aws

사용자 가이드

Amazon Elastic Compute Cloud

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Elastic Compute Cloud: 사용자 가이드

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

Amazon EC2란 무엇인가요? 1
Features 1
관련 서비스 2
EC2 액세스 3
요금 5
추정, 결제 및 비용 최적화 6
리소스 6
시작하기 자습서 7
1단계: 인스턴스 시작 8
2단계: 인스턴스에 연결 10
3단계: 인스턴스 정리 13
다음 단계 14
모범 사례 15
Amazon Machine Image(AMI) 17
AMI 특성 18
시작 권한 18
루트 디바이스 유형 19
AMI 루트 디바이스 유형 결정 21
가상화 유형 22
AMI 찾기 23
Systems Manager 파라미터 27
Systems Manager 퍼블릭 파라미터 30
AWS Marketplace의 유료 AMI 33
AWS Marketplace에서 AMI 판매 34
유료 AMI 찾기 34
유료 AMI 구입 36
제품 코드 검색 37
유료 지원 사용 38
유료 및 지원 AMI에 대한 청구서 38
구독 관리 39
AMI 수명 주기 40
AMI 생성 41
인스턴스 스토어 지원 AMI 생성 49
Windows Sysprep을 사용하여 AMI 생성 87

AMI 복사 104
AMI 저장 및 복원 116
소스 AMI 식별 126
AMI를 마지막으로 사용한 시점 확인 127
AMI 사용 중지 129
AMI 비활성화 135
AMI 등록 취소 141
부팅 모드 147
UEFI 부팅 모드에 대한 요구 사항 148
AMI 부팅 모드 파라미터 150
인스턴스 유형 부팅 모드 152
인스턴스 부팅 모드 155
운영 체제 부팅 모드 156
AMI 부팅 모드 설정 158
UEFI 변수 165
UEFI 보안 부팅 165
AMI 암호화 180
인스턴스 시작 시나리오 180
이미지 복사 시나리오 183
공유 AMI 185
확인된 공급 업체 186
공유 AMI 찾기 187
Linux용 공유 AMI 사용 준비 190
허용된 AMI 191
AMI를 퍼블릭으로 설정 205
AMI에 대한 퍼블릭 액세스 치단 209
조직 및 조직 단위와 AMI 공유 218
특정 AWS 계정과 AMI 공유 230
계정과 AMI 공유 취소 233
공유 Linux AMI 생성을 위한 권장 사항 234
AMI 이벤트 모니터링 240
이벤트 세부 정보 241
available 이벤트 242
failed 이벤트 243
deregistered 이벤트 243
disabled 이벤트 244

AMI 결제 이해 245
AMI 결제 필드 245
AMI 결제 정보 찾기 247
청구서의 AMI 요금 확인 250
AMI 할당량 250
AMI의 할당량 증가 요청 251
Instances 253
인스턴스 타입 254
사용 가능한 인스턴스 유형 255
하드웨어 사양 255
하이퍼바이저 유형 256
AMI 가상화 유형 256
Processors 257
인스턴스 유형 찾기 259
EC2 인스턴스 유형 파인더 267
Compute Optimizer 권장 사항 269
인스턴스 유형 변경 271
성능 순간 확장 가능 인스턴스 280
GPU 인스턴스 328
Mac 인스턴스 342
EBS 최적화 384
CPU 옵션 471
AMD SEV-SNP 648
프로세서 상태 제어 655
관리형 인스턴스 658
관리형 인스턴스에 대한 결제 658
관리형 인스턴스 식별 659
관리형 인스턴스 시작하기 661
결제 및 구매 옵션 661
온디맨드 인스턴스 662
예약 인스턴스 664
Spot Instances 728
전용 호스트 823
전용 인스턴스 880
용량 예약 887
시작 템플릿 997

제한 사항 998
권한 999
인스턴스 시작 제어 1006
생성 1008
수정(버전 관리) 1023
삭제 1027
인스턴스 시작 1029
자습서 1032
인스턴스 파라미터 참조 1054
인스턴스 시작 마법사를 사용하여 시작 1066
시작 템플릿을 사용하여 시작 1069
기존 인스턴스에서 시작 1076
AWS Marketplace AMI에서 시작 1078
인스턴스에 연결합니다 1080
일반 연결 사전 조건 1081
SSH를 사용하여 Linux 인스턴스에 연결 1087
RDP를 사용하여 Windows 인스턴스에 연결 1102
세션 관리자를 사용하여 연결 1111
EC2 Instance Connect를 사용한 연결 1112
EC2 Instance Connect 엔드포인트를 사용하여 연결 1146
인스턴스 상태 변경 1173
인스턴스 상태별 청구 1174
보류 중인 인스턴스 1176
중지된 인스턴스 1176
최대 절전 모드로 전환된 인스턴스 1176
재부팅 중인 인스턴스 1177
종료된 인스턴스 1177
인스턴스 상태 간 차이 1178
중지 및 시작 1180
최대 절전 모드 1193
재부팅 1222
Terminate 1224
만료 1237
자동 인스턴스 복구 1242
자동 인스턴스 복구의 주요 개념 1243
간소화된 자동 복구와 CloudWatch 작업 기반 복구의 차이점 1245

복원력 있는 시스템 구축 1246
자동 복구가 발생했는지 확인 1246
간소화된 자동 복구 1248
CloudWatch 작업 기반 복구 1253
인스턴스 메타데이터 1257
인스턴스 메타데이터 카테고리 1258
동적 데이터 카테고리 1271
인스턴스 메타데이터에 액세스 1272
IMDS 옵션 구성 1308
시작 시 명령 실행 1335
예시: AMI 시작 인덱스 값 1360
호스트가 EC2 인스턴스인지 감지 1364
인스턴스 자격 증명 문서 검사 1364
시스템 UUID 검사 1365
시스템 가상 머신 생성 식별자 검사 1366
인스턴스 자격 증명 문서 1371
인스턴스 자격 증명 문서 검색 1373
인스턴스 자격 증명 문서 확인 1374
퍼블릭 인증서 1385
클록 동기화 1442
하드웨어 패킷 타임스탬핑 1443
윤초 1444
로컬 Amazon Time Sync Service 사용 1445
퍼블릭 Amazon Time Sync Service 사용 1457
Linux 인스턴스의 타임스탬프 비교 1459
인스턴스의 시간대 변경 1460
디바이스 드라이버 관리 1463
네트워크 드라이버 1463
그래픽 드라이버 1463
스토리지 디바이스 드라이버 1464
AMD 드라이버 1464
NVIDIA 드라이버 1470
Windows에 ENA 드라이버 설치 1511
Windows PV 드라이버 1527
AWS NVMe 드라이버 1560
Windows 인스턴스 구성 1570

Windows 관련 시스템 설정 1570
Windows 인스턴스의 AWS 디바이스 드라이버 1571
Windows 시작 에이전트 1572
Windows용 EC2 Fast Launch 1729
Windows 관리자 암호 변경 1751
Windows 시스템 구성 요소 추가 1752
Windows에 WSL 설치 1757
Windows 유틸리티 1759
Windows 인스턴스 업그레이드 1761
현재 위치 업그레이드 수행 1762
자동 업그레이드 수행 1766
Nitro 기반 인스턴스 유형으로 마이그레이션 1776
업그레이드 문제 해결 1785
튜토리얼: RDS 데이터베이스에 EC2 인스턴스 연결 1786
자습서 목표 1786
컨텍스트 1786
아키텍처 1787
고려 사항 1788
자습서를 완료하는 데 걸리는 시간 1789
비용 1789
옵션 1: EC2 콘솔을 사용하여 자동으로 연결 1789
옵션 2: RDS 콘솔을 사용하여 자동으로 연결 1801
옵션 3: 수동으로 연결 1811
플릿 1821
기능 및 이점 1821
어떤 플릿 방법을 사용할까요? 1822
구성 옵션 1823
요청 유형 1825
지출 한도 1852
속성 기반 인스턴스 유형 선택 1854
인스턴스 가중치 부여 1890
할당 전략 1892
용량 재조정 1898
용량 예약 1903
EC2 플릿 사용 1904
EC2 집합 요청 상태 1905

EC2 집합 사전 조건 1906
EC2 집합 생성 1910
EC2 집합 태깅 1919
EC2 플릿 설명 1922
EC2 집합 수정 1926
EC2 집합 삭제 1928
스팟 플릿 사용 1933
스팟 플릿 요청 상태 1934
스팟 플릿 권한 1935
스팟 플릿 생성 1945
스팟 플릿 태깅 1953
스팟 플릿 설명 1962
스팟 플릿 요청 수정 1963
스팟 플릿 요청 취소(삭제) 1965
스팟 플릿의 자동 크기 조정 1968
EC2 플릿 모니터링 1977
CloudWatch를 사용하여 플릿 모니터링 1978
EventBridge를 사용하여 플릿 모니터링 1981
자습서 1999
튜토리얼: 인스턴스 가중치를 사용하도록 EC2 플릿 구성 2000
튜토리얼: 온디맨드 인스턴스를 기본 용량으로 사용하도록 EC2 플릿 구성 2004
튜토리얼: 대상으로 지정된 용량 예약을 사용하여 온디맨드 인스턴스를 시작하도록 EC2 플릿 구성 2005
튜토리얼: 인스턴스를 용량 블록으로 시작하도록 EC2 플릿을 구성 2011
EC2 플릿에 대한 CLI 구성 예제 2014
예 1: 스팟 인스턴스를 기본 구입 옵션으로 시작 2015
예 2: 온디맨드 인스턴스를 기본 구입 옵션으로 시작 2015
예 3: 온디맨드 인스턴스를 기본 용량으로 시작 2016
예제 4: 여러 용량 예약을 사용하여 온디맨드 인스턴스 시작 2016
예제 5: 총 목표 용량이 미사용 용량 예약 수보다 많은 경우 용량 예약을 사용하여 온디맨드 인스턴스 시작 2020
예제 6: 대상으로 지정된 용량 예약을 사용하여 온디맨드 인스턴스 시작 2023
예제 7: 대체 스팟 인스턴스를 시작하도록 용량 리밸런싱 구성 2027
예제 8: 용량 최적화 플릿에서 스팟 인스턴스 시작 2028
예제 9: 용량 최적화 플릿에서 우선순위를 사용하여 스팟 인스턴스 시작 2029
예제 10: price-capacity-optimized 플릿에서 스팟 인스턴스 시작 2031

예제 11: 속성 기반 인스턴스 유형 선택 구성 ... 2032
예제 CLI 구성 스팟 플릿 ... 2033
예제 1: 최저 가격의 가용 영역 ... 2034
예제 2: 목록에서 최저 가격의 가용 영역 ... 2035
예제 3: 목록에서 최저 가격의 인스턴스 유형 ... 2037
예 4. 요청에 대한 가격 재정의 ... 2038
예제 5: 다각화된 할당 전략 사용 ... 2040
예제 6: 인스턴스 가중치 사용 ... 2043
예제 7: 온디맨드 용량 사용 ... 2045
예제 8: 용량 리밸런싱 구성 ... 2046
예제 9: 용량 최적화 플릿 사용 ... 2047
예제 10: 속성과 함께 용량 최적화 플릿 사용 ... 2048
예제 11: priceCapacityOptimized 플릿 사용 ... 2049
예제 12: 속성 기반 인스턴스 유형 선택 구성 ... 2051
플릿 할당량 ... 2052
목표 용량의 할당량 증가 요청 ... 2053
네트워킹 ... 2054
리전 및 영역 ... 2055
리전 ... 2055
가용 영역 ... 2056
로컬 영역 ... 2058
Wavelength Zone ... 2059
AWS Outposts ... 2060
인스턴스 IP 주소 지정 ... 2062
프라이빗 IPv4 주소 ... 2063
퍼블릭 IPv4 주소 ... 2064
퍼블릭 IPv4 주소 최적화 ... 2065
IPv6 주소 ... 2067
다중 IP 주소 ... 2068
EC2 인스턴스 호스트 이름 ... 2069
링크-로컬 주소 ... 2069
IPv4 주소 ... 2070
IPv6 주소 ... 2076
보조 IP 주소 ... 2081
Windows의 IPv4 주소 ... 2085
EC2 인스턴스 호스트 이름 및 도메인 ... 2093

EC2 인스턴스 호스트 이름 및 도메인 이해 2093
호스트 이름 유형 2094
고유 IP 주소 가져오기 2109
BYOIP 정의 2110
요구 사항 및 할당량 2110
리전별 가용성 2111
로컬 영역 가용성 2111
사전 조건 2113
주소 범위 온보딩 2121
주소 범위 사용 2130
탄력적 IP 주소 2131
탄력적 IP 주소 요금 2132
탄력적 IP 주소 기본 사항 2132
탄력적 IP 주소 할당량 2133
탄력적 IP 주소 연결 2133
탄력적 IP 주소 전송 2138
탄력적 IP 주소 릴리스 2143
이메일 애플리케이션에 역방향 DNS 사용 2144
네트워크 인터페이스 2148
네트워크 인터페이스 개념 2150
네트워크 카드 2152
네트워크 인터페이스당 IP 주소 2154
네트워크 인터페이스 생성 2156
네트워크 인터페이스 연결 2161
IP 주소 관리 2165
네트워크 인터페이스 속성 수정 2168
여러 네트워크 인터페이스 2172
요청자 관리 네트워크 인터페이스 2175
접두사 위임 2177
네트워크 인터페이스 삭제 2188
네트워크 대역폭 2188
사용 가능한 인스턴스 대역폭 2189
대역폭 가중치 2192
인스턴스 대역폭 모니터링 2199
향상된 네트워킹 2200
ENA(Elastic Network Adapter) 2201

ENA Express 2216
Intel 82599 VF 2241
네트워크 성능 모니터링 2253
Linux에서 ENA 문제 해결 2264
Windows에서 ENA 문제 해결 2277
Linux에서 네트워크 지연 시간 개선 2295
Nitro 성능 고려 사항 2299
Windows에서 네트워크 성능 최적화 2309
Elastic Fabric Adapter 2310
EFA 기본 사항 2311
지원되는 인터페이스 및 라이브러리 2314
지원되는 인스턴스 유형 2314
지원되는 운영 체제 2324
EFA 제한 사항 2325
EFA 요금 2326
EFA 및 MPI 시작하기 2326
EFA 및 NCCL 시작하기 2343
네트워크 대역폭 극대화 2365
EFA 생성 및 연결 2372
EFA 분리 및 삭제 2377
EFA 모니터링 2377
EFA 설치 프로그램 확인 2383
릴리스 정보 2396
인스턴스 토폴로지 2406
작동 방법 2407
사전 조건 2411
예시 2412
배치 그룹 2425
배치 전략 2426
배치 그룹 생성 2431
인스턴스 배치 변경 2433
배치 그룹 삭제 2434
공유된 배치 그룹 2435
AWS Outposts에서의 배치 그룹 2438
네트워크 MTU 2439
점보 프레임(9001 MTU) 2439

경로 MTU 검색 2440
인스턴스의 MTU 설정 2441
문제 해결 2447
Virtual Private Cloud 2447
기본 VPC 2448
기본이 아닌 VPC 2449
인터넷 액세스 2450
공유 서브넷 2450
IPv6 전용 서브넷 2451
보안 2452
데이터 보호 2453
Amazon EBS 데이터 보안 2454
저장 시 암호화 2454
전송 중 암호화 2455
인프라 보안 2457
네트워크 격리 2458
물리적 호스트에서 격리 2458
네트워크 트래픽 제어 2458
복원성 2461
규정 준수 확인 2462
Identity and Access Management 2463
ID 기반 정책 2463
API에 대한 정책 예제 2474
콘솔용 예제 정책 2515
AWS 관리형 정책 2526
IAM 역할 2531
업데이트 관리 2542
Windows 인스턴스 모범 사례 2542
높은 수준의 보안 모범 사례 2543
업데이트 관리 2544
구성 관리 2546
변경 관리 2547
Amazon EC2 Windows 인스턴스에 대한 감사 및 책임 2547
키 페어 2548
키 페어 생성 2549
키 페어 설명 2556

키 페어 삭제 2561
Linux 인스턴스에 대한 퍼블릭 키 추가 또는 바꾸기 2562
지문 확인 2564
보안 그룹 2566
개요 2567
보안 그룹 생성 2568
인스턴스에 대한 보안 그룹 변경 2570
보안 그룹 삭제 2576
연결 추적 2577
다양한 사용 사례에 대한 보안 그룹 규칙 2582
NitroTPM 2588
요구 사항 2589
NitroTPM용 Linux AMI 활성화 2590
AMI가 NitroTPM에 대해 활성화되어 있는지 확인 2592
NitroTPM 활성화 또는 중지 2594
인스턴스가 NitroTPM에 대해 활성화되어 있는지 확인 2594
퍼블릭 인증 키 검색 2596
Windows 인스턴스용 Credential Guard 2598
사전 조건 2598
지원되는 인스턴스 시작 2599
메모리 무결성 비활성화 2600
Credential Guard 켜기 2601
Credential Guard가 실행 중인지 확인 2603
AWS PrivateLink 2604
인터페이스 VPC 엔드포인트 생성 2605
엔드포인트 정책 생성 2605
스토리지 2607
AWS 스토리지 요금 2608
Amazon EBS 2608
EBS 볼륨 한도 2609
Amazon EC2 인스턴스 스토어 2612
데이터 지속성 2614
인스턴스 저장소 볼륨 제한 2616
SSD 인스턴스 스토어 볼륨 2620
인스턴스 스토어 볼륨 추가 2624
M1 및 C1 인스턴스의 스왑 볼륨을 활성화합니다. 2632

인스턴스 스토어 볼륨 초기화 2635
루트 볼륨 2636
Amazon EBS 지원 인스턴스 2637
인스턴스 스토어 지원 인스턴스(Linux 인스턴스만 해당) 2638
인스턴스 종료 후에도 루트 볼륨 유지 2639
루트 볼륨 교체 2643
볼륨 디바이스 이름 2653
사용 가능한 디바이스 이름 2654
디바이스 이름 고려 사항 2656
블록 디바이스 매핑 2658
블록 디바이스 매핑의 개념 2658
AMI에 블록 디바이스 매핑 추가 2662
인스턴스에 블록 디바이스 매핑 추가 2666
Windows 인스턴스용 볼륨 연결 및 매핑 방법 2675
볼륨에 NVME 디스크 매핑 2675
볼륨에 비 NVME 디스크 매핑 2681
찢긴 쓰기 방지 2691
지원되는 블록 크기 2692
요구 사항 2693
인스턴스 지원 확인 2693
워크로드 구성 2695
Windows VSS EBS 스냅샷 2696
VSS란 무엇인가요? 2697
VSS 기반 Amazon EBS 스냅샷 솔루션의 작동 방식 2698
VSS 사전 조건 2699
VSS 스냅샷 생성 2711
VSS 스냅샷 문제 해결 2721
AWS VSS 솔루션의 복원 옵션 2725
버전 기록 2726
객체 스토리지, 파일 스토리지 및 파일 캐싱 2731
Amazon S3 2731
Amazon EFS 2733
Amazon FSx 2737
Amazon File Cache 2743
리소스 관리 2744
리소스에 대한 리전 선택 2744

리소스 찾기 2746
콘솔 단계 2746
명령줄 예제 2754
글로벌 보기(리전 간) 2757
Amazon EC2 Global View 2758
리소스 태깅 2761
태그 기본 사항 2761
리소스에 태그 지정 2762
태그 제한 2763
태그 및 액세스 관리 2764
결제를 위한 리소스 태깅 2764
리소스 태그 지정 권한 2765
태그 추가 및 제거 2768
태그로 리소스 필터링 2772
인스턴스 메타데이터를 사용하여 태그 보기 2774
Service quotas 2780
현재 할당량 보기 2780
증가 요청 2781
포트 25를 사용하여 전송되는 이메일 관련 제한 2782
리소스 모니터링 2783
인스턴스 상태 모니터링 2784
상태 확인 2785
상태 변경 이벤트 2793
예약된 이벤트 2795
CloudWatch를 사용하여 인스턴스 모니터링 2833
인스턴스 경보 2833
세부 모니터링 관리 2835
CloudWatch 지표 2838
CloudWatch 에이전트 설치 및 구성 2856
지표에 대한 통계 2861
모니터링 그래프 보기 2870
경보 만들기 2871
인스턴스를 중지, 종료, 재부팅 또는 복구하는 경보 만들기 2872
EventBridge를 사용하여 자동화 2884
Amazon EC2 이벤트 유형 2884
CloudTrail을 사용하여 API 호출 로깅 2885

CloudTrail의 Amazon EC2 API 관리 이벤트 2886
Amazon EC2 API 이벤트 예제 2887
EC2 Instance Connect를 사용한 연결 감사 2888
.NET 및 SQL Server 애플리케이션 모니터링 2890
프리 티어 사용량 추적 2891
문제 해결 2894
인스턴스 시작 문제 2894
잘못된 디바이스 이름 2895
인스턴스 제한 초과됨 2896
부족한 인스턴스 용량 2896
요청된 구성이 현재 지원되지 않습니다. 지원되는 구성은 설명서를 참조하세요. 2897
인스턴스 즉시 종료 2897
권한 부족 2899
Windows 시작 직후 높은 CPU 사용량(Windows 인스턴스만 해당) 2900
인스턴스 중지 문제 2900
인스턴스 강제 중지 2902
(선택 사항) 대체 인스턴스 생성 2903
인스턴스 종료 문제 2905
인스턴스 즉시 종료 2906
지연된 인스턴스 종료 2906
종료된 인스턴스가 계속 표시됨 2906
오류: 인스턴스가 종료되지 않을 수 있습니다. 'disableApiTermination' 인스턴스 특성 수정 . 2906
인스턴스가 자동으로 시작되거나 종료됨 2907
연결할 수 없는 인스턴스 2907
인스턴스 재부팅 2907
인스턴스 콘솔 출력 2908
연결할 수 없는 인스턴스의 스크린샷 캡처 2909
Windows 인스턴스에 대한 일반적인 스크린샷 2910
호스트 컴퓨터 실패 시 인스턴스 복구 2919
인스턴스가 오프라인으로 표시되고 예기치 않게 재부팅됨 2919
Linux 인스턴스 SSH 문제 2920
연결 문제의 일반적인 원인 2920
인스턴스 연결 중 오류 발생: 연결 시간 초과 2923
오류: 키를 로드할 수 없습니다... 예상: 모든 개인 키 2926
오류: 서버에서 사용자 키를 인식하지 못함 2926
오류: 사용 권한이 거부되었거나 [인스턴스] 포트 22에 의해 연결이 닫힘 2928

오류: 보호되지 않는 프라이빗 키 파일 2930
오류: 프라이빗 키는 '-----BEGIN RSA PRIVATE KEY-----'로 시작하고 '-----END RSA PRIVATE KEY-----'로 끝나야 합니다. 2932
오류: 호스트 키 확인 실패 2932
오류: 서버에서 키 거부 또는 지원되는 인증 방법이 없음 2933
인스턴스를 ping할 수 없음 2934
오류: 서버에서 예기치 않게 네트워크 연결을 차단함 2934
오류: EC2 Instance Connect에 대한 호스트 키 유효성 검사 실패 2934
EC2 Instance Connect를 사용하여 Ubuntu 인스턴스에 연결할 수 없음 2936
프라이빗 키를 분실했습니다. 인스턴스에 연결하려면 어떻게 해야 하나요? 2937
Linux 인스턴스 상태 확인 실패 2943
상태 점검 정보 검토 2944
시스템 로그 검색 2945
Linux 인스턴스의 시스템 로그 오류 문제 해결 2945
메모리 부족: 프로세스 중지 2947
ERROR: mmu_update failed(메모리 관리 업데이트 실패) 2948
I/O 오류(블록 디바이스 장애) 2949
I/O ERROR: neither local nor remote disk(분산된 블록 디바이스 손상) 2950
request_module: runaway loop modprobe(이전 Linux 버전에서 레거시 커널 modprobe 반복) 2951
"FATAL: kernel too old" 및 "fsck: No such file or directory while trying to open /dev"(커널과 AMI 불일치) 2952
"FATAL: Could not load /lib/modules" 또는 "BusyBox"(커널 모듈 누락) 2953
ERROR Invalid kernel(EC2 커널이 호환되지 않음) 2955
fsck: No such file or directory while trying to open... (파일 시스템을 찾을 수 없음) 2957
파일 시스템 마운트 관련 일반 오류(마운트 실패) 2959
VFS: Unable to mount root fs on unknown-block(루트 파일 시스템 불일치) 2961
Error: Unable to determine major/minor number of root device... (루트 파일 시스템/디바이스 불일치) 2962
XENBUS: Device with no driver... 2964
... days without being checked, check forced(파일 시스템 검사 필요) 2965
fsck died with exit status... (디바이스 누락) 2965
GRUB 프롬프트(grubdom>) 2967
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) 2970

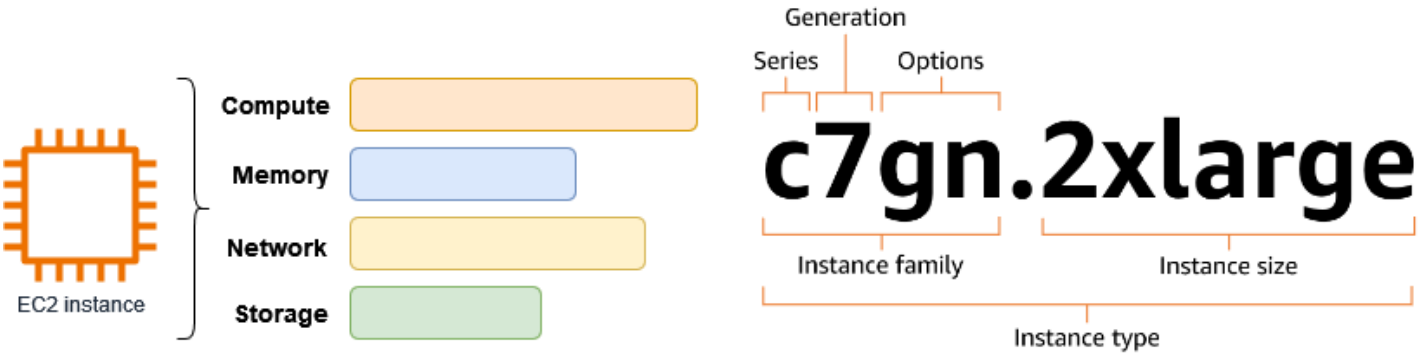
Unable to load SELinux Policy. Machine is in enforcing mode. Halting now(SELinux 정책을 가져올 수 없습니다. 시스템이 강제 실행 모드입니다. 중단됩니다). (잘못된 SELinux 구성) . 2971
XENBUS: Timeout connecting to devices(Xenbus 시간 초과) 2973
잘못된 볼륨에서 Linux 인스턴스 부팅 2974
Windows 인스턴스 RDP 문제 2975
원격 데스크톱으로 원격 컴퓨터에 연결할 수 없음 2976
macOS RDP 클라이언트 사용 중 오류 발생 2977
RDP에 바탕 화면이 아닌 빈 화면 표시 2978
관리자가 아닌 사용자로 인스턴스에 원격 로그온할 수 없음 2978
AWS Systems Manager를 사용하여 원격 데스크톱 연결 문제 해결 2978
원격 레지스트리를 사용하여 EC2 인스턴스에서 원격 데스크톱 활성화 2982
프라이빗 키를 분실했습니다. 내 Windows 인스턴스에 연결하려면 어떻게 해야 하나요? 2984
Windows 인스턴스 시작 문제 2984
"Password is not available" 2985
"Password not available yet" 2986
"Cannot retrieve Windows password" 2986
"Waiting for the metadata service" 2986
"Unable to activate Windows" 2991
"Windows is not genuine (0x80070005)" 2993
"No Terminal Server License Servers available to provide a license" 2993
"일부 설정이 사용자의 조직에 의해 관리됩니다.(Some settings are managed by your organization.)" 2993
Windows 인스턴스 문제 2994
AWS Systems Manager Sessions Manager를 Windows Server 2025 인스턴스에 연결할 수 없음 2995
EBS 볼륨이 Windows Server 2016 및 2019에서 초기화를 수행하지 않음 2995
DSRM(Directory Services Restore Mode)로 EC2 Windows 인스턴스 부팅 2996
인스턴스의 네트워크 연결이 끊어지거나 예약된 작업이 예정 시간에 실행되지 않음 2999
콘솔 출력을 가져올 수 없음 2999
네트워크에서 Windows Server 2012 R2를 이용할 수 없는 경우 3000
디스크 서명 충돌 3000
Windows 관리자 암호 재설정 3002
EC2Launch v2를 사용하여 암호 재설정 3003
EC2Launch를 사용하여 암호 재설정 3008
EC2Config를 사용하여 암호 재설정 3013
Sysprep 문제 해결 3018

Linux용 EC2Rescue 인스턴스 3020
EC2Rescue 설치 3021
EC2Rescue 명령 실행 3025
EC2Rescue 모듈 개발 3027
Windows 인스턴스용 EC2Rescue 3033
EC2Rescue GUI를 사용한 문제 해결 3035
EC2Rescue CLI를 사용한 문제 해결 3041
EC2Rescue 및 Systems Manager를 사용한 문제 해결 3049
EC2 직렬 콘솔 3052
사전 조건 3053
EC2 직렬 콘솔에 대한 액세스 구성 3060
EC2 직렬 콘솔에 연결 3070
EC2 직렬 콘솔 연결 해제 3080
EC2 직렬 콘솔을 사용하여 인스턴스 문제 해결 3080
진단 인터럽트 보내기 3089
지원되는 인스턴스 유형 3090
사전 조건 3090
진단 인터럽트 보내기 3093
문서 기록 3094
2018년 및 그 이전 기록 3122

Amazon EC2란 무엇인가요?

Amazon Elastic Compute Cloud(Amazon EC2)는 Amazon Web Services(AWS) 클라우드에서 온디맨드 확장 가능 컴퓨팅 용량을 제공합니다. Amazon EC2를 사용하면 하드웨어 비용이 절감되므로 애플리케이션을 더욱 빠르게 개발하고 배포할 수 있습니다. Amazon EC2를 사용하여 원하는 수의 가상 서버를 빌드하고 보안 및 네트워킹을 구성하며 스토리지를 관리할 수 있습니다. 용량을 추가(스케일 업)하여 월간 또는 연간 프로세스 또는 웹 사이트 트래픽 급증 등 컴퓨팅 사용량이 많은 작업을 처리할 수 있습니다. 사용량이 감소하면 용량을 다시 축소(스케일 다운)할 수 있습니다.

EC2 인스턴스는 AWS 클라우드의 가상 서버입니다. EC2 인스턴스를 시작할 때 지정하는 인스턴스 유형에 따라 인스턴스에 사용할 수 있는 하드웨어가 결정됩니다. 인스턴스 유형마다 서로 다른 컴퓨팅, 메모리, 네트워크 및 스토리지 리소스의 균형을 제공합니다. 자세한 내용은 Amazon EC2 인스턴스 유형 안내서를 참조하세요.



Amazon EC2의 기능

Amazon EC2는 다음의 대략적인 기능을 제공합니다.

인스턴스

가상 서버.

Amazon Machine Images (AMIs)

서버에 필요한 구성 요소(운영 체제와 추가 소프트웨어 포함)를 패키징하는 인스턴스용 사전 구성 템플릿.

인스턴스 타입

인스턴스의 다양한 CPU, 메모리, 스토리지, 네트워킹 용량 및 그래픽 하드웨어 구성.

Amazon EBS 볼륨

Amazon Elastic Block Store(Amazon EBS)를 사용하는 데이터에 대한 영구 스토리지 볼륨.

인스턴스 스토어 볼륨

인스턴스를 중단, 최대 절전 모드로 전환 또는 종료할 때 삭제되는 임시 데이터용 스토리지 볼륨.

키 페어

인스턴스에 대한 보안 로그인 정보. AWS는 퍼블릭 키를 저장하고 사용자는 프라이빗 키를 안전한 장소에 저장합니다.

보안 그룹

인스턴스에 도달할 수 있는 프로토콜, 포트 및 소스 IP 범위와 인스턴스가 연결할 수 있는 대상 IP 범위를 지정할 수 있는 가상 방화벽.

Amazon EC2는 전자 상거래 웹사이트 운영자 또는 서비스 공급자에 의한 신용카드 데이터의 처리, 저장 및 전송을 지원하며, 지불 카드(PCI) 보안 표준(DSS)을 준수하는 것으로 검증되었습니다. AWS PCI 규정 준수 패키지의 사본을 요청하는 방법 등 PCI DSS에 대해 자세히 알아보려면 PCI DSS 레벨 1을 참조하세요.

관련 서비스

Amazon EC2와 함께 사용할 서비스

Amazon EC2를 사용하여 배포한 인스턴스와 함께 다른 AWS 서비스를 사용할 수 있습니다.

Amazon EC2 Auto Scaling

애플리케이션의 로드를 처리할 수 있는 정확한 수의 Amazon EC2 인스턴스를 유지하는 데 도움이 됩니다.

AWS Backup

Amazon EC2 인스턴스와 여기에 연결된 Amazon EBS 볼륨의 백업을 자동화합니다.

Amazon CloudWatch

인스턴스와 Amazon EBS 볼륨을 모니터링합니다.

Elastic Load Balancing

수신되는 애플리케이션 트래픽을 여러 인스턴스로 자동 분산합니다.

Amazon GuardDuty

EC2 인스턴스에 대한 잠재적 무단 사용 또는 악의적 사용을 탐지합니다.

EC2 Image Builder

사용자 지정되어 안전하고 최신 상태인 서버 이미지의 생성, 관리 및 배포를 자동화합니다.

AWS Launch Wizard

개별 AWS 리소스를 수동으로 식별하고 프로비저닝할 필요 없이 타사 애플리케이션에 대한 AWS 리소스를 크기 조정, 구성 및 배포합니다.

AWS Systems Manager

이 안전한 엔드 투 엔드 관리 솔루션을 사용하여 EC2 인스턴스에서 대규모로 작업을 수행할 수 있습니다.

추가 컴퓨팅 서비스

Amazon EC2를 사용하는 대신 다른 AWS 컴퓨팅 서비스를 사용하여 인스턴스를 시작할 수 있습니다.

Amazon Lightsail

저렴하고 예측 가능한 월별 요금으로 프로젝트를 신속하게 배포하는 데 필요한 리소스를 제공하는 클라우드 플랫폼인 Amazon Lightsail을 사용하여 웹 사이트 또는 웹 애플리케이션을 구축합니다. Amazon EC2와 Lightsail을 비교하려면 Amazon Lightsail 또는 Amazon EC2 섹션을 참조하세요.

Amazon Elastic Container Service(Amazon ECS)

EC2 인스턴스 클러스터에서 컨테이너화된 애플리케이션을 배포, 관리하고 규모를 조정합니다. 자세한 내용은 AWS 컨테이너 서비스 선택을 참조하세요.

Amazon Elastic Kubernetes Service(Amazon EKS)

AWS에서 Kubernetes 애플리케이션을 실행합니다. 자세한 내용은 AWS 컨테이너 서비스 선택을 참조하세요.

Amazon EC2 액세스

다음 인터페이스를 사용하여 Amazon EC2 인스턴스를 생성하고 관리할 수 있습니다.

Amazon EC2 콘솔

Amazon EC2 인스턴스 및 리소스를 생성하고 관리하는 간단한 웹 인터페이스입니다. AWS 계정에 가입한 고객은 AWS Management Console에 로그인한 후 콘솔 홈페이지에서 EC2를 선택하여 Amazon EC2에 액세스할 수 있습니다.

AWS Command Line Interface

명령줄 쉘의 명령을 사용하여 AWS 서비스와 상호 작용할 수 있습니다. Windows, Mac, Linux에서 지원됩니다. AWS CLI에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서를 참조하세요. AWS CLI 명령 참조에서 Amazon EC2 명령을 찾아볼 수 있습니다.

AWS CloudFormation

Amazon EC2는 AWS CloudFormation을 사용한 리소스 생성을 지원합니다. AWS 리소스를 설명하는 템플릿(JSON 또는 YAML 형식)을 생성하면 AWS CloudFormation에서 이러한 리소스를 프로비저닝하고 구성합니다. CloudFormation 템플릿을 재사용하여 동일한 리전 및 계정 또는 여러 리전 및 계정에 동일한 리소스를 여러 번 프로비저닝할 수 있습니다. Amazon EC2에서 지원되는 리소스 유형 및 속성에 대한 자세한 내용은 AWS CloudFormation 사용 설명서에서 EC2 리소스 유형 참조를 참조하세요.

AWS SDK

HTTP나 HTTPS 요청을 직접 보내는 대신, 각 언어가 제공하는 고유의 API를 사용하여 애플리케이션을 빌드하는 것을 선호하는 개발자를 위해 AWS는, 라이브러리, 샘플 코드, 자습서 및 기타 리소스를 제공합니다. 이러한 라이브러리는 요청 암호화 서명, 요청 재시도, 오류 응답 처리와 같은 작업을 자동화하는 기본 기능을 제공하므로 더 쉽게 시작할 수 있습니다. 자세한 내용은 AWS 에서의 구축을 위한 도구를 참조하세요.

AWS Tools for PowerShell

SDK for .NET에서 공개하는 기능을 기반으로 하는 PowerShell 모듈 세트입니다. PowerShell용 도구를 사용하면 PowerShell 명령줄에서 AWS 리소스에 대한 작업을 스크립팅할 수 있습니다. 시작하려면 AWS Tools for Windows PowerShell 사용 설명서를 참조하십시오. Amazon EC2용 cmdlet은 AWS Tools for PowerShell cmdlet 참조에서 찾아볼 수 있습니다.

Query API

Amazon EC2에서는 쿼리 API를 제공합니다. 이러한 요청은 HTTP나 HTTPS의 메시지 교환 방식인 GET이나 POST이며, 미리 정해진 이름인 "`Action`"을 쿼리 변수로 사용합니다. Amazon EC2에 관련된 API 작업에 대한 자세한 내용은 Actions(Amazon EC2 API Reference)을 참조하십시오.

Amazon EC2 요금

Amazon EC2는 다음과 같은 요금 옵션을 제공합니다.

프리 티어

Amazon EC2를 무료로 시작할 수 있습니다. 프리 티어 옵션을 살펴보려면 AWS 프리 티어를 참조하세요.

온디맨드 인스턴스

장기 약정이나 선결제 없이 초 단위로, 최소 60초 사용한 인스턴스에 대한 요금을 지불하는 방식입니다.

절감형 플랜

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.

전용 호스트

온디맨드 또는 절감형 플랜의 일부로 고객 전용 물리적 EC2 서버를 사용하여 비용을 절감합니다. 기존 서버 기반 소프트웨어 라이선스를 사용하면 규정 준수 요구 사항을 충족하는 데 도움을 받을 수 있습니다.

온디맨드 용량 예약

원하는 기간 동안 특정 가용 영역의 EC2 인스턴스에 대해 용량을 예약합니다.

초당 청구

청구서에서 사용하지 않은 분 및 초 단위의 비용을 제거합니다.

Amazon EC2에 관련된 전체적인 요금 및 가격 목록과 구매 모델 관련 추가 정보는 Amazon EC2 요금을 참조하세요.

추정, 결제 및 비용 최적화

AWS 사용 사례에 대한 예상 비용을 계산하려면 AWS Pricing Calculator를 사용하세요.

Microsoft 워크로드를 AWS에 배포된 오픈 소스 및 클라우드 네이티브 서비스를 사용하는 최신 아키텍처로 변환하는 데 드는 비용을 추정하려면 AWS Modernization Calculator for Microsoft Workloads를 사용하세요.

청구 요금은 AWS 결제 및 비용 관리 콘솔의 청구 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 결제 및 비용 관리 사용 설명서를 참조하세요.

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

샘플 프로비저닝된 환경의 비용을 계산하려면 클라우드 경제 센터를 참조하세요. 프로비저닝된 환경의 비용을 계산할 때 EBS 볼륨에 대한 스냅샷 스토리지와 같은 부수적인 비용을 포함해야 합니다.

AWS Trusted Advisor 사용을 통해 AWS 환경의 비용, 보안 및 성능을 최적화할 수 있습니다.

AWS Cost Explorer를 사용하면 EC2 인스턴스의 비용과 사용량을 분석할 수 있습니다. 지난 13개월까지의 데이터를 볼 수 있고 향후 12개월 동안의 지출을 예상할 수 있습니다. 자세한 내용은 AWS Cost Management 사용 설명서의 AWS Cost Explorer를 사용한 비용 및 사용량 분석을 참조하세요.

리소스

- Amazon EC2의 기능
- AWS re:Post
- AWS Skill Builder
- AWS Support
- 실습 자습서
- 웹 호스팅
- AWS 기반 Windows

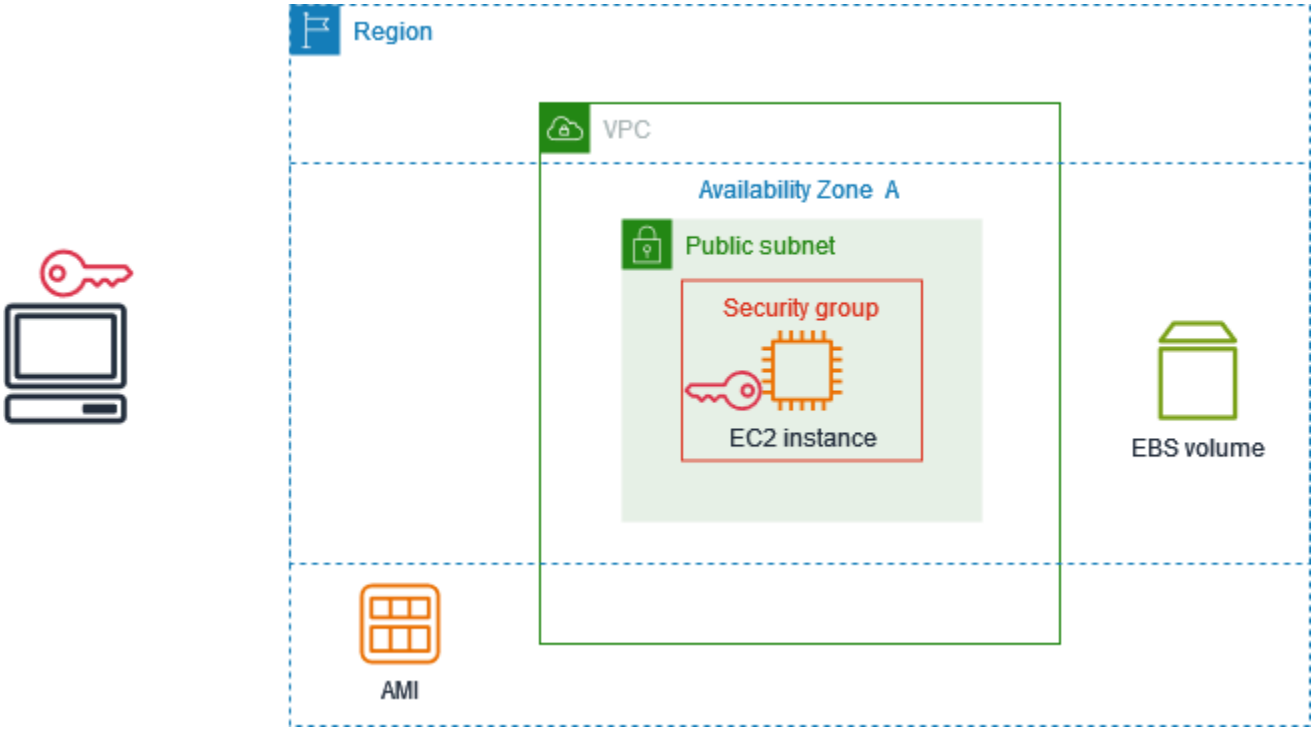
Amazon EC2 시작하기

이 자습서를 사용하여 Amazon Elastic Compute Cloud(Amazon EC2)를 시작하세요. EC2 인스턴스를 시작하고 연결하는 방법을 알아보게 됩니다. 인스턴스는 AWS 클라우드의 가상 서버입니다. Amazon EC2를 사용하여 인스턴스에서 실행되는 운영 체제와 애플리케이션을 설정하고 구성할 수 있습니다.

개요

다음 다이어그램은 이 자습서에서 사용할 주요 구성 요소를 보여줍니다.

- 이미지 - 인스턴스에서 실행할 소프트웨어(예: 운영 체제)가 포함된 템플릿입니다.
- 키 페어 - 인스턴스에 연결할 때 ID를 증명하는 데 사용하는 보안 자격 증명 세트입니다. 퍼블릭 키는 인스턴스에 있고 프라이빗 키는 컴퓨터에 있습니다.
- 네트워크 - Virtual Private Cloud(VPC)는 AWS 계정 전용 가상 네트워크입니다. 빠르게 시작할 수 있도록 계정에는 각 AWS 리전에 기본 VPC가 제공되며 각 기본 VPC에는 각 가용 영역에 기본 서브넷이 있습니다.
- 보안 그룹 - 인바운드 및 아웃바운드 트래픽을 제어하는 가상 방화벽 역할을 합니다.
- EBS 볼륨 - 이미지의 루트 볼륨이 필요합니다. 필요에 따라 데이터 볼륨을 추가할 수 있습니다.



이 자습서의 비용

AWS에 가입하면 AWS 프리 티어를 사용하여 Amazon EC2를 시작할 수 있습니다. AWS 계정을 생성한 지 12개월 미만이고 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.

프리 티어 이용 자격 여부를 결정하는 지침은 the section called “프리 티어 사용량 추적” 섹션을 참조하세요.

업무

- 1단계: 인스턴스 시작
- 2단계: 인스턴스에 연결
- 3단계: 인스턴스 정리
- 다음 단계

1단계: 인스턴스 시작

다음 절차의 설명에 따라 AWS Management Console을 사용하여 EC2 인스턴스를 시작할 수 있습니다. 이 자습서는 첫 번째 인스턴스를 빠르게 시작하도록 돕기 위한 것이므로 가능한 모든 옵션을 다루지는 않습니다.

인스턴스 시작

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 화면 상단의 탐색 모음에 현재 AWS 리전(예: 오하이오주)이 표시됩니다. 선택된 리전을 사용하거나 선택적으로 가까운 리전을 선택할 수 있습니다.
3. EC2 콘솔 대시보드의 시작 인스턴스 창에서 인스턴스 시작을 선택합니다.
4. 이름 및 태그(Name and tags) 아래의 이름(Name)에 인스턴스를 설명하는 이름을 입력합니다.
5. 애플리케이션 및 OS 이미지(Amazon Machine Image)(Application and OS Images (Amazon Machine Image))에서 다음을 수행합니다.
   a. 빠른 시작을 선택한 다음 인스턴스의 운영 체제(OS)를 선택합니다. 첫 번째 Linux 인스턴스의 경우 Amazon Linux를 선택하는 것이 좋습니다.
   b. Amazon Machine Image(AMI)에서 프리 티어 사용 가능으로 표시된 AMI를 선택합니다.
6. 인스턴스 유형에서 인스턴스 유형으로 프리 티어에 적합한 `t2.micro`를 선택합니다. `t2.micro`를 사용할 수 없는 리전에서는 `t3.micro`가 프리 티어에 적합합니다.

7. 키 페어(로그인)의 키 페어 이름에서 기존 키 페어를 선택하거나 새 키 페어 생성을 선택하여 첫 번째 키 페어를 생성합니다.

   > **Warning**
   >
   > 키 페어 없이 진행(권장되지 않음)을 선택하면 이 자습서에 설명된 방법을 사용하여 인스턴스에 연결할 수 없습니다.

8. 네트워크 설정에서 기본 VPC를 선택하고, 선택한 가용 영역에서 기본 서브넷을 사용하는 옵션을 선택하고, 어디서나(`0.0.0.0/0`) 인스턴스에 연결할 수 있는 규칙으로 보안 그룹을 구성한 것을 확인합니다.

   > **Warning**
   >
   > `0.0.0.0/0`을 지정하면 전 세계의 모든 IP 주소에서 트래픽을 사용하도록 설정됩니다. SSH 및 RDP 프로토콜의 경우, 테스트 환경에서 잠시 사용하는 것은 괜찮지만 프로덕션 환경에서는 안전하지 않습니다. 프로덕션 환경에서는 적절한 개별 IP 주소 또는 주소 범위에서만 액세스 권한을 부여해야 합니다.

   첫 번째 인스턴스의 경우에는 기본 설정을 사용하는 것이 좋습니다. 그렇지 않은 경우 다음과 같이 네트워크 설정을 업데이트할 수 있습니다:

   - (선택 사항) 특정 기본 서브넷을 사용하려면 편집을 선택한 다음 서브넷을 선택합니다.
   - (선택 사항) 다른 VPC를 사용하려면 편집을 선택한 다음 기존 VPC를 선택합니다. VPC가 퍼블릭 인터넷 액세스를 위해 구성되지 않은 경우에는 인스턴스에 연결할 수 없습니다.
   - (선택 사항) 특정 네트워크로 인바운드 연결 트래픽을 제한하려면 모든 위치 대신 사용자 지정을 선택하고 네트워크에 대한 CIDR 블록을 입력합니다.
   - (선택 사항) 다른 보안 그룹을 사용하려면 기존 보안 그룹 선택을 선택하고 기존 보안 그룹을 선택합니다. 보안 그룹에 네트워크의 연결 트래픽을 허용하는 규칙이 없는 경우 인스턴스에 연결할 수 없습니다. Linux 인스턴스의 경우 SSH 트래픽을 허용해야 합니다. Windows 인스턴스의 경우 RDP 트래픽을 허용해야 합니다.

9. 스토리지 구성에서 루트 볼륨은 구성했지만 데이터 볼륨은 구성하지 않은 것을 확인할 수 있습니다. 테스트 목적으로는 이 정도면 충분합니다.
10. 요약 패널에서 인스턴스 구성 요약을 검토하고 준비가 되면 인스턴스 시작을 선택합니다.
11. 실행이 성공하면 성공 알림에서 인스턴스의 ID를 선택하여 인스턴스 페이지를 열고 시작 상태를 모니터링합니다.

12. 인스턴스에 대한 확인란을 선택합니다. 초기 인스턴스 상태는 pending입니다. 인스턴스가 시작되면 상태가 running로 변경됩니다. 상태 및 경보 탭을 선택합니다. 인스턴스가 상태 검사를 통과하면 연결 요청을 받을 준비가 된 것입니다.

2단계: 인스턴스에 연결

사용하는 절차는 인스턴스의 운영 체제에 따라 달라집니다. 인스턴스에 연결할 수 없는 경우 지원이 필요하면 Amazon EC2 Linux 인스턴스 연결 문제 해결을 참조하세요.

Linux 인스턴스

모든 SSH 클라이언트를 사용하여 Linux 인스턴스에 연결할 수 있습니다. 컴퓨터에서 Windows를 실행 중인 경우 터미널을 열고 ssh 명령을 실행하여 SSH 클라이언트가 설치되어 있는지 확인합니다. 명령을 찾을 수 없으면 Windows용 OpenSSH를 설치하세요.

SSH를 사용하여 인스턴스에 연결하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 찾은 인스턴스를 선택한 다음 [Connect]를 선택합니다.
4. 인스턴스에 연결 페이지에서 SSH 클라이언트 탭을 선택합니다.
5. (선택 사항) 인스턴스를 시작할 때 키 페어를 만들고 Linux 또는 macOS를 실행하는 컴퓨터에 프라이빗 키(.pem 파일)를 다운로드한 경우 예제 chmod 명령을 실행하여 프라이빗 키에 대한 권한을 설정합니다.
6. 예제 SSH 명령을 복사합니다. 다음은 예제입니다. 여기서 *key-pair-name*.pem은 프라이빗 키 파일의 이름, *ec2-user*는 이미지와 연결된 사용자 이름, @ 기호 뒤의 문자열은 인스턴스의 퍼블릭 DNS 이름입니다.

   ```
   ssh -i key-pair-name.pem ec2-user@ec2-198-51-100-1.us-east-2.compute.amazonaws.com
   ```

7. 컴퓨터의 터미널 창에서 이전 단계에서 저장한 ssh 명령을 실행합니다. 현재 디렉터리에 프라이빗 키 파일이 없는 경우 이 명령에 키 파일의 정규화된 경로를 지정해야 합니다.

   다음은 응답의 예입니다.

   ```
   The authenticity of host 'ec2-198-51-100-1.us-east-2.compute.amazonaws.com
    (198-51-100-1)' can't be established.
   ECDSA key fingerprint is l4UB/neBad9tvkgJf1QZWxheQmR59WgrgzEimCG6kZY.
   ```

```
Are you sure you want to continue connecting (yes/no)?
```

8. (선택 사항) 보안 알림의 지문이 인스턴스를 처음 시작할 때 콘솔 출력에 포함된 인스턴스 지문과 일치하는지 확인합니다. 콘솔 출력을 가져오려면 작업, 모니터링 및 문제 해결, 시스템 로그 가져오기를 선택합니다. 이들 지문이 일치하지 않으면 누군가가 메시지 가로채기(man-in-the-middle) 공격을 시도하고 있는 것일 수 있습니다. 이들 지문이 일치하면 다음 단계를 계속 진행합니다.
9. **yes**를 입력합니다.

   다음은 응답의 예입니다.

```
Warning: Permanently added 'ec2-198-51-100-1.us-
east-2.compute.amazonaws.com' (ECDSA) to the list of known hosts.
```

Windows 인스턴스

RDP를 사용하여 Windows 인스턴스에 연결하려면 최초 관리자 암호를 검색한 다음 인스턴스에 연결할 때 이 암호를 입력해야 합니다. 인스턴스를 시작한 후 암호를 사용할 수 있으려면 몇 분 정도 걸립니다. 계정에 GetPasswordData 작업을 호출할 권한이 있어야 합니다. 자세한 내용은 Amazon EC2 API에 대한 액세스를 제어하는 정책 예제 섹션을 참조하세요.

관리자 계정의 기본 사용자 이름은 AMI에 포함된 운영 체제(OS)의 언어에 따라 다릅니다. 올바른 사용자 이름을 확인하려면 운영 체제의 언어를 확인한 다음 해당 사용자 이름을 선택합니다. 예를 들어, 영어 OS의 경우 사용자 이름은 `Administrator`이고, 프랑스 OS의 경우 사용자 이름은 `Administrateur`이며, 포르투갈어 OS의 경우 사용자 이름은 `Administrador`입니다. OS의 언어 버전에 해당 언어의 사용자 이름이 없는 경우 사용자 이름 `Administrator (Other)`를 선택합니다. 자세한 내용은 Microsoft 웹 사이트의 Localized Names for Administrator Account in Windows를 참조하세요.

최초 관리자 암호 검색

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 찾은 인스턴스를 선택한 다음 [Connect]를 선택합니다.
4. 인스턴스에 연결 페이지에서 RDP 클라이언트 탭을 선택합니다.
5. 관리자 계정의 기본 사용자 이름을 사용자 이름으로 선택합니다. 선택한 사용자 이름은 인스턴스를 시작하는 데 사용한 AMI에 포함된 운영 체제(OS)의 언어와 일치해야 합니다. 운영 체제와 동일한 언어의 사용자 이름이 없는 경우 관리자(기타)를 선택합니다.

6. 암호 가져오기를 선택합니다.
7. Windows 암호 가져오기 페이지에서 다음을 수행하세요.

   a. 프라이빗 키 파일 업로드를 선택하고 인스턴스를 시작할 때 지정한 프라이빗 키(`.pem`) 파일로 이동합니다. 파일을 선택하고 [열기(Open)]를 클릭하여 파일의 전체 콘텐츠를 이 창에 복사합니다.

   b. 암호 해독을 선택합니다. Windows 암호 가져오기 페이지가 닫히고 이전에 표시된 암호 가져오기 링크를 대체하는 암호 아래에 인스턴스의 기본 관리자 암호가 표시됩니다.

   c. 암호를 복사하여 안전한 장소에 저장합니다. 이 암호는 인스턴스에 연결하는 데 필요합니다.

다음 절차에서는 Windows용 원격 데스크톱 연결 클라이언트(MSTSC)를 사용합니다. 다른 RDP 클라이언트를 사용하는 경우 RDP 파일을 다운로드한 다음 RDP 클라이언트 설명서에서 RDP 연결 설정 단계를 참조하세요.

RDP 클라이언트를 사용하여 Windows 인스턴스에 연결

1. 인스턴스에 연결 페이지에서 원격 데스크톱 파일 다운로드를 선택합니다. 파일 다운로드가 완료되면 취소를 선택하여 인스턴스 페이지로 돌아갑니다. RDP 파일이 `Downloads` 폴더에 다운로드됩니다.
2. `mstsc.exe`를 실행하여 RDP 클라이언트를 엽니다.
3. 옵션 표시를 확장하고 열기를 선택한 다음 `Downloads` 폴더에서.rdp 파일을 선택합니다.
4. 기본적으로 컴퓨터는 인스턴스의 퍼블릭 IPv4 DNS 이름이고 사용자 이름은 관리자 계정입니다. 대신 IPv6를 사용하여 인스턴스에 연결하려면 인스턴스의 퍼블릭 IPv4 DNS 이름을 해당 IPv6 주소로 바꿉니다. 기본 설정을 검토하고 필요에 따라 변경합니다.
5. 연결을 선택합니다. 원격 연결 게시자를 알 수 없다는 경고가 표시되면 연결을 선택하여 계속합니다.
6. 앞서 저장한 암호를 입력한 후 확인을 선택합니다.
7. 자체 서명된 인증서의 특성으로 인해, 보안 인증서를 인증할 수 없다는 경고 메시지가 나타날 수도 있습니다. 다음 중 하나를 수행합니다.
   - 인증서를 신뢰하는 경우 예를 선택하여 인스턴스에 연결합니다.
   - [Windows] 계속하기 전에 인증서의 지문을 시스템 로그의 값과 비교하여 원격 컴퓨터의 ID를 확인합니다. 인증서 보기를 선택한 다음 세부 정보 탭에서 지문을 선택합니다. 이 값을 작업, 모니터링 및 문제 해결, 시스템 로그 가져오기의 `RDPCERTIFICATE-THUMBPRINT` 값과 비교합니다.

- [Mac OS X] 계속하기 전에 인증서의 지문을 시스템 로그의 값과 비교하여 원격 컴퓨터의 ID를 확인합니다. 인증서 보기를 선택하고 세부 정보를 확장한 다음 SHA1 지문을 선택합니다. 이 값을 작업, 모니터링 및 문제 해결, 시스템 로그 가져오기의 `RDPCERTIFICATE-THUMBPRINT` 값과 비교합니다.

8. RDP 연결에 성공하면 RDP 클라이언트는 Windows 로그인 화면을 표시한 다음 Windows 데스크톱을 표시합니다. 오류 메시지가 표시되면 the section called “원격 데스크톱으로 원격 컴퓨터에 연결할 수 없음”의 내용을 참조하세요. RDP 연결을 완료하면 RDP 클라이언트를 닫을 수 있습니다.

3단계: 인스턴스 정리

이 자습서용으로 생성한 인스턴스와 볼륨을 완료한 후에는 인스턴스를 종료하여 정리해야 합니다. 정리하기 전에 이 인스턴스로 추가 연습을 수행하려는 경우 다음 단계 섹션을 참조하세요.

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

해당 인스턴스 또는 프리 티어 한도에 포함되는 사용량에 대한 요금 발생은 인스턴스 상태가 `shutting down` 또는 `terminated`로 변경되는 즉시 발생이 중지됩니다. 나중에 사용하기 위해 인스턴스를 유지하되 요금 또는 프리 티어 한도에 포함되는 사용량이 발생하지 않도록 하려면 지금 인스턴스를 중지한 다음 나중에 다시 시작하면 됩니다. 자세한 내용은 Amazon EC2 인스턴스 중지 및 시작 섹션을 참조하세요.

인스턴스를 종료하려면

1. 탐색 창에서 인스턴스를 선택합니다. 인스턴스 목록에서 인스턴스를 선택합니다.
2. 인스턴스 상태, 인스턴스 종료(삭제)를 차례로 선택합니다.
3. 확인 메시지가 나타나면 종료(삭제)를 선택합니다.

   Amazon EC2가 인스턴스를 종료합니다. 인스턴스는 종료한 후에도 잠시 동안 콘솔에서 표시된 상태로 유지되며, 그 이후 항목이 자동으로 삭제됩니다. 종료된 인스턴스는 콘솔 디스플레이에서 직접 제거할 수 없습니다.

다음 단계

인스턴스를 시작한 후에는 다음 단계를 살펴볼 수 있습니다.

- 입문자용 자습서로 Amazon EC2 핵심 개념을 살펴봅니다. 자세한 내용은 EC2 인스턴스 시작을 위한 자습서 섹션을 참조하세요.
- 콘솔을 사용하여 Amazon EC2 프리 티어 사용량을 추적하는 방법을 알아봅니다. 자세한 내용은 the section called “프리 티어 사용량 추적” 섹션을 참조하세요.
- 사용량이 프리 티어 한도를 초과하는 경우 알려 주는 CloudWatch 경보를 구성합니다. 자세한 내용은 AWS Billing 사용 설명서에서 AWS 프리 티어 사용량 추적을 참조하세요.
- EBS 볼륨을 추가합니다. 자세한 내용은 Amazon EBS 사용 설명서의 Create an Amazon EBS volume을 참조하세요.
- Run 명령을 사용하여 EC2 인스턴스를 원격으로 관리하는 방법을 알아봅니다. 자세한 내용은 AWS Systems Manager 사용 설명서에서 AWS Systems Manager Run Command를 참조하십시오.
- 인스턴스 구매 옵션에 대해 알아봅니다. 자세한 내용은 Amazon EC2 결제 및 구매 옵션 섹션을 참조하세요.
- 인스턴스 유형에 대한 참고 정보를 얻습니다. 자세한 내용은 EC2 인스턴스 유형 파인더에서 권장 사항 받기 섹션을 참조하세요.

Amazon EC2 모범 사례

Amazon EC2를 최대한 활용하려면 다음과 같은 모범 사례를 수행하는 것이 좋습니다.

보안

- 가능한 경우 ID 제공업체 및 IAM 역할과 아이덴티티 페더레이션을 사용하여 AWS 리소스와 API에 대한 액세스를 관리합니다. 자세한 내용은 IAM 사용 설명서에서 IAM 정책 생성을 참조하세요.
- 보안 그룹에 대한 최소 허용 규칙을 구현합니다.
- 인스턴스에서 운영 체제와 애플리케이션을 정기적으로 패치, 업데이트 및 보안합니다. 자세한 내용은 업데이트 관리 단원을 참조하십시오. Windows 운영 체제와 관련된 지침은 Windows 인스턴스를 위한 보안 모범 사례 섹션을 참조하세요.
- Amazon Inspector를 사용하여 Amazon EC2 인스턴스에서 소프트웨어 취약성과 의도하지 않은 네트워크 노출을 자동으로 검색하고 스캔합니다. 자세한 내용은 Amazon Inspector 사용 설명서를 참조하세요.
- AWS Security Hub 제어를 사용하여 보안 모범 사례 및 보안 표준에 따라 Amazon EC2 리소스를 모니터링하세요. Security Hub 사용에 대한 자세한 내용은 AWS Security Hub 사용 설명서의 Amazon Elastic Compute Cloud 제어를 참조하세요.

스토리지

- 루트 디바이스 유형이 데이터 지속성, 백업 및 복구에 미치는 영향을 이해합니다. 자세한 내용은 루트 디바이스 유형 섹션을 참조하세요.
- 운영 체제와 데이터에 대해 별도의 Amazon EBS 볼륨을 사용합니다. 데이터를 포함하는 볼륨이 인스턴스 종료 이후에 지속되는지 확인합니다. 자세한 내용은 인스턴스가 종료될 때 데이터 보존 섹션을 참조하세요.
- 인스턴스에서 임시 데이터를 저장하는 데 사용 가능한 인스턴스 스토어를 사용합니다. 인스턴스를 중단하거나 최대 절전 모드로 전환하거나 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 사용 설명서의 Amazon EBS encryption을 참조하세요.

리소스 관리

- 인스턴스 메타데이터 및 사용자 지정 리소스 태그를 사용하여 AWS 리소스를 추적하고 식별합니다. 자세한 내용은 인스턴스 메타데이터를 사용하여 EC2 인스턴스를 관리합니다. 및 Amazon EC2 리소스 태깅 섹션을 참조하세요.
- Amazon EC2에 대한 현재 제한을 조회합니다. 실제로 필요할 시점보다 미리 제한 증가를 요청하도록 계획하세요. 자세한 내용은 Amazon EC2 서비스 할당량 섹션을 참조하세요.
- AWS Trusted Advisor를 사용하여 AWS 환경을 검사한 다음에 비용 절감, 시스템 가용성 및 성능 개선 또는 보안 격차를 해결할 기회가 있으면 권장 사항을 적용합니다. 자세한 내용은 AWS Support 사용 설명서의 AWS Trusted Advisor를 참조하세요.

백업 및 복구

- Amazon EBS 스냅샷을 사용하여 EBS 볼륨을 정기적으로 백업하고, 인스턴스에서 Amazon Machine Image(AMI)를 만들어 추후 인스턴스 시작을 위한 템플릿으로 구성을 저장합니다. 이 사용 사례를 달성하는 데 도움이 되는 AWS 서비스에 대한 자세한 내용은 AWS Backup 및 Amazon Data Lifecycle Manager를 참조하세요.
- 애플리케이션의 주요 구성 요소를 여러 가용 영역에 배포하고 데이터를 적절히 복제합니다.
- 인스턴스를 다시 시작할 때 IP 주소를 동적으로 지정하도록 애플리케이션을 설계합니다. 자세한 내용은 Amazon EC2 인스턴스 IP 주소 지정 섹션을 참조하세요.
- 이벤트를 모니터링하고 이에 대응하세요. 자세한 내용은 Amazon EC2 리소스 모니터링 섹션을 참조하세요.
- 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 섹션을 참조하세요. 자동 솔루션의 경우 Amazon EC2 Auto Scaling을 사용할 수 있습니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서를 참조하세요.
- 인스턴스 및 Amazon EBS 볼륨 복구 프로세스를 정기적으로 테스트하여 데이터와 서비스가 복원되는지 확인하세요.

네트워킹

- 애플리케이션의 TTL(Time-to-Live) 값을 IPv4 및 IPv6의 경우 255로 설정합니다. 더 작은 값을 사용하면 애플리케이션 트래픽이 전송되는 동안 TTL이 만료되어 인스턴스에 연결성 문제가 발생할 수 있습니다.

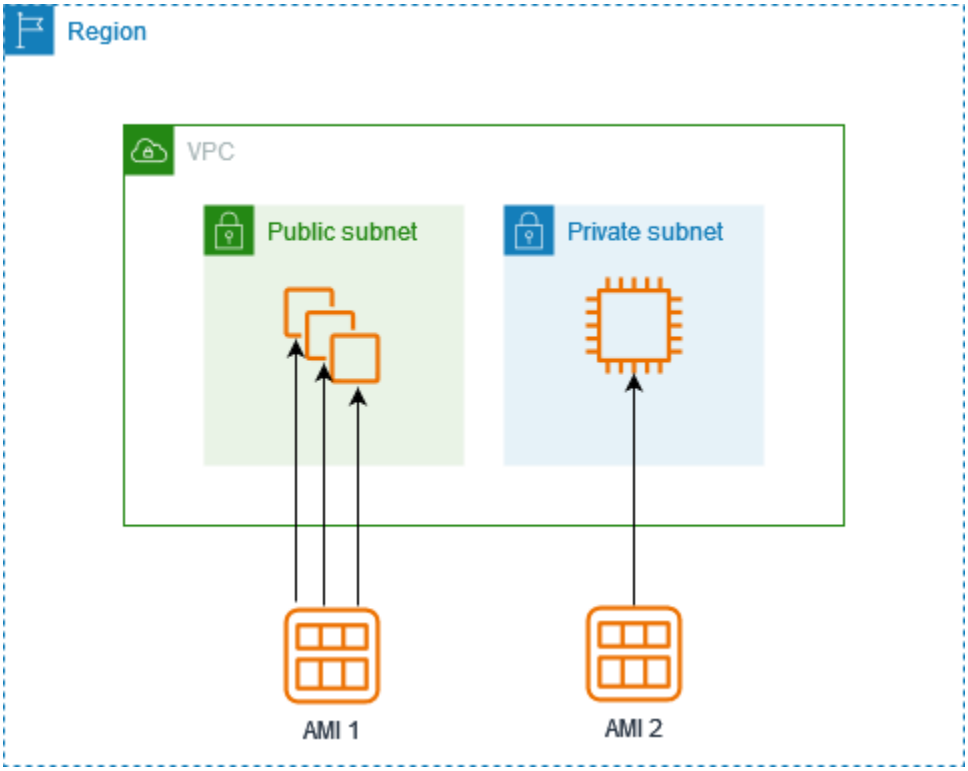
Amazon EC2의 Amazon Machine Images

Amazon Machine Image(AMI)는 Amazon EC2 인스턴스를 설정하고 부팅하는 데 필요한 소프트웨어를 제공하는 이미지입니다. 또한 각 AMI에는 시작하는 인스턴스에 연결할 블록 디바이스를 지정하는 블록 디바이스 매핑이 포함되어 있습니다. 인스턴스를 시작할 때 AMI를 지정해야 합니다. AMI는 인스턴스에 대해 선택한 인스턴스 유형과 호환되어야 합니다. AWS에서 제공하는 AMI, 퍼블릭 AMI, 다른 사람이 공유한 AMI 또는 AWS Marketplace에서 구매한 AMI를 사용할 수 있습니다.

AMI는 다음과 같은 경우에만 사용할 수 있습니다.

- 리전
- 운영 체제
- 프로세서 아키텍처
- 루트 디바이스 유형
- 가상화 유형

동일한 구성의 인스턴스가 여러 개 필요할 때는 한 AMI에서 여러 인스턴스를 시작할 수 있습니다. 다음 다이어그램과 같이 서로 다른 구성의 인스턴스가 필요할 때는 다양한 AMI를 사용하여 인스턴스를 시작할 수 있습니다.



Amazon EC2 인스턴스에서 AMI를 생성한 다음 이를 사용하여 동일한 구성의 인스턴스를 시작할 수 있습니다. AMI를 다른 AWS 리전에 복사한 다음 이를 사용하여 해당 리전에서 인스턴스를 시작할 수 있습니다. 동일한 구성으로 인스턴스를 시작할 수 있게 생성한 AMI를 다른 계정과 공유할 수도 있습니다. AWS Marketplace를 사용하여 AMI를 판매할 수 있습니다.

내용

- Amazon EC2의 AMI 유형 및 특성
- EC2 인스턴스의 요구 사항을 충족하는 AMI 찾기
- Amazon EC2 인스턴스용 AWS Marketplace의 유료 AMI
- Amazon EC2 AMI 수명 주기
- Amazon EC2 부팅 모드를 사용한 인스턴스 시작 동작
- EBS-backed AMI에서 암호화 사용
- Amazon EC2의 공유 AMI 사용량 이해
- Amazon EventBridge를 사용하여 AMI 이벤트 모니터링
- AMI 결제 정보 이해
- Amazon EC2의 AMI 할당량

Amazon EC2의 AMI 유형 및 특성

인스턴스를 시작할 때 선택하는 AMI가 선택하는 인스턴스 유형과 호환되어야 합니다. 다음 유형을 기준으로 사용할 AMI를 선택할 수 있습니다.

- 리전
- 운영 체제
- 프로세서 아키텍처
- 시작 권한
- 루트 디바이스 유형
- 가상화 유형

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 계정, 조직 또는 OU(조직 단위)에 시작 권한을 부여합니다.
암묵적	소유자는 AMI에 대한 암묵적인 시작 권한을 갖습니다.

Amazon 및 Amazon EC2 커뮤니티는 퍼블릭 AMI에 대한 다양한 선택권을 제공합니다. 자세한 내용은 Amazon EC2의 공유 AMI 사용량 이해 섹션을 참조하세요. 개발자들은 자신의 AMI에 비용을 부과할 수 있습니다. 자세한 내용은 Amazon EC2 인스턴스용 AWS Marketplace의 유료 AMI 섹션을 참조하세요.

루트 디바이스 유형

모든 AMI는 Amazon EBS에 의해 지원되는 유형 또는 인스턴스 스토어에 의해 지원되는 유형으로 분류됩니다.

- Amazon EBS 지원 AMI – AMI에서 시작된 인스턴스의 루트 디바이스는 Amazon EBS 스냅샷에서 생성된 Amazon Elastic Block Store(Amazon EBS) 볼륨입니다. Linux와 Windows AMI 모두 지원됩니다.
- Amazon 인스턴스 스토어 지원 AMI – AMI에서 시작한 인스턴스의 루트 디바이스는 Amazon S3에 저장된 템플릿으로부터 생성된 인스턴스 스토어 볼륨입니다. Linux AMI에서만 지원됩니다. Windows AMI는 루트 디바이스에 대한 인스턴스 스토어를 지원하지 않습니다.

자세한 내용은 Amazon EC2 인스턴스용 루트 볼륨 섹션을 참조하세요.

> **Note**
>
> 인스턴스 저장소 지원 AMI는 수명 종료로 간주되며 새 사용에 권장되지 않습니다. C1, C3, D2, I2, M1, M2, M3, R3, X1 같은 이전 인스턴스 유형에서만 지원됩니다.

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

특성	Amazon EBS 지원 AMI	Amazon 인스턴스 스토어 지원 AMI
루트 디바이스 볼륨	EBS 볼륨	인스턴스 스토어 볼륨
인스턴스의 부팅 시간	일반적으로 1분 이하	일반적으로 5분 이하
데이터 지속성	기본적으로 인스턴스가 종료되면 루트 볼륨이 삭제됩니다.* 다른 EBS 볼륨의 데이터는 기본적으로 인스턴스 종료 후에도 유지됩니다.	모든 인스턴스 스토어의 데이터는 인스턴스 수명 주기 동안만 유지됩니다.
중지 상태	중지 상태일 수 있습니다. 인스턴스가 중지되고 실행 중이지 않은 경우에도 루트 볼륨은 Amazon EBS에 유지됩니다.	중지 상태가 될 수 없습니다. 인스턴스가 실행 중이거나 종료되었습니다
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, EBS 볼륨 사용량 및 AMI를 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

* 기본적으로 EBS 루트 볼륨의 `DeleteOnTermination` 플래그는 `true`로 설정되어 있습니다. 이 플래그를 변경하여 종료 후에도 볼륨을 유지하는 방법에 대한 자세한 내용은 Amazon EC2 인스턴스가 종료된 후에도 Amazon EBS 루트 볼륨 유지 섹션을 참조하세요.

** `io2` EBS Block Express에서만 지원됩니다. 자세한 내용은 Amazon EBS 사용 설명서의 Provisioned IOPS SSD Block Express volumes를 참조하세요.

AMI의 루트 디바이스 유형 결정

루트 볼륨의 유형은 EC2 인스턴스를 시작하는 데 사용하는 AMI에 따라 결정됩니다. EC2 인스턴스의 루트 볼륨은 EBS 볼륨 또는 인스턴스 스토어 볼륨입니다.

Nitro 기반 인스턴스는 EBS 루트 볼륨만 지원합니다. 이전 세대 인스턴스 유형인 C1, C3, D2, I2, M1, M2, M3, R3, X1만 인스턴스 저장소 루트 볼륨이 지원되는 인스턴스 유형입니다.

Console

AMI의 루트 디바이스 유형을 결정하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 AMI를 선택하고 AMI를 선택합니다.
3. 세부 정보(Details) 탭에서 루트 디바이스 유형(Root device type)의 값을 다음과 같이 확인합니다.
   - `ebs` - EBS 지원 AMI입니다.
   - `instance store` - 인스턴스 저장소 지원 AMI입니다.

AWS CLI

AMI의 루트 디바이스 유형을 결정하는 방법

describe-images 명령을 사용합니다.

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query Images[].RootDeviceType
```

출력의 예시는 다음과 같습니다.

```
ebs
```

PowerShell

AMI의 루트 디바이스 유형을 결정하는 방법

Get-EC2Image cmdlet을 사용합니다.

```
(Get-EC2Image `
    -ImageId ami-0abcdef1234567890).RootDeviceType.Value
```

출력의 예시는 다음과 같습니다.

```
ebs
```

가상화 유형

Amazon Machine Image는 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. Windows AMI는 HVM AMI입니다.

다음 표에서는 HVM과 PV AMI를 비교합니다.

기능	HVM	PV
설명	HVM AMI는 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 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 하드웨어에서 실행할 수 있습니다. PV-GRUB 및 Amazon EC2에서의 사용에 대한 자세한 내용은 사용자 제공 커널을 참조하세요.
지원되는 인스턴스 유형	모든 최신 인스턴스 유형은 HVM AMI를 지원합니다.	C1, C3, M1, M3, M2, T1 등과 같은 전 세대 인스턴스 유형은 PV AMI를 지원합니다. 최신 세대 인스턴스 유형은 PV AMI를 지원하지 않습니다.

기능	HVM	PV
하드웨어 확장 지원	HVM 게스트는 하드웨어 확장을 활용하여 호스트 시스템의 기본 하드웨어에 빠르게 액세스할 수 있습니다. 향상된 네트워킹과 GPU 처리를 사용해야 합니다. 특수 네트워크 및 GPU 디바이스에 명령을 전달하려면 OS가 기본 하드웨어 플랫폼에 액세스할 수 있어야 하며, HVM 가상화가 이러한 액세스를 제공합니다. 자세한 내용은 Amazon EC2 인스턴스의 향상된 네트워킹 섹션을 참조하세요.	아니요. 향상된 네트워킹 또는 GPU 처리와 같은 특수 하드웨어 확장을 활용할 수 없습니다.
찾는 방법	콘솔 또는 describe-images 명령을 사용하여 AMI의 가상화 유형이 hvm로 설정되어 있는지 확인합니다.	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.

EC2 인스턴스의 요구 사항을 충족하는 AMI 찾기

AMI에는 인스턴스를 시작하는 데 필요한 운영 체제 및 루트 볼륨 유형과 같은 구성 요소와 애플리케이션이 포함됩니다. 인스턴스를 시작하려면 필요에 맞는 AMI를 찾아야 합니다.

AMI를 선택할 때는 시작하려는 인스턴스에 다음과 같은 요구 사항을 고려하세요.

- AMI의 AWS 리전(AMI ID는 리전마다 고유함)
- 운영 체제(예: Linux 또는 Windows)
- 아키텍처(예: 32비트, 64비트 또는 64비트 ARM)
- 루트 디바이스 유형(예: Amazon EBS 또는 인스턴스 스토어)
- 제공업체(예: Amazon Web Services)
- 추가 소프트웨어(예: SQL Server)

Console

인스턴스 시작 마법사를 사용하는 경우 AMI 목록에서 선택하거나 이미지 페이지를 사용하여 사용 가능한 모든 AMI를 검색할 수 있습니다.

인스턴스 시작 마법사를 사용하여 Quick Start 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. AMI ID는 각 AWS 리전에 고유합니다.
3. 콘솔 대시보드에서 인스턴스 시작(Launch instance)을 선택합니다.
4. 애플리케이션 및 OS 이미지(Amazon Machine Image)에서 빠른 시작을 선택하고 인스턴스의 운영 체제(OS)를 선택한 다음 Amazon Machine Image(AMI)에서 목록에 자주 사용되는 AMI 중 하나를 선택합니다. 사용하려는 AMI가 표시되지 않으면 더 많은 AMI 찾아보기(Browse more AMIs)를 선택하여 전체 AMI 카탈로그를 찾아볼 수 있습니다. 자세한 내용은 애플리케이션 및 OS 이미지(Amazon Machine Image) 섹션을 참조하세요.

AMI 페이지를 사용하여 AMI를 찾으려면 다음을 수행합니다.

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. AMI ID는 각 AWS 리전에 고유합니다.
3. 탐색 창에서 AMI를 선택합니다.
4. (선택 사항) 필터 및 검색 옵션을 사용하여 기준과 일치하는 AMI만 볼 수 있도록 표시되는 AMI 목록의 범위를 지정합니다.

예를 들어 AWS에서 제공하는 모든 AMI를 나열하려면 퍼블릭 이미지를 선택합니다. 그런 다음 검색 옵션을 사용하여 표시되는 AMI 목록의 범위를 추가로 지정할 수 있습니다. 검색(Search) 창을 선택하고 메뉴에서 소유자 별칭(Owner alias), = 연산자, 값 amazon을 차례로 선택합니다. 특정 플랫폼(예: Linux 또는 Windows)과 일치하는 AMI를 찾으려면 검색 창을 다시 선택하여 플랫폼을 선택한 다음 = 연산자를 선택한 다음 제공된 목록에서 운영 체제를 선택합니다.

5. (선택 사항) 기본 설정 아이콘을 선택하여 표시할 이미지 속성(예: 루트 디바이스 유형)을 선택합니다. 또는 목록에서 AMI를 선택하고 세부 정보(Details) 탭에서 속성을 조회할 수 있습니다.
6. AMI를 선택하기 전에 해당 AMI가 인스턴스 스토어 기반인지, Amazon EBS 기반인지 확인하고 이 차이점에 따른 영향을 잘 알고 있어야 합니다. 자세한 내용은 루트 디바이스 유형 섹션을 참조하세요.
7. 이 AMI에서 인스턴스를 시작하려면 해당 인스턴스를 선택하고 이미지로 인스턴스 시작을 선택합니다. 콘솔을 사용하여 인스턴스를 시작하는 방법에 대한 자세한 정보는 콘솔에서 인스턴스 시작 마법사를 사용하여 EC2 인스턴스 시작 섹션을 참조하세요. 인스턴스를 시작할 준비가 되지 않은 경우, 나중을 위해 AMI ID를 기록해 둡니다.

AWS CLI

describe-images 명령을 사용하여 요구 사항이 충족되는 AMI를 찾습니다. 기본적으로 이 명령에서는 퍼블릭이고, 본인이 소유하고, 본인과 공유된 모든 AMI가 반환됩니다.

Amazon에서 소유하는 AMI를 찾는 방법

describe-images 명령을 `--owners` 옵션과 함께 사용합니다.

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

Windows AMI를 찾는 방법

다음 필터를 추가하여 Windows AMI만 표시합니다.

```
--filters "Name=platform,Values=windows"
```

EBS 지원 AMI를 찾는 방법

다음 필터를 추가하여 Amazon EBS 기반 AMI만 표시합니다.

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

PowerShell

Get-EC2Image cmdlet을 사용하여 요구 사항이 충족되는 AMI를 찾습니다. 기본적으로 이 cmdlet에서는 퍼블릭이며, 본인이 소유하거나 본인과 공유된 모든 AMI가 반환됩니다.

Amazon에서 소유하는 AMI를 찾는 방법

Get-EC2Image 명령을 `-Owner` 파라미터와 함께 사용합니다.

```
Get-EC2Image -Owner amazon
```

Windows AMI를 찾는 방법

다음 필터를 추가하여 Windows AMI만 표시합니다.

```
-Filter @{Name="platform"; Values="windows"}
```

추가 예제는 AWS Tools for Windows PowerShell 사용 설명서의 Windows PowerShell을 사용하여 Amazon 머신 이미지 찾기를 참조하세요.

관련 리소스

특정 운영 체제용 AMI에 대한 자세한 내용은 다음을 참조하세요.

- Amazon Linux 2023 – Amazon Linux 2023 사용 설명서의 Amazon EC2 AL2023
- Ubuntu – Canonical Ubuntu 웹 사이트의 Amazon EC2 AMI Locator
- RHEL – Red Hat 웹 사이트의 Red Hat Enterprise Linux Images (AMI) Available on Amazon Web Services (AWS)
- Windows Server – AWS Windows AMI 참조

AWS Marketplace에서 구독할 수 있는 AMI에 대한 자세한 내용은 Amazon EC2 인스턴스용 AWS Marketplace의 유료 AMI 섹션을 참조하세요.

사용자가 인스턴스를 시작할 때 사용해야 하는 최신 AMI를 찾는 데 도움이 되는 Systems Manager 사용에 대한 자세한 내용은 다음을 참조하세요.

- Systems Manager 파라미터를 사용하여 AMI 참조

- Systems Manager 퍼블릭 파라미터를 사용하여 최신 AMI 참조

Systems Manager 파라미터를 사용하여 AMI 참조

Amazon EC2 콘솔에서 EC2 인스턴스 시작 마법사를 사용하여 인스턴스를 시작할 때 목록에서 AMI를 선택하거나 AMI ID(이 섹션에서 설명)를 가리키는 AWS Systems Manager 파라미터를 선택할 수 있습니다. 자동화 코드를 사용하여 인스턴스를 시작하는 경우 AMI ID 대신 Systems Manager 파라미터를 지정할 수 있습니다.

Systems Manager 파라미터는 Systems Manager 파라미터 스토어에서 생성할 수 있는 고객 정의 키-값 페어입니다. 파라미터 스토어는 애플리케이션 구성 값을 외부화할 수 있는 중앙 스토어를 제공합니다. 자세한 내용은 AWS Systems Manager 사용 설명서에서 AWS Systems Manager Parameter Store를 참조하세요.

AMI ID를 가리키는 파라미터를 생성할 때는 데이터 유형을 `aws:ec2:image`로 지정해야 합니다. 이 데이터 유형을 지정하면 파라미터가 생성되거나 수정될 때 파라미터 값이 AMI ID로 확인됩니다. 자세한 내용은 AWS Systems Manager 사용 설명서에서 Amazon Machine Image ID에 대한 기본 파라미터 지원을 참조하세요.

목차

- 사용 사례
- 권한
- 제한 사항
- Systems Manager 파라미터를 사용하여 인스턴스 시작

사용 사례

Systems Manager 파라미터를 사용하여 AMI ID를 가리키면 사용자가 인스턴스를 시작할 때 올바른 AMI를 더 쉽게 선택할 수 있습니다. Systems Manager 파라미터를 통해 자동화 코드의 유지 관리가 간소화될 수도 있습니다

사용자의 사용 편의성 향상

특정 AMI를 사용하여 인스턴스를 시작해야 하고 AMI가 정기적으로 업데이트되는 경우 사용자가 Systems Manager 파라미터를 선택하여 AMI를 찾도록 하는 것이 좋습니다. 사용자에게 Systems Manager 파라미터를 선택하도록 하면 인스턴스를 시작하는 데 최신 AMI가 사용될 수 있습니다.

예를 들어 조직에서 매달 최신 운영 체제 및 애플리케이션 패치가 포함된 새 버전의 AMI를 생성하며 사용자가 최신 버전의 AMI를 사용하여 인스턴스를 시작하도록 하려는 경우, 사용자가 최신 버전을 사용하도록 올바른 AMI ID를 가리키는 Systems Manager 파라미터(예: `golden-ami`)를 생성할 수 있습니다. 새 버전의 AMI가 생성될 때마다 이 파라미터의 AMI ID 값을 업데이트하여 항상 최신 AMI를 가리키도록 합니다. 사용자는 매번 동일한 Systems Manager 파라미터를 계속 선택하므로 AMI에 대한 정기 업데이트에 대해 알 필요가 없습니다. AMI에 대한 Systems Manager 파라미터를 사용하면 사용자가 인스턴스 시작에 대한 올바른 AMI를 더 쉽게 선택할 수 있습니다.

자동화 코드 유지 관리 간소화

자동화 코드를 사용하여 인스턴스를 시작하는 경우 AMI ID 대신 Systems Manager 파라미터를 지정할 수 있습니다. 새 버전의 AMI가 생성되면 최신 AMI를 가리키도록 이 파라미터의 AMI ID 값을 변경할 수 있습니다. 파라미터를 참조하는 자동화 코드는 새 버전의 AMI가 생성될 때마다 수정할 필요가 없습니다. 이를 통해 자동화 유지 관리가 간소화되며 배포 비용을 절감할 수 있습니다.

> **Note**
>
> Systems Manager 파라미터가 가리키는 AMI ID를 변경해도 실행 중인 인스턴스는 영향을 받지 않습니다.

권한

인스턴스 시작 마법사에서 AMI ID를 가리키는 Systems Manager 파라미터를 사용하는 경우 IAM 정책에 다음 권한을 추가해야 합니다.

- `ssm:DescribeParameters` – Systems Manager 파라미터를 보고 선택할 수 있는 권한을 부여합니다.
- `ssm:GetParameters` – Systems Manager 파라미터 값을 검색할 수 있는 권한을 부여합니다.

특정 Systems Manager 파라미터에 대한 액세스를 제한할 수도 있습니다. 자세한 내용과 IAM 정책 예는 예: EC2 시작 인스턴스 마법사 사용 섹션을 참조하세요.

제한 사항

AMI 및 Systems Manager 파라미터는 리전별로 다릅니다. 여러 리전에서 동일한 Systems Manager 파라미터 이름을 사용하려면 동일한 이름(예: `golden-ami`)을 가진 Systems Manager 파라미터를 각 리전에 생성합니다. 각 리전에서 Systems Manager 파라미터가 해당 리전의 AMI를 가리키도록 지정합니다.

파라미터 이름은 대/소문자를 구분합니다. 파라미터 이름의 백슬래시는 파라미터가 계층 구조의 일부인 경우에만 필요합니다(예: `/amis/production/golden-ami`). 파라미터가 계층의 일부가 아닌 경우 백슬래시를 생략할 수 있습니다.

Systems Manager 파라미터를 사용하여 인스턴스 시작

인스턴스를 시작할 때 AMI ID를 지정하는 대신 AMI ID를 가리키는 Systems Manager 파라미터를 지정할 수 있습니다.

프로그래밍 방식으로 파라미터를 지정하려면 다음 구문을 사용합니다. 여기서 `resolve:ssm`은 표준 접두사이고 `parameter-name`은 고유한 파라미터 이름입니다.

```
resolve:ssm:parameter-name
```

Systems Manager 파라미터는 버전을 지원합니다. 파라미터의 각 이터레이션에는 고유한 버전 번호가 지정됩니다. 다음과 같이 파라미터의 버전을 참조할 수 있습니다. 여기서 `version`은 고유한 버전 번호입니다. 버전이 지정되지 않은 경우 기본적으로 최신 버전의 파라미터가 사용됩니다.

```
resolve:ssm:parameter-name:version
```

AWS에서 제공하는 퍼블릭 파라미터를 사용하여 인스턴스를 시작하려면 Systems Manager 퍼블릭 파라미터를 사용하여 최신 AMI 참조를 참조하세요.

Console

Systems Manager 파라미터를 사용하여 AMI 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 시작(Launch instance)을 선택합니다.
4. Application and OS Images (Amazon Machine Image)(애플리케이션 및 OS 이미지(Amazon Machine Image))에서 Browse more AMIs(더 많은 AMI 찾아보기)를 선택하세요.
5. 검색 창 오른쪽에 있는 화살표 버튼을 선택한 다음 Search by Systems Manager parameter(Systems Manager 파라미터로 검색)를 선택하세요.
6. [Systems Manager 파라미터(Systems Manager parameter)]에서 파라미터를 선택합니다. 해당 AMI ID가 Currently resolves to(현재 확인된 값) 아래 나타납니다.

7. 검색을 선택합니다. AMI ID와 일치하는 AMI가 목록에 나타납니다.
8. 목록에서 해당 AMI를 선택하고 선택을 선택합니다.

인스턴스 시작 마법사를 사용하여 인스턴스를 시작하는 방법에 대한 자세한 내용은 콘솔에서 인스턴스 시작 마법사를 사용하여 EC2 인스턴스 시작 섹션을 참조하세요.

AWS CLI

Systems Manager 파라미터를 사용하여 인스턴스를 시작하려면

run-instances 명령을 `--image-id` 옵션과 함께 사용합니다. 이 예제에서는 AMI ID를 지정하는 **`golden-ami`**이라는 Systems Manager 파라미터를 사용합니다.

```
--image-id resolve:ssm:/golden-ami
```

Systems Manager 파라미터의 버전을 생성할 수 있습니다. 다음 예시에서는 **`golden-ami`** 파라미터의 버전 2를 지정합니다.

```
--image-id resolve:ssm:/golden-ami:2
```

PowerShell

Systems Manager 파라미터를 사용하여 인스턴스를 시작하려면

`-ImageId` 파라미터와 함께 New-EC2Instance cmdlet을 사용합니다. 이 예제에서는 AMI ID를 지정하는 **`golden-ami`**이라는 Systems Manager 파라미터를 사용합니다.

```
-ImageId "resolve:ssm:/golden-ami"
```

Systems Manager 파라미터의 버전을 생성할 수 있습니다. 다음 예시에서는 **`golden-ami`** 파라미터의 버전 2를 지정합니다.

```
-ImageId "resolve:ssm:/golden-ami:2"
```

Systems Manager 퍼블릭 파라미터를 사용하여 최신 AMI 참조

AWS Systems Manager는 AWS에서 유지 관리하는 퍼블릭 AMI에 대한 퍼블릭 파라미터를 제공합니다. 인스턴스를 시작할 때 퍼블릭 파라미터를 사용하여 최신 AMI를 사용하고 있는지 확인할 수 있습

니다. 예를 들어 퍼블릭 파라미터(/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-default-arm64)는 모든 리전에서 사용할 수 있으며 항상 특정 리전에서 arm64 아키텍처용 Amazon Linux 2023 AMI의 최신 버전을 가리킵니다.

퍼블릭 파라미터는 다음 경로에서 사용할 수 있습니다.

- Linux – /aws/service/ami-amazon-linux-latest
- Windows – /aws/service/ami-windows-latest

자세한 내용은 AWS Systems Manager 사용 설명서의 퍼블릭 파라미터 작업을 참조하세요.

Amazon Linux AMI 나열

AWS CLI

현재 AWS 리전에 있는 Linux AMI를 나열하려면

다음 get-parameters-by-path 명령을 사용합니다. --path 파라미터 값은 Linux AMI에만 해당됩니다.

```
aws ssm get-parameters-by-path \
    --path /aws/service/ami-amazon-linux-latest \
    --query "Parameters[].Name"
```

PowerShell

현재 AWS 리전에 있는 Linux AMI를 나열하려면

Get-SSMParametersByPath cmdlet을 사용합니다.

```
Get-SSMParametersByPath `
    -Path "/aws/service/ami-amazon-linux-latest" | `
    Sort-Object Name | Format-Table Name
```

Windows AMI 나열

AWS CLI

현재 AWS 리전에 있는 Windows AMI를 나열하려면

다음 get-parameters-by-path 명령을 사용합니다. `--path` 파라미터 값은 Windows AMI에만 해당됩니다.

```
aws ssm get-parameters-by-path \
    --path /aws/service/ami-windows-latest \
    --query "Parameters[].Name"
```

PowerShell

현재 AWS 리전에 있는 Windows AMI를 나열하려면

Get-SSMParametersByPath cmdlet을 사용합니다.

```
Get-SSMParametersByPath `
    -Path "/aws/service/ami-windows-latest" | `
    Sort-Object Name | Format-Table Name
```

퍼블릭 파라미터를 사용하여 인스턴스 시작

인스턴스를 시작할 때 퍼블릭 파라미터를 지정하려면 다음 `resolve:ssm:`*`public-parameter`* 구문을 사용합니다. 여기서 `resolve:ssm`은 표준 접두사이고 *`public-parameter`*는 퍼블릭 파라미터의 경로와 이름입니다.

AWS CLI

퍼블릭 파라미터를 사용하여 인스턴스를 시작하려면

run-instances 명령을 `--image-id` 옵션과 함께 사용합니다. 이 예에서는 최신 Amazon Linux 2023 AMI를 사용하여 인스턴스를 시작하기 위해 이미지 ID에 대한 Systems Manager 퍼블릭 파라미터를 지정합니다

```
--image-id resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-default-x86_64
```

PowerShell

퍼블릭 파라미터를 사용하여 인스턴스를 시작하려면

`-ImageId` 파라미터와 함께 New-EC2Instance cmdlet을 사용합니다. 이 예에서는 Windows Server 2022를 위한 최신 AMI를 사용하여 인스턴스를 시작하기 위해 이미지 ID에 대한 Systems Manager 퍼블릭 파라미터를 지정합니다.

```
-ImageId "resolve:ssm:/aws/service/ami-windows-latest/Windows_Server-2022-English-Full-Base"
```

Systems Manager 파라미터를 사용하는 추가 예는 Query for the latest Amazon Linux AMI IDs Using AWS Systems Manager Parameter Store 및 Query for the Latest Windows AMI Using AWS Systems Manager Parameter Store를 참조하세요.

Amazon EC2 인스턴스용 AWS Marketplace의 유료 AMI

유료 AMI는 AWS Marketplace에 판매용으로 등록된 AMI입니다. AWS Marketplace은(는) 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 웹사이트를 참조하세요.

Red Hat과 같은 조직의 서비스 계약에 따라 제공되는 AMI를 포함하여 타사의 AWS Marketplace에서 AMI를 구매할 수 있습니다. AMI를 생성하여 AWS Marketplace에서 다른 Amazon EC2 사용자에게 판매할 수도 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 Amazon EC2의 공유 AMI 사용량 이해 섹션을 참조하세요.

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 m5.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 참조하세요. AWS Marketplace는 Amazon EBS 지원 AMI를 지원합니다.

내용

- AWS Marketplace에서 AMI 판매
- 유료 AMI 찾기
- AWS Marketplace에서 유료 AMI 구입
- 인스턴스에서 AWS Marketplace 제품 코드 검색
- 지원되는 AWS Marketplace 제품에 대한 유료 지원 사용
- 유료 및 지원 AMI에 대한 청구서
- AWS Marketplace 구독 관리

AWS Marketplace에서 AMI 판매

AWS Marketplace을(를) 사용하여 AMI를 판매할 수 있습니다. AWS Marketplace은(는) 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 판매를 참조하세요.

유료 AMI 찾기

유료 AMI는 구매할 수 있는 Amazon Machine Image(AMI)입니다. 유료 AMI에는 제품 코드도 있습니다. AWS Marketplace에서 구매할 수 있는 AMI를 찾을 수 있습니다.

Console

유료 AMI를 찾는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 AMI를 선택합니다.
3. 퍼블릭 이미지를 첫 필터로 선택합니다.
4. 다음 중 하나를 수행합니다.
   - 제품 코드를 알고 있는 경우, 제품 코드(Product code)를 선택한 다음, =를 선택하고, 제품 코드를 입력합니다.
   - 제품 코드를 모르면 검색 창에서 Owner alias=aws-marketplace 필터를 지정합니다. 필요에 따라 추가 필터를 지정합니다.
5. AMI의 ID를 저장합니다.

AWS CLI

유료 AMI를 찾는 방법

다음 describe-images 명령을 사용합니다.

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

출력에는 많은 수의 이미지가 포함됩니다. 필요한 AMI를 결정하는 데 도움이 되도록 필터를 지정할 수 있습니다. AMI를 찾은 후 다음 명령에서 해당 ID를 지정하여 제품 코드를 가져옵니다.

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query Images[*].ProductCodes[].ProductCodeId
```

출력의 예시는 다음과 같습니다.

```
[
    "cdef1234abc567def8EXAMPLE"
]
```

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

```
aws ec2 describe-images \
    --filters "Name=product-code,Values=cdef1234abc567def8EXAMPLE" \
    --query "sort_by(Images, &CreationDate)[-1].[ImageId]"
```

PowerShell

유료 AMI를 찾는 방법

Get-EC2Image cmdlet을 사용합니다.

```
Get-EC2Image -Owner aws-marketplace
```

출력에는 많은 수의 이미지가 포함됩니다. 필요한 AMI를 결정하는 데 도움이 되도록 필터를 지정할 수 있습니다. AMI를 찾은 후 다음 명령에서 해당 ID를 지정하여 제품 코드를 가져옵니다.

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).ProductCodes
```

출력의 예시는 다음과 같습니다.

```
ProductCodeId             ProductCodeType
-------------             ---------------
cdef1234abc567def8EXAMPLE marketplace
```

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

```
(Get-EC2Image -Owner aws-marketplace -Filter @{"Name"="product-
code";"Value"="cdef1234abc567def8EXAMPLE"} | sort CreationDate -Descending | Select-
Object -First 1).ImageId
```

AWS Marketplace에서 유료 AMI 구입

AMI를 사용하여 Amazon EC2 인스턴스를 시작하려면 유료 AMI에 가입(구매)해야 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 Launch Wizard를 사용하여 유료 AMI를 구입할 수 있습니다. 자세한 내용은 AWS Marketplace AMI에서 Amazon EC2 인스턴스 시작 섹션을 참조하세요.

AWS Marketplace을(를) 사용하여 제품 구독

AWS Marketplace를 사용하려면 AWS 계정이 있어야 합니다. AWS Marketplace 제품에서 인스턴스를 시작하려면 Amazon EC2 서비스 사용에 가입하고 인스턴스를 시작할 제품을 구독해야 합니다. 다음 방법 중 하나를 사용하여 AWS Marketplace에서 제품을 구독할 수 있습니다.

- AWS Marketplace 웹 사이트: 1-Click 배포 기능을 사용하여 미리 구성된 소프트웨어를 빠르게 시작할 수 있습니다. 자세한 내용은 AMI-based products in AWS Marketplace를 참조하세요.
- Amazon EC2 Launch Wizard: AMI를 검색하고 마법사에서 직접 인스턴스를 시작할 수 있습니다. 자세한 내용은 AWS Marketplace AMI에서 Amazon EC2 인스턴스 시작 섹션을 참조하세요.

인스턴스에서 AWS Marketplace 제품 코드 검색

인스턴스 메타데이터를 사용하여 인스턴스에 대한 AWS Marketplace 제품 코드를 검색할 수 있습니다. 인스턴스에 제품 코드가 있는 경우 Amazon EC2에서 해당 코드를 반환합니다. 메타데이터 검색에 대한 자세한 내용은 EC2 인스턴스에 대한 인스턴스 메타데이터에 액세스 단원을 참조하세요.

IMDSv2

Linux

Linux 인스턴스에서 다음 명령을 실행합니다.

```
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" http://169.254.169.254/latest/
meta-data/product-codes
```

Windows

Windows 인스턴스에서 다음과 같은 cmdlet을 실행합니다.

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds"
 = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/product-codes
```

IMDSv1

Linux

Linux 인스턴스에서 다음 명령을 실행합니다.

```
curl http://169.254.169.254/latest/meta-data/product-codes
```

Windows

Windows 인스턴스에서 다음 명령을 실행합니다.

```
Invoke-RestMethod -Uri http://169.254.169.254/latest/meta-data/product-codes
```

지원되는 AWS Marketplace 제품에 대한 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.

제한 사항

- 제품 코드 속성을 설정하면 해당 속성을 변경하거나 제거할 수 없습니다.
- 지원 제품을 예약 인스턴스와 함께 사용할 수 없습니다. 항상 지원 제품의 판매자가 지정한 가격을 지불합니다.

AWS CLI

제품 코드를 AMI와 연결하는 방법

modify-image-attribute 명령을 사용합니다.

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --product-codes "cdef1234abc567def8EXAMPLE"
```

PowerShell

제품 코드를 AMI와 연결하는 방법

Edit-EC2ImageAttribute cmdlet을 사용합니다.

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -ProductCode "cdef1234abc567def8EXAMPLE"
```

유료 및 지원 AMI에 대한 청구서

매월 말 그 달에 사용한 유료 또는 지원 AMI에 대해 신용 카드로 청구되는 금액을 이메일로 수신하게 됩니다. 이 청구서는 정기 Amazon EC2 청구서와는 별개입니다. 자세한 내용은 AWS Marketplace 구매자 가이드에서 제품 요금 지불을 참조하세요.

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. 에 로그인합니다..AWS Marketplace
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 취소하는 방법

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 선택한 다음 인스턴스 상태, 인스턴스 종료(삭제)를 선택합니다.
   d. 확인 메시지가 나타나면 종료(삭제)를 선택합니다.
2. AWS Marketplace에 로그인하고 Marketplace 계정을 선택한 후 소프트웨어 구독 관리를 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 Launch Wizard를 통해 해당 AMI를 다시 구독해야 합니다.

Amazon EC2 AMI 수명 주기

Amazon Machine Image(AMI)는 인스턴스를 설정하고 부팅하는 데 필요한 소프트웨어 구성이 들어 있는 이미지입니다. 인스턴스를 시작할 때 AMI를 지정해야 합니다. Amazon에서 제공하는 AMI를 사용하거나 자체 AMI를 생성할 수 있습니다. AMI는 인스턴스를 시작하려는 AWS 리전에 위치해야 합니다.

AMI의 수명 주기에는 AMI 생성, 복사, 사용 중단, 비활성화 및 삭제(등록 취소)가 포함됩니다.

AMI를 생성합니다. Amazon에서는 인스턴스를 시작하는 데 사용할 수 있는 AMI를 제공하지만, 필요에 맞게 사용자 지정 AMI를 생성할 수도 있습니다. 사용자 지정 AMI를 생성하려면 기존 AMI에서 인스턴스를 시작하고 인스턴스를 사용자 지정한 다음(예: 소프트웨어 설치 및 운영 체제 설정 구성) 인스턴스에서 AMI를 생성합니다. 모든 인스턴스 사용자 지정은 새 AMI에 저장되므로 새 AMI에서 시작하는 인스턴스에 이러한 사용자 지정이 포함됩니다.

AMI를 복사합니다. AMI를 사용하여 AMI가 위치한 AWS 리전에서만 인스턴스를 시작할 수 있습니다. 여러 리전에서 동일한 구성으로 인스턴스를 시작해야 하는 경우 다른 리전에 AMI를 복사합니다.

AMI를 사용 중단합니다. AMI를 대체되거나 오래된 것으로 표시하려면 즉시 또는 향후 사용 중단 날짜를 설정할 수 있습니다. 사용되지 않는 AMI는 AMI 목록에서 숨겨지지만, 사용자와 서비스가 AMI ID를 알고 있는 경우 사용되지 않는 AMI를 계속 사용할 수 있습니다.

AMI를 비활성화합니다. AMI의 임시 사용을 방지하기 위해 AMI를 비활성화할 수 있습니다. 비활성화한 AMI는 새 인스턴스를 시작하는 데 사용할 수 없습니다. 그러나 AMI를 다시 활성화하면 인스턴스를 시작하는 데 다시 사용할 수 있습니다. AMI를 비활성화해도 AMI에서 이미 시작된 기존 인스턴스에는 영향을 주지 않습니다.

AMI를 등록 취소(삭제)합니다. AMI가 더 이상 필요하지 않으면 등록을 취소하여 새 인스턴스를 시작하는 데 사용되지 않도록 할 수 있습니다. AMI가 보존 규칙과 일치하면 해당 AMI는 휴지통으로 이동되고, 보존 기간이 만료되기 전에는 복원 가능하며, 기간이 만료된 후에는 영구적으로 삭제됩니다. 보존 규칙과 일치하지 않으면 즉시 영구적으로 삭제됩니다. AMI를 등록 취소해도 AMI에서 시작된 기존 인스턴스에는 영향이 없습니다.

AMI 수명 주기를 자동화합니다. Amazon Data Lifecycle Manager를 사용하여 Amazon EBS 지원 AMI와 해당하는 백업 스냅샷의 생성, 보존, 복사, 사용 중단 및 등록 취소를 자동화할 수 있습니다. EC2 Image Builder를 사용하여 사용자 지정 AMI의 생성, 관리, 배포를 자동화할 수도 있습니다. 자세한 내용은 Amazon EBS 사용자 가이드와 EC2 Image Builder 사용자 가이드의 Amazon Data Lifecycle Manager를 사용하여 백업 자동화를 참조하세요.

내용

- Amazon EBS 지원 AMI 생성
- 인스턴스 스토어 지원 AMI 생성
- Windows Sysprep을 사용하여 Amazon EC2 AMI 생성
- Amazon EC2 AMI 복사
- S3를 사용하여 AMI 저장 및 복원
- 새 Amazon EC2 AMI를 생성하는 데 사용된 소스 AMI 식별
- Amazon EC2 AMI를 마지막으로 사용한 시점 확인
- Amazon EC2 AMI 사용 중단
- Amazon EC2 AMI 비활성화
- Amazon EC2 AMI 등록 취소

Amazon EBS 지원 AMI 생성

Amazon EC2 인스턴스 또는 Amazon EC2 인스턴스의 루트 디바이스스냅샷에서 고유 Amazon EBS 지원 AMI를 생성할 수 있습니다.

인스턴스에서 Amazon EBS 지원 AMI를 생성하려면 기존 Amazon EBS 지원 AMI를 사용하여 인스턴스를 시작합니다. 이 AMI는 VM Import/Export를 사용하여 생성한 AWS Marketplace에서 받은 AMI 또는 액세스할 수 있는 기타 AMI일 수 있습니다. 특정 요구 사항에 맞게 인스턴스를 사용자 지정한 후 새 AMI를 생성하고 등록합니다. 그런 다음 새 AMI를 사용하여 사용자 지정이 있는 새 인스턴스를 시작할 수 있습니다.

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

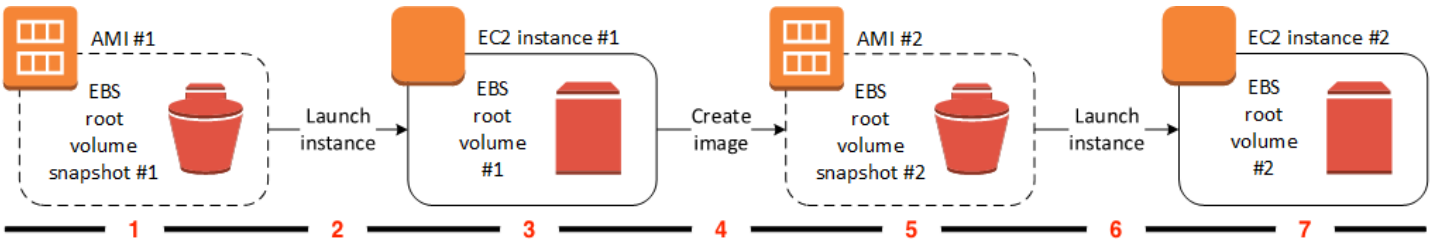
AMI 생성 프로세스는 인스턴스 스토어 지원 AMIs의 경우와는 다릅니다. 자세한 내용은 인스턴스 스토어 지원 AMI 생성 섹션을 참조하세요.

내용

- 인스턴스에서 AMI 생성 개요
- 인스턴스에서 AMI 생성
- 스냅샷에서 AMI 생성

인스턴스에서 AMI 생성 개요

다음 다이어그램은 실행 중인 EC2 인스턴스에서 Amazon EBS 지원 AMI를 생성하는 프로세스를 요약합니다. 기존 AMI로 시작하고, 인스턴스를 시작하고, 사용자 지정하고, 새 AMI를 생성하고, 마지막으로 새 AMI의 인스턴스를 시작합니다. 다이어그램의 숫자는 다음 설명의 숫자와 일치합니다.



1 – AMI #1: 기존 AMI로 시작

생성하려는 AMI와 유사한 기존 AMI를 찾습니다. 이는 AWS Marketplace에서 받은 AMI, VM Import/Export를 사용하여 생성한 AMI 또는 액세스할 수 있는 기타 AMI일 수 있습니다. 필요에 맞게 이 AMI를 사용자 지정합니다.

다이어그램에서 EBS 루트 볼륨 스냅샷 #1(EBS root volume snapshot #1)은 AMI가 Amazon EBS 지원 AMI이고 루트 볼륨에 대한 정보가 이 스냅샷에 저장됨을 나타냅니다.

2 – 기존 AMI에서 인스턴스 시작

AMI를 구성하는 방법은 새 AMI의 기반이 될 AMI에서 인스턴스를 시작한 다음 인스턴스를 사용자 지정하는 것입니다(다이어그램에서 3에 표시됨). 그런 다음 사용자 지정을 포함하는 새 AMI를 생성합니다(다이어그램에서 4에 표시됨).

3 - EC2 인스턴스 #1: 인스턴스 사용자 지정

인스턴스에 연결하고 필요에 맞게 인스턴스를 사용자 지정합니다. 새 AMI에 이러한 사용자 지정이 포함됩니다.

인스턴스에서 다음과 같은 작업을 수행하여 인스턴스를 사용자 지정할 수 있습니다.

- 소프트웨어 및 애플리케이션 설치
- 데이터 복사
- 임시 파일 삭제, 하드 드라이브 조각 모음을 통한 시작 속도 향상
- 추가 EBS 볼륨 연결

4 – 이미지 생성

인스턴스에서 AMI를 생성할 때 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. 인스턴스에 연결된 볼륨이 암호화된 경우 새 AMI는 Amazon EBS 암호화를 지원하는 인스턴스에서만 시작됩니다.

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.

5 - AMI #2: 새 AMI

프로세스가 완료되면 인스턴스의 루트 볼륨에서 새 AMI 및 스냅샷(스냅샷 #2(snapshot #2))이 생성됩니다. 인스턴스 스토어 볼륨 또는 EBS 볼륨을 인스턴스에 추가한 경우 루트 디바이스 볼륨 외에 새 AMI에 대한 블록 디바이스 매핑에는 이러한 볼륨에 대한 정보가 포함됩니다.

Amazon EC2는 자동으로 AMI를 등록합니다.

6 – 새 AMI에서 인스턴스 시작

새 AMI를 사용하여 인스턴스를 시작할 수 있습니다.

7 - EC2 인스턴스 #2: 새 인스턴스

새 AMI를 사용하여 인스턴스를 시작하면 Amazon EC2는 스냅샷을 사용하여 인스턴스의 루트 볼륨에 대한 새 EBS 볼륨을 생성합니다. 인스턴스를 사용자 지정할 때 인스턴스 스토어 볼륨 또는 EBS 볼륨을 추가한 경우 새 AMI에 대한 블록 디바이스 매핑에는 이러한 볼륨에 대한 정보가 포함되고 새 AMI에서 시작하는 인스턴스에 대한 블록 디바이스 매핑에는 이러한 볼륨에 대한 정보가 자동으로 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 Amazon EC2 인스턴스의 볼륨에 대한 블록 디바이스 매핑 섹션을 참조하세요.

EBS 지원 AMI에서 새 인스턴스를 만들 때는 루트 볼륨과 추가 EBS 저장소를 모두 프로덕션 환경에 배치하기 전에 초기화해야 합니다. 자세한 내용은 Amazon EBS 사용 설명서의 Initialize Amazon EBS volumes를 참조하세요.

인스턴스에서 AMI 생성

기존 인스턴스가 있는 경우 이 인스턴스에서 AMI를 생성할 수 있습니다.

Console

API 생성

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. AMI를 생성할 인스턴스를 선택하고 Actions(작업), Image and templates(이미지 및 템플릿), Create image(이미지 생성)를 차례로 선택합니다.

> **Tip**
> 이 옵션이 비활성화되어 있다면 Amazon EBS 지원 인스턴스가 아님을 의미합니다.

4. Create image(이미지 생성) 페이지에서 다음 정보를 지정합니다.
   a. Image name(이미지 이름)에 이미지의 고유 이름을 최대 127자로 입력합니다.
   b. Image description(이미지 설명)에 이미지에 대한 선택적 설명을 최대 255자로 입력합니다.
   c. 인스턴스 재부팅의 경우 확인란을 선택된 상태로 두거나(기본값) 선택을 취소합니다.
      - 인스턴스 재부팅을 선택하면 Amazon EC2가 새 AMI를 생성할 때 인스턴스가 재부팅되므로, 데이터가 사용되지 않는 동안 연결된 볼륨의 스냅샷을 생성하여 일관된 상태를 유지할 수 있습니다.
      - 인스턴스 재부팅 선택을 취소하면 Amazon EC2가 새 AMI를 생성할 때 인스턴스가 종료되거나 재부팅되지 않습니다.

> **Warning**
> 인스턴스 재부팅 선택을 취소하는 경우 생성된 이미지의 파일 시스템 무결성을 보장할 수 없습니다.

   d. Instance volumes(인스턴스 볼륨) - 다음과 같이 루트 볼륨을 수정하고, Amazon EBS 및 인스턴스 스토어 볼륨을 추가할 수 있습니다.

i. 루트 볼륨은 첫 번째 행에 정의됩니다.

- 루트 볼륨의 크기를 변경하려면 [크기(Size)] 에 필요한 값을 입력합니다.
- [종료 시 삭제 여부(Delete on termination)]를 선택할 경우 이 AMI에서 생성된 인스턴스를 종료하면 EBS 볼륨이 삭제됩니다. [종료 시 삭제 여부(Delete on termination)]를 선택 취소할 경우 인스턴스를 종료하면 EBS 볼륨이 삭제되지 않습니다. 자세한 내용은 인스턴스가 종료될 때 데이터 보존 섹션을 참조하세요.

ii. EBS 볼륨을 추가하려면 [볼륨 추가(Add volume)]를 선택합니다(새 행이 추가됨). 스토리지 유형에서 EBS를 선택하고 행의 필드를 작성합니다. 새 AMI에서 인스턴스를 시작하면 추가 볼륨이 인스턴스에 자동으로 연결됩니다. 빈 볼륨은 반드시 포맷하고 탑재해야 합니다. 스냅샷 기반 볼륨을 반드시 마운트해야 합니다.

iii. 인스턴스 스토어 볼륨을 추가하려면 Amazon EC2 AMI에 인스턴스 스토어 볼륨 추가를 참조하세요. 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.

e. [태그(Tags)] - 동일한 태그 또는 다른 태그를 사용하여 AMI와 스냅샷을 태깅할 수 있습니다.

- AMI와 스냅샷에 동일한 태그를 지정하려면 [이미지와 스냅샷을 함께 태깅(Tag image and snapshots together)]을 선택합니다. 생성된 AMI와 모든 스냅샷에 동일한 태그가 적용됩니다.
- AMI와 스냅샷에 다른 태그를 지정하려면 [이미지 및 스냅샷을 개별적으로 태깅(Tag image and snapshots separately)]을 선택합니다. 생성된 AMI와 스냅샷에 서로 다른 태그가 적용됩니다. 그러나 모든 스냅샷의 태그는 동일하며 각 스냅샷에 다른 태그를 지정할 수는 없습니다.

태그를 추가하려면 [태그 추가(Add tag)]를 선택하고 해당 태그에 대한 키와 값을 입력합니다. 각 태그에 대해 반복합니다.

f. AMI를 생성할 준비가 되면 Create image(이미지 생성)를 선택합니다.

5. 생성 중인 AMI의 상태를 보려면

a. 탐색 창에서 AMI를 선택합니다.

b. 필터를 Owned by me(내 소유)로 설정하고 목록에서 AMI를 찾습니다.

처음에 상태는 `pending`이지만 몇 분 후에 `available`로 변경되어야 합니다.

6. (선택 사항) 새 AMI에 대해 생성된 스냅샷을 보려면

   a. 이전 단계에서 찾은 AMI의 ID를 기록해 둡니다.

   b. 탐색 창에서 스냅샷을 선택합니다.

   c. 필터를 Owned by me(내 소유)로 설정한 다음 Description(설명) 열에 새 AMI ID가 있는 스냅샷을 찾습니다.

      이 AMI에서 인스턴스를 시작할 때, Amazon EC2가 이 스냅샷을 사용하여 루트 디바이스 볼륨을 생성합니다.

AWS CLI

API 생성

create-image 명령을 사용합니다.

```
aws ec2 create-image \
    --instance-id i-1234567890abcdef0 \
    --name "my-web-server" \
    --description "My web server image" \
    --no-reboot
```

PowerShell

API 생성

New-EC2Image cmdlet을 사용합니다.

```
New-EC2Image `
    -InstanceId i-1234567890abcdef0 `
    -Name "my-web-server" `
    -Description "My web server image" `
    -NoReboot $true
```

스냅샷에서 AMI 생성

인스턴스의 루트 디바이스 볼륨에 대한 스냅샷이 있으면 이 스냅샷에서 AMI를 생성할 수 있습니다.

> **Note**
>
> 대부분의 경우 Windows, Red Hat, SUSE 및 SQL Server용 AMI를 사용하려면 AMI에 올바른 라이선스 정보가 있어야 합니다. 자세한 내용은 AMI 결제 정보 이해 섹션을 참조하세요. 스냅샷에서 AMI를 생성할 때 `RegisterImage` 작업은 스냅샷의 메타데이터에서 올바른 결제 정보를 가져오지만, 이를 위해서는 적절한 메타데이터가 있어야 합니다. 올바른 결제 정보가 적용되었는지 확인하려면 새 AMI의 플랫폼 세부 정보 필드를 확인합니다. 필드가 비어 있거나 예상 운영 체제 코드(예: Windows, Red Hat, SUSE 또는 SQL)와 일치하지 않는 경우 AMI 생성에 실패한 것이므로 AMI를 삭제하고 인스턴스에서 AMI 생성의 지침을 따라야 합니다.

Console

스냅샷에서 AMI 생성

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [스냅샷(Snapshots)]을 선택합니다.
3. AMI를 생성할 스냅샷을 선택하고 Actions(작업), Create image from snapshot(스냅샷에서 이미지 생성)을 선택합니다.
4. 스냅샷에서 이미지 생성 페이지에서 다음 정보를 지정합니다.
   a. 이미지 이름(Image name)에 이미지를 설명하는 이름을 입력합니다.
   b. 설명(Description)에 이미지에 대한 간략한 설명을 입력합니다.
   c. 아키텍처(Architecture)에 대해 이미지 아키텍처를 선택합니다. 32비트의 경우 i386, 64비트의 경우 x86_64, 64비트 ARM의 경우 arm64, 64비트 macOS의 경우x86_64를 선택합니다.
   d. 루트 디바이스 이름(Root device name)에서 루트 디바이스 볼륨에 사용할 디바이스 이름을 입력합니다. 자세한 내용은 Amazon EC2 인스턴스의 볼륨 디바이스 이름 섹션을 참조하세요.
   e. 가상화 유형(Virtualization type)에서 이 AMI에서 시작된 인스턴스에서 사용할 가상화 유형을 선택합니다. 자세한 내용은 가상화 유형 섹션을 참조하세요.
   f. (반가상화 가상화에만 해당) 커널 ID(Kernel ID)에서 이미지의 운영 체제 커널을 선택합니다. 인스턴스의 루트 디바이스 볼륨 스냅샷을 사용하는 경우 원래 인스턴스와 동일한 커널 ID를 선택합니다. 확실하지 않다면 기본 커널을 사용하십시오.

g. (반가상화 가상화에만 해당) RAM 디스크 ID(RAM disk ID)에서 이미지의 RAM 디스크를 선택합니다. 특정 커널을 선택할 경우 해당 커널을 지원하는 드라이버가 설치된 특정 RAM 디스크를 선택해야 합니다.

h. 부트 모드에서 이미지의 부트 모드를 선택하거나, 이 AMI로 인스턴스가 시작될 때 인스턴스 유형에서 지원하는 부트 모드로 부팅되도록 기본값 사용을 선택합니다. 자세한 내용은 Amazon EC2 AMI의 부팅 모드 설정 섹션을 참조하세요.

i. (선택 사항) 블록 디바이스 매핑에서 루트 볼륨을 사용자 지정하고 데이터 볼륨을 더 추가합니다.

   각 볼륨에 대해 크기, 유형, 성능 특성, 종료 시 삭제 동작 및 암호화 상태를 지정할 수 있습니다. 루트 볼륨의 경우 크기는 스냅샷의 크기보다 작을 수 없습니다. 볼륨 유형의 경우 범용 SSD gp3이 기본 선택 사항입니다.

j. (선택 사항) 태그에서 새 AMI에 하나 이상의 태그를 추가할 수 있습니다. 태그를 추가하려면 [태그 추가(Add tag)]를 선택하고 해당 태그에 대한 키와 값을 입력합니다. 각 태그에 대해 반복합니다.

k. AMI를 생성할 준비가 되면 Create image(이미지 생성)를 선택합니다.

5. (Windows, Red Hat, SUSE, SQL Server 전용) 올바른 결제 정보가 적용되었는지 확인하려면 새 AMI의 플랫폼 세부 정보 필드를 확인합니다. 필드가 비어 있거나 예상 운영 체제 코드(예: Windows 또는 Red Hat)와 일치하지 않는 경우 AMI 생성에 실패한 것이므로 AMI를 삭제하고 인스턴스에서 AMI 생성의 지침을 따라야 합니다.

AWS CLI

AWS CLI를 사용하여 스냅샷에서 AMI를 생성하려면 다음을 수행하세요.

register-image 명령을 사용합니다.

```
aws ec2 register-image \
    --name my-image \
    --root-device-name /dev/xvda \
    --block-device-mappings DeviceName=/dev/
xvda,Ebs={SnapshotId=snap-0db2cf683925d191f}
```

PowerShell

PowerShell을 사용하여 스냅샷에서 AMI를 생성하려면 다음을 수행하세요.

Register-EC2Image cmdlet을 사용합니다.

```
$block = @{SnapshotId=snap-0db2cf683925d191f}
Register-EC2Image `
    -Name my-image `
    -RootDeviceName /dev/xvda `
    -BlockDeviceMapping @{DeviceName="/dev/xvda";Ebs=$block}
```

인스턴스 스토어 지원 AMI 생성

루트 디바이스 볼륨 유형은 인스턴스를 시작할 때 지정한 AMI에 따라 결정됩니다.

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.

Windows AMI는 루트 디바이스에 대한 인스턴스 스토어를 지원하지 않으므로 인스턴스 스토어 지원 Windows AMI를 생성할 수 없습니다.

> **Important**
>
> 다음 인스턴스 유형만 인스턴스 스토어 볼륨을 루트 디바이스로 지원하며, C1, C3, D2, I2, M1, M2, M3, R3 및 X1 등의 인스턴스 스토어 지원 AMI를 요구합니다.

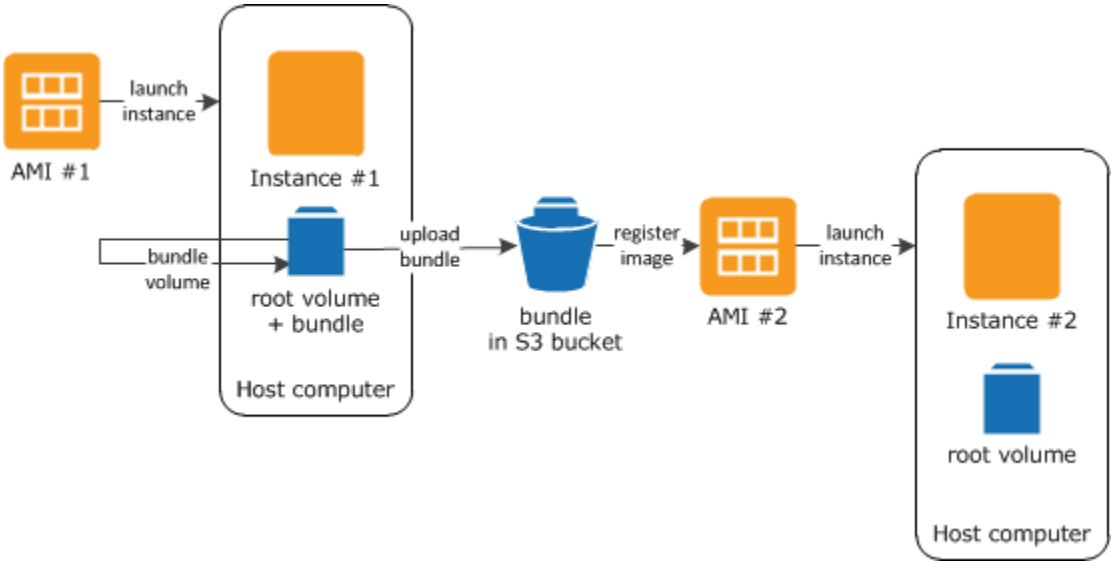
이 AMI의 생성 과정은 Amazon EBS 지원 AMI와 다릅니다. 자세한 내용은 Amazon EBS 지원 AMI 생성 섹션을 참조하세요.

내용

- AMI 생성 개요
- 사전 조건
- Amazon Linux 인스턴스에서 AMI 생성
- Amazon EC2 AMI 도구 설정
- Amazon EC2 AMI 도구 참조
- EBS 지원 AMI로 인스턴스 스토어 지원 AMI 변환

AMI 생성 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.

> **Note**
>
> 인스턴스 스토어 지원 Linux AMI의 S3 버킷에 객체를 업로드하려면 버킷에 대해 ACL을 사용하도록 설정해야 합니다. 그렇지 않으면 Amazon EC2는 업로드할 객체에 ACL을 설정할 수 없습니다. 대상 버킷에서 S3 객체 소유권에 대해 버킷 소유자 강제 설정을 사용하는 경우 ACL이 비활성화되어 있으므로 이 설정은 작동하지 않습니다. 자세한 내용은 객체 소유권 제어 및 버킷에 대해 ACL 사용 중지를 참조하십시오.

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 Amazon EC2 AMI 등록 취소 섹션을 참조하세요.

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 Amazon EC2 인스턴스의 볼륨에 대한 블록 디바이스 매핑 섹션을 참조하세요.

사전 조건

AMI를 만들려면 먼저 다음 작업을 완료해야 합니다.

- AMI 도구를 설치합니다. 자세한 내용은 Amazon EC2 AMI 도구 설정 섹션을 참조하세요.
- AWS CLI를 설치합니다. 자세한 내용은 AWS CLI 시작하기를 참조하세요.
- 번들용 S3 버킷이 있고 버킷에 ACL이 활성화되어 있는지 확인합니다. ACL 구성에 대한 자세한 내용은 ACL 구성을 참조하세요.
  - AWS Management Console을 사용하여 S3 버킷에 폴더를 생성하려면 https://console.aws.amazon.com/s3/에서 Amazon S3 콘솔을 열고 버킷 생성을 선택합니다.
  - AWS CLI을 사용하여 S3 버킷을 생성하려면 mb 명령을 사용할 수 있습니다. 설치된 AMI 도구 버전이 1.5.18 이상일 경우 `ec2-upload-bundle` 명령을 사용하여 S3 버킷을 생성할 수도 있습니다. 자세한 내용은 ec2-upload-bundle 섹션을 참조하세요.
- 번들의 파일이 S3 버킷에서 암호화되지 않았는지 확인합니다. AMI에 암호화가 필요한 경우 EBS 지원 AMI를 대신 사용할 수 있습니다. 자세한 내용은 EBS-backed AMI에서 암호화 사용 섹션을 참조하세요.
- AWS 계정 ID가 있어야 합니다. 자세한 내용은 AWS Account Management 참조 안내서의 View AWS 계정 identifiers를 참조하세요.
- AWS CLI를 사용하기 위한 자격 증명이 있는지 확인합니다. 자세한 내용은 AWS Command Line Interface 사용 설명서의 AWS CLI 인증 및 액세스 제어를 참조하세요.
- X.509 인증서와 그에 따른 프라이빗 키가 있어야 합니다.
  - X.509 인증서를 만들어야 할 경우 서명 인증서 관리 단원을 참조하세요. X.509 인증서 및 프라이빗 키는 AMI를 암호화하고 해독하는 데 사용됩니다.
  - [중국(베이징)] `$EC2_AMITOOL_HOME/etc/ec2/amitools/cert-ec2-cn-north-1.pem` 인증서를 사용합니다.
  - [AWS GovCloud(미국 서부)] `$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.

Amazon Linux 인스턴스에서 AMI 생성

다음 절차에서는 Amazon Linux 1을 실행하는 인스턴스 스토어 지원 인스턴스에서 AMI를 생성하는 방법을 설명합니다. 다른 Linux 배포를 실행하는 인스턴스에서는 작동하지 않을 수 있습니다.

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 조건의 사전 조건을 충족한다고 가정합니다.

다음 명령에서는 자신의 정보로 각각의 *### ## ## ###*를 바꿉니다.

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 등의 보안 복사 도구를 사용하여 컴퓨터의 X.509 인증서와 해당 프라이빗 키를 인스턴스의 `/tmp/cert` 디렉터리로 복사합니다. 다음 `-i` *`my-private-key`*`.pem` 명령의 scp 옵션은 X.509 프라이빗 키가 아니라 SSH를 사용하여 인스턴스에 연결하는 데 사용되는 프라이빗 키입니다. 예:

   ```
   you@your_computer:~ $ scp -i my-private-key.pem /
   path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem /
   path/to/cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem ec2-
   user@ec2-203-0-113-25.compute-1.amazonaws.com:/tmp/cert/
   pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem  100%  717     0.7KB/s   00:00
   cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem  100%  685     0.7KB/s   00:00
   ```

또는 이들은 일반 텍스트 파일이므로 텍스트 편집기에서 인증서와 키를 열고 내용을 /tmp/cert의 새 파일로 복사할 수 있습니다.

2. 인스턴스 내에서 ec2-bundle-vol 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_history가 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. /tmp에 사용 가능한 디스크 공간이 충분하지 않아서 번들을 저장할 수 없으면 -d */path/to/bundle/storage* 옵션을 사용하여 번들을 저장할 다른 위치를 지정합니다. 인스턴스 중에는 /mnt 또는 /media/ephemeral0에 사용자가 사용할 수 있는 휘발성 스토리지가 탑재된 인스턴스도 있으며, 새 Amazon EBS 볼륨을 생성, 연결 및 탑재하여 번들을 저장할 수도 있습니다. 자세한 내용은 Amazon EBS 사용 설명서의 Create an Amazon EBS volume을 참조하세요.

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 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(미국 서부) 리전의 경우 `--ec2cert` 파라미터를 사용하고 사전 조건에 따라 인증서를 지정합니다.

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 Amazon EC2 인스턴스의 볼륨에 대한 블록 디바이스 매핑 섹션을 참조하세요.

   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 > /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 단원을 참조하세요.

   ```
   <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 명령을 실행합니다.

```
[ec2-user ~]$ ec2-upload-bundle -b amzn-s3-demo-bucket/bundle_folder/bundle_name -
m /tmp/image.manifest.xml -a your_access_key_id -s your_secret_access_key
```

> **Important**
>
> US East (N. Virginia) 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 `/tmp` 명령을 사용하여 인스턴스의 rm 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 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 amzn-s3-demo-
bucket/bundle_folder/bundle_name/image.manifest.xml --name AMI_name --
virtualization-type hvm
```

> **Important**
>
> 이전에 ec2-upload-bundle 명령에 리전을 지정한 경우 이 명령에도 해당 리전을 다시 지정하세요.

Amazon EC2 AMI 도구 설정

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. 예:

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

2. wget 또는 curl과 같은 도구를 사용하여 RPM 파일을 다운로드합니다. 예:

   ```
   [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 OK임을 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 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 예시를 보면 이전 로드 오류에서 없다고 표시된 파일이 /usr/lib/ruby/site_ruby 및 /usr/lib64/ruby/site_ruby에 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 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. 예:

```
[ec2-user ~]$ sudo apt-get update -y && sudo apt-get install -y ruby unzip
```

2. wget 또는 curl과 같은 도구를 사용하여 .zip 파일을 다운로드합니다. 예:

```
[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. 예:

```
[ec2-user ~]$ export EC2_AMITOOL_HOME=/usr/local/ec2/ec2-ami-tools-x.x.x
```

5. 도구를 PATH 환경 변수에 추가합니다. 예:

```
[ec2-user ~]$ export PATH=$EC2_AMITOOL_HOME/bin:$PATH
```

6. ec2-ami-tools-version 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
```

Amazon EC2 AMI 도구 참조

AMI 도구 명령을 사용하여 인스턴스 스토어 지원 Linux AMI를 생성하고 관리할 수 있습니다. 도구를 설정하려면 Amazon EC2 AMI 도구 설정을 참조하세요.

액세스 키에 대한 자세한 내용은 IAM 사용 설명서의 IAM 사용자를 위한 액세스 키 관리를 참조하세요.

명령

- ec2-ami-tools-version
- ec2-bundle-image
- ec2-bundle-vol
- ec2-delete-bundle
- ec2-download-bundle
- ec2-migrate-manifest
- ec2-unbundle
- ec2-upload-bundle
- AMI 도구의 일반 옵션

ec2-ami-tools-version

설명

AMI 도구 버전을 설명합니다.

구문

ec2-ami-tools-version

출력

버전 정보입니다.

예제

이 예시 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.

Required: Yes

`-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.

Required: Yes

`-u, --user` 계정

사용자의 AWS 계정 ID(대시 없이)입니다.

Required: Yes

`-i, --image` path

번들링할 이미지에 대한 경로입니다.

Required: Yes

`-d, --destination` path

번들을 생성할 디렉터리입니다.

기본값: `/tmp`

Required: No

`--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. Amazon EC2 AMI 도구 설정의 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` 아키텍처

이미지 아키텍처. 명령줄에 아키텍처를 제공하지 않으면 번들링 시작 시 해당 메시지가 표시됩니다.

유효한 값: i386 | x86_64

Required: No

`--productcodes` code1,code2,...

등록 시 이미지에 연결할 제품 코드로, 쉼표로 구분됩니다.

Required: No

`-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

Required: No

`-p, --prefix` prefix

번들링된 AMI 파일의 파일 이름 접두사입니다.

기본값: 이미지 파일의 이름입니다. 예를 들어, 이미지 경로가 /var/spool/my-image/version-2/debian.img이면, 기본 접두사는 debian.img입니다.

Required: No

`--kernel` kernel_id

사용되지 않음. register-image를 사용하여 커널을 설정합니다.

Required: No

`--ramdisk` ramdisk_id

사용되지 않음. register-image를 사용하여 RAM 디스크를 설정합니다(필요한 경우).

필수 여부: 아니요

출력

번들링 프로세스의 단계 및 상태를 설명하는 상태 메시지입니다.

예제

이 예시에서는 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 지원 AMI 생성 섹션을 참조하세요.

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.

Required: Yes

`-k, --privatekey` path

사용자의 PEM 인코딩된 RSA 키 파일의 경로입니다.

Required: Yes

`-u, --user` 계정

사용자의 AWS 계정 ID(대시 없이)입니다.

Required: Yes

`-d, --destination` destination

번들을 생성할 디렉터리입니다.

기본값: `/tmp`

Required: No

`--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. Amazon EC2 AMI 도구 설정의 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` 아키텍처

이미지 아키텍처입니다. 명령줄에 이를 제공하지 않으면 번들링 시작 시 이를 제공하라는 메시지가 표시됩니다.

유효한 값: `i386` | `x86_64`

Required: No

`--productcodes` code1,code2,...

등록 시 이미지에 연결할 제품 코드로, 쉼표로 구분됩니다.

Required: No

`-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

Required: No

`-a, --all`

원격으로 마운트된 파일 시스템의 디렉터리를 포함하여 모든 디렉터리를 번들링합니다.

Required: No

`-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.

Required: No

`-i, --include` file1,file2,...

번들 작업에 포함할 파일 목록입니다. 지정된 파일은 중요한 정보를 포함할 수 있으므로 AMI에서 제외되지 않습니다.

Required: No

`--no-filter`

지정한 경우 지정한 파일이 중요한 정보를 포함할 수 있으므로 AMI에서 파일을 제외하지 않습니다.

Required: No

`-p, --prefix` prefix

번들링된 AMI 파일의 파일 이름 접두사입니다.

기본값: `image`

Required: No

`-s, --size` size

생성할 이미지 파일의 크기(MB, 1024 * 1024바이트)입니다. 최대 크기는 10240MB입니다.

기본값: 10240

Required: No

`--[no-]inherit`

이미지가 인스턴스의 메타데이터를 상속해야 하는지 여부를 나타냅니다(기본값은 상속). `--inherit`를 활성화했지만 인스턴스 메타데이터에 액세스할 수 없으면 번들링이 실패합니다.

Required: No

`-v, --volume` 볼륨

번들을 생성해 올 마운트된 볼륨의 절대 경로입니다.

기본값: 루트디렉터리입니다(/).

Required: No

`-P, --partition` type

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`

Required: No

`-S, --script` 스크립트

번들링 작업 직전에 실행할 사용자 정의 스크립트입니다. 스크립트에서 하나의 인수(볼륨의 마운트 지점)를 예상해야 합니다.

Required: No

`--fstab` path

이미지에 번들링할 fstab 경로입니다. 지정하지 않으면 Amazon EC2가 /etc/fstab을 번들링합니다.

Required: No

`--generate-fstab`

Amazon EC2-제공 fstab을 사용하여 볼륨을 번들링합니다.

Required: No

`--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.

Required: No

`--kernel` kernel_id

사용되지 않음. register-image를 사용하여 커널을 설정합니다.

Required: No

`--ramdisk`ramdisk_id

사용되지 않음. register-image를 사용하여 RAM 디스크를 설정합니다(필요한 경우).

필수 여부: 아니요

출력

번들링 단계 및 상태를 설명하는 상태 메시지입니다.

예제

이 예시에서는 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).

Required: Yes

`-a, --access-key` access_key_id

AWS 액세스 키 ID입니다.

Required: Yes

`-s, --secret-key` secret_access_key

AWS 보안 액세스 키입니다.

Required: Yes

`-t, --delegation-token` token

AWS 요청에 함께 전달되는 위임 토큰입니다. 자세한 내용은 IAM 사용 설명서의 임시 보안 자격 증명을 참조하세요.

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.

`--sigvversion`

요청 서명 시 사용하는 서명 버전입니다.

유효한 값: `2 | 4`

기본값: `4`

Required: No

`-m, --manifest`경로

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.

Required: No

`--retry`

모든 Amazon S3 오류에서 작업당 최대 5회 자동으로 재시도합니다.

Required: No

`-y, --yes`

모든 질문 메시지에 대한 답은 자동적으로 예라고 가정합니다.

필수 여부: 아니요

출력

Amazon EC2에서 삭제 프로세스의 단계 및 상황을 나타내는 상태 메시지를 표시합니다.

예제

이 예시에서는 Amazon S3에서 번들을 삭제합니다.

```
[ec2-user ~]$ ec2-delete-bundle -b amzn-s3-demo-bucket -a your_access_key_id -
s your_secret_access_key
Deleting files:
amzn-s3-demo-bucket/image.manifest.xml
amzn-s3-demo-bucket/image.part.00
amzn-s3-demo-bucket/image.part.01
amzn-s3-demo-bucket/image.part.02
amzn-s3-demo-bucket/image.part.03
amzn-s3-demo-bucket/image.part.04
amzn-s3-demo-bucket/image.part.05
amzn-s3-demo-bucket/image.part.06
Continue? [y/n]
y
Deleted amzn-s3-demo-bucket/image.manifest.xml
Deleted amzn-s3-demo-bucket/image.part.00
Deleted amzn-s3-demo-bucket/image.part.01
```

```
Deleted amzn-s3-demo-bucket/image.part.02
Deleted amzn-s3-demo-bucket/image.part.03
Deleted amzn-s3-demo-bucket/image.part.04
Deleted amzn-s3-demo-bucket/image.part.05
Deleted amzn-s3-demo-bucket/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).

Required: Yes

`-a, --access-key` access_key_id

AWS 액세스 키 ID입니다.

Required: Yes

`-s, --secret-key` secret_access_key

AWS 보안 액세스 키입니다.

Required: Yes

`-k, --privatekey` path

매니페스트를 해독하는 데 사용되는 프라이빗 키입니다.

Required: Yes

`--url` url

Amazon S3 서비스 URL입니다.

기본값: `https://s3.amazonaws.com/`

Required: No

`--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

Required: No

`-m, --manifest` file

매니페스트 파일 이름입니다(경로 제외). 매니페스트(`-m`) 또는 접두사(`-p`) 중 하나를 지정하는 것이 좋습니다.

Required: No

`-p, --prefix` prefix

번들링된 AMI 파일의 파일 이름 접두사입니다.

기본값: `image`

Required: No

`-d, --directory` directory

다운로드된 번들이 저장되는 디렉터리입니다. 존재하는 디렉터리여야 합니다.

기본값: 현재 작업 디렉터리입니다.

Required: No

`--retry`

모든 Amazon S3 오류에서 작업당 최대 5회 자동으로 재시도합니다.

필수 여부: 아니요

출력

다운로드 프로세스의 다양한 단계를 나타내는 상태 메시지가 표시됩니다.

예제

이 예시는 `bundled` 디렉터리(Linux mkdir 명령 사용)를 생성하고 `amzn-s3-demo-bucket` Amazon S3 버킷에서 번들을 다운로드합니다.

```
[ec2-user ~]$ mkdir bundled
[ec2-user ~]$ ec2-download-bundle -b amzn-s3-demo-bucket/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 amzn-s3-demo-bucket to mybundle/
image.manifest.xml ...
Downloading part image.part.00 from amzn-s3-demo-bucket/bundles/bundle_name to
 mybundle/image.part.00 ...
Downloaded image.part.00 from amzn-s3-demo-bucket
Downloading part image.part.01 from amzn-s3-demo-bucket/bundles/bundle_name to
 mybundle/image.part.01 ...
Downloaded image.part.01 from amzn-s3-demo-bucket
Downloading part image.part.02 from amzn-s3-demo-bucket/bundles/bundle_name to
 mybundle/image.part.02 ...
Downloaded image.part.02 from amzn-s3-demo-bucket
Downloading part image.part.03 from amzn-s3-demo-bucket/bundles/bundle_name to
 mybundle/image.part.03 ...
Downloaded image.part.03 from amzn-s3-demo-bucket
Downloading part image.part.04 from amzn-s3-demo-bucket/bundles/bundle_name to
 mybundle/image.part.04 ...
Downloaded image.part.04 from amzn-s3-demo-bucket
Downloading part image.part.05 from amzn-s3-demo-bucket/bundles/bundle_name to
 mybundle/image.part.05 ...
Downloaded image.part.05 from amzn-s3-demo-bucket
Downloading part image.part.06 from amzn-s3-demo-bucket/bundles/bundle_name to
 mybundle/image.part.06 ...
```

```
Downloaded image.part.06 from amzn-s3-demo-bucket
```

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.

Required: Yes

`-k, --privatekey` path

사용자의 PEM 인코딩된 RSA 키 파일의 경로입니다.

Required: Yes

`--manifest` path

매니페스트 파일 경로입니다.

Required: Yes

`-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. Amazon EC2 AMI 도구 설정의 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 내용을 알아보려면 Amazon EC2 사용 설명서의 사용자 제공 커널을 참조하세요.

필수 여부: 아니요

`--ramdisk` ramdisk_id

선택할 RAM 디스크의 ID입니다.

> **Important**
>
> 커널 및 RAM 디스크 대신 PV-GRUB를 사용하는 것이 좋습니다. 자세한 내용을 알아보려면 Amazon EC2 사용 설명서의 사용자 제공 커널을 참조하세요.

필수 여부: 아니요

출력

번들링 프로세스의 단계 및 상태를 설명하는 상태 메시지입니다.

예제

이 예시는 `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.

Required: Yes

`-m, --manifest` path

매니페스트 파일 경로입니다.

Required: Yes

`-s, --source` source_directory

번들이 포함된 디렉터리입니다.

기본값: 현재 디렉터리입니다.

Required: No

`-d, --destination` destination_directory

AMI 번들을 해제해 넣을 디렉터리입니다. 대상 디렉터리가 있어야 합니다.

기본값: 현재 디렉터리입니다.

필수 여부: 아니요

예제

이 Linux 및 UNIX 예시는 `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
```

출력

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 액세스 제어 목록(ACL)을 설정합니다. 자세한 내용은 인스턴스 스토어 지원 AMI 생성 섹션을 참조하세요.

> **Note**
>
> 인스턴스 스토어 지원 Linux AMI의 S3 버킷에 객체를 업로드하려면 버킷에 대해 ACL을 사용하도록 설정해야 합니다. 그렇지 않으면 Amazon EC2는 업로드할 객체에 ACL을 설정할 수 없습니다. 대상 버킷에서 S3 객체 소유권에 대해 버킷 소유자 강제 설정을 사용하는 경우 ACL이 비활성화되어 있으므로 이 설정은 작동하지 않습니다. 자세한 내용은 객체 소유권 제어 및 버킷에 대해 ACL 사용 중지를 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). 또한 버킷이 존재하지 않고 AMI 도구 버전이 1.5.18 이상인 경우 이 명령은 버킷에 대한 ACL을 설정합니다.

필수 여부: 예

`-a, --access-key` access_key_id

사용자의 AWS 액세스 키 ID입니다.

Required: Yes

`-s, --secret-key` secret_access_key

AWS 보안 액세스 키입니다.

Required: Yes

`-t, --delegation-token` token

AWS 요청에 함께 전달되는 위임 토큰입니다. 자세한 내용은 IAM 사용 설명서의 임시 보안 자격 증명을 참조하세요.

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.

Required: Yes

--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/

Required: No

--region region

대상 S3 버킷의 요청 서명에 사용할 리전입니다.

- 버킷이 없고 리전을 지정하지 않는 경우 이 도구는 (us-east-1에서) 위치 제한 없이 버킷을 만듭니다.
- 버킷이 없고 리전을 지정하는 경우 이 도구는 지정된 리전에 버킷을 만듭니다.
- 버킷이 있고 리전을 지정하지 않는 경우 이 도구는 버킷의 위치를 사용합니다.
- 버킷이 있고 us-east-1을 리전으로 지정하는 경우 이 도구는 오류 메시지 없이 버킷의 실제 위치를 사용하며 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

Required: No

`--acl` acl

번들링된 이미지의 액세스 제어 목록 정책입니다.

유효한 값: `public-read` | `aws-exec-read`

기본값: `aws-exec-read`

Required: No

`-d, --directory` directory

번들링된 AMI 파트가 포함된 디렉터리입니다.

기본값: 매니페스트 파일이 포함된 디렉터리(`-m` 옵션 참조)입니다.

Required: No

`--part` part

지정된 파트와 모든 후속 파트의 업로드를 시작합니다. 예를 들어 `--part 04`입니다.

Required: No

`--retry`

모든 Amazon S3 오류에서 작업당 최대 5회 자동으로 재시도합니다.

Required: No

`--skipmanifest`

매니페스트를 업로드하지 않습니다.

Required: No

`--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-1에서) 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 여부: 아니요

출력

Amazon EC2에서 업로드 프로세스의 단계 및 상황을 나타내는 상태 메시지를 표시합니다.

예제

이 예시에서는 `image.manifest.xml` 매니페스트에서 지정한 번들을 업로드합니다.

```
[ec2-user ~]$ ec2-upload-bundle -b amzn-s3-demo-bucket/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 amzn-s3-demo-bucket ...
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 AMI로 인스턴스 스토어 지원 AMI 변환

사용자 소유의 인스턴스 스토어 기반 Linux AMI를 Amazon EBS 기반 Linux AMI로 변환할 수 있습니다.

> **Important**
>
> 소유하지 않은 AMI는 변환할 수 없습니다.

인스턴스 스토어 기반 AMI를 Amazon EBS 기반 AMI로 변환하려면

1. Amazon EBS 기반 AMI에서 Amazon Linux 인스턴스를 시작합니다. 자세한 내용은 콘솔에서 인스턴스 시작 마법사를 사용하여 EC2 인스턴스 시작 섹션을 참조하세요. Amazon Linux 인스턴스에는 AWS CLI 및 AMI 도구가 미리 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 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. 예시:

```
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 CLI를 사용하도록 환경 변수를 구성합니다. 자세한 내용은 환경 변수를 참조하세요.

a. (권장) AWS 액세스 키, 보안 키 및 세션 토큰에 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
[ec2-user ~]$ export AWS_SESSION_TOKEN=your_session_token
```

b. 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 Elastic Block Store(Amazon EBS) 볼륨을 준비합니다.

a. create-volume 명령을 사용하여 인스턴스와 동일한 가용 영역에 빈 EBS 볼륨을 생성합니다. 명령 출력의 볼륨 ID를 메모해 둡니다.

> **Important**
>
> 이 EBS 볼륨은 크기가 원본 인스턴스 스토어 루트 볼륨보다 크거나 같아야 합니다.

```
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.

```
aws ec2 attach-volume \
    --volume-id vol-01234567890abcdef \
    --instance-id i-1234567890abcdef0 \
    --region us-west-2
```

5. 번들용 폴더를 생성합니다.

```
[ec2-user ~]$ mkdir /tmp/bundle
```

6. /tmp/bundle using the ec2-download-bundle 명령을 사용하여 인스턴스 스토어 기반 AMI용 번들을 /tmp/bundle로 다운로드합니다.

```
[ec2-user ~]$ ec2-download-bundle -b amzn-s3-demo-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 명령을 사용하여 번들에서 이미지 파일을 다시 구성합니다.

a. 디렉터리를 번들 폴더로 변경합니다.

```
[ec2-user ~]$ cd /tmp/bundle/
```

b. ec2-unbundle 명령을 실행합니다.

```
[ec2-user bundle]$ ec2-unbundle -m image.manifest.xml --privatekey /path/to/pk-
HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem
```

8. 번들링되지 않은 이미지의 파일을 새 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/sdb1이지만, 디바이스 이름이 이와 다를 수 있습니다. 볼륨이 파티셔닝되지 않은 경우 마운트할 디바이스는 /dev/sdb(디바이스 파티션 끝 숫자가 없음)와 비슷할 것입니다.

11. 새 EBS 볼륨에 대한 탑재 지점을 생성하고 볼륨을 탑재합니다.

```
[ec2-user bundle]$ sudo mkdir /mnt/ebs
[ec2-user bundle]$ sudo mount /dev/sdb1 /mnt/ebs
```

12. 주로 사용하는 텍스트 편집기(예: /etc/fstab 또는 vim)를 사용하여 EBS 볼륨의 nano 파일을 열고 인스턴스 스토어(휘발성) 볼륨에 대한 항목을 모두 제거합니다. EBS 볼륨은 /mnt/ebs에 탑재되므로 fstab 파일은 /mnt/ebs/etc/fstab에 있습니다.

```
[ec2-user bundle]$ sudo nano /mnt/ebs/etc/fstab
#
LABEL=/     /           ext4    defaults,noatime  1   1
tmpfs       /dev/shm    tmpfs   defaults        0   0
devpts      /dev/pts    devpts  gid=5,mode=620  0   0
sysfs       /sys        sysfs   defaults        0   0
proc        /proc       proc    defaults        0   0
/dev/sdb        /media/ephemeral0       auto    defaults,comment=cloudconfig    0
     2
```

이 예에서는 마지막 줄을 제거해야 합니다.

13. 볼륨 마운트를 해제하고 인스턴스에서 볼륨을 분리합니다.

```
[ec2-user bundle]$ sudo umount /mnt/ebs
[ec2-user bundle]$ aws ec2 detach-volume --volume-id vol-01234567890abcdef --
region us-west-2
```

14. 다음과 같이 새 EBS 볼륨에서 AMI를 생성합니다.

    a. 새 EBS 볼륨의 스냅샷을 생성합니다.

```
[ec2-user bundle]$ aws ec2 create-snapshot --region us-west-2 --description
 "your_snapshot_description" --volume-id vol-01234567890abcdef
```

b. 스냅샷이 완전한지 확인합니다.

```
[ec2-user bundle]$ aws ec2 describe-snapshots --region us-west-2 --snapshot-
id snap-0abcdef1234567890
```

c. describe-imagesaki 명령을 사용하여 원래의 AMI에 사용된 프로세스 아키텍처, 가상화 유형 및 커널 이미지(describe-images)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-0abcdef1234567890 --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_64이고 커널 이미지 ID가 aki-fc8f11cc입니다. 다음 단계에서는 이들 값을 사용합니다. 위 명령의 출력에 ari ID도 나열되면 이 ID도 메모해 둡니다.

d. 새 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=snap-0abcdef1234567890} --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 EBS 볼륨을 삭제할 수 있습니다.

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

Windows Sysprep을 사용하여 Amazon EC2 AMI 생성

Microsoft System Preparation(Windows Sysprep) 도구는 새 이미지를 캡처하기 전에 인스턴스별 시스템 구성이 제거된 상태로 일반 버전의 운영 체제를 생성합니다.

소프트웨어 및 설정으로 사전 설치 및 사전 구성된 안전한 맞추형 최신 “골드” 서버 이미지의 생성, 관리 및 배포를 자동화하려면 EC2 Image Builder를 사용하는 것이 좋습니다.

Windows Sysprep을 사용하면 Windows 시작 에이전트를 통해 표준화된 AMI를 생성할 수도 있습니다. 자세한 내용은 the section called “시작 에이전트와 함께 Windows Sysprep 사용” 단원을 참조하십시오.

> **Important**
>
> Windows Sysprep을 사용하여 인스턴스 백업을 생성하지 마세요. Windows Sysprep은 시스템에 고유한 정보를 삭제하는데, 이는 인스턴스 백업에 의도하지 않은 결과를 낳을 수 있습니다.

Windows Sysprep의 문제를 해결하려면 Amazon EC2 Windows 인스턴스 Sysprep 문제 해결의 내용을 참조하세요.

내용

- Windows Sysprep 단계
- 시작하기 전 준비 사항
- 시작 에이전트와 함께 Windows Sysprep 사용

Windows Sysprep 단계

Windows Sysprep은 다음 단계를 거칩니다.

- 일반화: Sysprep 도구는 이미지 고유 정보 및 설정을 삭제합니다. 예를 들어 Windows Sysprep은 보안 식별자(SID), 컴퓨터 이름, 이벤트 로그, 특정 드라이버를 제거합니다. 이 단계가 완료되면 운영 체제(OS)는 AMI를 생성할 준비가 됩니다.

  > **Note**
  >
  > `PersistAllDeviceInstalls` 설정이 true로 기본 설정되어 있기 때문에 Windows Sysprep을 Windows 시작 에이전트와 함께 실행할 때 시스템은 드라이버가 제거되는 것을 방지합니다.

- 특수화: 플러그-앤-플레이가 컴퓨터를 스캔하여 검색된 장치에 대해 드라이버를 설치합니다. Sysprep 도구는 컴퓨터 이름과 SID와 같은 OS 요건을 생성합니다. 필요한 경우 이 단계에서 명령을 실행할 수 있습니다.

- Out-of-Box Experience(OOBE): 시스템은 Windows 설치 축약 버전을 실행하여 사용자에게 시스템 언어, 표준 시간대, 등록된 조직과 같은 정보를 입력하도록 요청합니다. Windows 시작 에이전트와 함께 Windows Sysprep을 실행할 때 응답 파일은 이 단계를 자동으로 실행합니다.

시작하기 전 준비 사항

- Windows Sysprep을 수행하기 전에 Windows Sysprep을 실행할 단일 관리자 계정을 제외하고 로컬 사용자 계정 및 계정 프로파일을 모두 제거하는 것이 좋습니다. 추가 계정 및 프로파일로 Windows Sysprep를 수행하면 프로파일 데이터 손실 또는 Windows Sysprep 완료 실패 등 예기치 못한 동작이 발생할 수 있습니다.
- Sysprep 개요에 대해 자세히 알아보세요.
- 서버 역할에 대한 Sysprep 지원에 대해 알아보세요.

시작 에이전트와 함께 Windows Sysprep 사용

Windows 시작 에이전트 중 하나가 설치된 AMI로 시작할 때 Windows Sysprep을 사용하여 표준화된 Amazon Machine Image(AMI)를 생성할 수 있습니다.

EC2Launch v2와 함께 Windows Sysprep 사용

이 섹션에는 이미지가 준비될 때 EC2Launch v2 서비스에서 수행하는 작업의 세부 정보가 포함됩니다. 또한 EC2Launch v2 서비스와 함께 Windows Sysprep을 사용하여 표준화된 AMI를 생성하는 단계도 포함되어 있습니다.

EC2Launch v2와 함께 Windows Sysprep 사용 주제

- Windows Sysprep 작업
- Sysprep 이후
- EC2Launch v2와 함께 Windows Sysprep 실행

Windows Sysprep 작업

Windows Sysprep과 EC2Launch v2는 이미지를 준비할 때 다음 작업을 수행합니다.

1. EC2Launch 설정(EC2Launch settings) 대화 상자에서 Sysprep을 이용해 종료(Shutdown with Sysprep)를 선택하면 시스템은 `ec2launch sysprep` 명령을 실행합니다.

2. EC2Launch v2는 `unattend.xml`에서 레지스트리 값을 읽어 `HKEY_USERS\.DEFAULT\Control Panel\International\LocaleName` 파일의 내용을 편집합니다. 이 파일은 `C:\ProgramData\Amazon\EC2Launch\sysprep` 디렉터리에 있습니다.
3. 시스템이 `BeforeSysprep.cmd`를 실행합니다. 이 명령은 다음과 같은 레지스트리 키를 생성합니다.

   ```
   reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 1 /f
   ```

   레지스트리 키는 RDP 연결이 다시 활성화될 때까지 RDP 연결을 비활성화합니다. RDP 연결을 비활성화하는 것은 필수적인 보안 조치입니다. 왜냐하면 Windows Sysprep이 실행된 이후 첫 번째 부팅 세션 동안 RDP가 연결을 허용하고 관리자 암호가 비어 있는 짧은 시간이 있기 때문입니다.
4. EC2Launch v2 서비스는 다음 명령을 실행하여 Windows Sysprep을 호출합니다.

   ```
   sysprep.exe /oobe /generalize /shutdown /unattend: "C:\ProgramData\Amazon\EC2Launch\sysprep\unattend.xml"
   ```

일반화 단계

- EC2Launch v2는 컴퓨터 이름과 SID 같은 이미지 고유 정보 및 설정을 삭제합니다. 인스턴스가 도메인의 멤버인 경우에는 도메인에서 삭제됩니다. `unattend.xml` 응답 파일은 이 단계에 영향을 미치는 다음의 설정을 포함합니다.
  - PersistAllDeviceInstalls: 이 설정은 Windows 설치로 하여금 장치를 제거하고 재구성하지 못하도록 함으로써 이미지 준비 과정을 가속화하는데, 이는 Amazon AMI는 특정 드라이버가 실행되는 것을 요구하고 그 드라이버들을 재검색하는 데 시간이 걸리기 때문입니다.
  - DoNotCleanUpNonPresentDevices: 이 설정은 현재 존재하지 않는 장비들에 대한 플러그-앤-플레이 정보를 담고 있습니다.
- Windows Sysprep은 AMI를 생성하기 위한 준비를 하는 과정에서 OS를 종료합니다. 시스템은 새 인스턴스를 시작하거나 원본 인스턴스를 시작합니다.

특수화 단계

시스템은 컴퓨터 이름과 SID와 같은 OS 특정 요건을 생성합니다. 시스템은 또한 `unattend.xml` 응답 파일에 지정된 구성에 기반을 두어 다음 작업을 수행합니다.

- CopyProfile: Windows Sysprep을 구성해 내장된 관리자 프로파일을 비롯한 모든 사용자 프로파일을 삭제할 수 있습니다. 이 설정은 내장된 관리자 계정을 보유하고 있어서 계정에 대한 어떤 사용자 지정도 새 이미지로 전달됩니다. 기본 값은 `True`입니다.

  CopyProfile은 기본 프로파일을 기존의 로컬 관리자 프로파일로 바꿉니다. Windows Sysprep를 실행한 후 로그인한 모든 계정은 첫 로그인 시 해당 프로파일 및 콘텐츠의 사본을 받게 됩니다.

  새 이미지로 전달하고자 하는 사용자 프로파일에 대한 특정한 사용자 지정이 없다면 이 설정을 `False`로 변경하세요. Windows Sysprep은 모든 사용자 프로파일을 삭제하며, 이를 통해 시간과 디스크 공간을 절약합니다.
- TimeZone: 표준 시간대는 UTC(협정 세계시)로 기본 설정되어 있습니다.
- Synchronous command with order 1: 시스템은 다음 명령을 실행하여 관리자 계정을 활성화하고 암호 요건을 지정합니다.

```
net user Administrator /ACTIVE:YES /LOGONPASSWORDCHG:NO /EXPIRES:NEVER /
PASSWORDREQ:YES
```

- Synchronous command with order 2: 시스템은 관리자 암호를 암호화합니다. 이 보안 조치는 `setAdminAccount` 작업을 구성하지 않으면 Windows Sysprep이 완료된 후 인스턴스에 액세스할 수 없도록 설계되어 있습니다.

  시스템은 로컬 시작 에이전트 디렉터리(`C:\Program Files\Amazon\EC2Launch\`)에서 다음 명령을 실행합니다.

```
EC2Launch.exe internal randomize-password --username Administrator
```

- 원격 데스크톱 연결을 활성화하기 위해 시스템은 터미널 서버 `fDenyTSConnections` 레지스트리 키를 false로 설정합니다.

OOBE 단계

1. EC2Launch v2 응답 파일을 사용하여 다음과 같은 구성을 지정합니다.
   - `<InputLocale>en-US</InputLocale>`
   - `<SystemLocale>en-US</SystemLocale>`
   - `<UILanguage>en-US</UILanguage>`
   - `<UserLocale>en-US</UserLocale>`
   - `<HideEULAPage>true</HideEULAPage>`

- `<HideWirelessSetupInOOBE>true</HideWirelessSetupInOOBE>`
- `<ProtectYourPC>3</ProtectYourPC>`
- `<BluetoothTaskbarIconEnabled>false</BluetoothTaskbarIconEnabled>`
- `<TimeZone>UTC</TimeZone>`
- `<RegisteredOrganization>Amazon.com</RegisteredOrganization>`
- `<RegisteredOwner>EC2</RegisteredOwner>`

> **Note**
>
> 일반화 및 특수화 단계에서 EC2Launch v2는 OS의 상태를 모니터링합니다. EC2Launch v2는 OS가 Sysprep 단계에 있다는 것을 탐지하면 시스템 로그에 다음 메시지를 출력합니다. Windows is being configured. SysprepState=IMAGE_STATE_UNDEPLOYABLE

2. 시스템이 EC2Launch v2를 실행합니다.

Sysprep 이후

Windows Sysprep이 완료된 후 EC2Launch v2는 콘솔에 다음 메시지를 출력합니다.

```
Windows sysprep configuration complete.
```

그 다음에 EC2Launch v2는 다음 작업을 수행합니다.

1. `agent-config.yml` 파일의 내용을 읽고 구성된 작업을 실행합니다.
2. `preReady` 스테이지의 모든 작업을 실행합니다.
3. 실행이 완료된 후에는 `Windows is ready`라는 메시지를 인스턴스 시스템 로그에 전송합니다.
4. `PostReady` 스테이지의 모든 작업을 실행합니다.

EC2Launch v2에 대한 자세한 내용은 EC2Launch v2 에이전트를 사용하여 EC2 Windows 인스턴스 시작 중 태스크 수행 섹션을 참조하세요.

EC2Launch v2와 함께 Windows Sysprep 실행

Windows Sysprep과 EC2Launch v2를 이용해 표준화된 AMI를 생성하려면 다음 절차를 수행하세요.

1. Amazon EC2 콘솔에서 복제하려는 AMI를 찾습니다.

2. 실행을 시작해서 Windows 인스턴스에 연결합니다.
3. 그 AMI를 사용자 지정합니다.
4. Windows 시작 메뉴에서 Amazon EC2Launch 설정(Amazon EC2Launch settings)을 검색하고 선택합니다. Amazon EC2Launch 설정(EC2Launch settings) 대화 상자의 옵션 및 설정에 대한 자세한 내용은 Windows 인스턴스에 대한 EC2Launch v2 설정 구성 단원을 참조하세요.
5. Sysprep을 이용해 종료(Shutdown with Sysprep) 또는 Sysprep을 이용하지 않고 종료(Shutdown without Sysprep)를 선택합니다.

Windows Sysprep 실행 및 인스턴스 종료 여부 확인 요청을 받을 때 예를 클릭합니다. EC2Launch v2는 Windows Sysprep을 실행합니다. 그런 다음 인스턴스에서 로그오프되고 인스턴스가 종료됩니다. Amazon EC2 콘솔에서 [인스턴스(Instances)] 페이지를 보면 인스턴스 상태가 [Running]에서 [Stopping]으로 바뀐 다음 다시 [Stopped]로 바뀌는 것을 확인할 수 있습니다. 이 지점에서는 안전하게 현재 인스턴스에서 AMI를 생성할 수 있습니다.

다음 명령을 사용해서 명령줄에서 Windows Sysprep 도구를 수동으로 호출할 수 있습니다.

```
"%programfiles%\amazon\ec2launch\ec2launch.exe" sysprep --shutdown=true
```

EC2Launch와 함께 Windows Sysprep 사용

EC2Launch는 AMI에서 이미지 준비 프로세스를 자동화하고 보호하는 기본 응답 파일과 배치 파일을 Windows Sysrep에 제공합니다. 이 파일을 수정하는 것은 선택 사항입니다. 이 파일은 기본적으로 `C:\ProgramData\Amazon\EC2-Windows\Launch\Sysprep` 디렉터리에 있습니다.

> **Important**
>
> Windows Sysprep을 사용하여 인스턴스 백업을 생성하지 마세요. Windows Sysprep은 시스템에 특정한 정보를 제거합니다. 이 정보를 제거하면 인스턴스 백업에서 의도하지 않은 결과가 발생할 수 있습니다.

EC2Launch와 함께 Windows Sysprep 사용 주제

- Windows Sysprep에 대한 EC2Launch 응답 및 배치 파일
- EC2Launch와 함께 Windows Sysprep 실행
- 사용자 지정 AMI 시작 시 Server 2016 이후에 대한 메타데이터/KMS 경로 업데이트

Windows Sysprep에 대한 EC2Launch 응답 및 배치 파일

Windows Sysprep에 대한 EC2Launch 응답 파일과 배치 파일에는 다음 내용이 포함됩니다.

`Unattend.xml`

이 파일은 기본 응답 파일입니다. `SysprepInstance.ps1`을 실행하거나 사용자 인터페이스에서 ShutdownWithSysprep을 선택하는 경우 시스템이 이 파일에서 설정을 읽습니다.

`BeforeSysprep.cmd`

EC2Launch가 Windows Sysprep을 실행하기 전에 이 배치 파일이 명령을 실행하도록 사용자 지정합니다.

`SysprepSpecialize.cmd`

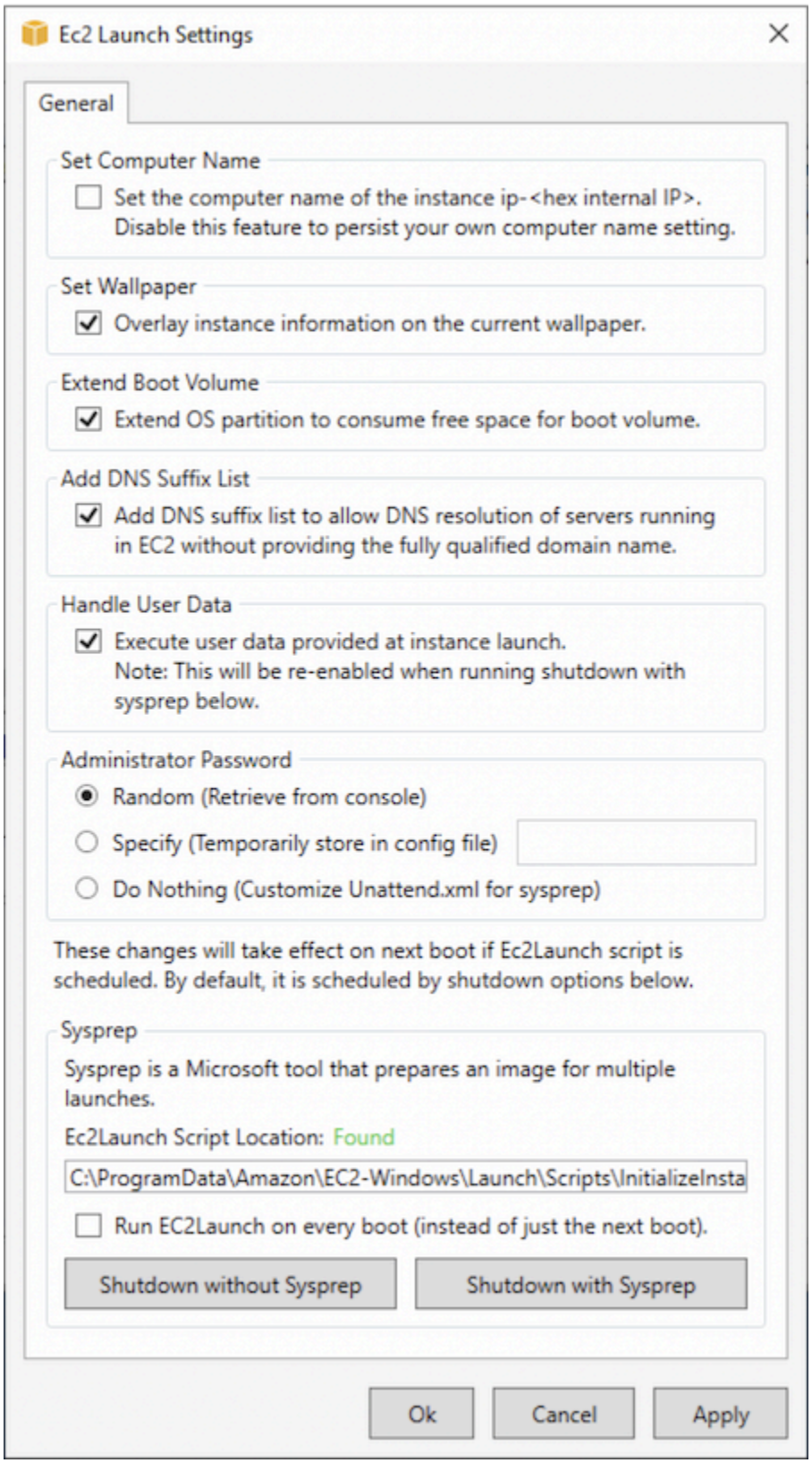
Windows Sysprep 특수화 단계 중에 명령을 실행하려면 이 배치 파일을 사용자 지정합니다.

EC2Launch와 함께 Windows Sysprep 실행

데스크톱 환경에서 Windows Server 2016 이후를 전체 설치할 때 EC2 Launch Settings 애플리케이션을 사용하여 EC2Launch와 함께 Windows Sysprep을 수동으로 실행할 수 있습니다.

EC2Launch Settings 애플리케이션을 사용하여 Windows Sysprep 실행

1. Amazon EC2 콘솔에서 Windows Server 2016 이후 AMI를 찾거나 생성합니다.
2. AMI에서 Windows 인스턴스를 시작합니다.
3. Windows 인스턴스에 연결하고 인스턴스를 사용자 지정합니다.
4. EC2LaunchSettings 애플리케이션을 검색하여 실행합니다. 기본적으로 `C:\ProgramData\Amazon\EC2-Windows\Launch\Settings` 디렉터리에 위치합니다.



5. 필요에 따라 옵션을 선택하거나 취소합니다. 이러한 설정은 `LaunchConfig.json` 파일에 저장되어 있습니다.

6. Administrator 암호에서 다음 중 하나를 수행합니다.

   - 임의(Random)를 선택합니다. EC2Launch는 암호를 생성하고 사용자의 키를 사용하여 암호를 암호화합니다. 인스턴스가 재부팅 또는 중지되었다가 시작된 경우 이 암호가 그대로 유지되도록 시스템은 인스턴스가 시작된 후 이 설정을 비활성화합니다.
   - 지정(Specify)을 선택하고 시스템 요구 사항을 충족하는 암호를 입력합니다. 암호는 `LaunchConfig.json`에 일반 텍스트로 저장되며 Windows Sysprep에서 관리자 암호를 설정한 후에 삭제됩니다. 지금 종료하면 암호는 지금 바로 설정됩니다. EC2Launch는 사용자의 키를 사용하여 암호를 암호화합니다.
   - DoNothing을 선택하고 `unattend.xml` 파일에 암호를 지정합니다. `unattend.xml`에 암호를 지정하지 않으면 관리자 계정이 비활성화됩니다.

7. Sysprep을 이용해 종료(Shutdown with Sysprep)를 선택합니다.

EC2Launch를 사용하여 Windows Sysprep을 수동으로 실행

1. Amazon EC2 콘솔에서 복제하려는 Windows Server 2016 이후 Datacenter 에디션 AMI를 찾거나 만듭니다.
2. 실행을 시작해서 Windows 인스턴스에 연결합니다.
3. 인스턴스를 사용자 지정합니다.
4. `LaunchConfig.json` 파일에서 설정을 지정합니다. 이 파일은 기본적으로 `C:\ProgramData\Amazon\EC2-Windows\Launch\Config` 디렉터리에 위치합니다.

   `adminPasswordType`에 대해 다음 값 중 하나를 지정할 수 있습니다.

   `Random`

   EC2Launch는 암호를 생성하고 사용자의 키를 사용하여 암호를 암호화합니다. 인스턴스가 재부팅 또는 중지되었다가 시작된 경우 이 암호가 그대로 유지되도록 시스템은 인스턴스가 시작된 후 이 설정을 비활성화합니다.

   `Specify`

   `adminPassword`에 지정한 암호가 EC2Launch에 사용됩니다. 암호가 시스템 요구 사항에 맞지 않으면 EC2Launch에서 임의의 암호를 대신 생성합니다. 암호는 `LaunchConfig.json`에 일반 텍스트로 저장되며 Windows Sysprep에서 관리자 암호를 설정한 후에 삭제됩니다. EC2Launch는 사용자의 키를 사용하여 암호를 암호화합니다.

`DoNothing`

`unattend.xml` 파일에 지정한 암호가 EC2Launch에 사용됩니다. `unattend.xml`에 암호를 지정하지 않으면 관리자 계정이 비활성화됩니다.

5. (선택 사항) `unattend.xml` 및 기타 구성 파일에서 설정을 지정합니다. 설치에 참가하려는 경우 이 파일을 변경할 필요가 없습니다. 파일은 기본적으로 `C:\ProgramData\Amazon\EC2-Windows\Launch\Sysprep` 디렉터리에 위치합니다.
6. Windows PowerShell에서 `./InitializeInstance.ps1 -Schedule`을 실행합니다. 스크립트는 기본적으로 `C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts` 디렉터리에 위치합니다. 이 스크립트는 다음 부팅 중에 초기화하도록 인스턴스를 예약합니다. 다음 단계에서 `SysprepInstance.ps1` 스크립트를 실행하기 전에 이 스크립트를 실행해야 합니다.
7. Windows PowerShell에서 `./SysprepInstance.ps1`을 실행합니다. 스크립트는 기본적으로 `C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts` 디렉터리에 위치합니다.

인스턴스에서 로그오프되고 인스턴스가 종료됩니다. Amazon EC2 콘솔에서 [인스턴스(Instances)] 페이지를 보면 인스턴스 상태가 [Running]에서 [Stopping]으로 바뀐 다음 [Stopped]로 바뀌는 것을 확인할 수 있습니다. 이 시점에서는 이 인스턴스에서 안전하게 AMI를 생성할 수 있습니다.

사용자 지정 AMI 시작 시 Server 2016 이후에 대한 메타데이터/KMS 경로 업데이트

사용자 지정 AMI를 시작할 때 Server 2016 이후에 대한 메타데이터/KMS 경로를 업데이트하려면 다음과 같이 하세요.

- EC2LaunchSettings GUI(C:\ProgramData\Amazon\EC2-Windows\Launch\Settings\Ec2LaunchSettings.exe)를 실행하고 Windows Sysprep을 사용하여 종료하는 옵션을 선택합니다.
- EC2LaunchSettings를 실행하고 Windows Sysprep 없이 종료한 다음 AMI를 생성합니다. 그러면 다음번 부팅할 때 EC2 시작 초기화 작업을 실행하도록 설정되며, 해당 인스턴스의 서브넷을 토대로 경로가 설정됩니다.
- PowerShell에서 AMI를 생성하기 전에 EC2 시작 초기화 작업을 수동으로 다시 예약합니다.

> **Important**
>
> 작업을 다시 예약하기 전에 기본 암호 재설정 동작을 적어 두세요.

- Windows 정품 인증 또는 인스턴스 메타데이터 오류와의 통신이 발생하는 실행 중인 인스턴스에서 경로를 업데이트하려면 "Windows를 정품 인증할 수 없음"을 참조하세요.

EC2Config와 함께 Windows Sysprep 사용

이 섹션에는 이미지가 준비될 때 EC2Config 서비스에서 수행하는 작업의 세부 정보가 포함됩니다. 또한 EC2Config 서비스와 함께 Windows Sysprep을 사용하여 표준화된 AMI를 생성하는 단계도 포함되어 있습니다.

EC2Config와 함께 Windows Sysprep 사용 주제

- Windows Sysprep 작업
- Sysprep 이후
- EC2Config 서비스와 함께 Windows Sysprep 실행하기

Windows Sysprep 작업

Windows Sysprep과 EC2Config 서비스는 이미지를 준비할 때 다음 작업을 수행합니다.

1. EC2 서비스 속성 대화 상자에서 Sysprep을 이용해 종료를 선택하면, 시스템은 ec2config.exe -sysprep 명령을 실행합니다.
2. EC2Config 서비스는 `BundleConfig.xml` 파일의 내용을 읽습니다. 기본적으로 이 파일은 `C:\Program Files\Amazon\Ec2ConfigService\Settings` 디렉터리에 위치합니다.

