면 예약 인스턴스 요금을 수신합니다.

EC2 집합을 사용해도 추가 요금이 부과되지 않으며 플릿이 시작되는 EC2 인스턴스에 대해서만 비용을 지불합니다.

목차

- EC2 집합 제한 사항 (p. 450)
- EC2 집합 제한 (p. 450)
- EC2 집합 구성 전략 (p. 451)
- EC2 집합 관리 (p. 460)

EC2 집합 제한 사항

EC2 집합에는 다음과 같은 제한이 적용됩니다.

- API 또는 AWS CLI를 통해서만 EC2 집합을 사용할 수 있습니다.
- EC2 집합 요청으로 AWS 리전을 확장할 수 없습니다. 리전마다 따로 EC2 집합을 생성해야 합니다.
- EC2 집합 요청으로 동일한 가용 영역의 서로 다른 서브넷을 확장할 수 없습니다.

EC2 집합 제한

스팟 요청 가격 제한, 인스턴스 제한 및 볼륨 제한과 같이 EC2 집합에서 시작된 인스턴스에 일반적인 Amazon EC2 제한이 적용됩니다. 또한 다음과 같은 제한이 적용됩니다.

- AWS 리전당 활성 EC2 집합 수: 1,000 *†
- 플릿당 시작 사양 수: 50개†
- 시작 사양의 사용자 데이터 크기: 16KB†

- EC2 집합당 대상 용량: 10,000
- 리전 내 모든 EC2 집합의 대상 용량: 100,000*

대상 용량의 기본 한도 이상이 필요한 경우, AWS 지원 센터 사례 생성 서식을 작성하여 한도 증가를 요청하십시오. 제한 유형에서 EC2 Fleet(EC2 플릿)을 선택하고 리전을 선택한 다음 Target Fleet Capacity per Fleet (in units)(플릿당 대상 플릿 용량(유닛)) 또는 Target Fleet Capacity per Region (in units)(리전당 대상 플릿 용량(유닛))을 선택하거나 둘 다 선택합니다.

*이러한 제한은 사용자의 EC2 집합 및 스팟 집합 모두에 적용됩니다.

† 이 숫자는 하드 제한입니다. 이러한 제한에 대한 한도 상승을 요청할 수 없습니다.

T3 스팟 인스턴스

T3 스팟 인스턴스를 CPU 크레딧 발생에 대한 유휴 시간 없이 즉시 짧은 기간 동안 사용할 계획인 경우 T3 스팟 인스턴스를 `standard` (p. 206) 모드로 시작하여 높은 비용 지불을 방지하는 것이 좋습니다.

T3 스팟 인스턴스를 `unlimited` (p. 197) 모드를 시작하고 CPU를 즉시 버스트하는 경우 버스팅에 대한 잉여 크레딧을 소모하게 됩니다. 인스턴스를 짧은 기간 동안 사용하는 경우 인스턴스에서 잉여 크레딧을 지불할 정도의 CPU 크레딧이 발생할 시간이 없습니다. 인스턴스를 종료할 때 잉여 크레딧에 대한 요금이 청구됩니다.

T3 스팟 인스턴스가 `Unlimited` 모드인 경우는 버스팅에 대한 CPU 크레딧이 발생할 정도로 인스턴스 실행이 긴 경우에만 적합합니다. 그렇지 않은 경우 잉여 크레딧에 대한 비용을 지불하면 T3 스팟 인스턴스가 M5 또는 C5 인스턴스에 비해 비싸집니다. 자세한 내용은 무제한 모드 대 고정 CPU 사용 시기 (p. 199) 단원을 참조하십시오.

T2 스팟 인스턴스

시작 크레딧은 효율적인 컴퓨팅 리소스를 제공하여 인스턴스를 구성함으로써 T2 인스턴스에 대한 생산적인 최초 시작 환경을 제공하는 것을 목적으로 합니다. 새 시작 크레딧에 액세스하기 위한 T2 인스턴스의 반복된 시작은 허용되지 않습니다. 지속적인 CPU가 필요한 경우 (일정 기간 동안 유휴 상태로 둠으로써) 크레딧을 얻고 T2 무제한 (p. 197)을 사용하거나 전용 CPU(예: `c4.large`)를 포함한 인스턴스 유형을 사용할 수 있습니다.

EC2 집합 구성 전략

EC2 집합은 온디맨드 인스턴스 및 스팟 인스턴스로 구성된 그룹입니다.

EC2 집합은 집합 요청에서 지정한 목표 용량을 충족하는 데 필요한 인스턴스 수만큼 시작하려고 시도합니다. 집합은 온디맨드 인스턴스 또는 스팟 인스턴스만으로 구성되거나 온디맨드 인스턴스 및 스팟 인스턴스의 조합으로 구성될 수 있습니다. 스팟 인스턴스에 대한 요청은 요청의 시간당 최대 가격이 현재 스팟 가격을 초과하고 사용 가능한 용량이 있으면 수행됩니다. 스팟 인스턴스가 중단될 경우 플릿은 대상 용량을 유지하려고 시도합니다.

또한 플릿에 대해 지불할 시간당 최대 금액을 설정할 수 있으며, EC2 집합은 최대 금액에 도달할 때까지 인스턴스를 실행합니다. 지불하려는 최대 금액에 도달하면 플릿은 목표 용량을 충족하지 않은 경우에도 인스턴스 실행을 중지합니다.

스팟 인스턴스 풀은 동일한 인스턴스 유형, 운영 체제, 가용 영역 및 네트워크 플랫폼을 가지는 미사용 EC2 인스턴스의 세트입니다. EC2 집합을 생성할때 인스턴스 유형, 가용 영역, 서브넷 및 최고 가격에 따라 달라지는 여러 시작 사양을 포함할 수 있습니다. 플릿은 요청에 포함된 시작 사양과 요청의 구성을 기반으로 요청을 이행하는 데 사용되는 스팟 인스턴스 풀을 선택합니다. 스팟 인스턴스는 선택한 풀에서 가져옵니다.

EC2 집합을 사용하면 코어나 인스턴스 수 또는 메모리 양을 기반으로 애플리케이션에 맞는 대량의 EC2 용량을 프로비저닝할 수 있습니다. 예를 들어 EC2 집합이 인스턴스 200개의 목표 용량을 시작하도록 지정할 수 있습니다. 이 가운데 130개는 온디맨드 인스턴스이고 나머지는 스팟 인스턴스입니다. 또는 코어당 RAM

이 최소 2GB인 코어 1,000개를 요청할 수 있습니다. 플릿은 Amazon EC2 옵션 조합을 결정하여 절대 최저 비용으로 해당 용량을 시작합니다.

적절한 구성 전략을 사용하여 요구에 맞는 EC2 집합을 생성하십시오.

내용

- EC2 집합 계획 (p. 452)
- EC2 집합 요청 유형 (p. 452)
- 스팟 인스턴스를 위한 할당 전략 (p. 453)
- 온디맨드 백업을 위한 EC2 집합 구성 (p. 454)
- 최고 가격 재정의 (p. 455)
- 지출 제어 (p. 455)
- EC2 집합 인스턴스 가중치 부여 (p. 456)
- 자습서: 인스턴스 가중치를 부여한 EC2 집합 사용 (p. 457)
- 자습서: 기본 용량이 온디맨드인 EC2 집합 사용 (p. 459)

EC2 집합 계획

EC2 집합을 계획할 때 다음을 수행하는 것이 좋습니다.

- 원하는 목표 용량의 동기 또는 비동기식 일회성 요청을 제출하는 EC2 집합을 생성할지 아니면 시간 경과에 따라 목표 용량을 유지하는 플릿을 생성할지 결정합니다. 자세한 내용은 EC2 집합 요청 유형 (p. 452) 단원을 참조하십시오.
- 인스턴스 유형을 결정하고 애플리케이션 요건을 만족합니다.
- 스팟 인스턴스를 EC2 집합에 포함하려면 플릿을 생성하기 전에 스팟 모범 사례를 살펴보십시오. 가능한 한 최저 가격으로 인스턴스를 프로비저닝할 수 있도록 플릿을 계획할 때 이 모범 사례를 사용하십시오.
- EC2 집합의 목표 용량을 결정합니다. 인스턴스 또는 사용자 지정 단위에서 목표 용량을 설정할 수 있습니다. 자세한 내용은 EC2 집합 인스턴스 가중치 부여 (p. 456) 단원을 참조하십시오.
- EC2 집합 목표 용량 중에서 온디맨드 용량 및 스팟 용량이어야 하는 부분을 결정합니다. 온디맨드 용량이나 스팟 용량 또는 둘 다 0을 지정할 수 있습니다.
- 인스턴스 가중치를 사용하는 경우에는 단위당 가격을 결정합니다. 단위당 가격을 계산하려면 인스턴스 시간당 가격을 이 인스턴스가 나타내는 단위 수(또는 가중치)로 나눕니다 인스턴스 가중치를 사용하지 않는 경우 단위당 기본 가격은 인스턴스 시간당 가격입니다.
- 플릿에 대해 지불할 최대 시간당 금액을 결정하십시오. 자세한 내용은 지출 제어 (p. 455) 단원을 참조하십시오.
- EC2 집합의 가능한 옵션을 살펴봅니다. 자세한 내용은 EC2 집합 JSON 구성 파일 참조 (p. 465) 단원을 참조하십시오. EC2 집합 구성 예제는 EC2 집합 구성의 예 (p. 474) 단원을 참조하십시오.

EC2 집합 요청 유형

EC2 집합에는 다음 세 가지 유형의 요청이 있습니다.

`instant`

요청 유형을 `instant`로 구성하면 EC2 집합이 원하는 용량을 얻기 위한 동기식 일회성 요청을 합니다. API 응답에서 시작할 수 없는 인스턴스에 대한 오류와 함께 시작된 인스턴스를 반환합니다.

`request`

요청 유형을 `request`로 구성하면 EC2 집합이 원하는 용량을 얻기 위한 비동기식 일회성 요청을 합니다. 그 뒤에 스팟 중단으로 인해 용량이 감소할 경우 플릿은 스팟 인스턴스를 보충하려고 하지 않으며 용량을 사용할 수 없는 경우 대체 스팟 인스턴스에서 요청을 제출하지 않습니다.

`maintain`

(기본값) 요청 유형을 `maintain`으로 구성하면 EC2 집합은 원하는 용량을 얻기 위한 비동기식 요청을 하고 중단된 모든 스팟 인스턴스를 자동으로 보충해 용량을 유지합니다.

세 가지 유형의 요청 모두에 할당 전략이 유익합니다. 자세한 내용은 스팟 인스턴스를 위한 할당 전략 (p. 453) 단원을 참조하십시오.

스팟 인스턴스를 위한 할당 전략

EC2 집합의 할당 전략에 따라, 시작 사양으로 표시되는 가능한 스팟 인스턴스 풀에서 스팟 인스턴스에 대한 요청을 이행하는 방법이 결정됩니다. 플릿에 지정할 수 있는 할당 전략은 다음과 같습니다.

`lowest-price`

스팟 인스턴스는 최저 가격의 풀에서 가져옵니다. 이는 기본 전략입니다.

`diversified`

스팟 인스턴스는 모든 풀에 두루 분산됩니다.

`capacity-optimized`

스팟 인스턴스는 시작하는 인스턴스의 수에 대한 용량이 최적화된 풀에서 가져옵니다.

`InstancePoolsToUseCount`

스팟 인스턴스는 지정한 스팟 풀 수에 걸쳐 배포됩니다. 이 파라미터는 `lowest-price`와 함께 사용하는 경우에만 유효합니다.

목표 용량 유지

스팟 가격 또는 스팟 인스턴스 풀의 가용 용량 변화로 인해 스팟 인스턴스가 종료된 후에는 `maintain` 유형의 EC2 집합가 대체 스팟 인스턴스를 시작합니다. 할당 전략이 `lowest-price`인 경우, 플릿은 현재 스팟 가격이 가장 낮은 풀에서 대체 인스턴스를 시작합니다. 할당 전략으로써 `InstancePoolsToUseCount`와 함께 `lowest-price`를 사용하는 경우 플릿이 최저 가격의 스팟 풀을 선택하여 지정한 스팟 풀 수에 걸쳐 스팟 인스턴스를 시작합니다. 할당 전략이 `capacity-optimized`인 경우, 플릿은 가용성이 뛰어난 스팟 인스턴스 용량이 있는 풀에서 대체 인스턴스를 시작합니다. 할당 전략이 `diversified`인 경우 플릿은 나머지 풀에 대체 스팟 인스턴스를 배포합니다.

비용 최적화를 위한 EC2 집합 구성

스팟 인스턴스 사용 비용을 최적화하려면 EC2 집합이 현재 스팟 가격을 기반으로 인스턴스 유형과 가용 영역의 가장 저렴한 조합을 자동으로 배포하도록 `lowest-price` 할당 전략을 지정합니다.

EC2 집합은 항상 퍼블릭 온디맨드 가격을 기반으로 가장 저렴한 인스턴스 유형을 온디맨드 인스턴스 목표 용량에 대해 선택하며, 스팟 인스턴스에 대해서는 계속해서 할당 전략(`lowest-price`, `capacity-optimized` 또는 `diversified`)을 따릅니다.

비용 최적화 및 다각화를 위한 EC2 집합 구성

저렴함과 다각화를 모두 충족하는 스팟 인스턴스 플릿을 생성하려면 `InstancePoolsToUseCount`와 함께 `lowest-price` 할당 전략을 사용하십시오. EC2 집합이 지정된 스팟 풀 수에 걸쳐 현재 스팟 가격을 기반으로 인스턴스 유형과 가용 영역의 가장 저렴한 조합을 자동으로 배포합니다. 이 조합을 통해 가장 비싼 스팟 인스턴스를 피할 수 있습니다.

용량 최적화를 위한 EC2 집합 구성

스팟 인스턴스에서 요금은 시간이 지나면서 수요 및 공급의 장기 추세에 따라 서서히 변화하지만 용량은 실시간으로 변동합니다. `capacity-optimized` 전략은 실시간 용량 데이터를 기준으로 가장 가용성이 높은 풀을 예측하여 자동으로 스팟 인스턴스를 가장 가용성이 높은 풀로 시작합니다. 이 전략은 빅 데이터 및 분

석, 이미지 및 미디어 렌더링, 기계 학습, 고성능 컴퓨팅과 같이 작업 재시작 및 체크포인트와 관련된 중단으로 인한 비용이 더 높을 수 있는 워크로드에 유용합니다. `capacity-optimized` 전략은 중단을 줄일 수 있는 가능성을 제공함으로써 전체 워크로드 비용을 낮출 수 있습니다.

적합한 할당 전략 선택

사용 사례를 바탕으로 플릿을 최적화할 수 있습니다.

플릿이 작거나 짧은 시간 동안 실행될 경우 모든 인스턴스가 단일 스팟 인스턴스 풀에 있더라도 스팟 인스턴스가 중단될 확률은 낮습니다. 따라서 `lowest-price` 전략이 요구를 충족시키는 동시에 최저 가격을 제공할 가능성이 높습니다.

플릿이 크거나 장시간 실행될 경우 스팟 인스턴스를 여러 풀로 분산하여 플릿의 가용성을 높일 수 있습니다. 예를 들어 EC2 집합이 풀 10개와 인스턴스 100개의 목표 용량을 지정하면 플릿이 각 풀에서 스팟 인스턴스 10개를 시작합니다. 풀에서 스팟 가격이 최고 가격을 초과하는 경우, 플릿 중 10%만 영향을 받습니다. 이 전략을 사용하면 플릿이 시간이 지나면서 어느 한 풀에서 발생하는 스팟 가격의 상승에 덜 민감해집니다.

`diversified` 전략 사용 시 EC2 집합은 온디맨드 가격보다 높거나 이 가격과 동일한 스팟 가격의 풀로 스팟 인스턴스를 시작하지 않습니다.

저렴하고 다각화된 플릿을 생성하려면 `lowest-price`과 함께 `InstancePoolsToUseCount` 전략을 사용하십시오. 스팟 인스턴스를 할당할 스팟 풀 수로써 낮은 수 또는 높은 수를 사용할 수 있습니다. 예를 들어 일괄 처리를 실행하는 경우 대기열이 비용 절감 효과를 극대화하는 동시에 컴퓨팅 파워를 항상 확보할 수 있도록 스팟 풀 수를 낮게(예: `InstancePoolsToUseCount=2`) 지정하는 것이 좋습니다. 웹 서비스를 실행하는 경우 스팟 인스턴스 풀을 일시적으로 사용할 수 없게 되었을 때 그 충격을 최소화할 수 있도록 스팟 풀 수를 낮게(예: `InstancePoolsToUseCount=10`) 지정하는 것이 좋습니다.

플릿이 작업 재시작 및 검사와 연관된 중단 비용이 높을 수 있는 워크로드를 실행하는 경우 `capacity-optimized` 전략을 사용하십시오. 이 전략은 중단을 줄일 수 있는 가능성을 제공함으로써 전체 워크로드 비용을 낮출 수 있습니다.

온디맨드 백업을 위한 EC2 집합 구성

주요 뉴스 이벤트나 게임 출시 중에 조정이 필요한 뉴스 웹 사이트와 같이 예측할 수 없는 조정 요구가 있을 경우 선호하는 옵션에 충분한 가용 용량이 없으면 온디맨드 인스턴스의 대체 인스턴스 유형을 지정하는 것이 좋습니다. 예를 들어 `c5.2xlarge` 온디맨드 인스턴스를 선호하지만 가용 용량이 부족하다면 피크 로드 중에 `c4.2xlarge` 인스턴스를 몇 개 사용하고 싶은 경우가 있을 수 있습니다. 이 경우 EC2 집합은 `c5.2xlarge` 인스턴스를 사용하여 목표 용량을 모두 충족하려고 하지만 용량이 부족하면 자동으로 `c4.2xlarge` 인스턴스를 시작하여 목표 용량을 충족합니다.

온디맨드 용량에 대한 인스턴스 유형 우선순위 지정

EC2 집합가 온디맨드 용량을 채우려고 시도하는 경우 기본적으로 최저 가격의 인스턴스 유형을 먼저 시작합니다. `AllocationStrategy`가 `prioritized`로 설정된 경우 EC2 집합가, 우선 순위를 통해, 온디맨드 용량을 채우기 위해 먼저 사용할 인스턴스 유형을 결정합니다. 시작 템플릿 재정의에 우선 순위를 할당하고 우선 순위가 가장 높은 것을 먼저 시작합니다.

예를 들어 서로 다른 인스턴스 유형인 `c3.large`, `c4.large`, `c5.large`를 각각 지닌 3개의 시작 템플릿 재정의를 구성했다고 가정해 보겠습니다. `c5.large`에 대한 온디맨드 가격은 `c4.large`에 대한 온디맨드 가격보다 낮습니다. `c3.large`가 가장 낮습니다. 우선 순위를 사용해 순서를 결정하지 않는 경우 플릿이 `c3.large`로 시작하여 온디맨드 용량을 채운 후 `c5.large`를 사용합니다. 종종 `c4.large`에 대한 미사용 예약 인스턴스가 있게 되므로 `c4.large`, `c3.large`, `c5.large`의 순서이도록 시작 템플릿 재정의 우선 순위를 설정할 수 있습니다.

온디맨드 인스턴스에 대해 용량 예약 사용

용량 예약을 위한 사용 전략을 `use-capacity-reservations-first`로 설정하여 온디맨드 인스턴스를 시작할 때 온디맨드 용량 예약부터 사용하도록 플릿을 구성할 수 있습니다. 이 설정을 온디맨드 인스턴스에 대한 할당 전략(`lowest-price` 또는 `prioritized`)과 함께 사용할 수 있습니다.

미사용 용량 예약을 사용하여 온디맨드 용량을 이행할 경우:

- 플릿은 미사용 용량 예약을 사용하여 온디맨드 용량을 목표 온디맨드 용량까지 이행합니다.
- 여러 인스턴스 풀에 미사용 용량 예약이 있는 경우, 온디맨드 할당 전략(`lowest-price` 또는 `prioritized`)이 적용됩니다.
- 미사용 용량 예약 수가 온디맨드 목표 용량보다 적을 경우, 온디맨드 할당 전략(`lowest-price` 또는 `prioritized`)에 따라 나머지 온디맨드 목표 용량이 시작됩니다.

미사용 온디맨드 용량 예약는 `instant` 유형의 플릿에만 사용할 수 있습니다.

온디맨드 용량을 이행하기 위해 용량 예약를 사용하도록 플릿을 구성하는 방법의 예는 EC2 집합 구성의 예 (p. 474) 단원을 참조하십시오. 자세한 내용은 온디맨드 용량 예약 (p. 415) 및 온디맨드 용량 예약 FAQ를 참조하십시오.

최고 가격 재정의

각 EC2 집합은 글로벌 최고 가격을 포함하거나 기본 가격(온디맨드 가격)을 사용할 수 있습니다 플릿은 각 시작 사양의 기본 최고 가격으로 이 가격을 사용합니다.

하나 이상의 시작 사양에서 최고 가격을 선택적으로 지정할 수 있습니다. 이 가격은 시작 사양에 특정한 것입니다. 시작 사양에 특정 가격이 포함되는 경우 EC2 집합은 글로벌 최고 가격 대신 이 최고 가격을 사용합니다. 특정 최고 가격을 포함하지 않는 다른 시작 사양은 글로벌 최고 가격을 계속해서 사용합니다.

지출 제어

EC2 집합은 `TotalTargetCapacity` 또는 `MaxTotalPrice`(지불할 최대 금액) 파라미터 중 하나를 충족하면 인스턴스 실행을 중지합니다. 플릿에 대해 시간당 지불하는 금액을 관리하려면 `MaxTotalPrice`를 지정할 수 있습니다. 최대 총 가격에 도달하면 EC2 집합은 목표 용량을 충족하지 않은 경우에도 인스턴스 실행을 중지합니다.

다음 예제와 같이 이 작업을 두 가지 시나리오로 수행할 수 있습니다. 첫 번째 시나리오에서 EC2 집합은 대상 용량을 충족했을 때 인스턴스 실행을 중지합니다. 두 번째 시나리오에서 EC2 집합은 지불할 최대 금액에 도달하면 인스턴스 실행을 중지합니다(`MaxTotalPrice`).

예: 대상 용량에 도달할 때 인스턴스 실행 중지

다음과 같은 `m4.large` 온디맨드 인스턴스 요청 시:

- 온디맨드 가격: 시간당 0.10 USD
- `OnDemandTargetCapacity`: 10
- `MaxTotalPrice`: 1.50 USD

EC2 집합은 온디맨드 인스턴스의 경우 최대 1.00 USD(10개 인스턴스 x 0.10 USD)가 `MaxTotalPrice` 1.50 USD를 초과하지 않기 때문에 10개의 온디맨드 인스턴스를 시작합니다.

예: 최대 총 가격에 도달할 때 인스턴스 실행 중지

다음과 같은 `m4.large` 온디맨드 인스턴스 요청 시:

- 온디맨드 가격: 시간당 0.10 USD
- `OnDemandTargetCapacity`: 10
- `MaxTotalPrice`: 0.80 USD

EC2 집합이 온디맨드 대상 용량(온디맨드 인스턴스 10개)을 시작하면 시간당 총 비용은 1.00 USD입니다. 온디맨드 인스턴스의 `MaxTotalPrice`에 대해 지정된 금액(0.80 USD) 보다 높습니다. 지불할 금액보다 더

많은 지출을 방지하기 위해 EC2 집합은 8개의 온디맨드 인스턴스(온디맨드 대상 용량 미만) 만 실행합니다. 더 많이 실행하면 온디맨드 인스턴스의 `MaxTotalPrice`를 초과할 수 있기 때문입니다.

EC2 집합 인스턴스 가중치 부여

EC2 집합을 생성할 때 각 인스턴스 유형이 애플리케이션의 성능에 기여할 수 있는 용량 단위를 정의할 수 있습니다. 인스턴스 가중치를 사용하여 각 시작 사양의 최대 가격을 조정할 수 있습니다.

기본적으로, 사용자가 지정하는 가격은 인스턴스 시간당 가격입니다. 인스턴스 가중치 기능을 사용할 때, 사용자가 지정하는 가격은 단위 시간당 가격입니다. 단위 시간당 가격은 인스턴스 유형에 따른 가격을 인스턴스가 나타내는 단위 수로 나누어 계산합니다. EC2 집합은 목표 용량을 인스턴스 가중치로 나누어 시작할 인스턴스 수를 계산합니다. 결과가 정수가 아닌 경우 플릿은 결과를 다음 정수로 반올림하므로 플릿 크기가 목표 용량을 밑돌지는 않습니다. 시작된 인스턴스의 용량이 요청된 목표 용량을 초과하더라도 플릿은 시작 사양에 지정한 어떤 풀이든 선택할 수 있습니다.

다음 표에는 목표 용량이 10인 EC2 집합의 단위당 가격을 결정하기 위한 계산의 예가 있습니다.

인스턴스 유형	인스턴스 가중치	목표 용량	시작된 인스턴스의 수	인스턴스 시간당 가격	단위 시간당 가격
`r3.xlarge`	2	10	5 (10을 2로 나눈 값)	0.05 USD	0.025 USD (0.05를 2로 나눈 값)
`r3.8xlarge`	8	10	2 (10을 8로 나눈 후 올림한 결과)	0.10 USD	0.0125 USD (0.10을 8로 나눈 값)

EC2 집합 인스턴스 가중치를 사용하여 다음과 같이 원하는 목표 용량을 이행 시점의 단위당 최저 가격으로 풀에서 프로비저닝합니다.

1. EC2 집합의 목표 용량을 인스턴스(기본값) 또는 선택한 단위(예: 가상 CPU 수, 메모리, 스토리지 또는 처리량)로 설정합니다.
2. 단위당 가격을 설정합니다.
3. 목표 용량에 대해 인스턴스 유형이 나타내는 단위 수를 의미하는 가중치를 시작 사양마다 지정합니다.

인스턴스 가중치 부여의 예

다음과 같은 구성의 EC2 집합 요청을 고려하십시오.

- 목표 용량은 24
- 인스턴스 유형이 `r3.2xlarge`이고 가중치가 6인 시작 사양
- 인스턴스 유형이 `c3.xlarge`이고 가중치가 5인 시작 사양

가중치는 목표 용량에 대하여 인스턴스 유형이 나타내는 단위 수를 의미합니다. 첫 번째 시작 사양에서 단위당 최저 가격(인스턴스 시간당 `r3.2xlarge`의 가격을 6으로 나눈 값)을 제공하는 경우 EC2 집합은 이 인스턴스 중 4개(24를 6으로 나눈 값)를 시작합니다.

두 번째 시작 사양에서 단위당 최저 가격(인스턴스 시간당 `c3.xlarge`에 대한 가격을 5로 나눈 값)을 제공하는 경우 EC2 집합은 이들 인스턴스 중 5개(24를 5로 나눈 결과를 올림한 값)를 시작합니다.

인스턴스 가중치 부여 및 할당 전략

다음과 같은 구성의 EC2 집합 요청을 고려하십시오.

- 목표 용량 스팟 인스턴스 30개
- 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

EC2 집합이 4개의 인스턴스(30을 8로 나눈 결과를 올림한 값)를 시작합니다. `lowest-price` 전략 사용 시, 4개의 인스턴스는 전부 단위당 최저 가격을 제공하는 풀에서 가져옵니다. `diversified` 전략 사용 시 플릿은 풀 3개에서 각각 인스턴스 1개를 시작하고 풀 3개 중 어디에 있든 네 번째 인스턴스가 단위당 최저 가격을 제공합니다.

자습서: 인스턴스 가중치를 부여한 EC2 집합 사용

이 자습서에서는 Example Corp이라는 가상의 회사를 통해 인스턴스 가중치를 사용하여 EC2 집합을 요청하는 프로세스를 설명합니다.

목표

제약 회사인 Example Corp은 암 퇴치 효과가 있는 화합물을 검출하는 데 Amazon EC2의 컴퓨팅 능력을 사용하려고 합니다.

계획

Example Corp은 먼저 스팟 모범 사례를 살펴봅니다. 그런 다음 Example Corp이 EC2 집합에 대해 다음과 같은 요구 사항을 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 요건에 적합할 것이라는 결정을 내립니다.

인스턴스 유형	메모리(GiB)	vCPUs
r3.2xlarge	61	8
r3.4xlarge	122	16
r3.8xlarge	244	32

목표 용량 단위

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 Corp은 단위 1개당 애플리케이션의 기본 용량(60GB 메모리, vCPU 8개)을 고려하여 기본 용량의 20배가 요구에 적합하겠다고 결정하고 그래서 EC2 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 EC2 집합 요청 시 1, 2 및 4의 인스턴스 가중치를 각 시작 구성에 할당합니다.

단위 시간당 가격

Example Corp은 인스턴스 시간당 온디맨드 가격을 시작 가격으로 사용합니다. 그 밖에 최근 스팟 가격을 사용하거나, 둘을 조합할 수도 있습니다. 단위 시간당 가격을 계산하려면 인스턴스 시간당 시작 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 가격으로 0.7 USD를 사용하기 때문에 세 가지 인스턴스 유형 모두에서 경쟁력이 있습니다. 또한 `r3.8xlarge` 시작 사양에서 단위 시간당 글로벌 가격으로 0.7 USD를, 단위 시간당 특정 가격으로 0.9 USD를 사용할 수도 있습니다.

권한 검증

Example Corp은 EC2 집합을 생성하기 전에 필요한 권한을 가진 IAM 역할이 있는지 확인합니다. 자세한 내용은 EC2 집합 사전 조건 (p. 461) 단원을 참조하십시오.

EC2 집합 생성

Example Corp은 EC2 집합에 대해 다음 구성으로 `config.json` 파일을 생성합니다.

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-07b3bc7625cdab851",
                "Version": "1"
            },
            "Overrides": [
                {
                    "InstanceType": "r3.2xlarge",
                    "SubnetId": "subnet-482e4972",
                    "WeightedCapacity": 1
                },
                {
                    "InstanceType": "r3.4xlarge",
                    "SubnetId": "subnet-482e4972",
                    "WeightedCapacity": 2
                },
                {
                    "InstanceType": "r3.8xlarge",
                    "MaxPrice": "0.90",
                    "SubnetId": "subnet-482e4972",
                    "WeightedCapacity": 4
                }
            ]
        }
    ],
    "TargetCapacitySpecification": {
```

```
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
    }
}
```

Example Corp은 다음 create-fleet 명령을 사용하여 EC2 집합을 생성합니다.

```
aws ec2 create-fleet \
    --cli-input-json file://config.json
```

자세한 내용은 EC2 집합 생성 (p. 468) 단원을 참조하십시오.

이행

할당 전략에서는 스팟 인스턴스가 어느 스팟 인스턴스 풀에서 온 것인지 확인합니다.

`lowest-price` 전략(기본 전략)을 사용할 경우 스팟 인스턴스는 이행 시점에 단위당 최저 가격의 풀에서 옵니다. 20단위의 용량을 제공하기 위해 EC2 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. EC2 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.

자습서: 기본 용량이 온디맨드인 EC2 집합 사용

이 자습서에서는 ABC Online이라는 가상의 회사를 통해 기본 용량인 온디맨드와 스팟 용량(사용할 수 있는 경우)이 있는 EC2 집합를 요청하는 프로세스를 설명합니다.

목표

식당 배달 회사인 ABC Online은 EC2 인스턴스 유형 및 구매 옵션에 Amazon EC2 용량을 프로비저닝하여 원하는 규모, 성능 및 비용을 달성하려고 합니다.

계획

ABC Online은 피크 시간의 운영을 위해 고정 용량이 필요하지만 저렴한 가격으로 더 큰 용량을 이용하고자 합니다. ABC Online은 EC2 집합에 대해 다음 요구 사항을 결정합니다.

- 온디맨드 인스턴스 용량 – ABC Online은 온디맨드 인스턴스 15개가 있어야 피크 시간의 트래픽을 수용할 수 있습니다.
- 스팟 인스턴스 용량 – ABC Online은 저렴한 비용으로 스팟 인스턴스 5개를 프로비저닝하여 성능을 개선하려고 합니다.

권한 검증

ABC Online은 EC2 집합을 생성하기 전에 필요한 권한을 가진 IAM 역할이 있는지 확인합니다. 자세한 내용은 EC2 집합 사전 조건 (p. 461) 단원을 참조하십시오.

EC2 집합 생성

ABC Online은 EC2 집합에 대해 다음 구성으로 `config.json` 파일을 생성합니다.

```
{
    "LaunchTemplateConfigs": [
        {
```

```
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-07b3bc7625cdab851",
                "Version": "2"
            }

        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
     "OnDemandTargetCapacity":15,
        "DefaultTargetCapacityType": "spot"
    }
}
```

ABC Online은 다음 create-fleet 명령을 사용하여 EC2 집합을 생성합니다.

```
aws ec2 create-fleet \
    --cli-input-json file://config.json
```

자세한 내용은 EC2 집합 생성 (p. 468) 단원을 참조하십시오.

이행

할당 전략에 따라 온디맨드 용량이 항상 충족되지만 사용 가능한 용량이 있는 경우 남아 있는 목표 용량이 스팟으로 충족되도록 결정됩니다.

EC2 집합 관리

EC2 집합을 사용하려면 총 목표 용량, 온디맨드 용량, 스팟 용량, 인스턴스에 대한 하나 이상의 시작 사양, 지불하려는 최고 가격을 포함하는 요청을 생성합니다. AMI, 인스턴스 유형, 서브넷 또는 가용 영역 및 하나 이상의 보안 그룹과 같이 인스턴스를 시작하기 위해 필릿에 필요한 정보를 정의하는 시작 템플릿을 플릿 요청에 포함해야 합니다. 인스턴스 유형, 서브넷, 가용 영역 및 지불하려는 최고 가격의 시작 사양 재정의를 지정할 수 있으며 가중치가 적용된 용량을 각 시작 사양 재정의에 할당할 수 있습니다.

플릿에 스팟 인스턴스가 포함되어 있으면 Amazon EC2에서 스팟 가격의 변화에 따라 플릿 목표 용량을 유지하려고 할 수 있습니다.

EC2 집합 요청은 요청이 만료되거나 삭제될 때까지 활성 상태로 유지됩니다. 플릿을 삭제할 때 삭제로 인해 해당 플릿의 인스턴스가 종료될지 여부를 지정할 수 있습니다.

목차

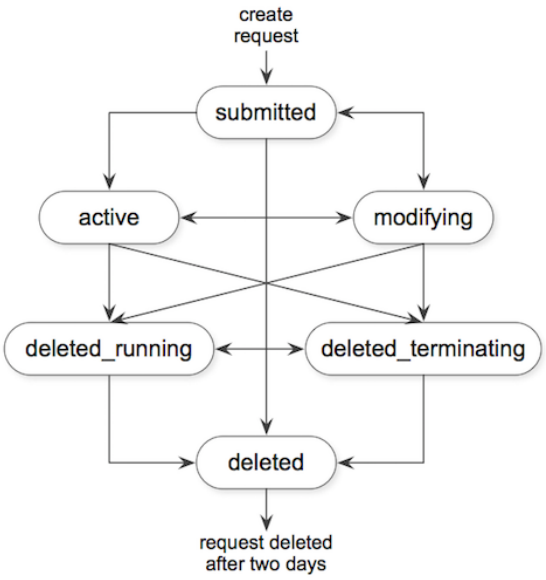
- EC2 집합 요청 상태 (p. 460)
- EC2 집합 사전 조건 (p. 461)
- EC2 집합 상태 확인 (p. 463)
- EC2 집합 JSON 구성 파일 생성 (p. 464)
- EC2 집합 생성 (p. 468)
- EC2 집합 태그 지정 (p. 471)
- EC2 집합 모니터링 (p. 460)
- EC2 집합 수정 (p. 473)
- EC2 집합 삭제 (p. 473)
- EC2 집합 구성의 예 (p. 474)

EC2 집합 요청 상태

EC2 집합 요청은 다음 상태 중 하나일 수 있습니다.

- `submitted` – EC2 집합 요청을 평가 중이며 Amazon EC2가 목표 개수만큼 인스턴스를 시작할 준비 중입니다(온디맨드 인스턴스, 스팟 인스턴스, 또는 둘 다 포함할 수 있음).
- `active` – EC2 집합 요청이 확인되었으며 Amazon EC2가 실행 중인 인스턴스를 목표 개수만큼 유지하려고 시도하고 있습니다. 요청은 수정하거나 삭제할 때까지 이 상태로 유지됩니다.
- `modifying` – EC2 집합 요청을 수정하고 있습니다. 수정이 완전히 처리되거나 요청이 삭제될 때까지 요청이 이 상태로 유지됩니다. `maintain` 요청 유형만 수정할 수 있습니다. 이 상태는 다른 요청 유형에는 적용되지 않습니다.
- `deleted_running` – EC2 집합 요청이 삭제되고 추가 인스턴스를 시작하지 않습니다. 중단되거나 종료될 때까지 기존 인스턴스가 계속 실행됩니다. 그 요청은 모든 인스턴스가 중단 또는 종료될 때까지 계속 이 상태로 유지됩니다.
- `deleted_terminating` – EC2 집합 요청이 삭제되고 해당 인스턴스가 종료됩니다. 그 요청은 모든 인스턴스가 종료될 때까지 계속 이 상태로 유지됩니다.
- `deleted` – EC2 집합이 삭제되고 실행 중인 인스턴스가 없습니다. 인스턴스가 종료되고 2일 후 요청이 삭제됩니다.

다음 그림은 EC2 집합 요청 상태의 전환을 나타냅니다. 플릿 제한을 초과하면 즉시 요청이 삭제됩니다.



EC2 집합 사전 조건

EC2 집합을 생성하려면 다음 사전 조건이 갖춰져야 합니다.

시작 템플릿

시작 템플릿에는 인스턴스 유형, 가용 영역, 지불하려는 최고 가격 등 시작할 인스턴스에 대한 정보가 포함됩니다. 자세한 내용은 시작 템플릿에서 인스턴스 시작 (p. 436) 단원을 참조하십시오.

EC2 집합의 서비스 연결 역할

`AWSServiceRoleForEC2Fleet` 역할은 사용자를 대신해 인스턴스를 요청, 시작, 종료 및 태그 지정할 권한을 EC2 집합에 부여합니다. Amazon EC2는 이 서비스 연결 역할을 사용하여 다음 작업을 완료합니다.

- `ec2:RequestSpotInstances` – 스팟 인스턴스 요청.
- `ec2:TerminateInstances` – 스팟 인스턴스 종료.
- `ec2:DescribeImages` – 스팟 인스턴스용 Amazon 머신 이미지(AMI)를 설명합니다.
- `ec2:DescribeInstanceStatus` – 스팟 인스턴스의 상태를 설명합니다.

- `ec2:DescribeSubnets` – 스팟 인스턴스용 서브넷에 대해 설명합니다.
- `ec2:CreateTags` – 스팟 인스턴스에 시스템 태그를 추가합니다.

AWS CLI 또는 API를 사용하여 EC2 집합을 만들려면 먼저 이 역할이 있어야 합니다.

> **Note**
>
> `instant` EC2 집합에는 이 역할이 필요하지 않습니다.

역할을 생성하려면 다음과 같이 IAM 콘솔을 사용하십시오.

EC2 집합에 대한 AWSServiceRoleForEC2Fleet 역할 생성

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 역할을 선택한 후 역할 생성을 선택합니다.
3. 신뢰할 수 있는 유형의 엔터티 선택에서 AWS 서비스를 선택합니다.
4. Choose the service that will use this role(이 역할을 사용할 서비스 선택)에서 EC2 - Fleet(EC2 - 플릿)을 선택한 다음 Next: Permissions(다음: 권한), Next: Tags(다음: 태그), 및 Next: Review(다음: 검토)를 선택합니다.
5. 검토 페이지에서 역할 만들기를 선택합니다.

EC2 집합이 더 이상 필요 없으면 AWSServiceRoleForEC2Fleet 역할을 삭제하는 것이 좋습니다. 계정에서 이 역할을 삭제한 후 다른 플릿을 생성하면 다시 역할을 만들 수 있습니다.

자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 사용을 참조하십시오.

암호화된 AMI 및 EBS 스냅샷과 함께 사용할 CMK에 대한 액세스 권한 부여

EC2 집합 요청에서 암호화된 AMI (p. 145) 또는 암호화된 Amazon EBS 스냅샷 (p. 968)을 지정하고 암호화용 고객 관리형 고객 마스터 키(CMK)를 사용하는 경우 AWSServiceRoleForEC2Fleet 역할에 CMK를 사용할 수 있는 권한을 부여해야 Amazon EC2가 사용자를 대신하여 스팟 인스턴스를 시작할 수 있습니다. 이렇게 하려면 다음 절차에 표시된 바와 같이 CMK에 권한을 추가해야 합니다.

권한을 제공할 때 권한 부여는 키 정책을 대체합니다. 자세한 내용은 AWS Key Management Service Developer Guide의 권한 부여 사용 및 AWS KMS에서 키 정책 사용 단원을 참조하십시오.

CMK를 사용할 수 있도록 AWSServiceRoleForEC2Fleet 역할 권한을 부여하려면

- create-grant 명령을 사용하여 CMK에 권한을 추가하고 허용된 작업을 수행할 수 있는 권한이 부여된 보안 주체(AWSServiceRoleForEC2Fleet 서비스 연결 역할)를 지정합니다. CMK는 CMK의 `key-id` 파라미터 및 ARN에 의해 지정됩니다. 보안 주체는 AWSServiceRoleForEC2Fleet 서비스 연결 역할의 `grantee-principal` 파라미터 및 ARN에 의해 지정됩니다.

```
aws kms create-grant \
    --region us-east-1 \
    --key-id arn:aws:kms:us-
east-1:444455556666:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/AWSServiceRoleForEC2Fleet \
    --operations "Decrypt" "Encrypt" "GenerateDataKey"
 "GenerateDataKeyWithoutPlaintext" "CreateGrant" "DescribeKey" "ReEncryptFrom"
 "ReEncryptTo"
```

EC2 집합 및 IAM 사용자

EC2 집합을 생성하거나 관리하는 IAM 사용자에게는 다음과 같이 필요한 권한을 부여해야 합니다.

EC2 집합의 IAM 사용자 권한을 부여하려면

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 정책을 선택합니다.
3. 정책 생성을 선택합니다.
4. 정책 생성 페이지에서 JSON 탭을 선택한 다음, 텍스트를 다음과 같이 바꾸고 정책 검토를 선택합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
              "iam:ListRoles",
              "iam:PassRole",
              "iam:ListInstanceProfiles"
            ],
            "Resource": "*"
        }
    ]
}
```

   `ec2:*`는 모든 Amazon EC2 API 작업을 호출할 수 있는 IAM 사용자 권한을 부여합니다. 사용자를 특정 Amazon EC2 API 작업으로 제한하려면 해당 작업을 대신 지정하십시오.

   IAM 사용자에는 기존 IAM 역할을 열거하는 `iam:ListRoles` 작업, EC2 집합 역할을 지정하는 `iam:PassRole` 작업 및 기존 인스턴스 프로파일을 열거하는 `iam:ListInstanceProfiles` 작업을 호출할 수 있는 권한이 있어야 합니다.

   (선택 사항) IAM 사용자가 IAM 콘솔을 사용하여 역할 또는 인스턴스 프로파일을 생성할 수 있도록 하려면 정책에 다음 작업도 추가해야 합니다.

   - `iam:AddRoleToInstanceProfile`
   - `iam:AttachRolePolicy`
   - `iam:CreateInstanceProfile`
   - `iam:CreateRole`
   - `iam:GetRole`
   - `iam:ListPolicies`
5. 정책 검토 페이지에 정책 이름과 설명을 입력한 다음 정책 생성을 선택합니다.
6. 탐색 창에서 사용자를 선택하고 사용자를 선택합니다.
7. 권한 탭에서 권한 추가를 선택합니다.
8. Attach existing policies directly(기존 정책 직접 연결)를 선택합니다. 앞에서 만든 정책을 선택하고 다음: 검토(Next: Review)를 선택합니다.
9. 권한 추가를 선택합니다.

EC2 집합 상태 확인

EC2 집합은 분마다 플릿의 인스턴스 상태를 확인합니다. 인스턴스의 상태는 `healthy` 또는 `unhealthy`입니다. 플릿은 Amazon EC2에서 제공하는 상태 확인을 사용하여 인스턴스 상태를 판단합니다. 세 번의 연속

상태 확인에서 인스턴스 상태 또는 시스템 상태가 `impaired`이면, 해당 인스턴스의 상태는 `unhealthy`입니다. 그렇지 않으면 상태는 `healthy`입니다. 자세한 내용은 인스턴스 상태 확인 (p. 603) 단원을 참조하십시오.

비정상 인스턴스를 교체하도록 EC2 집합을 구성할 수 있습니다. 상태 확인 교체를 활성화하면 상태가 `unhealthy`로 보고된 인스턴스가 교체됩니다. 플릿은 비정상 인스턴스가 교체되는 동안 최대 몇 분간 목표 용량을 밑돌 수 있습니다.

요구 사항

- 상태 확인 교체는 목표 용량을 유지하는 EC2 집합에서만 지원되고 일회성 플릿에서는 지원되지 않습니다.
- 비정상 인스턴스를 생성할 경우에만 이를 교체하도록 EC2 집합을 구성할 수 있습니다.
- IAM 사용자는 `ec2:DescribeInstanceStatus` 작업을 호출할 권한이 있는 경우에만 상태 확인 교체를 사용할 수 있습니다.

EC2 집합 JSON 구성 파일 생성

EC2 집합을 생성하려면 시작 템플릿, 총 목표 용량 및 기본 구매 옵션이 온디맨드인지 스팟인지 여부를 지정해야 합니다. 이 파라미터를 지정하지 않이면 플릿에 기본값이 사용됩니다. 플릿 구성 파라미터의 전체 목록을 보려면 다음과 같이 JSON 파일을 생성할 수 있습니다.

명령줄을 사용하여 가능한 모든 EC2 집합 파라미터가 포함된 JSON 파일을 생성하려면

- create-fleet(AWS CLI) 명령 및 `--generate-cli-skeleton` 파라미터를 사용하여 EC2 집합 JSON 파일을 생성합니다.

```
aws ec2 create-fleet \
    --generate-cli-skeleton
```

사용할 수 있는 EC2 집합 파라미터는 다음과 같습니다.

```
{
    "DryRun": true,
    "ClientToken": "",
    "SpotOptions": {
        "AllocationStrategy": "lowest-price",
        "InstanceInterruptionBehavior": "hibernate",
        "InstancePoolsToUseCount": 0,
        "SingleInstanceType": true,
        "SingleAvailabilityZone": true,
        "MaxTotalPrice": 0,
        "MinTargetCapacity": 0
    },
    "OnDemandOptions": {
        "AllocationStrategy": "prioritized",
        "SingleInstanceType": true,
        "SingleAvailabilityZone": true,
        "MaxTotalPrice": 0,
        "MinTargetCapacity": 0
    },
    "ExcessCapacityTerminationPolicy": "termination",
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "",
                "LaunchTemplateName": "",
                "Version": ""
            },
            "Overrides": [
```

```
                {
                    "InstanceType": "t2.micro",
                    "MaxPrice": "",
                    "SubnetId": "",
                    "AvailabilityZone": "",
                    "WeightedCapacity": null,
                    "Priority": null,
                    "Placement": {
                        "AvailabilityZone": "",
                        "Affinity": "",
                        "GroupName": "",
                        "PartitionNumber": 0,
                        "HostId": "",
                        "Tenancy": "dedicated",
                        "SpreadDomain": ""
                }
            ]
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 0,
        "OnDemandTargetCapacity": 0,
        "SpotTargetCapacity": 0,
        "DefaultTargetCapacityType": "spot"
    },
    "TerminateInstancesWithExpiration": true,
    "Type": "maintain",
    "ValidFrom": "1970-01-01T00:00:00",
    "ValidUntil": "1970-01-01T00:00:00",
    "ReplaceUnhealthyInstances": true,
    "TagSpecifications": [
        {
            "ResourceType": "fleet",
            "Tags": [
                {
                    "Key": "",
                    "Value": ""
                }
            ]
        }
    ]
}
```

EC2 집합 JSON 구성 파일 참조

Note

모든 파라미터 값에 소문자를 사용하십시오. 그렇지 않으면 Amazon EC2에서 JSON 파일을 사용하여 EC2 집합을 시작할 때 오류가 발생합니다.

AllocationStrategy(SpotOptions를 위한)

(선택 사항) EC2 집합에 지정된 스팟 인스턴스 풀에 스팟 인스턴스 목표 용량을 할당하는 방법을 나타냅니다. 유효한 값은 `lowest-price` 및 `diversified`입니다. 기본값은 `lowest-price`입니다. 필요에 맞는 할당 전략을 지정합니다. 자세한 내용은 스팟 인스턴스를 위한 할당 전략 (p. 453) 단원을 참조하십시오.

InstanceInterruptionBehavior

(선택) 스팟 인스턴스가 중단될 경우 동작입니다. 유효 값은 `hibernate`, `stop` 및 `terminate`입니다. 스팟 서비스는 중단된 스팟 인스턴스를 종료하도록 기본 설정되어 있습니다. 플릿 유형이 `maintain`인 경우 스팟 서비스가 중단된 스팟 인스턴스를 최대 절전 모드로 전환하거나 중지하도록 지정할 수 있습니다.

InstancePoolsToUseCount

대상 스팟 용량을 할당할 스팟 풀 수입니다. 스팟 AllocationStrategy가 `lowest-price`로 설정된 경우에만 유효합니다. EC2 집합은 가장 저렴한 스팟 풀을 선택하고 지정한 스팟 풀 수에 걸쳐 대상 스팟 용량을 균등하게 할당합니다.

SingleInstanceType

플릿이 단일 인스턴스 유형을 사용하여 플릿에 있는 모든 스팟 인스턴스를 시작하는 것을 나타냅니다.

SingleAvailabilityZone

플릿이 단일 가용 영역으로 모든 스팟 인스턴스를 시작하는 것을 나타냅니다.

MaxTotalPrice

스팟 인스턴스에 대해 지불할 최대 시간당 금액입니다.

MinTargetCapacity

플릿의 스팟 인스턴스에 대한 최소 대상 용량입니다. 최소 대상 용량에 도달하지 못하면 플릿은 아무 인스턴스도 시작하지 않습니다.

AllocationStrategy(OnDemandOptions를 위한)

온디맨드 용량을 채우기 위해 사용할 시작 템플릿 재정의 순서입니다. `lowest-price`를 지정하면 EC2 집합가 가격을 통해 순서를 결정하여 최저 가격을 먼저 시작합니다. 우선 순위를 지정하면 EC2 집합가 각 시작 템플릿 재정의에 할당된 우선 순위를 사용하여 최우선 순위를 먼저 시작합니다. 값을 지정하지 않으면 EC2 집합가 `lowest-price`를 기본값으로 사용합니다.

SingleInstanceType

플릿이 단일 인스턴스 유형을 사용하여 플릿에 있는 모든 온디맨드 인스턴스를 시작하는 것을 나타냅니다.

SingleAvailabilityZone

플릿이 단일 가용 영역으로 모든 온디맨드 인스턴스를 시작하는 것을 나타냅니다.

MaxTotalPrice

온디맨드 인스턴스에 대해 지불할 최대 시간당 금액입니다.

MinTargetCapacity

플릿의 온디맨드 인스턴스에 대한 최소 대상 용량입니다. 최소 대상 용량에 도달하지 못하면 플릿은 아무 인스턴스도 시작하지 않습니다.

ExcessCapacityTerminationPolicy

(선택 사항) EC2 집합의 총 목표 용량이 EC2 집합의 현재 크기보다 작아질 경우 실행 중인 인스턴스를 종료할지 여부를 나타냅니다. 유효한 값은 `no-termination` 및 `termination`입니다.

LaunchTemplateId

사용할 시작 템플릿의 ID입니다. 시작 템플릿 ID 또는 시작 템플릿 이름을 지정해야 합니다. 시작 템플릿은 Amazon 머신 이미지(AMI)를 지정해야 합니다. 시작 템플릿 생성에 대한 자세한 내용은 시작 템플릿에서 인스턴스 시작 (p. 436)을 참조하십시오.

LaunchTemplateName

사용할 시작 템플릿의 이름입니다. 시작 템플릿 ID 또는 시작 템플릿 이름을 지정해야 합니다. 시작 템플릿은 Amazon 머신 이미지(AMI)를 지정해야 합니다. 자세한 내용은 시작 템플릿에서 인스턴스 시작 (p. 436) 단원을 참조하십시오.

버전

시작 템플릿 버전 번호입니다.

InstanceType

(선택) 인스턴스 유형입니다. 입력하면 이 값이 시작 템플릿을 재정의합니다. 인스턴스 유형은 필요한 최소 하드웨어 사양(vCPU, 메모리 또는 스토리지)을 갖춰야 합니다.

MaxPrice

(선택) 스팟 인스턴스에 대해 지불하려는 단위 시간당 최고 가격입니다. 입력하면 이 값이 시작 템플릿을 재정의합니다. 기본 최고 가격(온디맨드 가격)을 사용하거나 지불하려는 최고 가격을 지정할 수 있습니다. 최고 가격이 지정한 인스턴스 유형의 스팟 가격보다 낮으면 스팟 인스턴스가 시작되지 않습니다.

SubnetId

(선택) 인스턴스를 시작할 서브넷 ID입니다. 입력하면 이 값이 시작 템플릿을 재정의합니다.

새 VPC를 생성하려면 Amazon VPC 콘솔로 이동합니다. 마친 후 JSON 파일로 돌아와 새 서브넷 ID를 입력합니다.

AvailabilityZone

(선택) 인스턴스를 시작할 가용 영역입니다. AWS가 스팟 인스턴스를 위한 영역을 선택하도록 하는 것이 기본값입니다. 원한다면 특정 영역을 지정할 수 있습니다. 입력하면 이 값이 시작 템플릿을 재정의합니다.

하나 이상의 가용 영역을 지정합니다. 영역에 서브넷이 2개 이상 있으면 알맞은 서브넷을 지정하십시오. 서브넷을 추가하려면 Amazon VPC 콘솔로 이동합니다. 마친 후 JSON 파일로 돌아와 새 서브넷 ID를 입력합니다.

WeightedCapacity

(선택) 지정된 인스턴스 유형에서 제공하는 단위 수입니다. 입력하면 이 값이 시작 템플릿을 재정의합니다.

우선 순위

시작 템플릿 재정의에 대한 우선 순위입니다. AllocationStrategy를 `prioritized`로 설정한 경우, EC2 집합가 우선 순위를 사용하여 온디맨드 용량을 채우기 위해 어떤 시작 템플릿 재정의를 먼저 사용할지 결정합니다. 가장 높은 우선 순위를 먼저 시작합니다. `0`부터 시작하는 모든 숫자가 유효한 값입니다. 숫자가 작을수록 우선 순위가 높아집니다. 숫자를 설정하지 않으면 재정의가 가장 낮은 우선 순위를 갖습니다.

TotalTargetCapacity

시작할 인스턴스 수입니다. vCPU, 메모리, 스토리지 같이 애플리케이션 워크로드에 중요한 인스턴스 또는 성능 특성을 선택할 수 있습니다. 요청 유형이 `maintain`일 경우, 목표 용량을 0으로 설정하고 나중에 용량을 추가할 수 있습니다.

OnDemandTargetCapacity

(선택 사항) 시작할 온디맨드 인스턴스 수입니다. 이 수는 `TotalTargetCapacity` 미만이어야 합니다.

SpotTargetCapacity

(선택 사항) 시작할 스팟 인스턴스 수입니다. 이 수는 `TotalTargetCapacity` 미만이어야 합니다.

DefaultTargetCapacityType

`TotalTargetCapacity`의 값이 `OnDemandTargetCapacity` 및`SpotTargetCapacity`의 조합된 값보다 크면 여기에 지정된 인스턴스 구매 옵션에 따라 두 값의 차이가 시작됩니다. 유효한 값은 `on-demand` 또는 `spot`입니다.

TerminateInstancesWithExpiration

(선택 사항) 기본적으로 EC2 집합 요청이 만료되면 Amazon EC2에서 인스턴스를 종료합니다. 기본값은 `true`입니다. 요청이 만료된 후에도 계속 실행하려면 이 파라미터의 값을 입력하지 마십시오.

Type

(선택 사항) EC2 집합이 원하는 용량을 얻기 위해 동기식 일회성 요청(`instant`) 또는 비동기식 일회성 요청(`request`)을 제출했으나 용량을 유지하려는 시도를 하지 않았거나, 용량을 사용할 수 없는 경우 대체 용량 풀에서 요청을 제출했거나, 원하는 용량을 얻기 위해 비동기식 요청을 제출하고 중단된 스팟 인스턴스를 보충해 원하는 용량을 계속해서 유지할지(`maintain`) 여부를 나타냅니다. 유효 값은 `instant`, `request` 및 `maintain`입니다. 기본 값은 `maintain`입니다. 자세한 내용은 EC2 집합 요청 유형 (p. 452) 단원을 참조하십시오.

ValidFrom

(선택) 특정 기간에만 유효한 요청을 생성하려면 시작 날짜를 입력합니다.

ValidUntil

(선택) 특정 기간에만 유효한 요청을 생성하려면 종료 날짜를 입력합니다.

ReplaceUnhealthyInstances

(선택 사항) 플릿에 대해 `maintain` 작업을 수행하도록 구성된 EC2 집합의 비정상 인스턴스를 교체하려면 `true`를 입력합니다. 그렇지 않은 경우 이 파라미터를 비워두십시오.

TagSpecifications

(선택 사항) 생성 시 EC2 집합 요청에 태그를 지정하기 위한 키-값 페어입니다. `ResourceType`의 값은 `fleet`이어야 합니다. 그렇지 않으면 집합 요청이 실패합니다. 시작 시 인스턴스에 태그를 지정하려면 시작 템플릿 (p. 438)에서 태그를 지정합니다. 시작 후 태그 지정에 대한 자세한 내용은 리소스에 태그 지정 (p. 1070) 단원을 참조하십시오.

EC2 집합 생성

EC2 집합을 생성할 때 인스턴스 유형, 가용 영역, 지불하려는 최고 가격 등 시작할 인스턴스에 대한 정보를 포함하는 시작 템플릿을 지정해야 합니다.

시작 템플릿을 재정의하는 여러 시작 사양이 포함된 EC2 집합을 생성할 수 있습니다. 시작 사양은 인스턴스 유형, 가용 영역, 서브넷, 최고 가격에 따라 달라질 수 있으며 다른 가중치 용량을 포함할 수 있습니다.

EC2 집합을 생성할 때 JSON 파일을 사용하여 시작할 인스턴스에 대한 정보를 지정하십시오. 자세한 내용은 EC2 집합 JSON 구성 파일 참조 (p. 465) 단원을 참조하십시오.

EC2 집합은 AWS CLI API로만 생성할 수 있습니다.

EC2 집합을 생성하려면(AWS CLI)

- create-fleet(AWS CLI) 명령을 사용하여 EC2 집합을 생성합니다.

```
aws ec2 create-fleet \
    --cli-input-json file://file_name.json
```

구성 파일에 대한 예시는 EC2 집합 구성의 예 (p. 474) 단원을 참조하십시오.

다음은 `request` 또는 `maintain` 유형의 플릿에 대한 예제 출력입니다.

```
{
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

다음은 목표 용량을 시작한 `instant` 유형의 플릿에 대한 예제 출력입니다.

```
{
  "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",
  "Errors": [],
  "Instances": [
    {
      "LaunchTemplateAndOverrides": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
          "Version": "1"
        },
        "Overrides": {
          "InstanceType": "c5.large",
          "AvailabilityZone": "us-east-1a"
        }
      },
      "Lifecycle": "on-demand",
      "InstanceIds": [
        "i-1234567890abcdef0",
        "i-9876543210abcdef9"
      ],
      "InstanceType": "c5.large",
      "Platform": null
    },
    {
      "LaunchTemplateAndOverrides": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
          "Version": "1"
        },
        "Overrides": {
          "InstanceType": "c4.large",
          "AvailabilityZone": "us-east-1a"
        }
      },
      "Lifecycle": "on-demand",
      "InstanceIds": [
        "i-5678901234abcdef0",
        "i-5432109876abcdef9"
      ],
      "InstanceType": "c4.large",
      "Platform": null
    },
  ]
}
```

다음은 시작되지 않은 인스턴스에 대한 오류와 함께 목표 용량의 일부를 시작한 `instant` 유형의 플릿에 대한 예제 출력입니다.

```
{
  "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",
  "Errors": [
    {
      "LaunchTemplateAndOverrides": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
          "Version": "1"
        },
        "Overrides": {
          "InstanceType": "c4.xlarge",
          "AvailabilityZone": "us-east-1a",
        }
      },
      "Lifecycle": "on-demand",
      "ErrorCode": "InsufficientInstanceCapacity",
```

```
      "ErrorMessage": "",
      "InstanceType": "c4.xlarge",
      "Platform": null
    },
  ],
  "Instances": [
    {
      "LaunchTemplateAndOverrides": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
          "Version": "1"
        },
        "Overrides": {
          "InstanceType": "c5.large",
          "AvailabilityZone": "us-east-1a"
        }
      },
      "Lifecycle": "on-demand",
      "InstanceIds": [
        "i-1234567890abcdef0",
        "i-9876543210abcdef9"
      ],
      "InstanceType": "c5.large",
      "Platform": null
    },
  ]
}
```

어떤 인스턴스도 시작하지 않은 `instant` 유형의 플릿에 대한 예제 출력입니다.

```
{
  "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",
  "Errors": [
    {
      "LaunchTemplateAndOverrides": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
          "Version": "1"
        },
        "Overrides": {
          "InstanceType": "c4.xlarge",
          "AvailabilityZone": "us-east-1a",
        }
      },
      "Lifecycle": "on-demand",
      "ErrorCode": "InsufficientCapacity",
      "ErrorMessage": "",
      "InstanceType": "c4.xlarge",
      "Platform": null
    },
    {
      "LaunchTemplateAndOverrides": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
          "Version": "1"
        },
        "Overrides": {
          "InstanceType": "c5.large",
          "AvailabilityZone": "us-east-1a",
        }
      },
      "Lifecycle": "on-demand",
      "ErrorCode": "InsufficientCapacity",
      "ErrorMessage": "",
      "InstanceType": "c5.large",
```

```
      "Platform": null
    },
  ],
  "Instances": []
}
```

EC2 집합 태그 지정

EC2 집합 요청을 쉽게 분류하고 관리할 수 있도록 사용자 지정 메타데이터로 이 요청에 태그를 지정할 수 있습니다. 자세한 내용은 Amazon EC2리소스에 태그 지정 (p. 1068) 단원을 참조하십시오.

EC2 집합 요청을 만들 때 또는 만든 후 요청에 태그를 지정할 수 있습니다. 집합 요청에 지정된 태그는 플릿에서 시작된 인스턴스에 지정되지 않습니다.

새 EC2 집합 요청에 태그를 지정하려면

EC2 집합 요청 생성 시 요청에 태그를 지정하려면 플릿을 생성하는 데 사용되는 JSON 파일 (p. 464)에 키-값 페어를 지정하십시오. `ResourceType`의 값은 `fleet`이어야 합니다. 다른 값을 지정하면 집합 요청이 실패합니다.

EC2 집합에서 시작한 인스턴스에 태그를 지정하려면

인스턴스가 플릿에서 시작할 때 이 인스턴스에 태그를 지정하려면 EC2 집합 요청에서 참조되는 시작 템플릿 (p. 438)에서 태그를 지정하십시오.

기존 EC2 집합 요청 및 인스턴스에 태그를 지정하려면(AWS CLI)

create-tags 명령을 사용하여 기존 리소스에 태그를 지정합니다.

```
aws ec2 create-tags \
    --resources fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE i-1234567890abcdef0 \
    --tags Key=purpose,Value=test
```

EC2 집합 모니터링

EC2 집합은 가용 용량이 있을 때 온디맨드 인스턴스를 시작하며, 최고 가격이 스팟 가격을 초과하고 가용 용량이 있을 때 스팟 인스턴스를 시작합니다. 온디맨드 인스턴스는 사용자가 종료할 때까지 실행되고 스팟 인스턴스는 중단되거나 사용자가 종료할 때까지 실행됩니다.

반환되는 실행 중 인스턴스 목록은 주기적으로 새로 고쳐지며 최신 상태가 아닐 수도 있습니다.

EC2 집합을 모니터링하려면(AWS CLI)

describe-fleets 명령을 사용하여 EC2 집합을 설명합니다.

```
aws ec2 describe-fleets
```

다음은 예제 출력입니다.

```
{
    "Fleets": [
        {
            "Type": "maintain",
            "FulfilledCapacity": 2.0,
            "LaunchTemplateConfigs": [
                {
                    "LaunchTemplateSpecification": {
```

```
                        "Version": "2",
                        "LaunchTemplateId": "lt-07b3bc7625cdab851"
                    }
                }
            ],
            "TerminateInstancesWithExpiration": false,
            "TargetCapacitySpecification": {
                "OnDemandTargetCapacity": 0,
                "SpotTargetCapacity": 2,
                "TotalTargetCapacity": 2,
                "DefaultTargetCapacityType": "spot"
            },
            "FulfilledOnDemandCapacity": 0.0,
            "ActivityStatus": "fulfilled",
            "FleetId": "fleet-76e13e99-01ef-4bd6-ba9b-9208de883e7f",
            "ReplaceUnhealthyInstances": false,
            "SpotOptions": {
                "InstanceInterruptionBehavior": "terminate",
                "InstancePoolsToUseCount": 1,
                "AllocationStrategy": "lowest-price"
            },
            "FleetState": "active",
            "ExcessCapacityTerminationPolicy": "termination",
            "CreateTime": "2018-04-10T16:46:03.000Z"
        }
    ]
}
```

describe-fleet-instances 명령을 사용하여 지정한 EC2 집합의 인스턴스를 설명합니다.

```
aws ec2 describe-fleet-instances \
    --fleet-id fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

```
{
    "ActiveInstances": [
        {
            "InstanceId": "i-09cd595998cb3765e",
            "InstanceHealth": "healthy",
            "InstanceType": "m4.large",
            "SpotInstanceRequestId": "sir-86k84j6p"
        },
        {
            "InstanceId": "i-09cf95167ca219f17",
            "InstanceHealth": "healthy",
            "InstanceType": "m4.large",
            "SpotInstanceRequestId": "sir-dvxi7fsm"
        }
    ],
    "FleetId": "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
}
```

describe-fleet-history 명령을 사용하여 지정한 시간 동안 지정한 EC2 집합의 기록을 설명합니다.

```
aws ec2 describe-fleet-history --fleet-request-id fleet-73fbd2ce-
aa30-494c-8788-1cee4EXAMPLE --start-time 2018-04-10T00:00:00Z
```

```
{
    "HistoryRecords": [],
    "FleetId": "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
    "LastEvaluatedTime": "1970-01-01T00:00:00.000Z",
    "StartTime": "2018-04-09T23:53:20.000Z"
```

```
}
```

EC2 집합 수정

submitted 또는 active 상태인 EC2 집합을 수정할 수 있습니다. 플릿을 수정할 때 플릿은 modifying 상태가 됩니다.

유형이 maintain인 EC2 집합만 수정할 수 있습니다. 유형이 request 또는 instant인 EC2 집합은 수정할 수 없습니다.

EC2 집합의 다음 파라미터를 수정할 수 있습니다.

- target-capacity-specification – TotalTargetCapacity, OnDemandTargetCapacity 및 SpotTargetCapacity의 목표 용량을 늘리거나 줄입니다.
- excess-capacity-termination-policy – EC2 집합의 총 목표 용량이 플릿의 현재 크기보다 작아지면 실행 중인 인스턴스를 종료할지 여부입니다. 유효한 값은 no-termination 및 termination입니다.

목표 용량을 늘리면 EC2 집합이 DefaultTargetCapacityType에 지정한 인스턴스 구입 옵션(온디맨드 인스턴스 또는 스팟 인스턴스)에 따라 추가 인스턴스를 시작합니다.

DefaultTargetCapacityType이 spot이면 EC2 집합이 할당 전략에 따라 추가 스팟 인스턴스를 시작합니다. 할당 전략이 lowest-price이면 플릿이 요청에 있는 최저 가격의 스팟 인스턴스 풀에서 인스턴스를 시작합니다. 할당 전략이 diversified이면 플릿이 요청에 있는 풀에 인스턴스를 배포합니다.

목표 용량을 줄이면 EC2 집합이 새 목표 용량을 초과하는 모든 열린 요청을 삭제합니다. 플릿 크기가 새 목표 용량에 도달할 때까지 플릿이 인스턴스를 종료하도록 요청할 수 있습니다. 할당 전략이 lowest-price이면 플릿이 단위당 최고 가격의 인스턴스를 종료합니다. 할당 전략이 diversified이면 플릿이 풀 전체의 인스턴스를 종료합니다. 또는 EC2 집합이 플릿을 현재 크기로 유지하되 중단된 스팟 인스턴스나 사용자가 수동으로 종료하는 인스턴스를 교체하지 않도록 요청할 수 있습니다.

목표 용량이 줄어 EC2 집합이 스팟 인스턴스를 종료할 때 인스턴스는 스팟 인스턴스 중단 공지를 받습니다.

EC2 집합을 수정하려면(AWS CLI)

modify-fleet 명령을 사용하여 지정된 EC2 집합의 목표 용량을 업데이트합니다.

```
aws ec2 modify-fleet \
    --fleet-id fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --target-capacity-specification TotalTargetCapacity=20
```

목표 용량을 줄이고 플릿은 현재 크기로 유지하려는 경우 다음과 같이 이전의 명령을 수정할 수 있습니다.

```
aws ec2 modify-fleet \
    --fleet-id fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --target-capacity-specification TotalTargetCapacity=10 \
    --excess-capacity-termination-policy no-termination
```

EC2 집합 삭제

EC2 집합이 더 이상 필요 없으면 삭제할 수 있습니다. 플릿을 삭제한 후에는 새로운 인스턴스를 시작하지 않습니다.

EC2 집합이 인스턴스를 종료할지 여부를 지정해야 합니다. 플릿이 삭제되면 인스턴스가 종료되도록 지정할 경우 deleted_terminating 상태가 되고, 그렇지 않으면 deleted_running 상태가 되어 인스턴스가 중단되거나 수동으로 종료될 때까지 계속 실행됩니다.

유형이 request 및 maintain인 플릿만 삭제할 수 있습니다. instant EC2 집합은 삭제할 수 없습니다.

EC2 집합을 삭제하려면(AWS CLI)

delete-fleets 명령과 `--terminate-instances` 파라미터를 사용하여 지정된 EC2 집합을 삭제하고 인스턴스를 종료합니다.

```
aws ec2 delete-fleets \
    --fleet-ids fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --terminate-instances
```

다음은 예제 출력입니다.

```
{
    "UnsuccessfulFleetDeletions": [],
    "SuccessfulFleetDeletions": [
        {
            "CurrentFleetState": "deleted_terminating",
            "PreviousFleetState": "active",
            "FleetId": "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
        }
    ]
}
```

`--no-terminate-instances` 파라미터를 사용해 이전의 명령을 수정하여 인스턴스를 수정하지 않고 지정된 EC2 집합을 삭제할 수 있습니다.

```
aws ec2 delete-fleets \
    --fleet-ids fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --no-terminate-instances
```

다음은 예제 출력입니다.

```
{
    "UnsuccessfulFleetDeletions": [],
    "SuccessfulFleetDeletions": [
        {
            "CurrentFleetState": "deleted_running",
            "PreviousFleetState": "active",
            "FleetId": "fleet-4b8aaae8-dfb5-436d-a4c6-3dafa4c6b7dcEXAMPLE"
        }
    ]
}
```

EC2 집합 구성의 예

다음 예제에서는 create-fleet 명령에 사용하여 EC2 집합을 생성할 수 있는 시작 구성을 보여줍니다. create-fleet 파라미터에 대한 자세한 내용은 EC2 집합 JSON 구성 파일 참조 (p. 465) 단원을 참고하십시오.

예

- 예 1: 스팟 인스턴스를 기본 구입 옵션으로 시작 (p. 475)
- 예 2: 온디맨드 인스턴스를 기본 구입 옵션으로 시작 (p. 475)
- 예 3: 온디맨드 인스턴스를 기본 용량으로 시작 (p. 475)
- 예 4: lowest-price 할당 전략을 사용하여 스팟 인스턴스 시작 (p. 476)
- 예 5: 용량 예약 및 prioritized 할당 전략을 사용하여 온디맨드 인스턴스 시작 (p. 476)
- 예 6: 총 목표 용량이 미사용 용량 예약 수보다 많은 경우 용량 예약 및 prioritized 할당 전략을 사용하여 온디맨드 인스턴스 시작 (p. 478)
- 예 7: 용량 예약 및 lowest-price 할당 전략을 사용하여 온디맨드 인스턴스 시작 (p. 480)

- 예 8: 총 목표 용량이 미사용 용량 예약 수보다 많은 경우 용량 예약 및 lowest-price 할당 전략을 사용하여 온디맨드 인스턴스 시작 (p. 483)

예 1: 스팟 인스턴스를 기본 구입 옵션으로 시작

다음 예제에서는 EC2 집합에 필요한 최소한의 파라미터, 즉 시작 템플릿, 목표 용량 및 기본 구매 옵션을 지정합니다. 시작 템플릿은 시작 템플릿 ID와 버전 번호로 식별됩니다. 플릿의 목표 용량은 인스턴스 2개이고 기본 구입 옵션은 `spot`이므로 플릿이 스팟 인스턴스 2개를 시작합니다.

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e8c754449b27161c",
                "Version": "1"
            }

        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
        "DefaultTargetCapacityType": "spot"
    }
}
```

예 2: 온디맨드 인스턴스를 기본 구입 옵션으로 시작

다음 예제에서는 EC2 집합에 필요한 최소한의 파라미터, 즉 시작 템플릿, 목표 용량 및 기본 구매 옵션을 지정합니다. 시작 템플릿은 시작 템플릿 ID와 버전 번호로 식별됩니다. 플릿의 목표 용량은 인스턴스 2개이고 기본 구입 옵션은 `on-demand`이므로 플릿이 온디맨드 인스턴스 2개를 시작합니다.

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e8c754449b27161c",
                "Version": "1"
            }

        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
        "DefaultTargetCapacityType": "on-demand"
    }
}
```

예 3: 온디맨드 인스턴스를 기본 용량으로 시작

다음 예제에서는 총 목표 용량인 인스턴스 2개를 플릿에 지정하고 목표 용량은 온디맨드 인스턴스 1개로 지정합니다. 기본 구매 옵션은 `spot`입니다. 지정한 대로 플릿은 온디맨드 인스턴스 1개를 시작하지만 총 목표 용량을 충족하려면 인스턴스를 하나 더 시작해야 합니다. 차이에 대한 구매 옵션이 `TotalTargetCapacity – OnDemandTargetCapacity = DefaultTargetCapacityType`으로 계산되므로 플릿에서 스팟 인스턴스 1개를 시작합니다.

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e8c754449b27161c",
```

```
                "Version": "1"
            }

        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
 "OnDemandTargetCapacity":1,
        "DefaultTargetCapacityType": "spot"
    }
}
```

예 4: lowest-price 할당 전략을 사용하여 스팟 인스턴스 시작

스팟 인스턴스의 할당 전략이 지정되어 있지 않으면 기본 할당 전략인 lowest-price가 사용됩니다. 다음 예제에서는 lowest-price 할당 전략을 사용합니다. 시작 템플릿을 재정의하고 서로 인스턴스 유형은 다르지만 가중치 용량과 서브넷이 동일한 시작 사양 3개가 있습니다. 총 목표 용량은 인스턴스 2개이고 기본 구매 옵션은 spot입니다. EC2 집합는 최저 가격이 지정된 시작 사양의 인스턴스 유형을 사용하여 스팟 인스턴스 2개를 시작합니다.

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e8c754449b27161c",
                "Version": "1"
            }
  "Overrides": [
        {
          "InstanceType": "c4.large",
          "WeightedCapacity": 1,
          "SubnetId": "subnet-a4f6c5d3"
        },
        {
          "InstanceType": "c3.large",
          "WeightedCapacity": 1,
          "SubnetId": "subnet-a4f6c5d3"
        },
        {
          "InstanceType": "c5.large",
          "WeightedCapacity": 1,
          "SubnetId": "subnet-a4f6c5d3"
        }
      ]

        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
        "DefaultTargetCapacityType": "spot"
    }
}
```

예 5: 용량 예약 및 prioritized 할당 전략을 사용하여 온디맨드 인스턴스 시작

용량 예약에 대한 사용 전략을 use-capacity-reservations-first로 설정하여 온디맨드 인스턴스를 시작할 때 온디맨드 용량 예약부터 사용하도록 플릿을 구성할 수 있습니다. 그리고 여러 인스턴스 풀에 미사용 용량 예약가 있는 경우, 선택한 온디맨드 할당 전략이 적용됩니다. 이 예의 온디맨드 할당 전략은 prioritized입니다.

이 예에서는 사용할 수 있는 미사용 용량 예약가 15개 있습니다. 이것은 이 플릿의 목표 온디맨드 용량인 온디맨드 인스턴스 12개보다 많습니다.

이 계정은 3개의 풀에 다음과 같은 미사용 용량 예약 15개를 가지고 있습니다. 각 풀의 용량 예약 수는 `AvailableInstanceCount`로 표시됩니다.

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "c4.large",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "c3.large",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}

{
    "CapacityReservationId": "cr-333",
    "InstanceType": "c5.large",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount":5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}
```

다음 플릿 구성에서는 이 예제와 관련된 구성만 보여 줍니다. 온디맨드 할당 전략은 `prioritized`이고, 용량 예약의 사용 전략은 `use-capacity-reservations-first`입니다. 총 목표 용량은 12이고, 기본 목표 용량 유형은 `on-demand`입니다.

> **Note**
>
> 플릿 유형은 `instant`여야 합니다. 다른 플릿 유형에는 용량 예약 가 지원되지 않습니다.

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-1234567890abcdefg",
                "Version": "1"
            }
            "Overrides": [
                {
                  "InstanceType": "c4.large",
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1,
                  "Priority": 1.0
                },
                {
                  "InstanceType": "c3.large",
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1,
                  "Priority": 2.0
                },
                {
                  "InstanceType": "c5.large",
```

```
                    "AvailabilityZone": "us-east-1a",
                    "WeightedCapacity": 1,
                    "Priority": 3.0
                  }
                ]
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 12,
        "DefaultTargetCapacityType": "on-demand"
    },
    "OnDemandOptions": {
        "AllocationStrategy": "prioritized"
        "CapacityReservationOptions": {
            "UsageStrategy": "use-capacity-reservations-first"
        }
    },
    "Type": "instant",
}
```

이전 구성을 사용하여 `instant` 플릿을 생성하면 목표 용량을 충족하기 위해 다음 12개의 인스턴스가 시작됩니다.

- us-east-1a의 c4.large 온디맨드 인스턴스 5개 – us-east-1a의 c4.large가 첫 번째 우선 순위이고 사용 가능한 미사용 c4.large 용량 예약는 5개입니다.
- us-east-1a의 c3.large 온디맨드 인스턴스 5개 – us-east-1a의 c3.large가 두 번째 우선 순위이며 사용 가능한 미사용 c3.large 용량 예약 5개
- us-east-1a의 c5.large 온디맨드 인스턴스 2개 – us-east-1a의 c5.large가 세 번째 우선 순위이고 사용 가능한 미사용 c5.large 용량 예약는 5개, 그중 2개만 있으면 목표 용량 충족 가능

플릿이 시작된 후, describe-capacity-reservations를 실행하여 미사용 용량 예약가 몇 개나 남아 있는지 확인할 수 있습니다. 이 예에서는 c4.large 및 c3.large 용량 예약가 모두 사용되었고 c5.large 용량 예약 3개는 미사용 상태로 남아 있음을 보여주는 다음과 같은 응답이 나타납니다.

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "c4.large",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "c3.large",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-333",
    "InstanceType": "c5.large",
    "AvailableInstanceCount": 3
}
```

예 6: 총 목표 용량이 미사용 용량 예약 수보다 많은 경우 용량 예약 및 `prioritized` 할당 전략을 사용하여 온디맨드 인스턴스 시작

용량 예약에 대한 사용 전략을 `use-capacity-reservations-first`로 설정하여 온디맨드 인스턴스를 시작할 때 온디맨드 용량 예약부터 사용하도록 플릿을 구성할 수 있습니다. 미사용 용량 예약 수가 온디맨드 목표 용량보다 적으면 선택한 온디맨드 할당 전략에 따라 나머지 온디맨드 목표 용량이 시작됩니다. 이 예의 온디맨드 할당 전략은 `prioritized`입니다.

이 예에서는 사용할 수 있는 미사용 용량 예약가 15개 있습니다. 이는 플릿의 온디맨드 목표 용량인 온디맨드 인스턴스 16개보다 적습니다.

이 계정은 3개의 풀에 다음과 같은 미사용 용량 예약 15개를 가지고 있습니다. 각 풀의 용량 예약 수는 `AvailableInstanceCount`로 표시됩니다.

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "c4.large",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}

{
    "CapacityReservationId": "cr-111",
    "InstanceType": "c3.large",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}

{
    "CapacityReservationId": "cr-111",
    "InstanceType": "c5.large",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount":5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}
```

다음 플릿 구성에서는 이 예제와 관련된 구성만 보여 줍니다. 온디맨드 할당 전략은 `prioritized`이고, 용량 예약의 사용 전략은 `use-capacity-reservations-first`입니다. 총 목표 용량은 16이고 기본 목표 용량 유형은 `on-demand`입니다.

> **Note**
>
> 플릿 유형은 `instant`여야 합니다. 다른 플릿 유형에는 용량 예약 가 지원되지 않습니다.

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e8c754449b27161c",
                "Version": "1"
            }
            "Overrides": [
                {
                  "InstanceType": "c4.large",
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1,
                  "Priority": 1.0
                },
                {
                  "InstanceType": "c3.large",
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1,
                  "Priority": 2.0
                },
```

```
                {
                  "InstanceType": "c5.large",
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1,
                  "Priority": 3.0
                }
              ]

        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 16,
        "DefaultTargetCapacityType": "on-demand"
    },
    "OnDemandOptions": {
        "AllocationStrategy": "prioritized"
        "CapacityReservationOptions": {
            "UsageStrategy": "use-capacity-reservations-first"
        }
    },
    "Type": "instant",
}
```

이전 구성을 사용하여 `instant` 플릿을 생성한 후, 목표 용량을 충족하기 위해 다음 16개의 인스턴스가 시작됩니다.

- us-east-1a의 c4.large 온디맨드 인스턴스 6개 – us-east-1a의 c4.large가 첫 번째 우선 순위이고 사용 가능한 미사용 c4.large 용량 예약는 5개입니다. 먼저 용량 예약를 사용하여 온디맨드 인스턴스 5개를 시작하고, 온디맨드 할당 전략(이 예에서는 `prioritized`)에 따라 온디맨드 인스턴스를 추가로 시작합니다.
- us-east-1a의 c3.large 온디맨드 인스턴스 5개 – us-east-1a의 c3.large가 두 번째 우선 순위이며 사용 가능한 미사용 c3.large 용량 예약 5개
- us-east-1a의 c5.large 온디맨드 인스턴스 5개 – us-east-1a의 c5.large가 세 번째 우선 순위이며 사용 가능한 미사용 c5.large 용량 예약 5개

플릿이 시작된 후, describe-capacity-reservations를 실행하여 미사용 용량 예약가 몇 개나 남아 있는지 확인할 수 있습니다. 이 예에서는 모든 풀의 용량 예약가 모두 사용되었음을 보여 주는 다음 응답이 나타납니다.

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "c4.large",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "c3.large",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-333",
    "InstanceType": "c5.large",
    "AvailableInstanceCount": 0
}
```

예 7: 용량 예약 및 `lowest-price` 할당 전략을 사용하여 온디맨드 인스턴스 시작

용량 예약에 대한 사용 전략을 `use-capacity-reservations-first`로 설정하여 온디맨드 인스턴스를 시작할 때 온디맨드 용량 예약부터 사용하도록 플릿을 구성할 수 있습니다. 그리고 여러 인스턴스 풀에 미사용 용량 예약가 있는 경우, 선택한 온디맨드 할당 전략이 적용됩니다. 이 예의 온디맨드 할당 전략은 `lowest-price`입니다.

이 예에서는 사용할 수 있는 미사용 용량 예약가 15개 있습니다. 이것은 이 플릿의 목표 온디맨드 용량인 온디맨드 인스턴스 12개보다 많습니다.

이 계정은 3개의 풀에 다음과 같은 미사용 용량 예약 15개를 가지고 있습니다. 각 풀의 용량 예약 수는 `AvailableInstanceCount`로 표시됩니다.

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "m5.large",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "m4.xlarge",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}

{
    "CapacityReservationId": "cr-333",
    "InstanceType": "m4.2xlarge",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount":5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}
```

다음 플릿 구성에서는 이 예제와 관련된 구성만 보여 줍니다. 온디맨드 할당 전략은 `lowest-price`이고, 용량 예약의 사용 전략은 `use-capacity-reservations-first`입니다. 총 목표 용량은 12이고, 기본 목표 용량 유형은 `on-demand`입니다.

이 예에서 온디맨드 인스턴스 가격은 다음과 같습니다.

- m5.large – 시간당 0.096 USD
- m4.xlarge – 시간당 0.20 USD
- m4.2xlarge – 시간당 0.40 USD

> **Note**
>
> 플릿 유형은 `instant`여야 합니다. 다른 플릿 유형에는 용량 예약 가 지원되지 않습니다.

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e8c754449b27161c",
                "Version": "1"
            }
            "Overrides": [
                {
                  "InstanceType": "m5.large",
```

```
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1
                },
                {
                  "InstanceType": "m4.xlarge",
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1
                },
                {
                  "InstanceType": "m4.2xlarge",
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1
                }
              ]

        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 12,
        "DefaultTargetCapacityType": "on-demand"
    },
    "OnDemandOptions": {
        "AllocationStrategy": "lowest-price"
        "CapacityReservationOptions": {
            "UsageStrategy": "use-capacity-reservations-first"
        }
    },
    "Type": "instant",
}
```

이전 구성을 사용하여 `instant` 플릿을 생성하면 목표 용량을 충족하기 위해 다음 12개의 인스턴스가 시작됩니다.

- us-east-1a의 m5.large 온디맨드 인스턴스 5개 – us-east-1a의 m5.large가 최저 가격이고 사용 가능한 미사용 m5.large 용량 예약 5개.
- us-east-1a의 m4.xlarge 온디맨드 인스턴스 5개 – us-east-1a의 m4.xlarge가 두 번째 최저 가격이며 사용 가능한 미사용 m4.xlarge 용량 예약 5개
- us-east-1a의 m4.2xlarge 온디맨드 인스턴스 2개 – us-east-1a의 m4.2xlarge가 세 번째 최저 가격이고 사용 가능한 미사용 m4.2xlarge 용량 예약는 5개이며, 그중 2개만 있으면 목표 용량 충족 가능

플릿이 시작된 후, describe-capacity-reservations를 실행하여 미사용 용량 예약가 몇 개나 남아 있는지 확인할 수 있습니다. 이 예에서는 m5.large 및 m4.xlarge 용량 예약가 모두 사용되었고 m4.2xlarge 용량 예약 3개는 사용되지 않은 상태로 남아 있음을 보여주는 다음 응답이 나타납니다.

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "m5.large",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "m4.xlarge",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-333",
    "InstanceType": "m4.2xlarge",
    "AvailableInstanceCount": 3
}
```

예 8: 총 목표 용량이 미사용 용량 예약 수보다 많은 경우 용량 예약 및 `lowest-price` 할당 전략을 사용하여 온디맨드 인스턴스 시작

용량 예약에 대한 사용 전략을 `use-capacity-reservations-first`로 설정하여 온디맨드 인스턴스를 시작할 때 온디맨드 용량 예약부터 사용하도록 플릿을 구성할 수 있습니다. 미사용 용량 예약 수가 온디맨드 목표 용량보다 적으면 선택한 온디맨드 할당 전략에 따라 나머지 온디맨드 목표 용량이 시작됩니다. 이 예의 온디맨드 할당 전략은 `lowest-price`입니다.

이 예에서는 사용할 수 있는 미사용 용량 예약가 15개 있습니다. 이는 플릿의 온디맨드 목표 용량인 온디맨드 인스턴스 16개보다 적습니다.

이 계정은 3개의 풀에 다음과 같은 미사용 용량 예약 15개를 가지고 있습니다. 각 풀의 용량 예약 수는 `AvailableInstanceCount`로 표시됩니다.

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "m5.large",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "m4.xlarge",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}

{
    "CapacityReservationId": "cr-333",
    "InstanceType": "m4.2xlarge",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount":5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}
```

다음 플릿 구성에서는 이 예제와 관련된 구성만 보여 줍니다. 온디맨드 할당 전략은 `lowest-price`이고, 용량 예약의 사용 전략은 `use-capacity-reservations-first`입니다. 총 목표 용량은 16이고 기본 목표 용량 유형은 `on-demand`입니다.

이 예에서 온디맨드 인스턴스 가격은 다음과 같습니다.

- m5.large – 시간당 0.096 USD
- m4.xlarge – 시간당 0.20 USD
- m4.2xlarge – 시간당 0.40 USD

> **Note**
>
> 플릿 유형은 `instant`여야 합니다. 다른 플릿 유형에는 용량 예약 가 지원되지 않습니다.

```
{
    "LaunchTemplateConfigs": [
        {
```

```
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e8c754449b27161c",
                "Version": "1"
            }
            "Overrides": [
                {
                  "InstanceType": "m5.large",
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1
                },
                {
                  "InstanceType": "m4.xlarge",
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1
                },
                {
                  "InstanceType": "m4.2xlarge",
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1
                }
              ]

        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 16,
        "DefaultTargetCapacityType": "on-demand"
    },
    "OnDemandOptions": {
        "AllocationStrategy": "lowest-price"
        "CapacityReservationOptions": {
            "UsageStrategy": "use-capacity-reservations-first"
        }
    },
    "Type": "instant",
}
```

이전 구성을 사용하여 `instant` 플릿을 생성한 후, 목표 용량을 충족하기 위해 다음 16개의 인스턴스가 시작됩니다.

- us-east-1a의 m5.large 온디맨드 인스턴스 6개 – us-east-1a의 m5.large가 최저 가격이고 사용 가능한 미사용 m5.large 용량 예약 5개. 먼저 용량 예약를 사용하여 온디맨드 인스턴스 5개를 시작하고, 온디맨드 할당 전략(이 예에서는 `lowest-price`)에 따라 온디맨드 인스턴스를 추가로 시작합니다.
- us-east-1a의 m4.xlarge 온디맨드 인스턴스 5개 – us-east-1a의 m4.xlarge가 두 번째 최저 가격이며 사용 가능한 미사용 m4.xlarge 용량 예약 5개
- us-east-1a의 m4.2xlarge 온디맨드 인스턴스 5개 – us-east-1a의 m4.2xlarge가 세 번째 최저 가격이며 사용 가능한 미사용 m4.2xlarge 용량 예약 5개

플릿이 시작된 후, describe-capacity-reservations를 실행하여 미사용 용량 예약가 몇 개나 남아 있는지 확인할 수 있습니다. 이 예에서는 모든 풀의 용량 예약가 모두 사용되었음을 보여 주는 다음 응답이 나타납니다.

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "m5.large",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "m4.xlarge",
    "AvailableInstanceCount": 0
}
```

```
{
    "CapacityReservationId": "cr-333",
    "InstanceType": "m4.2xlarge",
    "AvailableInstanceCount": 0
}
```

Linux 인스턴스에 연결

시작한 Linux 인스턴스에 연결하여 로컬 컴퓨터와 인스턴스 간에 파일을 전송합니다.

Windows 인스턴스에 연결해야 하는 경우 Windows 인스턴스용 Amazon EC2 사용 설명서의 Windows 인스턴스에 연결을 참조하십시오.

연결 옵션

로컬 컴퓨터의 운영 체제에 따라 로컬 컴퓨터에서 Linux 인스턴스로 연결하는 데 필요한 옵션이 결정됩니다.

Linux 및 macOS X에 대한 옵션

- SSH 클라이언트 (p. 487)
- EC2 Instance Connect (p. 490)
- AWS 시스템 관리자 Session Manager

Windows에 대한 옵션

- PuTTY (p. 498)
- SSH 클라이언트 (p. 487)
- AWS 시스템 관리자 Session Manager
- Windows Subsystem for Linux (p. 503)

Amazon EC2 콘솔은 Java SSH 클라이언트를 사용하여 브라우저에서 직접 인스턴스에 연결하는 옵션을 제공합니다. 하지만 이 옵션을 더 이상 지원하지 않는 브라우저가 많습니다. 자세한 내용은 내 브라우저를 사용하여 연결할 수 없음 (p. 1091) 단원을 참조하십시오.

인스턴스에 연결하기 위한 일반 사전 조건

Linux 인스턴스에 연결하려면 먼저 다음 일반 사전 조건을 확인하십시오.

- 인스턴스에 대한 정보 가져오기 (p. 485)
- 인스턴스로의 인바운드 트래픽 활성화 (p. 486)
- 프라이빗 키 찾기 (p. 486)
- (선택 사항) 인스턴스 지문 가져오기 (p. 486)

인스턴스에 대한 정보 가져오기

- 인스턴스의 ID 보기.

  Amazon EC2 콘솔을 사용하여(인스턴스 ID 열에서) 인스턴스의 ID를 가져올 수 있습니다. 원하는 경우 describe-instances(AWS CLI) 또는 Get-EC2Instance(Windows PowerShell용 AWS 도구) 명령을 사용할 수 있습니다.
- 인스턴스의 퍼블릭 DNS 이름을 가져옵니다.

  Amazon EC2 콘솔을 사용하여 인스턴스의 퍼블릭 DNS를 가져올 수 있습니다. 퍼블릭 DNS(IPv4) 열을 확인합니다. 이 열이 숨겨져 있으면 표시/숨기기 아이콘을 선택하고 퍼블릭 DNS(IPv4)를 선택합니다. 원하는 경우 describe-instances(AWS CLI) 또는 Get-EC2Instance(Windows PowerShell용 AWS 도구) 명령을 사용할 수 있습니다.

- (IPv6 전용) 인스턴스의 IPv6 주소를 가져옵니다.

  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 가져올 수 있습니다. IPv6 IP 필드를 확인합니다. 원하는 경우 describe-instances(AWS CLI) 또는 Get-EC2Instance(Windows PowerShell용 AWS 도구) 명령을 사용할 수 있습니다. IPv6에 대한 자세한 내용은 IPv6 주소 (p. 660) 단원을 참조하십시오.
- 인스턴스를 시작하는 데 사용한 AMI의 기본 사용자 이름을 가져옵니다.
  - Amazon Linux 2 또는 Amazon Linux AMI의 경우 사용자 이름은 `ec2-user`입니다.
  - CentOS AMI의 경우 사용자 이름은 `centos`입니다.
  - Debian AMI의 경우 사용자 이름은 `admin` 또는 `root`입니다.
  - Fedora AMI의 경우 사용자 이름은 `ec2-user` 또는 `fedora`입니다.
  - RHEL AMI의 경우 사용자 이름은 `ec2-user` 또는 `root`입니다.
  - SUSE AMI의 경우 사용자 이름은 `ec2-user` 또는 `root`입니다.
  - Ubuntu AMI의 경우 사용자 이름은 `ubuntu`입니다.
  - `ec2-user` 및 `root`를 사용할 수 없는 경우 AMI 공급자에게 문의하십시오.

인스턴스로의 인바운드 트래픽 활성화

- IP 주소에서 인스턴스로 인바운드 SSH 트래픽을 활성화합니다.

  인스턴스와 연관된 보안 그룹이 IP 주소로부터 들어오는 SSH 트래픽을 허용하는지 확인하십시오. VPC의 기본 보안 그룹은 기본적으로 수신 SSH 트래픽을 허용하지 않습니다. 시작 인스턴스 마법사가 생성한 보안 그룹은 기본적으로 SSH 트래픽을 활성화합니다. 자세한 내용은 Linux 인스턴스의 인바운드 트래픽 권한 부여 (p. 859) 단원을 참조하십시오.

프라이빗 키 찾기

- 프라이빗 키 찾기 및 권한 확인

  인스턴스를 시작할 때 지정한 키 페어를 찾기 위해 `.pem` 파일의 컴퓨터 상 위치에 대한 정규화된 경로를 얻습니다. 키 페어를 생성하는 방법에 대한 자세한 내용은 Amazon EC2를 사용하여 키 페어 생성을 참조하십시오.

  `.pem` 파일에 0777이 아닌 0400 권한이 있는지 확인하십시오. 자세한 내용은 오류: 보호되지 않는 프라이빗 키 파일 (p. 1090) 단원을 참조하십시오.

프라이빗 키의 권한을 설정하려면

1. 명령줄 셸에서 디렉터리를 인스턴스를 시작할 때 만든 프라이빗 키 파일의 위치로 변경합니다.
2. 다음 명령을 사용하여 사용자만 읽을 수 있도록 프라이빗 키 파일의 권한을 설정합니다.

   ```
   chmod 400 /path/my-key-pair.pem
   ```

   이러한 권한을 설정하지 않으면 이 키 페어를 사용하여 인스턴스에 연결할 수 없습니다. 자세한 내용은 오류: 보호되지 않는 프라이빗 키 파일 (p. 1090)을 참조하십시오.

(선택 사항) 인스턴스 지문 가져오기

메시지 가로채기(man-in-the-middle) 공격으로부터 보호하기 위해 인스턴스에 연결할 때 RSA 키 지문을 확인할 수 있습니다. 지문 확인 기능은 타사의 퍼블릭 AMI에서 인스턴스를 시작한 경우에 유용합니다.

먼저 인스턴스 지문을 가져옵니다. 그러면 인스턴스에 연결할 때 지문 확인 메시지가 표시됩니다. 얻은 지문과 확인용으로 표시된 지문을 비교할 수 있습니다. 이들 지문이 일치하지 않으면 누군가가 "메시지 가로채기(man-in-the-middle)" 공격을 시도하고 있는 것일 수 있습니다. 이 두 지문이 일치하면 확실하게 인스턴스에 연결할 수 있습니다.

인스턴스 지문 가져오기를 위한 사전 조건:

- 인스턴스 지문을 가져오려면 AWS CLI를 사용해야 합니다. AWS CLI 설치에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 AWS 명령줄 인터페이스 설치를 참조하십시오.
- 인스턴스는 `pending` 상태가 아닌 `running` 상태여야 합니다.

인스턴스 지문을 가져오려면

1. 인스턴스가 아닌 로컬 시스템에서 다음과 같이 get-console-output(AWS CLI) 명령을 사용하여 지문을 가져옵니다.

   ```
   $ aws ec2 get-console-output --instance-id instance_id
   ```

   다음은 살펴봐야 할 예제입니다.

   ```
   -----BEGIN SSH HOST KEY FINGERPRINTS -----
   ... 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f ...
   -----END SSH HOST KEY FINGERPRINTS-----
   ```

2. 생성된 출력에서 `SSH HOST KEY FINGERPRINTS` 섹션을 찾아서 RSA 지문을 적어 둡니다(예: 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f). `SSH HOST KEY FINGERPRINTS` 섹션은 인스턴스를 처음 부팅한 후에만 사용할 수 있습니다.

SSH를 사용하여 Linux 인스턴스에 연결

인스턴스를 시작한 후 인스턴스에 연결하고 바로 앞에 있는 컴퓨터를 사용하는 것처럼 인스턴스를 사용할 수 있습니다.

다음 지침에서는 SSH 클라이언트를 사용하여 인스턴스에 연결하는 방법을 설명합니다. 인스턴스에 연결을 시도하는 동안 오류가 발생한 경우 인스턴스 연결 문제 해결 (p. 1085) 단원을 참조하십시오.

사전 조건

Linux 인스턴스에 연결하려면 먼저 다음 사전 조건을 완료하십시오.

인스턴스가 준비되었는지 확인

인스턴스를 시작한 후, 연결할 수 있도록 인스턴스가 준비될 때까지 몇 분 정도 걸릴 수 있습니다. 인스턴스가 상태 확인을 통과했는지 확인합니다. 인스턴스 페이지의 상태 확인 열에서 이 정보를 볼 수 있습니다.

인스턴스에 연결하기 위한 일반 사전 조건 확인

자세한 내용은 인스턴스에 연결하기 위한 일반 사전 조건 (p. 485) 단원을 참조하십시오.

필요에 따라 로컬 컴퓨터에 SSH 클라이언트 설치

로컬 컴퓨터에 기본적으로 SSH 클라이언트가 설치되어 있을 수 있습니다. 명령줄에 ssh를 입력하여 이 상태를 확인할 수 있습니다. 컴퓨터에서 명령이 인식되지 않는 경우 SSH 클라이언트를 설치할 수 있습니다.

- Windows Server 2019 및 Windows 10의 최신 버전 - OpenSSH는 설치 가능한 구성 요소로 포함되어 있습니다. 자세한 내용은 Windows의 OpenSSH를 참조하십시오.

- Windows의 이전 버전 - OpenSSH를 다운로드하여 설치합니다. 자세한 내용은 Win32-OpenSSH를 참조하십시오.
- Linux 및 macOS X - OpenSSH를 다운로드하여 설치합니다. 자세한 내용은 http://www.openssh.com을 참조하십시오.

SSH 클라이언트를 사용하여 Linux 인스턴스에 연결

SSH 클라이언트를 사용하여 Linux 인스턴스에 연결하려면 다음 프로시저를 사용하십시오. 인스턴스에 연결을 시도하는 동안 오류가 발생한 경우 인스턴스 연결 문제 해결 (p. 1085) 단원을 참조하십시오.

SSH를 사용하여 인스턴스에 연결하려면

1. 터미널 창에서 ssh 명령을 사용하여 인스턴스에 연결합니다. 프라이빗 키(`.pem`)의 경로와 파일 이름, AMI의 사용자 이름 및 인스턴스의 퍼블릭 DNS 이름 또는 IPv6 주소를 지정합니다. 프라이빗 키, AMI의 사용자 이름, 인스턴스의 DNS 이름 또는 IPv6 주소를 확인하는 방법에 대한 자세한 내용은 프라이빗 키 찾기 (p. 486) 및 인스턴스에 대한 정보 가져오기 (p. 485) 단원을 참조하십시오. 인스턴스에 연결하려면 다음 중 하나를 수행합니다.
   - (퍼블릭 DNS) 인스턴스의 퍼블릭 DNS를 사용하여 연결하려면 다음 명령을 입력합니다.

     ```
     ssh -i /path/my-key-pair.pem ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com
     ```

   - (IPv6) 또는 인스턴스에 IPv6 주소가 있는 경우 인스턴스의 IPv6 주소를 사용하여 연결하려면 다음 명령을 입력합니다.

     ```
     ssh -i /path/my-key-pair.pem ec2-user@2001:db8:1234:1a00:9691:9503:25ad:1761
     ```

   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)?
   ```

2. (선택 사항) 보안 알림의 지문이 앞의 (선택 사항) 인스턴스 지문 가져오기 (p. 486)에서 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.
3. `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.
   ```

SCP를 사용하여 Linux에서 Linux 인스턴스로 파일 전송

로컬 컴퓨터와 Linux 인스턴스 간에 파일을 전송하는 한 가지 방법은 SCP(Secure Copy Protocol)를 사용하는 것입니다. 이 섹션에서는 SCP를 사용하여 파일을 전송하는 방법을 설명합니다. 이 절차는 SSH를 사용하여 인스턴스에 연결하는 절차와 비슷합니다.

사전 조건

- 파일을 인스턴스에 전송하기 위한 일반 사전 조건을 확인합니다.

  파일을 인스턴스에 전송하기 위한 일반 사전 조건은 인스턴스에 연결하기 위한 일반 사전 조건과 동일합니다. 자세한 내용은 인스턴스에 연결하기 위한 일반 사전 조건 (p. 485)을 참조하십시오.
- 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.org를 참조하십시오.

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. 인스턴스의 퍼블릭 DNS 이름을 사용하여 인스턴스로 파일을 전송합니다. 예를 들어, 프라이빗 키 파일의 이름이 `my-key-pair`이고, 전송할 파일이 `SampleFile.txt`이며, 사용자 이름이 `ec2-user`이고, 인스턴스의 퍼블릭 DNS 이름이 `ec2-198-51-100-1.compute-1.amazonaws.com`인 경우, 다음 명령을 사용하여 파일을 `ec2-user` 홈 디렉터리로 복사합니다.

```
scp -i /path/my-key-pair.pem /path/SampleFile.txt 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)?
```

2. (IPv6 전용) 또는 인스턴스의 IPv6 주소를 이용해 파일을 전송할 수도 있습니다. IPv6 주소는 이스케이프된(\) 대괄호([])로 묶어야 합니다.

```
scp -i /path/my-key-pair.pem /path/SampleFile.txt ec2-user@
\[2001:db8:1234:1a00:9691:9503:25ad:1761\]:~
```

3. (선택 사항) 보안 알림의 지문이 앞의 (선택 사항) 인스턴스 지문 가져오기 (p. 486)에서 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.
4. **`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
```

   "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
```

5. 반대 방향으로(Amazon EC2 인스턴스에서 로컬 컴퓨터로) 파일을 전송하려면 호스트 파라미터의 순서를 역순으로 지정하면 됩니다. 예를 들어, EC 인스턴스의 `SampleFile.txt` 파일을 로컬 컴퓨터의 홈 디렉터리에 `SampleFile2.txt`2.로 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
```

6. (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
```

EC2 Instance Connect를 사용하여 Linux 인스턴스에 연결

Amazon EC2 Instance Connect는 Secure Shell(SSH)을 사용하여 인스턴스에 연결하는 간단하고 안전한 방법입니다. EC2 Instance Connect를 사용하면 AWS Identity and Access Management(IAM) 정책 및 보안 주체를 사용하여 인스턴스에 대한 SSH 액세스를 제어하고 SSH 키를 공유 및 관리할 필요가 없습니다. EC2 Instance Connect를 사용하는 모든 연결 요청은 AWS CloudTrail에 로깅되므로 연결 요청을 감사할 수 있습니다 (p. 656).

Instance Connect를 이용하면 브라우저 기반 클라이언트, Amazon EC2 Instance Connect CLI 또는 사용자가 선택한 SSH 클라이언트를 사용하는 Linux 인스턴스에 연결할 수 있습니다.

EC2 Instance Connect을 사용하여 인스턴스에 연결할 때 Instance Connect API는 일회용 SSH 퍼블릭 키를 인스턴스 메타데이터 (p. 571)에 푸시하여 60초 동안 유지합니다. IAM 사용자에게 연결된 IAM 정책은 퍼블릭 키를 인스턴스 메타데이터로 푸시하도록 IAM 사용자에게 권한을 부여합니다. SSH 데몬은 Instance Connect가 설치될 때 구성된 `AuthorizedKeysCommand` 및 `AuthorizedKeysCommandUser`를 사용하여 인증을 위해 인스턴스 메타데이터에서 퍼블릭 키를 찾고 사용자를 인스턴스에 연결합니다.

> **Note**
>
> Windows를 실행하는 로컬 컴퓨터에서 Linux 인스턴스를 연결하는 경우 다음 설명서를 대신 참조하십시오. PuTTY를 사용하여 Windows에서 Linux 인스턴스에 연결 (p. 498) 및 Windows Subsystem for Linux를 사용하여 Windows에서 Linux 인스턴스에 연결 (p. 503).

목차

- EC2 Instance Connect 설정 (p. 490)
- EC2 Instance Connect를 사용하여 연결 (p. 495)
- EC2 Instance Connect 제거 (p. 497)

EC2 Instance Connect 설정

Amazon Linux 2 2.0.20190618 이상은 EC2 Instance Connect로 사전 구성되어 제공됩니다. 그 외 지원되는 Linux 배포판의 경우, Instance Connect 사용을 지원할 모든 인스턴스에 대해 Instance Connect를 설정해야 합니다. 각 인스턴스에 대해 한 번만 수행하면 됩니다.

Instance Connect를 설정하기 위한 작업

- 1단계: 인스턴스에 대한 네트워크 액세스 구성 (p. 491)
- 2단계: 인스턴스에 EC2 Instance Connect 설치 (p. 491)
- 3단계: (선택 사항) EC2 Instance Connect CLI 설치 (p. 493)
- 4단계: EC2 Instance Connect에 대한 IAM 권한 구성 (p. 494)

제한 사항

- 다음 Linux 배포판이 지원됩니다.
  - Amazon Linux 2(모든 버전)
  - Ubuntu 16.04 이상
- SSH 인증을 위한 `AuthorizedKeysCommand` 및 `AuthorizedKeysCommandUser` 설정을 구성했다면 EC2 Instance Connect 설치 시 이들 항목이 업데이트가 되지 않습니다. 따라서 Instance Connect를 사용할 수 없습니다.

사전 조건

- SSH를 사용하여 인스턴스에 연결하기 위한 일반 사전 조건을 확인합니다.

  자세한 내용은 인스턴스에 연결하기 위한 일반 사전 조건 (p. 485) 단원을 참조하십시오.
- 로컬 컴퓨터에 SSH 클라이언트를 설치합니다.

  로컬 컴퓨터에는 대개 기본적으로 SSH 클라이언트가 설치되어 있습니다. 명령줄에 ssh를 입력하여 SSH 클라이언트가 있는지 확인할 수 있습니다. 로컬 컴퓨터가 명령을 인식하지 않는 경우 SSH 클라이언트를 설치할 수 있습니다. Linux 또는 macOS에 SSH 클라이언트를 설치하는 방법은 http://www.openssh.com을 참조하십시오. Windows 10에 SSH 클라이언트를 설치하는 방법은 OpenSSH in Windows를 참조하십시오.
- 로컬 컴퓨터에 AWS CLI를 설치합니다.

  IAM 권한을 구성하려면 AWS CLI를 사용해야 합니다. AWS CLI 설치에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 AWS CLI 설치를 참조하십시오.
- [Ubuntu] 인스턴스에 AWS CLI를 설치합니다.

  Ubuntu 인스턴스에 EC2 Instance Connect를 설치하려면 해당 인스턴스에서 AWS CLI를 사용해야 합니다. AWS CLI 설치에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 AWS CLI 설치를 참조하십시오.

1단계: 인스턴스에 대한 네트워크 액세스 구성

EC2 Instance Connect을 설치하고 사용자가 인스턴스에 연결할 수 있도록 인스턴스에 대해 다음과 같은 네트워크 액세스를 구성해야 합니다.

- 인스턴스와 연관된 보안 그룹이 IP 주소에서 포트 22를 통한 인바운드 SSH 트래픽을 허용 (p. 859)하는지 확인하십시오. VPC의 기본 보안 그룹은 기본적으로 수신 SSH 트래픽을 허용하지 않습니다. 시작 마법사가 생성한 보안 그룹은 기본적으로 수신 SSH 트래픽을 허용합니다. 자세한 내용은 Linux 인스턴스의 인바운드 트래픽 권한 부여 (p. 859) 단원을 참조하십시오.
- (브라우저 기반 클라이언트) 인스턴스는 서비스용으로 게시된 권장 IP 블록에서 인바운드 SSH 트래픽을 허용하는 것이 좋습니다. `service` 파라미터에 대한 `EC2_INSTANCE_CONNECT` 필터를 사용하여 EC2 Instance Connect 하위 집합의 IP 주소 범위를 가져옵니다. 자세한 내용은 Amazon Web Services 일반 참조의 AWS IP 주소 범위를 참조하십시오.

2단계: 인스턴스에 EC2 Instance Connect 설치

EC2 Instance Connect를 설치하면 인스턴스에 SSH 데몬을 구성합니다. EC2 Instance Connect 설치 절차는 Amazon Linux 2 및 Ubuntu를 사용하여 시작된 인스턴스에 따라 다릅니다.

Amazon Linux 2

EC2 Instance Connect로 시작된 인스턴스에 Amazon Linux 2를 설치하려면

1. SSH를 사용하여 인스턴스에 연결합니다.

   인스턴스를 시작할 때 인스턴스에 할당된 SSH 키 페어 및 인스턴스를 시작하는 데 사용한 AMI의 기본 사용자 이름을 사용합니다. Amazon Linux 2의 경우, 기본 사용자 이름은 `ec2-user`입니다.

   예를 들어 인스턴스가 Amazon Linux 2를 사용하여 시작되었고 인스턴스의 퍼블릭 DNS가 `ec2-a-b-c-d.us-west-2.compute.amazonaws.com`이고 키 페어가 `my_ec2_private_key.pem`인 경우 다음 명령을 사용하여 인스턴스에 SSH를 추가합니다.

```
$ ssh -i my_ec2_private_key.pem ec2-user@ec2-a-b-c-d.us-
west-2.compute.amazonaws.com
```

인스턴스 연결에 대한 자세한 내용은 SSH를 사용하여 Linux 인스턴스에 연결 (p. 487) 단원을 참조하십시오.

2. 인스턴스에 EC2 Instance Connect 패키지를 설치합니다.

   Amazon Linux 2에서 yum install 명령을 사용합니다.

```
[ec2-user ~]$ sudo yum install ec2-instance-connect
```

   `/opt/aws/bin/` 폴더에 4개의 새 파일이 표시됩니다.

```
eic_curl_authorized_keys
eic_harvest_hostkeys
eic_parse_authorized_keys
eic_run_authorized_keys
```

3. (선택 사항) Instance Connect가 인스턴스에 성공적으로 설치되었는지 확인합니다.

   sudo less 명령을 사용하여 `/etc/ssh/sshd_config` 파일이 다음과 같이 올바르게 업데이트되었는지 확인합니다.

```
[ec2-user ~]$ sudo less /etc/ssh/sshd_config
```

   `/etc/ssh/sshd_config` 파일의 `AuthorizedKeysCommand` 및 `AuthorizedKeysCommandUser` 행에 다음 값이 포함되어 있으면 Instance Connect가 성공적으로 설치된 것입니다.

```
AuthorizedKeysCommand /opt/aws/bin/eic_run_authorized_keys %u %f
AuthorizedKeysCommandUser ec2-instance-connect
```

   - `AuthorizedKeysCommand`는 `eic_run_authorized_keys` 파일을 설정하여 인스턴스 메타데이터에서 키를 찾습니다
   - `AuthorizedKeysCommandUser`는 시스템 사용자를 `ec2-instance-connect`로 설정합니다.

   > **Note**
   >
   > 이전에 `AuthorizedKeysCommand` 및 `AuthorizedKeysCommandUser`를 구성한 경우 Instance Connect 설치가 값을 변경하지 않으며 Instance Connect를 사용할 수 없습니다.

Ubuntu

Ubuntu 16.04 이상으로 시작된 인스턴스에 EC2 Instance Connect를 설치하려면

1. SSH를 사용하여 인스턴스에 연결합니다.

   인스턴스를 시작할 때 인스턴스에 할당된 SSH 키 페어를 사용하고 인스턴스를 시작하는 데 사용한 AMI의 기본 사용자 이름을 사용합니다. Ubuntu AMI의 경우 사용자 이름은 `ubuntu`입니다.

   인스턴스가 Ubuntu를 사용하여 시작되었고 인스턴스의 퍼블릭 DNS가 `ec2-a-b-c-d.us-west-2.compute.amazonaws.com`이고 키 페어가 `my_ec2_private_key.pem`인 경우 다음 명령을 사용하여 인스턴스에 SSH를 추가합니다.

```
$ ssh -i my_ec2_private_key.pem ubuntu@ec2-a-b-c-d.us-west-2.compute.amazonaws.com
```

인스턴스 연결에 대한 자세한 내용은 SSH를 사용하여 Linux 인스턴스에 연결 (p. 487)단원을 참조하십시오.

2. (선택 사항) 인스턴스에 최신 Ubuntu AMI가 있는지 확인합니다.

   Ubuntu의 경우 apt-get update 명령을 사용하여 인스턴스의 모든 패키지를 업데이트합니다.

```
ubuntu:~$ sudo apt-get update
```

3. 인스턴스에 Instance Connect 패키지를 설치합니다.

   Ubuntu의 경우 sudo apt-get 명령을 사용하여 `.deb` 패키지를 설치할 수 있습니다.

```
ubuntu:~$ sudo apt-get install ec2-instance-connect
```

   `/usr/share/ec2-instance-connect/` 폴더에 4개의 새 파일이 표시됩니다.

```
eic_curl_authorized_keys
eic_harvest_hostkeys
eic_parse_authorized_keys
eic_run_authorized_keys
```

4. (선택 사항) Instance Connect가 인스턴스에 성공적으로 설치되었는지 확인합니다.

   sudo less 명령을 사용하여 `/lib/systemd/system/ssh.service.d/ec2-instance-connect.conf`가 다음과 같이 올바르게 업데이트되었는지 확인합니다.

```
ubuntu:~$ sudo less /lib/systemd/system/ssh.service.d/ec2-instance-connect.conf
```

   `/lib/systemd/system/ssh.service.d/ec2-instance-connect.conf` 파일의 `AuthorizedKeysCommand` 및 `AuthorizedKeysCommandUser` 행에 다음 값이 포함되어 있으면 Instance Connect가 성공적으로 설치된 것입니다.

```
AuthorizedKeysCommand /usr/share/ec2-instance-connect/eic_run_authorized_keys %u %f
AuthorizedKeysCommandUser ec2-instance-connect
```

   - `AuthorizedKeysCommand`는 `eic_run_authorized_keys` 파일을 설정하여 인스턴스 메타데이터에서 키를 찾습니다
   - `AuthorizedKeysCommandUser`는 시스템 사용자를 `ec2-instance-connect`로 설정합니다.

   > **Note**
   >
   > 이전에 `AuthorizedKeysCommand` 및 `AuthorizedKeysCommandUser`를 구성한 경우 Instance Connect 설치가 값을 변경하지 않으며 Instance Connect를 사용할 수 없습니다.

EC2 Instance Connect 패키지에 대한 자세한 내용은 GitHub 웹 사이트의 aws/aws-ec2-instance-connect-config를 참조하십시오.

3단계: (선택 사항) EC2 Instance Connect CLI 설치

EC2 Instance Connect CLI는 EC2 인스턴스 정보 쿼리, 임시 퍼블릭 키 생성 및 게시, 단일 명령 `mssh` *`instance_id`*를 통한 SSH 연결 설정 등 표준 SSH 호출과 유사한 인터페이스를 제공합니다.

Note

사용자가 브라우저 기반 클라이언트나 SSH 클라이언트만 사용하여 인스턴스에 연결하는 경우, EC2 Instance Connect CLI를 설치할 필요가 없습니다.

EC2 Instance Connect CLI 패키지를 설치하려면

`pip`를 사용하여 `ec2instanceconnectcli` 패키지를 설치합니다. 자세한 내용은 GitHub 웹 사이트의 aws/aws-ec2-instance-connect-cli와 Python Package Index(PyPI) 웹 사이트의 https://pypi.org/project/ec2instanceconnectcli/를 참조하십시오.

```
$ pip install ec2instanceconnectcli
```

4단계: EC2 Instance Connect에 대한 IAM 권한 구성

IAM 사용자가 EC2 Instance Connect를 사용하여 인스턴스에 연결할 경우 퍼블릭 키를 인스턴스에 푸시하도록 권한을 부여해야 합니다. 자세한 내용은 IAM 사용 설명서의 Amazon EC2 Instance Connect에 사용되는 작업, 리소스 및 조건 키를 참조하십시오.

다음 지침에서는 정책을 생성하고 AWS CLI를 사용하여 정책을 연결하는 방법을 설명합니다. AWS Management 콘솔을 사용하는 지침은 IAM 사용 설명서의 IAM 정책 만들기(콘솔) 및 사용자에게 직접 정책을 연결하여 권한을 추가를 참조하십시오.

제한

현재 Instance Connect에 대한 태그 기반 권한을 지원하지 않습니다.

EC2 Instance Connect의 IAM 사용자 권한을 부여하려면(AWS CLI)

1. 다음 사항이 포함된 JSON 정책 문서를 생성합니다.

   - `ec2-instance-connect:SendSSHPublicKey` 작업 이 작업은 퍼블릭 키를 인스턴스에 푸시할 수 있는 권한을 IAM 사용자에게 부여합니다. `ec2-instance-connect:SendSSHPublicKey`를 사용해 특정 EC2 인스턴스에 대한 액세스 제한을 고려하십시오. 그렇지 않을 경우, 이 권한이 있는 모든 IAM 사용자가 모든 EC2 인스턴스에 연결할 수 있습니다.
   - `ec2:osuser` 조건. 퍼블릭 키를 인스턴스로 푸시할 수 있는 OS 사용자의 이름을 지정합니다. 인스턴스를 시작하는 데 사용한 AMI의 기본 사용자 이름을 사용합니다. 예를 들어 Amazon Linux 2의 기본 사용자 이름은 Ubuntu에 경우 `ec2-user` 및 `ubuntu`입니다.
   - `ec2:DescribeInstances` 작업 이 작업은 래퍼가 이 작업을 호출하기 때문에 EC2 Instance Connect CLI를 사용할 때 필요합니다. IAM 사용자는 이미 다른 정책에서 이 작업을 호출할 권한을 보유할 수도 있습니다.

   다음은 정책 문서의 예입니다. 사용자가 SSH 클라이언트만을 사용하여 인스턴스에 연결하는 경우 `ec2:DescribeInstances` 작업의 설명문을 생략할 수 있습니다. EC2 Instance Connect를 사용하는 모든 EC2에 대한 사용자 액세스 권한을 부여하도록 `Resource`에서 지정된 인스턴스를 대체할 수 있습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": "ec2-instance-connect:SendSSHPublicKey",
        "Resource": [
            "arn:aws:ec2:region:account-id:instance/i-1234567890abcdef0",
            "arn:aws:ec2:region:account-id:instance/i-0598c7d356eba48d7"
        ],
        "Condition": {
```

```
            "StringEquals": {
                "ec2:osuser": "ami-username"
            }
        }
      },
      {
        "Effect": "Allow",
        "Action": "ec2:DescribeInstances",
        "Resource": "*"
      }
    ]
}
```

2. create-policy 명령을 사용하여 새 관리형 정책을 생성하고 사용자가 생성한 JSON 문서를 지정하여 새 정책의 콘텐츠로 사용합니다.

```
$ aws iam create-policy --policy-name my-policy --policy-document file://JSON-file-name
```

3. attach-user-policy 명령을 사용하여 지정된 IAM 사용자에게 관리형 정책을 연결합니다. `--user-name` 파라미터의 경우 IAM 사용자의 알기 쉬운 이름(ARN 아님)을 지정합니다.

```
$ aws iam attach-user-policy --policy-arn arn:aws:iam::account-id:policy/my-policy --
user-name IAM-friendly-name
```

EC2 Instance Connect를 사용하여 연결

다음 지침에서는 EC2 Instance Connect를 사용하여 Linux 인스턴스에 연결하는 방법을 설명합니다.

Instance Connect를 사용하여 연결하기 위한 옵션

- 브라우저 기반 클라이언트를 사용하여 연결 (p. 496)
- EC2 Instance Connect CLI를 사용하여 연결 (p. 496)
- 자체 키 및 SSH 클라이언트를 사용하여 연결 (p. 496)

제한 사항

- 다음 Linux 배포판이 지원됩니다.
  - Amazon Linux 2(모든 버전)
  - Ubuntu 16.04 이상
- 브라우저 기반 클라이언트를 사용하여 연결하려면 인스턴스에 퍼블릭 IPv4 주소가 있어야 합니다.
- EC2 Instance Connect CLI를 사용하여 연결하려면 프라이빗 IP 주소를 사용할 수 있으므로 인스턴스에 퍼블릭 IPv4 주소가 필요하지 않습니다. 인스턴스에 퍼블릭 IP 주소와 프라이빗 IP 주소가 모두 있는 경우 API는 먼저 퍼블릭 IP 주소를 사용하여 연결을 시도합니다.
- EC2 Instance Connect에서는 IPv6 주소를 사용한 연결을 지원하지 않습니다.
- Safari 브라우저는 현재 지원되지 않습니다.

사전 조건

- 인스턴스에 Instance Connect를 설치합니다.

  자세한 내용은 EC2 Instance Connect 설정 (p. 490) 단원을 참조하십시오.
- (선택 사항) 로컬 컴퓨터에 SSH 클라이언트를 설치합니다.

  사용자가 콘솔이나 EC2 Instance Connect CLI만 사용하여 인스턴스에 연결하는 경우 SSH 클라이언트를 설치할 필요가 없습니다. 로컬 컴퓨터에는 대개 기본적으로 SSH 클라이언트가 설치되어 있습니다. 명령

줄에 ssh를 입력하여 SSH 클라이언트가 있는지 확인할 수 있습니다. 로컬 컴퓨터가 명령을 인식하지 않는 경우 SSH 클라이언트를 설치할 수 있습니다. Linux 또는 macOS에 SSH 클라이언트를 설치하는 방법은 http://www.openssh.com을 참조하십시오. Windows 10에 SSH 클라이언트를 설치하는 방법은 OpenSSH in Windows를 참조하십시오.

- (선택 사항) 로컬 컴퓨터에 EC2 Instance Connect CLI를 설치합니다.

  사용자가 콘솔이나 SSH 클라이언트만 사용하여 인스턴스에 연결하는 경우 EC2 Instance Connect CLI를 설치할 필요가 없습니다. 자세한 내용은 3단계: (선택 사항) EC2 Instance Connect CLI 설치 (p. 493) 단원을 참조하십시오.

브라우저 기반 클라이언트를 사용하여 연결

Amazon EC2 콘솔에서 인스턴스를 선택하고 EC2 Instance Connect를 사용하여 연결하면 브라우저 기반 클라이언트를 사용하는 인스턴스에 연결할 수 있습니다. Instance Connect는 권한을 처리하고 성공적인 연결을 제공합니다.

Amazon EC2 콘솔에서 브라우저 기반 클라이언트를 사용하여 인스턴스에 연결하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택한 다음 연결을 선택합니다.
4. EC2 Instance Connect(브라우저 기반 SSH 연결), 연결을 선택합니다.

창이 열리고 인스턴스에 연결됩니다.

EC2 Instance Connect CLI를 사용하여 연결

Instance Connect CLI가 한 번의 호출로 다음 세 가지 작업을 수행하는 동안 인스턴스 ID 만 제공하여 EC2 Instance Connect CLI를 통해 인스턴스에 연결할 수 있습니다. 즉, 일회용 SSH 퍼블릭 키를 생성하고, 키를 60초 동안 유지되는 인스턴스로 푸시하고 사용자를 인스턴스에 연결하는 작업입니다. Instance Connect CLI에서 기본 SSH/SFTP 명령을 사용할 수 있습니다.

Amazon Linux 2

EC2 Instance Connect CLI를 사용하여 인스턴스에 연결하려면

다음과 같이 mssh 명령을 인스턴스 ID와 함께 사용합니다. AMI의 사용자 이름을 지정할 필요가 없습니다.

```
$ mssh i-001234a4bf70dec41EXAMPLE
```

Ubuntu

EC2 Instance Connect CLI를 사용하여 인스턴스에 연결하려면

다음과 같이 인스턴스 ID 및 Ubuntu AMI의 기본 사용자 이름과 함께 mssh 명령을 사용하십시오. AMI의 사용자 이름을 지정해야 합니다. 그렇지 않으면 인증 실패오류가 발생됩니다.

```
$ mssh ubuntu@i-001234a4bf70dec41EXAMPLE
```

자체 키 및 SSH 클라이언트를 사용하여 연결

자체 SSH 키를 사용하고 한편으로 EC2 Instance Connect API를 사용하면서 선택한 SSH 클라이언트의 인스턴스에 연결할 수 있습니다. 이를 통해 Instance Connect 기능에서 퍼블릭 키를 인스턴스로 푸시할 수 있습니다.

요구 사항

지원되는 RSA 키 유형은 OpenSSH 및 SSH2입니다. 지원되는 길이는 2048 및 4096입니다. 자세한 내용은 Amazon EC2로 사용자의 퍼블릭 키 가져오기 (p. 863) 단원을 참조하십시오.

자체 키 및 SSH 클라이언트를 사용하여 인스턴스에 연결하려면

1. (선택 사항) 새로운 SSH 프라이빗 및 퍼블릭 키를 생성합니다.

   다음 명령을 사용하여 새로운 SSH 프라이빗 및 퍼블릭 키 `my_rsa_key` 및 `my_rsa_key.pub`을 생성할 수 있습니다.

   ```
   $ ssh-keygen -t rsa -f my_rsa_key
   ```

2. SSH 퍼블릭 키를 인스턴스에 푸시합니다.

   send-ssh-public-key 명령을 사용하여 SSH 퍼블릭 키를 인스턴스에 푸시합니다. Amazon Linux 2를 사용하여 인스턴스를 시작한 경우 AMI의 기본 사용자 이름은 `ec2-user`입니다. Ubuntu를 사용하여 인스턴스를 시작한 경우 AMI의 기본 사용자 이름은 `ubuntu`입니다.

   다음 예는 지정된 가용 영역에 있는 지정된 인스턴스에 퍼블릭 키를 푸시하여 `ec2-user`를 인증합니다.

   ```
   $ aws ec2-instance-connect send-ssh-public-key --instance-id i-001234a4bf70dec41EXAMPLE
    --availability-zone us-west-2b --instance-os-user ec2-user --ssh-public-key
    file://my_rsa_key.pub
   ```

3. 프라이빗 키를 사용하여 인스턴스에 연결합니다.

   ssh 명령을 사용하여 퍼블릭 키가 인스턴스 메타데이터에서 제거되기 전에 프라이빗 키를 사용하여 인스턴스에 연결합니다(제거되기 전에 60초가 주어짐). 퍼블릭 키에 해당하는 프라이빗 키, 인스턴스를 시작하는 데 사용한 AMI의 기본 사용자 이름 및 인스턴스의 퍼블릭 DNS를 지정합니다.

   ```
   $ ssh -i my_rsa_key ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com
   ```

EC2 Instance Connect 제거

EC2 Instance Connect을 비활성화하려면 인스턴스에 연결하여 OS에 설치된 `ec2-instance-connect` 패키지를 제거하십시오. `sshd` 구성이 EC2 Instance Connect 설치 시 적용한 설정과 일치할 경우, `ec2-instance-connect` 제거 시 `sshd` 구성도 제거됩니다. EC2 Instance Connect 설치 후 `sshd` 구성을 수정했다면 이 구성을 수동으로 업데이트해야 합니다.

Amazon Linux 2

EC2 Instance Connect이 사전 구성된 Amazon Linux 2 2.0.20190618 이상에서 EC2 Instance Connect를 제거할 수 있습니다.

Amazon Linux 2로 시작된 인스턴스에서 EC2 Instance Connect를 제거하려면

1. SSH를 사용하여 인스턴스에 연결합니다. `ec2-user`에 해당하는 Amazon Linux 2 AMI에 대해 기본 사용자 이름을 시작한 경우 인스턴스에 대해 사용한 SSH 키 페어를 지정합니다.

   예를 들어 다음 ssh 명령은 키 페어 `my_ec2_private_key.pem`를 사용하여 퍼블릭 DNS 이름 `ec2-a-b-c-d.us-west-2.compute.amazonaws.com`에 연결됩니다.

   ```
   $ ssh -i my_ec2_private_key.pem ec2-user@ec2-a-b-c-d.us-
   west-2.compute.amazonaws.com
   ```

2. yum 명령을 사용하여 `ec2-instance-connect` 패키지를 제거합니다.

```
[ec2-user ~]$ sudo yum remove ec2-instance-connect
```

Ubuntu

Ubuntu AMI로 시작된 인스턴스에서 EC2 Instance Connect를 제거하려면

1. SSH를 사용하여 인스턴스에 연결합니다. 인스턴스를 시작할 때 사용한 SSH 키 페어와 Ubuntu AMI의 기본 사용자 이름(`ubuntu`)을 지정합니다.

   예를 들어 다음 ssh 명령은 키 페어 `my_ec2_private_key.pem`를 사용하여 퍼블릭 DNS 이름 `ec2-a-b-c-d.us-west-2.compute.amazonaws.com`에 연결됩니다.

   ```
   $ ssh -i my_ec2_private_key.pem ubuntu@ec2-a-b-c-d.us-west-2.compute.amazonaws.com
   ```

2. apt-get 명령을 사용하여 `ec2-instance-connect` 패키지를 제거합니다.

   ```
   ubuntu:~$ sudo apt-get remove ec2-instance-connect
   ```

PuTTY를 사용하여 Windows에서 Linux 인스턴스에 연결

인스턴스를 시작한 후 인스턴스에 연결하고 바로 앞에 있는 컴퓨터를 사용하는 것처럼 인스턴스를 사용할 수 있습니다.

다음 지침에서는 Windows용 무료 SSH 클라이언트인 PuTTY를 사용하여 인스턴스에 연결하는 방법을 설명합니다. 인스턴스에 연결을 시도하는 동안 오류가 발생한 경우 인스턴스 연결 문제 해결을 참조하십시오.

사전 조건

PuTTY를 사용하여 Linux 인스턴스에 연결하려면 먼저 다음 사전 조건을 완료하십시오.

인스턴스가 준비되었는지 확인

인스턴스를 시작한 후, 연결할 수 있도록 인스턴스가 준비될 때까지 몇 분 정도 걸릴 수 있습니다. 인스턴스가 상태 확인을 통과했는지 확인합니다. 인스턴스 페이지의 상태 확인 열에서 이 정보를 볼 수 있습니다.

인스턴스에 연결하기 위한 일반 사전 조건 확인

자세한 내용은 인스턴스에 연결하기 위한 일반 사전 조건 (p. 485) 단원을 참조하십시오.

로컬 컴퓨터에 PuTTY 설치

PuTTY를 PuTTY 다운로드 페이지에서 다운로드하고 설치합니다. 이미 이전 버전의 PuTTY가 설치되어 있다면 최신 버전을 다운로드하는 것이 좋습니다. 전체 제품군을 설치해야 합니다.

PuTTYgen을 사용하여 프라이빗 키 변환

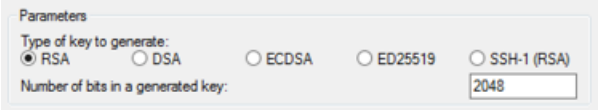
인스턴스를 시작할 때 지정한 키 페어에 대한 프라이빗 키(.pem 파일)를 찾습니다. PuTTY로 .pem 파일을 사용할 .ppk 파일로 변환합니다. 자세한 내용은 다음 단원의 단계를 참조하십시오.

PuTTYgen을 사용하여 프라이빗 키 변환

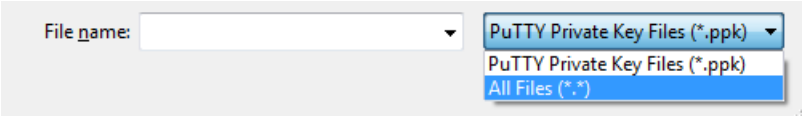
PuTTY에서는 SSH 키의 프라이빗 키 형식을 기본적으로 지원하지 않습니다. PuTTY에는 PuTTYgen이라는 도구가 있는데, 이 도구는 키를 필요한 PuTTY 형식으로 변환할 수 있습니다. PuTTY를 사용하여 인스턴스에 연결하려면 프라이빗 키(.pem 파일)를 이 형식(.ppk 파일)으로 변환해야 합니다.

키를 변환하려면

1. 시작 메뉴에서 모든 프로그램, PuTTY, PuTTYgen을 선택합니다.
2. Type of key to generate(생성할 키 유형)에서 RSA를 선택합니다. 이전 버전의 PuTTYgen을 사용하는 경우 SSH-2 RSA를 선택합니다.



3. 로드를 선택합니다. 기본적으로 PuTTYgen에는 확장명이 `.ppk`인 파일만 표시됩니다. `.pem` 파일을 찾으려면 모든 유형의 파일을 표시하는 옵션을 선택합니다.



4. 인스턴스를 시작할 때 지정한 키 페어의 `.pem` 파일을 선택하고 열기를 선택합니다. `.pem` 파일을 가져왔다는 알림이 PuTTYgen에 표시됩니다. 확인을 선택합니다.
5. Save private key(프라이빗 키 저장)을 선택하여 PuTTY에서 사용할 수 있는 형식으로 키를 저장합니다. PuTTYgen에서 암호 없이 키 저장에 대한 경고가 표시됩니다. 예를 선택합니다.

   Note

   프라이빗 키의 암호는 추가 보호 계층입니다. 프라이빗 키가 노출되더라도 암호 없이 사용할 수 없습니다. 암호문 사용의 단점은 인스턴스에 로그온하거나 인스턴스에 파일을 복사하기 위해 사용자가 개입해야 하기 때문에 자동화를 어렵게 만든다는 것입니다.
6. 키 페어에 사용한 것과 동일한 키 이름을 지정하고(예: `my-key-pair`) 저장을 선택합니다. PuTTY가 자동으로 `.ppk` 파일 확장자를 추가합니다.

이제 개인 키가 PuTTY에 사용하기에 올바른 형식으로 되어 있으므로 PuTTY의 SSH 클라이언트를 사용하여 인스턴스에 연결할 수 있습니다.

Linux 인스턴스에 연결

PuTTY를 사용하여 Linux 인스턴스에 연결하려면 다음 프로시저를 사용하십시오. 프라이빗 키에 대해 생성한 `.ppk` 파일이 필요합니다. 자세한 내용은 이전 단원의 PuTTYgen을 사용하여 프라이빗 키 변환 (p. 498)를 참조하십시오. 인스턴스에 연결을 시도하는 동안 오류가 발생한 경우 인스턴스 연결 문제 해결을 참조하십시오.

PuTTY를 사용하여 인스턴스에 연결하려면

1. PuTTY를 시작합니다. 즉, 시작 메뉴에서 모든 프로그램, PuTTY, PuTTY를 선택합니다.
2. 범주 창에서 세션을 선택하고 다음 필드를 작성합니다.

   a. 호스트 이름 상자에서 다음 중 하나를 수행합니다.
      - (퍼블릭 DNS) 인스턴스의 퍼블릭 DNS를 사용하여 연결하려면 *`user_name`*@*`public_dns_name`*을 입력합니다.
      - (IPv6) 또는 인스턴스에 IPv6 주소가 있는 경우 인스턴스의 IPv6 주소를 사용하여 연결하려면 *`user_name`*@*`ipv6_address`*를 입력합니다.

      인스턴스의 퍼블릭 DNS 이름 또는 IPv6 주소를 가져오는 방법에 대한 자세한 내용은 인스턴스에 대한 정보 가져오기 (p. 485) 단원을 참조하십시오.

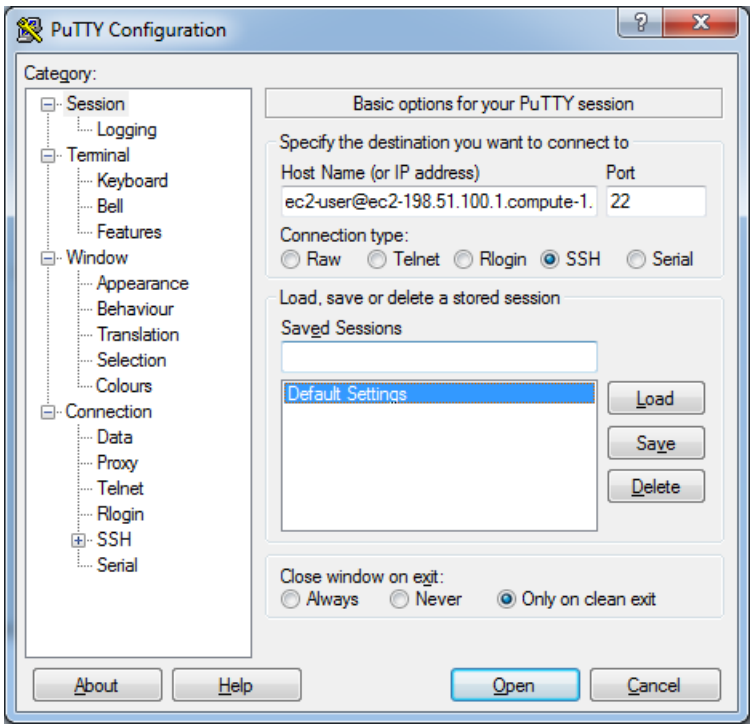
      *`user_name`*에는 AMI에 적합한 사용자 이름을 지정해야 합니다. 예:

      - Amazon Linux 2 또는 Amazon Linux AMI의 경우 사용자 이름은 `ec2-user`입니다.

- CentOS AMI의 경우 사용자 이름은 `centos`입니다.
- Debian AMI의 경우 사용자 이름은 `admin` 또는 `root`입니다.
- Fedora AMI의 경우 사용자 이름은 `ec2-user` 또는 `fedora`입니다.
- RHEL AMI의 경우 사용자 이름은 `ec2-user` 또는 `root`입니다.
- SUSE AMI의 경우 사용자 이름은 `ec2-user` 또는 `root`입니다.
- Ubuntu AMI의 경우 사용자 이름은 `ubuntu`입니다.
- `ec2-user` 및 `root`를 사용할 수 없는 경우 AMI 공급자에게 문의하십시오.

b. Port(포트) 값이 22인지 확인합니다.

c. 연결 유형 아래에서 SSH를 선택합니다.



3. (선택 사항) 세션의 활성 상태를 유지하기 위해 일정 간격으로 'keepalive' 데이터를 자동 전송하도록 PuTTY를 구성할 수 있습니다. 이는 세션 비활성으로 인한 인스턴스 연결 해제를 방지하는 데 유용한 기능입니다. 범주 창에서 연결을 선택한 다음, keepalive 간 초(Seconds between keepalives) 필드에 필요한 간격을 입력합니다. 예를 들어 비활성 상태가 되고 10분 후에 세션 연결이 해제되는 경우, 180을 입력하여 3분마다 keepalive 데이터를 전송하도록 PuTTY를 구성합니다.

4. 범주 창에서 연결, SSH를 확장한 다음 Auth를 선택합니다. 다음 작업을 완료합니다.

   a. 찾아보기를 선택합니다.

   b. 키 페어에 대해 생성한 `.ppk` 파일을 선택한 다음 열기를 선택합니다.

   c. (선택 사항) 이 세션을 나중에 다시 시작하려는 경우 세션 정보를 나중에 사용할 수 있게 저장할 수 있습니다. 범주에서 세션을 선택하고 저장된 세션(Saved Sessions)에 세션 이름을 입력한 다음 저장을 선택합니다.

   d. 열기를 선택합니다.

5. 이 인스턴스에 처음 연결한 경우 PuTTY에서 연결하려는 호스트를 신뢰할 수 있는지 묻는 보안 알림 대화 상자가 표시됩니다.

   a. (선택 사항) 보안 알림 대화 상자의 지문이 앞의 (선택 사항) 인스턴스 지문 가져오기 (p. 486)에서 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.

b. 예를 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 해결을 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. 501) 단원을 참조하십시오.

PSCP를 사용하려면 PuTTYgen을 사용하여 프라이빗 키 변환 (p. 498)에서 생성한 프라이빗 키가 필요합니다. 또한 Linux 인스턴스의 퍼블릭 DNS 주소도 필요합니다.

다음 예에서는 `Sample_file.txt` 파일을 Windows 컴퓨터의 C:\ 드라이브에서 Amazon Linux 인스턴스의 `ec2-user` 홈 디렉터리로 전송합니다.

```
pscp -i C:\path\my-key-pair.ppk C:\path\Sample_file.txt ec2-user@public_dns:/home/ec2-user/
Sample_file.txt
```

(IPv6 전용) 다음 예에서는 인스턴스의 IPv6 주소를 이용해 `Sample_file.txt` 파일을 전송합니다. IPv6 주소는 대괄호([])로 묶어야 합니다.

```
pscp -i C:\path\my-key-pair.ppk C:\path\Sample_file.txt ec2-user@[ipv6-address]:/home/ec2-
user/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 PuTTYgen을 사용하여 프라이빗 키 변환 (p. 498)에서 생성한 프라이빗 키가 필요합니다. 또한 Linux 인스턴스의 퍼블릭 DNS 주소도 필요합니다.

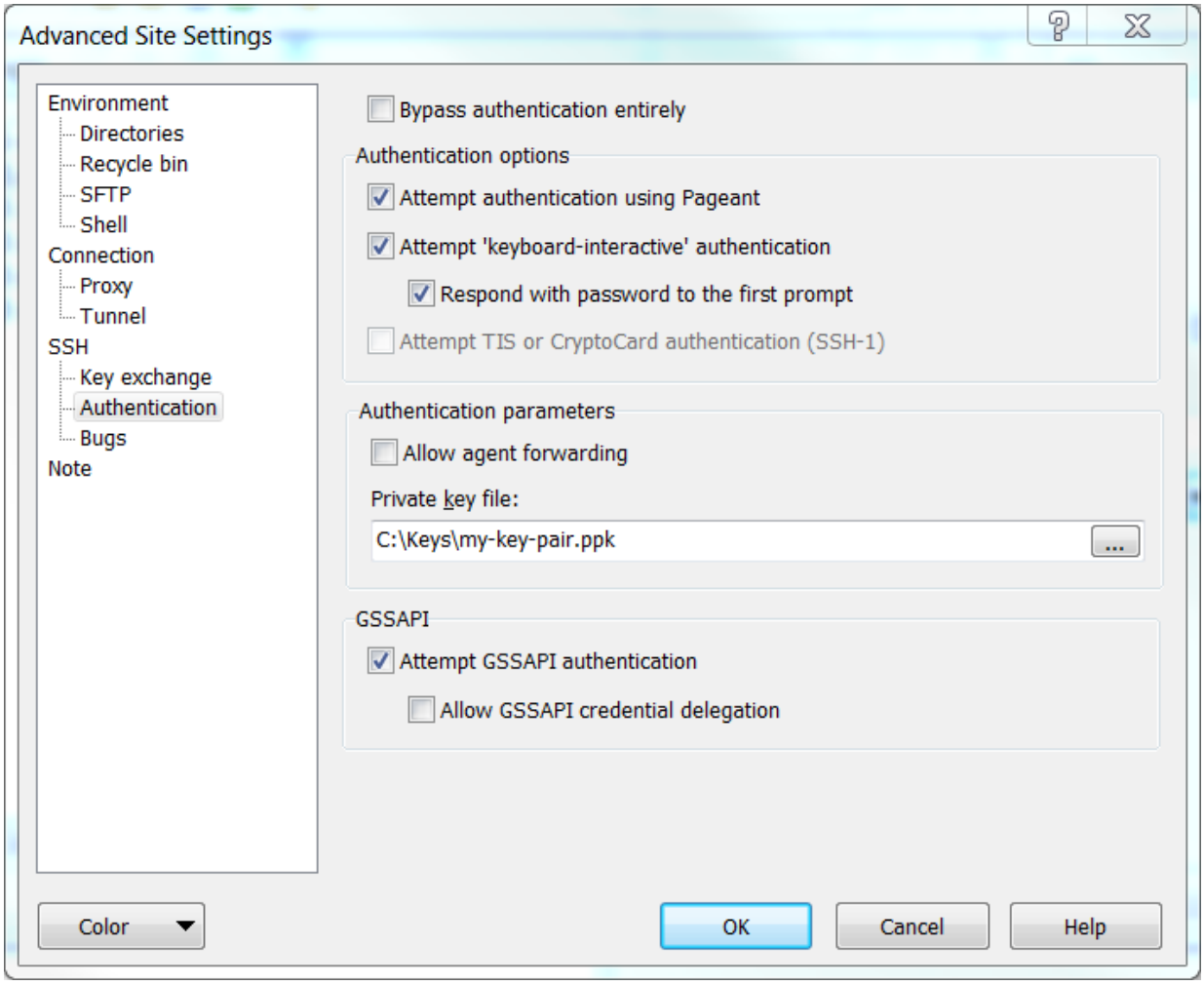
1. http://winscp.net/eng/download.php에서 WinSCP를 다운로드하여 설치합니다. 대부분 사용자의 경우 기본 설치 옵션을 그대로 사용해도 좋습니다.
2. WinSCP를 시작합니다.
3. WinSCP 로그인(WinSCP Login) 화면에서 호스트 이름(Host name)에 인스턴스의 퍼블릭 DNS 호스트 이름 또는 퍼블릭 IPv4 주소를 입력합니다.

   (IPv6 전용) 인스턴스의 IPv6 주소를 이용해 로그인하려면 인스턴스의 IPv6 주소를 입력합니다.
4. 사용자 이름에는 AMI의 기본 사용자 이름을 입력합니다.
   - Amazon Linux 2 또는 Amazon Linux AMI의 경우 사용자 이름은 `ec2-user`입니다.
   - CentOS AMI의 경우 사용자 이름은 `centos`입니다.
   - Debian AMI의 경우 사용자 이름은 `admin` 또는 `root`입니다.

- Fedora AMI의 경우 사용자 이름은 `ec2-user` 또는 `fedora`입니다.
- RHEL AMI의 경우 사용자 이름은 `ec2-user` 또는 `root`입니다.
- SUSE AMI의 경우 사용자 이름은 `ec2-user` 또는 `root`입니다.
- Ubuntu AMI의 경우 사용자 이름은 `ubuntu`입니다.
- `ec2-user` 및 `root`를 사용할 수 없는 경우 AMI 공급자에게 문의하십시오.

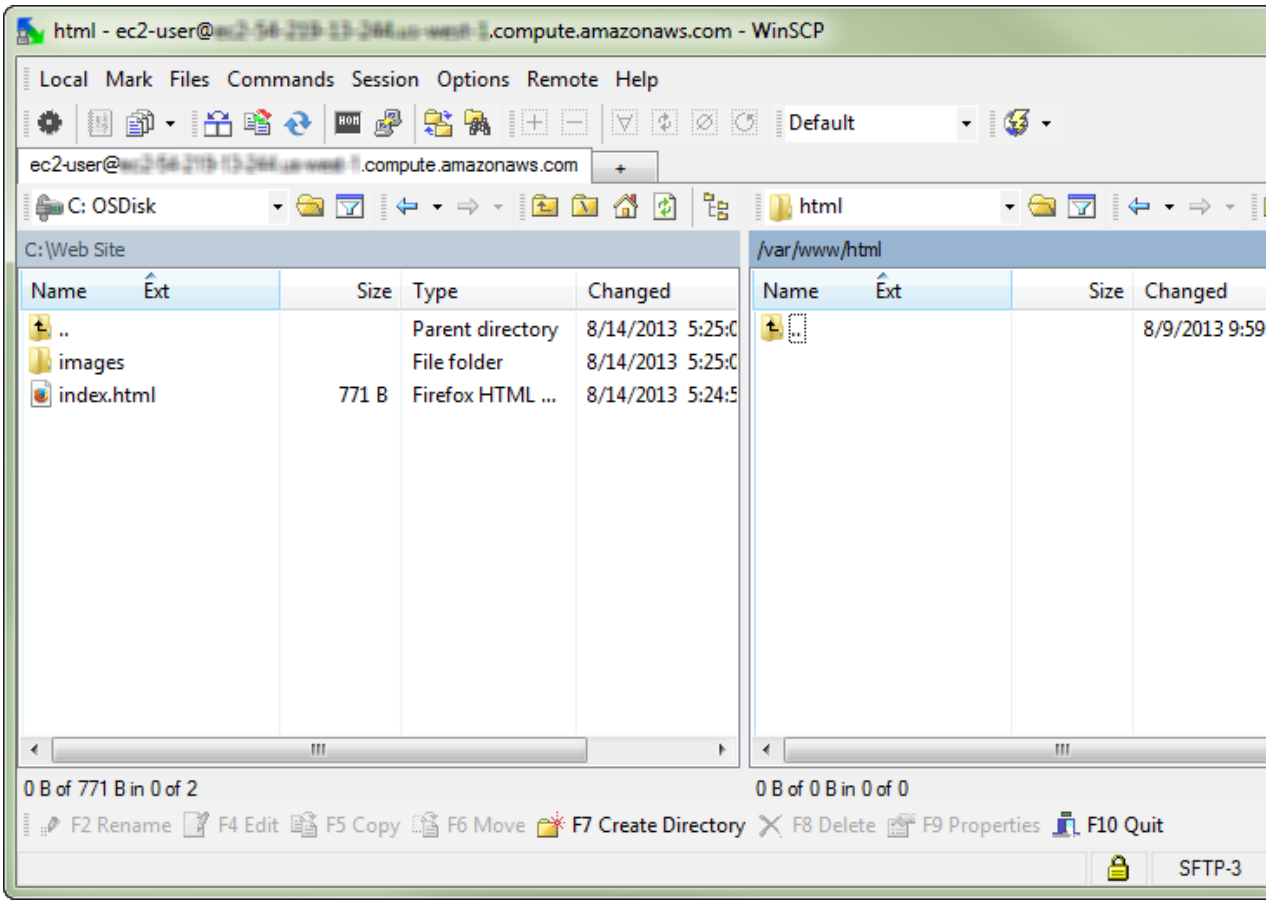
5. 인스턴스의 프라이빗 키를 지정합니다. 프라이빗 키에서는 프라이빗 키의 경로를 입력하거나 "..." 버튼을 선택하여 파일을 찾아봅니다. 고급 사이트 설정을 열려면 최신 버전의 WinSCP에 대해 고급을 선택합니다. Private key file(프라이빗 키 파일) 설정을 찾으려면 SSH에서 Authentication(인증)을 선택합니다.

   다음은 WinSCP 버전 5.9.4의 스크린샷입니다.



   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. 498) 단원을 참조하십시오.

6. (선택 사항) 왼쪽 패널에서 Directories(디렉터리)를 선택합니다. Remote directory(원격 디렉터리)에서 파일을 추가할 디렉터리의 경로를 입력합니다. 고급 사이트 설정을 열려면 최신 버전의 WinSCP에 대해 고급을 선택합니다. Remote directory(원격 디렉터리) 설정을 찾으려면 Environment(환경)에서 Directories(디렉터리)를 선택합니다.
7. Login(로그인)을 선택합니다. 예를 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/start의 프로젝트 설명서를 참조하십시오.

   "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
```

Windows Subsystem for Linux를 사용하여 Windows에서 Linux 인스턴스에 연결

인스턴스를 시작한 후 인스턴스에 연결하고 바로 앞에 있는 컴퓨터를 사용하는 것처럼 인스턴스를 사용할 수 있습니다.

다음 지침에서는 Windows Subsystem for Linux(WSL)에서 Linux 배포를 사용하여 인스턴스에 연결하는 방법을 설명합니다. WSL은 무료로 다운로드할 수 있으며 가상 머신이라는 오버헤드 없이 기본 Linux 명령줄 도구를 Windows는 물론 기존 Windows 데스크탑에서 바로 실행할 수 있습니다.

WSL을 설치하면 PuTTY 또는 PuTTYgen 대신 기본 Linux 환경을 사용해서 Linux EC2 인스턴스에 연결할 수 있습니다. Linux 환경에서는 Linux 인스턴스에 연결하고 .pem 키 파일의 권한을 변경할 수 있는 기본 SSH 클라이언트가 있으므로 Linux 인스턴스에 쉽게 연결할 수 있습니다. Amazon EC2 콘솔이 제공하는 SSH 명

령으로 Linux 인스턴스에 연결할 수 있으며, SSH 명령에서 얻은 VERBOSE 출력을 얻어 문제를 해결할 수 있습니다. 자세한 내용은 Windows Subsystem for Linux 설명서를 참조하십시오.

> **Note**
>
> WSL을 설치한 후 모든 사전 조건과 단계는 SSH를 사용하여 Linux 인스턴스에 연결 (p. 487)의 설명과 동일하며, 이용 경험은 기본 Linux를 사용하는 것과 동일합니다.

인스턴스에 연결을 시도하는 동안 오류가 발생한 경우 인스턴스 연결 문제 해결을 참조하십시오.

목차

- 사전 조건 (p. 487)
- WSL을 사용하여 Linux 인스턴스에 연결 (p. 504)
- SCP를 사용하여 Linux에서 Linux 인스턴스로 파일 전송 (p. 505)
- WSL 제거 (p. 506)

사전 조건

Linux 인스턴스에 연결하려면 먼저 다음 사전 조건을 완료하십시오.

인스턴스가 준비되었는지 확인

인스턴스를 시작한 후, 연결할 수 있도록 인스턴스가 준비될 때까지 몇 분 정도 걸릴 수 있습니다. 인스턴스가 상태 확인을 통과했는지 확인합니다. 인스턴스 페이지의 상태 확인 열에서 이 정보를 볼 수 있습니다.

인스턴스에 연결하기 위한 일반 사전 조건 확인

자세한 내용은 인스턴스에 연결하기 위한 일반 사전 조건 (p. 485) 단원을 참조하십시오.

로컬 컴퓨터에 Windows Subsystem for Linux(WSL) 및 Linux 배포 설치

Windows 10 설치 가이드의 지침을 이용하여 WSL과 Linux 배포를 설치하십시오. 지침 속 사례는 Linux의 Ubuntu 배포를 설치하는 것이지만, 다른 배포의 설치에도 활용할 수 있습니다. 변경 사항을 적용하려면 컴퓨터를 다시 시작하라는 안내가 표시됩니다.

Windows에서 WSL로 프라이빗 키 복사

WSL 터미널 창에서 `.pem` 파일(인스턴스 시작 시 지정한 키 페어의 경우)을 Windows에서 WSL로 복사합니다. 인스턴스에 연결할 때는 WSL에서 `.pem` 파일의 정규화된 경로를 확인하십시오. Windows 하드 드라이브의 경로를 지정하는 방법에 대한 자세한 내용은 내 C: 드라이브에는 어떻게 액세스하나요?를 참조하십시오.

```
cp /mnt/<Windows drive letter>/path/my-key-pair.pem ~/WSL-path/my-key-pair.pem
```

WSL을 사용하여 Linux 인스턴스에 연결

Windows Subsystem for Linux(WSL)를 사용하여 Linux 인스턴스에 연결하려면 다음 절차를 사용하십시오. 인스턴스에 연결을 시도하는 동안 오류가 발생한 경우 인스턴스 연결 문제 해결을 참조하십시오.

SSH를 사용하여 인스턴스에 연결하려면

1. 터미널 창에서 ssh 명령을 사용하여 인스턴스에 연결합니다. 프라이빗 키(`.pem`) 파일, AMI의 사용자 이름 및 인스턴스의 퍼블릭 DNS 이름을 지정합니다. 예를 들어 Amazon Linux 2 또는 Amazon Linux AMI를 사용한 경우 사용자 이름은 `ec2-user`입니다. 인스턴스의 AMI 및 DNS 이름에 대한 사용자 이름을 찾는 방법에 대한 자세한 내용은 인스턴스에 대한 정보 가져오기 (p. 485) 단원을 참조하십시오.

```
sudo 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)?
```

2. (IPv6 전용) 또는 IPv6 주소를 이용해 인스턴스에 연결할 수도 있습니다. 프라이빗 키(.pem) 파일 경로 및 적절한 사용자 이름과 IPv6 주소를 사용하여 ssh 명령을 지정합니다. 예를 들어 Amazon Linux 2 또는 Amazon Linux AMI를 사용한 경우 사용자 이름은 `ec2-user`입니다.

```
sudo ssh -i /path/my-key-pair.pem ec2-user@2001:db8:1234:1a00:9691:9503:25ad:1761
```

3. (선택 사항) 보안 알림의 지문이 앞의 (선택 사항) 인스턴스 지문 가져오기 (p. 486)에서 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.
4. `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.
```

SCP를 사용하여 Linux에서 Linux 인스턴스로 파일 전송

로컬 컴퓨터와 Linux 인스턴스 간에 파일을 전송하는 한 가지 방법은 SCP(Secure Copy Protocol)를 사용하는 것입니다. 이 섹션에서는 SCP를 사용하여 파일을 전송하는 방법을 설명합니다. 이 절차는 SSH를 사용하여 인스턴스에 연결하는 절차와 비슷합니다.

사전 조건

- 파일을 인스턴스에 전송하기 위한 일반 사전 조건을 확인합니다.

  파일을 인스턴스에 전송하기 위한 일반 사전 조건은 인스턴스에 연결하기 위한 일반 사전 조건과 동일합니다. 자세한 내용은 인스턴스에 연결하기 위한 일반 사전 조건 (p. 485)을 참조하십시오.
- 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.org를 참조하십시오.

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. 인스턴스의 퍼블릭 DNS 이름을 사용하여 인스턴스로 파일을 전송합니다. 예를 들어, 프라이빗 키 파일의 이름이 `my-key-pair`이고, 전송할 파일이 `SampleFile.txt`이며, 사용자 이름이 `ec2-user`이고, 인스턴스의 퍼블릭 DNS 이름이 `ec2-198-51-100-1.compute-1.amazonaws.com`인 경우, 다음 명령을 사용하여 파일을 `ec2-user` 홈 디렉터리로 복사합니다:

```
scp -i /path/my-key-pair.pem /path/SampleFile.txt ec2-
user@c2-198-51-100-1.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)?
```

2. (IPv6 전용) 또는 인스턴스의 IPv6 주소를 이용해 파일을 전송할 수도 있습니다. IPv6 주소는 이스케이프된(\) 대괄호([])로 묶어야 합니다.

```
scp -i /path/my-key-pair.pem /path/SampleFile.txt ec2-user@
\[2001:db8:1234:1a00:9691:9503:25ad:1761\]:~
```

3. (선택 사항) 보안 알림의 지문이 앞의 (선택 사항) 인스턴스 지문 가져오기 (p. 486)에서 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.
4. **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
```

   "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
```

5. 반대 방향으로(Amazon EC2 인스턴스에서 로컬 컴퓨터로) 파일을 전송하려면 호스트 파라미터의 순서를 역순으로 지정하면 됩니다. 예를 들어, EC 인스턴스의 `SampleFile.txt` 파일을 로컬 컴퓨터의 홈 디렉터리에 `SampleFile2.txt`2.로 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
```

6. (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
```

WSL 제거

Windows Subsystem for Linux의 제거 방법은 WSL 배포 제거 방법을 참조하십시오.

세션 관리자를 사용하여 Linux 인스턴스에 연결

세션 관리자는 대화형 원클릭 브라우저 기반 셸이나 AWS CLI를 통해 Amazon EC2 인스턴스를 관리할 수 있는 완전 관리형 AWS 시스템 관리자 기능입니다. 세션 관리자를 사용하여 계정의 인스턴스와 관련된 세션을 시작할 수 있습니다. 세션이 시작된 후에는 다른 연결 유형과 마찬가지로 bash 명령을 실행할 수 있습니다. 세션 관리자에 대한 자세한 내용은 AWS 시스템 관리자 사용자 안내서의 AWS 시스템 관리자 세션 관리자를 참조하십시오.

세션 관리자를 사용하여 인스턴스에 연결하기 전에 필요한 설정 단계가 완료되었는지 확인합니다. 자세한 내용과 지침은 세션 관리자 시작하기를 참조하십시오.

Amazon EC2 콘솔에서 세션 관리자를 사용하여 Linux 인스턴스에 연결하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택한 다음 연결을 선택합니다.
4. 연결 방법으로 세션 관리자를 선택합니다.
5. [Connect]를 선택합니다.

문제 해결

하나 이상의 시스템 관리자 작업(`ssm:`*`command-name`*)을 수행할 권한이 없다는 오류가 표시되면 Amazon EC2 콘솔에서 세션을 시작할 수 있도록 정책을 업데이트해야 합니다. 자세한 내용은 AWS 시스템 관리자 사용 설명서의 세션 관리자용 빠른 시작 기본 IAM 정책을 참조하십시오.

인스턴스 중지 및 시작

Amazon EBS 볼륨을 루트 디바이스로 사용하는 인스턴스를 중지했다가 다시 시작할 수 있습니다. 인스턴스 ID는 유지되지만 개요 (p. 507) 단원의 설명처럼 인스턴스는 변경될 수 있습니다.

인스턴스를 중지하면 인스턴스가 종료됩니다. 중지된 인스턴스에 대해 사용 요금이나 데이터 전송 요금이 부과되지는 않지만 모든 Amazon EBS 볼륨에 대한 스토리지 요금은 부과됩니다. 중지한 인스턴스를 시작할 때마다 최소 1분의 사용 요금이 부과됩니다. 1분 이후에는 사용한 시간(초)에 대해서만 요금이 부과됩니다. 예를 들어 인스턴스를 20초간 실행한 후 중지했다면 1분에 대한 요금이 부과됩니다. 인스턴스를 3분 40초간 실행한 경우 정확히 3분 40초에 대한 요금이 부과됩니다.

인스턴스가 중지되어 있는 동안 해당 루트 볼륨을 다른 볼륨과 마찬가지로 처리하고 수정할 수 있습니다. 예를 들어, 파일 시스템 문제를 복구하거나 소프트웨어를 업데이트할 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. 522) 단원을 참조하십시오. 인스턴스를 최대 절전 모드로 전환하려는 경우 Linux 인스턴스를 최대 절전 모드로 실행합니다. (p. 510) 단원을 참조하십시오. 자세한 내용은 재부팅, 중지, 최대 절전 모드 및 종료의 차이 (p. 429) 단원을 참조하십시오.

목차

- 개요 (p. 507)
- 인스턴스 중지 시 발생하는 상황(API) (p. 508)
- 인스턴스 중지 및 시작 (p. 508)
- 중지된 인스턴스 수정 (p. 509)
- 문제 해결 (p. 509)

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. 94) 단원을 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.
- 인스턴스를 중지했다가 다시 시작해도 프라이빗 IPv4 주소와 모든 IPv6 주소는 유지됩니다. AWS에서 퍼블릭 IPv4 주소를 해제했다가 인스턴스가 시작되면 새 주소를 할당합니다.
- 인스턴스가 연결된 탄력적 IP 주소를 유지합니다. 중지된 인스턴스와 연결된 모든 탄력적 IP 주소에 대한 요금이 부과됩니다. EC2-Classic을 사용하면 중단 시 탄력적 IP 주소가 인스턴스와 연결 해제됩니다. 자세한 내용은 EC2-Classic (p. 770) 단원을 참조하십시오.
- 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 EC2Config 서비스를 사용하여 Windows 인스턴스 구성(Windows 인스턴스용 Amazon EC2 사용 설명서)을 참조하십시오.
- 인스턴스가 Auto Scaling 그룹에 있는 경우, Amazon EC2 Auto Scaling 서비스는 중단된 인스턴스를 비정상으로 간주해 이를 종료하고 대체 인스턴스를 시작합니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Auto Scaling 인스턴스에 대한 상태 점검 단원을 참조하십시오.
- ClassicLink 인스턴스를 중지하면 연결되었던 VPC와의 연결이 해제됩니다. 인스턴스를 시작한 후 VPC에 다시 연결해야 합니다. ClassicLink에 대한 자세한 내용은 ClassicLink (p. 777) 단원을 참조하십시오.

자세한 내용은 재부팅, 중지, 최대 절전 모드 및 종료의 차이 (p. 429) 단원을 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 시 발생하는 상황(API)

`stop-instances` 명령을 사용하여 EC2 인스턴스를 중지하면 OS 수준에서 다음 항목이 등록됩니다.

- API 요청은 버튼 누름 이벤트를 게스트로 전송합니다.
- 다양한 시스템 서비스는 버튼 누름 이벤트의 결과로 중지됩니다. systemd는 시스템의 정상 종료를 처리합니다. 정상 종료는 하이퍼바이저에서 ACPI 종료 버튼 누름 이벤트에 의해 트리거됩니다.
- ACPI 종료가 시작됩니다.
- 정상 종료 프로세스가 종료되면 인스턴스가 종료됩니다. 구성 가능한 OS 종료 시간은 없습니다.
- 인스턴스 OS가 4분 이내에 완전히 종료되지 않으면 하드 재부팅이 수행됩니다.

인스턴스 중지 및 시작

콘솔이나 명령줄을 사용하여 Amazon EBS 기반 인스턴스를 중지했다가 시작할 수 있습니다.

기본적으로 shutdown 또는 poweroff 명령을 사용하여 Amazon EBS 지원 인스턴스에서 종료를 시작하면 인스턴스가 중지됩니다. 인스턴스가 중지되지 않고 종료되도록 이 동작을 변경할 수 있습니다. 자세한 내용은 인스턴스가 개시하는 종료 동작 변경 (p. 525) 단원을 참조하십시오.

콘솔을 사용하여 Amazon EBS 기반 인스턴스를 중지하고 시작하려면

1. 탐색 창에서 인스턴스를 선택하고 인스턴스를 선택합니다.
2. [Actions], [Instance State], [Stop]을 차례로 선택합니다. 중지가 비활성화되어 있으면 해당 인스턴스가 이미 중지되었거나 루트 디바이스가 인스턴스 스토어 볼륨인 것입니다.

   Warning

   인스턴스를 중지하면 인스턴스 스토어 볼륨의 데이터가 삭제됩니다. 인스턴스 스토어 볼륨의 데이터를 유지하려면 영구 스토리지에 백업하십시오.
3. 확인 대화 상자가 나타나면 예, 중지를 선택합니다. 인스턴스가 중지하는 데 몇 분 정도 걸릴 수 있습니다.
4. 인스턴스가 중지되어 있는 동안 특정 인스턴스 속성을 수정할 수 있습니다. 자세한 내용은 중지된 인스턴스 수정 (p. 509) 단원을 참조하십시오.
5. 중지된 인스턴스를 시작하려면 그 인스턴스를 선택하고 작업, 인스턴스 상태, 시작을 선택합니다.
6. 확인 대화 상자가 나타나면 예, 시작을 선택합니다. 인스턴스가 `running` 상태가 되는 데 몇 분 정도 걸릴 수 있습니다.

명령줄을 사용하여 Amazon EBS 기반 인스턴스를 중지하고 시작하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- stop-instances 및 start-instances(AWS CLI)
- Stop-EC2Instance 및 Start-EC2Instance(Windows PowerShell용 AWS 도구)

중지된 인스턴스 수정

AWS Management 콘솔 또는 명령줄 인터페이스를 사용하여 중지된 인스턴스의 인스턴스 유형, 사용자 데이터 및 EBS 최적화 속성을 변경할 수 있습니다. AWS Management 콘솔을 사용하여 `DeleteOnTermination`, 커널 또는 RAM 디스크 속성을 수정할 수 없습니다.

인스턴스 속성을 수정하려면

- 인스턴스 유형을 변경하려면 인스턴스 유형 변경 (p. 262)을 참조하십시오.
- 인스턴스의 사용자 데이터를 변경하려면 인스턴스 사용자 데이터로 작업 (p. 583)를 참조하십시오.
- 인스턴스의 EBS-최적화를 설정 또는 해제하려면 EBS-최적화 수정 (p. 996)을 참조하십시오.
- 인스턴스의 루트 볼륨의 `DeleteOnTermination` 속성을 변경하려면 다음(실행 인스턴스의 블록 디바이스 매핑 업데이트 (p. 1055))을 참조하십시오.

명령줄을 사용하여 인스턴스 속성을 수정하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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. 1092) 단원을 참조하십시오.

Linux 인스턴스를 최대 절전 모드로 실행합니다.

인스턴스를 최대 절전 모드로 전환하면 운영 체제에서 최대 절전 모드(디스크 일시 중단)를 수행하도록 신호를 보냅니다. 최대 절전 모드는 인스턴스 메모리(RAM)의 내용을 Amazon EBS 루트 볼륨에 저장합니다. 인스턴스의 Amazon EBS 루트 볼륨과 연결된 모든 Amazon EBS 데이터 볼륨을 유지합니다. 인스턴스를 시작할 때:

- Amazon EBS 루트 볼륨이 이전 상태로 복원됩니다
- RAM 내용이 다시 로드됩니다
- 이전에 인스턴스에서 실행되었던 프로세스가 재개됩니다.
- 이전에 연결된 데이터 볼륨이 다시 연결되고, 인스턴스는 해당 인스턴스 ID를 유지합니다.

인스턴스에 대해 최대 절전 모드가 활성화 (p. 515)되어 있고 최대 절전 모드 사전 조건 (p. 511)을 충족하는 경우 인스턴스를 최대 절전 모드로 전환할 수 있습니다.

인스턴스 또는 애플리케이션에서 최적의 생산성을 내기 위해 메모리 공간을 부트스트랩 및 빌드하는 데 시간이 오래 걸리는 경우 최대 절전 모드를 사용해 인스턴스를 사전 워밍힐 수 있습니다. 인스턴스를 사전 예열하려면 다음을 수행하십시오.

1. 최대 절전 모드가 활성화된 상태에서 시작하십시오.
2. 원하는 상태로 만듭니다.
3. 최대 절전 모드로 전환하여 필요에 따라 동일한 상태로 다시 시작할 수 있습니다.

`stopped` 상태인 경우에는 최대 절전 모드인 인스턴스에 사용 요금을 부과하지 않습니다. RAM의 콘텐츠를 Amazon EBS 루트 볼륨으로 전송한 경우 인스턴스가 `stopping` 상태이면 인스턴스 사용에 대한 비용이 부과되지 않습니다. (이 기능은 최대 절전 모드를 실행하지 않고 인스턴스를 중지 (p. 507)할 때와 다릅니다.) 데이터 전송 요금에 대해서는 사용량이 청구되지 않습니다. 그러나 RAM 내용에 대한 Amazon EBS 스토리지를 포함하여 모든 볼륨의 스토리지에 대해서는 요금이 부과됩니다.

인스턴스가 더 이상 필요하지 않을 경우 `stopped`(최대 절전 모드) 상태인 경우를 포함해 언제든지 인스턴스를 종료할 수 있습니다. 자세한 내용은 인스턴스 종료 (p. 522) 단원을 참조하십시오.

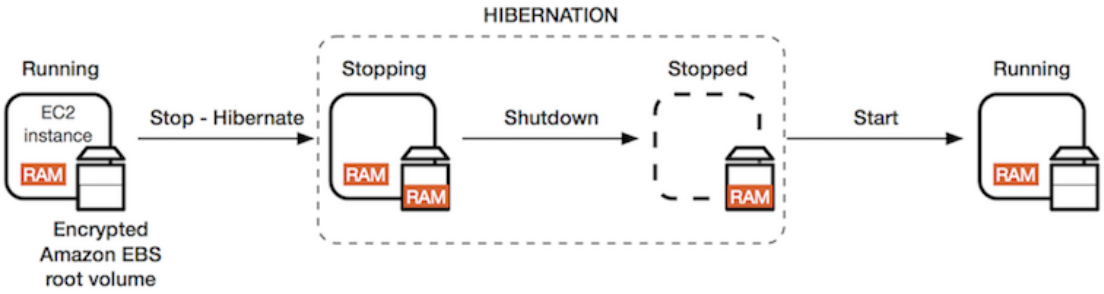
> **Note**
>
> Windows 인스턴스에서 최대 절전 모드 실행은 Windows 인스턴스용 Amazon EC2 사용 설명서 단원의 Windows 인스턴스 최대 절전 모드 실행을 참조합니다.

목차

- 최대 절전 모드 개요 (p. 510)
- 최대 절전 모드 필수 조건 (p. 511)
- 제한 사항 (p. 512)
- 최대 절전 모드를 지원하도록 기존 AMI 구성 (p. 513)
- 인스턴스에 대한 최대 절전 모드 활성화 (p. 515)
- 인스턴스에서 KASLR 비활성화하기(Ubuntu만 해당) (p. 516)
- 인스턴스를 최대 절전 모드로 전환 (p. 517)
- 최대 절전 모드의 인스턴스 시작 (p. 518)
- 최대 절전 모드 문제 해결 (p. 519)

최대 절전 모드 개요

다음 그림은 최대 절전 모드 프로세스 기본 개요를 보여줍니다.



실행 중인 인스턴스를 최대 절전 모드로 전환하면 다음과 같이 진행됩니다.

- 최대 절전 모드를 시작하면 인스턴스의 상태는 `stopping` 상태가 됩니다. 운영 체제에 최대 절전 모드(디스크 일시 중단)를 수행하도록 신호를 보냅니다. 최대 절전 모드를 실행하면 모든 프로세스가 동결되고, RAM의 콘텐츠를 Amazon EBS 루트 볼륨에 저장한 다음 일상적인 종료를 수행합니다.
- 종료가 완료되면 인스턴스가 `stopped` 상태가 됩니다.
- 모든 Amazon EBS 볼륨이 인스턴스에 연결된 상태로 유지되고 저장된 RAM 콘텐츠를 포함해 볼륨의 데이터도 유지됩니다.
- 대부분의 경우 인스턴스가 시작되면 새로운 기본 호스트 컴퓨터로 마이그레이션됩니다. 인스턴스를 중지했다가 시작할 때도 이렇게 됩니다.
- 인스턴스를 시작하면 인스턴스가 부팅되고, 운영 체제가 Amazon EBS 루트 볼륨에서 RAM의 콘텐츠를 읽은 다음 프로세스가 동결 해제되어 상태를 다시 시작합니다.
- 인스턴스를 최대 절전 모드에서 다시 시작할 때 프라이빗 IPv4 주소와 모든 IPv6 주소는 유지됩니다. AWS에서 퍼블릭 IPv4 주소를 해제했다가 인스턴스가 시작되면 새 주소를 할당합니다.
- 인스턴스가 연결된 탄력적 IP 주소를 유지합니다. 최대 절전 모드 인스턴스와 연결된 모든 엘라스틱 IP 주소에 대한 요금이 부과됩니다. EC2-Classic을 사용하면 최대 절전 모드 시 탄력적 IP 주소가 인스턴스와 연결 해제됩니다. 자세한 내용은 EC2-Classic (p. 770) 단원을 참조하십시오.
- ClassicLink 인스턴스를 최대 절전 모드로 전환하면 연결되었던 VPC와의 연결이 해제됩니다. 인스턴스를 시작한 후 VPC에 다시 연결해야 합니다. 자세한 정보는 ClassicLink (p. 777) 단원을 참조하십시오.

최대 절전 모드과 재부팅, 중지 및 종료 간의 차이점은 재부팅, 중지, 최대 절전 모드 및 종료의 차이 (p. 429) 단원을 참조하십시오.

최대 절전 모드 필수 조건

인스턴스를 최대 절전 모드로 전환하려면 다음 사전 조건을 충족해야 합니다.

- 지원되는 인스턴스 패밀리 - C3, C4, C5, M3, M4, M5, R3, R4, R5 및 T2.
- 인스턴스 RAM 크기 - 150GB 미만이어야 합니다. .
- 인스턴스 크기 - 베어 메탈 인스턴스에서는 지원되지 않습니다.
- 지원되는 AMI(최대 절전 모드를 지원하는 HVM AMI여야 함):
  - 2019.08.29 이후 Amazon Linux 2 AMI 릴리스
  - Amazon Linux AMI 2018.03, 2018.11.16 이후 릴리스
  - Ubuntu 18.04 LTS - 일련 번호 20190722.1 이상으로 릴리스된 Bionic AMI*
  - Ubuntu 16.04 LTS - Xenial AMI*(추가 구성 필수)

    *Ubuntu 18.04 LTS - Bionic 및 Ubuntu 16.04 LTS - Xenial의 인스턴스에서는 KASLR을 비활성화하는 것이 좋습니다. 자세한 내용은 인스턴스에서 KASLR 비활성화하기(Ubuntu만 해당) (p. 516) 단원을 참조하십시오.

최대 절전 모드를 지원하도록 자체 AMI를 구성하려면 최대 절전 모드를 지원하도록 기존 AMI 구성 (p. 513) 단원을 참조하십시오.

다른 버전의 Ubuntu 및 기타 운영 체제에 대한 지원도 곧 제공될 예정입니다.

Windows에 대해 지원되는 AMI에 대한 정보는, Windows 인스턴스용 Amazon EC2 사용 설명서 단원의 최대 절전 모드 요구 사항을 참조합니다.

- 루트 볼륨 유형 - 인스턴스 스토어 볼륨이 아닌 Amazon EBS 볼륨이어야 합니다.
- Amazon EBS 루트 볼륨 크기 - RAM 콘텐츠를 저장하고 예상한 사용량(예: OS 또는 애플리케이션)을 수용할 수 있을 정도로 커야 합니다. 최대 절전 모드를 활성화하면 RAM 저장을 시작할 수 있도록 루트 볼륨에 공간이 할당됩니다.
- Amazon EBS 루트 볼륨 암호화 - 최대 절전 모드를 사용하려면 최대 절전 모드 시 메모리에 있는 중요한 콘텐트를 보호할 수 있도록 루트 볼륨을 암호화해야 합니다. RAM 데이터가 Amazon EBS 루트 볼륨으로 이전하면 항상 암호화됩니다. 루트 볼륨 암호화는 인스턴스 시작 시 적용됩니다. 루트 볼륨이 암호화된 Amazon EBS 볼륨인지 확인하려면 다음 세 가지 옵션 중 하나를 사용합니다.
  - EBS '단일 단계' 암호화: 단일 실행 인스턴스 API 호출의 경우 암호화되지 않은 AMI에서 암호화된 EBS 지원 EC2 인스턴스를 시작하고, 그와 동시에 하이버네이션을 활성화할 수도 있습니다. 자세한 내용은 EBS 지원 AMI를 통한 암호화 사용 (p. 145) 단원을 참조하십시오.
  - 기본적인 EBS 암호화: 기본적으로 EBS 암호화를 활성화하여 AWS 계정에서 생성된 모든 새 EBS 볼륨이 암호화되도록 할 수 있습니다. 이러한 방식으로 인스턴스 실행 시 암호화 의도를 지정하지 않고 인스턴스에 대한 하이버네이션을 활성화할 수 있습니다. 자세한 내용은 암호화 기본 제공 (p. 971) 단원을 참조하십시오.
  - 암호화된 AMI: 암호화된 AMI를 사용하여 인스턴스를 시작하는 방식으로 EBS 암호화를 활성화할 수 있습니다. AMI에 암호화된 루트 스냅샷이 없을 경우, 이를 새로운 AMI 및 요청 암호화에 복사할 수 있습니다. 자세한 내용은 복사 중에 암호화되지 않은 이미지 암호화 (p. 149) 및 AMI 복사 (p. 154) 단원을 참조하십시오.
- 시작 시 최대 절전 모드 활성화 - 기존 인스턴스(실행 또는 중지됨)에 대해서는 최대 절전 모드를 활성화할 수 없습니다. 자세한 내용은 인스턴스에 대한 최대 절전 모드 활성화 (p. 515) 단원을 참조하십시오.
- 구입 옵션 - 이 기능은 온디맨드 인스턴스 및 예약 인스턴스에 사용할 수 있습니다. 따라서 스팟 인스턴스를 지원하지 않습니다. 자세한 내용은 중단된 스팟 인스턴스를 최대 절전 모드로 전환 (p. 374) 단원을 참조하십시오.

제한 사항

- 최대 절전 모드에서는 다음 작업을 수행할 수 없습니다.
  - 최대 절전 모드 인스턴스의 유형 또는 크기 변경
  - 최대 절전 모드가 활성화된 인스턴스에서 스냅샷 또는 AMI 생성
  - 최대 절전 모드 인스턴스에서 스냅샷 또는 AMI 생성
- 인스턴스 스토어 지원 인스턴스는 중지하거나 최대 절전 모드로 전환할 수 없습니다.*
- 150GB 이상의 RAM이 있는 인스턴스를 최대 절전 모드로 전환할 수 없습니다.
- Auto Scaling 그룹에 속하거나 Amazon ECS에서 사용하는 인스턴스는 최대 절전 모드로 전환할 수 없습니다. 인스턴스가 Auto Scaling 그룹에 있으며 최대 절전 모드로 전환하려고 하면, Amazon EC2 Auto Scaling 서비스가 중지된 인스턴스를 비정상으로 간주해 이를 종료하고 대체 인스턴스를 시작할 수 있습니다. 자세한 정보는 Amazon EC2 Auto Scaling 사용 설명서의 Auto Scaling 인스턴스에 대한 상태 점검 단원을 참조하십시오.
- 인스턴스는 60일까지만 최대 절전 모드로 유지할 수 있습니다. 인스턴스를 60일 이상 유지하려면 최대 절전 모드의 인스턴스를 시작하고, 중지하고, 다시 시작해야 합니다.
- 업그레이드 및 보안 패치를 사용해 플랫폼을 지속적으로 업데이트하는 과정에서 기존의 최대 절전 모드 인스턴스와 충돌할 수 있습니다. 필요한 업그레이드 및 보안 패치를 적용하기 위해 종료 또는 재부팅하고자 최대 절전 모드의 인스턴스를 다시 시작해야 하는 중요 업데이트에 대해서는 AWS에서 알려 드립니다.

*최대 절전 모드가 활성화된 C3 및 R3 인스턴스의 경우 인스턴스 스토어 볼륨을 사용하지 마십시오.

최대 절전 모드를 지원하도록 기존 AMI 구성

자체 AMI를 사용하여 시작한 인스턴스를 최대 절전 모드로 전환하려면 먼저 최대 절전 모드를 지원하도록 AMI를 구성해야 합니다. 자세한 내용은 인스턴스 소프트웨어 업데이트 (p. 530) 단원을 참조하십시오.

지원되는 AMI (p. 511) 중 하나를 사용하거나(Ubuntu 16.04 LTS 제외) 지원되는 AMI 중 하나를 기반으로 AMI를 생성하는 경우 최대 절전 모드를 지원하도록 구성할 필요가 없습니다. 이러한 AMI는 최대 절전 모드를 지원하도록 미리 구성되어 제공됩니다. 최대 절전 모드를 지원하도록 Ubuntu 16.04 LTS를 구성하려면 linux-aws-hwe 커널 패키지 버전 4.15.0-1058-aws 이상 및 ec2-hibinit-agent를 설치해야 합니다. 구성 단계의 경우 아래의 Ubuntu 16.04 - Xenial 탭을 선택합니다.

Amazon Linux 2

최대 절전 모드를 지원하도록 Amazon Linux 2 AMI를 구성하려면

1. 다음 명령을 사용하여 최신 커널을 4.14.138-114.102 이상으로 업데이트합니다.

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

2. 다음 명령을 사용하여 리포지토리에서 `ec2-hibinit-agent` 패키지를 설치합니다.

```
[ec2-user ~]$ sudo yum install ec2-hibinit-agent
```

3. 다음 명령을 사용하여 인스턴스를 재부팅하십시오.

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

4. 다음 명령을 사용하여 커널 버전이 4.14.138-114.102 이상으로 업데이트되었는지 확인합니다.

```
[ec2-user ~]$ uname -a
```

5. 인스턴스를 중지하고 AMI를 생성합니다. 자세한 내용은 인스턴스에서 Linux AMI 생성 (p. 113) 단원을 참조하십시오.

Amazon Linux

최대 절전 모드를 지원하도록 Amazon Linux AMI를 구성하려면

1. 다음 명령을 사용하여 최신 커널을 4.14.77-70.59 이상으로 업데이트합니다.

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

2. 다음 명령을 사용하여 리포지토리에서 `ec2-hibinit-agent` 패키지를 설치합니다.

```
[ec2-user ~]$ sudo yum install ec2-hibinit-agent
```

3. 다음 명령을 사용하여 인스턴스를 재부팅하십시오.

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

4. 다음 명령을 사용하여 커널 버전이 4.14.77-70.59 이상으로 업데이트되었는지 확인합니다.

```
[ec2-user ~]$ uname -a
```

5. 인스턴스를 중지하고 AMI를 생성합니다. 자세한 내용은 인스턴스에서 Linux AMI 생성 (p. 113) 단원을 참조하십시오.

Ubuntu 18.04 - Bionic

최대 절전 모드를 지원하도록 Ubuntu 18.04 LTS AMI를 구성하려면

1. 다음 명령을 사용하여 최신 커널을 4.15.0-1044 이상으로 업데이트합니다.

```
[ec2-user ~]$ sudo apt update
[ec2-user ~]$ sudo apt dist-upgrade
```

2. 다음 명령을 사용하여 리포지토리에서 `ec2-hibinit-agent` 패키지를 설치합니다.

```
[ec2-user ~]$ sudo apt install ec2-hibinit-agent
```

3. 다음 명령을 사용하여 인스턴스를 재부팅하십시오.

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

4. 다음 명령을 사용하여 커널 버전이 4.15.0-1044 이상으로 업데이트되었는지 확인합니다.

```
[ec2-user ~]$ uname -a
```

Ubuntu 16.04 - Xenial

최대 절전 모드를 지원하도록 Ubuntu 16.04 LTS AMI를 구성하려면

1. 다음 명령을 사용하여 최신 커널을 4.15.0-1058-aws 이상으로 업데이트합니다.

```
[ec2-user ~]$ sudo apt update
[ec2-user ~]$ sudo apt install linux-aws-hwe
```

> **Note**
>
> linux-aws-hwe 커널 패키지는 Canonical에서 완벽하게 지원됩니다. 패키지는 Ubuntu 16.04 LTS에 대한 표준 지원이 2021년 4월에 끝날 때까지 정기 업데이트를 계속 받으며, 확장된 보안 유지 관리 지원이 2024년에 끝날 때까지 추가 보안 업데이트를 받습니다. 자세한 내용은 Canonical Ubuntu 블로그의 Ubuntu 16.04 LTS용 Amazon EC2 Hibernation, 이제 사용 가능을 참조하십시오.

2. 다음 명령을 사용하여 리포지토리에서 `ec2-hibinit-agent` 패키지를 설치합니다.

```
[ec2-user ~]$ sudo apt install ec2-hibinit-agent
```

3. 다음 명령을 사용하여 인스턴스를 재부팅하십시오.

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

4. 다음 명령을 사용하여 커널 버전이 4.15.0-1058-aws 이상으로 업데이트되었는지 확인합니다.

```
[ec2-user ~]$ uname -a
```

인스턴스에 대한 최대 절전 모드 활성화

인스턴스를 최대 절전 모드로 전환하려면 먼저, 인스턴스에 대해 최대 절전 모드를 활성화해야 합니다. 최대 절전 모드를 활성화하려면 인스턴스를 시작하는 동안 수행해야 합니다.

> **Important**
>
> 인스턴스를 시작한 후에는 인스턴스에 대해 최대 절전 모드를 활성화하거나 비활성화할 수 없습니다.

Console

콘솔을 사용하여 최대 절전 모드를 활성화하려면

1. 인스턴스 시작 마법사를 사용하여 인스턴스 시작 (p. 431)의 절차를 따르십시오.
2. Amazon 머신 이미지 선택 페이지에서 최대 절전 모드를 지원하는 AMI를 선택합니다. 지원되는 AMI에 대한 자세한 내용은 최대 절전 모드 필수 조건 (p. 511) 단원을 참조하십시오.
3. Choose an Instance Type(인스턴스 유형 선택) 페이지에서 지원되는 인스턴스 유형을 선택하고 Next: Configure Instance Details(다음: 인스턴스 정보 구성)를 선택합니다. 지원되는 인스턴스 유형에 대한 자세한 내용은 최대 절전 모드 필수 조건 (p. 511) 단원을 참조하십시오.
4. Configure Instance Details(인스턴스 정보 구성) 페이지에서 Stop - Hibernate Behavior(중지 - 최대 절전 모드 동작)에 대해 Enable hibernation as an additional stop behavior(추가 중지 동작으로 최대 절전 모드 활성화) 확인란을 선택합니다.
5. 마법사에 표시되는 지침에 따라 계속합니다. 인스턴스 시작 검토 페이지에서 옵션 검토를 마쳤으면 시작을 선택합니다. 자세한 정보는 인스턴스 시작 마법사를 사용하여 인스턴스 시작 (p. 431) 단원을 참조하십시오.

AWS CLI

AWS CLI를 사용하여 최대 절전 모드를 활성화하려면

run-instances 명령을 사용하여 인스턴스를 시작합니다. `--hibernation-options Configured=true` 파라미터를 사용하여 최대 절전 모드를 활성화합니다.

```
aws ec2 run-instances --image-id ami-0abcdef1234567890 --instance-type m5.large --
hibernation-options Configured=true --count 1 --key-name MyKeyPair
```

Windows PowerShell용 AWS 도구

Windows PowerShell용 AWS 도구를 사용하여 최대 절전 모드를 활성화하려면

New-EC2Instance 명령을 사용하여 인스턴스를 시작합니다. `-HibernationOptions_Configured $true` 파라미터를 사용하여 최대 절전 모드를 활성화합니다.

```
New-EC2Instance -ImageId ami-0abcdef1234567890 -InstanceType m5.large -
HibernationOptions_Configured $true -MinCount 1 -MaxCount 1 -KeyName MyKeyPair
```

Console

콘솔을 사용하여 인스턴스에 대해 최대 절전 모드가 활성화되어 있는지 확인하려면

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

2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 세부 정보 창에서 Stop - Hibernation behavior(중지 - 최대 절전 모드 동작)를 살펴봅니다. 활성은 인스턴스에 대해 최대 절전 모드가 활성화되어 있음을 나타냅니다.

AWS CLI

AWS CLI을 사용하여 인스턴스에 대해 최대 절전 모드가 활성화되어 있는지 확인하려면

describe-instances 명령을 사용해 최대 절전 모드가 활성화된 인스턴스를 필터링하도록 `--filters "Name=hibernation-options.configured,Values=true"` 파라미터를 지정합니다.

```
aws ec2 describe-instances --filters "Name=hibernation-options.configured,Values=true"
```

출력의 다음 필드는 인스턴스에 대해 최대 절전 모드가 활성화되었음을 나타냅니다.

```
"HibernationOptions": {
    "Configured": true
}
```

Windows PowerShell용 AWS 도구

Windows PowerShell용 AWS 도구을 사용하여 인스턴스에 대해 최대 절전 모드가 활성화되어 있는지 확인하려면

Get-EC2Instance 명령을 사용해 최대 절전 모드가 활성화된 인스턴스를 필터링하도록 `-Filter @{ Name="hibernation-options.configured"; Value="true"}` 파라미터를 지정합니다.

```
Get-EC2Instance -Filter @{ Name="hibernation-options.configured"; Value="true"}
```

출력에는 최대 절전 모드로 활성화된 EC2 인스턴스가 나열됩니다.

인스턴스에서 KASLR 비활성화하기(Ubuntu만 해당)

Ubuntu 16.04 LTS - Xenial 또는 Ubuntu 18.04 LTS - Bionic(일련 번호 20190722.1 이상으로 릴리스됨)을 사용하여 새로 시작한 인스턴스에서 최대 절전 모드를 실행하려면 KASLR(커널 주소 공간 레이아웃 무작위화)을 비활성화하는 것이 좋습니다. Ubuntu 16.04 LTS 또는 Ubuntu 18.04 LTS에서는 KASLR이 기본적으로 활성화되어 있습니다. KASLR은 커널의 기본 주소 값을 무작위화하여 아직 발견되지 않은 메모리 액세스 취약점에 대한 노출 및 파급을 완화하는 데 도움이 되는 표준 Linux 커널 보안 기능입니다. KASLR을 활성화하면 인스턴스를 최대 절전 모드로 전환 한 후 다시 시작하지 못할 수도 있습니다.

KASLR에 대한 자세한 내용은 Ubuntu 기능을 참조하십시오.

Ubuntu로 시작된 인스턴스에서 KASLR을 비활성화하려면

1. SSH를 사용하여 인스턴스에 연결합니다. 자세한 내용은 SSH를 사용하여 Linux 인스턴스에 연결 (p. 487) 단원을 참조하십시오.
2. 원하는 편집기에서 `/etc/default/grub.d/50-cloudimg-settings.cfg` 파일을 엽니다. 다음 예제와 같이 `nokaslr` 옵션을 끝에 추가하려면 `GRUB_CMDLINE_LINUX_DEFAULT` 행을 편집하십시오.

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty1 console=ttyS0 nvme_core.io_timeout=4294967295
 nokaslr"
```

3. 파일을 저장하고 편집기를 종료합니다.
4. 다음 명령을 실행하여 grub 구성을 재구성합니다.

```
[ec2-user ~]$ sudo update-grub
```

5. 인스턴스를 재부팅합니다.

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

6. 다음 명령을 실행할 때 `nokaslr`이 추가되었는지 확인합니다.

```
[ec2-user ~]$ cat /proc/cmdline
```

명령의 출력에는 `nokaslr` 옵션이 포함되어야 합니다.

인스턴스를 최대 절전 모드로 전환

인스턴스에 대해 최대 절전 모드가 활성화 (p. 515)되어 있고 최대 절전 모드 사전 조건 (p. 511)을 충족하는 경우 인스턴스를 최대 절전 모드로 전환할 수 있습니다. 인스턴스를 최대 절전 모드로 전환할 수 없는 경우 정상 종료가 진행됩니다.

Console

콘솔을 사용하여 Amazon EBS 지원 인스턴스를 최대 절전 모드로 전환하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 작업, Instance State(인스턴스 상태), Stop - Hibernate(중지 - 최대 절전 모드)를 선택합니다. Stop - Hibernate(중지 - 최대 절전 모드)가 비활성화되어 있으면 해당 인스턴스가 이미 최대 절전 모드로 전환 또는 중지되었거나 최대 절전 모드로 전환할 수 없는 것입니다. 자세한 내용은 최대 절전 모드 필수 조건 (p. 511) 단원을 참조하십시오.
4. 확인 대화 상자가 나타나면 Yes, Stop - Hibernate(예, 중지 - 최대 절전 모드)를 선택합니다. 인스턴스가 최대 절전 모드로 전환하는 데 몇 분 정도 걸릴 수 있습니다. 인스턴스가 최대 절전 모드로 전환 중일 때는 Instance State(인스턴스 상태)가 중지 중으로 변경되고, 최대 절전 모드가 되면 중지됨으로 바뀝니다.

AWS CLI

AWS CLI를 사용하여 Amazon EBS 지원 인스턴스를 최대 절전 모드로 전환하려면

stop-instances 명령을 사용하여 `--hibernate` 파라미터를 지정합니다.

```
aws ec2 stop-instances --instance-ids i-1234567890abcdef0 --hibernate
```

Windows PowerShell용 AWS 도구

Windows PowerShell용 AWS 도구를 사용하여 Amazon EBS 지원 인스턴스를 최대 절전 모드로 전환하려면

Stop-EC2Instance 명령을 사용하여 `-Hibernate $true` 파라미터를 지정합니다.

```
Stop-EC2Instance -InstanceId i-1234567890abcdef0 -Hibernate $true
```

Console

콘솔을 사용하여 인스턴스에 대해 최대 절전 모드가 시작되었는지 확인하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 세부 정보 창에서 상태 전환 이유 메시지를 살펴봅니다. Client.UserInitiatedHibernate: User initiated hibernate 메시지는 인스턴스에 대해 최대 절전 모드가 시작되었음을 나타냅니다.

AWS CLI

AWS CLI를 사용하여 인스턴스에 대해 최대 절전 모드가 시작되었는지 확인하려면

describe-instances 명령을 사용해 최대 절전 모드가 시작된 인스턴스를 보려면 `state-reason-code` 필터를 지정합니다.

```
aws ec2 describe-instances --filters "Name=state-reason-
code,Values=Client.UserInitiatedHibernate"
```

출력의 다음 필드는 인스턴스에 대해 최대 절전 모드가 시작되었음을 나타냅니다.

```
"StateReason": {
    "Code": "Client.UserInitiatedHibernate"
}
```

Windows PowerShell용 AWS 도구

Windows PowerShell용 AWS 도구를 사용하여 인스턴스에 대해 최대 절전 모드가 시작되었는지 확인하려면

Get-EC2Instance 명령을 사용해 최대 절전 모드가 시작된 인스턴스를 보려면 `state-reason-code` 필터를 지정합니다.

```
Get-EC2Instance -Filter @{Name="state-reason-
code";Value="Client.UserInitiatedHibernate"}
```

출력에는 최대 절전 모드가 시작된 EC2 인스턴스가 나열됩니다.

최대 절전 모드의 인스턴스 시작

중지된 인스턴스를 다시 시작하는 것과 같은 방법으로 최대 절전 모드 인스턴스를 시작합니다.

Console

콘솔을 사용하여 최대 절전 모드 인스턴스를 시작하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 최대 절전 모드 인스턴스를 선택하고 작업, Instance State(인스턴스 상태), 시작을 선택합니다. 인스턴스가 `running` 상태가 되는 데 몇 분 정도 걸릴 수 있습니다. 이 과정에서 인스턴스 상태 확인 (p. 603)에는 그 인스턴스가 시작될 때까지 실패 상태가 표시됩니다.

AWS CLI

AWS CLI를 사용하여 최대 절전 모드 인스턴스를 시작하려면

아래와 같이 start-instances 명령을 사용합니다.

```
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

Windows PowerShell용 AWS 도구

Windows PowerShell용 AWS 도구을 사용하여 최대 절전 모드 인스턴스를 시작하려면

Start-EC2Instance 명령을 사용합니다.

```
Start-EC2Instance -InstanceId i-1234567890abcdef0
```

최대 절전 모드 문제 해결

이 정보를 사용하여 인스턴스를 최대 절전 모드로 전환할 때 발생할 수 있는 문제를 진단 및 수정합니다.

시작 직후 최대 절전 모드로 전환할 수 없음

인스턴스를 시작한 후 너무 빨리 인스턴스를 최대 절전 모드로 전환하려고 하면 오류 메시지가 표시됩니다.

시작 후 최대 절전 모드로 전환하려면 약 2~분 가량 기다려야 합니다.

`stopping`에서 `stopped`로 전환하는 데 너무 오래 걸리고 시작 후 메모리 상태가 복원되지 않음

최대 절전 모드 중인 인스턴스가 `stopping` 상태에서 `stopped` 상태로 전환되는데 너무 오래 걸리고 시작 후 메모리 상태가 복원되지 않는 경우 최대 절전 모드가 적절하게 구성되지 않았을 수 있습니다.

인스턴스 시스템 로그를 확인하고 최대 절전 모드와 관련된 메시지를 살펴보십시오. 시스템 로그에 액세스하려면 인스턴스에 연결 (p. 485)하거나 get-console-output 명령을 사용합니다. `hibinit-agent`에서 로그 줄을 찾습니다. 긴 줄에 실패라고 표시되거나 로그 줄이 없는 경우 시작 시 최대 절전 모드 구성에 실패했을 가능성이 큽니다.

예를 들어, 다음 메시지는 인스턴스 루트 볼륨이 충분히 크지 않음을 나타냅니다: `hibinit-agent: Insufficient disk space. Cannot create setup for hibernation. Please allocate a larger root device.`

`hibinit-agent`의 마지막 로그 줄이 `hibinit-agent: Running: swapoff /swap`이면 최대 절전 모드가 성공적으로 구성된 것입니다.

이러한 프로세스에서 어떠한 로그도 볼 수 없는 경우 AMI가 최대 절전 모드를 지원하지 않을 수 있습니다. 지원되는 AMI에 대한 내용은 최대 절전 모드 필수 조건 (p. 511) 단원을 참조하십시오. 자체 AMI를 사용하는 경우 최대 절전 모드를 지원하도록 기존 AMI 구성 (p. 513)의 지침을 따라야 합니다.

인스턴스가 `stopping` 상태에 멈춰 있음

인스턴스를 최대 절전 모드로 전환했는데 `stopping` 상태에 "멈춰" 있으면 강제로 중지할 수 있습니다. 자세한 내용은 인스턴스 중지 문제 해결 (p. 1092) 단원을 참조하십시오.

인스턴스 재부팅

인스턴스 재부팅은 운영 체제 재부팅과 같습니다. 대부분의 경우 인스턴스를 재부팅하는 데는 몇 분 밖에 걸리지 않습니다. 인스턴스를 재부팅하는 경우 퍼블릭 DNS 이름(IPv4), 프라이빗 IPv4 주소, IPv6 주소(해당되는 경우) 및 인스턴스 스토어 볼륨의 모든 데이터가 유지됩니다.

인스턴스를 중지했다가 다시 시작할 때와는 달리, 인스턴스를 재부팅해도 인스턴스 청구 (최소 1분 요금 포함)이 새로 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. 607) 단원을 참조하십시오.

Amazon EC2 콘솔, 명령줄 도구 또는 Amazon EC2 API를 사용하여 인스턴스에서 운영 체제 재부팅 명령을 실행하는 대신 인스턴스를 재부팅하는 것이 좋습니다. Amazon EC2 콘솔, 명령줄 도구 또는 Amazon EC2 API를 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 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 작업, 인스턴스 상태, 재부팅을 차례로 선택합니다.
4. 확인 메시지가 표시되면 예, 재부팅을 선택합니다. 인스턴스는 “실행 중” 상태로 유지됩니다.

명령줄을 사용하여 인스턴스를 재부팅하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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.

내용

- 만료 예약된 인스턴스 식별 (p. 520)
- 만료 예약된 인스턴스 관련 작업 (p. 521)

인스턴스 이벤트 유형에 대한 자세한 내용은 예약된 인스턴스 이벤트 (p. 607) 단원을 참조하십시오.

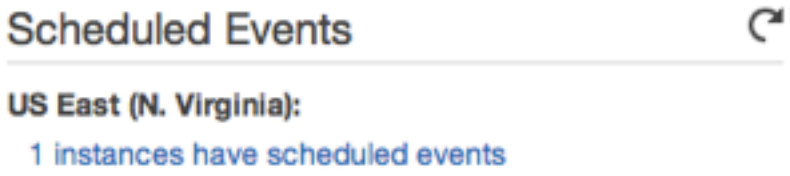
만료 예약된 인스턴스 식별

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 콘솔에

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 대시보드를 선택합니다. 예약된 이벤트에서, 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 이벤트 페이지로 이동합니다.
4. 이벤트 페이지에는 모든 리소스 및 연결된 이벤트가 나열됩니다. 만료가 예약되어 있는 인스턴스를 보려면 첫 번째 필터 목록에서 인스턴스 리소스를 선택하고 두 번째 필터 목록에서 인스턴스 중지 또는 만료를 선택합니다.
5. 필터 결과에 인스턴스에 대한 만료가 예약되어 있는 것으로 나타나면 해당 인스턴스를 선택하고 세부 정보 창의 시작 시간 필드에 표시된 날짜와 시간을 기록해 둡니다. 이 날짜가 인스턴스 만료 날짜입니다.

명령줄을 사용하여 만료 예약된 인스턴스를 식별하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- describe-instance-status(AWS CLI)
- Get-EC2InstanceStatus(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 이벤트를 선택합니다. 위의 만료 예약된 인스턴스 식별 (p. 521) 절차의 설명에 따라 필터 목록을 사용하여 만료될 인스턴스를 식별합니다.
2. 리소스 ID 열에서 인스턴스 ID를 선택하여 인스턴스 페이지로 이동합니다.
3. 인스턴스를 선택하고 설명 탭의 루트 디바이스 유형 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) 단원을 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 중단 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	인스턴스에서 EBS 지원 AMI를 생성하면 백업이 생깁니다. 예약된 만료 날짜까지 기다리거나(인스턴스가 중지되는 경우), 만료 날짜 전에 인스턴스를 중지합니다. 언제든지 인스턴스를 시작할 수 있습니다. 인스턴스 중지 및 시작과 인스턴스 중지 시 발생하는 결과(예: 인스턴스와 연결된 퍼블릭, 프라이빗 및 탄력적 IP 주소에 대한 영향)에 대한 자세한 내용은 인스턴스 중지 및 시작 (p. 507) 단원을 참조하십시오.
EBS	인스턴스에서 EBS 기반 AMI를 생성하고 대체 인스턴스를 시작합니다. 자세한 내용은 Amazon EBS 지원 Linux AMI 생성 (p. 112) 단원을 참조하십시오.
인스턴스 스토어	AMI 도구를 사용하여 인스턴스에서 인스턴스 스토어 기반 AMI를 생성하고 대체 인스턴스를 시작합니다. 자세한 내용은 인스턴스 스토어 기반 Linux AMI 생성 (p. 115) 단원을 참조하십시오.
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. 126) 단원을 참조하십시오.

인스턴스 종료

더 이상 필요하지 않은 인스턴스는 삭제할 수 있습니다. 이를 인스턴스 종료라고 합니다. 인스턴스 상태가 `shutting-down` 또는 `terminated`로 변경되는 즉시 해당 인스턴스에 대한 반복적인 요금 부과가 중단됩니다.

인스턴스를 종료한 후에는 그 인스턴스에 다시 연결하거나 재시작할 수 없습니다. 하지만 동일한 AMI를 사용해서 추가 인스턴스를 실행할 수 있습니다. 인스턴스를 중지했다가 다시 시작하거나 최대 절전 모드로 전환하면 인스턴스 중지 및 시작 (p. 507) 또는 Linux 인스턴스를 최대 절전 모드로 실행합니다. (p. 510) 단원을 참조하십시오. 자세한 내용은 재부팅, 중지, 최대 절전 모드 및 종료의 차이 (p. 429) 단원을 참조하십시오.

목차

- 인스턴스 종료 (p. 523)
- 인스턴스 종료 시 발생하는 상황(API) (p. 523)
- 인스턴스 종료 (p. 523)
- 인스턴스에 대한 종료 방지 기능 활성화 (p. 524)

- 인스턴스가 개시하는 종료 동작 변경 (p. 525)
- 인스턴스 종료 시 Amazon EBS 볼륨 보존 (p. 526)
- 문제 해결 (p. 527)

인스턴스 종료

인스턴스는 종료한 후에도 잠시 동안 콘솔에 표시되며 그 이후 항목이 자동으로 삭제됩니다. 종료된 인스턴스 항목을 사용자가 직접 삭제할 수는 없습니다. 인스턴스가 종료되면 태그 및 볼륨과 같은 리소스가 해당 인스턴스에서 점차 연결 해제되며, 조금 지나면 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 정보는 인스턴스 종료 시 Amazon EBS 볼륨 보존 (p. 526) 단원을 참조하십시오.

AWS Management 콘솔, 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. 524) 단원을 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. 525) 단원을 참조하십시오.

인스턴스 종료에 대한 스크립트를 사용하는 경우 것을 보장할 방법이 없기 때문에 인스턴스가 비정상적으로 종료될 종료 스크립트를 실행합니다. Amazon EC2 가 인스턴스를 안전하게 종료하고 시스템 종료 스크립트를 실행하도록 작동하지만, 하드웨어 장애 등 특정 이벤트가 이런 시스템 종료 스크립트 실행을 방해할 수 있습니다.

인스턴스 종료 시 발생하는 상황(API)

`terminate-instances` 명령을 사용하여 EC2 인스턴스를 종료하면 OS 수준에서 다음 항목이 등록됩니다.

- API 요청은 버튼 누름 이벤트를 게스트로 전송합니다.
- 다양한 시스템 서비스는 버튼 누름 이벤트의 결과로 중지됩니다. systemd는 시스템의 정상 종료를 처리합니다. 정상 종료는 하이퍼바이저에서 ACPI 종료 버튼 누름 이벤트에 의해 트리거됩니다.
- ACPI 종료가 시작됩니다.
- 정상 종료 프로세스가 종료되면 인스턴스가 종료됩니다. 구성 가능한 OS 종료 시간은 없습니다.

인스턴스 종료

AWS Management 콘솔 또는 명령줄을 사용해서 인스턴스를 종료할 수 있습니다.

콘솔을 사용한 인스턴스 종료 방법

1. 인스턴스를 종료하기 전에, Amazon EBS 볼륨이 종료 시 삭제되지 않는지 그리고 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 인스턴스를 선택합니다.
4. 인스턴스를 선택하고 작업, 인스턴스 상태, 종료를 선택합니다.
5. 확인 메시지가 나타나면 예, 종료를 선택합니다.

명령줄을 사용한 인스턴스 종료 방법

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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 하기 위해 해당 인스턴스에 대한 종료 방지 기능을 활성화할 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. 525) 단원을 참조하십시오.

제한 사항

스팟 인스턴스의 종료 방지 기능은 활성화할 수 없습니다. — 스팟 가격이 스팟 인스턴스에 대해 지불할 금액을 넘어서면 스팟 인스턴스가 종료됩니다. 하지만 스팟 인스턴스 중단을 처리할 수 있도록 애플리케이션을 준비하는 것은 가능합니다. 자세한 내용은 스팟 인스턴스 중단 (p. 372) 단원을 참조하십시오.

`DisableApiTermination` 속성으로는 Amazon EC2 Auto Scaling의 인스턴스 종료를 방지할 수 없습니다. Auto Scaling 그룹에 있는 인스턴스의 경우 Amazon EC2 종료 보호 대신 다음의 Amazon EC2 Auto Scaling 기능을 사용합니다.

- 인스턴스 보호 기능을 사용하면 확장 시 Auto Scaling 그룹에 속한 인스턴스가 종료되지 않습니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 인스턴스 보호를 참조하십시오.
- Amazon EC2 Auto Scaling의 비정상 인스턴스 종료를 방지하려면 `ReplaceUnhealthy` 프로세스를 일시 중단하십시오. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 조정 프로세스 일시 중단 및 재개 단원을 참조하십시오.
- 종료 정책을 선택하여 Amazon EC2 Auto Scaling가 어떤 인스턴스를 먼저 종료해야 할지 지정하십시오. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 종료 정책 사용자 지정을 참조하십시오.

실행 시에 인스턴스에 대한 종료 방지 기능 활성화 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 대시보드에서 인스턴스 시작을 선택하고 마법사의 지시를 따릅니다.
3. Configure Instance Details(인스턴스 세부 정보 구성) 페이지에서 종료 방지 기능 활성화 확인란을 선택합니다.

실행 중인 또는 중단된 인스턴스에 대한 종료 방지 기능 활성화 방법

1. 인스턴스를 선택하고 작업, 인스턴스 설정, 종료 방지 기능 변경을 선택합니다.

2. 예, 활성화를 선택합니다.

실행 중인 또는 중단된 인스턴스에 대한 종료 방지 기능 비활성화 방법

1. 인스턴스를 선택하고 작업, 인스턴스 설정, 종료 방지 기능 변경을 선택합니다.
2. 예, 비활성화를 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) 단원을 참조하십시오.

- modify-instance-attribute(AWS CLI)
- Edit-EC2InstanceAttribute(Windows PowerShell용 AWS 도구)

인스턴스가 개시하는 종료 동작 변경

기본적으로 (shutdown 또는 poweroff 등과 같은 명령을 사용하여) Amazon EBS 지원 인스턴스에서 종료를 시작하면 해당 인스턴스가 중지합니다. halt는 poweroff 명령을 실행하지 않는데, 이 명령을 사용하면 인스턴스가 종료되지 않고 대신 CPU를 HLT에 배치해 인스턴스가 실행 중인 상태로 남아 있습니다. 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.

Amazon EC2 또는 명령줄을 사용하여 `InstanceInitiatedShutdownBehavior` 속성을 업데이트할 수 있습니다. `InstanceInitiatedShutdownBehavior` 속성은 인스턴스 자체의 운영 체제에서 종료하는 경우에만 적용되며, `StopInstances` API 또는 Amazon EC2 콘솔을 사용하여 인스턴스를 중지하는 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 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 작업, 인스턴스 설정, 종료 동작 변경을 선택합니다. 현재 동작은 이미 선택된 상태입니다.
4. 동작을 변경하려면 종료 동작 목록에서 옵션을 선택하고 적용을 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) 단원을 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. `DeleteOnTermination` 속성의 기본값은 볼륨이 인스턴스의 루트 볼륨인지 여부에 따라 다릅니다.

기본적으로 인스턴스의 루트 볼륨의 `DeletionOnTermination` 속성은 `true`로 설정됩니다. 따라서 기본값은 인스턴스가 종료될 때 인스턴스의 루트 볼륨을 삭제하는 것입니다. AMI를 생성한 사람과 인스턴스를 시작한 사람이 `DeletionOnTermination` 속성을 설정할 수 있습니다. AMI를 생성한 사람 또는 인스턴스를 시작한 사람이 속성을 변경하면 새로운 설정이 원래 AMI 기본 설정을 재정의합니다. AMI를 사용하여 인스턴스를 시작한 후에는 `DeletionOnTermination` 속성에 대한 기본 설정을 확인하는 것이 좋습니다.

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. 추가 비용이 청구되지 않도록 하려면 볼륨을 삭제해야 합니다. 자세한 내용은 Amazon EBS 볼륨 삭제 (p. 926) 단원을 참조하십시오.

사용 중인 EBS 볼륨의 `DeleteOnTermination` 속성 값을 확인하려면 해당 인스턴스의 블록 디바이스 매핑을 검색합니다. 자세한 내용은 인스턴스 블록 디바이스 매핑에서 EBS 볼륨 보기 (p. 1056) 단원을 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. 526)
- 명령줄을 사용하여 실행 시 유지할 루트 볼륨 변경 (p. 527)
- 명령줄을 사용해서 실행 중인 인스턴스의 루트 볼륨이 유지되도록 변경 (p. 527)

콘솔을 사용하여 실행 시 유지할 루트 볼륨 변경

콘솔을 사용하면 인스턴스를 시작할 때 `DeleteOnTermination` 속성을 변경할 수 있습니다. 실행 중인 인스턴스의 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 인스턴스 시작을 선택합니다.
3. Amazon 머신 이미지(AMI) 선택 페이지에서 AMI를 선택한 후 선택을 선택합니다.
4. 마법사 안내에 따라 인스턴스 유형 선택 및 인스턴스 세부 정보 구성 설정을 완료합니다.
5. 스토리지 추가 페이지에서 루트 볼륨에 대한 종료 시 삭제 확인란 선택을 해제합니다.
6. 나머지 마법사 페이지를 완료한 후 시작을 선택합니다.

인스턴스의 세부 정보 창에서 루트 디바이스 볼륨의 세부 정보를 조회하여 설정을 확인할 수 있습니다. 블록 디바이스 옆의 루트 디바이스 볼륨 항목을 선택합니다. 종료 시 삭제의 기본 설정은 `True`입니다. 기본 설정을 변경하면 종료 시 삭제의 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. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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.json`에서 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 디바이스 볼륨이 유지되도록 변경할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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.json`에서 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. 1093) 단원을 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. 손상된 인스턴스가 배치 그룹에 있다면, 복구된 인스턴스는 배치 그룹에서 실행됩니다. 인스턴스 복구를 위한 Amazon CloudWatch 경보 사용에 대한 자세한 내용은 Amazon CloudWatch 경보에 복구 작업 추가 (p. 640) 단원을 참조하십시오. 인스턴스 복구 실패로 인한 문제를 해결하려면 인스턴스 복구 실패 문제 해결 (p. 528) 단원을 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
- 네트워크 연결성에 영향을 주는 물리적 호스트의 하드웨어 문제

인스턴스에 퍼블릭 IPv4 주소가 있는 경우 복구 후에도 해당 퍼블릭 IPv4 주소를 유지합니다.

요구 사항

복구 작업은 다음 특성을 지닌 인스턴스에만 지원됩니다.

- 다음 인스턴스 유형 중 하나를 사용합니다. A1, C3, C4, C5, C5n, , M3, M4, M5, M5a, M5n, P3, R3, R4, R5, R5a, R5n, T2, T3, T3a, X1 또는 X1e
- Virtual Private Cloud(VPC)에서 실행
- `default` 또는 `dedicated` 인스턴스 테넌시 사용
- EBS 볼륨만 있음(인스턴스 스토어 볼륨을 구성하지 않음)

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 중지 및 시작하는 것이 좋습니다. 자세한 내용은 인스턴스 중지 및 시작 (p. 507) 단원을 참조하십시오.

자동 복구가 실패하고 원래 시스템 상태 확인 실패의 근본 원인이 하드웨어 성능 저하로 확인되는 경우 이후에 인스턴스가 사용 중지될 수 있습니다.

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 익히는 데 도움이 되는 몇 가지 일반적인 작업입니다.

내용

- 일반적인 구성 시나리오 (p. 529)
- Linux 인스턴스의 소프트웨어 관리 (p. 529)
- Linux 인스턴스의 사용자 계정 관리 (p. 535)
- EC2 인스턴스에 대한 프로세서 상태 제어 (p. 537)
- Linux 인스턴스의 시간 설정 (p. 542)
- CPU 옵션 최적화 (p. 546)
- Linux 인스턴스의 호스트 이름 변경 (p. 561)
- Your Linux 인스턴스에 동적 DNS 설정 (p. 564)
- 시작 시 Linux 인스턴스에서 명령 실행 (p. 565)
- 인스턴스 메타데이터 및 사용자 데이터 (p. 571)

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. 529) 단원을 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. 535) 단원을 참조하십시오.

Amazon Linux 인스턴스의 기본 시간 구성에서는 Amazon Time Sync Service를 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. 542) 단원을 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. 561) 단원을 참조하십시오. 인스턴스에서 동적 DNS 서비스 공급자를 사용하도록 구성할 수 있습니다. 자세한 내용은 Your Linux 인스턴스에 동적 DNS 설정 (p. 564) 단원을 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 cloud-init 명령 및 shell 스크립트라는 두 가지 유형의 사용자 데이터를 전달할 수 있습니다. 자세한 내용은 시작 시 Linux 인스턴스에서 명령 실행 (p. 565) 단원을 참조하십시오.

Linux 인스턴스의 소프트웨어 관리

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.

내용

- 인스턴스 소프트웨어 업데이트 (p. 530)
- 리포지토리 추가 (p. 531)

- Amazon Linux에서 소프트웨어 패키지 찾기 (p. 532)
- 소프트웨어 패키지 설치 (p. 533)
- 소프트웨어 컴파일 준비 (p. 534)

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. 530) 단원을 참조하십시오.

기본적으로 Amazon Linux 인스턴스는 다음 리포지토리가 활성화된 상태로 시작됩니다.

- Amazon Linux 2: `amzn2-core` 및 `amzn2extra-docker`
- Amazon Linux AMI: `amzn-main` 및 `amzn-updates`

Amazon Web Services에서 업데이트하는 이러한 리포지토리의 다양한 패키지 이외에도 다른 리포지토리에 포함된 패키지를 설치할 수 있습니다. 자세한 내용은 리포지토리 추가 (p. 531) 단원을 참조하십시오. 활성화된 리포지토리에서 패키지를 찾는 방법은 Amazon Linux에서 소프트웨어 패키지 찾기 (p. 532) 단원을 참조하십시오. Amazon Linux 인스턴스에 소프트웨어를 설치하는 방법은 소프트웨어 패키지 설치 (p. 533) 단원을 참조하십시오.

리포지토리에 저장된 소프트웨어 패키지만 사용할 수 있는 것은 아닙니다. 일부 소프트웨어의 경우 인스턴스에서 소스 코드를 컴파일해야 합니다. 자세한 내용은 소프트웨어 컴파일 준비 (p. 534) 단원을 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 정보는 Amazon Linux에 적용됩니다. 기타 배포에 대한 자세한 내용은 해당 설명서를 참조하십시오.

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 혼동 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
```

3. 나열된 패키지를 검토하고 **y**를 입력한 후 Enter 키를 눌러 업데이트를 수락합니다. 시스템의 모든 패키지를 업데이트하는 데 몇 분이 걸릴 수 있습니다. yum 출력은 실행 중인 업데이트의 상태를 보여줍니다.
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.

1. 업데이트할 패키지의 이름과 함께 yum update 명령을 실행합니다.

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

2. 나열된 패키지 정보를 검토하고 **y**를 입력한 후 Enter 키를 눌러 업데이트를 수락합니다. 해결되어야 하는 패키지 종속성이 있는 경우 둘 이상의 패키지가 나열될 수 있습니다. yum 출력은 실행 중인 업데이트의 상태를 보여줍니다.
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 정보는 Amazon Linux에 적용됩니다. 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.

/etc/yum.repos.d에 yum 리포지토리를 추가하려면 다음을 수행합니다.

1. `.repo` 파일의 위치를 찾습니다. 위치는 추가할 리포지토리에 따라 다를 수 있습니다. 이 예시에서 `.repo` 파일은 `https://www.`*`example`*`.com/`*`repository`*`.repo`에 있습니다.
2. yum-config-manager 명령을 사용하여 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
```

리포지토리를 설치한 후 다음 절차에 따라 리포지토리를 활성화해야 합니다.

/etc/yum.repos.d에 yum 리포지토리를 활성화하려면 다음을 수행합니다.

- yum-config-manager 플래그와 함께 `--enable` *`repository`* 명령을 사용합니다. 다음 명령은 Fedora 프로젝트의 EPEL(Extra Packages for Enterprise Linux) 리포지토리를 활성화합니다. 이 리포지토리는 기본적으로 Amazon Linux AMI 인스턴스의 `/etc/yum.repos.d`에 있지만 활성화되지 않은 상태입니다.

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

> **Note**
>
> Amazon Linux 2에서 EPEL 리포지토리를 활성화하려면 다음 명령을 사용합니다.
>
> ```
> [ec2-user ~]$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-
> release-latest-7.noarch.rpm
> ```
>
> Red Hat, CentOS 등의 다른 배포에서 EPEL 리포지토리를 활성화하는 방법은 https://fedoraproject.org/wiki/EPEL의 EPEL 설명서를 참조하십시오.

Amazon Linux에서 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 정보는 Amazon Linux에 적용됩니다. 기타 배포에 대한 자세한 내용은 해당 설명서를 참조하십시오.

```
[ec2-user ~]$ sudo yum search "find"
```

다음은 Amazon Linux 2의 예제 출력입니다.

```
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
============================= N/S matched: find ==============================
findutils.x86_64 : The GNU versions of find utilities (find and xargs)
gedit-plugin-findinfiles.x86_64 : gedit findinfiles plugin
ocaml-findlib-devel.x86_64 : Development files for ocaml-findlib
perl-File-Find-Rule.noarch : Perl module implementing an alternative interface to
 File::Find
robotfindskitten.x86_64 : A game/zen simulation. You are robot. Your job is to find kitten.
mlocate.x86_64 : An utility for finding files by name
ocaml-findlib.x86_64 : Objective CAML package manager and build helper
perl-Devel-Cycle.noarch : Find memory cycles in objects
perl-Devel-EnforceEncapsulation.noarch : Find access violations to blessed objects
perl-File-Find-Rule-Perl.noarch : Common rules for searching for Perl things
perl-File-HomeDir.noarch : Find your home and other directories on any platform
perl-IPC-Cmd.noarch : Finding and running system commands made easy
perl-Perl-MinimumVersion.noarch : Find a minimum required version of perl for Perl code
texlive-xesearch.noarch : A string finder for XeTeX
valgrind.x86_64 : Tool for finding memory management bugs in programs
valgrind.i686 : Tool for finding memory management bugs in programs
```

다음은 Amazon Linux의 예제 출력입니다.

```
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
```

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.

Amazon Linux 2 및 Amazon Linux의 패키지에 대한 자세한 내용은 다음을 참조하십시오.

- 패키지 리포지토리 (p. 164)
- Extras Library(Amazon Linux 2) (p. 166)

소프트웨어 패키지 설치

yum 패키지 관리자는 소프트웨어 설치를 위한 탁월한 도구로서 활성화된 모든 리포지토리를 검색하여 다양한 소프트웨어 패키지를 찾을 뿐 아니라 소프트웨어 설치 과정에서 모든 종속성을 자동으로 처리합니다.

> **Important**
>
> 이 정보는 Amazon Linux에 적용됩니다. 기타 배포에 대한 자세한 내용은 해당 설명서를 참조하십시오.

리포지토리의 패키지를 설치하려면 yum install ***package*** 명령을 사용하여 *package* 자리에 설치할 소프트웨어의 이름을 넣으면 됩니다. 예를 들어 links 텍스트 기반 웹 브라우저를 설치하려면 다음 명령을 입력합니다.

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

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 및 autoconf 등의 몇 가지 개발 도구를 설치해야 합니다.

> **Important**
>
> 이 정보는 Amazon Linux에 적용됩니다. 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.gz`입니다. 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. yumdownloader --source 명령을 사용하여 설치된 패키지의 소스 코드를 다운로드할 수 있습니다.

- yumdownloader --source ***package*** 명령을 실행하여 *package*에 대한 소스 코드를 다운로드합니다. 예를 들어 `htop` 패키지의 소스 코드를 다운로드하려면 다음 명령을 입력합니다.

  ```
  [ec2-user ~]$ yumdownloader --source htop

  Loaded plugins: priorities, update-motd, upgrade-helper
  Enabling amzn-updates-source repository
  Enabling amzn-main-source repository
  amzn-main-source
                          | 1.9 kB  00:00:00
  amzn-updates-source
                          | 1.9 kB  00:00:00
  (1/2): amzn-updates-source/latest/primary_db
                          |  52 kB  00:00:00
  (2/2): amzn-main-source/latest/primary_db
                          | 734 kB  00:00:00
  htop-1.0.1-2.3.amzn1.src.rpm
  ```

  소스 RPM의 위치는 명령을 실행했던 디렉터리에 있습니다.

Linux 인스턴스의 사용자 계정 관리

각 Linux 인스턴스는 기본 Linux 시스템 사용자 계정으로 시작됩니다. 기본 사용자 이름은 인스턴스를 시작할 때 지정된 AMI에 의해 결정됩니다.Amazon Linux 2 또는 Amazon Linux AMI의 경우 사용자 이름은 `ec2-user`입니다. CentOS의 경우 사용자 이름은 `centos`입니다. Debian의 경우 사용자 이름은 `admin` 또는 `root`입니다. Fedora의 경우 사용자 이름은 `ec2-user` 또는 `fedora`입니다. RHEL의 경우 사용자 이름은 `ec2-user` 또는 `root`입니다. SUSE의 경우 사용자 이름은 `ec2-user` 또는 `root`입니다. Ubuntu의 사용자 이름은 `ubuntu`입니다. `ec2-user` 및 `root`를 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 사용자 및 그룹을 참조하십시오.

목차

- 고려 사항 (p. 535)
- 사용자 계정 생성 (p. 535)
- 사용자 계정 제거 (p. 537)

고려 사항

기본 사용자 계정을 사용하면 많은 애플리케이션에 적합합니다. 그러나 개인이 자신의 파일과 workspace를 가질 수 있도록 사용자 계정을 추가하도록 선택할 수 있습니다. 게다가 신규 사용자 계정을 생성하는 방법은 사용이 미숙할 수 있는 여러 사용자에게 기본 사용자 계정 액세스를 허용하는 방법보다 보안상 훨씬 안전합니다. 기본 사용자 계정은 잘못 사용될 경우 시스템에 심각한 손상을 줄 수 있기 때문입니다. 자세한 내용은 EC2 인스턴스의 보안 유지를 위한 팁을 참조하십시오.

Linux 시스템 사용자 계정을 사용하는 EC2 인스턴스에 대한 사용자 SSH 액세스를 활성화하려면 해당 사용자와 SSH 키를 공유해야 합니다. 또는 EC2 인스턴스 연결을 사용하면 SSH 키를 공유하고 관리하지 않아도 사용자에게 액세스 권한을 제공할 수 있습니다. 자세한 내용은 EC2 Instance Connect를 사용하여 Linux 인스턴스에 연결 (p. 490) 단원을 참조하십시오.

사용자 계정 생성

먼저 사용자 계정을 생성한 다음, 사용자의 연결을 허용하는 SSH 퍼블릭 키를 추가하고 인스턴스에 로그인하십시오.

사용자 계정 만들기

1. 새 키 페어를 생성합니다. (p. 862) 사용자 계정을 만들 사용자에게 `.pem` 파일을 제공해야 합니다. 이 파일을 사용하여 인스턴스에 연결해야 합니다.
2. 이전 단계에서 생성한 키 페어에서 퍼블릭 키를 검색합니다.

   ```
   $  ssh-keygen -y -f /path_to_key_pair/key-pair-name.pem
   ```

   이 명령은 다음 예제와 같이 퍼블릭 키를 반환합니다.

   ```
   ssh-rsa
    AAAAB3NzaC1yc2EAAAADAQABAAABAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6Vhz2ItxCih
   +PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/
   d6RJhJOI0iBXrlsLnBItntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/
   cQk+0FzZqaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi
   +z7wB3RbBQoQzd8v7yeb7OzlPnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE
   ```

3. 인스턴스에 연결합니다.

4. adduser 명령을 사용하여 사용자 계정을 생성하여 (`/etc/passwd` 파일의 항목으로) 시스템에 추가합니다. 이 명령은 해당 계정에 그룹과 홈 디렉터리도 생성합니다. 이 예제에서 사용자 계정 이름은 *newuser*입니다.

   - Amazon Linux and Amazon Linux 2

   ```
   [ec2-user ~]$ sudo adduser newuser
   ```

   - Ubuntu

     암호 없이 사용자 계정을 만들려면 `--disabled-password` 파라미터를 포함합니다.

   ```
   [ubuntu ~]$ sudo adduser newuser --disabled-password
   ```

5. 생성할 디렉터리와 파일이 정확한 소유권을 가질 수 있도록 새 계정으로 전환합니다.

   ```
   [ec2-user ~]$ sudo su - newuser
   ```

   프롬프트가 `ec2-user`에서 *newuser*로 바뀌며 shell 세션이 새 계정으로 전환된 것을 나타냅니다.

6. 사용자 계정에 SSH 퍼블릭 키를 추가합니다. 다음 하위 단계에 따라 먼저 사용자의 홈 디렉터리에 SSH 키 파일에 대한 디렉터리를 만든 다음, 키 파일을 만들고 마지막으로 퍼블릭 키를 키 파일에 붙여넣습니다.

   a. `.ssh` 디렉터리를 *newuser* 홈 디렉터리에 만들고 파일 권한을 `700`(소유자만 디렉터리를 읽거나, 쓰거나, 열 수 있음)으로 변경합니다.

   ```
   [newuser ~]$ mkdir .ssh
   ```

   ```
   [newuser ~]$ chmod 700 .ssh
   ```

   > **Important**
   >
   > 파일 권한이 정확하지 않으면 사용자가 로그인할 수 없습니다.

   b. `authorized_keys`라는 이름의 파일을 `.ssh` 디렉터리에 만들고 파일 권한을 `600`(소유자만 파일을 읽거나 쓸 수 있음)으로 변경합니다.

   ```
   [newuser ~]$ touch .ssh/authorized_keys
   ```

   ```
   [newuser ~]$ chmod 600 .ssh/authorized_keys
   ```

   > **Important**
   >
   > 파일 권한이 정확하지 않으면 사용자가 로그인할 수 없습니다.

   c. 자주 사용하는 텍스트 편집기(예: vim 또는 nano)로 `authorized_keys` 파일을 엽니다.

   ```
   [newuser ~]$ nano .ssh/authorized_keys
   ```

   2단계에서 검색한 퍼블릭 키를 파일에 붙여넣고 변경 내용을 저장합니다.

   > **Important**
   >
   > 퍼블릭 키를 연속된 한 줄에 붙여넣는지 확인합니다. 퍼블릭 키는 여러 줄로 분할되지 않아야 합니다.

이제 사용자는 `authorized_keys` 파일에 추가한 퍼블릭 키에 해당하는 프라이빗 키를 사용하여 인스턴스에서 *newuser* 계정에 로그인할 수 있습니다. Linux 인스턴스에 연결하는 다양한 방법에 대한 자세한 내용은 Linux 인스턴스에 연결 (p. 485) 단원을 참조하십시오.

사용자 계정 제거

사용자 계정이 더 이상 필요하지 않은 경우 더 이상 사용할 수 없도록 계정을 제거할 수 있습니다.

userdel 명령을 이용해 사용자 계정을 시스템에서 제거합니다. `-r` 파라미터를 지정하면 사용자의 홈 디렉터리와 메일 스풀이 삭제됩니다. 사용자의 홈 디렉터리와 메일 스풀을 보존하려면 `-r` 파라미터를 생략합니다.

```
[ec2-user ~]$ sudo userdel -r olduser
```

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 얕은 단계) ~ C6(코어의 전원이 꺼지는 가장 깊은 유휴 단계)의 숫자로 표시됩니다. P 상태는 코어의 성능(CPU 주파수)을 제어합니다. P 상태는 P0(코어가 인텔 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.

- 범용: `m4.10xlarge` | `m4.16xlarge` | `m5.metal` | `m5d.metal`
- 컴퓨팅 최적화: `c4.8xlarge` | `c5.metal` | `c5n.metal`
- 메모리가 최적화됨: `r4.8xlarge` | `r4.16xlarge` | `r5.metal` | `r5d.metal` | `u-6tb1.metal` | `u-9tb1.metal` | `u-12tb1.metal` | `x1.16xlarge` | `x1.32xlarge` | `x1e.8xlarge` | `x1e.16xlarge` | `x1e.32xlarge` | z1d.metal
- 스토리지 최적화: `d2.8xlarge` | `i3.8xlarge` | `i3.16xlarge` | `i3.metal` | `i3en.metal` | `h1.8xlarge` | `h1.16xlarge`
- 액셀러레이티드 컴퓨팅: `f1.16xlarge` | `g3.16xlarge` | `p2.16xlarge` | `p3.16xlarge`

다음 인스턴스 유형은 운영 체제에서 프로세서 C 상태를 제어할 수 있는 기능을 제공합니다.

- 범용: `m5.12xlarge` | `m5.24xlarge` | `m5d.12xlarge` | `m5d.24xlarge` | `m5n.12xlarge` | `m5n.24xlarge` | `m5dn.12xlarge` | `m5dn.24xlarge`
- 컴퓨팅 최적화: `c5.9xlarge` | `c5.12xlarge` | `c5.18xlarge` | `c5.24xlarge` | `c5d.9xlarge` | `c5d.12xlarge` | `c5d.18xlarge` | `c5d.24xlarge` | `c5n.9xlarge` | `c5n.18xlarge`
- 메모리 최적화: `r5.12xlarge` | `r5.24xlarge` | `r5d.12xlarge` | `r5d.24xlarge` | `r5n.12xlarge` | `r5n.24xlarge` | `r5dn.12xlarge` | `r5dn.24xlarge` | `z1d.6xlarge` | `z1d.12xlarge`
- 스토리지 최적화: `i3en.12xlarge` | `i3en.24xlarge`
- 액셀러레이티드 컴퓨팅: `p3dn.24xlarge`

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 Kinux 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 페이지의 예제에서는 turbostat 유틸리티(Amazon Linux에서 기본 제공됨)를 사용하여 프로세서 주파수 및 C 상태 정보를 표시하고 stress 명령(sudo yum install -y stress 실행을 통해 설치 가능)을 사용하여 워크로드를 시뮬레이션합니다.
> 출력에 C 상태 정보가 표시되지 않을 경우 명령에 --debug 옵션을 포함시킵니다(sudo turbostat --debug stress *<options>*).

내용

- 최고 Turbo Boost 주파수에서 최상의 성능 (p. 538)
- C 상태 심화 제한을 통한 고성능 및 저 지연 시간 (p. 539)
- 변동성이 가장 낮은 기준 성능 (p. 540)

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   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
...
10.002556 sec
```

이 예제에서는 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. vCPU 3 및 10(각각은 vCPU 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
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   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 Linux 2에서 절전 상태 심화를 제한하려면

1. 원하는 편집기를 사용하여 `/etc/default/grub` 파일을 엽니다.

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

2. `GRUB_CMDLINE_LINUX_DEFAULT` 라인을 수정하고 `intel_idle.max_cstate=1` 옵션을 추가하여 `C1` 을 유휴 코어의 최대 유휴 C 상태로 설정합니다.

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,115200n8 net.ifnames=0
 biosdevname=0 nvme_core.io_timeout=4294967295 intel_idle.max_cstate=1"
GRUB_TIMEOUT=0
```

3. 파일을 저장하고 편집기를 종료합니다.
4. 다음 명령을 실행하여 부팅 구성을 재구성합니다.

```
[ec2-user ~]$ grub2-mkconfig -o /boot/grub2/grub.cfg
```

5. 인스턴스를 재부팅하여 새 커널 옵션을 활성화합니다.

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

Amazon Linux AMI에서 절전 상태 심화를 제한하려면

1. 원하는 편집기를 사용하여 `/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 `C1`을 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
 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   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 워

크로드에 대한 자세한 내용은 http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/performance-xeon-e5-v3-advanced-vector-extensions-paper.pdf를 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 Linux 2에서 절전 상태 심화를 제한하고 Turbo Boost를 비활성화하려면

1. 원하는 편집기를 사용하여 `/etc/default/grub` 파일을 엽니다.

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

2. `GRUB_CMDLINE_LINUX_DEFAULT` 라인을 수정하고 `intel_idle.max_cstate=1` 옵션을 추가하여 `C1` 을 유휴 코어의 최대 유휴 C 상태로 설정합니다.

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,115200n8 net.ifnames=0
 biosdevname=0 nvme_core.io_timeout=4294967295 intel_idle.max_cstate=1"
GRUB_TIMEOUT=0
```

3. 파일을 저장하고 편집기를 종료합니다.
4. 다음 명령을 실행하여 부팅 구성을 재구성합니다.

```
[ec2-user ~]$ grub2-mkconfig -o /boot/grub2/grub.cfg
```

5. 인스턴스를 재부팅하여 새 커널 옵션을 활성화합니다.

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

6. `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"
```

7. 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"
```

Amazon Linux AMI에서 절전 상태 심화를 제한하고 Turbo Boost를 비활성화하려면

1. 원하는 편집기를 사용하여 `/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 `C1`을 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   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 코어도 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.

Amazon은 Amazon Time Sync Service를 제공합니다. 이 서비스는 모든 EC2 인스턴스에서 액세스할 수 있고 다른 AWS 서비스에 의해서도 사용됩니다. 이 서비스에서는 각 리전의 위성 연결 및 원자 기준 시계 집합을 사용하여 NTP(Network Time Protocol)를 통해 협정 세계시(UTC) 세계 표준의 정확한 현재 시간 판독값을 제공합니다. Amazon Time Sync Service는 UTC에 추가된 윤초를 자동으로 평활화합니다.

Amazon Time Sync Service는 VPC에서 실행 중인 모든 인스턴스의 `169.254.169.123` IP 주소에서 NTP를 통해 제공됩니다. 인스턴스에서 인터넷 액세스가 필요하지 않기 때문에 액세스를 허용하도록 보안 그룹 규칙이나 네트워크 ACL 규칙을 구성할 필요가 없습니다. 최신 버전의 Amazon Linux 2 및 Amazon Linux AMI는 기본적으로 Amazon Time Sync Service와 동기화됩니다.

다음 절차에 따라 chrony 클라이언트를 사용하여 인스턴스에서 Amazon Time Sync Service를 구성합니다. 또는 외부 NTP 소스를 사용할 수 있습니다. NTP 및 퍼블릭 타임 소스에 대한 자세한 내용은 http://www.ntp.org/를 참조하십시오. 외부 NTP 타임 소스가 작동하기 위해서는 인스턴스가 인터넷에 액세스해야 합니다.

Amazon Linux AMI에서 Amazon Time Sync Service 구성

> **Note**
>
> Amazon Linux 2에서 기본 chrony 구성은 Amazon Time Sync Service IP 주소를 사용하도록 이미 설정되어 있습니다.

Amazon Linux AMI에서 Amazon Time Sync Service에서 서버 항목을 추가하도록 chrony 구성 파일을 편집해야 합니다.

Amazon Time Sync Service를 사용하도록 인스턴스를 구성하려면

1. 인스턴스를 연결하고 NTP 서비스를 제거합니다.

```
[ec2-user ~]$ sudo yum erase 'ntp*'
```

2. chrony 패키지를 설치합니다.

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

3. vim 또는 nano과 같은 텍스트 편집기를 사용하여 /etc/chrony.conf 파일을 엽니다. 파일이 다음 라인을 포함하고 있는지 확인합니다.

```
server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
```

   이 라인이 존재할 경우, Amazon Time Sync Service가 이미 구성된 상태이기 때문에 다음 단계로 넘어갈 수 있습니다. 라인이 없는 경우에는 파일에 이미 존재하는 다른 server 또는 pool 문 뒤에 라인을 추가하고 변경 사항을 저장합니다.

4. chrony 데몬(chronyd)을 다시 시작합니다.

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

```
Starting chronyd:                                          [  OK  ]
```

   > **Note**
   >
   > RHEL 및 CentOS(최대 버전 6까지)에서 서비스 이름은 chrony이 아니라 chronyd입니다.

5. chkconfig 명령을 사용해서 매번 시스템이 부팅할 때마다 시작되도록 chronyd를 구성합니다.

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

6. chrony가 169.254.169.123 IP 주소를 사용하여 시간을 동기화하고 있는지 확인합니다.

```
[ec2-user ~]$ chronyc sources -v
```

```
210 Number of sources = 7

  .-- Source mode  '^' = server, '=' = peer, '#' = local clock.
 / .- Source state '*' = current synced, '+' = combined , '-' = not combined,
```

```
| /   '?' = unreachable, 'x' = time may be in error, '~' = time too variable.
||                                                 .- xxxx [ yyyy ] +/- zzzz
||      Reachability register (octal) -.           |  xxxx = adjusted offset,
||      Log2(Polling interval) --.      |          |  yyyy = measured offset,
||                                \     |          |  zzzz = estimated error.
||                                 |    |           \
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
^* 169.254.169.123               3   6    17    43    -30us[ -226us] +/-  287us
^- ec2-12-34-231-12.eu-west>     2   6    17    43   -388us[ -388us] +/-   11ms
^- tshirt.heanet.ie              1   6    17    44   +178us[  +25us] +/- 1959us
^? tbag.heanet.ie                0   6     0     -     +0ns[   +0ns] +/-    0ns
^? bray.walcz.net                0   6     0     -     +0ns[   +0ns] +/-    0ns
^? 2a05:d018:c43:e312:ce77:>     0   6     0     -     +0ns[   +0ns] +/-    0ns
^? 2a05:d018:dab:2701:b70:b>     0   6     0     -     +0ns[   +0ns] +/-    0ns
```

반환된 출력에서 ^*는 기본 설정된 타임 소스를 나타냅니다.

7. chrony에서 보고된 시간 동기화 지표를 확인합니다.

```
[ec2-user ~]$ chronyc tracking
```

```
Reference ID    : A9FEA97B (169.254.169.123)
Stratum         : 4
Ref time (UTC)  : Wed Nov 22 13:18:34 2017
System time     : 0.000000626 seconds slow of NTP time
Last offset     : +0.002852759 seconds
RMS offset      : 0.002852759 seconds
Frequency       : 1.187 ppm fast
Residual freq   : +0.020 ppm
Skew            : 24.388 ppm
Root delay      : 0.000504752 seconds
Root dispersion : 0.001112565 seconds
Update interval : 64.4 seconds
Leap status     : Normal
```

Ubuntu에서 Amazon Time Sync Service 구성

Amazon Time Sync Service에서 서버 항목을 추가하도록 chrony 구성 파일을 편집해야 합니다.

Amazon Time Sync Service를 사용하도록 인스턴스를 구성하려면

1. 인스턴스를 연결해 apt를 사용하여 chrony 패키지를 설치합니다.

```
ubuntu:~$ sudo apt install chrony
```

> **Note**
>
> 필요할 경우 sudo apt update를 실행하여 먼저 인스턴스를 업데이트합니다.

2. vim 또는 nano과 같은 텍스트 편집기를 사용하여 /etc/chrony/chrony.conf 파일을 엽니다. 파일에 이미 존재하는 server 또는 pool 문 앞에 다음 라인을 추가하고 변경 사항을 저장합니다.

```
server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
```

3. chrony 서비스를 다시 시작합니다.

```
ubuntu:~$ sudo /etc/init.d/chrony restart
```

```
[ ok ] Restarting chrony (via systemctl): chrony.service.
```

4. chrony가 169.254.169.123 IP 주소를 사용하여 시간을 동기화하고 있는지 확인합니다.

```
ubuntu:~$ chronyc sources -v
```

```
210 Number of sources = 7

  .-- Source mode  '^' = server, '=' = peer, '#' = local clock.
 / .- Source state '*' = current synced, '+' = combined , '-' = not combined,
| /   '?' = unreachable, 'x' = time may be in error, '~' = time too variable.
||                                                 .- xxxx [ yyyy ] +/- zzzz
||      Reachability register (octal) -.           |  xxxx = adjusted offset,
||      Log2(Polling interval) --.      |          |  yyyy = measured offset,
||                                \     |          |  zzzz = estimated error.
||                                 |    |           \
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
^* 169.254.169.123               3   6    17    12    +15us[  +57us] +/-  320us
^- tbag.heanet.ie                1   6    17    13  -3488us[-3446us] +/- 1779us
^- ec2-12-34-231-12.eu-west-     2   6    17    13   +893us[ +935us] +/- 7710us
^? 2a05:d018:c43:e312:ce77:6     0   6     0   10y     +0ns[   +0ns] +/-    0ns
^? 2a05:d018:d34:9000:d8c6:5     0   6     0   10y     +0ns[   +0ns] +/-    0ns
^? tshirt.heanet.ie              0   6     0   10y     +0ns[   +0ns] +/-    0ns
^? bray.walcz.net                0   6     0   10y     +0ns[   +0ns] +/-    0ns
```

반환된 출력에서 ^*는 기본 설정된 타임 소스를 나타냅니다.

5. chrony에서 보고된 시간 동기화 지표를 확인합니다.

```
ubuntu:~$ chronyc tracking
```

```
Reference ID    : 169.254.169.123 (169.254.169.123)
Stratum         : 4
Ref time (UTC)  : Wed Nov 29 07:41:57 2017
System time     : 0.000000011 seconds slow of NTP time
Last offset     : +0.000041659 seconds
RMS offset      : 0.000041659 seconds
Frequency       : 10.141 ppm slow
Residual freq   : +7.557 ppm
Skew            : 2.329 ppm
Root delay      : 0.000544 seconds
Root dispersion : 0.000631 seconds
Update interval : 2.0 seconds
Leap status     : Normal
```

SUSE Linux에서 Amazon Time Sync Service 구성

https://software.opensuse.org/package/chrony에서 chrony 설치

vim 또는 nano과 같은 텍스트 편집기를 사용하여 /etc/chrony.conf 파일을 엽니다. 파일이 다음 라인을 포함하고 있는지 확인합니다.

```
server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
```

이 라인이 표시되지 않는 경우, 추가합니다. 다른 서버 혹은 풀 라인 설명 yast 열고 chrony 서비스 실행

Amazon Linux에서 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.

> **Important**
>
> 이 정보는 Amazon Linux에 적용됩니다. 기타 배포에 대한 자세한 내용은 해당 설명서를 참조하십시오.

인스턴스의 표준 시간대를 변경하려면 다음을 수행합니다.

1. 인스턴스에서 사용할 표준 시간대를 식별합니다. `/usr/share/zoneinfo` 디렉터리에는 표준 시간대 데이터 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. 자주 사용하는 텍스트 편집기(vim 또는 nano)로 `/etc/sysconfig/clock` 파일을 엽니다. `/etc/sysconfig/clock`의 경우 `root` 소유이므로 편집기 명령으로 sudo를 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
   ```

CPU 옵션 최적화

Amazon EC2 인스턴스는 여러 개의 스레드를 하나의 CPU 코어에서 동시에 실행할 수 있도록 하는 멀티스레딩을 지원합니다. 각 스레드는 인스턴스에서 가상 CPU(vCPU)로 표현됩니다. 인스턴스에는 인스턴스 유형

에 따라 달라지는 기본 CPU 코어 수가 있습니다. 예를 들어 `m5.xlarge` 인스턴스 유형에는 기본적으로 2개의 CPU 코어와 코어당 2개의 스레드가 있어 vCPU는 총 4개입니다.—

> **Note**
>
> 각 vCPU는 T2 인스턴스용을 제외한 CPU 코어의 스레드입니다.

대부분의 경우 워크로드에 적합하도록 메모리와 vCPU 수를 결합한 Amazon EC2 인스턴스가 있습니다. 그러나 특정 워크로드 또는 비즈니스 필요를 위해 인스턴스를 최적화하는 다음 CPU 옵션을 지정할 수 있습니다.

- CPU 코어 수: 인스턴스에 대한 CPU 코어 수를 사용자 지정할 수 있습니다. 이를 통해 메모리 집약 워크로드용 RAM이 충분한 반면 CPU 코어를 적게 사용하는 인스턴스의 소프트웨어 라이선스 비용을 잠재적으로 최적화할 수 있습니다.
- 코어당 스레드: CPU 코어당 단일 스레드를 지정하여 멀티스레딩을 비활성화할 수 있습니다. HPC(고성능 컴퓨팅) 워크로드와 같은 특정 워크로드에 대해 이를 수행할 수 있습니다.

인스턴스 시작 중 이러한 CPU 옵션을 지정할 수 있습니다. CPU 옵션 지정에 따른 추가 요금이나 비용 경감은 없습니다. 기존 CPU 옵션으로 시작한 인스턴스와 동일하게 청구됩니다.

내용

- CPU 옵션 지정 규칙 (p. 547)
- 인스턴스 유형별/CPU당 CPU 코어 및 스레드 (p. 547)
- 인스턴스의 CPU 옵션 지정 (p. 559)
- 인스턴스의 CPU 옵션 보기 (p. 560)

CPU 옵션 지정 규칙

인스턴스의 CPU 옵션을 지정하려면 다음 규칙을 알아야 합니다.

- CPU 옵션은 인스턴스 시작 중에만 지정할 수 있으며 시작 후에는 수정할 수 없습니다.
- 인스턴스를 시작할 경우 요청에서 CPU 코어 수와 코어당 스레드를 모두 지정해야 합니다. 예제 요청은 인스턴스의 CPU 옵션 지정 (p. 559) 단원을 참조하십시오.
- 인스턴스의 vCPU 수는 코어당 스레드를 곱한 CPU 코어 수입니다. 사용자 지정 vCPU 수를 지정하려면 인스턴스 유형에 대해 유효한 CPU 코어 수와 코어당 스레드를 지정해야 합니다. 인스턴스의 기본 vCPU 수를 초과할 수 없습니다. 자세한 내용은 인스턴스 유형별/CPU당 CPU 코어 및 스레드 (p. 547) 단원을 참조하십시오.
- 멀티스레딩을 비활성화하려면 코어당 하나의 스레드를 지정하십시오.
- 기존 인스턴스의 인스턴스 유형을 변경 (p. 262)하면 CPU 옵션이 자동으로 새 인스턴스 유형의 기본 CPU 옵션으로 변경됩니다.
- 지정한 CPU 옵션은 인스턴스를 중지, 시작 또는 재부팅한 후에도 유지됩니다.

인스턴스 유형별/CPU당 CPU 코어 및 스레드

다음 표는 CPU 옵션 지정을 지원하는 인스턴스 유형을 나열합니다. 각 유형에 대해 표는 기본 및 지원되는 CPU 코어 수와 코어당 스레드를 표시합니다.

액셀러레이티드 컴퓨팅 인스턴스

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
`f1.2xlarge`	8	4	2	1, 2, 3, 4	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
f1.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
f1.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
g3.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
g3.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
g3.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
g3s.xlarge	4	2	2	1, 2	1, 2
g4dn.xlarge	4	2	2	1, 2	1, 2
g4dn.2xlarge	8	4	2	1, 2, 3, 4	1, 2
g4dn.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
g4dn.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
g4dn.12xlarge	48	24	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
g4dn.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
p2.xlarge	4	2	2	1, 2	1, 2
p2.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
p2.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
p3.2xlarge	8	4	2	1, 2, 3, 4	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
`p3.8xlarge`	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
`p3.16xlarge`	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
`p3dn.24xlarge`	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2

컴퓨팅 최적화 인스턴스

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
`c4.large`	2	1	2	1	1, 2
`c4.xlarge`	4	2	2	1, 2	1, 2
`c4.2xlarge`	8	4	2	1, 2, 3, 4	1, 2
`c4.4xlarge`	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
`c4.8xlarge`	36	18	2	2, 4, 6, 8, 10, 12, 14, 16, 18	1, 2
`c5.large`	2	1	2	1	1, 2
`c5.xlarge`	4	2	2	2	1, 2
`c5.2xlarge`	8	4	2	2, 4	1, 2
`c5.4xlarge`	16	8	2	2, 4, 6, 8	1, 2
`c5.9xlarge`	36	18	2	2, 4, 6, 8, 10, 12, 14, 16, 18	1, 2
`c5.12xlarge`	48	24	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
`c5.18xlarge`	72	36	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36	1, 2
`c5.24xlarge`	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20,	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
				22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	
`c5d.large`	2	1	2	1	1, 2
`c5d.xlarge`	4	2	2	2	1, 2
`c5d.2xlarge`	8	4	2	2, 4	1, 2
`c5d.4xlarge`	16	8	2	2, 4, 6, 8	1, 2
`c5d.9xlarge`	36	18	2	2, 4, 6, 8, 10, 12, 14, 16, 18	1, 2
`c5d.12xlarge`	48	24	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
`c5d.18xlarge`	72	36	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36	1, 2
`c5d.24xlarge`	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
`c5n.large`	2	1	2	1	1, 2
`c5n.xlarge`	4	2	2	2	1, 2
`c5n.2xlarge`	8	4	2	2, 4	1, 2
`c5n.4xlarge`	16	8	2	2, 4, 6, 8	1, 2
`c5n.9xlarge`	36	18	2	2, 4, 6, 8, 10, 12, 14, 16, 18	1, 2
`c5n.18xlarge`	72	36	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36	1, 2

범용 인스턴스

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
`m5.large`	2	1	2	1	1, 2
`m5.xlarge`	4	2	2	2	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m5.2xlarge	8	4	2	2, 4	1, 2
m5.4xlarge	16	8	2	2, 4, 6, 8	1, 2
m5.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
m5.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
m5.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
m5.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
m5a.large	2	1	2	1	1, 2
m5a.xlarge	4	2	2	2	1, 2
m5a.2xlarge	8	4	2	2, 4	1, 2
m5a.4xlarge	16	8	2	2, 4, 6, 8	1, 2
m5a.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
m5a.12xlarge	48	24	2	6, 12, 18, 24	1, 2
m5a.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
m5a.24xlarge	96	48	2	12, 18, 24, 36, 48	1, 2
m5ad.large	2	1	2	1	1, 2
m5ad.xlarge	4	2	2	2	1, 2
m5ad.2xlarge	8	4	2	2, 4	1, 2
m5ad.4xlarge	16	8	2	2, 4, 6, 8	1, 2
m5ad.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
m5ad.12xlarge	48	24	2	6, 12, 18, 24	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m5ad.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
m5ad.24xlarge	96	48	2	12, 18, 24, 36, 48	1, 2
m5d.large	2	1	2	1	1, 2
m5d.xlarge	4	2	2	2	1, 2
m5d.2xlarge	8	4	2	2, 4	1, 2
m5d.4xlarge	16	8	2	2, 4, 6, 8	1, 2
m5d.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
m5d.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
m5d.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
m5d.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
m5dn.large	2	1	2	1	1, 2
m5dn.xlarge	4	2	2	2	1, 2
m5dn.2xlarge	8	4	2	2, 4	1, 2
m5dn.4xlarge	16	8	2	2, 4, 6, 8	1, 2
m5dn.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
m5dn.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
m5dn.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m5dn.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
m5n.large	2	1	2	1	1, 2
m5n.xlarge	4	2	2	2	1, 2
m5n.2xlarge	8	4	2	2, 4	1, 2
m5n.4xlarge	16	8	2	2, 4, 6, 8	1, 2
m5n.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
m5n.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
m5n.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
m5n.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
t3.nano	2	1	2	1	1, 2
t3.micro	2	1	2	1	1, 2
t3.small	2	1	2	1	1, 2
t3.medium	2	1	2	1	1, 2
t3.large	2	1	2	1	1, 2
t3.xlarge	4	2	2	2	1, 2
t3.2xlarge	8	4	2	2, 4	1, 2
t3a.nano	2	1	2	1	1, 2
t3a.micro	2	1	2	1	1, 2
t3a.small	2	1	2	1	1, 2
t3a.medium	2	1	2	1	1, 2
t3a.large	2	1	2	1	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
`t3a.xlarge`	4	2	2	2	1, 2
`t3a.2xlarge`	8	4	2	2, 4	1, 2

메모리 최적화 인스턴스

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
`r4.large`	2	1	2	1	1, 2
`r4.xlarge`	4	2	2	1, 2	1, 2
`r4.2xlarge`	8	4	2	1, 2, 3, 4	1, 2
`r4.4xlarge`	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
`r4.8xlarge`	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
`r4.16xlarge`	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
`r5.large`	2	1	2	1	1, 2
`r5.xlarge`	4	2	2	2	1, 2
`r5.2xlarge`	8	4	2	2, 4	1, 2
`r5.4xlarge`	16	8	2	2, 4, 6, 8	1, 2
`r5.8xlarge`	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
`r5.12xlarge`	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
`r5.16xlarge`	64	32	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
`r5.24xlarge`	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
`r5a.large`	2	1	2	1	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r5a.xlarge	4	2	2	2	1, 2
r5a.2xlarge	8	4	2	2, 4	1, 2
r5a.4xlarge	16	8	2	2, 4, 6, 8	1, 2
r5a.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
r5a.12xlarge	48	24	2	6, 12, 18, 24	1, 2
r5a.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
r5a.24xlarge	96	48	2	12, 18, 24, 36, 48	1, 2
r5ad.large	2	1	2	1	1, 2
r5ad.xlarge	4	2	2	2	1, 2
r5ad.2xlarge	8	4	2	2, 4	1, 2
r5ad.4xlarge	16	8	2	2, 4, 6, 8	1, 2
r5ad.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
r5ad.12xlarge	48	24	2	6, 12, 18, 24	1, 2
r5ad.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
r5ad.24xlarge	96	48	2	12, 18, 24, 36, 48	1, 2
r5d.large	2	1	2	1	1, 2
r5d.xlarge	4	2	2	2	1, 2
r5d.2xlarge	8	4	2	2, 4	1, 2
r5d.4xlarge	16	8	2	2, 4, 6, 8	1, 2
r5d.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
r5d.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r5d.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
r5d.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
r5dn.large	2	1	2	1	1, 2
r5dn.xlarge	4	2	2	2	1, 2
r5dn.2xlarge	8	4	2	2, 4	1, 2
r5dn.4xlarge	16	8	2	2, 4, 6, 8	1, 2
r5dn.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
r5dn.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
r5dn.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
r5dn.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
r5n.large	2	1	2	1	1, 2
r5n.xlarge	4	2	2	2	1, 2
r5n.2xlarge	8	4	2	2, 4	1, 2
r5n.4xlarge	16	8	2	2, 4, 6, 8	1, 2
r5n.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
r5n.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r5n.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
r5n.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
x1.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
x1.32xlarge	128	64	2	4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64	1, 2
x1e.xlarge	4	2	2	1, 2	1, 2
x1e.2xlarge	8	4	2	1, 2, 3, 4	1, 2
x1e.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
x1e.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
x1e.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
x1e.32xlarge	128	64	2	4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64	1, 2
z1d.large	2	1	2	1	1, 2
z1d.xlarge	4	2	2	2	1, 2
z1d.2xlarge	8	4	2	2, 4	1, 2
z1d.3xlarge	12	6	2	2, 4, 6	1, 2
z1d.6xlarge	24	12	2	2, 4, 6, 8, 10, 12	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
z1d.12xlarge	48	24	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2

스토리지 최적화 인스턴스

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
d2.xlarge	4	2	2	1, 2	1, 2
d2.2xlarge	8	4	2	1, 2, 3, 4	1, 2
d2.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
d2.8xlarge	36	18	2	2, 4, 6, 8, 10, 12, 14, 16, 18	1, 2
h1.2xlarge	8	4	2	1, 2, 3, 4	1, 2
h1.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
h1.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
h1.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
i3.large	2	1	2	1	1, 2
i3.xlarge	4	2	2	1, 2	1, 2
i3.2xlarge	8	4	2	1, 2, 3, 4	1, 2
i3.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
i3.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
i3.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
i3en.large	2	1	2	1	1, 2
i3en.xlarge	4	2	2	2	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
`i3en.2xlarge`	8	4	2	2, 4	1, 2
`i3en.3xlarge`	12	6	2	2, 4, 6	1, 2
`i3en.6xlarge`	24	12	2	2, 4, 6, 8, 10, 12	1, 2
`i3en.12xlarge`	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
`i3en.24xlarge`	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2

인스턴스의 CPU 옵션 지정

인스턴스 시작 중 CPU 옵션을 지정할 수 있습니다. 다음 예는 `r4.4xlarge` 인스턴스 유형을 위한 것이며 다음과 같은 기본값 (p. 554)을 포함합니다.

- 기본 CPU 코어: 8
- 코어당 기본 스레드: 2
- 기본 vCPU: 16(8 * 2)
- 유효한 CPU 코어 수: 1, 2, 3, 4, 5, 6, 7, 8
- 코어당 유효한 스레드 수: 1, 2

멀티스레딩 비활성화

멀티스레딩을 비활성화하려면 코어당 하나의 스레드를 지정하십시오.

인스턴스 시작 중 멀티스레딩을 비활성화하려면(콘솔)

1. 인스턴스 시작 마법사를 사용하여 인스턴스 시작 (p. 431)의 절차를 따르십시오.
2. Configure Instance Details(인스턴스 정보 구성) 페이지에서 CPU options(CPU 옵션)에 대해 CPU 옵션 지정을 선택합니다.
3. 코어 수에 대해 필요한 CPU 코어 수를 선택합니다. 이 예에서 `r4.4xlarge` 인스턴스에 필요한 기본 CPU 코어 개수를 지정하려면 `8`을 선택합니다.
4. 멀티스레딩을 비활성화하려면 코어당 스레드로 1을 선택하십시오.
5. 마법사에 표시되는 지침에 따라 계속합니다. 인스턴스 시작 검토 페이지에서 옵션 검토를 마쳤으면 시작을 선택합니다. 자세한 내용은 인스턴스 시작 마법사를 사용하여 인스턴스 시작 (p. 431)를 참조하십시오.

인스턴스 시작 중 멀티스레딩을 비활성화하려면(AWS CLI)

run-instances AWS CLI 명령을 사용하여 `--cpu-options` 파라미터의 `ThreadsPerCore`에 값을 `1`로 지정합니다. `CoreCount`에 대해 CPU 코어 수를 지정합니다. 이 예에서 `r4.4xlarge` 인스턴스에 필요한 기본 CPU 코어 개수를 지정하려면 값을 `8`로 지정합니다.

```
aws ec2 run-instances --image-id ami-1a2b3c4d --instance-type r4.4xlarge --cpu-options
 "CoreCount=8,ThreadsPerCore=1" --key-name MyKeyPair
```

vCPU 수 사용자 지정

인스턴스의 코어당 CPU 코어와 스레드 수를 사용자 지정할 수 있습니다.

인스턴스 시작 중 사용자 지정 vCPU 수를 지정하려면(콘솔)

다음 예는 6개의 vCPU로 `r4.4xlarge` 인스턴스를 시작합니다.

1. 인스턴스 시작 마법사를 사용하여 인스턴스 시작 (p. 431)의 절차를 따르십시오.
2. Configure Instance Details(인스턴스 정보 구성) 페이지에서 CPU options(CPU 옵션)에 대해 CPU 옵션 지정을 선택합니다.
3. 6개의 vCPU를 얻기 위해 다음과 같이 3개의 CPU 코어와 코어당 2개의 스레드를 지정합니다.
   - 코어 수로 3을 선택합니다.
   - 코어당 스레드로 2를 선택합니다.
4. 마법사에 표시되는 지침에 따라 계속합니다. 인스턴스 시작 검토 페이지에서 옵션 검토를 마쳤으면 시작을 선택합니다. 자세한 내용은 인스턴스 시작 마법사를 사용하여 인스턴스 시작 (p. 431) 단원을 참조하십시오.

인스턴스 시작 중 vCPU 수를 사용자 지정하려면(AWS CLI)

다음 예는 6개의 vCPU로 `r4.4xlarge` 인스턴스를 시작합니다.

run-instances AWS CLI 명령을 사용하여 `--cpu-options` 파라미터에서 CPU 코어 수와 스레드 수를 지정합니다. 6개의 vCPU를 얻기 위해 3개의 CPU 코어와 코어당 2개의 스레드를 지정할 수 있습니다.

```
aws ec2 run-instances --image-id ami-1a2b3c4d --instance-type r4.4xlarge --cpu-options
 "CoreCount=3,ThreadsPerCore=2" --key-name MyKeyPair
```

또는 6개의 vCPU를 얻기 위해 6개의 CPU 코어와 코어당 하나의 스레드(멀티스레딩 비활성화)를 지정합니다.

```
aws ec2 run-instances --image-id ami-1a2b3c4d --instance-type r4.4xlarge --cpu-options
 "CoreCount=6,ThreadsPerCore=1" --key-name MyKeyPair
```

인스턴스의 CPU 옵션 보기

Amazon EC2 콘솔에서 또는 AWS CLI를 사용해 인스턴스를 설명하여 기존 인스턴스의 CPU 옵션을 볼 수 있습니다.

인스턴스의 CPU 옵션을 보려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 인스턴스를 선택하고 인스턴스를 선택합니다.
3. 설명을 선택하고 vCPU 수 필드를 확인합니다.
4. 코어 수와 코어당 스레드 수를 확인하려면 vCPU 수 필드 값을 선택합니다.

인스턴스의 CPU 옵션을 보려면(AWS CLI)

아래와 같이 describe-instances 명령을 사용합니다.

```
aws ec2 describe-instances --instance-ids i-123456789abcde123
```

```
...
    "Instances": [
        {
            "Monitoring": {
                "State": "disabled"
            },
            "PublicDnsName": "ec2-198-51-100-5.eu-central-1.compute.amazonaws.com",
            "State": {
                "Code": 16,
                "Name": "running"
            },
            "EbsOptimized": false,
            "LaunchTime": "2018-05-08T13:40:33.000Z",
            "PublicIpAddress": "198.51.100.5",
            "PrivateIpAddress": "172.31.2.206",
            "ProductCodes": [],
            "VpcId": "vpc-1a2b3c4d",
            "CpuOptions": {
                "CoreCount": 34,
                "ThreadsPerCore": 1
            },
            "StateTransitionReason": "",
            ...
        }
    ]
...
```

반환된 출력에서 `CoreCount` 필드는 인스턴스의 코어 수를 나타냅니다. `ThreadsPerCore` 필드는 코어당 스레드 수를 나타냅니다.

또는 인스턴스에 연결하고 lscpu와 같은 도구를 사용하여 인스턴스에 대한 CPU 정보를 봅니다.

AWS Config를 사용하여 종료된 인스턴스를 포함한 인스턴스의 구성 변경을 기록, 액세스, 감사 및 평가할 수 있습니다. 자세한 내용은 AWS Config Developer Guide의 AWS Config시작하기 를 참조하십시오.

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.internal`과 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 중지하고 다시 시작할 때마다 퍼블릭 IPv4 주소, 퍼블릭 DNS 이름, 시스템 호스트 이름 및 shell 프롬프트가 바뀝니다.

> **Important**
>
> 이 정보는 Amazon Linux에 적용됩니다. 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-12-34-56-78`과 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. • Amazon Linux 2: hostnamectl 명령으로 호스트 이름을 설정하여 정규화된 도메인 이름을 반영합니다(예: **webserver.mydomain.com**).

```
[ec2-user ~]$ sudo hostnamectl set-hostname webserver.mydomain.com
```

   • Amazon Linux AMI: 인스턴스에서 선호하는 텍스트 편집기로 `/etc/sysconfig/network` 구성 파일을 열고 `HOSTNAME` 항목을 변경하여 정규화된 도메인 이름을 반영합니다(예: **webserver.mydomain.com**).

```
HOSTNAME=webserver.mydomain.com
```

2. 인스턴스를 재부팅하여 새 호스트 이름을 적용합니다.

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

   또는 Amazon EC2 콘솔을 사용하여 재부팅할 수 있습니다(인스턴스 페이지에서 작업, 인스턴스 상태, 재부팅 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. • Amazon Linux 2: hostnamectl 명령으로 호스트 이름을 설정하여 원하는 시스템 호스트 이름을 반영합니다(예: **webserver**).

```
[ec2-user ~]$ sudo hostnamectl set-hostname webserver.localdomain
```

   • Amazon Linux AMI: 인스턴스에서 선호하는 텍스트 편집기로 `/etc/sysconfig/network` 구성 파일을 열고 `HOSTNAME` 항목을 변경하여 원하는 호스트 이름을 반영합니다(예: **webserver**).

```
HOSTNAME=webserver.localdomain
```

2. 선호하는 텍스트 편집기로 `/etc/hosts` 파일을 열고 **127.0.0.1**로 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(인스턴스 페이지에서 작업, 인스턴스 상태, 재부팅 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 **webserver**에서 제공한 프라이빗 이름(예: AWS)보다 더 유용한 시스템 이름(예: `ip-12-34-56-78`)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.d`에 `NICKNAME`이라는 환경 변수를 shell 프롬프트로 사용할 값으로 설정하는 파일을 생성합니다. 예를 들어 시스템 별칭을 **webserver**라고 설정하려면 다음 명령을 실행합니다.

   ```
   [ec2-user ~]$ sudo sh -c 'echo "export NICKNAME=webserver" > /etc/profile.d/prompt.sh'
   ```

2. 즐겨 찾는 텍스트 편집기(예: vim 또는 nano)에서 `/etc/bashrc`(Red Hat) 또는 `/etc/bash.bashrc`(Debian/Ubuntu) 파일을 엽니다. `/etc/bashrc` 및 `/etc/bash.bashrc`는 `root`가 소유하므로 sudo와 함께 편집기 명령을 사용해야 합니다.

3. 파일을 편집하여 호스트 이름 대신 별칭을 표시하도록 shell 프롬프트 변수(`PS1`)를 변경합니다. `/etc/bashrc` 또는 `/etc/bash.bashrc`에서 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]\\$ "
   # You might want to have e.g. tty in prompt (e.g. more virtual machines)
   # and console windows
   ```

   그 줄에서 `\h`(`hostname`에 대한 기호)를 `NICKNAME` 변수로 변경하십시오.

   ```
   # Turn on checkwinsize
   shopt -s checkwinsize
   [ "$PS1" = "\\s-\\v\\$ " ] && PS1="[\u@$NICKNAME \W]\\$ "
   # 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-xterm`이라는 이름의 파일을 만듭니다.

      ```
      [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. 선호하는 텍스트 편집기(예: vim 또는 nano)에서 `/etc/sysconfig/bash-prompt-xterm` 파일을 엽니다. `/etc/sysconfig/bash-prompt-xterm`는 `root`가 소유하므로 sudo와 함께 에디터 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

이 페이지의 절차는 Amazon Linux에서 사용하기 위한 것입니다. 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?

Your Linux 인스턴스에 동적 DNS 설정

EC2 인스턴스를 시작하면 인터넷에서 인스턴스에 접속하는 데 사용할 수 있는 퍼블릭 IP 주소와 퍼블릭 도메인 이름 시스템(DNS) 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.com`과 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 정보는 Amazon Linux에 적용됩니다. 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/free의 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.com에서 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/EPEL을 참조하십시오.

      - 대상 Amazon Linux 2:

      ```
      [ec2-user ~]$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
      ```

      - 대상 Amazon Linux AMI:

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

b. `noip` 패키지를 설치합니다.

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

c. 구성 파일을 생성합니다. 요청에 따라 로그인 및 암호 정보를 입력하고 후속 질문에 답하여 클라이언트를 구성합니다.

```
[ec2-user ~]$ sudo noip2 -C
```

3. noip 서비스를 활성화합니다.

   - 대상 Amazon Linux 2:

```
[ec2-user ~]$ sudo systemctl enable noip.service
```

   - 대상 Amazon Linux AMI:

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

4. noip 서비스를 시작합니다.

   - 대상 Amazon Linux 2:

```
[ec2-user ~]$ sudo systemctl start noip.service
```

   - 대상 Amazon Linux AMI:

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

   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 shell 스크립트 및 cloud-init 명령이라는 두 가지 유형의 사용자 데이터를 전달할 수 있습니다. 시작 마법사에 이 데이터를 일반 텍스트, 파일(명령줄 도구를 사용하여 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 Guide를 참조하십시오.

시작 시 Windows 인스턴스에서 명령을 실행하는 방법은 Windows 인스턴스용 Amazon EC2 사용 설명서에서 시작 시 Windows 인스턴스에서 명령 실행 및 Windows 인스턴스 구성 관리를 참조하십시오.

다음 예제에서는 Amazon Linux 2에 LAMP 웹 서버 설치 (p. 32)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 `mariadb` 패키지를 설치합니다.
- `httpd` 서비스를 시작하고 systemctl를 통해 활성화합니다.
- apache 그룹에 `ec2-user` 가 추가됩니다.
- 웹 디렉터리 및 해당 디렉터리에 들어 있는 파일에 적절한 소유권과 파일 권한을 설정합니다.
- 간단한 웹 페이지를 생성하여 웹 서버 및 PHP 엔진을 테스트합니다.

내용

- 사전 조건 (p. 566)
- 사용자 데이터 및 shell 스크립트 (p. 566)
- 사용자 데이터 및 콘솔 (p. 567)
- 사용자 데이터 및 cloud-init 명령 (p. 568)
- 사용자 데이터 및 AWS CLI (p. 569)

사전 조건

다음 예제에서는 인터넷에서 접속 가능한 퍼블릭 DNS 이름이 인스턴스에 지정되었다고 가정합니다. 자세한 내용은 1단계: 인스턴스 시작 (p. 28) 단원을 참조하십시오. 또한 SSH(포트 22), HTTP(포트 80), HTTPS(포트 443) 연결을 허용하도록 보안 그룹을 구성해야 합니다. 이 사전 조건에 대한 자세한 내용은 Amazon EC2로 설정 (p. 22) 단원을 참조하십시오.

또한 이러한 명령은 Amazon Linux 2에만 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. 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.

> **Important**
>
> 기본적으로 사용자 데이터 스크립트 및 cloud-init 명령은 최초로 인스턴스를 시작할 때만 실행됩니다. 인스턴스를 재시작할 때마다 사용자 데이터 스크립트 및 cloud-init 명령이 실행되도록 구성을 업데이트할 수 있습니다. 자세한 내용은 AWS Knowledge Center에서 How can I execute user data with every restart of my EC2 instance?(EC2 인스턴스를 재시작할 때마다 사용자 데이터를 실행하려면 어떻게 합니까?)를 참조하십시오.

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 단원을 참조하십시오.

사용자 데이터로 입력된 스크립트는 `root` 사용자 권한으로 실행되므로 스크립트에 sudo 명령을 사용하지 마십시오. 생성하는 모든 파일의 소유권은 `root`에 있습니다. 루트 이외의 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.

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.

사용자 데이터 스크립트는 처리 시 `/var/lib/cloud/instances/`*`instance-id`*`/`에서 복사 및 실행됩니다. 스크립트는 실행 후에는 삭제가 되지 않습니다. `/var/lib/cloud/instances/`*`instance-id`*`/`사용자 데이터를 파싱할 수 있는 스크립트가 AMI 인스턴스에 필요하다고 언급했습니다. 그렇지 않은 경우, 스크립트는 AMI에서 시작된 어떤 인스턴스에서든 이 디렉터리에 존재합니다.

사용자 데이터 및 콘솔

인스턴스를 시작할 때 인스턴스 사용자 데이터를 지정할 수 있습니다. 인스턴스의 루트 볼륨이 EBS 볼륨이면 인스턴스를 중지하고 사용자 데이터를 업데이트할 수도 있습니다.

시작 시 인스턴스 사용자 데이터 지정

AMI에서 인스턴스 시작 (p. 432)의 인스턴스 시작 절차를 따르되, 해당 절차의 the section called “3단계: 인스턴스 세부 정보 구성” (p. 433)에 이르면 사용자 데이터(User data) 필드에 shell 스크립트를 복사하고 나서 시작 절차를 완료합니다.

아래 예제 스크립트에서는 스크립트를 통해 웹 서버를 생성하고 구성합니다.

```
#!/bin/bash
yum update -y
amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
yum install -y httpd mariadb-server
systemctl start httpd
systemctl enable httpd
usermod -a -G apache ec2-user
chown -R ec2-user:apache /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
```

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. PHP 정보 페이지가 표시되지 않는 경우 사용하고 있는 보안 그룹이 HTTP(포트 80) 트래픽을 허용하는 규칙을 포함하고 있는지 확인하십시오. 자세한 내용은 보안 그룹에 규칙 추가 (p. 877) 단원을 참조하십시오.

(선택 사항) 스크립트에서 의도한 작업을 완료하지 못했거나 스크립트가 오류 없이 완료되었는지 여부를 확인하려는 경우 `/var/log/cloud-init-output.log`의 cloud-init 출력 로그 파일을 조사하여 출력에서 오류 메시지를 찾아 봅니다.

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 형식을 참조하십시오.

인스턴스 사용자 데이터 보기 및 업데이트

인스턴스 사용자 데이터를 수정하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 작업, 인스턴스 상태, 중지를 선택합니다.

Warning

인스턴스를 중지하면 인스턴스 스토어 볼륨의 데이터가 삭제됩니다. 인스턴스 스토어 볼륨의 데이터를 유지하려면 영구 스토리지에 백업하십시오.

4. 확인 메시지가 표시되면 예, 중지를 선택합니다. 인스턴스가 중지하는 데 몇 분 정도 걸릴 수 있습니다.
5. 인스턴스가 선택된 상태에서 작업을 선택하고 인스턴스 설정을 선택한 후 사용자 데이터 보기/변경을 선택합니다. 인스턴스가 실행 중일 때는 사용자 데이터를 변경할 수 없습니다. 하지만 볼 수는 있습니다.
6. 사용자 데이터 보기/변경 대화 상자에서 사용자 데이터를 업데이트하고 저장을 선택합니다.
7. 인스턴스를 다시 시작합니다. 새 사용자 데이터는 인스턴스를 다시 시작한 후에 인스턴스에 표시되지만 사용자 데이터 스크립트가 실행되지는 않습니다.

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-user의 `.ssh/authorized_keys` 파일을 구성하는 것입니다. 자세한 내용은 cloud-init (p. 166) 단원을 참조하십시오.

cloud-init 사용자 명령을 인스턴스 시작 시에 전달하는 방법은 스크립트를 전달하는 방법과 동일하지만 구문은 서로 다릅니다. cloud-init에 대한 자세한 내용은 http://cloudinit.readthedocs.org/en/latest/index.html을 참조하십시오.

Important

기본적으로 사용자 데이터 스크립트 및 cloud-init 명령은 최초로 인스턴스를 시작할 때만 실행됩니다. 인스턴스를 재시작할 때마다 사용자 데이터 스크립트 및 cloud-init 명령이 실행되도록 구성을 업데이트할 수 있습니다. 자세한 내용은 AWS Knowledge Center에서 How can I execute user data with every restart of my EC2 instance?(EC2 인스턴스를 재시작할 때마다 사용자 데이터를 실행하려면 어떻게 합니까?)를 참조하십시오.

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. 432)의 인스턴스 시작 절차를 따르되, 해당 절차의 the section called “3단계: 인스턴스 세부 정보 구성” (p. 433)에 이르면 사용자 데이터(User data) 필드에 cloud-init 명령 텍스트를 입력한 다음 시작 절차를 완료합니다.

   아래 예제에서는 명령을 통해 Amazon Linux 2에서 웹 서버를 생성하고 구성합니다. 명령을 cloud-init 명령으로 식별하려면 상단에 `#cloud-config` 행을 추가해야 합니다.

```
#cloud-config
repo_update: true
repo_upgrade: all

packages:
 - httpd
 - mariadb-server

runcmd:
 - [ sh, -c, "amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2" ]
 - systemctl start httpd
 - sudo systemctl enable httpd
 - [ sh, -c, "usermod -a -G apache ec2-user" ]
 - [ sh, -c, "chown -R ec2-user:apache /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. PHP 정보 페이지가 표시되지 않는 경우 사용하고 있는 보안 그룹이 HTTP(포트 80) 트래픽을 허용하는 규칙을 포함하고 있는지 확인하십시오. 자세한 내용은 보안 그룹에 규칙 추가 (p. 877) 단원을 참조하십시오.
3. (선택 사항) 명령에서 의도한 작업을 완료하지 못했거나 명령이 오류 없이 완료되었는지 여부를 확인하려는 경우 `/var/log/cloud-init-output.log`의 출력 로그 파일을 조사하여 출력에서 오류 메시지를 찾아 봅니다. 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.

사용자 데이터 및 AWS CLI

AWS CLI를 사용하여 인스턴스의 사용자 데이터를 지정, 수정 및 확인할 수 있습니다. 인스턴스 메타데이터를 사용하여 인스턴스의 사용자 데이터를 보는 방법에 대한 자세한 내용은 인스턴스 사용자 데이터 가져오기 (p. 584) 단원을 참조하십시오.

Windows에서 AWS CLI를 사용하는 대신 Windows PowerShell용 AWS 도구을 사용할 수 있습니다. 자세한 내용은 Windows 인스턴스용 Amazon EC2 사용 설명서의 사용자 데이터 및 Windows PowerShell용 도구을 참조하십시오.

예: 시작 시 사용자 데이터 지정

인스턴스를 시작할 때 사용자 데이터를 지정하려면 run-instances 명령을 `--user-data` 파라미터와 함께 사용합니다. run-instances를 사용하면 AWS CLI에서는 사용자 데이터의 base64 인코딩을 수행합니다.

다음 예에서는 스크립트를 명령줄에서 문자열로 지정하는 방법을 보여줍니다.

```
aws ec2 run-instances --image-id ami-abcd1234 --count 1 --instance-type m3.medium \
--key-name my-key-pair --subnet-id subnet-abcd1234 --security-group-ids sg-abcd1234 \
--user-data echo user data
```

다음 예에서는 텍스트 파일을 사용하여 스크립트를 지정하는 방법을 보여줍니다. `file://` 접두사를 사용하여 파일을 지정해야 합니다.

```
aws ec2 run-instances --image-id ami-abcd1234 --count 1 --instance-type m3.medium \
--key-name my-key-pair --subnet-id subnet-abcd1234 --security-group-ids sg-abcd1234 \
--user-data file://my_script.txt
```

다음은 shell 스크립트가 포함된 텍스트 파일의 예입니다.

```
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

예: 중지된 인스턴스의 사용자 데이터 수정

modify-instance-attribute 명령을 사용하여 중지된 인스턴스의 사용자 데이터를 수정할 수 있습니다. modify-instance-attribute를 사용하여 AWS CLI는 사용자 데이터의 base64 인코딩을 수행하지 않습니다.

- Linux 컴퓨터에서 base64 명령을 사용하여 사용자 데이터를 인코딩합니다.

```
base64 my_script.txt >my_script_base64.txt
```

- Windows 컴퓨터에서 certutil 명령을 사용하여 사용자 데이터를 인코딩합니다. AWS CLI에서 이 파일을 사용하기 전에 첫 번째(인증서 시작) 줄과 마지막(인증서 종료) 줄을 제거해야 합니다.

```
certutil -encode my_script.txt my_script_base64.txt
notepad my_script_base64.txt
```

`--attribute` 및 `--value` 파라미터를 사용하여 인코딩된 텍스트 파일로 사용자 데이터를 지정합니다. `file://` 접두사를 사용하여 파일을 지정해야 합니다.

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --attribute userData --
value file://my_script_base64.txt
```

예: 사용자 데이터 보기

인스턴스의 사용자 데이터를 가져오려면 describe-instance-attribute 명령을 사용합니다. describe-instance-attribute를 사용하여 AWS CLI는 사용자 데이터의 base64 디코딩을 수행하지 않습니다.

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --attribute userData
```

다음은 사용자 데이터 base64가 인코딩된 출력의 예입니다.

```
{
    "UserData": {
        "Value":
 "IyEvYmluL2Jhc2gKeXVtIHVwZGF0ZSAteQpzZXJ2aWNlIGh0dHBkIHN0YXJ0CmNoa2NvbmZpZyBodHRwZCBvbg=="
    },
    "InstanceId": "i-1234567890abcdef0"
}
```

- Linux 컴퓨터에서 `--query` 옵션을 사용하여 인코딩된 사용자 데이터를 가져오고, base64 명령을 사용하여 이를 디코딩합니다.

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --attribute
 userData --output text --query "UserData.Value" | base64 --decode
```

- Windows 컴퓨터에서 `--query` 옵션을 사용하여 코딩된 사용자 데이터를 가져오고, certutil 명령을 사용하여 이를 디코딩합니다. 인코딩된 출력과 디코딩된 출력은 각각 다른 파일에 저장됩니다.

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --attribute
 userData --output text --query "UserData.Value" >my_output.txt
certutil -decode my_output.txt my_output_decoded.txt
type my_output_decoded.txt
```

다음은 예제 출력입니다.

```
#!/bin/bash
```

```
yum update -y
service httpd start
chkconfig httpd on
```

인스턴스 메타데이터 및 사용자 데이터

인스턴스 메타데이터는 실행 중인 인스턴스를 구성 또는 관리하는 데 사용될 수 있는 인스턴스 관련 데이터입니다. 인스턴스 메타데이터는 예를 들어 호스트 이름, 이벤트 및 보안 그룹과 같은 범주로 분류됩니다.

인스턴스 메타데이터를 사용하여 인스턴스를 시작할 때 지정한 사용자 데이터에도 액세스할 수 있습니다. 예를 들어, 인스턴스를 구성하기 위한 파라미터를 지정하거나 단순 스크립트를 포함시킬 수 있습니다. 일반 AMI를 빌드하고 사용자 데이터를 사용하여 시작 시간에 제공되는 구성 파일을 수정할 수 있습니다. 예를 들어, 다양한 소규모 비즈니스용으로 웹 서버를 운영하는 경우 모두 동일한 일반 AMI를 사용하고 시작 시 사용자 데이터에서 지정하는 Amazon S3 버킷에서 콘텐츠를 가져올 수 있습니다. 언제든지 새 고객을 추가하려면 고객에 대한 버킷을 생성하고, 콘텐츠를 추가한 다음, 사용자 데이터에서 코드에 제공된 고유의 버킷 이름으로 AMI를 시작합니다. 1개 이상의 인스턴스를 동시에 시작하는 경우 해당 동일 시작 인스턴스의 모든 인스턴스에서 사용자 데이터를 이용할 수 있습니다. 동일한 예약에 속한 각 인스턴스에는 수행할 작업을 제어하는 코드를 작성할 수 있도록 고유의 `ami-launch-index` 번호가 있습니다. 예를 들어, 첫 번째 호스트는 클러스터의 초기 마스터 노드로 자체 선택될 수 있습니다. 자세한 AMI 시작 예는 예제: AMI 시작 색인 값 (p. 585) 단원을 참조하십시오.

또한, EC2 인스턴스에는 인스턴스가 시작되었을 때 생성되는 인스턴스 자격 증명 문서와 같은 동적 데이터가 포함됩니다. 자세한 내용은 동적 데이터 카테고리 (p. 593) 단원을 참조하십시오.

> **Important**
>
> 사용자는 인스턴스 자체 내에서 인스턴스 메타데이터 및 사용자 데이터에만 액세스할 수 있지만, 데이터는 인증 또는 암호화 방법으로 보호되지 않습니다. 인스턴스에 직접 액세스할 수 있는 모든 사람과 인스턴스에서 실행 중인 모든 소프트웨어는 메타데이터를 볼 수 있습니다. 따라서 암호 또는 수명이 긴 암호화 키와 같은 민감한 데이터를 사용자 데이터로 저장해서는 안 됩니다.

목차

- 인스턴스 메타데이터 서비스 구성 (p. 571)
- 인스턴스 메타데이터 가져오기 (p. 576)
- 인스턴스 사용자 데이터로 작업 (p. 583)
- 동적 데이터 가져오기 (p. 585)
- 예제: AMI 시작 색인 값 (p. 585)
- 인스턴스 메타데이터 카테고리 (p. 588)
- 인스턴스 자격 증명 문서 (p. 593)

인스턴스 메타데이터 서비스 구성

다음 방법 중 하나를 사용하여 실행 중인 인스턴스에서 인스턴스 메타데이터에 액세스할 수 있습니다.

- 인스턴스 메타데이터 서비스 버전 1(IMDSv1) – 요청/응답 방법
- 인스턴스 메타데이터 서비스 버전 2(IMDSv2) – 세션 지향 방법

기본적으로 IMDSv1 또는 IMDSv2를 사용하거나 둘 다 사용할 수 있습니다. 인스턴스 메타데이터 서비스는 특정 요청에 대해 IMDSv2에 고유한 `PUT` 또는 `GET` 헤더가 해당 요청에 있는지 여부에 따라 IMDSv1 및 IMDSv2 요청 간을 구별합니다.

로컬 코드 또는 사용자가 IMDSv2를 사용해야 하도록 각 인스턴스에서 인스턴스 메타데이터 서비스를 구성할 수 있습니다. IMDSv2를 사용해야 하도록 지정하면 IMDSv1는 더 이상 작동하지 않습니다. 자세한 내용은 인스턴스 메타데이터 옵션 구성 (p. 574) 단원을 참조하십시오.

인스턴스 메타데이터 서비스 버전 2 작동 방식

IMDSv2는 세션 지향 요청을 사용합니다. 세션 지향 요청을 사용하여 세션 기간을 정의하는 세션 토큰을 생성합니다. 세션 기간은 최소 1초에서 최대 6시간일 수 있습니다. 지정된 기간 중에는 후속 요청에 동일한 세션 토큰을 사용할 수 있습니다. 지정된 기간이 만료된 후에는 향후 요청에 사용할 새로운 세션 토큰을 생성할 수 있습니다.

다음 예제에서는 Linux 셸 스크립트와 IMDSv2를 사용하여 최상위 인스턴스 메타데이터 항목을 가져옵니다. 예제 명령:

- `PUT` 요청을 사용하여 6시간(21,600초) 동안 지속되는 세션 토큰을 생성합니다.
- `TOKEN`이라는 변수에 세션 토큰 헤더를 저장합니다.
- 토큰을 사용하여 최상위 메타데이터 항목을 요청합니다.

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/
```

토큰을 생성한 후에는 만료될 때까지 토큰을 재사용할 수 있습니다. 다음 예제 명령에서는 인스턴스를 시작하는 데 사용한 AMI의 ID를 가져오고 이전 예에서 `$TOKEN`에 저장한 토큰을 재사용합니다.

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/
meta-data/ami-id
```

IMDSv2를 사용하여 인스턴스 메타데이터를 요청하는 경우 요청에는 다음이 포함되어야 합니다.

1. `PUT` 요청을 사용하여 인스턴스 메타데이터 서비스의 세션을 초기화합니다. `PUT` 요청은 후속 `GET` 요청에 포함되어야 하는 토큰을 인스턴스 메타데이터 서비스에 반환합니다. 토큰은 IMDSv2를 사용하여 메타데이터에 액세스하는 데 필요합니다.
2. 인스턴스 메타데이터 서비스에 대한 모든 `GET` 요청에 토큰을 포함시킵니다. 토큰 사용이 `required`로 설정되면 유효한 토큰이 없거나 만료된 토큰이 있는 요청은 `401 - Unauthorized` HTTP 오류 코드를 수신합니다. 토큰 사용 요구 사항을 변경하는 방법에 대한 자세한 내용은 AWS CLI Command Reference의 modify-instance-metadata-options를 참조하십시오.
   - 토큰은 인스턴스에 특정한 키입니다. 토큰은 다른 EC2 인스턴스에서 유효하지 않으며 해당 토큰이 생성된 인스턴스 외부에서 사용하려고 시도하면 거부됩니다.
   - `PUT` 요청에는 토큰의 TTL(Time to Live)를 최대 6시간(21,600초)까지 초 단위로 지정하는 헤더가 포함되어야 합니다. 토큰은 논리 세션을 나타냅니다. TTL은 토큰이 유효한 시간 길이를 지정하며 따라서 세션 기간을 지정합니다.
   - 토큰이 만료된 후 인스턴스 메타데이터에 계속 액세스하려면 다른 `PUT`를 사용하여 새 세션을 생성해야 합니다.
   - 토큰을 재사용하거나 모든 요청에 새 토큰을 생성하도록 선택할 수 있습니다. 요청 수가 적은 경우 인스턴스 메타데이터 서비스에 액세스해야 할 때마다 토큰을 생성하고 즉시 사용하는 것이 더 간편할 수 있습니다. 하지만 효율성을 향상하려면 인스턴스 메타데이터를 요청해야 할 때마다 `PUT` 요청을 작성하는 대신 토큰에 더 긴 기간을 지정하고 토큰을 재사용할 수 있습니다. 각각 고유의 세션을 나타내는 동시 토큰 수에는 실질적인 제한이 없습니다. 하지만 IMDSv2는 표준 인스턴스 메타데이터 서비스 연결 및 조절 제한에 따라 여전히 제약됩니다. 자세한 내용은 Throttling (p. 582) 단원을 참조하십시오.

IMDSv2 인스턴스 메타데이터 요청에서는 HTTP `GET` 및 `HEAD` 메서드가 허용됩니다. `PUT` 요청은 X-Forwarded-For 헤더가 포함될 경우 거부됩니다.

기본적으로 `PUT` 요청에 대한 응답에는 IP 프로토콜 수준에서 `1`의 응답 홉 제한(TTL(Time to Live))이 있습니다. 홉 제한을 더 크게 만들어야 하는 경우 `modify-instance-metadata-options` 명령을 사용하여 홉 제한을 조정할 수 있습니다. 예를 들어, 인스턴스에서 실행 중인 컨테이너 서비스가 있는 경우 이전 버전

과의 호환성을 위해 더 큰 홉 제한이 필요할 수 있습니다. 자세한 내용은 AWS CLI Command Reference의 modify-instance-metadata-options를 참조하십시오.

인스턴스 메타데이터 서비스 버전 2 사용으로 전환

인스턴스 메타데이터 서비스 버전 2(IMDSv2) 사용은 선택적입니다. 인스턴스 메타데이터 서비스 버전 1(IMDSv1)은 계속해서 무기한 지원됩니다. IMDSv2 사용으로 마이그레이션하도록 선택하는 경우 다음과 같은 도구와 전환 경로를 사용하는 것이 좋습니다.

IMDSv2로 전환하는 데 도움이 되는 도구

소프트웨어가 IMDSv1을 사용하는 경우 다음 도구를 사용하면 IMDSv2를 사용하도록 소프트웨어를 재구성하는 데 도움이 됩니다.

- AWS 소프트웨어: 최신 버전의 AWS SDK 및 CLI는 IMDSv2를 지원합니다. IMDSv2를 사용하려면 EC2 인스턴스에 최신 버전의 AWS SDK 및 CLI가 있는지 확인합니다. CLI 업데이트에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 최신 버전의 AWS CLI로 업그레이드를 참조하십시오.
- CloudWatch: IMDSv2는 토큰 지원 세션을 사용하지만 IMDSv1은 사용하지 않습니다. `MetadataNoToken` CloudWatch 지표는 IMDSv1을 사용하는 인스턴스 메타데이터 서비스에 대한 호출 수를 추적합니다. 이 지표를 0까지 추적하면 모든 소프트웨어가 IMDSv2를 사용하도록 업그레이드되었는지 여부와 업그레이드된 시간을 확인할 수 있습니다. 자세한 내용은 인스턴스 지표 (p. 618) 단원을 참조하십시오.
- EC2 API 및 CLI로 업데이트: 기존 인스턴스의 경우 modify-instance-metadata-options CLI 명령(또는 ModifyInstanceMetadataOptions API)을 사용하여 IMDSv2를 사용해야 하도록 설정할 수 있습니다. 새 인스턴스의 경우 run-instances CLI 명령(또는 RunInstances API)과 `metadata-options` 파라미터를 사용하여 IMDSv2를 사용해야 하는 새 인스턴스를 시작할 수 있습니다.

  Auto Scaling 그룹에서 시작한 모든 새 인스턴스에서 IMDSv2를 사용해야 하는 경우 Auto Scaling 그룹에서 시작 템플릿을 사용해야 합니다. 시작 템플릿을 생성할 때 IMDSv2를 사용해야 하도록 `MetadataOptions` 파라미터를 구성합니다. 시작 구성을 사용하는 Auto Scaling 그룹의 경우 시작 구성을 시작 템플릿으로 바꿉니다. 시작 구성을 시작 템플릿으로 바꾸면 Auto Scaling 그룹이 새 시작 템플릿을 사용하여 새 인스턴스를 시작하지만, 기존 인스턴스에는 영향을 주지 않습니다. modify-instance-metadata-options CLI 명령(또는 ModifyInstanceMetadataOptions API)을 사용하여 기존 인스턴스에서 IMDSv2를 사용하도록 하거나 인스턴스를 종료하고 Auto Scaling 그룹이 시작 템플릿에서 정의한 인스턴스 메타데이터 옵션 설정으로 새 교체 인스턴스를 시작하도록 합니다.
- IAM 정책 및 SCP: IAM 조건을 통해 IAM 사용자가 IMDSv2를 사용하지 않는 인스턴스를 시작하지 못하도록 설정할 수 있습니다. 또한 IAM 조건을 사용하여 IAM 사용자가 IMDSv1을 다시 활성화하기 위해 실행 중인 인스턴스를 수정할 수 없도록 설정할 수 있으며, 인스턴스에서 인스턴스 메타데이터 서비스를 사용할 수 있도록 설정할 수 있습니다.

  `ec2:MetadataHttpTokens`, `ec2:MetadataHttpPutResponseHopLimit` 및 `ec2:MetadataHttpEndpoint` IAM 조건 키를 사용하여 RunInstances 및 ModifyInstanceMetadataOptions API와 해당 CLI 사용을 제어할 수 있습니다. 정책이 생성되고 API 호출의 파라미터가 조건 키를 사용하는 정책에 지정된 상태와 일치하지 않으면 API 또는 CLI 호출이 `UnauthorizedOperation` 응답과 함께 실패합니다. 이러한 조건 키는 IAM 정책 또는 AWS Organizations 서비스 제어 정책(SCP)에서 사용할 수 있습니다.

  추가로, IMDSv1에서 IMDSv2로 변경을 시행하기 위한 추가 보호 계층을 선택할 수 있습니다. EC2 역할 자격 증명을 통해 호출되는 API에 관한 액세스 관리 계층에서는 IAM 정책 또는 AWS Organizations 서비스 제어 정책(SCP)에서 새 조건 키를 사용할 수 있습니다. 특히, IAM 정책에서 값이 `2.0`인 정책 조건 키 `ec2:RoleDelivery`를 사용하여 IMDSv1에서 얻은 EC2 역할 자격 증명으로 API 호출을 수행하면 `UnauthorizedOperation` 응답이 수신됩니다. SCP에 따라 필요한 조건을 사용하여 동일한 작업을 더 광범위하게 수행할 수 있습니다. 이렇게 하면 지정된 조건에 맞지 않게 API를 호출할 경우 `UnauthorizedOperation` 오류가 수신되기 때문에 실제로 IMDSv1을 통해 제공된 자격 증명을 사용하여 API를 호출할 수 없습니다. 예제 IAM 정책은 인스턴스 메타데이터 작업 (p. 840) 단원을 참조하십시오. 자세한 내용은 AWS Organizations 사용 설명서의 서비스 제어 정책을 참조하십시오.

IMDSv2에 액세스해야 하도록 설정하는 권장 경로

위의 도구를 사용하여 이 경로를 따라 IMDSv2로 전환하는 것이 좋습니다.

1단계: 시작 시

EC2 인스턴스에서 역할 자격 증명을 사용하는 SDK, CLI 및 소프트웨어를 IMDSv2 호환 버전으로 업데이트합니다. CLI 업데이트에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 최신 버전의 AWS CLI로 업그레이드를 참조하십시오.

그런 다음, IMDSv2 요청을 사용하여 인스턴스 메타데이터에 직접 액세스하는(다시 말해서, SDK를 사용하지 않는) 소프트웨어를 변경합니다.

2단계: 전환 도중

CloudWatch 지표 `MetadataNoToken`을 사용하여 전환 진행률을 추적합니다. 이 지표는 인스턴스에서 IMDSv1을 사용하는 인스턴스 메타데이터 서비스에 대한 호출 수를 표시합니다. 자세한 내용은 인스턴스 지표 (p. 618) 단원을 참조하십시오.

3단계: 모든 인스턴스에서 모든 준비가 완료될 때

CloudWatch 지표 `MetadataNoToken`이 IMDSv1 사용량 0을 기록하면 모든 인스턴스에서 모든 준비가 완료됩니다. 이 단계에서 다음 작업을 수행할 수 있습니다.

- 기존 인스턴스의 경우: modify-instance-metadata-options 명령을 통해 IMDSv2를 사용해야 하도록 설정할 수 있습니다. 실행 중인 인스턴스에서 이렇게 변경할 수 있으며, 인스턴스를 다시 시작할 필요는 없습니다.
- 새 인스턴스의 경우: 새 인스턴스를 시작할 때 run-instances 명령을 사용하여 IMDSv2만 사용하도록 지정할 수 있습니다.

API 또는 AWS CLI를 통해서만 인스턴스 메타데이터 옵션을 지정할 수 있으며, 현재는 AWS Management 콘솔을 통해 이 옵션을 지정할 수 없습니다. 자세한 내용은 인스턴스 메타데이터 옵션 구성 (p. 574) 단원을 참조하십시오.

4단계: 모든 인스턴스가 IMDSv2로 전환될 때

`ec2:MetadataHttpTokens`, `ec2:MetadataHttpPutResponseHopLimit` 및 `ec2:MetadataHttpEndpoint` IAM 조건 키를 사용하여 RunInstances 및 ModifyInstanceMetadataOptions API와 해당 CLI 사용을 제어할 수 있습니다. 정책이 생성되고 API 호출의 파라미터가 조건 키를 사용하는 정책에 지정된 상태와 일치하지 않으면 API 또는 CLI 호출이 `UnauthorizedOperation` 응답과 함께 실패합니다. 예제 IAM 정책은 인스턴스 메타데이터 작업 (p. 840) 단원을 참조하십시오.

인스턴스 메타데이터 옵션 구성

인스턴스 메타데이터 옵션을 통해 신규 또는 기존 인스턴스를 구성하여 다음과 같이 설정할 수 있습니다.

- 인스턴스 메타데이터를 요청할 때 IMDSv2를 사용해야 하도록 설정
- `PUT` 응답 홉 제한 지정
- 인스턴스 메타데이터에 대한 액세스 비활성화

또한 IAM 정책 또는 SCP에서 IAM 조건 키를 사용하여 다음과 같이 설정할 수 있습니다.

- IMDSv2를 사용해야 하도록 구성된 경우에만 인스턴스를 시작하도록 허용
- 허용된 홉 수 제한
- 인스턴스 메타데이터에 대한 액세스 비활성화

신규 또는 기존 인스턴스에서 인스턴스 메타데이터 옵션을 구성하려면 AWS SDK 또는 CLI를 사용합니다. 자세한 내용은 AWS CLI Command Reference의 run-instances 및 modify-instance-metadata-options를 참조하십시오.

Note

작업을 조심스럽게 진행해야 하며 무엇이든 변경하기 전에 세심하게 테스트해야 합니다. 다음에 유의하십시오.

- IMDSv2를 사용해야 하도록 설정하면 인스턴스 메타데이터 액세스에 IMDSv1를 사용하는 애플리케이션이나 에이전트는 중단됩니다.
- 인스턴스 메타데이터에 대한 모든 액세스를 끄면 인스턴스 메타데이터 액세스를 이용하여 작동하는 애플리케이션이나 에이전트는 중단됩니다.

주제

- 새 인스턴스에 대한 인스턴스 메타데이터 옵션 구성 (p. 575)
- 기존 인스턴스에 대한 인스턴스 메타데이터 옵션 구성 (p. 575)

새 인스턴스에 대한 인스턴스 메타데이터 옵션 구성

인스턴스를 시작할 때 인스턴스에서 IMDSv2를 사용해야 하도록 설정할 수 있습니다. 또한 새 인스턴스에서 IMDSv2를 사용해야 하도록 설정되지 않은 경우 사용자가 새 인스턴스를 시작하지 못하게 하는 IAM 정책을 생성할 수 있습니다.

새 인스턴스에서 IMDSv2를 사용해야 하도록 설정하려면

다음 run-instances 예제에서는 `metadata-options`를 `HttpTokens=required`로 설정하여 `c3.large` 인스턴스를 시작합니다. 메타데이터 검색 요청에 대해 보안 토큰 헤더가 `required`로 설정되어 있으므로 인스턴스 메타데이터를 요청할 때 인스턴스가 IMDSv2를 사용해야 합니다.

Note

- 또한 `HttpTokens`의 값을 지정할 때 `HttpEndpoint`를 `enabled`로 설정해야 합니다.
- 이 예제에는 `--count` 및 `--security-group` 파라미터가 포함되어 있지 않습니다. `--count`의 기본값은 `1`입니다. 기본 VPC와 기본 보안 그룹이 있는 경우 이들이 사용됩니다.

```
aws ec2 run-instances \
    --image-id ami-1a2b3c4d \
    --instance-type c3.large \
    --key-name MyKeyPair \
    --metadata-options "HttpEndpoint=enabled,HttpTokens=required"
```

모든 새 인스턴스에서 IMDSv2를 사용해야 하도록 설정하려면

IAM 사용자가 인스턴스 메타데이터 요청 시 IMDSv2를 사용해야 하는 인스턴스만 시작할 수 있도록 하려면 IMDSv2를 사용해야 한다는 조건을 충족한 후에만 인스턴스를 시작할 수 있도록 지정할 수 있습니다. IAM 정책 예제는 인스턴스 메타데이터 작업 (p. 840) 단원을 참조하십시오.

기존 인스턴스에 대한 인스턴스 메타데이터 옵션 구성

기존 인스턴스에서 IMDSv2를 사용해야 하도록 설정할 수 있습니다. 또한 PUT 응답 홉 제한을 변경하고 기존 인스턴스의 인스턴스 메타데이터에 대한 액세스를 비활성화할 수 있으며, 사용자가 기존 인스턴스의 인스턴스 메타데이터 옵션을 수정하지 못하도록 하는 IAM 정책을 생성할 수도 있습니다.

기존 인스턴스에서 IMDSv2를 사용해야 하도록 설정하려면

기존 인스턴스의 경우 인스턴스 메타데이터 요청 시 IMDSv2를 사용해야 하도록 선택할 수 있습니다. modify-instance-metadata-options CLI 명령을 사용하고 `http-tokens` 파라미터를 `required`로 설정합니다.

Note

또한 `http-tokens`의 값을 지정할 때 `http-endpoint`를 `enabled`로 설정해야 합니다.

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567898abcdef0 \
    --http-tokens required \
    --http-endpoint enabled
```

기존 인스턴스에 대한 PUT 응답 홉 제한을 변경하려면

기존 인스턴스의 경우 `PUT` 응답 홉 제한에 대한 설정을 변경할 수 있습니다. modify-instance-metadata-options CLI 명령을 사용하고 `http-put-response-hop-limit` 파라미터를 필요한 홉 수로 설정합니다. 다음 예제에서는 옵 제한이 `3`으로 설정됩니다. 또한 `http-put-response-hop-limit`의 값을 지정할 때 `http-endpoint`를 `enabled`로 설정해야 합니다.

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567898abcdef0 \
    --http-put-response-hop-limit 3 \
    --http-endpoint enabled
```

기존 인스턴스의 인스턴스 메타데이터에 대한 액세스를 끄려면

기존 인스턴스의 경우, 사용 중인 인스턴스 메타데이터 서비스의 버전과 상관없이, 인스턴스 메타데이터 서비스의 HTTP 엔드포인트를 비활성화하여 인스턴스 메타데이터에 대한 액세스를 끌 수 있습니다. HTTP 엔드포인트를 활성화하여 언제든지 이 변경을 되돌릴 수 있습니다. modify-instance-metadata-options CLI 명령을 사용하고 `http-endpoint` 파라미터를 `disabled`로 설정합니다.

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567898abcdef0 \
    --http-endpoint disabled
```

modify-instance-metadata-options의 사용을 제어하려면

기존 인스턴스에서 인스턴스 메타데이터 옵션을 수정할 수 있는 IAM 사용자를 제어하기 위해, 지정된 역할을 가진 사용자 이외의 모든 사용자가 ModifyInstanceMetadataOptions API를 사용하지 못하도록 하는 정책을 지정할 수 있습니다. IAM 정책 예제는 인스턴스 메타데이터 작업 (p. 840) 단원을 참조하십시오.

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.

인스턴스 메타데이터는 몇 가지 범주로 분류될 수 있습니다. 각 인스턴스 메타데이터 범주에 대한 설명은 인스턴스 메타데이터 카테고리 (p. 588) 단원을 참조하십시오.

실행 중인 인스턴스 내에 있는 인스턴스 메타데이터의 모든 범주를 보려면 다음 URI를 사용하십시오.

```
http://169.254.169.254/latest/meta-data/
```

IP 주소 `169.254.169.254`는 링크-로컬 주소이며 인스턴스에서만 유효합니다. 자세한 내용은 Wikipedia에서 Link-local address를 참조하십시오.

인스턴스 메타데이터 및 사용자 데이터를 가져오기 위해 사용되는 HTTP 요청 비용은 청구되지 않습니다.

명령 형식은 IMDSv1를 사용하는지 또는 IMDSv2를 사용하는지에 따라 다릅니다. 기본적으로 두 가지 인스턴스 메타데이터 서비스를 모두 사용할 수 있습니다. IMDSv2를 사용해야 하도록 설정하려면 인스턴스 메타데이터 서비스 구성 (p. 571) 단원을 참조하십시오.

다음 예제와 같이 cURL과 같은 도구를 사용할 수 있습니다.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-
ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-
data/
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/
```

또한 인스턴스 메타데이터 쿼리 도구를 다운로드하면 전체 URI 또는 범주 이름을 입력하지 않아도 인스턴스 메타데이터 서비스 버전 1을 사용하여 인스턴스 메타데이터를 쿼리할 수 있습니다.

응답 및 오류 메시지

모든 인스턴스 메타데이터는 텍스트(HTTP 콘텐츠 유형 `text/plain`)로 반환됩니다.

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 서비스 버전 2를 사용하여 수행한 요청의 경우 다음 HTTP 오류 코드가 반환될 수 있습니다.

- `400 - Missing or Invalid Parameters` – `PUT` 요청이 유효하지 않습니다.
- `401 - Unauthorized` – `GET` 요청이 유효하지 않은 토큰을 사용합니다. 권장되는 작업은 새 토큰을 생성하는 것입니다.
- `403 - Forbidden` – 요청이 허용되지 않거나 인스턴스 메타데이터 서비스가 꺼져 있습니다.

인스턴스 메타데이터 가져오기 예제

예제

- 인스턴스 메타데이터의 사용 가능한 버전 가져오기 (p. 577)
- 최상위 수준 메타데이터 항목 가져오기 (p. 578)
- 사용 가능한 퍼블릭 키 목록 가져오기 (p. 580)
- 퍼블릭 키 0을 사용할 수 있는 형식 표시 (p. 581)
- 퍼블릭 키 0(OpenSSH 키 형식) 가져오기 (p. 581)
- 인스턴스에 대한 서브넷 ID 가져오기 (p. 582)

인스턴스 메타데이터의 사용 가능한 버전 가져오기

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.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-
ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/
1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
2014-11-05
2015-10-20
2016-04-19
2016-06-30
2016-09-02
latest
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/
1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
2014-11-05
2015-10-20
2016-04-19
2016-06-30
2016-09-02
latest
```

최상위 수준 메타데이터 항목 가져오기

이 예제는 최고 수준 메타데이터 항목을 가져옵니다. 자세한 내용은 인스턴스 메타데이터 카테고리 (p. 588) 단원을 참조하십시오.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-
ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-
data/
ami-id
ami-launch-index
```

```
ami-manifest-path
block-device-mapping/
events/
hostname
iam/
instance-action
instance-id
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
events/
hostname
iam/
instance-action
instance-id
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/
```

다음 예제에서는 이전 예제에서 얻은 몇 가지 최상위 메타데이터 항목의 값을 가져옵니다. IMDSv2 요청에서는 토큰이 만료되지 않았다고 가정하고 이전 예제 명령에서 생성한 저장된 토큰을 사용합니다.

IMDSv2

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/
latest/meta-data/ami-id
ami-0abcdef1234567890
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/ami-id
```

```
ami-0abcdef1234567890
```

IMDSv2

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/
latest/meta-data/reservation-id
r-0efghijk987654321
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/reservation-id
r-0efghijk987654321
```

IMDSv2

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/
latest/meta-data/local-hostname
ip-10-251-50-12.ec2.internal
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/local-hostname
ip-10-251-50-12.ec2.internal
```

IMDSv2

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/
latest/meta-data/public-hostname
ec2-203-0-113-25.compute-1.amazonaws.com
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-hostname
ec2-203-0-113-25.compute-1.amazonaws.com
```

사용 가능한 퍼블릭 키 목록 가져오기

이 예제를 통해 이용 가능한 퍼블릭 키 목록을 획득할 수 있습니다.

IMDSv2

```
[ec2-user ~]$ `curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-
data/public-keys/
0=my-public-key
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/
0=my-public-key
```

퍼블릭 키 0을 사용할 수 있는 형식 표시

이 예제는 퍼블릭 키 0을 이용할 수 있는 형식을 보여줍니다.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-
ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-
data/public-keys/0/
openssh-key
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/0/
openssh-key
```

퍼블릭 키 0(OpenSSH 키 형식) 가져오기

이 예제에서는 퍼블릭 키 0(OpenSSH 키 형식)을 획득합니다.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-
ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-
data/public-keys/0/openssh-key
ssh-rsa MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE my-public-key
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
ssh-rsa MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
```

```
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE my-public-key
```

인스턴스에 대한 서브넷 ID 가져오기

이 예제에서는 인스턴스에 대한 서브넷 ID를 가져옵니다.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-
ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-
data/network/interfaces/macs/02:29:96:8f:6a:2d/subnet-id
subnet-be9b61d7
```

IMDSv1

```
[ec2-user ~]$ 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, 모든 트랜잭션 중에 또는 많은 스레드나 프로세스에서 동시에 자격 증명을 쿼리하지 마십시오. 이렇게 하면 조절이 발생할 수 있습니다. 자격 증명 만료일이 다가오기 전까지는 자격 증명을 캐시에 저장하는 것이 좋습니다.

인스턴스 메타데이터 서비스를 액세스하는 동안 조절이 발생하면 지수 백오프 전략으로 쿼리를 다시 시도하십시오.

인스턴스 메타데이터 서비스 액세스 제한

로컬 방화벽 규칙을 사용하여 인스턴스 메타데이터 서비스에 대한 일부 또는 모든 프로세스의 액세스를 비활성화할 수 있습니다.

iptables를 사용하여 액세스 제한

다음 예제에서는 Linux iptables 및 해당 `owner` 모듈을 사용하여 Apache 웹 서버(기본 설치 사용자 ID `apache` 기준)가 169.254.169.254에 액세스할 수 없도록 설정합니다. 이 예제에서는 거부 규칙을 사용하여 해당 사용자로 실행하는 모든 프로세스의 모든 인스턴스 메타데이터 요청(IMDSv1이든 IMDSv2든 상관없이)을 거부합니다.

```
$ sudo iptables --append OUTPUT --proto tcp --destination 169.254.169.254 --match owner --
uid-owner apache --jump REJECT
```

또는 허용 규칙을 사용하여 특정 사용자 또는 그룹에 대한 액세스만 허용할 수도 있습니다. 허용 규칙을 사용하면 어떤 소프트웨어가 인스턴스 메타데이터에 액세스해야 하는지를 결정해야 하므로, 허용 규칙은 보안 관점에서 더 관리하기 쉬울 수 있습니다. 허용 규칙을 사용하면 인스턴스의 소프트웨나 구성을 나중에 변경하는 경우 소프트웨어가 메타데이터 서비스(액세스할 의도가 없는 서비스)에 액세스하도록 우발적으로 허용할

가능성이 낮습니다. 방화벽 규칙을 변경할 필요 없이 허용 그룹에서 사용자를 추가하고 제거할 수 있도록, 그룹 사용을 허용 규칙과 결합할 수도 있습니다.

다음 예제에서는 사용자 계정 `trustworthy-user`에서 실행 중인 프로세스를 제외하고 모든 프로세스가 인스턴스 메타데이터 서비스에 액세스할 수 없도록 설정합니다.

```
$ sudo iptables --append OUTPUT --proto tcp --destination 169.254.169.254 --match owner !
 --uid-owner trustworthy-user --jump REJECT
```

Note

- 로컬 방화벽 규칙을 사용하려면 이전 예제 명령을 필요에 맞게 조정해야 합니다.
- 기본적으로 iptables 규칙은 시스템 재부팅 후에 지속되지 않습니다. 여기서 설명하지 않는 OS 기능을 사용하여 이 규칙이 지속되도록 할 수 있습니다.
- iptables `owner` 모듈은 그룹이 지정된 로컬 사용자의 기본 그룹인 경우에만 그룹 멤버십과 일치합니다. 기타 그룹은 일치되지 않습니다.

PF 또는 IPFW를 사용하여 액세스 제한

FreeBSD 또는 OpenBSD를 사용하는 경우 PF 또는 IPFW를 사용할 수도 있습니다. 다음 예제에서는 인스턴스 메타데이터 서비스에 대한 액세스를 루트 사용자만으로 제한합니다.

PF

```
$ block out inet proto tcp from any to 169.254.169.254
```

```
$ pass out inet proto tcp from any to 169.254.169.254 user root
```

IPFW

```
$ allow tcp from any to 169.254.169.254 uid root
```

```
$ deny tcp from any to 169.254.169.254
```

Note

PF 및 IPFW 명령의 순서가 중요합니다. PF는 기본적으로 마지막 일치하는 규칙으로 설정되고 IPFW는 기본적으로 첫 번째 일치하는 규칙으로 설정됩니다.

인스턴스 사용자 데이터로 작업

인스턴스 사용자 데이터로 작업 시 다음 사항에 유의하십시오.

- 사용자 데이터는 base64로 인코딩해야 합니다. Amazon EC2 콘솔은 base64 인코딩을 수행하거나 base64로 인코딩된 입력을 수락할 수 있습니다.
- 사용자 데이터는 base64로 인코딩되기 전에 원시 16KB 형식으로 제한됩니다. base64 인코딩 이후 n 길이의 문자열 크기는 ceil(n/3)*4입니다.
- 사용자 데이터는 가져올 때 base64로 디코딩해야 합니다. 인스턴스 메타데이터 또는 콘솔을 사용하여 데이터를 가져오는 경우 데이터는 자동으로 디코딩됩니다.
- 사용자 데이터는 불투명 데이터로 취급됨: 제공한 것만을 살펴볼 수 있습니다. 해석 가능성은 인스턴스에 따라 다릅니다.

- 인스턴스를 중지하고 사용자 데이터를 수정한 다음 인스턴스를 시작할 경우 인스턴스를 시작할 때 업데이트된 사용자 데이터가 실행되지 않습니다.

시작 시 인스턴스 사용자 데이터 지정

인스턴스를 시작할 때 사용자 데이터를 지정할 수 있습니다. 자세한 내용은 인스턴스 시작 마법사를 사용하여 인스턴스 시작 (p. 431) 및 시작 시 Linux 인스턴스에서 명령 실행 (p. 565) 단원을 참조하십시오.

인스턴스 사용자 데이터 수정

루트 볼륨이 EBS 볼륨인 경우 인스턴스의 사용자 데이터를 중지된 상태에서 수정할 수 있습니다. 자세한 내용은 인스턴스 사용자 데이터 보기 및 업데이트 (p. 567) 단원을 참조하십시오.

인스턴스 사용자 데이터 가져오기

실행 중인 인스턴스 내에서 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.

이 예제는 쉼표로 구분된 텍스트로 제공된 사용자 데이터를 반환합니다.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-
ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/user-
data
1234,john,reboot,true | 4512,richard, | 173,,,
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

이 예제는 스크립트로 제공된 사용자 데이터를 반환합니다.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-
ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/user-
data
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/user-data
#!/bin/bash
yum update -y
service httpd start
```

```
chkconfig httpd on
```

고유의 컴퓨터에서 인스턴스용 사용자 데이터를 가져오려면 사용자 데이터 및 AWS CLI (p. 569) 단원을 참조하십시오.

동적 데이터 가져오기

실행 중인 인스턴스 내에서 동적 데이터를 가져오려면 다음 URI를 사용합니다.

```
http://169.254.169.254/latest/dynamic/
```

이 예제는 높은 수준의 인스턴스 자격 증명 범주를 가져오는 방법을 보여 줍니다.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-
ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/dynamic/
instance-identity/
rsa2048
pkcs7
document
signature
dsa2048
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/dynamic/instance-identity/
rsa2048
pkcs7
document
signature
dsa2048
```

동적 데이터 및 가져오기 방법의 예제에 대한 자세한 내용은 인스턴스 자격 증명 문서 (p. 593) 단원을 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 역할을 하도록 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 값을 이용하여 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 구성을 정의하고 `replicate-every=5min`는 두 번째 복제 구성을 하는 식으로 동작합니다. Alice는 서로 다른 인스턴스의 데이터 구분자로 파이프 기호(`|`)를 사용하는 ASCII 문자열로 이 데이터를 제공하려 합니다.

Alice는 run-instances 명령을 사용하여 4개의 인스턴스를 시작하고 사용자 데이터를 지정합니다.

```
aws ec2 run-instances --image-id ami-0abcdef1234567890 --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-0abcdef1234567890
- 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
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

Metadata	값
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-launch-index` 값을 사용하여 사용자 데이터의 어느 부분이 특정 인스턴스에 적용 가능한지를 결정할 수 있습니다.

1. Alice는 인스턴스 중 하나에 접속한 다음 해당 인스턴스의 `ami-launch-index`를 검색하여 복제본인지 확인합니다.

   IMDSv2

   ```
   [ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/meta-data/api/token"
    -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
   && curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-
   data/ami-launch-index
   2
   ```

   다음 단계의 경우 토큰이 만료되지 않았다고 가정하고 IMDSv2 요청에서 이전 IMDSv2 명령의 저장된 토큰을 사용합니다.

   IMDSv1

   ```
   [ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/ami-launch-index
   2
   ```

2. `ami-launch-index`를 변수로 저장합니다.

   IMDSv2

   ```
   [ec2-user ~]$ ami_launch_index=`curl -H "X-aws-ec2-metadata-token: $TOKEN" -v
    http://169.254.169.254/latest/meta-data/ami-launch-index`
   ```

   IMDSv1

   ```
   [ec2-user ~]$ ami_launch_index=`curl http://169.254.169.254/latest/meta-data/ami-
   launch-index`
   ```

3. 사용자 데이터를 변수로 저장합니다.

IMDSv2

```
[ec2-user ~]$ user_data=`curl -H "X-aws-ec2-metadata-token: $TOKEN" -v
 http://169.254.169.254/latest/user-data`
```

IMDSv1

```
[ec2-user ~]$ 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.

   IMDSv2

```
[ec2-user ~]$ echo $user_data | cut -d"|" -f"$ami_launch_index"replicate-every=5min
```

   IMDSv1

```
[ec2-user ~]$ echo $user_data | cut -d"|" -f"$ami_launch_index"
replicate-every=5min
```

인스턴스 메타데이터 카테고리

다음 표는 인스턴스 메타데이터의 카테고리를 목록으로 표시합니다.

> **Important**
>
> 다음 표의 범주 이름 중 일부는 해당 인스턴스에만 있는 데이터 자리 표시자입니다. 예를 들어, *mac*은 네트워크 인터페이스의 MAC 주소를 나타냅니다. 이 자리 표시자를 실제 값으로 바꿔야 합니다.

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 `unknown`입니다.	1.0
`ancestor-ami-ids`	이 AMI를 생성하기 위해 다시 번들링된 모든 인스턴스의 AMI ID. 이 값은 AMI 매니페스트 파일에 `ancestor-amis` 키가 있는 경우에만 존재합니다.	2007-10-10
`block-device-mapping/ami`	루트/부트 파일 시스템을 포함하는 가상 디바이스.	2007-12-15
`block-device-mapping/ebsN`	Amazon EBS 볼륨과 연결된 가상 디바이스. Amazon EBS 볼륨은 시작 시 존재하는 경우 또는 인스턴스를 마지	2007-12-15

테스트	설명	버전 소개
	막으로 시작한 시점에만 메타데이터에서 사용할 수 있습니다. N은 Amazon EBS 볼륨의 색인을 나타냅니다(`ebs1` 또는 `ebs2` 등).	
`block-device-mapping/ephemeral` N	모든 비 NVMe 인스턴스 스토어 볼륨의 가상 디바이스입니다. N은 각 볼륨의 인덱스를 나타냅니다. 블록 디바이스 매핑에 있는 인스턴스 스토어 볼륨 수는 인스턴스에 대한 실제 인스턴스 스토어 볼륨 수와 일치하지 않을 수도 있습니다. 인스턴스 유형은 인스턴스에 사용 가능한 인스턴스 스토어 볼륨 수를 결정합니다. 블록 디바이스 매핑에 있는 인스턴스 스토어 볼륨 수가 인스턴스에 사용 가능한 수를 초과한 경우 추가 인스턴스 스토어 볼륨이 무시됩니다.	2007-12-15
`block-device-mapping/root`	루트 디바이스 또는 루트(/ 또는 C:) 파일 시스템이 특정 인스턴스와 연결된 경우 가상 디바이스의 파티션과 연결된 가상 디바이스 또는 파티션입니다.	2007-12-15
`block-device-mapping/swap`	`swap`와 연결된 가상 디바이스. 항상 존재하는 것은 아님.	2007-12-15
`elastic-gpus/associations/`*`elastic-gpu-id`*	인스턴스에 연결된 탄력적 GPU가 있는 경우 탄력적 GPU에 대한 정보(해당 ID 및 연결 정보 포함)를 비롯한 JSON 문자열을 포함합니다.	2016-11-30
`elastic-inference/associations/`*`eia-id`*	인스턴스에 연결된 탄력적 추론 액셀러레이터가 있는 경우, 탄력적 추론 액셀러레이터에 대한 정보(해당 ID 및 유형 포함)를 비롯한 JSON 문자열을 포함합니다.	2018-11-29
`events/maintenance/history`	인스턴스에 대해 완료되거나 취소한 유지 관리 이벤트가 있다면, 이벤트에 관한 정보가 있는 JSON 문자열이 포함됩니다. 자세한 내용은 완료되거나 취소된 이벤트에 대한 이벤트 기록 보기 (p. 611)를 참조하십시오.	2018-08-17
`events/maintenance/scheduled`	인스턴스에 대해 활성화된 유지 관리 이벤트가 있다면, 이벤트에 관한 정보가 있는 JSON 문자열이 포함됩니다. 자세한 내용은 예약된 이벤트 확인 (p. 608) 단원을 참조하십시오.	2018-08-17
`hostname`	인스턴스의 프라이빗 IPv4 DNS 호스트 이름. 다중 네트워크 인터페이스가 존재하는 경우 eth0 디바이스를 의미함(디바이스 번호가 0인 디바이스).	1.0

테스트	설명	버전 소개
`iam/info`	인스턴스 시작 시 IAM 역할이 연결되어 있을 경우, 인스턴스의 LastUpdated date, InstanceProfileArn 및 InstanceProfileId 등 마지막으로 인스턴스 프로파일이 업데이트된 시간 관련 정보를 포함합니다. 그렇지 않을 경우 제공되지 않습니다.	2012-01-12
`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. 851) 참조). 그렇지 않을 경우 제공되지 않습니다.	2012-01-12
`identity-credentials/ec2/info`	[내부 전용으로 예약됨] AWS에서 나머지 Amazon EC2 인프라에 대해 인스턴스를 식별하는 데 사용하는 자격 증명 관련 정보.	2018-05-23
`identity-credentials/ec2/security-credentials/ec2-instance`	[내부 전용으로 예약됨] AWS에서 나머지 Amazon EC2 인프라에 대해 인스턴스를 식별하는 데 사용하는 자격 증명.	2018-05-23
`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. 178) 단원을 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
`metrics/vhostmd`	더 이상 사용할 수 없습니다.	2011-05-01

테스트	설명	버전 소개
`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/interface-id`	네트워크 인터페이스의 ID입니다.	2011-01-01
`network/interfaces/macs/mac/ipv4-associations/public-ip`	각 퍼블릭 IP 주소와 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
`network/interfaces/macs/mac/local-hostname`	인터페이스의 로컬 호스트이름.	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`	인터페이스의 퍼블릭 DNS(IPv4). 이 범주는 `enableDnsHostnames` 속성이 `true`로 설정된 경우에만 반환됩니다. 자세한 내용은 VPC에서 DNS 사용하기 단원을 참조하십시오.	2011-01-01
`network/interfaces/macs/mac/public-ipv4s`	인터페이스와 연결된 퍼블릭 IP 주소 또는 탄력적 IP 주소입니다. 인스턴스에는 다중 IPv4 주소가 있을 수 있습니다.	2011-01-01
`network/interfaces/macs/mac/security-groups`	네트워크 인터페이스에 속한 보안 그룹.	2011-01-01
`network/interfaces/macs/mac/security-group-ids`	네트워크 인터페이스에 속한 보안 그룹의 ID.	2011-01-01
`network/interfaces/macs/mac/subnet-id`	인터페이스가 위치하는 서브넷 ID.	2011-01-01

테스트	설명	버전 소개
`network/interfaces/macs/mac/subnet-ipv4-cidr-block`	인터페이스가 위치하는 서브넷의 IPv4 CIDR 블록.	2011-01-01
`network/interfaces/macs/mac/subnet-ipv6-cidr-blocks`	인터페이스가 위치하는 서브넷의 IPv6 CIDR 블록.	2016-06-30
`network/interfaces/macs/mac/vpc-id`	인터페이스가 위치하는 VPC의 ID.	2011-01-01
`network/interfaces/macs/mac/vpc-ipv4-cidr-block`	VPC의 기본 IPv4 CIDR 블록.	2011-01-01
`network/interfaces/macs/mac/vpc-ipv4-cidr-blocks`	VPC에 대한 IPv4 CIDR 블록.	2016-06-30
`network/interfaces/macs/mac/vpc-ipv6-cidr-blocks`	인터페이스가 위치하는 VPC의 IPv6 CIDR 블록.	2016-06-30
`placement/availability-zone`	인스턴스가 시작된 가용 영역.	2008-02-01
`product-codes`	인스턴스와 연결된 AWS Marketplace 제품 코드입니다(있는 경우).	2007-03-01
`public-hostname`	인스턴스의 퍼블릭 DNS. 이 범주는 `enableDnsHostnames` 속성이 `true`로 설정된 경우에만 반환됩니다. 자세한 내용은 Amazon VPC 사용 설명서의 VPC에서 DNS 사용 을 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. 시작 이후 인스턴스의 보안 그룹을 변경할 수 있습니다. 해당 변경은 여기 및 network/interfaces/macs/*mac*/security-groups에 반영됩니다.	1.0
`services/domain`	리전의 AWS 리소스에 대한 도메인입니다.	2014-02-25
`services/partition`	리소스가 있는 파티션. 표준 AWS 리전에서 파티션은 `aws`입니다. 리소스가 다른 파티션에 있는 경우 파티션은 `aws-`*`partitionname`*입니다. 예를 들어 중국(베이징) 리전에 있는 리소스의 파티션은 `aws-cn`입니다.	2015-10-20

테스트	설명	버전 소개
`spot/instance-action`	항목이 발생할 때 작업(최대 절전 모드, 중지 또는 종료)과 작업이 이루어지는 대략의 시간(UTC)입니다. 이 항목은 스팟 인스턴스가 최대 절전 모드, 중지, 종료로 표시된 경우에만 존재합니다. 자세한 내용은 instance-action (p. 376) 단원을 참조하십시오.	2016-11-15
`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. 자세한 내용은 termination-time (p. 377) 단원을 참조하십시오.	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. 593) 단원을 참조하십시오.	2009-04-04
`instance-identity/pkcs7`	문서의 신뢰성 및 서명 내용을 검증하는 데 사용됩니다. 인스턴스 자격 증명 문서 (p. 593) 단원을 참조하십시오.	2009-04-04
`instance-identity/signature`	출처 및 신뢰성을 검증하기 위해 다른 사용자가 사용할 수 있는 데이터. 인스턴스 자격 증명 문서 (p. 593) 단원을 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 확인하는 데 사용할 수 있는 서명 및 PKCS7 서명이 함께 제공됩니다.

인스턴스 자격 증명 문서는 인스턴스를 시작할 때 생성되고 인스턴스 메타데이터 (p. 571)를 통해 인스턴스에 공개됩니다. 이 문서는 인스턴스의 속성(예: 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 다음 명령을 사용합니다.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-
ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/dynamic/
instance-identity/document
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/dynamic/instance-identity/document
```

다음은 예제 출력입니다.

```
{
    "devpayProductCodes" : null,
    "marketplaceProductCodes" : [ "1abc2defghijklm3nopqrs4tu" ],
    "availabilityZone" : "us-west-2b",
    "privateIp" : "10.158.112.84",
    "version" : "2017-09-30",
    "instanceId" : "i-1234567890abcdef0",
    "billingProducts" : null,
    "instanceType" : "t2.micro",
    "accountId" : "123456789012",
    "imageId" : "ami-5fb8c835",
    "pendingTime" : "2016-11-19T16:32:11Z",
    "architecture" : "x86_64",
    "kernelId" : null,
    "ramdiskId" : null,
    "region" : "us-west-2"
}
```

인스턴스 자격 증명 서명을 검색하려면 실행 중인 인스턴스에서 다음 명령을 사용합니다.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-
ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/dynamic/
instance-identity/signature
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/dynamic/instance-identity/signature
```

다음은 예제 출력입니다.


```
dExamplesjNQhhJan7pORLpLSr7lJEF4V2DhKGlyoYVBoUYrY9njyBCmhEayaGrhtS/AWY+LPx
lVSQURF5n0gwPNCuO6ICT0fNrm5IH7w9ydyaexamplejJw8XvWPxbuRkcN0TAA1p4RtCAqm4ms
x2oALjWSCBExample=
```


PKCS7 서명을 검색하려면 실행 중인 인스턴스에서 다음 명령을 사용합니다.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-
ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/dynamic/
instance-identity/pkcs7
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254//latest/dynamic/instance-identity/pkcs7
```

다음은 예제 출력입니다.


```
MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE
```


PKCS7 서명 확인

PKCS7 서명을 사용해 해당 AWS 퍼블릭 인증서에 대해 유효성을 검증함으로써 인스턴스를 확인할 수 있습니다.

AWS 계정에서 제공하는 리전에 대한 AWS 퍼블릭 인증서는 다음과 같습니다.


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


홍콩 리전을 위한 AWS 퍼블릭 인증서는 다음과 같습니다.


```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq4CCQCO7MJe5Y3VLjAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
```

```
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xOTAyMDMwMjIxMjFaFw00
NTAyMDMwMjIxMjFaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbgwggEsBgcqhkjOOAQBMIIBHwKBgQDvQ9RzVvf4MAwGbqfX
blCvCoVb9957OkLGn/04CowHXJ+vTBR7eyIa6AoXltsQXBOmrJswToFKKxT4gbuw
jK7s9QQX4CmTRWcEgO2RXtZSVjOhsUQMh+yf7Ht4OVL97LWnNfGsX2cwjcRWHYgI
7lvnuBNBzLQHdSEwMNq0Bk76PwIVAMan6XIEEPnwr4e6u/RNnWBGKd9FAoGBAOCG
eSNmxpW4QFu4pIlAykm6EnTZKKHT87gdXkAkfoC5fAfOxxhnE2HezZHp9Ap2tMV5
8bWNvoPHvoKCQqwfm+OUBlAxC/3vqoVkKL2mG1KgUH9+hrtpMTkwO3RREnKe7I5O
x9qDimJpOihrL4I0dYvy9xUOoz+DzFAW8+ylWVYpA4GFAAKBgQDbnBAKSxWr9QHY
6Dt+EFdGz6lAZLedeBKpaP53Z1DTO34J0C55YbJTwBTFGqPtOLxnUVDlGiD6GbmC
80f3jvogPR1mSmGsydbNbZnbUEVWrRhe+y5zJ3g9qs/DWmDW0deEFvkhWVnLJkFJ
9pdOu/ibRPH1lE2nz6pK7GbOQtLyHTAJBgcqhkjOOAQDAzAAMC0CFQCoJlwGtJQC
cLoM4p/jtVFOj26xbgIUUS4pDKyHaG/eaygLTtFpFJqzWHc=
-----END CERTIFICATE-----
```


바레인 리전을 위한 AWS 퍼블릭 인증서는 다음과 같습니다.


```
-----BEGIN CERTIFICATE-----
MIIC7jCCAq4CCQCVWIgSmP8RhTAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xOTAyMDUxMzA2MjFaFw00
NTAyMDUxMzA2MjFaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbgwggEsBgcqhkjOOAQBMIIBHwKBgQDcwojQfgWdV1Qli00B
8n6cLZ38VE7ZmrjZ9OQV//Gst6S1h7euhC23YppKXi1zovefSDwFU54zi3/oJ++q
PHlP1WGL8IZ34BUgRTtG4TVolvp0smjkMvyRu5hIdKtzjV93Ccx15gVgyk+o1IEG
fZ2Kbw/Dd8JfoPS7KaSCmJKxXQIVAIZbIaDFRGa2qcMkW2HWASyNDl7bAoGBANtz
IdhfMq+l2I5iofY2oj3HI21Kj3LtZrWEg3W+/4rVhL3lTm0Nne1rl9yGujrjQwy5
Zp9V4A/w9w2O10Lx4K6hj34Eefy/aQnZwNdNhv/FQP7Az0fju+Yl6L13OOHQrL0z
Q+9cF7zEosekEnBQx3v6psNknKgD3Shgx+GO/LpCA4GFAAKBgQCVS7m77nuNAlZ8
wvUqcooxXMPkxJFl54NxAsAul9KP9KN4svm0O3Zrb7t2FOtXRM8zU3TqMpryq1o5
mpMPsZDg6RXo9BF7Hn0DoZ6PJTamkFA6md+NyTJWJKvXC7iJ8fGDBJqTciUHuCKr
12AztQ8bFWsrTgTzPE3p6U5ckcgV1TAJBgcqhkjOOAQDAy8AMCwCFB2NZGWm5EDl
86ayV3c1PEDukgQIAhQow38rQkN/VwHVeSW9DqEshXHjuQ==
-----END CERTIFICATE-----
```


AWS GovCloud (US-West) 리전을 위한 AWS 퍼블릭 인증서는 다음과 같습니다.


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbgwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


다른 리전에 대한 AWS 퍼블릭 인증서를 얻으려면 AWS Support에 문의하십시오.

PKCS7 서명을 확인하려면

1. 인스턴스에서 PKCS7 서명을 위한 임시 파일을 만듭니다.

```
[ec2-user ~]$ PKCS7=$(mktemp)
```

2. 임시 PKCS7 파일에 `-----BEGIN PKCS7-----` 헤더를 추가합니다.

```
[ec2-user ~]$ echo "-----BEGIN PKCS7-----" > $PKCS7
```

3. 인스턴스 메타데이터에서 얻은 PKCS7 서명의 내용을 추가하고 새 줄을 추가합니다.

   IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-
aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -s http://169.254.169.254/latest/
dynamic/instance-identity/pkcs7 >> $PKCS7
[ec2-user ~]$ echo "" >> $PKCS7
```

   IMDSv1

```
[ec2-user ~]$ curl -s http://169.254.169.254/latest/dynamic/instance-identity/pkcs7
 >> $PKCS7
[ec2-user ~]$ echo "" >> $PKCS7
```

4. `-----END PKCS7-----` 바닥글을 추가합니다.

```
[ec2-user ~]$ echo "-----END PKCS7-----" >> $PKCS7
```

5. 인스턴스 자격 증명 문서를 위한 임시 파일을 만듭니다.

```
[ec2-user ~]$ DOCUMENT=$(mktemp)
```

6. 인스턴스 메타데이터에서 얻은 문서의 내용을 임시 문서 파일에 추가합니다.

   IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-
aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -s http://169.254.169.254/latest/
dynamic/instance-identity/document > $DOCUMENT
```

   IMDSv1

```
[ec2-user ~]$ curl -s http://169.254.169.254/latest/dynamic/instance-identity/
document > $DOCUMENT
```

7. 텍스트 편집기를 열어 `AWSpubkey`라는 파일을 만듭니다. 파일에 위 AWS 퍼블릭 인증서의 내용을 복사해 붙여 넣은 다음 저장합니다.
8. OpenSSL 도구를 사용해 다음과 같이 서명을 확인합니다.

```
[ec2-user ~]$ openssl smime -verify -in $PKCS7 -inform PEM -content $DOCUMENT -certfile
 AWSpubkey -noverify > /dev/null
Verification successful
```

Amazon Elastic Inference

Amazon Elastic Inference(EI)는 딥러닝(DL) 추론 워크로드를 가속화하기 위해 Amazon EC2 인스턴스에 연결할 수 있는 리소스입니다. Amazon EI 액셀러레이터는 여러 크기로 제공되며 Amazon EC2 인스턴스에서 실행 중인 애플리케이션에 지능형 기능을 구축할 수 있는 비용 효율적인 방법입니다.

Amazon EI는 낮은 비용의 DL 추론 액셀러레이터와 인스턴스의 CPU 간에 TensorFlow, Apache MXNet 및 Open Neural Network Exchange(ONNX) 형식으로 정의된 모델 작업을 MXNet를 통해 배포합니다.

Amazon Elastic Inference에 대한 자세한 내용은 Amazon EI 개발자 안내서를 참조하십시오.

EC2 Linux 인스턴스 식별

애플리케이션이 EC2 인스턴스에서 실행 중인지 여부를 확인해야 할 수 있습니다.

Windows 인스턴스 식별에 대한 자세한 내용은 Windows 인스턴스용 Amazon EC2 사용 설명서에서 EC2 Windows 인스턴스 식별 단원을 참조하십시오.

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 (p. 593) 단원을 참조하십시오.

시스템 UUID 검사

시스템 UUID를 가져와 UUID의 시작 8진수에 문자 "ec2" 또는 "EC2"가 있는지 찾을 수 있습니다. 시스템이 EC2 인스턴스인지 확인하는 이 방법은 빠르지만 EC2 인스턴스가 아닌 시스템에도 이러한 문자로 시작하는 UUID가 있을 가능성이 약간이라도 있기 때문에 부정확할 수 있습니다. 또한 에서 Amazon Linux를 사용하지 않는 경우 배포의 SMBIOS 구현은 리틀 엔디안 형식으로 UUID를 나타낼 수 있으므로 "EC2" 문자는 EC2 시작 부분에 나타나지 않습니다.

Example : 하이퍼바이저에서 UUID 가져오기

`/sys/hypervisor/uuid`가 있을 경우 다음 명령 중 하나를 사용할 수 있습니다.

```
[ec2-user ~]$ cat /sys/hypervisor/uuid
```

다음 출력 예에서 UUID가 "ec2"로 시작합니다. 이는 시스템이 아마도 EC2 인스턴스라는 것을 나타냅니다.

```
ec2e1916-9099-7caf-fd21-012345abcdef
```

Example : DMI에서 UUID 가져오기(HVM 인스턴스만 해당)

HVM 인스턴스에 한해, 데스크톱 관리 인터페이스(DMI)를 사용할 수 있습니다.

`dmidecode` 도구를 사용하여 UUID를 반환할 수 있습니다. Amazon Linux에서, 인스턴스에 이미 설치되지 않은 경우 다음 명령을 사용하여 `dmidecode` 도구를 설치합니다.

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

그런 다음, 다음 명령을 실행합니다.

```
[ec2-user ~]$ sudo dmidecode --string system-uuid
```

또는 다음 명령을 사용합니다.

```
[ec2-user ~]$ sudo cat /sys/devices/virtual/dmi/id/product_uuid
```

다음 출력 예에서 UUID가 "EC2"로 시작합니다. 이는 시스템이 아마도 EC2 인스턴스라는 것을 나타냅니다.

```
EC2E1916-9099-7CAF-FD21-01234ABCDEF
```

다음 예제 출력에서 UUID는 리틀 엔디안 형식으로 표시됩니다.

```
45E12AEC-DCD1-B213-94ED-01234ABCDEF
```

Nitro 인스턴스에서는 다음 명령을 사용할 수 있습니다.

```
[ec2-user ~]$ cat /sys/devices/virtual/dmi/id/board_asset_tag
```

EC2 인스턴스 별로 고유한 인스턴스 ID가 반환됩니다.

```
i-0af01c0123456789a
```

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, 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 에이전트/CloudWatch Logs
CPU 사용률	CPUUtilization (p. 617)	
네트워크 사용률	NetworkIn (p. 617) NetworkOut (p. 617)	
디스크 성능	DiskReadOps (p. 617) DiskWriteOps (p. 617)	
디스크 읽기/쓰기	DiskReadBytes (p. 617) DiskWriteBytes (p. 617)	
메모리 사용률, 디스크 스왑 사용률, 디스크 공간 사용률, 페이지 파일 사용률, 로그 수집		[Linux 및 Windows Server 인스턴스] CloudWatch 에이전트를 사용하여 Amazon EC2 인스턴스 및 온프레미스 서버로부터 지표 및 로그 수집 [Windows Server 인스턴스의 이전 CloudWatch Logs 에이전트에서 마이그레이션] Windows 서버 인스턴스 로그 수집을 CloudWatch 에이전트로 마이그레이션

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.

주제

- 자동 모니터링 도구 (p. 601)
- 수동 모니터링 도구 (p. 602)

자동 모니터링 도구

다음과 같은 자동 모니터링 도구를 사용하여 Amazon EC2를 관찰하고 문제 발생 시 보고를 받을 수 있습니다.

- 시스템 상태 검사 - 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
  - 네트워크 연결성에 영향을 주는 물리적 호스트의 하드웨어 문제

  자세한 내용은 인스턴스 상태 확인 (p. 603) 단원을 참조하십시오.
- 인스턴스 상태 검사 - 개별 인스턴스에 대한 소프트웨어 및 네트워크 구성을 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. 603) 단원을 참조하십시오.
- Amazon CloudWatch 경보 - 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 Amazon EC2 Auto Scaling 정책으로 전송되는 알림입니다. 경보는 지속적인 상태 변경에 대해서만 작업을 호출합니다. CloudWatch 경보는 특정 상태가 되었다고 해서 작업을 호출하지는 않습니다. 이러한 상태가 변경되어 지정한 기간 동안 유지되어야 합니다. 자세한 내용은 CloudWatch를 사용해 인스턴스 모니터링하기 (p. 616) 단원을 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 Amazon CloudWatch Events이란 무엇입니까? 단원을 참조하십시오.
- Amazon CloudWatch Logs - Amazon EC2 인스턴스, AWS CloudTrail, 또는 기타 소스의 로그 파일을 모니터링, 저장 및 액세스합니다. 자세한 내용은 Amazon CloudWatch Logs User Guide를 참조하십시오.
- Amazon EC2 모니터링 스크립트 - 인스턴스에서 메모리, 디스크 및 스왑 파일 사용량을 모니터링할 수 있는 Perl 스크립트입니다. 자세한 내용은 Amazon EC2 Linux 인스턴스의 메모리 및 디스크 메트릭 모니터링을 참조하십시오.

- System Center Operations Manager용 AWS 관리 팩 - 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 정보는 AWS Management Pack for Microsoft System Center 단원을 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 측정치 세부 정보(탐색 창에서 인스턴스를 선택하고, 인스턴스를 선택한 다음 모니터링 탭 선택)
  - 볼륨 지표 정보(탐색 창에서 볼륨을 선택하고 볼륨을 선택한 다음 모니터링 탭 선택)
- 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 시각적으로 제공합니다.

인스턴스에서 예약된 특정 이벤트의 상태 또한 확인이 가능합니다. 이벤트 상태는 재부팅이나 만료 등 인스턴스에 대해 설정된 예정 활동에 대한 정보를 제공합니다. 또한 각 이벤트의 예약된 시작 시간과 종료 시간도 제공합니다.

목차

- 인스턴스 상태 확인 (p. 603)
- 예약된 인스턴스 이벤트 (p. 607)

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. 이벤트 상태 데이터는 Amazon EC2가 이미 각 인스턴스 상태(`pending`, `running`, `stopping` 등)에 대해 제공하는 정보와 Amazon CloudWatch가 모니터링하는 사용 지표(CPU 사용량, 네트워크 트래픽, 디스크 활동)를 보완합니다.

상태 확인은 1분마다 실행되며 통과 또는 실패 상태를 반환합니다. 모든 검사 결과가 통과인 경우 인스턴스의 전체 상태는 정상으로 표시됩니다. 하나 이상의 검사 결과가 실패인 경우에는 인스턴스의 전체 상태가 손상됨으로 표시됩니다. 상태 확인은 Amazon EC2에 내장된 기능으로 비활성화하거나 삭제할 수 없습니다.

상태 확인이 실패하면 상태 확인에 대한 해당 CloudWatch 지표가 증가합니다. 자세한 내용은 상태 확인 지표 (p. 623) 단원을 참조하십시오. 이러한 지표를 사용하여 상태 확인 결과를 기준으로 트리거되는 CloudWatch 경보를 생성할 수 있습니다. 예를 들어 특정 인스턴스의 상태 확인에서 실패 항목이 있을 때 알리는 경보를 생성할 수 있습니다. 자세한 내용은 상태 확인 경보 생성 및 수정 (p. 606) 단원을 참조하십시오.

Amazon EC2 인스턴스를 모니터링하고 기본 문제로 인해 인스턴스가 손상된 경우 인스턴스를 자동으로 복구하는 Amazon CloudWatch 경보를 생성할 수도 있습니다. 자세한 내용은 인스턴스 복구 (p. 528) 단원을 참조하십시오.

내용

- 상태 확인 유형 (p. 603)
- 상태 확인 결과 확인 (p. 604)
- 상태 확인 보고 (p. 605)
- 상태 확인 경보 생성 및 수정 (p. 606)

상태 확인 유형

상태 확인은 시스템 상태 확인과 인스턴스 상태 확인, 두 가지로 제공됩니다.

시스템 상태 확인

인스턴스가 실행되는 AWS 시스템을 모니터링합니다. 이러한 확인에서는 복구 시 AWS 개입이 필요한 인스턴스와 관련된 근본적인 문제를 찾아냅니다. 시스템 상태 확인이 실패한 경우, AWS에서 문제를 해결할 때까지 기다리거나 문제를 직접 해결할 수 있습니다. Amazon EBS가 지원하는 인스턴스의 경우, 직접 인스턴스

를 중지한 후 시작할 수 있으며 대부분의 경우 이 인스턴스를 새 호스트로 마이그레이션합니다. 인스턴스 스토어가 지원하는 인스턴스의 경우, 인스턴스를 종료하고 교체할 수 있습니다.

다음은 시스템 상태 확인의 실패 원인이 되는 몇 가지 문제의 예입니다.

- 네트워크 연결 끊김
- 시스템 전원 중단
- 물리적 호스트의 소프트웨어 문제
- 네트워크 연결성에 영향을 주는 물리적 호스트의 하드웨어 문제

인스턴스 상태 확인

개별 인스턴스에 대한 소프트웨어 및 네트워크 구성을 모니터링합니다. Amazon EC2에서는 네트워크 인터페이스(NIC)로 주소 확인 프로토콜(ARP)을 전송하여 인스턴스의 상태를 점검합니다. 이러한 확인에서는 복구 시 사용자의 개입이 필요한 문제를 찾아냅니다. 인스턴스 상태 확인이 실패할 경우에는 일반적으로 사용자가 인스턴스를 재부팅하거나 인스턴스 구성을 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 스타트업 구성
- 메모리가 모두 사용됨
- 파일 시스템 손상
- 호환되지 않는 커널

상태 확인 결과 확인

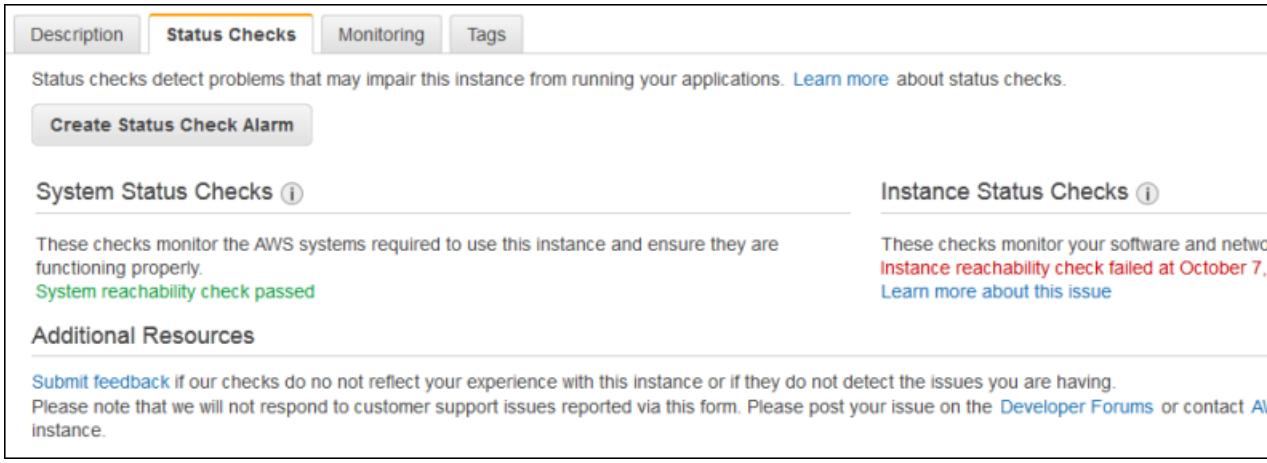
Amazon EC2는 상태 확인 결과를 확인하고 대응할 수 있는 몇 가지 방법이 있습니다.

콘솔을 사용해 상태 확인

AWS Management 콘솔을 사용해 상태 확인 결과를 확인할 수 있습니다.

상태 확인을 보려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스 페이지의 상태 검사 열에는 각 인스턴스의 운영 상태가 목록으로 표시됩니다.
4. 특정 인스턴스의 상태를 보려면 인스턴스를 선택하고 상태 검사 탭을 선택합니다.



상태 확인 실패로 표시된 인스턴스가 있고 이 인스턴스가 확인 불가 상태로 20분 넘게 유지될 경우, AWS Support를 선택하여 지원을 요청합니다. 시스템 또는 인스턴스 상태 확인 실패 문제를 직접 해결하려면 상태 확인에 실패한 인스턴스 문제 해결 (p. 1094) 단원을 참조하십시오.

5. 상태 확인에 대한 CloudWatch 지표를 검토하려면 인스턴스를 선택한 다음 모니터링 탭을 선택합니다. 다음 지표에 대한 그래프가 보일 때까지 스크롤합니다.
   - 상태 확인 실패(해당되는 경우)
   - 상태 확인 실패(인스턴스)
   - 상태 확인 실패(시스템)

명령줄을 사용해 상태 보기

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
```

`impaired`로 표시된 인스턴스의 상태를 모두 확인하려면 다음 명령을 사용합니다.

```
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-EC2InstanceStatus(Windows PowerShell용 AWS 도구)
- DescribeInstanceStatus(Amazon EC2 Query API)

상태 확인 실패로 표시된 인스턴스가 있는 경우, 상태 확인에 실패한 인스턴스 문제 해결 (p. 1094) 단원을 참조하십시오.

상태 확인 보고

상태가 impaired로 표시되지 않은 인스턴스임에도 불구하고 문제가 발생하는 경우 또는 impaired 상태의 인스턴스와 관련해 AWS에 문제에 대해 추가 정보를 보내고 싶은 경우 피드백을 제공할 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

인스턴스 상태를 보고하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택하고, Status Checks(상태 확인) 탭을 선택한 후 피드백 제출을 선택합니다.

4. 인스턴스 상태 보고 양식을 작성한 후 제출을 선택합니다.

명령줄을 사용해 상태 피드백 보고

아래와 같이 report-instance-status(AWS CLI) 명령을 사용해 손상된 상태의 인스턴스에 대한 피드백을 전송합니다.

```
aws ec2 report-instance-status --instances i-1234567890abcdef0 --status impaired --reason-codes code
```

또는 다음 명령을 사용합니다.

- Send-EC2InstanceStatus(Windows PowerShell용 AWS 도구)
- ReportInstanceStatus(Amazon EC2 Query API)

상태 확인 경보 생성 및 수정

상태 확인 지표 (p. 623)를 사용하여 인스턴스에 실패한 상태 확인이 있을 때 알리는 CloudWatch 경보를 생성할 수 있습니다.

콘솔을 사용해 상태 확인 경보 생성

다음 절차에 따라 상태 확인에 실패할 때 이메일을 통해 알림을 전송하거나 인스턴스를 중지, 종료 또는 복구하는 경보를 구성합니다.

상태 확인 경보 생성하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택하고, Status Checks(상태 확인) 탭을 선택한 후 상태 확인 경보 생성을 선택합니다.
4. 알림 보내기를 선택합니다. 기존 SNS 주제를 선택하거나, 주제 생성을 선택하여 새로운 주제를 생성합니다. 새로운 주제를 생성할 때는 수신자에 자신의 이메일 주소와 추가 수신자의 주소를 콤마로 구분하여 입력합니다.
5. (선택 사항) Take the action(작업 수행)을 선택한 후 원하는 작업을 선택합니다.
6. 다음 경우 항상에서 알고 싶은 상태 확인를 선택합니다.

   이전 단계에서 인스턴스 복구를 선택한 경우에는 상태 검사 실패(시스템)를 선택합니다.
7. 최소 발생 기간에서 원하는 평가 주기의 개수를 설정하고 연속 기간에서 경보가 실행되고 이메일 전송이 이루어지기 전에 적용할 평가 주기의 시간 단위를 설정합니다.
8. (선택 사항) 경보 이름에서 경보의 기본 이름을 다른 이름으로 변경합니다.
9. 경보 생성을 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 선택하여 가입 여부를 확인해야 합니다. 경고 알림은 확인된 주소로만 전송됩니다.

필요한 경우 인스턴스 상태 경보를 수정할 수 있습니다.

콘솔을 사용해 상태 확인 경보를 편집하려면

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

2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 작업을 선택한 후 CloudWatch 모니터링을 선택하고 경보 추가/편집을 선택합니다.
4. 경보 세부 정보(Alarm Details) 대화 상자에서 경보 이름을 선택합니다.
5. 경보 편집 대화 상자에서 필요한 설정을 변경한 후 저장을 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. 사용된 CloudWatch 지표는 `StatusCheckFailed`입니다.

AWS CLI를 사용해 상태 확인 경보를 생성하려면

1. 기존의 SNS 주제를 선택하거나 새로운 주제를 생성합니다. 자세한 내용은 AWS Command Line Interface 사용 설명서에서 Amazon SNS와 함께 AWS CLI 사용하기 단원을 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
```

기간은 Amazon CloudWatch 지표가 수집되는 시간 프레임(초)입니다. 이 예제에서는 60초와 5분을 곱셈하여 300초를 사용합니다. 평가 기간은 지표 값을 임계값과 비교해야 하는 연속 기간의 수입니다. 이 예제에서는 2를 사용합니다. 경보 작업은 경보가 트리거될 때 수행할 작업입니다. 이 예제에서는 Amazon SNS를 사용해 이메일을 보낼 수 있도록 경보를 구성합니다.

예약된 인스턴스 이벤트

AWS는 재부팅, 중단/시작 또는 만료 등 여러 가지 인스턴스 이벤트를 예약할 수 있습니다. 이러한 이벤트들은 자주 발생하지 않습니다. 예약된 이벤트의 영향을 받는 인스턴스가 존재하는 경우 AWS가 해당 이벤트가 발생하기 전에 AWS 계정에 연동되어 있는 이메일 주소로 이메일을 전송합니다. 이메일은 시작일과 종료일 등 이벤트에 대한 세부 정보를 제공합니다. 이벤트 기간을 제어할 수 있는 작업은 이벤트에 따라 다릅니다.

예약된 이벤트에 대한 세부 정보를 알 수 있도록 계정의 연락처 정보를 업데이트하려면 계정 설정 페이지로 이동합니다.

내용

- 예약된 이벤트 유형 (p. 608)
- 예약된 이벤트 확인 (p. 608)
- 중지 또는 만료 예약된 인스턴스 관련 작업 (p. 611)
- 재부팅 예약된 인스턴스 작업 (p. 612)
- 인스턴스의 유지 관리 예약 작업 (p. 613)
- 예약된 이벤트 다시 예약 (p. 614)

예약된 이벤트 유형

Amazon EC2에서는 예약된 시간에 이벤트가 발생하는 인스턴스에 대해 다음과 같은 유형의 이벤트를 지원합니다.

- Instance stop(인스턴스 중지): 예약된 시간에 인스턴스가 중지됩니다. 인스턴스를 다시 시작하면 새 호스트로 마이그레이션됩니다. 이러한 유형은 Amazon EBS가 지원하는 인스턴스에만 적용됩니다.
- Instance retirement(인스턴스 만료): 예약된 시간에 인스턴스가 Amazon EBS에서 지원되는 경우 중지되거나 인스턴스 스토어에서 지원되는 경우 종료됩니다.
- Instance reboot(인스턴스 재부팅): 예약된 시간에 인스턴스가 재부팅됩니다.
- System reboot(시스템 재부팅): 예약된 시간에 인스턴스의 호스트가 재부팅됩니다.
- System maintenance(시스템 유지 관리): 예약된 시간에 네트워크 또는 전력 유지 관리로 인스턴스가 일시적인 영향을 받을 수 있습니다.

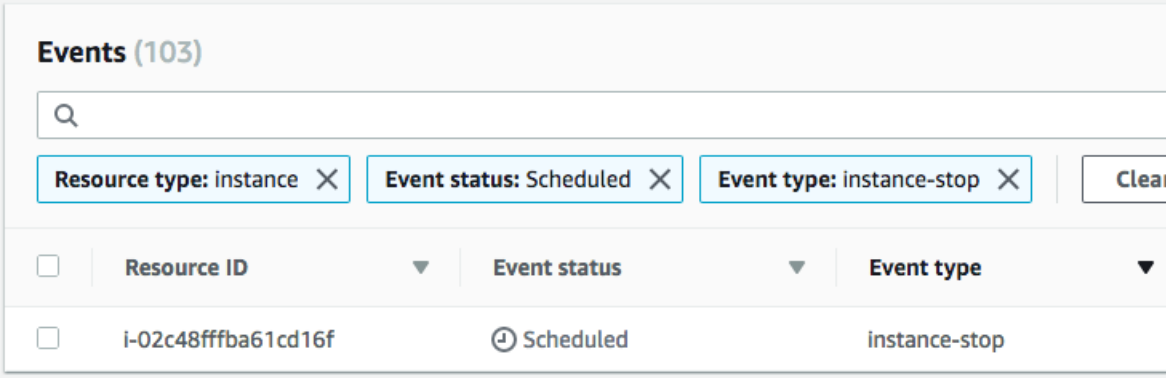
예약된 이벤트 확인

예약된 이벤트에 대한 알림 메시지를 이메일로 받는 것 외에도, 다음 방법 중 하나를 이용해 예약된 이벤트를 확인할 수 있습니다.

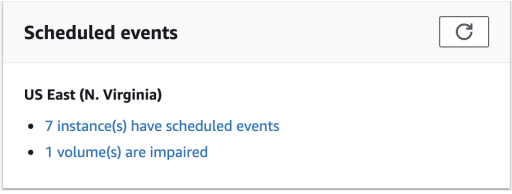
새로운 콘솔

콘솔을 사용해 인스턴스에 예약된 이벤트를 확인하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 다음 화면에서 예약된 이벤트를 볼 수 있습니다.
   - 탐색 창에서 이벤트를 선택합니다. 연동되어 있는 이벤트와 함께 모든 리소스가 표시됩니다. 리소스 ID, 리소스 유형, 가용 영역, 이벤트 상태 또는 이벤트 유형을 기준으로 필터링할 수 있습니다.



   - 또는 탐색 창에서 EC2 대시보드를 선택합니다. 연동되어 있는 이벤트와 함께 모든 리소스가 예약된 이벤트 아래 표시됩니다.

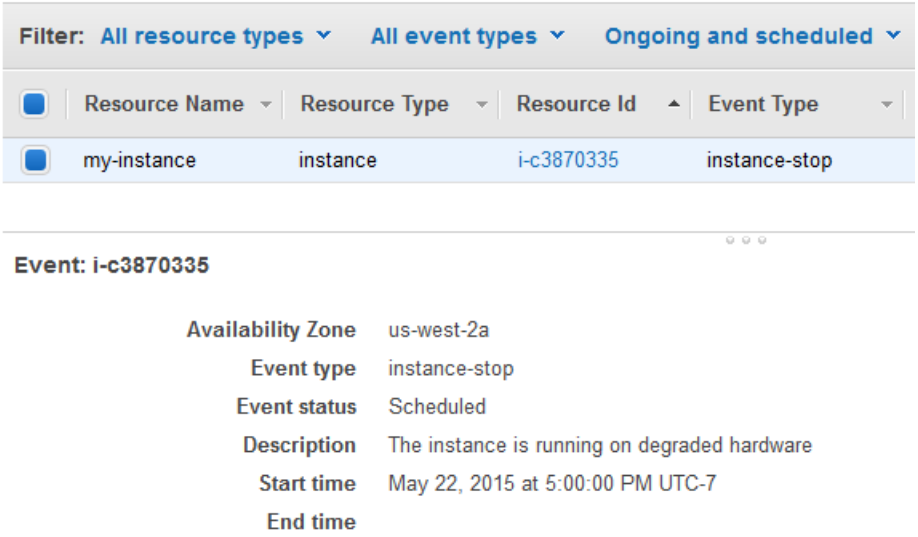


   - 일부 이벤트가 영향을 받는 리소스에 대해서도 표시됩니다. 예를 들어 탐색 창에서 인스턴스를 선택한 후 인스턴스를 하나 선택합니다. 인스턴스에 인스턴스 중지 또는 인스턴스 만료 이벤트가 연동되어 있는 경우 해당 이벤트가 하단 창에 표시됩니다.

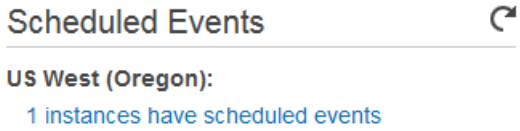
기존 콘솔

콘솔을 사용해 인스턴스에 예약된 이벤트를 확인하는 방법

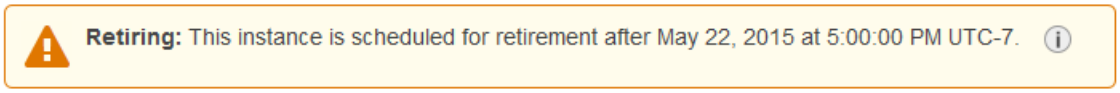
1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 다음 화면에서 예약된 이벤트를 볼 수 있습니다.
   - 탐색 창에서 이벤트를 선택합니다. 연동되어 있는 이벤트와 함께 모든 리소스가 표시됩니다. 표시 방식은 리소스 유형이나 특정 이벤트 유형으로 필터링할 수 있습니다. 또한 리소스를 선택하여 세부 정보를 확인할 수도 있습니다.



   - 또는 탐색 창에서 EC2 대시보드를 선택합니다. 연동되어 있는 이벤트와 함께 모든 리소스가 예약된 이벤트 아래 표시됩니다.



   - 일부 이벤트가 영향을 받는 리소스에 대해서도 표시됩니다. 예를 들어 탐색 창에서 인스턴스를 선택한 후 인스턴스를 하나 선택합니다. 인스턴스에 인스턴스 중지 또는 인스턴스 만료 이벤트가 연동되어 있는 경우 해당 이벤트가 하단 창에 표시됩니다.



AWS CLI

AWS CLI을 사용해 인스턴스에 예약된 이벤트를 확인하는 방법

describe-instance-status 명령을 사용합니다.

```
aws ec2 describe-instance-status --instance-id i-1234567890abcdef0 --query
 "InstanceStatuses[].Events"
```

다음 예제 출력은 재부팅 이벤트를 보여줍니다.

```
[
    "Events": [
        {
            "InstanceEventId": "instance-event-0d59937288b749b32",
            "Code": "system-reboot",
            "Description": "The instance is scheduled for a reboot",
            "NotAfter": "2019-03-15T22:00:00.000Z",
            "NotBefore": "2019-03-14T20:00:00.000Z",
            "NotBeforeDeadline": "2019-04-05T11:00:00.000Z"
         }

    ]
]
```

다음 예제 출력은 인스턴스 만료 이벤트를 표시합니다.

```
[
    "Events": [
        {
            "InstanceEventId": "instance-event-0e439355b779n26",

            "Code": "instance-stop",
            "Description": "The instance is running on degraded hardware",
            "NotBefore": "2015-05-23T00:00:00.000Z"
        }
    ]
]
```

PowerShell

Windows PowerShell용 AWS 도구을 사용해 인스턴스에 예약된 이벤트를 확인하는 방법

다음 Get-EC2InstanceStatus 명령을 사용합니다.

```
PS C:\> (Get-EC2InstanceStatus -InstanceId i-1234567890abcdef0).Events
```

다음 예제 출력은 인스턴스 만료 이벤트를 표시합니다.

```
Code          : instance-stop
Description   : The instance is running on degraded hardware
NotBefore     : 5/23/2015 12:00:00 AM
```

Instance metadata

인스턴스 메타데이터를 사용해 인스턴스에 예약된 이벤트를 확인하는 방법

인스턴스 메타데이터 서비스 버전 2 또는 인스턴스 메타데이터 서비스 버전 1을 사용하여 인스턴스 메타데이터 (p. 571)에서 인스턴스에 대해 활성화된 유지 관리 이벤트 정보를 검색할 수 있습니다.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-
ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-
data/events/maintenance/scheduled
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/events/maintenance/scheduled
```

다음은 예약된 시스템 재부팅 이벤트에 관한 정보를 JSON 형식으로 표시하는 예제 출력입니다.

```
[
  {
    "NotBefore" : "21 Jan 2019 09:00:43 GMT",
    "Code" : "system-reboot",
    "Description" : "scheduled reboot",
    "EventId" : "instance-event-0d59937288b749b32",
    "NotAfter" : "21 Jan 2019 09:17:23 GMT",
    "State" : "active"
  }
]
```

인스턴스 메타데이터를 사용하여 인스턴스에 대해 완료되거나 취소된 이벤트 관련 이벤트 기록을 확인하는 방법

인스턴스 메타데이터 서비스 버전 2 또는 인스턴스 메타데이터 서비스 버전 1을 사용하여 인스턴스 메타데이터 (p. 571)에서 인스턴스에 대해 완료되거나 취소된 이벤트 관련 정보를 검색할 수 있습니다.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-
ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-
data/events/maintenance/history
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/events/maintenance/history
```

다음은 취소된 시스템 재부팅 이벤트와 완료된 시스템 재부팅 이벤트 관련 정보를 JSON 형식으로 표현하는 예제 출력입니다.

```
[
  {
    "NotBefore" : "21 Jan 2019 09:00:43 GMT",
    "Code" : "system-reboot",
    "Description" : "[Canceled] scheduled reboot",
    "EventId" : "instance-event-0d59937288b749b32",
    "NotAfter" : "21 Jan 2019 09:17:23 GMT",
    "State" : "canceled"
  },
  {
    "NotBefore" : "29 Jan 2019 09:00:43 GMT",
    "Code" : "system-reboot",
    "Description" : "[Completed] scheduled reboot",
    "EventId" : "instance-event-0d59937288b749b32",
    "NotAfter" : "29 Jan 2019 09:17:23 GMT",
    "State" : "completed"
  }
]
```

중지 또는 만료 예약된 인스턴스 관련 작업

AWS가 인스턴스의 기본 호스트에서 복구 불가능한 결함을 감지하면 인스턴스의 루트 디바이스 유형에 따라 인스턴스의 중지 또는 종료를 예약합니다. 루트 디바이스가 EBS 볼륨이면 인스턴스 중단이 예약됩니다. 그

렇지 않고 루트 디바이스가 인스턴스 스토어 볼륨이면 인스턴스 종료가 예약됩니다. 자세한 내용은 인스턴스 만료 (p. 520) 단원을 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 중단 또는 종료되기 전에 반드시 저장하십시오.

Amazon EBS에서 지원되는 인스턴스 작업

인스턴스가 예약 시간에 중단될 때까지 기다릴 수 있습니다. 또는 직접 인스턴스를 중지한 후 시작하여 새 호스트로 마이그레이션하는 것도 가능합니다. 인스턴스 중단과 중단 후 인스턴스 구성을 변경하는 방법에 대한 자세한 내용은 인스턴스 중지 및 시작 (p. 507)을 참조하십시오.

예약된 인스턴스 중지 이벤트에 대한 응답으로 즉시 중지 및 시작을 자동화할 수 있습니다. 자세한 내용은 AWS Health 사용 설명서의 EC2 인스턴스에 대한 작업 자동화를 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 기본 호스트 유지 관리 등의 작업이 필요할 때는 인스턴스 또는 인스턴스의 기본 호스트가 재부팅되도록 예약할 수 있습니다. 적합한 특정 날짜 및 시간에 인스턴스가 재부팅되도록 대부분의 재부팅 이벤트를 다시 예약 (p. 614)할 수 있습니다.

링크된 EC2-Classic 인스턴스 (p. 780)를 중지하면 자동으로 VPC에서 링크가 해제되고 VPC 보안 그룹이 인스턴스와 더 이상 연결되지 않습니다. 인스턴스를 다시 시작한 후 VPC에 다시 연결할 수 있습니다.

재부팅 이벤트 유형 보기

다음 방법 중 하나를 사용하여 재부팅 이벤트가 인스턴스 재부팅인지 아니면 시스템 재부팅인지 확인할 수 있습니다.

새로운 콘솔

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. 필터 목록에서 리소스 유형: 인스턴스를 선택합니다.
4. 각 인스턴스의 이벤트 유형 열에서 값을 확인합니다. 값은 system-reboot 또는 instance-reboot입니다.

기존 콘솔

콘솔에서 예약된 재부팅 이벤트의 유형을 확인하는 방법

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

3. 필터 목록에서 인스턴스 리소스를 선택합니다.
4. 각 인스턴스의 이벤트 유형 열에서 값을 확인합니다. 값은 system-reboot 또는 instance-reboot입니다.

AWS CLI

AWS CLI를 사용하여 예약된 재부팅 이벤트의 유형을 확인하는 방법

describe-instance-status 명령을 사용합니다.

```
aws ec2 describe-instance-status --instance-id i-1234567890abcdef0
```

예약된 재부팅 이벤트의 경우 `Code`의 값은 `system-reboot` 또는 `instance-reboot`입니다. 다음 예제 출력은 `system-reboot` 이벤트를 보여줍니다.

```
[
    "Events": [
                {
                    "InstanceEventId": "instance-event-0d59937288b749b32",
                    "Code": "system-reboot",
                    "Description": "The instance is scheduled for a reboot",
                    "NotAfter": "2019-03-14T22:00:00.000Z",
                    "NotBefore": "2019-03-14T20:00:00.000Z",
                    "NotBeforeDeadline": "2019-04-05T11:00:00.000Z"
                }
               ]
]
```

인스턴스 재부팅 작업

예약된 유지 관리 기간 내에 인스턴스 재부팅이 실행될 때까지 기다리거나, 적합한 특정 날짜 및 시간으로 인스턴스 재부팅을 다시 예약 (p. 614)하거나, 편리한 시간에 직접 인스턴스를 재부팅 (p. 520)할 수 있습니다.

인스턴스가 재부팅된 후 예약된 이벤트가 삭제되고 이벤트 설명이 업데이트됩니다. 기본 호스트에서 보류되었던 점검이 완료되면 부팅이 완전히 끝난 이후에 인스턴스를 다시 사용할 수 있습니다.

시스템 재부팅 작업

시스템은 직접 재부팅할 수 없습니다. 예약된 유지 관리 기간 내에 시스템이 재부팅될 때까지 기다리거나, 적합한 날짜 및 시간으로 시스템 재부팅을 다시 예약 (p. 614)할 수 있습니다. 시스템 재부팅은 보통 분 단위로 완료됩니다. 시스템 재부팅이 발생한 후 인스턴스는 해당 IP 주소 및 DNS 이름을 그대로 유지하고 로컬 인스턴스 스토어 볼륨의 데이터가 보존됩니다. 시스템 재부팅이 완료되면 인스턴스에 예약된 이벤트가 삭제되며, 인스턴스 소프트웨어가 예상대로 실행되는지 확인할 수 있습니다.

또는 인스턴스를 다른 시간에 유지 관리해야 하며 시스템 재부팅을 다시 예약할 수 없는 경우 Amazon EBS 지원 인스턴스를 중지한 후 시작하여 새 호스트로 마이그레이션하는 것이 가능합니다. 그러나 로컬 인스턴스 스토어 볼륨에 저장된 데이터가 손실됩니다. 또한 예약된 시스템 재부팅 이벤트에 대한 응답으로 즉시 인스턴스 중지 및 시작을 자동화할 수 있습니다. 자세한 내용은 AWS Health 사용 설명서의 EC2 인스턴스에 대한 작업 자동화를 참조하십시오. 인스턴스 스토어 지원 인스턴스의 경우 시스템 재부팅을 다시 예약할 수 없는 경우 가장 최근 AMI에서 교체 인스턴스를 시작하고 예약된 유지 관리 기간 이전에 필요한 데이터를 모두 교체 인스턴스로 마이그레이션한 다음 원본 인스턴스를 종료할 수 있습니다.

인스턴스의 유지 관리 예약 작업

AWS에서 인스턴스의 기본 호스트를 유지 관리해야 하는 경우 인스턴스의 유지 관리가 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 인스턴스에 더 이상 예약된 이벤트가 없거나, 있는 경우 예약된 이벤트가 [완료]로 표시됩니다. 인스턴스 상태 설명을 새로 고치는 데 최대 1시간이 걸리는 경우도 있습니다. 완료된 유지 관리 이벤트는 Amazon EC2 콘솔 대시보드에 일주일까지 표시됩니다.

Amazon EBS에서 지원되는 인스턴스 작업

예약 시간에 유지 관리가 실행될 때까지 기다릴 수 있습니다. 또는 인스턴스를 중지한 후 시작하여 새 호스트로 마이그레이션하는 것도 가능합니다. 인스턴스 중단과 중단 후 인스턴스 구성을 변경하는 방법에 대한 자세한 내용은 인스턴스 중지 및 시작 (p. 507)을 참조하십시오.

예약된 유지 관리 이벤트에 대한 응답으로 즉시 중지 및 시작을 자동화할 수 있습니다. 자세한 내용은 AWS Health 사용 설명서의 EC2 인스턴스에 대한 작업 자동화를 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.

예약된 이벤트 다시 예약

이벤트가 적합한 특정 날짜와 시간에 발생하도록 다시 예약할 수 있습니다. 기한 날짜가 있는 이벤트만 다시 예약할 수 있습니다. 이벤트를 다시 예약할 때 제한 사항 (p. 615)이 더 있습니다.

다음 방법 중 하나를 사용하여 이벤트를 다시 예약할 수 있습니다.

새로운 콘솔

콘솔을 사용하여 이벤트를 다시 예약하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Events]를 선택합니다.
3. 필터 목록에서 리소스 유형: 인스턴스를 선택합니다.
4. 하나 이상의 인스턴스를 선택한 다음 작업, Schedule event(이벤트 예약)를 선택합니다.

   Deadline(기한) 값으로 표시되는 이벤트 기한이 있는 이벤트만 다시 예약할 수 있습니다. 선택한 이벤트 중 하나에 기한 날짜가 없으면 작업, Schedule event(이벤트 예약)가 비활성화됩니다.
5. New start time(새 시작 시간)에 이벤트에 대한 새 날짜 및 시간을 입력합니다. 새 날짜 및 시간은 Event deadline(이벤트 기한) 이전이어야 합니다.
6. 저장을 선택합니다.

   업데이트된 이벤트 시작 시간이 콘솔에 반영되려면 1-2분의 시간이 걸릴 수 있습니다.

기존 콘솔

콘솔을 사용하여 이벤트를 다시 예약하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 이벤트를 선택합니다.
3. 필터 목록에서 인스턴스 리소스를 선택합니다.

4. 하나 이상의 인스턴스를 선택한 다음 작업, Schedule Event(이벤트 예약)를 선택합니다.

   Event Deadline(이벤트 기한) 값으로 표시되는 이벤트 기한이 있는 이벤트만 다시 예약할 수 있습니다.
5. Event start time(이벤트 시작 시간)에 이벤트에 대한 새 날짜 및 시간을 입력합니다. 새 날짜 및 시간은 Event Deadline(이벤트 기한) 이전이어야 합니다.
6. Schedule Event(이벤트 예약)를 선택합니다.

   업데이트된 이벤트 시작 시간이 콘솔에 반영되려면 1-2분의 시간이 걸릴 수 있습니다.

AWS CLI

AWS CLI를 사용하여 이벤트를 다시 예약하려면

1. `NotBeforeDeadline` 값으로 표시되는 이벤트 기한이 있는 이벤트만 다시 예약할 수 있습니다. describe-instance-status 명령을 사용하여 `NotBeforeDeadline` 파라미터 값을 확인합니다.