   `BundleConfig.xml` 파일에는 다음 설정이 포함되어 있습니다. 이 설정은 변경할 수 있습니다.

   - AutoSysprep: Windows Sysprep을 자동적으로 사용할지 여부를 지정합니다. Windows Sysprep을 EC2 Service Properties 대화 상자로부터 실행하는 경우에는 이 값을 변경할 필요가 없습니다. 기본 값은 `No`입니다.
   - SetRDPCertificate: 원격 데스크톱 서버에 대한 자체 서명된 인증서를 설정합니다. 이렇게 함으로써 원격 데스크톱 프로토콜(RDP)을 안정적으로 이용해 인스턴스에 연결할 수 있습니다. 새 인스턴스가 인증서를 사용해야 하는 경우 이 값을 `Yes`로 변경합니다. Windows Server 2012 인스턴스는 자신의 고유한 인증서를 생성할 수 있으므로 이 설정은 이 인스턴스에서는 사용되지 않습니다. 기본 값은 `No`입니다.
   - SetPasswordAfterSysprep: 새로 실행된 인스턴스에 무작위 암호를 설정하고 이를 사용자 실행 키로 암호화하고 암호화된 암호를 콘솔에 출력합니다. 새 인스턴스에 무작위의 암호화된 암호를 설정해서는 안 되는 경우에는 설정 값을 `No`로 변경합니다. 기본 값은 `Yes`입니다.
   - PreSysprepRunCmd: 실행할 명령의 위치 명령은 기본적으로 `C:\Program Files\Amazon\Ec2ConfigService\Scripts\BeforeSysprep.cmd` 디렉터리에 위치합니다.
3. 시스템이 `BeforeSysprep.cmd`를 실행합니다. 이 명령은 다음과 같은 레지스트리 키를 생성합니다.

```
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v
 fDenyTSConnections /t REG_DWORD /d 1 /f
```

레지스트리 키는 RDP 연결이 다시 활성화될 때까지 RDP 연결을 비활성화합니다. RDP 연결을 비활성화하는 것은 필수적인 보안 조치입니다. 왜냐하면 Windows Sysprep이 실행된 이후 첫 번째 부팅 세션 동안 RDP가 연결을 허용하고 관리자 암호가 비어 있는 짧은 시간이 있기 때문입니다.

4. EC2Config 서비스는 다음 명령을 실행하여 Windows Sysprep을 호출합니다.

```
sysprep.exe /unattend: "C:\Program Files\Amazon\Ec2ConfigService\sysprep2008.xml" /
oobe /generalize /shutdown
```

일반화 단계

- 이 도구는 컴퓨터 이름과 SID 같은 이미지 고유 정보 및 설정을 삭제합니다. 인스턴스가 도메인의 멤버인 경우에는 도메인에서 삭제됩니다. `sysprep2008.xml` 응답 파일은 이 단계에 영향을 미치는 다음의 설정을 포함합니다.
  - PersistAllDeviceInstalls: 이 설정은 Windows 설치로 하여금 장치를 제거하고 재구성하지 못하도록 함으로써 이미지 준비 과정을 가속화하는데, 이는 Amazon AMI는 특정 드라이버가 실행되는 것을 요구하고 그 드라이버들을 재검색하는 데 시간이 걸리기 때문입니다.
  - DoNotCleanUpNonPresentDevices: 이 설정은 현재 존재하지 않는 장비들에 대한 플러그-앤-플레이 정보를 담고 있습니다.
- Windows Sysprep은 AMI를 생성하기 위한 준비를 하는 과정에서 OS를 종료합니다. 시스템은 새 인스턴스를 시작하거나 원본 인스턴스를 시작합니다.

특수화 단계

시스템은 컴퓨터 이름과 SID와 같은 OS 특정 요건을 생성합니다. 시스템은 또한 sysprep2008.xml 응답 파일에 지정된 구성에 기반을 두어 다음 작업을 수행합니다.

- CopyProfile: Windows Sysprep을 구성해 내장된 관리자 프로파일을 비롯한 모든 사용자 프로파일을 삭제할 수 있습니다. 이 설정은 내장된 관리자 계정을 보유하고 있어서 계정에 대한 어떤 사용자 지정도 새 이미지로 전달됩니다. 기본값은 True입니다.

  CopyProfile은 기본 프로파일을 기존의 로컬 관리자 프로파일로 바꿉니다. Windows Sysprep를 실행한 후 로그인한 모든 계정은 첫 로그인 시 해당 프로파일 및 콘텐츠의 사본을 받게 됩니다.

새 이미지로 전달하고자 하는 사용자 프로파일에 대한 특정한 사용자 지정이 없다면 이 설정을 False로 변경하세요. Windows Sysprep은 모든 사용자 프로파일을 삭제하며, 이를 통해 시간과 디스크 공간을 절약합니다.

- TimeZone: 표준 시간대는 UTC(협정 세계시)로 기본 설정되어 있습니다.
- Synchronous command with order 1: 시스템은 다음 명령을 실행하여 관리자 계정을 활성화하고 암호 요건을 지정합니다.

  net user Administrator /ACTIVE:YES /LOGONPASSWORDCHG:NO /EXPIRES:NEVER /PASSWORDREQ:YES
- Synchronous command with order 2: 시스템은 관리자 암호를 암호화합니다. 이 보안 조치는 ec2setpassword 설정을 활성화하지 않으면 Windows Sysprep이 완료된 후 인스턴스에 액세스할 수 없도록 설계되어 있습니다.

  C:\Program Files\Amazon\Ec2ConfigService\ScramblePassword.exe" -u Administrator
- Synchronous command with order 3: 시스템은 다음 명령을 실행합니다.

  C:\Program Files\Amazon\Ec2ConfigService\Scripts\SysprepSpecializePhase.cmd

  이 명령은 다음과 같은 레지스트리 키를 추가하여 RDP를 재활성화합니다.

  reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f

OOBE 단계

1. 시스템은 EC2Config 응답 파일을 이용해 다음과 같은 구성을 지정합니다.
   - <InputLocale>en-US</InputLocale>
   - <SystemLocale>en-US</SystemLocale>
   - <UILanguage>en-US</UILanguage>
   - <UserLocale>en-US</UserLocale>
   - <HideEULAPage>true</HideEULAPage>
   - <HideWirelessSetupInOOBE>true</HideWirelessSetupInOOBE>
   - <NetworkLocation>Other</NetworkLocation>
   - <ProtectYourPC>3</ProtectYourPC>
   - <BluetoothTaskbarIconEnabled>false</BluetoothTaskbarIconEnabled>

- <TimeZone>UTC</TimeZone>
- <RegisteredOrganization>Amazon.com</RegisteredOrganization>
- <RegisteredOwner>Amazon</RegisteredOwner>

> **Note**
>
> 일반화 및 특수화 단계에서 EC2Config 서비스는 OS의 상태를 모니터링합니다. EC2Config는 OS가 Sysprep 단계에 있다는 것을 탐지하면 시스템 로그에 다음 메시지를 출력합니다. EC2ConfigMonitorState: Windows 0개가 구성 중입니다. SysprepState=IMAGE_STATE_UNDEPLOYABLE

2. OOBE 단계가 완료되면 `SetupComplete.cmd`에 있는 `C:\Windows\Setup\Scripts\SetupComplete.cmd`가 실행됩니다. 2015년 4월 이전의 Amazon 퍼블릭 AMI에서는 이 파일이 비어 있었고 이미지 상에서 아무 것도 실행하지 않았습니다. 2015년 4월 이후 발표된 퍼블릭 AMI의 경우 파일에 call "C:\Program Files\Amazon\Ec2ConfigService\Scripts\PostSysprep.cmd" 값이 포함됩니다.
3. `PostSysprep.cmd`가 실행되고 다음 작업이 수행됩니다.
   - 로컬 관리자 암호가 만료되지 않도록 설정합니다. 로컬 관리자 암호가 만료되면, 관리자가 로그인할 수 없을 수도 있습니다.
   - MSSQLServer 머신 이름(설치된 경우)를 설정하여 그 이름이 AMI와 동기화되도록 합니다.

Sysprep 이후

Windows Sysprep이 완료된 후 EC2Config 서비스는 콘솔에 다음 메시지를 출력합니다.

```
Windows sysprep configuration complete.
   Message: Sysprep Start
   Message: Sysprep End
```

그 다음에 EC2Config는 다음 작업을 수행합니다.

1. config.xml 파일의 내용을 읽고 활성화된 모든 플러그인을 나열합니다.
2. 모든 “Before Windows is ready” 플러그인을 동시에 실행합니다.
   - Ec2SetPassword
   - Ec2SetComputerName
   - Ec2InitializeDrives

- Ec2EventLog
- Ec2ConfigureRDP
- Ec2OutputRDPCert
- Ec2SetDriveLetter
- Ec2WindowsActivate
- Ec2DynamicBootVolumeSize

3. 실행이 완료된 후에는 “Windows is ready”라는 메시지를 인스턴스 시스템 로그에 전송합니다.
4. 모든 “After Windows is ready” 플러그인을 동시에 실행합니다.
   - Amazon CloudWatch Logs
   - UserData
   - AWS Systems Manager(Systems Manager)

Windows 플러그인에 대한 자세한 내용은 EC2Config 서비스를 사용하여 EC2 레거시 Windows 운영 체제 인스턴스를 시작하는 동안 태스크를 수행합니다. 단원을 참조하세요.

EC2Config 서비스와 함께 Windows Sysprep 실행하기

Windows Sysprep과 EC2Config 서비스를 이용해 표준화된 AMI를 생성하려면 다음 절차를 수행하세요.

1. Amazon EC2 콘솔에서 복사하고자 하는 AMI의 위치를 지정하거나 생성합니다.
2. 실행을 시작해서 Windows 인스턴스에 연결합니다.
3. 그 AMI를 사용자 지정합니다.
4. EC2Config 서비스 응답 파일에서 구성 설정을 다음과 같이 지정합니다.

   ```
   C:\Program Files\Amazon\Ec2ConfigService\sysprep2008.xml
   ```

5. Windows 시작 메뉴에서 모든 프로그램을 선택한 후 EC2ConfigService 설정을 클릭합니다.
6. Ec2 서비스 속성 대화 상자에서 이미지 탭을 선택합니다. Ec2 서비스 속성 대화 상자의 옵션 및 설정에 대한 자세한 내용은 Ec2 서비스 속성을 참조하세요.
7. 관리자 암호에 대한 옵션을 선택하고 Shutdown with Sysprep(Sysprep을 이용해 종료) 또는 Shutdown without Sysprep(Sysprep을 사용하지 않고 종료)을 선택합니다. EC2Config는 선택한 암호 옵션에 기반하는 설정 파일을 편집합니다.

- Random(무작위): EC2Config는 암호를 생성하고 이를 사용자의 키로 암호화한 다음, 암호화된 암호를 콘솔에 표시합니다. 첫 실행 후 이 설정을 비활성화하여 인스턴스가 재부팅되거나 중단되고 시작된 경우에도 해당 암호가 계속 유지되도록 합니다.
- 지정: 암호를 암호화되지 않은 형태(클리어 텍스트)로 Windows Sysprep 응답 파일에 저장합니다. Windows Sysprep은 다음에 실행될 때 관리자 암호를 설정합니다. 지금 종료하면 암호는 지금 바로 설정됩니다. 서비스가 다시 시작할 때 관리자 암호는 제거됩니다. 이 암호는 나중에 다시 확인할 수 없기 때문에 이를 꼭 기억해두세요.
- 기존 유지: Windows Sysprep이 실행 중일 때 또는 EC2Config가 재시작할 때, 관리자 계정에 대한 기존 암호를 변경하지 않습니다. 이 암호는 나중에 다시 확인할 수 없기 때문에 이를 꼭 기억해두세요.

8. 확인을 선택합니다.

Windows Sysprep 실행 및 인스턴스 종료 여부 확인 요청을 받을 때 예를 클릭합니다. EC2Config가 Windows Sysprep을 실행하는 것을 확인할 수 있습니다. 그 다음 인스턴스에서 로그아웃되고 인스턴스는 종료됩니다. Amazon EC2 콘솔의 [인스턴스(Instances)] 페이지를 보면 인스턴스 상태가 [Running]에서 [Stopping]로 바뀐 다음 마지막으로 [Stopped]로 바뀌는 것을 확인할 수 있습니다. 이 지점에서는 안전하게 현재 인스턴스에서 AMI를 생성할 수 있습니다.

다음 명령을 사용해서 명령줄에서 Windows Sysprep 도구를 수동으로 호출할 수 있습니다.

```
"%programfiles%\amazon\ec2configservice\"ec2config.exe -sysprep""
```

> **Note**
>
> CMD 셸이 이미 C:\Program Files\Amazon\EC2ConfigService\ 디렉터리에 있는 경우에는 명령에 큰 따옴표가 필요 없습니다.

그러나 `Ec2ConfigService\Settings` 폴더에 지정된 XML 파일 옵션이 올바른지 주의 깊게 확인할 필요가 있습니다. 이것이 올바르게 지정되지 않은 경우는 인스턴스에 연결하지 못할 수 있습니다. 설정 파일에 대한 자세한 내용은 EC2Config 설정 파일 단원을 참조하세요. 명령줄에서 Windows Sysprep을 구성한 후 실행하는 예제는 `Ec2ConfigService\Scripts\InstallUpdates.ps1`의 내용을 참조하세요.

Amazon EC2 AMI 복사

여러 리전에 일관된 Amazon EC2 인스턴스 구성이 필요한 경우 단일 Amazon Machine Image(AMI)를 템플릿으로 사용하여 모든 인스턴스를 시작할 수 있습니다. 그러나 AMIs는 리전별 리소스입니다. 특정 AWS 리전에서 인스턴스를 시작하려면 AMI가 해당 리전에 있어야 합니다. 따라서 여러 리전에서 동일한 AMI를 사용하려면 소스 리전에서 각 대상 리전으로 복사해야 합니다.

AMI 복사에 사용하는 방법은 동일한 파티션 내 리전 전체 또는 상이한 파티션 전체에 복사하는지에 따라 달라집니다.

- 리전 간 복사 - 동일한 파티션 내 리전 전체에 AMI를 복사합니다(예: 상용 파티션 내 리전 전체). 이 복사 방법은 이 주제에 설명되어 있습니다.
- 파티션 간 복사 - 하나의 파티션에서 다른 파티션으로 AMI를 복사합니다(예: 상용 파티션에서 AWS GovCloud (US) 파티션으로). 이 복사 방법에 대한 내용은 AMI 저장 및 복원 단원을 참조하세요.
- 계정 간 복사 - 다른 AWS 계정에서 본인의 AWS 계정와 공유한 AMI의 사본을 생성합니다. 이 복사 방법은 이 주제에 설명되어 있습니다.

리전 간 및 계정 간 AMI 복사의 복사 작업은 가능한 한 신속하게 처리됩니다. 완료 시간을 제어해야 하는 경우 AMI가 필요한 기간 내에 복사되도록 15분~48시간 범위로 완료 기간을 지정할 수 있습니다. 시간 기반 AMI 복사 작업에 대해서는 추가 요금이 부과됩니다. 자세한 내용은 Amazon EBS 사용 설명서의 Time-based copies를 참조하세요.

내용

- 고려 사항
- 비용
- Amazon EC2 AMI를 복사할 수 있는 권한 부여
- AMI 복사
- 대기 중인 AMI 복사 작업 중지
- Amazon EC2 AMI 복사 작동 방식

고려 사항

- AMI를 복사할 권한 - IAM 정책을 사용하여 사용자에게 AMI 복사 권한을 부여하거나 거부할 수 있습니다. 2024년 10월 28일부터 소스 AMI에 대한 `CopyImage` 작업을 위한 리소스 수준 권한을 지정할 수 있습니다. 대상 AMI에 대한 리소스 수준 권한은 이전처럼 사용할 수 있습니다.

- 시작 권한 및 Amazon S3 버킷 권한 - AWS에서는 소스 AMI의 시작 권한 또는 Amazon S3 버킷 권한을 새 AMI로 복사하지 않습니다. 복사 작업이 완료된 후 시작 권한 및 Amazon S3 버킷 권한을 새 AMI에 적용할 수 있습니다.
- 태그 - 소스 AMI에 연결한 사용자 정의 AMI 태그만 복사할 수 있습니다. 다른 AWS 계정이 첨부한 시스템 태그(접두사 `aws:` 포함) 및 사용자 정의 태그는 복사되지 않습니다. AMI를 복사할 때 대상 AMI 및 해당 지원 스냅샷에 새 태그를 연결할 수 있습니다.
- 시간 기반 AMI 복사 할당량 - 누적 스냅샷 복사 처리량 할당량에 도달하면 후속 시간 기반 AMI 복사 요청이 실패합니다. 자세한 내용은 Amazon EBS 사용 설명서의 Quotas for time-based copies를 참조하세요.

비용

완료 시간이 지정되지 않은 경우 AMI 복사 요금이 부과되지 않습니다. 그러나 시간 기반 AMI 복사 작업에는 추가 요금이 적용됩니다. 자세한 내용은 Amazon EBS 사용 설명서의 Time-based copies를 참조하세요.

표준 스토리지 및 데이터 전송 요금이 적용됩니다. EBS 지원 AMI를 복사하면 추가 EBS 스냅샷의 스토리지에 대한 요금이 발생합니다.

Amazon EC2 AMI를 복사할 수 있는 권한 부여

EBS 지원 AMI 또는 인스턴스 스토어 지원 AMI를 복사하려면 다음 IAM 권한이 필요합니다.

- `ec2:CopyImage` - AMI를 복사합니다. EBS 지원 AMI의 경우 AMI의 지원 스냅샷을 복사할 권한도 부여합니다.
- `ec2:CreateTags` - 대상 AMI에 태그를 지정합니다. EBS 지원 AMI의 경우 대상 AMI의 지원 스냅샷에 태그를 지정할 권한도 부여합니다.

인스턴스 스토어 지원 AMI를 복사하는 경우 다음과 같은 추가 IAM 권한이 필요합니다.

- `s3:CreateBucket` - 새 AMI의 대상 리전에 S3 버킷 생성
- `s3:GetBucketAcl` - 소스 버킷의 ACL 권한 읽기
- `s3:ListAllMyBuckets` - 대상 리전에서 AMI용 기존 S3 버킷 찾기
- `s3:GetObject` - 소스 버킷에서 객체 읽기
- `s3:PutObject` - 대상 버킷에 객체 쓰기
- `s3:PutObjectAcl` - 대상 버킷에서 새 객체에 대한 권한 쓰기

> **Note**
>
> 2024년 10월 28일부터 소스 AMI에 대한 CopyImage 작업을 위한 리소스 수준 권한을 지정할 수 있습니다. 대상 AMI에 대한 리소스 수준 권한은 이전처럼 사용할 수 있습니다. 자세한 내용은 서비스 승인 참조의 Amazon EC2에서 정의한 작업 아래 테이블의 CopyImage를 참조하세요.

EBS 지원 AMI를 복사하고 대상 AMI 및 스냅샷에 태그를 지정하기 위한 IAM 정책 예제

다음 정책 예제에서는 EBS 지원 AMI를 복사하고 대상 AMI 및 지원 스냅샷에 태그를 지정할 수 있는 권한을 부여합니다.

> **Note**
>
> 2024년 10월 28일부터 Resource 요소에 스냅샷을 지정할 수 있습니다. 자세한 내용은 서비스 승인 참조의 Amazon EC2에서 정의한 작업 아래 테이블의 CopyImage를 참조하세요.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "PermissionToCopyAllImages",
        "Effect": "Allow",
        "Action": [
            "ec2:CopyImage",
            "ec2:CreateTags"
        ],
        "Resource": [
            "arn:aws:ec2:*::image/*",
            "arn:aws:ec2:*::snapshot/*"
        ]
    }]
}
```

EBS 지원 AMI를 복사하지만 새 스냅샷에 태그 지정을 거부하는 IAM 정책 예제

ec2:CopySnapshot 권한은 ec2:CopyImage 권한을 받을 때 자동으로 부여됩니다. 새 백업 스냅샷에 태그를 지정할 수 있는 권한을 명시적으로 거부하여 ec2:CreateTags 작업의 Allow 효과를 재정의할 수 있습니다.

다음 정책 예제에서는 EBS 지원 AMI를 복사할 수 있는 권한은 부여하지만 대상 AMI의 새 지원 스냅샷에 태그 지정을 거부합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "ec2:CopyImage",
                "ec2:CreateTags"
            ],
            "Resource": [
                "arn:aws:ec2:*::image/*",
                "arn:aws:ec2:*::snapshot/*"
            ]
        },
        {
            "Effect": "Deny",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:::snapshot/*"
        }
    ]
}
```

인스턴스 스토어 지원 AMI를 복사하고 대상 AMI에 태그를 지정하는 IAM 정책 예제

다음 정책 예제에서는 지정된 소스 버킷의 모든 인스턴스 스토어 지원 AMI를 지정된 리전에 복사하고 대상 AMI에 태그를 지정할 수 있는 권한을 부여합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Sid": "PermissionToCopyAllImages",
            "Effect": "Allow",
            "Action": [
                "ec2:CopyImage",
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:*::image/*"
        },
        {
            "Effect": "Allow",
            "Action": "s3:ListAllMyBuckets",
```

```
            "Resource": [
                "arn:aws:s3:::*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:GetBucketAcl",
                "s3:PutObjectAcl",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::amis-for-account-in-region-hash"
            ]
        }
    ]
}
```

AMI 소스 버킷의 Amazon 리소스 이름(ARN)을 찾으려면 Amazon EC2 콘솔(https://console.aws.amazon.com/ec2/)을 열고 탐색 창에서 [AMI(AMIs)]를 선택한 다음 [소스(Source)] 열에서 버킷 이름을 찾습니다.

> **Note**
>
> `s3:CreateBucket` 권한은 인스턴스 스토어 지원 AMI를 개별 리전에 처음 복사할 때만 필요합니다. 이후 리전에 이미 생성된 Amazon S3 버킷은 향후 해당 리전에 복사하는 모든 AMIs를 저장하는 데 사용됩니다.

AMI 복사

AMI를 동일한 리전 또는 동일한 파티션 내 다른 리전에 복사할 수 있습니다. 본인이 소유하는 AMI 또는 다른 계정에서 본인과 공유한 AMI를 복사할 수 있습니다.

Console

AMI 복사

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 콘솔 탐색 모음에서 AMI가 들어 있는 리전을 선택합니다.
3. 탐색 창에서 AMI를 선택하여 리전에서 사용할 수 있는 AMI 목록을 표시합니다.
4. 복사하려는 AMI가 없으면 다른 필터를 선택합니다. 내 소유 AMI, 프라이빗 이미지, 퍼블릭 이미지, 비활성화된 이미지로 필터링할 수 있습니다.
5. 복사할 AMI를 선택하고 작업, AMI 복사를 선택합니다.
6. Amazon Machine Image(AMI) 복사 페이지에서 다음 정보를 지정합니다.

   a. AMI 복사 이름: 새 AMI의 이름. Amazon EC2는 AMI에 대한 세부 정보를 표시할 때 운영 체제 정보를 제공하지 않으므로 이름에 이 정보를 포함할 수 있습니다.

   b. AMI 복사 설명: 원본과 사본을 구분할 수 있도록 설명에는 기본적으로 원본 AMI에 대한 정보가 포함됩니다. 필요에 따라 이 설명을 수정할 수 있습니다.

   c. 대상 리전: AMI를 복사할 리전. 자세한 내용은 리전 간 복사 및 교차 계정 복사(을)를 참조하세요.

   d. 태그 복사: AMI를 복사할 때 사용자 정의 AMI 태그를 포함하려면 이 확인란을 선택합니다. 다른 AWS 계정이 첨부한 시스템 태그(접두사 `aws:` 포함) 및 사용자 정의 태그는 복사되지 않습니다.

   e. 시간 기반 복사: 다음과 같이 복사 작업을 특정 기간 내에 완료할지, 가능한 한 신속하게 완료할지 지정할 수 있습니다.

      - 특정 기간 내에 복사를 완료하는 방법:
        - 시간 기반 복사 활성화를 선택합니다.
        - 복사 작업에 허용되는 시간(15분씩 증분)을 완료 기간에 입력합니다. 완료 기간은 AMI에 연결된 모든 스냅샷에 적용됩니다.

          자세한 내용은 Amazon EBS 사용 설명서의 Time-based copies를 참조하세요.
      - 가능한 한 신속하게 복사를 완료하는 방법:
        - 시간 기반 복사 활성화를 선택되지 않은 상태로 놓아둡니다.

   f. (EBS 지원 AMI만 해당) AMI 사본의 EBS 스냅샷 암호화: 대상 스냅샷을 암호화하거나 다른 키를 사용하여 다시 암호화하려면 이 확인란을 선택합니다. 암호화가 기본적으로 활성

화되면 AMI 사본의 EBS 스냅샷 암호화 확인란이 선택되고 선택을 취소할 수 없습니다. 자세한 내용은 암호화 및 복사 섹션을 참조하세요.

g. (EBS 지원 AMI만 해당) KMS 키: 대상 스냅샷을 암호화하는 데 사용하는 KMS 키입니다.

h. 태그: 새 AMI 및 새 스냅샷에 동일한 태그를 지정하거나 다른 태그를 지정할 수 있습니다.

- 새 AMI 및 새 스냅샷에 동일한 태그를 지정하려면 이미지와 스냅샷을 함께 태그 지정을 선택합니다. 새 AMI 및 생성된 모든 스냅샷에 동일한 태그가 적용됩니다.
- 새 AMI 및 새 스냅샷에 다른 태그를 지정하려면 이미지와 스냅샷을 별도로 태그 지정을 선택합니다. 새 AMI 및 생성된 스냅샷에 서로 다른 태그가 적용됩니다. 그러나 생성된 모든 스냅샷의 태그는 동일하며 각 스냅샷에 다른 태그를 지정할 수 없습니다.

태그를 추가하려면 [태그 추가(Add tag)]를 선택하고 해당 태그에 대한 키와 값을 입력합니다. 각 태그에 대해 반복합니다.

i. AMI를 복사할 준비가 되면 AMI 복사를 선택합니다.

새 AMI의 초기 상태는 `Pending`입니다. 상태가 `Available`인 경우 AMI 복사 작업이 완료된 것입니다.

AWS CLI

AMI 복사

copy-image 명령을 사용합니다. 원본 리전과 대상 리전을 모두 지정해야 합니다. `--source-region` 파라미터를 사용하여 원본 리전을 지정합니다. AWS CLI에 대한 리전을 아직 구성하지 않은 경우 `--region` 파라미터를 사용하여 대상 리전을 지정할 수 있습니다.

```
aws ec2 copy-image \
    --source-image-id ami-0abcdef1234567890 \
    --source-region us-west-2 \
    --name my-ami \
    --region us-east-1
```

AMI 복사 중에 대상 스냅샷을 암호화하는 경우, `--encrypted` 및 `--kms-key-id` 파라미터를 추가로 지정해야 합니다.

PowerShell

AMI 복사

Copy-EC2Image cmdlet을 사용합니다. 원본 리전과 대상 리전을 모두 지정해야 합니다. -SourceRegion 파라미터를 사용하여 원본 리전을 지정합니다. -Region 파라미터 또는 Set-AWSDefaultRegion cmdlet을 사용하여 대상 리전을 지정할 수 있습니다.

```
Copy-EC2Image `
    -SourceImageId ami-0abcdef1234567890 `
    -SourceRegion us-west-2 `
    -Name my-ami `
    -Region us-east-1
```

AMI 복사 중에 대상 스냅샷을 암호화하는 경우, -Encrypted 및 -KmsKeyId 파라미터를 추가로 지정해야 합니다.

대기 중인 AMI 복사 작업 중지

다음 절차에 따라 보류 중인 AMI 복사를 중지할 수 있습니다.

Console

AMI 복사 작업을 중지하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 모음의 리전 선택기에서 대상 리전을 선택합니다.
3. 탐색 창에서 AMI를 선택합니다.
4. 복사를 중지할 AMI를 선택하고 작업, AMI 등록 해제를 차례로 선택합니다.
5. 확인을 요청하면 AMI 등록 취소(Deregister AMI)를 선택합니다.

AWS CLI

AMI 복사 작업을 중지하려면

deregister-image 명령을 사용합니다.

```
aws ec2 deregister-image --image-id ami-0abcdef1234567890
```

PowerShell

AMI 복사 작업을 중지하려면

Unregister-EC2Image cmdlet을 사용합니다.

```
Unregister-EC2Image -ImageId ami-0abcdef1234567890
```

Amazon EC2 AMI 복사 작동 방식

소스 AMI를 복사하면 대상 AMI라고 하는 동일하지만 별개의 새 AMI가 생성됩니다. 대상 AMI에는 고유한 AMI ID가 있습니다. 대상 AMI에 영향을 미치지 않고 원본 AMI를 변경하거나 다시 등록할 수 있습니다. 반대의 경우도 마찬가지입니다.

EBS 지원 AMI의 경우 각 지원 스냅샷이 동일하지만 별개의 대상 스냅샷으로 복사됩니다. AMI를 새 리전으로 복사하는 경우 스냅샷은 증분이 아닌 전체 복사본이 됩니다. 암호화되지 않은 백업 스냅샷을 암호화하거나 새 KMS 키로 암호화하는 경우 스냅샷은 전체(비증분) 복사본입니다. 이후에 AMI 복사 작업을 수행하면 백업 스냅샷의 증분 복사본이 생성됩니다.

내용

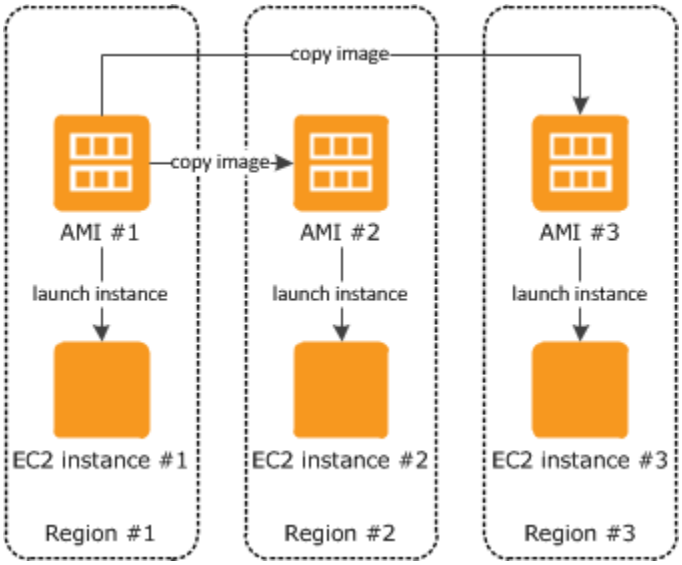
- 리전 간 복사
- 교차 계정 복사
- 시간 기반 AMI 복사 작업
- 암호화 및 복사

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 소스 AMI 및 다른 리전에 복사된 2개의 AMI 간 관계와 각각에서 시작된 EC2 인스턴스를 보여줍니다. 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. 예를 들면 `amis-for-123456789012-in-us-east-2-yhjmxvp6`입니다.

전제 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. 그렇게 하지 않으면 대상 리전의 새 AMI에서 시작된 인스턴스가 여전히 소스 리전의 리소스를 사용하여 성능과 비용에 영향을 줄 수 있습니다.

제한 사항

- 대상 리전은 300개의 동시 AMI 복사 작업으로 제한됩니다. 이는 시간 기반 AMI 복사 작업에도 적용됩니다.
- 반가상화(PV) AMI를 지원하지 않는 리전으로 PV AMI를 복사할 수 없습니다. 자세한 내용은 가상화 유형 섹션을 참조하세요.

교차 계정 복사

다른 AWS 계정의 AMI이 자신의 AWS 계정과 공유되는 경우 해당 공유 AMI를 복사할 수 있습니다. 이것을 교차 계정 액세스라고 합니다. 공유되는 AMI가 소스 AMI입니다. 소스 AMI를 복사할 때 새 AMI를 생성합니다. 새 AMI를 종종 대상 AMI라고 합니다.

AMI 비용

- 공유 AMI의 경우 공유 AMI의 계정에 리전의 스토리지 비용이 청구됩니다.
- 자신의 계정과 공유된 AMI를 복사하는 경우 계정에 생성된 대상 AMI의 소유자가 됩니다.
  - 소스 AMI의 소유자에게는 표준 Amazon EBS 또는 Amazon S3 전송 요금이 청구됩니다.
  - 대상 리전의 대상 AMI에 대한 스토리지 요금이 청구됩니다.

리소스 권한

다른 계정에서 공유된 AMI를 복사하려면 AMI 자체가 아니라 AMI를 지원하는 스토리지에 대한 읽기 권한을 소스 AMI 소유자가 부여해야 합니다. 스토리지는 연결된 EBS 스냅샷(Amazon EBS 지원 AMI의 경우) 또는 연결된 S3 버킷(인스턴스 스토어 지원 AMI) 중 하나입니다. 공유 AMI에 암호화된 스냅샷이 있는 경우, 소유자는 해당 키를 사용자와 공유해야 합니다. 리소스 권한 부여에 대한 자세한 내용은 EBS 스냅샷의 경우 Amazon EBS 사용 설명서의 Amazon EBS 스냅샷을 다른 AWS 계정와 공유, S3 버킷의 경우 Amazon S3 사용 설명서의 Amazon S3의 Identity and Access Management를 참조하세요.

> **Note**
>
> 소스 AMI에 연결된 태그는 계정 간에 대상 AMI로 복사되지 않습니다.

시간 기반 AMI 복사 작업

연결된 단일 스냅샷이 있는 EBS 지원 AMI에 대해 시간 기반 AMI 복사 작업을 시작하는 경우 개별 시간 기반 스냅샷 복사 작업과 동일한 방식으로 작동하며 동일한 처리량 제한이 적용됩니다.

연결된 여러 스냅샷이 있는 EBS 지원 AMI에 대해 시간 기반 AMI 복사 작업을 시작하는 경우 동시 시간 기반 스냅샷 복사 작업과 동일한 방식으로 작동하며 동일한 처리량 제한이 적용됩니다. 연결된 각 스냅샷은 별도의 스냅샷 복사 요청을 생성하며, 각 요청은 누적 스냅샷 복사 처리량 할당량에 포함됩니다. 사용자가 지정한 완료 기간은 연결된 각 스냅샷에 적용됩니다.

자세한 내용은 Amazon EBS 사용 설명서의 Time-based copies를 참조하세요.

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 AMI는 스냅샷을 사용하지 않기 때문에 복사를 사용하여 암호화 상태를 변경할 수 없습니다.

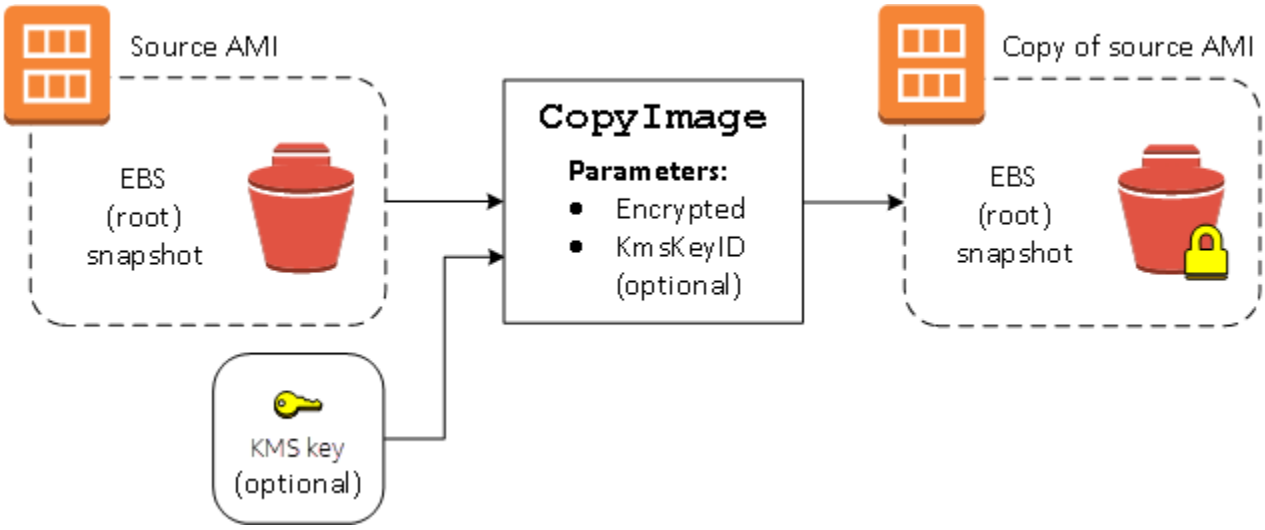
암호화 파라미터를 지정하지 않고 AMI를 복사하면 지원 스냅샷은 기본적으로 원래 암호화 상태로 복사됩니다. 따라서 소스 AMI가 암호화되지 않은 스냅샷으로 지원되는 경우 결과 대상 스냅샷도 암호화되지 않습니다. 마찬가지로 소스 AMI의 스냅샷이 암호화된 경우 결과 대상 스냅샷도 동일한 AWS KMS 키로 암호화됩니다. 여러 스냅샷이 지원하는 AMI의 경우, 각 대상 스냅샷에서 해당 소스 스냅샷의 암호화 상태가 유지됩니다.

AMI 복사 중에 대상 지원 스냅샷의 암호화 상태를 변경하려면 암호화 파라미터를 지정할 수 있습니다. 다음 예제는 대상 AMI의 암호화 상태를 변경하기 위해 CopyImage 작업으로 암호화 파라미터가 지정되는 기본이 아닌 사례를 보여줍니다.

암호화되지 않은 원본 AMI을(를) 암호화된 대상 AMI(으)로 복사

이 시나리오에서는 암호화되지 않은 루트 스냅샷으로 지원되는 AMI가 암호화된 루트 스냅샷이 있는 AMI로 복사됩니다. CopyImage 작업은 고객 관리형 키를 포함하여 2개의 암호화 파라미터를 사용하여 호출됩니다. 따라서 루트 스냅샷의 암호화 상태가 변경되므로 대상 AMI는 소스 스냅샷과 동일한 데이터를 포함하는 루트 스냅샷에 의해 지원되지만 지정된 키를 사용하여 암호화됩니다. 두 AMI 모두의 스냅샷에 대한 스토리지 비용과 각 AMI에서 시작되는 인스턴스에 대한 비용이 발생합니다.

> **Note**
>
> 기본적으로 암호화를 활성화하는 경우 AMI의 모든 스냅샷에서 `Encrypted` 파라미터를 `true`로 설정하는 것과 효과가 동일합니다.



`Encrypted` 파라미터를 설정하면 이 인스턴스에 대한 단일 스냅샷이 암호화됩니다. `KmsKeyId` 파라미터를 지정하지 않으면 스냅샷 복사본을 암호화하는 데 기본 고객 관리형 키가 사용됩니다.

암호화된 스냅샷을 포함하는 AMIs 복사에 대한 자세한 내용은 EBS-backed AMI에서 암호화 사용 섹션을 참조하세요.

S3를 사용하여 AMI 저장 및 복원

Amazon Machine Image(AMI)를 Amazon S3 버킷에 저장하고 AMI를 다른 S3 버킷으로 복사한 다음 S3 버킷에서 복원할 수 있습니다. S3 버킷을 사용하여 AMI를 저장하고 복원하면 한 AWS 파티션의 AMI를 다른 파티션으로, 예를 들어 기본 상용 파티션에서 AWS GovCloud (US) 파티션으로 복사할 수 있습니다. 또한 AMI를 S3 버킷에 저장하면 AMI의 아카이브 복사본을 만들 수도 있습니다.

S3를 사용하여 AMI를 저장하고 복원할 때 지원되는 API는 `CreateStoreImageTask`, `DescribeStoreImageTasks` 및 `CreateRestoreImageTask`입니다.

`CopyImage`는 AWS 파티션 내에서 AMI를 복사할 때 권장되는 API입니다. 그러나 `CopyImage`를 사용하여 AMI를 다른 파티션으로 복사할 수는 없습니다.

AWS 파티션에 대한 자세한 내용은 IAM 사용 설명서의 Amazon 리소스 이름(ARN) 페이지에서 *partition*을 참조하세요.

> **Warning**
>
> AWS 파티션 또는 AWS 리전 간에 데이터를 이동할 때는 해당하는 정부 규제 및 데이터 레지던시 요구 사항을 포함한 모든 관련 법률 및 비즈니스 요구 사항을 준수해야 합니다.

내용

- 사용 사례
- 제한 사항
- 비용
- AMI 저장 및 복원 작동 방식
- 이미지 저장 태스크 생성

사용 사례

저장 및 복원 API를 사용하여 다음을 수행합니다.

- AWS 파티션 간 AMI 복사
- AMI의 아카이브 복사본 만들기

AWS 파티션 간 AMI 복사

S3 버킷을 사용하여 AMI를 저장하고 복원하면 한 파티션에서 다른 AWS 파티션으로 또는 한 리전에서 다른 AWS 리전으로 AMI를 복사할 수 있습니다. 다음 예에서는 기본 상용 파티션의 AMI를 AWS GovCloud (US) 파티션으로, 특히 `us-east-2` 리전에서 `us-gov-east-1` 리전으로 복사합니다.

한 파티션에서 다른 파티션으로 AMI를 복사하려면 다음 단계를 수행합니다.

- `CreateStoreImageTask`를 사용하여 현재 리전의 S3 버킷에 AMI를 저장합니다. 이 예에서 S3 버킷은 `us-east-2`에 있습니다.
- `DescribeStoreImageTasks`를 사용하여 저장 태스크의 진행률을 모니터링합니다. 태스크가 완료되면 객체가 S3 버킷에 표시됩니다.
- 선택한 절차를 사용하여 저장된 AMI 객체를 대상 파티션의 S3 버킷에 복사합니다. 이 예에서 S3 버킷은 `us-gov-east-1`에 있습니다.

> **Note**
>
> 각 파티션마다 다른 AWS 자격 증명이 필요하기 때문에 S3 객체를 한 파티션에서 다른 파티션으로 직접 복사할 수 없습니다. 파티션에서 S3 객체를 복사하는 프로세스는 이 설명서에서 다루지 않습니다. AWS에서는 다음 복사 프로세스를 예로 제공하지만 사용자는 보안 요구 사항을 충족하는 복사 프로세스를 사용해야 합니다.
>
> - 파티션에서 하나의 AMI를 복사하기 위한 복사 프로세스는 다음과 같이 간단할 수 있습니다. 즉, 원본 버킷에서 중간 호스트(예: EC2 인스턴스 또는 랩톱)로 객체를 다운로드한 후 중간 호스트에서 대상 버킷으로 객체를 업로드합니다. 프로세스의 각 단계에 대해 파티션에 대한 AWS 자격 증명을 사용합니다.
> - 지속적인 사용을 위해서는 S3 멀티파트 다운로드 및 업로드를 사용하여 복사본을 관리하는 애플리케이션을 개발하는 것이 좋습니다.

- `CreateRestoreImageTask`를 사용하여 대상 파티션의 S3 버킷에서 AMI를 복원합니다. 이 예에서 S3 버킷은 `us-gov-east-1`에 있습니다.
- AMI를 설명하여 복원 태스크의 진행률을 모니터링하면 사용 가능한 상태가 될 때를 확인할 수 있습니다. 스냅샷을 설명하여 복원된 AMI를 구성하는 스냅샷의 진행 비율을 모니터링할 수도 있습니다.

AMI의 아카이브 복사본 만들기

S3 버킷에 AMI를 저장하여 아카이브 복사본을 만들 수 있습니다. AMI는 S3의 단일 객체로 압축되며 모든 AMI 메타데이터(공유 정보 제외)는 저장된 AMI의 일부로 보존됩니다. AMI 데이터는 스토리지 프로세스의 일부로 압축됩니다. 쉽게 압축할 수 있는 데이터가 포함된 AMI의 경우 S3의 객체 크기가 작아집니다. 비용을 절감하려면 저렴한 S3 스토리지 계층을 사용할 수 있습니다. 자세한 내용은 Amazon S3 스토리지 클래스 및 Amazon S3 요금을 참조하세요.

제한 사항

- AMI를 저장하려면 AWS 계정이(가) AMI와 해당 스냅샷을 소유하거나 AMI와 해당 스냅샷을 계정과 직접 공유해야 합니다. 공개 공유 전용인 AMI는 저장할 수 없습니다.
- 이러한 API는 EBS-backed AMI를 저장할 때만 사용할 수 있습니다.
- 반가상화(PV) AMI는 지원되지 않습니다.
- 저장할 수 있는 AMI의 크기(압축 전)는 5,000GB로 제한됩니다.
- 이미지 저장 요청의 할당량: 진행 중인 저장 작업(스냅샷 데이터) 1,200GB.
- 이미지 복원 요청의 할당량: 진행 중인 복원 작업(스냅샷 데이터) 600GB.

- 저장 태스크 중에는 스냅샷이 삭제되지 않아야 하며 저장을 수행하는 IAM 보안 주체가 스냅샷에 액세스할 수 있어야 합니다. 그렇지 않으면 저장 프로세스가 실패합니다.
- 동일한 S3 버킷에 AMI의 여러 복사본을 생성할 수 없습니다.
- S3 버킷에 저장된 AMI를 원래 AMI ID로 복원할 수 없습니다. AMI 별칭 지정을 사용하여 이 문제를 완화할 수 있습니다.
- 현재, 저장 및 복원 API는 AWS Command Line Interface, AWS SDK 및 Amazon EC2 API를 사용하는 경우에만 지원됩니다. Amazon EC2 콘솔을 사용하여 AMI를 저장하고 복원할 수는 없습니다.

비용

S3를 사용하여 AMI를 저장 및 복원하는 경우 저장 및 복원 API에 사용되는 서비스와 데이터 전송 요금이 부과됩니다. 이 API는 S3 및 EBS Direct API(API 내부에서 스냅샷 데이터에 액세스하는 데 사용됨)를 사용합니다. 자세한 내용은 Amazon S3 요금 및 Amazon EBS 요금을 참조하세요.

AMI 저장 및 복원 작동 방식

S3를 사용하여 AMI를 저장하고 복원하려면 다음 API를 사용합니다.

- `CreateStoreImageTask` – S3 버킷에 AMI 저장
- `DescribeStoreImageTasks` – AMI 저장 태스크의 진행률 제공
- `CreateRestoreImageTask` – S3 버킷에서 AMI 복원

API의 작동 방식

- CreateStoreImageTask
- DescribeStoreImageTasks
- CreateRestoreImageTask
- 파일 경로

CreateStoreImageTask

`CreateStoreImageTask` API는 AMI를 S3 버킷에 단일 객체로 저장합니다.

이 API는 AMI 및 해당 스냅샷의 모든 데이터를 읽은 다음 S3 멀티파트 업로드를 사용하여 데이털르 S3 객체에 저장하는 태스크를 생성합니다. 이 API는 리전에 특정되지 않은 AMI 메타데이터 대부분을 포함한 AMI의 모든 구성 요소와 AMI에 포함된 모든 EBS 스냅샷을 가져와서 S3의 단일 객체로 압축합니다.

S3에 사용되는 공간의 양을 줄이기 위해 업로드 프로세스의 일부로 데이터가 압축되므로 S3의 객체는 AMI의 스냅샷 크기 합계보다 작을 수 있습니다.

이 API를 호출하는 계정에 AMI와 스냅샷의 태그가 표시되는 경우 해당 태그가 보존됩니다.

S3의 객체는 AMI와 동일한 ID를 가지지만 `.bin` 확장자가 사용됩니다. 또한 AMI 이름, AMI 설명, AMI 등록 날짜, AMI 소유자 계정 및 저장 작업의 타임스탬프와 같은 데이터는 S3 객체에 S3 메타데이터 태그로 저장됩니다.

태스크를 완료하는 데 걸리는 시간은 AMI의 크기에 따라 다릅니다. 또한 태스크가 대기열에 있기 때문에 진행 중인 다른 태스크의 수에 따라서도 달라집니다. `DescribeStoreImageTasks` API를 직접적으로 호출하여 태스크의 진행을 추적할 수 있습니다.

진행 중인 모든 AMI 크기의 합계는 계정당 1,200GB의 EBS 스냅샷 데이터로 제한됩니다. 진행 중인 태스크가 한도보다 작아질 때까지 추가 태스크 생성이 거부됩니다. 예를 들어 200GB의 스냅샷 데이터가 포함된 AMI와 400GB의 스냅샷 데이터가 포함된 또 다른 AMI가 현재 저장 중인 경우 진행 중인 합계가 한도보다 작은 600GB이므로 다른 요청이 수락됩니다. 그러나 1,200GB의 스냅샷 데이터가 포함된 단일 AMI가 현재 저장 중인 경우에는 태스크가 완료될 때까지 추가 태스크가 거부됩니다.

DescribeStoreImageTasks

`DescribeStoreImageTasks` API는 AMI 저장 태스크의 진행 상황을 설명합니다. 지정된 AMI에 대한 태스크를 설명할 수 있습니다. AMI를 지정하지 않으면 지난 31일 동안 처리된 모든 이미지 저장 태스크의 페이지 매김 목록이 표시됩니다.

각 AMI 태스크에 대한 응답에는 태스크가 `InProgress`, `Completed` 또는 `Failed`인지 표시됩니다. `InProgress` 태스크의 경우 응답에 예상 진행률이 백분율로 표시됩니다.

태스크는 역순으로 나열됩니다.

현재로서는 이전 월의 태스크만 볼 수 있습니다.

CreateRestoreImageTask

`CreateRestoreImageTask` API는 `CreateStoreImageTask` 요청을 사용하여 이전에 생성된 S3 객체에서 AMI를 복원하는 태스크를 시작합니다.

복원 태스크는 저장 태스크가 수행된 동일한 리전 또는 다른 리전에서 수행될 수 있습니다.

AMI 객체를 복원할 S3 버킷은 복원 태스크가 요청된 동일한 리전에 있어야 합니다. AMI는 이 리전에 복원됩니다.

AMI는 저장된 AMI의 값에 해당하는 이름, 설명 및 블록 디바이스 매핑과 같은 메타데이터와 함께 복원됩니다. 이름은 이 계정의 리전 내 AMI에 대해 고유해야 합니다. 이름을 제공하지 않으면 새 AMI에 원래 AMI와 동일한 이름이 지정됩니다. 복원 프로세스 시 생성된 새 AMI ID가 AMI에 지정됩니다.

AMI 복원 태스크를 완료하는 데 걸리는 시간은 AMI의 크기에 따라 다릅니다. 또한 태스크가 대기열에 있기 때문에 진행 중인 다른 태스크의 수에 따라서도 달라집니다. AMI를 설명(describe-images)하거나 EBS 스냅샷을 설명(describe-snapshots)하여 태스크 진행률을 볼 수 있습니다. 태스크가 실패하면 AMI와 스냅샷이 실패 상태로 전환됩니다.

진행 중인 모든 AMI의 크기 합계는 계정당 300GB(복원 후 크기 기준)의 EBS 스냅샷 데이터로 제한됩니다. 진행 중인 태스크가 한도보다 작아질 때까지 추가 태스크 생성이 거부됩니다.

파일 경로

다음과 같은 방식으로 AMI를 저장 및 복원할 때 파일 경로를 사용할 수 있습니다.

- S3에 AMI를 저장할 때 파일 경로를 버킷 이름에 추가할 수 있습니다. 내부적으로 시스템은 경로를 버킷 이름과 분리한 다음 AMI 저장을 위해 생성된 객체 키에 경로를 추가합니다. 전체 객체 경로가 API 호출의 응답에 표시됩니다.
- AMI를 복원할 때 객체 키 파라미터가 제공되므로 경로를 객체 키 값의 시작 부분에 추가할 수 있습니다.

예시: 파일 경로가 추가된 버킷 이름

AMI를 저장할 때 버킷 이름 뒤에 파일 경로를 지정합니다.

```
amzn-s3-demo-bucket/path1/path2
```

결과 객체 키는 다음과 같습니다.

```
path1/path2/ami-0abcdef1234567890.bin
```

AMI를 복원할 때 버킷 이름과 객체 키를 모두 지정할 수 있습니다. 예시는 the section called “이미지 저장 태스크 생성” 섹션을 참조하세요.

이미지 저장 태스크 생성

S3 버킷에 AMI를 저장하면 이미지 저장 태스크가 생성됩니다. 이미지 저장 태스크를 사용하여 프로세스의 진행 상황과 결과를 모니터링할 수 있습니다.

내용

- AMI 보안
- S3를 사용하여 AMI를 저장하고 복원하기 위한 권한
- 이미지 저장 태스크 생성
- 이미지 복원 태스크 생성

AMI 보안

S3 버킷에 AMI의 콘텐츠를 보호할 수 있는 충분한 보안으로 구성되어 있고 AMI 객체가 버킷에 남아 있는 동안 보안이 유지되도록 하는 것이 중요합니다. 불가능하다면 이러한 API를 사용하지 않는 것이 좋습니다. S3 버킷에 대한 퍼블릭 액세스가 허용되지 않는지 확인합니다. 필수는 아니지만, AMI가 저장되는 S3 버킷에 대해 서버 측 암호화를 활성화하는 것이 좋습니다.

S3 버킷에 대한 적절한 보안 설정을 설정하는 방법에 대한 자세한 내용은 다음 보안 항목을 참조하세요.

- Amazon S3 스토리지에 대한 퍼블릭 액세스 차단
- Amazon S3 버킷에 대한 기본 서버 측 암호화 동작 설정
- AWS Config 규칙 s3-bucket-ssl-requests-only 준수를 위해 어떤 S3 버킷 정책을 사용할 수 있습니까?
- Amazon S3서버 액세스 로깅 활성화

AMI 스냅샷이 S3 객체에 복사되면 데이터는 TLS 연결을 통해 복사됩니다. AMI를 암호화된 스냅샷과 함께 저장할 수 있지만 스냅샷은 저장 프로세스의 일부로 복호화됩니다.

S3를 사용하여 AMI를 저장하고 복원하기 위한 권한

IAM 보안 주체가 Amazon S3를 사용하여 AMI를 저장 또는 복원하는 경우 이 보안 주체에 필요한 권한을 부여해야 합니다.

다음 예시 정책에는 IAM 보안 주체가 저장 및 복원 태스크를 수행하도록 허용하는 데 필요한 모든 작업이 포함되어 있습니다.

또한 보안 주체에게 특정 리소스에만 액세스할 수 있는 권한을 부여하는 IAM 정책을 생성할 수도 있습니다. 더 많은 예시 정책은 IAM 사용 설명서의 AWS 리소스에 대한 액세스 관리를 참조하세요.

> **Note**
>
> AMI를 구성하는 스냅샷이 암호화된 경우 또는 계정이 기본적으로 암호화에 대해 활성화된 경우 IAM 보안 주체에 KMS 키를 사용할 권한이 있어야 합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:DeleteObject",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:PutObject",
                "s3:PutObjectTagging",
                "s3:AbortMultipartUpload",
                "ebs:CompleteSnapshot",
                "ebs:GetSnapshotBlock",
                "ebs:ListChangedBlocks",
                "ebs:ListSnapshotBlocks",
                "ebs:PutSnapshotBlock",
                "ebs:StartSnapshot",
                "ec2:CreateStoreImageTask",
                "ec2:DescribeStoreImageTasks",
                "ec2:CreateRestoreImageTask",
                "ec2:GetEbsEncryptionByDefault",
                "ec2:DescribeTags",
                "ec2:CreateTags"
            ],
            "Resource": "*"
        }
    ]
}
```

이미지 저장 태스크 생성

S3 버킷에 AMI를 저장하려면 먼저 이미지 저장 태스크를 생성합니다. 태스크를 완료하는 데 걸리는 시간은 AMI의 크기에 따라 다릅니다. 성공하거나 실패할 때까지 태스크의 진행 상황을 추적할 수 있습니다.

AWS CLI

이미지 저장 태스크를 생성하려면 다음을 수행하세요.

create-store-image-task 명령을 사용합니다.

```
aws ec2 create-store-image-task \
    --image-id ami-0abcdef1234567890 \
    --bucket amzn-s3-demo-bucket
```

출력의 예시는 다음과 같습니다.

```
{
  "ObjectKey": "ami-0abcdef1234567890.bin"
}
```

이미지 저장 태스크의 진행 상황을 설명하려면 다음을 수행하세요.

describe-store-image-tasks 명령을 사용합니다.

```
aws ec2 describe-store-image-tasks \
    --image-ids ami-0abcdef1234567890 \
    --query StoreImageTaskResults[].StoreTaskState \
    --output text
```

출력의 예시는 다음과 같습니다.

```
InProgress
```

PowerShell

이미지 저장 태스크를 생성하려면 다음을 수행하세요.

New-EC2StoreImageTask cmdlet을 사용합니다.

```
New-EC2StoreImageTask `
    -ImageId ami-0abcdef1234567890 `
    -Bucket amzn-s3-demo-bucket
```

출력의 예시는 다음과 같습니다.

```
ObjectKey         : ami-0abcdef1234567890.bin
```

이미지 저장 태스크의 진행 상황을 설명하려면 다음을 수행하세요.

Get-EC2StoreImageTask cmdlet을 사용합니다.

```
(Get-EC2StoreImageTask -ImageId ami-0abcdef1234567890).StoreTaskState
```

출력의 예시는 다음과 같습니다.

```
InProgress
```

이미지 복원 태스크 생성

복원된 AMI의 이름을 지정해야 합니다. 이름은 이 계정의 리전 내 AMI에 대해 고유해야 합니다. 복원된 AMI에 새 AMI ID가 지정됩니다.

AWS CLI

이미지 복원 태스크를 생성하려면 다음을 수행하세요.

create-restore-image-task 명령을 사용합니다.

```
aws ec2 create-restore-image-task \
    --object-key ami-0abcdef1234567890.bin \
    --bucket amzn-s3-demo-bucket \
    --name "my-restored-ami"
```

출력의 예시는 다음과 같습니다.

```
{
   "ImageId": "ami-1234567890abcdef0"
}
```

PowerShell

이미지 복원 태스크를 생성하려면 다음을 수행하세요.

New-EC2RestoreImageTask cmdlet을 사용합니다.

```
New-EC2RestoreImageTask `
    -ObjectKey ami-0abcdef1234567890.bin `
    -Bucket amzn-s3-demo-bucket `
```

```
    -Name "my-restored-ami"
```

출력의 예제는 다음과 같습니다.

```
ImageId         : ami-1234567890abcdef0
```

새 Amazon EC2 AMI를 생성하는 데 사용된 소스 AMI 식별

새 AMI에서 소스 AMI ID(콘솔) 또는 `sourceImageId`(AWS CLI) 필드를 확인하여 새 AMI를 생성하는 데 사용된 소스 AMI를 식별할 수 있습니다. 이 필드에는 새 AMI를 생성하기 위해 복사된 원본 AMI의 ID가 포함되어 있습니다.

소스 AMI 리전(콘솔) 또는 `sourceImageRegion`(AWS CLI) 필드를 확인하여 소스 AMI가 있었던 리전을 찾을 수도 있습니다.

고려 사항

- 소스 AMI의 ID 및 리전은 다음 API 명령을 사용하여 AMI가 생성된 경우에만 표시됩니다.
  - CreateImage - 인스턴스에서 AMI를 생성합니다.
  - CopyImage - 동일한 리전 내에서 또는 동일한 파티션의 리전 간에 AMI를 복사합니다.
  - CreateRestoreImageTask - AMI를 다른 파티션에 복사합니다.

  AMI가 다른 API 명령으로 생성된 경우 소스 AMI의 ID와 리전이 표시되지 않습니다.
- 일부 오래된 AMI의 경우 소스 AMI의 ID와 리전을 사용하지 못할 수 있습니다.
- 소스 AMI가 삭제된 경우 소스 AMI의 ID 및 리전 필드는 새 AMI에 계속 표시됩니다.
- CreateImage(인스턴스에서 AMI 생성)를 사용하여 생성된 AMI의 경우 소스 AMI ID는 인스턴스를 시작하는 데 사용된 AMI의 ID입니다.

Console

AMI를 생성하는 데 사용된 소스 AMI를 식별하려면 다음과 같이 하세요.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 AMI를 선택합니다.
3. 세부 정보를 볼 AMI를 선택합니다.

   소스 AMI 정보는 소스 AMI ID 및 소스 AMI 리전 필드에 표시됩니다.

AWS CLI

AMI를 생성하는 데 사용된 소스 AMI를 식별하려면 다음과 같이 하세요.

describe-images 명령을 사용하고 AMI의 ID와 리전을 지정합니다.

```
aws ec2 describe-images \
    --region us-east-1 \
    --image-ids ami-0abcdef1234567890 \
    --query "Images[].{ID:SourceImageId,Region:SourceImageRegion}"
```

출력의 예시는 다음과 같습니다.

```
[
    {
        "ID": "ami-0abcdef1234567890",
        "Region": "us-west-2"
    }
}
```

PowerShell

AMI를 생성하는 데 사용된 소스 AMI를 식별하려면 다음과 같이 하세요.

Get-EC2Image cmdlet을 사용합니다.

```
Get-EC2Image -ImageId ami-0abcdef1234567890 | Select SourceImageId,
 SourceImageRegion
```

출력의 예시는 다음과 같습니다.

```
SourceImageId           SourceImageRegion
-------------           -----------------
ami-0abcdef1234567890 us-west-2
```

Amazon EC2 AMI를 마지막으로 사용한 시점 확인

Amazon EC2는 AMI가 인스턴스를 시작하는 데 마지막으로 사용된 날짜와 시간을 자동으로 추적합니다. 오랫동안 인스턴스를 시작하는 데 사용되지 않은 AMI가 있는 경우 해당 AMI가 등록 취소 또는 사용 중단에 적합한지 고려해 보세요.

고려 사항

- AMI를 사용하여 인스턴스를 시작하면 24시간이 지나서 사용량이 보고됩니다.
- 마지막 실행 시간을 확인하려면 AMI의 소유자여야 합니다.
- AMI 사용 데이터는 2017년 4월부터 사용할 수 있습니다.

Console

AMI의 마지막 시작 시간 보기

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 AMI를 선택합니다.
3. 필터 표시줄에서 내 소유를 선택합니다.
4. AMI에 대한 확인란을 선택합니다.
5. 세부 정보 탭에서 마지막 시작 시간을 찾습니다.

AWS CLI

AMI를 설명하여 마지막 시작 시간을 보는 방법

describe-images 명령을 사용합니다. `LastLaunchedTime`이 없는 경우 본인이 AMI를 소유하는지 확인합니다.

```
aws ec2 describe-images \
    --image-id ami-0abcdef1234567890 \
    --query Images[].LastLaunchedTime \
    --output text
```

출력의 예시는 다음과 같습니다.

```
2025-02-17T20:22:19Z
```

AMI의 마지막 시작 시간 속성을 보는 방법

describe-image-attribute 명령을 사용합니다. 본인이 지정된 AMI의 소유자여야 합니다.

```
aws ec2 describe-image-attribute \
```

```
    --image-id ami-0abcdef1234567890 \
    --attribute lastLaunchedTime \
    --query LastLaunchedTime.Value \
    --output text
```

출력의 예시는 다음과 같습니다.

```
2025-02-17T20:22:19Z
```

PowerShell

AMI를 설명하여 마지막 시작 시간을 보는 방법

Get-EC2Image cmdlet을 사용합니다. `LastLaunchedTime`이 없는 경우 본인이 AMI를 소유하는지 확인합니다.

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).LastLaunchedTime
```

출력의 예시는 다음과 같습니다.

```
2025-02-17T20:22:19Z
```

AMI의 마지막 시작 시간 속성을 보는 방법

Get-EC2ImageAttribute cmdlet을 사용합니다. 본인이 지정된 AMI의 소유자여야 합니다.

```
(Get-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute LastLaunchedTime).LastLaunchedTime
```

출력의 예제는 다음과 같습니다.

```
2025-02-17T20:22:19Z
```

Amazon EC2 AMI 사용 중단

AMI가 만료되어 사용되지 않도록 할 날짜를 지정하여 사용 중지할 수 있습니다. AMI가 만료되는 시점을 나타내는 AMI의 향후 사용 중지 날짜를 지정할 수도 있습니다. 예를 들어 더 이상 적극적으로 유지

관리되지 않는 AMI를 사용 중지하거나, 새 버전으로 대체된 AMI를 사용 중지할 수 있습니다. 기본적으로 사용 중지된 AMI는 AMI 목록에 표시되지 않으므로 새 사용자가 오래된 AMI를 사용할 수 없습니다. 그러나 기존 사용자, 그리고 시작 템플릿 및 Auto Scaling 그룹과 같은 시작 서비스는 해당 ID를 지정하여 사용 중지된 AMI 계속 사용할 수 있습니다. 사용자 및 서비스가 사용할 수 없도록 AMI 삭제하려면 등록 취소해야 합니다.

AMI I가 사용 중지된 후:

- AMI 사용자의 경우, 해당 ID를 지정하거나 사용 중지된 AMI가 표시되도록 지정하지 않는 한 사용 중지된 AMI는 DescribeImages API 호출에 표시되지 않습니다. AMI 소유자에게는 사용 중지된 AMI가 DescribeImages API 호출에 계속 표시됩니다.
- AMI 사용자의 경우 사용 중지된 AMI를 EC2 콘솔을 통해 선택할 수 없습니다. 예를 들어 사용 중지된 AMI는 시작 인스턴스 마법사의 AMI 카탈로그에 표시되지 않습니다. AMI 소유자는 EC2 콘솔에서 사용 중지된 AMI를 계속 볼 수 있습니다.
- AMI 사용자의 경우 사용 중지된 AMI 의 ID를 알고 있으면 API, CLI 또는 SDK를 통해 계속 사용 중지된 AMI를 사용하여 인스턴스를 시작할 수 있습니다.
- 시작 템플릿 및 Auto Scaling 그룹과 같은 시작 서비스는 사용 중지된 AMI를 계속 참조할 수 있습니다.
- 이후에 사용 중지된 AMI 사용하여 시작된 EC2 인스턴스는 영향을 받지 않으며 중지, 시작 및 재부팅할 수 있습니다.

프라이빗 및 퍼블릭 AMI를 모두 사용 중지할 수 있습니다.

내용

- 비용
- 고려 사항
- AMI 사용 중지
- 사용 중지된 AMI 설명
- AMI 사용 중단 취소

비용

AMI를 사용 중지하더라도 해당 AMI는 삭제되지 않습니다. AMI 소유자는 AMI의 스냅샷에 대한 비용을 계속 지불합니다. 스냅샷에 대한 지불을 중지하려면 AMI 소유자가 AMI를 등록 취소하여 삭제해야 합니다.

고려 사항

- AMI를 사용 중지하려면 해당 AMI의 소유자여야 합니다.
- 최근에 인스턴스를 시작하는 데 사용된 적이 없는 AMI는 사용을 중단하거나 등록을 해제(deregistering)하는 것이 좋습니다. 자세한 내용은 the section called “AMI를 마지막으로 사용한 시점 확인” 섹션을 참조하세요.
- Amazon Data Lifecycle Manager EBS 지원 AMI 정책을 생성하여 EBS 지원 AMI의 사용 중단을 자동화할 수 있습니다. 자세한 내용은 AMI 수명 주기 정책 생성을 참조하세요.
- 기본적으로 모든 퍼블릭 AMI의 사용 중단 날짜가 AMI 생성 날짜로부터 2년으로 설정됩니다. 사용 중단 날짜를 2년보다 짧게 설정할 수 있습니다. 사용 중단 날짜를 취소하거나 중단 날짜를 연장하려면 특정 AWS 계정과 공유만을 통해 AMI를 프라이빗으로 설정해야 합니다.

AMI 사용 중지

특정 날짜 및 시간에 AMI를 사용 중지할 수 있습니다. AMI의 소유자여야 합니다.

사용 중단 날짜의 상한은 지금부터 10년입니다. 단, 상한이 생성 날짜로부터 2년인 퍼블릭 AMI는 예외입니다. 과거의 날짜는 지정할 수 없습니다.

Console

특정 날짜에 AMI 사용 중단

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 왼쪽 탐색기에서 AMI(AMIs)를 선택합니다.
3. 필터 표시줄에서 내 소유를 선택합니다.
4. AMI를 선택하고 작업(Actions), AMI 사용 중단 관리(Manage AMI Deprecation)를 선택합니다. 여러 AMI를 선택하여 한 번에 여러 AMI의 사용 중단 날짜를 동일하게 설정할 수 있습니다.
5. 활성화 확인란을 선택한 다음 사용 중단 날짜 및 시간을 입력합니다.
6. 저장을 선택합니다.

AWS CLI

특정 날짜에 AMI 사용 중단

enable-image-deprecation 명령을 사용합니다. 초 단위로 값을 지정하면 Amazon EC2가 초를 가장 가까운 분으로 반올림합니다.

```
aws ec2 enable-image-deprecation \
    --image-id ami-0abcdef1234567890 \
    --deprecate-at "2025-04-15T13:17:12.000Z"
```

PowerShell

특정 날짜에 AMI 사용 중단

Enable-EC2ImageDeprecation cmdlet을 사용합니다. 초 단위로 값을 지정하면 Amazon EC2가 초를 가장 가까운 분으로 반올림합니다.

```
Enable-EC2ImageDeprecation `
    -ImageId ami-0abcdef1234567890 `
    -DeprecateAt 2025-04-15T13:17:12.000Z
```

사용 중지된 AMI 설명

AMI의 사용 중단 날짜 및 시간을 보고 사용 중단 날짜를 기준으로 AMI를 필터링할 수 있습니다.

Console

AMI의 사용 중단 날짜 보기

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 왼쪽 탐색기에서 AMI(AMIs)를 선택한 후 AMI를 선택합니다.
3. 사용 중단 시간 필드(AMI 옆에 있는 확인란을 선택한 경우 세부 정보 탭에 있음)를 확인합니다. 이 필드에는 AMI의 사용 중단 날짜 및 시간이 표시됩니다. 필드가 비어 있는 경우 AMI는 사용 중단되지 않습니다.

사용 중단 날짜별로 AMI 필터링

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 왼쪽 탐색기에서 AMI(AMIs)를 선택합니다.
3. 필터 표시줄에서 내 소유(Owned by me) 또는 프라이빗 이미지(Private images)를 선택합니다(프라이빗 이미지에는 사용자와 공유되는 AMI와 사용자가 소유한 AMI가 포함됨).
4. 검색 창에 **Deprecation time**을 입력한 다음(문자를 입력하면 사용 중단 시간(Deprecation time) 필터가 나타남) 연산자와 날짜 및 시간을 선택합니다.

AWS CLI

모든 AMI를 설명하는 경우 본인이 AMI 사용자 또는 AMI 소유자인지에 따라 결과가 다릅니다.

- AMI 사용자 - 기본적으로 모든 AMI를 설명할 때 본인과 공유되지만 본인이 소유하지 않는 사용 중단된 AMI는 제외됩니다. 사용 중단된 AMI를 결과에 포함하려면 `--include-deprecated` 옵션을 지정합니다.
- AMI 소유자 - 모든 AMI를 설명할 때 사용 중단된 AMI를 포함하여 본인이 소유하는 모든 AMI가 포함됩니다. 본인이 소유하는 사용 중단된 AMI는 `--no-include-deprecated` 옵션을 사용하여 제외할 수 없습니다.

계정에 대한 모든 AMI를 설명할 때 사용 중단된 AMI를 포함하는 방법

다음 describe-images 명령을 사용합니다.

```
aws ec2 describe-images
    --owners 123456789012 \
    --include-deprecated
```

계정의 사용 중단된 AMI를 설명하는 방법

다음 describe-images 명령을 사용합니다.

```
aws ec2 describe-images \
    --owners self \
    --query "Images[?DeprecationTime!=null].ImageId" \
    --output text
```

출력의 예시는 다음과 같습니다.

```
ami-0abcdef1234567890
```

AMI의 사용 중단 날짜 설명

다음 describe-images 명령을 사용합니다. `DeprecationTime`이 출력에 표시되지 않는 경우 AMI는 사용되지 않거나 향후 날짜에 사용되지 않도록 설정되어 있습니다.

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query Images[].DeprecationTime \
```

```
    --output text
```

출력의 예시는 다음과 같습니다.

```
2025-05-01T00:00:00.000Z
```

PowerShell

계정의 사용 중단된 AMI를 나열하는 방법

Get-EC2Image cmdlet을 사용합니다.

```
(Get-EC2Image -Owner self | Where-Object {$_.DeprecationTime -ne $null}).ImageId
```

출력의 예시는 다음과 같습니다.

```
ami-0abcdef1234567890
```

AMI의 사용 중단 날짜 설명

Get-EC2Image cmdlet을 사용합니다. `DeprecationTime`이 출력에 표시되지 않는 경우 AMI는 사용되지 않거나 향후 날짜에 사용되지 않도록 설정되어 있습니다.

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).DeprecationTime
```

출력의 예시는 다음과 같습니다.

```
2025-05-01T00:00:00.000Z
```

AMI 사용 중단 취소

사용 중단 날짜 및 시간이 제거되므로 AMI 사용 중단은 취소할 수 있습니다. 이 절차를 수행하려면 AMI 소유자여야 합니다.

Console

AMI의 사용 중단 취소

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

2. 왼쪽 탐색기에서 AMI(AMIs)를 선택합니다.
3. 필터 표시줄에서 내 소유를 선택합니다.
4. AMI를 선택하고 작업(Actions), AMI 사용 중단 관리(Manage AMI Deprecation)를 선택합니다. 여러 AMI를 선택하여 여러 AMI의 사용 중단을 한 번에 취소할 수 있습니다.
5. 활성화 확인란을 선택 취소한 다음 저장을 선택합니다.

AWS CLI

AMI의 사용 중단 취소

다음 disable-image-deprecation 명령을 사용합니다.

```
aws ec2 disable-image-deprecation --image-id ami-0abcdef1234567890
```

PowerShell

AMI의 사용 중단 취소

Disable-EC2ImageDeprecation cmdlet을 사용합니다.

```
Disable-EC2ImageDeprecation -ImageId ami-0abcdef1234567890
```

Amazon EC2 AMI 비활성화

인스턴시 시작에 사용되지 않도록 AMI를 비활성화할 수 있습니다. 비활성화된 AMI에서는 새 인스턴스를 시작할 수 없습니다. 비활성화된 AMI를 다시 활성화하면 인스턴스 시작에 다시 사용할 수 있습니다.

프라이빗 및 퍼블릭 AMI를 모두 비활성화할 수 있습니다.

거의 사용하지 않지만 장기간 유지해야 하는 비활성화된 EBS 지원 AMI의 스토리지 비용을 줄이려면 관련 스냅샷을 아카이빙할 수 있습니다. 자세한 내용은 Amazon EBS 사용 설명서의 Archive Amazon EBS snapshots를 참조하세요.

내용

- AMI 비활성화 작동 방식
- 비용
- 사전 조건

- 필수 IAM 권한
- AMI 비활성화
- 비활성화된 AMI 설명
- 비활성화된 AMI 다시 활성화

AMI 비활성화 작동 방식

> **Warning**
>
> AMI를 비활성화하면 시작 권한이 모두 제거됩니다.

AMI가 비활성화된 경우:

- AMI의 상태가 `disabled`로 변경됩니다.
- 비활성화된 AMI는 공유될 수 없습니다. AMI가 퍼블릭 상태였거나 이전에 공유된 경우 프라이빗으로 전환됩니다. AMI가 AWS 계정, 조직 또는 조직 단위와 공유된 경우 비활성화된 이들은 AMI에 대한 액세스 권한을 잃게 됩니다.
- 비활성화된 AMI는 기본적으로 DescribeImages API 호출에 표시되지 않습니다.
- 비활성화된 AMI는 내 소유 콘솔 필터 아래에 표시되지 않습니다. 비활성화된 AMI를 찾으려면 비활성화된 이미지 콘솔 필터를 사용합니다.
- 비활성화된 AMI는 EC2 콘솔에서 인스턴스 시작 선택이 불가능합니다. 예를 들어 비활성화된 AMI는 시작 인스턴스 마법사의 AMI 카탈로그나 시작 템플릿을 생성할 때 표시되지 않습니다.
- 시작 템플릿 및 Auto Scaling 그룹과 같은 시작 서비스는 비활성화된 AMI를 계속 참조할 수 있습니다. 비활성화된 AMI에서 후속 인스턴스를 시작하면 실패하므로 사용 가능한 AMI만 참조하도록 시작 템플릿과 Auto Scaling 그룹을 업데이트하는 것이 좋습니다.
- 이후에 비활성화된 AMI 사용하여 시작되었던 EC2 인스턴스는 영향을 받지 않으며 중지, 시작 및 재부팅할 수 있습니다.
- 비활성화된 AMI와 관련된 스냅샷은 삭제할 수 없습니다. 연결된 스냅샷을 삭제하려고 하면 `snapshot is currently in use` 오류가 발생합니다.

AMI가 다시 활성화된 경우:

- AMI의 상태가 `available`로 변경되고, 이를 사용하여 인스턴스를 시작할 수 있습니다.
- AMI가 공유될 수 있습니다.

- 비활성화되었을 때 액세스 권한을 잃은 AWS 계정, 조직 및 조직 단위는 자동으로 액세스 권한을 획득하고, AMI가 다시 공유될 수 있습니다.

비용

AMI를 비활성화하더라도 해당 AMI는 삭제되지 않습니다. AMI가 EBS 지원 AMI인 경우 AMI의 EBS 스냅샷에 대한 요금은 계속 지불합니다. AMI를 유지하려는 경우 스냅샷을 보관하여 스토리지 비용을 줄일 수 있습니다. 자세한 내용은 Amazon EBS 사용 설명서의 Archive Amazon EBS snapshots를 참조하세요. AMI와 스냅샷을 보관하지 않으려면 AMI 등록을 취소하고 스냅샷을 삭제해야 합니다. 자세한 내용은 AMI 등록 취소 섹션을 참조하세요.

사전 조건

AMI를 비활성화하거나 다시 활성화하려면 AMI의 소유자여야 합니다.

필수 IAM 권한

AMI를 비활성화하고 다시 활성화하려면 다음 IAM 권한이 있어야 합니다.

- `ec2:DisableImage`
- `ec2:EnableImage`

AMI 비활성화

EC2 콘솔 또는 AWS Command Line Interface(AWS CLI) 사용을 통해 AMI를 비활성화할 수 있습니다. 이 절차를 수행하려면 AMI 소유자여야 합니다.

Console

AMI 비활성화

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 AMI를 선택합니다.
3. 필터 표시줄에서 내 소유를 선택합니다.
4. AMI를 선택한 다음 작업, AMI 비활성화를 선택합니다. 한 번에 여러 개의 AMI를 선택하여 비활성화할 수 있습니다.
5. AMI 비활성화 창에서 AMI 비활성화를 선택합니다.

AWS CLI

AMI 비활성화

다음 disable-image 명령을 사용합니다.

```
aws ec2 disable-image --image-id ami-0abcdef1234567890
```

PowerShell

AMI 비활성화

Disable-EC2Image cmdlet을 사용합니다.

```
Disable-EC2Image -ImageId ami-0abcdef1234567890
```

비활성화된 AMI 설명

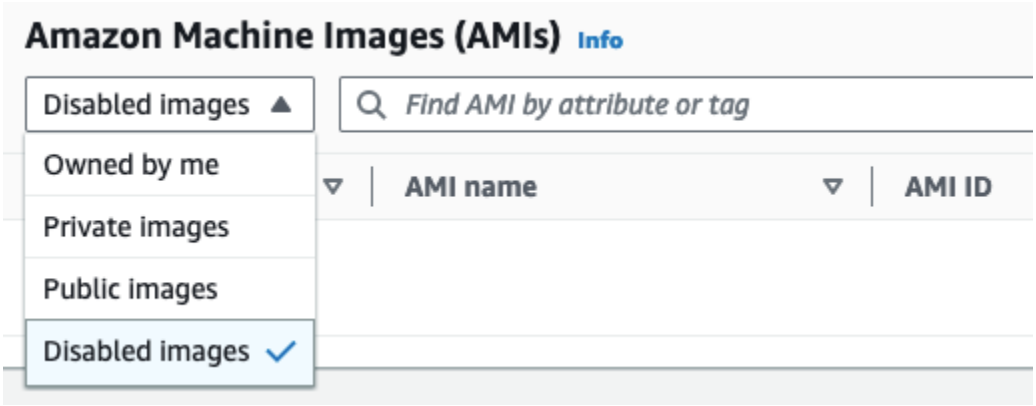
EC2 콘솔 또는 AWS CLI 사용을 통해 비활성화된 AMI를 볼 수 있습니다.

비활성화된 AMI를 보려면 AMI 소유자여야 합니다. 비활성화된 AMI는 프라이빗으로 전환되므로 소유자가 아닌 경우 비활성화된 AMI를 볼 수 없습니다.

Console

비활성화된 AMI 보기

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 AMI를 선택합니다.
3. 필터 막대에서 비활성화된 이미지를 선택합니다.



AWS CLI

기본적으로 모든 AMI를 설명하는 경우 비활성화된 AMI는 결과에 포함되지 않습니다. 비활성화된 AMI를 결과에 포함하려면 `--include-disabled` 옵션을 지정합니다. AMI가 비활성화된 경우 AMI의 `State` 필드는 `disabled`입니다.

계정에 대한 모든 AMI를 설명할 때 비활성화된 AMI를 포함하는 방법

다음 describe-images 명령을 사용합니다.

```
aws ec2 describe-images \
    --owners 123456789012 \
    --include-disabled
```

계정에 대한 비활성화된 AMI를 나열하는 방법

다음 describe-images 명령을 사용합니다.

```
aws ec2 describe-images \
    --owners self \
    --include-disabled \
    --filters Name=state,Values=disabled \
    --query Images[].ImageId \
    --output text
```

출력의 예시는 다음과 같습니다.

```
ami-0abcdef1234567890
```

AMI 상태를 설명하는 방법

다음 describe-images 명령을 사용합니다. `DeprecationTime`이 출력에 표시되지 않는 경우 AMI는 사용되지 않거나 향후 날짜에 사용되지 않도록 설정되어 있습니다.

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query Images[].State \
    --output text
```

출력의 예시는 다음과 같습니다.

```
disabled
```

PowerShell

기본적으로 모든 AMI를 설명하는 경우 비활성화된 AMI는 결과에 포함되지 않습니다. 비활성화된 AMI를 결과에 포함하려면 `-IncludeDisabled` 파라미터를 지정합니다. AMI가 비활성화된 경우 AMI의 `State` 필드는 `disabled`입니다.

계정에 대한 비활성화된 AMI를 나열하는 방법

Get-EC2Image cmdlet을 사용합니다.

```
(Get-EC2Image `
    -Owner self `
    -IncludeDisabled $true | Where-Object {$_.State -eq "disabled"}).ImageId
```

출력의 예시는 다음과 같습니다.

```
ami-0abcdef1234567890
```

AMI 상태를 설명하는 방법

Get-EC2Image cmdlet을 사용합니다.

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).State.Value
```

출력의 예시는 다음과 같습니다.

```
disabled
```

비활성화된 AMI 다시 활성화

비활성화된 AMI를 다시 활성화할 수 있습니다. 이 절차를 수행하려면 AMI 소유자여야 합니다.

Console

비활성화된 AMI 다시 활성화

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

3. 필터 막대에서 비활성화된 이미지를 선택합니다.
4. AMI를 선택한 다음 작업, AMI 활성화를 선택합니다. 여러 AMI를 선택하여 여러 AMI를 한 번에 다시 활성화할 수 있습니다.
5. AMI 활성화 창에서 활성화를 선택합니다.

AWS CLI

비활성화된 AMI 다시 활성화

다음 enable-image 명령을 사용합니다.

```
aws ec2 enable-image --image-id ami-0abcdef1234567890
```

PowerShell

비활성화된 AMI 다시 활성화

Enable-EC2Image cmdlet을 사용합니다.

```
Enable-EC2Image -ImageId ami-0abcdef1234567890
```

Amazon EC2 AMI 등록 취소

AMI를 등록 취소하면 Amazon EC2에서 AMI를 영구적으로 삭제합니다. AMI의 등록을 취소한 이후에는 새 인스턴스를 시작하기 위해 해당 AMI를 사용하는 것은 불가능합니다. AMI 사용을 마쳤으면 AMI의 등록을 취소할 수 있습니다.

우발적이거나 악의적인 AMI 등록 취소를 방지하기 위해 등록 취소 보호를 활성화할 수 있습니다. 실수로 EBS 지원 AMI를 등록 취소한 경우 영구 삭제되기 전에 허용된 기간에 복원하는 경우에만 휴지통을 사용하여 복원할 수 있습니다.

AMI 등록을 취소할 때 연결된 스냅샷을 선택적으로 동시에 삭제할 수 있습니다. 다만 여러 AMI와 연결된 스냅샷은 삭제하도록 지정된 경우에도 삭제되지 않지만 AMI는 여전히 등록 취소됩니다. 삭제되지 않은 스냅샷에서는 계속 스토리지 비용이 발생합니다.

AMI를 시작한 인스턴스에서는 AMI를 등록 취소해도 영향을 받지 않습니다. 이러한 인스턴스는 계속 사용할 수 있습니다. 기본적으로 AMI 등록 취소는 AMI 생성 프로세스 중에 생성된 스냅샷에 영향을 주지 않습니다. 이러한 인스턴스에 대한 사용 비용과 스냅샷에 대한 스토리지 비용은 계속 발생합니다.

따라서 불필요한 비용이 발생하지 않도록 필요하지 않는 스냅샷을 삭제하고 인스턴스를 종료하는 것이 좋습니다. 등록 취소 중에 자동으로 스냅샷을 삭제하거나 등록 취소 후에 수동으로 스냅샷을 삭제할 수 있습니다. 자세한 내용은 사용하지 않는 리소스로 인한 비용 방지 섹션을 참조하세요.

이후 등록이 취소된 AMI에서 시작된 인스턴스의 경우에도 `describe-instance-image-metadata` AWS CLI 명령을 사용하여 AMI에 대한 몇 가지 개략적인 정보를 볼 수 있습니다. 자세한 내용은 describe-instance-image-metadata 섹션을 참조하세요.

내용

- 고려 사항
- AMI 등록 취소
- 사용하지 않는 리소스로 인한 비용 방지
- 등록 취소로부터 Amazon EC2 AMI 보호

고려 사항

- 계정이 소유하지 않은 AMI는 등록 취소할 수 없습니다.
- AWS Backup 서비스에서 관리하는 AMI를 등록 취소하는 데 Amazon EC2를 사용할 수 없습니다. 대신 AWS Backup을 사용하여 백업 볼트의 해당 복구 지점을 삭제합니다. 자세한 내용은 AWS Backup 개발자 안내서의 백업 삭제를 참조하세요.

AMI 등록 취소

EBS 지원 AMI와 인스턴스 저장소 지원 AMI를 등록 취소할 수 있습니다. EBS 지원 AMI의 경우 연결된 스냅샷을 선택적으로 동시에 삭제할 수 있습니다. 그러나 스냅샷이 다른 AMI와 연결된 경우 삭제하도록 지정된 경우에도 스냅샷은 삭제되지 않습니다.

Console

AMI 등록 해제

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 AMI를 선택합니다.
3. 필터 막대에서 내 소유를 선택하여 사용 가능한 AMI를 나열하거나 비활성화된 이미지를 선택하여 비활성화된 AMI를 나열합니다.
4. 등록 취소할 AMI를 선택합니다.

5. 작업(Actions), AMI 등록 취소(AMI Deregister)를 선택합니다.
6. (선택 사항) 등록 취소 중에 연결된 스냅샷을 삭제하려면 연결된 스냅샷 삭제 확인란을 선택합니다.

> **Note**
>
> 스냅샷이 다른 AMI와 연결된 경우 확인란을 선택하더라도 스냅샷은 삭제되지 않습니다.

7. AMI 등록 취소를 선택합니다.

   콘솔이 목록에서 AMI를 제거하는 데 몇 분 정도 걸릴 수 있습니다. 상태를 새로 고치려면 새로 고침을 선택합니다.

AWS CLI

AMI 등록 해제

다음 deregister-image 명령을 사용합니다.

```
aws ec2 deregister-image --image-id ami-0abcdef1234567890
```

AMI 등록을 취소하고 연결된 스냅샷을 삭제하려면

다음 deregister-image 명령을 사용하고 `--delete-associated-snapshots` 파라미터를 지정합니다. 스냅샷이 다른 AMI와 연결된 경우 이 파라미터를 지정하더라도 스냅샷은 삭제되지 않습니다.

```
aws ec2 deregister-image \
    --image-id ami-0abcdef1234567890 \
    --delete-associated-snapshots
```

PowerShell

AMI 등록 해제

Unregister-EC2Image cmdlet을 사용합니다.

```
Unregister-EC2Image -ImageId ami-0abcdef1234567890
```

AMI 등록을 취소하고 연결된 스냅샷을 삭제하려면

Unregister-EC2Image cmdlet을 사용하고 -DeleteAssociatedSnapshots 파라미터를 지정합니다. 스냅샷이 다른 AMI와 연결된 경우 이 파라미터를 지정하더라도 스냅샷은 삭제되지 않습니다.

```
Unregister-EC2Image `
    -ImageId ami-0abcdef1234567890 `
    -DeleteAssociatedSnapshots
```

사용하지 않는 리소스로 인한 비용 방지

AMI를 등록 취소하더라도 기본적으로 AMI와 연결된 리소스가 모두 삭제되지는 않습니다. 이러한 리소스로는 EBS 지원 AMI에 대한 스냅샷 및 인스턴스 스토어 지원 AMI에 대한 Amazon S3의 파일이 포함됩니다. AMI를 등록 취소할 때 AMI에서 시작된 인스턴스는 종료 또는 중지되지 않습니다.

스냅샷과 파일을 저장하는 비용은 계속 발생하고 실행 중인 인스턴스에 대한 비용도 발생합니다.

이러한 유형의 불필요한 비용을 방지하려면 필요하지 않은 리소스를 삭제하는 것이 좋습니다.

EBS-backed AMI

- AMI 등록을 취소하면서 연결된 스냅샷을 삭제합니다. 자세한 내용은 AMI 등록 취소 섹션을 참조하세요.
- 연결된 스냅샷을 삭제하지 않고 AMI 등록을 취소하는 경우 수동으로 스냅샷을 삭제할 수 있습니다. AMI 생성 중에 생성된 인스턴스 루트 볼륨의 스냅샷은 다음과 같은 설명 형식을 갖습니다.

```
Created by CreateImage(i-1234567890abcdef0) for ami-0abcdef1234567890
```

- AMI에서 시작된 인스턴스가 더는 필요하지 않은 경우 해당 인스턴스를 중지하거나 종료할 수 있습니다. 인스턴스를 나열하려면 AMI의 ID 기준으로 필터링합니다.

인스턴스 저장소 지원 AMI

- ec2-delete-bundle(AMI 도구) 명령을 사용하여 Amazon S3에서 번들을 삭제합니다.
- 번들을 삭제한 후 Amazon S3 버킷이 비어 있고 해당 버킷을 더는 사용하지 않는다면 버킷을 삭제할 수 있습니다.
- AMI에서 시작된 인스턴스가 더는 필요하지 않은 경우 해당 인스턴스를 종료할 수 있습니다. 인스턴스를 나열하려면 AMI의 ID 기준으로 필터링합니다.

등록 취소로부터 Amazon EC2 AMI 보호

AMI에서 등록 취소 보호를 켜서 우발적이거나 악의적인 삭제를 방지할 수 있습니다. 등록 취소 보호를 켜면 IAM 권한이 상관없이 사용자는 AMI를 등록 취소할 수 없습니다. AMI를 등록 취소하려면 먼저 등록 취소 보호를 꺼야 합니다.

AMI에서 등록 취소 보호를 켤 때 24시간의 휴지 기간을 포함하는 옵션이 제공됩니다. 이 휴지 기간은 이 기능을 끈 후에도 등록 취소 보호가 계속 적용되는 기간입니다. 이 휴지 기간에는 AMI 등록을 취소할 수 없습니다. 휴지 기간이 끝나면 AMI를 등록 취소할 수 있습니다.

등록 취소 보호는 모든 기존 및 신규 AMI에서 기본적으로 꺼져 있습니다.

등록 취소 보호 켜기

등록 취소 보호를 켜려면 다음 절차를 따르세요.

Console

등록 취소 보호를 켜는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 AMI를 선택합니다.
3. 필터 막대에서 내 소유를 선택하여 사용 가능한 AMI를 나열하거나 비활성화된 이미지를 선택하여 비활성화된 AMI를 나열합니다.
4. 등록 취소 보호를 켜려는 AMI를 선택하고 작업, AMI 등록 취소 보호 관리를 선택합니다.
5. AMI 등록 취소 보호 관리 대화 상자에서 휴지 기간을 포함하거나 포함하지 않고 등록 취소 보호를 켤 수 있습니다. 다음 옵션 중 하나를 선택하세요.
   - 24시간의 휴지 기간을 포함하여 활성화 - 휴지 기간이 포함된 경우 등록 취소 보호가 꺼진 후에도 24시간 동안 AMI 등록을 취소할 수 없습니다.
   - 휴지 기간 없이 활성화 - 휴지 기간이 없으면 등록 취소 보호가 꺼진 후 즉시 AMI를 등록 취소할 수 있습니다.
6. 저장을 선택합니다.

AWS CLI

등록 취소 보호를 켜는 방법

enable-image-deregistration-protection 명령을 사용합니다. 선택적 휴지 기간을 활성화하려면 `--with-cooldown` 옵션을 포함합니다.

```
aws ec2 enable-image-deregistration-protection \
    --image-id ami-0abcdef1234567890 \
    --with-cooldown
```

PowerShell

등록 취소 보호를 켜는 방법

Enable-EC2ImageDeregistrationProtection cmdlet을 사용합니다. 선택적 휴지 기간을 활성화하려면 `-WithCooldown` 파라미터를 `true`로 설정합니다.

```
Enable-EC2ImageDeregistrationProtection `
    -ImageId ami-0abcdef1234567890 `
    -WithCooldown $true
```

등록 취소 보호 끄기

등록 취소 보호를 끄려면 다음 절차를 따르세요.

AMI에 대한 등록 취소 보호를 켤 때 24시간의 휴지 기간을 포함하도록 선택한 경우 등록 취소 보호를 꺼도 즉시 AMI를 등록 취소할 수 없습니다. 휴지 기간은 24시간이며, 이 기간에 이 기능을 끈 후에도 등록 취소 보호가 계속 적용됩니다. 이 휴지 기간에는 AMI 등록을 취소할 수 없습니다. 휴지 기간이 끝난 후 AMI를 등록 취소할 수 있습니다.

Console

등록 취소 보호를 끄는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 AMI를 선택합니다.
3. 필터 막대에서 내 소유를 선택하여 사용 가능한 AMI를 나열하거나 비활성화된 이미지를 선택하여 비활성화된 AMI를 나열합니다.
4. 등록 취소 보호를 끄려는 AMI를 선택하고 작업, AMI 등록 취소 보호 관리를 선택합니다.
5. AMI 등록 취소 보호 관리 대화 상자에서 비활성화를 선택합니다.
6. 저장을 선택합니다.

AWS CLI

등록 취소 보호를 끄는 방법

disable-image-deregistration-protection 명령을 사용합니다.

```
aws ec2 disable-image-deregistration-protection --image-id ami-0abcdef1234567890
```

PowerShell

등록 취소 보호를 끄는 방법

Disable-EC2ImageDeregistrationProtection cmdlet을 사용합니다.

```
Disable-EC2ImageDeregistrationProtection -ImageId ami-0abcdef1234567890
```

Amazon EC2 부팅 모드를 사용한 인스턴스 시작 동작

컴퓨터가 부팅될 때 실행되는 첫 번째 소프트웨어는 플랫폼을 초기화하고 운영 체제가 플랫폼별 작업을 수행할 수 있는 인터페이스를 제공합니다.

Amazon EC2에서는 부팅 모드 소프트웨어의 두 가지 변형, 즉 통합 확장 가능 펌웨어 인터페이스(UEFI)와 레거시 BIOS입니다.

AMI에서 가능한 부팅 모드 파라미터

AMI는 부트 모드 파라미터 값인 `uefi`, `legacy-bios` 또는 `uefi-preferred` 중에서 하나를 가질 수 있습니다. AMI 부트 모드 파라미터는 선택 사항입니다. 부팅 모드 파라미터가 없는 AMI의 경우 이러한 AMI에서 시작된 인스턴스는 인스턴스 유형의 기본 부팅 모드 값을 사용합니다.

AMI 부팅 모드 파라미터의 목적

AMI 부팅 모드 파라미터는 인스턴스를 시작할 때 사용할 부팅 모드를 Amazon EC2에 알립니다. 부팅 모드 파라미터가 `uefi`로 설정되면 EC2는 인스턴스를 UEFI에서 시작하려고 시도합니다. 운영 체제가 UEFI를 지원하도록 구성되지 않은 경우에는 인스턴스 시작에 실패할 수 있습니다.

UEFI 기본 부팅 모드 파라미터

`uefi-preferred` 부팅 모드 파라미터를 사용하여 UEFI와 레거시 BIOS를 모두 지원하는 AMI를 생성할 수 있습니다. 부팅 모드 파라미터가 `uefi-preferred`로 설정되고 인스턴스 유형이 UEFI를 지원

하는 경우 인스턴스가 UEFI에서 시작됩니다. 인스턴스 유형이 UEFI를 지원하지 않는 경우 인스턴스는 레거시 BIOS에서 시작됩니다.

> **Warning**
>
> UEFI 보안 부팅과 같은 일부 기능은 UEFI에서 부팅하는 인스턴스에서만 사용할 수 있습니다. UEFI를 지원하지 않는 인스턴스 유형과 함께 `uefi-preferred` AMI 부팅 모드 파라미터를 사용하면 인스턴스가 레거시 BIOS로 시작되고 UEFI 종속 기능이 비활성화됩니다. UEFI 종속 기능의 가용성에 의존하는 경우 AMI 부팅 모드 파라미터를 `uefi`로 설정하세요.

인스턴스 유형별 기본 부팅 모드

- Graviton 인스턴스 유형: UEFI
- 인텔 및 AMD 인스턴스 유형: 레거시 BIOS

영역 지원

UEFI 부팅은 Wavelength Zone 또는 AWS Outposts에서 지원되지 않습니다.

부팅 모드 주제

- UEFI 부팅 모드에서 EC2 인스턴스를 시작하기 위한 요구 사항
- Amazon EC2 AMI의 부팅 모드 파라미터 결정
- EC2 인스턴스 유형의 지원되는 부팅 모드 결정
- EC2 인스턴스의 부팅 모드 결정
- EC2 인스턴스에 대한 운영 체제의 부팅 모드 결정
- Amazon EC2 AMI의 부팅 모드 설정
- Amazon EC2 인스턴스의 UEFI 변수
- Amazon EC2 인스턴스의 UEFI 보안 부팅

UEFI 부팅 모드에서 EC2 인스턴스를 시작하기 위한 요구 사항

인스턴스의 부팅 모드는 AMI의 구성, 인스턴스에 포함된 운영 체제, 인스턴스 유형에 따라 결정됩니다. UEFI 부팅 모드에서 인스턴스를 시작하려면 다음 요구 사항을 충족해야 합니다.

AMI

AMI는 UEFI에 대해 다음과 같이 구성되어야 합니다.

- 운영 체제 – AMI에 포함된 운영 체제는 UEFI를 사용할 수 있도록 구성해야 합니다. 그렇지 않으면 인스턴스 시작에 실패합니다. 자세한 내용은 EC2 인스턴스에 대한 운영 체제의 부팅 모드 결정 섹션을 참조하세요.
- AMI 부팅 모드 파라미터 – AMI의 부팅 모드 파라미터를 `uefi` 또는 `uefi-preferred`로 설정해야 합니다. 자세한 내용은 Amazon EC2 AMI의 부팅 모드 파라미터 결정 섹션을 참조하세요.

Linux - 다음 Linux AMI는 UEFI를 지원합니다.

- Amazon Linux 2023
- Amazon Linux 2(Graviton 인스턴스 유형만 해당)

다른 Linux AMI의 경우 AMI를 구성하거나, VM Import/Export를 통해 AMI를 가져오거나, CloudEndure를 통해 AMI를 가져와야 합니다.

Windows – 다음 Windows AMI는 UEFI를 지원합니다.

- Windows_Server-2025-*(`BIOS-` 이름 접두사가 있는 AMI 제외)
- TPM-Windows_Server-2022-English-Full-Base
- TPM-Windows_Server-2022-English-Core-Base
- TPM-Windows_Server-2019-English-Full-Base
- TPM-Windows_Server-2019-English-Core-Base
- TPM-Windows_Server-2016-English-Full-Base
- TPM-Windows_Server-2016-English-Core-Base

인스턴스 유형

AWS Nitro 시스템에 구축된 모든 인스턴스는 UEFI와 레거시 BIOS를 모두 지원합니다. 단, 베어 메탈 인스턴스, DL1, G4ad, P4, u-3tb1, u-6tb1, u-9tb1, u-12tb1, u-18tb1, u-24tb1 및 VT1은 예외입니다. 자세한 내용은 the section called “인스턴스 유형 부팅 모드” 섹션을 참조하세요.

다음 표에서는 인스턴스의 부팅 모드(결과 인스턴스 부팅 모드 열로 표시됨)가 AMI의 부팅 모드 파라미터(열 1), AMI에 포함된 운영 체제의 부팅 모드 구성(열 2), 인스턴스 유형의 부팅 모드 지원(열 3)을 조합하여 결정된다는 것을 보여줍니다.

AMI 부팅 모드 파라미터	운영 체제 부팅 모드 구성	인스턴스 유형 부팅 모드 지원	결과 인스턴스 부팅 모드
UEFI	UEFI	UEFI	UEFI
레거시 BIOS	레거시 BIOS	레거시 BIOS	레거시 BIOS
UEFI 기본	UEFI	UEFI	UEFI
UEFI 기본	UEFI	UEFI 및 레거시 BIOS	UEFI
UEFI 기본	레거시 BIOS	레거시 BIOS	레거시 BIOS
UEFI 기본	레거시 BIOS	UEFI 및 레거시 BIOS	레거시 BIOS
부팅 모드가 지정되지 않음 - ARM	UEFI	UEFI	UEFI
부팅 모드가 지정되지 않음 - x86	레거시 BIOS	UEFI 및 레거시 BIOS	레거시 BIOS

Amazon EC2 AMI의 부팅 모드 파라미터 결정

AMI 부트 모드 파라미터는 선택 사항입니다. AMI는 부트 모드 파라미터 값인 `uefi`, `legacy-bios` 또는 `uefi-preferred` 중에서 하나를 가질 수 있습니다.

일부 AMI에는 부트 모드 파라미터가 없습니다. AMI에 부트 모드 파라미터가 없으면 AMI에서 시작된 인스턴스는 인스턴스 유형의 기본값(Graviton의 경우에 `uefi`, Intel 및 AMD 인스턴스 유형의 경우에 `legacy-bios`)을 사용합니다.

Console

AMI의 부팅 모드 파라미터를 확인하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 AMI를 선택한 후에 AMI를 선택합니다.
3. 부트 모드 필드를 검사합니다.

   - uefi 값은 UEFI가 AMI에서 지원된다는 것을 나타냅니다.

- uefi-preferred 값은 UEFI와 레거시 BIOS가 모두 AMI에서 지원된다는 것을 나타냅니다.
- 값이 없으면 AMI에서 시작된 인스턴스에서는 인스턴스 유형의 기본값을 사용합니다.

인스턴스 시작 시 AMI의 부팅 모드 파라미터를 확인하려면

인스턴스 시작 마법사를 사용하여 인스턴스를 시작하는 경우에는 AMI를 선택하는 단계에서 부트 모드 필드를 검사합니다. 자세한 내용은 애플리케이션 및 OS 이미지(Amazon Machine Image) 섹션을 참조하세요.

AWS CLI

AMI의 부팅 모드 파라미터를 확인하려면

describe-images 명령을 사용하여 AMI의 부팅 모드를 확인합니다.

```
aws ec2 describe-images \
    --region us-east-1 \
    --image-id ami-0abcdef1234567890 \
    --query Images[].BootMode \
    --output text
```

출력의 예시는 다음과 같습니다.

```
uefi
```

출력에서 `uefi` 값은 AMI가 UEFI를 지원함을 나타냅니다. `uefi-preferred` 값은 AMI가 UEFI와 레거시 BIOS를 지원함을 나타냅니다. 값이 없으면 AMI에서 시작된 인스턴스에서는 인스턴스 유형의 기본값을 사용합니다.

PowerShell

AMI의 부팅 모드 파라미터를 확인하려면

Get-EC2Image cmdlet을 사용하여 AMI의 부팅 모드를 확인합니다.

```
Get-EC2Image -Region us-east-1 `
    -ImageId ami-0abcdef1234567890 | Format-List Name, BootMode, TpmSupport
```

출력의 예시는 다음과 같습니다.

```
Name       : TPM-Windows_Server-2016-English-Full-Base-2023.05.10
```

```
BootMode   : uefi
TpmSupport : v2.0
```

출력에서 BootMode 값은 AMI의 부팅 모드를 나타냅니다. uefi 값은 AMI가 UEFI를 지원함을 나타냅니다. uefi-preferred 값은 AMI가 UEFI와 레거시 BIOS를 지원함을 나타냅니다. 값이 없으면 AMI에서 시작된 인스턴스에서는 인스턴스 유형의 기본값을 사용합니다.

EC2 인스턴스 유형의 지원되는 부팅 모드 결정

인스턴스 유형의 지원되는 부팅 모드를 확인할 수 있습니다.

Amazon EC2 콘솔에는 인스턴스 유형의 지원되는 부팅 모드가 표시되지 않습니다.

AWS CLI

describe-instance-types 명령을 사용하여 인스턴스 유형의 지원되는 부팅 모드를 결정합니다. --query 파라미터는 지원되는 부팅 모드만 반환하도록 출력을 필터링합니다.

다음 예제에서는 지정된 인스턴스 유형이 UEFI 및 레거시 BIOS 부팅 모드를 모두 지원합니다.

```
aws ec2 describe-instance-types \
    --instance-types m5.2xlarge \
    --query "InstanceTypes[*].SupportedBootModes"
```

출력의 예시는 다음과 같습니다.

```
[
    [
        "legacy-bios",
        "uefi"
    ]
]
```

다음 예시에서는 레거시 BIOS만 지원하는 t2.xlarge를 보여 줍니다.

```
aws ec2 describe-instance-types \
    --instance-types t2.xlarge \
    --query "InstanceTypes[*].SupportedBootModes"
```

출력의 예시는 다음과 같습니다.

```
[
    [
        "legacy-bios"
    ]
]
```

PowerShell

Get-EC2InstanceType cmdlet을 사용하여 인스턴스 유형의 지원되는 부팅 모드를 확인합니다.

다음 예시에서는 `m5.2xlarge`가 UEFI 및 레거시 BIOS 부트 모드를 모두 지원하는 사례를 보여줍니다.

```
Get-EC2InstanceType -InstanceType m5.2xlarge | Format-List InstanceType,
 SupportedBootModes
```

출력의 예시는 다음과 같습니다.

```
InstanceType       : m5.2xlarge
SupportedBootModes : {legacy-bios, uefi}
```

다음 예시에서는 레거시 BIOS만 지원하는 `t2.xlarge`를 보여 줍니다.

```
Get-EC2InstanceType -InstanceType t2.xlarge | Format-List InstanceType,
 SupportedBootModes
```

출력의 예시는 다음과 같습니다.

```
InstanceType       : t2.xlarge
SupportedBootModes : {legacy-bios}
```

UEFI를 지원하는 인스턴스 유형을 확인하려면 다음을 수행하세요.

UEFI를 지원하는 인스턴스 유형을 확인할 수 있습니다. Amazon EC2 콘솔에는 인스턴스 유형의 UEFI 지원이 표시되지 않습니다.

AWS CLI

사용 가능한 인스턴스 유형은 AWS 리전마다 다릅니다. 리전에서 UEFI를 지원하는 사용 가능한 인스턴스 유형을 확인하려면 describe-instance-types 명령을 사용합니다. UEFI를 지원하는 인스턴스

유형으로 결과 범위를 지정하려면 `--filters` 파라미터를 포함하고 `InstanceType` 값으로 출력 범위를 지정하려면 `--query` 파라미터를 포함합니다.

```
aws ec2 describe-instance-types \
    --filters Name=supported-boot-mode,Values=uefi \
    --query "InstanceTypes[*].[InstanceType]" --output text | sort
```

PowerShell

사용 가능한 인스턴스 유형은 AWS 리전마다 다릅니다. 리전에서 UEFI를 지원하는 사용 가능한 인스턴스 유형을 확인하려면 Get-EC2InstanceType cmdlet을 사용합니다.

```
Get-EC2InstanceType | `
 Where-Object {$_.SupportedBootModes -Contains "uefi"} | `
 Sort-Object InstanceType | `
 Format-Table InstanceType -GroupBy CurrentGeneration
```

UEFI 보안 부팅을 지원하고 비휘발성 변수를 유지하는 인스턴스 유형을 확인하려면 다음을 수행하세요.

베어 메탈 인스턴스는 UEFI 보안 부팅과 비휘발성 변수를 지원하지 않으므로 다음 예제에서는 해당 인스턴스가 출력에서 제외됩니다. UEFI 보안 부팅에 관한 자세한 내용은 Amazon EC2 인스턴스의 UEFI 보안 부팅 섹션을 참조하세요.

AWS CLI

describe-instance-types 명령을 사용하고 출력에서 베어 메탈 인스턴스를 제외합니다.

```
aws ec2 describe-instance-types \
    --filters Name=supported-boot-mode,Values=uefi Name=bare-metal,Values=false \
    --query "InstanceTypes[*].[InstanceType]" \
    --output text | sort
```

PowerShell

Get-EC2InstanceType cmdlet을 사용하고 출력에서 베어 메탈 인스턴스를 제외합니다.

```
Get-EC2InstanceType | `
```

```
    Where-Object { `
        $_.SupportedBootModes -Contains "uefi" -and `
        $_.BareMetal -eq $False
        } | `
    Sort-Object InstanceType  | `
    Format-Table InstanceType, SupportedBootModes, BareMetal, `
        @{Name="SupportedArchitectures";
Expression={$_.ProcessorInfo.SupportedArchitectures}}
```

EC2 인스턴스의 부팅 모드 결정

인스턴스의 부팅 모드는 Amazon EC2 콘솔의 부팅 모드 필드에 표시되며 AWS CLI에 `currentInstanceBootMode` 파라미터별로 표시됩니다.

인스턴스가 시작되면 부팅 모드 파라미터 값은 인스턴스를 시작하는 데 사용된 AMI의 부팅 모드 파라미터 값에 따라 다음과 같이 결정됩니다.

- 부팅 모드 파라미터가 `uefi`인 AMI는 `currentInstanceBootMode` 파라미터가 `uefi`인 인스턴스를 생성합니다.
- 부팅 모드 파라미터가 `legacy-bios`인 AMI는 `currentInstanceBootMode` 파라미터가 `legacy-bios`인 인스턴스를 생성합니다.
- 부팅 모드 파라미터가 `uefi-preferred`인 AMI는 인스턴스 유형이 UEFI를 지원하는 경우 `currentInstanceBootMode` 파라미터가 `uefi`인 인스턴스를 생성하고, 그렇지 않으면 `currentInstanceBootMode` 파라미터가 `legacy-bios`인 인스턴스를 생성합니다.
- 부팅 모드 파라미터 값이 없는 AMI는 AMI 아키텍처가 ARM인지 x86인지 여부와 인스턴스 유형에서 지원되는 부팅 모드에 따라 달라지는 `currentInstanceBootMode` 파라미터 값을 가진 인스턴스를 생성합니다. 기본 부팅 모드는 Graviton 인스턴스 유형에서 `uefi`, Intel 및 AMD 인스턴스 유형에서 `legacy-bios`입니다.

Console

인스턴스의 부팅 모드를 확인하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택한 다음 인스턴스를 선택합니다.
3. 세부 정보 탭에서 부팅 모드 필드를 검사합니다.

AWS CLI

인스턴스의 부팅 모드를 확인하려면

describe-instances 명령을 사용하여 인스턴스의 부팅 모드를 결정합니다. 또한 인스턴스 생성에 사용된 AMI의 부팅 모드를 결정할 수 있습니다.

```
aws ec2 describe-instances \
    --region us-east-1 \
    --instance-ids i-1234567890abcdef0 \
    --query Reservations[].Instances[].BootMode \
    --output text
```

출력의 예시는 다음과 같습니다.

```
uefi
```

PowerShell

인스턴스의 부팅 모드를 확인하려면

Get-EC2Image cmdlet을 사용하여 인스턴스의 부팅 모드를 확인합니다. 또한 인스턴스 생성에 사용된 AMI의 부팅 모드를 결정할 수 있습니다.

```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances | Format-List BootMode,
 CurrentInstanceBootMode, InstanceType, ImageId
```

출력의 예시는 다음과 같습니다.

```
BootMode                : uefi
CurrentInstanceBootMode : uefi
InstanceType            : c5a.large
ImageId                 : ami-0abcdef1234567890
```

EC2 인스턴스에 대한 운영 체제의 부팅 모드 결정

AMI의 부팅 모드는 인스턴스를 부팅하는 데 사용할 부팅 모드를 Amazon EC2에 안내합니다. 인스턴스의 운영 체제가 UEFI에 맞게 구성되어 있는지 확인하려면 SSH(Linux 인스턴스) 또는 RDP(Windows 인스턴스)를 사용하여 인스턴스에 연결해야 합니다.

인스턴스 운영 체제에 대한 지침을 사용하세요.

Linux

인스턴스 운영 체제의 부팅 모드 확인

1. SSH를 사용하여 Linux 인스턴스에 연결합니다.
2. 운영 체제의 부팅 모드를 보려면 다음 중 하나를 시도해 보세요.

   - 다음 명령을 실행합니다.

     ```
     [ec2-user ~]$ sudo /usr/sbin/efibootmgr
     ```

     UEFI 부팅 모드로 부팅된 인스턴스의 예상 출력

     ```
     BootCurrent: 0001
     Timeout: 0 seconds
     BootOrder: 0000,0001
     Boot0000* UiApp
     Boot0001* UEFI Amazon Elastic Block Store vol-xyz
     ```

   - `/sys/firmware/efi` 디렉터리의 존재를 확인하려면 다음 명령을 실행합니다. 이 디렉터리는 인스턴스가 UEFI를 사용하여 부팅하는 경우에만 존재합니다. 디렉터리가 없으면 명령이 `Legacy BIOS Boot Detected`을(를) 반환합니다.

     ```
     [ec2-user ~]$ [ -d /sys/firmware/efi ] && echo "UEFI Boot Detected" || echo
      "Legacy BIOS Boot Detected"
     ```

     UEFI 부팅 모드로 부팅된 인스턴스의 예상 출력

     ```
     UEFI Boot Detected
     ```

     레거시 BIOS 부팅 모드로 부팅된 인스턴스의 예상 출력

     ```
     Legacy BIOS Boot Detected
     ```

   - 다음 명령을 실행하여 EFI가 `dmesg` 출력에 표시되는지 확인합니다.

     ```
     [ec2-user ~]$ dmesg | grep -i "EFI"
     ```

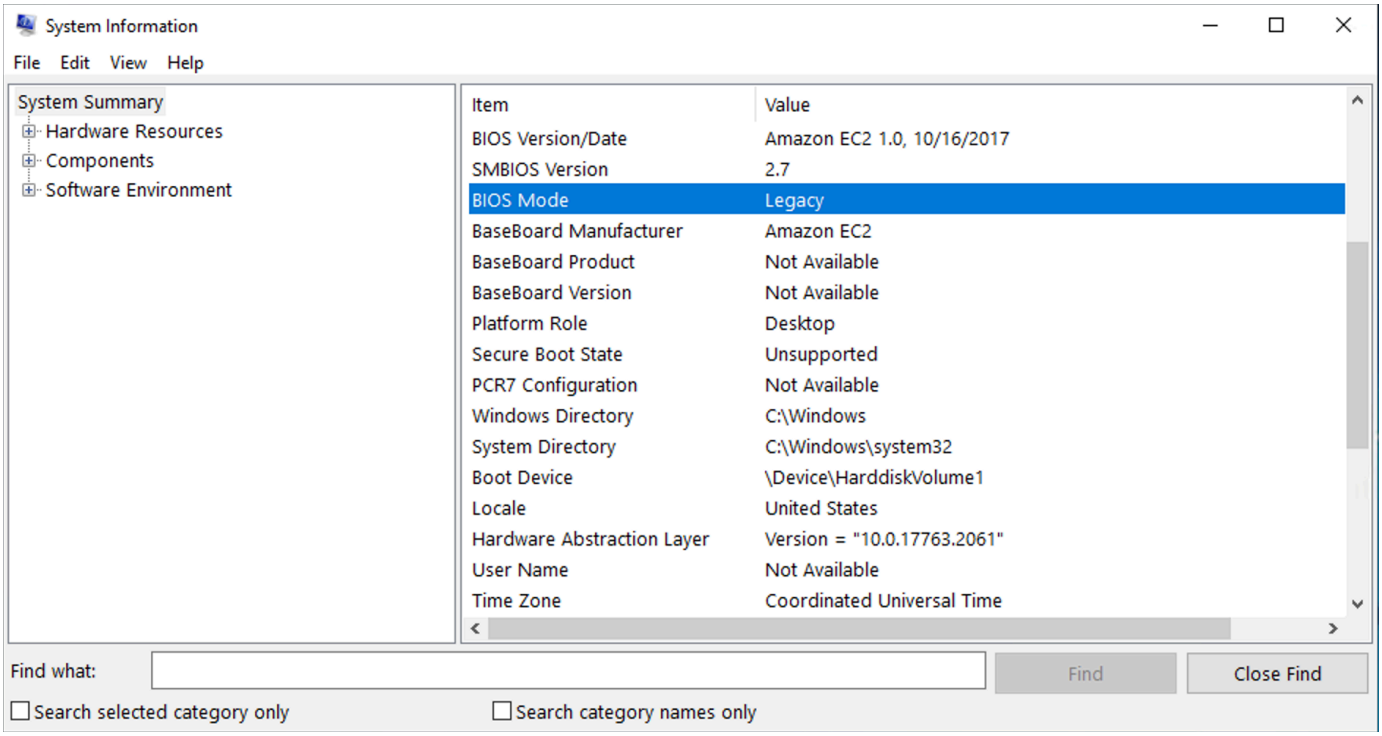
UEFI 부팅 모드로 부팅된 인스턴스의 예상 출력

```
[    0.000000] efi: Getting EFI parameters from FDT:
[    0.000000] efi: EFI v2.70 by EDK II
```

Windows

인스턴스 운영 체제의 부팅 모드 확인

1. RDP를 사용하여 Windows 인스턴스에 연결합니다.
2. 시스템 정보로 이동하여 BIOS 모드 행을 확인합니다.



Amazon EC2 AMI의 부팅 모드 설정

기본적으로 AMI는 AMI를 생성하는 데 사용된 EC2 인스턴스의 부팅 모드를 상속합니다. 예를 들어, 레거시 BIOS에서 실행되는 EC2 인스턴스에서 AMI를 생성하는 경우에는 새 AMI의 부팅 모드는 `legacy-bios`입니다. 부팅 모드가 `uefi-preferred`인 EC2 인스턴스에서 AMI를 생성하는 경우 새 AMI의 부팅 모드는 `uefi-preferred`입니다.

AMI를 등록할 때 AMI의 부팅 모드를 `uefi`, `legacy-bios` 또는 `uefi-preferred`로 설정할 수 있습니다.

AMI 부팅 모드가 `uefi-preferred`로 설정된 경우 인스턴스는 다음과 같이 부팅됩니다.

- UEFI와 레거시 BIOS를 모두 지원하는 인스턴스 유형(예: `m5.large`)의 경우 UEFI를 사용하여 인스턴스가 부팅됩니다.
- 레거시 BIOS만 지원하는 인스턴스 유형(예: `m4.large`)의 경우 레거시 BIOS를 사용하여 인스턴스가 부팅됩니다.

AMI 부팅 모드를 `uefi-preferred`로 설정하면 운영 체제가 UEFI와 레거시 BIOS를 모두 부팅할 수 있는 기능을 지원해야 합니다.

기존 레거시 BIOS 기반 인스턴스를 UEFI로 변환하거나 기존 UEFI 기반 인스턴스를 레거시 BIOS로 변환하려면 먼저 인스턴스의 볼륨 및 운영 체제가 선택한 부팅 모드를 지원하도록 수정해야 합니다. 다음에 볼륨의 스냅샷을 생성합니다. 마지막으로 스냅샷에서 AMI를 생성합니다.

고려 사항

- AMI 부팅 모드 파라미터를 설정해도 지정된 부팅 모드에 대한 운영 체제가 자동으로 구성되지는 않습니다. 먼저 선택한 부팅 모드를 사용한 부팅을 지원하도록 인스턴스의 볼륨 및 운영 체제를 적절하게 수정해야 합니다. 그렇게 하지 않으면 결과 AMI를 사용할 수 없습니다. 예를 들어 레거시 BIOS 기반 Windows 인스턴스를 UEFI로 변환하는 경우 Microsoft의 MBR2GPT 도구를 사용하여 시스템 디스크를 MBR에서 GPT로 변환할 수 있습니다. 필요한 수정 사항은 운영 체제마다 다릅니다. 자세한 내용은 운영 체제 설명서를 참조하세요.
- register-image 명령 또는 Register-EC2Image cmdlet을 사용하여 NitroTPM 및 UEFI Preferred를 모두 지원하는 AMI를 생성할 수는 없습니다.
- UEFI 보안 부팅과 같은 일부 기능은 UEFI에서 부팅하는 인스턴스에서만 사용할 수 있습니다. UEFI를 지원하지 않는 인스턴스 유형과 함께 `uefi-preferred` AMI 부팅 모드 파라미터를 사용하면 인스턴스가 레거시 BIOS로 시작되고 UEFI 종속 기능이 비활성화됩니다. UEFI 종속 기능의 가용성에 의존하는 경우 AMI 부팅 모드 파라미터를 `uefi`로 설정하세요.

AWS CLI

AMI의 부팅 모드를 설정하려면

1. 선택된 부팅 모드를 통한 부팅을 지원하도록 인스턴스의 볼륨 및 운영 체제를 적절하게 수정합니다. 필요한 수정 사항은 운영 체제마다 다릅니다. 자세한 내용은 운영 체제 설명서를 참조하세요.

   > **Warning**
   >
   > 이 단계를 수행하지 않으면 AMI를 사용할 수 없습니다.

2. 인스턴스의 볼륨 ID를 찾으려면 describe-instances 명령을 사용합니다. 다음 단계에서 이 볼륨의 스냅샷을 생성합니다.

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

   출력의 예시는 다음과 같습니다.

   ```
   [
       [
           {
               "DeviceName": "/dev/xvda",
               "Ebs": {
                   "AttachTime": "2024-07-11T01:05:51+00:00",
                   "DeleteOnTermination": true,
                   "Status": "attached",
                   "VolumeId": "vol-1234567890abcdef0"
               }
           }
       ]
   ]
   ```

3. 볼륨의 스냅샷을 생성하려면 create-snapshot 명령을 사용합니다. 이전 단계의 볼륨 ID를 사용합니다.

   ```
   aws ec2 create-snapshot \
       --volume-id vol-01234567890abcdef \
   ```

```
    --description "my snapshot"
```

출력의 예시는 다음과 같습니다.

```
{
    "Description": "my snapshot",
    "Encrypted": false,
    "OwnerId": "123456789012",
    "Progress": "",
    "SnapshotId": "snap-0abcdef1234567890",
    "StartTime": "",
    "State": "pending",
    "VolumeId": "vol-01234567890abcdef",
    "VolumeSize": 30,
    "Tags": []
}
```

4. 스냅샷이 completed 상태가 될 때까지 기다렸다가 다음 단계로 이동합니다. 스냅샷의 상태를 가져오려면 이전 단계에서 얻은 스냅샷 ID와 함께 describe-snapshots 명령을 사용합니다.

```
aws ec2 describe-snapshots \
    --snapshot-ids snap-0abcdef1234567890 \
    --query Snapshots[].State \
    --output text
```

출력의 예시는 다음과 같습니다.

```
completed
```

5. 새 AMI를 생성하려면 register-image 명령을 사용합니다. CreateSnapshot의 출력에서 SnapshotId 값을 복사합니다.

   - 부팅 모드를 UEFI로 설정하려면 uefi 값으로 --boot-mode 파라미터를 추가합니다.

```
aws ec2 register-image \
   --description "my image" \
   --name "my-image" \
   --block-device-mappings "DeviceName=/dev/
sda1,Ebs={SnapshotId=snap-0abcdef1234567890,DeleteOnTermination=true}" \
   --root-device-name /dev/sda1 \
   --virtualization-type hvm \
```

```
   --ena-support \
   --boot-mode uefi
```

- 부팅 모드를 uefi-preferred로 설정하려면 --boot-mode의 값을 uefi-preferred로 설정합니다.

```
aws ec2 register-image \
   --description "my description" \
   --name "my-image" \
   --block-device-mappings "DeviceName=/dev/
sda1,Ebs={SnapshotId=snap-0abcdef1234567890,DeleteOnTermination=true}" \
   --root-device-name /dev/sda1 \
   --virtualization-type hvm \
   --ena-support \
   --boot-mode uefi-preferred
```

6. (선택 사항) 새로 생성된 AMI가 앞서 지정한 부팅 모드인지 확인하려면 describe-images 명령을 사용합니다.

```
aws ec2 describe-images \
    --image-id ami-1234567890abcdef0 \
    --query Images[].BootMode \
    --output text
```

출력의 예시는 다음과 같습니다.

```
uefi
```

PowerShell

AMI의 부팅 모드를 설정하려면

1. 선택된 부팅 모드를 통한 부팅을 지원하도록 인스턴스의 볼륨 및 운영 체제를 적절하게 수정합니다. 필요한 수정 사항은 운영 체제마다 다릅니다. 자세한 내용은 운영 체제 설명서를 참조하세요.

> **Warning**
> 이 단계를 수행하지 않으면 AMI를 사용할 수 없습니다.

2. 인스턴스의 볼륨 ID를 찾으려면 Get-EC2Instance cmdlet을 사용합니다.

```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances.BlockDeviceMappings.Ebs
```

출력의 예시는 다음과 같습니다.

```
AssociatedResource  :
AttachTime          : 7/11/2024 1:05:51 AM
DeleteOnTermination : True
Operator            :
Status              : attached
VolumeId            : vol-01234567890abcdef
```

3. 볼륨의 스냅샷을 생성하려면 New-EC2Snapshot cmdlet을 사용합니다. 이전 단계의 볼륨 ID를 사용합니다.

```
New-EC2Snapshot `
    -VolumeId vol-01234567890abcdef `
    -Description "my snapshot"
```

출력의 예시는 다음과 같습니다.

```
AvailabilityZone          :
Description               : my snapshot
Encrypted                 : False
FullSnapshotSizeInBytes   : 0
KmsKeyId                  :
OwnerId                   : 123456789012
RestoreExpiryTime         :
SnapshotId                : snap-0abcdef1234567890
SseType                   :
StartTime                 : 4/25/2025 6:08:59 PM
State                     : pending
StateMessage              :
VolumeId                  : vol-01234567890abcdef
VolumeSize                : 30
```

4. 스냅샷이 `completed` 상태가 될 때까지 기다렸다가 다음 단계로 이동합니다. 스냅샷의 상태를 가져오려면 이전 단계에서 얻은 스냅샷 ID와 함께 Get-EC2Snapshot cmdlet을 사용합니다.

```
(Get-EC2Snapshot `
    -SnapshotId snap-0abcdef1234567890).State.Value
```

출력의 예시는 다음과 같습니다.

```
completed
```

5. 새 AMI를 생성하려면 Register-EC2Image cmdlet을 사용합니다. New-EC2Snapshot의 출력에서 SnapshotId 값을 복사합니다.

   - 부팅 모드를 UEFI로 설정하려면 uefi 값으로 -BootMode 파라미터를 추가합니다.

   ```
   $block = @{SnapshotId=snap-0abcdef1234567890}
   Register-EC2Image `
      -Description "my image" `
      -Name "my-image" `
      -BlockDeviceMapping @{DeviceName="/dev/xvda";Ebs=$block} `
      -RootDeviceName /dev/xvda `
      -EnaSupport $true `
      -BootMode uefi
   ```

   - 부팅 모드를 uefi-preferred로 설정하려면 -BootMode의 값을 uefi-preferred로 설정합니다.

   ```
   $block = @{SnapshotId=snap-0abcdef1234567890}
   Register-EC2Image `
      -Description "my image" `
      -Name "my-image" `
      -BlockDeviceMapping @{DeviceName="/dev/xvda";Ebs=$block} `
      -RootDeviceName /dev/xvda `
      -EnaSupport $true `
      -BootMode uefi-preferred
   ```

6. (선택 사항) 새로 생성된 AMI가 앞서 지정한 부팅 모드인지 확인하려면 Get-EC2Image cmdlet을 사용합니다.

   ```
   (Get-EC2Image `
       -ImageId ami-1234567890abcdef0).BootMode.Value
   ```

   출력의 예시는 다음과 같습니다.

```
uefi
```

Amazon EC2 인스턴스의 UEFI 변수

부팅 모드가 UEFI로 설정된 인스턴스를 시작하면 변수에 대한 키-값 스토어가 생성됩니다. 스토어는 UEFI 및 인스턴스 운영 체제에서 UEFI 변수를 저장하는 데 사용할 수 있습니다.

UEFI 변수는 부트 로더와 운영 체제에서 초기 시스템 시작을 구성하는 데 사용됩니다. 이를 통해 운영 체제는 부팅 순서와 같은 부팅 프로세스의 특정 설정을 관리하거나 UEFI 보안 부팅 키를 관리할 수 있습니다.

> **Warning**
>
> 인스턴스에 연결할 수 있는 사람(및 인스턴스에서 실행 중인 모든 소프트웨어) 또는 인스턴스에서 GetInstanceUefiData API를 사용할 권한이 있는 사람은 누구나 변수를 읽을 수 있습니다. 암호 또는 개인 식별 정보와 같은 민감한 데이터를 UEFI 변수 저장소에 저장해서는 안 됩니다.

UEFI 변수 지속성

- 2022년 5월 10일 또는 그 전에 시작된 인스턴스의 경우 UEFI 변수는 재부팅 또는 중지 시 초기화됩니다.
- 2022년 5월 11일 또는 그 후에 시작된 인스턴스의 경우 비휘발성으로 표시된 UEFI 변수는 재부팅 및 중지/시작 시 유지됩니다.
- 베어 메탈 인스턴스는 인스턴스 중지/시작 작업에서 UEFI 비휘발성 변수를 유지하지 않습니다.

Amazon EC2 인스턴스의 UEFI 보안 부팅

UEFI 보안 부팅은 Amazon EC2의 장기 보안 부팅 프로세스를 기반으로 구축되며, 재부팅 시 지속적인 위협으로부터 소프트웨어를 보호할 수 있는 추가적인 심층 방어 기능을 제공합니다. 이렇게 하면 인스턴스가 암호화 키로 서명된 소프트웨어만 부팅합니다. 키는 UEFI 비휘발성 변수 스토어의 키 데이터베이스에 저장됩니다. UEFI 보안 부팅은 인스턴스 부팅 흐름의 무단 수정을 방지합니다.

내용

- UEFI 보안 부팅이 Amazon EC2 인스턴스에서 작동하는 방식
- Amazon EC2의 UEFI 보안 부팅 요구 사항

- UEFI 보안 부팅에 대해 Amazon EC2 인스턴스가 활성화되어 있는지 확인
- 사용자 지정 UEFI 보안 부팅 키를 사용하여 Linux AMI 생성
- UEFI 보안 부팅에 대한 AWS 바이너리 blob 생성

UEFI 보안 부팅이 Amazon EC2 인스턴스에서 작동하는 방식

UEFI 보안 부팅은 UEFI에 지정된 기능으로 부팅 체인의 상태에 대한 확인을 제공합니다. 펌웨어 자체 초기화 후에 암호로 검증된 UEFI 바이너리만 실행되도록 설계되었습니다. 이러한 바이너리에는 UEFI 드라이버, 기본 부트로더 및 체인 로드 구성 요소가 포함됩니다.

UEFI 보안 부팅은 신뢰 체인에 사용되는 4개의 키 데이터베이스를 지정합니다. 데이터베이스는 UEFI 변수 스토어에 저장됩니다.

신뢰 체인은 다음과 같습니다.

플랫폼 키(PK) 데이터베이스

PK 데이터베이스는 신뢰의 루트입니다. 여기에는 키 교환 키(KEK) 데이터베이스를 업데이트하기 위해 신뢰 체인에 사용되는 단일 퍼블릭 PK 키가 포함되어 있습니다.

PK 데이터베이스를 변경하려면 업데이트 요청에 서명하기 위한 프라이빗 PK 키가 있어야 합니다. 여기에는 빈 PK 키를 작성하여 PK 데이터베이스를 삭제하는 작업이 포함됩니다.

키 교환 키(KEK) 데이터베이스

KEK 데이터베이스는 서명(db) 및 거부 목록(dbx) 데이터베이스를 업데이트하기 위해 신뢰 체인에 사용되는 퍼블릭 KEK 키 목록입니다.

퍼블릭 KEK 데이터베이스를 변경하려면 업데이트 요청에 서명하기 위한 프라이빗 PK 키가 있어야 합니다.

서명(db) 데이터베이스

db 데이터베이스는 모든 UEFI 부팅 바이너리를 검증하기 위해 신뢰 체인에 사용되는 퍼블릭 키 및 해시 목록입니다.

db 데이터베이스를 변경하려면 업데이트 요청에 서명하기 위한 프라이빗 PK 키 또는 프라이빗 KEK 키가 있어야 합니다.

서명 거부 목록(dbx) 데이터베이스

dbx 데이터베이스는 신뢰할 수 없는 퍼블릭 키 및 바이너리 해시 목록이며 신뢰 체인에서 해지 파일로 사용됩니다.

dbx 데이터베이스는 항상 다른 모든 키 데이터베이스보다 우선합니다.

dbx 데이터베이스를 변경하려면 업데이트 요청에 서명하기 위한 프라이빗 PK 키 또는 프라이빗 KEK 키가 있어야 합니다.

UEFI 포럼은 https://uefi.org/revocationlistfile에서 많은 알려진 불량 바이너리 및 인증서에 대해 퍼블릭적으로 사용 가능한 dbx를 유지 관리합니다.

Important

UEFI 보안 부팅은 모든 UEFI 바이너리에서 서명 검증을 시행합니다. UEFI 보안 부팅에서 UEFI 바이너리 실행을 허용하려면 위에서 설명한 프라이빗 db 키로 서명합니다.

기본적으로 UEFI 보안 부팅은 비활성화되어 있고 시스템은 `SetupMode`에 있습니다. 시스템이 `SetupMode`에 있을 때 암호화 서명 없이 모든 키 변수를 업데이트할 수 있습니다. PK가 설정되면 UEFI 보안 부팅이 활성화되고 SetupMode가 종료됩니다.

Amazon EC2의 UEFI 보안 부팅 요구 사항

지원되는 AMI와 지원되는 인스턴스 유형을 사용하여 Amazon EC2 인스턴스를 시작하면 해당 인스턴스는 UEFI 보안 부팅 데이터베이스에 대해 UEFI 부팅 바이너리를 자동으로 검증합니다. 추가 구성은 필요하지 않습니다. 시작 후 인스턴스에서 UEFI 보안 부팅을 구성할 수도 있습니다.

Note

UEFI 보안 부팅은 부팅 흐름 수정으로부터 인스턴스와 해당 운영 체제를 보호합니다. UEFI 보안 부팅이 활성화된 소스 AMI에서 새 AMI를 생성하고 AMI 내에서 `UefiData` 변경과 같이 복사 프로세스 중에 특정 파라미터를 수정하는 경우 UEFI 보안 부팅을 비활성화할 수 있습니다.

내용

- 지원 AMI
- 지원되는 인스턴스 유형

지원 AMI

Linux AMI

Linux 인스턴스를 시작하려면 Linux AMI에 UEFI 보안 부팅이 활성화되어 있어야 합니다.

Amazon Linux는 AL2023 릴리스 2023.1부터 UEFI 보안 부팅을 지원합니다. 하지만 UEFI 보안 부팅은 기본 AMI에서 활성화되지 않습니다. 자세한 정보는 AL2023 사용 설명서에서 UEFI 보안 부팅을 참조하세요. 이전 버전의 Amazon Linux AMI에서는 UEFI 보안 부팅이 지원되지 않습니다. 지원 AMI를 사용하려면 자체 Linux AMI에서 여러 구성 단계를 수행해야 합니다. 자세한 내용은 사용자 지정 UEFI 보안 부팅 키를 사용하여 Linux AMI 생성 섹션을 참조하세요.

Windows AMI

Windows 인스턴스를 시작하려면 Windows AMI에 UEFI 보안 부팅이 활성화되어 있어야 합니다. Microsoft 키를 사용하여 UEFI 보안 부팅을 위해 사전 구성된 AWS Windows AMI를 찾으려면 AWS Windows AMIs Reference의 Find Windows Server AMIs configured with NitroTPM and UEFI Secure Boot를 참조하세요.

현재는 import-image 명령을 사용하여 UEFI 보안 부팅이 사용 설정된 Windows 가져오기를 지원하지 않습니다.

지원되는 인스턴스 유형

UEFI를 지원하는 모든 가상화된 인스턴스 유형은 UEFI 보안 부팅도 지원합니다. UEFI 보안 부팅을 지원하는 인스턴스 유형은 UEFI 부팅 모드에 대한 요구 사항 섹션을 참조하세요.

> **Note**
>
> 베어 메탈 인스턴스 유형은 UEFI 보안 부팅을 지원하지 않습니다.

UEFI 보안 부팅에 대해 Amazon EC2 인스턴스가 활성화되어 있는지 확인

다음 절차에 따라 UEFI 보안 부팅에 대해 Amazon EC2가 활성화되어 있는지 확인할 수 있습니다.

Linux 인스턴스

`mokutil` 유틸리티를 사용하여 Linux 인스턴스가 UEFI 보안 부팅을 사용하도록 설정되어 있는지 확인할 수 있습니다. 인스턴스에 `mokutil`가 설치되지 않은 경우 설치해야 합니다. Amazon Linux 2의 설치 지침은 Find and install software packages on an Amazon Linux 2 instance를 참조하세요. 다른 Linux 배포는 관련 문서를 참조하세요.

UEFI 보안 부팅에 대해 Linux 인스턴스가 활성화되어 있는지 확인

인스턴스에 연결하고 터미널 창에서 root로 다음 명령을 실행합니다.

```
mokutil --sb-state
```

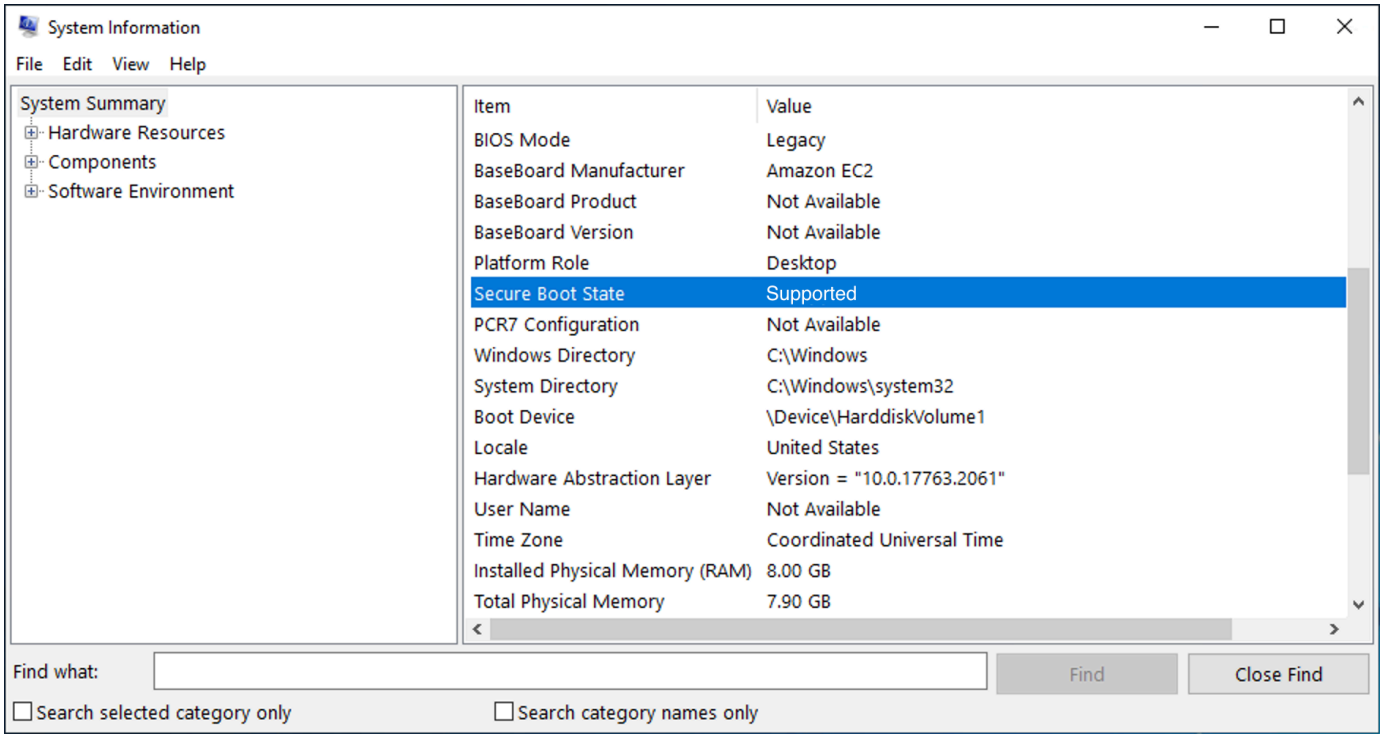
출력의 예시는 다음과 같습니다.

- UEFI 보안 부팅이 활성화된 경우 출력에 SecureBoot enabled가 포함됩니다.
- UEFI 보안 부팅이 활성화되지 않은 경우 출력에 SecureBoot disabled 또는 Failed to read SecureBoot가 포함됩니다.

Windows 인스턴스

UEFI 보안 부팅에 대해 Windows 인스턴스가 활성화되어 있는지 확인

1. 인스턴스에 연결합니다.
2. msinfo32 도구를 엽니다.
3. 보안 부팅 상태(Secure Boot State) 필드를 확인합니다. UEFI 보안 부팅이 활성화된 경우 값은 다음 이미지와 같이 지원됩니다.



Windows PowerShell Cmdlet Confirm-SecureBootUEFI를 사용하여 보안 부팅 상태를 확인할 수도 있습니다. cmdlet에 대한 자세한 내용은 Microsoft 설명서의 Confirm-SecureBootUEFI를 참조하세요.

사용자 지정 UEFI 보안 부팅 키를 사용하여 Linux AMI 생성

이 절차는 UEFI 보안 부팅과 사용자 지정 프라이빗 키를 사용하여 Linux AMI를 생성하는 방법을 보여 줍니다. Amazon Linux는 AL2023 릴리스 2023.1부터 UEFI 보안 부팅을 지원합니다. 자세한 정보는 AL2023 사용 설명서에서 UEFI 보안 부팅을 참조하세요.

> **Important**
>
> 다음 절차는 고급 사용자 전용입니다. 이러한 절차를 사용하려면 SSL 및 Linux 배포 부팅 흐름에 대한 충분한 지식이 있어야 합니다.

사전 조건

- 다음 도구가 사용됩니다.
  - OpenSSL – https://www.openssl.org/
  - efivar – https://github.com/rhboot/efivar
  - efitools – https://git.kernel.org/pub/scm/linux/kernel/git/jejb/efitools.git/
  - get-instance-uefi-dat 명령
- Linux 인스턴스는 UEFI 부팅 모드를 지원하는 Linux AMI로 시작되고 비휘발성 데이터가 있어야 합니다.

UEFI 보안 부팅 키 없이 새로 생성된 인스턴스는 `SetupMode`에 생성됩니다. 이를 통해 자체 키를 등록할 수 있습니다. 일부 AMI는 UEFI 보안 부팅으로 사전 구성되어 제공되며 기존 키를 변경할 수 없습니다. 키를 변경하려면 원래 AMI를 기반으로 새 AMI를 생성해야 합니다.

변수 스토어에 키를 전파하는 방법에는 두 가지가 있습니다. 이들 방법은 다음에 나오는 옵션 A와 옵션 B에 설명되어 있습니다. 옵션 A는 실제 하드웨어의 흐름을 모방하여 인스턴스 내에서 이 작업을 수행하는 방법을 설명합니다. 옵션 B는 AMI를 생성할 때 base64로 인코딩된 파일로 전달되는 바이너리 blob을 생성하는 방법을 설명합니다. 두 옵션 모두 신뢰 체인에 사용되는 3개의 키 페어를 먼저 생성해야 합니다.

UEFI 보안 부팅을 지원하는 Linux AMI를 생성하려면 먼저 3개의 키 페어를 생성한 다음 옵션 A 또는 옵션 B 중 하나만 완료합니다.

- 1단계
- 2단계(옵션 A): 인스턴스 내에서 변수 스토어에 키 추가
- 2단계(옵션 B): 미리 채워진 변수 스토어를 포함하는 바이너리 blob 생성

1단계

UEFI 보안 부팅은 신뢰 체인에 사용되는 플랫폼 키(PK), 키 교환 키(KEK) 및 서명 데이터베이스(db)의 세 가지 키 데이터베이스를 기반으로 합니다.[1]

인스턴스에서 각 키를 생성합니다. UEFI 보안 부팅 표준에 유효한 형식으로 퍼블릭 키를 준비하려면 각 키에 대한 인증서를 생성합니다. DER은 SSL 형식(형식의 바이너리 인코딩)을 정의합니다. 그런 다음 각 인증서를 UEFI 보안 부팅에서 이해하는 바이너리 형식인 UEFI 서명 목록으로 변환합니다. 마지막으로 관련 키로 각 인증서에 서명합니다.

업무

- 키 페어 생성 준비
- 키 페어 1: 플랫폼 키(PK) 생성
- 키 페어 2: 키 교환 키(KEK) 생성
- 키 페어 3: 서명 데이터베이스(db) 생성
- 프라이빗 키로 부팅 이미지(커널)에 서명

키 페어 생성 준비

키 페어를 생성하기 전에 키 생성에 사용할 전역 고유 식별자(GUID)를 생성합니다.

1. 인스턴스에 연결합니다.
2. 셸 프롬프트에서 다음 명령을 실행합니다.

```
uuidgen --random > GUID.txt
```

키 페어 1: 플랫폼 키(PK) 생성

PK는 UEFI 보안 부팅 인스턴스에 대한 신뢰 루트입니다. 프라이빗 PK는 KEK를 업데이트하는 데 사용되며, KEK는 승인된 키를 서명 데이터베이스(db)에 추가하는 데 사용할 수 있습니다.

X.509 표준은 키 페어를 생성하는 데 사용됩니다. 표준에 대한 자세한 내용은 Wikipedia의 X.509를 참조하세요.

PK 생성

1. 키를 생성합니다. 변수 이름을 PK로 지정해야 합니다.

```
openssl req -newkey rsa:4096 -nodes -keyout PK.key -new -x509 -sha256 -days 3650 -
subj "/CN=Platform key/" -out PK.crt
```

다음 파라미터가 지정됩니다.

- `-keyout PK.key` – 프라이빗 키 파일입니다.
- `-days 3650` - 인증서가 유효한 일 수입니다.
- `-out PK.crt` - UEFI 변수를 생성하는 데 사용되는 인증서입니다.
- CN=*Platform key* - 키의 일반 이름(CN)입니다. *### #* 대신 조직 이름을 입력할 수 있습니다.

2. 인증서를 생성합니다.

```
openssl x509 -outform DER -in PK.crt -out PK.cer
```

3. UEFI 서명 목록으로 인증서를 변환합니다.

```
cert-to-efi-sig-list -g "$(< GUID.txt)" PK.crt PK.esl
```

4. 프라이빗 PK(자체 서명)로 UEFI 서명 목록에 서명합니다.

```
sign-efi-sig-list -g "$(< GUID.txt)" -k PK.key -c PK.crt PK PK.esl PK.auth
```

키 페어 2: 키 교환 키(KEK) 생성

프라이빗 KEK는 시스템에서 부팅할 수 있는 승인된 서명 목록인 db에 키를 추가하는 데 사용됩니다.

KEK 생성

1. 키를 생성합니다.

```
openssl req -newkey rsa:4096 -nodes -keyout KEK.key -new -x509 -sha256 -days 3650 -
subj "/CN=Key Exchange Key/" -out KEK.crt
```

2. 인증서를 생성합니다.

```
openssl x509 -outform DER -in KEK.crt -out KEK.cer
```

3. UEFI 서명 목록으로 인증서를 변환합니다.

```
cert-to-efi-sig-list -g "$(< GUID.txt)" KEK.crt KEK.esl
```

4. 프라이빗 PK로 서명 목록에 서명합니다.

```
sign-efi-sig-list -g "$(< GUID.txt)" -k PK.key -c PK.crt KEK KEK.esl KEK.auth
```

키 페어 3: 서명 데이터베이스(db) 생성

db 목록에는 시스템에서 부팅할 권한이 있는 승인된 키가 포함되어 있습니다. 목록을 수정하려면 프라이빗 KEK가 필요합니다. 부팅 이미지는 이 단계에서 생성된 프라이빗 키로 서명됩니다.

db 생성

1. 키를 생성합니다.

```
openssl req -newkey rsa:4096 -nodes -keyout db.key -new -x509 -sha256 -days 3650 -
subj "/CN=Signature Database key/" -out db.crt
```

2. 인증서를 생성합니다.

```
openssl x509 -outform DER -in db.crt -out db.cer
```

3. UEFI 서명 목록으로 인증서를 변환합니다.

```
cert-to-efi-sig-list -g "$(< GUID.txt)" db.crt db.esl
```

4. 프라이빗 KEK로 서명 목록에 서명합니다.

```
sign-efi-sig-list -g "$(< GUID.txt)" -k KEK.key -c KEK.crt db db.esl db.auth
```

프라이빗 키로 부팅 이미지(커널)에 서명

Ubuntu 22.04의 경우 다음 이미지에 서명이 필요합니다.

```
/boot/efi/EFI/ubuntu/shimx64.efi
/boot/efi/EFI/ubuntu/mmx64.efi
/boot/efi/EFI/ubuntu/grubx64.efi
/boot/vmlinuz
```

이미지에 서명

다음 구문을 사용하여 이미지에 서명합니다.

```
sbsign --key db.key --cert db.crt --output /boot/vmlinuz /boot/vmlinuz
```

> **Note**
>
> 모든 새 커널에 서명해야 합니다. */boot/vmlinuz*는 일반적으로 마지막으로 설치된 커널에 심볼릭 링크로 연결됩니다.

부팅 체인과 필요한 이미지에 대해 알아보려면 배포 설명서를 참조하세요.

[1] ArchWiki 커뮤니티의 노고에 감사드립니다. PK 생성, KEK 생성, DB 생성 및 이미지 서명을 위한 명령은 ArchWiki 유지 관리 팀 및/또는 ArchWiki 기고자가 작성한 키 생성에서 가져왔습니다.

2단계(옵션 A): 인스턴스 내에서 변수 스토어에 키 추가

3개의 키 페어를 생성한 후 다음 단계를 완료하여 인스턴스에 연결하고 인스턴스 내에서 변수 스토어에 키를 추가할 수 있습니다. 또는 the section called “2단계, 옵션 B”에 대한 단계를 완료합니다.

옵션 A 단계:

- 1단계: UEFI 보안 부팅을 지원하는 인스턴스 시작
- 2단계: UEFI 보안 부팅을 지원하도록 인스턴스 구성
- 3단계: 인스턴스에서 AMI 생성

1단계: UEFI 보안 부팅을 지원하는 인스턴스 시작

다음 사전 조건으로 인스턴스를 시작하면 UEFI 보안 부팅을 지원하도록 인스턴스를 구성할 준비가 됩니다. 시작 시 인스턴스에서만 UEFI 보안 부팅 지원을 활성화할 수 있습니다. 나중에 활성화할 수 없습니다.

사전 조건

- AMI - Linux AMI는 UEFI 부팅 모드를 지원해야 합니다. AMI가 UEFI 부팅 모드를 지원하는지 확인하려면 AMI 부팅 모드 파라미터가 uefi여야 합니다. 자세한 내용은 Amazon EC2 AMI의 부팅 모드 파라미터 결정 섹션을 참조하세요.

참고: AWS에서는 Graviton 기반 인스턴스 유형의 UEFI를 지원하도록 구성된 Linux AMI만 제공합니다. AWS에서는 현재 UEFI 부팅 모드를 지원하는 x86_64 Linux AMI를 제공하지 않습니다. 모든 아키텍처의 UEFI 부팅 모드를 지원하도록 자체 AMI를 구성할 수 있습니다. UEFI 부팅 모드를 지원하도록 자체 AMI를 구성하려면 자체 AMI에서 여러 구성 단계를 수행해야 합니다. 자세한 내용은 Amazon EC2 AMI의 부팅 모드 설정 섹션을 참조하세요.

- 인스턴스 유형(Instance type) - UEFI를 지원하는 모든 가상화된 인스턴스 유형은 UEFI 보안 부팅도 지원합니다. 베어 메탈 인스턴스 유형은 UEFI 보안 부팅을 지원하지 않습니다. UEFI 보안 부팅을 지원하는 인스턴스 유형은 UEFI 부팅 모드에 대한 요구 사항 섹션을 참조하세요.
- UEFI 보안 부팅 릴리스 후 인스턴스를 시작합니다. UEFI 보안 부팅이 릴리스된 2022년 5월 10일 후에 시작된 인스턴스만 UEFI 보안 부팅을 지원할 수 있습니다.

인스턴스를 시작한 후 UEFI 데이터가 있는지 확인하여 UEFI 보안 부팅을 지원하도록 구성할 준비가 되었는지 확인할 수 있습니다(즉, 2단계로 진행할 수 있음). UEFI 데이터가 있으면 비휘발성 데이터가 지속되는 것입니다.

인스턴스가 2단계를 수행할 준비가 되었는지 확인

get-instance-uefi-data 명령을 사용하여 인스턴스 ID를 지정합니다.

```
aws ec2 get-instance-uefi-data --instance-id i-1234567890abcdef0
```

UEFI 데이터가 출력에 있는 경우 인스턴스는 2단계를 수행할 준비가 된 것입니다. 출력이 비어 있으면 UEFI 보안 부팅을 지원하도록 인스턴스를 구성할 수 없습니다. UEFI 보안 부팅 지원이 제공되기 전에 인스턴스가 시작된 경우 이러한 상황이 발생할 수 있습니다. 새 인스턴스를 시작하고 다시 시도합니다.

2단계: UEFI 보안 부팅을 지원하도록 인스턴스 구성

인스턴스의 UEFI 변수 스토어에 키 페어 등록

> **Warning**
> 키를 등록한 후 부팅 이미지에 서명해야 합니다. 그렇지 않으면 인스턴스를 부팅할 수 없습니다.

서명된 UEFI 서명 목록(PK, KEK 및 db)을 생성한 후에는 UEFI 펌웨어에 등록해야 합니다.

다음과 같은 경우에만 PK 변수에 쓸 수 있습니다.

- 아직 등록된 PK가 없으며 이는 SetupMode 변수가 1인 경우 표시됩니다. 다음 명령을 사용하여 이를 확인합니다. 출력은 1 또는 0입니다.

```
efivar -d -n 8be4df61-93ca-11d2-aa0d-00e098032b8c-SetupMode
```

- 새 PK는 기존 PK의 프라이빗 키로 서명됩니다.

UEFI 변수 스토어에 키 등록

인스턴스에서 다음 명령을 실행해야 합니다.

SetupMode가 활성화된 경우(값은 1) 인스턴스에서 다음 명령을 실행하여 키를 등록할 수 있습니다.

```
[ec2-user ~]$ efi-updatevar -f db.auth db
```

```
[ec2-user ~]$ efi-updatevar -f KEK.auth KEK
```

```
[ec2-user ~]$ efi-updatevar -f PK.auth PK
```

UEFI 보안 부팅이 활성화되었는지 확인

UEFI 보안 부팅이 활성화되었는지 확인하려면 UEFI 보안 부팅에 대해 Amazon EC2 인스턴스가 활성화되어 있는지 확인의 단계를 따르세요.

이제 get-instance-uefi-data CLI 명령을 사용하여 UEFI 변수 스토어를 내보내거나 다음 단계로 계속하고 부팅 이미지에 서명하여 UEFI 보안 부팅 지원 인스턴스로 재부팅할 수 있습니다.

3단계: 인스턴스에서 AMI 생성

인스턴스에서 AMI를 생성하기 위해 콘솔이나 CreateImage API, CLI 또는 SDK를 사용할 수 있습니다. 콘솔 지침은 Amazon EBS 지원 AMI 생성 섹션을 참조하세요. API 지침은 CreateImage를 참조하세요.

> **Note**
>
> CreateImage API는 인스턴스의 UEFI 변수 스토어를 AMI에 자동으로 복사합니다. 콘솔은 CreateImage API를 사용합니다. 이 AMI를 사용하여 인스턴스를 시작하면 인스턴스는 동일한 UEFI 변수 스토어를 갖게 됩니다.

2단계(옵션 B): 미리 채워진 변수 스토어를 포함하는 바이너리 blob 생성

3개의 키 페어를 생성한 후에는 UEFI 보안 부팅 키가 포함된 미리 채워진 변수 스토어가 포함된 이진 blob을 생성할 수 있습니다. 또는 the section called “2단계, 옵션 A”에 대한 단계를 완료합니다.

> **Warning**
>
> 키를 등록하기 전 부팅 이미지에 서명해야 합니다. 그렇지 않으면 인스턴스를 부팅할 수 없습니다.

옵션 B 단계:

- 1단계: 새 변수 스토어 생성 또는 기존 변수 스토어 업데이트
- 2단계: AMI 생성 시 이진 blob 업로드

1단계: 새 변수 스토어 생성 또는 기존 변수 스토어 업데이트

python-uefivars 도구를 사용하여 실행 중인 인스턴스 없이 오프라인으로 변수 스토어를 생성할 수 있습니다. 이 도구는 키에서 새 변수 스토어를 생성할 수 있습니다. 이 스크립트는 현재 EDK2 형식, AWS 형식 및 상위 수준 도구로 더 쉽게 편집할 수 있는 JSON 표현을 지원합니다.

실행 중인 인스턴스 없이 오프라인으로 변수 스토어 생성

1. 다음 링크에서 도구를 다운로드합니다.

   ```
   https://github.com/awslabs/python-uefivars
   ```

2. 다음 명령을 실행하여 키에서 새 변수 스토어를 생성합니다. 그러면 *your_binary_blob*.bin에 base64로 인코딩된 바이너리 blob이 생성됩니다. 이 도구는 `-I` 파라미터를 통해 바이너리 blob 업데이트도 지원합니다.

   ```
   ./uefivars.py -i none -o aws -O your_binary_blob.bin -P PK.esl -K KEK.esl --db
    db.esl --dbx dbx.esl
   ```

2단계: AMI 생성 시 이진 blob 업로드

register-image를 사용하여 UEFI 변수 스토어 데이터를 전달합니다. `--uefi-data` 파라미터에 대해 바이너리 blob을 지정하고 `--boot-mode` 파라미터에 대해 `uefi`를 지정합니다.

```
aws ec2 register-image \
    --name uefi_sb_tpm_register_image_test \
    --uefi-data $(cat your_binary_blob.bin) \
    --block-device-mappings "DeviceName=/dev/sda1,Ebs=
 {SnapshotId=snap-0123456789example,DeleteOnTermination=true}" \
    --architecture x86_64 \
    --root-device-name /dev/sda1 \
    --virtualization-type hvm \
    --ena-support \
    --boot-mode uefi
```

UEFI 보안 부팅에 대한 AWS 바이너리 blob 생성

다음 단계를 사용하여 AMI 생성 중 UEFI 보안 부팅 변수를 사용자 지정할 수 있습니다. 이 단계에서 사용되는 KEK는 2021년 9월 현재 최신 버전입니다. Microsoft에서 KEK를 업데이트하는 경우 최신 KEK를 사용해야 합니다.

AWS 바이너리 blob 생성

1. 빈 PK 서명 목록을 생성합니다.

   ```
   touch empty_key.crt
   cert-to-efi-sig-list empty_key.crt PK.esl
   ```

2. KEK 인증서를 다운로드합니다.

   ```
   https://go.microsoft.com/fwlink/?LinkId=321185
   ```

3. UEFI 서명 목록(`siglist`)에서 KEK 인증서를 래핑합니다.

   ```
   sbsiglist --owner 77fa9abd-0359-4d32-bd60-28f4e78f784b --type x509 --output
    MS_Win_KEK.esl MicCorKEKCA2011_2011-06-24.crt
   ```

4. Microsoft의 db 인증서를 다운로드합니다.

   ```
   https://www.microsoft.com/pkiops/certs/MicWinProPCA2011_2011-10-19.crt
   https://www.microsoft.com/pkiops/certs/MicCorUEFCA2011_2011-06-27.crt
   ```

5. db 서명 목록을 생성합니다.

```
sbsiglist --owner 77fa9abd-0359-4d32-bd60-28f4e78f784b --type x509 --output
 MS_Win_db.esl MicWinProPCA2011_2011-10-19.crt
sbsiglist --owner 77fa9abd-0359-4d32-bd60-28f4e78f784b --type x509 --output
 MS_UEFI_db.esl MicCorUEFCA2011_2011-06-27.crt
cat MS_Win_db.esl MS_UEFI_db.esl > MS_db.esl
```

6. 다음 링크에서 업데이트된 dbx 변경 요청을 다운로드합니다.

```
https://uefi.org/revocationlistfile
```

7. 이전 단계에서 다운로드한 dbx 변경 요청은 이미 Microsoft KEK로 서명되었으므로 이를 제거하거나 압축을 풀어야 합니다. 다음 링크를 사용할 수 있습니다.

```
https://gist.github.com/out0xb2/f8e0bae94214889a89ac67fceb37f8c0
```

```
https://support.microsoft.com/en-us/topic/microsoft-guidance-for-applying-secure-
boot-dbx-update-e3b9e4cb-a330-b3ba-a602-15083965d9ca
```

8. `uefivars.py` 스크립트를 사용하여 UEFI 변수 스토어를 빌드합니다.

```
./uefivars.py -i none -o aws -O uefiblob-microsoft-keys-empty-pk.bin -P ~/PK.esl -K
 ~/MS_Win_KEK.esl --db ~/MS_db.esl  --dbx ~/dbx-2021-April.bin
```

9. 바이너리 blob과 UEFI 변수 스토어를 확인합니다.

```
./uefivars.py -i aws -I uefiblob-microsoft-keys-empty-pk.bin -o json | less
```

10. Blob을 동일한 도구에 다시 전달하여 업데이트할 수 있습니다.

```
./uefivars.py -i aws -I uefiblob-microsoft-keys-empty-pk.bin -o aws -O uefiblob-
microsoft-keys-empty-pk.bin -P ~/PK.esl -K ~/MS_Win_KEK.esl --db ~/MS_db.esl  --dbx
 ~/dbx-2021-April.bin
```

예상 결과

```
Replacing PK
Replacing KEK
Replacing db
Replacing dbx
```

EBS-backed AMI에서 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 볼륨과 마찬가지로 AMI의 스냅샷은 기본 AWS KMS key 또는 지정한 고객 관리형 키로 암호화할 수 있습니다. 어느 경우든 선택한 KMS 키에 대한 사용 권한이 있어야 합니다.

암호화된 스냅샷이 있는 AMI는 AWS 계정 간에 공유할 수 있습니다. 자세한 내용은 Amazon EC2의 공유 AMI 사용량 이해 섹션을 참조하세요.

EBS 지원 AMI를 통한 암호화 주제

- 인스턴스 시작 시나리오
- 이미지 복사 시나리오

인스턴스 시작 시나리오

Amazon EC2는 `RunInstances` 작업과 블록 디바이스 매핑을 통해 제공된 파라미터를 사용하여 AMI에서 시작됩니다. AWS Management Console을 사용하거나 Amazon EC2 API 또는 CLI를 직접 사용할 수 있습니다. 자세한 내용은 Amazon EC2 인스턴스의 볼륨에 대한 블록 디바이스 매핑 단원을 참조하십시오. AWS CLI에서 블록 디바이스 매핑을 제어하는 예제는 시작, 목록 및 EC2 인스턴스 종료 단원을 참조하세요.

기본적으로 명시적인 암호화 파라미터가 없는 경우 `RunInstances` 작업은 AMI 원본 스냅샷에서 EBS 볼륨을 복원하는 동안 AMI 원본 스냅샷의 기존 암호화 상태를 유지합니다. 기본적으로 암호화를 활성화한 경우 스냅샷의 암호화 여부와 상관없이 AMI에서 생성된 모든 볼륨이 암호화됩니다. 기본적으로 암호화를 활성화하지 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 KMS 키(으)로 암호화된 볼륨으로 복원됩니다.
- 소유하지 않은 암호화된 스냅샷(예: 공유 AMI)은 AWS 계정의 기본 KMS 키로 암호화되는 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 KMS 키로 암호화되는 EBS 볼륨으로 복원됩니다.
- 소유한 암호화된 스냅샷은 동일한 KMS 키(으)로 암호화된 EBS 볼륨으로 복원됩니다. (즉, `Encrypted` 파라미터는 아무런 효과가 없습니다.)
- 소유하지 않은 암호화된 스냅샷(즉, 공유 AMI)은 AWS 계정의 기본 KMS 키로 암호화되는 볼륨으로 복원됩니다. (즉, `Encrypted` 파라미터는 아무런 효과가 없습니다.)

`Encrypted` 및 `KmsKeyId` 파라미터를 모두 설정하면 암호화 작업에 대해 기본이 아닌 KMS 키을(를) 지정할 수 있습니다. 결과는 다음 동작과 같습니다.

`Encrypted`와(과) **`KmsKeyId`**이(가) 모두 설정된 인스턴스

- 암호화되지 않은 스냅샷은 지정된 KMS 키(으)로 암호화된 EBS 볼륨으로 복원됩니다.
- 암호화된 스냅샷은 원래의 KMS 키이(가) 아니라 지정된 KMS 키(으)로 암호화된 EBS 볼륨으로 복원됩니다.

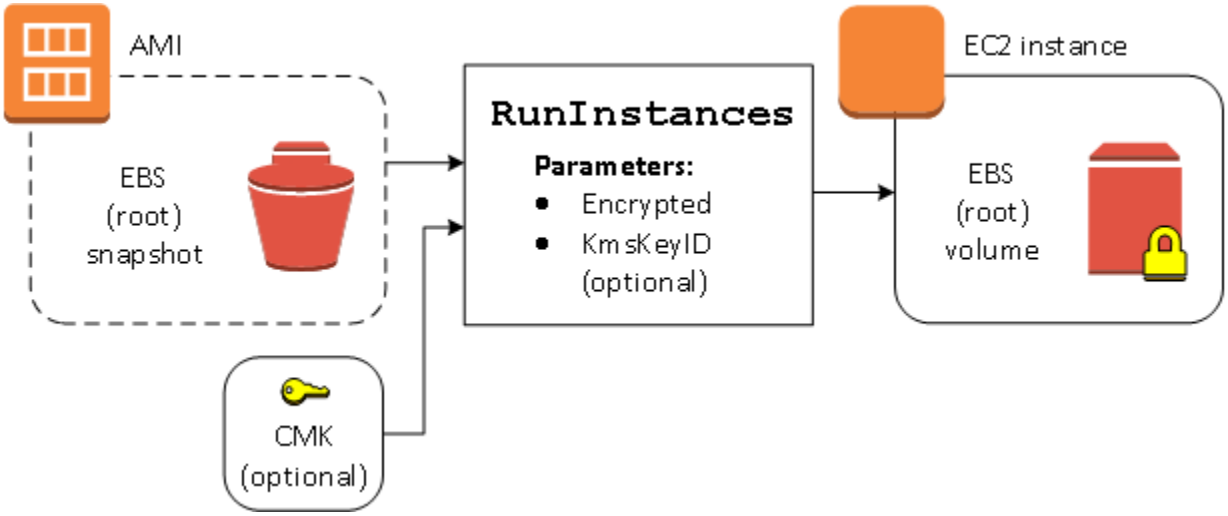
`KmsKeyId` 파라미터를 설정하지 않고 `Encrypted`을(를) 제공하면 오류가 발생합니다.

다음 섹션에서는 기본이 아닌 암호화 파라미터를 사용하여 AMI에서 인스턴스를 시작하는 예시를 볼 수 있습니다. 이러한 각각의 시나리오에서 `RunInstances` 작업에 입력된 파라미터에 의해 스냅샷에서 볼륨을 복원하는 동안 암호화 상태의 변경이 유발됩니다.

콘솔을 사용하여 AMI에서 인스턴스를 시작하는 방법에 대한 자세한 내용은 Amazon EC2 인스턴스 시작하기 섹션을 참조하세요.

시작 중에 볼륨 암호화

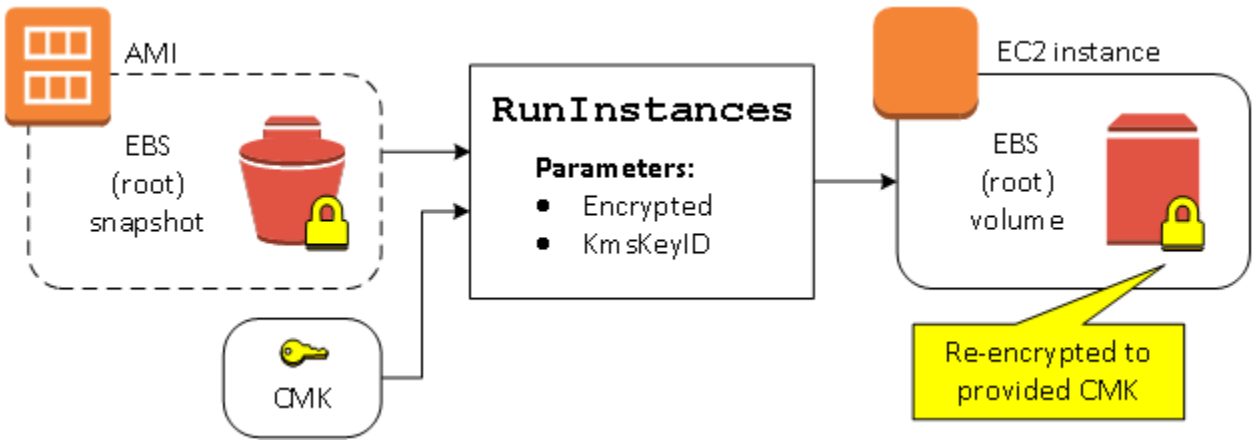
이 예시에서 암호화되지 않은 스냅샷이 지원하는 AMI는 암호화된 EBS 볼륨을 통해 EC2 인스턴스를 시작하는 데 사용됩니다.



`Encrypted` 파라미터만 사용하면 이 인스턴스의 볼륨이 암호화됩니다. `KmsKeyId` 파라미터는 선택 항목입니다. KMS 키 ID를 지정하지 않을 경우 볼륨을 암호화는 데 AWS 계정의 기본 KMS 키가 사용됩니다. 소유한 다른 KMS 키(으)로 사본을 암호화하려면 `KmsKeyId` 파라미터를 입력합니다.

시작 중에 볼륨 재암호화

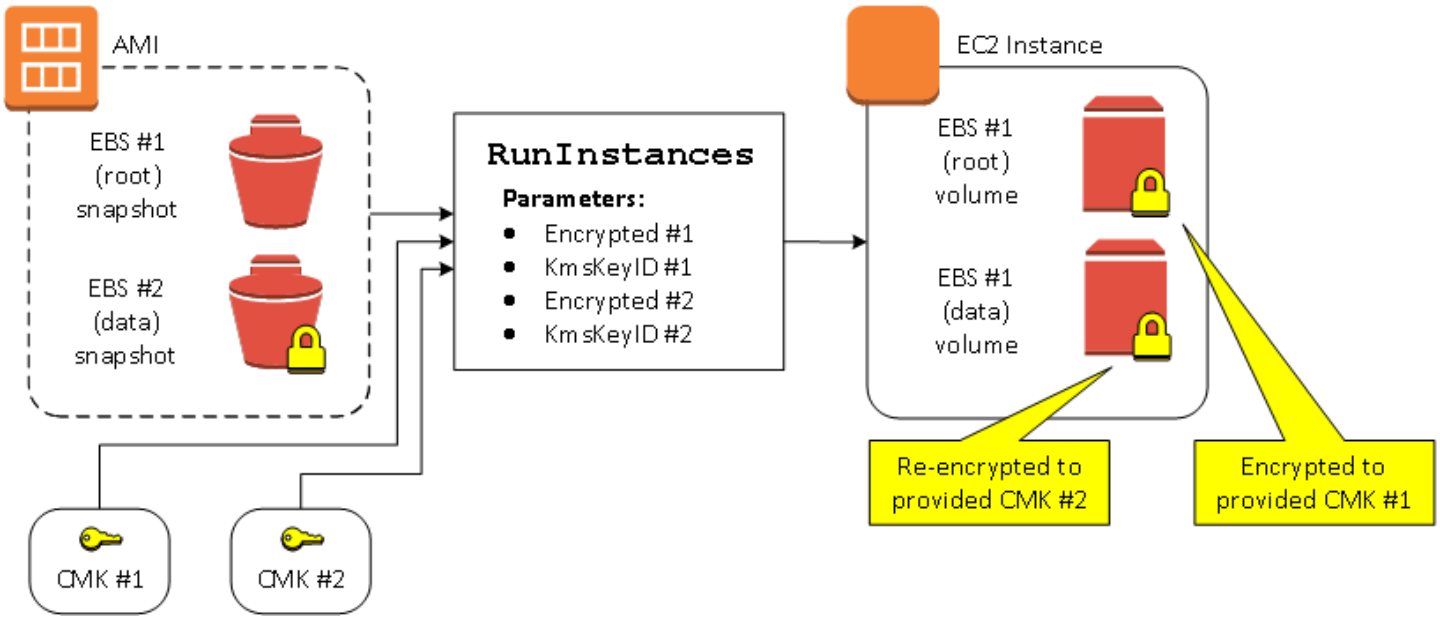
이 예시에서 암호화된 스냅샷이 지원하는 AMI는 새 KMS 키(으)로 암호화된 EBS 볼륨을 통해 EC2 인스턴스를 시작하는 데 사용됩니다.



AMI를 소유한 상태에서 암호화 파라미터를 입력하지 않을 경우, 결과 인스턴스는 해당 스냅샷과 동일한 KMS 키(으)로 암호화된 볼륨을 갖게 됩니다. AMI를 소유하지 않고 공유하며 암호화 파라미터를 입력하지 않을 경우, 볼륨이 기본 KMS 키(으)로 암호화됩니다. 설명된 대로 암호화 파라미터를 입력할 경우, 볼륨이 지정된 KMS 키(으)로 암호화됩니다.

시작 중에 여러 볼륨의 암호화 상태 변경

이 더 복잡한 예시에서 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 AMI는 `CopyImage` 작업을 사용하여 복사됩니다. AWS Management Console을 통하거나 Amazon EC2 API 또는 CLI를 직접 사용할 수 있습니다.

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 KMS 키(으)로 암호화된 스냅샷으로 복사됩니다.
- 소유하지 않은 암호화된 스냅샷(즉, 공유 AMI)은 AWS 계정의 기본 KMS 키로 암호화되는 스냅샷으로 복사됩니다.

암호화 파라미터를 입력하여 이러한 모든 기본 동작을 재정의할 수 있습니다. 사용 가능한 파라미터는 Encrypted 및 KmsKeyId입니다. Encrypted 파라미터만 설정할 경우 그 결과는 다음과 같습니다.

Encrypted이(가) 설정되었지만 **KmsKeyId**이(가) 지정되지 않은 경우의 이미지 복사 동작

- 암호화되지 않은 스냅샷은 AWS 계정의 기본 KMS 키로 암호화된 스냅샷으로 복사됩니다.
- 암호화된 스냅샷은 동일한 KMS 키(으)로 암호화된 스냅샷으로 복사됩니다. (즉, Encrypted 파라미터는 아무런 효과가 없습니다.)
- 소유하지 않은 암호화된 스냅샷(즉, 공유 AMI)은 AWS 계정의 기본 KMS 키로 암호화되는 볼륨으로 복사됩니다. (즉, Encrypted 파라미터는 아무런 효과가 없습니다.)

Encrypted 및 KmsKeyId 파라미터를 모두 설정하면 암호화 작업에 대해 고객 관리형 KMS 키을(를) 지정할 수 있습니다. 결과는 다음 동작과 같습니다.

Encrypted와(과) **KmsKeyId**이(가) 모두 설정된 경우의 이미지 복사 동작

- 암호화되지 않은 스냅샷은 지정된 KMS 키(으)로 암호화된 스냅샷으로 복사됩니다.
- 암호화된 스냅샷은 원래의 KMS 키(이)가 아니라 지정된 KMS 키(으)로 암호화된 스냅샷으로 복사됩니다.

KmsKeyId 파라미터를 설정하지 않고 Encrypted을(를) 제공하면 오류가 발생합니다.

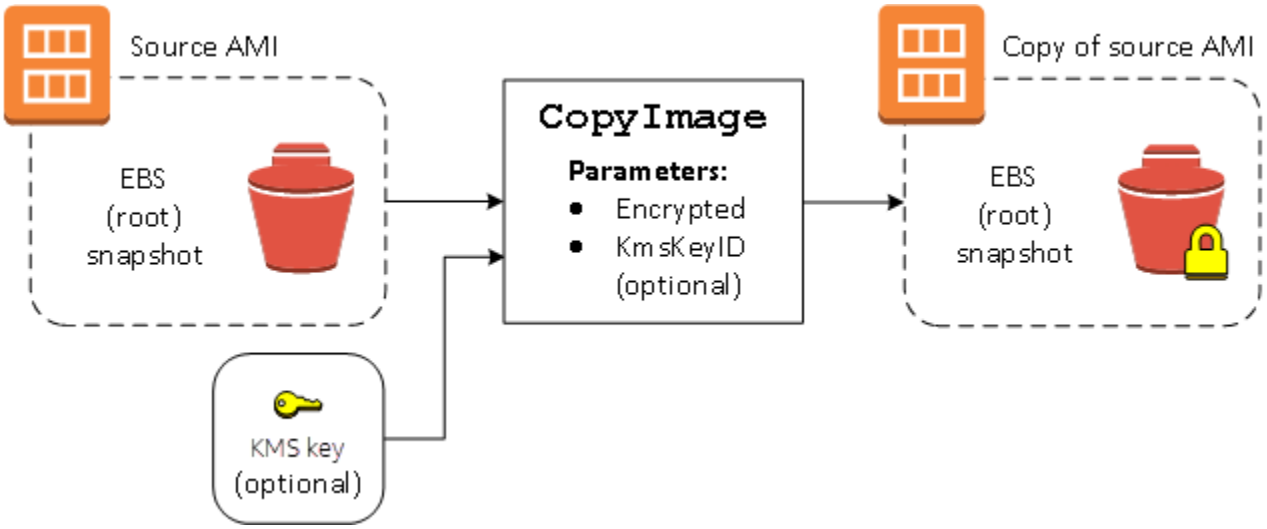
다음 섹션에서는 기본이 아닌 암호화 파라미터를 사용하여 AMI를 복사하여 암호화 상태에 변경이 유발되는 예시를 제공합니다.

콘솔 사용에 대한 자세한 지침은 Amazon EC2 AMI 복사 섹션을 참조하세요.

복사 중에 암호화되지 않은 이미지 암호화

이 시나리오에서는 암호화되지 않은 루트 스냅샷으로 지원되는 AMI가 암호화된 루트 스냅샷이 있는 AMI로 복사됩니다. CopyImage 작업은 고객 관리형 키를 포함하여 2개의 암호화 파라미터를 사용하여 호출됩니다. 따라서 루트 스냅샷의 암호화 상태가 변경되므로 대상 AMI는 소스 스냅샷과 동일한 데이터를 포함하는 루트 스냅샷에 의해 지원되지만 지정된 키를 사용하여 암호화됩니다. 두 AMI 모두의 스냅샷에 대한 스토리지 비용과 각 AMI에서 시작되는 인스턴스에 대한 비용이 발생합니다.

> **Note**
>
> 기본적으로 암호화를 활성화하는 경우 AMI의 모든 스냅샷에서 `Encrypted` 파라미터를 `true`로 설정하는 것과 효과가 동일합니다.



`Encrypted` 파라미터를 설정하면 이 인스턴스에 대한 단일 스냅샷이 암호화됩니다. `KmsKeyId` 파라미터를 지정하지 않으면 스냅샷 복사본을 암호화하는 데 기본 고객 관리형 키가 사용됩니다.

> **Note**
>
> 여러 스냅샷으로 이미지를 복사하고 각 이미지의 암호화 상태를 개별적으로 구성할 수도 있습니다.

Amazon EC2의 공유 AMI 사용량 이해

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

공유 AMI를 사용할 때는 사용자의 주의가 필요합니다. Amazon에서는 다른 Amazon EC2 사용자와 공유된 AMI의 무결성이나 보안성을 보장하지 않습니다. 따라서 공유 AMI를 사용할 때는 데이터 센터에서 외부 코드를 배포하는 경우와 마찬가지로 이런 AMI를 취급하고 그에 따라 적절한 조치를 취해야 합니다. 검증된 제공자 등 신뢰할 수 있는 출처에서 나온 AMI의 사용을 권장합니다.

확인된 공급 업체

Amazon EC2 콘솔에서 Amazon 또는 확인된 Amazon 파트너가 소유한 공용 AMI는 확인된 공급 업체로 표시됩니다.

describe-images AWS CLI 명령을 사용하여 확인된 공급 업체의 퍼블릭 AMI를 식별할 수도 있습니다. Amazon 또는 확인된 파트너가 소유한 퍼블릭 이미지에는 별칭이 있는 소유자(`amazon`, `aws-backup-vault` 또는 `aws-marketplace`)가 있습니다. CLI 출력에서 다음 값이 `ImageOwnerAlias`에 표시됩니다. 다른 사용자는 AMI 별칭을 사용할 수 없습니다. 이렇게 하면 Amazon 또는 확인된 공급 업체의 AMI를 쉽게 찾을 수 있습니다.

확인된 공급 업체가 되려면 AWS Marketplace에 판매자로 등록해야 합니다. 등록한 후에는 AWS Marketplace에 AMI를 표시할 수 있습니다. 자세한 내용을 알아보려면 AWS Marketplace 판매자 설명서https://docs.aws.amazon.com/marketplace/latest/userguide/user-guide-for-sellers.html의 Getting started as a seller(판매자로 시작하기)와 AMI-based products(AMI 기반 제품)를 참조하세요.

공유 AMI 주제

- Amazon EC2 인스턴스에 사용할 공유 AMI 찾기
- Linux용 공유 AMI 사용 준비
- 허용된 AMI를 사용하여 Amazon EC2에서 AMI 검색 및 사용 제어
- AMI를 Amazon EC2에서 공개적으로 사용할 수 있도록 설정
- AMI에 대한 퍼블릭 액세스 치단 이해
- 조직 및 조직 단위와 AMI 공유
- 특정 AWS 계정과 AMI 공유
- AWS 계정과 AMI 공유 취소
- 공유 Linux AMI 생성을 위한 권장 사항

다른 주제에 대한 정보를 찾고 있는 경우

- AMI 생성에 대한 자세한 내용은 the section called “인스턴스 스토어 지원 AMI 생성” 또는 the section called “AMI 생성” 섹션을 참조하세요.
- AWS Marketplace의 애플리케이션 구축, 제공, 유지 관리에 대한 자세한 내용은 AWS Marketplace 설명서를 참조하세요.

Amazon EC2 인스턴스에 사용할 공유 AMI 찾기

Amazon EC2 콘솔 또는 명령줄을 사용하여 Amazon EC2 인스턴스와 함께 사용할 퍼블릭 또는 프라이빗 공유 AMI를 찾을 수 있습니다.

AMI는 리전 리소스입니다. 공유 AMI(퍼블릭 또는 프라이빗)를 검색할 경우에는 공유되고 있는 동일한 리전에서 검색해야 합니다. AMI를 다른 리전에서 사용할 수 있도록 하려면 AMI를 해당 리전에 복사한 후 공유하세요. 자세한 내용은 Amazon EC2 AMI 복사 섹션을 참조하세요.

Console

콘솔은 필터 필드를 제공합니다. 검색 필드에 제공된 필터를 사용하여 검색 범위를 지정할 수도 있습니다.

공유 AMI를 찾는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 AMI를 선택합니다.
3. 첫 번째 필터에서 다음 옵션 중 하나를 선택합니다.
   - 프라이빗 이미지 - 사용자와 공유되는 모든 AMI를 나열합니다.
   - 퍼블릭 이미지 - 모든 퍼블릭 AMI를 나열합니다.
4. (선택 사항) Amazon의 퍼블릭 이미지만 표시하려면 검색 필드를 선택한 다음, 메뉴 옵션에서 소유자 별칭, =, amazon을 차례로 선택합니다.
5. (선택 사항) 필터를 추가하여 요구 사항을 충족하는 AMI로 검색 범위를 지정합니다.

콘솔을 사용해 확인된 공급 업체의 공유된 퍼블릭 AMI를 찾으려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 AMI 카탈로그(AMI Catalog)를 선택합니다.
3. 커뮤니티 AMI를 선택합니다.
4. 결과 구체화 창에서 확인된 공급 업체를 선택합니다. 확인된 공급 업체 레이블은 Amazon 또는 검증된 파트너의 AMI를 나타냅니다.

AWS CLI

describe-images 명령을 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: 확인된 공급 업체가 소유한 AMI 표시

다음의 명령은 확인된 공급 업체가 소유한 AMI를 표시합니다. 확인된 공급업체(Amazon 또는 확인된 파트너)가 소유한 퍼블릭 AMI에는 별칭이 있는 소유자가 있으며, 이는 계정 필드에 `amazon`, `aws-backup-vault` 또는 `aws-marketplace`로 표시됩니다. 이렇게 하면 확인된 공급 업체의 AMI를 쉽게 찾을 수 있습니다. 다른 사용자는 AMI 별칭을 사용할 수 없습니다.