```
aws ec2 describe-instance-status --instance-id i-1234567890abcdef0
```

   다음 예제 출력에서는 `NotBeforeDeadline`에 값이 포함되므로 다시 예약할 수 있는 `system-reboot` 이벤트를 보여줍니다.

```
[
    "Events": [
                {
                    "InstanceEventId": "instance-event-0d59937288b749b32",
                    "Code": "system-reboot",
                    "Description": "The instance is scheduled for a reboot",
                    "NotAfter": "2019-03-14T22:00:00.000Z",
                    "NotBefore": "2019-03-14T20:00:00.000Z",
                    "NotBeforeDeadline": "2019-04-05T11:00:00.000Z"
                }
              ]
]
```

2. 이벤트를 다시 예약하려면 modify-instance-event-start-time 명령을 사용합니다. `not-before` 파라미터를 사용하여 새 이벤트 시작 시간을 지정합니다. 새 이벤트 시작 시간은 `NotBeforeDeadline` 이전이어야 합니다.

```
aws ec2 modify-instance-event-start-time --instance-id i-1234567890abcdef0
 --instance-event-id instance-event-0d59937288b749b32 --not-
before 2019-03-25T10:00:00.000
```

   describe-instance-status 명령에서 업데이트된 `not-before` 파라미터 값을 반환하는 데 1-2분의 시간이 걸릴 수 있습니다.

제한 사항

- 이벤트 기한이 있는 이벤트만 다시 예약할 수 있습니다. 이 이벤트는 이벤트 기한까지 다시 예약될 수 있습니다. 콘솔의 Deadline(기한) 열 및 AWS CLI의 `NotBeforeDeadline` 필드는 이벤트에 기한이 있음을 나타냅니다.
- 아직 시작하지 않은 이벤트만 다시 예약할 수 있습니다. 콘솔의 시작 시간 열 및 AWS CLI의 `NotBefore` 필드는 이벤트 시작 시간을 나타냅니다. 다음 5분 내에 시작하도록 예약된 이벤트는 다시 예약할 수 없습니다.
- 새 이벤트 시작 시간은 현재 시간에서 최소 60분 내여야 합니다.

- 콘솔을 사용하여 여러 이벤트를 다시 예약하는 경우 이벤트 기한은 가장 이른 이벤트 기한의 이벤트에 의해 결정됩니다.

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. 616) 단원을 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 설명서를 참조하십시오.

내용

- 인스턴스에 대한 세부 모니터링 활성화 또는 비활성화 (p. 616)
- 인스턴스에 대해 얻을 수 있는 CloudWatch 측정치 나열 (p. 617)
- 인스턴스에 대한 측정치 통계 보기 (p. 628)
- 인스턴스에 대한 그래프 지표 (p. 636)
- 인스턴스에 대해 CloudWatch 경보 만들기 (p. 636)
- 인스턴스를 중지, 종료, 재부팅 또는 복구하는 경보 만들기 (p. 637)

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.

모니터링 유형	설명
기본	자동으로 5분 기간 동안 데이터를 무료로 사용할 수 있습니다.
세부	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. 요금에 대한 자세한 내용은 Amazon CloudWatch 제품 페이지를 참조하십시오.

세부 모니터링 활성화

인스턴스를 시작할 때 또는 인스턴스가 실행 중이거나 중지된 후에 인스턴스에 대한 세부 모니터링을 활성화할 수 있습니다. 인스턴스에 대한 세부 모니터링 기능을 활성화해도 인스턴스에 연결된 EBS 볼륨 모니터링

에는 영향을 주지 않습니다. 자세한 내용은 Amazon EBS의 Amazon CloudWatch 지표 (p. 1011) 단원을 참조하십시오.

기존 인스턴스에 대한 세부 모니터링을 활성화하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 작업, CloudWatch 모니터링, 세부 모니터링 활성화를 차례로 선택합니다.
4. 세부 모니터링 활성화 대화 상자에서 예, 활성화를 선택합니다.
5. 닫기를 선택합니다.

인스턴스 시작 시 세부 모니터링을 활성화하려면(콘솔)

AWS Management 콘솔을 사용해 인스턴스를 시작할 때 Configure Instance Details(인스턴스 세부 정보 구성) 페이지에서 모니터링에 있는 확인란을 선택합니다.

기존 인스턴스에 대한 세부 모니터링을 활성화하려면(AWS CLI)

다음 monitor-instances 명령을 사용하여 지정된 인스턴스에 대한 세부 모니터링을 활성화합니다.

```
aws ec2 monitor-instances --instance-ids i-1234567890abcdef0
```

인스턴스 시작 시 세부 모니터링을 활성화하려면(AWS CLI)

`--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 중이거나 중지된 후에 인스턴스에 대한 세부 모니터링을 비활성화할 수 있습니다.

세부 모니터링을 비활성화하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 작업, CloudWatch 모니터링, 세부 모니터링 비활성화를 차례로 선택합니다.
4. 세부 모니터링 비활성화 대화 상자에서 예, 비활성화를 선택합니다.
5. 닫기를 선택합니다.

세부 모니터링을 비활성화하려면(AWS CLI)

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 콘솔, AWS CLI 또는 API를 사용하여 Amazon EC2가 CloudWatch에 전송하는 측정치를 나열할 수 있습니다. 기본적으로 각 데이터

요소는 인스턴스의 시작 시간 이후 5분간 활동을 다룹니다. 세부 모니터링을 활성화한 경우 각 데이터 요소는 시작 시간부터 1분간 활동을 다룹니다.

이 측정치에 대한 통계를 얻는 방법에 대한 자세한 내용은 인스턴스에 대한 측정치 통계 보기 (p. 628) 단원을 참조하십시오.

목차

- 인스턴스 지표 (p. 618)
- CPU 크레딧 지표 (p. 620)
- Nitro 기반 인스턴스에 대한 Amazon EBS 지표 (p. 621)
- 상태 확인 지표 (p. 623)
- 트래픽 미러링 지표 (p. 623)
- Amazon EC2 지표 차원 (p. 623)
- Amazon EC2 사용량 지표 (p. 624)
- 콘솔을 사용해 측정치 나열하기 (p. 625)
- AWS CLI를 사용해 측정치 나열하기 (p. 627)

인스턴스 지표

`AWS/EC2` 네임스페이스에는 다음 인스턴스 지표가 포함되어 있습니다.

지표	설명
`CPUUtilization`	인스턴스에서 현재 사용 중인 할당된 EC2 컴퓨팅 유닛(ECU)의 비율(%)입니다. 이 지표는 선택한 인스턴스에서 애플리케이션을 실행하는 데 필요한 처리 능력을 식별합니다. 인스턴스 유형에 따라, 인스턴스에 전체 프로세스 코어가 할당되지 않았을 때 운영 체제의 도구에서 비율이 CloudWatch보다 낮게 표시할 수 있습니다. 단위: 백분율
`DiskReadOps`	지정된 시간 내에 인스턴스에 사용할 수 있는 모든 인스턴스 스토어 볼륨에서 읽기 작업 완료. 기간의 평균 IOPS(초당 I/O 작업 수)를 계산하려면 기간의 총 작업 수를 해당 기간의 초 수로 나누십시오. 인스턴스 스토어 볼륨이 없으면 값이 0이거나 측정치가 보고되지 않습니다. 단위: 개수
`DiskWriteOps`	지정된 시간 내에 인스턴스에 사용할 수 있는 모든 인스턴스 스토어 볼륨에 대한 쓰기 작업 완료. 기간의 평균 IOPS(초당 I/O 작업 수)를 계산하려면 기간의 총 작업 수를 해당 기간의 초 수로 나누십시오. 인스턴스 스토어 볼륨이 없으면 값이 0이거나 측정치가 보고되지 않습니다. 단위: 개수

지표	설명
`DiskReadBytes`	인스턴스에 사용할 수 있는 모든 인스턴스 스토어 볼륨에서 읽은 바이트 수. 이 지표는 애플리케이션이 인스턴스의 하드 디스크에서 읽는 데이터 볼륨을 결정하는 데 사용됩니다. 이를 사용하여 애플리케이션의 속도를 결정할 수 있습니다. 보고된 숫자는 해당 기간에 수신된 바이트 수입니다. 기본(5분) 모니터링을 사용하는 경우, 이 숫자를 300으로 나누어 바이트/초를 찾을 수 있습니다. 세부(1분) 모니터링의 경우 60으로 나눕니다. 인스턴스 스토어 볼륨이 없으면 값이 0이거나 측정치가 보고되지 않습니다. 단위: 바이트
`DiskWriteBytes`	인스턴스에 사용할 수 있는 모든 인스턴스 스토어 볼륨에 쓴 바이트 수. 이 지표는 애플리케이션이 인스턴스의 하드 디스크에 쓰는 데이터 볼륨을 결정하는 데 사용됩니다. 이를 사용하여 애플리케이션의 속도를 결정할 수 있습니다. 보고된 숫자는 해당 기간에 수신된 바이트 수입니다. 기본(5분) 모니터링을 사용하는 경우, 이 숫자를 300으로 나누어 바이트/초를 찾을 수 있습니다. 세부(1분) 모니터링의 경우 60으로 나눕니다. 인스턴스 스토어 볼륨이 없으면 값이 0이거나 측정치가 보고되지 않습니다. 단위: 바이트
`NetworkIn`	인스턴스가 모든 네트워크 인터페이스에서 받은 바이트 수입니다. 이 측정치는 단일 인스턴스로 들어오는 네트워크 트래픽의 볼륨을 식별합니다. 보고된 숫자는 해당 기간에 수신된 바이트 수입니다. 기본(5분) 모니터링을 사용하는 경우, 이 숫자를 300으로 나누어 바이트/초를 찾을 수 있습니다. 세부(1분) 모니터링의 경우 60으로 나눕니다. 단위: 바이트
`NetworkOut`	인스턴스가 모든 네트워크 인터페이스에서 보낸 바이트 수입니다. 이 측정치는 단일 인스턴스에서 나가는 네트워크 트래픽의 볼륨을 식별합니다. 보고된 숫자는 해당 기간에 전송된 바이트 수입니다. 기본(5분) 모니터링을 사용하는 경우, 이 숫자를 300으로 나누어 바이트/초를 찾을 수 있습니다. 세부(1분) 모니터링의 경우 60으로 나눕니다. 단위: 바이트

지표	설명
`NetworkPacketsIn`	인스턴스가 모든 네트워크 인터페이스에서 받은 패킷 수입니다. 이 지표는 단일 인스턴스에서 수신 트래픽의 볼륨을 패킷 수 기준으로 식별합니다. 기본 모니터링에서만 이 지표를 사용할 수 있습니다. 단위: 개수 Statistics: Minimum, Maximum, Average
`NetworkPacketsOut`	인스턴스가 모든 네트워크 인터페이스에서 보낸 패킷 수입니다. 이 지표는 단일 인스턴스에서 발신 트래픽의 볼륨을 패킷 수 기준으로 식별합니다. 기본 모니터링에서만 이 지표를 사용할 수 있습니다. 단위: 개수 Statistics: Minimum, Maximum, Average
`MetadataNoToken`	토큰을 사용하지 않는 방법으로 인스턴스 메타데이터 서비스에 성공적으로 액세스한 횟수입니다. 이 메트릭은 인스턴스 메타데이터 서비스 버전 1을 사용 중인 인스턴스 메타데이터에 토큰을 사용하지 않고 액세스하는 프로세스가 있는지 확인하는 데 사용됩니다. 모든 요청이 토큰 지원 세션을 사용하는 경우(예: 인스턴스 메타데이터 서비스 버전 2) 값은 0입니다. 자세한 내용은 인스턴스 메타데이터 서비스 버전 2 사용으로 전환 (p. 573) 단원을 참조하십시오. 단위: 개수

CPU 크레딧 지표

`AWS/EC2` 네임스페이스에는 성능 순간 확장 가능 인스턴스 (p. 194)에 대한 다음 CPU 크레딧 지표가 포함되어 있습니다.

지표	설명
`CPUCreditUsage`	CPU 사용률을 위해 인스턴스에서 소비되는 CPU 크레딧의 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 `Sum` 통계 대신 `Average` 통계를 사용하십시오. 단위: 크레딧 (vCPU-분)
`CPUCreditBalance`	시작 이후 인스턴스가 누적한 획득 CPU 크레딧 수입니다. T2 스탠다드의 경우 `CPUCreditBalance`에 누적된 시작 크레딧 수도 포함됩니다. 크레딧은 획득 이후에 크레딧 밸런스에 누적되고, 소비 시 크레딧 밸런스에서 소멸됩니다. 크레딧 밸런스는 최대 한도(인스턴스 크기에 따라 결정)가 있습니다. 한도에 도달하면 새로 획득한 크레딧이 모두 삭제됩니다. T2 스탠다드의 경우 시작 크레딧은 한도에 포함되지 않습니다.

지표	설명
	`CPUCreditBalance`의 크레딧은 인스턴스가 기준 CPU 사용률 이상으로 버스터를 하는 데 소비할 수 있습니다. 인스턴스가 실행 중인 동안 `CPUCreditBalance`의 크레딧은 만료되지 않습니다. T3 또는 T3a 인스턴스가 중지되면 `CPUCreditBalance` 값은 7일 동안 지속됩니다. 그 이후에는 누적된 크레딧이 모두 삭제됩니다. T2 인스턴스가 중지되면 `CPUCreditBalance` 값은 지속되지 않고 누적된 크레딧이 모두 삭제됩니다. CPU 크레딧 측정치는 5분 간격으로만 제공됩니다. 단위: 크레딧 (vCPU-분)
`CPUSurplusCreditBalance`	`CPUCreditBalance` 값이 0일 때 `unlimited` 인스턴스에서 소비된 잉여 크레딧의 수 입니다. 획득한 CPU 크레딧에 따라 `CPUSurplusCreditBalance` 값이 청산됩니다. 잉여 크레딧의 수가 인스턴스가 24시간 동안 획득할 수 있는 최대 크레딧 수를 초과한 경우 최대 값 이상으로 소비된 잉여 크레딧은 추가 요금으로 부과됩니다. CPU 크레딧 측정치는 5분 간격으로만 제공됩니다. 단위: 크레딧 (vCPU-분)
`CPUSurplusCreditsCharged`	획득한 CPU 크레딧으로 청산되지 않는 소비 잉여 크레딧의 수로, 추가 요금으로 부과됩니다. 소비된 잉여 크레딧은 다음이 발생할 때 요금이 부과됩니다. • 소비한 잉여 크레딧이 인스턴스가 24시간 동안 획득할 수 있는 최대 크레딧 수를 초과하는 경우. 해당 시간이 끝날 때 최대 값 이상으로 소비한 잉여 크레딧에 요금이 부과됩니다. • 인스턴스가 중지 또는 종료된 경우. • 인스턴스가 `unlimited`에서 `standard`로 전환됩니다. CPU 크레딧 측정치는 5분 간격으로만 제공됩니다. 단위: 크레딧 (vCPU-분)

Nitro 기반 인스턴스에 대한 Amazon EBS 지표

`AWS/EC2` 네임스페이스에는 베어 메탈 인스턴스가 아닌 Nitro 기반 인스턴스에 대한 다음 Amazon EBS 측정치가 포함되어 있습니다. Nitro 기반 인스턴스 유형의 목록은 Nitro 기반 인스턴스 (p. 181) 단원을 참조하십시오.

Nitro 기반 인스턴스에 대한 측정치 값은 항상 정수인 반면에 Xen 기반 인스턴스의 값은 소수를 지원합니다. 따라서 Nitro 기반 인스턴스의 낮은 인스턴스 CPU 사용률은 반내림되어 0으로 나타날 수 있습니다.

지표	설명
`EBSReadOps`	지정된 기간 내에 인스턴스에 연결된 모든 Amazon EBS 볼륨에서 완료된 읽기 작업입니다.

지표	설명
	해당 기간의 초당 평균 읽기 I/O 작업 수(읽기 IOPS)를 계산하려면 해당 기간의 총 작업 수를 해당 기간의 초 수로 나누십시오. 기본(5분) 모니터링을 사용하는 경우, 이 숫자를 300으로 나누어 읽기 IOPS를 계산할 수 있습니다. 세부(1분) 모니터링의 경우 60으로 나눕니다. 단위: 수
`EBSWriteOps`	지정된 기간 내에 인스턴스에 연결된 모든 EBS 볼륨으로의 완료된 쓰기 작업입니다. 해당 기간의 초당 평균 쓰기 I/O 작업 수(쓰기 IOPS)를 계산하려면 해당 기간의 총 작업 수를 해당 기간의 초 수로 나누십시오. 기본(5분) 모니터링을 사용하는 경우, 이 숫자를 300으로 나누어 쓰기 IOPS를 계산할 수 있습니다. 세부(1분) 모니터링의 경우 60으로 나눕니다. 단위: 수
`EBSReadBytes`	지정된 기간 내에 인스턴스에 연결된 모든 EBS 볼륨에서의 바이트 읽기 작업입니다. 보고된 숫자는 해당 기간에 읽은 바이트 수입니다. 기본(5분) 모니터링을 사용하는 경우, 이 숫자를 300으로 나누어 읽기 바이트/초를 찾을 수 있습니다. 세부(1분) 모니터링의 경우 60으로 나눕니다. 단위: 바이트
`EBSWriteBytes`	지정된 기간 내에 인스턴스에 연결된 모든 EBS 볼륨으로의 바이트 쓰기 작업입니다. 보고된 숫자는 해당 기간에 써진 바이트 수입니다. 기본(5분) 모니터링을 사용하는 경우, 이 숫자를 300으로 나누어 쓰기 바이트/초를 찾을 수 있습니다. 세부(1분) 모니터링의 경우 60으로 나눕니다. 단위: 바이트
`EBSIOBalance%`	더 작은 인스턴스 크기에서만 사용할 수 있습니다. 버스트 버킷에 남아 있는 I/O 크레딧의 비율에 대한 정보를 제공합니다. 기본 모니터링에서만 이 지표를 사용할 수 있습니다. `Sum` 통계는 이 지표에 적용할 수 없습니다. 단위: 백분율
`EBSByteBalance%`	더 작은 인스턴스 크기에서만 사용할 수 있습니다. 버스트 버킷에 남아 있는 처리량 크레딧의 비율에 대한 정보를 제공합니다. 기본 모니터링에서만 이 지표를 사용할 수 있습니다. `Sum` 통계는 이 지표에 적용할 수 없습니다. 단위: 백분율

EBS 볼륨에 제공되는 측정치에 대한 자세한 내용은 Amazon EBS 지표 (p. 1012) 단원을 참조하십시오. 스팟 집합에 제공되는 측정치에 대한 자세한 내용은 CloudWatch에 대한 스팟 집합 지표 (p. 359) 단원을 참조하십시오.

상태 확인 지표

`AWS/EC2` 네임스페이스에는 다음과 같은 상태 확인 지표가 포함되어 있습니다. 기본적으로 시스템 상태 지표는 1분 주기로 무료로 사용할 수 있습니다. 새로 시작된 인스턴스의 경우, 인스턴스에서 초기화 상태를 완료해야 상태 확인 지표 데이터를 얻을 수 있습니다(인스턴스가 실행 상태로 시작되는 몇 분 내). EC2 상태 확인에 대한 자세한 내용은 인스턴스 상태 확인 단원을 참조하십시오.

지표	설명
`StatusCheckFailed`	인스턴스가 마지막으로 인스턴스 상태 확인 및 시스템 상태 확인을 통과했는지 여부를 보고합니다. 이 지표는 0(통과) 또는 1(실패)이 될 수 있습니다. 기본적으로 이 지표는 1분 주기로 무료로 사용할 수 있습니다. 단위: 개수
`StatusCheckFailed_Instance`	인스턴스가 마지막으로 인스턴스 상태 확인을 통과했는지 여부를 보고합니다. 이 지표는 0(통과) 또는 1(실패)이 될 수 있습니다. 기본적으로 이 지표는 1분 주기로 무료로 사용할 수 있습니다. 단위: 개수
`StatusCheckFailed_System`	인스턴스가 마지막으로 시스템 상태 확인을 통과했는지 여부를 보고합니다. 이 지표는 0(통과) 또는 1(실패)이 될 수 있습니다. 기본적으로 이 지표는 1분 주기로 무료로 사용할 수 있습니다. 단위: 개수

트래픽 미러링 지표

`AWS/EC2` 네임스페이스에는 미러링된 트래픽에 대한 지표가 포함됩니다. 자세한 내용은 Amazon VPC 트래픽 미러링 가이드의 Amazon CloudWatch를 사용한 미러링된 트래픽 모니터링을 참조하십시오.

Amazon EC2 지표 차원

다음 차원을 사용하여 이전 표에 나열된 지표를 구체화할 수 있습니다.

차원	설명
`AutoScalingGroupName`	이 차원은 사용자가 지정된 용량 그룹의 모든 인스턴스에 대해 요청하는 데이터를 필터링합니다. Auto Scaling 그룹은 Auto Scaling를 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.

차원	설명
`ImageId`	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.
`InstanceId`	이 차원은 사용자가 식별된 인스턴스에 대해 요청하는 데이터만 필터링합니다. 이는 데이터를 모니터링할 정확한 인스턴스를 정확히 식별하는 데 도움이 됩니다.
`InstanceType`	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, m1.small 인스턴스와 m1.large 인스턴스의 데이터를 비교하여 애플리케이션에 대해 더 높은 비즈니스 가치를 가진 인스턴스를 결정할 수 있습니다. 세부 모니터링이 설정된 인스턴스에 사용할 수 있습니다.

Amazon EC2 사용량 지표

CloudWatch 사용량 지표를 사용하여 계정의 리소스 사용량을 확인할 수 있습니다. 이러한 지표를 사용하여 CloudWatch 그래프 및 대시보드에서 현재 서비스 사용량을 시각화합니다.

Amazon EC2 사용량 지표는 AWS 서비스 할당량에 해당합니다. 사용량이 서비스 할당량에 가까워지면 경고하는 경보를 구성할 수 있습니다. CloudWatch와 서비스 할당량의 통합에 대한 자세한 내용은 서비스 할당량 통합 및 사용량 지표를 참조하십시오.

Amazon EC2는 `AWS/Usage` 네임스페이스에 다음 지표를 게시합니다.

지표	설명
`ResourceCount`	계정에서 실행 중인 지정된 리소스의 수입니다. 리소스는 지표와 연결된 차원에 의해 정의됩니다. 이 지표에 대한 가장 유용한 통계는 1분 동안 사용되는 최대 리소스 수를 나타내는 `MAXIMUM`입니다.

다음 차원은 Amazon EC2에 의해 게시되는 사용량 지표를 구체화하는 데 사용됩니다.

차원	설명
`Service`	리소스가 포함된 AWS 서비스의 이름 Amazon EC2 사용량 지표의 경우 이 차원 값은 `EC2`입니다.
`Type`	보고되는 엔터티의 유형입니다. 현재 Amazon EC2 사용량 지표에 대한 유일한 유효 값은 `Resource`입니다.
`Resource`	실행 중인 리소스의 유형입니다. 현대 Amazon EC2 사용량 지표에 대한 유일한 유효 값은 실행 중인 인스턴스에 대한 정보를 반환하는 `vCPU`입니다.
`Class`	추적 중인 리소스의 클래스. `Resource` 차원의 값이 `vCPU`인 Amazon EC2 사용량 지표의 경우 유효한 값은 `Standard/OnDemand`, `F/OnDemand`, `G/OnDemand`, `Inf/OnDemand`, `P/OnDemand` 및 `X/OnDemand`입니다.

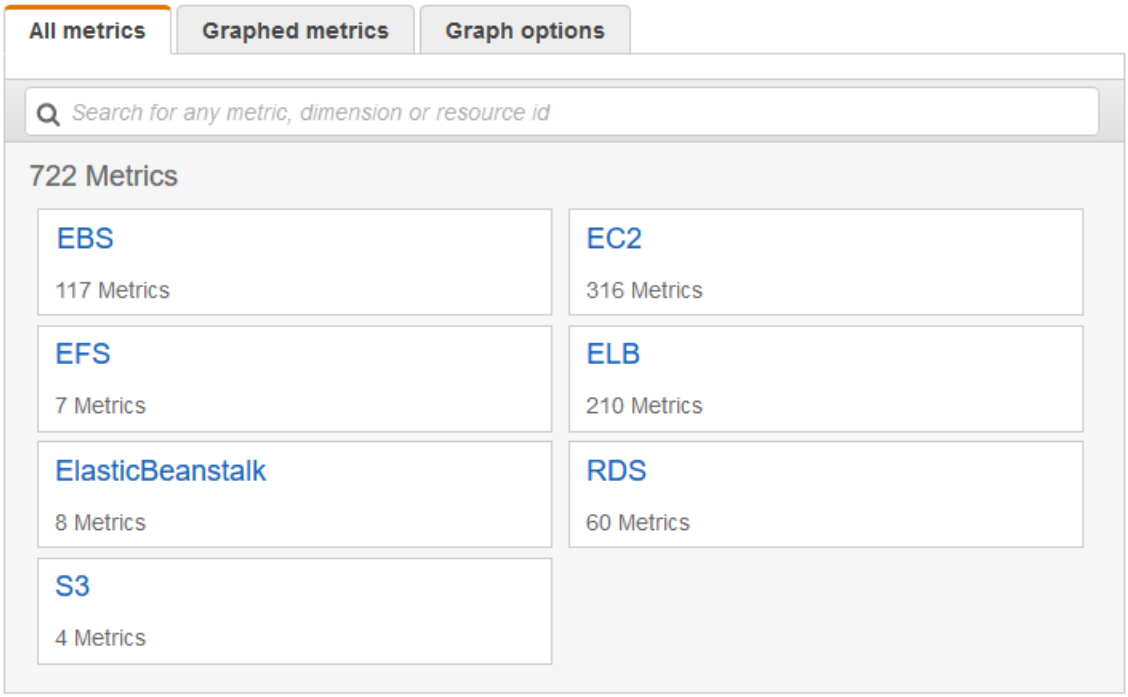
차원	설명
	이 차원의 값은 지표에서 보고하는 인스턴스 유형의 첫 글자를 정의합니다. 예를 들어, `Standard/OnDemand`는 유형이 A, C, D, H, I, M, R, T 및 Z로 시작하는 모든 실행 중인 인스턴스에 대한 정보를 반환하며 `G/OnDemand`는 유형이 G로 시작하는 모든 실행 중인 인스턴스에 대한 정보를 반환합니다.

콘솔을 사용해 측정치 나열하기

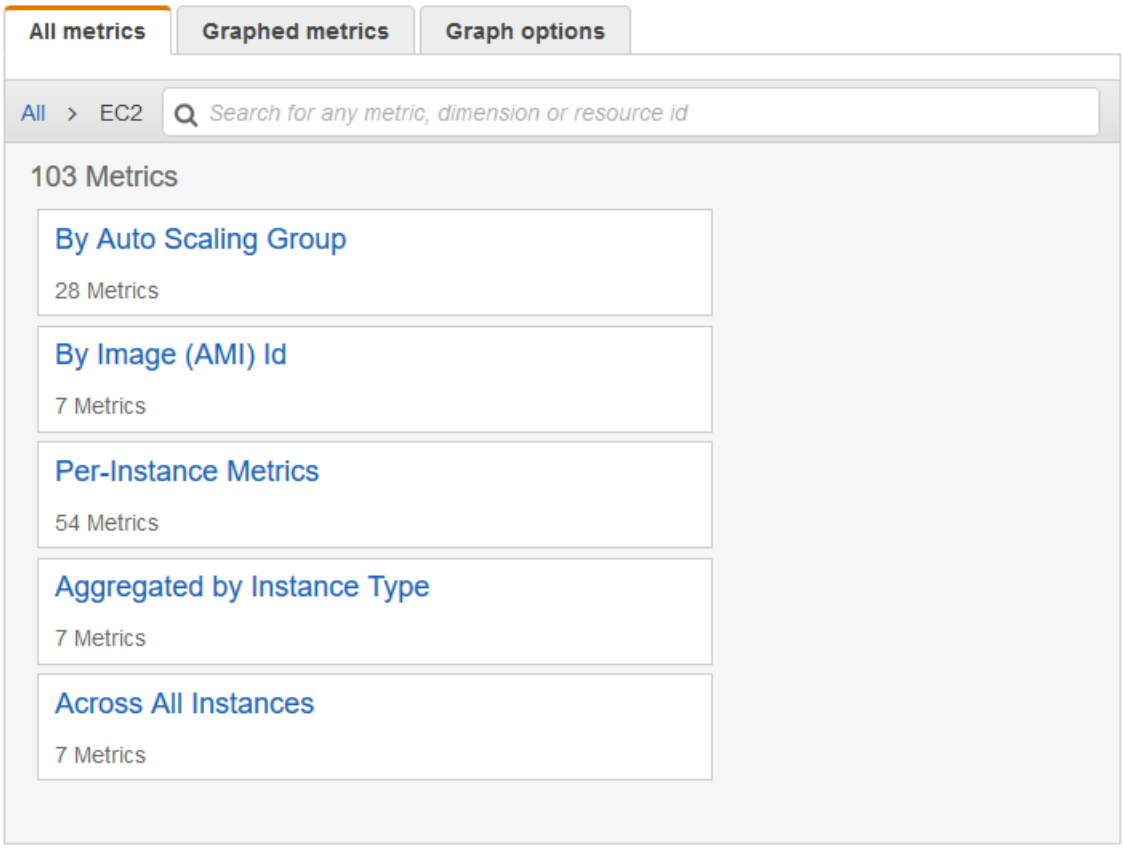
지표는 먼저 네임스페이스별로 그룹화된 다음, 각 네임스페이스 내에서 다양한 차원 조합별로 그룹화됩니다. 예를 들어, Amazon EC2에 의해 제공되는 모든 측정치나 인스턴스 ID, 인스턴스 유형, 이미지(AMI) ID 또는 Auto Scaling 그룹별로 제공되는 측정치를 볼 수 있습니다.

범주별로 사용 가능한 측정치를 보려면(콘솔)

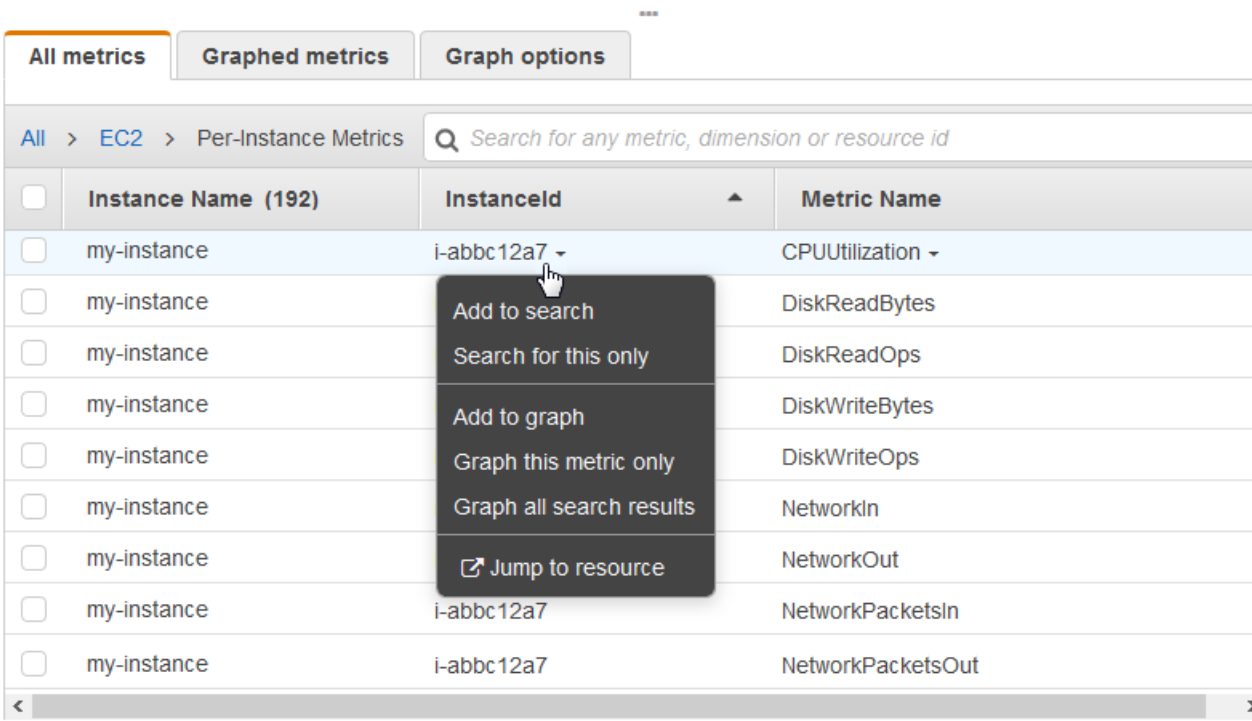
1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표를 선택합니다.
3. EC2 측정치 네임스페이스를 선택합니다.



4. 지표 차원(예: 인스턴스별 지표)을 선택합니다.



5. 측정치를 정렬하려면 열 머리글을 사용합니다. 측정치를 그래프로 표시하려면 측정치 옆에 있는 확인란을 선택합니다. 리소스로 필터링하려면 리소스 ID를 선택한 후 검색에 추가를 선택합니다. 지표로 필터링하려면 지표 이름을 선택한 후 검색에 추가를 선택합니다.



AWS CLI를 사용해 측정치 나열하기

list-metrics 명령을 사용하여 인스턴스에 대한 CloudWatch 측정치를 나열합니다.

Amazon EC2의 모든 측정치를 표시하려면(AWS CLI)

다음 예제는 Amazon EC2에 대한 모든 지표를 볼 수 있도록 `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": "NetworkIn"
    },
    ...
  ]
}
```

인스턴스에 대한 모든 측정치를 표시하려면(AWS CLI)

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(AWS CLI)

다음 예제는 지정한 지표의 결과만 보도록 `AWS/EC2` 네임스페이스와 지표 이름을 지정합니다.

```
aws cloudwatch list-metrics --namespace AWS/EC2 --metric-name CPUUtilization
```

인스턴스에 대한 측정치 통계 보기

인스턴스에 대한 CloudWatch 측정치 통계를 볼 수 있습니다.

내용

- 통계 개요 (p. 628)
- 특정 인스턴스에 대한 통계 얻기 (p. 629)
- 여러 인스턴스의 통계 집계하기 (p. 632)
- Auto Scaling 그룹별 통계 집계 (p. 634)
- AMI의 집계 통계 (p. 635)

통계 개요

통계는 지정한 기간에 걸친 지표 데이터 집계입니다. CloudWatch에서는 사용자 지정 데이터를 통해 제공되었거나 다른 AWS 서비스에서 CloudWatch에 제공한 지표 데이터 요소를 기반으로 통계를 제공합니다. 집계는 네임스페이스, 지표 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 지표에 대해 제출된 모든 값이 서로 더해진 값입니다. 이 통계는 지표의 총 볼륨을 확인할 때 유용할 수 있습니다.

통계	설명
`Average`	지정된 기간 중 `Sum`/`SampleCount`의 값입니다. 이 통계를 `Minimum` 및 `Maximum`과 비교하면 지표의 전체 범위와 평균 사용량이 `Minimum` 및 `Maximum`에 얼마나 근접했는지 확인할 수 있습니다. 이와 같은 비교를 통해 필요에 따라 리소스를 늘리거나 줄어야 하는 시점을 파악할 수 있습니다.
`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

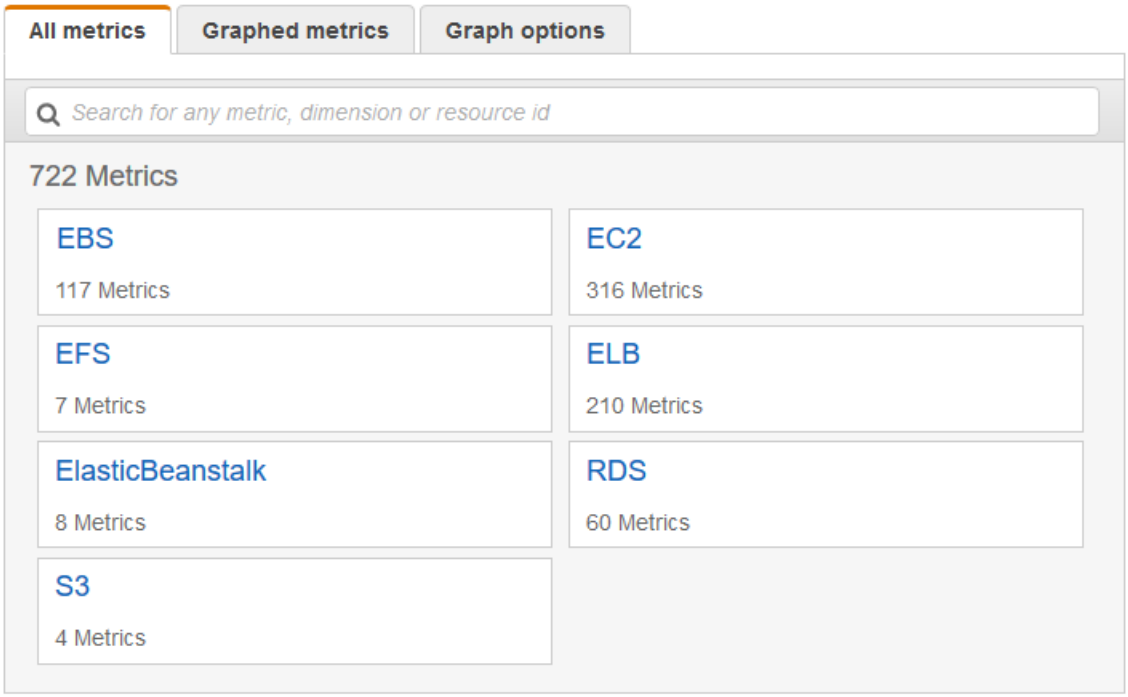
다음 예제는 AWS Management 콘솔 또는 AWS CLI 명령을 사용하여 특정 EC2 인스턴스의 최대 CPU 사용률을 확인하는 방법을 보여 줍니다.

요구 사항

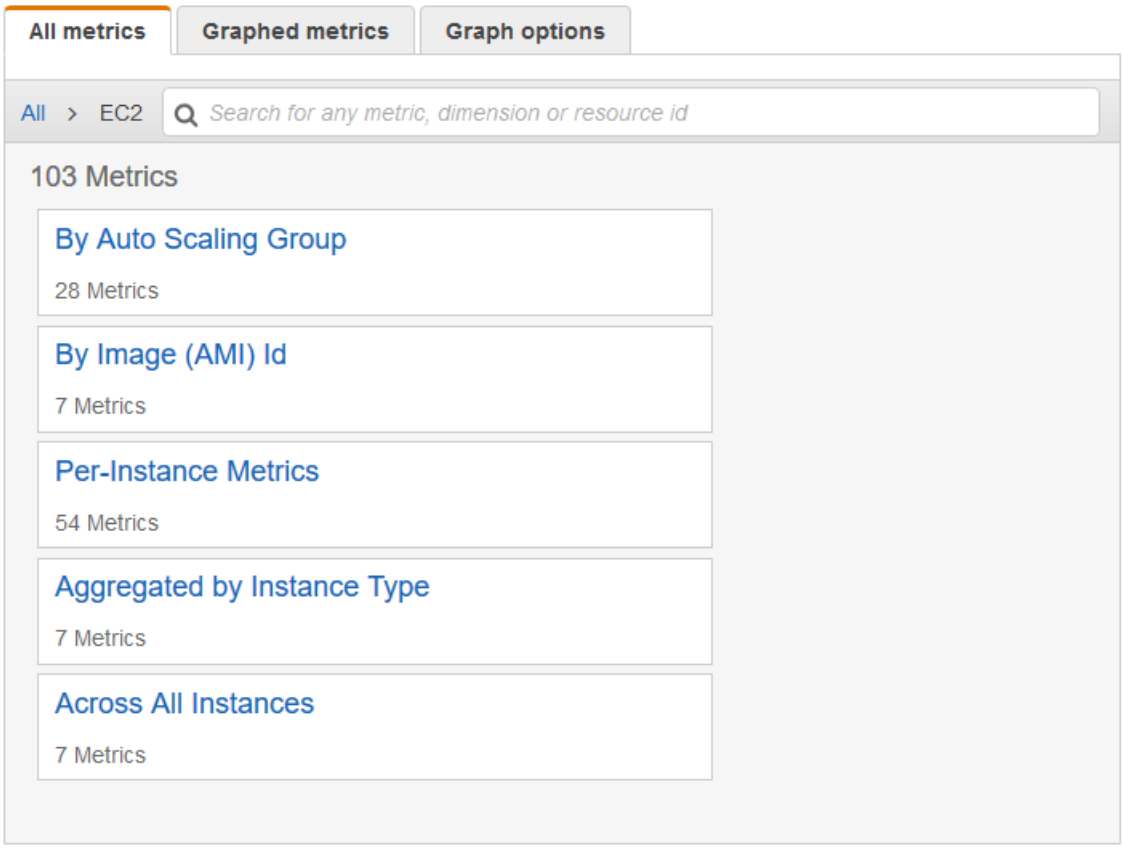
- 인스턴스의 ID가 필요합니다. 인스턴스 ID는 AWS Management 콘솔이나 describe-instances 명령을 사용하여 확인할 수 있습니다.
- 기본적으로 기본 모니터링이 사용되지만 세부 모니터링을 사용하도록 설정할 수 있습니다. 자세한 내용은 인스턴스에 대한 세부 모니터링 활성화 또는 비활성화 (p. 616) 단원을 참조하십시오.

특정 인스턴스에 대한 CPU 사용률을 표시하려면(콘솔)

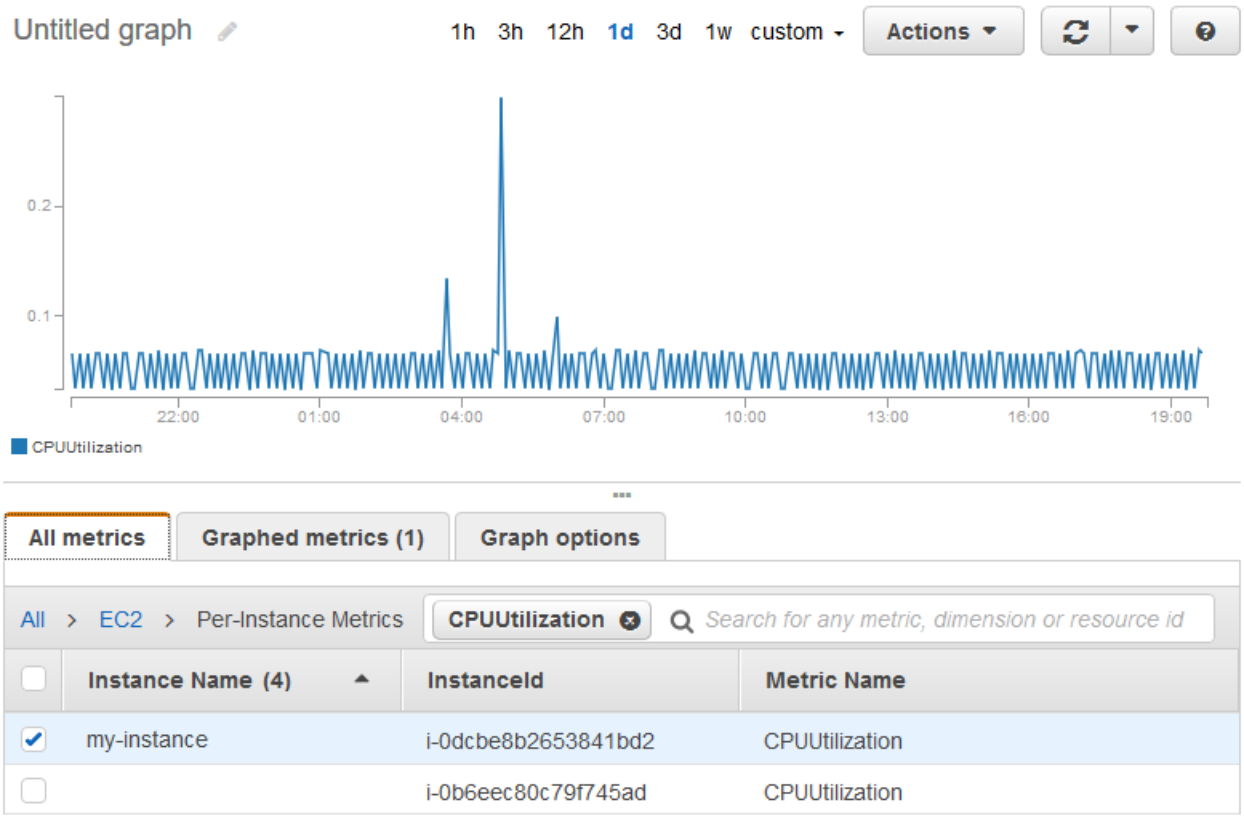
1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표를 선택합니다.
3. EC2 측정치 네임스페이스를 선택합니다.



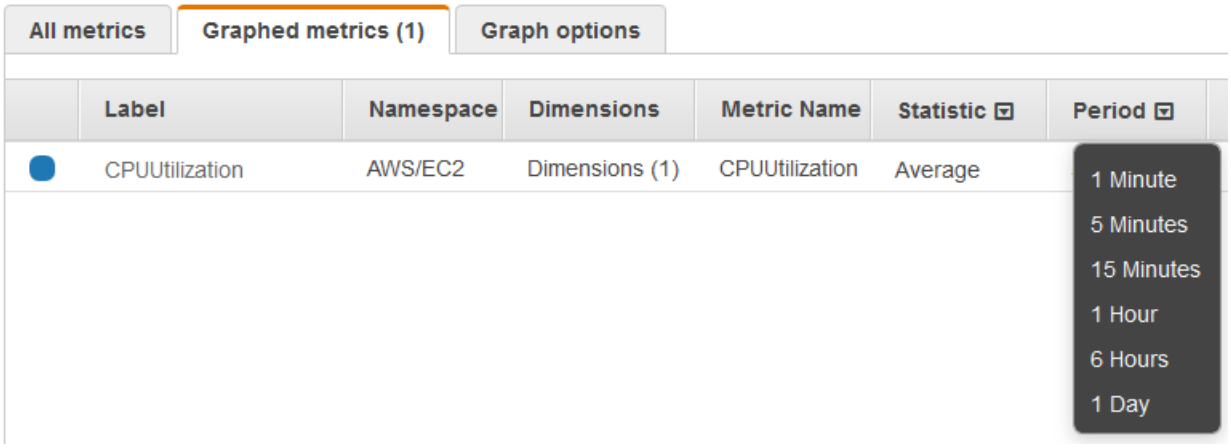
4. 인스턴스별 지표 차원을 선택합니다.



5. 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 사용자 지정을 선택합니다.



6. 측정치에 대한 통계 또는 기간을 변경하려면 그래프로 표시된 지표 탭을 선택합니다. 열 머리글이나 개별 값을 선택한 후 다른 값을 선택합니다.



특정 인스턴스에 대한 CPU 사용률을 확인하려면(AWS CLI)

다음 get-metric-statistics 명령을 사용하여, 지정된 기간 및 시간 간격을 사용하는 지정된 인스턴스의 CPUUtilization 측정치를 확인합니다.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --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": 99.670000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-19T07:18:00Z",
            "Maximum": 0.34000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-19T12:18:00Z",
            "Maximum": 0.34000000000000002,
            "Unit": "Percent"
        },
        ...
    ],
    "Label": "CPUUtilization"
}
```

여러 인스턴스의 통계 집계하기

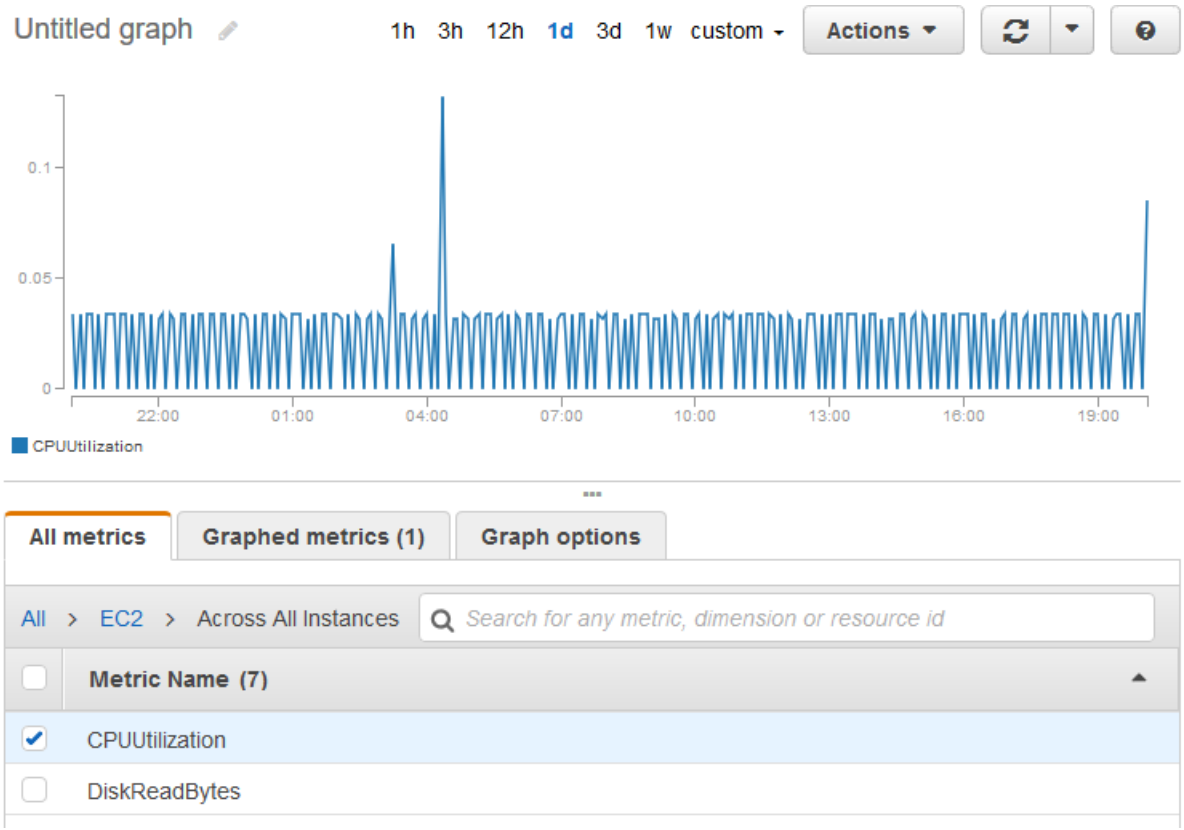
세부 모니터링이 활성화된 인스턴스에 대해서만 통계를 집계할 수 있습니다. 기본 모니터링을 사용하는 인스턴스는 집계에 포함되지 않습니다. 또한 Amazon CloudWatch에서는 리전 간 데이터는 집계하지 않습니다. 따라서 지표는 리전 간에 완전히 개별적입니다. 인스턴스 간에 집계된 통계를 얻으려면 1분 기간의 데이터를 제공하는 세부 모니터링(추가 비용 발생)을 활성화해야 합니다.

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(콘솔)

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표를 선택합니다.
3. EC2 네임스페이스를 선택한 후 전체 인스턴스를 선택합니다.
4. CPUUtilization을 포함하는 행을 선택합니다. 그러면 모든 EC2 인스턴스에 대한 지표 그래프가 표시됩니다. 그래프 이름을 지정하려면 연필 아이콘을 선택합니다. 시간 범위를 변경하려면 제공되는 값 중 하나를 선택하거나 사용자 지정을 선택합니다.



5. 측정치에 대한 통계 또는 기간을 변경하려면 그래프로 표시된 지표 탭을 선택합니다. 열 머리글이나 개별 값을 선택한 후 다른 값을 선택합니다.

인스턴스 간 평균 CPU 사용률을 얻으려면(AWS CLI)

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": 0.16670833333333332,
            "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. 지표는 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.

Auto Scaling 그룹의 인스턴스에 대한 DiskWriteBytes를 보려면(콘솔)

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표를 선택합니다.
3. EC2 네임스페이스를 선택한 후 Auto Scaling 그룹별을 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 사용자 지정을 선택합니다.
5. 측정치에 대한 통계 또는 기간을 변경하려면 그래프로 표시된 지표 탭을 선택합니다. 열 머리글이나 개별 값을 선택한 후 다른 값을 선택합니다.

Auto Scaling 그룹의 인스턴스에 대한 DiskWriteBytes를 보려면(AWS CLI)

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": 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. 지표는 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. 616) 단원을 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.

AMI의 평균 CPU 사용률을 보려면(콘솔)

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표를 선택합니다.
3. EC2 네임스페이스를 선택한 후 이미지(AMI) ID별을 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 사용자 지정을 선택합니다.
5. 측정치에 대한 통계 또는 기간을 변경하려면 그래프로 표시된 지표 탭을 선택합니다. 열 머리글이나 개별 값을 선택한 후 다른 값을 선택합니다.

이미지 ID에 대한 평균 CPU 사용률을 얻으려면(AWS CLI)

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": 0.041000000000000009,
            "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 모니터링 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. 617) 단원을 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 그래프를 참조하십시오.

인스턴스에 대해 CloudWatch 경보 만들기

CloudWatch 지표를 모니터링하는 CloudWatch 경보를 생성할 수 있습니다. 지표가 지정된 임계값에 도달하면 CloudWatch에서 자동으로 알림을 보냅니다. Amazon EC2 콘솔이나 CloudWatch 콘솔에 제공된 고급 옵션을 사용하여 CloudWatch 경보를 만들 수 있습니다.

CloudWatch 콘솔을 이용하여 경보 생성하기

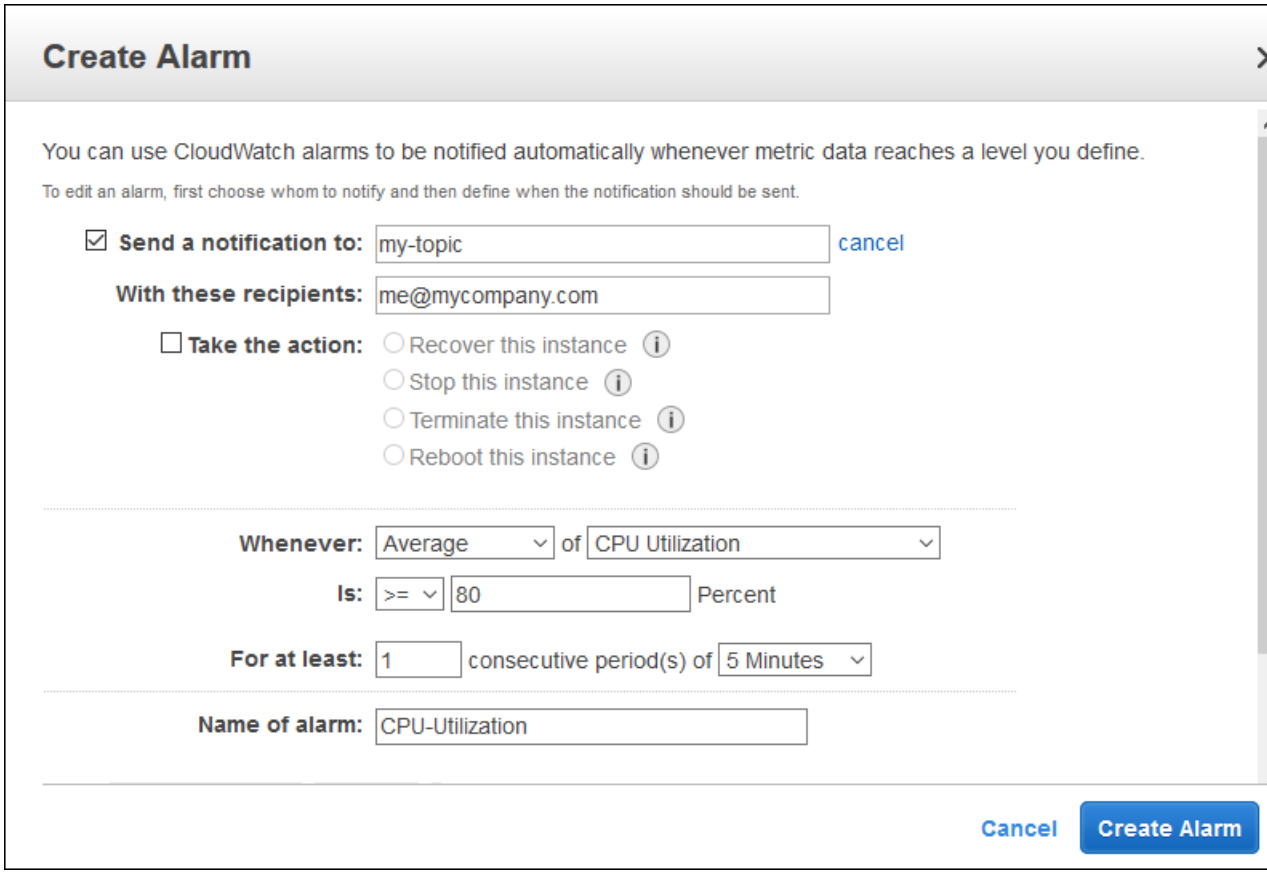
구체적인 예시는 Amazon CloudWatch 사용 설명서의 Amazon CloudWatch 경보 생성을 참조하십시오.

Amazon EC2 콘솔을 이용하여 경보 생성하기

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다.
4. 모니터링 탭에서 경보 생성을 선택합니다.
5. 경보 생성 대화 상자에서 다음 작업을 수행하십시오.
   a. 주제 생성을 선택합니다. Send a notification to(알림 보내기)에 SNS 주제의 이름을 입력합니다. 수신자들에게에 알림을 수신할 하나 이상의 이메일 주소를 입력합니다.

b. 정책에 대한 지표와 기준을 지정합니다. 예를 들어 다음 경우 항상(CPU 평균 사용률)를 기본 설정으로 유지할 수 있습니다. Is에서 `>=`을 선택하고 `80%`를 입력합니다. 최소 다음의 경우에서 연속 기간으로 `1(5 Minutes)`을 지정합니다.

c. 경보 생성을 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.

AWS는 `AWSServiceRoleForCloudWatchEvents` 서비스 연결 역할을 통해 사용자를 대신하여 경보 작업을 수행할 수 있습니다. AWS Management 콘솔에서 처음으로 경보 생성 시 IAM CLI, 혹은 IAM API, CloudWatch는 사용자를 대신해 서비스 연결 역할을 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 유휴 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 지표(`AWS/EC2` 네임스페이스)를 비롯한 인스턴스 측정치당 Amazon EC2 및 `InstanceId` 값이 실행 중인 유효한 Amazon EC2 인스턴스를 참조하는 경우 차원을 포함하는 모든 사용자 지정 지표에 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 생성을 참조하십시오.

권한

AWS Identity and Access Management(IAM) 사용자인 경우 경보를 만들거나 수정하려면 다음과 같은 권한이 있어야 합니다.

- `iam:CreateServiceLinkedRole`, `iam:GetPolicy`, `iam:GetPolicyVersion`, 및 `iam:GetRole` – Amazon EC2 작업을 수반하는 모든 경보
- `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 –
- 복구 작업을 수반하는 경보는 권한 제한이 없습니다.

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 정책을 참조하십시오.

내용

- Amazon CloudWatch 경보에 중지 작업 추가 (p. 638)
- Amazon CloudWatch 경보에 종료 작업 추가 (p. 639)
- Amazon CloudWatch 경보에 재부팅 작업 추가 (p. 640)
- Amazon CloudWatch 경보에 복구 작업 추가 (p. 640)
- Amazon CloudWatch 콘솔을 사용하여 경보 및 작업 기록 보기 (p. 642)
- Amazon CloudWatch 경보 작업 시나리오 (p. 642)

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.

유휴 인스턴스를 중지하는 경보를 생성하려면(Amazon EC2 콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다. 모니터링 탭에서 경보 생성을 선택합니다.
4. 경보 생성 대화 상자에서 다음 작업을 수행하십시오.

a. 경보가 트리거될 때 이메일을 받으려면 알림 보내기에서 기존 Amazon SNS 주제를 선택하거나, 주제 생성을 선택하여 새 주제를 만듭니다.

   새 주제를 만들려면 Send a notification to(알림 보내기)에 주제 이름을 입력한 다음 수신자들에게에 수신자의 이메일 주소를 쉼표로 구분하여 입력합니다. 경보를 만든 후에는 이 주제에 대한 알림을 받으려면 먼저 수락해야 하는 구독 확인 이메일이 전송됩니다.

b. 다음 작업을 수행을 선택하고 이 인스턴스를 중지합니다를 선택합니다.

c. 다음 경우 항상에서 사용하려는 통계를 선택한 다음 지표를 선택합니다. 이 예에서는 평균 및 CPU 사용률을 선택합니다.

d. 결과 값에 지표 임계값을 지정합니다. 이 예에서는 10%를 입력합니다.

e. 최소 다음의 경우에 경보의 평가 기간을 지정합니다. 이 예에서는 24 연속 기간(1시간)을 입력합니다.

f. 경보 이름을 변경하려면 경보 이름에 새 이름을 입력합니다. 경보 이름은 ASCII 문자만 포함해야 합니다.

   경보 이름을 입력하지 않으면 Amazon CloudWatch에서 이름이 자동으로 지정됩니다.

   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.

g. 경보 생성을 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 활성화를 참조하십시오.

유휴 인스턴스를 종료하는 경보를 생성하려면(Amazon EC2 콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다. 모니터링 탭에서 경보 생성을 선택합니다.
4. 경보 생성 대화 상자에서 다음 작업을 수행하십시오.

   a. 경보가 트리거될 때 이메일을 받으려면 알림 보내기에서 기존 Amazon SNS 주제를 선택하거나, 주제 생성을 선택하여 새 주제를 만듭니다.

      새 주제를 만들려면 Send a notification to(알림 보내기)에 주제 이름을 입력한 다음 수신자들에게에 수신자의 이메일 주소를 쉼표로 구분하여 입력합니다. 경보를 만든 후에는 이 주제에 대한 알림을 받으려면 먼저 수락해야 하는 구독 확인 이메일이 전송됩니다.

   b. 다음 작업을 수행을 선택하고 이 인스턴스를 종료합니다를 선택합니다.

   c. 다음 경우 항상에서 사용하려는 통계를 선택한 다음 지표를 선택합니다. 이 예에서는 평균 및 CPU 사용률을 선택합니다.

   d. 결과 값에 지표 임계값을 지정합니다. 이 예에서는 10%를 입력합니다.

   e. 최소 다음의 경우에 경보의 평가 기간을 지정합니다. 이 예에서는 24 연속 기간(1시간)을 입력합니다.

   f. 경보 이름을 변경하려면 경보 이름에 새 이름을 입력합니다. 경보 이름은 ASCII 문자만 포함해야 합니다.

경보 이름을 입력하지 않으면 Amazon CloudWatch에서 이름이 자동으로 지정됩니다.

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.

g. 경보 생성을 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 인스턴스 청구 (최소 1분 요금 포함)이 새로 시작되지 않습니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서에서 인스턴스 재부팅을 참조하십시오.

Important

재부팅과 복원 작업 간에 경합 상태가 발생하지 않도록 하려면 재부팅 경보와 복원 경보에 동일한 평가 기간 값을 설정하지 마십시오. 재부팅 경보를 각각 1분의 평가 기간 3회로 설정하는 것이 좋습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 경보 평가를 참조하십시오.

인스턴스를 재부팅하는 경보를 생성하려면(Amazon EC2 콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다. 모니터링 탭에서 경보 생성을 선택합니다.
4. 경보 생성 대화 상자에서 다음 작업을 수행하십시오.

   a. 경보가 트리거될 때 이메일을 받으려면 알림 보내기에서 기존 Amazon SNS 주제를 선택하거나, 주제 생성을 선택하여 새 주제를 만듭니다.

      새 주제를 만들려면 Send a notification to(알림 보내기)에 주제 이름을 입력한 다음 수신자들에게에 수신자의 이메일 주소를 쉼표로 구분하여 입력합니다. 경보를 만든 후에는 이 주제에 대한 알림을 받으려면 먼저 수락해야 하는 구독 확인 이메일이 전송됩니다.

   b. 다음 작업을 수행을 선택하고 이 인스턴스를 재부팅합니다를 선택합니다.

   c. 다음 경우 항상에서 상태 확인 실패(인스턴스)를 선택합니다.

   d. 최소 다음의 경우에 경보의 평가 기간을 지정합니다. 이 예에서는 3 연속 기간(1분)을 입력합니다.

   e. 경보 이름을 변경하려면 경보 이름에 새 이름을 입력합니다. 경보 이름은 ASCII 문자만 포함해야 합니다.

      경보 이름을 입력하지 않으면 Amazon CloudWatch에서 이름이 자동으로 지정됩니다.

   f. 경보 생성을 선택합니다.

Amazon CloudWatch 경보에 복구 작업 추가

Amazon EC2 인스턴스를 모니터링하는 Amazon CloudWatch 경보를 만들 수 있습니다. 기본 하드웨어 장애나 복구에 AWS 개입이 필요한 문제로 인해 인스턴스가 손상된 경우 해당 인스턴스를 자동으로 복구할 수 있

습니다. 종료한 인스턴스는 복구할 수 없습니다. 복구된 인스턴스는 인스턴스 ID, 프라이빗 IP 주소, 탄력적 IP 주소 및 모든 인스턴스 메타데이터를 포함하여 원본 인스턴스와 동일합니다.

CloudWatch은 복구 작업을 지원하지 않는 인스턴스에 대한 경보에 복구 작업을 추가할 수 없게 합니다.

`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 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.

복구 작업은 `StatusCheckFailed_Instance`가 아닌 `StatusCheckFailed_System`을 통해서만 사용할 수 있습니다.

다음과 같은 문제가 있을 경우 시스템 상태 확인이 실패할 수 있습니다.

- 네트워크 연결 끊김
- 시스템 전원 중단
- 물리적 호스트의 소프트웨어 문제
- 네트워크 연결성에 영향을 주는 물리적 호스트의 하드웨어 문제

복구 작업은 다음 특성을 지닌 인스턴스에만 지원됩니다.

- 게이트웨이 유형에 대한 권장 인스턴스 유형 A1, C3, C4, C5, C5n, , M3, M4, M5, M5a, M5n, P3, R3, R4, R5, R5a, R5n, T2, T3, T3a, X1 또는 X1e 중 하나를 사용합니다.
- `default` 또는 `dedicated` 인스턴스 테넌시 사용
- EBS 볼륨(인스턴스 스토어 볼륨을 구성하지 않음)만 사용합니다. 자세한 내용은 '이 인스턴스를 복구합니다'가 비활성화됨 단원을 참조하십시오.

인스턴스에 퍼블릭 IP 주소가 있는 경우 복구 후에도 해당 퍼블릭 IP 주소를 유지합니다.

> **Important**
>
> 재부팅과 복원 작업 간에 경합 상태가 발생하지 않도록 하려면 재부팅 경보와 복원 경보에 동일한 평가 기간 값을 설정하지 마십시오. 복구 경보는 각각 1분의 평가 기간 2회로 설정하는 것이 좋습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 경보 평가를 참조하십시오.

인스턴스를 복구하는 경보를 생성하려면(Amazon EC2 콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다. 모니터링 탭에서 경보 생성을 선택합니다.
4. 경보 생성 대화 상자에서 다음 작업을 수행하십시오.

   a. 경보가 트리거될 때 이메일을 받으려면 알림 보내기에서 기존 Amazon SNS 주제를 선택하거나, 주제 생성을 선택하여 새 주제를 만듭니다.

      새 주제를 만들려면 Send a notification to(알림 보내기)에 주제 이름을 입력한 다음 수신자들에게에 수신자의 이메일 주소를 쉼표로 구분하여 입력합니다. 경보를 만든 후에는 이 주제에 대한 이메일을 받기 전에 수락해야 하는 구독 확인 이메일이 전송됩니다.

      > **Note**
      >
      > - 경보가 트리거될 때 이메일 알림을 수신하려면 사용자는 지정된 SNS 주제를 구독해야 합니다.

- AWS 계정의 루트 사용자는 SNS 주제가 지정되어 있지 않더라도 자동 인스턴스 복구 작업이 발생하면 항상 이메일 알림을 받습니다.
- AWS 계정의 루트 사용자는 지정된 SNS 주제를 구독하지 않더라도 자동 인스턴스 복구 작업이 발생하면 항상 이메일 알림을 받습니다.

b. 다음 작업을 수행을 선택하고 이 인스턴스를 복구합니다를 선택합니다.

c. 다음 경우 항상에서 상태 확인 실패(시스템)을 선택합니다.

d. 최소 다음의 경우에 경보의 평가 기간을 지정합니다. 이 예에서는 2 연속 기간(1분)을 입력합니다.

e. 경보 이름을 변경하려면 경보 이름에 새 이름을 입력합니다. 경보 이름은 ASCII 문자만 포함해야 합니다.

   경보 이름을 입력하지 않으면 Amazon CloudWatch에서 이름이 자동으로 지정됩니다.

f. 경보 생성을 선택합니다.

Amazon CloudWatch 콘솔을 사용하여 경보 및 작업 기록 보기

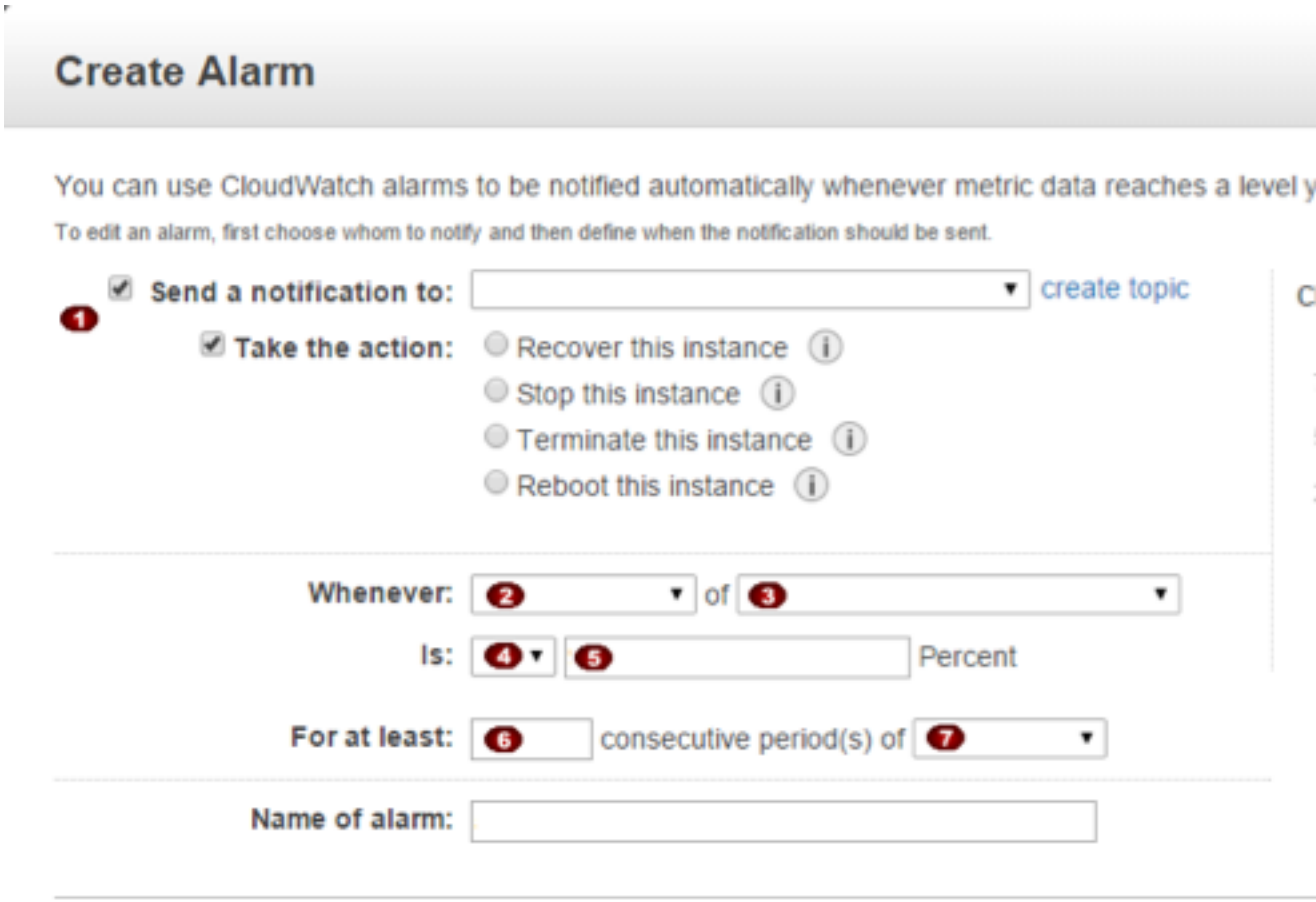
Amazon CloudWatch 콘솔에서 경보 및 작업 기록을 볼 수 있습니다. Amazon CloudWatch에서는 지난 2주간의 경보 및 작업 기록을 보관합니다.

트리거된 경보 및 작업 기록을 보려면(CloudWatch 콘솔)

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Alarms를 선택합니다.
3. 경보를 선택합니다.
4. 세부 정보 탭에 최근 상태 변화가 시간 및 지표 값과 함께 표시됩니다.
5. 최신 기록 항목을 보려면 기록 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 만듭니다.

설정	값
1	Stop
2	Maximum
3	CPUUtilization
4	<=
5	10%
6	60분
7	1

시나리오 2: 유휴 인스턴스 중지

인스턴스가 24시간 동안 유휴 상태인 경우 인스턴스를 중지하고 이메일을 보내는 경보를 만듭니다.

설정	값
1	Stop and email
2	Average
3	CPUUtilization
4	<=
5	5%
6	60분
7	24

시나리오 3: 트래픽이 비정상적으로 높은 웹 서버에 대해 이메일 보내기

인스턴스가 일일 아웃바운드 네트워크 트래픽인 10GB를 초과하는 경우 이메일을 보내는 경보를 만듭니다.

설정	값
1	이메일
2	Sum
3	NetworkOut
4	>
5	10GB
6	1 day
7	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
1	Stop and send SMS
2	Sum
3	NetworkOut
4	>
5	1GB

설정	값
6	1시간
7	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 Amazon EC2 Linux 인스턴스에 대한 메모리 및 디스크 지표 모니터링을 참조하십시오.

설정	값
1	Stop
2	Maximum
3	MemoryUtilization
4	>=
5	90%
6	1분
7	1

시나리오 6: 손상된 인스턴스 중지

5분 간격으로 수행된 연속 3회의 상태 확인에 실패한 인스턴스를 중지하는 경보를 만듭니다.

설정	값
1	Stop
2	Average
3	StatusCheckFailed_System
4	>=
5	1
6	15 minutes
7	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
1	Terminate
2	Maximum
3	NetworkOut
4	<=
5	100,000 bytes
6	5 minutes
7	1

CloudWatch 이벤트를 사용한 Amazon EC2 자동화

Amazon CloudWatch Events를 사용하여 AWS 서비스를 자동화하고 애플리케이션 가용성 문제나 리소스 변경 같은 시스템 이벤트에 자동으로 응답합니다. 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. 자동으로 트리거할 수 있는 작업은 다음과 같습니다.

- AWS Lambda 함수 호출
- Amazon EC2 Run Command 호출
- Amazon Kinesis Data Streams로 이벤트 릴레이
- AWS Step Functions 상태 머신 활성화
- Amazon SNS 주제 또는 Amazon SQS 대기열 알림

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 설명서를 참조하십시오.

Amazon EC2 Linux 인스턴스의 메모리 및 디스크 메트릭 모니터링

Amazon CloudWatch를 사용하여 운영 체제에서 EC2 인스턴스에 대한 지표와 로그를 수집할 수 있습니다.

CloudWatch 에이전트

CloudWatch 에이전트를 사용하여 Amazon EC2 인스턴스 및 온프레미스 서버에서 시스템 지표 및 로그 파일을 수집할 수 있습니다. 이 에이전트는 Windows Server와 Linux를 모두 지원하며 CPU당 코어와 같은 하

위 리소스 지표를 포함하여 수집할 지표를 선택할 수 있습니다. 모니터링 스크립트를 사용하는 대신 에이전트를 사용하여 측정치와 로그를 수집하는 것이 좋습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 에이전트를 사용하여 Amazon EC2 인스턴스 및 온프레미스 서버에서 지표 수집을 참조하십시오.

CloudWatch 모니터링 스크립트

> **Important**
>
> CloudWatch 에이전트를 사용하여 지표와 로그를 수집하는 것이 좋습니다. 아직도 기존 모니터링 스크립트를 사용하여 Linux 인스턴스에서 정보를 수집하는 고객에게는 모니터링 스크립트에 대한 정보가 제공됩니다.

모니터링 스크립트는 Amazon CloudWatch의 사용자 지정 지표를 생성하고 사용하는 방법을 보여줍니다. 이 예제는 Linux 인스턴스에 대한 메모리, 스왑 및 디스크 공간 사용률 측정치를 보고하는 완벽하게 작동하는 예로 구성된 Perl 스크립트입니다.

사용자 지정 지표에 대한 표준 Amazon CloudWatch 사용 요금이 이러한 스크립트 사용에 적용됩니다. 자세한 내용은 Amazon CloudWatch 요금 페이지를 참조하십시오.

내용

- 지원되는 시스템 (p. 647)
- 필요한 권한 (p. 647)
- 필수 패키지 설치 (p. 648)
- 모니터링 스크립트 설치 (p. 649)
- mon-put-instance-data.pl (p. 650)
- mon-get-instance-stats.pl (p. 652)
- 콘솔에서 사용자 지정 측정치 보기 (p. 653)
- 문제 해결 (p. 654)

지원되는 시스템

모니터링 스크립트는 다음 시스템을 사용하는 인스턴스에서 테스트되었습니다.

- Amazon Linux 2
- Amazon Linux AMI 2014.09.2 이상
- Red Hat Enterprise Linux 6.9 및 7.4
- SUSE Linux Enterprise Server 12
- Ubuntu Server 14.04 및 16.04

필요한 권한

스크립트가 IAM 역할을 인스턴스와 연결하여 다음 작업을 호출할 수 있는 권한을 가지고 있는지 확인합니다.

- cloudwatch:PutMetricData
- cloudwatch:GetMetricStatistics
- cloudwatch:ListMetrics
- ec2:DescribeTags

자세한 내용은 IAM 역할 작업 (p. 853) 단원을 참조하십시오.

필수 패키지 설치

일부 Linux 버전의 경우 모니터링 스크립트를 사용하려면 먼저 추가 Perl 모듈을 설치해야 합니다.

Amazon Linux 2 및 Amazon Linux AMI에 필요한 패키지를 설치하려면

1. 인스턴스에 로그온합니다. 자세한 내용은 Linux 인스턴스에 연결 (p. 485) 단원을 참조하십시오.
2. 명령 프롬프트에서 다음과 같이 패키지를 설치합니다.

```
sudo yum install -y perl-Switch perl-DateTime perl-Sys-Syslog perl-LWP-Protocol-https
 perl-Digest-SHA.x86_64
```

Ubuntu에 필요한 패키지를 설치하려면,

1. 인스턴스에 로그온합니다. 자세한 내용은 Linux 인스턴스에 연결 (p. 485) 단원을 참조하십시오.
2. 명령 프롬프트에서 다음과 같이 패키지를 설치합니다.

```
sudo apt-get update
sudo apt-get install unzip
sudo apt-get install libwww-perl libdatetime-perl
```

Red Hat Enterprise Linux 7에 필수 패키지를 설치하려면

1. 인스턴스에 로그온합니다. 자세한 내용은 Linux 인스턴스에 연결 (p. 485) 단원을 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 --enablerepo="rhui-REGION-rhel-server-optional" -y
sudo yum install zip unzip
```

Red Hat Enterprise Linux 6.9에 필요한 패키지를 설치하려면

1. 인스턴스에 로그온합니다. 자세한 내용은 Linux 인스턴스에 연결 (p. 485) 단원을 참조하십시오.
2. 명령 프롬프트에서 다음과 같이 패키지를 설치합니다.

```
sudo yum install perl-DateTime perl-CPAN perl-Net-SSLeay perl-IO-Socket-SSL perl-
Digest-SHA gcc -y
sudo yum install zip unzip
```

3. CPAN을 승격된 사용자로 실행합니다.

```
sudo cpan
```

   다음 프롬프트가 표시될 때까지 프롬프트를 통해 Enter 키를 누릅니다.

```
cpan[1]>
```

4. CPAN 프롬프트에서 아래의 각 명령을 실행합니다. 명령 하나를 실행하면 해당 사항이 설치되며, CPAN 프롬프트로 돌아가면 다음 명령을 실행합니다. 프로세스를 계속 진행하라는 메시지가 나타나면 이전과 같이 Enter 키를 누릅니다.

```
cpan[1]> install YAML
cpan[2]> install LWP::Protocol::https
cpan[3]> install Sys::Syslog
cpan[4]> install Switch
```

SUSE에 필요한 패키지를 설치하려면,

1. 인스턴스에 로그온합니다. 자세한 내용은 Linux 인스턴스에 연결 (p. 485) 단원을 참조하십시오.
2. SUSE Linux Enterprise Server 12를 실행하는 서버에서는 `perl-Switch` 패키지를 다운로드해야 할 수 있습니다. 다음 명령을 사용하여 패키지를 다운로드하고 설치합니다.

```
wget http://download.opensuse.org/repositories/devel:/languages:/perl/SLE_12_SP3/
noarch/perl-Switch-2.17-32.1.noarch.rpm
sudo rpm -i perl-Switch-2.17-32.1.noarch.rpm
```

3. 다음과 같이 필수 패키지를 설치합니다.

```
sudo zypper install perl-Switch perl-DateTime
sudo zypper install -y "perl(LWP::Protocol::https)"
```

모니터링 스크립트 설치

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 https://aws-cloudwatch.s3.amazonaws.com/downloads/
CloudWatchMonitoringScripts-1.2.2.zip -O
```

2. 다음 명령을 실행하여 다운로드한 모니터링 스크립트를 설치합니다.

```
unzip CloudWatchMonitoringScripts-1.2.2.zip && \
rm CloudWatchMonitoringScripts-1.2.2.zip && \
cd aws-scripts-mon
```

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 지표(메모리, 스왑, 디스크 사용률)에 대한 시스템 지표를 수집하여 Amazon CloudWatch에 전송합니다.
- mon-get-instance-stats.pl – 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 고지입니다.

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 지표는 애플리케이션 및 운영 체제에서 할당한 메모리만 사용량으로 계산하며, `--mem-used-incl-cache-buff` 옵션을 지정하는 경우 캐시 및 버퍼 메모리도 사용량에 포함시킵니다.
`--mem-used`	MemoryUsed 측정치를 수집하여 전송합니다. 이때, MB 단위로 보고합니다. 이 지표는 애플리케이션 및 운영 체제에서 할당한 메모리만 사용량으로 계산하며, `--mem-used-incl-cache-buff` 옵션을 지정하는 경우 캐시 및 버퍼 메모리도 사용량에 포함시킵니다.
`--mem-used-incl-cache-buff`	이 옵션을 추가하면 `--mem-util`, `--mem-used` 및 `--mem-avail`의 지표를 보고할 때 현재 캐시 및 버퍼용으로 사용 중인 메모리가 "사용됨"으로 표시됩니다.
`--mem-avail`	MemoryAvailable 측정치를 수집하여 전송합니다. 이때, MB 단위로 보고합니다. 이 지표는 애플리케이션 및 운영 체제에서 할당한 메모리만 사용량으로 계산하며, `--mem-used-incl-cache-buff` 옵션을 지정하는 경우 캐시 및 버퍼 메모리도 사용량에 포함시킵니다.
`--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=PATH`를 지정합니다. `/` 및 `/home`에 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.

이름	설명
`--disk-space-avail`	선택한 디스크의 DiskSpaceAvailable 측정치를 수집하여 전송합니다. 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.
`--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 DescribeTags를 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.

이름	설명
`--verbose`	스크립트가 수행한 작업에 대한 자세한 정보를 표시합니다.
`--help`	사용 정보를 표시합니다.
`--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.

다음 예제에서는 CloudWatch에 데이터를 게시하지 않고 간단한 테스트 실행을 수행합니다.

```
./mon-put-instance-data.pl --mem-util --verify --verbose
```

다음 예제에서는 사용 가능한 모든 메모리 지표를 수집하고 CloudWatch에 전송하여 사용되는 캐시 및 버퍼 메모리를 계산합니다.

```
./mon-put-instance-data.pl --mem-used-incl-cache-buff --mem-util --mem-used --mem-avail
```

다음 예제에서는 개별 인스턴스 지표를 보고하지 않고 Auto Scaling 그룹에 대한 집계 지표를 수집하여 Amazon CloudWatch에 전송합니다.

```
./mon-put-instance-data.pl --mem-util --mem-used --mem-avail --auto-scaling=only
```

다음 예제에서는 개별 인스턴스 지표를 보고하지 않고 인스턴스 유형, AMI ID 및 리전에 대한 집계 메트릭을 수집하여 Amazon CloudWatch에 전송합니다.

```
./mon-put-instance-data.pl --mem-util --mem-used --mem-avail --aggregated=only
```

CloudWatch에 보고된 지표에 대한 Cron 일정을 설정하려면 crontab -e 명령을 사용하여 crontab 편집을 시작합니다. 다음 명령을 추가하여 5분마다 메모리 및 디스크 공간 사용률을 CloudWatch에 보고합니다.

```
*/5 * * * * ~/aws-scripts-mon/mon-put-instance-data.pl --mem-used-incl-cache-buff --mem-
util --disk-space-util --disk-path=/ --from-cron
```

스크립트에 오류가 발생하면 시스템 로그에 오류 메시지가 기록됩니다.

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 `N`으로 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`	스크립트의 테스트 실행을 실시합니다. 이 옵션 역시 자격 증명이 제공되었는지 확인합니다.
`--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. 지표 보기(View Metrics)를 선택합니다.
4. 스크립트에 게시된 사용자 정의 측정치가 접두사 `System/Linux`와 함께 보기에 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
```

AWS CloudTrail을 사용하여 Amazon EC2 및 Amazon EBS API 호출 로깅

Amazon EC2 및 Amazon EBS는 Amazon EC2 및 Amazon EBS의 사용자, 역할 또는 AWS 서비스가 수행한 작업에 대한 레코드를 제공하는 서비스인 AWS CloudTrail과 통합됩니다. CloudTrail은 콘솔의 호출 및 API 코드 호출 등 Amazon EC2 및 Amazon EBS에 대한 모든 API 호출을 이벤트로 캡처합니다. 추적을 생성하면, Amazon EC2 및 Amazon EBS 이벤트를 비롯하여 CloudTrail 이벤트를 Amazon S3 버킷으로 지속적으로 배포할 수 있습니다. 추적을 구성하지 않은 경우 이벤트 기록에서 CloudTrail 콘솔의 최신 이벤트를 볼 수도 있습니다. CloudTrail에서 수집하는 정보를 사용하여 Amazon EC2 및 Amazon EBS에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

CloudTrail에 대한 자세한 내용은 AWS CloudTrail User Guide을 참조하십시오.

CloudTrail의 Amazon EC2 및 Amazon EBS 정보

CloudTrail은 계정 생성 시 AWS 계정에서 활성화됩니다. Amazon EC2 및 Amazon EBS에서 활동이 수행되면 해당 활동은 이벤트 기록에서 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. AWS 계정에서 최신 이벤트를 확인, 검색 및 다운로드할 수 있습니다. 자세한 내용은 CloudTrail 이벤트 기록에서 이벤트 보기를 참조하십시오.

Amazon EC2 및 Amazon EBS 이벤트를 비롯하여 AWS 계정의 이벤트 기록을 보유하려면 추적을 생성하십시오. 추적은 CloudTrail이 Amazon S3 버킷으로 로그 파일을 전송할 수 있도록 합니다. 콘솔에서 추적을 생성하면 기본적으로 모든 리전에 추적이 적용됩니다. 추적은 AWS 파티션에 있는 모든 리전의 이벤트를 로깅하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또는 CloudTrail 로그에서 수집된 이벤트 데이터를 추가 분석 및 처리하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 다음을 참조하십시오.

- 추적 생성 개요
- CloudTrail 지원 서비스 및 통합
- CloudTrail에 대한 Amazon SNS 알림 구성
- 여러 리전에서 CloudTrail 로그 파일 받기 및 여러 계정에서 CloudTrail 로그 파일 받기

모든 Amazon EC2 및 Amazon EBS 작업이 CloudTrail에서 로깅되고 Amazon EC2 API Reference에 문서화됩니다. 예를 들어 RunInstances, DescribeInstances 또는 CreateImage 작업을 호출하면 CloudTrail 로그 파일에 항목이 생성됩니다.

모든 이벤트 및 로그 항목에는 요청을 생성한 사용자에 대한 정보가 들어 있습니다. 자격 증명 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청을 루트로 했는지 아니면 IAM 사용자 자격 증명으로 했는지 여부
- 역할 또는 연합된 사용자에 대한 임시 보안 자격 증명을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청했는지 여부.

자세한 내용은 CloudTrail userIdentity 요소를 참조하십시오.

Amazon EC2 및 Amazon EBS 로그 파일 항목 이해

추적은 지정한 Amazon S3 버킷에 이벤트를 로그 파일로 제공할 수 있도록 해 주는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함됩니다. 이벤트는 어떤 소스로부터의 단일 요청을 나타내며 요청된 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보가 들어 있습니다. CloudTrail 로그 파일은 퍼블릭 API 호출의 순서가 지정된 스택 추적이 아니기 때문에 특정 순서로 표시되지 않습니다.

다음 로그 파일 레코드는 사용자가 인스턴스를 종료했음을 보여 줍니다.


```
{
   "Records":[
      {
         "eventVersion":"1.03",
         "userIdentity":{
            "type":"Root",
            "principalId":"123456789012",
            "arn":"arn:aws:iam::123456789012:root",
            "accountId":"123456789012",
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "userName":"user"
         },
         "eventTime":"2016-05-20T08:27:45Z",
         "eventSource":"ec2.amazonaws.com",
         "eventName":"TerminateInstances",
         "awsRegion":"us-west-2",
         "sourceIPAddress":"198.51.100.1",
         "userAgent":"aws-cli/1.10.10 Python/2.7.9 Windows/7botocore/1.4.1",
         "requestParameters":{
            "instancesSet":{
               "items":[{
                  "instanceId":"i-1a2b3c4d"
               }]
            }
         },
         "responseElements":{
            "instancesSet":{
               "items":[{
                  "instanceId":"i-1a2b3c4d",
                  "currentState":{
                     "code":32,
                     "name":"shutting-down"
                  },
                  "previousState":{
                     "code":16,
                     "name":"running"
                  }
               }]
            }
         },
         "requestID":"be112233-1ba5-4ae0-8e2b-1c302EXAMPLE",
         "eventID":"6e12345-2a4e-417c-aa78-7594fEXAMPLE",
         "eventType":"AwsApiCall",
```

```
        "recipientAccountId":"123456789012"
    }
  ]
}
```

AWS CloudTrail을 사용하여 EC2 Instance Connect를 통해 연결하는 사용자 감사

AWS CloudTrail을 사용하여 EC2 Instance Connect를 통해 인스턴스에 연결하는 사용자를 감사합니다.

AWS CloudTrail 콘솔을 사용하여 EC2 Instance Connect를 통해 SSH 활동을 감사하려면

1. https://console.aws.amazon.com/cloudtrail/에서 AWS CloudTrail 콘솔을 엽니다.
2. 올바른 리전에 있는지 확인합니다.
3. 탐색 창에서 Event history(이벤트 내역)를 선택합니다.
4. Filter(필터)에서 Event source(이벤트 소스), ec2-instance-connect.amazonaws.com을 선택합니다.
5. (선택 사항) Time range(시간 범위)에서 시간 범위를 선택합니다.
6. Refresh events(새로 고침 이벤트) 아이콘을 선택합니다.
7. 이 페이지에는 `SendSSHPublicKey` API 호출에 해당하는 이벤트가 표시됩니다. 화살표를 사용하여 이벤트를 확장하면 SSH 연결을 만드는 데 사용된 사용자 이름, AWS 액세스 키 및 소스 IP 주소와 같은 추가 세부 정보를 볼 수 있습니다.
8. 전체 이벤트 정보를 JSON 형식으로 표시하려면 View event(이벤트 보기)를 선택합니다. requestParameters 필드에는 SSH 연결을 만드는 데 사용된 대상 인스턴스 ID, OS 사용자 이름 및 퍼블릭 키가 포함되어 있습니다.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "ABCDEFGONGNOMOOCB6XYTQEXAMPLE",
        "arn": "arn:aws:iam::1234567890120:user/IAM-friendly-name",
        "accountId": "123456789012",
        "accessKeyId": "ABCDEFGUKZHNAW4OSN2AEXAMPLE",
        "userName": "IAM-friendly-name",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-09-21T21:37:58Z"}
        }
    },
    "eventTime": "2018-09-21T21:38:00Z",
    "eventSource": "ec2-instance-connect.amazonaws.com",
    "eventName": "SendSSHPublicKey ",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "123.456.789.012",
    "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
    "requestParameters": {
        "instanceId": "i-0123456789EXAMPLE",
        "osUser": "ec2-user",
        "SSHKey": {
            "publicKey": "ssh-rsa ABCDEFGHIJKLMNO01234567890EXAMPLE"
        }
    "responseElements": null,
    "requestID": "1a2s3d4f-bde6-11e8-a892-f7ec64543add",
    "eventID": "1a2w3d4r5-a88f-4e28-b3bf-30161f75be34",
    "eventType": "AwsApiCall",
    "recipientAccountId": "0987654321"
```

```
}
```

S3 버킷에서 CloudTrail 이벤트를 수집하도록 AWS 계정을 구성한 경우 프로그래밍 방식으로 정보를 다운로드하고 감사할 수 있습니다. 자세한 내용은 AWS CloudTrail User Guide의 CloudTrail 로그 파일 가져오기 및 보기를 참조하십시오.

Amazon EC2의 네트워킹

Amazon EC2는 다음과 같은 네트워킹 기능을 제공합니다.

기능

- Amazon EC2인스턴스 IP 어드레싱 (p. 658)
- 고유 IP 주소 가져오기(BYOIP) (p. 672)
- 탄력적 IP 주소 (p. 676)
- 탄력적 네트워크 인터페이스 (p. 683)
- Linux에서 향상된 네트워킹 (p. 706)
- Elastic Fabric Adapter(EFA) (p. 731)
- 배치 그룹 (p. 757)
- EC2 인스턴스에 대한 네트워크 MTU(최대 전송 단위) (p. 766)
- Virtual Private Cloud (p. 769)
- EC2-Classic (p. 770)

Amazon EC2인스턴스 IP 어드레싱

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)를 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 단원을 참조하십시오.

내용

- 프라이빗 IPv4 주소 및 내부 DNS 호스트 이름 (p. 658)
- 퍼블릭 IPv4 주소 및 외부 DNS 호스트 이름 (p. 659)
- 탄력적 IP 주소(IPv4) (p. 660)
- Amazon DNS 서버 (p. 660)
- IPv6 주소 (p. 660)
- 인스턴스에 대한 IP 주소 작업 (p. 661)
- 다중 IP 주소 (p. 665)

프라이빗 IPv4 주소 및 내부 DNS 호스트 이름

프라이빗 IPv4 주소는 인터넷을 통해 연결할 수 없는 IP 주소입니다. 프라이빗 IPv4 주소는 동일 VPC에서 인스턴스 간의 통신을 위해 사용될 수 있습니다. 프라이빗 IPv4 주소의 표준 및 사양에 대한 자세한 내용은 RFC 1918을 참조하십시오. DHCP를 사용하여 개인 IPv4 주소를 인스턴스에 할당합니다.

> **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 인스턴스에 기본 프라이빗 IPv4 주소가 할당됩니다. 또한 각 인스턴스에는 `ip-10-251-50-12.ec2.internal`과 같이 기본 프라이빗 IP 주소를 확인하는 내부 DNS 호스트 이름이

할당됩니다. 동일한 VPC에 있는 인스턴스 간의 통신에 내부 DNS 호스트 이름을 사용할 수 있지만, VPC 외부에 있는 내부 DNS 호스트 이름을 확인할 수 없습니다.

인스턴스에는 서브넷 IPv4 주소 범위 내의 기본 프라이빗 IP 주소가 할당됩니다. 자세한 내용은 Amazon VPC 사용 설명서의 VPC 및 서브넷 크기 조정을 참조하십시오. 인스턴스 시작 시 사용자가 기본 프라이빗 IP 주소를 지정하지 않으면 사용자 서브넷 IPv4 범위 내의 IP 주소가 할당됩니다. 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. 665) 단원을 참조하십시오.

프라이빗 IPv4 주소는 기본 주소인지 보조 주소인지 관계없이 인스턴스가 중지되었다가 다시 시작될 때 네트워크 인터페이스와 연결이 유지되고 인스턴스가 종료되면 해제됩니다.

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`. AWS에서는 외부 DNS 호스트 이름을 VPC 외부 인스턴스의 퍼블릭 IP 주소 및, VPC 내부 인스턴스의 프라이빗 IPv4 주소로 확인합니다. 퍼블릭 IP 주소는 네트워크 주소 변환(NAT)를 통해 기본 프라이빗 IP 주소로 매핑됩니다. 자세한 내용은 RFC 1631: IP 네트워크 주소 변환기(NAT)를 참조하십시오.

기본 VPC에서 인스턴스를 시작할 때 기본적으로 퍼블릭 IP 주소가 할당됩니다. 기본 VPC가 아닌 VPC로 인스턴스를 시작하는 경우 서브넷은 이 서브넷으로 시작되는 인스턴스가 퍼블릭 IPv4 주소 풀로부터 퍼블릭 IP 주소를 부여받는지 여부를 결정하는 속성을 갖습니다. 기본적으로 기본 서브넷이 아닌 서브넷에서 시작된 인스턴스에 퍼블릭 IP 주소가 할당되지 않습니다.

다음과 같이 인스턴스가 퍼블릭 IP 주소를 수신할지 여부를 제어할 수 있습니다.

- 서브넷의 퍼블릭 IP 주소 지정 속성 수정. 자세한 내용은 Amazon VPC 사용 설명서의 서브넷의 퍼블릭 IPv4 주소 지정 속성 수정 단원을 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. 663) 단원을 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. 어떤 경우에는 Amazon에서 귀하의 인스턴스로부터 퍼블릭 IP 주소를 해제하거나 새 인스턴스에 할당합니다:

- 인스턴스가 중지 또는 종료되면 인스턴스의 퍼블릭 IP 주소는 릴리스됩니다. 중지된 인스턴스가 다시 시작되면 새 퍼블릭 IP 주소가 할당됩니다.
- 탄력적 IP 주소를 인스턴스와 연결하는 경우 인스턴스의 퍼블릭 IP 주소가 릴리스됩니다. 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.
- 인스턴스의 퍼블릭 IP 주소가 릴리스된 가운데 탄력적 IP 주소와 연결된 보조 프라이빗 IP 주소를 보유한 경우 인스턴스는 새 퍼블릭 IP 주소를 수신하지 않습니다.

필요에 따라 인스턴스 간에 연결할 수 있는 영구 퍼블릭 IP 주소가 필요한 경우 탄력적 IP 주소를 대신하여 사용합니다.

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. 676) 단원을 참조하십시오.

탄력적 IP 주소가 인스턴스에 할당된 경우 IPv4 DNS 호스트 이름이 활성화되어 있으면 인스턴스에 DNS 호스트 이름이 할당됩니다. 자세한 내용은 Amazon VPC 사용 설명서의 VPC에서 DNS 사용을 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. 엘라스틱 IP 주소 및 사용 방법에 대한 자세한 내용은 탄력적 IP 주소 (p. 676) 단원을 참조하십시오.

IPv6에 대한 탄력적 IP 주소는 지원하지 않습니다.

Amazon DNS 서버

Amazon은 Amazon이 제공한 IPv4 DNS 호스트 이름을 IPv4 주소로 변환하는 DNS 서버를 제공합니다. Amazon DNS 서버는 사용자 VPC 네트워크 범위 +2의 범위에 위치합니다. 자세한 내용은 Amazon VPC 사용 설명서의 Amazon DNS 서버 단원을 참조하십시오.

IPv6 주소

IPv6 CIDR 블록과 VPC를 연결하고 IPv6 CIDR 블록과 서브넷을 연결할 수도 있습니다. VPC에 대한 IPv6 CIDR 블록은 Amazon의 IPv6 주소 풀에서 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 인스턴스는 IPv6 주소를 받습니다.

- 서브넷은 시작 중인 인스턴스에 IPv6 주소를 자동으로 할당하도록 구성됩니다. 자세한 정보는 서브넷의 IPv6 주소 지정 속성 수정 단원을 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. 인스턴스에 대해서는 IPv6 DNS 호스트 이름을 지원하지 않습니다.

인스턴스를 중지하고 시작할 때에는 IPv6 주소가 지속되다가 인스턴스를 종료하면 해제됩니다. IPv6 주소는 다른 네트워크 인터페이스에 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. 685) 단원을 참조하십시오.

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.

내용

- 퍼블릭, 프라이빗, 탄력적 IP 주소 결정 (p. 661)
- IPv6 주소 결정 (p. 662)
- 인스턴스 시작 시 퍼블릭 IPv4 주소 배정 (p. 663)
- 인스턴스에 IPv6 주소 할당 (p. 664)
- 인스턴스에 할당된 IPv6 주소 해제 (p. 665)

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. 571) 단원을 참조하십시오.

콘솔을 이용하여 인스턴스의 프라이빗 IPv4 주소를 결정하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다. 세부 정보 창의 프라이빗 IP 필드에서 프라이빗 IPv4 주소를 획득하고 프라이빗 DNS 필드에서 내부 DNS 호스트 이름을 획득합니다.
4. 인스턴스에 연결된 네트워크 인터페이스에 1개 이상의 보조 프라이빗 IPv4 주소가 할당된 경우 해당 IP 주소는 보조 프라이빗 IP 필드에서 획득할 수 있습니다.
5. 아니면, 탐색 창에서 네트워크 인터페이스를 선택한 후 인스턴스에 연결된 네트워크 인터페이스를 선택합니다.
6. 기본 프라이빗 IPv4 IP 필드에서 기본 프라이빗 IP 주소를 획득하고 프라이빗 DNS(IPv4) 필드에서 내부 DNS 호스트 이름을 획득합니다.
7. 네트워크 인터페이스에 보조 프라이빗 IP 주소가 할당된 경우 해당 IP 주소는 보조 프라이빗 IPv4 IP에서 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 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다. 세부 정보 창의 IPv4 퍼블릭 IP 필드에서 퍼블릭 IP 주소를 획득하고 퍼블릭 DNS(IPv4) 필드에서 외부 DNS 호스트 이름을 획득합니다.
4. 탄력적 IP 주소가 하나 이상 인스턴스에 연결되어 있는 경우에는 탄력적 IP 필드에서 탄력적 IP 주소를 획득합니다.

   Note

   인스턴스가 퍼블릭 IPv4 주소를 가지고 있지 않지만 사용자가 인스턴스에서 탄력적 IP 주소를 네트워크 인터페이스에 연결한 경우에는 IPv4 퍼블릭 IP 필드에 탄력적 IP 주소가 표시됩니다.

5. 아니면, 탐색 창에서 네트워크 인터페이스를 선택한 후 인스턴스에 연결된 네트워크 인터페이스를 선택합니다.
6. IPv4 퍼블릭 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. 자세한 내용은 Linux 인스턴스에 연결 (p. 485) 단원을 참조하십시오.
2. 다음 명령을 사용하여 프라이빗 IP 주소에 액세스합니다.

   IMDSv2

   ```
   [ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-
   aws-ec2-metadata-token-ttl-seconds: 21600"` \
   && curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/
   meta-data/local-ipv4
   ```

   IMDSv1

   ```
   [ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/local-ipv4
   ```

3. 다음 명령을 사용하여 퍼블릭 IP 주소에 액세스합니다.

   IMDSv2

   ```
   [ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-
   aws-ec2-metadata-token-ttl-seconds: 21600"` \
   && curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/
   meta-data/public-ipv4
   ```

   IMDSv1

   ```
   [ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-ipv4
   ```

   인스턴스와 탄력적 IP 주소가 연결된 경우 반환된 값은 탄력적 IP 주소입니다.

IPv6 주소 결정

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 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다. 세부 정보 창의 IPv6 IP에서 IPv6 주소를 가져옵니다.

인스턴스 메타데이터를 이용하여 인스턴스의 IPv6 주소를 결정하려면

1. 인스턴스에 연결합니다. 자세한 내용은 Linux 인스턴스에 연결 (p. 485) 단원을 참조하십시오.
2. 다음 명령을 사용하여 IPv6 주소를 봅니다(`http://169.254.169.254/latest/meta-data/network/interfaces/macs/`에서 MAC 주소를 가져올 수 있음).

   IMDSv2

   ```
   [ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
   && curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/network/interfaces/macs/mac-address/ipv6s
   ```

   IMDSv1

   ```
   [ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/network/interfaces/macs/mac-address/ipv6s
   ```

인스턴스 시작 시 퍼블릭 IPv4 주소 배정

각 서브넷은 퍼블릭 IP 주소가 할당되는 서브넷에서 인스턴스를 시작할지 여부를 결정하는 속성을 갖습니다. 기본적으로 기본이 아닌 서브넷의 이 속성은 false로 설정되고 기본 서브넷의 속성 값은 true입니다. 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. 659) 단원을 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. 676) 단원을 참조하십시오.

인스턴스 시작 시 퍼블릭 IP 주소 지정 기능에 액세스하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 인스턴스 시작을 선택합니다.
3. AMI와 인스턴스 유형을 선택한 후 다음: 인스턴스 세부 정보 구성(Next: Configure Instance Details)을 선택합니다.
4. 인스턴스 세부 정보 구성 페이지의 네트워크에서 VPC를 선택합니다. 퍼블릭 IP 자동 할당 목록이 표시됩니다. 활성화 또는 비활성화를 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 시작 마법사를 사용하여 인스턴스 시작 (p. 431) 단원을 참조하십시오. 마지막 인스턴스 시작 검토 페이지에서는 설정을 검토한 후 시작을 선택하여 키 페어를 선택하고 인스턴스를 시작합니다.
6. 인스턴스 페이지에서 새 인스턴스를 선택한 다음 세부 정보 창의 IPv4 퍼블릭 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. 676) 단원을 참조하십시오. 또한, 사용자는 서브넷의 퍼블릭 IPv4 주소 지정 동작을 변경할 수 있습니다. 자세한 정보는 서브넷의 퍼블릭 IPv4 주소 지정 속성 수정 단원을 참조하십시오.

명령줄을 사용한 퍼블릭 IP 주소 지정 기능의 활성화 또는 비활성화 방법

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- run-instances 명령(AWS CLI)에서 `--associate-public-ip-address` 또는 `--no-associate-public-ip-address` 옵션을 사용합니다.
- New-EC2Instance 명령(Windows PowerShell용 AWS 도구)에서 `-AssociatePublicIp` 파라미터를 사용합니다.

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.

IPv6는 모든 현재 세대 인스턴스 유형과 C3, R3 및 I2 이전 세대 인스턴스 유형에서 지원됩니다.

시작하는 과정에서 인스턴스에 IPv6 주소를 할당하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. IPv6을 지원하는 AMI와 인스턴스 유형을 선택하고 다음: 인스턴스 세부 정보 구성(Next: Configure Instance Details)을 선택합니다.
3. 인스턴스 세부 정보 구성 페이지의 네트워크에서 VPC를 선택하고 서브넷에서 서브넷을 선택합니다. IPv6 IP 자동 할당에 대해 활성화를 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 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 작업, 네트워킹, IP 주소 관리를 선택합니다.
4. IPv6 주소에서 새 IP 할당을 선택합니다. 서브넷 범위에 속한 IPv6 주소를 지정하거나, Amazon이 IPv6 주소를 자동으로 선택하도록 자동 할당 값을 그대로 둡니다.
5. 저장을 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 구성하기를 참조하십시오.

명령줄을 사용하여 IPv6 주소를 할당하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- run-instances(AWS CLI) 명령에서 `--ipv6-addresses` 옵션을 사용합니다.
- New-EC2Instance(Windows PowerShell용 AWS 도구) 명령에서 `-NetworkInterface`에 대한 `Ipv6Addresses` 속성을 사용합니다.
- 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 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 작업, 네트워킹, IP 주소 관리를 선택합니다.
4. IPv6 주소에서 할당을 해제할 IPv6 주소에 대해 할당 해제를 선택합니다.
5. 예, 업데이트를 선택합니다.

명령줄을 사용하여 IPv6 주소 할당을 해제하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- unassign-ipv6-addresses(AWS CLI)
- Unregister-EC2Ipv6AddressList(Windows PowerShell용 AWS 도구).

다중 IP 주소

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. 685) 단원을 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.

내용

- 다중 IP 주소 동작 방법 (p. 665)
- 다중 IPv4 주소 작업 (p. 666)
- 다중 IPv6 주소 작업 (p. 670)

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 인스턴스에 연결하지 않아도 됩니다.
- 연결된 IPv6 CIDR 블록이 있는 서브넷의 네트워크 인터페이스에 다중 IPv6 주소를 할당할 수 있습니다.
- 네트워크 인터페이스의 서브넷 IPv4 주소 CIDR 블록 범위 내에서 보조 IPv4를 선택해야 합니다.
- 네트워크 인터페이스의 서브넷 IPv6 CIDR 블록 범위 내에서 IPv6 주소를 선택해야 합니다.
- 보안 그룹을 개별 IP 주소가 아니라 네트워크 인터페이스와 연결합니다. 따라서 네트워크 인터페이스에 지정한 각 IP 주소가 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 기존 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 인스턴스에 연결되어도 기본 프라이빗 IPv4 주소, 보조 프라이빗 IPv4 주소, 탄력적 IP 주소 및 IPv6 주소는 보조 네트워크 인터페이스에 연결 상태를 유지합니다.
- 기본 네트워크 인터페이스는 인스턴스에서 분리할 수 없지만 기본 네트워크 인터페이스의 보조 프라이빗 IPv4 주소는 다른 네트워크 인터페이스로 재할당이 가능합니다.

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.

내용

- 보조 프라이빗 IPv4 주소 할당 (p. 666)
- 인스턴스에 운영 체제를 구성하여 보조 프라이빗 IPv4 주소 인식 (p. 668)
- 탄력적 IP 주소와 보조 프라이빗 IPv4 주소 연결 (p. 668)
- 보조 프라이빗 IPv4 주소 확인 (p. 669)
- 보조 프라이빗 IPv4 주소 할당 해제 (p. 669)

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.

- 인스턴스를 시작할 때 보조 프라이빗 IPv4 주소를 할당하려면 (p. 667)
- 명령줄을 이용하여 시작 중에 보조 IPv4 주소를 할당하려면 (p. 667)
- 네트워크 인터페이스에 보조 프라이빗 IPv4 주소를 할당하려면 (p. 667)
- 명령줄을 이용하여 기존 인스턴스에 보조 프라이빗 IPv4 주소를 할당하려면 (p. 668)

인스턴스를 시작할 때 보조 프라이빗 IPv4 주소를 할당하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 인스턴스 시작을 선택합니다.
3. AMI를 선택하고 인스턴스 유형을 선택한 후 다음: 인스턴스 세부 정보 구성(Next: Configure Instance Details)을 선택합니다.
4. 인스턴스 세부 정보 구성 페이지의 네트워크에서 VPC를 선택하고 서브넷에서 서브넷을 선택합니다.
5. 네트워크 인터페이스 섹션에서 다음을 수행하고 다음: 스토리지 추가(Next: Add Storage)를 선택합니다.
   - 다른 네트워크 인터페이스를 추가하려면 디바이스 추가를 선택합니다. 콘솔을 사용하여 인스턴스 시작 시 네트워크 인터페이스를 최대 두 개 지정할 수 있습니다. 인스턴스를 시작한 후 탐색 창에서 네트워크 인터페이스를 선택하여 네트워크 인터페이스를 추가합니다. 연결 가능한 총 네트워크 인터페이스의 수는 인스턴스 유형에 따라 다릅니다. 자세한 내용은 인스턴스 유형별/네트워크 인터페이스당 IP 주소 (p. 685) 단원을 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. 677) 단원을 참조하십시오.
   - 각 네트워크 인터페이스에 대해 보조 IP 주소에서 IP 추가를 선택한 후 서브넷 범위 내의 프라이빗 IP 주소를 입력하거나, 기본 설정인 `Auto-assign`을 수락하여 Amazon의 주소 선택을 허용합니다.
6. 다음 스토리지 추가 페이지에서 사용자는 볼륨을 지정하여 AMI에 의해 지정된 볼륨 옆에 인스턴스(루트 디바이스 볼륨 등)를 연결한 다음 다음: 태그 추가(Next: Add Tags)를 선택합니다.
7. 태그 추가 페이지에서 인스턴스에 태그(예: 사용자에게 친숙한 이름)를 지정한 후 다음: 보안 그룹 구성(Next: Configure Security Group)을 선택합니다.
8. 보안 그룹 구성 페이지에서 기존 보안 그룹을 선택하거나 새 보안 그룹을 생성합니다. 검토 및 시작을 선택합니다.
9. 인스턴스 시작 검토 페이지에서 설정을 검토한 후 시작을 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. 668) 단원을 참조하십시오.

명령줄을 이용하여 시작 중에 보조 IPv4 주소를 할당하려면

- 다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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 네트워크 인터페이스를 선택한 후 인스턴스에 연결된 네트워크 인터페이스를 선택합니다.
3. 작업, IP 주소 관리를 선택합니다.

4. IPv4 주소에서 새 IP 할당을 선택합니다.
5. 인스턴스의 서브넷 범위 내에 있는 특정 IPv4 주소를 입력합니다. 또는 필드를 공란으로 남기면 Amazon 에서 IP 주소를 자동으로 선택합니다.
6. (선택 사항) 재할당 허용을 선택하면 다른 네트워크 인터페이스가 이미 할당된 경우 보조 프라이빗 IP 주소가 재할당됩니다.
7. 예, 업데이트를 선택합니다.

대안으로, 사용자는 인스턴스에 보조 프라이빗 IPv4 주소를 할당할 수 있습니다. 탐색 창에서 인스턴스를 선택하고 인스턴스를 선택한 후 작업, 네트워킹, IP 주소 관리를 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. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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-utils는 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. 696) 단원을 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 구성을 참조합니다.

탄력적 IP 주소와 보조 프라이빗 IPv4 주소 연결

탄력적 IP 주소와 보조 프라이빗 IPv4 주소를 연결하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 탄력적 IP를 선택합니다.
3. 작업을 선택한 후 주소 연결을 선택합니다.
4. 네트워크 인터페이스에서 네트워크 인터페이스를 선택한 다음 프라이빗 IP 목록에서 보조 IP 주소를 선택합니다.
5. 연결을 선택합니다.

명령줄을 이용하여 탄력적 IP 주소와 보조 프라이빗 IPv4 주소를 연결하려면

- 다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- associate-address(AWS CLI)
- Register-EC2Address(Windows PowerShell용 AWS 도구)

보조 프라이빗 IPv4 주소 확인

네트워크 인터페이스에 할당된 프라이빗 IPv4 주소를 확인하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 네트워크 인터페이스를 선택합니다.
3. 확인할 프라이빗 IP 주소를 갖는 네트워크 인터페이스를 선택합니다.
4. 세부 정보 창의 세부 정보 탭에서 네트워크 인터페이스에 할당된 기본 프라이빗 IPv4 주소 및 모든 보조 프라이빗 IPv4 주소의 기본 프라이빗 IPv4 IP 및 보조 프라이빗 IPv4 IP 필드 확인란을 선택합니다.

인스턴스에 할당된 프라이빗 IPv4 주소를 확인하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 확인할 프라이빗 IPv4 주소를 갖는 인스턴스를 선택합니다.
4. 세부 정보 창의 설명 탭에서 네트워크 인터페이스를 통해 할당된 기본 프라이빗 IPv4 주소 및 모든 보조 프라이빗 IPv4 주소의 프라이빗 IP 및 보조 프라이빗 IP 필드 확인란을 선택합니다.

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 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 작업, 네트워킹, IP 주소 관리를 선택합니다.
4. IPv4 주소에서 할당을 해제할 IPv4 주소에 대해 할당 해제를 선택합니다.
5. 예, 업데이트를 선택합니다.

네트워크 인터페이스에서 보조 프라이빗 IPv4 주소의 할당을 해제하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 네트워크 인터페이스를 선택합니다.
3. 네트워크 인터페이스를 선택하고 작업, IP 주소 관리를 선택합니다.
4. IPv4 주소에서 할당을 해제할 IPv4 주소에 대해 할당 해제를 선택합니다.
5. 예, 업데이트를 선택합니다.

명령줄을 이용하여 보조 프라이빗 IPv4 주소의 할당을 해제하려면

- 다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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.

내용

- 다중 IPv6 주소 할당 (p. 670)
- IPv6 주소 확인 (p. 671)
- IPv6 주소 할당 해제 (p. 672)

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 설명서의 VPC 및 서브넷 을 참조하십시오.

시작 중에 다중 IPv6 주소 할당

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 대시보드에서 인스턴스 시작을 선택합니다.
3. AMI를 선택하고 인스턴스 유형을 선택한 후 다음: 인스턴스 세부 정보 구성(Next: Configure Instance Details)을 선택합니다. IPv6를 지원하는 인스턴스 유형을 선택해야 합니다. 자세한 내용은 인스턴스 유형 (p. 178) 단원을 참조하십시오.
4. 인스턴스 세부 정보 구성 페이지의 네트워크 목록에서 VPC를 선택한 다음 서브넷 목록에서 서브넷을 선택합니다.
5. 네트워크 인터페이스 섹션에서 다음을 수행하고 다음: 스토리지 추가(Next: Add Storage)를 선택합니다.
   - 기본 네트워크 인터페이스(eth0)에 단일 IPv6 주소를 할당하려면 IPv6 IP에서 IP 추가를 선택합니다. 보조 IPv6 주소를 추가하려면 IP 추가를 다시 선택합니다. 서브넷 범위에 속한 IPv6 주소를 직접 입력하거나, Amazon가 서브넷에 속한 IPv6 주소를 자동으로 선택하도록 기본 자동 할당 값을 그대로 둘 수 있습니다.
   - 다른 네트워크 인터페이스를 추가하려면 디바이스 추가를 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 네트워크 인터페이스를 선택하여 네트워크 인터페이스를 추가합니다. 연결 가능한 총 네트워크 인터페이스의 수는 인스턴스 유형에 따라 다릅니다. 자세한 내용은 인스턴스 유형별/네트워크 인터페이스당 IP 주소 (p. 685) 단원을 참조하십시오.
6. 마법사의 다음 단계를 수행하여 볼륨을 연결하고 인스턴스에 태그를 지정합니다.
7. 보안 그룹 구성 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. 880) 단원을 참조하십시오. 검토 및 시작을 선택합니다.
8. 인스턴스 시작 검토 페이지에서 설정을 검토한 후 시작을 선택하여 키 페어를 선택하고 인스턴스를 시작합니다. Amazon EC2를 처음 사용하며 아직 키 페어를 생성하지 않은 경우 키 페어를 생성하라는 메시지가 마법사에 표시됩니다.

인스턴스 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 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 작업, 네트워킹, IP 주소 관리를 선택합니다.
4. IPv6 주소에서 추가할 IPv6 주소에 대해 새 IP 할당을 선택합니다. 서브넷 범위에 속한 IPv6 주소를 지정하거나, Amazon이 IPv6 주소를 자동으로 선택하도록 자동 할당 값을 그대로 둡니다.
5. 예, 업데이트를 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 네트워크 인터페이스를 선택합니다.
3. 네트워크 인터페이스를 선택하고 작업, IP 주소 관리를 선택합니다.
4. IPv6 주소에서 추가할 IPv6 주소에 대해 새 IP 할당을 선택합니다. 서브넷 범위에 속한 IPv6 주소를 지정하거나, Amazon이 IPv6 주소를 자동으로 선택하도록 자동 할당 값을 그대로 둡니다.
5. 예, 업데이트를 선택합니다.

CLI 개요

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다. 세부 정보 창에서 IPv6 IP 필드를 검토합니다.

네트워크 인터페이스에 할당된 IPv6 주소를 확인하려면

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

3. 네트워크 인터페이스를 선택합니다. 세부 정보 창에서 IPv6 IP 필드를 검토합니다.

CLI 개요

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 작업, 네트워킹, IP 주소 관리를 선택합니다.
4. IPv6 주소에서 할당을 해제할 IPv6 주소에 대해 할당 해제를 선택합니다.
5. 예, 업데이트를 선택합니다.

네트워크 인터페이스에 할당된 IPv6 주소를 해제하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 네트워크 인터페이스를 선택합니다.
3. 네트워크 인터페이스를 선택하고 작업, IP 주소 관리를 선택합니다.
4. IPv6 주소에서 할당을 해제할 IPv6 주소에 대해 할당 해제를 선택합니다.
5. 저장을 선택합니다.

CLI 개요

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- unassign-ipv6-addresses(AWS CLI)
- Unregister-EC2Ipv6AddressList(Windows PowerShell용 AWS 도구).

고유 IP 주소 가져오기(BYOIP)

온프레미스 네트워크에서 AWS 계정으로 모든 퍼블릭 IPv4 주소 범위의 일부 또는 전체를 가져올 수 있습니다. 주소 범위를 계속해서 소유할 수 있지만 AWS에서는 인터넷에 대한 이러한 주소 범위를 알립니다. 주소 범위를 AWS로 가져오고 나면 이러한 주소가 계정에 주소 풀로 나타납니다. 주소 풀에서 탄력적 IP 주소를

생성하여 AWS 리소스(예: EC2 인스턴스, NAT 게이트웨이 및 Network Load Balancer)와 함께 사용할 수 있습니다.

> **Important**
>
> BYOIP는 일부 리전에서는 사용할 수 없습니다. 지원되는 리전 목록은 고유 IP 주소 가져오기에 대한 FAQ를 참조하십시오.

요구 사항

- 주소 범위를 ARIN(미국 인터넷 번호 등록 협회), RIPE(Réseaux IP Européens Network Coordination Centre) 또는 APNIC(아시아 태평양 지역 네트워크 정보 센터)와 같은 RIR(지역 인터넷 등록 기관)에 등록해야 합니다. 주소 범위를 사업체 또는 기관에 등록해야 하며 개인에게는 등록할 수 없습니다.
- 지정 가능한 가장 구체적인 주소 범위는 /24입니다.
- 각 주소 범위를 한 번에 하나의 리전으로 가져올 수 있습니다.
- 리전당 5개의 주소 범위를 AWS 계정으로 가져올 수 있습니다.
- IP 주소 범위의 주소에는 명확한 기록이 있어야 합니다. 당사는 IP 주소 범위의 평판을 조사할 수 있으며, 좋지 않은 평판이 있거나 악의적인 동작과 연관된 IP 주소가 포함될 경우 IP 주소 범위를 거부할 권한을 보유합니다.
- 사용하는 IP 주소를 소유해야 합니다. 즉, 다음 항목만 지원됩니다.
  - ARIN - "직접 할당" 및 "직접 지정" 네트워크 유형
  - RIPE - "ALLOCATED PA", "LEGACY" 및 "ASSIGNED PI" 할당 상태
  - APNIC – "ALLOCATED PORTABLE" 및 "ASSIGNED PORTABLE" 할당 상태

AWS 계정으로 주소 범위를 가져오도록 준비하기

자신만 주소 범위를 AWS 계정으로 가져올 수 있도록 하려면 주소 범위를 알릴 수 있는 권한을 Amazon에 부여해야 합니다. 또한 서명된 권한 부여 메시지를 통해 주소 범위를 소유하고 있다는 증명을 제공해야 합니다.

ROA(라우팅 오리진 권한 부여)는 RIR를 통해 생성할 수 있는 라우팅 공지에 대한 암호화 설명입니다. 이 문서에는 주소 범위, 주소 범위를 알리도록 허용된 자율 시스템 번호(ASN)와 만료 날짜가 포함되어 있습니다. ROA는 Amazon에 특정 AS 번호로 주소 범위를 알릴 수 있는 권한을 부여합니다. 그러나 사용자의 AWS 계정에는 AWS에 주소 범위를 가져올 수 있는 권한을 부여하지 않습니다. AWS 계정에 AWS로 주소 범위를 가져올 수 있는 권한을 부여하려면 주소 범위에 대한 등록 데이터 액세스 프로토콜(RDAP) 설명에 자체 서명된 X509 인증서를 게시해야 합니다. 이 인증서에는 제공한 권한 부여-컨텍스트 서명을 확인하기 위해 AWS가 사용하는 퍼블릭 키가 포함되어 있습니다. 프라이빗 키는 안전하게 보관해야 하고, 권한 부여-컨텍스트 메시지에 서명하는 데 사용됩니다.

이 작업의 명령은 Linux에서 지원됩니다. Windows에서는 Windows Subsystem for Linux를 사용하여 Linux 명령을 실행할 수 있습니다.

작업

- ROA 객체 생성 (p. 673)
- 자체 서명된 X509 인증서 생성 (p. 674)
- 서명된 권한 부여 메시지 생성 (p. 674)

ROA 객체 생성

ROA 객체를 생성하여 Amazon ASN 16509 및 14618이 주소 범위를 공급하도록 권한을 부여하고 현재 주소 범위의 공급 권한이 있는 ASN이 주소 범위를 공급하도록 권한을 부여합니다. 가져오려는 가장 작은 접두사 크기에 대한 최대 길이를 설정해야 합니다(예: /24). Amazon이 ROA를 사용할 수 있을 때까지 최대 24시간이 걸릴 수 있습니다. 자세한 내용은 다음 단원을 참조하십시오.

- ARIN - ROA 요청
- RIPE - ROA 관리
- APNIC — 라우팅 관리

자체 서명된 X509 인증서 생성

다음 절차에 따라 자체 서명된 X509 인증서를 생성하여 RIR의 RDAP 레코드에 추가하십시오. openssl 명령에는 OpenSSL 버전 1.0.2 이상이 필요합니다.

자체 서명된 X509 인증서를 생성하여 RIR 레코드에 추가하려면

1. 다음과 같이 RSA 2048비트 키 페어를 생성합니다.

```
openssl genrsa -out private.key 2048
```

2. 다음 명령을 사용하여 키 페어에서 퍼블릭 X509 인증서를 생성합니다. 이 예제에서 인증서는 365일이 지나면 만료되므로, 이 기간이 지난 후에는 신뢰할 수 없습니다. 따라서 만료 날짜를 적절하게 설정해야 합니다. 정보를 입력하라는 메시지가 표시되면 기본값을 수락할 수 있습니다.

```
openssl req -new -x509 -key private.key -days 365 | tr -d "\n" > publickey.cer
```

3. X509 인증서로 RIR에 대한 RDAP 레코드를 업데이트합니다. 인증서에서 `-----BEGIN CERTIFICATE-----` 및 `-----END CERTIFICATE-----`를 복사해야 합니다. 이전 단계에서 tr -d "\n" 명령을 사용해 아직 제거하지 않은 경우에는 줄바꿈 문자를 제거해야 합니다. 인증서를 보려면 다음 명령을 실행합니다.

```
cat publickey.cer
```

   ARIN의 경우 주소 범위의 "공개 주석" 섹션에 인증서를 추가합니다.

   RIPE의 경우 주소 범위의 새 "설명" 필드에 인증서를 추가합니다.

   APNIC의 경우 이메일을 통해 퍼블릭 키를 helpdesk@apnic.net로 전송하여 “설명” 필드에 수동으로 추가합니다. IP 주소의 APNIC 공인 연락처를 사용하여 이메일을 전송합니다.

서명된 권한 부여 메시지 생성

서명된 권한 부여 메시지의 형식은 다음과 같으며, 여기서 날짜는 메시지의 만료 날짜입니다.

```
1|aws|account|cidr|YYYYMMDD|SHA256|RSAPSS
```

먼저 일반 텍스트 권한 부여 메시지를 생성하고 다음과 같이 `text_message`라는 변수에 저장합니다. 예제 계정 번호, 주소 범위 및 만료 날짜를 해당 값으로 바꿉니다.

```
text_message="1|aws|123456789012|198.51.100.0/24|20191201|SHA256|RSAPSS"
```

그런 다음, 생성한 키 페어를 사용하여 `text_message`의 권한 부여 메시지에 서명하고 다음과 같이 `signed_message`라는 변수에 저장합니다.

```
signed_message=$(echo $text_message | tr -d "\n" | openssl dgst -sha256 -sigopt
 rsa_padding_mode:pss -sigopt rsa_pss_saltlen:-1 -sign private.key -keyform PEM | openssl
 base64 | tr -- '+=/' '-_~' | tr -d "\n")
```

AWS에서 사용할 수 있도록 주소 범위 프로비저닝

AWS에서 사용할 수 있도록 주소 범위를 프로비저닝하는 경우 주소 범위를 소유하고 있는지 확인하고 Amazon에 해당 주소 범위를 알릴 수 있는 권한을 부여하고 있는 것입니다. 또한 서명된 권한 부여 메시지를 통해 주소 범위를 소유하고 있음을 확인합니다. 이 메시지는 X509 인증서로 RDAP 레코드를 업데이트할 때 사용한 자체 서명된 X509 키 페어로 서명됩니다.

주소 범위를 프로비저닝하려면 다음 provision-byoip-cidr 명령을 사용합니다. 예제 주소 범위를 해당 주소 범위로 바꿉니다. `--cidr-authorization-context` 옵션은 ROA 메시지가 아니라 이전에 생성한 변수를 사용합니다.

```
aws ec2 provision-byoip-cidr --cidr address-range --cidr-authorization-context
 Message="$text_message",Signature="$signed_message"
```

주소 범위 프로비저닝은 비동기 작업이므로, 호출이 즉시 반환되지만 주소 범위는 상태가 `pending-provision`에서 `provisioned`로 변경되어야 사용할 준비가 된 것입니다. 프로비저닝 프로세스를 완료하려면 최대 5주까지 걸릴 수 있습니다. 프로비저닝한 주소 범위의 상태를 모니터링하려면 다음 describe-byoip-cidrs 명령을 사용합니다.

```
aws ec2 describe-byoip-cidrs --max-results 5
```

주소 풀에서 탄력적 IP 주소를 생성하려면 assign-address 명령을 사용합니다. `--public-ipv4-pool` 옵션을 사용하여 `describe-byoip-cidrs`에서 반환된 주소 풀의 ID를 지정할 수 있습니다. 또는 `--address` 옵션을 사용하여 프로비저닝한 주소 범위에서 주소를 지정할 수 있습니다.

AWS를 통해 주소 범위 알리기

주소 범위가 프로비저닝되면 알릴 준비가 된 것입니다. 프로비저닝한 정확한 주소 범위를 알려야 합니다. 프로비저닝한 주소 범위의 일부만 알릴 수 없습니다.

AWS를 통해 알리기 전에 다른 위치에서 주소 범위 알리기를 중지하는 것이 좋습니다. 다른 위치에서 IP 주소 범위 알리기를 계속하면 당사가 해당 주소 범위를 안정적으로 지원하지 못하거나 문제를 제대로 해결하지 못할 수 있습니다. 특히, 해당 주소 범위로의 트래픽이 당사 네트워크로 들어오는지 보장할 수 없습니다.

가동 중지 시간을 최소화하기 위해 주소를 알리기 전에 주소 풀의 주소를 사용하도록 AWS 리소스를 구성하고, 동시에 현재 위치에서 주소 알리기를 중지한 다음 AWS를 통해 주소 알리기를 시작할 수 있습니다. 주소 풀의 탄력적 IP 주소 할당에 대한 자세한 내용은 탄력적 IP 주소 할당 (p. 677) 단원을 참조하십시오.

주소 범위를 알리려면 다음 provision-byoip-cidr 명령을 사용합니다.

```
aws ec2 advertise-byoip-cidr --cidr address-range
```

> **Important**
>
> 매번 다른 주소 범위를 지정하더라도 최소 10초마다 advertise-byoip-cidr 명령을 실행할 수 있습니다.

주소 범위 알리기를 중지하려면 다음 withdraw-byoip-cidr 명령을 사용합니다.

```
aws ec2 withdraw-byoip-cidr --cidr address-range
```

> **Important**
>
> 매번 다른 주소 범위를 지정하더라도 최소 10초마다 withdraw-byoip-cidr 명령을 실행할 수 있습니다.

주소 범위 프로비저닝 취소

AWS에서 주소 범위 사용을 중지하려면 주소 풀에서 여전히 할당되는 탄력적 IP 주소를 모두 해제하고, 주소 범위 알리기를 중지한 다음, 주소 범위 프로비저닝을 취소합니다.

각 탄력적 IP 주소를 해제하려면 다음 release-address 명령을 사용합니다.

```
aws ec2 release-address --allocation-id eipalloc-12345678
```

주소 범위 알리기를 중지하려면 다음 withdraw-byoip-cidr 명령을 사용합니다.

```
aws ec2 withdraw-byoip-cidr --cidr address-range
```

주소 범위 프로비저닝을 취소하려면 다음 deprovision-byoip-cidr 명령을 사용합니다.

```
aws ec2 deprovision-byoip-cidr --cidr address-range
```

탄력적 IP 주소

탄력적 IP 주소는 동적 클라우드 컴퓨팅을 위해 고안된 정적 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 IPv4 주소가 없는 경우 탄력적 IP 주소를 인스턴스에 연결하여 인터넷 통신을 활성화할 수 있습니다. 예를 들어 로컬 컴퓨터에서 인스턴스에 연결할 수 있습니다.

현재는 IPv6에 대한 탄력적 IP 주소를 지원하지 않습니다.

내용

- 탄력적 IP 주소 기본 사항 (p. 676)
- 탄력적 IP 주소 작업 (p. 677)
- 이메일 애플리케이션에 역방향 DNS 사용 (p. 683)
- 탄력적 IP 주소 제한 (p. 683)

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 인스턴스와 연결하면 해당 인스턴스의 기본 네트워크 인터페이스와도 연결됩니다. 탄력적 IP 주소를 인스턴스에 연결된 네트워크 인터페이스와 연결하면 해당 인스턴스와도 연결됩니다.
- 탄력적 IP 주소를 인스턴스 또는 기본 네트워크 인터페이스와 연결하면, 인스턴스의 퍼블릭 IPv4 주소(있는 경우)는 Amazon의 퍼블릭 IPv4 주소 풀로 다시 릴리스됩니다. 퍼블릭 IPv4 주소를 재사용할 수 없으며, 퍼블릭 IPv4 주소를 탄력적 IP 주소로 변환할 수 없습니다. 자세한 내용은 퍼블릭 IPv4 주소 및 외부 DNS 호스트 이름 (p. 659) 단원을 참조하십시오.
- 탄력적 IP 주소는 리소스에서 연결 해제했다가 다른 리소스와 다시 연결할 수 있습니다. 인스턴스에 대해 실행 중인 연결은 탄력적 IP 주소 연결을 해제하고 다른 인스턴스에 다시 연결한 후에도 잠시 동안 계속 작동합니다. 다시 연결한 탄력적 IP 주소를 사용하여 이러한 연결을 다시 여는 것이 좋습니다.

- 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 정보는 Amazon EC2 요금을 참조하십시오.
- 탄력적 IP 주소는 특정 네트워크 경계 그룹에서만 사용할 수 있습니다.
- 탄력적 IP 주소를 이전에 퍼블릭 IPv4 주소가 있던 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.
- AWS 계정으로 가져온 IP 주소 풀에서 탄력적 IP 주소를 할당하는 경우 해당 IP 주소는 탄력적 IP 주소 한도에 포함되지 않습니다.
- 탄력적 IP 주소를 할당할 때 탄력적 IP 주소를 네트워크 경계 그룹에 연결할 수 있습니다. Amazon은 여기서 CIDR 블록을 알립니다. 네트워크 경계 그룹을 설정하면 CIDR 블록이 이 그룹으로 제한됩니다. 네트워크 경계 그룹을 지정하지 않으면 리전(예: `us-west-2`)의 모든 가용 영역을 포함하는 경계 그룹이 설정됩니다.

탄력적 IP 주소 작업

다음 섹션에서는 탄력적 IP 주소를 이용한 작업 방법에 대해 살펴보겠습니다.

작업

- 탄력적 IP 주소 할당 (p. 677)
- 탄력적 IP 주소 설명 (p. 678)
- 탄력적 IP 주소 태그 지정 (p. 679)
- 실행 중인 인스턴스 또는 네트워크 인터페이스와 탄력적 IP 주소 연결 (p. 680)
- 탄력적 IP 주소 연결 해제 (p. 681)
- 탄력적 IP 주소 해제 (p. 681)
- 탄력적 IP 주소 복구 (p. 682)

탄력적 IP 주소 할당

퍼블릭 IPv4 주소의 Amazon 풀 또는 AWS 계정으로 가져온 사용자 지정 IP 주소 풀에서 탄력적 IP 주소를 할당할 수 있습니다. AWS 계정으로 고유한 IP 주소 범위 가져오기에 대한 자세한 내용은 고유 IP 주소 가져오기(BYOIP) (p. 672) 단원을 참조하십시오.

다음 방법 중 하나를 사용하여 탄력적 IP 주소를 할당할 수 있습니다.

새로운 콘솔

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 Elastic IP address(탄력적 IP 주소 할당)를 선택합니다.
4. 사용할 범위에 따라 범위에서 VPC 또는 EC2-Classic을 선택합니다.
5. (VPC 범위에만 해당)Public IPv4 address pool(퍼블릭 IPv4 주소 풀)의 경우 다음 중 하나를 선택합니다.
   - Amazon의 IP 주소 풀 — IPv4 주소를 Amazon의 IP 주소 풀에서 할당하려는 경우.

- My pool of public IPv4 addresses(내 퍼블릭 IPv4 주소 풀) — AWS 계정으로 가져온 IP 주소 풀에서 IPv4 주소를 할당하려는 경우. IP 주소 풀이 없는 경우에는 이 옵션을 사용할 수 없습니다.
- Customer owned pool of IPv4 addresses(고객 소유 IPv4 주소 풀) — AWS Outpost에서 사용하기 위해 온프레미스 네트워크에서 만든 풀에서 IPv4 주소를 할당하려는 경우. AWS Outpost가 없는 경우 이 옵션이 비활성화됩니다.

6. [Allocate]를 선택합니다.

기존 콘솔

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. IPv4 주소 풀로 Amazon 풀을 선택합니다.
5. 할당을 선택하고 확인 화면을 닫습니다.

AWS CLI

탄력적 IP 주소를 할당하려면

allocate-address AWS CLI 명령을 사용합니다.

PowerShell

탄력적 IP 주소를 할당하려면

New-EC2Address Windows PowerShell용 AWS 도구 명령을 사용합니다.

탄력적 IP 주소 설명

다음 방법 중 하나를 사용하여 탄력적 IP 주소를 설명할 수 있습니다.

새로운 콘솔

탄력적 IP 주소를 설명하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 탄력적 IP를 선택합니다.
3. 보려는 탄력적 IP 주소를 선택하고 작업, 세부 정보 보기를 선택합니다.

기존 콘솔

탄력적 IP 주소를 설명하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 탄력적 IP를 선택합니다.
3. 리소스 속성 목록에서 필터를 선택하여 검색을 시작합니다. 단일 검색에 여러 필터를 사용할 수 있습니다.

AWS CLI

탄력적 IP 주소를 설명하려면

describe-addresses AWS CLI 명령을 사용합니다.

PowerShell

탄력적 IP 주소를 설명하려면

Get-EC2Address Windows PowerShell용 AWS 도구 명령을 사용합니다.

탄력적 IP 주소 태그 지정

탄력적 IP 주소에 사용자 지정 태그를 할당하여 용도, 소유자, 환경 등 다양한 방식으로 주소를 분류할 수 있습니다. 그러면 할당한 사용자 지정 태그를 기반으로 특정 탄력적 IP 주소를 빠르게 찾을 수 있습니다.

VPC 범위에 있는 탄력적 IP 주소에만 태그를 지정할 수 있습니다.

Note

탄력적 IP 주소 태그를 사용한 비용 할당 추적은 지원되지 않습니다.

다음 방법 중 하나를 사용하여 탄력적 IP 주소에 태그를 지정할 수 있습니다.

새로운 콘솔

탄력적 IP 주소를 태그하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Elastic IPs]를 선택합니다.
3. 태그를 지정할 탄력적 IP 주소를 선택하고 작업, 세부 정보 보기를 선택합니다.
4. 태그 섹션에서 태그 관리를 선택합니다.
5. 태그 키 및 값 페어를 지정합니다.
6. (선택 사항) 태그 추가를 선택하여 태그를 추가합니다.
7. 저장을 선택합니다.

기존 콘솔

탄력적 IP 주소를 태그하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Elastic IPs]를 선택합니다.
3. 태그를 지정할 탄력적 IP 주소를 선택하고 태그를 선택합니다.
4. [Add/Edit Tags]를 선택합니다.
5. 태그 추가/편집 대화 상자에서 태그 생성을 선택하고 태그에 대한 키와 값을 지정합니다.
6. (선택 사항) 태그 생성을 선택하여 탄력적 IP 주소에 추가 태그를 추가합니다.
7. 저장을 선택합니다.

AWS CLI

탄력적 IP 주소를 태그하려면

create-tags AWS CLI 명령을 사용합니다.

```
aws ec2 create-tags --resources eipalloc-12345678 --tags Key=Owner,Value=TeamA
```

PowerShell

탄력적 IP 주소를 태그하려면

New-EC2Tag Windows PowerShell용 AWS 도구 명령을 사용합니다.

`New-EC2Tag` 명령에는 탄력적 IP 주소 태그에 사용할 키-값 페어를 지정하는 `Tag` 파라미터가 필요합니다. 다음 명령은 `Tag` 파라미터를 생성합니다.

```
PS C:\> $tag = New-Object Amazon.EC2.Model.Tag
PS C:\> $tag.Key = "Owner"
PS C:\> $tag.Value = "TeamA"
```

```
PS C:\> New-EC2Tag -Resource eipalloc-12345678 -Tag $tag
```

실행 중인 인스턴스 또는 네트워크 인터페이스와 탄력적 IP 주소 연결

인스턴스와 탄력적 IP 주소를 연결하여 인터넷과 통신을 활성화하는 경우 인스턴스가 퍼블릭 서브넷에 위치해야 합니다. 자세한 내용은 Amazon VPC 사용 설명서의 인터넷 게이트웨이 단원을 참조하십시오.

다음 방법 중 하나를 사용하여 탄력적 IP 주소를 인스턴스 또는 네트워크 인터페이스에 연결할 수 있습니다.

새로운 콘솔

인스턴스와 엘라스틱 IP 주소를 연결하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 탄력적 IP를 선택합니다.
3. 연결할 탄력적 IP 주소를 선택하고 작업, 탄력적 IP 주소 연결을 선택합니다.
4. 리소스 유형에서 인스턴스를 선택합니다.
5. 예를 들어 탄력적 IP 주소를 연결할 인스턴스를 선택합니다. 텍스트를 입력하여 특정 인스턴스를 검색할 수도 있습니다.
6. (선택 사항) 프라이빗 IP 주소에 탄력적 IP 주소를 연결할 프라이빗 IP 주소를 지정합니다.
7. 연결을 선택합니다.

네트워크 인터페이스와 탄력적 IP 주소를 연결하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 탄력적 IP를 선택합니다.
3. 연결할 탄력적 IP 주소를 선택하고 작업, 탄력적 IP 주소 연결을 선택합니다.
4. [Resource type(리소스 유형)]에서 Network interface(네트워크 인터페이스)을 선택합니다.
5. 네트워크 인터페이스에서 탄력적 IP 주소를 연결할 네트워크 인터페이스를 선택합니다. 텍스트를 입력하여 특정 네트워크 인터페이스를 검색할 수도 있습니다.
6. (선택 사항) 프라이빗 IP 주소에 탄력적 IP 주소를 연결할 프라이빗 IP 주소를 지정합니다.
7. 연결을 선택합니다.

기존 콘솔

인스턴스와 엘라스틱 IP 주소를 연결하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 탄력적 IP를 선택합니다.
3. 탄력적 IP 주소를 선택하고 작업, 주소 연결을 선택합니다.

4. 인스턴스에서 인스턴스를 선택한 후 연결을 선택합니다.

AWS CLI

탄력적 IP 주소를 연결하려면

associate-address AWS CLI 명령을 사용합니다.

PowerShell

탄력적 IP 주소를 연결하려면

Register-EC2Address Windows PowerShell용 AWS 도구 명령을 사용합니다.

탄력적 IP 주소 연결 해제

언제든지 인스턴스 또는 네트워크 인터페이스에서 탄력적 IP 주소의 연결을 해제할 수 있습니다. 탄력적 IP 주소의 연결을 해제한 후 다른 리소스와 다시 연결할 수 있습니다.

다음 방법 중 하나를 사용하여 탄력적 IP 주소의 연결을 해제할 수 있습니다.

새로운 콘솔

탄력적 IP 주소의 연결을 해제하고 다시 연결하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 탄력적 IP를 선택합니다.
3. 연결을 해제할 탄력적 IP 주소를 선택하고 작업, 탄력적 IP 주소 연결 해제를 선택합니다.
4. 연결 해제를 선택합니다.

기존 콘솔

탄력적 IP 주소의 연결을 해제하고 다시 연결하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 탄력적 IP를 선택합니다.
3. 탄력적 IP 주소를 선택하고 작업을 선택한 후 주소 연결 해제를 선택합니다.
4. 주소 연결 해제를 선택합니다.

AWS CLI

엘라스틱 IP 주소를 연결 해제하려면

disassociate-address AWS CLI 명령을 사용합니다.

PowerShell

엘라스틱 IP 주소를 연결 해제하려면

Unregister-EC2Address Windows PowerShell용 AWS 도구 명령을 사용합니다.

탄력적 IP 주소 해제

탄력적 IP 주소가 더 이상 필요하지 않으면 다음 방법 중 하나를 사용하여 해제하는 것이 좋습니다. 해제한 주소는 인스턴스와 연결되어서는 안 됩니다.

새로운 콘솔

엘라스틱 IP 주소를 해제합니다

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Elastic IPs]를 선택합니다.
3. 해제할 탄력적 IP 주소를 선택하고 작업, 탄력적 IP 주소 해제를 선택합니다.
4. 릴리스를 선택합니다.

기존 콘솔

엘라스틱 IP 주소를 해제합니다

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Elastic IPs]를 선택합니다.
3. 탄력적 IP 주소를 선택하고 작업을 선택한 후 주소 릴리스를 선택합니다. 메시지가 나타나면 릴리스를 선택합니다.

AWS CLI

엘라스틱 IP 주소를 해제합니다

release-address AWS CLI 명령을 사용합니다.

PowerShell

엘라스틱 IP 주소를 해제합니다

Remove-EC2Address Windows PowerShell용 AWS 도구 명령을 사용합니다.

탄력적 IP 주소 복구

탄력적 IP 주소를 릴리스한 경우 해당 주소를 복구할 수 있습니다. 다음 규칙이 적용됩니다.

- 탄력적 IP 주소가 다른 AWS 계정에 할당되었거나, 탄력적 IP 주소 한도를 초과하는 경우에는 탄력적 IP 주소를 복구할 수 없습니다.
- 탄력적 IP 주소와 연결된 태그는 복구할 수 있습니다.
- Amazon EC2 API 또는 명령줄 도구만을 사용하여 탄력적 IP 주소를 복구할 수 있습니다.

AWS CLI

탄력적 IP 주소를 복구하려면

다음과 같이 allocate-address AWS CLI 명령을 사용하고, `--address` 파라미터를 사용하여 IP 주소를 지정합니다.

```
aws ec2 allocate-address --domain vpc --address 203.0.113.3
```

PowerShell

탄력적 IP 주소를 복구하려면

다음과 같이 New-EC2Address Windows PowerShell용 AWS 도구 명령을 사용하고 `-Address` 파라미터를 사용하여 IP 주소를 지정합니다.

```
PS C:\> New-EC2Address -Address 203.0.113.3 -Domain vpc -Region us-east-1
```

이메일 애플리케이션에 역방향 DNS 사용

인스턴스에서 타사에 이메일을 보내려는 경우 탄력적 IP 주소를 하나 이상 프로비저닝하여 AWS에 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 지정하면 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 탄력적 IP 주소 수를 확인하려면 https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 열고 탐색 창에서 탄력적 IP를 선택합니다.

탄력적 IP 주소에 대한 현재 계정 제한을 확인하려면 다음 중 하나를 수행합니다.

- https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 열고 탐색 창에서 제한을 선택한 다음 검색 필드에 **`IP`**를 입력합니다.
- https://console.aws.amazon.com/servicequotas/에서 Service Quotas 콘솔을 열고 검색 필드에 **`Amazon EC2`**를 입력한 다음 Amazon Elastic Compute Cloud(Amazon EC2)를 선택합니다. 검색 필드에 **`IP`**을 입력합니다.

아키텍처에서 추가 탄력적 IP 주소를 보증한다고 생각되면 Service Quotas 콘솔에서 직접 할당량 증가를 요청할 수 있습니다.

탄력적 네트워크 인터페이스

탄력적 네트워크 인터페이스(이 문서에서는 네트워크 인터페이스로 표시)는 VPC에서 가상 네트워크 카드를 나타내는 논리적 네트워킹 구성 요소입니다.

네트워크 인터페이스에는 다음 속성이 포함될 수 있습니다.

- VPC의 IPv4 주소 범위 중 기본 프라이빗 IPv4 주소
- VPC의 IPv4 주소 범위 중 하나 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

계정에서 네트워크 인터페이스를 생성 및 구성하고 VPC의 인스턴스에 연결할 수 있습니다. 계정에는 사용자가 다른 리소스 및 서비스를 사용할 수 있도록 AWS 서비스가 생성하고 관리하는 요청자 관리 네트워크 인터페이스가 있을 수도 있습니다. 이러한 네트워크 인터페이스는 사용자가 직접 관리할 수 없습니다. 자세한 내용은 요청자 관리 네트워크 인터페이스 (p. 705) 단원을 참조하십시오.

모든 네트워크 인터페이스에는 eni-로 시작하는 리소스 식별자가 있습니다.

> **Important**
>
> '탄력적 네트워크 인터페이스'는 약어 'ENI'로 부르기도 합니다. 이것은 일부 인스턴스 유형에서 네트워크 성능을 최적화하는 사용자 지정 인스턴스인 탄력적 네트워크 어댑터(ENA)와는 다릅니다. 자세한 내용은 Linux에서 향상된 네트워킹 (p. 706) 단원을 참조하십시오.

내용

- 네트워크 인터페이스 기본 사항 (p. 684)
- 인스턴스 유형별/네트워크 인터페이스당 IP 주소 (p. 685)
- 네트워크 인터페이스 시나리오 (p. 694)
- 네트워크 인터페이스 구성 모범 사례 (p. 696)
- 네트워크 인터페이스 작업 (p. 697)
- 요청자 관리 네트워크 인터페이스 (p. 705)

네트워크 인터페이스 기본 사항

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.

사용자가 네트워크 인터페이스의 속성을 수정할 수도 있습니다. 예를 들어 보안 그룹을 변경하고 IP 주소를 관리할 수 있습니다.

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. 685) 단원을 참조하십시오.

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 동작을 참조하십시오. 퍼블릭 IPv4 주소는 Amazon의 퍼블릭 IPv4 주소 풀에서 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. 659) 단원을 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 동작을 참조하십시오. 인스턴스를 시작하면 생성된 기본 네트워크 인터페이스(eth0)에 IPv6 주소가 할당됩니다.

자세한 내용은 IPv6 주소 (p. 660) 단원을 참조하십시오.

IP 트래픽 모니터링

네트워크 인터페이스에서 VPC 흐름 로그를 활성화하여 네트워크 인터페이스로 주고 받는 IP 트래픽에 대한 정보를 캡처합니다. 흐름 로그를 생성하고 난 다음 Amazon CloudWatch Logs의 데이터를 확인하고 가져올 수 있습니다. 자세한 내용은 Amazon VPC 사용 설명서의 VPC 흐름 로그를 참조하십시오.

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. IPv6 주소는 퍼블릭이며 인터넷으로는 접속할 수 없습니다. 자세한 내용은 다중 IP 주소 (p. 665) 단원을 참조하십시오. VPC에서 IPv6에 대한 자세한 내용은 Amazon VPC 사용 설명서의 VPC에서 IP 주소 지정 단원을 참조하십시오.

인스턴스 유형	최대 네트워크 인터페이스	인터페이스당 프라이빗 IPv4 주소 수	인터페이스당 IPv6 주소 수
`a1.medium`	2	4	4
`a1.large`	3	10	10
`a1.xlarge`	4	15	15
`a1.2xlarge`	4	15	15
`a1.4xlarge`	8	30	30
`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
`c5.large`	3	10	10
`c5.xlarge`	4	15	15

인스턴스 유형	최대 네트워크 인터페이스	인터페이스당 프라이빗 IPv4 주소 수	인터페이스당 IPv6 주소 수
c5.2xlarge	4	15	15
c5.4xlarge	8	30	30
c5.9xlarge	8	30	30
c5.12xlarge	8	30	30
c5.18xlarge	15	50	50
c5.24xlarge	15	50	50
c5.metal	15	50	50
c5d.large	3	10	10
c5d.xlarge	4	15	15
c5d.2xlarge	4	15	15
c5d.4xlarge	8	30	30
c5d.9xlarge	8	30	30
c5d.12xlarge	8	30	30
c5d.18xlarge	15	50	50
c5d.24xlarge	15	50	50
c5d.metal	15	50	50
c5n.large	3	10	10
c5n.xlarge	4	15	15
c5n.2xlarge	4	15	15
c5n.4xlarge	8	30	30
c5n.9xlarge	8	30	30
c5n.18xlarge	15	50	50
c5n.metal	15	50	50
cc2.8xlarge	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
f1.2xlarge	4	15	15

인스턴스 유형	최대 네트워크 인터페이스	인터페이스당 프라이빗 IPv4 주소 수	인터페이스당 IPv6 주소 수
f1.4xlarge	8	30	30
f1.16xlarge	8	50	50
g2.2xlarge	4	15	IPv6는 지원되지 않습니다
g2.8xlarge	8	30	IPv6는 지원되지 않습니다
g3s.xlarge	4	15	15
g3.4xlarge	8	30	30
g3.8xlarge	8	30	30
g3.16xlarge	15	50	50
g4dn.xlarge	3	10	10
g4dn.2xlarge	3	10	10
g4dn.4xlarge	3	10	10
g4dn.8xlarge	4	15	15
g4dn.12xlarge	8	30	30
g4dn.16xlarge	4	15	15
h1.2xlarge	4	15	15
h1.4xlarge	8	30	30
h1.8xlarge	8	30	30
h1.16xlarge	15	50	50
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
i3.16xlarge	15	50	50
i3.metal	15	50	50

인스턴스 유형	최대 네트워크 인터페이스	인터페이스당 프라이빗 IPv4 주소 수	인터페이스당 IPv6 주소 수
i3en.large	3	10	10
i3en.xlarge	4	15	15
i3en.2xlarge	4	15	15
i3en.3xlarge	4	15	15
i3en.6xlarge	8	30	30
i3en.12xlarge	8	30	30
i3en.24xlarge	15	50	50
i3en.metal	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
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
m5.large	3	10	10
m5.xlarge	4	15	15
m5.2xlarge	4	15	15
m5.4xlarge	8	30	30
m5.8xlarge	8	30	30

인스턴스 유형	최대 네트워크 인터페이스	인터페이스당 프라이빗 IPv4 주소 수	인터페이스당 IPv6 주소 수
m5.12xlarge	8	30	30
m5.16xlarge	15	50	50
m5.24xlarge	15	50	50
m5.metal	15	50	50
m5a.large	3	10	10
m5a.xlarge	4	15	15
m5a.2xlarge	4	15	15
m5a.4xlarge	8	30	30
m5a.8xlarge	8	30	30
m5a.12xlarge	8	30	30
m5a.16xlarge	15	50	50
m5a.24xlarge	15	50	50
m5ad.large	3	10	10
m5ad.xlarge	4	15	15
m5ad.2xlarge	4	15	15
m5ad.4xlarge	8	30	30
m5ad.8xlarge	8	30	30
m5ad.12xlarge	8	30	30
m5ad.16xlarge	15	50	50
m5ad.24xlarge	15	50	50
m5d.large	3	10	10
m5d.xlarge	4	15	15
m5d.2xlarge	4	15	15
m5d.4xlarge	8	30	30
m5d.8xlarge	8	30	30
m5d.12xlarge	8	30	30
m5d.16xlarge	15	50	50
m5d.24xlarge	15	50	50
m5d.metal	15	50	50
m5dn.large	3	10	10

인스턴스 유형	최대 네트워크 인터페이스	인터페이스당 프라이빗 IPv4 주소 수	인터페이스당 IPv6 주소 수
m5dn.xlarge	4	15	15
m5dn.2xlarge	4	15	15
m5dn.4xlarge	8	30	30
m5dn.8xlarge	8	30	30
m5dn.12xlarge	8	30	30
m5dn.16xlarge	15	50	50
m5dn.24xlarge	15	50	50
m5n.large	3	10	10
m5n.xlarge	4	15	15
m5n.2xlarge	4	15	15
m5n.4xlarge	8	30	30
m5n.8xlarge	8	30	30
m5n.12xlarge	8	30	30
m5n.16xlarge	15	50	50
m5n.24xlarge	15	50	50
p2.xlarge	4	15	15
p2.8xlarge	8	30	30
p2.16xlarge	8	30	30
p3.2xlarge	4	15	15
p3.8xlarge	8	30	30
p3.16xlarge	8	30	30
p3dn.24xlarge	15	50	50
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

인스턴스 유형	최대 네트워크 인터페이스	인터페이스당 프라이빗 IPv4 주소 수	인터페이스당 IPv6 주소 수
r4.4xlarge	8	30	30
r4.8xlarge	8	30	30
r4.16xlarge	15	50	50
r5.large	3	10	10
r5.xlarge	4	15	15
r5.2xlarge	4	15	15
r5.4xlarge	8	30	30
r5.8xlarge	8	30	30
r5.12xlarge	8	30	30
r5.16xlarge	15	50	50
r5.24xlarge	15	50	50
r5.metal	15	50	50
r5a.large	3	10	10
r5a.xlarge	4	15	15
r5a.2xlarge	4	15	15
r5a.4xlarge	8	30	30
r5a.8xlarge	8	30	30
r5a.12xlarge	8	30	30
r5a.16xlarge	15	50	50
r5a.24xlarge	15	50	50
r5ad.large	3	10	10
r5ad.xlarge	4	15	15
r5ad.2xlarge	4	15	15
r5ad.4xlarge	8	30	30
r5ad.8xlarge	8	30	30
r5ad.12xlarge	8	30	30
r5ad.16xlarge	15	50	50
r5ad.24xlarge	15	50	50
r5d.large	3	10	10
r5d.xlarge	4	15	15

인스턴스 유형	최대 네트워크 인터페이스	인터페이스당 프라이빗 IPv4 주소 수	인터페이스당 IPv6 주소 수
r5d.2xlarge	4	15	15
r5d.4xlarge	8	30	30
r5d.8xlarge	8	30	30
r5d.12xlarge	8	30	30
r5d.16xlarge	15	50	50
r5d.24xlarge	15	50	50
r5d.metal	15	50	50
r5dn.large	3	10	10
r5dn.xlarge	4	15	15
r5dn.2xlarge	4	15	15
r5dn.4xlarge	8	30	30
r5dn.8xlarge	8	30	30
r5dn.12xlarge	8	30	30
r5dn.16xlarge	15	50	50
r5dn.24xlarge	15	50	50
r5n.large	3	10	10
r5n.xlarge	4	15	15
r5n.2xlarge	4	15	15
r5n.4xlarge	8	30	30
r5n.8xlarge	8	30	30
r5n.12xlarge	8	30	30
r5n.16xlarge	15	50	50
r5n.24xlarge	15	50	50
t1.micro	2	2	IPv6는 지원되지 않습니다
t2.nano	2	2	2
t2.micro	2	2	2
t2.small	3	4	4
t2.medium	3	6	6
t2.large	3	12	12
t2.xlarge	3	15	15

인스턴스 유형	최대 네트워크 인터페이스	인터페이스당 프라이빗 IPv4 주소 수	인터페이스당 IPv6 주소 수
t2.2xlarge	3	15	15
t3.nano	2	2	2
t3.micro	2	2	2
t3.small	3	4	4
t3.medium	3	6	6
t3.large	3	12	12
t3.xlarge	4	15	15
t3.2xlarge	4	15	15
t3a.nano	2	2	2
t3a.micro	2	2	2
t3a.small	2	4	4
t3a.medium	3	6	6
t3a.large	3	12	12
t3a.xlarge	4	15	15
t3a.2xlarge	4	15	15
u-6tb1.metal	5	30	30
u-9tb1.metal	5	30	30
u-12tb1.metal	5	30	30
u-18tb1.metal	15	50	50
u-24tb1.metal	15	50	50
x1.16xlarge	8	30	30
x1.32xlarge	8	30	30
x1e.xlarge	3	10	10
x1e.2xlarge	4	15	15
x1e.4xlarge	4	15	15
x1e.8xlarge	4	15	15
x1e.16xlarge	8	30	30
x1e.32xlarge	8	30	30
z1d.large	3	10	10
z1d.xlarge	4	15	15

인스턴스 유형	최대 네트워크 인터페이스	인터페이스당 프라이빗 IPv4 주소 수	인터페이스당 IPv6 주소 수
`z1d.2xlarge`	4	15	15
`z1d.3xlarge`	8	30	30
`z1d.6xlarge`	8	30	30
`z1d.12xlarge`	15	50	50
`z1d.metal`	15	50	50

네트워크 인터페이스 시나리오

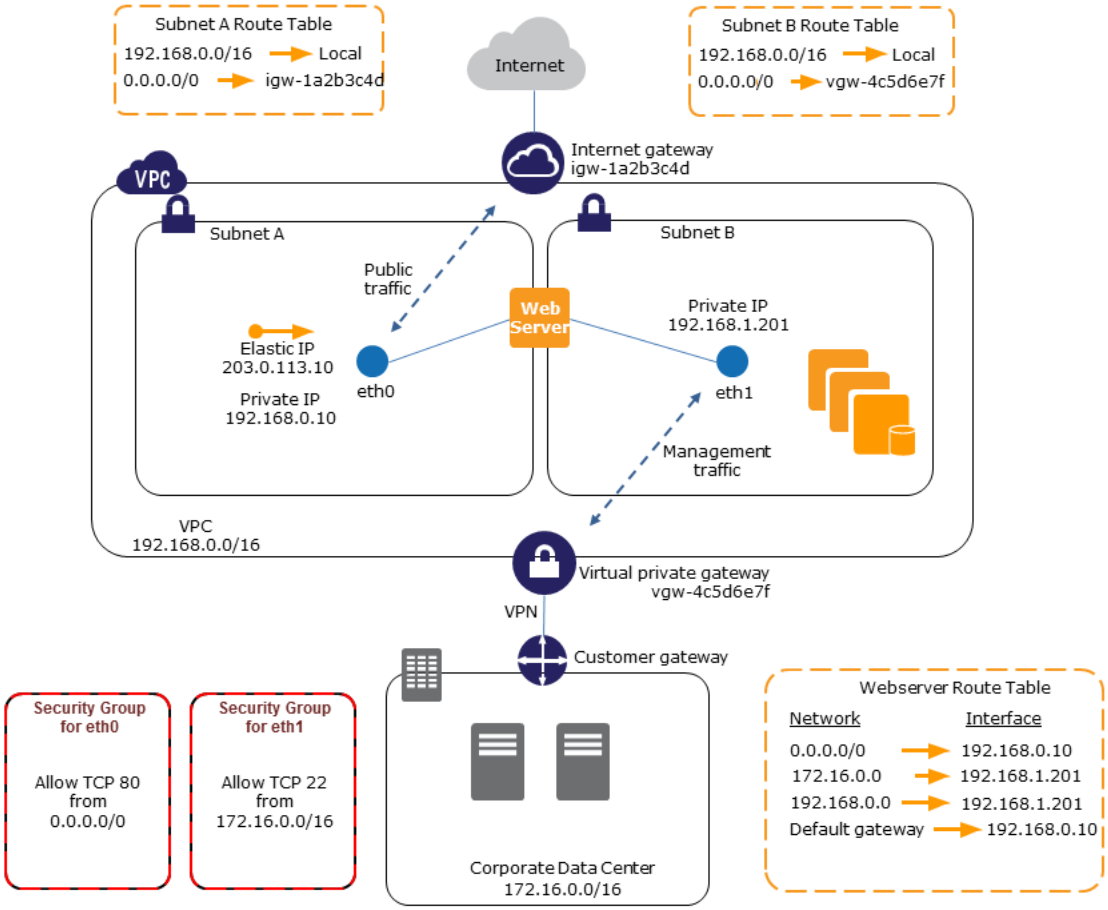
다음을 수행하려는 경우 여러 네트워크 인터페이스를 하나의 인스턴스에 연결하면 유용합니다.

- 관리 네트워크 생성
- VPC에서 네트워크 및 보안 어플라이언스 사용
- 별도의 서브넷에 워크로드/역할이 있는 이중 홈 인스턴스 생성
- 저예산 고가용성 솔루션 생성

관리 네트워크 생성

네트워크 인터페이스를 사용하여 관리 네트워크를 생성할 수 있습니다. 이 시나리오에서는 인스턴스의 주 네트워크 인터페이스(eth0)가 퍼블릭 트래픽을 처리하고 보조 네트워크 인터페이스(eth1)가 백엔드 관리 트래픽을 처리합니다. 또한 보조 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 `0.0.0.0/0`의 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 홈 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 홈 상태로 연결할 수 있습니다. 이중 홈 인스턴스를 통한 라우팅 네트워크 패킷 대신 각 이중 홈 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 인스턴

스에 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 보조 네트워크 인터페이스를 분리할 수 있습니다. 하지만 기본 네트워크 인터페이스(eth0)는 분리할 수 없습니다.
- 인스턴스가 동일한 가용 영역과 VPC에 있지만 서로 다른 서브넷에 있는 경우 네트워크 인터페이스를 한 인스턴스에서 다른 인스턴스로 이동할 수 있습니다.
- CLI, API 또는 SDK에서 인스턴스를 시작할 때 기본 네트워크 인터페이스(eth0) 및 추가 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. 666) 단원을 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 드라이브 이름(`75-persistent-net-generator.rules`, 여기서 `70-persistent-net.rules`를 생성)에 매핑합니다.

핫플러그 스크립트

DHCP에서 사용하기에 적합한 인터페이스 구성 파일을 생성합니다(`/etc/sysconfig/network-scripts/ifcfg-ethN`). 또한 라우팅 구성 파일도 생성합니다(`/etc/sysconfig/network-scripts/route-ethN`).

DHCP 스크립트

네트워크 인터페이스에서 새 DHCP 임대를 수신할 때마다 이 스크립트는 인스턴스 메타데이터에 탄력적 IP 주소를 쿼리합니다. 각 탄력적 IP 주소마다 라우팅 정책 데이터베이스에 규칙을 추가하여 해당 주

소의 아웃바운드 트래픽에 올바른 네트워크 인터페이스가 사용되도록 합니다. 또한 각 프라이빗 IP 주소를 네트워크 인터페이스에 부 주소로 추가합니다.

ec2ifup ethN

표준 ifup의 기능을 확장합니다. 이 스크립트는 구성 파일 `ifcfg-ethN` 및 `route-ethN`을 다시 쓴 후 ifup을 실행합니다.

ec2ifdown ethN

표준 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=no`를 `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 콘솔 또는 명령줄을 사용하여 네트워크 인터페이스 작업을 수행할 수 있습니다.

내용

- 네트워크 인터페이스 생성 (p. 698)
- 네트워크 인터페이스 삭제 (p. 698)
- 네트워크 인터페이스에 대한 세부 정보 보기 (p. 698)
- 인스턴스 시작 시 네트워크 인터페이스 연결 (p. 699)
- 중지되었거나 실행 중인 인스턴스에 네트워크 인터페이스 연결 (p. 700)
- 인스턴스에서 네트워크 인터페이스 분리 (p. 701)
- 보안 그룹 변경 (p. 701)
- 원본 또는 대상 확인 변경 (p. 702)
- 탄력적 IP 주소(IPv4) 연결 (p. 702)
- 탄력적 IP 주소(IPv4) 연결 해제 (p. 703)
- IPv6 주소 할당 (p. 703)
- IPv6 주소 할당 해제 (p. 704)
- 종료 동작 변경 (p. 704)
- 설명 추가 또는 편집 (p. 705)
- 태그 추가 또는 편집 (p. 705)

네트워크 인터페이스 생성

서브넷에서 네트워크 인터페이스를 생성할 수 있습니다. 네트워크 인터페이스는 일단 생성되고 나면 다른 서브넷으로 옮길 수 없으며 동일 가용 영역의 인스턴스에만 네트워크 인터페이스를 연결할 수 있습니다.

콘솔을 사용하여 네트워크 인터페이스를 생성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 네트워크 인터페이스를 선택합니다.
3. 네트워크 인터페이스 생성을 선택합니다.
4. 설명에 설명 이름을 입력합니다.
5. 서브넷에서 서브넷을 선택합니다.
6. 프라이빗 IP(또는 IPv4 프라이빗 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. 보안 그룹에서 하나 이상의 보안 그룹을 선택합니다.
9. 예, 생성을 선택합니다.

명령줄을 사용하여 네트워크 인터페이스를 생성하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- create-network-interface(AWS CLI)
- New-EC2NetworkInterface(Windows PowerShell용 AWS 도구)

네트워크 인터페이스 삭제

인스턴스를 삭제하려면 먼저 네트워크 인터페이스를 분리해야 합니다. 네트워크 인터페이스를 삭제하면 인터페이스와 연결된 모든 속성이 해제되고 다른 인스턴스에서 사용할 수 있도록 프라이빗 IP 주소나 탄력적 IP 주소가 해제됩니다.

콘솔을 사용하여 네트워크 인터페이스를 삭제하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 네트워크 인터페이스를 선택합니다.
3. 네트워크 인터페이스를 선택하고 삭제를 선택합니다.
4. 네트워크 인터페이스 삭제 대화 상자에서 예, 삭제를 선택합니다.

명령줄을 사용하여 네트워크 인터페이스를 삭제하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- delete-network-interface(AWS CLI)
- Remove-EC2NetworkInterface(Windows PowerShell용 AWS 도구)

네트워크 인터페이스에 대한 세부 정보 보기

계정에서 모든 네트워크 인스턴스를 볼 수 있습니다.

콘솔을 사용하여 네트워크 인터페이스를 설명하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 네트워크 인터페이스를 선택합니다.
3. 네트워크 인터페이스를 선택합니다.
4. 세부 정보를 확인하려면 세부 정보를 선택합니다.

명령줄을 사용하여 네트워크 인터페이스를 설명하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- describe-network-interface-attribute(AWS CLI)
- Get-EC2NetworkInterfaceAttribute(Windows PowerShell용 AWS 도구)

인스턴스 시작 시 네트워크 인터페이스 연결

인스턴스를 시작할 때 기존 네트워크 인터페이스를 지정하거나 네트워크 인터페이스를 추가적으로 연결할 수 있습니다.

> **Note**
>
> 인스턴스에 네트워크 인터페이스를 연결할 대 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. 인스턴스 시작을 선택합니다.
3. AMI와 인스턴스 유형을 선택하고 다음: 인스턴스 세부 정보 구성(Next: Configure Instance Details)을 선택합니다.
4. 인스턴스 세부 정보 구성 페이지의 네트워크에서 VPC를, 서브넷에서 서브넷을 선택합니다.
5. 콘솔을 사용하면 네트워크 인터페이스 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. 685) 단원을 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 IP에서 IP 추가를 선택합니다. 보조 IPv6 주소를 추가하려면 IP 추가를 다시 선택합니다. 서브넷 범위에 속한 IPv6 주소를 직접 입력하거나, Amazon가 서브넷에 속한 IPv6 주소를 자동으로 선택하도록 기본 자동 할당 값을 그대로 둘 수 있습니다.
7. 다음: 스토리지 추가를 선택합니다.
8. 스토리지 추가 페이지에서 사용자는 볼륨을 지정하여 AMI에 의해 지정된 볼륨 옆에 인스턴스(루트 디바이스 볼륨 등)를 연결한 다음 다음: 태그 추가를 선택합니다.
9. 태그 추가 페이지에서 인스턴스에 태그(예: 사용자에게 친숙한 이름)를 지정한 후 다음: 보안 그룹 구성 (Next: Configure Security Group)을 선택합니다.
10. 보안 그룹 구성 페이지에서 보안 그룹을 선택하거나 새 보안 그룹을 만들 수 있습니다. 검토 및 시작을 선택합니다.

    Note

    5단계에서 기존 네트워크 인터페이스를 지정한 경우, 이 단계에서 어떤 옵션을 선택하든 상관없이 인스턴스는 그 네트워크 인터페이스에 대한 보안 그룹과 연결됩니다.
11. 인스턴스 시작 검토 페이지에 기본 및 추가 네트워크 인터페이스에 대한 세부 정보가 표시됩니다. 설정을 검토한 다음 시작을 선택하여 키 페어를 선택하고 인스턴스를 시작합니다. Amazon EC2를 처음으로 사용하는 것이고 키 페어를 생성하지 않은 경우 새로운 키 페어를 생성하는 메시지가 마법사에 표시됩니다.

명령줄 사용하여 인스턴스 시작 시 네트워크 인터페이스를 연결하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- run-instances(AWS CLI)
- New-EC2Instance(Windows PowerShell용 AWS 도구)

중지되었거나 실행 중인 인스턴스에 네트워크 인터페이스 연결

Amazon EC2 콘솔의 인스턴스 또는 네트워크 인터페이스 페이지를 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. 659) 단원을 참조하십시오.

Instances 페이지를 사용하여 인스턴스에 네트워크 인터페이스를 연결하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 작업, 네트워킹, 네트워크 인터페이스 연결을 선택합니다.
4. 네트워크 인터페이스 연결 대화 상자에서 네트워크 인터페이스를 선택한 다음 연결을 선택합니다.

Network Interfaces 페이지를 사용하여 네트워크 인터페이스를 인스턴스에 연결하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 네트워크 인터페이스를 선택합니다.
3. 네트워크 인터페이스를 선택하고 연결을 선택합니다.

4. 네트워크 인터페이스 연결 대화 상자에서 인스턴스를 선택한 다음 연결을 선택합니다.

명령줄 사용하여 인스턴스에 네트워크 인터페이스를 연결하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- attach-network-interface(AWS CLI)
- Add-EC2NetworkInterface(Windows PowerShell용 AWS 도구)

인스턴스에서 네트워크 인터페이스 분리

Amazon EC2 콘솔의 인스턴스 또는 네트워크 인터페이스 페이지를 사용하면 언제든지 EC2 인스턴스에 연결된 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 인스턴스를 선택합니다.
3. 작업, 네트워킹, 네트워크 인터페이스 분리를 선택합니다.
4. 네트워크 인터페이스 분리 대화 상자에서 네트워크 인터페이스를 선택하고 분리를 선택합니다.

Amazon EC2 콘솔을 사용하면 Elastic Load Balancing 로드 밸런서, Lambda 함수, WorkSpace 또는 NAT 게이트웨이와 같은 다른 서비스에서 리소스에 연결된 네트워크 인터페이스를 분리할 수 없습니다. 리소스가 삭제되면 해당 리소스의 네트워크 인터페이스가 삭제됩니다.

네트워크 인터페이스 페이지를 사용하여 인스턴스에서 네트워크 인터페이스를 분리하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 네트워크 인터페이스를 선택합니다.
3. 네트워크 인터페이스를 선택하고 설명을 확인하여 네트워크 인터페이스가 다른 유형의 리소스가 아닌 인스턴스에 연결되어 있는지 확인합니다. 리소스가 EC2 인스턴스인 경우 분리를 선택합니다.

   네트워크 인터페이스가 인스턴스의 기본 네트워크 인터페이스인 경우 분리 버튼이 비활성화됩니다.
4. 확인 메시지가 나타나면 예, 분리를 선택합니다.
5. 네트워크 인터페이스가 인스턴스에서 분리되지 않으면 강제 분리를 선택하고 다시 시도합니다. 이 옵션은 최후의 수단으로만 사용하는 것이 좋습니다. 강제 분리하면 인스턴스를 다시 시작할 때까지 동일한 인덱스에 다른 네트워크 인터페이스가 연결되지 않을 수 있습니다. 또한 인스턴스를 다시 시작할 때까지 네트워크 인터페이스가 분리되었음을 인스턴스 메타데이터에 반영되지 않게 할 수 있습니다.

명령줄을 사용하여 네트워크 인터페이스를 분리하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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 네트워크 인터페이스의 서브넷과 동일한 VPC를 지정해야 합니다.

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 네트워크 인터페이스를 선택합니다.
3. 네트워크 인터페이스를 선택하고 작업, 보안 그룹 변경을 선택합니다.
4. 보안 그룹 변경 대화 상자에서 사용할 보안 그룹을 선택하고 저장을 선택합니다.

명령줄을 사용하여 네트워크 인터페이스의 보안 그룹을 변경하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- modify-network-interface-attribute(AWS CLI)
- Edit-EC2NetworkInterfaceAttribute(Windows PowerShell용 AWS 도구)

원본 또는 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`를 설정해야 합니다. 기본 값은 `enabled`입니다.

콘솔을 사용하여 네트워크 인터페이스에 대한 원본/대상 확인을 변경하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 네트워크 인터페이스를 선택합니다.
3. 네트워크 인터페이스를 선택하고 작업, 소스/대상 확인을 선택합니다.
4. 대화 상자에서 활성화(활성화된 경우) 또는 비활성화(비활성화된 경우)를 선택하고 저장을 선택합니다.

명령줄을 사용하여 네트워크 인터페이스에 대한 원본/대상 확인을 변경하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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 엘라스틱 IP 주소를 연결하려면

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

2. 탐색 창에서 네트워크 인터페이스를 선택합니다.
3. 네트워크 인터페이스를 선택하고 작업, 주소 연결을 선택합니다.
4. 탄력적 IP 주소 연결 대화 상자의 주소 목록에서 탄력적 IP 주소를 선택합니다.
5. 프라이빗 IP 주소에 연결에서 탄력적 IP 주소와 연결할 프라이빗 IPv4 주소를 선택합니다.
6. 재연결 허용을 선택하여 탄력적 IP 주소가 현재 다른 인스턴스나 네트워크 인터페이스와 연결되어 있는 경우 지정된 네트워크 인터페이스와 연결될 수 있도록 한 다음 주소 연결을 선택합니다.

명령줄을 사용하여 탄력적 IP 주소를 연결하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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 탄력적 IP 주소를 연결하는 방법은 이 방법뿐입니다.

Amazon EC2 콘솔이나 명령줄을 사용하여 탄력적 IP 주소를 분리할 수 있습니다.

콘솔을 사용하여 엘라스틱 IP 주소 연결을 끊으려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 네트워크 인터페이스를 선택합니다.
3. 네트워크 인터페이스를 선택하고 작업, 주소 연결 해제를 선택합니다.
4. IP 주소 연결 해제 대화 상자에서 예, 연결 해제를 선택합니다.

명령줄을 사용하여 탄력적 IP 주소 연결을 끊으려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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 네트워크 인터페이스를 선택한 후 네트워크 인터페이스를 선택합니다.
3. 작업, IP 주소 관리를 선택합니다.
4. IPv6 주소에서 새 IP 할당을 선택합니다. 서브넷의 주소 범위에 속한 IPv6 주소를 지정합니다. AWS가 자동으로 주소를 선택하도록 하려면 자동 할당 값을 그대로 둡니다.

5. 예, 업데이트를 선택합니다.

명령줄을 사용하여 네트워크 인터페이스에 IPv6 주소를 할당하려면

- 다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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 네트워크 인터페이스를 선택한 후 네트워크 인터페이스를 선택합니다.
3. 작업, IP 주소 관리를 선택합니다.
4. IPv6 주소에서 제거할 IPv6 주소에 대해 할당 해제를 선택합니다.
5. 예, 업데이트를 선택합니다.

명령줄을 사용하여 네트워크 인터페이스에서 IPv6 주소 할당을 해제하려면

- 다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.
  - unassign-ipv6-addresses(AWS CLI)
  - Unregister-EC2Ipv6AddressList(Windows PowerShell용 AWS 도구).

종료 동작 변경

인스턴스에 연결된 네트워크 인터페이스의 종료 동작을 설정할 수 있습니다. 연결된 인스턴스를 종료할 때 네트워크 인터페이스를 자동으로 삭제할지를 지정할 수 있습니다.

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 네트워크 인터페이스를 선택합니다.
3. 네트워크 인터페이스를 선택하고 작업, 종료 방식 변경을 선택합니다.
4. 인스턴스를 종료할 때 네트워크 인터페이스를 삭제하려면 종료 방식 변경 대화 상자에서 종료 시 삭제 확인란을 선택합니다.

명령줄을 사용하여 네트워크 인터페이스에 대한 종료 동작을 변경하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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 네트워크 인터페이스를 선택합니다.
3. 네트워크 인터페이스를 선택하고 작업, 설명 변경을 선택합니다.
4. 설명 변경 대화 상자에서 네트워크 인터페이스에 대한 설명을 입력하고 저장을 선택합니다.

명령줄을 사용하여 네트워크 인터페이스에 대한 설명을 변경하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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 EC2리소스에 태그 지정 (p. 1068) 단원을 참조하십시오.

콘솔을 사용하여 네트워크 인터페이스에 대한 태그를 추가하거나 편집하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 네트워크 인터페이스를 선택합니다.
3. 네트워크 인터페이스를 선택합니다.
4. 세부 정보 창에서 태그, 태그 추가/편집을 선택합니다.
5. 태그 추가/편집 대화 상자에서 생성할 각 태그에 대해 태그 생성을 선택하고 키와 값(선택 사항)을 입력합니다. 완료되면 저장을 선택합니다.

명령줄을 사용하여 네트워크 인터페이스에 대한 태그를 추가 또는 편집하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- create-tags(AWS CLI)
- New-EC2Tag(Windows PowerShell용 AWS 도구)

요청자 관리 네트워크 인터페이스

요청자 관리 네트워크 인터페이스는 AWS 서비스가 사용자의 VPC에 생성하는 네트워크 인터페이스입니다. 이 네트워크 인터페이스는 Amazon RDS 인스턴스와 같은 다른 서비스의 인스턴스를 나타낼 수도 있고, 사용자가 AWS PrivateLink 서비스, Amazon ECS 작업 등 다른 서비스 또는 리소스에 액세스하도록 해줄 수도 있습니다.

요청자 관리 네트워크 인터페이스는 수정 또는 분리할 수 없습니다. 네트워크 인터페이스가 나타내는 리소스를 삭제할 경우 AWS 서비스가 자동으로 네트워크 인터페이스를 분리하고 삭제합니다. 요청자 관리 네트워

크 인터페이스의 보안 그룹을 변경하려면 해당 서비스에 대한 콘솔 또는 명령줄 도구를 사용해야 할 수 있습니다. 자세한 내용은 서비스별 설명서를 참조하십시오.

요청자 관리 네트워크 인터페이스에 태그를 지정할 수 있습니다. 자세한 내용은 태그 추가 또는 편집 (p. 705) 단원을 참조하십시오.

본인의 계정에 포함된 요청자 관리 네트워크 인터페이스를 볼 수 있습니다.

콘솔을 사용하여 요청자 관리 네트워크 인터페이스를 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 네트워크 인터페이스를 선택합니다.
3. 네트워크 인터페이스를 선택하고 세부 정보 창에서 다음 정보를 확인합니다.
   - 연결 소유자: 사용자가 네트워크 인터페이스를 생성한 경우 이 필드는 사용자의 AWS 계정 ID를 표시합니다. 그렇지 않은 경우, 네트워크 인터페이스를 생성한 보안 주체 또는 서비스의 별칭 또는 ID가 여기에 표시됩니다.
   - 설명: 네트워크 인터페이스의 용도에 관한 정보를 제공합니다. 예: "VPC 엔드포인트 인터페이스".

명령줄을 사용하여 요청자 관리 네트워크 인터페이스를 보려면

1. describe-network-interfaces AWS CLI 명령을 사용하여 계정의 네트워크 인터페이스를 설명합니다.

```
aws ec2 describe-network-interfaces
```

2. 다른 AWS 서비스가 네트워크 인터페이스를 관리하는 경우, 출력에서 `RequesterManaged` 필드가 `true`로 표시됩니다.

```
{
            "Status": "in-use",
           ...
            "Description": "VPC Endpoint Interface vpce-089f2123488812123",
            "NetworkInterfaceId": "eni-c8fbc27e",
            "VpcId": "vpc-1a2b3c4d",
            "PrivateIpAddresses": [
                {
                    "PrivateDnsName": "ip-10-0-2-227.ec2.internal",
                    "Primary": true,
                    "PrivateIpAddress": "10.0.2.227"
                }
            ],
            "RequesterManaged": true,
           ...
        }
```

또는 Get-EC2NetworkInterface Windows PowerShell용 도구 명령을 사용합니다.

Linux에서 향상된 네트워킹

향상된 네트워킹에서는 지원되는 인스턴스 유형 (p. 707)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 따르는 추가 요금은 없습니다.

내용

- 향상된 네트워킹 유형 (p. 707)
- 인스턴스에서 향상된 네트워킹 기능 활성화 (p. 707)
- Linux 인스턴스에서 ENA(Elastic Network Adapter)를 사용하여 향상된 네트워킹 활성화 (p. 707)
- Linux 인스턴스에서 intel 82599 VF 인터페이스를 사용하여 향상된 네트워킹 활성화 (p. 719)
- ENA(Elastic Network Adapter) 문제 해결 (p. 725)

향상된 네트워킹 유형

인스턴스 유형에 따라 다음 중 한 가지 메커니즘을 사용하여 향상된 네트워킹을 활성화할 수 있습니다.

ENA(Elastic Network Adapter)

탄력적 네트워크 어댑터(ENA)는 지원되는 인스턴스 유형에 대해 최대 100Gbps의 네트워크 속도를 지원합니다.

A1, C5, C5d, C5n, F1, G3, G4, H1, I3, I3en, Inf1, `m4.16xlarge`, M5, M5a, M5ad, M5d, M5dn, M5n, P2, P3, R4, R5, R5a, R5ad, R5d, R5dn, R5n, T3, T3a, `u-6tb1.metal`, `u-9tb1.metal`, `u-12tb1.metal`, `u-18tb1.metal`, `u-24tb1.metal`, X1, X1e, and z1d 인스턴스는 향상된 네트워킹을 위해 Elastic Network Adapter를 사용합니다.

intel 82599 Virtual Function(VF) 인터페이스

intel 82599 Virtual Function 인터페이스는 지원되는 인스턴스 유형에 대해 최대 10Gbps의 네트워크 속도를 지원합니다.

C3, C4, D2, I2, M4(`m4.16xlarge` 제외) 및 R3 인스턴스는 향상된 네트워킹을 위해 intel 82599 VF 인터페이스를 사용합니다.

각 인스턴스 유형에 대해 지원되는 네트워크 속도에 대한 자세한 내용은 Amazon EC2 인스턴스 유형을 참조하십시오.

인스턴스에서 향상된 네트워킹 기능 활성화

인스턴스 유형에서 향상된 네트워킹을 위해 ENA를 지원하는 경우 Linux 인스턴스에서 ENA(Elastic Network Adapter)를 사용하여 향상된 네트워킹 활성화 (p. 707)의 절차를 따르십시오.

인스턴스 유형에서 향상된 네트워킹을 위해 intel 82599 VF 인터페이스를 지원하는 경우 Linux 인스턴스에서 intel 82599 VF 인터페이스를 사용하여 향상된 네트워킹 활성화 (p. 719)의 절차를 따르십시오.

Linux 인스턴스에서 ENA(Elastic Network Adapter)를 사용하여 향상된 네트워킹 활성화

Amazon EC2는 ENA(Elastic Network Adapter)를 통해 향상된 네트워킹을 제공합니다. 향상된 네트워킹을 사용하려면 필수 ENA 모듈을 설치하고 ENA 지원을 활성화해야 합니다.

다음 AMI에는 필수 ENA 모듈이 포함되어 있으며 ENA 지원이 활성화되어 있습니다.

- Amazon Linux 2
- Amazon Linux AMI 2018.03
- Ubuntu 14.04(`linux-aws` 커널) 이상
- Red Hat Enterprise Linux 7.4 이상

- SUSE Linux Enterprise Server 12 SP2 이상
- CentOS 7.4.1708 이상
- FreeBSD 11.1 이상
- Debian GNU/Linux 9 이상

내용

- 요구 사항 (p. 708)
- 향상된 네트워킹 기능 활성화 여부 확인 (p. 708)
- Amazon Linux AMI에서 향상된 네트워킹 기능 활성화 (p. 710)
- Ubuntu에서 향상된 네트워킹 사용 (p. 711)
- Linux에서 향상된 네트워킹 활성화 (p. 712)
- DKMS를 사용하여 Ubuntu에서 향상된 네트워킹 사용 (p. 714)
- 문제 해결 (p. 716)
- 운영 체제 최적화 (p. 716)

요구 사항

ENA를 사용하여 향상된 네트워킹을 준비하려면 인스턴스를 다음과 같이 설정하십시오.

- 다음 지원되는 인스턴스 유형을 선택합니다. A1, C5, C5d, C5n, F1, G3, G4, H1, I3, I3en, Inf1, `m4.16xlarge`, M5, M5a, M5ad, M5d, M5dn, M5n, P2, P3, R4, R5, R5a, R5ad, R5d, R5dn, R5n, T3, T3a, `u-6tb1.metal`, `u-9tb1.metal`, `u-12tb1.metal`, `u-18tb1.metal`, `u-24tb1.metal`, X1, X1e, and z1d.
- 지원되는 Linux 커널 버전과 지원 배포판을 사용하여 인스턴스를 시작합니다. 그러면 인스턴스에 대해 향상된 ENA 네트워킹이 자동으로 활성화됩니다. 자세한 내용은 ENA Linux Kernel Driver 릴리스 정보를 참조하십시오.
- 인스턴스가 인터넷에 연결되어 있는지 확인합니다.
- AWS CLI 또는 Windows PowerShell용 AWS 도구를 자신이 선택한 컴퓨터에 설치하고 구성합니다(로컬 데스크톱/노트북 권장). 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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 이 경우에도 데이터를 보존할 수 있습니다.

향상된 네트워킹 기능 활성화 여부 확인

향상된 네트워킹 기능이 활성화되었는지를 알아보려면 인스턴스에 `ena` 모듈 가 설치되어 있는지, 그리고 `enaSupport` 속성이 설정되어 있는지를 확인해야 합니다. 인스턴스가 이러한 두 조건을 충족하는 경우 ethtool -i eth*n* 명령을 사용하면 해당 모듈이 네트워크 인터페이스에 사용 중이라고 표시됩니다.

커널 모듈(ena)

`ena` 모듈이 설치되어 있는지 확인하려면 다음 예와 같이 modinfo 명령을 사용합니다.

```
[ec2-user ~]$ modinfo ena
filename:       /lib/modules/4.14.33-59.37.amzn2.x86_64/kernel/drivers/amazon/net/ena/
ena.ko
version:        1.5.0g
license:        GPL
```

```
description:    Elastic Network Adapter (ENA)
author:         Amazon.com, Inc. or its affiliates
srcversion:     692C7C68B8A9001CB3F31D0
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:
retpoline:      Y
intree:         Y
name:           ena
...
```

위 Amazon Linux 사례에는 `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 향상된 네트워킹 사용 (p. 711) 단원을 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-ids instance_id --query
 "Reservations[].Instances[].EnaSupport"
```

- Get-EC2InstanceWindows 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 명령과 확인하고자 하는 인터페이스 이름을 사용하여 해당 인터페이스에서 `ena` 모듈이 사용되고 있는지를 확인할 수 있습니다. 단일 인터페이스를 사용하는 경우(기본 설정), `eth0`으로 표시됩니다. 운영 체제가 예측 가능한 네트워크 이름 (p. 713)을 지원하는 경우 이는 `ens5`와 같은 이름일 수 있습니다.

다음 예시에서는 표시된 드라이버가 `vif`로, `ena` 모듈이 로드되지 않았음을 알 수 있습니다.

```
[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
```

이 예제의 경우, ena 모듈이 이미 설치되었고 최소 버전 요건을 충족하는 것을 알 수 있습니다. 이 인스턴스는 향상된 네트워킹이 올바르게 구성된 상태입니다.

```
[ec2-user ~]$ ethtool -i eth0
driver: ena
version: 1.5.0g
firmware-version:
expansion-rom-version:
bus-info: 0000:00:05.0
supports-statistics: yes
supports-test: no
supports-eeprom-access: no
supports-register-dump: no
supports-priv-flags: no
```

Amazon Linux AMI에서 향상된 네트워킹 기능 활성화

Amazon Linux 2 및 최신 버전의 Amazon Linux AMI에는 ENA가 설치된 향상된 네트워킹에 필요한 모듈이 포함되어 있으며 ENA 지원이 활성화되어 있습니다. 따라서 지원되는 인스턴스 유형에서 Amazon Linux의 HVM 버전을 사용하여 인스턴스를 시작하면, 확장 네트워크 기능이 이미 해당 인스턴스에서 활성화된 상태입니다. 자세한 내용은 향상된 네트워킹 기능 활성화 여부 확인 (p. 708) 단원을 참조하십시오.

구형 Amazon Linux AMI를 사용하여 인스턴스를 시작했는데 아직 확장 네트워크 기능이 활성화되어 있지 않다면 다음 절차에 따라 확장 네트워크를 활성화할 수 있습니다.

Amazon Linux AMI에서 향상된 네트워킹을 활성화하려면

1. 인스턴스에 연결합니다.
2. 인스턴스 상에서 다음 명령을 사용하여 인스턴스를 ena를 포함한 최신 커널과 커널 모듈로 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 향상된 네트워킹 기능 활성화 여부 확인 (p. 708)에서 modinfo ena 명령을 사용하여 ena 모듈이 설치되어 있고 최소 권장 버전 요건을 충족하는지를 확인합니다.
5. [EBS 지원 인스턴스] 로컬 컴퓨터에서 Amazon EC2 콘솔 또는 stop-instances(AWS CLI), Stop-EC2Instance(Windows PowerShell용 AWS 도구) 명령 중 하나를 사용하여 인스턴스를 중지하십시오. 인스턴스를 AWS OpsWorks에서 관리할 경우 AWS OpsWorks 콘솔에서 인스턴스를 중지해야 인스턴스 상태가 동기화됩니다.

   [인스턴스 스토어 지원 인스턴스] 속성을 수정하기 위해 인스턴스를 중지할 수 없습니다. 그 대신 이 절차(Amazon Linux AMI에서 향상된 네트워킹을 활성화하려면(인스턴스 스토어 지원 인스턴스) (p. 711))로 넘어가십시오.

6. 사용자의 로컬 컴퓨터에서 다음 명령 중 하나를 사용하여 확장 네트워크 속성을 활성화합니다.

   - 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. 112) 의 설명에 따라 인스턴스에서 AMI를 생성합니다. 생성된 AMI는 인스턴스의 향상된 네트워킹 `enaSupport` 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 향상된 네트워킹 기능 활성화 여부 확인 (p. 708)에서 ethtool -i eth*n* 명령을 사용하여 `ena` 모듈이 설치되어 있고 최소 권장 버전 요건을 충족하는지를 확인합니다.

   향상된 네트워킹을 활성화한 이후에 인스턴스에 연결할 수 없는 경우 ENA(Elastic Network Adapter) 문제 해결 (p. 725) 단원을 참조하십시오.

Amazon Linux AMI에서 향상된 네트워킹을 활성화하려면(인스턴스 스토어 지원 인스턴스)

이전 절차에서 인스턴스를 중지한 단계까지 진행합니다. 인스턴스 스토어 기반 Linux AMI 생성 (p. 115)에 설명된 것처럼 새 AMI를 생성하고, AMI를 등록할 때 향상된 네트워킹 속성을 활성화합니다.

- register-image(AWS CLI)

```
aws ec2 register-image --ena-support ...
```

- Register-EC2Image(Windows PowerShell용 AWS 도구)

```
Register-EC2Image -EnaSupport $true ...
```

Ubuntu에서 향상된 네트워킹 사용

최신 Ubuntu HVM AMI에는 ENA가 설치된 향상된 네트워킹에 필요한 모듈이 포함되어 있으며 ENA 지원이 활성화되어 있습니다. 따라서 지원되는 인스턴스 유형에서 최신 Ubuntu HVM AMI를 사용하여 인스턴스를 시작하면, 확장 네트워크 기능이 이미 해당 인스턴스에서 활성화된 상태입니다. 자세한 내용은 향상된 네트워킹 기능 활성화 여부 확인 (p. 708) 단원을 참조하십시오.

이전의 AMI를 사용하여 인스턴스를 시작했고 확장 네트워킹 기능이 활성화되어 있지 않은 경우에는 `linux-aws` 커널 패키지를 설치하여 최신 확장 네트워킹 드라이버를 가져오고 필요한 속성을 업데이트할 수 있습니다.

linux-aws 커널 패키지(Ubuntu 16.04 이상 버전)를 설치하려면

Ubuntu 16.04와 18.04는 Ubuntu 사용자 지정 커널(linux-aws 커널 패키지)과 함께 제공됩니다. 다른 커널을 사용하려면 AWS Support에 문의하십시오.

linux-aws 커널 패키지(Ubuntu Trusty 14.04)를 설치하려면

1. 인스턴스에 연결합니다.

2. 패키지 캐시와 패키지를 업데이트합니다.

```
ubuntu:~$ sudo apt-get update && sudo apt-get upgrade -y linux-aws
```

   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. [EBS 지원 인스턴스] 로컬 컴퓨터에서 Amazon EC2 콘솔 또는 stop-instances(AWS CLI), Stop-EC2Instance(Windows PowerShell용 AWS 도구) 명령 중 하나를 사용하여 인스턴스를 중지하십시오. 인스턴스를 AWS OpsWorks에서 관리할 경우 AWS OpsWorks 콘솔에서 인스턴스를 중지해야 인스턴스 상태가 동기화됩니다.

   [인스턴스 스토어 지원 인스턴스] 속성을 수정하기 위해 인스턴스를 중지할 수 없습니다. 그 대신 이 절차(Ubuntu에서 확장 네트워킹 기능을 사용하려면(인스턴스 스토어 지원 인스턴스) (p. 712))로 넘어가십시오.

4. 사용자의 로컬 컴퓨터에서 다음 명령 중 하나를 사용하여 확장 네트워크 속성을 활성화합니다.

   - 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
```

5. (선택 사항) Amazon EBS 지원 Linux AMI 생성 (p. 112) 의 설명에 따라 인스턴스에서 AMI를 생성합니다. 생성된 AMI는 인스턴스의 향상된 네트워킹 `enaSupport` 속성을 상속합니다. 따라서 이 AMI를 사용하여 기본적으로 향상된 네트워킹 기능이 활성화된 상태로 다른 인스턴스를 시작할 수 있습니다.
6. 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.

Ubuntu에서 확장 네트워킹 기능을 사용하려면(인스턴스 스토어 지원 인스턴스)

이전 절차에서 인스턴스를 중지한 단계까지 진행합니다. 인스턴스 스토어 기반 Linux AMI 생성 (p. 115)에 설명된 것처럼 새 AMI를 생성하고, AMI를 등록할 때 향상된 네트워킹 속성을 활성화합니다.

- register-image(AWS CLI)

```
aws ec2 register-image --ena-support ...
```

- Register-EC2Image(Windows PowerShell용 AWS 도구)

```
Register-EC2Image -EnaSupport $true ...
```

Linux에서 향상된 네트워킹 활성화

Red Hat Enterprise Linux, SUSE Linux Enterprise Server 및 CentOS용 최신 AMI에는 ENA가 포함된 향상된 네트워킹에 필요한 모듈이 포함되어 있으며 ENA 지원이 활성화되어 있습니다. 따라서 지원되는 인스턴스 유형에 최신 AMI를 사용하여 인스턴스를 시작하면 향상된 네트워킹이 이미 해당 인스턴스에서 활성화된 상태입니다. 자세한 내용은 향상된 네트워킹 기능 활성화 여부 확인 (p. 708) 단원을 참조하십시오.

다음 절차는 Amazon Linux AMI 또는 Ubuntu를 제외한 다른 Linux 배포판에서 향상된 네트워킹을 활성화하는 일반적인 방법입니다. 명령 구문, 파일 위치 또는 패키지와 도구 지원을 비롯한 자세한 내용은 해당 Linux 배포판 설명서를 참조하십시오.

Linux에서 향상된 네트워킹을 활성화하려면

1. 인스턴스에 연결합니다.
2. https://github.com/amzn/amzn-drivers의 GitHub에서 인스턴스의 `ena` 모듈에 대한 소스 코드를 복제합니다. (SUSE SLES 12 SP2 이상에는 기본적으로 ENA 2.02가 포함되어 있으므로 ENA 드라이버를 다운로드하고 컴파일할 필요가 없습니다. SLES 12 SP2 이상 버전에서는 원하는 드라이버 버전을 재고 커널에 추가하라는 요청을 제출해야 합니다.)

```
git clone https://github.com/amzn/amzn-drivers
```

3. 인스턴스에 `ena` 모듈을 컴파일하고 설치합니다. 이러한 단계는 Linux 배포판에 따라 달라집니다. Red Hat Enterprise Linux에서 모듈을 컴파일하는 방법에 대한 자세한 내용은 AWS 지식 센터 문서를 참조하십시오.
4. 모듈 종속성을 갱신하려면 sudo depmod 명령을 실행합니다.
5. 인스턴스에서 `initramfs`를 업데이트하여 부팅 시 새 모듈이 로드되도록 합니다. 예를 들어 배포에서 dracut을 지원하는 경우 다음 명령을 사용할 수 있습니다.

```
dracut -f -v
```

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.

```
rpm -qa | grep -e '^systemd-[0-9]\+\|^udev-[0-9]\+'
systemd-208-11.el7_0.2.x86_64
```

   위의 Red Hat Enterprise Linux 7 예제에서, systemd 버전은 208이므로, 해당 네트워크 인터페이스 이름을 비활성화해야 합니다.

   b. `net.ifnames=0`의 `GRUB_CMDLINE_LINUX` 줄에 `/etc/default/grub` 옵션을 추가하여 예측 가능한 네트워크 인터페이스 이름을 비활성화합니다.

```
sudo sed -i '/^GRUB\_CMDLINE\_LINUX/s/\"$/\ net\.ifnames\=0\"/' /etc/default/grub
```

   c. GRUB 구성 파일을 재구축합니다.

```
sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

7. [EBS 지원 인스턴스] 로컬 컴퓨터에서 Amazon EC2 콘솔 또는 stop-instances(AWS CLI), Stop-EC2Instance(Windows PowerShell용 AWS 도구) 명령 중 하나를 사용하여 인스턴스를 중지하십시오. 인스턴스를 AWS OpsWorks에서 관리할 경우 AWS OpsWorks 콘솔에서 인스턴스를 중지해야 인스턴스 상태가 동기화됩니다.

   [인스턴스 스토어 지원 인스턴스] 속성을 수정하기 위해 인스턴스를 중지할 수 없습니다. 그 대신 이 절차(Linux에서 향상된 네트워킹을 활성화하려면(인스턴스 스토어 –지원 인스턴스) (p. 714))로 넘어가십시오.
8. 사용자의 로컬 컴퓨터에서 다음 명령 중 하나를 사용하여 확장 네트워크 `enaSupport` 속성을 활성화합니다.

- 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. 112) 의 설명에 따라 인스턴스에서 AMI를 생성합니다. 생성된 AMI는 인스턴스의 향상된 네트워킹 `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 있는 경우 AMI 생성 전에 이 파일을 삭제해야 합니다. 이 파일에 원본 인스턴스의 이더넷 어댑터에 대한 MAC 주소가 포함되어 있습니다. 이 파일로 다른 인스턴스가 부팅되면 운영 체제에서 디바이스를 찾지 못하고, `eth0`이 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 네트워킹을 활성화한 이후에 인스턴스에 연결할 수 없는 경우 ENA(Elastic Network Adapter) 문제 해결 (p. 725) 단원을 참조하십시오.

Linux에서 향상된 네트워킹을 활성화하려면(인스턴스 스토어 –지원 인스턴스)

이전 절차에서 인스턴스를 중지한 단계까지 진행합니다. 인스턴스 스토어 기반 Linux AMI 생성 (p. 115)에 설명된 것처럼 새 AMI를 생성하고, AMI를 등록할 때 향상된 네트워킹 속성을 활성화합니다.

- register-image(AWS CLI)

```
aws ec2 register-image --ena-support ...
```

- Register-EC2Image(Windows PowerShell용 AWS 도구)

```
Register-EC2Image -EnaSupport ...
```

DKMS를 사용하여 Ubuntu에서 향상된 네트워킹 사용

이 방법은 테스트 및 피드백 목적으로만 사용됩니다. 프로덕션 배포에 사용하기 위한 것이 아닙니다. 프로덕션 배포는 Ubuntu에서 향상된 네트워킹 사용 (p. 711)를 참조하십시오.

Important

DKMS를 사용하면 구독에 대한 지원 계약이 무효화됩니다. kmod 구성을 사용하면 사용 가능한 최신 커널 모듈을 실행할 수 있습니다.

Ubuntu에서 ENA를 사용하여 향상된 네트워킹 기능을 활성화하려면(EBS 지원 인스턴스)

1. Ubuntu에서 향상된 네트워킹 사용 (p. 711)의 1과 2단계를 따르십시오.

2. 커널 모듈을 컴파일하도록 `build-essential` 패키지를 설치하고 커널을 업데이트할 때마다 `dkms` 모듈이 다시 빌드되도록 `ena` 패키지를 설치합니다.

```
ubuntu:~$ sudo apt-get install -y build-essential dkms
```

3. https://github.com/amzn/amzn-drivers의 GitHub에서 인스턴스의 `ena` 모듈에 대한 소스를 복제합니다.

```
ubuntu:~$ git clone https://github.com/amzn/amzn-drivers
```

4. `amzn-drivers` 패키지를 `/usr/src/` 디렉터리로 이동하여 DKMS에서 커널이 업데이트될 때마다 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
```

5. `ena` 버전을 대체하여 다음 값을 사용하여 DKMS 구성 파일을 생성합니다.

   파일을 생성합니다.

```
ubuntu:~$ sudo touch /usr/src/amzn-drivers-1.0.0/dkms.conf
```

   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"
```

6. DKMS를 사용하여 인스턴스에 `ena` 모듈을 추가 및 빌드하고 설치합니다.

   모듈을 DKMS에 추가합니다.

```
ubuntu:~$ sudo dkms add -m amzn-drivers -v 1.0.0
```

   dkms 명령을 사용하여 모듈을 빌드합니다.

```
ubuntu:~$ sudo dkms build -m amzn-drivers -v 1.0.0
```

   dkms를 사용하여 모듈을 설치합니다.

```
ubuntu:~$ sudo dkms install -m amzn-drivers -v 1.0.0
```

7. 부팅 시 올바른 모듈이 로드되도록 `initramfs`를 다시 빌드합니다.

```
ubuntu:~$ sudo update-initramfs -u -k all
```

8. 향상된 네트워킹 기능 활성화 여부 확인 (p. 708)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)
```

9. Ubuntu에서 향상된 네트워킹 사용 (p. 711)의 3단계를 계속하십시오.

문제 해결

ENA 어댑터 문제 해결에 대한 자세한 내용은 ENA(Elastic Network Adapter) 문제 해결 (p. 725) 단원을 참조하십시오.

운영 체제 최적화

향상된 네트워킹을 지원하는 인스턴스에서 네트워크 성능을 최대화하려는 경우 기본 운영 체제 구성을 수정해야 할 수도 있습니다. 높은 수준의 네트워크 성능을 필요로 하는 애플리케이션에는 다음 구성 변경을 권장합니다.

이러한 운영 체제 최적화 외에도, 네트워크 트래픽의 최대 전송 단위(MTU)를 고려하여 워크로드 및 네트워크 아키텍처에 맞게 조정해야 합니다. 자세한 내용은 EC2 인스턴스에 대한 네트워크 MTU(최대 전송 단위) (p. 766) 단원을 참조하십시오.

AWS는 클러스터 배치 그룹에서 시작된 인스턴스 간의 평균 왕복 지연 시간(50us)과 99.9 백분위 테일 지연 시간(200us)을 정기적으로 측정합니다. 애플리케이션에서 일관되게 낮은 지연 시간을 요구하는 경우 고정 성능 Nitro 기반 인스턴스에 최신 버전의 ENA 드라이버를 사용하는 것이 좋습니다.

이러한 절차는 Amazon Linux 2 및 Amazon Linux AMI용으로 작성되었습니다. 하지만 커널 버전이 3.9 이상인 다른 Linux 배포판에서도 작동할 수 있습니다. 자세한 내용은 해당 시스템 관련 설명서를 참조하십시오.

향상된 네트워킹을 지원하도록 Amazon Linux 인스턴스를 최적화하는 방법

1. 인스턴스의 클록 소스를 확인합니다.

```
cat /sys/devices/system/clocksource/clocksource0/current_clocksource
```

2. 클록 소스가 `xen`이면 다음 하위 단계를 완료합니다. 그렇지 않으면 Step 3 (p. 717)로 건너뜁니다.

   a. GRUB 구성을 편집하여 `xen_nopvspin=1` 및 `clocksource=tsc`를 커널 부트 옵션에 추가합니다.

- Amazon Linux 2의 경우 아래와 같이 `/etc/default/grub` 파일을 편집하고 이러한 옵션을 `GRUB_CMDLINE_LINUX_DEFAULT` 줄에 추가합니다.

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,115200n8 net.ifnames=0
 biosdevname=0 nvme_core.io_timeout=4294967295 xen_nopvspin=1 clocksource=tsc"
GRUB_TIMEOUT=0
```

- Amazon Linux AMI의 경우 아래와 같이 `/boot/grub/grub.conf` 파일을 편집하고 이러한 옵션을 `kernel` 줄에 추가합니다.

```
kernel /boot/vmlinuz-4.14.62-65.117.amzn1.x86_64 root=LABEL=/ console=tty1
 console=ttyS0 selinux=0 nvme_core.io_timeout=4294967295 xen_nopvspin=1
 clocksource=tsc
```

b. (Amazon Linux 2에만 해당) 이러한 변경 사항이 적용되도록 GRUB 구성 파일을 다시 빌드합니다.

```
sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

3. 해당 인스턴스 유형이 EC2 인스턴스에 대한 프로세서 상태 제어 (p. 537)에서 지원되는 유형으로 나열되는 경우 지연 시간이 짧은 시스템 성능을 유지하려면 시스템에서 C 상태가 심화되지 않도록 해야 합니다. 자세한 내용은 C 상태 심화 제한을 통한 고성능 및 저 지연 시간 (p. 539) 단원을 참조하십시오.

a. GRUB 구성을 편집하여 `intel_idle.max_cstate=1`을 커널 부트 옵션에 추가합니다.

- Amazon Linux 2의 경우 아래와 같이 `/etc/default/grub` 파일을 편집하고 이 옵션을 `GRUB_CMDLINE_LINUX_DEFAULT` 줄에 추가합니다.

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,115200n8
 net.ifnames=0 biosdevname=0 nvme_core.io_timeout=4294967295 xen_nopvspin=1
 clocksource=tsc intel_idle.max_cstate=1"
GRUB_TIMEOUT=0
```

- Amazon Linux AMI의 경우 아래와 같이 `/boot/grub/grub.conf` 파일을 편집하고 이 옵션을 `kernel` 줄에 추가합니다.

```
kernel /boot/vmlinuz-4.14.62-65.117.amzn1.x86_64 root=LABEL=/ console=tty1
 console=ttyS0 selinux=0 nvme_core.io_timeout=4294967295 xen_nopvspin=1
 clocksource=tsc intel_idle.max_cstate=1
```

b. (Amazon Linux 2에만 해당) 이러한 변경 사항이 적용되도록 GRUB 구성 파일을 다시 빌드합니다.

```
sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

4. 예약된 커널 메모리가 충분하여 패킷 버퍼 할당의 높은 속도를 유지할 수 있는지 확인합니다(기본값은 너무 작을 수 있음).

a. 선택한 편집기에서 (`root`로 또는 sudo를 사용하여) `/etc/sysctl.conf` 파일을 엽니다.

b. 해당 인스턴스 유형에 맞는 예약된 커널 메모리 값(KB)을 사용하여 이 파일에 `vm.min_free_kbytes` 줄을 추가합니다. 기본적으로 이 값을 가용 시스템 메모리의 1-3%로 설정하고 애플리케이션 요건에 맞춰 이 값을 높거나 낮게 조정해야 합니다.

```
vm.min_free_kbytes = 1048576
```

c. 다음 명령을 사용하여 이 구성을 적용합니다.

```
sudo sysctl -p
```

d. 다음 명령을 사용하여 설정이 적용되었는지 확인합니다.

```
sudo sysctl -a 2>&1 | grep min_free_kbytes
```

5. 인스턴스를 재부팅하여 새 구성을 로드합니다.

```
sudo reboot
```

6. (선택 사항) 모두가 동일 NUMA 노드에 속하는 다양한 CPU와 연결되도록 패킷 수신 인터럽트를 수동으로 분배합니다. 하지만 irqbalancer가 전역적으로 비활성화되어 있으므로 이 작업을 수행할 때 각별히 주의해야 합니다.

   Note

   이 단계의 구성 변경 사항은 재부팅 후에는 유지되지 않습니다.

   a. `smp_affinity.sh`라는 파일을 생성한 후 그 안에 다음 코드 블록을 붙여 넣습니다.

```
#/bin/sh
service irqbalance stop
affinity_values=(00000001 00000002 00000004 00000008 00000010 00000020 00000040
 00000080)
irqs=($(grep eth /proc/interrupts|awk '{print $1}'|cut -d : -f 1))
irqLen=${#irqs[@]}
for (( i=0; i<${irqLen}; i++ ));
do
  echo $(printf "0000,00000000,00000000,00000000,${affinity_values[$i]}") > /proc/
irq/${irqs[$i]}/smp_affinity;
  echo "IRQ ${irqs[$i]} =" $(cat /proc/irq/${irqs[$i]}/smp_affinity);
done
```

   b. 다음 명령으로 스크립트를 실행합니다.

```
sudo bash ./smp_affinity.sh
```

7. (선택 사항) 핸들에서 IRQ를 받는 vCPU가 과부하 상태인 경우 또는 애플리케이션 네트워크 처리가 CPU에 의존하는 경우 수신 패킷 스티어링(RPS)을 이용하여 네트워크 처리 부분을 다른 코어로 분담시킬 수 있습니다. NUMA 간에 노드가 잠기지 않도록 RPS에 사용되는 코어가 동일 NUMA 노드에 속해 있는지 확인합니다. 예를 들면 패킷 처리 시 코어 8-15개를 사용하려면 다음 명령을 사용합니다.

   Note

   이 단계의 구성 변경 사항은 재부팅 후에는 유지되지 않습니다.

```
for i in `seq 0 7`; do echo $(printf "0000,00000000,00000000,00000000,0000ff00") | sudo
 tee /sys/class/net/eth0/queues/rx-$i/rps_cpus; done
```

8. (선택 사항) 가능하면 동일 NUMA 노드에서 모든 처리를 유지합니다.

   a. numactl을 설치합니다.

```
sudo yum install -y numactl
```

   b. 네트워크 처리 프로그램 실행 시 이 프로그램을 단일 NUMA 노드에 바인딩합니다. 예를 들면 다음 명령은 셸 스크립트인 `run.sh`를 NUMA 노드 0으로 바인딩합니다.

```
numactl --cpunodebind=0 --membind=0 run.sh
```

   c. 하이퍼스레딩이 활성화된 경우 CPU 코어당 단일 하드웨어 스레드만 사용하도록 애플리케이션을 구성할 수 있습니다.

- lscpu 명령을 사용하여 NUMA 노드로 매핑되는 CPU 코어를 볼 수 있습니다.

```
lscpu | grep NUMA
```

결과:

```
NUMA node(s):      2
NUMA node0 CPU(s): 0-15,32-47
NUMA node1 CPU(s): 16-31,48-63
```

- 다음 명령을 사용하여 물리적 CPU에 속하는 하드웨어 스레드를 볼 수 있습니다.

```
cat /sys/devices/system/cpu/cpu0/topology/thread_siblings_list
```

결과:

```
0,32
```

이 예에서는 스레드 0과 32가 CPU 0으로 매핑됩니다.

- 스레드 32-47(실제로는 0-15와 동일한 CPU의 하드웨어 스레드)에서 실행되지 않도록 하려면 다음 명령을 사용합니다.

```
numactl --physcpubind=+0-15 --membind=0 ./run.sh
```

9. 다양한 트래픽 클래스에 대해 여러 탄력적 네트워크 인터페이스를 사용합니다. 예를 들어 백엔드 데이터베이스를 사용하는 웹 서버를 실행하려는 경우 탄력적 네트워크 인터페이스 하나를 웹 서버 프런트 엔드로 사용하고, 다른 하나를 데이터베이스 연결용으로 사용합니다.

Linux 인스턴스에서 intel 82599 VF 인터페이스를 사용하여 향상된 네트워킹 활성화

Amazon EC2에서는 Intel `ixgbevf` 드라이버를 사용하는 Intel 82599 VF 인터페이스를 통해 향상된 네트워킹 기능을 제공합니다.

내용

- 요구 사항 (p. 719)
- 향상된 네트워킹 기능 활성화 여부 확인 (p. 720)
- Amazon Linux에서 향상된 네트워킹 기능 사용 (p. 721)
- Ubuntu에서 향상된 네트워킹 사용 (p. 722)
- 다른 Linux 배포판에서 향상된 네트워킹 기능 사용 (p. 723)
- 연결 문제 해결 (p. 725)

요구 사항

intel 82599 VF 인터페이스를 사용하여 향상된 네트워킹을 준비하려면 인스턴스를 다음과 같이 설정하십시오.

- C3, C4, D2, I2, M4(m4.16xlarge 제외) 및 R3의 지원되는 인스턴스 유형에서 선택합니다.
- 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 현재 Amazon Linux HVM AMI를 사용하여 Amazon EBS– 지원, 확장 네트워크를 지원하는 인스턴스를 시작하면, 인스턴스 설정에서 확장 네트워크 기능이 처음부터 –활성화되어 있습니다.

> **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.

- 인스턴스가 인터넷에 연결되어 있는지 확인합니다.
- AWS CLI 또는 Windows PowerShell용 AWS 도구를 자신이 선택한 컴퓨터에 설치하고 구성합니다(로컬 데스크톱/노트북 권장). 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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 이 경우에도 데이터를 보존할 수 있습니다.

향상된 네트워킹 기능 활성화 여부 확인

`ixgbevf`모듈 가 인스턴스에 설치되어 있고 `sriovNetSupport` 속성이 설정된 경우 intel 82599 VF 인터페이스를 사용하는 향상된 네트워킹이 활성화됩니다.

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 않으면 `SriovNetSupport`는 비어 있습니다. 속성이 설정된 경우 다음 예제 출력과 같이 값이 간단합니다.

```
"SriovNetSupport": {
    "Value": "simple"
},
```

이미지 속성(sriovNetSupport)

다음 명령 중 하나를 사용하여 AMI에 향상된 네트워킹 `sriovNetSupport` 속성 세트가 있는지 확인할 수 있습니다.

- describe-images(AWS CLI)

```
aws ec2 describe-images --image-id ami_id --query "Images[].SriovNetSupport"
```

- Get-EC2Image (Windows PowerShell용 AWS 도구)

```
(Get-EC2Image -ImageId ami-id).SriovNetSupport
```

속성이 설정되지 않으면 `SriovNetSupport`는 비어 있습니다. 속성이 설정된 경우 값은 간단합니다.

네트워크 인터페이스 드라이버

다음 명령을 사용하여 특정 인터페이스에서 모듈이 사용되고 있는지 확인할 수 있습니다. 여기서 확인하고자 하는 인터페이스 이름을 대체합니다. 단일 인터페이스를 사용하는 경우(기본 설정), `eth0`으로 표시됩니다. 운영 체제가 예측 가능한 네트워크 이름 (p. 723)을 지원하는 경우 이는 `ens5`와 같은 이름일 수 있습니다.

다음 예시에서는 표시된 드라이버가 `ixgbevf`이므로, `vif` 모듈이 로드되지 않았습니다.

```
[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
```

이 예제에서는 `ixgbevf` 모듈이 로드됩니다 이 인스턴스는 향상된 네트워킹이 올바르게 구성된 상태입니다.

```
[ec2-user ~]$ ethtool -i eth0
driver: ixgbevf
version: 4.0.3
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
```

Amazon Linux에서 향상된 네트워킹 기능 사용

최신 Amazon Linux HVM AMI에는 향상된 네트워킹에 요구되는 `ixgbevf` 모듈이 설치되어 있으며 필요한 `sriovNetSupport` 속성 세트를 갖추고 있습니다. 따라서 현재 Amazon Linux HVM AMI를 사용하여 인스턴스 유형을 시작하는 경우 인스턴스에 대해 확장 네트워크 기능이 이미 활성화되어 있습니다. 자세한 내용은 향상된 네트워킹 기능 활성화 여부 확인 (p. 720) 단원을 참조하십시오.

지난 세대의 Amazon Linux AMI를 사용하여 인스턴스를 시작했고 확장 네트워크 기능이 활성화되어 있지 않은 경우에는 다음 절차에 따라 확장 네트워크를 활성화할 수 있습니다.

> **Warning**
>
> 향상된 네트워킹 속성을 활성화한 다음에는 다시 비활성화할 수 없습니다.

향상된 네트워킹을 활성화하려면

1. 인스턴스 연결 후.
2. 인스턴스 상에서 다음 명령을 사용하여 인스턴스를 `ixgbevf`를 포함한 최신 커널과 커널 모듈로 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 향상된 네트워킹 기능 활성화 여부 확인 (p. 720)에서 modinfo ixgbevf 명령을 사용하여 `ixgbevf` 모듈이 설치되어 있고 최소 권장 버전 요건을 충족하는지를 확인합니다.

5. [EBS 지원 인스턴스] 로컬 컴퓨터에서 Amazon EC2 콘솔 또는 stop-instances (AWS CLI), Stop-EC2Instance (Windows PowerShell용 AWS 도구) 명령 중 하나를 사용하여 인스턴스를 중지하십시오. 인스턴스를 AWS OpsWorks에서 관리할 경우 AWS OpsWorks 콘솔에서 인스턴스를 중지해야 인스턴스 상태가 동기화됩니다.

   [인스턴스 스토어 지원 인스턴스] 속성을 수정하기 위해 인스턴스를 중지할 수 없습니다. 그 대신 이 절차(향상된 네트워킹 기능을 활성화하려면(인스턴스 스토어 지원 인스턴스) (p. 722))로 넘어가십시오.
6. 사용자의 로컬 컴퓨터에서 다음 명령 중 하나를 사용하여 확장 네트워크 속성을 활성화합니다.

   - 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. 112) 의 설명에 따라 인스턴스에서 AMI를 생성합니다. 생성된 AMI는 인스턴스의 확장 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 향상된 네트워킹 기능 활성화 여부 확인 (p. 720)에서 ethtool -i eth*n* 명령을 사용하여 `ixgbevf` 모듈이 설치되어 있고 최소 권장 버전 요건을 충족하는지를 확인합니다.

향상된 네트워킹 기능을 활성화하려면(인스턴스 스토어 지원 인스턴스)

이전 절차에서 인스턴스를 중지한 단계까지 진행합니다. 인스턴스 스토어 기반 Linux AMI 생성 (p. 115)에 설명된 것처럼 새 AMI를 생성하고, AMI를 등록할 때 향상된 네트워킹 속성을 활성화합니다.

- register-image(AWS CLI)

```
aws ec2 register-image --sriov-net-support simple ...
```

- Register-EC2Image(Windows PowerShell용 AWS 도구)

```
Register-EC2Image -SriovNetSupport "simple" ...
```

Ubuntu에서 향상된 네트워킹 사용

시작하기 전에 인스턴스에서 향상된 네트워킹이 이미 활성화되어 있는지 확인 (p. 720)합니다.

최신 빠른 시작 Ubuntu HVM AMI에는 향상된 네트워킹을 위한 필수 드라이버가 포함되어 있습니다. `ixgbevf` 버전 2.16.4 이하를 사용하는 경우 `linux-aws` 커널 패키지를 설치하여 최신 향상된 네트워킹 드라이버를 가져올 수 있습니다.

다음 절차는 Ubuntu 인스턴스에서 `ixgbevf` 모듈을 컴파일하는 일반적인 방법입니다.

linux-aws 커널 패키지를 설치하려면

1. 인스턴스에 연결합니다.
2. 패키지 캐시와 패키지를 업데이트합니다.

```
ubuntu:~$ sudo apt-get update && sudo apt-get upgrade -y linux-aws
```

> **Important**
>
> 업데이트 과정에서 `grub` 설치 메시지가 표시되는 경우, `/dev/xvda`를 사용하여 `grub`을 설치하고 `/boot/grub/menu.lst`의 현재 버전을 유지하도록 선택합니다.

다른 Linux 배포판에서 향상된 네트워킹 기능 사용

시작하기 전에 인스턴스에서 향상된 네트워킹이 이미 활성화되어 있는지 확인 (p. 720)합니다. 최신 빠른 시작 HVM AMI에는 향상된 네트워킹을 위한 필수 드라이버가 포함되어 있으므로 추가 단계를 수행할 필요가 없습니다.

다음 절차는 Amazon Linux 또는 Ubuntu를 제외한 다른 Linux 배포판에서 intel 82599 VF 인터페이스를 사용하여 향상된 네트워킹을 활성화해야 하는 경우에 사용되는 일반적인 방법입니다. 명령 구문과 파일 위치, 패키지 및 도구 지원을 비롯한 자세한 내용은 사용 Linux 배포판의 전용 문서를 참조하십시오.

Linux에서 향상된 네트워킹을 활성화하려면

1. 인스턴스에 연결합니다.
2. Sourceforge에서 인스턴스 `ixgbevf` 모듈의 소스 다운로드: https://sourceforge.net/projects/e1000/files/ixgbevf%20stable/.

   2.14.2 버전을 포함하여 2.16.4 이전의 `ixgbevf` 버전은 Ubuntu 일부 버전을 포함한 일부 Linux 배포판에서 제대로 빌드되지 않습니다.
3. 인스턴스에 `ixgbevf` 모듈을 컴파일하고 설치합니다.

   > **Warning**
   >
   > 현재 사용 중인 커널을 기준으로 `ixgbevf` 모듈을 컴파일한 다음 새 커널에 맞게 드라이버를 재구축하지 않고 커널 업그레이드를 진행하면, 다음번 재부팅에서 시스템이 배포 버전의 `ixgbevf` 모듈로 돌아갈 수 있습니다. 이 경우 배포 버전이 향상된 네트워킹과 호환되지 않으면 시스템에 접속하지 못하는 결과가 발생할 수 있습니다.
4. 모듈 종속성을 갱신하려면 sudo depmod 명령을 실행합니다.
5. 인스턴스에서 `initramfs`를 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 Enterprise Linux 7 예제에서, systemd 버전은 208이므로, 해당 네트워크 인터페이스 이름을 비활성화해야 합니다.

   b. `net.ifnames=0`의 `GRUB_CMDLINE_LINUX` 줄에 `/etc/default/grub` 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. [EBS 지원 인스턴스] 로컬 컴퓨터에서 Amazon EC2 콘솔 또는 stop-instances(AWS CLI), Stop-EC2Instance(Windows PowerShell용 AWS 도구) 명령 중 하나를 사용하여 인스턴스를 중지하십시오. 인스턴스를 AWS OpsWorks에서 관리할 경우 AWS OpsWorks 콘솔에서 인스턴스를 중지해야 인스턴스 상태가 동기화됩니다.

   [인스턴스 스토어 지원 인스턴스] 속성을 수정하기 위해 인스턴스를 중지할 수 없습니다. 그 대신 이 절차(향상된 네트워킹 기능 사용(인스턴스 스토어 지원 인스턴스)– (p. 724))로 넘어가십시오.

8. 사용자의 로컬 컴퓨터에서 다음 명령 중 하나를 사용하여 확장 네트워크 속성을 활성화합니다.

   - 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"
   ```

9. (선택 사항) Amazon EBS 지원 Linux AMI 생성 (p. 112) 의 설명에 따라 인스턴스에서 AMI를 생성합니다. 생성된 AMI는 인스턴스의 확장 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 있는 경우 AMI 생성 전에 이 파일을 삭제해야 합니다. 이 파일에 원본 인스턴스의 이더넷 어댑터에 대한 MAC 주소가 포함되어 있습니다. 이 파일로 다른 인스턴스가 부팅되면 운영 체제에서 디바이스를 찾지 못하고, `eth0`이 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)–

이전 절차에서 인스턴스를 중지한 단계까지 진행합니다. 인스턴스 스토어 기반 Linux AMI 생성 (p. 115)에 설명된 것처럼 새 AMI를 생성하고, AMI를 등록할 때 향상된 네트워킹 속성을 활성화합니다.

- register-image(AWS CLI)

```
aws ec2 register-image --sriov-net-support simple ...
```

- Register-EC2Image(Windows PowerShell용 AWS 도구)

```
Register-EC2Image -SriovNetSupport "simple" ...
```

연결 문제 해결

향상된 네트워킹 기능을 활성화하는 도중 연결이 해제된 경우, 커널이 `ixgbevf` 모듈과 호환되지 않아 발생한 문제일 수 있습니다. 사용 중인 Linux 배포판과 함께 제공되는 `ixgbevf` 모듈 버전을 설치하여 인스턴스에 사용해 보십시오.

PV 또는 AMI 인스턴스에서 향상된 네트워킹 기능을 활성화하면 인스턴스 접속이 불가능해질 수 있습니다.

자세한 내용은 EC2 인스턴스에서 향상된 네트워킹을 활성화하고 구성하려면 어떻게 해야 합니까?를 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. 725) 섹션에서 시작하십시오.

인스턴스에 연결할 수 있는 경우 이 주제의 이후 섹션에서 다루는 장애 탐지 및 복구 메커니즘을 사용하여 진단 정보를 수집할 수 있습니다.

내용

- 연결 문제 해결 (p. 725)
- 연결 유지 메커니즘 (p. 726)
- 레지스터 읽기 시간 초과 (p. 727)
- 통계 (p. 727)
- syslog의 드라이버 오류 로그 (p. 730)

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. 726) 단계로 넘어갑니다.

2. 로컬 컴퓨터에서 다음 명령을 사용하여 확장 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 `ena`의 단계에 따라 현재 커널 버전으로 Linux 인스턴스에서 ENA(Elastic Network Adapter)를 사용하여 향상된 네트워킹 활성화 (p. 707) 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. 115)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 이러한 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 구성을 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_complete`을 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_ctx_err: N`

대기열 *N*에 대해 `ena_com_prepare_tx`가 실패한 횟수입니다. 이 값은 항상 0이어야 합니다. 그렇지 않은 경우 드라이버 로그를 참조하십시오.

`queue_N_tx_missing_tx_comp: codeN`

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_id`입니다. 유효한 `req_id`는 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_refil_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 `sysfs`에 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:1e: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.

호스트 속성 설정이 지원되지 않음

호스트 속성은 이 디바이스에 대해 지원되지 않습니다.

rx 대기열에 대해 버퍼를 할당하지 못함

복구할 수 있는 오류이며 오류가 발생된 시점에 메모리 부족 문제가 발생했을 수 있습니다.

기능 *x*가 지원되지 않음

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.
- **27**: ENA(Elastic Network Adapter) 드라이버가 snmpd로부터의 이더넷 기능 폴링을 지원하지 않습니다.

AENQ를 구성하지 못함

ENA가 AENQ 구성을 지원하지 않습니다.

지원되지 않는 AENQ 이벤트를 설정하려고 시도

이 오류는 ENA에서 지원되지 않는 AENQ 이벤트 그룹을 설정하려고 시도했음을 나타냅니다.

Elastic Fabric Adapter(EFA)

Elastic Fabric Adapter(EFA)(EFA)는 네트워크 디바이스로 Amazon EC2 인스턴스에 연결하여 HPC(고성능 컴퓨팅) 및 기계 학습 애플리케이션 속도를 높일 수 있습니다. EFA는 AWS 클라우드가 제공하는 확장성, 유연성, 탄력성을 통해 온프레미스 HPC 클러스터의 애플리케이션 성능을 확보합니다.

EFA는 전통적으로 클라우드 기반 HPC 시스템에서 사용하는 TCP 전송보다 지연율이 낮고 일정하며 더 높은 처리량을 제공합니다. 또한 대규모 HPC 및 기계 학습 애플리케이션에서 중요한 인스턴스 간 통신 성능을 확장합니다. 이는 기존 AWS 네트워크 인프라에서 작업하도록 최적화되어 애플리케이션 요구량에 따라 크기를 변경합니다.

EFA는 Libfabric 1.9.0과 통합되며 HPC 애플리케이션을 위한 Open MPI 4.0.2 및 Intel MPI 2019 업데이트 6과 기계 학습 애플리케이션을 위한 NCCL(Nvidia Collective Communications Library)을 지원합니다.

Note

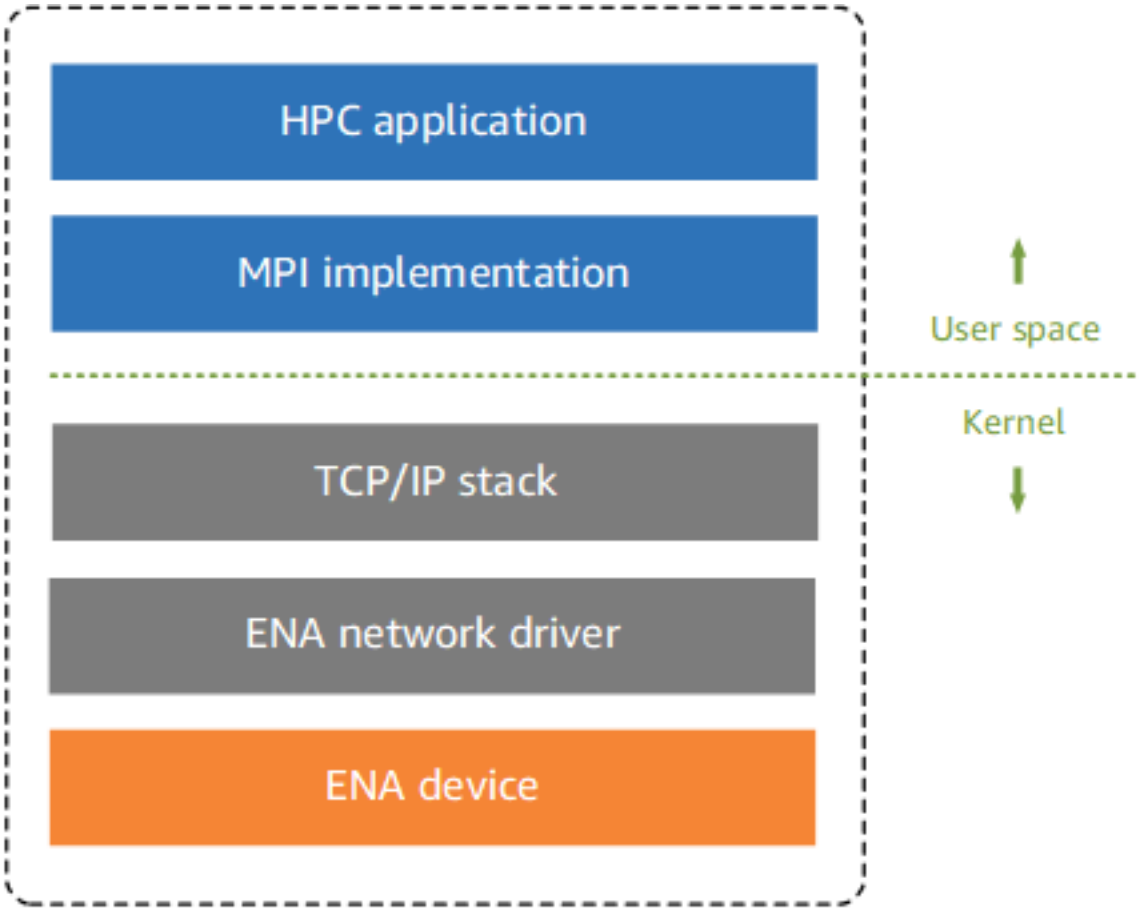
Windows 인스턴스에서는 EFAs에서 제공하는 OS 우회 기능을 지원하지 않습니다. EFA를 Windows 인스턴스에 연결한 경우 인스턴스는 추가적인 EFA 기능이 없는 ENA(Elastic Network Adapter)로 작동합니다.

목차

- EFA 기본 사항 (p. 731)
- 지원되는 인터페이스 및 라이브러리 (p. 733)
- 지원되는 인스턴스 유형 (p. 733)
- 지원되는 AMI (p. 733)
- EFA 제한 사항 (p. 733)
- EFA 및 MPI 시작하기 (p. 733)
- EFA 및 NCCL 시작하기 (p. 740)
- EFA로 작업 (p. 754)
- EFA 모니터링 (p. 757)

EFA 기본 사항

EFA는 추가 기능이 있는 ENA(Elastic Network Adapter)입니다. 따라서 추가적인 OS 우회 기능을 포함한 모든 ENA의 기능을 제공합니다. OS 우회는 HPC 및 기계 학습 애플리케이션이 네트워크 인터페이스 하드웨어와 직접 통신하도록 하는 액세스 모델로서 낮은 지연율과 신뢰성 높은 전송 기능을 제공합니다.



Traditional HPC software stack in EC2

기존의 HPC 애플리케이션은 시스템의 네트워크 전송 인터페이스에 MPI(Message Passing Interface)를 사용했습니다. AWS 클라우드에서 MPI를 사용하는 애플리케이션 인터페이스를 의미하며 이는 인스턴스 간 네트워크 통신을 위해 운영 체제의 TCP/IP 스택과 ENA 디바이스 드라이버를 사용한다는 의미입니다.

EFA에서 HPC 애플리케이션은 Libfabric API에 MPI 또는 NCCL 인터페이스를 사용합니다. Libfabric API는 운영 체제 커널을 우회하여 EFA 디바이스와 직접 통신을 통해 네트워크에 패킷을 전송합니다. 이는 오버헤드를 줄이고 HPC 애플리케이션이 더욱 효율적으로 실행되도록 합니다.

> **Note**
>
> Libfabric은 OFI(OpenFabrics Interface) 프레임워크의 핵심 구성 요소로서 OFI의 사용자 공간 API를 정의하고 내보냅니다. 자세한 정보는 Libfabric OpenFabrics 웹 사이트를 참조하십시오.

EFAs 및 ENA 간의 차이점

ENA(Elastic Network Adapter)는 VPC 네트워크를 지원하는 데 필요한 기존 IP 네트워크 기능을 제공합니다. EFAs는 ENA와 마찬가지로 모든 기존 IP 네트워크 기능을 제공하지만 OS 우회 기능을 함께 제공합니다. OS 우회는 HPC 및 기계 학습 애플리케이션이 운영 체제 커널을 우회하여 EFA 디바이스와 직접 통신할 수 있도록 합니다.

지원되는 인터페이스 및 라이브러리

EFA는 다음 인터페이스 및 라이브러리를 지원합니다.

- Open MPI 4.0.2
- Intel MPI 2019 업데이트 6
- NCCL(Nvidia Collective Communications Library) 2.4.2 이상

지원되는 인스턴스 유형

다음 인스턴스 유형은 EFAs를 지원합니다. `c5n.18xlarge`, `c5n.metal`, `i3en.24xlarge`, `i3en.metal`, `inf1.24xlarge`, `m5dn.24xlarge`, `m5n.24xlarge`, `r5dn.24xlarge`, `r5n.24xlarge`, and `p3dn.24xlarge`.

지원되는 AMI

다음 AMI는 EFAs를 지원합니다. Amazon Linux, Amazon Linux 2, RHEL 7.6, RHEL 7.7, CentOS 7, Ubuntu 16.04, and Ubuntu 18.04.

EFA 제한 사항

EFA에는 다음과 같은 제한 사항이 있습니다.

- 인스턴스당 한 개의 EFA를 연결할 수 있습니다.
- EFA OS 우회 트래픽은 단일 서브넷으로 제한됩니다. 즉 EFA 트래픽은 서브넷 간 전송이 불가능합니다. EFA의 일반 IP 트래픽은 서브넷 간 전송이 가능합니다.
- EFA OS 우회 트래픽은 라우팅되지 않습니다. EFA의 일반 IP 트래픽은 라우팅이 가능합니다.
- EFA은 보안 그룹 자체 내의 모든 인바운드 및 아웃바운드 트래픽을 허용하는 보안 그룹에 구성되어야 합니다.

EFA 및 MPI 시작하기

이 자습서는 HPC 워크로드를 위한 MPI 지원 인스턴스와 EFA를 시작하는 데 도움이 됩니다. 이 자습서에서는 다음 단계를 수행합니다.

목차

- 1단계: EFA를 사용한 보안 그룹을 준비합니다. (p. 734)
- 2단계: 임시 인스턴스 실행 (p. 734)
- 3단계: EFA 소프트웨어 설치 (p. 735)
- 4단계: Ptrace 보호 비활성화 (p. 736)
- 5단계: (선택 사항) Intel MPI 설치 (p. 736)
- 6단계: HPC 애플리케이션 설치 (p. 737)
- 7단계: EFA 지원 AMI 생성 (p. 738)
- 8단계: 클러스터 배치 그룹에 EFA 지원 인스턴스 시작 (p. 738)
- 9단계: 임시 인스턴스 종료 (p. 739)
- 10단계: 암호 없는 SSH 사용 (p. 739)

1단계: EFA를 사용한 보안 그룹을 준비합니다.

EFA는 보안 그룹 자체 내의 모든 인바운드 및 아웃바운드 트래픽을 허용하는 보안 그룹이 필요합니다.

EFA를 사용한 보안 그룹을 생성합니다.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 보안 그룹을 선택한 다음, 보안 그룹 생성을 선택합니다.
3. 보안 그룹 생성 창에서 다음을 수행하십시오.
   a. 보안 그룹 이름의 경우 `EFA-enabled security group`과 같은 보안 그룹의 고유한 이름을 입력합니다.
   b. (선택 사항) 설명에 보안 그룹에 대한 간략한 설명을 입력합니다.
   c. VPC에서는 EFA를 사용한 인스턴스를 시작하려는 VPC를 입력합니다.
   d. 생성을 선택합니다.
4. 생성한 보안 그룹을 선택하고 설명 탭에서 그룹 ID를 복사합니다.
5. 인바운드 및 아웃바운드 탭에서 다음을 수행합니다.
   a. 편집을 선택합니다.
   b. 유형에서 모든 트래픽을 선택합니다.
   c. 소스에서 사용자 지정을 선택합니다.
   d. 복사한 보안 그룹 ID를 필드에 붙여넣습니다.
   e. Save를 선택합니다.

2단계: 임시 인스턴스 실행

EFA 소프트웨어 구성 요소를 설치하고 구성하는 데 사용할 수 있는 임시 인스턴스를 실행합니다. 이 인스턴스를 사용해 EFA를 사용한 AMI을 생성하여 EFA를 사용한 인스턴스를 실행할 수 있습니다.

임시 인스턴스를 실행합니다.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 인스턴스 시작을 선택합니다.
3. AMI 선택 페이지에서 지원되는 다음 AMIs 중 하나에 대해 선택을 선택합니다. Amazon Linux, Amazon Linux 2, RHEL 7.6, RHEL 7.7, CentOS 7, Ubuntu 16.04, and Ubuntu 18.04.
4. 인스턴스 유형 선택 페이지에서 지원되는 다음 인스턴스 유형 중 하나를 선택하고 Next: Configure Instance Details(다음: 인스턴스 세부 정보 구성)를 선택합니다. `c5n.18xlarge`, `c5n.metal`, `i3en.24xlarge`, `i3en.metal`, `inf1.24xlarge`, `m5dn.24xlarge`, `m5n.24xlarge`, `r5dn.24xlarge`, `r5n.24xlarge`, and `p3dn.24xlarge`.
5. [Configure Instance Details] 페이지에서 다음을 수행합니다.
   a. Elastic Fabric Adapter(EFA)에서 활성화를 선택합니다.
   b. 네트워크 인터페이스 항목의 디바이스 eth0에서 새 네트워크 인터페이스를 선택합니다.
   c. Next: Add Storage(다음: 스토리지 추가)를 선택합니다.
6. 스토리지 추가 페이지에서 인스턴스에 연결할 볼륨과 AMI에서 지정한 볼륨(예: 루트 디바이스 볼륨)을 지정합니다. 다음: 태그 추가를 선택합니다.
7. 태그 추가 페이지에서 임시 인스턴스를 식별할 태그를 지정하고 다음: 보안 그룹 구성을 선택합니다.
8. 보안 그룹 구성 페이지의 보안 그룹 할당에서 기존 보안 그룹 선택을 선택한 다음, 1단계에서 생성한 보안 그룹을 선택합니다.
9. 인스턴스 시작 검토 페이지에서 설정을 검토한 후 시작을 선택하여 키 페어를 선택하고 인스턴스를 시작합니다.

3단계: EFA 소프트웨어 설치

EFA를 사용하는 커널, EFA 드라이버, Libfabric 및 임시 인스턴스에서 EFA를 지원하는 데 필요한 OPEN MPI 스택을 설치합니다.

이 단계는 EFA를 Open MPI에서 사용할지 또는 Intel MPI에서 사용할지에 따라 다릅니다.

EFA 소프트웨어를 설치하려면

1. 2단계에서 시작한 인스턴스에 연결합니다. 자세한 내용은 Linux 인스턴스에 연결 (p. 485)를 참조하십시오.
2. 모든 소프트웨어 패키지가 최신 상태로 업데이트되어 있는지 확인하기 위해, 인스턴스에서 퀵 소프트웨어 업데이트를 실행합니다. 이 프로세스는 몇 분 정도 걸릴 수 있습니다.

   - Amazon Linux, Amazon Linux 2, RHEL 7.6/7.7, CentOS 7

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

   - Ubuntu 16.04 및 Ubuntu 18.04

     ```
     $ sudo apt-get update -y
     ```

     ```
     $ sudo apt-get upgrade -y
     ```

3. EFA 소프트웨어 설치 파일을 다운로드합니다. 최신 안정 버전을 다운로드하려면 다음 명령을 사용하십시오.

   ```
   $ curl -O https://s3.us-west-2.amazonaws.com/aws-efa-installer/aws-efa-installer-1.8.3.tar.gz
   ```

   또한 위의 명령에서 버전 번호를 `latest`로 바꾸면 최신 버전을 얻을 수 있습니다.
4. 소프트웨어 설치 파일은 압축된 `.tar.gz` 파일 형태로 되어 있습니다. 압축 `.tar.gz` 파일에서 파일을 추출하고 추출된 디렉터리로 이동하십시오.

   ```
   $ tar -xf aws-efa-installer-1.8.3.tar.gz
   ```

   ```
   $ cd aws-efa-installer
   ```

5. EFA 소프트웨어를 설치합니다.

   - EFA를 Open MPI에서 사용하려는 경우 Libfabric 및 Open MPI와 함께 EFA 소프트웨어를 설치해야 하며 4단계: Intel MPI 설치를 건너뛰어야 합니다.

     Libfabric 및 Open MPI와 함께 EFA 소프트웨어를 설치하려면 다음 명령을 실행합니다.

     ```
     $ sudo ./efa_installer.sh -y
     ```

     Libfabric은 `/opt/amazon/efa` 디렉터리에 설치되는 반면, Open MPI는 `/opt/amazon/openmpi` 디렉터리에 설치됩니다.
   - Intel MPI에서만 EFA를 사용하려는 경우 Libfabric 및 Open MPI 없이 EFA 소프트웨어를 설치할 수 있습니다. 이 경우 Intel MPI는 임베디드 Libfabric을 사용합니다. 이렇게 하려면 4단계: Intel MPI 설치를 완료해야 합니다.

     Libfabric 및 Open MPI없이 EFA 소프트웨어를 설치하려면 다음 명령을 실행합니다.

```
$ sudo ./efa_installer.sh -y --minimal
```

6. 인스턴스에서 로그아웃한 후 다시 로그인합니다.
7. EFA 소프트웨어 구성 요소가 성공적으로 설치되었는지 확인합니다.

```
$ fi_info -p efa
```

이 명령은 Libfabric EFA 인스턴스에 대한 정보를 반환합니다. 다음은 해당 명령 출력을 보여주는 예제입니다.

```
provider: efa
    fabric: EFA-fe80::94:3dff:fe89:1b70
    domain: efa_0-rdm
    version: 2.0
    type: FI_EP_RDM
    protocol: FI_PROTO_EFA
provider: efa
    fabric: EFA-fe80::94:3dff:fe89:1b70
    domain: efa_0-dgrm
    version: 2.0
    type: FI_EP_DGRAM
    protocol: FI_PROTO_EFA
provider: efa;ofi_rxd
    fabric: EFA-fe80::94:3dff:fe89:1b70
    domain: efa_0-dgrm
    version: 1.0
    type: FI_EP_RDM
    protocol: FI_PROTO_RXD
```

4단계: Ptrace 보호 비활성화

HPC 애플리케이션의 성능을 향상시키기 위해 Libfabric에서는 프로세스가 동일한 인스턴스에서 실행 중일 때 프로세스 간 통신에 인스턴스의 로컬 메모리를 사용합니다.

공유 메모리 기능은 ptrace 보호에서 지원되지 않는 CMA(Cross Memory Attach)를 사용합니다. Ubuntu와 같이 기본적으로 ptrace 보호가 활성화되어 있는 Linux 배포 버전을 사용하는 경우 이를 비활성화해야 합니다. Linux 배포 버전에 ptrace 보호가 기본적으로 활성화되어 있지 않으면 이 단계를 건너뜁니다.

ptrace 보호를 비활성화하려면

다음 중 하나를 수행하십시오.

- 테스트 목적으로 ptrace 보호를 일시적으로 비활성화하려면 다음 명령을 실행하십시오.

```
$ sudo sysctl -w kernel.yama.ptrace_scope=0
```

- ptrace 보호를 영구적으로 비활성화하려면 `kernel.yama.ptrace_scope = 0`을 `/etc/sysctl.d/10-ptrace.conf`에 추가하고 인스턴스를 재부팅합니다.

5단계: (선택 사항) Intel MPI 설치

> **Important**
>
> Open MPI를 사용하려면 이 단계를 건너뜁니다. Intel MPI를 사용하려는 경우에만 이 단계를 수행하십시오.

Intel MPI에는 추가 설치 및 환경 변수 구성이 필요합니다.

사전 조건

다음 단계를 수행하는 사용자가 sudo 권한을 가지고 있는지 확인하십시오.

Intel MPI를 설치하려면

1. Intel MPI 설치 파일을 다운로드하려면 Intel Developer Zone 웹 사이트를 참조하십시오.

   설치 파일을 다운로드하려면 먼저 등록해야 합니다. 등록한 후 다음을 수행합니다.

   a. 제품에서 Intel MPI Library for Linux를 선택합니다.
   b. 버전에서 2019 업데이트 6을 선택한 다음 전체 제품을 선택합니다.
2. 설치 파일은 압축된 `.tar.gz` 파일로 패키징되어 있습니다. 압축 `.tar.gz` 파일에서 파일을 추출하고 추출된 디렉터리로 이동하십시오.

   ```
   $ tar -xf file_name.tgz
   ```

   ```
   $ cd directory_name
   ```
3. 원하는 텍스트 편집기에서 `silent.cfg` 파일을 엽니다. 10번 줄에서 `ACCEPT_EULA=decline`을 `ACCEPT_EULA=accept`로 변경합니다. 변경 내용을 저장하고 파일을 닫습니다.
4. 설치 스크립트를 실행합니다.

   ```
   $ sudo ./install.sh -s silent.cfg
   ```

   Intel MPI는 `/opt/intel/impi/` 디렉터리에 기본적으로 설치됩니다.
5. 해당 셸 시작 스크립트에 Intel MPI 환경 변수를 추가하여 인스턴스가 시작될 때마다 설정되도록 하십시오. 셸에 따라 다음 중 하나를 수행합니다.

   - bash의 경우, `/home/username/.bashrc` 및 `/home/username/.bash_profile`에 다음 환경 변수를 추가하십시오.

     ```
     source /opt/intel/compilers_and_libraries/linux/mpi/intel64/bin/mpivars.sh
     ```
   - csh 및 tcsh의 경우 다음 환경 변수를 `/home/username/.cshrc`에 추가하십시오.

     ```
     source /opt/intel/compilers_and_libraries/linux/mpi/intel64/bin/mpivars.csh
     ```
6. 인스턴스에서 로그아웃한 후 다시 로그인합니다.
7. 다음 명령을 실행하여 Intel MPI가 성공적으로 설치되었는지 확인하십시오.

   ```
   $ which mpicc
   ```

   반환된 경로에 `/opt/intel/` 하위 디렉터리가 포함되어 있는지 확인합니다.

   Note

   Intel MPI를 더 이상 사용하지 않으려면 셸 시작 스크립트에서 환경 변수를 제거하십시오.

6단계: HPC 애플리케이션 설치

임시 인스턴스에 HPC 애플리케이션을 설치합니다. 설치 절차는 HPC 애플리케이션에 따라 다릅니다. Linux 인스턴스에 소프트웨어를 설치하는 데 대한 자세한 정보는 Linux 인스턴스에서 소프트웨어 관리를 참조하십시오.

Note

설치 안내를 위한 HPC 애플리케이션 문서를 참조해야 할 수도 있습니다.

7단계: EFA 지원 AMI 생성

필수 소프트웨어 구성 요소를 설치한 뒤 AMI를 사용하여 EFA를 사용하는 인스턴스를 재사용하고 시작합니다.

임시 인스턴스에서 AMI를 생성하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 1단계에서 생성한 인스턴스를 선택하고 작업, 이미지, 이미지 생성을 선택합니다.
4. 이미지 생성 창에서 다음을 수행합니다.
   a. 이미지 이름에 AMI를 설명하는 이름을 입력합니다.
   b. (선택 사항) 이미지 설명에 AMI에 대한 간략한 설명을 입력합니다.
   c. 이미지 생성을 선택한 후 닫기를 선택합니다.
5. 탐색 창에서 AMI를 선택합니다.
6. 목록에서 생성한 AMI을 찾습니다. 상태가 `pending`에서 `available`으로 바뀌기를 기다렸다가 다음 단계를 수행하십시오.

8단계: 클러스터 배치 그룹에 EFA 지원 인스턴스 시작

6단계에서 생성한 EFA이 지원되는 AMI를 사용한 클러스터 배치 그룹과 1단계에서 생성한 EFA를 지원하는 보안 그룹에 EFA를 사용한 인스턴스를 시작합니다.

Note

클러스터 배치 그룹에 EFA 지원 인스턴스를 시작할 필요는 없습니다. 그러나 EFA가 지원되는 인스턴스를 클러스터 배치 그룹에서 실행하면 단일 가용 영역의 지연율이 낮은 그룹 인스턴스를 시작하기 때문에 권장합니다.

클러스터 배치 그룹에 EFA를 사용한 인스턴스 시작

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 인스턴스 시작을 선택합니다.
3. AMI 선택 페이지에서 내 AMI를 선택하고 6단계에서 생성한 AMI를 찾은 뒤 선택을 선택합니다.
4. 인스턴스 유형 선택 페이지에서 지원되는 다음 인스턴스 유형 중 하나를 선택하고 Next: Configure Instance Details(다음: 인스턴스 세부 정보 구성)를 선택합니다. `c5n.18xlarge`, `c5n.metal`, `i3en.24xlarge`, `i3en.metal`, `inf1.24xlarge`, `m5dn.24xlarge`, `m5n.24xlarge`, `r5dn.24xlarge`, `r5n.24xlarge`, and `p3dn.24xlarge`.
5. [Configure Instance Details] 페이지에서 다음을 수행합니다.
   a. 인스턴스 수에 시작할 EFA를 사용한 인스턴스 수를 입력합니다.
   b. 네트워크 및 서브넷에서 인스턴스를 시작할 VPC와 서브넷을 선택합니다.
   c. 배치 그룹에서 배치 그룹에 인스턴스를 추가를 선택합니다.
   d. 배치 그룹 이름에서 새 배치 그룹 추가를 선택하고 배치 그룹을 설명하는 이름을 입력한 뒤 배치 그룹 전략에서 클러스터를 선택합니다.
   e. EFA에서 활성화를 선택합니다.

   f. 네트워크 인터페이스 항목의 디바이스 eth0에서 새 네트워크 인터페이스를 선택합니다. 하나의 기본 IPv4 주소와 하나 이상의 보조 IPv4 주소를 입력할 수도 있습니다. 연결된 IPv6 CIDR 블록이 있는 서브넷에서 인스턴스를 시작하는 경우 기본 IPv6 주소 및 하나 이상의 보조 IPv6 주소를 입력할 수도 있습니다.
   g. Next: Add Storage(다음: 스토리지 추가)를 선택합니다.
6. 스토리지 추가 페이지에서 인스턴스를 연결할 볼륨과 AMI로 지정한 볼륨(루트 디바이스 볼륨 등)을 지정하고 다음: 태그 추가를 선택합니다.
7. 태그 추가 페이지에서 사용자에게 친숙한 이름 등의 인스턴스 태그를 지정한 후 다음: 보안 그룹 구성(Next: Configure Security Group)을 선택합니다.
8. 보안 그룹 구성 페이지의 보안 그룹 할당에서 기존 보안 그룹 선택을 선택한 다음, 1단계에서 생성한 보안 그룹을 선택합니다.
9. 검토 및 시작을 선택합니다.
10. 인스턴스 시작 검토 페이지에서 설정을 검토한 후 시작을 선택하여 키 페어를 선택하고 인스턴스를 시작합니다.

9단계: 임시 인스턴스 종료

이 단계에서는 1단계에서 시작한 임시 인스턴스가 더 이상 필요하지 않습니다. 추가 요금이 발생하지 않도록 해당 인스턴스를 종료할 수 있습니다.

임시 인스턴스 종료

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 1단계에서 생성한 임시 인스턴스를 선택하고 작업, 인스턴스 상태, 종료, 네, 종료합니다를 선택합니다.

10단계: 암호 없는 SSH 사용

클러스터의 모든 인스턴스에서 실행하도록 애플리케이션을 활성화하려면 리더 노드에서 멤버 노드로의 암호 없는 SSH 액세스를 사용해야 합니다. 리더 노드는 애플리케이션을 실행할 인스턴스입니다. 클러스터의 나머지 인스턴스는 멤버 노드입니다,

클러스터의 인스턴스 간에 암호 없는 SSH를 사용하려면

1. 클러스터의 인스턴스 1개를 리더 노드로 선택하고 연결합니다.
2. `strictHostKeyChecking`을 비활성화하고 리더 노드에서 `ForwardAgent`를 활성화합니다. 원하는 텍스트 편집기를 사용하여 `~/.ssh/config`를 열고 다음을 추가합니다.

```
Host *
    ForwardAgent yes
Host *
    StrictHostKeyChecking no
```

3. RSA 키 페어 생성:

```
$ ssh-keygen -t rsa -N "" -f /home/ubuntu/.ssh/id_rsa
```

   키 페어는 `$HOME/.ssh/` 디렉터리에 생성됩니다.
4. 리더 노드에서 프라이빗 키의 사용 권한을 변경합니다.

```
$ chmod 600 ~/.ssh/id_rsa
```

5. 원하는 텍스트 편집기를 사용하여 `~/.ssh/id_rsa.pub`를 열고 키를 추가합니다.
6. 클러스터의 각 멤버 노드에 대해 다음을 수행합니다.
   a. 인스턴스에 연결합니다.
   b. 원하는 텍스트 편집기를 사용하여 `~/.ssh/authorized_keys`를 열고 앞에서 복사한 퍼블릭 키를 추가합니다.
7. 암호 없는 SSH가 예상대로 작동하는지 테스트하려면 리더 노드에 연결하고 다음 명령을 실행합니다.

```
$ ssh member_node_private_ip
```

   키 또는 암호를 입력하라는 메시지가 표시되지 않은 상태에서 멤버 노드에 연결해야 합니다.

EFA 및 NCCL 시작하기

NCCL(Nvidia Collective Communications Library)은 단일 노드 또는 여러 노드에서 여러 GPU를 위한 표준 집합적 통신 루틴의 라이브러리입니다. NCCL은 EFA, Libfabric 및 MPI와 함께 사용하여 다양한 기계 학습 워크로드를 지원할 수 있습니다. 자세한 내용은 NCCL 웹 사이트를 참조하십시오.

Note

- EFA와 함께 사용하는 NCCL은 p3dn.24xlarge 인스턴스에서만 지원됩니다.
- NCCL 2.4.2 이상만 EFA에서 지원됩니다.

다음 자습서는 기계 학습 워크로드를 위한 NCCL 지원 인스턴스 클러스터와 EFA를 시작하는 데 도움이 됩니다.

- 기본 AMI 사용 (p. 740)
- AWS 딥 러닝 AMI 사용 (p. 750)

기본 AMI 사용

아래 단계는 다음 기본 AMI 중 하나로 시작하는 데 도움이 됩니다. Amazon Linux, Amazon Linux 2, RHEL 7.6, RHEL 7.7, CentOS 7, Ubuntu 16.04, and Ubuntu 18.04.

목차

- 1단계: EFA를 사용한 보안 그룹을 준비합니다. (p. 741)
- 2단계: 임시 인스턴스 실행 (p. 741)
- 3단계: EFA 소프트웨어 설치 (p. 742)
- 4단계: Nvidia GPU 드라이버 및 Nvidia CUDA 도구 키트 설치 (p. 743)
- 5단계: NCCL 설치 (p. 744)
- 6단계: aws-ofi-nccl 플러그인 설치 (p. 745)
- 7단계: NCCL 테스트 설치 (p. 746)
- 8단계: EFA 및 NCCL 구성 테스트 (p. 746)
- 9단계: 기계 학습 애플리케이션 설치 (p. 748)
- 10단계: EFA 및 NCCL 지원 AMI 생성 (p. 748)
- 11단계: 임시 인스턴스 종료 (p. 748)
- 12단계: 클러스터 배치 그룹에 EFA 및 NCCL 지원 인스턴스 시작 (p. 748)

- 13단계: 암호 없는 SSH 사용 (p. 749)

1단계: EFA를 사용한 보안 그룹을 준비합니다.

EFA는 보안 그룹 자체 내의 모든 인바운드 및 아웃바운드 트래픽을 허용하는 보안 그룹이 필요합니다.

EFA를 사용한 보안 그룹을 생성합니다.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 보안 그룹을 선택한 다음, 보안 그룹 생성을 선택합니다.
3. 보안 그룹 생성 창에서 다음을 수행하십시오.
   a. 보안 그룹 이름의 경우 `EFA-enabled security group`과 같은 보안 그룹의 고유한 이름을 입력합니다.
   b. (선택 사항) 설명에 보안 그룹에 대한 간략한 설명을 입력합니다.
   c. VPC에서는 EFA를 사용한 인스턴스를 시작하려는 VPC를 입력합니다.
   d. 생성을 선택합니다.
4. 생성한 보안 그룹을 선택하고 설명 탭에서 그룹 ID를 복사합니다.
5. 인바운드 및 아웃바운드 탭에서 다음을 수행합니다.
   a. 편집을 선택합니다.
   b. 유형에서 모든 트래픽을 선택합니다.
   c. 소스에서 사용자 지정을 선택합니다.
   d. 복사한 보안 그룹 ID를 필드에 붙여넣습니다.
   e. Save를 선택합니다.

2단계: 임시 인스턴스 실행

EFA 소프트웨어 구성 요소를 설치하고 구성하는 데 사용할 수 있는 임시 인스턴스를 실행합니다. 이 인스턴스를 사용해 EFA를 사용한 AMI을 생성하여 EFA를 사용한 인스턴스를 실행할 수 있습니다.

임시 인스턴스를 실행합니다.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 인스턴스 시작을 선택합니다.
3. AMI 선택 페이지에서 다음 AMI 중 하나를 선택합니다. Amazon Linux, Amazon Linux 2, RHEL 7.6, RHEL 7.7, CentOS 7, Ubuntu 16.04, and Ubuntu 18.04.
4. 인스턴스 유형 선택 페이지에서 `p3dn.24xlarge`를 선택하고 다음: 인스턴스 세부 정보 구성을 선택합니다.
5. [Configure Instance Details] 페이지에서 다음을 수행합니다.
   a. Elastic Fabric Adapter(EFA)에서 활성화를 선택합니다.
   b. 네트워크 인터페이스 항목의 디바이스 eth0에서 새 네트워크 인터페이스를 선택합니다.
   c. Next: Add Storage(다음: 스토리지 추가)를 선택합니다.
6. 스토리지 추가 페이지에서 인스턴스에 연결할 볼륨과 AMI에서 지정한 볼륨(예: 루트 디바이스 볼륨)을 지정합니다. 다음: 태그 추가를 선택합니다.
7. 태그 추가 페이지에서 임시 인스턴스를 식별할 태그를 지정하고 다음: 보안 그룹 구성을 선택합니다.
8. 보안 그룹 구성 페이지의 보안 그룹 할당에서 기존 보안 그룹 선택을 선택합니다. 그런 다음, 1단계에서 생성한 보안 그룹을 선택합니다.
9. 인스턴스 시작 검토 페이지에서 설정을 검토한 후 시작을 선택하여 키 페어를 선택하고 인스턴스를 시작합니다.

3단계: EFA 소프트웨어 설치

EFA를 사용하는 커널, EFA 드라이버, Libfabric 및 임시 인스턴스에서 EFA를 지원하는 데 필요한 OPEN MPI 스택을 설치합니다.

EFA 소프트웨어를 설치하려면

1. 2단계에서 시작한 인스턴스에 연결합니다. 자세한 내용은 Linux 인스턴스에 연결 (p. 485)를 참조하십시오.
2. 모든 소프트웨어 패키지가 최신 상태로 업데이트되어 있는지 확인하기 위해, 인스턴스에서 퀵 소프트웨어 업데이트를 실행합니다. 이 프로세스는 몇 분 정도 걸릴 수 있습니다.

   - Amazon Linux, Amazon Linux 2, RHEL 7.6/7.7, CentOS 7

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

   - Ubuntu 16.04 및 Ubuntu 18.04

     ```
     $ sudo apt-get update -y
     ```

     ```
     $ sudo apt-get upgrade -y
     ```

3. EFA 소프트웨어 설치 파일을 다운로드합니다. 최신 안정 버전을 다운로드하려면 다음 명령을 사용하십시오.

   ```
   $ curl -O https://s3.us-west-2.amazonaws.com/aws-efa-installer/aws-efa-
   installer-1.8.3.tar.gz
   ```

   또한 위의 명령에서 버전 번호를 `latest`로 바꾸면 최신 버전을 얻을 수 있습니다.
4. 소프트웨어 설치 파일은 압축된 `.tar.gz` 파일 형태로 되어 있습니다. 압축 `.tar.gz` 파일에서 파일을 추출하고 추출된 디렉터리로 이동하십시오.

   ```
   $ tar -xf aws-efa-installer-1.8.3.tar.gz
   ```

   ```
   $ cd aws-efa-installer
   ```

5. EFA 소프트웨어 설치 스크립트를 실행합니다.

   ```
   $ sudo ./efa_installer.sh -y
   ```

   Libfabric은 `/opt/amazon/efa` 디렉터리에 설치되는 반면, Open MPI는 `/opt/amazon/openmpi` 디렉터리에 설치됩니다.
6. 인스턴스에서 로그아웃한 후 다시 로그인합니다.
7. EFA 소프트웨어 구성 요소가 성공적으로 설치되었는지 확인합니다.

   ```
   $ fi_info -p efa
   ```

   이 명령은 Libfabric EFA 인스턴스에 대한 정보를 반환합니다. 다음은 해당 명령 출력을 보여주는 예제입니다.

   ```
   provider: efa
       fabric: EFA-fe80::94:3dff:fe89:1b70
       domain: efa_0-rdm
   ```

```
    version: 2.0
    type: FI_EP_RDM
    protocol: FI_PROTO_EFA
provider: efa
    fabric: EFA-fe80::94:3dff:fe89:1b70
    domain: efa_0-dgrm
    version: 2.0
    type: FI_EP_DGRAM
    protocol: FI_PROTO_EFA
provider: efa;ofi_rxd
    fabric: EFA-fe80::94:3dff:fe89:1b70
    domain: efa_0-dgrm
    version: 1.0
    type: FI_EP_RDM
    protocol: FI_PROTO_RXD
```

4단계: Nvidia GPU 드라이버 및 Nvidia CUDA 도구 키트 설치

Nvidia GPU 드라이버 및 Nvidia CUDA 도구 키를 설치하려면

1. Nvidia GPU 드라이버 및 Nvidia CUDA 도구 키트를 설치하는 데 필요한 유틸리티를 설치합니다.

   - Amazon Linux, Amazon Linux 2, RHEL 7.6/7.7, CentOS 7

     ```
     $ sudo yum groupinstall 'Development Tools' -y
     ```

   - Ubuntu 16.04 및 Ubuntu 18.04

     ```
     $ sudo apt-get install build-essential -y
     ```

2. Nvidia GPU 드라이버를 사용하려면, 우선 `nouveau` 오픈 소스 드라이버를 비활성화해야 합니다.

   a. 현재 실행 중인 커널의 버전에 맞는 gcc 컴파일러와 커널 헤더 패키지를 설치합니다.

      - Amazon Linux, Amazon Linux 2, RHEL 7.6/7.7, CentOS 7

        ```
        $ sudo yum install -y gcc kernel-devel-$(uname -r)
        ```

      - Ubuntu 16.04 및 Ubuntu 18.04

        ```
        $ sudo apt-get install -y gcc make linux-headers-$(uname -r)
        ```

   b. `nouveau`를 `/etc/modprobe.d/blacklist.conf` 블랙리스트 파일에 추가합니다.

      ```
      $ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
      blacklist vga16fb
      blacklist nouveau
      blacklist rivafb
      blacklist nvidiafb
      blacklist rivatv
      EOF
      ```

   c. 원하는 텍스트 편집기를 사용하여 `/etc/default/grub`를 열고 다음을 추가합니다.

      ```
      $ GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
      ```

   d. Grub 구성을 다시 빌드합니다.

      - Amazon Linux, Amazon Linux 2, RHEL 7.6/7.7, CentOS 7

```
$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

- Ubuntu 16.04 및 Ubuntu 18.04

```
$ sudo update-grub
```

3. 인스턴스를 재부팅하고 다시 연결합니다.
4. Nvidia CUDA 도구 키트 설치 프로그램을 다운로드합니다.

```
$ wget http://developer.download.nvidia.com/compute/cuda/10.1/Prod/local_installers/
cuda_10.1.243_418.87.00_linux.run
```

5. Nvidia CUDA 도구 키트 설치 프로그램을 실행합니다.

```
$ sudo sh cuda_10.1.243_418.87.00_linux.run
```

   라이선스 계약에 동의하라는 메시지가 표시되면 `accept`를 입력하고 Enter를 누릅니다.
6. CUDA Installer 메뉴에서, 모든 항목이 선택되었는지 확인하고 설치를 강조 표시한 다음 Enter를 누릅니다.
7. 다음 명령문을 셸 시작 스크립트에 추가하여 인스턴스가 시작될 때마다 CUDA 경로가 설정되도록 하십시오.

```
export PATH=/usr/local/cuda-10.1/bin:/usr/local/cuda-10.1/NsightCompute-2019.1${PATH:+:
${PATH}}
export LD_LIBRARY_PATH=/usr/local/cuda-10.1/lib64\${LD_LIBRARY_PATH:+:
${LD_LIBRARY_PATH}}
```

   - bash 셸의 경우 명령문을 `/home/`*`username`*`/.bashrc` 및 `/home/`*`username`*`/.bash_profile`에 추가합니다.
   - tcsh 셸의 경우 명령문을 `/home/`*`username`*`/.cshrc`에 추가합니다.
8. Nvidia GPU 드라이버가 작동하는지 확인하려면 다음 명령을 실행합니다.

```
$ nvidia-smi -q | head
```

   명령을 실행하면 Nvidia GPU, Nvidia GPU 드라이버 및 Nvidia CUDA 도구 키트에 대한 정보가 반환되어야 합니다.

5단계: NCCL 설치

NCCL을 설치합니다. NCCL에 대한 자세한 내용은 NCCL 리포지토리를 참조하십시오.

사전 조건

- NCCL은 Nvidia CUDA 7.0 이상이 필요합니다. 최신 버전 설치에 대한 자세한 내용은 Nvidia 웹 사이트에서 CUDA 도구 키트 10.1 Update 2 다운로드를 참조하십시오.

NCCL을 설치하려면

1. 홈 디렉터리로 이동합니다.

```
$ cd $HOME
```

2. 공식 NCCL 리포지토리를 인스터스에 복제하고 복제된 로컬 리포지토리로 이동합니다.

```
$ git clone https://github.com/NVIDIA/nccl.git
```

```
$ cd nccl
```

3. NCCL을 빌드 및 설치하고 CUDA 설치 디렉터리를 지정합니다. 다음 명령은 CUDA가 기본 디렉터리에 설치되었다고 가정합니다.

```
$ make -j src.build
```

6단계: aws-ofi-nccl 플러그인 설치

aws-ofi-nccl 플러그인은 NCCL의 연결 지향 전송 API를 Libfabric의 연결 없는 안정적인 인터페이스에 매핑합니다. 이렇게 하면 NCCL 기반 애플리케이션을 실행하면서 Libfabric을 네트워크 공급자로 사용할 수 있습니다. aws-ofi-nccl 플러그인에 대한 자세한 내용은 aws-ofi-nccl 리포지토리를 참조하십시오.

aws-ofi-nccl 플러그인을 설치하려면

1. 홈 디렉터리로 이동합니다.

```
$ cd $HOME
```

2. aws-ofi-nccl 플러그인을 설치하는 데 필요한 유틸리티를 설치합니다. 필요한 유틸리티를 설치하려면 다음 명령을 실행합니다.

   - Amazon Linux, Amazon Linux 2, RHEL 7.6/7.7, CentOS 7

   ```
   $ sudo yum install libudev-devel -y
   ```

   - Ubuntu 16.04 및 Ubuntu 18.04

   ```
   $ sudo apt-get install libudev-dev libtool autoconf -y
   ```

3. 공식 AWS aws-ofi-nccl 리포지토리의 `aws` 브랜치를 인스턴스에 복제하고 복제된 로컬 리포지토리로 이동합니다.

```
$ git clone https://github.com/aws/aws-ofi-nccl.git -b aws
```

```
$ cd aws-ofi-nccl
```

4. `configure` 스크립트를 생성하려면 `autogen.sh` 스크립트를 실행합니다.

```
$ ./autogen.sh
```

5. make 파일을 생성하려면 `configure` 스크립트를 실행하고 MPI, Libfabric, NCCL 및 CUDA 설치 디렉터리를 지정합니다.

```
$ ./configure --with-mpi=/opt/amazon/openmpi --with-libfabric=/opt/amazon/efa --with-
nccl=$HOME/nccl/build --with-cuda=/usr/local/cuda-10.1
```

6. aws-ofi-nccl 플러그인을 설치합니다.

```
$ sudo make
```

```
$ sudo make install
```

7단계: NCCL 테스트 설치

NCCL 테스트를 설치합니다. NCCL 테스트를 통해 NCCL이 올바르게 설치되었는지와 예상대로 작동하는지 확인할 수 있습니다. NCCL 테스트에 대한 자세한 내용은 nccl-tests 리포지토리를 참조하십시오.

NCCL 테스트를 설치하려면

1. 홈 디렉터리로 이동합니다.

```
$ cd $HOME
```

2. 공식 nccl-tests 리포지토리를 인스터스에 복제하고 복제된 로컬 리포지토리로 이동합니다.

```
$ git clone https://github.com/NVIDIA/nccl-tests.git
```

```
$ cd nccl-tests
```

3. Libfabric 디렉터리를 `LD_LIBRARY_PATH` 변수에 추가합니다.

   - Amazon Linux, Amazon Linux 2, RHEL 7.6/7.7, CentOS 7

```
$ export LD_LIBRARY_PATH=/opt/amazon/efa/lib64:$LD_LIBRARY_PATH
```

   - Ubuntu 16.04 및 Ubuntu 18.04

```
$ export LD_LIBRARY_PATH=/opt/amazon/efa/lib:$LD_LIBRARY_PATH
```

4. (Amazon Linux, Amazon Linux 2, RHEL 7.6/7.7, CentOS 7에만 해당) 기본적으로 make 파일은 *mpi_home*`/lib` 디렉터리에서 필요한 라이브러리를 찾습니다. 한편, EFA와 함께 설치된 Open MPI의 경우 라이브러리는 *mpi_home*`/lib64`에 위치해 있습니다. make 파일의 경로를 업데이트하려면 다음 명령을 실행합니다.

```
$ sed -i s/'NVLDFLAGS += -L$(MPI_HOME)\/lib -lmpi'/'NVLDFLAGS += -L$(MPI_HOME)\/lib64 -
lmpi'/ src/Makefile
```

5. NCCL 테스트를 설치하고 MPI, NCCL 및 CUDA 설치 디렉터리를 지정합니다.

```
$ make MPI=1 MPI_HOME=/opt/amazon/openmpi NCCL_HOME=$HOME/nccl/build CUDA_HOME=/usr/
local/cuda-10.1
```

8단계: EFA 및 NCCL 구성 테스트

테스트를 실행하여 임시 인스턴스가 EFA 및 NCCL에 올바르게 구성되었는지 확인합니다.

EFA 및 NCCL 구성을 테스트하려면

1. 테스트를 실행할 호스트를 지정하는 호스트 파일을 생성합니다. 다음 명령은 인스턴스 자체에 대한 참조를 포함하는 `my-hosts`라는 호스트 파일을 생성합니다.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-
aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/
meta-data/local-ipv4 >> my-hosts
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/local-ipv4 >> my-hosts
```

2. 테스트를 실행하고 호스트 파일(`--hostfile`) 및 사용할 GPU 수(`-n`)를 지정합니다. 다음 명령은 인스턴스 자체의 8개 GPU에서 `all_reduce_perf` 테스트를 실행하고 다음 환경 변수를 지정합니다.

   - `FI_PROVIDER="efa"`—는 Fabric 인터페이스 공급자를 지정합니다. 이를 `"efa"`로 설정해야 합니다.
   - `FI_EFA_TX_MIN_CREDITS=64`—는 발신자가 수신자로부터 요청하는 최소 발송 크레딧 수를 지정합니다. EFA를 사용하는 NCCL 작업에 권장되는 값은 `64`입니다. 이 값은 256MB보다 큰 메시지 전송시에만 증가되어야 합니다.
   - `NCCL_DEBUG=INFO`—를 통해 상세하게 디버깅을 출력할 수 있습니다. 또한 `VERSION`을 지정하여 테스트 시작 시 NCCL 버전만 인쇄하거나 `WARN`을 지정하여 오류 메시지만 수신할 수 있습니다.
   - `NCCL_TREE_THRESHOLD=0`—은 테스트를 위해 트리 알고리즘을 비활성화합니다.

   NCCL 테스트 알고리즘에 대한 자세한 내용은 공식 nccl-tests 리포지토리에서 NCCL Tests README를 참조하십시오.

   - Amazon Linux, Amazon Linux 2, RHEL 7.6/7.7, CentOS 7

```
$ /opt/amazon/openmpi/bin/mpirun \
 -x FI_PROVIDER="efa" \
 -x FI_EFA_TX_MIN_CREDITS=64 \
 -x LD_LIBRARY_PATH=$HOME/nccl/build/lib:/usr/local/cuda-10.1/lib64:/opt/amazon/efa/
lib64:/opt/amazon/openmpi/lib64:$LD_LIBRARY_PATH \
 -x NCCL_DEBUG=INFO \
 -x NCCL_TREE_THRESHOLD=0 \
 --hostfile my-hosts -n 8 -N 8 \
 --mca btl tcp,self --mca btl_tcp_if_exclude lo,docker0 --bind-to none \
 $HOME/nccl-tests/build/all_reduce_perf -b 8 -e 1G -f 2 -g 1 -c 1 -n 100
```

   - Ubuntu 16.04 및 Ubuntu 18.04

```
$ /opt/amazon/openmpi/bin/mpirun \
 -x FI_PROVIDER="efa" \
 -x FI_EFA_TX_MIN_CREDITS=64 \
 -x LD_LIBRARY_PATH=$HOME/nccl/build/lib:/usr/local/cuda-10.1/lib64:/opt/amazon/efa/
lib:/opt/amazon/openmpi/lib:$LD_LIBRARY_PATH \
 -x NCCL_DEBUG=INFO \
 -x NCCL_TREE_THRESHOLD=0 \
 --hostfile my-hosts -n 8 -N 8 \
 --mca btl tcp,self --mca btl_tcp_if_exclude lo,docker0 --bind-to none \
 $HOME/nccl-tests/build/all_reduce_perf -b 8 -e 1G -f 2 -g 1 -c 1 -n 100
```

9단계: 기계 학습 애플리케이션 설치

기계 학습 애플리케이션을 임시 인스턴스에 설치합니다. 설치 절차는 기계 학습 애플리케이션에 따라 다릅니다. Linux 인스턴스에 소프트웨어를 설치하는 데 대한 자세한 정보는 Linux 인스턴스에서 소프트웨어 관리를 참조하십시오.

> **Note**
>
> 설치 안내를 위한 기계 학습 애플리케이션 설명서를 참조해야 할 수도 있습니다.

10단계: EFA 및 NCCL 지원 AMI 생성

필수 소프트웨어 구성 요소를 설치한 뒤 AMI를 사용하여 EFA를 사용하는 인스턴스를 재사용하고 시작합니다.

임시 인스턴스에서 AMI를 생성하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 1단계에서 생성한 인스턴스를 선택하고 작업, 이미지, 이미지 생성을 선택합니다.
4. 이미지 생성 창에서 다음을 수행합니다.
   a. 이미지 이름에 AMI를 설명하는 이름을 입력합니다.
   b. (선택 사항) 이미지 설명에 AMI에 대한 간략한 설명을 입력합니다.
   c. 이미지 생성을 선택한 후 닫기를 선택합니다.
5. 탐색 창에서 AMI를 선택합니다.
6. 목록에서 생성한 AMI을 찾습니다. 상태가 `pending`에서 `available`으로 바뀌기를 기다렸다가 다음 단계를 수행하십시오.

11단계: 임시 인스턴스 종료

이 단계에서는 1단계에서 시작한 임시 인스턴스가 더 이상 필요하지 않습니다. 추가 요금이 발생하지 않도록 해당 인스턴스를 종료할 수 있습니다.

임시 인스턴스 종료

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 1단계에서 생성한 임시 인스턴스를 선택하고 작업, 인스턴스 상태, 종료, 네, 종료합니다를 선택합니다.

12단계: 클러스터 배치 그룹에 EFA 및 NCCL 지원 인스턴스 시작

앞서 생성한 EFA를 사용한 AMI 및 EFA를 사용한 보안 그룹을 사용하여 클러스터 배치 그룹에 EFA 및 NCCL을 사용한 인스턴스를 시작합니다.

클러스터 배치 그룹에 EFA 및 NCCL을 사용한 인스턴스를 시작하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 인스턴스 시작을 선택합니다.
3. AMI 선택 페이지에서 내 AMI를 선택하고 앞서 생성한 AMI를 찾은 뒤 선택을 선택합니다.
4. 인스턴스 유형 선택 페이지에서 p3dn.24xlarge와 다음: 인스턴스 세부 정보 구성을 차례대로 선택합니다.
5. [Configure Instance Details] 페이지에서 다음을 수행합니다.
   a. 인스턴스 수에서 시작할 EFA 수 및 NCCL 사용 인스턴스 수를 입력합니다.

   b. 네트워크 및 서브넷에서 인스턴스를 시작할 VPC와 서브넷을 선택합니다.
   c. 배치 그룹에서 배치 그룹에 인스턴스를 추가를 선택합니다.
   d. 배치 그룹 이름에서 새 배치 그룹에 추가를 선택한 다음 배치 그룹을 설명하는 이름을 입력합니다. 배치 그룹 전략에서 클러스터를 선택합니다.
   e. EFA에서 활성화를 선택합니다.
   f. 네트워크 인터페이스 항목의 디바이스 eth0에서 새 네트워크 인터페이스를 선택합니다. 하나의 기본 IPv4 주소와 하나 이상의 보조 IPv4 주소를 입력할 수도 있습니다. 연결된 IPv6 CIDR 블록이 있는 서브넷에서 인스턴스를 시작하는 경우 기본 IPv6 주소 및 하나 이상의 보조 IPv6 주소를 지정할 수도 있습니다.
   g. Next: Add Storage(다음: 스토리지 추가)를 선택합니다.
6. 스토리지 추가 페이지에서 인스턴스에 연결할 볼륨과 AMI에서 지정한 볼륨(예: 루트 디바이스 볼륨)을 지정합니다. 다음: 태그 추가를 선택합니다.
7. 태그 추가 페이지에서 사용자에게 친숙한 이름 등의 인스턴스 태그를 지정한 후 다음: 보안 그룹 구성(Next: Configure Security Group)을 선택합니다.
8. 보안 그룹 구성 페이지의 보안 그룹 할당에서 기존 보안 그룹 선택을 선택한 다음 앞에서 생성한 보안 그룹을 선택합니다.
9. 검토 및 시작을 선택합니다.
10. 인스턴스 시작 검토 페이지에서 설정을 검토한 후 시작을 선택하여 키 페어를 선택하고 인스턴스를 시작합니다.

13단계: 암호 없는 SSH 사용

클러스터의 모든 인스턴스에서 실행하도록 애플리케이션을 활성화하려면 리더 노드에서 멤버 노드로의 암호 없는 SSH 액세스를 사용해야 합니다. 리더 노드는 애플리케이션을 실행할 인스턴스입니다. 클러스터의 나머지 인스턴스는 멤버 노드입니다,

클러스터의 인스턴스 간에 암호 없는 SSH를 사용하려면

1. 클러스터의 인스턴스 1개를 리더 노드로 선택하고 연결합니다.
2. `strictHostKeyChecking`을 비활성화하고 리더 노드에서 `ForwardAgent`를 활성화합니다. 원하는 텍스트 편집기를 사용하여 `~/.ssh/config`를 열고 다음을 추가합니다.

```
Host *
    ForwardAgent yes
Host *
    StrictHostKeyChecking no
```

3. RSA 키 페어 생성:

```
$ ssh-keygen -t rsa -N "" -f /home/ubuntu/.ssh/id_rsa
```

   키 페어는 `$HOME/.ssh/` 디렉터리에 생성됩니다.
4. 리더 노드에서 프라이빗 키의 사용 권한을 변경합니다.

```
$ chmod 600 ~/.ssh/id_rsa
```

5. 원하는 텍스트 편집기를 사용하여 `~/.ssh/id_rsa.pub`를 열고 키를 추가합니다.
6. 클러스터의 각 멤버 노드에 대해 다음을 수행합니다.
   a. 인스턴스에 연결합니다.
   b. 원하는 텍스트 편집기를 사용하여 `~/.ssh/authorized_keys`를 열고 앞에서 복사한 퍼블릭 키를 추가합니다.

7. 암호 없는 SSH가 예상대로 작동하는지 테스트하려면 리더 노드에 연결하고 다음 명령을 실행합니다.

```
$ ssh member_node_private_ip
```

키 또는 암호를 입력하라는 메시지가 표시되지 않은 상태에서 멤버 노드에 연결해야 합니다.

AWS 딥 러닝 AMI 사용

아래 단계는 다음 AWS 딥 러닝 AMI 중 하나로 시작하는 데 도움이 됩니다.

- Deep Learning AMI(Amazon Linux 2) 25.0 이상 버전
- Deep Learning AMI(Amazon Linux) 25.0 이상 버전
- Deep Learning AMI(Ubuntu 18.04) 25.0 이상 버전
- Deep Learning AMI(Ubuntu 16.04) 25.0 이상 버전

자세한 내용은 AWS Deep Learning AMI 사용 설명서를 참조하십시오.

목차

- 1단계: EFA를 사용한 보안 그룹을 준비합니다. (p. 750)
- 2단계: 임시 인스턴스 실행 (p. 751)
- 3단계: EFA 및 NCCL 구성 테스트 (p. 751)
- 4단계: 기계 학습 애플리케이션 설치 (p. 752)
- 5단계: EFA 및 NCCL 지원 AMI 생성 (p. 752)
- 6단계: 임시 인스턴스 종료 (p. 752)
- 7단계: 클러스터 배치 그룹에 EFA 및 NCCL 지원 인스턴스 시작 (p. 753)
- 8단계: 암호 없는 SSH 사용 (p. 753)

1단계: EFA를 사용한 보안 그룹을 준비합니다.

EFA는 보안 그룹 자체 내의 모든 인바운드 및 아웃바운드 트래픽을 허용하는 보안 그룹이 필요합니다.

EFA를 사용한 보안 그룹을 생성합니다.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 보안 그룹을 선택한 다음, 보안 그룹 생성을 선택합니다.
3. 보안 그룹 생성 창에서 다음을 수행하십시오.
   a. 보안 그룹 이름의 경우 `EFA-enabled security group`과 같은 보안 그룹의 고유한 이름을 입력합니다.
   b. (선택 사항) 설명에 보안 그룹에 대한 간략한 설명을 입력합니다.
   c. VPC에서는 EFA를 사용한 인스턴스를 시작하려는 VPC를 입력합니다.
   d. 생성을 선택합니다.
4. 생성한 보안 그룹을 선택하고 설명 탭에서 그룹 ID를 복사합니다.
5. 인바운드 및 아웃바운드 탭에서 다음을 수행합니다.
   a. 편집을 선택합니다.
   b. 유형에서 모든 트래픽을 선택합니다.
   c. 소스에서 사용자 지정을 선택합니다.

d. 복사한 보안 그룹 ID를 필드에 붙여넣습니다.
e. Save를 선택합니다.

2단계: 임시 인스턴스 실행

EFA 소프트웨어 구성 요소를 설치하고 구성하는 데 사용할 수 있는 임시 인스턴스를 실행합니다. 이 인스턴스를 사용해 EFA를 사용한 AMI을 생성하여 EFA를 사용한 인스턴스를 실행할 수 있습니다.

임시 인스턴스를 실행합니다.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 인스턴스 시작을 선택합니다.
3. AMI 선택 페이지에서 지원되는 AWS Deep Learning AMI 25.0 이상 버전을 선택합니다.
4. 인스턴스 유형 선택 페이지에서 `p3dn.24xlarge`를 선택하고 다음: 인스턴스 세부 정보 구성을 선택합니다.
5. [Configure Instance Details] 페이지에서 다음을 수행합니다.
   a. Elastic Fabric Adapter(EFA)에서 활성화를 선택합니다.
   b. 네트워크 인터페이스 항목의 디바이스 eth0에서 새 네트워크 인터페이스를 선택합니다.
   c. Next: Add Storage(다음: 스토리지 추가)를 선택합니다.
6. 스토리지 추가 페이지에서 인스턴스에 연결할 볼륨과 AMI에서 지정한 볼륨(예: 루트 디바이스 볼륨)을 지정합니다. 다음: 태그 추가를 선택합니다.
7. 태그 추가 페이지에서 임시 인스턴스를 식별할 태그를 지정하고 다음: 보안 그룹 구성을 선택합니다.
8. 보안 그룹 구성 페이지의 보안 그룹 할당에서 기존 보안 그룹 선택을 선택합니다. 그런 다음, 1단계에서 생성한 보안 그룹을 선택합니다.
9. 인스턴스 시작 검토 페이지에서 설정을 검토한 후 시작을 선택하여 키 페어를 선택하고 인스턴스를 시작합니다.

3단계: EFA 및 NCCL 구성 테스트

테스트를 실행하여 임시 인스턴스가 EFA 및 NCCL에 올바르게 구성되었는지 확인합니다.

EFA 및 NCCL 구성을 테스트하려면

1. 테스트를 실행할 호스트를 지정하는 호스트 파일을 생성합니다. 다음 명령은 인스턴스 자체에 대한 참조를 포함하는 `my-hosts`라는 호스트 파일을 생성합니다.

   IMDSv2

   ```
   [ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
   && curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/local-ipv4 >> my-hosts
   ```

   IMDSv1

   ```
   [ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/local-ipv4 >> my-hosts
   ```

2. 테스트를 실행하고 호스트 파일(`--hostfile`) 및 사용할 GPU 수(`-n`)를 지정합니다. 다음 명령은 인스턴스 자체의 8개 GPU에서 `all_reduce_perf` 테스트를 실행하고 다음 환경 변수를 지정합니다.

   - `FI_PROVIDER="efa"`—는 Fabric 인터페이스 공급자를 지정합니다. 이를 `"efa"`로 설정해야 합니다.

- `FI_EFA_TX_MIN_CREDITS=64`—는 발신자가 수신자로부터 요청하는 최소 발송 크레딧 수를 지정합니다. EFA를 사용하는 NCCL 작업에 권장되는 값은 `64`입니다. 이 값은 256MB보다 큰 메시지 전송시에만 증가되어야 합니다.
- `NCCL_DEBUG=INFO`—를 통해 상세하게 디버깅을 출력할 수 있습니다. 또한 `VERSION`을 지정하여 테스트 시작 시 NCCL 버전만 인쇄하거나 `WARN`을 지정하여 오류 메시지만 수신할 수 있습니다.
- `NCCL_TREE_THRESHOLD=0`—은 테스트를 위해 트리 알고리즘을 비활성화합니다.

NCCL 테스트 알고리즘에 대한 자세한 내용은 공식 nccl-tests 리포지토리에서 NCCL Tests README를 참조하십시오.

```
/opt/amazon/openmpi/bin/mpirun \
 -x FI_PROVIDER="efa" \
 -x FI_EFA_TX_MIN_CREDITS=64 \
 -x NCCL_DEBUG=INFO \
 -x NCCL_TREE_THRESHOLD=0 \
 --hostfile my-hosts -n 8 -N 8 \
 --mca btl tcp,self --mca btl_tcp_if_exclude lo,docker0 --bind-to none \
 $HOME/src/bin/efa-tests/efa-cuda-10.0/all_reduce_perf -b 8 -e 1G -f 2 -g 1 -c 1 -n 100
```

4단계: 기계 학습 애플리케이션 설치

기계 학습 애플리케이션을 임시 인스턴스에 설치합니다. 설치 절차는 기계 학습 애플리케이션에 따라 다릅니다. Linux 인스턴스에 소프트웨어를 설치하는 데 대한 자세한 정보는 Linux 인스턴스에서 소프트웨어 관리를 참조하십시오.

Note

설치 안내를 위한 기계 학습 애플리케이션 설명서를 참조해야 할 수도 있습니다.

5단계: EFA 및 NCCL 지원 AMI 생성

필수 소프트웨어 구성 요소를 설치한 뒤 AMI를 사용하여 EFA를 사용하는 인스턴스를 재사용하고 시작합니다.

임시 인스턴스에서 AMI를 생성하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 1단계에서 생성한 인스턴스를 선택하고 작업, 이미지, 이미지 생성을 선택합니다.
4. 이미지 생성 창에서 다음을 수행합니다.
   a. 이미지 이름에 AMI를 설명하는 이름을 입력합니다.
   b. (선택 사항) 이미지 설명에 AMI에 대한 간략한 설명을 입력합니다.
   c. 이미지 생성을 선택한 후 닫기를 선택합니다.
5. 탐색 창에서 AMI를 선택합니다.
6. 목록에서 생성한 AMI을 찾습니다. 상태가 `pending`에서 `available`으로 바뀌기를 기다렸다가 다음 단계를 수행하십시오.

6단계: 임시 인스턴스 종료

이 단계에서는 1단계에서 시작한 임시 인스턴스가 더 이상 필요하지 않습니다. 추가 요금이 발생하지 않도록 해당 인스턴스를 종료할 수 있습니다.

임시 인스턴스 종료

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 1단계에서 생성한 임시 인스턴스를 선택하고 작업, 인스턴스 상태, 종료, 네, 종료합니다를 선택합니다.

7단계: 클러스터 배치 그룹에 EFA 및 NCCL 지원 인스턴스 시작

앞서 생성한 EFA를 사용한 AMI 및 EFA를 사용한 보안 그룹을 사용하여 클러스터 배치 그룹에 EFA 및 NCCL을 사용한 인스턴스를 시작합니다.

클러스터 배치 그룹에 EFA 및 NCCL을 사용한 인스턴스를 시작하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 인스턴스 시작을 선택합니다.
3. AMI 선택 페이지에서 내 AMI를 선택하고 앞서 생성한 AMI를 찾은 뒤 선택을 선택합니다.
4. 인스턴스 유형 선택 페이지에서 p3dn.24xlarge와 다음: 인스턴스 세부 정보 구성을 차례대로 선택합니다.
5. [Configure Instance Details] 페이지에서 다음을 수행합니다.
   a. 인스턴스 수에서 시작할 EFA 수 및 NCCL 사용 인스턴스 수를 입력합니다.
   b. 네트워크 및 서브넷에서 인스턴스를 시작할 VPC와 서브넷을 선택합니다.
   c. 배치 그룹에서 배치 그룹에 인스턴스를 추가를 선택합니다.
   d. 배치 그룹 이름에서 새 배치 그룹에 추가를 선택한 다음 배치 그룹을 설명하는 이름을 입력합니다. 배치 그룹 전략에서 클러스터를 선택합니다.
   e. EFA에서 활성화를 선택합니다.
   f. 네트워크 인터페이스 항목의 디바이스 eth0에서 새 네트워크 인터페이스를 선택합니다. 하나의 기본 IPv4 주소와 하나 이상의 보조 IPv4 주소를 입력할 수도 있습니다. 연결된 IPv6 CIDR 블록이 있는 서브넷에서 인스턴스를 시작하는 경우 기본 IPv6 주소 및 하나 이상의 보조 IPv6 주소를 지정할 수도 있습니다.
   g. Next: Add Storage(다음: 스토리지 추가)를 선택합니다.
6. 스토리지 추가 페이지에서 인스턴스에 연결할 볼륨과 AMI에서 지정한 볼륨(예: 루트 디바이스 볼륨)을 지정합니다. 다음: 태그 추가를 선택합니다.
7. 태그 추가 페이지에서 사용자에게 친숙한 이름 등의 인스턴스 태그를 지정한 후 다음: 보안 그룹 구성(Next: Configure Security Group)을 선택합니다.
8. 보안 그룹 구성 페이지의 보안 그룹 할당에서 기존 보안 그룹 선택을 선택한 다음 앞에서 생성한 보안 그룹을 선택합니다.
9. 검토 및 시작을 선택합니다.
10. 인스턴스 시작 검토 페이지에서 설정을 검토한 후 시작을 선택하여 키 페어를 선택하고 인스턴스를 시작합니다.

8단계: 암호 없는 SSH 사용

클러스터의 모든 인스턴스에서 실행하도록 애플리케이션을 활성화하려면 리더 노드에서 멤버 노드로의 암호 없는 SSH 액세스를 사용해야 합니다. 리더 노드는 애플리케이션을 실행할 인스턴스입니다. 클러스터의 나머지 인스턴스는 멤버 노드입니다,

클러스터의 인스턴스 간에 암호 없는 SSH를 사용하려면

1. 클러스터의 인스턴스 1개를 리더 노드로 선택하고 연결합니다.
2. `strictHostKeyChecking`을 비활성화하고 리더 노드에서 `ForwardAgent`를 활성화합니다. 원하는 텍스트 편집기를 사용하여 `~/.ssh/config`를 열고 다음을 추가합니다.

```
Host *
    ForwardAgent yes
Host *
    StrictHostKeyChecking no
```

3. RSA 키 페어 생성:

```
$ ssh-keygen -t rsa -N "" -f /home/ubuntu/.ssh/id_rsa
```

   키 페어는 `$HOME/.ssh/` 디렉터리에 생성됩니다.

4. 리더 노드에서 프라이빗 키의 사용 권한을 변경합니다.

```
$ chmod 600 ~/.ssh/id_rsa
```

5. 원하는 텍스트 편집기를 사용하여 `~/.ssh/id_rsa.pub`를 열고 키를 추가합니다.
6. 클러스터의 각 멤버 노드에 대해 다음을 수행합니다.
   a. 인스턴스에 연결합니다.
   b. 원하는 텍스트 편집기를 사용하여 `~/.ssh/authorized_keys`를 열고 앞에서 복사한 퍼블릭 키를 추가합니다.
7. 암호 없는 SSH가 예상대로 작동하는지 테스트하려면 리더 노드에 연결하고 다음 명령을 실행합니다.

```
$ ssh member_node_private_ip
```

   키 또는 암호를 입력하라는 메시지가 표시되지 않은 상태에서 멤버 노드에 연결해야 합니다.

EFA로 작업

Amazon EC2에서 다른 의 탄력적 네트워크 인터페이스처럼 EFA를 생성하고 사용하고 관리할 수 있습니다. 그러나 탄력적 네트워크 인터페이스와 달리 EFAs는 실행 중인 상태에서 인스턴스에 연결하거나 연결 해제할 수 없습니다.

EFA 요구 사항

EFA를 사용하려면 다음을 수행하여야 합니다.

- 지원되는 다음 인스턴스 유형 중 하나를 사용해야 합니다. `c5n.18xlarge`, `c5n.metal`, `i3en.24xlarge`, `i3en.metal`, `inf1.24xlarge`, `m5dn.24xlarge`, `m5n.24xlarge`, `r5dn.24xlarge`, `r5n.24xlarge`, and `p3dn.24xlarge`.
- 지원되는 다음 AMIs 중 하나를 사용해야 합니다. Amazon Linux, Amazon Linux 2, RHEL 7.6, RHEL 7.7, CentOS 7, Ubuntu 16.04, and Ubuntu 18.04.
- EFA 소프트웨어 구성 요소를 설치합니다. 자세한 내용은 3단계: EFA 소프트웨어 설치 (p. 735) 및 5단계: (선택 사항) Intel MPI 설치 (p. 736) 단원을 참조하십시오.
- 보안 그룹 자체 내의 모든 인바운드 및 아웃바운드 트래픽을 허용하는 보안 그룹을 사용합니다. 자세한 내용은 1단계: EFA를 사용한 보안 그룹을 준비합니다. (p. 734) 단원을 참조하십시오.

목차

- EFA 생성 (p. 755)
- 정지된 인스턴스에 EFA를 연결 (p. 755)
- 인스턴스를 시작할 때 EFA 연결 (p. 755)

- 시작 템플릿에 EFA 추가 (p. 756)
- EFA에 IP 주소 할당 (p. 756)
- EFA에서 IP 주소 할당 취소 (p. 756)
- 보안 그룹 변경 (p. 756)
- EFA를 분리 (p. 756)
- EFAs 보기 (p. 757)
- EFA 제거 (p. 757)

EFA 생성

VPC의 서브넷에 EFA를 생성할 수 있습니다. EFA는 일단 생성되고 나면 다른 서브넷으로 옮길 수 없으며 동일 가용 영역의 인스턴스에만 네트워크 인터페이스를 연결할 수 있습니다.

콘솔을 사용하여 새로운 EFA를 생성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 네트워크 인터페이스를 선택합니다.
3. 네트워크 인터페이스 생성을 선택합니다.
4. 설명에 EFA를 설명하는 이름을 입력합니다.
5. 서브넷에서 EFA를 생성하려는 서브넷을 선택합니다.
6. 프라이빗 IP에 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. 보안 그룹에서 하나 이상의 보안 그룹을 선택합니다.
9. EFA에서 활성화를 선택합니다.
10. 예, 생성을 선택합니다.

AWS CLI를 사용하여 새로운 EFA를 생성하려면

다음 예제에서와 같이 create-network-interface 명령을 사용하고 `interface-type`에서 `efa`을 지정합니다.

```
$ aws ec2 create-network-interface --subnet-id subnet-01234567890 --description example_efa
 --interface-type efa
```

정지된 인스턴스에 EFA를 연결

`stopped` 상태에 있는 지원되는 모든 인스턴스에 EFA를 연결할 수 있습니다. `running` 상태인 인스턴스에는 EFA를 연결할 수 없습니다. 지원되는 인스턴스 유형에 대한 자세한 내용은 지원되는 인스턴스 유형 (p. 733)를 참조하십시오.

탄력적 네트워크 인터페이스를 인스턴스에 연결하는 방법을 사용해 EFA를 연결할 수 있습니다. 자세한 내용은 중지되었거나 실행 중인 인스턴스에 네트워크 인터페이스 연결 (p. 700) 단원을 참조하십시오.

인스턴스를 시작할 때 EFA 연결

인스턴스를 시작할 때 기존 EFA 연결(AWS CLI)

다음 예제에서와 같이 run-instances 명령을 사용하고 NetworkInterfaceId에서 EFA의 ID를 입력합니다.

```
$ aws ec2 run-instances --image-id ami_id --count 1 --instance-
type c5n.18xlarge --key-name my_key_pair --network-interfaces
 DeviceIndex=0,NetworkInterfaceId=efa_id,Groups=sg_id,SubnetId=subnet_id
```

인스턴스를 시작할 때 새 EFA 연결(AWS CLI)

다음 예제에서와 같이 run-instances 명령을 사용하고 InterfaceType에 대해 `efa`를 지정합니다.

```
$ aws ec2 run-instances --image-id ami_id --count 1 --instance-
type c5n.18xlarge --key-name my_key_pair --network-interfaces
 DeviceIndex=0,InterfaceType=efa,Groups=sg_id,SubnetId=subnet_id
```

시작 템플릿에 EFA 추가

EFA 지원 인스턴스 시작에 필요한 구성 정보가 포함된 시작 템플릿을 생성할 수 있습니다. EFA 지원 시작 템플릿을 생성하려면 새 시작 템플릿을 생성하고 지원되는 인스턴스 유형과 EFA 지원 AMI, EFA 지원 보안 그룹을 지정합니다. 자세한 내용은 EFA 및 MPI 시작하기 (p. 733) 단원을 참조하십시오.

시작 템플릿을 활용하여 다른 AWS 서비스에 AWS Batch와 같이 EFA 지원 인스턴스를 시작할 수 있습니다.

시작 템플릿 생성에 대한 자세한 내용은 시작 템플릿 생성 (p. 438) 단원을 참조하십시오.

EFA에 IP 주소 할당

유동 IP(IPv4) 주소가 있는 경우 EFA에 할당할 수 있습니다. EFA가 연결된 IPv6 CIDR 블록이 있는 서브넷에 프로비저닝된 경우 EFA에 하나 이상의 IPv6 주소를 할당할 수 있습니다.

탄력적 네트워크 인터페이스에 IP 주소를 할당하는 것과 동일한 방법을 사용하여 EFA에 유동 IP(IPv4) 및 IPv6 주소를 할당할 수 있습니다. 자세한 내용은 다음을 참조하십시오.

- 탄력적 IP 주소(IPv4) 연결 (p. 702)
- IPv6 주소 할당 (p. 703)

EFA에서 IP 주소 할당 취소

탄력적 네트워크 인터페이스에 IP 주소를 할당을 취소한 것과 동일한 방법을 사용하여 EFA에 유동 IP(IPv4) 및 IPv6 주소 할당을 취소할 수 있습니다. 자세한 내용은 다음을 참조하십시오.

- 탄력적 IP 주소(IPv4) 연결 해제 (p. 703)
- IPv6 주소 할당 해제 (p. 704)

보안 그룹 변경

EFA과 연결된 보안 그룹을 변경할 수 있습니다. OS 우회 기능을 사용하기 위해 EFA는 보안 그룹 자체 내의 모든 인바운드 및 아웃바운드 트래픽을 허용하는 보안 그룹에 구성되어야 합니다.

탄력적 네트워크 인터페이스와 연결된 보안 그룹을 변경한 것과 동일한 방법을 사용하여 EFA에 연결된 보안 그룹을 변경할 수 있습니다. 자세한 내용은 보안 그룹 변경 (p. 701) 단원을 참조하십시오.

EFA를 분리

인스턴스에서 EFA을 분리하려면 먼저 인스턴스를 정지해야 합니다. 실행 중인 인스턴스에서 EFA의 연결을 중단시킬 수 없습니다.

탄력적 네트워크 인터페이스를 인스턴스에서 연결 중단하는 방법을 사용해 EFA를 연결 중단할 있습니다. 자세한 내용은 인스턴스에서 네트워크 인터페이스 분리 (p. 701) 단원을 참조하십시오.

EFAs 보기

계정의 모든 EFAs를 볼 수 있습니다.

탄력적 네트워크 인터페이스를 볼 때와 동일한 방법으로 EFAs를 확인합니다. 자세한 내용은 네트워크 인터페이스에 대한 세부 정보 보기 (p. 698) 단원을 참조하십시오.

EFA 제거

EFA을 제거하려면 먼저 인스턴스에서 분리하여야 합니다. 인스턴스에 연결된 상태에서 EFA를 제거할 수 없습니다.

탄력적 네트워크 인터페이스 제거와 동일한 방법으로 EFAs를 제거합니다. 자세한 내용은 네트워크 인터페이스 삭제 (p. 698) 단원을 참조하십시오.

EFA 모니터링

다음 기능을 사용해 Elastic Fabric Adapter(EFA)의 성능을 모니터링할 수 있습니다.

Amazon VPC 플로우 로그

Amazon VPC 플로우 로그를 생성하여 EFA로 들어오고 나가는 트래픽에 대한 세부 정보를 캡처할 수 있습니다. 플로우 로그 데이터를 Amazon CloudWatch Logs 및 Amazon S3로 게시할 수 있습니다. 플로우 로그를 생성한 다음 선택된 대상의 데이터를 가져와 확인할 수 있습니다. 자세한 정보는 Amazon VPC 사용 설명서에서 VPC 흐름 로그를 참조하십시오.

탄력적 네트워크 인터페이스에 플로우 로그를 생성한 것과 동일한 방법으로 EFA에 플로우 로그를 생성할 수 있습니다. 자세한 내용은 Amazon VPC 사용 설명서의 플로우 로그 생성을 참조하십시오.

다음 예제에서와 같이 플로우 로그 항목에서 EFA 트래픽은 MAC 주소 유형인 `srcAddress`와 `destAddress`으로 식별됩니다.

```
version accountId  eniId        srcAddress        destAddress       sourcePort destPort
 protocol packets bytes start      end        action log-status
2       3794735123 eni-10000001 01:23:45:67:89:ab 05:23:45:67:89:ab -          -        -
      9       5689  1521232534 1524512343 ACCEPT OK
```

Amazon CloudWatch

Amazon CloudWatch는 EFAs를 실시간으로 모니터링할 수 있는 지표를 제공합니다. 지표를 수집 및 추적하고, 사용자 지정 대시보드를 생성할 수 있으며, 지정된 지표가 지정한 임계값에 도달하면 사용자에게 알리거나 조치를 취하도록 경보를 설정할 수 있습니다. 자세한 정보는 CloudWatch를 사용해 인스턴스 모니터링하기 (p. 616) 단원을 참조하십시오.

배치 그룹

새 EC2 인스턴스를 시작하면 EC2 서비스는 모든 인스턴스가 기본 하드웨어 전반에 분산되도록 하여 상호 관련 오류의 위험을 줄입니다. 그러나 배치 그룹을 사용하면 워크로드의 요구 사항을 충족하도록 하는 독립

적 인스턴스의 그룹의 배치에 영향을 줄 수 있습니다. 워크로드의 유형에 따라 다음 배치 전략 중 하나를 사용하여 배치 그룹을 생성할 수 있습니다.

- 클러스터 – 인스턴스를 가용 영역 안에 서로 근접하게 패킹합니다. 이 전략은 워크로드가 HPC 애플리케이션에서 일반적인 긴밀히 결합된 노드 간 통신에 필요한 낮은 지연 시간의 네트워크 성능을 달성할 수 있습니다.
- 파티션 – 인스턴스를 논리적 파티션에 분산해, 한 파티션에 있는 인스턴스 그룹이 다른 파티션의 인스턴스 그룹과 기본 하드웨어를 공유하지 않게 합니다. 이 전략은 일반적으로 Hadoop, Cassandra, Kafka 등 대규모의 분산 및 복제된 워크로드에 필요합니다.
- 분산 – 소규모의 인스턴스 그룹을 다른 기본 하드웨어로 분산하여 상호 관련 오류를 줄입니다.

배치 그룹 생성은 무료입니다.

내용

- 클러스터 배치 그룹 (p. 758)
- 파티션 배치 그룹 (p. 759)
- 분산형 배치 그룹 (p. 760)
- 배치 그룹 규칙 및 제한 사항 (p. 760)
- 배치 그룹 생성 (p. 761)
- 배치 그룹으로 인스턴스 시작 (p. 762)
- 배치 그룹의 인스턴스 설명 (p. 763)
- 인스턴스에 대한 배치 그룹 변경 (p. 764)
- 배치 그룹 삭제 (p. 765)

클러스터 배치 그룹

클러스터 배치 그룹은 단일 가용 영역 내에 있는 인스턴스의 논리적 그룹입니다. 클러스터 배치 그룹은 동일한 리전의 피어링된 VPC에 걸쳐 적용될 수 있습니다. 동일한 클러스터 배치 그룹의 인스턴스는 TCP/IP 트래픽에 최대 10Gbps의 더 높은 흐름당 처리량 제한을 제공하며 네트워크의 동일한 높은 양방향 대역폭 세그먼트에 배치됩니다.

다음 이미지는 클러스터 배치 그룹에 배치되는 인스턴스를 보여줍니다.

클러스터 배치 그룹은 짧은 네트워크 지연 시간, 높은 네트워크 처리량 또는 둘 다의 이점을 활용할 수 있는 애플리케이션에 권장됩니다. 또한 대부분의 네트워크 트래픽이 그룹 내 인스턴스 간에 전송되는 경우에도 권

장됩니다. 배치 그룹에 가장 짧은 지연 시간과 가장 높은 초당 패킷 네트워크 성능을 제공하려면 향상된 네트워킹을 지원하는 인스턴스 유형을 선택하십시오. 자세한 내용은 향상된 네트워킹 (p. 706)을 참조하십시오.

다음과 같은 방법으로 인스턴스를 시작하는 것이 좋습니다.

- 단일 시작 요청을 사용하여 배치 그룹에 필요한 수의 인스턴스를 시작합니다.
- 배치 그룹의 모든 인스턴스에 동일한 인스턴스 유형을 사용합니다.

나중에 배치 그룹에 인스턴스를 더 추가하거나 배치 그룹에서 두 가지 이상의 인스턴스 유형을 시작하려고 하면 용량 부족 오류가 발생할 가능성이 커집니다.

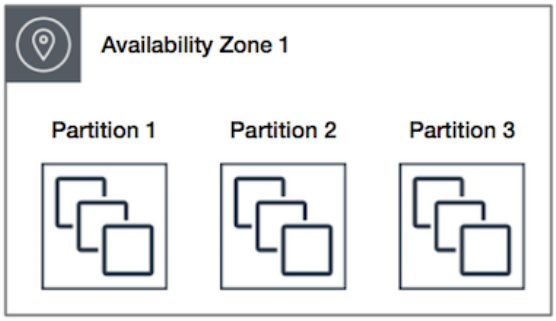
배치 그룹의 인스턴스를 중지한 후 다시 시작하면 인스턴스가 계속 배치 그룹에서 실행됩니다. 그러나 인스턴스에 대해 용량이 부족한 경우 시작에 실패합니다.

이미 인스턴스를 실행한 배치 그룹의 인스턴스를 시작할 때 용량 오류가 발생하는 경우, 배치 그룹의 모든 인스턴스를 중지하고 시작한 후 다시 실행해 보십시오. 인스턴스를 시작하면 요청한 모든 인스턴스를 수용할 용량이 있는 하드웨어로 인스턴스가 마이그레이션될 수 있습니다.

파티션 배치 그룹

파티션 배치 그룹은 애플리케이션에 대한 상관 관계가 있는 하드웨어 장애 가능성을 줄이는 데 도움이 됩니다. 파티션 배치 그룹을 사용할 때 Amazon EC2는 각 그룹을 파티션이라고 하는 논리 세그먼트로 나눕니다. Amazon EC2는 배치 그룹 내 각 파티션에 자체 랙 세트가 있는지 확인합니다. 각 랙은 자체 네트워크 및 전원이 있습니다. 배치 그룹 내 두 파티션이 동일한 랙을 공유하지 않으므로 애플리케이션 내 하드웨어 장애의 영향을 격리시킬 수 있습니다.

다음 이미지는 단일 가용 영역에 있는 파티션 배치 그룹을 시각적으로 간단하게 표현한 것입니다. 여기서는 세 개의 파티션인 파티션 1, 파티션 2 및 파티션 3이 있는 파티션 배치 그룹에 배치된 인스턴스를 보여줍니다. 각 파티션은 여러 인스턴스로 구성됩니다. 각 파티션에 있는 인스턴스는 다른 파티션에 있는 인스턴스와 랙을 공유하지 않기 때문에 단일 하드웨어 장애의 영향을 관련 파티션으로만 국한할 수 있습니다.



파티션 배치 그룹은 HDFS, HBase, Cassandra 같은 대규모 분산 및 복제 워크로드를 별개의 랙으로 분산해 배포하는 데 사용될 수 있습니다. 인스턴스를 파티션 배치 그룹으로 시작하면 Amazon EC2는 사용자가 지정한 수의 파티션에 인스턴스를 균일하게 배포합니다. 인스턴스를 특정 파티션으로 시작하면 인스턴스가 배치되는 위치에 대한 제어를 강화할 수도 있습니다.

파티션 배치 그룹은 동일한 리전의 여러 가용 영역에서 파티션을 가질 수 있습니다. 파티션 배치 그룹은 가용 영역당 파티션을 최대 7개까지 가질 수 있습니다. 파티션 배치 그룹에서 실행할 수 있는 인스턴스 숫자는 계정 제한의 적용을 받습니다.

또한 파티션 배치 그룹은 파티션 확인 기능도 제공합니다. 어떤 인스턴스가 어떤 파티션에 있는지 확인할 수 있습니다. 이 정보를 HDFS, HBase, Cassandra와 같은 토폴로지 인식 애플리케이션과 공유할 수 있습니다. 이러한 애플리케이션은 이 정보를 이용하여 데이터 가용성 및 내구성을 높이기 위한 데이터 복제 결정을 지능적으로 수립합니다.

하나의 파티션 배치 그룹에서 하나의 인스턴스를 시작하고 요청을 이행하기 위한 고유의 하드웨어가 충분하지 않은 경우에는 요청이 실패합니다. Amazon EC2에서는 시간이 지나면서 더 많은 고유 하드웨어를 사용할 수 있기 때문에 나중에 다시 요청을 시도할 수 있습니다.

분산형 배치 그룹

분산형 배치 그룹은 각각 고유한 랙에 배치된 인스턴스 그룹이며 랙마다 자체 네트워크 및 전원이 있습니다.

다음 이미지는 분산형 배치 그룹에 배치되는 단일 가용 영역에 있는 인스턴스 7개를 보여줍니다. 7개의 인스턴스는 7개의 서로 다른 랙에 배치됩니다.

서로 떨어져 있어야 하는 중요 인스턴스의 수가 적은 애플리케이션에서는 분산형 배치 그룹이 권장됩니다. 분산형 배치 그룹에서 인스턴스를 시작하면 인스턴스가 동일한 랙을 공유할 때 장애가 동시에 발생할 수 있는 위험이 줄어듭니다. 분산형 배치 그룹은 별개의 랙에 대한 액세스를 제공하기 때문에 시간에 따라 인스턴스를 시작하거나 인스턴스 유형을 혼합할 때 적합합니다.

분산형 배치 그룹은 동일한 리전의 여러 가용 영역에 적용될 수 있습니다. 그룹당 가용 영역별로 최대 7개의 실행 중인 인스턴스를 가질 수 있습니다.

하나의 분산형 배치 그룹에서 하나의 인스턴스를 시작하고 요청을 이행하기 위한 고유의 하드웨어가 충분하지 않은 경우에는 요청이 실패합니다. Amazon EC2에서는 시간이 지나면서 더 많은 고유 하드웨어를 사용할 수 있기 때문에 나중에 다시 요청을 시도할 수 있습니다.

배치 그룹 규칙 및 제한 사항

일반 규칙 및 제한 사항

배치 그룹을 사용하기 전에 다음 규칙에 유의해야 합니다.

- 배치 그룹에 지정하는 이름은 해당 리전의 AWS 계정 내에서 고유해야 합니다.
- 여러 배치 그룹을 병합할 수는 없습니다.
- 인스턴스는 한 번에 하나의 배치 그룹에서 시작될 수 있습니다. 여러 배치 그룹으로 확장될 수 없습니다.
- 온디맨드 용량 예약 (p. 416) 및 영역 예약 인스턴스 (p. 276)에서는 특정 가용 영역의 EC2 인스턴스에 용량 예약을 제공합니다. 용량 예약은 배치 그룹 내 인스턴스에서 사용이 가능합니다. 하지만 배치 그룹에 대한 용량을 명시적으로 예약할 수는 없습니다.
- 테넌시가 `host`인 인스턴스는 배치 그룹에서 시작할 수 없습니다.

클러스터 배치 그룹 규칙 및 제한 사항

클러스터 배치 그룹에는 다음 규칙이 적용됩니다.

- 인스턴스를 클러스터 배치 그룹으로 시작할 때는 다음 인스턴스 유형 중 하나를 사용해야 합니다.
  - 범용: A1, M4, M5, M5a, M5ad, M5d, M5dn 및 M5n
  - 최적화된 컴퓨팅: C3, C4, C5, C5d, C5n 및 `cc2.8xlarge`
  - 메모리 최적화: `cr1.8xlarge`, R3, R4, R5, R5a, R5ad, R5d, R5dn, R5n, X1, X1e 및 z1d
  - 스토리지 최적화: D2, H1, `hs1.8xlarge`, I2, I3 및 I3en

  - 액셀러레이티드 컴퓨팅: F1, G2, G3, G4dn, P2, P3 및 P3dn
- 클러스터 배치 그룹은 여러 가용 영역을 포괄할 수 없습니다.
- 두 인스턴스의 속도가 느려지면 한 클러스터 배치 그룹에 있는 두 인스턴스 간에 트래픽의 최대 네트워크 처리 속도도 느려집니다. 많은 양을 처리해야 하는 애플리케이션의 경우, 요구 사항을 충족하는 네트워크 연결을 지원하는 인스턴스 유형을 선택하십시오.
- 향상된 네트워킹을 지원하는 인스턴스에는 다음 규칙이 적용됩니다.
  - 클러스터 배치 그룹 내부의 인스턴스는 단일 흐름 트래픽에 최대 10Gbps를 사용할 수 있습니다. 클러스터 배치 그룹 외부의 인스턴스는 단일 흐름 트래픽에 최대 5Gbps를 사용할 수 있습니다.
  - 동일한 리전 내에서 퍼블릭 IP 주소 공간이나 VPC 엔드포인트를 통해 Amazon S3 버킷과 주고받는 트래픽은 사용 가능한 인스턴스 집계 대역폭을 전부 사용할 수 있습니다.
- 하나의 클러스터 배치 그룹으로 여러 인스턴스 유형을 시작할 수 있습니다. 그러나 이렇게 하면 시작에 성공하는 데 필요한 용량이 원활하게 제공될 가능성이 낮아집니다. 클러스터 배치 그룹의 모든 인스턴스에 동일한 인스턴스 유형을 사용하는 것이 좋습니다.
- 인터넷으로 가는 네트워크 트래픽과 AWS Direct Connect 연결을 통해 온프레미스 리소스로 가는 네트워크 트래픽은 5Gbps로 제한됩니다.

파티션 배치 그룹 규칙 및 제한 사항

파티션 배치 그룹에는 다음 규칙이 적용됩니다.

- 파티션 배치 그룹은 가용 영역당 파티션을 최대 7개까지 지원합니다. 파티션 배치 그룹에서 실행할 수 있는 인스턴스 숫자는 계정 제한의 적용을 받습니다.
- 인스턴스를 파티션 배치 그룹으로 시작하면 Amazon EC2는 인스턴스를 모든 파티션에 균일하게 배포합니다. Amazon EC2는 모든 파티션에 대한 균일한 인스턴스 배포를 보장하지 않습니다.
- 전용 인스턴스가 있는 파티션 배치 그룹은 파티션을 최대 2개까지 가질 수 있습니다.
- 파티션 배치 그룹은 전용 호스트에서는 지원되지 않습니다.

분산형 배치 그룹 규칙 및 제한 사항

분산형 배치 그룹에는 다음 규칙이 적용됩니다.

- 분산형 배치 그룹은 각 그룹의 가용 영역당 실행 인스턴스를 최대 7개까지 지원합니다. 예를 들어 가용 영역이 세 개인 리전에서는 그룹에서 총 21개의 실행 인스턴스를 실행할 수 있습니다(영역당 7개). 동일한 가용 영역과 동일한 분산 배치 그룹에서 여덟 번째 인스턴스를 시작하면 그 인스턴스는 시작되지 않습니다. 가용 영역에 인스턴스가 여덟 개 이상 있어야 하며, 분산 배치 그룹을 여러 개 사용할 것을 권장합니다. 여러 분산 배치 그룹을 사용해도 인스턴스가 그룹 간에 분산된다고 보장할 수는 없지만, 각 그룹에 분산함으로써 특정 종류의 실패로 인한 영향을 제한할 수는 있습니다.
- 분산형 배치 그룹은 전용 인스턴스 또는 전용 호스트에서 지원되지 않습니다.

배치 그룹 생성

다음 방법 중 하나를 사용하여 배치 그룹을 생성할 수 있습니다.

새로운 콘솔

콘솔을 사용하여 배치 그룹을 생성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 배치 그룹과 배치 그룹 생성을 차례로 선택합니다.

3. 그룹의 이름을 지정합니다.
4. 그룹의 배치 전략을 선택합니다. 파티션을 선택하는 경우 그룹 내 파티션 수를 선택합니다.
5. 그룹 생성을 선택합니다.

기존 콘솔

콘솔을 사용하여 배치 그룹을 생성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 배치 그룹과 배치 그룹 생성을 차례로 선택합니다.
3. 그룹의 이름을 지정합니다.
4. 그룹의 배치 전략을 선택합니다. 파티션을 선택하는 경우 그룹 내 파티션 수를 지정합니다.
5. Create를 선택합니다.

AWS CLI

AWS CLI를 사용하여 배치 그룹을 생성하려면

create-placement-group 명령을 사용합니다. 이 예에서 배치 그룹 이름은 `my-cluster`이고 배치 전략은 `cluster`입니다.

```
aws ec2 create-placement-group --group-name my-cluster --strategy cluster
```

AWS CLI를 사용하여 파티션 배치 그룹을 생성하려면

create-placement-group 명령을 사용합니다. `partition` 값을 사용하여 `--strategy` 파라미터를 지정하고, 원하는 파티션 수를 사용하여 `--partition-count` 파라미터를 지정합니다. 이 예제에서 파티션 배치 그룹은 이름이 `HDFS-Group-A`이며 파티션 5개로 생성됩니다.

```
aws ec2 create-placement-group --group-name HDFS-Group-A --strategy partition --
partition-count 5
```

PowerShell

Windows PowerShell용 AWS 도구를 사용하여 배치 그룹을 생성하려면

New-EC2PlacementGroup 명령을 사용합니다.

배치 그룹으로 인스턴스 시작

배치 그룹 규칙 및 제한 사항이 충족 (p. 760)되는 경우 다음 방법 중 하나를 사용하여 인스턴스를 배치 그룹으로 시작할 수 있습니다.

Console

콘솔을 사용하여 인스턴스를 배치 그룹으로 시작하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. [Launch Instance]를 선택합니다. 마법사의 안내에 따라 다음 작업을 주의하여 수행하십시오.

- 인스턴스 유형 선택 페이지에서 배치 그룹으로 실행할 인스턴스 유형을 선택합니다.
- 인스턴스 세부 정보 구성 페이지에서 배치 그룹에 다음 필드를 적용할 수 있습니다.
  - 나중에 배치 그룹에 인스턴스를 추가하지 못할 수 있으므로 인스턴스 개수에 이 배치 그룹에서 필요한 총 인스턴스 개수를 입력합니다.
  - 배치 그룹에서 배치 그룹에 인스턴스를 추가합니다. 확인란을 선택합니다. 이 페이지에 배치 그룹이 표시되지 않으면 배치 그룹으로 시작할 수 있는 인스턴스 유형을 선택했는지 확인합니다. 그렇지 않으면 이 옵션을 사용할 수 없습니다.
  - 배치 그룹 이름에서 기존 배치 그룹에 인스턴스를 추가하거나 생성한 새 배치 그룹에 인스턴스를 추가하도록 선택할 수 있습니다.
  - 배치 그룹 전략에서 적합한 전략을 선택합니다. 파티션을 선택하면 Target partition(대상 파티션)에서 Auto distribution(자동 분산)을 선택하여 Amazon EC2에서 인스턴스를 그룹 내 모든 파티션에 골고루 배포하도록 하거나 인스턴스를 시작할 파티션을 지정합니다.

AWS CLI

AWS CLI를 사용하여 인스턴스를 배치 그룹으로 시작하려면

run-instances 명령을 사용하고 `--placement "GroupName = my-cluster"` 파라미터를 사용해 배치 그룹 이름을 지정합니다. 이 예제에서 배치 그룹의 이름은 `my-cluster`입니다.

```
aws ec2 run-instances --placement "GroupName = my-cluster"
```

AWS CLI를 사용하여 인스턴스를 파티션 배치 그룹의 특정 파티션으로 시작하려면

run-instances 명령어를 이용하고 `--placement "GroupName = HDFS-Group-A, PartitionNumber = 3"` 파라미터를 이용해 배치 그룹 이름과 파티션을 지정하십시오. 이 예제에서 파티션 배치 그룹은 이름이 `HDFS-Group-A`이며 파티션 숫자는 `3`개입니다.

```
aws ec2 run-instances --placement "GroupName = HDFS-Group-A, PartitionNumber = 3"
```

PowerShell

Windows PowerShell용 AWS 도구를 사용하여 인스턴스를 배치 그룹으로 시작하려면

New-EC2Instance 명령을 사용하고 `-Placement_GroupName` 파라미터를 사용해 배치 그룹 이름을 지정합니다.

배치 그룹의 인스턴스 설명

다음 방법 중 하나를 사용하여 인스턴스의 배치 정보를 볼 수 있습니다. AWS CLI를 사용하여 파티션 번호별로 파티션 배치 그룹을 필터링할 수도 있습니다.

Console

콘솔을 사용하여 인스턴스의 배치 그룹 및 파티션 번호를 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 세부 정보 창에서 배치 그룹을 살펴봅니다. 인스턴스가 배치 그룹에 없다면, 필드는 빈칸으로 표시됩니다. 그렇지 않다면 배치 그룹 이름이 표시됩니다. 배치 그룹이 파티션 배치 그룹이면 인스턴스의 파티션 번호에 대한 파티션 번호를 검사합니다.

AWS CLI

AWS CLI를 사용하여 파티션 배치 그룹 내 인스턴스의 파티션 번호를 보려면

describe-instances 명령어를 이용하고 `--instance-id` 파라미터를 지정합니다.

```
aws ec2 describe-instances --instance-id i-0123a456700123456
```

응답에는 배치 정보가 포함되며, 이 정보는 인스턴스의 배치 그룹 이름과 파티션 숫자를 포함합니다.

```
"Placement": {
     "AvailabilityZone": "us-east-1c",
     "GroupName": "HDFS-Group-A",
     "PartitionNumber": 3,
     "Tenancy": "default"
}
```

AWS CLI를 사용하여 특정 파티션 배치 그룹 및 파티션 번호로 인스턴스를 필터링하려면

describe-instances 명령어를 사용하고 `placement-group-name` 및 `placement-partition-number` 필터를 이용해 `--filters` 파라미터를 지정합니다. 이 예제에서 파티션 배치 그룹은 이름이 `HDFS-Group-A`이며 파티션 숫자는 `7`개입니다.

```
aws ec2 describe-instances --filters "Name = placement-group-name, Values = HDFS-Group-A" "Name = placement-partition-number, Values = 7"
```

응답은 지정된 배치 그룹에 있는 지정된 파티션의 인스턴스를 모두 나열합니다. 다음은 반환된 인스턴스에 대한 인스턴스 ID, 인스턴스 유형과 배치 정보만 표시하는 출력 예시입니다.

```
"Instances": [
                {
                    "InstanceId": "i-0a1bc23d4567e8f90",
                    "InstanceType": "r4.large",
                    },

                    "Placement": {
                        "AvailabilityZone": "us-east-1c",
                        "GroupName": "HDFS-Group-A",
                        "PartitionNumber": 7,
                        "Tenancy": "default"
                    }

                {
                    "InstanceId": "i-0a9b876cd5d4ef321",
                    "InstanceType": "r4.large",
                    },

                    "Placement": {
                        "AvailabilityZone": "us-east-1c",
                        "GroupName": "HDFS-Group-A",
                        "PartitionNumber": 7,
                        "Tenancy": "default"
                    }
               ],
```

인스턴스에 대한 배치 그룹 변경

다음과 같은 방법으로 인스턴스의 배치 그룹을 변경할 수 있습니다.

- 기존 인스턴스를 배치 그룹으로 이동
- 한 배치 그룹에서 다른 배치 그룹으로 인스턴스 이동
- 배치 그룹에서 인스턴스 제거

인스턴스를 이동하거나 제거하기 전에 인스턴스가 `stopped` 상태여야 합니다. AWS CLI 또는 AWS SDK를 사용하여 인스턴스를 이동하거나 제거할 수 있습니다.

AWS CLI

AWS CLI를 사용하여 인스턴스를 배치 그룹으로 이동하려면

1. stop-instances 명령을 사용하여 인스턴스를 중지합니다.
2. modify-instance-placement 명령을 사용하고 인스턴스가 이동할 배치 그룹의 이름을 지정합니다.

```
aws ec2 modify-instance-placement --instance-id i-0123a456700123456 --group-
name MySpreadGroup
```

3. start-instances 명령을 사용하여 인스턴스를 시작합니다.

PowerShell

Windows PowerShell용 AWS 도구를 사용하여 인스턴스를 배치 그룹으로 이동하려면

1. Stop-EC2Instance 명령을 사용하여 인스턴스를 중지합니다.
2. Edit-EC2InstancePlacement 명령을 사용하고 인스턴스가 이동할 배치 그룹의 이름을 지정합니다.
3. Start-EC2Instance 명령을 사용하여 인스턴스를 중지합니다.

AWS CLI

AWS CLI를 사용하여 배치 그룹에서 인스턴스를 제거하려면

1. stop-instances 명령을 사용하여 인스턴스를 중지합니다.
2. modify-instance-placement 명령을 사용하고 배치 그룹 이름에 대해 빈 문자열을 지정합니다.

```
aws ec2 modify-instance-placement --instance-id i-0123a456700123456 --group-name ""
```

3. start-instances 명령을 사용하여 인스턴스를 시작합니다.

PowerShell

Windows PowerShell용 AWS 도구를 사용하여 배치 그룹에서 인스턴스를 제거하려면

1. Stop-EC2Instance 명령을 사용하여 인스턴스를 중지합니다.
2. Edit-EC2InstancePlacement 명령을 사용하고 배치 그룹 이름에 대해 빈 문자열을 지정합니다.
3. Start-EC2Instance 명령을 사용하여 인스턴스를 중지합니다.

배치 그룹 삭제

대체해야 하거나 더 이상 필요하지 않은 배치 그룹을 삭제할 수 있습니다. 다음 방법 중 하나를 사용하여 배치 그룹을 삭제할 수 있습니다.

Important

배치 그룹을 삭제하려면 배치 그룹에 인스턴스가 없어야 합니다. 배치 그룹으로 시작한 모든 인스턴스를 종료 (p. 523)하거나, 인스턴스를 다른 배치 그룹으로 이동 (p. 765)하거나, 인스턴스를 배치 그룹에서 제거 (p. 765)해야 합니다. 인스턴스 화면에서 인스턴스를 선택하고 세부 정보 창에서 배치 그룹의 값을 확인하면 인스턴스를 종료 또는 이동하기 전에 인스턴스가 해당 배치 그룹에 있는지 확인할 수 있습니다.

새로운 콘솔

콘솔을 사용하여 배치 그룹을 삭제하려면 다음을 수행합니다.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 배치 그룹을 선택합니다.
3. 배치 그룹을 선택하고 삭제를 선택합니다.
4. 확인 메시지가 나타나면 **Delete**를 입력한 다음 삭제를 선택합니다.

기존 콘솔

콘솔을 사용하여 배치 그룹을 삭제하려면 다음을 수행합니다.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 배치 그룹을 선택합니다.
3. 배치 그룹을 선택하고 배치 그룹 삭제를 선택합니다.
4. 확인 메시지가 나타나면 삭제를 선택합니다.

AWS CLI

AWS CLI를 사용하여 배치 그룹을 삭제하려면 다음을 수행합니다.

delete-placement-group 명령을 사용하고 배치 그룹 이름을 지정하여 배치 그룹을 삭제합니다. 이 예제에서 배치 그룹의 이름은 `my-cluster`입니다.

```
aws ec2 delete-placement-group --group-name my-cluster
```

PowerShell

Windows PowerShell용 AWS 도구를 사용하여 배치 그룹을 삭제하려면 다음을 수행합니다.

Remove-EC2PlacementGroup 명령을 사용하여 배치 그룹을 삭제합니다.

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 지원되

는 최대 MTU는 인스턴스 유형에 따라 다릅니다. 모든 Amazon EC2 인스턴스 유형은 1500MTU를 지원하며, 현재 다수의 인스턴스 크기가 9001MTU 또는 점보 프레임을 지원합니다.

내용

- 점보 프레임(9001 MTU) (p. 767)
- 경로 MTU 검색 (p. 767)
- 두 호스트 간 경로 MTU 확인 (p. 768)
- Linux 인스턴스에서 MTU 확인 및 설정 (p. 768)
- 문제 해결 (p. 769)

점보 프레임(9001 MTU)

점보 프레임에서는 패킷당 페이로드 크기를 늘려 1500바이트 이상의 데이터가 허용됩니다. 그 결과, 패킷 오버헤드에 해당하지 않는 패킷의 비율의 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.

그러나 클러스터 배치 그룹 내부에 함께 배치된 인스턴스의 경우, 점보 프레임이 최고의 네트워크 처리 속도를 달성하는 데 도움을 주므로 사용이 권장됩니다. 자세한 내용은 배치 그룹 (p. 757) 단원을 참조하십시오.

AWS Direct Connect를 통한 VPC와 온프레미스 네트워크 간의 트래픽에 점보 프레임을 사용할 수 있습니다. 자세한 내용과 점보 프레임 기능 확인 방법은 AWS Direct Connect 사용 설명서의 네트워크 MTU 설정을 참조하십시오.

모든 현재 세대의 인스턴스 (p. 183)는 점보 프레임을 지원합니다. 다음의 이전 세대 인스턴스는 점보 프레임을 지원합니다: C3, G2, I2, M3, 및 R3.

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 대상에 연결할 수 없음 프로토콜이 있는 사용자 지정 ICMP 규칙을 추가해야 합니다. 자세한 내용은 경로 MTU 검색 규칙 (p. 883) 단원을 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(여러 Linux 배포에서 기본적으로 사용할 수 있는 `iputils` 패키지의 일부로,에서 다운로드하고 설치할 수 있는 Amazon Linux 포함)을 사용하여 두 호스트 사이에 경로 MTU를 확인할 수 있습니다.

tracepath를 사용하여 경로 MTU를 확인하려면

다음 명령을 사용하여 EC2 인스턴스와 다른 호스트 간의 경로 MTU를 확인합니다. DNS 이름 또는 IP 주소를 대상으로 사용할 수 있습니다. 대상이 다른 EC2 인스턴스인 경우에는 보안 그룹에서 인바운드 UDP 트래픽을 허용하는지 확인합니다. 이 예제에서는 EC2 인스턴스와 `amazon.com` 간의 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.

Linux 인스턴스에서 MTU 확인 및 설정

일부 인스턴스는 점보 프레임을 사용하도록 구성되어 있는 반면, 표준 프레임 크기를 사용하도록 구성된 인스턴스도 있습니다. VPC 내의 네트워크 트래픽에 점보 프레임을 사용하거나, 인터넷 트래픽에 표준 프레임을 사용할 수 있습니다. 어떤 사용 사례이든 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

다음 ip 명령을 사용하여 현재 MTU 값을 확인할 수 있습니다. 예제 출력에서 *`mtu 9001`*은 이 인스턴스가 점보 프레임을 사용함을 나타냅니다.

```
[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
```

Linux 인스턴스에서 MTU 값을 설정하려면

1. ip 명령을 사용하여 MTU 값을 설정할 수 있습니다. 다음 명령은 원하는 MTU 값을 1500으로 설정하지만, 9001을 대신 사용할 수도 있습니다.

   ```
   [ec2-user ~]$ sudo ip link set dev eth0 mtu 1500
   ```

2. (선택 사항) 재부팅 후에 네트워크 MTU 설정을 유지하려면 운영 체제 유형을 기반으로 다음 구성 파일을 수정하십시오.

- Amazon Linux 2의 경우, `/etc/sysconfig/network-scripts/ifcfg-eth0` 파일에 다음 줄을 추가합니다.

```
MTU=1500
```

  `/etc/dhcp/dhclient.conf` 파일에 다음 줄을 추가합니다.

```
request subnet-mask, broadcast-address, time-offset, routers, domain-name, domain-
search, domain-name-servers, host-name, nis-domain, nis-servers, ntp-servers;
```

- Amazon Linux의 경우, 다음 줄을 `/etc/dhcp/dhclient-eth0.conf` 파일에 추가합니다.

```
interface "eth0" {
supersede interface-mtu 1500;
}
```

- 다른 Linux 배포의 경우, 해당 설명서를 참조하십시오.

3. (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 나타남을 참조하십시오.

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 Amazon EC2 리소스를 VPC의 서브넷으로 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 인터넷 또는 자체 데이터 센터에 연결합니다.

AWS 계정이 생성되면 각 리전에서 기본 VPC가 생성됩니다. 기본 VPC는 이미 구성되어 즉시 사용할 수 있는 VPC입니다. 기본 VPC로 인스턴스를 즉시 시작할 수 있습니다. 또는 기본이 아닌 VPC를 직접 생성할 수 있으며 필요에 따라 구성할 수 있습니다.

2013년 12월 4일 이전에 AWS 계정을 생성한 경우 일부 리전에서 EC2-Classic 플랫폼 지원을 받을 수 있습니다. 2013년 12월 4일 이후에 AWS 계정을 생성한 경우 EC2-Classic 지원이 되지 않으므로 VPC에서 리소스를 시작해야 합니다. 자세한 내용은 EC2-Classic (p. 770) 단원을 참조하십시오.

Amazon VPC 문서

Amazon VPC에 대한 자세한 내용은 다음 문서를 참조하십시오.

가이드	설명
Amazon VPC 사용 설명서	Amazon VPC의 주요 개념을 설명하고 기능 사용에 대한 지침을 제공합니다.
Amazon VPC Peering Guide	VPC 피어링 연결을 설명하고 기능 사용에 대한 지침을 제공합니다.

가이드	설명
AWS Site-to-Site VPN 네트워크 관리자 안내서	네트워크 관리자가 고객 게이트웨이를 구성하도록 돕습니다.

EC2-Classic

EC2-Classic 사용을 통해 다른 고객과 공유하는 단일 일반 네트워크에서 인스턴스가 실행됩니다. Amazon VPC 사용을 통해 AWS 계정에 속하도록 논리적으로 독립된 Virtual Private Cloud(VPC)에서 인스턴스가 실행됩니다.

Amazon EC2의 최초 출시 당시 EC2-Classic 플랫폼이 도입되었습니다. 2013년 12월 4일 이후에 AWS 계정을 생성한 경우 EC2-Classic을 지원하지 않으니 VPC에서 Amazon EC2 인스턴스를 시작해야 합니다.

계정이 EC2-Classic을 지원하지 않는 경우 기본 VPC가 생성됩니다. 기본적으로 인스턴스를 시작할 때 이를 기본 VPC에 시작합니다. 또는 기본 VPC가 아닌 VPC를 생성하고 인스턴스를 시작할 때 이를 지정할 수 있습니다.

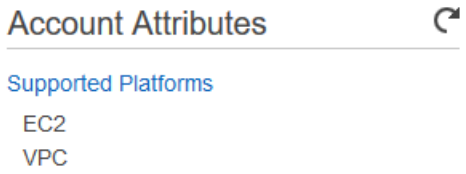
지원되는 플랫폼 감지

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 계정 속성에서 지원 가능 플랫폼을 찾습니다.

EC2-Classic를 지원하는 계정

대시보드에서 계정 속성 아래에 다음과 같은 정보가 표시되어 계정에서 리전의 EC2-Classic 플랫폼과 VPC 모두 지원하지만 리전에 기본 VPC가 없음을 나타냅니다.

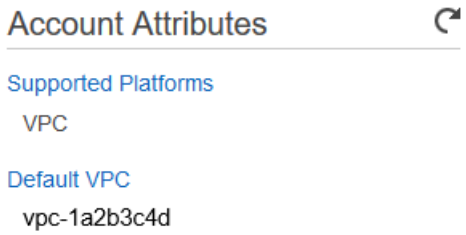


describe-account-attributes 명령의 출력에는 `supported-platforms` 속성의 `EC2` 및 `VPC` 값이 포함됩니다.

```
aws ec2 describe-account-attributes --attribute-names supported-platforms
{
    "AccountAttributes": [
        {
            "AttributeName": "supported-platforms",
            "AttributeValues": [
                {
                    "AttributeValue": "EC2"
                },
                {
                    "AttributeValue": "VPC"
                }
            ]
        }
    ]
}
```

VPC를 필요로 하는 계정

대시보드에서 계정 속성 아래에 다음과 같은 정보가 표시되어 이 리전에서 인스턴스를 시작하려면 계정에 VPC가 필요하고, 이 리전에서 EC2-Classic 플랫폼을 지원하지 않으며, 리전에 식별자가 `vpc-1a2b3c4d`인 기본 VPC가 있음을 나타냅니다.



describe-account-attributes 명령의 출력에는 `supported-platforms` 속성의 `VPC` 값만 포함됩니다.

```
aws ec2 describe-account-attributes --attribute-names supported-platforms
{
    "AccountAttributes": [
        {
            "AttributeValues": [
                {
                    "AttributeValue": "VPC"
                }
            ]
            "AttributeName": "supported-platforms",
        }
    ]
}
```

EC2-Classic에서 사용 가능한 인스턴스 유형

최신 인스턴스 유형 대부분은 VPC를 필요로 합니다. 다음은 EC2-Classic에서 지원되는 인스턴스 유형입니다.

- 범용: M1, M3 및 T1
- 컴퓨팅 최적화: C1, C3 및 CC2
- 메모리 최적화: CR1, M2 및 R3
- 스토리지 최적화: D2, HS1 및 I2
- 액셀러레이티드 컴퓨팅: G2

계정에서 EC2-Classic을 지원하며 기본이 아닌 VPC를 생성하지 않은 경우 다음 중 하나를 수행하여 VPC를 필요로 하는 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.
- 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 설명서의 인터넷 게이트웨이 단원을 참조하십시오.
- VPC의 기본 보안 그룹 및 각 서브넷과 연결된 기본 네트워크 ACL입니다. 자세한 정보는 Amazon VPC 사용 설명서의 VPC의 보안 그룹 단원을 참조하십시오.

EC2-Classic에 다른 리소스가 있는 경우 단계를 밟아 VPC로 마이그레이션할 수 있습니다. 자세한 내용은 Linux 내 EC2-Classic 인스턴스에서 VPC 내 Linux 인스턴스로 마이그레이션 (p. 788) 단원을 참조하십시오.

EC2-Classic과 VPC의 인스턴스 간 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)	인스턴스에서 EC2-Classic 퍼블릭 IPv4 주소 풀로부터 퍼블릭 IPv4 주소를 수신합니다.	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 다중 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 중지하면 탄력적 IP는 인스턴스에서 연결 해제됩니다.	인스턴스를 중지하면 탄력적 IP는 인스턴스와 연결된 상태를 유지합니다.	인스턴스를 중지하면 탄력적 IP는 인스턴스와 연결된 상태를 유지합니다.
탄력적 IP 주소 연결	탄력적 IP 주소를 인스턴스와 연결할 수 있습니다.	탄력적 IP 주소는 네트워크 인터페이스의 속성입니다. 인스턴스에 연결된 네트워크 인터페이스를 업데이트하여 탄력적 IP 주소를 인스턴스와 연결합니다.	탄력적 IP 주소는 네트워크 인터페이스의 속성입니다. 인스턴스에 연결된 네트워크 인터페이스를 업데이트하여 탄력적 IP 주소를 인스턴스와 연결합니다.
탄력적 IP 주소 재연결	탄력적 IP 주소가 이미 다른 인스턴스와 이미 연결된 경우 주소가 자동으로 새 인스턴스와 연결됩니다.	탄력적 IP 주소가 이미 다른 인스턴스와 이미 연결된 경우 주소가 자동으로 새 인스턴스와 연결됩니다.	탄력적 IP 주소가 다른 인스턴스와 이미 연결된 경우 재연결을 허용한 경우에만 성공합니다.
탄력적 IP 주소 태그 지정	탄력적 IP 주소에 태그를 적용할 수 없습니다.	탄력적 IP 주소에 태그를 적용할 수 있습니다.	탄력적 IP 주소에 태그를 적용할 수 있습니다.
DNS 호스트 이름	기본적으로 DNS 호스트 이름을 사용하도록 되어 있습니다.	기본적으로 DNS 호스트 이름을 사용하도록 되어 있습니다.	기본적으로 DNS 호스트 이름을 사용하지 않도록 되어 있습니다.

특성	EC2-Classic	기본 VPC	기본이 아닌 VPC
보안 그룹	보안 그룹에서 다른 AWS 계정에 속한 보안 그룹을 참조할 수 있습니다.	보안 그룹에서 VPC 또는 VPC 피어링 연결의 피어 VPC에서 보안 그룹을 참조할 수 있습니다.	보안 그룹에서는 사용자의 VPC 내 보안 그룹만 참조할 수 있습니다.
보안 그룹 연결	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 연결을 해제하고 해당 주소를 새 인스턴스에 할당 -> 기존 인스턴스 종료).	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.
보안 그룹 규칙	인바운드 트래픽에만 규칙을 추가할 수 있습니다.	인바운드 및 아웃바운드 모두에 규칙을 지정 할 수 있습니다.	인바운드 및 아웃바운드 모두에 규칙을 지정 할 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.

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 후에는 해당 보안 그룹을 변경할 수 없습니다. 그러나 보안 그룹에 규칙을 추가하거나 보안 그룹에서 규칙을 제거할 수 있으며, 잠시 후 보안 그룹과 연결된 모든 인스턴스에 해당 변경 내용이 자동으로 적용됩니다.

AWS 계정에서 자동으로 EC2-Classic 리전별 기본 보안 그룹을 보유합니다. 기본 보안 그룹을 삭제하려는 경우 다음과 같은 오류가 표시됩니다. Client.InvalidGroup.Reserved: The security group 'default' is reserved.

사용자 지정 보안 그룹을 생성할 수 있습니다. 보안 그룹 이름은 해당 리전의 사용자 계정 내에서 고유해야 합니다. EC2-Classic에서 사용할 보안 그룹을 생성하려면 VPC에 대해 VPC 없음을 선택하십시오.

기본 및 사용자 지정 보안 그룹에 인바운드 규칙을 추가할 수 있습니다. EC2-Classic 보안 그룹에 대한 아웃바운드 규칙을 변경할 수 없습니다. 보안 그룹을 생성할 때 동일한 리전에 있는 EC2-Classic에 대해 다른 보안 그룹을 소스 또는 대상으로 사용할 수 있습니다. 다른 AWS 계정에 대한 보안 그룹을 지정하려면 AWS 계정 ID를 접두사로 추가합니다. 예: `111122223333/sg-edcd9784`.

EC2-Classic에서는 계정별로 각 리전에 최대 500개의 보안 그룹이 있을 수 있습니다. 하고 보안 그룹에 최대 100개의 규칙을 추가할 수 있습니다.

IP 주소 지정 및 DNS

Amazon은 Amazon이 제공한 IPv4 DNS 호스트 이름을 IPv4 주소로 변환하는 DNS 서버를 제공합니다. EC2-Classic의 경우 Amazon DNS 서버는 `172.16.0.23`에 위치합니다.

EC2-Classic에서 사용자 지정 방화벽 구성을 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.

```
grep nameserver /etc/resolv.conf
```

탄력적 IP 주소

계정이 EC2-Classic을 지원하는 경우 EC2-Classic 플랫폼에 사용할 수 있는 탄력적 IP 주소 풀과 VPC에 사용할 수 있는 탄력적 IP 주소 풀이 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 주소를 VPC 플랫폼에서 사용하도록 마이그레이션할 수 있습니다. 탄력적 IP 주소는 다른 리전으로 마이그레이션할 수 없습니다.

콘솔을 사용하여 EC2-Classic에서 사용할 탄력적 IP 주소를 할당하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 탄력적 IP를 선택합니다.
3. Allocate new address를 선택합니다.
4. Classic을 선택한 후 할당을 선택합니다. 확인 화면을 닫습니다.

EC2-Classic에서 탄력적 IP 주소 마이그레이션

계정이 EC2-Classic을 지원하는 경우 EC2-Classic 플랫폼에서 사용할 목적으로 할당한 탄력적 IP 주소를 동일 리전 내 VPC로 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.

VPC로 탄력적 IP 주소를 마이그레이션한 이후 EC2-Classic에서 이를 사용할 수 없습니다. 하지만 필요한 경우 이를 EC2-Classic에 복원할 수 있습니다. 처음부터 VPC에서 사용할 목적으로 할당한 탄력적 IP 주소는 EC2-Classic으로 마이그레이션할 수 없습니다.

탄력적 IP 주소를 마이그레이션하려면 연결되어 있는 인스턴스가 없어야 합니다. 탄력적 IP 주소를 인스턴스에서 연결 해제하기 위한 자세한 내용은 탄력적 IP 주소 연결 해제 (p. 681) 단원을 참조하십시오.

EC2-Classic 탄력적 IP 주소는 계정에 가질 수 있는 만큼 마이그레이션할 수 있습니다. 하지만 탄력적 IP 주소를 마이그레이션하는 경우 VPC에 대한 탄력적 IP 주소 한도에서 차감됩니다. 따라서 한도를 초과하

는 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. 683) 단원을 참조하십시오.

계정에 할당되고 24시간이 지나지 않은 탄력적 IP 주소는 마이그레이션할 수 없습니다.

Amazon EC2 콘솔 또는 Amazon VPC 콘솔을 사용하여 EC2-Classic에서 탄력적 IP 주소를 마이그레이션할 수 있습니다. 이 옵션은 계정이 EC2-Classic을 지원하는 경우에만 사용 가능합니다.

Amazon EC2 콘솔을 사용하여 탄력적 IP 주소를 이동하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 탄력적 IP를 선택합니다.
3. 탄력적 IP 주소를 선택하고 작업, VPC 범위로 이동을 선택합니다.
4. 확인 대화 상자가 나타나면 탄력적 IP 이동을 선택합니다.

Amazon EC2 콘솔 또는 Amazon VPC 콘솔을 사용하여 탄력적 IP 주소를 EC2-Classic으로 복원할 수 있습니다.

Amazon EC2 콘솔을 사용하여 탄력적 IP 주소를 EC2-Classic으로 복원하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 탄력적 IP를 선택합니다.
3. 탄력적 IP 주소를 선택하고 작업, EC2 범위로 복원을 선택합니다.
4. 확인 대화 상자가 나타나면 복원을 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 이동한 후 할당 ID 필드의 탄력적 IP 페이지에서 할당 ID를 확인할 수 있습니다.

탄력적 IP 주소를 이동할 때 5분이 지나도 완료되지 않으면 Premium Support에 문의하십시오.

명령줄을 사용하여 탄력적 IP 주소를 이동하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- move-address-to-vpc(AWS CLI)
- Move-EC2AddressToVpc(Windows PowerShell용 AWS 도구)

명령줄을 사용하여 탄력적 IP 주소를 EC2-Classic으로 복원하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- restore-address-to-classic(AWS CLI)
- Restore-EC2AddressToClassic(Windows PowerShell용 AWS 도구)

명령줄을 사용하여 주소 이동 상태를 설명하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- describe-moving-addresses(AWS CLI)
- Get-EC2Address(Windows PowerShell용 AWS 도구)

EC2-Classic 및 VPC 간 리소스 공유 및 액세스

AWS 계정의 일부 리소스와 기능은 EC2-Classic 및 VPC 간에서 공유 또는 액세스될 수 있습니다(예: ClassicLink). 자세한 내용은 ClassicLink (p. 777) 단원을 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. 788)을(를) 참조하십시오.

다음은 EC2-Classic와 VPC 간에 공유나 액세스가 가능한 리소스입니다.

리소스	참고
AMI	
번들 작업	
EBS 볼륨	
탄력적 IP 주소(IPv4)	탄력적 IP 주소는 EC2-Classic에서 VPC로 마이그레이션할 수 있습니다. 처음부터 EC2-VPC에서 사용할 목적으로 할당한 탄력적 IP 주소는 EC2-Classic으로 마이그레이션하지 못합니다. 자세한 내용은 EC2-Classic에서 탄력적 IP 주소 마이그레이션 (p. 774) 단원을 참조하십시오.
인스턴스	EC2-Classic 인스턴스는 퍼블릭 IPv4 주소를 할당받은 VPC의 인스턴스와 통신할 수 있으며, 프라이빗 IPv4 주소를 가지고 있는 인스턴스와의 통신을 원하는 경우 ClassicLink를 사용할 수 있습니다. EC2-Classic에서 VCP로 인스턴스를 마이그레이션할 수 없습니다. 그러나 EC2-Classic 내 인스턴스에서 VPC 내 인스턴스로는 애플리케이션을 마이그레이션할 수 있습니다. 자세한 내용은 Linux 내 EC2-Classic 인스턴스에서 VPC 내 Linux 인스턴스로 마이그레이션 (p. 788) 단원을 참조하십시오.
키 페어	
로드 밸런서	ClassicLink를 사용하는 경우 VPC 내 링크한 EC2-Classic 인스턴스에 로드 밸런서를 등록할 수 있습니다. 이때 VPC가 인스턴스처럼 동일한 가용 영역에 서브넷을 보유해야 합니다. EC2-Classic에서 VCP로 로드 밸런서를 마이그레이션 할 수 없습니다. VPC 내 EC2-Classic에 인스턴스를 등록할 수 없습니다.
배치 그룹	
예약 인스턴스	예약 인스턴스에 대한 네트워크 플랫폼을 EC2-Classic에서 VPC로 변경할 수 있습니다. 자세한 내

리소스	참고
	용은 예약 인스턴스 수정 (p. 297) 단원을 참조하십시오.
보안 그룹	링크한 EC2-Classic 인스턴스는 ClassicLink을(를) 통해서 VPC 보안 그룹을 사용하여 VPC로 그리고 VPC로부터 트래픽을 제어할 수 있습니다. VPC 인스턴스에서 EC2-Classic 보안 그룹을 사용할 수 없습니다. EC2-Classic에서 VCP로 보안 그룹을 마이그레이션할 수 없습니다. EC2-Classic의 보안 그룹에서 VPC의 보안 그룹으로 규칙을 복사할 수 있습니다. 자세한 내용은 보안 그룹 생성 (p. 876) 단원을 참조하십시오.
스냅샷	

다음은 EC2-Classic와 VPC 간에 공유나 이동이 불가능한 리소스입니다.

- 스팟 인스턴스

ClassicLink

ClassicLink를 사용하면 같은 리전 내에 있는 해당 계정의 VPC에 EC2-Classic 인스턴스를 연결할 수 있습니다. VPC 보안 그룹과 EC2-Classic 인스턴스를 연결하는 경우 이를 통해 EC2-Classic 인스턴스와 프라이빗 IPv4 주소를 사용하는 VPC 내 인스턴스와 통신할 수 있습니다. ClassicLink를 사용하면 이러한 플랫폼의 인스턴스 간 통신을 위해 퍼블릭 IPv4 주소 또는 탄력적 IP 주소를 사용할 필요가 없습니다.

ClassicLink는 EC2-Classic 플랫폼을 지원하는 계정을 갖는 모든 사용자에게 제공되며 모든 EC2-Classic 인스턴스에 사용할 수 있습니다. 리소스를 VPC로 마이그레이션한 방법에 대한 자세한 내용은 Linux 내 EC2-Classic 인스턴스에서 VPC 내 Linux 인스턴스로 마이그레이션 (p. 788) 단원을 참조하십시오.

ClassicLink 사용에 따르는 추가 요금은 없습니다. 데이터 전송 및 인스턴스 사용에 대한 표준 요금이 그대로 적용됩니다.

내용

- ClassicLink 기본 사항 (p. 777)
- ClassicLink 제한 사항 (p. 780)
- ClassicLink 작업 (p. 780)
- ClassicLink에 대한 예제 IAM 정책 (p. 784)
- 예: 3티어 웹 애플리케이션의 ClassicLink 보안 그룹 구성 (p. 786)

ClassicLink 기본 사항

두 단계를 통해 EC2-Classic를 사용하여 ClassicLink 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 인스턴스 화면을 사용하여 인스턴스를 나열하고 VPC로 필터링하면 VPC에 링크된 EC2-Classic 인스턴스는 결과로 반환되지 않습니다. 링크된 EC2-Classic 인스턴스를 확인하는 방법에 대한 자세한 내용은 ClassicLink 가능 VPC 및 링크된 인스턴스 보기 (p. 782) 단원을 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. 783) 단원을 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 중지하면 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 AWS, Amazon Redshift, Amazon ElastiCache 및 Elastic Load Balancing Amazon RDS 서비스에 액세스할 수 있습니다. 그러나 VPC의 인스턴스는 AWS를 사용하여 EC2-Classic 플랫폼이 프로비저닝한 ClassicLink 서비스에 액세스할 수 없습니다.

Elastic Load Balancing를 사용하는 경우 연결된 EC2-Classic 인스턴스를 로드 밸런서에 등록할 수 있습니다. ClassicLink가 활성화된 VPC에서 로드 밸런서를 생성하고 인스턴스가 실행되는 가용 영역을 활성화해야 합니다. 링크된 EC2-Classic 인스턴스를 종료하면 로드 밸런서가 인스턴스의 등록을 해제합니다.

Amazon EC2 Auto Scaling 사용 시에는 시작될 때 지정한 Amazon EC2 Auto Scaling 가능 VPC에 자동으로 링크되는 인스턴스가 포함된 ClassicLink 그룹을 생성할 수 있습니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서에서 VPC에 EC2-Classic 인스턴스 링크를 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 네트워크 인터페이스를 선택합니다.
3. 설명 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 기본 프라이빗 IPv4 IP 필드에서 프라이빗 IP 주소를 확인합니다.

ClassicLink 사용 제어

기본적으로 IAM 사용자에게는 ClassicLink 사용 권한이 없습니다. 사용자에게 VPC의 IAM를 활성화/비활성화하고, ClassicLink 가능 VPC에 인스턴스를 링크하거나 링크를 해제하고, ClassicLink 가능 VPC 및 링크된 ClassicLink 인스턴스를 조회하는 권한을 부여하는 EC2-Classic 정책을 생성할 수 있습니다. IAM의 Amazon EC2 정책에 대한 자세한 내용은 IAM에 대한 Amazon EC2 정책 (p. 806) 단원을 참조하십시오.

ClassicLink 작업 관련 정책에 대한 자세한 내용은 ClassicLink에 대한 예제 IAM 정책 (p. 784)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 제외되며, 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: 3티어 웹 애플리케이션의 ClassicLink 보안 그룹 구성 (p. 786) 단원을 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 `local`인 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 `local`인 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 테이블을 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 `10.0.0.0/8`이고 대상이 `local`인 라우팅 테이블에 정적 경로가 자동으로 추가됩니다.

자세한 내용과 예시는 Amazon VPC Peering Guide의 ClassicLink로 구성 단원을 참조하십시오.

ClassicLink 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 중지하면 자동으로 VPC에서 링크가 해제되고 VPC 보안 그룹이 인스턴스에 더 이상 연결되지 않습니다. 인스턴스를 다시 시작한 후 VPC에 다시 연결할 수 있습니다.
- 다른 리전이나 다른 EC2-Classic 계정에 속하는 VPC에는 AWS 인스턴스를 링크할 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 Guide의 내용을 참조하십시오.
- 링크된 EC2-Classic 인스턴스에는 VPC 탄력적 IP 주소를 연결할 수 없습니다.
- IPv6 통신에 대해 EC2-Classic 인스턴스를 활성화할 수 없습니다. IPv6 CIDR 블록을 VPC와 연결하고 IPv6 주소를 VPC의 리소스에 할당할 수 있지만 ClassicLinked 인스턴스와 VPC의 리소스 간 통신은 IPv4를 통해서만 이루어집니다.
- EC2-Classic 프라이빗 IP 주소 범위 `10/8`과 충돌하는 라우팅이 있는 VPC에서는 ClassicLink를 활성화할 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. 779) 단원을 참조하십시오.
- 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. 규제 또는 보안 요구 사항으로 인해 VPC의 테넌시를 `dedicated`로 설정한 경우 VPC에 EC2-Classic 인스턴스를 연결하면 공유된 테넌시 리소스가 프라이빗 IP 주소를 사용하여 격리된 리소스를 직접 처리할 수 있으므로 이러한 요구 사항을 준수하지 않을 수 있습니다. ClassicLink용 전용 VPC를 활성화하려는 경우 AWS 지원 요청에 대한 자세한 이유를 기재하십시오.
- EC2-Classic 인스턴스를 `172.16.0.0/16` 범위의 VPC에 연결하고 해당 VPC 내의 `172.16.0.23/32` IP 주소로 DNS 서버를 실행하는 경우 연결된 EC2-Classic 인스턴스는 VPC DNS 서버에 액세스할 수 없습니다. 이 문제를 해결하려면 DNS 서버를 VPC 내에서 다른 IP 주소로 실행합니다.
- ClassicLink는 VPC 외부의 전이 관계를 지원하지 않습니다. 연결된 EC2-Classic 인스턴스는 VPC에 연결된 VPN 연결, VPC 게이트웨이 엔드포인트, NAT 게이트웨이 또는 인터넷 게이트웨이에 액세스할 수 없습니다. 마찬가지로 VPN 연결의 반대편에 있는 리소스 또는 인터넷 게이트웨이는 연결된 EC2-Classic 인스턴스에 액세스할 수 없습니다.

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.

작업

- VPC에서 ClassicLink 활성화 (p. 781)

- ClassicLink가 활성화된 VPC 생성 (p. 781)
- VPC에 인스턴스 링크 (p. 781)
- 시작 시 VPC에 인스턴스 연결 (p. 782)
- ClassicLink 가능 VPC 및 링크된 인스턴스 보기 (p. 782)
- ClassicLink DNS 지원 활성화 (p. 783)
- ClassicLink DNS 지원 비활성화 (p. 783)
- VPC에서 인스턴스 연결 해제 (p. 783)
- VPC에 대해 ClassicLink 비활성화 (p. 784)

VPC에서 ClassicLink 활성화

VPC에 EC2-Classic 인스턴스를 링크하려면 우선 VPC에서 ClassicLink를 활성화해야 합니다. VPC에 ClassicLink 프라이빗 IP 주소 범위와 충돌하는 라우팅이 있는 경우 VPC에서 EC2-Classic를 활성화할 수 없습니다. 자세한 내용은 ClassicLink의 라우팅 (p. 779) 단원을 참조하십시오.

VPC에서 ClassicLink를 활성화하려면 다음을 수행합니다.

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 VPC를 선택합니다.
3. VPC를 선택한 다음 작업, ClassicLink 활성화를 선택합니다.
4. 확인 대화 상자에서 예, 활성화를 선택합니다.
5. (선택 사항) 퍼블릭 DNS 호스트 이름이 프라이빗 IP 주소로 확인되도록 하려면 인스턴스를 연결하기 전에 VPC에 대해 ClassicLink DNS 지원을 활성화합니다. 자세한 내용은 ClassicLink DNS 지원 활성화 (p. 783) 단원을 참조하십시오.

ClassicLink가 활성화된 VPC 생성

ClassicLink 콘솔에서 VPC 마법사를 사용하면 새 VPC를 생성할 때 Amazon VPC를 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 VPC 마법사 시작를 선택합니다.
3. VPC 구성 옵션 중 하나를 선택하고 선택을 선택합니다.
4. 마법사 다음 페이지의 ClassicLink 활성화에서 예를 선택합니다. 마법사의 나머지 단계를 완료하여 VPC를 생성합니다. VPC 마법사 사용에 대한 자세한 정보는 Amazon VPC 사용 설명서에서 Amazon VPC용 시나리오 단원을 참조하십시오.
5. (선택 사항) 퍼블릭 DNS 호스트 이름이 프라이빗 IP 주소로 확인되도록 하려면 인스턴스를 연결하기 전에 VPC에 대해 ClassicLink DNS 지원을 활성화합니다. 자세한 내용은 ClassicLink DNS 지원 활성화 (p. 783) 단원을 참조하십시오.

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.

퍼블릭 DNS 호스트 이름이 프라이빗 IP 주소로 확인되도록 하려면 인스턴스를 연결하기 전에 VPC에 대해 ClassicLink DNS 지원을 활성화합니다. 자세한 내용은 ClassicLink DNS 지원 활성화 (p. 783) 단원을 참조하십시오.

VPC에 인스턴스를 링크하려면 다음을 수행합니다.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 실행 중인 EC2-Classic 인스턴스를 선택한 다음, 작업, ClassicLink, VPC에 연결을 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 VPC에 연결을 선택합니다.

시작 시 VPC에 인스턴스 연결

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 인스턴스 시작을 선택합니다.
3. AMI를 선택하고 인스턴스 유형을 선택합니다. 인스턴스 세부 정보 구성 페이지의 네트워크 목록에서 EC2-Classic으로 시작을 선택합니다.

   Note

   T2 인스턴스 유형 등의 일부 인스턴스 유형은 VPC로만 시작할 수 있습니다. EC2-Classic으로 시작할 수 있는 인스턴스 유형을 선택해야 합니다.
4. VPC 링크(ClassicLink) 섹션의 VPC에 연결에서 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. 432) 단원을 참조하십시오.

ClassicLink 가능 VPC 및 링크된 인스턴스 보기

ClassicLink 콘솔에서 모든 Amazon VPC 가능 VPC를, EC2-Classic 콘솔에서 링크된 Amazon EC2 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 VPC를 선택합니다.
3. VPC를 선택하고 요약 탭에서 ClassicLink 필드를 찾습니다. 값이 활성화이면 VPC에서 ClassicLink가 활성화된 것입니다.
4. 또는 ClassicLink 열을 찾고 각 VPC에 표시된 값(활성화 또는 비활성화)을 확인합니다. 해당 열이 보이지 않는 경우 테이블 열 편집(기어 모양 아이콘)을 선택하고 ClassicLink 속성을 선택한 다음 닫기를 선택합니다.

링크된 EC2-Classic 인스턴스를 확인하려면 다음을 수행합니다.

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

2. 탐색 창에서 인스턴스를 선택합니다.
3. EC2-Classic 인스턴스를 선택하고 설명 탭에서 ClassicLink 필드를 찾습니다. 인스턴스가 VPC에 링크된 경우 필드에 인스턴스가 링크된 VPC의 ID가 표시됩니다. 인스턴스가 VPC에 링크되지 않은 경우 필드에 연결 해제됨이 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 비활성화할 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 단원을 참조하십시오.

ClassicLink DNS 지원을 활성화하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 VPC를 선택합니다.
3. VPC를 선택한 다음 작업, ClassicLink DNS 지원 편집을 선택합니다.
4. 예를 선택하여 ClassicLink DNS 지원을 활성화한 다음, 저장을 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

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 VPC를 선택합니다.
3. VPC를 선택한 다음 작업, ClassicLink DNS 지원 편집을 선택합니다.
4. 아니요를 선택하여 ClassicLink DNS 지원을 비활성화한 다음, 저장을 선택합니다.

VPC에서 인스턴스 연결 해제

ClassicLink 인스턴스와 VPC 간에 EC2-Classic 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.

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

3. 작업 목록에서 ClassicLink, 인스턴스 링크 해제를 선택합니다. 둘 이상의 인스턴스를 선택하여 동일한 VPC에서 링크를 해제할 수 있습니다.
4. 확인 대화 상자에서 예를 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.

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 VPC를 선택합니다.
3. VPC를 선택한 다음, 작업, ClassicLink 비활성화를 선택합니다.
4. 확인 대화 상자에서 예, 비활성화를 선택합니다.

ClassicLink에 대한 예제 IAM 정책

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.