```
aws ec2 describe-images \
    --owners amazon aws-marketplace \
    --query 'Images[*].[ImageId]' \
    --output text
```

예: 계정에서 소유한 AMI 나열

다음 명령은 지정된 AWS 계정이 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"
```

PowerShell

Get-EC2Image cmdlet(Tools for Windows PowerShell)을 사용하여 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.

```
Get-EC2Image -ExecutableUser all
```

예: 명시적 시작 권한으로 AMI 나열

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

```
Get-EC2Image -ExecutableUser self
```

예: 확인된 공급 업체가 소유한 AMI 표시

다음의 명령은 확인된 공급 업체가 소유한 AMI를 표시합니다. 확인된 공급업체(Amazon 또는 확인된 파트너)가 소유한 퍼블릭 AMI에는 별칭이 있는 소유자가 있으며, 이는 계정 필드에 amazon, aws-backup-vault 또는 aws-marketplace로 표시됩니다. 이렇게 하면 확인된 공급 업체의 AMI를 쉽게 찾을 수 있습니다. 다른 사용자는 AMI 별칭을 사용할 수 없습니다.

```
Get-EC2Image -Owner amazon aws-marketplace
```

예: 계정에서 소유한 AMI 나열

다음 명령은 지정된 AWS 계정이 소유한 AMI를 나열합니다.

```
Get-EC2Image -Owner 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.

```
-Filter @{Name="root-device-type"; Values="ebs"}
```

Linux용 공유 AMI 사용 준비

Linux용 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 참조하세요.

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, 문제 없음이 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 사용자가 있는지 확인합니다. 수퍼유저 권한을 가진 사용자에게 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. 예시:

   ```
   [ec2-user ~]$ sudo rm /etc/rsyslog.conf
   [ec2-user ~]$ sudo service rsyslog restart
   ```

6. 모든 cron 작업의 유효성을 확인합니다.

보안을 위협하는 것으로 생각되는 퍼블릭 AMI를 발견했다면 AWS 보안 팀에 연락하세요. 자세한 정보는 AWS 보안 센터 섹션을 참조하세요.

허용된 AMI를 사용하여 Amazon EC2에서 AMI 검색 및 사용 제어

AWS 계정의 사용자가 Amazon Machine Image(AMIs)를 검색하고 사용하는 것을 제어하려면 허용된 AMI 기능을 사용할 수 있습니다. 이 기능을 사용하면 계정 내에서 표시하고 사용할 수 있도록 AMI가 충족해야 하는 기준을 지정할 수 있습니다. 기준이 활성화되면 인스턴스를 시작하는 사용자는 지정된 기준을 준수하는 AMI만 볼 수 있습니다. 예를 들어 신뢰할 수 있는 AMI 공급자 목록을 기준으로 지정할 수 있으며 이러한 공급자의 AMI만 표시되고 사용할 수 있습니다.

허용된 AMI 설정을 활성화하기 전에 감사 모드를 활성화하여 표시되거나 표시되지 않고 사용할 수 있는 AMI를 미리 볼 수 있습니다. 이렇게 하면 계정의 사용자가 의도한 AMI만 보고 사용할 수 있도록 필요에 따라 기준을 세분화할 수 있습니다. 또한 describe-instance-image-metadata 명령을 실행하고 응답을 필터링하여 지정된 기준을 충족하지 않는 AMI로 시작된 인스턴스를 식별할 수 있습니다. 이 정보는 기준을 준수하는 AMI를 사용하도록 시작 구성을 업데이트하거나(예: 시작 템플릿에서 다른 AMI 지정) 이러한 AMI를 허용하도록 기준을 조정하는 결정의 지침이 될 수 있습니다.

계정에서 직접 지정하거나 선언적 정책을 사용하여 계정 수준에서 허용된 AMI 설정을 지정합니다. 이러한 설정은 AMI의 검색 및 사용을 제어하려는 각 AWS 리전에서 구성되어야 합니다. 선언적 정책을 사용하면 여러 리전과 여러 계정에 동시에 설정을 적용할 수 있습니다. 선언적 정책을 사용 중인 경우 계정 내에서 직접 설정을 수정할 수 없습니다. 이 주제에서는 계정 내에서 직접 설정을 구성하는 방법을 설명합니다. 선언적 정책 사용에 대한 자세한 내용은 AWS Organizations 사용 설명서의 선언적 정책을 참조하세요.

> **Note**
>
> 허용된 AMI 기능은 퍼블릭 AMI 또는 계정과 공유된 AMI의 검색 및 사용만 제어합니다. 계정이 소유한 AMI는 제한하지 않습니다. 설정한 기준과 관계없이 본인의 계정에서 생성된 AMI를 계정의 사용자가 항상 검색하고 사용할 수 있습니다.

허용된 AMI의 주요 이점

- 규정 준수 및 보안: 사용자는 지정된 기준을 충족하는 AMI만 검색하고 사용할 수 있으므로 규정을 준수하지 않는 AMI 사용의 위험이 줄어듭니다.
- 효율적인 관리: 허용되는 AMI의 수를 줄이면 다른 AMI를 더 쉽고 효율적으로 관리할 수 있습니다.

- 중앙 집중식 계정 수준 구현: 계정 내에서 직접 또는 선언적 정책을 통해 계정 수준에서 허용되는 AMI 설정을 구성합니다. 이를 통해 효율적인 중앙 집중식 방식으로 전체 계정에서 AMI 사용량을 제어할 수 있습니다.

내용

- 허용된 AMI 작동 방식
- 허용된 AMI 구현 모범 사례
- 필수 IAM 권한
- 허용된 AMI의 설정 관리

허용된 AMI 작동 방식

계정에서 검색하고 사용할 수 있는 AMI를 자동으로 필터링하고 결정하는 기준을 지정합니다. JSON 구성에서 기준을 지정한 다음 API 활성화 작업을 실행하여 기준을 활성화합니다.

허용된 AMI 기준에 대한 JSON 구성

허용된 AMI의 핵심 구성은 허용된 AMI에 대한 기준을 정의하는 JSON 구성입니다.

지원되는 기준

현재 지원되는 유일한 기준은 AMI 공급업체입니다. 유효한 값은 다음과 같이 AWS 및 AWS 계정 ID로 정의되는 별칭입니다.

- `amazon` - AWS에서 생성한 AMI를 식별하는 별칭
- `aws-marketplace` – AWS Marketplace에서 확인된 공급업체가 생성한 AMI를 식별하는 별칭
- `aws-backup-vault` - 논리적 에어 갭 백업 저장소 계정에 있는 AWS 백업 AMI를 식별하는 별칭 논리적 에어 갭 저장소 AWS 백업 기능을 사용하는 경우 이 별칭이 AMI 공급업체로 포함되어 있는지 확인하세요.
- AWS 계정 ID - 하나 이상의 12자리 AWS 계정 ID
- `none` - 계정이 생성한 AMI만 검색하여 사용할 수 있음을 나타냅니다. 퍼블릭 또는 공유 AMI는 검색하고 사용할 수 없습니다. `none`을 지정하면 별칭 또는 계정 ID를 지정할 수 없습니다.

JSON 구성

AMI 기준은 JSON 형식으로 지정됩니다. 다음은 두 개의 별칭과 세 개의 AWS 계정 ID를 지정하는 예시입니다.

```
{
    "ImageCriteria": [
        {
            "ImageProviders": [
                "amazon",
                "aws-marketplace",
                "123456789012",
                "112233445566",
                "009988776655"
            ]
        }
    ]
}
```

기준 평가 방법

JSON 구성의 기준은 or 조건을 사용하여 평가됩니다. 즉, AMI는 ImageCriteria 객체의 지정된 공급자 중 하나만 일치하면 허용된 것으로 간주됩니다. 예를 들어 amazon 및 AWS 계정 ID를 모두 공급자로 지정하면 두 공급자AMI가 허용됩니다.

JSON 구성의 제한

- ImageCriteria 객체: 단일 구성에서 최대 10개의 ImageCriteria 객체를 지정할 수 있습니다.
- ImageProviders 값: 모든 ImageCriteria 객체에서 최대 200개의 값을 지정할 수 있습니다.

제한의 예

AMI 공급업체 계정을 그룹화하는 데 다양한 ImageProviders 목록이 사용되는 다음 예제를 생각해 보겠습니다.

```
{
    "ImageCriteria": [
        {
            "ImageProviders": ["amazon", "aws-marketplace"]
        },
        {
            "ImageProviders": ["123456789012", "112233445566", "121232343454"]
        },
        {
            "ImageProviders": ["998877665555", "987654321098"]
        }
```

```
        // Up to 7 more ImageCriteria objects can be added
        // Up to 193 more ImageProviders values can be added
    ]
}
```

이 예시에서는 다음이 적용됩니다.

- `imageCriteria` 객체는 3개입니다(10개 한도에 도달할 때까지 최대 7개를 더 추가할 수 있음).
- 모든 객체에 총 7개의 `imageProviders` 값이 있습니다(200개 한도에 도달할 때까지 최대 193개를 더 추가할 수 있음).

이 예에서는 `or` 조건을 사용하여 평가되므로 모든 `ImageCriteria` 객체에서 지정된 AMI 공급업체의 AMI가 허용됩니다.

허용된 AMI 작업

허용된 AMI 기능에는 이미지 기준을 관리하기 위한 활성화, 비활성화 및 감사 모드의 세 가지 운영 모드가 있습니다. 이를 통해 이미지 기준을 활성화 또는 비활성화하거나 필요에 따라 검토할 수 있습니다.

활성화됨

허용된 AMI가 활성화된 경우:

- `ImageCriteria`가 적용됩니다.
- 허용된 AMI만 EC2 콘솔과 이미지를 사용하는 API(예: 이미지를 사용하는 다른 작업을 설명, 복사, 저장 또는 수행하는 API)에서 검색할 수 있습니다.
- 인스턴스는 허용된 AMI를 사용해서만 시작할 수 있습니다.

비활성

허용된 AMI가 비활성화된 경우:

- `ImageCriteria`가 적용되지 않습니다.
- AMI 검색 또는 사용에는 제한이 없습니다.

감사 모드

감사 모드에서:

- `ImageCriteria`가 적용되지만 AMI 검색 또는 사용에는 제한이 없습니다.
- EC2 콘솔의 각 AMI에 대해 허용된 이미지 필드에는 허용되는 AMI가 활성화된 경우 계정의 사용자가 AMI를 검색하고 사용할 수 있는지 여부를 나타내는 예 또는 아니요가 표시됩니다.
- 명령줄에서 `describe-image` 작업에 대한 응답에는 허용된 AMI가 활성화된 경우 계정의 사용자가 AMI를 검색하고 사용할 수 있는지 여부를 나타내는 `"ImageAllowed": true` 또는 `"ImageAllowed": false`가 포함됩니다.
- EC2 콘솔에서 AMI 카탈로그는 허용된 AMI가 활성화된 경우 계정의 사용자가 검색하거나 사용할 수 없는 AMI 옆에 허용되지 않음을 표시합니다.

허용된 AMI 구현 모범 사례

허용된 AMI를 구현할 때는 이러한 모범 사례를 고려하여 원활한 전환을 보장하고 AWS 환경의 잠재적 중단을 최소화할 수 있습니다.

1. 감사 모드 활성화

   먼저 감사 모드에서 허용된 AMI를 활성화합니다. 이 모드를 사용하면 실제로 액세스를 제한하지 않고 기준의 영향을 받는 AMI를 확인하여 위험 없는 평가 기간을 확보할 수 있습니다.

2. 허용된 AMI 기준 설정

   조직의 보안 정책, 규정 준수 요구 사항 및 운영 요구 사항에 맞는 AMI 공급업체를 신중하게 설정합니다.

   > **Note**
   >
   > AWS에서 생성한 AMI를 허용하도록 `amazon` 별칭을 지정하여 사용하는 AWS 관리형 서비스가 계정에서 EC2 인스턴스를 계속 시작할 수 있도록 하는 것이 좋습니다.

3. 예상 비즈니스 프로세스에 미치는 영향 확인

   콘솔 또는 CLI를 사용하여 지정된 기준을 충족하지 않는 AMI로 시작된 인스턴스를 식별할 수 있습니다. 이 정보는 기준을 준수하는 AMI를 사용하도록 시작 구성을 업데이트하거나(예: 시작 템플릿에서 다른 AMI 지정) 이러한 AMI를 허용하도록 기준을 조정하는 결정의 지침이 될 수 있습니다.

   콘솔: ec2-instance-launched-with-allowed-ami AWS Config 규칙을 사용하여 실행 중이거나 중지된 인스턴스가 허용된 AMI 기준을 충족하는 AMI로 시작되었는지 확인합니다. AMI가 허용된 AMI 기준을 충족하지 않는 경우 규칙은 NON_COMPLIANT이고, 충족하는 경우 COMPLIANT입니다. 규칙은 허용된 AMI 설정이 활성화 또는 감사 모드로 설정된 경우에만 작동합니다.

CLI: describe-instance-image-metadata 명령을 실행하고 응답을 필터링하여 지정된 기준을 충족하지 않는 AMI로 시작된 인스턴스를 식별합니다.

콘솔 및 CLI 지침은 허용되지 않는 AMI에서 시작된 인스턴스 찾기 섹션을 참조하세요.

4. 허용된 AMI 활성화

   기준이 예상 비즈니스 프로세스에 부정적인 영향을 미치지 않을 것을 확인한 후 허용된 AMI를 활성화합니다.

5. 인스턴스 시작 모니터링

   애플리케이션 및 Amazon EMR, Amazon ECR, Amazon EKS, AWS Elastic Beanstalk와 같이 사용하는 AWS 관리형 서비스 전반의 AMI에서 인스턴스 시작을 계속 모니터링합니다. 예상치 못한 문제가 있는지 확인하고 허용된 AMI 기준을 필요한 만큼 조정합니다.

6. 새 AMI 파일럿

   현재 허용된 AMI 설정을 준수하지 않는 타사 AMI를 테스트하는 경우 AWS는 다음 접근 방식을 권장합니다.

   - 별도의 AWS 계정 사용: 비즈니스 크리티컬 리소스에 액세스할 수 없는 계정을 생성합니다. 허용된 AMI 설정이 이 계정에서 활성화되어 있지 않은지 또는 테스트하려는 AMI가 명시적으로 허용되는지 확인하여 테스트할 수 있습니다.
   - 다른 AWS 리전에서 테스트: 타사 AMI를 사용할 수 있지만 허용된 AMI 설정을 아직 활성화하지 않은 리전을 사용합니다.

   이러한 접근 방식은 새로운 AMI를 테스트하는 동안 비즈니스 크리티컬 리소스의 보안을 유지하는 데 도움이 됩니다.

필수 IAM 권한

허용된 AMI 기능을 사용하려면 다음 IAM 권한이 필요합니다.

- `GetAllowedImagesSettings`
- `EnableAllowedImagesSettings`
- `DisableAllowedImagesSettings`
- `ReplaceImageCriteriaInAllowedImagesSettings`

허용된 AMI의 설정 관리

허용된 AMI의 설정을 관리할 수 있습니다. 이러한 설정은 계정당 리전 기준입니다.

업무

- 허용된 AMI 활성화
- 허용된 AMI 기준 설정
- 허용된 AMI 비활성화
- 허용된 AMI 기준 가져오기
- 허용되는 AMI 찾기
- 허용되지 않는 AMI에서 시작된 인스턴스 찾기

허용된 AMI 활성화

허용된 AMI를 활성화하고 허용된 AMI 기준을 지정할 수 있습니다. 액세스는 제한하지 않는 기준의 영향을 받는 AMI가 표시되는 감사 모드에서 시작하는 것이 좋습니다.

Console

허용된 AMI를 활성화하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 대시보드를 선택합니다.
3. 계정 속성(오른쪽 상단)에서 허용된 AMI를 선택합니다.
4. 허용된 AMI 탭에서 관리를 선택합니다.
5. 허용된 AMI 설정에서 감사 모드 또는 활성화됨을 선택합니다. 감사 모드에서 시작하여 기준을 테스트한 다음에 이 단계로 돌아와서 허용된 AMI를 활성화하는 것이 좋습니다.
6. (선택 사항) AMI 기준의 경우 JSON 형식으로 기준을 입력합니다.
7. 업데이트를 선택합니다.

AWS CLI

허용된 AMI를 활성화하려면

enable-allowed-images-settings 명령을 사용합니다.

```
aws ec2 enable-allowed-images-settings --allowed-images-settings-state enabled
```

그 대신에 감사 모드를 활성화하려면 `enabled` 대신에 `audit-mode`를 지정합니다.

```
aws ec2 enable-allowed-images-settings --allowed-images-settings-state audit-mode
```

PowerShell

허용된 AMI를 활성화하려면

Enable-EC2AllowedImagesSetting cmdlet을 사용합니다.

```
Enable-EC2AllowedImagesSetting -AllowedImagesSettingsState enabled
```

그 대신에 감사 모드를 활성화하려면 `enabled` 대신에 `audit-mode`를 지정합니다.

```
Enable-EC2AllowedImagesSetting -AllowedImagesSettingsState audit-mode
```

허용된 AMI 기준 설정

허용된 AMI를 활성화한 후 허용된 AMI 기준을 설정하거나 바꿀 수 있습니다.

올바른 구성과 유효한 값은 허용된 AMI 기준에 대한 JSON 구성 섹션을 참조하세요.

Console

허용된 AMI 기준을 설정하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 대시보드를 선택합니다.
3. 계정 속성(오른쪽 상단)에서 허용된 AMI를 선택합니다.
4. 허용된 AMI 탭에서 관리를 선택합니다.
5. AMI 기준에 JSON 형식으로 기준을 입력합니다.
6. 업데이트를 선택합니다.

AWS CLI

허용된 AMI 기준을 설정하는 방법

다음과 같이 replace-image-criteria-in-allowed-images-settings 명령을 사용하여 Amazon 및 지정된 계정의 AMI를 허용합니다.

```
aws ec2 replace-image-criteria-in-allowed-images-settings \
    --image-criteria ImageProviders=amazon,123456789012
```

PowerShell

허용된 AMI 기준을 설정하는 방법

다음과 같이 Set-EC2ImageCriteriaInAllowedImagesSetting cmdlet을 사용하여 Amazon 및 지정된 계정의 AMI를 허용합니다.

```
$imageCriteria = New-Object Amazon.EC2.Model.ImageCriterionRequest
$imageCriteria.ImageProviders = @("amazon", "123456789012")
Set-EC2ImageCriteriaInAllowedImagesSetting -ImageCriterion $imageCriteria
```

허용된 AMI 비활성화

다음과 같이 허용된 AMI를 비활성화할 수 있습니다.

Console

허용된 AMI를 비활성화하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 대시보드를 선택합니다.
3. 계정 속성(오른쪽 상단)에서 허용된 AMI를 선택합니다.
4. 허용된 AMI 탭에서 관리를 선택합니다.
5. 허용된 AMI 설정에서 비활성화됨을 선택합니다.
6. 업데이트를 선택합니다.

AWS CLI

허용된 AMI를 비활성화하려면

disable-allowed-images-settings 명령을 사용합니다.

```
aws ec2 disable-allowed-images-settings
```

PowerShell

허용된 AMI를 비활성화하려면

Disable-EC2AllowedImagesSetting cmdlet을 사용합니다.

```
Disable-EC2AllowedImagesSetting
```

허용된 AMI 기준 가져오기

허용된 AMI 설정의 현재 상태와 허용된 AMI 기준을 가져올 수 있습니다.

Console

허용된 AMI 상태 및 기준을 가져오는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 대시보드를 선택합니다.
3. 계정 속성(오른쪽 상단)에서 허용된 AMI를 선택합니다.
4. 허용된 AMI 탭에 허용된 AMI 설정이 활성화됨, 비활성화됨 또는 감사 모드로 설정되어 있습니다.
5. 허용된 AMI의 상태가 활성화됨 또는 감사 모드인 경우 AMI 기준에 JSON 형식으로 AMI 기준이 표시됩니다.

AWS CLI

허용된 AMI 상태 및 기준을 가져오는 방법

get-allowed-images-settings 명령을 사용합니다.

```
aws ec2 get-allowed-images-settings
```

다음 예제 출력에서 상태는 audit-mode이고 이미지 공급자 목록에는 2개의 공급자(amazon 및 지정된 계정)가 포함되어 있습니다.

```
{
    "State": "audit-mode",
    "ImageCriteria": [
        {
```

```
            "ImageProviders": [
                "amazon",
                "123456789012"
            ]
        }
    ],
    "ManagedBy": "account"
}
```

PowerShell

허용된 AMI 상태 및 기준을 가져오는 방법

Get-EC2AllowedImagesSetting cmdlet을 사용합니다.

```
Get-EC2AllowedImagesSetting | `
    Select State, ManagedBy, @{Name='ImageProviders';
 Expression={($_.ImageCriteria.ImageProviders)}}
```

다음 예제 출력에서 상태는 audit-mode이고 이미지 공급자 목록에는 2개의 공급자(amazon 및 지정된 계정)가 포함되어 있습니다.

```
State      ManagedBy ImageProviders
-----      --------- --------------
audit-mode account   {amazon, 123456789012}
```

허용되는 AMI 찾기

현재 허용된 AMI 기준에 따라 허용되거나 허용되지 않는 AMI를 찾을 수 있습니다.

> **Note**
>
> 허용된 AMI가 감사 모드로 되어 있어야 합니다.

Console

허용된 AMI 기준이 AMI에서 충족되는지 확인하는 방법

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

2. 탐색 창에서 AMI를 선택합니다.
3. AMI를 선택합니다.
4. 세부 정보 탭(확인란을 선택한 경우) 또는 요약 영역(AMI ID를 선택한 경우)에서 허용된 이미지 필드를 찾습니다.

   - 예 - 허용된 AMI 기준이 AMI에서 충족됩니다. 허용된 AMI를 활성화하면 본인 계정의 사용자가 이 AMI를 사용할 수 있게 됩니다.
   - 아니요 - 허용된 AMI 기준이 AMI에서 충족되지 않습니다.
5. 탐색 창에서 AMI 카탈로그(AMI Catalog)를 선택합니다.

   허용되지 않음으로 표시된 AMI는 허용된 AMI 기준을 충족하지 않는 AMI를 나타냅니다. 허용된 AMI가 활성화된 경우 계정의 사용자가 이 AMI를 보거나 사용할 수 없습니다.

AWS CLI

허용된 AMI 기준이 AMI에서 충족되는지 확인하는 방법

describe-images 명령을 사용합니다.

```
aws ec2 describe-images \
    --image-id ami-0abcdef1234567890 \
    --query Images[].ImageAllowed \
    --output text
```

출력의 예시는 다음과 같습니다.

```
True
```

허용된 AMI 기준이 충족되는 AMI를 찾는 방법

describe-images 명령을 사용합니다.

```
aws ec2 describe-images \
    --filters "Name=image-allowed,Values=true" \
    --max-items 10 \
    --query Images[].ImageId
```

출력의 예시는 다음과 같습니다.

```
ami-000eaaa8be2fd162a
ami-000f82db25e50de8e
ami-000fc21eb34c7a9a6
ami-0010b876f1287d7be
ami-0010b929226fe8eba
ami-0010957836340aead
ami-00112c992a47ba871
ami-00111759e194abcc1
ami-001112565ffcafa5e
ami-0011e45aaee9fba88
```

PowerShell

허용된 AMI 기준이 AMI에서 충족되는지 확인하는 방법

Get-EC2Image cmdlet을 사용합니다.

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).ImageAllowed
```

출력의 예시는 다음과 같습니다.

```
True
```

허용된 AMI 기준이 충족되는 AMI를 찾는 방법

Get-EC2Image cmdlet을 사용합니다.

```
Get-EC2Image `
    -Filter @{Name="image-allows";Values="true"} `
    -MaxResult 10 | `
    Select ImageId
```

출력의 예시는 다음과 같습니다.

```
ami-000eaaa8be2fd162a
ami-000f82db25e50de8e
ami-000fc21eb34c7a9a6
ami-0010b876f1287d7be
ami-0010b929226fe8eba
ami-0010957836340aead
ami-00112c992a47ba871
ami-00111759e194abcc1
```

```
ami-001112565ffcafa5e
ami-0011e45aaee9fba88
```

허용되지 않는 AMI에서 시작된 인스턴스 찾기

허용된 AMI 기준이 충족되지 않는 AMI를 사용하여 시작된 인스턴스를 식별할 수 있습니다.

Console

인스턴스가 허용되지 않는 AMI를 사용하여 시작되었는지 확인하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다.
4. 세부 정보 탭의 인스턴스 세부 정보에서 허용된 이미지를 찾습니다.
   - 예 - 허용된 AMI 기준이 AMI에서 충족됩니다.
   - 아니요 - 허용된 AMI 기준이 AMI에서 충족되지 않습니다.

AWS CLI

허용되지 않는 AMI를 사용하여 시작된 인스턴스를 찾는 방법

describe-instance-image-metadata 명령을 `image-allowed` 필터와 함께 사용합니다.

```
aws ec2 describe-instance-image-metadata \
    --filters "Name=image-allowed,Values=false" \
    --query "InstanceImageMetadata[*].[InstanceId,ImageMetadata.ImageId]" \
    --output table
```

출력의 예시는 다음과 같습니다.

```
-----------------------------------------------------
|          DescribeInstanceImageMetadata          |
+----------------------+--------------------------+
|  i-08fd74f3f1595fdbd |  ami-09245d5773578a1d6   |
|  i-0b1bf24fd4f297ab9 |  ami-07cccf2bd80ed467f   |
|  i-026a2eb590b4f7234 |  ami-0c0ec0a3a3a4c34c0   |
|  i-006a6a4e8870c828f |  ami-0a70b9d193ae8a799   |
|  i-0781e91cfeca3179d |  ami-00c257e12d6828491   |
```

```
|  i-02b631e2a6ae7c2d9 |  ami-0bfddf4206f1fa7b9  |
+----------------------+-------------------------+
```

PowerShell

허용되지 않는 AMI를 사용하여 시작된 인스턴스를 찾는 방법

Get-EC2InstanceImageMetadata cmdlet을 사용합니다.

```
Get-EC2InstanceImageMetadata `
    -Filter @{Name="image-allowed";Values="false"} | `
    Select InstanceId, @{Name='ImageId'; Expression={($_.ImageMetadata.ImageId)}}
```

출력의 예시는 다음과 같습니다.

```
InstanceId          ImageId
----------          -------
i-08fd74f3f1595fdbd ami-09245d5773578a1d6
i-0b1bf24fd4f297ab9 ami-07cccf2bd80ed467f
i-026a2eb590b4f7234 ami-0c0ec0a3a3a4c34c0
i-006a6a4e8870c828f ami-0a70b9d193ae8a799
i-0781e91cfeca3179d ami-00c257e12d6828491
i-02b631e2a6ae7c2d9 ami-0bfddf4206f1fa7b9
```

AWS Config

ec2-instance-launched-with-allowed-ami AWS Config 규칙을 추가하고, 요구 사항에 맞게 구성한 다음에 이를 사용하여 인스턴스를 평가합니다.

자세한 내용은 AWS Config 개발자 안내서의 AWS Config 규칙 추가 및 ec2-instance-launched-with-allowed-ami를 참조하세요.

AMI를 Amazon EC2에서 공개적으로 사용할 수 있도록 설정

AMI를 모든 AWS 계정과 공유하여 공개적으로 사용할 수 있도록 할 수 있습니다.

AMI의 퍼블릭 공유를 방지하려는 경우 AMI에 대한 퍼블릭 액세스 차단을 활성화할 수 있습니다. 이렇게 하면 AMI를 공개하려는 모든 시도가 차단되어 AMI 데이터의 무단 액세스 및 잠재적 오용을 방지하는 데 도움이 됩니다. 퍼블릭 액세스 차단을 활성화해도 이미 공개적으로 사용 가능한 AMI에는 영향을 미치지 않으며 공개 사용이 유지됩니다. 자세한 내용은 AMI에 대한 퍼블릭 액세스 치단 이해 섹션을 참조하세요.

특정 계정만 AMI를 사용하여 인스턴스를 시작하도록 허용하려면 특정 AWS 계정과 AMI 공유을(를) 참조하세요.

내용

- 고려 사항
- AMI를 모든 AWS 계정과 공유(공개 공유)

고려 사항

AMI를 퍼블릭으로 설정하기 전에 다음 사항을 고려하세요.

- 소유권 - AMI를 공개하려면 사용자의 AWS 계정이 AMI를 소유해야 합니다.
- 리전 - AMI는 리전 리소스입니다. AMI를 공유하면 공유한 리전에서만 사용할 수 있습니다. AMI를 다른 리전에서 사용할 수 있도록 하려면 AMI를 해당 리전에 복사한 후 공유하세요. 자세한 내용은 Amazon EC2 AMI 복사 섹션을 참조하세요.
- 퍼블릭 액세스 차단 - AMI를 공개적으로 공유하려면 AMI를 공개적으로 공유할 각 리전에서 AMI에 대한 퍼블릭 액세스 차단을 비활성화해야 합니다. AMI를 공개적으로 공유한 후에는 AMI에 대한 퍼블릭 액세스 차단을 다시 활성화하여 AMI가 더 이상 퍼블릭 공유되지 않도록 할 수 있습니다.
- 퍼블릭으로 설정할 수 없는 일부 AMI - 다음 구성 요소 중 하나가 AMI에 있으면 퍼블릭으로 설정할 수 없습니다(단, 특정 AWS 계정과 AMI를 공유할 수는 있음).
  - 암호화된 볼륨
  - 암호화된 볼륨의 스냅샷
  - 제품 코드
- 민감한 데이터 노출 방지(Avoid exposing sensitive data) - AMI를 공유할 때 민감한 데이터의 노출을 방지하려면 공유 Linux AMI 생성을 위한 권장 사항에 설명된 보안 고려 사항을 읽고 권장 작업을 따르세요.
- 사용 - AMI를 공유하면 사용자는 AMI에서만 인스턴스를 시작할 수 있습니다. 삭제, 공유 또는 수정할 수 없습니다. 그러나 AMI를 사용하여 인스턴스를 시작한 후에는 시작한 인스턴스에서 AMI를 생성할 수 있습니다.
- 자동 사용 중단 - 기본적으로 모든 퍼블릭 AMI의 사용 중단 날짜가 AMI 생성 날짜로부터 2년으로 설정됩니다. 사용 중단 날짜를 2년보다 짧게 설정할 수 있습니다. 사용 중단 날짜를 취소하거나 중단 날짜를 연장하려면 특정 AWS 계정과 공유만을 통해 AMI를 프라이빗으로 설정해야 합니다.
- 사용되지 않는 AMI 제거 - 퍼블릭 AMI가 사용 중단일에 도달한 후 6개월 이상 AMI에서 새 인스턴스를 시작하지 않은 경우 AWS에서는 퍼블릭 공유 속성을 제거하여 더 이상 사용되지 않는 AMI가 퍼블릭 AMI 목록에 표시되지 않도록 합니다.

- 결제 - 다른 AWS 계정이 인스턴스를 시작하기 위해 AMI를 사용하는 경우에는 요금이 청구되지 않습니다. AMI를 사용하여 인스턴스를 시작하는 계정에는 AMI가 시작하는 인스턴스에 대한 요금이 청구됩니다.

AMI를 모든 AWS 계정과 공유(공개 공유)

AMI를 퍼블릭으로 설정한 후에는 콘솔의 커뮤니티 AMI에서 사용할 수 있습니다. 이는 EC2 콘솔의 왼쪽 탐색기에 있는 AMI 카탈로그에서 또는 콘솔을 사용하여 인스턴스를 시작할 때 액세스할 수 있습니다. AMI 공개 후 AMI가 커뮤니티 AMI(Community AMIs)에 표시되는 데 약간의 시간이 걸릴 수 있다는 점에 유의하세요.

Console

퍼블릭 AMI 설정

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 AMI를 선택합니다.
3. 목록에서 AMI를 선택한 후 작업(Actions), AMI 권한 수정(Edit AMI permissions)을 선택합니다.
4. AMI 가용성에서 퍼블릭을 선택합니다.
5. 변경 사항 저장을 선택합니다.

AWS CLI

각 AMI에는 소유자를 제외하고 해당 AMI를 사용한 인스턴스 시작이 허용된 AWS 계정을 제어할 수 있는 `launchPermission` 속성이 존재합니다. AMI의 `launchPermission` 속성을 변경하여 이 AMI를 퍼블릭 설정(모든 AWS 계정에 시작 권한 허용)하거나 사용자가 지정한 AWS 계정과만 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 사용하여 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}]"
```

PowerShell

각 AMI에는 소유자를 제외하고 해당 AMI를 사용한 인스턴스 시작이 허용된 AWS 계정을 제어할 수 있는 `launchPermission` 속성이 존재합니다. AMI의 `launchPermission` 속성을 변경하여 이 AMI를 퍼블릭 설정(모든 AWS 계정에 시작 권한 허용)하거나 사용자가 지정한 AWS 계정과만 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 Edit-EC2ImageAttribute 명령을 사용하여 지정한 AMI의 `launchPermission` 목록에 `all` 그룹을 추가합니다.

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission `
    -OperationType add `
    -UserGroup all
```

2. AMI의 시작 권한을 확인하려면 다음 Get-EC2ImageAttribute 명령을 사용합니다.

```
Get-EC2ImageAttribute `
    -ImageId 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.

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission `
    -OperationType remove `
    -UserGroup all
```

AMI에 대한 퍼블릭 액세스 차단 이해

AMI의 퍼블릭 공유를 방지하려면 계정 수준에서 AMI에 대한 퍼블릭 액세스 차단을 활성화할 수 있습니다.

퍼블릭 액세스 차단을 활성화하면 AMI를 퍼블릭으로 설정하려는 모든 시도가 자동으로 차단됩니다. 하지만 이미 퍼블릭 AMI가 있는 경우에는 계속 공개적으로 사용할 수 있습니다.

AMI를 공개적으로 공유하려면 퍼블릭 액세스 차단을 비활성화해야 합니다. 공유를 완료한 후에는 퍼블릭 액세스 차단을 다시 활성화하여 의도하지 않은 AMI의 퍼블릭 공유를 방지하는 것이 좋습니다.

> **Note**
>
> 이 설정은 계정 수준에서 직접 구성되거나 선언적 정책을 사용하여 구성됩니다. AMI의 퍼블릭 공유를 방지하려는 각 AWS 리전에 구성해야 합니다. 선언적 정책을 사용하면 여러 리전과 여러 계정에 동시에 설정을 적용할 수 있습니다. 선언적 정책을 사용 중인 경우 계정 내에서 직접 설정을 수정할 수 없습니다. 이 주제에서는 계정 내에서 직접 설정을 구성하는 방법을 설명합니다. 선언적 정책 사용에 대한 자세한 내용은 AWS Organizations 사용 설명서의 선언적 정책을 참조하세요.

관리자 사용자만 AMI에 대한 퍼블릭 액세스 차단을 활성화하거나 비활성화할 수 있도록 IAM 권한을 제한할 수 있습니다.

주제

- 기본 설정
- AMI에 대한 퍼블릭 액세스 차단 설정 관리

기본 설정

AMI에 대한 퍼블릭 액세스 차단 설정은 계정이 신규 또는 기존 계정인지와 퍼블릭 AMI가 있는지 여부에 따라 기본적으로 활성화 또는 비활성화됩니다. 다음 표 기본 설정이 나와 있습니다.

AWS 계정	AMI에 대한 퍼블릭 액세스 차단 기본 설정
신규 계정	활성화됨
퍼블릭 AMI가 없는 기존 계정[1]	활성화됨
하나 이상의 퍼블릭 AMI가 있는 기존 계정	비활성

[1] 2023년 7월 15일 또는 그 이후에 계정에 하나 이상의 퍼블릭 AMI가 있는 경우, 이후에 모든 AMI를 프라이빗으로 전환했더라도 AMI에 대한 퍼블릭 액세스 차단은 계정에 대해 기본적으로 비활성화됩니다.

AMI에 대한 퍼블릭 액세스 차단 설정 관리

AMI에 대한 퍼블릭 액세스 차단 설정을 관리하여 AMI를 공개적으로 공유할 수 있는지 여부를 제어할 수 있습니다. Amazon EC2 콘솔 또는 AWS CLI를 사용하여 AMI에 대한 현재 퍼블릭 액세스 차단 상태를 활성화하거나 비활성화하거나 볼 수 있습니다.

AMI에 대한 퍼블릭 액세스 차단 상태 보기

AMI의 퍼블릭 공유가 계정에서 차단되었는지 확인하려면 AMI에 대한 퍼블릭 액세스 차단 상태를 확인해 볼 수 있습니다. AMI의 퍼블릭 공유가 차단되었는지 여부를 확인하고자 하는 각 AWS 리전의 상태를 확인해야 합니다.

필수 권한

AMI에 대한 현재 퍼블릭 액세스 치단 설정을 가져오려면 `GetImageBlockPublicAccessState` IAM 권한이 있어야 합니다.

Console

지정된 리전의 AMI에 대한 퍼블릭 액세스 차단 상태를 보는 방법

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

2. 탐색 모음(화면 상단)에서 AMI에 대한 퍼블릭 액세스 차단 상태를 확인하고자 하는 리전을 선택합니다.
3. 대시보드가 표시되지 않는다면 데이터 세트 그룹의 탐색 창에서 EC2 대시보드를 선택합니다.
4. 계정 속성에서 데이터 보호 및 보안을 선택합니다.
5. AMI에 대한 공개 액세스 차단에서 퍼블릭 액세스 필드를 선택합니다. 값은 새 퍼블릭 공유 차단됨 또는 새 퍼블릭 공유 허용됨으로 표시됩니다.

AWS CLI

AMI의 퍼블릭 액세스 차단 상태를 가져오려면

get-image-block-public-access-state 명령을 사용합니다. 이때 값은 `block-new-sharing` 또는 `unblocked`입니다.

예제: 특정 리전

```
aws ec2 get-image-block-public-access-state --region us-east-1
```

`ManagedBy` 필드는 설정을 구성한 엔터티를 나타냅니다. 이 예제의 `account`는 해당 설정이 계정에서 직접 구성되었음을 나타냅니다. 값이 `declarative-policy`이면 설정이 선언적 정책에 의해 구성되었음을 의미합니다. 자세한 내용은 AWS Organizations 사용 설명서의 선언적 정책을 참조하세요.

```
{
    "ImageBlockPublicAccessState": "block-new-sharing",
    "ManagedBy": "account"
}
```

예제 계정 내 모든 리전

```
echo -e "Region   \t Public Access State" ; \
echo -e "-------------- \t ----------------------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
```

```
    do (output=$(
        aws ec2 get-image-block-public-access-state \
            --region $region \
            --output text)
        echo -e "$region \t $output"
    );
done
```

출력의 예시는 다음과 같습니다.

```
Region           Public Access State
--------------   ----------------------
ap-south-1       block-new-sharing
eu-north-1       unblocked
eu-west-3        block-new-sharing
...
```

PowerShell

AMI의 퍼블릭 액세스 차단 상태를 가져오려면

Get-EC2ImageBlockPublicAccessState cmdlet을 사용합니다. 이때 값은 `block-new-sharing` 또는 `unblocked`입니다.

예제: 특정 리전

```
Get-EC2ImageBlockPublicAccessState -Region us-east-1
```

출력의 예시는 다음과 같습니다.

```
block-new-sharing
```

예제 계정 내 모든 리전

```
(Get-EC2Region).RegionName | `
    ForEach-Object {
        [PSCustomObject]@{
            Region   = $_
            PublicAccessState = (Get-EC2ImageBlockPublicAccessState -Region $_)
        }
```

```
} | `
Format-Table -AutoSize
```

출력의 예시는 다음과 같습니다.

```
Region         PublicAccessState
------         -----------------
ap-south-1     block-new-sharing
eu-north-1     block-new-sharing
eu-west-3      block-new-sharing
...
```

AMI에 대한 퍼블릭 액세스 차단 활성화

AMI의 퍼블릭 공유를 방지하려면 계정 수준에서 AMI에 대한 퍼블릭 액세스 차단을 활성화합니다. AMI의 퍼블릭 공유를 방지하려는 각 AWS 리전에서 AMI에 대한 퍼블릭 액세스 차단을 활성화해야 합니다. 이미 퍼블릭 AMI가 있는 경우에는 계속 공개적으로 사용할 수 있습니다.

필수 권한

AMI에 대해 퍼블릭 액세스 차단 설정을 활성화하려면 `EnableImageBlockPublicAccess` IAM 권한이 있어야 합니다.

고려 사항

- 이 설정을 구성하는 데에는 최대 10분이 걸릴 수 있습니다. 이 시간 동안 퍼블릭 액세스 상태를 설명하면 응답은 `unblocked`입니다. 구성이 완료되면 응답은 `block-new-sharing`입니다.

Console

지정된 리전의 AMI에 대한 퍼블릭 액세스 차단 활성화

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 모음(화면 상단)에서 AMI에 대한 공개 액세스 차단을 활성화하려는 리전을 선택합니다.
3. 대시보드가 표시되지 않는다면 데이터 세트 그룹의 탐색 창에서 EC2 대시보드를 선택합니다.
4. 계정 속성에서 데이터 보호 및 보안을 선택합니다.
5. AMI에 대한 퍼블릭 액세스 차단에서 관리를 선택합니다.
6. 새 퍼블릭 공유 차단 확인란을 선택한 다음 업데이트를 선택합니다.

AWS CLI

AMI에서 퍼블릭 액세스 치단을 활성화하려면

enable-image-block-public-access 명령을 사용합니다.

예제: 특정 리전

```
aws ec2 enable-image-block-public-access \
--region us-east-1 \
--image-block-public-access-state block-new-sharing
```

출력의 예시는 다음과 같습니다.

```
{
    "ImageBlockPublicAccessState": "block-new-sharing"
}
```

예제 계정 내 모든 리전

```
echo -e "Region   \t Public Access State" ; \
echo -e "-------------- \t ----------------------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
        aws ec2 enable-image-block-public-access \
            --region $region \
            --image-block-public-access-state block-new-sharing \
            --output text)
        echo -e "$region \t $output"
    );
done
```

출력의 예시는 다음과 같습니다.

```
Region           Public Access State
--------------   ----------------------
ap-south-1       block-new-sharing
```

```
eu-north-1       block-new-sharing
eu-west-3        block-new-sharing
...
```

PowerShell

AMI에서 퍼블릭 액세스 치단을 활성화하려면

Enable-EC2ImageBlockPublicAccess 명령을 사용합니다.

예제: 특정 리전

```
Enable-EC2ImageBlockPublicAccess `
    -Region us-east-1 `
    -ImageBlockPublicAccessState block-new-sharing
```

출력의 예시는 다음과 같습니다.

```
Value
-----
block-new-sharing
```

예제 계정 내 모든 리전

```
(Get-EC2Region).RegionName | `
    ForEach-Object {
    [PSCustomObject]@{
        Region            = $_
        PublicAccessState = (
        Enable-EC2ImageBlockPublicAccess `
         -Region $_ `
         -ImageBlockPublicAccessState block-new-sharing)
    }
} | `
Format-Table -AutoSize
```

출력의 예시는 다음과 같습니다.

```
Region         PublicAccessState
------         -----------------
ap-south-1     block-new-sharing
eu-north-1     block-new-sharing
```

```
eu-west-3       block-new-sharing
...
```

AMI에 대한 퍼블릭 액세스 차단 비활성화

계정의 사용자가 AMI를 공개적으로 공유할 수 있도록 허용하려면 계정 수준에서 퍼블릭 액세스 차단을 비활성화하세요. AMI의 퍼블릭 공유를 허용하려는 각 AWS 리전에서 AMI에 대한 퍼블릭 액세스 차단을 비활성화해야 합니다.

필수 권한

AMI에 대해 퍼블릭 액세스 차단 설정을 비활성화하려면 `DisableImageBlockPublicAccess` IAM 권한이 있어야 합니다.

고려 사항

- 이 설정을 구성하는 데에는 최대 10분이 걸릴 수 있습니다. 이 시간 동안 퍼블릭 액세스 상태를 설명하면 응답은 `block-new-sharing`입니다. 구성이 완료되면 응답은 `unblocked`입니다.

Console

지정된 리전의 AMI에 대한 퍼블릭 액세스 차단 비활성화

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 모음(화면 상단)에서 AMI에 대한 퍼블릭 액세스 차단을 비활성화할 리전을 선택합니다.
3. 대시보드가 표시되지 않는다면 데이터 세트 그룹의 탐색 창에서 EC2 대시보드를 선택합니다.
4. 계정 속성에서 데이터 보호 및 보안을 선택합니다.
5. AMI에 대한 퍼블릭 액세스 차단에서 관리를 선택합니다.
6. 새 퍼블릭 공유 차단 확인란을 선택 취소한 다음 업데이트를 선택합니다.
7. 확인 메시지가 나타나면 **`confirm`**을 입력한 다음 퍼블릭 공유 허용을 선택합니다.

AWS CLI

AMI에서 퍼블릭 액세스 차단을 비활성화하려면

disable-image-block-public-access 명령을 사용합니다.

예제: 특정 리전

```
aws ec2 disable-image-block-public-access --region us-east-1
```

출력의 예시는 다음과 같습니다.

```
{
   "ImageBlockPublicAccessState": "unblocked"
}
```

예제 계정 내 모든 리전

```
echo -e "Region   \t Public Access State" ; \
echo -e "-------------- \t ----------------------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
        aws ec2 disable-image-block-public-access \
            --region $region \
            --output text)
        echo -e "$region \t $output"
    );
done
```

출력의 예시는 다음과 같습니다.

```
Region           Public Access State
--------------   ----------------------
ap-south-1       unblocked
eu-north-1       unblocked
eu-west-3        unblocked
...
```

PowerShell

AMI에서 퍼블릭 액세스 치단을 비활성화하려면

Disable-EC2ImageBlockPublicAccess cmdlet을 사용합니다.

예제: 특정 리전

```
Disable-EC2ImageBlockPublicAccess -Region us-east-1
```

출력의 예시는 다음과 같습니다.

```
Value
-----
unblocked
```

예제 계정 내 모든 리전

```
(Get-EC2Region).RegionName | `
    ForEach-Object {
    [PSCustomObject]@{
        Region            = $_
        PublicAccessState = (Disable-EC2ImageBlockPublicAccess -Region $_)
    }
} | `
Format-Table -AutoSize
```

출력의 예시는 다음과 같습니다.

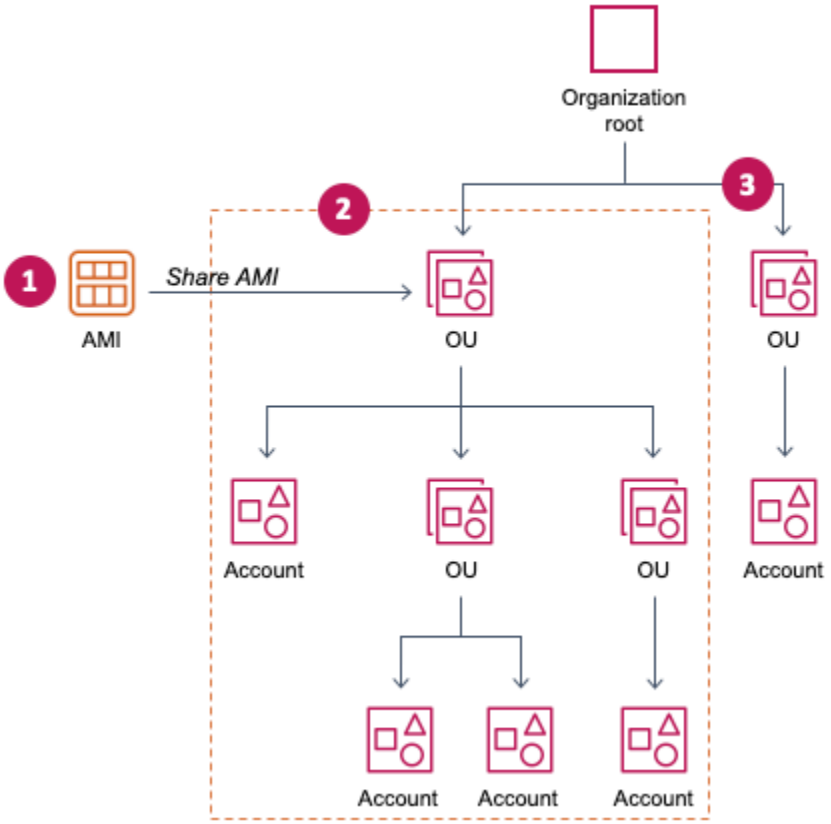
```
Region         PublicAccessState
------         -----------------
ap-south-1     unblocked
eu-north-1     unblocked
eu-west-3      unblocked
...
```

조직 및 조직 단위와 AMI 공유

AWS Organizations는 사용자가 생성해 중앙에서 관리하는 단일 조직으로 여러 AWS 계정을 통합할 수 있는 관리 서비스입니다. 특정 계정과 공유하는 것 외에도 조직 또는 조직 단위(OU)와 AMI를 공유할 수 있습니다.

조직이란 AWS 계정을 통합하고 중앙 집중식으로 관리하기 위해 생성하는 주체입니다. 위에는 루트, 조직 루트 아래에는 조직 단위가 있는 나무형 계층 구조로 계정을 조직할 수 있습니다. 각 계정은 루트에 바로 추가하거나 계층 구조 내의 OU 중 하나에 배치할 수 있습니다. 자세한 내용은 AWS Organizations 사용 설명서의 AWS Organizations 용어 및 개념을 참조하세요.

조직 또는 OU와 AMI를 공유하면 모든 하위 계정이 AMI에 액세스할 수 있습니다. 예를 들어 다음 다이어그램에서 AMI는 최상위 OU와 공유됩니다(숫자 1의 화살표로 표시). 최상위 OU 아래에 중첩된 모든 OU 및 계정(숫자 2의 점선으로 표시)은 또한 AMI에 액세스할 수 있습니다. 조직의 계정과 점선 외부의 OU(숫자 3으로 표시)는 AMI가 공유되는 OU의 하위가 아니므로 AMI에 액세스할 수 없습니다.



주제

- 고려 사항
- 조직 또는 조직 단위의 ARN 가져오기
- 조직 및 OU가 KMS 키를 사용할 수 있도록 허용
- 조직 또는 OU와 AMI 공유 관리

고려 사항

특정 조직 또는 조직 단위와 AMI를 공유하는 경우 다음 사항을 고려하세요.

- 소유권 - AMI를 공유하려면 사용자의 AWS 계정이 AMI를 소유해야 합니다.
- 공유 제한 - AMI 소유자는 자신이 속하지 않은 조직 및 OU를 포함한 모든 조직 또는 OU와 AMI를 공유할 수 있습니다.

리전 내에서 AMI를 공유할 수 있는 최대 엔터티 수는 Amazon EC2 서비스 할당량을 참조하세요.

- 태그 - 사용자 정의 태그(AMI에 연결하는 태그)는 공유할 수 없습니다. AMI를 공유하면 AMI가 공유되는 조직 또는 OU의 AWS 계정에서 사용자 정의 태그를 사용할 수 없습니다.
- ARN 형식(ARN format) - 명령에 조직 또는 OU를 지정할 때 올바른 ARN 형식을 사용해야 합니다. ID만 지정하면 오류가 발생합니다(예: `o-123example` 또는 `ou-1234-5example`만 지정하는 경우).

  올바른 ARN 형식:

  - 조직 ARN: `arn:aws:organizations::`*`account-id`*`:organization/`*`organization-id`*
  - OU ARN: `arn:aws:organizations::`*`account-id`*`:ou/`*`organization-id`*`/`*`ou-id`*

  위치:

  - *`account-id`*은(는) 123456789012와 같은 12자리 관리 계정 번호입니다. 관리 계정 번호를 모르는 경우 관리 계정 번호가 포함된 ARN을 가져올 조직 또는 조직 단위를 설명할 수 있습니다. 자세한 내용은 조직 또는 조직 단위의 ARN 가져오기 섹션을 참조하세요.
  - *`organization-id`*은(는) `o-123example`와 같은 조직 ID입니다.
  - *`ou-id`*은(는) `ou-1234-5example`와 같은 조직 단위 ID입니다.

  ARN 형식에 대한 자세한 내용은 IAM 사용 설명서의 Amazon 리소스 이름(ARN)을 참조하세요.
- 암호화 및 키 - 암호화되지 않거나 암호화된 스냅샷의 지원을 받는 AMI를 공유할 수 있습니다.
  - 암호화된 스냅샷은 고객 관리형 키로 암호화해야 합니다. 기본 AWS 관리형 키로 암호화된 스냅샷의 지원을 받는 AMI는 공유할 수 없습니다.
  - 암호화된 스냅샷의 지원을 받는 AMI를 공유하는 경우 조직 또는 OU가 스냅샷을 암호화하는 데 사용된 고객 관리형 키를 사용하도록 허용해야 합니다. 자세한 내용은 조직 및 OU가 KMS 키를 사용할 수 있도록 허용 단원을 참조하세요.
- 리전 - AMI는 리전 리소스입니다. AMI를 공유하면 공유한 리전에서만 사용할 수 있습니다. AMI를 다른 리전에서 사용할 수 있도록 하려면 AMI를 해당 리전에 복사한 후 공유하세요. 자세한 내용은 Amazon EC2 AMI 복사 섹션을 참조하세요.
- 사용 - AMI를 공유하면 사용자는 AMI에서만 인스턴스를 시작할 수 있습니다. 삭제, 공유 또는 수정할 수 없습니다. 그러나 AMI를 사용하여 인스턴스를 시작한 후에는 시작한 인스턴스에서 AMI를 생성할 수 있습니다.
- 결제 - 다른 AWS 계정이 인스턴스를 시작하기 위해 AMI를 사용하는 경우에는 요금이 청구되지 않습니다. AMI를 사용하여 인스턴스를 시작하는 계정에는 AMI가 시작하는 인스턴스에 대한 요금이 청구됩니다.

조직 또는 조직 단위의 ARN 가져오기

조직 및 조직 단위 ARN에는 12자리 관리 계정 번호가 포함되어 있습니다. 관리 계정 번호를 모르는 경우 각각에 대한 ARN을 가져올 조직 및 조직 단위를 설명할 수 있습니다. 다음 예제에서 123456789012는 관리 계정의 계정 ID입니다.

필수 권한

ARN을 가져오기 전에 조직 및 조직 단위를 설명할 수 있는 권한이 있어야 합니다. 다음 샘플 정책은 필수 권한을 제공합니다.

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

AWS CLI

조직의 ARN을 가져오려면

describe-organization 명령을 사용합니다. 조직 ARN만 반환하는 `--query` 옵션을 추가합니다.

```
aws organizations describe-organization --query 'Organization.Arn'
```

출력의 예시는 다음과 같습니다.

```
"arn:aws:organizations::123456789012:organization/o-1234567abc"
```

조직의 ARN을 가져오려면

describe-organizational-unit 명령을 사용합니다. `--query` 파라미터를 사용하여 조직 단위 ARN만 반환합니다.

```
aws organizations describe-organizational-unit \
```

```
    --organizational-unit-id ou-a123-b4567890 \
    --query 'OrganizationalUnit.Arn'
```

출력의 예시는 다음과 같습니다.

```
"arn:aws:organizations::123456789012:ou/o-1234567abc/ou-a123-b4567890"
```

PowerShell

조직의 ARN을 가져오려면

Get-ORGOrganization cmdlet을 사용합니다.

```
(Get-ORGOrganization).Arn
```

출력의 예시는 다음과 같습니다.

```
arn:aws:organizations::123456789012:organization/o-1234567abc
```

조직의 ARN을 가져오려면

Get-ORGOrganizationalUnit cmdlet을 사용합니다.

```
(Get-ORGOrganizationalUnit -OrganizationalUnitId "ou-a123-b4567890").Arn
```

출력의 예시는 다음과 같습니다.

```
arn:aws:organizations::123456789012:ou/o-1234567abc/ou-a123-b4567890
```

조직 및 OU가 KMS 키를 사용할 수 있도록 허용

암호화된 스냅샷에서 지원하는 AMI를 공유하는 경우 조직 또는 조직 단위(OU)가 스냅샷을 암호화하는 데 사용되는 KMS 키를 사용하도록 허용해야 합니다.

> **Note**
>
> 암호화된 스냅샷은 고객 관리형 키로 암호화해야 합니다. 기본 AWS 관리형 키로 암호화된 스냅샷의 지원을 받는 AMI는 공유할 수 없습니다.

KMS 키에 대한 액세스를 제어하려면 키 정책에서 aws:PrincipalOrgID 및 aws:PrincipalOrgPaths 조건 키를 사용하여 지정된 작업에 대한 특정 보안 주체 권한만 허용할 수 있습니다. 보안 주체는 사용자, IAM 역할, 페더레이션 사용자 또는 AWS 계정 루트 사용자일 수 있습니다.

조건 키는 다음과 같이 사용됩니다.

- aws:PrincipalOrgID - 지정된 ID로 표시되는 조직에 속한 모든 보안 주체를 허용합니다.
- aws:PrincipalOrgPaths – 지정된 경로로 표시되는 OU에 속한 모든 보안 주체를 허용합니다.

조직(해당 조직에 속한 OU 및 계정 포함)에 KMS 키를 사용할 수 있는 권한을 부여하려면 다음 명령문을 키 정책에 추가합니다.

```
{
    "Sid": "Allow access for organization root",
    "Effect": "Allow",
    "Principal": "*",
    "Action": [
        "kms:Describe*",
        "kms:List*",
        "kms:Get*",
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:CreateGrant"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:PrincipalOrgID": "o-123example"
        }
    }
}
```

특정 OU(및 KMS 키에 속한 계정)에 KMS 키를 사용할 수 있는 권한을 부여하려면 다음 예제와 유사한 정책을 사용할 수 있습니다.

```
{
        "Sid": "Allow access for specific OUs and their descendants",
        "Effect": "Allow",
```

```
        "Principal": "*",
        "Action": [
            "kms:Describe*",
            "kms:List*",
            "kms:Get*",
            "kms:Encrypt",
            "kms:Decrypt",
            "kms:ReEncrypt*",
            "kms:GenerateDataKey*",
            "kms:CreateGrant"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:PrincipalOrgID": "o-123example"
            },
            "ForAnyValue:StringLike": {
                "aws:PrincipalOrgPaths": [
                    "o-123example/r-ab12/ou-ab12-33333333/*",
                    "o-123example/r-ab12/ou-ab12-22222222/*"
                ]
            }
        }
}
```

더 많은 예시 조건문은 IAM 사용 설명서의 aws:PrincipalOrgID 및 aws:PrincipalOrgPaths 섹션을 참조하세요.

계정 간 액세스 권한에 대한 내용은 AWS Key Management Service개발자 안내서의 다른 계정의 사용자가 KMS를 사용하도록 허용을 참조하세요.

조직 또는 OU와 AMI 공유 관리

조직 및 조직 단위(OU)와의 AMI 공유를 관리하여 해당 조직 및 조직 단위가 Amazon EC2 인스턴스를 시작할 수 있는지 여부를 제어할 수 있습니다.

AMI가 공유되는 조직 및 OU 보기

AMI를 공유한 조직 및 OU를 확인할 수 있습니다.

Console

AMI를 공유한 조직 및 OU를 확인하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 AMI를 선택합니다.
3. 목록에서 AMI를 선택하고 권한(Permissions) 탭을 선택한 다음 공유 조직/OU(Shared organizations/OUs)까지 아래로 스크롤합니다.

   공유되는 AMI를 찾으려면 Amazon EC2 인스턴스에 사용할 공유 AMI 찾기 단원을 참조하세요.

AWS CLI

AMI를 공유한 조직 및 OU를 확인하려면

describe-image-attribute 명령을 `launchPermission` 속성과 함께 사용합니다.

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

다음은 응답의 예입니다.

```
{
    "ImageId": "ami-0abcdef1234567890",
    "LaunchPermissions": [
        {
            "OrganizationalUnitArn": "arn:aws:organizations::111122223333:ou/
o-123example/ou-1234-5example"
        }
    ]
}
```

PowerShell

AMI를 공유한 조직 및 OU를 확인하려면

Get-EC2ImageAttribute cmdlet을 사용합니다.

```
Get-EC2ImageAttribute `
```

```
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission
```

조직 또는 OU와 AMI 공유

AMI를 조직 또는 OU와 공유할 수 있습니다.

> **Note**
>
> AMI를 공유하기 위해서 해당 AMI의 레퍼런스인 Amazon EBS 스냅샷을 함께 공유할 필요는 없습니다. AMI만 공유하면 됩니다. 그러면 시스템이 시작에 필요한 EBS 스냅샷 액세스 권한을 자동으로 인스턴스에 제공합니다. 그러나 AMI가 참조하는 스냅샷을 암호화하는 데 사용되는 KMS 키를 공유해야 합니다. 자세한 내용은 조직 및 OU가 KMS 키를 사용할 수 있도록 허용 단원을 참조하세요.

Console

조직 또는 OU와 AMI를 공유하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 AMI를 선택합니다.
3. 목록에서 AMI를 선택한 후 작업(Actions), AMI 권한 수정(Edit AMI permissions)을 선택합니다.
4. AMI 가용성(AMI availability)에서 프라이빗(Private)을 선택합니다.
5. 공유 조직/OU(Shared organizations/OUs) 옆에서 조직/OU ARN 추가(Add organization/OU ARN)를 선택합니다.
6. 조직/OU ARN(Organization/OU ARN)에 AMI를 공유할 조직 ARN 또는 OU ARN을 입력한 다음 AMI 공유(Share AMI)를 선택합니다. ID뿐 아니라 전체 ARN을 지정해야 합니다.

   이 AMI를 다수의 조직 또는 OU와 공유하려면 이 단계를 반복하여 필요한 모든 조직 또는 OU를 추가합니다.
7. 작업을 마쳤으면 [변경 사항 저장(Save changes)]을 선택합니다.
8. (선택 사항) AMI를 공유한 조직 또는 OU를 보려면 목록에서 AMI를 선택하고 권한(Permissions) 탭을 선택한 다음 공유 조직/OU(Shared organizations/OUs)까지 아래로 스크롤합니다. 공유되는 AMI를 찾으려면 Amazon EC2 인스턴스에 사용할 공유 AMI 찾기 섹션을 참조하세요.

AWS CLI

조직과 AMI를 공유하려면

modify-image-attribute 명령을 사용하여 지정한 조직에 특정 AMI의 시작 권한을 부여합니다.

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --launch-permission
 "Add=[{OrganizationArn=arn:aws:organizations::123456789012:organization/
o-123example}]"
```

OU와 AMI를 공유하려면

modify-image-attribute 명령은 지정한 OU에 특정 AMI의 시작 권한을 허용하는 데 사용됩니다. ID 뿐 아니라 전체 ARN을 지정해야 합니다.

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --launch-permission
 "Add=[{OrganizationalUnitArn=arn:aws:organizations::123456789012:ou/o-123example/
ou-1234-5example}]"
```

PowerShell

Edit-EC2ImageAttribute 명령(Tools for Windows PowerShell)을 사용하여 다음 예시와 같이 AMI를 공유할 수 있습니다.

조직 또는 OU와 AMI를 공유하려면

다음 명령은 지정한 조직에 특정 AMI의 시작 권한을 허용하는 데 사용됩니다.

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission `
    -OperationType add `
    -OrganizationArn "arn:aws:organizations::123456789012:organization/o-123example"
```

조직 또는 OU와 AMI 공유를 중단하려면

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

```
Edit-EC2ImageAttribute `
```

```
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission `
    -OperationType remove `
    -OrganizationArn "arn:aws:organizations::123456789012:organization/o-123example"
```

모든 조직, OU 및 AWS 계정과 AMI 공유 중지

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

```
Reset-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission
```

조직 또는 OU와 AMI 공유 중지

조직 또는 OU와의 AMI 공유를 중지할 수 있습니다.

> **Note**
>
> AMI가 공유되는 조직 또는 OU에 있는 경우 특정 계정과의 AMI 공유를 중지할 수 없습니다. 계정에 대한 시작 권한을 제거하여 AMI 공유를 중지하려고 하면 Amazon EC2 성공 메시지를 반환합니다. 그러나 AMI는 계속해서 계정과 공유됩니다.

Console

조직 또는 OU와 AMI 공유를 중단하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 AMI를 선택합니다.
3. 목록에서 AMI를 선택한 후 작업(Actions), AMI 권한 수정(Edit AMI permissions)을 선택합니다.
4. 공유 조직/OU(Shared organizations/OUs)에서 AMI 공유를 중지할 조직 또는 OU를 선택한 다음 선택한 항목 제거(Remove selected)를 선택합니다.
5. 작업을 마쳤으면 [변경 사항 저장(Save changes)]을 선택합니다.
6. (선택 사항) 조직 또는 OU와의 AMI 공유를 중지했는지 확인하려면 목록에서 AMI를 선택하고 권한(Permissions) 탭을 선택한 다음 공유 조직/OU(Shared organizations/OUs)까지 아래로 스크롤합니다.

AWS CLI

조직 또는 OU와 AMI 공유를 중단하려면

modify-image-attribute 명령을 사용합니다. 이 예제는 지정한 조직에서 특정 AMI의 시작 권한을 제거합니다.

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --launch-permission
 "Remove=[{OrganizationArn=arn:aws:organizations::123456789012:organization/
o-123example}]"
```

모든 조직, OU 및 AWS 계정과 AMI 공유 중지

reset-image-attribute 명령을 사용합니다. 이 예제는 지정한 AMI에서 퍼블릭 및 명시적 시작 권한을 모두 제거합니다. AMI의 소유자는 언제나 시작 권한을 가지며 이 명령에 영향을 받지 않습니다.

```
aws ec2 reset-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --attribute launchPermission
```

PowerShell

조직 또는 OU와 AMI 공유를 중단하려면

Edit-EC2ImageAttribute cmdlet을 사용합니다. 이 예제는 지정한 조직에서 특정 AMI의 시작 권한을 제거합니다.

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission `
    -OperationType remove `
    -OrganizationArn "arn:aws:organizations::123456789012:organization/o-123example"
```

모든 조직, OU 및 AWS 계정과 AMI 공유 중지

Reset-EC2ImageAttribute cmdlet을 사용합니다. 이 예제는 지정한 AMI에서 퍼블릭 및 명시적 시작 권한을 모두 제거합니다. AMI의 소유자는 언제나 시작 권한을 가지며 이 명령에 영향을 받지 않습니다.

```
Reset-EC2ImageAttribute `
```

```
    -ImageId ami-0abcdef1234567890 `
    -Attribute LaunchPermission
```

특정 AWS 계정과 AMI 공유

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

AWS 계정 ID는 12자리 숫자(예: 012345678901)이며, AWS 계정을 고유하게 식별합니다. 자세한 내용은 AWS Account Management 참조 안내서의 AWS 계정 식별자 보기를 참조하세요.

고려 사항

특정 AWS 계정과 AMI를 공유하는 경우 다음 사항을 고려하세요.

- 소유권 - AMI를 공유하려면 사용자의 AWS 계정이 AMI를 소유해야 합니다.
- 공유 제한 - 리전 내에서 AMI를 공유할 수 있는 최대 엔터티 수는 Amazon EC2 서비스 할당량을 참조하세요.
- 태그 - 사용자 정의 태그(AMI에 연결하는 태그)는 공유할 수 없습니다. AMI를 공유하면 AMI가 공유되는 AWS 계정에서 사용자 정의 태그를 사용할 수 없습니다.
- 스냅샷 - AMI를 공유하기 위해 해당 AMI의 레퍼런스인 Amazon EBS 스냅샷을 함께 공유할 필요는 없습니다. AMI 자체만 공유할 수 있으며, 시스템에서 시작을 위해 참조된 EBS 스냅샷에 대한 액세스 권한을 인스턴스에 제공합니다. 그러나 AMI가 참조하는 스냅샷을 암호화하는 데 사용되는 모든 KMS 키를 공유해야 합니다. 자세한 내용은 Amazon EBS 사용 설명서의 Share an Amazon EBS snapshot을 참조하세요.
- 암호화 및 키 - 암호화되지 않거나 암호화된 스냅샷의 지원을 받는 AMI를 공유할 수 있습니다.
  - 암호화된 스냅샷은 KMS 키로 암호화해야 합니다. 기본 AWS 관리형 키로 암호화된 스냅샷의 지원을 받는 AMI는 공유할 수 없습니다.
  - 암호화된 스냅샷에서 지원하는 AMI를 공유하는 경우 AWS 계정이 스냅샷을 암호화하는 데 사용된 KMS 키를 사용하도록 허용해야 합니다. 자세한 내용은 조직 및 OU가 KMS 키를 사용할 수 있도록 허용 섹션을 참조하세요. 암호화에 고객 관리형 키를 사용할 때 Auto Scaling 인스턴스를 시작하는 데 필요한 키 정책을 설정하려면 Amazon EC2 Auto Scaling User Guide의 Required AWS KMS key policy for use with encrypted volumes를 참조하세요.
- 리전 - AMI는 리전 리소스입니다. AMI를 공유할 때 해당하는 리전에서만 사용할 수 있습니다. AMI를 다른 리전에서 사용할 수 있도록 하려면 AMI를 해당 리전에 복사한 후 공유하세요. 자세한 내용은 Amazon EC2 AMI 복사 섹션을 참조하세요.

- 사용 - AMI를 공유하면 사용자는 AMI에서만 인스턴스를 시작할 수 있습니다. 삭제, 공유 또는 수정할 수 없습니다. 그러나 AMI를 사용하여 인스턴스를 시작한 후에는 해당 인스턴스에서 AMI를 생성할 수 있습니다.
- 공유 AMI 복사 - 다른 계정의 사용자가 공유 AMI를 복사하려는 경우 AMI를 지원하는 스토리지에 대한 읽기 권한을 해당 사용자에게 부여해야 합니다. 자세한 내용은 교차 계정 복사 섹션을 참조하세요.
- 결제 - 다른 AWS 계정이 인스턴스를 시작하기 위해 AMI를 사용하는 경우에는 요금이 청구되지 않습니다. AMI를 사용하여 인스턴스를 시작하는 계정에는 AMI가 시작하는 인스턴스에 대한 요금이 청구됩니다.

Console

명시적 시작 권한 허용

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 AMI를 선택합니다.
3. 목록에서 AMI를 선택한 후 작업(Actions), 이미지 권한 수정(Edit AMI permissions)을 선택합니다.
4. 프라이빗(Private)을 선택합니다.
5. 공유 계정(Shared accounts)에서 계정 ID 추가(Add account ID)를 선택합니다.
6. AWS 계정 ID에서 AMI를 공유하려는 AWS 계정 ID를 입력한 다음 Share AMI(AMI 공유)를 선택합니다.

   이 AMI를 다수의 계정과 공유하려면 모든 필수 계정 ID가 추가될 때까지 단계 5 및 6을 반복합니다.
7. 작업을 마쳤으면 변경 사항 저장(Save changes)을 선택합니다.
8. (선택 사항) AMI를 공유하는 AWS 계정 계정 ID를 보려면 목록에서 AMI를 선택하고 Permissions(권한) 탭을 선택합니다. 공유되는 AMI를 찾으려면 Amazon EC2 인스턴스에 사용할 공유 AMI 찾기 단원을 참조하세요.

AWS CLI

modify-image-attribute 명령을 사용하여 다음 예시와 같이 AMI를 공유할 수 있습니다.

명시적 시작 권한 허용

다음 예제는 지정된 AWS 계정에 지정된 AMI에 대한 시작 권한을 부여합니다.

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --launch-permission "Add=[{UserId=123456789012}]"
```

특정 계정에서 시작 권한을 제거하는 방법

다음 예제는 지정된 AWS 계정에서 지정된 AMI에 대한 시작 권한을 제거합니다.

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --launch-permission "Remove=[{UserId=123456789012}]"
```

모든 시작 권한을 제거하는 방법

다음 예제는 지정한 AMI에서 퍼블릭 및 명시적 시작 권한을 모두 제거합니다. AMI의 소유자는 언제나 시작 권한을 가지며 이 명령에 영향을 받지 않습니다.

```
aws ec2 reset-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --attribute launchPermission
```

PowerShell

Edit-EC2ImageAttribute 명령(Tools for Windows PowerShell)을 사용하여 다음 예시와 같이 AMI를 공유할 수 있습니다.

명시적 시작 권한 허용

다음 예제는 지정된 AWS 계정에 지정된 AMI에 대한 시작 권한을 부여합니다.

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission `
    -OperationType add `
    -UserId "123456789012"
```

특정 계정에서 시작 권한을 제거하는 방법

다음 예제는 지정된 AWS 계정에서 지정된 AMI에 대한 시작 권한을 제거합니다.

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission -OperationType remove `
    -UserId "123456789012"
```

모든 시작 권한을 제거하는 방법

다음 예제는 지정한 AMI에서 퍼블릭 및 명시적 시작 권한을 모두 제거합니다. AMI의 소유자는 언제나 시작 권한을 가지며 이 명령에 영향을 받지 않습니다.

```
Reset-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission
```

AWS 계정과 AMI 공유 취소

AMI의 시작 권한에 계정을 추가하여 Amazon Machine Image(AMI)를 특정 AWS 계정과 공유할 수 있습니다. AMI가 AWS 계정과 공유된 경우 더 이상 계정과 이를 공유하지 않으려면 AMI의 시작 권한에서 해당 계정을 제거합니다. `cancel-image-launch-permission` AWS CLI 명령을 실행하여 이 작업을 수행할 수 있습니다. 이 명령을 실행하면 지정된 AMI에 대한 시작 권한에서 AWS 계정이 제거됩니다. 사용자의 AWS 계정과 공유되는 AMI를 찾으려면 Amazon EC2 인스턴스에 사용할 공유 AMI 찾기 섹션을 참조하세요.

예를 들어, 사용자와 공유되었지만 사용되지 않거나 사용 중단된 AMI로 인스턴스를 시작할 가능성을 줄이기 위해 사용자 계정과 AMI 공유를 취소할 수 있습니다. 계정과 AMI 공유를 취소하면 해당 AMI는 EC2 콘솔의 AMI 목록이나 describe-images의 출력에 더 이상 표시되지 않습니다.

주제

- 제한 사항
- 계정과 AMI 공유 취소

제한 사항

- 사용자의 AWS 계정과만 공유되는 AMI의 시작 권한에서 사용자의 계정을 제거할 수 있습니다. `cancel-image-launch-permission`을 사용하여 조직 또는 조직 단위(OU)와 공유되는 AMI의 시작 권한에서 계정을 제거하거나 퍼블릭 AMI에 대한 액세스를 제거할 수 없습니다.

- AMI의 시작 권한에서 계정을 영구적으로 제거할 수는 없습니다. AMI 소유자가 AMI를 계정과 다시 공유할 수 있습니다.
- AMI는 리전 리소스입니다. `cancel-image-launch-permission`을 실행할 때 AMI가 있는 리전을 지정해야 합니다. 명령에서 리전을 지정하거나 AWS_DEFAULT_REGION 환경 변수를 사용합니다.
- AWS CLI 및 SDK만 AMI의 시작 권한에서 계정 제거를 지원합니다. EC2 콘솔은 현재 이 작업을 지원하지 않습니다.

계정과 AMI 공유 취소

> **Note**
>
> 계정과 AMI 공유를 취소한 후에는 이를 실행 취소할 수 없습니다. AMI에 대한 액세스 권한을 다시 얻으려면 AMI 소유자가 AMI를 계정과 공유해야 합니다.

AWS CLI

계정과의 AMI 공유를 취소하려면

cancel-image-launch-permission 명령을 사용합니다.

```
aws ec2 cancel-image-launch-permission \
    --image-id ami-0abcdef1234567890 \
    --region us-east-1
```

PowerShell

계정과의 AMI 공유를 취소하려면

Stop-EC2ImageLaunchPermission cmdlet을 사용하세요.

```
Stop-EC2ImageLaunchPermission `
    -ImageId ami-0abcdef1234567890 `
    -Region us-east-1
```

공유 Linux AMI 생성을 위한 권장 사항

AMI의 안정성을 높이고 공격 대상 영역을 최소화하려면 다음 지침을 사용하세요.

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

목차

- 루트 사용자의 암호 방식 원격 로그인 비활성화
- 로컬 루트 액세스 비활성화
- SSH 호스트 키 페어 삭제
- 퍼블릭 키 자격 증명 설치
- sshd DNS 확인 비활성화(선택 사항)
- 민감한 데이터 제거

AWS Marketplace용 AMI를 구축하는 경우에는 AWS Marketplace 판매자 가이드의 AMI 구축 모범 사례에서 지침, 정책 및 모범 사례를 참조하세요.

안전한 AMI 공유에 대한 추가 내용은 다음을 참조하세요.

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

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 참조하세요.

로컬 루트 액세스 비활성화

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

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

> **Note**
>
> 이 명령은 sudo 사용에는 영향을 주지 않습니다.

SSH 호스트 키 페어 삭제

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

시스템에서 다음의 키 파일을 모두 제거합니다.

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

- ssh_host_ed25519_key
- ssh_host_ed25519_key.pub

이런 파일은 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 공유 설명서를 참조하세요.

> **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 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.

> **Note**
>
> 다음 예에서 IP 주소 http://169.254.169.254/는 링크-로컬 주소이며 인스턴스에서만 유효합니다.

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" 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 사용자에게나 적용할 수 있으며, `root` 사용자로 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 참조하세요.

민감한 데이터 제거

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 대처하세요.

- `--exclude` *`directory`*에서 `ec2-bundle-vol` 옵션을 사용해 번들에 포함하지 않아야 할 보안 정보가 담긴 디렉터리나 하위 디렉터리를 선택하지 않는 방법을 권장합니다. 특히, 이미지를 번들링

할 때 모든 사용자 소유 SSH 퍼블릭/프라이빗 키 페어와 SSH `authorized_keys` 파일을 제외하세요. Amazon 퍼블릭 AMI는 이들 파일을 루트 사용자의 경우 `/root/.ssh`에 저장하고 일반 사용자의 경우 `/home/user_name/.ssh/`에 저장합니다. 자세한 내용은 ec2-bundle-vol 섹션을 참조하세요.

- 번들링 전에는 항상 셸 기록을 삭제합니다. 동일한 AMI로 하나 이상의 번들을 업로드하려고 시도하면 셸 기록에 액세스 키가 포함됩니다. 다음 명령은 인스턴스 내에서 번들링을 실시하기 전 마지막 단계로 실행해야 합니다.

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

> **Warning**
>
> 위 경고에서 설명한 shred의 제한은 여기에도 적용됩니다.
> bash는 종료 시점에서 현재 세션의 이력을 디스크에 기록한다는 점을 유의하세요.
> `~/.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 주의하세요.

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

Amazon EventBridge를 사용하여 AMI 이벤트 모니터링

Amazon Machine Image(AMI)의 상태가 변경되면 Amazon EC2는 Amazon EventBridge(이전의 Amazon CloudWatch Events)로 전송되는 이벤트를 생성합니다. 이벤트는 JSON 형식의 기본 EventBridge 이벤트 버스로 전송됩니다. Amazon EventBridge를 사용하여 이러한 이벤트를 감지하고 대응할 수 있습니다. 이벤트에 대한 응답으로 작업을 트리거하는 EventBridge에서 규칙을 생성하여 이를 수행합니다. 예를 들어 AMI 생성 프로세스가 완료된 시점을 감지한 다음 Amazon SNS 주제를 호출하여 사용자에게 이메일 알림을 전송하는 EventBridge 규칙을 생성할 수 있습니다.

AMI가 다음 상태가 되면 Amazon EC2는 `EC2 AMI State Change` 이벤트를 생성합니다.

- `available`
- `failed`

- `deregistered`
- `disabled`

이벤트는 최선의 작업을 기반으로 생성됩니다.

다음 표에는 AMI 작업과 AMI의 가능한 상태가 나열되어 있습니다. 표에서 예는 해당 작업이 실행될 때 AMI의 가능한 상태를 나타냅니다.

AMI 작업	available	failed	deregistered	disabled
CopyImage	예	예		
CreateImage	예	예		
CreateRestoreImageTask	예	예		
DeregisterImage			예	
DisableImage				예
EnableImage	예			
RegisterImage	예	예		

EC2 AMI State Change 이벤트

- 이벤트 세부 정보
- available 이벤트
- failed 이벤트
- deregistered 이벤트
- disabled 이벤트

이벤트 세부 정보

이벤트에서 다음 필드를 사용하여 작업을 트리거하는 규칙을 생성할 수 있습니다.

`"source": "aws.ec2"`

Amazon EC2에서 시작된 이벤트를 식별합니다.

`"detail-type": "EC2 AMI State Change"`

이벤트 이름을 식별합니다.

`"detail": { "ImageId": "ami-0abcdef1234567890", "State": "available", }`

AMI ID와 AMI 상태(`available`, `failed`, `deregistered` 또는 `disabled`)를 제공합니다.

자세한 내용은 Amazon EventBridge 사용 설명서의 다음 섹션을 참조하세요.

- Amazon EventBridge 이벤트
- Amazon EventBridge 이벤트 패턴
- Amazon EventBridge 규칙

Lambda 함수를 생성하고 해당 Lambda 함수를 실행하는 EventBridge 규칙을 생성하는 방법에 대한 튜토리얼은 AWS Lambda 사용자 가이드의 Tutorial: Log the state of an Amazon EC2 instance using EventBridge를 참조하세요.

available 이벤트

다음은 `CreateImage`, `CopyImage`, `RegisterImage`, `CreateRestoreImageTask` 또는 `EnableImage` 작업이 성공한 후 AMI가 `available` 상태가 될 때 Amazon EC2가 생성하는 이벤트의 예입니다.

`"State": "available"`은 작업이 성공했음을 나타냅니다.

```
{
    "version": "0",
    "id": "example-9f07-51db-246b-d8b8441bcdf0",
    "detail-type": "EC2 AMI State Change",
    "source": "aws.ec2",
    "account": "012345678901",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-1",
    "resources": ["arn:aws:ec2:us-east-1::image/ami-0abcdef1234567890"],
    "detail": {
        "RequestId": "example-9dcc-40a6-aa77-7ce457d5442b",
```

```
        "ImageId": "ami-0abcdef1234567890",
        "State": "available",
        "ErrorMessage": ""
    }
}
```

failed 이벤트

다음은 CreateImage, CopyImage, RegisterImage 또는 CreateRestoreImageTask 작업이 성공한 후 AMI가 failed 상태가 될 때 Amazon EC2가 생성하는 이벤트의 예입니다.

다음 필드는 관련 정보를 제공합니다.

- "State": "failed" - 작업이 실패했음을 나타냅니다.
- "ErrorMessage": "" - 실패한 작업의 이유를 제공합니다.

```
{
    "version": "0",
    "id": "example-9f07-51db-246b-d8b8441bcdf0",
    "detail-type": "EC2 AMI State Change",
    "source": "aws.ec2",
    "account": "012345678901",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-1",
    "resources": ["arn:aws:ec2:us-east-1::image/ami-0abcdef1234567890"],
    "detail": {
        "RequestId": "example-9dcc-40a6-aa77-7ce457d5442b",
        "ImageId": "ami-0abcdef1234567890",
        "State": "failed",
        "ErrorMessage": "Description of failure"
    }
}
```

deregistered 이벤트

다음은 DeregisterImage 작업이 성공한 후 AMI가 deregistered 상태가 될 때 Amazon EC2가 생성하는 이벤트의 예입니다. 작업이 실패하면 이벤트가 생성되지 않습니다. DeregisterImage는 동기 작업이기 때문에 모든 실패는 즉시 알려집니다.

"State": "deregistered"는 DeregisterImage 작업이 성공했음을 나타냅니다.

```
{
    "version": "0",
    "id": "example-9f07-51db-246b-d8b8441bcdf0",
    "detail-type": "EC2 AMI State Change",
    "source": "aws.ec2",
    "account": "012345678901",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-1",
    "resources": ["arn:aws:ec2:us-east-1::image/ami-0abcdef1234567890"],
    "detail": {
        "RequestId": "example-9dcc-40a6-aa77-7ce457d5442b",
        "ImageId": "ami-0abcdef1234567890",
        "State": "deregistered",
        "ErrorMessage": ""
    }
}
```

disabled 이벤트

다음은 DisableImage 작업이 성공한 후 AMI가 disabled 상태가 될 때 Amazon EC2가 생성하는 이벤트의 예입니다. 작업이 실패하면 이벤트가 생성되지 않습니다. DisableImage는 동기 작업이기 때문에 모든 실패는 즉시 알려집니다.

"State": "disabled"는 DisableImage 작업이 성공했음을 나타냅니다.

```
{
    "version": "0",
    "id": "example-9f07-51db-246b-d8b8441bcdf0",
    "detail-type": "EC2 AMI State Change",
    "source": "aws.ec2",
    "account": "012345678901",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-1",
    "resources": ["arn:aws:ec2:us-east-1::image/ami-0abcdef1234567890"],
    "detail": {
        "RequestId": "example-9dcc-40a6-aa77-7ce457d5442b",
        "ImageId": "ami-0abcdef1234567890",
        "State": "disabled",
        "ErrorMessage": ""
    }
}
```

AMI 결제 정보 이해

다수의 Amazon Machine Image(AMI) 중에서 선택하여 인스턴스를 시작할 수 있습니다. AMI는 다양한 운영 체제 플랫폼 및 기능을 지원합니다. 인스턴스를 시작할 때 선택한 AMI가 최종 AWS 결제 금액에 미치는 영향을 이해하려면 관련 운영 체제 플랫폼 및 결제 정보를 조사하면 됩니다. 온디맨드 또는 스팟 인스턴스를 시작하거나 예약 인스턴스를 구매하기 전에 이 작업을 수행하세요.

다음은 AMI를 미리 조사하는 것이 필요에 가장 적합한 AMI를 선택하는 데 어떤 도움이 되는지 보여주는 두 가지 예입니다.

- 스팟 인스턴스의 경우 AMI의 [플랫폼 세부 정보(Platform details)]를 사용하여 해당 AMI가 스팟 인스턴스에 대해 지원되는지 확인할 수 있습니다.
- 예약 인스턴스를 구매하는 경우 AMI의 [플랫폼 세부 정보(Platform details)]에 매핑되는 운영 체제 플랫폼([플랫폼(Platform)])을 선택할 수 있습니다.

인스턴스 요금에 대한 자세한 내용은 Amazon EC2 요금을 참조하세요.

목차

- AMI 결제 정보 필드
- AMI 결제 및 사용 세부 정보 찾기
- 청구서의 AMI 요금 확인

AMI 결제 정보 필드

다음 필드는 AMI와 연결된 결제 정보를 제공합니다.

플랫폼 세부 정보

AMI의 결제 코드와 연결된 플랫폼 세부 정보입니다. 예를 들어 `Red Hat Enterprise Linux`입니다.

사용 작업

AMI와 연결된 Amazon EC2 인스턴스 및 결제 코드의 작업입니다. 예를 들어 `RunInstances:0010`입니다. [사용 작업(Usage operation)]은 AWS 비용 및 사용 보고서(CUR) 및 AWS 가격 목록 API의 lineitem/Operation 열에 해당합니다.

Amazon EC2 콘솔의 인스턴스 또는 AMI 페이지나 describe-images 또는 Get-EC2Image 명령에서 반환되는 응답에서 이러한 필드를 볼 수 있습니다.

샘플 데이터: 플랫폼별 사용 작업

다음 표에는 Amazon EC2 콘솔의 인스턴스 또는 AMI 페이지나 describe-images 또는 Get-EC2Image 명령에서 반환된 응답에 표시될 수 있는 플랫폼 세부 정보와 사용 작업 값이 나와 있습니다.

플랫폼 세부 정보	사용 작업 [2]
Linux/UNIX	RunInstances
Red Hat BYOL Linux	RunInstances:00g0 [3]
Red Hat Enterprise Linux	RunInstances:0010
Red Hat Enterprise Linux with HA	RunInstances:1010
Red Hat Enterprise Linux with SQL Server Standard and HA	RunInstances:1014
Red Hat Enterprise Linux with SQL Server Enterprise and HA	RunInstances:1110
Red Hat Enterprise Linux with SQL Server Standard	RunInstances:0014
Red Hat Enterprise Linux with SQL Server Web	RunInstances:0210
Red Hat Enterprise Linux with SQL Server Enterprise	RunInstances:0110
SQL Server Enterprise	RunInstances:0100
SQL Server Standard	RunInstances:0004

플랫폼 세부 정보	사용 작업 [2]
SQL Server Web	RunInstances:0200
SUSE Linux	RunInstances:000g
Ubuntu Pro	RunInstances:0g00
Windows	RunInstances:0002
Windows BYOL	RunInstances:0800
Windows with SQL Server Enterprise [1]	RunInstances:0102
Windows with SQL Server Standard [1]	RunInstances:0006
Windows with SQL Server Web [1]	RunInstances:0202

[1] AMI에 두 개의 소프트웨어 라이선스가 연결된 경우 플랫폼 세부 정보 필드에 모두 표시됩니다.

[2] 스팟 인스턴스를 실행 중인 경우 AWS 비용 및 사용 보고서의 lineitem/Operation은 여기에 나열된 사용 작업 값과 다를 수 있습니다. 예를 들어, `lineitem/Operation`에 `RunInstances:0010:SV006`이 표시되는 경우 Amazon EC2가 미국 동부(버지니아 북부)의 영역 6에서 Red Hat Enterprise Linux 스팟 인스턴스 시간을 실행하고 있음을 의미합니다.

[3] 사용 보고서에 RunInstances (Linux/UNIX)로 표시됩니다.

AMI 결제 및 사용 세부 정보 찾기

다음은 청구서의 AMI 요금을 확인하는 데 도움이 되는 속성입니다.

- 플랫폼 세부 정보
- 사용 작업
- AMI ID

Console

특정 AMI의 AMI 결제 정보를 찾는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 AMI를 선택합니다.
3. AMI를 선택합니다.
4. 세부 정보 탭에서 플랫폼 세부 정보 및 사용 작업을 찾습니다.

특정 인스턴스의 AMI 결제 정보를 찾는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다.
4. 세부 정보 탭에서 인스턴스 세부 정보를 확장하고 플랫폼 세부 정보 및 사용 작업을 찾습니다.

AWS CLI

특정 AMI의 AMI 결제 정보를 찾는 방법

describe-images 명령을 사용합니다.

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query "Images[].
{PlatformDetails:PlatformDetails,UsageOperation:UsageOperation}"
```

다음은 Linux AMI의 예시 출력입니다.

```
[
    {
        "PlatformDetails": "Linux/UNIX",
        "UsageOperation": "RunInstances"
    }
]
```

특정 인스턴스의 AMI 결제 정보를 찾는 방법

아래와 같이 describe-instances 명령을 사용합니다.

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query "Reservations[].Instances[].
{PlatformDetails:PlatformDetails,UsageOperation:UsageOperation}"
```

다음은 Windows 인스턴스의 예시 출력입니다.

```
[
    {
        "PlatformDetails": "Windows",
        "UsageOperation": "RunInstances:0002"
    }
]
```

PowerShell

특정 AMI의 AMI 결제 정보를 찾는 방법

Get-EC2Image cmdlet을 사용합니다.

```
Get-EC2Image `
    -ImageId ami-0abcdef1234567890 | `
    Format-List PlatformDetails, UsageOperation
```

다음은 Linux AMI의 예시 출력입니다.

```
PlatformDetails : Linux/UNIX
UsageOperation  : RunInstances
```

특정 인스턴스의 AMI 결제 정보를 찾는 방법

Get-EC2Instance cmdlet을 사용합니다.

```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances | `
    Format-List PlatformDetails, UsageOperation
```

다음은 Windows 인스턴스의 예시 출력입니다.

```
PlatformDetails : Windows
```

```
UsageOperation  : RunInstances:0002
```

청구서의 AMI 요금 확인

계획되지 않은 비용이 발생하지 않도록 AWS 비용 및 사용 보고서(CUR)의 인스턴스에 대한 결제 정보가 인스턴스 시작에 사용한 AMI에 연결된 결제 정보와 일치하는지 확인할 수 있습니다.

결제 정보를 확인하려면 CUR에서 인스턴스 ID를 찾아 `lineitem/Operation` 열에서 해당 값을 확인합니다. 이 값은 AMI에 연결된 [사용 작업(Usage operation)] 값과 일치해야 합니다.

예를 들어 `ami-0123456789EXAMPLE` AMI의 결제 정보는 다음과 같습니다.

- 플랫폼 세부 정보 = `Red Hat Enterprise Linux`
- 사용 작업 = `RunInstances:0010`

이 AMI를 사용하여 인스턴스를 시작한 경우 CUR에서 인스턴스 ID를 찾아 `lineitem/Operation` 열에서 해당 값을 확인할 수 있습니다. 이 예시에서 값은 `RunInstances:0010`이어야 합니다.

Amazon EC2의 AMI 할당량

다음 할당량이 AMI 생성 및 공유에 적용됩니다. 할당량은 AWS 리전별로 적용됩니다.

할당량 이름	설명	리전별 기본 할당량
AMI	리전별로 허용되는 최대 퍼블릭 및 프라이빗 AMI 수입니다. 여기에는 사용 가능, 보류 중 및 비활성화된 AMI 및 휴지통의 AMI가 포함됩니다.	50,000
퍼블릭 AMI	리전별로 허용되는 최대 퍼블릭 AMI 수입니다(휴지통의 퍼블릭 AMI 포함).	5
AMI 공유	리전 내에서 AMI를 공유할 수 있는 최대 엔터티(조직, 조직 단위(OU), 계정) 수입니다. 조	1,000

할당량 이름	설명	리전별 기본 할당량
	직 또는 OU와 AMI를 공유하는 경우 조직 또는 OU의 계정 수는 할당량에 포함되지 않습니다.	

할당량을 초과했는데 더 많은 AMI를 생성하거나 공유하려는 경우 다음과 같이 하면 됩니다.

- 총 AMI 할당량 또는 퍼블릭 AMI 할당량을 초과한 경우 사용하지 않은 이미지의 등록 취소를 고려하세요.
- 퍼블릭 AMI 할당량을 초과한 경우 하나 이상의 퍼블릭 AMI를 프라이빗으로 전환하는 것을 고려하세요.
- AMI 공유 할당량을 초과한 경우 별도의 계정 대신 조직 또는 OU와 AMI를 공유하는 것을 고려하세요.
- AMI의 할당량 증가를 요청하세요.

AMI의 할당량 증가 요청

AMI의 기본 할당량보다 더 많이 필요한 경우 할당량 증가를 요청할 수 있습니다.

AMI의 할당량 증가 요청 방법

1. https://console.aws.amazon.com/servicequotas/에서 Service Quotas 콘솔을 엽니다.
2. 탐색 창에서 AWS 서비스를 선택합니다.
3. 목록에서 Amazon Elastic Compute Cloud(Amazon EC2)를 선택하거나 검색 상자에 서비스 이름을 입력합니다.
4. 증가를 요청할 AMI 할당량을 선택합니다. 선택할 수 있는 AMI 할당량은 다음과 같습니다.
   - AMI
   - 퍼블릭 AMI
   - AMI 공유
5. 할당량 증가 요청을 선택합니다.
6. Change quota value(할당량 값 변경)에 새 할당량 값을 입력한 다음, Request(요청)를 선택합니다.

보류 중이거나 최근에 해결된 요청을 보려면 탐색 창에서 대시보드를 선택합니다. 보류 중인 요청의 경우 요청 상태를 선택하여 요청 접수증을 엽니다. 요청의 초기 상태는 Pending(보류 중)입니다. 상태가 Quota requested(할당량 요청됨)로 변경되면 Support Center case number(지원 센터 케이스 번호) 아래에 케이스 번호가 표시됩니다. 이 케이스 번호를 선택하여 요청의 티켓을 엽니다.

요청이 해결되면 할당량에 대한 적용된 할당량 값이 새 값으로 설정됩니다.

자세한 내용은 Service Quotas 사용 설명서를 참조하세요.

Amazon EC2 인스턴스

Amazon EC2 인스턴스는 AWS 클라우드 환경의 가상 서버입니다. 인스턴스를 처음 시작할 때(인스턴스 시작)부터 삭제할 때(인스턴스 종료)까지 인스턴스를 완전히 제어할 수 있습니다. 인스턴스를 시작할 때 다양한 운영 체제 중에서 선택할 수 있습니다. 인스턴스에 연결하고 필요에 맞게 인스턴스를 사용자 지정할 수 있습니다. 예를 들어, 운영 체제를 구성하고, 운영 체제 업데이트를 설치하고, 인스턴스에 애플리케이션을 설치할 수 있습니다.

Amazon EC2는 다양한 인스턴스 유형을 제공합니다. 애플리케이션을 실행하는 데 필요한 컴퓨팅 리소스, 메모리, 스토리지 및 네트워크 성능을 제공하는 인스턴스 유형을 선택할 수 있습니다.

Amazon EC2에서는 사용한 만큼만 지불하면 됩니다. 인스턴스에 대한 요금 청구는 인스턴스를 시작하고 실행 상태로 전환될 때 시작됩니다. 인스턴스를 중지하면 청구가 중지되고 인스턴스를 시작하면 청구가 재개됩니다. 인스턴스를 종료하면 종료 중 상태로 전환되면서 요금 청구가 중지됩니다.

Amazon EC2는 인스턴스의 성능과 비용을 최적화하는 데 사용할 수 있는 기능을 제공합니다. 예를 들어 Amazon EC2 Fleet 또는 Amazon EC2 Auto Scaling을 사용하여 인스턴스 사용률 변화에 따라 용량을 늘리거나 줄일 수 있습니다. 스팟 인스턴스 또는 절감형 플랜을 사용하여 인스턴스 비용을 줄일 수 있습니다.

관리형 인스턴스는 Amazon EKS Auto Mode와 같은 서비스 제공업체가 관리합니다. 관리형 인스턴스의 설정은 직접 수정할 수 없습니다. 관리형 인스턴스는 관리형 필드에서 true 값으로 식별됩니다. 자세한 내용은 Amazon EC2 관리형 인스턴스 섹션을 참조하세요.

기능 및 태스크

- Amazon EC2 인스턴스 유형
- Amazon EC2 관리형 인스턴스
- Amazon EC2 결제 및 구매 옵션
- Amazon EC2 시작 템플릿에 인스턴스 시작 파라미터 저장
- Amazon EC2 인스턴스 시작하기
- EC2 인스턴스에 연결
- Amazon EC2 인스턴스 상태 변경
- 자동 인스턴스 복구
- 인스턴스 메타데이터를 사용하여 EC2 인스턴스를 관리합니다.
- 호스트가 EC2 인스턴스인지 감지
- Amazon EC2 인스턴스의 인스턴스 자격 증명 문서

- EC2 인스턴스의 정밀 클럭 및 시간 동기화
- EC2 인스턴스의 디바이스 드라이버 관리
- Amazon EC2 Windows 인스턴스 구성
- EC2 Windows 인스턴스를 새 버전의 Windows Server로 업그레이드
- 자습서: Amazon RDS 데이터베이스에 Amazon EC2 인스턴스 연결

Amazon EC2 인스턴스 유형

인스턴스를 시작할 때 지정하는 인스턴스 유형에 따라 인스턴스에 사용되는 호스트 컴퓨터의 하드웨어가 결정됩니다. 각 인스턴스 유형은 서로 다른 컴퓨팅, 메모리, 스토리지 용량을 제공하며, 이 용량에 따라 한 인스턴스 패밀리로 분류됩니다. 인스턴스에서 실행하려는 애플리케이션 또는 소프트웨어의 요구 사항에 따라 인스턴스 유형을 선택하세요. 특성 및 사용 사례에 대한 자세한 내용은 Amazon EC2 인스턴스 유형을 참조하세요.

Amazon EC2는 CPU, 메모리 및 인스턴스 스토리지와 같은 호스트 컴퓨터의 일부 리소스를 특정 인스턴스에 전용으로 할당합니다. Amazon EC2는 네트워크 및 디스크 하위 시스템과 같은 호스트 컴퓨터의 기타 리소스를 인스턴스 간에 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.

목차

- 사용 가능한 인스턴스 유형
- 하드웨어 사양
- 하이퍼바이저 유형
- AMI 가상화 유형
- Processors
- Amazon EC2 인스턴스 유형 찾기
- EC2 인스턴스 유형 파인더에서 권장 사항 받기

- Compute Optimizer에서 EC2 인스턴스 권장 사항 받기
- Amazon EC2 인스턴스 유형 변경
- 성능 순간 확장 가능 인스턴스
- GPU 인스턴스를 사용한 성능 가속화
- Amazon EC2 Mac 인스턴스
- Amazon EBS 최적화 인스턴스 유형
- Amazon EC2 인스턴스의 CPU 옵션
- Amazon EC2 인스턴스용 AMD SEV-SNP
- Amazon EC2 Linux 인스턴스에 대한 프로세서 상태 제어

사용 가능한 인스턴스 유형

Amazon EC2는 각 사용 사례에 맞게 최적화된 다양한 인스턴스 유형을 제공합니다. 인스턴스 유형은 CPU, 메모리, 스토리지, 네트워킹 용량의 다양한 조합으로 구성되며 애플리케이션에 적합한 리소스 조합을 선택할 수 있는 유연성을 제공합니다. 각 인스턴스 유형에는 하나 이상의 인스턴스 크기가 포함되므로 대상 워크로드의 요구 사항에 맞게 리소스를 확장할 수 있습니다.

인스턴스 유형 명명 규칙

이름은 패밀리, 세대, 프로세서 패밀리, 추가 기능 및 크기에 기반합니다. 자세한 내용은 Amazon EC2 인스턴스 유형 안내서의 명명 규칙을 참조하세요.

인스턴스 유형 찾기

지원되는 리전, 컴퓨팅 리소스, 스토리지 리소스와 같은 요구 사항을 충족하는 인스턴스 유형을 확인하려면 Amazon EC2 인스턴스 유형 가이드에서 Amazon EC2 인스턴스 유형 찾기 및 Amazon EC2 인스턴스 유형 사양을 참조하세요.

하드웨어 사양

인스턴스 유형 사양에 대한 자세한 내용은 Amazon EC2 인스턴스 유형 안내서에서 사양을 참조하세요. 가격 책정에 대한 자세한 내용은 Amazon EC2 온디맨드 가격 책정을 참조하세요.

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 유형을 변경할 수 있습니다. 자세한 내용은 Amazon EC2 인스턴스 유형 변경 섹션을 참조하세요.

하이퍼바이저 유형

Amazon EC2는 Xen 및 Nitro 하이퍼바이저를 지원합니다.

Nitro 기반 인스턴스

- 범용: M5 | M5a | M5ad | M5d | M5dn | M5n | M5zn | M6a | M6g | M6gd | M6i | M6id | M6idn | M6in | M7a | M7g | M7gd | M7i | M7i-flex | M8g | M8gd | T3 | T3a | T4g
- 컴퓨팅 최적화: C5 | C5a | C5ad | C5d | C5n | C6a | C6g | C6gd | C6gn | C6i | C6id | C6in | C7a | C7g | C7gd | C7gn | C7i | C7i-flex | C8g | C8gd
- 메모리 최적화: R5 | R5a | R5ad | R5b | R5d | R5dn | R5n | R6a | R6g | R6gd | R6i | R6idn | R6in | R6id | R7a | R7g | R7gd | R7i | R7iz | R8g | R8gd | U-3tb1 | U-6tb1 | U-9tb1 | U-12tb1 | U-18tb1 | U-24tb1 | U7i-6tb | U7i-8tb | U7i-12tb | U7in-16tb | U7in-24tb | U7in-32tb | U7inh-32tb | X2gd | X2idn | X2iedn | X2iezn | X8g | z1d
- 스토리지 최적화: D3 | D3en | I3en | I4g | I4i | I7i | I7ie | I8g | Im4gn | Is4gen
- 가속 컴퓨팅: DL1 | DL2q | F2 | G4ad | G4dn | G5 | G5g | G6 | G6e | Gr6 | Inf1 | Inf2 | P3dn | P4d | P4de | P5 | P5e | P5en | P6-B200 | Trn1 | Trn1n | Trn2 | Trn2u | VT1
- 고성능 컴퓨팅: Hpc6a | Hpc6id | Hpc7a | Hpc7g
- 이전 세대: A1

지원되는 Nitro 하이퍼바이저 버전에 대한 자세한 내용은 Amazon EC2 인스턴스 유형 가이드에서 네트워크 기능 지원을 참조하세요.

Xen 기반 인스턴스

- 범용: M1 | M2 | M3 | M4 | T1 | T2
- 컴퓨팅 최적화: C1 | C3 | C4
- 메모리 최적화: R3 | R4 | X1 | X1e
- 스토리지 최적화: D2 | H1 | I2 | I3
- 가속화된 컴퓨팅: F1 | G3 | P2 | P3

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 가상화 유형 섹션을 참조하세요.

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.

Processors

EC2 인스턴스는 다양한 프로세서를 지원합니다.

Processors

- 인텔 프로세서
- AMD 프로세서
- AWS Graviton 프로세서
- AWS Trainium
- AWS Inferentia

인텔 프로세서

Intel 프로세서에서 실행되는 Amazon EC2 인스턴스에는 다음과 같은 프로세서 기능이 포함될 수 있습니다. Intel 프로세서에서 실행되는 모든 인스턴스가 이러한 프로세서 기능을 모두 지원하는 것은 아닙니다. 각 인스턴스 유형에 사용할 수 있는 기능에 대한 자세한 내용은 Amazon EC2 인스턴스 유형을 참조하세요.

- 인텔 AES New Instructions(AES-NI) — 인텔 AES-NI 암호화 명령 세트는 더 빠른 데이터 보호와 더 강력한 보안을 제공할 수 있도록 기존 AES(고급 암호화 표준) 알고리즘을 개선합니다. 현재 모든 세대의 EC2 인스턴스에서 이 프로세서 기능을 지원합니다.
- 인텔 Advanced Vector Extensions(인텔 AVX, 인텔 AVX2 및 인텔 AVX-512): — 인텔 AVX 및 인텔 AVX2는 256비트 그리고 인텔 AVX-512는 512비트 명령 세트 확장으로서 FP(부동 소수점) 집약적 애플리케이션을 위해 설계되었습니다. 인텔 AVX Instructions는 이미지 및 오디오/비디오 처리, 과학 시뮬레이션, 재무 분석, 3D 모델링 및 분석과 같은 애플리케이션의 성능을 향상시킵니다. 이 기능은 HVM AMI로 실행된 인스턴스에서만 사용할 수 있습니다.
- 인텔 터보 부스트 기술 — 인텔 터보 부스트 기술 프로세서는 기본 작동 주파수보다 빠른 속도로 코어를 자동으로 실행합니다.

- 인텔 딥 러닝 부스트(인텔 DL 부스트) — AI 딥 러닝 사례를 가속화합니다. 2세대 인텔 제온 확장형 프로세서는 Vector Neural Network Instruction(VNNI/INT8)을 통해 인텔 AVX-512를 확장하므로 이미지 인식/세분화, 객체 감지, 음성 인식, 언어 번역, 추천 시스템, 강화 학습 등의 경우에 이전 세대의 인텔 제온 확장형 프로세서(FP32)보다 딥 러닝 추론 성능을 대폭 강화합니다. VNNI와 호환되지 않는 Linux 배포도 있습니다.

  `M5n`, `R5n`, `M5dn`, `M5zn`, `R5b`, `R5dn`, `D3`, `D3en` 및 `C6i` 인스턴스가 VNNI를 지원합니다. `C5` 및 `C5d` 인스턴스는 `12xlarge`, `24xlarge` 및 `metal` 인스턴스에 대해서만 VNNI를 지원합니다.

64비트 CPU에 대한 업계의 이름 지정 규칙으로 인해 혼란이 발생할 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 인텔 하드웨어에서 실행되고 있음에도 불구하고 Ubuntu나 Windows를 실행하는 인스턴스에 대한 내부 시스템 정보는 CPU 아키텍처를 AMD64로 표시하는 이유가 이 때문입니다.

AMD 프로세서

AMD EPYC 프로세서에서 실행되는 Amazon EC2 인스턴스는 워크로드의 비용과 성능을 모두 최적화하는 데 도움이 될 수 있습니다. 이러한 인스턴스는 다음과 같은 프로세서 기능을 지원할 수 있습니다. AMD 프로세서에서 실행되는 모든 인스턴스가 이러한 프로세서 기능을 모두 지원하는 것은 아닙니다. 각 인스턴스 유형에 사용할 수 있는 기능에 대한 자세한 내용은 Amazon EC2 인스턴스 유형을 참조하세요.

- AMD 보안 메모리 암호화(SME)
- AMD 투명 단일 키 메모리 암호화(TSME)
- AMD 고급 벡터 확장(AVX)
- AMD Secure Encrypted Virtualization-Secure Nested Paging(SEV-SNP)
- 벡터 신경망 지침(VNNI)
- BFloat16

AWS Graviton 프로세서

AWS Graviton은 Amazon EC2 인스턴스에서 실행되는 워크로드에 대해 최고의 가격 대비 성능을 제공하도록 설계된 프로세서 패밀리입니다.

자세한 내용은 Getting started with Graviton을 참조하세요.

AWS Trainium

AWS Trainium 기반 인스턴스는 비용 효율적인 고성능 딥 러닝 훈련을 위해 특별히 개발되었습니다. 이러한 인스턴스를 사용하면 음성 인식, 추천, 사기 탐지, 이미지 및 동영상 분류와 같은 광범위한 애플리케이션 세트 전반에서 사용되는 자연어 처리, 컴퓨터 비전 및 추천 모델을 훈련할 수 있습니다. PyTorch 및 TensorFlow와 같이 널리 사용되는 ML 프레임워크에서 기존 워크플로를 사용할 수 있습니다.

AWS Inferentia

AWS Inferentia 기반 인스턴스는 기계 학습을 가속화하도록 설계되었습니다. 지연 시간이 짧은 고성능 기계 학습 추론을 제공합니다. 이러한 인스턴스는 자연어 처리, 객체 감지 및 분류, 콘텐츠 개인화 및 필터링, 음성 인식과 같은 애플리케이션에 대한 딥 러닝(DL) 모델을 배포하는 데 최적화되어 있습니다.

시작할 수 있는 다양한 방법이 있습니다.

- 기계 학습 모델을 시작하는 가장 쉬운 방법으로 완전 관리형 서비스인 SageMaker AI를 사용합니다. 자세한 내용은 Amazon SageMaker AI 개발자 안내서의 SageMaker 시작하기를 참조하세요.
- 딥 러닝 AMI를 사용하여 Inf1 또는 Inf2 인스턴스를 시작합니다. 자세한 내용은 AWS Deep Learning AMIs 개발자 안내서에서 DLAMI가 포함된 AWS Inferentia를 참조하세요.
- 자체 AMI를 사용하여 Inf1 또는 Inf2 인스턴스를 시작하고 AWS Neuron SDK를 설치하면 AWS Inferentia용 딥 러닝 모델을 컴파일, 실행 및 프로파일링할 수 있습니다.
- Inf1 또는 Inf2 인스턴스 및 Amazon ECS 최적화 AMI를 사용하여 컨테이너 인스턴스를 시작합니다. 자세한 내용은 Amazon Elastic Container Service 개발자 안내서의 Amazon Linux 2(Inferentia) AMI를 참조하세요.
- Inf1 인스턴스를 실행하는 노드가 있는 Amazon EKS 클러스터를 생성합니다. 자세한 내용은 Amazon EKS 사용 설명서에서 Inferentia 지원을 참조하세요.

Amazon EC2 인스턴스 유형 찾기

인스턴스를 시작하려면 먼저 사용할 인스턴스 유형을 선택해야 합니다. 선택한 인스턴스 유형은 컴퓨팅, 메모리 또는 스토리지 리소스 등 워크로드에 필요한 리소스에 따라 다를 수 있습니다. 워크로드에 적합한 여러 인스턴스 유형을 식별하고 테스트 환경에서 성능을 평가하는 것이 유리할 수 있습니다. 로드 상태에서 애플리케이션의 성능 측정을 대신할 수 없습니다.

EC2 인스턴스 유형 찾기를 사용하여 EC2 인스턴스 유형에 대한 제안과 지침을 얻을 수 있습니다. 자세한 내용은 the section called “EC2 인스턴스 유형 파인더” 섹션을 참조하세요.

이미 EC2 인스턴스를 실행 중인 경우 AWS Compute Optimizer를 사용하여 성능 향상, 비용 절감 또는 두 가지 모두를 위해 사용해야 하는 인스턴스 유형에 대한 권장 사항을 확인할 수 있습니다. 자세한 내용은 the section called “Compute Optimizer 권장 사항” 섹션을 참조하세요.

업무

- 콘솔을 사용하여 인스턴스 유형 찾기
- AWS CLI를 사용하여 인스턴스 유형 설명
- AWS CLI를 사용하여 인스턴스 유형 찾기
- Tools for PowerShell을 사용하여 인스턴스 유형 찾기

콘솔을 사용하여 인스턴스 유형 찾기

Amazon EC2 콘솔을 사용하여 필요에 맞는 인스턴스 유형을 찾을 수 있습니다.

콘솔을 사용하여 인스턴스 유형을 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 인스턴스 유형을 선택합니다.
4. (선택 사항) 기본 설정(기어 모양) 아이콘을 선택하여 표시할 인스턴스 유형 속성(예: 온디맨드 Linux 요금)을 선택한 다음 확인을 선택합니다. 또는 인스턴스 유형의 이름을 선택하여 세부 정보 페이지를 열고 콘솔을 통해 사용 가능한 모든 속성을 볼 수 있습니다. 콘솔에 API 또는 명령줄을 통해 사용할 수 있는 모든 속성이 표시되지는 않습니다.
5. 인스턴스 유형 속성을 사용하여 표시된 인스턴스 유형 목록을 필요에 맞는 인스턴스 유형으로만 필터링합니다. 예를 들어, 다음 속성을 기준으로 필터링할 수 있습니다.

    - 가용 영역(Availability zones): 가용 영역, 로컬 영역 또는 Wavelength 영역의 이름입니다. 자세한 내용은 the section called “리전 및 영역” 섹션을 참조하세요.
    - vCPU(vCPUs) 또는 코어(Cores): vCPU 또는 코어의 개수입니다.
    - 메모리(Memory, GiB): 메모리 크기(GiB)입니다.
    - 네트워크 성능(Network performance): 네트워크 성능(Gigabits)입니다.
    - 로컬 인스턴스 스토리지(Local instance storage): 인스턴스 유형에 로컬 인스턴스 스토리지가 있는지 여부를 나타냅니다(`true` | `false`).

6. (선택 사항) 항목별로 비교해 보려면 여러 인스턴스 유형에 대한 확인란을 선택합니다. 비교 항목이 화면 아래쪽에 표시됩니다.
7. (선택 사항) 추가 검토를 위해 인스턴스 유형 목록을 쉼표로 구분된 값(.csv) 파일에 저장하려면 작업(Actions)에서 목록 CSV 다운로드(Download list CSV)를 선택합니다. 이 파일에는 사용자가 설정한 필터와 일치하는 모든 인스턴스 유형이 포함됩니다.
8. (선택 사항) 필요에 맞는 인스턴스 유형을 사용하여 인스턴스를 시작하려면 인스턴스 유형의 확인란을 선택하고 작업(Actions)에서 인스턴스 시작(Launch instance)을 선택합니다. 자세한 내용은 콘솔에서 인스턴스 시작 마법사를 사용하여 EC2 인스턴스 시작 섹션을 참조하세요.

AWS CLI를 사용하여 인스턴스 유형 설명

특정 인스턴스 유형을 설명하기 위해 describe-instance-types 명령을 사용할 수 있습니다.

인스턴스 유형을 자세히 설명

다음 명령은 지정된 인스턴스 유형의 모든 세부 정보를 표시합니다. 출력이 길기 때문에 여기서는 생략합니다.

```
aws ec2 describe-instance-types \
    --instance-types t2.micro \
    --region us-east-2
```

인스턴스 유형 설명 및 출력 필터링

다음 명령은 지정된 인스턴스 유형의 네트워킹 세부 정보를 표시합니다.

```
aws ec2 describe-instance-types \
    --instance-types t2.micro \
    --region us-east-2 \
    --query "InstanceTypes[].NetworkInfo"
```

출력의 예시는 다음과 같습니다.

```
[
    {
        "NetworkPerformance": "Low to Moderate",
        "MaximumNetworkInterfaces": 2,
        "MaximumNetworkCards": 1,
        "DefaultNetworkCardIndex": 0,
```

```
        "NetworkCards": [
            {
                "NetworkCardIndex": 0,
                "NetworkPerformance": "Low to Moderate",
                "MaximumNetworkInterfaces": 2,
                "BaselineBandwidthInGbps": 0.064,
                "PeakBandwidthInGbps": 1.024
            }
        ],
        "Ipv4AddressesPerInterface": 2,
        "Ipv6AddressesPerInterface": 2,
        "Ipv6Supported": true,
        "EnaSupport": "unsupported",
        "EfaSupported": false,
        "EncryptionInTransitSupported": false,
        "EnaSrdSupported": false
    }
]
```

다음 명령은 지정된 인스턴스 유형의 사용 가능한 메모리를 표시합니다.

```
aws ec2 describe-instance-types \
    --instance-types t2.micro \
    --region us-east-2 \
    --query "InstanceTypes[].MemoryInfo"
```

출력의 예시는 다음과 같습니다.

```
[
    {
        "SizeInMiB": 1024
    }
]
```

AWS CLI를 사용하여 인스턴스 유형 찾기

describe-instance-types 및 describe-instance-type-offerings 명령을 사용하여 필요에 맞는 인스턴스 유형을 찾을 수 있습니다.

예시

- 예제 1: 가용 영역으로 인스턴스 유형 찾기

- 예제 2: 가용 메모리 크기로 인스턴스 유형 찾기
- 예제 3: 가용 인스턴스 스토리지로 인스턴스 유형 찾기
- 예제 4: 최대 절전 모드를 지원하는 인스턴스 유형 찾기

예제 1: 가용 영역으로 인스턴스 유형 찾기

다음 예시는 지정된 가용 영역에서 제공되는 인스턴스 유형을 보여줍니다.

```
aws ec2 describe-instance-type-offerings \
    --location-type "availability-zone" \
    --filters "Name=location,Values=us-east-2a" \
    --region us-east-2 \
    --query "InstanceTypeOfferings[*].[InstanceType]" --output text | sort
```

출력은 알파벳순으로 정렬된 인스턴스 유형 목록입니다. 다음은 출력의 시작 부분만 보여줍니다.

```
a1.2xlarge
a1.4xlarge
a1.large
a1.medium
a1.metal
a1.xlarge
c4.2xlarge
   ...
```

예제 2: 가용 메모리 크기로 인스턴스 유형 찾기

다음 예제는 다음 명령을 사용하여 64GiB(65536MiB)의 메모리가 있는 현재 세대 인스턴스 유형만 보여줍니다.

```
aws ec2 describe-instance-types \
    --filters "Name=current-generation,Values=true" "Name=memory-info.size-in-
mib,Values=65536" \
    --region us-east-2 \
    --query "InstanceTypes[*].[InstanceType]" --output text | sort
```

출력은 알파벳순으로 정렬된 인스턴스 유형 목록입니다. 다음은 출력의 시작 부분만 보여줍니다.

```
c5a.8xlarge
c5ad.8xlarge
```

```
c6a.8xlarge
c6g.8xlarge
c6gd.8xlarge
c6gn.8xlarge
c6i.8xlarge
c6id.8xlarge
c6in.8xlarge
   ...
```

예제 3: 가용 인스턴스 스토리지로 인스턴스 유형 찾기

다음 예제에서는 인스턴스 스토어 볼륨이 있는 모든 R7 인스턴스에 대한 인스턴스 스토리지의 총 크기를 보여줍니다.

```
aws ec2 describe-instance-types \
    --filters "Name=instance-type,Values=r7*" "Name=instance-storage-
supported,Values=true" \
    --region us-east-2 \
    --query "InstanceTypes[].[InstanceType, InstanceStorageInfo.TotalSizeInGB]" \
    --output table
```

출력의 예시는 다음과 같습니다.

```
----------------------------
|  DescribeInstanceTypes   |
+----------------+---------+
|  r7gd.xlarge   |  237    |
|  r7gd.8xlarge  |  1900   |
|  r7gd.16xlarge |  3800   |
|  r7gd.medium   |  59     |
|  r7gd.4xlarge  |  950    |
|  r7gd.2xlarge  |  474    |
|  r7gd.metal    |  3800   |
|  r7gd.large    |  118    |
|  r7gd.12xlarge |  2850   |
+----------------+---------+
```

예제 4: 최대 절전 모드를 지원하는 인스턴스 유형 찾기

다음 예제에서는 최대 절전 모드를 지원하는 인스턴스 유형을 보여줍니다.

```
aws ec2 describe-instance-types \
```

```
    --filters "Name=hibernation-supported,Values=true" \
    --region us-east-2 \
    --query "InstanceTypes[*].[InstanceType]" \
    --output text | sort
```

출력은 알파벳순으로 정렬된 인스턴스 유형 목록입니다. 다음은 출력의 시작 부분만 보여줍니다.

```
c4.2xlarge
c4.4xlarge
c4.8xlarge
c4.large
c4.xlarge
c5.12xlarge
c5.18xlarge
c5.2xlarge
c5.4xlarge
c5.9xlarge
...
```

Tools for PowerShell을 사용하여 인스턴스 유형 찾기

Get-EC2InstanceType 및 Get-EC2InstanceTypeOffering cmdlet을 사용하여 필요에 맞는 인스턴스 유형을 찾을 수 있습니다.

예시

- 가용 영역으로 인스턴스 유형 찾기
- 가용 메모리 크기로 인스턴스 유형 찾기
- 가용 인스턴스 스토리지로 인스턴스 유형 찾기
- 최대 절전 모드를 지원하는 인스턴스 유형 찾기

가용 영역으로 인스턴스 유형 찾기

다음 예시는 지정된 가용 영역에서 제공되는 인스턴스 유형을 보여줍니다.

```
(Get-EC2InstanceTypeOffering `
    -LocationType "availability-zone" `
    -Region us-east-2 `
    -Filter @{Name="location"; Values="us-east-2a"}).InstanceType | Sort-Object `
```

가용 메모리 크기로 인스턴스 유형 찾기

다음 예제는 다음 명령을 사용하여 64GiB(65536MiB)의 메모리가 있는 현재 세대 인스턴스 유형만 보여줍니다.

```
(Get-EC2InstanceType `
    -Filter @{Name="current-generation"; Values="true"},
            @{Name="memory-info.size-in-mib"; Values="65536"}).InstanceType | Sort-
Object
```

가용 인스턴스 스토리지로 인스턴스 유형 찾기

다음 예제에서는 인스턴스 스토어 볼륨이 있는 모든 R7 인스턴스에 대한 인스턴스 스토리지의 총 크기를 보여줍니다.

```
Get-EC2InstanceType `
    -Filter @{Name="instance-type"; Values="r7*"},
            @{Name="instance-storage-supported"; Values="true"} | `
     Select InstanceType, @{Name="TotalSizeInGB";
 Expression={($_.InstanceStorageInfo.TotalSizeInGB)}}
```

출력의 예시는 다음과 같습니다.

```
InstanceType  TotalSizeInGB
------------  -------------
r7gd.8xlarge           1900
r7gd.16xlarge          3800
r7gd.xlarge             237
r7gd.4xlarge            950
r7gd.medium              59
r7gd.2xlarge            474
r7gd.large              118
r7gd.metal             3800
r7gd.12xlarge          2850
```

최대 절전 모드를 지원하는 인스턴스 유형 찾기

다음 예제에서는 최대 절전 모드를 지원하는 인스턴스 유형을 보여줍니다.

```
(Get-EC2InstanceType `
    -Filter @{Name="hibernation-supported"; Values="true"}).InstanceType | Sort-Object
```

EC2 인스턴스 유형 파인더에서 권장 사항 받기

EC2 인스턴스 유형 찾기는 사용 사례, 워크로드 유형, CPU 제조업체 선호도, 가격 및 성능 우선순위 지정 방법은 물론 지정할 수 있는 추가 파라미터를 고려합니다. 그런 다음 이 데이터를 사용하여 새 워크로드에 가장 적합한 Amazon EC2 인스턴스 유형에 대한 제안과 지침을 제공합니다.

사용 가능한 인스턴스 유형이 너무 많기 때문에 워크로드에 적합한 인스턴스 유형을 찾는 데 시간이 많이 걸리고 복잡할 수 있습니다. EC2 인스턴스 유형 찾기를 사용하면 최신 인스턴스 유형을 최신 상태로 유지하고 워크로드를 위한 최적의 가격 대비 성능을 달성할 수 있습니다.

Amazon EC2 콘솔을 사용하여 EC2 인스턴스 유형에 대한 제안과 지침을 받을 수 있습니다. Amazon Q로 직접 이동하여 인스턴스 유형 조언을 구할 수도 있습니다. 자세한 내용은 Amazon Q Developer 사용 설명서를 참조하세요.

기존 워크로드에 대한 인스턴스 유형 권장 사항을 찾고 있는 경우 AWS Compute Optimizer를 사용합니다. 자세한 내용은 Compute Optimizer에서 EC2 인스턴스 권장 사항 받기 단원을 참조하십시오.

EC2 인스턴스 유형 찾기 사용

Amazon EC2 콘솔에서 인스턴스 시작 마법사, 시작 템플릿 생성 시 또는 인스턴스 유형 페이지의 EC2 인스턴스 유형 찾기에서 인스턴스 유형 제안을 받을 수 있습니다.

다음 지침에 따라 Amazon EC2 콘솔의 EC2 인스턴스 유형 찾기를 사용하여 EC2 인스턴스 유형에 대한 제안과 지침을 확인하세요. 이 단계의 애니메이션을 보려면 애니메이션 보기: EC2 인스턴스 유형 찾기를 사용하여 인스턴스 유형 제안 받기 섹션을 참조하세요.

EC2 인스턴스 유형 찾기를 사용하여 인스턴스 유형 제안을 받으려면 다음을 수행합니다

1. 다음 중 하나를 사용하여 프로세스를 시작합니다.
   - 인스턴스 시작 절차를 따릅니다. 인스턴스 유형 옆에서 조언 받기 링크를 선택합니다.
   - 절차에 따라 시작 템플릿을 생성합니다. 인스턴스 유형 옆에서 조언 받기 링크를 선택합니다.
   - 탐색 창에서 인스턴스 유형을 선택한 다음 인스턴스 유형 찾기 버튼을 선택합니다.
2. 인스턴스 유형 선택에 대한 조언 받기 화면에서 다음을 수행합니다.
   a. 워크로드 유형, 사용 사례, 우선순위, CPU 제조업체에 대한 옵션을 선택하여 인스턴스 유형 요구 사항을 지정합니다.
   b. (선택 사항) 워크로드에 대해 더 자세한 요구 사항을 지정하려면 다음을 수행합니다.
      i. 고급 파라미터를 확장합니다.

ii. 파라미터를 추가하려면 파라미터를 선택하고 추가를 선택한 다음 파라미터에 대한 값을 지정합니다. 추가하려는 각 파라미터에 대해 이 과정을 반복합니다. 최소값 또는 최대값을 표시하지 않으려면 필드를 비워둡니다.

iii. 파라미터를 추가한 후 제거하려면 파라미터 옆의 X를 선택합니다.

c. 인스턴스 유형 조언 받기를 선택합니다.

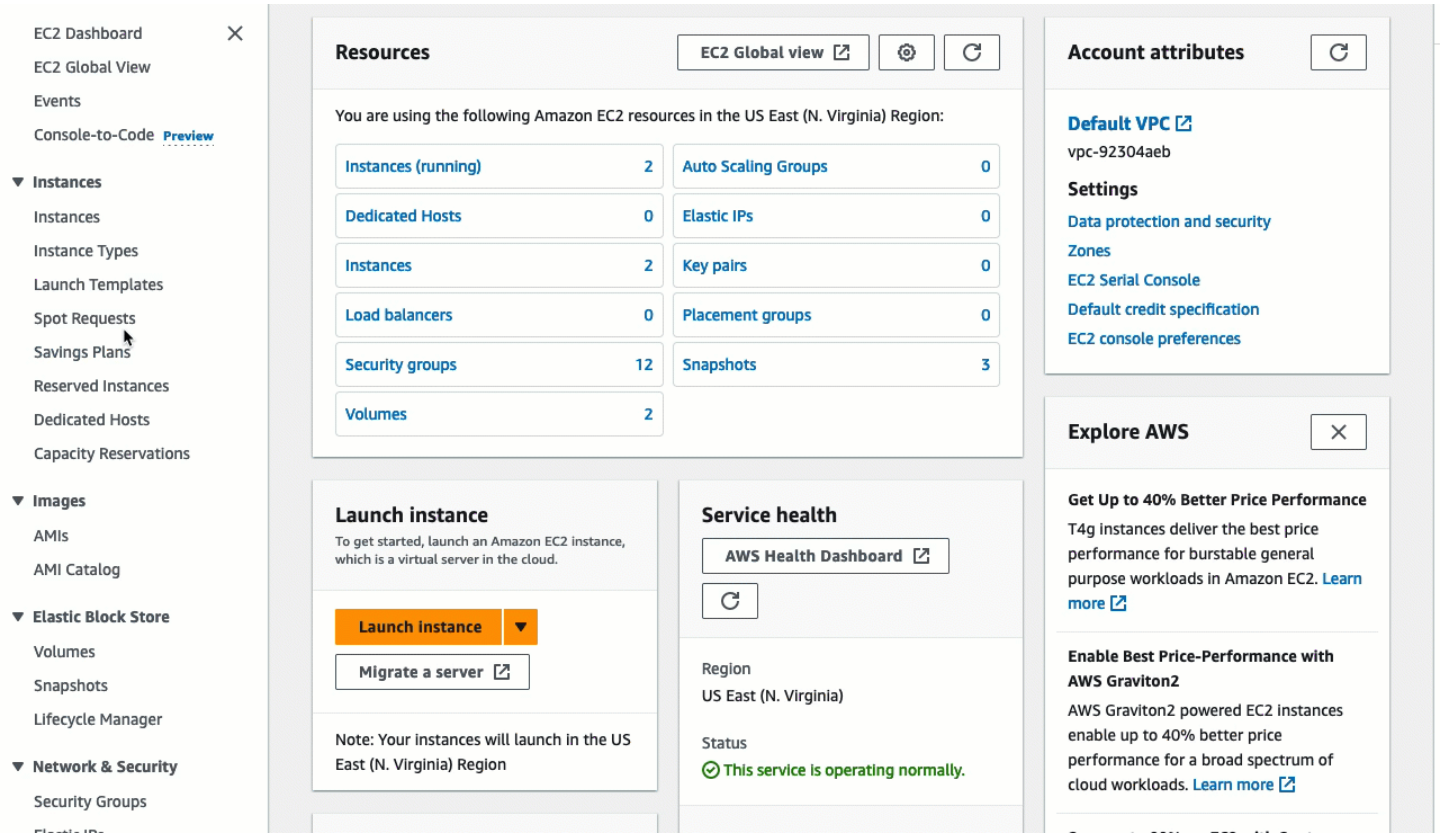
Amazon EC2는 지정된 요구 사항과 일치하는 인스턴스 패밀리를 제안합니다.

3. 제안된 인스턴스 패밀리 내의 각 인스턴스 유형에 대한 세부 정보를 보려면 권장 인스턴스 패밀리 세부 정보 보기를 선택합니다.

4. 요구 사항을 충족하는 인스턴스 유형을 선택한 다음 작업, 인스턴스 시작 또는 작업, 시작 템플릿 생성을 선택합니다.

또는 인스턴스 시작 마법사 또는 시작 템플릿 페이지에서 프로세스를 시작한 후 원래 흐름으로 돌아가고 싶다면 사용하려는 인스턴스 유형을 기록해 두세요. 그런 다음 인스턴스 시작 마법사 또는 시작 템플릿의 인스턴스 유형에서 인스턴스 유형을 선택하고 절차를 완료하여 인스턴스를 시작하거나 시작 템플릿을 생성합니다.

애니메이션 보기: EC2 인스턴스 유형 찾기를 사용하여 인스턴스 유형 제안 받기



Compute Optimizer에서 EC2 인스턴스 권장 사항 받기

AWS Compute Optimizer는 성능 향상, 비용 절감 또는 두 가지 모두에 도움이 되는 Amazon EC2 권장 사항을 제공합니다. 이러한 권장 사항을 사용하여 새 인스턴스 유형으로 변경할지 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. 자세한 내용은 AWS Compute Optimizer 사용 설명서의 Amazon EC2 instance metrics를 참조하세요.

내용

- 요구 사항
- 결과 분류
- 권장 사항 보기
- 권장 사항 평가를 위한 고려 사항

요구 사항

Compute Optimizer에서 권장 사항을 받으려면 먼저 Compute Optimizer를 옵트인해야 합니다. 자세한 내용은 AWS Compute Optimizer 사용 설명서의 AWS Compute Optimizer 시작하기를 참조하세요.

Compute Optimizer는 일부 인스턴스 유형에 대한 권장 사항을 생성하지만 모든 인스턴스 유형에 대한 권장 사항을 생성하지는 않습니다. 지원되지 않는 인스턴스 유형을 사용하는 경우 Compute Optimizer가 권장 사항을 생성하지 않습니다. 지원되는 인스턴스 유형 목록은 AWS Compute Optimizer 사용 설명서의 Amazon EC2 instance requirements를 참조하세요.

결과 분류

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 12시간 미만이거나, 인스턴스가 실행된 지 30시간 미만이거나, Compute Optimizer에서 지원하지 않는 인스턴스 유형인 경우 이 문제가 생길 수 있습니다.

권장 사항 보기

Compute Optimizer를 옵트인하면 Amazon 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.

Amazon EC2 콘솔을 사용하여 인스턴스의 권장 사항을 보려면 다음을 수행하세요.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances(인스턴스)를 선택합니다.
3. 인스턴스 ID를 선택하여 인스턴스 세부 정보 페이지를 엽니다.
4. 인스턴스 세부 정보 페이지의 상단 요약 섹션에서 AWS Compute Optimizer 조사 결과를 찾습니다. 조사 결과가 있는 경우 조사 결과 분류와 세부 정보를 볼 수 있는 링크가 표시됩니다. 그렇지 않으면 이 인스턴스에 사용할 수 있는 권장 사항이 없음이 표시됩니다.
5. 조사 결과가 있는 경우 세부 정보 보기를 선택합니다. 그러면 Compute Optimizer 콘솔에서 EC2 인스턴스에 대한 권장 사항 페이지가 열립니다. 현재 인스턴스 유형은 현재로 표시됩니다. 옵션 1, 옵션 2, 옵션 3으로 표시된 최대 3개의 인스턴스 유형 권장 사항도 있습니다. 이 페이지에는 인스턴스에 대한 최신 CloudWatch 지표 데이터도 표시됩니다.

모든 리전의 모든 인스턴스에 대한 권장 사항을 보려면 다음을 수행하세요.

Compute Optimizer 콘솔을 사용하여 모든 리전의 모든 Amazon EC2 인스턴스에 대한 권장 사항을 볼 수 있습니다. 자세한 내용은 AWS Compute Optimizer 사용 설명서의 Viewing EC2 instances recommendations와 Viewing EC2 instance details를 참조하세요.

권장 사항 평가를 위한 고려 사항

권장 사항을 받으면 그에 대한 조치를 취할지 여부를 결정해야 합니다. 인스턴스 유형을 변경하기 전에 다음 사항을 고려하세요.

- 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 참조하세요.
- 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 내용은 성능 버스트 가능 인스턴스에 대한 주요 개념 및 정의 섹션을 참조하세요.
- 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 유형 변경을 위한 호환성 섹션을 참조하세요.
- 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.

Amazon EC2 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. 이러한 경우 인스턴스 유형을 변경하여 인스턴스의 크기를 조정할 수 있습니다. 예를 들어 `t2.micro` 인스턴스가 워크로드에 비해 너무 작은 경우 `t2.large`와 같은 더 큰 T2 인스턴스 유형으로 변경하여 크기를 늘릴 수 있습니다. 또는 `m5.large` 등의 다른 인스턴스 유형으로 변경할 수 있습니다. 또는 IPv6 지원과 같은 일부 기능의 장점을 활용하기 위해 이전 세대에서 현재 세대 인스턴스 유형으로 변경할 수도 있습니다.

기존 워크로드를 처리하는 데 가장 적합한 인스턴스 유형에 대한 권장 사항이 필요한 경우 AWS Compute Optimizer를 사용할 수 있습니다. 자세한 내용은 Compute Optimizer에서 EC2 인스턴스 권장 사항 받기 단원을 참조하십시오.

인스턴스 유형을 변경하면 새 인스턴스 유형의 요금이 청구되기 시작합니다. 모든 인스턴스 유형의 온디맨드 요금은 Amazon EC2 온디맨드 요금을 참조하세요.

인스턴스 유형을 변경하지 않고 인스턴스에 스토리지를 추가하려면 인스턴스에 EBS 볼륨을 추가합니다. 자세한 내용은 Amazon EBS 사용 설명서의 Attach an Amazon EBS volume to an instance를 참조하세요.

어떤 지침을 따라야 하나요?

인스턴스 유형 변경에 대한 여러 가지 지침이 있습니다. 사용 지침은 인스턴스의 루트 볼륨과 인스턴트 유형이 인스턴스의 현재 구성과 호환되는지 여부에 따라 다릅니다. 호환성 결정 방법에 대한 자세한 내용은 인스턴스 유형 변경을 위한 호환성 섹션을 참조하세요.

다음 표를 사용하여 따라야 할 지침을 확인하세요.

루트 볼륨	호환성	다음 지침을 따릅니다.
EBS	호환됨	인스턴스 유형 변경
EBS	호환되지 않음	새 인스턴스 유형으로 마이그레이션
인스턴스 스토어	해당 사항 없음	새 인스턴스 유형으로 마이그레이션

인스턴스 유형 변경을 위한 호환성

원하는 인스턴스 유형이 인스턴스의 현재 구성과 호환되는 경우 인스턴스 유형을 변경할 수 있습니다. 원하는 인스턴스 유형이 인스턴스의 현재 구성과 호환되지 않는 경우 인스턴스 유형과 호환되는 구성을 갖는 새 인스턴스를 시작한 다음 애플리케이션을 새 인스턴스로 마이그레이션해야 합니다.

호환성은 다음을 통해 결정됩니다.

가상화 유형

Linux AMI는 PV(반가상화) 또는 HVM(하드웨어 가상 머신)의 두 가지 유형의 가상화를 사용합니다. PV AMI에서 시작한 인스턴스를 시작하는 경우 HVM 전용의 인스턴스 유형으로 변경할 수 없습니다. 자세한 내용은 가상화 유형 단원을 참조하십시오. 인스턴스의 가상화 유형을 확인하려면 Amazon EC2 콘솔에서 Instances(인스턴스) 화면의 세부 정보 창에서 가상화(Virtualization) 값을 확인합니다.

아키텍처

AMI는 프로세서의 아키텍처에 고유하기 때문에 프로세서 아키텍처가 현재 인스턴스 유형과 동일한 인스턴스를 선택해야 합니다. 예:

- 현재 인스턴스 유형에 Arm 아키텍처 기반 프로세서가 탑재된 경우, Arm 아키텍처 기반 프로세서를 지원하는 인스턴스 유형(예: C6g 및 M6g)으로 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 인스턴스 유형을 변경하는 경우 이러한 인스턴스 유형으로 제한됩니다.

네트워크 어댑터

한 네트워크 어댑터에서 다른 네트워크 어댑터로 드라이버를 전환하는 경우 운영 체제에서 새 어댑터를 생성할 때 네트워크 어댑터 설정이 재설정됩니다. 설정을 다시 구성하려면 관리자 권한이 있는 로컬 계정에 액세스해야 할 수 있습니다. 다음은 한 네트워크 어댑터에서 다른 네트워크 어댑터로 전환하는 예입니다.

- AWS PV(T2 인스턴스)에서 인텔 82599 VF(M4 인스턴스)로
- 인텔 82599 VF(대부분의 M4 인스턴스)에서 ENA(M5 인스턴스)로
- ENA(M5 인스턴스)에서 고대역폭 ENA(M5n 인스턴스)로

향상된 네트워킹

향상된 네트워킹을 지원하는 인스턴스 유형을 사용하려면 필요한 드라이버가 설치되어 있어야 합니다. 예를 들어 Nitro 기반 인스턴스에는 Elastic Network Adapter(ENA) 드라이버가 설치된 EBS 지원 AMI가 필요합니다. 향상된 네트워킹을 지원하지 않는 인스턴스 유형에서 향상된 네트워킹을 지원하는 인스턴스 유형으로 변경하려면 인스턴스에 ENA 드라이버 또는 ixgbevf 드라이버를 적절하게 설치해야 합니다.

> **Note**
>
> ENA Express를 활성화한 상태에서 인스턴스의 크기를 조정하는 경우, 새 인스턴스 유형도 ENA Express를 지원해야 합니다. ENA Express를 지원하는 인스턴스 유형의 목록은 ENA Express를 지원하는 인스턴스 유형 섹션을 참조하세요.
> ENA Express를 지원하는 인스턴스 유형에서 지원하지 않는 인스턴스 유형으로 변경하려면, 인스턴스의 크기를 조정하기 전에 ENA Express가 현재 활성화되어 있지 않은지 확인하세요.

NVMe

EBS 볼륨이 Nitro 기반 인스턴스에 NVMe 블록 디바이스로 표시됩니다. NVMe를 지원하지 않는 인스턴스 유형에서 NVMe를 지원하는 인스턴스 유형으로 변경하는 경우 먼저 인스턴스에 NVMe 드라이버를 설치해야 합니다. 또한 블록 디바이스 매핑에서 지정한 디바이스의 디바이스 이름은 NVMe 디바이스 이름(`/dev/nvme[0-26]n1`)을 사용하여 변경됩니다.

[Linux 인스턴스] 따라서 `/etc/fstab`를 사용하여 부팅 시 파일 시스템을 마운트하려면 디바이스 이름 대신 UUID/레이블을 사용해야 합니다.

볼륨 제한

인스턴스에 연결할 수 있는 Amazon EBS 볼륨의 최대 수는 인스턴스 유형 및 인스턴스 크기에 따라 달라집니다. 자세한 내용은 Amazon EC2 인스턴스의 Amazon EBS 볼륨 한도 단원을 참조하십시오.

현재 인스턴스에 연결된 볼륨 수와 같거나 더 많은 볼륨 수를 지원하는 인스턴스 유형이나 인스턴스 크기로만 변경할 수 있습니다. 현재 연결된 볼륨 수를 지원하지 않는 인스턴스 유형이나 인스턴스 크기로 변경하면 요청이 실패합니다. 예를 들어 32개의 볼륨이 연결된 `m7i.4xlarge` 인스턴스에서 최대 27개의 볼륨을 지원하는 `m6i.4xlarge`(으)로 변경하면 요청이 실패합니다.

NitroTPM

NitroTPM이 활성화된 AMI와 NitroTPM을 지원하는 인스턴스 유형을 사용하여 인스턴스를 시작하면 NitroTPM이 활성화된 상태로 인스턴스가 시작됩니다. NitroTPM도 지원하는 인스턴스 유형으로만 변경할 수 있습니다.

Amazon EC2 인스턴스의 인스턴스 유형 변경

필요한 인스턴스 유형이 인스턴스의 현재 구성과 호환되는 경우 다음 지침에 따라 Amazon EBS 지원 인스턴스의 인스턴스 유형을 변경합니다. 자세한 내용은 the section called “호환성” 단원을 참조하십시오.

고려 사항

- 인스턴스를 중단해야 인스턴스 유형을 변경할 수 있습니다. 가동 중지는 인스턴스가 중단되었을 때 계획해야 합니다. 인스턴스 중단하고 인스턴스 유형을 변경하는 것은 몇 분이 걸릴 수 있으며, 인스턴스를 다시 시작하는 시간은 애플리케이션의 시작 스크립트에 따라 달라질 수 있습니다. 자세한 내용은 Amazon EC2 인스턴스 중지 및 시작 단원을 참조하십시오.
- 인스턴스를 중지했다가 시작하면 인스턴스가 새 하드웨어로 이동됩니다. 인스턴스에 엘라스틱 IP가 아닌 퍼블릭 IPv4 주소가 있는 경우 주소를 해제하고 인스턴스에 새 퍼블릭 IPv4 주소를 제공합니다.

인스턴스 수명 주기 전반에 걸친 IP 주소 동작에 대한 자세한 내용은 인스턴스 상태 간 차이을 참조하세요.

- 스팟 인스턴스의 인스턴스 유형은 변경할 수 없습니다.
- [Windows 인스턴스] 인스턴스 유형을 변경하기 전에 AWS PV 드라이버 패키지를 업데이트하는 것이 좋습니다. 자세한 내용은 the section called “PV 드라이버 업그레이드” 단원을 참조하십시오.
- 인스턴스가 Auto Scaling 그룹에 있는 경우, Amazon EC2 Auto Scaling 서비스는 중단된 인스턴스를 비정상으로 간주해 이를 종료하고 대체 인스턴스를 시작합니다. 이를 방지하기 위해서는 인스턴스 유형을 변경하는 동안 그룹에 대한 조정 프로세스를 일시 중지할 수 있습니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Auto Scaling 그룹에 대한 프로세스 일시 중단 및 재개를 참조하세요.
- NVMe 인스턴스 스토어 볼륨이 있는 인스턴스의 인스턴스 유형을 변경하면 AMI 또는 인스턴스 블록 디바이스 매핑에 지정되지 않은 경우에도 모든 NVMe 인스턴스 스토어 볼륨을 사용할 수 있으므로 업데이트된 인스턴스에 추가 인스턴스 스토어 볼륨이 있을 수 있습니다. 그렇지 않으면 업데이트한 인스턴스는 원본 인스턴스를 시작할 때 지정한 것과 동일한 수의 인스턴스 스토어 볼륨을 갖습니다.
- 인스턴스에 연결할 수 있는 Amazon EBS 볼륨의 최대 수는 인스턴스 유형 및 인스턴스 크기에 따라 달라집니다. 인스턴스에 이미 연결된 볼륨 수를 지원하지 않는 인스턴스 유형이나 인스턴스 크기로 변경할 수 없습니다. 자세한 내용은 Amazon EC2 인스턴스의 Amazon EBS 볼륨 한도 단원을 참조하십시오.
- [Linux 인스턴스] `AWSSupport-MigrateXenToNitroLinux` 런북을 사용하여 호환되는 Linux 인스턴스를 Xen 인스턴스 유형에서 Nitro 인스턴스 유형으로 마이그레이션할 수 있습니다. 자세한 내용은 AWS Systems Manager Automation 실행서 참조에서 AWSSupport-MigrateXenToNitroLinux runbook를 참조하세요.
- [Windows 인스턴스] 호환되는 Windows 인스턴스를 Xen 인스턴스 유형에서 Nitro 인스턴스 유형으로 마이그레이션하는 방법에 대한 추가 지침은 최신 세대 인스턴스 유형으로 마이그레이션을 참조하세요.

Amazon EBS 지원 인스턴스의 인스턴스 유형 변경

1. (선택 사항) 새로운 인스턴스 유형이 기존 인스턴스에 드라이버가 설치되어 있어야 하는 유형인 경우, 먼저 인스턴스에 연결하여 해당 드라이버를 설치해야 합니다. 자세한 내용은 인스턴스 유형 변경을 위한 호환성 단원을 참조하십시오.
2. [Windows 인스턴스] 고정 IP 주소 지정을 사용하도록 Windows 인스턴스를 구성했고 향상된 네트워킹을 지원하지 않는 인스턴스 유형에서 향상된 네트워킹을 지원하는 인스턴스 유형으로 변경하는 경우에는 고정 IP 주소 지정을 다시 구성하면 잠재적인 IP 주소 충돌에 대한 경고를 받을 수 있습니다. 이를 방지하려면 인스턴스 유형을 변경하기 전에 해당 인스턴스의 네트워

크 인터페이스에서 DHCP를 활성화합니다. 인스턴스에서 네트워크 및 공유 센터(Network and Sharing Center)를 열고, 네트워크 인터페이스의 인터넷 프로토콜 버전 4(TCP/IPv4) 속성(Internet Protocol Version 4 (TCP/IPv4) Properties)을 열고, 자동으로 IP 주소 획득(Obtain an IP address automatically)을 선택합니다. 인스턴스 유형을 변경하고 네트워크 인터페이스에서 고정 IP 주소 지정을 다시 구성합니다.

3. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
4. 탐색 창에서 Instances(인스턴스)를 선택합니다.
5. 인스턴스를 선택하고 인스턴스 상태, 인스턴스 중지를 차례로 선택합니다. 확인 메시지가 표시되면 [중지(Stop)]를 선택합니다. 인스턴스가 중지하는 데 몇 분 정도 걸릴 수 있습니다.
6. 인스턴스를 선택한 상태에서 작업, 인스턴스 설정, 인스턴스 유형 변경을 차례로 선택합니다. 인스턴스가 `stopped` 상태가 아닌 경우에는 이 옵션이 회색으로 표시됩니다.
7. 인스턴스 유형 변경(Change instance type) 페이지에서 다음을 수행합니다.

   a. 인스턴스 유형(Instance type)에서 원하는 인스턴스 유형을 선택합니다.

      인스턴스 유형이 목록에 없으면 인스턴스 구성과 호환되지 않는 것입니다. 대신 새 EC2 인스턴스를 시작하여 새 인스턴스 유형으로 마이그레이션 지침을 따릅니다.

   b. (선택 사항) 선택한 인스턴스 유형이 EBS 최적화를 지원하는 경우 EBS 최적화(EBS-optimized)를 선택하여 EBS 최적화를 사용하거나 EBS 최적화(EBS-optimized)를 선택 취소하여 EBS 최적화를 비활성화합니다.

      선택한 인스턴스 유형이 기본적으로 EBS에 최적화되었을 경우 EBS 최적화가 선택되고 이를 선택 취소할 수 없습니다.

   c. (선택 사항) 새 인스턴스 유형에서 vCPU 옵션을 구성합니다.

      기존 인스턴스의 인스턴스 유형을 변경하면 Amazon EC2는 가능한 경우 기존 인스턴스의 CPU 옵션 설정을 새 인스턴스에 적용합니다. 새 인스턴스 유형이 이 설정을 지원하지 않으면 CPU 옵션이 없음으로 재설정됩니다. 이 옵션은 새 인스턴스 유형에 대한 기본 vCPU 수를 사용합니다.

      선택한 인스턴스 유형이 vCPU 구성을 지원하는 경우 고급 세부 정보 패널에서 CPU 옵션 지정을 선택하여 새 인스턴스 유형에 맞게 vCPU를 구성합니다.

   d. 변경을 선택하여 새로운 설정을 승인합니다.

8. 중지된 인스턴스를 다시 시작하려면 인스턴스를 선택하고 인스턴스 상태(Instance state), 인스턴스 시작(Start instance)을 선택합니다. 인스턴스가 `running` 상태가 되는 데 몇 분 정도 걸릴 수 있습니다. 인스턴스가 시작되지 않으면 인스턴스 유형 변경 문제 해결 섹션을 참조하세요.

9. [Windows 인스턴스] 인스턴스가 Windows Server 2016 또는 Windows Server 2019에서 EC2Launch v1으로 실행하는 경우 Windows 인스턴스에 연결하고 다음 EC2Launch PowerShell 스크립트를 실행하여 인스턴스 유형을 변경한 후에 인스턴스를 구성합니다.

> **Important**
>
> 초기화 인스턴스 EC2 Launch 스크립트를 활성화하면 관리자 암호가 재설정됩니다. 관리자 암호 재설정은 구성 파일을 수정하여 비활성화할 수 있는데 초기화 작업에 대한 설정에서 이를 지정하면 됩니다. 암호 재설정을 사용하지 않도록 설정하는 방법에 대한 단계는 초기화 작업 구성(EC2Launch) 또는 설정 변경(EC2Launch v2)을 참조하세요.

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -
Schedule
```

새 EC2 인스턴스를 시작하여 새 인스턴스 유형으로 마이그레이션

원하는 새 인스턴스 유형과 호환되는 구성을 가진 EBS 지원 인스턴스인 경우에만 EC2 인스턴스의 인스턴스 유형을 변경할 수 있습니다. 그렇지 않으면 구성 또는 인스턴스가 새 인스턴스 유형과 호환되지 않거나 인스턴스 스토어 기반 인스턴스인 경우 원하는 인스턴스 유형과 호환되는 대체 인스턴스를 시작해야 합니다. 호환성 결정 방법에 대한 자세한 내용은 인스턴스 유형 변경을 위한 호환성 섹션을 참조하세요.

마이그레이션 프로세스 개요

- 원본 인스턴스의 데이터를 백업합니다.
- 원하는 새 인스턴스 유형과 호환되는 구성으로 새 인스턴스를 시작하고 원본 인스턴스에 연결된 EBS 볼륨을 연결합니다.
- 새 인스턴스에 애플리케이션을 설치합니다.
- 데이터를 복원합니다.
- 원본 인스턴스가 탄력적 IP 주소가 갖고 있는 경우 해당 주소를 새 인스턴스와 연결하여 사용자가 중단 없이 애플리케이션을 계속 사용할 수 있도록 해야 합니다.

새 인스턴스로 마이그레이션하려면 다음을 수행하세요.

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

2. 여전히 필요한 데이터를 다음과 같이 백업합니다.

   - 인스턴스에 연결하고 인스턴스 스토어 볼륨의 데이터를 영구 스토리지에 복사합니다.
   - EBS 볼륨의 스냅샷을 생성하여 동일한 데이터로 새 볼륨을 생성하거나 원본 인스턴스에서 볼륨을 분리하여 새 인스턴스에 연결할 수 있습니다.

3. 탐색 창에서 Instances(인스턴스)를 선택합니다.
4. 인스턴스 시작(Launch Instances)을 선택합니다. 인스턴스를 구성할 때 다음을 수행합니다.

   a. 원하는 인스턴스 유형을 지원하는 AMI를 선택합니다. 예를 들어 새 인스턴스 유형의 프로세서 유형을 지원하는 AMI를 선택해야 합니다. 또한 현재 세대 인스턴스 유형에 HVM AMI가 필요합니다.

   b. 원하는 새 인스턴스 유형을 선택합니다. 원하는 인스턴스 유형을 사용할 수 없으면 선택한 AMI의 구성과 호환되지 않는 것입니다.

   c. 동일한 트래픽이 새 인스턴스에 도달하도록 허용하려면 원본 인스턴스에서 사용된 것과 동일한 VPC와 보안 그룹을 선택합니다.

   d. 새 인스턴스 구성을 마치면 키 페어를 선택하고 인스턴스를 시작하는 단계를 수행합니다. 인스턴스가 `running` 상태가 되는 데 몇 분 정도 걸릴 수 있습니다.

5. EBS 스냅샷에 데이터를 백업한 경우 스냅샷에서 볼륨을 생성한 다음 새 인스턴스에 볼륨을 연결합니다.

   원본 인스턴스에서 새 인스턴스로 EBS 볼륨을 이동하려면 원본 인스턴스에서 볼륨을 분리한 다음 새 인스턴스에 볼륨을 연결합니다.

6. 애플리케이션과 기타 필요한 소프트웨어를 새 인스턴스에 설치합니다.
7. 원래 인스턴스의 인스턴스 스토어 볼륨에서 백업한 데이터를 복원합니다.
8. 원본 인스턴스가 탄력적 IP 주소를 가지고 있는 경우 다음과 같이 이를 새 인스턴스에 할당합니다.

   a. 탐색 창에서 Elastic IPs를 선택합니다.

   b. 원래 인스턴스와 연결된 탄력적 IP 주소를 선택하고 작업, 탄력적 IP 주소 연결 해제를 차례로 선택합니다. 확인 메시지가 나타나면 연결 해제를 선택합니다.

   c. 탄력적 IP 주소를 선택한 상태에서 작업, 탄력적 IP 주소 연결을 차례로 선택합니다.

   d. 리소스 유형에서 인스턴스를 선택합니다.

   e. 인스턴스에서 새 인스턴스를 선택합니다.

   f. (선택 사항) 프라이빗 IP 주소에 탄력적 IP 주소를 연결할 프라이빗 IP 주소를 지정합니다.

   g. 연결(Associate)을 선택합니다.

9. (선택 사항) 원래 인스턴스가 더 이상 필요하지 않을 경우 이를 종료할 수 있습니다. 인스턴스를 선택하고 새 인스턴스가 아닌 원본 인스턴스를 종료하고 있는지 확인한 다음(예를 들어 이름이나 시작 시간) 인스턴스 상태(Instance state), 인스턴스 종료(Terminate instance)를 선택하세요.

인스턴스 유형 변경 문제 해결

다음 정보를 사용하여 인스턴스 유형을 변경할 때 발생할 수 있는 문제를 진단하고 수정합니다.

인스턴스 유형 변경 후 인스턴스가 시작되지 않음

가능한 원인: 새 인스턴스 유형에 대한 요구 사항이 충족되지 않음

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 참조하세요.

가능한 원인: AMI가 인스턴스 유형을 지원하지 않음

EC2 콘솔을 사용하여 인스턴스 유형을 변경하는 경우 선택한 AMI에서 지원하는 인스턴스 유형만 사용할 수 있습니다. 그러나 AWS CLI를 사용하여 인스턴스를 시작하면 호환되지 않는 AMI와 인스턴스 유형을 지정할 수 있습니다. AMI와 인스턴스 유형이 호환되지 않으면 인스턴스를 시작할 수 없습니다. 자세한 내용은 인스턴스 유형 변경을 위한 호환성 단원을 참조하십시오.

가능한 원인: 인스턴스가 클러스터 배치 그룹에 있음

인스턴스가 클러스터 배치 그룹에 있으며 인스턴스 유형을 변경한 후 인스턴스가 시작되지 않으면 다음을 시도하세요.

1. 클러스터 배치 그룹의 모든 인스턴스를 중지합니다.
2. 영향을 받는 인스턴스의 인스턴스 유형을 변경합니다.
3. 클러스터 배치 그룹의 모든 인스턴스를 시작합니다.

인스턴스 유형 변경 후 인터넷에서 연결할 수 없는 애플리케이션 또는 웹 사이트

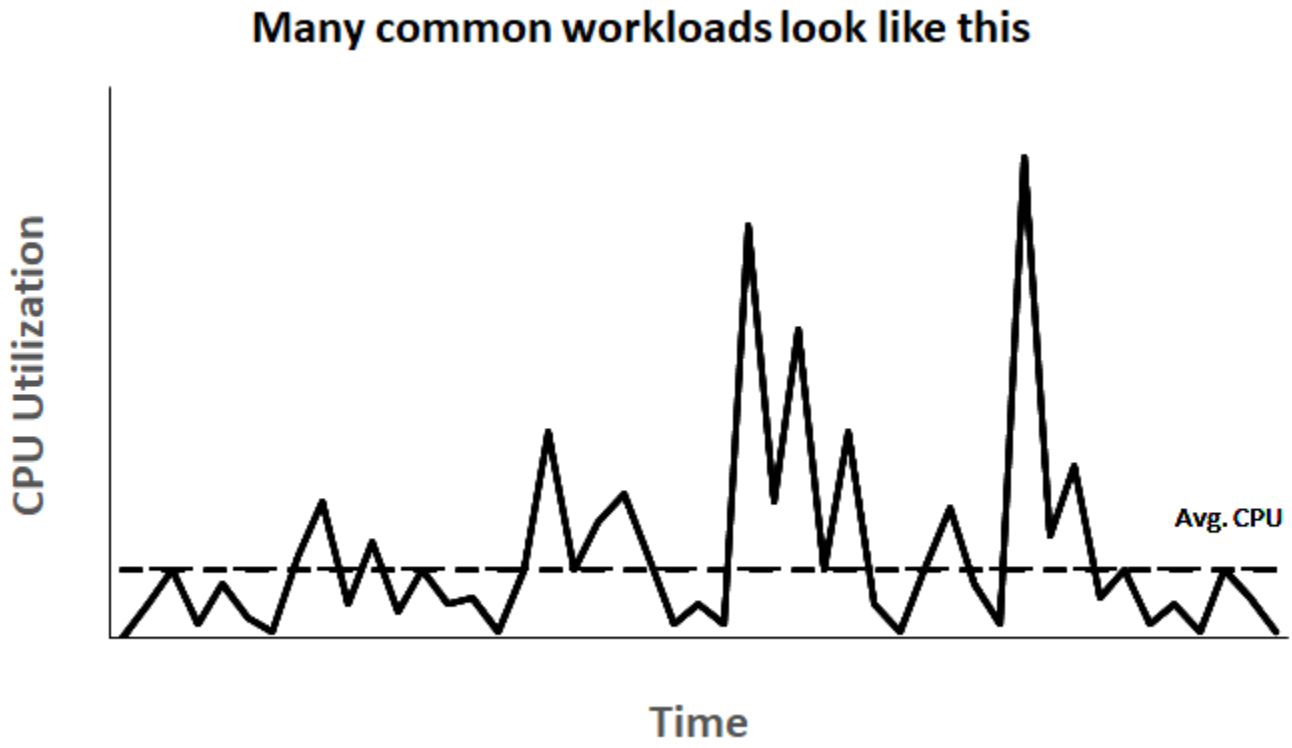
가능한 원인: 퍼블릭 IPv4 주소가 릴리스됨

인스턴스 유형을 변경할 때 먼저 인스턴스를 중지해야 합니다. 인스턴스를 중지할 때 인스턴스에 퍼블릭 IPv4 주소를 해제하고 인스턴스에 새 퍼블릭 IPv4 주소를 제공합니다.

인스턴스 중지 및 시작 사이에 퍼블릭 IPv4 주소를 유지하려면 인스턴스가 실행 중일 때 추가 비용 없이 탄력적 IP 주소를 사용하는 것이 좋습니다. 자세한 내용은 탄력적 IP 주소 단원을 참조하십시오.

성능 순간 확장 가능 인스턴스

대부분의 범용 워크로드는 평균적으로 사용량이 많지 않으며 높은 수준의 지속적인 CPU 성능을 요구하지 않습니다. 다음 그래프는 고객이 AWS 클라우드에서 실행하는 다양한 일반적인 워크로드의 CPU 사용률을 보여 줍니다.



CPU 사용률이 낮거나 중간 정도인 이러한 워크로드는 CPU 사이클의 낭비를 초래하고 결과적으로 사용한 것보다 더 많은 비용을 지불하게 됩니다. 이를 극복하기 위해 저비용 버스트 가능 범용 인스턴스인 T 인스턴스를 활용할 수 있습니다.

T 인스턴스 패밀리는 기준 CPU 성능을 제공하며, 필요에 따라 언제든지 기준 이상으로 버스트할 수 있습니다. 기준 CPU는 대규모 마이크로 서비스, 웹 서버, 중소 규모의 데이터베이스, 데이터 로깅, 코드 리포지토리, 가상 데스크톱, 개발 및 테스트 환경, 비즈니스 크리티컬 애플리케이션을 비롯한 대부분의 범용 워크로드의 요구 사항을 충족하도록 정의되었습니다. T 인스턴스는 컴퓨팅, 메모리 및 네트워크 리소스를 균형있게 제공하며 CPU 사용량이 낮거나 중간 정도인 광범위한 범용 애플리케이션을 실행할 수 있는 가장 비용 효율적인 솔루션을 제공합니다. M 인스턴스에 비해 최대 15%의 비용을 절감할 수 있으며, 더 작고 경제적인 인스턴스 크기 덕분에 비용을 추가로 절감할 수 있으며, vCPU 2개와 0.5GiB의 메모리를 제공합니다. 나노, 마이크로, 소형, 중형 등의 작은 T 인스턴스 크기는 적은 양의 메모리가 필요하고 높은 CPU 사용량이 요구되지 않는 워크로드에 적합합니다.

> **Note**
>
> 이 항목에서는 버스트 가능한 CPU에 대해 설명합니다. 버스트 가능 네트워크 성능에 대한 자세한 내용은 Amazon EC2 인스턴스 네트워크 대역폭 섹션을 참조하세요.

EC2 버스트 가능 인스턴스 유형

EC2 버스트 가능 인스턴스는 T4g, T3a, T3 인스턴스 유형과 이전 세대 T2 인스턴스 유형으로 구성됩니다.

T4g 인스턴스 유형은 최신 세대의 버스트 가능 인스턴스입니다. 최적의 가격 대비 성능을 제공하며 모든 EC2 인스턴스 유형 중에서 가장 비용이 낮습니다. T4g 인스턴스 유형은 ARM 기반 AWS Graviton2 프로세서를 기반으로 하며, 운영 체제 공급업체, 독립 소프트웨어 공급업체, 유명 AWS 서비스 및 애플리케이션의 광범위한 에코시스템 지원을 제공합니다.

다음 표에는 버스트 가능 인스턴스 유형 간의 주요 차이점이 요약되어 있습니다.

Type	설명	프로세서 패밀리
최신 세대		
T4g	T3에 비해 최대 40% 더 높은 가격 대비 성능과 20% 저렴한 비용을 제공하는 최저가 EC2 인스턴스 유형	Arm Neoverse N1 코어가 장착된 AWS Graviton2 프로세서
T3a	T3 인스턴스에 비해 10% 저렴한 비용의 최저가 x86 기반 인스턴스	AMD 1세대 EPYC 프로세서
T3	이전 세대 T2 인스턴스에 비해 최대 30% 낮은 가격 대비 성능으로 x86 워크로드를 위한 최고의 가격 대비 최대 성능 제공	인텔 제온 스케일러블(Skylake, Cascade Lake 프로세서)
이전 세대		

Type	설명	프로세서 패밀리
T2	이전 세대 버스트 가능 인스턴스	인텔 제온 프로세서

인스턴스 요금에 대한 자세한 내용과 기타 사양은 Amazon EC2 요금 및 Amazon EC2 인스턴스 유형을 참조하세요. 버스트 가능 네트워크 성능에 대한 자세한 내용은 Amazon EC2 인스턴스 네트워크 대역폭 섹션을 참조하세요.

계정이 12개월이 아직 안 된 경우 특정 사용 한도 내에서 무료로 `t2.micro` 인스턴스(또는 `t3.micro`를 사용할 수 없는 리전에서는 `t2.micro` 인스턴스)를 사용할 수 있습니다. 자세한 내용은 AWS 프리 티어 섹션을 참조하세요.

T 인스턴스에 대해 지원되는 구매 옵션

- On-Demand Instances
- Reserved Instances
- 전용 인스턴스(T3에만 해당)
- 전용 호스트(T3 전용, `standard` 모드만 해당)
- 스팟 인스턴스

자세한 정보는 Amazon EC2 결제 및 구매 옵션을 참조하세요.

목차

- 모범 사례
- 성능 버스트 가능 인스턴스에 대한 주요 개념 및 정의
- 성능 순간 확장 가능 인스턴스의 무제한 모드
- 성능 순간 확장 가능 인스턴스의 스탠다드 모드
- 성능 버스트 가능 인스턴스 구성
- 성능 버스트 가능 인스턴스에 대한 CPU 크레딧 모니터링

모범 사례

다음 모범 사례를 따르면 성능 순간 확장 가능 인스턴스의 이점을 최대한 활용할 수 있습니다.

- 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 `t3.micro` 이상이어야 합니다. 시간이 지나면서 워크로드의 메모리 및 CPU 요구 사항이 증가함에 따라 유연하게 T 인스턴스를 더 큰 규모의 동일한 인스턴스 유형으로 확장하거나 다른 인스턴스 유형을 선택할 수 있습니다.
- 계정에 대해 AWS Compute Optimizer를 활성화하고 워크로드에 대한 Compute Optimizer 권장 사항을 검토하십시오. Compute Optimizer는 성능을 향상시키기 위해 인스턴스를 확장해야 하는지 또는 비용을 절감하기 위해 축소해야 하는지 여부를 평가하는 데 도움이 됩니다. Compute Optimizer는 시나리오에 따라 다른 인스턴스 유형을 권장할 수도 있습니다. 자세한 내용은 AWS Compute Optimizer 사용 설명서의 EC2 인스턴스 권장 사항 보기를 참조하세요.

성능 버스트 가능 인스턴스에 대한 주요 개념 및 정의

기존 Amazon EC2 인스턴스 유형은 고정된 CPU 리소스를 제공하는 반면, 성능 순간 확장 가능 인스턴스는 기본 수준의 CPU 사용률을 제공하면서 기본 수준 이상으로 CPU 사용률을 버스트하는 기능을 제공합니다. 이렇게 하면 기준 CPU와 추가 버스트 CPU 사용량에 대해서만 비용을 지불하면 되므로 컴퓨팅 비용이 절감됩니다. 기준 사용률과 버스트 기능은 CPU 크레딧에 의해 좌우됩니다. 성능 순간 확장 가능 인스턴스는 CPU 사용량에 대해 크레딧을 사용하는 유일한 인스턴스 유형입니다.

각 버스트 가능 성능 인스턴스는 CPU 기준 미만으로 유지되면 지속적으로 크레딧을 얻고, 기준선 이상으로 버스트될 때 크레딧을 지속적으로 소비합니다. 적립되거나 소비되는 크레딧 금액은 인스턴스의 CPU 사용률에 따라 달라집니다.

- CPU 사용률이 기준 미만인 경우 적립되는 크레딧은 소비되는 크레딧보다 많습니다.
- CPU 사용률이 기준과 같을 경우 적립되는 크레딧은 소비되는 크레딧과 같습니다.
- CPU 사용률이 기준을 초과할 경우 소비되는 크레딧이 적립되는 크레딧보다 많습니다.

적립되는 크레딧이 소비되는 크레딧보다 많을 경우의 차액을 획득한 크레딧이라고 하며, 이를 나중에 기준 CPU 사용률 이상으로 버스트하는 데 사용할 수 있습니다. 마찬가지로, 소비되는 크레딧이 적립되는 크레딧보다 많을 경우 인스턴스 동작은 크레딧 구성 모드(표준 모드 또는 무제한 모드)에 따라 달라집니다.

표준 모드에서 소비되는 크레딧이 적립되는 크레딧보다 많을 경우 인스턴스는 획득한 크레딧을 사용하여 기준 CPU 사용률을 초과하여 버스트합니다. 획득한 크레딧이 남아 있지 않으면 인스턴스가 기준 CPU 사용률로 점진적으로 저하되고 크레딧이 더 많이 적립될 때까지 기준 이상으로 버스트할 수 없습니다.

무제한 모드에서는 인스턴스가 기준 CPU 사용률 이상으로 버스트하면 인스턴스는 먼저 획득한 크레딧을 사용하여 버스트합니다. 획득한 크레딧이 남아 있지 않으면 인스턴스는 버스트에 잉여 크레딧을 사용합니다. 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.

내용

- 핵심 개념 및 정의
- CPU 크레딧 적립
- CPU 크레딧 획득률
- CPU 크레딧 누적 한도
- 획득한 CPU 크레딧의 수명
- 기준 사용률

핵심 개념 및 정의

다음 주요 개념 및 정의는 버스트 가능 성능 인스턴스에 적용할 수 있습니다.

CPU 사용률

CPU 사용률은 인스턴스에서 현재 사용 중인 할당된 EC2 컴퓨팅 유닛의 비율(%)입니다. 이 지표는 인스턴스에서 사용되고 있는 할당된 CPU 사이클의 비율을 측정합니다. CPU 사용률 CloudWatch 지표는 코어당 CPU 사용량이 아니라 인스턴스당 CPU 사용량을 나타냅니다. 인스턴스의 기준 CPU 사양도 인스턴스당 CPU 사용량을 기준으로 합니다. AWS Management Console 또는 AWS CLI를 사용하여 CPU 사용률을 측정하려면 특정 인스턴스에 대한 통계 가져오기 섹션을 참조하세요.

CPU 크레딧

vCPU 시간의 단위입니다.

예:

CPU 크레딧 1개 = vCPU 1개 * 100% 사용률 * 1분

CPU 크레딧 1개 = vCPU 1개 * 50% 사용률 * 2분

CPU 크레딧 1개 = vCPU 2개 * 25% 사용률 * 2분

기준 사용률

기준 사용률은 획득하는 CPU 크레딧 수가 사용 중인 CPU 크레딧 수와 일치할 때 순 크레딧 밸런스 0에서 CPU를 사용할 수 있는 수준입니다. 기준 사용률을 기준이라고도 합니다. 기준 사용률은 vCPU 사용률의 백분율로 표시되며 기준 사용률(%) = (획득한 크레딧 수/vCPU 수)/60분으로 계산됩니다.

각 버스트 가능한 성능 인스턴스 유형의 기준 사용률은 크레딧 표를 참조하세요.

획득 크레딧

인스턴스가 실행 중일 때 지속적으로 적립되는 크레딧입니다.

시간당 적립되는 크레딧 수 = 기준 사용률(%) * vCPU 수 * 60분

예:

vCPU가 2개인 t3.nano는 기본 사용률이 5%로, 시간당 6 크레딧을 획득하며, 다음과 같이 계산됩니다.

vCPU 2개 * 5% 기준 * 60분 = 시간당 6 크레딧

소비 또는 사용되는 크레딧

인스턴스가 실행 중일 때 지속적으로 소비되는 크레딧입니다.

분당 소비되는 CPU 크레딧 = vCPU 수 * CPU 사용률 * 1분

획득한 크레딧

인스턴스가 기준 사용률에 필요한 것보다 적은 크레딧을 사용하는 경우 사용되지 않은 CPU 크레딧입니다. 즉, 획득한 크레딧 = (적립되는 크레딧 - 기준 미만으로 사용되는 크레딧)입니다.

예:

t3.nano가 한 시간 동안 5% 기준보다 적은 2% CPU 사용률로 실행될 경우 획득한 크레딧은 다음과 같이 계산됩니다.

획득한 CPU 크레딧 = (시간당 적립되는 크레딧 - 시간당 소비되는 크레딧) = 6 - 2 vCPU * 2% CPU 사용률 * 60분 = 6 - 2.4 = 시간당 획득한 크레딧 3.6입니다.

크레딧 누적 한도

인스턴스 크기에 따라 다르지만 일반적으로 24시간 동안 적립되는 최대 크레딧 수와 같습니다.

예:

t3.nano의 경우 크레딧 누적 한도 = 24 * 6 = 144 크레딧입니다.

시작 크레딧

표준 모드로 구성된 T2 인스턴스에만 적용됩니다. 시작 크레딧은 새 T2 인스턴스에 할당되는 제한된 수의 CPU 크레딧으로, 표준 모드로 시작할 때 기준 이상으로 버스트할 수 있습니다.

잉여 크레딧

획득한 크레딧 잔액이 소진된 후 인스턴스가 소비하는 크레딧입니다. 잉여 크레딧은 버스트 가능 인스턴스가 오랜 기간 동안 고성능을 유지할 수 있도록 고안되었으며 무제한 모드에서만 사용됩니다. 잉여 크레딧 잔액은 인스턴스가 무제한 모드에서 버스트하기 위해 소비한 크레딧 수를 확인하는 데 사용됩니다.