예제

- ClassicLink 작업 관련 전체 권한 (p. 784)
- ClassicLink에서 VPC 활성화 및 비활성화 (p. 784)
- 인스턴스 링크 (p. 785)
- 인스턴스 링크 해제 (p. 786)

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": "*"
    }
   ]
}
```

ClassicLink에서 VPC 활성화 및 비활성화

다음 정책은 'ClassicLink' 태그가 있는 VPC에서 `purpose=classiclink`를 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 인스턴스를 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: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 "EC2-Classic" 태그가 있을 때만 VPC에서 링크된 `unlink=true` 인스턴스의 링크를 해제할 권한을 부여합니다. 두 번째 명령문에서는 VPC에서 인스턴스의 링크를 해제하는 데 필요한 VPC 리소스를 사용할 권한을 부여합니다.

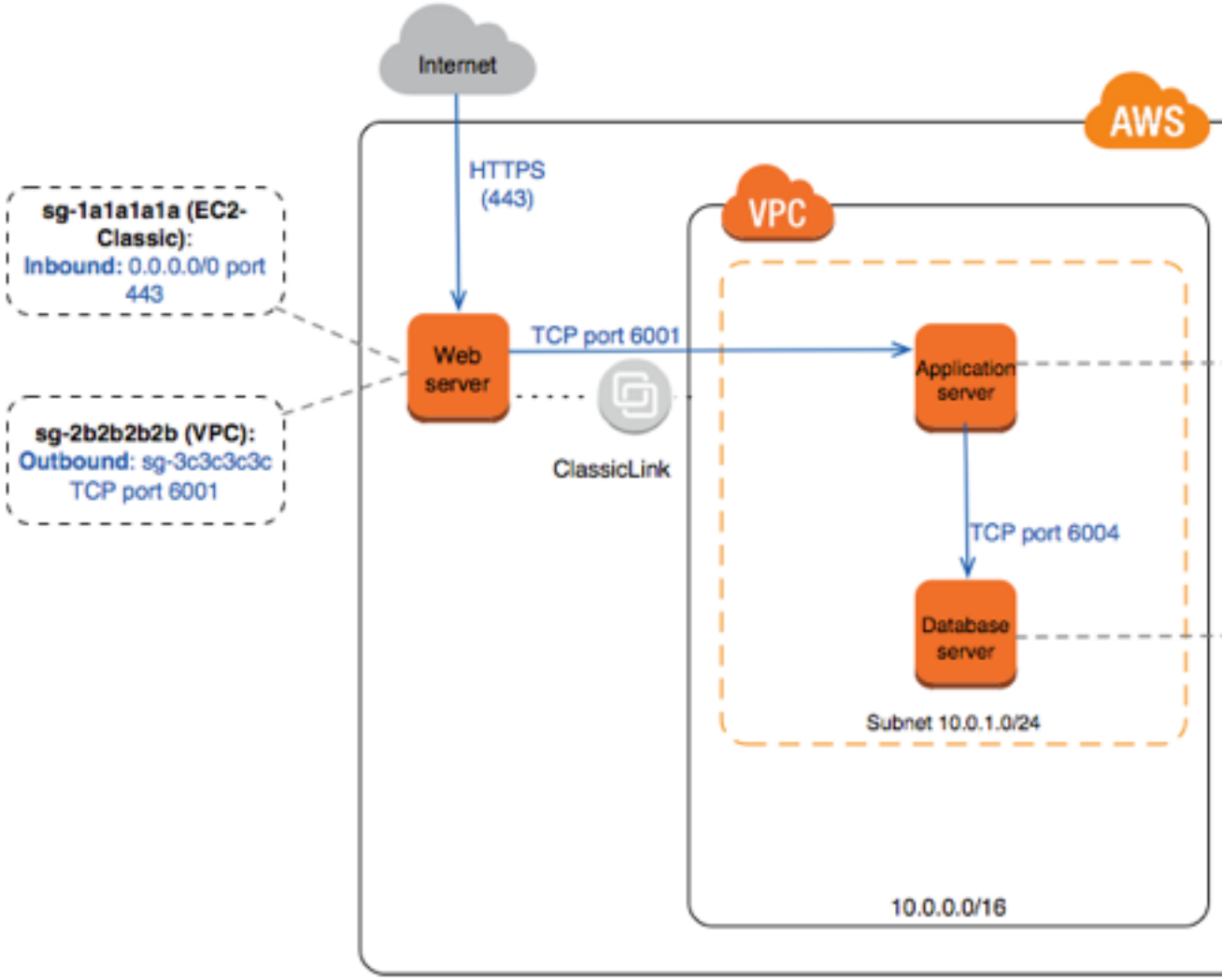
```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": "ec2:DetachClassicLinkVpc",
      "Resource": [
         "arn:aws:ec2:region:account:instance/*"
      ],
      "Condition": {
         "StringEquals": {
            "ec2:ResourceTag/unlink":"true"
         }
      }
   },
   {
      "Effect": "Allow",
      "Action": "ec2:DetachClassicLinkVpc",
      "Resource": [
         "arn:aws:ec2:region:account:vpc/*"
         ]
      }
   ]
}
```

예: 3티어 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.

Inbound			
Source	Type	Port Range	Comments
0.0.0.0/0	HTTPS	443	인터넷 트래픽의 웹 서버 전송을 허용합니다.

다음은 VPC 보안 그룹(`sg-2b2b2b2b`)에 적용되는 보안 그룹 규칙입니다.

Outbound

Destination	Type	Port Range	Comments
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.

Inbound			
Source	Type	Port Range	Comments
sg-2b2b2b2b	TCP	6001	웹 서버(또는 `sg-2b2b2b2b`와 연동되어 있는 기타 인스턴스)에서 애플리케이션 서버로 특정 유형의 트래픽을 전송할 수 있도록 허용합니다.
Outbound			
Destination	Type	Port Range	Comments
sg-4d4d4d4d	TCP	6004	애플리케이션 서버에서 데이터베이스 서버(또는 `sg-4d4d4d4d`와 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.

Inbound			
Source	Type	Port Range	Comments
sg-3c3c3c3c	TCP	6004	애플리케이션 서버(또는 `sg-3c3c3c3c`와 연동되어 있는 기타 인스턴스)에서 데이터베이스 서버로 특정 유형의 트래픽을 전송할 수 있도록 허용합니다.

Linux 내 EC2-Classic 인스턴스에서 VPC 내 Linux 인스턴스로 마이그레이션

2013년 12월 4일 이전에 AWS 계정을 생성한 경우 일부 리전에서 EC2-Classic 지원을 받을 수 있습니다. 향상된 네트워킹 및 새로운 인스턴스 유형과 같은 일부 Amazon EC2 리소스 및 기능에는 Virtual Private Cloud(VPC)가 필요합니다. 일부 리소스는 EC2-Classic 및 VPC 간에 공유될 수 있으며 일부는 공유될 수 없습니다. 자세한 내용은 EC2-Classic 및 VPC 간 리소스 공유 및 액세스 (p. 776) 단원을 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 EC2-Classic에서 VPC로 Windows 인스턴스 마이그레이션 단원을 참조하십시오.

내용

- VPC로 전체 마이그레이션 (p. 789)
- ClassicLink를 사용하여 VPC로 증분식 마이그레이션 (p. 795)

VPC로 전체 마이그레이션

다음 작업을 완료하여 EC2-Classic에서 VPC로 애플리케이션을 전체 마이그레이션합니다.

작업

- 1 단계: VPC 생성 (p. 789)
- 2단계: 보안 그룹 구성 (p. 789)
- 3단계: EC2-Classic 인스턴스에서 AMI 생성 (p. 790)
- 4단계: VPC로 인스턴스 시작 (p. 791)
- 예제: 간단한 웹 애플리케이션 마이그레이션 (p. 792)

1 단계: VPC 생성

VPC 사용을 시작하려면 계정에 VPC가 있는지 확인하십시오. 다음 방법 중 하나를 사용하여 VPC를 생성할 수 있습니다.

- AWS 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 기본 VPC 및 기본 서브넷을 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 Amazon VPC 콘솔 마법사 구성을 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 VPC 및 서브넷을 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 Amazon EC2 인스턴스에 대한 Linux 보안 그룹 (p. 872) 단원을 참조하십시오.

새 보안 그룹에 보안 그룹 규칙을 복사하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 보안 그룹을 선택합니다.
3. EC2-Classic 인스턴스와 연동되어 있는 보안 그룹을 선택한 다음 작업과 새로 복사를 차례대로 선택합니다.
4. 보안 그룹 생성 대화 상자에서 새 보안 그룹의 이름과 설명을 지정합니다. VPC 목록에서 해당 VPC를 선택합니다.
5. 인바운드 탭이 EC2-Classic 보안 그룹의 규칙으로 채워집니다. 필요에 따라 규칙을 수정할 수 있습니다. 아웃바운드 탭에는 모든 아웃바운드 트래픽을 허용하는 규칙이 자동으로 생성되어 있습니다. 보안 그룹 규칙 수정에 대한 자세한 내용은 Amazon EC2 인스턴스에 대한 Linux 보안 그룹 (p. 872) 단원을 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. Create를 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 인스턴스 페이지로 이동하고 인스턴스를 선택한 다음 설명 탭의 루트 디바이스 유형 필드에서 정보를 봅니다. 값이 `ebs`인 경우 EBS 기반 인스턴스이고, 값이 `instance-store`인 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. 95) 단원을 참조하십시오.

인스턴스 루트 디바이스 유형	작업
EBS	인스턴스에서 EBS 기반 AMI를 생성합니다. 자세한 내용은 Amazon EBS 지원 Linux AMI 생성 (p. 112) 단원을 참조하십시오.
인스턴스 스토어	AMI 도구를 사용하여 인스턴스에서 인스턴스 스토어 기반 AMI를 생성합니다. 자세한 내용은 인스턴스 스토어 기반 Linux AMI 생성 (p. 115) 단원을 참조하십시오.

인스턴스 루트 디바이스 유형	작업
인스턴스 스토어	인스턴스 스토어 지원 인스턴스를 EBS 지원 인스턴스로 변환합니다. 자세한 내용은 인스턴스 스토어 기반 AMI를 Amazon EBS 기반 AMI로 변환 (p. 126) 단원을 참조하십시오.

(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 단원을 참조하십시오.

- Amazon EBS 볼륨 (p. 890)
- Amazon EBS 볼륨 생성 (p. 908)
- Amazon EBS 볼륨을 인스턴스에 연결 (p. 910)

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 단원을 참조하십시오.

- Amazon EBS 스냅샷 (p. 926)
- Amazon EBS 스냅샷 만들기 (p. 929)
- 스냅샷에서 Amazon EBS 볼륨 복구 (p. 909)

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 인스턴스 시작을 선택합니다.
3. Amazon Machine Image(AMI) 선택 페이지에서 나의 AMI 범주를 선택하고 생성한 AMI를 선택합니다.
4. 인스턴스 유형 선택 페이지에서 인스턴스 유형을 선택하고 다음: 인스턴스 세부 정보 구성(Next: Configure Instance Details)을 선택합니다.
5. 인스턴스 세부 정보 구성 페이지의 네트워크 목록에서 VPC를 선택합니다. 서브넷 목록에서 필요한 서브넷을 선택합니다. 기타 필요한 세부 정보를 구성한 다음 보안 그룹 구성 페이지에 도달할 때까지 마법사의 다음 페이지로 이동합니다.
6. 기존 그룹 선택(Select an existing group)을 선택하고 이전에 생성한 보안 그룹을 선택합니다. 검토 및 시작을 선택합니다.
7. 인스턴스 정보를 검토한 다음 시작을 선택하여 키 페어를 지정하고 인스턴스를 시작합니다.

마법사의 각 단계에서 구성할 수 있는 파라미터에 대한 자세한 내용은 인스턴스 시작 마법사를 사용하여 인스턴스 시작 (p. 431) 단원을 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 AMI를 선택합니다.
4. 필터 목록에서 내 소유가 선택되어 있는지 확인한 다음 AMI를 선택합니다.
5. 권한 탭에서 편집을 선택합니다. 새 AWS 계정의 계정 번호를 입력하고 권한 추가와 저장을 차례대로 선택합니다.

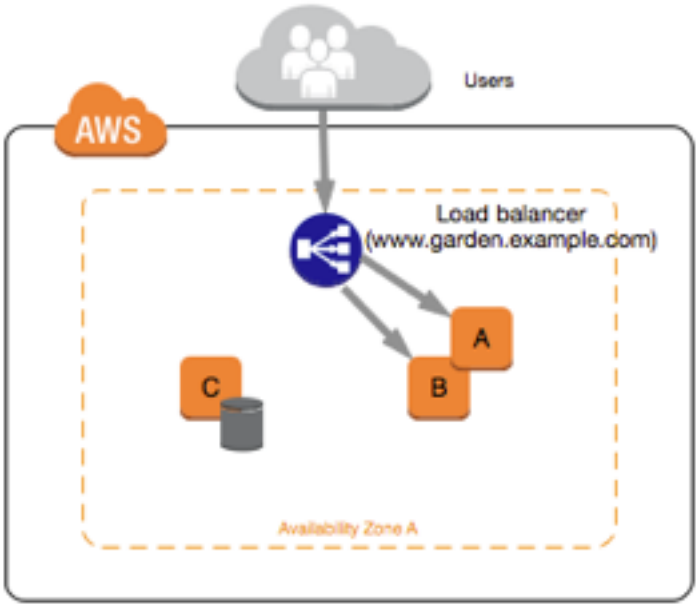
기본 VPC로 인스턴스를 시작하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 새 AWS 계정으로 전환합니다.
3. 탐색 창에서 AMI를 선택합니다.
4. 필터 목록에서 프라이빗 이미지를 선택합니다. EC2-Classic 계정에서 공유한 AMI와 시작을 차례대로 선택합니다.
5. 인스턴스 유형 선택 페이지에서 인스턴스 유형을 선택하고 다음: 인스턴스 세부 정보 구성(Next: Configure Instance Details)을 선택합니다.
6. 인스턴스 세부 정보 구성 페이지의 네트워크 목록에서 기본 VPC를 선택해야 합니다. 기타 필요한 세부 정보를 구성한 다음 보안 그룹 구성 페이지에 도달할 때까지 마법사의 다음 페이지로 이동합니다.
7. 기존 그룹 선택(Select an existing group)을 선택하고 이전에 생성한 보안 그룹을 선택합니다. 검토 및 시작을 선택합니다.
8. 인스턴스 정보를 검토한 다음 시작을 선택하여 키 페어를 지정하고 인스턴스를 시작합니다.

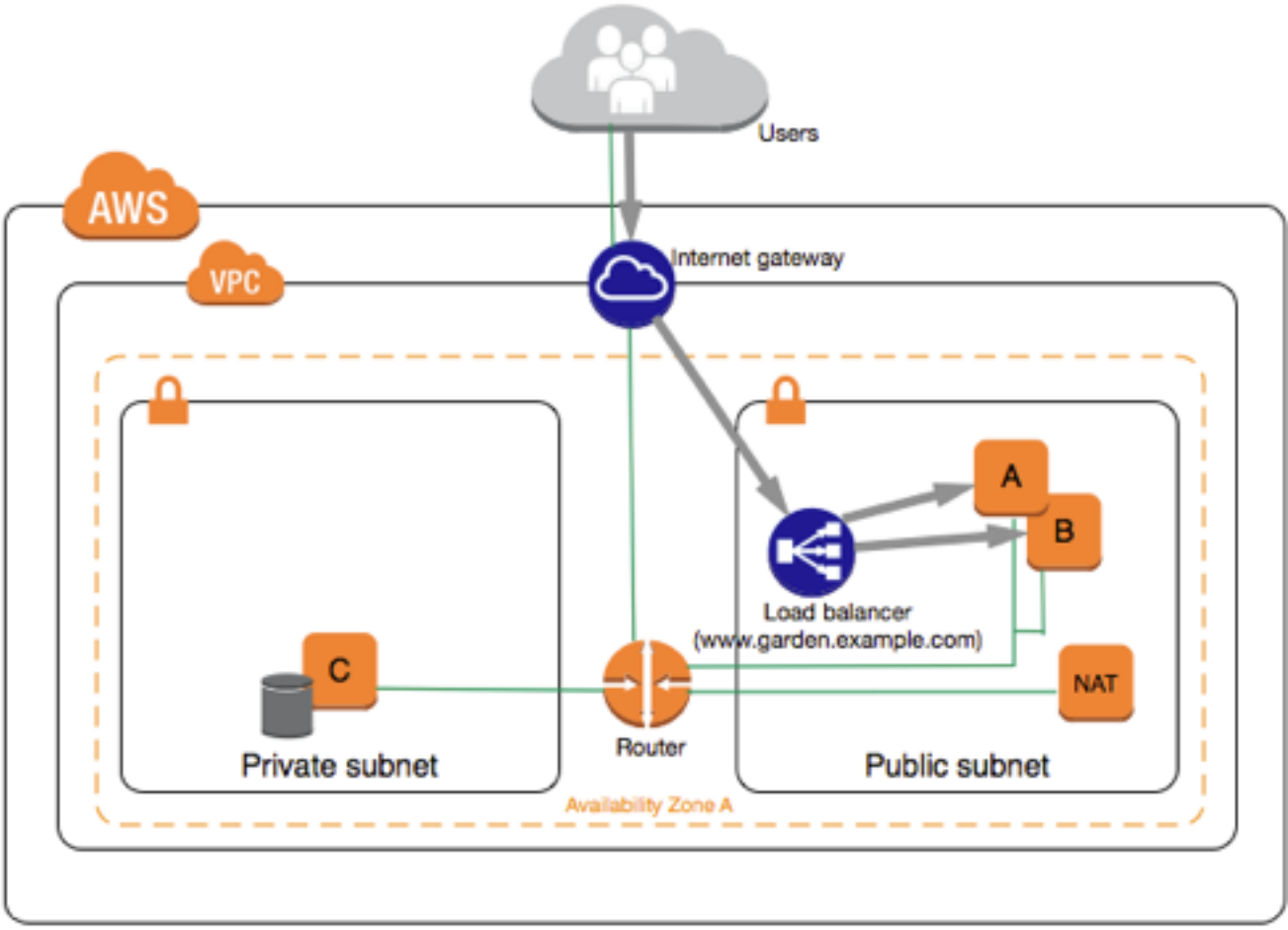
마법사의 각 단계에서 구성할 수 있는 파라미터에 대한 자세한 내용은 인스턴스 시작 마법사를 사용하여 인스턴스 시작 (p. 431) 단원을 참조하십시오.

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 Elastic Load Balancing를 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.com`을 등록하고 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 VPC로 마이그레이션하고 퍼블릭 서브넷에서 로드 밸런서를 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 퍼블릭 및 프라이빗 서브넷이 있는 VPC(NAT)를 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. 790) 단원을 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. 789) 단원을 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. 791) 단원을 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 정보는 NAT 인스턴스 단원을 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 VPC로 마이그레이션합니다. 마이그레이션이 완료되면 탄력적 IP 주소를 VPC의 새로운 웹 서버와 연동시킵니다. 자세한 내용은 EC2-Classic에서 탄력적 IP 주소 마이그레이션 (p. 774) 단원을 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 정보는 Amazon VPC의 Elastic Load Balancing 단원을 참조하십시오.
- DNS 레코드 업데이트: 퍼블릭 서브넷에서 로드 밸런서를 설정한 후에는 `www.garden.example.com` 도메인이 새로운 로드 밸런서를 가리키는지 확인해야 합니다. 이렇게 하려면 Route 53에서 DNS 레코드를 업데이트하고 별칭 레코드 세트를 업데이트해야 합니다. Route 53 사용에 대한 자세한 내용은 Route 53 시작하기를 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 .

For more information about ClassicLink, see ClassicLink (p. 777).

주제

- 1단계: 마이그레이션 시퀀스 준비 (p. 795)
- 2 단계: VPC 생성 (p. 796)
- 3단계: ClassicLink에 대해 VPC 활성화 (p. 796)
- 4단계: EC2-Classic 인스턴스에서 AMI 생성 (p. 796)
- 5단계: VPC로 인스턴스 시작 (p. 797)
- 6단계: VPC에 EC2-Classic 인스턴스 연결 (p. 798)
- 7단계: VPC 마이그레이션 완료 (p. 798)

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 Amazon VPC 콘솔 마법사 구성을 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 VPC 및 서브넷을 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. For more information about ClassicLink, see ClassicLink (p. 777).

VPC에서 ClassicLink를 활성화하려면 다음을 수행합니다.

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 VPC를 선택합니다.
3. VPC를 선택한 다음 작업 목록에서 ClassicLink 활성화를 선택합니다.
4. 확인 대화 상자에서 예, 활성화를 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 인스턴스 페이지로 이동하고 인스턴스를 선택한 다음 설명 탭의 루트 디바이스 유형 필드에서 정보를 봅니다. 값이 `ebs`인 경우 EBS 기반 인스턴스이고, 값이 `instance-store`인 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. 95) 단원을 참조하십시오.

인스턴스 루트 디바이스 유형	작업
EBS	인스턴스에서 EBS 기반 AMI를 생성합니다. 자세한 내용은 Amazon EBS 지원 Linux AMI 생성 (p. 112) 단원을 참조하십시오.

인스턴스 루트 디바이스 유형	작업
인스턴스 스토어	AMI 도구를 사용하여 인스턴스에서 인스턴스 스토어 기반 AMI를 생성합니다. 자세한 내용은 인스턴스 스토어 기반 Linux AMI 생성 (p. 115) 단원을 참조하십시오.
인스턴스 스토어	인스턴스 스토어 지원 인스턴스를 EBS 지원 인스턴스로 변환합니다. 자세한 내용은 인스턴스 스토어 기반 AMI를 Amazon EBS 기반 AMI로 변환 (p. 126) 단원을 참조하십시오.

(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 단원을 참조하십시오.

- Amazon EBS 볼륨 (p. 890)
- Amazon EBS 볼륨 생성 (p. 908)
- Amazon EBS 볼륨을 인스턴스에 연결 (p. 910)

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 단원을 참조하십시오.

- Amazon EBS 스냅샷 (p. 926)
- Amazon EBS 스냅샷 만들기 (p. 929)
- 스냅샷에서 Amazon EBS 볼륨 복구 (p. 909)

5단계: VPC로 인스턴스 시작

마이그레이션 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 인스턴스 시작을 선택합니다.
3. Amazon Machine Image(AMI) 선택 페이지에서 나의 AMI 범주를 선택하고 생성한 AMI를 선택합니다.
4. 인스턴스 유형 선택 페이지에서 인스턴스 유형을 선택하고 다음: 인스턴스 세부 정보 구성(Next: Configure Instance Details)을 선택합니다.
5. 인스턴스 세부 정보 구성 페이지의 네트워크 목록에서 VPC를 선택합니다. 서브넷 목록에서 필요한 서브넷을 선택합니다. 기타 필요한 세부 정보를 구성한 다음 보안 그룹 구성 페이지에 도달할 때까지 마법사의 다음 페이지로 이동합니다.
6. 기존 그룹 선택(Select an existing group)을 선택하고 이전에 생성한 보안 그룹을 선택합니다. 검토 및 시작을 선택합니다.
7. 인스턴스 정보를 검토한 다음 시작을 선택하여 키 페어를 지정하고 인스턴스를 시작합니다.

마법사의 각 단계에서 구성할 수 있는 파라미터에 대한 자세한 내용은 인스턴스 시작 마법사를 사용하여 인스턴스 시작 (p. 431) 단원을 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 인스턴스를 선택합니다.
3. EC2-Classic 인스턴스를 선택한 다음 작업, ClassicLink 및 VPC에 연결을 차례대로 선택합니다.

   Note

   인스턴스가 `running` 상태인지 확인합니다.
4. 대화 상자에서 ClassicLink 지원 VPC를 선택합니다(ClassicLink에 대해 활성화된 VPC만 표시됨).
5. VPC의 보안 그룹을 하나 이상 선택하여 인스턴스와 연결합니다. 완료되면 VPC에 연결을 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 VPC 플랫폼으로 마이그레이션할 수 있습니다. 자세한 내용은 EC2-Classic에서 탄력적 IP 주소 마이그레이션 (p. 774) 단원을 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 EC2의 보안

AWS에서 클라우드 보안을 가장 중요하게 생각합니다. AWS 고객은 보안에 가장 보안에 민감한 조직의 요구 사항에 부합하도록 구축된 데이터 센터 및 네트워크 아키텍처의 혜택을 누릴 수 있습니다.

보안은 AWS와 귀하의 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드의 보안 – AWS는 AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호합니다. AWS는 또한 안전하게 사용할 수 있는 서비스를 제공합니다. 타사 감사자는 AWS 규정 준수 프로그램의 일환으로 보안 효과를 정기적으로 테스트하고 검증합니다. Amazon EC2에 적용되는 규정 준수 프로그램에 대해 자세히 알아보려면 규정 준수 프로그램 제공 AWS 범위 내 서비스를 참조하십시오.
- 클라우드 내 보안 – 귀하의 책임은 귀하가 사용하는 AWS 서비스에 의해 결정됩니다. 또한 귀하는 데이터의 민감도, 회사 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 Amazon EC2를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 보안 및 규정 준수 목표에 맞게 Amazon EC2를 구성하는 방법을 보여줍니다. 또한 Amazon EC2 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법을 알아봅니다.

목차

- Amazon EC2의 인프라 보안 (p. 799)
- Amazon EC2 및 인터페이스 VPC 엔드포인트 (p. 800)
- Amazon EC2의 복원성 (p. 802)
- Amazon EC2의 데이터 보호 (p. 802)
- Amazon EC2에 대한 자격 증명 및 액세스 관리 (p. 803)
- Amazon EC2 키 페어 (p. 861)
- Amazon EC2 인스턴스에 대한 Linux 보안 그룹 (p. 872)
- Amazon EC2의 업데이트 관리 (p. 885)
- Amazon EC2에 대한 규정 준수 확인 (p. 885)

Amazon EC2의 인프라 보안

관리형 서비스인 Amazon EC2는 Amazon Web Services: 보안 프로세스 개요 백서에 설명된 AWS 글로벌 네트워크 보안 절차로 보호됩니다.

AWS에서 게시한 API 호출을 사용하여 네트워크를 통해 Amazon EC2에 액세스합니다. 클라이언트가 TLS(전송 계층 보안) 1.0 이상을 지원해야 합니다. TLS 1.2 이상을 권장합니다. 클라이언트는 Ephemeral Diffie-Hellman(DHE) 또는 Elliptic Curve Ephemeral Diffie-Hellman(ECDHE)과 같은 PFS(전달 완전 보안, Perfect Forward Secrecy)가 포함된 암호 제품군도 지원해야 합니다. Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

또한 요청은 액세스 키 ID 및 IAM 주체와 연결된 보안 액세스 키를 사용해 서명해야 합니다. 또는 AWS Security Token Service(AWS STS)를 사용하여 임시 보안 자격 증명을 생성하여 요청에 서명할 수 있습니다.

네트워크 격리

Virtual Private Cloud(VPC)는 AWS 클라우드에서 논리적으로 격리된 고유한 영역의 가상 네트워크입니다. 별도의 VPC를 사용하여 워크로드별 또는 조직체별로 인프라를 격리합니다.

서브넷은 VPC의 IP 주소 범위입니다. 인스턴스를 시작할 때 VPC의 서브넷에서 인스턴스를 시작합니다. 서브넷을 사용하여 단일 VPC 내의 애플리케이션 티어(예: 웹, 애플리케이션 및 데이터베이스)를 격리합니다. 인터넷에서 직접 액세스하면 안 되는 경우 프라이빗 서브넷을 인스턴스에 사용합니다.

퍼블릭 인터넷을 통해 트래픽을 보내지 않고 VPC에서 Amazon EC2 API를 호출하려면 AWS PrivateLink를 사용합니다.

물리적 호스트에서 격리

동일한 물리적 호스트에 있는 다양한 EC2 인스턴스는 개별 물리적 호스트에 있는 것처럼 서로 격리됩니다. 하이퍼바이저는 CPU와 메모리를 격리하며, 인스턴스에는 원시 디스크 디바이스에 대한 액세스 대신 가상화된 디스크가 제공됩니다.

인스턴스를 중지하거나 종료하면 인스턴스에 할당된 메모리는 새 인스턴스에 할당되기 전에 하이퍼바이저에서 스크러빙(0으로 설정)되며 스토리지의 모든 블록은 재설정됩니다. 이 방법으로 데이터가 의도하지 않게 다른 인스턴스에 노출되지 않도록 보호합니다.

네트워크 MAC 주소는 AWS 네트워크 인프라에서 인스턴스에 동적으로 할당됩니다. IP 주소는 AWS 네트워크 인프라에서 인스턴스에 동적으로 할당되거나, 인증된 API 요청을 통해 EC2 관리자가 할당합니다. AWS 네트워크에서 인스턴스는 할당된 MAC 및 IP 주소에서만 트래픽을 전송할 수 있습니다. 그렇지 않으면 트래픽이 끊어집니다.

기본적으로 인스턴스는 특정하게 주소 지정되지 않은 트래픽을 수신할 수 없습니다. 인스턴스에서 NAT(Network Address Translation), 라우팅 또는 방화벽 서비스를 실행해야 하는 경우 네트워크 인터페이스에 대한 소스/대상 확인을 비활성화할 수 있습니다.

네트워크 트래픽 제어

EC2 인스턴스의 네트워크 트래픽을 제어하기 위해 다음 옵션을 고려해 보십시오.

- 보안 그룹 (p. 872)을 사용하여 인스턴스에 대한 액세스를 제한합니다. 예를 들어, 기업 네트워크의 주소 범위에서만 트래픽을 허용할 수 있습니다.
- 인터넷에서 직접 액세스하면 안 되는 경우 프라이빗 서브넷을 인스턴스에 사용합니다. 프라이빗 서브넷에 있는 인스턴스에서 인터넷에 액세스하려면 배스천 호스트 또는 NAT 게이트웨이를 사용합니다.
- AWS Virtual Private Network 또는 AWS Direct Connect를 사용하여 원격 네트워크에서 VPC로 프라이빗 연결을 설정합니다. 자세한 내용은 네트워크-Amazon VPC 연결 옵션을 참조하십시오.
- VPC 흐름 로그를 사용하여 인스턴스에 도달하는 트래픽을 모니터링합니다.
- AWS Security Hub를 사용하여 인스턴스에서 의도하지 않게 네트워크에 액세스할 수 있는지 확인합니다.
- EC2 Instance Connect (p. 490)를 사용하여 SSH 키를 공유하고 관리할 필요 없이 SSH(Secure Shell)를 통해 인스턴스에 연결합니다.
- AWS 시스템 관리자 Session Manager를 사용하여 인바운드 SSH 포트를 열고 SSH 키를 관리하는 대신 원격으로 인스턴스에 액세스합니다.
- AWS 시스템 관리자 Run Command를 사용하여 인바운드 SSH 포트를 열고 SSH 키를 관리하는 대신 공통 관리 작업을 자동화합니다.

Amazon VPC에서는 각 Amazon EC2 인스턴스에 대한 네트워크 액세스를 제한하는 것 외에도 인라인 게이트웨이, 프록시 서버 및 다양한 네트워크 모니터링 옵션과 같은 네트워크 보안 제어를 추가로 구현할 수 있습니다.

자세한 내용은 AWS 보안 모범 사례 백서를 참조하십시오.

Amazon EC2 및 인터페이스 VPC 엔드포인트

인터페이스 VPC 엔드포인트를 사용하도록 Amazon EC2를 구성하여 VPC의 보안 상태를 향상시킬 수 있습니다. 인터페이스 엔드포인트는 VPC와 Amazon EC2 사이의 모든 네트워크 트래픽을 Amazon 네트워크로

제한하여 Amazon EC2 API에 비공개로 액세스할 수 있는 AWS PrivateLink 기술로 구동됩니다. 인터페이스 엔드포인트를 사용하면 인터넷 게이트웨이, NAT 디바이스 또는 가상 프라이빗 게이트웨이가 필요 없습니다.

AWS PrivateLink를 구성하는 것이 필수는 아니지만 구성하는 것이 좋습니다. AWS PrivateLink 및 VPC 엔드포인트에 대한 자세한 내용은 인터페이스 VPC 엔드포인트(AWS PrivateLink)를 참조하십시오.

주제

- 인터페이스 VPC 엔드포인트 생성 (p. 801)
- 인터페이스 VPC 엔드포인트 정책 생성 (p. 801)

인터페이스 VPC 엔드포인트 생성

다음 서비스 이름을 사용하여 Amazon EC2에 대한 엔드포인트를 생성합니다.

- **`com.amazonaws.`*`region`*`.ec2`** — Amazon EC2 API 작업에 대한 엔드포인트를 생성합니다.

자세한 내용은 Amazon VPC 사용 설명서의 인터페이스 엔드포인트 생성을 참조하십시오.

인터페이스 VPC 엔드포인트 정책 생성

VPC 엔드포인트에 정책을 연결하여 Amazon EC2 API에 대한 액세스를 제어할 수 있습니다. 이 정책은 다음을 지정합니다.

- 작업을 수행할 수 있는 보안 주체.
- 수행할 수 있는 작업.
- 작업을 수행할 수 있는 리소스

> **Important**
>
> Amazon EC2에 대한 인터페이스 VPC 엔드포인트에 기본값이 아닌 정책이 적용되면 `RequestLimitExceeded` 실패와 같은 특정 API 요청 실패가 AWS CloudTrail 또는 Amazon CloudWatch에 기록되지 않을 수 있습니다.

자세한 내용은 Amazon VPC 사용 설명서의 VPC 엔드포인트를 통해 서비스에 대한 액세스 제어를 참조하십시오.

다음 예는 암호화되지 않은 볼륨을 생성하거나 암호화되지 않은 볼륨으로 인스턴스를 시작할 수 있는 권한을 거부하는 VPC 엔드포인트 정책을 보여 줍니다. 또한 이 정책 예에서는 다른 모든 Amazon EC2 작업을 수행할 수 있는 권한을 부여합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Action": "ec2:*",
        "Effect": "Allow",
        "Resource": "*"
    },
    {
        "Action": [
            "ec2:CreateVolume"
        ],
        "Effect": "Deny",
        "Resource": "*",
```

```
        "Condition": {
            "Bool": {
                "ec2:Encrypted": "false"
            }
        }
    },
    {
        "Action": [
            "ec2:RunInstances"
        ],
        "Effect": "Deny",
        "Resource": "*",
        "Condition": {
            "Bool": {
                "ec2:Encrypted": "false"
            }
        }
    }]
}
```

Amazon EC2의 복원성

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 중심으로 구축됩니다. 리전은 물리적으로 분리되고 격리된 다수의 가용 영역을 제공하며 이러한 가용 영역은 짧은 지연 시간, 높은 처리량 및 높은 중복성을 갖춘 네트워크를 통해 연결되어 있습니다. 가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS 리전 및 가용 영역에 대한 자세한 내용은 AWS 글로벌 인프라를 참조하십시오.

AWS 글로벌 인프라 외에도 Amazon EC2는 데이터 복원성을 지원하기 위해 다음과 같은 기능을 제공합니다.

- 리전 간 AMI 복사
- 리전 간 EBS 스냅샷 복사
- Amazon 데이터 수명 주기 관리자를 사용하여 EBS 스냅샷 자동화
- Amazon EC2 Auto Scaling을 사용하여 플릿의 상태 및 가용성 유지 관리
- Elastic Load Balancing을 사용하여 단일 가용 영역 또는 여러 가용 영역의 여러 인스턴스 간에 수신 트래픽 분산

Amazon EC2의 데이터 보호

Amazon Elastic Compute Cloud(Amazon EC2)는 AWS 공동 책임 모델을 준수하며, 여기에는 데이터 보호에 대한 규정과 지침이 포함됩니다. AWS는 모든 AWS 서비스를 실행하는 글로벌 인프라를 보호할 책임이 있습니다. AWS는 고객 콘텐츠 및 개인 데이터를 처리하기 위한 보안 구성 제어를 포함하여 이 인프라에서 호스팅되는 데이터에 대한 제어를 유지합니다. 데이터 컨트롤러 또는 데이터 프로세서의 역할을 담당하는 AWS 고객과 APN 파트너는 AWS 클라우드에 입력한 모든 개인 데이터에 대해 책임이 있습니다.

데이터 보호를 위해 AWS 계정 자격 증명을 보호하고 AWS Identity and Access Management (IAM)을 사용하여 개별 사용자 계정을 설정하여 각 사용자에게 직무를 수행하는 데 필요한 권한만 부여하는 것이 좋습니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정마다 멀티 팩터 인증(MFA)을 사용합니다.

- TLS를 사용하여 AWS 리소스와 통신합니다.
- AWS CloudTrail로 API 및 사용자 활동 로깅을 설정합니다.
- AWS 암호화 솔루션을 AWS 서비스 내의 모든 기본 보안 컨트롤과 함께 사용합니다.
- Amazon S3에 저장된 개인 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용합니다.

함수 이름 및 태그와 같은 메타데이터나 자유 형식 필드에 고객 계정 번호 등의 중요 식별 정보를 절대 입력하지 마십시오. 메타데이터에 입력하는 모든 데이터는 진단 로그에 포함하기 위해 선택될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 자격 증명 정보를 URL에 포함시키지 마십시오.

데이터 보호에 대한 자세한 내용은 AWS 보안 블로그에서 AWS 공동 책임 모델 및 GDPR 블로그 게시물을 참조하십시오.

저장 데이터 암호화

Amazon EBS 암호화는 EBS 볼륨과 스냅샷에 대한 암호화 솔루션입니다. 이 암호화는 AWS Key Management Service(AWS KMS) 고객 마스터 키(CMK)를 사용합니다. 자세한 내용은 Amazon EBS Encryption (p. 968) 단원을 참조하십시오.

인스턴스의 하드웨어 모듈에 구현된 XTS-AES-256 암호를 사용하여 NVMe 인스턴스 스토어 볼륨의 데이터를 암호화합니다. 하드웨어 모듈을 사용하여 암호화 키를 생성하며, 암호화 키는 각 NVMe 인스턴스 스토리지 디바이스에 고유합니다. 인스턴스가 중지되거나 종료되면 모든 암호화 키가 손상되어 복구가 불가능해집니다. 이 암호화를 비활성화할 수 없으며, 사용자 자신의 암호화 키를 제공할 수 없습니다.

전송 중 데이터 암호화

AWS는 모든 유형의 EC2 인스턴스 간 보안 프라이빗 연결을 제공합니다. 또한 256비트 암호화의 AEAD 알고리즘을 사용하여 동일한 VPC 또는 피어 VPC의 지원되는 인스턴스 간 전송 중인 트래픽을 자동으로 암호화합니다. 이 암호화 기능은 기본 하드웨어의 오프라인 기능을 사용하며 네트워크 성능에는 영향을 주지 않습니다. C5n, G4, I3en, M5dn, M5n, P3dn, R5dn 및 R5n 인스턴스는 추가적인 전송 중 트래픽 암호화를 지원합니다.

SSH는 Linux 인스턴스에 대한 원격 액세스를 위해 보안 통신 채널을 제공합니다. AWS 시스템 관리자 Session Manager 및 Run Command를 사용하는 인스턴스에 대한 원격 액세스는 TLS 1.2를 사용하여 암호화되며, 연결을 생성하기 위한 요청은 SigV4를 사용하여 서명됩니다.

TLS(전송 계층 보안)와 같은 암호화 프로토콜을 사용하여 클라이언트와 인스턴스 간에 전송 중인 민감한 데이터를 암호화합니다.

Amazon EC2에 대한 자격 증명 및 액세스 관리

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.

내용

- 인스턴스에 대한 네트워크 액세스 (p. 804)

- Amazon EC2 권한 속성 (p. 804)
- IAM 및 Amazon EC2 (p. 804)
- IAM에 대한 Amazon EC2 정책 (p. 806)
- IAM의 Amazon EC2 역할 (p. 850)
- Linux 인스턴스의 인바운드 트래픽 권한 부여 (p. 859)

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. 859) 단원을 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. 101) 단원을 참조하십시오.

각 Amazon EBS 스냅샷에는 스냅샷을 사용할 수 있는 AWS 계정을 제어하는 `createVolumePermission` 속성이 있습니다. 자세한 내용은 Amazon EBS 스냅샷 공유 (p. 938) 단원을 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?
- IAM에서 작업을 수행하려면 어떠한 Amazon EC2 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.

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 그룹을 선택한 다음, 새 그룹 생성을 선택합니다.
3. 그룹 이름에서 그룹 이름을 입력한 다음, 다음 단계를 선택합니다.
4. 정책 연결 페이지에서 AWS 관리형 정책을 선택한 후 다음 단계를 선택합니다. 예를 들어 Amazon EC2의 경우 다음 AWS 관리형 정책 중 하나가 적합할 수 있습니다.
   - PowerUserAccess
   - ReadOnlyAccess
   - AmazonEC2FullAccess
   - AmazonEC2ReadOnlyAccess
5. 그룹 생성을 선택합니다.

그룹 이름 아래에 새 그룹이 나열됩니다.

IAM 사용자를 생성하고, 그룹에 사용자를 추가하고, 사용자의 암호를 생성하려면 다음을 수행합니다.

1. 탐색 창에서 사용자를 선택한 다음 사용자 추가를 선택합니다.
2. User name에 사용자 이름을 입력합니다.
3. 액세스 유형(Access type)에서 프로그래밍 방식 액세스(Programmatic access) 및 AWS Management 콘솔 액세스를 모두 선택합니다.
4. 콘솔 암호(Console password)의 경우 다음 중 하나를 선택합니다.
   - 자동 생성된 암호(Autogenerated password). 각 사용자는 현재 유효한 암호 정책(있는 경우)에 따라 임의로 생성되는 암호를 받습니다. Final(최종) 페이지에 이르면 암호를 보거나 다운로드할 수 있습니다.
   - Custom password(사용자 지정 비밀 번호). 입력란에 입력하는 암호가 각 사용자에게 할당됩니다.
5. Next: Permissions(다음: 권한)를 선택합니다.
6. 권한 설정 페이지에서 그룹에 사용자 추가를 선택합니다. 이전에 생성한 그룹 옆에 있는 확인란을 선택하고 다음: 검토(Next: Review)를 선택합니다.
7. Create user를 선택합니다.
8. 사용자의 액세스 키(액세스 키 ID와 보안 액세스 키)를 보려면 보고자 하는 각 암호와 보안 액세스 키 옆에 있는 표시를 선택합니다. 액세스 키를 저장하려면 .csv 다운로드(Download .csv)를 선택한 후 안전한 위치에 파일을 저장합니다.

   > **Important**
   >
   > 이 단계를 완료한 후에는 보안 액세스 키를 검색할 수 없으며, 키를 분실한 경우 새로 생성해야 합니다.
9. 닫기를 선택합니다.
10. 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.

- IAM에 대한 Amazon EC2 정책 (p. 806)
- IAM의 Amazon EC2 역할 (p. 850)
- AWS Identity and Access Management (IAM)
- IAM 사용 설명서

IAM에 대한 Amazon EC2 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 정책을 참조하십시오. 사용자 지정 IAM 정책 관리 및 생성에 대한 자세한 내용은 IAM 정책 관리 단원을 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 사용자가 해당 작업에 사용할 수 있는 리소스를 별도로 지정할 수 없습니다. 대신 사용자에게 해당 작업에 대해 모든 리소스 작업을 할 수 있도록 허용해야 합니다.

작업	주제
정책의 기본적인 구조 이해	정책 구문 (p. 807)
정책의 작업 정의	Amazon EC2 작업 (p. 807)
정책의 특정 리소스 정의	Amazon EC2의 Amazon 리소스 이름 (ARN) (p. 808)
리소스 사용에 조건 적용	Amazon EC2의 조건 키 (p. 809)
Amazon EC2에서 사용 가능한 리소스 수준 권한 작업	Amazon EC2에 사용되는 작업, 리소스 및 조건 키(IAM 사용 설명서)
정책 테스트	사용자에게 필요한 권한이 있는지 확인 (p. 810)
CLI 또는 SDK용 예제 정책	AWS CLI 또는 AWS SDK 작업을 위한 정책 예제 (p. 813)
Amazon EC2 콘솔용 예제 정책	Amazon EC2 콘솔 작업을 위한 예제 정책 (p. 843)

정책 구조

다음 항목에서는 IAM 정책의 구조에 대해 설명합니다.

내용

- 정책 구문 (p. 807)

- Amazon EC2 작업 (p. 807)
- Amazon EC2 API 작업에 지원되는 리소스 수준 권한 (p. 808)
- Amazon EC2의 Amazon 리소스 이름(ARN) (p. 808)
- Amazon EC2의 조건 키 (p. 809)
- 사용자에게 필요한 권한이 있는지 확인 (p. 810)

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.

- 효과: effect는 `Allow` 또는 `Deny`일 수 있습니다. 기본적으로 IAM 사용자에게는 리소스 및 API 작업을 사용할 권한이 없으므로 모든 요청이 거부됩니다. 명시적 허용은 기본 설정을 무시합니다. 명시적 거부는 모든 허용을 무시합니다.
- Action: action은 권한을 부여하거나 거부할 특정 API 작업입니다. 작업을 지정하는 방법에 대한 자세한 내용은 Amazon EC2 작업 (p. 807) 단원을 참조하십시오.
- 리소스: 작업의 영향을 받는 리소스입니다. 일부 Amazon EC2 API 작업의 경우 작업이 생성하거나 수정할 수 있는 리소스를 정책에 구체적으로 포함할 수 있습니다. Amazon 리소스 이름(ARN)을 사용하거나 명령문이 모든 리소스에 적용됨을 표시하는 와일드카드(*)를 사용하여 리소스를 지정합니다. 자세한 내용은 Amazon EC2 API 작업에 지원되는 리소스 수준 권한 (p. 808) 단원을 참조하십시오.
- 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. 809) 단원을 참조하십시오.

IAM용 예제 Amazon EC2 정책 명령문에 대한 자세한 내용은 AWS CLI 또는 AWS SDK 작업을 위한 정책 예제 (p. 813) 단원을 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 시작되는 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 작업을 참조하십시오.

Amazon EC2 API 작업에 지원되는 리소스 수준 권한

리소스 수준 권한은 사용자가 작업을 수행할 수 있는 리소스를 지정할 수 있는 기능입니다. Amazon EC2는 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.

IAM 정책 설명에서 리소스를 지정하려면 Amazon 리소스 이름(ARN)을 사용합니다. ARN 값을 지정하는 방법에 대한 자세한 내용은 Amazon EC2의 Amazon 리소스 이름(ARN) (p. 808) 단원을 참조하십시오. API 작업이 개별 ARN을 지원하지 않는 경우 와일드카드(*)를 사용하여 모든 리소스가 작업의 영향을 받을 수 있도록 지정해야 합니다.

리소스 수준 권한을 지원하는 Amazon EC2 API 작업과, 정책에서 사용할 수 있는 ARN 및 조건 키를 식별하는 테이블을 보려면 IAM 사용 설명서의 Amazon EC2에 사용되는 작업, 리소스 및 조건 키를 참조하십시오. Amazon EC2에 대한 조건 키는 이후 단원에서 자세히 설명합니다.

Amazon EC2 API 작업에 사용하는 IAM 정책에는 태그 기반의 리소스 수준 권한을 적용할 수 있습니다. 이를 통해 사용자가 생성, 수정 또는 사용할 수 있는 리소스를 더욱 정확하게 제어할 수 있습니다. 자세한 내용은 생성 시 리소스 태그 지정에 대한 권한 부여 (p. 811) 단원을 참조하십시오.

Amazon EC2의 Amazon 리소스 이름(ARN)

각 IAM 정책 명령문은 ARN을 사용하여 지정한 리소스에 적용됩니다.

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"
```

다음과 같이 * 와일드카드를 사용하여 특정 계정에 속하는 모든 인스턴스를 지정할 수도 있습니다.

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*"
```

다음과 같이 * 와일드카드를 사용하여 특정 계정에 속하는 모든 Amazon EC2 리소스를 지정할 수도 있습니다.

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:*"
```

모든 리소스를 지정해야 하거나 특정 API 작업이 ARN을 지원하지 않는 경우 다음과 같이 `Resource` 요소에 * 와일드카드를 사용합니다.

```
"Resource": "*"
```

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. 단일 명령문에서 여러 리소스를 지정하려면 다음과 같이 각 ARN을 쉼표로 구분합니다.

```
"Resource": ["arn1", "arn2"]
```

Amazon EC2 리소스에 대한 ARN 목록은 IAM 사용 설명서의 Amazon EC2에서 정의한 리소스 유형을 참조하십시오.

Amazon EC2의 조건 키

정책 명령문에서 정책이 적용되는 시점을 제어하는 조건을 지정할 수 있습니다. 각 조건에는 하나 이상의 키-값 페어가 포함됩니다. 조건 키는 대/소문자를 구분하지 않습니다. AWS 전체 범위 조건 키 및 추가적인 서비스별 조건 키가 정의되어 있습니다.

Amazon EC2에 대한 서비스별 조건 키 목록은 IAM 사용 설명서에서 Amazon EC2의 조건 키를 참조하십시오. Amazon EC2는 AWS 전체 범위 조건 키도 구현합니다. 자세한 내용은 IAM 사용 설명서의 모든 요청에서 사용 가능한 정보를 참조하십시오.

IAM 정책에 조건 키를 사용하려면 `Condition` 명령문을 사용합니다. 예를 들어 다음 정책은 사용자에게 임의의 보안 그룹에 대한 인바운드 및 아웃바운드 규칙을 추가하고 제거하는 권한을 부여합니다. `ec2:Vpc` 조건 키를 사용하여 특정 VPC의 보안 그룹에서만 이러한 작업을 수행할 수 있도록 지정합니다.

```
{
"Version": "2012-10-17",
  "Statement":[{
    "Effect":"Allow",
    "Action": [
       "ec2:AuthorizeSecurityGroupIngress",
       "ec2:AuthorizeSecurityGroupEgress",
       "ec2:RevokeSecurityGroupIngress",
       "ec2:RevokeSecurityGroupEgress"],
     "Resource": "arn:aws:ec2:region:account:security-group/*",
      "Condition": {
        "StringEquals": {
          "ec2:Vpc": "arn:aws:ec2:region:account:vpc/vpc-11223344556677889"
        }
      }
    }
  ]
```

```
}
```

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 정책 변수를 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 조건 단원을 참조하십시오.

모든 Amazon EC2 작업은 `aws:RequestedRegion` 및 `ec2:Region` 조건 키를 지원합니다. 자세한 내용은 예제: 특정 리전에 대한 액세스 제한 (p. 814) 단원을 참조하십시오.

`ec2:SourceInstanceARN` 키는 요청이 이루어진 인스턴스의 ARN을 지정하는 조건에 사용할 수 있습니다. 이 조건 키는 AWS 전체 범위에서 사용 가능하고 서비스에 특정하지 않습니다. 정책 예제는 EC2 인스턴스가 볼륨을 연결 또는 분리하도록 허용 및 예제: 특정 인스턴스가 기타 AWS 서비스에서 리소스를 확인하도록 허용 (p. 839)을 참조하십시오. `ec2:SourceInstanceARN` 키는 명령문에서 `Resource` 요소에 대한 ARN을 채우는 변수로 사용할 수 없습니다.

Amazon EC2용 예제 정책 명령문은 AWS CLI 또는 AWS SDK 작업을 위한 정책 예제 (p. 813) 단원을 참조하십시오.

사용자에게 필요한 권한이 있는지 확인

IAM 정책을 생성한 후에는 사용자에게 필요한 특정 API 작업 및 리소스를 사용할 권한이 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 `DryRunOperation`이 반환되고, 그렇지 않은 경우 `UnauthorizedOperation`이 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 CLI Command Reference의 decode-authorization-message 단원을 참조하십시오.

생성 시 리소스 태그 지정에 대한 권한 부여

일부 리소스 생성 Amazon EC2 API 작업에서는 리소스를 생성할 때 태그를 지정할 수 있습니다. 자세한 내용은 리소스에 태그 지정 (p. 1070) 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` 작업에 대한 IAM 정책 정의에서 `ec2:CreateAction` 조건 키와 함께 `Condition` 요소를 사용하여 리소스를 만드는 작업에 태그 지정 권한을 부여합니다.

다음 예제의 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"
      ],
      "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 권한이 필요하지 않습니다. 하지만 사용자가 태그를 사용하여 리소스 생성을 시도하는 경우, 사용자에게 ec2:CreateTags 작업을 사용할 권한이 없다면 요청은 실패합니다.

시작 템플릿에 태그가 제공되는 경우 ec2:CreateTags 작업도 평가됩니다. 정책에 대한 예는 시작 템플릿의 태그 (p. 832)을 참조하십시오.

특정 태그에 대한 액세스 제어

IAM 정책의 Condition 요소에 추가 조건을 사용하여 리소스에 적용할 수 있는 태그 키와 값을 제어할 수 있습니다.

앞 단원의 예제에 다음 조건 키를 사용할 수 있습니다.

- aws:RequestTag: 특정 태그 키 또는 태그 키 및 값이 요청에 존재해야 함을 표시. 요청에서 다른 태그도 지정할 수 있습니다.
  - 특정한 태그와 키 및 가치의 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 StringLike:를 적용하려면) purpose 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-center가 허용됩니다.

    ```
    "ForAllValues:StringEquals": { "aws:TagKeys": ["environment","cost-center"] }
    ```

  - 요청에서 지정된 태그 키 중 최소한 1개의 존재를 적용하려면 ForAnyValue 변경자와 함께 사용하십시오. 예를 들어 요청:에 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. Amazon EC2 API 작업에서 태그 지정을 지원하는지 알아보려면 IAM 사용 설명서의 Amazon EC2에 사용되는 작업, 리소스 및 조건 키를 참조하십시오.

사용자가 리소스를 생성할 때 강제로 태그를 지정하도록 하려면 리소스 생성 작업에서aws: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.

조건의 경우 조건 키는 대소문자를 구분하지 않고 조건 값은 대소문자를 구분합니다. 따라서 태그 키의 대소문자 구별을 설정하려면 태그 키가 조건의 값으로 지정된 aws:TagKeys 조건 키를 사용합니다.

예제 IAM 정책은 AWS CLI 또는 AWS SDK 작업을 위한 정책 예제 (p. 813) 단원을 참조하십시오. 다중 값 조건에 대한 자세한 내용은 IAM 사용 설명서의 다중 키 값을 테스트하는 조건 생성 단원을 참조하십시오.

EC2 리소스 태그를 사용하여 액세스 제어

또한 정책의 `Condition` 요소에 태그 정보를 제공하여 태그를 기반으로 액세스를 제어할 수 있습니다. 이를 통해 사용자가 수정, 사용 또는 삭제할 수 있는 EC2 리소스를 더욱 정확하게 제어할 수 있습니다.

예를 들어 사용자가 인스턴스를 종료할 수 있도록 허용하지만 인스턴스에 `environment=production` 태그가 있는 경우 작업을 거부하는 정책을 만들 수 있습니다. 이렇게 하려면 `ec2:ResourceTag` 조건 키를 사용하여 리소스에 연결된 태그를 기반으로 리소스에 대한 액세스를 허용하거나 거부합니다.

```
"StringEquals": { "ec2:ResourceTag/environment": "production" }
```

Amazon EC2 API 작업에서 `ec2:ResourceTag` 조건 키를 사용한 액세스 제어를 지원하는지 알아보려면 IAM 사용 설명서의 Amazon EC2에 사용되는 작업, 리소스 및 조건 키를 참조하십시오. `Describe` 작업은 리소스 수준 권한을 지원하지 않기 때문에 조건 없이 별도의 명령문에 지정해야 합니다.

예제 IAM 정책은 AWS CLI 또는 AWS SDK 작업을 위한 정책 예제 (p. 813) 단원을 참조하십시오.

> **Note**
>
> 태그를 기준으로 리소스에 대한 사용자 액세스를 허용 또는 거부하는 경우 동일한 리소스에서 태그를 추가 또는 제거할 수 있도록 사용자를 명시적으로 거부할 것을 고려해야 합니다. 그렇지 않으면 사용자가 제한을 피해 태그를 수정하여 리소스에 대한 액세스 권한을 얻을 수 있습니다.

AWS CLI 또는 AWS SDK 작업을 위한 정책 예제

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. 843) 단원을 참조하십시오. Amazon VPC와 관련된 IAM 정책의 예는 Amazon VPC에 대한 Identity and Access Management를 참조하십시오.

예

- 예: 읽기 전용 액세스 (p. 813)
- 예제: 특정 리전에 대한 액세스 제한 (p. 814)
- 인스턴스 작업 (p. 814)
- 볼륨 작업 (p. 816)
- 스냅샷 작업 (p. 818)
- 인스턴스 시작(RunInstances) (p. 825)
- 예제: 예약 인스턴스 작업 (p. 835)
- 예제: 리소스에 태그 지정 (p. 836)
- 예제: IAM 역할 작업 (p. 838)
- 예제: 라우팅 테이블 작업 (p. 839)
- 예제: 특정 인스턴스가 기타 AWS 서비스에서 리소스를 확인하도록 허용 (p. 839)
- 예제: 시작 템플릿 사용 (p. 840)
- 인스턴스 메타데이터 작업 (p. 840)

예: 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 있습니다. API 작업이 리소스 수준 권한을 지원하지 않는 경우에도 * 와일드카드가 필요합니다. Amazon EC2

API 작업에 사용할 수 있는 ARN에 대한 자세한 내용은 IAM 사용 설명서의 Amazon EC2에 사용되는 작업, 리소스 및 조건 키를 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": "*"
    }
   ]
}
```

예제: 특정 리전에 대한 액세스 제한

다음 정책은 유럽(프랑크푸르트) 리전이 아닌 경우 모든 Amazon EC2 API 작업을 사용할 수 있는 사용자 권한을 거부합니다. 모든 Amazon EC2 API 작업에서 지원하는 전역 조건 키 `aws:RequestedRegion`을 사용합니다.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect": "Deny",
      "Action": "ec2:*",
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:RequestedRegion": "eu-central-1"
        }
      }
    }
  ]
}
```

또는 Amazon EC2에 고유하고 모든 Amazon EC2 API 작업에서 지원하는 조건 키 `ec2:Region`을 사용할 수 있습니다.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect": "Deny",
      "Action": "ec2:*",
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "ec2:Region": "eu-central-1"
        }
      }
    }
  ]
}
```

인스턴스 작업

예제

- 예제: 모든 인스턴스를 실행, 중지, 시작 및 종료 (p. 815)
- 예제: 모든 인스턴스를 설명할 수 있지만 특정 인스턴스만 중지, 시작 및 종료 (p. 815)

예제: 모든 인스턴스를 실행, 중지, 시작 및 종료

다음 정책은 사용자에게 `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. Amazon EC2 API 작업에 사용할 수 있는 ARN에 대한 자세한 내용은 IAM 사용 설명서의 Amazon EC2에 사용되는 작업, 리소스 및 조건 키를 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": "*"
    }
   ]
}
```

예제: 모든 인스턴스를 설명할 수 있지만 특정 인스턴스만 중지, 시작 및 종료

다음 정책은 모든 인스턴스를 설명하고, 인스턴스 i-1234567890abcdef0 및 i-0598c7d356eba48d7만 시작 및 중지하고, 리소스 태그가 "미국 동부(버지니아 북부) 지역"인 `us-east-1`의 인스턴스(`purpose=test`)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. Amazon EC2 API 작업에 사용할 수 있는 ARN에 대한 자세한 내용은 IAM 사용 설명서의 Amazon EC2에 사용되는 작업, 리소스 및 조건 키를 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 미국 동부(버지니아 북부) 지역 태그를 갖는 `us-east-1`의 모든 인스턴스(`"purpose=test"`)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"
         }
      }
   }

   ]
}
```

볼륨 작업

예제

- 예제: 볼륨 연결 및 연결 해제 (p. 816)
- 예제: 볼륨 생성 (p. 817)
- 예제: 태그를 사용하여 볼륨 생성 (p. 817)

예제: 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 `volume_user` 사용자 이름을 값으로 하는 IAM라는 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}"
        }
      }
    }
  ]
}
```

예제: 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":{
         "NumericLessThan": {
             "ec2:VolumeSize" : "20"
          },
          "Bool":{
              "ec2:Encrypted" : "true"
          }
       }
    }
  ]
}
```

예제: 태그를 사용하여 볼륨 생성

다음 정책에는 사용자가 태그 `aws:RequestTag` 및 `costcenter=115`를 사용하여 생성하는 볼륨에 태그를 지정해야 하는 `stack=prod` 조건 키가 포함됩니다. `aws:TagKeys` 조건 키는 `ForAllValues` 변경자를 사용하여 요청에서 키 `costcenter` 및 `stack`만 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, 사용자가 `CreateTags` 작업을 사용할 권한도 가지고 있어야 합니다. 두 번째 문은 `ec2:CreateAction` 조건 키를 사용하여 사용자가 `CreateVolume`의 컨텍스트에서만 태그를 생성하도록 허용합니다. 사용자는 기존의 볼륨이나 다른 어떤 리소스에도 태그를 지정할 수 없습니다. 자세한 내용은 생성 시 리소스 태그 지정에 대한 권한 부여 (p. 811) 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"
          }
       }
    }
  ]
}
```

스냅샷 작업

다음은 `CreateSnapshot`(EBS 볼륨의 특정 시점 스냅샷)과 `CreateSnapshots`(다중 볼륨 스냅샷)에 대한 정책 예제입니다.

예제

- 예제: 스냅샷 생성 (p. 819)
- 예제: 스냅샷 생성 (p. 819)
- 예제: 태그를 사용하여 스냅샷 생성 (p. 819)
- 예제: 태그를 사용하여 스냅샷 생성 (p. 820)
- 예제: 스냅샷 권한 설정 수정 (p. 825)

예제: 스냅샷 생성

다음 정책은 고객이 CreateSnapshot API 작업을 사용하는 것을 허용합니다. 고객은 볼륨이 암호화되고 볼륨 크기가 20GiB 미만인 경우에만 스냅샷을 생성할 수 있습니다.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshot",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*"
      },
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshot",
         "Resource":"arn:aws:ec2:us-east-1:123456789012:volume/*",
         "Condition":{
            "NumericLessThan":{
               "ec2:VolumeSize":"20"
            },
            "Bool":{
               "ec2:Encrypted":"true"
            }
         }
      }
   ]
}
```

예제: 스냅샷 생성

다음 정책은 고객이 CreateSnapshots API 작업을 사용하는 것을 허용합니다. 고객은 인스턴스의 모든 볼륨이 GP2 유형인 경우에만 스냅샷을 생성할 수 있습니다.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshots",
         "Resource":[
"arn:aws:ec2:us-east-1::snapshot/*",
"arn:aws:ec2:*:*:instance/*"
   ]
      },
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshots",
         "Resource":"arn:aws:ec2:us-east-1:*:volume/*",
         "Condition":{
            "StringLikeIfExists":{
               "ec2:VolumeType":"gp2"
             }
     }

      }
   ]
}
```

예제: 태그를 사용하여 스냅샷 생성

다음 정책에는 고객이 태그 `aws:RequestTag` 및 `costcenter=115`를 모든 새로운 스냅샷에 적용해야 하는 `stack=prod` 조건 키가 포함됩니다. `aws:TagKeys` 조건 키는 `ForAllValues` 변경자를 사용하여 요청

에서 키 `costcenter` 및 `stack`만 지정될 수 있음을 표시합니다. 조건 중 하나라도 충족하지 않으면 요청은 실패합니다.

태그를 적용하는 리소스 생성 작업의 경우, 고객이 `CreateTags` 작업을 사용할 권한도 가지고 있어야 합니다. 세 번째 문은 `ec2:CreateAction` 조건 키를 사용하여 고객이 `CreateSnapshot`의 컨텍스트에서만 태그를 생성하도록 허용합니다. 고객은 기존의 볼륨이나 다른 어떤 리소스에도 태그를 지정할 수 없습니다. 자세한 내용은 생성 시 리소스 태그 지정에 대한 권한 부여 (p. 811) 단원을 참조하십시오.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshot",
         "Resource":"arn:aws:ec2:us-east-1:123456789012:volume/*"
      },
      {
         "Sid":"AllowCreateTaggedSnapshots",
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshot",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "StringEquals":{
               "aws:RequestTag/costcenter":"115",
               "aws:RequestTag/stack":"prod"
            },
            "ForAllValues:StringEquals":{
               "aws:TagKeys":[
                  "costcenter",
                  "stack"
               ]
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":"ec2:CreateTags",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "StringEquals":{
               "ec2:CreateAction":"CreateSnapshot"
            }
         }
      }
   ]
}
```

예제: 태그를 사용하여 스냅샷 생성

다음 정책에는 고객이 태그 `aws:RequestTag` 및 `costcenter=115`를 모든 새로운 스냅샷에 적용해야 하는 `stack=prod` 조건 키가 포함됩니다. `aws:TagKeys` 조건 키는 `ForAllValues` 변경자를 사용하여 요청에서 키 `costcenter` 및 `stack`만 지정될 수 있음을 표시합니다. 조건 중 하나라도 충족하지 않으면 요청은 실패합니다.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshots",
         "Resource":[
"arn:aws:ec2:us-east-1::snapshot/*",
```

```
"arn:aws:ec2:*:*:instance/*",
"arn:aws:ec2:*:*:volume/*"

   ]
      },
      {
         "Sid":"AllowCreateTaggedSnapshots",
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshots",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "StringEquals":{
               "aws:RequestTag/costcenter":"115",
               "aws:RequestTag/stack":"prod"
            },
            "ForAllValues:StringEquals":{
               "aws:TagKeys":[
                  "costcenter",
                  "stack"
               ]
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":"ec2:CreateTags",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "StringEquals":{
               "ec2:CreateAction":"CreateSnapshots"
            }
         }
      }
   ]
}
```

다음 정책은 고객이 태그를 지정하지 않고 스냅샷을 생성하는 것을 허용합니다. `CreateTags` 작업은 `CreateSnapshot` 또는 `CreateSnapshots` 요청에서 태그가 지정되는 경우에만 평가됩니다. 태그가 지정된 경우 태그는 `purpose=test`여야 합니다. 다른 어떤 태그도 요청에서 허용되지 않습니다.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshot",
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":"ec2:CreateTags",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "StringEquals":{
               "aws:RequestTag/purpose":"test",
               "ec2:CreateAction":"CreateSnapshot"
            },
            "ForAllValues:StringEquals":{
               "aws:TagKeys":"purpose"
            }
         }
      }
   ]
}
```

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshots",
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":"ec2:CreateTags",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "StringEquals":{
               "aws:RequestTag/purpose":"test",
               "ec2:CreateAction":"CreateSnapshots"
            },
            "ForAllValues:StringEquals":{
               "aws:TagKeys":"purpose"
            }
         }
      }
   ]
}
```

다음 정책은 소스 볼륨이 고객에 대해 `User:`*`username`*으로 태그 지정된 경우, 그리고 스냅샷 자체가 `Environment:Dev` 및 `User:`*`username`*으로 태그 지정된 경우에만 스냅샷 생성을 허용합니다. 고객은 스냅샷에 추가 태그를 추가할 수 있습니다.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshot",
         "Resource":"arn:aws:ec2:us-east-1:123456789012:volume/*",
         "Condition":{
            "StringEquals":{
               "ec2:ResourceTag/User":"${aws:username}"
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshot",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "StringEquals":{
               "aws:RequestTag/Environment":"Dev",
               "aws:RequestTag/User":"${aws:username}"
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":"ec2:CreateTags",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*"
      }
   ]
}
```

`CreateSnapshots`에 대한 다음 정책은 고객에 대해 `User:`*`username`*으로 태그 지정된 경우, 그리고 스냅샷 자체가 `Environment:Dev` 및 `User:`*`username`*으로 태그 지정된 경우에만 스냅샷 생성을 허용합니다.

```
{
   "Version":"2012-10-17",
   "Statement":[
 {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshots",
         "Resource":"arn:aws:ec2:us-east-1:*:instance/*",
 },
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshots",
         "Resource":"arn:aws:ec2:us-east-1:123456789012:volume/*",
         "Condition":{
            "StringEquals":{
               "ec2:ResourceTag/User":"${aws:username}"
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshots",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "StringEquals":{
               "aws:RequestTag/Environment":"Dev",
               "aws:RequestTag/User":"${aws:username}"
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":"ec2:CreateTags",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*"
      }
   ]
}
```

다음 정책은 스냅샷이 고객에 대해 User:username으로 태그 지정된 경우에만 스냅샷 삭제를 허용합니다.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"ec2:DeleteSnapshot",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "StringEquals":{
               "ec2:ResourceTag/User":"${aws:username}"
            }
         }
      }
   ]
}
```

다음 정책은 고객의 스냅샷 생성을 허용하지만 생성되는 스냅샷이 태그 키 `value=stack`을 보유한 경우에는 작업을 거부합니다.

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

```
            "Action":[
               "ec2:CreateSnapshot",
               "ec2:CreateTags"
            ],
            "Resource":"*"
         },
         {
            "Effect":"Deny",
            "Action":"ec2:CreateSnapshot",
            "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
            "Condition":{
               "ForAnyValue:StringEquals":{
                  "aws:TagKeys":"stack"
               }
            }
         }
      ]
}
```

다음 정책은 고객의 스냅샷 생성을 허용하지만 생성되는 스냅샷이 태그 키 `value=stack`을 보유한 경우에는 작업을 거부합니다.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "ec2:CreateSnapshots",
            "ec2:CreateTags"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Deny",
         "Action":"ec2:CreateSnapshots",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "ForAnyValue:StringEquals":{
               "aws:TagKeys":"stack"
            }
         }
      }
   ]
}
```

다음 정책은 여러 작업을 정책 하나로 결합하도록 허용합니다. 스냅샷이 `us-east-1` 리전에서 생성된 경우에만 (`CreateSnapshots`의 컨텍스트에서) 스냅샷을 생성할 수 있습니다. 스냅샷이 `us-east-1` 리전에서 생성되고 인스턴스 유형이 `t2*`인 경우에만 (`CreateSnapshots`의 컨텍스트에서) 스냅샷을 생성할 수 있습니다.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "ec2:CreateSnapshots",
            "ec2:CreateSnapshot",
            "ec2:CreateTags"
         ],
         "Resource": [
            "arn:aws:ec2:*:*:instance/*",
```

```
            "arn:aws:ec2:*:*:snapshot/*",
            "arn:aws:ec2:*:*:volume/*"
         ],
         "Condition":{
            "StringEqualsIgnoreCase": {
            "ec2:Region": "us-east-1"
            },
            "StringLikeIfExists": {
            "ec2:InstanceType": [
            "t2.*"
            ]
            }
         }
      }
   ]
}
```

예제: 스냅샷 권한 설정 수정

다음 정책에서는 스냅샷에 `User:username` 태그가 지정된 경우에만 스냅샷 수정을 허용합니다. 여기서 *username*은 고객의 AWS 계정 사용자 이름입니다. 이 조건이 충족되지 않으면 요청은 실패합니다.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"ec2: ModifySnapshotAttribute",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "StringEquals":{
               "ec2:ResourceTag/user-name":"${aws:username}"
            }
         }
      }
   ]
}
```

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. VPC로 시작하는 경우 서브넷을 입력 받아 네트워크 인터페이스를 생성합니다. Amazon EBS 지원 AMI에서 시작하면 볼륨이 생성됩니다. 따라서 사용자가 이러한 Amazon EC2 리소스를 사용할 권한을 가지고 있어야 합니다. 사용자가 `RunInstances`에서 선택적 파라미터를 지정하도록 요구하거나 사용자가 파라미터에 특정 값을 사용하도록 제한하는 정책 명령문을 생성할 수 있습니다.

인스턴스를 시작하는 데 필요한 리소스 수준 권한에 대한 자세한 내용은 IAM 사용 설명서의 Amazon EC2에 사용되는 작업, 리소스 및 조건 키를 참조하십시오.

기본적으로는 사용자에게 생성된 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 인스턴스 작업 (p. 814) 단원을 참조하십시오.

리소스

- AMI (p. 826)
- 인스턴스 유형 (p. 827)
- 서브넷 (p. 828)

- EBS 볼륨 (p. 829)
- Tags (p. 829)
- 시작 템플릿의 태그 (p. 832)
- 탄력적 GPU (p. 833)
- 시작 템플릿 (p. 833)

AMI

다음 정책은 사용자가 지정된 AMI(`ami-9e1670f7` 및 `ami-45cf5c3c`)만 사용하여 인스턴스를 시작하도록 허용합니다. 다른 명령문에서 해당 권한을 부여하지 않는 경우 다른 AMI를 사용하여 인스턴스를 시작할 수 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/*",
        "arn:aws:ec2:region:account:subnet/*",
        "arn:aws:ec2:region:account:network-interface/*"
      ]
    }
   ]
}
```

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.

```
{
   "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:InstanceType`이 `t2.micro` 또는 `t2.small`인지 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 시작할 수 있는 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.

```
{
   "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 수 있는 사용자 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.

```
{
   "Version": "2012-10-17",
   "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/*"
            ]
        }
    ]
}
```

Tags

다음 정책은 사용자가 인스턴스를 시작하고 생성 중에 인스턴스에 태그를 지정하는 것을 허용합니다. 태그를 적용하는 리소스 생성 작업의 경우, 사용자가 `CreateTags` 작업을 사용할 권한을 가지고 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 생성 시 리소스 태그 지정에 대한 권한 부여 (p. 811) 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 `aws:RequestTag` 및 `RunInstances`를 사용하여 `environment=production`에 의해 생성되는 인스턴스와 볼륨에 사용자가 태그를 지정해야 하는 `purpose=webserver` 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: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 `ForAnyValue` 조건에서 `aws:TagKeys` 변경자를 사용하여 요청에서 적어도 하나의 태그가 지정되어야 하고 태그에 키 `environment` 또는 `webserver`가 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": {
          "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"
          }
       }
    }
  ]
}
```

시작 템플릿의 태그

아래 예제에서 사용자는 특정 시작 템플릿(`lt-09477bcd97b0d310e`)을 사용하는 경우에만 인스턴스를 시작할 수 있습니다. `ec2:IsLaunchTemplateResource` 조건 키는 사용자가 시작 템플릿에 지정된 모든 리소스를 재정의할 수 없도록 합니다. 이 명령문의 두 번째 부분은 사용자가 생성 시 인스턴스에 태그를 지정하도록 허용합니다. 이 부분은 시작 템플릿의 인스턴스에 대해 태그가 지정되어 있는 경우에 반드시 필요합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": "*",
      "Condition": {
         "ArnLike": {
             "ec2:LaunchTemplate": "arn:aws:ec2:region:account:launch-template/
lt-09477bcd97b0d310e"
          },
          "Bool": {
             "ec2:IsLaunchTemplateResource": "true"
          }
       }
    },
    {
      "Effect": "Allow",
      "Action": [
         "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:region:account:instance/*",
      "Condition": {
         "StringEquals": {
             "ec2:CreateAction" : "RunInstances"
          }
       }
    }
```

```
  ]
}
```

탄력적 GPU

다음 정책에서 사용자는 인스턴스를 시작하고 탄력적 GPU를 지정하여 인스턴스에 연결합니다. 사용자는 모든 지역에서 인스턴스를 시작할 수 있지만 `us-east-2` 지역에서 시작 작업 동안 탄력적 GPU만 연결할 수 있습니다.

`ec2:ElasticGpuType` 조건 키는 `ForAnyValue` 한정자를 사용하여 요청에서 엘라스틱 GPU 유형 `eg1.medium` 및 `eg1.large`만 허용됨을 표시합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:*:account:elastic-gpu/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:Region": "us-east-2"
                },
                "ForAnyValue:StringLike": {
                    "ec2:ElasticGpuType": [
                        "eg1.medium",
                        "eg1.large"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:*::image/ami-*",
                "arn:aws:ec2:*:account:network-interface/*",
                "arn:aws:ec2:*:account:instance/*",
                "arn:aws:ec2:*:account:subnet/*",
                "arn:aws:ec2:*:account:volume/*",
                "arn:aws:ec2:*:account:key-pair/*",
                "arn:aws:ec2:*:account:security-group/*"
            ]
        }
    ]
}
```

시작 템플릿

아래 예제에서 사용자는 특정 시작 템플릿(`lt-09477bcd97b0d310e`)을 사용하는 경우에만 인스턴스를 시작할 수 있습니다. 사용자는 `RunInstances` 작업에서 파라미터를 지정하여 시작 템플릿의 모든 파라미터를 재정의할 수 있습니다.

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

```
      "Action": "ec2:RunInstances",
      "Resource": "*",
      "Condition": {
         "ArnLike": {
             "ec2:LaunchTemplate": "arn:aws:ec2:region:account:launch-template/
lt-09477bcd97b0d310e"
          }
       }
    }
  ]
}
```

아래 예제에서 사용자는 시작 템플릿을 사용하는 경우에만 인스턴스를 시작할 수 있습니다. 정책은 `ec2:IsLaunchTemplateResource` 조건 키를 사용하여 사용자가 시작 템플릿의 기존 ARN을 재정의하지 못하도록 합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": "*",
      "Condition": {
         "ArnLike": {
             "ec2:LaunchTemplate": "arn:aws:ec2:region:account:launch-template/*"
          },
          "Bool": {
             "ec2:IsLaunchTemplateResource": "true"
          }
       }
    }
  ]
}
```

아래 예제는 사용자가 시작 템플릿을 사용하는 경우에만 인스턴스를 시작하도록 허용하는 정책입니다. 사용자는 요청 시 서브넷 및 네트워크 인터페이스 파라미터를 재정의할 수 없으며, 시작 템플릿에서만 이러한 파라미터들을 지정할 수 있습니다. 이 명령문의 첫 번째 부분은 NotResource 요소를 사용하여 서브넷 및 네트워크 인터페이스를 제외한 다른 모든 리소스를 허용합니다. 이 명령문의 두 번째 부분은 시작 템플릿에서 나온 경우에만 서브넷 및 네트워크 인터페이스 리소스를 허용합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "NotResource": ["arn:aws:ec2:region:account:subnet/*",
                      "arn:aws:ec2:region:account:network-interface/*" ],
      "Condition": {
         "ArnLike": {
             "ec2:LaunchTemplate": "arn:aws:ec2:region:account:launch-template/*"
          }
       }
    },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": ["arn:aws:ec2:region:account:subnet/*",
                   "arn:aws:ec2:region:account:network-interface/*" ],
      "Condition": {
         "ArnLike": {
```

```
            "ec2:LaunchTemplate": "arn:aws:ec2:region:account:launch-template/*"
         },
         "Bool": {
            "ec2:IsLaunchTemplateResource": "true"
         }
      }
   }
  ]
}
```

아래 예제는 시작 템플릿을 사용하고 있고 시작 템플릿에 `Purpose=Webservers` 태그가 있는 경우에만 인스턴스를 시작할 수 있도록 허용합니다. 사용자는 `RunInstances` 작업의 어떤 시작 템플릿 파라미터도 재정의할 수 없습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "NotResource": "arn:aws:ec2:region:account:launch-template/*",
      "Condition": {
         "ArnLike": {
             "ec2:LaunchTemplate": "arn:aws:ec2:region:account:launch-template/*"
          },
         "Bool": {
             "ec2:IsLaunchTemplateResource": "true"
          }
       }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": "arn:aws:ec2:region:account:launch-template/*",
      "Condition": {
       "StringEquals": {
           "ec2:ResourceTag/Purpose": "Webservers"
        }
       }
     }
  ]
}
```

예제: 예약 인스턴스 작업

다음 정책에서는 계정에서 예약 인스턴스를 확인, 수정 및 구입할 수 있는 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` 요소에서는 와일드카드(*)를 사용하여 사용자가 해당 작업에서 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 확인 및 수정하도록 허용하되 새 예약 인스턴스를 구매할 수는 없게 하려면

```
{
  "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeReservedInstances", "ec2:ModifyReservedInstances",
        "ec2:DescribeAvailabilityZones"
      ],
      "Resource": "*"
    }
  ]
}
```

예제: 리소스에 태그 지정

다음 정책은 태그에 키 `CreateTags` 및 값 `environment`이 포함된 경우에만 사용자가 `production` 작업을 사용하여 인스턴스에 태그를 적용하는 것을 허용합니다. `ForAllValues` 변경자는 `aws:TagKeys` 조건 키와 함께 사용되어 요청에서 키 `environment`만 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": {
                    "aws:RequestTag/environment": "production"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": [
                        "environment"
                    ]
                }
            }
        }
    ]
}
```

다음 정책은 키가 `owner`이고 값이 IAM username인 태그를 이미 가진 태그 지정 가능한 리소스에 태그를 지정하는 것을 허용합니다. 또한 사용자는 요청에서 키가 `anycompany:environment-type`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/anycompany:environment-type": ["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 `ec2:DeleteTags` 작업을 사용할 권한이 없어도 태그가 지정된 리소스를 삭제할 수 있습니다 . 삭제 작업을 수행할 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 username인 키로 리소스에 태그가 지정된 경우에 한해 어떤 리소스에서든 `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"]
        }
      }
    }
  ]
}
```

예제: 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. 851) 단원을 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-*"
        }
    ]
}
```

예제: 라우팅 테이블 작업

다음 정책은 VPC `vpc-ec43eb89`에만 연결된 라우팅 테이블의 경로에 대해 사용자가 추가, 제거 및 바꾸기 작업을 수행할 수 있도록 허용합니다. `ec2:Vpc` 조건 키에 대한 VPC를 지정하려면 VPC의 전체 ARN을 지정해야 합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteRoute",
                "ec2:CreateRoute",
                "ec2:ReplaceRoute"
            ],
            "Resource": [
                "arn:aws:ec2:region:account:route-table/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:Vpc": "arn:aws:ec2:region:account:vpc/vpc-ec43eb89"
                }
            }
        }
    ]
}
```

예제: 특정 인스턴스가 기타 AWS 서비스에서 리소스를 확인하도록 허용

다음은 IAM 역할에 연결할 수 있는 정책의 예제입니다. 이 정책은 인스턴스가 다양한 AWS 서비스에서 리소스를 확인하도록 허용합니다. `ec2:SourceInstanceARN` 조건 키를 사용하여 요청이 이루어진 인스턴스가 `i-093452212644b0dd6` 인스턴스가 되도록 지정합니다. 동일한 IAM 역할이 다른 인스턴스와 연결된 경우에는 다른 인스턴스에서 이러한 작업을 수행할 수 없습니다.

`ec2:SourceInstanceARN` 키는 AWS 전체 범위 조건 키이므로 Amazon EC2뿐 아니라 다른 서비스 작업에 사용할 수 있습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVolumes",
                "s3:ListAllMyBuckets",
```

```
                "dynamodb:ListTables",
                "rds:DescribeDBInstances"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "ArnEquals": {
                    "ec2:SourceInstanceARN": "arn:aws:ec2:region:account:instance/
i-093452212644b0dd6"
                }
            }
        }
    ]
}
```

예제: 시작 템플릿 사용

아래 정책은 특정 시작 템플릿(`lt-09477bcd97b0d3abc`)에서만 시작 템플릿 버전을 생성하고 시작 템플릿을 수정할 수 있도록 허용합니다. 사용자는 다른 시작 템플릿은 사용할 수 없습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:CreateLaunchTemplateVersion",
        "ec2:ModifyLaunchTemplate"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:ec2:region:account:launch-template/lt-09477bcd97b0d3abc"
    }
  ]
}
```

아래 정책은 시작 템플릿에 `Purpose=Testing` 태그가 지정되어 있는 경우에 모든 시작 템플릿 및 시작 템플릿 버전을 삭제할 수 있도록 허용합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:DeleteLaunchTemplate",
        "ec2:DeleteLaunchTemplateVersions"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:ec2:region:account:launch-template/*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/Purpose": "Testing"
        }
      }
    }
  ]
}
```

인스턴스 메타데이터 작업

다음 정책은 사용자가 인스턴스 메타데이터 서비스 버전 2(IMDSv2)를 사용하여 인스턴스 메타데이터 (p. 571)만 검색할 수 있도록 합니다. 다음 정책 4개를 결합하여 명령문 4개를 가진 정책 하나로 만들 수 있

습니다. 정책이 하나로 결합되면 이 정책을 서비스 제어 정책(SCP)으로 사용할 수 있습니다. 이 정책은 기존 IAM 정책에 적용한 deny 정책(기존 권한 취소 및 제한)이나 계정, 조직 단위(OU) 또는 전체 조직에서 전역적으로 적용되는 SCP와 똑같이 원활하게 적용됩니다.

Note

보안 주체에 RunInstances로 인스턴스를 시작할 수 있는 권한을 제공하는 정책과 함께 다음 RunInstances 메타데이터 옵션 정책을 사용해야 합니다. 또한 보안 주체에 RunInstances 권한이 없으면 인스턴스를 시작할 수 없습니다. 자세한 내용은 인스턴스 작업 (p. 814) 및 인스턴스 시작 (RunInstances) (p. 825)의 정책을 참조하십시오.

Important

Auto Scaling 그룹을 사용할 때 모든 새 인스턴스에서 IMDSv2를 사용해야 하는 경우 Auto Scaling 그룹에서 시작 템플릿을 사용해야 합니다.
Auto Scaling 그룹이 시작 템플릿을 사용하는 경우 새 Auto Scaling 그룹이 생성될 때 IAM 보안 주체의 `ec2:RunInstances` 권한이 확인됩니다. 또한 새 시작 템플릿이나 새 버전의 시작 템플릿을 사용하도록 기존 Auto Scaling 그룹이 업데이트될 때도 확인됩니다.
`RunInstances`에 대한 IAM 보안 주체의 IMDSv1 사용에 대한 제한은 시작 템플릿을 사용하는 Auto Scaling 그룹이 생성 또는 업데이트될 때에만 확인됩니다. `Latest` 또는 `Default` 시작 템플릿을 사용하도록 구성된 Auto Scaling 그룹의 경우 새 버전의 시작 템플릿을 생성할 때 권한이 확인되지 않습니다. 권한을 확인하려면 특정 버전의 시작 템플릿을 사용하도록 Auto Scaling 그룹을 구성해야 합니다.

Auto Scaling 그룹에서 시작한 인스턴스에 대해 IMDSv2를 사용하도록 하려면 다음과 같은 추가 단계가 필요합니다.

1. 생성되는 새 보안 주체에 대해 IAM 권한 경계 또는 서비스 제어 정책(SCP)을 사용하여 조직의 모든 계정에 대해 시작 구성의 사용을 비활성화합니다. Auto Scaling 그룹 권한이 있는 기존 IAM 보안 주체의 경우 이 조건 키로 연결된 정책을 업데이트합니다. 시작 구성의 사용을 비활성화하려면 값이 `null`로 지정된 `"autoscaling:LaunchConfigurationName"` 조건 키를 사용하여 관련 SCP, 권한 경계 또는 IAM 정책을 생성하거나 수정합니다.
2. 새 시작 템플릿의 경우 시작 템플릿에서 인스턴스 메타데이터 옵션을 구성합니다. 기존 시작 템플릿의 경우 새 버전의 시작 템플릿을 만들고 새 버전에서 인스턴스 메타데이터 옵션을 구성합니다.
3. 보안 주체에게 시작 템플릿을 사용할 권한을 부여하는 정책에서 `"autoscaling:LaunchTemplateVersionSpecified": "true"`를 지정하여 `$latest` 및 `$default`의 연결을 제한합니다. 특정 버전의 시작 템플릿으로 사용을 제한하면 인스턴스 메타데이터 옵션이 구성된 버전을 사용하여 새 인스턴스가 시작되도록 할 수 있습니다. 자세한 내용은 Amazon EC2 Auto Scaling API 참조의 LaunchTemplateSpecification, 특히 `Version` 파라미터를 참조하십시오.
4. 시작 구성을 사용하는 Auto Scaling 그룹의 경우 시작 구성을 시작 템플릿으로 바꿉니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 시작 구성을 시작 템플릿으로 바꾸기를 참조하십시오.
5. 시작 템플릿을 사용하는 Auto Scaling 그룹의 경우 인스턴스 메타데이터 옵션이 구성된 새 시작 템플릿을 사용하거나 인스턴스 메타데이터 옵션이 구성된 현재 시작 템플릿의 새 버전을 사용해야 합니다. 자세한 내용은 AWS CLI Command Reference의 update-auto-scaling-group을 참조하십시오.

예

- IMDSv2의 사용 요구 (p. 842)
- 최대 홉 제한 지정 (p. 842)
- 인스턴스 메타데이터 옵션을 수정할 수 있는 사용자 제한 (p. 842)
- IMDSv2에서 역할 자격 증명을 검색하도록 요구 (p. 843)

IMDSv2의 사용 요구

다음 정책에서는 IMDSv2(`"ec2:MetadataHttpTokens": "required"`로 표시) 사용을 요구하도록 인스턴스도 옵트인되지 않으면 RunInstances API를 호출할 수 없도록 지정합니다. 인스턴스가 IMDSv2를 요구하도록 지정하지 않으면 RunInstances API를 호출할 때 `UnauthorizedOperation` 오류가 발생합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RequireImdsV2",
            "Effect": "Deny",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
                "StringNotEquals": {
                    "ec2:MetadataHttpTokens": "required"
                }
            }
        }
    ]
}
```

최대 홉 제한 지정

다음 정책은 홉 제한도 지정하지 않으면(홉 제한은 3을 초과할 수 없음) RunInstances API를 호출할 수 없도록 지정합니다. 그렇게 하지 않으면 RunInstances API를 호출할 때 `UnauthorizedOperation` 오류가 발생합니다.

> **Note**
>
> 다음 정책과 앞에 나온 정책이 SCP를 통해 계정에 적용되면 EC2 콘솔을 사용하여 인스턴스를 시작할 수 없습니다. 콘솔에서 아직 `MetadataHttpTokens` 파라미터와 `MetadataHttpPutResponseHopLimit` 파라미터를 지원하지 않기 때문입니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "MaxImdsHopLimit",
            "Effect": "Deny",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
                "NumericGreaterThan": {
                    "ec2:MetadataHttpPutResponseHopLimit": "3"
                }
            }
        }
    ]
}
```

인스턴스 메타데이터 옵션을 수정할 수 있는 사용자 제한

다음 정책은 일반 관리자들이 인스턴스 메타데이터 옵션을 수정할 수 있는 기능을 제거하고 `ec2-imds-admins` 역할이 있는 사용자만 변경할 수 있도록 허용합니다. `ec2-imds-admins` 역할이 아닌 보안 주체가 ModifyInstanceMetadataOptions API를 호출하려고 하면 `UnauthorizedOperation` 오류가 발생합니다. 이 명령문을 사용하여 ModifyInstanceMetadataOptions API 사용을 제어할 수 있습니다. 지금은 ModifyInstanceMetadataOptions API의 세부적인 액세스 제어(조건)가 없습니다.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowOnlyImdsAdminsToModifySettings",
            "Effect": "Deny",
            "Action": "ec2:ModifyInstanceMetadataOptions",
            "Resource": "*",
            "Condition": {
                "StringNotLike": {
                    "aws:PrincipalARN": "arn:aws:iam::*:role/ec2-imds-admins"
                }
            }
        }
    ]
}
```

IMDSv2에서 역할 자격 증명을 검색하도록 요구

다음 정책은 이 정책이 역할에 적용되고, EC2 서비스가 이 역할을 맡으며, 그 결과로 생긴 자격 증명이 요청에 서명하는 데 사용되며, IMDSv2에서 검색한 EC2 역할 자격 증명으로 요청에 서명해야 한다고 지정합니다. 그렇게 하지 않으면 모든 API 호출에서 `UnauthorizedOperation` 오류가 발생합니다. 이 명령문/정책은 일반적으로 적용됩니다. EC2 역할 자격 증명으로 요청에 서명하지 않으면 요청은 아무 효과가 없습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RequireAllEc2RolesToUseV2",
            "Effect": "Deny",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "NumericLessThan": {
                    "ec2:RoleDelivery": "2.0"
                }
            }
        }
    ]
}
```

Amazon EC2 콘솔 작업을 위한 예제 정책

IAM 콘솔에서 Amazon EC2 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.

> **Tip**
>
> 콘솔에서 작업을 수행하는 데 필요한 API 작업을 파악하려는 경우 AWS CloudTrail 등의 서비스를 사용할 수 있습니다. 자세한 내용은 AWS CloudTrail User Guide를 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.

예제

- 예: 읽기 전용 액세스 (p. 844)
- 예제: EC2 시작 마법사 사용 (p. 845)

- 예제: 볼륨 작업 (p. 847)
- 예제: 보안 그룹 작업 (p. 848)
- 예제: 탄력적 IP 주소 작업 (p. 849)
- 예제: 예약 인스턴스 작업 (p. 850)

Amazon EC2 콘솔용 정책을 생성하는 방법에 대한 자세한 내용은 AWS 보안 블로그 게시물 Granting Users Permission to Work in the Amazon EC2 Console을 참조하십시오.

예: 읽기 전용 액세스

사용자가 Amazon EC2 콘솔에서 모든 리소스를 조회하도록 허용하려면 다음 예제와 같은 정책을 사용합니다. 예: 읽기 전용 액세스 (p. 813). 다른 명령문에서 해당 권한을 부여하지 않는 경우 이러한 리소스에 대해 작업을 수행하거나 새 리소스를 생성할 수는 없습니다.

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**
>
> 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. Amazon EC2 API 작업에 사용할 수 있는 ARN에 대한 자세한 내용은 IAM 사용 설명서의 Amazon EC2에 사용되는 작업, 리소스 및 조건 키를 참조하십시오.

인스턴스 및 CloudWatch 측정치 조회

다음 정책은 사용자로 하여금 인스턴스 페이지의 모니터링 탭에 있는 CloudWatch 경보 및 지표뿐만 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": "*"
   }
   ]
}
```

예제: 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.

기본 시작 마법사에 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 옵션을 조회합니다.
- `ec2:DescribeSubnets`: 선택한 VPC에 대한 모든 사용 가능한 서브넷을 조회합니다.
- `ec2:DescribeSecurityGroups` 또는 `ec2:CreateSecurityGroup`: 기존 보안 그룹을 조회하고 선택하거나 새로 생성합니다.
- `ec2:DescribeKeyPairs` 또는 `ec2:CreateKeyPair`: 기존 키 페어를 선택하거나 새로 생성합니다.
- `ec2:AuthorizeSecurityGroupIngress`: 인바운드 규칙을 추가합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeImages",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeVpcs",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:CreateSecurityGroup",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateKeyPair"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "*"
        }
    ]
}
```

정책에 API 작업을 추가하여 다음과 같이 사용자에게 더 많은 옵션을 제공할 수 있습니다.

- `ec2:DescribeAvailabilityZones`: 특정 가용 영역을 조회하고 선택합니다.
- `ec2:DescribeNetworkInterfaces`: 선택한 서브넷의 기존 네트워크 인터페이스를 조회하고 선택합니다.
- 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 생성 시 리소스 태그 지정에 대한 권한 부여 (p. 811) 단원을 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.

특정 인스턴스 유형, 서브넷, 지역에 대한 액세스 제한

다음 정책은 Amazon이 소유한 AMI를 사용하여 `t2.micro` 인스턴스를 시작하되 특정 서브넷 (`subnet-1a2b3c4d`)으로만 시작하도록 허용합니다. 사용자는 sa-east-1 지역에서만 시작할 수 있습니다. 사용자가 다른 지역을 선택하거나 시작 마법사에서 다른 인스턴스 유형, AMI 또는 서브넷을 선택하면 시작에 실패합니다.

첫 번째 명령문은 위 예제에 설명된 대로 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 인스턴스 시작(RunInstances) (p. 825) 단원을 참조하십시오. 세 번째, 네 번째 명령문은 각각 인스턴스 및 AMI 리소스 사용 권한을 부여하지만 인스턴스가 `t2.micro` 인스턴스이고 AMI를 Amazon이 소유한 경우로 한정합니다.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "ec2:DescribeInstances", "ec2:DescribeImages",
         "ec2:DescribeKeyPairs", "ec2:CreateKeyPair", "ec2:DescribeVpcs",
 "ec2:DescribeSubnets", "ec2:DescribeSecurityGroups", "ec2:CreateSecurityGroup",
 "ec2AuthorizeSecurityGroupIngress"
   ],
   "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": "t2.micro"
         }
```

```
      }
   },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
            "arn:aws:ec2:sa-east-1::image/ami-*"
      ],
      "Condition": {
         "StringEquals": {
            "ec2:Owner": "amazon"
         }
      }
   }
   ]
}
```

예제: 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 볼륨 연결 대화 상자의 미리 구성된 목록에서 인스턴스를 선택할 수 있습니다. 그러나 이렇게 하면 사용자가 콘솔의 인스턴스 페이지에서 모든 인스턴스를 조회할 수 있으므로 이 작업을 생략할 수 있습니다.

첫 번째 명령문에서 `ec2:DescribeAvailabilityZones` 작업은 볼륨을 생성할 때 사용자가 가용 영역을 선택할 수 있도록 하는 데 필요합니다.

사용자는 볼륨 생성 중 또는 생성 후 본인이 작성한 볼륨에 태그를 지정할 수 없습니다.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVolumes",
        "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/*"
   }
   ]
}
```

예제: 보안 그룹 작업

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.

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"
         }
      }
   }
   ]
}
```

보안 그룹 생성 대화 상자 작업

사용자가 Amazon EC2 콘솔에서 보안 그룹 생성 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 권한이 있으면 사용자가 새 보안 그룹을 생성할 수 있지만 규칙을 추가할 수는 없습니다. 보안 그룹 생성 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 새로 복사 기능을 사용하도록 허용하려는 경우에 유용합니다. 이 기능은 보안 그룹 생성 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 보안 그룹 생성 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 보안 그룹 생성 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-1a2b3c4d`에 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"
         }
      }
    }
   ]
}
```

예제: 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`: 탄력적 IP 주소를 할당합니다.
- `ec2:ReleaseAddress`: 탄력적 IP 주소를 해제합니다.
- `ec2:AssociateAddress`: 인스턴스 또는 네트워크 인터페이스에 탄력적 IP 주소를 연결합니다.
- `ec2:DescribeNetworkInterfaces` 및 `ec2:DescribeInstances`: 주소 연결 화면에서 작업합니다. 탄력적 IP 주소를 연결할 수 있는 네트워크 인터페이스나 가용 인스턴스가 화면에 표시됩니다.

- `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": "*"
        }
    ]
}
```

예제: 예약 인스턴스 작업

다음 정책을 IAM 사용자에 연결할 수 있습니다. 그렇게 하면 사용자가 계정의 예약 인스턴스를 보고 수정할 수 있을 뿐만 아니라 AWS Management 콘솔에서 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.

IAM의 Amazon EC2 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 인스턴스

에 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, 기존 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. 이 방법은 IAM 사용자를 대상으로 정책을 생성할 때와 비슷합니다. 역할을 변경하면 모든 인스턴스에 변경 내용이 전파됩니다.

IAM 역할을 생성할 때, 애플리케이션에 필요한 특정 API 호출에 대한 액세스를 제한하는 최소 권한 IAM 정책을 연결합니다.

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 역할을 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. 808) 및 다음 예제: 예제: IAM 역할 작업 (p. 838)단원을 참조하십시오.

주제

- 인스턴스 프로파일 (p. 851)
- 인스턴스 메타데이터에서 보안 자격 증명 검색 (p. 851)
- IAM 사용자에게 IAM 역할을 인스턴스에 전달할 수 있는 권한 부여 (p. 852)
- IAM 역할 작업 (p. 853)

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 역할을 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 프로파일을 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 IAM라는 `s3access` 역할의 보안 자격 증명을 검색합니다.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-
ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-
data/iam/security-credentials/s3access
```

IMDSv1

```
[ec2-user ~]$ 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" : "ASIAIOSFODNN7EXAMPLE",
  "SecretAccessKey" : "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
  "Token" : "token",
  "Expiration" : "2017-05-17T15:09:54Z"
}
```

인스턴스에서 실행되는 애플리케이션, AWS CLI 및 Windows PowerShell용 도구 명령의 경우, 임시 보안 자격 증명을 명시적으로 받지 않아도 됩니다. AWS SDK, AWS CLI, Windows PowerShell용 도구이 EC2 인스턴스 메타데이터 서비스에서 자동으로 자격 증명을 받아 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 요청하기를 참조하십시오.

인스턴스 메타데이터에 대한 자세한 내용은 인스턴스 메타데이터 및 사용자 데이터 (p. 571) 단원을 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.

내용

- IAM 역할 만들기 (p. 853)
- IAM 역할로 인스턴스 시작 (p. 855)
- IAM 역할을 인스턴스에 연결 (p. 856)
- IAM 역할 교체 (p. 857)
- IAM 역할 분리 (p. 858)

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 콘솔을 엽니다.
2. 탐색 창에서 역할, 역할 생성을 선택합니다.
3. 역할 유형 선택 페이지에서 EC2와 EC2 사용 사례를 선택합니다. 다음: 권한을 선택합니다.
4. 권한 정책 연결(Attach permissions policy) 페이지에서 인스턴스가 필요로 하는 리소스에 대한 액세스 권한을 부여하는 AWS 관리형 정책을 선택합니다.
5. 검토 페이지에서 역할의 이름을 입력하고 역할 생성을 선택합니다.

또는 AWS CLI를 사용하여 IAM 역할을 만들 수 있습니다. 다음 예제에서는 IAM 역할이 Amazon S3 버킷을 사용하도록 허용하는 정책을 사용하여 이 역할을 만듭니다.

IAM 역할과 인스턴스 프로파일을 생성하려면(AWS CLI)

1. 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"
    }
  ]
}
```

2. `s3access` 역할을 만들고 create-role 명령을 사용하여 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"
    }
}
```

3. 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": ["*"]
    }
  ]
}
```

4. put-role-policy 명령을 사용하여 액세스 정책을 역할에 연결합니다.

```
aws iam put-role-policy --role-name s3access --policy-name S3-Permissions --policy-
document file://ec2-role-access-policy.json
```

5. create-instance-profile 명령을 사용하여 `s3access-profile`라는 인스턴스 프로파일을 만듭니다.

```
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"
    }
}
```

6. `s3access` 인스턴스 프로파일에 `s3access-profile` 역할을 추가합니다.

```
aws iam add-role-to-instance-profile --instance-profile-name s3access-profile --role-
name s3access
```

또는 다음 Windows PowerShell용 AWS 도구 명령을 사용합니다.

- New-IAMRole
- Register-IAMRolePolicy
- New-IAMInstanceProfile

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 역할 작업 문제 해결 단원을 참조하십시오.

IAM 역할로 인스턴스를 시작하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 대시보드에서 인스턴스 시작을 선택합니다.
3. AMI와 인스턴스 유형을 선택한 후 다음: 인스턴스 세부 정보 구성(Next: Configure Instance Details)을 선택합니다.
4. 인스턴스 세부 정보 구성(Configure Instance Details) 페이지의 IAM 역할에서 생성한 IAM 역할을 선택합니다.

   > **Note**
   >
   > IAM 역할 생성 시 생성된 인스턴스 프로파일 이름이 IAM 역할 목록에 표시됩니다. 콘솔을 사용하여 IAM 역할을 생성한 경우 인스턴스 프로파일이 자동으로 생성되어 역할과 동일한 이름이 지정된 상태입니다. IAM, API 또는 AWS CLI SDK를 사용하여 AWS 역할을 생성한 경우 인스턴스 프로파일의 이름이 다를 수 있습니다.
5. 기타 세부 정보를 구성하고 지침에 따라 마법사의 나머지 절차를 완료하거나, 검토 후 시작을 선택하여 기본 설정을 수락하고 인스턴스 시작 검토 페이지로 바로 이동합니다.
6. 설정을 검토한 다음 시작을 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.

   IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-
aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/
meta-data/iam/security-credentials/role_name
```

IMDSv1

```
[ec2-user ~]$ 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.

IAM 역할로 인스턴스를 시작하려면(AWS CLI)

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 역할이 없는 인스턴스에 연결하려면, 인스턴스가 `stopped` 또는 `running` 상태에 있어야 합니다.

IAM 역할을 인스턴스에 연결하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 나서 작업, 인스턴스 설정, IAM 역할 연결/바꾸기를 차례대로 선택합니다.
4. 인스턴스에 연결할 IAM 역할을 선택한 후 적용을 선택합니다.

IAM 역할을 인스턴스에 연결하려면(AWS CLI)

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 교체

IAM 역할이 이미 연결된 인스턴스의 IAM 역할을 교체하려면, 인스턴스는 `running` 상태에 있어야 합니다. 기존 인스턴스를 먼저 분리하지 않고 인스턴스에 대한 IAM 역할을 변경하려는 경우 이 작업을 수행할 수 있습니다. 예를 들어 인스턴스에서 실행 중인 애플리케이션에서 수행된 API 작업이 중단되지 않도록 하기 위해 이 작업을 수행할 수 있습니다.

인스턴스에서 IAM 역할을 대체하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 나서 작업, 인스턴스 설정, IAM 역할 연결/바꾸기를 차례대로 선택합니다.
4. 인스턴스에 연결할 IAM 역할을 선택한 후 적용을 선택합니다.

인스턴스에서 IAM 역할을 대체하려면(AWS CLI)

1. 필요한 경우 IAM 인스턴스 프로파일 연결을 설명하여 교체할 IAM 인스턴스 프로파일의 연결 ID를 가져옵니다.

```
aws ec2 describe-iam-instance-profile-associations
```

2. 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

IAM 역할 분리

실행 중이거나 중지된 인스턴스에서 IAM 역할을 분리할 수 있습니다.

인스턴스에서 IAM 역할 분리하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 나서 작업, 인스턴스 설정, IAM 역할 연결/바꾸기를 차례대로 선택합니다.
4. IAM 역할에서 역할 없음을 선택합니다. 적용을 선택합니다.
5. 확인 대화 상자에서 예, 분리를 선택합니다.

인스턴스에서 IAM 역할을 분리하려면(AWS CLI)

1. 필요한 경우 describe-iam-instance-profile-associations를 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-profile명령을 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

Linux 인스턴스의 인바운드 트래픽 권한 부여

보안 그룹을 통해 인스턴스로 들어오는 트래픽 및 인스턴스에 도달할 수 있는 트래픽의 유형을 제어할 수 있습니다. 예를 들어 홈 네트워크에 속하는 컴퓨터만 SSH을 사용하여 인스턴스에 액세스하도록 허용할 수 있습니다. 인스턴스가 웹 서버인 경우 외부 사용자가 웹 서버의 콘텐츠를 탐색할 수 있도록 모든 IP 주소가 HTTP 또는 HTTPS를 사용하여 인스턴스에 액세스하도록 허용할 수 있습니다.

기본 보안 그룹과 새로 생성한 보안 그룹에는 인터넷에서 해당 인스턴스에 대한 액세스를 허용하지 않는 기본 규칙이 포함되어 있습니다. 자세한 내용은 기본 보안 그룹 (p. 875) 및 사용자 지정 보안 그룹 (p. 875) 단원을 참조하십시오. 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 단원을 참조하십시오.

시작하기 전에

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 Check IP 서비스를 사용할 수도 있습니다. 고정 IP 주소 없이 ISP 또는 방화벽을 경유하여 연결하는 경우에는 클라이언트 컴퓨터가 사용하는 IP 주소의 범위를 알아내야 합니다.

> **Warning**
>
> `0.0.0.0/0`을 사용하면 모든 IPv4 주소에서 SSH을 사용하여 인스턴스에 액세스할 수 있습니다. `::/0`을 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.

EC2 Instance Connect을 사용하여 인스턴스에 대한 SSH 액세스를 지원할지 여부를 결정하십시오. EC2 Instance Connect을 사용하지 않을 경우, 이를 제거하거나 IAM 정책에서 `ec2-instance-connect:SendSSHPublicKey` 작업을 거부하는 것을 고려하십시오. 자세한 내용은 EC2 Instance Connect 제거 (p. 497) 그리고 EC2 Instance Connect의 IAM 사용자 권한 구성 (p. 494)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. Linux을 사용하여 IP 주소에서 SSH 인스턴스에 연결할 수 있게 하는 규칙을 보안 그룹에 추가해야 합니다.

보안 그룹에 IPv4를 통한 인바운드 SSH 트래픽에 대한 규칙을 추가하려면(콘솔)

1. Amazon EC2 콘솔의 탐색 창에서 인스턴스를 선택합니다. 인스턴스를 선택하고 설명 탭을 확인합니다. 인스턴스에 연결된 보안 그룹이 보안 그룹에 나열됩니다. 인바운드 규칙 보기를 선택하여 인스턴스에 적용되고 있는 규칙 목록을 표시합니다.
2. 탐색 창에서 보안 그룹을 선택합니다. 인스턴스에 연결된 보안 그룹 중 하나를 선택합니다.
3. 세부 정보 창의 인바운드 탭에서 편집을 선택합니다. 대화 상자에서 규칙 추가를 선택하고 유형 목록에서 SSH를 선택합니다.

4. 필드를 로컬 컴퓨터의 퍼블릭 IPv4 주소로 자동으로 채우려면 소스 필드에서 내 IP를 선택하면 됩니다. 또는 사용자 지정을 선택하고 컴퓨터 또는 네트워크의 퍼블릭 IPv4 주소를 CIDR 표기법으로 지정해도 됩니다. 예를 들어, IPv4 주소가 `203.0.113.25`인 경우 이 단일 IPv4 주소를 CIDR 표기법으로 나열하려면 `203.0.113.25/32`를 지정합니다. 회사에서 주소를 범위로 할당하는 경우 전체 범위(예: `203.0.113.0/24`)를 지정합니다.

   IP 주소 확인에 대한 자세한 내용은 시작하기 전에 (p. 859) 단원을 참조하십시오.
5. 저장을 선택합니다.

IPv6 주소로 인스턴스를 시작했는데 IPv6 주소를 사용해 인스턴스에 연결하려면 SSH를 통한 인바운드 IPv6 트래픽을 허용하는 규칙을 추가해야 합니다.

보안 그룹에 IPv6를 통한 인바운드 SSH 트래픽에 대한 규칙을 추가하려면(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 보안 그룹을 선택합니다. 인스턴스에 대한 보안 그룹을 선택합니다.
3. 인바운드, 편집, 규칙 추가를 선택합니다.
4. 유형의 경우 SSH를 선택합니다.
5. 소스 필드에 컴퓨터의 IPv6 주소를 CIDR 표기법으로 지정합니다. 예를 들어, IPv6 주소가 `2001:db8:1234:1a00:9691:9503:25ad:1761`인 경우 단일 IP 주소를 CIDR 표기법으로 나열하려면 `2001:db8:1234:1a00:9691:9503:25ad:1761/128`을 지정합니다. 회사에서 주소를 범위로 할당하는 경우 전체 범위(예: `2001:db8:1234:1a00::/64`)를 지정합니다.
6. 저장을 선택합니다.

   Note

   인스턴스 자체가 아닌 로컬 시스템에서 다음 명령을 실행해야 합니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

명령줄을 사용하여 보안 그룹에 규칙을 추가하려면 다음을 수행합니다.

1. 다음 명령 중 하나를 사용하여 인스턴스와 관련된 보안 그룹을 찾습니다.

   - describe-instance-attribute(AWS CLI)

   ```
   aws ec2 describe-instance-attribute --instance-id instance_id --attribute groupSet
   ```

   - Get-EC2InstanceAttribute(Windows PowerShell용 AWS 도구)

   ```
   PS C:\> (Get-EC2InstanceAttribute -InstanceId instance_id -Attribute groupSet).Groups
   ```

   두 명령 모두 다음 단계에서 사용하는 보안 그룹 ID를 반환합니다.
2. 다음 명령 중 하나를 사용하여 보안 그룹에 규칙을 추가하십시오.

   - authorize-security-group-ingress(AWS CLI)

   ```
   aws ec2 authorize-security-group-ingress --group-id security_group_id --protocol tcp
    --port 22 --cidr cidr_ip_range
   ```

   - Grant-EC2SecurityGroupIngress(Windows PowerShell용 AWS 도구)

   `Grant-EC2SecurityGroupIngress` 명령에는 보안 그룹 규칙에 사용할 프로토콜, 포트 범위 및 IP 주소 범위를 설명하는 `IpPermission` 파라미터가 필요합니다. 다음 명령은 `IpPermission` 파라미터를 생성합니다.

```
PS C:\> $ip1 = @{ IpProtocol="tcp"; FromPort="22"; ToPort="22";
 IpRanges="cidr_ip_range" }
```

```
PS C:\> Grant-EC2SecurityGroupIngress -GroupId security_group_id -IpPermission
 @($ip1)
```

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.

인스턴스를 시작한 이후에는 해당 보안 그룹을 변경할 수 없습니다. 자세한 내용은 Amazon VPC 사용 설명서의 인스턴스의 보안 그룹 변경 단원을 참조하십시오.

Amazon EC2 키 페어

Amazon EC2는 퍼블릭 키 암호화 기법을 사용하여 로그인 정보를 암호화 및 복호화합니다. 퍼블릭 키 암호화 기법은 퍼블릭 키를 사용하여 데이터를 암호화하고, 수신자가 프라이빗 키를 사용하여 해당 데이터를 복호화하는 방식입니다. 퍼블릭 키와 프라이빗 키를 키 페어라고 합니다. 퍼블릭 키 암호화 기법을 사용하면 암호 대신 프라이빗 키를 사용하여 인스턴스에 안전하게 액세스할 수 있습니다.

인스턴스를 시작할 때 키 페어를 지정합니다. 기존 키 페어를 지정할 수도 있고 시작 시 새 키 페어를 생성하여 지정할 수도 있습니다. 부팅 시 퍼블릭 키 내용이 인스턴스에 `~/.ssh/authorized_keys`의 항목으로 배치됩니다. 인스턴스에 로그인하려면 인스턴스에 연결할 때 프라이빗 키를 지정해야 합니다. 자세한 내용은 인스턴스 시작 (p. 430) 및 Linux 인스턴스에 연결 (p. 485) 단원을 참조하십시오..

키 페어 만들기

Amazon EC2를 사용하여 키 페어를 만들 수 있습니다. 자세한 내용은 Amazon EC2를 사용해 키 페어 생성 (p. 862) 단원을 참조하십시오.

또는 타사 도구를 사용하여 Amazon EC2로 퍼블릭 키를 가져올 수도 있습니다. 자세한 내용은 Amazon EC2로 사용자의 퍼블릭 키 가져오기 (p. 863) 단원을 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. 리전당 최대 5,000개의 키 페어를 가질 수 있습니다.

인스턴스 시작 및 인스턴스로의 연결

인스턴스를 시작할 때는 인스턴스에 연결하는 데 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. 868) 단원을 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. 535) 단원을 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 AWS 계정 루트 사용자에 사용하는 것과 동일한 프라이빗 키 파일을 여러 사용자에게 배포하지 않아도 됩니다.

목차

- Amazon EC2를 사용해 키 페어 생성 (p. 862)
- Amazon EC2로 사용자의 퍼블릭 키 가져오기 (p. 863)
- 키 페어에 맞는 퍼블릭 키 검색(Linux) (p. 865)
- 키 페어에 맞는 퍼블릭 키 검색(Windows) (p. 865)
- 인스턴스에서 키 페어의 퍼블릭 키 검색 (p. 865)
- 키 페어의 지문 확인 (p. 866)
- 키 페어 삭제 (p. 866)
- 인스턴스의 키 페어 추가 또는 교체 (p. 867)
- 프라이빗 키를 분실했을 때 Linux 인스턴스에 연결하는 방법 (p. 868)

Amazon EC2를 사용해 키 페어 생성

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 인스턴스의 키 페어 추가 또는 교체 (p. 867) 단원을 참조하십시오.

다음 방법 중 하나를 사용하여 키 페어를 만들 수 있습니다.

새로운 콘솔

키 페어를 만들려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Key Pairs]를 선택합니다.
3. Create key pair(키 페어 생성)를 선택합니다.
4. 이름에 키 페어를 설명하는 이름을 입력합니다.
5. 파일 형식에서 프라이빗 키를 저장할 형식을 선택합니다. OpenSSH에서 사용할 수 있는 형식으로 프라이빗 키를 저장하려면 pem을 선택합니다. PuTTY에서 사용할 수 있는 형식으로 프라이빗 키를 저장하려면 ppk를 선택합니다.
6. Create key pair(키 페어 생성)를 선택합니다.

기존 콘솔

키 페어를 만들려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창의 NETWORK & SECURITY에서 Key Pairs를 선택합니다.

Note

탐색 창은 Amazon EC2 콘솔의 왼쪽에 있습니다. 창이 보이지 않는 경우 창이 최소화되었을 수 있으니 화살표를 선택해 확대하십시오.

3. 키 페어 생성을 선택합니다.
4. 키 페어 이름에서 새 키 페어의 이름을 입력한 다음 생성을 선택합니다.
5. 브라우저에서 프라이빗 키 파일이 자동으로 다운로드됩니다. 기본 파일 이름은 키 페어의 이름으로 지정된 이름이며, 파일 이름 확장명은 `.pem`입니다. 안전한 장소에 프라이빗 키 파일을 저장합니다.

   Important

   이때가 사용자가 프라이빗 키 파일을 저장할 수 있는 유일한 기회입니다. 인스턴스를 시작할 때 키 페어의 이름을 제공하고, 인스턴스에 연결할 때마다 해당 프라이빗 키를 제공해야 합니다.
6. macOS 또는 Linux 컴퓨터에서 SSH 클라이언트를 사용하여 Linux 인스턴스에 연결하려면 사용자만 프라이빗 키 파일을 읽을 수 있도록 다음 명령으로 해당 파일의 권한을 설정합니다.

```
chmod 400 my-key-pair.pem
```

   이러한 권한을 설정하지 않으면 이 키 페어를 사용하여 인스턴스에 연결할 수 없습니다. 자세한 내용은 오류: 보호되지 않는 프라이빗 키 파일 (p. 1090) 단원을 참조하십시오.

AWS CLI

키 페어를 만들려면

create-key-pair AWS CLI 명령을 사용합니다.

PowerShell

키 페어를 만들려면

New-EC2KeyPair Windows PowerShell용 AWS 도구 명령을 사용합니다.

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 사용하여 키 페어를 만들 수 있습니다. 또는 Java, Ruby, Python 등 각종 프로그래밍 언어에서 제공하는 표준 라이브러리를 사용하여 RSA 키 페어를 만들어도 됩니다.

요구 사항

- 지원되는 형식은 다음과 같습니다.
  - OpenSSH 퍼블릭 키 형식 (`~/.ssh/authorized_keys` 형식). EC2 Instance Connect API를 사용하는 동안 SSH를 사용하여 연결하는 경우 SSH2 형식도 지원됩니다.
  - Base64 인코딩 DER 형식
  - RFC4716에 지정된 SSH 퍼블릭 키 파일 형식
  - SSH 프라이빗 키 파일 형식은 PEM이어야 합니다(예를 들어 `ssh-keygen -m PEM`을 사용하여 OpenSSH 키를 PEM 형식으로 변환).
- RSA 키를 생성하십시오. Amazon EC2에서 DSA 키는 허용되지 않습니다.
- 지원되는 길이는 1024, 2048, 4096입니다. EC2 Instance Connect API를 사용하는 동안 SSH를 사용하여 연결하는 경우 지원되는 길이는 2048 및 4096입니다.

타사 도구를 이용한 키 페어 만들기

1. 타사 도구로 원하는 키 페어를 생성합니다.
2. 퍼블릭 키는 로컬 파일에 저장합니다. 예: `~/.ssh/my-key-pair.pub`(Linux) 또는 `C:\keys\my-key-pair.pub`(Windows). 이 파일의 파일 이름 확장자는 중요하지 않습니다.
3. 프라이빗 키를 확장자가 `.pem`인 다른 로컬 파일에 저장합니다. 예: `~/.ssh/my-key-pair.pem`(Linux) 또는 `C:\keys\my-key-pair.pem`(Windows). 프라이빗 키 파일은 안전한 장소에 저장합니다. 인스턴스를 시작할 때 키 페어의 이름을 제공하고, 인스턴스에 연결할 때마다 해당 프라이빗 키를 제공해야 합니다.

키 페어를 만든 후 다음 방법 중 하나를 사용하여 키 페어를 Amazon EC2로 가져옵니다.

새로운 콘솔

퍼블릭 키 가져오기

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Key Pairs]를 선택합니다.
3. 키 페어 가져오기를 선택합니다.
4. 이름에 키 페어를 설명하는 이름을 입력합니다.
5. 찾아보기를 선택하여 퍼블릭 키를 탐색하고 선택하거나 퍼블릭 키의 내용을 퍼블릭 키 내용 필드에 붙여 넣습니다.
6. 키 페어 가져오기를 선택합니다.
7. 가져온 키 페어가 키 페어 목록에 나타나는지 확인합니다.

기존 콘솔

퍼블릭 키 가져오기

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창의 NETWORK & SECURITY에서 Key Pairs를 선택합니다.
3. 키 페어 가져오기를 선택합니다.
4. 키 페어 가져오기 대화 상자에서 찾아보기를 선택하고 이전에 저장한 퍼블릭 키 파일을 선택합니다. 키 페어 이름 필드에 키 페어의 이름을 입력하고 가져오기를 선택합니다.
5. 가져온 키 페어가 키 페어 목록에 나타나는지 확인합니다.

AWS CLI

퍼블릭 키 가져오기

import-key-pair AWS CLI 명령을 사용합니다.

키 페어를 성공적으로 가져왔는지 확인하려면

describe-key-pairs AWS CLI 명령을 사용합니다.

PowerShell

퍼블릭 키 가져오기

Import-EC2KeyPair Windows PowerShell용 AWS 도구 명령을 사용합니다.

키 페어를 성공적으로 가져왔는지 확인하려면

Get-EC2KeyPair Windows PowerShell용 AWS 도구 명령을 사용합니다.

키 페어에 맞는 퍼블릭 키 검색(Linux)

로컬 Linux 또는 macOS 컴퓨터에서 ssh-keygen 명령을 사용하여 키 페어의 퍼블릭 키를 검색할 수 있습니다. 프라이빗 키(`.pem` 파일)를 다운로드한 경로를 지정합니다.

```
ssh-keygen -y -f /path_to_key_pair/my-key-pair.pem
```

이 명령은 다음 예제와 같이 퍼블릭 키를 반환합니다.

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnBItntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0FzZ
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb7OzlPnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE
```

명령이 실패하는 경우 다음 명령을 실행하여 사용자 자신만 볼 수 있도록 키 페어 파일에 대한 권한이 변경되어 있는지 확인합니다.

```
chmod 400 my-key-pair.pem
```

키 페어에 맞는 퍼블릭 키 검색(Windows)

로컬 Windows 컴퓨터에서 PuTTYgen을 사용하여 키 페어의 퍼블릭 키를 가져올 수 있습니다.

PuTTYgen을 시작하고 로드를 선택합니다. `.ppk` 및 `.pem` 파일을 선택합니다. PuTTYgen의 OpenSSH authorized_keys 파일에 붙여 넣기 위한 퍼블릭 키 아래에 퍼블릭 키가 표시됩니다. 또한 퍼블릭 키 저장을 선택하고, 파일 이름을 지정한 다음, 파일을 저장하고, 파일을 열어서 퍼블릭 키를 볼 수도 있습니다.

인스턴스에서 키 페어의 퍼블릭 키 검색

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.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-
ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-
data/public-keys/0/openssh-key
```

다음은 예제 출력입니다.

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnBItntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0FzZ
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb7OzlPnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE my-key-pair
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
```

다음은 예제 출력입니다.

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnBItntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0FzZ
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb7OzlPnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE my-key-pair
```

인스턴스에 연결하는 데 사용하는 키 페어를 변경하면 새 퍼블릭 키가 표시되도록 인스턴스 메타데이터가 업데이트되지 않습니다. 대신 인스턴스 메타데이터는 인스턴스를 시작할 때 지정한 키 페어의 퍼블릭 키를 계속 표시합니다. 자세한 내용은 인스턴스 메타데이터 가져오기 (p. 576) 단원을 참조하십시오.

Linux 인스턴스에서는 퍼블릭 키 내용이 `~/.ssh/authorized_keys` 내 항목에 있습니다. 이 파일을 편집기에서 열 수 있습니다. 다음은 이름이 **`my-key-pair`**인 키 페어에 대한 예시 항목입니다. 항목은 퍼블릭 키와 이 키 페어의 이름 순서로 구성됩니다.

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnBItntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0FzZ
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb7OzlPnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE my-key-pair
```

키 페어의 지문 확인

Amazon EC2 콘솔에서 키 페어 페이지를 보면 지문 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.

키 페어 페이지에 표시된 SSH2 지문을 사용하여 로컬 컴퓨터에 있는 프라이빗 키가 AWS에 저장된 퍼블릭 키와 일치하는지 확인할 수 있습니다. 프라이빗 키 파일을 다운로드한 컴퓨터에서 프라이빗 키 파일에서 SSH2 지문을 생성합니다. 이 경우 출력이 콘솔에 표시된 지문과 일치해야 합니다.

AWS를 사용하여 키 페어를 생성한 경우 다음 예제와 같이 OpenSSL 도구를 사용하여 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 생성하고 AWS에 퍼블릭 키를 업로드한 경우, OpenSSL 도구를 사용하여 다음 예제와 같이 지문을 생성할 수 있습니다.

```
$ openssl rsa -in path_to_private_key -pubout -outform DER | openssl md5 -c
```

OpenSSH 7.8 이상을 사용하여 OpenSSH 키 페어를 생성하고 AWS에 퍼블릭 키를 업로드한 경우 ssh-keygen을 사용하여 다음 예제와 같이 지문을 생성할 수 있습니다.

```
$ ssh-keygen -ef path_to_private_key -m PEM | openssl rsa -RSAPublicKey_in -outform DER |
 openssl md5 -c
```

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 수는 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 Amazon EC2 Auto Scaling에서 대체 인스턴스를 시작하지만, 키 페어를 찾을 수 없으면 인스턴스 시작에 실패합니다.

다음 방법 중 하나를 사용하여 키 페어를 삭제할 수 있습니다.

새로운 콘솔

키 페어를 삭제하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Key Pairs]를 선택합니다.
3. 삭제할 키 페어를 선택하고 삭제를 선택합니다.
4. 확인 필드에 `Delete`를 입력한 다음 삭제를 선택합니다.

기존 콘솔

키 페어를 삭제하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창의 NETWORK & SECURITY에서 Key Pairs를 선택합니다.
3. 키 페어를 선택하고 삭제를 선택합니다.
4. 메시지가 나타나면 예를 선택합니다.

AWS CLI

키 페어를 삭제하려면

delete-key-pair AWS CLI 명령을 사용합니다.

PowerShell

키 페어를 삭제하려면

Remove-EC2KeyPair Windows PowerShell용 AWS 도구 명령을 사용합니다.

> **Note**
>
> 인스턴스에서 Linux AMI를 생성한 후 AMI를 사용하여 다른 리전 또는 계정에서 새 인스턴스를 시작하는 경우 새 인스턴스에 원본 인스턴스의 키 페어가 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 페어를 사용하여 원격 액세스 제공에 대한 자세한 내용은 Linux 인스턴스의 사용자 계정 관리 (p. 535) 단원을 참조하십시오.

인스턴스의 키 페어 추가 또는 교체

인스턴스의 기본 시스템 계정에 액세스하는 데 사용되는 키 페어를 변경할 수 있습니다. 예를 들어 조직 내 사용자가 별도의 키 페어를 사용하여 시스템 사용자 계정에 액세스해야 할 경우 인스턴스에 키 페어를 추가

할 수 있습니다. 또는 누군가가(예: 퇴사자) `.pem` 파일의 사본을 보유하고 있어 이를 통한 인스턴스 연결을 금지하려는 경우 키 페어를 새 페어로 교체할 수 있습니다.

> **Note**
>
> 이러한 절차는 기본 사용자 계정(예: `ec2-user`)의 키 페어를 수정하는 것입니다. 인스턴스에 사용자 계정을 추가하는 것에 대한 자세한 내용은 Linux 인스턴스의 사용자 계정 관리 (p. 535) 단원을 참조하십시오.

키 페어를 추가 또는 교체하는 방법

1. Amazon EC2 콘솔 (p. 862) 또는 타사 도구 (p. 863)를 사용하여 새 키 페어를 만듭니다.
2. 새 키 페어에서 퍼블릭 키를 검색합니다. 자세한 내용은 키 페어에 맞는 퍼블릭 키 검색(Linux) (p. 865) 또는 키 페어에 맞는 퍼블릭 키 검색(Windows) (p. 865) 단원을 참조하십시오.
3. 기존의 프라이빗 키 파일을 사용하여 인스턴스에 연결합니다.
4. 원하는 텍스트 편집기를 사용하여 인스턴스에서 `.ssh/authorized_keys` 파일을 엽니다. 새 키 페어의 퍼블릭 키 정보를 기존 퍼블릭 키 정보 아래에 붙여넣습니다. 파일을 선택합니다.
5. 인스턴스 연결을 해제하고 새 프라이빗 키 파일을 사용하여 인스턴스에 연결할 수 있는지 테스트합니다.
6. (선택 사항) 기존 키 페어를 교체하는 경우 인스턴스에 연결하고 `.ssh/authorized_keys` 파일에서 원래 키 페어의 퍼블릭 키 정보를 삭제합니다.

> **Note**
>
> Auto Scaling 그룹을 사용 중인 경우(예: Elastic Beanstalk 환경에서 사용), 교체하려는 키 페어가 시작 구성에서 지정되지 않았는지 확인하십시오. 비정상 인스턴스가 발견될 경우 Amazon EC2 Auto Scaling에서 대체 인스턴스를 시작하지만, 키 페어를 찾을 수 없으면 인스턴스 시작에 실패합니다.

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 분리한 후 다른 인스턴스에 데이터 볼륨으로 연결하고, 새 퍼블릭 키로 `authorized_keys` 파일을 수정하고, 해당 볼륨을 원본 인스턴스로 되돌린 뒤 인스턴스를 다시 시작합니다. 인스턴스 시작, 연결, 중지에 대한 자세한 내용은 인스턴스 수명 주기 (p. 426)에서 확인하십시오.

인스턴스 스토어를 지원하는 루트 볼륨이 있는 인스턴스에는 이 절차가 지원되지 않습니다. 인스턴스의 루트 디바이스 유형을 확인하려면 Amazon EC2 콘솔을 열고 인스턴스를 선택하여 인스턴스를 선택하고 세부 정보 창에서 루트 디바이스 유형의 값을 확인합니다. 이때 값은 `ebs` 또는 `instance store`입니다. 루트 디바이스가 인스턴스 스토어 볼륨인 경우 이 절차를 사용하여 인스턴스에 대한 액세스 권한을 다시 얻을 수 없습니다. 인스턴스에 연결하려면 프라이빗 키가 있어야 합니다.

키 페어가 다른 EBS 지원 인스턴스에 연결하는 단계

- 1단계: 새 키 페어 생성 (p. 869)
- 2단계: 원본 인스턴스와 루트 볼륨에 대한 정보 가져오기 (p. 869)
- 3단계: 원본 인스턴스 중지 (p. 869)
- 4단계: 임시 인스턴스 시작 (p. 869)
- 5단계: 원본 인스턴스에서 루트 볼륨을 분리하고 임시 인스턴스에 연결 (p. 869)
- 6단계: 임시 인스턴스에 마운트된 원본 볼륨의 authorized_keys에 새 퍼블릭 키 추가 (p. 870)
- 7단계: 임시 인스턴스에서 원본 볼륨을 마운트 해제하고 분리한 다음 원본 인스턴스에 다시 연결 (p. 871)
- 8단계: 새 키 페어를 사용하여 원본 인스턴스에 연결 (p. 872)

- 9단계: 정리 (p. 872)

1단계: 새 키 페어 생성

새 키 페어는 Amazon EC2 콘솔이나 타사 도구를 사용해 만들 수 있습니다. 새 키 페어의 이름을 잃어버린 프라이빗 키와 동일하게 지정하려면 먼저 기존 키 페어를 삭제해야 합니다. 새 키 페어 생성에 대한 자세한 내용은 Amazon EC2를 사용해 키 페어 생성 (p. 862) 또는 Amazon EC2로 사용자의 퍼블릭 키 가져오기 (p. 863) 단원을 참조하십시오.

2단계: 원본 인스턴스와 루트 볼륨에 대한 정보 가져오기

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택한 후 연결할 인스턴스를 선택합니다. (이후의 내용에서는 이를 원본 인스턴스라고 지칭함)
3. 설명 탭에서 이 절차를 완료하는 데 필요한 다음 정보를 저장하십시오.
   - 원본 인스턴스의 인스턴스 ID, AMI ID 및 가용 영역을 메모합니다.
   - 루트 디바이스 필드에서 루트 볼륨의 디바이스 이름(예: `/dev/sda1` 또는 `/dev/xvda`)을 기록합니다 링크를 선택하고 EBS ID 필드에 있는 볼륨 ID(vol-xxxxxxxxxxxxxxxxx)를 메모합니다.

3단계: 원본 인스턴스 중지

작업을 선택하고 인스턴스 상태를 선택한 후 정지를 선택합니다. 중지가 비활성화되어 있으면 해당 인스턴스가 이미 중지되었거나 루트 디바이스가 인스턴스 스토어 볼륨인 것입니다.

> **Warning**
>
> 인스턴스를 중지하면 인스턴스 스토어 볼륨의 데이터가 삭제됩니다. 인스턴스 스토어 볼륨의 데이터를 유지하려면 영구 스토리지에 백업하십시오.

4단계: 임시 인스턴스 시작

인스턴스 시작을 선택한 후 시작 마법사를 사용하여 다음 옵션으로 임시 인스턴스를 시작합니다.

- AMI 선택 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. 112) 단원을 참조하십시오.
- 인스턴스 유형 선택 페이지에서 마법사에 의해 자동 선택된 기본 인스턴스 유형을 그대로 유지합니다.
- 인스턴스 세부 정보 구성 페이지에서 원본 인스턴스와 동일한 가용 영역을 지정합니다. VPC에서 인스턴스를 시작하는 경우 가용 영역에서 서브넷을 선택합니다.
- 태그 추가 페이지에서 인스턴스에 `Name=Temporary` 태그를 추가하여 임시 인스턴스임을 표시합니다.
- 검토 페이지에서 시작을 선택합니다. 새 키 페어를 생성하고, 컴퓨터에서 안전한 위치에 다운로드한 후 인스턴스 시작을 선택합니다.

5단계: 원본 인스턴스에서 루트 볼륨을 분리하고 임시 인스턴스에 연결

1. 탐색 창에서 볼륨을 선택하고 원본 인스턴스에 대한 루트 디바이스 볼륨을 선택합니다(전 단계에서 기록해 둔 볼륨 ID). 작업, 볼륨 분리를 선택하고 예, 분리를 선택합니다. 볼륨이 `available` 상태가 될 때까지 기다리십시오. (새로 고침 아이콘을 클릭해야 할 수도 있습니다.)

2. 해당 볼륨을 선택한 상태에서 작업을 선택한 후 볼륨 연결을 선택합니다. 임시 인스턴스의 인스턴스 ID를 선택하고 디바이스에서 지정된 디바이스(예: `/dev/sdf`)를 메모한 후 연결을 선택합니다.

   Note

   AWS Marketplace AMI에서 원본 인스턴스를 시작했고 볼륨에 AWS Marketplace 코드가 포함되어 있는 경우 볼륨을 연결하기 전에 먼저 임시 인스턴스를 중지해야 합니다.

6단계: 임시 인스턴스에 마운트된 원본 볼륨의 `authorized_keys`에 새 퍼블릭 키 추가

1. 임시 인스턴스에 연결합니다.
2. 임시 인스턴스에서 인스턴스에 연결한 볼륨을 마운트해야 해당 파일 시스템에 액세스할 수 있습니다. 예를 들어 디바이스 이름이 `/dev/sdf`인 경우 다음 명령을 사용하여 볼륨을 `/mnt/tempvol`로 마운트합니다.

   Note

   디바이스 이름은 인스턴스에서 다르게 표시될 수 있습니다. 예를 들면 `/dev/sdf`로 탑재된 디바이스가 인스턴스에서는 `/dev/xvdf`로 표시되기도 합니다. Red Hat 중 일부 버전(CentOS 등 변형 버전 포함)은 후행 문자가 4자씩 늘어나기도 하며, 이 경우 `/dev/sdf`가 `/dev/xvdk`로 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
```

   위 예에서 `/dev/xvda` 및 `/dev/xvdf`는 파티션 볼륨이고, `/dev/xvdg`는 파티션 볼륨이 아닙니다. 볼륨이 파티셔닝된 경우 이후의 단계에서는 원시 디바이스(`/dev/xvdf1`)) 대신에 파티션(`/dev/xvdf`)을 마운트해야 합니다.

   b. 임시 디렉터리를 만들어 볼륨을 마운트합니다.

```
[ec2-user ~]$ sudo mkdir /mnt/tempvol
```

   c. 임시 탑재 지점에 볼륨(또는 파티션)을 탑재하되, 이전에 인식된 볼륨 이름이나 디바이스 이름을 사용합니다. 필요한 명령은 사용자 운영 체제의 파일 시스템에 따라 다릅니다.

   - Amazon Linux, Ubuntu 및 Debian

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

   - Amazon Linux 2, CentOS, SLES 12 및 RHEL 7.x

```
[ec2-user ~]$ sudo mount -o nouuid /dev/xvdf1 /mnt/tempvol
```

   Note

   파일 시스템이 손상되었다는 오류가 발생한다면, 다음 명령을 실행해 fsck 유틸리티를 사용하여 파일 시스템을 확인하고 문제를 해결하십시오.

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

3. 임시 인스턴스에서 다음 명령을 사용하여 마운팅된 볼륨의 `authorized_keys`를 임시 인스턴스에 대한`authorized_keys`의 새로운 퍼블릭 키로 업데이트합니다.

   Important

   다음 예는 Amazon Linux 사용자 이름 `ec2-user`를 사용합니다. Ubuntu 인스턴스의 경우 `ubuntu`처럼 다른 사용자 이름으로 바꿔야 할 수 있습니다.

```
[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 sudo를 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. 그런 다음 sudo를 사용하여 실패한 복사 명령을 다시 실행합니다.

```
[ec2-user ~]$ sudo cp .ssh/authorized_keys /mnt/tempvol/home/ec2-user/.ssh/
authorized_keys
```

권한이 변경되었는지 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
```

7단계: 임시 인스턴스에서 원본 볼륨을 마운트 해제하고 분리한 다음 원본 인스턴스에 다시 연결

1. 임시 인스턴스에서 연결된 볼륨을 마운트 해제해야 이 볼륨을 원본 인스턴스에 다시 연결할 수 있습니다. 예를 들어 다음 명령을 사용하면 `/mnt/tempvol`에서 볼륨을 탑재 해제할 수 있습니다.

```
[ec2-user ~]$ sudo umount /mnt/tempvol
```

2. 임시 인스턴스에서 볼륨 분리(이전 단계에서 볼륨 마운트 해제함): Amazon EC2 콘솔에서 원본 인스턴스의 루트 디바이스 볼륨을 선택하고(이전 단계에서 볼륨 ID 기록함) 작업, 볼륨 분리를 선택하고 예, 분리를 선택합니다. 볼륨이 `available` 상태가 될 때까지 기다리십시오. (새로 고침 아이콘을 클릭해야 할 수도 있습니다.)
3. 볼륨을 원본 인스턴스에 다시 연결: 볼륨을 선택한 상태에서 작업, 볼륨 연결을 선택합니다. 원본 인스턴스의 인스턴스 ID를 선택하고, 앞서 원래 루트 디바이스 연결을 위해 메모한 디바이스 이름(`/dev/sda1` 또는 `/dev/xvda`)을 지정한 뒤 연결을 선택합니다.

Important

원래 연결 디바이스와 같은 이름을 지정하지 않으면 원본 인스턴스를 시작할 수 없습니다. Amazon EC2는 루트 디바이스 볼륨이 `sda1` 또는 `/dev/xvda`에 있다고 예상합니다.

8단계: 새 키 페어를 사용하여 원본 인스턴스에 연결

원본 인스턴스를 선택하고 작업을 선택한 후 인스턴스 상태를 선택하고 시작을 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.

9단계: 정리

(선택 사항) 임시 인스턴스를 더 이상 사용하지 않는 경우 해당 인스턴스는 종료해도 됩니다. 임시 인스턴스를 선택하고 작업을 선택한 후 인스턴스 상태를 선택하고 종료를 선택합니다.

Amazon EC2 인스턴스에 대한 Linux 보안 그룹

보안 그룹은 하나 이상의 인스턴스에 대한 트래픽을 제어하는 가상 방화벽 역할을 합니다. 인스턴스를 시작할 때 하나 이상의 보안 그룹을 지정할 수 있습니다. 지정하지 않으면 기본 보안 그룹을 사용합니다. 연결된 인스턴스에서 트래픽을 주고 받을 수 있도록 하는 규칙을 각 보안 그룹에 추가할 수 있습니다. 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.

VPC에서 인스턴스를 시작할 때 해당 VPC에 대해 생성된 보안 그룹을 지정해야 합니다. 인스턴스를 시작한 이후에는 해당 보안 그룹을 변경할 수 없습니다. 보안 그룹은 네트워크 인터페이스와 연결됩니다. 인스턴스의 보안 그룹을 변경하면 기본 네트워크 인터페이스(eth0)와 연결된 보안 그룹이 변경됩니다. 자세한 내용은 Amazon VPC 사용 설명서의 인스턴스의 보안 그룹 변경 단원을 참조하십시오. 다른 네트워크 인터페이스와 연결된 보안 그룹을 변경할 수도 있습니다. 자세한 내용은 보안 그룹 변경 (p. 701) 단원을 참조하십시오.

보안 그룹으로 충족되지 않는 요구 사항이 있는 경우 보안 그룹을 사용하면서 인스턴스에 대한 자체 방화벽을 유지합니다.

Windows 인스턴스에 대한 트래픽을 허용해야 하는 경우 Windows 인스턴스용 Amazon EC2 사용 설명서의 Windows 인스턴스에 대한 Amazon EC2 보안 그룹 단원을 참조하십시오.

목차

- 보안 그룹 규칙 (p. 873)
  - 연결 추적 (p. 874)
- 기본 보안 그룹 (p. 875)
- 사용자 지정 보안 그룹 (p. 875)
- 보안 그룹 작업 (p. 876)
  - 보안 그룹 생성 (p. 876)
  - 보안 그룹 설명 (p. 877)
  - 보안 그룹에 규칙 추가 (p. 877)
  - 보안 그룹 규칙 업데이트 (p. 878)
  - 보안 그룹에서 규칙 삭제 (p. 879)

- 보안 그룹 삭제 (p. 879)
- 보안 그룹 규칙 참조 (p. 880)
  - 웹 서버 규칙 (p. 880)
  - 데이터베이스 서버 규칙 (p. 881)
  - 컴퓨터에서 인스턴스 연결에 대한 규칙 (p. 882)
  - 보안 그룹이 동일한 인스턴스에서 인스턴스에 대한 연결 규칙 (p. 882)
  - 경로 MTU 검색 규칙 (p. 883)
  - Ping/ICMP 규칙 (p. 883)
  - DNS 서버 규칙 (p. 883)
  - Amazon EFS 규칙 (p. 884)
  - Elastic Load Balancing 규칙 (p. 884)
  - VPC 피어링 규칙 (p. 885)

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.
- 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. 874) 단원을 참조하십시오.
- 언제든지 규칙을 추가하고 제거할 수 있습니다. 변경 내용은 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. 874) 단원을 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.
- 소스 또는 대상: 트래픽에 대한 소스(인바운드 규칙) 또는 대상(아웃바운드 규칙). 다음 옵션 중 하나를 지정합니다.
  - 개별 IPv4 주소. `/32` 접두사 길이를 사용해야 합니다(예: `203.0.113.1/32`).

- 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`).
- CIDR 블록 표기법으로 표시된 IPv6 주소의 범위(예: `2001:db8:1234:1a00::/64`).
- (VPC에만 해당) AWS 서비스용 접두사 목록 ID. 예: `pl-1a2b3c4d`. 자세한 내용은 Amazon VPC 사용 설명서의 게이트웨이 VPC 엔드포인트를 참조하십시오.
- 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
  - 동일한 VPC에 대한 다른 보안 그룹
  - VPC 피어링 연결에서 동일한 VPC 또는 피어 VPC에 대한 다른 보안 그룹.
- (선택 사항) 설명: 나중에 쉽게 식별할 수 있도록 규칙에 대한 설명을 입력할 수 있습니다. 설명 길이는 최대 255자입니다. 허용되는 문자는 a-z, A-Z, 0-9, 공백 및 ._-:/()#,@[]+=;{}!$*.

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 EC2인스턴스 IP 어드레싱 (p. 658) 단원을 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 Guide의 무효 보안 그룹 규칙으로 작업 단원을 참조하십시오.

특정 포트에 대한 규칙이 여러 개 있는 경우 최대 허용 규칙을 적용합니다. 예를 들어, IP 주소 `203.0.113.1`의 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 추적되며, 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 포트(0-65535)에 대해 모든 응답 트래픽(`0.0.0.0/0`)을 허용하는 해당 규칙이 있을 경우 해당 트래픽 흐름은 추적되지 않습니다. 그러므로 응답 트래픽이 추적 정보가 아니라 응답 트래픽을 허용하는 인바운드 또는 아웃바운드를 기반으로 흐름이 허용됩니다.

다음 예제의 보안 그룹에는 TCP 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

TCP traffic on port 22 (SSH) to and from the instance is tracked, 인바운드 규칙은 `203.0.113.1/32`의 트래픽만 허용하며 모든 IP 주소(`0.0.0.0/0`)를 허용하는 것은 아니기 때문에 인스턴스에서 양방향으로 이동하는 포트 22(SSH)의 TCP 트래픽이 추적됩니다. 인바운드 및 아웃바운드 규칙이 모든 TCP 트래픽(`0.0.0.0/0`)을 허용하기 때문에 인스턴스에서 양방향으로 이동하는 포트 80(HTTP)의 TCP 트래픽이 추적되지 않습니다. 규칙에 관계없이 ICMP 트래픽은 항상 추적됩니다. 보안 그룹에서 아웃바운드 규칙을 제거할 경우 포트 80을 통해 전송되는 트래픽(HTTP)을 비롯하여 해당 인스턴스로/에서 송/수신되는 모든 트래픽이 추적됩니다.

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).

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 ACL을 참조하십시오.

기본 보안 그룹

AWS 계정에 각 지역의 기본 VPC에 대한 기본 보안 그룹이 자동으로 생성됩니다. 인스턴스를 시작할 때 보안 그룹을 지정하지 않을 경우 VPC에 대해 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 및 아웃바운드 규칙을 추가하거나 제거할 수 있습니다.

기본 보안 그룹을 삭제할 수 없습니다. 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.

a-z, A-Z, 0-9, 공백 및 ._-:/()#,@[]+=&;{}!$*

보안 그룹 이름은 sg-로 시작할 수 없습니다. 보안 그룹 이름은 VPC 내에서 고유해야 합니다.

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. 아웃바운드 규칙도 변경할 수 있습니다.

보안 그룹에 추가할 수 있는 규칙에 대한 자세한 내용은 보안 그룹 규칙 참조 (p. 880) 단원을 참조하십시오.

보안 그룹 작업

Amazon EC2 콘솔을 사용하여 보안 그룹과 보안 그룹 규칙을 생성하고 보고 업데이트하고 삭제할 수 있습니다.

작업

- 보안 그룹 생성 (p. 876)
- 보안 그룹 설명 (p. 877)
- 보안 그룹에 규칙 추가 (p. 877)
- 보안 그룹 규칙 업데이트 (p. 878)
- 보안 그룹에서 규칙 삭제 (p. 879)
- 보안 그룹 삭제 (p. 879)

보안 그룹 생성

Amazon EC2 콘솔을 이용해 사용자 지정 보안 그룹을 만들 수 있습니다. 보안 그룹을 생성할 VPC를 지정해야 합니다.

콘솔을 사용하여 보안 그룹을 새로 만들려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 보안 그룹을 선택합니다.
3. 보안 그룹 생성을 선택합니다.
4. 보안 그룹의 이름과 설명을 지정합니다.
5. VPC에서 VPC의 ID를 선택합니다.
6. 규칙을 추가할 수 있습니다. 또는 생성을 선택하여 지금 보안 그룹을 생성하고 나중에 규칙을 추가할 수 있습니다. 규칙 추가에 대한 자세한 내용은 보안 그룹에 규칙 추가 (p. 877) 단원을 참조하십시오.

명령줄을 사용하여 보안 그룹을 생성하려면

- create-security-group(AWS CLI)
- New-EC2SecurityGroup(Windows PowerShell용 AWS 도구)

Amazon EC2 콘솔을 사용하면 기존의 보안 그룹에서 새 보안 그룹으로 규칙을 복사할 수 있습니다.

콘솔을 사용하여 보안 그룹을 복사하려면

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

3. 복사할 보안 그룹을 선택하고 작업과 새로 복사를 차례로 선택합니다.
4. 보안 그룹 생성 대화 상자가 열리고 기존 보안 그룹의 규칙으로 채워집니다. 새 보안 그룹의 이름과 설명을 지정합니다. VPC에서 VPC의 ID를 선택합니다. 완료했으면 생성을 선택합니다.

인스턴스를 시작할 때 인스턴스에 보안 그룹을 할당할 수 있습니다. 규칙을 추가하거나 제거하면 해당 보안 그룹을 할당한 모든 인스턴스에 변경 내용이 자동으로 적용됩니다.

인스턴스를 시작한 이후에는 해당 보안 그룹을 변경할 수 없습니다. 자세한 내용은 Amazon VPC 사용 설명서의 인스턴스의 보안 그룹 변경 단원을 참조하십시오.

보안 그룹 설명

Amazon EC2 콘솔이나 명령줄을 사용하여 보안 그룹에 대한 정보를 볼 수 있습니다.

콘솔을 사용하여 보안 그룹을 설명하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 보안 그룹을 선택합니다.
3. (선택 사항) 필터 목록에서 VPC ID를 선택한 다음 VPC의 ID를 선택합니다.
4. 보안 그룹을 선택합니다. 설명 탭에 일반 정보가 표시되고, 인바운드 탭에 인바운드 규칙이 표시되고, 아웃바운드 탭에 아웃바운드 규칙이 표시되며, 태그 탭에는 태그가 표시됩니다.

명령줄을 사용하여 하나 이상의 보안 그룹을 나타내려면

- describe-security-groups(AWS CLI)
- Get-EC2SecurityGroup(Windows PowerShell용 AWS 도구)

보안 그룹에 규칙 추가

보안 그룹에 규칙을 추가할 경우 잠시 후 보안 그룹과 연결된 인스턴스에 새 규칙이 자동으로 적용됩니다.

특정 유형의 액세스를 위한 보안 그룹 규칙 선택에 대한 자세한 내용은 보안 그룹 규칙 참조 (p. 880) 단원을 참조하십시오.

콘솔을 사용하여 보안 그룹에 규칙을 추가하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 보안 그룹을 선택하고 보안 그룹을 선택합니다.
3. 인바운드 탭에서 편집을 선택합니다.
4. 대화 상자에서 규칙 추가를 선택하고 다음과 같이 실행합니다.
   - 유형에는 프로토콜을 선택합니다.
   - 사용자 지정 TCP 또는 UDP 프로토콜을 선택하는 경우 포트 범위에 포트 범위를 지정합니다.
   - 사용자 지정 ICMP 프로토콜을 선택하는 경우 프로토콜에서 ICMP 유형 이름을 선택하고, 해당되는 경우 포트 범위에서 코드 이름을 선택합니다.
   - 소스의 경우 다음 중 하나를 선택합니다.
     - 사용자 지정: 제공되는 필드에 IP 주소(CIDR 표기법), CIDR 블록 또는 다른 보안 그룹을 지정해야 합니다.
     - 위치 무관: `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, 위치 무관 옵션을 선택하면 IPv4 트래픽에 대해 한 개(`0.0.0.0/0`), IPv6 트래픽에 대해 한 개(`::/0`), 총 2개의 규칙이 생성됩니다.

- 내 IP: 로컬 컴퓨터의 퍼블릭 IPv4 주소를 자동으로 추가합니다.
- 필요한 경우 설명에 규칙에 대한 설명을 지정합니다.

추가할 수 있는 규칙의 유형에 대한 자세한 내용은 보안 그룹 규칙 참조 (p. 880) 단원을 참조하십시오.

5. 저장을 선택합니다.
6. 아웃바운드 규칙을 지정할 수도 있습니다. 아웃바운드 탭에서 편집, 규칙 추가를 선택하고 다음 작업을 수행합니다.
   - 유형에는 프로토콜을 선택합니다.
   - 사용자 지정 TCP 또는 UDP 프로토콜을 선택하는 경우 포트 범위에 포트 범위를 지정합니다.
   - 사용자 지정 ICMP 프로토콜을 선택하는 경우 프로토콜에서 ICMP 유형 이름을 선택하고, 해당되는 경우 포트 범위에서 코드 이름을 선택합니다.
   - 대상의 경우 다음 중 하나를 선택합니다.
     - 사용자 지정: 제공되는 필드에 IP 주소(CIDR 표기법), CIDR 블록 또는 다른 보안 그룹을 지정해야 합니다.
     - 위치 무관: `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, 위치 무관 옵션을 선택하면 IPv4 트래픽에 대해 한 개(`0.0.0.0/0`), IPv6 트래픽에 대해 한 개(`::/0`), 총 2개의 규칙이 생성됩니다.
     - 내 IP: 로컬 컴퓨터의 IP 주소를 자동으로 추가합니다.
   - 필요한 경우 설명에 규칙에 대한 설명을 지정합니다.
7. 저장을 선택합니다.

명령줄을 사용하여 보안 그룹에 수신 규칙을 한 개 이상 추가하려면

- authorize-security-group-ingress(AWS CLI)
- Grant-EC2SecurityGroupIngress(Windows PowerShell용 AWS 도구)

명령줄을 사용하여 보안 그룹에 발신 규칙을 한 개 이상 추가하려면

- authorize-security-group-egress(AWS CLI)
- Grant-EC2SecurityGroupEgress(Windows PowerShell용 AWS 도구)

보안 그룹 규칙 업데이트

콘솔을 사용하여 기존 보안 그룹의 프로토콜, 포트 범위 또는 소스/목적지를 수정하면 콘솔은 기존 규칙을 삭제하고 새 규칙을 추가합니다.

콘솔을 사용하여 보안 그룹 규칙을 업데이트하려면

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

3. 업데이트할 보안 그룹을 선택하고 인바운드 트래픽에 대한 규칙을 업데이트하려면 인바운드 규칙을, 아웃바운드 트래픽에 대한 규칙을 업데이트하려면 아웃바운드 규칙을 선택합니다.
4. 편집을 선택합니다. 필요에 따라 규칙 항목을 수정한 다음, 저장을 선택합니다.

Amazon EC2 API 또는 명령줄 도구를 사용하여 기존 규칙의 프로토콜, 포트 범위, 소스 및 대상 등을 업데이트하려 할 때 규칙을 수정할 수 없습니다. 대신 기존 규칙을 삭제하고 새 규칙을 추가해야 합니다. 규칙 설명을 업데이트하기 위해 update-security-group-rule-descriptions-ingress 및 update-security-group-rule-descriptions-egress 명령을 사용할 수 있습니다.

명령줄을 사용하여 수신 보안 그룹 규칙에서 설명을 업데이트하려면

- update-security-group-rule-descriptions-ingress(AWS CLI)
- Update-EC2SecurityGroupRuleIngressDescription(Windows PowerShell용 AWS 도구)

명령줄을 사용하여 발신 보안 그룹 규칙에서 설명을 업데이트하려면

- update-security-group-rule-descriptions-egress(AWS CLI)
- Update-EC2SecurityGroupRuleEgressDescription(Windows PowerShell용 AWS 도구)

보안 그룹에서 규칙 삭제

보안 그룹에서 규칙을 삭제할 경우 보안 그룹과 연결된 인스턴스에 해당 변경 내용이 자동으로 적용됩니다.

콘솔을 사용하여 보안 그룹 규칙을 삭제하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 보안 그룹을 선택합니다.
3. 보안 그룹을 선택합니다.
4. 인바운드 탭(인바운드 규칙) 또는 아웃바운드 탭(아웃바운드 규칙)에서 편집을 선택합니다. 삭제할 각 규칙 옆의 삭제(x 아이콘)를 선택합니다.
5. 저장을 선택합니다.

명령줄을 사용하여 보안 그룹에서 수신 규칙을 한 개 이상 제거하려면

- revoke-security-group-ingress(AWS CLI)
- Revoke-EC2SecurityGroupIngress(Windows PowerShell용 AWS 도구)

명령줄을 사용하여 보안 그룹에서 발신 규칙을 한 개 이상 제거하려면

- revoke-security-group-egress(AWS CLI)
- Revoke-EC2SecurityGroupEgress(Windows PowerShell용 AWS 도구)

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.

콘솔을 사용하여 보안 그룹을 삭제하려면

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

2. 탐색 창에서 보안 그룹을 선택합니다.
3. 보안 그룹을 선택한 다음 작업, 보안 그룹 삭제를 선택합니다.
4. 예, 삭제를 선택합니다.

명령줄을 사용하여 보안 그룹을 삭제하려면

- 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.

예제

- 웹 서버 규칙 (p. 880)
- 데이터베이스 서버 규칙 (p. 881)
- 컴퓨터에서 인스턴스 연결에 대한 규칙 (p. 882)
- 보안 그룹이 동일한 인스턴스에서 인스턴스에 대한 연결 규칙 (p. 882)
- 경로 MTU 검색 규칙 (p. 883)
- Ping/ICMP 규칙 (p. 883)
- DNS 서버 규칙 (p. 883)
- Amazon EFS 규칙 (p. 884)
- Elastic Load Balancing 규칙 (p. 884)
- VPC 피어링 규칙 (p. 885)

웹 서버 규칙

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
TCP	6	80(HTTP)	::/0	임의의 IPv6 주소에서 인바운드 HTTP 액세스를 허용함
TCP	6	443(HTTPS)	::/0	임의의 IPv6 주소에서 인바운드

프로토콜 유형	프로토콜 번호	포트	소스 IP	참고
				HTTPS 액세스를 허용함

데이터베이스 서버 규칙

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 RDS 사용 설명서를 참조하십시오.

원본 IP의 경우 다음 중 하나를 지정합니다.

- 로컬 네트워크의 특정 IP 주소 또는 IP 주소 범위(CIDR 블록 표기법)
- 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)

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

프로토콜 유형	프로토콜 번호	포트	목적지 IP	참고
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

컴퓨터에서 인스턴스 연결에 대한 규칙

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(CIDR 블록 표기법). 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(CIDR 블록 표기법). IPv6를 위해 VPC가 활성화되어 있고 인스턴스에 IPv6 주소가 있는 경우 IPv6 주소 또는 범위를 입력할 수 있습니다.

보안 그룹이 동일한 인스턴스에서 인스턴스에 대한 연결 규칙

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 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

경로 MTU 검색 규칙

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 과 같은 ICMP 메시지를 반환합니다.

```
Destination Unreachable: Fragmentation Needed and Don't Fragment was Set
```

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

Ping/ICMP 규칙

`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(CIDR 블록 표기법).

`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(CIDR 블록 표기법).

DNS 서버 규칙

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 IP 주소 또는 IP 주소 범위(CIDR 블록 표기법)
- 네트워크에서 DNS 서버로의 액세스를 필요로 하는 인스턴스 세트에 대한 보안 그룹의 ID

프로토콜 유형	프로토콜 번호	포트
TCP	6	53
UDP	17	53

Amazon EFS 규칙

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(CIDR 블록 표기법).	로컬 컴퓨터로부터의 인바운드 SSH 액세스를 허용합니다.

Elastic Load Balancing 규칙

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 정보는 Classic Load Balancer 사용 설명서의 Classic Load Balancer에 대한 보안 그룹 구성 및 Application Load Balancer 사용 설명서의 Application Load Balancer에 대한 보안 그룹을 참조하십시오.

VPC 피어링 규칙

피어링된 VPC의 보안 그룹을 참조하도록 VPC 보안 그룹의 인바운드 또는 아웃바운드 규칙을 업데이트할 수 있습니다. 그렇게 하면 피어링된 VPC의 참조 보안 그룹과 연결된 인스턴스 간에 트래픽을 주고받을 수 있습니다. VPC 피어링을 위한 보안 그룹을 구성하는 방법에 대한 자세한 내용은 피어 VPC 그룹을 참조하도록 보안 그룹 업데이트를 참조하십시오.

Amazon EC2의 업데이트 관리

EC2 인스턴스에서 운영 체제와 애플리케이션을 정기적으로 패치, 업데이트 및 보호하는 것이 좋습니다. AWS 시스템 관리자 Patch Manager를 사용하여 운영 체제와 애플리케이션 모두에 대해 보안 관련 업데이트를 설치하는 프로세스를 자동화할 수 있습니다. 또는 자동 업데이트 서비스를 사용하거나, 애플리케이션 공급업체에서 제공하는 업데이트를 설치하기 위한 권장 프로세스를 사용할 수 있습니다.

Amazon EC2에 대한 규정 준수 확인

타사 감사자는 여러 AWS 규정 준수 프로그램의 일환으로 Amazon EC2의 보안 및 규정 준수를 평가합니다. 여기에는 SOC, PCI, FedRAMP, HIPAA 등이 포함됩니다.

특정 규정 준수 프로그램의 범위 내에 있는 AWS 서비스 목록은 규정 준수 프로그램 제공 범위 내 AWS 서비스를 참조하십시오. 일반적인 내용은 AWS 규정 준수 프로그램을 참조하십시오.

AWS Artifact를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다. 자세한 내용은 AWS Artifact에서 보고서 다운로드를 참조하십시오.

Amazon EC2 사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표, 적용되는 법률과 규정에 따라 결정됩니다. AWS에서는 규정 준수를 지원하기 위해 다음과 같은 리소스를 제공합니다.

- 보안 및 규정 준수 빠른 시작 안내서 – 이 배포 가이드에서는 아키텍처 고려 사항에 대해 설명하고 보안 및 규정 준수에 중점을 둔 기본 AWS 환경을 배포하기 위한 단계를 제공합니다.
- HIPAA 보안 및 규정 준수 기술 백서 설계 – 이 백서는 기업에서 AWS를 사용하여 HIPAA를 준수하는 애플리케이션을 만드는 방법을 설명합니다.
- AWS 규정 준수 리소스 – 이 워크북 및 안내서 모음은 귀사가 속한 업계 및 위치에 적용될 수 있습니다.
- AWS Config Developer Guide의 규칙을 사용하여 리소스 평가 – AWS Config는 리소스 구성이 내부 사례, 업계 지침, 규정을 얼마나 잘 준수하는지 평가합니다.
- AWS Security Hub – 이 AWS 제품으로 보안 업계 표준 및 모범 사례 규정 준수 여부를 확인하는 데 도움이 되는 AWS 내 보안 상태에 대한 포괄적인 관점을 제공합니다.

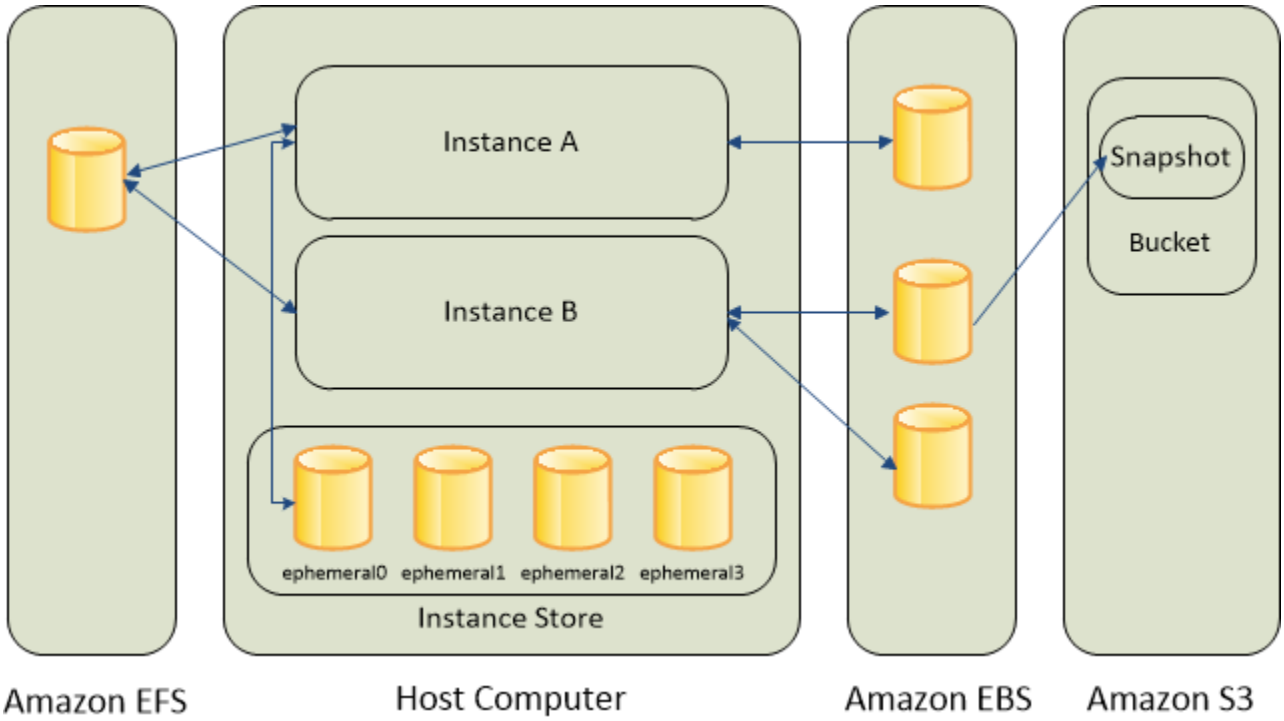
스토리지

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 (p. 888)
- Amazon EC2 인스턴스 스토어 (p. 1027)
- Amazon Elastic File System (Amazon EFS) (p. 1040)
- Amazon Simple Storage Service (Amazon S3) (p. 1044)

다음 그림은 이러한 스토리지 옵션과 인스턴스 간의 관계를 보여줍니다.



Amazon EBS

Amazon EBS는 내구성이 있는 블록 수준 스토리지 볼륨을 제공하여 실행 중인 인스턴스에 연결하는 것이 가능합니다. 세분화된 업데이트를 자주 수행하는 데이터의 경우 기본 스토리지 디바이스로 Amazon EBS를 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 암

호화 기능을 사용하여 EBS 볼륨을 암호화된 볼륨으로 생성할 수도 있습니다. 자세한 내용은 Amazon EBS Encryption (p. 968) 단원을 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 (p. 888) 단원을 참조하십시오.

Amazon EC2 인스턴스 스토어

여러 인스턴스는 호스트 컴퓨터에 물리적으로 연결된 디스크의 스토리지에 액세스할 수 있습니다. 이러한 디스크 스토리지를 인스턴스 스토어라고 합니다. 인스턴스 스토어는 인스턴스에 블록 수준의 임시 스토리지를 제공합니다. 인스턴스 스토어에 저장된 데이터는 연관 인스턴스의 수명 기간 동안에만 유지되고, 해당 인스턴스를 중지하거나 종료하면 인스턴스 스토어 볼륨의 데이터가 손실됩니다. 자세한 내용은 Amazon EC2 인스턴스 스토어 (p. 1027) 단원을 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 Elastic File System (Amazon EFS) (p. 1040) 단원을 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 쉽게수행할 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. 1044) 단원을 참조하십시오.

스토리지 추가

AMI에서 인스턴스를 실행할 때마다 해당 인스턴스에 대한 루트 스토리지 디바이스가 생성됩니다. 루트 스토리지 디바이스에는 인스턴스를 부팅하기 위해 필요한 모든 정보가 포함됩니다. 블록 디바이스 매핑을 사용하면 AMI를 생성하거나 인스턴스를 실행할 때 루트 디바이스 볼륨과 스토리지 볼륨을 지정할 수 있습니다. 자세한 내용은 블록 디바이스 매핑 (p. 1049) 단원을 참조하십시오.

실행 중인 인스턴스에 EBS 볼륨을 연결할 수도 있습니다. 자세한 내용은 Amazon EBS 볼륨을 인스턴스에 연결 (p. 910) 단원을 참조하십시오.

스토리지 요금

스토리지 요금에 대한 자세한 내용을 보려면 AWS 요금을 열고 서비스 요금 아래로 스크롤하여 스토리지를 선택한 다음 스토리지 옵션을 선택하여 해당 스토리지 옵션의 요금 페이지를 여십시오. 스토리지 비용 추정에 대한 자세한 내용은 AWS Pricing Calculator를 참조하십시오.

Amazon Elastic Block Store (Amazon EBS)

Amazon Elastic Block Store(Amazon EBS)는 EC2 인스턴스에 사용할 수 있는 블록 수준 스토리지 볼륨을 제공합니다. EBS 볼륨은 형식이 지정되지 않은 원시 블록 디바이스처럼 동작합니다. 이러한 볼륨을 인스턴스에 디바이스로 마운트할 수 있습니다. 동일한 인스턴스에 여러 볼륨을 탑재하고 한 번에 여러 인스턴스에 볼륨을 탑재할 수 있습니다. 이러한 볼륨 위에 파일 시스템을 생성하거나 하드 드라이브와 같은 블록 디바이스를 사용하는 것처럼 볼륨을 사용할 수 있습니다. 인스턴스에 연결된 볼륨의 구성을 동적으로 변경할 수 있습니다.

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 Elastic Block Store 요금에 대한 자세한 정보는 Amazon EBS 페이지의 비용 예측 단원을 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 볼륨 한도 증가 요청을 참조하십시오.

데이터에 빠르게 액세스하고 장기적으로 지속해야 하는 경우 Amazon EBS를 사용하는 것이 좋습니다. EBS 볼륨은 세분화된 업데이트가 필요하고 형식이 지정되지 않은 블록 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.

목차

- Amazon EBS의 기능 (p. 889)
- Amazon EBS 볼륨 (p. 890)
- Amazon EBS 스냅샷 (p. 926)
- Amazon EBS 데이터 서비스 (p. 957)
- Linux 인스턴스의 Amazon EBS 및 NVMe (p. 981)
- Amazon EBS 최적화 인스턴스 (p. 984)
- Amazon EBS 인스턴스의 Linux 볼륨 성능 (p. 997)
- Amazon EBS의 Amazon CloudWatch 지표 (p. 1011)
- Amazon EBS용 Amazon CloudWatch Events (p. 1017)

Amazon EBS의 기능

- 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 지역 어디서나 새 볼륨으로 복원할 수 있습니다. 스냅샷을 다른 지역에 복사한 다음 새 볼륨에 복원하면 지리적 확장, 데이터 센터 마이그레이션 및 재해 복구를 위해 여러 AWS 지역을 쉽게 활용할 수 있습니다.
- Amazon EBS는 범용 SSD(`gp2`), 프로비저닝된 IOPS SSD(`io1`), 처리량에 최적화된 HDD(`st1`), Cold HDD(`sc1`) 볼륨 유형을 제공합니다. 다음은 각 볼륨 유형에 대한 성능 및 사용 사례의 요약입니다.
  - 범용 SSD 볼륨에서는 3 IOPS/GiB를 기본 성능으로 제공하며, 시간을 연장할 경우 최대 3,000 IOPS까지 버스트할 수 있습니다. 이러한 볼륨은 부트 볼륨, 중소 규모 데이터베이스, 개발 및 테스트 환경 등의 광범위한 사용 사례에 적합합니다. 자세한 내용은 범용 SSD(`gp2`) 볼륨 (p. 894) 단원을 참조하십시오.
  - 프로비저닝된 IOPS SSD 볼륨은 최대 64,000 IOPS 및 1,000MiB/s의 처리량을 지원합니다. 따라서 예측 가능한 방식으로 EC2 인스턴스당 수만 IOPS까지 확장할 수 있습니다. 자세한 정보는 프로비저닝된 IOPS SSD(`io1`) 볼륨 (p. 897) 단원을 참조하십시오.
  - 처리량에 최적화된 HDD 볼륨은 IOPS가 아닌 처리량으로 성능을 정의하는 저비용 마그네틱 스토리지를 제공합니다. 이러한 볼륨은 Amazon EMR, ETL, 데이터 웨어하우스, 로그 처리 같은 대용량 순차 워크로드에 적합합니다. 자세한 내용은 처리량에 최적화된 HDD(`st1`) 볼륨 (p. 898) 단원을 참조하십시오.
  - Cold HDD 볼륨은 IOPS가 아닌 처리량으로 성능을 정의하는 저비용 마그네틱 스토리지를 제공합니다. 이 볼륨은 순차적인 대용량 콜드 데이터 워크로드에 적합합니다. 데이터에 자주 액세스할 필요가 없고 비용을 절약해야 한다면 저렴한 블록 스토리지로 이러한 볼륨이 적합합니다. 자세한 내용은 Cold HDD(`sc1`) 볼륨 (p. 901) 단원을 참조하십시오.
- 암호화된 볼륨으로 EBS 볼륨을 생성하여 규제/감사 데이터 및 애플리케이션에 대한 다양한 저장 시 데이터 암호화 요구 사항을 충족할 수 있습니다. 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. 968) 단원을 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 지역에서 복사할 수 있습니다. 자세한 내용은 Amazon EBS 스냅샷 (p. 926) 단원을 참조하십시오.

- 대역폭, 처리량, 지연 시간, 평균 대기열 길이 등의 성능 지표가 AWS Management 콘솔을 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 Amazon EBS 인스턴스의 Linux 볼륨 성능 (p. 997) 단원을 참조하십시오.

Amazon EBS 볼륨

Amazon EBS 볼륨은 내구성이 있는 블록 수준 스토리지 디바이스이며 하나 이상의 인스턴스를 연결하는 것이 가능합니다. 인스턴스의 시스템 드라이브 또는 데이터베이스 애플리케이션용 스토리지 등 자주 업데이트해야 하는 데이터의 경우 EBS 볼륨을 기본 스토리지로 사용할 수 있습니다. 연속으로 디스크 스캔을 수행하는 처리량 집약적 애플리케이션에도 해당 볼륨을 사용할 수 있습니다. EBS 볼륨은 EC2 인스턴스의 실행 주기와는 독립적으로 유지됩니다.

여러 EBS 볼륨을 단일 인스턴스에 연결할 수 있습니다. 볼륨 및 인스턴스는 동일 가용 영역에 위치해야 합니다. 볼륨을 인스턴스에 연결하면 다른 물리적 하드 드라이브처럼 사용할 수 있습니다. EBS 볼륨은 유연합니다. 현재 세대 인스턴스 유형에 연결된 현재 세대 볼륨의 경우 크기를 동적으로 늘리고 프로비저닝된 IOPS 용량을 수정하며 라이브 프로덕션 볼륨의 볼륨 유형을 변경할 수 있습니다.

Amazon EBS는 범용 SSD(`gp2`), 프로비저닝된 IOPS SSD (`io1`), 처리량에 최적화된 HDD (`st1`), Cold HDD (`sc1`) 및 Magnetic(`standard`, 이전 세대 유형) 같은 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. 892) 단원을 참조하십시오.

내용

- EBS 볼륨 사용의 이점 (p. 890)
- Amazon EBS 볼륨 유형 (p. 892)
- EBS 볼륨의 크기 및 구성에 대한 제약 (p. 905)
- Amazon EBS 볼륨 생성 (p. 908)
- 스냅샷에서 Amazon EBS 볼륨 복구 (p. 909)
- Amazon EBS 볼륨을 인스턴스에 연결 (p. 910)
- Amazon EBS 다중 연결을 사용하여 여러 인스턴스에 볼륨 연결 (p. 911)
- Amazon EBS 볼륨을 Linux에서 사용할 수 있도록 만들기 (p. 914)
- Amazon EBS 볼륨에 대한 정보 보기 (p. 917)
- 볼륨 상태 모니터링 (p. 918)
- 인스턴스에서 Amazon EBS 볼륨 분리 (p. 924)
- Amazon EBS 볼륨 삭제 (p. 926)

EBS 볼륨 사용의 이점

EBS 볼륨은 인스턴스 스토어 볼륨과 차별화된 이점을 제공합니다.

데이터 가용성

EBS 볼륨을 생성하면 단일 하드웨어 구성 요소의 장애로 인한 데이터 손실을 방지하기 위해 해당 가용 영역 내에서 자동으로 복제됩니다. 동일한 가용 영역에 있는 EC2 인스턴스에 EBS 볼륨을 연결할 수 있습니다. 볼륨을 연결한 후에 인스턴스는 하드 드라이브 또는 기타 물리 드라이브와 같은 원시 블록 디바이스처럼 보입니다. 이 시점에 인스턴스는 로컬 드라이브와 동일한 방식으로 볼륨과 상호 작용할 수 있습니다. 인스턴스에 연결하고 ext3 등의 파일 시스템으로 EBS 볼륨을 포맷한 다음 애플리케이션을 설치할 수 있습니다.

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

io1 EBS 볼륨을 최대 16개의 Nitro 기반 인스턴스에 연결할 수 있습니다. 자세한 내용은 Amazon EBS 다중 연결을 사용하여 여러 인스턴스에 볼륨 연결 (p. 911) 단원을 참조하십시오. 아니면 EBS 볼륨을 단일 인스턴스에 연결할 수 있습니다.

추가 비용 없이 EBS 볼륨 및 EBS 기반 인스턴스의 루트 디바이스 볼륨의 데이터를 모니터링할 수 있습니다. 측정치의 모니터링에 대한 자세한 내용은 Amazon EBS의 Amazon CloudWatch 지표 (p. 1011) 단원을 참조하십시오. 볼륨 상태 추적에 대한 자세한 내용은 Amazon EBS용 Amazon CloudWatch Events (p. 1017) 단원을 참조하십시오.

데이터 지속성

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

실행 중인 인스턴스에 연결된 EBS 볼륨은 사용자가 EC2 콘솔에서 해당 인스턴스에 대한 EBS 볼륨을 구성할 때 종료 시 삭제 확인란을 선택하지 않은 경우 인스턴스가 종료될 때 해당 데이터가 원래 상태를 유지한 채로 해당 인스턴스에서 자동으로 분리될 수 있습니다. 그러면 해당 볼륨은 새 인스턴스로 재연결되어 빠른 복구가 가능합니다. 종료 시 삭제 확인란이 선택된 경우에는 EC2 인스턴스 종료 시 볼륨이 삭제됩니다. EBS 기반 인스턴스를 사용하는 경우 연결된 볼륨에 저장된 데이터에 영향을 주지 않고 해당 인스턴스를 중지하고 다시 시작할 수 있습니다. 해당 볼륨은 정지-시작 주기 동안 연결 상태를 유지합니다. 이를 통해 사용자는 필요할 때 처리 및 스토리지 리소스만을 사용하여 볼륨에서 데이터를 무기한으로 처리 및 저장할 수 있습니다. 데이터는 볼륨이 완전히 삭제될 때까지 볼륨에서 유지됩니다. 삭제된 EBS 볼륨에서 사용된 물리 블록 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. 968) 단원을 참조하십시오.

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

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

데이터 암호화

단순 데이터 암호화의 경우 Amazon EBS 암호화 기능으로 암호화된 EBS 볼륨을 생성할 수 있습니다. 모든 EBS 볼륨 유형은 암호화를 지원합니다. 암호화된 EBS 볼륨을 사용하여 규제/감사 데이터 및 애플리케이션에 대한 다양한 유휴 데이터 암호화 요구 사항을 충족할 수 있습니다. Amazon EBS 암호화는 256비트 고급 암호화 표준 알고리즘(AES-256) 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. 968) 단원을 참조하십시오.

Amazon EBS 암호화은 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. Amazon EBS 암호화를 사용하여 별도로 생성된 CMK(고객 마스터 키)를 선택하는 경우를 제외하고 이 키가 AWS KMS에 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 단원을 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 계속에서 작성하면 새 볼륨의 기준으로 사용될 볼륨 스냅샷을 주기적으로 생성할 수 있습니다. 이 스냅샷을 사용하여 새로운 EBS 볼륨을 여러 개 생성하거나 가용 영역 간에 볼륨을 이동할 수 있습니다. 암호화된 EBS 볼륨의 스냅샷은 자동으로 암호화됩니다.

스냅샷에서 새로운 볼륨을 생성하는 경우 새로 생성된 스냅샷이 생성될 시점의 원본 볼륨 사본과 정확히 일치합니다. 암호화된 스냅샷에서 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 퍼블릭으로 설정할 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 EC2리소스에 태그 지정 (p. 1068) 단원을 참조하십시오.

볼륨을 자동으로 백업하려는 경우 Amazon 데이터 수명 주기 관리자 (p. 947) 또는 AWS Backup를 사용할 수 있습니다.

유연성

EBS 볼륨은 프로덕션 중에 라이브 구성 변경을 지원합니다. 서비스 중단 없이 볼륨 유형, 볼륨 크기, IOPS 용량을 수정할 수 있습니다. 자세한 내용은 Amazon EBS 탄력적 볼륨 (p. 958) 단원을 참조하십시오.

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

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 Amazon EBS 인스턴스의 Linux 볼륨 성능 (p. 997)을 참조하십시오.

요금에 대한 자세한 내용은 Amazon EBS 요금을 참조하십시오.

볼륨 특성

다음 표는 각 볼륨 유형에 대한 사용 사례 및 성능 특성을 설명합니다. 기본 볼륨 유형은 범용 SSD(`gp2`)입니다.

	SSD(Solid-State Drive)		하드 디스크 드라이브(HDD)	
볼륨 유형	범용 SSD (`gp2`)	프로비저닝된 IOPS SSD (`io1`)	처리량에 최적화된 HDD (`st1`)	Cold HDD (`sc1`)
설명	다양한 워크로드에 사용할 수 있으며 가격 대비 성능이 우수한 범용 SSD 볼륨	지연 시간이 짧거나 처리량이 많은 미션 크리티컬 워크로드에 적합한 고성능 SSD 볼륨	자주 액세스하는 처리량 집약적 워크로드에 적합한 저비용 HDD 볼륨	자주 액세스하지 않는 워크로드에 적합한 최저 비용 HDD 볼륨
사용 사례	• 대부분의 워크로드에 추천	• 지속적인 IOPS 성능이나 16,000 IOPS 또는	• 저비용으로 일관되고 높은 처	• 자주 액세스하지 않는 대용량

	SSD(Solid-State Drive)		하드 디스크 드라이브(HDD)	
	• 시스템 부트 볼륨 • 가상 데스크톱 • 지연 시간이 짧은 대화형 앱 • 개발 및 테스트 환경	250MiB/s 이상의 볼륨당 처리량을 필요로 하는 중요한 비즈니스 애플리케이션 • 라지 데이터베이스 워크로드. 예: • MongoDB • Cassandra • Microsoft SQL Server • MySQL • PostgreSQL • Oracle	리량을 요구하는 스트리밍 워크로드 • 빅 데이터 • 데이터 웨어하우스 • 로그 처리 • 부트 볼륨이 될 수 없음	데이터를 위한 처리량 중심의 스토리지 • 스토리지 비용이 최대한 낮아야 하는 시나리오 • 부트 볼륨이 될 수 없음
API 이름	`gp2`	`io1`	`st1`	`sc1`
볼륨 크기	1GiB - 16TiB	4GiB - 16TiB	500GiB - 16TiB	500GiB - 16TiB
볼륨당 최대 IOPS	16,000(16KiB I/O)*	64,000(16KiB I/O)†	500(1MiB I/O)	250(1MiB I/O)
볼륨당 최대 처리량	250MiB/s*	1,000MiB/s†	500MiB/s	250MiB/s
인스턴스당 최대 IOPS ††	80,000	80,000	80,000	80,000
인스턴스당 최대 처리량 ††	2,375MB/s	2,375MB/s	2,375MB/s	2,375MB/s
기준 성능 속성	IOPS	IOPS	MiB/s	MiB/s

* 처리량 한도는 볼륨 크기에 따라 128MiB/s~250 MiB/s입니다. 170GiB보다 작은 볼륨은 최대 128MiB/s의 처리량을 제공합니다. 170GiB에서 334GiB까지의 볼륨은 버스트 크레딧을 사용할 수 있는 경우 최대 250MiB/s의 처리량을 제공합니다. 334 GiB 이상의 볼륨은 버스트 크레딧에 상관없이 250MiB/s를 제공합니다. 이전 `gp2` 볼륨은 볼륨을 수정하지 않는 한 전체 성능에 도달할 수 없습니다. 자세한 내용은 Amazon EBS 탄력적 볼륨 (p. 958) 단원을 참조하십시오.

† 최대 IOPS 및 처리량은 32,000 IOPS 이상으로 프로비저닝된 Nitro 기반 인스턴스 (p. 181)에서만 보장됩니다. 다른 인스턴스는 최대 32,000 IOPS 및 500 MiB/s의 성능을 보장합니다. 이전 `io1` 볼륨은 볼륨을 수정하지 않는 한 전체 성능에 도달할 수 없습니다. 자세한 내용은 Amazon EBS 탄력적 볼륨 (p. 958) 단원을 참조하십시오.

†† 이 처리량을 달성하려면 EBS 최적화 (p. 984)를 지원하는 인스턴스가 있어야 합니다.

이전 세대 볼륨 유형

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 볼륨을 참조하십시오.

하드 디스크 드라이브(HDD)	
볼륨 유형	Magnetic
사용 사례	데이터에 자주 액세스하지 않는 워크로드

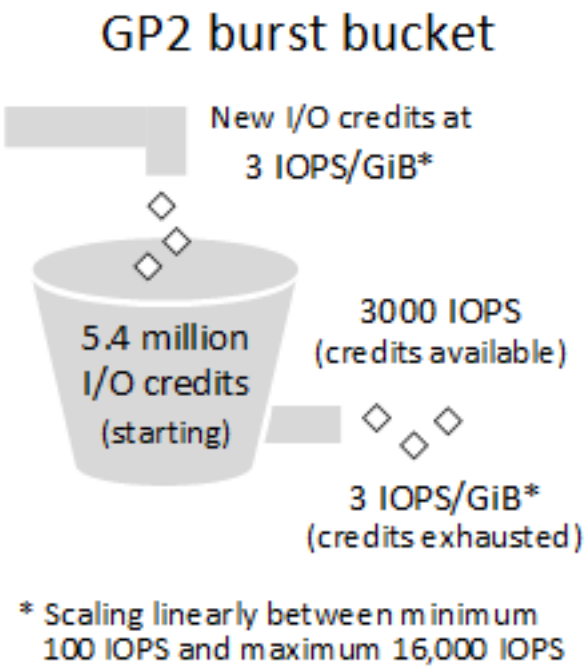
하드 디스크 드라이브(HDD)	
API 이름	`standard`
볼륨 크기	1GiB - 1TiB
볼륨당 최대 IOPS	40–200
볼륨당 최대 처리량	40-90MiB/s
인스턴스당 최대 IOPS	80,000
인스턴스당 최대 처리량	1,750MiB/s
기준 성능 속성	IOPS

범용 SSD(gp2) 볼륨

범용 SSD(gp2) 볼륨은 광범위한 작업에서 이상적으로 사용될 수 있는 비용 효과적인 스토리지를 제공합니다. 이러한 볼륨은 시간을 연장할 경우 3,000IOPS의 버스트 기능까지 지원되어 지연 시간이 한 자릿수 밀리초에 불과합니다. 최소 100 IOPS(33.33GiB 이하)와 최대 16,000 IOPS(5,334GiB 이상) 사이에서, 기준 성능은 볼륨 크기의 GiB당 3 IOPS로 일정하게 확장됩니다. AWS는 프로비저닝된 성능의 99%의 시간에 제공하기 위해 gp2 볼륨을 설계합니다. gp2 볼륨 크기는 1GiB~16TiB입니다.

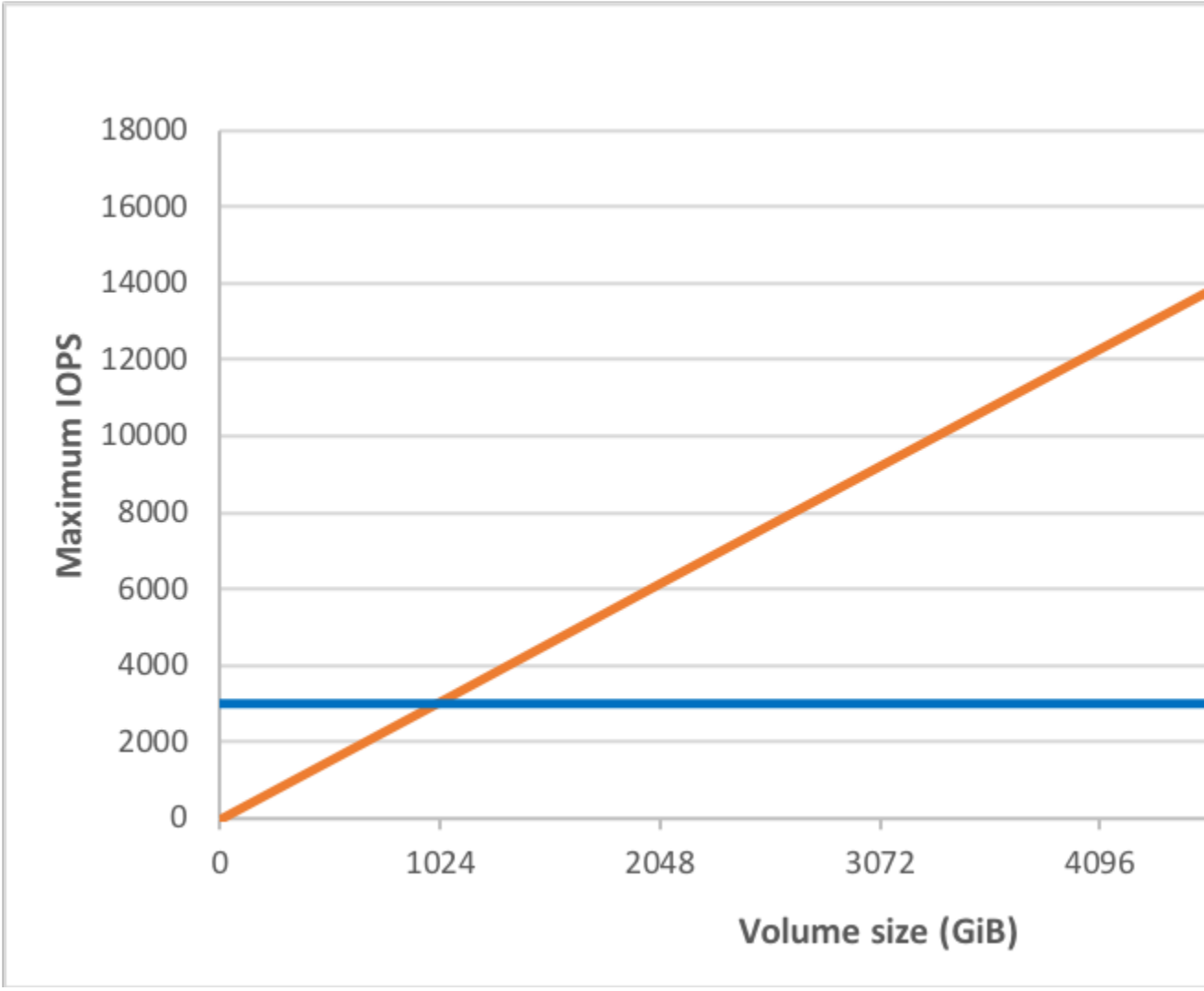
I/O 크레딧 및 버스트 성능

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 밸런스는 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 밸런스에서 I/O 크레딧을 사용하여 최대 3,000 IOPS까지 필요한 성능 수준을 버스트할 수 있습니다. 볼륨이 초당 획득한 I/O 크레딧 이하를 사용하는 경우 미사용 I/O 크레딧은 I/O 크레딧 밸런스에 가산됩니다. 볼륨의 최대 I/O 크레딧은 초기 크레딧 밸런스(540만 I/O 크레딧)와 동일합니다.

볼륨의 기준 성능이 최대 버스트 성능보다 높으면 I/O 크레딧이 사용되지 않습니다. 볼륨이 Nitro 기반 인스턴스 (p. 181)에 연결된 경우 버스트 잔고가 보고되지 않습니다. 비 Nitro 기반 인스턴스의 경우 보고된 버스트 잔고는 100%입니다.

볼륨의 버스트 구간은 볼륨의 크기, 필요한 버스트 IOPS 및 버스트가 시작되는 크레딧 밸런스에 의해 결정됩니다. 방법은 다음 수식과 같습니다.

```
                              (Credit balance)
Burst duration  =  ------------------------------------
                   (Burst IOPS) - 3(Volume size in GiB)
```

다음 표는 여러 볼륨 크기 및 관련 볼륨 기준 성능(I/O 크레딧 누적 비율), 최대 3,000 IOPS에서의 버스트 기간(전체 크레딧 밸런스에서 시작 시) 및 볼륨이 빈 크레딧 밸런스를 다시 채우는 데 걸리는 초 단위 시간을 보여줍니다.

볼륨 크기(GiB)	기준 성능(IOPS)	지속적으로 3,000 IOPS(초) 구동 시 버스트 지속 시간	IO 구동이 없을 시 빈 크레딧 밸런스를 채우는데 소요되는 시간(초)
1	100	1802	54,000
100	300	2,000	18,000
250	750	2,400	7,200
334(최대 처리량에 대한 최소 크기)	1002	2703	5389
500	1,500	3,600	3,600
750	2,250	7,200	2,400
1,000	3,000	해당 사항 없음*	해당 사항 없음*
5,334(최대 IOPS에 대한 최소 크기)	16,000	해당 사항 없음*	해당 사항 없음*
16,384(16TiB, 최대 볼륨 크기)	16,000	해당 사항 없음*	해당 사항 없음*

* 볼륨의 기준 성능이 최대 버스트 성능을 초과합니다.

I/O 크레딧 밸런스가 0이 되면 어떻게 되나요?

`gp2` 볼륨에서 I/O 크레딧 잔고 전부가 소진되면 볼륨의 최대 IOPS 성능이 기준 IOPS 성능 수준(볼륨이 크레딧을 획득하는 속도)으로 유지되고 볼륨의 최대 처리량은 최대 I/O 크기를 곱한 기준 IOPS로 줄어듭니다. 처리량은 250MiB/s를 초과할 수 업습니다. I/O 요구가 기준 성능 수준 이하로 떨어지고 미사용 크레딧이 I/O 크레딧 밸런스에 추가되면, 볼륨의 최대 IOPS 성능이 다시 기준 수준을 초과합니다. 예를 들어, 크레딧 잔고가 0인 100GiB `gp2` 볼륨은 300 IOPS의 기준 성능 및 75MiB/s의 처리 한도를 갖습니다(초당 300I/O 작업 * I/O 작업당 256KiB = 75MiB/s). 볼륨이 커지면 기준 성능도 높아지고 크레딧 밸런스가 더 빨리 보충됩니다. IPOS 측정 방법에 대한 자세한 내용은 I/O 특성 및 모니터링 (p. 1000) 단원을 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 16,000 IOPS 이상으로 유지해야 하는 워크로드인 경우 `io1` 볼륨으로 전환하는 것을 고려해야 합니다.

CloudWatch 지표 및 경보를 사용하여 버스트 버킷 잔고를 모니터링하는 방법은 `gp2`, `st1`, `sc1` 볼륨에 대한 버스트 버킷 잔고 모니터링 (p. 905) 단원을 참조하십시오.

처리량 성능

`gp2` 볼륨의 처리량은 다음 공식을 사용하여 계산할 수 있습니다. 최대 처리량 한도는 250MiB/s입니다.

```
Throughput in MiB/s = ((Volume size in GiB) × (IOPS per GiB) × (I/O size in KiB))
```

V = 볼륨 크기, I = I/O 크기, R = I/O 속도, T = 처리량이라고 가정하면 이 공식을 다음과 같이 축약할 수 있습니다.

```
T = VIR
```

최대 처리량을 달성하는 최소 볼륨 크기는 다음과 같이 계산됩니다.

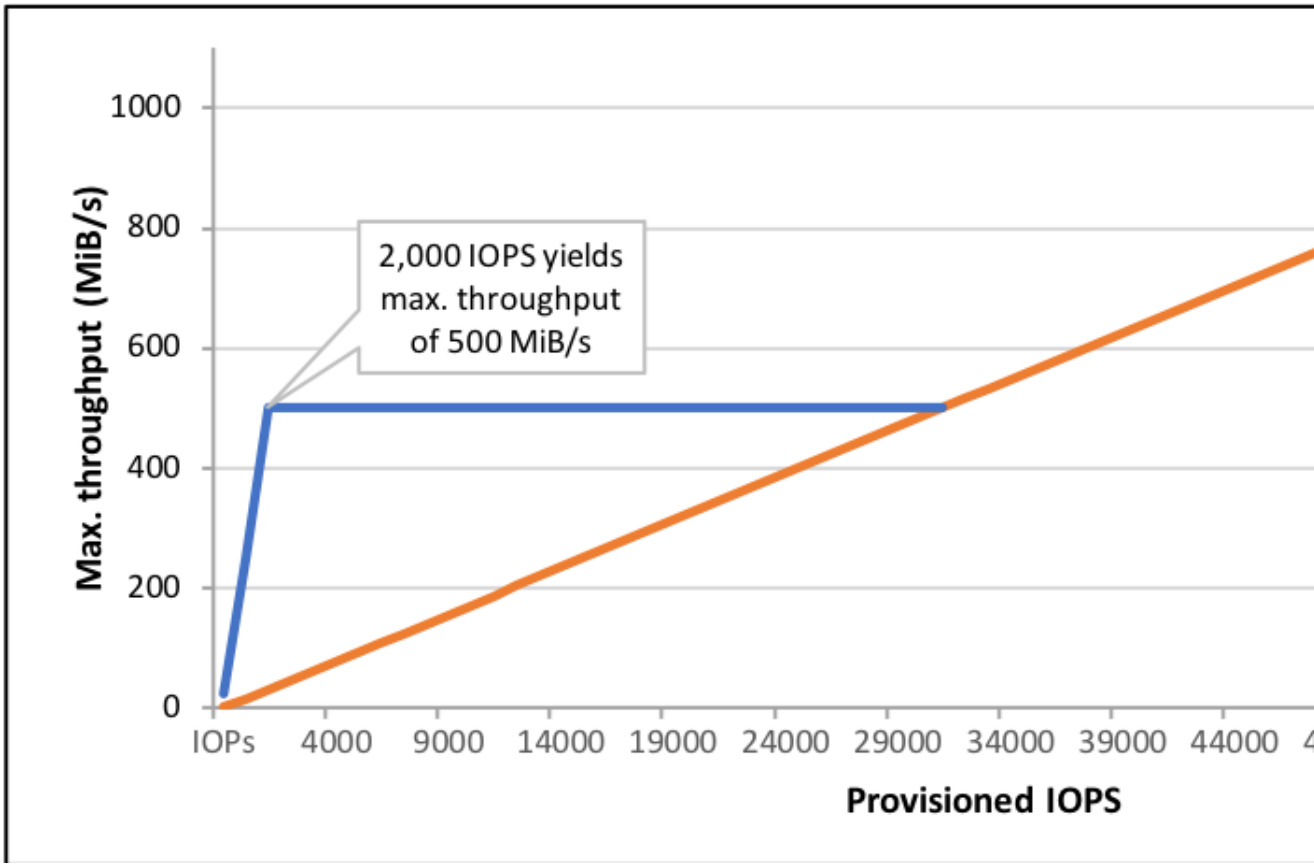
```
        T
V  =  -----
       I R

             250 MiB/s
   =  ---------------------
      (256 KiB)(3 IOPS/GiB)


                [(250)(2^20)(Bytes)]/s
   =  -----------------------------------------
      (256)(2^10)(Bytes)([3 IOP/s]/[(2^30)(Bytes)])


      (250)(2^20)(2^30)(Bytes)
   =  ------------------------
           (256)(2^10)(3)

   =  357,913,941,333 Bytes

   =  333# GiB (334 GiB in practice because volumes are provisioned in whole gibibytes)
```

프로비저닝된 IOPS SSD(io1) 볼륨

프로비저닝된 IOPS SSD (io1) 볼륨은 스토리지 성능과 일관성에 민감한 I/O 집약적 워크로드, 특히 데이터베이스 워크로드 요구 사항을 충족하도록 설계되었습니다. 버킷과 크레딧 모델을 사용해 성능을 계산하는 gp2와 달리 io1 볼륨을 사용하면 볼륨을 생성할 때 일정한 IOPS 속도를 지정할 수 있으며, Amazon EBS는 해당 기간 중 프로비저닝된 성능의 99.9%까지 전송합니다.

io1 볼륨 크기는 4GiB~16TiB입니다. 100 IOPS부터 Nitro 기반 인스턴스 (p. 181) 인스턴스에서는 볼륨당 최대 64,000 IOPS, 그리고 다른 인스턴스에서는 최대 32,000까지 프로비저닝할 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. 지원되는 인스턴스 유형에서, 크기가 1,280GiB 이상인 볼륨은 최대 64,000 IOPS(50 × 1,280GiB = 64,000)까지 프로비저닝할 수 있습니다.

최대 32,000 IOPS로 프로비저닝된 io1 볼륨은 최대 256KiB의 I/O 크기를 지원하고 최대 500MiB/s의 처리량을 제공합니다. I/O 크기가 최대일 때 2,000 IOPS에서 피크 처리량에 도달합니다. 32,000 IOPS 이상(상한 64,000 IOPS)으로 프로비저닝된 볼륨은 최대 16KiB의 I/O 크기를 지원하고 최대 1,000MiB/s의 처리량을 제공합니다. 다음 그래프에 이러한 성능 특성이 예시되어 있습니다.



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

> **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.

처리량에 최적화된 HDD(`st1`) 볼륨

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

처리량에 최적화된 HDD(`st1`) 볼륨은 Cold HDD(`sc1`) 볼륨과 비슷하지만 자주 액세스하는 데이터를 지원하도록 설계되었습니다.

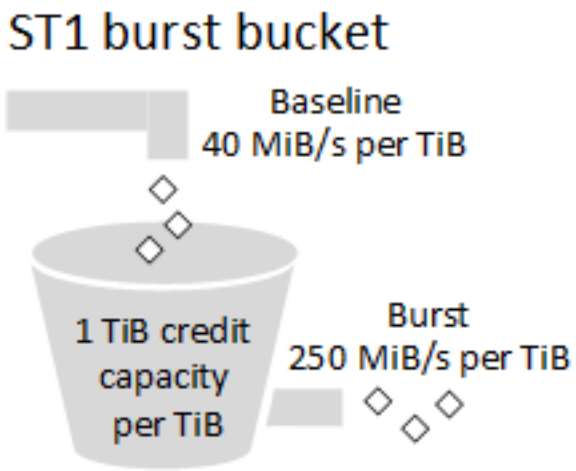
이 볼륨 유형은 대용량 순차 I/O와 관련된 워크로드에 최적화되어 있으며, 소량의 랜덤 I/O 워크로드를 처리하는 고객에게는 `gp2` 사용을 권장합니다. 자세한 내용은 HDD 기반 소량 읽기/쓰기의 비효율성 (p. 905) 단원을 참조하십시오.

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 크레

딧을 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.

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

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

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

0.5TiB~16TiB 범위의 볼륨 크기를 기준으로 기준 처리량은 20MiB/s~500MiB/s(한도)이며, 다음과 같이 12.5TiB에서 한도에 도달합니다.

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

버스트 처리량은 125MiB/s~500MiB/s(한도)이며, 다음과 같이 2TiB에서 한도에 도달합니다.

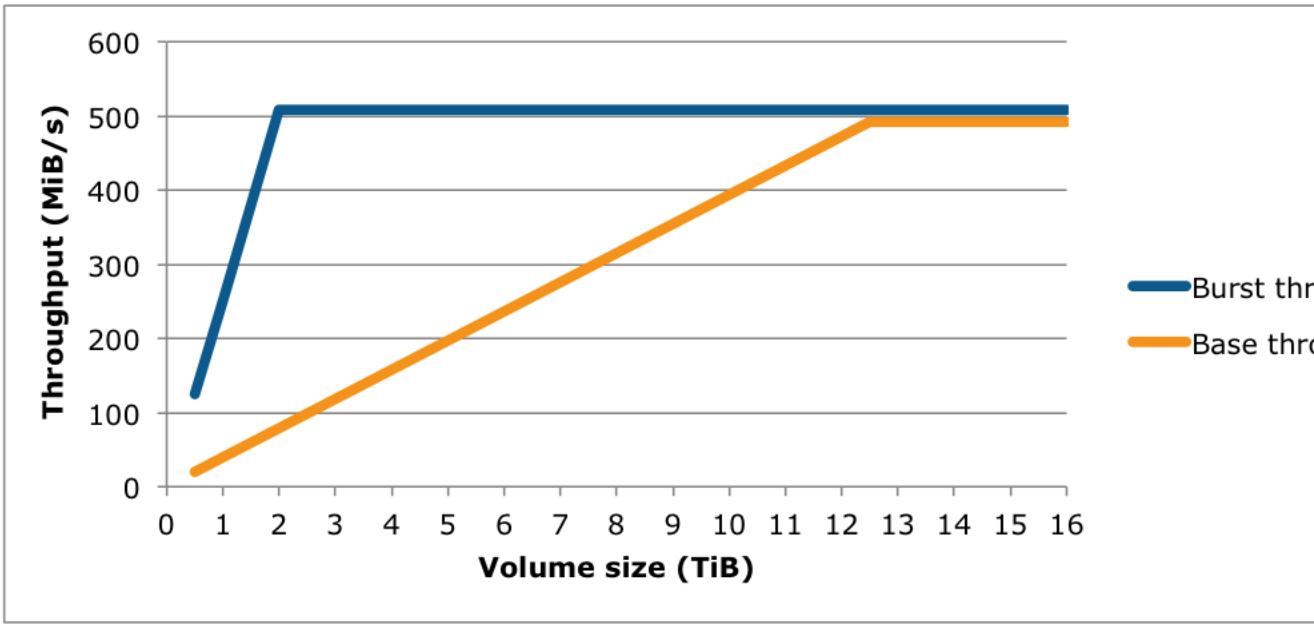
```
         250 MiB/s
2 TiB x ---------- = 500 MiB/s
          1 TiB
```

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
2	80	500
3	120	500

볼륨 크기(TiB)	ST1 기준 처리량(MiB/s)	ST1 버스트 처리량(MiB/s)
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 지표 및 경보를 사용하여 버스트 버킷 잔고를 모니터링하는 방법은 `gp2`, `st1`, `sc1` 볼륨에 대한 버스트 버킷 잔고 모니터링 (p. 905) 단원을 참조하십시오.

Cold HDD(sc1) 볼륨

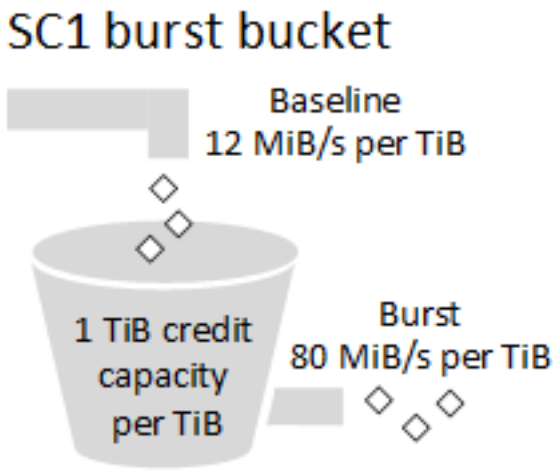
Cold HDD(sc1) 볼륨은 IOPS가 아닌 처리량으로 성능을 정의하는 저비용 마그네틱 스토리지를 제공합니다. 처리량 제한이 st1보다 낮은 sc1은 대용량 순차 콜드 데이터 워크로드에 적합합니다. 데이터에 자주 액세스할 필요가 없고 비용을 절약해야 한다면 저렴한 블록 스토리지로 sc1이 적합합니다. 부팅 가능한 sc1 볼륨은 지원되지 않습니다.

Cold HDD(sc1) 볼륨은 처리량에 최적화된 HDD(st1) 볼륨과 비슷하지만 자주 액세스하지 않는 데이터를 지원하도록 설계되었습니다.

> **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. 905) 단원을 참조하십시오.

처리량 크레딧 및 버스트 성능

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



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

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

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

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

0.5TiB~16TiB 범위의 볼륨 크기를 기준으로 기준 처리량은 6MiB/s~192MiB/s(최대)이며, 다음과 같이 16TiB에서 한도에 도달합니다.

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

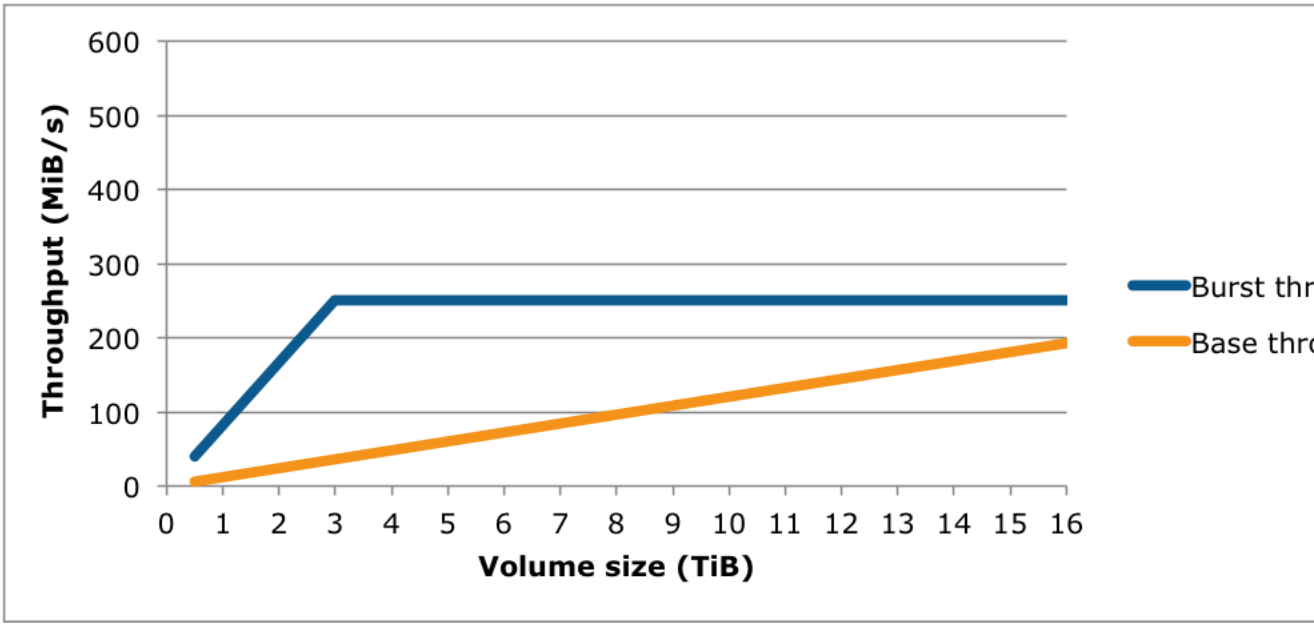
버스트 처리량은 40MiB/s~250MiB/s(한도)이며, 다음과 같이 3.125TiB에서 한도에 도달합니다.

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

다음 표는 `sc1`의 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
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 지표 및 경보를 사용하여 버스트 버킷 잔고를 모니터링하는 방법은 `gp2`, `st1`, `sc1` 볼륨에 대한 버스트 버킷 잔고 모니터링 (p. 905) 단원을 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 볼륨 유형입니다. 새로운 애플리케이션에는 새로운 볼륨 유형 중에서 선택해서 사용하는 것이 좋습니다. 자세한 내용은 이전 세대 볼륨을 참조하십시오.

CloudWatch 지표 및 경보를 사용하여 버스트 버킷 잔고를 모니터링하는 방법은 `gp2`, `st1`, `sc1` 볼륨에 대한 버스트 버킷 잔고 모니터링 (p. 905) 단원을 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.

```
   5 TiB            5 TiB
----------- = ------------------- = 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.

```
      5 TiB
------------------- = 20972 s = 5.83 hours (optimal)
 0.000238418 TiB/s


  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

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

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

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

HDD 기반 소량 읽기/쓰기의 비효율성

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

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

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 (p. 984) 단원을 참조하십시오.

`gp2`, `st1`, `sc1` 볼륨에 대한 버스트 버킷 잔고 모니터링

`gp2`, `st1`, `sc1` 볼륨에 대해 Amazon CloudWatch에서 제공하는 EBS `BurstBalance` 지표를 사용하여 버스트 버킷 수준을 모니터링할 수 있습니다. 이 측정치는 버스트 버킷에 남아 있는 I/O 크레딧(`gp2`의 경우)의 비율 또는 처리량 크레딧의 비율(`st1` 및 `sc1`의 경우)을 보여줍니다. `BurstBalance` 지표 및 I/O와 관련된 기타 지표에 대한 자세한 내용은 I/O 특성 및 모니터링 (p. 1000) 단원을 참조하십시오. CloudWatch에서는 `BurstBalance` 값이 특정 수준 밑으로 떨어질 경우 이를 알리도록 경보를 설정할 수도 있습니다. 자세한 내용은 Amazon CloudWatch 경보 생성을 참조하십시오.

EBS 볼륨의 크기 및 구성에 대한 제약

Amazon EBS 볼륨의 크기는 블록 데이터 스토리지의 물리 및 산술뿐 아니라 운영 체제(OS) 및 파일 시스템 디자이너의 구현 결정에 따라서도 제약을 받습니다. AWS는 서비스의 안정성을 보호하기 위해 볼륨 크기에 추가 제한을 부과합니다.

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

목차

- 스토리지 용량 (p. 906)
- 서비스 제한 (p. 906)
- 파티셔닝 체계 (p. 906)

- 데이터 블록 크기 (p. 907)

스토리지 용량

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

파티셔닝 체계	최대 주소 지정 가능한 블록	이론적 최대 크기(블록 x 블록 크기)	Ext4에서 구현되는 최대 크기*	XFS에서 구현되는 최대 크기**	NTFS에서 구현되는 최대 크기	EBS에서 지원되는 최대 크기
MBR	2^{32}	2TiB	2TiB	2TiB	2TiB	2TiB
GPT	2^{64}	64ZiB	1EiB = 1024^2TiB (RHEL7에서 인증된 50TiB)	500TiB (RHEL7에서 인증됨)	256TiB	16TiB

* https://ext4.wiki.kernel.org/index.php/Ext4_Howto 및 https://access.redhat.com/solutions/1532

** https://access.redhat.com/solutions/1532

서비스 제한

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

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

- EBS는 현재 16TiB의 최대 볼륨 크기를 지원합니다. 즉, EBS 볼륨의 크기를 16TiB까지 만들 수 있지만, OS가 해당 용량을 모두 인식하는지 여부는 자체적인 설계 특성 및 볼륨 파티셔닝 방법에 따라 결정됩니다.
- Linux 부트 볼륨은 MBR 또는 GPT 파티셔닝 체계를 사용할 수 있습니다. MBR은 최대 2047GiB(2TiB - 1GiB)의 부트 볼륨을 지원합니다. GRUB2의 GPT는 2TiB 이상의 부트 볼륨을 지원합니다. Linux AMI가 MBR를 사용하는 경우 부트 볼륨은 2047GiB으로 제한되지만, 비부트 볼륨은 이러한 제한이 없습니다. 자세한 내용은 Amazon EBS 볼륨을 Linux에서 사용할 수 있도록 만들기 (p. 914) 단원을 참조하십시오.

파티셔닝 체계

다른 영향 중에서도 특히, 파티셔닝 체계는 단일 볼륨에서 여러 논리적 데이터 블록을 고유하게 주소 지정할 수 있는 방법을 결정합니다. 자세한 내용은 데이터 블록 크기 (p. 907) 단원을 참조하십시오. 사용 중인 일반적인 파티셔닝 체계는 마스터 부트 레코드(MBR) 및 GUID 파티션 테이블(GPT)입니다. 이러한 체계 간의 중요한 차이점은 다음과 같이 요약할 수 있습니다.

MBR

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

```
(2^32 - 1) × Block size = Number of addressable blocks
```

MBR 볼륨의 블록 크기는 관례적으로 512바이트로 제한됩니다. 따라서:

```
(2^32 - 1) × 512 bytes = 2 TiB - 512 bytes
```

MBR 볼륨에 대한 이 2TiB 제한을 증가시키기 위한 엔지니어링 해결 방법은 업계에서 광범위하게 채택되는 방식과 일치하지 않습니다. 따라서 AWS에서 크기가 더 크게 표시되더라도 Linux와 Windows에서는 MBR 볼륨이 2TiB보다 더 큰 크기로 인식되지 않습니다.

GPT

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

```
(2^64 - 1) × Block size = Number of addressable blocks
```

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

```
(2^64 - 1) × 4,096 bytes
   = 2^64 x 4,096 bytes - 1 x 4,096 bytes
   = 2^64 x 2^12 bytes - 4,096 bytes
   = 2^70 x 2^6 bytes - 4,096 bytes
   = 64 ZiB - 4,096 bytes
```

실제 컴퓨터 시스템은 이러한 이론적 최대 크기와 비슷한 크기를 지원하지 않습니다. 구현되는 파일 시스템 크기는 현재 ext4의 경우 50TiB 및 NTFS의 경우 256TiB로 제한되며, 이 두 가지 크기는 모두 AWS에서 부과하는 16TiB 제한을 초과합니다.

데이터 블록 크기

최신 하드 드라이브의 데이터 스토리지는 논리적 블록 주소 지정을 통해 관리됩니다. 논리적 블록 주소 지정은 운영 체제가 기본 하드웨어에 대한 많은 지식 없이 논리적 블록에서 데이터를 읽고 쓸 수 있도록 하는 추상적 계층입니다. OS는 스토리지 디바이스를 이용하여 블록을 물리적 섹터에 매핑합니다. EBS는 운영 체제에 512바이트 섹터를 공급하며, 운영 체제는 섹터 크기보다 몇 배 더 큰 데이터 블록을 사용하여 디스크에서 데이터를 읽고 씁니다.

논리 데이터 블록의 업계 기본 크기는 현재 4,096바이트(4KiB)입니다. 특정 워크로드는 더 작거나 더 큰 블록 크기에서 이점을 얻을 수 있기 때문에 파일 시스템은 포맷 중 지정할 수 없는 비 기본 블록 크기를 지원합니다. 비 기본 블록 크기를 사용해야 하는 시나리오는 이 주제의 범위를 벗어나지만, 블록 크기 선택은 볼륨의 스토리지 용량에 영향을 미칩니다. 다음 표에는 스토리지 용량이 블록 크기의 함수로 표시됩니다.

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

현재 EBS에서 부과하는 볼륨 크기(16TiB)에 대한 제한은 4KiB 데이터 블록에서 지원되는 최대 크기와 같습니다.

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 지원되는 인스턴스 유형에만 연결될 수 있습니다. 자세한 내용은 지원되는 인스턴스 유형 (p. 969) 단원을 참조하십시오.

고성능 스토리지 시나리오용으로 볼륨을 생성하는 경우 프로비저닝된 IOPS SSD(`io1`) 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 EBS 최적화 인스턴스 (p. 984) 단원을 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. 1001) 단원을 참조하십시오.

볼륨 생성 방법

- EBS 볼륨을 생성하여 실행 인스턴스에 연결할 수 있습니다. 자세한 내용은 아래 절차를 참조하십시오.
- 인스턴스 시작 시 블록 디바이스 매핑을 지정하여 EBS 볼륨을 생성하고 연결할 수 있습니다. 자세한 내용은 인스턴스 시작 마법사를 사용하여 인스턴스 시작 (p. 431) 및 블록 디바이스 매핑 (p. 1049) 단원을 참조하십시오.
- 이전에 생성된 스냅샷에서 볼륨을 복구할 수 있습니다. 자세한 내용은 스냅샷에서 Amazon EBS 볼륨 복구 (p. 909) 단원을 참조하십시오.

콘솔을 이용하여 새 (비어 있는) 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 정보는 리소스 위치 (p. 1059) 단원을 참조하십시오.
3. 탐색 창에서 ELASTIC BLOCK STORE, 볼륨을 선택합니다.
4. 볼륨 생성을 선택합니다.
5. 볼륨 유형에서 볼륨 유형을 선택합니다. 자세한 내용은 Amazon EBS 볼륨 유형 (p. 892) 단원을 참조하십시오.
6. 크기(GiB)에서 볼륨의 크기를 입력합니다. 자세한 내용은 EBS 볼륨의 크기 및 구성에 대한 제약 (p. 905) 단원을 참조하십시오.
7. 프로비저닝된 IOPS SSD 볼륨의 경우 IOPS에서 볼륨이 지원해야 하는 최대 IOPS(초당 입/출력 작업) 수를 입력합니다.
8. 가용 영역에서 볼륨을 생성할 가용 영역을 선택합니다. EBS 볼륨은 동일한 가용 영역 내의 EC2 인스턴스에만 연결될 수 있습니다.
9. (선택 사항) 인스턴스 유형이 EBS 암호화를 지원하고 사용자가 볼륨을 암호화하려는 경우 Encrypt this volume(이 볼륨 암호화)를 선택하고 CMK를 선택합니다. 기본적으로 암호화가 이 리전에서 활성화되는 경우 EBS 암호화가 활성화되고 EBS 암호화에 대한 기본 CMK가 선택됩니다. 마스터 키에서 다른 CMK를 선택하거나 액세스할 수 있는 키의 전체 ARN을 붙여넣을 수 있습니다. 자세한 내용은 Amazon EBS Encryption (p. 968) 단원을 참조하십시오.
10. (선택 사항) 추가 태그 생성(Create additional tags)을 선택하여 볼륨에 태그를 추가합니다. 각 태그에 대해 태그 키와 태그 값을 제공합니다. 자세한 내용은 Amazon EC2리소스에 태그 지정 (p. 1068) 단원을 참조하십시오.
11. 볼륨 생성을 선택합니다. 볼륨 상태가 사용 가능인 경우 볼륨을 인스턴스에 연결할 수 있습니다. 자세한 내용은 Amazon EBS 볼륨을 인스턴스에 연결 (p. 910) 단원을 참조하십시오.

명령줄을 이용하여 새 (비어 있는) EBS 볼륨을 생성하려면,

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- create-volumeAWS CLI
- New-EC2VolumeWindows PowerShell용 AWS 도구

스냅샷에서 Amazon EBS 볼륨 복구

Amazon EBS에 저장된 스냅샷의 데이터로 Amazon S3 볼륨을 복구할 수 있습니다. 스냅샷의 ID를 알고 있어야 하며 스냅샷에 대한 액세스 권한이 있어야 합니다. 스냅샷 복사에 대한 자세한 내용은 Amazon EBS 스냅샷 (p. 926) 단원을 참조하십시오.

EBS 스냅샷은 속도, 편리성 및 비용으로 인해 Amazon EC2에서 선호하는 백업 도구입니다. 스냅샷에서 볼륨을 복원할 경우 모든 데이터를 그대로 유지한 상태로 과거의 특정 시점에서 해당 상태를 재생성합니다. 복원된 볼륨을 인스턴스에 연결하면 여러 리전에서 데이터를 복제하고 테스트 환경을 생성하며 손상된 프로덕션 볼륨 전체를 바꾸거나 특정 파일 및 디렉터리를 검색하여 연결된 다른 볼륨으로 전송할 수 있습니다. 자세한 내용은 Amazon EBS 스냅샷 (p. 926) 단원을 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 볼륨 데이터의 나머지 로드를 진행합니다.

EBS 성능

새 EBS 볼륨은 사용 가능하지만 초기화(이전에는 사전 워밍이라고 함)가 필요하지 않은 시점에 최고 성능을 발휘합니다.

스냅샷에서 복원된 볼륨의 경우, 스토리지 블록에 액세스하려면 먼저 스토리지 블록을 Amazon S3에서 풀다운하고 볼륨에 기록해야 합니다. 이 예비 작업에는 시간이 걸리며, 이로 인해 각 블록에 처음 액세스할 때 I/O 작업의 지연 시간이 상당히 증가할 수 있습니다. 모든 블록을 다운로드하고 볼륨에 기록한 후에 볼륨 성능이 구현됩니다.

대부분의 애플리케이션은 볼륨 수명 주기 동안 초기화 비용을 분할 상환할 수 있습니다. 프로덕션 환경에서 이 초기 성능 저하를 방지하려면 다음 옵션 중 하나를 사용할 수 있습니다.

- 전체 볼륨을 강제로 즉시 초기화합니다. 자세한 내용은 Amazon EBS 볼륨 초기화 (p. 1001) 단원을 참조하십시오.
- 스냅샷에서 빠른 스냅샷 복원을 활성화하여 스냅샷에서 생성된 EBS 볼륨이 생성 시 완전히 초기화되고 모든 프로비저닝된 성능을 즉시 제공하도록 보장합니다. 자세한 내용은 Amazon EBS 빠른 스냅샷 복원 (p. 978) 단원을 참조하십시오.

EBS 암호화

암호화된 스냅샷에서 복구된 새 EBS 볼륨은 자동으로 암호화됩니다. 암호화되지 않은 스냅샷에서 복원하는 동안 즉석에서 볼륨을 암호화할 수도 있습니다. 암호화된 볼륨은 EBS 암호화를 지원하는 인스턴스 유형에만 연결할 수 있습니다. 자세한 내용은 지원되는 인스턴스 유형 (p. 969) 단원을 참조하십시오.

스냅샷에서 볼륨 생성

스냅샷에서 볼륨을 생성하려면 다음 절차에 따르십시오.

콘솔을 사용하여 스냅샷에서 EBS 볼륨을 생성하려면

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

2. 탐색 모음에서 스냅샷이 있는 리전을 선택합니다.

   다른 리전에 있는 볼륨으로 스냅샷으로 복원하려면 스냅샷을 새 리전에 복사한 다음 해당 리전에서 볼륨을 복원할 수 있습니다. 자세한 내용은 Amazon EBS 스냅샷 복사 (p. 934) 단원을 참조하십시오.
3. 탐색 창에서 ELASTIC BLOCK STORE, 볼륨을 선택합니다.
4. 볼륨 생성을 선택합니다.
5. 볼륨 유형에서 볼륨 유형을 선택합니다. 자세한 내용은 Amazon EBS 볼륨 유형 (p. 892) 단원을 참조하십시오.
6. 스냅샷 ID에서 볼륨을 복원할 스냅샷의 ID 또는 설명을 입력한 다음 제안 옵션 목록에서 선택합니다.
7. (선택 사항) 이 볼륨 암호화를 선택하면 볼륨의 암호화 상태를 변경할 수 있습니다. 암호화 기본 제공 (p. 971)이 활성화되어 있는 경우 선택 사항입니다. 마스터 키에서 CMK를 선택하여 EBS 암호화를 위한 기본 CMK 이외의 CMK를 지정합니다.
8. 크기(GiB)에서 볼륨의 크기를 입력하거나 스냅샷의 기본 크기가 적절한지 확인합니다.

   볼륨 크기와 스냅샷을 모두 지정한 경우 크기는 스냅샷 크기보다 크거나 같아야 합니다. 볼륨 유형과 스냅샷을 선택하면 볼륨의 최소 및 최대 크기가 크기 옆에 표시됩니다. 자세한 내용은 EBS 볼륨의 크기 및 구성에 대한 제약 (p. 905) 단원을 참조하십시오.
9. 프로비저닝된 IOPS SSD 볼륨의 경우 IOPS에서 볼륨이 지원해야 하는 최대 IOPS(초당 입/출력 작업) 수를 입력합니다.
10. 가용 영역에서 볼륨을 생성할 가용 영역을 선택합니다. 동일한 가용 영역의 EC2 인스턴스에만 EBS 볼륨을 연결할 수 있습니다.
11. (선택 사항) 추가 태그 생성(Create additional tags)을 선택하여 볼륨에 태그를 추가합니다. 각 태그에 대해 태그 키와 태그 값을 제공합니다.
12. 볼륨 생성을 선택합니다.
13. 스냅샷에서 볼륨을 복원한 후에는 이 볼륨을 인스턴스에 연결하여 사용할 수 있습니다. 자세한 내용은 Amazon EBS 볼륨을 인스턴스에 연결 (p. 910) 단원을 참조하십시오.
14. 볼륨이 스냅샷의 기본 볼륨보다 큰 볼륨의 스냅샷으로 복구된 경우 볼륨의 파일 시스템을 확장하여 추가 공간을 활용할 수 있어야 합니다. 자세한 내용은 Amazon EBS 탄력적 볼륨 (p. 958) 단원을 참조하십시오.

명령줄을 사용하여 스냅샷에서 EBS 볼륨을 생성하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- create-volumeAWS CLI
- New-EC2VolumeWindows PowerShell용 AWS 도구

Amazon EBS 볼륨을 인스턴스에 연결

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

사전 조건

- 인스턴스에 연결할 수 있는 볼륨 수를 확인합니다. 자세한 내용은 인스턴스 볼륨 제한 (p. 1046) 단원을 참조하십시오.
- 볼륨이 암호화된 경우에는 Amazon EBS 암호화를 지원하는 인스턴스에만 연결할 수 있습니다. 자세한 내용은 지원되는 인스턴스 유형 (p. 969) 단원을 참조하십시오.
- 볼륨에 AWS Marketplace 제품 코드가 있는 경우:
  - 해당 볼륨은 중지된 인스턴스에만 연결될 수 있습니다.

- 이 경우 볼륨에 있는 AWS Marketplace 코드에 구독되어 있어야 합니다.
- 인스턴스 구성(인스턴스 유형, 운영 체제)에서 해당 AWS Marketplace 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 ELASTIC BLOCK STORE, 볼륨을 선택합니다.
3. 사용 가능한 볼륨을 선택한 후 작업, 볼륨 연결을 선택합니다.
4. 인스턴스의 경우, 인스턴스 이름 혹은 ID를 입력하십시오. 옵션 목록에서 인스턴스를 선택하십시오.(보기 볼륨과 동일한 가용 영역의 인스턴스만 표시됩니다.)
5. 디바이스의 경우, 제안된 디바이스 이름을 사용하거나 지원되는 다른 디바이스 이름을 입력할 수 있습니다. 자세한 내용은 Linux 인스턴스의 디바이스 명명 (p. 1047) 단원을 참조하십시오.
6. 연결을 선택합니다.
7. 인스턴스에 연결하고 볼륨을 탑재합니다. 자세한 내용은 Amazon EBS 볼륨을 Linux에서 사용할 수 있도록 만들기 (p. 914) 단원을 참조하십시오.

명령줄을 사용하여 EBS 볼륨을 인스턴스에 연결하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- attach-volumeAWS CLI
- Add-EC2Volume(Windows PowerShell용 AWS 도구)

Amazon EBS 다중 연결을 사용하여 여러 인스턴스에 볼륨 연결

Amazon EBS 다중 연결을 사용하면 단일 프로비저닝된 IOPS SSD(IO1) 볼륨을 동일한 가용 영역에 있는 최대 16개의 Nitro 기반 인스턴스에 연결할 수 있습니다. 여러 다중 연결 지원 볼륨을 인스턴스 또는 인스턴스 집합에 연결할 수 있습니다. 볼륨이 연결된 각 인스턴스는 공유된 볼륨에 대한 전체 읽기 및 쓰기 권한을 가집니다. 다중 연결을 사용하면 동시 쓰기 작업을 관리하는 클러스터링된 Linux 애플리케이션에서 더 쉽게 더 높은 애플리케이션 가용성을 얻을 수 있습니다.

다중 연결 지원 볼륨은 다음을 비롯하여 일반 Amazon EBS 볼륨에서 지원하는 여러 기능을 지원합니다.

- 태그 지정 (p. 1068)
- Amazon EBS 스냅샷 (p. 926)
- Amazon EBS 스냅샷 수명 주기 자동화 (p. 947)
- Amazon EBS Encryption (p. 968)
- Amazon EBS의 Amazon CloudWatch 지표 (p. 1011)
- Amazon EBS용 Amazon CloudWatch Events (p. 1017)

주제

- 고려 사항 및 제한 (p. 267)
- 성능 (p. 912)
- 다중 연결을 사용한 작업 (p. 912)
- 모니터링 (p. 914)

- 요금 및 결제 (p. 914)

고려 사항 및 제한

- 다중 연결 지원 볼륨은 I/O 차단 기능을 지원하지 않습니다. I/O 차단 프로토콜은 데이터 일관성을 유지하기 위해 공유된 스토리지 환경에서 쓰기 액세스를 제어합니다. 애플리케이션은 데이터 일관성을 유지하기 위해 연결된 인스턴스에 대해 쓰기 순서를 제공해야 합니다.
- 다중 연결 지원 볼륨은 동일한 가용 영역에 있는 최대 16개의 Nitro 기반 인스턴스 (p. 181)에 연결할 수 있습니다.
- 다중 연결은 프로비저닝된 IOPS SSD(IO1) 볼륨 (p. 897)에서만 지원됩니다.
- 다중 연결은 `us-east-1`, `us-east-2`, `us-west-2`, `eu-west-1` 및 `ap-northeast-2` 리전에서 사용할 수 있습니다.
- 다중 연결 지원 볼륨은 부팅 볼륨으로 만들 수 없습니다.
- 다중 연결 지원 볼륨은 인스턴스당 하나의 블록 디바이스 매핑에 연결할 수 있습니다.
- 볼륨 생성 후에는 다중 연결을 활성화하거나 비활성화할 수 없습니다.
- 다중 연결 지원 볼륨의 볼륨 유형, 크기 또는 프로비저닝된 IOPS는 변경할 수 없습니다.
- 인스턴스 시작 중에는 Amazon EC2 콘솔 또는 RunInstances API를 사용하여 다중 연결을 활성화할 수 없습니다.
- Amazon EBS 인프라 계층에서 문제가 있는 다중 연결 지원 볼륨은 연결된 모든 인스턴스에서 사용할 수 없습니다. Amazon EC2 또는 네트워킹 계층의 문제는 일부 연결된 인스턴스에만 영향을 줄 수 있습니다.

성능

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

예를 들어 `50,000` 프로비저닝된 IOPS를 사용하여 `io1` 다중 연결 지원 볼륨을 생성하고 `m5.8xlarge` 인스턴스와 `c5.12xlarge` 인스턴스에 연결한다고 가정합니다. `m5.8xlarge` 및 `c5.12xlarge` 인스턴스는 각각 최대 `30,000` 및 `40,000` IOPS를 지원합니다. 각 인스턴스는 최대 IOPS가 볼륨의 프로비저닝된 IOPS인 `50,000`보다 작기 때문에 최대 IOPS를 구동할 수 있습니다. 그러나 두 인스턴스 모두 볼륨에 대한 I/O를 동시에 구동하는 경우 결합된 IOPS는 볼륨의 프로비저닝된 성능인 `50,000 IOPS`를 초과할 수 없습니다. 볼륨이 `50,000` IOPS를 초과하지 않습니다.

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

다중 연결을 사용한 작업

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

목차

- 다중 연결 활성화 (p. 912)
- 인스턴스에 볼륨 연결 (p. 913)
- 종료 시 삭제 (p. 913)

다중 연결 활성화

생성하는 동안에만 Amazon EBS 볼륨에 대한 다중 연결을 활성화할 수 있습니다.

다음 방법 중 하나를 사용하여 생성 중 Amazon EBS 볼륨에 대한 다중 연결을 활성화합니다.

Console

볼륨 생성 중에 다중 연결을 활성화하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 볼륨을 선택합니다.
3. 볼륨 생성을 선택합니다.
4. 볼륨 유형의 경우 프로비저닝된 IOPS SSD(IO1)를 선택합니다.
5. 크기 및 IOPS의 경우 필요한 볼륨 크기와 프로비저닝할 IOPS 수를 선택합니다.
6. 가용 영역의 경우 인스턴스가 있는 가용 영역과 동일한 가용 영역을 선택합니다.
7. Multi-Attach(다중 연결)의 경우 활성화를 선택합니다.
8. 볼륨 생성을 선택합니다.

Command line

볼륨 생성 중에 다중 연결을 활성화하려면

create-volume 명령을 사용하고 `--multi-attach-enabled` 파라미터를 지정합니다.

```
$ aws ec2 create-volume --volume-type io1 --multi-attach-enabled --size 100 --iops 2000
 --region us-west-2 --availability-zone us-west-2b
```

인스턴스에 볼륨 연결

일반 볼륨을 연결하는 것과 동일한 방식으로 다중 연결 지원 볼륨을 인스턴스에 연결합니다. 자세한 내용은 Amazon EBS 볼륨을 인스턴스에 연결 (p. 910) 단원을 참조하십시오.

종료 시 삭제

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

종료 시 삭제 동작을 예측 가능하도록 하려면 볼륨이 연결된 모든 인스턴스에 대해 종료 시 삭제를 활성화 또는 비활성화합니다.

기본적으로 볼륨이 인스턴스에 연결되면 블록 디바이스 매핑에 대한 종료 시 삭제 설정이 false로 설정됩니다. 다중 연결 지원 볼륨에 대해 종료 시 삭제를 설정하려면 블록 디바이스 매핑을 수정합니다.

연결된 인스턴스가 종료될 때 볼륨을 삭제하려면 연결된 모든 인스턴스에 대해 블록 디바이스 매핑에서 종료 시 삭제를 활성화합니다. 연결된 인스턴스가 종료된 후 볼륨을 유지하려면 연결된 모든 인스턴스에 대해 블록 디바이스 매핑에서 종료 시 삭제를 비활성화합니다. 자세한 내용은 인스턴스 종료 시 Amazon EBS 볼륨 보존 (p. 526) 단원을 참조하십시오.

시작 시 또는 시작된 후에 인스턴스의 종료 시 삭제 설정을 수정할 수 있습니다. 인스턴스 시작 중에 종료 시 삭제를 활성화하거나 비활성화하면 시작 시 연결된 볼륨에만 설정이 적용됩니다. 시작 후 인스턴스에 볼륨을 연결하는 경우 해당 볼륨에 대해 종료 시 삭제 동작을 명시적으로 설정해야 합니다.

명령줄 도구만 사용하여 인스턴스의 종료 시 삭제 설정을 수정할 수 있습니다.

기존 인스턴스에 대한 종료 시 삭제 설정을 수정하려면

modify-instance-attribute 명령을 사용하고 `--block-device-mappings option`에서 `DeleteOnTermination` 속성을 지정합니다.

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

`mapping.json`에서 다음을 지정합니다.

```
[
  {
    "DeviceName": "/dev/sdf",
    "Ebs": {
      "DeleteOnTermination": true|false
    }
  }
]
```

모니터링

Amazon EBS 볼륨에 대한 CloudWatch 지표를 사용하여 다중 연결 지원 볼륨을 모니터링할 수 있습니다. 자세한 내용은 Amazon EBS의 Amazon CloudWatch 지표 (p. 1011) 단원을 참조하십시오.

연결된 모든 인스턴스에서 데이터가 집계됩니다. 연결된 개별 인스턴스에 대한 지표를 모니터링할 수는 없습니다.

요금 및 결제

Amazon EBS 다중 연결 사용에 따르는 추가 비용은 없습니다. 프로비저닝된 IOPS SSD(IO1) 볼륨에 적용되는 표준 요금이 청구됩니다. 자세한 내용은 Amazon EBS 요금을 참조하십시오.

Amazon EBS 볼륨을 Linux에서 사용할 수 있도록 만들기

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

다른 볼륨을 생성할 때 기준으로 사용하거나 백업을 목적으로 EBS 볼륨의 스냅샷을 생성할 수 있습니다. 자세한 내용은 Amazon EBS 스냅샷 (p. 926) 단원을 참조하십시오.

Windows 인스턴스의 볼륨에 대한 지침은 Windows 인스턴스용 Amazon EC2 사용 설명서에서 Windows에서 볼륨을 사용 가능하게 만들기 단원을 참조하십시오.

연결된 볼륨 포맷 및 탑재

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

EBS 볼륨을 Linux에서 포맷 및 탑재

1. SSH를 사용하여 인스턴스에 연결합니다. 자세한 내용은 Linux 인스턴스에 연결 (p. 485) 단원을 참조하십시오.
2. 디바이스는 블록 디바이스 매핑에 지정한 것과는 다른 디바이스 이름으로 인스턴스에 연결할 수 있습니다. 자세한 내용은 Linux 인스턴스의 디바이스 명명 (p. 1047) 단원을 참조하십시오. lsblk 명령을 사용하면 사용 가능한 디스크 디바이스 및 마운트 포인트(해당하는 경우)가 표시되어 사용 가능한 올바른 디바이스 이름을 결정하는 데 도움을 받을 수 있습니다. lsblk 명령의 출력에서는 전체 디바이스 경로 중 맨 앞에 `/dev/`가 생략됩니다.

다음은 EBS 볼륨을 NVMe 블록 디바이스로 노출하는, Nitro 기반 인스턴스 (p. 181)의 예시 출력입니다. 루트 디바이스는 `/dev/nvme0n1`입니다. 연결된 볼륨은 아직 탑재되지 않은 `/dev/nvme1n1`입니다.

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

다음은 T2 인스턴스의 예시 출력입니다. 루트 디바이스는 `/dev/xvda`입니다. 연결된 볼륨은 아직 탑재되지 않은 `/dev/xvdf`입니다.

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

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

   file -s 명령을 사용하면 파일 시스템 유형 등의 디바이스 정보를 확인할 수 있습니다. 다음 예시 출력에서처럼 출력에 `data`만 표시된다면, 디바이스에는 파일 시스템이 없으며 사용자가 직접 생성해야 합니다.

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

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

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

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

   > **Warning**
   >
   > 이미 데이터가 있는 볼륨(예: 스냅샷에서 복원한 볼륨)을 탑재한다면 이 명령을 사용하지 마십시오. 아니면 볼륨을 포맷하여 기존 데이터를 삭제합니다.

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

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

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

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

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

6. 다음 명령을 사용하여 이전 단계에서 생성한 디렉터리에 볼륨을 탑재합니다.

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

7. 새 볼륨 마운트의 파일 권한을 검토하여 사용자 및 애플리케이션이 볼륨에 기록할 수 있는지 확인합니다. 파일 권한에 대한 자세한 내용은 Linux Documentation Project에서 File security 단원을 참조하십시오.
8. 탑재 지점은 인스턴스를 재부팅하면 자동으로 보존되지 않습니다. 재부팅 후에도 이 EBS 볼륨을 자동으로 탑재하고 싶다면, 재부팅 후에도 연결된 볼륨을 자동으로 탑재 (p. 916) 단원을 참조하십시오.

재부팅 후에도 연결된 볼륨을 자동으로 탑재

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

`/etc/fstab`에 있는 `/dev/xvdf` 같은 디바이스 이름을 사용할 수 있습니다. 하지만 디바이스의 128비트 UUID(Universally Unique Identifier)를 사용할 것을 권장합니다. 디바이스 이름은 바꿀 수 있지만, UUID는 파티션 수명이 다할 때까지 유지됩니다. UUID를 사용하면 하드웨어 재구성 후 시스템을 부팅할 수 없게 되는 경우가 줄어듭니다. 자세한 내용은 EBS 디바이스 식별 (p. 982) 단원을 참조하십시오.

재부팅 후 연결된 볼륨을 자동으로 탑재하는 방법

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

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

2. blkid 명령을 사용하여 디바이스의 UUID를 찾습니다.

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

   Ubuntu 18.04의 경우 lsblk 명령을 사용합니다.

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

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

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

4. 다음 항목을 `/etc/fstab`에 추가해 디바이스를 지정된 탑재 지점에 탑재합니다. 필드는 blkid가 반환하는 UUID 값(또는 Ubuntu 18.04의 경우 lsblk), 탑재 지점, 파일 시스템, 권장하는 파일 시스템 탑재 옵션입니다. 자세한 내용은 fstab의 매뉴얼 페이지(man fstab 실행)를 참조하십시오.

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

   Note

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

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

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

오류 메시지가 표시된다면, 파일의 오류를 처리하십시오.

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

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

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

Amazon EBS 볼륨에 대한 정보 보기

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

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

설명이 포함된 정보 보기

콘솔을 사용하여 EBS 볼륨에 대한 정보를 확인하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 볼륨을 선택합니다.
3. 볼륨에 대한 자세한 정보를 확인하려면 선택합니다. 세부 정보 창에서 볼륨에 대해 제공된 정보를 검사할 수 있습니다.
4. 세부 정보 창에서 볼륨에 대해 제공된 정보를 검사할 수 있습니다.

인스턴스에 연결된 EBS 볼륨을 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 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) 단원을 참조하십시오.

- describe-volumes(AWS CLI)
- Get-EC2Volume (Windows PowerShell용 AWS 도구)

여유 디스크 공간 보기

사용 가능한 디스크 공간 등 EBS 볼륨에 대한 추가 정보를 인스턴스의 Linux 운영 체제에서 가져올 수 있습니다. 예를 들어, 다음 명령을 사용합니다.

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

볼륨 상태 모니터링

Amazon Web Services(AWS)에서는 Amazon Elastic Block Store(Amazon EBS) 볼륨을 모니터링하는 데 사용할 수 있는 데이터를 자동으로 제공합니다.

목차

- EBS 볼륨 상태 확인 (p. 918)
- EBS 볼륨 이벤트 (p. 920)
- 손상된 볼륨 작업 (p. 921)
- 자동 활성화된 IO 볼륨 속성 작업 (p. 923)

모니터링에 대한 자세한 내용은 Amazon EBS의 Amazon CloudWatch 지표 (p. 1011) 및 Amazon EBS용 Amazon CloudWatch Events (p. 1017) 단원을 참조하십시오.

EBS 볼륨 상태 확인

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

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

Amazon EBS에서 볼륨의 데이터가 잠재적으로 일치하지 않는 것으로 확인하면 데이터 손상을 방지하기 위해 기본적으로 연결된 EC2 인스턴스에서 볼륨으로의 I/O가 비활성화됩니다. I/O가 비활성화되면 다음 볼륨 상태 확인에 실패하고 볼륨 상태는 `impaired`가 됩니다. 또한 I/O가 비활성화되었으며 볼륨에 대한 I/O를 활성화하여 볼륨의 손상된 상태를 해결할 수 있다고 알려주는 이벤트가 표시됩니다. I/O를 활성화한 다음 인스턴스에서 볼륨을 계속 사용할지 아니면 fsck와 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 아니고, 볼륨이 손상된 경우 볼륨을 즉시 사용할 수 있게 하려면 I/O를 자동으로 활성화하도록 볼륨을 구성하여 기본 동작을 무시할 수 있습니다. IO 자동 활성화 볼륨 속성(API의 `autoEnableIO`)을 활성화하면 볼륨 상태 검사를 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 상

태 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 성능 상태 확인에 대해 `warning` 상태를 표시할 수 있습니다. 이는 원래 그런 것이므로 초기화 중에는 `warning` 볼륨에 대한 `io1` 상태를 무시해도 됩니다. 자세한 내용은 Amazon EBS 볼륨 초기화 (p. 1001) 단원을 참조하십시오.

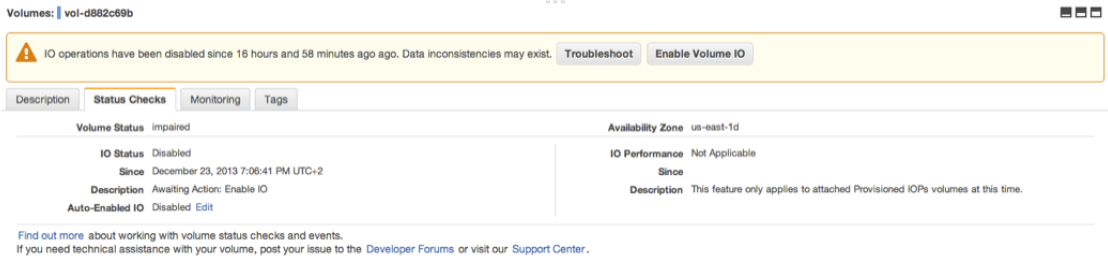
다음 표에는 Amazon EBS 볼륨에 대한 상태가 나와 있습니다.

볼륨 상태	I/O 활성화 상태	I/O 성능 상태(프로비저닝된 IOPS 볼륨에만 사용 가능)
`ok`	활성화됨(I/O 활성화 또는 I/O 자동 활성화)	정상(볼륨 성능이 예상대로임)
`warning`	활성화됨(I/O 활성화 또는 I/O 자동 활성화)	성능 저하(볼륨 성능이 예상보다 낮음) 심각한 성능 저하(볼륨 성능이 예상보다 훨씬 낮음)
`impaired`	활성화됨(I/O 활성화 또는 I/O 자동 활성화) 비활성화됨(볼륨이 오프라인이고 복구 보류 중이거나 사용자가 I/O를 활성화하기를 기다리는 중)	중단됨(볼륨 성능이 저하됨) 사용할 수 없음(I/O가 비활성화되어 I/O 성능을 확인할 수 없음)
`insufficient-data`	활성화됨(I/O 활성화 또는 I/O 자동 활성화) 데이터 부족	데이터 부족

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 볼륨을 선택합니다. 볼륨 상태 열에 각 볼륨의 작업 상태가 나열됩니다.
3. 볼륨의 상태 세부 정보를 보려면 볼륨을 선택하고 상태 검사를 선택합니다.



4. 상태 검사에 실패한 볼륨이 있는 경우(상태가 손상됨) 손상된 볼륨 작업 (p. 921) 단원을 참조하십시오.

또는 탐색기에서 이벤트 창을 선택하여 인스턴스와 볼륨에 대한 모든 이벤트를 볼 수 있습니다. 자세한 내용은 EBS 볼륨 이벤트 (p. 920) 단원을 참조하십시오.

명령줄로 볼륨 상태 정보를 보려면

다음 명령 중 하나를 사용하여 Amazon EBS 볼륨의 상태를 볼 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- describe-volume-status(AWS CLI)
- Get-EC2VolumeStatus(Windows PowerShell용 AWS 도구)

EBS 볼륨 이벤트

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

데이터가 잠재적으로 일치하지 않는 볼륨에서 I/O를 자동으로 활성화하려면 IO 자동 활성화 볼륨 속성(API의 `autoEnableIO`)의 설정을 변경합니다. 이 속성 변경에 대한 자세한 내용은 손상된 볼륨 작업 (p. 921) 단원을 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/O가 비활성화됩니다. I/O를 명시적으로 활성화하면 이벤트 설명이 IO Enabled로 변경됩니다.

IO Enabled

이 볼륨에 대해 I/O 작업이 명시적으로 활성화되었습니다.

IO Auto-Enabled

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

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

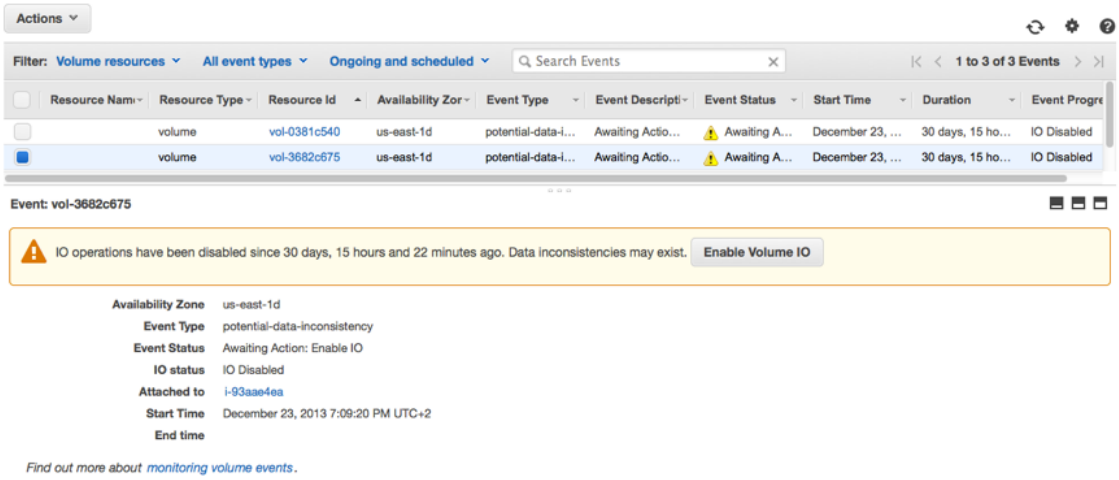
`io1` 볼륨 전용입니다. 볼륨 성능이 저하되었습니다.

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 이벤트를 선택합니다. 이벤트가 있는 모든 인스턴스와 볼륨이 나열됩니다.

3. 볼륨을 기준으로 필터링하여 볼륨 상태만 볼 수 있습니다. 특정 상태 유형을 기준으로 필터링할 수도 있습니다.
4. 특정 이벤트를 보려는 볼륨을 선택합니다.



I/O가 비활성화된 볼륨이 있는 경우 손상된 볼륨 작업 (p. 921) 단원을 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. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- describe-volume-status(AWS CLI)
- Get-EC2VolumeStatus(Windows PowerShell용 AWS 도구)

손상된 볼륨 작업

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

옵션

- 옵션 1: 인스턴스에 연결된 볼륨에 대한 일관성 확인 수행 (p. 921)
- 옵션 2: 다른 인스턴스를 사용하여 볼륨에 대한 일관성 확인 수행 (p. 922)
- 옵션 3: 볼륨이 더 이상 필요하지 않은 경우 볼륨 삭제 (p. 923)

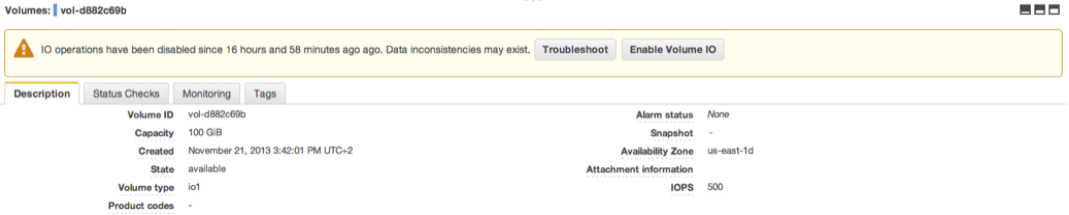
옵션 1: 인스턴스에 연결된 볼륨에 대한 일관성 확인 수행

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 볼륨을 선택합니다.

c. I/O 작업을 활성화할 볼륨을 선택합니다.
d. 세부 정보 창에서 볼륨 IO 활성화를 선택한 다음, 예, 활성화를 선택합니다.



3. 볼륨의 데이터를 확인합니다.

   a. fsck 명령을 실행합니다.
   b. (선택 사항) 애플리케이션 또는 시스템 로그에 관련 오류 메시지가 있는지 검토합니다.
   c. 볼륨 손상 상태가 20분 이상 지속된 경우 AWS 지원 센터에 문의할 수 있습니다. 문제 해결을 선택한 다음 상태 검사 문제 해결 대화 상자에서 고객 지원을 선택하여 지원 사례를 제출합니다.

명령줄로 볼륨에 대한 I/O를 활성화하려면

다음 명령 중 하나를 사용하여 Amazon EBS 볼륨에 대한 이벤트 정보를 볼 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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.

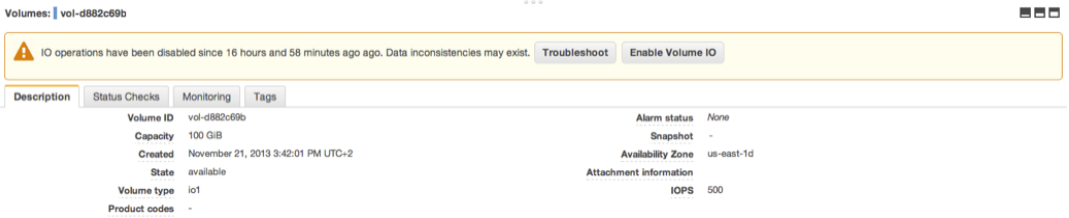
> **Important**
>
> 이 절차를 수행하면 볼륨 I/O가 비활성화된 상태에서 일시 중지된 쓰기 I/O가 손실될 수 있습니다.

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 볼륨을 선택합니다.
   c. 분리할 볼륨을 선택합니다.
   d. 작업, 볼륨 강제 분리를 선택합니다. 확인 메시지가 나타납니다.
3. 볼륨에서 I/O를 활성화합니다.

   a. 탐색 창에서 볼륨을 선택합니다.
   b. 이전 단계에서 분리한 볼륨을 선택합니다.
   c. 세부 정보 창에서 볼륨 IO 활성화를 선택한 다음, 예, 활성화를 선택합니다.



4. 볼륨을 다른 인스턴스에 연결합니다. 자세한 내용은 인스턴스 시작 (p. 430) 및 Amazon EBS 볼륨을 인스턴스에 연결 (p. 910) 단원을 참조하십시오.
5. 볼륨의 데이터를 확인합니다.
   a. fsck 명령을 실행합니다.
   b. (선택 사항) 애플리케이션 또는 시스템 로그에 관련 오류 메시지가 있는지 검토합니다.
   c. 볼륨 손상 상태가 20분 이상 지속된 경우 AWS 지원 센터에 문의할 수 있습니다. 문제 해결을 선택하고 문제 해결 대화 상자에서 고객 지원을 선택하여 지원 사례를 제출합니다.

명령줄로 볼륨에 대한 I/O를 활성화하려면

다음 명령 중 하나를 사용하여 Amazon EBS 볼륨에 대한 이벤트 정보를 볼 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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. 926) 단원을 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. 909) 단원을 참조하십시오.

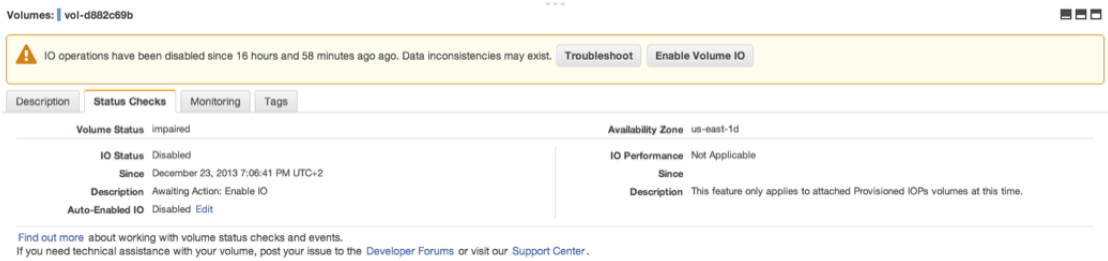
자동 활성화된 IO 볼륨 속성 작업

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 아니고, 볼륨이 손상된 상태인 경우 볼륨을 즉시 사용할 수 있게 하려면 I/O를 자동으로 활성화하도록 볼륨을 구성하여 기본 동작을 재정의할 수 있습니다. 자동 활성화된 IO 볼륨 속성(API의 `autoEnableIO`)을 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 EBS 볼륨 이벤트 (p. 920) 단원을 참조하십시오.

이 절차에서는 볼륨의 자동 활성화된 IO 속성을 보고 수정하는 방법을 설명합니다.

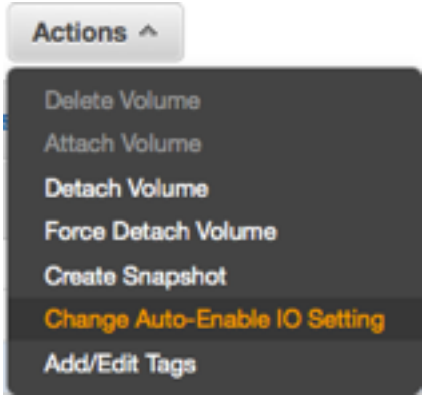
콘솔에서 볼륨의 자동 활성화된 IO 속성을 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 볼륨을 선택합니다.
3. 볼륨을 선택하고 상태 검사를 선택합니다. 자동 활성화된 IO는 볼륨에 대한 현재의 설정(활성화 또는 비활성)을 표시합니다.

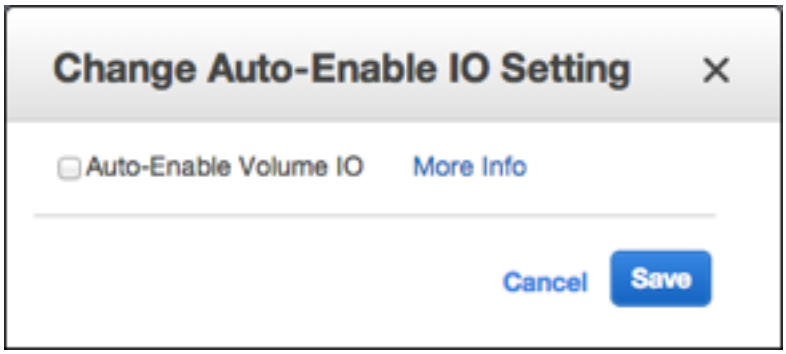


콘솔에서 볼륨의 자동 활성화된 IO 속성을 수정하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 볼륨을 선택합니다.
3. 볼륨을 선택하고 작업, IO 자동 활성화 설정 변경을 선택합니다. 또는 상태 검사 탭을 선택하고 자동 활성화된 IO에서 편집을 선택합니다.



4. 볼륨 IO 자동 활성화 확인란을 선택하여 손상된 볼륨에 대한 I/O를 자동으로 활성화합니다. 이 기능을 비활성화하려면 확인란의 선택을 취소합니다.



5. Save를 선택합니다.

명령줄로 볼륨의 AutoEnableIO 속성을 보거나 수정하려면

다음 명령 중 하나를 사용하여 Amazon EBS 볼륨의 `autoEnableIO` 속성을 볼 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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 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. 926) 단원을 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 Amazon EBS 볼륨에 대한 정보 보기 (p. 917) 단원을 참조하십시오.

분리된(탑재를 해제하지 않고) 볼륨을 다시 연결할 수 있지만 동일 탑재 지점을 가져올 수는 없습니다. 분리된 상태에서 진행 중인 볼륨 쓰기 작업이 있으면 볼륨의 데이터가 동기화되지 않을 수 있습니다.

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 볼륨을 선택합니다.
4. 볼륨을 선택한 후 작업, 볼륨 분리를 선택합니다.
5. 확인 대화 상자에서 예, 분리를 선택합니다.

명령줄을 사용하여 인스턴스에서 EBS 볼륨을 분리하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- detach-volume(AWS CLI)
- Dismount-EC2Volume(Windows PowerShell용 AWS 도구)

문제 해결

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

> **Note**
>
> 데이터 손실에 대비하여 볼륨을 해제하기 전 볼륨 스냅샷을 만들어 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-volumes를 참조하십시오.
- 볼륨이 `detaching` 상태를 유지하는 경우 강제 분리를 선택하여 강제 분리할 수 있습니다. 이 옵션은 오류가 발생한 인스턴에서 볼륨 분리 또는 삭제할 목적으로 볼륨을 분리하는 경우에만 최후의 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 forum에 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.

```
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. 자세한 내용은 인스턴스에서 Amazon EBS 볼륨 분리 (p. 924) 단원을 참조하십시오.

콘솔을 사용하여 EBS 볼륨을 삭제하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 볼륨을 선택합니다.
3. 볼륨을 선택한 후 작업, 볼륨 삭제를 선택합니다.
4. 확인 대화 상자에서 예, 삭제를 선택합니다.

명령줄을 사용하여 EBS 볼륨을 삭제하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- delete-volume(AWS CLI)
- Remove-EC2Volume(Windows PowerShell용 AWS 도구)

Amazon EBS 스냅샷

지정 시간 스냅샷을 만들어 Amazon S3에 Amazon EBS 볼륨의 데이터를 백업할 수 있습니다. 스냅샷은 증분식 백업이어서 마지막 스냅샷 이후 변경된 디바이스의 블록만이 저장됩니다. 그러면 스냅샷을 만드는 데 필요한 시간이 최소화되며 데이터를 복제하지 않으므로 스토리지 비용이 절약됩니다. 스냅샷을 삭제하면 해

당 스냅샷에 고유한 데이터만 제거됩니다. 각 스냅샷에는 (스냅샷을 만든 시점의) 데이터를 새 EBS 볼륨에 복원하는 데 필요한 모든 정보가 들어 있습니다.

스냅샷을 기반으로 EBS 볼륨을 생성하는 경우, 새 볼륨은 해당 스냅샷을 생성하는 데 사용된 원본 볼륨과 정확히 일치합니다. 복제된 볼륨은 사용자가 즉시 사용할 수 있도록 백그라운드에서 데이터를 로드합니다. 아직 로드되지 않은 데이터에 액세스하는 경우, 볼륨은 요청한 데이터를 Amazon S3에서 즉시 다운로드한 후 백그라운드에서 볼륨의 나머지 데이터를 계속해서 로드합니다. 자세한 내용은 Amazon EBS 스냅샷 만들기 (p. 929) 단원을 참조하십시오.

다중 볼륨 스냅샷

스냅샷을 사용하여 여러 EBS 볼륨에 걸쳐 있는 파일 시스템 또는 대규모 데이터베이스 등의 중요한 워크로드 백업을 생성할 수 있습니다. 다중 볼륨 스냅샷을 통해 EC2 인스턴스에 연결된 여러 EBS 볼륨에서 정확한 특정 시점, 데이터 조정 및 충돌 일치 스냅샷을 생성할 수 있습니다. 스냅샷은 여러 EBS 볼륨에서 자동으로 생성되기 때문에 더 이상 인스턴스를 중지하거나, 중단 일관성을 유지하기 위해 볼륨 간을 조정할 필요가 없습니다. 자세한 내용은 Amazon EBS 스냅샷 만들기 (p. 929) 아래에서 다중 볼륨 EBS 스냅샷을 생성하기 위한 단계를 참조하십시오.

CloudWatch 이벤트를 통해 EBS 스냅샷의 상태를 추적할 수 있습니다. 자세한 내용은 Amazon EBS용 Amazon CloudWatch Events (p. 1017) 단원을 참조하십시오.

목차

- 증분 스냅샷의 작동 방법 (p. 927)
- 스냅샷 복사 및 공유 (p. 928)
- 스냅샷에 대한 암호화 지원 (p. 929)
- Amazon EBS 스냅샷 만들기 (p. 929)
- Amazon EBS 스냅샷 삭제 (p. 932)
- Amazon EBS 스냅샷 복사 (p. 934)
- Amazon EBS 스냅샷 정보 보기 (p. 937)
- Amazon EBS 스냅샷 공유 (p. 938)
- EBS 스냅샷 콘텐츠 액세스 (p. 940)
- Amazon EBS 스냅샷 수명 주기 자동화 (p. 947)

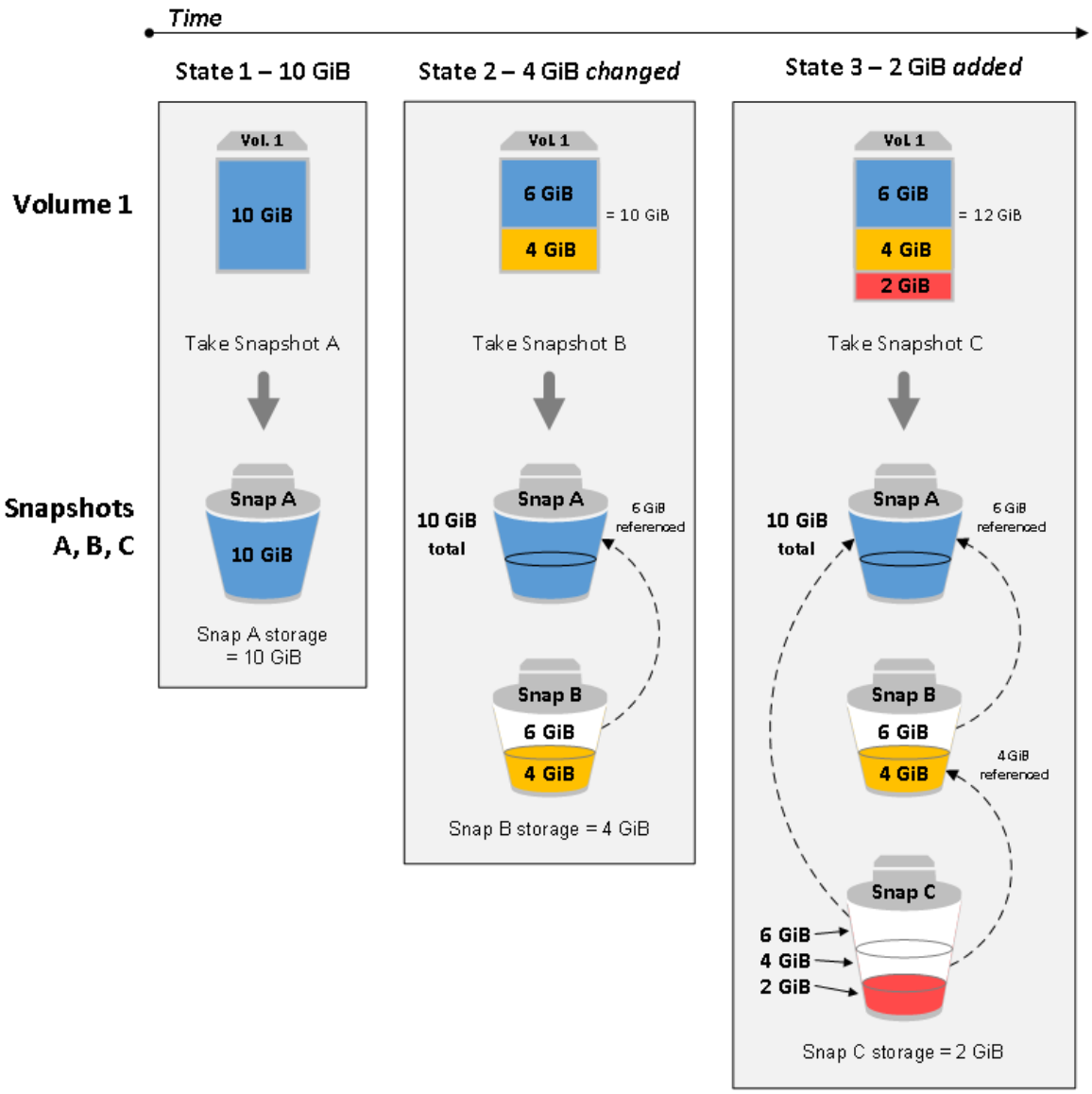
증분 스냅샷의 작동 방법

이 단원에서는 EBS 스냅샷이 특정 시점의 볼륨 상태를 캡처하는 방법과 변화하는 볼륨의 연속 스냅샷에 이러한 변경 기록이 표시되는 방법에 대해 설명합니다.

아래 다이어그램에서 볼륨 1은 세 가지 시점에 표시됩니다. 이 각 세 가지 볼륨 상태에 대한 스냅샷이 만들어집니다.

- 상태 1의 볼륨에는 10GiB의 데이터가 있습니다. 스냅 A는 이 볼륨의 첫 번째 스냅샷이므로 10GiB 데이터 전체를 복사해야 합니다.
- 상태 2의 볼륨에는 여전히 10GiB의 데이터가 있지만 4GiB가 변경되었습니다. 스냅 B는 스냅 A를 만든 후 변경된 4GiB만 복사하고 저장해야 합니다. 스냅 A에 이미 복사되어 저장된 변경되지 않은 나머지 6GiB 데이터는 (다시) 복사되는 것이 아니라 스냅 B에서 참조됩니다. 이는 파선 모양 화살표로 표시됩니다.
- 상태 3에서는 2GiB의 데이터가 볼륨에 추가되어 총 12GiB가 되었습니다. 스냅 C는 스냅 B를 만든 후 추가된 2GiB를 복사해야 합니다. 파선 모양 화살표로 표시되었듯이 스냅 C는 스냅 B에 저장된 4GiB의 데이터 및 스냅 A에 저장된 6GiB의 데이터를 참조합니다.
- 이 세 스냅샷에 필요한 총 스토리지는 16GiB입니다.

볼륨의 여러 스냅샷 간의 관계



Note

새 CMK로 스냅샷을 복사 및 암호화하면 항상 전체(증분 아님) 복사가 생성되고 추가 지연 및 스토리지 비용이 발생합니다.

스냅샷을 삭제할 때 데이터가 관리되는 방법에 대한 자세한 내용은 Amazon EBS 스냅샷 삭제 (p. 932) 단원을 참조하십시오.

스냅샷 복사 및 공유

액세스 권한을 수정하여 AWS 계정 간에 스냅샷을 공유할 수 있습니다. 자체 스냅샷뿐 아니라 공유된 스냅샷의 복사본을 만들 수 있습니다. 자세한 내용은 Amazon EBS 스냅샷 공유 (p. 938) 단원을 참조하십시오.

스냅샷은 해당 스냅샷이 생성된 AWS 리전으로 제한됩니다. EBS 볼륨의 스냅샷을 생성한 후, 해당 스냅샷을 사용하여 동일한 리전에 새로운 볼륨을 생성할 수 있습니다. 자세한 내용은 스냅샷에서 Amazon EBS 볼륨 복구 (p. 909) 단원을 참조하십시오. 또한 리전 전반에 스냅샷을 복사하면 지리적 확장, 데이터 센터 마이그레이션 및 재해 복구를 위해 여러 리전을 사용할 수 있습니다. 스냅샷의 복사 및 액세스는 스냅샷이

completed 상태인 경우에 가능합니다. 자세한 내용은 Amazon EBS 스냅샷 복사 (p. 934) 단원을 참조하십시오.

스냅샷에 대한 암호화 지원

EBS 스냅샷은 EBS 암호화를 전적으로 지원합니다.

- 암호화된 볼륨의 스냅샷은 자동으로 암호화됩니다.
- 암호화된 스냅샷에서 생성되는 볼륨은 자동으로 암호화됩니다.
- 사용자가 소유하거나 액세스 권한이 있는 암호화되지 않은 스냅샷에서 생성한 볼륨은 즉시 암호화할 수 있습니다.
- 자신이 소유한 암호화되지 않은 스냅샷을 복사할 때 복사 프로세스 중에 해당 스냅샷을 암호화할 수 있습니다.
- 자신이 소유하거나 액세스 권한이 있는 암호화된 스냅샷을 복사할 때는 복사 프로세스 중에 다른 키로 해당 스냅샷을 다시 암호화할 수 있습니다.
- 암호화되지 않은 스냅샷에서 생성된 암호화된 볼륨의 첫 번째 스냅샷은 항상 전체 스냅샷입니다.
- 소스 스냅샷과 비교하여 다른 CMK를 가진 재암호화된 볼륨의 첫 번째 스냅샷은 항상 전체 스냅샷입니다.

> **Note**
>
> 새 CMK로 스냅샷을 복사 및 암호화하면 항상 전체(증분 아님) 복사가 생성되고 추가 지연 및 스토리지 비용이 발생합니다.

가능한 스냅샷 암호화 시나리오에 대한 전체 문서는 Amazon EBS 스냅샷 만들기 (p. 929)및 Amazon EBS 스냅샷 복사 (p. 934)에서 제공됩니다.

자세한 내용은 Amazon EBS Encryption (p. 968) 단원을 참조하십시오.

Amazon EBS 스냅샷 만들기

EBS 볼륨의 특정 시점 스냅샷을 생성하여 새 볼륨이나 데이터 백업의 기준으로 사용할 수 있습니다. 볼륨의 스냅샷이 주기적으로 생성되는 경우 스냅샷은 증분식이어서 새 스냅샷은 마지막 스냅샷 이후 변경된 블록만 저장합니다.

스냅샷은 비동기적으로 생성됩니다. 특정 시점 스냅샷은 즉시 생성되지만 스냅샷이 완료될 때까지, 즉 수정된 블록이 pending로 모두 이동할 때까지 스냅샷 상태는 Amazon S3입니다. 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.

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.

스냅샷을 보다 쉽게 관리하기 위해 생성 중에 스냅샷에 태그를 지정하거나 나중에 태그를 추가할 수도 있습니다. 예를 들어, 스냅샷이 생성된 원래 볼륨 또는 원래 볼륨을 인스턴스에 연결하는 데 사용된 디바이스 이름을 설명하는 태그를 적용할 수 있습니다. 자세한 내용은 Amazon EC2리소스에 태그 지정 (p. 1068) 단원을 참조하십시오.

스냅샷 암호화

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 Encryption (p. 968) 단원을 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. 938) 단원을 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. 938) 단원을 참조하십시오.

> **Note**
>
> 새 CMK로 스냅샷을 복사 및 암호화하면 항상 전체(증분 아님) 복사가 생성되고 추가 지연 및 스토리지 비용이 발생합니다.

다중 볼륨 스냅샷

EC2 인스턴스에 연결된 모든 EBS 볼륨의 특정 시점 스냅샷인 다중 볼륨 스냅샷을 생성할 수 있습니다. 또한 다중 볼륨 스냅샷의 생성 및 보존을 자동화하는 수명 주기 정책을 생성할 수도 있습니다. 자세한 내용은 Amazon EBS 스냅샷 수명 주기 자동화 (p. 947) 단원을 참조하십시오.

스냅샷이 생성된 후 각 스냅샷은 개별 스냅샷으로 처리됩니다. 단일 볼륨 스냅샷과 마찬가지로, 복원, 삭제, 교차 리전/계정 복사 등의 모든 스냅샷 작업을 수행할 수 있습니다. 또한 단일 볼륨 스냅샷과 마찬가지로 다중 볼륨 스냅샷에도 태그를 지정할 수 있습니다. 복원, 복사 또는 보존 중에 한꺼번에 관리할 수 있도록 다중 볼륨 스냅샷에 태그를 지정하는 것이 좋습니다.

다중 볼륨, 충돌 일치 스냅샷은 일반적으로 세트로 복원됩니다. 인스턴스 ID, 이름 또는 기타 관련 세부 정보로 세트에 태그를 지정하면 충돌 일치 세트에 있는 스냅샷을 식별하는 데 도움이 됩니다. 소스 볼륨에서 해당 스냅샷으로 태그를 자동 복사하도록 선택할 수 있습니다. 그러면 액세스 정책, 연결 정보, 비용 할당 등의 스냅샷 메타데이터를 소스 볼륨과 일치하도록 설정하는 데 도움이 됩니다.

다중 볼륨 스냅샷은 생성 후 다른 모든 스냅샷과 마찬가지로 동작합니다. 리전 및 계정에서 복원, 삭제, 복사 등의 모든 작업을 수행할 수 있습니다. 스냅샷에 태그를 지정할 수도 있습니다. 복원, 복사 또는 보존 중에 한꺼번에 관리할 수 있도록 다중 볼륨 스냅샷에 태그를 지정하는 것이 좋습니다.

스냅샷을 생성하면 정확한 특정 시점에 생성된 스냅샷이 EC2 콘솔에 표시됩니다. 스냅샷은 한꺼번에 관리되므로 볼륨 세트의 스냅샷 하나가 실패하면 다른 모든 스냅샷도 오류 상태를 표시합니다.

고려 사항

다음은 스냅샷을 생성할 때 고려할 사항입니다.

- 루트 디바이스 역할을 하는 EBS 볼륨의 스냅샷을 생성할 때는 인스턴스를 중지한 후 스냅샷을 생성해야 합니다.
- 최대 절전 모드가 활성화된 인스턴스에서는 스냅샷을 생성할 수 없습니다.
- 최대 절전 모드 인스턴스에서는 스냅샷을 생성할 수 없습니다.
- 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 다섯 개의 `pending` 스냅샷, 단일 `st1` 또는 `sc1` 볼륨에 대해 한 개의 `pending` 스냅샷으로 제한됩니다. 동일 볼륨에 대해 여러 개의 동실 스냅샷을 생성하려 할 때 `ConcurrentSnapshotLimitExceeded` 오류가 표시되면 한 개 이상의 `pending` 스냅샷이 완료될 때까지 기다린 후 해당 볼륨의 다른 스냅샷을 생성하십시오.
- AWS Marketplace 제품 코드를 가진 볼륨에서 스냅샷을 만들면 해당 제품 코드가 스냅샷으로 전파됩니다.

스냅샷 생성

지정된 볼륨에서 스냅샷을 생성하려면 다음 절차에 따르십시오.

콘솔을 이용하여 스냅샷을 생성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창의 Elastic Block Store에서 스냅샷을 선택합니다.
3. Create Snapshot을 클릭합니다.
4. 리소스 유형 선택에서 볼륨을 선택합니다.
5. 볼륨에서 볼륨을 선택합니다.
6. (선택 사항) 스냅샷에 대한 설명을 입력합니다.
7. (선택 사항) 태그 추가를 선택하여 스냅샷에 태그를 추가합니다. 각 태그에 대해 태그 키와 태그 값을 제공합니다.
8. Create Snapshot을 클릭합니다.

명령줄을 이용하여 스냅샷을 생성하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- create-snapshot (AWS CLI)
- New-EC2Snapshot(Windows PowerShell용 AWS 도구)

다중 볼륨 스냅샷 생성

인스턴스의 볼륨에서 스냅샷을 생성하려면 다음 절차에 따르십시오.

콘솔을 사용하여 다중 볼륨 스냅샷을 생성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창의 Elastic Block Store에서 스냅샷을 선택합니다.
3. Create Snapshot을 클릭합니다.
4. 리소스 유형 선택에서 인스턴스를 선택합니다.
5. 연결된 모든 EBS 볼륨의 동시 백업을 생성할 인스턴스 ID를 선택합니다. 다중 볼륨 스냅샷은 인스턴스당 최대 40개의 EBS 볼륨을 지원합니다.
6. (선택 사항) Exclude root volume(루트 볼륨 제외)을 설정합니다.
7. (선택 사항) Copy tags from volume(볼륨에서 태그 복사) 플래그를 설정하여 태그를 소스 볼륨에서 해당 스냅샷으로 자동으로 복사합니다. 그러면 액세스 정책, 연결 정보, 비용 할당 등의 스냅샷 메타데이터가 소스 볼륨과 일치하도록 설정됩니다.
8. (선택 사항) 태그 추가를 선택하여 스냅샷에 태그를 추가합니다. 각 태그에 대해 태그 키와 태그 값을 제공합니다.
9. Create Snapshot을 클릭합니다.

   스냅샷 생성 중에 스냅샷이 함께 관리됩니다. 볼륨 세트의 스냅샷 중 하나가 실패하면 볼륨 세트의 다른 스냅샷이 오류 상태로 이동합니다. CloudWatch Events를 사용하여 스냅샷의 진행 상황을 모니터링할 수 있습니다. 스냅샷 생성 프로세스가 완료되면 CloudWatch는 해당 인스턴스의 상태 및 모든 관련 스냅샷 세부 정보가 포함된 이벤트를 생성합니다.

명령줄을 사용하여 다중 볼륨 스냅샷을 생성하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- create-snapshots(AWS CLI)
- New-EC2SnapshotBatch(Windows PowerShell용 AWS 도구)

Amazon EBS 스냅샷 삭제

스냅샷을 삭제하면 해당 스냅샷에서만 참조하는 데이터만 제거됩니다. 고유한 데이터는 해당 데이터를 참조하는 모든 스냅샷을 삭제하지 않는 한 삭제되지 않습니다. 볼륨의 이전 스냅샷을 삭제해도 해당 볼륨의 이후 스냅샷으로 볼륨을 복원하는 기능에는 영향을 주지 않습니다.

볼륨의 스냅샷을 삭제해도 해당 볼륨에는 아무런 영향을 미치지 않습니다. 볼륨을 삭제해도 해당 볼륨에서 만들어진 스냅샷에는 아무런 영향을 미치지 않습니다.

정기적으로 볼륨의 스냅샷을 만드는 경우, 스냅샷은 증분식으로 늘어납니다. 다시 말해 새 스냅샷에는 마지막 스냅샷 이후로 변경된 디바이스 블록만 저장됩니다. 스냅샷은 증분식으로 저장되지만 스냅샷 삭제 프로세스는 볼륨을 복구하기 위해 가장 최근의 스냅샷만을 유지할 수 있도록 설계됩니다. 볼륨에 존재했고, 이전 스냅샷 또는 스냅샷 시리즈에 저장되며 나중에 해당 볼륨에서 삭제되는 데이터는 여전히 이전 스냅샷의 고유한 데이터로 간주됩니다. 이 고유 데이터는 고유한 데이터를 참조하는 모든 스냅샷을 삭제하지 않는 한 스냅샷 시퀀스에서 삭제되지 않습니다.

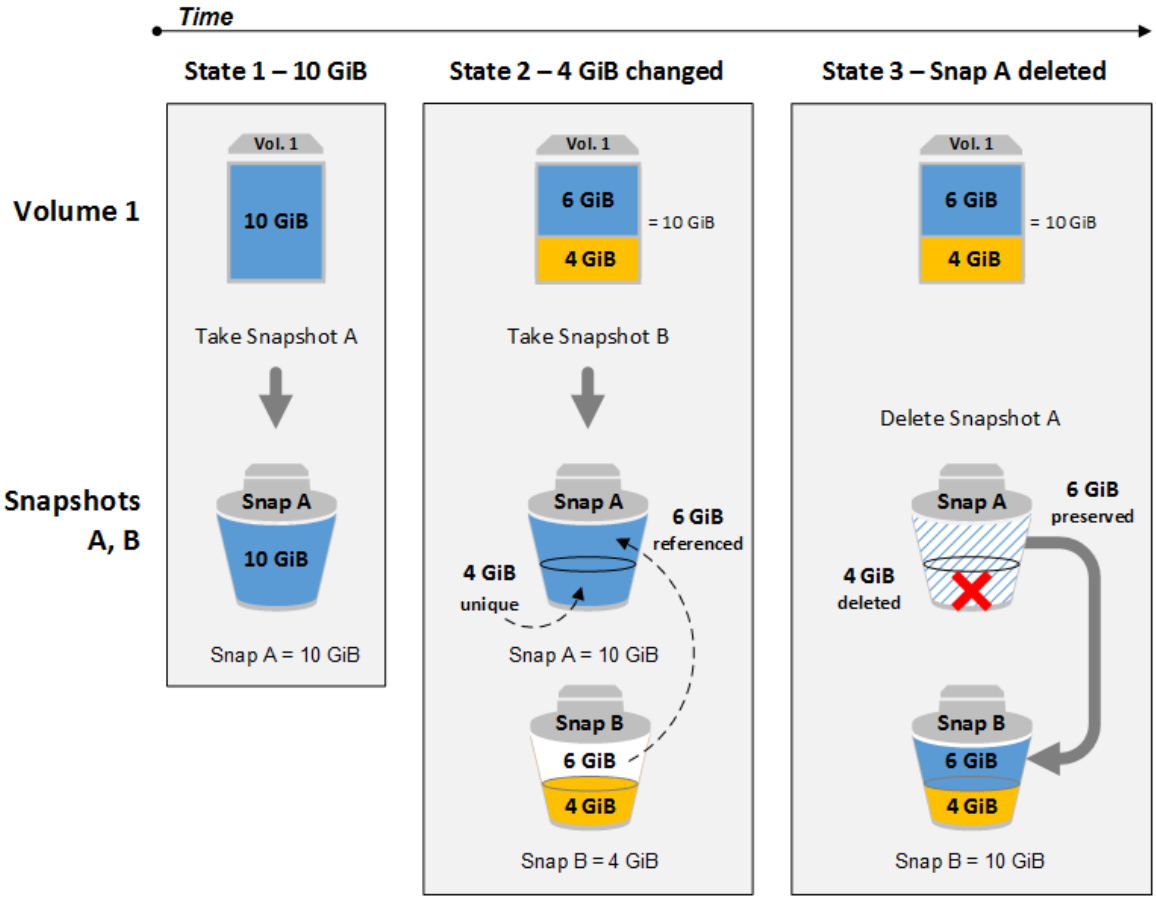
스냅샷을 삭제해도 조직의 데이터 스토리지 비용이 줄어들지 않을 수 있습니다. 다른 스냅샷은 해당 스냅샷의 데이터를 참조할 수 있으며, 참조된 데이터는 항상 보존됩니다. 이후의 스냅샷에서 사용 중인 데이터가 포함된 스냅샷을 삭제하는 경우, 참조된 데이터와 관련된 비용이 이후의 스냅샷에 할당됩니다. 스냅샷이 데이터를 저장하는 방법에 대한 자세한 내용은 증분 스냅샷의 작동 방법 (p. 927) 및 아래 예를 참조하십시오.

다중 볼륨 스냅샷을 삭제하려면 스냅샷을 생성했을 때 그룹에 적용한 태그를 사용하여 다중 볼륨 그룹의 모든 스냅샷을 검색합니다. 그런 다음 스냅샷을 개별적으로 삭제합니다. 다중 볼륨 스냅샷 그룹의 개별 스냅샷을 삭제할 수 있습니다.

다음 다이어그램에서 볼륨 1은 세 가지 시점에 표시됩니다. 스냅샷이 첫 두 상태를 각각 캡처했으며, 세 번째에서는 스냅샷이 삭제되었습니다.

- 상태 1의 볼륨에는 10GiB의 데이터가 있습니다. 스냅 A는 이 볼륨의 첫 번째 스냅샷이므로 10GiB 데이터 전체를 복사해야 합니다.
- 상태 2의 볼륨에는 여전히 10GiB의 데이터가 있지만 4GiB가 변경되었습니다. 스냅 B는 스냅 A를 만든 후 변경된 4GiB만 복사하고 저장해야 합니다. 스냅 A에 이미 복사되어 저장된 변경되지 않은 나머지 6GiB 데이터는 (다시) 복사되는 것이 아니라 스냅 B에서 참조됩니다. 이는 파선 모양 화살표로 표시됩니다.
- 상태 3에서 볼륨은 상태 2 이후로 변경되지 않았지만 스냅샷 A가 삭제되었습니다. 스냅샷 A에 저장된 6GiB의 데이터를 스냅샷 B에서 참조했었지만 굵은 화살표로 표시되었듯이 이제 스냅샷 B로 이동했습니다. 결과적으로 스냅 A에서 보존된 변경되지 않은 6GiB의 데이터와 스냅 B에서 변경된 4GiB의 데이터를 합쳐, 여전히 10GiB의 데이터를 저장한 것이 됩니다.

예 1: 다른 스냅샷에서 참조된 데이터가 포함된 스냅샷 삭제



등록된 AMI에서 사용된 EBS 볼륨의 루트 디바이스에 대한 스냅샷을 삭제할 수 없음에 유의하십시오. 스냅샷을 삭제하기 전 우선 AMI를 등록해야 합니다. 자세한 내용은 Linux AMI 등록 취소 (p. 158) 단원을 참조하십시오.

콘솔을 이용하여 스냅샷을 삭제하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스냅샷을 선택합니다.
3. 스냅샷을 선택한 다음 작업 목록에서 삭제를 선택합니다.
4. 예, 삭제를 선택합니다.

명령줄을 이용하여 스냅샷을 삭제하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- delete-snapshot(AWS CLI)
- Remove-EC2Snapshot(Windows PowerShell용 AWS 도구)

> **Note**
>
> 진행 중인 스냅샷을 삭제할 수는 있지만, 삭제가 적용되려면 해당 스냅샷이 완전해야 합니다. 여기에는 시간이 오래 걸릴 수 있습니다. 동시 스냅샷 제한 상태(스냅샷 다섯 개를 진행 중)에서 스냅샷을 추가로 만들려고 하면 `ConcurrentSnapshotLimitExceeded` 오류를 받을 수 있습니다.

Amazon EBS 스냅샷 복사

Amazon EBS를 사용하면 볼륨의 특정 시점 스냅샷을 생성할 수 있으며 이 스냅샷은 Amazon S3에 저장됩니다. 스냅샷을 생성하고 Amazon S3로 복사를 완료한 경우(스냅샷 상태가 `completed`일 때) 이를 하나의 AWS 리전에서 다른 리전으로 또는 동일한 리전 내에서 복사할 수 있습니다. Amazon S3 서버 측 암호화(256 비트 AES)는 복사 작업 중에 전송 중인 스냅샷 데이터를 보호합니다. 스냅샷 복사본은 원본 스냅샷의 ID와 다른 ID를 받습니다.

다중 볼륨 스냅샷을 다른 AWS 리전에 복사하려면 스냅샷을 생성했을 때 다중 볼륨 스냅샷 그룹에 적용한 태그를 사용하여 스냅샷을 검색합니다. 그런 다음 스냅샷을 다른 리전에 개별적으로 복사합니다.

Amazon RDS 스냅샷 복사에 대한 자세한 내용은 Amazon RDS 사용 설명서의 DB 스냅샷 복사를 참조하십시오.

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. 938) 단원을 참조하십시오.

AWS 리전 및 계정 간의 스냅샷 복사와 관련된 요금 정보는 Amazon EBS 요금을 참조하십시오. 단일 계정 및 리전 내에서의 스냅샷 복사 작업은 스냅샷 복사본의 암호화 상태가 변경되지 않을 경우 실제로 데이터가 복사되지 않기 때문에 아무런 비용도 발생하지 않습니다.

> **Note**
>
> 새 지역으로 스냅샷을 복사하면 항상 전체(증분 아님) 복사가 생성되고 추가 지연 및 스토리지 비용이 발생합니다.
>
> **Note**
>
> 새 CMK로 스냅샷을 복사 및 암호화하면 항상 전체(증분 아님) 복사가 생성되고 추가 지연 및 스토리지 비용이 발생합니다.

사용 사례

- 지리적 확장: 새 AWS 리전에서 애플리케이션 시작.
- 마이그레이션: 새 리전으로 애플리케이션을 마이그레이션하여 가용성을 향상하고 비용을 최소화.
- 재해 복구: 서로 다른 지리적 위치에 있는 데이터와 로그를 정기적인 시간 가격으로 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 보호하는 데 도움이 됩니다.

사전 조건

- 공유 스냅샷 및 사용자가 생성한 스냅샷 등 `completed` 상태인 액세스 가능 스냅샷을 복사할 수 있습니다.
- AWS Marketplace, VM Import/Export 및 AWS Storage Gateway 스냅샷을 복사할 수 있지만 대상 리전에서 해당 스냅샷이 지원되는지 확인해야 합니다.

제한

- 각 계정에서는 단일 대상 리전으로 최대 5개의 동시 스냅샷 복사를 요청할 수 있습니다.
- 사용자 정의 태그는 원본 스냅샷에서 새로운 스냅샷으로 복사되지 않습니다. 복사 작업 도중이나 이후에 사용자 정의 태그를 추가할 수 있습니다. 자세한 내용은 Amazon EC2리소스에 태그 지정 (p. 1068) 단원을 참조하십시오.

- `CopySnapshot` 작업을 통해 생성된 스냅샷에는 어떠한 용도로도 사용할 수 없는 임의 볼륨 ID가 있습니다.

리전 간 증분 복사

스냅샷 복사가 증분 복사인지 여부는 가장 최근 완료된 스냅샷 복사에 따라 결정됩니다. 리전 간에 스냅샷을 복사할 때 다음 조건이 충족되면 복사는 증분 복사입니다.

- 이전에 스냅샷이 대상 리전에 복사되었습니다.
- 가장 최근 스냅샷 복사가 여전히 대상 리전에 있습니다.
- 대상 리전에 있는 스냅샷의 모든 복사본이 암호화되지 않거나 동일한 CMK를 사용하여 암호화되었습니다.

가장 최근 스냅샷 복사본이 삭제된 경우 다음 복사본은 전체 복사이며, 증분 복사가 아닙니다. 다른 복사를 시작할 때 첫 번째 복사가 여전히 보류 중이면 두 번째 복사도 전체 복사입니다. 첫 번째 복사가 완료되었지만 다른 복사를 시작할 때 두 번째 복사가 여전히 보류 중이면 세 번째 복사는 첫 번째 복사에 증분됩니다.

대상 리전에서 볼륨의 가장 최근 스냅샷 복사를 추적할 수 있도록 스냅샷에 볼륨 ID와 생성 시간을 표시하는 태그를 지정하는 것이 좋습니다.

스냅샷 복사가 증분인지 확인하려면 copySnapshot (p. 1022) CloudWatch 이벤트를 점검하십시오.

암호화 및 스냅샷 복사

스냅샷을 복사할 때는 복사본을 암호화하거나 원본과 다른 CMK(고객 마스터 키)를 지정할 수 있습니다. 그러면 복사된 스냅샷은 새로운 CMK를 사용합니다. 하지만 복사 작업 중 스냅샷의 암호화 상태를 변경하면 증분식이 아닌 전체 복사본이 생성되어 많은 양의 데이터가 전송되고 스토리지 요금이 많이 발생할 수 있습니다.

다른 AWS 계정에서 공유한 암호화된 스냅샷을 복사하려면 해당 스냅샷의 사용 권한과 함께 스냅샷 암호화에 사용된 고객 마스터 키(CMK) 사용 권한도 필요합니다. 자신에게 공유해 준 암호화된 스냅샷을 사용할 때는 자체 CMK로 스냅샷을 복사하여 다시 암호화하는 것이 좋습니다. 이를 통해 원본 CMK가 손상되거나 소유자가 키를 취소하는 바람에 자신이 스냅샷으로 만든 어떤 암호화된 볼륨에도 액세스하지 못하게 되는 상황을 피할 수 있습니다. 자세한 내용은 Amazon EBS 스냅샷 공유 (p. 938) 단원을 참조하십시오.

`Encrypted` 파라미터를 `true`로 설정하여 EBS 스냅샷 복사본에 암호화를 적용합니다. encryption by default(암호화 기본 제공) (p. 971)가 활성화되어 있는 경우 `Encrypted` 파라미터는 선택 사항입니다.

선택적으로 `KmsKeyId`를 사용하여 스냅샷 복사본을 암호화하는 데 사용할 사용자 지정 키를 지정할 수 있습니다. 기본 암호화가 설정되어 있어도 `Encrypted` 파라미터도 `true`로 설정해야 합니다. `KmsKeyId`가 지정되지 않은 경우 암호화에 사용되는 키는 소스 스냅샷 및 해당 소유권의 암호화 상태에 따라 달라집니다.

다음 표에서는 가능한 각 설정 조합에 대한 암호화 결과를 설명합니다.

암호화 결과: 스냅샷 복사

파라미터 **`Encrypted`** 설정 여부	암호화 기본 설정 여부	소스 스냅샷	기본값(KmsKeyId가 지정되지 않음)	사용자 지정 (KmsKeyId가 지정됨)
아니요	아니요	암호화되지 않은 소유 스냅샷	암호화되지 않음	해당 사항 없음
아니요	아니요	암호화된 소유 스냅샷	동일한 키로 암호화됨	
아니요	아니요	암호화되지 않은 공유 스냅샷	암호화되지 않음	
아니요	아니요	암호화된 공유 스냅샷	기본 CMK로 암호화됨*	

파라미터 **Encrypted** 설정 여부	암호화 기본 설정 여부	소스 스냅샷	기본값(KmsKeyId가 지정되지 않음)	사용자 지정 (KmsKeyId가 지정됨)
예	아니요	암호화되지 않은 소유 스냅샷	기본 CMK로 암호화됨	지정된 CMK로 암호화됨**
예	아니요	암호화된 소유 스냅샷	동일한 키로 암호화됨	
예	아니요	암호화되지 않은 공유 스냅샷	기본 CMK로 암호화됨	
예	아니요	암호화된 공유 스냅샷	기본 CMK로 암호화됨	
아니요	예	암호화되지 않은 소유 스냅샷	기본 CMK로 암호화됨	해당 사항 없음
아니요	예	암호화된 소유 스냅샷	동일한 키로 암호화됨	
아니요	예	암호화되지 않은 공유 스냅샷	기본 CMK로 암호화됨	
아니요	예	암호화된 공유 스냅샷	기본 CMK로 암호화됨	
예	예	암호화되지 않은 소유 스냅샷	기본 CMK로 암호화됨	지정된 CMK로 암호화됨
예	예	암호화된 소유 스냅샷	동일한 키로 암호화됨	
예	예	암호화되지 않은 공유 스냅샷	기본 CMK로 암호화됨	
예	예	암호화된 공유 스냅샷	기본 CMK로 암호화됨	

* 이는 AWS 계정 및 리전의 EBS 암호화에 사용되는 기본 CMK입니다. 기본적으로 EBS에 대한 고유한 AWS 관리형 CMK이거나 고객 관리형 CMK를 지정할 수 있습니다. 자세한 내용은 EBS 암호화에 대한 기본 키 (p. 970) 단원을 참조하십시오.

** 이는 복사 작업용으로 지정된 고객 관리형 CMK입니다. 이 CMK는 AWS 계정 및 리전에 대한 기본 CMK 대신 사용됩니다.

스냅샷 복사

다음 절차에 따라 Amazon EC2 콘솔을 사용하여 스냅샷을 복사합니다.

콘솔을 사용하여 스냅샷을 복사하려면,

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스냅샷을 선택합니다.
3. 복사할 스냅샷을 선택한 다음 작업 목록에서 복사를 선택합니다.
4. 스냅샷 복사 대화 상자에서 필요한 경우 다음을 업데이트합니다.
   - 대상 리전: 스냅샷 사본을 작성할 리전 선택.
   - 설명: 기본적으로, 설명에는 소스 스냅샷에 대한 정보가 포함되어 사용자는 원본과 사본을 구분할 수 있습니다. 필요한 경우 이 설명을 수정할 수 있습니다.
   - 암호화: 원본 스냅샷이 암호화되지 않은 경우에는 사본을 암호화할 수 있습니다. 암호화 기본 제공 (p. 971)을 활성화한 경우 암호화 옵션이 설정되어 스냅샷 콘솔로부터 설정 해제할 수 없습니다.

암호화 옵션이 설정된 경우, 아래 설명된 필드에서 하나를 선택하여 고객 관리형 CMK로 암호화하도록 선택할 수 있습니다.

암호화된 스냅샷에서 암호화를 제거할 수 없습니다.

Note

새 CMK로 스냅샷을 복사 및 암호화하면 항상 전체(증분 아님) 복사가 생성되고 추가 지연 및 스토리지 비용이 발생합니다.

- 마스터 키: 이 스냅샷을 암호화하는 데 사용할 고객 마스터 키(CMK). 계정의 기본 키가 처음에는 표시되지만 선택적으로 계정의 마스터 키를 선택하거나 다른 계정의 키 ARN을 입력/붙여넣기 할 수 있습니다. 새로운 마스터 암호화 키는 IAM 콘솔 https://console.aws.amazon.com/iam/에서 생성할 수 있습니다.

5. 복사를 선택합니다.
6. 스냅샷 복사 확인 대화 상자에서 스냅샷을 선택해 지정된 리전의 스냅샷 페이지로 이동하거나 닫기를 선택합니다.

   복사 프로세스 진행률을 확인하려면 대상 리전으로 전환한 다음, 스냅샷 페이지를 새로 고칩니다. 복사 진행률이 페이지 상단에 표시됩니다.

오류를 확인하려면

암호화 키 사용 권한 없이 암호화된 스냅샷을 복사하려고 하면 작업이 자동으로 실패합니다. 페이지를 새로 고칠 때까지는 콘솔에 오류 상태가 표시되지 않습니다. 또한 다음 예와 같이 명령줄에서 스냅샷의 상태를 확인할 수 있습니다.

```
aws ec2 describe-snapshots --snapshot-id snap-0123abcd
```

키 권한 부족으로 복사에 실패한 경우에는 "StateMessage": "Given key ID is not accessible" 메시지가 표시됩니다.

암호화된 스냅샷을 복사하려면 기본 CMK에 대한 `DescribeKey` 권한이 있어야 합니다. 이러한 권한을 명시적으로 거부하면 복사에 실패합니다. CMK 키 관리에 대한 자세한 내용은 고객 마스터 키에 대한 액세스 제어를 참조하십시오.

명령줄을 이용하여 스냅샷을 복사하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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 스냅샷을 선택합니다.
3. 목록을 줄이려면 필터 목록에서 옵션을 선택합니다. 예를 들어, 자신이 소유한 스냅샷만을 확인하려면 내 소유를 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. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- describe-snapshots(AWS CLI)
- Get-EC2Snapshot(Windows PowerShell용 AWS 도구)

Amazon EBS 스냅샷 공유

스냅샷 권한을 수정하여 본인이 지정한 AWS 계정과 공유할 수 있습니다. 허가된 사용자는 원본 스냅샷에 아무런 영향을 주지 않으면서 공유된 스냅샷을 토대로 자체 EBS 볼륨을 생성할 수 있습니다.

원한다면 암호화되지 않은 스냅샷을 모든 AWS 사용자에게 공개할 수 있습니다. 암호화된 스냅샷은 공개적으로 사용 가능하게 만들 수 없습니다.

암호화된 스냅샷을 공유할 때는 해당 스냅샷을 암호화하는 데 사용된 고객 관리형 CMK도 공유해야 합니다. 생성 당시에 또는 나중에 고객 관리형 CMK에 교차 계정 권한을 적용할 수 있습니다.

> **Important**
>
> 스냅샷을 공유하면 다른 사람들이 해당 스냅샷의 모든 데이터에 액세스할 수 있게 됩니다. 그러므로 스냅샷의 모든 데이터를 공유하려는 사용자하고만 스냅샷을 공유하십시오.

고려 사항

다음은 스냅샷을 공유할 때 고려할 사항입니다.

- 스냅샷은 생성된 리전으로 제한됩니다. 스냅샷을 다른 리전과 공유하려면 스냅샷을 해당 리전에 복사합니다. 자세한 내용은 Amazon EBS 스냅샷 복사 (p. 934) 단원을 참조하십시오.
- 스냅샷에서 더 긴 리소스 ID 형식을 사용하는 경우 더 긴 ID를 지원하는 다른 계정과 공유해야 합니다. 자세한 내용은 Resource ID (p. 1060)를 참조하십시오.
- AWS는 기본 CMK로 암호화된 스냅샷의 공유를 금지합니다. 그 대신, 고객 관리형 CMK로 공유하려는 스냅샷을 암호화해야 합니다. 자세한 내용은 AWS Key Management Service Developer Guide의 키 생성 단원을 참조하십시오.
- 공유된 CMK 사용자가 암호화된 스냅샷에 액세스하는 경우, 키에 대해 `kms:DescribeKey`, `kms:CreateGrant`, `GenerateDataKey` 및 `kms:ReEncrypt` 작업을 수행할 수 있는 권한을 부여해야 합니다. 자세한 내용은 AWS Key Management Service Developer Guide의 사용자 지정 마스터 키에 대한 액세스 제어를 참조하십시오.

콘솔을 사용하여 암호화되지 않은 스냅샷 공유

콘솔을 사용해 스냅샷을 공유하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스냅샷을 선택합니다.
3. 스냅샷을 선택한 다음 작업과 권한 수정을 선택합니다.
4. 다음과 같이 스냅샷을 퍼블릭 스냅샷으로 만들거나 특정 AWS 계정과 공유합니다.
   - 스냅샷을 퍼블릭으로 설정하려면 퍼블릭을 선택합니다.

     암호화된 스냅샷 또는 AWS Marketplace 제품 코드가 있는 스냅샷에는 이 옵션을 사용할 수 없습니다.
   - 하나 이상의 AWS 계정과 스냅샷을 공유하려면 프라이빗을 선택하고 AWS 계정 번호에 AWS 계정 ID(하이픈 제외)를 입력한 다음 권한 추가를 선택합니다. 모든 추가 AWS 계정에 대해 반복합니다.
5. Save를 선택합니다.

본인에게 개인적으로 공유해 준 암호화되지 않은 스냅샷을 사용하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스냅샷을 선택합니다.
3. 프라이빗 스냅샷 필터를 선택합니다.
4. ID 또는 설명으로 해당 스냅샷을 찾습니다. 이 스냅샷도 다른 스냅샷처럼 사용할 수 있습니다. 예를 들어, 이 스냅샷으로 볼륨을 생성하거나 스냅샷을 다른 리전으로 복사할 수 있습니다.

콘솔을 사용하여 암호화된 않은 스냅샷 공유

콘솔을 사용하여 암호화된 스냅샷을 공유하려면

1. https://console.aws.amazon.com/kms에서 AWS KMS 콘솔을 엽니다.
2. AWS 리전을 변경하려면 페이지의 오른쪽 위 모서리에 있는 리전 선택기를 사용합니다.
3. 탐색 창에서 Customer managed keys(고객 관리형 키)를 선택합니다.
4. 별칭 열에서 스냅샷을 암호화하는 데 사용한 고객 관리형 키의 별칭(텍스트 링크)을 선택합니다. 키 세부 정보가 새 페이지에서 열립니다.
5. Key policy(키 정책) 섹션에는 정책 보기 또는 기본 보기가 표시됩니다. 정책 보기에는 주요 정책 문서가 표시됩니다. 기본 보기에는 키 관리자, 키 삭제, 키 사용 및 기타 AWS 계정에 대한 섹션에 표시됩니다. 콘솔에서 정책을 생성하고 사용자 지정하지 않은 경우 기본 보기가 표시됩니다. 기본 보기를 사용할 수 없는 경우 정책 보기에서 정책을 수동으로 편집해야 합니다. 자세한 내용은 AWS Key Management Service Developer Guide의 키 정책 보기(콘솔)를 참조하십시오.

   이용 가능한 보기에 따라 정책 보기 또는 기본 보기를 사용하여 다음과 같이 하나 이상의 AWS 계정 ID를 정책에 추가합니다.

   - (정책 보기) Edit(편집)를 선택합니다. `"Allow use of the key"` 및 `"Allow attachment of persistent resources"` 명령문에 하나 이상의 AWS 계정 ID를 추가합니다. [Save changes]를 선택합니다. 다음 예에서는 AWS 계정 ID `444455556666`이 정책에 추가됩니다.

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {"AWS": [
    "arn:aws:iam::111122223333:user/CMKUser",
    "arn:aws:iam::444455556666:root"
  ]},
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*"
},
{
  "Sid": "Allow attachment of persistent resources",
  "Effect": "Allow",
  "Principal": {"AWS": [
    "arn:aws:iam::111122223333:user/CMKUser",
    "arn:aws:iam::444455556666:root"
  ]},
  "Action": [
    "kms:CreateGrant",
    "kms:ListGrants",
    "kms:RevokeGrant"
  ],
  "Resource": "*",
```

```
  "Condition": {"Bool": {"kms:GrantIsForAWSResource": true}}
}
```

- (기본 보기) 아래로 스크롤하여 Other AWS accounts(다른 AWS 계정)로 이동합니다. Add other AWS accounts(다른 AWS 계정 추가)를 선택하고 메시지가 표시되면 AWS 계정 ID를 입력합니다. 다른 계정을 추가하려면 Add another AWS account(다른 AWS 계정 추가)를 선택하고 AWS 계정 ID를 입력합니다. AWS 계정을 모두 추가했으면 변경 사항 저장을 선택합니다.

6. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
7. 탐색 창에서 스냅샷을 선택합니다.
8. 스냅샷을 선택한 다음 작업과 권한 수정을 선택합니다.
9. 각각의 AWS 계정에 대해 AWS 계정 번호에 AWS 계정 ID를 입력하고 권한 추가를 선택합니다. AWS 계정을 모두 추가했으면 저장을 선택합니다.

본인에게 공유해 준 암호화된 스냅샷을 사용하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스냅샷을 선택합니다.
3. 프라이빗 스냅샷 필터를 선택합니다. 원한다면 암호화 필터도 추가합니다.
4. ID 또는 설명으로 해당 스냅샷을 찾습니다.
5. 스냅샷을 선택하고 작업 및 복사를 선택합니다.
6. (선택 사항) 대상 리전을 선택합니다.
7. 스냅샷의 사본은 마스터 키에 표시된 키로 암호화됩니다. 기본적으로 선택한 키는 계정의 기본 CMK입니다. 고객 관리형 CMK를 선택하려면 입력 상자 안을 클릭하여 사용 가능한 키 목록을 조회합니다.
8. 복사를 선택합니다.

명령줄을 사용하여 스냅샷 공유

스냅샷에 대한 권한은 스냅샷의 `createVolumePermission` 속성을 사용하여 지정됩니다. 스냅샷을 퍼블릭으로 설정하려면 그룹을 `all`로 설정합니다. 특정 AWS 계정과 스냅샷을 공유하려면 사용자를 AWS 계정 ID로 설정합니다.

명령줄을 이용하여 스냅샷 권한을 수정하려면

다음 명령 중 하나를 사용합니다.

- modify-snapshot-attribute(AWS CLI)
- Edit-EC2SnapshotAttribute(Windows PowerShell용 AWS 도구)

명령줄을 사용하여 스냅샷 권한을 보려면

다음 명령 중 하나를 사용합니다.

- describe-snapshot-attribute(AWS CLI)
- Get-EC2SnapshotAttribute(Windows PowerShell용 AWS 도구)

명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

EBS 스냅샷 콘텐츠 액세스

Amazon Elastic Block Store(EBS) 직접 API를 사용하여 EBS 스냅샷에서 직접 데이터를 읽고 두 스냅샷의 차이를 식별할 수 있습니다. EBS 스냅샷에서 블록의 세부 정보를 보고 두 스냅샷의 블록 차이를 비교하며 스냅샷에 있는 데이터에 직접 액세스할 수 있습니다. EBS용 백업 서비스를 제공하는 Independent Software

Vendor(ISV)는 EBS direct APIs를 사용하여 EBS 스냅샷을 통해 더 쉽고 비용 효과적으로 EBS 볼륨에 대한 증분 변경 사항을 추적할 수 있습니다. EBS 스냅샷에서 새 볼륨을 만들지 않고도 이 과정이 가능하며 EC2 인스턴스를 사용하여 차이를 비교할 수 있습니다.

이 사용 설명서에는 EBS direct APIs를 구성하는 요소의 개요와 이 요소를 효과적으로 사용하는 방법의 예가 나와 있습니다. API의 작업, 데이터 형식, 파라미터 및 오류에 대한 자세한 내용은 EBS direct APIs 참조를 참조하십시오. EBS direct APIs에 대해 지원되는 AWS 리전, 엔드포인트 및 서비스 할당량은 AWS 일반 참조의 Amazon Elastic Block Store 엔드포인트 및 할당량을 참조하십시오.

목차

- EBS direct APIs 이해 (p. 941)
- IAM 사용자의 권한 (p. 942)
- 명령줄을 사용한 EBS direct APIs 작업 (p. 944)
- API 또는 AWS SDK를 사용하여 EBS direct APIs 작업 (p. 946)
- EBS direct APIs FAQ (p. 946)

EBS direct APIs 이해

EBS direct APIs를 시작하려면 먼저 다음과 같은 핵심 요소를 이해해야 합니다.

스냅샷

스냅샷은 EBS 볼륨에서 데이터를 백업하는 기본 방법입니다. 스토리지 비용을 절약하기 위해 이전 스냅샷 이후로 변경된 볼륨 데이터만 연속 스냅샷에 증분식으로 포함시킵니다. 자세한 내용은 Amazon EBS 스냅샷 (p. 926) 단원을 참조하십시오.

> **Note**
>
> EBS direct APIs에서는 퍼블릭 스냅샷이 지원되지 않습니다.

블록

블록은 스냅샷에 있는 데이터의 조각입니다. 각 스냅샷에 수천 개의 블록을 포함할 수 있습니다. 스냅샷에 있는 모든 블록은 크기가 고정되어 있습니다.

블록 인덱스

블록 인덱스는 스냅샷에 있는 블록의 오프셋 위치이며 블록을 식별하는 데 사용됩니다. 논리 블록에 있는 데이터의 논리적 오프셋을 식별하려면 `BlockIndex` 값을 `BlockSize` 값과 곱하십시오(`BlockIndex * BlockSize`).

블록 토큰

블록 토큰은 스냅샷에 있는 블록의 식별 해시이며 블록 데이터를 찾는 데 사용됩니다.

> **Note**
>
> EBS direct APIs에서 반환된 블록 토큰은 임시입니다. 같은 스냅샷에 대해 또 다른 `ListSnapshotBlocks` 또는 `ListChangedBlocks` 요청을 실행하면 블록 토큰이 변경됩니다.

스냅샷 블록 나열

`ListSnapshotBlocks` API 작업은 지정된 스냅샷에 있는 블록의 블록 인덱스와 블록 토큰을 반환합니다. 자세한 내용은 EBS direct APIs 참조의 ListSnapshotBlocks를 참조하십시오.

변경된 블록 나열

`ListChangedBlocks` API 작업은 볼륨/스냅샷 계보가 같은 2개의 지정된 스냅샷에서 차이가 있는 블록의 블록 인덱스와 블록 토큰을 반환합니다. 자세한 내용은 EBS direct APIs 참조의 ListChangedBlocks를 참조하십시오.

스냅샷 블록 가져오기

`GetSnapshotBlock` API 작업은 지정된 스냅샷 ID, 블록 인덱스 및 블록 토큰의 블록 내 데이터를 반환합니다. 자세한 내용은 EBS direct APIs 참조의 GetSnapshotBlock을 참조하십시오.

API 사용

`ListSnapshotBlocks` 또는 `ListChangedBlocks` API 작업을 사용하여 데이터를 가져올 블록의 블록 인덱스와 블록 토큰을 식별할 수 있습니다. 그런 다음 `GetSnapshotBlock` API 작업을 사용하여 스냅샷의 블록에서 데이터를 가져옵니다. AWS CLI를 사용하여 이 작업을 실행하는 방법의 예가 이 설명서 뒷부분에 나옵니다.

IAM 사용자의 권한

IAM 사용자는 다음 정책이 있어야 EBS direct APIs를 사용할 수 있습니다.

> **Important**
>
> IAM 사용자에게 다음 정책을 할당할 때는 주의해야 합니다. 이 정책을 할당하면 CopySnapshot 또는 CreateVolume 작업과 같은 EC2 API를 통해 동일한 리소스에 액세스하는 것이 거부된 사용자에게 액세스 권한을 부여할 수 있습니다.

다음 정책은 EBS direct APIs에 대한 모든 액세스 권한을 부여합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ebs:*"
            ],
            "Resource": "arn:aws:ec2:*::snapshot/*"
        }
    ]
}
```

다음 정책은 특정 AWS 리전에 있는 특정 스냅샷에 대한 액세스 권한을 부여합니다. 정책에서 *<SnapshotID>*를 스냅샷 ID로 바꾸고 *<Region>*을 스냅샷의 리전으로 바꾸십시오.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ebs:ListSnapshotBlocks"
            ],
            "Resource": "arn:aws:ec2:<Region>::snapshot/<SnapshotID>"
        }
    ]
}
```

다음 정책은 특정 키/값 태그가 있는 스냅샷에 EBS direct APIs의 액세스 권한을 부여합니다. 정책에서 *<Key>*를 태그의 키 값으로 바꾸고 *<Value>*를 태그 값으로 바꾸십시오.

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

```
            "Effect": "Allow",
            "Action": [
                "ebs:ListChangedBlocks",
                "ebs:ListSnapshotBlocks",
                "ebs:GetSnapshotBlock"
            ],
            "Resource": "arn:aws:ec2:*::snapshot/*",
            "Condition": {
                "StringEqualsIgnoreCase": {
                    "aws:ResourceTag/<Key>": "<Value>"
                }
            }
        }
    ]
}
```

다음 권한은 특정 스냅샷에 대한 EBS direct APIs의 액세스를 거부합니다. 정책에서 *<SnapshotID>*를 스냅샷 ID로 바꾸십시오.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ebs:ListSnapshotBlocks",
                "ebs:ListChangedBlocks",
                "ebs:GetSnapshotBlock"
            ],
            "Resource": "arn:aws:ec2:*::snapshot/*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "ebs:ListSnapshotBlocks",
                "ebs:ListChangedBlocks",
                "ebs:GetSnapshotBlock"
            ],
            "Resource": "arn:aws:ec2:us-east-2::<SnapshotID>"
        }
    ]
}
```

다음 정책은 특정 시간 범위에 모든 스냅샷에 대한 액세스 권한을 부여합니다. 정책에서 표시된 날짜 및 시간 범위를 정책의 날짜 및 시간 범위로 바꿔야 합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ebs:ListChangedBlocks",
                "ebs:ListSnapshotBlocks",
                "ebs:GetSnapshotBlock"
            ],
            "Resource": "arn:aws:ec2:*::snapshot/*",
            "Condition": {
                "DateGreaterThan": {
                    "aws:CurrentTime": "2018-05-29T00:00:00Z"
                },
                "DateLessThan": {
                    "aws:CurrentTime": "2020-05-29T23:59:59Z"
```

```
                }
            }
        }
    ]
}
```

다음 정책은 AWS Key Management Service(AWS KMS)에서 특정 키 ID를 사용하여 암호화된 스냅샷을 해독할 수 있는 액세스 권한을 부여합니다. 정책에서 *<AccountId>*를 AWS KMS 키의 AWS 계정 ID로 바꾸고 *<KeyId>*를 EBS direct APIs로 액세스할 스냅샷을 암호화하는 데 사용되는 키의 ID로 바꾸십시오.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:kms:us-west-2:<AccountId>:key/<KeyId>"
        }
    ]
}
```

자세한 내용은 IAM 사용 설명서의 IAM 사용자의 권한 변경을 참조하십시오.

명령줄을 사용한 EBS direct APIs 작업

다음 예제에서는 AWS Command Line Interface(AWS CLI)를 통해 EBS direct APIs를 이용하는 방법을 보여줍니다. AWS CLI 설치 및 구성에 대한 자세한 내용은 AWS CLI 버전 1 설치 및 빠르게 AWS CLI 구성을 참조하십시오.

Example 예: 스냅샷에 있는 블록의 블록 인덱스 및 블록 토큰 가져오기

다음 `list-snapshot-blocks` 명령 예제는 **us-east-1** AWS 리전의 **snap-0987654321** 스냅샷에 있는 블록의 블록 인덱스 및 블록 토큰을 반환합니다. `--starting-block-index` 및 `--max-results` 파라미터는 블록 인덱스가 **1000**보다 큰 처음 **100**개 블록으로 결과를 제한합니다.

```
aws ebs list-snapshot-blocks --region us-east-1 --snapshot-id snap-0987654321 --starting-
block-index 1000 --max-results 100
```

다음은 응답의 예입니다. 블록에 있는 데이터를 가져오려면 `get-snapshot-block` 명령을 사용하고 블록의 블록 인덱스와 블록 토큰을 지정하십시오. 블록 토큰은 기재된 만료 시간까지 유효합니다.

```
{
    "Blocks": [
        {
            "BlockIndex": 1001,
            "BlockToken": "AAABAV3/PNhXOynVdMYHUpPsetaSvjLB1dtIGfbJv5OJ0sX855EzGTWos4a4"
        },
        {
            "BlockIndex": 1002,
            "BlockToken": "AAABATGQIgwr0WwIuqIMjCA/Sy7e/YoQFZsHejzGNvjKauzNgzeI13YHBfQB"
        },
        {
            "BlockIndex": 1007,
            "BlockToken": "AAABAZ9CTuQtUvp/dXqRWw4d07eOgTZ3jvn6hiW30W9duM8MiMw6yQayzF2c"
        },
        {
```

```
            "BlockIndex": 1012,
            "BlockToken": "AAABAQdzxhw0rVV6PNmsfo/YRIxo9JPR85XxPf1BLjg0Hec6pygYr6laE1p0"
        },
        {
            "BlockIndex": 1030,
            "BlockToken": "AAABAaYvPax6mv+iGWLdTUjQtFWouQ7Dqz6nSD9L+CbXnvpkswA6iDID523d"
        },
        {
            "BlockIndex": 1031,
            "BlockToken": "AAABATgWZC0XcFwUKvTJbUXMiSPg59KVxJGL+BWBClkw6spzCxJVqDVaTskJ"
        },
        ...
    ],
    "ExpiryTime": 1576287332.806,
    "VolumeSize": 32212254720,
    "BlockSize": 524288
}
```

Example 예: 볼륨/스냅샷 계보가 동일한 두 스냅샷에서 차이가 있는 블록의 블록 인덱스 및 블록 토큰 가져오기

다음 `list-changed-blocks` 명령 예제는 **us-east-1** AWS 리전의 **snap-1234567890** 스냅샷과 **snap-0987654321** 스냅샷에서 차이가 있는 블록의 블록 인덱스 및 블록 토큰을 반환합니다. `--starting-block-index` 및 `--max-results` 파라미터는 블록 인덱스가 **0**보다 큰 처음 **500**개 블록으로 결과를 제한합니다.

```
aws ebs list-changed-blocks --region us-east-1 --first-snapshot-id snap-1234567890 --
second-snapshot-id snap-0987654321 --starting-block-index 0 --max-results 500
```

다음은 응답의 예입니다. 여기서는 블록 인덱스 0, 6000, 6001, 6002, 6003이 두 스냅샷에서 차이가 있습니다. 또한 응답에 기재된 두 번째 블록 토큰이 없으므로 지정된 첫 번째 스냅샷 ID에만 블록 인덱스 6001, 6002, 6003이 있고 두 번째 스냅샷 ID에는 없습니다.

블록에 있는 데이터를 가져오려면 `get-snapshot-block` 명령을 사용하고 블록의 블록 인덱스와 블록 토큰을 지정하십시오. 블록 토큰은 기재된 만료 시간까지 유효합니다.

```
{
    "ChangedBlocks": [
        {
            "BlockIndex": 0,
            "FirstBlockToken": "AAABAVahm9SO60Dyi0ORySzn2ZjGjW/
KN3uygGlS0QOYWesbzBbDnX2dGpmC",
            "SecondBlockToken":
 "AAABAf8o0o6UFi1rDbSZGIRaCEdDyBu9TlvtCQxxoKV8qrUPQP7vcM6iWGSr"
        },
        {
            "BlockIndex": 6000,
            "FirstBlockToken": "AAABAbYSiZvJ0/
R9tz8suI8dSzecLjN4kkazK8inFXVintPkdaVFLfCMQsKe",
            "SecondBlockToken":
 "AAABAZnqTdzFmKRpsaMAsDxviVqEI/3jJzI2crq2eFDCgHmyNf777elD9oVR"
        },
        {
            "BlockIndex": 6001,
            "FirstBlockToken": "AAABASBpSJ2UAD3PLxJnCt6zun4/
T4sU25Bnb8jB5Q6FRXHFqAIAqE04hJoR"
        },
        {
            "BlockIndex": 6002,
            "FirstBlockToken": "AAABASqX4/
NWjvNceoyMUljcRd0DnwbSwNnes1UkoP62CrQXvn47BY5435aw"
        },
```

```
        {
            "BlockIndex": 6003,
            "FirstBlockToken":
 "AAABASmJ0O5JxAOce25rF4P1sdRtyIDsX12tFEDunnePYUKOf4PBROuICb2A"
        },
        ...
    ],
    "ExpiryTime": 1576308931.973,
    "VolumeSize": 32212254720,
    "BlockSize": 524288,
    "NextToken": "AAADARqElNng/sV98CYk/bJDCXeLJmLJHnNSkHvLzVaO0zsPH/QM3Bi3zF//O6Mdi/
BbJarBnp8h"
}
```

Example 예: 블록의 데이터 가져 오기

다음 `get-snapshot-block` 명령 예제는 **`us-east-1`** AWS 리전의 **`snap-1234567890`** 스냅샷에서 블록 토큰이 **`AAABASBpSJ2UAD3PLxJnCt6zun4/T4sU25Bnb8jB5Q6FRXHFqAIAqE04hJoR`**인 블록 인덱스 **`6001`**의 데이터를 반환합니다. Windows 컴퓨터에서 **`C:\Temp`** 디렉터리의 `output.txt` 파일로 이진 데이터가 출력됩니다. Linux 또는 Unix 컴퓨터에서 명령을 실행할 때는 출력 경로를 **`/tmp/output.txt`**로 바꾸어 **`/tmp`** 디렉터리의 `output.txt` 파일로 데이터를 출력하십시오.

```
aws ebs get-snapshot-block --region us-east-1 --snapshot-id snap-1234567890 --block-
index 6001 --block-token AAABASBpSJ2UAD3PLxJnCt6zun4/T4sU25Bnb8jB5Q6FRXHFqAIAqE04hJoR C:/
Temp/output.txt
```

다음은 응답의 예입니다. 여기서는 반환된 데이터의 크기, 데이터의 유효성을 검사하기 위한 체크섬 및 체크섬을 생성하는 데 사용되는 체크섬 알고리즘을 보여줍니다. 요청 명령에서 지정한 디렉터리와 파일에 이진 데이터가 자동으로 저장됩니다.

```
{
    "DataLength": "524288",
    "Checksum": "cf0Y6/Fn0oFa4VyjQPOa/iD0zhTflPTKzxGv2OKowXc=",
    "ChecksumAlgorithm": "SHA256"
}
```

API 또는 AWS SDK를 사용하여 EBS direct APIs 작업

EBS direct APIs 참조에는 각 서비스의 작업과 데이터 형식에 맞는 설명과 구문이 나와 있습니다. AWS SDK 중 하나를 통해 사용 중인 프로그래밍 언어나 플랫폼에 맞는 API에 액세스할 수도 있습니다. 자세한 정보는 AWS SDK를 참조하십시오.

EBS direct APIs에는 AWS 서명 버전 4 서명이 필요합니다. 이 서명 생성에 대한 자세한 내용은 AWS 일반 참조의 서명 버전 4 서명 프로세스를 참조하십시오.

HTTP 요청을 수동으로 생성하려는 경우에만 HTTP 요청에 서명하는 방법을 알면 됩니다. AWS 명령줄 인터페이스(AWS CLI) 또는 AWS SDK 중 하나를 이용해 AWS에 요청할 경우 도구를 구성할 때 지정한 액세스 키를 사용하여 요청에 자동으로 서명됩니다. 따라서 이러한 도구를 사용할 경우 요청에 서명하는 방법을 알 필요가 없습니다.

EBS direct APIs FAQ

보류 중 상태일 때 EBS direct APIs를 사용하여 스냅샷에 액세스할 수 있습니까?

아니요. 완료됨 상태인 경우에만 스냅샷에 액세스할 수 있습니다.

EBS direct APIs에서 번호순으로 블록 인덱스를 반환합니까?

예. 반환되는 블록 인덱스는 고유하며 번호순입니다.

`MaxResults` 파라미터 값이 100 미만인 요청을 제출할 수 있습니까?

아니요. 사용할 수 있는 최소 `MaxResult` 파라미터 값은 100입니다. `MaxResult` 파라미터 값이 100 미만인 요청을 제출할 경우 스냅샷에 있는 블록이 100개를 초과하면 API가 최소한 100개의 결과를 반환합니다.

API 요청을 동시에 실행할 수 있습니까?

API 요청을 동시에 실행할 수 있습니다. 병목 현상을 방지하려면 계정에서 다른 워크로드가 실행 중일 수 있다는 사실을 감안해야 합니다. 또한 조절, 시간 초과 및 서비스 사용 불가를 처리하려면 EBS direct APIs 워크플로우에 재시도 메커니즘을 빌드해야 합니다.

`ListChangedBlocks` 작업을 실행할 때 스냅샷에 블록이 있어도 빈 응답을 받을 수 있습니까?

예. 스냅샷에 변경된 블록이 거의 없으면 응답이 비어 있을 수 있지만 API가 다음 페이지 토큰 값을 반환합니다. 다음 페이지 토큰 값을 사용하여 결과의 다음 페이지로 계속 진행하십시오. API가 다음 페이지 토큰 값 null을 반환하면 마지막 결과 페이지에 도달했음을 확인할 수 있습니다.

`NextToken` 파라미터가 **`StartingBlockIndex`** 파라미터와 함께 지정되면 둘 중 어느 것이 사용됩니까?

`NextToken`이 사용되고 `StartingBlockIndex`가 무시됩니다.

블록 토큰과 다음 토큰은 얼마 동안 유효합니까?

블록 토큰은 7일간 유효하고 다음 토큰은 60분간 유효합니다.

암호화된 스냅샷이 지원됩니까?

예. EBS 직접 API를 사용하여 암호화된 스냅샷에 액세스할 수 있습니다.

암호화된 스냅샷에 액세스하려면 스냅샷 암호화에 사용된 키와 AWS KMS 암호 해독 작업에 대한 액세스 권한이 사용자에게 있어야 합니다. 사용자에게 할당할 AWS KMS 정책에 대해서는 이 설명서 앞부분의

> `ListSnapshotBlocks` 또는 `ListChangedBlocks` API 작업을 사용하여 데이터를 가져올 블록의 블록 인덱스와 블록 토큰을 식별할 수 있습니다. 그런 다음 `GetSnapshotBlock` API 작업을 사용하여 스냅샷의 블록에서 데이터를 가져옵니다. AWS CLI를 사용하여 이 작업을 실행하는 방법의 예가 이 설명서 뒷부분에 나옵니다.

(p.) 단원을 참조하십시오.

퍼블릭 스냅샷이 지원됩니까?

아니요. 퍼블릭 스냅샷이 지원되지 않습니다.

목록 스냅샷 블록은 스냅샷의 모든 블록 인덱스와 블록 토큰을 반환합니까? 아니면 데이터가 쓰여진 블록 인덱스와 블록 토큰만 반환합니까?

데이터가 쓰여진 블록 인덱스와 토큰만 반환합니다.

Amazon EBS 스냅샷 수명 주기 자동화

Amazon EBS 볼륨을 백업하기 위해 만든 스냅샷의 생성, 보관, 삭제를 Amazon 데이터 수명 주기 관리자로 자동화할 수 있습니다. 스냅샷 관리를 자동화하여 다음과 같은 이점을 누려 보십시오.

- 정기적인 백업 일정을 실행하여 중요한 데이터를 보호합니다.
- 감사 기관이나 내부 규정 준수 부서에서 요구하는 백업을 보관합니다.
- 오래된 백업을 삭제하여 스토리지 비용을 절감합니다.

Amazon CloudWatch Events 및 AWS CloudTrail의 모니터링 기능을 Amazon 데이터 수명 주기 관리자와 조합하면 추가 비용 없이 EBS 볼륨의 완벽한 백업 솔루션을 얻을 수 있습니다.

목차

- Amazon 데이터 수명 주기 관리자 작동 방식 (p. 948)
- Amazon 데이터 수명 주기 관리자에 대한 고려 사항 (p. 949)
- 사전 조건 (p. 950)
- 콘솔을 사용하여 백업 관리 (p. 951)
- AWS CLI를 사용하여 백업 관리 (p. 953)
- API를 사용하여 백업 관리 (p. 956)
- 스냅샷 수명 주기 모니터링 (p. 956)

Amazon 데이터 수명 주기 관리자 작동 방식

다음은 Amazon 데이터 수명 주기 관리자의 핵심 요소입니다.

요소

- 스냅샷 (p. 948)
- 대상 리소스 태그 (p. 948)
- 스냅샷 태그 (p. 948)
- 수명 주기 정책 (p. 949)

스냅샷

스냅샷은 EBS 볼륨에서 데이터를 백업하는 기본 방법입니다. 스토리지 비용을 절약하기 위해 이전 스냅샷 이후로 변경된 볼륨 데이터만 연속 스냅샷에 증분식으로 포함시킵니다. 특정 볼륨의 스냅샷 시리즈에서 스냅샷 하나를 삭제하면 해당 스냅샷에 있는 데이터만 제거됩니다. 캡처된 볼륨 기록의 나머지는 보존됩니다.

자세한 내용은 Amazon EBS 스냅샷 (p. 926) 단원을 참조하십시오.

대상 리소스 태그

Amazon 데이터 수명 주기 관리자는 백업할 EBS 볼륨을 리소스 태그로 파악합니다. 태그란 사용자가 AWS 리소스(EBS 볼륨 및 스냅샷 포함)에 할당할 수 있는 사용자 지정 가능한 메타데이터입니다. (아래에 설명된) Amazon 데이터 수명 주기 관리자 정책은 단일 태그를 사용하여 백업용 볼륨을 대상으로 합니다. 여러 정책을 실행하려면 복수의 태그를 볼륨에 할당할 수 있습니다.

'\' 또는 '=' 문자는 태그 키에 사용할 수 없습니다.

자세한 내용은 Amazon EC2리소스에 태그 지정 (p. 1068) 단원을 참조하십시오.

스냅샷 태그

Amazon 데이터 수명 주기 관리자는 정책에 따라 생성된 모든 스냅샷에 다음 태그를 적용하여 다른 방법으로 생성된 스냅샷과 구분합니다.

- `aws:dlm:lifecycle-policy-id`
- `aws:dlm:lifecycle-schedule-name`

스냅샷을 생성할 때 사용자 지정 태그가 적용되도록 지정할 수도 있습니다.

'\' 또는 '=' 문자는 태그 키에 사용할 수 없습니다.

볼륨을 정책과 연관시키기 위해 Amazon 데이터 수명 주기 관리자이 사용하는 대상 태그는 정책에 의해 생성된 스냅샷에 선택적으로 적용될 수 있습니다.

수명 주기 정책

수명 주기 정책은 다음의 핵심 설정으로 이루어집니다.

- 리소스 유형—정책에 따라 관리하는 AWS 리소스의 유형입니다. `VOLUME`을 사용하여 개별 볼륨의 스냅샷을 생성하거나 `INSTANCE`를 사용하여 인스턴스의 볼륨에서 다중 볼륨 스냅샷을 생성합니다. 자세한 내용은 다중 볼륨 스냅샷 (p. 930) 단원을 참조하십시오.
- 대상 태그—정책에 따라 관리할 수 있도록 EBS 볼륨 또는 EC2 인스턴스와 연결해야 하는 태그입니다.
- 일정—스냅샷을 생성하기 위한 시작 시간과 간격입니다.
- 보존 — 총 스냅샷 수 또는 각 스냅샷 수명에 따라 스냅샷을 보존할 수 있습니다.

예를 들어 태그 계정이 `account=Finance`인 모든 EBS 볼륨을 관리하는 정책을 만들고, 24시간마다 0900에 스냅샷을 생성하여 최근 스냅샷 다섯 개를 보관할 수 있습니다. 늦어도 0959에는 스냅샷 생성이 시작됩니다.

Amazon 데이터 수명 주기 관리자에 대한 고려 사항

AWS 계정에는 Amazon 데이터 수명 주기 관리자와 관련하여 다음과 같은 할당량이 있습니다.

- 리전마다 최대 100개의 수명 주기 정책을 생성할 수 있습니다.
- 리소스당 최대 45개의 태그를 추가할 수 있습니다.
- 수명 주기 정책당 하나의 일정을 생성할 수 있습니다.

다음은 수명 주기 정책에서 고려할 사항입니다.

- 정책의 활성화 상태를 활성으로 설정해야 정책이 스냅샷을 생성하기 시작합니다. 생성과 동시에 활성화되도록 정책을 구성할 수 있습니다.
- 지정된 시작 시간 후 한 시간 이내에 정책에 따라 첫 번째 스냅샷이 생성됩니다.
- 대상 태그를 변경하거나 삭제하여 정책을 수정하면 그러한 태그가 있는 EBS 볼륨에는 더 이상 해당 정책이 적용되지 않습니다.
- 정책의 일정 이름을 수정하면 예전의 일정 이름으로 생성된 스냅샷에는 더 이상 해당 정책이 적용되지 않습니다.
- 새 시간 간격을 사용하도록 시간을 기준으로 하는 보존 일정을 수정하면 새 간격은 새 스냅샷에만 사용됩니다. 새로운 일정은 이 정책에 따라 생성된 기존 스냅샷의 보존 일정에 영향을 미치지 않습니다.
- 스냅샷 수에서 각 스냅샷 수명으로 정책의 보존 일정을 변경할 수 없습니다. 이렇게 변경하려면 새 정책을 생성해야 합니다.
- 보존 일정이 각 스냅샷의 수명을 기준으로 하는 정책을 비활성화하면, 정책이 비활성화된 동안 보존 기간이 만료되는 스냅샷이 무기한 보존됩니다. 이러한 스냅샷은 수동으로 삭제해야 합니다. 정책을 다시 활성화하면 보존 기간이 만료될 경우 Amazon 데이터 수명 주기 관리자가 스냅샷 삭제를 재개합니다.
- 정책이 적용되는 리소스를 삭제하면 정책은 이전에 생성한 스냅샷을 더 이상 관리하지 않습니다. 더 이상 필요하지 않은 스냅샷은 수동으로 삭제해야 합니다.
- EBS 볼륨 또는 EC2 인스턴스를 백업하도록 여러 정책을 생성할 수 있습니다. EBS 볼륨에 12시간마다 스냅샷을 만드는 정책 A의 대상 태그인 태그 A와 24시간마다 스냅샷을 만드는 정책 B의 대상 태그인 태그 B, 이렇게 두 개의 태그가 있다면 Amazon 데이터 수명 주기 관리자는 양쪽 정책의 일정에 따라 스냅샷을 생성합니다.

다음은 수명 주기 정책과 빠른 스냅샷 복원 (p. 978)에 적용되는 고려 사항입니다.

- 빠른 스냅샷 복원에 대해 활성화된 스냅샷은 수명 주기 정책을 삭제 또는 비활성화하거나, 수명 주기 정책에 대한 빠른 스냅샷 복원을 비활성화하거나, 가용 영역에 대한 빠른 스냅샷 복원을 비활성화하더라도 활성화된 상태로 유지됩니다. 이러한 스냅샷에 대한 빠른 스냅샷 복원을 수동으로 비활성화할 수 있습니다.

- 빠른 스냅샷 복원을 활성화하고 빠른 스냅샷 복원에 대해 활성화할 수 있는 최대 스냅샷 수를 초과하는 경우 Amazon 데이터 수명 주기 관리자는 예약된 대로 스냅샷을 생성하지만 빠른 스냅샷 복원에 대해 스냅샷을 활성화하지 않습니다. 빠른 스냅샷 복원에 대해 활성화된 스냅샷을 삭제한 후 Amazon 데이터 수명 주기 관리자가 생성하는 다음 스냅샷은 빠른 스냅샷 복원에 대해 활성화됩니다.
- 스냅샷에 대해 빠른 스냅샷 복원을 활성화할 때 스냅샷을 최적화하려면 TiB당 60분 걸립니다. Amazon 데이터 수명 주기 관리자에서 다음 스냅샷을 생성하기 전에 각 스냅샷이 완전히 최적화되도록 일정을 생성하는 것이 좋습니다.

수명 주기 정책 및 다중 연결 (p. 911) 지원 볼륨에 적용되는 고려 사항은 다음과 같습니다.

- 다중 볼륨 스냅샷의 인스턴스 태그를 기반으로 수명 주기 정책을 생성할 때 Amazon 데이터 수명 주기 관리자에서는 연결된 각 인스턴스에 대해 볼륨의 스냅샷을 시작합니다. 타임스탬프 태그를 사용하여 연결된 인스턴스에서 생성된 시간 일관성 있는 스냅샷 집합을 식별합니다.

사전 조건

Amazon 데이터 수명 주기 관리자에는 다음과 같은 사전 조건이 필요합니다.

사전 조건

- Amazon 데이터 수명 주기 관리자 권한 (p. 950)
- IAM 사용자의 권한 (p. 951)

Amazon 데이터 수명 주기 관리자 권한

Amazon 데이터 수명 주기 관리자는 스냅샷을 관리하는 데 필요한 권한을 얻기 위해 IAM 역할을 사용합니다. AWS Management 콘솔을 사용하여 수명 주기 정책을 처음 생성하면 Amazon 데이터 수명 주기 관리자가 AWSDataLifecycleManagerDefaultRole 역할을 생성합니다. 다음 create-default-role 명령으로 이 역할을 생성할 수도 있습니다.

```
aws dlm create-default-role
```

또는 필요한 권한이 있는 사용자 지정 IAM 역할을 만든 다음, 수명 주기 정책을 생성할 때 이를 선택하는 방법도 있습니다.

사용자 지정 IAM 역할을 생성하려면

1. 다음 권한의 역할을 만듭니다.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "ec2:CreateSnapshot",
            "ec2:CreateSnapshots",
            "ec2:DeleteSnapshot",
            "ec2:DescribeVolumes",
            "ec2:DescribeInstances",
            "ec2:DescribeSnapshots"
         ],
         "Resource": "*"
      },
      {
         "Effect": "Allow",
         "Action": [
```

```
            "ec2:CreateTags"
         ],
         "Resource": "arn:aws:ec2:*::snapshot/*"
      }
   ]
}
```

자세한 정보는 IAM 사용 설명서의 역할 생성 단원을 참조하십시오.

2. 역할에 신뢰 관계를 추가합니다.

   a. IAM 콘솔에서 역할을 선택합니다.
   b. 생성된 역할을 선택하고 신뢰 관계를 선택합니다.
   c. 신뢰 관계 편집을 선택하고 다음 정책을 추가한 뒤 신뢰 정책 업데이트를 선택합니다.

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

IAM 사용자의 권한

IAM 사용자는 다음과 같은 Amazon 데이터 수명 주기 관리자 사용 권한을 가지고 있어야 합니다.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": ["iam:PassRole", "iam:ListRoles"],
         "Resource": "arn:aws:iam::123456789012:role/AWSDataLifecycleManagerDefaultRole"
      },
      {
         "Effect": "Allow",
         "Action": "dlm:*",
         "Resource": "*"
      }
   ]
}
```

자세한 정보는 IAM 사용 설명서에서 IAM 사용자의 권한 변경을 참조하십시오.

콘솔을 사용하여 백업 관리

다음 예제에서는 Amazon 데이터 수명 주기 관리자에서 AWS Management 콘솔을 사용하여 EBS 볼륨의 백업을 관리하는 방법을 보여 줍니다.

작업

- 수명 주기 정책 생성 (p. 952)
- 수명 주기 정책 보기 (p. 952)
- 수명 주기 정책 수정 (p. 953)

- 수명 주기 정책 삭제 (p. 953)

수명 주기 정책 생성

다음 절차에 따라 수명 주기 정책을 생성합니다.

수명 주기 정책 생성 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Elastic Block Store, Lifecycle Manager(수명 주기 관리자)를 선택한 다음, Create snapshot lifecycle policy(스냅샷 수명 주기 정책 생성)를 선택합니다.
3. 필요에 따라 정책에 대한 다음 정보를 제공합니다.
   - 설명 – 정책 설명입니다.
   - 리소스 유형 – 백업할 리소스의 유형입니다. `VOLUME`을 사용하여 개별 볼륨의 스냅샷을 생성하거나 `INSTANCE`를 사용하여 인스턴스의 볼륨에서 다중 볼륨 스냅샷을 생성합니다.
   - 이러한 태그가 있는 대상 – 백업할 볼륨 또는 인스턴스를 식별하는 리소스 태그입니다.
   - 수명 주기 정책 태그 – 수명 주기 정책에 대한 태그입니다.
   - 일정 이름 – 일정의 이름입니다.
   - n시간마다 정책 실행 – 정책 실행 간의 시간 간격입니다(시간). 지원되는 값은 1, 2, 3, 4, 6, 8, 12, 24입니다.
   - hh:mm UTC에 시작 – 정책 실행이 시작하도록 예약된 시간입니다. 첫 번째 정책 실행은 예약된 시간 후 한 시간 이내에 시작됩니다.
   - Retain(보존) – 총 스냅샷 수 또는 각 스냅샷 수명에 따라 스냅샷을 보존할 수 있습니다. 개수를 기준으로 보존하는 경우 범위는 1-1000입니다. 최대 수에 도달한 후에는 새 스냅샷을 생성할 때 가장 오래된 스냅샷이 삭제됩니다. 수명을 기반으로 보존하는 경우 범위는 1일-100년입니다. 각 스냅샷의 보존 기간이 만료되면 스냅샷이 삭제됩니다. 보존 기간은 생성 간격보다 크거나 같아야 합니다.
   - 리전 간 복사–최대 3개의 추가 리전으로 각 스냅샷을 복사할 수 있습니다. 리전마다 다른 보존 정책을 선택하고 태그를 모두 복사할지 또는 복사하지 않을지 선택할 수 있습니다. 소스 스냅샷이 암호화되거나 암호화가 기본적으로 활성화된 경우 스냅샷 복사가 암호화됩니다. 소스 스냅샷이 암호화 해제된 경우 암호화를 활성화할 수 있습니다. CMK를 지정하지 않으면 스냅샷은 각 대상 리전에서 EBS 암호화의 기본 키를 사용하여 암호화됩니다. 리전당 동시 스냅샷 복사본 수를 초과하지 않도록 해야 합니다.
   - 정보 태그 지정–소스 볼륨에 있는 모든 사용자 정의 태그를 이 정책에 따라 생성된 스냅샷으로 복사할지 여부를 선택합니다. 또한 Amazon 데이터 수명 주기 관리자에서 적용되는 태그 외에도 스냅샷의 추가 태그를 지정할 수 있습니다. 리소스 유형이 `INSTANCE`인스턴스인 경우 `instance-id` 및 `timestamp`와 같은 변수 태그를 사용하여 자동으로 스냅샷에 태그를 지정하도록 선택할 수 있습니다. 변수 태그의 값은 태그가 추가될 때 결정됩니다.
   - 빠른 스냅샷 복원–빠른 스냅샷 복원을 활성화할지 여부와 어떤 가용 영역에서 활성화할지를 선택합니다. 빠른 스냅샷 복원에 대해 활성화할 수 있는 최대 스냅샷 수도 지정할 수 있습니다.
   - IAM 역할–스냅샷을 생성, 삭제 및 설명하고 볼륨을 설명할 권한이 있는 IAM 역할입니다. AWS가 기본 역할 AWSDataLifecycleManagerDefaultRole을 제공하거나, 고객이 사용자 지정 IAM 역할을 만듭니다.
   - 생성 후 정책 상태–다음 예약 시간에 정책 실행을 시작하려면 정책 활성화를 선택하고, 정책을 실행하지 않으려면 정책 비활성화를 선택합니다.
4. 정책 생성을 선택합니다.

수명 주기 정책 보기

다음 절차에 따라 수명 주기 정책을 봅니다.

수명 주기 정책을 보려면

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

2. 탐색 창에서 Elastic Block Store, Lifecycle Manager(수명 주기 관리자)를 선택합니다.
3. 목록에서 수명 주기 정책을 선택합니다. 세부 정보 탭에는 정책에 대한 정보가 표시됩니다.

수명 주기 정책 수정

다음 절차에 따라 수명 주기 정책을 수정합니다.

수명 주기 정책을 수정하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Elastic Block Store, Lifecycle Manager(수명 주기 관리자)를 선택합니다.
3. 목록에서 수명 주기 정책을 선택합니다.
4. 작업, 스냅샷 수명 주기 정책 수정을 선택합니다.
5. 필요에 따라 정책 설정을 수정합니다. 예를 들어 일정을 수정하거나, 태그를 추가 또는 제거하거나, 정책을 활성화 또는 비활성화할 수 있습니다.
6. Update policy를 선택합니다.

수명 주기 정책 삭제

다음 절차에 따라 수명 주기 정책을 삭제합니다.

수명 주기 정책을 삭제하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Elastic Block Store, Lifecycle Manager(수명 주기 관리자)를 선택합니다.
3. 목록에서 수명 주기 정책을 선택합니다.
4. 작업, 스냅샷 수명 주기 정책 삭제를 선택합니다.
5. 확인 메시지가 나타나면 스냅샷 수명 주기 정책 삭제를 선택합니다.

AWS CLI를 사용하여 백업 관리

다음 예에서는 Amazon 데이터 수명 주기 관리자에서 AWS CLI를 사용하여 EBS 볼륨의 백업을 관리하는 방법을 보여 줍니다.

예제

- 수명 주기 정책 생성 (p. 953)
- 수명 주기 정책 표시 (p. 954)
- 수명 주기 정책 수정 (p. 955)
- 수명 주기 정책 삭제 (p. 956)

수명 주기 정책 생성

create-lifecycle-policy 명령을 사용하여 수명 주기 정책을 생성합니다. 구문을 간소화하기 위해 이 예제에서는 정책 세부 정보가 포함된 JSON 파일(`policyDetails.json`)을 참조합니다.

이 예제에서는 리소스 유형 `VOLUME`을 사용하여 지정된 대상 태그가 있는 모든 볼륨의 스냅샷을 생성합니다. 지정된 대상 태그가 있는 모든 인스턴스에 대한 모든 볼륨의 스냅샷을 생성하려면 리소스 유형 `INSTANCE`를 대신 사용합니다.

```
aws dlm create-lifecycle-policy --description "My volume policy" --state ENABLED --
execution-role-arn arn:aws:iam::12345678910:role/AWSDataLifecycleManagerDefaultRole --
policy-details file://policyDetails.json
```

다음은 `policyDetails.json` 파일의 예입니다.

```
{
   "ResourceTypes": [
      "VOLUME"
   ],
   "TargetTags": [
      {
         "Key": "costcenter",
         "Value": "115"
      }
   ],
   "Schedules":[
      {
         "Name": "DailySnapshots",
         "TagsToAdd": [
            {
               "Key": "type",
               "Value": "myDailySnapshot"
            }
         ],
         "CreateRule": {
            "Interval": 24,
            "IntervalUnit": "HOURS",
            "Times": [
               "03:00"
            ]
         },
         "RetainRule": {
            "Count": 5
         },
         "CopyTags": false
      }
   ]
}
```

성공할 경우, 이 명령은 새로 생성된 정책의 ID를 반환합니다. 다음은 예제 출력입니다.

```
{
   "PolicyId": "policy-0123456789abcdef0"
}
```

수명 주기 정책 표시

get-lifecycle-policy 명령을 사용하여 수명 주기 정책에 대한 정보를 표시합니다.

```
aws dlm get-lifecycle-policy --policy-id policy-0123456789abcdef0
```

다음은 예제 출력입니다. 여기에는 사용자가 지정한 정보와 AWS에서 삽입한 메타데이터가 들어 있습니다.

```
{
   "Policy":{
      "Description": "My first policy",
      "DateCreated": "2018-05-15T00:16:21+0000",
      "State": "ENABLED",
      "ExecutionRoleArn":
 "arn:aws:iam::210774411744:role/AWSDataLifecycleManagerDefaultRole",
      "PolicyId": "policy-0123456789abcdef0",
      "DateModified": "2018-05-15T00:16:22+0000",
      "PolicyDetails": {
        "PolicyType":"EBS_SNAPSHOT_MANAGEMENT",
```

```
            "ResourceTypes": [
               "VOLUME"
            ],
            "TargetTags": [
               {
                  "Value": "115",
                  "Key": "costcenter"
               }
            ],
            "Schedules": [
               {
                  "TagsToAdd": [
                     {
                        "Value": "myDailySnapshot",
                        "Key": "type"
                     }
                  ],
                  "RetainRule": {
                     "Count": 5
                  },
                  "CopyTags": false,
                  "CreateRule": {
                     "Interval": 24,
                     "IntervalUnit": "HOURS",
                     "Times": [
                        "03:00"
                     ]
                  },
                  "Name": "DailySnapshots"
               }
            ]
         }
      }
}
```

수명 주기 정책 수정

update-lifecycle-policy 명령을 사용하여 수명 주기 정책의 정보를 수정합니다. 구문을 간단히 하기 위해 이 예에서는 정책의 세부 정보가 들어 있는 JSON 파일(`policyDetailsUpdated.json`)을 참조합니다.

```
aws dlm update-lifecycle-policy --state DISABLED --execution-role-arn
 arn:aws:iam::12345678910:role/AWSDataLifecycleManagerDefaultRole" --policy-details
 file://policyDetailsUpdated.json
```

다음은 `policyDetailsUpdated.json` 파일의 예입니다.

```
{
   "ResourceTypes":[
      "VOLUME"
   ],
   "TargetTags":[
      {
         "Key": "costcenter",
         "Value": "120"
      }
   ],
   "Schedules":[
      {
         "Name": "DailySnapshots",
         "TagsToAdd": [
            {
               "Key": "type",
               "Value": "myDailySnapshot"
```

```
            }
         ],
         "CreateRule": {
            "Interval": 12,
            "IntervalUnit": "HOURS",
            "Times": [
               "15:00"
            ]
         },
         "RetainRule": {
            "Count" :5
         },
         "CopyTags": false
      }
   ]
}
```

업데이트된 정책을 보려면 `get-lifecycle-policy` 명령을 사용하십시오. 상태, 태그 값, 스냅샷 간격, 스냅샷 시작 시간이 변경된 것을 알 수 있습니다.

수명 주기 정책 삭제

delete-lifecycle-policy 명령을 사용하여 수명 주기 정책을 삭제하고, 정책에 지정된 대상 태그를 해제하여 재사용할 수 있도록 합니다.

```
aws dlm delete-lifecycle-policy --policy-id policy-0123456789abcdef0
```

API를 사용하여 백업 관리

Amazon 데이터 수명 주기 관리자 API 참조에 Amazon 데이터 수명 주기 관리자 쿼리 API의 데이터 유형과 각 작업에 대한 설명 및 구문이 나와 있습니다.

또는 AWS SDK 중 하나를 사용하여 사용 중인 프로그래밍 언어나 플랫폼에 맞게 조정된 방식으로 API에 액세스할 수도 있습니다. 자세한 정보는 AWS SDK를 참조하십시오.

스냅샷 수명 주기 모니터링

다음 기능을 사용하여 스냅샷의 수명 주기를 모니터링할 수 있습니다.

기능

- 콘솔 및 AWS CLI (p. 956)
- CloudWatch 이벤트 (p. 956)
- AWS CloudTrail (p. 957)

콘솔 및 AWS CLI

Amazon EC2 콘솔 또는 AWS CLI를 사용하여 수명 주기 정책을 볼 수 있습니다. 정책에 따라 생성된 각 스냅샷에는 타임스탬프 및 해당 정책과 관련된 태그가 있습니다. 이 태그로 스냅샷을 필터링하여 백업이 의도대로 생성되고 있는지 확인할 수 있습니다. 콘솔에서 수명 주기 정책을 보는 방법에 대한 자세한 내용은 수명 주기 정책 보기 (p. 952) 단원을 참조하십시오. CLI를 사용하여 수명 주기 정책에 대한 정보를 표시하는 방법은 수명 주기 정책 표시 (p. 954) 단원을 참조하십시오.

CloudWatch 이벤트

Amazon EBS 및 Amazon 데이터 수명 주기 관리자는 수명 주기 정책 작업과 관련된 이벤트를 발생시킵니다. AWS Lambda 및 Amazon CloudWatch Events를 사용하여 이벤트 알림을 프로그래밍 방식으로 처리할 수 있습니다. 자세한 내용은 Amazon CloudWatch Events 사용 설명서를 참조하십시오.

다음과 같은 이벤트를 사용할 수 있습니다.

- `createSnapshot` - `CreateSnapshot` 작업이 성공하거나 실패할 때 생기는 Amazon EBS 이벤트입니다. 자세한 내용은 Amazon EBS용 Amazon CloudWatch Events (p. 1017) 단원을 참조하십시오.
- `DLM Policy State Change` - 수명 주기 정책이 오류 상태가 될 때 생기는 Amazon 데이터 수명 주기 관리자 이벤트입니다. 이 이벤트에는 오류의 원인에 대한 설명이 포함되어 있습니다. 다음은 IAM 역할에서 부여한 권한이 충분할 때 나타나는 이벤트의 예입니다.

```
{
   "version": "0",
   "id": "01234567-0123-0123-0123-0123456789ab",
   "detail-type": "DLM Policy State Change",
   "source": "aws.dlm",
   "account": "123456789012",
   "time": "2018-05-25T13:12:22Z",
   "region": "us-east-1",
   "resources": [
      "arn:aws:dlm:us-east-1:123456789012:policy/policy-0123456789abcdef"
   ],
   "detail": {
      "state": "ERROR",
      "cause": "Role provided does not have sufficient permissions",
      "policy_id": "arn:aws:dlm:us-east-1:123456789012:policy/policy-0123456789abcdef"
   }
}
```

다음은 한도를 초과할 때 발생하는 이벤트의 예입니다.

```
{
   "version": "0",
   "id": "01234567-0123-0123-0123-0123456789ab",
   "detail-type": "DLM Policy State Change",
   "source": "aws.dlm",
   "account": "123456789012",
   "time": "2018-05-25T13:12:22Z",
   "region": "us-east-1",
   "resources": [
      "arn:aws:dlm:us-east-1:123456789012:policy/policy-0123456789abcdef"
   ],
   "detail":{
      "state": "ERROR",
      "cause": "Maximum allowed active snapshot limit exceeded",
      "policy_id": "arn:aws:dlm:us-east-1:123456789012:policy/policy-0123456789abcdef"
   }
}
```

AWS CloudTrail

AWS CloudTrail로 사용자 활동과 API 사용을 추적하여 내부 정책 및 규제 표준에 대한 준수 사실을 입증할 수 있습니다. 자세한 정보는 AWS CloudTrail User Guide를 참조하십시오.

Amazon EBS 데이터 서비스

Amazon EBS는 다음 데이터 서비스를 제공합니다.

데이터 서비스

- Amazon EBS 탄력적 볼륨 (p. 958)
- Amazon EBS Encryption (p. 968)

- Amazon EBS 빠른 스냅샷 복원 (p. 978)

Amazon EBS 탄력적 볼륨

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

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

내용

- 볼륨 수정 시 요구 사항 (p. 958)
- EBS 볼륨에 대한 수정 요청 (p. 959)
- 볼륨 수정 진행률 모니터링 (p. 962)
- 볼륨 크기 조정 후 Linux 파일 시스템 확장 (p. 965)

볼륨 수정 시 요구 사항

Amazon EBS 볼륨을 수정할 때 다음과 같은 요구 사항과 제한 사항이 적용됩니다. EBS 볼륨에 대한 일반 요구 사항에 대한 자세한 내용은 EBS 볼륨의 크기 및 구성에 대한 제약 (p. 905)을 참조하십시오.

Amazon EC2 인스턴스 지원

탄력적 볼륨을 지원하는 인스턴스는 다음과 같습니다.

- 모든 현재 세대 인스턴스 (p. 179)
- 이전 세대 인스턴스 패밀리 C1, C3, CC2, CR1, G2, I2, M1, M3 및 R3

인스턴스 유형에서 탄력적 볼륨을 지원하지 않는 경우에는 탄력적 볼륨이 지원되지 않는 경우의 EBS 볼륨 수정 (p. 962) 단원을 참조하십시오.

Linux 볼륨에 대한 요구 사항

Linux AMI에서 부팅 볼륨 2TiB(2,048GiB) 이상을 사용하려면 GUID 파티션 테이블(GPT)과 GRUB 2가 필요합니다. 현재 여러 Linux AMI에서도 부팅 볼륨 크기를 최대 2TiB까지만 지원하는 MBR 파티셔닝 체계를 사용하고 있습니다. 인스턴스가 2TiB 이상의 부팅 볼륨에서 부팅되지 않는 경우, 사용 중인 AMI의 부팅 볼륨 크기가 2TiB 미만으로 제한된 상태일 수 있습니다. 부팅 볼륨이 아닌 볼륨에는 이 Linux 인스턴스에 대한 제한이 적용되지 않습니다. Windows 볼륨에 적용되는 요구 사항의 경우에는 Windows 인스턴스용 Amazon EC2 사용 설명서의 Windows 볼륨에 대한 요구 사항을 참조하십시오.

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

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

GPT 파티셔닝을 사용하는 Amazon Linux 인스턴스는 다음 정보를 반환합니다.

```
GPT fdisk (gdisk) version 0.8.10

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

```
    GPT: present

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

MBR 파티셔닝을 사용하는 SUSE 인스턴스는 다음 정보를 반환합니다.

```
GPT fdisk (gdisk) version 0.8.8

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

제한 사항

- 다중 연결 지원 Amazon EBS 볼륨에서는 탄력적 볼륨 작업이 지원되지 않습니다.
- 새로운 볼륨 크기는 지원되는 볼륨 용량을 초과할 수 없습니다. 자세한 내용은 EBS 볼륨의 크기 및 구성에 대한 제약 (p. 905) 단원을 참조하십시오.
- 볼륨이 2016년 11월 3일 23:40 UTC 이전에 연결된 경우에는 탄력적 볼륨 지원을 초기화해야 합니다. 자세한 내용은 탄력적 볼륨 지원 초기화 (p. 961)를 참조하십시오.
- 지원되지 않는 이전 세대의 인스턴스 유형을 사용하거나 볼륨을 수정하려는 중에 오류가 발생한 경우, 탄력적 볼륨이 지원되지 않는 경우의 EBS 볼륨 수정 (p. 962) 단원을 참조하십시오.
- 루트 볼륨으로 인스턴스에 연결된 `gp2` 볼륨은 `st1` 또는 `sc1` 볼륨으로 수정될 수 없습니다. 분리되거나 `st1` 또는 `sc1`으로 수정되었다면, 루트 볼륨으로 인스턴스에 연결할 수 없습니다.
- 요청한 볼륨 크기가 `st1`과 `sc1`의 최소 크기 이하라면, `gp2` 볼륨은 `st1` 또는 `sc1` 볼륨으로 수정될 수 없습니다.
- 경우에 따라 수정을 진행하기 위해 볼륨을 분리하거나 인스턴스를 중단해야 합니다. EBS 볼륨을 수정하려고 시도할 때 오류 메시지가 표시되는 경우 또는 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.
  - 루트(부트) 볼륨의 경우, 인스턴스를 중단하고 수정 사항을 적용한 다음 인스턴스를 다시 시작합니다.
- 기존 `io1` 볼륨에서 32,000 IOPS를 초과하여 프로비저닝하려는 경우 다음 중 하나를 수행하여 전체 성능 개선을 확인해야 할 수도 있습니다.
  - 볼륨을 분리한 후 다시 연결합니다.
  - 인스턴스를 다시 시작합니다.
- EBS 볼륨 크기 축소는 지원되지 않습니다. 그러나 비슷한 볼륨을 만든 후 rsync 같은 애플리케이션 수준 도구를 사용하여 그 볼륨으로 데이터를 마이그레이션할 수 있습니다.
- 완전히 초기화되지 않은 볼륨을 수정하는 경우 수정 시간이 늘어납니다. 자세한 내용은 Amazon EBS 볼륨 초기화 (p. 1001) 단원을 참조하십시오.
- 볼륨을 수정한 후 동일한 볼륨에 추가 수정 사항을 적용하려면 볼륨이 `in-use` 또는 `available` 상태가 되도록 6시간 이상 기다립니다.
- `m3.medium` 인스턴스는 볼륨 수정을 완전하게 지원하지만 `m3.large`, `m3.xlarge` 및 `m3.2xlarge` 인스턴스는 일부 볼륨 수정 기능을 지원하지 않을 수 있습니다.

EBS 볼륨에 대한 수정 요청

탄력적 볼륨을 사용하면 Amazon EBS 볼륨을 분리하지 않고도 크기, 성능 및 볼륨 크기를 동적으로 수정할 수 있습니다.

다음 절차에 따라 볼륨을 수정합니다.

1. (선택 사항) 중요한 데이터가 포함된 볼륨을 수정하려면 먼저 변경 내용을 롤백해야 할 경우를 대비하여 볼륨의 스냅샷을 생성하는 것이 바람직합니다. 자세한 내용은 Amazon EBS 스냅샷 만들기 (p. 929) 단원을 참조하십시오.
2. 볼륨 수정을 요청합니다.
3. 볼륨 수정의 진행 상황을 모니터링합니다. 자세한 내용은 볼륨 수정 진행률 모니터링 (p. 962) 단원을 참조하십시오.
4. 볼륨 크기가 수정된 경우 볼륨의 파일 시스템을 확장하여 스토리지 용량 증가를 활용합니다. 자세한 내용은 볼륨 크기 조정 후 Linux 파일 시스템 확장 (p. 965) 단원을 참조하십시오.

내용

- 탄력적 볼륨을 사용하여 EBS 볼륨 수정(콘솔) (p. 960)
- 탄력적 볼륨을 사용하여 EBS 볼륨 수정(AWS CLI) (p. 960)
- 탄력적 볼륨 지원 초기화(필요한 경우) (p. 961)
- 탄력적 볼륨이 지원되지 않는 경우의 EBS 볼륨 수정 (p. 962)

탄력적 볼륨을 사용하여 EBS 볼륨 수정(콘솔)

다음 절차를 사용하여 EBS 볼륨을 수정합니다.

콘솔을 사용하여 EBS 볼륨을 수정하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 볼륨을 선택하고, 수정할 볼륨을 선택한 다음, 작업, 볼륨 수정을 선택합니다.
3. 볼륨 수정 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 볼륨 유형의 값을 선택합니다.
   - 크기를 수정하려면 크기에 허용된 정수 값을 입력합니다.
   - 볼륨 유형으로 Provisioned IOPS SSD(io1)(프로비저닝된 IOPS SSD(io1))를 선택한 경우에는 IOPS에 허용되는 정수 값을 입력합니다.
4. 볼륨 설정 변경을 완료했으면 수정을 선택합니다. 확인 메시지가 표시되면 예를 선택합니다.
5. 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. 965) 단원을 참조하십시오.

탄력적 볼륨을 사용하여 EBS 볼륨 수정(AWS CLI)

modify-volume 명령을 사용하여 볼륨의 구성 설정을 하나 이상 수정합니다. 예를 들어 크기가 100GiB이고 유형이 `gp2`인 볼륨을 가지고 있는 경우, 다음 명령이 IOPS가 10,000이고 크기가 200GiB이며 유형이 `io1`인 볼륨에 대한 구성을 변경합니다.

```
aws ec2 modify-volume --volume-type io1 --iops 10000 --size 200 --volume-
id vol-1111111111111111
```

다음은 예제 출력입니다.

```
{
    "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,
        "OriginalSize": 100
    }
}
```

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. 965) 단원을 참조하십시오.

탄력적 볼륨 지원 초기화(필요한 경우)

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

- 볼륨을 분리한 후 다시 연결합니다.
- 인스턴스 중지 및 시작

다음 절차 중 하나를 사용하여 인스턴스가 볼륨 수정이 가능한 상태인지를 확인합니다.

인스턴스가 콘솔을 사용할 준비가 되었는지를 확인하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 열 표시/숨기기(Show/Hide Columns) 아이콘(기어 모양)을 선택합니다. 시작 시간 및 블록 디바이스 속성을 선택한 다음 닫기를 선택합니다.
4. 시작 시간(Launch Time) 열을 기준으로 인스턴스의 목록을 정렬합니다. 컷오프 날짜 이전에 시작된 인스턴스의 경우 디바이스가 연결된 시기를 확인합니다. 다음 예에서, 첫 번째 인스턴스가 컷오프 날짜 이전에 시작되었으며 해당 루트 볼륨이 컷오프 날짜 이전에 연결되었으므로 첫 번째 인스턴스에 대한 볼륨 수정을 초기화해야 합니다. 다른 인스턴스는 컷오프 날짜 이후에 시작되었으므로 사용 준비가 되었습니다.

Instance ID	Launch Time	Block Devices
i-e905622e	February 25, 2016 at 1:49:35 PM UTC-8	/dev/xvda=vol-e6b46410:attached:2016-02-25T21:49:35.000Z:true
i-719f99a8	December 8, 2016 at 2:21:51 PM UTC-8	/dev/xvda=vol-bad60e7a:attached:2016-01-15T18:36:12.000Z:true
i-006b02c1b78381e57	May 17, 2017 at 1:52:52 PM UTC-7	/dev/sda1=vol-0de9250441c73024c:attached:2017-05-17T20:52:53.000Z:true, xvdb=vol-0863a86c393496d3d:attached:2017-05-17T20:52:53.000Z:false
i-e3d172ed	May 17, 2017 at 2:48:54 PM UTC-7	/dev/sda1=vol-04c34d0b:attached:2015-01-21T21:19:46.000Z:true

인스턴스가 CLI를 사용할 준비가 되었는지를 확인하려면

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

```
aws ec2 describe-instances --query "Reservations[*].Instances[*].
[InstanceId,LaunchTime<='2016-11-01',BlockDeviceMappings[*][Ebs.AttachTime<='2016-11-01']]"
 --output text
```

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

```
i-e905622e              True
```

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

탄력적 볼륨이 지원되지 않는 경우의 EBS 볼륨 수정

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

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

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

```
[ec2-user ~]$ df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/xvda1            7.9G  943M  6.9G  12% /
tmpfs                 1.9G     0  1.9G   0% /dev/shm
```

새로 확장된 볼륨이 크기에 반영되지 않을 경우 인스턴스에서 새 공간을 사용할 수 있도록 디바이스의 파일 시스템을 확장해야 합니다. 자세한 내용은 볼륨 크기 조정 후 Linux 파일 시스템 확장 (p. 965) 단원을 참조하십시오.

볼륨 수정 진행률 모니터링

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

> **Note**
>
> 드물지만 일시적 AWS 결함으로 인해 `failed` 상태가 될 수 있습니다. 이는 볼륨 상태를 나타내는 것이 아니라 단지 볼륨 수정이 실패했음을 나타내는 것입니다. 이 경우 볼륨 수정을 다시 시도합니다.

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

볼륨 수정 변경 사항은 다음과 같이 적용됩니다.

- 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 적용되려면 최대 24시간이 걸리며, 볼륨이 완전히 초기화되지 않은 경우처럼 특별한 경우에는 더 걸릴 수도 있습니다. 일반적으로 완전히 사용된 1TiB 볼륨은 새 성능 구성으로 마이그레이션하는 데 약 6시간이 걸립니다.

다음 방법 중 하나를 사용하여 볼륨 수정의 진행 상황을 모니터링합니다.

내용

- 볼륨 수정의 진행 상황 모니터링(콘솔) (p. 963)
- 볼륨 수정의 진행 상황 모니터링(AWS CLI) (p. 963)

- 볼륨 수정의 진행 상황 모니터링(CloudWatch 이벤트) (p. 964)

볼륨 수정의 진행 상황 모니터링(콘솔)

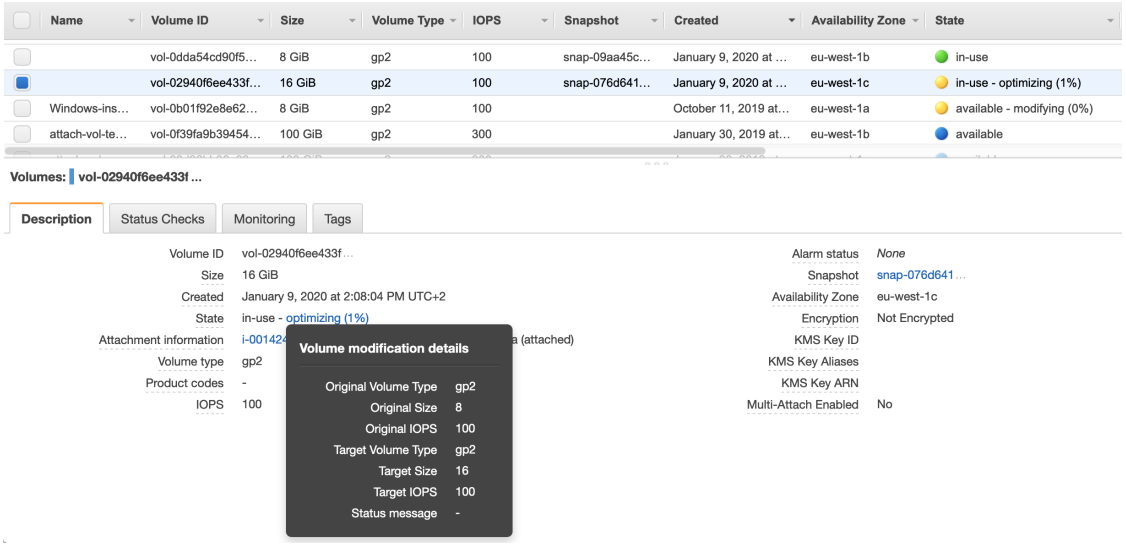
다음 절차를 사용하여 하나 이상의 볼륨 수정 진행 상황을 모니터링합니다.

콘솔을 사용하여 수정의 진행 상황을 모니터링하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 볼륨을 선택합니다.
3. 볼륨을 선택합니다. 볼륨 수정 상태가 상태 열과 세부 정보 창의 상태 필드에 표시됩니다. 다음 스크린샷에서 선택한 볼륨의 수정 상태는 optimizing입니다. 목록에 있는 다음 볼륨의 상태는 modifying입니다.

   Note

   또한 상태 열에는 볼륨의 가용성 상태가 creating, available, in-use, deleting, deleted 또는 error로 표시됩니다.
4. 이 스크린샷과 같이 상태 필드에서 텍스트를 선택하여 가장 최근의 수정 작업에 대한 전/후 정보를 표시합니다.



볼륨 수정의 진행 상황 모니터링(AWS CLI)

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

```
aws ec2 describe-volumes-modifications --volume-
id vol-1111111111111111 vol-2222222222222222
```

다음 예제 출력에서 볼륨 수정의 여전히 `modifying` 상태입니다.

```
{
    "VolumesModifications": [
        {
            "TargetSize": 200,
            "TargetVolumeType": "io1",
            "ModificationState": "modifying",
            "VolumeId": "vol-1111111111111111",
```

```
            "TargetIops": 10000,
            "StartTime": "2017-01-19T22:21:02.959Z",
            "Progress": 0,
            "OriginalVolumeType": "gp2",
            "OriginalIops": 300,
            "OriginalSize": 100
        },
        {
            "TargetSize": 2000,
            "TargetVolumeType": "sc1",
            "ModificationState": "modifying",
            "VolumeId": "vol-2222222222222222",
            "StartTime": "2017-01-19T22:23:22.158Z",
            "Progress": 0,
            "OriginalVolumeType": "gp2",
            "OriginalIops": 300,
            "OriginalSize": 1000
        }
    ]
}
```

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

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

다음은 두 볼륨에 대한 정보가 포함된 출력 예제입니다.

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

볼륨 수정의 진행 상황 모니터링(CloudWatch 이벤트)

CloudWatch 이벤트를 사용하여 볼륨 수정 이벤트에 대한 알림 규칙을 생성할 수 있습니다. 규칙을 사용하여 Amazon SNS로 알림 메시지를 생성하거나 일치하는 이벤트에 대한 응답으로 Lambda 함수를 호출할 수 있습니다.

CloudWatch 이벤트를 사용하여 수정 진행 상황을 모니터링하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 이벤트, 규칙 생성을 선택합니다.
3. 서비스별 이벤트와 일치시킬 이벤트 패턴을 작성에 대해 사용자 지정 이벤트 패턴을 선택합니다.
4. 사용자 지정 이벤트 패턴 작성의 내용을 다음과 같이 바꾸고 저장을 선택합니다.

```
{
  "source": [
    "aws.ec2"
  ],
  "detail-type": [
```

```
    "EBS Volume Notification"
  ],
  "detail": {
    "event": [
      "modifyVolume"
    ]
  }
}
```

다음은 이벤트 데이터 예제입니다.

```
{
   "version": "0",
   "id": "01234567-0123-0123-0123-012345678901",
   "detail-type": "EBS Volume Notification",
   "source": "aws.ec2",
   "account": "012345678901",
   "time": "2017-01-12T21:09:07Z",
   "region": "us-east-1",
   "resources": [
      "arn:aws:ec2:us-east-1:012345678901:volume/vol-03a55cf56513fa1b6"
   ],
   "detail": {
      "result": "optimizing",
      "cause": "",
      "event": "modifyVolume",
      "request-id": "01234567-0123-0123-0123-0123456789ab"
   }
}
```

볼륨 크기 조정 후 Linux 파일 시스템 확장

EBS 볼륨 크기를 늘리고 난 후에는 파일 시스템 관련 명령을 사용하여 파일 시스템의 크기를 늘려야 합니다. 볼륨이 `optimizing` 상태가 되자마자 파일 시스템 크기 조정을 시작할 수 있습니다.

> **Important**
>
> 중요한 데이터가 포함된 파일 시스템을 확장하려면 먼저 변경 내용을 롤백해야 할 경우를 대비하여 파일 시스템이 저장된 볼륨 스냅샷을 생성하는 것이 바람직합니다. 자세한 내용은 Amazon EBS 스냅샷 만들기 (p. 929) 단원을 참조하십시오. Linux AMI가 MBR 파티셔닝 체계를 사용하는 경우 부팅 볼륨 크기는 2TiB로 제한됩니다. 자세한 내용은 Linux 볼륨에 대한 요구 사항 (p. 958) 및 EBS 볼륨의 크기 및 구성에 대한 제약 (p. 905) 단원을 참조하십시오.

Windows 파일 시스템 확장에 대한 자세한 내용은 Windows 인스턴스용 Amazon EC2 사용 설명서의 볼륨 크기 조정 후 Windows 파일 시스템 확장을 참조하십시오.

다음 작업에서 인스턴스의 부팅 볼륨 크기를 8GB에서 16GB로, 추가 볼륨의 크기를 8GB에서 30GB로 조정했다고 가정해봅시다.

작업

- 볼륨의 파일 시스템 식별 (p. 965)
- 파티션 확장(필요한 경우) (p. 966)
- 파일 시스템 확장 (p. 967)

볼륨의 파일 시스템 식별

인스턴스의 각 볼륨에서 사용 중인 파일 시스템을 확인하려면 인스턴스를 연결 (p. 485)하여 file -s 명령을 실행합니다.

예제: Nitro 기반 인스턴스의 파일 시스템

다음 예제는 XFS 파일 시스템에 부팅 볼륨과 추가 볼륨을 가지고 있는 Nitro 기반 인스턴스 (p. 181)를 보여줍니다.

```
[ec2-user ~]$ sudo file -s /dev/nvme?n*
/dev/nvme0n1:     x86 boot sector ...
/dev/nvme0n1p1:   SGI XFS filesystem data ...
/dev/nvme0n1p128: data
/dev/nvme1n1:     SGI XFS filesystem data ...
```

예제: T2 인스턴스의 파일 시스템

다음 예제는 ext4 파일 시스템에 부팅 볼륨을, XFS 파일 시스템에 추가 볼륨을 가지고 있는 T2 인스턴스를 보여줍니다.

```
[ec2-user ~]$ sudo file -s /dev/xvd*
/dev/xvda:  DOS/MBR boot sector ..
/dev/xvda1: Linux rev 1.0 ext4 filesystem data ...
/dev/xvdf:  SGI XFS filesystem data ...
```

파티션 확장(필요한 경우)

EBS 볼륨에 파일 시스템과 데이터를 포함하는 파티션이 있을 수 있습니다. 볼륨 크기를 늘려도 파티션 크기가 늘어나지 않습니다. 크기가 조정된 볼륨에서 파일 시스템을 확장하기 전에 볼륨에 볼륨 크기를 늘려야 하는 파티션이 있는지 여부를 확인하십시오.

lsblk 명령을 사용하여 인스턴스에 연결된 블록 디바이스에 대한 정보를 표시합니다. 크기가 조정된 볼륨에 파티션이 있고 파티션에 볼륨의 새 크기가 반영되지 않는 경우, growpart 명령을 사용하여 파티션을 확장합니다. LVM 파티션 확장에 대한 자세한 내용은 논리 볼륨 확장을 참조하십시오.

예제: Nitro 기반 인스턴스의 파티션

다음 예제는 Nitro 기반 인스턴스의 볼륨을 보여줍니다.

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

- 루트 볼륨 `/dev/nvme0n1`에는 `/dev/nvme0n1p1`라는 파티션이 있습니다. 루트 볼륨에 새 크기인 16GB가 반영되는 동안 파티션의 크기에 원래 크기인 8GB가 반영되기 때문에 파일 시스템을 확장하려면 먼저 파티션 크기를 늘려야 합니다.
- 볼륨 `/dev/nvme1n1`에는 파티션이 없습니다. 볼륨 크기에 새 크기 30GB가 반영됩니다.

루트 볼륨에서 파티션을 확장하려면 다음 growpart 명령을 사용합니다. 디바이스 이름과 파티션 번호 사이에 공백이 있다는 점에 유의하십시오.

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

다시 lsblk 명령을 사용하여 파티션에 늘어난 볼륨 크기가 반영되었는지 확인할 수 있습니다.

```
[ec2-user ~]$ lsblk
NAME          MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
```

```
nvme1n1       259:0    0  30G  0 disk /data
nvme0n1       259:1    0  16G  0 disk
##nvme0n1p1   259:2    0  16G  0 part /
##nvme0n1p128 259:3    0   1M  0 part
```

예제: T2 인스턴스의 파티션

다음 예제는 T2 인스턴스의 볼륨을 보여줍니다.

```
[ec2-user ~]$ lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda    202:0    0  16G  0 disk
##xvda1 202:1    0   8G  0 part /
xvdf    202:80   0  30G  0 disk
##xvdf1 202:81   0   8G  0 part /data
```

- 루트 볼륨 `/dev/xvda`에는 `/dev/xvda1`라는 파티션이 있습니다. 볼륨 크기가 16GB일 때 파티션의 크기가 여전히 8GB이기 때문에 파티션 크기를 늘려야 합니다.
- 볼륨 `/dev/xvdf`에는 `/dev/xvdf1`라는 파티션이 있습니다. 볼륨 크기가 30GB일 때 파티션의 크기가 여전히 8GB이기 때문에 파티션 크기를 늘려야 합니다.

각 볼륨에서 파티션을 확장하려면 다음 growpart 명령을 사용합니다. 디바이스 이름과 파티션 번호 사이에 공백이 있다는 점에 유의하십시오.

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

다시 lsblk 명령을 사용하여 파티션에 늘어난 볼륨 크기가 반영되었는지 확인할 수 있습니다.

```
[ec2-user ~]$ lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda    202:0    0  16G  0 disk
##xvda1 202:1    0  16G  0 part /
xvdf    202:80   0  30G  0 disk
##xvdf1 202:81   0  30G  0 part /data
```

파일 시스템 확장

파일 시스템 관련 명령을 사용하여 새 볼륨 용량에 맞게 파일 시스템의 크기를 조정합니다. 여기 나온 예제 이외의 파일 시스템에 대한 확장 지침은 해당 파일 시스템의 설명서를 참조하십시오.

예제: ext2, ext3 또는 ext4 파일 시스템 확장

df -h 명령을 사용하여 각 볼륨에 대한 파일 시스템의 크기를 확인합니다. 이 예제에서는 `/dev/xvda1`과 `/dev/xvdf` 모두에 볼륨의 원래 크기인 8GB가 반영되었습니다.

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

resize2fs 명령을 사용하여 각 볼륨에서 파일 시스템을 확장합니다.

```
[ec2-user ~]$ sudo resize2fs /dev/xvda1
[ec2-user ~]$ sudo resize2fs /dev/xvdf1
```

다시 df -h 명령을 사용하여 각 파일 시스템에 늘어난 볼륨 크기가 반영되었는지 확인할 수 있습니다.

```
[ec2-user ~]$ df -h
Filesystem       Size  Used Avail Use% Mounted on
/dev/xvda1        16G  1.9G  14G  12% /
/dev/xvdf1        30G   45M  30G   1% /data
...
```

예제: XFS 파일 시스템 확장

df -h 명령을 사용하여 각 볼륨에 대한 파일 시스템의 크기를 확인합니다. 이 예제에서는 각 파일 시스템에 원래 볼륨 크기인 8GB가 반영되었습니다.

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

XFS 파일 시스템을 확장하려면 XFS 도구가 아직 설치되어 있지 않은 경우 다음과 같이 설치합니다.

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

xfs_growfs 명령을 사용하여 각 볼륨에서 파일 시스템을 확장합니다. 이 예제에서 `/` 및 `/data`는 df -h의 출력에 나와 있는 볼륨 탑재 지점입니다.

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

다시 df -h 명령을 사용하여 각 파일 시스템에 늘어난 볼륨 크기가 반영되었는지 확인할 수 있습니다.

```
[ec2-user ~]$ df -h
Filesystem       Size  Used Avail Use% Mounted on
/dev/nvme0n1p1    16G  1.6G   15G  10% /
/dev/nvme1n1      30G   33M   30G   1% /data
...
```

Amazon EBS Encryption

Amazon EBS 암호화는 자체 키 관리 인프라를 사용자가 직접 구축, 유지 및 보호할 필요가 없는 EBS 볼륨에 대한 간단한 암호화 솔루션을 제공합니다. 암호화된 볼륨 및 스냅샷을 생성할 때 AWS Key Management Service(AWS KMS) 고객 마스터 키(CMK)를 사용합니다.

암호화 작업은 EC2 인스턴스를 호스팅하는 서버에서 진행되며, 인스턴스와 인스턴스에 연결된 EBS 스토리지 간 유휴 데이터 및 전송 중 데이터의 보안을 모두 보장합니다.

목차

- EBS의 암호화 방식 (p. 969)
- 요구 사항 (p. 969)
- EBS 암호화에 대한 기본 키 (p. 970)
- 암호화 기본 제공 (p. 971)
- EBS 리소스 암호화 (p. 971)
- 암호화 시나리오 (p. 972)

- API 및 CLI를 사용한 암호화 기본값 설정 (p. 978)

EBS의 암호화 방식

EC2 인스턴스의 부팅 및 데이터 볼륨을 모두 암호화할 수 있습니다. 암호화된 EBS 볼륨을 만들고 지원되는 인스턴스 유형에 이 볼륨을 연결하면 다음 유형의 데이터가 암호화됩니다.

- 볼륨 내부에 있는 데이터
- 볼륨과 인스턴스 사이에서 이동하는 모든 데이터
- 볼륨에서 생성된 모든 스냅샷
- 그런 스냅샷에서 생성된 모든 볼륨

EBS는 산업 표준 AES-256 알고리즘을 사용하여 데이터 키로 볼륨을 암호화합니다. 데이터 키는 암호화된 데이터와 함께 디스크에 저장되지만 EBS가 고객의 CMK로 암호화하기 전에는 저장되지 않습니다. 데이터 키가 디스크에서 일반 텍스트로 보이는 일은 결코 없습니다. 볼륨의 스냅샷은 동일한 데이터 키를 공유하며 그런 스냅샷에서 생성한 후속 볼륨도 데이터 키를 공유합니다. 자세한 내용은 AWS Key Management Service Developer Guide의 데이터 키를 참조하십시오.

Amazon EBS는 AWS KMS와 함께 작동하여 다음과 같이 EBS 볼륨을 암호화 및 해독합니다.

1. Amazon EBS는 데이터 키를 해독할 수 있도록 AWS KMS에 CreateGrant 요청을 보냅니다.
2. Amazon EBS는 볼륨을 암호화하는 데 사용할 CMK를 지정하도록 AWS KMS에 GenerateDataKeyWithoutPlaintext 요청을 보냅니다.
3. AWS KMS는 새 데이터 키를 생성하고 지정된 CMK 하에서 암호화한 후, 볼륨 메타데이터와 함께 저장할 암호화된 데이터 키를 Amazon EBS로 보냅니다.
4. 암호화된 볼륨을 인스턴스에 연결하면 Amazon EC2는 암호화된 데이터 키를 지정하여 AWS KMS에 Decrypt 요청을 보냅니다.
5. AWS KMS는 암호화된 데이터 키를 해독하고, 해독한 데이터 키를 Amazon EC2로 보냅니다.
6. Amazon EC2는 하이퍼바이저 메모리에서 일반 텍스트 데이터 키를 사용하여 디스크 I/O를 볼륨으로 암호화합니다. 볼륨이 인스턴스에 연결되어 있는 동안에는 일반 텍스트 형태의 데이터 키가 메모리에 유지됩니다.

자세한 내용은 AWS Key Management Service Developer Guide에서 Amazon Elastic Block Store(Amazon EBS)가 AWS KMS를 사용하는 방식 및 AWS KMS 로그 파일 항목을 참조하십시오.

요구 사항

시작하기 전에 다음 요구 사항을 충족하는지 확인하십시오.

지원되는 볼륨 유형

암호화는 모든 EBS 볼륨 유형에서 지원됩니다. 암호화된 볼륨에서는 지연 시간에 대한 영향을 최소화한 채 암호화되지 않은 볼륨과 동일한 IOPS 성능을 기대할 수 있습니다. 암호화되지 않은 볼륨에 액세스하는 것과 동일한 방법으로 암호화된 볼륨에 액세스할 수 있습니다. 암호화 및 암호 해독은 중단 없이 처리되므로 사용자나 사용자의 애플리케이션에서 별도로 조치할 부분은 없습니다.

지원되는 인스턴스 유형

Amazon EBS 암호화는 아래에 나열된 인스턴스 유형에서 사용할 수 있습니다. 이러한 인스턴스 유형에는 암호화된 볼륨과 암호화되지 않은 볼륨을 동시에 연결할 수 있습니다.

- 범용: A1, M3, M4, M5, M5a, M5ad, M5d, M5dn, M5n, T2, T3 및 T3a
- 최적화된 컴퓨팅: C3, C4, C5, C5d 및 C5n

- 메모리 최적화: `cr1.8xlarge`, R3, R4, R5, R5a, R5ad, R5d, R5dn, R5n, `u-6tb1.metal`, `u-9tb1.metal`, `u-12tb1.metal`, `u-18tb1.metal`, `u-24tb1.metal`, X1, X1e 및 z1d
- 스토리지 최적화: D2, `h1.2xlarge`, `h1.4xlarge`, I2, I3, I3en 및 I3en
- 액셀러레이티드 컴퓨팅: F1, G2, G3, G4, P2 및 P3

IAM 사용자의 권한

CMK를 EBS 암호화의 기본 키로 구성하면, 기본 키 정책은 필수 KMS 작업에 대한 액세스 권한이 있는 IAM 사용자가 이 키를 사용하여 EBS 리소스를 암호화하거나 해독하도록 허용합니다. EBS 암호화를 사용하기 위해 다음 작업을 호출할 수 있는 권한을 IAM 사용자에게 부여해야 합니다.

- `kms:CreateGrant`
- `kms:Decrypt`
- `kms:DescribeKey`
- `kms:GenerateDataKeyWithoutPlainText`
- `kms:ReEncrypt`

최소 권한의 주체를 따르려면 `kms:CreateGrant`에 대한 모든 액세스 권한을 허용하지 마십시오. 대신에 다음 예제에 표시된 대로 AWS 서비스에서 사용자를 대신하여 승인을 생성하는 경우에만 사용자는 CMK에 대해 승인을 생성할 수 있습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "kms:CreateGrant",
            "Resource": [
                "arn:aws:kms:us-east-2:123456789012:key/abcd1234-a123-456d-a12b-
a123b4cd56ef"
            ],
            "Condition": {
                "Bool": {
                    "kms:GrantIsForAWSResource": true
                }
            }
        }
    ]
}
```

자세한 내용은 AWS Key Management Service Developer Guide의 기본 키 정책을 참조하십시오.