스탠다드 모드

크레딧 구성 모드로, 크레딧 잔액에 적립된 크레딧을 사용하여 인스턴스를 기준 이상으로 버스트할 수 있습니다.

무제한 모드

크레딧 구성 모드로, 필요할 때마다 원하는 기간 동안 높은 CPU 사용률을 유지하여 인스턴스를 기준 이상으로 버스트할 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. 인스턴스 실행에 장기간 높은 CPU 사용률이 필요한 경우, vCPU-시간당 추가 고정 요금으로 인스턴스를 실행할 수 있습니다.

다음 표에는 버스트 가능 인스턴스 유형 간의 주요 크레딧 차이점이 요약되어 있습니다.

Type	지원되는 CPU 크레딧 유형	크레딧 구성 모드	인스턴스 시작과 중지 사이에 획득하는 CPU 크레딧 수명
최신 세대			
T4g	적립 크레딧, 획득 크레딧, 소비 크레딧, 잉여 크레딧(무제한 모드만 해당)	표준, 무제한(기본값)	7일(크레딧이 인스턴스 중지 후 7일 동안 유지됨)

Type	지원되는 CPU 크레딧 유형	크레딧 구성 모드	인스턴스 시작과 중지 사이에 획득하는 CPU 크레딧 수명
T3a	적립 크레딧, 획득 크레딧, 소비 크레딧, 잉여 크레딧(무제한 모드만 해당)	표준, 무제한(기본값)	7일(크레딧이 인스턴스 중지 후 7일 동안 유지됨)
T3	적립 크레딧, 획득 크레딧, 소비 크레딧, 잉여 크레딧(무제한 모드만 해당)	표준, 무제한(기본값)	7일(크레딧이 인스턴스 중지 후 7일 동안 유지됨)
이전 세대			
T2	적립 크레딧, 획득 크레딧, 소비 크레딧, 시작 크레딧(표준 모드만 해당), 잉여 크레딧(무제한 모드만 해당)	표준(기본값), 무제한	0일(인스턴스가 중지되면 크레딧이 소실됨)

> **Note**
>
> 무제한 모드에서는 전용 호스트에서 시작되는 T3 인스턴스에 대해 지원하지 않습니다.

CPU 크레딧 적립

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 기본 사용률에 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 사용률 수준 이상으로 버스트해야 할 경우 누적된 크레딧을 소모합니다. 성능 순간 확

장 가능 인스턴스가 획득한 크레딧이 많을수록 추가 CPU 사용률이 필요할 때 기준 사용률 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 코어의 백분율로 나타낸 기준 사용률(단일 vCPU 사용 시) 등이 나와 있습니다.

인스턴스 유형	시간당 지급되는 CPU 크레딧	누적 가능한 최대 지급된 크레딧*	vCPU***	vCPU당 기준 사용률
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%**
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%**

인스턴스 유형	시간당 지급되는 CPU 크레딧	누적 가능한 최대 지급된 크레딧*	vCPU***	vCPU당 기준 사용률
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%**
T4g				
t4g.nano	6	144	2	5%**
t4g.micro	12	288	2	10%**
t4g.small	24	576	2	20%**
t4g.medium	24	576	2	20%**
t4g.large	36	864	2	30%**
t4g.xlarge	96	2304	4	40%**
t4g.2xlarge	192	4608	8	40%**

* 누적될 수 있는 크레딧은 수는 24시간 동안 획득할 수 있는 크레딧의 수와 동일합니다.

** 테이블의 기준 사용률(%)은 vCPU당입니다. CloudWatch에서 CPU 사용률은 vCPU 기준으로 표시됩니다. 예를 들어 기준 수준으로 작동하는 `t3.large` 인스턴스에 대한 CPU 사용률은 CloudWatch CPU 지표의 30%로 표시됩니다. 기준 사용률을 계산하는 방법에 대한 자세한 내용은 기준 사용률 섹션을 참조하세요.

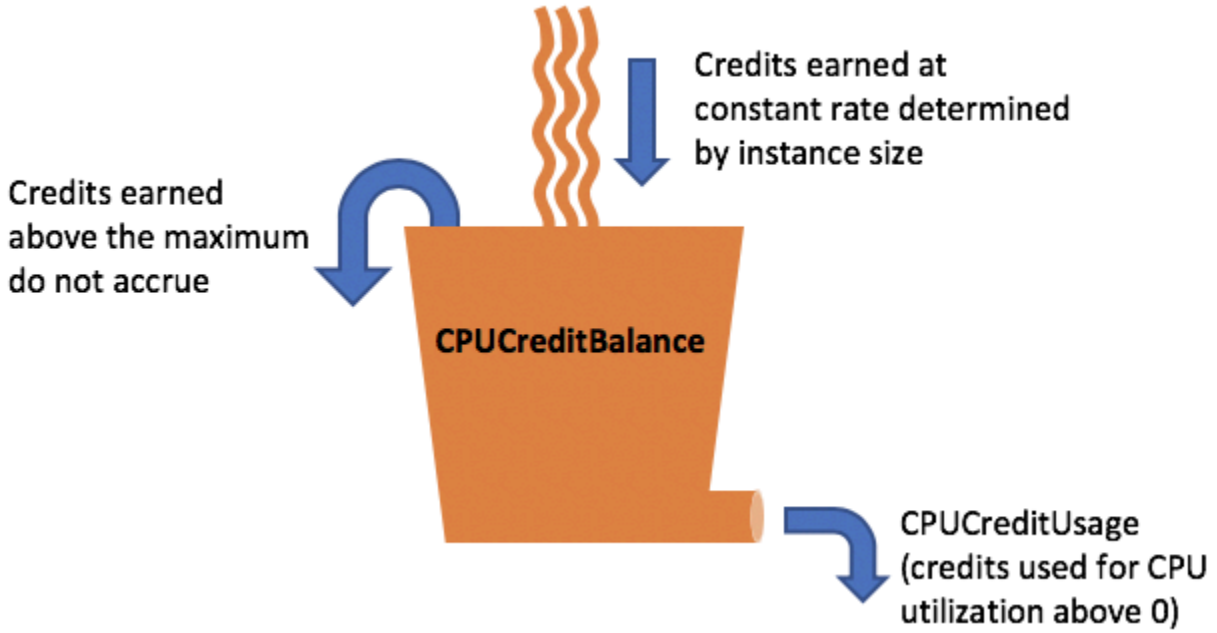
*** T2 및 T4g 인스턴스를 제외하고, 각 vCPU는 인텔 제온 코어 또는 AMD EPYC 코어의 스레드입니다.

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.

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 섹션을 참조하세요.

T4g, T3a, T3 인스턴스에서는 시작 크레딧을 획득하지 않습니다. 이러한 인스턴스는 unlimited로 시작하도록 기본 설정되어 있으므로 시작 크레딧 없이도 시작하자마자 즉시 버스트할 수 있습니다. 전용 호스트에서 시작되는 T3 인스턴스는 기본적으로 standard로 시작되며, unlimited 모드에서는 전용 호스트의 T3 인스턴스에 대해 지원하지 않습니다.

획득한 CPU 크레딧의 수명

실행 중인 인스턴스의 CPU 크레딧은 만료 기간이 없습니다.

T2의 경우 CPU 크레딧 밸런스는 인스턴스 중지와 시작 사이의 기간 동안 지속하지 않습니다. T2 인스턴스를 중지하면 인스턴스는 누적된 크레딧을 모두 상실합니다.

T4g, T3a, T3의 경우 인스턴스가 중지된 후 CPU 크레딧 밸런스가 7일 동안 지속되며 이후 크레딧이 소멸됩니다. 7일 이내에 인스턴스를 시작하면 크레딧이 상실되지 않습니다.

자세한 내용은 CloudWatch 지표 표에서 CPUCreditBalance 항목을 참조하세요.

기준 사용률

기준 사용률은 획득하는 CPU 크레딧 수가 사용 중인 CPU 크레딧 수와 일치할 때 순 크레딧 밸런스 0에서 CPU를 사용할 수 있는 수준입니다. 기준 사용률을 기준이라고도 합니다.

기준 사용률은 vCPU 사용률의 백분율로 표시되며 다음과 같이 계산됩니다.

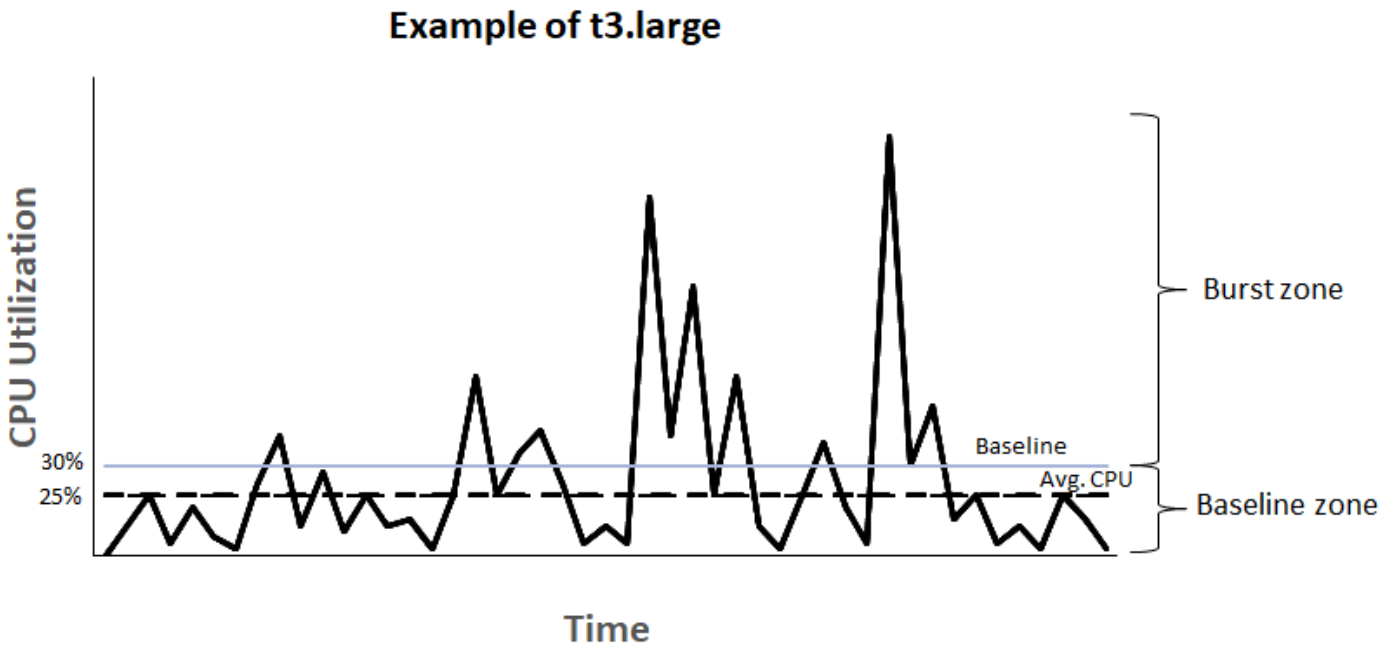
```
(number of credits earned/number of vCPUs)/60 minutes = % baseline
utilization
```

예를 들어 vCPU가 2개인 t3.nano 인스턴스는 시간당 6 크레딧을 획득하여 기준 사용률이 5% 로, 다음과 같이 계산됩니다.

```
(6 credits earned/2 vCPUs)/60 minutes = 5% baseline utilization
```

vCPU가 2개인 t3.large 인스턴스는 시간당 36 크레딧을 획득하여 기준 사용률이 30%입니다 ((36/2)/60).

다음 그래프는 평균 CPU 사용률이 기준보다 낮은 t3.large의 예를 보여줍니다.



성능 순간 확장 가능 인스턴스의 무제한 모드

unlimited로 구성된 성능 순간 확장 가능 인스턴스는 필요한 경우 언제든지 원하는 기간 동안 높은 CPU 사용률을 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 unlimited로 구성된 인스턴스는 추가 요금 없이 충분한 성능을 제공합니다. 인스턴스 실행에 장기간 높은 CPU 사용률이 필요한 경우, vCPU-시간당 추가 고정 요금으로 인스턴스를 실행할 수 있습니다. 요금에 대한 자세한 내용은 Amazon EC2 요금 및 T2/T3/T4 무제한 모드 요금을 참조하세요.

AWS 프리 티어 혜택에 따라 t2.micro 또는 t3.micro 인스턴스를 사용하고 이 인스턴스를 unlimited 모드에서 사용하는 경우 24시간 롤링 기간 동안 평균 사용률이 인스턴스의 기준 사용률을 초과하면 요금이 적용될 수 있습니다.

T4g, T3a, T3 인스턴스는 기본값을 변경하지 않는 한 기본적으로 unlimited로 시작됩니다. 24시간 동안 평균 CPU 사용량이 기준을 초과하면 잉여 크레딧에 대한 요금이 발생합니다. 스팟 인스턴스를 unlimited으로 시작하고 CPU 크레딧 누적에 대한 유휴 시간 없이 즉시 짧은 기간 동안 사용하려는

경우, 추가 크레딧에 대한 요금이 발생합니다. 더 높은 비용을 지불하지 않으려면 표준 모드에서 스팟 인스턴스를 시작하는 것이 좋습니다. 자세한 내용은 잉여 크레딧으로 요금 발생 가능 및 성능 버스트 가능 인스턴스 시작 섹션을 참조하세요.

> **Note**
>
> 전용 호스트에서 시작되는 T3 인스턴스는 기본적으로 `standard`로 시작되며, `unlimited` 모드에서는 전용 호스트의 T3 인스턴스에 대해 지원하지 않습니다.

내용

- 성능 버스트 가능 인스턴스의 무제한 모드 개념
  - 무제한 성능 순간 확장 가능 인스턴스의 작동 방식
  - 무제한 모드 대 고정 CPU 사용 시기
  - 잉여 크레딧으로 요금 발생 가능
  - T2 무제한 인스턴스에는 시작 크레딧이 없음
  - 무제한 모드 활성화
  - 무제한과 스탠다드 간 전환 시 크레딧에 발생하는 현상
  - 크레딧 사용량 모니터링
- 성능 버스트 가능 인스턴스의 무제한 모드 예제
  - 예 1: T3 무제한의 크레딧 사용 설명
  - 예 2: T2 무제한의 크레딧 사용 설명

성능 버스트 가능 인스턴스의 무제한 모드 개념

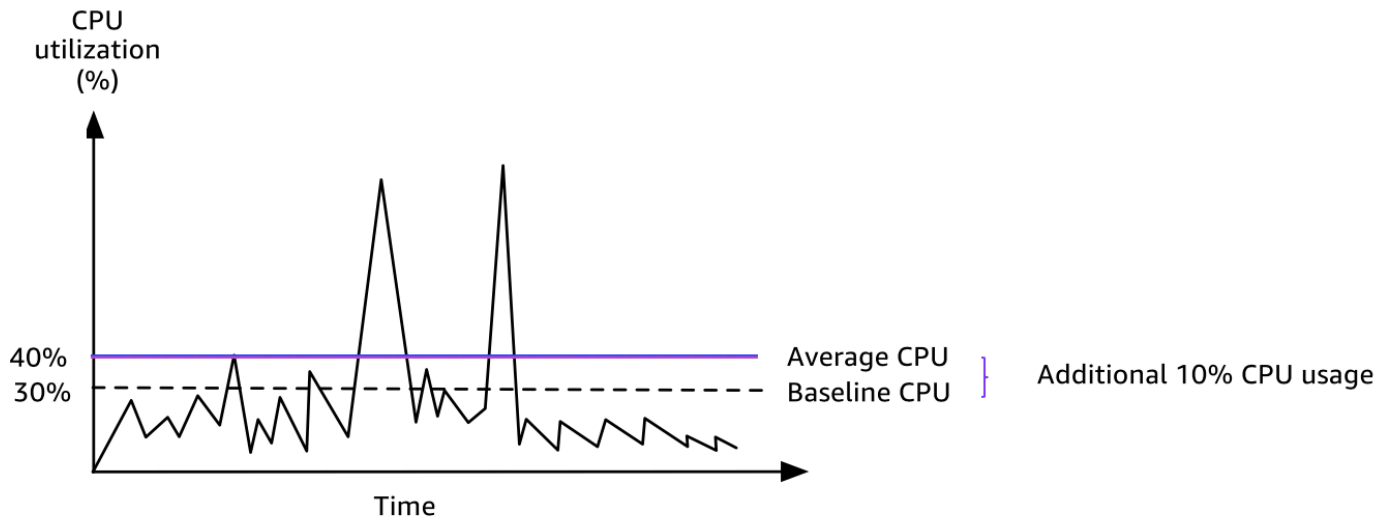
`unlimited` 모드는 성능 순간 확장 가능 인스턴스에 사용할 수 있는 크레딧 구성 옵션입니다. 이 모드는 실행 중인 또는 중지된 인스턴스에 대해 언제든지 활성화 또는 비활성화할 수 있습니다. 버스트 가능 성능 인스턴스 패밀리별로 각 AWS 리전의 계정 수준에서 `unlimited`를 기본 크레딧 옵션으로 설정하면 계정의 모든 새로운 버스트 가능 성능 인스턴스가 기본 크레딧 옵션을 사용하여 시작됩니다.

무제한 성능 순간 확장 가능 인스턴스의 작동 방식

`unlimited`로 구성된 성능 순간 확장 가능 인스턴스의 CPU 크레딧 밸런스가 감소하면 잉여 크레딧을 사용하여 기준 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.large의 CPU 사용량을 보여줍니다. t3.large에 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.



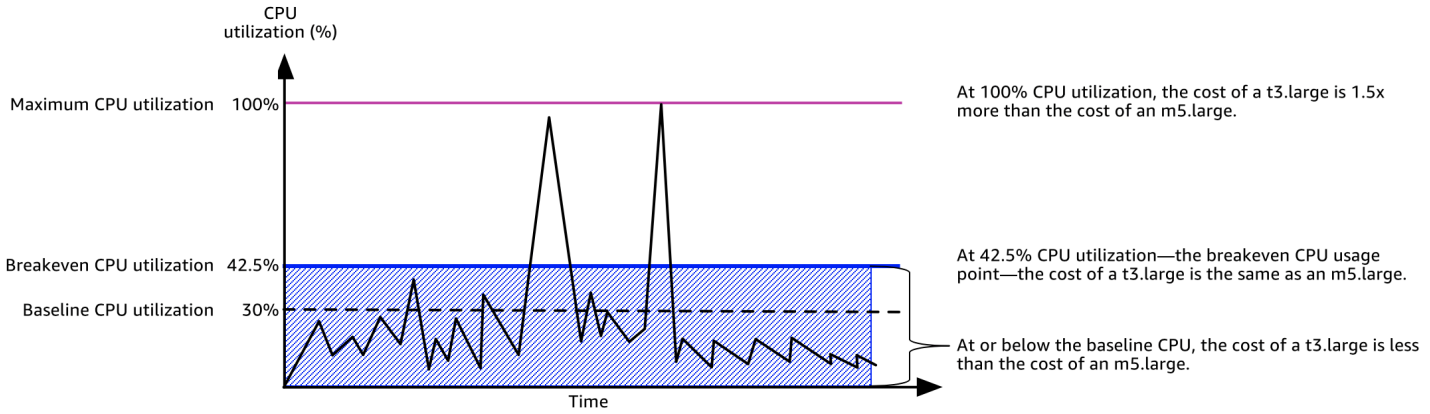
각 인스턴스 유형별 vCPU 당 기준 사용률 및 각 인스턴스 유형에서 얻은 크레딧 수에 대한 자세한 내용은 크레딧 표를 참조하세요.

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.large이 m5.large와 동일한 비용의 손익분기 CPU 사용량을 보여줍니다. t3.large에 대한 손익분기 CPU 사용량은 42.5%입니다. 평균 CPU 사용량이 42.5%인 경우 t3.large을 실행하는 비용은 m5.large와 동일하며 평균 CPU 사용량이 42.5%를 초과하면 비용이 더 많이 듭니다. 작업 부하가 42.5% 미만의 평균 CPU 사용량이 필요한 경우 t3.large와 동일한 성능을 얻는 동안 m5.large의 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	vCPU당 T3 기준 사용률(%)	잉여 크레딧에 대한 vCPU 시간당 요금	vCPU 분당 요금	vCPU당 사용 가능한 추가 버스트 시간(분)	사용 가능한 추가 CPU %	손익분기 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	$0.00083 USD	15	12.5%	42.5%

* 가격은 us-east-1 및 Linux OS를 기준으로 합니다.

이 테이블에서는 다음 정보를 제공합니다.

- A열은 인스턴스 유형인 t3.large을 표시합니다.
- B열은 t3.large에 대한 vCPU 수를 나타냅니다.
- C열은 시간당 t3.large의 가격을 보여줍니다.
- D열은 시간당 m5.large의 가격을 보여줍니다.
- D열은 t3.large과 m5.large 사이의 가격 차이를 보여줍니다.
- F열은 30%인 t3.large의 vCPU당 기준 사용률을 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 분당 고정 추가 요금을 보여줍니다.
- I열은 t3.large이 시간당 100% CPU에서 버스트 가능하고 m5.large와 같은 시간당 가격을 지불하는 추가 시간(분)을 보여줍니다.
- J열은 m5.large로 동일한 가격을 지불하면서 인스턴스가 버스트 가능한 기준선에 대한 추가 CPU 사용량(%)을 보여줍니다.
- K열은 t3.large이 m5.large보다 많은 비용을 들이지 않고 버스트 가능한 손익분기 CPU 사용량(%)을 보여줍니다. t3.large 비용 및 그 어떤 비용도 m5.large보다 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 수를 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. 요금에 대한 자세한 내용은 T2/T3/T4g 무제한 모드 요금을 참조하세요.

이전에 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 unlimited에서 standard로 전환됩니다.

소비한 잉여 크레딧은 CloudWatch 지표 CPUSurplusCreditBalance에 의해 추적이 가능합니다. 요금이 부과된 잉여 크레딧은 CloudWatch 지표 CPUSurplusCreditsCharged에 의해 추적이 가능합니다. 자세한 내용은 성능 순간 확장 가능 인스턴스에 대한 추가 CloudWatch 측정치 섹션을 참조하세요.

T2 무제한 인스턴스에는 시작 크레딧이 없음

T2 스탠다드 인스턴스에서는 시작 크레딧을 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 standard에서 unlimited으로 전환된 경우 남은 CPUCreditBalance가 전달되기 전에 CPUCreditBalance에서 누적된 시작 크레딧이 모두 삭제됩니다.

T4g, T3a, T3 인스턴스는 기본적으로 무제한 모드에서 시작되므로 시작 크레딧을 받지 않아 시작 즉시 버스트할 수 있습니다. 무제한 모드 크레딧 구성을 통해 T4G, T3a, T3 인스턴스는 필요한 만큼의 CPU를 사용하여 필요한 시간만큼 얼마든지 기준 이상으로 버스트할 수 있습니다.

무제한 모드 활성화

실행 중이거나 중지된 인스턴스에서 언제든지 unlimited에서 standard로, standard에서 unlimited로 전환할 수 있습니다. 자세한 내용은 시작 시 크레딧 사양 구성 및 성능 버스트 가능 인스턴스의 크레딧 사양 관리 섹션을 참조하세요.

버스트 가능 성능 인스턴스 패밀리별로 각 AWS 리전의 계정 수준에서 unlimited를 기본 크레딧 옵션으로 설정하면 계정의 모든 새로운 버스트 가능 성능 인스턴스가 기본 크레딧 옵션을 사용하여 시작됩니다. 자세한 내용은 계정의 기본 크레딧 사양 관리 섹션을 참조하세요.

Amazon EC2 콘솔 또는 AWS CLI를 사용하여 버스트 가능 성능 인스턴스가 unlimited 또는 standard로 구성되었는지 확인할 수 있습니다. 자세한 내용은 성능 버스트 가능 인스턴스 구성 섹션을 참조하세요.

무제한과 스탠다드 간 전환 시 크레딧에 발생하는 현상

CPUCreditBalance는 인스턴스에서 누적한 크레딧 수를 추적하는 CloudWatch 측정치입니다. CPUSurplusCreditBalance는 인스턴스에서 사용한 잉여 크레딧 수를 추적하는 CloudWatch 측정치입니다.

unlimited로 구성된 인스턴스를 standard로 변경하면 다음이 발생합니다.

- CPUCreditBalance 값은 변경되지 않은 채 전달됩니다.
- CPUSurplusCreditBalance 값은 즉시 요금이 부과됩니다.

standard 인스턴스가 unlimited로 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 성능 버스트 가능 인스턴스에 대한 CPU 크레딧 모니터링 섹션을 참조하세요.

성능 버스트 가능 인스턴스의 무제한 모드 예제

다음은 unlimited로 구성된 인스턴스에 크레딧 사용을 설명하는 예입니다.

예시:

- 예 1: T3 무제한의 크레딧 사용 설명
- 예 2: T2 무제한의 크레딧 사용 설명

예 1: T3 무제한의 크레딧 사용 설명

이 예에서는 t3.nano로 시작된 unlimited 인스턴스의 CPU 사용률과 CPU 사용률 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 CPUCreditBalance에 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 CPUCreditBalance가 감소한 이후에도 인스턴스가 잉여 크레딧을 사용하여 어떻게 버스트를 할 수 있는지 보여줍니다. 아래 워크플로는 그래프에서 번호가 매겨진 지점을 참조합니다.

P1 – 그래프의 0시간에서 인스턴스는 unlimited로 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 CPUCreditUsage는 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.

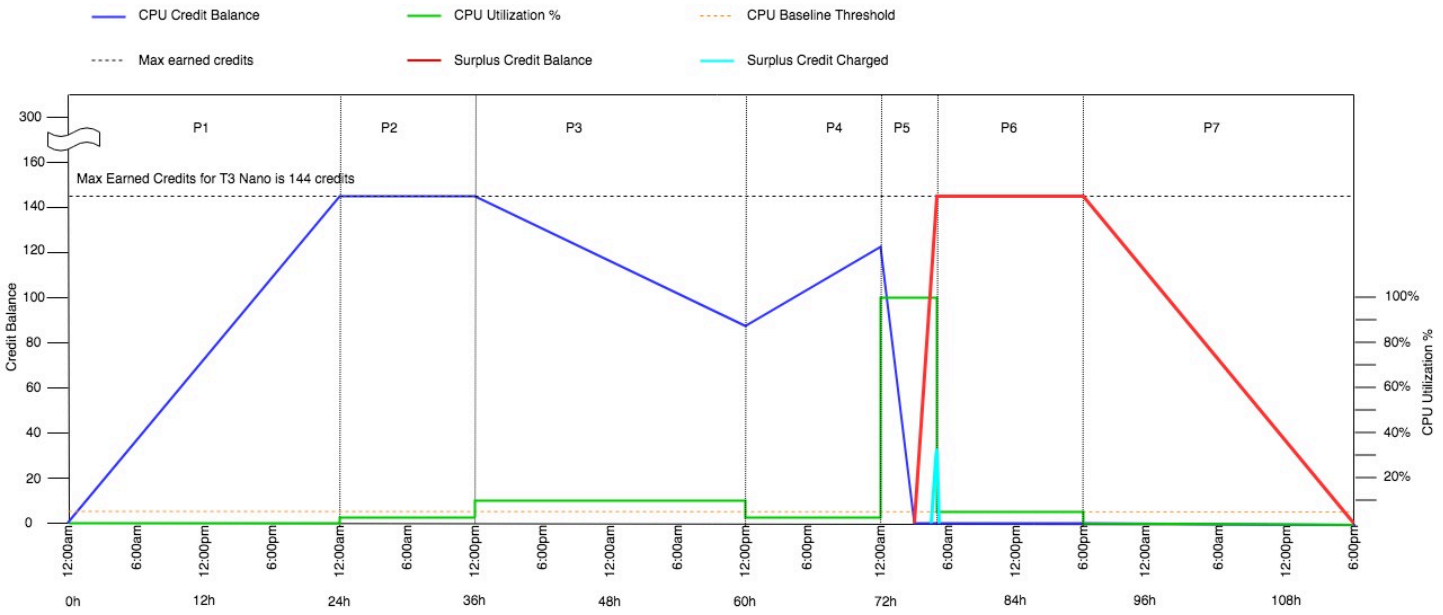
P4 – 향후 12시간 동안 CPU 사용률은 2.5%(기준보다 낮음)로 감소하며, 이를 위해서는 36 크레딧을 사용해야 합니다. 인스턴스에서는 동시에 72 크레딧을 획득할 수 있습니다. 인스턴스는 사용하는 크레딧보다 더 많은 크레딧을 획득하므로 CPUCreditBalance 값은 122 크레딧으로 증가합니다.

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

CPUCreditBalance를 소진하고 높은 CPU 사용률을 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 CPUSurplusCreditBalance를 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 CPUSurplusCreditBalance를 청산하는 데 사용합니다.



예 2: T2 무제한의 크레딧 사용 설명

이 예에서는 t2.nano로 시작된 unlimited 인스턴스의 CPU 사용률과 CPU 사용률 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 CPUCreditBalance에 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 CPUCreditBalance가 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.

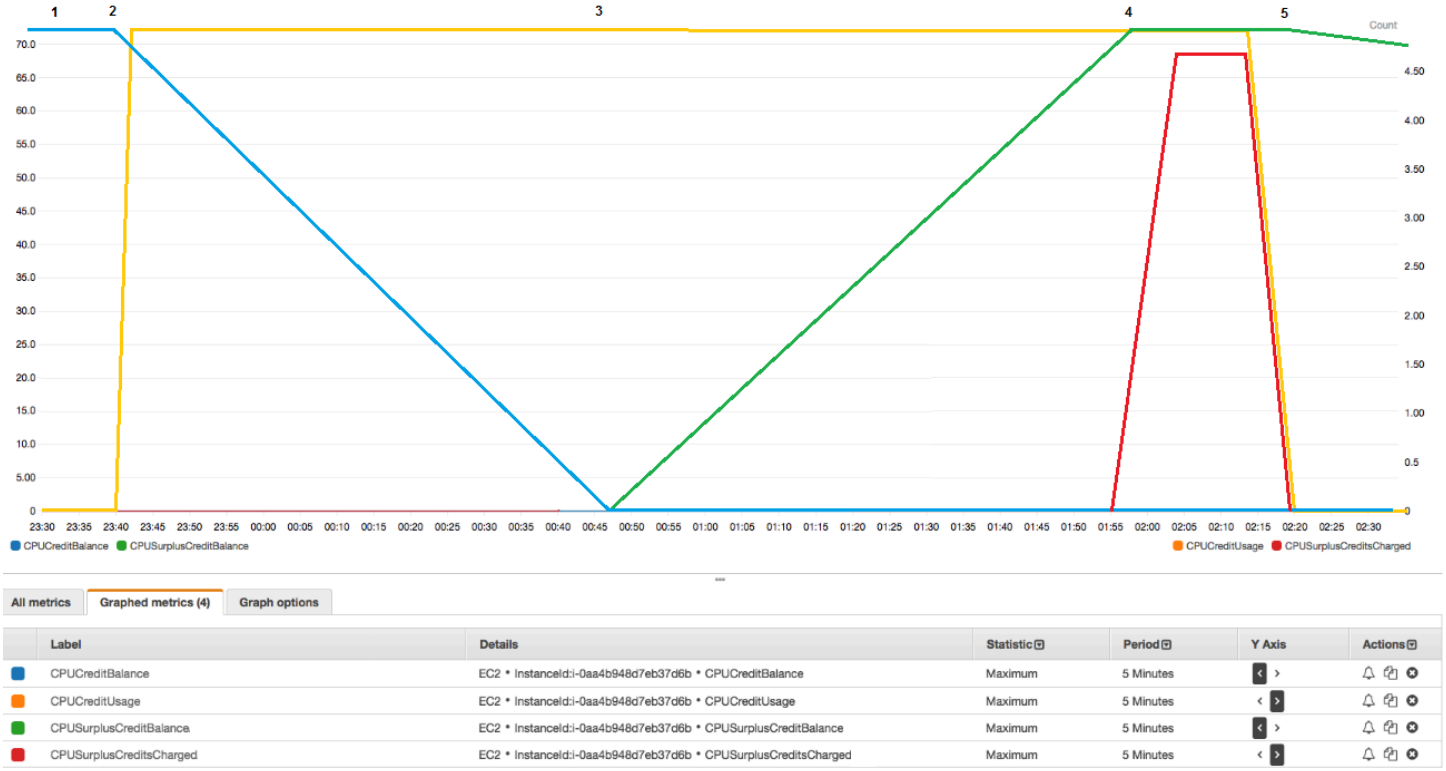
1 – 처음 10분 동안 CPUCreditUsage가 0이고 CPUCreditBalance 값이 최대 72로 유지됩니다.

2 – 23:40에 CPU 사용률이 증가하면서 인스턴스가 CPU 크레딧을 사용하고, 이에 따라 CPUCreditBalance 값이 줄어듭니다.

3 – 00:47경, 인스턴스에서 전체 CPUCreditBalance가 고갈되고 높은 CPU 사용률을 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 크레딧은 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 CPUSurplusCreditBalance를 청산하는 데 사용됩니다. CPUSurplusCreditBalance 값이 줄어들어 0이 되고 나면 인스턴스는 5분마다 0.25개씩 CPUCreditBalance 획득 크레딧을 누적하기 시작합니다.



청구서 계산(Linux 인스턴스)

잉여 크레딧은 vCPU-시간당 0.05 USD입니다. 인스턴스는 01:55부터 02:20까지 약 25개의 잉여 크레딧을 소비했으며, 이는 0.42 vCPU-시간에 해당됩니다. 이 인스턴스의 추가 요금은 0.42 vCPU-시간 x 0.05 USD/vCPU-시간 = 0.021 USD, 반올림하여 0.02 USD입니다. 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

청구서 계산(Windows 인스턴스)

잉여 크레딧은 vCPU-시간당 0.096 USD입니다. 인스턴스는 01:55부터 02:20까지 약 25개의 잉여 크레딧을 소비했으며, 이는 0.42 vCPU-시간에 해당됩니다. 이 인스턴스의 추가 요금은 0.42 vCPU-시간 x 0.096 USD/vCPU-시간 = 0.04032 USD, 반올림하여 0.04 USD입니다. 여기 이 T2 무제한 인스턴스에 대한 월말 청구서가 나와 있습니다.

Amazon Elastic Compute Cloud running Windows		
$0.0081 per On Demand Windows t2.nano Instance Hour	720.000 Hrs	$5.83

Amazon Elastic Compute Cloud T2 CPU Credits		
$0.096 per vCPU-Hour of T2 CPU credits	0.420 vCPU-Hours	$0.04

발생하는 모든 요금을 매시간 공지하는 청구서 알림을 설정하고 필요 시 조치를 취할 수 있습니다.

성능 순간 확장 가능 인스턴스의 스탠다드 모드

standard로 구성된 성능 순간 확장 가능 인스턴스는 평균 CPU 사용률이 인스턴스의 기준 CPU 사용률보다 일관되게 낮은 워크로드에 적합합니다. 기준 이상으로 버스트하려면 인스턴스는 CPU 크레딧 밸런스에 누적한 크레딧을 사용합니다. 인스턴스가 획득한 크레딧이 부족해지면 CPU 사용률이 점차적으로 기준 수준으로 떨어지기 때문에 획득한 CPU 크레딧 밸런스가 고갈되어도 급격한 성능 저하가 발생하지 않습니다. 자세한 내용은 성능 버스트 가능 인스턴스에 대한 주요 개념 및 정의 섹션을 참조하세요.

내용

- 성능 버스트 가능 인스턴스의 표준 모드 개념
  - 스탠다드 성능 순간 확장 가능 인스턴스의 작동 방식
  - 시작 크레딧
  - 시작 크레딧 한도
  - 시작 크레딧과 획득 크레딧의 차이
- 성능 버스트 가능 인스턴스의 표준 모드 예제
  - 예 1: T3 스탠다드의 크레딧 사용 설명
  - 예 2: T2 스탠다드의 크레딧 사용 설명
    - 기간 1: 1 – 24시간
    - 기간 2: 25 – 36시간
    - 기간 3: 37 – 61시간
    - 기간 4: 62 – 72시간
    - 기간 5: 73 – 75시간
    - 기간 6: 76 – 90시간
    - 기간 7: 91 – 96시간

성능 버스트 가능 인스턴스의 표준 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. 버스트 가능 성능 인스턴스 패밀리별로 각 AWS 리전의 계정 수준에서 standard를 기본 크레딧 옵션으로 설정하면 계정의 모든 새로운 버스트 가능 성능 인스턴스가 기본 크레딧 옵션을 사용하여 시작됩니다.

스탠다드 성능 순간 확장 가능 인스턴스의 작동 방식

standard로 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 크레딧이 모두 손실되고 크레딧 밸런스가 0으로 재설정됩니다. 인스턴스가 다시 시작되면 새로운 세트의 시작 크레딧이 지급되고 획득 크레딧이 누적되기 시작합니다. T4g, T3a, T3 표준 인스턴스의 경우 인스턴스가 중지된 후 CPU 크레딧 밸런스가 7일 동안 지속되며 이후 크레딧이 소멸됩니다. 7일 이내에 인스턴스를 시작하면 크레딧이 상실되지 않습니다.

T2 표준 인스턴스는 획득 크레딧과 시작 크레딧이라는 두 가지 유형의 CPU 크레딧을 획득합니다. 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.

T4g, T3a, T3 인스턴스는 무제한 모드를 지원하므로 시작 크레딧을 획득하지 않습니다. 무제한 모드 크레딧 구성을 통해 T4G, T3a, T3 인스턴스는 필요한 만큼의 CPU를 사용하여 필요한 시간만큼 얼마든지 기준 이상으로 버스트할 수 있습니다.

시작 크레딧

T2 표준 인스턴스는 시작 시 vCPU당 30개의 시작 크레딧을 받고 T1 표준 인스턴스는 15개의 시작 크레딧을 받습니다. 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 표를 참조하세요.

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 전환 또는 크기가 더 큰 인스턴스 사용을 고려하세요.

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 표를 참조하세요.
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 표를 참조하세요.
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 표에서 `CPUCreditBalance` 항목을 참조하세요.

성능 버스트 가능 인스턴스의 표준 모드 예제

다음은 인스턴스가 `standard`로 구성되었을 때의 크레딧 사용을 설명하는 예입니다.

예시:

- 예 1: T3 스탠다드의 크레딧 사용 설명
- 예 2: T2 스탠다드의 크레딧 사용 설명

예 1: T3 스탠다드의 크레딧 사용 설명

이 예에서는 t3.nano로 시작된 standard 인스턴스가 획득 크레딧을 획득, 축적, 사용하는 방식을 보여줍니다. 이로써 누적된 획득 크레딧이 크레딧 밸런스에 반영되는 방식을 알 수 있습니다.

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 표를 참조하세요.

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 standard로 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 CPUCreditUsage는 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.

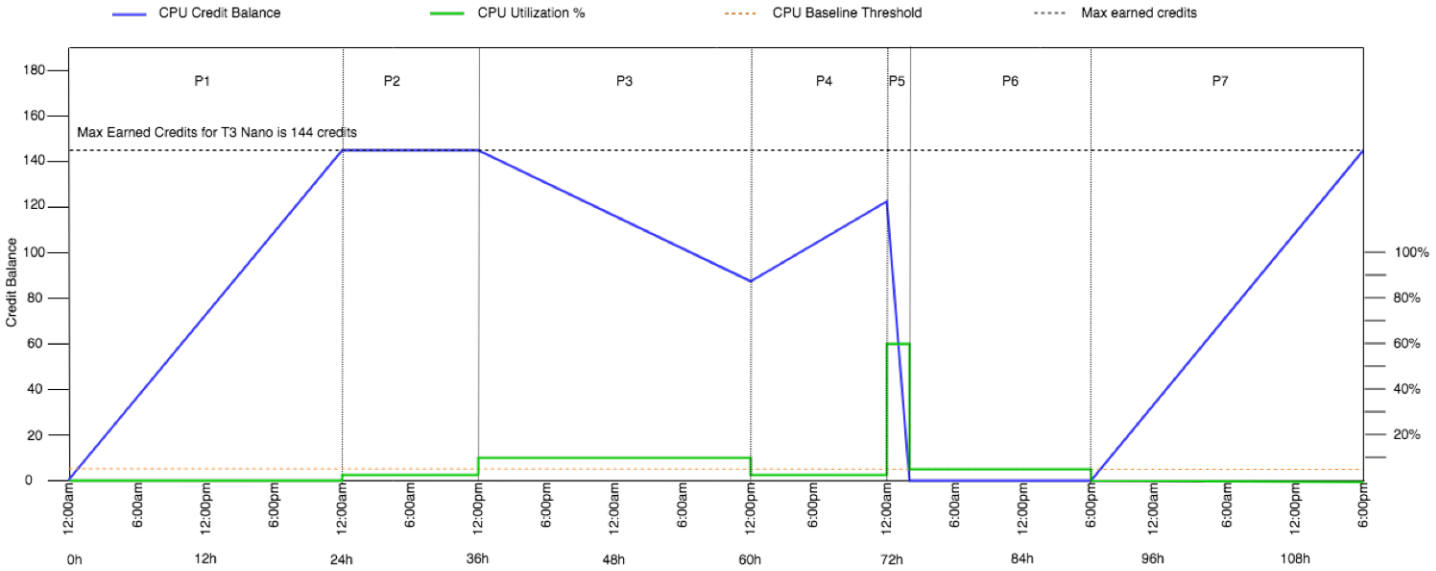
P3 – 향후 24시간 동안 CPU 사용률은 7%(기준보다 높음)이며, 이를 위해서는 57.6 크레딧을 사용해야 합니다. 인스턴스는 획득한 것보다 더 많은 크레딧을 사용하므로 CPUCreditBalance 값은 86.4 크레딧으로 감소합니다.

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 60% CPU 사용률로 버스트하고 전체 CPUCreditBalance 값인 122크레딧을 소진합니다. 이 기간이 종료되는 시점에 CPUCreditBalance가 0이고, CPU 사용률은 강제로 5%의 기준 사용률 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 CPUCreditBalance에 누적됩니다.



예 2: T2 스탠다드의 크레딧 사용 설명

이 예제는 t2.nano로 실행된 standard 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 표를 참조하세요. 제한에 대한 자세한 내용은 시작 크레딧 한도 섹션을 참조하세요.

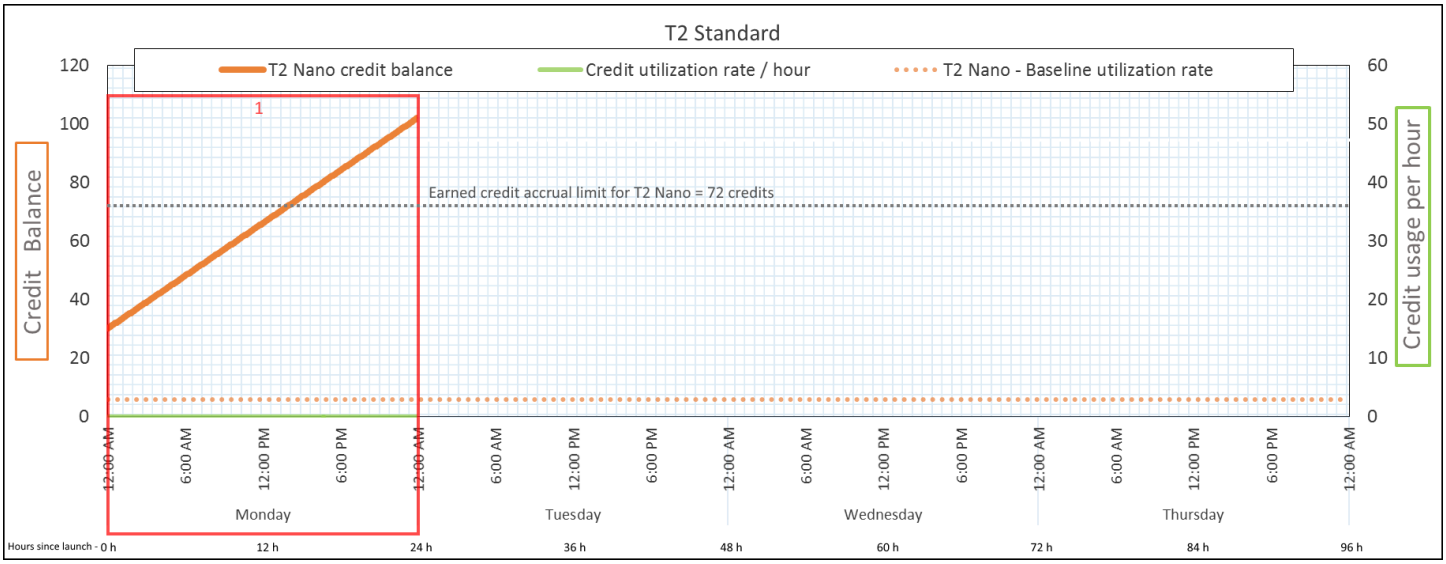
T2 스탠다드 인스턴스를 시작하고 즉시 사용할 수 있습니다. 또는 T2 스탠다드 인스턴스를 시작하고 애플리케이션을 실행하기 전에 며칠 동안 유휴 상태로 둘 수 있습니다. 인스턴스를 사용했는지 아니면 유휴 상태로 두었는지에 따라 크레딧이 사용되는지 또는 누적되는지가 결정됩니다. 인스턴스가 시작 시간부터 24시간 동안 유휴 상태로 유지된 경우, 잔고에 획득 누적 크레딧과 시작 누적 크레딧이 모두

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)

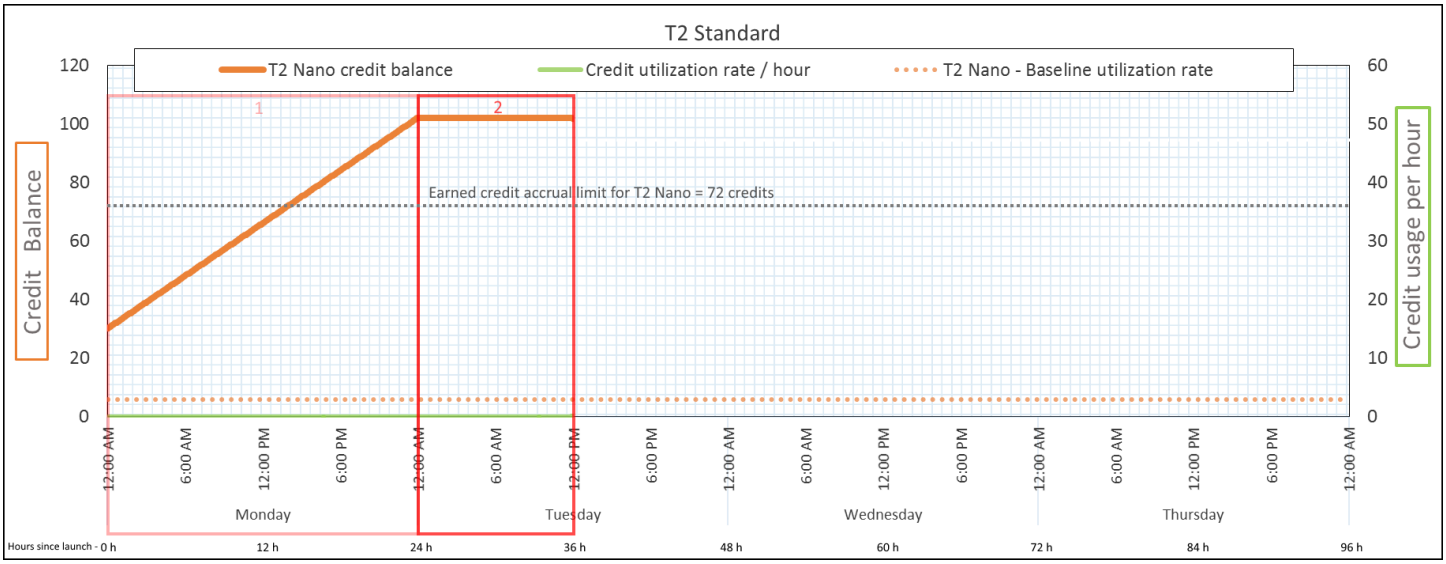
결론

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

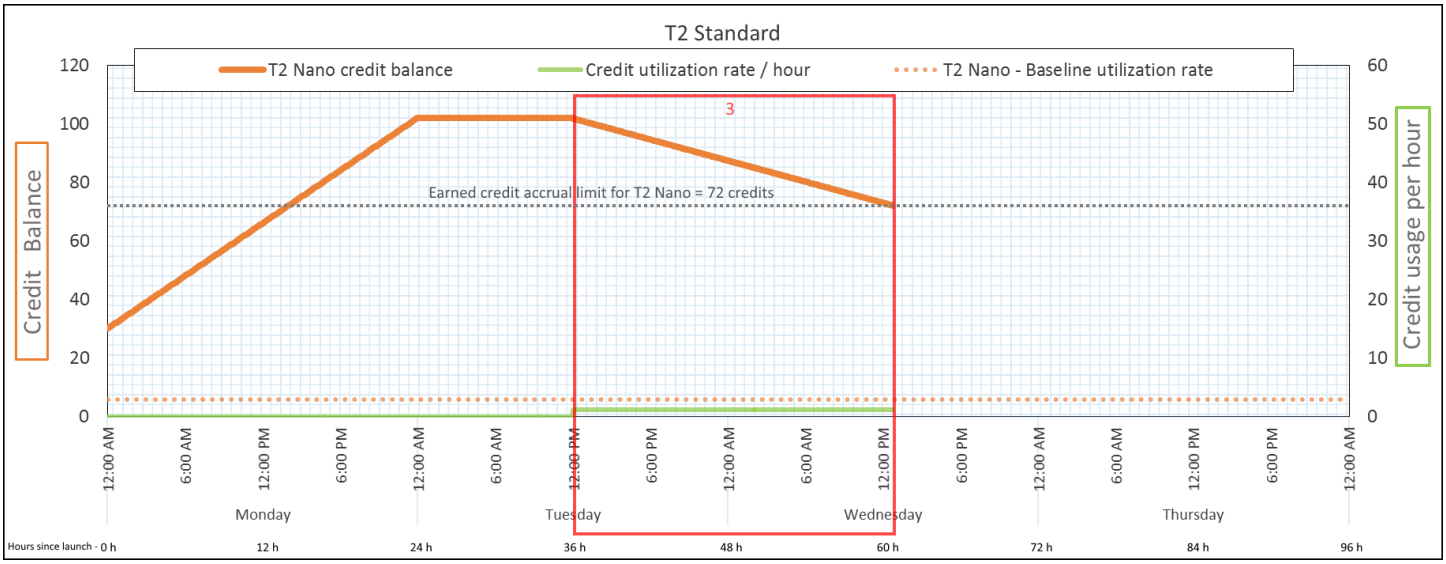
결론

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 한

도에 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 40%) – 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)

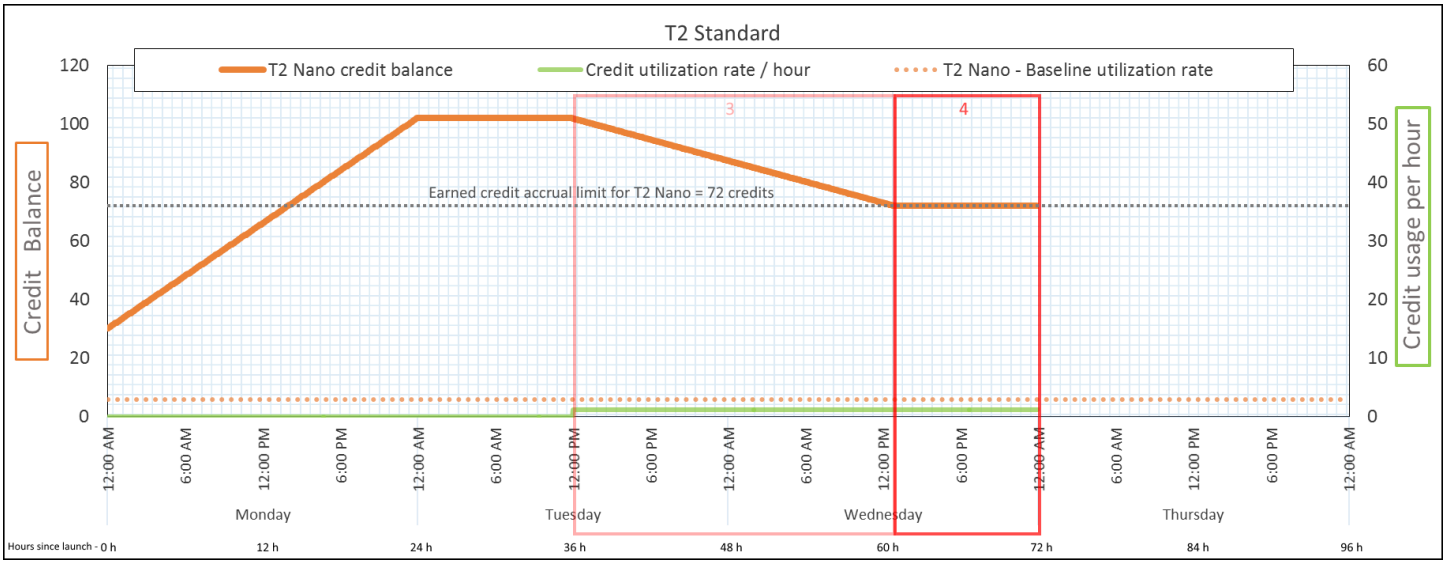
결론

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.



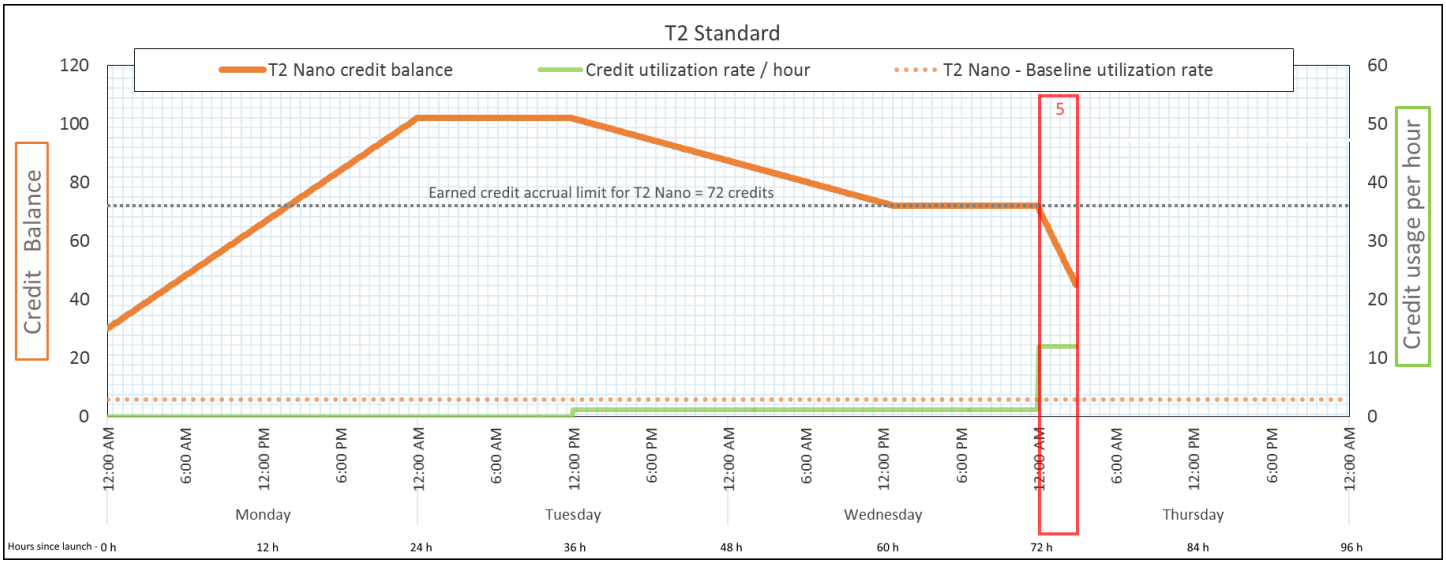
크레딧 사용률	24시간당 28.8 크레딧(시간당 1.2 크레딧, 2% CPU 사용률, 크레딧 획득률 40%) &endash; 11시간 동안 13.2 크레딧—
크레딧 획득률	24시간당 72 크레딧
크레딧 폐기율	24시간당 43.2 크레딧(크레딧 획득률 60%)
크레딧 밸런스	72 크레딧(시작 크레딧 0, 획득 크레딧 72) &endash;— 잔고가 한도에 이름

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

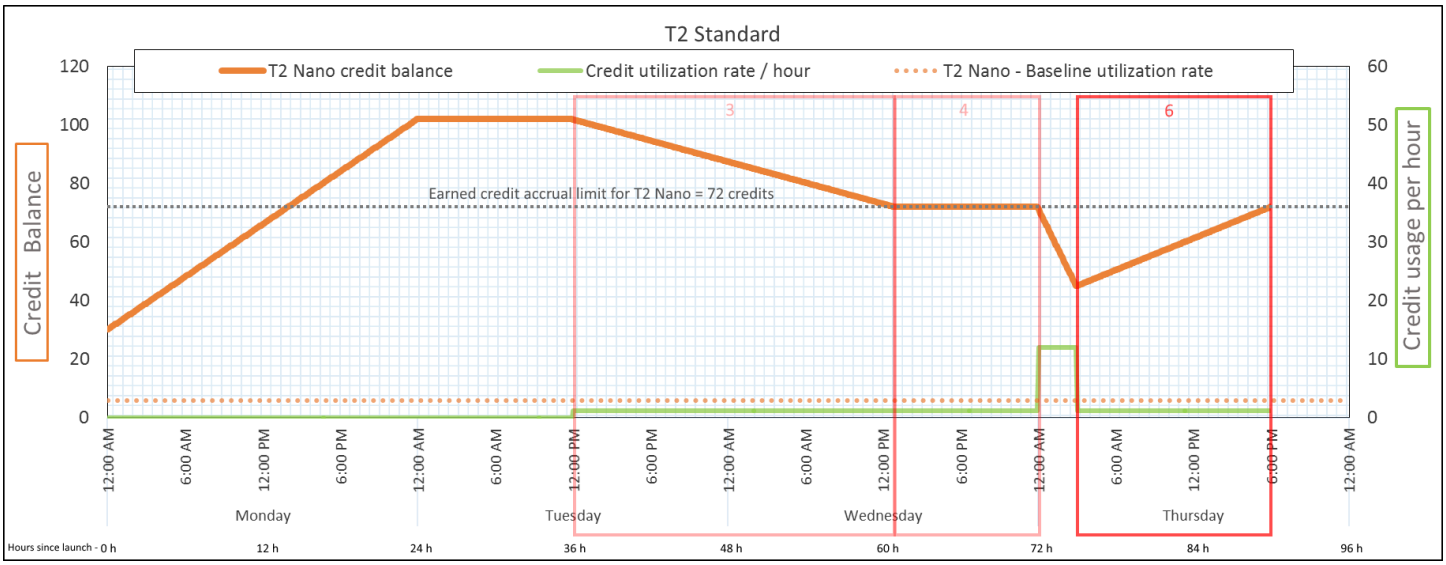
인스턴스가 획득한 것보다 더 많은 크레딧을 사용한 경우 크레딧 밸런스가 감소합니다.

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)

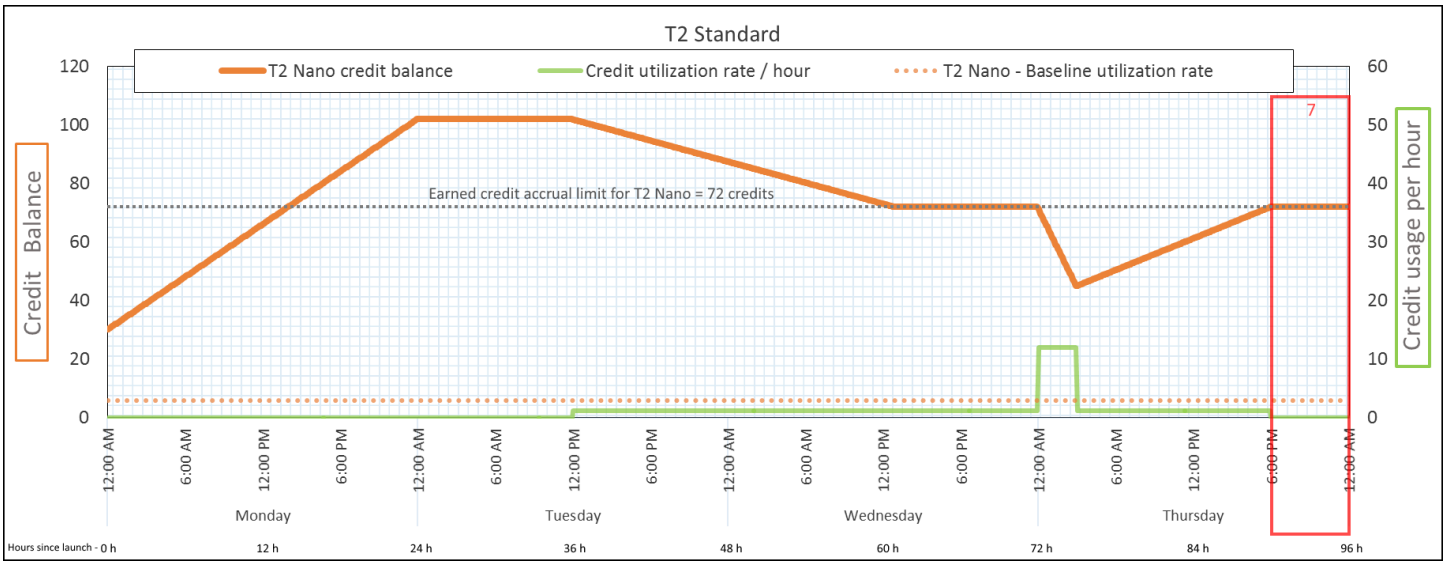
결론

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 표를 참조하세요.

성능 버스트 가능 인스턴스 구성

성능 버스트 기능이 있는 인스턴스(T 인스턴스) 시작, 모니터링 및 수정 절차는 서로 유사합니다. 주요 차이점은 인스턴스가 시작할 때 기본 적용되는 크레딧 사양입니다.

각 T 인스턴스 패밀리는 다음과 같은 기본 크레딧 사양과 함께 제공됩니다.

- T4g, T3a, T3 인스턴스는 `unlimited`로 시작
- 전용 호스트의 T3 인스턴스는 `standard`로만 시작됩니다.
- T2 인스턴스는 `standard`로 시작

계정의 기본 크레딧 사양을 변경할 수 있습니다.

업무

- 시작 시 크레딧 사양 구성
- 크레딧 사양을 unlimited로 설정하도록 Auto Scaling 그룹 구성
- 성능 버스트 가능 인스턴스의 크레딧 사양 관리
- 계정의 기본 크레딧 사양 관리

시작 시 크레딧 사양 구성

T 인스턴스를 `unlimited` 또는 `standard` 크레딧 사양으로 시작할 수 있습니다.

다음 절차에서는 EC2 콘솔 또는 AWS CLI를 사용하는 방법을 설명합니다. Auto Scaling 사용에 대한 자세한 내용은 크레딧 사양을 unlimited로 설정하도록 Auto Scaling 그룹 구성 섹션을 참조하세요.

Console

시작 시 인스턴스의 크레딧 사양을 구성하려면

1. 인스턴스 시작 절차를 따릅니다.
2. 인스턴스 유형(Instance type)에서 T 인스턴스 유형을 선택합니다.
3. Advanced details(고급 세부 정보)를 확장합니다. 크레딧 사양에서 크레딧 사양을 선택합니다.
4. Summary(요약) 패널에서 인스턴스 구성을 검토한 다음 Launch instance(인스턴스 시작)를 선택합니다.

AWS CLI

시작 시 인스턴스의 크레딧 사양을 설정하려면

run-instances 명령을 `--credit-specification` 옵션과 함께 사용합니다.

```
--credit-specification CpuCredits=unlimited
```

PowerShell

시작 시 인스턴스의 크레딧 사양을 설정하려면

`-CreditSpecification_CpuCredit` 파라미터와 함께 New-EC2Instance cmdlet을 사용합니다.

```
-CreditSpecification_CpuCredit unlimited
```

크레딧 사양을 unlimited로 설정하도록 Auto Scaling 그룹 구성

T 인스턴스가 시작되거나 시작되면 우수한 부트스트랩 경험을 위해 CPU 크레딧이 필요합니다. 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.

Console

크레딧 사양을 설정하는 시작 템플릿을 생성하려면

1. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Create a launch template using advanced settings 절차를 따르세요.
2. 시작 템플릿 콘텐츠(Launch template contents)의 인스턴스 유형(Instance type)에서 인스턴스 크기를 선택합니다.
3. Auto Scaling 그룹에서 인스턴스를 `unlimited`으로 시작하려면 고급 세부 정보(Advanced details) 아래의 크레딧 사양(Credit specification)에서 무제한(Unlimited)을 선택합니다.

4. 시작 템플릿 파라미터 정의를 완료한 경우 시작 템플릿 생성을 선택합니다.

AWS CLI

크레딧 사양을 설정하는 시작 템플릿을 생성하려면

create-launch-template 명령을 사용합니다.

```
aws ec2 create-launch-template \
    --launch-template-name my-launch-template \
    --version-description FirstVersion \
    --launch-template-data CreditSpecification={CpuCredits=unlimited}
```

PowerShell

크레딧 사양을 설정하는 시작 템플릿을 생성하려면

New-EC2LaunchTemplate cmdlet을 사용합니다. 다음과 같이 시작 템플릿 데이터의 크레딧 사양을 정의합니다.

```
$creditSpec = New-Object Amazon.EC2.Model.CreditSpecificationRequest
$creditSpec.CpuCredits = "unlimited"
$launchTemplateData = New-Object Amazon.EC2.Model.RequestLaunchTemplateData
$launchTemplateData.CreditSpecification = $creditSpec
```

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.

Console

시작 템플릿을 사용하여 Auto Scaling 그룹 생성

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 생성을 참조하세요.

기존 Auto Scaling 그룹에 시작 템플릿 추가

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.

AWS CLI

시작 템플릿을 사용하여 Auto Scaling 그룹 생성

create-auto-scaling-group 명령을 사용하여 `--launch-template` 파라미터를 지정합니다.

기존 Auto Scaling 그룹에 시작 템플릿 추가

update-auto-scaling-group 명령을 사용하고 `--launch-template` 파라미터를 지정합니다.

PowerShell

시작 템플릿을 사용하여 Auto Scaling 그룹 생성

New-ASAutoScalingGroup cmdlet을 사용하고 `-LaunchTemplate_LaunchTemplateId` 또는 `-LaunchTemplate_LaunchTemplateName` 파라미터를 지정합니다.

기존 Auto Scaling 그룹에 시작 템플릿 추가

Update-ASAutoScalingGroup cmdlet을 사용하고 `-LaunchTemplate_LaunchTemplateId` 또는 `-LaunchTemplate_LaunchTemplateName` 파라미터를 지정합니다.

성능 버스트 가능 인스턴스의 크레딧 사양 관리

실행 중이거나 중지된 T 인스턴스의 크레딧 사양을 `unlimited`와 `standard` 간에 언제든지 전환할 수 있습니다.

unlimited 모드에서는 인스턴스가 잉여 크레딧을 사용할 수 있으며 이로 인해 추가 요금이 발생할 수 있습니다. 자세한 내용은 잉여 크레딧으로 요금 발생 가능 섹션을 참조하세요.

Console

인스턴스의 크레딧 사양을 관리하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 인스턴스를 선택합니다.
3. (선택 사항) 인스턴스를 선택합니다. 세부 정보 탭에서 크레딧 사양을 찾습니다. 이때 값은 unlimited 또는 standard입니다.
4. (선택 사항) 여러 인스턴스의 크레딧 사양을 동시에 수정하려면 인스턴스를 모두 선택합니다.
5. 작업(Actions), 인스턴스 설정(Instance settings), 크레딧 사양 변경(Change credit specification)을 선택합니다. 이 옵션은 T 인스턴스를 선택한 경우에만 활성화됩니다.
6. 무제한 모드에서 각 인스턴스 ID 옆의 확인란을 선택하거나 선택 취소합니다.

AWS CLI

인스턴스의 크레딧 사양을 가져오는 방법

describe-instance-credit-specifications 명령을 사용합니다. 인스턴스 ID를 지정하지 않으면 크레딧 사양이 unlimited인 모든 인스턴스가 반환됩니다. 출력에는 이전에 unlimited 크레딧 사양으로 구성된 인스턴스도 포함됩니다. 예를 들어 T3 인스턴스가 unlimited로 구성된 가운데 이를 M4 인스턴스로 크기 조정하는 경우 Amazon EC2에서 M4 인스턴스를 반환합니다.

```
aws ec2 describe-instance-credit-specifications \
    --instance-id i-1234567890abcdef0 \
    --query InstanceCreditSpecifications[].CpuCredits \
    --output text
```

출력의 예시는 다음과 같습니다.

```
unlimited
```

인스턴스의 크레딧 사양을 설정하려면

modify-instance-credit-specification 명령을 사용합니다.

```
aws ec2 modify-instance-credit-specification \
```

```
    --region us-east-1 \
    --instance-credit-specification
 "InstanceId=i-1234567890abcdef0,CpuCredits=unlimited"
```

PowerShell

인스턴스의 크레딧 사양을 가져오는 방법

Get-EC2CreditSpecification cmdlet을 사용합니다.

```
(Get-EC2CreditSpecification `
    -InstanceId i-1234567890abcdef0).CpuCredits
```

출력의 예시는 다음과 같습니다.

```
unlimited
```

인스턴스의 크레딧 사양을 설정하려면

Edit-EC2InstanceCreditSpecification cmdlet을 사용합니다.

```
Edit-EC2InstanceCreditSpecification `
    -Region us-east-1 `
    -InstanceCreditSpecification @({InstanceId="i-1234567890abcdef0"
 CpuCredits="unlimited"})
```

계정의 기본 크레딧 사양 관리

각 T 인스턴스 패밀리는 ???기본 크레딧 사양과 함께 제공됩니다. AWS 리전별로 계정 수준에서 각 T 인스턴스 패밀리에 대해 기본 크레딧 사양을 변경할 수 있습니다. 기본 크레딧 사양의 유효한 값은 `unlimited` 및 `standard`입니다.

EC2 콘솔에서 인스턴스 시작 마법사를 사용하여 인스턴스를 시작하면 크레딧 사양에 대해 선택한 값이 계정 수준 기본 크레딧 사양을 재정의합니다. AWS CLI를 사용하여 인스턴스를 시작하면 계정에 속한 새 T 인스턴스가 모두 기본 크레딧 사양을 사용하여 시작됩니다. 실행 중이거나 중지된 기존 인스턴스의 크레딧 사양은 영향을 받지 않습니다.

고려 사항

인스턴스 패밀리의 기본 크레딧 사양은 5분 동안 한 번만 수정할 수 있으며, 24시간 동안 4회까지 수정할 수 있습니다.

Console

기본 크레딧 사양을 관리하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. AWS 리전을(를) 변경하려면 페이지의 오른쪽 상단 모서리에 있는 리전 선택기를 사용합니다.
3. 왼쪽 탐색 창에서 [EC2 대시보드(EC2 Dashboard)]를 선택합니다.
4. 계정 속성에서 [기본 크레딧 사양(Default credit specification)]을 선택합니다.
5. 관리를 선택합니다.
6. 각 인스턴스 패밀리에 대해 [무제한(Unlimited)] 또는 [표준(Standard)]을 선택한 다음 [업데이트(Update)]를 선택합니다.

AWS CLI

기본 크레딧 사양을 가져오려면

get-default-credit-specification 명령을 사용하세요.

```
aws ec2 get-default-credit-specification \
    --region us-east-1 \
    --instance-family t2 \
    --query InstanceFamilyCreditSpecifications[].CpuCredits \
    --output text
```

출력의 예시는 다음과 같습니다.

```
standard
```

기본 크레딧 사양을 설정하려면

modify-default-credit-specification 명령을 사용하십시오. 다음 예제는 값을 `unlimited`로 설정합니다.

```
aws ec2 modify-default-credit-specification \
    --region us-east-1 \
    --instance-family t2 \
    --cpu-credits unlimited
```

PowerShell

기본 크레딧 사양을 가져오려면

Get-EC2DefaultCreditSpecification cmdlet을 사용합니다.

```
(Get-EC2DefaultCreditSpecification `
    -Region us-east-1 `
    -InstanceFamily t2).CpuCredits
```

출력의 예시는 다음과 같습니다.

```
standard
```

기본 크레딧 사양을 설정하려면

Edit-EC2DefaultCreditSpecification cmdlet을 사용합니다. 다음 예제는 값을 `unlimited`로 설정합니다.

```
Edit-EC2DefaultCreditSpecification `
    -Region us-east-1 `
    -InstanceFamily t2 `
    -CpuCredit unlimited
```

성능 버스트 가능 인스턴스에 대한 CPU 크레딧 모니터링

EC2는 지표를 Amazon CloudWatch로 전송합니다. CPU 크레딧 지표는 CloudWatch 콘솔의 Amazon EC2 인스턴스별 지표에서 또는 각 인스턴스에 대한 지표를 나열하는 AWS CLI를 사용하여 확인할 수 있습니다. 자세한 내용은 인스턴스에 사용 가능한 CloudWatch 지표 섹션을 참조하세요.

내용

- 성능 순간 확장 가능 인스턴스에 대한 추가 CloudWatch 측정치
- CPU 크레딧 사용량 계산

성능 순간 확장 가능 인스턴스에 대한 추가 CloudWatch 측정치

버스트 가능한 성능 인스턴스에는 다음과 같은 추가 CloudWatch 지표가 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 사용률을 유지하기 위해 소비되는 잉여 CPU 크레딧 수.
- `CPUSurplusCreditsCharged` – 24시간 동안 획득할 수 있는 최대 CPU 크레딧 수를 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 사용 가능한 CloudWatch 지표 섹션을 참조하세요.

측정치	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 사용하세요. 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.

측정치	설명
	인스턴스가 실행 중인 동안 CPUCreditBalance 의 크레딧은 만료되지 않습니다. T4g, T3a 또는 T3 인스턴스가 중지되면 CPUCreditBalance 값은 7일 동안 유지됩니다. 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	unlimited 값이 0일 때 CPUCreditBalance 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 standard로 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.

스탠다드 인스턴스에 대한 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 표를 참조하세요.

예제

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

버스트 가능한 성능 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 CPU를 버스트할 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 balance라는 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 수는 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 `CPUSurplusCreditBalance`로 지정되고 `CPUCreditBalance`가 `0`으로 설정됩니다. 즉 누적할 수 있는 최대 크레딧 수는 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 `CPUSurplusCreditBalance`로 추적된 모든 잉여 크레딧에 요금이 부과됩니다. 인스턴스가 `unlimited`에서 `standard`로 전환되면 나머지 모든 `CPUSurplusCreditBalance`에도 요금이 부과됩니다.

GPU 인스턴스를 사용한 성능 가속화

GPU 기반 인스턴스는 수천 개의 컴퓨팅 코어로 NVIDIA GPU에 대한 액세스를 제공합니다. 이러한 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.

GPU 기반 인스턴스를 활성화하거나 최적화하려면 먼저 다음과 같이 적절한 드라이버를 설치해야 합니다.

- P3 또는 G4dn 인스턴스와 같이 NVIDIA GPU가 연결된 인스턴스에 NVIDIA 드라이버를 설치하려면 NVIDIA 드라이버 섹션을 참조하세요.
- G4ad 인스턴스와 같이 AMD GPU가 연결된 인스턴스에 AMD 드라이버를 설치하려면 AMD 드라이버 섹션을 참조하세요.

내용

- Amazon EC2 GPU 기반 인스턴스에서 NVIDIA GRID 가상 애플리케이션 활성화
- Amazon EC2 인스턴스의 GPU 설정 최적화
- G4ad Linux 인스턴스에서 듀얼 4K 디스플레이 설정
- GPU 가속 인스턴스 시작하기

Amazon EC2 GPU 기반 인스턴스에서 NVIDIA GRID 가상 애플리케이션 활성화

NVIDIA GPU가 있는 GPU 기반 인스턴스에서 GRID 가상 애플리케이션을 활성화하려면(NVIDIA GRID 가상 워크스테이션은 기본적으로 활성화되어 있음) 드라이버의 제품 유형을 정의해야 합니다. 사용하는 프로세스는 인스턴스의 운영 체제에 따라 달라집니다.

Linux 인스턴스

Linux 인스턴스에서 GRID 가상 애플리케이션을 활성화하려면 다음을 수행하세요.

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 `0`과 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
   ```

Windows 인스턴스

Windows 인스턴스에서 GRID 가상 애플리케이션을 활성화하려면 다음을 수행하세요.

1. regedit.exe를 실행하여 레지스트리 편집기를 엽니다.
2. `HKEY_LOCAL_MACHINE\SOFTWARE\NVIDIA Corporation\Global\GridLicensing`으로 이동합니다.

3. 오른쪽 창에서 마우스 오른쪽 버튼을 클릭하여 컨텍스트 메뉴를 열고 새로 생성(New)과 DWORD를 차례로 선택합니다.
4. 이름에 FeatureType을 입력한 다음 `Enter`를 입력합니다.
5. FeatureType에서 마우스 오른쪽 버튼을 클릭하여 컨텍스트 메뉴를 열고 수정(Modify)을 선택합니다.
6. 값 데이터(Value data)에서 NVIDIA GRID 가상 애플리케이션에 대해 `0`을 입력하고 확인(OK)을 선택합니다.
7. 오른쪽 창에서 마우스 오른쪽 버튼을 클릭하여 컨텍스트 메뉴를 열고 새로 생성(New)과 DWORD를 차례로 선택합니다.
8. 이름(Name)에 IgnoreSP를 입력한 다음 `Enter`를 누릅니다.
9. IgnoreSP에서 마우스 오른쪽 버튼을 클릭하여 컨텍스트 메뉴를 열고 수정(Modify)을 선택합니다.
10. 값 데이터(Value data)에 1을 입력하고 확인(OK)을 선택합니다.
11. 레지스트리 편집기를 닫습니다.

Amazon EC2 인스턴스의 GPU 설정 최적화

NVIDIA GPU 인스턴스에서 최고의 성능을 달성하기 위해 수행할 수 있는 몇 가지 GPU 설정 최적화가 있습니다. 이러한 인스턴스 유형 중 일부에서 NVIDIA 드라이버는 GPU 클록 속도에 변화를 주는 자동 부스트 기능을 사용합니다. 자동 부스트를 비활성화하고 GPU 클록 속도를 최대 주파수로 설정하면 GPU 인스턴스의 성능을 최대로 유지할 수 있습니다.

Linux에서 GPU 설정 최적화

1. GPU 설정을 영구적으로 구성합니다. 이 명령은 실행하는 데 몇 분이 소요될 수 있습니다.

   ```
   [ec2-user ~]$ sudo nvidia-persistenced
   ```

2. [G3 및 P2 인스턴스에 대해] 모든 GPU의 자동 부스트 기능을 비활성화합니다.

   ```
   [ec2-user ~]$ sudo nvidia-smi --auto-boost-default=0
   ```

3. 모든 GPU 클록 속도를 최대 주파수로 설정합니다. 다음 명령에 지정된 메모리와 그래픽 클록 속도를 사용합니다.

   일부 버전의 NVIDIA 드라이버는 응용 프로그램 클럭 속도 설정을 지원하지 않으며 무시할 수 있는 오류("`Setting applications clocks is not supported for GPU...`")를 표시합니다.

- G3 인스턴스:

```
[ec2-user ~]$ sudo nvidia-smi -ac 2505,1177
```

- G4dn 인스턴스:

```
[ec2-user ~]$ sudo nvidia-smi -ac 5001,1590
```

- G5 인스턴스:

```
[ec2-user ~]$ sudo nvidia-smi -ac 6250,1710
```

- G6 및 Gr6 인스턴스:

```
[ec2-user ~]$ sudo nvidia-smi -ac 6251,2040
```

- G6e 인스턴스:

```
[ec2-user ~]$ sudo nvidia-smi -ac 9001,2520
```

- P2 인스턴스:

```
[ec2-user ~]$ sudo nvidia-smi -ac 2505,875
```

- P3 및 P3dn 인스턴스::

```
[ec2-user ~]$ sudo nvidia-smi -ac 877,1530
```

- P4d 인스턴스:

```
[ec2-user ~]$ sudo nvidia-smi -ac 1215,1410
```

- P4de instances:

```
[ec2-user ~]$ sudo nvidia-smi -ac 1593,1410
```

- P5 인스턴스:

```
[ec2-user ~]$ sudo nvidia-smi -ac 2619,1980
```

Windows에서 GPU 설정 최적화

1. PowerShell 창을 열고 NVIDIA 설치 폴더를 탐색합니다.

```
PS C:\> cd "C:\Windows\System32\DriverStore\FileRepository\nvgridsw_aws.inf_*\"
```

2. [G3 및 P2 인스턴스에 대해] 모든 GPU의 자동 부스트 기능을 비활성화합니다.

```
PS C:\> .\nvidia-smi --auto-boost-default=0
```

3. 모든 GPU 클록 속도를 최대 주파수로 설정합니다. 다음 명령에 지정된 메모리와 그래픽 클록 속도를 사용합니다.

   일부 버전의 NVIDIA 드라이버는 응용 프로그램 클럭 속도 설정을 지원하지 않으며 무시할 수 있는 오류("Setting applications clocks is not supported for GPU...")를 표시합니다.

   - G3 인스턴스:

   ```
   PS C:\> .\nvidia-smi -ac "2505,1177"
   ```

   - G4dn 인스턴스:

   ```
   PS C:\> .\nvidia-smi -ac "5001,1590"
   ```

   - G5 인스턴스:

   ```
   PS C:\> .\nvidia-smi -ac "6250,1710"
   ```

   - G6 및 Gr6 인스턴스:

   ```
   PS C:\> .\nvidia-smi -ac "6251,2040"
   ```

   - G6e 인스턴스:

   ```
   PS C:\> .\nvidia-smi -ac "9001,2520"
   ```

   - P2 인스턴스:

   ```
   PS C:\> .\nvidia-smi -ac "2505,875"
   ```

- P3 및 P3dn 인스턴스::

```
PS C:\> .\nvidia-smi -ac "877,1530"
```

G4ad Linux 인스턴스에서 듀얼 4K 디스플레이 설정

G4ad 인스턴스를 시작한 후 듀얼 4K 디스플레이를 설정할 수 있습니다.

AMD 드라이버를 설치하고 듀얼 스크린을 구성하려면 다음을 수행하세요.

1. Linux 인스턴스에 연결하여 듀얼 4K(2x4k)를 사용하려는 대상 GPU의 PCI 버스 주소를 가져옵니다.

   ```
   lspci -vv | grep -i amd
   ```

   출력은 다음과 비슷합니다.

   ```
   00:1e.0 Display controller: Advanced Micro Devices, Inc. [*AMD*/ATI] Device 7362 (rev
    c3)
   Subsystem: Advanced Micro Devices, Inc. [AMD/ATI] Device 0a34
   ```

2. 참고로 위의 출력에서 PCI 버스 주소는 00:1e.0입니다. `/etc/modprobe.d/amdgpu.conf`라는 이름의 파일을 만들고 다음을 추가합니다.

   ```
   options amdgpu virtual_display=0000:00:1e.0,2
   ```

3. Linux에 AMD 드라이버를 설치하려면 EC2 인스턴스용 AMD 드라이버 섹션을 참조하세요. AMD GPU 드라이버가 이미 설치되어 있는 경우, dkms를 통해 amdgpu 커널 모듈을 다시 빌드해야 합니다.

4. 아래 xorg.conf 파일을 사용하여 듀얼(2x4K) 화면 토폴로지를 정의하고 파일을 `/etc/X11/xorg.conf:`에 저장합니다.

   ```
   ~$ cat /etc/X11/xorg.conf
   Section "ServerLayout"
       Identifier     "Layout0"
       Screen          0 "Screen0"
       Screen        1 "Screen1"
       InputDevice     "Keyboard0" "CoreKeyboard"
       InputDevice     "Mouse0" "CorePointer"
       Option          "Xinerama" "1"
   ```

```
EndSection
Section "Files"
    ModulePath "/opt/amdgpu/lib64/xorg/modules/drivers"
    ModulePath "/opt/amdgpu/lib/xorg/modules"
    ModulePath "/opt/amdgpu-pro/lib/xorg/modules/extensions"
    ModulePath "/opt/amdgpu-pro/lib64/xorg/modules/extensions"
    ModulePath "/usr/lib64/xorg/modules"
    ModulePath "/usr/lib/xorg/modules"
EndSection
Section "InputDevice"
    # generated from default
    Identifier     "Mouse0"
    Driver         "mouse"
    Option         "Protocol" "auto"
    Option         "Device" "/dev/psaux"
    Option         "Emulate3Buttons" "no"
    Option         "ZAxisMapping" "4 5"
EndSection
Section "InputDevice"
    # generated from default
    Identifier     "Keyboard0"
    Driver         "kbd"
EndSection

Section "Monitor"
    Identifier     "Virtual"
    VendorName     "Unknown"
    ModelName      "Unknown"
    Option         "Primary" "true"
EndSection

Section "Monitor"
    Identifier     "Virtual-1"
    VendorName     "Unknown"
    ModelName      "Unknown"
    Option         "RightOf" "Virtual"
EndSection

Section "Device"
    Identifier     "Device0"
    Driver         "amdgpu"
    VendorName     "AMD"
    BoardName      "Radeon MxGPU V520"
    BusID          "PCI:0:30:0"
```

```
EndSection

Section "Device"
    Identifier     "Device1"
    Driver         "amdgpu"
    VendorName     "AMD"
    BoardName      "Radeon MxGPU V520"
    BusID          "PCI:0:30:0"
EndSection

Section "Extensions"
    Option         "DPMS" "Disable"
EndSection

Section "Screen"
    Identifier     "Screen0"
    Device         "Device0"
    Monitor        "Virtual"
    DefaultDepth   24
    Option         "AllowEmptyInitialConfiguration" "True"
    SubSection "Display"
        Virtual    3840 2160
        Depth      32
    EndSubSection
EndSection

Section "Screen"
    Identifier     "Screen1"
    Device         "Device1"
    Monitor        "Virtual"
    DefaultDepth   24
    Option         "AllowEmptyInitialConfiguration" "True"
    SubSection "Display"
        Virtual    3840 2160
        Depth      32
    EndSubSection
EndSection
```

5. 대화형 데스크톱 설정 지침에 따라 DCV를 설정합니다.
6. DCV 설정이 완료되면 재부팅합니다.
7. 드라이버가 작동하는지 확인합니다.

```
dmesg | grep amdgpu
```

응답은 다음과 같아야 합니다.

```
Initialized amdgpu
```

8. DISPLAY=:0 xrandr -q에 대한 출력에서 2개의 가상 디스플레이가 연결되어 있음을 확인해야 합니다.

```
~$ DISPLAY=:0 xrandr -q
Screen 0: minimum 320 x 200, current 3840 x 1080, maximum 16384 x 16384
Virtual connected primary 1920x1080+0+0 (normal left inverted right x axis y axis)
 0mm x 0mm
 4096x3112  60.00
 3656x2664  59.99
 4096x2160  60.00
 3840x2160  60.00
 1920x1200  59.95
 1920x1080  60.00
 1600x1200  59.95
 1680x1050  60.00
 1400x1050  60.00
 1280x1024  59.95
 1440x900 59.99
 1280x960 59.99
 1280x854 59.95
 1280x800 59.96
 1280x720 59.97
 1152x768 59.95
 1024x768 60.00 59.95
 800x600  60.32 59.96 56.25
 848x480  60.00 59.94
 720x480  59.94
 640x480  59.94 59.94
Virtual-1 connected 1920x1080+1920+0 (normal left inverted right x axis y axis) 0mm x
 0mm
 4096x3112  60.00
 3656x2664  59.99
 4096x2160  60.00
 3840x2160  60.00
 1920x1200  59.95
```

```
 1920x1080  60.00
 1600x1200  59.95
 1680x1050  60.00
 1400x1050  60.00
 1280x1024  59.95
 1440x900 59.99
 1280x960 59.99
 1280x854 59.95
 1280x800 59.96
 1280x720 59.97
 1152x768 59.95
 1024x768 60.00 59.95
 800x600  60.32 59.96 56.25
 848x480  60.00 59.94
 720x480  59.94
640x480  59.94 59.94
```

9. DCV에 연결할 때 해상도를 2x4K로 변경하여 듀얼 모니터 지원이 DCV에 등록되었는지 확인합니다.

Linux용 대화형 데스크톱 설정

Linux 인스턴스에 AMD GPU 드라이버가 설치되어 있고 AMD GPU가 사용 중인지 확인한 후 대화형 데스크톱 관리자를 설치할 수 있습니다. 호환성 및 성능을 최대화하려면 MATE 데스크톱 환경을 사용하는 것이 좋습니다.

전제 조건

텍스트 편집기를 열고 다음을 `xorg.conf`라는 파일로 저장합니다. 인스턴스에 이 파일이 필요합니다.

```
Section "ServerLayout"
Identifier     "Layout0"
Screen          0 "Screen0"
InputDevice     "Keyboard0" "CoreKeyboard"
InputDevice     "Mouse0" "CorePointer"
EndSection
Section "Files"
ModulePath "/opt/amdgpu/lib64/xorg/modules/drivers"
ModulePath "/opt/amdgpu/lib/xorg/modules"
```

```
ModulePath "/opt/amdgpu-pro/lib/xorg/modules/extensions"
ModulePath "/opt/amdgpu-pro/lib64/xorg/modules/extensions"
ModulePath "/usr/lib64/xorg/modules"
ModulePath "/usr/lib/xorg/modules"
EndSection
Section "InputDevice"
# generated from default
Identifier     "Mouse0"
Driver         "mouse"
Option         "Protocol" "auto"
Option         "Device" "/dev/psaux"
Option         "Emulate3Buttons" "no"
Option         "ZAxisMapping" "4 5"
EndSection
Section "InputDevice"
# generated from default
Identifier     "Keyboard0"
Driver         "kbd"
EndSection
Section "Monitor"
Identifier     "Monitor0"
VendorName     "Unknown"
ModelName      "Unknown"
EndSection
Section "Device"
Identifier     "Device0"
Driver         "amdgpu"
VendorName     "AMD"
BoardName      "Radeon MxGPU V520"
BusID          "PCI:0:30:0"
EndSection
Section "Extensions"
Option         "DPMS" "Disable"
EndSection
Section "Screen"
Identifier     "Screen0"
Device         "Device0"
Monitor        "Monitor0"
DefaultDepth   24
Option         "AllowEmptyInitialConfiguration" "True"
SubSection "Display"
    Virtual    3840 2160
    Depth      32
EndSubSection
```

```
EndSection
```

Amazon Linux 2에서 대화형 데스크톱을 설정하려면

1. EPEL 리포지토리를 설치합니다.

```
[ec2-user ~]$ sudo amazon-linux-extras install epel -y
```

2. MATE 데스크톱을 설치합니다.

```
[ec2-user ~]$ sudo amazon-linux-extras install mate-desktop1.x -y
[ec2-user ~]$ sudo yum groupinstall "MATE Desktop" -y
[ec2-user ~]$ sudo systemctl disable firewalld
```

3. `xorg.conf` 파일을 `/etc/X11/xorg.conf`에 복사합니다.
4. 인스턴스를 재부팅합니다.

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

5. (선택 사항) Amazon DCV를 고성능 디스플레이 프로토콜로 사용하려면 Amazon DCV 서버를 설치한 다음 원하는 클라이언트를 사용하여 Amazon DCV 세션에 연결합니다.

Ubuntu에서 대화형 데스크톱을 설정하려면

1. MATE 데스크톱을 설치합니다.

```
$ sudo apt install xorg-dev ubuntu-mate-desktop -y
$ sudo apt purge ifupdown -y
```

2. `xorg.conf` 파일을 `/etc/X11/xorg.conf`에 복사합니다.
3. 인스턴스를 재부팅합니다.

```
$ sudo reboot
```

4. 적절한 버전의 Ubuntu에 대한 AMF 인코더를 설치합니다.

```
$ sudo apt install ./amdgpu-pro-20.20-*/amf-amdgpu-pro_20.20-*_amd64.deb
```

5. (선택 사항) Amazon DCV를 고성능 디스플레이 프로토콜로 사용하려면 Amazon DCV 서버를 설치한 다음 원하는 클라이언트를 사용하여 Amazon DCV 세션에 연결합니다.

6. DCV 설치 후 DCV 사용자에게 비디오 사용 권한을 부여합니다.

```
$ sudo usermod -aG video dcv
```

CentOS에서 대화형 데스크톱을 설정하려면

1. EPEL 리포지토리를 설치합니다.

```
$ sudo yum update -y
$ sudo yum install epel-release -y
```

2. MATE 데스크톱을 설치합니다.

```
$ sudo yum groupinstall "MATE Desktop" -y
$ sudo systemctl disable firewalld
```

3. `xorg.conf` 파일을 `/etc/X11/xorg.conf`에 복사합니다.
4. 인스턴스를 재부팅합니다.

```
$ sudo reboot
```

5. (선택 사항) Amazon DCV를 고성능 디스플레이 프로토콜로 사용하려면 Amazon DCV 서버를 설치한 다음 원하는 클라이언트를 사용하여 Amazon DCV 세션에 연결합니다.

GPU 가속 인스턴스 시작하기

다음 목록에 표시된 것과 같은 최신 세대의 GPU 가속 인스턴스 유형은 딥 러닝 및 고성능 컴퓨팅(HPC) 애플리케이션을 위한 최고 성능 기능을 제공합니다. 인스턴스 유형 링크를 선택하고 해당 기능을 자세히 알아보세요.

- P6 패밀리
- P5 패밀리

가속 인스턴스 유형의 전체 인스턴스 유형 사양 목록은 Amazon EC2 인스턴스 유형 참조의 가속 컴퓨팅을 참조하세요.

소프트웨어 구성

최신 세대의 GPU 가속 인스턴스 유형을 시작하는 가장 쉬운 방법은 필요한 모든 소프트웨어가 미리 구성되어 있는 AWS Deep Learning AMI를 사용하여 인스턴스를 시작하는 것입니다. GPU 가속 인스턴스 유형과 함께 사용할 수 있는 최신 AWS Deep Learning AMIs는 AWS Deep Learning AMIs 개발자 안내서의 P6 지원 DLAMI를 참조하세요.

사용자 지정 AMI를 구축하여 딥 러닝 또는 HPC 애플리케이션을 호스팅하는 인스턴스를 시작해야 하는 경우 다음과 같은 최소 소프트웨어 버전을 기본 이미지 위에 설치하는 것이 좋습니다.

인스턴스 유형	NVIDIA 드라이버	CUDA	NVIDIA GDRCopy	EFA 설치 프로그램	NCCL	EFA K8s*
P5	530	12.1	2.3	1.24.1	2.18.3	0.4.4
P5e	550	12.1	2.3	1.24.1	2.18.3	0.5.5
P5en	550	12.1	2.3	1.24.1	2.18.3	0.5.6
P6-B200	570	12.8	2.5	1.4.1	2.26.2-1	0.5.10

* EFA K8s 열에는 `aws-efa-k8s-device-plugin`의 최소 권장 버전이 포함되어 있습니다.

> **Note**
>
> EFA 설치 관리자 버전 1.4.1을 사용하는 경우 `aws-ofi-nccl plugin`이 함께 제공됩니다. 이전 버전의 EFA 설치 관리자는 `aws-ofi-nccl plugin` 버전 `1.7.2-aws` 이상을 사용합니다.

또한 심화된 C 상태를 사용하지 않도록 인스턴스를 구성하는 것이 좋습니다. 자세한 내용은 Amazon Linux 2 사용 설명서의 C 상태 심화 제한을 통한 고성능 및 저지연 시간을 참조하세요. 최신 AWS Deep Learning Base GPU AMI는 심화된 C 상태를 사용하지 않도록 미리 구성되어 있습니다.

네트워킹 및 Elastic Fabric Adapter (EFA) 구성에 대한 자세한 내용은 여러 네트워크 카드를 사용하여 Amazon EC2 인스턴스의 네트워크 대역폭 극대화를 참조하세요.

Amazon EC2 Mac 인스턴스

EC2 Mac 인스턴스는 iPhone, iPad, Mac, Vision Pro, Apple Watch, Apple TV 및 Safari와 같은 Apple 플랫폼용 애플리케이션의 개발, 구축, 테스트 및 서명에 적합합니다. SSH 또는 Apple Remote Desktop(ARD)을 사용하여 Mac 인스턴스에 연결할 수 있습니다.

> **Note**
>
> 청구 단위는 전용 호스트입니다. 해당 호스트에서 실행되는 인스턴스에는 추가 요금이 부과되지 않습니다.

Amazon EC2 Mac 인스턴스는 기본적으로 macOS 운영 체제를 지원합니다.

- EC2 x86 Mac 인스턴스(`mac1.metal`)는 3.2GHz Intel 8세대(Coffee Lake) Core i7 프로세서로 구동되는 2018 Mac mini 하드웨어를 기반으로 구축됩니다.
- EC2 M1 Mac 인스턴스(`mac2.metal`)는 Apple Silicon M1 프로세서로 구동되는 2020 Mac mini 하드웨어를 기반으로 구축됩니다.
- EC2 M1 Ultra Mac 인스턴스(`mac2-m1ultra.metal`)는 Apple Silicon M1 Ultra 프로세서로 구동되는 2022 Mac Studio 하드웨어를 기반으로 구축됩니다.
- EC2 M2 Mac 인스턴스(`mac2-m2.metal`)는 Apple Silicon M2 프로세서로 구동되는 2023 Mac mini 하드웨어를 기반으로 구축됩니다.
- EC2 M2 Pro Mac 인스턴스(`mac2-m2pro.metal`)는 Apple Silicon M2 Pro 프로세서로 구동되는 2023 Mac mini 하드웨어를 기반으로 구축됩니다.

내용

- 고려 사항
- 인스턴스 준비
- EC2 macOS AMI
- EC2 macOS Init
- macOS용 Amazon EC2 System Monitor
- 관련 리소스
- AWS Management Console 또는 AWS CLI를 사용하여 Mac 인스턴스 시작
- SSH 또는 GUI를 사용하여 인스턴스에 연결

- Amazon EC2 인스턴스에서 운영 체제 및 소프트웨어 업데이트
- Mac 인스턴스에서 EBS 볼륨 크기 늘리기
- Amazon EC2 Mac 인스턴스를 중지 또는 종료
- Amazon EC2 Mac 인스턴스의 시스템 무결성 보호 구성
- Amazon EC2 Mac 전용 호스트에 대해 지원되는 macOS 버전 찾기
- macOS AMI 알림 구독
- AWS Systems Manager Parameter Store API를 사용하여 macOS AMI ID 검색
- Amazon EC2 macOS AMI 릴리스 정보

고려 사항

Mac 인스턴스에는 다음과 같은 고려 사항이 적용됩니다.

- Mac 인스턴스는 전용 호스트에서 베어 메탈 인스턴스로만 사용할 수 있으며, 전용 호스트를 릴리스하기 전의 최소 할당 기간은 24시간입니다. 전용 호스트당 하나의 Mac 인스턴스를 시작할 수 있습니다. 전용 호스트를 AWS 조직 내의 AWS 계정 또는 조직 단위나 전체 AWS 조직과 공유할 수 있습니다.
- Mac 인스턴스는 다양한 AWS 리전에서 사용할 수 있습니다. 각 AWS 리전의 Mac 인스턴스 가용성 목록은 리전별 Amazon EC2 인스턴스 유형을 참조하세요.
- Mac 인스턴스는 온디맨드 인스턴스로만 사용할 수 있습니다. 스팟 인스턴스 또는 예약 인스턴스로 사용할 수 없습니다. Savings Plan을 구매하여 Mac 인스턴스 비용을 절감할 수 있습니다.
- 서로 다른 Mac 인스턴스 유형과 특정 macOS Amazon Machine Image(AMI)의 호환성은 다양합니다. 자세한 내용은 Amazon EC2 macOS AMI 릴리스 정보 섹션을 참조하세요.
- EBS 핫플러그가 지원됩니다.
- AWS는 Apple 하드웨어의 내부 SSD를 관리하거나 지원하지 않습니다. Amazon EBS 볼륨을 대신 사용하는 것이 좋습니다. EBS 볼륨은 다른 EC2 인스턴스에서와 마찬가지로 Mac 인스턴스에서도 동일한 탄력성, 가용성 및 내구성 이점을 제공합니다.
- 최적의 성능을 위해 Mac 인스턴스에는 IOPS가 10,000이고 처리량이 400MiB/s인 Amazon EBS 볼륨을 사용하는 것이 좋습니다. 자세한 내용은 Amazon EBS 사용 설명서의 Amazon EBS volume types를 참조하세요.
- Mac 인스턴스는 Amazon EC2 Auto Scaling을 지원합니다.
- x86 Mac 인스턴스에서는 자동 소프트웨어 업데이트가 비활성화됩니다. 인스턴스를 프로덕션에 배치하기 전에 업데이트를 적용하고 인스턴스에서 테스트하는 것이 좋습니다. 자세한 내용은 Amazon EC2 인스턴스에서 운영 체제 및 소프트웨어 업데이트 섹션을 참조하세요.

- Mac 인스턴스를 중지하거나 종료하면 전용 호스트에서 스크러빙 워크플로가 수행됩니다. 자세한 내용은 Amazon EC2 Mac 인스턴스를 중지 또는 종료 섹션을 참조하세요.
- **Important**

  외부 볼륨에서 Mac 하드웨어를 부팅할 때는 Apple Intelligence 기능을 사용할 수 없습니다. EC2 Mac 인스턴스는 기본적으로 외부 EBS 볼륨에서 부팅되므로 Apple Intelligence 기능을 지원하지 않습니다.
- **Warning**

  FileVault를 사용하지 마세요. FileVault를 활성화하면 파티션이 잠기므로 호스트가 부팅되지 않습니다. 데이터 암호화가 필요한 경우 Amazon EBS 암호화를 사용하여 부팅 문제와 성능 영향을 방지합니다. Amazon EBS 암호화를 사용하면 호스트 서버에서 암호화 작업이 진행되어 인스턴스와 인스턴스에 연결된 EBS 스토리지 간 저장 데이터와 전송 중 데이터의 보안을 모두 보장합니다. 자세한 내용은 Amazon EBS 사용 설명서의 Amazon EBS encryption을 참조하세요.

인스턴스 준비

Mac 인스턴스를 시작한 후에는 인스턴스가 준비될 때까지 기다려야 연결할 수 있습니다. x86 Mac 인스턴스 또는 Apple Silicon Mac 인스턴스가 있는 AWS 판매 AMI의 경우 시작 시간은 약 6분에서 20분 사이입니다. 선택한 Amazon EBS 볼륨 크기, 사용자 데이터에 대한 추가 스크립트 포함 또는 사용자 지정 macOS AMI에 추가로 로드된 소프트웨어에 따라 시작 시간이 늘어날 수 있습니다.

아래와 같은 작은 셸 스크립트를 사용하여 describe-instance-status API를 폴링하여 인스턴스가 연결할 준비가 된 시점을 알 수 있습니다. 다음 명령에서 예제 인스턴스 ID를 사용자의 ID로 대체합니다.

```
for i in $(seq 1 200); do aws ec2 describe-instance-status --instance-
ids=i-1234567890abcdef0 \
    --query='InstanceStatuses[0].InstanceStatus.Status'; sleep 5; done;
```

EC2 macOS AMI

Amazon EC2 macOS는 Amazon EC2 Mac 인스턴스에서 실행되는 개발자 워크로드를 위한 안정적이고 안전한 고성능 환경을 제공하도록 설계되었습니다. EC2 macOS AMI에는 시작 구성 도구와 널리 사용되는 주요 AWS 라이브러리 및 도구 등 AWS와 쉽게 통합하는 데 사용할 수 있도록 하는 패키지가 포함되어 있습니다.

EC2 macOS AMI에 대한 자세한 내용은 Amazon EC2 macOS AMI 릴리스 정보 섹션을 참조하세요.

AWS에서는 정기적으로 업데이트된 EC2 macOS AMI를 제공하며, 여기에는 AWS에서 소유한 패키지에 대한 업데이트와 완벽하게 테스트된 최신 macOS 버전이 포함되어 있습니다. 또한 AWS는 완벽하게 테스트되고 검증된 최신 마이너 버전 업데이트 또는 메이저 버전 업데이트로 업데이트된 AMI를 제공합니다. Mac 인스턴스에 대한 데이터 또는 사용자 지정을 유지할 필요가 없는 경우 현재 AMI를 사용하여 새 인스턴스를 시작한 다음 이전 인스턴스를 종료하여 최신 업데이트를 받을 수 있습니다. 그렇지 않은 경우 Mac 인스턴스에 적용할 업데이트를 선택할 수 있습니다.

macOS AMI 알림을 구독하는 방법에 대한 자세한 내용은 macOS AMI 알림 구독 섹션을 참조하세요.

EC2 macOS Init

EC2 macOS 초기화는 시작 시 EC2 Mac 인스턴스를 초기화하는 데 사용됩니다. 우선 순위 그룹을 사용하여 태스크의 논리적 그룹을 동시에 실행합니다.

launchd plist 파일은 `/Library/LaunchDaemons/com.amazon.ec2.macos-init.plist`입니다. EC2 macOS 초기화 파일은 `/usr/local/aws/ec2-macos-init`에 있습니다.

자세한 내용은 https://github.com/aws/ec2-macos-init를 참조하세요.

macOS용 Amazon EC2 System Monitor

macOS용 Amazon EC2 System Monitor는 Amazon CloudWatch에 CPU 사용률 지표를 제공합니다. 이러한 지표는 사용자 지정 직렬 디바이스를 통해 1분 단위로 CloudWatch로 전송됩니다. 다음과 같이 이 에이전트를 활성화하거나 비활성화할 수 있습니다. 기본적으로 활성화됩니다.

```
sudo setup-ec2monitoring [enable | disable]
```

> **Note**
>
> macOS용 Amazon EC2 System Monitor는 현재 Apple Silicon Mac 인스턴스에서 지원되지 않습니다.

관련 리소스

요금에 대한 자세한 내용은 요금을 참조하세요.

Mac 인스턴스에 대한 자세한 내용은 Amazon EC2 Mac 인스턴스를 참조하세요.

Mac 인스턴스의 하드웨어 사양 및 네트워크 성능에 대한 자세한 내용은 범용 인스턴스를 참조하세요.

AWS Management Console 또는 AWS CLI를 사용하여 Mac 인스턴스 시작

EC2 Mac 인스턴스는 전용 호스트를 필요로 합니다. 먼저 계정에 호스트를 할당한 다음 호스트에서 인스턴스를 시작해야 합니다.

AWS Management Console 또는 AWS CLI를 사용하여 Mac 인스턴스를 시작할 수 있습니다.

콘솔을 사용하여 Mac 인스턴스 시작

Mac 인스턴스를 전용 호스트로 시작하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 다음과 같이 전용 호스트를 할당합니다.
    a. 탐색 창에서 전용 호스트를 선택합니다.
    b. [전용 호스트 할당(Allocate Dedicated Host)]을 선택하고 다음을 수행합니다.
        i. 인스턴스 패밀리에서 mac1, mac2, mac2-m2, mac2-m2pro 또는 mac2-m1ultra를 선택합니다. 인스턴스 제품군이 목록에 표시되지 않으면 현재 선택한 리전에서 지원되지 않는 것입니다.
        ii. 인스턴스 유형에서 선택한 인스턴스 패밀리에 따라 mac1.metal, mac2.metal, mac2-m2.metal, mac2-m2pro.metal 또는 mac2-m1ultra.metal을 선택합니다.
        iii. [가용 영역(Availability Zone)]에서 전용 호스트의 가용 영역을 선택합니다.
        iv. Quantity(수량)에서 1을 그대로 둡니다.
        v. 할당을 선택합니다.
3. 다음과 같이 호스트에서 인스턴스를 시작합니다.
    a. 생성한 전용 호스트를 선택하고 다음을 수행합니다.
        i. Actions(작업), Launch instance(s) onto host(인스턴스를 호스트로 시작)를 차례로 선택합니다.
        ii. Application and OS Images (Amazon Machine Image)(애플리케이션 및 OS 이미지(Amazon Machine Image))에서 macOS AMI를 선택합니다.
        iii. 인스턴스 유형에서 적절한 인스턴스 유형(mac1.metal, mac2.metal, mac2-m2.metal, mac2-m2pro.metal 또는 mac2-m1ultra.metal)을 선택합니다.

iv. Advanced details(고급 세부 정보)에서 Tenancy(테넌시), Tenancy host by(다음 기준에 따른 테넌시 호스트), Tenancy host ID(테넌시 호스트 ID)가 생성한 전용 호스트에 따라 사전 구성되어 있는지 확인합니다. 필요한 경우 Tenancy affinity(테넌시 선호도)를 업데이트합니다.

v. 필요에 따라 EBS 볼륨, 보안 그룹 및 키 페어를 지정하여 마법사를 완료합니다.

vi. 요약(Summary) 패널에서 인스턴스 실행(Launch instance)을 선택합니다.

b. 확인 페이지에서 인스턴스가 실행 중인지 확인할 수 있습니다. 모든 인스턴스 보기(View all instances)를 선택하여 확인 페이지를 닫고 콘솔로 돌아갑니다. 인스턴스의 초기 상태는 `pending`입니다. 상태가 `running`으로 변경되고 상태 확인을 통과하면 인스턴스가 준비됩니다.

AWS CLI를 사용하여 Mac 인스턴스 시작

전용 호스트 할당

다음 allocate-hosts 명령을 사용하여 Mac 인스턴스 전용 호스트를 할당하고, `instance-type`은 `mac1.metal`, `mac2.metal`, `mac2-m2.metal`, `mac2-m2pro.metal` 또는 `mac2-m1ultra.metal`로 바꾸고 `region` 및 `availability-zone`은 환경에 적합한 것으로 바꿉니다.

```
aws ec2 allocate-hosts --region us-east-1 --instance-type mac1.metal --availability-
zone us-east-1b --auto-placement "on" --quantity 1
```

호스트에서 인스턴스 시작

다음 run-instances 명령을 사용하여 Mac 인스턴스를 시작하고, 다시 `instance-type`은 `mac1.metal`, `mac2.metal`, `mac2-m2.metal`, `mac2-m2pro.metal` 또는 `mac2-m1ultra.metal`로 바꾸고 `region` 및 `availability-zone`을 이전에 사용한 것으로 바꿉니다.

```
aws ec2 run-instances --region us-east-1 --instance-type mac1.metal --placement
 Tenancy=host --image-id ami_id --key-name my-key-pair
```

인스턴스의 초기 상태는 `pending`입니다. 상태가 `running`으로 변경되고 상태 확인을 통과하면 인스턴스가 준비됩니다. 다음 describe-instance-status 명령을 사용하여 인스턴스의 상태 정보를 표시합니다.

```
aws ec2 describe-instance-status --instance-ids i-017f8354e2dc69c4f
```

다음은 실행 중이고 상태 확인을 통과한 인스턴스에 대한 예시 출력입니다.


```
{
    "InstanceStatuses": [
        {
            "AvailabilityZone": "us-east-1b",
            "InstanceId": "i-017f8354e2dc69c4f",
            "InstanceState": {
                "Code": 16,
                "Name": "running"
            },
            "InstanceStatus": {
                "Details": [
                    {
                        "Name": "reachability",
                        "Status": "passed"
                    }
                ],
                "Status": "ok"
            },
            "SystemStatus": {
                "Details": [
                    {
                        "Name": "reachability",
                        "Status": "passed"
                    }
                ],
                "Status": "ok"
            }
        }
    ]
}
```


SSH 또는 GUI를 사용하여 인스턴스에 연결

SSH 또는 그래픽 사용자 인터페이스(GUI)를 사용하여 Mac 인스턴스에 연결할 수 있습니다.

여러 사용자가 OS에 동시에 액세스할 수 있습니다. 일반적으로 포트 5900에 내장된 화면 공유 서비스로 인한 1:1 user:GUI 세션이 있습니다. macOS에서 SSH를 사용하면 `sshd_config` 파일의 '최대 세션' 제한까지 여러 세션을 지원합니다.

SSH를 사용하여 인스턴스에 연결

Amazon EC2 Mac 인스턴스는 기본적으로 원격 루트 SSH를 허용하지 않습니다. ec2-user 계정은 SSH를 사용하여 원격으로 로그인하도록 구성됩니다. ec2-user 계정에는 sudo 권한도 있습니다. 인스턴스에 연결한 후에 다른 사용자를 추가할 수 있습니다.

SSH를 사용하여 인스턴스에 연결할 수 있도록 하려면 SSH 액세스를 허용하는 보안 그룹과 키 페어를 사용하여 인스턴스를 시작하고 인스턴스에 인터넷 연결이 있는지 확인합니다. 인스턴스에 연결할 때 키 페어에 대한 `.pem` 파일을 제공합니다.

SSH 클라이언트를 사용하여 Mac 인스턴스에 연결하려면 다음 프로시저를 사용하세요. 인스턴스에 연결을 시도하는 동안 오류가 발생한 경우 Amazon EC2 Linux 인스턴스 연결 문제 해결 섹션을 참조하세요.

SSH를 사용하여 인스턴스에 연결하려면

1. 명령줄에서 ssh를 입력하여 로컬 컴퓨터에 SSH 클라이언트가 설치되어 있는지 확인합니다. 컴퓨터가 명령을 인식하지 못하는 경우 운영 체제의 SSH 클라이언트를 검색하여 설치합니다.
2. 인스턴스의 퍼블릭 DNS 이름을 가져옵니다. Amazon EC2 콘솔을 사용하여 [세부 정보(Details)]와 [네트워킹(Networking)] 탭 모두에서 퍼블릭 DNS 이름을 찾을 수 있습니다. AWS CLI를 사용하면 describe-instances 명령을 사용하여 퍼블릭 DNS 이름을 찾을 수 있습니다.
3. 인스턴스를 시작할 때 지정한 키 페어에 대한 `.pem` 파일을 찾습니다.
4. 다음 ssh 명령에서 인스턴스의 퍼블릭 DNS 이름과 `.pem` 파일을 지정하여 인스턴스에 연결합니다.

```
ssh -i /path/key-pair-name.pem ec2-user@instance-public-dns-name
```

무차별 암호 대입 공격을 방지하기 위해 암호 인증이 비활성화됩니다. SSH 구성을 변경하기 전에 `/usr/local/aws/ec2-macos-init/init.toml`을 열고 `secureSSHDConfig`를 `false`로 설정합니다.

인스턴스의 GUI(그래픽 사용자 인터페이스)에 연결

다음 절차에 따라 VNC, Apple Remote Desktop(ARD) 또는 Apple Remote Desktop Sharing Desktop(macOS에 포함)을 사용하여 인스턴스의 GUI에 연결합니다.

> **Note**
>
> macOS 10.14 이상에서는 시스템 기본 설정에서 화면 공유를 사용하도록 설정한 경우에만 제어할 수 있습니다.

ARD 클라이언트 또는 VNC 클라이언트를 사용하여 인스턴스에 연결하려면

1. 로컬 컴퓨터에 ARD 클라이언트 또는 ARD를 지원하는 VNC 클라이언트가 설치되어 있는지 확인합니다. macOS에서는 기본 제공되는 화면 공유 애플리케이션을을 활용할 수 있습니다. 그렇지 않은 경우 운영 체제의 ARD를 검색하여 설치합니다.
2. 로컬 컴퓨터에서 SSH를 사용하여 인스턴스에 연결합니다.
3. 다음과 같이 passwd 명령을 사용하여 ec2-user 계정의 암호를 설정합니다.

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

4. 다음 명령을 사용하여 macOS 화면 공유를 설치하고 시작합니다.

   ```
   [ec2-user ~]$ sudo launchctl enable system/com.apple.screensharing
   sudo launchctl load -w /System/Library/LaunchDaemons/com.apple.screensharing.plist
   ```

5. exit를 입력하고 Enter 키를 눌러 인스턴스와의 연결을 끊습니다.
6. 컴퓨터에서 다음 ssh 명령을 사용하여 인스턴스에 연결합니다. 이전 섹션에 표시된 옵션 외에도 -L 옵션을 사용하여 포트 전달을 활성화하고 로컬 포트 5900의 모든 트래픽을 인스턴스의 ARD 서버로 전달합니다.

   ```
   ssh -L 5900:localhost:5900 -i /path/key-pair-name.pem ec2-user@instance-public-dns-name
   ```

7. 로컬 컴퓨터에서 ARD를 지원하는 ARD 클라이언트 또는 VNC 클라이언트를 사용하여 `localhost:5900`에 연결합니다. 예를 들어 다음과 같이 macOS에서 화면 공유 애플리케이션을 사용합니다.

   a. Finder를 열고 실행을 선택합니다.
   b. 서버에 연결을 선택합니다.
   c. 서버 주소 필드에 `vnc://localhost:5900`을 입력합니다.
   d. 메시지가 나타나면 사용자 이름 **`ec2-user`**와 ec2-user 계정에 대해 생성한 암호를 사용하여 로그인합니다.

Mac 인스턴스에서 macOS 화면 해상도 수정

ARD를 지원하는 ARD 또는 VNC 클라이언트를 사용하여 EC2 Mac 인스턴스에 연결한 후에는 displayplacer와 같이 공개적으로 사용 가능한 macOS 도구 또는 유틸리티를 사용하여 macOS 환경의 화면 해상도를 수정할 수 있습니다.

displayplacer를 사용하여 화면 해상도를 수정하려면

1. displayplacer를 설치합니다.

```
[ec2-user ~]$ brew tap jakehilborn/jakehilborn && brew install displayplacer
```

2. 현재 화면 정보 및 가능한 화면 해상도를 표시합니다.

```
[ec2-user ~]$ displayplacer list
```

3. 원하는 화면 해상도를 적용합니다.

```
[ec2-user ~]$ displayplacer "id:<screenID> res:<width>x<height> origin:(0,0)
 degree:0"
```

예시:

```
RES="2560x1600"
displayplacer "id:69784AF1-CD7D-B79B-E5D4-60D937407F68 res:${RES} scaling:off
 origin:(0,0) degree:0"
```

Amazon EC2 인스턴스에서 운영 체제 및 소프트웨어 업데이트

다음 주제에서는 Apple 실리콘 Mac 인스턴스(Mac2, Mac2-m1ultra, Mac2-m2, Mac2-m2pro)와 x86 Mac 인스턴스(Mac1)에서 운영 체제 및 소프트웨어를 업데이트하는 방법을 설명합니다.

> **Warning**
>
> 베타 또는 평가판 macOS 버전 설치는 Apple Silicon Mac 인스턴스에서만 사용할 수 있습니다. Amazon EC2는 베타 또는 평가판 macOS 버전을 지원하지 않으며 사전 프로덕션 macOS 버전으로 업데이트 후에도 인스턴스가 계속 작동할 것을 보장하지 않습니다.

Amazon EC2 x86 Mac 인스턴스에 베타 또는 평가판 macOS 버전을 설치하려고 하면 인스턴스를 중지하거나 종료할 때 Amazon EC2 Mac 전용 호스트의 성능이 저하되고 해당 호스트에서 새 인스턴스를 시작하거나 실행하지 못하게 됩니다.

Note

AWS가 공식 AMI를 릴리스하기 전에 인플레이스 macOS 업데이트를 수행하는 경우 선택한 호스트에만 업데이트가 적용됩니다. 다른 호스트가 있거나 새 호스트를 시작하는 경우 해당 호스트에서도 동일한 업데이트 프로세스를 수행해야 합니다. 각 macOS 버전은 기본 Apple Mac 하드웨어에 최소 펌웨어 버전이 있어야 합니다. 인플레이스 업데이트는 선택한 호스트의 펌웨어만 업데이트하고 다른 기존 호스트나 새 호스트로 전송되지 않습니다. Amazon EC2 Mac 전용 호스트와 호환되는 macOS 버전을 확인하려면 Amazon EC2 Mac 전용 호스트에 지원되는 macOS 버전 찾기를 참조하세요.

Apple Silicon Mac 인스턴스에서 소프트웨어 업데이트

사전 조건

네트워크 드라이버 구성 업데이트로 인해 ENA 드라이버 버전 1.0.2는 macOS 13.3 이상과 호환되지 않습니다. 베타, 평가판 또는 프로덕션 macOS 버전 13.3 이상을 설치하고 최신 ENA 드라이버를 설치하지 않은 경우 다음 절차에 따라 새 버전의 드라이버를 설치하세요.

ENA 드라이버의 새 버전을 설치하려면

1. 터미널 창에서 SSH를 사용하여 Apple Silicon Mac 인스턴스에 연결합니다.
2. 다음 명령을 사용하여 Homebrew를 업데이트하고 ENA 애플리케이션을 `Applications` 파일에 다운로드합니다.

```
[ec2-user ~]$ brew update
```

```
[ec2-user ~]$ brew install amazon-ena-ethernet-dext
```

3. exit를 입력하고 Return 키를 눌러 인스턴스와의 연결을 끊습니다.
4. VNC 클라이언트를 사용하여 ENA 애플리케이션을 활성화합니다.

a. 인스턴스의 GUI(그래픽 사용자 인터페이스)에 연결을(를) 사용하여 VNC 클라이언트를 설정합니다.

b. 화면 공유 애플리케이션을 사용하여 인스턴스에 연결했다면 Applications 폴더로 이동하여 ENA 애플리케이션을 엽니다.

c. 활성화를 선택합니다.

d. 드라이버가 제대로 활성화되었는지 확인하려면 터미널 창에서 다음 명령을 실행합니다. 명령의 출력은 이전 드라이버가 종료 상태이고 새 드라이버가 활성화 상태임을 나타냅니다.

```
systemextensionsctl list;
```

e. 인스턴스를 다시 시작한 후에는 새 드라이버만 표시됩니다.

소프트웨어 업데이트 수행

Apple Silicon Mac 인스턴스에서는 여러 단계를 완료하여 현재 위치 운영 체제 업데이트를 수행해야 합니다. 여기에는 Amazon EBS 루트 볼륨의 소유권을 EBS 루트 볼륨 관리 사용자에게 위임하는 것이 포함됩니다. Amazon EC2 API를 사용하여 이를 자동으로 수행하거나 인스턴스에서 명령을 실행하여 수동으로 수행하도록 선택할 수 있습니다.

Automated volume ownership delegation (Recommended)

고려 사항

- 볼륨 소유권 위임 작업을 완료하는 데 30~90분이 소요될 수 있습니다. 이 시간 동안에는 인스턴스에 연결할 수 없습니다.
- 다음 macOS 버전이 지원됩니다.
  - Mac2 | Mac2-m1ultra - macOS Ventura(버전 13.0 이상)
  - Mac2-m2 | Mac2-m2pro - macOS Ventura(버전 13.2 이상)
- 인스턴스에는 부팅 가능한 볼륨이 하나만 있어야 하며, 연결된 각 볼륨에는 추가 관리 사용자가 1명만 있을 수 있습니다.

1단계: EBS 루트 볼륨 관리 사용자의 암호를 설정하고 보안 토큰 활성화

Amazon EBS 루트 볼륨 관리 사용자(`ec2-user`)의 암호를 설정하고 보안 토큰을 활성화해야 합니다.

> **Note**
>
> 암호와 보안 토큰은 GUI를 사용하여 Apple 실리콘 Mac 인스턴스에 처음 연결할 때 설정됩니다. 이전에 GUI를 사용하여 인스턴스에 연결한 경우 이 단계를 수행할 필요가 없습니다.

1. SSH를 이용해 인스턴스에 연결합니다.
2. `ec2-user` 사용자의 암호를 설정합니다.

```
$ sudo /usr/bin/dscl . -passwd /Users/ec2-user
```

3. `ec2-user` 사용자의 보안 토큰을 활성화합니다. `-oldPassword`의 경우 이전 단계와 동일한 암호를 지정합니다. `-newPassword`의 경우 다른 암호를 지정합니다. 다음 명령은 `.txt` 파일에 이전 암호와 새 암호가 저장되어 있다고 가정합니다.

```
$ sysadminctl -oldPassword `cat old_password.txt` -newPassword `cat
 new_password.txt`
```

4. 보안 토큰이 활성화되어 있는지 확인합니다.

```
$ sysadminctl -secureTokenStatus ec2-user
```

2단계: Amazon EBS 루트 볼륨의 소유권을 EBS 루트 볼륨 관리 사용자에게 위임

소유권을 위임하려면 볼륨 소유권 위임 작업을 생성해야 합니다.

1. create-delegate-mac-volume-ownership-task 명령을 사용하여 작업을 생성합니다. `--instance-id`의 경우 인스턴스의 ID를 지정합니다. `--mac-credentials`의 경우 다음 자격 증명을 지정합니다.
   - 내부 디스크 관리 사용자
     - 사용자 이름 - 기본 관리 사용자(`aws-managed-user`)만 지원되고 기본적으로 사용됩니다. 다른 관리 사용자는 지정할 수 없습니다.
     - 암호 - `aws-managed-user`의 기본 암호를 변경하지 않은 경우 공백인 기본 암호를 지정합니다. 그렇지 않으면 암호를 지정합니다.
   - Amazon EBS 루트 볼륨 관리 사용자

- 사용자 이름 - 기본 관리 사용자를 변경하지 않은 경우 ec2-user를 지정합니다. 그렇지 않으면 관리 사용자의 사용자 이름을 지정합니다.
- 암호 - 위의 1단계에서 루트 볼륨 관리자에 설정한 암호를 지정합니다.

```
aws ec2 create-delegate-mac-volume-ownership-task \
--instance-id i-1234567890abcdef0 \
--mac-credentials file://mac-credentials.json
```

다음은 이전 예제에서 참조되는 mac-credentials.json 파일의 내용입니다.

```
{
  "internalDiskPassword":"internal-disk-admin_password",
  "rootVolumeUsername":"root-volume-admin_username",
  "rootVolumepassword":"root-volume-admin_password"
}
```

2. 볼륨 소유권 위임 작업이 완료되고 인스턴스가 정상 상태로 돌아갈 때까지 기다립니다. describe-mac-modification-tasks 명령을 사용합니다. --mac-modification-task-id의 경우 이전 단계의 볼륨 소유권 위임 작업 ID를 지정합니다.

```
aws ec2 describe-mac-modification-tasks \
--mac-modification-task-id task-id
```

3. 볼륨 소유권 위임 작업이 완료되면 3단계로 진행합니다.

3단계: 소프트웨어 업데이트

Amazon EBS 루트 볼륨의 소유권을 위임한 후 x86 Mac 인스턴스에서 소프트웨어 업데이트(아래)에 설명된 단계에 따라 소프트웨어를 업데이트합니다.

Manual volume ownership delegation

이 절차를 진행하면서 두 개의 암호를 생성합니다. 한 암호는 Amazon EBS 루트 볼륨 관리 사용자(ec2-user)용이고 다른 암호는 내부 디스크 관리 사용자(aws-managed-user)용입니다. 절차를 진행하면서 비밀번호를 사용하게 되므로 이 비밀번호를 기억해 두세요.

> **Note**
>
> macOS Big Sur에서 이 절차를 사용하면 macOS Big Sur 11.7.3에서 macOS Big Sur 11.7.4로 업데이트하는 것과 같은 사소한 업데이트만 수행할 수 있습니다. macOS Monterey 이상의 경우 주요 소프트웨어 업데이트를 수행할 수 있습니다.

내부 디스크에 액세스하려면

1. 로컬 컴퓨터의 터미널에서 다음 명령으로 SSH를 사용하여 Apple Silicon Mac 인스턴스에 연결합니다. 자세한 내용은 SSH를 사용하여 인스턴스에 연결 섹션을 참조하세요.

```
ssh -i /path/key-pair-name.pem ec2-user@instance-public-dns-name
```

2. 다음 명령을 사용하여 macOS 화면 공유를 설치하고 시작합니다.

```
[ec2-user ~]$ sudo launchctl enable system/com.apple.screensharing
sudo launchctl load -w /System/Library/LaunchDaemons/
com.apple.screensharing.plist
```

3. 다음 명령을 실행하여 `ec2-user`에 대한 암호를 설정합니다. 암호는 나중에 사용할 것이므로 기억해 두세요.

```
[ec2-user ~]$ sudo /usr/bin/dscl . -passwd /Users/ec2-user
```

4. exit를 입력하고 Return 키를 눌러 인스턴스와의 연결을 끊습니다.
5. 로컬 컴퓨터의 터미널에서 다음 명령으로 VNC 포트에 대한 SSH 터널을 사용하여 인스턴스에 다시 연결합니다.

```
ssh -i /path/key-pair-name.pem -L 5900:localhost:5900 ec2-user@instance-public-
dns-name
```

> **Note**
>
> 다음 VNC 연결 및 GUI 단계가 완료될 때까지 이 SSH 세션을 종료하지 마세요. 인스턴스가 다시 시작되면 연결이 자동으로 닫힙니다.

6. 로컬 컴퓨터에서 다음 단계에 따라 `localhost:5900`에 연결합니다.

a. Finder를 열고 실행을 선택합니다.

b. 서버에 연결을 선택합니다.

c. 서버 주소 필드에 `vnc://localhost:5900`을 입력합니다.

7. macOS 창에서 3단계에서 생성한 암호를 사용하여 `ec2-user`로 Apple Silicon Mac 인스턴스의 원격 세션에 연결합니다.

8. 다음 옵션 중 하나를 사용하여 InternalDisk라는 내부 디스크에 액세스합니다.

   a. macOS Ventura 이상인 경우: 시스템 설정을 열고 왼쪽 창에서 일반을 선택한 다음 창의 오른쪽 하단에서 시동 디스크를 선택합니다.

   b. macOS Monterey 이하인 경우: 시스템 환경 설정을 열고 시동 디스크를 선택한 다음 창의 왼쪽 아래에 있는 잠금 아이콘을 선택하여 창을 잠금 해제합니다.

> **문제 해결 도움말**
>
> 내부 디스크를 마운트해야 하는 경우 터미널에서 다음 명령을 실행합니다.
>
> ```
> APFSVolumeName="InternalDisk" ; SSDContainer=$(diskutil list | grep
>  "Physical Store disk0" -B 3 | grep "/dev/disk" | awk {'print $1'} ) ;
>  diskutil apfs addVolume $SSDContainer APFS $APFSVolumeName
> ```

9. InternalDisk라는 내부 디스크를 선택하고 다시 시작을 선택합니다. 메시지가 표시되면 다시 시작을 선택합니다.

> **Important**
>
> 내부 디스크의 이름이 InternalDisk가 아닌 Macintosh HD인 경우 전용 호스트를 업데이트할 수 있도록 인스턴스를 중지했다가 다시 시작해야 합니다. 자세한 내용은 Amazon EC2 Mac 인스턴스를 중지 또는 종료 섹션을 참조하세요.

다음 절차에 따라 관리자에게 소유권을 위임합니다. SSH로 인스턴스에 다시 연결하면 특수 관리자 사용자(`aws-managed-user`)를 사용하여 내부 디스크에서 부팅합니다. `aws-managed-user`의 초기 암호는 비어 있으므로 처음 연결할 때 암호를 덮어써야 합니다. 그런 다음 부팅 볼륨이 변경되었으므로 macOS 화면 공유를 설치하고 시작하는 단계를 반복해야 합니다.

Amazon EBS 볼륨의 소유권을 관리자에게 위임하려면

1. 로컬 컴퓨터의 터미널에서 다음 명령을 사용하여 Apple Silicon Mac 인스턴스에 연결합니다.

```
ssh -i /path/key-pair-name.pem aws-managed-user@instance-public-dns-name
```

2. WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED! 경고가 표시되면 다음 명령 중 하나를 사용하여 이 문제를 해결합니다.

   a. 다음 명령을 사용하여 알려진 호스트를 지웁니다. 그런 다음 이전 단계를 반복합니다.

```
rm ~/.ssh/known_hosts
```

   b. 이전 단계의 SSH 명령에 다음을 추가합니다.

```
-o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no
```

3. 다음 명령을 실행하여 aws-managed-user에 대한 암호를 설정합니다. aws-managed-user 초기 암호는 비어 있으므로 처음 연결할 때 암호를 덮어써야 합니다.

   a.
```
[aws-managed-user ~]$ sudo /usr/bin/dscl . -passwd /Users/aws-managed-user password
```

   b. 프롬프트 Permission denied. Please enter user's old password:가 표시되면 Enter 키를 누릅니다.

> 문제 해결 도움말
>
> passwd: DS error: eDSAuthFailed 오류가 발생하면 다음 명령을 사용합니다.
>
> ```
> [aws-managed-user ~]$ sudo passwd aws-managed-user
> ```

4. 다음 명령을 사용하여 macOS 화면 공유를 설치하고 시작합니다.

```
[aws-managed-user ~]$ sudo launchctl enable system/com.apple.screensharing
sudo launchctl load -w /System/Library/LaunchDaemons/
com.apple.screensharing.plist
```

5. exit를 입력하고 Return 키를 눌러 인스턴스와의 연결을 끊습니다.

6. 로컬 컴퓨터의 터미널에서 다음 명령으로 VNC 포트에 대한 SSH 터널을 사용하여 인스턴스에 다시 연결합니다.

```
ssh -i /path/key-pair-name.pem -L 5900:localhost:5900 aws-managed-user@instance-public-dns-name
```

7. 로컬 컴퓨터에서 다음 단계에 따라 `localhost:5900`에 연결합니다.

   a. Finder를 열고 실행을 선택합니다.

   b. 서버에 연결을 선택합니다.

   c. 서버 주소 필드에 `vnc://localhost:5900`을 입력합니다.

8. macOS 창에서 3단계에서 생성한 암호를 사용하여 `aws-managed-user`로 Apple Silicon Mac 인스턴스의 원격 세션에 연결합니다.

> **Note**
>
> Apple ID로 로그인하라는 메시지가 표시되면 나중에 설정을 선택합니다.

9. 다음 옵션 중 하나를 사용하여 Amazon EBS 볼륨에 액세스합니다.

   a. macOS Ventura 이상인 경우: 시스템 설정을 열고 왼쪽 창에서 일반을 선택한 다음 창의 오른쪽 하단에서 시동 디스크를 선택합니다.

   b. macOS Monterey 이하인 경우: 시스템 환경 설정을 열고 시동 디스크를 선택한 다음 창의 왼쪽 아래에 있는 잠금 아이콘을 사용하여 창을 잠금 해제합니다.

> **Note**
>
> 재부팅할 때까지 관리자 암호를 입력하라는 메시지가 표시되면 위에서 `aws-managed-user`에 대해 설정한 암호를 사용하세요. 이 암호는 `ec2-user` 또는 인스턴스의 기본 관리자 계정에 대해 설정한 암호와 다를 수 있습니다. 다음 지침은 인스턴스의 관리자 암호를 사용할 시기를 지정합니다.

10. Amazon EBS 볼륨(시동 디스크 창에서 이름이 InternalDisk로 지정되지 않은 볼륨)을 선택하고 다시 시작을 선택합니다.

> **Note**
>
> Apple Silicon Mac 인스턴스에 부팅 가능한 Amazon EBS 볼륨이 여러 개 연결되어 있는 경우 각 볼륨에 고유한 이름을 사용해야 합니다.

11. 다시 시작을 확인한 다음 메시지가 표시되면 사용자 인증을 선택합니다.
12. 이 볼륨의 사용자 권한 부여 창에서 관리자 사용자(기본값 `ec2-user`)가 선택되어 있는지 확인한 다음 권한 부여를 선택합니다.
13. 이전 절차의 3단계에서 생성한 `ec2-user` 암호를 입력한 다음 계속을 선택합니다.
14. 메시지가 표시되면 특수 관리자 사용자(`aws-managed-user`)의 암호를 입력합니다.
15. 로컬 컴퓨터의 터미널에서 사용자 이름 `ec2-user`와 함께 SSH를 사용하여 인스턴스에 다시 연결합니다.

> **문제 해결 도움말**
>
> `WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!` 경고가 표시되면 다음 명령을 실행하고 SSH를 사용하여 인스턴스에 다시 연결합니다.
>
> ```
> rm ~/.ssh/known_hosts
> ```

16. 소프트웨어 업데이트를 수행하려면 x86 Mac 인스턴스에서 소프트웨어 업데이트 아래 명령을 사용하세요.

x86 Mac 인스턴스에서 소프트웨어 업데이트

x86 Mac 인스턴스에서 `softwareupdate` 명령을 사용하여 Apple에서 운영 체제 업데이트를 설치할 수 있습니다.

x86 Mac 인스턴스에 Apple의 운영 체제 업데이트를 설치하려면

1. 다음 명령을 사용하여 사용 가능한 업데이트가 포함된 패키지를 나열합니다.

```
[ec2-user ~]$ softwareupdate --list
```

2. 모든 업데이트를 설치하거나 특정 업데이트만 설치합니다. 특정 업데이트를 설치하려면 다음 명령을 사용합니다.

```
[ec2-user ~]$ sudo softwareupdate --install label
```

대신 모든 업데이트를 설치하려면 다음 명령을 사용합니다.

```
[ec2-user ~]$ sudo softwareupdate --install --all --restart
```

시스템 관리자는 AWS Systems Manager를 사용하여 x86 Mac 인스턴스에서 사전 승인된 운영 체제 업데이트를 롤아웃할 수 있습니다. 자세한 내용은 AWS Systems Manager 사용 설명서를 참조하세요.

Homebrew를 통해 EC2 macOS AMI에 패키지에 대한 업데이트를 설치하여 인스턴스에 이러한 패키지의 최신 버전을 보유할 수 있습니다. Homebrew를 사용하여 Amazon EC2 macOS에서 일반적인 macOS 애플리케이션을 설치하고 실행할 수도 있습니다. 자세한 내용은 Homebrew 설명서를 참조하세요.

Homebrew를 사용하여 업데이트를 설치하려면

1. 다음 명령을 사용하여 Homebrew를 업데이트합니다.

   ```
   [ec2-user ~]$ brew update
   ```

2. 다음 명령을 사용하여 사용 가능한 업데이트가 포함된 패키지를 나열합니다.

   ```
   [ec2-user ~]$ brew outdated
   ```

3. 모든 업데이트를 설치하거나 특정 업데이트만 설치합니다. 특정 업데이트를 설치하려면 다음 명령을 사용합니다.

   ```
   [ec2-user ~]$ brew upgrade package name
   ```

   대신 모든 업데이트를 설치하려면 다음 명령을 사용합니다.

   ```
   [ec2-user ~]$ brew upgrade
   ```

Mac 인스턴스에서 EBS 볼륨 크기 늘리기

Mac 인스턴스에서 Amazon EBS 볼륨 크기를 늘릴 수 있습니다. 자세한 내용은 Amazon EBS 사용 설명서의 Amazon EBS Elastic Volumes를 참조하세요.

볼륨의 크기를 늘린 후에는 다음과 같이 APFS 컨테이너의 크기를 늘려야 합니다.

가용 디스크 공간을 늘리기

1. 다시 시작이 필요한지 확인합니다. 실행 중인 Mac 인스턴스에서 기존 EBS 볼륨의 크기를 변경한 경우 인스턴스를 재부팅해야 새 크기를 사용할 수 있습니다. 시작 시간 중에 디스크 공간을 수정한 경우에는 재부팅할 필요가 없습니다.

   디스크 크기의 현재 상태 보기:

```
[ec2-user ~]$  diskutil list external physical
/dev/disk0 (external, physical):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:                 GUID_partition_scheme            *322.1 GB     disk0
   1:                 EFI EFI                           209.7 MB     disk0s1
   2:                 Apple_APFS Container disk2        321.9 GB     disk0s2
```

2. 다음 명령을 복사하여 붙여넣습니다.

```
[ec2-user ~]$ PDISK=$(diskutil list physical external | head -n1 | cut -d" " -f1)
APFSCONT=$(diskutil list physical external | grep "Apple_APFS" | tr -s " " | cut -
d" " -f8)
yes | sudo diskutil repairDisk $PDISK
```

3. 다음 명령을 복사하여 붙여넣습니다.

```
[ec2-user ~]$ sudo diskutil apfs resizeContainer $APFSCONT 0
```

Amazon EC2 Mac 인스턴스를 중지 또는 종료

Mac 인스턴스를 중지하면 `stopping` 상태가 약 15분 동안 유지된 다음 `stopped` 상태가 됩니다.

Mac 인스턴스를 중지하거나 종료하면 Amazon EC2는 기본 전용 호스트에서 스크러빙 워크플로를 수행하여 내부 SSD를 지우고 영구 NVRAM 변수를 지우고 최신 디바이스 펌웨어로 업데이트합니다. 이렇게 하면 Mac 인스턴스가 다른 EC2 Nitro 인스턴스와 동일한 보안 및 데이터 프라이버시를 제공할 수 있습니다. 또한 최신 macOS AMI를 실행할 수 있습니다. 스크러빙 워크플로 중에 전용 호스트는 일시적으로 보류 상태가 됩니다. x86 Mac 인스턴스에서 스크러빙 워크플로를 완료하는 데 최대 50분이 걸릴 수 있습니다. Apple 실리콘 Mac 인스턴스에서 스크러빙 워크플로를 완료하는 데 최대 110분이 걸릴 수 있습니다. 또한 x86 Mac 인스턴스에서 디바이스 펌웨어를 업데이트해야 하는 경우 스크러빙 워크플로를 완료하는 데 최대 3시간이 걸릴 수 있습니다.

스크러빙 워크플로가 완료된 후 전용 호스트이(가) `available` 상태가 될 때까지 중지된 Mac 인스턴스를 시작하거나 새 Mac 인스턴스를 시작할 수 없습니다.

전용 호스트가 `pending` 상태가 되면 계량 및 청구가 일시 중지됩니다. 스크러빙 워크플로 시간에 대해서는 요금이 청구되지 않습니다.

Mac 인스턴스에 대해 전용 호스트를 릴리스합니다.

Mac 인스턴스 사용을 마친 후에는 전용 호스트를 릴리스할 수 있습니다. 전용 호스트를 릴리스하려면 먼저 Mac 인스턴스를 중지하거나 종료해야 합니다. 할당 기간이 최소값인 24시간을 초과하기 전까지는 호스트를 릴리스할 수 없습니다.

전용 호스트를 릴리스하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 [인스턴스 상태(Instance state)]를 선택한 다음 [인스턴스 중지(Stop instance)] 또는 [인스턴스 종료(Terminate instance)]를 선택합니다.
4. 탐색 창에서 전용 호스트를 선택합니다.
5. 전용 호스트를 선택하고 [작업(Actions)], [호스트 릴리스(Release host)]를 차례로 선택합니다.
6. 확인 메시지가 나타나면 릴리스(Release)]를 선택합니다.

Amazon EC2 Mac 인스턴스의 시스템 무결성 보호 구성

x86 Mac 인스턴스와 Apple 실리콘 Mac 인스턴스의 시스템 무결성 보호(SIP) 설정을 구성할 수 있습니다. SIP는 무단 코드 실행 및 시스템 수준 수정을 방지하는 데 도움이 되는 중요한 macOS 보안 기능입니다. 자세한 내용은 시스템 무결성 보호 정보를 참조하세요.

SIP를 완전히 활성화 또는 비활성화하거나 특정 SIP 설정을 선택적으로 활성화 또는 비활성화할 수 있습니다. SIP를 일시적으로만 비활성화하여 필요한 작업을 수행한 다음 가능한 한 빨리 다시 활성화하는 것이 좋습니다. SIP를 비활성화한 상태로 두면 인스턴스가 악성 코드에 취약해질 수 있습니다.

SIP 구성은 Amazon EC2 Mac 인스턴스가 지원되는 모든 AWS 리전에서 지원됩니다.

주제

- 고려 사항
- 기본 SIP 구성

- SIP 구성 확인
- Apple 실리콘 Mac 인스턴스의 사전 조건
- SIP 설정 구성
- SIP 구성 작업 상태 확인

고려 사항

- 다음 Amazon EC2 Mac 인스턴스 유형과 macOS 버전이 지원됩니다.
  - Mac1 | Mac2 | Mac2-m1ultra - macOS Ventura(버전 13.0 이상)
  - Mac2-m2 | Mac2-m2pro - macOS Ventura(버전 13.2 이상)

  > **Note**
  >
  > macOS의 베타 및 미리 보기 버전은 지원되지 않습니다.

- 사용자 지정 SIP 구성을 지정하여 개별 SIP 설정을 선택적으로 활성화하거나 비활성화할 수 있습니다. 사용자 지정 구성을 구현하는 경우 인스턴스에 연결하고 설정을 확인하여 요구 사항이 제대로 구현되고 의도한 대로 작동하는지 확인합니다.

  SIP 구성은 macOS 업데이트에 따라 변경될 수 있습니다. macOS 버전 업그레이드 후 사용자 지정 SIP 설정을 검토하여 보안 구성의 지속적인 호환성과 적절한 기능을 보장하는 것이 좋습니다.
- x86 Mac 인스턴스의 경우 SIP 설정은 인스턴스 수준에서 적용됩니다. 인스턴스에 연결된 모든 루트 볼륨은 구성된 SIP 설정을 자동으로 상속합니다.

  Apple 실리콘 Mac 인스턴스의 경우 SIP 설정은 볼륨 수준에서 적용됩니다. 인스턴스에 연결된 루트 볼륨은 SIP 설정을 상속하지 않습니다. 다른 루트 볼륨을 연결하는 경우 SIP 설정을 필요한 상태로 재구성해야 합니다.
- SIP 구성 작업을 완료하는 데 최대 90분이 소요될 수 있습니다. SIP 구성 작업이 진행되는 동안에는 인스턴스에 연결할 수 없습니다.
- SIP 구성은 이후에 인스턴스에서 생성한 스냅샷 또는 AMI로 전송되지 않습니다.
- Apple 실리콘 Mac 인스턴스에는 부팅 가능한 볼륨이 하나만 있어야 하며 연결된 각 볼륨에는 추가 관리 사용자가 1명만 있을 수 있습니다.

기본 SIP 구성

다음 표에는 x86 Mac 인스턴스와 Apple 실리콘 Mac 인스턴스의 기본 SIP 구성이 나열되어 있습니다.

	Apple 실리콘 Mac 인스턴스	x86 Mac 인스턴스
Apple 내부	활성화됨	비활성화됨
파일 시스템 보호	활성화됨	비활성화됨
기본 시스템	활성화됨	활성화됨
디버깅 제한 사항	활성화됨	활성화됨
Dtrace 제한 사항	활성화됨	활성화됨
Kext 서명	활성화됨	활성화됨
Nvram 보호	활성화됨	활성화됨

SIP 구성 확인

변경 전후에 SIP 구성을 확인하여 예상대로 구성되었는지 확인하는 것이 좋습니다.

Amazon EC2 Mac 인스턴스의 SIP 구성을 확인하려면

SSH를 사용하여 인스턴스에 연결한 다음 명령줄에서 다음 명령을 실행합니다.

```
$ csrutil status
```

출력의 예시는 다음과 같습니다.

```
System Integrity Protection status: enabled.

Configuration:
    Apple Internal: enabled
```

```
    Kext Signing: disabled
    Filesystem Protections: enabled
    Debugging Restrictions: enabled
    DTrace Restrictions: enabled
    NVRAM Protections: enabled
    BaseSystem Verification: disabled
```

Apple 실리콘 Mac 인스턴스의 사전 조건

Apple 실리콘 Mac 인스턴스의 SIP 설정을 구성하려면 먼저 Amazon EBS 루트 볼륨 관리 사용자(`ec2-user`)의 암호를 설정하고 보안 토큰을 활성화해야 합니다.

> **Note**
>
> 암호와 보안 토큰은 GUI를 사용하여 Apple 실리콘 Mac 인스턴스에 처음 연결할 때 설정됩니다. 이전에 GUI를 사용하여 인스턴스에 연결했거나 x86 Mac 인스턴스를 사용하는 경우 이 단계를 수행할 필요가 없습니다.

EBS 루트 볼륨 관리 사용자의 암호를 설정하고 보안 토큰을 활성화하려면

1. SSH를 사용하여 인스턴스에 연결합니다.
2. `ec2-user` 사용자의 암호를 설정합니다.

   ```
   $ sudo /usr/bin/dscl . -passwd /Users/ec2-user
   ```

3. `ec2-user` 사용자의 보안 토큰을 활성화합니다. `-oldPassword`의 경우 이전 단계와 동일한 암호를 지정합니다. `-newPassword`의 경우 다른 암호를 지정합니다. 다음 명령은 `.txt` 파일에 이전 암호와 새 암호가 저장되어 있다고 가정합니다.

   ```
   $ sysadminctl -oldPassword `cat old_password.txt` -newPassword `cat
    new_password.txt`
   ```

4. 보안 토큰이 활성화되어 있는지 확인합니다.

   ```
   $ sysadminctl -secureTokenStatus ec2-user
   ```

SIP 설정 구성

인스턴스의 SIP 설정을 구성할 때 모든 SIP 설정을 활성화 또는 비활성화하거나, 특정 SIP 설정을 선택적으로 활성화 또는 비활성화하는 사용자 지정 구성을 지정할 수 있습니다.

> **Note**
>
> 사용자 지정 구성을 구현하는 경우 인스턴스에 연결하고 설정을 확인하여 요구 사항이 제대로 구현되고 의도한 대로 작동하는지 확인합니다.
> SIP 구성은 macOS 업데이트에 따라 변경될 수 있습니다. macOS 버전 업그레이드 후 사용자 지정 SIP 설정을 검토하여 보안 구성의 지속적인 호환성과 적절한 기능을 보장하는 것이 좋습니다.

인스턴스의 SIP 설정을 구성하려면 SIP 구성 작업을 생성해야 합니다. SIP 구성 작업은 인스턴스의 SIP 설정을 지정합니다.

Apple 실리콘 Mac 인스턴스의 SIP 구성을 생성할 때 다음 자격 증명을 지정해야 합니다.

- 내부 디스크 관리 사용자
  - 사용자 이름 - 기본 관리 사용자(`aws-managed-user`)만 지원되고 기본적으로 사용됩니다. 다른 관리 사용자는 지정할 수 없습니다.
  - 암호 - `aws-managed-user`의 기본 암호를 변경하지 않은 경우 공백인 기본 암호를 지정합니다. 그렇지 않으면 암호를 지정합니다.
- Amazon EBS 루트 볼륨 관리 사용자
  - 사용자 이름 - 기본 관리 사용자를 변경하지 않은 경우 `ec2-user`를 지정합니다. 그렇지 않으면 관리 사용자의 사용자 이름을 지정합니다.
  - 암호 - 항상 암호를 지정해야 합니다.

다음 방법을 사용하여 SIP 구성 작업을 생성합니다.

Console

콘솔을 사용하여 SIP 구성 작업을 생성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택한 다음 Amazon EC2 Mac 인스턴스를 선택합니다.
3. 보안 탭에서 Mac 수정, 시스템 무결성 보호 수정을 선택합니다.

4. 모든 SIP 설정을 활성화하려면 SIP 활성화를 선택합니다. 모든 SIP 설정을 비활성화하려면 SIP 활성화를 선택 취소합니다.
5. 특정 SIP 설정을 선택적으로 활성화하거나 비활성화하는 사용자 지정 구성을 지정하려면 사용자 지정 SIP 구성 지정을 선택한 다음 활성화할 SIP 설정을 선택하거나 비활성화할 SIP 설정을 선택 취소합니다.
6. 루트 볼륨 사용자와 내부 디스크 소유자의 자격 증명을 지정합니다.
7. SIP 수정 작업 생성을 선택합니다.

AWS CLI

AWS CLI를 사용하여 SIP 구성 작업을 생성하려면

create-mac-system-integrity-protection-modification-task 명령을 사용합니다.

모든 SIP 설정 활성화 또는 비활성화

모든 SIP 설정을 완전히 활성화하거나 비활성화하려면 `--mac-system-integrity-protection-status` 파라미터만 사용합니다.

다음 예제 명령은 모든 SIP 설정을 활성화합니다.

```
aws ec2 create-mac-system-integrity-protection-modification-task \
--instance-id i-0abcdef9876543210 \
--mac-system-integrity-protection-status enabled \
--mac-credentials file://mac-credentials.json
```

사용자 지정 SIP 구성 지정

특정 SIP 설정을 선택적으로 활성화하거나 비활성화하는 사용자 지정 SIP 구성을 지정하려면 `--mac-system-integrity-protection-status` 및 `--mac-system-integrity-protection-configuration` 파라미터를 지정합니다. 이 경우 `mac-system-integrity-protection-status`를 사용하여 전체 SIP 상태를 지정하고 `mac-system-integrity-protection-configuration`을 사용하여 개별 SIP 설정을 선택적으로 활성화 또는 비활성화합니다.

다음 예제 명령은 `NvramProtections` 및 `FilesystemProtections`를 제외한 모든 SIP 설정을 활성화하는 SIP 구성 작업을 생성합니다.

```
aws ec2 create-mac-system-integrity-protection-modification-task \
```

```
--instance-id i-0abcdef9876543210 \
--mac-system-integrity-protection-status enabled \
--mac-system-integrity-protection-configuration
 "NvramProtections=disabled, FilesystemProtections=disabled" \
--mac-credentials file://mac-credentials.json
```

다음 예제 명령은 DtraceRestrictions를 제외한 모든 SIP 설정을 비활성화하는 SIP 구성 작업을 생성합니다.

```
aws ec2 create-mac-system-integrity-protection-modification-task \
--instance-id i-0abcdef9876543210 \
--mac-system-integrity-protection-status disabled \
--mac-system-integrity-protection-configuration "DtraceRestrictions=enabled" \
--mac-credentials file://mac-credentials.json
```

mac-credentials.json 파일의 내용

다음은 이전 예제에서 참조되는 mac-credentials.json 파일의 내용입니다.

```
{
  "internalDiskPassword":"internal-disk-admin_password",
  "rootVolumeUsername":"root-volume-admin_username",
  "rootVolumepassword":"root-volume-admin_password"
}
```

SIP 구성 작업 상태 확인

다음 방법 중 하나를 사용하여 SIP 구성 작업의 상태를 확인합니다.

Console

콘솔을 사용하여 SIP 구성 작업을 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택한 다음 Amazon EC2 Mac 인스턴스를 선택합니다.
3. 보안 탭에서 아래로 스크롤하여 Mac 수정 작업 섹션으로 이동합니다.

AWS CLI

AWS CLI를 사용하여 SIP 구성 작업의 상태를 확인하려면

describe-mac-modification-tasks 명령을 사용합니다.

Amazon EC2 Mac 전용 호스트에 대해 지원되는 macOS 버전 찾기

Amazon EC2 Mac 전용 호스트에서 지원하는 최신 macOS 버전을 볼 수 있습니다. 이 기능을 사용하면 전용 호스트가 선호하는 macOS 버전에서 인스턴스 시작을 지원할 수 있는지 검증할 수 있습니다.

각 macOS 버전을 부팅하려면 기본 Apple Mac에 최소 펌웨어 버전이 있어야 합니다. 할당된 Mac 전용 호스트가 장기간 유휴 상태로 유지되거나 장기 실행 중인 인스턴스가 있는 경우 Apple Mac 펌웨어 버전이 오래되었을 수 있습니다.

최신 macOS 버전을 지원하기 위해 할당된 Mac 전용 호스트에서 인스턴스를 중지하거나 종료할 수 있습니다. 그러면 호스트 스크러빙 워크플로가 시작되고 기본 Apple Mac에서 최신 macOS 버전을 지원하도록 펌웨어가 업데이트됩니다. 장기 실행 인스턴스가 있는 전용 호스트는 실행 중인 인스턴스를 중지하거나 종료하면 자동으로 업데이트됩니다.

스크러빙 워크플로에 대한 자세한 내용은 Amazon EC2 Mac 인스턴스를 중지 또는 종료 섹션을 참조하세요.

Mac 인스턴스 시작에 대한 자세한 내용은 AWS Management Console 또는 AWS CLI를 사용하여 Mac 인스턴스 시작 섹션을 참조하세요.

Amazon EC2 콘솔 또는 AWS CLI를 사용하여 할당된 전용 호스트에서 지원되는 최신 macOS 버전에 대한 정보를 볼 수 있습니다.

Console

콘솔을 사용하여 전용 호스트 펌웨어 정보를 보는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 전용 호스트(Dedicated Hosts)를 선택합니다.
3. 전용 호스팅 세부 정보 페이지의 지원되는 최신 macOS 버전에서 호스트가 지원할 수 있는 최신 macOS 버전을 확인할 수 있습니다.

AWS CLI

AWS CLI를 사용하여 전용 호스트 펌웨어 정보를 보는 방법

`describe-mac-hosts` 명령을 사용하고 `region`을 적절한 AWS 리전으로 바꿉니다.

```
$ aws ec2 describe-mac-hosts --region us-east-1
  {
      "MacHosts": [
          {
              "HostId": "h-07879acf49EXAMPLE",
              "MacOSLatestSupportedVersions": [
                  "14.3",
                  "13.6.4",
                  "12.7.3"
              ]
          }
      ]
  }
```

macOS AMI 알림 구독

새 AMI가 배포되거나 BridgeOS가 업데이트될 때 알림을 받으려면 Amazon SNS를 사용하여 알림을 구독합니다.

EC2 macOS AMI에 대한 자세한 내용은 Amazon EC2 macOS AMI 릴리스 정보 섹션을 참조하세요.

macOS AMI 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 변경합니다. 구독을 신청하는 SNS 알림이 이 리전에서 생성되었기 때문에 이 리전을 사용해야 합니다.
3. 탐색 창에서 구독을 선택합니다.
4. Create subscription을 선택합니다.
5. 구독 생성 대화 상자에서 다음과 같이 수행합니다.
   a. 주제 ARN에 다음 Amazon 리소스 이름(ARN) 중 하나를 복사하여 붙여넣습니다.
      - **arn:aws:sns:us-east-1:898855652048:amazon-ec2-macos-ami-updates**
      - **arn:aws:sns:us-east-1:898855652048:amazon-ec2-bridgeos-updates**
   b. 프로토콜에서 다음 중 하나를 선택합니다.
      - 이메일:

엔드포인트에서 알림을 받을 이메일 주소를 입력합니다. 구독을 생성하고 나면 `AWS Notification - Subscription Confirmation`이라는 제목의 확인 메시지를 받게 됩니다. 이메일을 열고 [구독 확인(Confirm subscription)]을 선택하여 구독을 완료합니다.

- SMS:

  [엔드포인트(Endpoint)]에 알림을 받는 데 사용할 수 있는 전화번호를 입력합니다.

- AWS Lambda, Amazon SQS, Amazon Data Firehose(알림은 JSON 형식으로 제공됨):

  [엔드포인트(Endpoint)]에서 알림을 받는 데 사용할 수 있는 Lambda 함수, SQS 대기열 또는 Firehose 스트림의 ARN을 입력합니다.

c. Create subscription을 선택합니다.

macOS AMI가 배포될 때마다, `amazon-ec2-macos-ami-updates` 주제의 구독자에게 알림이 발송됩니다. bridgeOS가 업데이트될 때마다 `amazon-ec2-bridgeos-updates` 주제의 구독자에게 알림이 발송됩니다. 이런 알림을 더 이상 받지 않기를 원하는 경우, 다음 절차를 수행해서 구독을 해제하세요.

macOS AMI 알림을 구독 취소하려면

1. https://console.aws.amazon.com/sns/v3/home에서 Amazon SNS 콘솔을 엽니다.
2. 필요한 경우 탐색 모음에서 리전을 미국 동부(버지니아 북부)로 변경합니다. SNS 알림이 이 리전에 생성되었기 때문에 이 리전을 사용해야 합니다.
3. 탐색 창에서 구독을 선택합니다.
4. 구독을 선택한 후 작업, 구독 삭제를 선택합니다. 확인 메시지가 나타나면 삭제를 선택합니다.

AWS Systems Manager Parameter Store API를 사용하여 macOS AMI ID 검색

인스턴스를 시작할 때 AMI를 지정해야 합니다. AMI는 AWS 리전, 운영 체제, 프로세서 아키텍처에 따라 다릅니다. AWS Systems Manager Parameter Store API를 쿼리하여 AWS 리전의 모든 macOS AMI를 보고 최신 macOS AMI를 검색할 수 있습니다. 이러한 퍼블릭 파라미터를 사용하면 macOS AMI ID를 수동으로 조회할 필요가 없습니다. 퍼블릭 파라미터는 x86 및 ARM64 macOS AMI에서 모두 사용할 수 있으며 기존 AWS CloudFormation 템플릿과 통합할 수 있습니다.

필수 권한

이 작업을 수행하려면 IAM 보안 주체에게 `ssm:GetParameter` API 작업을 직접적으로 호출할 수 있는 권한이 있어야 합니다.

AWS CLI를 사용하여 현재 AWS 리전의 모든 macOS AMI 목록을 보려면

다음 get-parameters-by-path 명령을 사용하여 현재 리전에 있는 모든 macOS AMI의 목록을 볼 수 있습니다.

```
aws ssm get-parameters-by-path --path /aws/service/ec2-macos --recursive --query
 "Parameters[].Name"
```

AWS CLI를 사용하여 최신 메이저 macOS AMI의 AMI ID를 검색하려면

다음 get-parameter 명령을 서브 파라미터 `image_id`와 함께 사용합니다. 다음 예제에서 `sonoma`를 macOS 지원 메이저 버전으로 바꾸고, `x86_64_mac`을 프로세서로 바꾸고, `region-code`를 최신 macOS AMI ID가 필요한 지원되는 AWS 리전으로 바꿉니다.

```
aws ssm get-parameter --name /aws/service/ec2-macos/sonoma/x86_64_mac/latest/image_id
 --region region-code
```

자세한 내용은 AWS Systems Manager 사용 설명서의 Calling AMI public parameters for macOS를 참조하세요.

Amazon EC2 macOS AMI 릴리스 정보

다음 정보에서는 EC2 macOS AMI에 기본적으로 포함된 패키지에 대한 세부 정보를 제공하고 각 EC2 macOS AMI 릴리스의 변경 내용을 요약합니다.

macOS AMI 알림을 구독하는 방법에 대한 자세한 내용은 macOS AMI 알림 구독 섹션을 참조하세요.

Mac 인스턴스는 다음 운영 체제 중 하나를 실행할 수 있습니다.

- macOS Mojave(버전 10.14)(x86 Mac 인스턴스에만 해당)
- macOS Catalina(버전 10.15)(x86 Mac 인스턴스에만 해당)
- macOS Big Sur(버전 11)(x86 및 M1 Mac 인스턴스)
- macOS Monterey(버전 12)(x86 및 M1 Mac 인스턴스)
- macOS Ventura(버전 13)(모든 Mac 인스턴스, M2 및 M2 Pro Mac 인스턴스는 macOS Ventura 버전 13.2 이상 지원)
- macOS Sonoma(버전 14)(모든 Mac 인스턴스)

- macOS Sequoia(버전 15)(모든 Mac 인스턴스)

macOS Sequoia에 대한 로컬 네트워크 개인 정보 보호 정책 승인

macOS Sequoia(버전 15)에는 Amazon EC2 인스턴스 메타데이터 서비스(IMDS)를 비롯한 로컬 IP 기반 서비스 사용자에게 영향을 미치는 새로운 로컬 네트워크 개인 정보 보호 기능이 있습니다.

> **Important**
>
> 로컬 IP 기반 서비스에 중단 없이 액세스할 수 있는지 확인하려면 다음 단계를 사용하여 로컬 네트워크 개인 정보 보호 정책을 승인합니다.

로컬 네트워크 개인 정보 보호 정책을 승인하려면

1. 인스턴스의 GUI(그래픽 사용자 인터페이스)에 연결.
2. 화면의 프롬프트에 따라 로컬 네트워크 개인 정보 보호 정책을 승인합니다.
3. 정책을 승인한 후 EC2 Mac 인스턴스의 AMI를 생성합니다. 자세한 내용은 Amazon EBS 지원 AMI 생성 섹션을 참조하세요.

새로 생성된 AMI에서 시작된 모든 EC2 Mac 인스턴스는 로컬 네트워크 개인 정보 보안 권한을 유지합니다.

Amazon EC2 macOS AMI에 포함된 기본 패키지

다음 테이블에서는 EC2 macOS AMI에 기본적으로 포함되는 패키지를 설명합니다.

패키지	릴리스 정보
EC2 macOS Init	https://github.com/aws/ec2-macos-init/tags
EC2 macOS Utils	https://github.com/aws/ec2-macos-utils/tags
Amazon SSM Agent	https://github.com/aws/amazon-ssm-agent/releases
AWS Command Line Interface(AWS CLI) 버전 2	https://raw.githubusercontent.com/aws/aws-cli/v2/CHANGELOG.rst

패키지	릴리스 정보
Xcode용 명령줄 도구	https://developer.apple.com/documentation/xcode-release-notes
Homebrew	https://github.com/Homebrew/brew/releases
EC2 Instance Connect	https://github.com/aws/aws-ec2-instance-connect-config/releases
Safari	https://developer.apple.com/documentation/safari-release-notes

Amazon EC2 macOS AMI 업데이트

다음 테이블에서는 EC2 macOS AMI 릴리스에 포함된 변경 내용을 설명합니다. 모든 EC2 macOS AMI에 적용되는 변경 내용도 있지만 이러한 AMI 중 서브 세트에만 적용되는 변경 내용도 있습니다.

EC2 macOS AMI 업데이트

릴리스	변경
2025.06.27	모든 AMI • `awscli`가 2.27.40으로 업데이트됨 • Homebrew가 4.5.7로 업데이트됨 • x86_64 이미지가 AWS ENA 이더넷 덱스트로 마이그레이션됨 macOS Ventura 13.7.6 릴리스 • macOS Ventura 13.7.6의 보안 콘텐츠 • Safari가 18.5로 업데이트됨 • Safari 18.5의 보안 콘텐츠 macOS Sonoma 14.7.6 릴리스 • macOS Sonoma 14.7.6의 보안 콘텐츠

릴리스	변경
	• Safari가 18.5로 업데이트됨 • Safari 18.5의 보안 콘텐츠 macOS Sequoia 15.5 릴리스 • macOS Sequoia 15.5의 보안 콘텐츠 • 명령줄 도구가 16.4로 업데이트됨 • AWS ENA 이더넷이 1.0.9로 업데이트됨

2025.05.21	모든 AMI
	
	• `awscli`가 2.27.7로 업데이트됨
	• `ec2-macos-init` 가 1.5.11로 업데이트됨
	• `ec2-macos-utils` 가 1.0.5로 업데이트됨
	• Homebrew가 4.5.1로 업데이트됨
	
	macOS Ventura 13.7.5 릴리스
	
	• macOS Ventura 13.7.5의 보안 콘텐츠
	• Safari가 18.4로 업데이트됨
	  • Safari 18.4의 보안 콘텐츠
	
	macOS Sonoma 14.7.5 릴리스
	
	• macOS Sonoma 14.7.5의 보안 콘텐츠
	• Safari가 18.4로 업데이트됨
	  • Safari 18.4의 보안 콘텐츠
	
	macOS Sequoia 15.4.1 릴리스
	
	• macOS Sequoia 15.4.1의 보안 콘텐츠
	• 명령줄 도구가 16.3으로 업데이트됨

릴리스	변경

2025.05.05	모든 AMI
	• `awscli`가 2.27.1로 업데이트됨
	• `ec2-macos-init` 가 1.5.11로 업데이트됨
	• `ec2-macos-system-monitor` 가 1.3.1로 업데이트됨
	• `ec2-macos-utils` 가 1.0.5로 업데이트됨
	• Homebrew가 4.4.32로 업데이트됨
	
	macOS Ventura 13.7.5 릴리스
	• macOS Ventura 13.7.5의 보안 콘텐츠
	• Safari가 18.4로 업데이트됨
	  • Safari 18.4의 보안 콘텐츠
	
	macOS Sonoma 14.7.5 릴리스
	• macOS Sonoma 14.7.5의 보안 콘텐츠
	• Safari가 18.4로 업데이트됨
	  • Safari 18.4의 보안 콘텐츠
	
	macOS Sequoia 15.4.1 릴리스
	• macOS Sequoia 15.4.1의 보안 콘텐츠

릴리스	변경
2025.03.18	모든 AMI • `awscli`가 2.24.2로 업데이트됨 • Homebrew가 4.4.20으로 업데이트됨 macOS Sequoia 15.3.1 릴리스 • macOS Sequoia 15.3.1의 보안 콘텐츠 macOS Sonoma 14.7.4 릴리스 • macOS Sonoma 14.7.4의 보안 콘텐츠 • Safari가 18.3으로 업데이트됨 macOS Ventura 13.7.4 릴리스 • macOS Ventura 13.7.4의 보안 콘텐츠 • Safari가 18.3으로 업데이트됨

릴리스	변경
2025.01.24	모든 AMI • `awscli`가 2.22.33로 업데이트됨 • Homebrew가 4.4.15로 업데이트됨 macOS Sequoia 15.2 릴리스 • macOS Sequoia 15.2의 보안 콘텐츠 • 명령줄 도구가 16.2로 업데이트됨 macOS Sonoma 14.7.2 릴리스 • macOS Sonoma 14.7.2의 보안 콘텐츠 • Safari가 18.2로 업데이트됨 • 명령줄 도구가 16.2로 업데이트됨 macOS Ventura 13.7.2 릴리스 • macOS Ventura 13.7.2의 보안 콘텐츠 • Safari가 18.2로 업데이트됨

릴리스	변경
2024.12.20	모든 AMI • Homebrew가 4.4.8로 업데이트됨 • `aws-cli`가 2.22.5로 업데이트됨 • `amazon-ssm-agent` 가 3.3.987.0으로 업데이트됨 macOS Sequoia 15.1.1 릴리스 • macOS Sequoia 15.1.1의 보안 콘텐츠 macOS Sonoma 14.7.1 릴리스 • macOS Sonoma 14.7.1의 보안 콘텐츠 • Safari가 18.1.1로 업데이트됨 macOS Ventura 13.7.1 릴리스 • macOS Ventura 13.7.1의 보안 콘텐츠 • Safari가 18.1.1로 업데이트됨

릴리스	변경

2024.10.28	모든 AMI
	• Homebrew가 4.4.2로 업데이트됨
	• `aws-cli`가 2.18.13으로 업데이트됨
	• `amazon-ssm-agent` 가 3.3.987.0으로 업데이트됨
	• `ec2-macos-init` 가 1.5.10으로 업데이트됨
	• `ec2-macos-utils` 가 1.0.4로 업데이트됨
	
	macOS Sequoia 15.0 릴리스
	• macOS Sequoia 15의 보안 콘텐츠
	
	macOS Sonoma 14.7 릴리스
	• macOS Sonoma 14.7의 보안 콘텐츠
	• 명령줄 도구가 16.0으로 업데이트됨
	• Safari가 18.0.1로 업데이트됨
	  • Safari 18의 보안 콘텐츠
	
	macOS Ventura 13.7 릴리스
	• macOS Ventura 13.7의 보안 콘텐츠
	• Safari가 18.0.1로 업데이트됨
	  • Safari 18의 보안 콘텐츠

릴리스	변경

2024.08.20	모든 AMI
	• Homebrew가 4.3.14로 업데이트됨
	• `aws-cli`가 2.17.29로 업데이트됨
	
	macOS Sonoma 14.6.1 릴리스
	• 게시된 CVE 항목 없음
	
	macOS Ventura 13.6.9 릴리스
	• 게시된 CVE 항목 없음
	• Safari가 17.6으로 업데이트됨
	  • Safari 17.6의 보안 콘텐츠
	
	macOS Monterey 12.7.6 릴리스
	• macOS Monterey 12.7.6의 보안 콘텐츠
	• Safari가 17.6으로 업데이트됨
	  • Safari 17.6의 보안 콘텐츠

릴리스	변경

2024.06.07	모든 AMI
	• Homebrew가 4.3.1-1로 업데이트됨
	• `aws-cli`가 2.15.56로 업데이트됨
	• `amazon-ssm-agent` 가 3.3.380.0-1로 업데이트됨
	
	macOS Sonoma 14.5 릴리스
	• macOS Sonoma 14.5의 보안 콘텐츠
	
	macOS Ventura 13.6.7 릴리스
	• macOS Ventura 13.6.7의 보안 콘텐츠
	• Safari를 17.5로 업데이트함
	  • Safari 17.5의 보안 콘텐츠
	
	macOS Monterey 12.7.5 릴리스
	• macOS Monterey 12.7.5의 보안 콘텐츠
	• Safari를 17.5로 업데이트함
	  • Safari 17.5의 보안 콘텐츠

릴리스	변경

2024.04.12	모든 AMI
	• Homebrew가 4.2.16-1로 업데이트됨
	• `aws-cli`가 2.15.36로 업데이트됨
	
	macOS Sonoma 14.4.1 릴리스
	• macOS Sonoma 14.4.1의 보안 콘텐츠
	
	macOS Ventura 13.6.6 릴리스
	• macOS Ventura 13.6.6의 보안 콘텐츠
	• Safari를 17.4.1로 업데이트함
	  • Safari 17.4.1의 보안 콘텐츠
	
	macOS Monterey
	• Safari를 17.4.1로 업데이트함
	  • Safari 17.4.1의 보안 콘텐츠

Amazon EBS 최적화 인스턴스 유형

Amazon EBS 최적화 인스턴스는 최적화된 구성 스택을 사용하며 Amazon EBS I/O를 위한 전용 대역폭을 추가로 제공합니다. 이 최적화는 Amazon EBS I/O와 인스턴스의 다른 트래픽 간의 경합을 최소화하여 EBS 볼륨에 최상의 성능을 제공합니다.

EBS 최적화 인스턴스에 연결된 경우 범용 SSD(`gp2` 및 `gp3`) 볼륨은 지정된 해의 시간 중 프로비저닝된 IOPS 성능 99%의 90% 이상을 제공되도록 설계되며, 프로비저닝된 IOPS SSD(`io1` 및 `io2`) 볼륨은 지정된 해의 시간 중 프로비저닝된 IOPS 성능 99.9%의 90% 이상을 제공되도록 설계됩니다. 처리량 최적화 HDD(`st1`) 및 콜드 HDD(`sc1`)는 지정된 한 해의 시간 중 예상 처리량 성능 99%의 90% 이상을 제공됩니다. 매 시간 총 처리량 목표 99%를 달성하기 위해, 준수하지 않는 기간은 대략적으로 균등하게 분산됩니다. 자세한 내용은 Amazon EBS 사용 설명서의 Amazon EBS volume types를 참조하세요.

일부 인스턴스 유형은 기본적으로 EBS에 최적화되어 있으므로 최적화를 활성화할 필요도 없고 비활성화하려고 해도 효과가 없습니다. 다른 인스턴스 유형은 선택적으로 EBS 최적화를 지원하며 추가 시간당 요금을 지불하면 시작 중 또는 이후에 활성화할 수 있습니다. 일부 인스턴스 유형은 EBS 최적화를 지원하지 않습니다.

자세한 인스턴스 유형 사양 및 기능은 Amazon EC2 인스턴스 유형 가이드를 참조하세요.

고려 사항

- 인스턴스 유형의 EBS 성능은 인스턴스의 성능 제한 또는 연결된 볼륨의 집계된 성능 중 작은 쪽의 제한을 받습니다. 최대한의 EBS 성능을 달성하려기 위해서는 인스턴스에 최대 인스턴스 성능과 같거나 그 이상의 결합된 성능을 제공하는 연결된 볼륨이 있어야 합니다. 예를 들어 `r6i.16xlarge`에 대해 `80,000` IOPS를 달성하려면 인스턴스에 각각 `16,000` IOPS로 프로비저닝된 볼륨이 적어도 5 gp3 이상 있어야 합니다(5 볼륨 x `16,000` IOPS = `80,000` IOPS). 애플리케이션에 필요한 것보다 많은 전용 Amazon EBS 처리량을 제공하는 인스턴스 유형을 선택하는 것이 좋습니다. 그렇게 하지 않으면 Amazon EBS와 Amazon EC2 간의 연결에 성능 제한이 발생할 수 있습니다.
- 인스턴스에 연결할 수 있는 Amazon EBS 볼륨의 최대 수는 인스턴스 유형 및 인스턴스 크기에 따라 달라집니다. 자세한 내용은 Amazon EC2 인스턴스의 Amazon EBS 볼륨 한도 섹션을 참조하세요.
- 최대 IOPS와 처리량 제한은 상호 의존적입니다. I/O 크기에 따라 여러 제한 중 하나에 도달하여 전체 성능에 영향을 미칠 수 있습니다. 최적의 결과를 얻으려면 워크로드를 계획할 때 두 제한을 모두 고려하세요.

기본적으로 EBS 최적화됨

다음 인스턴스 유형은 기본적으로 EBS에 최적화되어 있습니다. EBS 최적화를 활성화할 필요가 없으며 EBS 최적화를 비활성화해도 효과가 없습니다.

인스턴스

- 범용
- 컴퓨팅 최적화
- 메모리 최적화
- 스토리지 최적화
- 액셀러레이티드 컴퓨팅
- 고성능 컴퓨팅

범용

> **Note**
>
> M8g, M8gd 인스턴스 유형은 구성 가능한 대역폭 가중치를 지원합니다. 이러한 인스턴스 유형을 사용하면 네트워킹 성능 또는 Amazon EBS 성능에 맞게 인스턴스의 대역폭을 최적화할 수 있습니다. 다음 표에는 이러한 인스턴스 유형에 대한 기본 Amazon EBS 대역폭 성능이 나와 있습니다. 자세한 내용은 EC2 인스턴스 대역폭 가중치 구성 섹션을 참조하세요.

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
a1.medium [1]	300	3500	37.50	437.50	2500	20000
a1.large [1]	525	3500	65.62	437.50	4000	20000
a1.xlarge [1]	800	3500	100.00	437.50	6000	20000
a1.2xlarge [1]	1750	3500	218.75	437.50	10000	20000
a1.4xlarge [2]	3500		437.5		20000	
a1.metal [2]	3500		437.5		20000	
m4.large [2]	450		56.25		3600	
m4.xlarge [2]	750		93.75		6000	
m4.2xlarge [2]	1000		125.0		8000	
m4.4xlarge [2]	2000		250.0		16000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
m4.10xlarge [2]	4000		500.0		32000	
m4.16xlarge [2]	10000		1250.0		65000	
m5.large [1]	650	4750	81.25	593.75	3600	18750
m5.xlarge [1]	1150	4750	143.75	593.75	6000	18750
m5.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
m5.4xlarge [2]	4750		593.75		18750	
m5.8xlarge [2]	6800		850.0		30000	
m5.12xlarge [2]	9500		1187.5		40000	
m5.16xlarge [2]	13600		1700.0		60000	
m5.24xlarge [2]	19000		2375.0		80000	
m5.metal [2]	19000		2375.0		80000	
m5a.large [1]	650	2880	81.25	360.00	3600	16000
m5a.xlarge [1]	1085	2880	135.62	360.00	6000	16000

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
m5a.2xlarge [1]	1580	2880	197.50	360.00	8333	16000
m5a.4xlarge [2]	2880		360.0		16000	
m5a.8xlarge [2]	4750		593.75		20000	
m5a.12xlarge [2]	6780		847.5		30000	
m5a.16xlarge [2]	9500		1187.5		40000	
m5a.24xlarge [2]	13750		1718.75		60000	
m5ad.large [1]	650	2880	81.25	360.00	3600	16000
m5ad.xlarge [1]	1085	2880	135.62	360.00	6000	16000
m5ad.2xlarge [1]	1580	2880	197.50	360.00	8333	16000
m5ad.4xlarge [2]	2880		360.0		16000	
m5ad.8xlarge [2]	4750		593.75		20000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
m5ad.12xlarge [2]	6780		847.5		30000	
m5ad.16xlarge [2]	9500		1187.5		40000	
m5ad.24xlarge [2]	13750		1718.75		60000	
m5d.large [1]	650	4750	81.25	593.75	3600	18750
m5d.xlarge [1]	1150	4750	143.75	593.75	6000	18750
m5d.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
m5d.4xlarge [2]	4750		593.75		18750	
m5d.8xlarge [2]	6800		850.0		30000	
m5d.12xlarge [2]	9500		1187.5		40000	
m5d.16xlarge [2]	13600		1700.0		60000	
m5d.24xlarge [2]	19000		2375.0		80000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
m5d.metal [2]	19000		2375.0		80000	
m5dn.large [1]	650	4750	81.25	593.75	3600	18750
m5dn.xlarge [1]	1150	4750	143.75	593.75	6000	18750
m5dn.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
m5dn.4xlarge [2]	4750		593.75		18750	
m5dn.8xlarge [2]	6800		850.0		30000	
m5dn.12xlarge [2]	9500		1187.5		40000	
m5dn.16xlarge [2]	13600		1700.0		60000	
m5dn.24xlarge [2]	19000		2375.0		80000	
m5dn.metal [2]	19000		2375.0		80000	
m5n.large [1]	650	4750	81.25	593.75	3600	18750

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
m5n.xlarge [1]	1150	4750	143.75	593.75	6000	18750
m5n.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
m5n.4xlarge [2]	4750		593.75		18750	
m5n.8xlarge [2]	6800		850.0		30000	
m5n.12xlarge [2]	9500		1187.5		40000	
m5n.16xlarge [2]	13600		1700.0		60000	
m5n.24xlarge [2]	19000		2375.0		80000	
m5n.metal [2]	19000		2375.0		80000	
m5zn.large [1]	800	3170	100.00	396.25	3333	13333
m5zn.xlarge [1]	1564	3170	195.50	396.25	6667	13333
m5zn.2xlarge [2]	3170		396.25		13333	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
m5zn.3xlarge [2]	4750		593.75		20000	
m5zn.6xlarge [2]	9500		1187.5		40000	
m5zn.12xlarge [2]	19000		2375.0		80000	
m5zn.metal [2]	19000		2375.0		80000	
m6a.large [1]	650	10000	81.25	1250.00	3600	40000
m6a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m6a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m6a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m6a.8xlarge [2]	10000		1250.0		40000	
m6a.12xlarge [2]	15000		1875.0		60000	
m6a.16xlarge [2]	20000		2500.0		80000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
m6a.24xlarge [2]	30000		3750.0		120000	
m6a.32xlarge [2]	40000		5000.0		160000	
m6a.48xlarge [2]	40000		5000.0		240000	
m6a.metal [2]	40000		5000.0		240000	
m6g.medium [1]	315	4750	39.38	593.75	2500	20000
m6g.large [1]	630	4750	78.75	593.75	3600	20000
m6g.xlarge [1]	1188	4750	148.50	593.75	6000	20000
m6g.2xlarge [1]	2375	4750	296.88	593.75	12000	20000
m6g.4xlarge [2]	4750		593.75		20000	
m6g.8xlarge [2]	9500		1187.5		40000	
m6g.12xlarge [2]	14250		1781.25		50000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
m6g.16xlarge [2]	19000		2375.0		80000	
m6g.metal [2]	19000		2375.0		80000	
m6gd.medium [1]	315	4750	39.38	593.75	2500	20000
m6gd.large [1]	630	4750	78.75	593.75	3600	20000
m6gd.xlarge [1]	1188	4750	148.50	593.75	6000	20000
m6gd.2xlarge [1]	2375	4750	296.88	593.75	12000	20000
m6gd.4xlarge [2]	4750		593.75		20000	
m6gd.8xlarge [2]	9500		1187.5		40000	
m6gd.12xlarge [2]	14250		1781.25		50000	
m6gd.16xlarge [2]	19000		2375.0		80000	
m6gd.metal [2]	19000		2375.0		80000	
m6i.large [1]	650	10000	81.25	1250.00	3600	40000

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
m6i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m6i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m6i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m6i.8xlarge [2]	10000		1250.0		40000	
m6i.12xlarge [2]	15000		1875.0		60000	
m6i.16xlarge [2]	20000		2500.0		80000	
m6i.24xlarge [2]	30000		3750.0		120000	
m6i.32xlarge [2]	40000		5000.0		160000	
m6i.metal [2]	40000		5000.0		160000	
m6id.large [1]	650	10000	81.25	1250.00	3600	40000
m6id.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m6id.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
m6id.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m6id.8xlarge [2]	10000		1250.0		40000	
m6id.12xlarge [2]	15000		1875.0		60000	
m6id.16xlarge [2]	20000		2500.0		80000	
m6id.24xlarge [2]	30000		3750.0		120000	
m6id.32xlarge [2]	40000		5000.0		160000	
m6id.metal [2]	40000		5000.0		160000	
m6idn.large [1]	1562	25000	195.31	3125.00	6250	100000
m6idn.xlarge [1]	3125	25000	390.62	3125.00	12500	100000
m6idn.2xlarge [1]	6250	25000	781.25	3125.00	25000	100000
m6idn.4xlarge [1]	12500	25000	1562.50	3125.00	50000	100000

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
m6idn.8xlarge [2]	25000		3125.0		100000	
m6idn.12xlarge [2]	37500		4687.5		150000	
m6idn.16xlarge [2]	50000		6250.0		200000	
m6idn.24xlarge [2]	75000		9375.0		300000	
m6idn.32xlarge [2]	100000		12500.0		400000	
m6idn.metal [2]	100000		12500.0		400000	
m6in.large [1]	1562	25000	195.31	3125.00	6250	100000
m6in.xlarge [1]	3125	25000	390.62	3125.00	12500	100000
m6in.2xlarge [1]	6250	25000	781.25	3125.00	25000	100000
m6in.4xlarge [1]	12500	25000	1562.50	3125.00	50000	100000
m6in.8xlarge [2]	25000		3125.0		100000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
m6in.12xlarge [2]	37500		4687.5		150000	
m6in.16xlarge [2]	50000		6250.0		200000	
m6in.24xlarge [2]	75000		9375.0		300000	
m6in.32xlarge [2]	100000		12500.0		400000	
m6in.metal [2]	100000		12500.0		400000	
m7a.medium [1]	325	10000	40.62	1250.00	2500	40000
m7a.large [1]	650	10000	81.25	1250.00	3600	40000
m7a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m7a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m7a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m7a.8xlarge [2]	10000		1250.0		40000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
m7a.12xlarge [2]	15000		1875.0		60000	
m7a.16xlarge [2]	20000		2500.0		80000	
m7a.24xlarge [2]	30000		3750.0		120000	
m7a.32xlarge [2]	40000		5000.0		160000	
m7a.48xlarge [2]	40000		5000.0		240000	
m7a.metal-48xl [2]	40000		5000.0		240000	
m7g.medium [1]	315	10000	39.38	1250.00	2500	40000
m7g.large [1]	630	10000	78.75	1250.00	3600	40000
m7g.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m7g.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m7g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
m7g.8xlarge [2]	10000		1250.0		40000	
m7g.12xlarge [2]	15000		1875.0		60000	
m7g.16xlarge [2]	20000		2500.0		80000	
m7g.metal [2]	20000		2500.0		80000	
m7gd.medium [1]	315	10000	39.38	1250.00	2500	40000
m7gd.large [1]	630	10000	78.75	1250.00	3600	40000
m7gd.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m7gd.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m7gd.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m7gd.8xlarge [2]	10000		1250.0		40000	
m7gd.12xlarge [2]	15000		1875.0		60000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
m7gd.16xlarge [2]	20000		2500.0		80000	
m7gd.metal [2]	20000		2500.0		80000	
m7i.large [1]	650	10000	81.25	1250.00	3600	40000
m7i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m7i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m7i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m7i.8xlarge [2]	10000		1250.0		40000	
m7i.12xlarge [2]	15000		1875.0		60000	
m7i.16xlarge [2]	20000		2500.0		80000	
m7i.24xlarge [2]	30000		3750.0		120000	
m7i.48xlarge [2]	40000		5000.0		240000	
m7i.metal-24xl [2]	30000		3750.0		120000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
m7i.metal-48xl [2]	40000		5000.0		240000	
m7i-flex.large [1]	312	10000	39.06	1250.00	2500	40000
m7i-flex.xlarge [1]	625	10000	78.12	1250.00	3600	40000
m7i-flex.2xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m7i-flex.4xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m7i-flex.8xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m7i-flex.12xlarge [1]	7500	15000	937.50	1875.00	30000	60000
m7i-flex.16xlarge [1]	10000	20000	1250.00	2500.00	40000	80000
m8g.medium [1]	315	10000	39.38	1250.00	2500	40000
m8g.large [1]	630	10000	78.75	1250.00	3600	40000
m8g.xlarge [1]	1250	10000	156.25	1250.00	6000	40000

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
m8g.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m8g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m8g.8xlarge [2]	10000		1250.0		40000	
m8g.12xlarge [2]	15000		1875.0		60000	
m8g.16xlarge [2]	20000		2500.0		80000	
m8g.24xlarge [2]	30000		3750.0		120000	
m8g.48xlarge [2]	40000		5000.0		240000	
m8g.metal-24xl [2]	30000		3750.0		120000	
m8g.metal-48xl [2]	40000		5000.0		240000	
m8gd.medium[1]	315	10000	39.38	1250.00	2500	40000
m8gd.large[1]	630	10000	78.75	1250.00	3600	40000

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
m8gd.xlarge[1]	1250	10000	156.25	1250.00	6000	40000
m8gd.2xlarge[1]	2500	10000	312.50	1250.00	12000	40000
m8gd.4xlarge[1]	5000	10000	625.00	1250.00	20000	40000
m8gd.8xlarge[2]	10000		1250.0		40000	
m8gd.12xlarge[2]	15000		1875.0		60000	
m8gd.16xlarge[2]	20000		2500.0		80000	
m8gd.24xlarge[2]	30000		3750.0		120000	
m8gd.48xlarge[2]	40000		5000.0		240000	
m8gd.metal-24xl[2]	30000		3750.0		120000	
m8gd.metal-48xl[2]	40000		5000.0		240000	
mac1.metal[2]	14000		1750.0		80000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
mac2.metal [2]	10000		1250.0		55000	
mac2-m1ultra.metal [2]	10000		1250.0		55000	
mac2-m2.metal [2]	8000		1000.0		55000	
mac2-m2pro.metal [2]	8000		1000.0		55000	
t3.nano [1]	43	2085	5.38	260.62	250	11800
t3.micro [1]	87	2085	10.88	260.62	500	11800
t3.small [1]	174	2085	21.75	260.62	1000	11800
t3.medium [1]	347	2085	43.38	260.62	2000	11800
t3.large [1]	695	2780	86.88	347.50	4000	15700
t3.xlarge [1]	695	2780	86.88	347.50	4000	15700
t3.2xlarge [1]	695	2780	86.88	347.50	4000	15700
t3a.nano [1]	45	2085	5.62	260.62	250	11800
t3a.micro [1]	90	2085	11.25	260.62	500	11800
t3a.small [1]	175	2085	21.88	260.62	1000	11800
t3a.medium [1]	350	2085	43.75	260.62	2000	11800

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
t3a.large [1]	695	2780	86.88	347.50	4000	15700
t3a.xlarge [1]	695	2780	86.88	347.50	4000	15700
t3a.2xlarge [1]	695	2780	86.88	347.50	4000	15700
t4g.nano [1]	43	2085	5.38	260.62	250	11800
t4g.micro [1]	87	2085	10.88	260.62	500	11800
t4g.small [1]	174	2085	21.75	260.62	1000	11800
t4g.medium [1]	347	2085	43.38	260.62	2000	11800
t4g.large [1]	695	2780	86.88	347.50	4000	15700
t4g.xlarge [1]	695	2780	86.88	347.50	4000	15700
t4g.2xlarge [1]	695	2780	86.88	347.50	4000	15700

[1] 이러한 인스턴스는 24시간에 한 번 이상 30분 동안 최대 성능을 유지할 수 있습니다. 이후에는 기준 성능으로 돌아갑니다.

[2] 이러한 인스턴스는 명시된 성능을 무기한으로 유지할 수 있습니다. 최대 성능이 30분 이상 지속되어야 하는 워크로드가 있는 경우 이러한 인스턴스 중 하나를 사용합니다.

컴퓨팅 최적화

> **Note**
>
> C8g, C8gd 인스턴스 유형은 구성 가능한 대역폭 가중치를 지원합니다. 이러한 인스턴스 유형을 사용하면 네트워킹 성능 또는 Amazon EBS 성능에 맞게 인스턴스의 대역폭을 최적화할 수

있습니다. 다음 표에는 이러한 인스턴스 유형에 대한 기본 Amazon EBS 대역폭 성능이 나와 있습니다. 자세한 내용은 EC2 인스턴스 대역폭 가중치 구성 섹션을 참조하세요.

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
c4.large [2]	500		62.5		4000	
c4.xlarge [2]	750		93.75		6000	
c4.2xlarge [2]	1000		125.0		8000	
c4.4xlarge [2]	2000		250.0		16000	
c4.8xlarge [2]	4000		500.0		32000	
c5.large [1]	650	4750	81.25	593.75	4000	20000
c5.xlarge [1]	1150	4750	143.75	593.75	6000	20000
c5.2xlarge [1]	2300	4750	287.50	593.75	10000	20000
c5.4xlarge [2]	4750		593.75		20000	
c5.9xlarge [2]	9500		1187.5		40000	
c5.12xlarge [2]	9500		1187.5		40000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
c5.18xlarge [2]	19000		2375.0		80000	
c5.24xlarge [2]	19000		2375.0		80000	
c5.metal [2]	19000		2375.0		80000	
c5a.large [1]	200	3170	25.00	396.25	800	13300
c5a.xlarge [1]	400	3170	50.00	396.25	1600	13300
c5a.2xlarge [1]	800	3170	100.00	396.25	3200	13300
c5a.4xlarge [1]	1580	3170	197.50	396.25	6600	13300
c5a.8xlarge [2]	3170		396.25		13300	
c5a.12xlarge [2]	4750		593.75		20000	
c5a.16xlarge [2]	6300		787.5		26700	
c5a.24xlarge [2]	9500		1187.5		40000	
c5ad.large [1]	200	3170	25.00	396.25	800	13300

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
c5ad.xlarge [1]	400	3170	50.00	396.25	1600	13300
c5ad.2xlarge [1]	800	3170	100.00	396.25	3200	13300
c5ad.4xlarge [1]	1580	3170	197.50	396.25	6600	13300
c5ad.8xlarge [2]	3170		396.25		13300	
c5ad.12xlarge [2]	4750		593.75		20000	
c5ad.16xlarge [2]	6300		787.5		26700	
c5ad.24xlarge [2]	9500		1187.5		40000	
c5d.large [1]	650	4750	81.25	593.75	4000	20000
c5d.xlarge [1]	1150	4750	143.75	593.75	6000	20000
c5d.2xlarge [1]	2300	4750	287.50	593.75	10000	20000
c5d.4xlarge [2]	4750		593.75		20000	
c5d.9xlarge [2]	9500		1187.5		40000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
c5d.12xlarge [2]	9500		1187.5		40000	
c5d.18xlarge [2]	19000		2375.0		80000	
c5d.24xlarge [2]	19000		2375.0		80000	
c5d.metal [2]	19000		2375.0		80000	
c5n.large [1]	650	4750	81.25	593.75	4000	20000
c5n.xlarge [1]	1150	4750	143.75	593.75	6000	20000
c5n.2xlarge [1]	2300	4750	287.50	593.75	10000	20000
c5n.4xlarge [2]	4750		593.75		20000	
c5n.9xlarge [2]	9500		1187.5		40000	
c5n.18xlarge [2]	19000		2375.0		80000	
c5n.metal [2]	19000		2375.0		80000	
c6a.large [1]	650	10000	81.25	1250.00	3600	40000
c6a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
c6a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c6a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c6a.8xlarge [2]	10000		1250.0		40000	
c6a.12xlarge [2]	15000		1875.0		60000	
c6a.16xlarge [2]	20000		2500.0		80000	
c6a.24xlarge [2]	30000		3750.0		120000	
c6a.32xlarge [2]	40000		5000.0		160000	
c6a.48xlarge [2]	40000		5000.0		240000	
c6a.metal [2]	40000		5000.0		240000	
c6g.medium [1]	315	4750	39.38	593.75	2500	20000
c6g.large [1]	630	4750	78.75	593.75	3600	20000
c6g.xlarge [1]	1188	4750	148.50	593.75	6000	20000

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
c6g.2xlarge [1]	2375	4750	296.88	593.75	12000	20000
c6g.4xlarge [2]	4750		593.75		20000	
c6g.8xlarge [2]	9500		1187.5		40000	
c6g.12xlarge [2]	14250		1781.25		50000	
c6g.16xlarge [2]	19000		2375.0		80000	
c6g.metal [2]	19000		2375.0		80000	
c6gd.medium [1]	315	4750	39.38	593.75	2500	20000
c6gd.large [1]	630	4750	78.75	593.75	3600	20000
c6gd.xlarge [1]	1188	4750	148.50	593.75	6000	20000
c6gd.2xlarge [1]	2375	4750	296.88	593.75	12000	20000
c6gd.4xlarge [2]	4750		593.75		20000	
c6gd.8xlarge [2]	9500		1187.5		40000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
c6gd.12xlarge [2]	14250		1781.25		50000	
c6gd.16xlarge [2]	19000		2375.0		80000	
c6gd.metal [2]	19000		2375.0		80000	
c6gn.medium [1]	760	9500	95.00	1187.50	2500	40000
c6gn.large [1]	1235	9500	154.38	1187.50	5000	40000
c6gn.xlarge [1]	2375	9500	296.88	1187.50	10000	40000
c6gn.2xlarge [1]	4750	9500	593.75	1187.50	20000	40000
c6gn.4xlarge [2]	9500		1187.5		40000	
c6gn.8xlarge [2]	19000		2375.0		80000	
c6gn.12xlarge [2]	28500		3562.5		120000	
c6gn.16xlarge [2]	38000		4750.0		160000	
c6i.large [1]	650	10000	81.25	1250.00	3600	40000

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
c6i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c6i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c6i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c6i.8xlarge [2]	10000		1250.0		40000	
c6i.12xlarge [2]	15000		1875.0		60000	
c6i.16xlarge [2]	20000		2500.0		80000	
c6i.24xlarge [2]	30000		3750.0		120000	
c6i.32xlarge [2]	40000		5000.0		160000	
c6i.metal [2]	40000		5000.0		160000	
c6id.large [1]	650	10000	81.25	1250.00	3600	40000
c6id.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c6id.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c6id.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
c6id.8xlarge [2]	10000		1250.0		40000	
c6id.12xlarge [2]	15000		1875.0		60000	
c6id.16xlarge [2]	20000		2500.0		80000	
c6id.24xlarge [2]	30000		3750.0		120000	
c6id.32xlarge [2]	40000		5000.0		160000	
c6id.metal [2]	40000		5000.0		160000	
c6in.large [1]	1562	25000	195.31	3125.00	6250	100000
c6in.xlarge [1]	3125	25000	390.62	3125.00	12500	100000
c6in.2xlarge [1]	6250	25000	781.25	3125.00	25000	100000
c6in.4xlarge [1]	12500	25000	1562.50	3125.00	50000	100000
c6in.8xlarge [2]	25000		3125.0		100000	
c6in.12xlarge [2]	37500		4687.5		150000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
c6in.16xlarge [2]	50000		6250.0		200000	
c6in.24xlarge [2]	75000		9375.0		300000	
c6in.32xlarge [2]	100000		12500.0		400000	
c6in.metal [2]	100000		12500.0		400000	
c7a.medium [1]	325	10000	40.62	1250.00	2500	40000
c7a.large [1]	650	10000	81.25	1250.00	3600	40000
c7a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c7a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c7a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c7a.8xlarge [2]	10000		1250.0		40000	
c7a.12xlarge [2]	15000		1875.0		60000	
c7a.16xlarge [2]	20000		2500.0		80000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
c7a.24xlarge [2]	30000		3750.0		120000	
c7a.32xlarge [2]	40000		5000.0		160000	
c7a.48xlarge [2]	40000		5000.0		240000	
c7a.metal-48xl [2]	40000		5000.0		240000	
c7g.medium [1]	315	10000	39.38	1250.00	2500	40000
c7g.large [1]	630	10000	78.75	1250.00	3600	40000
c7g.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c7g.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c7g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c7g.8xlarge [2]	10000		1250.0		40000	
c7g.12xlarge [2]	15000		1875.0		60000	
c7g.16xlarge [2]	20000		2500.0		80000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
c7g.metal [2]	20000		2500.0		80000	
c7gd.medium [1]	315	10000	39.38	1250.00	2500	40000
c7gd.large [1]	630	10000	78.75	1250.00	3600	40000
c7gd.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c7gd.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c7gd.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c7gd.8xlarge [2]	10000		1250.0		40000	
c7gd.12xlarge [2]	15000		1875.0		60000	
c7gd.16xlarge [2]	20000		2500.0		80000	
c7gd.metal [2]	20000		2500.0		80000	
c7gn.medium [1]	521	10000	65.12	1250.00	2083	40000
c7gn.large [1]	1042	10000	130.25	1250.00	4167	40000

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
c7gn.xlarge [1]	2083	10000	260.38	1250.00	8333	40000
c7gn.2xlarge [1]	4167	10000	520.88	1250.00	16667	40000
c7gn.4xlarge [1]	8333	10000	1041.62	1250.00	33333	40000
c7gn.8xlarge [1]	16667	20000	2083.38	2500.00	66667	80000
c7gn.12xlarge [1]	25000	30000	3125.00	3750.00	100000	120000
c7gn.16xlarge [1]	33333	40000	4166.62	5000.00	133333	160000
c7gn.metal [1]	33333	40000	4166.62	5000.00	133333	160000
c7i.large [1]	650	10000	81.25	1250.00	3600	40000
c7i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c7i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c7i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c7i.8xlarge [2]	10000		1250.0		40000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
c7i.12xlarge [2]	15000		1875.0		60000	
c7i.16xlarge [2]	20000		2500.0		80000	
c7i.24xlarge [2]	30000		3750.0		120000	
c7i.48xlarge [2]	40000		5000.0		240000	
c7i.metal-24xl [2]	30000		3750.0		120000	
c7i.metal-48xl [2]	40000		5000.0		240000	
c7i-flex.large [1]	312	10000	39.06	1250.00	2500	40000
c7i-flex.xlarge [1]	625	10000	78.12	1250.00	3600	40000
c7i-flex.2xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c7i-flex.4xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c7i-flex.8xlarge [1]	5000	10000	625.00	1250.00	20000	40000

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
c7i-flex.12xlarge [1]	7500	15000	937.50	1875.00	30000	60000
c7i-flex.16xlarge [1]	10000	20000	1250.00	2500.00	40000	80000
c8g.medium [1]	315	10000	39.38	1250.00	2500	40000
c8g.large [1]	630	10000	78.75	1250.00	3600	40000
c8g.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c8g.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c8g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c8g.8xlarge [2]	10000		1250.0		40000	
c8g.12xlarge [2]	15000		1875.0		60000	
c8g.16xlarge [2]	20000		2500.0		80000	
c8g.24xlarge [2]	30000		3750.0		120000	
c8g.48xlarge [2]	40000		5000.0		240000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
c8g.metal-24xl [2]	30000		3750.0		120000	
c8g.metal-48xl [2]	40000		5000.0		240000	
c8gd.medium[1]	315	10000	39.38	1250.00	2500	40000
c8gd.large[1]	630	10000	78.75	1250.00	3600	40000
c8gd.xlarge[1]	1250	10000	156.25	1250.00	6000	40000
c8gd.2xlarge[1]	2500	10000	312.50	1250.00	12000	40000
c8gd.4xlarge[1]	5000	10000	625.00	1250.00	20000	40000
c8gd.8xlarge[2]	10000		1250.0		40000	
c8gd.12xlarge[2]	15000		1875.0		60000	
c8gd.16xlarge[2]	20000		2500.0		80000	
c8gd.24xlarge[2]	30000		3750.0		120000	
c8gd.48xlarge[2]	40000		5000.0		240000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
c8gd.metal-24xl[2]	30000		3750.0		120000	
c8gd.metal-48xl[2]	40000		5000.0		240000	

[1] 이러한 인스턴스는 24시간에 한 번 이상 30분 동안 최대 성능을 유지할 수 있습니다. 이후에는 기준 성능으로 돌아갑니다.

[2] 이러한 인스턴스는 명시된 성능을 무기한으로 유지할 수 있습니다. 최대 성능이 30분 이상 지속되어야 하는 워크로드가 있는 경우 이러한 인스턴스 중 하나를 사용합니다.

메모리 최적화

> **Note**
>
> - R8g, X8g, X8g 인스턴스 유형은 구성 가능한 대역폭 가중치를 지원합니다. 이러한 인스턴스 유형을 사용하면 네트워킹 성능 또는 Amazon EBS 성능에 맞게 인스턴스의 대역폭을 최적화할 수 있습니다. 다음 표에는 이러한 인스턴스 유형에 대한 기본 Amazon EBS 대역폭 성능이 나와 있습니다. 자세한 내용은 EC2 인스턴스 대역폭 가중치 구성 섹션을 참조하세요.
> - U7i 인스턴스에서 IOPS 성능을 극대화하려면 io2 BlockExpress 볼륨을 사용하는 것이 좋습니다.

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
r4.large [2]	425		53.125		3000	
r4.xlarge [2]	850		106.25		6000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
r4.2xlarge [2]	1700		212.5		12000	
r4.4xlarge [2]	3500		437.5		18750	
r4.8xlarge [2]	7000		875.0		37500	
r4.16xlarge [2]	14000		1750.0		75000	
r5.large [1]	650	4750	81.25	593.75	3600	18750
r5.xlarge [1]	1150	4750	143.75	593.75	6000	18750
r5.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
r5.4xlarge [2]	4750		593.75		18750	
r5.8xlarge [2]	6800		850.0		30000	
r5.12xlarge [2]	9500		1187.5		40000	
r5.16xlarge [2]	13600		1700.0		60000	
r5.24xlarge [2]	19000		2375.0		80000	
r5.metal [2]	19000		2375.0		80000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
r5a.large [1]	650	2880	81.25	360.00	3600	16000
r5a.xlarge [1]	1085	2880	135.62	360.00	6000	16000
r5a.2xlarge [1]	1580	2880	197.50	360.00	8333	16000
r5a.4xlarge [2]	2880		360.0		16000	
r5a.8xlarge [2]	4750		593.75		20000	
r5a.12xlarge [2]	6780		847.5		30000	
r5a.16xlarge [2]	9500		1187.5		40000	
r5a.24xlarge [2]	13570		1696.25		60000	
r5ad.large [1]	650	2880	81.25	360.00	3600	16000
r5ad.xlarge [1]	1085	2880	135.62	360.00	6000	16000
r5ad.2xlarge [1]	1580	2880	197.50	360.00	8333	16000
r5ad.4xlarge [2]	2880		360.0		16000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
r5ad.8xlarge [2]	4750		593.75		20000	
r5ad.12xlarge [2]	6780		847.5		30000	
r5ad.16xlarge [2]	9500		1187.5		40000	
r5ad.24xlarge [2]	13570		1696.25		60000	
r5b.large [1]	1250	10000	156.25	1250.00	5417	43333
r5b.xlarge [1]	2500	10000	312.50	1250.00	10833	43333
r5b.2xlarge [1]	5000	10000	625.00	1250.00	21667	43333
r5b.4xlarge [2]	10000		1250.0		43333	
r5b.8xlarge [2]	20000		2500.0		86667	
r5b.12xlarge [2]	30000		3750.0		130000	
r5b.16xlarge [2]	40000		5000.0		173333	
r5b.24xlarge [2]	60000		7500.0		260000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
r5b.metal [2]	60000		7500.0		260000	
r5d.large [1]	650	4750	81.25	593.75	3600	18750
r5d.xlarge [1]	1150	4750	143.75	593.75	6000	18750
r5d.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
r5d.4xlarge [2]	4750		593.75		18750	
r5d.8xlarge [2]	6800		850.0		30000	
r5d.12xlarge [2]	9500		1187.5		40000	
r5d.16xlarge [2]	13600		1700.0		60000	
r5d.24xlarge [2]	19000		2375.0		80000	
r5d.metal [2]	19000		2375.0		80000	
r5dn.large [1]	650	4750	81.25	593.75	3600	18750
r5dn.xlarge [1]	1150	4750	143.75	593.75	6000	18750
r5dn.2xlarge [1]	2300	4750	287.50	593.75	12000	18750

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
r5dn.4xlarge [2]	4750		593.75		18750	
r5dn.8xlarge [2]	6800		850.0		30000	
r5dn.12xlarge [2]	9500		1187.5		40000	
r5dn.16xlarge [2]	13600		1700.0		60000	
r5dn.24xlarge [2]	19000		2375.0		80000	
r5dn.metal [2]	19000		2375.0		80000	
r5n.large [1]	650	4750	81.25	593.75	3600	18750
r5n.xlarge [1]	1150	4750	143.75	593.75	6000	18750
r5n.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
r5n.4xlarge [2]	4750		593.75		18750	
r5n.8xlarge [2]	6800		850.0		30000	
r5n.12xlarge [2]	9500		1187.5		40000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
r5n.16xlarge [2]	13600		1700.0		60000	
r5n.24xlarge [2]	19000		2375.0		80000	
r5n.metal [2]	19000		2375.0		80000	
r6a.large [1]	650	10000	81.25	1250.00	3600	40000
r6a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r6a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r6a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r6a.8xlarge [2]	10000		1250.0		40000	
r6a.12xlarge [2]	15000		1875.0		60000	
r6a.16xlarge [2]	20000		2500.0		80000	
r6a.24xlarge [2]	30000		3750.0		120000	
r6a.32xlarge [2]	40000		5000.0		160000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
r6a.48xlarge [2]	40000		5000.0		240000	
r6a.metal [2]	40000		5000.0		240000	
r6g.medium [1]	315	4750	39.38	593.75	2500	20000
r6g.large [1]	630	4750	78.75	593.75	3600	20000
r6g.xlarge [1]	1188	4750	148.50	593.75	6000	20000
r6g.2xlarge [1]	2375	4750	296.88	593.75	12000	20000
r6g.4xlarge [2]	4750		593.75		20000	
r6g.8xlarge [2]	9500		1187.5		40000	
r6g.12xlarge [2]	14250		1781.25		50000	
r6g.16xlarge [2]	19000		2375.0		80000	
r6g.metal [2]	19000		2375.0		80000	
r6gd.medium [1]	315	4750	39.38	593.75	2500	20000
r6gd.large [1]	630	4750	78.75	593.75	3600	20000

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
r6gd.xlarge [1]	1188	4750	148.50	593.75	6000	20000
r6gd.2xlarge [1]	2375	4750	296.88	593.75	12000	20000
r6gd.4xlarge [2]	4750		593.75		20000	
r6gd.8xlarge [2]	9500		1187.5		40000	
r6gd.12xlarge [2]	14250		1781.25		50000	
r6gd.16xlarge [2]	19000		2375.0		80000	
r6gd.metal [2]	19000		2375.0		80000	
r6i.large [1]	650	10000	81.25	1250.00	3600	40000
r6i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r6i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r6i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r6i.8xlarge [2]	10000		1250.0		40000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
r6i.12xlarge [2]	15000		1875.0		60000	
r6i.16xlarge [2]	20000		2500.0		80000	
r6i.24xlarge [2]	30000		3750.0		120000	
r6i.32xlarge [2]	40000		5000.0		160000	
r6i.metal [2]	40000		5000.0		160000	
r6idn.large [1]	1562	25000	195.31	3125.00	6250	100000
r6idn.xlarge [1]	3125	25000	390.62	3125.00	12500	100000
r6idn.2xlarge [1]	6250	25000	781.25	3125.00	25000	100000
r6idn.4xlarge [1]	12500	25000	1562.50	3125.00	50000	100000
r6idn.8xlarge [2]	25000		3125.0		100000	
r6idn.12xlarge [2]	37500		4687.5		150000	
r6idn.16xlarge [2]	50000		6250.0		200000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
r6idn.24xlarge [2]	75000		9375.0		300000	
r6idn.32xlarge [2]	100000		12500.0		400000	
r6idn.metal [2]	100000		12500.0		400000	
r6in.large [1]	1562	25000	195.31	3125.00	6250	100000
r6in.xlarge [1]	3125	25000	390.62	3125.00	12500	100000
r6in.2xlarge [1]	6250	25000	781.25	3125.00	25000	100000
r6in.4xlarge [1]	12500	25000	1562.50	3125.00	50000	100000
r6in.8xlarge [2]	25000		3125.0		100000	
r6in.12xlarge [2]	37500		4687.5		150000	
r6in.16xlarge [2]	50000		6250.0		200000	
r6in.24xlarge [2]	75000		9375.0		300000	
r6in.32xlarge [2]	100000		12500.0		400000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
r6in.metal [2]	100000		12500.0		400000	
r6id.large [1]	650	10000	81.25	1250.00	3600	40000
r6id.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r6id.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r6id.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r6id.8xlarge [2]	10000		1250.0		40000	
r6id.12xlarge [2]	15000		1875.0		60000	
r6id.16xlarge [2]	20000		2500.0		80000	
r6id.24xlarge [2]	30000		3750.0		120000	
r6id.32xlarge [2]	40000		5000.0		160000	
r6id.metal [2]	40000		5000.0		160000	
r7a.medium [1]	325	10000	40.62	1250.00	2500	40000
r7a.large [1]	650	10000	81.25	1250.00	3600	40000

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
r7a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r7a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r7a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r7a.8xlarge [2]	10000		1250.0		40000	
r7a.12xlarge [2]	15000		1875.0		60000	
r7a.16xlarge [2]	20000		2500.0		80000	
r7a.24xlarge [2]	30000		3750.0		120000	
r7a.32xlarge [2]	40000		5000.0		160000	
r7a.48xlarge [2]	40000		5000.0		240000	
r7a.metal-48xl [2]	40000		5000.0		240000	
r7g.medium [1]	315	10000	39.38	1250.00	2500	40000
r7g.large [1]	630	10000	78.75	1250.00	3600	40000

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
r7g.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r7g.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r7g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r7g.8xlarge [2]	10000		1250.0		40000	
r7g.12xlarge [2]	15000		1875.0		60000	
r7g.16xlarge [2]	20000		2500.0		80000	
r7g.metal [2]	20000		2500.0		80000	
r7gd.medium [1]	315	10000	39.38	1250.00	2500	40000
r7gd.large [1]	630	10000	78.75	1250.00	3600	40000
r7gd.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r7gd.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r7gd.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
r7gd.8xlarge [2]	10000		1250.0		40000	
r7gd.12xlarge [2]	15000		1875.0		60000	
r7gd.16xlarge [2]	20000		2500.0		80000	
r7gd.metal [2]	20000		2500.0		80000	
r7i.large [1]	650	10000	81.25	1250.00	3600	40000
r7i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r7i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r7i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r7i.8xlarge [2]	10000		1250.0		40000	
r7i.12xlarge [2]	15000		1875.0		60000	
r7i.16xlarge [2]	20000		2500.0		80000	
r7i.24xlarge [2]	30000		3750.0		120000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
r7i.48xlarge [2]	40000		5000.0		240000	
r7i.metal-24xl [2]	30000		3750.0		120000	
r7i.metal-48xl [2]	40000		5000.0		240000	
r7iz.large [1]	792	10000	99.00	1250.00	3600	40000
r7iz.xlarge [1]	1584	10000	198.00	1250.00	6667	40000
r7iz.2xlarge [1]	3168	10000	396.00	1250.00	13333	40000
r7iz.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r7iz.8xlarge [2]	10000		1250.0		40000	
r7iz.12xlarge [2]	19000		2375.0		76000	
r7iz.16xlarge [2]	20000		2500.0		80000	
r7iz.32xlarge [2]	40000		5000.0		160000	
r7iz.metal-16xl [2]	20000		2500.0		80000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
r7iz.metal-32xl [2]	40000		5000.0		160000	
r8g.medium [1]	315	10000	39.38	1250.00	2500	40000
r8g.large [1]	630	10000	78.75	1250.00	3600	40000
r8g.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r8g.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r8g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r8g.8xlarge [2]	10000		1250.0		40000	
r8g.12xlarge [2]	15000		1875.0		60000	
r8g.16xlarge [2]	20000		2500.0		80000	
r8g.24xlarge [2]	30000		3750.0		120000	
r8g.48xlarge [2]	40000		5000.0		240000	
r8g.metal-24xl [2]	30000		3750.0		120000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
r8g.metal-48xl [2]	40000		5000.0		240000	
r8gd.medium[1]	315	10000	39.38	1250.00	2500	40000
r8gd.large[1]	630	10000	78.75	1250.00	3600	40000
r8gd.xlarge[1]	1250	10000	156.25	1250.00	6000	40000
r8gd.2xlarge[1]	2500	10000	312.50	1250.00	12000	40000
r8gd.4xlarge[1]	5000	10000	625.00	1250.00	20000	40000
r8gd.8xlarge[2]	10000		1250.0		40000	
r8gd.12xlarge[2]	15000		1875.0		60000	
r8gd.16xlarge[2]	20000		2500.0		80000	
r8gd.24xlarge[2]	30000		3750.0		120000	
r8gd.48xlarge[2]	40000		5000.0		240000	
r8gd.metal-24xl[2]	30000		3750.0		120000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
r8gd.metal-48xl[2]	40000		5000.0		240000	
u-3tb1.56xlarge [2]	19000		2375.0		80000	
u-6tb1.56xlarge [2]	38000		4750.0		160000	
u-6tb1.112xlarge [2]	38000		4750.0		160000	
u-6tb1.metal [2]	38000		4750.0		160000	
u-9tb1.112xlarge [2]	38000		4750.0		160000	
u-9tb1.metal [2]	38000		4750.0		160000	
u-12tb1.112xlarge [2]	38000		4750.0		160000	
u-12tb1.metal [2]	38000		4750.0		160000	
u-18tb1.112xlarge [2]	38000		4750.0		160000	
u-18tb1.metal [2]	38000		4750.0		160000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
u-24tb1.112xlarge [2]	38000		4750.0		160000	
u-24tb1.metal [2]	38000		4750.0		160000	
u7i-6tb.112xlarge [2]	100000		12500.0		560000	
u7i-8tb.112xlarge [2]	100000		12500.0		560000	
u7i-12tb.224xlarge [2]	60000		7500.0		420000	
u7in-16tb.224xlarge [2]	100000		12500.0		420000	
u7in-24tb.224xlarge [2]	100000		12500.0		420000	
u7in-32tb.224xlarge [2]	100000		12500.0		420000	
u7inh-32tb.480xlarge [2]	160000		20000.0		840000	
x1.16xlarge [2]	7000		875.0		40000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
x1.32xlarge [2]	14000		1750.0		80000	
x1e.xlarge [2]	500		62.5		3700	
x1e.2xlarge [2]	1000		125.0		7400	
x1e.4xlarge [2]	1750		218.75		10000	
x1e.8xlarge [2]	3500		437.5		20000	
x1e.16xlarge [2]	7000		875.0		40000	
x1e.32xlarge [2]	14000		1750.0		80000	
x2gd.medium [1]	315	4750	39.38	593.75	2500	20000
x2gd.large [1]	630	4750	78.75	593.75	3600	20000
x2gd.xlarge [1]	1188	4750	148.50	593.75	6000	20000
x2gd.2xlarge [1]	2375	4750	296.88	593.75	12000	20000

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
x2gd.4xlarge [2]	4750		593.75		20000	
x2gd.8xlarge [2]	9500		1187.5		40000	
x2gd.12xlarge [2]	14250		1781.25		60000	
x2gd.16xlarge [2]	19000		2375.0		80000	
x2gd.metal [2]	19000		2375.0		80000	
x2idn.16xlarge [2]	40000		5000.0		173333	
x2idn.24xlarge [2]	60000		7500.0		260000	
x2idn.32xlarge [2]	80000		10000.0		260000	
x2idn.metal [2]	80000		10000.0		260000	
x2iedn.xlarge [1]	2500	20000	312.50	2500.00	8125	65000
x2iedn.2xlarge [1]	5000	20000	625.00	2500.00	16250	65000

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
x2iedn.4xlarge [1]	10000	20000	1250.00	2500.00	32500	65000
x2iedn.8xlarge [2]	20000		2500.0		65000	
x2iedn.16xlarge [2]	40000		5000.0		130000	
x2iedn.24xlarge [2]	60000		7500.0		195000	
x2iedn.32xlarge [2]	80000		10000.0		260000	
x2iedn.metal [2]	80000		10000.0		260000	
x2iezn.2xlarge [2]	3170		396.25		13333	
x2iezn.4xlarge [2]	4750		593.75		20000	
x2iezn.6xlarge [2]	9500		1187.5		40000	
x2iezn.8xlarge [2]	12000		1500.0		55000	
x2iezn.12xlarge [2]	19000		2375.0		80000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
x2iezn.metal [2]	19000		2375.0		80000	
x8g.medium [1]	315	10000	39.38	1250.00	2500	40000
x8g.large [1]	630	10000	78.75	1250.00	3600	40000
x8g.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
x8g.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
x8g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
x8g.8xlarge [2]	10000		1250.0		40000	
x8g.12xlarge [2]	15000		1875.0		60000	
x8g.16xlarge [2]	20000		2500.0		80000	
x8g.24xlarge [2]	30000		3750.0		120000	
x8g.48xlarge [2]	40000		5000.0		240000	
x8g.metal-24xl [2]	30000		3750.0		120000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
x8g.metal-48xl [2]	40000		5000.0		240000	
z1d.large [1]	800	3170	100.00	396.25	3333	13333
z1d.xlarge [1]	1580	3170	197.50	396.25	6667	13333
z1d.2xlarge [2]	3170		396.25		13333	
z1d.3xlarge [2]	4750		593.75		20000	
z1d.6xlarge [2]	9500		1187.5		40000	
z1d.12xlarge [2]	19000		2375.0		80000	
z1d.metal [2]	19000		2375.0		80000	

[1] 이러한 인스턴스는 24시간에 한 번 이상 30분 동안 최대 성능을 유지할 수 있습니다. 이후에는 기준 성능으로 돌아갑니다.

[2] 이러한 인스턴스는 명시된 성능을 무기한으로 유지할 수 있습니다. 최대 성능이 30분 이상 지속되어야 하는 워크로드가 있는 경우 이러한 인스턴스 중 하나를 사용합니다.

스토리지 최적화

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
d2.xlarge [2]	750		93.75		6000	
d2.2xlarge [2]	1000		125.0		8000	
d2.4xlarge [2]	2000		250.0		16000	
d2.8xlarge [2]	4000		500.0		32000	
d3.xlarge [1]	850	2800	106.25	350.00	5000	15000
d3.2xlarge [1]	1700	2800	212.50	350.00	10000	15000
d3.4xlarge [2]	2800		350.0		15000	
d3.8xlarge [2]	5000		625.0		30000	
d3en.xlarge [1]	850	2800	106.25	350.00	5000	15000
d3en.2xlarge [1]	1700	2800	212.50	350.00	10000	15000
d3en.4xlarge [2]	2800		350.0		15000	
d3en.6xlarge [2]	4000		500.0		25000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
d3en.8xlarge [2]	5000		625.0		30000	
d3en.12xlarge [2]	7000		875.0		40000	
h1.2xlarge [2]	1750		218.75		12000	
h1.4xlarge [2]	3500		437.5		20000	
h1.8xlarge [2]	7000		875.0		40000	
h1.16xlarge [2]	14000		1750.0		80000	
i3.large [2]	425		53.125		3000	
i3.xlarge [2]	850		106.25		6000	
i3.2xlarge [2]	1700		212.5		12000	
i3.4xlarge [2]	3500		437.5		16000	
i3.8xlarge [2]	7000		875.0		32500	
i3.16xlarge [2]	14000		1750.0		65000	
i3.metal [2]	19000		2375.0		80000	
i3en.large [1]	576	4750	72.10	593.75	3000	20000

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
i3en.xlarge [1]	1153	4750	144.20	593.75	6000	20000
i3en.2xlarge [1]	2307	4750	288.39	593.75	12000	20000
i3en.3xlarge [1]	3800	4750	475.00	593.75	15000	20000
i3en.6xlarge [2]	4750		593.75		20000	
i3en.12xlarge [2]	9500		1187.5		40000	
i3en.24xlarge [2]	19000		2375.0		80000	
i3en.metal [2]	19000		2375.0		80000	
i4g.large [1]	625	10000	78.12	1250.00	2500	40000
i4g.xlarge [1]	1250	10000	156.25	1250.00	5000	40000
i4g.2xlarge [1]	2500	10000	312.50	1250.00	10000	40000
i4g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
i4g.8xlarge [2]	10000		1250.0		40000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
i4g.16xlarge [2]	20000		2500.0		80000	
i4i.large [1]	625	10000	78.12	1250.00	2500	40000
i4i.xlarge [1]	1250	10000	156.25	1250.00	5000	40000
i4i.2xlarge [1]	2500	10000	312.50	1250.00	10000	40000
i4i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
i4i.8xlarge [2]	10000		1250.0		40000	
i4i.12xlarge [2]	15000		1875.0		60000	
i4i.16xlarge [2]	20000		2500.0		80000	
i4i.24xlarge [2]	30000		3750.0		120000	
i4i.32xlarge [2]	40000		5000.0		160000	
i4i.metal [2]	40000		5000.0		160000	
i7i.large [1]	625	10000	78.12	1250.00	2500	40000
i7i.xlarge [1]	1250	10000	156.25	1250.00	5000	40000
i7i.2xlarge [1]	2500	10000	312.50	1250.00	10000	40000

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
i7i.4xlarge[1]	5000	10000	625.00	1250.00	20000	40000
i7i.8xlarge[2]	10000		1250.0		40000	
i7i.12xlarge[2]	15000		1875.0		60000	
i7i.16xlarge[2]	20000		2500.0		80000	
i7i.24xlarge[2]	30000		3750.0		120000	
i7i.48xlarge[2]	60000		7500.0		240000	
i7i.metal-24xl[2]	30000		3750.0		120000	
i7i.metal-48xl[2]	60000		7500.0		240000	
i7ie.large [1]	625	10000	78.12	1250.00	2500	40000
i7ie.xlarge[1]	1250	10000	156.25	1250.00	5000	40000
i7ie.2xlarge[1]	2500	10000	312.50	1250.00	10000	40000
i7ie.3xlarge[1]	3750	10000	468.75	1250.00	15000	40000
i7ie.6xlarge[1]	7500	10000	937.50	1250.00	30000	40000

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
i7ie.12xlarge [2]	15000		1875.0		60000	
i7ie.18xlarge [2]	22500		2812.5		90000	
i7ie.24xlarge [2]	30000		3750.0		120000	
i7ie.48xlarge [2]	60000		7500.0		240000	
i7ie.metal-24xl [2]	30000		3750.0		120000	
i7ie.metal-48xl [2]	60000		7500.0		240000	
i8g.large [1]	625	10000	78.12	1250.00	2500	40000
i8g.xlarge [1]	1250	10000	156.25	1250.00	5000	40000
i8g.2xlarge [1]	2500	10000	312.50	1250.00	10000	40000
i8g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
i8g.8xlarge [2]	10000		1250.0		40000	
i8g.12xlarge [2]	15000		1875.0		60000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
i8g.16xlarge [2]	20000		2500.0		80000	
i8g.24xlarge [2]	30000		3750.0		120000	
i8g.48xlarge[2]	60000		7500.0		240000	
i8g.metal-24xl [2]	30000		3750.0		120000	
im4gn.large [1]	1250	10000	156.25	1250.00	5000	40000
im4gn.xlarge [1]	2500	10000	312.50	1250.00	10000	40000
im4gn.2xlarge [1]	5000	10000	625.00	1250.00	20000	40000
im4gn.4xlarge [2]	10000		1250.0		40000	
im4gn.8xlarge [2]	20000		2500.0		80000	
im4gn.16xlarge [2]	40000		5000.0		160000	
is4gen.medium [1]	625	10000	78.12	1250.00	2500	40000

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16KiB I/O)	최대 IOPS(16KiB I/O)
is4gen.large [1]	1250	10000	156.25	1250.00	5000	40000
is4gen.xlarge [1]	2500	10000	312.50	1250.00	10000	40000
is4gen.2xlarge [1]	5000	10000	625.00	1250.00	20000	40000
is4gen.4xlarge [2]	10000		1250.0		40000	
is4gen.8xlarge [2]	20000		2500.0		80000	

[1] 이러한 인스턴스는 24시간에 한 번 이상 30분 동안 최대 성능을 유지할 수 있습니다. 이후에는 기준 성능으로 돌아갑니다.

[2] 이러한 인스턴스는 명시된 성능을 무기한으로 유지할 수 있습니다. 최대 성능이 30분 이상 지속되어야 하는 워크로드가 있는 경우 이러한 인스턴스 중 하나를 사용합니다.

액셀러레이티드 컴퓨팅

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16KiB I/O)	최대 IOPS(16KiB I/O)
dl1.24xlarge [2]	19000		2375.0		80000	
dl2q.24xlarge [2]	19000		2375.0		80000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
f1.2xlarge [2]	1700		212.5		12000	
f1.4xlarge [2]	3500		437.5		44000	
f1.16xlarge [2]	14000		1750.0		75000	
f2.6xlarge [2]	7500		937.5		30000	
f2.12xlarge [2]	15000		1875.0		60000	
f2.48xlarge [2]	60000		7500.0		240000	
g3.4xlarge [2]	3500		437.5		20000	
g3.8xlarge [2]	7000		875.0		40000	
g3.16xlarge [2]	14000		1750.0		80000	
g4ad.xlarge [1]	400	3170	50.00	396.25	1700	13333
g4ad.2xlarge [1]	800	3170	100.00	396.25	3400	13333
g4ad.4xlarge [1]	1580	3170	197.50	396.25	6700	13333
g4ad.8xlarge [2]	3170		396.25		13333	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
g4ad.16xlarge [2]	6300		787.5		26667	
g4dn.xlarge [1]	950	3500	118.75	437.50	3000	20000
g4dn.2xlarge [1]	1150	3500	143.75	437.50	6000	20000
g4dn.4xlarge [2]	4750		593.75		20000	
g4dn.8xlarge [2]	9500		1187.5		40000	
g4dn.12xlarge [2]	9500		1187.5		40000	
g4dn.16xlarge [2]	9500		1187.5		40000	
g4dn.metal [2]	19000		2375.0		80000	
g5.xlarge [1]	700	3500	87.50	437.50	3000	15000
g5.2xlarge [1]	850	3500	106.25	437.50	3500	15000
g5.4xlarge [2]	4750		593.75		20000	
g5.8xlarge [2]	16000		2000.0		65000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
g5.12xlarge [2]	16000		2000.0		65000	
g5.16xlarge [2]	16000		2000.0		65000	
g5.24xlarge [2]	19000		2375.0		80000	
g5.48xlarge [2]	19000		2375.0		80000	
g5g.xlarge [1]	1188	4750	148.50	593.75	6000	20000
g5g.2xlarge [1]	2375	4750	296.88	593.75	12000	20000
g5g.4xlarge [2]	4750		593.75		20000	
g5g.8xlarge [2]	9500		1187.5		40000	
g5g.16xlarge [2]	19000		2375.0		80000	
g5g.metal [2]	19000		2375.0		80000	
g6.xlarge [1]	1000	5000	125.00	625.00	4000	20000
g6.2xlarge [1]	2000	5000	250.00	625.00	8000	20000

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
g6.4xlarge [2]	8000		1000.0		32000	
g6.8xlarge [2]	16000		2000.0		64000	
g6.12xlarge [2]	20000		2500.0		80000	
g6.16xlarge [2]	20000		2500.0		80000	
g6.24xlarge [2]	30000		3750.0		120000	
g6.48xlarge [2]	60000		7500.0		240000	
g6e.xlarge [1]	1000	5000	125.00	625.00	4000	20000
g6e.2xlarge [1]	2000	5000	250.00	625.00	8000	20000
g6e.4xlarge [2]	8000		1000.0		32000	
g6e.8xlarge [2]	16000		2000.0		64000	
g6e.12xlarge [2]	20000		2500.0		80000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
g6e.16xlarge [2]	20000		2500.0		80000	
g6e.24xlarge [2]	30000		3750.0		120000	
g6e.48xlarge [2]	60000		7500.0		240000	
gr6.4xlarge [2]	8000		1000.0		32000	
gr6.8xlarge [2]	16000		2000.0		64000	
inf1.xlarge [1]	1190	4750	148.75	593.75	4000	20000
inf1.2xlarge [1]	1190	4750	148.75	593.75	6000	20000
inf1.6xlarge [2]	4750		593.75		20000	
inf1.24xlarge [2]	19000		2375.0		80000	
inf2.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
inf2.8xlarge [2]	10000		1250.0		40000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
inf2.24xlarge [2]	30000		3750.0		120000	
inf2.48xlarge [2]	60000		7500.0		240000	
p3.2xlarge [2]	1750		218.75		10000	
p3.8xlarge [2]	7000		875.0		40000	
p3.16xlarge [2]	14000		1750.0		80000	
p3dn.24xlarge [2]	19000		2375.0		80000	
p4d.24xlarge [2]	19000		2375.0		80000	
p4de.24xlarge [2]	19000		2375.0		80000	
p5.48xlarge [2]	80000		10000.0		260000	
p5e.48xlarge [2]	80000		10000.0		260000	
p5en.48xlarge [2]	100000		12500.0		400000	

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16 KiB I/O)	최대 IOPS(16KiB I/O)
p6-b200.48xlarge [2]	100000		12500.0		400000	
trn1.2xlarge [1]	5000	20000	625.00	2500.00	16250	65000
trn1.32xlarge [2]	80000		10000.0		260000	
trn1n.32xlarge [2]	80000		10000.0		260000	
trn2.48xlarge [2]	80000		10000.0		260000	
trn2u.48xlarge [2]	80000		10000.0		260000	
vt1.3xlarge [1]	2375	4750	296.88	593.75	10000	20000
vt1.6xlarge [2]	4750		593.75		20000	
vt1.24xlarge [2]	19000		2375.0		80000	

[1] 이러한 인스턴스는 24시간에 한 번 이상 30분 동안 최대 성능을 유지할 수 있습니다. 이후에는 기준 성능으로 돌아갑니다.

[2] 이러한 인스턴스는 명시된 성능을 무기한으로 유지할 수 있습니다. 최대 성능이 30분 이상 지속되어야 하는 워크로드가 있는 경우 이러한 인스턴스 중 하나를 사용합니다.

고성능 컴퓨팅

인스턴스 크기	기준 대역폭(Mbps)	최대 대역폭(Mbps)	기준 처리량(MB/s, 128KiB I/O)	최대 처리량(MB/s, 128KiB I/O)	기준 IOPS(16KiB I/O)	최대 IOPS(16KiB I/O)
hpc6a.48xlarge [1]	87	2085	10.88	260.62	500	11000
hpc6id.32xlarge [1]	87	2085	10.88	260.62	500	11000
hpc7a.12xlarge [1]	87	2085	10.88	260.62	500	11000
hpc7a.24xlarge [1]	87	2085	10.88	260.62	500	11000
hpc7a.48xlarge [1]	87	2085	10.88	260.62	500	11000
hpc7a.96xlarge [1]	87	2085	10.88	260.62	500	11000
hpc7g.4xlarge [1]	87	2085	10.88	260.62	500	11000
hpc7g.8xlarge [1]	87	2085	10.88	260.62	500	11000
hpc7g.16xlarge [1]	87	2085	10.88	260.62	500	11000

[1] 이러한 인스턴스는 24시간에 한 번 이상 30분 동안 최대 성능을 유지할 수 있습니다. 이후에는 기준 성능으로 돌아갑니다.

[2] 이러한 인스턴스는 명시된 성능을 무기한으로 유지할 수 있습니다. 최대 성능이 30분 이상 지속되어야 하는 워크로드가 있는 경우 이러한 인스턴스 중 하나를 사용합니다.

EBS 최적화 지원됨

다음 인스턴스 유형은 EBS 최적화를 지원하지만 EBS 최적화가 기본적으로 활성화되지 않습니다. 설명된 EBS 성능 수준을 달성하려면 시작 중 또는 시작 후에 추가 시간당 요금을 지불하고 EBS 최적화를 활성화해야 합니다.

인스턴스 크기	최대 대역폭(Mbps)	최대 처리량(MB/s, 128KiB I/O)	최대 IOPS(16KiB I/O)
c1.xlarge	1000	125.0	8000
c3.xlarge	500	62.5	4000
c3.2xlarge	1000	125.0	8000
c3.4xlarge	2000	250.0	16000
i2.xlarge	500	62.5	4000
i2.2xlarge	1000	125.0	8000
i2.4xlarge	2000	250.0	16000
m1.large	500	62.5	4000
m1.xlarge	1000	125.0	8000
m2.2xlarge	500	62.5	4000
m2.4xlarge	1000	125.0	8000
m3.xlarge	500	62.5	4000
m3.2xlarge	1000	125.0	8000
r3.xlarge	500	62.5	4000
r3.2xlarge	1000	125.0	8000
r3.4xlarge	2000	250.0	16000

> **Note**
>
> `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.

Amazon EBS 최적화 성능 극대화

인스턴스 유형의 EBS 성능은 인스턴스의 성능 제한 또는 연결된 볼륨의 집계된 성능 중 작은 쪽의 제한을 받습니다. 최대한의 EBS 성능을 달성하려기 위해서는 인스턴스에 최대 인스턴스 성능과 같거나 그 이상의 결합된 성능을 제공하는 연결된 볼륨이 있어야 합니다. 예를 들어 `r6i.16xlarge`에 대해 `80,000` IOPS를 달성하려면 인스턴스에 각각 `16,000` IOPS로 프로비저닝된 볼륨이 적어도 5 gp3 이상 있어야 합니다(5 볼륨 x `16,000` IOPS = `80,000` IOPS). 애플리케이션에 필요한 것보다 많은 전용 Amazon EBS 처리량을 제공하는 인스턴스 유형을 선택하는 것이 좋습니다. 그렇게 하지 않으면 Amazon EBS와 Amazon EC2 간의 연결에 성능 제한이 발생할 수 있습니다.

> **Important**
>
> 구성 가능한 대역폭 가중치를 사용하는 경우 인스턴스의 EBS 대역폭 제한이 변경될 수 있습니다. 네트워킹 대역폭을 증가시키는 VPC-1 가중치 구성이 있는 인스턴스의 경우 IOPS 제한에 도달하기 전에 EBS 대역폭 제한에 도달하여 EBS 볼륨의 IOPS가 낮아질 수 있습니다. IOPS 저하는 특히 I/O 크기가 클수록 두드러집니다. 항상 특정 워크로드를 테스트하여 선택한 대역폭 가중치로 성능 요구 사항을 충족할 수 있는지 확인하세요. 자세한 내용은 EC2 인스턴스 대역폭 가중치 구성 섹션을 참조하세요.

`EBSIOBalance%` 및 `EBSByteBalance%` 지표를 사용하여 인스턴스 크기가 올바르게 설정되는지 여부를 확인할 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 내용은 CloudWatch를 사용하여 인스턴스 모니터링 섹션을 참조하세요.

고용량 메모리 인스턴스는 SAP HANA 인 메모리 데이터베이스의 프로덕션 배포를 비롯하여 클라우드에서 대규모 인 메모리 데이터베이스를 실행하도록 설계되었습니다. EBS 성능을 최대화하려면 동일한 프로비저닝된 성능의 `io1` 또는 `io2` 볼륨이 짝수로 포함된 고용량 메모리 인스턴스를 사용합니다. 예를 들어 IOPS가 많은 워크로드의 경우 프로비저닝된 IOPS가 40,000인 `io1` 또는 `io2` 볼륨 4개를 사

용하여 최대 160,000의 인스턴스 IOPS를 얻을 수 있습니다. 마찬가지로 처리량이 많은 워크로드의 경우 프로비저닝된 IOPS가 48,000인 `io1` 또는 `io2` 볼륨 6개를 사용하여 최대 4,750MB/s 처리량을 얻을 수 있습니다. 추가 권장 사항은 SAP HANA용 스토리지 구성을 참조하세요.

고려 사항

- 2020년 2월 26일 이후 시작된 G4dn, I3en, Inf1, M5a, M5ad, R5a, R5ad, T3, T3a 및 Z1d 인스턴스는 EBS에 최적화된 최대 성능을 제공합니다. 2020년 2월 26일 이전에 시작된 인스턴스에서 최대 성능을 얻으려면 해당 인스턴스를 중지했다가 시작합니다.
- 2019년 12월 3일 이후에 시작된 C5, C5d, C5n, M5, M5d, M5n, M5dn, R5, R5d, R5n, R5dn, 및 P3dn 인스턴스는 EBS에 최적화된 최대 성능을 제공합니다. 2019년 12월 3일 이전에 시작된 인스턴스에서 최대 성능을 얻으려면 해당 인스턴스를 중지했다가 시작합니다.
- 2020년 3월 12일 이후에 시작된 `u-6tb1.metal`, `u-9tb1.metal` 및 `u-12tb1.metal` 인스턴스는 EBS에 최적화된 최대 성능을 제공합니다. 2020년 3월 12일 이전에 시작된 이러한 유형의 인스턴스는 성능이 더 낮을 수 있습니다. 2020년 3월 12일 이전에 시작된 인스턴스에서 최대 성능을 얻으려면 계정 팀에 문의하여 추가 비용 없이 인스턴스를 업그레이드하세요.

EBS 최적화 EC2 인스턴스 유형 찾기

각 리전에서 EBS 최적화를 지원하는 인스턴스 유형을 볼 수 있습니다.

Console

기본적으로 EBS에 최적화된 인스턴스 유형을 찾으려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스 유형을 선택합니다.
3. EBS optimization support = default 필터를 추가합니다.
4. (선택 사항) 기본 설정 아이콘을 클릭한 다음 EBS 최대 IOPS 및 EBS 기준 IOPS와 같은 관련 열을 켭니다.
5. (선택 사항) 필터를 추가하여 관심 있는 특정 인스턴스 유형에 범위를 추가로 지정합니다.

AWS CLI

기본적으로 EBS에 최적화된 인스턴스 유형을 찾으려면

다음 describe-instance-types 명령을 사용합니다.

```
aws ec2 describe-instance-types \
    --filters Name=ebs-info.ebs-optimized-support,Values=default  \
    --query 'InstanceTypes[].{InstanceType:InstanceType, \
        "MaxBandwidth(Mb/s)":EbsInfo.EbsOptimizedInfo.MaximumBandwidthInMbps,\
        MaxIOPS:EbsInfo.EbsOptimizedInfo.MaximumIops,\
        "MaxThroughput(MB/s)":EbsInfo.EbsOptimizedInfo.MaximumThroughputInMBps}' \
    --output=table
```

EBS 최적화를 선택적으로 지원하는 인스턴스 유형을 찾으려면

다음 describe-instance-types 명령을 사용합니다.

```
aws ec2 describe-instance-types \
    --filters Name=ebs-info.ebs-optimized-support,Values=supported \
    --query 'InstanceTypes[].{InstanceType:InstanceType, \
        "MaxBandwidth(Mb/s)":EbsInfo.EbsOptimizedInfo.MaximumBandwidthInMbps, \
        MaxIOPS:EbsInfo.EbsOptimizedInfo.MaximumIops, \
        "MaxThroughput(MB/s)":EbsInfo.EbsOptimizedInfo.MaximumThroughputInMBps}' \
    --output=table
```

다음은 eu-west-1의 예제 출력입니다.

```
-------------------------------------------------------------------------
|                          DescribeInstanceTypes                        |
+--------------+----------------------+----------+----------------------+
| InstanceType | MaxBandwidth(Mb/s)   | MaxIOPS  |  MaxThroughput(MB/s) |
+--------------+----------------------+----------+----------------------+
|  i2.2xlarge  |  1000                |  8000    |  125.0               |
|  m2.4xlarge  |  1000                |  8000    |  125.0               |
|  m2.2xlarge  |  500                 |  4000    |  62.5                |
|  c1.xlarge   |  1000                |  8000    |  125.0               |
|  i2.xlarge   |  500                 |  4000    |  62.5                |
|  m3.xlarge   |  500                 |  4000    |  62.5                |
|  m1.xlarge   |  1000                |  8000    |  125.0               |
|  r3.4xlarge  |  2000                |  16000   |  250.0               |
|  r3.2xlarge  |  1000                |  8000    |  125.0               |
|  c3.xlarge   |  500                 |  4000    |  62.5                |
|  m3.2xlarge  |  1000                |  8000    |  125.0               |
|  r3.xlarge   |  500                 |  4000    |  62.5                |
|  i2.4xlarge  |  2000                |  16000   |  250.0               |
|  c3.4xlarge  |  2000                |  16000   |  250.0               |
|  c3.2xlarge  |  1000                |  8000    |  125.0               |
|  m1.large    |  500                 |  4000    |  62.5                |
```

```
+--------------+----------------------+----------+-----------------------+
```

PowerShell

기본적으로 EBS에 최적화된 인스턴스 유형을 찾으려면

Get-EC2InstanceType cmdlet을 사용합니다.

```
Get-EC2InstanceType `
    -Filter @{Name="ebs-info.ebs-optimized-support"; Values="default"} | `
    Select InstanceType, `
        @{Name="MaxBandwidth(Mb/s)";
 Expression={($_.EbsInfo.EbsOptimizedInfo.MaximumBandwidthInMbps)}}, `
        @{Name="MaxIOPS"; Expression={($_.EbsInfo.EbsOptimizedInfo.MaximumIops)}}, `
        @{Name="MaxThroughput (MB/s)";
 Expression={($_.EbsInfo.EbsOptimizedInfo.MaximumThroughputInMBps)}}
```

EBS 최적화를 선택적으로 지원하는 인스턴스 유형을 찾으려면

Get-EC2InstanceType cmdlet을 사용합니다.

```
Get-EC2InstanceType `
    -Filter @{Name="ebs-info.ebs-optimized-support"; Values="supported"} | `
    Select InstanceType, `
        @{Name="MaxBandwidth(Mb/s)";
 Expression={($_.EbsInfo.EbsOptimizedInfo.MaximumBandwidthInMbps)}}, `
        @{Name="MaxIOPS"; Expression={($_.EbsInfo.EbsOptimizedInfo.MaximumIops)}}, `
        @{Name="MaxThroughput (MB/s)";
 Expression={($_.EbsInfo.EbsOptimizedInfo.MaximumThroughputInMBps)}}
```

다음은 `eu-west-1`의 예제 출력입니다.

```
InstanceType MaxBandwidth(Mb/s) MaxIOPS MaxThroughput (MB/s)
------------ ------------------ ------- --------------------
m2.4xlarge                 1000    8000              125.000
i2.2xlarge                 1000    8000              125.000
c1.xlarge                  1000    8000              125.000
m2.2xlarge                  500    4000               62.500
r3.2xlarge                 1000    8000              125.000
m3.xlarge                   500    4000               62.500
r3.4xlarge                 2000   16000              250.000
m1.xlarge                  1000    8000              125.000
i2.xlarge                   500    4000               62.500
```

```
c3.xlarge                    500    4000                62.500
c3.4xlarge                  2000   16000               250.000
c3.2xlarge                  1000    8000               125.000
i2.4xlarge                  2000   16000               250.000
r3.xlarge                    500    4000                62.500
m3.2xlarge                  1000    8000               125.000
m1.large                     500    4000                62.500
```

Amazon EC2 인스턴스에 대한 EBS 최적화 활성화

선택적으로 EBS 최적화를 지원하는 이전 세대 인스턴스 유형에 대해서만 수동으로 EBS 최적화를 활성화할 수 있습니다. 이러한 인스턴스 유형에 대한 EBS 최적화를 활성화하면 시간당 추가 요금이 발생합니다.

사전 조건

- EBS 최적화 활성화가 인스턴스 유형에 필요한지 확인합니다. 자세한 내용은 EBS 최적화 지원됨 섹션을 참조하세요.
- 시작 후 EBS 최적화를 활성화하려면 인스턴스를 중지해야 합니다.

> **Warning**
>
> 인스턴스를 중지하면 인스턴스 스토어 볼륨의 데이터가 삭제됩니다. 인스턴스 스토어 볼륨의 데이터를 유지하려면 영구 스토리지에 백업하세요.

Console

시작 시 Amazon EBS 최적화를 활성화하려면

인스턴스 시작 마법사에서 필요한 인스턴스 유형을 선택합니다. 고급 세부 정보 섹션을 확장한 다음 EBS 최적화 인스턴스에서 활성화를 선택합니다.

선택한 인스턴스 유형이 Amazon EBS 최적화를 지원하지 않을 경우 드롭다운이 비활성화됩니다. 인스턴스 유형이 기본적으로 Amazon EBS에 최적화되어 있는 경우 활성화가 이미 선택되어 있습니다.

시작 이후 Amazon EBS 최적화를 활성화하려면

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

2. 탐색 창에서 인스턴스를 선택하고 인스턴스를 선택합니다.
3. 인스턴스를 중지합니다. 작업, 인스턴스 상태, 인스턴스 중지를 차례로 선택합니다.
4. 인스턴스를 선택한 상태에서 작업, 인스턴스 설정, 인스턴스 유형 변경을 차례로 선택합니다.
5. EBS 최적화를 선택한 다음 적용을 선택합니다.

   해당 인스턴스가 기본적으로 Amazon EBS에 최적화되어 있거나 Amazon EBS 최적화를 지원하지 않을 경우 확인란이 비활성화됩니다.
6. 인스턴스를 다시 시작합니다. 인스턴스 상태, 인스턴스 시작을 차례로 선택합니다.

AWS CLI

시작 시 Amazon EBS 최적화를 활성화하려면

run-instances 명령을 `--ebs-optimized` 옵션과 함께 사용합니다.

시작 이후 Amazon EBS 최적화를 활성화하려면

1. 인스턴스가 실행 중인 경우 stop-instances 명령을 사용하여 인스턴스를 중지합니다.

```
aws ec2 stop-instances --instance-ids i-1234567890abcdef0
```

2. modify-instance-attribute 명령을 `--ebs-optimized` 옵션과 함께 사용하여 EBS 최적화를 활성화합니다.

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --ebs-optimized
```

PowerShell

시작 시 Amazon EBS 최적화를 활성화하려면

New-EC2Instance cmdlet을 `-EbsOptimized` 옵션과 함께 사용합니다.

시작 이후 Amazon EBS 최적화를 활성화하려면

1. 인스턴스가 실행 중인 경우 Stop-EC2Instance cmdlet을 사용하여 인스턴스를 중지합니다.

```
Stop-EC2Instance -InstanceId i-1234567890abcdef0
```

2. Edit-EC2InstanceAttribute cmdlet을 `-EbsOptimized` 옵션과 함께 사용하여 EBS 최적화를 활성화합니다.

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -EbsOptimized $true
```

Amazon EC2 인스턴스의 CPU 옵션

많은 Amazon EC2 인스턴스가 많은 스레드를 하나의 CPU 코어에서 동시에 실행할 수 있도록 하는 동시 멀티스레딩(SMT)을 지원합니다. 각 스레드는 인스턴스에서 가상 CPU(vCPU)로 표현됩니다. 인스턴스에는 인스턴스 유형에 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**
>
> T2, C7a, M7a, R7a 및 Apple 실리콘 Mac 인스턴스 그리고 AWS Graviton 프로세서를 기반으로 하는 인스턴스의 코어당 스레드 수는 수정할 수 없습니다.

대부분의 경우 워크로드에 적합하도록 메모리와 vCPU 수를 결합한 Amazon EC2 인스턴스가 있습니다. 그러나 특정 워크로드 또는 비즈니스 요구 사항에 맞게 인스턴스를 최적화하려면 인스턴스 시작 도중 및 이후에 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 SMT를 비활성화할 수 있습니다. HPC(고성능 컴퓨팅) 워크로드와 같은 특정 워크로드에 대해 이를 수행할 수 있습니다.

요금

CPU 옵션 지정에 따른 추가 요금이나 비용 경감은 없습니다. 기존 CPU 옵션으로 시작한 인스턴스와 동일하게 청구됩니다.

내용

- Amazon EC2 인스턴스의 CPU 옵션 지정 규칙

- 지원되는 Amazon EC2 인스턴스 유형의 CPU 옵션
- Amazon EC2 인스턴스의 CPU 옵션 지정
- Amazon EC2 인스턴스의 CPU 스레드 및 코어 보기

Amazon EC2 인스턴스의 CPU 옵션 지정 규칙

인스턴스의 CPU 옵션을 지정하려면 다음 규칙을 알아야 합니다.

- 베어 메탈 인스턴스에는 CPU 옵션을 지정할 수 없습니다.
- 인스턴스 시작 도중 및 이후 모두 CPU 옵션을 지정할 수 있습니다.
- CPU 옵션을 지정하려면 요청에서 CPU 코어 수와 코어당 스레드 수를 모두 지정해야 합니다. 예제 요청은 Amazon EC2 인스턴스의 CPU 옵션 지정 단원을 참조하세요.
- 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 지원되는 Amazon EC2 인스턴스 유형의 CPU 옵션 섹션을 참조하세요.
- 하이퍼스레딩이라고도 하는 동시 멀티스레딩(SMT)을 비활성화하려면 코어당 스레드 하나를 지정하세요.
- 콘솔에서 기존 인스턴스의 인스턴스 유형을 변경하면 Amazon EC2는 가능한 경우 기존 인스턴스의 CPU 옵션 설정을 새 인스턴스에 적용합니다. 새 인스턴스 유형이 이 설정을 지원하지 않으면 CPU 옵션이 기본 CPU 옵션 사용으로 재설정됩니다. 이 옵션은 새 인스턴스 유형에 대한 기본 vCPU 수를 사용합니다.

  새 인스턴스의 설정을 업데이트하려면 인스턴스 유형 변경 보기의 고급 세부 정보에서 CPU 옵션 지정을 선택합니다.
- 지정한 CPU 옵션은 인스턴스를 중지, 시작 또는 재부팅한 후에도 유지됩니다.

지원되는 Amazon EC2 인스턴스 유형의 CPU 옵션

다음 표는 CPU 옵션 지정을 지원하는 인스턴스 유형을 나열합니다.

내용

- 범용 인스턴스
- 컴퓨팅 최적화 인스턴스
- 메모리 최적화 인스턴스

- 스토리지 최적화 인스턴스
- 액셀러레이티드 컴퓨팅 인스턴스
- 고성능 컴퓨팅 인스턴스

범용 인스턴스

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m2.xlarge	2	2	1	1, 2	1
m2.2xlarge	4	4	1	1, 2, 3, 4	1
m2.4xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
m3.large	2	1	2	1	1, 2
m3.xlarge	4	2	2	1, 2	1, 2
m3.2xlarge	8	4	2	1, 2, 3, 4	1, 2
m4.large	2	1	2	1	1, 2
m4.xlarge	4	2	2	1, 2	1, 2
m4.2xlarge	8	4	2	1, 2, 3, 4	1, 2
m4.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
m4.10xlarge	40	20	2	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
m4.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
m5.large	2	1	2	1	1, 2
m5.xlarge	4	2	2	2	1, 2
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	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
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	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	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

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
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	4, 6, 8, 10, 12, 14, 16	1, 2
m5ad.12xlarge	48	24	2	6, 12, 18, 24	1, 2
m5ad.16xlarge	64	32	2	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

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m5d.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
m5d.16xlarge	64	32	2	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	1, 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

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m5dn.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
m5dn.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
m5dn.24xlarge	96	48	2	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	1, 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

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m5n.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
m5n.24xlarge	96	48	2	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
m5zn.large	2	1	2	1	1, 2
m5zn.xlarge	4	2	2	1, 2	1, 2
m5zn.2xlarge	8	4	2	2, 4	1, 2
m5zn.3xlarge	12	6	2	2, 4, 6	1, 2
m5zn.6xlarge	24	12	2	2, 4, 6, 8, 10, 12	1, 2
m5zn.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
m6a.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
m6a.xlarge	4	2	2	1, 2	1, 2
m6a.2xlarge	8	4	2	1, 2, 3, 4	1, 2
m6a.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
m6a.8xlarge	32	16	2	4, 6, 8, 10, 12, 14, 16	1, 2
m6a.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 16, 24	1, 2
m6a.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 32	1, 2
m6a.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 32, 48	1, 2
m6a.32xlarge	128	64	2	8, 12, 16, 20, 24, 28, 32, 64	1, 2
m6a.48xlarge	192	96	2	8, 12, 16, 20, 24, 28, 32, 64, 96	1, 2
m6g.large	2	2	1	1, 2	1
m6g.xlarge	4	4	1	1, 2, 3, 4	1
m6g.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m6g.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
m6g.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1
m6g.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m6g.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
m6gd.large	2	2	1	1, 2	1
m6gd.xlarge	4	4	1	1, 2, 3, 4	1
m6gd.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
m6gd.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m6gd.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1
m6gd.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m6gd.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
m6i.large	2	1	2	1	1, 2
m6i.xlarge	4	2	2	1, 2	1, 2
m6i.2xlarge	8	4	2	2, 4	1, 2
m6i.4xlarge	16	8	2	2, 4, 6, 8	1, 2
m6i.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m6i.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
m6i.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
m6i.24xlarge	96	48	2	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
m6i.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
m6id.large	2	1	2	1	1, 2
m6id.xlarge	4	2	2	1, 2	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m6id.2xlarge	8	4	2	2, 4	1, 2
m6id.4xlarge	16	8	2	2, 4, 6, 8	1, 2
m6id.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
m6id.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
m6id.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
m6id.24xlarge	96	48	2	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

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m6id.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
m6idn.large	2	1	2	1	1, 2
m6idn.xlarge	4	2	2	1, 2	1, 2
m6idn.2xlarge	8	4	2	1, 2, 3, 4	1, 2
m6idn.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
m6idn.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
m6idn.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m6idn.16xlarge	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2
m6idn.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
m6idn.32xlarge	128	64	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
m6in.large	2	1	2	1	1, 2
m6in.xlarge	4	2	2	1, 2	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m6in.2xlarge	8	4	2	1, 2, 3, 4	1, 2
m6in.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
m6in.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
m6in.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2
m6in.16xlarge	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m6in.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
m6in.32xlarge	128	64	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
m7a.large	2	2	1	1, 2	1
m7a.xlarge	4	4	1	1, 2, 3, 4	1
m7a.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
m7a.4xlarge	16	16	1	1, 2, 4, 6, 8, 10, 12, 14, 16	1
m7a.8xlarge	32	32	1	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m7a.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 12, 18, 24, 30, 36, 42, 48	1
m7a.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 16, 24, 32, 40, 48, 56, 64	1
m7a.24xlarge	96	96	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 24, 36, 48, 60, 72, 84, 96	1
m7a.32xlarge	128	128	1	4, 6, 8, 10, 12, 14, 16, 32, 48, 64, 80, 96, 112, 128	1
m7a.48xlarge	192	192	1	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 48, 72, 96, 120, 144, 168, 192	1
m7g.large	2	2	1	1, 2	1
m7g.xlarge	4	4	1	1, 2, 3, 4	1
m7g.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m7g.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
m7g.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1
m7g.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m7g.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
m7gd.large	2	2	1	1, 2	1
m7gd.xlarge	4	4	1	1, 2, 3, 4	1
m7gd.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
m7gd.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m7gd.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1
m7gd.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m7gd.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
m7i.large	2	1	2	1	1, 2
m7i.xlarge	4	2	2	1, 2	1, 2
m7i.2xlarge	8	4	2	1, 2, 3, 4	1, 2
m7i.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m7i.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
m7i.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2
m7i.16xlarge	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m7i.24xlarge	96	48	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1, 2
m7i.48xlarge	192	96	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96	1, 2
m7i-flex.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
m7i-flex.xlarge	4	2	2	1, 2	1, 2
m7i-flex.2xlarge	8	4	2	1, 2, 3, 4	1, 2
m7i-flex.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
m7i-flex.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
m7i-flex.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2
m7i-flex.16xlarge	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2
m8g.large	2	2	1	1, 2	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m8g.xlarge	4	4	1	1, 2, 3, 4	1
m8g.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
m8g.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
m8g.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m8g.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m8g.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m8g.24xlarge	96	96	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m8g.48xlarge	192	192	1	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164, 166, 168,	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
				170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192	
m8gd.large	2	2	1	1, 2	1
m8gd.xlarge	4	4	1	1, 2, 3, 4	1
m8gd.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
m8gd.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
m8gd.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m8gd.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m8gd.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m8gd.24xlarge	96	96	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
m8gd.48xlarge	192	192	1	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164, 166, 168,	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
				170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192	
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
t3a.xlarge	4	2	2	2	1, 2
t3a.2xlarge	8	4	2	2, 4	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
t4g.nano	2	2	1	1, 2	1
t4g.micro	2	2	1	1, 2	1
t4g.small	2	2	1	1, 2	1
t4g.medium	2	2	1	1, 2	1
t4g.large	2	2	1	1, 2	1
t4g.xlarge	4	4	1	1, 2, 3, 4	1
t4g.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1

컴퓨팅 최적화 인스턴스

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c3.large	2	1	2	1	1, 2
c3.xlarge	4	2	2	1, 2	1, 2
c3.2xlarge	8	4	2	1, 2, 3, 4	1, 2
c3.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
c3.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
c4.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
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	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

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c5.24xlarge	96	48	2	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
c5a.large	2	1	2	1	1, 2
c5a.xlarge	4	2	2	1, 2	1, 2
c5a.2xlarge	8	4	2	1, 2, 3, 4	1, 2
c5a.4xlarge	16	8	2	1, 2, 3, 4, 8	1, 2
c5a.8xlarge	32	16	2	1, 2, 3, 4, 8, 12, 16	1, 2
c5a.12xlarge	48	24	2	1, 2, 3, 4, 8, 12, 16, 20, 24	1, 2
c5a.16xlarge	64	32	2	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32	1, 2
c5a.24xlarge	96	48	2	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48	1, 2
c5ad.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
c5ad.xlarge	4	2	2	1, 2	1, 2
c5ad.2xlarge	8	4	2	1, 2, 3, 4	1, 2
c5ad.4xlarge	16	8	2	1, 2, 3, 4, 8	1, 2
c5ad.8xlarge	32	16	2	1, 2, 3, 4, 8, 12, 16	1, 2
c5ad.12xlarge	48	24	2	1, 2, 3, 4, 8, 12, 16, 20, 24	1, 2
c5ad.16xlarge	64	32	2	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32	1, 2
c5ad.24xlarge	96	48	2	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48	1, 2
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

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c5d.9xlarge	36	18	2	2, 4, 6, 8, 10, 12, 14, 16, 18	1, 2
c5d.12xlarge	48	24	2	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	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

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c5n.18xlarge	72	36	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36	1, 2
c6a.large	2	1	2	1	1, 2
c6a.xlarge	4	2	2	1, 2	1, 2
c6a.2xlarge	8	4	2	1, 2, 3, 4	1, 2
c6a.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
c6a.8xlarge	32	16	2	4, 6, 8, 10, 12, 14, 16	1, 2
c6a.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 16, 24	1, 2
c6a.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 32	1, 2
c6a.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 32, 48	1, 2
c6a.32xlarge	128	64	2	8, 12, 16, 20, 24, 28, 32, 64	1, 2
c6a.48xlarge	192	96	2	8, 12, 16, 20, 24, 28, 32, 64, 96	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c6g.large	2	2	1	1, 2	1
c6g.xlarge	4	4	1	1, 2, 3, 4	1
c6g.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
c6g.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
c6g.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c6g.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c6g.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
c6gd.large	2	2	1	1, 2	1
c6gd.xlarge	4	4	1	1, 2, 3, 4	1
c6gd.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
c6gd.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c6gd.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1
c6gd.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c6gd.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
c6gn.medium	1	1	1	1	1
c6gn.large	2	2	1	1, 2	1
c6gn.xlarge	4	4	1	1, 2, 3, 4	1
c6gn.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c6gn.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
c6gn.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1
c6gn.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c6gn.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
c6i.large	2	1	2	1	1, 2
c6i.xlarge	4	2	2	1, 2	1, 2
c6i.2xlarge	8	4	2	2, 4	1, 2
c6i.4xlarge	16	8	2	2, 4, 6, 8	1, 2
c6i.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c6i.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
c6i.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
c6i.24xlarge	96	48	2	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
c6i.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
c6id.large	2	1	2	1	1, 2
c6id.xlarge	4	2	2	1, 2	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c6id.2xlarge	8	4	2	2, 4	1, 2
c6id.4xlarge	16	8	2	2, 4, 6, 8	1, 2
c6id.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
c6id.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
c6id.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
c6id.24xlarge	96	48	2	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

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c6id.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
c6in.large	2	1	2	1	1, 2
c6in.xlarge	4	2	2	1, 2	1, 2
c6in.2xlarge	8	4	2	1, 2, 3, 4	1, 2
c6in.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
c6in.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
c6in.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c6in.16xlarge	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2
c6in.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
c6in.32xlarge	128	64	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
c7a.large	2	2	1	1, 2	1
c7a.xlarge	4	4	1	1, 2, 3, 4	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c7a.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
c7a.4xlarge	16	16	1	1, 2, 4, 6, 8, 10, 12, 14, 16	1
c7a.8xlarge	32	32	1	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32	1
c7a.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 12, 18, 24, 30, 36, 42, 48	1
c7a.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 16, 24, 32, 40, 48, 56, 64	1
c7a.24xlarge	96	96	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 24, 36, 48, 60, 72, 84, 96	1
c7a.32xlarge	128	128	1	4, 6, 8, 10, 12, 14, 16, 32, 48, 64, 80, 96, 112, 128	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c7a.48xlarge	192	192	1	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 48, 72, 96, 120, 144, 168, 192	1
c7g.large	2	2	1	1, 2	1
c7g.xlarge	4	4	1	1, 2, 3, 4	1
c7g.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
c7g.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
c7g.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c7g.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c7g.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
c7gd.large	2	2	1	1, 2	1
c7gd.xlarge	4	4	1	1, 2, 3, 4	1
c7gd.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
c7gd.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c7gd.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1
c7gd.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c7gd.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
c7gn.large	2	2	1	1, 2	1
c7gn.xlarge	4	4	1	1, 2, 3, 4	1
c7gn.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
c7gn.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c7gn.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1
c7gn.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c7gn.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
c7i.large	2	1	2	1	1, 2
c7i.xlarge	4	2	2	1, 2	1, 2
c7i.2xlarge	8	4	2	1, 2, 3, 4	1, 2
c7i.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c7i.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
c7i.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2
c7i.16xlarge	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c7i.24xlarge	96	48	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1, 2
c7i.48xlarge	192	96	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96	1, 2
c7i-flex.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
c7i-flex.xlarge	4	2	2	1, 2	1, 2
c7i-flex.2xlarge	8	4	2	1, 2, 3, 4	1, 2
c7i-flex.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
c7i-flex.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
c7i-flex.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2
c7i-flex.16xlarge	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2
c8g.large	2	2	1	1, 2	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c8g.xlarge	4	4	1	1, 2, 3, 4	1
c8g.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
c8g.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
c8g.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c8g.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c8g.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c8g.24xlarge	96	96	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c8g.48xlarge	192	192	1	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164, 166, 168,	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
				170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192	
c8gd.large	2	2	1	1, 2	1
c8gd.xlarge	4	4	1	1, 2, 3, 4	1
c8gd.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
c8gd.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
c8gd.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c8gd.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c8gd.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c8gd.24xlarge	96	96	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
c8gd.48xlarge	192	192	1	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164, 166, 168,	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
				170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192	

메모리 최적화 인스턴스

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r3.large	2	1	2	1	1, 2
r3.xlarge	4	2	2	1, 2	1, 2
r3.2xlarge	8	4	2	1, 2, 3, 4	1, 2
r3.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
r3.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
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

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
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	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
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
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	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	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

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
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	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	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
r5b.large	2	1	2	1	1, 2
r5b.xlarge	4	2	2	1, 2	1, 2
r5b.2xlarge	8	4	2	2, 4	1, 2
r5b.4xlarge	16	8	2	2, 4, 6, 8	1, 2
r5b.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r5b.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
r5b.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
r5b.24xlarge	96	48	2	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
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	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	1, 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

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r5dn.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
r5dn.24xlarge	96	48	2	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	1, 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	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
r5n.24xlarge	96	48	2	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
r6a.large	2	1	2	1	1, 2
r6a.xlarge	4	2	2	1, 2	1, 2
r6a.2xlarge	8	4	2	1, 2, 3, 4	1, 2
r6a.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
r6a.8xlarge	32	16	2	4, 6, 8, 10, 12, 14, 16	1, 2
r6a.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 16, 24	1, 2
r6a.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 32	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r6a.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 32, 48	1, 2
r6a.32xlarge	128	64	2	8, 12, 16, 20, 24, 28, 32, 64	1, 2
r6a.48xlarge	192	96	2	8, 12, 16, 20, 24, 28, 32, 64, 96	1, 2
r6g.large	2	2	1	1, 2	1
r6g.xlarge	4	4	1	1, 2, 3, 4	1
r6g.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
r6g.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
r6g.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r6g.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r6g.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
r6gd.large	2	2	1	1, 2	1
r6gd.xlarge	4	4	1	1, 2, 3, 4	1
r6gd.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
r6gd.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r6gd.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1
r6gd.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r6gd.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
r6i.large	2	1	2	1	1, 2
r6i.xlarge	4	2	2	1, 2	1, 2
r6i.2xlarge	8	4	2	2, 4	1, 2
r6i.4xlarge	16	8	2	2, 4, 6, 8	1, 2
r6i.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r6i.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
r6i.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
r6i.24xlarge	96	48	2	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
r6i.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
r6idn.large	2	1	2	1	1, 2
r6idn.xlarge	4	2	2	1, 2	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r6idn.2xlarge	8	4	2	1, 2, 3, 4	1, 2
r6idn.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
r6idn.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
r6idn.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2
r6idn.16xlarge	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r6idn.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
r6idn.32xlarge	128	64	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
r6in.large	2	1	2	1	1, 2
r6in.xlarge	4	2	2	1, 2	1, 2
r6in.2xlarge	8	4	2	1, 2, 3, 4	1, 2
r6in.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
r6in.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r6in.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2
r6in.16xlarge	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2
r6in.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

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r6in.32xlarge	128	64	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
r6id.large	2	1	2	1	1, 2
r6id.xlarge	4	2	2	1, 2	1, 2
r6id.2xlarge	8	4	2	2, 4	1, 2
r6id.4xlarge	16	8	2	2, 4, 6, 8	1, 2
r6id.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
r6id.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
r6id.16xlarge	64	32	2	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
r6id.24xlarge	96	48	2	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
r6id.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
r7a.large	2	2	1	1, 2	1
r7a.xlarge	4	4	1	1, 2, 3, 4	1
r7a.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
r7a.4xlarge	16	16	1	1, 2, 4, 6, 8, 10, 12, 14, 16	1
r7a.8xlarge	32	32	1	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r7a.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 12, 18, 24, 30, 36, 42, 48	1
r7a.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 16, 24, 32, 40, 48, 56, 64	1
r7a.24xlarge	96	96	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 24, 36, 48, 60, 72, 84, 96	1
r7a.32xlarge	128	128	1	4, 6, 8, 10, 12, 14, 16, 32, 48, 64, 80, 96, 112, 128	1
r7a.48xlarge	192	192	1	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 48, 72, 96, 120, 144, 168, 192	1
r7g.large	2	2	1	1, 2	1
r7g.xlarge	4	4	1	1, 2, 3, 4	1
r7g.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r7g.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
r7g.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1
r7g.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r7g.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
r7gd.large	2	2	1	1, 2	1
r7gd.xlarge	4	4	1	1, 2, 3, 4	1
r7gd.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
r7gd.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r7gd.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1
r7gd.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r7gd.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
r7i.large	2	1	2	1	1, 2
r7i.xlarge	4	2	2	1, 2	1, 2
r7i.2xlarge	8	4	2	1, 2, 3, 4	1, 2
r7i.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r7i.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
r7i.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2
r7i.16xlarge	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r7i.24xlarge	96	48	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1, 2
r7i.48xlarge	192	96	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96	1, 2
r7iz.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
r7iz.xlarge	4	2	2	1, 2	1, 2
r7iz.2xlarge	8	4	2	1, 2, 3, 4	1, 2
r7iz.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
r7iz.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
r7iz.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2
r7iz.16xlarge	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r7iz.32xlarge	128	64	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
r8g.large	2	2	1	1, 2	1
r8g.xlarge	4	4	1	1, 2, 3, 4	1
r8g.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
r8g.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
r8g.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r8g.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r8g.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r8g.24xlarge	96	96	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r8g.48xlarge	192	192	1	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164, 166, 168,	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
				170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192	
r8gd.large	2	2	1	1, 2	1
r8gd.xlarge	4	4	1	1, 2, 3, 4	1
r8gd.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
r8gd.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
r8gd.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r8gd.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r8gd.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r8gd.24xlarge	96	96	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
r8gd.48xlarge	192	192	1	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164, 166, 168,	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
				170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192	
u-3tb1.56xlarge	224	112	2	8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64, 68, 72, 76, 80, 84, 88, 92, 96, 100, 104, 108, 112	1, 2
u-6tb1.56xlarge	224	224	1	16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
u-6tb1.112xlarge	448	224	2	16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224	1, 2
u-9tb1.112xlarge	448	224	2	16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
u-12tb1.112xlarge	448	224	2	16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224	1, 2
u-18tb1.112xlarge	448	224	2	16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
u-24tb1.112xlarge	448	224	2	16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
u7i-6tb.112xlarge	448	224	2	8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64, 68, 72, 76, 80, 84, 88, 92, 96, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140, 144, 148, 152, 156, 160, 164, 168, 172, 176, 180, 184, 188, 192, 196, 200, 204, 208, 212, 216, 220, 224	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
u7i-8tb.112xlarge	448	224	2	8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64, 68, 72, 76, 80, 84, 88, 92, 96, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140, 144, 148, 152, 156, 160, 164, 168, 172, 176, 180, 184, 188, 192, 196, 200, 204, 208, 212, 216, 220, 224	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
u7i-12tb.224xlarge	896	448	2	16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224, 232, 240, 248, 256, 264, 272, 280, 288, 296, 304, 312, 320, 328, 336, 344, 352, 360, 368, 376, 384, 392, 400, 408, 416, 424, 432, 440, 448	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
u7in-16tb.224xlarge	896	448	2	16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224, 232, 240, 248, 256, 264, 272, 280, 288, 296, 304, 312, 320, 328, 336, 344, 352, 360, 368, 376, 384, 392, 400, 408, 416, 424, 432, 440, 448	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
u7in-24tb.224xlarge	896	448	2	16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224, 232, 240, 248, 256, 264, 272, 280, 288, 296, 304, 312, 320, 328, 336, 344, 352, 360, 368, 376, 384, 392, 400, 408, 416, 424, 432, 440, 448	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
u7in-32tb.224xlarge	896	448	2	16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224, 232, 240, 248, 256, 264, 272, 280, 288, 296, 304, 312, 320, 328, 336, 344, 352, 360, 368, 376, 384, 392, 400, 408, 416, 424, 432, 440, 448	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
u7inh-32tb.480xlarge	1920	960	2	32, 48, 64, 80, 96, 112, 128, 144, 160, 176, 192, 208, 224, 240, 256, 272, 288, 304, 320, 336, 352, 368, 384, 400, 416, 432, 448, 464, 480, 496, 512, 528, 544, 560, 576, 592, 608, 624, 640, 656, 672, 688, 704, 720, 736, 752, 768, 784, 800, 816, 832, 848, 864, 880, 896, 912, 928, 944, 960	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
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

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
x1e.32xlarge	128	64	2	4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64	1, 2
x2gd.large	2	2	1	1, 2	1
x2gd.xlarge	4	4	1	1, 2, 3, 4	1
x2gd.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
x2gd.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
x2gd.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
x2gd.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
x2gd.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
x2idn.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
x2idn.24xlarge	96	48	2	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

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
x2idn.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
x2iedn.xlarge	4	2	2	1, 2	1, 2
x2iedn.2xlarge	8	4	2	2, 4	1, 2
x2iedn.4xlarge	16	8	2	2, 4, 6, 8	1, 2
x2iedn.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
x2iedn.16xlarge	64	32	2	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
x2iedn.24xlarge	96	48	2	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
x2iedn.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
x2iezn.2xlarge	8	4	2	2, 4	1, 2
x2iezn.4xlarge	16	8	2	2, 4, 6, 8	1, 2
x2iezn.6xlarge	24	12	2	2, 4, 6, 8, 10, 12	1, 2
x2iezn.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
x2iezn.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
x8g.large	2	2	1	1, 2	1
x8g.xlarge	4	4	1	1, 2, 3, 4	1
x8g.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
x8g.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
x8g.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
x8g.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
x8g.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
x8g.24xlarge	96	96	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
x8g.48xlarge	192	192	1	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164, 166, 168,	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
				170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192	
z1d.large	2	1	2	1	1, 2
z1d.xlarge	4	2	2	1, 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
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

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
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
d3.xlarge	4	2	2	1, 2	1, 2
d3.2xlarge	8	4	2	2, 4	1, 2
d3.4xlarge	16	8	2	2, 4, 6, 8	1, 2
d3.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
d3en.xlarge	4	2	2	1, 2	1, 2
d3en.2xlarge	8	4	2	2, 4	1, 2
d3en.4xlarge	16	8	2	2, 4, 6, 8	1, 2
d3en.6xlarge	24	12	2	2, 4, 6, 8, 10, 12	1, 2
d3en.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
d3en.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
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
i2.xlarge	4	2	2	1, 2	1, 2
i2.2xlarge	8	4	2	1, 2, 3, 4	1, 2
i2.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
i2.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	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

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
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	1, 2	1, 2
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

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
i3en.24xlarge	96	48	2	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
i4g.large	2	2	1	1, 2	1
i4g.xlarge	4	4	1	1, 2, 3, 4	1
i4g.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
i4g.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
i4g.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
i4g.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
i4i.large	2	1	2	1	1, 2
i4i.xlarge	4	2	2	1, 2	1, 2
i4i.2xlarge	8	4	2	1, 2, 3, 4	1, 2
i4i.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
i4i.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
i4i.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2
i4i.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
i4i.24xlarge	96	48	2	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

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
i4i.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
i7i.large	2	1	2	1	1, 2
i7i.xlarge	4	2	2	1, 2	1, 2
i7i.2xlarge	8	4	2	1, 2, 3, 4	1, 2
i7i.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
i7i.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
i7i.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
i7i.16xlarge	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2
i7i.24xlarge	96	48	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
i7i.48xlarge	192	96	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96	1, 2
i7ie.large	2	1	2	1	1, 2
i7ie.xlarge	4	2	2	1, 2	1, 2
i7ie.2xlarge	8	4	2	1, 2, 3, 4	1, 2
i7ie.3xlarge	12	6	2	1, 2, 3, 4, 5, 6	1, 2
i7ie.6xlarge	24	12	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
i7ie.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2
i7ie.18xlarge	72	36	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
i7ie.24xlarge	96	48	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1, 2
i7ie.48xlarge	192	96	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96	1, 2
i8g.large	2	2	1	1, 2	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
i8g.xlarge	4	4	1	1, 2, 3, 4	1
i8g.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
i8g.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
i8g.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
i8g.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
i8g.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
i8g.24xlarge	96	96	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
i8g.48xlarge	192	192	1	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164, 166, 168,	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
				170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192	
im4gn.large	2	2	1	1, 2	1
im4gn.xlarge	4	4	1	1, 2, 3, 4	1
im4gn.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
im4gn.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
im4gn.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
im4gn.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
is4gen.medium	1	1	1	1	1
is4gen.large	2	2	1	1, 2	1
is4gen.xlarge	4	4	1	1, 2, 3, 4	1
is4gen.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
is4gen.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
is4gen.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1

액셀러레이티드 컴퓨팅 인스턴스

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
dl1.24xlarge	96	48	2	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
dl2q.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34,	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
				36, 38, 40, 42, 44, 46, 48	
f1.2xlarge	8	4	2	1, 2, 3, 4	1, 2
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
f2.6xlarge	24	12	2	1, 2, 3, 6, 9, 12	1, 2
f2.12xlarge	48	24	2	1, 2, 3, 6, 9, 12, 15, 18, 21, 24	1, 2
f2.48xlarge	192	96	2	4, 6, 8, 10, 12, 24, 36, 48, 60, 72, 84, 96	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

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
g3.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
g4ad.xlarge	4	2	2	2	1, 2
g4ad.2xlarge	8	4	2	2, 4	1, 2
g4ad.4xlarge	16	8	2	2, 4, 8	1, 2
g4ad.8xlarge	32	16	2	2, 4, 8, 16	1, 2
g4ad.16xlarge	64	32	2	2, 4, 8, 16, 32	1, 2
g4dn.xlarge	4	2	2	2	1, 2
g4dn.2xlarge	8	4	2	2, 4	1, 2
g4dn.4xlarge	16	8	2	2, 4, 6, 8	1, 2
g4dn.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
g4dn.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
g4dn.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
g5g.xlarge	4	4	1	1, 2, 3, 4	1
g5g.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
g5g.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
g5g.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
g5g.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
g6.xlarge	4	2	2	1, 2	1, 2
g6.2xlarge	8	4	2	1, 2, 3, 4	1, 2
g6.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
g6.8xlarge	32	16	2	1, 2, 4, 6, 8, 10, 12, 14, 16	1, 2
g6.12xlarge	48	24	2	1, 2, 3, 6, 9, 12, 15, 18, 21, 24	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
g6.16xlarge	64	32	2	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32	1, 2
g6.24xlarge	96	48	2	1, 2, 3, 4, 5, 6, 12, 18, 24, 30, 36, 42, 48	1, 2
g6.48xlarge	192	96	2	4, 6, 8, 10, 12, 24, 36, 48, 60, 72, 84, 96	1, 2
g6e.xlarge	4	2	2	1, 2	1, 2
g6e.2xlarge	8	4	2	1, 2, 3, 4	1, 2
g6e.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
g6e.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
g6e.12xlarge	48	24	2	3, 6, 9, 12, 15, 18, 21, 24	1, 2
g6e.16xlarge	64	32	2	4, 8, 12, 16, 20, 24, 28, 32	1, 2
g6e.24xlarge	96	48	2	6, 12, 18, 24, 30, 36, 42, 48	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
g6e.48xlarge	192	96	2	12, 24, 36, 48, 60, 72, 84, 96	1, 2
gr6.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
gr6.8xlarge	32	16	2	1, 2, 4, 6, 8, 10, 12, 14, 16	1, 2
inf1.xlarge	4	2	2	2	1, 2
inf1.2xlarge	8	4	2	2, 4	1, 2
inf1.6xlarge	24	12	2	2, 4, 6, 8, 10, 12	1, 2
inf1.24xlarge	96	48	2	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
inf2.xlarge	4	2	2	1, 2	1, 2
inf2.8xlarge	32	16	2	4, 6, 8, 10, 12, 14, 16	1, 2
inf2.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 32, 48	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
inf2.48xlarge	192	96	2	4, 8, 12, 16, 20, 24, 28, 32, 64, 96	1, 2
p3.2xlarge	8	4	2	1, 2, 3, 4	1, 2
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
p4d.24xlarge	96	48	2	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

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
p4de.24xlarge	96	48	2	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
p5.48xlarge	192	96	2	12, 24, 36, 48, 60, 72, 84, 96	1, 2
p5e.48xlarge	192	96	2	12, 24, 36, 48, 60, 72, 84, 96	1, 2
p5en.48xlarge	192	96	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
p6-b200.48xlarge	192	96	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96	1, 2
trn1.2xlarge	8	4	2	2, 4	1, 2
trn1.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
trn1n.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
trn2.48xlarge	192	96	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96	1, 2

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
trn2u.48xlarge	192	96	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96	1, 2
vt1.3xlarge	12	6	2	6	1, 2
vt1.6xlarge	24	12	2	6, 12	1, 2
vt1.24xlarge	96	48	2	6, 12, 48	1, 2

고성능 컴퓨팅 인스턴스

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
hpc6id.32xlarge	64	64	1	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28,	1

인스턴스 유형	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
				30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	

Amazon EC2 인스턴스의 CPU 옵션 지정

인스턴스 시작 도중 또는 이후에 CPU 옵션을 지정할 수 있습니다.

업무

- 동시 멀티스레딩 비활성화
- 시작 시 vCPU 사용자 지정 수 지정
- 시작 템플릿에 vCPU 사용자 지정 수 지정
- EC2 인스턴스의 CPU 옵션 변경

동시 멀티스레딩 비활성화

하이퍼스레딩이라고도 하는 동시 멀티스레딩(SMT)을 비활성화하려면 코어당 스레드 하나를 지정하세요.

Console

인스턴스 시작 중 SMT 비활성화

1. 콘솔에서 인스턴스 시작 마법사를 사용하여 EC2 인스턴스 시작 절차를 수행하고 필요에 따라 인스턴스를 구성합니다.
2. 고급 세부 정보를 확장하고 CPU 옵션 지정 확인란을 선택합니다.
3. 코어 수에 대해 필요한 CPU 코어 수를 선택합니다. 이 예에서 `r5.4xlarge` 인스턴스에 필요한 기본 CPU 코어 개수를 지정하려면 8을 선택합니다.
4. SMT를 비활성화하려면 코어당 스레드로 1을 선택하세요.

5. Summary(요약) 패널에서 인스턴스 구성을 검토한 다음 Launch instance(인스턴스 시작)를 선택합니다. 자세한 내용은 콘솔에서 인스턴스 시작 마법사를 사용하여 EC2 인스턴스 시작 섹션을 참조하세요.

AWS CLI

인스턴스 시작 중 SMT 비활성화

run-instances AWS CLI 명령을 사용하여 1 파라미터의 ThreadsPerCore에 값을 --cpu-options로 지정합니다. CoreCount에 대해 CPU 코어 수를 지정합니다. 이 예에서 r7i.4xlarge 인스턴스에 필요한 기본 CPU 코어 개수를 지정하려면 값을 8로 지정합니다.

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type r7i.4xlarge \
    --cpu-options "CoreCount=8,ThreadsPerCore=1" \
    --key-name my-key-pair
```

PowerShell

인스턴스 시작 중 SMT 비활성화

New-EC2Instance 명령을 사용하고 -CpuOptions 파라미터의 ThreadsPerCore에 값을 1로 지정합니다. CoreCount에 대해 CPU 코어 수를 지정합니다. 이 예에서 r7i.4xlarge 인스턴스에 필요한 기본 CPU 코어 개수를 지정하려면 값을 8로 지정합니다.

```
New-EC2Instance `
    -ImageId 'ami-0abcdef1234567890' `
    -InstanceType 'r7i.4xlarge' `
    -CpuOptions @{CoreCount=8; ThreadsPerCore=1} `
    -KeyName 'my-key-pair'
```

> **Note**
>
> 기존 인스턴스에 대해 SMT를 비활성화하려면 EC2 인스턴스의 CPU 옵션 변경에 표시된 프로세스에 따라 코어당 실행되는 스레드 수를 1로 변경합니다.

시작 시 vCPU 사용자 지정 수 지정

EC2 콘솔 또는 AWS CLI에서 인스턴스를 시작할 때 인스턴스의 코어당 CPU 코어와 스레드 수를 사용자 지정할 수 있습니다. 이 섹션의 예제는 `r5.4xlarge` 인스턴스 유형을 사용하며 다음과 같은 기본 설정을 포함합니다.

- CPU 코어: 8
- 코어당 스레드: 2

기본적으로 인스턴스 유형에 사용할 수 있는 최대 수의 vCPU를 사용하여 인스턴스를 시작합니다. 이 인스턴스 유형의 경우 총 16개의 vCPU(각각 2개의 스레드를 실행하는 코어 8개)입니다. 이 인스턴스 유형에 대한 자세한 내용은 메모리 최적화 인스턴스 섹션을 참조하세요.

다음 예제는 4개의 vCPU로 `r5.4xlarge` 인스턴스를 시작합니다.

Console

인스턴스 시작 중 vCPU 수 사용자 지정

1. 콘솔에서 인스턴스 시작 마법사를 사용하여 EC2 인스턴스 시작 절차를 수행하고 필요에 따라 인스턴스를 구성합니다.
2. 고급 세부 정보를 확장하고 CPU 옵션 지정 확인란을 선택합니다.
3. 4개의 vCPU를 얻기 위해 다음과 같이 2개의 CPU 코어와 코어당 2개의 스레드를 지정합니다.
   - 코어 수로 2를 선택합니다.
   - 코어당 스레드로 2를 선택합니다.
4. Summary(요약) 패널에서 인스턴스 구성을 검토한 다음 Launch instance(인스턴스 시작)를 선택합니다. 자세한 내용은 콘솔에서 인스턴스 시작 마법사를 사용하여 EC2 인스턴스 시작 섹션을 참조하세요.

AWS CLI

인스턴스 시작 중 vCPU 수 사용자 지정

run-instances AWS CLI 명령을 사용하여 `--cpu-options` 파라미터에서 CPU 코어 수와 스레드 수를 지정합니다. 4개의 vCPU를 얻기 위해 2개의 CPU 코어와 코어당 2개의 스레드를 지정할 수 있습니다.

```
aws ec2 run-instances \
```

```
    --image-id ami-0abcdef1234567890 \
    --instance-type r7i.4xlarge \
    --cpu-options "CoreCount=2,ThreadsPerCore=2" \
    --key-name my-key-pair
```

또는 4개의 vCPU를 얻기 위해 4개의 CPU 코어와 코어당 1개의 스레드(SMT 비활성화)를 지정합니다.

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type r7i.4xlarge \
    --cpu-options "CoreCount=4,ThreadsPerCore=1" \
    --key-name my-key-pair
```

PowerShell

인스턴스 시작 중 vCPU 수 사용자 지정

New-EC2Instance 명령을 사용하고 `-CpuOptions` 파라미터에서 CPU 코어 수와 스레드 수를 지정합니다. 4개의 vCPU를 얻기 위해 2개의 CPU 코어와 코어당 2개의 스레드를 지정할 수 있습니다.

```
New-EC2Instance `
    -ImageId 'ami-0abcdef1234567890' `
    -InstanceType 'r7i.4xlarge' `
    -CpuOptions @{CoreCount=2; ThreadsPerCore=2} `
    -KeyName 'my-key-pair'
```

또는 4개의 vCPU를 얻기 위해 4개의 CPU 코어와 코어당 1개의 스레드(SMT 비활성화)를 지정합니다.

```
New-EC2Instance `
    -ImageId 'ami-0abcdef1234567890' `
    -InstanceType 'r7i.4xlarge' `
    -CpuOptions @{CoreCount=4; ThreadsPerCore=1} `
    -KeyName 'my-key-pair'
```

시작 템플릿에 vCPU 사용자 지정 수 지정

시작 템플릿에서 인스턴스의 코어당 CPU 코어와 스레드 수를 사용자 지정할 수 있습니다. 이 섹션의 예제는 `r5.4xlarge` 인스턴스 유형을 사용하며 다음과 같은 기본 설정을 포함합니다.

- CPU 코어: 8
- 코어당 스레드: 2

기본적으로 인스턴스 유형에 사용할 수 있는 최대 수의 vCPU를 사용하여 인스턴스를 시작합니다. 이 인스턴스 유형의 경우 총 16개의 vCPU(각각 2개의 스레드를 실행하는 코어 8개)입니다. 이 인스턴스 유형에 대한 자세한 내용은 메모리 최적화 인스턴스 섹션을 참조하세요.

다음 예제에서는 vCPU가 4개 있는 `r5.4xlarge` 인스턴스의 구성을 지정하는 시작 템플릿을 생성합니다.

Console

시작 템플릿에 vCPU 사용자 지정 수 지정

1. 파라미터를 지정하여 시작 템플릿 생성 절차를 수행하고 필요에 따라 시작 템플릿을 구성합니다.
2. 고급 세부 정보를 확장하고 CPU 옵션 지정 확인란을 선택합니다.
3. 4개의 vCPU를 얻기 위해 다음과 같이 2개의 CPU 코어와 코어당 2개의 스레드를 지정합니다.
   - 코어 수로 2를 선택합니다.
   - 코어당 스레드로 2를 선택합니다.
4. 요약 패널에서 인스턴스 구성을 검토한 다음 인스턴스 시작을 선택합니다. 자세한 내용은 Amazon EC2 시작 템플릿에 인스턴스 시작 파라미터 저장 섹션을 참조하세요.

AWS CLI

시작 템플릿에 vCPU 사용자 지정 수 지정

create-launch-template AWS CLI 명령을 사용하여 `CpuOptions` 파라미터에서 CPU 코어 수와 스레드 수를 지정합니다. 4개의 vCPU를 얻기 위해 2개의 CPU 코어와 코어당 2개의 스레드를 지정할 수 있습니다.

```
aws ec2 create-launch-template \
    --launch-template-name TemplateForCPUOptions \
    --version-description CPUOptionsVersion1 \
    --launch-template-data file://template-data.json
```

다음은 이 예제의 인스턴스 구성에 대한 CPU 옵션 등 시작 템플릿 데이터를 포함하는 JSON 파일 예제입니다.

```
{
    "NetworkInterfaces": [{
        "AssociatePublicIpAddress": true,
        "DeviceIndex": 0,
        "Ipv6AddressCount": 1,
        "SubnetId": "subnet-0abcdef1234567890"
    }],
    "ImageId": "ami-0abcdef1234567890",
    "InstanceType": "r5.4xlarge",
    "TagSpecifications": [{
        "ResourceType": "instance",
        "Tags": [{
            "Key":"Name",
            "Value":"webserver"
        }]
    }],
    "CpuOptions": {
        "CoreCount":2,
        "ThreadsPerCore":2
    }
}
```

또는 4개의 vCPU를 얻기 위해 4개의 CPU 코어와 코어당 1개의 스레드(SMT 비활성화)를 지정합니다.

```
{
    "NetworkInterfaces": [{
        "AssociatePublicIpAddress": true,
        "DeviceIndex": 0,
        "Ipv6AddressCount": 1,
        "SubnetId": "subnet-0abcdef1234567890"
    }],
    "ImageId": "ami-0abcdef1234567890",
    "InstanceType": "r5.4xlarge",
    "TagSpecifications": [{
        "ResourceType": "instance",
        "Tags": [{
            "Key":"Name",
            "Value":"webserver"
        }]
```

```
    }],
    "CpuOptions": {
        "CoreCount":4,
        "ThreadsPerCore":1
    }
}
```

PowerShell

시작 템플릿에 vCPU 사용자 지정 수 지정

New-EC2LaunchTemplate을 사용합니다.

```
New-EC2LaunchTemplate `
    -LaunchTemplateName 'TemplateForCPUOptions' `
    -VersionDescription 'CPUOptionsVersion1' `
    -LaunchTemplateData (Get-Content -Path 'template-data.json' | ConvertFrom-Json)
```

EC2 인스턴스의 CPU 옵션 변경

시간이 지나면서 요구 사항이 변하면 기존 인스턴스의 CPU 옵션 구성을 변경하는 것이 좋습니다. 인스턴스에서 실행되는 각 스레드를 가상 CPU(vCPU)라고 합니다. Amazon EC2 콘솔, AWS CLI, API 또는 SDK에서 기존 인스턴스에서 실행되는 vCPU 수를 변경할 수 있습니다. 이렇게 변경하려면 먼저 인스턴스 상태가 `Stopped`이어야 합니다.

Console

인스턴스의 활성 vCPU 수를 변경하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 인스턴스를 선택하고 인스턴스를 선택합니다.
3. 인스턴스가 실행 중인 경우 진행하기 전에 인스턴스를 중지해야 합니다. 인스턴스 상태, 인스턴스 중지를 차례로 선택합니다.
4. vCPU 구성을 변경하려면 작업 , 인스턴스 설정, CPU 옵션 변경을 선택합니다.
5. 다음 CPU 옵션 중 하나를 선택합니다.
   - 인스턴스를 해당 인스턴스 유형의 기본 vCPU 수로 재설정하려면 기본 CPU 옵션 사용을 선택합니다. 기본값은 모든 CPU 코어에 대해 모든 스레드를 실행하는 것입니다.
   - 인스턴스의 vCPU 수를 구성하려면 CPU 옵션 지정을 선택합니다.

6. CPU 옵션 지정을 선택하면 활성 vCPU 필드가 표시됩니다.

   a. 첫 번째 선택기를 사용하여 각 CPU 코어의 스레드 수를 구성합니다. 동시 멀티스레딩을 비활성화하려면 1을 선택합니다.

   b. 두 번째 선택기를 사용하여 인스턴스의 CPU 수를 구성합니다.

   CPU 옵션 선택기를 변경하면 다음 필드가 동적으로 업데이트됩니다.

   - 활성 vCPU: 사용자의 선택을 기반으로 코어당 스레드를 곱한 CPU 코어 수입니다. 예를 들어 스레드 2개와 코어 4개를 선택한 경우 vCPU 8개가 됩니다.
   - 총 vCPU: 인스턴스 유형의 최대 vCPU 수입니다. 예를 들어 `m6i.4xlarge` 인스턴스 유형의 경우 16개의 vCPU(각각 2개의 스레드를 실행하는 코어 8개)입니다.

7. 변경을 선택합니다.

AWS CLI

인스턴스의 활성 vCPU 수를 변경하려면

modify-instance-cpu-options 명령을 사용하여 `--core-count` 파라미터에서 실행되는 CPU 코어 수와 `--threads-per-core` 파라미터에서 코어당 실행되는 스레드 수를 지정합니다.

다음 예제는 지정된 인스턴스에서 vCPU 8개를 실행할 수 있는 `m6i.4xlarge` 인스턴스 유형의 두 가지 구성을 보여줍니다. 이 인스턴스 유형의 기본값은 16개의 vCPU(각각 2개의 스레드를 실행하는 코어 8개)입니다.

예제 1: 코어당 스레드 2개로 CPU 코어 4개를 실행하여 총 8개의 vCPU를 실행합니다.

```
aws ec2 modify-instance-cpu-options \
    --instance-id 1234567890abcdef0 \
    --core-count=4 \
    --threads-per-core=2
```

예제 2: 동시 멀티스레딩을 사용하지 않도록 설정하려면 코어당 실행되는 스레드 수를 1로 변경할 수 있습니다. 또한 결과 구성에서는 총 8개의 vCPU(CPU 코어 8개, 코어당 스레드 1개)를 실행합니다.

```
aws ec2 modify-instance-cpu-options \
    --instance-id 1234567890abcdef0 \
    --core-count=8 \
```

```
    --threads-per-core=1
```

PowerShell

인스턴스의 활성 vCPU 수를 변경하려면

Edit-EC2InstanceCpuOption cmdlet을 사용하고 `-CoreCount` 파라미터에서 실행되는 CPU 코어 수와 `ThreadsPerCore` 파라미터에서 코어당 실행되는 스레드 수를 지정합니다.

예제 1: 코어당 스레드 2개로 CPU 코어 4개를 실행하여 총 8개의 vCPU를 실행합니다.

```
Edit-EC2InstanceCpuOption `
    -InstanceId 'i-1234567890abcdef0' `
    -CoreCount 4 `
    -ThreadsPerCore 2
```

예제 2: 동시 멀티스레딩을 사용하지 않도록 설정하려면 코어당 실행되는 스레드 수를 1로 변경할 수 있습니다. 또한 결과 구성에서는 총 8개의 vCPU(CPU 코어 8개, 코어당 스레드 1개)를 실행합니다.

```
Edit-EC2InstanceCpuOption `
    -InstanceId 'i-1234567890abcdef0' `
    -CoreCount 8 `
    -ThreadsPerCore 1
```

Amazon EC2 인스턴스의 CPU 스레드 및 코어 보기

인스턴스를 설명하여 기존 인스턴스의 CPU 옵션을 볼 수 있습니다.

Console

인스턴스의 CPU 옵션을 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 인스턴스를 선택하고 인스턴스를 선택합니다.
3. 세부 정보 탭의 호스트 및 배치 그룹에서 vCPU 수를 찾습니다.

AWS CLI

인스턴스의 CPU 옵션을 보려면

아래와 같이 describe-instances 명령을 사용합니다.

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

출력의 예시는 다음과 같습니다. CoreCount 필드는 인스턴스의 코어 수를 나타냅니다. ThreadsPerCore 필드는 코어당 스레드 수를 나타냅니다.

```
[
    {
        "CoreCount": 24,
        "ThreadsPerCore": 2
    },
]
```

PowerShell

인스턴스의 CPU 옵션을 보려면

Get-EC2Instance cmdlet을 사용합니다.

```
(Get-EC2Instance `
    -InstanceId 'i-1234567890abcdef0').Instances.CpuOptions
```

출력의 예시는 다음과 같습니다.

```
AmdSevSnp CoreCount ThreadsPerCore
--------- --------- --------------
          24        2
```

또는 인스턴스에 연결하여 다음 시스템 도구 중 하나를 사용하여 CPU 정보를 볼 수 있습니다.

- Windows 인스턴스의 Windows Task Manager
- Linux 인스턴스의 lscpu 명령

AWS Config를 사용하여 종료된 인스턴스를 포함한 인스턴스의 구성 변경을 기록, 액세스, 감사 및 평가할 수 있습니다. 자세한 내용은 AWS Config 개발자 가이드에서 AWS Config 시작하기를 참조하세요.

Amazon EC2 인스턴스용 AMD SEV-SNP

AMD SEV-SNP(AMD Secure Encrypted Virtualization-Secure Nested Paging)는 CPU 기능으로, 다음과 같은 속성을 제공합니다.

- 증명 - AMD SEV-SNP를 사용하면 인스턴스의 상태와 ID를 검증하는 데 사용할 수 있는 암호화 측정값이 포함된 서명된 증명 보고서를 검색하고, 정품 AMD 하드웨어에서 실행되고 있음을 확인할 수 있습니다. 자세한 내용은 AMD SEV-SNP로 Amazon EC2 인스턴스 증명 섹션을 참조하세요.
- 메모리 암호화 - AMD EPYC(Milan), AWS Graviton2 및 인텔 제온 스케일러블(Ice Lake) 프로세서부터 인스턴스 메모리는 항상 암호화됩니다. AMD SEV-SNP에 대해 활성화된 인스턴스는 메모리 암호화에 인스턴스별 키를 사용합니다.

내용

- 개념 및 용어
- 요구 사항
- 고려 사항
- 요금
- AMD SEV-SNP를 지원하는 EC2 인스턴스 유형 찾기
- EC2 인스턴스에서 AMD SEV-SNP 활성화
- AMD SEV-SNP로 Amazon EC2 인스턴스 증명

개념 및 용어

AMD SEV-SNP를 사용하기 전에 다음 개념과 용어를 숙지해야 합니다.

AMD SEV-SNP 증명 보고서

AMD SEV-SNP 증명 보고서는 인스턴스가 CPU에 요청할 수 있는 문서입니다. AMD SEV-SNP 증명 보고서는 인스턴스의 상태와 ID를 검증하고 승인된 AMD 환경에서 실행 중인지 확인하는 데 사용할 수 있습니다. 보고서에는 초기 인스턴스 메모리 콘텐츠 및 vCPU의 초기 상태를 포함하여 인스턴스의 초기 부팅 상태에 대한 암호화 해시인 시작 측정값이 포함됩니다. AMD SEV-SNP 증명 보고서에는 AMD의 신뢰 루트에 다시 연결되는 VLEK 서명이 포함되어 있습니다.

VLEK

VLEK(Versioned Loaded Endorsement Key)는 AMD에서 인증하고 AMD CPU에서 AMD SEV-SNP 증명 보고서에 서명하는 데 사용하는 버전이 지정된 서명 키입니다. VLEK 서명은 AMD에서 제공하는 인증서를 사용하여 확인할 수 있습니다.

OVMF 이진수

OVMF(Open Virtual Machine Firmware)는 인스턴스에 대한 UEFI 환경을 제공하는 데 사용되는 초기 부팅 코드입니다. 초기 부팅 코드는 AMI의 코드가 부팅되기 전에 실행됩니다. 또한 OVMF는 AMI에 제공된 부트 로더를 찾아 실행합니다. 자세한 내용은 OVMF 리포지토리를 참조하세요.

요구 사항

AMD SEV-SNP를 사용하려면 다음을 수행해야 합니다.

- 지원되는 인스턴스 유형 중 하나를 사용해야 합니다.
  - 범용: `m6a.large` | `m6a.xlarge` | `m6a.2xlarge` | `m6a.4xlarge` | `m6a.8xlarge`
  - 컴퓨팅 최적화: `c6a.large` | `c6a.xlarge` | `c6a.2xlarge` | `c6a.4xlarge` | `c6a.8xlarge` | `c6a.12xlarge` | `c6a.16xlarge`
  - 메모리 최적화: `r6a.large` | `r6a.xlarge` | `r6a.2xlarge` | `r6a.4xlarge`
- 지원되는 AWS 리전에서 인스턴스를 시작합니다. 현재는 미국 동부(오하이오)와 유럽(아일랜드)만 지원됩니다.
- `uefi` 또는 `uefi-preferred` 부팅 모드의 AMI와 AMD SEV-SNP를 지원하는 운영 체제를 사용하세요. 운영 체제의 AMD SEV-SNP 지원에 관한 자세한 내용은 해당 운영 체제의 설명서를 참조하세요. AWS의 경우 AMD SEV-SNP는 AL2023, RHEL 9.3, SLES 15 SP4 및 Ubuntu 23.04 이상에서 지원됩니다.

고려 사항

AMD SEV-SNP는 인스턴스를 시작할 때만 활성화할 수 있습니다. 인스턴스 시작에 대해 AMD SEV-SNP가 활성화되어 있으면 다음 규칙이 적용됩니다.

- AMD SEV-SNP는 활성화한 이후에 비활성화할 수 없습니다. 인스턴스 수명 주기 동안 활성화됩니다.
- AMD SEV-SNP를 지원하는 다른 인스턴스 유형으로만 인스턴스 유형을 변경할 수 있습니다.
- 최대 절전 및 Nitro Enclaves는 지원되지 않습니다.
- 전용 호스트는 지원되지 않습니다.

- 인스턴스의 기본 호스트가 유지 관리 일정이 예정된 경우 이벤트 14일 전에 예약된 이벤트 알림을 받게 됩니다. 인스턴스를 새 호스트로 이동하려면 인스턴스를 수동으로 중지하거나 다시 시작해야 합니다.

요금

AMD SEV-SNP가 활성화된 상태에서 Amazon EC2 인스턴스를 시작하면 선택한 인스턴스 유형의 온디맨드 시간당 요금의 10% 에 해당하는 추가 시간당 사용 요금이 부과됩니다.

이 AMD SEV-SNP 사용 요금은 Amazon EC2 인스턴스 사용에 대한 별도의 요금입니다. 예약 인스턴스, 절감형 플랜 및 운영 체제 사용량은 이 요금에 영향을 미치지 않습니다.

AMD SEV-SNP가 활성화된 상태에서 스팟 인스턴스를 시작하도록 구성하면 선택한 인스턴스 유형의 온디맨드 시간당 요금의 10% 에 해당하는 시간당 사용 요금이 추가로 부과됩니다. 할당 전략에서 가격을 입력으로 사용하는 경우 스팟 플릿에는 이 추가 요금이 포함되지 않습니다. 스팟 가격만 사용됩니다.

AMD SEV-SNP를 지원하는 EC2 인스턴스 유형 찾기

AMD SEV-SNP를 지원하는 인스턴스 유형을 찾을 수 있습니다. Amazon EC2 콘솔에는 인스턴스 유형에 대한 이 정보가 표시되지 않습니다.

AWS CLI

AMD SEV-SNP를 지원하는 인스턴스 유형을 찾으려면

다음 describe-instance-types 명령을 사용합니다.

```
aws ec2 describe-instance-types \
    --filters Name=processor-info.supported-features,Values=amd-sev-snp \
    --query InstanceTypes[*].[InstanceType] \
    --output text | sort
```

출력의 예시는 다음과 같습니다.

```
c6a.12xlarge
c6a.16xlarge
c6a.2xlarge
c6a.4xlarge
```

```
c6a.8xlarge
c6a.large
c6a.xlarge
m6a.2xlarge
m6a.4xlarge
m6a.8xlarge
m6a.large
m6a.xlarge
r6a.2xlarge
r6a.4xlarge
r6a.large
r6a.xlarge
```

PowerShell

AMD SEV-SNP를 지원하는 인스턴스 유형을 찾으려면

Get-EC2InstanceType cmdlet을 사용합니다.

```
(Get-EC2InstanceType `
    -Filter @{Name="processor-info.supported-features"; Values="amd-sev-
snp"}).InstanceType.Value | Sort-Object
```

출력의 예시는 다음과 같습니다.

```
c6a.12xlarge
c6a.16xlarge
c6a.2xlarge
c6a.4xlarge
c6a.8xlarge
c6a.large
c6a.xlarge
m6a.2xlarge
m6a.4xlarge
m6a.8xlarge
m6a.large
m6a.xlarge
r6a.2xlarge
r6a.4xlarge
r6a.large
r6a.xlarge
```

EC2 인스턴스에서 AMD SEV-SNP 활성화

AMD SEV-SNP가 활성화된 상태에서 인스턴스를 시작할 수 있습니다. 시작 후에는 AMD SEV-SNP를 활성화할 수 없습니다.

AMD SEV-SNP가 활성화된 상태에서 인스턴스 시작

Amazon EC2 콘솔을 사용하여 AMD SEV-SNP를 활성화할 수는 없습니다.

AWS CLI

AMD SEV-SNP가 활성화된 상태에서 인스턴스를 시작하려면

`--cpu-options` 옵션과 함께 run-instances 명령을 사용합니다. 추가 요구 사항은 AMD SEV-SNP 요구 사항을 참조하세요.

```
--cpu-options AmdSevSnp=enabled
```

PowerShell

AMD SEV-SNP가 활성화된 상태에서 인스턴스를 시작하려면

`-CpuOption` 파라미터와 함께 New-EC2Instance cmdlet을 사용합니다.

```
-CpuOption @{AmdSevSnp="enabled"}
```

EC2 인스턴스에 AMD SEV-SNP가 활성화되어 있는지 확인

AMD SEV-SNP가 활성화된 인스턴스를 찾을 수 있습니다. Amazon EC2 콘솔에는 이 정보가 표시되지 않습니다.

AWS CLI

인스턴스에 AMD SEV-SNP가 활성화되어 있는지 확인하려면

describe-instances 명령을 사용합니다.

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

출력의 예시는 다음과 같습니다. CpuOptions에 AmdSevSnp가 없으면 AMD SEV-SNP가 비활성화된 것입니다.

```
[
    {
        "AmdSevSnp": "enabled",
        "CoreCount": 1,
        "ThreadsPerCore": 2
    }
]
```

PowerShell

인스턴스에 AMD SEV-SNP가 활성화되어 있는지 확인하려면

Get-EC2Instance cmdlet을 사용합니다.

```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances.CpuOptions
```

출력의 예시는 다음과 같습니다. AmdSevSnp 값이 없으면 AMD SEV-SNP가 비활성화된 것입니다.

```
AmdSevSnp CoreCount ThreadsPerCore
--------- --------- --------------
enabled   1         2
```

AWS CloudTrail

인스턴스 시작 요청에 대한 AWS CloudTrail 이벤트에서 다음 속성은 해당 인스턴스에 AMD SEV-SNP가 활성화되어 있음을 나타냅니다.

```
"cpuOptions": {"AmdSevSnp": "enabled"}
```

AMD SEV-SNP로 Amazon EC2 인스턴스 증명

증명은 인스턴스가 상태 및 ID를 증명할 수 있는 프로세스입니다. 인스턴스에서 AMD SEV-SNP를 활성화한 후에는 기본 프로세서에 AMD SEV-SNP 증명 보고서를 요청할 수 있습니다. AMD SEV-SNP 증명 보고서에는 초기 게스트 메모리 콘텐츠 및 초기 vCPU 상태에 대한 시작 측정값이라는 암호화 해시가 포함되어 있습니다. 증명 보고서에는 AMD의 신뢰 루트에 다시 연결되는 VLEK 서명이 포함되어

있습니다. 증명 보고서에 포함된 시작 측정값을 사용하여 인스턴스가 정품 AMD 환경에서 실행되고 있는지 확인하고 인스턴스를 시작하는 데 사용된 초기 부팅 코드를 확인할 수 있습니다.

전제 조건

AMD SEV-SNP에 대해 활성화된 인스턴스를 시작합니다. 자세한 내용은 EC2 인스턴스에서 AMD SEV-SNP 활성화 섹션을 참조하세요.

단계

- 1단계: 증명 보고서 가져오기
- 2단계: 증명 보고서 서명 확인

1단계: 증명 보고서 가져오기

이 단계에서는 snpguest 유틸리티를 설치 및 구축한 다음, 이를 사용하여 AMD SEV-SNP 증명 보고서와 인증서를 요청합니다.

1. 인스턴스에 연결합니다.
2. snpguest repository에서 snpguest 유틸리티를 구축하려면 다음 명령을 실행합니다.

```
$ git clone https://github.com/virtee/snpguest.git
$ cd snpguest
$ cargo build -r
$ cd target/release
```

3. 증명 보고서에 대한 요청을 생성합니다. 유틸리티는 호스트에서 증명 보고서를 요청하고 제공된 요청 데이터를 사용해 이진 파일로 작성합니다.

   다음 예제에서는 무작위 요청 문자열을 생성하고 요청 파일(request-file.txt)로 사용합니다. 명령이 증명 보고서를 반환하면 지정한 파일 경로에 증명 보고서가 저장됩니다(report.bin). 이 경우 유틸리티는 보고서를 현재 디렉터리에 저장합니다.

```
$ ./snpguest report report.bin request-file.txt --random
```

4. 호스트 메모리에서 인증서를 요청하고 PEM 파일로 저장합니다. 다음 예제에서는 snpguest 유틸리티와 같은 디렉터리에 파일을 저장합니다. 지정된 디렉터리에 인증서가 이미 있는 경우 해당 인증서를 덮어씁니다.

```
$ ./snpguest certificates PEM ./
```

2단계: 증명 보고서 서명 확인

증명 보고서에는 AMD에서 AWS용으로 발급한 VLEK(Versioned Loaded Endorsement Key)라는 인증서로 서명됩니다. 이 단계에서는 VLEK 인증서가 AMD에서 발급되었고, 증명 보고서가 해당 VLEK 인증서로 서명되었는지 확인할 수 있습니다.

1. 공식 AMD 웹 사이트에서 현재 디렉터리로 VLEK 신뢰 루트 인증서를 다운로드합니다.

```
$ sudo curl --proto '=https' --tlsv1.2 -sSf https://kdsintf.amd.com/vlek/v1/Milan/
cert_chain -o ./cert_chain.pem
```

2. openssl을(를) 사용하여 VLEK 인증서가 AMD 신뢰 루트 인증서에 의해 서명되었는지 확인합니다.

```
$ sudo openssl verify --CAfile ./cert_chain.pem vlek.pem
```

출력의 예시는 다음과 같습니다.

```
certs/vcek.pem: OK
```

3. snpguest 유틸리티를 사용하여 증명 보고서가 VLEK 인증서로 서명되었는지 확인합니다.

```
$ ./snpguest verify attestation ./ report.bin
```

다음은 예시 출력입니다.

```
Reported TCB Boot Loader from certificate matches the attestation report.
Reported TCB TEE from certificate matches the attestation report.
Reported TCB SNP from certificate matches the attestation report.
Reported TCB Microcode from certificate matches the attestation report.
VEK signed the Attestation Report!
```

Amazon EC2 Linux 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.

> **Note**
>
> AWS Graviton 프로세서는 절전 모드를 기본으로 제공하며 고정 주파수로 작동합니다. 따라서 운영 체제가 C 상태 및 P 상태를 제어하는 기능을 제공하지 않습니다.

C-states 및P-state

다음 인스턴스 유형은 운영 체제에서 프로세서 C 상태 및 P 상태를 제어할 수 있는 기능을 제공합니다.

- 범용: `m4.10xlarge` | `m4.16xlarge`
- 컴퓨팅 최적화: `c4.8xlarge`
- 메모리 최적화: `r4.8xlarge` | `r4.16xlarge` | `x1.16xlarge` | `x1.32xlarge` | `x1e.8xlarge` | `x1e.16xlarge` | `x1e.32xlarge`
- 스토리지 최적화: `d2.8xlarge` | `i3.8xlarge` | `i3.16xlarge` | `h1.8xlarge` | `h1.16xlarge`
- 가속 컴퓨팅: `f1.16xlarge` | `g3.16xlarge` | `p2.16xlarge` | `p3.16xlarge`
- 베어 메탈: Intel 및 AMD 프로세서가 탑재된 모든 베어 메탈

C-state 전용

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` | `m5zn.6xlarge` | `m5zn.12xlarge` | `m6a.24xlarge` | `m6a.48xlarge` | `m6i.16xlarge` | `m6i.32xlarge` | `m6id.16xlarge` | `m6id.32xlarge` | `m6idn.16xlarge` | `m6in.16xlarge` | `m6in.32xlarge` | `m7a.medium` | `m7a.large` | `m7a.xlarge` | `m7a.2xlarge` | `m7a.4xlarge` | `m7a.8xlarge` | `m7a.12xlarge` | `m7a.16xlarge` | `m7a.24xlarge` | `m7a.32xlarge` | `m7a.48xlarge` | `m7i.large` | `m7i.xlarge` | `m7i.2xlarge` | `m7i.4xlarge` | `m7i.8xlarge` | `m7i.12xlarge` | `m7i.16xlarge` | `m7i.24xlarge` | `m7i.48xlarge`
- 컴퓨팅 최적화: `c5.9xlarge` | `c5.12xlarge` | `c5.18xlarge` | `c5.24xlarge` | `c5a.24xlarge` | `c5ad.24xlarge` | `c5d.9xlarge` | `c5d.12xlarge` | `c5d.18xlarge` | `c5d.24xlarge`

| c5n.9xlarge | c5n.18xlarge | c6a.24xlarge | c6a.32xlarge | c6a.48xlarge | c6i.16xlarge | c6i.32xlarge | c6id.24xlarge | c6id.32xlarge | c6in.32xlarge | c7a.medium | c7a.large | c7a.xlarge | c7a.2xlarge | c7a.4xlarge | c7a.8xlarge | c7a.12xlarge | c7a.16xlarge | c7a.24xlarge | c7a.32xlarge | c7a.48xlarge | c7i.large | c7i.xlarge | c7i.2xlarge | c7i.4xlarge | c7i.8xlarge | c7i.12xlarge | c7i.16xlarge | c7i.24xlarge | c7i.48xlarge

- 메모리 최적화: r5.12xlarge | r5.24xlarge | r5b.12xlarge | r5d.12xlarge | r5d.24xlarge | r5n.12xlarge | r5n.24xlarge | r5dn.12xlarge | r5dn.24xlarge | r6a.24xlarge | r6a.48xlarge | r6i.16xlarge | r6i.32xlarge | r6id.16xlarge | r6id.32xlarge | r6in.16xlarge | r6in.32xlarge | r7a.medium | r7a.large | r7a.xlarge | r7a.2xlarge | r7a.4xlarge | r7a.8xlarge | r7a.12xlarge | r7a.16xlarge | r7a.24xlarge | r7a.32xlarge | r7a.48xlarge | r7i.large | r7i.xlarge | r7i.2xlarge | r7i.4xlarge | r7i.8xlarge | r7i.12xlarge | r7i.16xlarge | r7i.24xlarge | r7i.48xlarge | r7iz.large | r7iz.xlarge | r7iz.2xlarge | r7iz.4xlarge | r7iz.8xlarge | r7iz.12xlarge | r7iz.16xlarge | r7iz.32xlarge | u-3tb1.56xlarge | u-6tb1.56xlarge | u-6tb1.112xlarge | u-9tb1.112xlarge | u-12tb1.112xlarge | u-18tb1.112xlarge | u-24tb1.112xlarge | u7i-6tb.112xlarge | u7i-8tb.112xlarge | u7i-12tb.224xlarge | u7in-16tb.224xlarge | u7in-24tb.224xlarge | u7in-32tb.224xlarge | u7inh-32tb.480xlarge | x2idn.32xlarge | x2iedn.16xlarge | x2iedn.32xlarge | x2iezn.12xlarge | z1d.6xlarge | z1d.12xlarge
- 스토리지 최적화: d3en.12xlarge | dl1.24xlarge | i3en.12xlarge | i3en.24xlarge | i4i.16xlarge | i7i.large | i7i.xlarge | i7i.2xlarge | i7i.4xlarge | i7i.8xlarge | i7i.12xlarge | i7i.16xlarge | i7i.24xlarge | i7i.48xlarge | i7ie.large | i7ie.xlarge | i7ie.2xlarge | i7ie.3xlarge | i7ie.6xlarge | i7ie.12xlarge | i7ie.18xlarge | i7ie.24xlarge | i7ie.48xlarge | r5b.12xlarge | r5b.24xlarge
- 가속 컴퓨팅: dl1.24xlarge | f2.6xlarge | f2.12xlarge | f2.48xlarge | g5.24xlarge | g5.48xlarge | g6.24xlarge | g6.48xlarge | g6e.12xlarge | g6e.24xlarge | g6e.48xlarge | inf1.24xlarge | p3dn.24xlarge | p4d.24xlarge | p4de.24xlarge | p5.48xlarge | p5e.48xlarge | p5en.48xlarge | p6-b200.48xlarge | trn1.32xlarge | trn2.3xlarge | trn2.48xlarge | trn2a.3xlarge | trn2a.48xlarge | trn2n.3xlarge | trn2n.48xlarge | trn2p.48xlarge | trn2u.48xlarge | vt1.24xlarge

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 성능

을 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 고려하세요.

다양한 프로세서 구성 및 Amazon Linux 구성의 영향을 모니터링하는 방법에 대한 자세한 내용은 Amazon Linux 2 사용 설명서의 Amazon EC2 Amazon Linux 인스턴스의 프로세서 상태 제어를 참조하십시오. 이러한 절차는 Amazon Linux용으로 작성 및 적용되었지만 Kinux 커널 3.9 이상의 다른 Linux 배포판에서도 적용될 수 있습니다. Linux 배포판 및 프로세서 상태 제어에 대한 자세한 내용은 시스템별 설명서를 참조하세요.

Amazon EC2 관리형 인스턴스

Amazon EC2 관리형 인스턴스는 EKS Auto Mode를 통해 Amazon EKS와 같은 지정된 서비스 제공업체가 프로비저닝하고 관리하는 EC2 인스턴스입니다. 관리형 인스턴스는 인스턴스의 운영 제어를 서비스 제공업체에 위임할 수 있어 간소화된 방법으로 Amazon EC2에서 컴퓨팅 워크로드를 실행할 수 있습니다.

위임된 제어는 관리형 인스턴스에 도입된 유일한 변경 사항입니다. 기술 사양 및 결제는 비관리형 EC2 인스턴스와 같습니다. 관리형 인스턴스를 사용하면 서비스 제공업체에 제어를 위임할 수 있으므로 서비스 제공업체의 운영 전문성과 모범 사례를 활용할 수 있습니다. 인스턴스를 관리할 때 서비스 제공업체는 인스턴스 프로비저닝, 소프트웨어 구성, 용량 규모 조정, 인스턴스 장애 및 교체 처리, 인스턴스 종료와 같은 작업을 담당합니다.

관리형 인스턴스의 설정을 직접 수정하거나 해당 인스턴스를 종료할 수 없습니다. 서비스 및 특정 운영은 사용자와 서비스 제공업체 간의 계약에 따라 결정됩니다. 그러나 관리형 인스턴스에서 태그를 추가, 수정 또는 제거하여 AWS 환경 내에서 인스턴스를 분류할 수 있습니다.

내용

- 관리형 인스턴스에 대한 결제
- 관리형 인스턴스 식별
- 관리형 인스턴스 시작하기

관리형 인스턴스에 대한 결제

Amazon EC2 관리형 인스턴스에는 비관리형 Amazon EC2 인스턴스와 동일한 기본 요금에 더해 서비스 제공업체에 대한 별도의 요금이 발생합니다. 이 추가 요금은 인스턴스를 관리하는 서비스 제공업체

가 부과하며 별도로 청구됩니다. 이러한 요금은 관리형 인스턴스의 운영 및 유지 관리를 위해 제공되는 서비스의 비용에 해당합니다.

온디맨드 인스턴스, 예약 인스턴스, 스팟 인스턴스 및 절감형 플랜을 포함한 모든 Amazon EC2 구매 옵션을 관리형 인스턴스에 사용할 수 있습니다. EC2에서 직접 컴퓨팅을 소싱한 다음 서비스 제공업체에 제공하면 계정에 적용된 기존 예약 인스턴스 또는 절감형 플랜의 이점을 누릴 수 있으므로 사용 가능한 가장 비용 효율적인 컴퓨팅 용량을 사용할 수 있습니다.

예를 들어 Amazon EKS Auto Mode를 사용하는 경우 기본 인스턴스에 대한 표준 EC2 인스턴스 요금에 더해 사용자를 대신한 인스턴스 관리에 대해 Amazon EKS에서 발생하는 추가 요금을 지불합니다. 그런 다음 절감형 플랜에 가입하기로 결정하면 절감형 플랜으로 인해 EC2 인스턴스 속도는 감소하지만 Amazon EKS의 추가 요금은 변동이 없습니다.

관리형 인스턴스 식별

관리형 인스턴스는 관리형 필드에서 true 값으로 식별됩니다. 서비스 제공업체는 운영자 필드(콘솔) 또는 `Principal` 필드(CLI)으로 식별됩니다.

다음 절차에 따라 관리형 인스턴스를 식별합니다.

Console

관리형 인스턴스를 식별하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 확인할 인스턴스를 선택합니다.
4. 세부 정보 탭(확인란을 선택한 경우) 또는 요약 영역(인스턴스 ID를 선택한 경우)에서 관리형 필드를 찾습니다.
   - true 값은 관리형 인스턴스를 나타냅니다.
   - false 값은 비관리형 인스턴스를 나타냅니다.
5. 관리형이 true로 설정된 경우 운영자 필드에는 인스턴스 관리를 담당하는 서비스 제공업체를 식별하는 값이 표시됩니다. 예를 들어 eks.amazonaws.com 값은 Amazon EKS를 서비스 제공업체로 식별합니다.

AWS CLI

관리형 인스턴스를 식별하려면

describe-instances 명령을 사용하여 인스턴스 ID를 지정합니다.

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

출력의 예시는 다음과 같습니다. Managed가 true라면 인스턴스가 관리형 인스턴스이고 Principal이 포함되어 있습니다. 주체는 인스턴스를 관리하는 서비스 제공업체입니다. 예를 들어의 eks.amazonaws.com 값은 Amazon EKS를 서비스 제공업체로 식별합니다.

```
[
    {
        "Managed": true,
        "Principal": "eks.amazonaws.com"
    }
]
```

관리형 인스턴스를 찾는 방법

describe-instances 명령어를 사용하고 값이 true인 operator.managed 필터를 지정합니다. --query 옵션에서는 관리형 인스턴스의 ID만 표시됩니다.

```
aws ec2 describe-instances \
    --filters "Name=operator.managed,Values=true" \
    --query Reservations[*].Instances[].InstanceId
```

PowerShell

관리형 인스턴스를 식별하려면

Get-EC2Instance cmdlet을 사용합니다.

```
(Get-EC2Instance -InstanceId i-1234567890abcdef0).Instances.Operator
```

출력의 예시는 다음과 같습니다.

```
Managed Principal
------- ---------
True    eks.amazonaws.com
```

관리형 인스턴스를 찾는 방법

Get-EC2Instance cmdlet을 사용합니다. 이 예제는 관리형 인스턴스의 ID만 표시합니다.

```
(Get-EC2Instance -Filter @{Name="operator.managed";
 Values="true"}).Instances.InstanceId
```

관리형 인스턴스 시작하기

관리형 인스턴스 사용에 대한 지침은 Amazon EKS 사용 설명서의 EKS Auto Mode를 사용하여 클러스터 인프라 자동화를 참조하세요.

Amazon EC2 결제 및 구매 옵션

다음 옵션을 사용하여 Amazon EC2 비용을 최적화할 수 있습니다.

- 온디맨드 인스턴스 - 시작하는 인스턴스에 대한 비용을 초 단위로 지불합니다.
- 절감형 플랜(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 및 리전을 포함하여 일관된 인스턴스 구성을 약정하여 Amazon EC2 비용을 절감할 수 있습니다.
- 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 - 특정 가용 영역의 EC2 인스턴스에 대해 용량을 예약합니다.

특정 인스턴스 구성을 약정할 수 없지만 사용량은 약정할 수 있는 경우 절감형 플랜을 구매하여 온디맨드 인스턴스 비용을 줄이세요. 용량 예약이 필요한 경우 특정 가용 영역에 대한 예약 인스턴스 또는 용량 예약을 구매합니다. GPU 인스턴스 클러스터 예약에 용량 블록을 사용할 수 있습니다. 스팟 인스턴스는 애플리케이션이 실행되는 시간을 유연하게 조정할 수 있고 애플리케이션이 중단될 수 있는 경우에 선택하는 비용 효율적인 방법입니다. 전용 호스트 또는 전용 인스턴스에서는 기존의 서버 바인딩 소프트웨어 라이선스를 사용할 수 있으므로 규정 준수 요구 사항을 해결하고 비용을 절감하는 데 도움이 될 수 있습니다.

자세한 내용은 Amazon EC2 요금 및 Amazon EC2 관리형 인스턴스를 참조하세요.

Amazon EC2용 온디맨드 인스턴스 구매

온디맨드 인스턴스를 사용하면 장기 약정 없이 초 단위로 컴퓨팅 용량에 대해 비용을 지불합니다. 인스턴스의 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(최소 60초)에 대해서만 비용을 지불하면 됩니다. 실행 중인 온디맨드 인스턴스에 대한 초당 요금은 고정되어 있으며, Amazon EC2 요금, 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, Spot Instances 또는 Amazon EC2용 예약 인스턴스 개요를 사용합니다.

목차

- 온디맨드 인스턴스 할당량
  - 온디맨드 인스턴스 할당량 및 사용량 모니터링
  - 할당량 증가 요청
- 온디맨드 인스턴스 가격 쿼리

온디맨드 인스턴스 할당량

각 리전에서 AWS 계정당 실행할 수 있는 온디맨드 인스턴스 수에는 할당량이 있습니다. 온디맨드 인스턴스 할당량은 인스턴스 유형과 상관없이 실행 중인 온디맨드 인스턴스에 사용되는 가상 중앙 처리 유닛(vCPU) 수의 측면에서 관리됩니다. 각 할당량 유형은 하나 이상의 인스턴스 패밀리에 대한 최대 vCPU 수를 지정합니다.

계정에는 다음과 같은 온디맨드 인스턴스 할당량이 포함되어 있습니다. 보류 중이거나, 중지 중이거나, 중지되었거나, 최대 절전 모드 상태가 된 인스턴스는 온디맨드 인스턴스 할당량에 포함되지 않습니다. 용량 예약 수는 사용되는 경우에도 온디맨드 인스턴스 할당량에 포함됩니다.

명칭	기본값	조정 가능
온디맨드 DL 인스턴스 실행	0	예
온디맨드 F 인스턴스 실행	0	예

명칭	기본값	조정 가능
온디맨드 G 및 VT 인스턴스 실행	0	예
온디맨드 HPC 인스턴스 실행	0	예
온디맨드 고용량 메모리 인스턴스 실행	0	예
온디맨드 Inf 인스턴스 실행	0	예
온디맨드 P 인스턴스 실행	0	예
온디맨드 표준(A, C, D, H, I, M, R, T, Z) 인스턴스 실행	5	예
온디맨드 Trn 인스턴스 실행	0	예
온디맨드 X 인스턴스 실행	0	예

다양한 인스턴스 패밀리, 세대, 크기에 대한 자세한 내용은 Amazon EC2 인스턴스 유형 가이드를 참조하세요.

vCPU 수가 계정 할당량을 초과하지만 않으면 변화하는 애플리케이션 요구 사항을 충족하는 모든 인스턴스 유형을 조합하여 시작할 수 있습니다. 예를 들어 표준 인스턴스 할당량이 256개 vCPU인 경우 32개의 `m5.2xlarge` 인스턴스(32x8 vCPU) 또는 16개의 `c5.4xlarge` 인스턴스(16x16 vCPU)를 시작할 수 있습니다. 자세한 내용은 EC2 온디맨드 인스턴스 제한 섹션을 참조하세요.

업무

- 온디맨드 인스턴스 할당량 및 사용량 모니터링
- 할당량 증가 요청

온디맨드 인스턴스 할당량 및 사용량 모니터링

다음 방법을 사용하여 각 리전에 대한 온디맨드 인스턴스 할당량을 보고 관리할 수 있습니다.

Service Quotas 콘솔을 사용하여 현재 할당량 보기

1. Service Quotas 콘솔(https://console.aws.amazon.com/servicequotas/home/services/ec2/quotas/)을 엽니다.

2. 탐색 모음에서 리전을 선택합니다.
3. 필터 필드에 **`On-Demand`**을(를) 입력합니다.
4. 적용된 할당량 값 열에는 계정의 각 온디맨드 인스턴스 할당량 유형에 대한 최대 vCPU 수가 표시됩니다.

AWS Trusted Advisor 콘솔을 사용하여 현재 할당량 보기

AWS Trusted Advisor 콘솔에서 서비스 한도 페이지를 엽니다.

CloudWatch 경보 구성

Amazon CloudWatch 지표 통합을 통해 할당량에 대해 EC2 사용량을 모니터링할 수 있습니다. 할당량 도달에 대해 경고를 받도록 경보를 구성할 수도 있습니다. 자세한 내용은 CloudWatch 사용 설명서의 Service Quotas 및 Amazon CloudWatch 경보를 참조하세요.

할당량 증가 요청

Amazon EC2는 사용량에 따라 자동으로 온디맨드 인스턴스 할당량을 늘리지만 필요한 경우 할당량 증가를 요청할 수 있습니다. 예를 들어, 현재 할당량에서 허용하는 것보다 많은 인스턴스를 시작하려는 경우 Amazon EC2 서비스 할당량에 설명된 이전 섹션의 Service Quotas 콘솔에 를 사용하여 할당량 증가를 요청할 수 있습니다.

온디맨드 인스턴스 가격 쿼리

Price List Service API 또는 AWS Price List API를 사용하여 온디맨드 인스턴스 가격을 쿼리할 수 있습니다. 자세한 내용은 AWS Billing 사용 설명서에서 AWS Price List API 사용을 참조하세요.

Amazon EC2용 예약 인스턴스 개요

> **Important**
>
> 예약 인스턴스보다 절감형 플랜을 권장합니다. 절감형 플랜은 예약 인스턴스처럼 AWS 컴퓨팅 비용을 절약하고 더 저렴한 가격(온디맨드 요금에서 최대 72% 할인)을 제공하는 가장 쉽고 가장 유연한 방법입니다. 하지만 절감형 플랜은 예약 인스턴스와는 다릅니다. 예약 인스턴스에서는 특정 인스턴스 구성을 커밋하지만, 절감형 플랜에서는 요구 사항에 가장 적합한 인스턴스 구성을 사용할 수 있는 유연성을 제공합니다. 절감형 플랜을 사용하려면 일관된 사용량을 커밋합니다(시간당 USD로 측정됨). 자세한 내용은 AWS Savings Plans 사용 설명서를 참조하세요.

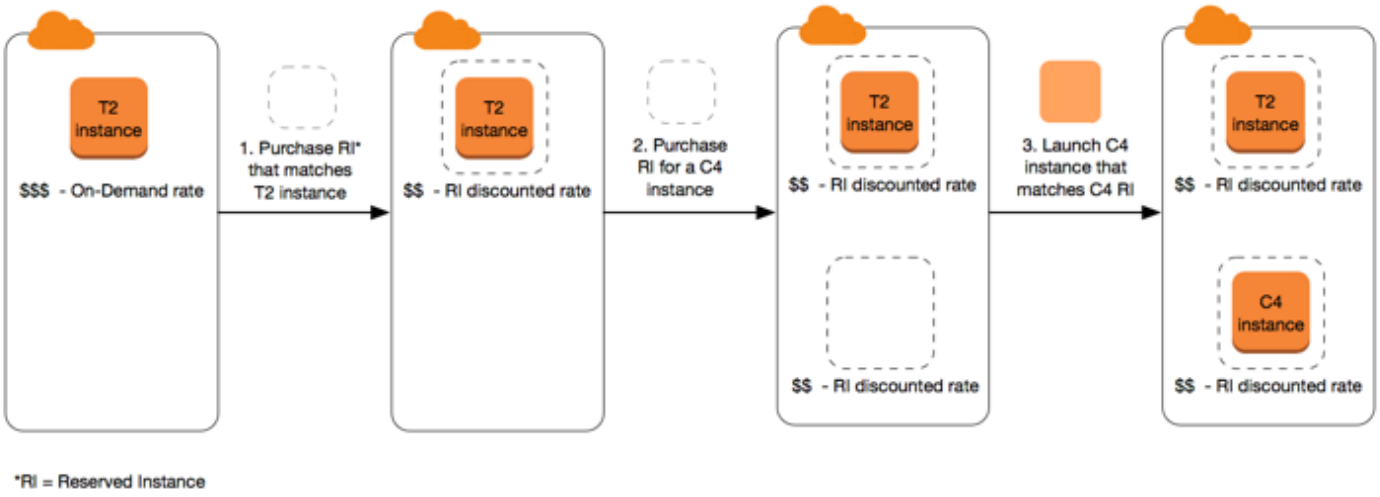
예약 인스턴스는 온디맨드 인스턴스 요금과 비교하여 Amazon EC2 비용을 대폭 절감하는 효과를 제공합니다. 예약 인스턴스는 물리적 인스턴스가 아니며 계정에서 온디맨드 인스턴스를 사용할 때 적용되는 결제 할인에 가깝습니다. 이러한 온디맨드 인스턴스의 경우 결제 할인 혜택을 받으려면 인스턴스 유형 및 지역과 같은 특정 속성에 부합해야 합니다.

예약 인스턴스 주제

- 예약 인스턴스 예제 시나리오
- 예약 인스턴스 요금을 결정하는 주요 변수
- 리전 및 영역 예약 인스턴스(범위)
- 예약 인스턴스 유형(제공 클래스)
- 예약 인스턴스 할인 적용 방식
- 예약 인스턴스 사용
- 예약 인스턴스 결제 작동 방식
- Amazon EC2용 예약 인스턴스 구입
- 예약 인스턴스 마켓플레이스에서 Amazon EC2용 예약 인스턴스 판매
- 예약 인스턴스 수정
- 전환형 예약 인스턴스 교환
- 예약 인스턴스 할당량

예약 인스턴스 예제 시나리오

다음 다이어그램에는 예약 인스턴스 구입 및 사용에 대한 기본 시나리오가 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.

- 인스턴스 유형: 예를 들어, `m4.large`입니다. 이는 인스턴스 패밀리(예: `m4`)와 인스턴스 크기(예: `large`)로 구성됩니다.
- 리전: 예약 인스턴스를 구매한 리전입니다.
- 테넌시: 인스턴스가 공유된 하드웨어(기본)에서 실행되는지 단일 테넌트(전용) 하드웨어에서 실행되는지 여부입니다. 자세한 내용은 Amazon EC2 전용 인스턴스 섹션을 참조하세요.
- 플랫폼: Windows 또는 Linux/Unix와 같은 운영 체제입니다. 자세한 내용은 플랫폼 선택 섹션을 참조하세요.

기간 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.

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.

> **Important**
>
> 예약 인스턴스를 구입한 이후에는 구입을 취소할 수 없습니다. 그러나 변경이 필요한 경우 예약 인스턴스를 수정, 교환 또는 판매할 수 있습니다.

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 Marketplace에서 서드 파티 판매자가 제공하는 저렴하고 기간이 짧은 예약 인스턴스를 찾을 수도 있습니다. 자세한 내용은 예약 인스턴스 마켓플레이스에서 Amazon EC2용 예약 인스턴스 판매 섹션을 참조하세요.

제공 클래스

컴퓨팅 요구 사항이 변경되면 제공 클래스에 따라 예약 인스턴스를 수정 또는 교환할 수 있습니다.

- 표준: 가장 큰 할인 혜택을 제공하지만 수정만 가능합니다. 스탠다드 예약 인스턴스는 교환할 수 없습니다.
- 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) 섹션을 참조하세요.

> **Important**
>
> 예약 인스턴스를 구입한 이후에는 구입을 취소할 수 없습니다. 그러나 변경이 필요한 경우 예약 인스턴스를 수정, 교환 또는 판매할 수 있습니다.

자세한 내용은 Amazon EC2 예약 인스턴스 요금 페이지를 참조하세요.

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.

범위는 요금에 영향을 미치지 않습니다. 리전 또는 영역 예약 인스턴스의 요금은 동일합니다. 예약 인스턴스 요금에 대한 자세한 내용은 예약 인스턴스 요금을 결정하는 주요 변수 섹션과 Amazon EC2 예약 인스턴스 요금을 참조하세요.

예약 인스턴스의 범위를 지정하는 방법을 자세히 알아보려면 RI 속성과 특히 가용 영역 글머리표를 참조하세요.

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
용량을 예약할 수 있는 기능	리전 예약 인스턴스에서는 용량을 예약하지 않습니다.	영역 예약 인스턴스에서는 지정된 가용 영역에서 용량을 예약합니다.
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.
인스턴스 크기 유연성	크기에 상관없이 인스턴스 패밀리 내 인스턴스 사용량에 예약 인스턴스 할인이 적용됩니다.	인스턴스 크기 유연성 없음 — 지정된 인스턴스 유형 및 크기의 인스턴스 사용량에만 예약 인스턴스 할인이 적용됩니다.

	리전 예약 인스턴스	영역 예약 인스턴스
	기본 테넌시가 있는 Amazon Linux/Unix 예약 인스턴스에 대해서만 지원됩니다. 자세한 내용은 정규화 인자에 의해 결정되는 인스턴스 크기 유연성 섹션을 참조하세요.	
구매 대기열에 추가	리전 예약 인스턴스에 대한 구매를 대기열에 추가할 수 있습니다.	영역 예약 인스턴스에 대한 구매는 대기열에 추가할 수 없습니다.

자세한 정보와 지침은 예약 인스턴스 할인 적용 방식 섹션을 참조하세요.

예약 인스턴스 유형(제공 클래스)

예약 인스턴스의 제공 클래스는 스탠다드 또는 컨버터블입니다. 스탠다드 예약 인스턴스는 컨버터블 예약 인스턴스보다 훨씬 많은 할인을 제공하지만 스탠다드 예약 인스턴스는 교환할 수 없습니다. 컨버터블 예약 인스턴스는 교환할 수 있습니다. 스탠다드 및 컨버터블 예약 인스턴스을 수정할 수 있습니다.

예약 인스턴스의 구성은 한 기간 동안 단일 인스턴스 유형, 플랫폼, 범위 및 테넌시로 구성됩니다. 컴퓨팅 요구 사항이 변경되면 예약 인스턴스를 수정하거나 교환할 수 있습니다.

스탠다드 및 컨버터블 예약 인스턴스의 차이점

스탠다드 및 컨버터블 예약 인스턴스의 차이점은 다음과 같습니다.

	표준 예약 인스턴스	컨버터블 예약형 인스턴스
예약 인스턴스 수정	일부 속성을 수정할 수 있습니다. 자세한 내용은 예약 인스턴스 수정 섹션을 참조하세요.	일부 속성을 수정할 수 있습니다. 자세한 내용은 예약 인스턴스 수정 섹션을 참조하세요.
예약 인스턴스 교환	교환할 수 없습니다.	기간 동안 인스턴스 패밀리, 인스턴스 유형, 플랫폼, 범위 또는

	표준 예약 인스턴스	컨버터블 예약형 인스턴스
		테넌시를 비롯한 새 속성이 있는 다른 전환형 예약 인스턴스와 교환할 수 있습니다. 자세한 내용은 전환형 예약 인스턴스 교환 섹션을 참조하세요.
예약 인스턴스 Marketplace에서 판매	예약 인스턴스 Marketplace에서 판매할 수 있습니다.	예약 인스턴스 Marketplace에서 판매할 수 없습니다.
예약 인스턴스 Marketplace에서 구매	예약 인스턴스 Marketplace에서 구매할 수 있습니다.	예약 인스턴스 Marketplace에서 구매할 수 없습니다.

예약 인스턴스 할인 적용 방식

예약 인스턴스는 물리적 인스턴스가 아니며 계정에서 온디맨드 인스턴스를 실행할 때 적용되는 결제 할인에 가깝습니다. 결제 할인 혜택을 받으려면 온디맨드 인스턴스가 예약 인스턴스의 특정 사양과 일치해야 합니다.

예약 인스턴스를 구입했으며 예약 인스턴스의 사양과 일치하는 온디맨드 인스턴스가 이미 실행 중인 경우 결제 할인이 즉시 자동으로 적용됩니다. 인스턴스를 따로 재시작할 필요가 없습니다. 실행 중인 적격 온디맨드 인스턴스가 없는 경우 예약 인스턴스와 동일한 사양의 온디맨드 인스턴스를 시작합니다. 자세한 내용은 예약 인스턴스 사용 섹션을 참조하세요.

예약 인스턴스의 오퍼링 클래스(스탠더드 또는 컨버터블)는 결제 할인이 적용되는 방식에 영향을 주지 않습니다.

주제

- 영역 예약 인스턴스 적용 방식
- 리전 예약 인스턴스 적용 방식
- 인스턴스 크기 유연성
- 예약 인스턴스 적용의 예

영역 예약 인스턴스 적용 방식

특정 가용 영역의 용량을 예약하기 위해 구매한 예약 인스턴스를 영역 예약 인스턴스라고 합니다.

- 예약 인스턴스 할인은 해당 가용 영역에서 일치하는 인스턴스 사용량에 적용됩니다.
- 단, 실행할 인스턴스의 속성(테넌시, 플랫폼, 가용 영역, 인스턴스 유형 및 인스턴스 크기)이 예약 인스턴스의 속성과 일치해야 합니다.

예를 들어 가용 영역 us-east-1a에 대해 `c4.xlarge` 기본 테넌시 Linux/Unix 표준 예약 인스턴스 2개를 구매하면 가용 영역 us-east-1a에서 실행되는 최대 2개의 `c4.xlarge` 기본 테넌시 Linux/Unix 인스턴스가 예약 인스턴스 할인 혜택을 받을 수 있습니다.

리전 예약 인스턴스 적용 방식

리전에 대해 구매한 예약 인스턴스를 리전 예약 인스턴스라고 하며, 이 인스턴스는 가용 영역 및 인스턴스 크기 유연성을 제공합니다.

- 해당 리전에 있는 모든 가용 영역의 인스턴스 사용량에 예약 인스턴스 할인이 적용됩니다.
- 크기에 상관없이 인스턴스 패밀리 내 인스턴스 사용량에 예약 인스턴스 할인이 적용됩니다. 이를 인스턴스 크기 유연성이라고 합니다.

인스턴스 크기 유연성

인스턴스 크기 유연성을 통해 예약 인스턴스 할인은 동일한 제품군이 있는 인스턴스의 인스턴스 사용량에 적용됩니다. 인스턴스 패밀리 내 인스턴스 사용량에 대해 예약 인스턴스 할인이 적용됩니다. 예약 인스턴스는 정규화 인자에 따라 인스턴스 패밀리 내 최소 인스턴스 크기부터 최대 인스턴스 크기까지 순차적으로 적용됩니다. 예약 인스턴스 할인이 적용되는 방법의 예는 시나리오 2: 정규화 인수를 사용하는 단일 계정의 예약 인스턴스 섹션을 참조하세요.

제한 사항

- 지원됨: 인스턴스 크기 유연성은 리전 단위의 예약 인스턴스에서만 지원됩니다.
- 지원되지 않음: 다음과 같은 예약 인스턴스에는 인스턴스 크기 유연성이 지원되지 않습니다.
  - 특정 가용 영역(영역 예약 인스턴스)용으로 구입한 예약 인스턴스
  - G4ad, G4dn, G5, G5g, G6, G6e, Gr6, hpc7a, P5, Inf1, Inf2, u7i-6tb, u7i-8tb 인스턴스용 예약 인스턴스
  - 예약 인스턴스 for Windows Server, Windows Server with SQL Standard, Windows Server with SQL Server Enterprise, Windows Server with SQL Server Web, RHEL, SUSE Linux Enterprise Server
  - 전용 테넌시를 포함하는 예약 인스턴스

정규화 인자에 의해 결정되는 인스턴스 크기 유연성

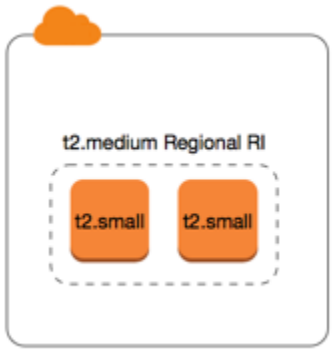
인스턴스 크기 유연성은 인스턴스 크기의 정규화 인자에 의해 결정됩니다. 리전의 모든 가용 영역에서 예약의 인스턴스 크기에 따라 모든 할인 또는 일부 할인이 동일한 인스턴스 패밀리의 실행 중인 인스턴스에 적용됩니다. 속성 중 인스턴스 패밀리, 테넌시 및 플랫폼만 일치하면 됩니다.

다음 표는 인스턴스 패밀리 내 서로 다른 크기 및 그에 따른 정규화 인자를 나열한 것입니다. 이 배율은 예약 인스턴스의 할인 요금을 정규화된 인스턴스 패밀리 사용량에 적용하는 데 사용됩니다.

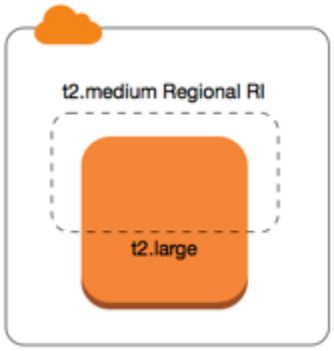
인스턴스 크기	정규화 인자
nano	0.25
micro	0.5
small	1
medium	2
large	4
xlarge	8
2xlarge	16
3xlarge	24
4xlarge	32
6xlarge	48
8xlarge	64
9xlarge	72
10xlarge	80
12xlarge	96
16xlarge	128
18xlarge	144

인스턴스 크기	정규화 인자
24xlarge	192
32xlarge	256
48xlarge	384
56xlarge	448
112xlarge	896

예를 들어, `t2.medium` 인스턴스의 정규화 인자는 2입니다. US East (N. Virginia)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 US East (N. Virginia) 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 섹션을 참조하세요.

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.

`metal` 인스턴스 크기에는 단일 정규화 인자가 없습니다. 베어 메탈 인스턴스는 동일한 인스턴스 패밀리 내에서 가상화된 인스턴스 크기와 동일한 정규화 인자를 갖습니다. 예를 들어, `i3.metal` 인스턴스의 정규화 인자는 `i3.16xlarge` 인스턴스의 정규화 인자와 동일합니다.

인스턴스 크기	정규화 인자
a1.metal	32
m5zn.metal | x2iezn.metal z1d.metal	96
c6g.metal | c6gd.metal | i3.metal | m6g.metal | m6gd.metal | r6g.metal | r6gd.metal | x2gd.metal	128
c5n.metal	144
c5.metal | c5d.metal | i3en.metal | m5.metal | m5d.metal | m5dn.metal | m5n.metal | r5.metal | r5b.metal | r5d.metal | r5dn.metal | r5n.metal	192
c6i.metal | c6id.metal | m6i.metal | m6id.metal | r6d.metal | r6id.metal	256
u-18tb1.metal | u-24tb1.metal	448
u-6tb1.metal | u-9tb1.metal | u-12tb1.metal	896

예를 들어 `i3.metal` 인스턴스의 정규화 인자는 128입니다. US East (N. Virginia)에서 `i3.metal` 기본 테넌시 Amazon Linux/Unix 예약 인스턴스를 구입하면 다음과 같은 요금 혜택이 적용될 수 있습니다.

- 리전에서 사용자의 계정에 실행 중인 i3.16xlarge가 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 US East (N. Virginia)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.

- 시나리오 1: 단일 계정의 예약 인스턴스
- 시나리오 2: 정규화 인수를 사용하는 단일 계정의 예약 인스턴스
- 시나리오 3: 연결된 계정의 리전 예약 인스턴스
- 시나리오 4: 연결된 계정의 영역 예약 인스턴스

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.large는 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 사용량에 전체 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.xlarge는 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.

시나리오 2: 정규화 인수를 사용하는 단일 계정의 예약 인스턴스

계정 A에서 다음 온디맨드 인스턴스를 실행 중입니다.

- 2 x m3.xlarge Amazon Linux, 가용 영역 us-east-1a의 기본 테넌시 인스턴스
- 2 x m3.large Amazon Linux, 가용 영역 us-east-1b의 기본 테넌시 인스턴스

계정 A에서 다음 예약 인스턴스를 구입합니다.

- 1 x m3.2xlarge Amazon Linux, us-east-1 리전의 기본 테넌시 예약 인스턴스

예약 인스턴스의 혜택은 다음과 같이 적용됩니다.

- us-east-1의 m3.2xlarge 리전 예약 인스턴스는 기존 테넌시가 포함된 리전 Amazon Linux 예약 인스턴스이므로 가용 영역 및 인스턴스 크기 유연성을 제공합니다. 정규화 인수를 기반으로 인스턴

스 패밀리 내에서 가장 작은 인스턴스 크기부터 가장 큰 인스턴스 크기까지 적용하기 때문에 먼저 m3.large 인스턴스에 적용한 다음 m3.xlarge 인스턴스에 적용합니다.

m3.large 인스턴스는 시간당 정규화 단위 4개와 같습니다.

m3.xlarge 인스턴스는 시간당 정규화 단위 8개와 같습니다.

m3.2xlarge 인스턴스는 시간당 정규화 단위 16개와 같습니다.

혜택은 다음과 같이 적용됩니다.

m3.2xlarge 리전 예약 인스턴스는 2 x m3.large 사용량에 대한 모든 이점을 제공합니다. 이러한 인스턴스가 함께 시간당 8개의 정규화된 단위를 차지하기 때문입니다. 이렇게 하면 m3.xlarge 인스턴스에 적용할 8개의 정규화된 단위/시간이 남습니다.

나머지 8개의 정규화된 단위/시간과 함께 m3.2xlarge 리전 예약 인스턴스는 각 m3.xlarge 인스턴스가 8개의 정규화된 단위/시간과 동일하기 때문에 1 x m3.xlarge 사용량에 대한 모든 이점을 제공합니다. 나머지 m3.xlarge 사용량은 온디맨드 요금이 부과됩니다.

시나리오 3: 연결된 계정의 리전 예약 인스턴스

예약 인스턴스가 구입 계정 내 사용량에 먼저 적용된 후 조직의 다른 계정에서 해당하는 사용량에 적용됩니다. 자세한 내용은 예약 인스턴스 및 통합 결제 섹션을 참조하세요. 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.xlarge가 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.

시나리오 4: 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 계정에 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 예제는 이를 설명하는 데 도움이 될 수 있습니다.

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 AWS Cost and Usage Report의 Understanding your reservations를 참조하세요.

> **Note**
>
> 영역 예약 인스턴스는 소유 계정에 대해서만 용량을 예약하며 다른 AWS 계정과 공유할 수 없습니다. 다른 AWS 계정과 용량을 공유해야 하는 경우 EC2 온디맨드 용량 예약으로 컴퓨팅 용량 예약를 사용하세요.

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 온디맨드 인스턴스를 시작할 경우 온디맨드 인스턴스를 구성할 때 다음 정보를 지정해야 합니다.

플랫폼

예약 인스턴스의 플랫폼(제품 설명)과 일치하는 Amazon Machine Image(AMI)를 지정해야 합니다. 예를 들어 예약 인스턴스에 대해 `Linux/UNIX`를 지정한 경우 Amazon Linux AMI 또는 Ubuntu AMI에서 인스턴스를 시작할 수 있습니다.

인스턴스 유형

영역 예약 인스턴스를 구매한 경우 예약 인스턴스(예: `t3.large`)와 동일한 인스턴스 유형을 지정해야 합니다. 자세한 내용은 영역 예약 인스턴스 적용 방식 섹션을 참조하세요.

리전 예약 인스턴스를 구매한 경우 예약 인스턴스의 인스턴스 유형과 동일한 인스턴스 패밀리의 인스턴스 유형을 지정해야 합니다. 예를 들어 예약 인스턴스에 대해 `t3.xlarge`를 지정한 경우 T3 패

밀리에서 인스턴스를 시작해야 하지만 `t3.medium`과 같이 모든 크기를 지정할 수 있습니다. 자세한 내용은 리전 예약 인스턴스 적용 방식 섹션을 참조하세요.

가용 영역

특정 가용 영역에 대해 영역 예약 인스턴스를 구입한 경우 동일한 가용 영역으로 해당 인스턴스를 시작해야 합니다.

리전 단위의 예약 인스턴스를 구입한 경우 예약 인스턴스로 지정한 리전 내 모든 가용 영역으로 인스턴스를 시작할 수 있습니다.

테넌시

인스턴스의 테넌시(예: `dedicated` 또는 `shared`)는 예약 인스턴스의 테넌시와 일치해야 합니다. 자세한 내용은 Amazon EC2 전용 인스턴스 섹션을 참조하세요.

예를 들어 실행 중인 온디맨드 인스턴스에 예약 인스턴스를 적용하는 방법은 예약 인스턴스 할인 적용 방식 섹션을 참조하세요. 자세한 내용은 Amazon EC2 예약 인스턴스가 AWS 결제에 예상대로 적용되지 않는 이유는 무엇입니까?를 참조하세요.

다양한 방법으로 예약 인스턴스 할인을 사용하는 온디맨드 인스턴스를 시작할 수 있습니다. 다양한 시작 방법에 대한 자세한 내용은 Amazon EC2 인스턴스 시작하기 섹션을 참조하세요. 또한 Amazon EC2 Auto Scaling을 사용하여 인스턴스를 시작할 수 있습니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서를 참조하세요.

예약 인스턴스 결제 작동 방식

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 옵션에 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 대기열에 추가 섹션을 참조하세요.

AWS 프리 티어는 신규 AWS 계정에 사용할 수 있습니다. AWS 프리 티어를 사용하여 Amazon EC2 인스턴스를 실행하며 예약 인스턴스를 구입하는 경우 표준 요금이 부과됩니다. 자세한 내용은 AWS 프리 티어 섹션을 참조하세요.

내용

- 사용량 결제
- 청구서 보기
- 예약 인스턴스 및 통합 결제
- 예약 인스턴스 할인 요금 티어

사용량 결제

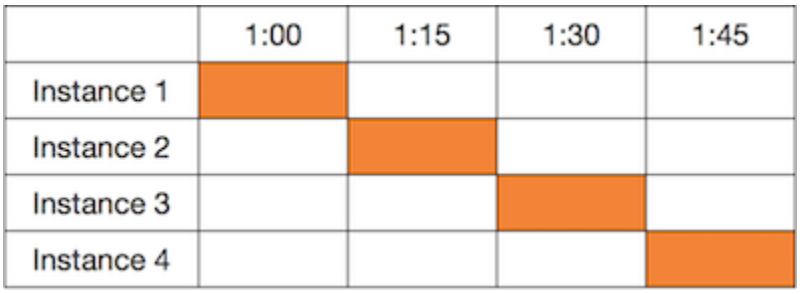
예약 인스턴스는 선택한 기간 동안 인스턴스 실행 여부와 상관없이 매 시간 청구됩니다. 각 시간은 표준 24시간 시계의 정각(0분 0초)에 시작합니다. 예를 들어 1:00:00부터 1:59:59까지가 1시간입니다. 인스턴스 상태에 대한 자세한 내용은 Amazon EC2 인스턴스 상태 변경 단원을 참조하세요.

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.

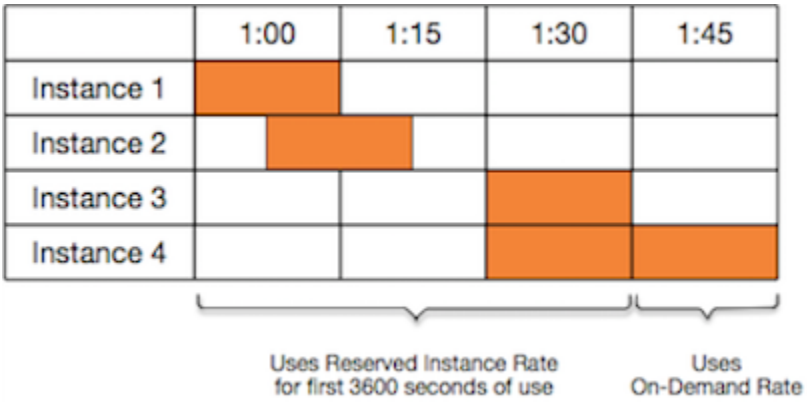
예약 인스턴스 결제 혜택은 매 시간마다 최대 3,600초(1시간)의 인스턴스 사용량에 적용될 수 있습니다. 여러 인스턴스를 동시에 실행할 수 있지만, 예약 인스턴스 할인 혜택은 매 시간마다 총 3,600초에 대해서만 받을 수 있고 매 시간 3,600초를 초과하는 인스턴스 사용량에는 온디맨드 요금이 청구됩니다.

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 15분(900초)씩 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 결제 및 비용 관리 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 예약 파악을 참조하세요.

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 Organizations의 통합 결제 섹션을 참조하세요.

예약 인스턴스를 구매한 계정을 해지하면 예약 인스턴스가 만료될 때까지 지급인 계정에 예약 인스턴스 요금이 계속 청구됩니다. 해지된 계정은 90일 후에 영구적으로 삭제되며, 멤버 계정에는 더 이상 예약 인스턴스 결제 할인의 혜택이 적용되지 않습니다.

> **Note**
>
> 영역 예약 인스턴스는 소유 계정에 대해서만 용량을 예약하며 다른 AWS 계정과 공유할 수 없습니다. 다른 AWS 계정과 용량을 공유해야 하는 경우 EC2 온디맨드 용량 예약으로 컴퓨팅 용량 예약를 사용하세요.

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 Server를 사용하는 Linux용 예약 인스턴스에는 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
- 구매 시 할인 티어 적용
- 요금 티어 교차
- 요금 티어 통합 결제

예약 인스턴스 요금 할인 계산

리전의 모든 예약 인스턴스에 대한 정가를 계산하여 계정에 대한 요금 티어를 확인할 수 있습니다. 각 예약의 시간당 부과 요금(hourly recurring price)에 약정 기간의 총 시간을 곱한 다음, 구매 시 할인이 적용되지 않은 선결제 금액(fixed price: 고정 가격이라고도 함)을 더합니다. 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. 선결제 요금 열을 표시하려면 오른쪽 상단 모서리에서 설정 ( ) 을 선택하고 선결제 요금을 켠 다음, 확인을 선택합니다.

명령줄을 사용하여 예약 인스턴스의 고정 가격을 확인하려면

- describe-reserved-instances(AWS CLI)
- Get-EC2ReservedInstance(AWS Tools for Windows PowerShell)

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 Amazon EC2용 예약 인스턴스 구입 섹션을 참조하세요.

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 섹션을 참조하세요.

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 섹션을 참조하세요.

Amazon EC2용 예약 인스턴스 구입

Amazon EC2용 예약 인스턴스를 구입하려면 Amazon EC2 콘솔, 명령줄 도구 또는 SDK를 사용하여 AWS와 타사 판매자의 예약 인스턴스 제품을 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. Amazon EC2 예약 인스턴스 마켓플레이스에서 타사 판매자의 예약 인스턴스를 구매할 때 선택한 것과 비슷하지만 선결제 가격이 더 낮은 제품이 있는 경우 AWS에서는 더 저렴한 선결제 가격으로 해당 제품을 판매합니다.

구입을 확정하기 전에 구매하기로 결정한 예약 인스턴스의 세부 정보를 검토하고 모든 파라미터가 정확한지 확인하세요. 예약 인스턴스 Marketplace의 서드 파티 판매자 또는 AWS로부터 예약 인스턴스를 구매한 후에는 구매를 취소할 수 없습니다. 구매를 미래 날짜로 대기열에 넣고 예정된 시간 전에 대기 중인 구매를 취소할 수 있습니다.

예약 인스턴스를 구매하고 수정하려면 가용 영역을 설명할 수 있는 권한과 같은 적절한 권한이 사용자에게 있는지 확인하세요. 자세한 내용은 the section called “예약 인스턴스 작업”(API) 또는 the section called “예약 인스턴스 작업”(콘솔)를 참조하세요.

업무

- 플랫폼 선택
- 구매 대기열에 추가
- 스탠다드 예약 인스턴스 구매
- 전환형 예약 인스턴스 구매
- 예약 인스턴스 Marketplace에서 구매
- 대기 중인 구매 취소
- 예약 인스턴스 갱신

플랫폼 선택

Amazon EC2는 예약 인스턴스에 대해 다음 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
- Red Hat Enterprise Linux with HA
- Windows
- SQL Server Standard가 설치된 Windows

- SQL Server Web이 설치된 Windows
- SQL Server Enterprise가 설치된 Windows

고려 사항

- Red Hat Enterprise Linux, SUSE Linux 또는 Ubuntu Pro용 기존 구독(BYOS)을 가져오는 경우 Linux/Unix 플랫폼용 서비스를 선택해야 합니다.
- macOS 또는 Ubuntu Pro(EC2 구독 포함, 즉 BYOS가 아님)를 실행하는 인스턴스에서는 예약 인스턴스가 지원되지 않습니다. 온디맨드 인스턴스 요금으로 비용 절감을 원한다면 절감형 플랜으로 macOS 및 Ubuntu Pro(EC2 구독 포함)를 사용하는 것이 좋습니다. 자세한 내용은 절감형 플랜 사용 설명서를 참조하세요.

인스턴스가 특정 예약 인스턴스에서 실행되도록 하려면 예약 인스턴스의 플랫폼이 인스턴스를 시작하는 데 사용된 AMI의 플랫폼과 일치해야 합니다. Linux AMI의 경우 AMI 플랫폼이 일반 값 Linux/UNIX를 사용하는지, 아니면 SUSE Linux와 같은 보다 구체적인 값을 사용하는지 확인하는 것이 중요합니다.

Console

AMI 플랫폼을 확인하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 AMI를 선택합니다.
3. AMI를 선택합니다.
4. 세부 정보 탭에서 플랫폼 세부 정보의 값을 기록해 둡니다.

AWS CLI

AMI 플랫폼을 확인하려면

describe-images 명령을 사용하고 `PlatformDetails` 값을 확인합니다.

```
aws ec2 describe-images \
    --image-id ami-0abcdef1234567890 \
    --query Images[*].PlatformDetails
```

출력의 예시는 다음과 같습니다.

```
[
```

```
    "Linux/UNIX"
]
```

PowerShell

AMI 플랫폼을 확인하려면

Get-EC2Image cmdlet을 사용하고 PlatformDetails 값을 확인합니다.

```
Get-EC2Image `
    -ImageId ami-0abcdef1234567890 | `
    Select PlatformDetails
```

출력의 예시는 다음과 같습니다.

```
PlatformDetails
---------------
Linux/UNIX
```

구매 대기열에 추가

기본적으로 예약 인스턴스를 구매할 때는 즉시 구매됩니다. 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.

Amazon EC2 콘솔에서 대기 중인 구매에 대한 날짜를 설정할 수 있으며, 해당 날짜의 UTC 00:00까지 구매가 대기됩니다. 대기 중인 구매에 대해 다른 시간을 지정하려면 AWS SDK 또는 명령줄 도구를 사용합니다.

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. 자세한 내용은 대기 중인 구매 취소 섹션을 참조하세요.

스탠다드 예약 인스턴스 구매

특정 가용 영역에서 표준 예약 인스턴스를 구입하고 용량을 예약할 수 있습니다. 또는 용량 예약을 포기하고 리전 단위의 표준 예약 인스턴스를 구입할 수 있습니다;.

구매를 완료했으며 예약 인스턴스의 사양과 일치하는 인스턴스가 이미 실행 중인 경우 결제 혜택이 즉시 적용됩니다. 인스턴스를 재시작할 필요가 없습니다. 실행 중인 적합 인스턴스가 없는 경우, 인스턴스를 시작하고 예약 인스턴스에 대해 지정한 조건과 정확히 일치하는지 확인합니다. 자세한 내용은 예약 인스턴스 사용 섹션을 참조하세요.

예를 들어 실행 중인 인스턴스에 예약 인스턴스를 적용하는 방법은 예약 인스턴스 할인 적용 방식 섹션을 참조하세요.

Console

표준형 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 [표준(Standard)]을 선택하여 표준 예약 인스턴스을(를) 표시합니다.
4. 용량 예약을 구입하려면 구입 화면의 상단 오른쪽 모서리 부분에서 용량이 예약된 제공만 표시를 켭니다. 이 설정을 켜면 가용 영역 필드가 나타납니다.

   리전 예약 인스턴스을(를) 구입하려면 이 설정을 끕니다. 이 설정을 끄면 가용 영역 필드가 사라집니다.
5. 필요에 따라 다른 구성을 선택하고 [검색(Search)]을 선택합니다.
6. 구매하려는 각 예약 인스턴스에 대해 원하는 수량을 입력하고 [장바구니에 추가(Add to Cart)]를 선택합니다.

   예약 인스턴스 Marketplace에서 표준 예약 인스턴스를 구매하려면 검색 결과의 [판매자(Seller)] 열에서 [서드 파티(3rd party)]를 찾습니다. 기간 열에 비 표준 약정이 표시됩니다. 자세한 내용은 예약 인스턴스 Marketplace에서 구매 섹션을 참조하세요.
7. 선택한 예약 인스턴스의 요약을 보려면 [장바구니 보기(View cart)]를 선택합니다.
8. 주문 시각(Order on)이 지금(Now)인 경우 [모두 주문(Order all)]을 선택한 직후 구매가 완료됩니다. 구매를 대기시키려면 지금을 선택하고 날짜를 선택하십시오. 장바구니에서 적합한 각 상품에 대해 다른 날짜를 선택할 수 있습니다. 구매는 선택한 날짜의 00:00(UTC)까지 대기열에 배치됩니다.
9. 주문을 완료하려면 [모두 주문(Order all)]을 선택합니다.

   주문 당시 선택한 조건과 비슷하지만 가격이 더 낮은 상품이 있는 경우 AWS는 더 저렴한 상품을 판매합니다.
10. 닫기를 선택하세요.

주문 상태가 상태 열에 나열됩니다. 주문이 완료되면 상태 값이 Payment-pending에서 Active(으)로 바뀝니다. 예약 인스턴스이(가) Active인 경우 사용할 준비가 된 것입니다.

상태가 Retired로 바뀌면 AWS에서 결제를 수신하지 못한 것일 수 있습니다.

AWS CLI

표준형 예약 인스턴스를 구매하려면

1. describe-reserved-instances-offerings 명령을 사용하여 사용 가능한 예약 인스턴스를 찾습니다. 표준형 예약 인스턴스만 반환하도록 --offering-class 옵션에 standard를 지정합니다. 추가 기준을 적용하여 결과를 좁힐 수 있습니다. 예를 들어 1년 동안만 Linux/UNIX에 대해 기본 테넌시가 포함된 리전 단위의 t2.large 예약 인스턴스를 구매하려는 경우 다음 명령을 사용합니다.

```
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 Marketplace의 예약 인스턴스만 찾으려면 marketplace 필터를 사용하고 요청에 기간을 지정하지 않습니다. 기간이 1년 또는 3년 기간보다 짧을 수 있기 때문입니다.

```
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 이전 직접 호출에 다음 옵션을 추가합니다.

```
--purchase-time "2020-12-01T00:00:00Z"
```

3. describe-reserved-instances 명령을 사용하여 예약 인스턴스의 상태를 가져옵니다.

```
aws ec2 describe-reserved-instances \
    --reserved-instances-ids b847fa93-e282-4f55-b59a-1342fec06327 \
    --query ReservedInstances[].State
```

PowerShell

표준형 예약 인스턴스를 구매하려면

1. Get-EC2ReservedInstancesOffering cmdlet을 사용하여 사용 가능한 예약 인스턴스를 찾습니다. `standard` 파라미터에 대해 `-OfferingClass`를 지정하여 표준 예약 인스턴스만 반환합니다. 추가 기준을 적용하여 결과를 좁힐 수 있습니다. 예를 들어 1년 동안만 `Linux/UNIX`에 대해 기본 테넌시가 포함된 리전 단위의 `t2.large` 예약 인스턴스를 구매하려는 경우 다음 명령을 사용합니다.

```
Get-EC2ReservedInstancesOffering `
    -InstanceType "t2.large" `
    -OfferingClass "standard" `
    -ProductDescription "Linux/UNIX" `
    -InstanceTenancy "default" `
    -Filters @{Name="duration"; Values="31536000"} `
             @{Name="scope"; Values="Region"
```

예약 인스턴스 Marketplace의 예약 인스턴스만 찾으려면 `marketplace` 필터를 사용하고 요청에 기간을 지정하지 않습니다. 기간이 1년 또는 3년 기간보다 짧을 수 있기 때문입니다.

```
Get-EC2ReservedInstancesOffering `
    -InstanceType t2.large `
```

```
    -OfferingClass "standard" `
    -ProductDescription "Linux/UNIX" `
    -InstanceTenancy default `
    -Filters @{Name="marketplace"; Values="true"}
```

요구 사항에 맞는 예약 인스턴스를 찾은 경우 상품 ID를 기록합니다. 예시:

```
bec624df-a8cc-4aad-a72f-4f8abc34caf2
```

2. New-EC2ReservedInstance cmdlet을 사용하여 예약 인스턴스를 구매합니다. 이전 단계에서 얻은 예약 인스턴스 상품 ID를 지정하고 예약을 위한 인스턴스 수를 지정해야 합니다.

```
New-EC2ReservedInstance `
    -ReservedInstancesOfferingId "bec624df-a8cc-4aad-a72f-4f8abc34caf2" `
    -InstanceCount 1
```

기본적으로 구매는 즉시 완료됩니다. 또는 구매를 대기시키려면 다음 매개 변수를 이전 호출에 추가하세요.

```
-PurchaseTime "2020-12-01T00:00:00Z"
```

3. Get-EC2ReservedInstance cmdlet을 사용하여 예약 인스턴스의 상태를 가져옵니다.

```
Get-EC2ReservedInstance `
    -ReservedInstancesId b847fa93-e282-4f55-b59a-1342fec06327 | `
    Select State
```

전환형 예약 인스턴스 구매

특정 가용 영역에서 전환형 예약 인스턴스를 구입하고 용량을 예약할 수 있습니다. 또는 용량 예약을 포기하고 리전 단위의 전환형 예약 인스턴스를 구입할 수 있습니다;.

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 섹션을 참조하세요.

예를 들어 실행 중인 인스턴스에 예약 인스턴스를 적용하는 방법은 예약 인스턴스 할인 적용 방식 섹션을 참조하세요.

Console

전환형 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 클래스에 대해 [컨버터블(Convertible)]을 선택하여 전환형 예약 인스턴스을(를) 표시합니다.
4. 용량 예약을 구입하려면 구입 화면의 상단 오른쪽 모서리 부분에서 용량이 예약된 제공만 표시를 켭니다. 이 설정을 켜면 가용 영역 필드가 나타납니다.

   리전 예약 인스턴스을(를) 구입하려면 이 설정을 끕니다. 이 설정을 끄면 가용 영역 필드가 사라집니다.
5. 필요에 따라 다른 구성을 선택하고 검색을 선택합니다.
6. 구매하려는 각 전환형 예약 인스턴스에 대해 수량을 입력하고 [장바구니에 추가(Add to Cart)]를 선택합니다.
7. 선택한 내역을 보려면 [장바구니 보기(View cart)]를 선택합니다.
8. 주문 시각(Order on)이 지금(Now)인 경우 [모두 주문(Order all)]을 선택한 직후 구매가 완료됩니다. 구매를 대기시키려면 지금을 선택하고 날짜를 선택하십시오. 장바구니에서 적합한 각 상품에 대해 다른 날짜를 선택할 수 있습니다. 구매는 선택한 날짜의 00:00(UTC)까지 대기열에 배치됩니다.
9. 주문을 완료하려면 [모두 주문(Order all)]을 선택합니다.

   주문 당시 선택한 조건과 비슷하지만 가격이 더 낮은 상품이 있는 경우 AWS는 더 저렴한 상품을 판매합니다.
10. 닫기를 선택하세요.

    주문 상태가 상태 열에 나열됩니다. 주문이 완료되면 상태 값이 `Payment-pending`에서 `Active`(으)로 바뀝니다. 예약 인스턴스이(가) `Active`인 경우 사용할 준비가 된 것입니다.

    상태가 `Retired`로 바뀌면 AWS에서 결제를 수신하지 못한 것일 수 있습니다.

AWS CLI

전환형 예약 인스턴스를 구매하려면

1. describe-reserved-instances-offerings 명령을 사용하여 사용 가능한 예약 인스턴스를 찾습니다. 전환형 예약 인스턴스만 반환하도록 `--offering-class` 옵션에 `convertible`을 지정합니다. 추가 기준을 적용하여 결과를 좁힐 수 있습니다. 예를 들어 `Linux/UNIX`에 대해 기본 테넌시가 포함된 리전 단위의 `t2.large` 예약 인스턴스를 구매하려는 경우 다음 명령을 사용합니다.

```
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 이전 직접 호출에 다음 옵션을 추가합니다.

```
--purchase-time "2020-12-01T00:00:00Z"
```

3. describe-reserved-instances 명령을 사용하여 예약 인스턴스의 상태를 가져옵니다.

```
aws ec2 describe-reserved-instances \
    --reserved-instances-ids b847fa93-e282-4f55-b59a-1342fec06327 \
    --query ReservedInstances[].State
```

PowerShell

전환형 예약 인스턴스를 구매하려면

1. Get-EC2ReservedInstancesOffering cmdlet을 사용하여 사용 가능한 예약 인스턴스를 찾습니다. `convertible` 파라미터에 대해 `-OfferingClass`를 지정하여 전환형 예약 인스턴스만 반환합니다. 추가 기준을 적용하여 결과를 좁힐 수 있습니다. 예를 들어 `Linux/UNIX`에 대해 기본 테넌시가 포함된 리전 단위의 `t2.large` 예약 인스턴스를 구매하려는 경우 다음 명령을 사용합니다.

```
Get-EC2ReservedInstancesOffering `
    -InstanceType "t2.large" `
    -OfferingClass "convertible" `
    -ProductDescription "Linux/UNIX" `
    -InstanceTenancy "default" `
    -Filters @{Name="scope"; Values="Region"
```

요구 사항에 맞는 예약 인스턴스를 찾은 경우 상품 ID를 기록합니다. 예시:

```
bec624df-a8cc-4aad-a72f-4f8abc34caf2
```

2. New-EC2ReservedInstance cmdlet을 사용하여 예약 인스턴스를 구매합니다. 이전 단계에서 얻은 예약 인스턴스 상품 ID를 지정하고 예약을 위한 인스턴스 수를 지정해야 합니다.

```
New-EC2ReservedInstance `
    -ReservedInstancesOfferingId "bec624df-a8cc-4aad-a72f-4f8abc34caf2" `
    -InstanceCount 1
```

기본적으로 구매는 즉시 완료됩니다. 또는 구매를 대기시키려면 다음 매개 변수를 이전 호출에 추가하세요.

```
-PurchaseTime "2020-12-01T00:00:00Z"
```

3. Get-EC2ReservedInstance cmdlet을 사용하여 예약 인스턴스의 상태를 가져옵니다.

```
Get-EC2ReservedInstance `
    -ReservedInstancesId b847fa93-e282-4f55-b59a-1342fec06327 | `
    Select State
```

예약 인스턴스 Marketplace에서 구매

예약 인스턴스 Marketplace에서 예약 인스턴스를 소유하고 있지만 더 이상 필요로 하지 않는 서드 파티 판매자로부터 예약 인스턴스를 구매할 수 있습니다. Amazon EC2 콘솔 또는 명령줄 도구를 사용하여 이 작업을 수행할 수 있습니다. 이 프로세스는 AWS에서 예약 인스턴스를 구매하는 것과 비슷합니다. 자세한 내용은 스탠다드 예약 인스턴스 구매 섹션을 참조하세요.

예약 인스턴스 Marketplace에서 구매한 예약 인스턴스와 AWS로부터 직접 구매한 예약 인스턴스 간에는 몇 가지 차이가 있습니다.

- 기간 - 서드 파티로부터 구매하는 예약 인스턴스는 남은 기간이 표준 약정 기간보다 짧습니다. AWS의 표준 약정 기간은 1년 또는 3년입니다.
- 선결제 가격 - 서드 파티 예약 인스턴스는 다양한 선결제 가격으로 판매될 수 있습니다. 사용 요금이나 기본 요금은 AWS에서 예약 인스턴스를 처음 구매할 때 설정된 요금과 동일하게 유지됩니다.
- 예약 인스턴스 유형 - 예약 인스턴스 Marketplace에서는 Amazon EC2 표준 예약 인스턴스만 구매할 수 있습니다. 컨버터블 예약 인스턴스, Amazon RDS 및 Amazon ElastiCache 예약 인스턴스는 예약 인스턴스 Marketplace에서 구매할 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 지원(으)로 문의하세요.

대기 중인 구매 취소

최대 3년 전에 구매를 대기할 수 있습니다. 예약된 시간 전에 언제든지 대기 중인 구매를 취소할 수 있습니다.

Console

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. [작업(Actions), [대기 중인 예약 인스턴스 삭제(Delete queued Reserved Instances)를 차례로 선택합니다.
5. 확인 메시지가 나타나면 [삭제(Delete)], [닫기(Close)]를 차례로 클릭합니다.

AWS CLI

대기 중인 구매 취소

delete-queued-reserved-instances 명령을 사용합니다.

```
aws ec2 delete-queued-reserved-instances \
    --reserved-instances-ids b847fa93-e282-4f55-b59a-1342fec06327
```

PowerShell

대기 중인 구매 취소

Remove-EC2QueuedReservedInstance cmdlet을 사용합니다.

```
Remove-EC2QueuedReservedInstance `
    -ReservedInstancesId b847fa93-e282-4f55-b59a-1342fec06327
```

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.

Amazon EC2 콘솔을 사용하여 예약 인스턴스를 갱신해야 합니다.

대기 중인 구매를 사용하여 예약 인스턴스 갱신

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 예약 인스턴스를 선택합니다.
3. 갱신할 예약 인스턴스를 선택합니다.
4. 작업(Actions), 예약 인스턴스 갱신(Renew Reserved Instances)를 선택합니다.
5. 주문을 완료하려면 [모두 주문(Order all)]을 선택한 다음 [닫기(Close)]를 선택합니다.

예약 인스턴스 마켓플레이스에서 Amazon EC2용 예약 인스턴스 판매

Amazon EC2 예약 인스턴스 마켓플레이스는 AWS 고객 및 타사 판매자의 미사용 표준형 예약 인스턴스를 쉽게 판매할 수 있는 플랫폼입니다. 이러한 예약 인스턴스는 기간 및 요금 옵션이 다를 수 있습니다. 인스턴스를 새 AWS 리전으로 이동하거나, 다른 인스턴스 유형으로 변경하거나, 예약 인스턴스 기간이 만료되기 전에 프로젝트를 완료하거나, 비즈니스 요구 사항이 변경되거나, 용량이 초과되는 경우 등 더 이상 필요하지 않은 경우 예약 인스턴스를 판매할 수 있습니다.

예약 인스턴스 Marketplace에 예약 인스턴스를 등록하는 즉시 잠재적 구매자들이 예약 인스턴스를 찾을 수 있습니다. 모든 예약 인스턴스는 남은 약정 기간 및 시간당 요금에 따라 분류됩니다.

예약 인스턴스 마켓플레이스를 통해 타사 판매자의 예약 인스턴스를 구매하려는 구매자의 요청을 이행하기 위해 AWS에서는 지정된 그룹에서 선결제 가격이 가장 낮은 예약 인스턴스를 판매합니다. AWS는 구매자의 주문이 모두 이행될 때까지 그다음 낮은 가격으로 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. 구매자가 인스턴스를 계속 사용하는 경우 AWS는 해당 예약 인스턴스가 판매된 시점부터 온디맨드 요금을 적용합니다.

예약 인스턴스 Marketplace에서 미사용 예약 인스턴스를 판매하려면 특정 자격 기준을 충족해야 합니다.

예약 인스턴스 Marketplace의 예약 인스턴스 구매에 대한 자세한 내용은 예약 인스턴스 Marketplace에서 구매 섹션을 참조하세요.

목차

- 규제 및 제한
- 판매자로 등록
- 지급금 은행 계좌
- 세금 정보
- 예약 인스턴스 요금 책정
- 예약 인스턴스 나열
- 예약 인스턴스 항목 상태
- 항목 기간
- 예약 인스턴스 판매 후 절차

- 판매 대금 정산
- 구매자와의 정보 공유

규제 및 제한

미사용 예약을 판매하려면 먼저 예약 인스턴스 Marketplace에서 판매자로 등록해야 합니다. 자세한 내용은 판매자로 등록 단원을 참조하세요.

예약 인스턴스 판매 시 다음과 같은 제한 및 제약이 적용됩니다.

- 예약형 인스턴스 Marketplace에서는 Amazon EC2 표준 리전 및 영역 예약형 인스턴스만 판매할 수 있습니다.
- 예약 인스턴스 마켓플레이스에서는 Amazon EC2 컨버터블 예약 인스턴스를 판매할 수 없습니다.
- 다른 AWS 서비스(예: Amazon RDS 및 Amazon ElastiCache)에 대한 예약형 인스턴스는 예약형 인스턴스 Marketplace에서 판매할 수 없습니다.
- 표준 예약 인스턴스의 남은 사용 기간이 한 달 이상이어야 합니다.
- 기본적으로 비활성화된 리전에서는 스탠다드 예약 인스턴스를 판매할 수 없습니다.
- 예약 인스턴스 Marketplace에서 허용되는 최소 가격은 0.00 USD입니다.
- 선결제 없음, 부분 선결제 또는 전체 선결제 예약 인스턴스는 최소 30일 동안 계정에서 활성 상태인 경우 예약 인스턴스 마켓플레이스에서 판매할 수 있습니다. 또한 예약 인스턴스에 선불 지급이 있는 경우 AWS에서 선불 지급을 받은 후에만 예약 인스턴스를 판매할 수 있습니다.
- 대량 구매 할인을 사용하여 예약 인스턴스를 구매한 경우 예약 인스턴스 마켓플레이스에서 예약 인스턴스를 판매할 수 없습니다.
- 예약 인스턴스 Marketplace의 리스팅을 직접 수정할 수는 없습니다. 하지만 판매 등록을 취소하고 새 파라미터를 지정한 다음 다시 등록하는 방식으로 변경하는 것은 가능합니다. 자세한 내용은 예약 인스턴스 요금 책정 단원을 참조하세요. 판매 등록하기 전에 예약 인스턴스를 수정할 수도 있습니다. 자세한 내용은 예약 인스턴스 수정 단원을 참조하세요.
- AWS는 예약 인스턴스 Marketplace에서 판매하는 각 표준 예약 인스턴스에 대해 총 선결제 가격의 12%를 서비스 수수료로 청구합니다. 선결제 금액은 판매자가 판매 등록한 표준 예약 인스턴스에 책정한 가격입니다;.
- 판매자로 등록하는 경우 지정한 은행에 미국 주소가 있어야 합니다. 자세한 내용은 AWS Marketplace 판매자 가이드에서 유료 제품에 대한 추가 판매자 요구 사항을 참조하세요.
- Amazon Web Services India Private Limited(AWS India) 고객은 미국 은행 계좌를 가지고 있더라도 예약 인스턴스 Marketplace에서 예약 인스턴스를 판매할 수 없습니다. 자세한 내용은 AWS 계정과 AWS India 계정의 차이점은 무엇인가요?를 참조하세요.

판매자로 등록

> **Note**
>
> AWS 계정 루트 사용자만 계정을 판매자로 등록할 수 있습니다.

예약 인스턴스 Marketplace에서 판매하려면 먼저 판매자로 등록해야 합니다. 등록 과정에서 다음 정보를 제공해야 합니다.

- 은행 정보 - AWS에서 예약 인스턴스를 판매하는 경우 판매 대금을 지급하기 위해 사용자의 은행 정보가 필요합니다. 이때 미국 소재지가 있는 은행을 선택해야 합니다. 자세한 내용은 지급금 은행 계좌 섹션을 참조하세요.
- 세금 정보 - 모든 판매자는 세금 신고 의무를 결정하기 위해서 세금 신고서를 작성해야 합니다. 자세한 내용은 세금 정보 섹션을 참조하세요.

AWS에서 판매자 등록에 필요한 과정을 모두 마치면 등록 확인과 함께 예약 인스턴스 Marketplace에서 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. 자세한 내용은 AWS Marketplace 판매자 가이드에서 유료 제품에 대한 추가 판매자 요구 사항을 참조하세요.

지급금을 받을 기본 은행 계좌를 등록하려면

1. [예약 인스턴스 Marketplace 판매자 등록(Reserved Instance Marketplace Seller Registration)] 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 완료되는 데 보통 2일 정도 걸립니다.

지급금을 받을 기본 은행 계좌를 변경하려면

1. [예약 인스턴스 Marketplace 판매자 등록(Reserved Instance Marketplace Seller Registration)] 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 확인하세요. 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 참고하세요.

- 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 상담하세요.
- IRS 세금 신고 규정을 가장 효율적으로 준수할 수 있는 방법은 인터뷰에 나오는 모든 질문에 답변하고 요청된 모든 정보를 제공하는 것입니다.
- 답변을 확인하세요. 오타나 사업자 등록 번호가 잘못 기재되지 않도록 유의해야 합니다. 이에 따라 세금 신고서를 다시 작성해야 할 수 있습니다.

판매자의 세금 인터뷰 답변 및 IRS 보고 임계값에 따라 Amazon은 Form 1099-K를 제출할 수 있습니다. Amazon은 세금 계정이 임계 수준에 도달한 년도의 다음 해 1월 31일까지 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 사이트의 Form 1099-K FAQ 섹션을 참조하세요.

예약 인스턴스 요금 책정

예약 인스턴스 가격을 설정할 때 다음 항목을 고려하세요.

- 선결제 가격 - 판매자는 판매할 예약 인스턴스에 대한 선결제 가격만 책정할 수 있습니다. 선결제 가격은 구매자가 각 예약 인스턴스를 구매할 때 지불하는 일회성 가격입니다.

  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 Marketplace에서 허용되는 최소 가격은 0.00 USD입니다.
- 한도 - 다음과 같은 예약 인스턴스 판매 한도는 AWS 계정의 수명이 다할 때까지 적용됩니다. 연간 한도는 아닙니다.
  - 최대 50,000 USD의 예약 인스턴스를 판매할 수 있습니다.
  - 최대 5,000개의 예약 인스턴스를 판매할 수 있습니다.

  이 한도는 일반적으로 늘릴 수 없지만 요청 시 건별로 평가합니다. 한도 증가를 요청하려면 서비스 한도 증가 양식을 작성하세요. 한도 유형은 EC2 예약 인스턴스 판매를 선택합니다.
- 수정 불가 - 판매 등록을 직접 변경할 수 없습니다. 하지만 판매 등록을 취소하고 새 파라미터를 지정한 다음 다시 등록하는 방식으로 변경하는 것은 가능합니다.
- 취소 가능 - 현재 `active` 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.

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. 리스팅의 판매되지 않은 부분만 예약 인스턴스 Marketplace에서 더 이상 사용할 수 없게 됩니다.

Console

예약 인스턴스 Marketplace의 예약 인스턴스 나열

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 예약 인스턴스를 선택합니다.
3. 판매 등록할 예약 인스턴스를 선택하고 작업, 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.

Amazon EC2 콘솔에서 리스팅을 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 페이지 하단에 있는 내 항목 탭을 선택합니다.

AWS CLI

Reserved Instance Marketplace에서 예약 인스턴스를 관리하려면

1. describe-reserved-instances 명령을 사용하여 예약 인스턴스 목록을 가져옵니다. 등록하려는 예약 인스턴스의 ID를 기록해 둡니다.
2. create-reserved-instances-listing 명령을 사용합니다. 예약 인스턴스의 ID, 인스턴스의 수 및 가격표를 지정해야 합니다.
3. 판매 등록을 보려면 describe-reserved-instances-listings 명령을 사용합니다.
4. 리스팅을 취소하려면 cancel-reserved-instances-listing 명령을 사용합니다.

PowerShell

Reserved Instance Marketplace에서 예약 인스턴스를 관리하려면

1. Get-EC2ReservedInstance cmdlet을 사용하여 예약 인스턴스 목록을 가져옵니다. 등록하려는 예약 인스턴스의 ID를 기록해 둡니다.
2. New-EC2ReservedInstancesListing cmdlet을 사용합니다. 예약 인스턴스의 ID, 인스턴스의 수 및 가격표를 지정해야 합니다.
3. 리스팅을 보려면 Get-EC2ReservedInstancesListing cmdlet을 사용합니다.
4. 리스팅을 취소하려면 Stop-EC2ReservedInstancesListing cmdlet을 사용합니다.

예약 인스턴스 항목 상태

다음과 같이 예약 인스턴스 페이지의 내 항목 탭에 있는 항목 상태에 판매 등록의 현재 상태가 표시됩니다.

[리스팅 상태(Listing State)]에 표시되는 정보는 예약 인스턴스 Marketplace의 리스팅 상태에 대한 것입니다. 이 상태 정보는 예약 인스턴스 페이지의 상태 열에 표시되는 상태 정보와는 다릅니다. 이 상태 정보는 보유한 예약의 상태입니다.

- active - 구매 가능한 항목입니다.
- 취소됨(canceled) - 리스팅이 취소되어 예약 인스턴스 Marketplace에서 구매할 수 없습니다.
- closed - 판매 등록되지 않은 예약 인스턴스입니다. 항목 판매가 완료된 예약 인스턴스의 경우에도 상태가 `closed`로 표시됩니다.

항목 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.

리스팅의 일부만 판매된 경우 AWS는 리스팅에서 해당 예약 인스턴스를 사용 중지하고 예약 인스턴스의 남은 개수와 동일한 수의 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. 리스팅에 있는 모든 예약 인스턴스가 판매된 경우 AWS는 리스팅을 `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.

Reserved Instance 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.

Reserved Instance 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. 리스팅의 판매되지 않은 부분만 예약 인스턴스 Marketplace에서 더 이상 사용할 수 없게 됩니다.

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 대금을 송금하는 활동이 포함될 수 있습니다.

Console

예약 인스턴스 리스팅의 상태를 추적하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 예약 인스턴스를 선택합니다.
3. 내 리스팅 탭에서 리스팅 상태의 값을 찾습니다. 또한 이 탭에서 리스팅 기간, 리스팅 가격, 리스팅에서 Available(판매 가능), Pending(보류), Sold(판매), Cancelled(취소) 상태의 인스턴스 개수 정보도 제공됩니다.

AWS CLI

예약 인스턴스 리스팅의 상태를 추적하려면

describe-reserved-instances-listings 명령을 적절한 필터와 함께 사용하여 리스팅에 대한 정보를 얻습니다.

```
aws ec2 describe-reserved-instances-listings
```

PowerShell

예약 인스턴스 리스팅의 상태를 추적하려면

Get-EC2ReservedInstancesListing cmdlet을 사용합니다.

```
Get-EC2ReservedInstancesListing
```

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 유의하세요. 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 Marketplace에서 판매하는 각 표준 예약 인스턴스에 대해 총 선결제 금액의 12%를 서비스 수수료로 청구합니다.

구매자와의 정보 공유

예약 인스턴스 Marketplace에서 판매할 경우 AWS는 미국 규정에 따라 구매자 명세서에 판매자의 상호명을 기재하여 제공합니다. 또한 구매자가 인보이스 또는 다른 세금 관련 이유에 대해 문의하기 위해 지원에 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 지원에 문의하세요.

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 및 세대 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 섹션을 참조하세요.

예약 인스턴스의 전부 또는 하위 집합을 수정할 수 있습니다. 원래 예약 인스턴스를 둘 이상의 새 예약 인스턴스로 분리할 수 있습니다. 예를 들어, us-east-1a에서 10개의 예약을 보유하고 있으며 5개의 인스턴스를 us-east-1b로 옮기는 경우, 수정 요청에 따라 us-east-1a의 인스턴스 5개에 대한 예약 하나와 us-east-1b의 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 섹션을 참조하세요.

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

- 인스턴스 크기 수정을 위한 지원
- 수정 요청 제출
- 수정 요청 문제 해결

수정 요건 및 제한 사항

이러한 속성을 다음과 같이 수정할 수 있습니다.

수정 가능한 속성	지원하는 플랫폼	제한 사항 및 고려 사항
같은 리전 내에서 가용 영역 변경	Linux 및 Windows	-
가용 영역에서 리전으로 범위 변경(반대 방향도 마찬가지)	Linux 및 Windows	영역 예약 인스턴스는 가용 영역으로 범위가 지정되고 해당 가용 영역의 용량을 예약합니다. 범위를 가용 영역에서 리전으로(즉, 영역에서 리전으로) 변경할 경우 용량 예약 혜택을 받을 수 없습니다. 리전 예약 인스턴스는 리전으로 범위가 지정됩니다. 해당 리전의 모든 가용 영역에서 실행되는 인스턴스에 예약 인스턴스 할인이 적용될 수 있습니다. 또한 예약 인스턴스 할인은 선택한 인스턴스 패밀리에 속하는 모든 크기의 인스턴스 사용량에 적용됩니다. 범위를 리전에서 가용 영역으로(즉, 리전에서 영역으로) 변경하면 가용 영역 유연성과 인스턴스 크기 유연성이 사라집니다(있는 경우).

수정 가능한 속성	지원하는 플랫폼	제한 사항 및 고려 사항
		자세한 내용은 예약 인스턴스 할인 적용 방식 섹션을 참조하세요.
동일한 인스턴스 패밀리와 세대 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 인스턴스 크기 수정을 위한 지원 섹션을 참조하세요.

요구 사항

Amazon EC2에서는 새 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 Marketplace에 나열되지 않았습니다.
- 원래 예약과 새 구성의 인스턴스 공간 크기는 일치해야 합니다. 자세한 내용은 인스턴스 크기 수정을 위한 지원 섹션을 참조하세요.
- 원래 예약 인스턴스는 유형 혼합 없이 모두 스탠다드 예약 인스턴스이거나 모두 전환형 예약 인스턴스입니다.
- 원래 예약 인스턴스가 스탠다드 예약 인스턴스인 경우 동일한 시간 내에 만료되어야 합니다.

- 인스턴스 크기를 수정하려는 경우 예약 인스턴스는 인스턴스 크기 유연성을 지원해야 합니다. 인스턴스 크기 유연성을 지원하지 않는 예약 인스턴스 목록은 인스턴스 크기 유연성의 내용을 참조하세요.

인스턴스 크기 수정을 위한 지원

다음 요구 사항이 충족되는 경우 예약 인스턴스의 인스턴스 크기를 수정할 수 있습니다.

요구 사항

- 플랫폼은 Linux/UNIX입니다.
- 동일한 인스턴스 패밀리(예: T 등의 문자로 표시됨)와 세대(예: 2 등의 숫자로 표시됨)의 다른 인스턴스 크기를 선택해야 합니다.

  예를 들어, 예약 인스턴스를 `t2.small`에서 `t2.large`로 수정할 수 있습니다. 둘 다 동일한 T2 제품군 및 세대에 있기 때문입니다. 그러나 T2에서 M2로 또는 T2에서 T3으로 예약 인스턴스를 수정할 수 없습니다. 이 두 가지 경우 모두 대상 인스턴스 패밀리와 세대가 원래 예약 인스턴스와 동일하지 않기 때문입니다.
- 인스턴스 크기 유연성을 지원하는 경우에만 예약 인스턴스 크기를 수정할 수 있습니다. 인스턴스 크기 유연성을 지원하지 않는 예약 인스턴스 목록은 인스턴스 크기 유연성의 내용을 참조하세요.
- `t1.micro`의 크기는 하나뿐이므로 `t1.micro` 인스턴스에 대한 예약 인스턴스의 인스턴스 크기를 수정할 수 없습니다.
- 원본 및 새 예약 인스턴스는 인스턴스 공간 크기가 같아야 합니다.

목차

- 인스턴스 공간 크기
- 베어 메탈 인스턴스에 대한 정규화 인자

인스턴스 공간 크기

각 예약 인스턴스에는 인스턴스 공간 크기가 있으며, 이 공간 크기는 인스턴스 크기의 정규화 인자와 예약된 인스턴스 개수에 따라 결정됩니다. 예약 인스턴스에서 인스턴스 크기를 수정하면 새 구성의 공간이 원래 구성의 공간과 일치해야 합니다. 그렇지 않으면 수정 요청이 처리되지 않습니다.

예약 인스턴스의 인스턴스 공간 크기는 정규화 인자에 인스턴스 수를 곱하여 산출합니다. Amazon EC2 콘솔에서 정규화 인자는 유닛으로 측정됩니다. 다음 표에서는 인스턴스 패밀리의 인스턴스 크기

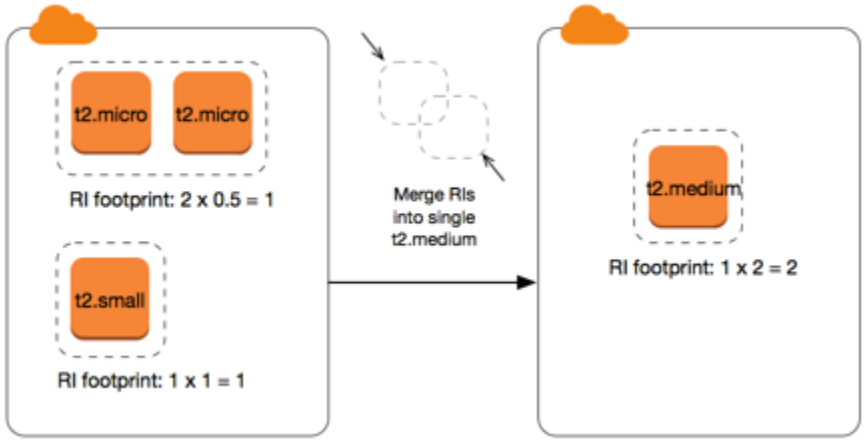
에 대한 정규화 인자를 설명합니다. 예를 들어 `t2.medium`은 정규화 인자 2를 가지므로, `t2.medium` 인스턴스 4개 예약의 공간은 8유닛입니다.

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
48xlarge	384

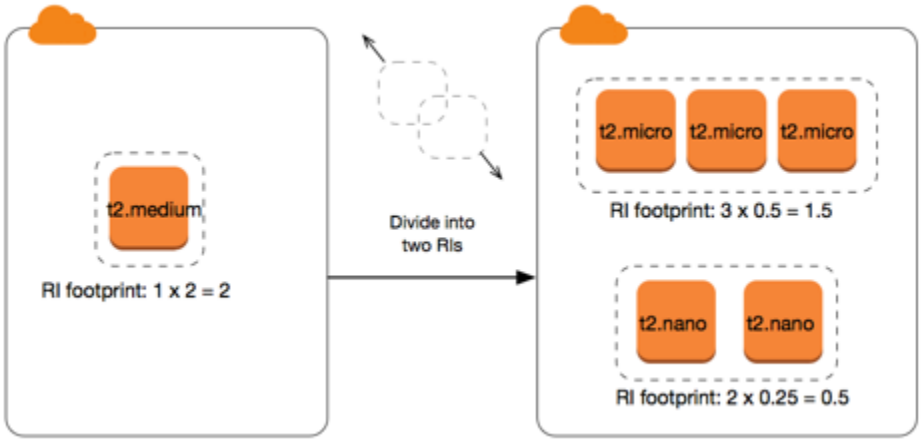
인스턴스 크기	정규화 인자
56xlarge	448
112xlarge	896

예약은 예약의 인스턴스 공간 크기가 변동되지 않는 선에서 동일한 인스턴스 패밀리 내의 다른 인스턴스 크기로 다양하게 할당할 수 있습니다. 예를 들어 `t2.large`(1 @ 4유닛) 인스턴스 1개에 대한 예약을 `t2.small`(4 @ 1유닛) 인스턴스 4개로 나눌 수 있습니다. 마찬가지로 `t2.small` 인스턴스 4개에 대한 예약을 `t2.large` 인스턴스 1개로 결합할 수 있습니다. 그러나 새 예약(4유닛)의 공간이 원래 예약(2유닛)의 공간보다 크기 때문에 `t2.small` 인스턴스 2개에 대한 예약을 `t2.large` 인스턴스 1개로 변경할 수 없습니다.

다음 예에서는 `t2.micro` 인스턴스(1유닛) 2개가 포함된 예약과 `t2.small` 인스턴스(1유닛) 1개가 포함된 예약이 있습니다. 이 두 예약을 `t2.medium` 인스턴스(2유닛) 1개가 포함된 단일 예약에 병합하면 새 예약의 공간이 결합된 예약의 공간과 같습니다.



둘 이상의 예약으로 나누도록 예약을 수정할 수도 있습니다. 다음 예에는 `t2.medium` 인스턴스(2유닛)가 포함된 예약이 있습니다. 이 예약을 `t2.nano` 인스턴스(.5유닛) 2개가 포함된 예약과 `t2.micro` 인스턴스(1.5유닛) 3개가 포함된 예약으로 나눌 수 있습니다.



베어 메탈 인스턴스에 대한 정규화 인자

동일한 인스턴스 패밀리 내의 다른 크기를 사용하여 metal 인스턴스가 포함된 예약을 수정할 수 있습니다. 마찬가지로 동일한 인스턴스 패밀리 내의 metal 크기를 사용하여 베어 메탈 인스턴스가 아닌 인스턴스가 포함된 예약을 수정할 수 있습니다. 일반적으로 베어 메탈 인스턴스는 동일한 인스턴스 패밀리 내에서 사용 가능한 최대 인스턴스 크기와 크기가 같습니다. 예를 들어 i3.metal 인스턴스는 i3.16xlarge 인스턴스와 크기가 동일하므로 정규화 인자도 동일합니다.

다음 표에서는 베어 메탈 인스턴스가 있는 인스턴스 패밀리의 베어 메탈 인스턴스 크기에 대한 정규화 인자를 설명합니다. metal 인스턴스의 정규화 인자는 다른 인스턴스 크기와 달리 인스턴스 패밀리에 따라 다릅니다.

인스턴스 크기	정규화 인자
a1.metal	32
m5zn.metal | x2iezn.metal z1d.metal	96
c6g.metal | c6gd.metal | i3.metal | m6g.metal | m6gd.metal | r6g.metal | r6gd.metal | x2gd.metal	128
c5n.metal	144
c5.metal | c5d.metal | i3en.metal | m5.metal | m5d.metal | m5dn.metal | m5n.metal | r5.metal | r5b.metal | r5d.metal | r5dn.metal | r5n.metal	192

인스턴스 크기	정규화 인자
c6i.metal | c6id.metal | m6i.metal | m6id.metal | r6d.metal | r6id.metal	256
u-18tb1.metal | u-24tb1.metal	448
u-6tb1.metal | u-9tb1.metal | u-12tb1.metal	896

예를 들어 i3.metal 인스턴스의 정규화 인자는 128입니다. i3.metal 기본 테넌시 Amazon Linux/Unix 예약 인스턴스를 구입하면 다음과 같이 예약을 나눌 수 있습니다.

- i3.16xlarge의 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.
- i3.8xlarge의 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.
- i3.4xlarge의 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.

수정 요청 제출

예약 인스턴스를 수정하기 전에 해당하는 제한 사항을 읽어야 합니다. 인스턴스 크기를 수정하기 전에 수정하려는 원래 예약의 총 인스턴스 크기 범위를 계산하고 새 구성의 총 인스턴스 크기 범위와 일치하는지 확인하세요.

Console

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.

   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 수정합니다.

   - 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. [Continue]를 선택합니다.
5. 대상 구성 지정을 마칠 때 수정 선택 사항을 확정하려면 [수정 사항 제출(Submit modifications)]을 선택합니다.
6. 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.

AWS CLI

예약 인스턴스를 수정하려면

modify-reserved-instances 명령을 사용합니다. JSON 파일에 구성 세부 정보를 제공할 수 있습니다.

```
aws ec2 modify-reserved-instances \
    --reserved-instances-ids b847fa93-e282-4f55-b59a-1342f5bd7c02 \
    --target-configurations file://configuration.json
```

수정 요청의 상태를 가져오려면

describe-reserved-instances-modifications 명령을 사용합니다. 상태는 `processing`, `fulfilled` 또는 `failed`입니다.

```
aws ec2 describe-reserved-instances-modifications \
    --reserved-instances-modification-ids rimod-d3ed4335-b1d3-4de6-ab31-0f13aaf46687
 \
    --query ReservedInstancesModifications[].Status
```

PowerShell

예약 인스턴스를 수정하려면

Edit-EC2ReservedInstance cmdlet을 사용합니다. `Amazon.EC2.Model.ReservedInstancesConfiguration` 유형의 객체에 구성 세부 정보를 제공할 수 있습니다.

```
Edit-EC2ReservedInstance `
    -ReservedInstancesId b847fa93-e282-4f55-b59a-1342f5bd7c02 `
    -TargetConfiguration $configuration
```

수정 요청의 상태를 가져오려면

Get-EC2ReservedInstancesModification cmdlet을 사용합니다. 상태는 `processing`, `fulfilled` 또는 `failed`입니다.

```
Get-EC2ReservedInstancesModification `
    -ReservedInstancesModificationId rimod-d3ed4335-b1d3-4de6-ab31-0f13aaf46687 | `
    Select Status
```

수정 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 사항을 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 확인하세요.

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, 새 컨버터블 예약형 인스턴스가 교환 중인 원래 컨버터블 예약형 인스턴스보다 가치가 높거나 같아야 합니다.

컨버터블 예약 인스턴스를 교환하면 현재 예약에 대한 인스턴스의 수가 새 컨버터블 예약 인스턴스의 구성 값보다 크거나 같은 여러 인스턴스로 교환됩니다. Amazon EC2는 교환 결과로 받을 수 있는 예약 인스턴스의 수를 계산합니다.

스탠다드 예약 인스턴스는 교환할 수 없지만 수정할 수는 있습니다. 자세한 정보는 예약 인스턴스 수정을 참조하세요.

목차

- 전환형 예약 인스턴스 교환 요건
- 전환형 예약 인스턴스 교환 계산
- 전환형 예약 인스턴스 병합
- 전환형 예약 인스턴스의 일부 교환
- 교환 요청 제출

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
- 만료되기 전까지 24시간 이상이 남아있어야 합니다.

다음 규칙이 적용됩니다.

- 컨버터블 예약 인스턴스는 AWS에서 현재 제공하는 다른 컨버터블 예약 인스턴스로만 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 일부 교환 섹션을 참조하세요. 예약 인스턴스 변경에 대한 자세한 내용은 예약 인스턴스 수정 섹션을 참조하세요.
- 전체 선결제 전환형 예약 인스턴스를 부분 선결제 전환형 예약 인스턴스로 교환할 수 있으며 그 반대로도 교환할 수 있습니다.

> **Note**
>
> 교환에 필요한 총 선불 지급액(트루업 요금)이 0.00 USD 미만인 경우 AWS는 트루업 요금이 0.00 USD 이상이 되도록 컨버터블 예약 인스턴스에 인스턴스 수량을 자동으로 제공합니다.

> **Note**
>
> 새로운 컨버터블 예약 인스턴스의 총 금액(선결제 금액 + 시간당 요금 * 남은 시간 수)이 교환한 컨버터블 예약 인스턴스의 총 금액보다 낮은 경우 AWS는 총 금액이 교환한 컨버터블 예약 인스턴스의 금액과 같거나 그 이상이 되도록 컨버터블 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 컨버터블 예약 인스턴스의 총 금액(시간당 요금 * 남은 시간 수)이 교환한 컨버터블 예약 인스턴스의 총 금액보다 낮은 경우 AWS는 총 금액이 교환한 컨버터블 예약 인스턴스의 금액과 같거나 그 이상이 되도록 컨버터블 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 섹션을 참조하세요.
- Amazon EC2에서는 컨버터블 예약형 인스턴스를 교환할 때 연결된 예약을 사용용 중지하고 종료 날짜를 새 예약에 전송합니다. 교환 후 Amazon EC2에서는 이전 예약의 종료 날짜와 새 예약의 시작 날짜를 모두 교환 날짜와 동일하게 설정합니다. 예를 들어, 기간이 16개월 남은 3년 예약을 교환하는 경우 새 예약은 교환한 컨버터블 예약형 인스턴스의 예약과 종료 날짜가 동일한 16개월 예약입니다.

전환형 예약 인스턴스 교환 계산

전환형 예약 인스턴스 교환은 무료입니다. 하지만 트루업(true-up) 비용을 지불해야 할 수 있습니다. 이 비용은 소유했던 컨버터블 예약형 인스턴스와 교환을 통해 받는 새 컨버터블 예약형 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 Convertible Reserved Instances + 1 additional Convertible Reserved
 Instance
```

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 새 예약이 비례 할당 계산 기준으로 600 USD일 경우 100 USD가 청구됩니다.

```
$600 prorated upfront cost of new reservations - $500 remaining upfront cost of old
 reservations = $100 difference
```

전환형 예약 인스턴스 병합

둘 이상의 컨버터블 예약형 인스턴스를 병합하는 경우 새 컨버터블 예약형 인스턴스의 기간은 기존 컨버터블 예약형 인스턴스와 동일하거나 컨버터블 예약형 인스턴스보다 길어야 합니다. 새 전환형 예약 인스턴스의 만료 날짜는 더 나중에 오는 만료 날짜입니다.

예를 들어 계정에 다음과 같은 전환형 예약 인스턴스가 있다고 가정합시다.

Reserved Instance ID	기간	만료 날짜
aaaa1111	1년	2018-12-31
bbbb2222	1년	2018-07-31
cccc3333	3년	2018-06-30
dddd4444	3년	2019-12-31

- aaaa1111과 bbbb2222를 병합하여 1년 전환형 예약 인스턴스로 변경할 수 있습니다. 3년 전환형 예약 인스턴스로는 변경할 수 없습니다. 새 전환형 예약 인스턴스의 만료 날짜는 2018-12-31입니다.
- bbbb2222과 cccc3333를 병합하여 3년 전환형 예약 인스턴스로 변경할 수 있습니다. 1년 전환형 예약 인스턴스로는 변경할 수 없습니다. 새 전환형 예약 인스턴스의 만료 날짜는 2018-07-31입니다.
- cccc3333과 dddd4444를 병합하여 3년 전환형 예약 인스턴스로 변경할 수 있습니다. 1년 전환형 예약 인스턴스로는 변경할 수 없습니다. 새 전환형 예약 인스턴스의 만료 날짜는 2019-12-31입니다.

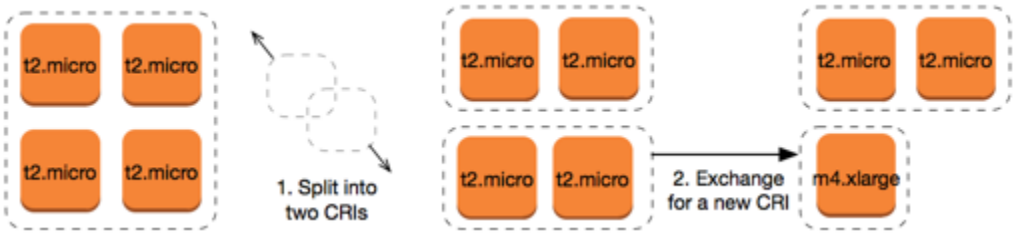
전환형 예약 인스턴스의 일부 교환

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,

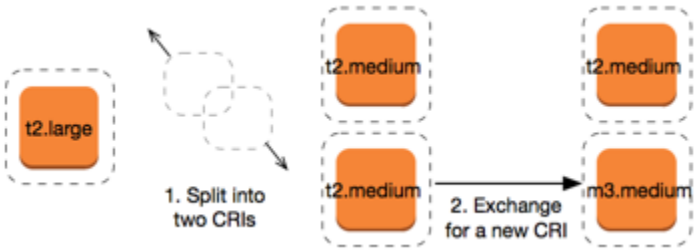
1. `t2.micro` 전환형 예약 인스턴스를 각각 두 인스턴스를 가진 두 `t2.micro` 전환형 예약 인스턴스로 분할하여 수정합니다.
2. 새 `t2.micro` 전환형 예약 인스턴스 중 하나를 `m4.xlarge` 전환형 예약 인스턴스 하나와 교환합니다.



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 및 교환 요청 제출 섹션을 참조하세요.

교환 요청 제출

전환형 예약 인스턴스는 교환할 수 있습니다. 교환되는 예약 인스턴스는 사용 중지됩니다.

Console

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. 원하는 구성의 속성을 선택하고제품 찾기를 선택합니다.
4. 새 전환형 예약 인스턴스을(를) 선택합니다. 화면 하단에서 교환에 따라 받은 예약 인스턴스의 수와 추가 비용을 볼 수 있습니다.
5. 요구 사항에 맞는 전환형 예약 인스턴스을(를) 선택한 경우 검토를 선택합니다.
6. 교환을 선택한 다음닫기를 선택합니다.

AWS CLI

전환형 예약 인스턴스를 교환하려면

1. describe-reserved-instances-offerings 명령을 사용하여 필요에 맞는 새 전환형 예약 인스턴스를 찾습니다.
2. get-reserved-instances-exchange-quote 명령을 사용하여 교환 견적서를 받습니다. 이 견적서에는 교환에서 가져오는 예약 인스턴스의 수 및 교환에 대한 트루업 비용이 포함되어 있습니다.
3. accept-reserved-instances-exchange-quote 명령을 사용하여 교환을 수행합니다.

PowerShell

전환형 예약 인스턴스를 교환하려면

1. Get-EC2ReservedInstancesOffering cmdlet을 사용하여 필요에 맞는 새 전환형 예약 인스턴스를 찾습니다.

2. GetEC2-ReservedInstancesExchangeQuote cmdlet을 사용하여 교환에 대한 견적서를 받습니다. 이 견적서에는 교환에서 가져오는 예약 인스턴스의 수 및 교환에 대한 트루업 비용이 포함되어 있습니다.
3. Approve-EC2ReservedInstancesExchangeQuote cmdlet을 사용하여 교환을 수행합니다.

예약 인스턴스 할당량

매달 새로운 예약 인스턴스를 구매할 수 있습니다. 매달 구매할 수 있는 새 예약 인스턴스의 수는 다음과 같이 월별 할당량에 따라 결정됩니다.

할당량 설명	기본 할당량
새로운 리전별 예약 인스턴스	리전당 월별 20개
새로운 영역별 예약 인스턴스	가용 영역당 월별 20개

예를 들어, 3개의 가용 영역이 있는 리전에서 기본 할당량은 월별 80개의 새로운 예약 인스턴스이며 다음과 같이 계산됩니다.

- 해당 리전의 리전별 예약 인스턴스 20개
- 영역별 추가 예약 인스턴스 60개(3개의 가용 영역 각각에 대해 20개)

`running` 상태의 인스턴스는 할당량에 포함됩니다. `pending`, `stopping`, `stopped`, `hibernated` 상태의 인스턴스는 할당량에 포함되지 않습니다.

구매한 예약 인스턴스 수 보기

구매하는 예약 인스턴스의 수는 Instance count(인스턴스 수) 필드(콘솔) 또는 `InstanceCount` 파라미터(AWS CLI)로 표시됩니다. 새 예약 인스턴스를 구매할 때 할당량은 총 인스턴스 수를 기준으로 측정됩니다. 예를 들어, 인스턴스 수가 10개인 단일 예약 인스턴스 구성을 구매하는 경우 할당량 중 1개가 아닌 10개를 사용한 것으로 계산됩니다.

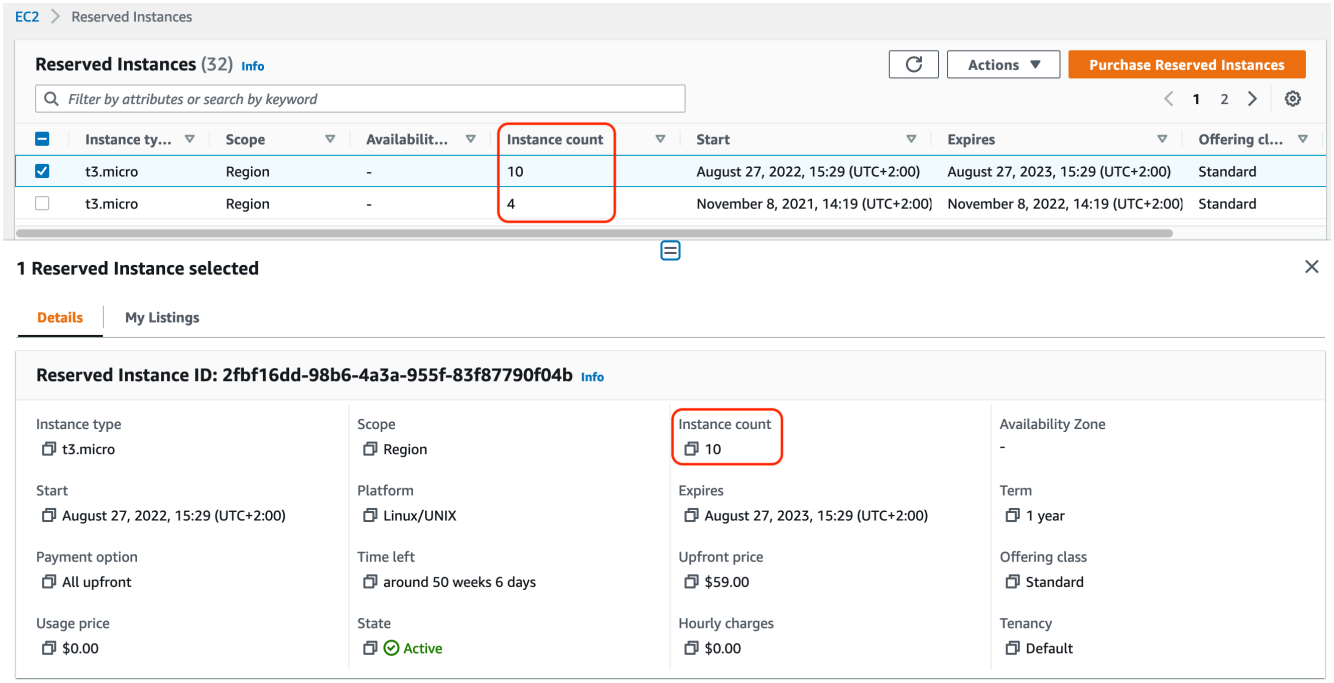
Amazon EC2 또는 AWS CLI를 사용하여 구매한 예약 인스턴스 수를 확인할 수 있습니다.

Console

구매한 예약 인스턴스 수 보기

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 예약 인스턴스를 선택합니다.
3. 테이블에서 예약 인스턴스 구성을 선택하고 인스턴스 개수(인스턴스 수) 필드를 확인합니다.

   다음 스크린샷에서 선택한 줄은 `t3.micro` 인스턴스 유형에 대한 단일 예약 인스턴스 구성을 나타냅니다. 테이블 보기의 인스턴스 수 열과 세부 정보 보기의 인스턴스 수 개수 필드(스크린샷에 나와 있음)는 이 구성에 대해 10개의 예약 인스턴스가 있음을 나타냅니다.



AWS CLI

구매한 예약 인스턴스 수 보기

describe-reserved-instances 명령을 사용하고 예약 인스턴스 구성의 ID를 지정합니다.

```
aws ec2 describe-reserved-instances \
    --reserved-instances-ids a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --output table
```

출력의 예시는 다음과 같습니다. `InstanceCount` 필드는 이 구성에 대해 10개의 예약 인스턴스가 있음을 나타냅니다.

```
-------------------------------------------------------------------
|                    DescribeReservedInstances                    |
+-----------------------------------------------------------------+
||                       ReservedInstances                       ||
|+----------------------+----------------------------------------+|
||  CurrencyCode        |  USD                                   ||
||  Duration            |  31536000                              ||
||  End                 |  2023-08-27T13:29:44+00:00             ||
||  FixedPrice          |  59.0                                  ||
||  InstanceCount       |  10                                    ||
||  InstanceTenancy     |  default                               ||
||  InstanceType        |  t3.micro                              ||
||  OfferingClass       |  standard                              ||
||  OfferingType        |  All Upfront                           ||
||  ProductDescription  |  Linux/UNIX                            ||
||  ReservedInstancesId |  a1b2c3d4-5678-90ab-cdef-EXAMPLE11111  ||
||  Scope               |  Region                                ||
||  Start               |  2022-08-27T13:29:45.938000+00:00      ||
||  State               |  active                                ||
||  UsagePrice          |  0.0                                   ||
|+----------------------+----------------------------------------+|
|||                      RecurringCharges                       |||
||+---------------------------------+--------------------------+||
|||  Amount                         |  0.0                     |||
|||  Frequency                      |  Hourly                  |||
||+---------------------------------+--------------------------+||
```

PowerShell

구매한 예약 인스턴스 수 보기

Get-EC2ReservedInstance Cmdlet을 사용하고 예약 인스턴스 구성의 ID를 지정합니다.

```
Get-EC2ReservedInstance -ReservedInstancesId a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력의 예시는 다음과 같습니다. InstanceCount 필드는 이 구성에 대해 10개의 예약 인스턴스가 있음을 나타냅니다.

```
AvailabilityZone    :
CurrencyCode        : USD
Duration            : 31536000
End                 : 1/12/2017 8:57:08 PM
```

```
FixedPrice          : 0
InstanceCount       : 10
InstanceTenancy     : default
InstanceType        : t3.medium
OfferingClass       : standard
OfferingType        : All Upfront
ProductDescription  : Windows
RecurringCharges    : {}
ReservedInstancesId : a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
Scope               : Region
Start               : 10/12/2016 4:00:00 PM
State               : active
Tags                : {}
UsagePrice          : 0
```

고려 사항

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.

예약 인스턴스 할당량 확인 및 할당량 증가 요청

Amazon EC2 콘솔은 할당량 정보를 제공합니다. 할당량 증가를 요청할 수도 있습니다. 자세한 내용은 현재 할당량 보기 및 증가 요청 섹션을 참조하세요.

Spot Instances

스팟 인스턴스는 온디맨드 가격보다 저렴한 비용으로 제공되는 예비 EC2 용량을 사용하는 인스턴스입니다. 스팟 인스턴스는 큰 할인율로 미사용 EC2 인스턴스를 요청할 수 있게 해주므로 사용자는 Amazon EC2 비용을 대폭 낮출 수 있습니다. 스팟 인스턴스는의 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. 용량을 사용할 수 있을 때마다 스팟 인스턴스가 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 섹션을 참조하세요.

EC2 인스턴스의 다른 구매 옵션을 비교하려면 Amazon EC2 결제 및 구매 옵션 섹션을 참조하세요.

개념

스팟 인스턴스를 시작하기 전에 다음 개념을 익혀야 합니다.

- 스팟 용량 풀 - 인스턴스 유형(예: `m5.large`)과 가용 영역이 동일한 미사용 EC2 인스턴스의 집합입니다.
- 스팟 가격 - 스팟 인스턴스의 시간당 현재 가격입니다.
- 스팟 인스턴스 요청 - 스팟 인스턴스를 요청합니다. 용량이 가용 상태가 되면 Amazon EC2는 사용자의 요청을 이행합니다. 스팟 인스턴스 요청은 일회성 또는 영구적입니다. Amazon EC2에서는 요청에 연결된 스팟 인스턴스가 중단된 후 스팟 인스턴스 요청을 자동으로 다시 제출합니다.
- EC2 인스턴스 재조정 권장 사항 – Amazon EC2는 인스턴스 재조정 권장 사항 신호를 보내 스팟 인스턴스가 중단될 위험이 높다는 것을 알려줍니다. 이 신호는 2분 스팟 인스턴스 중단 알림을 기다릴 필요 없이 기존 또는 신규 스팟 인스턴스에서 워크로드를 사전에 재조정할 수 있는 기회를 제공합니다.
- 스팟 인스턴스 중단 - Amazon EC2에 다시 용량이 필요한 경우 Amazon EC2는 스팟 인스턴스를 종료 또는 중지하거나 최대 절전 모드로 전환합니다. Amazon EC2는 스팟 인스턴스 중단 2분 전에 경고하는 스팟 인스턴스 중단 공지를 제공합니다.

스팟 인스턴스와 온디맨드 인스턴스의 차이점

다음 표에는 스팟 인스턴스와 온디맨드 인스턴스의 주요 차이점이 나열되어 있습니다.

	Spot Instances	On-Demand Instances
시작 시간	스팟 인스턴스 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 인스턴스 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 장기적인 수요와 공급에 따라 다릅니다.	온디맨드 인스턴스의 시간당 가격은 고정된 가격입니다.
리밸런싱 권고	인스턴스 중단 위험이 높아질 때 실행 중인 스팟 인스턴스에 대해 Amazon EC2가 생성하는 신호입니다.	온디맨드 인스턴스가 중단(중지, 최대 절전 또는 종료)되는 시간을 결정합니다.
인스턴스 중단	Amazon EBS 지원 스팟 인스턴스를 중지하고 시작할 수 있습니다. 또한 Amazon EC2에서 용량을 더 이상 사용할 수 없는 경우 개별 스팟 인스턴스를 중단할 수 있습니다.	온디맨드 인스턴스가 중단(중지, 최대 절전 또는 종료)되는 시간을 결정합니다.

요금 및 비용 절감

스팟 인스턴스에 대해 스팟 가격을 지불합니다. 이 가격은 Amazon EC2에서 설정되며 스팟 인스턴스의 장기적 수요 및 공급에 따라 점진적으로 조정됩니다. 스팟 인스턴스는 사용자가 종료하거나, 용량을 더 이상 사용할 수 없게 되거나, Amazon EC2 Auto Scaling 그룹이 스케일 인 중에 종료할 때까지 실행됩니다.

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 섹션을 참조하세요.

스팟 인스턴스는 절감형 플랜에 포함되지 않습니다. 절감형 플랜을 사용하는 경우 스팟 인스턴스를 사용하여 이미 절감한 비용 외에 추가 비용 절감을 제공하지 않습니다. 또한 스팟 인스턴스에 대한 지출에는 컴퓨팅 절감형 플랜의 약정이 적용되지 않습니다.

가격 보기

AWS 리전 및 인스턴스 유형당 현재(5분마다 업데이트됨) 최저 스팟 가격을 보려면 Amazon EC2 스팟 인스턴스 요금 페이지를 참조하세요.

지난 3개월 동안의 스팟 가격 기록을 보려면 Amazon EC2 콘솔 또는 describe-spot-price-history를 사용합니다. 자세한 내용은 스팟 인스턴스 요금 기록 보기 섹션을 참조하세요.

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 플릿 또는 모든 스팟 인스턴스의 스팟 인스턴스를 사용하여 얻은 절감액을 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 섹션을 참조하세요.

결제 보기

청구서에서 서비스 사용에 대한 세부 정보를 확인할 수 있습니다. 자세한 내용은 AWS Billing 사용 설명서에서 결제 보기를 참조하세요.

Amazon EC2 스팟 모범 사례

Amazon EC2는 AWS 클라우드의 예비 EC2 컴퓨팅 용량을 온디맨드 요금에 비해 최대 90% 할인된 가격으로 사용할 수 있도록 제공합니다. 온디맨드 인스턴스와 스팟 인스턴스 간의 유일한 차이점은 Amazon EC2에서 용량을 다시 회수해야 할 때 Amazon EC2가 2분 전 알림을 통해 스팟 인스턴스를 중단할 수 있다는 것입니다. 스팟 인스턴스를 최상의 상태로 사용하려면 사용 모범 사례를 이해하고 적용하는 것이 중요합니다.

스팟 인스턴스는 유연한 상태 비저장, 내결함성 애플리케이션에 권장됩니다. 예를 들어 스팟 인스턴스는 빅 데이터, 컨테이너화된 워크로드, CI/CD, 상태 비저장 웹 서버, 고성능 컴퓨팅(HPC), 렌더링 워크로드에 적합합니다.

스팟 인스턴스는 실행 중인 동안에는 온디맨드 인스턴스와 정확히 동일합니다. 그러나 스팟은 실행 중인 인스턴스를 워크로드를 완료할 수 있을 만큼 충분히 오래 유지할 수 있다고 보장하지 않습니다. 또

한 스팟은 찾고 있는 인스턴스의 즉각적인 가용성을 보장하거나 요청한 총 용량을 항상 확보할 수 있다고 보장하지 않습니다. 또한 스팟 인스턴스 가용성은 수요와 공급에 따라 달라지기 때문에 스팟 인스턴스 중단 및 용량은 시간이 지남에 따라 변할 수 있으며 과거의 성능이 미래의 결과를 보장하지 않습니다.

스팟 인스턴스는 유연성이 없거나 상태 저장이거나 내결함성이 없거나 인스턴스 노드 간에 밀접하게 연결된 워크로드에 적합하지 않습니다. 수시로 전체 목표 용량을 완전히 사용할 수 없는 경우를 허용하지 않는 워크로드에는 스팟 인스턴스를 권장하지 않습니다. 스팟 모범 사례를 따라 인스턴스 유형과 가용 영역을 유연하게 조정하면 고가용성을 확보할 수 있는 최상의 기회를 제공하지만, 온디맨드 인스턴스에 대한 수요가 급증하면 스팟 인스턴스의 워크로드가 중단될 수 있으므로 용량을 사용할 수 있다고 보장할 수는 없습니다.

이러한 워크로드에 스팟 인스턴스를 사용하거나 중단이나 가용성 손실을 처리하기 위해 온디맨드 인스턴스로 장애 조치를 시도하지 마세요. 온디맨드 인스턴스로 장애 조치하면 실수로 다른 스팟 인스턴스가 중단될 수도 있습니다. 또한 인스턴스 유형과 가용 영역 조합의 스팟 인스턴스가 중단되면 동일한 조합으로 온디맨드 인스턴스를 구하기가 어려워질 수도 있습니다.

숙련된 스팟 사용자인지 또는 스팟 인스턴스를 처음 사용하는지 관계없이 현재 스팟 인스턴스 중단 또는 가용성에 문제가 발생하는 경우 다음 모범 사례를 따라 스팟 서비스를 사용하는 최상의 환경을 제공하는 것이 좋습니다.

스팟 모범 사례

- 개별 인스턴스에서 중단 대비
- 인스턴스 유형 및 가용 영역에 대한 유연성 유지
- 속성 기반 인스턴스 유형 선택 사용
- 스팟 배치 점수를 사용하여 최적의 리전 및 가용 영역 식별
- EC2 Auto Scaling 또는 EC2 플릿을 사용하여 총 용량 관리
- 가격 및 용량 최적화 할당 전략 사용
- 통합 AWS 서비스를 사용하여 스팟 인스턴스 관리
- 어느 스팟 요청 방법을 사용하는 것이 최선인가요?

개별 인스턴스에서 중단 대비

스팟 인스턴스 중단을 정상적으로 처리하는 가장 좋은 방법은 내결함성이 있도록 애플리케이션을 설계하는 것입니다. EC2 인스턴스 리밸런싱 권고 및 스팟 인스턴스 중단 공지를 활용하여 이를 달성할 수 있습니다.

EC2 인스턴스 리밸런싱 권고는 스팟 인스턴스의 중단 위험이 높아질 때 알림을 보내는 신호입니다. 이 신호는 스팟 인스턴스 중단 2분 전 공지에 앞서 스팟 인스턴스를 사전에 관리할 수 있는 기회를 제공합니다. 중단 위험이 높아지지 않는 신규 또는 기존 스팟 인스턴스에 대한 워크로드를 리밸런싱하도록 결정할 수 있습니다. Auto Scaling 및 EC2 플릿의 용량 리밸런싱 특성을 사용하여 이 신호를 손쉽게 사용할 수 있습니다.

스팟 인스턴스 중단 공지는 Amazon EC2에서 스팟 인스턴스를 중단하기 2분 전에 생성되는 경고입니다. 워크로드가 '시간에 유연'한 경우 중단 시 스팟 인스턴스를 종료하는 대신 중지하거나 최대 절전 모드로 전환하도록 구성할 수 있습니다. Amazon EC2는 스팟 인스턴스 중단 시 자동으로 중지하거나 최대 절전 모드로 전환하며 가용 용량이 있을 때 인스턴스를 자동으로 다시 시작합니다.

Amazon EventBridge에서 리밸런싱 권고 및 중단 알림을 캡처하는 규칙을 생성한 다음 워크로드 진행에 대한 검사점을 트리거하거나 중단을 정상적으로 처리하는 것이 좋습니다. 자세한 내용은 리밸런싱 권고 신호 모니터링 섹션을 참조하세요. 이벤트 규칙을 생성하고 사용하는 방법을 안내하는 자세한 예제는 Amazon EC2 스팟 인스턴스 중단 공지 활용을 참조하세요.

자세한 내용은 EC2 인스턴스 리밸런싱 권고 및 스팟 인스턴스 중단 섹션을 참조하세요.

인스턴스 유형 및 가용 영역에 대한 유연성 유지

스팟 용량 풀은 인스턴스 유형(예: `m5.large`)과 가용 영역(예: us-east-1a)이 동일한 미사용 EC2 인스턴스의 집합입니다. 요청하는 인스턴스 유형과 워크로드를 배포할 수 있는 가용 영역에 대한 유연성이 있어야 합니다. 그러면 스팟에서 필요한 컴퓨팅 용량을 찾고 할당할 가능성이 높아집니다. 예를 들어 c4, m5 및 m4 제품군의 large를 사용해도 무방하면 `c5.large`를 요청하지 마세요.

특정 요구 사항에 따라 컴퓨팅 요구 사항을 충족하기 위해 유연하게 선택할 수 있는 인스턴스 유형을 평가할 수 있습니다. 워크로드를 수직으로 확장할 수 있는 경우 요청에 더 큰 인스턴스 유형(vCPU 및 메모리 추가)을 포함해야 합니다. 수평으로만 확장할 수 있는 경우 이전 세대 인스턴스 유형을 포함해야 합니다. 이러한 인스턴스는 온디맨드 고객의 수요가 적기 때문입니다.

일반적으로 각 워크로드에 대해 최소 10개의 인스턴스 유형을 유연하게 선택할 수 있어야 합니다. 또한 모든 가용 영역이 사용자의 VPC에서 사용하도록 구성되고 워크로드에 맞게 선택되어야 합니다.

속성 기반 인스턴스 유형 선택 사용

속성 기반 인스턴스 유형 선택을 통해 실행하려는 워크로드의 인스턴스 속성(예: vCPU, 메모리, 스토리지)을 지정할 수 있습니다. 그러면 EC2 Auto Scaling 또는 EC2 플릿이 지정된 속성과 일치하는 인스턴스를 자동으로 식별하여 시작합니다. 이렇게 하면 특정 인스턴스 유형을 수동으로 선택하는 데 드는 수고를 덜 수 있지만, 이를 위해서는 각 인스턴스 유형을 심층적으로 이해해야 합니다.

더욱이 속성 기반 인스턴스 유형 선택을 통해 새로 릴리스된 인스턴스 유형이 제공되면 자동으로 사용할 수 있습니다. 이를 통해 점점 더 다양해지는 스팟 인스턴스 용량에 원활하게 액세스할 수 있습니다.

속성 기반 인스턴스 유형 선택은 고성능 컴퓨팅(HPC) 및 빅 데이터 워크로드 등 실행되는 인스턴스 유형에 대해 유연한 워크로드 및 프레임워크에 이상적입니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 속성 기반 인스턴스 유형 선택을 사용하여 혼합 인스턴스 그룹 생성과 이 설명서에서 EC2 플릿 또는 스팟 플릿의 인스턴스 유형 선택에 대한 속성 지정의 내용을 참조하세요.

스팟 배치 점수를 사용하여 최적의 리전 및 가용 영역 식별

스팟 인스턴스는 미사용 EC2 용량이고, 이 용량은 EC2 공급 및 수요에 따라 변동됩니다. 따라서 특정 시간의 특정 위치에 필요한 정확한 스팟 용량을 항상 확보하지 못할 수도 있습니다. 이러한 예측 불가능성을 완화하기 위해 스팟 배치 점수 기능을 사용할 수 있습니다. 이 기능은 먼저 해당 위치에서 스팟 인스턴스를 시작할 필요 없이 필요한 스팟 용량을 충족하기에 충분한 용량을 보유할 가능성이 높은 리전 또는 가용 영역에 대한 권장 사항을 제공합니다.

스팟 배치 점수는 인스턴스 유형과 해당 인스턴스가 사용할 수 있는 리전 또는 가용 영역을 유연하게 조정할 수 있는 워크로드에 가장 적합합니다. 필요한 스팟 용량, 인스턴스 유형 요구 사항, 리전 또는 가용 영역에 대한 권장 사항 등을 지정하기만 하면 됩니다. 그에 따라 각 리전 또는 가용 영역에 대해 1~10점 범위의 점수를 받게 되는데, 이는 해당 위치에서 요청된 스팟 용량을 성공적으로 프로비저닝할 가능성을 나타냅니다. 10점은 스팟 요청이 성공할 가능성이 높음을 나타냅니다.

용량이 시간에 따라 달라질 수 있으므로 스팟 배치 점수는 특정 시점의 권장 사항이라는 점에 유의해야 합니다. 사용 가능한 용량을 보장하거나 중단 위험을 예측하지는 못합니다.

Amazon EC2 콘솔, AWS CLI, 또는 SDK에서 스팟 배치 점수 기능을 사용할 수 있습니다. 자세한 내용은 스팟 배치 점수 섹션을 참조하세요.

EC2 Auto Scaling 또는 EC2 플릿을 사용하여 총 용량 관리

스팟을 사용할 때는 개별 인스턴스가 아니라 총 용량(vCPU, 메모리, 스토리지 또는 네트워크 처리량이 포함된 단위)의 측면에서 생각할 수 있습니다. Auto Scaling과 EC2 플릿을 사용하면 목표 용량을 시작하고 유지 관리할 수 있으며 중단되거나 수동으로 종료된 모든 항목을 대체할 리소스를 자동으로 요청할 수 있습니다. Auto Scaling 또는 EC2 플릿을 구성할 때는 애플리케이션 요구 사항에 따라 인스턴스 유형과 목표 용량만 지정하면 됩니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Auto Scaling 그룹과 이 사용 설명서의 EC2 집합 생성 섹션을 참조하세요.

가격 및 용량 최적화 할당 전략 사용

Auto Scaling 그룹의 할당 전략은 예비 용량이 있는 스팟 용량 풀을 수동으로 찾을 필요 없이 목표 용량을 프로비저닝하는 데 도움이 됩니다. `price-capacity-optimized` 전략은 가장 사용 가능하고 가격도 가장 낮을 수 있는 스팟 용량 풀에서 인스턴스를 자동으로 프로비저닝하므로 이 전략을 사용하는 것이 좋습니다. EC2 플릿에서 `price-capacity-optimized` 할당 전략을 활용할 수도 있습니다. 스팟 인스턴스 용량이 최적의 용량을 가진 풀에서 소싱되기 때문에 스팟 인스턴스가 회수될 가능성이 줄어듭니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 여러 인스턴스 유형에 대한 할당 전략과 이 사용 설명서의 워크로드의 중단 비용이 높은 경우 섹션을 참조하세요.

통합 AWS 서비스를 사용하여 스팟 인스턴스 관리

다른 AWS 서비스가 스팟과 통합되어 개별 인스턴스 또는 플릿을 관리할 필요 없이 전체 컴퓨팅 비용을 절감할 수 있습니다. 해당 워크로드에 대해 Amazon EMR, Amazon Elastic Container Service, AWS Batch, Amazon Elastic Kubernetes Service, Amazon SageMaker AI, AWS Elastic Beanstalk 및 Amazon GameLift Servers와 같은 솔루션을 고려하는 것이 좋습니다. 이러한 서비스의 스팟 모범 사례에 대한 자세한 내용은 Amazon EC2 스팟 인스턴스 워크샵 웹 사이트를 참조하세요.

어느 스팟 요청 방법을 사용하는 것이 최선인가요?

다음 표를 사용하여 스팟 인스턴스를 요청할 때 어느 API를 사용할지 결정합니다.

API	언제 사용해야 하나요?	사용 사례	이 API를 사용해야 하나요?
CreateAutoScalingGroup	• 단일 구성 또는 혼합 구성을 사용하는 여러 인스턴스가 필요합니다. • 구성 API를 통해 수명 주기 관리를 자동화하려고 합니다.	원하는 인스턴스 수를 유지하면서 인스턴스의 수명 주기를 관리하는 Auto Scaling 그룹을 생성합니다. 지정된 최소 및 최대 한도 사이의 수평적 크기 조정(인스턴스 추가)을 지원합니다.	예
CreateFleet	• 단일 구성 또는 혼합 구성을 사용하는	인스턴스 유형, AMI, 가용 영역 또는 서브넷에 따라 바뀌는 여	예 - `instant` 모드에서(Auto Scaling이 필요하지 않은 경우)

API	언제 사용해야 하나요?	사용 사례	이 API를 사용해야 하나요?
	여러 인스턴스가 필요합니다. • 인스턴스 수명 주기를 자체 관리하려고 합니다. • Auto Scaling이 필요하지 않은 경우 `instant` 유형 플릿을 사용하는 것을 권장합니다.	러 출시 사양을 사용하여 단일 요청으로 온디맨드 인스턴스와 스팟 인스턴스의 플릿을 생성합니다. 스팟 인스턴스 할당 전략의 기본값은 단위당 `lowest-price` 이지만, `price-capacity-optimized` , `capacity-optimized` 또는 `diversified` 로 변경할 수 있습니다.	
RunInstances	• 이미 RunInstances API를 사용하여 온디맨드 인스턴스를 시작하고 있으며 단일 파라미터를 변경하여 스팟 인스턴스 시작으로 변경하려고 합니다. • 인스턴스 유형이 다른 여러 인스턴스는 필요하지 않습니다.	AMI와 하나의 인스턴스 유형을 사용하여 지정된 수의 인스턴스를 시작합니다.	아니요 - RunInstances가 단일 요청에서 혼합 인스턴스 유형을 허용하지 않기 때문

API	언제 사용해야 하나요?	사용 사례	이 API를 사용해야 하나요?
RequestSpotFleet	• RequestSpotFleet API는 계획된 투자가 없는 레거시 API이므로 사용하지 않는 것이 좋습니다. • 인스턴스 수명 주기를 관리하려면 CreateFleet API를 사용하세요. • 인스턴스 수명 주기를 관리하지 않으려면 CreateAutoScalingGroup API를 사용하세요.	사용하지 않습니다. RequestSpotFleet은 계획된 투자가 없는 레거시 API입니다.	아니요
RequestSpotInstances	• RequestSpotInstances API는 계획된 투자가 없는 레거시 API이므로 사용하지 않는 것이 좋습니다.	사용하지 않습니다. RequestSpotInstances는 계획된 투자가 없는 레거시 API입니다.	아니요

스팟 인스턴스의 작동 방식

스팟 인스턴스를 시작하려면 스팟 인스턴스 요청을 직접 생성하거나 Amazon EC2를 통해 자동으로 스팟 인스턴스 요청을 생성합니다. 스팟 인스턴스 요청이 이행되면 스팟 인스턴스가 시작됩니다.

여러 다양한 서비스를 사용하여 스팟 인스턴스를 시작할 수 있습니다. 자세한 내용은 Amazon EC2 스팟 인스턴스 시작하기를 참조하세요. 이 사용 설명서에서는 EC2를 사용하여 스팟 인스턴스를 시작하는 다음 방법에 대해 설명합니다.

- Amazon EC2 콘솔의 인스턴스 시작 마법사를 사용하거나 run-instances 명령을 사용하여 스팟 인스턴스 요청을 생성할 수 있습니다. 자세한 내용은 스팟 인스턴스 관리 섹션을 참조하세요.
- EC2 플릿을 생성하고 원하는 스팟 인스턴스 수를 지정할 수 있습니다. Amazon EC2는 EC2 플릿에 지정된 모든 스팟 인스턴스에 대해 사용자를 대신하여 스팟 인스턴스 요청을 생성합니다. 자세한 내용은 EC2 집합 생성 섹션을 참조하세요.
- 스팟 플릿을 요청을 생성하고 원하는 스팟 인스턴스 수를 지정할 수 있습니다. Amazon EC2는 스팟 플릿 요청에 지정된 모든 스팟 인스턴스에 대해 사용자를 대신하여 스팟 인스턴스 요청을 생성합니다. 자세한 내용은 스팟 플릿 생성 섹션을 참조하세요.

사용 가능한 용량이 있는 경우 스팟 인스턴스가 시작됩니다. 스팟 인스턴스는 사용자가 중지 또는 종료하거나 Amazon EC2에 의해 중단(스팟 인스턴스 중단이라고 함)될 때까지 실행됩니다. Amazon EC2는 인스턴스를 중단할 때 스팟 인스턴스를 중지, 종료 또는 최대 절전 모드로 전환할 수 있습니다.

스팟 인스턴스를 사용할 때는 중단에 대비해야 합니다. Amazon EC2는 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 보내 중단 2분 전에 이를 인스턴스에 경고합니다. 스팟 인스턴스에 대한 종료 방지 기능은 활성화할 수 없습니다. 자세한 내용은 스팟 인스턴스 중단 섹션을 참조하세요.

내용

- 스팟 인스턴스 요청 상태
- 시작 그룹에서 스팟 인스턴스 시작
- 가용 영역 그룹에서 스팟 인스턴스 시작
- VPC에서 스팟 인스턴스 시작
- 성능 버스트 가능 인스턴스 시작
- 단일 테넌트 하드웨어에서 시작

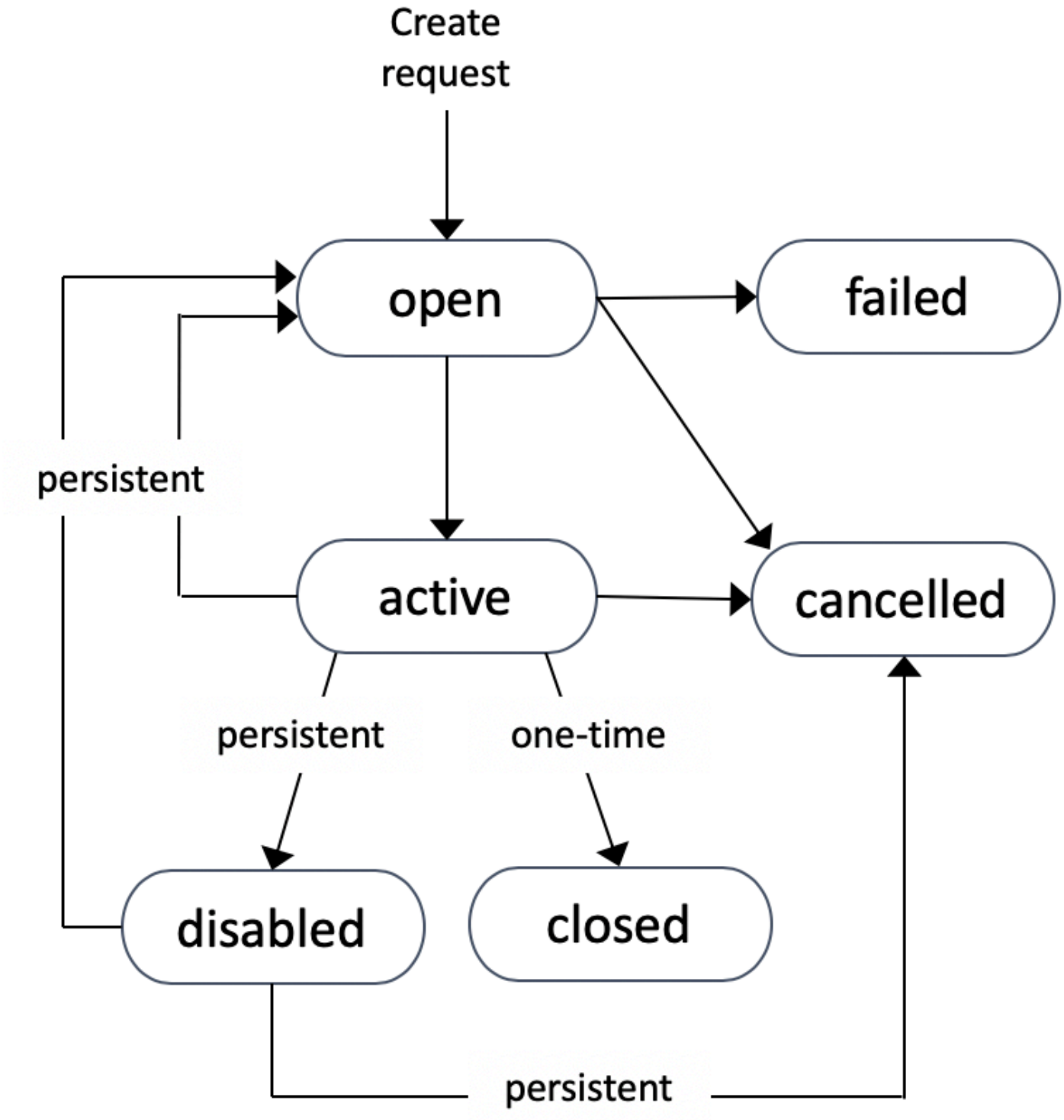
스팟 인스턴스 요청 상태

스팟 인스턴스 요청 상태는 다음 중 하나일 수 있습니다.

- open - 요청이 이행될 때까지 대기 중입니다.

- `active` - 요청이 이행되었으며 요청에 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. 용량을 사용할 수 없는 경우 스팟 인스턴스가 종료되고 스팟 인스턴스 요청이 종료됩니다.

영구적 스팟 인스턴스 요청은 요청이 이행되더라도 요청이 만료되거나 사용자가 요청을 취소할 때까지 활성 상태로 유지됩니다. 용량을 사용할 수 없는 경우 스팟 인스턴스가 중단됩니다. 인스턴스가 중단된 후 용량을 다시 사용할 수 있게 되면 스팟 인스턴스가 중지된 경우 시작되고 최대 절전 모드인 경우 재개됩니다. 용량을 사용할 수 있는 경우 스팟 인스턴스를 중지하고 다시 시작할 수 있습니다. 스팟 인스턴스가 종료되는 경우(스팟 인스턴스가 중지 또는 실행 중 상태인지 여부와 상관없이) 스팟 인스턴스 요청이 다시 열리고 Amazon EC2가 새 스팟 인스턴스를 시작합니다. 자세한 내용은 스팟 인스턴스 중지, 스팟 인스턴스 시작, 스팟 인스턴스 종료 섹션을 참조하세요.

이 상태를 통해 스팟 인스턴스 요청의 상태뿐 아니라 시작된 스팟 인스턴스의 상태도 추적할 수 있습니다. 자세한 내용은 스팟 인스턴스 요청 상태 가져오기 섹션을 참조하세요.

시작 그룹에서 스팟 인스턴스 시작

스팟 인스턴스 요청에서 시작 그룹을 지정하여 해당 인스턴스를 모두 시작할 수 있는 경우에만 스팟 인스턴스 세트를 시작하도록 Amazon EC2에 알립니다. 또한 스팟 서비스가 시작 그룹에 있는 인스턴스 중 하나를 종료해야 하는 경우 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 영역에서 스팟 인스턴스 세트를 시작하도록 Amazon EC2에 알립니다. Amazon EC2는 가용 영역 그룹의 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.

- [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 인스턴스의 서브넷을 지정해야 합니다.

성능 버스트 가능 인스턴스 시작

T 인스턴스 유형은 버스트 가능 성능 인스턴스입니다. 버스트 가능 성능 인스턴스 유형을 사용하여 스팟 인스턴스를 시작하고 CPU 크레딧 발생에 대한 유휴 시간 없이 즉시 짧은 기간 동안 버스트 가능 성능 스팟 인스턴스를 사용할 계획인 경우 표준 모드로 시작하여 높은 비용 지불을 방지하는 것이 좋습니다. 버스팅 가능 성능 스팟 인스턴스를 무제한 모드로 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.

무제한 모드는 버스팅에 대한 CPU 크레딧이 발생할 정도로 인스턴스 실행이 긴 경우에만 버스팅 가능 성능 스팟 인스턴스에 적합합니다. 그렇게 하지 않으면 잉여 크레딧 비용을 지불하면 버스트 가능 성능 스팟 인스턴스가 다른 인스턴스를 사용하는 것보다 비용이 많이 듭니다. 자세한 내용은 무제한 모드 대 고정 CPU 사용 시기 섹션을 참조하세요.

T2 인스턴스(표준 모드로 구성된 경우)는 시작 크레딧을 받습니다. T2 인스턴스는 시작 크레딧을 받는 유일한 버스트 가능 성능 인스턴스입니다. 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 스팟 인스턴스에 무제한 모드를 사용하거나 전용 CPU를 포함한 인스턴스 유형을 사용할 수 있습니다.

단일 테넌트 하드웨어에서 시작

스팟 인스턴스를 단일 테넌트 하드웨어에서 실행할 수 있습니다. 전용 스팟 인스턴스는 다른 AWS 계정에 속하는 인스턴스로부터 물리적으로 격리됩니다. 자세한 내용은 Amazon EC2 전용 인스턴스 및 Amazon EC2 전용 인스턴스를 참조하세요.

전용 스팟 인스턴스를 실행하려면 다음 중 하나를 수행합니다.

- 스팟 인스턴스 요청을 생성할 때 테넌시를 `dedicated`로 지정합니다. 자세한 내용은 스팟 인스턴스 관리 섹션을 참조하세요.
- VPC에서 인스턴스 테넌시 `dedicated`를 사용하여 스팟 인스턴스를 요청합니다. 자세한 내용은 기본 테넌시를 사용하는 VPC로 전용 인스턴스 시작 섹션을 참조하세요. VPC에서 인스턴스 테넌시 `dedicated`로 스팟 인스턴스를 요청한 경우 테넌시 `default`로 스팟 인스턴스를 요청할 수 없습니다.

T 인스턴스를 제외한 모든 인스턴스 패밀리가 전용 스팟 인스턴스를 지원합니다. 지원되는 각 인스턴스 패밀리에서 가장 큰 인스턴스 크기 또는 메탈 크기만이 전용 스팟 인스턴스를 지원합니다.

스팟 인스턴스 요금 기록 보기

스팟 인스턴스 가격은 Amazon EC2에서 정하고, 스팟 인스턴스 용량의 장기적인 공급 수요 추세에 따라 점진적으로 조정됩니다.

스팟 요청이 이행되면 스팟 인스턴스는 온디맨드 가격을 초과하지 않는 현재 스팟 가격으로 시작됩니다. 인스턴스 유형, 운영 체제 및 가용 영역을 기준으로 필터링하여 지난 90일 동안의 스팟 가격 기록을 볼 수 있습니다.

현재 스팟 인스턴스 가격은 Amazon EC2 스팟 인스턴스 요금을 참조하세요.

Console

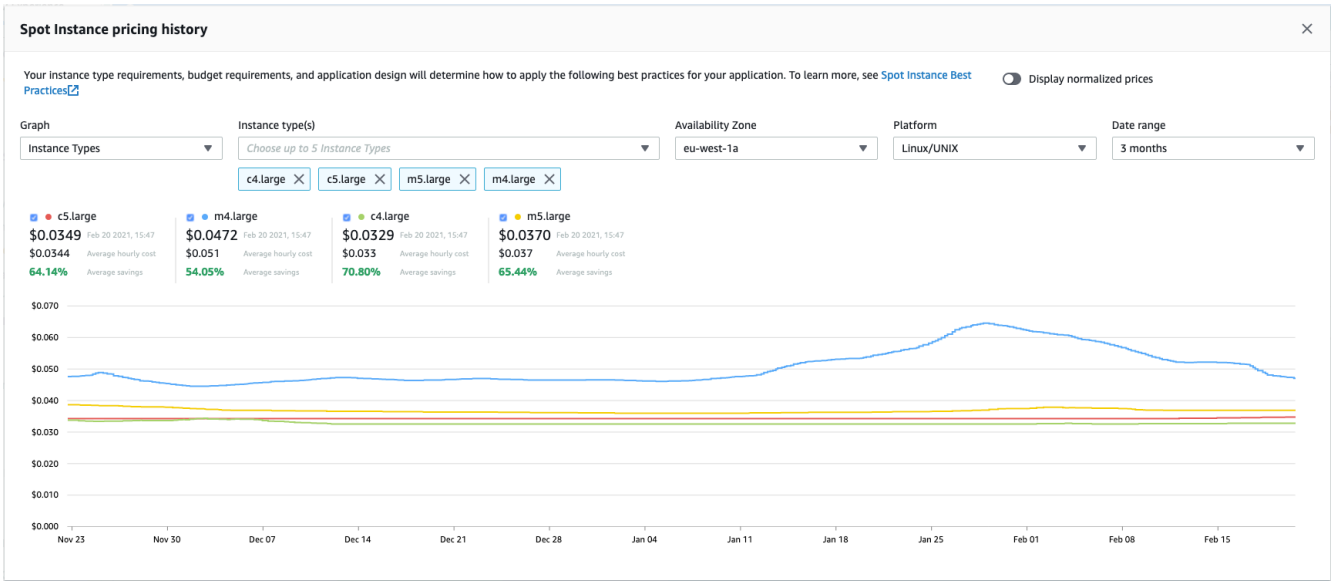
스팟 가격 기록을 보는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. 요금 내역을 선택합니다.

4. 그래프에서 가용 영역 또는 인스턴스 유형별로 요금 내역을 비교하도록 선택합니다.

   - 가용 영역(Availability Zones)을 선택한 경우 인스턴스 유형(Instance type), 운영 체제(플랫폼(Platform)) 및 가격 기록을 볼 날짜 범위(Date range)를 선택합니다.
   - 인스턴스 유형(Instance Types)을 선택한 경우 최대 5개의 인스턴스 유형(Instance type(s)), 가용 영역(Availability Zone), 운영 체제(플랫폼(Platform)) 및 가격 기록을 볼 날짜 범위(Date range)를 선택합니다.

   다음 스크린샷은 여러 인스턴스 유형에 대한 요금 비교를 보여 줍니다.



5. 그래프 위로 마우스를 가져가면(포인터 이동) 선택한 날짜 범위의 특정 시간의 가격이 표시됩니다. 요금은 그래프 위의 정보 블록에 표시됩니다. 맨 위 행에 표시된 요금은 특정 날짜의 요금을 보여줍니다. 두 번째 행에 표시된 요금은 선택한 날짜 범위의 평균 요금을 보여줍니다.
6. vCPU당 요금을 표시하려면 정규화된 요금 표시를 켭니다. 인스턴스 유형에 대한 요금을 표시하려면 정규화된 가격 표시를 끕니다.

AWS CLI

스팟 가격 기록을 보는 방법

다음 describe-spot-price-history 명령을 사용합니다.

```
aws ec2 describe-spot-price-history \
    --instance-types c6i.xlarge \
    --product-descriptions "Linux/UNIX" \
    --start-time 2025-04-01T00:00:00 \
```

```
    --end-time 2025-04-02T00:00:0
```

PowerShell

스팟 가격 기록을 보는 방법

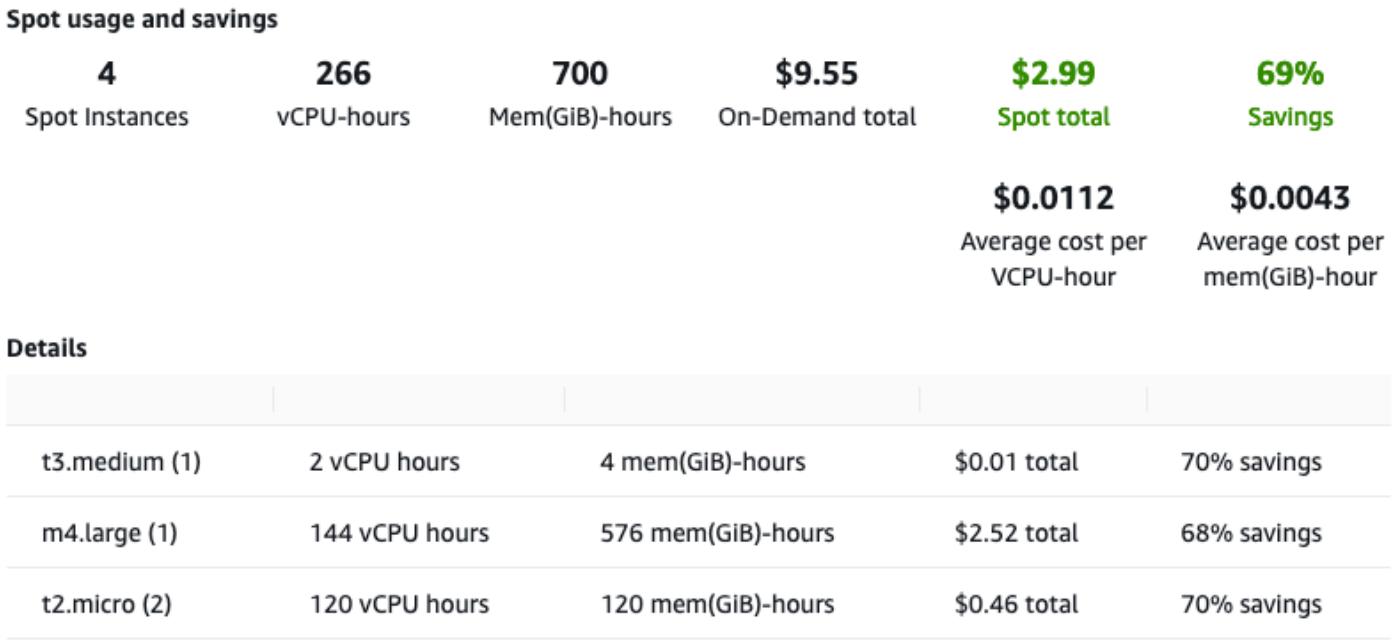
Get-EC2SpotPriceHistory cmdlet을 사용합니다.

```
Get-EC2SpotPriceHistory `
    -InstanceType c6i.xlarge `
    -ProductDescription "Linux/UNIX" `
    -UtcStartTime 2025-04-01T00:00:00 `
    -UtcEndTime 2025-04-02T00:00:0
```

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.

[비용 절감(Savings)] 섹션의 다음 스크린샷에는 스팟 플릿에 대한 스팟 사용 및 비용 절감 정보가 표시됩니다.



다음 사용 및 비용 절감 정보를 볼 수 있습니다.

- 스팟 인스턴스 – 스팟 플릿에서 시작되고 종료되는 스팟 인스턴스의 수입니다. 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 플릿을 구성하는 여러 인스턴스 유형입니다. 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 플릿에 대한 비용 절감 정보를 보는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. 스팟 플릿 요청의 ID를 선택하고 [비용 절감(Savings)] 섹션으로 스크롤합니다.

   또는 스팟 플릿 요청 ID 옆에 있는 확인란을 선택하고 절감 탭을 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 보는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. [비용 절감 요약(Savings Summary)]을 선택합니다.

스팟 인스턴스 요청 생성

스팟 인스턴스를 사용하려면 원하는 인스턴스 수, 인스턴스 유형, 가용 영역이 포함된 스팟 인스턴스 요청을 생성합니다. 용량을 사용할 수 있는 경우 Amazon EC2는 요청을 즉시 이행합니다. 그렇게 하지 않으면 요청이 이행될 수 있을 때까지 또는 사용자가 요청을 취소할 때까지 Amazon EC2가 대기합니다.

Amazon EC2 콘솔의 인스턴스 시작 마법사 또는 run-instances 명령을 사용하여 온디맨드 인스턴스를 시작하는 것과 동일한 방식으로 스팟 인스턴스를 요청할 수 있습니다. 이 방법은 다음과 같은 이유로만 권장됩니다.

- 이미 인스턴스 시작 마법사 또는 run-instances 명령을 사용하여 온디맨드 인스턴스를 시작하고 있으며 단일 파라미터를 변경하여 스팟 인스턴스 시작으로 변경하려고 합니다.
- 인스턴스 유형이 다른 여러 인스턴스는 필요하지 않습니다.

이 방법은 여러 인스턴스 유형을 지정할 수 없고 동일한 요청에서 스팟 인스턴스와 온디맨드 인스턴스를 시작할 수 없기 때문에 일반적으로 스팟 인스턴스를 시작하는 데 권장되지 않습니다. 스팟 인스턴스와 여러 인스턴스 유형의 온디맨드 인스턴스를 포함하는 플릿을 시작하는 방법을 비롯하여 스팟 인스턴스를 시작하는 데 선호되는 방법은 어느 스팟 요청 방법을 사용하는 것이 최선인가요? 섹션을 참조하세요.

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 인스턴스 요청 상태 가져오기 섹션을 참조하세요.

Console

스팟 인스턴스 요청을 생성하려면

1~9단계는 온디맨드 인스턴스를 시작하는 데 사용하는 단계와 동일합니다. 10단계에서는 스팟 인스턴스 요청을 구성합니다.

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

2. 화면 상단의 탐색 모음에서 리전을 선택합니다.
3. Amazon EC2 콘솔 대시보드에서 인스턴스 시작을 선택합니다.
4. (선택 사항) 이름 및 태그(Name and tags)에서 인스턴스의 이름을 지정하고 스팟 인스턴스 요청, 인스턴스, 볼륨 및 탄력적 그래픽에 태깅할 수 있습니다. 태그에 대한 자세한 내용은 Amazon EC2 리소스 태깅 섹션을 참조하세요.

   a. 이름(Name)에 인스턴스를 설명하는 이름을 입력합니다.

      인스턴스 이름은 태그이며, 여기서 키는 이름이고 값은 사용자가 지정하는 이름입니다. 이름을 지정하지 않으면 인스턴스를 시작할 때 자동으로 생성되는 ID로 인스턴스를 식별할 수 있습니다.

   b. 스팟 인스턴스 요청, 인스턴스, 볼륨 및 탄력적 그래픽에 태깅하려면 추가 태그 추가(Add additional tags)를 선택합니다. 태그 추가(Add tag)를 선택한 다음 키와 값을 입력하고 태그를 지정할 리소스 유형을 선택합니다. 추가할 각 추가 태그에 대해 태그 추가(Add tag)를 다시 선택합니다.
5. 애플리케이션 및 OS 이미지(Amazon Machine Image)(Application and OS Images (Amazon Machine Image))에서 인스턴스의 운영 체제(OS)를 선택한 다음 AMI를 선택합니다. 자세한 내용은 애플리케이션 및 OS 이미지(Amazon Machine Image) 섹션을 참조하세요.
6. 인스턴스 유형(Instance type)에서 하드웨어 구성 및 인스턴스 크기에 대한 요구 사항을 충족하는 인스턴스 유형을 선택합니다. 자세한 내용은 인스턴스 유형 섹션을 참조하세요.
7. 키 페어(로그인)(Key pair (login))에서 기존 키 페어를 선택하거나 새로운 키 페어 생성(Create new key pair)을 선택하여 새로 생성합니다. 자세한 내용은 Amazon EC2 키 페어 및 Amazon EC2 인스턴스 섹션을 참조하세요.

   > **Important**
   >
   > 키 페어 없이 진행(Proceed without key pair)(권장하지 않음) 옵션을 선택할 경우 사용자가 다른 방법으로 로그인할 수 있도록 구성된 AMI를 선택해야만 인스턴스에 연결할 수 있습니다.
8. 네트워크 설정(Network settings)에서 기본 설정을 사용하거나 편집(Edit)을 선택하여 필요에 따라 네트워크 설정을 구성합니다.

   보안 그룹은 네트워크 설정의 일부를 구성하고 인스턴스에 대한 방화벽 규칙을 정의합니다. 이 규칙은 인스턴스에 전달되는 수신 네트워크 트래픽을 정의합니다.

   자세한 내용은 네트워크 설정 섹션을 참조하세요.

9. 선택한 AMI에는 루트 디바이스 볼륨을 포함한 하나 이상의 스토리지 볼륨이 있습니다. 스토리지 구성(Configure storage) 페이지에서 새 볼륨 추가(Add new volume)를 선택하여 인스턴스에 연결할 추가 볼륨을 지정할 수 있습니다. 자세한 내용은 스토리지 구성 섹션을 참조하세요.
10. 고급 세부 정보(Advanced details)에서 스팟 인스턴스 요청을 다음과 같이 구성합니다.

    a. 구매 옵션에서 스팟 인스턴스 요청 확인란을 선택합니다.

    b. 스팟 인스턴스 요청에 대한 기본 구성을 유지하거나 사용자 지정(Customize)(오른쪽)을 선택하여 스팟 인스턴스 요청에 대한 사용자 지정 설정을 지정할 수 있습니다.

    사용자 지정(Customize)을 선택하면 다음 필드가 나타납니다.

    i. 최고 가격(Maximum price): 스팟 가격(온디맨드 가격으로 제한됨)으로 스팟 인스턴스를 요청하거나 지불할 의향이 있는 최대 금액을 지정할 수 있습니다.

    > **Warning**
    >
    > 최고 가격을 지정하면 최고 가격 없음(No maximum price)을 선택하는 경우보다 인스턴스가 더 자주 중단됩니다.
    > 최고 가격을 지정하는 경우 USD $0.001보다 커야 합니다. USD $0.001 미만의 값을 지정하면 시작이 실패합니다.

    - 최고 가격 없음(No maximum price): 스팟 인스턴스가 현재 스팟 가격으로 시작됩니다. 가격은 온디맨드 가격을 초과하지 않습니다. (권장)
    - 최고 가격 설정(인스턴스/시간당)(Set your maximum price (per instance/hour)): 지불할 의향이 있는 최고 금액을 지정할 수 있습니다.
      - 현재 스팟 가격보다 낮은 최고가를 지정하면 스팟 인스턴스가 시작되지 않습니다.
      - 현재 스팟 가격보다 높은 최고 가격을 지정하면 스팟 인스턴스가 현재 스팟 가격으로 시작되고 현재 스팟 가격을 기준으로 요금이 부과됩니다. 스팟 인스턴스가 실행된 후 스팟 가격이 최고 가격보다 높아지면 Amazon EC2가 스팟 인스턴스를 중단합니다.
      - 지정한 최고 가격과 관계없이 항상 현재 스팟 가격을 기준으로 비용이 청구됩니다.

    스팟 가격 추세를 검토하려면 스팟 인스턴스 요금 기록 보기 섹션을 참조하세요.

ii. 요청 유형(Request type): 선택하는 스팟 인스턴스 요청 유형에 따라 스팟 인스턴스가 중단될 때의 동작이 결정됩니다.

- 일회성(One-time): Amazon EC2가 스팟 인스턴스에 대해 일회성 요청을 보냅니다. 스팟 인스턴스가 중단되면 요청이 다시 제출되지 않습니다.
- 영구 요청(Persistent request): Amazon EC2가 스팟 인스턴스에 대한 영구 요청을 합니다. 스팟 인스턴스가 중단되면 중단된 스팟 인스턴스를 보충하기 위해 요청이 다시 제출됩니다.

값을 지정하지 않으면 기본값은 일회성 요청입니다.

iii. 유효 기간 종료(Valid to): 영구 스팟 인스턴스 요청의 만료 날짜입니다.

일회성 요청에는 이 필드가 지원되지 않습니다. 일회성 요청은 요청의 모든 인스턴스가 시작되거나 요청을 취소할 때까지 활성 상태로 유지됩니다.

- 요청 만료 날짜 없음(No request expiry date): 요청이 취소될 때까지 활성 상태로 유지됩니다.
- 요청 만료 날짜 설정(Set your request expiry date): 영구 요청은 지정한 날짜까지 또는 취소될 때까지 활성 상태로 유지됩니다.

iv. 인터럽트 방식(Interruption behavior): 선택하는 동작에 따라 스팟 인스턴스가 중단될 때의 동작이 결정됩니다.

- 영구 요청의 경우 유효한 값은 중지(Stop)와 최대 절전(Hibernate)입니다. 인스턴스가 중지되면 EBS 볼륨 스토리지에 대한 요금이 적용됩니다.

> **Note**
>
> 이제 온디맨드 인스턴스와 동일한 최대 절전 모드 기능을 스팟 인스턴스에서 사용합니다. 최대 절전 모드를 활성화하려면 여기에서 최대 절전 모드를 선택하거나 시작 인스턴스 마법사의 아래쪽에 나타나는 중지 - 최대 절전 모드 동작 필드에서 활성화를 선택하면 됩니다. 최대 절전 모드 사전 조건은 EC2 인스턴스 최대 절전 모드를 위한 사전 조건의 섹션을 참조하세요.

- 일회성 요청의 경우 종료(Terminate)만 유효합니다.

값을 지정하지 않으면 기본값은 영구 스팟 인스턴스 요청에 유효하지 않은 종료(Terminate)입니다. 기본값을 유지하고 영구 스팟 인스턴스 요청을 시작하려고 하면 오류가 발생합니다.

자세한 내용은 스팟 인스턴스 중단 동작 섹션을 참조하세요.

11. 요약(Summary) 패널의 인스턴스 수(Number of instances)에 시작할 인스턴스 수를 입력합니다.

> **Note**
>
> Amazon EC2는 각 스팟 인스턴스에 대해 별도의 요청을 생성합니다.

12. 요약(Summary) 패널에서 인스턴스의 세부 정보를 검토하고 필요에 따라 변경합니다. 스팟 인스턴스 요청을 제출한 후에는 요청의 파라미터를 변경할 수 없습니다. 요약(Summary) 패널에서 해당 링크를 선택하여 인스턴스 시작 마법사의 섹션으로 직접 이동할 수 있습니다. 자세한 내용은 요약 섹션을 참조하세요.
13. 인스턴스를 시작할 준비가 되면 인스턴스 시작(Launch instance)을 선택합니다.

    인스턴스가 시작하지 않거나 상태가 `terminated`이 아닌 `running`로 변경되는 경우, Amazon EC2 인스턴스 시작 문제 해결 섹션을 참조하세요.

AWS CLI

run-instances를 사용하여 스팟 인스턴스 요청 생성

run-instances 명령을 사용하고 다음과 같이 `--instance-market-options` 파라미터에 스팟 인스턴스 옵션을 지정합니다.

```
--instance-market-options file://spot-options.json
```

다음은 JSON 파일에 지정할 데이터 구조입니다. `ValidUntil` 및 `InstanceInterruptionBehavior`도 지정할 수 있습니다. 데이터 구조에서 필드를 지정하지 않으면 기본값이 사용됩니다.

다음 예제에서는 `persistent` 요청을 만듭니다.

```
{
```

```
  "MarketType": "spot",
  "SpotOptions": {
    "SpotInstanceType": "persistent"
  }
}
```

request-spot-instances를 사용하여 스팟 인스턴스 요청을 생성하는 방법

> **Note**
>
> 스팟 인스턴스는 계획된 투자가 없는 레거시 API이므로 request-spot-instances 명령을 사용하여 스팟 인스턴스를 요청하지 않는 것이 좋습니다. 자세한 내용은 어느 스팟 요청 방법을 사용하는 것이 최선인가요? 섹션을 참조하세요.

request-spot-instances 명령을 사용하여 일회성 요청을 생성합니다.

```
aws ec2 request-spot-instances \
    --instance-count 5 \
    --type "one-time" \
    --launch-specification file://specification.json
```

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 인스턴스 요청 예제 시작 사양 섹션을 참조하세요. 스팟 요청 콘솔에서 시작 사양 파일을 다운로드하는 경우 request-spot-fleet 명령을 대신 사용해야 합니다(스팟 요청 콘솔은 스팟 플릿을 사용하여 스팟 인스턴스 요청을 지정함).

PowerShell

스팟 인스턴스 요청을 생성하려면

New-EC2Instance cmdlet을 사용하고 `-InstanceMarketOption` 파라미터를 사용하여 스팟 인스턴스 옵션을 지정합니다.

```
-InstanceMarketOptions $marketOptions
```

다음과 같이 스팟 인스턴스 옵션에 대한 데이터 구조를 생성합니다.

```
$spotOptions = New-Object Amazon.EC2.Model.SpotMarketOptions
$spotOptions.SpotInstanceType="persistent"
$marketOptions = New-Object Amazon.EC2.Model.InstanceMarketOptionsRequest
$marketOptions.MarketType = "spot"
$marketOptions.SpotOptions = $spotOptions
```

스팟 인스턴스 요청 예제 시작 사양

다음 예제에서는 request-spot-instances 명령과 함께 사용하여 스팟 인스턴스 요청을 생성할 수 있는 시작 구성을 보여줍니다. 자세한 내용은 스팟 인스턴스 관리 섹션을 참조하세요.

> **Important**
>
> 스팟 인스턴스는 계획된 투자가 없는 레거시 API이므로 request-spot-instances 명령을 사용하여 스팟 인스턴스를 요청하지 않는 것이 좋습니다. 자세한 내용을 알아보려면 어느 스팟 요청 방법을 사용하는 것이 최선인가요? 섹션을 참조하세요.

예시

- 예 1: 스팟 인스턴스 시작
- 예제 2: 지정된 가용 영역에서 스팟 인스턴스 시작
- 예제 3: 지정된 서브넷에서 스팟 인스턴스 시작
- 예제 4: 전용 스팟 인스턴스 시작

예 1: 스팟 인스턴스 시작

다음 예제에는 가용 영역 또는 서브넷이 포함되지 않습니다. Amazon EC2는 가용 영역을 선택합니다. Amazon EC2는 선택된 가용 영역의 기본 서브넷에서 인스턴스를 시작합니다.

```
{
  "ImageId": "ami-0abcdef1234567890",
  "KeyName": "my-key-pair",
  "SecurityGroupIds": [ "sg-1a2b3c4d5e6f7g8h9" ],
  "InstanceType": "m5.medium",
  "IamInstanceProfile": {
```

```
      "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

예제 2: 지정된 가용 영역에서 스팟 인스턴스 시작

다음 예제에는 가용 영역이 포함됩니다. Amazon EC2는 지정된 가용 영역의 기본 서브넷에서 인스턴스를 시작합니다.

```
{
  "ImageId": "ami-0abcdef1234567890",
  "KeyName": "my-key-pair",
  "SecurityGroupIds": [ "sg-1a2b3c4d5e6f7g8h9" ],
  "InstanceType": "m5.medium",
  "Placement": {
    "AvailabilityZone": "us-west-2a"
  },
  "IamInstanceProfile": {
      "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

예제 3: 지정된 서브넷에서 스팟 인스턴스 시작

다음 예제에는 서브넷이 포함됩니다. Amazon EC2는 지정된 서브넷에서 인스턴스를 시작합니다. VPC가 기본이 아닌 VPC인 경우, 인스턴스는 기본적으로 퍼블릭 IPv4 주소를 받지 않습니다.

```
{
  "ImageId": "ami-0abcdef1234567890",
  "SecurityGroupIds": [ "sg-1a2b3c4d5e6f7g8h9" ],
  "InstanceType": "m5.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 지정하세요. 네트워크 인터페이스를 지정할 때 이전 코드 블록에 표시된 `SubnetId` 및 `SecurityGroupIds` 필드를 사용하는 대신 네트워크 인터페이스를 사용하여 서브넷 ID 및 보안 그룹 ID를 포함해야 합니다.

```
{
  "ImageId": "ami-0abcdef1234567890",
  "KeyName": "my-key-pair",
  "InstanceType": "m5.medium",
  "NetworkInterfaces": [
    {
      "DeviceIndex": 0,
      "SubnetId": "subnet-1a2b3c4d5e6f7g8h9",
      "Groups": [ "sg-1a2b3c4d5e6f7g8h9" ],
      "AssociatePublicIpAddress": true
    }
  ],
  "IamInstanceProfile": {
      "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

예제 4: 전용 스팟 인스턴스 시작

다음 예제에서는 dedicated의 테넌시를 사용하여 스팟 인스턴스를 요청합니다. 전용 스팟 인스턴스는 VPC에서 시작되어야 합니다.

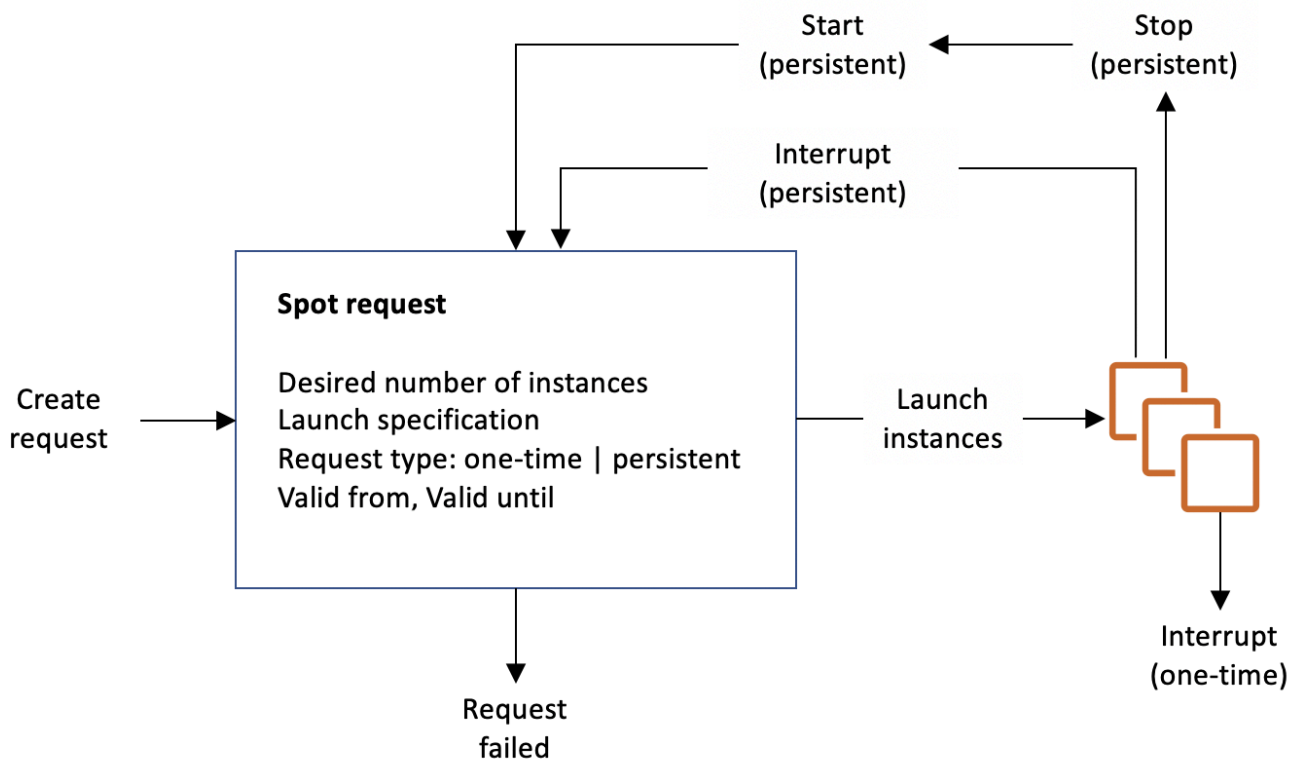
```
{
  "ImageId": "ami-0abcdef1234567890",
  "KeyName": "my-key-pair",
  "SecurityGroupIds": [ "sg-1a2b3c4d5e6f7g8h9" ],
  "InstanceType": "c5.8xlarge",
  "SubnetId": "subnet-1a2b3c4d5e6f7g8h9",
  "Placement": {
    "Tenancy": "dedicated"
  }
}
```

스팟 인스턴스 요청 상태 가져오기

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.

다음 그림에서는 스팟 인스턴스 요청이 작동하는 방식을 보여 줍니다. 요청 유형(일회성 또는 영구적)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.



내용

- 요청 상태 정보 가져오기
- 스팟 요청 상태 코드
- EC2 스팟 인스턴스 요청 이행 이벤트
- 스팟 요청의 상태 변경

요청 상태 정보 가져오기

스팟 인스턴스 요청에 대한 상태 정보를 가져올 수 있습니다.

Console

요청 상태 정보를 가져오는 방법

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 탭에서 상태 필드를 선택합니다.

AWS CLI

요청 상태 정보를 가져오는 방법

다음 describe-spot-instance-requests 명령을 사용합니다.

```
aws ec2 describe-spot-instance-requests --spot-instance-request-
ids sir-0e54a519c9EXAMPLE
```

PowerShell

요청 상태 정보를 가져오는 방법

Get-EC2SpotInstanceRequest cmdlet을 사용합니다.

```
Get-EC2SpotInstanceRequest -SpotInstanceRequestId sir-0e54a519c9EXAMPLE
```

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`

스팟 요청이 `active`이고 Amazon EC2에서 스팟 인스턴스를 시작하는 중입니다.

`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`

EC2 용량 관리 필요성으로 인해 인스턴스가 중지되었습니다.

`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 예약 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`이고(영구 요청이 아닌 경우) 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`

표준 용량 관리 프로세스로 인해 인스턴스가 종료되었습니다.

`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. 자세한 내용은 Amazon EBS 사용 설명서의 Quotas for Amazon EBS를 참조하세요.

`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 문의하여 지원을 받으세요.

EC2 스팟 인스턴스 요청 이행 이벤트

스팟 인스턴스 요청이 이행되면 Amazon EC2가 EC2 스팟 인스턴스 요청 이행 이벤트를 Amazon EventBridge로 전송합니다. 이 이벤트가 발생할 때마다 Lambda 함수를 호출하거나 Amazon SNS 주제를 알리는 등의 조치를 취하는 규칙을 생성할 수 있습니다.

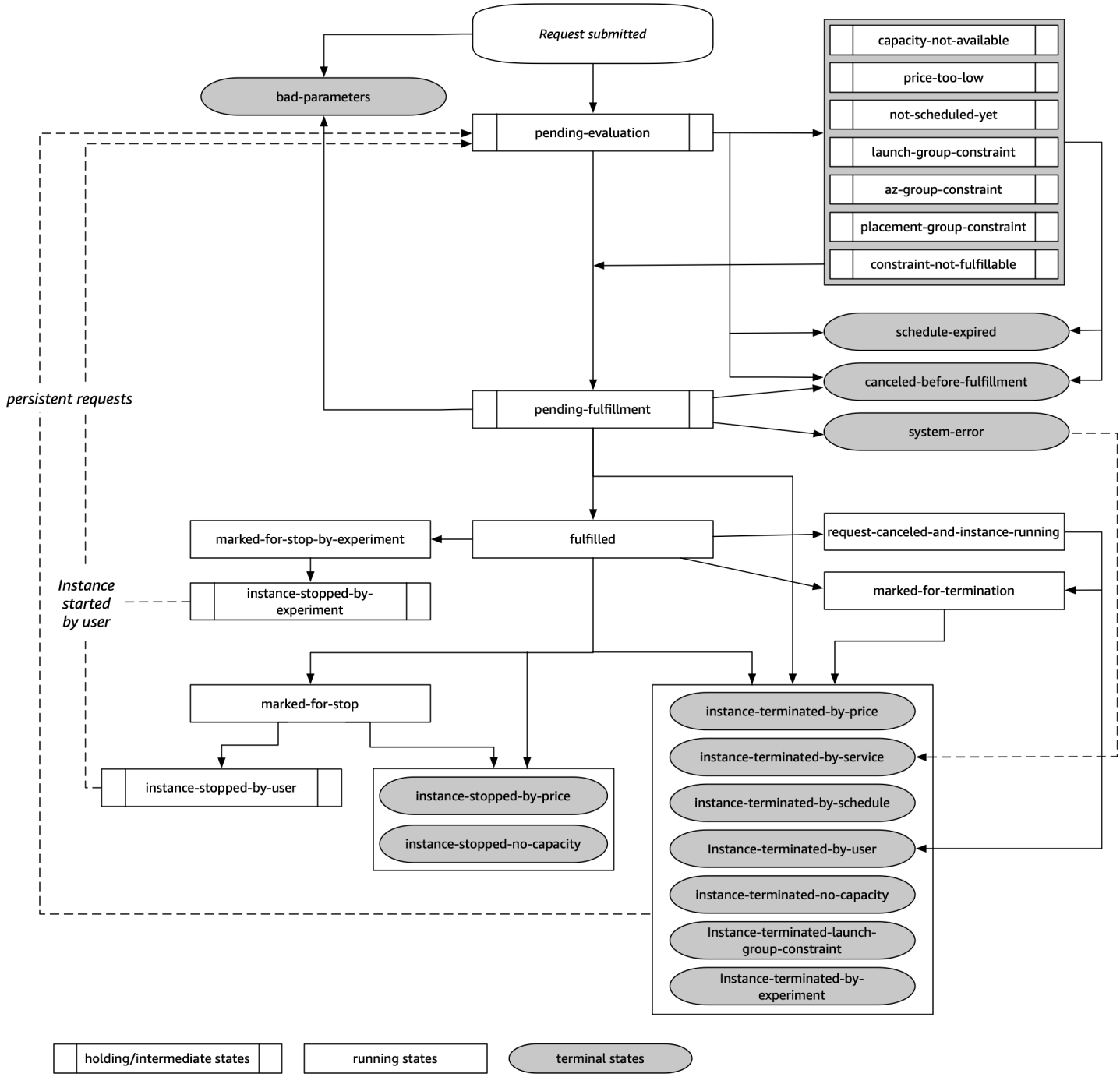
다음은 이 이벤트의 예제 데이터입니다.

```
{
    "version": "0",
    "id": "01234567-1234-0123-1234-012345678901",
    "detail-type": "EC2 Spot Instance Request Fulfillment",
    "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": {
        "spot-instance-request-id": "sir-0e54a519c9EXAMPLE",
        "instance-id": "i-1234567890abcdef0"
    }
}
```

자세한 내용은 Amazon EventBridge 사용 설명서를 참조하세요.

스팟 요청의 상태 변경

다음 다이어그램에서는 제출부터 종료까지 전체 수명 주기 동안 스팟 요청이 따를 수 있는 경로를 보여줍니다. 각 단계는 노드로 묘사되며 각 노드의 상태 코드는 스팟 요청 및 스팟 인스턴스의 상태를 설명합니다.



평가 보류

하나 이상의 요청 파라미터가 유효하지 않은 경우(`bad-parameters`)를 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 용량이 없는 경우 가용 용량이 가용할 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 영향을 받을 수 있습니다.

상태 코드	요청 상태	인스턴스 상태
capacity-not-available	open	해당 사항 없음
price-too-low	open	해당 사항 없음
not-scheduled-yet	open	해당 사항 없음
launch-group-constraint	open	해당 사항 없음
az-group-constraint	open	해당 사항 없음
placement-group-constraint	open	해당 사항 없음
constraint-not-fulfillable	open	해당 사항 없음

평가/이행 보류-끝

특정 기간 동안에만 유효한 요청을 생성하는 경우 요청이 이행 보류 단계에 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` [1]	`cancelled`	해당 사항 없음
`bad-parameters`	`failed`	해당 사항 없음
`system-error`	`closed`	해당 사항 없음

[1] 사용자가 요청을 취소하는 경우.

이행 보류

지정한 제약 조건(있는 경우)이 충족되면 스팟 요청이 `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 인스턴스에 대한 모든 사양이 충족되면 스팟 요청이 이행됩니다. Amazon EC2가 스팟 인스턴스를 시작합니다. 이 작업은 몇 분 정도 걸릴 수 있습니다. 중단된 스팟 인스턴스가 최대 절전 모드로 전환

되거나 중지되는 경우 요청을 다시 이행할 수 있거나 요청이 취소될 때까지 인스턴스는 이 상태를 유지합니다.

상태 코드	요청 상태	인스턴스 상태
fulfilled	active	pending → running
fulfilled	active	stopped → running

스팟 인스턴스를 중지하면 스팟 인스턴스를 다시 시작할 수 있거나 요청을 취소할 때까지 스팟 요청이 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 [1]	disabled 또는 cancelled [2]	stopped

[1] 인스턴스를 중지하거나 인스턴스에서 종료 명령을 실행하면 스팟 인스턴스가 instance-stopped-by-user 상태가 됩니다. 인스턴스를 중지한 후 다시 시작할 수 있습니다. 다시 시작하면 스팟 인스턴스 요청이 pending-evaluation 상태가 되고 제약 조건이 충족되면 Amazon EC2가 새 스팟 인스턴스를 시작합니다.

[2] 스팟 인스턴스를 중지하지만 요청을 취소하지 않는 경우 스팟 요청 상태는 disabled입니다. 스팟 인스턴스가 중지되고 요청이 만료되는 경우 요청 상태는 cancelled입니다.

이행됨-끝

스팟 인스턴스는 인스턴스 유형에 사용 가능한 용량이 있고 인스턴스를 종료하지 않는 한 계속 실행됩니다. Amazon EC2가 스팟 인스턴스를 종료해야 하는 경우 스팟 요청은 터미널 상태가 됩니다. 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

상태 코드	요청 상태	인스턴스 상태
marked-for-stop	active	running
marked-for-termination	active	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 [1]	terminated
instance-terminated-no-capacity	closed(일회), open(영구)	running †
instance-terminated-no-capacity	closed(일회), open(영구)	terminated
instance-terminated-launch-group-constraint	closed(일회), open(영구)	terminated

[1] 인스턴스를 종료하되 요청을 취소하지 않는 경우 요청 상태는 closed입니다. 인스턴스를 종료하고 요청을 취소하는 경우 요청 상태는 cancelled입니다. 스팟 요청을 취소하기 전에 스팟 인스턴스를 종료해도 Amazon EC2에서 스팟 인스턴스가 종료되었음을 감지하기 전까지 지연이 발생할 수 있습니다. 이 경우 요청 상태는 closed 또는 cancelled일 수 있습니다.

† 용량이 다시 필요하고 인스턴스가 중단 시 종료되도록 구성된 경우 Amazon EC2가 스팟 인스턴스를 중단하면 상태가 즉시 instance-terminated-no-capacity로 설정됩니다(marked-for-termination으로 설정되지 않음). 그러나 인스턴스가 스팟 인스턴스 중단 알림을 수신한 2분 기간을 반영하기 위해 인스턴스는 2분 동안 running 상태를 유지합니다. 2분 후 인스턴스 상태가 terminated로 설정됩니다.

중단 시험

AWS Fault Injection Service를 사용하여 스팟 인스턴스 중단을 시작하여 스팟 인스턴스의 애플리케이션이 어떻게 응답하는지 테스트할 수 있습니다. AWS FIS가 스팟 인스턴스를 중지하면 스팟 요청이 marked-for-stop-by-experiment 상태, 이어서 instance-stopped-by-experiment 상태로 진입합니다. AWS FIS가 인스턴스를 종료하면 스팟 요청은 instance-terminated-by-experiment 상태가 됩니다. 자세한 내용은 the section called “중단 시작” 섹션을 참조하세요.

상태 코드	요청 상태	인스턴스 상태
marked-for-stop-by-experiment	active	running
instance-stopped-by-experiment	disabled	stopped
instance-terminated-by-experiment	closed	terminated

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.

스팟 인스턴스 요청 태깅

스팟 인스턴스 요청을 쉽게 분류하고 관리하려면 사용자 지정 메타데이터로 이 요청을 태깅할 수 있습니다. 스팟 인스턴스 요청을 생성할 때 또는 생성한 후 스팟 인스턴스 요청에 태그를 할당할 수 있습니다. Amazon EC2 콘솔이나 명령줄 도구를 사용하여 태그를 지정할 수 있습니다.

스팟 인스턴스 요청을 태깅하는 경우 스팟 인스턴스 요청에서 시작된 인스턴스 및 볼륨은 자동으로 태깅되지 않습니다. 스팟 인스턴스 요청에서 시작된 인스턴스 및 볼륨을 명시적으로 태깅해야 합니다. 시작 중 또는 이후에 스팟 인스턴스 및 볼륨에 태그를 할당할 수 있습니다.

태그 작동 방식에 대한 자세한 내용은 Amazon EC2 리소스 태깅 섹션을 참조하세요.

내용

- 사전 조건
- 새 스팟 인스턴스 요청을 태깅하려면
- 기존 스팟 인스턴스 요청을 태깅하려면
- 스팟 인스턴스 요청 태그 보기

사전 조건

사용자에게 리소스에 태그를 지정할 수 있는 권한을 부여합니다. IAM 정책 및 예제 정책에 대한 자세한 내용은 예: 태그 리소스 섹션을 참조하세요.

생성하는 IAM 정책은 스팟 인스턴스 요청을 생성할 때 사용하는 방법에 따라 결정됩니다.

- 인스턴스 시작 마법사 또는 `run-instances`를 사용하여 스팟 인스턴스를 요청하는 경우 To grant a user the permission to tag resources when using the launch instance wizard or run-instances 섹션을 참조하세요.
- `request-spot-instances` 명령을 사용하여 스팟 인스턴스를 요청하는 방법에 대한 자세한 내용은 To grant a user the permission to tag resources when using request-spot-instances 섹션을 참조하세요.

사용자에게 인스턴스 시작 마법사 또는 run-instances를 사용할 때 리소스에 태그를 지정할 수 있는 권한 부여

다음을 포함하는 IAM 정책을 만듭니다.

- `ec2:RunInstances` 작업 사용자에게 인스턴스 시작 권한이 부여됩니다.

- Resource에 spot-instances-request을 지정합니다. 이렇게 하면 사용자가 스팟 인스턴스를 요청하는 스팟 인스턴스 요청을 생성할 수 있습니다.
- ec2:CreateTags 작업 사용자에게 태그 생성 권한이 부여됩니다.
- Resource에 *을 지정합니다. 이를 통해 사용자가 인스턴스 시작 중에 생성된 모든 리소스에 태그를 지정할 수 있습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowLaunchInstances",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
                "arn:aws:ec2:us-east-1:*:security-group/*",
                "arn:aws:ec2:us-east-1:*:key-pair/*",
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*",
                "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
            ]
        },
        {
            "Sid": "TagSpotInstanceRequests",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "*"
        }
    ]
}
```

RunInstances 작업을 사용하여 스팟 인스턴스 요청을 생성하고 생성 시 스팟 인스턴스 요청을 태깅하는 경우 Amazon EC2가 RunInstances 문에서 spot-instances-request 리소스를 평가하는 방법을 알고 있어야 합니다. 이는 IAM 정책에서 다음과 같이 평가됩니다.

- 생성 시 스팟 인스턴스 요청을 태깅하지 않으면 Amazon EC2가 RunInstances 문에서 `spot-instances-request` 리소스를 평가하지 않습니다.
- 생성 시 스팟 인스턴스 요청을 태깅하면 Amazon EC2가 RunInstances 문에서 `spot-instances-request` 리소스를 평가합니다.

따라서 `spot-instances-request` 리소스의 경우 IAM 정책에 다음 규칙이 적용됩니다.

- RunInstances를 사용하여 스팟 인스턴스 요청을 생성하고 생성 시 스팟 인스턴스 요청을 태깅하지 않으려는 경우 `spot-instances-request` 리소스를 명시적으로 허용할 필요가 없습니다. 호출이 성공합니다.
- RunInstances를 사용하여 스팟 인스턴스 요청을 생성하고 생성 시 스팟 인스턴스 요청을 태깅하려는 경우 RunInstances allow 문에 `spot-instances-request` 리소스를 포함해야 합니다. 그러지 않으면 호출이 실패합니다.
- RunInstances를 사용하여 스팟 인스턴스 요청을 생성하고 생성 시 스팟 인스턴스 요청을 태깅하려는 경우 CreateTags allow 문에서 `spot-instances-request` 리소스를 지정하거나 * 와일드카드를 포함해야 합니다. 그러지 않으면 호출이 실패합니다.

스팟 인스턴스 요청에 지원되지 않는 정책을 포함한 예제 IAM 정책은 스팟 인스턴스 작업 섹션을 참조하세요.

사용자에게 request-spot-instances를 사용할 때 리소스에 태그를 지정할 수 있는 권한 부여

다음을 포함하는 IAM 정책을 만듭니다.

- `ec2:RequestSpotInstances` 작업 사용자에게 스팟 인스턴스 요청을 생성할 수 있는 권한이 부여됩니다.
- `ec2:CreateTags` 작업 사용자에게 태그 생성 권한이 부여됩니다.
- `Resource`에 `spot-instances-request`을 지정합니다. 이렇게 하면 사용자가 스팟 인스턴스 요청만 태깅할 수 있습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "TagSpotInstanceRequest",
            "Effect": "Allow",
            "Action": [
```

```
                "ec2:RequestSpotInstances",
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:us-east-1:111122223333:spot-instances-request/*"
        }
    ]
}
```

새 스팟 인스턴스 요청을 태깅하려면

AWS CLI 및 PowerShell 예제에서 스팟 인스턴스 요청을 다음과 같이 구성합니다.

- `ResourceType`에 `spot-instances-request`을 지정합니다. 다른 값을 지정하면 스팟 인스턴스 요청이 실패합니다.
- `Tags`에 대해 키-값 페어를 지정합니다. 둘 이상의 키-값 페어를 지정할 수 있습니다.

Console

새 스팟 인스턴스 요청에 태그를 지정하는 방법

1. 스팟 인스턴스 관리의 절차를 따르세요.
2. 태그를 추가하려면 태그 추가 페이지에서 태그 추가를 선택하고 해당 태그에 대한 키와 값을 입력합니다. 각 추가 태그에 다른 태그 추가를 선택합니다.

   각 태그에 대해 동일한 태그로 스팟 인스턴스 요청, 스팟 인스턴스 및 볼륨을 태깅할 수 있습니다. 세 가지 모두를 태깅하려면 [인스턴스(Instances)], [볼륨(Volumes)] 및 [스팟 인스턴스 요청(Spot Instance Requests)]이 선택되어 있는지 확인합니다. 한두 개만 태그를 지정하려면 태그를 지정할 리소스가 선택되어 있고 나머지 리소스는 선택 취소되어 있는지 확인합니다.
3. 필수 필드를 입력하여 스팟 인스턴스 요청을 생성한 다음 [시작(Launch)]을 선택합니다. 자세한 내용은 스팟 인스턴스 관리 섹션을 참조하세요.

AWS CLI

새 스팟 인스턴스 요청에 태그를 지정하는 방법

request-spot-instances 명령을 `--tag-specification` 옵션과 함께 사용합니다.

태그 사양은 스팟 인스턴스 요청에 `Environment=Production` 및 `Cost-Center=123`이라는 2개의 태그를 추가합니다.

```
aws ec2 request-spot-instances \
    --instance-count 5 \
    --type "one-time" \
    --launch-specification file://specification.json \
    --tag-specification 'ResourceType=spot-instances-
request,Tags=[{Key=Environment,Value=Production},{Key=Cost-Center,Value=123}]'
```

PowerShell

새 스팟 인스턴스 요청에 태그를 지정하는 방법

Request-EC2SpotInstance cmdlet을 `-TagSpecification` 파라미터와 함께 사용합니다.

```
-TagSpecification $tagspec
```

태그 사양은 다음과 같이 정의됩니다. 스팟 인스턴스 요청에 `Environment=Production` 및 `Cost-Center=123`이라는 2개의 태그를 추가합니다.

```
$tag1 = @{Key="Environment"; Value="Production"}
$tag2 = @{Key="Cost-Center"; Value="123"}
$tagspec = New-Object Amazon.EC2.Model.TagSpecification
$tagspec.ResourceType = "spot-instances-request"
$tagspec.Tags = @($tag1,$tag2)
```

기존 스팟 인스턴스 요청을 태깅하려면

Console

기존 스팟 인스턴스 요청에 태그를 지정하는 방법

스팟 인스턴스 요청을 생성한 후 콘솔을 사용하여 스팟 인스턴스 요청에 태그를 추가할 수 있습니다.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. 스팟 인스턴스 요청을 선택합니다.
4. 태그 탭을 선택하고 태그 생성을 선택합니다.

콘솔을 사용하여 기존 스팟 인스턴스를 태깅하려면

스팟 인스턴스 요청에서 스팟 인스턴스가 시작된 후 콘솔을 사용하여 인스턴스에 태그를 추가할 수 있습니다. 자세한 내용은 콘솔을 사용하여 태그 추가 섹션을 참조하세요.

AWS CLI

기존 스팟 인스턴스 요청 또는 스팟 인스턴스에 태그를 지정하려면

create-tags 명령을 사용하여 기존 리소스에 태그를 지정합니다. 다음 예제에서는 기존 스팟 인스턴스 요청 및 스팟 인스턴스에 purpose=test 태그가 지정됩니다.

```
aws ec2 create-tags \
    --resources sir-0e54a519c9EXAMPLE i-1234567890abcdef0 \
    --tags Key=purpose,Value=test
```

PowerShell

기존 스팟 인스턴스 요청 또는 스팟 인스턴스에 태그를 지정하려면

New-EC2Tag cmdlet을 사용합니다. 다음 예제는 기존 스팟 인스턴스 요청 및 스팟 인스턴스에 purpose=test 태그를 추가합니다.

```
New-EC2Tag `
    -Resource sir-0e54a519c9EXAMPLE, i-1234567890abcdef0 `
    -Tag @{Key="purpose"; Value="test"}
```

스팟 인스턴스 요청 태그 보기

Console

스팟 인스턴스 요청 태그를 보는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. 스팟 인스턴스 요청을 선택하고 [태그(Tags)] 탭을 선택합니다.

AWS CLI

스팟 인스턴스 요청 태그를 설명하려면

describe-spot-instance-requests 명령을 사용하여 지정된 스팟 인스턴스 요청의 구성을 볼 수 있습니다. 여기에는 요청에 지정된 모든 태그가 포함됩니다.

```
aws ec2 describe-spot-instance-requests \
    --spot-instance-request-ids sir-0e54a519c9EXAMPLE \
    --query "SpotInstanceRequests[*].Tags"
```

출력의 예시는 다음과 같습니다.

```
[
    [
        {
            "Key": "Environment",
            "Value": "Production"
        },
        {
            "Key": "Department",
            "Value": "101"
        }
    ]
]
```

PowerShell

스팟 인스턴스 요청 태그를 설명하려면

Get-EC2SpotInstanceRequest cmdlet을 사용합니다.

```
(Get-EC2SpotInstanceRequest `
    -SpotInstanceRequestId sir-0e54a519c9EXAMPLE).Tags
```

출력의 예시는 다음과 같습니다.

```
Key         Value
---         -----
Environment Production
Department  101
```

스팟 인스턴스 요청 취소

스팟 인스턴스 요청이 더 이상 필요하지 않은 경우 취소할 수 있습니다. open, active 또는 disabled 상태인 스팟 인스턴스 요청만 취소할 수 있습니다.

- 요청이 아직 이행되지 않았고 인스턴스가 시작되지 않았을 때 스팟 인스턴스 요청 상태는 open입니다.
- 요청이 이행되었고 그 결과로 스팟 인스턴스가 시작된 경우 스팟 인스턴스 요청 상태는 active입니다.
- 스팟 인스턴스를 중지하면 스팟 인스턴스 요청이 disabled 상태가 됩니다.

스팟 인스턴스 요청이 active 상태이고 실행 중인 스팟 인스턴스가 연결되어 있을 때 요청을 취소하면 인스턴스가 종료되지 않습니다. 스팟 인스턴스 종료에 대한 자세한 내용은 스팟 인스턴스 종료 섹션을 참조하세요.

Console

스팟 인스턴스 요청을 취소하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. 스팟 인스턴스 요청을 선택합니다.
4. 작업을 선택한 후, 요청 취소를 선택합니다.
5. (선택 사항) 연결된 스팟 인스턴스에 대한 작업을 완료했으면 종료할 수 있습니다. 스팟 요청 취소 대화 상자에서 인스턴스 종료를 선택한 다음 확인을 선택합니다.

AWS CLI

스팟 인스턴스 요청을 취소하는 방법

다음 cancel-spot-instance-requests 명령을 사용합니다.

```
aws ec2 cancel-spot-instance-requests --spot-instance-request-
ids sir-0e54a519c9EXAMPLE
```

PowerShell

스팟 인스턴스 요청을 취소하는 방법

Stop-EC2SpotInstanceRequest cmdlet을 사용합니다.

```
Stop-EC2SpotInstanceRequest -SpotInstanceRequestId sir-0e54a519c9EXAMPLE
```

스팟 인스턴스 관리

Amazon EC2는 용량이 사용 가능할 때 스팟 인스턴스를 시작합니다. 스팟 인스턴스는 중단되거나 사용자가 직접 종료할 때까지 실행됩니다.

내용

- 스팟 인스턴스 찾기
- 특정 요청에 따라 시작된 인스턴스 찾기
- 스팟 인스턴스 중지
- 스팟 인스턴스 시작
- 스팟 인스턴스 종료

스팟 인스턴스 찾기

스팟 인스턴스는 온디맨드 인스턴스와 함께 콘솔의 인스턴스 페이지에 나타납니다. 다음 절차에 따라 스팟 인스턴스를 찾습니다.

Console

스팟 인스턴스를 찾는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 모든 스팟 인스턴스를 찾으려면 검색 창에서 인스턴스 수명 주기=스팟을 선택합니다.
4. 인스턴스가 스팟 인스턴스인지 확인하려면 인스턴스를 선택하고 세부 정보 탭을 선택한 다음 수명 주기 값을 확인합니다. 스팟 인스턴스의 값은 `spot` 이고 온디맨드 인스턴스의 값은 `normal`입니다.

AWS CLI

스팟 인스턴스를 찾는 방법

다음 describe-instances 명령을 사용합니다.

```
aws ec2 describe-instances --filters "Name=instance-lifecycle,Values=spot"
```

인스턴스가 스팟 인스턴스인지 여부를 확인하려면

다음 describe-instances 명령을 사용합니다.

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query "Reservations[*].Instances[*].InstanceLifecycle" \
    --output text
```

출력이 spot이라면 인스턴스는 스팟 인스턴스입니다. 출력이 없다면 인스턴스는 온디맨드 인스턴스입니다.

PowerShell

스팟 인스턴스를 찾는 방법

Get-EC2Instance cmdlet을 사용합니다.

```
Get-EC2Instance -Filter @{Name="instance-lifecycle"; Values="spot"}
```

인스턴스가 스팟 인스턴스인지 여부를 확인하려면

Get-EC2Instance cmdlet을 사용합니다.

```
(Get-EC2Instance -InstanceId i-1234567890abcdef0).Instances.InstanceLifecycle
```

출력이 Spot이라면 인스턴스는 스팟 인스턴스입니다. 출력이 없다면 인스턴스는 온디맨드 인스턴스입니다.

특정 요청에 따라 시작된 인스턴스 찾기

다음 절차를 사용하여 특정 스팟 인스턴스 또는 스팟 플릿 요청에서 시작된 스팟 인스턴스를 찾을 수 있습니다.

Console

요청에 대한 스팟 인스턴스를 찾는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다. 목록에는 스팟 인스턴스 요청과 스팟 플릿 요청이 모두 포함되어 있습니다.

3. 스팟 인스턴스 요청이 이행된 경우 용량은 스팟 인스턴스의 ID입니다. 스팟 플릿의 경우 [용량(Capacity)]은 요청된 용량 중 이행된 용량을 나타냅니다. 스팟 플릿의 인스턴스 ID를 보려면 확장 화살표를 선택하거나 플릿을 선택한 후 [인스턴스(Instances)]를 선택합니다.
4. 스팟 플릿의 경우 용량은 요청된 용량 중 이행된 용량을 나타냅니다. 스팟 플릿의 인스턴스 ID를 보려면 플릿 ID를 선택하여 세부 정보 페이지를 열고 인스턴스 창을 찾습니다.

AWS CLI

요청에 대한 스팟 인스턴스를 찾는 방법

다음 describe-spot-instance-requests 명령을 사용합니다.

```
aws ec2 describe-spot-instance-requests \
    --spot-instance-request-ids sir-0e54a519c9EXAMPLE \
    --query "SpotInstanceRequests[*].{ID:InstanceId}"
```

다음은 예 출력입니다.

```
[
    {
        "ID": "i-1234567890abcdef0"
    },
    {
        "ID": "i-0598c7d356eba48d7"
    }
]
```

PowerShell

요청에 대한 스팟 인스턴스를 찾는 방법

Get-EC2SpotInstanceRequest cmdlet을 사용합니다.

```
(Get-EC2SpotInstanceRequest -SpotInstanceRequestId sir-0e54a519c9EXAMPLE).InstanceId
```

스팟 인스턴스 중지

지금 스팟 인스턴스가 필요하지 않지만, 나중에 Amazon EBS 볼륨에 유지된 데이터를 잃어버리지 않고 다시 시작하려면 스팟 인스턴스를 중지할 수 있습니다. 스팟 인스턴스를 중지하는 단계는 온디맨드 인스턴스를 중지하는 단계와 비슷합니다.

> **Note**
>
> 스팟 인스턴스가 중지되었을 때 일부 인스턴스 속성을 수정할 수 있지만 인스턴스 유형은 수정할 수 없습니다.
> 중지된 스팟 인스턴스에 대해 사용 요금이나 데이터 전송 요금이 부과되지는 않지만 모든 Amazon EBS 볼륨에 대한 스토리지 요금은 부과됩니다.

제한 사항

- 스팟 인스턴스가 `persistent` 스팟 인스턴스 요청에서 시작된 경우에만 스팟 인스턴스를 중지할 수 있습니다.
- 연결된 스팟 인스턴스 요청이 취소된 경우에는 스팟 인스턴스를 중지할 수 없습니다. 스팟 인스턴스 요청이 취소되면 스팟 인스턴스를 종료하는 작업만 수행할 수 있습니다.
- 플릿 또는 시작 그룹이나 가용 영역 그룹의 일부인 경우 스팟 인스턴스를 중지할 수 없습니다.

Console

스팟 인스턴스를 중지하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 스팟 인스턴스를 선택합니다. 스팟 인스턴스의 인스턴스 ID를 저장하지 않은 경우 the section called “스팟 인스턴스 찾기”를 참조하세요.
4. 인스턴스 상태, 인스턴스 중지를 차례로 선택합니다.
5. 확인 메시지가 표시되면 [Stop]을 선택합니다.

AWS CLI

스팟 인스턴스를 중지하는 방법

스팟 인스턴스를 수동으로 중지하려면 stop-instances 명령을 사용합니다.

```
aws ec2 stop-instances --instance-ids i-1234567890abcdef0
```

PowerShell

스팟 인스턴스를 중지하는 방법

Stop-EC2Instance cmdlet을 사용합니다.

```
Stop-EC2Instance -InstanceId i-1234567890abcdef0
```

스팟 인스턴스 시작

이전에 중지한 스팟 인스턴스를 시작할 수 있습니다.

사전 조건

다음 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 그룹이나 가용 영역 그룹의 일부인 경우 스팟 인스턴스를 시작할 수 없습니다.

스팟 인스턴스를 시작하는 단계는 온디맨드 인스턴스를 시작하는 단계와 비슷합니다.

Console

스팟 인스턴스를 시작하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 스팟 인스턴스를 선택합니다. 스팟 인스턴스의 인스턴스 ID를 저장하지 않은 경우 the section called “스팟 인스턴스 찾기”를 참조하세요.
4. 인스턴스 상태, 인스턴스 시작을 차례로 선택합니다.

AWS CLI

스팟 인스턴스를 시작하는 방법

스팟 인스턴스를 수동으로 시작하려면 start-instance 명령을 사용합니다.

```
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

PowerShell

스팟 인스턴스를 시작하는 방법

Start-EC2Instance cmdlet을 사용합니다.

```
Start-EC2Instance -InstanceId i-1234567890abcdef0
```

스팟 인스턴스 종료

영구 스팟 인스턴스 요청에서 시작된 실행 중이거나 중지된 스팟 인스턴스를 종료하면 새 스팟 인스턴스를 시작할 수 있도록 스팟 인스턴스 요청이 open 상태로 전환됩니다. 새로운 스팟 인스턴스가 시작되지 않도록 먼저 스팟 인스턴스 요청을 취소해야 합니다.

실행 중인 스팟 인스턴스를 보유한 active 스팟 인스턴스 요청을 취소하는 경우 실행 중인 스팟 인스턴스가 자동으로 종료되지 않습니다. 스팟 인스턴스를 수동으로 종료해야 합니다.

중지된 스팟 인스턴스를 보유한 disabled 스팟 인스턴스 요청을 취소하는 경우 중지된 스팟 인스턴스가 Amazon EC2 스팟 서비스에 의해 자동으로 종료됩니다. 스팟 인스턴스 요청을 취소할 때와 스팟 서비스에서 스팟 인스턴스를 종료할 때 사이에는 짧은 지연이 있을 수 있습니다.

자세한 내용은 스팟 인스턴스 요청 취소 섹션을 참조하세요.

Console

스팟 인스턴스를 수동으로 종료하는 방법

1. 인스턴스를 종료하기 전에 Amazon EBS 볼륨이 종료 시 삭제되지 않는지 그리고 인스턴스 스토어 볼륨에서 영구 스토리지(예: Amazon EBS 또는 Amazon S3)로 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. 스팟 인스턴스를 선택합니다. 스팟 인스턴스의 인스턴스 ID를 저장하지 않은 경우 the section called “스팟 인스턴스 찾기”를 참조하세요.
5. 인스턴스 상태, 인스턴스 종료(삭제)를 차례로 선택합니다.
6. 확인 메시지가 나타나면 종료(삭제)를 선택합니다.

AWS CLI

스팟 인스턴스를 수동으로 종료하는 방법

스팟 인스턴스를 수동으로 종료하려면 terminate-instances 명령을 사용합니다.

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0 i-0598c7d356eba48d7
```

PowerShell

스팟 인스턴스를 수동으로 종료하는 방법

Remove-EC2Instance cmdlet을 사용합니다.

```
Remove-EC2Instance -InstanceId i-1234567890abcdef0
```

스팟 인스턴스 중단

Amazon EC2에 용량이 다시 필요할 때 반환하는 대가로 예비 EC2 용량에서 대폭 할인된 금액으로 스팟 인스턴스를 시작할 수 있습니다. Amazon EC2에서 스팟 인스턴스를 회수하는 이벤트를 스팟 인스턴스 중단이라고 합니다.

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. Amazon EC2에서 스팟 인스턴스를 중단시킬 수 있는 이유는 다음과 같습니다.

용량

Amazon EC2는 스팟 인스턴스가 다시 필요할 때 스팟 인스턴스를 중단할 수 있습니다. EC2는 주로 용량을 재활용하기 위해 인스턴스를 회수하지만 호스트 유지 관리 또는 하드웨어 중단과 같은 다른 이유로 회수할 수도 있습니다.

가격

스팟 가격이 최고 가격보다 높습니다.

스팟 요청에서 최고 가격을 지정할 수 있습니다. 그러나 최고 가격을 지정하면 지정하지 않을 때보다 인스턴스가 더 자주 중단됩니다.

제약 조건

요청에 시작 그룹 또는 가용 영역 그룹과 같은 제약 조건이 포함되는 경우 제약 조건을 더 이상 충족할 수 없으면 스팟 인스턴스가 그룹으로 종료됩니다.

Amazon EC2가 스팟 인스턴스를 중단하면 스팟 요청을 생성할 때 지정한 중단 동작에 따라 인스턴스를 종료, 중지 또는 최대 절전 모드로 전환합니다.

내용

- 스팟 인스턴스 중단 동작
- 스팟 인스턴스 중단 준비
- 스팟 인스턴스 중단 시작
- 스팟 인스턴스 중단 공지
- 중단된 스팟 인스턴스 찾기
- Amazon EC2가 스팟 인스턴스를 종료했는지 확인
- 중단된 스팟 인스턴스에 대한 청구

스팟 인스턴스 중단 동작

스팟 요청을 생성할 때 중단 동작을 지정할 수 있습니다. 사용 가능한 중단 동작은 다음과 같습니다.

- Stop
- 최대 절전 모드
- Terminate

기본 동작은 Amazon EC2가 스팟 인스턴스가 중단되면 해당 인스턴스를 종료한다는 것입니다.

중단된 스팟 인스턴스 중지

스팟 인스턴스가 중단되면 Amazon EC2가 스팟 인스턴스를 중지하도록 지정할 수 있습니다. 스팟 인스턴스 요청 유형은 `persistent`여야 합니다. 스팟 인스턴스 요청에서 시작 그룹을 지정할 수 없습니다. EC2 플릿 또는 스팟 플릿의 경우 요청 유형은 `maintain`이어야 합니다.

고려 사항

- Amazon EC2만 중단된 중지된 스팟 인스턴스를 다시 시작할 수 있습니다.
- `persistent` 스팟 인스턴스 요청에 의해 시작된 스팟 인스턴스의 경우 Amazon EC2는 용량이 동일한 가용 영역에서 사용 가능한 경우 중지된 인스턴스와 동일한 인스턴스 유형에 대해 중지된 인스턴스를 다시 시작합니다(동일한 시작 사양을 사용해야 함).
- 스팟 인스턴스가 중지되었을 때 일부 인스턴스 속성을 수정할 수 있지만 인스턴스 유형은 수정할 수 없습니다. EBS 볼륨을 분리하거나 삭제한 경우에는 스팟 인스턴스를 시작해도 연결되지 않습니다. 루트 볼륨을 분리하고 Amazon EC2가 스팟 인스턴스를 시작하려고 하면 인스턴스가 시작되지 않고 Amazon EC2가 중지된 인스턴스를 종료합니다.
- 중지된 상태의 스팟 인스턴스를 종료할 수 있습니다.
- 스팟 인스턴스 요청이나 EC2 플릿 또는 스팟 플릿을 취소하면 Amazon EC2에서는 중지된 상태의 연결된 스팟 인스턴스를 모두 종료합니다.
- 중단된 스팟 인스턴스가 중지 상태인 동안에는 유지 중인 EBS 볼륨에 대한 요금만 부과됩니다. EC2 플릿 및 스팟 플릿을 사용하는 경우 중지된 인스턴스가 많으면 해당 계정의 EBS 볼륨 수 제한을 초과할 수 있습니다. 스팟 인스턴스가 중단될 때 요금이 부과되는 방식에 대한 자세한 내용은 중단된 스팟 인스턴스에 대한 청구 섹션을 참조하세요.
- 인스턴스 중지의 영향을 잘 알고 있어야 합니다. 인스턴스가 중지되면 어떻게 되는지에 대한 자세한 내용은 인스턴스 상태 간 차이 섹션을 참조하세요.

중단된 스팟 인스턴스를 최대 절전 모드로 전환

스팟 인스턴스가 중단되면 Amazon EC2가 스팟 인스턴스를 최대 절전 모드로 전환하도록 지정할 수 있습니다. 자세한 내용은 Amazon EC2 인스턴스를 최대 절전 모드로 전환 섹션을 참조하세요.

Amazon EC2에서는 이제 현재 온디맨드 인스턴스에서 사용할 수 있는 최대 절전 모드 환경이 스팟 인스턴스에 제공됩니다. 이제 스팟 인스턴스 최대 절전 모드에 대해 다음이 지원되는 더 광범위한 지원이 제공됩니다.

- 더 지원되는 AMI
- 더 지원되는 인스턴스 패밀리
- 사용자가 시작한 최대 절전 모드

중단된 스팟 인스턴스 종료

Amazon EC2는 스팟 인스턴스를 중단할 때 중지, 최대 절전 모드로 전환 등의 다른 중단 동작을 지정하지 않는 한 기본적으로 인스턴스를 종료합니다. 자세한 내용은 Amazon EC2 인스턴스 종료 섹션을 참조하세요.

스팟 인스턴스 중단 준비

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.

스팟 인스턴스 중단에 대비할 수 있도록 다음 모범 사례를 따르는 것이 좋습니다.

- Auto Scaling 그룹을 사용하여 스팟 요청을 생성합니다. 스팟 인스턴스가 중단되면 Auto Scaling 그룹이 대체 인스턴스를 자동으로 시작합니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 여러 인스턴스 유형과 구매 옵션을 제공하는 Auto Scaling 그룹 섹션을 참조하세요.
- 필수 소프트웨어 구성이 포함된 Amazon Machine Image(AMI)를 사용하여 요청이 이행되는 즉시 인스턴스를 실행할 준비가 되었는지 확인합니다. 시작 시 사용자 데이터를 사용하여 명령을 실행할 수도 있습니다.
- 인스턴스가 중지되거나 종료되면 인스턴스 스토어 볼륨의 데이터가 손실됩니다. 인스턴스 스토어 볼륨의 중요 데이터는 Amazon S3, Amazon EBS 또는 Amazon DynamoDB 등의 보다 영구적인 스토리지에 백업합니다.
- 스팟 인스턴스가 종료되어도 영향을 받지 않을 장소에 중요한 데이터를 정기적으로 저장합니다. 예를 들어, Amazon S3, Amazon EBS 또는 DynamoDB를 사용할 수 있습니다.
- 작업을 작은 부분으로 분리하거나(눈금, Hadoop 또는 대기열 기반 아키텍처 사용), 작업을 자주 저장할 수 있도록 검사점을 사용합니다.
- Amazon EC2는 인스턴스 중단 위험이 높아질 때 스팟 인스턴스에 리밸런싱 권고 신호를 전송합니다. 리밸런싱 권고를 활용하면 스팟 인스턴스 중단 2분 전 공지를 기다릴 필요 없이 스팟 인스턴스 중단을 사전 예방적으로 관리할 수 있습니다. 자세한 내용은 EC2 인스턴스 리밸런싱 권고 섹션을 참조하세요.
- 스팟 인스턴스 중단 2분 전 공지를 사용하여 스팟 인스턴스의 상태를 모니터링합니다. 자세한 내용은 스팟 인스턴스 중단 공지 섹션을 참조하세요.
- 이러한 경고를 즉시 제공하기 위해 모든 노력을 기울이고 있지만 경고를 보내기 전에 스팟 인스턴스가 중단될 수도 있습니다. 따라서 리밸런싱 권고 신호 및 중단 공지를 모니터링하는 경우에도 애플리케이션을 테스트하여 예기치 않은 인스턴스 중단이 정상적으로 처리되는지 확인해야 합니다. 이

렇게 하려면 온디맨드 인스턴스를 사용하여 애플리케이션을 실행한 다음 온디맨드 인스턴스를 직접 종료합니다.

- AWS Fault Injection Service로 제어된 결함 주입 실험을 실행하여 스팟 인스턴스가 중단될 때 애플리케이션이 어떻게 응답하는지 테스트합니다. 자세한 내용은 AWS FIS 사용 설명서의 자습서: AWS Fault Injection Service를 사용한 스팟 인스턴스 중단 테스트를 참조하세요.

스팟 인스턴스 중단 시작

Amazon EC2 콘솔에서 스팟 인스턴스 요청 또는 스팟 플릿 요청을 선택하고 스팟 인스턴스 중단을 시작하여 스팟 인스턴스의 애플리케이션이 중단되는 것을 어떻게 처리하는지 테스트할 수 있습니다. 스팟 인스턴스 중단을 시작하면 Amazon EC2에서 스팟 인스턴스가 2분 후에 중단될 것임을 알리고 2분 후에 스팟 인스턴스가 중단됩니다.

스팟 인스턴스 중단을 수행하는 기본 서비스는 AWS Fault Injection Service(AWS FIS)입니다. AWS FIS에 대한 자세한 내용은 AWS Fault Injection Service 섹션을 참조하세요.

> **Note**
>
> 중단 동작은 `terminate`, `stop`, 및 `hibernate`입니다. 중단 동작을 `hibernate`로 설정한 경우 스팟 인스턴스 중단을 시작하면 최대 절전 모드 프로세스가 즉시 시작됩니다.

스팟 인스턴스 중단 시작은 아시아 태평양(자카르타), 아시아 태평양(오사카), 중국(베이징), 중국(닝샤) 및 중동(UAE)을 제외한 모든 AWS 리전에서 지원됩니다.

내용

- 스팟 인스턴스 중단 시작
- 스팟 인스턴스 중단 확인
- 할당량

스팟 인스턴스 중단 시작

EC2 콘솔을 사용하여 스팟 인스턴스 중단을 빠르게 시작할 수 있습니다. 스팟 인스턴스 요청을 선택하면 하나의 스팟 인스턴스의 중단을 시작할 수 있습니다. 스팟 플릿 요청을 선택하면 여러 스팟 인스턴스의 중단을 한 번에 시작할 수 있습니다.

스팟 인스턴스 중단을 테스트하기 위한 고급 실험의 경우 AWS FIS 콘솔을 사용하여 고유한 실험을 생성할 수 있습니다.

EC2 콘솔을 사용하여 스팟 인스턴스 요청에서 하나의 인스턴스 중단을 시작하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Spot Requests(스팟 요청)를 선택합니다.
3. 스팟 인스턴스 요청을 선택한 다음 Actions(작업), Initiate interruption(중단 시작)을 선택합니다. 여러 스팟 인스턴스 요청을 선택하여 중단을 시작할 수는 없습니다.
4. Initiate Spot Instance interruption(스팟 인스턴스 중단 시작) 대화 상자의 Service access(서비스 액세스)에서 기본 역할을 사용하거나 기존 역할을 선택합니다. 기존 역할을 선택하려면 기존 서비스 역할 사용을 선택한 다음 IAM 역할에서 사용할 역할을 선택합니다.
5. 스팟 인스턴스 중단을 시작할 준비가 되면 Initiate interruption(중단 시작)을 선택합니다.

EC2 콘솔을 사용하여 스팟 플릿 요청에서 하나 이상의 인스턴스 중단을 시작하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Spot Requests(스팟 요청)를 선택합니다.
3. 스팟 플릿 요청을 선택한 다음 작업, 중단 시작을 선택합니다. 여러 스팟 플릿 요청을 선택하여 중단을 시작할 수는 없습니다.
4. 스팟 인스턴스 수 지정 대화 상자에서 중단할 인스턴스 수에 중단할 스팟 인스턴스 수를 입력한 다음 확인을 선택합니다.

> **Note**
>
> 이 수는 플릿의 스팟 인스턴스 수 또는 AWS FIS가 실험당 중단할 수 있는 스팟 인스턴스 수에 대한 할당량을 초과할 수 없습니다.

5. Initiate Spot Instance interruption(스팟 인스턴스 중단 시작) 대화 상자의 Service access(서비스 액세스)에서 기본 역할을 사용하거나 기존 역할을 선택합니다. 기존 역할을 선택하려면 기존 서비스 역할 사용을 선택한 다음 IAM 역할에서 사용할 역할을 선택합니다.
6. 스팟 인스턴스 중단을 시작할 준비가 되면 Initiate interruption(중단 시작)을 선택합니다.

AWS FIS 콘솔을 사용하여 스팟 인스턴스 중단 테스트를 위한 고급 실험 생성

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Spot Requests(스팟 요청)를 선택합니다.
3. Actions(작업), Create advanced experiments(고급 실험 생성)를 선택합니다.

AWS FIS 콘솔이 열립니다. 자세한 내용은 AWS Fault Injection Service 사용 설명서의 자습서: AWS FIS를 사용한 스팟 인스턴스 중단 테스트를 참조하세요.

스팟 인스턴스 중단 확인

중단을 시작하면 다음과 같이 진행됩니다.

- 스팟 인스턴스가 인스턴스 리밸런싱 권고를 수신합니다.
- AWS FIS에서 스팟 인스턴스를 중단하기 2분 전에 스팟 인스턴스 중단 공지가 생성됩니다.
- 2분 후 스팟 인스턴스가 중단됩니다.
- AWS FIS에 의해 중지된 스팟 인스턴스는 다시 시작할 때까지 중지된 상태로 유지됩니다.

중단을 시작한 후 인스턴스가 중단되었는지 확인

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 별도의 브라우저 탭 또는 창으로 Spot Requests(스팟 요청)와 Instances(인스턴스)를 엽니다.
3. 스팟 요청에서 스팟 인스턴스 요청 또는 스팟 플릿 요청을 선택합니다. 초기 상태는 `fulfilled`입니다. 인스턴스가 중단된 후 중단 동작에 따라 상태가 다음과 같이 변경됩니다.
   - `terminate` - 상태가 `instance-terminated-by-experiment`로 변경됩니다.
   - `stop` - 상태가 `marked-for-stop-by-experiment`으로 변경되었다가 `instance-stopped-by-experiment`로 변경됩니다.
4. Instances(인스턴스)에서 스팟 인스턴스를 선택합니다. 초기 상태는 `Running`입니다. 스팟 인스턴스 중단 알림을 받고 2분 후 중단 동작에 따라 상태가 다음과 같이 변경됩니다.
   - `stop` - 상태가 `Stopping`으로 변경되었다가 `Stopped`로 변경됩니다.
   - `terminate` - 상태가 `Shutting-down`으로 변경되었다가 `Terminated`로 변경됩니다.

할당량

AWS 계정에는 AWS FIS가 실험당 중단할 수 있는 스팟 인스턴스 수에 대한 기본 할당량은 다음과 같습니다.

명칭	기본값	조정 가능	설명
		예	

명칭	기본값	조정 가능	설명
aws:ec2:send-spot-instance-interruptions에 대한 대상 스팟 인스턴스	지원되는 각 리전: 5개		실험당 태그를 사용하여 대상을 식별할 때 aws:ec2:send-spot-instance-interruptions가 대상으로 지정할 수 있는 스팟 인스턴스의 최대 수입니다.

할당량 증가를 요청할 수 있습니다. 자세한 내용은 Service Quotas 사용 설명서의 할당량 증가 요청을 참조하세요.

AWS FIS에 대한 할당량을 보려면 Service Quotas 콘솔을 엽니다. 탐색 창에서 AWS 서비스를 선택하고 AWS Fault Injection Service을 선택합니다. 또한 AWS Fault Injection Service 사용 설명서에서 모든 AWS Fault Injection Service에 대한 할당량을 볼 수 있습니다.

스팟 인스턴스 중단 공지

스팟 인스턴스 중단 공지는 Amazon EC2가 스팟 인스턴스를 중지 또는 종료하기 2분 전에 생성되는 경고입니다. 최대 절전을 중지 행동으로 지정할 경우 중지 공지를 수신하지만, 최대 절전 과정은 즉시 시작되므로 2분 경고를 받지 않습니다.

스팟 인스턴스 중단을 정상적으로 처리하는 가장 좋은 방법은 내결함성이 있도록 애플리케이션을 설계하는 것입니다. 이를 위해 스팟 인스턴스 중단 공지를 활용할 수 있습니다. 5초마다 이러한 중단 공지를 확인하는 것이 좋습니다.

중단 공지는 EventBridge 이벤트 및 스팟 인스턴스의 인스턴스 메타데이터 항목으로 제공됩니다. 중단 알림은 최선의 노력을 한 후 발송됩니다.

EC2 Spot Instance Interruption Warning 이벤트를 트리거합니다

Amazon EC2에서는 스팟 인스턴스를 중단할 때 실제 중단 2분 전에 이벤트를 전송합니다(최대 절전 모드는 즉시 시작하기 때문에 중단 공지를 2분 전에 받지 않는 최대 절전 모드는 제외). 이 이벤트는 Amazon EventBridge에서 감지할 수 있습니다. EventBridge 이벤트에 대한 자세한 내용은 Amazon EventBridge User Guidehttps://docs.aws.amazon.com/eventbridge/latest/userguide/를 참조하세요. 이벤트 규칙을 생성하고 사용하는 방법을 안내하는 자세한 예제는 Amazon EC2 스팟 인스턴스 중단 공지 활용을 참조하세요.

다음은 스팟 인스턴스 중단 이벤트의 예제입니다. 가능한 instance-action 값은 hibernate, stop 또는 terminate입니다.

```
{
    "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-2a:instance/i-1234567890abcdef0"],
    "detail": {
        "instance-id": "i-1234567890abcdef0",
        "instance-action": "action"
    }
}
```

> **Note**
>
> 스팟 인스턴스 중단 이벤트의 ARN 형식은 arn:aws:ec2:*availability-zone*:instance/*instance-id*입니다. 이 형식은 EC2 리소스 ARN 형식과 다릅니다.

instance-action

instance-action 항목은 해당 작업과 작업이 이루어지는 대략적 시간(UTC 기준)을 지정합니다.

스팟 인스턴스가 Amazon EC2에 의해 중지되거나 종료되도록 표시된 경우 인스턴스 메타데이터에 instance-action 항목이 있습니다. 표시하지 않은 경우에는 이 항목이 없습니다. 인스턴스 메타데이터 서비스 버전 2(IMDSv2)를 사용하여 다음과 같이 instance-action을 검색할 수 있습니다.

Linux

```
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" http://169.254.169.254/latest/meta-data/spot/instance-action
```

Windows

```
[string]$token = Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/meta-data/spot/instance-action
```

다음 예시 출력에서는 이 인스턴스가 중지될 시간을 알려줍니다.

```
{"action": "stop", "time": "2017-09-18T08:22:00Z"}
```

다음 예시 출력에서는 이 인스턴스가 종료될 시간을 알려줍니다.

```
{"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 인스턴스 메타데이터에 없고 사용자가 이를 검색하려 하면 HTTP 404 오류를 수신하게 됩니다.

termination-time

`termination-time` 항목은 인스턴스가 종료 신호를 수신할 때 대략적인 시간(UTC 기준)을 지정합니다.

> **Note**
>
> 이 항목은 이전 버전과의 호환성을 위해 보존되며, 그 대신 `instance-action`을 사용해야 합니다.

Amazon EC2에서 스팟 인스턴스를 종료 대상으로 표시한 경우(중단 동작이 `terminate`로 설정된 스팟 인스턴스 중단 또는 영구 스팟 인스턴스 요청 취소로 인해) `termination-time` 항목이 인스턴스 메타데이터에 존재합니다. 표시하지 않은 경우에는 이 항목이 없습니다. 다음과 같이 IMDSv2를 사용하여 `termination-time`을 검색할 수 있습니다.

Linux

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-
token-ttl-seconds: 21600"`
```

```
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 termination_scheduled; fi
```

Windows

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/meta-data/spot/termination-time
```

출력의 예시는 다음과 같습니다.

```
2015-01-05T18:02:00Z
```

스팟 인스턴스 중단이 없거나 중단 동작이 `stop` 또는 `hibernate`로 설정되었기 때문에 Amazon EC2가 인스턴스를 종료할 준비가 되지 않은 경우 또는 사용자가 스팟 인스턴스를 직접 종료한 경우 `termination-time` 항목이 인스턴스 메타데이터에 없거나(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 인스턴스 찾기

EC2 인스턴스를 설명할 때 결과에는 스팟 인스턴스가 포함됩니다. 스팟 인스턴스의 인스턴스 수명 주기는 `spot`입니다. 스팟 인스턴스의 인스턴스 상태는 구성한 중단 동작에 따라 `stopped` 또는 `terminated` 중 하나입니다. 최대 절전 모드로 전환된 스팟 인스턴스의 인스턴스 상태는 `stopped`입니다.

중단 이유에 대한 자세한 내용은 스팟 요청 상태 코드를 확인하세요. 자세한 내용은 the section called “스팟 인스턴스 요청 상태 가져오기” 섹션을 참조하세요.

Console

중단된 스팟 인스턴스를 찾는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 필터 인스턴스 수명 주기=스팟을 적용합니다.

4. 구성한 중단 동작에 따라 인스턴스 상태=중지됨 또는 인스턴스 상태=종료됨 필터를 적용합니다.
5. 각 스팟 인스턴스에 대해 세부 정보 탭의 인스턴스 세부 정보에서 상태 전환 메시지를 찾습니다. 다음 코드는 스팟 인스턴스가 중단되었음을 가리킵니다.

   - `Server.SpotInstanceShutdown`
   - `Server.SpotInstanceTermination`

AWS CLI

중단된 스팟 인스턴스를 찾는 방법

`--filters` 옵션과 함께 describe-instances 명령을 사용합니다. 출력에 인스턴스 ID만 나열하려면 `--query` 옵션을 포함합니다.

중단 동작이 스팟 인스턴스 종료인 경우 다음 예제를 사용합니다.

```
aws ec2 describe-instances \
    --filters Name=instance-lifecycle,Values=spot \
              Name=instance-state-name,Values=terminated \
              Name=state-reason-code,Values=Server.SpotInstanceTermination \
    --query "Reservations[*].Instances[*].InstanceId"
```

중단 동작이 스팟 인스턴스 중지인 경우 다음 예제를 사용합니다.

```
aws ec2 describe-instances \
    --filters Name=instance-lifecycle,Values=spot \
              Name=instance-state-name,Values=stopped \
              Name=state-reason-code,Values=Server.SpotInstanceShutdown \
    --query "Reservations[*].Instances[*].InstanceId"
```

PowerShell

중단된 스팟 인스턴스를 찾는 방법

Get-EC2Instance cmdlet을 사용합니다.

중단 동작이 스팟 인스턴스 종료인 경우 다음 예제를 사용합니다.

```
(Get-EC2Instance `
    -Filter @{Key="instance-lifecycle"; Values="spot"} `
```

```
            @{Key="instance-state-name"; Values="terminated"} `
            @{Key="state-reason-code";
 Values="Server.SpotInstanceTermination"}).Instances.InstanceId
```

중단 동작이 스팟 인스턴스 중지인 경우 다음 예제를 사용합니다.

```
(Get-EC2Instance `
    -Filter @{Key="instance-lifecycle"; Values="spot"} `
            @{Key="instance-state-name"; Values="stopped"} `
            @{Key="state-reason-code";
 Values="Server.SpotInstanceTermination"}).Instances.InstanceId
```

Amazon EC2가 스팟 인스턴스를 종료했는지 확인

스팟 인스턴스는 스팟 인스턴스 중단에 대한 응답으로 Amazon EC2가 종료할 때까지 또는 사용자가 직접 종료할 때까지 실행됩니다. 자세한 내용은 the section called “중단 동작” 섹션을 참조하세요.

스팟 인스턴스가 종료된 경우 AWS CloudTrail을 사용하여 Amazon EC2가 스팟 인스턴스를 종료했는지 확인할 수 있습니다. CloudTrail 로그에 `BidEvictedEvent`가 포함되어 있으면 Amazon EC2가 스팟 인스턴스를 종료한 것입니다. 대신 `TerminateInstances` 이벤트가 표시되면 사용자가 스팟 인스턴스를 종료한 것입니다.

또는 Amazon EC2가 스팟 인스턴스를 중단할 것이라는 알림을 받으려면 Amazon EventBridge를 사용하여 EC2 스팟 인스턴스 중단 경고 이벤트에 응답합니다.

CloudTrail에서 BidEvictedEvent 이벤트를 보려면 다음을 수행하세요.

1. https://console.aws.amazon.com/cloudwatch/에서 CloudTrail 콘솔을 엽니다.
2. 탐색 창에서 Event history(이벤트 내역)를 선택합니다.
3. 필터 목록에서 이벤트 이름을 선택한 다음 오른쪽의 필터 필드에 **`BidEvictedEvent`**를 입력합니다.
4. (선택 사항) 시간 범위를 선택합니다.
5. 목록이 비어 있지 않은 경우 결과 항목에서 BidEvictedEvent를 선택하여 세부 정보 페이지를 엽니다. 이벤트 레코드 창에서 스팟 인스턴스의 ID를 포함하여 스팟 인스턴스에 대한 정보를 찾을 수 있습니다. 다음은 이벤트 레코드의 예입니다.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
```

```
        "accountId": "123456789012",
        "invokedBy": "ec2.amazonaws.com"
    },
    "eventTime": "2016-08-16T22:30:00Z",
    "eventSource": "ec2.amazonaws.com",
    "userAgent": "ec2.amazonaws.com",
    "sourceIPAddress": "ec2.amazonaws.com",
    "eventName": "BidEvictedEvent",
    "awsRegion": "us-east-2",
    "eventID": "d27a6096-807b-4bd0-8c20-a33a83375054",
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "123456789012",
    "RequestParameters": null,
    "ResponseElements": null,
    "serviceEventDetails": {
        "instanceIdSet": [
          "i-1eb2ac8eEXAMPLE"
        ]
    }
}
```


6. `BidEvictedEvent` 이벤트 항목을 찾지 못한 경우 이벤트 이름으로 **`TerminateInstances`**를 입력합니다. `TerminateInstances` 이벤트 기록에 대한 자세한 내용은 the section called “Amazon EC2 API 이벤트 예제” 섹션을 참조하세요.

중단된 스팟 인스턴스에 대한 청구

스팟 인스턴스가 중단되면 다음과 같이 인스턴스 및 EBS 볼륨 사용량에 대해 요금이 부과되고 다른 요금도 청구될 수 있습니다.

인스턴스 사용

스팟 인스턴스를 중단한 엔터티	운영 체제	첫 번째 시간에 중단됨	첫 번째 시간 후 어떤 시간에든 중단됨
사용자가 스팟 인스턴스를 중지하거나 종료하는 경우	Windows 및 Linux(SUSE 제외)	사용된 시간(초)에 대해 부과	사용된 시간(초)에 대해 부과
	SUSE	일부 시간만 사용한 경우에도 전체 시간에 대해 부과	사용한 전체 시간에 대해 부과되며, 중단된

스팟 인스턴스를 중단한 엔터티	운영 체제	첫 번째 시간에 중단됨	첫 번째 시간 후 어떤 시간에든 중단됨
			일부 시간에 대해 전체 시간 요금 부과
Amazon EC2에 의해 스팟 인스턴스가 중단되는 경우	Windows 및 Linux(SUSE 제외)	무료	사용된 시간(초)에 대해 부과
	SUSE	무료	사용한 전체 시간에 대해 부과되지만, 중단된 일부 시간에 대한 요금은 무료

EBS 볼륨 사용량

중단된 스팟 인스턴스가 중지 상태인 동안에는 유지 중인 EBS 볼륨에 대한 요금만 부과됩니다.

EC2 플릿 및 스팟 플릿을 사용하는 경우 중지된 인스턴스가 많으면 해당 계정의 EBS 볼륨 수 제한을 초과할 수 있습니다.

EC2 인스턴스 리밸런싱 권고

EC2 인스턴스 리밸런싱 권고(rebalance recommendation)는 스팟 인스턴스의 중단 위험이 높아질 때 알림을 보내는 신호입니다. 이 신호는 스팟 인스턴스 중단 2분 전 공지보다 일찍 도착할 수 있으므로 스팟 인스턴스를 사전에 관리할 수 있는 기회를 제공합니다. 중단 위험이 높아지지 않는 신규 또는 기존 스팟 인스턴스에 대한 워크로드를 리밸런싱하도록 결정할 수 있습니다.

Amazon EC2에서 항상 스팟 인스턴스 중단 2분 전 공지보다 먼저 리밸런싱 권고 신호를 전송할 수 있는 것은 아닙니다. 따라서 리밸런싱 권고 신호가 중단 2분 전 공지와 함께 도착할 수도 있습니다.

재분배 권장 사항은 EventBridge 이벤트 및 스팟 인스턴스의 인스턴스 메타데이터 항목으로 제공됩니다. 이벤트는 최선의 작업을 기반으로 발생됩니다.

> **Note**
>
> 리밸런싱 권고는 2020년 11월 5일 00:00 UTC 이후에 시작된 스팟 인스턴스에만 지원됩니다.

내용

- 수행할 수 있는 리밸런싱 작업
- 리밸런싱 권고 신호 모니터링
- 리밸런싱 권고 신호를 사용하는 서비스

수행할 수 있는 리밸런싱 작업

다음은 수행할 수 있는 리밸런싱 작업의 일부입니다.

정상 종료

스팟 인스턴스에 대한 리밸런싱 권고 신호를 수신하면 인스턴스 종료 절차를 시작할 수 있습니다. 이러한 절차에는 중지 전에 프로세스를 완료하는 작업 등이 포함될 수 있습니다. 예를 들어 시스템 또는 애플리케이션 로그를 Amazon Simple Storage Service(Amazon S3)에 업로드하거나 Amazon SQS 작업자를 종료하거나 Domain Name System(DNS)의 등록 취소를 완료할 수 있습니다. 외부 스토리지에 작업을 저장하고 나중에 다시 시작할 수도 있습니다.

새 작업 예약 차단

스팟 인스턴스에 대한 리밸런싱 권고 신호가 수신되면 예약된 작업이 완료될 때까지 인스턴스를 계속 사용하면서 인스턴스에 새 작업이 예약되는 것을 차단할 수 있습니다.

새로운 대체 인스턴스를 사전 예방적으로 시작

리밸런싱 권고 신호가 생성될 때 대체 스팟 인스턴스를 자동으로 시작하도록 Auto Scaling 그룹, EC2 플릿 또는 스팟 플릿을 구성할 수 있습니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 용량 리밸런싱을 사용하여 Amazon EC2 스팟 중단 처리와 본 사용 설명서에서 EC2 플릿 및 스팟 플릿의 용량 리밸런싱을 사용하여 위험이 있는 스팟 인스턴스 교체의 내용을 참조하세요.

리밸런싱 권고 신호 모니터링

리밸런싱 권고 신호를 모니터링하여 신호가 생성될 때 이전 섹션에서 지정한 작업을 수행할 수 있습니다. 리밸런싱 권고 신호는 Amazon EventBridge(이전의 Amazon CloudWatch Events)로 전송되는 이벤트와 스팟 인스턴스의 인스턴스 메타데이터로 제공됩니다.

리밸런싱 권고 신호 모니터링:

- Amazon EventBridge 사용
- 인스턴스 메타데이터 사용

Amazon EventBridge 사용

스팟 인스턴스에 대한 리밸런싱 권고 신호가 생성되면 신호에 대한 이벤트가 Amazon EventBridge로 전송됩니다. EventBridge에서 규칙에 정의된 패턴과 일치하는 이벤트 패턴이 감지되는 경우 EventBridge는 규칙에 정의된 대상을 호출합니다.

다음은 리밸런싱 권고 신호에 대한 예제 이벤트입니다.

```
{
    "version": "0",
    "id": "12345678-1234-1234-1234-123456789012",
    "detail-type": "EC2 Instance Rebalance Recommendation",
    "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"
    }
}
```

다음 필드는 규칙에 정의되는 이벤트 패턴을 형성합니다.

`"detail-type": "EC2 Instance Rebalance Recommendation"`

리밸런싱 권고 이벤트를 식별합니다.

`"source": "aws.ec2"`

Amazon EC2에서 시작된 이벤트를 식별합니다.

EventBridge 규칙 생성

EventBridge 규칙을 작성하고 이벤트 패턴이 규칙과 일치할 때 수행할 작업을 자동화할 수 있습니다.

다음 예제에서는 Amazon EC2에서 리밸런싱 권고 신호가 생성될 때마다 이메일, 문자 메시지 또는 모바일 푸시 알림을 보내는 EventBridge 규칙을 생성합니다. 이 신호는 규칙에 정의된 작업을 트리거하는 `EC2 Instance Rebalance Recommendation` 이벤트로 생성됩니다.

EventBridge 규칙을 생성하기 전에 이메일, 문자 메시지 또는 모바일 푸시 알림에 대한 Amazon SNS 주제를 생성해야 합니다.

리밸런싱 권고 이벤트에 대한 EventBridge 규칙을 생성하려면

1. https://console.aws.amazon.com/events/에서 Amazon EventBridge 콘솔을 엽니다.
2. [규칙 생성(Create rule)]을 선택합니다.
3. 규칙 세부 정보 정의(Define rule detail)에 대해 다음을 수행하십시오:
   a. 규칙의 이름을 입력하고 선택적으로 설명을 입력합니다.

      규칙은 동일한 지역과 동일한 이벤트 버스의 다른 규칙과 동일한 이름을 가질 수 없습니다.

   b. 이벤트 버스(Event bus)에서 기본값(default)을 선택합니다. 계정의 AWS 서비스가 이벤트를 생성하면 항상 계정의 기본 이벤트 버스로 이동합니다.
   c. 규칙 유형(Rule type)에서 이벤트 패턴이 있는 규칙(Rule with an event pattern)을 생성합니다.
   d. 다음을 선택합니다.
4. 이벤트 패턴 빌드(Build event pattern)에서 다음을 수행하십시오:
   a. 이벤트 소스(Event source)에서 AWS 이벤트 또는 EventBridge 파트너 이벤트(events or EventBridge partner events)를 선택합니다.
   b. Event pattern(이벤트 패턴)에 `EC2 Instance Rebalance Recommendation` 이벤트와 일치하도록 다음 이벤트 패턴을 지정하고 Save(저장)를 선택합니다.

      ```
      {
       "source": ["aws.ec2"],
       "detail-type": ["EC2 Instance Rebalance Recommendation"]
      }
      ```

      이벤트 패턴을 추가하려면 다음과 같이 이벤트 패턴 양식(Event pattern form)을 선택하여 템플릿을 사용하거나 사용자 정의 패턴(JSON 편집기)(Custom pattern (JSON editor))을 선택하여 고유한 패턴을 지정할 수 있습니다.

      i. 템플릿을 사용하여 이벤트 패턴을 생성하려면 다음을 수행하세요.

         A. 이벤트 패턴 양식(Event pattern form)을 선택합니다.
         B. 이벤트 소스(Event source)에서 AWS 서비스(services)를 선택합니다.
         C. AWS 서비스(Service)에서 EC2 스팟 플릿(EC2 Spot Fleet)을 선택합니다.
         D. Event type(이벤트 유형)에서 EC2 Instance Rebalance Recommendation(EC2 인스턴스 재조정 권장 사항)을 선택합니다.

E. 템플릿을 사용자 지정하려면 패턴 편집(Edit pattern)을 선택하고 예시 이벤트 패턴과 일치하도록 변경합니다.

ii. (대안) 사용자 정의 이벤트 패턴을 지정하려면 다음을 수행하세요.

A. 사용자 정의 패턴(JSON 편집기)을 선택합니다.

B. 이벤트 패턴(Event pattern) 상자에서 이 예시의 이벤트 패턴을 추가합니다.

c. 다음을 선택합니다.

5. 대상 선택(Select target(s))에서 다음을 수행합니다.

a. 대상 유형(Target types)에서 AWS 서비스(service)를 선택합니다.

b. 대상 선택(Select a target)에서 SNS 주제(SNS topic)를 선택하여 이벤트가 발생할 때 이메일, 문자 메시지 또는 모바일 푸시 알림을 보내도록 합니다.

c. [주제(Topic)]에서 기존 주제를 선택합니다. 먼저 Amazon SNS 콘솔을 사용하여 Amazon SNS 주제를 생성해야 합니다. 자세한 내용은 Amazon Simple Notification Service 개발자 안내서에서 A2P(Application-to-Person) 메시징에 Amazon SNS 사용을 참조하세요.

d. (선택 사항) 추가 설정(Additional settings)에서 선택적으로 추가 설정을 구성할 수 있습니다. 자세한 설명은 Amazon EventBridge 사용자 가이드의 Creating Amazon EventBridge rules that react to events(이벤트에 응답하는 Amazon EventBridge 규칙 생성)(16단계)를 참조하세요.

e. 다음을 선택합니다.

6. (선택 사항) 태그(Tags)에서 선택적으로 하나 이상의 태그를 규칙에 할당하고 다음(Next)을 선택할 수 있습니다.

7. 검토 및 생성(Review and create)에서 다음을 수행합니다.

a. 규칙의 세부 정보를 검토하고 필요에 따라 수정합니다.

b. 규칙 생성을 선택합니다.

자세한 내용은 Amazon EventBridge 사용 설명서에서 Amazon EventBridge 규칙 및 Amazon EventBridge 이벤트 패턴을 참조하세요.

인스턴스 메타데이터 사용

인스턴스 메타데이터 범주 `events/recommendations/rebalance`는 스팟 인스턴스에 대한 리밸런싱 권고 신호가 생성되는 대략적인 시간(UTC)을 제공합니다.

리밸런싱 권고에 따라 조치를 취할 기회를 놓치지 않도록 5초마다 리밸런싱 권고 신호를 확인하는 것이 좋습니다.

스팟 인스턴스에서 리밸런싱 권고를 수신하는 경우 신호가 생성된 시간이 인스턴스 메타데이터에 나타납니다. 다음과 같이 신호가 생성된 시간을 검색할 수 있습니다.

IMDSv2

Linux

Linux 인스턴스에서 다음 명령을 실행합니다.

IMDSv2

```
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" http://169.254.169.254/latest/meta-data/events/recommendations/rebalance
```

Windows

Windows 인스턴스에서 다음 cmdlet을 실행합니다.

```
[string]$token = Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/events/recommendations/rebalance
```

IMDSv1

Linux

Linux 인스턴스에서 다음 명령을 실행합니다.

```
curl http://169.254.169.254/latest/meta-data/events/recommendations/rebalance
```

Windows

Windows 인스턴스에서 다음 cmdlet을 실행합니다.

```
Invoke-RestMethod -Uri http://169.254.169.254/latest/meta-data/events/
recommendations/rebalance
```

다음은 스팟 인스턴스에 대한 리밸런싱 권고 신호가 생성된 시간(UTC)을 나타내는 예시 출력입니다.

```
{"noticeTime": "2020-10-27T08:22:00Z"}
```

인스턴스에 대한 신호가 생성되지 않은 경우 `events/recommendations/rebalance`가 표시되지 않으며 이를 검색하려고 하면 HTTP 404 오류가 발생합니다.

리밸런싱 권고 신호를 사용하는 서비스

Amazon EC2 Auto Scaling, EC2 플릿 및 스팟 플릿에서는 리밸런싱 권고 신호를 사용하여 실행 중인 인스턴스에 스팟 인스턴스 중단 2분 전 공지가 수신되기 전에 새 스팟 인스턴스로 플릿을 미리 보강하여 워크로드 가용성을 유지할 수 있습니다. 이러한 서비스에서는 스팟 인스턴스의 가용성에 영향을 미치는 변경 사항을 모니터링하여 사전 예방적으로 대응할 수 있습니다. 자세한 내용은 다음을 참조하세요.

- Amazon EC2 Auto Scaling User Guide의 Use Capacity Rebalancing to handle Amazon EC2 Spot interruptions
- 이 사용 설명서의 EC2 플릿 및 스팟 플릿 주제의 EC2 플릿 및 스팟 플릿의 용량 리밸런싱을 사용하여 위험이 있는 스팟 인스턴스 교체

스팟 배치 점수

스팟 배치 점수 기능은 스팟 용량 요구 사항에 따라 AWS 리전 또는 가용 영역을 추천할 수 있습니다. 스팟 용량은 변동하기 때문에 항상 필요한 용량을 확보할 수 있을지 확신할 수 없습니다. 스팟 배치 점수는 스팟 요청이 리전 또는 가용 영역에서 성공할 가능성을 나타냅니다.

> **Note**
>
> 스팟 배치 점수는 사용 가능한 용량이나 중단 위험 측면에서 어떠한 보장도 제공하지 않습니다. 스팟 배치 점수는 권장 사항으로만 사용됩니다.

사용 사례

다음과 같은 용도에 스팟 배치 점수 기능을 사용할 수 있습니다.

- 용량 요구 증가 또는 현재 리전의 가용 용량 감소에 대응하여 필요에 따라 다른 리전에서 스팟 컴퓨팅 용량 재배치 및 크기 조정
- 단일 가용 영역 워크로드를 실행할 최적의 가용 영역 식별
- 스팟 기반 워크로드 확장을 위한 최적의 리전을 선택할 수 있도록 향후 스팟 용량 요구 사항 시뮬레이션
- 스팟 용량 요구 사항을 충족하기 위한 최적의 인스턴스 유형 조합 확인

내용

- 제한 사항
- 비용
- 스팟 배치 점수의 작동 방식
- 스팟 배치 점수에 필요한 권한
- 스팟 배치 점수 계산

제한 사항

- 목표 용량 제한 - 스팟 배치 점수 목표 용량 제한은 최근 스팟 사용량을 기준으로 하며 잠재적인 사용량 증가를 고려합니다. 최근 스팟 사용량이 없는 경우 스팟 요청 제한에 맞춰 낮은 기본 제한을 제공합니다.
- 요청 구성 제한 - 스팟 배치 점수 기능의 의도된 용도와 관련이 없는 패턴을 감지하는 경우 24시간 내에 새로운 요청 구성의 수를 제한할 수 있습니다. 한도에 도달하면 이미 사용한 요청 구성을 다시 시도할 수 있지만 다음 24시간 기간까지는 새 요청 구성을 지정할 수 없습니다.
- 최소 인스턴스 유형 수 - 인스턴스 유형을 지정하는 경우 최소 세 가지 인스턴스 유형을 지정해야 합니다. 그렇게 하지 않으면 Amazon EC2가 낮은 스팟 배치 점수를 반환합니다. 마찬가지로 인스턴스 속성을 지정하는 경우 3가지 이상의 서로 다른 인스턴스 유형으로 확인되어야 합니다. 인스턴스 유형은 이름이 다른 경우 다른 것으로 간주됩니다. 예를 들어 m5.8xlarge, m5a.8xlarge 및 m5.12xlarge는 모두 다른 것으로 간주됩니다.

비용

스팟 배치 점수 기능 사용에 따른 추가 요금은 없습니다.

스팟 배치 점수의 작동 방식

스팟 배치 점수 기능을 사용할 때 먼저 스팟 인스턴스에 대한 컴퓨팅 요구 사항을 지정하면 Amazon EC2가 스팟 요청이 성공할 가능성이 큰 상위 10개 리전 또는 가용 영역을 반환합니다. 각 리전 또는 가용 영역은 1~10의 척도로 점수가 매겨지며 10은 스팟 요청이 성공할 가능성이 매우 높음을 나타내고 1은 스팟 요청이 성공할 가능성이 없음을 나타냅니다.

스팟 배치 점수 기능을 사용하려면 다음 단계를 따르세요.

- 1단계: 스팟 요구 사항 지정
- 2단계: 스팟 배치 점수 응답 필터링
- 3단계: 권장 사항 검토
- 4단계: 권장 사항 사용

1단계: 스팟 요구 사항 지정

먼저 다음과 같이 원하는 목표 스팟 용량과 컴퓨팅 요구 사항을 지정합니다.

1. 목표 스팟 용량과 목표 용량 단위(선택 사항)를 지정합니다.

   인스턴스 또는 vCPU 수 또는 메모리 양(MiB)을 기준으로 원하는 목표 스팟 용량을 지정할 수 있습니다. vCPU 수 또는 메모리 양으로 목표 용량을 지정하려면 목표 용량 단위를 `vcpu` 또는 `memory-mib`로 지정해야 합니다. 그렇게 하지 않으면 기본적으로 인스턴스 수가 사용됩니다.

   vCPU 수 또는 메모리 양을 기준으로 목표 용량을 지정하면 총 용량을 계산할 때 이 단위를 사용할 수 있습니다. 예를 들어 크기가 다른 인스턴스를 혼합하여 사용하려는 경우 목표 용량을 총 vCPU 수로 지정할 수 있습니다. 그러면 스팟 배치 점수 기능은 vCPU 수를 기준으로 요청의 각 인스턴스 유형을 고려하고 목표 용량을 합산할 때 총 인스턴스 수 대신 총 vCPU 수를 계산합니다.

   예를 들어 총 목표 용량을 vCPU 30개로 지정하고 인스턴스 유형 목록이 c5.xlarge(vCPU 4개), m5.2xlarge(vCPU 8개) 및 r5.large(vCPU 2개)로 구성되어 있다고 가정해 보겠습니다. 총 30개의 vCPU를 얻기 위해 2개의 c5.xlarge(2*4 vCPU), 2개의 m5.2xlarge(2*8 vCPU) 및 3개의 r5.large(3*2 vCPU)를 혼합할 수 있습니다.
2. 인스턴스 유형 또는 인스턴스 속성을 지정합니다.

   사용할 인스턴스 유형을 지정하거나 컴퓨팅 요구 사항에 필요한 인스턴스 속성을 지정한 다음 Amazon EC2에서 이러한 속성을 가진 인스턴스 유형을 식별하도록 할 수 있습니다. 이를 속성 기반 인스턴스 유형 선택이라고 합니다.

동일한 스팟 배치 점수 요청에서 인스턴스 유형과 인스턴스 속성을 모두 지정할 수 없습니다.

인스턴스 유형을 지정하는 경우 최소 3가지의 서로 다른 인스턴스 유형을 지정해야 합니다. 그렇게 하지 않으면 Amazon EC2가 낮은 스팟 배치 점수를 반환합니다. 마찬가지로 인스턴스 속성을 지정하는 경우 3가지 이상의 서로 다른 인스턴스 유형으로 확인되어야 합니다.

스팟 요구 사항을 지정하는 다양한 방법의 예는 구성의 예 섹션을 참조하세요.

2단계: 스팟 배치 점수 응답 필터링

Amazon EC2는 각 리전 또는 가용 영역에 대한 스팟 배치 점수를 계산하고 스팟 요청이 성공할 가능성이 있는 상위 10개 리전 또는 상위 10개 가용 영역을 반환합니다. 기본값은 점수가 매겨진 리전 목록을 반환하는 것입니다. 모든 스팟 용량을 단일 가용 영역으로 시작하려는 경우 점수가 매겨진 가용 영역 목록을 요청하는 것이 유용합니다.

리전 필터를 지정하여 응답에서 반환될 리전의 범위를 좁힐 수 있습니다.

리전 필터와 점수가 매겨진 가용 영역에 대한 요청을 결합할 수 있습니다. 이러한 방식으로 점수가 매겨진 가용 영역은 필터링한 리전으로 제한됩니다. 리전에서 가장 높은 점수의 가용 영역을 찾기 위해 해당 리전만 지정하면 응답에서 해당 리전의 모든 가용 영역에 대한 점수가 매겨진 목록을 반환합니다.

3단계: 권장 사항 검토

각 리전 또는 가용 영역에 대한 스팟 배치 점수는 목표 용량, 인스턴스 유형의 구성, 기록 및 현재 스팟 사용 추세 및 요청 시간을 기준으로 계산됩니다. 스팟 용량은 지속적으로 변동하기 때문에 동일한 스팟 배치 점수 요청이 다른 시간에 계산될 때 다른 점수를 산출할 수 있습니다.

리전 및 가용 영역은 1~10의 척도로 점수가 매겨집니다. 10점은 스팟 요청이 성공할 가능성이 높지만 보장되지는 않음을 나타냅니다. 1점은 스팟 요청이 성공할 가능성이 전혀 없음을 나타냅니다. 다른 리전 또는 가용 영역에 대해 동일한 점수가 반환될 수 있습니다.

낮은 점수가 반환되면 컴퓨팅 요구 사항을 편집하고 점수를 다시 계산할 수 있습니다. 하루 중 다른 시간에 동일한 컴퓨팅 요구 사항에 대한 스팟 배치 점수 권장 사항을 요청할 수도 있습니다.

4단계: 권장 사항 사용

스팟 배치 점수는 스팟 요청이 스팟 배치 점수 구성(목표 용량, 목표 용량 단위, 인스턴스 유형 또는 인스턴스 속성)과 정확히 동일한 구성을 갖고 `capacity-optimized` 할당 전략을 사용하도록 구성된 경우에만 관련이 있습니다. 그렇게 하지 않으면 사용 가능한 스팟 용량을 얻을 가능성이 점수와 일치하지 않습니다.

스팟 배치 점수는 지침으로 사용되며 스팟 요청이 전체 또는 부분적으로 이행됨을 보장하는 점수는 없지만 다음 정보를 사용하여 최상의 결과를 얻을 수 있습니다.

- 동일한 구성 사용 - 스팟 배치 점수는 Auto Scaling 그룹, EC2 플릿 또는 스팟 플릿의 스팟 요청 구성(목표 용량, 목표 용량 단위 및 인스턴스 유형 또는 인스턴스 속성)이 스팟 배치 점수를 얻기 위해 입력한 것과 동일한 경우에만 관련이 있습니다.

  스팟 배치 점수 요청에서 속성 기반 인스턴스 유형 선택을 사용한 경우 속성 기반 인스턴스 유형 선택을 사용하여 Auto Scaling 그룹, EC2 플릿 또는 스팟 플릿을 구성할 수 있습니다. 자세한 내용은 속성 기반 인스턴스 유형 선택을 사용하여 혼합 인스턴스 그룹 생성 및 EC2 플릿 또는 스팟 플릿의 인스턴스 유형 선택에 대한 속성 지정 섹션을 참조하세요.

  > **Note**
  >
  > vCPU 수 또는 메모리 양으로 목표 용량을 지정하고 스팟 배치 점수 구성에서 인스턴스 유형을 지정한 경우 현재 Auto Scaling 그룹, EC2 플릿 또는 스팟 플릿에서 이 구성을 생성할 수 없습니다. 대신 `WeightedCapacity` 파라미터를 사용하여 인스턴스 가중치를 수동으로 설정해야 합니다.

- **`capacity-optimized`** 할당 전략 사용 - 모든 점수는 플릿 요청이 모든 가용 영역(리전 전체 용량 요청) 또는 단일 가용 영역(하나의 가용 영역에서 용량을 요청하는 경우), 스팟 용량 요청이 성공하기 위한 `capacity-optimized` 스팟 할당 전략을 사용하도록 구성될 것이라고 가정합니다. `lowest-price` 등의 다른 할당 전략을 사용하는 경우 사용 가능한 스팟 용량을 얻을 가능성은 점수와 일치하지 않습니다.
- 점수에 따라 즉각 조치 - 스팟 배치 점수 권장 사항은 요청 시 사용 가능한 스팟 용량을 반영하며 동일하게 구성해도 스팟 용량 변동으로 인해 서로 다른 시간에 계산될 때 다른 점수를 산출할 수 있습니다. 점수가 10이면 스팟 용량 요청이 성공할 가능성이 높지만 보장되지는 않습니다. 최상의 결과를 얻으려면 점수에 따라 즉시 조치를 취하는 것이 좋습니다. 또한 용량 요청을 시도할 때마다 새로운 점수를 얻는 것이 좋습니다.

스팟 배치 점수에 필요한 권한

기본적으로 IAM ID(사용자, 역할 또는 그룹)는 the section called “스팟 배치 점수”을 사용할 수 있는 권한이 없습니다. IAM ID가 스팟 배치 점수를 사용하도록 허용하려면 `ec2:GetSpotPlacementScores` EC2 API 작업 사용 권한을 부여하는 IAM 정책을 생성해야 합니다. 그런 다음 이러한 권한이 필요한 IAM 자격 증명에 정책을 연결합니다.

다음은 `ec2:GetSpotPlacementScores` EC2 API 작업 사용 권한을 부여하는 IAM 정책의 예입니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:GetSpotPlacementScores",
            "Resource": "*"
        }
    ]
}
```

IAM 정책 편집에 대한 자세한 내용은 IAM 사용 설명서의 IAM 정책 편집을 참조하세요.

액세스 권한을 제공하려면 사용자, 그룹 또는 역할에 권한을 추가하세요:

- AWS IAM Identity Center의 사용자 및 그룹:

  권한 세트를 생성합니다. AWS IAM Identity Center 사용 설명서의 권한 세트 생성의 지침을 따릅니다.
- 보안 인증 공급자를 통해 IAM에서 관리되는 사용자:

  ID 페더레이션을 위한 역할을 생성합니다. IAM 사용 설명서의 Create a role for a third-party identity provider (federation)의 지침을 따릅니다.
- IAM 사용자:
  - 사용자가 맡을 수 있는 역할을 생성합니다. IAM 사용 설명서에서 Create a role for an IAM user의 지침을 따릅니다.
  - (권장되지 않음)정책을 사용자에게 직접 연결하거나 사용자를 사용자 그룹에 추가합니다. IAM 사용 설명서에서 사용자(콘솔)에 권한 추가의 지침을 따르세요.

스팟 배치 점수 계산

대상 목표 용량 및 컴퓨팅 요구 사항에 따라 스팟 배치 점수를 계산할 수 있습니다. 자세한 내용은 the section called “스팟 배치 점수의 작동 방식” 단원을 참조하십시오.

필요한 권한

필수 권한이 있는지 확인하세요. 자세한 내용은 the section called “필수 권한” 단원을 참조하십시오.

옵션

- 인스턴스 속성을 사용하여 계산

- 인스턴스 유형을 사용하여 계산
- AWS CLI를 사용하여 계산

자동화된 솔루션을 찾고 있나요? 이 사용 설명서의 수동 단계를 따르는 대신 스팟 배치 점수 추적기 대시보드를 구축하여 자동으로 점수를 캡처하고 이를 Amazon CloudWatch에 저장할 수 있습니다. 자세한 내용은 AWS에서의 스팟 배치 점수 추적기 대시보드 구축 지침을 참조하세요.

인스턴스 속성을 사용하여 계산

인스턴스 속성을 지정하여 스팟 배치 점수 계산

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. 스팟 인스턴스 요청 옆의 아래쪽 화살표를 선택하고 스팟 배치 점수 계산을 선택합니다.
4. 요구 사항 입력(Enter requirements)을 선택합니다.
5. 목표 용량(Target capacity)에서 인스턴스(instances), vCPU(vCPUs) 또는 메모리(MiB)(memory (MiB)) 양을 기준으로 원하는 용량을 입력합니다.
6. 인스턴스 유형 요구 사항(Instance type requirements)에서 컴퓨팅 요구 사항을 지정하고 Amazon EC2가 이러한 요구 사항에 따라 최적의 인스턴스 유형을 식별할 수 있도록 하려면 컴퓨팅 요구 사항에 맞는 인스턴스 속성 지정(Specify instance attributes that match your compute requirements)을 선택합니다.
7. vCPU(vCPUs)에 원하는 최소 및 최대 vCPU 수를 입력합니다. 무한을 지정하려면 최소 없음(No minimum), 최대 없음(No maximum) 또는 둘 다 선택합니다.
8. 메모리(GiB)(Memory (GiB))에 원하는 최소 및 최대 메모리 양을 입력합니다. 무한을 지정하려면 최소 없음(No minimum), 최대 없음(No maximum) 또는 둘 다 선택합니다.
9. CPU 아키텍처(CPU architecture)에서 필요한 인스턴스 아키텍처를 선택합니다.
10. (선택 사항) 추가 인스턴스 속성(Additional instance attributes)에서 필요에 따라 하나 이상의 속성을 지정하여 컴퓨팅 요구 사항을 더 자세히 표현할 수 있습니다. 각 추가 속성은 요청에 추가 제약 조건을 추가합니다. 추가 속성을 생략할 수 있으며 생략 시 기본값이 사용됩니다. 각 속성과 기본값에 대한 설명은 get-spot-placement-scores를 참조하세요.
11. (선택 사항) 지정한 속성을 가진 인스턴스 유형을 보려면 일치하는 인스턴스 유형 미리 보기(Preview matching instance types)를 확장합니다. 배치 평가에 사용되는 인스턴스 유형을 제외하려면 인스턴스를 선택한 다음 선택한 인스턴스 유형 제외(Exclude selected instance types)를 선택합니다.
12. 배치 점수 로드(Load placement scores)를 선택하고 결과를 검토합니다.

13. (선택 사항) 특정 리전에 대한 스팟 배치 점수를 표시하려면 평가할 리전(Regions to evaluate)에서 평가할 리전을 선택한 다음 배치 점수 계산(Calculate placement scores)을 선택합니다.
14. (선택 사항) 표시된 리전에서 가용 영역에 대한 스팟 배치 점수를 표시하려면 가용 영역당 배치 점수 제공(Provide placement scores per Availability Zone) 확인란을 선택합니다. 점수가 매겨진 가용 영역 목록은 모든 스팟 용량을 단일 가용 영역으로 시작하려는 경우에 유용합니다.
15. (선택 사항) 컴퓨팅 요구 사항을 편집하고 새 배치 점수를 얻으려면 편집(Edit)을 선택하고 필요에 따라 조정한 다음 배치 점수 계산(Calculate placement scores)을 선택합니다.

인스턴스 유형을 사용하여 계산

인스턴스 유형을 지정하여 스팟 배치 점수 계산(콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. 스팟 인스턴스 요청 옆의 아래쪽 화살표를 선택하고 스팟 배치 점수 계산을 선택합니다.
4. 요구 사항 입력(Enter requirements)을 선택합니다.
5. 목표 용량(Target capacity)에서 인스턴스(instances), vCPU(vCPUs) 또는 메모리(MiB)(memory (MiB)) 양을 기준으로 원하는 용량을 입력합니다.
6. 사용할 인스턴스 유형을 지정하려면 인스턴스 유형 요구 사항(Instance type requirements)에서 수동으로 인스턴스 유형 선택(Manually select instance types)을 선택합니다.
7. 인스턴스 유형 선택(Select instance types)을 선택하고 사용할 인스턴스 유형을 선택한 다음 선택(Select)을 선택합니다. 인스턴스 유형을 빠르게 찾으려면 필터 막대를 사용하여 여러 속성으로 인스턴스 유형을 필터링합니다.
8. 배치 점수 로드(Load placement scores)를 선택하고 결과를 검토합니다.
9. (선택 사항) 특정 리전에 대한 스팟 배치 점수를 표시하려면 평가할 리전(Regions to evaluate)에서 평가할 리전을 선택한 다음 배치 점수 계산(Calculate placement scores)을 선택합니다.
10. (선택 사항) 표시된 리전에서 가용 영역에 대한 스팟 배치 점수를 표시하려면 가용 영역당 배치 점수 제공(Provide placement scores per Availability Zone) 확인란을 선택합니다. 점수가 매겨진 가용 영역 목록은 모든 스팟 용량을 단일 가용 영역으로 시작하려는 경우에 유용합니다.
11. (선택 사항) 인스턴스 유형 목록을 편집하고 새 배치 점수를 얻으려면 편집(Edit)을 선택하고 필요에 따라 조정한 다음 배치 점수 계산(Calculate placement scores)을 선택합니다.

AWS CLI를 사용하여 계산

스팟 배치 점수 계산

1. (선택 사항) 스팟 배치 점수 구성에 지정할 수 있는 가능한 모든 파라미터를 생성하려면 get-spot-placement-scores 명령과 `--generate-cli-skeleton` 파라미터를 사용합니다.

```
aws ec2 get-spot-placement-scores \
    --region us-east-1 \
    --generate-cli-skeleton
```

출력의 예시는 다음과 같습니다.

```
{
    "InstanceTypes": [
        ""
    ],
    "TargetCapacity": 0,
    "TargetCapacityUnitType": "vcpu",
    "SingleAvailabilityZone": true,
    "RegionNames": [
        ""
    ],
    "InstanceRequirementsWithMetadata": {
        "ArchitectureTypes": [
            "x86_64_mac"
        ],
        "VirtualizationTypes": [
            "hvm"
        ],
        "InstanceRequirements": {
            "VCpuCount": {
                "Min": 0,
                "Max": 0
            },
            "MemoryMiB": {
                "Min": 0,
                "Max": 0
            },
            "CpuManufacturers": [
                "amd"
            ],
```

```
            "MemoryGiBPerVCpu": {
                "Min": 0.0,
                "Max": 0.0
            },
            "ExcludedInstanceTypes": [
                ""
            ],
            "InstanceGenerations": [
                "previous"
            ],
            "SpotMaxPricePercentageOverLowestPrice": 0,
            "OnDemandMaxPricePercentageOverLowestPrice": 0,
            "BareMetal": "excluded",
            "BurstablePerformance": "excluded",
            "RequireHibernateSupport": true,
            "NetworkInterfaceCount": {
                "Min": 0,
                "Max": 0
            },
            "LocalStorage": "included",
            "LocalStorageTypes": [
                "hdd"
            ],
            "TotalLocalStorageGB": {
                "Min": 0.0,
                "Max": 0.0
            },
            "BaselineEbsBandwidthMbps": {
                "Min": 0,
                "Max": 0
            },
            "AcceleratorTypes": [
                "fpga"
            ],
            "AcceleratorCount": {
                "Min": 0,
                "Max": 0
            },
            "AcceleratorManufacturers": [
                "amd"
            ],
            "AcceleratorNames": [
                "vu9p"
            ],
```

```
            "AcceleratorTotalMemoryMiB": {
                "Min": 0,
                "Max": 0
            }
        }
    },
    "DryRun": true,
    "MaxResults": 0,
    "NextToken": ""
}
```

2. 이전 단계의 출력을 사용하여 JSON 구성 파일을 생성하고 다음과 같이 구성합니다.

   a. `TargetCapacity`에 대해 인스턴스, vCPU 또는 메모리(MiB) 양으로 스팟 용량을 입력합니다.

   b. `TargetCapacityUnitType`에 대해 목표 용량의 단위를 입력합니다. 이 파라미터를 생략하면 기본적으로 `units`가 사용됩니다.

      유효한 값 :`units`(인스턴스 수로 변환됨) | `vcpu` | `memory-mib`

   c. `SingleAvailabilityZone`의 경우 점수가 매겨진 가용 영역 목록을 반환하는 응답에 대해 `true`를 지정합니다. 점수가 매겨진 가용 영역 목록은 모든 스팟 용량을 단일 가용 영역으로 시작하려는 경우에 유용합니다. 이 파라미터를 생략하면 기본값으로 `false`가 사용되고 응답에서 점수가 매겨진 리전 목록을 반환합니다.

   d. (선택 사항) `RegionNames`에 대해 필터로 사용할 리전을 지정합니다. 리전 코드(예: `us-east-1`)를 지정해야 합니다.

      리전 필터를 사용하면 응답에서 지정한 리전만 반환합니다. `SingleAvailabilityZone`에 대해 `true`를 지정한 경우 응답은 지정된 리전의 가용 영역만 반환합니다.

   e. `InstanceTypes` 또는 `InstanceRequirements` 중 하나를 포함할 수 있지만 둘 다 동일한 구성에 포함할 수는 없습니다.

      JSON 구성에서 다음 중 하나를 지정합니다.

      - 인스턴스 유형 목록을 지정하려면 `InstanceTypes` 파라미터에 인스턴스 유형을 지정합니다. 세 가지 이상의 서로 다른 인스턴스 유형을 지정합니다. 인스턴스 유형을 하나 또는 2개만 지정하는 경우 스팟 배치 점수는 낮은 점수를 반환합니다. 인스턴스 유형 목록은 Amazon EC2 인스턴스 유형을 참조하세요.
      - Amazon EC2가 해당 속성과 일치하는 인스턴스 유형을 식별하도록 인스턴스 속성을 지정하려면 `InstanceRequirements` 구조에 있는 속성을 지정합니다.

VCpuCount, MemoryMiB 및 CpuManufacturers의 값을 입력해야 합니다. 다른 속성을 생략할 수 있으며 생략 시 기본값이 사용됩니다. 각 속성과 기본값에 대한 설명은 get-spot-placement-scores를 참조하세요.

구성에 대한 예시는 구성의 예 섹션을 참조하세요.

3. JSON 파일에 지정한 요구 사항에 대한 스팟 배치 점수를 얻으려면 get-spot-placement-scores 명령을 사용하고 --cli-input-json 파라미터를 사용하여 JSON 파일의 이름과 경로를 지정합니다.

```
aws ec2 get-spot-placement-scores \
    --region us-east-1 \
    --cli-input-json file://file_name.json
```

SingleAvailabilityZone이 false로 설정되거나 생략된 경우의 출력 예(생략된 경우 기본적으로 false 사용) – 점수가 매겨진 리전 목록이 반환됩니다.

```
"SpotPlacementScores": [
    {
        "Region": "us-east-1",
        "Score": 7
    },
    {
        "Region": "us-west-1",
        "Score": 5
    },
   ...
```

SingleAvailabilityZone이 true로 설정된 경우의 출력 예 – 점수가 매겨진 가용 영역 목록이 반환됩니다.

```
"SpotPlacementScores": [
    {
        "Region": "us-east-1",
        "AvailabilityZoneId": "use1-az1",
        "Score": 8
    },
    {
        "Region": "us-east-1",
        "AvailabilityZoneId": "usw2-az3",
```

```
        "Score": 6
    },
    ...
```

구성의 예

AWS CLI를 사용하는 경우 다음 예제 구성을 사용할 수 있습니다.

구성의 예

- 예시: 인스턴스 유형 및 목표 용량 지정
- 예시: 인스턴스 유형 및 목표 용량(메모리) 지정
- 예시: 속성 기반 인스턴스 유형 선택을 위한 속성 지정
- 예시: 속성 기반 인스턴스 유형 선택을 위한 속성을 지정하고 점수가 매겨진 가용 영역의 목록 반환

예시: 인스턴스 유형 및 목표 용량 지정

다음 예제 구성은 3개의 서로 다른 인스턴스 유형과 스팟 인스턴스 500개의 대상 스팟 용량을 지정합니다.

```
{
    "InstanceTypes": [
        "m5.4xlarge",
        "r5.2xlarge",
        "m4.4xlarge"
    ],
    "TargetCapacity": 500
}
```

예시: 인스턴스 유형 및 목표 용량(메모리) 지정

다음 예제 구성은 세 가지 인스턴스 유형과 500,000MiB 메모리의 대상 스팟 용량을 지정합니다. 여기서 시작할 스팟 인스턴스의 수는 총 500,000MiB의 메모리를 제공해야 합니다.

```
{
    "InstanceTypes": [
        "m5.4xlarge",
        "r5.2xlarge",
        "m4.4xlarge"
```

```
    ],
    "TargetCapacity": 500000,
    "TargetCapacityUnitType": "memory-mib"
}
```

예시: 속성 기반 인스턴스 유형 선택을 위한 속성 지정

다음 예제 구성은 속성 기반 인스턴스 유형 선택을 위해 구성되며 그 뒤에 예제 구성에 대한 텍스트 설명이 나옵니다.

```
{
    "TargetCapacity": 5000,
    "TargetCapacityUnitType": "vcpu",
    "InstanceRequirementsWithMetadata": {
        "ArchitectureTypes": ["arm64"],
        "VirtualizationTypes": ["hvm"],
        "InstanceRequirements": {
            "VCpuCount": {
                "Min": 1,
                "Max": 12
            },
            "MemoryMiB": {
                "Min": 512
            }
        }
    }
}
```

InstanceRequirementsWithMetadata

속성 기반 인스턴스 유형 선택을 사용하려면 구성에 `InstanceRequirementsWithMetadata` 구조를 포함하고 스팟 인스턴스에 대해 원하는 속성을 지정해야 합니다.

앞의 예에서는 다음과 같은 필수 인스턴스 속성이 지정되었습니다.

- `ArchitectureTypes` - 인스턴스 유형의 아키텍처 유형은 `arm64`여야 합니다.
- `VirtualizationTypes` - 인스턴스 유형의 가상화 유형은 `hvm`이어야 합니다.
- `VCpuCount` - 인스턴스 유형에 최소 1개, 최대 12개의 vCPU가 있어야 합니다.
- `MemoryMiB` - 인스턴스 유형에 최소 512MiB의 메모리가 있어야 합니다. `Max` 파라미터를 생략하면 최대 제한이 없는 것입니다.

지정할 수 있는 몇 가지 다른 선택적 속성이 있습니다. 속성 목록은 get-spot-placement-scores를 참조하세요.

`TargetCapacityUnitType`

`TargetCapacityUnitType` 파라미터는 목표 용량의 단위를 지정합니다. 이 예제에서 목표 용량은 `5000`이고 목표 용량 단위 유형은 `vcpu`입니다. 이들은 함께 원하는 목표 용량 vCPU 5000개를 지정합니다. 여기서 시작할 스팟 인스턴스 수는 총 5000개의 vCPU를 제공해야 합니다.

예시: 속성 기반 인스턴스 유형 선택을 위한 속성을 지정하고 점수가 매겨진 가용 영역의 목록 반환

다음 예제 구성은 속성 기반 인스턴스 유형 선택을 위해 구성됩니다.
`"SingleAvailabilityZone": true`를 지정하면 응답에서 점수가 매겨진 가용 영역 목록을 반환합니다.

```
{
    "TargetCapacity": 1000,
    "TargetCapacityUnitType": "vcpu",
    "SingleAvailabilityZone": true,
    "InstanceRequirementsWithMetadata": {
        "ArchitectureTypes": ["arm64"],
        "VirtualizationTypes": ["hvm"],
        "InstanceRequirements": {
            "VCpuCount": {
                "Min": 1,
                "Max": 12
            },
            "MemoryMiB": {
                "Min": 512
            }
        }
    }
}
```

스팟 인스턴스 데이터 피드를 사용하여 스팟 인스턴스 비용 추적

스팟 인스턴스 요금을 쉽게 이해할 수 있도록 Amazon EC2에서는 스팟 인스턴스 사용량 및 요금을 설명하는 데이터 피드를 제공합니다. 이 데이터 피드는 데이터 피드를 구독할 때 지정하는 Amazon S3 버킷으로 전송됩니다.

일반적으로 데이터 피드 파일은 1시간에 한 번씩 버킷에 도착합니다. 특정 시간 동안 스팟 인스턴스가 실행되지 않는 경우 해당 시간에 대한 데이터 피드 파일이 수신되지 않습니다.

일반적으로 각 시간의 스팟 인스턴스 사용량은 단일 데이터 파일로 제공됩니다. 이러한 파일은 버킷으로 전송되기 전에 압축(gzip)됩니다. 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**
>
> AWS 계정당 스팟 인스턴스 데이터 피드를 한 개만 생성할 수 있습니다.

스팟 인스턴스 데이터 피드는 중국(베이징), 중국(닝샤), AWS GovCloud(미국) 및 기본적으로 비활성화된 리전을 제외한 모든 AWS 리전에서 지원됩니다.

내용

- 데이터 피드 파일 이름 및 형식
- Amazon S3 버킷 요구 사항
- 스팟 인스턴스 데이터 피드 구독
- 데이터 피드에서 데이터 보기
- 스팟 인스턴스 데이터 피드 삭제

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 **amzn-s3-demo-bucket**이고 접두사가**my-prefix**인 경우 파일 이름은 다음과 같습니다.

```
amzn-s3-demo-bucket.s3.amazonaws.com/my-prefix/111122223333.2023-12-09-07.001.b959dbc6.gz
```

버킷 이름에 대한 자세한 내용은 Amazon S3 사용 설명서에서 버킷 이름 지정 규칙을 참조하세요.

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. 예를 들어 SpotUsage:c1.medium 입니다.
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 요청에 대해 지정된 최고가입니다.
MarketPrice	Timestamp 필드에 지정된 시간의 스팟 가격입니다.
Charge	이 인스턴스 사용량에 대해 청구된 가격입니다.
Version	데이터 피드 버전. 가능한 버전은 1.0입니다.

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 고려하세요.

- 버킷에 대한 FULL_CONTROL 권한이 있어야 합니다. 버킷 소유자인 경우 기본적으로 이 권한이 있습니다. 그렇게 하지 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. 자세한 내용은 Amazon CloudWatch Logs 사용 설명서의 Amazon S3로 전송된 로그를 참조하세요.

  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.
- AWS Key Management Service(SSE-KMS)에 저장된 AWS KMS 키로 서버 측 암호화를 사용하여 Amazon S3 버킷을 암호화하는 경우 고객 관리형 키를 사용해야 합니다. 자세한 내용은 Amazon CloudWatch Logs 사용 설명서의 Amazon S3 버킷 서버 측 암호화 섹션을 참조하세요.

스팟 인스턴스 데이터 피드 구독

언제든지 스팟 인스턴스 데이터 피드를 구독할 수 있습니다. Amazon EC2 콘솔에서는 이 작업을 완료할 수 없습니다.

버킷에 충분한 권한이 없다는 오류가 발생하는 경우 스팟 인스턴스의 데이터 피드 문제 해결을 참조하세요.

AWS CLI

데이터 피드를 구독하려면

create-spot-datafeed-subscription 명령을 사용합니다.

접두사를 사용하여 버킷을 지정하려면 다음 예제를 사용합니다.

```
aws ec2 create-spot-datafeed-subscription \
    --bucket amzn-s3-demo-bucket \
    --prefix my-prefix
```

접두사 없이 버킷을 지정하려면 다음 예제를 사용합니다.

```
aws ec2 create-spot-datafeed-subscription \
    --bucket amzn-s3-demo-bucket
```

PowerShell

데이터 피드를 구독하려면

New-EC2SpotDatafeedSubscription cmdlet을 사용합니다.

접두사를 사용하여 버킷을 지정하려면 다음 예제를 사용합니다.

```
New-EC2SpotDatafeedSubscription `
    -Bucket amzn-s3-demo-bucket `
    -Prefix my-prefix
```

접두사 없이 버킷을 지정하려면 다음 예제를 사용합니다.

```
New-EC2SpotDatafeedSubscription `
    -Bucket amzn-s3-demo-bucket
```

데이터 피드에서 데이터 보기

AWS Management Console에서 AWS CloudShell을 엽니다. 다음 s3 sync 명령을 사용하여 데이터 피드를 위해 S3 버킷에서 .gz 파일을 가져와 지정한 폴더에 저장합니다.

```
aws s3 sync s3://amzn-s3-demo-bucket ./data-feed
```

.gz 파일의 콘텐츠를 표시하려면 S3 버킷의 콘텐츠를 저장한 폴더로 이동합니다.

```
cd data-feed
```

ls 명령을 사용하여 파일 이름을 확인합니다. zcat 명령을 파일 이름과 함께 사용하여 압축 파일의 콘텐츠를 표시합니다. 다음은 예시 명령입니다.

```
zcat  111122223333.2023-12-09-07.001.b959dbc6.gz
```

출력의 예시는 다음과 같습니다.

```
#Version: 1.0
```

```
#Fields: Timestamp UsageType Operation InstanceID MyBidID MyMaxPrice MarketPrice Charge
 Version
2023-12-09 07:13:47 UTC USE2-SpotUsage:c7a.medium       RunInstances:SV050
 i-0c3e0c0b046e050df     sir-pwq6nmfp    0.0510000000 USD        0.0142000000 USD
  0.0142000000 USD        1
```

스팟 인스턴스 데이터 피드 삭제

스팟 인스턴스 데이터 피드 작업이 끝나면 데이터 피드를 삭제할 수 있습니다.

AWS CLI

데이터 피드를 삭제하려면

delete-spot-datafeed-subscription 명령을 사용합니다.

```
aws ec2 delete-spot-datafeed-subscription
```

PowerShell

데이터 피드를 삭제하려면

Remove-EC2SpotDatafeedSubscription cmdlet을 사용합니다.

```
Remove-EC2SpotDatafeedSubscription
```

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 수임할 수 있으므로 서비스 연결 역할은 AWS 서비스에 권한을 위임하는 안전한 방법을 제공합니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할을 참조하세요.

Amazon EC2는 AWSServiceRoleForEC2Spot이라는 이름의 서비스 연결 역할을 사용하여 사용자 대신 스팟 인스턴스를 시작하고 관리합니다.

AWSServiceRoleForEC2Spot으로 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 상황에서는 서비스 연결 역할을 수동으로 생성할 필요가 없습니다. 사용자가 콘솔을 사용하여 스팟 인스턴스를 처음으로 요청하면 Amazon EC2가 AWSServiceRoleForEC2Spot 서비스 연결 역할을 생성합니다.

Amazon EC2가 이 서비스 연결 역할을 지원하기 시작한 2017년 10월 이전에 활성 스팟 인스턴스 요청을 보유한 경우 Amazon EC2에는 사용자의 AWS 계정에 AWSServiceRoleForEC2Spot 역할이 이미 생성되어 있습니다. 자세한 내용은 IAM 사용 설명서의 내 계정에 표시되는 새 역할을 참조하세요.

AWS CLI 또는 API를 사용하여 스팟 인스턴스를 요청하는 경우 먼저 이 역할이 있는지 확인해야 합니다.

콘솔을 사용하여 AWSServiceRoleForEC2Spot 생성

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 역할을 선택합니다.
3. 역할 생성(Create role)을 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.

AWS CLI를 사용하여 AWSServiceRoleForEC2Spot 생성

다음과 같이 create-service-linked-role 명령을 사용합니다.

```
aws iam create-service-linked-role --aws-service-name spot.amazonaws.com
```

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 스냅샷에 사용할 고객 관리형 키에 대한 액세스 권한 부여

스팟 인스턴스 요청에서 암호화된 AMI 또는 암호화된 Amazon EBS 스냅샷을 지정하고 암호화용 고객 관리형 키를 사용하는 경우 Amazon EC2에서 자동으로 인스턴스를 시작하려면 고객 관리형 키를 사용

할 권한을 AWSServiceRoleForEC2Spot 역할에 부여해야 할 수 있습니다. 이렇게 하려면 다음 절차에 표시된 바와 같이 고객 관리형 키에 대한 권한 부여를 추가해야 합니다.

권한을 제공할 때 권한 부여는 키 정책을 대체합니다. 자세한 내용은 AWS Key Management Service 개발자 안내서에서 권한 부여 사용 및 AWS KMS의 키 정책 사용을 참조하세요.

AWSServiceRoleForEC2Spot 역할에 고객 관리형 키를 사용할 수 있는 권한 부여

- create-grant 명령을 사용하여 고객 관리형 키에 대한 권한 부여를 추가하고 권한 부여에 의해 허용되는 작업을 수행할 수 있는 권한이 부여된 보안 주체(AWSServiceRoleForEC2Spot 서비스 연결 역할)를 지정합니다. 고객 관리형 키는 `key-id` 파라미터와 고객 관리형 키의 ARN으로 지정됩니다. 보안 주체는 AWSServiceRoleForEC2Spot 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/aws-service-role/
spot.amazonaws.com/AWSServiceRoleForEC2Spot \
    --operations "Decrypt" "Encrypt" "GenerateDataKey"
 "GenerateDataKeyWithoutPlaintext" "CreateGrant" "DescribeKey" "ReEncryptFrom"
 "ReEncryptTo"
```

스팟 인스턴스 할당량

리전별로 AWS 계정당 실행 중 스팟 인스턴스 및 대기 중 스팟 인스턴스 요청 수에는 할당량이 있습니다. 대기 중 스팟 인스턴스 요청이 이행된 후에는 실행 중 인스턴스가 할당량 계산에 포함되므로 이 요청은 더 이상 할당량 계산에 포함되지 않습니다.

스팟 인스턴스 할당량은 실행 중인 스팟 인스턴스가 사용 중이거나 미결 요청의 이행 보류 중에 사용할 가상 중앙 처리 유닛(vCPU) 수를 기준으로 관리됩니다. 스팟 인스턴스를 종료하고 스팟 인스턴스 요청을 취소하지 않으면 Amazon EC2가 스팟 인스턴스 종료를 감지하고 요청을 닫을 때까지 해당 요청이 스팟 인스턴스 vCPU 할당량 계산에 반영됩니다.

스팟 인스턴스에는 다음과 같은 할당량 유형이 있습니다.

명칭	기본값	조정 가능
모든 DL 스팟 인스턴스 요청	0	예
모든 F 스팟 인스턴스 요청	0	예
모든 G 및 VT 스팟 인스턴스 요청	0	예
모든 Inf 스팟 인스턴스 요청	0	예
모든 P4, P3 및 P2 스팟 인스턴스 요청	0	예
모든 P5 스팟 인스턴스 요청	0	예
모든 표준(A, C, D, H, I, M, R, T, Z) 스팟 인스턴스 요청	5	예
모든 Trn 스팟 인스턴스 요청	0	예
모든 X 스팟 인스턴스 요청	0	예

Amazon EC2는 사용량에 따라 자동으로 스팟 인스턴스 할당량을 조정하지만 필요한 경우 할당량 증가를 요청할 수 있습니다. 예를 들어 현재 할당량에서 허용되는 것보다 많은 스팟 인스턴스를 실행하려는 경우 할당량 증가를 요청할 수 있습니다. 스팟 인스턴스 요청을 제출한 후 `Max spot instance count exceeded` 오류가 발생할 경우에도 할당량 증가를 요청할 수 있습니다. 할당량 증가를 요청하려면 Amazon EC2 서비스 할당량에서 설명한 Service Quotas 콘솔을 사용하세요.

변화하는 애플리케이션 요구 사항을 충족하는 모든 인스턴스 유형 조합을 시작할 수 있습니다. 예를 들어 모든 표준 스팟 인스턴스 요청 할당량이 256개 vCPU인 경우, 32개의 `m5.2xlarge` 스팟 인스턴스(32x8 vCPU) 또는 16개의 `c5.4xlarge` 스팟 인스턴스(16x16 vCPU)를 요청할 수 있습니다.

Amazon CloudWatch 지표 통합을 통해 할당량에 대해 EC2 사용량을 모니터링할 수 있습니다. 할당량 도달에 대해 경고를 받도록 경보를 구성할 수도 있습니다. 자세한 내용은 Service Quotas 사용 설명서의 Service Quotas 및 Amazon CloudWatch 경보 및 을 참조하세요.

Amazon EC2 전용 호스트

Amazon EC2 전용 호스트는 사용자를 위한 완전 전용인 물리적 서버입니다. 선택적으로 인스턴스 용량을 다른 AWS 계정과 공유하도록 선택할 수 있습니다. 자세한 내용은 크로스 계정 Amazon EC2 전용 호스트 공유 단원을 참조하세요.

전용 호스트는 인스턴스 배치에 대한 가시성과 제어 기능을 제공하며 호스트 선호도를 지원합니다. 즉, 특정 호스트에서 인스턴스를 시작하고 실행할 수 있으며 인스턴스가 특정 호스트에서만 실행되도록 할 수 있습니다. 자세한 내용은 Amazon EC2 전용 호스트 자동 배치 및 호스트 선호도 섹션을 참조하세요.

전용 호스트는 포괄적인 기존 보유 라이선스 사용(BYOL) 지원을 제공합니다. 라이센스 조항에 따라 Windows Server, SQL Server, SUSE Linux Enterprise Server, Red Hat Enterprise Linux 또는 VM, 소켓 또는 물리적 코어에 바인딩된 기타 소프트웨어 라이선스를 포함하여 기존 소켓당, 코어당 또는 VM당 소프트웨어 라이선스를 사용할 수 있습니다.

전용 하드웨어에서 인스턴스를 실행해야 하지만 인스턴스 배치에 대한 가시성이나 제어 기능이 필요하지 않고 소켓당 또는 코어당 소프트웨어 라이선스를 사용할 필요가 없는 경우 전용 인스턴스를 대신 사용할 수 있습니다. 전용 인스턴스와 전용 호스트 모두 전용 물리적 서버로 Amazon EC2 인스턴스를 시작하는 데 사용할 수 있습니다. 전용 호스트의 인스턴스와 전용 인스턴스는 성능이나 보안상의 차이나 물리적 차이는 없습니다. 그러나 이들 사이에는 몇 가지 주요 차이점이 있습니다. 다음 테이블에서는 전용 인스턴스와 전용 호스트의 주요 차이점 중 일부를 요약하여 설명합니다.

	전용 호스트	Dedicated Instance
전용 물리적 서버	사용자 전용 인스턴스 용량을 갖춘 물리적 서버입니다.	단일 고객 계정 전용 물리적 서버
인스턴스 용량 공유	다른 계정과 인스턴스 용량을 공유할 수 있음.	지원되지 않음
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

	전용 호스트	Dedicated Instance
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 Amazon EC2 전용 호스트 복구 섹션을 참조하세요.	지원
Bring Your Own License(BYOL)	지원	부분적 지원 *
용량 예약	지원되지 않음	지원

* 소프트웨어 보증을 통한 라이선스 이동성을 갖춘 Microsoft SQL Server와 Windows 가상 데스크톱 액세스(VDA) 라이선스를 전용 인스턴스와 함께 사용할 수 있습니다.

전용 인스턴스에 대한 자세한 정보는 Amazon EC2 전용 인스턴스 섹션을 참조하세요.

전용 호스트 제한 사항

전용 호스트를 할당하기 전에 다음 제한 및 제약에 유의하세요.

- 전용 호스트에서 RHEL 및 SUSE Linux를 실행하려면 자체 AMI를 가져와야 합니다. AWS에서 제공하거나 전용 호스트로 AWS Marketplace에서 사용할 수 있는 RHEL 및 SUSE Linux AMI는 사용할 수 없습니다. 자체 AMI를 만드는 방법에 대한 자세한 내용은 Amazon EC2 전용 호스트에 기존 보유 소프트웨어 라이선스 사용을 참조하세요.

  이 제한은 고용량 메모리 인스턴스(`u-6tb1.metal`, `u-9tb1.metal`, `u-12tb1.metal`, `u-18tb1.metal` 및 `u-24tb1.metal`)용으로 할당된 호스트에는 적용되지 않습니다. 이러한 호스트에는 AWS에서 제공하는 RHEL 및 SUSE Linux AMI 또는 AWS Marketplace에서 사용 가능한 AMI를 사용할 수 있습니다.
- 리전별 AWS 계정당 각 인스턴스 패밀리에서 실행 중인 전용 호스트 수에는 제한이 있습니다. 할당량은 실행 중인 인스턴스에만 적용됩니다. 인스턴스가 보류 중, 중지 중 또는 중지된 상태이면 할당

량에 포함되지 않습니다. 계정 할당량을 보거나 할당량 증가를 요청하려면 Service Quotas 콘솔을 사용합니다.

- 호스트 Resource Groups를 지정하는 시작 템플릿을 사용하는 경우 Auto Scaling 그룹이 지원됩니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 고급 설정을 사용한 시작 템플릿 생성하기를 참조하세요.
- Amazon RDS 인스턴스는 지원되지 않습니다.
- AWS프리 티어는 전용 호스트에서 제공되지 않습니다.
- 인스턴스 배치 제어는 전용 호스트에서 인스턴스 시작을 관리하는 것을 말합니다. 전용 호스트는 배치 그룹으로 시작할 수 없습니다.
- 가상화된 인스턴스 유형에 대해 호스트를 할당하는 경우 호스트가 할당된 후에는 인스턴스 유형을 `.metal` 인스턴스 유형으로 수정할 수 없습니다. 예를 들어 `m5.large` 인스턴스 유형에 대해 호스트를 할당하는 경우 인스턴스 유형을 `m5.metal`로 수정할 수 없습니다.

  마찬가지로 `.metal` 인스턴스 유형에 대해 호스트를 할당하는 경우 호스트가 할당된 후에는 인스턴스 유형을 가상화된 인스턴스 유형으로 수정할 수 없습니다. 예를 들어 `m5.metal` 인스턴스 유형에 대해 호스트를 할당하는 경우 인스턴스 유형을 `m5.large`로 수정할 수 없습니다.

내용

- Amazon EC2 전용 호스트 요금 및 청구
- Amazon EC2 전용 호스트 인스턴스 용량 구성
- Amazon EC2 전용 호스트의 버스트 가능 T3 인스턴스
- Amazon EC2 전용 호스트에 기존 보유 소프트웨어 라이선스 사용
- Amazon EC2 전용 호스트 자동 배치 및 호스트 선호도
- 계정에서 사용할 Amazon EC2 전용 호스트 할당
- Amazon EC2 전용 호스트에서 Amazon EC2 인스턴스 시작
- Amazon EC2 인스턴스를 호스트 리소스 그룹으로 인스턴스 시작
- 기존 Amazon EC2 전용 호스트의 자동 배치 설정 수정
- 기존 Amazon EC2 전용 호스트에 대해 지원되는 인스턴스 유형 수정
- Amazon EC2 인스턴스의 Amazon EC2 전용 호스트 테넌시 및 선호도 수정
- Amazon EC2 전용 호스트 릴리스
- Nitro 기반 Amazon EC2 전용 호스트로 마이그레이션

- 전용 호스트 청구 할인을 위한 전용 호스트 예약 구매
- 크로스 계정 Amazon EC2 전용 호스트 공유
- AWS Outposts의 Amazon EC2 전용 호스트
- Amazon EC2 전용 호스트 복구
- Amazon EC2 전용 호스트의 호스트 유지 관리
- Amazon EC2 전용 호스트 상태 모니터링
- AWS Config를 사용하여 Amazon EC2 전용 호스트 구성 변경 내용 추적

Amazon EC2 전용 호스트 요금 및 청구

전용 호스트의 요금은 결제 옵션에 따라 다릅니다.

결제 옵션

- 온디맨드 전용 호스트
- Dedicated Host Reservations
- Savings Plans
- 전용 호스트의 Windows Server에 대한 요금

온디맨드 전용 호스트

계정에 전용 호스트를 할당하면 온디맨드 결제가 자동으로 활성화됩니다.

전용 호스트에 대한 온디맨드 요금은 인스턴스 패밀리 및 리전에 따라 다릅니다. 인스턴스를 시작하기로 선택한 인스턴스의 수량이나 크기에 관계없이 활성 전용 호스트에 대해 초당(최소 60초) 비용을 지불합니다. 온디맨드 요금에 대한 자세한 내용은 Amazon EC2 전용 호스트 온디맨드 요금을 참조하세요.

언제든 온디맨드 전용 호스트를 해제하여 이에 대한 요금 청구를 중단시킬 수 있습니다. 전용 호스트 해제에 대한 자세한 내용은 Amazon EC2 전용 호스트 릴리스 섹션을 참조하세요.

Dedicated Host Reservations

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 1년 및 3년인 경우에 가능합니다. 일부 인스턴스 패밀리만 선결제가 없는 예약에 대해 3년을 지원합니다.
- 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 단일 가용 영역에서 동일한 인스턴스 패밀리를 지원하는 하나 이상의 호스트를 포함할 수 있습니다. 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 가용 영역은 연결하고자 하는 전용 호스트의 인스턴스 패밀리 및 가용 영역과 일치해야 합니다.

하나의 예약이 전용 호스트와 연결되면 예약 기간이 끝날 때까지 전용 호스트를 해제할 수 없습니다.

예약 요금에 대한 자세한 내용은 Amazon EC2 전용 호스트 요금 섹션을 참조하세요.

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 절감형 플랜 사용 설명서를 참조하세요.

> **Note**
>
> Savings Plans는 `u-6tb1.metal`, `u-9tb1.metal`, `u-12tb1.metal`, `u-18tb1.metal` 및 `u-24tb1.metal` 전용 호스트에서 지원되지 않습니다.

전용 호스트의 Windows Server에 대한 요금

Microsoft 라이선스 약관에 따라 기존 Windows Server 및 SQL Server 라이선스를 전용 호스트에서 사용할 수 있습니다. 기존 보유 라이선스를 사용하는 경우 소프트웨어 사용에 따른 추가 요금이 없습니다.

또한 Amazon에서 제공하는 Windows Server AMI를 사용하여 전용 호스트에서 최신 Windows Server 버전도 실행할 수 있습니다. 이는 전용 호스트에서 실행할 수 있는 기존 SQL Server 라이선스가 있지만 SQL Server 워크로드를 실행하려면 Windows Server가 필요한 시나리오에서 일반적입니다. Amazon에서 제공하는 Windows Server AMI는 최신 세대의 인스턴스 유형에서만 지원됩니다. 자세한 내용은 Amazon EC2 전용 호스트 요금을 참조하세요.

Amazon EC2 전용 호스트 인스턴스 용량 구성

전용 호스트는 다양한 패밀리 및 크기의 인스턴스를 실행할 수 있도록 다양한 구성(물리적 코어, 소켓, vCPU)을 지원합니다.

계정에 전용 호스트를 할당할 때 단일 인스턴스 유형 또는 동일한 인스턴스 패밀리 내의 여러 인스턴스 유형을 지원하는 구성을 선택할 수 있습니다. 호스트에서 실행할 수 있는 인스턴스 수는 선택한 구성에 따라 달라집니다.

내용

- 단일 인스턴스 유형 지원
- 여러 인스턴스 유형 지원

단일 인스턴스 유형 지원

한 가지 인스턴스 유형만 지원하는 전용 호스트를 할당할 수 있습니다. 이 구성을 사용하면 전용 호스트에서 시작하는 모든 인스턴스는 호스트 할당 시 지정한 인스턴스 유형과 동일해야 합니다.

예를 들어 `m5.4xlarge` 인스턴스 유형만 지원하는 호스트를 할당할 수 있습니다. 이 경우 해당 호스트에서는 `m5.4xlarge` 인스턴스만 실행할 수 있습니다.

호스트에서 시작할 수 있는 인스턴스 수는 호스트에서 제공하는 물리적 코어 수와 지정된 인스턴스 유형에서 사용하는 코어 수에 따라 달라집니다. 예를 들어 `m5.4xlarge` 인스턴스에 호스트를 할당하는 경우 호스트는 48개의 물리적 코어를 제공하며 각 `m5.4xlarge` 인스턴스는 8개의 물리적 코어를 사용합니다. 즉, 해당 호스트에서 최대 6개의 인스턴스를 시작할 수 있습니다(물리적 코어 48개/인스턴스당 코어 8개 = 인스턴스 6개).

여러 인스턴스 유형 지원

동일한 인스턴스 패밀리 내의 여러 인스턴스 유형을 지원하는 전용 호스트를 할당할 수 있습니다. 이렇게 하면 동일한 인스턴스 패밀리에 속하고 호스트에 충분한 인스턴스 용량이 있는 한 동일한 호스트에서 다른 인스턴스 유형을 실행할 수 있습니다.

예를 들어 R5 인스턴스 패밀리 내의 다양한 인스턴스 유형을 지원하는 호스트를 할당할 수 있습니다. 이 경우 해당 호스트에서 호스트의 물리적 코어 용량 내에서 R5 인스턴스 유형의 모든 조합(예: `r5.large`, `r5.xlarge`, `r5.2xlarge`, `r5.4xlarge`)을 시작할 수 있습니다.

다음 인스턴스 패밀리는 여러 인스턴스 유형을 지원하는 전용 호스트를 지원합니다.

- 범용: A1 | M5 | M5n | M6i | M7i | T3
- 컴퓨팅 최적화: C5 | C5n | C6i | C7i
- 메모리 최적화: R5 | R5n | R6i | R7i

호스트에서 실행할 수 있는 인스턴스 수는 호스트에서 제공하는 물리적 코어 수와 호스트에서 실행하는 각 인스턴스 유형에서 사용하는 코어 수에 따라 달라집니다. 예를 들어 48개의 물리적 코어를 제공하는 R5 호스트를 할당하고 2개의 `r5.2xlarge` 인스턴스(코어 4개 x 인스턴스 2개) 및 3개의 `r5.4xlarge` 인스턴스(코어 8개 x 인스턴스 3개)를 실행하는 경우 해당 인스턴스는 총 32개의 코어를 사용하며 남은 16개 코어 범위 내에서 R5 인스턴스를 특정 조합으로 실행할 수 있습니다.

하지만 각 인스턴스 패밀리에서 각 인스턴스 유형에 실행할 수 있는 인스턴스 수에는 제한이 있습니다. 예를 들어 R5 전용 호스트는 32개의 물리적 코어를 사용하는 최대 2개의 `r5.8xlarge` 인스턴스를 지원합니다. 이 경우 더 작은 유형의 R5 인스턴스를 추가로 사용하여 호스트의 코어 용량을 채울 수 있습니다. 각 인스턴스 패밀리에 대해 지원되는 인스턴스 유형 수는 전용 호스트 구성표를 참조하세요.

다음 표는 인스턴스 유형 조합의 예를 보여줍니다.

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`	

인스턴스 패밀리	인스턴스 유형 조합 예	
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`	

고려 사항

여러 인스턴스 유형을 지원하는 전용 호스트를 사용하여 작업할 때는 다음 사항에 유의하세요.

- 동일한 호스트에서 여러 인스턴스 유형을 사용하는 것은 동일한 인스턴스 패밀리 내에서만 가능합니다.
- 인스턴스 유형을 혼용하는 경우 호스트 사용률을 극대화하려면 먼저 더 큰 인스턴스 유형을 시작한 다음 더 작은 인스턴스 유형을 시작하는 것이 좋습니다.
  - 전용 호스트에서 인스턴스 유형의 조합 및 시작 순서에 따라 호스트의 사용률을 극대화하는 것이 물리적으로 불가능할 수 있습니다. 호스트에서 인스턴스 유형을 혼용하는 경우 일부 용량은 호스트에서 사용 가능한 것으로 표시되지만 사용할 수 없을 수도 있습니다. 예를 들어, r5n 호스트에서 사용 가능한 vCPU가 16개로 표시될 수 있지만 r5n.4xlarge가 16개 vCPU에서 실행되더라도 호스트에서 4xlarge 인스턴스를 시작할 수 없는 경우가 있습니다.

> **Note**
>
> 여러 인스턴스 유형에 대한 A1 전용 호스트를 활성화하면 해당 호스트에서 `a1.xlarge` 및 `a1.2xlarge` 인스턴스의 혼합만 시작할 수 있습니다. 해당 호스트에서 `a1.medium` 또는 `a1.large` 인스턴스를 시작하는 경우 호스트에서 동일한 인스턴스 유형을 더 많이 시작하는 것으로만 제한됩니다. 단일 `a1.4xlarge` 인스턴스에서는 호스트의 모든 용량이 사용됩니다. `a1.medium` 또는 `a1.large` 인스턴스에 대한 호스트가 필요한 경우 해당 인스턴스 유형에 대해 별도의 호스트를 할당하는 것이 좋습니다.

Amazon EC2 전용 호스트의 버스트 가능 T3 인스턴스

전용 호스트는 버스트 가능한 성능 T3 인스턴스를 지원합니다. T3 인스턴스는 전용 하드웨어에서 적합한 BYOL 라이선스 소프트웨어를 비용 효율적으로 사용할 수 있는 방법을 제공합니다. T3 인스턴스의

vCPU 공간이 적기 때문에 더 적은 수의 호스트에서 워크로드를 통합하고 코어별 라이선스 사용률을 극대화할 수 있습니다.

T3 전용 호스트는 CPU 사용률이 낮거나 중간 정도인 BYOL 소프트웨어를 실행하는 데 가장 적합합니다. 여기에는 Windows Server, Windows 데스크톱, SQL Server, SUSE Enterprise Linux Server, Red Hat Enterprise Linux 및 Oracle Database와 같은 적합한 소켓당, 코어당 또는 VM당 소프트웨어 라이선스가 포함됩니다. T3 전용 호스트에 적합한 워크로드의 예로는 중소 규모의 데이터베이스, 가상 데스크톱, 개발 및 테스트 환경, 코드 리포지토리, 제품 프로토타입을 들 수 있습니다. T3 전용 호스트는 CPU 사용률이 지속적으로 높은 워크로드 또는 상호 연관된 CPU 버스트가 동시에 발생하는 워크로드에 대해서는 권장되지 않습니다.

전용 호스트의 T3 인스턴스는 공유 테넌시 하드웨어의 T3 인스턴스와 동일한 크레딧 모델을 사용합니다. 그러나, `standard` 크레딧 모드만 지원되며, `unlimited` 크레딧 모드는 지원되지 않습니다. `standard` 모드에서 전용 호스트의 T3 인스턴스는 공유 테넌시 하드웨어의 버스트 가능 인스턴스와 동일한 방식으로 크레딧을 획득, 지출 및 누적합니다. 버스트 가능 인스턴스는 기본 CPU 성능 외에 필요할 경우 기준 수준 이상으로 버스트할 수 있는 기능을 제공합니다. 기준 이상으로 버스트하려면 인스턴스는 CPU 크레딧 밸런스에 누적한 크레딧을 사용합니다. 누적된 크레딧이 고갈되면 CPU 사용률이 기준 수준으로 낮아집니다. `standard` 모드에 대한 자세한 내용은 스탠다드 성능 순간 확장 가능 인스턴스의 작동 방식 섹션을 참조하세요.

T3 전용 호스트는 단일 호스트의 여러 인스턴스 크기, 호스트 리소스 그룹 및 BYOL을 포함하여 Amazon EC2 전용 호스트에서 제공하는 모든 기능을 지원합니다.

지원되는 T3 인스턴스 크기 및 구성

T3 전용 호스트는 기본 CPU 성능과 필요 시 더 높은 수준으로 버스트할 수 있는 기능을 제공하여 호스트의 CPU 리소스를 공유하는 범용의 버스트 가능 T3 인스턴스를 실행합니다. 이를 통해 48개의 코어가 있는 T3 전용 호스트는 호스트당 최대 192개의 인스턴스를 지원할 수 있습니다. 호스트의 리소스를 효율적으로 활용하고 최상의 인스턴스 성능을 제공하기 위해 Amazon EC2 인스턴스 배치 알고리즘은 호스트에서 시작할 수 있는 지원되는 인스턴스 수와 인스턴스 크기 조합의 수를 자동으로 계산합니다.

T3 전용 호스트는 동일한 호스트에서 여러 인스턴스 유형을 지원합니다. 모든 T3 인스턴스 크기가 전용 호스트에서 지원됩니다. 호스트의 CPU 한도까지 다양한 T3 인스턴스 조합을 실행할 수 있습니다.

다음 테이블에서는 지원되는 인스턴스 유형을 나열하고 각 인스턴스 유형의 성능을 요약하며 시작할 수 있는 각 크기의 최대 인스턴스 수를 보여 줍니다.

인스턴스 유형	vCPU	메모리 (GiB)	vCPU당 기본 CPU 사용률	네트워크 버스트 대역폭 (Gbps)	Amazon EBS 버스트 대역폭 (Mbps)	전용 호스트당 최대 인스턴스 수
t3.na	2	0.5	5%	5	최대 2,085개	192
t3.m	2	1	10%	5	최대 2,085개	192
t3.sn	2	2	20%	5	최대 2,085개	192
t3.m	2	4	20%	5	최대 2,085개	192
t3.la	2	8	30%	5	2,780	96
t3.xl	4	16	40%	5	2,780	48
t3.2x e	8	32	40%	5	2,780	24

T3 전용 호스트의 CPU 사용률 모니터링

`DedicatedHostCPUUtilization` Amazon CloudWatch 지표를 사용하여 전용 호스트의 vCPU 사용률을 모니터링할 수 있습니다. 지표는 `EC2` 네임스페이스 및 `Per-Host-Metrics` 차원에서 사용할 수 있습니다. 자세한 내용은 전용 호스트 지표 섹션을 참조하세요.

Amazon EC2 전용 호스트에 기존 보유 소프트웨어 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. Microsoft 라이선싱에 대한 자세한 내용은 Amazon Web Services 및 Microsoft 라이선싱을 참조하세요.
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 참조하세요.
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(예: Windows Server 또는 Windows SQL Server)에 대해 해당 인스턴스를 사용해야 할 수 있습니다. 가져온 Windows AMI는 Amazon Windows KMS 서버에 대해 활성화할 수 없습니다.

> **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 AWS Config를 사용하여 Amazon EC2 전용 호스트 구성 변경 내용 추적 섹션을 참조하세요.

Amazon EC2 전용 호스트 자동 배치 및 호스트 선호도

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 미지정 호스트 테넌시 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 `Default`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).

계정에서 사용할 Amazon EC2 전용 호스트 할당

전용 호스트를 사용하려면 먼저 계정에서 호스트를 할당해야 합니다. 전용 호스트를 할당한 후에는 전용 호스트 용량을 계정에서 즉시 사용할 수 있으며 전용 호스트에서 인스턴스를 시작할 수 있습니다.

계정에 전용 호스트를 할당할 때 단일 인스턴스 유형 또는 동일한 인스턴스 패밀리 내의 여러 인스턴스 유형을 지원하는 구성을 선택할 수 있습니다. 호스트에서 실행할 수 있는 인스턴스 수는 선택한 구성에 따라 달라집니다. 자세한 내용은 Amazon EC2 전용 호스트 인스턴스 용량 구성을 참조하세요.

Console

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 호스트로 선택한 인스턴스 패밀리 내의 여러 인스턴스 크기를 지원할지 또는 특정 인스턴스 유형만 지원할지 여부를 지정합니다. 다음 중 하나를 수행하세요.

   - 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 서로 다른 인스턴스 크기를 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.
   - 선택한 인스턴스 패밀리 내의 단일 인스턴스 유형을 지원하도록 전용 호스트를 구성하려면 여러 인스턴스 유형 지원의 선택을 취소한 다음 인스턴스 유형에서 지원할 인스턴스 유형을 선택합니다. 이렇게 하면 전용 호스트에서 단일 인스턴스 유형을 시작할 수 있습니다.

예를 들어 이 옵션을 선택하고 m5.4xlarge를 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 Amazon EC2 전용 호스트 자동 배치 및 호스트 선호도 섹션을 참조하세요.
7. 전용 호스트의 호스트 복구를 사용하려면 Host recovery(호스트 복구)에서 활성화를 선택합니다. 자세한 내용은 Amazon EC2 전용 호스트 복구 섹션을 참조하세요.
8. 수량에 할당할 전용 호스트 수를 입력합니다.
9. (선택 사항) [새 태그 추가(Add new tag)]를 선택하고 태그 키와 태그 값을 입력합니다.
10. 할당을 선택합니다.

AWS CLI

전용 호스트를 할당하려면

allocate-hosts 명령을 사용합니다. 다음 예제에서는 m5 가용 영역에서 us-east-1a 인스턴스 패밀리의 여러 인스턴스 유형을 지원하는 전용 호스트를 할당합니다. 또한 호스트 복구를 활성화하고 자동 배치를 비활성화합니다.

```
aws ec2 allocate-hosts \
    --instance-family "m5" \
    --availability-zone "us-east-1a" \
    --auto-placement "off" \
    --host-recovery "on" \
    --quantity 1
```

다음 예제에서는 지정된 가용 영역에서 대상이 지정되지 않은 인스턴스 시작을 지원하는 전용 호스트를 할당하고, 호스트 복구를 활성화하고, 자동 배치를 활성화합니다.

```
aws ec2 allocate-hosts \
    --instance-type "m5.large" \
    --availability-zone "eu-west-1a" \
    --auto-placement "on" \
    --host-recovery "on" \
    --quantity 1
```

PowerShell

전용 호스트를 할당하려면

New-EC2Host cmdlet을 사용합니다. 다음 예제에서는 m5 가용 영역에서 us-east-1a 인스턴스 패밀리의 여러 인스턴스 유형을 지원하는 전용 호스트를 할당합니다. 또한 호스트에서 호스트 복구가 활성화되고 자동 배치가 비활성화됩니다.

```
New-EC2Host `
    -InstanceFamily m5 `
    -AvailabilityZone us-east-1a `
    -AutoPlacement Off `
    -HostRecovery On `
    -Quantity 1
```

다음 예제에서는 지정된 가용 영역에서 대상이 지정되지 않은 인스턴스 시작을 지원하는 전용 호스트를 할당하고, 호스트 복구를 활성화합니다.

```
New-EC2Host `
    -InstanceType m5.large `
    -AvailabilityZone eu-west-1a `
    -AutoPlacement On `
    -HostRecovery On `
    -Quantity 1
```

Amazon EC2 전용 호스트에서 Amazon EC2 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.

고려 사항

- SQL Server, SUSE 및 Amazon EC2에서 제공하는 RHEL AMI는 전용 호스트와 함께 사용할 수 없습니다.
- 여러 인스턴스 크기를 지원하는 전용 호스트의 경우 크기가 큰 인스턴스부터 시작한 다음 필요에 따라 더 작은 인스턴스로 나머지 인스턴스 용량을 채우는 것이 좋습니다.
- 인스턴스를 시작하기 전에 제한 사항에 유의하세요. 자세한 내용은 전용 호스트 제한 사항 섹션을 참조하세요.

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. Dedicated Hosts(전용 호스트) 페이지에서 호스트를 선택하고 Actions(작업), Launch Instance(s) onto host(호스트에서 인스턴스 시작)를 선택합니다.
4. Application and OS Images(애플리케이션 및 OS 이미지) 섹션의 목록에서 AMI를 선택합니다.
5. Instance type(인스턴스 유형) 섹션에서 시작할 인스턴스 유형을 선택합니다.

> **Note**
>
> 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. Key pair(키 페어) 섹션에서 인스턴스와 연결할 키 페어를 선택합니다.
7. 고급 세부 정보 섹션의 테넌시 선호도에서 다음 중 하나를 선택합니다.

- 끄기 - 호스트 선호도가 비활성화됩니다. 인스턴스가 지정된 호스트에서 시작하지만, 중지될 경우 반드시 동일한 전용 호스트에서 다시 시작하지는 않습니다.
- 전용 호스트 ID - 호스트 선호도가 활성화됩니다. 중지된 인스턴스는 호스트에 용량이 있는 경우 항상 이 지정된 호스트에서 다시 시작됩니다. 호스트에 용량이 없는 경우 인스턴스를 다시 시작할 수 없습니다. 다른 호스트와 선호도를 설정해야 합니다.

선호도에 대한 자세한 내용은 Amazon EC2 전용 호스트 자동 배치 및 호스트 선호도 섹션을 참조하세요.

> **Note**
>
> 테넌시 및 호스트 옵션은 선택한 호스트에 따라 사전 구성됩니다.

8. 필요에 따라 나머지 인스턴스 옵션을 구성합니다. 자세한 내용은 Amazon EC2 인스턴스 구성 파라미터에 대한 참조 섹션을 참조하세요.
9. 인스턴스 시작을 선택합니다.

인스턴스 시작 마법사를 사용하여 전용 호스트에서 인스턴스 시작

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances(인스턴스), Launch instance(인스턴스 시작)를 선택합니다.
3. Application and OS Images(애플리케이션 및 OS 이미지) 섹션의 목록에서 AMI를 선택합니다.
4. Instance type(인스턴스 유형) 섹션에서 시작할 인스턴스 유형을 선택합니다.
5. Key pair(키 페어) 섹션에서 인스턴스와 연결할 키 페어를 선택합니다.
6. Advanced details(고급 세부 정보) 섹션에서 다음을 수행합니다.
   a. Tenancy(테넌시)에서 Dedicated Host(전용 호스트)를 설정합니다.
   b. Target host by(대상 호스트)에서 Host ID(호스트 ID)를 선택합니다.
   c. Target host ID(대상 호스트 ID)에서 인스턴스를 시작할 호스트를 선택합니다.
   d. 테넌시 선호도에서 다음 중 하나를 선택합니다.
      - 끄기 - 호스트 선호도가 비활성화됩니다. 인스턴스가 지정된 호스트에서 시작하지만, 중지될 경우 반드시 동일한 전용 호스트에서 다시 시작하지는 않습니다.
      - 전용 호스트 ID - 호스트 선호도가 활성화됩니다. 중지된 인스턴스는 호스트에 용량이 있는 경우 항상 이 지정된 호스트에서 다시 시작됩니다. 호스트에 용량이 없는 경우 인스턴스를 다시 시작할 수 없습니다. 다른 호스트와 선호도를 설정해야 합니다.

      선호도에 대한 자세한 내용은 Amazon EC2 전용 호스트 자동 배치 및 호스트 선호도 섹션을 참조하세요.
7. 필요에 따라 나머지 인스턴스 옵션을 구성합니다. 자세한 내용은 Amazon EC2 인스턴스 구성 파라미터에 대한 참조 섹션을 참조하세요.
8. 인스턴스 시작을 선택합니다.

AWS CLI

전용 호스트에서 인스턴스를 시작하려면

run-instances 명령을 사용하고 `--placement` 옵션에서 인스턴스 선호도, 테넌시, 호스트를 지정합니다.

```
--placement Affinity=Host,Tenancy=dedicated,HostId=h-07879acf49EXAMPLE
```

PowerShell

전용 호스트에서 인스턴스를 시작하려면

New-EC2Instance cmdlet을 사용하고 `-Placement` 파라미터에서 인스턴스 선호도, 테넌시, 호스트를 지정합니다.

```
-Placement_Affinity Host `
-Placement_Tenancy dedicated `
-Placement_HostId h-07879acf49EXAMPLE
```

Amazon EC2 인스턴스를 호스트 리소스 그룹으로 인스턴스 시작

전용 호스트는 AWS License Manager와도 통합됩니다. License Manager를 사용하면 단일 개체로 관리되는 전용 호스트 모음인 호스트 Resource Groups를 만들 수 있습니다. 호스트 Resource Groups를 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 참조하세요.

가용 인스턴스 용량이 있는 전용 호스트가 포함된 호스트 Resource Groups에서 인스턴스를 시작하면 Amazon EC2가 그 호스트에서 인스턴스를 시작합니다. 가용 인스턴스 용량이 있는 호스트가 호스트 Resource Groups에 없으면 Amazon EC2가 호스트 Resource Groups에 새 호스트를 자동으로 할당하고 그 호스트에서 인스턴스를 시작합니다. 자세한 내용은 AWS License Manager 사용 설명서의 호스트 리소스 그룹을 참조하세요.

요구 사항 및 제한

- 코어 또는 소켓 기반 라이선스 구성을 AMI와 연결해야 합니다.
- Amazon EC2에서 제공하는 SQL Server, SUSE 및 RHEL AMI는 전용 호스트와 함께 사용할 수 없습니다.
- 호스트 ID를 선택하여 특정 호스트를 대상으로 지정할 수 없으며, 호스트 Resource Groups에서 인스턴스를 시작할 때 인스턴스 선호도를 활성화할 수 없습니다.

Console

호스트 Resource Groups에서 인스턴스를 시작하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances(인스턴스), Launch instance(인스턴스 시작)를 선택합니다.
3. Application and OS Images(애플리케이션 및 OS 이미지) 섹션의 목록에서 AMI를 선택합니다.
4. Instance type(인스턴스 유형) 섹션에서 시작할 인스턴스 유형을 선택합니다.
5. Key pair(키 페어) 섹션에서 인스턴스와 연결할 키 페어를 선택합니다.
6. Advanced details(고급 세부 정보) 섹션에서 다음을 수행합니다.
   a. Tenancy(테넌시)에서 Dedicated Host(전용 호스트)를 설정합니다.
   b. Target host by(대상 호스트)에서 Host resource group(호스트 리소스 그룹)을 선택합니다.
   c. Tenancy host resource group(테넌시 호스트 리소스 그룹)에서 인스턴스를 시작할 호스트 리소스 그룹을 선택합니다.
   d. Tenancy affinity(테넌시 선호도)에 대해 다음 중 하나를 수행합니다.
      - Off(끄기) - 인스턴스가 지정된 호스트에서 시작하지만, 중지될 경우 반드시 동일한 전용 호스트에서 다시 시작하지는 않습니다.
      - 전용 호스트 ID 선택 - 중지된 경우 인스턴스가 항상 특정 호스트에서 다시 시작합니다.

      선호도에 대한 자세한 내용은 Amazon EC2 전용 호스트 자동 배치 및 호스트 선호도 섹션을 참조하세요.
7. 필요에 따라 나머지 인스턴스 옵션을 구성합니다. 자세한 내용은 Amazon EC2 인스턴스 구성 파라미터에 대한 참조 섹션을 참조하세요.
8. 인스턴스 시작을 선택합니다.

AWS CLI

호스트 Resource Groups에서 인스턴스를 시작하려면

run-instances 명령을 사용합니다. `--placement` 옵션에서 테넌시를 생략하고 호스트 리소스 그룹의 ARN을 지정합니다.

```
--placement HostResourceGroupArn=arn:aws:resource-groups:us-
east-2:123456789012:group/my-resource-group
```

PowerShell

호스트 Resource Groups에서 인스턴스를 시작하려면

New-EC2Instance cmdlet을 사용합니다. `-Placement` 파라미터에서 테넌시를 생략하고 호스트 리소스 그룹의 ARN을 지정합니다.

```
-Placement_HostResourceGroupArn arn:aws:resource-groups:us-
east-2:123456789012:group/my-resource-group
```

기존 Amazon EC2 전용 호스트의 자동 배치 설정 수정

AWS 계정에 할당한 후 전용 호스트의 자동 배치 설정을 수정할 수 있습니다.

Console

전용 호스트의 자동 배치를 수정하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 전용 호스트(Dedicated Hosts)를 선택합니다.
3. 호스트를 선택하고 작업(Actions), 호스트 수정(Modify host)을 선택합니다.
4. 인스턴스 자동 배치(Instance auto-placement)에서 활성화(Enable)를 선택하여 자동 배치를 활성화하거나 활성화(Enable)를 선택 취소하여 자동 배치를 비활성화합니다. 자세한 내용은 Amazon EC2 전용 호스트 자동 배치 및 호스트 선호도 섹션을 참조하세요.
5. 저장(Save)을 선택합니다.

AWS CLI

전용 호스트의 자동 배치를 수정하려면

modify-hosts 명령을 사용합니다.

```
aws ec2 modify-hosts \
    --auto-placement on \
    --host-ids h-012a3456b7890cdef
```

PowerShell

전용 호스트의 자동 배치를 수정하려면

Edit-EC2Host cmdlet을 사용합니다.

```
Edit-EC2Host `
    -AutoPlacement 1 `
    -HostId h-012a3456b7890cdef
```

기존 Amazon EC2 전용 호스트에 대해 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.

여러 인스턴스 유형을 지원하는 전용 호스트를 단일 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.

가상화된 인스턴스 유형에 대해 호스트를 할당하는 경우 호스트가 할당된 후에는 인스턴스 유형을 `.metal` 인스턴스 유형으로 수정할 수 없습니다. 예를 들어 `m5.large` 인스턴스 유형에 대해 호스트를 할당하는 경우 인스턴스 유형을 `m5.metal`로 수정할 수 없습니다. 마찬가지로 `.metal` 인스턴스 유형에 대해 호스트를 할당하는 경우 호스트가 할당된 후에는 인스턴스 유형을 가상화된 인스턴스 유형으로 수정할 수 없습니다. 예를 들어 `m5.metal` 인스턴스 유형에 대해 호스트를 할당하는 경우 인스턴스 유형을 `m5.large`로 수정할 수 없습니다.

Console

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.

AWS CLI

전용 호스트에 지원되는 인스턴스 유형을 수정하려면

modify-hosts 명령을 사용합니다.

다음 예제에서는 m5 인스턴스 패밀리 내의 여러 인스턴스 유형을 지원하도록 전용 호스트를 수정합니다.

```
aws ec2 modify-hosts \
    --instance-family m5 \
    --host-ids h-012a3456b7890cdef
```

다음예제에서는 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 cmdlet을 사용합니다.

다음 예제에서는 m5 인스턴스 패밀리 내의 여러 인스턴스 유형을 지원하도록 전용 호스트를 수정합니다.

```
Edit-EC2Host `
    -InstanceFamily m5 `
    -HostId h-012a3456b7890cdef
```

다음예제에서는 m5.xlarge 인스턴스만 지원하도록 전용 호스트를 수정합니다.

```
Edit-EC2Host `
    -InstanceType m5.xlarge `
    -HostId h-012a3456b7890cdef
```

Amazon EC2 인스턴스의 Amazon EC2 전용 호스트 테넌시 및 선호도 수정

인스턴스를 시작한 후에 인스턴스의 테넌시를 변경할 수 있습니다. 또한 인스턴스에 대한 선호도를 수정하여 특정 호스트를 대상으로 하거나 계정에서 속성이 일치하는 사용 가능한 전용 호스트에서 실행되도록 할 수 있습니다. 인스턴스 테넌시 또는 선호도를 수정하려면 인스턴스가 stopped 상태여야 합니다.

인스턴스의 운영 체제 세부 정보 및 SQL Server 설치 여부는 지원되는 변환에 영향을 줍니다. 인스턴스에 사용할 수 있는 테넌시 변환 경로에 대한 자세한 내용은 License Manager 사용 설명서의 Tenancy conversion을 참조하세요.

> **Note**
>
> T3 인스턴스의 경우 전용 호스트에서 인스턴스를 시작해야 host의 테넌시를 사용할 수 있습니다. T3 인스턴스의 경우 테넌시를 host에서 dedicated 또는 default로 변경할 수 없습니다. 이러한 지원되지 않는 테넌시 변경 사항 중 하나를 만들려고 하면 InvalidRequest 오류 코드가 표시됩니다.

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. 인스턴스 상태, 중지를 차례로 선택합니다.
4. 인스턴스를 선택한 상태에서 작업, 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 Amazon EC2 전용 인스턴스 섹션을 참조하세요.
     - 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 Amazon EC2 전용 호스트 자동 배치 및 호스트 선호도 섹션을 참조하세요.
6. 저장을 선택합니다.

AWS CLI

인스턴스 테넌시 또는 선호도를 수정하려면

modify-instance-placement 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 \
    --tenancy host \
    --host-id h-012a3456b7890cdef
```

PowerShell

인스턴스 테넌시 또는 선호도를 수정하려면

Edit-EC2InstancePlacement cmdlet을 사용합니다. 다음 예는 지정된 인스턴스의 선호도를 `default`에서 `host`로 변경하고 인스턴스가 선호도를 갖는 전용 호스트를 지정합니다.

```
Edit-EC2InstancePlacement `
    -InstanceId i-1234567890abcdef0 `
    -Affinity host `
    -Tenancy host `
    -HostId h-012a3456b7890cdef
```

Amazon EC2 전용 호스트 릴리스

전용 호스트가 더 이상 필요하지 않을 경우 해당 호스트에서 실행 중인 인스턴스를 중지하고 다른 호스트에서 시작하도록 지시한 후 호스트를 릴리스합니다.

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.

Console

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.

AWS CLI

전용 호스트를 해제하려면

release-hosts 명령을 사용합니다.

```
aws ec2 release-hosts --host-ids h-012a3456b7890cdef
```

PowerShell

전용 호스트를 해제하려면

Remove-EC2Host cmdlet을 사용합니다.

```
Remove-EC2Host -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 보세요.

중지된 인스턴스는 계속 사용할 수 있으며 인스턴스 페이지에 나열됩니다. 또한 `host` 테넌시 설정을 유지합니다.

Nitro 기반 Amazon EC2 전용 호스트로 마이그레이션

Nitro 시스템은 우수한 성능과 고가용성, 철저한 보안을 지원하기 위해 AWS가 구축한 하드웨어 및 소프트웨어 구성 요소의 모음입니다. Nitro 기반 전용 호스트는 Xen 기반 전용 호스트에 비해 향상된 가격 대비 성능을 제공합니다. 계정에 Xen 기반 전용 호스트가 있는 경우 워크로드를 Nitro 기반 전용 호스트로 마이그레이션하는 것이 좋습니다. 자세한 내용은 AWS 니트로 시스템을 참조하세요.

Xen 기반 전용 호스트에서 Nitro 기반 전용 호스트로 마이그레이션하려면 전용 호스트의 Xen 기반 인스턴스를 Nitro 기반 인스턴스 유형으로 마이그레이션하고 새 Nitro 기반 전용 호스트를 할당한 다음 마이그레이션된 Nitro 기반 인스턴스를 새 Nitro 기반 전용 호스트로 이동해야 합니다.

이 주제에서는 Xen 기반 전용 호스트에서 Nitro 기반 전용 호스트로 마이그레이션하는 자세한 단계를 제공합니다.

마이그레이션 단계

- 1단계: Xen 기반 전용 호스트 식별
- 2단계: Xen 기반 인스턴스를 Nitro 기반 인스턴스 유형으로 마이그레이션
- 3단계: Nitro 기반 전용 호스트 할당
- 4단계: 마이그레이션된 인스턴스를 새 Nitro 기반 전용 호스트로 이동
- 5단계: 미사용 Xen 기반 전용 호스트 해제

1단계: Xen 기반 전용 호스트 식별

다음 전용 호스트는 Xen 기반이며 Nitro 기반 전용 호스트로 마이그레이션할 수 있습니다.

- 범용: M3 | M4
- 컴퓨팅 최적화: C3 | C4
- 메모리 최적화: R3 | R4 | X1 | X1e
- 스토리지 최적화: D2 | H1 | I2 | I3
- 가속화된 컴퓨팅: F1 | G3 | P2 | P3

계정에 Xen 기반 전용 호스트가 있는지 확인하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 패널에서 전용 호스트를 선택합니다.
3. 검색 필드에서 인스턴스 패밀리 필터를 사용하여 위의 Xen 기반 전용 호스트를 검색합니다. 예를 들어 인스턴스 패밀리 = m3입니다.

2단계: Xen 기반 인스턴스를 Nitro 기반 인스턴스 유형으로 마이그레이션

Xen 기반 전용 호스트에서 실행되는 인스턴스도 Xen 기반입니다. 이러한 인스턴스를 Nitro 기반 전용 호스트로 이동하려면 먼저 Nitro 기반 인스턴스 유형으로 마이그레이션해야 합니다.

> **Important**
>
> 인스턴스 마이그레이션을 시작하기 전에 데이터를 백업하는 것이 좋습니다. 자세한 내용은 Amazon EC2 인스턴스에서 다중 볼륨 Amazon EBS 스냅샷 생성을 참조하세요.

Xen 기반 전용 호스트에서 실행되는 인스턴스를 찾으려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 패널에서 전용 호스트를 선택합니다.
3. 마이그레이션하려는 Xen 기반 호스트를 선택한 다음 실행 중인 인스턴스 탭을 선택합니다. 탭에 선택한 호스트에서 실행되는 모든 인스턴스가 나열됩니다.

Linux 인스턴스를 마이그레이션하려면 Amazon EC2 인스턴스 유형 변경 섹션을 참조하세요.

Windows 인스턴스를 마이그레이션하려면 EC2 Windows 인스턴스를 Nitro 기반 인스턴스 유형으로 마이그레이션 섹션을 참조하세요.

> **Note**
>
> 인스턴스를 마이그레이션하려는 Nitro 기반 전용 호스트와 일치하는 인스턴스 유형으로 마이그레이션해야 합니다. 예를 들어 M7i 전용 호스트로 마이그레이션하려는 경우 인스턴스를 M7i 인스턴스 유형으로 마이그레이션해야 합니다.

3단계: Nitro 기반 전용 호스트 할당

지원되는 Nitro 기반 전용 호스트를 찾으려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스 유형을 선택합니다.
3. 다음 필터를 적용합니다.
   - 하이퍼바이저 = nitro
   - 전용 호스트 지원 = true

적합한 Nitro 기반 인스턴스 유형을 찾은 후 새 전용 호스트를 할당합니다.

4단계: 마이그레이션된 인스턴스를 새 Nitro 기반 전용 호스트로 이동

Nitro 기반 전용 호스트를 할당하고 전용 호스트가 `available` 상태가 되면 이전에 Nitro 기반 인스턴스 유형으로 마이그레이션한 인스턴스를 새 전용 호스트로 이동할 수 있습니다.

인스턴스를 새 Nitro 기반 전용 호스트로 이동하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 패널에서 인스턴스를 선택합니다.
3. 마이그레이션한 인스턴스를 선택한 다음 작업, 인스턴스 설정, 인스턴스 배치 수정을 차례로 선택합니다.
4. 대상 전용 호스트에서 새 Nitro 기반 전용 호스트를 선택한 다음 저장을 선택합니다.
5. 인스턴스를 다시 시작합니다. 인스턴스를 선택하고 인스턴스 상태, 인스턴스 시작을 차례로 선택합니다.

5단계: 미사용 Xen 기반 전용 호스트 해제

Xen 기반 전용 호스트에서 새 Nitro 기반 전용 호스트로 워크로드를 마이그레이션한 후 더 이상 필요하지 않은 경우 Xen 기반 전용 호스트를 해제할 수 있습니다.

전용 호스트 청구 할인을 위한 전용 호스트 예약 구매

전용 호스트 예약은 온디맨드 전용 호스트 요금에 비해 최대 70% 할인된 요금이 적용됩니다. 계정에 할당된 활성 전용 호스트가 있어야 전용 호스트 예약을 구매할 수 있습니다. 자세한 내용은 Dedicated Host Reservations 섹션을 참조하세요.

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. 제품 및 서비스 찾기 화면에서 다음을 수행합니다.
   a. 인스턴스 패밀리에서 전용 호스트 예약을 구매할 전용 호스트의 인스턴스 패밀리를 선택합니다.
   b. 결제 옵션에서 원하는 결제 옵션을 선택하고 구성합니다.
4. 다음을 선택합니다.
5. 전용 호스트 예약을 연결할 전용 호스트를 선택하고 다음을 선택합니다.
6. (선택 사항) 전용 호스트 예약에 태그를 할당합니다.
7. 주문을 검토한 후 구입을 선택합니다.

AWS CLI

예약을 구매하려면

1. describe-host-reservation-offerings 명령을 사용하여 요구 사항에 맞는 사용 가능한 상품 목록을 나열합니다. 다음 예는 m4 인스턴스 패밀리의 인스턴스를 지원하고 사용 기간이 1년인 상품 목록을 나열합니다.

   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. 구매할 상품의 ID를 기록합니다.

2. purchase-host-reservation 명령을 사용하여 상품을 구입하고 이전 단계에서 기록한 `offeringId`를 입력합니다. 다음 예제에서는 지정된 예약을 구매하여 이미 AWS 계정에 할당된 특정 전용 호스트와 연결하며 키가 `purpose`이고 값이 `production`인 태그를 적용합니다.

```
aws ec2 purchase-host-reservation \
    --offering-id hro-03f707bf363b6b324 \
    --host-id-set h-013abcd2a00cbd123 \
    --tag-specifications 'ResourceType=host-
reservation,Tags={Key=purpose,Value=production}'
```

PowerShell

예약을 구매하려면

1. Get-EC2HostReservationOffering cmdlet을 사용하여 요구 사항에 맞는 사용 가능한 상품 목록을 나열합니다. 다음 예는 `m5` 인스턴스 패밀리의 인스턴스를 지원하고 사용 기간이 1년인 상품 목록을 나열합니다.

   기간은 초 단위로 지정됩니다. 1년 기간에는 31,536,000초가 포함되고, 3년 기간에는 94,608,000초가 포함됩니다.

```
$filter = @{Name="instance-family"; Values="m5"}
Get-EC2HostReservationOffering `
    -Filter $filter `
    -MaxDuration 31536000
```

   이 명령은 조건과 일치하는 상품 목록을 반환합니다. 구매할 상품의 ID를 기록합니다.

2. New-EC2HostReservation cmdlet을 사용하여 상품을 구매하고 이전 단계에서 기록한 상품 ID를 입력합니다. 다음 예제에서는 특정 예약을 구매하고 이미 AWS 계정에 할당된 특정 전용 호스트와 연결합니다.

```
New-EC2HostReservation `
```

```
    -OfferingId hro-03f707bf363b6b324 `
    -HostIdSet h-013abcd2a00cbd123
```

크로스 계정 Amazon EC2 전용 호스트 공유

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
- 전용 호스트 공유 제한 사항
- 관련 서비스
- 여러 가용 영역에서 공유
- 공유된 전용 호스트 권한
- 결제 및 측정
- 전용 호스트 제한
- 호스트 복구 및 전용 호스트 공유
- Amazon EC2 전용 호스트를 여러 AWS 계정 간에 공유
- 다른 AWS 계정과 공유되는 전용 호스트의 공유 해제
- AWS 계정에서 공유된 Amazon EC2 전용 호스트 보기

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 Organizations를 통해 공유를 사용하도록 설정해야 합니다. 자세한 내용은 AWS RAM 사용 설명서에서 AWS Organizations를 사용하여 공유 사용을 참조하세요.

전용 호스트 공유 제한 사항

`u-6tb1.metal`, `u-9tb1.metal`, `u-12tb1.metal`, `u-18tb1.metal` 및 `u-24tb1.metal` 인스턴스 유형에 할당된 전용 호스트는 공유할 수 없습니다.

관련 서비스

AWS Resource Access Manager

전용 호스트 공유는 AWS Resource Access Manager(AWS RAM)와 통합됩니다. AWS RAM은 모든 AWS 계정 또는 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 참조하세요.

여러 가용 영역에서 공유

리전의 가용 영역에 걸쳐 리소스가 배포될 수 있도록 각 계정의 이름에 가용 영역을 독립적으로 매핑합니다. 이로 인해 계정 전체에서 가용 영역 이름의 차이가 발생할 수 있습니다. 예를 들어 AWS 계정의 `us-east-1a` 가용 영역은 다른 AWS 계정에 대한 `us-east-1a`로 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/ram/homeAWS RAM에서 콘솔을 엽니다.
2. 현재 리전의 가용 영역 ID는 화면 오른쪽에 있는 Your AZ ID(AZ ID) 패널에 표시됩니다.

공유된 전용 호스트 권한

소유자에 대한 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.

소비자에 대한 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 Amazon EC2 전용 호스트 복구 섹션을 참조하세요.

Amazon EC2 전용 호스트를 여러 AWS 계정 간에 공유

소유자가 전용 호스트를 공유하면 소비자가 호스트에서 인스턴스를 시작할 수 있게 됩니다. 소비자는 가용 용량이 허용하는 만큼 많은 인스턴스를 공유된 호스트에서 시작할 수 있습니다.

> **Important**
>
> 전용 호스트에서 BYOL 라이선스를 공유할 수 있는 적절한 라이선스 권리가 있는지 확인할 책임은 사용자에게 있습니다.

자동 배치가 활성화된 상태로 전용 호스트를 공유하는 경우 의도하지 않은 전용 호스트 사용으로 이어질 수 있으므로 다음 내용에 유의하세요.

- 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. 리소스 공유는 여러 AWS 계정에서 리소스를 공유할 수 있게 해주는 AWS RAM 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 내에서 공유가 사용되는 경우 공유 전용 호스트에 대한 액세스 권한이 조직의 소비자에게 자동으로 부여됩니다. 그렇게 하지 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.

Console

Amazon EC2 콘솔을 사용하여 소유하고 있는 전용 호스트를 공유하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 전용 호스트를 선택합니다.
3. 공유할 전용 호스트를 선택하고 작업, 호스트 공유를 선택합니다.
4. 전용 호스트를 추가할 리소스 공유를 선택하고 호스트 공유를 선택합니다.

   소비자가 공유 호스트에 액세스하려면 몇 분이 걸릴 수 있습니다.

AWS RAM 콘솔을 사용하여 소유하고 있는 전용 호스트를 공유하려면

AWS RAM 사용 설명서의 리소스 공유 생성을 참조하세요.

AWS CLI

소유하고 있는 전용 호스트를 공유하려면 다음을 수행하세요.

create-resource-share 명령을 사용합니다.

```
aws ram create-resource-share \
    --name my-resource-share \
    --resource-arns arn:aws:ec2:us-east-2:123456789012:dedicated-
host/h-07879acf49EXAMPLE
```

PowerShell

소유하고 있는 전용 호스트를 공유하려면 다음을 수행하세요.

New-RAMResourceShare cmdlet을 사용합니다.

```
New-RAMResourceShare `
    -Name my-resource-share `
    -ResourceArn arn:aws:ec2:us-east-2:123456789012:dedicated-
host/h-07879acf49EXAMPLE
```

다른 AWS 계정과 공유되는 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.

Console

소유하고 있는 공유 전용 호스트를 공유 해제하려면 다음을 수행하세요.

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

2. 탐색 창에서 전용 호스트를 선택합니다.
3. 공유를 해제할 전용 호스트를 선택하고 공유 탭을 선택합니다.
4. 공유 탭에는 전용 호스트가 추가된 리소스 공유가 나열됩니다. 전용 호스트를 제거할 리소스 공유를 선택하고 리소스 공유에서 호스트 제거를 선택합니다.

AWS RAM 콘솔을 사용하여 소유하고 있는 공유 전용 호스트를 공유 해제하려면

AWS RAM 사용 설명서의 리소스 공유 업데이트를 참조하세요.

AWS CLI

소유하고 있는 공유 전용 호스트를 공유 해제하려면 다음을 수행하세요.

disassociate-resource-share 명령을 사용합니다.

```
aws ram disassociate-resource-share \
    --resource-share-arn arn:aws:ram:us-east-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE \
 --resource-arns arn:aws:ec2:us-east-2:123456789012:dedicated-host/h-07879acf49EXAMPLE
```

PowerShell

소유하고 있는 공유 전용 호스트를 공유 해제하려면 다음을 수행하세요.

Disconnect-RAMResourceShare cmdlet을 사용합니다.

```
Disconnect-RAMResourceShare `
    -ResourceShareArn "arn:aws:ram:us-east-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE" `
    -ResourceArn "arn:aws:ec2:us-east-2:123456789012:dedicated-host/h-07879acf49EXAMPLE"
```

AWS 계정에서 공유된 Amazon EC2 전용 호스트 보기

다른 계정과 공유 중인 전용 호스트 및 자신의 계정과 공유된 전용 호스트를 볼 수 있습니다. 전용 호스트를 소유하는 경우 소비자가 시작한 인스턴스를 포함하여 해당 호스트에서 실행되는 인스턴스를 모두 볼 수 있습니다. 전용 호스트가 공유되는 경우 공유된 호스트에서 자신이 시작한 인스턴스만 볼 수 있고 다른 소비자가 시작한 인스턴스는 볼 수 없습니다.

소유자와 소비자는 다음 방법 중 하나를 사용하여 공유 전용 호스트를 식별할 수 있습니다.

Console

공유 전용 호스트를 식별하려면 다음을 수행하세요.

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.
3. [소유자(Owner)] 열에 전용 호스트 소유자의 AWS 계정 ID가 표시됩니다.
4. 호스트에서 실행 중인 인스턴스를 보려면 인스턴스 탭을 선택합니다.

AWS CLI

공유 전용 호스트를 식별하려면 다음을 수행하세요.

describe-hosts 명령을 사용합니다. 이 명령은 소유하고 있는 전용 호스트 및 자신과 공유된 전용 호스트를 반환합니다. `Owner` 값은 전용 호스트 소유자의 계정 ID입니다. `Instances` 목록은 호스트에서 실행 중인 인스턴스를 설명합니다.

```
aws ec2 describe-hosts --filter "Name=state,Values=available"
```

PowerShell

공유 전용 호스트를 식별하려면 다음을 수행하세요.

Get-EC2host cmdlet을 사용합니다. cmdlet은 소유하고 있는 전용 호스트 및 자신과 공유된 전용 호스트를 반환합니다. 응답의 `Owner` 값은 전용 호스트 소유자의 계정 ID입니다. `Instances` 목록은 호스트에서 실행 중인 인스턴스를 설명합니다.

```
Get-EC2Host -Filter @{Name="state"; Values="available"}
```

AWS Outposts의 Amazon EC2 전용 호스트

AWS Outposts는 AWS 인프라, 서비스, API 및 도구를 온프레미스로 확장하는 완전관리형 서비스입니다. AWS 관리형 인프라에 대한 로컬 액세스를 제공하는 AWS Outposts를 통해 AWS 리전에서 사용하는 것과 동일한 프로그래밍 인터페이스를 사용해 온프레미스에서 애플리케이션을 구축하고 실행할 수

있으며, 짧은 지연 시간과 로컬 데이터 처리가 필요한 경우에 로컬 컴퓨팅 및 스토리지 리소스를 사용할 수 있습니다.

Outpost는 고객 사이트에 배포된 AWS 컴퓨팅 및 스토리지 용량 풀입니다. AWS는 이 용량을 AWS 리전의 일부로 운영, 모니터링 및 관리합니다.

계정에서 소유한 Outposts에 전용 호스트를 할당할 수 있습니다. 따라서 전용 물리적 서버가 필요한 기존 소프트웨어 라이선스와 워크로드를 AWS Outposts로 더 쉽게 가져올 수 있습니다. 또한 Outpost의 특정 하드웨어 자산을 대상으로 지정하여 워크로드 간 지연 시간을 최소화할 수 있습니다.

전용 호스트를 사용하면 Amazon EC2에서 적격 소프트웨어 라이선스를 사용할 수 있으므로 자체 라이선스를 사용할 때의 유연성과 비용 효율성을 얻을 수 있습니다. 가상 머신, 소켓 또는 물리적 코어에 바인딩된 기타 소프트웨어 라이센스는 해당 라이선스 조건에 따라 전용 호스트에서도 사용할 수 있습니다. Outposts는 항상 BYOL 워크로드에 적합한 단일 테넌트 환경이었지만 전용 호스트를 사용하면 전체 Outpost 배포와 달리 필요한 라이선스를 단일 호스트로 제한할 수 있습니다.

또한 Outpost에서 전용 호스트를 사용하면 인스턴스 유형 배포의 유연성이 향상되고 인스턴스 배치를 보다 세부적으로 제어할 수 있습니다. 인스턴스 시작을 위해 특정 호스트를 대상으로 지정하고 호스트 선호도를 사용하여 인스턴스가 항상 해당 호스트에서 실행되도록 하거나 자동 배치를 사용하여 구성과 사용 가능한 용량이 일치하는 사용 가능한 호스트에서 인스턴스를 시작할 수 있습니다.

목차

- 사전 조건
- 지원 기능
- 고려 사항
- AWS Outposts에서 Amazon EC2 전용 호스트 할당

사전 조건

사이트에 Outposts가 설치되어 있어야 합니다. 자세한 내용은 AWS Outposts 사용 설명서에서 Outposts 생성 및 Outposts 용량 주문을 참조하세요.

지원 기능

- 지원되는 인스턴스 패밀리는 다음과 같습니다. `C5`, `M5`, `R5`, `C5d`, `M5d`, `R5d`, `G4dn`, `i3en`
- Outposts의 전용 호스트는 여러 인스턴스 크기를 지원하도록 구성할 수 있습니다. `C5`, `M5`, `R5`, `C5d`, `M5d`, `R5d` 인스턴스 패밀리의 경우 여러 인스턴스 크기가 지원됩니다. 자세한 내용은 Amazon EC2 전용 호스트 인스턴스 용량 구성 섹션을 참조하세요.

- Outposts의 전용 호스트는 자동 배치와 대상 인스턴스 실행을 지원합니다. 자세한 내용은 Amazon EC2 전용 호스트 자동 배치 및 호스트 선호도 섹션을 참조하세요.
- Outposts의 전용 호스트는 호스트 선호도를 지원합니다. 자세한 내용은 Amazon EC2 전용 호스트 자동 배치 및 호스트 선호도 섹션을 참조하세요.
- Outposts의 전용 호스트는 AWS RAM과의 공유를 지원합니다. 자세한 내용은 크로스 계정 Amazon EC2 전용 호스트 공유 섹션을 참조하세요.

고려 사항

- 전용 호스트 예약은 Outposts에서 지원되지 않습니다.
- 호스트 리소스 그룹과 AWS License Manager는 Outposts에서 지원되지 않습니다.
- Outposts의 전용 호스트는 버스트 가능한 T3 인스턴스를 지원하지 않습니다.
- Outposts의 전용 호스트는 호스트 복구를 지원하지 않습니다.
- Outposts에서 전용 호스트 테넌시가 포함된 인스턴스에서는 간소화된 자동 복구가 지원되지 않습니다.

AWS Outposts에서 Amazon EC2 전용 호스트 할당

AWS 리전의 전용 호스트와 동일한 방식으로 Outposts에 전용 호스트를 할당하고 사용합니다.

사전 조건

Outposts에서 서브넷을 생성합니다. 자세한 내용은 AWS Outposts 사용 설명서에서 서브넷 생성을 참조하세요.

Outpost에 전용 호스트를 할당하려면 다음 방법 중 하나를 사용합니다.

Console

AWS Outposts 콘솔을 사용하여 Outpost에 전용 호스트를 할당하는 방법

1. https://console.aws.amazon.com/outposts/에서 AWS Outposts 콘솔을 엽니다.
2. 탐색 창에서 Outposts를 선택합니다. Outpost를 선택한 다음 작업(Actions), 전용 호스트 할당(Allocate Dedicated Host)을 선택합니다.
3. 필요에 따라 전용 호스트를 구성합니다. 자세한 내용은 계정에서 사용할 Amazon EC2 전용 호스트 할당 섹션을 참조하세요.

> **Note**
>
> 가용 영역(Availability Zone) 및 Outpost ARN은 가용 영역과 선택한 Outpost의 ARN으로 미리 채워져야 합니다.

4. 할당을 선택합니다.

Amazon EC2 콘솔을 사용하여 Outpost에 전용 호스트를 할당하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 전용 호스트(Dedicated Hosts)를 선택한 후 전용 호스트 할당(Allocate Dedicated Host)을 선택합니다.
3. 가용 영역(Availability Zone)에서 Outpost와 연결된 가용 영역을 선택합니다.
4. Outpost ARN에 Outpost의 ARN을 입력합니다.
5. Outpost의 특정 하드웨어 자산을 대상으로 지정하려면 Outpost의 특정 하드웨어 자산 대상 지정에서 활성화를 선택합니다. 대상으로 지정할 각 하드웨어 자산에 대해 자산 ID 추가를 선택한 다음 하드웨어 자산의 ID를 입력합니다.

> **Note**
>
> 수량에 지정하는 값은 지정한 자산 ID의 수와 같아야 합니다. 예를 들어 자산 ID 3개를 지정한 경우 수량도 3이어야 합니다.

6. 필요에 따라 나머지 전용 호스트 설정을 구성합니다. 자세한 내용은 계정에서 사용할 Amazon EC2 전용 호스트 할당 섹션을 참조하세요.
7. 할당을 선택합니다.

AWS CLI

Outpost에 전용 호스트를 할당하는 방법

allocate-hosts 명령을 사용합니다. `--availability-zone`에서 Outpost와 연결된 가용 영역을 지정합니다. `--outpost-arn`에서 Outpost의 ARN을 지정합니다. 선택적으로 `--asset-ids`의 경우 대상으로 지정할 Outpost 하드웨어 자산의 ID를 지정합니다.

```
aws ec2 allocate-hosts \
```

```
    --availability-zone "us-east-1a" \
    --outpost-arn "arn:aws:outposts:us-east-1a:111122223333:outpost/
op-4fe3dc21baEXAMPLE" \
    --asset-ids asset_id \
    --instance-family "m5" \
    --auto-placement "off" \
    --quantity 1
```

PowerShell

Outpost에 전용 호스트를 할당하는 방법

New-EC2Host cmdlet을 사용합니다. Outpost와 연결된 가용 영역을 지정합니다. 선택적으로 -AssetId의 경우 대상으로 지정할 Outpost 하드웨어 자산의 ID를 지정합니다.

```
New-EC2Host `
    -AvailabilityZone "us-east-1a" `
    -OutpostArn "arn:aws:outposts:us-east-1a:111122223333:outpost/
op-4fe3dc21baEXAMPLE" `
    -AssetId asset_id `
    -InstanceFamily "m5" `
    -AutoPlacement "off" `
    -Quantity 1
```

Outpost에서 전용 호스트로 인스턴스 시작

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 전용 호스트(Dedicated Hosts)를 선택합니다. 이전 단계에서 할당한 전용 호스트를 선택하고 작업(Actions), 호스트에서 인스턴스 시작(Launch instance onto host)을 선택합니다.
3. 필요에 따라 인스턴스를 구성한 다음 인스턴스를 시작합니다. 자세한 내용은 Amazon EC2 전용 호스트에서 Amazon EC2 인스턴스 시작 섹션을 참조하세요.

Amazon EC2 전용 호스트 복구

전용 호스트 자동 복구는 전용 호스트에서 특정한 오류 상태가 감지되면 새 대체 호스트로 인스턴스를 다시 시작합니다. 호스트 복구는 수동 개입의 필요성을 줄이고 시스템 파워 또는 네트워크 연결 이벤트에서 예기치 않은 전용 호스트 장애가 발생하는 경우 운영 부담을 줄입니다. 기타 전용 호스트 문제는 복구를 위해 수동 개입이 필요합니다.

내용

- Amazon EC2 전용 호스트 복구 작동 방식
- 지원되는 인스턴스 유형
- 요금
- Amazon EC2 전용 호스트 복구 관리
- Amazon EC2 전용 호스트에 대한 호스트 복구 설정 보기
- Amazon EC2 전용 호스트 복구에서 지원되지 않는 인스턴스를 수동으로 복구

Amazon EC2 전용 호스트 복구 작동 방식

전용 호스트 미 호스트 리소스 그룹 복구 프로세스는 호스트 수준 상태 확인을 사용하여 전용 호스트 가용성을 평가하고 기본 시스템 오류를 감지합니다. 전용 호스트 장애 유형에 따라 전용 호스트 자동 복구가 가능한지 여부가 결정됩니다. 시스템 상태 확인이 실패할 수 있는 문제의 예를 들면 다음과 같습니다.

- 네트워크 연결 끊김
- 시스템 전원 중단
- 물리적 호스트의 하드웨어 또는 소프트웨어 문제

> **Important**
>
> 호스트가 사용 중지되도록 예약된 경우 전용 호스트 자동 복구가 수행되지 않습니다.

전용 호스트 자동 복구

전용 호스트에서 시스템 전원 또는 네트워크 연결 장애가 감지되면 전용 호스트 자동 복구가 시작되고 Amazon EC2가 자동으로 원본 전용 호스트와 동일한 가용 영역에서 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
- 예약

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
- 탄력적인 IP 주소
- EBS 볼륨 장치
- 모든 인스턴스 메타데이터

또한 내장된 AWS License Manager 통합 기능은 라이선스 추적 및 관리를 자동화합니다.

> **Note**
>
> AWS License Manager 통합은 AWS License Manager가 제공되는 리전에서만 지원됩니다.

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 Health Dashboard 이벤트로 알림이 전송됩니다. 호스트 복구가 성공적으로 완료되면 두 번째 알림이 전송됩니다.

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(AWS License Manager에서 알림 설정을 구성한 경우). 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 License Manager의 라이선스 구성 및 License Manager 설정을 참조하세요.

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 저장소 지원 루트 볼륨이 있는 인스턴스 등 호스트 복구를 지원하지 않는 인스턴스가 있는 경우 전용 호스트가 해제되지 않습니다. 대신, 만료 표시되고 permanent-failure 상태가 됩니다.

전용 호스트 자동 복구가 없는 시나리오

호스트가 사용 중지되도록 예약된 경우 전용 호스트 자동 복구가 수행되지 않습니다. Amazon CloudWatch 이벤트인 AWS Health Dashboard에서 사용 중지 알림을 받게 되고 AWS 계정 소유자 이메일 주소에 전용 호스트 실패와 관련된 메시지가 전송됩니다. 사용 중지되는 호스트에서 인스턴스를 수동으로 복구하려면 지정된 기간 내에 만료 알림에 설명된 해결 단계를 수행합니다.

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.

지원되는 인스턴스 유형

호스트 복구는 A1, C3, C4, C5, C5n, C6a, C6g, C6i, Inf1, G3, G5g, M3, M4, M5, M5n, M5zn, M6a, M6g, M6i, P2, P3, R3, R4, R5, R5b, R5n, R6g, R6i, T3, X1, X1e, X2iezn, u-6tb1, u-9tb1, u-12tb1, u-18tb1, u-24tb1 인스턴스 패밀리에 대해 지원됩니다.

지원되지 않는 인스턴스를 복구하려면 Amazon EC2 전용 호스트 복구에서 지원되지 않는 인스턴스를 수동으로 복구 섹션을 참조하세요.

> **Note**
>
> 지원되는 메탈 인스턴스 유형의 전용 호스트 자동 복구는 비 메탈 인스턴스 유형보다 감지하고 복구하는 데 시간이 오래 걸립니다.

요금

호스트 복구 사용에 대한 추가 요금은 없지만 일반적으로 전용 호스트 요금이 적용됩니다. 자세한 내용은 Amazon EC2 전용 호스트 요금을 참조하세요.

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.

Amazon EC2 전용 호스트 복구 관리

전용 호스트 자동 복구는 전용 호스트에서 특정한 오류 상태가 감지되면 새 대체 호스트로 인스턴스를 다시 시작합니다. 전용 호스트 할당 시 또는 할당 후 호스트 복구를 활성화할 수 있습니다.

다음 절차에 따라 호스트 할당 시 호스트 복구를 활성화합니다.

Console

할당 시 호스트 복구를 활성화하려면 다음을 수행하세요.

Amazon EC2 콘솔을 사용하여 전용 호스트를 할당할 때 호스트 복구에 대해 활성화를 선택합니다. 자세한 내용은 계정에서 사용할 Amazon EC2 전용 호스트 할당 섹션을 참조하세요.

AWS CLI

할당 시 호스트 복구를 활성화하려면 다음을 수행하세요.

allocate-hosts 명령을 사용합니다.

```
aws ec2 allocate-hosts \
    --instance-type m5.large \
    --availability-zone eu-west-1a \
```

```
    --auto-placement on \
    --host-recovery on \
    --quantity 1
```

PowerShell

할당 시 호스트 복구를 활성화하려면 다음을 수행하세요.

New-EC2Host cmdlet을 사용합니다.

```
New-EC2Host `
    -InstanceType m5.large `
    -AvailabilityZone eu-west-1a `
    -AutoPlacement on `
    -HostRecovery on `
    -Quantity 1
```

다음 절차에 따라 전용 호스트의 호스트 복구를 관리합니다.

Console

할당 후 호스트 복구를 관리하려면 다음을 수행하세요.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 전용 호스트(Dedicated Hosts)를 선택합니다.
3. 전용 호스트를 선택합니다.
4. 작업, 호스트 수정을 선택합니다.
5. 호스트 복구에 대해 활성화를 선택하거나 선택을 취소합니다.
6. 저장을 선택합니다.

AWS CLI

할당 후 호스트 복구를 활성화하려면 다음을 수행하세요.

modify-hosts 명령을 사용합니다.

```
aws ec2 modify-hosts \
    --host-recovery on \
```

```
    --host-ids h-012a3456b7890cdef
```

할당 후 호스트 복구를 비활성화하려면 다음을 수행하세요.

modify-hosts 명령을 사용하고 `host-recovery` 파라미터를 `off` 값으로 지정합니다.

```
aws ec2 modify-hosts \
    --host-recovery off \
    --host-ids h-012a3456b7890cdef
```

PowerShell

할당 후 호스트 복구를 활성화하려면 다음을 수행하세요.

Edit-host cmdlet을 사용합니다.

```
Edit-EC2Host `
    -HostRecovery on `
    -HostId h-012a3456b7890cdef
```

할당 후 호스트 복구를 비활성화하려면 다음을 수행하세요.

Edit-EC2Host cmdlet을 사용합니다.

```
Edit-EC2Host `
    -HostRecovery off `
    -HostId h-012a3456b7890cdef
```

Amazon EC2 전용 호스트에 대한 호스트 복구 설정 보기

언제든지 전용 호스트에 대한 호스트 복구 구성을 볼 수 있습니다.

Console

전용 호스트에 대한 호스트 복구 구성을 보는 방법

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.

AWS CLI

전용 호스트에 대한 호스트 복구 구성을 보는 방법

describe-hosts 명령을 사용합니다.

```
aws ec2 describe-hosts \
    --host-ids h-012a3456b7890cdef \
    --query Hosts[].HostRecovery
```

출력의 예시는 다음과 같습니다.

```
on
```

PowerShell

전용 호스트에 대한 호스트 복구 구성을 보는 방법

Get-EC2Host cmdlet을 사용합니다.

```
(Get-EC2Host -HostId h-012a3456b7890cdef).Hosts | Select HostRecovery
```

출력의 예시는 다음과 같습니다.

```
HostRecovery
------------
on
```

Amazon EC2 전용 호스트 복구에서 지원되지 않는 인스턴스를 수동으로 복구

호스트 복구는 인스턴스 스토어 볼륨을 사용하는 인스턴스 복구를 지원하지 않습니다. 자동으로 복구할 수 없는 인스턴스를 수동으로 복구하려면 아래 지침을 따릅니다.

> **Warning**
>
> 인스턴스가 중단되거나 최대 절전 모드로 전환되거나 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 백업하세요.

EBS-backed 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 Amazon EC2 인스턴스 중지 및 시작 섹션을 참조하세요.

인스턴스 스토어 기반 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.

Amazon EC2 전용 호스트의 호스트 유지 관리

호스트 유지 관리를 통해 드물지만 전용 호스트가 저하되는 경우에는 해당 호스트에서 실행 중인 인스턴스를 정상 대체 전용 호스트로 자동으로 마이그레이션합니다. 그러면 워크로드의 다운타임을 최소화하고 전용 호스트 관리를 간소화하는 데 도움이 됩니다. 계획된 일상적인 Amazon EC2 유지 관리를 위해 호스트 유지 관리도 수행됩니다.

Amazon EC2는 두 가지 유형의 호스트 유지 관리를 지원합니다.

- 라이브 마이그레이션 호스트 유지 관리 - 인스턴스를 중지했다가 다시 시작하지 않고도 인스턴스가 24시간 이내에 자동으로 대체 호스트로 마이그레이션됩니다.
- 재부팅 기반 호스트 유지 관리 - 인스턴스는 인스턴스 재부팅 예약 이벤트에 예약되며 이벤트가 진행되는 동안 자동으로 중지되었다가 대체 호스트에서 다시 시작됩니다.

내용

- 호스트 유지 관리와 호스트 복구 비교
- 고려 사항
- 관련 서비스
- 요금
- Amazon EC2 전용 호스트의 호스트 유지 관리 작동 방식
- Amazon EC2 전용 호스트의 호스트 유지 관리 설정 구성

호스트 유지 관리와 호스트 복구 비교

다음 표에서는 호스트 복구와 호스트 유지 관리 간의 주요 차이점을 보여줍니다.

	호스트 복구	호스트 유지 관리
인스턴스 도달 가능성	연결 불가	연결 가능
전용 호스트 상태	`under-assessment`	`permanent-failure`
호스트 리소스 그룹	지원	지원되지 않음

호스트 복구에 대한 자세한 내용은 호스트 복구를 참조하세요.

고려 사항

- 호스트 유지 관리는 중국 리전 및 AWS GovCloud (US) Regions을 제외한 모든 AWS 리전에서 사용할 수 있습니다.
- AWS Outposts, AWS 로컬 영역 및 AWS Wavelength 영역에서는 호스트 유지 관리가 지원되지 않습니다.
- 호스트 리소스 그룹 내에 이미 있는 호스트에 대해 호스트 유지 관리를 켜거나 끌 수 없습니다. 호스트 리소스 그룹에 추가된 호스트는 해당 호스트 유지 관리 설정을 유지합니다. 자세한 내용은 호스트 리소스 그룹을 참조하세요.
- 호스트 유지 관리는 인스턴스 저장소 지원 루트 볼륨이 있으므로 C1, C3, D2, I2, M1, M2, M3, R3, X1 인스턴스 유형에서 지원되지 않습니다.

관련 서비스

전용 호스트가 AWS License Manager와 통합됨 - Amazon EC2 전용 호스트 전체에서 라이선스를 추적합니다(AWS License Manager를 사용할 수 있는 리전에서만 지원됨). 자세한 내용은 AWS License Manager 사용 설명서를 참조하세요.

AWS 계정에 새 전용 호스트에 대한 라이선스가 충분히 있어야 합니다. 예약된 이벤트가 완료된 후 호스트가 해제되면 성능이 저하된 호스트와 연결된 라이선스가 해제됩니다.

요금

호스트 유리 관리 사용에 대한 추가 요금은 없지만 일반적으로 전용 호스트 요금이 적용됩니다. 자세한 내용은 Amazon EC2 전용 호스트 요금을 참조하세요.

호스트 유지 관리가 시작된 직후 성능이 저하된 전용 호스트에 대해서는 더 이상 요금이 청구되지 않습니다. 대체 전용 호스트에 대한 요금 청구는 `available` 상태가 된 후에 시작됩니다.

성능 저하된 전용 호스트가 온디맨드 요금을 사용하여 청구된 경우 대체 전용 호스트도 온디맨드 요금을 사용하여 청구됩니다. 성능이 저하된 전용 호스트에 활성 전용 호스트 예약이 있는 경우, 새 전용 호스트로 전송됩니다.

Amazon EC2 전용 호스트의 호스트 유지 관리 작동 방식

호스트 유지 관리가 활성화된 전용 호스트에서 성능 저하가 감지되면 자동으로 계정에 대체 전용 호스트가 할당됩니다. 전용 호스트 대체는 새로운 호스트 ID를 받지만 다음과 같이 원본 전용 호스트와 동일한 속성을 유지합니다.

- 자동 배치 설정
- 가용 영역
- 전용 호스트 예약 연결
- 호스트 선호도
- 호스트 유지 관리 설정
- 호스트 복구 설정
- 인스턴스 유형
- Tags

대체 호스트가 할당되면 인스턴스에 따라 라이브 마이그레이션 호스트 유지 관리 또는 재부팅 기반 호스트 유지 관리를 사용하여 인스턴스를 마이그레이션합니다.

성능이 저하된 호스트에 더 이상 실행 중인 인스턴스가 없으면 계정에서 영구적으로 릴리스됩니다.

라이브 마이그레이션 호스트 유지 관리

라이브 마이그레이션 호스트 유지 관리가 필요한 인스턴스는 중지 및 다시 시작 없이 24시간 이내에 자동으로 대체 호스트로 마이그레이션됩니다. 마이그레이션된 인스턴스는 다음과 같은 기존 속성을 유지합니다.

- 인스턴스 ID
- 인스턴스 메타데이터
- Amazon EBS 볼륨 연결

- 탄력적 IP 주소 및 프라이빗 IP 주소
- 메모리, CPU, 네트워킹 상태

인스턴스 크기가 더 큰 일부 인스턴스는 마이그레이션 도중 약간의 성능 저하가 발생할 수 있습니다.

인스턴스가 자동으로 대체 호스트로 마이그레이션되면 이메일 및 AWS Health 대시보드 알림이 발송됩니다. 성능 저하 및 교체 호스트의 ID, 실시간 마이그레이션 호스트 유지 관리를 사용하여 자동으로 마이그레이션된 인스턴스에 대한 정보 및 나머지 인스턴스에 대한 정보가 알림에 포함됩니다.

재부팅 기반 호스트 유지 관리

재부팅 기반 호스트 유지 관리가 필요한 인스턴스는 알림을 받은 날짜로부터 14일 동안 인스턴스 재부팅 예약 이벤트가 예약됩니다. 예약된 이벤트 전에 성능이 저하된 전용 호스트에서 인스턴스에 계속 액세스할 수 있습니다.

재부팅 이벤트는 원래 이벤트 날짜와 시간으로부터 7일 이내의 날짜로 예약을 변경할 수 있습니다. 자세한 내용은 EC2 인스턴스에 대해 예약된 이벤트 다시 예약 섹션을 참조하세요.

Amazon EC2는 이러한 인스턴스를 위해 대체 호스트의 용량을 자동으로 예약합니다. 이 예약 용량에서는 인스턴스를 실행할 수 없습니다.

Amazon EC2 콘솔에는 예약 용량이 사용된 용량으로 표시됩니다. 인스턴스가 성능이 저하된 호스트와 대체 호스트 모두에서 실행되고 있는 것처럼 보일 수 있습니다. 그러나 인스턴스는 중지되거나 대체 호스트의 예약 용량으로 마이그레이션될 때까지 성능이 저하된 호스트에서만 계속 실행됩니다.

예약된 이벤트 날짜 및 시간에 인스턴스가 자동으로 중지되고 대체 호스트의 예약 용량으로 다시 시작됩니다. 마이그레이션된 인스턴스는 다음과 같은 기존 속성을 유지합니다.

- 인스턴스 ID
- 인스턴스 메타데이터
- Amazon EBS 볼륨 연결
- 탄력적 IP 주소 및 프라이빗 IP 주소

그러나 마이그레이션 중에 인스턴스가 중지되었다가 다시 시작되므로 메모리, CPU, 네트워킹 상태는 유지되지 않습니다.

또한 예약된 이벤트 전에 언제든지 이러한 인스턴스를 수동으로 중지했다가 다시 시작하여 대체 호스트 또는 다른 호스트로 마이그레이션할 수도 있습니다. 다른 호스트에서 다시 시작하려면 인스턴스의

호스트 선호도를 수정해야 할 수 있습니다. 예약된 이벤트 전에 인스턴스를 중지하면 대체 호스트의 예약 용량이 릴리스되어 사용할 수 있게 됩니다.

호스트 유지 관리 상태

호스트의 성능이 저하되면 `permanent-failure` 상태가 됩니다. `permanent-failure` 상태인 전용 호스트에서는 인스턴스를 시작할 수 없습니다.

대체 호스트가 할당되면 라이브 마이그레이션 호스트 유지 관리를 지원하는 인스턴스가 성능 저하된 호스트에서 자동으로 마이그레이션되고, 나머지 인스턴스에 대해 예약된 이벤트가 예약될 때까지 `pending` 상태로 유지됩니다. 이러한 작업이 완료되면 대체 호스트는 `available` 상태로 전환됩니다.

대체 호스트가 `available` 상태로 전환되면 계정에서 다른 호스트를 사용하는 것과 동일한 방식으로 사용할 수 있습니다. 그러나 대체 호스트의 일부 인스턴스 용량은 재부팅 기반 호스트 마이그레이션이 필요한 인스턴스를 위해 예약되어 있습니다. 이 예약 용량으로는 새 인스턴스를 시작할 수 없습니다.

성능이 저하된 호스트에 더 이상 실행 중인 인스턴스가 없으면 `released, permanent-failure` 상태가 되며 계정에서 영구적으로 릴리스됩니다. 참고로, 호스트와 해당 리소스가 콘솔에 잠깐 표시됩니다.

자동 마이그레이션

일부 인스턴스는 대체 호스트로 자동으로 마이그레이션할 수 없습니다.

EBS 지원 루트 볼륨이 있는 인스턴스

이러한 인스턴스의 경우에는 알림 날짜로부터 28일 동안 인스턴스 종료 이벤트가 예약됩니다. 예약된 이벤트의 날짜 및 시간에 인스턴스가 중지됩니다. 대체 호스트 또는 다른 호스트에서 인스턴스를 다시 시작할 때 수동으로 중지하는 것이 좋습니다. 다른 호스트에서 다시 시작하려면 인스턴스의 호스트 선호도를 수정해야 할 수 있습니다.

인스턴스 저장소 지원 루트 볼륨이 있는 인스턴스

이러한 인스턴스의 경우에는 알림 날짜로부터 28일 동안 인스턴스 종료 이벤트가 예약됩니다. 예약된 이벤트 날짜 및 시간에 인스턴스가 영구적으로 종료됩니다. 예약된 이벤트 전에 대체 호스트에서 대체 인스턴스를 수동으로 시작한 다음 필요한 데이터를 대체 인스턴스로 마이그레이션하는 것이 좋습니다.

C1, C3, D2, I2, M1, M2, M3, R3, X1 인스턴스에는 인스턴스 저장소 지원 루트 볼륨이 있습니다.

예약된 이벤트 전에 성능이 저하된 전용 호스트에서 인스턴스에 계속 액세스할 수 있습니다.

Amazon EC2 전용 호스트의 호스트 유지 관리 설정 구성

호스트 유지 관리를 활성화하면 예약된 유지 관리 이벤트 중에 전용 호스트에서 실행되는 인스턴스가 새 전용 호스트로 자동 복구됩니다.

호스트 유지 관리를 비활성화하면 성능이 저하된 호스트를 제거하고 28일 이내에 수동으로 인스턴스를 다른 호스트로 마이그레이션하라는 이메일 알림을 받게 됩니다. 전용 호스트 예약이 있는 경우 대체 호스트가 할당됩니다. 28일이 지나면 성능이 저하된 호스트에서 실행 중인 인스턴스가 종료되고 호스트가 자동으로 해제됩니다.

Console

전용 호스트에 대한 호스트 유지 관리를 활성화하려면 다음을 수행하세요.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 전용 호스트(Dedicated Hosts)를 선택합니다.
3. 전용 호스트 > 작업 > 호스트 수정을 선택합니다.
4. 호스트 유지 관리 필드에서 켜기를 선택합니다.

전용 호스트에 대한 호스트 유지 관리를 비활성화하려면 다음을 수행하세요.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 전용 호스트(Dedicated Hosts)를 선택합니다.
3. 전용 호스트 > 작업 > 호스트 수정을 선택합니다.
4. 호스트 유지 관리 필드에서 끄기를 선택합니다.

AWS CLI

전용 호스트에 대한 호스트 유지 관리를 활성화하려면 다음을 수행하세요.

modify-hosts 명령을 사용합니다.

```
aws ec2 modify-hosts \
    --host-maintenance on \
    --host-ids h-0d123456bbf78910d
```

전용 호스트에 대한 호스트 유지 관리를 비활성화하려면 다음을 수행하세요.

modify-hosts 명령을 사용합니다.

```
aws ec2 modify-hosts \
    --host-maintenance off \
    --host-ids h-0d123456bbf78910d
```

PowerShell

전용 호스트에 대한 호스트 유지 관리를 활성화하려면 다음을 수행하세요.

Edit-EC2Host cmdlet을 사용합니다.

```
Edit-EC2Host `
    -HostMaintenance on `
    -HostId h-0d123456bbf78910d
```

전용 호스트에 대한 호스트 유지 관리를 비활성화하려면 다음을 수행하세요.

Edit-EC2Host cmdlet을 사용합니다.

```
Edit-EC2Host `
    -HostMaintenance off `
    -HostId h-0d123456bbf78910d
```

Amazon EC2 전용 호스트 상태 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 명령을 사용합니다.

```
aws ec2 describe-hosts --host-id h-012a3456b7890cdef
```

PowerShell

전용 호스트의 상태를 보려면

Get-EC2Host cmdlet을 사용합니다.

```
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 Console 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 수정되고 있거나, 호스트 복구가 진행 중입니다.
permanent-failure	복구할 수 없는 오류가 감지되었습니다. 인스턴스 및 이메일을 통해 제거 알림이 제공됩니다. 인스턴스는 계속 실행할 수 있습니다. 이 상태의 전용 호스트의 모든 인스턴스를 중지하거나 종료하면 AWS에서 호스트를 사용 중지합니다. AWS는 이 상태에서 인스턴스를 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.

AWS Config를 사용하여 Amazon EC2 전용 호스트 구성 변경 내용 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 - 켜짐(On)이면 AWS Config에서 하나 이상의 AWS 리소스 유형을 기록 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 활성화하세요. 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 섹션을 참조하세요.

> **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 Machine Image(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 아

이콘을 선택합니다. 보다 자세한 내용은 AWS Config 개발자 안내서의 AWS Config 콘솔에서 구성 세부 사항 보기를 참조하세요.

- 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.
- 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 콘솔을 사용하여 AWS Config 설정을 참조하세요.

자세한 내용은 AWS Config 콘솔에서 구성 세부 정보 보기를 참조하세요.

명령줄 또는 API를 사용하여 AWS Config를 활성화하려면

- AWS CLI: AWS CLI 개발자 안내서의 구성 세부 정보 보기(AWS Config).
- Amazon EC2 API: GetResourceConfigHistory.

Amazon EC2 전용 인스턴스

기본적으로 EC2 인스턴스는 공유 테넌시 하드웨어에서 실행됩니다. 즉, 여러 AWS 계정이 동일한 물리적 하드웨어를 공유할 수 있습니다.

전용 인스턴스는 단일 AWS 계정 전용 하드웨어에서 실행되는 EC2 인스턴스입니다. 즉, 전용 인스턴스는 해당 계정이 단일 지급인 계정에 연결되어 있더라도 다른 AWS 계정에 속한 인스턴스로부터 호스트 하드웨어 수준에서 물리적으로 격리됩니다. 하지만 전용 인스턴스는 전용 인스턴스가 아닌 동일한 AWS 계정의 다른 인스턴스와 하드웨어를 공유할 수 있습니다.

전용 인스턴스는 인스턴스 배치에 대한 가시성이나 제어 기능을 제공하지 않으며 호스트 선호도를 지원하지 않습니다. 전용 인스턴스를 중지했다가 시작하면 동일한 호스트에서 실행되지 않을 수 있습니다. 마찬가지로 인스턴스를 시작하거나 실행할 특정 호스트를 대상으로 지정할 수 없습니다. 또한 전용 인스턴스는 기존 보유 라이선스 사용(BYOL)에 대한 제한적인 지원을 제공합니다.

인스턴스 배치에 대한 가시성 및 제어 기능과 보다 포괄적인 BYOL 지원이 필요한 경우 대신 전용 호스트 사용을 고려해보세요. 전용 인스턴스와 전용 호스트 모두 전용 물리적 서버로 Amazon EC2 인스턴스를 시작하는 데 사용할 수 있습니다. 전용 호스트의 인스턴스와 전용 인스턴스는 성능이나 보안상의 차이나 물리적 차이는 없습니다. 그러나 이들 사이에는 몇 가지 주요 차이점이 있습니다. 다음 테이블에서는 전용 인스턴스와 전용 호스트의 주요 차이점 중 일부를 요약하여 설명합니다.

	전용 호스트	Dedicated Instance
전용 물리적 서버	사용자 전용 인스턴스 용량을 갖춘 물리적 서버입니다.	단일 고객 계정 전용 물리적 서버
인스턴스 용량 공유	다른 계정과 인스턴스 용량을 공유할 수 있음.	지원되지 않음
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

	전용 호스트	Dedicated Instance
대상 지정 인스턴스 배치	물리 서버 내 인스턴스 배치 방법에 대한 추가 가시성 및 제어 제공	지원되지 않음
자동 인스턴스 복구	지원 자세한 내용은 Amazon EC2 전용 호스트 복구 섹션을 참조하세요.	지원
Bring Your Own License(BYOL)	지원	부분적 지원 *
용량 예약	지원되지 않음	지원

* 소프트웨어 보증을 통한 라이선스 이동성을 갖춘 Microsoft SQL Server와 Windows 가상 데스크톱 액세스(VDA) 라이선스를 전용 인스턴스와 함께 사용할 수 있습니다.

자세한 내용은 Amazon EC2 전용 호스트 섹션을 참조하세요.

내용

- 전용 인스턴스 기본 사항
- 지원 기능
- 전용 인스턴스 제한 사항
- 전용 인스턴스 가격
- 기본 테넌시를 사용하는 VPC로 전용 인스턴스 시작
- EC2 인스턴스의 테넌시 변경
- VPC의 인스턴스 테넌시 변경

전용 인스턴스 기본 사항

VPC는 `default` 또는 `dedicated` 테넌시 중 하나를 가질 수 있습니다. 기본적으로 VPC에는 `default` 테넌시가 있고 `default` 테넌시 VPC에 시작된 인스턴스에는 `default` 테넌시가 있습니다. 전용 인스턴스를 시작하려면 다음 작업을 수행합니다.

- 테넌시가 `dedicated`인 VPC를 생성하여 VPC의 모든 인스턴스가 전용 인스턴스로 실행되도록 합니다. 자세한 내용은 기본 테넌시를 사용하는 VPC로 전용 인스턴스 시작 섹션을 참조하세요.
- 테넌시가 `default`인 VPC를 생성하고 전용 인스턴스로 실행하려는 인스턴스에 대해 `dedicated`의 테넌시를 수동으로 지정합니다. 자세한 내용은 기본 테넌시를 사용하는 VPC로 전용 인스턴스 시작 섹션을 참조하세요.

지원 기능

전용 인스턴스는 다음의 기능과 AWS 서비스 통합을 지원합니다.

Features

- 예약 인스턴스
- 자동 크기 조정
- 자동 복구
- 전용 스팟 인스턴스
- 성능 순간 확장 가능 인스턴스

예약 인스턴스

전용 인스턴스의 용량을 예약하려면 전용 예약 인스턴스나 용량 예약을 구매합니다. 자세한 내용은 Amazon EC2용 예약 인스턴스 개요 및 EC2 온디맨드 용량 예약으로 컴퓨팅 용량 예약(을)를 참조하세요.

전용 예약 인스턴스를 구입하면 대폭 할인된 사용 요금으로 전용 인스턴스를 시작할 수 있는 용량이 제공됩니다. 사용 요금 인하는 전용 테넌시 인스턴스를 시작할 경우에만 적용됩니다. 기본 테넌시가 있는 예약 인스턴스를 구입하면 `default` 테넌시가 있는 실행 인스턴스에만 적용되고 `dedicated` 테넌시가 있는 실행 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.

자동 크기 조정

Amazon EC2 Auto Scaling을 사용하여 전용 인스턴스를 시작할 수 있습니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 고급 설정을 사용한 시작 템플릿 생성하기를 참조하세요.

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 자동 인스턴스 복구 섹션을 참조하세요.

전용 스팟 인스턴스

스팟 인스턴스 요청을 생성할 때 `dedicated`의 테넌시를 지정하여 전용 스팟 인스턴스를 실행할 수 있습니다. 자세한 내용은 단일 테넌트 하드웨어에서 시작 섹션을 참조하세요.

성능 순간 확장 가능 인스턴스

the section called “성능 순간 확장 가능 인스턴스”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 인스턴스는 임의의 CPU 동작을 나타내지만 평균 CPU 사용량이 기본 사용량 이하인 다양한 워크로드를 사용하는 고객을 대상으로 합니다. 자세한 내용은 the section called “주요 개념” 섹션을 참조하세요.

Amazon EC2 에는 성능 변동성을 식별하고 수정할 수 있는 시스템이 있습니다. 그러나 상관 관계가 있는 CPU 사용 패턴을 보이는 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 제한 사항

전용 인스턴스를 사용할 때는 다음 사항에 유의하세요.

- 일부 AWS 서비스나 그 기능은 인스턴스 테넌시가 `dedicated`로 설정된 VPC에서 지원되지 않습니다. 이에 관한 제한 사항이 있는지 확인하려면 각 서비스의 설명서를 참조하세요.
- 일부 인스턴스 유형은 인스턴스 테넌시가 `dedicated`로 설정된 VPC에서 시작할 수 없습니다. 지원되는 인스턴스 유형에 대한 자세한 내용은 Amazon EC2 전용 인스턴스를 참조하세요.
- Amazon EBS를 기반으로 하는 전용 인스턴스를 시작하는 경우 EBS 볼륨은 단일 테넌트 하드웨어에서 실행되지 않습니다.

전용 인스턴스 가격

전용 인스턴스 요금은 온디맨드 인스턴스 요금과 다릅니다. 자세한 내용은 Amazon EC2 전용 인스턴스를 참조하세요.

기본 테넌시를 사용하는 VPC로 전용 인스턴스 시작

VPC를 생성할 경우 VPC의 인스턴스 테넌시를 지정하는 옵션이 제공됩니다. `dedicated`의 인스턴스 테넌시가 있는 VPC에서 인스턴스를 시작하면 인스턴스가 사용자 전용 하드웨어에서 전용 인스턴스로 실행됩니다.

`host` 테넌시를 사용하여 인스턴스를 시작하는 자세한 내용은 Amazon EC2 전용 호스트에서 Amazon EC2 인스턴스 시작 섹션을 참조하세요.

VPC 테넌시 옵션에 대한 자세한 내용은 Amazon VPC 사용 설명서의 VPC 생성을 참조하세요.

요구 사항

- 지원되는 인스턴스 유형을 선택합니다. 자세한 내용은 Amazon EC2 전용 인스턴스를 참조하세요.

Console

기본 테넌시 VPC에서 전용 인스턴스를 시작하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances(인스턴스), Launch instance(인스턴스 시작)를 선택합니다.
3. Application and OS Images(애플리케이션 및 OS 이미지) 섹션의 목록에서 AMI를 선택합니다.
4. Instance type(인스턴스 유형) 섹션에서 시작할 인스턴스 유형을 선택합니다.
5. Key pair(키 페어) 섹션에서 인스턴스와 연결할 키 페어를 선택합니다.
6. Advanced details(고급 세부 정보) 섹션의 Tenancy(테넌시)에서 Dedicated(전용)를 선택합니다.
7. 필요에 따라 나머지 인스턴스 옵션을 구성합니다. 자세한 내용은 Amazon EC2 인스턴스 구성 파라미터에 대한 참조 섹션을 참조하세요.
8. 인스턴스 시작을 선택합니다.

AWS CLI

시작 시 인스턴스의 테넌시 옵션을 설정하려면

run-instances 명령을 사용하고 `--placement` 옵션과 함께 `Tenancy`를 포함합니다.

```
--placement Tenancy=dedicated
```

PowerShell

시작 시 인스턴스의 테넌시 옵션을 설정하려면

`-Placement_Tenancy` 파라미터와 함께 New-EC2Instance cmdlet을 사용합니다.

```
-Placement_Tenancy dedicated
```

EC2 인스턴스의 테넌시 변경

시작 후 중지된 인스턴스의 테넌시를 변경할 수 있습니다. 변경한 내용은 다음에 인스턴스가 시작될 때 적용됩니다.

Virtual Private Cloud(VPC)의 테넌시를 변경할 수도 있습니다. 자세한 내용은 the section called “VPC의 테넌시 변경” 섹션을 참조하세요.

제한 사항

- AWS Management Console을 사용하여 인스턴스의 테넌시를 변경할 수 없습니다.
- 인스턴스는 `stopped` 상태여야 합니다.
- 인스턴스의 운영 체제 세부 정보 및 SQL Server 설치 여부는 지원되는 변환에 영향을 줍니다. 인스턴스에 사용할 수 있는 테넌시 변환 경로에 대한 자세한 내용은 License Manager 사용 설명서의 Tenancy conversion을 참조하세요.
- T3 인스턴스의 경우 전용 호스트에서 인스턴스를 시작해야 `host`의 테넌시를 사용할 수 있습니다. 테넌시를 `host`에서 `dedicated` 또는 `default`로 변경할 수 없습니다. 이러한 지원되지 않는 테넌시 변경 사항 중 하나를 만들려고 하면 `InvalidRequest` 오류 코드가 표시됩니다.

AWS CLI

인스턴스의 테넌시 값을 수정하려면

modify-instance-placement 명령을 사용합니다.

```
aws ec2 modify-instance-placement \
    --instance-id i-1234567890abcdef0 \
```

```
    --tenancy dedicated
```

PowerShell

인스턴스의 테넌시 값을 수정하려면

Edit-EC2InstancePlacement cmdlet을 사용합니다.

```
Edit-EC2InstancePlacement `
    -InstanceId i-1234567890abcdef0 `
    -Tenancy Dedicated
```

VPC의 인스턴스 테넌시 변경

Virtual Private Cloud(VPC)의 인스턴스 테넌시는 생성한 후에 dedicated에서 default로 변경할 수 있습니다. VPC의 인스턴스 테넌시를 수정해도 VPC에 있는 기존 인스턴스의 테넌시에는 영향을 미치지 않습니다. 다음에 VPC에서 인스턴스를 시작할 때 인스턴스 시작 시 다르게 지정하지 않는 한 default 테넌시가 유지됩니다.

또는 특정 인스턴스의 테넌시를 변경할 수 있습니다. 자세한 내용은 the section called “인스턴스의 테넌시 변경” 섹션을 참조하세요.

제한 사항

- VPC의 인스턴스 테넌시는 생성한 후에 default에서 dedicated로 변경할 수 없습니다.
- AWS Management Console을 사용하여 VPC의 인스턴스 테넌시를 수정할 수 없습니다.

AWS CLI

VPC의 인스턴스 테넌시 속성을 수정하려면

modify-vpc-tenancy 명령을 사용합니다. 지원되는 유일한 테넌시 값은 default입니다.

```
aws ec2 modify-vpc-tenancy \
    --vpc-id vpc-1234567890abcdef0 \
    --instance-tenancy default
```

PowerShell

VPC의 인스턴스 테넌시 속성을 수정하려면

Edit-EC2VpcTenancy cmdlet을 사용합니다. 지원되는 유일한 테넌시 값은 `Default`입니다.

```
Edit-EC2VpcTenancy `
    -VpcId vpc-1234567890abcdef0 `
    -InstanceTenancy Default
```

온디맨드 용량 예약 및 ML용 용량 블록

용량 예약을 사용하면 특정 가용 영역의 Amazon EC2 인스턴스에 대한 컴퓨팅 용량을 예약할 수 있습니다. 다양한 다른 사용 사례에 적용되는 두 가지 용량 예약 유형이 있습니다.

용량 예약 유형

- 온디맨드 용량 예약
- ML용 용량 블록

다음은 온디맨드 용량 예약의 몇 가지 일반적인 사용 사례입니다.

- 이벤트 규모 조정 – 필요할 때 규모를 조정할 수 있도록비즈니스에 중요한 이벤트에 앞서 온디맨드 용량 예약을 생성합니다.
- 규제 요구 사항 및 재해 복구 – 온디맨드 용량 예약을 사용하여 고가용성에 대한 규제 요구 사항을 충족하고 재해 복구를 위해 다른 가용 영역 또는 리전의 용량을 예약합니다.

다음은 ML용 용량 블록의 몇 가지 일반적인 사용 사례입니다.

- 기계 학습(ML) 모델 훈련 및 미세 조정 – 예약한 GPU 인스턴스에 중단 없이 액세스하여 ML 모델 훈련 및 미세 조정을 완료합니다.
- ML 실험 및 프로토타입 – GPU 인스턴스가 단기간 필요한 실험을 실행하고 프로토타입을 구축합니다.

온디맨드 용량 예약 사용 시기

용량 요구 사항이 엄격하고, 현재 또는 장래의 비즈니스 크리티컬 워크로드에 용량 보증이 필요한 경우 온디맨드 용량 예약을 사용합니다. 온디맨드 용량 예약을 사용하면 필요한 기간에 언제든지 예약한 Amazon EC2 용량에 액세스할 수 있습니다.

ML용 용량 블록 사용 시점

미래 날짜에 시작하도록 정의된 기간에 중단 없이 GPU 인스턴스에 액세스해야 할 때 ML용 용량 블록을 사용합니다. 용량 블록은 ML 모델 훈련 및 미세 조정, 간단한 실험 실행, 향후 일시적으로 급증하는 추론 수요 처리에 적합합니다. 용량 블록을 사용하면 특정 날짜에 GPU 리소스에 액세스하여 ML 워크로드를 실행할 수 있습니다.

EC2 온디맨드 용량 예약으로 컴퓨팅 용량 예약

Amazon EC2 용량 예약을 사용하면 특정 가용 영역의 Amazon EC2 인스턴스의 컴퓨팅 용량을 원하는 기간 동안 예약할 수 있습니다. 특정 수준의 장기 또는 단기 용량이 보장되어야 하는 현재 또는 장래의 비즈니스 크리티컬 워크로드를 위한 엄격한 용량 요구 사항이 있는 경우, 필요할 때 언제든지 필요한 기간 동안 Amazon EC2 용량에 액세스할 수 있도록 용량 예약을 생성하는 것이 좋습니다.

언제든지 용량 예약을 생성할 수 있고, 시작 시기를 선택할 수 있습니다. 즉시 사용할 수 있는 용량 예약을 요청하거나, 장래 날짜를 위한 용량 예약을 요청할 수 있습니다.

- 즉시 사용할 수 있는 용량 예약을 요청하는 경우, 용량 예약을 즉시 사용할 수 있게 되며 기간 약정이 없습니다. 용량 예약은 언제든지 수정할 수 있고, 언제든지 취소하여 예약 용량을 해제하고 요금이 발생하지 않도록 할 수 있습니다.
- 장래 날짜 용량 예약을 요청하는 경우 용량 예약을 사용할 수 있는 장래 날짜를 지정합니다. 또한 지정된 날짜 이후에 요청된 용량을 계정에서 유지하기 위해서는 약정 기간을 지정해야 합니다. 요청된 날짜와 시간에 용량 예약을 사용할 수 있게 되고 약정 기간이 시작됩니다. 약정 기간 동안에는 인스턴스 수나 약정 기간을 초기 약정 미만으로 줄이거나 용량 예약을 취소할 수 없습니다. 약정 기간이 경과한 후에는 어떤 식으로든 용량 예약을 수정하거나 더 이상 필요하지 않은 경우 취소할 수 있습니다.

용량 예약는 해당 속성과 일치하는 인스턴스에서만 사용할 수 있습니다. 기본적으로 용량 예약은 일치하는 속성(인스턴스 유형, 플랫폼, 가용 영역, 테넌시)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. 인스턴스가 용량 예약이나 용량 예약 리소스 그룹에서만 실행되도록 지정할 수도 있습니다.

> **Important**
>
> 장래 날짜 용량 예약은 증분 인스턴스를 시작하고 충당하는 데 도움이 되며, 실행 중인 기존 인스턴스는 충당하지 않습니다. 실행 중인 기존 인스턴스를 충당해야 하는 경우 즉시 시작하는 용량 예약을 대신 사용하세요.

인스턴스 유형, 플랫폼, 가용 영역, 테넌시 등의 속성이 일치하는 지원되는 모든 Amazon EC2 인스턴스는 용량 예약에서 실행할 수 있습니다. Amazon EC2 인스턴스는 사용자가 직접 시작하거나(비관리형 인스턴스), AWS 서비스에서 사용자를 대신하여 시작할 수 있습니다(관리형 인스턴스). 속성이 일치하는 실행 중인 인스턴스와 자동으로 일치하는 열린 용량 예약의 경우 특히 그렇습니다. 예를 들어 다음 서비스에서 사용자를 대신해 시작한 관리형 인스턴스는 사용자가 생성하고 관리하는 용량 예약에서 실행할 수 있습니다.

- Amazon EC2 Auto Scaling
- Amazon EMR
- AWS ParallelCluster
- Amazon EKS
- Amazon ECS
- AWS Batch
- AWS Elastic Beanstalk
- Amazon SageMaker AI

내용

- Amazon EC2 용량 예약 개념
- 용량 예약, 예약 인스턴스 및 Savings Plans의 차이점
- 지원되는 플랫폼
- 할당량
- 제한 사항
- 용량 예약 요금 및 결제
- 용량 예약 생성
- 용량 예약 상태 보기
- 인스턴스를 기존 용량 예약으로 시작
- 활성 용량 예약 수정
- 인스턴스의 용량 예약 설정 수정
- 용량 예약 간 용량 이동
- 기존 용량 예약에서 용량 분할
- 용량 예약 취소

- 용량 예약 그룹
- 클러스터 배치 그룹에 용량 예약 생성
- Local Zones의 용량 예약
- Wavelength 영역의 용량 예약
- AWS Outposts의 용량 예약
- 공유 용량 예약
- 용량 예약 플릿
- CloudWatch 지표를 사용하여 용량 예약 모니터링
- 용량 예약 사용률 저하 모니터링
- 장래 날짜 용량 예약의 상태 변경 모니터링

Amazon EC2 용량 예약 개념

용량 예약에는 다음과 같은 주요 개념이 적용됩니다.

주제

- 현재 날짜 및 시간
- 현재 날짜 및 시간
- 약정 기간
- 장래 날짜 용량 예약 평가
- 용량 예약 속성
- 인스턴스 일치 기준

현재 날짜 및 시간

시작 날짜 및 시간은 용량 예약을 사용할 수 있게 되는 시점을 정의합니다. 용량 예약은 즉시 시작하거나 장래 날짜에 시작할 수 있습니다.

- 용량 예약을 즉시 시작하도록 선택하면 용량 예약을 생성한 직후 예약 용량을 사용할 수 있게 되고 용량 예약이 활성 상태가 되는 즉시 요금 청구가 시작됩니다. 기간 약정을 체결할 필요는 없습니다. 필요에 따라 언제든지 용량 예약을 수정할 수 있으며 언제든지 취소하여 용량을 해제하고 요금이 발생하지 않도록 할 수 있습니다.
- 장래 날짜에 용량 예약을 시작하도록 선택하는 경우 예약 용량이 필요한 장래 날짜 및 시간과 요청된 용량 예약이 프로비저닝된 후 계정에서 유지하기로 약정하는 최소 기간인 약정 기간을 지정합니

다. 지정된 장래 날짜에 용량 예약을 사용할 수 있게 되고 용량 예약이 활성 상태가 되면 해당 시점에 요금 청구가 시작됩니다. 약정 기간은 계정에 용량 예약이 프로비저닝되는 즉시 시작됩니다. 이 기간 동안에는 인스턴스 수를 약정한 인스턴스 수 미만으로 줄이거나, 약정 기간 이전의 종료 날짜를 선택하거나, 용량 예약을 취소할 수 없습니다. 하지만 약정 기간이 경과한 후에는 자유롭게 어떤 식으로든 용량 예약을 수정하거나 취소하여 예약 용량을 해제하고 요금이 발생하지 않도록 할 수 있습니다.

현재 날짜 및 시간

종료 날짜 및 시간은 용량 예약이 종료되고 예약 용량이 계정에서 해제되는 시점을 정의합니다. 특정 날짜 및 시간에 자동으로 종료되도록 용량 예약을 구성하거나, 수동으로 취소할 때까지 무기한 활성 상태로 유지되도록 구성할 수 있습니다.

용량 예약이 자동으로 종료되도록 구성하면 용량 예약은 지정된 시간으로부터 1시간 이내에 만료됩니다. 예를 들어 `5/31/2019, 13:30:55`를 지정하면 용량 예약이 `5/31/2019 13:30:55~14:30:55`에 만료됩니다.

예약이 종료되면 예약 용량이 계정에서 해제되고 더 이상 인스턴스를 용량 예약 대상으로 지정할 수 없습니다. 예약 용량으로 실행 중인 인스턴스가 중단되지 않은 상태로 계속 실행됩니다. 용량 예약 대상으로 지정된 인스턴스가 중지되는 경우 용량 예약 대상 지정 기본 설정을 제거하거나 다른 용량 예약 대상으로 지정되도록 구성할 때까지 해당 인스턴스를 다시 시작할 수 없습니다. 자세한 내용은 인스턴스의 용량 예약 설정 수정 섹션을 참조하세요.

약정 기간

약정 기간은 장래 날짜 용량 예약에만 적용됩니다.

약정 기간은 계정에서 장래 날짜 용량 예약을 프로비저닝 후 활성 상태로 유지하기로 약정하는 최소 기간입니다. 장래 날짜 용량 예약은 약정 기간보다 더 오래 유지할 수는 있지만 더 짧게 유지할 수는 없습니다. 약정 기간 동안에는 다음이 적용됩니다.

- 약정 기간 중에는 용량 예약을 취소할 수 없습니다.
- 인스턴스 수를 약정한 인스턴스 수 미만으로 줄일 수는 없지만 늘릴 수는 있습니다.
- 약정 기간 내의 날짜나 시간에 자동으로 종료되도록 용량 예약을 구성할 수 없습니다. 약정 기간 중에 종료 날짜와 시간을 연장할 수 있습니다.

Amazon EC2는 사용자가 지정하는 약정 기간을 사용하여 요청을 지원할 수 있는지 여부를 평가합니다. 최소 약정 기간은 14일입니다. 요청을 평가하는 동안 Amazon EC2는 이보다 짧은 약정 기간을 지원할 수 있다고 결정할 수 있습니다. 이 경우 Amazon EC2는 더 짧은 약정 기간으로 장래 날짜 용량 예

약 일정을 예약합니다. 즉, 사용자는 처음에 요청한 것보다 짧은 기간 동안 계정에서 용량 예약을 유지하기로 약정하는 것입니다.

장래 날짜 용량 예약 평가

장래 날짜 용량 예약을 요청하면 Amazon EC2는 요청을 평가하여 용량 가용성과 지정한 약정 기간에 따라 지원할 수 있는지 여부를 결정합니다. 평가는 일반적으로 5일 이내에 완료됩니다. Amazon EC2는 요청을 평가할 때 다음과 같은 여러 요소를 고려합니다.

- 예상되는 용량 공급
- 약정 기간
- 시작 날짜와 비교하여 얼마나 일찍 용량 예약을 요청하는지 여부
- 요청의 크기

5~120일 전에 장래 날짜 용량 예약을 요청할 수 있습니다. 요청 지원 역량을 개선하기 위해 미리 최소 56일(8주) 전에 요청하는 것이 좋습니다. 최소 약정 기간은 14일이고 최소 인스턴스 수는 100개의 vCPU입니다.

요청이 평가되는 동안 용량 예약은 `assessing` 상태로 유지됩니다.

지원할 수 있는 요청인 경우 용량 예약이 `scheduled` 상태로 전환되고 요청된 날짜 및 시간에 제공되도록 예약됩니다. 용량 예약이 `scheduled` 상태에 있는 동안 총 인스턴스 수는 0으로 유지됩니다. 예약된 용량 예약은 요청된 날짜에 `active` 상태가 되고 사용할 수 있게 됩니다.

요청을 지원할 수 없는 경우 용량 예약이 `unsupported` 상태로 전환됩니다. 지원되지 않는 용량 예약은 제공되지 않습니다.

장래 날짜 용량 예약은 `assessing` 상태인 동안 취소할 수 있습니다.

자세한 내용은 장래 날짜 용량 예약 생성 섹션을 참조하세요.

용량 예약 속성

용량 예약을 생성할 때 다음 속성을 지정해야 합니다.

- 가용 영역
- 인스턴스 유형
- 플랫폼(운영 체제 유형)

- 테넌시(`default` 또는 `dedicated`)

이러한 속성과 일치하는 인스턴스만 용량 예약에서 시작하거나 실행할 수 있습니다.

인스턴스 일치 기준

인스턴스 일치 기준 또는 인스턴스 자격에 따라 용량 예약에서 시작하고 실행할 수 있는 인스턴스가 결정됩니다. 용량 예약에는 다음 중 하나의 일치 기준이 있을 수 있습니다.

- Open - 용량 예약은 일치하는 속성(인스턴스 유형, 플랫폼, 가용 영역)이 있는 모든 인스턴스에 자동으로 연결됩니다. 일치하는 속성이 있는 신규 및 기존 인스턴스는 추가 구성 없이 용량 예약에서 자동으로 실행됩니다.
- Targeted - 용량 예약은 일치하는 속성(인스턴스 유형, 플랫폼, 가용 영역)이 있고 예약을 명시적으로 대상 지정하는 인스턴스만 허용합니다. 인스턴스가 예약 용량에서 시작되거나 실행되려면 구체적으로 용량 예약을 대상으로 지정해야 합니다. 이렇게 하면 예약 용량에서 실행할 수 있는 인스턴스를 명시적으로 제어할 수 있으며 의도하지 않은 예약 용량 사용을 방지할 수 있습니다.

장래 날짜 용량 예약을 요청할 때 대상 지정 일치 기준만 지정할 수 있습니다. 이렇게 하면 용량 예약이 제공하는 용량이 제공 시 실행 중인 인스턴스나 예약 용량에 증분 또는 추가됩니다. 계정에서 용량 예약이 활성화된 후 필요한 경우 인스턴스 일치 기준을 open으로 변경할 수 있습니다. 하지만 일치하는 모든 인스턴스가 용량 예약에서 자동으로 실행되므로 의도하지 않은 용량 사용량이 발생하고 요청된 전체 인스턴스 수 때문에 새 인스턴스를 시작하지 못할 수 있습니다.

용량 예약, 예약 인스턴스 및 Savings Plans의 차이점

다음 표에 용량 예약, 예약 인스턴스 및 Savings Plans 간의 주요 차이점이 설명되어 있습니다.

	Capacity Reservations	영역 예약 인스턴스	리전 예약 인스턴스	Savings Plans
용어	즉시 사용 용량 예약에는 약정이 필요하지 않습니다. 즉시 사용 용량 예약은 필요할 때마다 생성하거나 취소할 수 있습니다.	고정 1년 또는 3년 약정 필요		

	Capacity Reservations	영역 예약 인스턴스	리전 예약 인스턴스	Savings Plans
	장래 날짜 용량 예약에서는 계정에서 용량을 유지하기로 약정하는 약정 기간을 지정합니다. 약정 기간이 경과한 후에는 언제든지 용량 예약을 취소할 수 있습니다.			
용량 혜택	특정 가용 영역에서 예약된 용량입니다.		예약된 용량이 없습니다.	
결제 할인	결제 할인이 제공되지 않습니다. †	결제 할인을 제공합니다.		
인스턴스 제한	리전별 온디맨드 인스턴스 한도가 적용됩니다.	기본값은 가용 영역당 20입니다. 한도 증가를 요청할 수 있습니다.	기본값은 리전당 20입니다. 한도 증가를 요청할 수 있습니다.	제한 없음.

† 용량 예약을 Savings Plans 또는 리전 예약 인스턴스와 결합하여 할인을 받을 수 있습니다.

자세한 정보는 다음을 참조하세요.

- Amazon EC2용 예약 인스턴스 개요
- 절감형 플랜 사용 설명서

지원되는 플랫폼

인스턴스와 일치하는지 확인하려면 올바른 플랫폼으로 용량 예약을 생성해야 합니다. 용량 예약은 `platform`에 대해 다음 값을 지원합니다.

- Linux/UNIX
- SQL Server Standard가 설치된 Linux

- SQL Server Web이 설치된 Linux
- SQL Server Enterprise가 설치된 Linux
- SUSE Linux
- Red Hat Enterprise Linux
- SQL Server Standard가 설치된 RHEL
- SQL Server Enterprise가 설치된 RHEL
- SQL Server 웹이 설치된 RHEL
- HA가 설치된 RHEL
- HA 및 SQL Server Standard가 설치된 RHEL
- HA 및 SQL Server Enterprise가 설치된 RHEL
- Ubuntu Pro
- Windows
- SQL Server가 설치된 Windows
- SQL Server Web이 설치된 Windows
- SQL Server Standard가 설치된 Windows
- SQL Server Enterprise가 설치된 Windows

인스턴스가 특정 용량 예약에서 실행되도록 하려면 용량 예약의 플랫폼이 인스턴스를 시작하는 데 사용된 AMI의 플랫폼과 일치해야 합니다. Linux AMI의 경우 AMI 플랫폼이 일반 값 Linux/UNIX를 사용하는지, 아니면 SUSE Linux와 같은 보다 구체적인 값을 사용하는지 확인하는 것이 중요합니다.

Console

AMI 플랫폼을 확인하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 AMI를 선택합니다.
3. AMI를 선택합니다.
4. 세부 정보 탭에서 플랫폼 세부 정보의 값을 기록해 둡니다.

AWS CLI

AMI 플랫폼을 확인하려면

describe-images 명령을 사용하고 PlatformDetails 값을 확인합니다.

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query Images[*].PlatformDetails
```

출력의 예시는 다음과 같습니다.

```
[
    "Linux/UNIX"
]
```

PowerShell

AMI 플랫폼을 확인하려면

Get-EC2Image cmdlet을 사용하고 PlatformDetails 값을 확인합니다.

```
Get-EC2Image `
    -ImageId ami-0abcdef1234567890 | `
    Select PlatformDetails
```

출력의 예시는 다음과 같습니다.

```
PlatformDetails
---------------
Linux/UNIX
```

할당량

용량 예약이 가능한 인스턴스 수는 계정의 온디맨드 인스턴스 할당량을 기반으로 합니다. 이 할당량이 허용하는 인스턴스 수에서 이미 실행 중인 인스턴스 수를 차감한 인스턴스 수에 대해 용량을 예약할 수 있습니다.

assessing, scheduled, pending, active, delayed 상태의 용량 예약은 온디맨드 인스턴스 할당량에 포함됩니다.

제한 사항

용량 예약을 생성하기 전에 다음 제한 및 제약에 유의하세요.

- 활성 및 미사용 용량 예약 수는 온디맨드 인스턴스 제한에 포함됩니다.
- 용량 예약은 AWS 계정 간에 이전할 수 없습니다. 그러나 용량 예약은 다른 AWS 계정과 공유할 수 있습니다. 자세한 내용은 공유 용량 예약 섹션을 참조하세요.
- 영역 예약 인스턴스 결제 할인은 용량 예약에 적용되지 않습니다.
- 클러스터 배치 그룹에서 용량 예약을 생성할 수 있습니다. 분산형 및 파티션 배치 그룹은 지원되지 않습니다.
- 용량 예약은 전용 호스트와 함께 사용할 수 없습니다. 용량 예약은 전용 인스턴스와 함께 사용할 수 없습니다.
- [Windows 인스턴스] 용량 예약은 기존 보유 라이선스 사용(BYOL)과 함께 사용할 수 없습니다.
- [Red Hat 인스턴스] 용량 예약은 Bring Your Own License(BYOL)와 함께 사용할 수 없습니다.
- 최대 절전 모드 인스턴스를 시작하려고 한 후에는 용량 예약에서 해당 인스턴스를 다시 시작할 수 있도록 보장하지 않습니다.
- 최소 100개의 vCPU가 있는 인스턴스 수를 위한 장래 날짜 용량 예약을 요청할 수 있습니다. 예를 들어 `m5.xlarge` 인스턴스를 위한 장래 날짜 용량 예약을 요청하는 경우 25개 이상의 인스턴스(25 * m5.xlarge = 100 vCPU)를 요청해야 합니다.
- C, I, M, R 또는 T 인스턴스 패밀리의 인스턴스 유형에 대해서만 장래 날짜 용량 예약을 요청할 수 있습니다.

용량 예약 요금 및 결제

이 섹션의 주제에서는 용량 예약의 요금 및 결제 개요를 제공합니다.

주제

- 요금
- 결제
- 결제 할인
- 청구서 보기

요금

예약 용량에서 인스턴스를 실행하는지 여부와 무관하게 동등한 온디맨드 요금이 용량 예약에 청구됩니다. 예약을 사용하지 않는 경우 Amazon 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 리전 예약 인스턴스에 대한 결제 할인이 용량 예약에 적용됩니다. 자세한 내용은 결제 할인 섹션을 참조하세요.

자세한 정보는 Amazon EC2 요금을 참조하세요.

결제

결제는 계정에서 용량 예약이 프로비저닝되는 즉시 시작되며 용량 예약이 계정에 프로비저닝된 상태로 유지되는 동안 계속됩니다. 따라서 장래 날짜 용량 예약의 경우 요청된 장래 날짜에 계정에 용량 예약이 프로비저닝된 후에만 청구가 시작됩니다.

용량 예약은 초당 요금으로 청구됩니다. 다시 말해서 사용 시간이 한 시간 미만이라도 요금이 부과됩니다. 예를 들어 용량 예약이 24시간 15분 동안 계정에서 프로비저닝된 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 0.10 USD의 온디맨드 요율이 적용됩니다. 이 예에서는 용량 예약이 이 계정에 대해 다섯 시간 동안 프로비저닝된 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	4	5	6	Total cost
Unused Capacity Reservation	$0.10	$0.00	$0.00	$0.00	$0.00	$0.00	**$0.10**
On-demand Instance Usage	$0.00	$0.10	$0.10	$0.10	$0.10	$0.10	**$0.50**
Hourly cost	**$0.10**	**$0.10**	**$0.10**	**$0.10**	**$0.10**	**$0.10**	**$0.60**

결제 할인

Savings Plans 및 리전 예약 인스턴스에 대한 청구 할인이 용량 예약에 적용됩니다. AWS는 이 할인을 속성이 일치하는 용량 예약에 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 정보는 다음을 참조하세요.

- Amazon EC2용 예약 인스턴스 개요
- 절감형 플랜 사용 설명서
- 청구 및 구매 옵션

청구서 보기

계정으로 청구되는 요금과 비용은 AWS 결제 및 비용 관리 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 용량 예약 항목을 참조하세요.

용량 예약 생성

언제든지 용량 예약을 생성하여 특정 가용 영역에서 사용할 수 있는 컴퓨팅 용량을 확보할 수 있습니다. 용량 예약은 즉시 시작하거나 장래 날짜에 시작할 수 있습니다. 용량 예약이 `active` 상태가 되어야만 용량을 사용할 수 있습니다.

> **Note**
>
> `open` 인스턴스 일치 기준으로 용량 예약을 생성하고 용량 예약이 활성화되는 시점에 일치하는 속성이 있는 인스턴스를 실행 중인 경우, 해당 인스턴스는 예약 용량에서 자동으로 실행됩니다. 이를 방지하려면 `targeted` 인스턴스 일치 기준을 사용하세요. 자세한 내용은 인스턴스 일치 기준 섹션을 참조하세요.

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, 요청을 줄여서 시도하세요. 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 시도하세요. 자세한 내용은 온디맨드 인스턴스 할당량 섹션을 참조하세요.

주제

- 즉시 사용할 수 있는 용량 예약 생성
- 장래 날짜 용량 예약 생성

즉시 사용할 수 있는 용량 예약 생성

즉시 사용할 수 있는 용량 예약을 생성합니다.

Console

용량 예약을 생성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 용량 예약를 선택한 후 용량 예약 생성을 선택합니다.
3. 인스턴스 세부 정보 섹션에서 다음 설정을 구성합니다.
   a. 인스턴스 유형 - 용량을 예약할 인스턴스 유형입니다.
   b. 플랫폼 - 인스턴스에 사용할 운영 체제입니다. 자세한 내용은 지원되는 플랫폼 섹션을 참조하세요.
   c. 가용 영역 - 용량을 예약할 가용 영역입니다.
   d. 테넌시 - 예약 용량에 사용할 테넌시 유형입니다. 공유 하드웨어에서 용량을 예약하려면 기본값을 선택하고 계정 전용 하드웨어에서 용량을 예약하려면 전용을 선택합니다.
   e. (선택 사항) 배치 그룹 ARN - 용량 예약을 생성할 클러스터 배치 그룹의 ARN입니다. 자세한 내용은 클러스터 배치 그룹에 용량 예약 생성 섹션을 참조하세요.
   f. 총 인스턴스 수 - 용량을 예약할 인스턴스 수입니다. 선택한 인스턴스 유형에 남은 온디맨드 인스턴스 할당량을 초과하는 수량을 지정하는 경우 요청이 실패합니다.
4. 예약 세부 정보 섹션에서 다음 설정을 구성합니다.
   a. 용량 예약 시작 - 즉시를 선택합니다.
   b. 용량 예약 종료 - 다음 옵션 중 하나를 선택합니다.
      - 수동 - 명시적으로 취소할 때까지 용량을 예약합니다.

- 특정 시간 - 지정된 날짜 및 시간에 용량 예약을 자동으로 취소합니다.

c. 인스턴스 자격 - 다음 옵션 중 하나를 선택합니다.

- open - (기본값) 용량 예약은 일치하는 속성(인스턴스 유형, 플랫폼, 가용 영역, 테넌시)이 있는 모든 인스턴스에 연결됩니다. 일치하는 속성이 있는 인스턴스를 시작할 경우 예약 용량으로 자동 배치됩니다.
- targeted - 용량 예약은 일치하는 속성(인스턴스 유형, 플랫폼, 가용 영역, 테넌시)이 있고 예약을 명시적으로 대상 지정하는 인스턴스만 허용합니다.

5. 생성(Create)을 선택합니다.

AWS CLI

용량 예약을 생성하려면

create-capacity-reservation 명령을 사용합니다.

```
aws ec2 create-capacity-reservation \
    --availability-zone az_name \
    --instance-type instance_type \
    --instance-count number_of_instances \
    --instance-platform operating_system \
    --instance-match-criteria open|targeted
```

PowerShell

용량 예약을 생성하려면

Add-EC2CapacityReservation cmdlet을 사용합니다.

```
Add-EC2CapacityReservation `
    -AvailabilityZone az_name `
    -InstanceType instance_type `
    -InstanceCount number_of_instances `
    -InstancePlatform operating_system `
    -InstanceMatchCriterion open|targeted
```

장래 날짜 용량 예약 생성

장래 날짜 및 시간에 예약 용량을 사용하려면 장래 날짜 용량 예약을 요청합니다.

장래 날짜 용량 예약 요청은 지원 가능 여부를 결정하기 위한 평가를 거칩니다. 자세한 내용은 장래 날짜 용량 예약 평가 섹션을 참조하세요.

고려 사항

- C, I, M, R 또는 T 인스턴스 패밀리의 인스턴스 유형에 대해서만 장래 날짜의 용량 예약을 요청할 수 있습니다. 자세한 내용은 Amazon EC2 인스턴스 유형 명명 규칙을 참조하세요.
- 최소 100개의 vCPU가 있는 인스턴스 수를 위한 장래 날짜 용량 예약을 요청할 수 있습니다. 예를 들어 `m5.xlarge` 인스턴스를 위한 장래 날짜 용량 예약을 요청하는 경우 25개 이상의 인스턴스에 대한 용량(25 * m5.xlarge = 100 vCPU)을 요청해야 합니다.
- 5~120일 전에 장래 날짜 용량 예약을 요청할 수 있습니다. 하지만 지원 가능성을 높이려면 최소 56일(8주) 전에 요청하는 것이 좋습니다.
- 최소 약정 기간은 14일입니다.

Console

용량 예약을 생성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 용량 예약를 선택한 후 용량 예약 생성을 선택합니다.
3. 인스턴스 세부 정보 섹션에서 다음 설정을 구성합니다.
   a. 인스턴스 유형 - 용량을 예약할 인스턴스 유형입니다.
   b. 플랫폼 - 인스턴스에 사용할 운영 체제입니다. 자세한 내용은 지원되는 플랫폼 섹션을 참조하세요.
   c. 가용 영역 - 용량을 예약할 가용 영역입니다.
   d. 테넌시 - 예약 용량에 사용할 테넌시 유형입니다. 공유 하드웨어에서 용량을 예약하려면 기본값을 선택하고 계정 전용 하드웨어에서 용량을 예약하려면 전용을 선택합니다.
   e. 총 인스턴스 수 - 용량을 예약할 인스턴스 수입니다. 선택한 인스턴스 유형에 남은 온디맨드 인스턴스 할당량을 초과하는 수량을 지정하는 경우 요청이 실패합니다.
4. 예약 세부 정보 섹션에서 다음 설정을 구성합니다.
   a. 용량 예약 시작 - 특정 시간을 선택합니다.
   b. 시작 날짜 - 용량 예약을 사용해야 하는 날짜와 시간을 지정합니다. 자세한 내용은 현재 날짜 및 시간 섹션을 참조하세요.

c. 약정 기간 - 용량 예약이 제공된 후 용량 예약을 유지하기로 약정하는 최소 기간을 지정합니다. 자세한 내용은 약정 기간 섹션을 참조하세요.

d. 용량 예약 종료 - 다음 옵션 중 하나를 선택합니다.

- 취소할 때 - 명시적으로 취소할 때까지 용량을 예약합니다.
- 특정 시간 - 지정된 날짜 및 시간에 용량 예약을 자동으로 취소합니다.

5. 생성(Create)을 선택합니다.

AWS CLI

용량 예약을 생성하려면

create-capacity-reservation 명령을 사용합니다.

```
aws ec2 create-capacity-reservation \
    --availability-zone az_name \
    --instance-type instance_type \
    --instance-count number_of_instances \
    --instance-platform operating_system \
    --instance-match-criteria targeted \
    --delivery-preference incremental \
    --commitment-duration commitment_in_seconds \
    --start-date YYYY-MMDDThh:mm:ss.sssZ
```

PowerShell

용량 예약을 생성하려면

Add-EC2CapacityReservation cmdlet을 사용합니다.

```
Add-EC2CapacityReservation `
    -AvailabilityZone az_name `
    -InstanceType instance_type `
    -InstanceCount number_of_instances `
    -InstancePlatform operating_system `
    -InstanceMatchCriterion targeted `
    -DeliveryPreference incremental `
    -CommitmentDuration commitment_in_seconds `
    -StartDate  YYYY-MMDDThh:mm:ss.sssZ
```

용량 예약 상태 보기

Amazon EC2는 용량 예약 상태를 지속적으로 모니터링합니다.

Amazon EC2 API에서 따르는 최종 일관성 모델로 인해 용량 예약 생성 후 용량 예약이 `active` 상태로 표시되려면 최대 5분이 걸릴 수 있습니다. 이 시간 동안 용량 예약은 `pending` 상태로 유지될 수 있습니다. 하지만 용량 예약이 이미 사용 가능할 수도 있으며, 이 경우 인스턴스를 용량 예약으로 시작하려는 시도가 성공합니다.

Console

용량 예약을 보는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 용량 예약를 선택합니다.
3. 용량 예약을 선택합니다.

AWS CLI

용량 예약을 설명하려면

describe-capacity-reservations 명령을 사용합니다.

예를 들어, 다음 명령은 모두 용량 예약를 설명합니다.

```
aws ec2 describe-capacity-reservations
```

출력의 예시는 다음과 같습니다.

```
{
    "CapacityReservations": [
        {
            "CapacityReservationId": "cr-1234abcd56EXAMPLE",
            "EndDateType": "unlimited",
            "AvailabilityZone": "eu-west-1a",
            "InstanceMatchCriteria": "open",
            "Tags": [],
            "EphemeralStorage": false,
            "CreateDate": "2019-08-16T09:03:18.000Z",
            "AvailableInstanceCount": 1,
            "InstancePlatform": "Linux/UNIX",
            "TotalInstanceCount": 1,
```

```
            "State": "active",
            "Tenancy": "default",
            "EbsOptimized": true,
            "InstanceType": "a1.medium",
            "PlacementGroupArn": "arn:aws:ec2:us-east-1:123456789012:placement-
group/MyPG"
        },
        {
            "CapacityReservationId": "cr-abcdEXAMPLE9876ef",
            "EndDateType": "unlimited",
            "AvailabilityZone": "eu-west-1a",
            "InstanceMatchCriteria": "open",
            "Tags": [],
            "EphemeralStorage": false,
            "CreateDate": "2019-08-07T11:34:19.000Z",
            "AvailableInstanceCount": 3,
            "InstancePlatform": "Linux/UNIX",
            "TotalInstanceCount": 3,
            "State": "cancelled",
            "Tenancy": "default",
            "EbsOptimized": true,
            "InstanceType": "m5.large"
        }
    ]
}
```

PowerShell

용량 예약을 설명하려면

Get-EC2CapacityReservation cmdlet을 사용합니다.

```
Get-EC2CapacityReservation `
    -CapacityReservationId cr-1234abcd56EXAMPLE
```

용량 예약 상태

용량 예약은 다음과 같은 상태가 가능합니다.

State	설명
active	- 용량을 사용할 수 있습니다.

State	설명
`expired`	예약 요청에서 지정한 날짜 및 시간에 용량 예약이 자동으로 만료됩니다. 예약 용량을 더 이상 사용할 수 없습니다.
`cancelled`	용량 예약이 취소되었습니다. 예약 용량을 더 이상 사용할 수 없습니다.
`pending`	용량 예약 요청이 성공했지만 용량 프로비저닝이 여전히 대기 중입니다.
`failed`	용량 예약 요청이 실패했습니다. 잘못된 요청 파라미터, 용량 제약 조건 또는 인스턴스 제한 제약 조건이 요청 실패의 원인일 수 있습니다. 60분 동안 실패한 요청을 볼 수 있습니다.
`scheduled`	(장래 날짜 용량 예약만 해당) 장래 날짜 용량 예약 요청이 승인되었으며 요청된 시작 날짜에 용량 예약이 제공될 예정입니다.
`assessing`	(장래 날짜 용량 예약만 해당) Amazon EC2가 장래 날짜 용량 예약 요청을 평가하고 있습니다. 자세한 내용은 장래 날짜 용량 예약 평가 섹션을 참조하세요.
`delayed`	(장래 날짜 용량 예약만 해당) Amazon EC2에서 요청된 장래 날짜 용량 예약 프로비저닝이 지연되었습니다. Amazon EC2가 요청된 시작 날짜 및 시간까지 요청된 용량을 제공할 수 없습니다.
`unsupported`	(장래 날짜 용량 예약만 해당) Amazon EC2가 용량 제약으로 인해 장래 날짜 용량 예약 요청을 지원할 수 없습니다. 지원되지 않는 요청을 30일 동안 볼 수 있습니다. 용량 예약이 제공되지 않습니다.

인스턴스를 기존 용량 예약으로 시작

다음과 같은 용량 예약에서만 인스턴스를 시작할 수 있습니다.

- 일치하는 속성(인스턴스 유형, 플랫폼, 가용 영역, 테넌시)이 있는 용량 예약
- 사용 가능한 용량이 충분한 용량 예약
- 상태가 `active`인 용량 예약

인스턴스를 시작할 때 인스턴스를 임의 open 용량 예약, 특정 용량 예약 또는 용량 예약 그룹으로 시작할지 여부를 지정할 수 있습니다.

또는 일치하는 속성 및 가용 용량이 있는 open 용량 예약이 있더라도 용량 예약에서 실행되지 않도록 인스턴스를 구성할 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.

Console

인스턴스를 기존 용량 예약으로 시작하려면

1. 절차에 따라 인스턴스를 시작하되 다음 단계를 완료하여 배치 그룹 및 용량 예약 설정을 지정할 때까지 인스턴스를 시작하지 마세요.
2. 고급 세부 정보를 열고 다음을 수행합니다.
   a. 배치 그룹의 경우 인스턴스를 시작할 클러스터 배치 그룹을 선택합니다.
   b. 용량 예약(Capacity Reservation)에서 용량 예약 구성에 따라 다음 옵션 중 하나를 선택합니다.
      - 없음 - 인스턴스가 용량 예약으로 시작되지 않도록 합니다. 인스턴스는 온디맨드 용량으로 실행됩니다.
      - 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 지정 - 선택한 용량 예약에서 인스턴스를 시작합니다. 선택한 용량 예약의 용량이 선택한 인스턴스 수에 맞게 충분하지 않으면 인스턴스가 시작되지 않습니다.
      - 용량 예약 리소스 그룹 지정 - 선택한 용량 예약 그룹에 일치하는 속성 및 가용 용량이 있는 용량 예약에서 인스턴스를 시작합니다. 선택한 그룹에 일치하는 속성 및 가용 용량이 있는 용량 예약이 없는 경우 인스턴스는 온디맨드 용량으로 시작됩니다.
      - 용량 예약만 지정 - 선택한 용량 예약에서 인스턴스를 시작합니다. 용량 예약 ID를 지정하지 않으면 인스턴스는 open 용량 예약에서 시작됩니다. 용량을 사용할 수 없는 경우 인스턴스 시작이 실패합니다.
      - 용량 예약 리소스 그룹만 지정 - 용량 예약 리소스 그룹의 용량 예약에서 인스턴스를 시작합니다. 용량 예약 리소스 그룹 ARN을 지정하지 않으면 인스턴스는 open 용량 예약에서 시작됩니다. 용량을 사용할 수 없는 경우 인스턴스 시작이 실패합니다.
3. Summary(요약) 패널에서 인스턴스 구성을 검토한 다음 Launch instance(인스턴스 시작)를 선택합니다.

AWS CLI

인스턴스를 기존 용량 예약으로 시작하려면

run-instances 명령을 사용하고 `--capacity-reservation-specification` 옵션을 지정합니다.

다음 예제는 일치하는 속성 및 가용 용량이 있는 열려 있는 용량 예약으로 인스턴스를 시작합니다.

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --count 1 \
    --instance-type t2.micro \
    --key-name my-key-pair \
    --subnet-id subnet-0abcdef1234567890 \
    --capacity-reservation-specification CapacityReservationPreference=open
```

다음 예제는 `targeted` 용량 예약으로 인스턴스를 시작합니다.

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --count 1 \
    --instance-type t2.micro \
    --key-name my-key-pair \
    --subnet-id subnet-0abcdef1234567890 \
    --capacity-reservation-specification \
        CapacityReservationTarget={CapacityReservationId=cr-1234abcd56EXAMPLE}
```

다음 예제는 지정된 용량 예약 그룹으로 인스턴스를 시작합니다.

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --count 1 \
    --instance-type t2.micro \
    --key-name my-key-pair \
    --subnet-id subnet-0abcdef1234567890 \
    --capacity-reservation-specification \

 CapacityReservationTarget={CapacityReservationResourceGroupArn=arn:aws:resource-
groups:us-west-2:123456789012:group/my-cr-group}
```

다음 예제는 용량 예약으로만 인스턴스를 시작합니다. 용량 예약 ID가 지정되지 않았으므로 인스턴스는 일치하는 속성 및 사용 가능한 용량이 있는 임의의 열려 있는 용량 예약에서 시작됩니다.

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --count 1 \
    --instance-type t2.micro \
    --key-name my-key-pair \
    --subnet-id subnet-0abcdef1234567890 \
    --capacity-reservation-specification \
        CapacityReservationPreference=capacity-reservations-only
```

다음 예제는 특정 용량 예약으로만 인스턴스를 시작합니다. 지정된 용량 예약에서 용량을 사용할 수 없는 경우 인스턴스 시작이 실패합니다.

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --count 1 \
    --instance-type t2.micro \
    --key-name my-key-pair \
    --subnet-id subnet-0abcdef1234567890 \
    --capacity-reservation-specification \
        CapacityReservationPreference=capacity-reservations-only \
        CapacityReservationTarget={CapacityReservationId=cr-1234abcd56EXAMPLE}
```

PowerShell

인스턴스를 기존 용량 예약으로 시작하려면

New-EC2Instance cmdlet을 사용합니다.

다음 예제는 일치하는 속성 및 가용 용량이 있는 열려 있는 용량 예약으로 인스턴스를 시작합니다.

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType t2.micro `
    -KeyName "my-key-pair" `
    -SubnetId subnet-0abcdef1234567890 `
    -CapacityReservationSpecification_CapacityReservationPreference "open"
```

다음 예제는 `targeted` 용량 예약으로 인스턴스를 시작합니다.

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType t2.micro `
    -KeyName "my-key-pair" `
    -SubnetId subnet-0abcdef1234567890 `
    -CapacityReservationTarget_CapacityReservationId cr-1234abcd56EXAMPLE
```

다음 예제는 지정된 용량 예약 그룹으로 인스턴스를 시작합니다.

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType t2.micro `
    -KeyName "my-key-pair" `
    -SubnetId subnet-0abcdef1234567890 `
    -CapacityReservationTarget_CapacityReservationResourceGroupArn `
        "arn:aws:resource-groups:us-west-2:123456789012:group/my-cr-group"
```

다음 예제는 용량 예약으로만 인스턴스를 시작합니다. 용량 예약 ID가 지정되지 않았으므로 인스턴스는 일치하는 속성 및 사용 가능한 용량이 있는 임의의 열려 있는 용량 예약에서 시작됩니다.

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType t2.micro `
    -KeyName "my-key-pair" `
    -SubnetId subnet-0abcdef1234567890 `
    -CapacityReservationSpecification_CapacityReservationPreference "capacity-
reservations-only"
```

다음 예제는 특정 용량 예약으로만 인스턴스를 시작합니다. 지정된 용량 예약에서 용량을 사용할 수 없는 경우 인스턴스 시작이 실패합니다.

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType t2.micro `
    -KeyName "my-key-pair" `
    -SubnetId subnet-0abcdef1234567890 `
    -CapacityReservationSpecification_CapacityReservationPreference "capacity-
reservations-only" `
    -CapacityReservationTarget_CapacityReservationId cr-1234abcd56EXAMPLE
```

활성 용량 예약 수정

용량이 필요한 워크로드에 기존 용량 예약이 적합하지 않은 경우 인스턴스 수량, 인스턴스 자격(open 또는 targeted), 종료 시간(At specific time 또는 Manually)을 수정할 수 있습니다. 선택한 인스턴스 유형에 대해 남은 온디맨드 인스턴스 제한을 초과하는 새 인스턴스 수량을 지정하는 경우 업데이트가 실패합니다.

허용되는 수정은 용량 예약의 상태에 따라 달라집니다.

- assessing 또는 scheduled 상태 - 태그만 수정할 수 있습니다.
- pending 상태 - 어떤 식으로도 용량 예약을 수정할 수 없습니다.
- active 상태이지만 여전히 약정 기간 내 - 인스턴스 수를 약정된 인스턴스 수 미만으로 줄이거나 약정된 기간 이전의 종료 날짜를 설정할 수 없습니다. 다른 모든 수정은 허용됩니다.
- 약정 기간이 없거나 약정 기간이 경과한 active 상태 - 모든 수정이 허용됩니다.
- expired, cancelled, unsupported 또는 failed 상태 - 어떤 식으로도 용량 예약을 수정할 수 없습니다.

고려 사항

- 생성 후에는 인스턴스 유형, 플랫폼, 가용 영역 또는 테넌시를 변경할 수 없습니다. 이러한 속성을 수정해야 하는 경우에는 예약을 취소하고 나서 필요한 속성으로 예약을 다시 생성하는 것이 좋습니다.
- targeted에서 open으로 인스턴스 자격을 변경하여 기존 용량 예약을 수정하는 경우, 용량 예약의 속성과 일치하고 CapacityReservationPreference 파라미터가 open으로 설정되어 있으며 아직 용량 예약에서 실행되지 않은 실행 중인 인스턴스는 자동으로 수정된 용량 예약을 사용합니다.
- 인스턴스 자격을 변경하려면 용량 예약이 완전히 유휴 상태여야 합니다(사용량 0). 예약 내에서 인스턴스가 실행 중일 때는 Amazon EC2에서 인스턴스 자격을 수정할 수 없기 때문입니다.

Console

용량 예약을 수정하려면

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 총 용량, 용량 예약 종료 또는 인스턴스 자격 옵션을 수정하고 저장을 선택합니다.

AWS CLI

용량 예약을 수정하려면

modify-capacity-reservation 명령을 사용합니다. 다음 예제는 지정된 용량 예약을 수정하여 8개의 인스턴스에 대한 용량을 예약합니다.

```
aws ec2 modify-capacity-reservation \
    --capacity-reservation-id cr-1234567890abcdef0 \
    --instance-count 8
```

PowerShell

용량 예약을 수정하려면

Edit-EC2CapacityReservation cmdlet을 사용합니다. 다음 예제는 지정된 용량 예약을 수정하여 8개의 인스턴스에 대한 용량을 예약합니다.

```
Edit-EC2CapacityReservation `
    -CapacityReservationId cr-1234567890abcdef0 `
    -InstanceCount 8
```

인스턴스의 용량 예약 설정 수정

중지된 인스턴트에 대한 다음 용량 예약 설정은 다음과 같이 언제든지 수정할 수 있습니다.

- 속성(인스턴스 유형, 플랫폼, 가용 영역, 테넌시)과 사용 가능한 용량이 일치하는 용량 예약에서 시작하세요.
- 특정 용량 예약에서 인스턴스를 시작합니다.
- 용량 예약 그룹에 일치하는 속성 및 가용 용량이 있는 용량 예약에서 시작합니다.
- 인스턴스가 용량 예약에서 시작되지 않도록 합니다.

Console

인스턴스 용량 예약 설정을 수정하려면

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, 인스턴스 설정, 용량 예약 설정 수정을 선택합니다.
4. 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.
   - 없음 - 인스턴스가 용량 예약으로 시작되지 않도록 합니다. 인스턴스는 온디맨드 용량으로 실행됩니다.
   - 용량 예약 지정 - 선택한 용량 예약에서 인스턴스를 시작합니다. 선택한 용량 예약의 용량이 선택한 인스턴스 수에 맞게 충분하지 않으면 인스턴스가 시작되지 않습니다.
   - 용량 예약 그룹 지정 - 선택한 용량 예약 그룹에 일치하는 속성 및 가용 용량이 있는 용량 예약에서 인스턴스를 시작합니다. 선택한 그룹에 일치하는 속성 및 가용 용량이 있는 용량 예약이 없는 경우 인스턴스는 온디맨드 용량으로 시작됩니다.
   - 용량 예약만 지정 - 선택한 용량 예약에서 인스턴스를 시작합니다. 용량 예약 ID를 지정하지 않으면 인스턴스는 open 용량 예약에서 시작됩니다. 용량을 사용할 수 없는 경우 인스턴스 시작이 실패합니다.
   - 용량 예약 리소스 그룹만 지정 - 용량 예약 리소스 그룹의 용량 예약에서 인스턴스를 시작합니다. 용량 예약 리소스 그룹 ARN을 지정하지 않으면 인스턴스는 open 용량 예약에서 시작됩니다. 용량을 사용할 수 없는 경우 인스턴스 시작이 실패합니다.

AWS CLI

인스턴스 용량 예약 설정을 수정하려면

modify-instance-capacity-reservation-attributes 명령을 사용합니다.

다음 예제는 용량 예약 기본 설정을 none으로 변경합니다.

```
aws ec2 modify-instance-capacity-reservation-attributes \
    --instance-id i-1234567890abcdef0 \
    --capacity-reservation-specification CapacityReservationPreference=none
```

다음 예제는 대상을 특정 용량 예약으로 변경합니다.

```
aws ec2 modify-instance-capacity-reservation-attributes \
    --instance-id i-1234567890abcdef0 \
    --capacity-reservation-specification \
    CapacityReservationTarget={CapacityReservationId=cr-1234567890abcdef0}
```

다음 예제는 대상을 특정 용량 예약 그룹으로 지정합니다.

```
aws ec2 modify-instance-capacity-reservation-attributes \
    --instance-id i-1234567890abcdef0 \
    --capacity-reservation-specification \

 CapacityReservationTarget={CapacityReservationResourceGroupArn=arn:aws:resource-
groups:us-west-2:123456789012:group/my-cr-group}
```

다음 예제는 용량 예약 기본 설정을 capacity-reservation-only으로 변경합니다. 용량 예약이 지정되지 않았으므로 인스턴스는 일치하는 속성 및 사용 가능한 용량이 있는 임의의 열려 있는 용량 예약에서 시작됩니다.

```
aws ec2 modify-instance-capacity-reservation-attributes \
    --instance-id i-1234567890abcdef0 \
    --capacity-reservation-specification CapacityReservationPreference=capacity-
reservation-only
```

다음 예제는 용량 예약 기본 설정을 capacity-reservation-only로 변경하고 대상을 특정 용량 예약으로 변경합니다. 지정된 용량 예약에서 용량을 사용할 수 없는 경우 인스턴스 시작이 실패합니다.

```
aws ec2 modify-instance-capacity-reservation-attributes \
    --instance-id i-1234567890abcdef0 \
    --capacity-reservation-specification \
        CapacityReservationPreference=capacity-reservation-only \
        CapacityReservationTarget={CapacityReservationId=cr-1234567890abcdef0}
```

PowerShell

인스턴스 용량 예약 설정을 수정하려면

Edit-EC2InstanceCapacityReservationAttribute cmdlet을 사용합니다.

다음 예제는 용량 예약 기본 설정을 none으로 변경합니다.

```
Edit-EC2InstanceCapacityReservationAttribute `
    -InstanceId i-1234567890abcdef0 `
    -CapacityReservationSpecification_CapacityReservationPreference "none"
```

다음 예제는 대상을 특정 용량 예약으로 변경합니다.

```
Edit-EC2InstanceCapacityReservationAttribute `
    -InstanceId i-1234567890abcdef0 `
    -CapacityReservationTarget_CapacityReservationId cr-1234567890abcdef0
```

다음 예제는 대상을 특정 용량 예약 그룹으로 지정합니다.

```
Edit-EC2InstanceCapacityReservationAttribute `
    -InstanceId i-1234567890abcdef0 `
    -CapacityReservationTarget_CapacityReservationResourceGroupArn `
        "arn:aws:resource-groups:us-west-2:123456789012:group/my-cr-group"
```

다음 예제는 용량 예약 기본 설정을 capacity-reservation-only으로 변경합니다. 용량 예약이 지정되지 않았으므로 인스턴스는 일치하는 속성 및 사용 가능한 용량이 있는 임의의 열려 있는 용량 예약에서 시작됩니다.

```
Edit-EC2InstanceCapacityReservationAttribute `
    -InstanceId i-1234567890abcdef0 `
    -CapacityReservationSpecification_CapacityReservationPreference "capacity-
reservation-only"
```

다음 예제는 용량 예약 기본 설정을 capacity-reservation-only로 변경하고 대상을 특정 용량 예약으로 변경합니다. 지정된 용량 예약에서 용량을 사용할 수 없는 경우 인스턴스 시작이 실패합니다.

```
Edit-EC2InstanceCapacityReservationAttribute `
    -InstanceId i-1234567890abcdef0 `
    -CapacityReservationSpecification_CapacityReservationPreference "capacity-
reservation-only" `
    -CapacityReservationTarget_CapacityReservationId cr-1234567890abcdef0
```

용량 예약 간 용량 이동

한 용량 예약에서 다른 용량 예약으로 용량을 이동하여 필요에 따라 예약된 컴퓨팅 리소스를 재분배할 수 있습니다. 예를 들어 사용량이 늘어나면서 용량 예약에 추가 용량이 필요하고 다른 용량 예약에 가용 용량이 있는 경우, 두 예약 간에 용량을 재할당할 수 있습니다.

용량 이동을 위한 전제 조건

두 용량 예약은 다음 요구 사항을 전제 조건으로 충족해야 합니다.

- 두 예약 모두 활성 상태여야 합니다.
- 두 예약 모두 사용자 AWS 계정의 소유여야 합니다. 다른 AWS 계정이 소유한 예약 간에는 용량을 이동할 수 없습니다.
- 두 예약 모두 다음이 동일해야 합니다.
  - 인스턴스 유형
  - 플랫폼
  - 가용 영역
  - Tenancy
  - 배치 그룹
  - 종료 시간

대상 용량 예약 인스턴스 자격(open 또는 `targeted`)과 태그가 원본 예약과 일치할 필요는 없습니다. 원본 예약의 용량은 줄어들고 대상 예약의 용량은 증가했다는 점을 제외하고 두 예약의 구성은 동일하게 유지됩니다.

이동할 인스턴스의 양을 지정할 때 기본적으로 사용 가능한 용량이 먼저 이동되고, 그 다음으로 실행 중인 적격 인스턴스(예약에서 사용된 용량)가 이동됩니다. 예를 들어 사용된 인스턴스 5개와 사용 가능한 인스턴스 3개가 있는 예약에서 인스턴스 4개를 이동하면 사용 가능한 인스턴스 3개와 사용된 인스턴스 1개가 이동합니다.

> **Note**
>
> 사용 가능한 용량보다 큰 이동할 수량을 지정하여 예약에서 사용된 용량을 이동하면 용량 예약 사양이 open인 시작된 인스턴스만 이동됩니다.

고려 사항

한 예약에서 다른 예약으로 용량을 이동할 때 다음 고려 사항이 적용됩니다.

- 사용된 용량은 동일한 계정 집합과 공유되는 open 인스턴스 자격 요건을 가진 용량 예약 간에만 이동할 수 있습니다.
- 사용된 용량을 이동하면 적격 인스턴스가 무작위로 선택됩니다. 실행 중인 어떤 인스턴스가 이동되는지 지정할 수는 없습니다. 이동 수량을 충족할 충분한 수의 적격 인스턴스를 찾지 못하면 이동 작업이 실패합니다.
- 원본 예약에서 모든 용량을 이동하면 용량 예약이 자동으로 취소됩니다.

- 장래 날짜 용량 예약 - 약정 기간 동안 장래 날짜 용량 예약의 용량을 이동할 수 없습니다.

> **Note**
>
> 용량 블록에서 용량을 이동하는 것은 지원되지 않습니다.

용량 이동

용량을 소스 용량 예약에서 대상 용량 예약으로 이동할 수 있습니다.

Console

용량을 이동하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 용량 예약을 선택합니다.
3. 이동할 용량이 있는 온디맨드 용량 예약 ID를 선택합니다.
4. 작업, 용량 관리에서 이동을 선택합니다.
5. 용량 이동 페이지의 대상 용량 예약 아래 목록에서 예약을 선택합니다.
6. 이동할 수량에서 슬라이더를 사용하거나 원본 용량 예약에서 대상 용량 예약으로 이동할 인스턴스 수를 입력합니다.
7. 요약을 검토하고 준비가 되면 이동을 선택합니다.

AWS CLI

용량을 이동하려면

move-capacity-reservation-instances 명령을 사용합니다. 다음 예제는 10개의 인스턴스를 지정된 소스 용량 예약에서 지정된 대상 용량 예약으로 이동합니다.

```
aws ec2 move-capacity-reservation-instances \
    --source-capacity-reservation-id cr-1234567890abcdef0 \
    --destination-capacity-reservation-id cr-021345abcdef56789 \
    --instance-count 10
```

PowerShell

용량을 이동하려면

Move-EC2CapacityReservationInstance cmdlet을 사용합니다. 다음 예제는 10개의 인스턴스를 지정된 소스 용량 예약에서 지정된 대상 용량 예약으로 이동합니다.

```
Move-EC2CapacityReservationInstance `
    -SourceCapacityReservationId cr-1234567890abcdef0 `
    -DestinationCapacityReservationId cr-021345abcdef56789 `
    -InstanceCount 10
```

기존 용량 예약에서 용량 분할

기존 용량 예약에서 용량을 분할하여 새 용량 예약을 생성할 수 있습니다. 용량을 분할하면 원래 예약의 일부를 특정 워크로드에 할당하거나 다른 AWS 계정와 공유할 수 있습니다. 예를 들어 용량 예약을 다른 계정과 부분적으로 공유하려면 용량 중 일부를 분할하여 더 작은 크기의 용량 예약을 생성합니다. 그러면 AWS Resource Access Manager를 사용하여 더 작은 크기의 용량 예약을 다른 계정과 공유할 수 있습니다.

기존 용량 예약에서 용량을 분할하면 새 용량 예약이 자동으로 생성됩니다. 분할된 인스턴스 수로 인해 총 용량이 감소하는 것을 제외하고 기존 용량 예약은 변경되지 않습니다. 기존 용량 예약에서 실행 중인 인스턴스는 영향을 받지 않습니다. 기존 예약을 하나의 새 용량 예약으로만 분할할 수 있습니다.

새 용량 예약은 태그를 제외하고 기존 용량 예약과 동일한 구성을 갖습니다. 기본적으로 새 용량 예약에는 태그가 없습니다. 분할 작업 중 새 태그를 지정할 수 있습니다. 필요한 경우 새 용량 예약을 생성한 후 수정할 수도 있습니다.

분할할 인스턴스의 양을 지정할 때 기본적으로 사용 가능한 용량이 먼저 분할되고, 그 다음으로 자격이 있는 실행 중인 인스턴스(예약에서 사용된 용량)가 분할됩니다. 예를 들어 사용된 인스턴스 5개와 사용 가능한 인스턴스 3개가 있는 용량 예약에서 인스턴스 4개를 분할하면 사용 가능한 인스턴스 3개와 사용된 인스턴스 1개가 새 예약으로 분할됩니다.

용량 분할을 위한 전제 조건

용량 예약은 다음 요구 사항을 전제 조건으로 충족해야 합니다.

- 원본 예약이 활성 상태여야 합니다.
- 원본 예약이 사용자 AWS 계정의 소유여야 합니다.

> **Note**
>
> 사용 가능한 용량보다 큰 분할할 수량을 지정하여 예약에서 사용된 용량을 분할하면 용량 예약 사양이 open인 시작된 인스턴스만 분할됩니다.

고려 사항

한 예약에서 새 예약으로 용량을 분할할 때 다음 고려 사항이 적용됩니다.

- 사용된 용량은 어떤 계정과도 공유되지 않는 'open' 인스턴스 자격이 있는 용량 예약인 경우에만 분할할 수 있습니다.
- 사용된 용량을 분할하면 적격 인스턴스가 무작위로 선택됩니다. 실행 중인 어떤 인스턴스가 분할되는지는 지정할 수 없습니다. 분할 수량을 충족할 충분한 수의 적격 인스턴스를 찾지 못하면 분할 작업이 실패합니다.
- 기존 예약에서 분할할 수 있는 최대 인스턴스 수는 예약 크기에서 1을 뺀 값입니다. 예를 들어 예약의 총 용량이 인스턴스 5개인 경우 최대 4개의 인스턴스를 새 예약으로 분할할 수 있습니다.
- 장래 날짜 용량 예약 - 약정 기간 동안 장래 날짜 용량 예약의 용량을 분할할 수 없습니다.
- 리소스 그룹 - 기존 용량 예약이 리소스 그룹에 속하는 경우 새 용량 예약이 리소스 그룹에 자동으로 추가되지 않습니다. 필요한 경우 새 용량 예약을 생성한 후 리소스 그룹에 추가할 수 있습니다.
- 공유 - 기존 용량 예약을 소비자 계정과 공유하는 경우 새 용량 예약이 소비자 계정과 자동으로 공유되지 않습니다. 필요한 경우 새 용량 예약을 생성한 후 공유할 수 있습니다.
- 클러스터 배치 그룹 - 기존 용량 예약이 클러스터 배치 그룹의 일부인 경우 동일한 클러스터 배치 그룹에 새 용량 예약이 생성됩니다.

> **Note**
>
> 용량 블록에서 용량을 분할하는 것은 지원되지 않습니다.

태그를 사용하여 용량 예약 분할을 위한 액세스 제어

태그를 사용하여 Amazon EC2 리소스에 대한 액세스를 제어할 수 있습니다. 여기에는 기존 용량 예약에서 용량을 분할하여 새 용량 예약을 생성하는 것도 포함됩니다. 자세한 내용은 IAM 사용 설명서의 태그를 사용하여 AWS 리소스에 대한 액세스 제어를 참조하세요.

태그를 사용하여 용량 예약 분할에 대한 액세스를 제어하려면 IAM 정책이 원본 용량 예약과 새로 생성된 용량 예약 모두에 대해 평가되므로 정책 문에 리소스 태그와 요청 태그를 모두 지정해야 합니다. 다음 예제 정책에는 원본 용량 예약에 대한 `Owner=ExampleDepartment1` 태그가 있는 `ec2:ResourceTag` 조건 키와 새로 생성된 용량 예약에 대한 `stack=production` 태그가 있는 `ec2:RequestTag` 조건 키가 포함되어 있습니다.

```
{
  "Statement": [
    {
      "Sid": "AllowSourceCapacityReservation",
      "Effect": "Allow",
      "Action": "ec2:CreateCapacityReservationBySplitting",
      "Resource": "arn:aws:ec2:region:account:capacity-reservation/
cr-1234567890abcdef0",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/Owner": "ExampleDepartment1"
        }
      }
    },
    {
      "Sid": "AllowNewlyCreatedCapacityReservation",
      "Effect": "Allow",
      "Action": ["ec2:CreateCapacityReservationBySplitting", "ec2:CreateTags"],
      "Resource": "arn:aws:ec2:region:account:capacity-reservation/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/stack": "production"
        }
      }
    }
  ]
}
```

용량 분할

기존 용량 예약에서 용량을 분할하여 새 용량 예약을 생성할 수 있습니다.

Console

용량을 분할하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 용량 예약을 선택합니다.
3. 분할할 용량이 있는 온디맨드 용량 예약 ID를 선택합니다.
4. 작업, 용량 관리에서 분할을 선택합니다.
5. 용량 예약 분할 페이지의 분할할 양에서 슬라이더를 사용하거나 현재 예약에서 분할할 인스턴스 수를 입력합니다.
6. (선택 사항) 새 용량 예약에 대한 태그를 추가합니다.
7. 요약을 검토하고 준비가 되면 분할을 선택합니다.

AWS CLI

용량을 분할하려면

`create-capacity-reservation-by-splitting` 명령을 사용합니다. 다음 예제는 지정된 용량 예약에서 10개의 인스턴스를 분할하여 새 용량 예약을 생성합니다.

```
aws ec2 create-capacity-reservation-by-splitting \
    --source-capacity-reservation-id cr-1234567890abdef0 \
    --instance-count 10
```

PowerShell

용량을 분할하려면

New-EC2CapacityReservationBySplitting cmdlet을 사용합니다. 다음 예제는 지정된 용량 예약에서 10개의 인스턴스를 분할하여 새 용량 예약을 생성합니다.

```
New-EC2CapacityReservationBySplitting `
    -SourceCapacityReservationId cr-1234567890abdef0 `
    -InstanceCount 10
```

용량 예약 취소

다음 상태 중 하나인 용량 예약은 취소할 수 있습니다.

- `assessing`
- `active`이고 약정 기간이 없거나 약정 기간이 경과한 용량 예약. 약정 기간 중에는 장래 날짜 용량 예약을 취소할 수 없습니다.

> **Note**
>
> 용량 블록은 수정하거나 취소할 수 없습니다. 자세한 내용은 ML용 용량 블록 섹션을 참조하세요.

장래 날짜 용량 예약이 `delayed` 상태가 되면 약정 기간이 면제되며, `active` 상태가 되는 즉시 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 리전 예약 인스턴스가 있는 경우)로 용량 예약 외부에서 계속 정상적으로 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 섹션을 참조하세요.

Console

용량 예약을 취소하려면

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 CLI

용량 예약을 취소하려면

cancel-capacity-reservation 명령을 사용합니다.

```
aws ec2 cancel-capacity-reservation \
    --capacity-reservation-id cr-1234567890abcdef0
```

PowerShell

용량 예약을 취소하려면

Remove-EC2CapacityReservation cmdlet을 사용합니다.

```
Remove-EC2CapacityReservation `
    -CapacityReservationId cr-1234567890abcdef0
```

용량 예약 그룹

AWS Resource Groups를 사용하여 리소스 그룹이라고 하는 용량 예약의 논리적 모음을 생성할 수 있습니다. 리소스 그룹은 모두 동일한 AWS 리전에 있는 AWS 리소스의 논리적 그룹입니다. Resource Groups에 대한 자세한 내용은 AWS Resource Groups 사용 설명서의 리소스 그룹이란 무엇입니까?를 참조하세요.

내 계정에서 내가 소유하고 있는 용량 예약, 그리고 다른 AWS 계정에서 나와 공유한 용량 예약을 단일 리소스 그룹에 포함할 수 있습니다. 또한 서로 다른 속성(인스턴스 유형, 플랫폼, 가용 영역, 테넌시)이 있는 용량 예약을 단일 리소스 그룹에 포함할 수도 있습니다.

용량 예약에 대한 리소스 그룹을 생성할 때 인스턴스를 개별 용량 예약 대신 용량 예약의 그룹으로 지정할 수 있습니다. 용량 예약 그룹을 대상으로 하는 인스턴스는 일치하는 속성(인스턴스 유형, 플랫폼, 가용 영역, 테넌시)과 사용 가능한 용량이 있는 그룹의 모든 용량 예약과 일치합니다. 그룹에 일치하는 속성 및 가용 용량이 있는 용량 예약이 없는 경우 인스턴스는 온디맨드 용량을 사용하여 실행됩니다. 이후 단계에서 대상 지정 그룹에 일치하는 용량 예약이 추가되면 인스턴스가 예약 용량과 자동으로 일치되고 해당 예약 용량으로 이동합니다.

그룹에서 용량 예약의 의도하지 않은 사용을 방지하려면 명시적으로 용량 예약을 지정하는 인스턴스만 허용하도록 그룹에서 용량 예약를 구성합니다. 이렇게 하려면 Amazon EC2 콘솔을 사용하여 용량 예약을 생성할 때 인스턴스 자격을 이 예약을 지정하는 인스턴스만으로 설정합니다. AWS CLI를 사용하는 경우 용량 예약을 생성할 때 `--instance-match-criteria targeted`를 지정합니다. 이렇게 하면 그룹 또는 그룹의 용량 예약을 명시적으로 지정하는 인스턴스만 그룹에서 실행할 수 있습니다.

실행 중인 인스턴스가 있을 때 그룹의 용량 예약이 취소되거나 만료되는 경우 인스턴스는 일치하는 속성 및 가용 용량이 있는 그룹에서 다른 용량 예약 인스턴스로 자동으로 이동됩니다. 일치하는 속성 및

가용 용량이 있는 그룹에 남은 용량 예약이 없는 경우 인스턴스는 온디맨드 용량으로 실행됩니다. 이후 단계에서 대상 지정 그룹에 일치하는 용량 예약이 추가되면 인스턴스가 예약 용량으로 자동으로 이동합니다.

업무

- 용량 예약 그룹 생성
- 그룹에 용량 예약 추가
- 그룹에서 용량 예약 제거
- 용량 예약 그룹 삭제

용량 예약 그룹 생성

다음 예제를 사용하여 다음과 같은 요청 파라미터로 용량 예약에 대한 리소스 그룹을 생성할 수 있습니다.

- `AWS::EC2::CapacityReservationPool` - 리소스 그룹이 인스턴스 시작 대상으로 지정될 수 있도록 합니다.
- `allowed-resource-types`가 `AWS::EC2::CapacityReservation`으로 설정된 `AWS::ResourceGroups::Generic` - 리소스 그룹이 용량 예약만 허용하도록 합니다.

그룹을 생성한 후 그룹에 용량 예약을 추가할 수 있습니다.

AWS CLI

용량 예약의 그룹을 생성하려면

create-group AWS CLI 명령을 사용합니다.

```
aws resource-groups create-group \
    --name MyCRGroup \
    --configuration \
        '{"Type": "AWS::EC2::CapacityReservationPool"}' \
        '{"Type": "AWS::ResourceGroups::Generic", "Parameters": [{"Name": "allowed-
resource-types", "Values": ["AWS::EC2::CapacityReservation"]}]}'
```

PowerShell

용량 예약의 그룹을 생성하려면

New-RGGroup cmdlet을 사용합니다.

```
New-RGGroup `
    -Name MyCRGroup `
    -Configuration `
        @{"Type"="AWS::EC2::CapacityReserationPool"} `
        @{"Type"="AWS::ResourceGroups::Generic"; "Parameters"=@{"allowed-resource-
types"=@{"Values"="AWS::EC2::CapacityReservations"}}}
```

그룹에 용량 예약 추가

공유된 용량 예약을 그룹에 추가한 후 해당 용량 예약이 공유되지 않을 경우, 해당 용량 예약이 그룹에서 자동으로 제거됩니다.

AWS CLI

그룹에 용량 예약을 추가하려면

group-resources 명령을 사용합니다.

다음 예제는 2개의 용량 예약을 지정된 그룹에 추가합니다.

```
aws resource-groups group-resources \
    --group MyCRGroup \
    --resource-arns \
        arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-1234567890abcdef1
 \
        arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-54321abcdef567890
```

PowerShell

그룹에 용량 예약을 추가하려면

Add-RGResource cmdlet을 사용합니다.

다음 예제는 2개의 용량 예약을 지정된 그룹에 추가합니다.

```
Add-RGResource `
    -Group MyCRGroup `
    -ResourceArn `
        "arn:aws:ec2:sa-east-1:123456789012:capacity-
reservation/cr-1234567890abcdef1", `
```

```
        "arn:aws:ec2:sa-east-1:123456789012:capacity-
reservation/cr-54321abcdef567890"
```

그룹에서 용량 예약 제거

다음 예제를 사용하여 그룹에서 용량 예약을 제거할 수 있습니다.

AWS CLI

그룹에서 용량 예약을 제거하려면

ungroup-resources 명령을 사용합니다.

다음 예제는 지정된 그룹에서 2개의 용량 예약을 제거합니다.

```
aws resource-groups ungroup-resources \
    --group MyCRGroup \
    --resource-arns \
        arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-0e154d26a16094dd
 \
        arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-54321abcdef567890
```

PowerShell

그룹에서 용량 예약을 제거하려면

Remove-RGResource cmdlet을 사용합니다.

다음 예제는 지정된 그룹에서 2개의 용량 예약을 제거합니다.

```
Remove-RGResource `
    -Group MyCRGroup `
    -ResourceArn `
        "arn:aws:ec2:sa-east-1:123456789012:capacity-
reservation/cr-0e154d26a16094dd", `
        "arn:aws:ec2:sa-east-1:123456789012:capacity-
reservation/cr-54321abcdef567890"
```

용량 예약 그룹 삭제

다음 예제를 사용하여 용량 예약 그룹을 삭제할 수 있습니다.

AWS CLI

그룹을 삭제하려면

delete-group 명령을 사용합니다.

```
aws resource-groups delete-group --group MyCRGroup
```

PowerShell

그룹 삭제

Remove-RGGroup cmdlet을 사용합니다.

```
Remove-RGGroup -GroupName MyCRGroup
```

클러스터 배치 그룹에 용량 예약 생성

클러스터 배치 그룹에서 용량 예약을 생성하여 워크로드에 대한 Amazon EC2 컴퓨팅 용량을 예약할 수 있습니다. 클러스터 배치 그룹은 짧은 네트워크 대기 시간과 높은 네트워크 처리량이라는 이점을 제공합니다.

클러스터 배치 그룹에서 용량 예약을 생성하면 필요할 때 필요한 만큼 클러스터 배치 그룹의 컴퓨팅 용량에 액세스할 수 있습니다. 이는 컴퓨팅 크기 조정이 필요한 고성능(HPC) 워크로드를 위한 용량을 예약하는 데 이상적입니다. 이를 통해 필요할 때 다시 확장할 수 있도록 용량을 사용할 수 있는 상태로 유지하면서 클러스터를 축소할 수 있습니다.

주제

- 제한 사항
- 클러스터 배치 그룹의 용량 예약 작업

제한 사항

클러스터 배치 그룹에서 용량 예약을 생성할 때 다음 사항에 유의하세요.

- 기존의 용량 예약이 배치 그룹에 없으면 용량 예약을 수정하여 배치 그룹의 용량을 예약할 수 없습니다. 배치 그룹에서 용량을 예약하려면 배치 그룹에서 용량 예약을 생성해야 합니다.
- 배치 그룹에서 용량 예약을 생성한 후에는 배치 그룹 외부에서 용량을 예약하도록 수정할 수 없습니다.

- 배치 그룹에서 기존 용량 예약을 수정하거나 배치 그룹에서 추가 용량 예약을 생성하여 배치 그룹에서 예약 용량을 늘릴 수 있습니다. 그러나 용량 부족 오류가 발생할 가능성이 높아집니다.
- 클러스터 배치 그룹에서 생성된 용량 예약은 공유할 수 없습니다.
- `active` 용량 예약이 있는 클러스터 배치 그룹은 삭제할 수 없습니다. 클러스터 배치 그룹의 모든 용량 예약을 취소해야 클러스터 배치 그룹을 삭제할 수 있습니다.

클러스터 배치 그룹의 용량 예약 작업

클러스터 배치 그룹과 함께 용량 예약 사용을 시작하려면 다음 단계를 수행하세요.

> **Note**
>
> 기존 클러스터 배치 그룹에서 용량 예약을 생성하려면 1단계를 건너뜁니다. 그런 다음 2단계와 3단계에서 기존 클러스터 배치 그룹의 ARN을 지정합니다.

업무

- 1단계: (조건부) 용량 예약에 사용할 클러스터 배치 그룹 생성
- 2단계: 클러스터 배치 그룹에서 용량 예약 생성
- 3단계: 클러스터 배치 그룹으로 인스턴스 시작

1단계: (조건부) 용량 예약에 사용할 클러스터 배치 그룹 생성

새 클러스터 배치 그룹을 생성해야 하는 경우에만 이 단계를 수행합니다. 기존 클러스터 배치 그룹을 사용하려면 이 단계를 건너뛰고 2단계와 3단계에서 해당 클러스터 배치 그룹의 ARN을 사용합니다.

Console

클러스터 배치 그룹을 생성하려면 다음을 수행하세요.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 배치 그룹(Placement Groups)과 배치 그룹 생성(Create placement group)을 차례로 선택합니다.
3. 이름(Name)에 배치 그룹을 설명하는 이름을 지정합니다.
4. 배치 전략(Placement strategy)에서 클러스터(Cluster)를 선택합니다.
5. 그룹 생성을 선택합니다.

6. 배치 그룹 표의 그룹 ARN 열에 생성한 클러스터 배치 그룹의 ARN을 기록해 둡니다. 이 정보는 다음 단계에 필요합니다.

AWS CLI

클러스터 배치 그룹을 생성하려면 다음을 수행하세요.

create-placement-group 명령을 사용합니다.

```
aws ec2 create-placement-group \
    --group-name MyPG \
    --strategy cluster
```

다음 단계에 필요하므로 출력에 반환된 배치 그룹 ARN을 기록해 둡니다.

PowerShell

클러스터 배치 그룹을 생성하려면 다음을 수행하세요.

New-EC2PlacementGroup cmdlet을 사용합니다.

```
New-EC2PlacementGroup `
    -GroupName my-placement-group `
    -Strategy "cluster"
```

다음 단계에 필요하므로 출력에 반환된 배치 그룹 ARN을 기록해 둡니다.

2단계: 클러스터 배치 그룹에서 용량 예약 생성

용량 예약을 생성하는 것과 동일한 방식으로 클러스터 배치 그룹에서 용량 예약을 생성합니다. 그러나 용량 예약을 생성할 클러스터 배치 그룹의 ARN도 지정해야 합니다. 자세한 내용은 용량 예약 생성 섹션을 참조하세요.

고려 사항

- 지정된 클러스터 배치 그룹은 `available` 상태여야 합니다. 클러스터 배치 그룹이 `pending`, `deleting` 또는 `deleted` 상태인 경우 요청이 실패합니다.
- 용량 예약과 클러스터 배치 그룹은 동일한 가용 영역에 있어야 합니다. 용량 예약 생성 요청이 클러스터 배치 그룹의 가용 영역과 다른 가용 영역을 지정하면 요청이 실패합니다.
- 클러스터 배치 그룹에서 지원하는 인스턴스 유형에 대해서만 용량 예약을 생성할 수 있습니다. 지원되지 않는 인스턴스를 지정하면 요청이 실패합니다.

- 클러스터 배치 그룹에서 open 용량 예약을 생성하고 일치하는 속성(배치 그룹 ARN, 인스턴스 유형, 가용 영역, 플랫폼 및 테넌시)이 있는 실행 중인 기존 인스턴스가 있는 경우 해당 인스턴스는 용량 예약에서 자동으로 실행됩니다.
- 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 시도하세요. 애플리케이션이 인스턴스 유형 및 크기 면에서 유연한 경우 다른 인스턴스 속성으로 사용해 봅니다.
  - 요청한 수량이 선택한 인스턴스 패밀리에 대한 온디맨드 인스턴스 제한을 초과합니다. 인스턴스 패밀리에 대한 온디맨드 인스턴스 제한을 늘리고 다시 시도하세요. 자세한 내용은 온디맨드 인스턴스 할당량 섹션을 참조하세요.

Console

용량 예약을 생성하려면

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 페이지에서 필요에 따라 인스턴스 유형, 플랫폼, 가용 영역, 테넌시, 수량 및 종료 날짜를 지정합니다.
4. 배치 그룹에서 용량 예약을 생성할 클러스터 배치 그룹의 ARN을 선택합니다.
5. 생성(Create)을 선택합니다.

자세한 내용은 용량 예약 생성 섹션을 참조하세요.

AWS CLI

용량 예약을 생성하려면

create-capacity-reservation 명령을 사용합니다. `--placement-group-arn`에 대해 용량 예약을 생성할 클러스터 배치 그룹의 ARN을 지정합니다.

```
aws ec2 create-capacity-reservation \
    --instance-type instance_type \
    --instance-platform platform \
    --availability-zone-id az_id \
    --instance-count quantity \
    --placement-group-arn "placement_group_arn"
```

PowerShell

용량 예약을 생성하려면

Add-EC2CapacityReservation cmdlet을 사용합니다. -PlacementGroupArn에 대해 용량 예약을 생성할 클러스터 배치 그룹의 ARN을 지정합니다.

```
Add-EC2CapacityReservation `
    -InstanceType instance_type `
    -InstancePlatform platform `
    -AvailabilityZoneId az_id `
    -InstanceCount quantity `
    -PlacementGroupArn "placement_group_arn"
```

3단계: 클러스터 배치 그룹으로 인스턴스 시작

인스턴스를 용량 예약으로 시작하는 것과 동일한 방식으로 클러스터 배치 그룹의 용량 예약으로 인스턴스를 시작합니다. 그러나 인스턴스를 시작할 클러스터 배치 그룹의 ARN도 지정해야 합니다. 자세한 내용은 용량 예약 생성 섹션을 참조하세요.

고려 사항

- 용량 예약이 open이면 인스턴스 시작 요청에 용량 예약을 지정할 필요가 없습니다. 인스턴스에 지정한 배치 그룹의 용량 예약과 일치하는 속성(배치 그룹 ARN, 인스턴스 유형, 가용 영역, 플랫폼 및 테넌시)이 있는 경우 인스턴스가 용량 예약에서 자동으로 실행됩니다.
- 용량 예약이 대상 인스턴스 시작만 허용하는 경우 요청에 클러스터 배치 그룹과 함께 목표 용량 예약을 지정해야 합니다.
- 용량 예약이 용량 예약 그룹에 있는 경우 요청에 클러스터 배치 그룹과 함께 목표 용량 예약 그룹을 지정해야 합니다. 자세한 내용은 용량 예약 그룹 섹션을 참조하세요.

자세한 내용은 인스턴스를 기존 용량 예약으로 시작 섹션을 참조하세요.

Console

인스턴스를 기존 용량 예약으로 시작하려면

1. 절차에 따라 인스턴스를 시작하되 다음 단계를 완료하여 배치 그룹 및 용량 예약 설정을 지정할 때까지 인스턴스를 시작하지 마세요.
2. 고급 세부 정보를 열고 다음을 수행합니다.

a. 배치 그룹의 경우 인스턴스를 시작할 클러스터 배치 그룹을 선택합니다.

b. 용량 예약(Capacity Reservation)에서 용량 예약 구성에 따라 다음 옵션 중 하나를 선택합니다.

- 열림 - 일치하는 속성과 충분한 용량이 있는 클러스터 배치 그룹의 open 용량 예약으로 인스턴스를 시작합니다.
- ID별 대상 - 대상 인스턴스 시작만 허용하는 용량 예약으로 인스턴스를 시작합니다.
- 그룹별 대상 지정 - 선택한 용량 예약 그룹에서 일치하는 속성 및 가용 용량이 있는 용량 예약으로 인스턴스를 시작합니다.

3. Summary(요약) 패널에서 인스턴스 구성을 검토한 다음 Launch instance(인스턴스 시작)를 선택합니다. 자세한 내용은 콘솔에서 인스턴스 시작 마법사를 사용하여 EC2 인스턴스 시작 섹션을 참조하세요.

AWS CLI

인스턴스를 기존 용량 예약으로 시작하려면

run-instances 명령을 사용합니다. 특정 용량 예약 또는 용량 예약 그룹을 대상으로 지정해야 하는 경우 `--capacity-reservation-specification` 파라미터를 지정합니다. `--placement`에 대해 `GroupName` 파라미터를 지정한 다음 이전 단계에서 생성한 배치 그룹의 이름을 지정합니다.

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --count quantity \
    --instance-type instance_type \
    --key-name key_pair_name \
    --subnet-id subnet-0abcdef1234567890 \
    --capacity-reservation-specification
 CapacityReservationTarget={CapacityReservationId=capacity_reservation_id} \
    --placement "GroupName=cluster_placement_group_name"
```

PowerShell

인스턴스를 기존 용량 예약으로 시작하려면

New-EC2Instance cmdlet을 사용합니다. `-Placement`에 대해 `GroupName` 파라미터를 지정한 다음 이전 단계에서 생성한 배치 그룹의 이름을 지정합니다.

```
New-EC2Instance `
```

```
    -ImageId ami-0abcdef1234567890 `
    -InstanceType instance_type `
    -KeyName key_pair_name `
    -SubnetId subnet-0abcdef1234567890 `
    -CapacityReservationTarget_CapacityReservationId capacity_reservation_id `
    -Placement_GroupName cluster_placement_group_name
```

Local Zones의 용량 예약

Local Zones는 사용자와 지리적으로 가까운 AWS 리전의 확장입니다. Local Zones에서 생성된 리소스는 지연 시간이 매우 짧은 통신으로 로컬 사용자에게 제공될 수 있습니다. 자세한 내용은 AWS Local Zones를 참조하세요.

로컬 영역에 새 서브넷을 생성하여 VPC를 상위 AWS 리전에서 로컬 영역으로 확장할 수 있습니다. 로컬 영역에 서브넷을 생성하면 VPC도 해당 로컬 영역으로 확장됩니다. 로컬 영역의 서브넷은 VPC의 다른 서브넷과 동일하게 작동합니다.

Local Zones를 사용하면 사용자와 가까운 여러 위치에 용량 예약을 배치할 수 있습니다. 일반 가용 영역에서 용량 예약을 생성하고 사용하는 것과 같은 방식으로 Local Zones에서 용량 예약을 생성하고 사용합니다. 동일한 기능 및 인스턴스 일치 동작이 적용됩니다. Local Zones에서 지원되는 요금 모델에 대한 자세한 내용은 AWS Local Zones FAQ를 참조하세요.

고려 사항

로컬 영역에서는 용량 예약 그룹을 사용할 수 없습니다.

로컬 영역에서 용량 예약을 사용하려면

1. AWS 계정에서 사용할 로컬 영역을 사용하도록 설정합니다. 자세한 내용은 AWS Local Zones User Guide의 Getting started with AWS Local Zones를 참조하세요.
2. 로컬 영역에 용량 예약을 생성합니다. [가용 영역(Availability Zone)]에서 로컬 영역을 선택합니다. 로컬 영역은 AWS 리전 코드 뒤에 위치를 나타내는 식별자를 붙여 표시됩니다(예: `us-west-2-lax-1a`). 자세한 내용은 용량 예약 생성 섹션을 참조하세요.
3. 로컬 영역에서 서브넷을 만듭니다. [가용 영역(Availability Zone)]에서 로컬 영역을 선택합니다. 자세한 내용은 Amazon VPC 사용 설명서의 VPC에서 서브넷 만들기를 참조하세요.
4. 인스턴스 시작. [서브넷(Subnet)]에서 로컬 영역의 서브넷(예: `subnet-123abc | us-west-2-lax-1a`)을 선택하고, 용량 예약에서 로컬 영역에서 생성한 용량 예약에 필요한 사양(open 또는 ID로 지정)을 선택합니다. 자세한 내용은 인스턴스를 기존 용량 예약으로 시작 섹션을 참조하세요.

Wavelength 영역의 용량 예약

AWS Wavelength을(를) 사용하면 개발자는 모바일 디바이스 및 최종 사용자에게 매우 짧은 지연 시간을 제공하는 애플리케이션을 빌드할 수 있습니다. Wavelength는 표준 AWS 컴퓨팅 및 스토리지 서비스를 통신 사업자의 5G 네트워크 엣지에 배포합니다. Amazon Virtual Private Cloud(VPC)를 하나 이상의 Wavelength Zone으로 확장할 수 있습니다. 그런 다음 Amazon EC2 인스턴스와 같은 AWS 리소스를 사용하여 리전의 AWS 서비스에 대한 극도로 짧은 대기 시간 및 연결을 필요로 하는 애플리케이션을 실행할 수 있습니다. 자세한 내용은 AWS Wavelength Zone을 참조하세요.

온디맨드 용량 예약을 생성할 때 Wavelength Zone을 선택하고 Wavelength Zone에 연결된 서브넷을 지정하여 Wavelength Zone의 용량 예약으로 인스턴스를 시작할 수 있습니다. Wavelength 영역은 AWS 리전 코드 뒤에 위치를 나타내는 식별자를 붙여 표시됩니다(예: `us-east-1-wl1-bos-wlz-1`).

Wavelength Zone은 일부 리전에서 사용할 수 없습니다. Wavelength Zone을 지원하는 리전에 대한 자세한 내용은 AWS Wavelength 개발자 안내서의 사용 가능한 Wavelength Zone을 참조하세요.

고려 사항

Wavelength Zone에서는 용량 예약 그룹을 사용할 수 없습니다.

Wavelength Zone에서 용량 예약을 사용하려면

1. AWS 계정에서 사용할 Wavelength 영역을 사용하도록 설정합니다. 자세한 정보는 AWS Wavelength개발자 가이드에서 AWS Wavelength 시작하기를 참조하세요.
2. Wavelength Zone에 용량 예약을 생성합니다. [가용 영역(Availability Zone)]에서 Wavelength를 선택합니다. Wavelength는 AWS 리전 코드 뒤에 위치를 나타내는 식별자를 붙여 표시됩니다(예: `us-east-1-wl1-bos-wlz-1`). 자세한 내용은 용량 예약 생성 섹션을 참조하세요.
3. Wavelength Zone에서 서브넷을 생성합니다. [가용 영역(Availability Zone)]에서 Wavelength Zone을 선택합니다. 자세한 내용은 Amazon VPC 사용 설명서의 VPC에서 서브넷 만들기를 참조하세요.
4. 인스턴스 시작. [서브넷(Subnet)]에서 Wavelength Zone의 서브넷(예: `subnet-123abc | us-east-1-wl1-bos-wlz-1`)을 선택하고, 용량 예약에서 Wavelength에서 생성한 용량 예약에 필요한 사양(open 또는 ID로 지정)을 선택합니다. 자세한 내용은 인스턴스를 기존 용량 예약으로 시작 섹션을 참조하세요.

AWS Outposts의 용량 예약

AWS Outposts은(는) AWS 인프라, 서비스, API 및 도구를 고객 온프레미스로 확장하는 완전관리형 서비스입니다. AWS 관리형 인프라에 대한 로컬 액세스를 제공하는 AWS Outposts을(를) 통해 고객은

AWS 리전에서 사용하는 것과 동일한 프로그래밍 인터페이스를 사용해 온프레미스에서 애플리케이션을 구축하고 실행할 수 있으며, 짧은 지연 시간과 로컬 데이터 처리가 필요한 경우에 로컬 컴퓨팅 및 스토리지 리소스를 사용할 수 있습니다.

Outposts는 고객 사이트에 배포된 AWS 컴퓨팅 및 스토리지 용량 풀입니다. AWS는 이 용량을 AWS 리전의 일부로 운영, 모니터링 및 관리합니다.

계정에 생성한 Outposts에서 용량 예약을 생성할 수 있습니다. 이렇게 하면 사이트의 Outposts에서 컴퓨팅 용량을 예약할 수 있습니다. 일반 가용 영역에서 용량 예약을 생성하고 사용하는 것과 같은 방식으로 Outposts에서 용량 예약을 생성하고 사용합니다. 동일한 기능 및 인스턴스 일치 동작이 적용됩니다.

또한 AWS Resource Access Manager를 사용하여 Outposts의 용량 예약을 조직 내의 다른 AWS 계정과 공유할 수 있습니다. 용량 예약 공유에 대한 자세한 내용은 공유 용량 예약 섹션을 참조하세요.

전제 조건

사이트에 Outpost가 설치되어 있어야 합니다. 자세한 내용은 AWS Outposts 사용 설명서에서 Outposts 생성 및 Outposts 용량 주문을 참조하세요.

고려 사항

- Outposts에서는 용량 예약 그룹을 사용할 수 없습니다.

Outposts에서 용량 예약을 사용하려면

1. Outposts에서 서브넷을 생성합니다. 자세한 내용은 AWS Outposts 사용 설명서에서 서브넷 생성을 참조하세요.
2. Outposts에서 용량 예약을 생성합니다.
   a. AWS Outposts 콘솔(https://console.aws.amazon.com/outposts/)을 엽니다.
   b. 탐색 창에서 [Outposts]를 선택한 다음 [작업(Actions)], [용량 예약 생성(Create Capacity Reservation)]을 차례로 선택합니다.
   c. 필요에 따라 용량 예약을 구성한 다음 [생성(Create)]을 선택합니다. 자세한 내용은 용량 예약 생성 섹션을 참조하세요.

> **Note**
>
> 인스턴스 유형(Instance Type) 드롭다운 목록에는 선택한 Outposts에서 지원하는 인스턴스 유형만 나열되고, 가용 영역(Availability Zone) 드롭다운에는 선택한 Outposts가 연결된 가용 영역만 나열됩니다.

3. 인스턴스를 용량 예약으로 시작 서브넷(Subnet)에서 1단계에서 생성한 서브넷을 선택하고, 용량 예약(Capacity Reservation)에서 2단계에서 생성한 용량 예약을 선택합니다. 자세한 내용은 AWS Outposts 사용 설명서에서 Outposts에서 인스턴스 시작을 참조하세요.

공유 용량 예약

용량 예약 공유를 통해 용량 예약 소유자는 예약 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

용량 예약 공유를 위한 사전 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.
- 새 AWS 계정 또는 결제 기록이 제한적인 AWS 계정에는 용량 예약 공유가 제공되지 않습니다.

- AWS 조직 또는 AWS 조직의 조직 단위와 용량 예약을 공유하려면 AWS Organizations를 통해 공유를 사용하도록 설정해야 합니다. 자세한 내용은 AWS RAM 사용 설명서에서 AWS Organizations를 사용하여 공유 사용을 참조하세요.

관련 서비스

용량 예약 공유는 AWS Resource Access Manager(AWS RAM)와 통합됩니다. AWS RAM은 모든 AWS 계정 또는 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 참조하세요.

여러 가용 영역에서 공유

리전의 가용 영역에 걸쳐 리소스가 배포될 수 있도록 각 계정의 이름에 가용 영역을 독립적으로 매핑합니다. 이로 인해 계정 전체에서 가용 영역 이름의 차이가 발생할 수 있습니다. 예를 들어 AWS 계정의 `us-east-1a` 가용 영역은 다른 AWS 계정에 대한 `us-east-1a`로 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 보려면

1. https://console.aws.amazon.com/ram/homeAWS RAM에서 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 명심하세요.

- 소비자가 용량 예약의 속성과 일치하는 인스턴스를 실행하고 `CapacityReservationPreference` 파라미터를 `open`으로 설정하고 예약된 용량에서 아직 실행하지 않는 경우, 공유된 용량 예약을 자동으로 사용합니다.

- 소비자가 일치하는 속성(인스턴스 유형, 플랫폼, 가용 영역, 테넌시)이 있고 `CapacityReservationPreference` 파라미터가 `open`으로 설정된 인스턴스를 시작하면 자동으로 공유 용량 예약으로 시작됩니다.

용량 예약을 공유하려면 리소스 공유에 추가해야 합니다. 리소스 공유는 여러 AWS 계정에서 리소스를 공유할 수 있게 해주는 AWS RAM 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 생성해야 합니다.

AWS Organizations의 조직에 속해 있고 조직 내에서 공유를 사용하는 경우, 공유 사전 조건을 충족한다면 공유 용량 예약에 대한 액세스 권한이 조직의 소비자에게 부여됩니다. 용량 예약을 외부 계정과 공유하는 경우, 소비자는 리소스 공유에 가입하라는 초대장을 받고 초대를 수락한 후 공유 용량 예약에 대한 액세스 권한을 얻을 수 있습니다.

> **Important**
>
> 사용자와 공유되는 용량 예약으로 인스턴스를 시작하기 전에, 콘솔에서 확인하거나 describe-capacity-reservations AWS CLI 명령을 사용하여 공유된 용량 예약에 액세스할 수 있는지 확인하세요. 콘솔에서 공유 용량 예약을 확인하거나 AWS CLI를 사용하여 설명할 수 있는 경우에 사용할 수 있으며, 인스턴스를 시작할 수 있습니다. 용량 예약으로 인스턴스를 시작하려고 할 때 공유 실패로 인해 액세스할 수 없는 경우, 인스턴스는 온디맨드 용량으로 시작됩니다.

Console

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 생성을 참조하세요.

AWS CLI

소유하고 있는 용량 예약을 공유하는 방법

create-resource-share 명령을 사용합니다.

```
aws ram create-resource-share \
    --name my-resource-share \
    --resource-arns arn:aws:ec2:us-east-2:123456789012:capacity-
reservation/cr-1234abcd56EXAMPLE
```

PowerShell

소유하고 있는 용량 예약을 공유하는 방법

New-RAMResourceShare cmdlet을 사용합니다.

```
New-RAMResourceShare `
    -Name my-resource-share `
    -ResourceArn "arn:aws:ec2:us-east-2:123456789012:capacity-
reservation/cr-1234abcd56EXAMPLE"
```

용량 예약 공유 중지

용량 예약 소유자는 언제든지 용량 예약의 공유를 중지할 수 있습니다. 다음 규칙이 적용됩니다.

- 공유가 중지될 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 용량 예약의 공유를 중지하려면 리소스 공유에서 제거해야 합니다.

Console

Amazon EC2 콘솔을 사용하여 소유하고 있는 용량 예약의 공유를 중지하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 용량 예약를 선택합니다.
3. 용량 예약을 선택하고 공유(Sharing) 탭을 선택합니다.

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 용량 예약의 공유를 중지하려면

AWS RAM 사용 설명서에서 리소스 공유 업데이트를 참조하세요.

AWS CLI

소유하고 있는 용량 예약 공유를 중지하는 방법

disassociate-resource-share 명령을 사용합니다.

```
aws ram disassociate-resource-share \
    --resource-share-arn arn:aws:ram:us-east-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE \
 --resource-arns arn:aws:ec2:us-east-2:123456789012:capacity-reservation/cr-1234abcd56EXAMPLE
```

PowerShell

소유하고 있는 용량 예약 공유를 중지하는 방법

Disconnect-RAMResourceShare cmdlet을 사용합니다.

```
Disconnect-RAMResourceShare `
    -ResourceShareArn "arn:aws:ram:us-east-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE" `
    -ResourceArn "arn:aws:ec2:us-east-2:123456789012:capacity-reservation/cr-1234abcd56EXAMPLE"
```

공유 Amazon EC2 용량 예약에 대한 결제 할당

기본적으로 용량 예약이 공유되면 소유자에게는 용량 예약에서 실행 중인 인스턴스와 용량 예약에서 사용 가능한 용량(미사용 용량이라고도 함)에 대해 요금이 결제되고, 소비자는 공유 용량 예약에서 실행 중인 인스턴스에 대해서만 요금이 결제됩니다.

필요한 경우 용량 예약 소유자는 용량 예약에서 사용 가능한 모든 용량의 결제를 용량 예약이 공유되는 계정 중 하나에 할당할 수 있습니다. 결제가 다른 계정에 할당되면 해당 계정은 용량 예약에서 사용 가능한 용량의 결제 소유자가 됩니다. 그 시점부터 용량 예약에서 사용 가능한 용량에 대한 요금은 소유

자의 계정이 아닌 할당된 계정으로 청구됩니다. 용량 예약 소유자와 용량 예약이 공유된 계정에는 용량 예약에서 실행 중인 인스턴스에 대한 요금이 계속 결제됩니다.

> **Important**
>
> 용량 예약 소유자는 여전히 리소스 소유자이며 용량 예약을 관리할 책임이 있습니다. 결제가 할당된 계정에는 추가 권한이 부여되지 않으며 어떤 방식으로도 용량 예약을 취소, 수정 또는 공유할 수 없습니다.

주제

- 작동 방법
- 고려 사항
- 공유 EC2 용량 예약의 결제를 다른 계정에 할당
- 공유 EC2 용량 예약에 대한 결제 할당 요청 보기
- 공유 EC2 용량 예약의 결제 수락 또는 거부
- 공유 EC2 용량 예약에 대한 결제 할당 요청 취소 또는 철회
- 공유 용량 예약에 대한 결제 할당 요청 모니터링

작동 방법

용량 예약 소유자만 공유 용량 예약의 청구를 다른 계정에 할당할 수 있습니다. 결제는 용량 예약이 공유되고 용량 예약 소유자와 동일한 AWS Organizations 결제자 계정으로 통합된 계정에만 할당할 수 있습니다.

용량 예약의 사용 가능한 용량을 다른 계정에 청구하려면 용량 예약 소유자가 필요한 계정으로 요청을 시작해야 합니다. 지정된 계정에서 요청을 수신하며 12시간 이내에 요청을 수락하거나 거부해야 합니다.

- 수락하면 용량 예약에서 사용 가능한 용량(미사용 용량이라고도 함)에 대한 결제 소유자가 됩니다. 그 시점부터는 용량 예약에서 사용 가능한 용량에 대한 요금이 소유자의 계정이 아닌 해당 계정으로 청구됩니다. 수락한 후에는 용량 예약 소유자만 할당된 계정에서 청구를 취소할 수 있습니다.
- 거부하는 경우 용량 예약 소유자는 용량 예약의 사용 가능한 용량에 대한 청구 소유자로 남습니다. 용량 예약에서 사용 가능한 용량에 대한 요금은 소유자의 계정으로 계속 청구됩니다.
- 12시간 이내에 요청을 수락하거나 거부하지 않으면 예약이 만료되고 용량 예약의 사용 가능한 용량에 대한 요금이 소유자의 계정으로 계속 청구됩니다.

청구가 다른 계정에 할당된 기간 동안에는 소유자의 CUR 대신 할당된 계정의 비용 및 사용량 보고서(CUR)에 `Reservation` 및 `UnusedBox` 항목이 표시됩니다.

다음 테이블에는 다른 계정에 청구가 할당되기 전에 용량 예약 소유자 및 소비자 계정에 대한 CUR에 표시되는 항목이 나와 있습니다.

Account	결제가 할당되기 전의 CUR 항목
용량 예약 소유자	• `Reservation` • `BoxUsage *` • `UnusedBox`
용량 예약이 공유되는 소비자 계정	• `BoxUsage *`

다음 테이블에는 다른 계정에 청구가 할당된 후에 용량 예약 소유자 및 소비자 계정에 대한 CUR에 표시되는 항목이 나와 있습니다.

Account	결제가 할당된 후의 CUR 항목
용량 예약 소유자	• `BoxUsage *`
결제가 할당된 소비자 계정	• `Reservation` • `BoxUsage *` • `UnusedBox`

Account	결제가 할당된 후의 CUR 항목
용량 예약이 공유되는 다른 소비자 계정	• BoxUsage *

> **Note**
>
> - * 용량 예약에 실행 중인 인스턴스가 있는 경우에만 계정의 CUR에 BoxUsage 항목이 표시됩니다. CUR 항목에 대한 자세한 내용은 용량 예약 모니터링을 참조하세요.
> - CUR의 용량 예약 ARN을 사용하여 용량 예약을 소유한 사람을 확인합니다. ARN에 AWS 계정 ID가 포함되어 있는 경우 사용자는 용량 예약 소유자입니다. 그렇게 하지 않으면 용량 예약은 다른 계정에서 소유하지만 사용자에게 결제가 할당됩니다.
> - 소유자가 용량 예약에 할당하는 비용 할당 태그는 소비자 계정의 CUR에 표시되지 않습니다. 비용 할당 태그는 용량 예약 소유자의 CUR에만 표시됩니다.

고려 사항

공유 용량 예약의 결제를 할당할 때는 다음 사항에 유의하세요.

- 부분 또는 분할 결제 할당을 할 수 없습니다. 용량 예약의 사용 가능한 모든 용량에 대한 결제는 한 번에 하나의 계정에 할당할 수 있습니다.
- 용량 예약의 사용 가능한 용량은 시간이 지남에 따라 변경될 수 있습니다. 이렇게 하면 할당된 계정의 결제에 영향을 미칩니다. 예를 들어 용량 예약 소유자가 용량 예약의 크기를 늘리거나 다른 소비자 계정이 용량 예약에서 실행 중인 인스턴스를 중지하거나 종료하면 사용 가능한 용량이 증가할 수 있습니다.
- 결제는 동일한 AWS Organizations 결제자 계정으로 통합된 소비자 계정에만 할당할 수 있습니다. 조직에서 퇴사하거나 용량 예약이 더 이상 공유되지 않는 경우 소비자 계정에서 결제가 자동으로 취소됩니다.
- 용량 예약 소유자만 요청이 수락된 후 보류 중인 결제 할당 요청을 취소하고 할당된 계정에서 결제를 취소할 수 있습니다.

공유 EC2 용량 예약의 결제를 다른 계정에 할당

공유 용량 예약의 사용 가능한 용량을 다른 계정에 청구하려면 용량 예약 소유자가 필요한 계정으로 요청을 시작해야 합니다. Amazon EC2 콘솔에서는 이 요청을 이전 요청이라고 합니다.

용량 예약 소유자는 다음과 같은 경우에만 용량 예약의 사용 가능한 용량에 대한 결제를 계정에 할당할 수 있습니다.

- 용량 예약은 이미 해당 계정과 공유되어 있습니다.
- 계정은 용량 예약 소유자와 동일한 AWS Organizations 결제자 계정으로 통합됩니다.

결제는 요청을 수락한 경우에만 지정된 계정에 할당됩니다.

용량 예약 소유자가 요청을 시작하면 요청된 계정으로 Amazon EventBridge 이벤트가 전송됩니다. 자세한 내용은 공유 용량 예약에 대한 결제 할당 요청 모니터링 섹션을 참조하세요.

Console

공유 용량 예약의 결제를 할당하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 패널에서 용량 예약을 선택한 다음 공유 용량 예약을 선택합니다.
3. 사용 가능한 용량 결제 섹션에서 결제 할당을 선택합니다.
4. 결제 할당 화면에서 결제를 할당할 소비자 계정을 선택한 다음 요청을 선택합니다.

AWS CLI

공유 용량 예약의 결제를 할당하려면

associate-capacity-reservation-billing-owner 명령을 사용합니다. `--capacity-reservation-id`의 경우 공유 용량 예약의 ID를 지정합니다. `--unused-reservation-billing-owner-id`의 경우 결제를 할당할 AWS 계정의 ID를 지정합니다.

```
aws ec2 associate-capacity-reservation-billing-owner \
    --capacity-reservation-id cr-01234567890abcdef \
    --unused-reservation-billing-owner-id 123456789012
```

PowerShell

공유 용량 예약의 결제를 할당하려면

Register-EC2CapacityReservationBillingOwner cmdlet을 사용합니다. - CapacityReservationId의 경우 공유 용량 예약의 ID를 지정합니다. - UnusedReservationBillingOwnerId의 경우 결제를 할당할 AWS 계정의 ID를 지정합니다.

```
Register-EC2CapacityReservationBillingOwner `
    -CapacityReservationId cr-01234567890abcdef `
    -UnusedReservationBillingOwnerId 123456789012
```

공유 EC2 용량 예약에 대한 결제 할당 요청 보기

용량 예약 소유자는 자신이 시작한 가장 최근의 결제 할당 요청만 볼 수 있습니다. 또한 소비자 계정에서는 자신에게 전송된 가장 최근의 결제 할당 요청만 볼 수 있습니다.

cancelled, expired 또는 revoked 상태로 전환된 후 24시간 동안 요청을 볼 수 있습니다. 24시간이 경과하면 더 이상 볼 수 없습니다.

Console

(용량 예약 소유자) 사용자가 시작한 요청을 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 패널에서 용량 예약을 선택한 다음 요청을 보려는 공유 용량 예약을 선택합니다.
3. 사용 가능한 용량 결제 섹션에는 최신 요청과 현재 상태가 표시됩니다.

(소비자 계정) 사용자에게 전송된 요청 대상

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 패널에서 용량 예약을 선택합니다.
3. 보류 중인 요청이 있는 경우 화면 상단에 보류 중인 결제 할당 요청 배너가 표시됩니다. 배너가 표시되지 않으면 보류 중인 요청이 없는 것입니다.

   요청을 보려면 배너에서 요청 검토를 선택합니다.

AWS CLI

(용량 예약 소유자) 사용자가 시작한 요청을 보려면

describe-capacity-reservation-billing-requests 명령을 사용합니다.

```
aws ec2 describe-capacity-reservation-billing-requests \
    --role odcr-owner
```

(소비자 계정) 사용자에게 전송된 요청을 보려면

describe-capacity-reservation-billing-requests 명령을 사용합니다.

```
aws ec2 describe-capacity-reservation-billing-requests \
    --role unused-reservation-billing-owner
```

PowerShell

(용량 예약 소유자) 사용자가 시작한 요청을 보려면

Get-EC2CapacityReservationBillingRequest cmdlet을 사용합니다.

```
Get-EC2CapacityReservationBillingRequest `
    -Role odcr-owner
```

(소비자 계정) 사용자에게 전송된 요청을 보려면

Get-EC2CapacityReservationBillingRequest cmdlet을 사용합니다.

```
Get-EC2CapacityReservationBillingRequest `
    -Role unused-reservation-billing-owner
```

요청은 다음 상태 중 하나일 수 있습니다.

State	설명			
`pending`	요청이 수락되거나 거부되지 않았지만 아직 만료되지 않았습니다.			
`accepted`	지정된 계정에서 요청을 수락했습니다. 용량 예약의 사용 가능한 용량에 대한 결제는 소비자 계정에 할당됩니다.			

State	설명
rejected	소비자 계정에서 요청을 거부했습니다.
cancelled	요청이 pending 상태인 동안 용량 예약 소유자에 의해 요청이 취소되었습니다.
revoked	다음 사유 중 하나로 인해 소비자 계정에서 청구가 취소되었습니다. • 용량 예약 소유자가 명시적으로 취소했습니다. • 용량 예약은 더 이상 소비자 계정과 공유되지 않습니다. • 소비자 계정은 더 이상 AWS 조직에 속하지 않습니다.
expired	소비자 계정에서 12시간 이내에 요청을 수락하거나 거부하지 않아 요청이 만료되었습니다.

공유 EC2 용량 예약의 결제 수락 또는 거부

사용자와 공유되는 용량 예약에 대한 결제 할당 요청을 받은 경우 이를 수락하거나 거부할 수 있습니다. 요청은 수락되거나 거부될 때까지 pending 상태로 유지됩니다.

요청을 수락하면 accepted 상태가 되고 그 시점부터 해당 용량 예약의 사용 가능한 용량 또는 미사용 용량에 대한 결제가 계정에 할당됩니다. 요청을 수락한 후에는 용량 예약 소유자만 계정에서 결제를 취소할 수 있습니다.

요청을 거부하면 `rejected` 상태가 되고 용량 예약의 사용 가능한 용량에 대한 결제는 용량 예약 소유자에게 할당됩니다.

요청이 12시간 이내에 수락되지 않거나 거부되면 요청은 만료됩니다. 요청이 만료되면 용량 예약의 미사용 용량에 대한 결제는 용량 예약 소유자에게 계속 할당됩니다.

요청이 수락되거나 거부되면 용량 예약 소유자의 계정으로 Amazon EventBridge 이벤트가 전송됩니다. 요청이 만료되면 용량 예약 소유자와 소비자 계정으로 Amazon EventBridge 이벤트가 전송됩니다. 자세한 내용은 공유 용량 예약에 대한 결제 할당 요청 모니터링 섹션을 참조하세요.

Console

요청을 수락 또는 거부하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 패널에서 용량 예약을 선택합니다.
3. 보류 중인 요청이 있는 경우 화면 상단에 보류 중인 결제 할당 요청 배너가 표시됩니다. 배너가 표시되지 않으면 보류 중인 요청이 없는 것입니다.

   요청을 보려면 배너에서 요청 검토를 선택합니다.
4. 수락하거나 거부할 요청을 선택한 다음 수락 또는 거부를 선택합니다.

AWS CLI

요청을 수락하려면

accept-capacity-reservation-billing-ownership 명령을 사용합니다. `--capacity-reservation-id`의 경우 요청을 수락할 용량 예약의 ID를 지정합니다.

```
aws ec2 accept-capacity-reservation-billing-ownership \
    --capacity-reservation-id cr-01234567890abcdef
```

요청을 거부하려면

reject-capacity-reservation-billing-ownership 명령을 사용합니다. `--capacity-reservation-id`의 요청을 거부할 용량 예약의 ID를 지정합니다.

```
aws ec2 reject-capacity-reservation-billing-ownership \
    --capacity-reservation-id cr-01234567890abcdef
```

PowerShell

요청을 수락하려면

Approve-EC2CapacityReservationBillingOwnership cmdlet을 사용합니다.

```
Approve-EC2CapacityReservationBillingOwnership `
    -CapacityReservationId cr-01234567890abcdef
```

요청을 거부하려면

Deny-EC2CapacityReservationBillingOwnership cmdlet을 사용합니다.

```
Deny-EC2CapacityReservationBillingOwnership `
    -CapacityReservationId cr-01234567890abcdef
```

공유 EC2 용량 예약에 대한 결제 할당 요청 취소 또는 철회

용량 예약 소유자만 pending 결제 할당 요청을 취소할 수 있습니다. 보류 중인 요청이 취소되면 cancelled 상태가 되고 용량 예약의 사용 가능 용량 또는 미사용 용량에 대한 결제는 용량 예약 소유자에게 계속 할당됩니다.

요청이 accepted된 이후에는 용량 예약 소유자만 할당된 계정에서 결제를 취소할 수 있습니다. 결제가 취소되면 요청은 revoked 상태가 되고 용량 예약의 사용 가능한 용량에 대한 결제가 용량 예약 소유자에게 다시 할당됩니다.

요청이 취소되거나 취소되면 용량 예약 소유자 및 지정된 소비자 계정으로 Amazon EventBridge 이벤트가 전송됩니다. 자세한 내용은 공유 용량 예약에 대한 결제 할당 요청 모니터링 섹션을 참조하세요.

Console

요청을 취소하거나 철회하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 패널에서 수용 인원 예약을 선택한 다음 요청을 취소하거나 철회할 용량 예약을 선택합니다.
3. 사용 가능한 용량 결제 섹션에서 요청의 현재 상태에 따라 이전 취소 또는 이전 철회를 선택합니다.

AWS CLI

요청을 취소하거나 철회하려면

disassociate-capacity-reservation-billing-owner 명령을 사용합니다. `--unused-reservation-billing-owner-id`의 경우 요청이 전송된 AWS 계정의 ID를 지정합니다.

```
aws ec2 disassociate-capacity-reservation-billing-owner \
    --capacity-reservation-id cr-01234567890abcdef \
    --unused-reservation-billing-owner-id 123456789012
```

PowerShell

요청을 취소하거나 철회하려면

Unregister-EC2CapacityReservationBillingOwner cmdlet을 사용합니다. `-UnusedReservationBillingOwnerId`의 경우 요청이 전송된 AWS 계정의 ID를 지정합니다.

```
Unregister-EC2CapacityReservationBillingOwner `
    -CapacityReservationId cr-01234567890abcdef `
    -UnusedReservationBillingOwnerId 123456789012
```

공유 용량 예약에 대한 결제 할당 요청 모니터링

Amazon EC2는 결제 할당 요청 상태가 변경될 때 Amazon EventBridge 이벤트를 전송합니다.

- 요청이 `accepted | rejected | expired | revoked` 상태가 되면 이벤트가 용량 예약 소유자에게 전송됩니다.
- 요청이 `pending | expired | cancelled | revoked` 상태가 되면 이벤트가 요청된 소비자 계정으로 전송됩니다.

Amazon EventBridge에 대한 자세한 내용은 Amazon EventBridge 사용 설명서를 참조하세요.

다음은 Amazon EventBridge 이벤트 패턴입니다.

```
{
   "version":"0",
   "id":"12345678-1234-1234-1234-123456789012",
```

```
   "detail-type":"On-Demand Capacity Reservation Billing Ownership Request pending|
accepted|rejected|cancelled|revoked|expired",
   "source":"aws.ec2",
   "account":"account_id",
   "time":"state_change_timestamp",
   "region":"region",
   "resources":[
      "arn:aws:ec2:region:cr_owner_account_id:capacity-reservation/cr_id"
   ],
   "detail":{
      "capacity-reservation-id":"cr_id",
      "updateTime":timestamp,
      "ownerAccountId":"cr_owner_account_id",
      "unusedReservationChargesOwnerID":"consumer_account_id",
      "status":"pending|accepted|rejected|cancelled|revoked|expired",
      "statusMessage":"message"
   }
}
```

다음은 소비자 계정(111111111111)이 공유 용량 예약(cr-01234567890abcdef)에 대한 청구 할당 요청을 수락할 때 용량 예약 소유자(222222222222)에게 전송되는 이벤트의 예시입니다.

```
{
   "version":"0",
   "id":"12345678-1234-1234-1234-123456789012",
   "detail-type":"On-Demand Capacity Reservation Billing Ownership Request accepted",
   "source":"aws.ec2",
   "account":"222222222222",
   "time":"2024-09-01Thh:59:59Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ec2:us-east-1:222222222222:capacity-reservation/cr-01234567890abcdef"
   ],
   "detail":{
      "capacity-reservation-id":"cr-01234567890abcdef",
      "updateTime":"2024-08-01Thh:59:59Z",
      "ownerAccountId":"222222222222",
      "unusedReservationChargesOwnerID":"111111111111",
      "status":"accepted",
      "statusMessage":"billing transfer status message"
   }
}
```

공유된 용량 예약 권한

소유자에 대한 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.

소비자에 대한 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.

기본적으로 용량 예약 소유자에게는 용량 예약 내에서 실행하는 인스턴스와 사용하지 않은 예약 용량에 대한 요금이 결제되며 소비자에게는 공유 용량 예약 내에서 실행하는 인스턴스에 대한 요금이 결제됩니다. 그러나 공유 용량 예약의 사용 가능한 용량에 대한 결제를 특정 소비자 계정에 할당할 수 있습니다. 자세한 내용은 공유 Amazon EC2 용량 예약에 대한 결제 할당 섹션을 참조하세요.

용량 예약 소유자가 다른 지급인 계정에 속해 있고 용량 예약이 리전 예약 인스턴스 또는 절감형 플랜의 적용을 받는 경우 용량 예약 소유자에게 리전 예약 인스턴스 또는 절감형 플랜에 대한 요금이 계속 청구됩니다. 이러한 경우 용량 예약 소유자는 리전 예약 인스턴스 또는 절감형 플랜에 대해 비용을 지불하고 소비자에게는 공유 용량 예약에서 실행되는 인스턴스에 대한 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.

용량 예약 플릿

온디맨드 용량 예약 플릿은 용량 예약의 그룹입니다.

용량 예약 플릿 요청에는 용량 예약 플릿을 시작하는 데 필요한 모든 구성 정보가 포함됩니다. 단일 요청을 사용하여 지정한 목표 용량까지 여러 인스턴스 유형에 걸쳐 워크로드에 사용할 대량의 Amazon EC2 용량을 예약할 수 있습니다.

용량 예약 플릿을 생성한 후 용량 예약 플릿을 수정하거나 취소하여 플릿의 용량 예약을 총괄적으로 관리할 수 있습니다.

주제

- 용량 예약 플릿 작동 방식
- 고려 사항
- 요금
- 용량 예약 플릿 개념 및 계획
- 용량 예약 플릿 생성
- 용량 예약 플릿 수정
- 용량 예약 플릿 취소
- 용량 예약 플릿 구성 예
- 용량 예약 플릿에 서비스 연결 역할 사용

용량 예약 플릿 작동 방식

용량 예약 플릿을 생성할 때 플릿은 플릿 요청에 지정한 총 목표 용량을 충족하는 개별 용량 예약을 생성하려고 시도합니다.

플릿이 용량을 예약하는 인스턴스 수는 지정하는 총 목표 용량과 인스턴스 유형 가중치에 따라 달라집니다. 용량을 예약하는 인스턴스 유형은 사용하는 할당 전략과 인스턴스 유형 우선순위에 따라 달라집니다.

플릿이 생성될 때 용량이 부족하여 총 목표 용량을 즉시 충족할 수 없는 경우 플릿은 요청된 용량을 예약할 때까지 비동기로 용량 예약을 시도합니다.

플릿이 총 목표 용량에 도달하면 해당 용량을 유지하려고 시도합니다. 플릿의 용량 예약이 취소되면 플릿은 플릿 구성에 따라 하나 이상의 용량 예약을 자동으로 생성하여 손실된 용량을 대체하고 총 목표 용량을 유지합니다.

플릿의 용량 예약은 개별적으로 관리할 수 없습니다. 플릿을 수정하여 총체적으로 관리해야 합니다. 플릿을 수정하면 플릿의 용량 예약이 변경 사항을 반영하도록 자동으로 업데이트됩니다.

현재 용량 예약 플릿은 open 인스턴스 일치 기준을 지원하며, 플릿에서 시작된 모든 용량 예약은 이 인스턴스 일치 기준을 자동으로 사용합니다. 이 기준을 사용하면 일치하는 속성(인스턴스 유형, 플랫폼, 가용 영역, 테넌시)이 있는 새 인스턴스와 기존 인스턴스가 플릿에서 생성한 용량 예약에서 자동으로 실행됩니다. 용량 예약 플릿은 대상 인스턴스 일치 기준을 지원하지 않습니다.

고려 사항

용량 예약 플릿으로 작업할 때는 다음 사항에 유의하세요.

- 용량 예약 플릿은 AWS CLI 및 AWS API를 사용하여 생성하고 수정하고 보고 취소할 수 있습니다.
- 플릿의 용량 예약은 개별적으로 관리할 수 없습니다. 플릿을 수정하거나 취소하여 총체적으로 관리해야 합니다.
- 용량 예약 플릿은 여러 리전에 걸쳐 있을 수 없습니다.
- 용량 예약 플릿은 여러 가용 영역에 걸쳐 있을 수 없습니다.
- 용량 예약 플릿에 의해 생성된 용량 예약에는 자동으로 다음 AWS 생성 태그가 태깅됩니다.
  - 키 - `aws:ec2-capacity-reservation-fleet`
  - 값 - *`fleet_id`*

  이 태그를 사용하여 용량 예약 플릿에 의해 생성된 용량 예약을 식별할 수 있습니다.

요금

용량 예약 플릿을 사용하는 데 따른 추가 비용은 없습니다. 용량 예약 플릿에 의해 생성된 개별 용량 예약에 대한 요금이 청구됩니다. 용량 예약의 요금 청구에 대한 자세한 내용은 용량 예약 요금 및 결제 섹션을 참조하세요.

용량 예약 플릿 개념 및 계획

다음 정보는 용량 예약 플릿을 계획하는 방법을 설명하고 총 목표 용량, 할당 전략, 인스턴스 유형 가중치, 인스턴스 유형 우선순위를 포함한 용량 예약 플릿 개념을 설명합니다.

주제

- 용량 예약 플릿 계획
- 총 목표 용량
- 할당 전략

- 인스턴스 유형 가중치
- 인스턴스 유형 우선순위

용량 예약 플릿 계획

용량 예약 플릿을 계획할 때 다음을 수행하는 것이 좋습니다.

1. 워크로드에 필요한 컴퓨팅 용량을 결정합니다.
2. 사용하려는 인스턴스 유형 및 가용 영역을 결정합니다.
3. 요구 사항 및 기본 설정에 따라 각 인스턴스 유형에 우선순위를 할당합니다. 자세한 내용은 인스턴스 유형 우선순위 섹션을 참조하세요.
4. 워크로드에 적합한 용량 가중치 시스템을 만듭니다. 각 인스턴스 유형에 가중치를 할당하고 총 목표 용량을 결정합니다. 자세한 내용은 인스턴스 유형 가중치 및 총 목표 용량 섹션을 참조하세요.
5. 용량 예약이 무기한 필요한지 아니면 특정 기간 동안만 필요한지 결정합니다.

총 목표 용량

총 목표 용량은 용량 예약 플릿이 예약하는 총 컴퓨팅 용량을 정의합니다. 총 목표 용량은 용량 예약 플릿을 생성할 때 지정합니다. 플릿이 생성되고 나면, Amazon EC2가 용량 예약을 자동으로 생성하여 총 목표 용량까지 용량을 예약합니다.

용량 예약 플릿이 용량을 예약하는 인스턴스 수는 총 목표 용량과 용량 예약 플릿의 각 인스턴스 유형에 대해 지정하는 인스턴스 유형 가중치(`total target capacity//instance type weight==number of instances`).)에 따라 결정됩니다.

워크로드에 의미 있는 단위를 기준으로 총 목표 용량을 할당할 수 있습니다. 예를 들어 워크로드에 특정 수의 vCPU가 필요한 경우 필요한 vCPU 수를 기준으로 총 목표 용량을 할당할 수 있습니다. 워크로드에 2048개의 vCPU가 필요한 경우, 총 목표 용량을 2048로 지정한 다음 플릿의 인스턴스 유형이 제공하는 vCPU 수에 따라 인스턴스 유형 가중치를 할당합니다. 예시는 인스턴스 유형 가중치에서 확인하세요.

할당 전략

용량 예약 플릿의 할당 전략에 따라 용량 예약 플릿 구성의 인스턴스 유형 사양에서 예약 용량 요청을 이행하는 방법이 결정됩니다.

현재는 `prioritized` 할당 전략만 지원됩니다. 이 전략을 통해 용량 예약 플릿은 용량 예약 플릿 구성의 각 인스턴스 유형 사양에 지정한 우선순위를 사용하여 용량 예약을 생성합니다. 우선순위 값이 낮을

수록 사용 우선순위가 높음을 나타냅니다. 예를 들어 다음과 같은 인스턴스 유형 및 우선순위를 사용하는 용량 예약 플릿을 생성한다고 가정해 보겠습니다.

- m4.16xlarge - 우선순위 = 1
- m5.16xlarge - 우선순위 = 3
- m5.24xlarge - 우선순위 = 2

플릿은 먼저 m4.16xlarge의 용량 예약을 생성하려고 시도합니다. Amazon EC2의 m4.16xlarge 용량이 부족한 경우, 플릿은 m5.24xlarge의 용량 예약을 생성하려고 시도합니다. Amazon EC2의 m5.24xlarge 용량이 부족한 경우, 플릿은 m5.16xlarge의 용량 예약을 생성합니다.

인스턴스 유형 가중치

인스턴스 유형 가중치는 용량 예약 플릿의 각 인스턴스 유형에 할당하는 가중치입니다. 이 가중치는 특정 인스턴스 유형의 각 인스턴스가 플릿의 총 목표 용량에서 차지하는 용량 단위 수를 결정합니다.

워크로드에 의미 있는 단위를 기준으로 가중치를 할당할 수 있습니다. 예를 들어 워크로드에 특정 수의 vCPU가 필요한 경우 용량 예약 플릿의 각 인스턴스 유형에서 제공하는 vCPU 수를 기준으로 가중치를 할당할 수 있습니다. 이 경우 m4.16xlarge 및 m5.24xlarge 인스턴스를 사용하여 용량 예약 플릿을 생성한다면, 다음과 같이 각 인스턴스의 vCPU 수에 해당하는 가중치를 할당합니다.

- m4.16xlarge — 64개의 vCPU, 가중치 = 64단위
- m5.24xlarge — 96개의 vCPU, 가중치 = 96단위

인스턴스 유형 가중치에 따라 용량 예약 플릿이 용량을 예약하는 인스턴스의 수가 결정됩니다. 예를 들어 총 목표 용량이 384단위이고 앞의 예와 동일한 인스턴스 유형과 가중치를 사용하는 용량 예약 플릿의 경우, 6개의 m4.16xlarge 인스턴스(총 대상 용량 384/인스턴스 유형 64개 가중치=인스턴스 6개) 또는 4개의 m5.24xlarge 인스턴스(384/96 = 4) 용량을 예약할 수 있습니다.

인스턴스 유형 가중치를 할당하지 않거나 가중치가 1인 인스턴스 유형 가중치를 할당할 경우, 총 목표 용량은 온전히 인스턴스 수를 기준으로 합니다. 예를 들어 총 목표 용량이 384단위이고 앞의 예와 동일한 인스턴스 유형을 사용하지만 가중치를 생략하거나 두 인스턴스 유형 모두의 가중치를 1로 지정한 용량 예약 플릿의 경우, 플릿은 384 m4.16xlarge 인스턴스 또는 384 m5.24xlarge 인스턴스 중 하나의 용량을 예약할 수 있습니다.

인스턴스 유형 우선순위

인스턴스 유형 우선순위는 플릿의 인스턴스 유형에 할당하는 값입니다. 우선순위는 플릿에 지정된 인스턴스 유형 중 플릿이 우선적으로 사용해야 할 인스턴스 유형을 결정하는 데 사용됩니다.

우선순위 값이 낮을수록 사용 우선순위가 높음을 나타냅니다.

용량 예약 플릿 생성

용량 예약 플릿을 생성하면, 지정된 총 목표 용량까지 플릿 요청에 지정된 인스턴스 유형의 용량 예약이 자동으로 생성됩니다. 용량 예약 플릿이 용량을 예약하는 인스턴스 수는 요청에 지정하는 총 목표 용량 및 인스턴스 유형 가중치에 따라 달라집니다. 자세한 내용은 인스턴스 유형 가중치 및 총 목표 용량 섹션을 참조하세요.

플릿을 생성할 때 사용할 인스턴스 유형과 각 인스턴스 유형의 우선순위를 지정해야 합니다. 자세한 내용은 할당 전략 및 인스턴스 유형 우선순위 섹션을 참조하세요.

> **Note**
>
> 용량 예약 플릿을 처음 생성할 때 계정에 AWSServiceRoleForEC2CapacityReservationFleet 서비스 연결 역할이 자동으로 생성됩니다. 자세한 내용은 용량 예약 플릿에 서비스 연결 역할 사용 섹션을 참조하세요.

현재 용량 예약 플릿은 open 인스턴스 일치 기준만 지원합니다.

AWS CLI

용량 예약 플릿을 생성하려면

create-capacity-reservation-fleet 명령을 사용합니다.

```
aws ec2 create-capacity-reservation-fleet \
    --total-target-capacity 24 \
    --allocation-strategy prioritized \
    --instance-match-criteria open \
    --tenancy default \
    --end-date 2021-12-31T23:59:59.000Z \
    --instance-type-specifications file://instanceTypeSpecification.json
```

다음은 instanceTypeSpecification.json의 내용입니다.

```
[
  {
    "InstanceType": "m5.xlarge",
```

```
    "InstancePlatform": "Linux/UNIX",
    "Weight": 3.0,
    "AvailabilityZone":"us-east-1a",
    "EbsOptimized": true,
    "Priority" : 1
  }
]
```

출력의 예시는 다음과 같습니다.

```
{
    "Status": "submitted",
    "TotalFulfilledCapacity": 0.0,
    "CapacityReservationFleetId": "crf-abcdef01234567890",
    "TotalTargetCapacity": 24
}
```

PowerShell

용량 예약 플릿 생성

New-EC2CapacityReservationFleet cmdlet을 사용합니다.

```
New-EC2CapacityReservationFleet `
    -TotalTargetCapacity 24 `
    -AllocationStrategy "prioritized" `
    -InstanceMatchCriterion "open" `
    -Tenancy "default" `
    -EndDate 2021-12-31T23:59:59.000Z `
    -InstanceTypeSpecification $specification
```

사양은 다음과 같이 정의됩니다.

```
$specification = New-Object Amazon.EC2.Model.ReservationFleetInstanceSpecification
$specification.InstanceType = "m5.xlarge"
$specification.InstancePlatform = "Linux/UNIX"
$specification.Weight = 3.0
$specification.AvailabilityZone = "us-east-1a"
$specification.EbsOptimized = $true
$specification.Priority = 1
```

용량 예약 플릿 수정

용량 예약 플릿의 총 목표 용량과 날짜를 언제든지 수정할 수 있습니다. 용량 예약 플릿의 총 목표 용량을 수정하면 플릿이 자동으로 새 용량 예약을 생성하거나, 새 총 목표 용량을 충족하도록 플릿의 기존 용량 예약을 수정 또는 취소합니다. 플릿의 종료 날짜를 수정하면 모든 개별 용량 예약의 종료 날짜가 그에 따라 업데이트됩니다.

고려 사항

- 플릿을 수정하고 나면 그 상태가 `modifying`으로 전환됩니다. `modifying` 상태인 동안에는 플릿에 대해 추가 수정을 시도할 수 없습니다.
- 용량 예약 플릿에 사용되는 테넌시, 가용 영역, 인스턴스 유형, 인스턴스 플랫폼, 우선순위 또는 가중치는 수정할 수 없습니다. 이러한 파라미터를 변경해야 하는 경우 기존 플릿을 취소하고 필요한 파라미터가 적용된 새 플릿을 생성해야 할 수 있습니다.
- `--end-date`와 `--remove-end-date`를 같은 명령에 지정할 수는 없습니다.

AWS CLI

용량 예약 플릿을 수정하려면

modify-capacity-reservation-fleet 명령을 사용합니다.

예제 1: 총 목표 용량 수정

```
aws ec2 modify-capacity-reservation-fleet \
    --capacity-reservation-fleet-id crf-01234567890abcedf \
    --total-target-capacity 160
```

예제 2: 종료 날짜 수정

```
aws ec2 modify-capacity-reservation-fleet \
    --capacity-reservation-fleet-id crf-01234567890abcedf \
    --end-date 2021-07-04T23:59:59.000Z
```

예제 3: 종료 날짜 제거

```
aws ec2 modify-capacity-reservation-fleet \
    --capacity-reservation-fleet-id crf-01234567890abcedf \
    --remove-end-date
```

PowerShell

용량 예약 플릿을 수정하려면

Edit-EC2CapacityReservationFleet cmdlet을 사용합니다.

예제 1: 총 목표 용량 수정

```
Edit-EC2CapacityReservationFleet `
    -CapacityReservationFleetId crf-01234567890abcedf `
    -TotalTargetCapacity 160
```

예제 2: 종료 날짜 수정

```
Edit-EC2CapacityReservationFleet `
    -CapacityReservationFleetId crf-01234567890abcedf `
    -EndDate 2021-07-04T23:59:59.000Z
```

예제 3: 종료 날짜 제거

```
Edit-EC2CapacityReservationFleet `
    -CapacityReservationFleetId crf-01234567890abcedf `
    -RemoveEndDate
```

용량 예약 플릿 취소

용량 예약 플릿과 플릿에 의해 예약된 용량이 더 이상 필요하지 않은 경우 취소할 수 있습니다. 플릿을 취소하면 상태가 cancelled로 바뀌고 더 이상 새 용량 예약을 생성할 수 없습니다. 또한 플릿의 모든 개별 용량 예약이 취소됩니다. 이전에 예약 용량에서 실행 중이었던 인스턴스는 공유 용량에서 계속 정상적으로 실행됩니다.

AWS CLI

용량 예약 플릿 취소

cancel-capacity-reservation-fleets 명령을 사용합니다.

```
aws ec2 cancel-capacity-reservation-fleets \
    --capacity-reservation-fleet-ids crf-abcdef01234567890
```

출력의 예시는 다음과 같습니다.

```
{
    "SuccessfulFleetCancellations": [
        {
            "CurrentFleetState": "cancelling",
            "PreviousFleetState": "active",
            "CapacityReservationFleetId": "crf-abcdef01234567890"
        }
    ],
    "FailedFleetCancellations": []
}
```

PowerShell

용량 예약 플릿 취소

Stop-EC2CapacityReservationFleet cmdlet을 사용합니다.

```
Stop-EC2CapacityReservationFleet `
    -CapacityReservationFleetId crf-abcdef01234567890
```

용량 예약 플릿 구성 예

다음 예에서는 m5.4xlarge와 m5.12xlarge의 두 가지 인스턴스 유형을 사용하는 용량 예약 플릿을 생성합니다.

지정된 인스턴스 유형에서 제공하는 vCPU 수에 따라 가중치 시스템을 사용합니다. 총 목표 용량은 vCPU 480개입니다. m5.4xlarge는 16개의 vCPU를 제공하며 가중치 16이 할당되고, m5.12xlarge는 48개의 vCPU 제공하며 가중치 48이 할당됩니다. 이 가중치 시스템은 30개의 m5.4xlarge 인스턴스(480/16=30) 또는 10개의 m5.12xlarge 인스턴스(480/48=10) 용량을 예약하는 용량 예약 플릿을 구성합니다.

이 플릿은 m5.12xlarge 용량을 우선으로 하고(우선순위 1 할당), m5.4xlarge에는 더 낮은 우선순위 2를 할당하도록 구성됩니다. 따라서 플릿이 m5.12xlarge 용량 예약을 먼저 시도하고 Amazon EC2에 m5.12xlarge 용량이 부족한 경우에만 m5.4xlarge 용량 예약을 시도합니다.

이 플릿은Windows 인스턴스의 용량을 예약하며 October 31, 2021 23:59:59(UTC)에 예약이 자동으로 만료됩니다.

```
aws ec2 create-capacity-reservation-fleet \
```

```
    --total-target-capacity 480 \
    --allocation-strategy prioritized \
    --instance-match-criteria open \
    --tenancy default \
    --end-date 2021-10-31T23:59:59.000Z \
    --instance-type-specifications file://instanceTypeSpecification.json
```

다음은 `instanceTypeSpecification.json`의 내용입니다.

```
[
    {
        "InstanceType": "m5.4xlarge",
        "InstancePlatform":"Windows",
        "Weight": 16,
        "AvailabilityZone":"us-east-1a",
        "EbsOptimized": true,
        "Priority" : 2
    },
    {
        "InstanceType": "m5.12xlarge",
        "InstancePlatform":"Windows",
        "Weight": 48,
        "AvailabilityZone":"us-east-1a",
        "EbsOptimized": true,
        "Priority" : 1
    }
]
```

용량 예약 플릿에 서비스 연결 역할 사용

온디맨드 용량 예약 플릿은 AWS Identity and Access Management(IAM) 서비스 연결 역할을 사용합니다. 서비스 연결 역할은용량 예약 플릿에 직접 연결된 고유한 유형의 IAM 역할입니다. 서비스 연결 역할은 용량 예약 플릿에서 사전 정의하며 서비스에서 다른 AWS 서비스를 자동으로 호출하기 위해 필요한 모든 권한을 포함합니다.

서비스 연결 역할을 사용하면 필요한 권한을 수동으로 추가할 필요가 없으므로용량 예약 플릿을 더 쉽게 설정할 수 있습니다. 용량 예약 플릿에서 서비스 연결 역할의 권한을 정의하므로 다르게 정의되지 않은 한, 용량 예약 플릿만 해당 역할을 수임할 수 있습니다. 정의된 권한에는 신뢰 정책과 권한 정책이 포함되며 이 권한 정책은 다른 IAM 엔티티에 연결할 수 없습니다.

먼저 관련 리소스를 삭제한 후에만 서비스 연결 역할을 삭제할 수 있습니다. 이렇게 하면 리소스에 대한 액세스 권한을 부주의로 삭제할 수 없기 때문에 용량 예약 플릿 리소스가 보호됩니다.

용량 예약 플릿의 서비스 연결 역할 권한

용량 예약 플릿은 AWSServiceRoleForEC2CapacityReservationFleet이라는 서비스 연결 역할을 사용하여 사용자를 대신해 용량 예약 플릿에서 용량 예약을 생성, 설명, 수정 및 취소합니다.

AWSServiceRoleForEC2CapacityReservationFleet 서비스 연결 역할은 역할을 수임하기 위해 다음 엔터티를 신뢰합니다.

- `capacity-reservation-fleet.amazonaws.com`

이 역할에서는 `AWSEC2CapacityReservationFleetRolePolicy` AWS 관리형 정책을 사용합니다. 자세한 내용은 AWS 관리형 정책: AWSEC2CapacityReservationFleetRolePolicy 섹션을 참조하세요.

IAM 엔터티(사용자, 그룹, 역할 등)가 서비스 링크 역할을 생성하고 편집하거나 삭제할 수 있도록 권한을 구성할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 권한을 참조하세요.

용량 예약 플릿의 서비스 연결 역할 생성

서비스 링크 역할은 수동으로 생성할 필요가 없습니다. `create-capacity-reservation-fleet` AWS CLI 명령 또는 `CreateCapacityReservationFleet` API를 사용하여 용량 예약 플릿을 생성할 경우 서비스 연결 역할이 자동으로 생성됩니다.

이 서비스 연결 역할을 삭제했다가 다시 생성해야 하는 경우 동일한 프로세스를 사용하여 계정에서 역할을 다시 생성할 수 있습니다. 용량 예약 플릿을 생성할 때, 용량 예약 플릿이 서비스 연결 역할을 다시 생성합니다.

용량 예약 플릿의 서비스 연결 역할 편집

용량 예약 플릿에서는 AWSServiceRoleForEC2CapacityReservationFleet 서비스 연결 역할을 편집할 수 없습니다. 서비스 링크 역할을 생성한 후에는 다양한 개체가 역할을 참조할 수 있기 때문에 역할 이름을 변경할 수 없습니다. 하지만 IAM을 사용하여 역할의 설명을 편집할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 설명 편집을 참조하세요.

용량 예약 플릿의 서비스 연결 역할 삭제

서비스 연결 역할이 필요한 기능 또는 서비스가 더 이상 필요 없는 경우에는 해당 역할을 삭제하는 것이 좋습니다. 따라서 적극적으로 모니터링하거나 유지하지 않는 미사용 엔터티가 없도록 합니다. 단, 서비스 연결 역할에 대한 리소스를 먼저 삭제해야 수동으로 삭제할 수 있습니다.

> **Note**
>
> 리소스를 삭제하려 할 때 용량 예약 플릿 서비스가 역할을 사용 중이면 삭제에 실패할 수 있습니다. 이 문제가 발생하면 몇 분 기다렸다가 작업을 다시 시도하세요.

AWSServiceRoleForEC2CapacityReservationFleet 서비스 연결 역할을 삭제하려면

1. `delete-capacity-reservation-fleet` AWS CLI 명령 또는 `DeleteCapacityReservationFleet` API를 사용하여 계정에서 용량 예약 플릿을 삭제합니다.
2. IAM 콘솔, AWS CLI 또는 AWS API를 사용하여 AWSServiceRoleForEC2CapacityReservationFleet 서비스 연결 역할을 삭제합니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 삭제를 참조하세요.

용량 예약 플릿 서비스 연결 역할을 지원하는 리전

용량 예약 플릿에서는 서비스를 사용할 수 있는 모든 리전에서 서비스 연결 역할 사용을 지원합니다. 자세한 내용은 AWS 리전 및 엔드포인트 섹션을 참조하세요.

CloudWatch 지표를 사용하여 용량 예약 모니터링

CloudWatch 지표를 사용하면 사용량 임계값이 충족될 때 알림을 보내도록 CloudWatch 경보를 설정하여 용량 예약를 효율적으로 모니터링하고 사용하지 않은 용량을 식별할 수 있습니다. 이를 통해 일정한 용량 예약 볼륨을 유지하고 더 높은 수준의 사용률을 달성할 수 있습니다.

용량 예약은 5분마다 CloudWatch로 지표 데이터를 전송합니다. 5분 미만 동안 활성 상태인 용량 예약에는 지표가 지원되지 않습니다.

CloudWatch 콘솔에서 지표를 보는 방법에 대한 자세한 내용은 Amazon CloudWatch 지표 사용을 참조하세요. 경보 생성에 대한 자세한 내용은 Amazon CloudWatch 경보 생성을 참조하세요.

목차

- 용량 예약 사용량 지표
- 용량 예약 지표 차원
- 용량 예약의 CloudWatch 지표 보기

용량 예약 사용량 지표

AWS/EC2CapacityReservations 네임스페이스에는 예약에 대해 지정한 임계값 내에서 온디맨드 용량을 모니터링하고 유지 관리하는 데 사용할 수 있는 다음과 같은 사용량 지표가 포함됩니다.

측정치	설명
UsedInstanceCount	현재 사용 중인 인스턴스 수입니다. 단위: 수
AvailableInstanceCount	사용 가능한 인스턴스 수입니다. 단위: 수
TotalInstanceCount	예약한 총 인스턴스 수입니다. 단위: 수
InstanceUtilization	현재 사용 중인 예약 용량 인스턴스의 비율입니다. 단위: 백분율

용량 예약 지표 차원

다음 차원을 사용하여 선택한 리전 및 계정 내에서 이전 표에 나열된 지표를 구체화할 수 있습니다.

차원	설명
(차원 없음)	이 차원은 모든 용량 예약에 대해 지정된 지표를 필터링합니다.
CapacityReservationId	이 차원은 식별된 용량 예약에 대해 지정된 지표를 필터링합니다.
InstanceType	이 차원은 식별된 인스턴스 유형에 대해 지정된 지표를 필터링합니다.

차원	설명
AvailabilityZone	이 차원은 식별된 가용 영역에 대해 지정된 지표를 필터링합니다.
InstanceMatchCriteria	이 차원은 식별된 인스턴스 일치 기준(open 또는 targeted)에 대해 지정된 지표를 필터링합니다.
InstancePlatform	이 차원은 식별된 플랫폼에 대해 지정된 지표 데이터를 필터링합니다.
Tenancy	이 차원은 식별된 테넌시에 대해 지정된 지표를 필터링합니다.

용량 예약의 CloudWatch 지표 보기

지표는 먼저 서비스 네임스페이스별로 그룹화된 다음, 각 네임스페이스 내에서 지원되는 차원별로 그룹화됩니다. 다음 절차에 따라 용량 예약에 대한 지표를 볼 수 있습니다.

CloudWatch 콘솔을 사용하여 용량 예약 지표를 보려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 필요한 경우 리전을 변경합니다. 탐색 모음에서 용량 예약이 상주하는 리전을 선택합니다. 자세한 설명은 리전 및 엔드포인트를 참조하십시오.
3. 탐색 창에서 지표(Metrics)를 선택합니다.
4. 모든 지표에 대해 EC2 용량 예약을 선택합니다.
5. 모든 용량 예약에서, 용량 예약 기준, 인스턴스 유형 기준, 가용 영역 기준, 플랫폼 기준, 인스턴스 일치 기준, 테넌시 기준 지표 차원을 선택하면 지표가 차원 없음, CapacityReservationId, InstanceType, AvailabilityZone, Platform, InstanceMatchCriteria, Tenancy에 따라 그룹화됩니다.
6. 지표를 정렬하려면 열 머리글을 사용합니다. 지표를 그래프로 표시하려면 지표 옆에 있는 확인란을 선택합니다.

AWS CLI를 사용하여 용량 예약 지표를 보는 방법

다음 list-metrics 명령을 사용합니다.

```
aws cloudwatch list-metrics --namespace "AWS/EC2CapacityReservations"
```

용량 예약 사용률 저하 모니터링

다음을 사용하여 용량 예약 사용률 저하를 모니터링할 수 있습니다.

주제

- Amazon EventBridge 이벤트
- 이메일 및 AWS Health 대시보드 알림

Amazon EventBridge 이벤트

계정의 용량 예약의 사용량이 일정 기간 동안 20% 미만이면 AWS Health가 Amazon EventBridge로 이벤트를 전송합니다. EventBridge에서는 이러한 이벤트에 대한 응답으로 프로그래밍 작업을 트리거하는 규칙을 설정할 수 있습니다. 예를 들어 7일 동안 사용률이 20% 미만인 용량 예약을 자동으로 취소하는 규칙을 생성할 수 있습니다.

EventBridge의 이벤트는 JSON 객체로 표현됩니다. 이 이벤트에 고유한 필드는 JSON 객체의 "세부 정보" 섹션에 포함되어 있습니다. "이벤트" 필드에는 이벤트 이름이 포함됩니다. "결과" 필드에는 이벤트를 트리거한 작업의 완료 상태가 포함됩니다. 자세한 내용은 Amazon EventBridge 사용 설명서의 Amazon EventBridge 이벤트 패턴을 참조하세요.

자세한 내용은 Amazon EventBridge 사용 설명서를 참조하세요.

이 기능은 AWS GovCloud (US)에서 지원되지 않습니다.

이벤트

용량 예약의 용량 사용량이 20% 미만이면 AWS Health가 다음 이벤트를 전송합니다.

- `AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION`

  다음은 새로 생성된 용량 예약의 용량 사용량이 24시간 동안 20% 미만일 경우 생성되는 이벤트의 예입니다.

```
{
    "version": "0",
    "id": "b3e00086-f271-12a1-a36c-55e8ddaa130a",
    "detail-type": "AWS Health Event",
    "source": "aws.health",
```

```
    "account": "123456789012",
    "time": "2023-03-10T12:03:38Z",
    "region": "ap-south-1",
    "resources": [
        "cr-01234567890abcdef"
    ],
    "detail": {
        "eventArn": "arn:aws:health:ap-south-1::event/EC2/
AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION/
AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION_cr-01234567890abcdef-6211-4d50-9286-0c9fbc243f04",
        "service": "EC2",
        "eventTypeCode": "AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION",
        "eventTypeCategory": "accountNotification",
        "startTime": "Fri, 10 Mar 2023 12:03:38 GMT",
        "endTime": "Fri, 10 Mar 2023 12:03:38 GMT",
        "eventDescription": [
            {
                "language": "en_US",
                "latestDescription": "A description of the event will be provided
 here"
            }
        ],
        "affectedEntities": [
            {
                "entityValue": "cr-01234567890abcdef"
            }
        ]
    }
    }
```

- AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION_SUMMARY

  다음은 새로 생성된 용량 예약 중 하나 이상의 용량 사용량이 7일 동안 20% 미만일 경우 생성되는 이벤트의 예입니다.

```
{
    "version": "0", "id":"7439d42b-3c7f-ad50-6a88-25e2a70977e2",
    "detail-type": "AWS Health Event",
    "source": "aws.health",
    "account": "123456789012",
    "time": "2023-03-07T06:06:01Z",
    "region": "us-east-1",
    "resources": [
```

```
        "cr-01234567890abcdef | us-east-1b | t3.medium | Linux/UNIX | 0.0%",
        "cr-09876543210fedcba | us-east-1a | t3.medium | Linux/UNIX | 0.0%"
    ],
    "detail": {
        "eventArn": "arn:aws:health:us-east-1::event/
EC2/AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION_SUMMARY/
AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION_SUMMARY_726c1732-d6f6-4037-b9b8-
bec3c2d3ba65",
        "service": "EC2",
        "eventTypeCode": "AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION_SUMMARY",
        "eventTypeCategory": "accountNotification",
        "startTime": "Tue, 7 Mar 2023 06:06:01 GMT",
        "endTime": "Tue, 7 Mar 2023 06:06:01 GMT",
        "eventDescription": [
            {
                "language": "en_US",
                "latestDescription": "A description of the event will be provided
 here"
            }
        ],
        "affectedEntities": [
            {
                "entityValue": "cr-01234567890abcdef | us-east-1b | t3.medium |
 Linux/UNIX | 0.0%"
            },
            {
                "entityValue": "cr-09876543210fedcba | us-east-1a | t3.medium |
 Linux/UNIX | 0.0%"
            }
        ]
    }
}
```

EventBridge 규칙 생성

용량 예약 사용률이 20% 미만으로 떨어질 경우 이메일 알림을 받으려면 Amazon SNS 주제를 생성한 다음 `AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION` 이벤트에 대한 EventBridge 규칙을 생성합니다.

Amazon SNS 주제를 생성하려면

1. https://console.aws.amazon.com/sns/v3/home에서 Amazon SNS 콘솔을 엽니다.

2. 탐색 창에서 토픽을 선택한 다음, 토픽 생성을 선택합니다.
3. 유형에서 표준을 선택합니다.
4. 이름에 새 주제의 이름을 입력합니다.
5. 주제 생성을 선택합니다.
6. 구독 생성을 선택합니다.
7. 프로토콜에서 이메일을 선택한 다음 엔드포인트에 알림을 받는 데 사용할 이메일 주소를 입력합니다.
8. 구독 생성을 선택합니다.
9. 위에 입력한 이메일 주소로 `AWS Notification - Subscription Confirmation`(이)라는 제목의 이메일 메시지를 받게 됩니다. 지시에 따라 구독을 확인합니다.

EventBridge 규칙을 만들려면

1. https://console.aws.amazon.com/events/에서 Amazon EventBridge 콘솔을 엽니다.
2. 탐색 창에서 규칙(Rules)을 선택한 후 규칙 생성(Create rule)을 선택합니다.
3. 이름에 새 규칙의 이름을 입력합니다.
4. 규칙 유형(Rule type)에서 이벤트 패턴이 있는 규칙(Rule with an event pattern)을 생성합니다.
5. 다음을 선택합니다.
6. 이벤트 패턴에서 다음을 수행합니다.
   a. 이벤트 소스에서 AWS 서비스를 선택합니다.
   b. AWS 서비스에서 AWS Health를 선택합니다.
   c. 이벤트 유형에서 EC2 ODCR 사용률 부족 알림을 선택합니다.
7. 다음을 선택합니다.
8. 대상 1에서 다음을 수행합니다.
   a. 대상 유형에서 AWS서비스를 선택합니다.
   b. 대상 선택에서 SNS 주제를 선택합니다.
   c. 주제에서 앞서 생성한 주제를 선택합니다.
9. 다음을 선택한 후 다음을 다시 한번 선택합니다.
10. 규칙 생성을 선택합니다.

이메일 및 AWS Health 대시보드 알림

계정에서 용량 예약의 용량 사용률이 20% 미만으로 떨어지면 AWS Health가 다음 이메일과 AWS Health Dashboard 알림을 보냅니다.

- 새로 생성된 각 용량 예약 중 지난 24시간 동안 사용률이 20% 미만인 각 용량 예약에 대한 개별 알림.
- 지난 7일 동안 사용률이 20% 미만인 모든 용량 예약에 대한 요약 알림.

이메일 알림과 AWS Health Dashboard 알림은 용량 예약을 소유한 AWS 계정에 연결된 이메일 주소로 전송됩니다. 이 알림에는 다음 정보가 포함됩니다.

- 용량 예약의 ID입니다.
- 용량 예약의 가용 영역.
- 용량 예약의 평균 사용률.
- 용량 예약의 인스턴스 유형 및 플랫폼(운영 체제).

또한 계정에서 용량 예약의 용량 사용률이 24시간과 7일 동안 20% 미만이면 AWS Health가 EventBridge로 이벤트를 전송합니다. EventBridge를 사용하면, 이메일 알림을 전송하거나 AWS Lambda 함수를 트리거하는 등의 자동 작업을 이러한 이벤트에 대한 응답으로 설정하는 규칙을 만들 수 있습니다. 자세한 내용은 용량 예약 사용률 저하 모니터링 섹션을 참조하세요.

장래 날짜 용량 예약의 상태 변경 모니터링

Amazon EC2는 장래 날짜 용량 예약의 상태가 변경될 때 Amazon EventBridge로 이벤트를 전송합니다.

다음은 이 이벤트의 예입니다. 이 예제에서는 장래 날짜 용량 예약이 `scheduled` 상태로 전환되었습니다. `detail-type` 필드에 강조 표시된 상태를 기록해 둡니다.

```
{
   "version":"0",
   "id":"12345678-1234-1234-1234-123456789012",
   "detail-type":"EC2 Capacity Reservation Scheduled",
   "source":"aws.ec2",
   "account":"123456789012",
   "time":"yyyy-mm-ddThh:mm:ssZ",
   "region":"us-east-1",
```

```
   "resources":[
      "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-1234567890abcdefg"
   ],
   "detail":{
      "capacity-reservation-id":"cr-1234567890abcdefg",
      "state":"scheduled"
   }
}
```

detail-type 필터에 사용할 수 있는 값은 다음과 같습니다.

- Scheduled
- Active
- Delayed
- Unsupported
- Failed
- Expired

이러한 상태에 대한 자세한 내용은 용량 예약 상태 보기 섹션을 참조하세요.

이러한 이벤트를 모니터링한 다음 특정 작업이 발생할 때 트리거하는 Amazon EventBridge 이벤트를 생성할 수 있습니다. 자세한 내용은 Amazon EventBridge에서 이벤트에 반응하는 규칙 생성을 참조하세요.

모든 상태 변경 이벤트를 모니터링하는 규칙을 생성할 때 다음 이벤트 패턴을 사용할 수 있습니다.

```
{
  "source": ["aws.ec2"],
  "detail-type": [{
    "prefix": "EC2 Capacity Reservation"
  }]
}
```

특정 상태 변경만 모니터링하는 규칙을 생성할 때 다음 이벤트 패턴을 사용할 수 있습니다.

```
{
  "source": ["aws.ec2"],
  "detail-type": [{
```

```
    "prefix": "EC2 Capacity Reservation state"
  }]
}
```

예를 들어 다음 이벤트 패턴은 장래 날짜 용량 예약이 `active` 상태가 될 때 전송되는 이벤트를 모니터링합니다.

```
{
  "source": ["aws.ec2"],
  "detail-type": [{
    "prefix": "EC2 Capacity Reservation Active"
  }]
}
```

ML용 용량 블록

ML용 용량 블록을 사용하면 향후 날짜에 GPU 기반 가속 컴퓨팅 인스턴스를 예약하여 단기간의 기계 학습(ML) 워크로드를 지원할 수 있습니다. 용량 블록 내부에서 실행되는 인스턴스는 지연 시간이 짧은 페타비트 규모의 비차단 네트워킹을 위해 Amazon EC2 UltraClusters 내부에 자동으로 서로 가깝게 배치됩니다.

용량 블록을 사용하면 미래 날짜에 GPU 인스턴스 용량을 사용할 수 있는 시점을 확인하고, 가장 적합한 시간에 시작하도록 용량 블록을 예약할 수 있습니다. 용량 블록을 예약하면 필요한 시간에 해당하는 비용만 결제하면서 GPU 인스턴스의 용량 보증을 예측할 수 있습니다. 한 번에 며칠 또는 몇 주 동안 ML 워크로드를 지원하는 GPU가 필요하고 GPU 인스턴스를 사용하지 않는 동안에는 예약을 결제하지 않으려는 경우에 용량 블록을 사용하는 것이 좋습니다.

다음은 용량 블록의 몇 가지 일반적인 사용 사례입니다.

- ML 모델 훈련 및 미세 조정 – 예약한 GPU 인스턴스에 중단 없이 액세스하여 ML 모델 훈련 및 미세 조정을 완료합니다.
- ML 실험 및 프로토타입 – GPU 인스턴스가 단기간 필요한 실험을 실행하고 프로토타입을 구축합니다.

예약 시작 시간을 향후 최대 8주로 설정하여 용량 블록을 예약할 수 있습니다. 각 용량 블록에는 최대 64개의 인스턴스가 있을 수 있으며, 용량 블록 전체에 최대 256개의 인스턴스가 있을 수 있습니다.

용량 블록을 사용하여 `p6-b200`, `p5`, `p5e`, `p5en`, `p4d`, `p4de`, `trn1`, `trn2` 인스턴스를 예약할 수 있습니다. 최대 182일의 예약 기간을 지정할 수 있습니다.

용량 블록을 예약하려면 필요한 인스턴스 유형, 인스턴스 수, 시간, 가장 이른 시작 날짜, 가장 늦은 종료 날짜를 포함하여 필요한 용량부터 지정합니다. 그러면 사양에 알맞게 제공되는 사용 가능한 용량 블록을 확인할 수 있습니다. 용량 블록 상품에는 시작 시간, 가용 영역, 예약 가격과 같은 세부 정보가 포함됩니다. 용량 블록 상품의 가격은 상품이 제공되는 당시에 사용 가능한 공급과 수요에 따라 다릅니다. 용량 블록 예약 후에는 가격이 변경되지 않습니다. 자세한 내용은 용량 블록 요금 및 결제 섹션을 참조하세요.

용량 블록 상품을 구매하면 선택한 날짜 및 인스턴스 수에 대한 예약이 생성됩니다. 용량 블록 예약이 시작되면 시작 요청에 예약 ID를 지정하여 인스턴스 시작을 대상으로 지정할 수 있습니다.

예약한 모든 인스턴스는 용량 블록 종료 시간 30분 전까지 사용할 수 있습니다. 용량 블록 예약이 30분 남았을 때, 용량 블록에서 실행 중인 모든 인스턴스가 종료되기 시작합니다. 30분은 다음 고객에게 용량 블록을 제공하기 전에 인스턴스를 정리하는 데 사용됩니다. 종료 프로세스가 시작되기 10분 전에 EventBridge를 통해 이벤트가 발생합니다. 자세한 내용은 EventBridge를 사용한 용량 블록 모니터링 섹션을 참조하세요.

주제

- 지원하는 플랫폼
- 고려 사항
- 관련 리소스
- 용량 블록 요금 및 결제
- 용량 블록의 사전 조건
- 용량 블록 찾기 및 구매
- 용량 블록을 사용하여 인스턴스 시작
- 용량 블록 보기
- 용량 블록 연장
- EventBridge를 사용한 용량 블록 모니터링
- AWS CloudTrail로 용량 블록 API 직접 호출 로깅

지원하는 플랫폼

ML용 용량 블록에서는 현재 기본 테넌시가 있는 `p6-b200.48xlarge`, `p5.48xlarge`, `p5e.48xlarge`, `p5en.48xlarge`, `p4d.24xlarge`, `p4de.24xlarge`, `trn1.32xlarge`, `trn2.48xlarge` 인스턴스를 지원합니다. AWS Management Console을 사용하여 용량 블록을 구매하는 경우 기본 플랫폼 옵션은 Linux/UNIX입니다. AWS Command Line Interface(AWS CLI) 또는 AWS SDK를 사용하여 용량 블록을 구매하는 경우 다음과 같은 플랫폼 옵션을 사용할 수 있습니다.

- Linux/Unix
- Red Hat Enterprise Linux
- HA가 설치된 RHEL
- SUSE Linux
- Ubuntu Pro

고려 사항

용량 블록을 사용하기 전에 다음 세부 정보와 제한 사항을 고려하세요.

- 각 용량 블록에는 최대 64개의 인스턴스가 있을 수 있으며, 용량 블록 전체에 최대 256개의 인스턴스가 있을 수 있습니다.
- 30분 이내에 시작할 수 있는 용량 블록 상품을 설명할 수 있습니다.
- 용량 블록은 협정 세계시(UTC) 오전 11:30에 종료됩니다.
- 용량 블록에서 실행 중인 인스턴스의 종료 프로세스는 예약 마지막 날의 협정 세계시(UTC) 오전 11:00에 시작됩니다.
- 용량 블록은 시작 시간을 기준으로 향후 8주까지 예약할 수 있습니다.
- 용량 블록 취소는 허용되지 않습니다.
- 용량 블록은 이동하거나 분할할 수 없습니다.
- 용량 블록은 AWS 계정 간 또는 AWS 조직 내에서 공유할 수 없습니다.
- 용량 블록은 용량 예약 그룹에서 사용할 수 없습니다.
- AWS 조직의 모든 계정에서 용량 블록에 예약할 수 있는 특정 날짜의 총 인스턴스 수는 64개를 초과할 수 없습니다.
- 용량 블록을 사용하려면 인스턴스에서 구체적인 예약 ID를 대상으로 지정해야 합니다.
- 용량 블록의 인스턴스는 온디맨드 인스턴스 한도 계산에 포함되지 않습니다.
- 사용자 지정 AMI를 사용하는 P5 인스턴스의 경우 EFA에 필요한 소프트웨어 및 구성이 있는지 확인합니다.
- Amazon EKS 관리형 노드 그룹의 경우 Amazon EC2 Capacity Blocks for ML이 포함된 관리형 노드 그룹 생성을 참조하세요. Amazon EKS 자체 관리형 노드 그룹의 경우 자체 관리형 노드가 포함된 ML용 용량 블록 사용을 참조하세요.
- 중지된 온디맨드 Amazon EC2 인스턴스를 Amazon EC2용 용량 블록으로 이동할 수 없으며, 인스턴스를 용량 블록에서 온디맨드로 이동할 수 없습니다.

관련 리소스

용량 블록 생성 후 용량 블록을 사용하여 다음 작업을 수행할 수 있습니다.

- 용량 블록으로 인스턴스 시작 자세한 내용은 용량 블록을 사용하여 인스턴스 시작 섹션을 참조하세요.
- Amazon EC2 Auto Scaling 그룹을 생성합니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Use Capacity Blocks for machine learning workloads를 참조하세요.

> **Note**
>
> Amazon EC2 Auto Scaling 또는 Amazon EKS를 사용하는 경우 용량 블록 예약을 시작할 때 규모 조정을 실행하도록 예약할 수 있습니다. 예약 규모 조정 기능을 통해 AWS에서는 자동으로 재시도를 처리하므로 일시적 실패를 처리하기 위한 재시도 로직 구현을 걱정하지 않아도 됩니다.

- AWS ParallelCluster를 통해 ML 워크플로를 개선합니다. 자세한 내용은 Enhancing ML workflows with AWS ParallelCluster and Amazon EC2 Capacity Blocks for ML을 참조하세요.

AWS ParallelCluster에 대한 자세한 정보는 AWS ParallelCluster란 무엇입니까?를 참조하세요.

용량 블록 요금 및 결제

Amazon EC2 Capacity Blocks for ML을 사용하면 예약한 만큼만 비용을 지불합니다. 용량 블록 가격은 구매 당시 용량 블록에 사용 가능한 공급 및 수요에 따라 다릅니다. 용량 블록 상품을 예약하기 전에 가격을 볼 수 있습니다. 용량 블록 가격은 예약 시 선불로 청구됩니다. 다양한 날짜의 용량 블록을 검색하면 사용 가능한 가장 저렴한 용량 블록 상품이 표시됩니다. 용량 블록을 예약한 후에는 가격이 변경되지 않습니다.

용량 블록을 사용하는 경우 인스턴스 실행 시 사용하는 운영 체제의 비용을 지불해야 합니다. 운영 체제 가격에 대한 자세한 내용은 Amazon EC2 Capacity Blocks for ML Pricing을 참조하세요.

결제

용량 블록 상품 가격은 선불로 청구됩니다. 용량 블록 구매 후 5분~12시간 내에 AWS 계정에 요금이 청구됩니다. 결제가 처리되는 동안에는 용량 블록 예약 리소스가 `payment-pending` 상태로 유지됩니다. 블록 시작 시간 최소 5분 전 또는 12시간 이내(둘 중 먼저 도래하는 시점)에 결제를 처리할 수 없는 경우 용량 블록이 해제되고 예약 상태가 `payment-failed`로 변경됩니다.

결제가 처리되면 용량 블록 리소스 상태가 `payment-pending`에서 `scheduled`로 변경됩니다. 일회성 선결제 금액이 반영된 인보이스가 수신됩니다. 인보이스를 통해서 용량 블록 예약 ID와 결제한 금액을 연결할 수 있습니다.

용량 블록 예약이 시작되면 예약에서 인스턴스가 실행되는 동안 사용하는 운영 체제만을 기준으로 요금이 청구됩니다. AWS Cost and Usage Report에서 월 사용량에 대한 연간 청구서의 사용량 및 관련 요금을 확인할 수 있습니다.

> **Note**
>
> 절감형 플랜과 예약 인스턴스 할인은 용량 블록에 적용되지 않습니다.

청구서 보기

AWS 결제 및 비용 관리 콘솔에서 청구서를 볼 수 있습니다. 용량 블록 선결제 금액은 예약을 구매한 달에 표시됩니다.

예약이 시작되면 청구서의 별도 줄에 블록 예약 사용 및 미사용 시간이 표시됩니다. 이러한 항목을 사용하여 예약에서 사용된 시간을 확인할 수 있습니다. 프리미엄 운영 체제를 사용하는 경우 사용한 시간에 대한 사용 요금만 행에 표시됩니다. 자세한 내용은 용량 블록 요금 및 결제 섹션을 참조하세요. 미사용 시간에 대한 추가 요금은 없습니다.

자세한 내용은 AWS 결제 및 비용 관리 사용 설명서에서 결제 보기를 참조하세요.

예약을 구매한 달과 다른 달에 용량 블록이 시작되면 선결제 금액과 예약 사용량이 별도의 청구 월 아래에 표시됩니다. AWS Cost and Usage Report에서는 사용량을 해당 선결제 가격과 연결할 수 있도록 용량 블록 예약 ID는 선결제 요금의 reservation/ReservationARN 항목에 나열되고 lineitem/ResourceID는 연간 청구서에 나열됩니다.

용량 블록의 사전 조건

사용하려는 인스턴스 유형을 지원하는 AWS 리전에서 인스턴스를 시작해야 합니다.

`p6-b200.48xlarge` 인스턴스가 포함된 용량 블록은 다음 AWS 리전에서 사용할 수 있습니다.

리전 코드	리전 이름
us-west-2	미국 서부(오리건)
us-east-2	미국 동부(오하이오)

p5.48xlarge 인스턴스가 포함된 용량 블록은 다음 AWS 리전에서 사용할 수 있습니다.

리전 코드	리전 이름
us-east-1	미국 동부(버지니아 북부)
us-east-2	미국 동부(오하이오)
us-west-1	미국 서부(캘리포니아 북부)
us-west-2	미국 서부(오리건)
eu-north-1	유럽(스톡홀름)
eu-west-2	유럽(런던)
sa-east-1	남아메리카(상파울루)
ap-northeast-1	아시아 태평양(도쿄)
ap-south-1	아시아 태평양(뭄바이)
ap-southeast-2	아시아 태평양(시드니)
ap-southeast-3	아시아 태평양(자카르타)

p5e.48xlarge 인스턴스가 포함된 용량 블록은 다음 AWS 리전에서 사용할 수 있습니다.

리전 코드	리전 이름
us-east-1	미국 동부(버지니아 북부)
us-east-2	미국 동부(오하이오)
us-west-1	미국 서부(캘리포니아 북부)
us-west-2	미국 서부(오리건)
eu-north-1	유럽(스톡홀름)

리전 코드	리전 이름
eu-west-2	유럽(런던)
sa-east-1	남아메리카(상파울루)
ap-northeast-1	아시아 태평양(도쿄)
ap-south-1	아시아 태평양(뭄바이)
ap-southeast-2	아시아 태평양(시드니)
ap-southeast-3	아시아 태평양(자카르타)

`p5en.48xlarge` 인스턴스가 포함된 용량 블록은 다음 AWS 리전에서 사용할 수 있습니다.

리전 코드	리전 이름
us-east-1	미국 동부(버지니아 북부)
us-east-2	미국 동부(오하이오)
us-west-1	미국 서부(캘리포니아 북부)
us-west-2	미국 서부(오리건)
eu-north-1	유럽(스톡홀름)
eu-west-2	유럽(런던)
eu-south-2	유럽(스페인)
sa-east-1	남아메리카(상파울루)
ap-northeast-1	아시아 태평양(도쿄)
ap-northeast-2	아시아 태평양(서울)
ap-south-1	아시아 태평양(뭄바이)

리전 코드	리전 이름
ap-southeast-3	아시아 태평양(자카르타)

`p4d.24xlarge` 인스턴스가 포함된 용량 블록은 다음 AWS 리전에서 사용할 수 있습니다.

리전 코드	리전 이름
us-east-1	미국 동부(버지니아 북부)
us-east-2	미국 동부(오하이오)
us-west-2	미국 서부(오리건)

`p4de.24xlarge` 인스턴스가 포함된 용량 블록은 다음 AWS 리전에서 사용할 수 있습니다.

리전 코드	리전 이름
us-east-1	미국 동부(버지니아 북부)
us-west-2	미국 서부(오리건)

`trn1.32xlarge` 인스턴스가 포함된 용량 블록은 다음 AWS 리전에서 사용할 수 있습니다.

리전 코드	리전 이름
us-east-1	미국 동부(버지니아 북부)
us-east-2	미국 동부(오하이오)
us-west-1	미국 서부(캘리포니아 북부)
us-west-2	미국 서부(오리건)
eu-north-1	유럽(스톡홀름)
ap-south-1	아시아 태평양(뭄바이)

리전 코드	리전 이름
ap-southeast-2	아시아 태평양(시드니)
ap-southeast-4	아시아 태평양(멜버른)

`trn2.48xlarge` 인스턴스가 포함된 용량 블록은 다음 AWS 리전에서 사용할 수 있습니다.

리전 코드	리전 이름
us-east-2	미국 동부(오하이오)

> **Note**
>
> 64개 인스턴스의 용량 블록 크기는 모든 AWS 리전의 모든 인스턴스 유형에 대해 지원되지 않습니다.

용량 블록 찾기 및 구매

용량 블록을 예약하려면 먼저 필요성과 일치하는 용량을 사용할 수 있는 시간 블록을 찾아야 합니다. 예약할 수 있는 용량 블록을 찾으려면 다음과 같은 사항을 지정합니다.

- 필요한 인스턴스 수
- 인스턴스가 필요한 기간
- 예약이 필요한 날짜 범위

사용 가능한 용량 블록 상품을 검색하려면 예약 기간과 인스턴스 수를 지정합니다. 1일 단위로 최대 14일 및 7일 단위로 182일까지 예약 기간을 지정해야 합니다. 각 용량 블록에는 최대 64개의 인스턴스가 있을 수 있으며, 용량 블록 전체에 최대 256개의 인스턴스가 있을 수 있습니다.

사양과 일치하는 용량 블록을 요청하면 최대 3개의 사용 가능한 블록의 세부 정보가 제공됩니다. 모든 용량 블록은 오전 11시 30분 UTC에 종료되므로 같은 날 시작하는 블록의 기간은 원하는 기간과 가장 가깝게 일치합니다. 어떤 블록의 기간은 원하는 기간보다 약간 짧고, 어떤 블록의 기간은 원하는 기간보다 약간 깁니다.

상품 세부 정보에는 예약 시작 시간, 예약 가용 영역 및 예약 가격이 포함됩니다. 자세한 내용은 용량 블록 요금 및 결제 섹션을 참조하세요.

표시된 용량 블록 상품을 구매하거나 검색 기준을 수정하여 사용 가능한 다른 옵션을 확인할 수 있습니다. 상품에 미리 정의된 만료 시간이 없지만 선착순으로만 상품이 제공됩니다.

용량 블록 상품을 구매하면 용량 블록이 예약되었다는 것이 확인되는 응답이 즉시 표시됩니다. 확인 후에는 예약 유형이 `capacity-block`이며 `start-date`가 구매한 상품의 시작 시간으로 설정된 새 용량 예약이 계정에 표시됩니다. 용량 블록 예약은 `payment-pending` 상태로 생성됩니다. 선결제 금액이 처리되면 예약 상태가 `scheduled`로 변경됩니다. 자세한 내용은 결제 섹션을 참조하세요.

Console

용량 블록을 찾고 구매하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 화면 상단의 탐색 모음에서 AWS 리전을 선택합니다. 이 선택은 64개 인스턴스의 용량 블록 크기는 모든 리전의 모든 인스턴스 유형에 대해 지원되지 않기 때문에 중요합니다.
3. 탐색 창에서 용량 예약, 용량 블록 구매를 선택합니다.
4. 용량 속성에서 용량 블록 검색 파라미터를 정의할 수 있습니다. 기본적으로 플랫폼은 Linux입니다. 다른 운영 체제를 선택하려면 AWS CLI를 사용합니다. 자세한 내용은 지원하는 플랫폼 섹션을 참조하세요.
5. 총 용량에서 예약하려는 인스턴스 수를 선택합니다.
6. 기간에서 예약이 필요한 일수 또는 주 수를 입력합니다.
7. 용량 블록을 검색할 날짜 범위에서 예약을 시작할 가장 빠른 날짜를 입력합니다.
8. 용량 블록 찾기를 선택합니다.
9. 사양을 충족하는 용량 블록을 사용할 수 있으면 권장 용량 블록 아래에 상품이 표시됩니다. 사양을 충족하는 상품이 여러 가지라면 가장 빠른 용량 블록 상품이 표시됩니다. 다른 용량 블록 상품을 보려면 검색 입력 내용을 조정하고 용량 블록 찾기를 다시 선택합니다.
10. 구매하려는 용량 블록 상품을 찾으면 다음을 선택합니다.
11. (선택 사항) 태그 추가 페이지에서 새 태그 추가를 선택합니다.
12. 검토 및 구매 페이지에 시작 및 종료 날짜, 기간, 총 인스턴스 수, 가격이 나열됩니다.

> **Note**
>
> 용량 블록을 예약한 후에는 취소할 수 없습니다.

13. 용량 블록 구매 팝업 창에서 confirm(확인)이라고 입력한 다음에 구매를 선택합니다.

AWS CLI

용량 블록을 찾는 방법

`describe-capacity-block-offerings` 명령을 사용합니다.

```
aws ec2 describe-capacity-block-offerings \
    --instance-type p5.48xlarge \
    --instance-count 16 \
    --start-date-range 2023-08-14T00:00:00Z \
    --end-date-range 2023-10-22-T00:00:00Z \
    --capacity-duration 48
```

용량 블록을 구매하려면

이전 예제의 출력에서 얻은 용량 블록의 상품 ID와 함께 `purchase-capacity-block` 명령을 사용합니다.

```
aws ec2 purchase-capacity-block \
  --capacity-block-offering-id cbr-0123456789abcdefg \
  --instance-platform Linux/UNIX
```

PowerShell

용량 블록을 찾는 방법

Get-EC2CapacityBlockOffering cmdlet을 사용합니다.

```
Get-EC2CapacityBlockOffering `
    -InstanceType `
    -CapacityDurationHour 48 `
    -InstanceCount 16
```

용량 블록을 구매하려면

이전 예제의 출력에서 얻은 용량 블록의 상품 ID와 함께 New-EC2EC2CapacityBlock cmdlet을 사용합니다.

```
New-EC2EC2CapacityBlock `
    -CapacityBlockOfferingId cbr-0123456789abcdefg `
    -InstancePlatform Linux/UNIX
```

용량 블록을 사용하여 인스턴스 시작

용량 블록을 사용하려면 인스턴스를 시작할 때 용량 블록 예약 ID를 지정해야 합니다. 인스턴스를 용량 예약으로 내보내면 시작된 인스턴스 수만큼 사용 가능한 용량이 감소합니다. 예를 들어, 구매한 인스턴스 용량이 8개 인스턴스이고 4개 인스턴스를 시작하면 사용 가용한 용량이 4만큼 감소합니다.

예약이 종료되기 전에 용량 블록에서 실행 중인 인스턴스를 종료하면 그 자리에서 새 인스턴스를 시작할 수 있습니다. 용량 블록에서 인스턴스를 중지하거나 종료하는 경우 다른 인스턴스를 시작하여 바꿀 수 있으려면 인스턴스를 정리하는 데 몇 분 정도 걸립니다. 이 시간에는 인스턴스가 중지 또는 `shutting-down` 상태로 됩니다. 이 프로세스가 완료되면 인스턴스 상태가 `stopped` 또는 `terminated`로 변경됩니다. 그러면 용량 블록의 사용 가능한 용량이 업데이트되어 사용 가능한 다른 인스턴스가 표시됩니다.

요구 사항

- 용량 블록이 있는 가용 영역과 다른 가용 영역의 서브넷에서 인스턴스를 시작할 수 없습니다.
- 용량 블록의 플랫폼과 다른 플랫폼의 AMI를 사용하여 인스턴스를 시작할 수 없습니다.

Console

용량 블록을 사용하여 인스턴스를 시작하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 화면 상단의 탐색 모음에서 용량 블록 예약의 리전을 선택합니다.
3. Amazon EC2 콘솔 대시보드에서 인스턴스 시작을 선택합니다.
4. 인스턴스 시작 절차를 따릅니다.
5. 고급 세부 정보를 확장하고 인스턴스를 다음과 같이 구성합니다.
   a. 구매 옵션에서 용량 블록을 선택합니다.
   b. 용량 예약에서 용량 예약 지정을 선택합니다.
   c. 용량 예약 대상 ID에서 용량 블록 예약의 ID를 선택합니다.
6. 인스턴스 시작을 선택합니다.

AWS CLI

용량 블록을 사용하여 인스턴스를 시작하려면

run-instances 명령을 `MarketType` 옵션과 함께 사용합니다.

```
--instance-market-options MarketType='capacity-block'
```

`capacity-reservation-specification` 옵션도 지정해야 합니다.

```
--capacity-reservation-specification
 CapacityReservationTarget={CapacityReservationId=cr-0abc123def456789a}
```

PowerShell

용량 블록을 사용하여 인스턴스를 시작하려면

`-InstanceMarketOption` 파라미터와 함께 New-EC2Instance cmdlet을 사용합니다.

```
-InstanceMarketOption $marketOptions
```

다음과 같이 시장 옵션을 정의합니다.

```
$marketOptions = New-Object Amazon.EC2.Model.InstanceMarketOptionsRequest
$marketOptions.MarketType = "capacity-block"
```

`-CapacityReservationTarget_CapacityReservationId` 옵션도 지정해야 합니다.

```
-CapacityReservationTarget_CapacityReservationId cr-0abc123def456789a
```

관련 리소스

- 용량 블록을 대상으로 하는 시작 템플릿을 생성하려면 the section called “시작 템플릿” 섹션을 참조하세요.
- EC2 플릿을 사용하여 인스턴스를 용량 블록으로 시작하려면 튜토리얼: 인스턴스를 용량 블록으로 시작하도록 EC2 플릿을 구성 섹션을 참조하세요.
- 용량 블록이 포함된 EKS 관리형 노드 그룹을 설정하려면 Amazon EKS 사용 설명서의 ML용 용량 블록이 포함된 관리형 노드 그룹 생성을 참조하세요.

- 용량 블록을 사용하여 AWS ParallelCluster를 설정하려면 AWS ParallelCluster 기반 ML을 참조하세요.

용량 블록 보기

용량 블록 예약 후 AWS 계정에서 용량 블록 예약을 볼 수 있습니다. `start-date`와 `end-date`를 보며 언제 예약이 시작하고 종료되는지 확인할 수 있습니다. 용량 블록 예약이 시작되기 전에는 사용 가능한 용량이 0으로 표시됩니다. 태그 키 `aws:ec2capacityreservation:incrementalRequestedQuantity`의 태그 값을 통해 용량 블록에서 사용할 수 있는 인스턴스 수를 확인할 수 있습니다.

용량 블록 예약이 시작되면 예약 상태가 `scheduled`에서 `active`로 변경됩니다. Amazon EventBridge를 통해 이벤트가 발생하여 용량 블록을 사용할 수 있다고 알려줍니다. 자세한 내용은 EventBridge를 사용한 용량 블록 모니터링 섹션을 참조하세요.

용량 블록에는 다음과 같은 상태가 있습니다.

- `payment-pending` – 선결제 금액이 아직 처리되지 않았습니다.
- `payment-failed` – 12시간 기간 안에 결제를 처리하지 못했습니다. 용량 블록이 해제되었습니다.
- `scheduled` – 결제는 처리되었고 용량 블록 예약은 아직 시작되지 않았습니다.
- `active` – 예약 용량을 사용할 수 없습니다.
- `expired` - 용량 블록 예약이 예약 요청 시 지정한 날짜 및 시간에 자동으로 만료되었습니다. 예약 용량을 더 이상 사용할 수 없습니다.

Console

용량 블록을 보는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 용량 예약를 선택합니다.
3. 용량 예약 개요 페이지에서 모든 용량 예약 리소스에 대한 세부 정보가 있는 리소스 테이블을 참조할 수 있습니다. 용량 블록 예약을 찾으려면 용량 예약 ID 위의 드롭다운 목록에서 용량 블록을 선택합니다. 테이블에서 시작 및 종료 날짜, 기간, 상태와 같은 용량 블록에 대한 정보를 확인할 수 있습니다.
4. 용량 블록에 대해 자세히 알아보려면 보려는 용량 블록의 예약 ID를 선택합니다. 용량 예약 세부 정보 페이지에 예약의 모든 속성과 용량 블록에서 사용 중이며 사용 가능한 인스턴스 수가 표시됩니다.

> **Note**
>
> 용량 블록 예약이 시작되기 전에는 사용 가능한 용량이 0으로 표시됩니다. 태그 키 `aws:ec2capacityreservation:incrementalRequestedQuantity`의 다음 태그 값을 사용하여 용량 블록 예약이 시작될 때 사용할 수 있는 인스턴스 수를 확인할 수 있습니다.

AWS CLI

용량 블록을 보는 방법

기본적으로 describe-capacity-reservations 명령을 사용하면 온디맨드 용량 예약과 용량 블록 예약이 모두 나열됩니다. 용량 블록 예약만 보려면 `capacity-block` 유형의 예약을 필터링합니다.

```
aws ec2 describe-capacity-reservations \
    --filters Name=reservation-type,Values=capacity-block
```

PowerShell

용량 블록을 보는 방법

Get-EC2CapacityReservation cmdlet을 사용합니다. 기본적으로 온디맨드 용량 예약과 용량 블록 예약이 모두 나열됩니다. 용량 블록 예약만 보려면 `capacity-block` 유형의 예약을 필터링합니다.

```
Get-EC2CapacityReservation `
    -Filter @{Name="reservation-type"; Values="capacity-block"}
```

용량 블록 연장

용량 블록을 사용하면 워크로드를 위한 컴퓨팅 용량을 예약하여 가용성과 일관성을 보장할 수 있습니다. 변화하는 요구 사항을 수용하기 위해 필요에 따라 기존 용량 블록의 기간을 연장할 수 있습니다.

용량 블록을 연장하려면 용량 블록이 `active` 또는 `scheduled` 상태여야 하며, `payment-pending`인 연장이 없어야 합니다. 용량 블록 기간이 만료되기 전에 최소 1시간 또는 최대 56일까지 연장을 요청할 수 있습니다. 용량 블록을 1일 증분 단위로 최대 총 14일, 7일 증분 단위로 최대 총 182일(26주)만큼 연장할 수 있습니다. 용량 블록을 연장하면 종료 날짜가 업데이트되어 인스턴스가 중단 없이 계속 실행될 수 있습니다.

- 용량 블록에 적용할 수 있는 연장 횟수에는 제한이 없습니다.
- 블록을 연장한 후에도 용량 예약 ID는 동일하게 유지됩니다.
- 용량 블록을 지원할 수 있는 충분한 용량이 있는 경우에만 용량 블록을 연장할 수 있으며, 연장은 보장되지 않습니다.

결제

용량 블록 상품 가격은 선불로 청구됩니다. 청구서가 지불될 때까지 연장은 `payment-pending` 상태를 유지합니다. 12시간 내 또는 용량 블록이 종료되기 최대 35분 전(둘 중 먼저 도래하는 시점)에 결제를 처리할 수 없는 경우 연장이 성공하지 못하고 상태가 `payment-failed`로 변경됩니다. 용량 블록 예약은 `active`로 유지되며 원래 종료 날짜에 종료됩니다.

결제가 성공적으로 처리되면 용량 블록 연장 상태가 `payment-succeeded`로 변경되고 용량 블록 예약의 종료 날짜가 새 종료 날짜로 업데이트됩니다. 연장의 세부 정보는 콘솔의 용량 블록 연장 세부 정보 섹션에서 또는 describe-capacity-block-extension-history 명령을 사용하여 볼 수 있습니다.

용량 블록 연장

Console

용량 블록을 연장하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 용량 예약를 선택합니다.
3. 용량 예약 개요 페이지에서 모든 용량 예약 리소스에 대한 세부 정보가 있는 리소스 테이블을 볼 수 있습니다. 연장하려는 용량 블록의 예약 ID를 선택합니다.
4. 작업 드롭다운 메뉴에서 용량 블록 연장을 선택합니다.
5. 기간에서 예약을 연장해야 하는 일수 또는 주 수를 입력합니다.
6. 용량 블록 찾기를 선택합니다.
7. 사양을 충족하는 용량 블록을 사용할 수 있으면 권장 용량 블록 아래에 상품이 표시됩니다. 다른 용량 블록 상품을 보려면 검색 입력 내용을 조정하고 용량 블록 찾기를 다시 선택합니다.
8. 구매하려는 용량 블록 상품을 찾으면 연장을 선택합니다.
9. 용량 블록 연장 팝업 창에서 confirm을 입력한 다음 연장을 선택합니다.

AWS CLI

용량 블록 연장을 찾는 방법

describe-capacity-block-extension-offerings 명령을 사용합니다. 다음 예제는 지정된 예약에 대한 48시간 용량 블록 연장을 검색합니다.

```
aws ec2 describe-capacity-block-extension-offerings \
    --capacity-reservation-id cr-1234567890abcdefg \
    --capacity-block-extension-duration-hours 48
```

용량 블록을 연장하려면

purchase-capacity-block-extension 명령을 사용합니다. 이전 예제의 출력에서 얻은 확장 상품 ID를 지정합니다.

```
aws ec2 purchase-capacity-block-extension \
    --capacity-block-extension-offering-id cbe-0123456789abcdefg \
    --capacity-reservation-id cr-1234567890abcdefg
```

PowerShell

용량 블록 연장을 찾는 방법

Get-EC2CapacityBlockExtensionOffering cmdlet을 사용합니다. 다음 예제는 지정된 예약에 대한 48시간 용량 블록 연장을 검색합니다.

```
Get-EC2CapacityBlockExtensionOffering `
    -CapacityReservationId cr-1234567890abcdefg `
    -CapacityBlockExtensionDurationHour 48
```

용량 블록을 연장하려면

Invoke-EC2CapacityBlockExtension cmdlet을 사용합니다. 이전 예제의 출력에서 얻은 확장 상품 ID를 지정합니다.

```
Invoke-EC2CapacityBlockExtension `
    -CapacityBlockExtensionOfferingId cbe-0123456789abcdefg `
    -CapacityReservationId cr-1234567890abcdefg
```

EventBridge를 사용한 용량 블록 모니터링

용량 블록 예약이 시작되면 용량을 사용할 준비가 되었다는 이벤트가 EventBridge를 통해 Amazon EC2에서 발생합니다. 용량 블록 예약 종료 40분 전에 예약에서 실행 중인 모든 인스턴스가 10분 후에

종료되기 시작한다는 다른 EventBridge 이벤트가 수신됩니다. EventBridge 이벤트에 자세한 내용은 Amazon EventBridge Events를 참조하세요.

용량 블록에 발생하는 이벤트의 이벤트 구조는 다음과 같습니다.

용량 블록 전송됨

다음 예에서는 용량 블록 전송됨의 이벤트를 보여줍니다.

```
{
  "customer_event_id": "[Capacity Reservation Id]-delivered",
  "detail_type": "Capacity Block Reservation Delivered",
  "source": "aws.ec2",
  "account": "[Customer Account ID]",
  "time": "[Current time]",
  "resources": [
    "[ODCR ARN]"
  ],
  "detail": {
    "capacity-reservation-id": "[ODCR ID]",
    "end-date": "[ODCR End Date]"
  }
}
```

용량 블록 만료 경고

다음 예에서는 용량 블록 만료 경고의 이벤트를 보여줍니다.

```
{
  "customer_event_id": "[Capacity Reservation Id]-approaching-expiry",
  "detail_type": "Capacity Block Reservation Expiration Warning",
  "source": "aws.ec2",
  "account": "[Customer Account ID]",
  "time": "[Current time]",
  "resources": [
    "[ODCR ARN]"
  ],
  "detail": {
    "capacity-reservation-id": "[ODCR ID]",
    "end-date": "[ODCR End Date]"
  }
}
```

AWS CloudTrail로 용량 블록 API 직접 호출 로깅

용량 블록은 용량 블록에서 사용자, 역할 또는 서비스를 통해 수행된 작업의 레코드를 제공하는 AWS 서비스인 AWS CloudTrail과 통합되어 있습니다. CloudTrail에서는 용량 블록에 대한 API 직접 호출을 이벤트로 캡처합니다. 캡처되는 직접 호출에는 용량 블록 콘솔로부터의 호출과 용량 블록 API 작업에 대한 코드 직접 호출이 포함됩니다. 추적을 생성하면 용량 블록 이벤트를 포함한 CloudTrail 이벤트를 지속적으로 Amazon S3 버킷에 전송할 수 있습니다. 트레일을 구성하지 않은 경우에도 CloudTrail 콘솔의 이벤트 기록에서 최신 이벤트를 볼 수 있습니다. CloudTrail에서 수집된 정보를 사용하여 용량 블록에 적용된 요청, 요청이 적용된 IP 주소, 요청을 적용한 사람, 요청이 수행된 시간 및 추가 세부 정보를 결정할 수 있습니다.

CloudTrail에 대한 자세한 내용은 AWS CloudTrail 사용 설명서를 참조하세요.

CloudTrail의 용량 블록 정보

CloudTrail은 계정 생성 시 AWS 계정에서 사용되도록 설정됩니다. 용량 블록에서 활동이 발생하면 해당 활동이 이벤트 기록의 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. AWS 계정에서 최신 이벤트를 확인, 검색 및 다운로드할 수 있습니다. 자세한 내용은 CloudTrail 이벤트 기록을 사용하여 이벤트 보기를 참조하세요.

용량 블록 이벤트를 포함한 이벤트를 AWS 계정에 지속적으로 기록하려면 추적을 생성합니다. CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 콘솔에서 추적을 생성하면 기본적으로 모든 AWS 리전에 추적이 적용됩니다. 추적은 AWS 파티션에 있는 모든 리전의 이벤트를 로깅하고 지정된 Amazon S3 버킷으로 로그 파일을 전송합니다. 추가적으로, CloudTrail 로그에서 수집된 이벤트 데이터를 추가 분석 및 처리하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 다음 자료를 참조하세요.

- 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 용량 블록 작업은 CloudTrail을 통해 로깅되고 Amazon EC2 API 참조로 문서화됩니다. 예를 들어, `CapacityBlockScheduled` 직접 호출 및 `CapacityBlockActive` 작업을 통해 CloudTrail 로그 파일에 항목이 생성됩니다.

모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에 대한 정보가 포함됩니다. ID 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청을 루트로 했는지 아니면 AWS Identity and Access Management(IAM) 사용자 보안 인증으로 했는지 여부입니다.
- 역할 또는 페더레이션 사용자에 대한 임시 보안 인증을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청했는지.

자세한 내용은 CloudTrail userIdentity 요소를 참조하세요.

용량 블록 로그 파일 이해

추적이란 지정한 Amazon S3 버킷에 이벤트를 로그 파일로 입력할 수 있게 하는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함될 수 있습니다. 이벤트는 모든 소스로부터의 단일 요청을 나타내며 요청 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보가 들어 있습니다. CloudTrail 로그 파일은 퍼블릭 API 직접 호출의 주문 스택 트레이스가 아니므로 특정 순서로 표시되지 않습니다.

다음 예에서는 다음에 대한 CloudTrail 로그 항목을 보여줍니다.

- TerminateCapacityBlocksInstances
- CapacityBlockPaymentFailed
- CapacityBlockScheduled
- CapacityBlockActive
- CapacityBlockFailed
- CapacityBlockExpired

> **Note**
>
> 데이터 프라이버시의 예에서 일부 필드가 수정되었습니다.

TerminateCapacityBlocksInstances

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "accountId": "123456789012",
    "invokedBy": "AWS Internal;"
  },
```

```
  "eventTime": "2023-10-02T00:06:08Z",
  "eventSource": "ec2.amazonaws.com",
  "eventName": "TerminateCapacityBlockInstances",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.25",
  "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a1b2c3d4-EXAMPLE",
  "readOnly": false,
  "resources": [
    {
      "accountId": "123456789012",
      "type": "AWS::EC2::Instance",
      "ARN": "arn:aws:ec2:US East (N. Virginia):123456789012:instance/
i-1234567890abcdef0"
    }
    {
      "accountId": "123456789012",
      "type": "AWS::EC2::Instance",
      "ARN": "arn:aws::ec2:US East (N. Virginia):123456789012:instance/
i-0598c7d356eba48d7"
    }
  ],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "123456789012",
  "serviceEventDetails": {
      "capacityReservationId": "cr-12345678",
      }
}
```

CapacityBlockPaymentFailed

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "accountId": "123456789012",
    "invokedBy": "AWS Internal;"
  },
  "eventTime": "2023-10-02T00:06:08Z",
  "eventSource": "ec2.amazonaws.com",
  "eventName": "CapacityBlockPaymentFailed",
  "awsRegion": "us-east-1",
```

```
  "sourceIPAddress": "203.0.113.25",
  "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a1b2c3d4-EXAMPLE",
  "readOnly": false,
  "resources": [
    {
      "ARN": "arn:aws:ec2:US East (N. Virginia):123456789012:capacity-reservation/
cr-12345678",
      "accountId": "123456789012",
      "type": "AWS::EC2::CapacityReservation"
    }
  ],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "123456789012",
  "serviceEventDetails": {
      "capacityReservationId": "cr-12345678",
      "capacityReservationState": "payment-failed"
      }
}
```


CapacityBlockScheduled


```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "accountId": "123456789012",
    "invokedBy": "AWS Internal;"
  },
  "eventTime": "2023-10-02T00:06:08Z",
  "eventSource": "ec2.amazonaws.com",
  "eventName": "CapacityBlockScheduled",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.25",
  "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a1b2c3d4-EXAMPLE",
  "readOnly": false,
  "resources": [
    {
```

```
      "ARN": "arn:aws:ec2:US East (N. Virginia):123456789012:capacity-reservation/
cr-12345678",
      "accountId": "123456789012",
      "type": "AWS::EC2::CapacityReservation"
    }
  ],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "123456789012",
  "serviceEventDetails": {
      "capacityReservationId": "cr-12345678",
      "capacityReservationState": "scheduled"
      }
}
```

CapacityBlockActive

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "accountId": "123456789012",
    "invokedBy": "AWS Internal;"
  },
  "eventTime": "2023-10-02T00:06:08Z",
  "eventSource": "ec2.amazonaws.com",
  "eventName": "CapacityBlockActive",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.25",
  "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a1b2c3d4-EXAMPLE",
  "readOnly": false,
  "resources": [
    {
      "ARN": "arn:aws:ec2:US East (N. Virginia):123456789012:capacity-reservation/
cr-12345678",
      "accountId": "123456789012",
      "type": "AWS::EC2::CapacityReservation"
    }
  ],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "123456789012",
  "serviceEventDetails": {
```

```
      "capacityReservationId": "cr-12345678",
      "capacityReservationState": "active"
      }
 }
```

CapacityBlockFailed

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "accountId": "123456789012",
    "invokedBy": "AWS Internal;"
  },
  "eventTime": "2023-10-02T00:06:08Z",
  "eventSource": "ec2.amazonaws.com",
  "eventName": "CapacityBlockFailed",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.25",
  "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a1b2c3d4-EXAMPLE",
  "readOnly": false,
  "resources": [
    {
      "ARN": "arn:aws:