EBS 암호화에 대한 기본 키

Amazon EBS는 사용자가 AWS 리소스를 저장하는 각 리전에 고유한 AWS 관리형 CMK를 자동으로 생성합니다. 이 키의 별칭은 `alias/aws/ebs`입니다. 기본적으로 Amazon EBS는 암호화에 이 키를 사용합니다. 또는 생성한 대칭 고객 관리형 CMK를 EBS 암호화의 기본 키로 지정할 수 있습니다. 자체 CMK를 사용하여 키 생성, 교체 및 비활성화 기능을 비롯한 다양한 작업을 수행할 수 있습니다.

> **Important**
>
> Amazon EBS에서는 비대칭 CMK가 지원되지 않습니다. 자세한 내용은 AWS Key Management Service 개발자 안내서의 대칭 및 비대칭 키 사용을 참조하십시오.

리전에 대한 EBS 암호화의 기본 키를 구성하려면

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

2. 탐색 모음에서 해당 리전을 선택합니다.
3. 계정 속성, 설정을 선택합니다.
4. Change the default key(기본 키 변경)를 선택한 후 사용 가능한 키를 선택합니다.
5. 업데이트를 선택합니다.

암호화 기본 제공

사용자의 AWS 계정을 구성하여 생성하는 새 EBS 볼륨 및 스냅샷 사본의 암호화를 적용할 수 있습니다. 예를 들어 Amazon EBS는 인스턴스 시작 시 생성된 EBS 볼륨과 암호화되지 않은 스냅샷에서 복사하는 스냅샷을 암호화합니다. 암호화되지 않은 EBS 리소스에서 암호화된 리소스로의 이전 예제는 암호화되지 않은 리소스의 암호화 (p. 972) 단원을 참조하십시오.

기본적으로 암호화는 기존 EBS 볼륨이나 스냅샷에 영향을 미치지 않습니다.

고려 사항

- 암호화 기본 제공은 리전별 설정입니다. 특정 기능에 대해 이 기능을 활성화하면 해당 리전의 개별 볼륨 또는 스냅샷에 대해 비활성화할 수 없습니다.
- 암호화 기본 제공을 활성화하면 인스턴스 유형이 EBS 암호화를 지원할 때에만 인스턴스를 시작할 수 있습니다. 자세한 내용은 지원되는 인스턴스 유형 (p. 969) 단원을 참조하십시오.
- AWS Server Migration Service(SMS)을 사용하여 서버를 마이그레이션할 때 기본적으로 암호화를 켜지 마십시오. 기본적으로 암호화가 이미 설정되어 있고 델타 복제 오류가 발생하는 경우 이 기능을 해제하십시오. 대신 복제 작업을 생성할 때 AMI 암호화를 활성화하십시오.

리전에서 암호화를 기본적으로 활성화하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 모음에서 해당 리전을 선택합니다.
3. 탐색 창에서 EC2 대시보드를 선택합니다.
4. 페이지의 오른쪽 위 모서리에서 계정 속성, 설정을 선택합니다.
5. EBS 스토리지 아래에서 Always encrypt new EBS volumes(항상 새 EBS 볼륨 암호화)를 선택합니다.
6. 업데이트를 선택합니다.

기존 스냅샷이나 암호화된 볼륨과 연동되어 있는 CMK는 변경할 수 없습니다. 하지만 스냅샷 복사 작업 중에 다른 CMK와 연동시킬 수는 있습니다. 복사된 스냅샷은 새로운 CMK로 암호화됩니다.

EBS 리소스 암호화

암호화를 기본으로 (p. 971) 사용하여 암호화를 활성화하거나 암호화하려는 볼륨을 생성할 때 암호화를 활성화하여 EBS 볼륨을 암호화합니다.

볼륨 암호화 시 볼륨을 암호화하는 데 사용할 대칭 CMK를 지정할 수 있습니다. CMK를 지정하지 않은 경우 암호화에 사용되는 키는 소스 스냅샷 및 해당 소유권의 암호화 상태에 따라 달라집니다. 자세한 내용은 암호화 결과표 (p. 976)를 참조하십시오.

기존 스냅샷이나 볼륨과 연동되어 있는 CMK는 변경할 수 없습니다. 하지만 스냅샷 복사 작업 중에 다른 CMK와 연동시킬 수는 있습니다. 복사된 스냅샷은 새로운 CMK로 암호화됩니다.

암호화를 통한 비어 있는 새 볼륨 생성

비어 있는 새 EBS 볼륨을 생성하는 경우 특정 볼륨 생성 작업에 대한 암호화를 활성화하여 암호화할 수 있습니다. EBS 암호화를 기본으로 활성화한 경우 볼륨은 자동으로 암호화됩니다. 기본적으로 볼륨은 EBS 암호화에 대한 기본 키에 암호화됩니다. 또는 특정 볼륨 생성 작업에 대해 다른 대칭 CMK를 지정할 수 있습니다.

볼륨은 최초로 사용 가능한 시점까지 암호화되므로 데이터가 항상 안전한 상태를 유지합니다. 자세한 절차는 Amazon EBS 볼륨 생성 (p. 908) 단원을 참조하십시오.

기본적으로 볼륨을 생성할 때 선택한 CMK가 볼륨에서 생성한 스냅샷과 암호화된 스냅샷에서 복원한 볼륨을 암호화합니다. 암호화된 볼륨 또는 스냅샷으로부터 암호화를 제거할 수 없습니다. 즉, 암호화된 스냅샷 또는 암호화된 스냅샷의 사본에서 복원된 볼륨은 항상 암호화됩니다.

암호화된 볼륨의 퍼블릭 스냅샷은 지원하지 않지만 암호화된 스냅샷을 특정 계정과 공유할 수는 있습니다. 자세한 지침은 Amazon EBS 스냅샷 공유 (p. 938) 단원을 참조하십시오.

암호화되지 않은 리소스의 암호화

암호화되지 않은 기존 볼륨 또는 스냅샷을 암호화는 직접적인 방법은 없지만 볼륨 또는 스냅샷을 생성하여 암호화할 수 있습니다. 암호화를 기본적으로 활성화한 경우 Amazon EBS는 EBS 암호화에 대한 사용자의 기본 키를 사용하여, 결과로 얻은 새로운 볼륨 또는 스냅샷을 암호화합니다. 암호화를 기본적으로 활성화하지 않은 경우라도 개별 볼륨 또는 스냅샷을 생성할 때 암호화를 활성화할 수 있습니다. 기본적으로 또는 개별 생성 작업에서 암호화를 활성화하는 경우 사용자는 EBS 암호화에 대한 기본 키를 재정의하고 대칭 고객 관리형 CMK를 선택할 수 있습니다. 자세한 내용은 Amazon EBS 볼륨 생성 (p. 908) 및 Amazon EBS 스냅샷 복사 (p. 934) 단원을 참조하십시오.

스냅샷 사본을 고객 관리형 CMK로 암호화하려면 암호화되지 않은 스냅샷(활성화되지 않은 암호화 기본 제공) 복사 (p. 974)에 표시된 대로 암호화를 활성화하고 키를 명시해야 합니다.

> **Important**
>
> Amazon EBS에서는 비대칭 CMK가 지원되지 않습니다. 자세한 내용은 AWS Key Management Service 개발자 안내서의 대칭 및 비대칭 키 사용을 참조하십시오.

EBS 지원 AMI에서 인스턴스를 시작할 때 새 암호화 상태를 적용할 수도 있습니다. 이는 EBS 지원 AMI에, 설명된 대로 암호화할 수 있는 EBS 볼륨의 스냅샷이 포함되기 때문입니다. 자세한 내용은 EBS 지원 AMI를 통한 암호화 사용 (p. 145) 단원을 참조하십시오.

암호화 시나리오

암호화된 EBS 리소스를 생성하면 볼륨 생성 파라미터에서 다른 고객 관리형 CMK를 지정하거나 AMI 또는 인스턴스에 대한 블록 장치 매핑을 지정하지 않는 한, 사용자 계정의 EBS 암호화에 대한 기본 키에 의해 암호화됩니다. 자세한 내용은 EBS 암호화에 대한 기본 키 (p. 970) 단원을 참조하십시오.

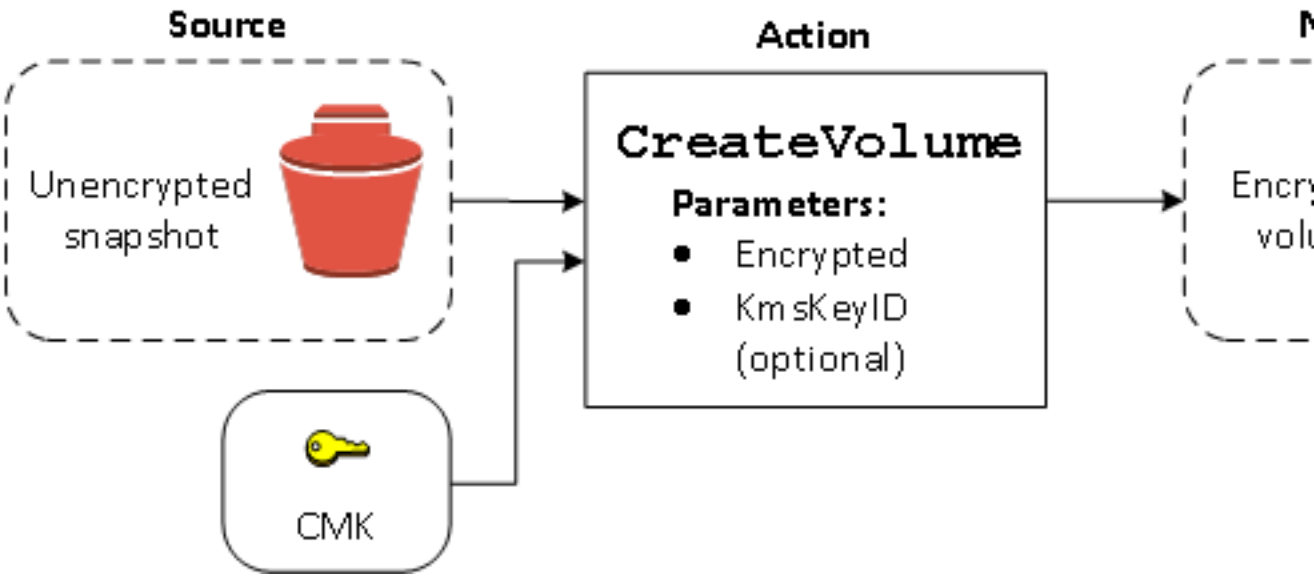
다음 예제에서는 볼륨 및 스냅샷의 암호화 상태를 관리할 수 있는 방법을 보여줍니다. 암호화 사례의 전체 목록은 암호화 결과표 (p. 976)를 참조하십시오.

예제

- 암호화되지 않은 볼륨(활성화되지 않은 암호화 기본 제공) 복원 (p. 972)
- 암호화되지 않은 볼륨(활성화된 암호화 기본 제공) 복원 (p. 973)
- 암호화되지 않은 스냅샷(활성화되지 않은 암호화 기본 제공) 복사 (p. 974)
- 암호화되지 않은 스냅샷(활성화된 암호화 기본 제공) 복사 (p. 974)
- 암호화된 볼륨의 재암호화 (p. 975)
- 암호화된 볼륨의 재암호화 (p. 976)
- 암호화된 볼륨과 암호화되지 않은 볼륨 간 데이터 마이그레이션 (p. 976)
- 암호화 결과 (p. 976)

암호화되지 않은 볼륨(활성화되지 않은 암호화 기본 제공) 복원

암호화 기본 제공을 활성화하지 않은 상태에서는 암호화되지 않은 스냅샷에서 복원한 볼륨이 기본적으로 암호화되지 않습니다. 그러나 `Encrypted` 파라미터와, 선택적으로, `KmsKeyId` 파라미터를 설정하여, 결과로 얻은 볼륨을 암호화할 수 있습니다. 다음 다이어그램에서 프로세스를 보여 줍니다.

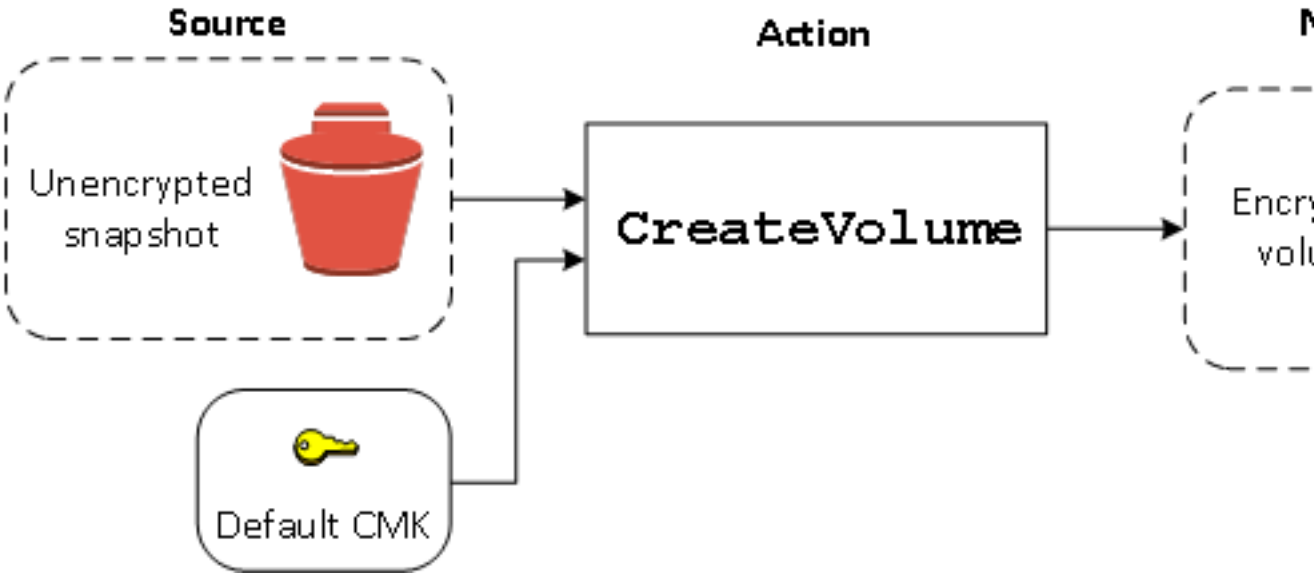


`KmsKeyId` 파라미터를 그대로 놓아두면 결과로 얻은 볼륨이 EBS 암호화에 대한 기본 키를 사용하여 암호화됩니다. 키 ID를 지정하여 해당 볼륨을 다른 CMK로 암호화해야 합니다.

자세한 내용은 스냅샷에서 Amazon EBS 볼륨 복구 (p. 909) 단원을 참조하십시오.

암호화되지 않은 볼륨(활성화된 암호화 기본 제공) 복원

암호화를 기본적으로 활성화한 경우 암호화되지 않은 스냅샷에서 복원된 볼륨에 대한 암호화가 필수이며 사용자의 기본 CMK를 사용하는 데 암호화 파라미터는 필요하지 않습니다. 다음 다이어그램은 이러한 간단한 기본 사례를 보여 줍니다.

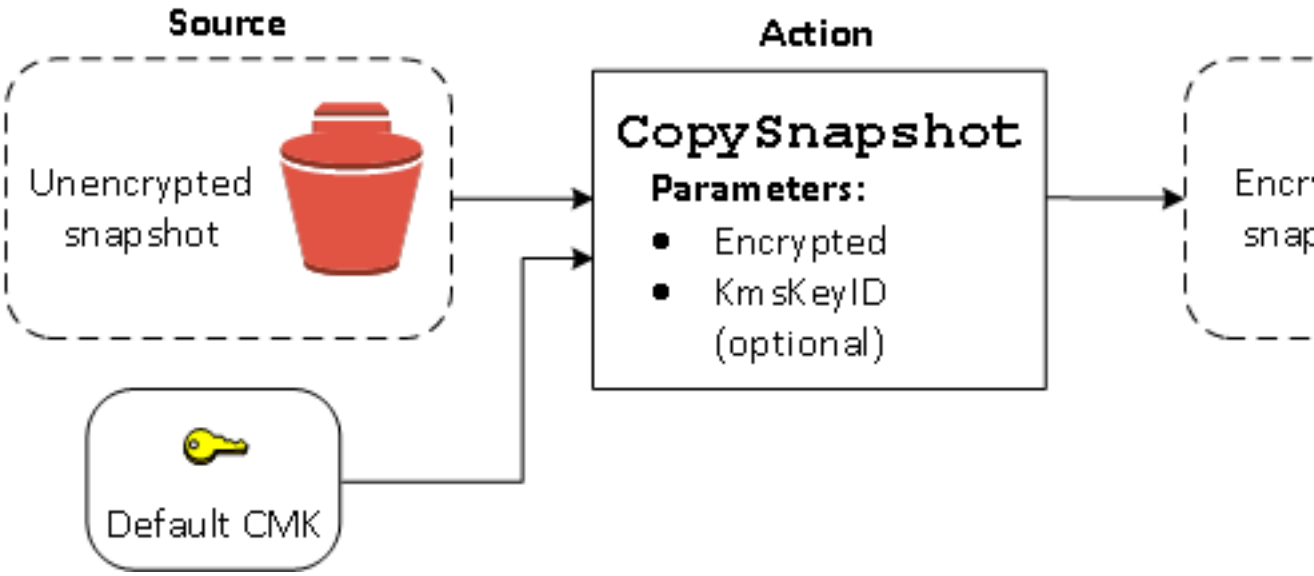


복원된 볼륨을 대칭 고객 관리형 CMK로 암호화하려면 암호화되지 않은 볼륨(활성화되지 않은 암호화 기본 제공) 복원 (p. 972)에서처럼 `Encrypted` 및 `KmsKeyId` 파라미터를 모두 입력해야 합니다.

암호화되지 않은 스냅샷(활성화되지 않은 암호화 기본 제공) 복사

암호화 기본 제공을 활성화하지 않은 상태에서는 암호화되지 않은 스냅샷의 사본이 기본적으로 암호화되지 않습니다. 그러나 `Encrypted` 파라미터와, 선택적으로, `KmsKeyId` 파라미터를 설정하여, 결과로 얻은 볼륨을 암호화할 수 있습니다. `KmsKeyId`를 생략하면 결과로 얻은 스냅샷이 사용자의 기본 CMK로 암호화됩니다. 키 ID를 지정하여 해당 볼륨을 다른 대칭 CMK로 암호화해야 합니다.

다음 다이어그램에서 프로세스를 보여 줍니다.



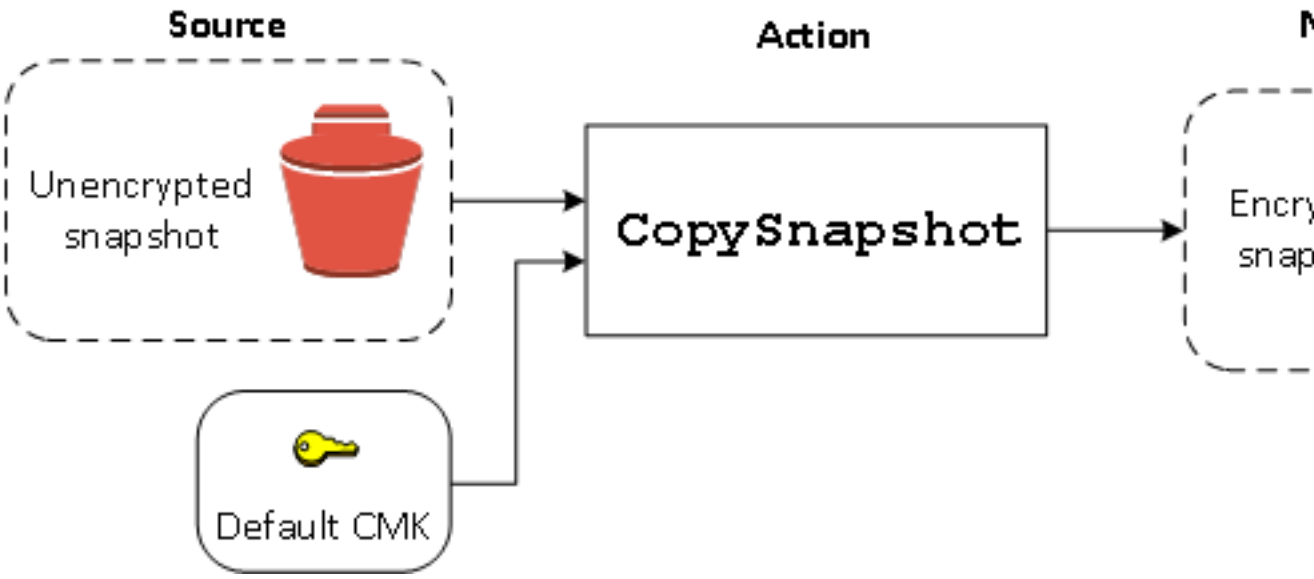
Note

새 CMK로 스냅샷을 복사 및 암호화하면 항상 전체(증분 아님) 복사가 생성되고 추가 지연 및 스토리지 비용이 발생합니다.

암호화되지 않은 스냅샷을 암호화된 스냅샷에 복사한 다음 암호화된 스냅샷에서 볼륨을 생성하여 EBS 볼륨을 암호화할 수 있습니다. 자세한 내용은 Amazon EBS 스냅샷 복사 (p. 934) 단원을 참조하십시오.

암호화되지 않은 스냅샷(활성화된 암호화 기본 제공) 복사

암호화를 기본적으로 활성화한 경우 암호화되지 않은 스냅샷의 복사가 필수이며 사용자의 기본 CMK가 사용된다면 암호화 파라미터는 필요하지 않습니다. 다음 다이어그램에서 기본 사례를 보여 줍니다.

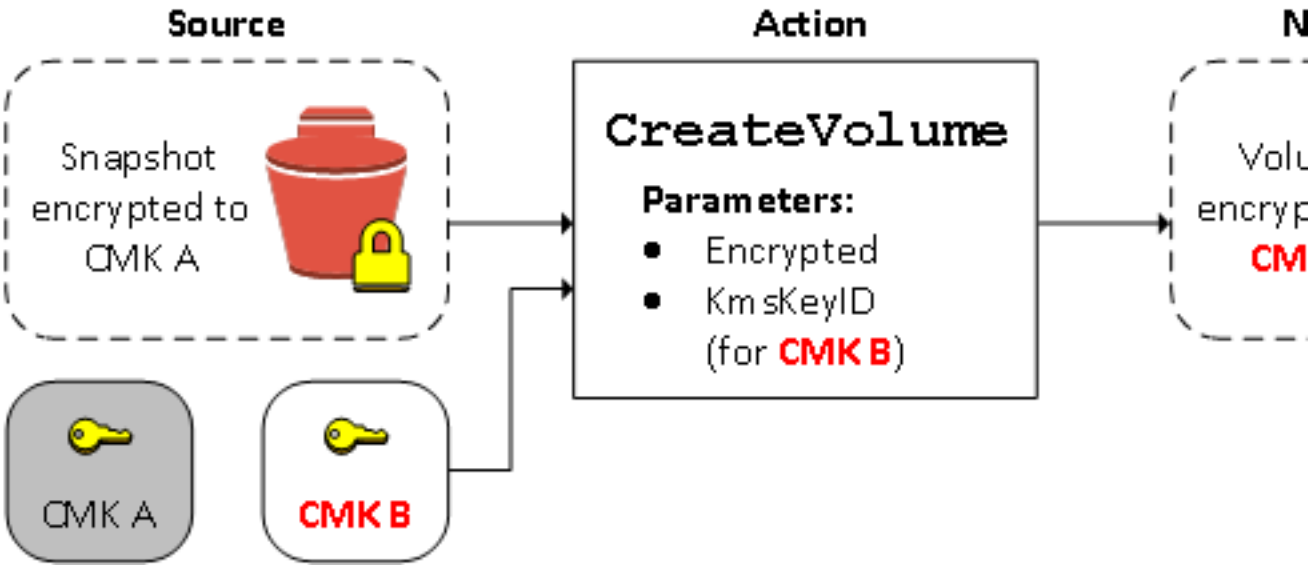


Note

새 CMK로 스냅샷을 복사 및 암호화하면 항상 전체(증분 아님) 복사가 생성되고 추가 지연 및 스토리지 비용이 발생합니다.

암호화된 볼륨의 재암호화

암호화된 스냅샷에서 `CreateVolume` 작업을 수행할 때 다른 CMK를 통해 재암호화하는 옵션이 있습니다. 다음 다이어그램에서 프로세스를 보여 줍니다. 이 예에서는 CMK A와 CMK B, 2개의 CMK가 있습니다. 원본 스냅샷은 CMK A로 암호화됩니다. 볼륨을 생성하는 동안 파라미터로 지정된 CMK B의 키 ID를 통해 원본 데이터가 자동으로 암호 해독된 다음 CMK B로 재암호화됩니다.



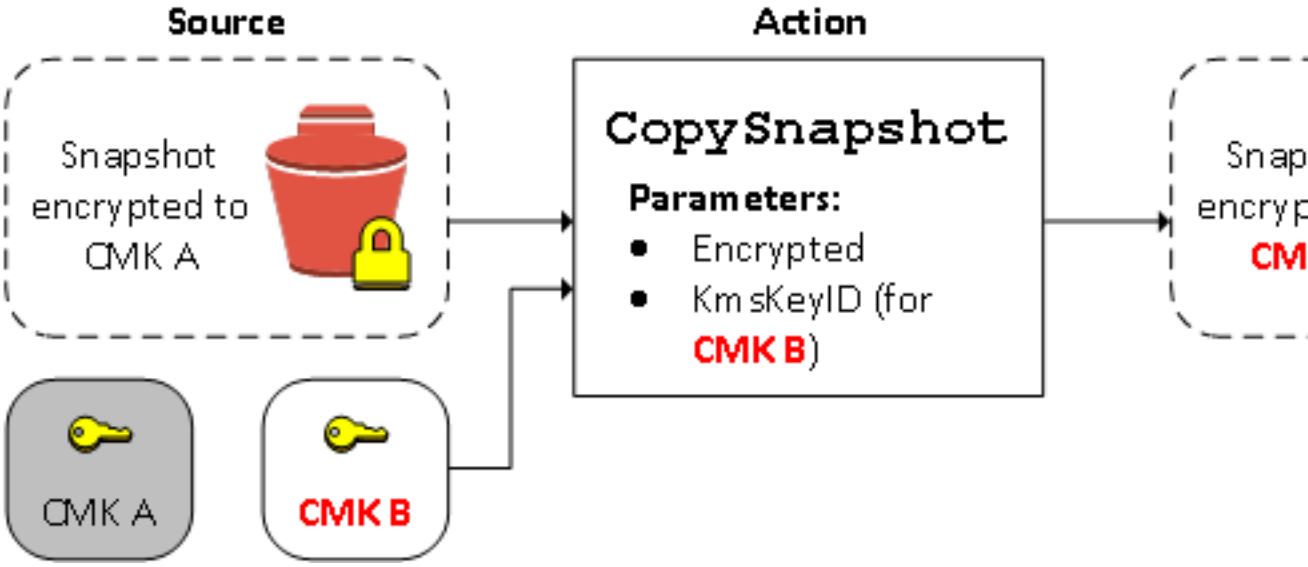
Note

새 CMK로 스냅샷을 복사 및 암호화하면 항상 전체(증분 아님) 복사가 생성되고 추가 지연 및 스토리지 비용이 발생합니다.

자세한 내용은 스냅샷에서 Amazon EBS 볼륨 복구 (p. 909) 단원을 참조하십시오.

암호화된 볼륨의 재암호화

복사 중에 스냅샷을 암호화하는 기능을 사용하여 자신이 소유하고 있는 이미 암호화된 스냅샷에 새 대칭 CMK를 적용할 수 있습니다. 새 CMK를 사용하여 결과 복사본에서 복원된 볼륨에만 액세스할 수 있습니다. 다음 다이어그램에서 프로세스를 보여 줍니다. 이 예에서는 CMK A와 CMK B, 2개의 CMK가 있습니다. 원본 스냅샷은 CMK A로 암호화됩니다. 복사하는 동안 파라미터로 지정된 CMK B의 키 ID를 통해 원본 데이터가 자동으로 CMK B로 재암호화됩니다.



Note

새 CMK로 스냅샷을 복사 및 암호화하면 항상 전체(증분 아님) 복사가 생성되고 추가 지연 및 스토리지 비용이 발생합니다.

관련된 시나리오에서 자신과 공유된 스냅샷의 복사본에 새 암호화 파라미터를 적용하도록 선택할 수도 있습니다. 기본적으로 이 복사본은 스냅샷의 소유자가 공유하는 CMK로 암호화됩니다. 하지만 복사 프로세스 중에 자신이 관리하는 다른 CMK를 사용하여 공유 스냅샷의 복사본을 만드는 게 좋습니다. 이를 통해 원래 CMK가 손상되거나 소유자가 어떤 이유로든 CMK를 취소하는 경우 볼륨에 대한 액세스 권한을 보호할 수 있습니다. 자세한 내용은 암호화 및 스냅샷 복사 (p. 935) 단원을 참조하십시오.

암호화된 볼륨과 암호화되지 않은 볼륨 간 데이터 마이그레이션

암호화된 볼륨과 암호화되지 않은 볼륨에 모두 액세스할 수 있는 경우, 둘 사이에서 자유롭게 데이터를 전송할 수 있습니다. EC2는 암호화 및 복호화 작업을 투명하게 수행합니다.

예를 들어 rsync 명령을 사용하여 데이터를 복사합니다. 다음 명령에서 소스 데이터는 `/mnt/source`에 있고 대상 볼륨은 `/mnt/destination`에 마운트되어 있습니다.

```
[ec2-user ~]$ sudo rsync -avh --progress /mnt/source/ /mnt/destination/
```

암호화 결과

다음 표에서는 가능한 각 설정 조합에 대한 암호화 결과를 설명합니다.

암호화를 활성화합니까?	암호화를 기본 설정합니까?	볼륨의 소스	기본값(CMK가 지정되지 않음)	사용자 지정(CMK가 지정됨)
아니요	아니요	새(빈) 볼륨	암호화되지 않음	해당 사항 없음

암호화를 활성화합니까?	암호화를 기본 설정합니까?	볼륨의 소스	기본값(CMK가 지정되지 않음)	사용자 지정(CMK가 지정됨)
아니요	아니요	암호화되지 않은 소유 스냅샷	암호화되지 않음	
아니요	아니요	암호화된 소유 스냅샷	동일한 키로 암호화됨	
아니요	아니요	암호화되지 않은 공유 스냅샷	암호화되지 않음	
아니요	아니요	암호화된 공유 스냅샷	기본 CMK로 암호화됨*	
예	아니요	새 볼륨	기본 CMK로 암호화됨	지정된 CMK로 암호화됨**
예	아니요	암호화되지 않은 소유 스냅샷	기본 CMK로 암호화됨	
예	아니요	암호화된 소유 스냅샷	동일한 키로 암호화됨	
예	아니요	암호화되지 않은 공유 스냅샷	기본 CMK로 암호화됨	
예	아니요	암호화된 공유 스냅샷	기본 CMK로 암호화됨	
아니요	예	새(빈) 볼륨	기본 CMK로 암호화됨	해당 사항 없음
아니요	예	암호화되지 않은 소유 스냅샷	기본 CMK로 암호화됨	
아니요	예	암호화된 소유 스냅샷	동일한 키로 암호화됨	
아니요	예	암호화되지 않은 공유 스냅샷	기본 CMK로 암호화됨	
아니요	예	암호화된 공유 스냅샷	기본 CMK로 암호화됨	
예	예	새 볼륨	기본 CMK로 암호화됨	지정된 CMK로 암호화됨
예	예	암호화되지 않은 소유 스냅샷	기본 CMK로 암호화됨	
예	예	암호화된 소유 스냅샷	동일한 키로 암호화됨	
예	예	암호화되지 않은 공유 스냅샷	기본 CMK로 암호화됨	
예	예	암호화된 공유 스냅샷	기본 CMK로 암호화됨	

* 이는 AWS 계정 및 리전의 EBS 암호화에 사용되는 기본 CMK입니다. 기본적으로 EBS에 대한 고유한 AWS 관리형 CMK이거나 고객 관리형 CMK를 지정할 수 있습니다. 자세한 내용은 EBS 암호화에 대한 기본 키 (p. 970) 단원을 참조하십시오.

** 이는 실행 시 볼륨용으로 지정된 고객 관리형 CMK입니다. 이 CMK는 AWS 계정 및 리전에 대한 기본 CMK 대신 사용됩니다.

API 및 CLI를 사용한 암호화 기본값 설정

다음 API 작업 및 CLI 명령을 사용하여 암호화 기본 제공과 기본 고객 마스터 키(CMK)를 관리할 수 있습니다.

API 작업	CLI 명령	설명
DisableEbsEncryptionByDefault	disable-ebs-encryption-by-default	암호화를 기본적으로 비활성화합니다.
EnableEbsEncryptionByDefault	enable-ebs-encryption-by-default	암호화를 기본적으로 활성화합니다.
GetEbsDefaultKmsKeyId	get-ebs-default-kms-key-id	기본 CMK를 설명합니다.
GetEbsEncryptionByDefault	get-ebs-encryption-by-default	암호화 기본 제공을 활성화할지 여부를 나타냅니다.
ModifyEbsDefaultKmsKeyId	modify-ebs-default-kms-key-id	EBS 볼륨을 암호화하는 데 사용되는 기본 CMK를 변경합니다.
ResetEbsDefaultKmsKeyId	reset-ebs-default-kms-key-id	AWS 관리형 기본 CMK를 EBS 볼륨을 암호화하는 데 사용되는 기본 CMK로 재설정합니다.

Amazon EBS 빠른 스냅샷 복원

Amazon EBS 빠른 스냅샷 복원을 사용하면 생성 시 완전히 초기화된 스냅샷에서 볼륨을 생성할 수 있습니다. 이렇게 하면 처음 액세스할 때 블록에 대한 I/O 작업 지연 시간이 없어집니다. 빠른 스냅샷 복원을 사용하여 생성된 볼륨은 프로비저닝 된 모든 성능을 즉시 제공합니다.

시작하려면 특정 가용 영역의 특정 스냅샷에 대해 빠른 스냅샷 복원을 활성화합니다. 각 스냅샷 및 가용 영역 페어는 하나의 빠른 스냅샷 복원을 나타냅니다. 리전당 최대 50개의 빠른 스냅샷 복원을 활성화할 수 있습니다. 활성화된 가용 영역 중 하나에 있는 이러한 스냅샷 중 하나에서 볼륨을 생성하면 빠른 스냅샷 복원을 사용하여 볼륨이 복원됩니다.

목차

- 빠른 스냅샷 복원 상태 (p. 979)
- 볼륨 생성 크레딧 (p. 979)
- 빠른 스냅샷 복원 관리 (p. 979)
- 빠른 스냅샷 복원이 활성화된 스냅샷 보기 (p. 980)
- 빠른 스냅샷 복원을 사용하여 복원된 볼륨 보기 (p. 980)

빠른 스냅샷 복원 상태

스냅샷에 대해 빠른 스냅샷 복원을 활성화하면 스냅샷이 다음 상태 중 하나일 수 있습니다.

- `enabling` — 빠른 스냅샷 복원 활성화를 요청했습니다.
- `optimizing` — 빠른 스냅샷 복원이 활성화되고 있습니다. 스냅샷을 최적화하려면 TiB당 60분이 소요됩니다.
- `enabled` — 빠른 스냅샷 복원이 활성화되었습니다.
- `disabling` — 빠른 스냅샷 복원을 비활성화하도록 요청했거나 빠른 스냅샷 복원을 활성화하도록 요청한 요청이 실패했습니다.
- `disabled` — 빠른 스냅샷 복원이 비활성화되었습니다. 필요에 따라 빠른 스냅샷 복원을 다시 활성화할 수 있습니다.

볼륨 생성 크레딧

빠른 스냅샷 복원의 최대 성능 이점을 얻는 볼륨 수는 스냅샷에 대한 볼륨 생성 크레딧에 의해 결정됩니다. 가용 영역당 스냅샷별로 하나의 크레딧 버킷이 있습니다. 빠른 스냅샷 복원이 활성화된 스냅샷에서 생성하는 각 볼륨은 크레딧 버킷에서 하나의 크레딧을 사용합니다.

크레딧 버킷의 크기는 스냅샷에서 생성된 볼륨의 크기가 아니라 스냅 샷의 크기에 따라 다릅니다. 각 스냅샷의 크레딧 버킷 크기는 다음과 같이 계산됩니다.

```
MAX (1, MIN (10, FLOOR(1024/snapshot_size_gib)))
```

크레딧을 사용하면 크레딧 버킷은 시간 경과에 따라 재충전됩니다. 각 크레딧 버킷의 재충전 비율은 다음과 같이 계산됩니다.

```
MIN (10, 1024/snapshot_size_gib)
```

예를 들어 크기가 100GiB인 스냅샷에 대해 빠른 스냅샷 복원을 활성화하는 경우 크레딧 버킷의 최대 크기는 10 크레딧이며 재충전 비율은 시간당 10 크레딧입니다. 크레딧 버킷이 가득차면 이 스냅샷에서 10개의 초기화된 볼륨을 동시에 생성할 수 있습니다.

Cloudwatch 지표를 사용하여 크레딧 버킷의 크기와 각 버킷에서 사용할 수 있는 크레딧의 수를 모니터링할 수 있습니다. 자세한 내용은 빠른 스냅샷 복원 지표 (p. 1015) 단원을 참조하십시오.

빠른 스냅샷 복원이 활성화된 스냅샷에서 볼륨을 생성한 후, describe-volumes를 사용하여 볼륨을 설명하고 출력에서 `fastRestored` 필드를 확인하여 볼륨이 빠른 스냅샷 복원을 사용하여 초기화된 볼륨으로 생성되었는지 여부를 확인할 수 있습니다.

빠른 스냅샷 복원 관리

다음 절차를 수행하여 스냅샷에 대해 빠른 스냅샷 복원을 활성화합니다. 스냅샷을 소유하고 있어야 합니다. 공유된 스냅샷에 대해서는 빠른 스냅샷 복원을 활성화할 수 없습니다.

빠른 스냅샷 복원을 활성화하거나 비활성화하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Snapshots(스냅샷)을 선택합니다.
3. 스냅샷을 선택합니다.
4. Actions(작업), Manage Fast Snapshot Restore(빠른 스냅샷 관리)를 선택합니다.
5. 가용 영역을 선택하거나 선택 취소한 다음 Save(저장)를 선택합니다.
6. 활성화된 경우 빠른 스냅샷 복원의 상태를 추적하려면 Description(설명) 탭의 Fast Snapshot Restore(빠른 스냅샷 복원)를 참조하십시오.

AWS CLI를 사용하여 빠른 스냅샷 복원을 관리하려면

- enable-fast-snapshot-restores
- disable-fast-snapshot-restores
- describe-fast-snapshot-restores

빠른 스냅샷 복원이 활성화된 스냅샷 보기

다음 절차를 수행하여 스냅샷에 대한 빠른 스냅샷 복원 상태를 확인합니다.

콘솔을 사용하여 빠른 스냅샷 복원 상태를 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Snapshots(스냅샷)을 선택합니다.
3. 스냅샷을 선택합니다.
4. Description(설명) 탭에서 Fast Snapshot Restore(빠른 스냅샷 복원)를 확인합니다. 빠른 스냅샷 복원의 상태가 나타나 있습니다. 예를 들면 “2 Availability Zones optimizing(2개의 가용 영역 최적화 중)” 또는 “2 Availability Zones enabled(2개의 가용 영역 활성화됨)”와 같이 표시됩니다.

AWS CLI를 사용하여 빠른 스냅샷 복원이 활성화된 스냅샷을 보려면

빠른 스냅샷 복원이 활성화된 스냅샷을 확인하려면 describe-fast-snapshot-restores 명령을 사용합니다.

```
aws ec2 describe-fast-snapshot-restores --filters Name=state,Values=enabled
```

다음은 예제 출력입니다.

```
{
    "FastSnapshotRestores": [
        {
            "SnapshotId": "snap-0e946653493cb0447",
            "AvailabilityZone": "us-east-2a",
            "State": "enabled",
            "StateTransitionReason": "Client.UserInitiated - Lifecycle state transition",
            "OwnerId": "123456789012",
            "EnablingTime": "2020-01-25T23:57:49.596Z",
            "OptimizingTime": "2020-01-25T23:58:25.573Z",
            "EnabledTime": "2020-01-25T23:59:29.852Z"
        },
        {
            "SnapshotId": "snap-0e946653493cb0447",
            "AvailabilityZone": "us-east-2b",
            "State": "enabled",
            "StateTransitionReason": "Client.UserInitiated - Lifecycle state transition",
            "OwnerId": "123456789012",
            "EnablingTime": "2020-01-25T23:57:49.596Z",
            "OptimizingTime": "2020-01-25T23:58:25.573Z",
            "EnabledTime": "2020-01-25T23:59:29.852Z"
        }
    ]
}
```

빠른 스냅샷 복원을 사용하여 복원된 볼륨 보기

볼륨의 가용 영역에서 빠른 스냅샷 복원이 활성화된 스냅샷을 기반으로 볼륨을 생성하면 빠른 스냅샷 복원을 사용하여 볼륨이 복원됩니다.

빠른 스냅샷 복원이 활성화된 스냅샷을 기반으로 생성된 볼륨을 확인하려면 describe-volumes 명령을 사용합니다.

```
aws ec2 describe-volumes --filters Name=fast-restored,Values=true
```

다음은 예제 출력입니다.

```
{
    "Volumes": [
        {
            "Attachments": [],
            "AvailabilityZone": "us-east-2a",
            "CreateTime": "2020-01-26T00:34:11.093Z",
            "Encrypted": true,
            "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/8c5b2c63-b9bc-45a3-
a87a-5513e232e843",
            "Size": 20,
            "SnapshotId": "snap-0e946653493cb0447",
            "State": "available",
            "VolumeId": "vol-0d371921d4ca797b0",
            "Iops": 100,
            "VolumeType": "gp2",
            "FastRestored": true
        }
    ]
}
```

Linux 인스턴스의 Amazon EBS 및 NVMe

EBS 볼륨이 Nitro 기반 인스턴스 (p. 181)에 NVMe 블록 디바이스로 표시됩니다. 디바이스 이름은 `/dev/nvme0n1`, `/dev/nvme1n1` 등입니다. 블록 디바이스 매핑에서 사용자가 지정하는 디바이스 이름은 NVMe 디바이스 이름(`/dev/nvme[0-26]n1`)을 이용해 바뀝니다. 블록 디바이스 드라이버는 블록 디바이스 매핑에서 볼륨에 대해 지정된 순서와는 다른 순서로 NVMe 디바이스 이름을 할당할 수 있습니다.

> **Note**
>
> 블록과 디바이스 간의 인터페이스가 무엇이든 간에 Amazon EBS 제품 세부 정보에 명시된 EBS 성능 보장은 유효합니다.

내용

- NVMe 드라이버 설치 또는 업그레이드 (p. 981)
- EBS 디바이스 식별 (p. 982)
- NVMe EBS 볼륨 작업 (p. 984)
- I/O 작업 시간제한 (p. 984)

NVMe 드라이버 설치 또는 업그레이드

NVMe 볼륨에 액세스하려면 NVMe 드라이버가 설치되어 있어야 합니다. 인스턴스는 NVMe EBS 볼륨, NVMe 인스턴스 스토어 볼륨, 두 NVMe 볼륨 유형 모두 또는 NVMe 볼륨 없음을 지원합니다. 자세한 내용은 네트워킹 및 스토리지 기능 요약 (p. 183)을 참조하십시오.

다음 AMI는 필수 NVMe 드라이버를 포함합니다.

- Amazon Linux 2
- Amazon Linux AMI 2018.03
- Ubuntu 14.04(`linux-aws` 커널) 이상

- Red Hat Enterprise Linux 7.4 이상
- SUSE Linux Enterprise Server 12 SP2 이상
- CentOS 7.4.1708 이상
- FreeBSD 11.1 이상
- Debian GNU/Linux 9 이상

Windows 인스턴스의 NVMe 드라이버에 대한 자세한 내용은 Windows 인스턴스용 Amazon EC2 사용 설명서의 Windows 인스턴스의 Amazon EBS 및 NVMe를 참조하십시오.

인스턴스에 NVMe 드라이버가 있는지 확인

다음 명령을 사용하여 인스턴스에 NVMe 드라이버가 있는지 확인하고 드라이버 버전을 확인할 수 있습니다.

```
$ modinfo nvme
```

인스턴스에 NVMe 드라이버가 있는 경우 명령은 드라이버에 대한 정보를 반환합니다.

NVMe 드라이버 업데이트

인스턴스에 NVMe 드라이버가 있는 경우 다음 절차에 따라 드라이버를 최신 버전으로 업데이트할 수 있습니다.

1. 인스턴스에 연결합니다.
2. 패키지 캐시를 업데이트하고 다음과 같이 필요한 패키지 업데이트를 가져옵니다.
   - Amazon Linux 2, Amazon Linux, CentOS 및 Red Hat Enterprise Linux:

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

   - Ubuntu 및 Debian:

     ```
     [ec2-user ~]$ sudo apt-get update -y
     ```

3. Ubuntu 16.04 이상에는 Nitro 기반 인스턴스에 필요한 NVMe 및 ENA 드라이버가 포함된 `linux-aws` 패키지가 포함되어 있습니다. `linux-aws` 패키지를 업그레이드하여 다음과 같이 최신 버전을 받습니다.

   ```
   [ec2-user ~]$ sudo apt-get install --only-upgrade -y linux-aws
   ```

   Ubuntu 14.04의 경우 다음과 같이 최신 `linux-aws` 패키지를 설치할 수 있습니다.

   ```
   [ec2-user ~]$ sudo apt-get install linux-aws
   ```

4. 인스턴스를 재부팅하여 최신 커널 버전을 로드합니다.

   ```
   sudo reboot
   ```

5. 재부팅이 끝난 후 인스턴스에 다시 연결합니다.

EBS 디바이스 식별

EBS는 단일 루트 I/O 가상화(SR-IOV)를 사용하여 NVMe 사양을 사용하는 Nitro 기반 인스턴스에서 볼륨 연결을 제공합니다. 이러한 디바이스는 운영 체제의 표준 NVMe 드라이버에 의존합니다. 이러한 드라이버는

일반적으로 인스턴스 부팅 중에 PCI 버스를 스캔하여 연결된 디바이스를 검색한 다음, 블록 디바이스 매핑에서 디바이스가 지정되는 방식이 아닌, 디바이스가 응답하는 순서에 기초하여 디바이스 노드를 생성합니다. Linux에서 NVMe 디바이스 이름은 `/dev/nvme<x>n<y>` 패턴을 따릅니다. <x>는 열거 순서이며, EBS의 경우, <y>는 1입니다. 경우에 따라 디바이스는 후속 인스턴스가 시작되는 것과 다른 순서로 검색에 응답하기도 하는데, 이로 인해 디바이스 이름이 변경됩니다.

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

- Nitro 기반 인스턴스의 경우, EBS 볼륨을 연결하거나 `AttachVolume` 또는 `RunInstances` API 호출이 NVMe 컨트롤러 식별의 벤더별 데이터 필드에서 캡처되는 동안 Amazon EC2 콘솔에 지정된 블록 디바이스 매핑. 2017.09.01 버전 이후의 Amazon Linux AMI를 사용하는 경우, 이 데이터를 읽고 블록 디바이스 매핑의 심볼 링크를 생성하는 `udev` 규칙이 제공됩니다.
- NVMe EBS 볼륨에는 디바이스 식별에서 일련 번호로 설정된 EBS 볼륨 ID가 있습니다.
- 디바이스를 포맷할 때 파일 시스템 수명 기간 동안 지속되는 UUID가 생성됩니다. 이와 동시에 디바이스 레이블을 지정할 수 있습니다. 자세한 내용은 Amazon EBS 볼륨을 Linux에서 사용할 수 있도록 만들기 (p. 914) 및 잘못된 볼륨에서 부팅 (p. 1117) 단원을 참조하십시오.

Amazon Linux AMI

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

```
[ec2-user ~]$ sudo /sbin/ebsnvme-id /dev/nvme1n1
Volume ID: vol-01324f611e2463981
/dev/sdf
```

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

기타 Linux AMI

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

다음은 볼륨 ID와 디바이스 이름을 가져오는 예입니다. NVMe 컨트롤러 벤더별 확장자(컨트롤러 식별의 바이트 384:4095)를 통해 디바이스 이름을 구할 수 있습니다.

```
[ec2-user ~]$ sudo nvme id-ctrl -v /dev/nvme1n1
NVME Identify Controller:
vid     : 0x1d0f
ssvid   : 0x1d0f
sn      : vol01234567890abcdef
mn      : Amazon Elastic Block Store
...
0000: 2f 64 65 76 2f 73 64 6a 20 20 20 20 20 20 20 20 "/dev/sdf..."
```

lsblk 명령은 사용 가능한 디바이스와 각각의 마운트 지점(해당되는 경우)을 나열합니다. 그러면 사용할 올바른 디바이스 이름을 판단할 수 있습니다. 이 예에서 `/dev/nvme0n1p1`은 루트 디바이스에 마운트되어 있고 `/dev/nvme1n1`은 연결되었지만 마운트되어 있지 않습니다.

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

NVMe EBS 볼륨 작업

NVMe EBS 볼륨을 포맷하고 탑재하려면 Amazon EBS 볼륨을 Linux에서 사용할 수 있도록 만들기 (p. 914)을 참조하십시오.

Linux kernel 4.2 이상을 사용 중이라면 NVMe EBS 볼륨의 볼륨 크기를 변경할 경우 인스턴스에 자동으로 반영됩니다. 그 전의 Linux 커널에서는 EBS 볼륨을 분리하고 연결하거나 인스턴스를 재부팅해야 크기 변경이 반영될 수 있습니다. Linux 커널 3.19 이상에서는 다음과 같이 hdparm 명령을 실행하여 NVMe 디바이스를 강제로 재스캔할 수 있습니다.

```
[ec2-user ~]$ sudo hdparm -z /dev/nvme1n1
```

NVMe EBS 볼륨을 분리할 때 인스턴스가 볼륨을 분리하기 전에 파일 시스템 캐시 또는 메타데이터를 플러시할 기회가 없습니다. 따라서 NVMe EBS 볼륨을 분리하기 전에 먼저 볼륨을 동기화하고 마운트를 해제해야 합니다. 볼륨이 분리되지 않으면 인스턴스에서 Amazon EBS 볼륨 분리 (p. 924)의 설명대로 `force-detach` 명령을 시도할 수 있습니다.

I/O 작업 시간제한

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

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

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

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

Amazon EBS 최적화 인스턴스

Amazon EBS 최적화 인스턴스는 최적화된 구성 스택을 사용하며 Amazon 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 제공합니다. EBS 최적화 인스턴스에 연결된 경우 범용 SSD(`gp2`) 볼륨은 99%의 시간 동안 기본 성능 및 버스트 성능을 제공하도록 설계되며, 프로비저닝된 IOPS SSD(`io1`) 볼륨은 99.9%의 시간 동안 프로비저닝 성능을 제공하도록 설계됩니다. 처리량에 최적화된 HDD(`st1`) 및 Cold HDD(`sc1`)는 모두 99%의 기간 동안 90%의 버스트 처리량에 대해 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. 892) 단원을 참조하십시오.

목차

- 지원되는 인스턴스 유형 (p. 985)
- 시작 시 EBS 최적화 활성화 (p. 996)
- 실행 중인 인스턴스에 대해 EBS 최적화 활성화 (p. 996)

지원되는 인스턴스 유형

다음 표에서는 어떤 인스턴스 유형이 EBS 최적화를 지원하는지를 보여 줍니다. 여기에는 Amazon EBS에 대한 전용 대역폭, 스트리밍 읽기 워크로드 및 128KiB I/O 크기로 해당 연결에서 달성할 수 있는 일반적인 최대 집계 처리량, 16KiB I/O 크기를 사용할 경우 인스턴스가 지원할 수 있는 IOPS 최대량이 포함됩니다. 애플리케이션에 필요한 것보다 많은 전용 Amazon EBS 처리량을 제공하는 EBS에 최적화된 인스턴스를 선택해야 합니다. 그렇게 하지 않으면 Amazon EBS와 Amazon EC2 간의 연결이 성능 병목 현상으로 변할 수 있습니다.

기본적으로 EBS에 최적화됨

다음 표에는 EBS 최적화를 지원하고 EBS 최적화가 기본적으로 활성화되는 인스턴스 유형이 나열됩니다. EBS 최적화를 활성화할 필요가 없으며 EBS 최적화를 비활성화해도 효과가 없습니다.

인스턴스 크기	최대 대역폭(Mbps)	최대 처리량(MB/s, 128KiB I/O)	최대 IOPS(16KiB I/O)
`a1.medium` *	3,500	437.5	20,000건
`a1.large` *	3,500	437.5	20,000건
`a1.xlarge` *	3,500	437.5	20,000건
`a1.2xlarge` *	3,500	437.5	20,000건
`a1.4xlarge`	3,500	437.5	20,000건
`c4.large`	500	62.5	4,000
`c4.xlarge`	750	93.75	6,000
`c4.2xlarge`	1,000	125	8,000
`c4.4xlarge`	2,000건	250	16,000
`c4.8xlarge`	4,000	500	32,000
`c5.large` *	4,750	593.75	20,000건
`c5.xlarge` *	4,750	593.75	20,000건
`c5.2xlarge` *	4,750	593.75	20,000건
`c5.4xlarge`	4,750	593.75	20,000건
`c5.9xlarge`	9,500	1,187.5	40,000
`c5.12xlarge`	9,500	1,187.5	40,000
`c5.18xlarge`	19,000	2,375	80,000
`c5.24xlarge`	19,000	2,375	80,000
`c5.metal`	19,000	2,375	80,000
`c5d.large` *	4,750	593.75	20,000건
`c5d.xlarge` *	4,750	593.75	20,000건
`c5d.2xlarge` *	4,750	593.75	20,000건

인스턴스 크기	최대 대역폭(Mbps)	최대 처리량(MB/s, 128KiB I/O)	최대 IOPS(16KiB I/O)
c5d.4xlarge	4,750	593.75	20,000건
c5d.9xlarge	9,500	1,187.5	40,000
c5d.12xlarge	9,500	1,187.5	40,000
c5d.18xlarge	19,000	2,375	80,000
c5d.24xlarge	19,000	2,375	80,000
c5d.metal	19,000	2,375	80,000
c5n.large *	4,750	593.75	20,000건
c5n.xlarge *	4,750	593.75	20,000건
c5n.2xlarge *	4,750	593.75	20,000건
c5n.4xlarge	4,750	593.75	20,000건
c5n.9xlarge	9,500	1,187.5	40,000
c5n.18xlarge	19,000	2,375	80,000
c5n.metal	19,000	2,375	80,000
d2.xlarge	750	93.75	6,000
d2.2xlarge	1,000	125	8,000
d2.4xlarge	2,000건	250	16,000
d2.8xlarge	4,000	500	32,000
f1.2xlarge	1,700	212.5	12,000
f1.4xlarge	3,500	437.5	44,000
f1.16xlarge	14,000	1,750	75,000
g3s.xlarge	850	106.25	5,000
g3.4xlarge	3,500	437.5	20,000건
g3.8xlarge	7,000	875	40,000
g3.16xlarge	14,000	1,750	80,000
g4dn.xlarge *	3,500	437.5	20,000건
g4dn.2xlarge *	3,500	437.5	20,000건
g4dn.4xlarge	4,750	593.75	20,000건
g4dn.8xlarge	9,500	1,187.5	40,000
g4dn.12xlarge	9,500	1,187.5	40,000
g4dn.16xlarge	9,500	1,187.5	40,000

인스턴스 크기	최대 대역폭(Mbps)	최대 처리량(MB/s, 128KiB I/O)	최대 IOPS(16KiB I/O)
h1.2xlarge	1,750	218.75	12,000
h1.4xlarge	3,500	437.5	20,000건
h1.8xlarge	7,000	875	40,000
h1.16xlarge	14,000	1,750	80,000
i3.large	425	53.13	3000
i3.xlarge	850	106.25	6000
i3.2xlarge	1,700	212.5	12,000
i3.4xlarge	3,500	437.5	16,000
i3.8xlarge	7,000	875	32,500
i3.16xlarge	14,000	1,750	65,000
i3.metal	19,000	2,375	80,000
i3en.large *	4,750	593.75	20,000건
i3en.xlarge *	4,750	593.75	20,000건
i3en.2xlarge *	4,750	593.75	20,000건
i3en.3xlarge *	4,750	593.75	20,000건
i3en.6xlarge	4,750	593.75	20,000건
i3en.12xlarge	9,500	1,187.5	40,000
i3en.24xlarge	19,000	2,375	80,000
i3en.metal	19,000	2,375	80,000
m4.large	450	56.25	3,600
m4.xlarge	750	93.75	6,000
m4.2xlarge	1,000	125	8,000
m4.4xlarge	2,000건	250	16,000
m4.10xlarge	4,000	500	32,000
m4.16xlarge	10,000개	1,250	65,000
m5.large *	4,750	593.75	18,750
m5.xlarge *	4,750	593.75	18,750
m5.2xlarge *	4,750	593.75	18,750
m5.4xlarge	4,750	593.75	18,750
m5.8xlarge	6,800	850	30,000개

인스턴스 크기	최대 대역폭(Mbps)	최대 처리량(MB/s, 128KiB I/O)	최대 IOPS(16KiB I/O)
`m5.12xlarge`	9,500	1,187.5	40,000
`m5.16xlarge`	13,600	1,700	60,000
`m5.24xlarge`	19,000	2,375	80,000
`m5.metal`	19,000	2,375	80,000
`m5a.large` *	2,880	360	16,000
`m5a.xlarge` *	2,880	360	16,000
`m5a.2xlarge` *	2,880	360	16,000
`m5a.4xlarge`	2,880	360	16,000
`m5a.8xlarge`	4,750	593.75	20,000건
`m5a.12xlarge`	6,780	847.5	30,000개
`m5a.16xlarge`	9,500	1,187.50	40,000
`m5a.24xlarge`	13,570	1,696.25	60,000
`m5ad.large` *	2,880	360	16,000
`m5ad.xlarge` *	2,880	360	16,000
`m5ad.2xlarge` *	2,880	360	16,000
`m5ad.4xlarge`	2,880	360	16,000
`m5ad.8xlarge`	4,750	593.75	20,000건
`m5ad.12xlarge`	6,780	847.5	30,000개
`m5ad.16xlarge`	9,500	1,187.5	40,000
`m5ad.24xlarge`	13,570	1,696.25	60,000
`m5d.large` *	4,750	593.75	18,750
`m5d.xlarge` *	4,750	593.75	18,750
`m5d.2xlarge` *	4,750	593.75	18,750
`m5d.4xlarge`	4,750	593.75	18,750
`m5d.8xlarge`	6,800	850	30,000개
`m5d.12xlarge`	9,500	1,187.5	40,000
`m5d.16xlarge`	13,600	1,700	60,000
`m5d.24xlarge`	19,000	2,375	80,000
`m5d.metal`	19,000	2,375	80,000
`m5dn.large` *	4,750	593.75	18,750

인스턴스 크기	최대 대역폭(Mbps)	최대 처리량(MB/s, 128KiB I/O)	최대 IOPS(16KiB I/O)
`m5dn.xlarge` *	4,750	593.75	18,750
`m5dn.2xlarge` *	4,750	593.75	18,750
`m5dn.4xlarge`	4,750	593.75	18,750
`m5dn.8xlarge`	6,800	850	30,000개
`m5dn.12xlarge`	9,500	1,187.5	40,000
`m5dn.16xlarge`	13,600	1,700	60,000
`m5dn.24xlarge`	19,000	2,375	80,000
`m5n.large` *	4,750	593.75	18,750
`m5n.xlarge` *	4,750	593.75	18,750
`m5n.2xlarge` *	4,750	593.75	18,750
`m5n.4xlarge`	4,750	593.75	18,750
`m5n.8xlarge`	6,800	850	30,000개
`m5n.12xlarge`	9,500	1,187.5	40,000
`m5n.16xlarge`	13,600	1,700	60,000
`m5n.24xlarge`	19,000	2,375	80,000
`p2.xlarge`	750	93.75	6,000
`p2.8xlarge`	5,000	625	32,500
`p2.16xlarge`	10,000개	1,250	65,000
`p3.2xlarge`	1,750	218.75	10,000개
`p3.8xlarge`	7,000	875	40,000
`p3.16xlarge`	14,000	1,750	80,000
`p3dn.24xlarge`	19,000	2,375	80,000
`r4.large`	425	53.13	3,000
`r4.xlarge`	850	106.25	6,000
`r4.2xlarge`	1,700	212.5	12,000
`r4.4xlarge`	3,500	437.5	18,750
`r4.8xlarge`	7,000	875	37,500
`r4.16xlarge`	14,000	1,750	75,000
`r5.large` *	4,750	593.75	18,750
`r5.xlarge` *	4,750	593.75	18,750

인스턴스 크기	최대 대역폭(Mbps)	최대 처리량(MB/s, 128KiB I/O)	최대 IOPS(16KiB I/O)
`r5.2xlarge` *	4,750	593.75	18,750
`r5.4xlarge`	4,750	593.75	18,750
`r5.8xlarge`	6,800	850	30,000개
`r5.12xlarge`	9,500	1,187.5	40,000
`r5.16xlarge`	13,600	1,700	60,000
`r5.24xlarge`	19,000	2,375	80,000
`r5.metal`	19,000	2,375	80,000
`r5a.large` *	2,880	360	16,000
`r5a.xlarge` *	2,880	360	16,000
`r5a.2xlarge` *	2,880	360	16,000
`r5a.4xlarge`	2,880	360	16,000
`r5a.8xlarge`	4,750	593.75	20,000건
`r5a.12xlarge`	6,780	847.5	30,000개
`r5a.16xlarge`	9,500	1,187.5	40,000
`r5a.24xlarge`	13,570	1,696.25	60,000
`r5ad.large` *	2,880	360	16,000
`r5ad.xlarge` *	2,880	360	16,000
`r5ad.2xlarge` *	2,880	360	16,000
`r5ad.4xlarge`	2,880	360	16,000
`r5ad.8xlarge`	4,750	593.75	20,000건
`r5ad.12xlarge`	6,780	847.5	30,000개
`r5ad.16xlarge`	9,500	1,187.5	40,000
`r5ad.24xlarge`	13,570	1,696.25	60,000
`r5d.large` *	4,750	593.75	18,750
`r5d.xlarge` *	4,750	593.75	18,750
`r5d.2xlarge` *	4,750	593.75	18,750
`r5d.4xlarge`	4,750	593.75	18,750
`r5d.8xlarge`	6,800	850	30,000개
`r5d.12xlarge`	9,500	1,187.5	40,000
`r5d.16xlarge`	13,600	1,700	60,000

인스턴스 크기	최대 대역폭(Mbps)	최대 처리량(MB/s, 128KiB I/O)	최대 IOPS(16KiB I/O)
r5d.24xlarge	19,000	2,375	80,000
r5d.metal	19,000	2,375	80,000
r5dn.large *	4,750	593.75	18,750
r5dn.xlarge *	4,750	593.75	18,750
r5dn.2xlarge *	4,750	593.75	18,750
r5dn.4xlarge	4,750	593.75	18,750
r5dn.8xlarge	6,800	850	30,000개
r5dn.12xlarge	9,500	1,187.5	40,000
r5dn.16xlarge	13,600	1,700	60,000
r5dn.24xlarge	19,000	2,375	80,000
r5n.large *	4,750	593.75	18,750
r5n.xlarge *	4,750	593.75	18,750
r5n.2xlarge *	4,750	593.75	18,750
r5n.4xlarge	4,750	593.75	18,750
r5n.8xlarge	6,800	850	30,000개
r5n.12xlarge	9,500	1,187.5	40,000
r5n.16xlarge	13,600	1,700	60,000
r5n.24xlarge	19,000	2,375	80,000
t3.nano *	2,085	260.57	11,800
t3.micro *	2,085	260.57	11,800
t3.small *	2,085	260.57	11,800
t3.medium *	2,085	260.57	11,800
t3.large *	2,780	347.5	15,700
t3.xlarge *	2,780	347.5	15,700
t3.2xlarge *	2,780	347.5	15,700
t3a.nano *	2,085	260.57	11,800
t3a.micro *	2,085	260.57	11,800
t3a.small *	2,085	260.57	11,800
t3a.medium *	2,085	260.57	11,800
t3a.large *	2,780	347.5	15,700

인스턴스 크기	최대 대역폭(Mbps)	최대 처리량(MB/s, 128KiB I/O)	최대 IOPS(16KiB I/O)
`t3a.xlarge` *	2,780	347.5	15,700
`t3a.2xlarge` *	2,780	347.5	15,700
`u-6tb1.metal`	19,000	2,375	80,000
`u-9tb1.metal`	19,000	2,375	80,000
`u-12tb1.metal`	19,000	2,375	80,000
`u-18tb1.metal`	28,000	3,500	160,000
`u-24tb1.metal`	28,000	3,500	160,000
`x1.16xlarge`	7,000	875	40,000
`x1.32xlarge`	14,000	1,750	80,000
`x1e.xlarge`	500	62.5	3,700
`x1e.2xlarge`	1,000	125	7,400
`x1e.4xlarge`	1,750	218.75	10,000개
`x1e.8xlarge`	3,500	437.5	20,000건
`x1e.16xlarge`	7,000	875	40,000
`x1e.32xlarge`	14,000	1,750	80,000
`z1d.large` *	3,170	396.25	13,333
`z1d.xlarge` *	3,170	396.25	13,333
`z1d.2xlarge`	3,170	396.25	13,333
`z1d.3xlarge`	4,750	593.75	20,000건
`z1d.6xlarge`	9,500	1,187.5	40,000
`z1d.12xlarge`	19,000	2,375	80,000
`z1d.metal`	19,000	2,375	80,000

2020년 2월 26일 이후에 시작되는 G4dN, I3en, Inf1, M5a, M5ad, R5a, R5ad, T3, T3a 및 Z1d 인스턴스는 기본적으로 위에 나열된 최대 성능을 지원합니다. 2020년 2월 26일 이전에 시작된 인스턴스에 대해 최대 성능을 얻으려면 인스턴스를 중지했다가 시작합니다.

2019년 12월 3일 이후에 시작되는 C5, C5d, C5n, M5, M5d, M5n, M5dn, R5, R5d, R5n, R5dn, P3dn, `u-6tb1.metal`, `u-9tb1.metal` 및 `u-12tb1.metal` 인스턴스는 기본적으로 위에 나열된 최대 성능을 지원합니다. 2019년 12월 3일 이전에 시작된 인스턴스에 대해 최대 성능을 얻으려면 인스턴스를 중지했다가 시작합니다.

* 이러한 인스턴스 유형에서는 24시간마다 최소 한 번씩 30분의 최대 성능을 지원합니다. 30분 넘는 시간 동안 지속적으로 최대 성능이 필요한 워크로드가 있는 경우 다음 표의 기준 성능에 따라 인스턴스 유형을 선택합니다.

인스턴스 크기	기준 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)
a1.medium	300	37.5	2,500
a1.large	525	65.625	4,000
a1.xlarge	800	100	6,000
a1.2xlarge	1,750	218.75	10,000개
c5.large	650	81.25	4,000
c5.xlarge	1,150	143.75	6,000
c5.2xlarge	2,300	287.5	10,000개
c5d.large	650	81.25	4,000
c5d.xlarge	1,150	143.75	6,000
c5d.2xlarge	2,300	287.5	10,000개
c5n.large	650	81.25	4,000
c5n.xlarge	1,150	143.75	6,000
c5n.2xlarge	2,300	287.5	10,000개
g4dn.xlarge	950	118.75	3,000
g4dn.2xlarge	1,150	143.75	6,000
i3en.large	577	72.1	3,000
i3en.xlarge	1,154	144,2	6,000
i3en.2xlarge	2,307	288.39	12,000
i3en.3xlarge	3,800	475	15,000
inf1.xlarge	1,190	148.75	4,000
inf1.2xlarge	1,190	148.75	6,000
m5.large	650	81.25	3,600
m5.xlarge	1,150	143.75	6,000
m5.2xlarge	2,300	287.5	12,000
m5a.large	650	81.25	3,600
m5a.xlarge	1,085	135.63	6,000
m5a.2xlarge	1,580	197.5	8,333
m5ad.large	650	81.25	3,600
m5ad.xlarge	1,085	135.63	6,000
m5ad.2xlarge	1,580	197.5	8,333

인스턴스 크기	기준 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)
m5d.large	650	81.25	3,600
m5d.xlarge	1,150	143.75	6,000
m5d.2xlarge	2,300	287.5	12,000
m5dn.large	650	81.25	3,600
m5dn.xlarge	1,150	143.75	6,000
m5dn.2xlarge	2,300	287.5	12,000
m5n.large	650	81.25	3,600
m5n.xlarge	1,150	143.75	6,000
m5n.2xlarge	2,300	287.5	12,000
r5.large	650	81.25	3,600
r5.xlarge	1,150	143.75	6,000
r5.2xlarge	2,300	287.5	12,000
r5a.large	650	81.25	3,600
r5a.xlarge	1,085	135.63	6,000
r5a.2xlarge	1,580	197.5	8,333
r5ad.large	650	81.25	3,600
r5ad.xlarge	1,085	135.63	6,000
r5ad.2xlarge	1,580	197.5	8,333
r5d.large	650	81.25	3,600
r5d.xlarge	1,150	143.75	6,000
r5d.2xlarge	2,300	287.5	12,000
r5dn.large	650	81.25	3,600
r5dn.xlarge	1,150	143.75	6,000
r5dn.2xlarge	2,300	287.5	12,000
r5n.large	650	81.25	3,600
r5n.xlarge	1,150	143.75	6,000
r5n.2xlarge	2,300	287.5	12,000
t3.nano	43	5.43	250
t3.micro	87	10.86	500
t3.small	174	21.71	1,000

인스턴스 크기	기준 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)
`t3.medium`	347	43.43	2,000건
`t3.large`	695	86.86	4,000
`t3.xlarge`	695	86.86	4,000
`t3.2xlarge`	695	86.86	4,000
`t3a.nano`	45	5.63	250
`t3a.micro`	90	11.25	500
`t3a.small`	175	21.88	1,000
`t3a.medium`	350	43.75	2,000건
`t3a.large`	695	86.86	4,000
`t3a.xlarge`	695	86.86	4,000
`t3a.2xlarge`	695	86.86	4,000
`z1d.large`	800	100	3,333
`z1d.xlarge`	1,580	197.5	6,667

`EBSIOBalance%` 및 `EBSByteBalance%` 지표로 인스턴스 크기를 적절히 설정했는지 판단할 수 있습니다. CloudWatch 콘솔에서 이 지표를 확인하고 사용자가 지정한 임계값에 따라 트리거될 경보를 설정할 수 있습니다. 이 지표는 백분율로 표현됩니다. 일관되게 낮은 균형 백분율을 나타내는 인스턴스는 규모를 늘리기에 적합한 대상입니다. 균형 백분율이 100% 이하로 결코 떨어지지 않는 인스턴스는 규모를 줄이기에 적합한 대상입니다. 자세한 정보는 CloudWatch를 사용해 인스턴스 모니터링하기 (p. 616) 단원을 참조하십시오.

EBS 최적화 지원됨

다음 표에는 EBS 최적화를 지원하지만 EBS 최적화가 기본적으로 활성화되지 않는 인스턴스 유형이 나열됩니다. 이러한 인스턴스를 시작할 때 또는 인스턴스를 실행한 후 EBS 최적화를 활성화할 수 있습니다. 설명된 수준의 성능을 달성하려면 인스턴스에서 EBS 최적화를 활성화해야 합니다. 기본적으로 EBS 최적화되지 않은 인스턴스에 EBS 최적화를 사용하도록 설정할 경우 전용 용량을 위해 소정의 시간당 추가 요금이 청구됩니다. 요금 정보는 온디맨드 인스턴스에 대한 Amazon EC2 요금 페이지에서 EBS 최적화 인스턴스를 참조하십시오.

인스턴스 크기	최대 대역폭(Mbps)	최대 처리량(MB/s, 128KiB I/O)	최대 IOPS(16KiB I/O)
`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
`g2.2xlarge`	1,000	125	8,000
`i2.xlarge`	500	62.5	4,000
`i2.2xlarge`	1,000	125	8,000

인스턴스 크기	최대 대역폭(Mbps)	최대 처리량(MB/s, 128KiB I/O)	최대 IOPS(16KiB I/O)
`i2.4xlarge`	2,000건	250	16,000
`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
`r3.xlarge`	500	62.5	4,000
`r3.2xlarge`	1,000	125	8,000
`r3.4xlarge`	2,000건	250	16,000

`i2.8xlarge`, `c3.8xlarge`, `r3.8xlarge` 인스턴스에는 전용 EBS 대역폭이 없으므로 EBS 최적화에 영향을 미치지 않습니다. 이러한 인스턴스에서 네트워크 트래픽과 Amazon EBS 트래픽은 동일한 10기가비트 네트워크 인터페이스를 공유합니다.

시작 시 EBS 최적화 활성화

EBS 최적화에 대한 속성을 설정하여 인스턴스에 대한 최적화를 활성화할 수 있습니다.

콘솔로 인스턴스 시작 시 Amazon EBS 최적화를 활성화하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 인스턴스 시작을 선택합니다.
3. 1단계: Amazon 머신 이미지(AMI) 선택에서 AMI를 선택합니다.
4. 2단계: 인스턴스 유형 선택에서 Amazon EBS 최적화를 지원하는 인스턴스 유형을 선택합니다.
5. 3단계: 인스턴스 세부 정보 구성에서 필요한 필드 정보를 모두 입력하고 EBS 최적 인스턴스로 시작을 선택합니다. 이전 단계에서 선택한 인스턴스 유형이 Amazon EBS 최적화를 지원하지 않을 경우 이 옵션이 제공되지 않습니다. 선택한 인스턴스 유형이 기본적으로 Amazon EBS 최적화를 지원할 경우에는 이 옵션이 선택되며 선택을 취소할 수 없습니다.
6. 지시에 따라 마법사를 완료하고 인스턴스를 시작합니다.

명령줄을 사용하여 인스턴스 시작 시 EBS 최적화를 활성화하려면

해당 명령과 함께 다음 옵션 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- run-instances(AWS CLI)를 사용한 `--ebs-optimized`
- `-EbsOptimized` with New-EC2Instance (Windows PowerShell용 AWS 도구)

실행 중인 인스턴스에 대해 EBS 최적화 활성화

실행 중인 인스턴스에 대해 Amazon EBS 최적화 인스턴스 속성을 수정하여 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 인스턴스를 클릭하고 해당 인스턴스를 선택합니다.
3. 작업을 클릭하고 인스턴스 상태를 선택한 후 중지를 클릭합니다.

   Warning

   인스턴스를 중지하면 인스턴스 스토어 볼륨의 데이터가 삭제됩니다. 인스턴스 스토어 볼륨의 데이터를 유지하려면 영구 스토리지에 백업하십시오.
4. 확인 대화 상자가 나타나면 예, 중지를 클릭합니다. 인스턴스가 중지하는 데 몇 분 정도 걸릴 수 있습니다.
5. 인스턴스를 선택된 상태에서 작업을 클릭하고 인스턴스 설정을 선택한 후 종료 동작 변경을 클릭합니다.
6. 인스턴스 유형 변경 대화 상자에서 다음 중 하나를 수행하십시오.
   - 해당 인스턴스의 인스턴스 유형이 기본적으로 Amazon EBS 최적화되었을 경우 EBS 최적이 선택되고 이를 변경할 수 없습니다. 해당 인스턴스에 대해 Amazon EBS 최적화가 이미 활성화되었으므로 취소를 선택합니다.
   - 해당 인스턴스의 인스턴스 유형이 Amazon EBS 최적화를 지원할 경우 EBS 최적, 적용을 선택합니다.
   - 해당 인스턴스의 인스턴스 유형이 Amazon EBS 최적화를 지원하지 않을 경우 EBS 최적을 선택할 수 없습니다. 인스턴스 유형에서 Amazon EBS 최적화를 지원하는 인스턴스 유형을 선택하고 EBS 최적, 적용을 선택합니다.
7. 작업, 인스턴스 상태, 시작을 차례로 선택합니다.

명령줄을 사용하여 실행 중인 인스턴스에 대해 EBS 최적화를 활성화하려면

해당 명령과 함께 다음 옵션 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- modify-instance-attribute(AWS CLI)를 사용한 `--ebs-optimized`
- Edit-EC2InstanceAttribute(Windows PowerShell용 AWS 도구)를 사용한 `-EbsOptimized`

Amazon EBS 인스턴스의 Linux 볼륨 성능

I/O 특성, 인스턴스와 볼륨의 구성 등 여러 가지 요인이 Amazon EBS 볼륨에 영향을 끼칠 수 있습니다. Amazon EBS 및 Amazon EC2 제품 세부 정보 페이지의 지침을 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.

Note

EBS 볼륨 유형의 성능에 대한 AWS 업데이트가 기존 볼륨에 즉시 적용되지 않을 수 있습니다. 기존 볼륨의 전체 성능을 확인하려면 먼저 볼륨에 대해 `ModifyVolume` 작업을 수행해야 할 수 있습니다. 자세한 정보는 Linux에서 EBS 볼륨의 크기, IOPS 또는 유형 수정을 참조하십시오.

내용

- Amazon EBS 성능 팁 (p. 998)
- I/O 특성 및 모니터링 (p. 1000)
- Amazon EBS 볼륨 초기화 (p. 1001)
- Linux의 RAID 구성 (p. 1003)
- EBS 볼륨 벤치마크 (p. 1007)

Amazon EBS 성능 팁

이러한 팁은 다양한 사용자 시나리오에서 최적의 EBS 볼륨 성능을 달성하는 방법에 대한 모범 사례를 보여줍니다.

EBS 최적화 인스턴스 사용

EBS 최적화 처리량을 지원하지 않는 인스턴스에서는 네트워크 트래픽이 사용자의 인스턴스와 EBS 볼륨 간 트래픽과 경합할 수 있습니다. EBS 최적화 인스턴스에서는 이 두 유형의 트래픽이 분리되어 있습니다. 일부 EBS 최적화 인스턴스 구성은 추가 요금을 요구하지만(예: C3, R3, M3), 일부는 추가 요금 없이 항상 EBS에 최적화됩니다(예: M4, C4, C5, D2). 자세한 내용은 Amazon EBS 최적화 인스턴스 (p. 984) 단원을 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. 1000) 단원을 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. 1007) 단원을 참조하십시오.

스냅샷에서 볼륨을 초기화하는 경우 성능 저하에 유의

스냅샷에서 복원된 새 EBS 볼륨의 각 데이터 블록에 처음 액세스할 때 지연 시간이 상당히 증가합니다. 다음 옵션 중 하나를 사용하여 이 성능 저하를 방지할 수 있습니다.

- 볼륨을 프로덕션에 투입하기 전에 각 블록에 액세스합니다. 이 프로세스를 초기화(이전에는 사전 워밍이라고 함)라고 합니다. 자세한 내용은 Amazon EBS 볼륨 초기화 (p. 1001) 단원을 참조하십시오.
- 스냅샷에서 빠른 스냅샷 복원을 활성화하여 스냅샷에서 생성된 EBS 볼륨이 생성 시 완전히 초기화되고 모든 프로비저닝된 성능을 즉시 제공하도록 보장합니다. 자세한 내용은 Amazon EBS 빠른 스냅샷 복원 (p. 978) 단원을 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 제한할 수 있는 다른 요소로는 인스턴스가 지원할 수 있는 수준 이상의 처리량을 구동하는 경우, 스냅샷에서 복원한 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 (p. 892) 단원을 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. 1000) 단원을 참조하십시오.

`st1` 및 `sc1`에서 처리량이 높은 읽기 중심 워크로드의 미리 읽기 향상

일부 워크로드는 읽기 중심이며 운영 체제 페이지 캐시를 통해(예: 파일 시스템에서) 블록 디바이스에 액세스합니다. 이 경우 최대 처리량을 획득하려면 미리 읽기 설정을 1MiB로 구성하는 것이 좋습니다. 이것은 HDD 볼륨에만 적용되는 블록 디바이스별 설정입니다.

블록 디바이스의 현재 미리읽기 값을 검사하려면 다음 명령을 사용합니다.

```
[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 범용 SSD이나 `gp2`보다는 `st1`(`sc1`) 볼륨 사용을 고려해야 합니다.

최신 Linux 커널 사용

간접 서술자를 지원하는 최신 Linux 커널을 사용합니다. 현재 세대 EC2 인스턴스뿐만 아니라 Linux 커널 3.8 이상 버전도 모두 이 지원을 제공합니다. 평균 I/O 크기가 44KiB 정도인 경우에는 간접 서술자가 지원되지 않는 인스턴스 또는 커널을 사용 중일 수 있습니다. Amazon CloudWatch 측정치에서 평균 I/O 크기를 도출하는 방법에 관한 내용은 I/O 특성 및 모니터링 (p. 1000) 단원을 참조하십시오.

`st1` 또는 `sc1` 볼륨의 처리량을 극대화하려면 `xen_blkfront.max` 파라미터(Linux 커널 버전 4.6 미만) 또는 `xen_blkfront.max_indirect_segments` 파라미터(Linux 커널 버전 4.6 이상)에 값 256을 적용하는 것이 좋습니다. 해당 파라미터는 OS 부팅 명령줄에서 설정할 수 있습니다.

예를 들어 이전 커널을 사용하는 Amazon Linux AMI에서 `/boot/grub/menu.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
```

이후 커널의 경우 명령은 다음과 같을 것입니다.

```
kernel /boot/vmlinuz-4.9.20-11.31.amzn1.x86_64 root=LABEL=/ console=tty1 console=ttyS0
 xen_blkfront.max_indirect_segments=256
```

이 설정을 적용하려면 인스턴스를 재부팅합니다.

자세한 내용은 반가상화 (PV-GRUB)에 대한 (p. 173) 단원을 참조하십시오. 다른 Linux 배포판, 특히 GRUB 부트 로더를 사용하지 않는 경우에는 다른 방식으로 커널 파라미터를 조정해야 할 수 있습니다.

EBS I/O 특성에 관한 자세한 내용은 이 주제를 다룬 Amazon EBS: Designing for Performance re:Invent 발표를 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. 1003) 단원을 참조하십시오.

Amazon CloudWatch을 통한 성능 추적

Amazon Web Services는 Amazon EBS를 통해 보고 분석할 수 있는 Amazon CloudWatch에 대한 성능 지표와 볼륨의 상태를 모니터링하는 데 사용할 수 있는 상태 검사를 제공합니다. 자세한 내용은 볼륨 상태 모니터링 (p. 918) 단원을 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/O 작업으로 병합을 시도합니다. 예를 들어, SSD 볼륨의 경우 1건의 1,024KiB I/O 작업은 4건(1,024÷256=4)의 작업으로 계산되지만 각각 32KiB인 8건의 연속하는 I/O 작업은 1건(8×32=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 디바이스에 대해 보류 중인 I/O 요청 수입니다. 지연 시간은 I/O 작업의 실제 종단 간 클라이언트 시간입니다. 다시 말해 EBS로 I/O를 전송한 후 EBS로부터 I/O 읽기 또는 쓰기가 완료되었다는 승인을 받기까지 소요된 시간입니다. 대기열 길이를 I/O 크기 및 지연 시간에 따라 정확히 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 250MiB/s의 볼륨 처리량 제한이 있습니다. 256KiB I/O 크기를 사용하는 경우, 볼륨은 1000 IOPS에서 처리량 제한에 도달합니다(1000 x 256KiB = 250MiB). I/O 크기가 작다면(예: 16KiB), 처리량이 250MiB/s에 훨씬 못 미치기 때문에 동일한 볼륨이 3,000 IOPS를 유지할 수 있습니다. (이 예제는 볼륨의 I/O가 인스턴스의 처리량 제한에 도달하지 않는다고 가정합니다.) 각 EBS 볼륨 유형의 처리량 제한에 대한 자세한 내용은 Amazon EBS 볼륨 유형 (p. 892) 단원을 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` 볼륨의 순차 I/O를 사용한다면 인스턴스 내에서 측정했을 때 예상보다 높은 IOPS를 관찰할 수 있습니다. 인스턴스 운영 체제가 순차 I/O를 병합하고 1,024KiB 크기 단위로 계산되는 경우에 이런 결과가 발생합니다. 워크로드가 소용량 또는 랜덤 I/O를 사용하는 경우 예상보다 적은 처리량을 관찰할 수 있습니다. 이는 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 EBS 최적화 인스턴스 (p. 984) 단원을 참조하십시오. EBS 볼륨에 충분한 I/O를 구동하고 있지 않은 경우에도 IOPS가 예상과 다를 수 있습니다.

CloudWatch로 I/O 특성 모니터링

각 볼륨의 CloudWatch 지표 (p. 918)로 이러한 I/O 특성을 모니터링할 수 있습니다. 고려해야 할 중요 측정치로는 다음 항목이 포함됩니다.

- `BurstBalance`
- `VolumeReadBytes`
- `VolumeWriteBytes`
- `VolumeReadOps`
- `VolumeWriteOps`
- `VolumeQueueLength`

`BurstBalance`는 `gp2`, `st1`, `sc1` 볼륨에 대한 버스트 버킷 잔고를 남은 잔고에 대한 비유로 표시합니다. 버스트 버킷이 모두 사용되면 볼륨 I/O(`gp2` 볼륨용) 또는 볼륨 처리량(`st1` 및 `sc1` 볼륨용)이 기준 수준으로 스로틀링됩니다. `BurstBalance` 값을 확인하여 이런 이유로 볼륨이 조절되는지 판단합니다.

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`를 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 발표를 참조하십시오.

Amazon EBS 볼륨 초기화

새 EBS 볼륨은 사용 가능하지만 초기화(이전에는 사전 워밍이라고 함)가 필요하지 않은 시점에 최고 성능을 발휘합니다.

스냅샷에서 복원된 볼륨의 경우, 스토리지 블록에 액세스하려면 먼저 스토리지 블록을 Amazon S3에서 풀다운하고 볼륨에 기록해야 합니다. 이 예비 작업에는 시간이 걸리며, 이로 인해 각 블록에 처음 액세스할 때 I/O

작업의 지연 시간이 상당히 증가할 수 있습니다. 모든 블록을 다운로드하고 볼륨에 기록한 후에 볼륨 성능이 구현됩니다.

> **Important**
>
> 스냅샷에서 복원한 `io1` 볼륨을 초기화할 경우 볼륨의 성능이 예상 수준보다 50퍼센트 이하로 떨어질 수 있으며, 이로 인해 볼륨에서 I/O 성능 상태 확인에 대해 `warning` 상태를 표시할 수 있습니다. 이는 원래 그런 것이므로 초기화 중에는 `warning` 볼륨에 대한 `io1` 상태를 무시해도 됩니다. 자세한 내용은 EBS 볼륨 상태 확인 (p. 918) 단원을 참조하십시오.

대부분의 애플리케이션은 볼륨 수명 주기 동안 초기화 비용을 분할 상환할 수 있습니다. 프로덕션 환경에서 이 초기 성능 저하를 방지하려면 다음 옵션 중 하나를 사용할 수 있습니다.

- 전체 볼륨을 강제로 즉시 초기화합니다. 자세한 내용은 Amazon EBS 볼륨 초기화 (p. 1001) 단원을 참조하십시오.
- 스냅샷에서 빠른 스냅샷 복원을 활성화하여 스냅샷에서 생성된 EBS 볼륨이 생성 시 완전히 초기화되고 모든 프로비저닝된 성능을 즉시 제공하도록 보장합니다. 자세한 내용은 Amazon EBS 빠른 스냅샷 복원 (p. 978) 단원을 참조하십시오.

Linux에서 Amazon EBS 볼륨 초기화

새 EBS 볼륨은 사용 가능하지만 초기화(이전에는 사전 워밍이라고 함)가 필요하지 않은 시점에 최고 성능을 발휘합니다. 스냅샷에서 복원된 볼륨의 경우 dd 또는 fio 유틸리티를 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 /
```

   여기서 새로운 볼륨인 `/dev/xvdf`가 연결되었지만 마운트되지는 않았음을 확인할 수 있습니다. `MOUNTPOINT` 열 아래에 나열된 경로가 없기 때문입니다.

3. dd 또는 fio 유틸리티를 사용하여 디바이스의 모든 블록을 읽습니다. dd 명령은 Linux 시스템에 기본으로 설치되지만, fio는 다중 스레드 읽기를 허용하기 때문에 훨씬 더 빠릅니다.

   > **Note**
   >
   > 이 단계는 EC2 인스턴스 대역폭, 볼륨에 대해 프로비저닝된 IOPS 및 볼륨 크기에 따라 몇 분에서 몇 시간까지 걸릴 수 있습니다.

   [dd] `if`(입력 파일) 파라미터는 초기화할 드라이브로 설정해야 합니다. `of`(파일 출력) 파라미터를 Linux null 가상 디바이스인 `/dev/null`로 설정해야 합니다. `bs` 파라미터는 읽기 작업의 블록 크기를 설정합니다. 최적의 성능을 얻으려면 이 값을 1MB로 설정해야 합니다.

   > **Important**
   >
   > dd를 잘못 사용하면 볼륨 데이터가 쉽게 삭제될 수 있습니다. 아래 예제 명령을 정확하게 따라야 합니다. 읽고 있는 디바이스의 이름에 따라 `if=/dev/`*`xvdf`* 파라미터만 다를 수 있습니다.

```
[ec2-user ~]$ sudo dd if=/dev/xvdf of=/dev/null bs=1M
```

   [fio] 시스템에 fio가 설치되어 있는 경우, 다음 명령을 사용하여 볼륨을 초기화할 수 있습니다. `--filename`(입력 파일) 파라미터는 초기화할 드라이브로 설정해야 합니다.

```
[ec2-user ~]$ sudo fio --filename=/dev/xvdf --rw=read --bs=128k --iodepth=32 --
ioengine=libaio --direct=1 --name=volume-initialize
```

fio을 Amazon Linux에 설치하려면 다음 명령을 사용하십시오.

```
sudo yum install -y fio
```

Ubuntu에 fio를 설치하려면 다음 명령을 사용합니다.

```
sudo apt-get install -y fio
```

작업이 끝나면 읽기 작업에 대한 보고서가 나타납니다. 이제 볼륨을 사용할 준비가 되었습니다. 자세한 내용은 Amazon EBS 볼륨을 Linux에서 사용할 수 있도록 만들기 (p. 914) 단원을 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 가용성 및 내구성 단원을 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 Windows의 RAID 구성 단원을 참조하십시오.

내용

- RAID 구성 옵션 (p. 1003)
- Linux에서 RAID 어레이 생성 (p. 1004)
- RAID 배열에 볼륨 스냅샷 생성 (p. 1007)

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 내결함성보다 더 중요한 경우(예: 데이터 복제가 이미 별도로 설정되어 있는 사용량이 많은 데이터베이스)	I/O가 스트라이프의 볼륨에 분산됩니다. 볼륨을 추가하면 처리량 및 IOPS도 그에 따라 바로 추가됩니다.	스트라이프의 성능이 세트에서 가장 성능이 떨어지는 볼륨으로 제한됩니다. 단일 볼륨이 손실되면 어레이의 데이터가 완전히 손실됩니다.

구성	--set-visible-to-all-users	장점	단점
RAID 1	내결함성이 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 500GiB Amazon EBS `io1` 볼륨 2개는 각각 가용 대역폭이 8,000 IOPS이고 처리량이 1,000MiB/s인 1000GiB RAID 0 어레이를 생성하거나, 가용 대역폭이 4,000 IOPS이고 처리량이 500MiB/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 Wiki를 참조하십시오.

Linux에서 RAID 어레이 생성

다음 절차에 따라 RAID 어레이를 생성합니다. Windows 인스턴스에 대한 지침은 Windows 인스턴스용 Amazon EC2 사용 설명서의 Windows에서 RAID 어레이 생성을 참조하십시오.

Linux에서 RAID 어레이를 생성하려면

1. 어레이에 대한 Amazon EBS 볼륨을 생성합니다. 자세한 내용은 Amazon EBS 볼륨 생성 (p. 908) 단원을 참조하십시오.

   > **Important**
   >
   > 어레이에 대해 크기 및 IOPS 성능 값이 동일한 볼륨을 생성합니다. EC2 인스턴스의 가용 대역폭을 초과하는 어레이를 생성하지 마십시오. 자세한 내용은 Amazon EBS 최적화 인스턴스 (p. 984) 단원을 참조하십시오.

2. 어레이를 호스팅할 인스턴스에 Amazon EBS 볼륨을 연결합니다. 자세한 내용은 Amazon EBS 볼륨을 인스턴스에 연결 (p. 910) 단원을 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
```

다음은 예제 출력입니다.

```
Personalities : [raid1]
md0 : active raid1 xvdg[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
```

다음은 예제 출력입니다.

```
/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, 레이블 *MY_RAID*로 ext4 파일 시스템을 생성하려면 다음 명령을 사용합니다.

```
[ec2-user ~]$ sudo mkfs.ext4 -L MY_RAID /dev/md0
```

애플리케이션의 요구 사항 또는 운영 체제의 제한에 따라 다른 파일 시스템 유형(예: ext3 또는 XFS)을 사용할 수 있습니다. 해당 파일 시스템 생성 명령은 파일 시스템 설명서를 참조하십시오.

6. 부팅할 때 RAID 배열이 자동으로 다시 수집되도록 하려면 RAID 정보가 포함된 구성 파일을 만듭니다.

```
[ec2-user ~]$ sudo mdadm --detail --scan | sudo tee -a /etc/mdadm.conf
```

Note

Amazon Linux 이외의 Linux 배포를 사용하는 경우 이 파일을 다른 위치에 배치해야 할 수도 있습니다. 자세한 내용은 Linux 시스템에서 man mdadm.conf를 참조하십시오.

7. 새 RAID 구성을 위해 블록 디바이스 모듈을 올바르게 미리 로드하려면 새 ramdisk 이미지를 만듭니다.

```
[ec2-user ~]$ sudo dracut -H -f /boot/initramfs-$(uname -r).img $(uname -r)
```

8. RAID 어레이에 대한 마운트 지점을 생성합니다.

```
[ec2-user ~]$ sudo mkdir -p /mnt/raid
```

9. 마지막으로 생성한 탑재 지점에 RAID 디바이스를 탑재합니다.

```
[ec2-user ~]$ sudo mount LABEL=MY_RAID /mnt/raid
```

이제 RAID 디바이스를 사용할 준비가 되었습니다.

10. (선택 사항) 시스템을 재부팅할 때마다 이 Amazon EBS 볼륨을 탑재하려면 디바이스에 대한 항목을 `/etc/fstab` 파일에 추가합니다.

    a. 수정 도중 실수로 이 파일이 손상되거나 삭제되는 경우에 대비하여 `/etc/fstab` 파일의 백업을 생성합니다.

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

    b. 자주 사용하는 텍스트 편집기를 사용하여 `/etc/fstab` 파일(예: nano 또는 vim)을 엽니다.

    c. "`UUID=`"로 시작하는 줄을 주석으로 처리하고, 파일 끝에 다음 형식으로 RAID 볼륨 파일의 새 줄을 추가합니다.

    ```
    device_label mount_point file_system_type fs_mntops fs_freq fs_passno
    ```

    이 줄의 마지막 세 필드는 파일 시스템 마운트 옵션, 파일 시스템의 덤프 빈도 및 부팅 시 파일 시스템 확인 순서입니다. 어떤 값을 입력해야 하는지 모르는 경우 아래 예제에 제공된 값을 사용하십시오(`defaults,nofail 0 2`)). `/etc/fstab` 항목에 대한 자세한 내용은 fstab 매뉴얼 페이지(명령줄에서 man fstab 입력)를 참조하십시오. 예를 들어, MY_RAID 레이블이 있는 디바이스에 `/mnt/raid` 탑재 지점에서 ext4 파일 시스템을 탑재하려면 `/etc/fstab`에 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/fstab`에 새 항목을 추가한 다음에는 해당 항목이 작동하는지 확인해야 합니다. 그런 다음 sudo mount -a 명령을 실행하여 `/etc/fstab`에 있는 모든 파일 시스템을 탑재합니다.

    ```
    [ec2-user ~]$ sudo mount -a
    ```

이전 명령에서 오류가 발생하지 않으면 `/etc/fstab` 파일이 정상이고 다음 부팅 시 파일 시스템이 자동으로 탑재됩니다. 명령에서 오류가 발생하면 오류를 검토한 다음 `/etc/fstab`를 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
```

RAID 배열에 볼륨 스냅샷 생성

스냅샷을 사용하여 RAID 배열의 EBS 볼륨에 데이터를 백업하려는 경우 스냅샷이 일관되어야 합니다. 이러한 볼륨의 스냅샷은 독립적으로 생성되기 때문입니다. 동기화되지 않은 스냅샷을 사용하여 RAID 배열의 EBS 볼륨을 복원할 경우 배열의 무결성이 손상됩니다.

RAID 배열에 일관된 스냅샷 집합을 생성하려면 EBS 다중 볼륨 스냅샷을 사용합니다. 다중 볼륨 스냅샷을 통해 EC2 인스턴스에 연결된 여러 EBS 볼륨에서 특정 시점, 데이터 조정 및 충돌 일치 스냅샷을 생성할 수 있습니다. 스냅샷은 여러 EBS 볼륨에서 자동으로 생성되기 때문에 일관성을 유지하기 위해 인스턴스를 중지하여 볼륨 간을 조정할 필요가 없습니다. 자세한 내용은 Amazon EBS 스냅샷 만들기 아래에서 다중 볼륨 스냅샷 만들기 단계를 참조하십시오.

EBS 볼륨 벤치마크

I/O 워크로드를 시뮬레이션하여 Amazon EBS 볼륨의 성능을 테스트할 수 있습니다. 프로세스는 다음과 같습니다.

1. EBS에 최적화된 인스턴스 시작.
2. 새 EBS 볼륨을 생성합니다.
3. EBS에 최적화된 인스턴스에 볼륨 추가.
4. 블록 디바이스를 구성하고 마운트합니다.
5. I/O 성능 벤치마크를 위한 도구 설치.
6. 볼륨의 I/O 성능 벤치마크.
7. 요금이 계속 발생하지 않도록 볼륨 삭제 및 인스턴스 종료.

> **Important**
>
> 일부 절차를 수행할 경우 자신이 벤치마크하는 EBS 볼륨에 있는 기존 데이터가 소멸되는 결과를 낳게 됩니다. 벤치마킹 절차는 프로덕션 볼륨이 아니라 테스트 목적으로 특별히 생성된 볼륨에 적용하기 위한 것입니다.

인스턴스 설정

EBS 볼륨에서 최적의 성능을 얻으려면 EBS에 최적화된 인스턴스를 사용하는 것이 좋습니다. EBS에 최적화된 인스턴스는 인스턴스와 함께 Amazon EC2와 Amazon EBS 사이의 전용 처리량을 제공합니다. EBS에 최적화된 인스턴스는 Amazon EC2 ~ Amazon EBS 간에 전용 대역폭을 전송하며, 인스턴스 유형에 따라 지정할 수 있습니다. 자세한 내용은 Amazon EBS 최적화 인스턴스 (p. 984) 단원을 참조하십시오.

EBS에 최적화된 인스턴스를 생성하려면 Amazon EC2 콘솔을 사용하여 인스턴스를 시작할 때 EBS 최적화 인스턴스로 시작(Launch as an EBS-Optimized instance)을 선택하거나 명령줄을 사용할 때 --ebs-optimized를 지정합니다. 이 옵션을 지원하는 현재 세대 인스턴스를 시작해야 합니다. 자세한 내용은 Amazon EBS 최적화 인스턴스 (p. 984) 단원을 참조하십시오.

프로비저닝된 IOPS SSD(io1) 볼륨 설정

io1 볼륨을 생성하려면 Amazon EC2 콘솔을 사용하여 볼륨 생성 시 프로비저닝된 IOPS SSD를 선택하거나 명령줄에서 --type io1 --iops n을 지정합니다. 여기서 n은 100~64,000 범위의 정수입니다. EBS 볼륨 사양에 대한 자세한 내용은 Amazon EBS 볼륨 유형 (p. 892) 단원을 참조하십시오. EBS 볼륨 생성에 대한 자세한 내용은 Amazon EBS 볼륨 생성 (p. 908) 단원을 참조하십시오. 인스턴스에 볼륨 연결에 대한 자세한 내용은 Amazon EBS 볼륨을 인스턴스에 연결 (p. 910) 단원을 참조하십시오.

예제 테스트의 경우 6개의 볼륨이 있는 RAID 어레이를 생성하는 것이 좋습니다. 이 어레이는 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_vol0 && sudo mkfs.xfs /dev/md0
[ec2-user ~]$ sudo mount -t xfs /dev/md0 /mnt/p_iops_vol0
[ec2-user ~]$ sudo chown ec2-user:ec2-user /mnt/p_iops_vol0/
```

처리량에 최적화된 HDD(st1) 또는 Cold HDD(sc1) 볼륨 설정

st1 볼륨을 생성하려면 Amazon EC2 콘솔을 사용하여 볼륨을 생성할 때 처리량에 최적화된 HDD을 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. 908) 단원을 참조하십시오. 인스턴스에 이러한 볼륨 연결에 대한 자세한 내용은 Amazon EBS 볼륨을 인스턴스에 연결 (p. 910) 단원을 참조하십시오.

AWS는 AWS CloudFormation에 사용할 JSON 템플릿을 제공하여 이 설정 절차를 간소화합니다. 템플릿에 액세스하고 이를 JSON 파일로 저장합니다. AWS CloudFormation에서는 사용자의 고유 SSH 키를 구성하고, 가장 간단한 방법으로 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/cloudformation에서 AWS CloudFormation 콘솔을 엽니다.
2. 스택 생성을 선택합니다.
3. Amazon S3에 템플릿 업로드를 선택하고 이전에 얻은 JSON 템플릿을 선택합니다.
4. 스택에 "ebs-perf-testing" 같은 이름을 붙이고 인스턴스 유형(기본은 r3.8xlarge)과 SSH 키를 선택합니다.
5. 다음을 두 번 선택한 다음, 스택 생성을 선택합니다.
6. 새로운 스택의 상태가 CREATE_IN_PROGRESS에서 COMPLETE로 바뀐 후에 출력을 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. 1009) 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-devel`에 대해 종속성이 있습니다.)
	fio을 Amazon Linux에 설치하려면 다음 명령을 실행하십시오.
	`[ec2-user ~]$ sudo yum install -y fio`
	Ubuntu에 fio를 설치하려면 다음 명령을 실행합니다.
	`sudo apt-get install -y fio`

Oracle Orion 보정 도구	Oracle 데이터베이스와 함께 사용할 스토리지 시스템의 I/O 성능을 보정합니다.

이러한 벤치마크 도구는 다양한 테스트 파라미터를 지원합니다. 볼륨이 지원하는 작업에 근접하는 명령을 사용해야 합니다. 아래 제공된 명령은 사용자가 시작하는 데 도움이 되는 예시용입니다.

볼륨 대기열 길이 선택

워크로드와 볼륨 유형에 따라 최적의 볼륨 대기열 길이를 선택합니다

SSD 지원 볼륨에서 대기열 길이

SSD 지원 볼륨의 워크로드에 대한 최적의 대기열 길이를 확인하려면 사용 가능한 모든 1000 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 3,000인 볼륨은 대기열 길이 3을 목표로 해야 합니다. 이 값을 높이거나 낮추면서 튜닝을 시도하여 애플리케이션에 가장 적합한 설정을 찾아야 합니다.

HDD 지원 볼륨에서 대기열 길이

HDD 지원 볼륨에서 워크로드에 가장 적합한 대기열 길이를 알아내려면 1MiB 순차 I/O를 수행하는 동시에 최소 4 이상의 대기열 길이를 목표로 하는 것이 좋습니다. 그러면 애플리케이션 성능을 모니터링하고 애플리케이션 요구 사항을 기준으로 해당 값을 조정할 수 있습니다. 예를 들어 버스트 처리량은 500MiB/s, IOPS는 500인 2TiB `st1` 볼륨의 경우 1,024KiB, 512KiB 또는 256KiB 순차 I/O를 수행하는 동시에 각각 4, 8 또는 16 대기열 길이를 목표로 해야 합니다. 이 값을 높이거나 낮추면서 튜닝을 시도하여 애플리케이션에 가장 적합한 설정을 찾아야 합니다.

C 상태 비활성화

벤치마킹 실행 전에 프로세서 C 상태를 비활성화해야 합니다. 일시적으로 지원 CPU 내 유휴 코어가 C 상태가 되어 전력을 절감할 수 있습니다. 코어가 호출되어 처리를 재개할 때 코어가 다시 완전히 작동하기까지 특정 시간이 흐릅니다. 이 지연 시간이 프로세서 벤치마킹 루틴을 방해할 수 있습니다. C 상태 및 이를 지원하는 EC2 인스턴스 유형에 대한 자세한 내용은 EC2 인스턴스에 대한 프로세서 상태 제어를 참조하십시오.

Linux 시스템에서 C 상태 비활성화

Amazon Linux, RHEL 및 CentOS에서 다음과 같이 C 상태를 비활성화할 수 있습니다.

1. C 상태 수를 가져옵니다.

```
$ cpupower idle-info | grep "Number of idle states:"
```

2. c1부터 cN까지 C 상태를 비활성화합니다. 이상적인 경우 코어는 c0 상태여야 합니다.

```
$ for i in `seq 1 $((N-1))`; do cpupower idle-set -d $i; done
```

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. 1001) 단원을 참조하십시오.

볼륨 테스트를 마치면 정리에 도움이 되는 Amazon EBS 볼륨 삭제 (p. 926) 및 인스턴스 종료 (p. 522) 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_vol0 --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_vol0 --name fio_test_file --direct=1 --
rw=randread --bs=16k --size=1G --numjobs=16 --time_based --runtime=180 --group_reporting --
norandommap
```

결과를 해석하는 방법에 대한 자세한 내용은 Inspecting disk IO performance with fio 자습서를 참조하십시오.

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. 998)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 IO 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 EBS의 Amazon CloudWatch 지표

CloudWatch 지표는 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(`io1`) 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 다릅니다. 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 지표에 대한 데이터 요소를 CloudWatch로 전송합니다. Amazon EBS 범용 SSD(gp2), 처리량에 최적화된 HDD(st1), 콜드 HDD(sc1) 및 Magnetic(표준) 볼륨은 5분 지표를 CloudWatch에 자동으로 전송합니다. 프로비저닝된 IOPS SSD(io1) 볼륨은 1분 지표를 CloudWatch에 자동으로 전송합니다. 볼륨이 인스턴스에 연결될 때만 CloudWatch에 데이터가 보고됩니다.

Nitro 기반 인스턴스에서는 이러한 지표 중 일부가 다릅니다. Nitro 시스템에 기초한 인스턴스 유형 목록은 Nitro 기반 인스턴스 (p. 181) 단원을 참조하십시오.

`AWS/EBS` 네임스페이스에는 다음 지표가 포함되어 있습니다.

Metrics

- 볼륨 지표 (p. 1012)
- 빠른 스냅샷 복원 지표 (p. 1015)

볼륨 지표

지표	설명
`VolumeReadBytes`	지정된 기간의 읽기 작업에 대한 정보를 제공합니다. `Sum` 통계는 해당 기간 동안 전송된 총 바이트 수를 보고합니다. `Average` 통계는 Nitro 기반 인스턴스에 연결된 볼륨을 제외하고 해당 기간 동안 각 읽기 작업의 평균 크기를 보고합니다. 여기에서 평균은 지정된 기간 동안의 평균을 나타냅니다. `SampleCount` 통계를 보면 해당 기간 동안 총 읽기 작업 수(Nitro 기반 인스턴스에 연결된 볼륨의 작업은 제외)를 알 수 있습니다. 여기에서 샘플 개수는 통계 계산 시 사용된 데이터 요소의 수를 나타냅니다. Xen 인스턴스의 경우 볼륨에서 읽기 작업이 있을 때에만 데이터가 보고됩니다. 이 지표의 `Minimum` 및 `Maximum` 통계는 Nitro 기반 인스턴스에 연결된 볼륨에서만 지원됩니다. 단위: 바이트
`VolumeWriteBytes`	지정된 기간의 쓰기 작업에 대한 정보를 제공합니다. `Sum` 통계는 해당 기간 동안 전송된 총 바이트 수를 보고합니다. `Average` 통계는 Nitro 기반 인스턴스에 연결된 볼륨을 제외하고 해당 기간 동안 각 쓰기 작업의 평균 크기를 보고합니다. 여기에서 평균은 지정된 기간 동안의 평균을 나타냅니다. `SampleCount` 통계를 보면 해당 기간 동안 총 쓰기 작업 수(Nitro 기반 인스턴스에 연결된 볼륨의 작업은 제외)를 알 수 있습니다. 여기에서 샘플 개수는 통계 계산 시 사용된 데이터 요소의 수를 나타냅니다. Xen 인스턴스의 경우 볼륨에서 쓰기 작업이 있을 때에만 데이터가 보고됩니다. 이 지표의 `Minimum` 및 `Maximum` 통계는 Nitro 기반 인스턴스에 연결된 볼륨에서만 지원됩니다. 단위: 바이트

지표	설명
`VolumeReadOps`	지정된 기간의 총 읽기 작업 수입니다. 해당 기간의 초당 평균 읽기 작업 수(읽기 IOPS)를 계산하려면 해당 기간의 총 읽기 작업 수를 해당 기간의 초 수로 나누십시오. 이 지표의 `Minimum` 및 `Maximum` 통계는 Nitro 기반 인스턴스에 연결된 볼륨에서만 지원됩니다. 단위: 개수
`VolumeWriteOps`	지정된 기간의 총 쓰기 작업 수입니다. 해당 기간의 초당 평균 쓰기 작업 수(쓰기 IOPS)를 계산하려면 해당 기간의 총 쓰기 작업 수를 해당 기간의 초 수로 나누십시오. 이 지표의 `Minimum` 및 `Maximum` 통계는 Nitro 기반 인스턴스에 연결된 볼륨에서만 지원됩니다. 단위: 개수
`VolumeTotalReadTime`	Note 이 지표는 다중 연결 지원 볼륨에서 지원되지 않습니다. 지정된 기간 동안 완료된 모든 읽기 작업에서 사용한 총 시간(초)입니다. 여러 요청이 동시에 제출된 경우 이 총계가 기간 길이보다 클 수 있습니다. 예를 들어, 5분(300초) 동안 700개의 작업이 완료되고 작업당 1초가 걸린 경우 값은 700초입니다. Xen 인스턴스의 경우 볼륨에서 읽기 작업이 있을 때에만 데이터가 보고됩니다. 이 지표에 대한 `Average` 통계는 Nitro 기반 인스턴스에 연결된 볼륨과는 관련이 없습니다. 이 지표의 `Minimum` 및 `Maximum` 통계는 Nitro 기반 인스턴스에 연결된 볼륨에서만 지원됩니다. 단위: 초
`VolumeTotalWriteTime`	Note 이 지표는 다중 연결 지원 볼륨에서 지원되지 않습니다. 지정된 기간 동안 완료된 모든 쓰기 작업에서 사용한 총 시간(초)입니다. 여러 요청이 동시에 제출된 경우 이 총계가 기간 길이보다 클 수 있습니다. 예를 들어, 5분(300초) 동안 700개의 작업이 완료되고 작업당 1초가 걸린 경우 값은 700초입니다. Xen 인스턴스의 경우 볼륨에서 쓰기 작업이 있을 때에만 데이터가 보고됩니다. 이 지표에 대한 `Average` 통계는 Nitro 기반 인스턴스에 연결된 볼륨과는 관련이 없습니다. 이 지표의 `Minimum` 및 `Maximum` 통계는 Nitro 기반 인스턴스에 연결된 볼륨에서만 지원됩니다. 단위: 초

지표	설명
`VolumeIdleTime`	Note 이 지표는 다중 연결 지원 볼륨에서 지원되지 않습니다. 지정된 기간 동안 읽기 또는 쓰기 작업이 제출되지 않은 총 시간(초)입니다. 이 지표에 대한 `Average` 통계는 Nitro 기반 인스턴스에 연결된 볼륨과는 관련이 없습니다. 이 지표의 `Minimum` 및 `Maximum` 통계는 Nitro 기반 인스턴스에 연결된 볼륨에서만 지원됩니다. 단위: 초
`VolumeQueueLength`	지정된 기간 동안 완료 대기 중인 읽기 및 쓰기 작업 요청 수입니다. 이 지표에 대한 `Sum` 통계는 Nitro 기반 인스턴스에 연결된 볼륨과는 관련이 없습니다. 이 지표의 `Minimum` 및 `Maximum` 통계는 Nitro 기반 인스턴스에 연결된 볼륨에서만 지원됩니다. 단위: 개수
`VolumeThroughputPercentage`	Note 이 지표는 다중 연결 지원 볼륨에서 지원되지 않습니다. 프로비저닝된 IOPS SSD 볼륨에만 사용됩니다. Amazon EBS 볼륨에 대해 프로비저닝된 총 IOPS 중 전송된 IOPS(초당 I/O 작업 수)의 비율(%)입니다. 프로비저닝된 IOPS SSD 볼륨은 99.9% 시간 동안 프로비저닝된 성능을 제공합니다. 쓰기 중 1분 동안 보류 중인 다른 I/O 요청이 없으면 지표 값이 100%가 됩니다. 또한 사용자가 취한 조치(예: 사용량 피크 시 볼륨 스냅샷 생성, EBS에 최적화되지 않은 인스턴스에서 볼륨 실행, 볼륨 데이터에 최초로 액세스 등)로 인해 볼륨의 I/O 성능이 일시적으로 저하될 수 있습니다. 단위: 백분율
`VolumeConsumedReadWriteOps`	프로비저닝된 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. 단위: 개수

지표	설명
`BurstBalance`	범용 SSD(`gp2`), 처리량에 최적화된 HDD(`st1`) 및 Cold HDD(`sc1`) 볼륨에만 사용됩니다. 버스트 버킷에 남아 있는 I/O 크레딧(`gp2`의 경우)의 비율 또는 처리량 크레딧의 비율(`st1` 및 `sc1`의 경우) 정보를 보여 줍니다. 볼륨이 활성 상태일 때만 CloudWatch에 데이터가 보고되고, 볼륨이 연결되지 않은 경우에는 데이터가 보고되지 않습니다. 이 지표에 대한 `Sum` 통계는 Nitro 기반 인스턴스에 연결된 볼륨과는 관련이 없습니다. 볼륨의 기준 성능이 최대 버스트 성능을 초과하면 크레딧은 사용되지 않습니다. 보고된 버스트 잔고는 0%(Nitro 기반 인스턴스) 또는 100%(비 Nitro 기반 인스턴스) 중 하나입니다. 자세한 내용은 I/O 크레딧 및 버스트 성능 (p. 894) 단원을 참조하십시오. 단위: 백분율

빠른 스냅샷 복원 지표

지표	설명
`FastSnapshotRestoreCreditsBucketSize`	누적될 수 있는 최대 볼륨 생성 크레딧 수. 이 지표는 가용 영역당 스냅샷별로 보고됩니다. 가장 유용한 통계는 `Average`입니다. `Minimum` 및 `Maximum` 통계의 결과는 `Average`의 통계와 동일하며 대신 사용될 수 있습니다.
`FastSnapshotRestoreCreditsBalance`	사용 가능한 볼륨 생성 크레딧 수. 이 지표는 가용 영역당 스냅샷별로 보고됩니다. 가장 유용한 통계는 `Average`입니다. `Minimum` 및 `Maximum` 통계의 결과는 `Average`의 통계와 동일하며 대신 사용될 수 있습니다.

Amazon EBS 지표의 차원

지원되는 차원은 볼륨 ID(`VolumeId`)입니다. 사용 가능한 모든 통계는 볼륨 ID로 필터링됩니다.

볼륨 지표 (p. 1012)의 경우 지원되는 차원은 볼륨 ID(`VolumeId`)입니다. 사용 가능한 모든 통계는 볼륨 ID로 필터링됩니다.

빠른 스냅샷 복원 지표 (p. 1015)의 경우 지원되는 차원은 스냅샷 ID(`SnapshotId`) 및 가용 영역(`AvailabilityZone`)입니다.

Amazon EC2 콘솔의 그래프

볼륨을 생성한 후 Amazon EC2 콘솔로 가서 볼륨의 모니터링 그래프를 볼 수 있습니다. 콘솔의 볼륨 페이지에서 볼륨을 선택하고 모니터링을 선택합니다. 다음 표에는 표시되는 그래프가 나열되어 있습니다. 오른쪽 열에는 CloudWatch API의 원시 데이터 측정치로 각 그래프가 생성되는 방법이 설명되어 있습니다. 모든 그래프의 기간은 5분입니다.

그래프	원시 지표를 사용하여 설명
Read Bandwidth (KiB/s)	`Sum(VolumeReadBytes) / Period / 1024`

그래프	원시 지표를 사용하여 설명
Write Bandwidth (KiB/s)	`Sum(VolumeWriteBytes) / Period / 1024`
읽기 처리 속도 (IOPS)	`Sum(VolumeReadOps) / Period`
쓰기 처리 속도 (IOPS)	`Sum(VolumeWriteOps) / Period`
평균 대기열 길이 (작업)	`Avg(VolumeQueueLength)`
% Time Spent Idle	`Sum(VolumeIdleTime) / Period × 100`

평균 읽기 크기 (KiB/작업)	`Avg(VolumeReadBytes) / 1024`
	
	Nitro 기반 인스턴스의 경우, 다음 공식에서 CloudWatch 지표 수식을 사용하여 평균 읽기 크기를 도출합니다.
	
	`(Sum(VolumeReadBytes) / Sum(VolumeReadOps)) / 1024`
	
	`VolumeReadBytes` 및 `VolumeReadOps` 지표는 EBS CloudWatch 콘솔에서 사용할 수 있습니다.

평균 쓰기 크기 (KiB/작업)	`Avg(VolumeWriteBytes) / 1024`
	
	Nitro 기반 인스턴스의 경우, 다음 공식에서 CloudWatch 지표 수식을 사용하여 평균 쓰기 크기를 도출합니다.
	
	`(Sum(VolumeWriteBytes) / Sum(VolumeWriteOps)) / 1024`
	
	`VolumeWriteBytes` 및 `VolumeWriteOps` 지표는 EBS CloudWatch 콘솔에서 사용할 수 있습니다.

평균 읽기 지연 시간 (ms/작업)	`Avg(VolumeTotalReadTime) × 1000`
	
	Nitro 기반 인스턴스의 경우, 다음 공식에서 CloudWatch 지표 수식을 사용하여 평균 읽기 지연 시간을 도출합니다.
	
	`(Sum(VolumeTotalReadTime) / Sum(VolumeReadOps)) × 1000`
	
	`VolumeTotalReadTime` 및 `VolumeReadOps` 지표는 EBS CloudWatch 콘솔에서 사용할 수 있습니다.

평균 쓰기 지연 시간 (ms/작업)	`Avg(VolumeTotalWriteTime) × 1000`
	
	Nitro 기반 인스턴스의 경우, 다음 공식에서 CloudWatch 지표 수식을 사용하여 평균 쓰기 지연 시간을 도출합니다.
	
	`(Sum(VolumeTotalWriteTime) / Sum(VolumeWriteOps)) * 1000`
	
	`VolumeTotalWriteTime` 및 `VolumeWriteOps` 지표는 EBS CloudWatch 콘솔에서 사용할 수 있습니다.

평균 지연 시간 그래프 및 평균 크기 그래프의 경우 기간 중 완료된 총 작업(그래프에 해당하는 읽기 또는 쓰기) 수를 기준으로 평균을 계산합니다.

Amazon EBS용 Amazon CloudWatch Events

Amazon EBS는 Amazon CloudWatch Events를 기반으로 다양한 볼륨, 스냅샷 및 암호화 상태 변경에 대한 알림을 보냅니다. CloudWatch 이벤트에서는 볼륨, 스냅샷 또는 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.

CloudWatch의 이벤트는 JSON 객체로 표현됩니다. 이 이벤트에 고유한 필드는 JSON 객체의 "세부 정보" 섹션에 포함되어 있습니다. "이벤트" 필드에는 이벤트 이름이 포함됩니다. "결과" 필드에는 이벤트를 트리거한 작업의 완료 상태가 포함됩니다. 자세한 내용은 Amazon CloudWatch Events 사용 설명서에서 CloudWatch 이벤트의 이벤트 패턴을 참조하십시오.

자세한 정보는 Amazon CloudWatch 사용 설명서의 이벤트 사용을 참조하십시오.

목차

- EBS 볼륨 이벤트 (p. 1017)
- EBS 스냅샷 이벤트 (p. 1020)
- EBS 볼륨 수정 이벤트 (p. 1023)
- EBS 빠른 스냅샷 복원 이벤트 (p. 1024)
- AWS Lambda를 이용한 CloudWatch 이벤트 처리 (p. 1025)

EBS 볼륨 이벤트

Amazon EBS는 다음과 같은 볼륨 이벤트가 발생하는 경우 CloudWatch 이벤트에 이벤트를 보냅니다.

이벤트

- 볼륨 생성(createVolume) (p. 1017)
- 볼륨 삭제(deleteVolume) (p. 1018)
- 볼륨 연결 또는 다시 연결(attachVolume, reattachVolume) (p. 1019)

볼륨 생성(`createVolume`)

볼륨 생성 작업이 완료되면 `createVolume` 이벤트가 AWS 계정으로 전송됩니다. 그러나 저장, 로깅, 아카이빙은 되지 않습니다. 이 이벤트에서 `available` 또는 `failed` 결과가 발생할 수 있습니다. 잘못된 KMS 키가 제공되면 아래 예와 같이 생성이 실패합니다.

이벤트 데이터

아래 목록에 성공적인 `createVolume` 이벤트에 대해 EBS가 발생시키는 JSON 객체의 예를 열거했습니다.

```
{
   "version": "0",
   "id": "01234567-0123-0123-0123-012345678901",
   "detail-type": "EBS Volume Notification",
   "source": "aws.ec2",
   "account": "012345678901",
   "time": "yyyy-mm-ddThh:mm:ssZ",
   "region": "us-east-1",
   "resources": [
      "arn:aws:ec2:us-east-1:012345678901:volume/vol-01234567"
   ],
   "detail": {
      "result": "available",
      "cause": "",
      "event": "createVolume",
```

```
      "request-id": "01234567-0123-0123-0123-0123456789ab"
    }
}
```

아래 목록에 `createVolume` 이벤트 실패 이후 EBS가 발생시키는 JSON 객체의 예를 열거했습니다. 실패의 원인은 비활성화된 KMS 키였습니다.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "EBS Volume Notification",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "sa-east-1",
  "resources": [
    "arn:aws:ec2:sa-east-1:0123456789ab:volume/vol-01234567",
  ],
  "detail": {
    "event": "createVolume",
    "result": "failed",
    "cause": "arn:aws:kms:sa-east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab
 is disabled.",
    "request-id": "01234567-0123-0123-0123-0123456789ab",
  }
}
```

아래에 `createVolume` 이벤트 실패 이후 EBS가 발생시키는 JSON 객체의 예를 제시했습니다. 실패의 원인은 보류 중인 KMS 키 가져오기였습니다.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "EBS Volume Notification",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "sa-east-1",
  "resources": [
    "arn:aws:ec2:sa-east-1:0123456789ab:volume/vol-01234567",
  ],
  "detail": {
    "event": "createVolume",
    "result": "failed",
    "cause": "arn:aws:kms:sa-east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab
 is pending import.",
    "request-id": "01234567-0123-0123-0123-0123456789ab",
  }
}
```

볼륨 삭제(`deleteVolume`)

볼륨 삭제 작업이 완료되면 `deleteVolume` 이벤트가 AWS 계정으로 전송됩니다. 그러나 저장, 로깅, 아카이빙은 되지 않습니다. 이 이벤트에 `deleted` 결과가 있습니다. 삭제가 완료되지 않으면 이벤트가 전송되지 않습니다.

이벤트 데이터

아래 목록에 성공적인 `deleteVolume` 이벤트에 대해 EBS가 발생시키는 JSON 객체의 예를 열거했습니다.

```
{
```

```
   "version": "0",
   "id": "01234567-0123-0123-0123-012345678901",
   "detail-type": "EBS Volume Notification",
   "source": "aws.ec2",
   "account": "012345678901",
   "time": "yyyy-mm-ddThh:mm:ssZ",
   "region": "us-east-1",
   "resources": [
      "arn:aws:ec2:us-east-1:012345678901:volume/vol-01234567"
   ],
   "detail": {
      "result": "deleted",
      "cause": "",
      "event": "deleteVolume",
      "request-id": "01234567-0123-0123-0123-0123456789ab"
   }
}
```

볼륨 연결 또는 다시 연결(attachVolume, reattachVolume)

볼륨이 인스턴스에 연결하지 못하거나 다시 연결하지 못하면 AWS 계정으로 `attachVolume` 또는 `reattachVolume` 이벤트가 전송됩니다. 그러나 저장, 로깅, 아카이빙은 되지 않습니다. KMS 키를 사용하여 EBS 볼륨을 암호화할 경우 키가 무효해지면, EBS는 해당 키가 나중에 인스턴스에 연결하거나 다시 연결하는 데 사용될 경우 이벤트를 발송합니다(아래 예 참조).

이벤트 데이터

아래 목록에 `attachVolume` 이벤트 실패 이후 EBS가 발생시키는 JSON 객체의 예를 열거했습니다. 실패의 원인은 보류 중인 KMS 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": ""
  }
}
```

EBS 스냅샷 이벤트

Amazon EBS는 다음과 같은 볼륨 이벤트가 발생하는 경우 CloudWatch 이벤트에 이벤트를 보냅니다.

이벤트

- 스냅샷 생성(createSnapshot) (p. 1020)
- 스냅샷 생성(createSnapshots) (p. 1021)
- 스냅샷 복사(copySnapshot) (p. 1022)
- 스냅샷 공유(shareSnapshot) (p. 1023)

스냅샷 생성(`createSnapshot`)

스냅샷 생성 작업이 완료되면 `createSnapshot` 이벤트가 AWS 계정으로 전송됩니다. 그러나 저장, 로깅, 아카이빙은 되지 않습니다. 이 이벤트에서 `succeeded` 또는 `failed` 결과가 발생할 수 있습니다.

이벤트 데이터

아래 목록에 성공적인 `createSnapshot` 이벤트에 대해 EBS가 발생시키는 JSON 객체의 예를 열거했습니다. `detail` 섹션에서 `source` 필드에는 소스 볼륨의 ARN이 들어 있습니다. `StartTime` 및 `EndTime` 필드는 스냅샷 생성이 시작된 시점과 완료된 시점을 나타냅니다.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
     "arn:aws:ec2:us-west-2::snapshot/snap-01234567"
  ],
  "detail": {
    "event": "createSnapshot",
    "result": "succeeded",
    "cause": "",
    "request-id": "",
    "snapshot_id": "arn:aws:ec2:us-west-2::snapshot/snap-01234567",
    "source": "arn:aws:ec2:us-west-2::volume/vol-01234567",
    "StartTime": "yyyy-mm-ddThh:mm:ssZ",
    "EndTime": "yyyy-mm-ddThh:mm:ssZ"   }
}
```

스냅샷 생성(createSnapshots)

다중 볼륨 스냅샷 생성 작업이 완료되면 `createSnapshots` 이벤트가 AWS 계정으로 전송됩니다. 이 이벤트에서 `succeeded` 또는 `failed` 결과가 발생할 수 있습니다.

이벤트 데이터

아래 목록에 성공적인 `createSnapshots` 이벤트에 대해 EBS가 발생시키는 JSON 객체의 예를 열거했습니다. `detail` 섹션의 `source` 필드에는 다중 볼륨 스냅샷 세트의 소스 볼륨 ARN이 포함되어 있습니다. `StartTime` 및 `EndTime` 필드는 스냅샷 생성이 시작된 시점과 완료된 시점을 나타냅니다.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Multi-Volume Snapshots Completion Status",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2::us-east-1:snapshot/snap-01234567",
    "arn:aws:ec2::us-east-1:snapshot/snap-012345678"
  ],
  "detail": {
    "event": "createSnapshots",
    "result": "succeeded",
    "cause": "",
    "request-id": "",
    "startTime": "yyyy-mm-ddThh:mm:ssZ",
    "endTime": "yyyy-mm-ddThh:mm:ssZ",
    "snapshots": [
      {
        "snapshot_id": "arn:aws:ec2::us-east-1:snapshot/snap-01234567",
        "source": "arn:aws:ec2::us-east-1:volume/vol-01234567",
        "status": "completed"
      },
      {
        "snapshot_id": "arn:aws:ec2::us-east-1:snapshot/snap-012345678",
        "source": "arn:aws:ec2::us-east-1:volume/vol-012345678",
        "status": "completed"
      }
    ]
  }
}
```

아래 목록에 `createSnapshots` 이벤트 실패 이후 EBS가 발생시키는 JSON 객체의 예를 열거했습니다. 실패의 원인은 하나 이상의 스냅샷을 완료하기 못했기 때문입니다. `snapshot_id`의 값은 실패한 스냅샷의 ARN입니다. `StartTime` 및 `EndTime`은 스냅샷 생성 작업이 시작 및 종료된 시기를 나타냅니다.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Multi-Volume Snapshots Completion Status",
 "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2::us-east-1:snapshot/snap-01234567",
    "arn:aws:ec2::us-east-1:snapshot/snap-012345678"
  ],
 "detail": {
```

```
    "event": "createSnapshots",
    "result": "failed",
    "cause": "Snapshot snap-01234567 is in status deleted",
    "request-id": "",
    "startTime": "yyyy-mm-ddThh:mm:ssZ",
    "endTime": "yyyy-mm-ddThh:mm:ssZ",
    "snapshots": [
      {
        "snapshot_id": "arn:aws:ec2::us-east-1:snapshot/snap-01234567",
        "source": "arn:aws:ec2::us-east-1:volume/vol-01234567",
        "status": "error"
      },
      {
        "snapshot_id": "arn:aws:ec2::us-east-1:snapshot/snap-012345678",
        "source": "arn:aws:ec2::us-east-1:volume/vol-012345678",
        "status": "deleted"
      }
    ]
  }
}
```

스냅샷 복사(copySnapshot)

스냅샷 복사 작업이 완료되면 copySnapshot 이벤트가 AWS 계정으로 전송됩니다. 그러나 저장, 로깅, 아카이빙은 되지 않습니다. 이 이벤트에서 succeeded 또는 failed 결과가 발생할 수 있습니다.

이벤트 데이터

아래 목록에는 copySnapshot 이벤트가 성공한 후 EBS가 전송하는 JSON 객체의 예가 나와 있습니다. snapshot_id의 값은 새로 생성된 스냅샷의 ARN입니다. detail 섹션에서 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": "succeeded",
    "cause": "",
    "request-id": "",
    "snapshot_id": "arn:aws:ec2:us-west-2::snapshot/snap-01234567",
    "source": "arn:aws:ec2:eu-west-1::snapshot/snap-76543210",
    "StartTime": "yyyy-mm-ddThh:mm:ssZ",
    "EndTime": "yyyy-mm-ddThh:mm:ssZ",
    "Incremental": "True"
  }
}
```

아래 목록에 copySnapshot 이벤트 실패 이후 EBS가 발생시키는 JSON 객체의 예를 열거했습니다. 실패의 원인은 잘못된 소스 스냅샷 ID였습니다. snapshot_id의 값은 실패한 스냅샷의 ARN입니다. detail 섹션에서 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. 그러나 저장, 로깅, 아카이빙은 되지 않습니다. 결과는 항상 `succeeded`입니다.

이벤트 데이터

아래에 `shareSnapshot` 이벤트 완료 이후 EBS가 발생시키는 JSON 객체의 예를 열거했습니다. `detail` 섹션에서 `source`의 값은 스냅샷을 공유한 사용자의 AWS 계정 번호입니다. `StartTime` 및 `EndTime`은 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": "01234567-01234-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-west-2::snapshot/snap-01234567"
  ],
  "detail": {
    "event": "shareSnapshot",
    "result": "succeeded",
    "cause": "",
    "request-id": "",
    "snapshot_id": "arn:aws:ec2:us-west-2::snapshot/snap-01234567",
    "source": 012345678901,
    "StartTime": "yyyy-mm-ddThh:mm:ssZ",
    "EndTime": "yyyy-mm-ddThh:mm:ssZ"
  }
}
```

EBS 볼륨 수정 이벤트

Amazon EBS는 볼륨이 수정될 때 CloudWatch 이벤트에 `modifyVolume` 이벤트를 보냅니다. 그러나 저장, 로깅, 아카이빙은 되지 않습니다.

```
{
   "version": "0",
   "id": "01234567-0123-0123-0123-012345678901",
   "detail-type": "EBS Volume Notification",
   "source": "aws.ec2",
   "account": "012345678901",
   "time": "yyyy-mm-ddThh:mm:ssZ",
   "region": "us-east-1",
   "resources": [
      "arn:aws:ec2:us-east-1:012345678901:volume/vol-03a55cf56513fa1b6"
   ],
   "detail": {
      "result": "optimizing",
      "cause": "",
      "event": "modifyVolume",
      "request-id": "01234567-0123-0123-0123-0123456789ab"
   }
}
```

EBS 빠른 스냅샷 복원 이벤트

Amazon EBS는 스냅샷에 대한 빠른 스냅샷 복원 상태가 변경되면 이벤트를 CloudWatch 이벤트로 보냅니다.

다음은 이 이벤트의 예제 데이터입니다.

```
{
   "version": "0",
   "id": "01234567-0123-0123-0123-012345678901",
   "detail-type": "EBS Fast Snapshot Restore State-change Notification",
   "source": "aws.ec2",
   "account": "123456789012",
   "time": "yyyy-mm-ddThh:mm:ssZ",
   "region": "us-east-1",
   "resources": [
      "arn:aws:ec2:us-east-1::snapshot/snap-03a55cf56513fa1b6"
   ],
   "detail": {
      "snapshot-id": "snap-1234567890abcdef0",
      "state": "optimizing",
      "zone": "us-east-1a",
      "message": "Client.UserInitiated - Lifecycle state transition",
   }
}
```

가능한 `state` 값은 `enabling`, `optimizing`, `enabled`, `disabling`, `disabled`입니다.

`message`에 가능한 값은 다음과 같습니다.

`Client.InvalidSnapshot.InvalidState - The requested snapshot transitioned to an invalid state (Error)`

빠른 스냅샷 복원을 요청하는 요청이 실패하고 상태가 `disabling` 또는 `disabled`로 전환되었습니다. 빠른 스냅샷 복원은 이 스냅 샷에 대해 활성화할 수 없습니다.

`Client.UserInitiated`

상태가 `enabling` 또는 `disabling`으로 전환되었습니다.

`Client.UserInitiated - Lifecycle state transition`

상태가 `optimizing`, `enabled` 또는 `disabled`로 전환되었습니다.

`Server.InsufficientCapacity - There was insufficient capacity available to satisfy the request`

불충분한 용량으로 인해 빠른 스냅샷 복원을 요청하는 요청이 실패하고 상태가 `disabling` 또는 `disabled`로 전환되었습니다. 잠시 기다렸다가 다시 시도하십시오.

`Server.InternalError - An internal error caused the operation to fail`

내부 오류로 인해 빠른 스냅샷 복원을 요청하는 요청이 실패하고 상태가 `disabling` 또는 `disabled`로 전환되었습니다. 잠시 기다렸다가 다시 시도하십시오.

AWS Lambda를 이용한 CloudWatch 이벤트 처리

Amazon EBS와 CloudWatch 이벤트를 사용하여 데이터 백업 워크플로를 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 일치하는 `createSnapshot` 이벤트를 Amazon EBS가 발생시킬 때 CloudWatch에 의해 호출됩니다. 이는 스냅샷이 완료되었음을 의미합니다. 호출되면 함수가 `us-east-2`에서 `us-east-1`로 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 안내서를 참조하십시오.

3. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
4. 이벤트, 규칙 생성, Select event source(이벤트 소스 선택), Amazon EBS 스냅샷을 선택합니다.
5. Specific Event(s)(특정 이벤트)에서 createSnapshot(스냅샷 생성)을, Specific Result(s)(특정 결과)에서 성공을 선택합니다.
6. Rule target(대상 실행)에서 이전에 생성한 샘플 함수를 찾아 선택합니다.
7. 대상, 대상 추가를 선택합니다.
8. Lambda 함수에서 이전에 생성한 Lambda 함수를 선택하고 세부 정보 구성을 선택합니다.
9. 규칙 세부 정보 구성 페이지에서 이름 및 설명에 값을 입력합니다. 상태 확인란을 선택하여 함수를 활성화합니다(활성으로 설정).
10. 규칙 생성을 선택합니다.

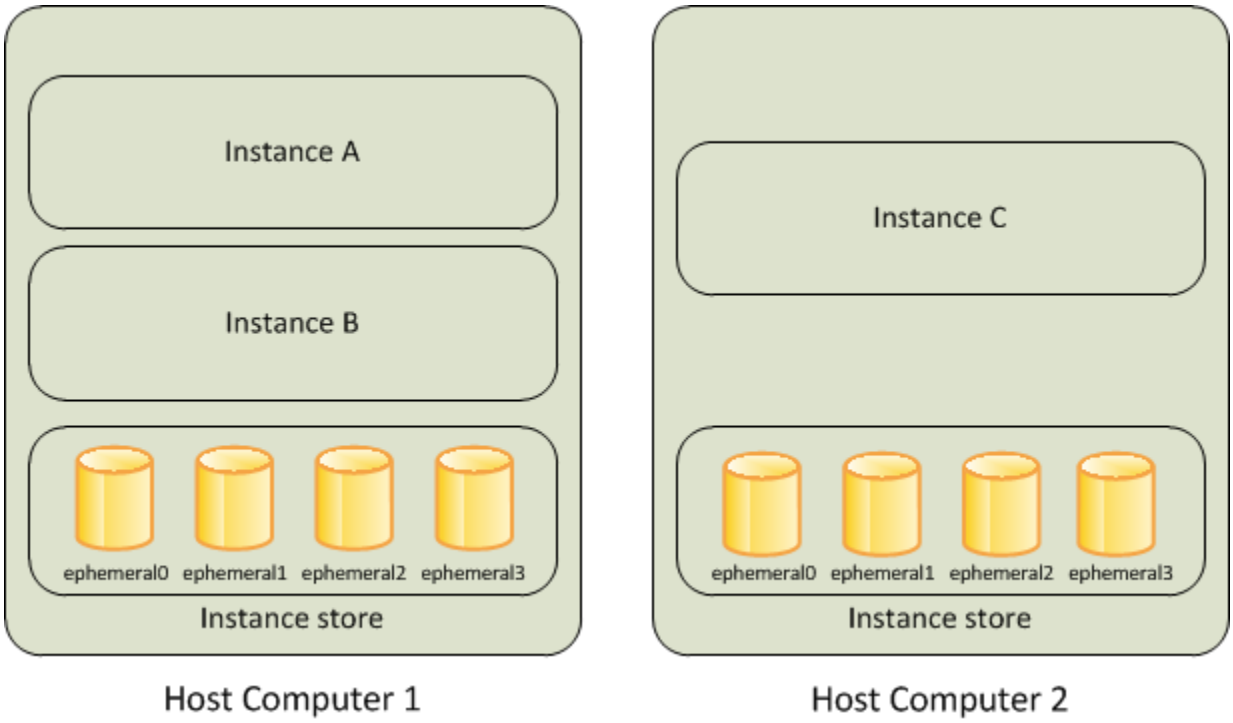
이제 규칙 탭에 규칙이 표시됩니다. 표시된 예에서 구성된 이벤트는 다음에 스냅샷을 복사할 때 EBS가 발생시킵니다.

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.

인스턴스 스토어 볼륨의 가상 디바이스는 `ephemeral[0-23]`입니다. 인스턴스 스토어 볼륨 1개를 지원하는 인스턴스 유형은 `ephemeral0`를 갖습니다. 인스턴스 스토어 볼륨 2개를 지원하는 인스턴스 유형은 `ephemeral0` 및 `ephemeral1` 등을 갖습니다.



내용

- 인스턴스 스토어 수명 (p. 1027)
- 인스턴스 스토어 볼륨 (p. 1028)
- EC2 인스턴스에 인스턴스 스토어 볼륨을 추가할 수 있습니다. (p. 1033)
- SSD 인스턴스 스토어 볼륨 (p. 1036)
- 인스턴스 스토리지 스왑 볼륨 (p. 1037)
- 인스턴스 스토어 볼륨의 디스크 성능 최적화 (p. 1039)

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
- 인스턴스가 중지됨
- 인스턴스가 종료됨

그러므로 중요한 장기 데이터의 경우 인스턴스 스토어에 의존하지 마십시오. 오히려 Amazon S3, Amazon EBS 또는 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 유형을 변경하면 인스턴스 스토어가 새 인스턴스 유형에 연결되지 않습니다. 자세한 정보는 인스턴스 유형 변경 (p. 262) 단원을 참조하십시오.

인스턴스 스토어 볼륨

인스턴스 유형은 사용 가능한 인스턴스 스토어의 크기와 인스턴스 스토어 볼륨에서 사용되는 하드웨어 유형을 결정합니다. 인스턴스 스토어 볼륨은 인스턴스의 사용 요금의 일부로 포함됩니다. 인스턴스를 시작할 때 사용할 인스턴스 스토어 볼륨을 지정해야 합니다(기본적으로 사용할 수 있는 NVMe 인스턴스 스토어 볼륨은 제외). 그런 다음 사용하기에 앞서 인스턴스 스토어 볼륨을 포맷하고 마운트해야 합니다. 인스턴스를 시작한 이후에는 사용 가능한 인스턴스 스토어 볼륨을 연결할 수 없습니다. 자세한 내용은 EC2 인스턴스에 인스턴스 스토어 볼륨을 추가할 수 있습니다. (p. 1033) 단원을 참조하십시오.

일부 인스턴스 유형은 NVMe 또는 SATA 기반 SSD(Solid State Drive)를 사용하여 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. 1036) 단원을 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 유형을 참조하십시오.

인스턴스 유형	인스턴스 스토리지 볼륨	유형	초기화 필요*	TRIM 지원**
`c1.medium`	1 x 350GB†	HDD	✓	
`c1.xlarge`	4 x 420GB(1.6TB)	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	✓	
`c5d.large`	1 x 50GB	NVMe SSD		✓
`c5d.xlarge`	1 x 100GB	NVMe SSD		✓
`c5d.2xlarge`	1 x 200GB	NVMe SSD		✓
`c5d.4xlarge`	1 x 400GB	NVMe SSD		✓
`c5d.9xlarge`	1 x 900GB	NVMe SSD		✓

인스턴스 유형	인스턴스 스토리지 볼륨	유형	초기화 필요*	TRIM 지원**
`c5d.12xlarge`	2 x 900GB(1.8TB)	NVMe SSD		✓
`c5d.18xlarge`	2 x 900GB(1.8TB)	NVMe SSD		✓
`c5d.24xlarge`	4 x 900GB(3.6TB)	NVMe SSD		✓
`c5d.metal`	4 x 900GB(3.6TB)	NVMe SSD		✓
`cc2.8xlarge`	4 x 840GB(3.36TB)	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		
`f1.2xlarge`	1 x 470GB	NVMe SSD		✓
`f1.4xlarge`	1 x 940GB	NVMe SSD		✓
`f1.16xlarge`	4 x 940GB(3.76TB)	NVMe SSD		✓
`g2.2xlarge`	1 x 60GB	SSD	✓	
`g2.8xlarge`	2 x 120GB(240GB)	SSD	✓	
`g4dn.xlarge`	1 x 125GB	NVMe SSD		✓
`g4dn.2xlarge`	1 x 225GB	NVMe SSD		✓
`g4dn.4xlarge`	1 x 225GB	NVMe SSD		✓
`g4dn.8xlarge`	1 x 900GB	NVMe SSD		✓
`g4dn.12xlarge`	1 x 900GB	NVMe SSD		✓
`g4dn.16xlarge`	1 x 900GB	NVMe SSD		✓
`h1.2xlarge`	1 x 2000GB(2TB)	HDD		
`h1.4xlarge`	2 x 2000GB(4TB)	HDD		
`h1.8xlarge`	4 x 2000GB(8TB)	HDD		
`h1.16xlarge`	8 x 2000GB(16TB)	HDD		
`hs1.8xlarge`	24 x 2,000GB(48TB)	HDD	✓	
`i2.xlarge`	1 x 800GB	SSD		✓
`i2.2xlarge`	2 x 800GB(1.6TB)	SSD		✓
`i2.4xlarge`	4 x 800GB(3.2TB)	SSD		✓
`i2.8xlarge`	8 x 800GB(6.4TB)	SSD		✓
`i3.large`	1 x 475GB	NVMe SSD		✓

인스턴스 유형	인스턴스 스토리지 볼륨	유형	초기화 필요*	TRIM 지원**
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		✓
i3.metal	8 x 1,900GB(15.2TB)	NVMe SSD		✓
i3en.large	1 x 1,250GB	NVMe SSD		✓
i3en.xlarge	1 x 2,500GB	NVMe SSD		✓
i3en.2xlarge	2 x 2,500GB(5TB)	NVMe SSD		✓
i3en.3xlarge	1 x 7,500GB	NVMe SSD		✓
i3en.6xlarge	2 x 7,500GB(15TB)	NVMe SSD		✓
i3en.12xlarge	4 x 7,500GB(30TB)	NVMe SSD		✓
i3en.24xlarge	8 x 7,500GB(60TB)	NVMe SSD		✓
i3en.metal	8 x 7,500GB(60TB)	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(1.6TB)	HDD	✓	
m2.xlarge	1 x 420GB	HDD	✓	
m2.2xlarge	1 x 850GB	HDD	✓	
m2.4xlarge	2 x 840GB(1.68TB)	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	✓	
m5ad.large	1 x 75GB	NVMe SSD		✓
m5ad.xlarge	1 x 150GB	NVMe SSD		✓
m5ad.2xlarge	1 x 300GB	NVMe SSD		✓
m5ad.4xlarge	2 x 300GB(600GB)	NVMe SSD		✓
m5ad.12xlarge	2 x 900GB(1.8TB)	NVMe SSD		✓
m5ad.24xlarge	4 x 900GB(3.6TB)	NVMe SSD		✓

인스턴스 유형	인스턴스 스토리지 볼륨	유형	초기화 필요*	TRIM 지원**
`m5d.large`	1 x 75GB	NVMe SSD		✓
`m5d.xlarge`	1 x 150GB	NVMe SSD		✓
`m5d.2xlarge`	1 x 300GB	NVMe SSD		✓
`m5d.4xlarge`	2 x 300GB(600GB)	NVMe SSD		✓
`m5d.8xlarge`	2 x 600GB(1.2TB)	NVMe SSD		✓
`m5d.12xlarge`	2 x 900GB(1.8TB)	NVMe SSD		✓
`m5d.16xlarge`	4 x 600GB(2.4TB)	NVMe SSD		✓
`m5d.24xlarge`	4 x 900GB(3.6TB)	NVMe SSD		✓
`m5d.metal`	4 x 900GB(3.6TB)	NVMe SSD		✓
`m5dn.large`	1 x 75GB	NVMe SSD		✓
`m5dn.xlarge`	1 x 150GB	NVMe SSD		✓
`m5dn.2xlarge`	1 x 300GB	NVMe SSD		✓
`m5dn.4xlarge`	2 x 300GB(600GB)	NVMe SSD		✓
`m5dn.8xlarge`	2 x 600GB(1.2TB)	NVMe SSD		✓
`m5dn.12xlarge`	2 x 900GB(1.8TB)	NVMe SSD		✓
`m5dn.16xlarge`	4 x 600GB(2.4TB)	NVMe SSD		✓
`m5dn.24xlarge`	4 x 900GB(3.6TB)	NVMe SSD		✓
`p3dn.24xlarge`	2 x 900GB(1.8TB)	NVMe 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		✓
`r5ad.large`	1 x 75GB	NVMe SSD		✓
`r5ad.xlarge`	1 x 150GB	NVMe SSD		✓
`r5ad.2xlarge`	1 x 300GB	NVMe SSD		✓
`r5ad.4xlarge`	2 x 300GB(600GB)	NVMe SSD		✓
`r5ad.12xlarge`	2 x 900GB(1.8TB)	NVMe SSD		✓
`r5ad.24xlarge`	4 x 900GB(3.6TB)	NVMe SSD		✓
`r5d.large`	1 x 75GB	NVMe SSD		✓
`r5d.xlarge`	1 x 150GB	NVMe SSD		✓

인스턴스 유형	인스턴스 스토리지 볼륨	유형	초기화 필요*	TRIM 지원**
`r5d.2xlarge`	1 x 300GB	NVMe SSD		✓
`r5d.4xlarge`	2 x 300GB(600GB)	NVMe SSD		✓
`r5d.8xlarge`	2 x 600GB(1.2TB)	NVMe SSD		✓
`r5d.12xlarge`	2 x 900GB(1.8TB)	NVMe SSD		✓
`r5d.16xlarge`	4 x 600GB(2.4TB)	NVMe SSD		✓
`r5d.24xlarge`	4 x 900GB(3.6TB)	NVMe SSD		✓
`r5d.metal`	4 x 900GB(3.6TB)	NVMe SSD		✓
`r5dn.large`	1 x 75GB	NVMe SSD		✓
`r5dn.xlarge`	1 x 150GB	NVMe SSD		✓
`r5dn.2xlarge`	1 x 300GB	NVMe SSD		✓
`r5dn.4xlarge`	2 x 300GB(600GB)	NVMe SSD		✓
`r5dn.8xlarge`	2 x 600GB(1.2TB)	NVMe SSD		✓
`r5dn.12xlarge`	2 x 900GB(1.8TB)	NVMe SSD		✓
`r5dn.16xlarge`	4 x 600GB(2.4TB)	NVMe SSD		✓
`r5dn.24xlarge`	4 x 900GB(3.6TB)	NVMe SSD		✓
`x1.16xlarge`	1 x 1,920GB	SSD		
`x1.32xlarge`	2 x 1,920GB(3.84TB)	SSD		
`x1e.xlarge`	1 x 120GB	SSD		
`x1e.2xlarge`	1 x 240GB	SSD		
`x1e.4xlarge`	1 x 480GB	SSD		
`x1e.8xlarge`	1 x 960GB	SSD		
`x1e.16xlarge`	1 x 1,920GB	SSD		
`x1e.32xlarge`	2 x 1,920GB(3.84TB)	SSD		
`z1d.large`	1 x 75GB	NVMe SSD		✓
`z1d.xlarge`	1 x 150GB	NVMe SSD		✓
`z1d.2xlarge`	1 x 300GB	NVMe SSD		✓
`z1d.3xlarge`	1 x 450GB	NVMe SSD		✓
`z1d.6xlarge`	1 x 900GB	NVMe SSD		✓
`z1d.12xlarge`	2 x 900GB(1.8TB)	NVMe SSD		✓
`z1d.metal`	2 x 900GB(1.8TB)	NVMe SSD		✓

* 특정 인스턴스에 연결된 볼륨은 초기화되지 않을 경우 최초 쓰기 페널티를 받게 됩니다. 자세한 내용은 인스턴스 스토어 볼륨의 디스크 성능 최적화 (p. 1039) 단원을 참조하십시오.

** 자세한 내용은 인스턴스 스토어 볼륨 TRIM 지원 (p. 1037) 단원을 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. 1037) 단원을 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 열거되고 디바이스 이름이 할당됩니다. 따라서 AMI 또는 인스턴스에 대한 블록 디바이스 매핑에 이를 포함하는 것은 효과가 없습니다. 자세한 내용은 블록 디바이스 매핑 (p. 1049) 단원을 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. 93) 단원을 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. 890) 단원을 참조하십시오.

인스턴스 실행 시에만 인스턴스에 대한 인스턴스 스토어 볼륨을 지정할 수 있습니다. 인스턴스를 실행한 이후에는 인스턴스 스토어 볼륨을 연결할 수 없습니다.

인스턴스 유형을 변경하면 인스턴스 스토어가 새 인스턴스 유형에 연결되지 않습니다. 자세한 내용은 인스턴스 유형 변경 (p. 262) 단원을 참조하십시오.

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. 블록 디바이스 매핑에 있는 인스턴스 스토어 볼륨 수가 인스턴스에 사용 가능한 인스턴스 스토어 볼륨 수를 초과한 경우 추가 볼륨이 무시됩니다. 인스턴스 유형별 인스턴스 스토어 볼륨 지원에 대한 자세한 내용은 인스턴스 스토어 볼륨 (p. 1028) 단원을 참조하십시오.

인스턴스에 대해 선택한 인스턴스 유형이 비 NVMe 인스턴스 스토어 볼륨을 지원하는 경우 인스턴스를 실행할 때 인스턴스에 대한 블록 디바이스 매핑에 추가해야 합니다. NVMe 인스턴스 스토어 볼륨은 기본적으로 사용 가능합니다. 인스턴스를 실행한 후에는 인스턴스에 대한 인스턴스 스토어 볼륨이 사용하기에 앞서 포맷되고 마운트되었는지 확인해야 합니다. 인스턴스 스토어 지원 인스턴스의 루트 볼륨은 기본적으로 마운트됩니다.

내용

- AMI에 인스턴스 스토어 볼륨 추가 (p. 1033)
- 인스턴스에 인스턴스 스토어 볼륨 추가 (p. 1034)
- 인스턴스 스토어 볼륨을 인스턴스에서 사용 가능하도록 만들기 (p. 1035)

AMI에 인스턴스 스토어 볼륨 추가

인스턴스 스토어 볼륨을 포함하는 블록 디바이스 매핑으로 AMI를 생성할 수 있습니다. 인스턴스 스토어 볼륨을 지원하는 인스턴스 유형과 해당 블록 디바이스 매핑에 인스턴스 스토어 볼륨을 지정하는 AMI로 인스턴

스를 시작하는 경우, 인스턴스에는 이러한 인스턴스 스토어 볼륨이 포함됩니다. 블록 디바이스 매핑에 있는 인스턴스 스토어 볼륨 수가 인스턴스에 사용 가능한 인스턴스 스토어 볼륨 수를 초과한 경우 추가 인스턴스 스토어 볼륨이 무시됩니다.

고려 사항

- 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.
- 인스턴스를 실행할 때 AMI 블록 디바이스 매핑에서 지정된 비 NVMe 인스턴스 스토어 볼륨을 생략하거나 인스턴스 스토어 볼륨을 추가할 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 인스턴스를 선택하고 인스턴스를 선택합니다.
3. 작업, 이미지, 이미지 생성을 차례로 선택합니다.
4. 이미지 생성 대화 상자에 이미지의 이름 및 설명을 입력합니다.
5. 추가할 각 인스턴스 스토어 볼륨에서 새 볼륨 추가를 선택한 다음 볼륨 유형에서 인스턴스 스토어 볼륨을 선택하고 디바이스에서 디바이스 이름을 선택합니다. (자세한 내용은 Linux 인스턴스의 디바이스 명명 (p. 1047) 단원을 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. 이미지 생성을 선택합니다.

명령줄을 사용하여 AMI에 인스턴스 스토어 볼륨을 추가하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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.

고려 사항

- M3 인스턴스의 경우, 인스턴스의 블록 디바이스 매핑에서 지정하지 않더라도 인스턴스 스토어 볼륨을 받을 수 있습니다.
- 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 인스턴스 시작을 선택합니다.

3. 1단계: Amazon 머신 이미지(AMI) 선택(Step 1: Choose an Amazon Machine Image (AMI))에서 사용할 AMI를 선택하고 선택을 선택합니다.
4. 마법사를 따라 1단계: Amazon 머신 이미지(AMI) 선택(Step 1: Choose an Amazon Machine Image (AMI)), 2단계: 인스턴스 유형 선택(Step 2: Choose an Instance Type) 및 3단계: 인스턴스 세부 정보 구성(Step 3: Configure Instance Details)을 완료합니다.
5. 4단계: 스토리지 추가(Step 4: Add Storage)에서 필요에 따라 기존 항목을 수정합니다. 추가할 각 인스턴스 스토어 볼륨에서 새 볼륨 추가를 선택한 다음 볼륨 유형에서 인스턴스 스토어 볼륨을 선택하고 디바이스에서 디바이스 이름을 선택합니다. 사용할 수 있는 인스턴스 스토어 볼륨의 개수는 인스턴스 유형에 따라 다릅니다.
6. 마법사를 완료하고 인스턴스를 시작합니다.
7. (선택 사항) 인스턴스에 사용 가능한 인스턴스 스토어 볼륨을 확인하려면 lsblk 명령을 실행합니다.

명령줄을 사용하여 인스턴스에 대한 블록 디바이스 매핑을 업데이트하려면

해당 명령과 함께 다음 옵션 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 참조하십시오.

- run-instances(AWS CLI)를 사용한 `--block-device-mappings`
- New-EC2Instance(Windows PowerShell용 AWS 도구)를 사용한 `-BlockDeviceMapping`

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 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. 1037) 단원을 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. 1057) 단원을 참조하십시오.

Windows 인스턴스의 경우 Windows 디스크 관리를 사용하여 인스턴스 스토어 볼륨을 볼 수도 있습니다. 자세한 정보는 Windows 디스크 관리를 이용하여 디스크 나열을 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`를 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

Linux의 SSD 인스턴스 스토어 볼륨에 최상의 IOPS 성능을 보장하려면 Amazon Linux의 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 인스턴스 시작 시 인스턴스에 대한 SSD 인스턴스 스토어 볼륨을 매핑해야 합니다. SSD 인스턴스 볼륨의 데이터는 연결된 인스턴스의 수명 기간 동안만 지속됩니다. 자세한 내용은 EC2 인스턴스에 인스턴스 스토어 볼륨을 추가할 수 있습니다. (p. 1033) 단원을 참조하십시오.

NVMe SSD 볼륨

C5d, G4, I3, I3en, F1, M5ad, M5d, `p3dn.24xlarge`, R5ad, R5d 및 z1d 인스턴스는 NVMe(Non-Volatile Memory Express) SSD(Solid State Drive) 인스턴스 스토어 볼륨을 제공합니다. NVMe 볼륨에 액세스하려면 NVMe 드라이버 (p. 981)가 설치되어 있어야 합니다. 다음 AMI가 이 요구 사항을 충족합니다.

- Amazon Linux 2
- Amazon Linux AMI 2018.03
- Ubuntu 14.04(`linux-aws` 커널) 이상
- Red Hat Enterprise Linux 7.4 이상
- SUSE Linux Enterprise Server 12 SP2 이상
- CentOS 7.4.1708 이상
- FreeBSD 11.1 이상
- Debian GNU/Linux 9 이상

인스턴스에 연결한 후 lspci 명령을 사용하여 NVMe 디바이스를 나열할 수 있습니다. 다음은 4개의 NVMe 디바이스를 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 다음 명령을 사용하여 NVMe 모듈이 로드되었는지 확인하십시오.

- Amazon Linux, Amazon Linux 2, Ubuntu 14/16, Red Hat Enterprise Linux, SUSE Linux Enterprise Server, CentOS 7

  ```
  $ lsmod | grep nvme
  nvme           48813  0
  ```

- Ubuntu 18

  ```
  $ cat /lib/modules/$(uname -r)/modules.builtin | grep nvme
  s/nvme/host/nvme-core.ko
  kernel/drivers/nvme/host/nvme.ko
  ```

```
kernel/drivers/nvmem/nvmem_core.ko
```

NVMe 볼륨은 NVMe 1.0e 사양을 준수합니다. NVMe 볼륨에 NVMe 명령을 사용할 수 있습니다. Amazon Linux에서는 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 하드웨어 모듈에 구현된 XTS-AES-256 블록 암호를 사용하여 NVMe 인스턴스 스토리지의 데이터를 암호화합니다. 하드웨어 모듈을 사용하여 암호화 키를 생성하며, 암호화 키는 각 NVMe 인스턴스 스토리지 디바이스에 고유합니다. 인스턴스가 중지되거나 종료되면 모든 암호화 키가 손상되어 복구가 불가능해집니다. 이 암호화를 비활성화할 수 없으며, 사용자 자신의 암호화 키를 제공할 수 없습니다.

비 NVMe SSD 볼륨

C3, G2, I2, M3, R3, X1 인스턴스는 비 NVMe SSD를 사용하여 높은 랜덤 I/O 성능을 제공하는 인스턴스 스토어 볼륨을 지원합니다. 인스턴스 유형별 인스턴스 스토어 볼륨 지원에 대한 자세한 내용은 인스턴스 스토어 볼륨 (p. 1028) 단원을 참조하십시오.

인스턴스 스토어 볼륨 TRIM 지원

다음 인스턴스는 TRIM을 통해 SSD 볼륨을 지원합니다. C5d, F1, G4, I2, I3, I3en, M5ad, M5d, `p3dn.24xlarge`, R3, R5ad, R5d 및 z1d.

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 액세스하는 속도를 높이려면 볼륨을 포맷할 때 TRIM 작업을 건너뛰어야 합니다.

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 향상시킬 수 있습니다. Linux에서는 `fstrim` 명령을 사용하여 주기적 TRIM을 활성화합니다.

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 페이징용으로 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 유형 변경 (p. 262) 단원을 참조하십시오.

`c1.medium` 및 `m1.small` 인스턴스 유형에는 작업 가능한 물리적 메모리의 양이 제한되어 있으며, 시작 시간에 Linux AMIs용 가상 메모리의 역할을 할 수 있는 900MiB의 스왑 볼륨이 부여됩니다. Linux 커널에서는 이 스왑 공간을 루트 디바이스의 파티션으로 간주하지만 이 공간은 루트 디바이스 유형에 상관없이 실제로는 별도의 인스턴스 스토어 볼륨입니다.

Amazon Linux는 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 `xvda3`를 사용할 수 있지만 해당 볼륨은 활성화되지 않은 상태입니다(`MOUNTPOINT` 필드가 공란임). swapon 명령을 사용하면 스왑 볼륨을 활성화할 수 있습니다.

> **Note**
>
> `/dev/`를 사용하여 디바이스 이름 앞에 lsblk를 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 ~]$ 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 스토어 볼륨 (p. 1028) 단원을 참조하십시오.

1. 인스턴스에 연결된 블록 디바이스 목록을 확인하여 인스턴스 스토어 볼륨에 사용할 디바이스 이름을 얻습니다.

   ```
   [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/xdvb`입니다. Amazon Linux 인스턴스이기 때문에 인스턴스 스토어 볼륨은 포맷된 후 `/media/ephemeral0`에 마운트됩니다. 모든 Linux 운영 체제에서 이러한 과정이 자동으로 수행되는 것은 아닙니다.

2. (선택 사항) 인스턴스 스토어 볼륨이 마운트되면(lsblk 명령 출력에 `MOUNTPOINT`로 목록 표시) 다음 명령으로 마운트를 해제합니다.

```
[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 swapspace 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 중단되면 인스턴스 스토어 볼륨 데이터가 손실됩니다. 여기에는 Step 3 (p. 1039)에서 생성한 인스턴스 스토어 스왑 공간 포맷도 포함됩니다. 따라서 인스턴스 스토어 스왑 공간을 사용하도록 구성한 인스턴스를 중단했다가 다시 시작할 경우에는 새로운 인스턴스 스토어 볼륨에서 Step 1 (p. 1038)부터 Step 5 (p. 1039)까지 반복해야 합니다.

인스턴스 스토어 볼륨의 디스크 성능 최적화

Amazon EC2가 디스크를 가상화하는 방식으로 인해, 일부 인스턴스 스토어 볼륨의 특정 위치에서 첫 번째 쓰기는 이후의 쓰기보다 느리게 수행됩니다. 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 스토어 볼륨 (p. 1028) 단원을 참조하십시오.

지연 시간 또는 처리량에 대한 높은 유연성이 필요한 경우 Amazon EBS 사용을 권장합니다.

인스턴스 스토어 볼륨을 초기화하려면 초기화할 스토어(예: `dd` 또는 `/dev/sdb`)에 따라 다음 `/dev/nvme1n1` 명령을 사용합니다.

> **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
```

파일 스토리지

클라우드 파일 스토리지는 공유 파일 시스템을 통해 서버와 애플리케이션에 데이터에 대한 액세스를 제공하도록 클라우드에 데이터를 저장하는 방법입니다. 이러한 호환성 덕분에 클라우드 파일 스토리지는 공유 파일 시스템을 사용하는 워크로드에 적합하며 코드 변경 없이 간단하게 통합할 수 있습니다.

블록 스토리지를 기본 사양으로 사용하며 확장성이 없거나 데이터 보호를 위한 중복성이 거의 없는 컴퓨팅 인스턴스에서의 단일 노드 파일 서버에서 직접 클러스터링하는 솔루션, Amazon Elastic File System (Amazon EFS) (p. 1040) or Amazon FSx for Windows File Server (p. 1044) 같은 완전 관리형 솔루셩에 이르는 다양한 파일 스토리지 솔루션이 존재합니다.

Amazon Elastic File System (Amazon EFS)

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 및 Amazon Elastic File System 상품 페이지를 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. 1041)

- 1단계: EFS 파일 시스템 만들기 (p. 1041)
- 2단계: 파일 시스템 마운트 (p. 1041)
- 3단계: 파일 시스템 테스트 (p. 1043)
- 4단계: 정리 (p. 1043)

사전 조건

- EC2 인스턴스 및 EFS 탑재 대상과 연결할 보안 그룹(예: efs-sg)을 생성하고 다음 규칙을 추가합니다.
  - 컴퓨터에서 인바운드 SSH로 EC2 인스턴스에 연결하도록 허용합니다(소스는 네트워크의 CIDR 블록).
  - 이 보안 그룹과 연결된 EC2 인스턴스에서 인바운드 NFS로 EFS 탑재 대상을 통해 파일 시스템에 연결하도록 허용합니다(소스는 보안 그룹 자체). 자세한 내용은 Amazon EFS 규칙 (p. 884) 및 Amazon Elastic File System 사용 설명서의 Amazon EC2 인스턴스 및 탑재 대상을 위한 보안 그룹 관련 문서를 참조하십시오.
- 키 페어를 생성합니다. 인스턴스를 구성할 경우 키 페어를 지정해야 하며, 키 페어를 지정하지 않으면 연결할 수 없습니다. 자세한 내용은 키 페어 생성 (p. 22) 단원을 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 만들기를 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 network access(네트워크 액세스 구성) 페이지에서 다음을 수행합니다.
   a. VPC에서 인스턴스에 사용할 VPC를 선택합니다.
   b. 탑재 대상 생성에서 가용 영역을 모두 선택합니다.
   c. 각 가용 영역에 대해 보안 그룹이 사전 조건 (p. 1041)에서 만든 보안 그룹인지 확인합니다.
   d. 다음 단계를 선택합니다.
4. Configure file system settings(파일 시스템 설정 구성) 페이지에서 다음을 수행합니다.
   a. 키=이름 태그의 경우 값에 파일 시스템 이름을 입력합니다.
   b. Choose throughput mode(처리량 모드 선택)에서 기본 옵션인 Bursting(버스팅)을 유지합니다.
   c. 성능 모드 선택에서 기본 옵션인 범용을 그대로 사용합니다.
   d. 다음 단계를 선택합니다.
5. Configure client access(클라이언트 액세스 구성) 페이지에서 기본 설정을 유지하고 다음 단계를 선택합니다.
6. 검토 및 생성 페이지에서 파일 시스템 생성을 선택합니다.
7. 파일 시스템이 생성되면 파일 시스템 ID를 나중에 참조할 수 있도록 기록해 둡니다.

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 재부팅된 후 다시 마운트되도록 합니다. T2 인스턴스는 반드시 서브넷에서 시작되어야 합니다. 기본 VPC 또는 기본이 아닌 VPC를 를 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 마운트를 참조하십시오.

두 개의 인스턴스를 실행하여 하나의 EFS 파일 시스템을 마운트하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 인스턴스 시작을 선택합니다.
3. Amazon 머신 이미지 선택 페이지에서 Amazon Linux AMI와 HVM 가상화 유형을 선택합니다.
4. 인스턴스 유형 선택 페이지에서 기본 인스턴스 유형인 `t2.micro`를 그대로 사용하고 Next: Configure Instance Details(다음: 인스턴스 세부 정보 구성)를 선택합니다.
5. 인스턴스 세부 정보 구성 페이지에서 다음을 수행합니다.

   a. 인스턴스의 수에 2를 입력합니다.

   b. [기본 VPC] 기본 VPC가 있는 경우 네트워크의 기본값이 됩니다. 기본 VPC와 서브넷의 기본값을 그대로 유지하여 Amazon EC2에서 인스턴스에 대해 선택한 가용 영역의 기본 서브넷을 사용합니다.

      [기본이 아닌 VPC] 네트워크에서 해당 VPC를 선택하고 서브넷에서 퍼블릭 서브넷을 선택합니다.

   c. [기본이 아닌 VPC] 퍼블릭 IP 자동 할당에서 활성화를 선택합니다. 그렇지 않으면 인스턴스가 퍼블릭 IP 주소 또는 퍼블릭 DNS 이름을 받지 못합니다.

   d. 고급 세부 정보에서 텍스트를 선택하고 사용자 데이터에 다음 스크립트를 붙여 넣습니다. FILE_SYSTEM_ID를 사용자 파일 시스템의 ID로 업데이트합니다. 원한다면 MOUNT_POINT를 탑재된 파일 시스템의 디렉터리로 업데이트할 수 있습니다.

      IMDSv2

```
#!/bin/bash
yum update -y
yum install -y nfs-utils
FILE_SYSTEM_ID=fs-xxxxxxxx
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600"`
AVAILABILITY_ZONE=$(curl -H "X-aws-ec2-metadata-token: $TOKEN" -s
 http://169.254.169.254/latest/meta-data/placement/availability-zone)
REGION=${AVAILABILITY_ZONE:0:-1}
MOUNT_POINT=/mnt/efs
mkdir -p ${MOUNT_POINT}
chown ec2-user:ec2-user ${MOUNT_POINT}
echo ${FILE_SYSTEM_ID}.efs.${REGION}.amazonaws.com:/ ${MOUNT_POINT} nfs4
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,_netdev 0 0
 >> /etc/fstab
mount -a -t nfs4
```

      IMDSv1

```
#!/bin/bash
yum update -y
yum install -y nfs-utils
FILE_SYSTEM_ID=fs-xxxxxxxx
AVAILABILITY_ZONE=$(curl -s http://169.254.169.254/latest/meta-data/placement/
availability-zone )
REGION=${AVAILABILITY_ZONE:0:-1}
MOUNT_POINT=/mnt/efs
mkdir -p ${MOUNT_POINT}
chown ec2-user:ec2-user ${MOUNT_POINT}
echo ${FILE_SYSTEM_ID}.efs.${REGION}.amazonaws.com:/ ${MOUNT_POINT} nfs4
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,_netdev 0 0
 >> /etc/fstab
```

```
mount -a -t nfs4
```

e. 마법사의 6단계로 진행합니다.

6. Configure Security Group(보안 그룹 구성) 페이지에서 Select an existing security group(기존 보안 그룹 선택)을 선택하고 사전 조건 (p. 1041)에서 생성한 보안 그룹을 선택한 다음, Review and Launch(검토 후 시작)를 선택합니다.
7. 인스턴스 시작 검토 페이지에서 시작을 선택합니다.
8. Select an existing key pair or create a new key pair(기존 키 페어 선택 또는 새 키 페어 생성) 대화 상자에서 Choose an existing key pair(기존 키 페어 선택)를 선택하고 키 페어를 선택합니다. 승인 확인란을 선택하고 인스턴스 시작을 선택합니다.
9. 탐색 창에서 인스턴스를 선택하여 인스턴스의 상태를 확인합니다. 처음에 인스턴스 상태는 `pending`입니다. 이 상태가 `running`으로 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. 485) 단원을 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.

```
file-system-id.efs.aws-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 인스턴스를 선택합니다.

3. 종료할 인스턴스를 선택합니다.
4. 작업, 인스턴스 상태, 종료를 차례로 선택합니다.
5. 확인 메시지가 나타나면 예, 종료를 선택합니다.

파일 시스템을 삭제하려면

1. https://console.aws.amazon.com/efs/에서 Amazon Elastic File System 콘솔을 엽니다.
2. 삭제한 파일 시스템을 선택합니다.
3. 작업, 파일 시스템 삭제를 차례로 선택합니다.
4. 확인 메시지가 표시되면 파일 시스템 ID를 입력하고 파일 시스템 삭제를 선택합니다.

Amazon FSx for Windows File Server

Amazon FSx for Windows File Server는 다양한 기능과 성능, 호환성을 바탕으로 엔터프라이즈 애플리케이션을 AWS로 쉽게 리프트 앤 시프트할 수 있는, 기본 Windows 파일 시스템을 바탕으로 하는 완전 관리형 Windows 파일 서버를 제공합니다.

Amazon FSx는 Microsoft Windows Server에 구축된 완전 관리형 파일 스토리지를 제공하는 다양한 엔터프라이즈 Windows 워크로드 모음을 지원합니다. Amazon FSx는 Windows 파일 시스템 기능과 네트워크를 통한 파일 스토리지에 액세스하는 업계 표준 SMB(Server Message Block) 프로토콜을 기본적으로 지원합니다. Amazon FSx는 AWS Cloud의 엔터프라이즈 애플리케이션에 최적화되어 있으며, Windows 호환성과 엔터프라이즈 성능 및 기능, 지속적인 1밀리초 미만 지연 시간을 지원합니다.

Windows 개발자와 관리자가 이용 중인 Amazon FSx 상의 파일 스토리지, 코드, 애플리케이션과 도구를 이용하면 어떤 변경도 없이 작업을 계속 진행할 수 있습니다. Amazon FSx에 이상적인 Windows 애플리케이션과 워크로드에는 비즈니스 애플리케이션, 홈 디렉터리, 웹 지원, 콘텐츠 관리, 데이터 분석, 소프트웨어 빌드 설정 및 미디어 처리 워크로드 등이 있습니다.

완전 관리형 서비스인 Amazon FSx for Windows File Server는 파일 서버 및 스토리지 볼륨 설정과 프로비저닝을 위한 관리 부담이 없습니다. 또한 Windows 소프트웨어를 최신 상태로 유지하고, 하드웨어 오류를 감지하고 처리하며, 백업을 수행하기도 합니다. Microsoft Active Directory용 AWS Directory Service, Amazon WorkSpaces, AWS Key Management Service, AWS CloudTrail 같은 다른 AWS 서비스와의 풍부한 통합도 제공합니다.

자세한 내용은 Amazon FSx for Windows File Server 사용 설명서 단원을 참조하십시오.

Amazon Simple Storage Service (Amazon S3)

Amazon S3는 인터넷 데이터의 리포지토리입니다. Amazon S3는 안정적이고 빠르며 비용이 저렴한 데이터 스토리지 인프라에 대한 액세스를 제공합니다. 언제든지 Amazon EC2 내에서 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 다수의 디바이스에 중복으로 저장하고, 많은 개별 클라이언트 또는 애플리케이션 스레드가 이런 데이터 객체에 대해 연속적 읽기 및 쓰기 액세스를 할 수 있도록 지원합니다. Amazon S3에 저장된 중복 데이터를 사용해서 인스턴스 또는 애플리케이션 장애로부터 빠르게 복구할 수 있습니다.

Amazon EC2는 Amazon 머신 이미지(AMI)를 저장하기 위해 Amazon S3를 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. 888) 단원을 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 URL을 사용해서 검색할 수 있습니다. 예를 들어, 키 값이 `/photos/mygarden.jpg`인 객체는 *`aws-s3-bucket1`* 버킷에 저장되며, 다음 URL을 사용하여 주소를 지정할 수 있습니다. `https://`*`aws-s3-bucket1`*`.s3.amazonaws.com/photos/mygarden.jpg`

Amazon S3에 대한 자세한 내용은 Amazon S3제품 페이지를 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 여러가지가 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 유틸리티는 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.

```
[ec2-user ~]$ wget https://my_bucket.s3.amazonaws.com/path-to-file
```

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 가이드를 참조하십시오.

AWS Command Line Interface

AWS Command Line Interface(AWS CLI)는 AWS 서비스를 관리하는 통합 도구입니다. AWS CLI를 통해 사용자는 인증을 하고 Amazon S3에서 제한되는 항목을 다운로드하고 다른 항목을 업로드할 수도 있게 됩니다. 이 도구의 설치 및 구성 등에 대한 자세한 내용은 AWS Command Line Interface 세부 정보 페이지 단원을 참조하십시오.

aws s3 cp 명령은 Unix cp 명령과 비슷합니다. 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.

```
[ec2-user ~]$ aws s3 cp s3:///my_bucket/my_folder/my_file.ext my_copied_file.ext
```

다음 명령을 사용해서 인스턴스에서 Amazon S3로 객체를 복사합니다.

```
[ec2-user ~]$ aws s3 cp my_copied_file.ext s3:///my_bucket/my_folder/my_file.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. 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.

```
[ec2-user ~]$ aws s3 sync s3://remote_S3_bucket local_directory
```

Amazon S3 API

개발자라면 API를 사용해서 Amazon S3의 데이터에 액세스할 수 있습니다. 자세한 내용은 Amazon Simple Storage Service 개발자 가이드를 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 체제와 인스턴스 유형에 따라 다릅니다. 인스턴스에 추가할 볼륨의 수를 고려할 때 I/O 대역폭 증가 또는 스토리지 용량 증가의 필요성 여부를 고려해야 합니다.

내용

- Linux 볼륨 제한 (p. 1046)
- Windows 볼륨 제한 (p. 1046)
- 인스턴스 유형 제한 (p. 1047)
- 대역폭 및 용량 비교 (p. 1047)

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 인스턴스를 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

드라이버	볼륨 제한
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 업그레이드를 참조하십시오.

디바이스 이름이 볼륨과 연결된 방식에 대한 자세한 내용은 Windows 인스턴스용 Amazon EC2 사용 설명서의 Windows EC2 인스턴스의 볼륨에 디스크 매핑을 참조하십시오.

인스턴스 유형 제한

A1, C5, C5d, C5n, G4, I3en, Inf1, M5, M5a, M5ad, M5d, M5dn, M5n, `p3dn.24xlarge`, R5, R5a, R5ad, R5d, R5dn, R5n, T3, T3a 및 z1d 인스턴스는 네트워크 인터페이스, EBS 볼륨 및 NVMe 인스턴스 스토어 볼륨을 포함해 최대 28개의 연결을 지원합니다. 모든 인스턴스에는 최소 한 개의 네트워크 인터페이스 연결이 있습니다. NVMe 인스턴스 스토어 볼륨은 자동으로 연결됩니다. 예를 들어, EBS 전용 인스턴스에 그 밖의 네트워크 인터페이스 연결이 없는 경우, 해당 인터페이스에 EBS 볼륨을 최대 27개까지 연결할 수 있습니다. 인스턴스에 NVMe 인스턴스 스토어 볼륨이 2개인 추가 네트워크 인터페이스가 1개 있는 경우 해당 인터페이스에 EBS 볼륨 24개를 연결할 수 있습니다. 자세한 내용은 탄력적 네트워크 인터페이스 (p. 683) 및 인스턴스 스토어 볼륨 (p. 1028) 단원을 참조하십시오.

`c5.metal`, `c5d.metal`, `i3.metal`, `m5.metal`, `m5d.metal`, `r5.metal`, `r5d.metal` 및 `z1d.metal` 인스턴스는 최대 31개의 EBS 볼륨을 지원합니다.

`u-6tb1.metal`, `u-9tb1.metal`, `u-12tb1.metal` 인스턴스는 최대 13개의 EBS 볼륨을 지원합니다. `u-18tb1.metal` 및 `u-24tb1.metal`인스턴스는 최대 19개의 EBS 볼륨을 지원합니다.

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 EBS 최적화 인스턴스 (p. 984)의 지침을 준수하여 볼륨에 프로비저닝된 IPOS와 인스턴스가 최대 성능에서 사용 가능한 대역폭을 일치시킵니다. RAID 구성의 경우 볼륨이 8개 이상인 어레이는 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.

인스턴스에 지원할 수 있는 볼륨의 수는 운영 체제에 따라 결정됩니다. 자세한 내용은 인스턴스 볼륨 제한 (p. 1046) 단원을 참조하십시오.

내용

- 사용 가능한 디바이스 이름 (p. 1048)
- 디바이스 이름 고려 사항 (p. 1048)

Windows 인스턴스의 디바이스 이름에 대한 자세한 내용은 Windows 인스턴스용 Amazon EC2 사용 설명서의 Windows 인스턴스에서의 디바이스 이름 지정 단원을 참조하십시오.

사용 가능한 디바이스 이름

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. 모든 인스턴스 유형이 HVM AMI를 지원합니다. 이전 세대의 일부 인스턴스 유형은 PV AMI를 지원합니다. 인스턴스의 가상화 유형에 따라 사용 가능하며 권장되는 디바이스 이름이 다르기 때문에 AMI의 가상화 유형에 주의해야 합니다. 자세한 내용은 Linux AMI 가상화 유형 (p. 95) 단원을 참조하십시오.

다음 표는 블록 디바이스 매핑에서 또는 EBS 볼륨에 연결 시 지정할 수 있는 사용 가능한 디바이스 이름을 나열합니다.

가상화 유형	응시 가능	루트 전용	EBS 볼륨 추천	인스턴스 스토리지 볼륨
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]
HVM	/dev/sd[a-z] /dev/xvd[b-c][a-z]	AMI에 따라 다름 /dev/sda1 또는 /dev/xvda	/dev/sd[f-p] *	/dev/sd[b-e] /dev/sd[b-h] (h1.16xlarge) /dev/sd[b-y] (d2.8xlarge) /dev/sd[b-i] (i2.8xlarge) **

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

** NVMe 인스턴스 스토어 볼륨은 자동으로 열거되고 NVMe 디바이스 이름이 할당됩니다.

인스턴스 스토어 볼륨에 대한 자세한 내용은 Amazon EC2 인스턴스 스토어 (p. 1027) 단원을 참조하십시오. NVMe EBS 볼륨에 대한 자세한 내용은 Linux 인스턴스의 Amazon EBS 및 NVMe (p. 981) 단원을 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.
- 인스턴스의 NVMe 인스턴스 스토어 볼륨의 수는 인스턴스의 크기에 따라 다릅니다. NVMe 인스턴스 스토어 볼륨은 자동으로 열거되고NVMe 디바이스 이름(`/dev/nvme[0-26]n1`)가 할당됩니다.

- 커널의 블록 디바이스 드라이버에 따라 디바이스는 사용자가 지정한 것과는 다른 이름에 연결될 수 있습니다. 예를 들어 `/dev/sdh`라는 디바이스 이름을 지정할 경우 디바이스 이름이 `/dev/xvdh` 또는 `/dev/hdh`로 바뀔 수 있습니다. 대부분의 경우 뒤에 오는 문자는 그대로 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). 이 경우 각 디바이스 이름에서 뒤에 오는 문자는 같은 수로 늘어납니다. 예를 들어 `/dev/sdb`를 `/dev/xvdf`로 이름 변경했다면, `/dev/sdc`는 `/dev/xvdg`로 변경됩니다. Amazon Linux은 이름이 변경된 디바이스에 대해 사용자가 지정한 이름으로 가는 심볼 링크를 만듭니다. 운영 체제가 다를 경우 다르게 작동할 수 있습니다.
- HVM AMI는 루트 디바이스용으로 예약된 `/dev/sda1` 그리고 `/dev/sda2`를 제외하고는 디바이스 이름에 후행 번호 사용을 지원하지 않습니다. `/dev/sda2`을 사용하는 것이 가능하지만 HVM 인스턴스와 함께 이 디바이스 매핑을 사용하지 않는 것이 좋습니다.
- PV AMI를 사용할 때 후행 숫자가 있거나 없는 동일한 디바이스 문자를 공유하는 볼륨은 연결할 수 없습니다. 예를 들어, 볼륨을 `/dev/sdc`로 연결한 다음 다른 볼륨을 `/dev/sdc1`에 연결하면 인스턴스에서는 `/dev/sdc`만을 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).
- 일부 사용자 지정 커널은 사용을 `/dev/sd[f-p]` 또는 `/dev/sd[f-p][1-6]`으로 제한하는 제약 조건이 있을 수 있습니다. `/dev/sd[q-z]` 또는 `/dev/sd[q-z][1-6]`을 사용하는 데 문제가 있을 경우 `/dev/sd[f-p]` 또는 `/dev/sd[f-p][1-6]`으로 전환해 보십시오.

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. 910)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 볼륨 지속 설정 (p. 19)단원을 참조하십시오.

내용

- 블록 디바이스 매핑의 개념 (p. 1049)
- AMI 블록 디바이스 매핑 (p. 1052)
- 인스턴스 블록 디바이스 매핑 (p. 1054)

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 EC2는 두 가지 유형의 블록 디바이스를 지원합니다.

- 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 열거되고 디바이스 이름이 할당됩니다. 따라서 블록 디바이스 매핑에 이를 포함하는 것은 효과가 없습니다.

내용

- 블록 디바이스 매핑 항목 (p. 1050)
- 블록 디바이스 매핑 인스턴스 스토어 경고 (p. 1050)
- 블록 디바이스 매핑 예제 (p. 1051)
- 운영 체제에서 디바이스 사용 방법 (p. 1051)

블록 디바이스 매핑 항목

블록 디바이스 매핑을 생성할 때 인스턴스에 연결할 각 블록 디바이스에 다음 정보를 지정합니다.

- Amazon EC2 내에서 사용되는 디바이스 이름 볼륨을 마운트할 때 인스턴스용 블록 디바이스 드라이버가 실제 볼륨 이름을 할당합니다. 할당된 이름이 Amazon EC2에서 권장하는 이름과 다를 수 있습니다. 자세한 내용은 Linux 인스턴스의 디바이스 명명 (p. 1047) 단원을 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 자동으로 열거되고 디바이스 이름이 할당되므로 블록 디바이스 매핑에 이를 포함하는 것은 효과가 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 내려받습니다. 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 `gp2`입니다.
- [EBS 볼륨] 볼륨이 지원하는 초당 입력/출력 작업 수(IOPS) (`gp2`, `st1`, `sc1` 또는 `standard` 볼륨에서는 사용되지 않음.)

블록 디바이스 매핑 인스턴스 스토어 경고

블록 디바이스 매핑에 인스턴스 스토어 볼륨이 있는 AMIs에서 인스턴스를 시작하는 경우 고려해야 할 몇 가지 경고 사항이 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 지원하는 데 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.

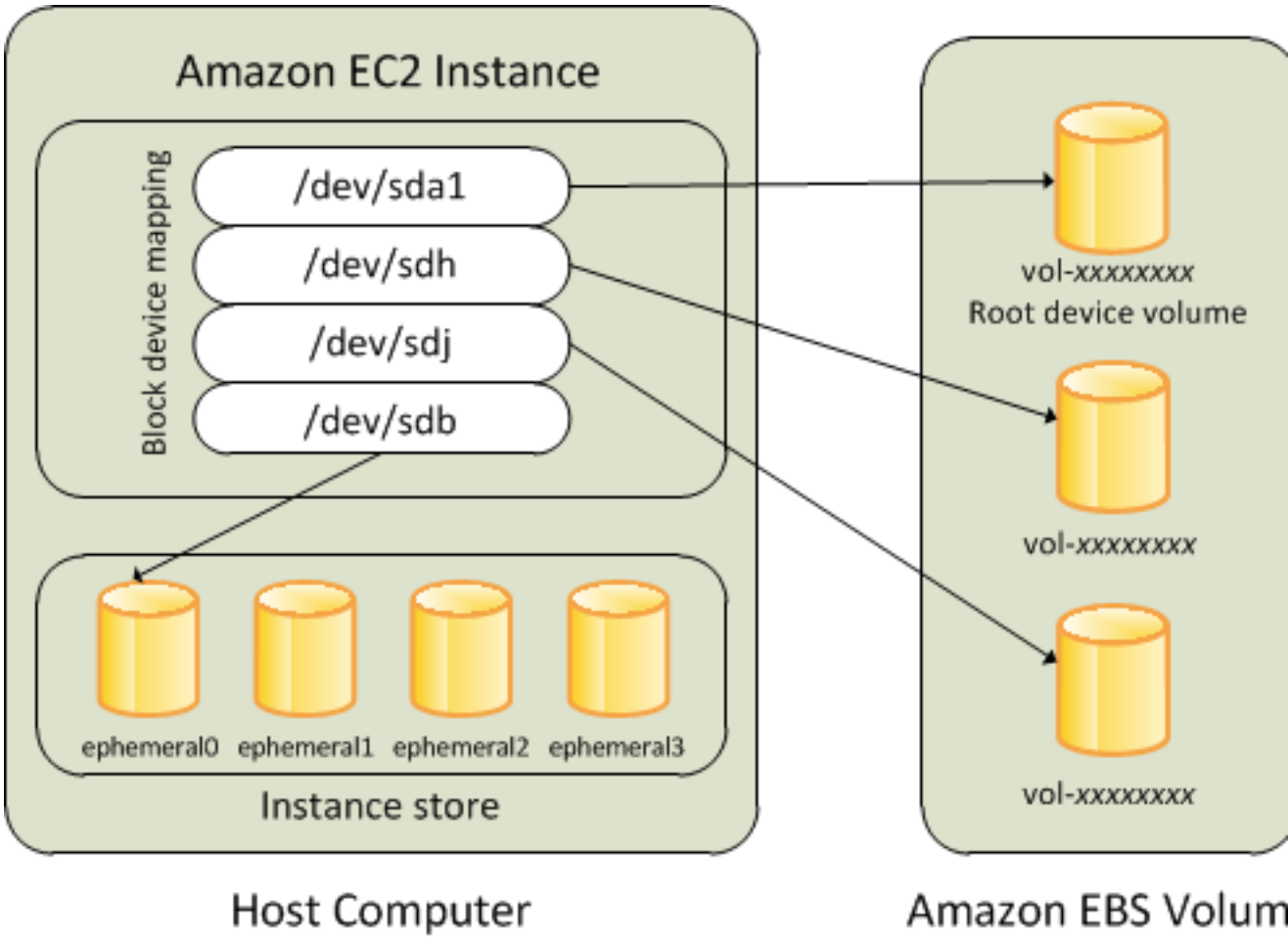
  Note

  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 볼륨

이 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. `/dev/sdb`를 `ephemeral0`으로 매핑하고 두 개의 EBS 볼륨을 각각 `/dev/sdh` 및 `/dev/sdj`로 매핑합니다. 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. 1052) 및 인스턴스 실행 시 블록 디바이스 매핑 업데이트 (p. 1054)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 마운트될지

가 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. 1027) 단원을 참조하십시오. 인스턴스용 블록 디바이스 드라이버에 따라 볼륨 포맷 및 마운트 시 어느 디바이스가 사용될지가 결정됩니다. 자세한 내용은 Amazon EBS 볼륨을 인스턴스에 연결 (p. 910) 단원을 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.

내용

- AMI용 블록 디바이스 매핑 지정 (p. 1052)
- AMI 블록 디바이스 매핑에서 EBS 볼륨 보기 (p. 1053)

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 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 작업, 이미지, 이미지 생성을 선택합니다.
4. 이미지 생성 대화 상자에서 새 볼륨 추가를 선택합니다.
5. 유형 목록에서 볼륨 유형을 선택하고 디바이스 목록에서 디바이스 이름을 선택합니다. EBS 볼륨의 경우 스냅샷, 볼륨 크기와 볼륨 유형을 선택적으로 지정할 수 있습니다.
6. 이미지 생성을 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.

--block-device-mappings 파라미터를 사용하여 블록 디바이스 매핑을 지정합니다. JSON으로 인코딩된 인수는 명령 줄에서 직접 제공하거나 파일 참조로 제공할 수 있습니다.

```
--block-device-mappings [mapping, ...]
```

```
--block-device-mappings [file://mapping.json]
```

인스턴스 스토어 볼륨을 추가하려면 다음 매핑을 사용합니다.

```
{
    "DeviceName": "/dev/sdf",
    "VirtualName": "ephemeral0"
}
```

비어 있는 100 GiB gp2 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 AMI를 선택합니다.
3. 필터 목록에서 EBS 이미지를 선택하여 EBS 지원 AMI 목록을 가져옵니다.
4. 원하는 AMI를 선택한 후 세부 정보 탭을 확인합니다. 루트 디바이스에서 최소한으로 사용 가능한 정보는 다음과 같습니다.
   - 루트 디바이스 유형 (`ebs`)
   - Root Device Name(예: `/dev/sda1`)

- 블록 디바이스(예: `/dev/sda1=snap-1234567890abcdef0:8:true`)

  AMI가 블록 디바이스 매핑을 사용하여 추가 EBS 볼륨으로 생성된 경우 블록 디바이스 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 덮어 쓰거나 병합됩니다.

제한

- 루트 볼륨에서는 다음 항목만 수정할 수 있습니다. 볼륨 크기, 볼륨 유형 및 종료 시 삭제 여부 플래그.
- EBS 볼륨을 수정할 때 크기는 줄일 수 없습니다. 그러므로 AMI의 블록 디바이스 매핑에서 지정된 스냅샷과 크기가 같거나 큰 스냅샷을 지정해야 합니다.

내용

- 인스턴스 실행 시 블록 디바이스 매핑 업데이트 (p. 1054)
- 실행 인스턴스의 블록 디바이스 매핑 업데이트 (p. 1055)
- 인스턴스 블록 디바이스 매핑에서 EBS 볼륨 보기 (p. 1056)
- 인스턴스 스토어 볼륨용 인스턴스 블록 디바이스 매핑 보기 (p. 1057)

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 인스턴스 시작을 선택합니다.
3. Amazon 머신 이미지(AMI) 선택 페이지에서 사용할 AMI를 선택하고 선택을 선택합니다.
4. 마법사 안내에 따라 인스턴스 유형 선택 및 인스턴스 세부 정보 구성 설정을 완료합니다.
5. 스토리지 추가 페이지에서 루트 볼륨, EBS 볼륨 및 인스턴스 스토어 볼륨을 다음과 같이 수정할 수 있습니다.

   - 루트 볼륨 크기를 변경하려면 유형 열 아래에 있는 루트 볼륨으로 이동한 후 크기 필드를 변경합니다.
   - 인스턴스를 실행하는 데 사용된 AMI의 블록 디바이스 매핑에서 지정된 EBS 볼륨을 표시하지 않으려면 해당 볼륨을 찾아 Delete 아이콘을 클릭합니다.
   - EBS 볼륨을 추가하려면 새 볼륨 추가를 선택하고 유형 목록에서 EBS를 선택한 후 필드(디바이스, 스냅샷 등)를 작성합니다.
   - 인스턴스가 실행된 AMI의 블록 디바이스 매핑에서 지정된 인스턴스 스토어 볼륨을 표시하지 않으려면 해당 볼륨으로 이동한 다음 삭제 아이콘을 선택합니다.

- 인스턴스 스토어 볼륨을 추가하려면, 새 볼륨 추가를 선택하고, 유형 목록에서 인스턴스 스토어를 선택한 후 디바이스에서 디바이스 이름을 선택합니다.

6. 나머지 마법사 페이지를 완료한 다음 시작을 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 식별하는 데 충분하므로 `/dev/sdh`에 스냅샷 ID를 지정할 필요가 없음에 유의하십시오.

```
{
    "DeviceName": "/dev/sdh",
    "Ebs": {
      "VolumeSize": 300
    }
}
```

추가 인스턴스 스토어 볼륨 `/dev/sdc`를 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

modify-instance-attribute AWS CLI 명령을 사용하여 실행 인스턴스의 블록 디바이스 매핑을 업데이트할 수 있습니다. 이 속성을 변경하기 전에 인스턴스를 중지할 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.json`에서 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 인스턴스를 선택합니다.
3. 검색 창에 루트 디바이스 유형을 입력한 후 EBS를 선택합니다. 이렇게 하면 EBS 기반 인스턴스 목록이 표시됩니다.
4. 원하는 인스턴스를 선택한 후 설명 탭에 표시되는 세부 정보를 확인합니다. 루트 디바이스에서 최소한으로 사용 가능한 정보는 다음과 같습니다.

   - 루트 디바이스 유형 (`ebs`)
   - Root device(예: `/dev/sda1`)
   - 블록 디바이스(예: `/dev/sda1`, `/dev/sdh` 및 `/dev/sdj`)

   인스턴스가 블록 디바이스 매핑을 사용하여 추가 EBS 볼륨으로 실행된 경우 블록 디바이스 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 블록 디바이스 옆의 항목을 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 블록 디바이스 매핑에 있는 비 NVMe 인스턴스 스토어 볼륨을 쿼리할 수 있습니다. NVMe 인스턴스 스토어 볼륨은 포함되지 않습니다.

전체 인스턴스 메타데이터를 요청하기 위한 기본 URI는 `http://169.254.169.254/latest/`입니다. 자세한 내용은 인스턴스 메타데이터 및 사용자 데이터 (p. 571) 단원을 참조하십시오.

우선, 실행 중인 인스턴스에 연결합니다. 인스턴스에서 이 쿼리를 사용하여 블록 디바이스 매핑을 가져옵니다.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-
ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-
data/block-device-mapping/
```

IMDSv1

```
[ec2-user ~]$ 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. `swap` 디바이스는 페이지 파일용입니다. 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.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-
ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-
data/block-device-mapping/ephemeral0
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/block-device-mapping/
ephemeral0
```

인스턴스 유형은 인스턴스에 사용 가능한 인스턴스 스토어 볼륨 수를 결정합니다. 블록 디바이스 매핑에 있는 인스턴스 스토어 볼륨 수가 인스턴스에 사용 가능한 인스턴스 스토어 볼륨 수를 초과한 경우 추가 볼륨이 무시됩니다. 인스턴스에 대한 인스턴스 스토어 볼륨을 확인하려면 lsblk 명령을 실행합니다. 각 인스턴스 유형에서 지원되는 인스턴스 스토어 볼륨 수를 알아보려면 인스턴스 스토어 볼륨 (p. 1028) 단원을 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.

내용

- 리소스 위치 (p. 1059)
- 리소스 ID (p. 1060)
- 리소스 목록화 및 필터링 (p. 1065)
- Amazon EC2리소스에 태그 지정 (p. 1068)
- Amazon EC2 서비스 제한 (p. 1079)
- Amazon EC2 사용 보고서 (p. 1081)

리소스 위치

일부 리소스는 모든 리전에서 사용할 수 있고(글로벌) 일부 리소스는 상주하는 리전 또는 가용 영역에만 해당합니다.

리소스	유형	설명
AWS 계정	전 세계	모든 리전에 동일한 AWS 계정을 사용할 수 있습니다.
키 페어	글로벌 또는 리전	Amazon EC2를 사용하여 생성하는 키 페어는 이를 생성한 리전과 연동됩니다. 자체 RSA 키 페어를 생성하여 사용할 리전에 업로드할 수 있습니다. 따라서 각 리전에 업로드하여 키를 전역적으로 사용 가능하게 만들 수 있습니다. 자세한 내용은 Amazon EC2 키 페어 (p. 861) 단원을 참조하십시오.
Amazon EC2 리소스 식별자	리전	AMI ID, 인스턴스 ID, EBS 볼륨 ID, EBS 스냅샷 ID 등 각 리소스 식별자는 해당 리전에 묶여 있으며, 리소스를 생성한 리전에서만 사용할 수 있습니다.
사용자가 공급한 리소스 이름	리전	보안 그룹 이름, 키 페어 이름 등 각 리소스 이름은 해당 리전에 묶여 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 묶여 있습니다. 한 리전의 AMI를 다른 리전으로 복사할

리소스	유형	설명
		수 있습니다. 자세한 내용은 AMI 복사 (p. 150) 단원을 참조하십시오.
탄력적 IP 주소	리전	탄력적 IP 주소는 리전에 묶여 있으며 동일한 리전의 인스턴스에만 연결할 수 있습니다.
보안 그룹	리전	보안 그룹은 리전에 묶여 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 묶여 있으며 동일한 리전에서 볼륨을 생성하는 데만 사용할 수 있습니다. 한 리전의 스냅샷을 다른 리전으로 복사할 수 있습니다. 자세한 내용은 Amazon EBS 스냅샷 복사 (p. 934) 단원을 참조하십시오.
EBS 볼륨	가용 영역	Amazon EBS 볼륨은 가용 영역에 묶여 있으며 동일한 가용 영역의 인스턴스에만 연결될 수 있습니다.
인스턴스	가용 영역	인스턴스는 이를 실행한 가용 영역에 묶여 있습니다. 그러나 인스턴스 ID는 그 리전에 묶여 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 Amazon EC2 및 Amazon EBS 리소스 유형에 더 긴 ID를 도입하고 있습니다. 영숫자 문자 조합 길이는 8자 형식이었는데, 새 ID는 17자 형식입니다(예: 인스턴스 ID `i-1234567890abcdef0`).

지원되는 리소스 유형에는 옵트인 기간이 있고, 그 기간 동안 리소스 ID 형식 및 기한을 선택할 수 있습니다. 기한 이후에는 리소스가 기본적으로 더 긴 ID 형식을 갖습니다. 특정 리소스 유형에 대한 기한이 지난 이후에는 해당 리소스 유형에 대해 더 긴 ID 형식을 비활성화할 수 없습니다.

리소스 유형마다 옵트인 기간과 기한이 다릅니다. 다음 표에는 지원되는 리소스 유형이 옵트인 기간 및 기한과 함께 나와 있습니다.

Resource type	옵트인 기간	기한
`instance` | `snapshot` |`reservation` | `volume`	더 이상 사용할 수 없음	2016년 12월 15일
`bundle` | `conversion-task` | `customer-gateway` | `dhcp-options` | `elastic-ip-allocation` | `elastic-ip-association` | `export-task` | `flow-log` | `image` | `import-task` |	2018년 2월 9일 ~2018년 6월 30일	2018년 6월 30일

Resource type	옵트인 기간	기한
`internet-gateway` | `network-acl` | `network-acl-association` | `network-interface` | `network-interface-attachment` | `prefix-list` | `route-table` | `route-table-association` | `security-group` | `subnet` | `subnet-cidr-block-association` | `vpc` | `vpc-cidr-block-association` | `vpc-endpoint` | `vpc-peering-connection` | `vpn-connection` | `vpn-gateway`		

옵트인 기간 사이

옵트인 기간 사이에는 리소스에 대한 더 긴 ID를 활성화 또는 비활성화할 수 있습니다. 한 리소스 유형에 더 긴 ID를 활성화한 후에는 생성하는 모든 새 리소스에 더 긴 ID가 생성됩니다.

> **Note**
>
> 리소스 ID는 생성 이후 변경되지 않습니다. 따라서 옵트인 기간 사이에 더 긴 ID를 활성화 또는 비활성화하는 것은 기존 리소스 ID에 영향을 미치지 않습니다.

AWS 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 FAQ의 더 긴 EC2 및 EBS 리소스 ID를 참조하십시오.

기한 이후

기한이 지난 이후에는 리소스 유형에 대해 더 긴 ID를 비활성화할 수 없습니다. 생성하는 모든 새로운 리소스는 더 긴 ID를 포함하여 생성됩니다.

더 긴 ID 작업

더 긴 ID의 설정 또는 설정 해제는 IAM 사용자 및 IAM 역할별로 가능합니다. 기본적으로 IAM 사용자 또는 역할은 루트 사용자와 동일한 설정으로 지정됩니다.

주제

- 더 긴 ID 설정 보기 (p. 1061)
- 더 긴 ID 설정 수정 (p. 1062)

더 긴 ID 설정 보기

콘솔 또는 명령줄 도구를 사용하여 긴 ID를 지원하는 리소스 유형을 볼 수 있습니다.

콘솔을 사용하여 더 긴 ID 설정을 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 화면 상단의 탐색 모음에서 더 긴 ID 설정을 보고자 하는 리전을 선택합니다.
3. 대시보드의 계정 속성에서 리소스 ID 길이 관리를 선택하십시오.

4. 고급 리소스 ID 관리를 확장하여 더 긴 ID를 지원하는 리소스 유형과 기한을 봅니다.

명령줄을 사용하여 더 긴 ID 설정을 보려면

다음 명령 중 하나를 사용합니다.

- describe-id-format(AWS CLI)

```
aws ec2 describe-id-format  --region region
```

- Get-EC2IdFormat (Windows PowerShell용 AWS 도구)

```
Get-EC2IdFormat -Region region
```

명령줄을 사용하여 특정 IAM 사용자 또는 IAM 역할에 대한 더 긴 ID 설정을 보려면

다음 명령 중 하나를 사용하고 요청 내에서 IAM 사용자, IAM 역할 또는 루트 계정 사용자의 ARN을 지정합니다.

- describe-identity-id-format(AWS CLI)

```
aws ec2 describe-identity-id-format --principal-arn arn-of-iam-principal --region region
```

- Get-EC2IdentityIdFormat(Windows PowerShell용 AWS 도구)

```
Get-EC2IdentityIdFormat -PrincipalArn arn-of-iam-principal -Region region
```

명령줄을 사용하여 특정 리전에 대해 집계된 더 긴 ID 설정을 보려면

describe-aggregate-id-format AWS CLI 명령을 사용하여 전체 리전에 대해 집계된 더 긴 ID 설정은 물론 각 리소스 유형에 대한 모든 ARN의 집계된 더 긴 ID 설정을 봅니다. 이 명령은 빠른 감사를 수행하여 특정 리전이 더 긴 ID에 대해 완전히 옵트인되었는지 확인하는 데 유용합니다.

```
aws ec2 describe-aggregate-id-format --region region
```

명시적으로 사용자 지정 더 긴 ID 설정을 정의한 사용자를 식별하려면

describe-principal-id-format AWS CLI 명령을 사용하여 명시적으로 더 긴 ID 설정을 지정한 루트 사용자와 모든 IAM 역할 및 IAM 사용자에 대한 더 긴 ID 형식 설정을 봅니다. 이 명령은 기본 더 긴 ID 설정을 재정의한 IAM 사용자 및 IAM 역할을 식별하는 데 유용합니다.

```
aws ec2 describe-principal-id-format --region region
```

더 긴 ID 설정 수정

콘솔 또는 명령줄 도구를 사용하여 여전히 옵트인 기간 내에 있는 리소스 유형의 더 긴 ID 설정을 수정할 수 있습니다.

> **Note**
>
> 이 단원의 AWS CLI 및 Windows PowerShell용 AWS 도구 명령은 각 리전별로만 사용됩니다. 별도로 지정되지 않는 한 기본 리전이 적용됩니다. 다른 리전의 설정을 수정하려면 명령에 `region` 파라미터를 포함하십시오.

콘솔을 사용하여 더 긴 ID 설정을 수정하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 화면 상단의 탐색 모음에서 더 긴 ID 설정을 수정하고자 하는 리전을 선택합니다.
3. 대시보드의 계정 속성에서 리스스 ID 길이 관리를 선택하십시오.
4. 다음 중 하나를 수행하십시오.
   - 모든 리전 전체의 모든 IAM 사용자에 대해 모든 지원 리소스의 더 긴 ID를 활성화하려면 긴 ID로 전환, 예, 긴 ID로 전환을 선택합니다.

     Important

     IAM 사용자 및 IAM 역할에서 이 작업을 수행하려면 `ec2:ModifyIdentityIdFormat` 권한이 필요합니다.
   - IAM 사용자 계정에 대한 특정 리소스 유형의 더 긴 ID 설정을 수정하려면 고급 리소스 ID 관리를 확장하고, 내 IAM 역할/사용자 열에 있는 해당 확인란을 선택하여 더 긴 ID를 활성화하거나, 확인란을 선택 해제하여 더 긴 ID를 비활성화합니다.
   - 모든 IAM 사용자에 대한 특정 리소스 유형의 더 긴 ID 설정을 수정하려면 고급 리소스 ID 관리를 확장하고, 모든 IAM 역할/사용자 열에 있는 해당 확인란을 선택하여 더 긴 ID를 활성화하거나, 확인란을 선택 해제하여 더 긴 ID를 비활성화합니다.

명령줄을 사용하여 IAM 사용자 계정에 대한 더 긴 ID 설정을 수정하려면

다음 명령 중 하나를 사용합니다.

Note

루트 사용자로서 이런 명령을 사용 중인 경우에는 IAM 사용자 또는 역할이 스스로 이러한 설정을 명시적으로 재정의하지 않는 한 이러한 변경 사항이 전체 AWS 계정에 적용됩니다.

- modify-id-format(AWS CLI)

```
aws ec2 modify-id-format --resource resource_type --use-long-ids
```

명령을 사용하여 모든 지원되는 리소스에 대한 더 긴 ID 설정을 수정할 수 있습니다. 이를 수행하려면 `resource_type` 파라미터를 `all-current`로 대체합니다.

```
aws ec2 modify-id-format --resource all-current --use-long-ids
```

Note

더 긴 ID를 비활성화하려면 `use-long-ids` 파라미터를 `no-use-long-ids`로 대체합니다.

- Edit-EC2IdFormat (Windows PowerShell용 AWS 도구)

```
Edit-EC2IdFormat -Resource resource_type -UseLongId boolean
```

명령을 사용하여 모든 지원되는 리소스에 대한 더 긴 ID 설정을 수정할 수 있습니다. 이를 수행하려면 `resource_type` 파라미터를 `all-current`로 대체합니다.

```
Edit-EC2IdFormat -Resource all-current -UseLongId boolean
```

명령줄을 사용하여 특정 IAM 사용자 또는 IAM 역할에 대한 더 긴 ID 설정을 수정하려면

다음 명령 중 하나를 사용하고 요청 내에서 IAM 사용자, IAM 역할 또는 루트 사용자의 ARN을 지정합니다.

- modify-identity-id-format(AWS CLI)

```
aws ec2 modify-identity-id-format --principal-arn arn-of-iam-principal --
resource resource_type --use-long-ids
```

명령을 사용하여 모든 지원되는 리소스에 대한 더 긴 ID 설정을 수정할 수 있습니다. 이를 수행하려면 `all-current` 파라미터에 대해 `--resource`를 지정합니다.

```
aws ec2 modify-identity-id-format --principal-arn arn-of-iam-principal --resource all-
current --use-long-ids
```

> **Note**
>
> 더 긴 ID를 비활성화하려면 `use-long-ids` 파라미터를 `no-use-long-ids`로 대체합니다.

- Edit-EC2IdentityIdFormat(Windows PowerShell용 AWS 도구)

```
Edit-EC2IdentityIdFormat -PrincipalArn arn-of-iam-principal -Resource resource_type -
UseLongId boolean
```

명령을 사용하여 모든 지원되는 리소스에 대한 더 긴 ID 설정을 수정할 수 있습니다. 이를 수행하려면 `all-current` 파라미터에 대해 `-Resource`를 지정합니다.

```
Edit-EC2IdentityIdFormat -PrincipalArn arn-of-iam-principal -Resource all-current -
UseLongId boolean
```

긴 ID 설정에 대한 액세스 제어

IAM 사용자와 역할은 관련된 IAM 정책을 통해 명시적으로 권한을 부여 받지 않는 한, 기본적으로 다음 작업을 사용할 수 있는 권한이 없습니다.

- `ec2:DescribeIdFormat`
- `ec2:DescribeIdentityIdFormat`
- `ec2:DescribeAggregateIdFormat`
- `ec2:DescribePrincipalIdFormat`
- `ec2:ModifyIdFormat`
- `ec2:ModifyIdentityIdFormat`

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",
        "ec2:DescribeAggregateIdFormat",
        "ec2:DescribePrincipalIdFormat"
      ],
      "Resource": "*"
    }
  ]
}
```

다음 작업에 대한 리소스 수준 권한은 지원되지 않습니다.

- `ec2:DescribeIdFormat`
- `ec2:DescribeIdentityIdFormat`
- `ec2:DescribeAggregateIdFormat`
- `ec2:DescribePrincipalIdFormat`
- `ec2:ModifyIdFormat`
- `ec2:ModifyIdentityIdFormat`

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.

내용

- 고급 검색 (p. 1065)
- 콘솔을 이용하여 리소스 목록화 (p. 1066)
- 콘솔을 이용하여 리소스를 필터링 (p. 1067)
- CLI 및 API를 이용하여 목록화 및 필터링 (p. 1068)

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 인스턴스 상태를 선택합니다.
  3. 추천 값 목록에서 중지됨을 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 접두사로 사용합니다.

- 부분 검색

  필드별로 검색하는 경우 부분 문자열을 입력하여 해당 필드에 있는 문자열에 포함된 모든 리소스를 검색할 수 있습니다. 예를 들어, 인스턴스 유형으로 검색한 후 **`t2`**를 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 태그별로 검색한 다음 **`^s.*`**를 입력하면 's'로 시작하는 이름 태그를 갖는 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

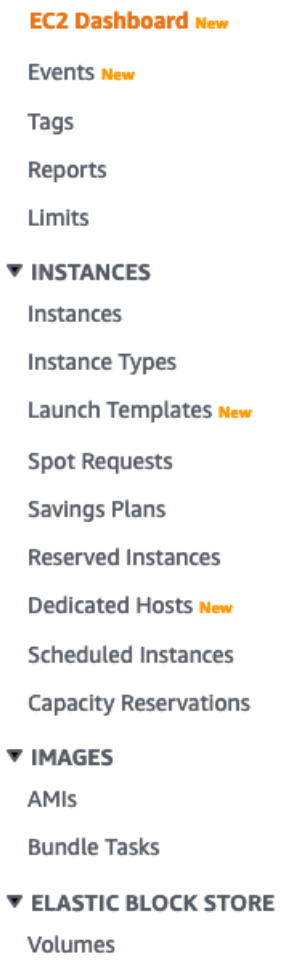
일반적으로, 키 필드가 동일한 다중 필터(예를 들어, tag:Name, search, Instance State)는 자동으로 OR로 조인됩니다. AND로 조인되면 대부분의 필더가 논리에 맞지 않기 때문에 이 설정은 의도적인 것입니다. 예를 들어, Instance State=running AND Instance State=stopped로 검색하면 검색 결과가 제공되지 않을 것입니다. 많은 경우에 서로 다른 키 필드에서 보완적인 검색 용어를 사용함으로써 검색 결과를 조정할 수 있고 이 경우 AND 규칙이 자동으로 대체 적용됩니다. tag: 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 AMI 또는 인스턴스 등 리소스에 해당하는 옵션을 선택합니다.



3. 이 페이지는 사용 가능한 모든 리소스를 표시합니다.

콘솔을 이용하여 리소스를 필터링

사용자는 Amazon EC2콘솔을 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 `MySG`를 입력합니다. 그러면 `MySG` 및`MySG2` 등 문자열의 일부로 `MySG3`가 있는 모든 값이 검색 결과에 표시됩니다. 결과를 MySG만으로 제한하려면 검색 필드에 `\bMySG\b`를 입력합니다. 유형이 `m1.small` 또는 `m1.large`인 모든 인스턴스를 목록화하려면 검색 필드에 `m1.small|m1.large`를 입력합니다.

상태가 **`us-east-1b`**인 **`available`** 가용 영역에서 볼륨을 목록화하려면

1. 탐색 창에서 볼륨을 선택합니다.
2. 검색 상자를 클릭하고 메뉴에서 연결 상태를 선택한 후 분리됨을 선택합니다. (분리된 볼륨은 동일 가용 영역에 있는 인스턴스에 연결하는 데 사용될 수 있습니다.)
3. 검색 상자를 다시 클릭한 후 상태를 선택하고 사용 가능을 선택합니다.
4. 검색 상자를 다시 클릭한 후 가용 영역을 선택하고 `us-east-1b`를 선택합니다.

5. 이 기준에 부합하는 모든 볼륨이 표시됩니다.

퍼블릭 64비트 Amazon EBS 지원 Linux AMI를 목록화하려면,

1. 탐색 창에서 AMI를 선택합니다.
2. 필터 창에서 퍼블릭 이미지, EBS 이미지를 선택한 후 필터 목록에서 Windows를 선택합니다.
3. 검색 필드에 `x86_64`을 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) 목록을 생성할 수 있습니다. 응답에는 모든 리소스에 대한 정보가 포함됩니다.

결과 리소스 목록은 길이가 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 0개 또는 1개의 문자에 해당합니다.

예를 들어 필터 값으로 `database`를 사용하면 설명이 `database`와 일치하는 EBS 스냅샷만이 표시됩니다. `*database*`를 지정한 경우 설명에 `database`가 포함된 모든 스냅샷이 반환됩니다. `database?`를 지정한 경우 설명이 다음 패턴(`database`와 일치하거나 `database` 뒤에 한 개의 문자가 따라옴) 중 하나와 일치하는 스냅샷만이 반환됩니다.

물음표의 수에 따라 결과에 포함될 최대 문자 수가 결정됩니다. 예를 들어 `database????`를 지정한 경우 설명이 `database`+4개의 문자와 일치하는 스냅샷만이 반환됩니다. 설명에 `database`+5개 이상의 문자가 포함된 경우 해당 스냅샷은 검색 결과에서 제외됩니다.

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 CLI Command Reference의 관련 `describe` 명령을 참조하십시오.
- Windows PowerShell의 경우 PowerShell용 AWS 도구 Cmdlet Reference의 관련 `Get` 명령을 참조하십시오.
- Query API의 경우 Amazon EC2 API Reference의 관련 `Describe` API 작업을 참조하십시오.

Amazon EC2리소스에 태그 지정

고유의 메타데이터를 태그의 형태로 각 리소스에 배정하면 인스턴스, 이미지 및 기타 Amazon EC2 리소스를 쉽게 관리할 수 있습니다. 이 주제에서는 태그를 설명하고 태그를 생성하는 방법을 보여 줍니다.

Warning

태그 키 및 해당 값은 다양한 API 호출에 의해 반환됩니다. `DescribeTags`에 대한 액세스를 거부해도 다른 API가 반환한 태그에 대한 액세스는 자동으로 거부되지 않습니다. 민감한 데이터를 태그에 포함하지 않는 것이 가장 좋습니다.

목차

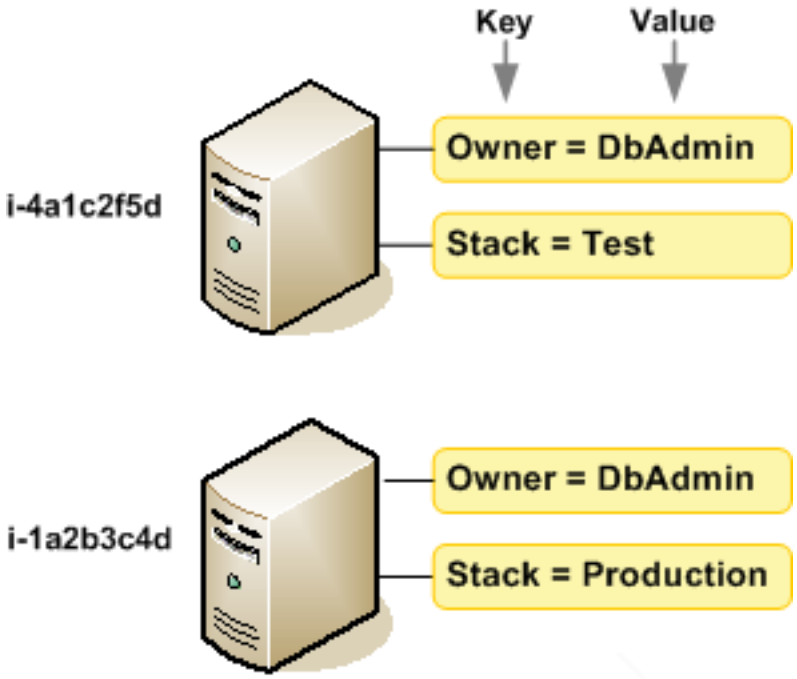
- 태그 기본 사항 (p. 1069)
- 리소스에 태그 지정 (p. 1070)
- 태그 제한 (p. 1072)
- 리소스에 결제용 태그 지정 (p. 1073)
- 콘솔을 사용한 태그 작업 (p. 1073)
- CLI 또는 API를 사용한 태그 작업 (p. 1076)

태그 기본 사항

태그란 AWS 리소스에 할당되는 레이블을 말합니다. 각 태그는 사용자가 정의하는 키와 선택적 값으로 구성됩니다.

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.— 예를 들어, 계정의 Amazon EC2 인스턴스에 대해 각 인스턴스의 소유자나 스택 수준을 추적하는 데 도움이 되는 태그 세트를 정의할 수 있습니다.

다음 다이어그램은 태그 지정 방식을 설명합니다. —이 예에서는 두 가지 태그, 즉 `Owner`라는 키가 있는 태그 하나와 `Stack`이라는 키가 있는 태그 하나를 각 인스턴스에 배정했습니다. 또한 각 태그에는 연결된 값이 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. 효과적인 리소스 태그 지정 전략을 구현하는 방법에 대한 자세한 내용은 AWS 백서 태그 지정 모범 사례를 참조하십시오.

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 콘솔, 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 EC2에 대한 자격 증명 및 액세스 관리 (p. 803) 단원을 참조하십시오.

리소스에 태그 지정

계정에 이미 존재하는 대부분의 Amazon EC2 리소스에 태그를 지정할 수 있습니다. 아래의 표 (p. 1070)에는 태그 지정을 지원하는 리소스가 나와 있습니다.

Amazon EC2 콘솔을 사용하는 경우, 관련 리소스 화면에서 태그 탭을 사용하여 리소스에 태그를 적용하거나 태그 화면을 사용할 수 있습니다. 일부 리소스 화면을 사용하면 리소스를 생성할 때 리소스에 대해 태그를 지정할 수 있습니다. 예를 들어 `Name`의 키가 있는 태그와 지정하는 값이 있습니다. 대부분의 경우, 콘솔은 리소스 생성 직후(리소스 생성 중이 아니라) 태그를 적용합니다. 콘솔은 `Name` 태그에 따라 리소스를 조직할 수 있지만 이 태그는 Amazon EC2 서비스에 대한 의미가 없습니다.

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 롤백됩니다. 이는 태그를 사용하여 리소스가 생성되거나 아예 리소스가 생성되지 않도록 하고 언제든 태그 지정되지 않은 리소스가 남지 않게 합니다. 생성 시 리소스에 태그를 지정하면 리소스 생성 후 사용자 지정 태그 지정 스크립트를 실행할 필요가 없습니다.

다음 표는 태그를 지정할 수 있는 Amazon EC2 리소스와 Amazon EC2 API, AWS CLI 또는 AWS SDK를 사용하여 생성 시 태그를 지정할 수 있는 리소스를 설명합니다.

Amazon EC2 리소스 태그 지정 지원

리소스	태그 지원	생성 시 태그 지정 지원
AFI	예	예
AMI	예	아니요
번들 작업	아니요	아니요
용량 예약	예	예
클라이언트 VPN 엔드포인트	예	예
클라이언트 VPN 경로	아니요	아니요
고객 게이트웨이	예	아니요
전용 호스트	예	예
전용 호스트 예약	예	아니요
DHCP 옵션	예	아니요

리소스	태그 지원	생성 시 태그 지정 지원
EBS 스냅샷	예	예
EBS 볼륨	예	예
EC2 집합	예	예
외부 전용 인터넷 게이트웨이	예	아니요
탄력적 IP 주소	예	아니요
Elastic Graphics 액셀러레이터	예	아니요
인스턴스	예	예
인스턴스 스토어 볼륨	해당 사항 없음	해당 사항 없음
인터넷 게이트웨이	예	아니요
IP 주소 풀(BYOIP)	예	아니요
키 페어	예	아니요
시작 템플릿	예	예
시작 템플릿 버전	아니요	아니요
로컬 게이트웨이	예	아니요
로컬 게이트웨이 라우팅 테이블	예	아니요
로컬 게이트웨이 가상 인터페이스	예	아니요
로컬 게이트웨이 가상 인터페이스 그룹	예	아니요
로컬 게이트웨이 라우팅 테이블 VPC 연결	예	아니요
로컬 게이트웨이 라우팅 테이블 가상 인터페이스 그룹 연결	예	아니요
NAT 게이트웨이	예	예
네트워크 ACL	예	아니요
네트워크 인터페이스	예	아니요
배치 그룹	예	아니요
예약 인스턴스	예	아니요
예약 인스턴스 목록	아니요	아니요
라우팅 테이블	예	아니요
스팟 플릿 요청	예	예
스팟 인스턴스 요청	예	아니요
보안 그룹	예	아니요

리소스	태그 지원	생성 시 태그 지정 지원
서브넷	예	아니요
트래픽 미러 필터	예	예
트래픽 미러 세션	예	예
트래픽 미러 대상	예	예
전송 게이트웨이	예	예
전송 게이트웨이 라우팅 테이블	예	예
전송 게이트웨이 VPC 연결	예	예
가상 프라이빗 게이트웨이	예	아니요
VPC	예	아니요
VPC 엔드포인트	예	예
VPC 엔드포인트 서비스	예	예
VPC 엔드포인트 서비스 구성	예	예
VPC 흐름 로그	예	예
VPC 피어링 연결	예	아니요
VPN 연결	예	아니요

Amazon EC2 콘솔에서 Amazon EC2 인스턴스 시작 마법사를 사용하여 생성할 때 인스턴스와 볼륨에 태그를 지정할 수 있습니다. 볼륨 화면을 사용하여 생성 시 EBS 볼륨에 태그를 지정하거나 스냅샷 화면을 사용하여 EBS 스냅샷에 태그를 지정할 수 있습니다. 또는 리소스를 만들 때 리소스 생성 Amazon EC2 API(예: RunInstances)를 사용하여 태그를 적용하십시오.

생성 시 태그를 지원하는 Amazon EC2 API 작업에 IAM 정책의 태그 기반 리소스 수준 권한을 적용하여 생성 시 리소스에 태그를 지정할 수 있는 사용자와 그룹을 세밀하게 제어할 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 예제: 리소스에 태그 지정 (p. 836) 단원을 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
- 각 리소스에 대해 각 태그 키는 고유하며 하나의 값만 가질 수 있습니다.
- 최대 키 길이 – UTF-8의 유니코드 문자 128자
- 최대 값 길이 – 유니코드 문자 256자(UTF-8)

- EC2는 태그에 모든 문자를 사용할 수 있지만, 다른 서비스에는 제한이 적용되기도 합니다. 서비스에서 허용되는 문자는 UTF-8로 표현할 수 있는 문자, 숫자 및 공백과 특수 문자 + - = . _ : / @입니다.
- 태그 키와 값은 대/소문자를 구분합니다.
- `aws:` 접두사는 AWS용으로 예약되어 있습니다. 태그에 이 접두사가 있는 태그 키가 있는 경우 태그의 키 또는 값을 편집하거나 삭제할 수 없습니다. `aws:` 접두사가 지정된 태그는 리소스당 태그 수 제한에 포함되지 않습니다.

태그에만 기초하여 리소스를 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. 1070) 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 내용은 AWS Billing and Cost Management 사용 설명서의 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 사용을 참조하십시오.

> **Note**
>
> 방금 보고서를 활성화한 경우, 24시간 후에 이번 달의 데이터를 볼 수 있습니다.

비용 할당 태그는 비용에 기여하는 리소스를 나타낼 수 있지만, 리소스를 삭제하거나 비활성화한다고 해서 비용이 항상 절감되는 것은 아닙니다. 예를 들어, 원본 데이터가 포함된 스냅샷을 삭제하더라도 다른 스냅샷에서 참조하는 스냅샷 데이터는 보존됩니다. 자세한 내용은 AWS Billing and Cost Management 사용 설명서의 Amazon Elastic Block Store 볼륨 및 스냅샷을 참조하십시오.

> **Note**
>
> 태그된 탄력적 IP 주소는 비용 할당 보고서에 나타나지 않습니다.

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. 1065) 단원을 참조하십시오.

태그를 중앙에서 통합 생성 및 관리할 수 있는 AWS Management 콘솔의 Tag Editor는 사용하기 쉽고 최상의 결과를 발휘합니다. 자세한 내용은 AWS Management Console 시작하기의 Tag Editor 작업을 참조하십시오.

내용

- 태그 표시 (p. 1074)

- 개별 리소스에 대한 태그 추가 및 삭제 (p. 1074)
- 리소스 그룹에 태그 추가 및 삭제 (p. 1075)
- 인스턴스 시작 시 태그 추가 (p. 1076)
- 태그를 기준으로 리소스 목록 필터링 (p. 1076)

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 태그 표시

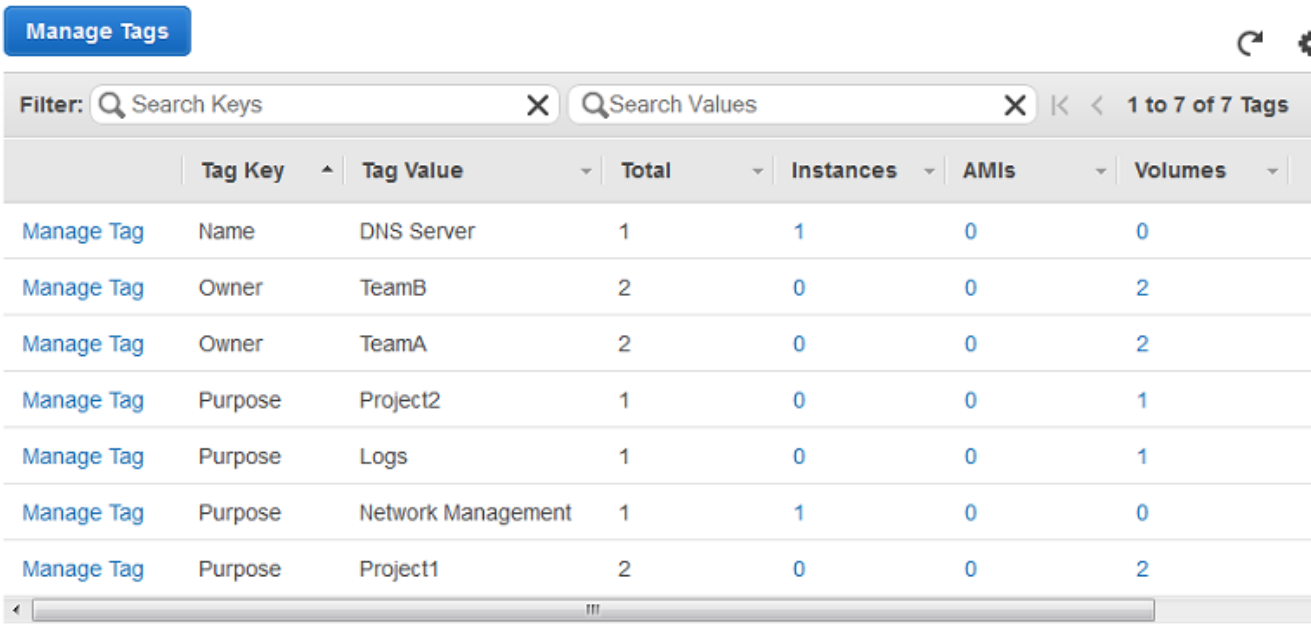
Amazon EC2 콘솔에서 리소스 관련 페이지를 선택하면 이러한 리소스의 목록이 표시됩니다. 예를 들어, 탐색 창에서 인스턴스를 선택하는 경우 콘솔에 Amazon EC2 인스턴스 목록이 표시됩니다. 이러한 목록 중 하나(예: 인스턴스)에서 리소스를 선택하는 경우 해당 리소스에서 태그를 지원하면 관련 태그를 보고 관리할 수 있습니다. 대부분의 리소스 페이지에서는 세부 정보 창의 태그 탭에서 태그를 볼 수 있습니다.

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.

- 태그 탭에서 열 표시를 선택합니다. 새 열이 콘솔에 추가됩니다.
- 열 표시/숨기기 기어 모양 아이콘을 선택하고 열 표시/숨기기 대화 상자의 태그 키에서 태그 키를 선택합니다.

모든 리소스에 대한 태그 표시

Amazon EC2 콘솔의 탐색 창에서 태그를 선택하여 모든 리소스의 태그를 표시할 수 있습니다. 다음 이미지는 리소스 유형별로 사용 중인 모든 태그를 나열하는 태그 창을 보여줍니다.



	Tag Key	Tag Value	Total	Instances	AMIs	Volumes
Manage Tag	Name	DNS Server	1	1	0	0
Manage Tag	Owner	TeamB	2	0	0	2
Manage Tag	Owner	TeamA	2	0	0	2
Manage Tag	Purpose	Project2	1	0	0	1
Manage Tag	Purpose	Logs	1	0	0	1
Manage Tag	Purpose	Network Management	1	1	0	0
Manage Tag	Purpose	Project1	2	0	0	2

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 정보는 리소스 위치 (p. 1059) 단원을 참조하십시오.
3. 탐색 창에서 리소스 유형(예: 인스턴스])을 선택합니다.
4. 리소스 목록에서 리소스를 선택한 다음 태그, 태그 추가/편집을 선택합니다.
5. 태그 추가/편집 대화 상자에서 각 태그에 대한 키와 값을 지정하고 저장을 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 정보는 리소스 위치 (p. 1059) 단원을 참조하십시오.
3. 탐색 창에서 리소스 유형(예: 인스턴스)을 선택합니다.
4. 리소스 목록에서 리소스를 선택하고 태그를 선택합니다.
5. 태그 추가/편집을 선택하고 태그의 삭제 아이콘을 선택한 다음 저장을 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 정보는 리소스 위치 (p. 1059) 단원을 참조하십시오.
3. 탐색 창에서 태그를 선택합니다.
4. 콘텐츠 창 맨 위에서 태그 관리를 선택합니다.
5. 필터에서 태그를 추가할 리소스 유형(예: 인스턴스)을 선택합니다.
6. 리소스 목록에서 태그를 추가할 각 리소스 옆의 확인란을 선택합니다.
7. 태그 추가에서 키 및 값에 태그 키와 값을 입력한 다음 태그 추가를 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 정보는 리소스 위치 (p. 1059) 단원을 참조하십시오.
3. 탐색 창에서 태그와 태그 관리를 선택합니다.
4. 사용 중인 태그를 보려면 열 표시/숨기기 기어 모양 아이콘을 선택하고, 열 표시/숨기기 대화 상자에서 조회할 태그 키를 선택한 다음 닫기를 클릭합니다.
5. 필터에서 태그를 제거할 리소스 유형(예: 인스턴스)을 선택합니다.
6. 리소스 목록에서 태그를 제거할 각 리소스 옆에 있는 확인란을 선택합니다.
7. 태그 제거의 키 상자에 태그의 이름을 입력한 다음 태그 제거를 선택합니다.

인스턴스 시작 시 태그 추가

마법사 시작을 사용하여 태그를 추가하려면

1. 탐색 모음에서 인스턴스를 시작할 지역을 선택합니다. 일부 Amazon EC2 리소스는 리전 간에 공유될 수 있지만 그렇지 않은 리소스도 있으므로 잘 선택해야 합니다. 요구 사항을 충족하는 리전을 선택합니다. 자세한 정보는 리소스 위치 (p. 1059) 단원을 참조하십시오.
2. 인스턴스 시작을 선택합니다.
3. Amazon 머신 이미지(AMI) 선택 페이지에서는 Amazon 머신 이미지(AMI)라 불리는 일련의 기본 구성들을 목록으로 표시합니다. 사용할 AMI를 선택하고 선택을 선택합니다. AMI 선택에 대한 자세한 내용은 Linux AMI 찾기 (p. 96) 단원을 참조하십시오.
4. 인스턴스 세부 정보 구성 페이지에서 필요에 따라 인스턴스 설정을 구성하고 다음: 스토리지 추가(Next: Add Storage)를 선택합니다.
5. 스토리지 추가 페이지에서 인스턴스에 대한 추가 스토리지 볼륨을 지정할 수 있습니다. 모두 마쳤으면 다음: 태그 추가(Next: Add Tags)를 선택합니다.
6. 태그 추가 페이지에서 인스턴스나 볼륨 또는 이 둘의 태그를 지정합니다. 태그 추가를 선택하여 리소스에 한 개 이상의 태그를 인스턴스에 추가할 수 있습니다. 모두 마쳤으면 다음: 보안 그룹 구성(Next: Configure Security Group)을 선택합니다.
7. 보안 그룹 구성 페이지에서 소유하는 기존 보안 그룹 중 하나를 선택하거나 마법사를 통해 새 보안 그룹을 생성합니다. 작업을 마치면 검토 후 시작(Review and Launch)을 선택합니다.
8. 설정을 검토합니다. 선택한 항목에 만족하면 시작을 선택합니다. 기존 키 페어를 선택하거나 새 키 페어를 생성하고, 승인 확인란을 선택하고 인스턴스 시작을 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. 세부 정보 창에서 태그를 선택합니다.
   c. 목록에서 태그를 찾고 열 표시를 선택합니다.
2. 태그 열의 오른쪽 위에 있는 필터 아이콘을 선택하여 필터 목록을 표시합니다.
3. 태그 값을 선택하고 필터 적용을 선택하여 결과 목록을 필터링합니다.

   Note

   필터에 대한 자세한 내용은 리소스 목록화 및 필터링 (p. 1065) 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 덮어씁니다.	create-tags	New-EC2Tag	CreateTags

작업	AWS CLI	Windows PowerShell용 AWS 도구	API 작업
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 보여줍니다.

> **Note**
>
> JSON 형식의 파라미터를 명령줄에 입력하는 방법은 운영 체제에 따라 다릅니다. Linux, macOS 또는 Unix 및 Windows PowerShell에서는 작은따옴표(')를 사용하여 JSON 데이터 구조를 묶습니다. Windows 명령줄에서 명령을 사용할 때 작은따옴표를 사용하지 마십시오. 자세한 정보는 AWS 명령줄 인터페이스에 대한 파라미터 값 지정을 참조하십시오.

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	New-EC2Instance	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}]'
```

예제 6: 리소스에 태그 추가

이 예제에서는 지정된 이미지에 태그 `Stack=production`을 추가하거나 태그 키가 `Stack`인 AMI의 기존 태그를 덮어씁니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

```
aws ec2 create-tags --resources ami-78a54011 --tags Key=Stack,Value=production
```

예제 7: 여러 리소스에 태그 추가

이 예제에서는 AMI와 인스턴스에 대해 두 개의 태그를 추가(또는 덮어쓰기)합니다. 태그 중 하나에 값이 없는 키(`webserver`)만 포함되어 있습니다(값을 빈 문자열로 설정). 다른 태그는 키(`stack`)와 값(`Production`)으로 구성됩니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

```
aws ec2 create-tags --resources ami-1a2b3c4d i-1234567890abcdef0 --tags
 Key=webserver,Value=  Key=stack,Value=Production
```

예제 8: 특수 문자에 태그 추가

이 예제에서는 `[Group]=test` 태그를 인스턴스에 추가합니다. 대괄호(`[` 및 `]`)는 특수 문자이므로 백슬래시(`\`)로 이스케이프되어야 합니다.

```
aws ec2 create-tags --resources i-1234567890abcdef0 --tags Key=\[Group]\,Value=test
```

Windows PowerShell을 사용하는 경우 문자를 백슬래시(`\`)로 분리하고 큰 따옴표(`"`)로 묶은 다음 전체 키 및 값 구조를 작은 따옴표(`'`)로 묶으십시오.

```
aws ec2 create-tags --resources i-1234567890abcdef0 --tags 'Key=\"[Group]\",Value=test'
```

Linux 또는 OS X를 사용하는 경우 전체 키 및 값 구조를 작은 따옴표(`'`)로 묶은 다음 특수 문자를 포함하는 요소를 큰 따옴표(`"`)로 묶으십시오.

```
aws ec2 create-tags --resources i-1234567890abcdef0 --tags 'Key="[Group]",Value=test'
```

Amazon EC2 서비스 제한

Amazon EC2는 사용자가 사용할 수 있는 서로 다른 리소스를 제공합니다. 이러한 리소스로는 이미지, 인스턴스, 볼륨 및 스냅샷이 있습니다. AWS 계정을 생성하면 이러한 리소스에 대한 기본 제한(할당량이라고도 함)이 리전별로 설정됩니다. 예를 들어 한 리전에서 시작할 수 있는 인스턴스 수에는 제한이 있습니다. 즉, 미국 서부(오레곤) 리전에서 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. 실제로 필요할 시점보다 미리 제한 증가를 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 제한 페이지에서는 리전별로 Amazon EC2 및 Amazon VPC에서 제공하는 리소스의 현재 제한을 볼 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 제한을 선택합니다.
4. 목록에서 리소스를 찾습니다. 검색 필드를 사용하여 리소스 이름이나 리소스 그룹으로 목록을 필터링할 수 있습니다. 현재 제한 열에는 해당 리소스에 대한 계정의 현재 최대값이 표시됩니다.

제한 증가 요청

Amazon EC2 콘솔의 제한 페이지에서 Amazon EC2 또는 Amazon VPC에서 리전별로 제공하는 리소스에 대한 제한 증가를 요청할 수 있습니다.

제한 증가를 요청하려면 다음을 수행합니다.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 모음에서 리전을 선택합니다.
3. 탐색 창에서 제한을 선택합니다.
4. 목록에서 리소스를 선택하고 사용량 제한 증가 요청을 선택합니다.
5. 제한 증가 양식에서 필수 필드를 기재합니다. 응답은 사용자가 지정한 연락 수단을 통해 제공됩니다.

포트 25를 사용하여 전송되는 이메일 관련 제한

Amazon EC2는 기본적으로 모든 인스턴스에 대해 포트 25에서 트래픽을 제한합니다. 이 제한을 제거하도록 요청할 수 있습니다. 자세한 내용은 AWS Knowledge Center의 EC2 인스턴스에서 포트 25의 제한을 제거할 수 있습니까?를 참조하십시오.

Amazon EC2 사용 보고서

AWS는 EC2 인스턴스의 비용 및 사용, 예약 인스턴스의 사용을 분석할 수 있는 Cost Explorer라는 무료 보고 도구를 제공합니다.

Cost Explorer는 사용 및 비용을 차트로 보기 위해 사용할 수 있는 무료 도구입니다. 지난 13개월까지의 데이터를 볼 수 있고 향후 3개월 동안의 지출을 예상해볼 수 있습니다. Cost Explorer를 사용하면 시간 경과에 따라 AWS 리소스에 지출하는 금액의 패턴을 보고 추가 질의가 필요한 영역을 식별하며 비용을 이해하는 데 사용할 수 있는 추세를 알아볼 수 있습니다. 또한 데이터의 시간 범위를 지정하고 일별 또는 월별 시간 데이터를 볼 수도 있습니다.

Cost Explorer를 사용할 경우 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 걸차 사용자의 인스턴스 사용이 어떻게 분포되어 있습니까?
- AWS 계정에 걸차 사용자의 인스턴스 사용이 어떻게 분포되어 있습니까?
- 예약 인스턴스를 얼마나 잘 사용하고 있습니까?
- 예약 인스턴스를 통해 비용을 절약하고 있습니까?

Cost Explorer에서 Amazon EC2 보고서를 보려면

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

   Cost Explorer에서 보고서가 열립니다. 고정된 필터 설정에 따라 사용 및 비용 추세에 대한 정보를 표시하는 미리 구성된 보기가 제공됩니다.

   > **Note**
   >
   > Cost Explorer를 사용하려면 먼저 계정에서 Cost Explorer를 활성화해야 합니다. 자세한 내용은 Cost Explorer 활성화를 참조하십시오.

보고서 저장을 포함하여 Cost Explorer에서 보고서 작업에 대한 자세한 정보는 Cost Explorer로 비용 분석을 참조하십시오.

인스턴스 문제 해결

다음 문서는 인스턴스 관련 문제를 해결하는 데 도움이 됩니다.

내용

- 인스턴스 시작 문제 해결 (p. 1083)
- 인스턴스 연결 문제 해결 (p. 1085)
- 인스턴스 중지 문제 해결 (p. 1092)
- 인스턴스 종료 문제 해결 (p. 1093)
- 상태 확인에 실패한 인스턴스 문제 해결 (p. 1094)
- 연결할 수 없는 인스턴스 문제 해결 (p. 1115)
- 잘못된 볼륨에서 부팅 (p. 1117)
- 사용 Linux용 EC2Rescue (p. 1119)
- 진단 인터럽트 보내기(고급 사용자 전용) (p. 1128)

Windows 인스턴스와 관련하여 추가적인 도움이 필요하면 Windows 인스턴스용 Amazon EC2 사용 설명서의 Windows 인스턴스 문제 해결을 참조하십시오.

인스턴스 시작 문제 해결

다음 문제 때문에 인스턴스를 시작할 수 없습니다.

시작 문제

- 인스턴스 제한을 초과함 (p. 1083)
- 부족한 인스턴스 용량 (p. 1084)
- 인스턴스 즉시 종료 (p. 1084)

인스턴스 제한을 초과함

설명

새 인스턴스를 시작하려 할 때 혹은 중지된 인스턴스를 다시 시작하려 할 때 `InstanceLimitExceeded` 오류가 발생합니다.

원인

새 인스턴스를 시작하거나 중지된 인스턴스를 다시 시작하려 할 때 `InstanceLimitExceeded` 오류가 발생하면, 한 리전에서 시작할 수 있는 인스턴스 한도에 도달한 것입니다. AWS 계정을 생성할 때 리전별로 실행할 수 있는 인스턴스의 기본 한도가 설정됩니다.

솔루션

리전을 기준으로 인스턴스 한도 증가를 요청할 수 있습니다. 자세한 내용은 Amazon EC2 서비스 제한 (p. 1079) 단원을 참조하십시오.

부족한 인스턴스 용량

설명

새 인스턴스를 시작하려 할 때 혹은 중지된 인스턴스를 다시 시작하려 할 때 `InsufficientInstanceCapacity` 오류가 발생합니다.

원인

인스턴스를 시작하거나 중지된 인스턴스를 다시 시작하려 할 때 `InsufficientInstanceCapacity` 오류가 발생하면 현재 AWS에 요청에 대한 서비스를 제공할 수 있을 만큼 가용 온디맨드 용량이 충분하지 않은 것입니다.

솔루션

다음에 따라 문제를 해결하십시오.

- 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 유형 변경 (p. 262) 단원을 참조하십시오.
- 클러스터 배치 그룹으로 인스턴스를 시작한 경우 용량 부족 오류가 발생할 수 있습니다. 자세한 내용은 배치 그룹 규칙 및 제한 사항 (p. 760) 단원을 참조하십시오.
- 어떤 기간에 대해서든 Amazon EC2 용량을 예약할 수 있는 온디맨드 용량 예약을 생성해 보십시오. 자세한 내용은 온디맨드 용량 예약 (p. 415)를 참조하십시오.
- 장기 용량 예약인 예약 인스턴스를 구입해 보십시오. 자세한 정보는 Amazon EC2 예약 인스턴스를 참조하십시오.

인스턴스 즉시 종료

설명

인스턴스가 다시 시작한 후 `pending` 상태에서 `terminated` 상태로 즉시 변경됩니다.

원인

인스턴스가 즉시 종료되는 이유에는 다음과 같이 몇 가지가 있습니다.

- EBS 볼륨 제한에 도달했습니다.
- EBS 스냅샷이 손상되었습니다.
- 루트 EBS 볼륨이 암호화되었는데 사용자는 암호 해독을 위하여 KMS 키에 액세스할 권한이 없습니다.
- 인스턴스를 시작하는 데 사용한 인스턴스 스토어 지원 AMI에 필수 부분(image.part.xx 파일).

솔루션

Amazon EC2 콘솔 또는 AWS Command Line Interface를 사용하여 종료 이유를 파악할 수 있습니다.

Amazon EC2 콘솔을 사용해서 종료 이유를 파악하는 방법

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

2. 탐색 창에서 인스턴스를 선택하고 인스턴스를 선택합니다.
3. 설명 탭의 상태 전환 이유 레이블 옆에서 이유를 확인합니다.

AWS Command Line Interface을 사용해서 종료 이유를 파악하는 방법

1. describe-instances 명령을 사용하여 인스턴스 ID를 지정합니다.

```
aws ec2 describe-instances --instance-id instance_id
```

2. 명령으로 반환된 JSON 응답을 검토하고 `StateReason` 응답 요소의 값을 확인합니다.

   다음 코드 블록은 `StateReason` 응답 요소의 예를 보여 줍니다.

```
"StateReason": {
  "Message": "Client.VolumeLimitExceeded: Volume limit exceeded",
  "Code": "Server.InternalError"
},
```

이 문제를 해결하려면

확인한 종료 이유에 따라 다음 작업 중 하나를 선택합니다.

- 이유가 **`Client.VolumeLimitExceeded: Volume limit exceeded`**인 경우, EBS 볼륨 한도에 도달한 것입니다. 자세한 내용은 인스턴스 볼륨 제한 (p. 1046) 단원을 참조하십시오. Amazon EBS 볼륨 제한 증가 요청을 제출하려면 AWS Support 센터에서 사례 생성 양식을 작성하십시오. 자세한 내용은 Amazon EC2 서비스 제한 (p. 1079) 단원을 참조하십시오.
- 이유가 **`Client.InternalError: Client error on launch`**인 경우, 루트 볼륨이 암호화되었는데 사용자는 암호 해독을 위하여 KMS 키에 액세스할 권한이 없는 것입니다. 필요한 KMS 키에 액세스할 권한을 얻으려면 IAM 사용자에게 해당 KMS 권한을 추가하십시오. 자세한 정보는 AWS Key Management Service Developer Guide의 AWS KMS에서 키 정책 사용을 참조하십시오.

인스턴스 연결 문제 해결

다음은 인스턴스에 연결하는 중에 발생할 수 있는 문제와 오류 메시지입니다.

내용

- 인스턴스 연결 중 오류 발생: 연결 시간 초과 (p. 1086)
- 오류: ... 키를 로드할 수 없습니다. 예상: 모든 프라이빗 키 (p. 1087)
- 오류: 서버에서 사용자 키를 인식하지 못함 (p. 1088)
- 오류: 호스트 키를 찾을 수 없음. 권한 거부(퍼블릭 키) 또는 인증 실패, 권한 거부 (p. 1089)
- 오류: 보호되지 않는 프라이빗 키 파일 (p. 1090)
- 오류: 프라이빗 키는 '-----BEGIN RSA PRIVATE KEY-----'로 시작하고 '-----END RSA PRIVATE KEY-----'로 끝나야 합니다. (p. 1090)
- 오류: 서버에서 키 거부 또는 지원되는 인증 방법이 없음 (p. 1091)
- 내 브라우저를 사용하여 연결할 수 없음 (p. 1091)
- 인스턴스를 ping할 수 없음 (p. 1091)
- 오류: 서버에서 예기치 않게 네트워크 연결을 차단함 (p. 1092)

Windows 인스턴스와 관련하여 추가적인 도움이 필요하면 Windows 인스턴스용 Amazon EC2 사용 설명서의 Windows 인스턴스 문제 해결을 참조하십시오.

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 인스턴스를 선택한 다음 인스턴스를 선택합니다.
  3. 콘솔 페이지 하단의 설명 탭에서 보안 그룹 옆에 있는 인바운드 규칙 보기를 선택하여 선택한 인스턴스에 대해 유효한 규칙 목록을 표시합니다.
  4. Linux 인스턴스의 경우: 인바운드 규칙 보기를 선택하면 트래픽이 허용되는 포트가 표시된 창이 나타납니다. 해당 컴퓨터에서 포트 22(SSH)로의 트래픽을 허용하는 규칙이 있는지 확인합니다.

     Windows 인스턴스의 경우: 인바운드 규칙 보기를 선택하면 트래픽이 허용되는 포트가 표시된 창이 나타납니다. 해당 컴퓨터에서 포트 3389(RDP)로의 트래픽을 허용하는 규칙이 있는지 확인합니다.

     인스턴스를 다시 시작할 때마다 새 IP 주소(와 호스트 이름)가 할당됩니다. 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. 859) 단원을 참조하십시오.

     보안 그룹 규칙에 대한 자세한 내용은 Amazon VPC 사용 설명서의 보안 그룹에 규칙을 참조하십시오.
- 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 인스턴스를 선택한 다음 인스턴스를 선택합니다.
  3. 설명 탭에서 VPC ID와 서브넷 ID의 값을 기록합니다.
  4. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
  5. 탐색 창에서 인터넷 게이트웨이를 선택합니다. VPC에 인터넷 게이트웨이가 연결되어 있는지 확인합니다. 또는 인터넷 게이트웨이 생성을 선택하여 인터넷 게이트웨이를 만듭니다. 인터넷 게이트웨이를 선택한 후 VPC에 연결을 선택하고 지침에 따라 VPC에 연결합니다.
  6. 탐색 창에서 서브넷을 선택한 후 해당 서브넷을 선택합니다.
  7. 라우팅 테이블 탭에서 대상 위치로 `0.0.0.0/0` 경로가 있으며, VPC의 대상으로 해당 인터넷 게이트웨이가 있는지 확인합니다. IPv6 주소를 이용해 인스턴스에 연결하는 경우 인터넷 게이트웨이를 가리키는 모든 IPv6 트래픽(`::/0`)에 대한 경로가 있는지 확인합니다. 그렇지 않으면 다음을 수행하십시오.

     a. 라우팅 테이블의 ID(rtb-xxxxxxxx)를 선택해 해당 라우팅 테이블로 이동합니다.
     b. 라우팅 탭에서 라우팅 편집을 선택합니다. 라우팅 추가를 선택하고 대상 위치로 `0.0.0.0/0`을, 대상으로 인터넷 게이트웨이를 사용합니다. IPv6의 경우 라우팅 추가를 선택하고 대상 위치로 `::/0`을, 대상으로 인터넷 게이트웨이를 사용합니다.
     c. 라우팅 저장을 선택합니다.
- 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 서브넷을 선택하고 해당 서브넷을 선택합니다.

3. 설명 탭에서 네트워크 ACL을 찾아 ID(acl-xxxxxxxx)를 선택합니다.
4. 네트워크 ACL을 선택합니다. 인바운드 규칙 탭에서 규칙이 해당 컴퓨터로부터의 트래픽을 허용하는지 확인합니다. 허용하지 않을 경우 해당 컴퓨터로부터의 트래픽을 차단하는 규칙을 삭제하거나 수정합니다.
5. 아웃바운드 규칙 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.

  컴퓨터에 방화벽이 있을 경우 이 방화벽에서 해당 컴퓨터의 포트 22(Linux 인스턴스) 또는 포트 3389(Windows 인스턴스)로부터의 트래픽을 허용하는지 확인합니다.
- 인스턴스에 퍼블릭 IPv4 주소가 있는지 확인합니다. 퍼블릭 IP 주소가 없을 경우 인스턴스와 탄력적 IP 주소를 연결할 수 있습니다. 자세한 내용은 탄력적 IP 주소 (p. 676) 단원을 참조하십시오.
- 인스턴스에서 CPU 부하를 확인합니다. 서버 과부하가 발생했을 수 있습니다. AWS는 Amazon CloudWatch 지표 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. 616) 단원을 참조하십시오.
  - 부하가 가변적이면 Auto Scaling 및 Elastic Load Balancing을 사용하여 인스턴스를 자동으로 확장하거나 축소할 수 있습니다.
  - 부하가 꾸준히 증가하는 경우 더 큰 인스턴스 유형으로 전환할 수 있습니다. 자세한 내용은 인스턴스 유형 변경 (p. 262) 단원을 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 구성하기를 참조하십시오.
- 로컬 컴퓨터에 IPv6 주소가 있고 IPv6를 사용하도록 컴퓨터를 구성해야 합니다.

오류: ... 키를 로드할 수 없습니다. 예상: 모든 프라이빗 키

인스턴스에 연결하여 오류 메시지 `unable to load key ... Expecting: ANY PRIVATE KEY`를 수신하려고 할 때 프라이빗 키가 저장된 파일이 잘못 구성되었습니다. 프라이빗 키 파일은 `.pem`으로 끝날 경우에도 여전히 잘못 구성될 수 있습니다. 인증서가 누락된 이유로 프라이빗 키 파일이 잘못 구성될 수 있습니다.

프라이빗 키 파일이 잘못 구성된 경우, 다음 단계에 따라 오류를 해결하십시오.

1. 새 키 페어를 생성합니다. 자세한 내용은 Amazon EC2를 사용해 키 페어 생성 (p. 862) 단원을 참조하십시오.
2. 새 키 페어를 인스턴스에 추가합니다. 자세한 내용은 프라이빗 키를 분실했을 때 Linux 인스턴스에 연결하는 방법 (p. 868) 단원을 참조하십시오.
3. 새 키 페어를 사용하여 인스턴스에 연결합니다.

오류: 서버에서 사용자 키를 인식하지 못함

SSH를 사용하여 인스턴스에 연결하는 경우

- 연결 시 `ssh -vvv`를 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: boguspem.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: boguspem.pem
debug1: read PEM private key done: type RSA
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).
```

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. 498) 단원을 참조하십시오.

  Note

  PuTTYgen에서 프라이빗 키 파일을 불러온 후 생성이 아니라 프라이빗 키 저장(Save Private Key)을 선택합니다.
- AMI에 적합한 사용자 이름을 사용하여 연결하고 있는지 확인합니다. PuTTY 구성(PuTTY Configuration) 창에서 호스트 이름(Host name) 상자에 사용자 이름을 입력합니다.
  - Amazon Linux 2 또는 Amazon Linux AMI의 경우 사용자 이름은 `ec2-user`입니다.
  - CentOS AMI의 경우 사용자 이름은 `centos`입니다.
  - Debian AMI의 경우 사용자 이름은 `admin` 또는 `root`입니다.

  - Fedora AMI의 경우 사용자 이름은 `ec2-user` 또는 `fedora`입니다.
  - RHEL AMI의 경우 사용자 이름은 `ec2-user` 또는 `root`입니다.
  - SUSE AMI의 경우 사용자 이름은 `ec2-user` 또는 `root`입니다.
  - Ubuntu AMI의 경우 사용자 이름은 `ubuntu`입니다.
  - `ec2-user` 및 `root`를 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. 859) 단원을 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 있는지 또한 인스턴스에 대한 올바른 프라이빗 키(`.pem`)) 파일을 지정했는지 확인합니다.

적합한 사용자 이름은 다음과 같습니다.

- Amazon Linux 2 또는 Amazon Linux AMI의 경우 사용자 이름은 `ec2-user`입니다.
- CentOS AMI의 경우 사용자 이름은 `centos`입니다.
- Debian AMI의 경우 사용자 이름은 `admin` 또는 `root`입니다.
- Fedora AMI의 경우 사용자 이름은 `ec2-user` 또는 `fedora`입니다.
- RHEL AMI의 경우 사용자 이름은 `ec2-user` 또는 `root`입니다.
- SUSE AMI의 경우 사용자 이름은 `ec2-user` 또는 `root`입니다.
- Ubuntu AMI의 경우 사용자 이름은 `ubuntu`입니다.
- `ec2-user` 및 `root`를 사용할 수 없는 경우 AMI 공급자에게 문의하십시오.

예를 들어 SSH 클라이언트를 사용하여 Amazon Linux 인스턴스에 연결하려면 다음 명령을 사용합니다.

```
ssh -i /path/my-key-pair.pem ec2-user@public-dns-hostname
```

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. 설명 탭에서 키 페어 이름의 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. 868) 단원을 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 홈 디렉토리에서의 권한이 변경되었을 수 있습니다. `/home/ec2-user/.ssh/authorized_keys`에 대한 권한은 소유자로만 제한되어야 합니다.

인스턴스에서 권한을 확인하려면

1. 인스턴스를 중지하고 루트 볼륨을 분리합니다. 자세한 내용은 인스턴스 중지 및 시작 (p. 507) 및 인스턴스에서 Amazon EBS 볼륨 분리 (p. 924) 단원을 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. 910) 단원을 참조하십시오.
3. 임시 인스턴스에 연결하고 마운트 지점을 생성한 후 연결한 볼륨을 마운트합니다. 자세한 내용은 Amazon EBS 볼륨을 Linux에서 사용할 수 있도록 만들기 (p. 914) 단원을 참조하십시오.
4. 임시 인스턴스에서 연결된 볼륨의 `/home/ec2-user/` 디렉토리의 권한을 확인합니다. 필요하다면 다음과 같이 권한을 조정합니다.

```
[ec2-user ~]$ chmod 600 mount_point/home/ec2-user/.ssh/authorized_keys
```

```
[ec2-user ~]$ chmod 700 mount_point/home/ec2-user/.ssh
```

```
[ec2-user ~]$ chmod 700 mount_point/home/ec2-user
```

5. 볼륨을 마운트 해제하고 임시 인스턴스에서 분리한 다음 원본 인스턴스에 다시 연결합니다. 루트 볼륨에 올바른 이름을 지정해야 합니다(예: `/dev/xvda`).
6. 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 모든 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` 및 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.

```
[ec2-user ~]$ chmod 0400 .ssh/my_private_key.pem
```

오류: 프라이빗 키는 '-----BEGIN RSA PRIVATE KEY-----'로 시작하고 '-----END RSA PRIVATE KEY-----'로 끝나야 합니다.

ssh-keygen과 같은 타사 도구를 사용하여 RSA 키 페어를 생성하는 경우 OpenSSH 키 형식으로 프라이빗 키가 생성됩니다. 인스턴스에 연결할 때 암호를 해독하기 위해 OpenSSH 형식의 프라이빗 키를 사용하면 `Private key must begin with "-----BEGIN RSA PRIVATE KEY-----" and end with "-----END RSA PRIVATE KEY-----"` 오류가 발생합니다.

이 오류를 해결하려면 프라이빗 키가 PEM 형식이어야 합니다. 다음 명령을 사용하여 PEM 형식의 프라이빗 키를 생성합니다.

```
ssh-keygen -m PEM
```

오류: 서버에서 키 거부 또는 지원되는 인증 방법이 없음

PuTTY를 사용하여 인스턴스에 연결하고 Error: Server refused our key(오류: 서버에서 키 거부) 또는 Error: No supported authentication methods available(오류: 지원되는 인증 방법이 없음) 중 하나가 발생하면 AMI에 적합한 사용자 이름으로 연결했는지 확인하십시오. PuTTY Configuration(PuTTY 구성) 창에서 사용자 이름에 사용자 이름을 입력합니다.

적합한 사용자 이름은 다음과 같습니다.

- Amazon Linux 2 또는 Amazon Linux AMI의 경우 사용자 이름은 `ec2-user`입니다.
- CentOS AMI의 경우 사용자 이름은 `centos`입니다.
- Debian AMI의 경우 사용자 이름은 `admin` 또는 `root`입니다.
- Fedora AMI의 경우 사용자 이름은 `ec2-user` 또는 `fedora`입니다.
- RHEL AMI의 경우 사용자 이름은 `ec2-user` 또는 `root`입니다.
- SUSE AMI의 경우 사용자 이름은 `ec2-user` 또는 `root`입니다.
- Ubuntu AMI의 경우 사용자 이름은 `ubuntu`입니다.
- `ec2-user` 및 `root`를 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. 498) 단원을 참조하십시오.

내 브라우저를 사용하여 연결할 수 없음

Amazon EC2 콘솔은 Java SSH 클라이언트를 사용하여 브라우저에서 직접 인스턴스에 연결하는 옵션을 제공합니다. 브라우저에서 NPAPI를 지원하지 않으면 연결할 때 NPAPI deprecation on Chrome(Chrome에서 NPAPI 운영 중단)이라는 오류 메시지가 표시됩니다. 메시지에서는 다른 브라우저를 사용할 것을 권장합니다. 하지만 이러한 브라우저의 최근 버전에서도 NPAPI를 지원하지 않으므로 이를 사용하여 인스턴스에 연결할 수 없으며 다른 방법으로 인스턴스에 연결해야 합니다.

자세한 내용은 다음을 참조하십시오.

- 일반: NPAPI Wikipedia 기사
- Chrome: NPAPI 운영 중단 기사
- Firefox: NPAPI 운영 중단 기사
- Safari: NPAPI 운영 중단 기사

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.

오류: 서버에서 예기치 않게 네트워크 연결을 차단함

Putty를 사용하여 인스턴스에 연결할 때 "Server unexpectedly closed network connection.(서버에서 예기치 않게 네트워크 연결을 차단했습니다.)"라는 오류가 발생하는 경우 연결이 끊어지지 않도록 Putty 구성의 연결 페이지에서 keepalives를 활성화했는지 확인합니다. 일부 서버는 지정된 기간 내 데이터를 수신하지 않을 때 클라이언트의 연결을 끊습니다. keepalives 간격을 59초로 설정합니다.

keepalives를 활성화한 후에도 문제가 계속 발생하는 경우 Putty 구성의 연결 페이지에서 Nagle 알고리즘을 비활성화합니다.

인스턴스 중지 문제 해결

Amazon EBS 인스턴스를 중지한 후 이 인스턴스가 `stopping` 상태로 멈춰 있는 것 같이 보일 경우 기본 호스트 컴퓨터에 문제가 있을 수 있습니다.

인스턴스가 `running` 상태에 있지 않은 동안에는 인스턴스 사용 요금이 부과되지 않습니다.

인스턴스의 콘솔 또는 AWS CLI 사용을 강제로 중단합니다.

- 인스턴스의 콘솔 사용을 강제로 중단하려면 중지된 인스턴스를 선택하고 작업, 인스턴스 상태, 중지, 예, 강제 중지를 선택합니다.
- 인스턴스의 AWS CLI 사용을 강제로 중단하려면 stop-instances 명령과 다음과 같은 `--force` 옵션을 사용합니다.

```
aws ec2 stop-instances --instance-ids i-0123ab456c789d01e --force
```

10분 후에도 인스턴스가 중지되지 않는 경우 Amazon EC2 forum에 도움을 요청하는 글을 게시합니다. 해결 방법을 신속히 찾아내려면 인스턴스 ID를 포함하고 자신이 이미 수행했던 단계에 대해 설명하십시오. 지원 플랜이 있는 경우에는 지원 센터에서 기술 지원 사례를 요청할 수 있습니다.

대체 인스턴스 생성

Amazon EC2 forum 또는 지원 센터의 지원을 기다리는 동안 문제 해결을 시도하려면 대체 인스턴스를 생성합니다. 중지된 인스턴스의 AMI를 생성하고 새로운 AMI를 사용하여 새 인스턴스를 시작합니다.

콘솔을 사용하여 대체 인스턴스를 생성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택하고 멈춘 인스턴스를 선택합니다.
3. 작업, 이미지, 이미지 생성을 차례로 선택합니다.
4. 이미지 생성 대화 상자에서 다음 필드를 채운 다음 이미지 생성을 선택합니다.

   a. AMI에 대한 이름 및 설명을 지정합니다.
   b. 재부팅 안 함을 선택합니다.

   자세한 내용은 인스턴스에서 Linux AMI 생성 (p. 113) 단원을 참조하십시오.
5. AMI에서 새로운 인스턴스를 시작하고 새로운 인스턴스가 작동하는지 확인합니다.
6. 멈춰 있는 인스턴스를 선택하고 작업, 인스턴스 상태, 종료를 차례로 선택합니다. 또한 인스턴스가 종료 중 상태로 멈추는 경우 Amazon EC2에서 몇 시간 내에 해당 인스턴스를 자동으로 종료합니다.

CLI를 사용하여 대체 인스턴스를 생성하려면

1. create-image(AWS CLI) 명령 및 다음 `--no-reboot` 옵션을 사용하여 멈춰 있는 인스턴스에서 AMI를 생성합니다.

```
aws ec2 create-image --instance-id i-0123ab456c789d01e --name "AMI" --description "AMI
 for replacement instance" --no-reboot
```

2. run-instances(AWS CLI) 명령을 다음과 같이 사용하여 AMI에서 새 인스턴스를 시작합니다.

```
aws ec2 run-instances --image-id ami-1a2b3c4d --count 1 --instance-type c3.large --key-
name MyKeyPair --security-groups MySecurityGroup
```

3. 새로운 인스턴스가 작동 중인지 확인합니다.
4. terminate-instances(AWS CLI) 명령을 다음과 같이 멈춰 있는 인스턴스를 종료합니다.

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0
```

이전 절차에 설명된 대로 AMI를 만들 수 없으면 다음과 같이 대체 인스턴스를 설정할 수 있습니다.

(대안) 콘솔을 사용하여 대체 인스턴스를 생성하려면

1. 인스턴스를 선택하고 설명, 블록 디바이스를 선택합니다. 각 볼륨을 선택하고 볼륨 ID를 적습니다. 어느 볼륨이 루트 볼륨인지 적어두어야 합니다.
2. 탐색 창에서 볼륨을 선택합니다. 인스턴스에 해당하는 각 볼륨을 선택하고 작업, 스냅샷 생성을 차례로 선택합니다.
3. 탐색 창에서 스냅샷을 선택합니다. 방금 만든 스냅샷을 선택한 후 작업, 볼륨 생성을 선택합니다.
4. 멈춰 있는 인스턴스와 동일한 운영 체제에서 인스턴스를 시작합니다. 루트 볼륨의 볼륨 ID와 디바이스 이름을 적어둡니다.
5. 탐색 창에서 인스턴스를 선택하고 방금 시작한 인스턴스를 선택한 후, 작업, 인스턴스 상태를 차례로 선택하고 중지를 선택합니다.
6. 탐색 창에서 볼륨을 선택하고 중지된 인스턴스의 루트 볼륨을 선택한 후, 작업, 볼륨 분리를 선택합니다.
7. 멈춰 있는 인스턴스에서 만든 루트 볼륨을 선택하고 작업, 볼륨 연결을 선택한 후, 이 볼륨을 새 인스턴스에 루트 볼륨으로 연결합니다(기록해 놓은 디바이스 이름 사용). 루트 이외의 다른 추가 볼륨을 인스턴스에 연결합니다.
8. 탐색 창에서 인스턴스를 선택하고 대체 인스턴스를 선택합니다. 작업, 인스턴스 상태, 시작을 차례로 선택합니다. 인스턴스가 작동 중인지 확인합니다.
9. 멈춰 있는 인스턴스를 선택하고 작업, 인스턴스 상태, 종료를 차례로 선택합니다. 또한 인스턴스가 종료 중 상태로 멈추는 경우 Amazon EC2에서 몇 시간 내에 해당 인스턴스를 자동으로 종료합니다.

인스턴스 종료 문제 해결

인스턴스가 `running` 상태에 있지 않은 동안에는 인스턴스 사용 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 forum에 도움을 요청하는 글을 게시하십시오. 해결 방법을 신속히 찾아내려면 인스턴스 ID를 포함하고 자신이 이미 수행했던 단계에 대해 설명하십시오. 지원 플랜이 있는 경우에는 지원 센터에서 기술 지원 사례를 요청할 수 있습니다.

종료된 인스턴스가 계속 표시됨

인스턴스를 종료한 후에도 인스턴스는 잠깐 동안 콘솔에서 표시된 후 삭제됩니다. 상태가 `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 시작하는 것을 볼 수 있습니다. 일반적으로 이러한 동작은 사용자가 Amazon EC2 Auto Scaling 또는 Elastic Beanstalk를 사용하여 정의한 기준에 따라 컴퓨팅 리소스의 규모를 자동으로 확장/축소해 왔음을 의미합니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서 또는 AWS Elastic Beanstalk 개발자 안내서 단원을 참조하십시오.

상태 확인에 실패한 인스턴스 문제 해결

다음 정보는 인스턴스가 상태 점검에 실패할 경우 문제의 문제 해결에 도움이 될 수 있습니다. 먼저 애플리케이션에 문제가 있는지 여부를 결정합니다. 인스턴스에서 애플리케이션이 예상대로 실행되고 있지 않은지 확인하면 상태 점검 정보와 시스템 로그를 검토합니다.

목차

- 상태 점검 정보 검토 (p. 1095)
- 시스템 로그 검색 (p. 1095)
- Linux 기반 인스턴스의 시스템 로그 오류 문제 해결 (p. 1096)
- 메모리 부족: 프로세스 중지 (p. 1097)
- ERROR: mmu_update failed(메모리 관리 업데이트 실패) (p. 1097)
- I/O 오류(블록 디바이스 장애) (p. 1098)
- I/O ERROR: neither local nor remote disk(분산된 블록 디바이스 손상) (p. 1099)
- request_module: runaway loop modprobe(이전 Linux 버전에서 레거시 커널 modprobe 반복) (p. 1100)
- "FATAL: kernel too old" 및 "fsck: No such file or directory while trying to open /dev"(커널과 AMI 불일치) (p. 1100)
- "FATAL: Could not load /lib/modules" 또는 "BusyBox"(커널 모듈 누락) (p. 1101)
- ERROR Invalid kernel(EC2 커널이 호환되지 않음) (p. 1102)
- fsck: No such file or directory while trying to open... (파일 시스템을 찾을 수 없음) (p. 1103)
- 파일 시스템 마운트 관련 일반 오류(마운트 실패) (p. 1105)
- VFS: Unable to mount root fs on unknown-block(루트 파일 시스템 불일치) (p. 1106)
- Error: Unable to determine major/minor number of root device... (루트 파일 시스템/디바이스 불일치) (p. 1107)
- XENBUS: Device with no driver... (p. 1108)

- ... days without being checked, check forced(파일 시스템 검사 필요) (p. 1109)
- fsck died with exit status... (디바이스 누락) (p. 1109)
- GRUB 프롬프트(grubdom>) (p. 1110)
- 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. 1112)
- 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. 1113)
- XENBUS: Timeout connecting to devices(Xenbus 시간 초과) (p. 1114)

상태 점검 정보 검토

Amazon EC2 콘솔을 사용하여 손상된 인스턴스를 찾아내려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택한 다음 인스턴스를 선택합니다.
3. 세부 정보 창에서 상태 검사 탭을 클릭하여 모든 시스템 상태 검사 및 인스턴스 상태 검사에 대한 개별 결과를 확인합니다.

시스템 상태 확인이 실패한 경우 다음 옵션 중 하나를 시도할 수 있습니다.

- 인스턴스 복구 경보를 만듭니다. 자세한 내용은 인스턴스를 중지, 종료, 재부팅 또는 복구하는 경보 만들기 (p. 637)을 참조하십시오.
- 인스턴스 유형을 Nitro 기반 인스턴스 (p. 181)로 변경하면 필수 ENA 및 NVMe 드라이버가 없는 인스턴스에서 마이그레이션한 경우 상태 점검이 실패합니다. 자세한 내용은 인스턴스 크기 조정을 위한 호환성 (p. 262) 단원을 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 forum에 게시합니다.
- 인스턴스가 Auto Scaling 그룹에 있는 경우, Amazon EC2 Auto Scaling 서비스가 자동으로 교체 인스턴스를 시작합니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Auto Scaling 인스턴스에 대한 상태 점검 단원을 참조하십시오.
- 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 인스턴스를 선택하고 인스턴스를 선택합니다.
3. 작업, 인스턴스 상태], 재부팅을 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 작업, 인스턴스 설정, 시스템 로그 가져오기를 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, 상태 검사 탭의 피드백 제출을 선택하면 검색 테스트를 개선하는 데 도움이 될 수 있습니다.
8. 문제가 해결되지 않으면 해당 문제를 Amazon EC2 forum에 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. 1097)
- ERROR: mmu_update failed(메모리 관리 업데이트 실패) (p. 1097)

디바이스 오류

- I/O 오류(블록 디바이스 장애) (p. 1098)
- I/O ERROR: neither local nor remote disk(분산된 블록 디바이스 손상) (p. 1099)

커널 오류

- request_module: runaway loop modprobe(이전 Linux 버전에서 레거시 커널 modprobe 반복) (p. 1100)
- "FATAL: kernel too old" 및 "fsck: No such file or directory while trying to open /dev"(커널과 AMI 불일치) (p. 1100)
- "FATAL: Could not load /lib/modules" 또는 "BusyBox"(커널 모듈 누락) (p. 1101)
- ERROR Invalid kernel(EC2 커널이 호환되지 않음) (p. 1102)

파일 시스템 오류

- fsck: No such file or directory while trying to open... (파일 시스템을 찾을 수 없음) (p. 1103)
- 파일 시스템 마운트 관련 일반 오류(마운트 실패) (p. 1105)
- VFS: Unable to mount root fs on unknown-block(루트 파일 시스템 불일치) (p. 1106)
- Error: Unable to determine major/minor number of root device... (루트 파일 시스템/디바이스 불일치) (p. 1107)
- XENBUS: Device with no driver... (p. 1108)
- ... days without being checked, check forced(파일 시스템 검사 필요) (p. 1109)
- fsck died with exit status... (디바이스 누락) (p. 1109)

운영 체제 오류

- GRUB 프롬프트(grubdom>) (p. 1110)
- 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. 1112)
- 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. 1113)
- XENBUS: Timeout connecting to devices(Xenbus 시간 초과) (p. 1114)

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 사용됨

권장 조치

이 인스턴스 유형의 경우	조치
Amazon EBS 지원	다음 중 하나를 수행하십시오. • 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	다음 중 하나를 수행하십시오. • 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.i686.img

ERROR: mmu_update failed with rc=-22
```

예상 원인

Amazon Linux 관련 문제

권장 조치

문제를 개발자 포럼에 게시하거나, AWS Support에 문의하십시오.

I/O 오류(블록 디바이스 장애)

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	조치
Amazon EBS 지원	다음 절차를 수행하십시오. 1. 인스턴스를 중지합니다. 2. 볼륨을 분리합니다. 3. 볼륨 복구를 시도합니다.

이 인스턴스 유형의 경우	조치
	Note Amazon EBS 볼륨의 스냅샷을 정기적으로 생성하는 것이 좋습니다. 그러면 오류로 인한 데이터 손실의 위험을 크게 줄일 수 있습니다. 4. 볼륨을 인스턴스에 다시 연결합니다. 5. 볼륨을 분리합니다.
인스턴스 스토어 지원	인스턴스를 종료하고 새 인스턴스를 시작합니다. Note 데이터를 복구할 수 없습니다. 백업에서 복구합니다. Note 백업용으로 Amazon S3 또는 Amazon EBS를 사용하는 것이 좋습니다. 인스턴스 스토어 볼륨이 하나의 호스트 및 하나의 디스크 오류와 연결됩니다.

I/O 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)

0MB HIGHMEM available.
...

request_module: runaway loop modprobe binfmt-464c

request_module: runaway loop modprobe binfmt-464c

request_module: runaway loop modprobe binfmt-464c

request_module: runaway loop modprobe binfmt-464c

request_module: runaway loop modprobe binfmt-464c
```

권장 조치

이 인스턴스 유형의 경우	조치
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: 1. 인스턴스를 중지합니다. 2. 최신 커널이 사용되도록 kernel 및 ramdisk 특성을 수정합니다. 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.

```
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	조치
Amazon EBS 지원	다음 절차를 수행하십시오. 1. 인스턴스를 중지합니다. 2. 최신 커널이 사용되도록 구성을 수정합니다. 3. 인스턴스를 시작합니다.
인스턴스 스토어 지원	다음 절차를 수행하십시오. 1. 최신 커널을 사용하는 AMI를 만듭니다. 2. 인스턴스를 종료합니다. 3. 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... ...
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
Done.
Begin: Running /scripts/init-premount ...
Done.
Begin: Mounting root file system... ...
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/sda1`로 올바르게 연결되지 않았습니다.

권장 조치

이 인스턴스 유형의 경우	조치
Amazon EBS 지원	다음 절차를 수행하십시오. 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	다음 절차를 수행하십시오. 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	조치
Amazon EBS 지원	다음 절차를 수행하십시오. 1. 인스턴스를 중지합니다. 2. 작동 중인 커널로 대체합니다. 3. 대체 커널을 설치합니다. 4. 올바른 커널로 변경하여 AMI를 수정합니다.
인스턴스 스토어 지원	다음 절차를 수행하십시오. 1. 인스턴스를 종료하고 올바른 커널을 사용하여 새 인스턴스를 시작합니다. 2. 올바른 커널을 사용하여 AMI를 만듭니다. 3. (선택 사항) AWS Support에 데이터 복구 기술 지원을 요청합니다.

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	조치
Amazon EBS 지원	다음 절차를 수행하십시오. 1. 인스턴스를 중지하고 루트 볼륨을 분리한 다음, 볼륨의 /etc/fstab를 복구/수정하고 볼륨을 인스턴스에 연결한 다음, 인스턴스를 시작합니다. 2. 수정된 /etc/fstab가 포함되도록 ramdisk를 수정합니다(해당되는 경우). 3. 최신 ramdisk를 사용하도록 AMI를 수정합니다. 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.
인스턴스 스토어 지원	다음 절차를 수행하십시오. 1. 인스턴스를 종료하고 새 인스턴스를 시작합니다.

이 인스턴스 유형의 경우	조치
	2. 잘못된 Amazon EBS 볼륨을 모두 분리하고 인스턴스를 재부팅합니다. 3. (선택 사항) 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	• Amazon EBS 볼륨 분리 또는 실패. • 파일 시스템 손상. • ramdisk와 AMI 조합의 불일치(예: Debian ramdisk와 SUSE AMI).
인스턴스 스토어 지원	• 드라이브 실패. • 파일 시스템 손상.

인스턴스 유형	예상 원인
	• ramdisk와 조합의 불일치(예: Debian ramdisk와 SUSE AMI).

권장 조치

이 인스턴스 유형의 경우	조치
Amazon EBS 지원	다음 절차를 수행하십시오. 1. 인스턴스를 중지합니다. 2. 루트 볼륨을 분리합니다. 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	다음 중 하나를 시도하십시오. • 새 인스턴스를 시작합니다. • (선택 사항) 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	• 디바이스가 올바르게 연결되지 않았습니다. • 루트 디바이스가 올바른 디바이스 지점에서 연결되지 않았습니다.

인스턴스 유형	예상 원인
	• 필요한 형식의 파일 시스템이 아닙니다. • 레거시 커널(예: 2.6.16-XenU)이 사용되었습니다. • 인스턴스의 최신 커널 업데이트(잘못된 업데이트 또는 업데이트 버그)
인스턴스 스토어 지원	하드웨어 디바이스 실패.

권장 조치

이 인스턴스 유형의 경우	조치
Amazon EBS 지원	다음 중 하나를 수행하십시오. • 인스턴스를 중지했다가 다시 시작합니다. • 올바른 디바이스 지점(예: /dev/sda 대신에 /dev/sda1)에서 연결되도록 루트 볼륨을 수정합니다. • 중지하고 현대식 커널을 사용하도록 수정합니다. • Linux 배포 설명서를 참조하여 알려진 업데이트 버그가 있는지 확인합니다. 커널을 변경하거나 다시 설치합니다.
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 인스턴스 커널을 선택했습니다.

권장 조치

이 인스턴스 유형의 경우	조치
Amazon EBS 지원	다음 절차를 수행하십시오. 1. 인스턴스를 중지합니다. 2. 볼륨을 분리합니다. 3. 디바이스 매핑 문제를 해결합니다. 4. 인스턴스를 시작합니다. 5. 디바이스 매핑 문제를 해결하도록 AMI를 수정합니다.
인스턴스 스토어 지원	다음 절차를 수행하십시오. 1. 적절히 수정(블록 디바이스를 올바르게 매핑)하여 새 AMI를 만듭니다. 2. 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)
Initalizing 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 인스턴스 커널을 선택했습니다.

권장 조치

이 인스턴스 유형의 경우	조치
Amazon EBS 지원	다음 절차를 수행하십시오. 1. 인스턴스를 중지합니다. 2. 볼륨을 분리합니다. 3. 디바이스 매핑 문제를 해결합니다. 4. 인스턴스를 시작합니다. 5. 디바이스 매핑 문제를 해결하도록 AMI를 수정합니다.
인스턴스 스토어 지원	다음 절차를 수행하십시오. 1. 적절히 수정(블록 디바이스를 올바르게 매핑)하여 새 AMI를 만듭니다. 2. 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	조치
Amazon EBS 지원	다음 중 하나 이상을 시도하여 문제를 해결하십시오. • 인스턴스를 중지하고 볼륨을 기존의 실행 중인 인스턴스에 연결합니다. • 일관성 검사를 수동으로 실행합니다. • 관련 유틸리티를 포함하도록 ramdisk를 수정합니다. • 일관성 요구 사항을 제거하도록 파일 시스템 튜닝 매개 변수를 수정합니다(권장되지 않음).
인스턴스 스토어 지원	다음 중 하나 이상을 시도하여 문제를 해결하십시오. • 올바른 도구로 ramdisk 번들을 다시 구성합니다. • 일관성 요구 사항을 제거하도록 파일 시스템 튜닝 매개 변수를 수정합니다(권장되지 않음). • 인스턴스를 종료하고 새 인스턴스를 시작합니다. • (선택 사항) 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	• 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	• 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	조치
Amazon EBS 지원	옵션 1: AMI를 수정하고 인스턴스를 다시 시작합니다. 1. 표준 위치(/boot/grub/menu.lst)에서 GRUB 구성 파일을 만들도록 원본 AMI를 수정합니다. 2. GRUB 버전에서 기본 파일 시스템 유형을 지원하는지 확인하고 필요할 경우 GRUB을 업그레이드합니다. 3. 적합한 GRUB 이미지(hd0-첫 번째 드라이브 또는 hd00 – 첫 번째 드라이브, 첫 번째 파티션)를 선택합니다. 4. 인스턴스를 종료하고 만든 AMI를 사용하여 새 인스턴스를 시작합니다. 옵션 2: 기존 인스턴스 수정: 1. 인스턴스를 중지합니다. 2. 루트 파일 시스템을 분리합니다. 3. 루트 파일 시스템을 작동하는 것으로 알려진 인스턴스에 연결합니다. 4. 파일 시스템을 마운트합니다. 5. GRUB 구성 파일을 만듭니다. 6. GRUB 버전에서 기본 파일 시스템 유형을 지원하는지 확인하고 필요할 경우 GRUB을 업그레이드합니다. 7. 파일 시스템을 분리합니다.

이 인스턴스 유형의 경우	조치
	8. 원래 인스턴스에 연결합니다. 9. 적합한 GRUB 이미지(첫 번째 디스크 또는 첫 번째 디스크의 첫 번째 파티션)를 사용하도록 커널 속성을 수정합니다. 10.인스턴스를 시작합니다.
인스턴스 스토어 지원	옵션 1: AMI를 수정하고 인스턴스를 다시 시작합니다. 1. 표준 위치(/boot/grub/menu.lst)에서 GRUB 구성 파일을 사용하여 새 AMI를 만듭니다. 2. 적합한 GRUB 이미지(hd0-첫 번째 드라이브 또는 hd00 – 첫 번째 드라이브, 첫 번째 파티션)를 선택합니다. 3. GRUB 버전에서 기본 파일 시스템 유형을 지원하는지 확인하고 필요할 경우 GRUB을 업그레이드합니다. 4. 인스턴스를 종료하고 만든 AMI를 사용하여 새 인스턴스를 시작합니다. 옵션 2: 인스턴스를 종료하고 올바른 커널을 지정하여 새 인스턴스를 시작합니다. Note 기존 인스턴스에서 데이터를 복구하려면 AWS Support에 문의하십시오.

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)

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	조치
Amazon EBS 지원	다음 중 하나를 수행하십시오. • 하드 코딩이 제거되도록 AMI를 수정하고 인스턴스를 다시 시작합니다. • 하드 코딩된 MAC 주소가 제거되도록 인스턴스를 수정합니다. 또는 다음 절차를 수행하십시오. 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. • 하드 코딩된 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	조치
Amazon EBS 지원	다음 절차를 수행하십시오. 1. 실패한 인스턴스를 중지합니다. 2. 실패한 인스턴스의 루트 볼륨을 분리합니다. 3. 루트 볼륨을 실행 중인 다른 Linux 인스턴스(나중에 복구 인스턴스로 불립니다)에 연결합니다. 4. 복구 인스턴스에 연결하여 실패한 인스턴스의 루트 볼륨을 마운트합니다. 5. 마운트된 루트 볼륨에서 SELinux를 비활성화합니다. 이 과정은 Linux 배포판에 따라 차이가 있습니다. 자세한 내용은 운영 체제별 설명서를 참조하십시오. Note 일부 시스템에서는 `SELINUX=disabled` 파일에서 `/mount_point/etc/sysconfig/selinux`로 설정함으로써 SELinux를 비활성화합니다. 이 파일에서 *`mount_point`*는 복구 인스턴스에서 볼륨을 마운트한 위치입니다. 6. 복구 인스턴스에서 루트 볼륨을 마운트 해제하고 분리한 다음 원본 인스턴스에 다시 연결합니다. 7. 인스턴스를 시작합니다.
인스턴스 스토어 지원	다음 절차를 수행하십시오. 1. 인스턴스를 종료하고 새 인스턴스를 시작합니다. 2. (선택 사항) 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 연결되어 있지 않습니다.
- 이 인스턴스에 오래된 인스턴스 커널이 사용되고 있습니다.

권장 조치

이 인스턴스 유형의 경우	조치
Amazon EBS 지원	다음 중 하나를 수행하십시오. • 현대식 커널을 사용하도록 AMI 및 인스턴스를 수정하고 인스턴스를 다시 시작합니다. • 인스턴스를 재부팅합니다.
인스턴스 스토어 지원	다음 중 하나를 수행하십시오. • 인스턴스를 종료합니다. • 현대식 커널을 사용하도록 AMI를 수정하고 이 AMI를 사용하여 새 인스턴스를 시작합니다.

연결할 수 없는 인스턴스 문제 해결

다음 방법을 사용하여 연결할 수 없는 인스턴스의 문제를 해결할 수 있습니다.

목차

- 인스턴스 재부팅 (p. 1115)
- 인스턴스 콘솔 출력 (p. 1115)
- 연결할 수 없는 인스턴스의 스크린샷 캡처 (p. 1116)
- 호스트 컴퓨터 실패 시 인스턴스 복구 (p. 1117)

인스턴스 재부팅

연결할 수 없게 된 인스턴스를 재부팅하는 기능도 문제 해결과 일반 인스턴스 관리용으로 유용합니다.

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. 520) 단원을 참조하십시오.

> **Warning**
>
> Windows 인스턴스의 경우, 이 작업으로 하드 재부팅이 수행되며, 이로 인해 데이터가 손상될 수 있습니다.

인스턴스 콘솔 출력

콘솔 출력은 문제 진단을 위한 유용한 도구입니다. 인스턴스가 종료되거나 SSH 데몬을 시작하기 전에 연결할 수 없게 되는 서비스 구성 문제 또는 커널 문제를 해결하는 데 특히 유용합니다.

Linux/Unix 인스턴스의 경우, 컴퓨터에 연결된 물리적 컴퓨터에 일반적으로 표시되는 정확한 콘솔 출력이 인스턴스 콘솔 출력에 표시됩니다. 콘솔 출력은 인스턴스 전환 상태(시작, 중지, 재부팅 및 종료) 직후에 게시된 버퍼링된 정보를 반환합니다. 게시된 출력은 지속적으로 업데이트되지 않습니다. 최대값일 것 같을 때만 업데이트됩니다.

Windows 인스턴스의 경우 인스턴스 콘솔 출력에 마지막 세 개의 시스템 이벤트 로그 오류가 포함됩니다.

옵션으로 인스턴스 수명 주기 동안 언제든지 최신 직렬 콘솔 출력을 검색할 수 있습니다. 이 옵션은 Nitro 기반 인스턴스 (p. 181)에서만 지원됩니다. Amazon EC2 콘솔에서는 지원하지 않습니다.

> **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 인스턴스를 선택하고 인스턴스를 선택합니다.
3. 작업, 인스턴스 설정, 시스템 로그 가져오기를 차례로 선택합니다.

명령줄을 사용하여 콘솔 출력 가져오기

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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. 1096)를 참조하십시오.

연결할 수 없는 인스턴스의 스크린샷 캡처

SSH 또는 RDP를 통해 인스턴스에 연결할 수 없는 경우 인스턴스의 스크린샷을 캡처하여 이미지로 볼 수 있습니다. 이미지에는 인스턴스의 상태에 관한 가시성이 제공되므로 더 빠르게 문제를 해결할 수 있습니다. 인스턴스가 실행 중이거나 인스턴스가 중단된 후에 스크린샷을 생성할 수 있습니다. 이 스크린샷을 위한 데이터 전송 비용은 따로 들지 않습니다. 이미지는 JPG 형식으로 생성되며, 100KB보다 크지 않습니다. 인스턴스가 NVIDIA GRID 드라이버를 사용할 때나 베어 메탈 인스턴스(`*.metal` 유형의 인스턴스)에서는 이 기능이 지원되지 않습니다. 이 기능은 다음 리전에서 사용 가능합니다.

- 미국 동부(버지니아 북부) 지역
- 미국 동부(오하이오) 리전
- 미국 서부(오레곤) 지역
- 미국 서부(캘리포니아 북부) 리전
- 유럽(아일랜드) 리전
- 유럽(프랑크푸르트) 리전
- 아시아 태평양(도쿄) 리전
- 아시아 태평양(서울) 리전
- 아시아 태평양(싱가포르) 리전
- 아시아 태평양(시드니) 리전)
- 남아메리카(상파울루) 리전)
- 아시아 태평양(뭄바이) 리전
- 캐나다(중부) 리전)
- 유럽(런던) 리전
- 유럽(파리) 리전

인스턴스 콘솔에 액세스하려면

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

2. 왼쪽 탐색 창에서 인스턴스를 선택합니다.
3. 캡처할 인스턴스를 선택합니다.
4. 작업, 인스턴스 설정을 차례로 선택합니다.
5. 인스턴스 스크린샷 가져오기를 선택합니다.

다운로드할 이미지를 마우스 오른쪽 버튼으로 클릭하고 저장합니다.

명령줄을 이용하여 스크린을 캡처하려면

다음 명령 중 하나를 사용할 수 있습니다. 반환되는 내용은 base64-encoded입니다. 명령줄 인터페이스에 대한 자세한 내용은 Amazon EC2에 액세스 (p. 3) 단원을 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 중지 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 중지합니다.
3. 인스턴스를 시작합니다.
4. 중요 데이터를 복원합니다.

자세한 내용은 인스턴스 중지 및 시작 (p. 507) 단원을 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.

자세한 내용은 인스턴스 스토어 기반 Linux AMI 생성 (p. 115) 단원을 참조하십시오.

잘못된 볼륨에서 부팅

`/dev/xvda` 또는 `/dev/sda`에 연결된 볼륨이 아닌 다른 볼륨이 인스턴스의 루트 볼륨이 되는 경우가 있을 수 있습니다. 이 상황은 다른 인스턴스의 루트 볼륨이나 루트 볼륨의 스냅샷에서 생성된 볼륨을 기존 루트 볼륨의 인스턴스에 연결한 경우에 발생할 수 있습니다.

이 문제는 Linux의 첫 번째 ramdisk의 작동 방식 때문에 야기됩니다. 보통 `/`에서 `/etc/fstab` 로 정의된 볼륨을 선택하게 되는데, 일부 배포에서는 이를 볼륨 파티션에 연결된 레이블로 확인합니다. 특히 `/etc/fstab`의 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 확인한 경우 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/xvdf1` 볼륨 대신에 `/dev/xvda1`이 ramdisk 실행 후 인스턴스가 부팅되는 루트 디바이스가 될 수 있습니다. 이 문제를 해결하려면 동일한 e2label 명령을 사용하여 부팅 볼륨이 되게 하지 않으려는 볼륨의 레이블을 변경합니다.

경우에 따라 `/etc/fstab`에서 UUID를 지정하면 이 문제를 해결할 수 있습니다. 하지만 두 볼륨 모두 동일한 스냅샷에서 생성된 경우 또는 두 번째 스냅샷이 기본 볼륨에서 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 ext4 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/
   ```

연결된 xfs 볼륨의 레이블을 변경하려면

- xfs_admin 명령을 사용하여 볼륨의 레이블을 /가 아닌 다른 레이블로 변경합니다.

  ```
  [ec2-user ~]$ sudo xfs_admin -L old/ /dev/xvdf1
  writing all SBs
  new label = "old/"
  ```

그림과 같이 볼륨 라벨을 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.

사용 Linux용 EC2Rescue

Linux용 EC2Rescue는 100개 이상의 Amazon EC2 Linux 인스턴스에서 실행이 가능하고 모듈 라이브러리를 사용하여 일반 문제를 진단하고 해결하는 사용하기 쉬운 오픈 소스 도구입니다. 에 대한 몇 가지 일반화된 사용 사례로는 syslog 및 패키지 관리자 로그 수집, 리소스 사용률 데이터 수집, 문제가 알려진 커널 파라미터 및 일반 OpenSSH 문제 진단/해결이 있습니다.

> **Note**
>
> Windows 인스턴스를 사용하는 경우 Windows Server용 EC2Rescue를 참조하십시오.

내용

- Linux용 EC2Rescue 설치 (p. 1119)
- Linux용 EC2Rescue로 작업 (p. 1122)
- EC2Rescue 모듈 개발 (p. 1124)

Linux용 EC2Rescue 설치

다음 사전 조건을 충족하는 Amazon EC2 Linux 인스턴스에 Linux용 EC2Rescue 도구를 설치할 수 있습니다.

사전 조건

- 지원되는 운영 체제:
  - Amazon Linux 2
  - Amazon Linux 2016.09+
  - SLES 12+
  - RHEL 7+
  - Ubuntu 16.04+
- 소프트웨어 요구 사항:
  - Python 2.7.9+ 또는 3.2+

시스템에 필수 Python 버전이 있다면 표준 빌드를 설치할 수 있습니다. 없다면 Python의 최소 사본이 포함된 번들링된 빌드를 설치할 수 있습니다.

표준 빌드를 설치하려면

1. 작동하는 Linux 인스턴스에서 Linux용 EC2Rescue 도구를 다운로드합니다.

```
curl -O https://s3.amazonaws.com/ec2rescuelinux/ec2rl.tgz
```

2. (선택 사항) 계속 진행하기 전에 Linux용 EC2Rescue 설치 파일의 서명을 확인할 수 있습니다. 자세한 내용은 (선택 사항) Linux용 EC2Rescue의 서명 확인 (p. 1120) 단원을 참조하십시오.
3. sha256 해시 파일 다운로드:

```
curl -O https://s3.amazonaws.com/ec2rescuelinux/ec2rl.tgz.sha256
```

4. tarball의 무결성 확인:

```
sha256sum -c ec2rl.tgz.sha256
```

5. tarball의 압축을 풉니다.

```
tar -xvf ec2rl.tgz
```

6. 도움말 파일을 나열하여 설치를 확인합니다.

```
cd ec2rl-<version_number>
./ec2rl help
```

번들링된 빌드를 설치하려면

다운로드 링크 및 제한 사항 목록은 GitHub에서 Linux용 EC2Rescue를 참조하십시오.

(선택 사항) Linux용 EC2Rescue의 서명 확인

다음은 Linux 기반 운영 체제용 Linux용 EC2Rescue 패키지의 유효성을 확인하는 권장 절차입니다.

인터넷에서 애플리케이션을 다운로드할 때 소프트웨어 게시자의 자격 증명을 인증하고 애플리케이션이 게시된 후 변경되거나 손상되지 않았는지 확인하는 것이 좋습니다. 이를 통해 바이러스나 기타 악성 코드가 포함된 애플리케이션 버전을 설치하는 것을 방지할 수 있습니다.

이 단원의 절차를 수행한 후에 Linux용 EC2Rescue용 소프트웨어가 변경되거나 손상된 것을 발견한 경우 설치 파일을 실행하지 마십시오. 대신 Amazon Web Services에 문의하십시오.

Linux 기반 운영 체제용 Linux용 EC2Rescue 파일은 보안 디지털 서명을 위한 Pretty Good Privacy 표준의 오픈 소스 구현(OpenPGP)인 GnuPG를 사용하여 서명됩니다. GnuPG(GPG라고도 함)는 디지털 서명을 통해 확인한 후 인증 및 무결성을 제공합니다. AWS는 다운로드된 Linux용 EC2Rescue 패키지를 확인할 수 있도록 퍼블릭 키 및 서명을 게시합니다. PGP 및 GnuPG(GPG)에 대한 자세한 내용은 http://www.gnupg.org를 참조하십시오.

첫 번째 단계는 소프트웨어 게시자와 신뢰를 구축하는 것입니다. 소프트웨어 게시자의 퍼블릭 키를 다운로드하고, 퍼블릭 키의 소유자가 정당한 소유자인지 확인한 다음, 퍼블릭 키를 인증 키에 추가합니다. 인증 키는 알려진 퍼블릭 키의 모음입니다. 퍼블릭 키의 신뢰성을 설정한 후 이를 사용하여 애플리케이션의 서명을 확인할 수 있습니다.

작업

- GPG 도구 설치 (p. 1120)
- 퍼블릭 키 인증 및 가져오기 (p. 1121)
- 패키지의 서명 확인 (p. 1121)

GPG 도구 설치

Linux 또는 Unix 운영 체제를 사용하는 경우 일반적으로 GPG 도구가 이미 설치되어 있을 수 있습니다. 시스템에 도구가 설치되어 있는지 테스트하려면 명령 프롬프트에 gpg2를 입력합니다. GPG 도구가 설치되어 있는 경우 GPG 명령 프롬프트가 표시됩니다. GPG 도구가 설치되어 있지 않은 경우 명령을 찾을 수 없다는 오류가 표시됩니다. 리포지토리에서 GnuPG 패키지를 설치할 수 있습니다.

Debian 기반 Linux에서 GPG 도구를 설치하려면

- 터미널에서 다음 명령을 실행합니다.

```
apt-get install gnupg2
```

Red Hat 기반 Linux에서 GPG 도구를 설치하려면

- 터미널에서 다음 명령을 실행합니다.

```
yum install gnupg2
```

퍼블릭 키 인증 및 가져오기

프로세스의 다음 단계는 Linux용 EC2Rescue 퍼블릭 키를 인증하고 이를 신뢰할 수 있는 키로 GPG 인증 키에 추가하는 것입니다.

Linux용 EC2Rescue 퍼블릭 키를 인증하고 가져오려면

1. 명령 프롬프트에서 다음 명령을 사용하여 퍼블릭 GPG 빌드 키의 사본을 받습니다.

```
curl -O https://s3.amazonaws.com/ec2rescuelinux/ec2rl.key
```

2. `ec2rl.key`를 저장한 디렉터리의 명령 프롬프트에서 다음 명령을 사용하여 Linux용 EC2Rescue 퍼블릭 키를 인증 키로 가져옵니다.

```
gpg2 --import ec2rl.key
```

이 명령은 다음과 같은 결과를 반환합니다.

```
gpg: /home/ec2-user/.gnupg/trustdb.gpg: trustdb created
gpg: key 2FAE2A1C: public key "ec2autodiag@amazon.com <EC2 Rescue for Linux>" imported
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)
```

패키지의 서명 확인

GPG 도구를 설치하고, Linux용 EC2Rescue 퍼블릭 키를 인증 및 가져오고, Linux용 EC2Rescue 퍼블릭 키가 신뢰할 수 있는지 확인하면 Linux용 EC2Rescue 설치 스크립트의 서명을 확인할 준비가 된 것입니다.

Linux용 EC2Rescue 설치 스크립트 서명을 확인하려면

1. 명령 프롬프트에서 다음 명령을 실행하여 설치 스크립트용 서명 파일을 다운로드합니다.

```
curl -O https://s3.amazonaws.com/ec2rescuelinux/ec2rl.tgz.sig
```

2. `ec2rl.tgz.sig` 및 Linux용 EC2Rescue 설치 파일을 저장한 디렉터리의 명령 프롬프트에서 다음 명령을 실행하여 서명을 확인합니다. 두 파일이 모두 있어야 합니다.

```
gpg2 --verify ./ec2rl.tgz.sig
```

출력은 다음과 같아야 합니다.

```
gpg: Signature made Thu 12 Jul 2018 01:57:51 AM UTC using RSA key ID 6991ED45
gpg: Good signature from "ec2autodiag@amazon.com <EC2 Rescue for Linux>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: E528 BCC9 0DBF 5AFA 0F6C  C36A F780 4843 2FAE 2A1C
     Subkey fingerprint: 966B 0D27 85E9 AEEC 1146  7A9D 8851 1153 6991 ED45
```

출력에 `Good signature from "ec2autodiag@amazon.com <EC2 Rescue for Linux>"` 문구가 포함된 경우 서명을 확인했고 Linux용 EC2Rescue 설치 스크립트 실행을 계속할 수 있음을 의미합니다.

출력에 `BAD signature` 문구가 포함된 경우 절차를 올바르게 수행했는지 확인합니다. 계속해서 이 응답을 받게 되면 Amazon Web Services에 문의하고 이전에 다운로드한 설치 파일을 실행하지 마십시오.

다음은 표시될 수 있는 경고에 대한 세부 정보입니다.

- WARNING: This key is not certified with a trusted signature! There is no indication that the signature belongs to the owner.이는 사용자가 Linux용 EC2Rescue에 대한 신뢰할 수 있는 퍼블릭 키를 소유하고 있다는 개인적인 신뢰 수준을 가리킬 뿐입니다. Amazon Web Services 사무실을 방문하여 직접 키를 받는 것이 이상적입니다. 그러나 대부분의 경우 웹 사이트에서 다운로드합니다. 이 경우 웹 사이트는 Amazon Web Services 웹 사이트입니다.
- gpg2: no ultimately trusted keys found. 이는 사용자(또는 사용자가 신뢰하는 다른 사용자)가 특정 키를 "궁극적으로 신뢰"하지 않음을 뜻합니다.

자세한 내용은 http://www.gnupg.org를 참조하십시오.

Linux용 EC2Rescue로 작업

다음은 이 도구를 사용하여 시작할 때 수행할 수 있는 일반적인 작업입니다.

작업

- Linux용 EC2Rescue 실행 (p. 1122)
- 결과 업로드 (p. 1123)
- 백업 생성 (p. 1123)
- 도움말 보기 (p. 1123)

Linux용 EC2Rescue 실행

다음 예제에 표시된 대로 Linux용 EC2Rescue를 실행할 수 있습니다.

Example 예: 모든 모듈 실행

모든 모듈을 실행하려면 다음과 같이 옵션 없이 Linux용 EC2Rescue를 실행하십시오.

```
./ec2rl run
```

일부 모듈에는 루트 액세스가 필요합니다. 루트 사용자가 아니라면 다음과 같이 sudo를 사용하여 이 모듈을 실행하십시오.

```
sudo ./ec2rl run
```

Example 예: 특정 모듈 실행

특정 모듈만 실행하려면 --only-modules 파라미터를 사용하십시오.

```
./ec2rl run --only-modules=module_name --arguments
```

예를 들어 이 명령에서는 다음과 같이 dig 모듈을 실행하여 `amazon.com` 도메인을 쿼리합니다.

```
./ec2rl run --only-modules=dig --domain=amazon.com
```

Example 예: 결과 확인

다음과 같이 `/var/tmp/ec2rl`에서 결과를 확인할 수 있습니다.

```
cat /var/tmp/ec2rl/logfile_location
```

예를 들어 다음과 같이 dig 모듈의 로그 파일을 확인합니다.

```
cat /var/tmp/ec2rl/2017-05-11T15_39_21.893145/mod_out/run/dig.log
```

결과 업로드

AWS Support에서 S3 버킷에서 산출한 결과를 요청하거나 이 결과를 공유하도록 요청한 경우 Linux용 EC2Rescue CLI 도구를 사용하여 이 결과를 업로드합니다. Linux용 EC2Rescue 명령의 출력에는 사용해야 할 명령이 포함되어야 합니다.

Example 예: 결과를 AWS Support에 업로드

```
./ec2rl upload --upload-directory=/var/tmp/ec2rl/2017-05-11T15_39_21.893145 --support-
url="URLProvidedByAWSSupport"
```

Example 예: 결과를 S3 버킷에 업로드

```
./ec2rl upload --upload-directory=/var/tmp/ec2rl/2017-05-11T15_39_21.893145 --presigned-
url="YourPresignedS3URL"
```

Amazon S3에 대해 미리 서명된 URL 생성에 대한 자세한 내용은 미리 서명된 URL을 사용하여 객체 업로드를 참조하십시오.

백업 생성

다음 명령을 사용하여 인스턴스, 하나 이상의 볼륨 또는 특정 디바이스 ID에 대한 백업을 생성합니다.

Example 예: Amazon 머신 이미지(AMI)를 사용해 인스턴스를 백업

```
./ec2rl run --backup=ami
```

Example 예: 인스턴스와 연결된 모든 볼륨을 백업

```
./ec2rl run --backup=allvolumes
```

Example 예: 특정 볼륨 백업

```
./ec2rl run --backup=volumeID
```

도움말 보기

Linux용 EC2Rescue에는 사용 가능한 각 명령에 대한 정보와 구문을 제공하는 도움말 파일이 포함되어 있습니다.

Example 예: 일반적인 도움말 표시

```
./ec2rl help
```

Example 예: 사용 가능한 모듈 나열

```
./ec2rl list
```

Example 예: 특정 모듈에 대한 도움말 표시

```
./ec2rl help module_name
```

예를 들어 다음 명령을 사용하여 dig 모듈에 대한 도움말 파일을 표시합니다.

```
./ec2rl help dig
```

EC2Rescue 모듈 개발

모듈은 데이터 직렬화 표준인 YAML로 작성됩니다. 모듈의 YAML 파일은 모듈과 모듈의 속성을 나타내는 단일 문서로 구성됩니다.

모듈 속성 추가

다음 표에는 사용 가능한 모듈 속성이 나열되어 있습니다.

속성	설명
name	모듈의 이름입니다. 이름 길이는 18자보다 작거나 같아야 합니다.
version	모듈의 버전 번호입니다.
title	모듈에 대해 간단하게 설명하는 제목입니다. 이 값의 길이는 50자보다 작거나 같아야 합니다.

helptext	모듈에 대해 확장된 설명입니다. 각 줄의 길이는 75자보다 작거나 같아야 합니다. 모듈이 필수 또는 선택 사항으로 인수를 사용하는 경우 해당 인수를 helptext 값에 포함하십시오.
	
	다음 예를 참조하십시오.
	
	`helptext: !!str |`
	`  Collect output from ps for system analysis`
	`  Consumes --times= for number of times to repeat`
	`  Consumes --period= for time period between repetition`

placement	모듈을 실행해야 하는 단계입니다. 지원되는 값: • prediagnostic • run • postdiagnostic

속성	설명
language	모듈 코드가 작성된 언어입니다. 지원되는 값: • bash • python Note Python 코드는 Python 2.7.9+ 및 Python 3.2+와 모두 호환되어야 합니다.
remediation	모듈에서 문제 해결을 지원하는지 여부를 나타냅니다. 지원되는 값은 `True` 또는 `False`입니다. 없는 경우에는 모듈의 기본값은 `False`이며, 문제 해결을 지원하지 않는 모듈에 대해 선택 가능한 속성이 됩니다.
content	전체 스크립트 코드입니다.
constraint	제약 값을 포함하는 객체의 이름입니다.
domain	모듈을 그룹화하거나 분류하는 방법에 대한 서술자입니다. 포함된 모듈의 집합은 다음 도메인을 사용합니다. • 애플리케이션입니다 • net • os • performance
class	모듈이 수행하는 작업 유형에 대한 서술자입니다. 포함된 모듈의 집합은 다음 클래스를 사용합니다. • 수집(프로그램에서 출력 수집) • 진단(일련의 기준에 따른 통과/실패) • 모음(파일 복사 및 특정 파일에 기록)
distro	이 모듈이 지원하는 Linux 배포판의 목록입니다. 포함된 모듈의 집합은 다음 배포판을 사용합니다. • alami(Amazon Linux) • rhel • ubuntu • suse
필수	CLI 옵션에서 모듈이 사용하는 필수 인수입니다.
선택 사항	모듈이 사용할 수 있는 선택적 인수입니다.
software	모듈에서 사용되는 소프트웨어 실행 파일입니다. 이 속성은 기본적으로 설치되지 않은 소프트웨어를 지정하기 위한 용도입니다. Linux용 EC2Rescue 로직은 모듈을 실행하기 전에 이러한 프로그램의 존재 여부와 실행 가능 여부를 확인합니다.

속성	설명
package	실행 파일을 위한 소스 소프트웨어 패키지입니다. 이 속성은 추가 정보를 가져오거나 다운로드하기 위한 URL을 비롯하여 소프트웨어가 포함된 패키지에 대해 확장된 세부 정보를 제공하기 위한 용도입니다.
sudo	모듈을 실행하는 데 루트 액세스가 필요한지 여부를 나타냅니다. 모듈 스크립트에서 sudo 확인을 구현할 필요가 없습니다. 이 값이 true이면 Linux용 EC2Rescue 로직은 실행 사용자에게 루트 액세스가 있는 경우에만 모듈을 실행합니다.
perfimpact	모듈이 실행되는 환경에 중요한 성능 영향을 줄 수 있는지 여부를 나타냅니다. 이 값이 true이고 `--perfimpact=true` 인수가 없으면 모듈을 건너뜁니다.
parallelexclusive	상호 배타성이 필요한 프로그램을 지정합니다. 예를 들어, "bpf"를 지정하는 모든 모듈은 직렬 방식으로 실행됩니다.

환경 변수 추가

다음 표에는 사용 가능한 환경 변수가 나열되어 있습니다.

환경 변수	설명
`EC2RL_CALLPATH`	`ec2rl.py`의 경로입니다. 이 경로는 라이브러리 디렉터리를 찾고, 공급된 Python 모듈을 활용하는 데 사용할 수 있습니다.
`EC2RL_WORKDIR`	진단 도구에 대한 기본 tmp 디렉터리입니다. 기본값: `/var/tmp/ec2rl`.
`EC2RL_RUNDIR`	모든 출력이 저장될 디렉터리입니다. 기본값: `/var/tmp/ec2rl/<date&timestamp>`.
`EC2RL_GATHEREDDIR`	수집한 모듈 데이터를 배치하기 위한 루트 디렉터리입니다. 기본값:`/var/tmp/ec2rl/<date&timestamp>/mod_out/gathered/`.
`EC2RL_NET_DRIVER`	인스턴스에서 알파벳 순서 중 첫 번째 비 가상 네트워크 인터페이스에 사용하는 드라이버입니다. 예제: • xen_netfront • ixgbevf • ena

환경 변수	설명
`EC2RL_SUDO`	Linux용 EC2Rescue가 루트로 실행되면 true이고, 그렇지 않으면 false입니다.
`EC2RL_VIRT_TYPE`	인스턴스 메타데이터에 의해 제공된 가상화 유형입니다. 예제: • default-hvm • default-paravirtual
`EC2RL_INTERFACES`	시스템에서 인터페이스의 열거 목록입니다. 이 값은 `eth0`, `eth1` 등 이름을 포함하는 문자열입니다. 이 환경 변수는 `functions.bash`를 통해 생성되며 해당 변수를 제공한 모듈에 대해서만 사용할 수 있습니다.

YAML 구문 사용

모듈 YAML 파일을 생성할 때 다음 사항을 고려해야 합니다.

- 삼중 하이픈(`---`)은 문서의 명시적 시작을 나타냅니다.
- `!ec2rlcore.module.Module` 태그는 데이터 스트림에서 객체를 생성할 때 호출할 생성자를 YAML 구문 분석기에 알려줍니다. `module.py` 파일 내부에서 생성자를 찾을 수 있습니다.
- `!!str` 태그는 데이터 유형을 확인하지 않고 대신 콘텐츠를 문자열 리터럴로 해석하도록 YAML 구문 분석기에 지시합니다.
- 파이프 문자(`|`)는 값이 리터럴 방식의 스칼라임을 YAML 구문 분석기에 알려줍니다. 이 경우 구문 분석기에는 모든 공백이 포함됩니다. 이는 들여쓰기 및 줄 바꿈 문자가 유지되므로 모듈에 중요합니다.
- YAML 표준 들여쓰기는 공백 2개이며, 다음 예에서 확인할 수 있습니다. 스크립트에 대해 표준 들여쓰기(Python의 경우 공백 4개)를 유지한 다음, 모듈 파일 내에서 전체 콘텐츠를 공백 2개로 들여쓰기해야 합니다.

예제 모듈

예제(`mod.d/ps.yaml`):

```
--- !ec2rlcore.module.Module
# Module document. Translates directly into an almost-complete Module object
name: !!str ps
path: !!str
version: !!str 1.0
title: !!str Collect output from ps for system analysis
helptext: !!str |
  Collect output from ps for system analysis
  Requires --times= for number of times to repeat
  Requires --period= for time period between repetition
placement: !!str run
package:
  - !!str
language: !!str bash
content: !!str |
  #!/bin/bash
  error_trap()
  {
```

```
      printf "%0.s=" {1..80}
      echo -e "\nERROR: "$BASH_COMMAND" exited with an error on line ${BASH_LINENO[0]}"
      exit 0
  }
  trap error_trap ERR

  # read-in shared function
  source functions.bash
  echo "I will collect ps output from this $EC2RL_DISTRO box for $times times every $period
 seconds."
  for i in $(seq 1 $times); do
      ps auxww
      sleep $period
  done
constraint:
  requires_ec2: !!str False
  domain: !!str performance
  class: !!str collect
  distro: !!str alami ubuntu rhel suse
  required: !!str period times
  optional: !!str
  software: !!str
  sudo: !!str False
  perfimpact: !!str False
  parallelexclusive: !!str
```

진단 인터럽트 보내기(고급 사용자 전용)

> **Warning**
>
> 진단 인터럽트를 고급 사용자를 대상으로 하는 기능입니다. 잘못 사용하면 인스턴스에 부정적인 영향을 줄 수 있습니다. 진단 인터럽트를 인스턴스에 전송하면 인스턴스가 고장 나거나 재부팅될 수 있으며 이는 데이터 손실로 이어질 수 있습니다.

연결할 수 없거나 응답이 없는 Linux 인스턴스에 진단 인터럽트를 보내 커널 패닉을 수동으로 트리거 할 수 있습니다.

Linux 운영 체제는 일반적으로 커널 패닉이 발생할 때 충돌하고 재부트됩니다. 운영 체제의 특정 동작은 해당 구성에 따라 다릅니다. 커널 패닉을 사용하여 인스턴스의 운영 체제 커널이 크래시 덤프 파일 생성과 같은 작업을 수행하도록 할 수도 있습니다. 그런 다음 크래시 덤프 파일의 정보를 사용하여 근본 원인 분석을 수행하고 인스턴스를 디버그할 수 있습니다.

크래시 덤프 데이터는 인스턴스 자체에서 운영 체제에 의해 로컬로 생성됩니다.

인스턴스에 진단 인터럽트를 전송하기 전에 운영 시스템에 대한 문서를 읽고 필요한 구성을 변경하시기를 권장합니다.

목차

- 지원되는 인스턴스 유형 (p. 1128)
- 사전 조건 (p. 1129)
- 진단 인터럽트 보내기 (p. 1131)

지원되는 인스턴스 유형

진단 인터럽트는 A1을 제외한 모든 Nitro 기반 인스턴스 유형에서 지원됩니다. 자세한 내용은 Nitro 기반 인스턴스 (p. 181) 단원을 참조하십시오.

사전 조건

진단 인터럽트를 사용하기 전에 인스턴스의 운영 체제를 구성해야 합니다. 이는 커널 패닉이 발생할 때 필요한 조치가 수행되도록 합니다.

커널 패닉이 발생할 때 크래시 덤프를 생성하도록 Amazon Linux 2를 구성하려면

1. 인스턴스에 연결합니다.
2. kexec 및 kdump를 설치합니다.

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

3. 보조 커널에 적절한 양의 메모리를 예약하도록 커널을 구성하십시오. 예약할 메모리 크기는 인스턴스의 사용 가능한 총 메모리에 따라 다릅니다. 원하는 텍스트 편집기를 사용하여 `/etc/default/grub` 파일을 열고, `GRUB_CMDLINE_LINUX_DEFAULT`로 시작하는 줄을 찾은 다음 `crashkernel` 매개 변수를 다음 형식으로 추가하십시오: `crashkernel=`*`memory_to_reserve`* 예를 들어 `160MB`를 예약하려면 `grub` 파일을 다음과 같이 수정합니다.

```
GRUB_CMDLINE_LINUX_DEFAULT="crashkernel=160M console=tty0 console=ttyS0,115200n8
 net.ifnames=0 biosdevname=0 nvme_core.io_timeout=4294967295 rd.emergency=poweroff
 rd.shell=0"
GRUB_TIMEOUT=0
GRUB_DISABLE_RECOVERY="true"
```

4. 변경 내용을 저장하고 `grub` 파일을 닫습니다.
5. GRUB2 구성 파일을 재구축합니다.

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

6. Intel 및 AMD 프로세서 기반의 인스턴스에서 `send-diagnostic-interrupt` 명령은 알 수 없는 NMI(Non-Maskable Interrupt)를 인스턴스에 보냅니다. 알 수 없는 NMI를 수신하면 커널이 충돌하도록 구성해야합니다. 원하는 텍스트 편집기를 사용하여 `/etc/sysctl.conf` 파일을 열고 다음을 추가합니다.

```
kernel.unknown_nmi_panic=1
```

7. 재부팅하고 인스턴스에 다시 연결하십시오.
8. 커널이 올바른 `crashkernel` 매개 변수로 부팅되었는지 확인하십시오.

```
$ grep crashkernel /proc/cmdline
```

다음 예제 출력은 구성 성공을 나타냅니다.

```
BOOT_IMAGE=/boot/vmlinuz-4.14.128-112.105.amzn2.x86_64 root=UUID=a1e1011e-
e38f-408e-878b-fed395b47ad6 ro crashkernel=160M console=tty0 console=ttyS0,115200n8
 net.ifnames=0 biosdevname=0 nvme_core.io_timeout=4294967295 rd.emergency=poweroff
 rd.shell=0
```

9. kdump 서비스가 실행 중인지 확인합니다.

```
[ec2-user ~]$ systemctl status kdump.service
```

다음 예제 출력은 kdump 서비스가 실행 중인 경우의 결과를 보여줍니다.

```
kdump.service - Crash recovery kernel arming
```

```
   Loaded: loaded (/usr/lib/systemd/system/kdump.service; enabled; vendor preset:
 enabled)
   Active: active (exited) since Fri 2019-05-24 23:29:13 UTC; 22s ago
  Process: 2503 ExecStart=/usr/bin/kdumpctl start (code=exited, status=0/SUCCESS)
 Main PID: 2503 (code=exited, status=0/SUCCESS)
```

Note

기본적으로 크래시 덤프 파일은 `/var/crash/`에 저장됩니다. 위치를 변경하려면 원하는 텍스트 편집기를 사용하여 `/etc/kdump.conf` 파일을 수정하십시오.

커널 패닉이 발생할 때 크래시 덤프를 생성하도록 Amazon Linux를 구성하려면

1. 인스턴스에 연결합니다.
2. kexec 및 kdump를 설치합니다.

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

3. 보조 커널에 적절한 양의 메모리를 예약하도록 커널을 구성하십시오. 예약할 메모리 크기는 인스턴스의 사용 가능한 총 메모리에 따라 다릅니다.

```
$ sudo grubby --args="crashkernel=memory_to_reserve" --update-kernel=ALL
```

예를 들어, 크래시 커널을 위해 `160MB`를 예약하려면 다음 명령을 사용하십시오.

```
$ sudo grubby --args="crashkernel=160M" --update-kernel=ALL
```

4. Intel 및 AMD 프로세서 기반의 인스턴스에서 `send-diagnostic-interrupt` 명령은 알 수 없는 NMI(Non-Maskable Interrupt)를 인스턴스에 보냅니다. 알 수 없는 NMI를 수신하면 커널이 충돌하도록 구성해야합니다. 원하는 텍스트 편집기를 사용하여 `/etc/sysctl.conf` 파일을 열고 다음을 추가합니다.

```
kernel.unknown_nmi_panic=1
```

5. 재부팅하고 인스턴스에 다시 연결하십시오.
6. 커널이 올바른 `crashkernel` 매개 변수로 부팅되었는지 확인하십시오.

```
$ grep crashkernel /proc/cmdline
```

다음 예제 출력은 구성 성공을 나타냅니다.

```
root=LABEL=/ console=tty1 console=ttyS0 selinux=0 nvme_core.io_timeout=4294967295
 LANG=en_US.UTF-8 KEYTABLE=us crashkernel=160M
```

7. kdump 서비스가 실행 중인지 확인합니다.

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

서비스가 실행 중이면 명령이 `Kdump is operational` 응답을 반환합니다.

Note

기본적으로 크래시 덤프 파일은 `/var/crash/`에 저장됩니다. 위치를 변경하려면 원하는 텍스트 편집기를 사용하여 `/etc/kdump.conf` 파일을 수정하십시오.

SUSE Linux Enterprise, Ubuntu 또는 Red Hat Enterprise Linux를 구성하려면

다음 웹사이트를 참조하십시오.

- SUSE Linux Enterprise
- Ubuntu
- Red Hat Enterprise Linux (RHEL)

진단 인터럽트 보내기

필요한 구성 변경을 완료하면 AWS CLI 또는 Amazon EC2 API를 사용하여 인스턴스에 진단 인터럽트를 보낼 수 있습니다.

인스턴스에 진단 인터럽트를 보내려면(AWS CLI)

send-diagnostic-interrupt 명령을 사용하고 인스턴스 ID를 지정하십시오.

```
aws ec2 send-diagnostic-interrupt --instance-id i-1234567890abcdef0
```

문서 기록

다음 표에서는 Amazon EC2 설명서에 대한 중요 추가 사항을 설명합니다. 사용자로부터 받은 의견을 수렴하기 위해 설명서가 자주 업데이트됩니다.

현재 API 버전: 2016-11-15

기능	API 버전	설명	릴리스 날짜
로컬 영역를 활성화하는 셀프 서비스 옵션	2016-11-15	AWS Management 콘솔 또는 AWS CLI를 사용하여 로컬 영역를 활성화할 수 있습니다. 자세한 내용은 로컬 영역 활성화를 참조하십시오.	2020년 3월 5일
Amazon EBS 다중 연결	2016-11-15	이제 단일 프로비저닝된 IOPS SSD(IO1) 볼륨을 동일한 가용 영역에 있는 최대 16개의 Nitro 기반 인스턴스에 연결할 수 있습니다. 자세한 내용은 Amazon EBS 다중 연결을 사용하여 여러 인스턴스에 볼륨 연결 (p. 911) 단원을 참조하십시오.	2020년 2월 14일
AMI와 연결된 플랫폼 세부 정보 및 결제 정보	2016-11-15	온디맨드 인스턴스 또는 스팟 인스턴스를 시작하거나 예약 인스턴스를 구매하기 전에 Amazon Machine Image(AMI)와 연결된 플랫폼 세부 정보 및 결제 정보를 확인할 수 있습니다. 자세한 내용은 결제 정보 가져오기 (p. 155) 단원을 참조하십시오.	2020년 2월 6일
스팟 인스턴스 중지 및 시작	2016-11-15	이제 `stop` 중단 동작에 의존하는 대신, Amazon EBS에서 지원되는 스팟 인스턴스를 마음대로 중지하고 시작할 수 있습니다. 자세한 내용은 스팟 인스턴스 중지 (p. 331) 단원을 참조하십시오.	2020년 1월 13일
리소스에 태그 지정	2016-11-15	외부 전용 인터넷 게이트웨이, 로컬 게이트웨이, 로컬 게이트웨이 라우팅 테이블, 로컬 게이트웨이 가상 인터페이스, 로컬 게이트웨이 가상 인터페이스 그룹, 로컬 게이트웨이 라우팅 테이블 VPC 연결 및 로컬 게이트웨이 라우팅 테이블 가상 인터페이스 그룹 연결에 태그를 지정할 수 있습니다. 자세한 내용은 리소스에 태그 지정 (p. 1070) 단원을 참조하십시오.	2020년 1월 10일
세션 관리자	2016-11-15	Amazon EC2 콘솔에서 인스턴스로 세션 관리자 세션을 시작할 수 있습니다. 자세한 내용은 세션 관리자를 사용하여 Linux 인스턴스에 연결 (p. 506) 단원을 참조하십시오.	2019년 12월 18일
EC2 집합에서 온디맨드 용량 예약 지원	2016-11-15	온디맨드 인스턴스를 시작할 때 온디맨드 용량 예약부터 사용하도록 EC2 집합를 구성할 수 있습니다. 자세한 내용은 온디맨드 인스턴스에 대해 용량 예약 사용 (p. 454) 단원을 참조하십시오.	2019년 12월 16일
인스턴스 유형에 대한 권장 사항	2016-11-15	AWS Compute Optimizer는 성능 향상, 비용 절감 또는 두 가지 모두를 위한 Amazon EC2 인스턴스 권장 사항을 제공합니다. 자세한 내용은 인스턴스 유형에 대한 권장 사항 가져오기 (p. 265) 단원을 참조하십시오.	2019년 12월 3일

기능	API 버전	설명	릴리스 날짜
Inf1 인스턴스	2016-11-15	Inf1 인스턴스는 저렴한 비용으로 우수한 성능을 제공하도록 설계된 기계 학습 추론 칩인 AWS Inferentia를 사용합니다.	2019년 12월 3일
AWS Outposts 지원	2016-11-15	Outpost 서브넷에서 인스턴스를 시작할 수 있습니다. 자세한 내용은 AWS Outposts 사용 설명서를 참조하십시오.	2019년 12월 3일
로컬 영역 지원	2016-11-15	로컬 영역 서브넷에서 인스턴스를 시작할 수 있습니다. 자세한 내용은 리전, 가용 영역 및 로컬 영역를 참조하십시오.	2019년 12월 3일
전용 호스트 및 호스트 리소스 그룹	2016-11-15	이제 전용 호스트를 호스트 리소스 그룹과 함께 사용할 수 있습니다. 자세한 내용은 호스트 리소스 그룹에서 인스턴스 시작 (p. 389) 단원을 참조하십시오.	2019년 12월 2일
전용 호스트 공유	2016-11-15	이제 AWS 계정 전체에 전용 호스트를 공유할 수 있습니다. 자세한 내용은 공유된 전용 호스트 작업 (p. 401) 단원을 참조하십시오.	2019년 12월 2일
계정 수준의 기본 크레딧 사양	2016-11-15	AWS 리전별로 계정 수준에서 성능 순간 확장 가능 인스턴스 패밀리마다 기본 크레딧 사양을 설정할 수 있습니다. 자세한 내용은 계정의 기본 크레딧 사양 설정 (p. 223) 단원을 참조하십시오.	2019년 11월 25일
전용 호스트	2016-11-15	한 인스턴스 패밀리의 여러 인스턴스 유형을 지원하도록 전용 호스트를 구성할 수 있습니다. 자세한 내용은 전용 호스트 작업 (p. 384) 단원을 참조하십시오.	2019년 11월 21일
Amazon EBS 빠른 스냅샷 복원	2016-11-15	EBS 스냅샷에서 빠른 스냅샷 복원을 활성화하여 스냅샷에서 생성된 EBS 볼륨이 생성 시 완전히 초기화되고 모든 프로비저닝된 성능을 즉시 제공하도록 할 수 있습니다. 자세한 내용은 Amazon EBS 빠른 스냅샷 복원 (p. 978) 단원을 참조하십시오.	2019년 11월 20일
인스턴스 메타데이터 서비스 버전 2	2016-11-15	인스턴스 메타데이터를 요청하는 세션 지향 방법인 인스턴스 메타데이터 서비스 버전 2를 사용할 수 있습니다 자세한 내용은 인스턴스 메타데이터 서비스 구성 (p. 571) 단원을 참조하십시오.	2019년 11월 19일
Elastic Fabric Adapter(EFA)	2016-11-15	이제 Elastic Fabric Adapter를 Intel MPI 2019 Update 6에 사용할 수 있습니다. 자세한 내용은 EFA 및 MPI 시작하기 (p. 733) 단원을 참조하십시오.	2019년 11월 15일
온디멘드 Windows 인스턴스에 대한 최대 절전 모드 지원	2016-11-15	온디멘드 Windows 인스턴스를 최대 절전 모드로 실행할 수 있습니다. 지원되는 Windows AMI에 대한 자세한 내용은 최대 절전 모드 필수 조건 (p. 511) 단원을 참조하십시오.	2019년 10월 14일
예약 인스턴스 대기열 구매	2016-11-15	최대 3년 전에 예약 인스턴스 구매를 대기할 수 있습니다. 자세한 내용은 구매 대기 (p. 287) 단원을 참조하십시오.	2019년 10월 4일

기능	API 버전	설명	릴리스 날짜
G4 인스턴스	2016-11-15	G4 인스턴스는 NVIDIA Tesla GPU를 사용하며 DirectX 또는 OpenGL을 사용하는 그래픽 애플리케이션과 함께 CUDA 또는 기계 학습 프레임워크를 사용하여 범용 GPU 컴퓨팅을 위한 경제적인 고성능 플랫폼을 제공합니다.	2019년 9월 19일
진단 인터럽트	2016-11-15	연결할 수 없거나 응답이 없는 인스턴스에 진단 인터럽트를 보내 커널 패닉(Linux 인스턴스) 또는 블루스크린/중지 오류(Windows 인스턴스)를 트리거할 수 있습니다. 자세한 내용은 진단 인터럽트 보내기 (고급 사용자 전용) (p. 1128) 단원을 참조하십시오.	2019년 8월 14일
용량 최적화 할당 전략	2016-11-15	이제 EC2 집합 또는 스팟 집합을 사용하여 시작 중인 인스턴스 수에 대한 용량이 최적화된 스팟 풀에서 스팟 인스턴스를 시작할 수 있습니다. 자세한 내용은 용량 최적화를 위한 EC2 집합 구성 (p. 453) 단원을 참조하십시오.	2019년 8월 12일
온디맨드 용량 예약	2016-11-15	이제 AWS 계정에 걸쳐 용량 예약을 공유할 수 있습니다. 자세한 내용은 공유된 용량 예약에 대한 작업 (p. 422) 단원을 참조하십시오.	2019년 7월 29일
Elastic Fabric Adapter(EFA)	2016-11-15	EFA는 이제 Open MPI 3.1.4 및 Intel MPI 2019 Update 4를 지원합니다. 자세한 내용은 Elastic Fabric Adapter(EFA) (p. 731) 단원을 참조하십시오.	2019년 7월 26일
생성 시 시작 템플릿 태그	2016-11-15	생성 시 시작 템플릿 태그를 지정할 수 있습니다. 자세한 내용은 리소스에 태그 지정 (p. 1070) 단원을 참조하십시오.	2019년 7월 24일
최대 총 요금	2016-11-15	EC2 집합 및 스팟 집합에서 모든 온디맨드 인스턴스 및 스팟 인스턴스에 대해 최대 시간당 요금을 지정할 수 있습니다. 자세한 내용은 EC2 집합의 지출 제어 (p. 455) 및 스팟 집합의 지출 제어 (p. 318) 단원을 참조하십시오.	2019년 7월 1일
EC2 Instance Connect	2016-11-15	EC2 Instance Connect는 Secure Shell(SSH)을 사용하여 인스턴스에 연결하는 간단하고 안전한 방법입니다. 자세한 내용은 EC2 Instance Connect를 사용하여 Linux 인스턴스에 연결 (p. 490) 단원을 참조하십시오.	2019년 6월 27일
호스트 복구	2016-11-15	전용 호스트에서 예상치 못한 하드웨어 오류가 발생하는 경우 새 호스트에서 인스턴스를 자동으로 다시 시작합니다. 자세한 내용은 호스트 복구 (p. 404) 단원을 참조하십시오.	2019년 6월 5일
Amazon EBS 다중 볼륨 스냅샷	2016-11-15	EC2 인스턴스에 연결된 여러 EBS 볼륨에서 정확한 특정 시점, 데이터 조정 및 충돌 일치 스냅샷을 생성합니다.	2019년 5월 29일
Amazon EBS 암호화 기본 제공	2016-11-15	리전에서 기본적으로 암호화를 사용하도록 설정하면 해당 리전에서 생성한 모든 새 EBS 볼륨이 EBS 암호화용 기본 CMK를 사용하여 암호화됩니다. 자세한 내용은 암호화 기본 제공 (p. 971) 단원을 참조하십시오.	2019년 5월 23일

기능	API 버전	설명	릴리스 날짜
VPC 엔드포인트, 엔드포인트 서비스 및 엔드포인트 서비스 구성을 태그 지정합니다.	2016-11-15	VPC 엔드포인트, 엔드포인트 서비스 및 엔드포인트 서비스 구성을 태그 지정할 수 있습니다. 자세한 내용은 리소스에 태그 지정 (p. 1070) 단원을 참조하십시오.	2019년 5월 13일
I3en 인스턴스	2016-11-15	100Gbps의 네트워크 대역폭을 제공하는 새 인스턴스	2019년 5월 8일
Elastic Fabric Adapter(EFA)	2016-11-15	인스턴스에 Elastic Fabric Adapter를 연결하여 HPC(고성능 컴퓨팅) 애플리케이션 속도를 높일 수 있습니다. 자세한 내용은 Elastic Fabric Adapter(EFA) (p. 731) 단원을 참조하십시오.	2019년 4월 29일
T3a 인스턴스	2016-11-15	AMD EYPC 프로세서를 제공하는 새 인스턴스입니다.	2019년 4월 24일
M5ad 및 R5ad 인스턴스	2016-11-15	AMD EYPC 프로세서를 제공하는 새 인스턴스입니다.	2019년 3월 27일
전용 호스트 예약 태그 지정	2016-11-15	전용 호스트 예약에 태그를 지정할 수 있습니다. 자세한 내용은 전용 호스트 예약 태그 지정 (p. 400) 단원을 참조하십시오.	2019년 3월 14일
M5, M5d, R5, R5d, z1d에 대한 베어 메탈 인스턴스	2016-11-15	애플리케이션이 호스트 서버의 물리적 리소스에 직접 액세스할 수 있는 새 인스턴스입니다.	2019년 2월 13일
파티션 배치 그룹	2016-11-15	파티션 배치 그룹은 인스턴스를 논리적 파티션에 분산해, 한 파티션에 있는 인스턴스가 다른 파티션의 인스턴스와 기본 하드웨어를 공유하지 않게 합니다. 자세한 내용은 파티션 배치 그룹 (p. 759) 단원을 참조하십시오.	2018년 12월 20일
p3dn.24xlarge 인스턴스	2016-11-15	새 p3dn.24xlarge 인스턴스는 100Gbps의 네트워크 대역폭을 제공합니다.	2018년 12월 7일
EC2 Linux 인스턴스 최대 절전 모드 설정	2016-11-15	Linux 인스턴스가 최대 절전 모드를 사용하도록 설정되고 최대 절전 모드 사전 조건을 충족하는 경우 해당 인스턴스를 최대 절전 모드로 설정할 수 있습니다. 자세한 내용은 Linux 인스턴스를 최대 절전 모드로 실행합니다. (p. 510) 단원을 참조하십시오.	2018년 11월 28일
Amazon Elastic Inference 액셀러레이터	2016-11-15	Amazon EI 액셀러레이터를 인스턴스에 연결하여 GPU 구동 가속 기능을 추가함으로써 딥러닝 추론 실행 비용을 줄일 수 있습니다. 자세한 내용은 Amazon Elastic Inference (p. 598) 단원을 참조하십시오.	2018년 11월 28일
100Gbps의 네트워크 대역폭을 제공하는 인스턴스	2016-11-15	새 C5n 인스턴스는 최대 100Gbps의 네트워크 대역폭을 사용할 수 있습니다.	2018년 11월 26일
ARM 기반 프로세서를 제공하는 인스턴스	2016-11-15	새로운 A1 인스턴스는 상당한 비용 절감을 제공하며 확장 및 ARM 기반 워크로드에 매우 적합합니다.	2018년 11월 26일

기능	API 버전	설명	릴리스 날짜
스팟 콘솔에서 인스턴스 플릿 권장	2016-11-15	스팟 콘솔은 애플리케이션 요구에 맞는 최소 하드웨어 사양(vCPU, 메모리 및 스토리지)을 충족하기 위해 스팟 모범 사례(인스턴스 다각화)에 기초한 인스턴스 플릿을 권장합니다. 자세한 내용은 스팟 집합 요청 생성 (p. 339) 단원을 참조하십시오.	2018년 11월 20일
새 EC2 집합 요청 유형: `instant`	2016-11-15	EC2 집합에서는 이제 새로운 요청 유형인 `instant`를 지원합니다. 이 요청 유형은 인스턴스 유형 및 구매 모델 간에 용량을 동기적으로 프로비저닝하는 데 사용할 수 있습니다. `instant` 요청은 시작된 인스턴스를 API 응답으로 반환하며 추가 작업을 수행하지 않으므로 인스턴스가 시작되는 경우와 시점을 제어할 수 있습니다. 자세한 내용은 EC2 집합 요청 유형 (p. 452) 단원을 참조하십시오.	2018년 11월 14일
AMD EYPC 프로세서를 제공하는 인스턴스	2016-11-15	새로운 범용(M5a) 및 메모리 최적화된 인스턴스(R5a)는 마이크로서비스, 중소 규모 데이터베이스, 가상 데스크톱, 개발 및 테스트 환경, 비즈니스 애플리케이션 등을 위한 저가형 옵션을 제공합니다.	2018년 11월 6일
스팟 절감 정보	2016-11-15	단일 스팟 집합 또는 모든 스팟 인스턴스에 대해 스팟 인스턴스 사용에서 얻는 절감을 볼 수 있습니다. 자세한 내용은 스팟 인스턴스 구입으로 절감되는 비용 (p. 323) 단원을 참조하십시오.	2018년 11월 5일
CPU 옵션 최적화를 위한 콘솔 지원	2016-11-15	인스턴스를 시작하면 Amazon EC2 콘솔을 사용하여 특정 워크로드 또는 비즈니스 필요에 맞도록 CPU 옵션을 최적화할 수 있습니다. 자세한 내용은 CPU 옵션 최적화 (p. 546) 단원을 참조하십시오.	2018년 10월 31일
인스턴스에서 시작 템플릿 생성을 위한 콘솔 지원	2016-11-15	Amazon EC2 콘솔을 사용하는 새 시작 템플릿을 위한 기초로 인스턴스를 사용하는 시작 템플릿을 생성할 수 있습니다. 자세한 내용은 시작 템플릿 생성 (p. 438) 단원을 참조하십시오.	2018년 10월 30일
온디맨드 용량 예약	2016-11-15	원하는 기간 동안 특정 가용 영역의 Amazon EC2 인스턴스에 대해 용량을 예약할 수 있습니다. 이렇게 하면 예약 인스턴스(RI)에서 제공되는 결제 할인과는 별도로 용량 예약을 생성 및 관리할 수 있습니다. 자세한 내용은 온디맨드 용량 예약 (p. 415) 단원을 참조하십시오.	2018년 10월 25일
고유 IP 주소 가져오기(BYOIP)	2016-11-15	온프레미스 네트워크에서 AWS 계정으로 모든 퍼블릭 IPv4 주소 범위의 일부 또는 전체를 가져올 수 있습니다. 주소 범위를 AWS로 가져오고 나면 이러한 주소가 계정에 주소 풀로 나타납니다. 주소 풀에서 탄력적 IP 주소를 생성하여 AWS 리소스와 함께 사용할 수 있습니다. 자세한 내용은 고유 IP 주소 가져오기(BYOIP) (p. 672) 단원을 참조하십시오.	2018년 10월 23일
g3s.xlarge 인스턴스	2016-11-15	g3s.xlarge 인스턴스의 도입과 함께 액셀러레이티드 컴퓨팅 G3 인스턴스 패밀리를 확장합니다.	2018년 10월 11일

기능	API 버전	설명	릴리스 날짜
생성 및 콘솔 지원 시 전용 호스트 태그	2016-11-15	생성 시 전용 호스트를 태그 지정할 수 있으므로 Amazon EC2 콘솔을 사용하여 전용 호스트 태그를 관리할 수 있습니다. 자세한 내용은 전용 호스트 할당 (p. 385) 단원을 참조하십시오.	2018년 10월 08일
고용량 메모리 인스턴스	2016-11-15	이러한 인스턴스는 대량의 인 메모리 데이터베이스를 실행하도록 설계되었습니다. 베어 메탈 성능에 호스트 하드웨어에 대한 직접 액세스를 제공합니다. 자세한 내용은 메모리 최적화 인스턴스 (p. 232) 단원을 참조하십시오.	2018년 9월 27일
f1.4xlarge 인스턴스	2016-11-15	f1.4xlarge 인스턴스의 도입과 함께 액셀러레이티드 컴퓨팅 F1 인스턴스 패밀리를 확장합니다.	2018년 9월 25일
스팟 집합에 대한 예약된 조정 작업의 콘솔 지원	2016-11-15	날짜 및 시간을 기준으로 플릿의 현재 용량을 늘리거나 줄입니다. 자세한 내용은 예약된 조정을 사용하여 스팟 집합 조정 (p. 365) 단원을 참조하십시오.	2018년 9월 20일
T3 인스턴스	2016-11-15	T3; 인스턴스는 차세대 버스트 가능 범용 인스턴스 유형으로서, 기본 수준의 CPU 성능을 제공하면서도 필요에 따라 언제든지 CPU 사용량을 버스트할 수 있습니다. 자세한 내용은 성능 순간 확장 가능 인스턴스 (p. 194) 단원을 참조하십시오.	2018년 8월 21일
EC2 집합를 위한 할당 전략	2016-11-15	온디맨드 용량을 가격(최저 가격 우선)을 기준으로 채울지 또는 우선 순위(최우선 순위 우선)를 기준으로 채울지 지정할 수 있습니다. 대상 스팟 용량을 할당할 스팟 풀 수를 지정할 수 있습니다. 자세한 내용은 스팟 인스턴스를 위한 할당 전략 (p. 453) 단원을 참조하십시오.	2018년 7월 26일
스팟 집합를 위한 할당 전략	2016-11-15	온디맨드 용량을 가격(최저 가격 우선)을 기준으로 채울지 또는 우선 순위(최우선 순위 우선)를 기준으로 채울지 지정할 수 있습니다. 대상 스팟 용량을 할당할 스팟 풀 수를 지정할 수 있습니다. 자세한 내용은 스팟 인스턴스를 위한 할당 전략 (p. 316) 단원을 참조하십시오.	2018년 7월 26일
R5 및 R5d 인스턴스	2016-11-15	R5 및 R5d 인스턴스는 고성능 데이터베이스, 배포된 인 메모리 캐시 및 인 메모리 분석에 가장 적합합니다. R5d 인스턴스에는 NVMe 인스턴스 스토어 볼륨이 연결되어 있습니다. 자세한 내용은 메모리 최적화 인스턴스 (p. 232) 단원을 참조하십시오.	2018년 7월 25일
z1d 인스턴스	2016-11-15	이러한 인스턴스는 전자 설계 자동화(EDA) 및 관계형 데이터베이스 등 대용량 메모리와 함께 높은 코어당 성능을 필요로 하는 애플리케이션을 위해 설계되었습니다. 이러한 인스턴스에는 NVME 인스턴스 스토어 볼륨이 연결되어 있습니다. 자세한 내용은 메모리 최적화 인스턴스 (p. 232) 단원을 참조하십시오.	2018년 7월 25일

기능	API 버전	설명	릴리스 날짜
스냅샷 수명 주기 자동화	2016-11-15	Amazon 데이터 수명 주기 관리자를 사용하여 EBS 볼륨의 스냅샷 생성 및 삭제를 자동화할 수 있습니다. 자세한 내용은 Amazon EBS 스냅샷 수명 주기 자동화 (p. 947) 단원을 참조하십시오.	2018년 7월 12일
시작 템플릿 CPU 옵션	2016-11-15	명령행 도구를 사용하여 시작 템플릿을 생성할 때 특정 워크로드 또는 비즈니스 요구 사항에 맞춰 CPU 옵션을 최적화할 수 있습니다. 자세한 내용은 시작 템플릿 생성 (p. 438) 단원을 참조하십시오.	2018년 7월 11일
전용 호스트 태그	2016-11-15	전용 호스트에 태그를 지정할 수 있습니다. 자세한 내용은 전용 호스트 태그 지정 (p. 395) 단원을 참조하십시오.	2018년 7월 3일
`i3.metal` 인스턴스	2016-11-15	`i3.metal` 인스턴스에서는 애플리케이션이 프로세서, 메모리 등 호스트 서버의 물리적 리소스에 직접 액세스할 수 있습니다. 자세한 내용은 스토리지 최적화 인스턴스 (p. 242) 단원을 참조하십시오.	2018년 5월 17일
최신 콘솔 출력 받기	2016-11-15	get-console-output AWS CLI 명령을 사용하면 일부 인스턴스 유형의 최신 콘솔 출력을 검색할 수 있습니다.	2018년 5월 9일
CPU 옵션 최적화	2016-11-15	인스턴스를 시작하면 특정 워크로드 또는 비즈니스 필요에 맞도록 CPU 옵션을 최적화할 수 있습니다. 자세한 내용은 CPU 옵션 최적화 (p. 546) 단원을 참조하십시오.	2018년 5월 8일
EC2 집합	2016-11-15	EC2 집합를 사용하여 서로 다른 EC2 인스턴스 유형과 가용 영역, 온디맨드 인스턴스, 예약 인스턴스 및 스팟 인스턴스 구매 모델에서 인스턴스 그룹을 시작할 수 있습니다. 자세한 내용은 EC2 플릿 시작 (p. 449) 단원을 참조하십시오.	2018년 5월 2일
스팟 집합의 온디맨드 인스턴스	2016-11-15	항상 인스턴스 용량을 사용할 수 있도록 스팟 집합 요청에 온디맨드 용량에 대한 요청을 포함할 수 있습니다. 자세한 내용은 스팟 집합 작동 방식 (p. 316) 단원을 참조하십시오.	2018년 5월 2일
생성 시 EBS 스냅샷 태그 지정	2016-11-15	생성 중 스냅샷에 태그를 적용할 수 있습니다. 자세한 내용은 Amazon EBS 스냅샷 만들기 (p. 929) 단원을 참조하십시오.	2018년 4월 2일
배치 그룹 변경	2016-11-15	배치 그룹 내로 또는 밖으로 인스턴스를 이동하거나, 인스턴스의 배치 그룹을 변경할 수 있습니다. 자세한 내용은 인스턴스에 대한 배치 그룹 변경 (p. 764) 단원을 참조하십시오.	2018년 3월 1일
더 긴 리소스 ID	2016-11-15	더 많은 리소스 유형에 대해 더 긴 ID 형식을 활성화할 수 있습니다. 자세한 내용은 리소스 ID (p. 1060) 단원을 참조하십시오.	2018년 2월 9일

기능	API 버전	설명	릴리스 날짜
네트워크 성능 개선	2016-11-15	클러스터 배치 그룹 외부의 인스턴스에서 이제 다른 인스턴스 또는 Amazon S3 사이에서 네트워크 트래픽을 전송 또는 수신할 때 증가된 대역폭의 혜택을 누릴 수 있습니다. 자세한 내용은 네트워킹 및 스토리지 기능 (p. 182) 단원을 참조하십시오.	2018년 1월 24일
탄력적 IP 주소 태그 지정	2016-11-15	탄력적 IP 주소에 태그를 지정할 수 있습니다. 자세한 내용은 탄력적 IP 주소 태그 지정 (p. 679) 단원을 참조하십시오.	2017년 12월 21일
Amazon Linux 2	2016-11-15	Amazon Linux 2는 Amazon Linux의 새 버전입니다. 애플리케이션에 우수한 성능, 안정적이고 안전한 기반 환경을 제공합니다. 자세한 내용은 Amazon Linux (p. 161) 단원을 참조하십시오.	2017년 12월 13일
Amazon Time Sync Service	2016-11-15	Amazon Time Sync Service를 사용하여 인스턴스에서 정확한 시간을 유지할 수 있습니다. 자세한 내용은 Linux 인스턴스의 시간 설정 (p. 542) 단원을 참조하십시오.	2017년 11월 29일
T2 무제한	2016-11-15	T2 무제한 인스턴스는 필요한 기간 동안 기준 이상으로 워크로드를 버스트할 수 있습니다. 자세한 내용은 성능 순간 확장 가능 인스턴스 (p. 194) 단원을 참조하십시오.	2017년 11월 29일
시작 템플릿	2016-11-15	시작 템플릿에는 인스턴스를 시작하기 위한 파라미터가 전부 또는 일부 포함되어 있기 때문에 인스턴스를 시작할 때마다 이를 지정할 필요가 없습니다. 자세한 내용은 시작 템플릿에서 인스턴스 시작 (p. 436) 단원을 참조하십시오.	2017년 11월 29일
분산형 배치	2016-11-15	서로 떨어져 있어야 하는 중요 인스턴스의 수가 적은 애플리케이션에서는 분산형 배치 그룹이 권장됩니다. 자세한 내용은 분산형 배치 그룹 (p. 760) 단원을 참조하십시오.	2017년 11월 29일
H1 인스턴스	2016-11-15	H1 인스턴스는 고성능 빅 데이터 워크로드용으로 설계되었습니다. 자세한 내용은 스토리지 최적화 인스턴스 (p. 242) 단원을 참조하십시오.	2017년 11월 28일
M5 인스턴스	2016-11-15	M5 인스턴스는 차세대 범용 컴퓨팅 인스턴스로서, 이들은 컴퓨팅, 메모리, 스토리지 및 네트워크 리소스를 균형 있게 제공합니다.	2017년 11월 28일
스팟 인스턴스 최대 절전 모드	2016-11-15	스팟 서비스는 중단 시 스팟 인스턴스를 최대 절전 모드로 전환할 수 있습니다. 자세한 내용은 중단된 스팟 인스턴스를 최대 절전 모드로 전환 (p. 374) 단원을 참조하십시오.	2017년 11월 28일
스팟 집합 대상 추적	2016-11-15	스팟 집합에 대해 대상 추적 조정 정책을 설정할 수 있습니다. 자세한 내용은 대상 추적 정책을 사용하여 스팟 집합 조정 (p. 362) 단원을 참조하십시오.	2017년 11월 17일
Elastic Load Balancing와 스팟 집합 통합	2016-11-15	스팟 집합에 하나 이상의 로드 밸런서를 연결할 수 있습니다.	2017년 11월 10일

기능	API 버전	설명	릴리스 날짜
X1e 인스턴스	2016-11-15	X1e 인스턴스는 고성능 데이터베이스, 인메모리 데이터베이스 및 기타 메모리 집약적인 엔터프라이즈 애플리케이션에 가장 적합합니다. 자세한 내용은 메모리 최적화 인스턴스 (p. 232) 단원을 참조하십시오.	2017년 11월 28일
C5 인스턴스	2016-11-15	C5 인스턴스는 컴퓨팅 집약적 애플리케이션을 위하여 설계되었습니다. 자세한 내용은 컴퓨팅 최적화 인스턴스 (p. 227) 단원을 참조하십시오.	2017년 11월 6일
전환형 예약 인스턴스 병합 및 분할	2016-11-15	둘 이상의 전환형 예약 인스턴스를 새 전환형 예약 인스턴스 하나로 교환(병합)할 수 있습니다. 전환형 예약 인스턴스 한 개를 작은 예약 여러 개로 분리하는 수정 과정을 이용할 수도 있습니다. 자세한 내용은 전환형 예약 인스턴스 교환 (p. 304) 단원을 참조하십시오.	2017년 11월 6일
P3 인스턴스	2016-11-15	P3 인스턴스는 차세대 컴퓨팅 최적화 GPU 인스턴스입니다. 자세한 내용은 Linux 액셀러레이티드 컴퓨팅 인스턴스 (p. 248) 단원을 참조하십시오.	2017년 10월 25일
VPC 테넌시 수정	2016-11-15	VPC의 인스턴스 테넌시 속성은 `dedicated`에서 `default`로 변경할 수 있습니다. 자세한 내용은 VPC의 테넌시 변경 (p. 414) 단원을 참조하십시오.	2017년 10월 16일
초 단위 결제	2016-11-15	Amazon EC2는 Linux 기반 사용량에 대해 초 단위로 요금을 청구하며 최소 1분 요금이 부과됩니다.	2017년 10월 2일
중단 시 중지	2016-11-15	스팟 인스턴스가 중단되면 Amazon EC2가 이를 중지하거나 종료하도록 지정할 수 있습니다. 자세한 내용은 중단 동작 (p. 373) 단원을 참조하십시오.	2017년 9월 18일
NAT 게이트웨이에 태그 지정	2016-11-15	NAT 게이트웨이에 태그를 지정할 수 있습니다. 자세한 내용은 리소스에 태그 지정 (p. 1070) 단원을 참조하십시오.	2017년 9월 7일
보안 그룹 규칙 설명	2016-11-15	보안 그룹 규칙에 설명을 추가할 수 있습니다. 자세한 내용은 보안 그룹 규칙 (p. 873) 단원을 참조하십시오.	2017년 8월 31일
탄력적 IP 주소 복구	2016-11-15	VPC에서 사용하기 위해 탄력적 IP 주소를 해제한 경우 복구할 수 있습니다. 자세한 내용은 탄력적 IP 주소 복구 (p. 682) 단원을 참조하십시오.	2017년 8월 11일
스팟 집합 인스턴스 태그 지정	2016-11-15	시작하는 인스턴스에 태그를 자동으로 지정하도록 스팟 집합을 구성할 수 있습니다.	2017년 7월 24일
G3 인스턴스	2016-11-15	G3 인스턴스는 DirectX 또는 OpenGL을 사용하는 그래픽 애플리케이션을 위해 비용 효율적인 고성능 플랫폼을 제공합니다. 또한 G3 인스턴스는 해상도가 최대 4096x2160인 모니터 4대를 지원하는 NVIDIA GRID 가상 워크스테이션 기능도 제공합니다. 자세한 내용은 Linux 액셀러레이티드 컴퓨팅 인스턴스 (p. 248) 단원을 참조하십시오.	2017년 7월 13일

기능	API 버전	설명	릴리스 날짜
F1 인스턴스	2016-11-15	F1 인스턴스는 차세대 액셀러레이티드 컴퓨팅 인스턴스를 나타냅니다. 자세한 내용은 Linux 액셀러레이티드 컴퓨팅 인스턴스 (p. 248) 단원을 참조하십시오.	2017년 4월 19일
생성 시 태그 리소스	2016-11-15	생성 단계에서 인스턴스와 볼륨에 태그를 적용할 수 있습니다. 자세한 내용은 리소스에 태그 지정 (p. 1070) 단원을 참조하십시오. 또한 태그 기반 리소스 권한을 사용하여 적용되는 태그를 제어할 수도 있습니다. 자세한 내용은 생성 시 리소스 태그 지정에 대한 권한 부여 (p. 811) 단원을 참조하십시오.	2017년 3월 28일
I3 인스턴스	2016-11-15	I3 인스턴스는 차세대 스토리지 최적화 인스턴스를 나타냅니다. 자세한 내용은 스토리지 최적화 인스턴스 (p. 242) 단원을 참조하십시오.	2017년 2월 23일
연결된 EBS 볼륨을 수정합니다	2016-11-15	대부분의 EC2 인스턴스에 연결된 대다수 EBS의 경우, 볼륨을 분리하거나 인스턴스를 중지하지 않고도 볼륨 크기, 유형, IOPS를 수정할 수 있습니다. 자세한 내용은 Amazon EBS 탄력적 볼륨 (p. 958) 단원을 참조하십시오.	2017년 2월 13일
IAM 역할 연결	2016-11-15	기존 인스턴스에 대한 IAM 역할을 연결, 분리하거나 교체할 수 있습니다. 자세한 내용은 IAM의 Amazon EC2 역할 (p. 850) 단원을 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. 327) 단원을 참조하십시오.	2017년 1월 19일
IPv6 지원	2016-11-15	IPv6 CIDR을 VPC와 서브넷에 연결하고 IPv6 주소를 VPC의 인스턴스에 할당할 수 있습니다. 자세한 내용은 Amazon EC2인스턴스 IP 어드레싱 (p. 658)을 참조하십시오.	2016년 12월 1일
R4 인스턴스	2016-09-15	R4 인스턴스는 차세대 메모리 최적화 인스턴스를 나타냅니다. R4 인스턴스는 BI(비즈니스 인텔리전스), 데이터 마이닝 및 분석, 인 메모리 데이터베이스, 분산형 웹 규모 인 메모리 캐싱, 비정형 빅 데이터의 애플리케이션 성능 실시간 처리 등 메모리 집약적이고 지연 시간에 민감한 워크로드에 매우 적합합니다. 자세한 내용은 메모리 최적화 인스턴스 (p. 232)를 참조하십시오.	2016년 11월 30일
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 응답성과 뛰어난 성능이 필요한 애플리케이션용입니다. 자세한 내용은 성능 순간 확장 가능 인스턴스 (p. 194) 단원을 참조하십시오.	2016년 11월 30일

기능	API 버전	설명	릴리스 날짜
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. 248) 단원을 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. 361) 단원을 참조하십시오.	2016년 9월 1일
ENA(Elastic Network Adapter)	2016-04-01	이제 ENA를 사용하여 네트워킹 수준을 높일 수 있습니다. 자세한 내용은 향상된 네트워킹 유형 (p. 707) 단원을 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. 1060) 단원을 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. 934) 단원을 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. 1116) 단원을 참조하십시오.	2016년 5월 24일
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. 232) 단원을 참조하십시오.	2016년 5월 18일
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. 892) 단원을 참조하십시오.	2016년 4월 19일
Amazon EC2에 대한 새로운 NetworkPacketsIn 및 NetworkPacketsOut 측정치를 추가함.		Amazon EC2에 대한 새로운 NetworkPacketsIn 및 NetworkPacketsOut 측정치를 추가함. 자세한 내용은 인스턴스 지표 (p. 618) 단원을 참조하십시오.	2016년 3월 23일
스팟 집합에 대한 CloudWatch 지표		이제 스팟 집합에 대한 CloudWatch 지표를 얻을 수 있습니다. 자세한 내용은 CloudWatch에 대한 스팟 집합 지표 (p. 359) 단원을 참조하십시오.	2016년 3월 21일

기능	API 버전	설명	릴리스 날짜
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. 308) 단원을 참조하십시오.	2016년 1월 13일
더 긴 리소스 ID	2015-10-01	일부 Amazon EC2 및 Amazon EBS 리소스 유형에 좀더 긴 ID를 도입하고 있습니다. 옵트 인 기간 동안 지원되는 리소스 유형에 더 긴 ID 형식을 사용할 수 있습니다. 자세한 내용은 리소스 ID (p. 1060) 단원을 참조하십시오.	2016년 1월 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 비활성화할 수 있습니다. 자세한 내용은 ClassicLink DNS 지원 활성화 (p. 783) 단원을 참조하십시오.	2016년 1월 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 응답성과 뛰어난 성능이 필요한 애플리케이션용입니다. 자세한 내용은 성능 순간 확장 가능 인스턴스 (p. 194) 단원을 참조하십시오.	2015년 12월 15일
전용 호스트	2015-10-01	Amazon EC2 전용 호스트는 고객 전용의 인스턴스 용량을 갖춘 물리적 서버입니다. 자세한 내용은 전용 호스트 (p. 381) 단원을 참조하십시오.	2015년 11월 23일
스팟 인스턴스 기간	2015-10-01	이제 스팟 인스턴스의 지속 시간을 지정할 수 있습니다. 자세한 정보는 스팟 인스턴스의 지속 시간 정의 (p. 326) 단원을 참조하십시오.	2015년 10월 6일
스팟 집합 요청 수정	2015-10-01	이제 스팟 집합 요청의 목표 용량을 수정할 수 있습니다. 자세한 내용은 스팟 집합 요청 수정 (p. 349) 단원을 참조하십시오.	2015년 9월 29일
스팟 집합 다각화 할당 전략	2015-04-15	단일 스팟 집합 요청을 사용하여 여러 스팟 풀에서 스팟 인스턴스를 할당할 수 있습니다. 자세한 정보는 스팟 인스턴스를 위한 할당 전략 (p. 316) 단원을 참조하십시오.	2015년 9월 15일
스팟 집합 인스턴스 가중치 부여	2015-04-15	이제 각 인스턴스 유형이 애플리케이션의 성능에 기여하는 용량 단위를 정의하고, 그에 따라 각 스팟 풀의 스팟 인스턴스에 대해 지불할 금액을 조정할 수 있습니다. 자세한 정보는 스팟 집합 인스턴스 가중치 부여 (p. 319) 단원을 참조하십시오.	2015년 8월 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. 637) 단원을 참조하십시오.	2015년 7월 23일

기능	API 버전	설명	릴리스 날짜
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 응답성과 뛰어난 성능이 필요한 애플리케이션용입니다. 자세한 내용은 성능 순간 확장 가능 인스턴스 (p. 194) 단원을 참조하십시오.	2015년 6월 16일
M4 인스턴스		컴퓨팅, 메모리 및 네트워크 리소스의 균형을 제공하는 차세대 범용 인스턴스입니다. M4 인스턴스는 AVX2가 포함된 사용자 지정 2.4GHz intel® Xeon® E5 2676v3(Haswell) 프로세서에 의해 구동됩니다.	2015년 6월 11일
스팟 집합	2015-04-15	별도의 스팟 인스턴스 요청을 관리하는 대신 스팟 인스턴스의 모음 또는 집합을 관리할 수 있습니다. 자세한 내용은 스팟 집합 작동 방식 (p. 316) 단원을 참조하십시오.	2015년 5월 18일
탄력적 IP 주소의 EC2-Classic 마이그레이션	2015-04-15	EC2-Classic에서 사용하도록 할당한 탄력적 IP 주소를 VPC에서 사용하도록 마이그레이션할 수 있습니다. 자세한 내용은 EC2-Classic에서 탄력적 IP 주소 마이그레이션 (p. 774) 단원을 참조하십시오.	2015년 5월 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 가져오기를 참조하십시오.	2015년 4월 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 4월 7일
D2 인스턴스		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. 2.4GHz intel® ® 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. 자세한 내용은 스토리지 최적화 인스턴스 (p. 242) 단원을 참조하십시오.	2015년 3월 24일

기능	API 버전	설명	릴리스 날짜
EC2 인스턴스용 자동 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. 복구된 인스턴스는 인스턴스 ID, IP 주소 및 모든 인스턴스 메타데이터를 포함하여 원본 인스턴스와 동일합니다. 자세한 내용은 인스턴스 복구 (p. 528) 단원을 참조하십시오.	2015년 1월 12일
C4 인스턴스		경제적인 가격으로 매우 우수한 성능을 제공하는 컴퓨팅에 최적화된 차세대 인스턴스입니다. C4 인스턴스는 사용자 지정 2.9GHz intel® ® E5-2666 v3(Haswell) 프로세서를 기반으로 합니다. 터보 부스트 기능이 추가된 C4 인스턴스의 프로세서 클록 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, 게놈 분석 및 계산 유체역학에 매우 적합합니다. 자세한 내용은 컴퓨팅 최적화 인스턴스 (p. 227) 단원을 참조하십시오.	2015년 1월 11일
ClassicLink	2014-10-01	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. 777) 단원을 참조하십시오.	2015년 1월 7일
스팟 인스턴스 종료 공지		스팟 인스턴스 중단을 방지하는 가장 좋은 방법은 내결함성을 유지하도록 애플리케이션을 설계하는 것입니다. 또한 Amazon EC2가 스팟 인스턴스를 종료하기 2분 전에 경고하는 스팟 인스턴스 종료 공지를 이용할 수 있습니다. 자세한 내용은 스팟 인스턴스 중단 공지 (p. 376) 단원을 참조하십시오.	2015년 1월 5일
`DescribeVolumes` 페이지 매김 지원	2014-09-01	이제 `DescribeVolumes` API 호출에서는 `MaxResults` 및 `NextToken` 파라미터를 사용한 결과의 페이지 매김을 지원합니다. 자세한 내용은 Amazon EC2 API Reference의 DescribeVolumes를 참조하십시오.	2014년 10월 23일

기능	API 버전	설명	릴리스 날짜
T2 인스턴스	2014-06-15	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 응답성과 뛰어난 성능이 필요한 애플리케이션용입니다. 자세한 내용은 성능 순간 확장 가능 인스턴스 (p. 194) 단원을 참조하십시오.	2014년 6월 30일
새 EC2 서비스 제한 페이지		Amazon EC2 콘솔의 EC2 서비스 제한 페이지에서는 리전별로 Amazon EC2 및 Amazon VPC에서 제공하는 리소스의 현재 제한을 볼 수 있습니다.	2014년 6월 19일
Amazon EBS 범용 SSD 볼륨	2014-05-01	범용 SSD볼륨은 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. 894) 단원을 참조하십시오.	2014년 6월 16일
Amazon EBS 암호화	2014-05-01	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. 968) 단원을 참조하십시오.	2014년 5월 21일
R3 인스턴스	2014-02-01	경제적인 가격의 GiB RAM이 장착된 고성능 차세대 메모리 최적화 인스턴스. 이러한 인스턴스는 R3 인스턴스의 고급 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. Amazon EC2 인스턴스 유형별 하드웨어 사양에 대한 자세한 내용은 Amazon EC2 인스턴스 유형을 참조하십시오.	2014년 4월 9일
새 Amazon Linux AMI 릴리스		Amazon Linux AMI 2014.03이 릴리스되었습니다.	2014년 3월 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. 1081) 단원을 참조하십시오.	2014년 1월 28일

기능	API 버전	설명	릴리스 날짜
추가 M3 인스턴스	2013-10-15	이제 M3 인스턴스 크기 `m3.medium` 및 `m3.large`가 지원됩니다. Amazon EC2 인스턴스 유형별 하드웨어 사양에 대한 자세한 내용은 Amazon EC2 인스턴스 유형을 참조하십시오.	2014년 1월 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. 242) 단원을 참조하십시오.	2013년 12월 19일
M3 인스턴스 업데이트	2013-10-15	M3 인스턴스 크기 `m3.xlarge` 및 `m3.2xlarge`에서는 이제 SSD 볼륨이 장착된 인스턴스 스토어를 지원합니다.	2013년 12월 19일
Linux 가상 머신 가져오기	2013-10-15	VM Import 프로세스에서 이제 Linux 인스턴스 가져오기를 지원합니다. 자세한 내용은 VM Import/Export 사용 설명서 단원을 참조하십시오.	2013년 12월 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 EC2에 대한 자격 증명 및 액세스 관리 (p. 803) 단원을 참조하십시오.	2013년 11월 20일
C3 인스턴스	2013-10-15	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, 게놈 분석 및 계산 유체 역학에 매우 적합합니다. Amazon EC2 인스턴스 유형별 하드웨어 사양에 대한 자세한 내용은 Amazon EC2 인스턴스 유형을 참조하십시오.	2013년 11월 14일
AWS Marketplace에서 인스턴스 시작		이제 AWS Marketplace 시작 마법사를 사용하여 Amazon EC2에서 인스턴스를 시작할 수 있습니다. 자세한 내용은 AWS Marketplace 인스턴스 시작 (p. 448) 단원을 참조하십시오.	2013년 11월 11일
G2 인스턴스	2013-10-01	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. 248) 단원을 참조하십시오.	2013년 11월 4일

기능	API 버전	설명	릴리스 날짜
새로운 시작 마법사		재설계된 새로운 EC2 시작 마법사가 제공됩니다. 자세한 내용은 인스턴스 시작 마법사를 사용하여 인스턴스 시작 (p. 431) 단원을 참조하십시오.	2013년 10월 10일
Amazon EC2 예약 인스턴스의 인스턴스 유형 수정	2013-10-01	이제 동일 패밀리(예: M1, M2, M3, C1) 내에서 Linux 예약 인스턴스의 인스턴스 유형을 수정할 수 있습니다. 자세한 내용은 예약 인스턴스 수정 (p. 297) 단원을 참조하십시오.	2013년 10월 9일
새 Amazon Linux AMI 릴리스		Amazon Linux AMI 2013.09가 릴리스되었습니다.	2013년 9월 30일
Amazon EC2 예약 인스턴스 수정	2013-08-15	이제 리전에서 예약 인스턴스를 수정할 수 있습니다. 자세한 내용은 예약 인스턴스 수정 (p. 297) 단원을 참조하십시오.	2013년 9월 11일
퍼블릭 IP 주소 배정	2013-07-15	이제는 VPC에서 인스턴스를 시작할 때 퍼블릭 IP 주소를 배정할 수 있습니다. 자세한 내용은 인스턴스 시작 시 퍼블릭 IPv4 주소 배정 (p. 663) 단원을 참조하십시오.	2013년 8월 20일
리소스 수준의 권한 부여	2013-06-15	Amazon EC2에서는 새로운 Amazon 리소스 이름(ARN)과 조건 키를 지원합니다. 자세한 내용은 IAM에 대한 Amazon EC2 정책 (p. 806) 단원을 참조하십시오.	2013년 7월 8일
증분형 스냅샷 사본	2013-02-01	이제 증분형 스냅샷 사본을 사용할 수 있습니다. 자세한 내용은 Amazon EBS 스냅샷 복사 (p. 934) 단원을 참조하십시오.	2013년 6월 11일
새 태그 페이지		Amazon EC2 콘솔에 새 태그 페이지가 있습니다. 자세한 내용은 Amazon EC2리소스에 태그 지정 (p. 1068) 단원을 참조하십시오.	2013년 4월 4일
새 Amazon Linux AMI 릴리스		Amazon Linux AMI 2013.03이 릴리스되었습니다.	2013년 3월 27일
추가 EBS에 최적화된 인스턴스 유형	2013-02-01	이제 인스턴스 유형 `c1.xlarge`, `m2.2xlarge`, `m3.xlarge` 및 `m3.2xlarge`를 EBS에 최적화된 인스턴스로 시작할 수 있습니다. 자세한 내용은 Amazon EBS 최적화 인스턴스 (p. 984) 단원을 참조하십시오.	2013년 3월 19일
리전 간 AMI 복사	2013-02-01	리전 간에 AMI를 복사하여 둘 이상의 AWS 리전에서 일관된 인스턴스를 빠르고 쉽게 시작할 수 있습니다. 자세한 내용은 AMI 복사 (p. 150) 단원을 참조하십시오.	2013년 3월 11일

기능	API 버전	설명	릴리스 날짜
인스턴스를 기본 VPC로 시작	2013-02-01	AWS 계정에서는 리전별로 EC2-Classic 또는 VPC에서, 또는 VPC에서만 인스턴스를 시작할 수 있습니다. 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.	2013년 3월 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 1월 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 12월 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. 934) 단원을 참조하십시오.	2012년 12월 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 Amazon EBS의 Amazon CloudWatch 지표 (p. 1011) 단원을 참조하십시오. 또한 프로비저닝된 IOPS SSD 볼륨에 대한 새로운 상태 확인도 추가했습니다. 자세한 내용은 EBS 볼륨 상태 확인 (p. 918) 단원을 참조하십시오.	2012년 11월 20일
Linux 커널		AKI ID를 업데이트하고, 배포 커널을 재구성했으며, PVOps 섹션을 업데이트했습니다.	2012년 11월 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 유형을 참조하십시오.	2012년 10월 31일
스팟 인스턴스 요청 상태	2012-10-01	스팟 인스턴스 요청 상태를 사용하면 스팟 요청의 상태를 쉽게 확인할 수 있습니다.	2012년 10월 14일
새 Amazon Linux AMI 릴리스		Amazon Linux AMI 2012.09이 릴리스되었습니다.	2012년 10월 11일
Amazon EC2 예약 인스턴스 마켓플레이스	2012-08-15	예약 인스턴스 마켓플레이스은 더 이상 필요하지 않은 Amazon EC2 예약 인스턴스를 보유한 판매자와 추가 용량을 원하는 구매자를 연결합니다. 예약 인스턴스 마켓플레이스을 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 9월 11일

기능	API 버전	설명	릴리스 날짜
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. 892) 단원을 참조하십시오.	2012년 7월 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 7월 18일
Amazon EC2 인스턴스의 IAM 역할	2012-06-01	Amazon EC2의 IAM 역할은 다음을 제공합니다. • 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	2012년 6월 11일
더 쉽게 시작하고 중단 가능성을 처리할 수 있게 하는 스팟 인스턴스 기능		이제 다음과 같이 스팟 인스턴스를 관리할 수 있습니다. • Auto Scaling 시작 구성을 사용하여 스팟 인스턴스에 대해 지불할 금액을 지정하고 스팟 인스턴스에 대해 지불할 금액을 지정하기 위한 일정을 설정합니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Auto Scaling 그룹에서 스팟 인스턴스 시작을 참조하십시오. • 인스턴스가 시작되거나 종료될 때 알림을 받습니다. • AWS CloudFormation 템플릿을 사용하여 AWS 리소스를 포함하는 스택에서 스팟 인스턴스를 시작합니다.	2012년 6월 7일
EC2 인스턴스 내보내기와 Amazon EC2 상태 확인을 위한 타임스탬프	2012-05-01	상태 확인이 실패한 날짜 및 시간을 나타내는 인스턴스 상태 및 시스템 상태의 타임스탬프에 대한 지원을 추가했습니다.	2012년 5월 25일
EC2 인스턴스 내보내기와, Amazon VPC에 대한 인스턴스 및 시스템 상태 확인 시 타임스탬프	2012-05-01	Citrix Xen, Microsoft Hyper-V 및 VMware vSphere로 EC2 인스턴스 내보내기 지원을 추가했습니다. 인스턴스 및 시스템 상태 확인 시 타임스탬프 지원을 추가했습니다.	2012년 5월 25일
클러스터 컴퓨팅 에이트 엑스트라 라지 인스턴스	2012-04-01	VPC에서 `cc2.8xlarge` 인스턴스에 대한 지원을 추가했습니다.	2012년 4월 26일
AWS Marketplace AMI	2012-04-01	AWS Marketplace AMI 지원을 추가했습니다.	2012년 4월 19일
새 Linux AMI 릴리스		Amazon Linux AMI 2012.03이 릴리스되었습니다.	2012년 3월 28일

기능	API 버전	설명	릴리스 날짜
새 AKI 버전		AWS GovCloud (US) 리전용 AKI 버전 1.03 및 AKI를 릴리스했습니다.	2012년 3월 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 3월 7일
예약된 인스턴스 요금 계층	2011-12-15	예약 인스턴스 요금 계층에 기본 제공되는 할인 요금을 활용하는 방법을 설명하는 새로운 섹션을 추가했습니다.	2012년 3월 5일
Amazon Virtual Private Cloud의 EC2 인스턴스용 ENI	2011-12-01	VPC의 EC2 인스턴스용 ENI(탄력적 네트워크 인터페이스)에 대한 새로운 섹션을 추가했습니다. 자세한 내용은 탄력적 네트워크 인터페이스 (p. 683) 단원을 참조하십시오.	2011년 12월 21일
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 12월 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 12월 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. 603) 단원을 참조하십시오.	2011년 11월 16일
Amazon EC2 클러스터 컴퓨팅 인스턴스 유형		Amazon EC2에 클러스터 컴퓨팅 에이트 엑스트라 라지(cc2.8xlarge) 지원을 추가했습니다.	2011년 11월 14일
새 PDX 리전 및 AKI		새 US-West 2 리전용 새 AKI 릴리스에 대한 정보를 추가했습니다.	2011년 11월 8일
Amazon VPC의 스팟 인스턴스	2011-07-15	Amazon VPC의 스팟 인스턴스 지원에 대한 정보를 추가했습니다. 이 업데이트로 사용자가 Virtual Private Cloud(VPC)에서 스팟 인스턴스를 시작할 수 있습니다. 스팟 인스턴스 사용자가 VPC에서 스팟 인스턴스를 시작하면 Amazon VPC의 혜택을 누릴 수 있습니다.	2011년 10월 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 9월 26일

기능	API 버전	설명	릴리스 날짜
CLI 도구 사용자를 위한 간소화된 VM Import 프로세스	2011-07-15	이제 `ImportInstance` 및 `ImportVolume` 기능의 향상으로 VM Import 프로세스가 간소화되어 가져오기 작업을 생성한 후 이미지가 Amazon EC2로 업로드됩니다. 또한 `ResumeImport`가 도입되면서 사용자가 완료되지 않은 업로드를 작업이 중지된 시점부터 다시 시작할 수 있습니다.	2011년 9월 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 단원을 참조하십시오.	2011년 8월 24일
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 6월 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 자체 Linux 커널 활성화 (p. 171) 단원을 참조하십시오.	2011년 6월 20일
스팟 인스턴스 가용 영역 요금 변경	2011-05-15	스팟 인스턴스 가용 영역 요금 기능에 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 5월 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 EC2에 대한 자격 증명 및 액세스 관리 (p. 803) 단원을 참조하십시오.	2011년 4월 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 자체 Linux 커널 활성화 (p. 171) 단원을 참조하십시오.	2011년 4월 26일

기능	API 버전	설명	릴리스 날짜
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. 410) 단원을 참조하십시오.	2011년 3월 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. 273) 단원을 참조하십시오.	2011년 3월 27일
새 Amazon Linux 참조 AMI		새 Amazon Linux 참조 AMI가 CentOS 참조 AMI를 대체합니다. CentOS 5.4 AMI에서 클러스터 인스턴스의 클록 드리프트 해결이라는 섹션을 포함하여 CentOS 참조 AMI에 대한 정보를 삭제했습니다.	2011년 3월 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. 571) 및 인스턴스 메타데이터 카테고리 (p. 588) 단원을 참조하십시오.	2011년 3월 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 3월 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. 924) 단원을 참조하십시오.	2011년 2월 23일
인스턴스 종료 방지		이제 AWS Management Console을 사용하여 인스턴스가 종료되는 것을 방지할 수 있습니다. 자세한 내용은 인스턴스에 대한 종료 방지 기능 활성화 (p. 524) 단원을 참조하십시오.	2011년 2월 23일
CentOS 5.4 AMI에서 클러스터 인스턴스의 클록 드리프트 해결		Amazon의 CentOS 5.4 AMI에서 실행 중인 클러스터 인스턴스의 클록 드리프트 문제를 해결하는 방법에 대한 정보를 추가했습니다.	2011년 1월 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 단원을 참조하십시오.	2010년 12월 15일

기능	API 버전	설명	릴리스 날짜
인스턴스 기본 모니터링	2010-08-31	EC2 인스턴스 기본 모니터링에 대한 정보를 추가했습니다.	2010년 12월 12일
필터와 태그	2010-08-31	리소스 목록, 필터링 및 태그에 대한 정보를 추가했습니다. 자세한 내용은 리소스 목록화 및 필터링 (p. 1065) 및 Amazon EC2리소스에 태그 지정 (p. 1068) 단원을 참조하십시오.	2010년 9월 19일
멱등성 인스턴스 시작	2010-08-31	인스턴스 실행 시 멱등성 유지에 대한 정보를 추가했습니다. 자세한 내용은 Amazon EC2 API Reference의 멱등성 보장을 참조하십시오.	2010년 9월 19일
마이크로 인스턴스	2010-06-15	Amazon EC2는 특정 애플리케이션 유형을 위한 `t1.micro` 인스턴스 유형을 제공합니다. 자세한 내용은 다음을 참조하십시오. 성능 순간 확장 가능 인스턴스 (p. 194).	2010년 9월 8일
Amazon EC2용 AWS Identity and Access Management		Amazon EC2가 이제 AWS Identity and Access Management(IAM)와 통합되었습니다. 자세한 내용은 Amazon EC2에 대한 자격 증명 및 액세스 관리 (p. 803) 단원을 참조하십시오.	2010년 9월 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 유형을 참조하십시오.	2010년 7월 12일
Amazon VPC IP 주소 지정	2010-06-15	Amazon VPC 사용자는 이제 IP 주소를 지정하여 VPC에서 시작된 인스턴스를 배정할 수 있습니다.	2010년 7월 12일
Amazon EBS 볼륨에 대한 Amazon CloudWatch 모니터링		이제 Amazon EBS 볼륨에 대해 Amazon CloudWatch 모니터링을 자동으로 사용할 수 있습니다. 자세한 내용은 Amazon EBS의 Amazon CloudWatch 지표 (p. 1011) 단원을 참조하십시오.	2010년 6월 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 유형을 참조하십시오.	2010년 2월 22일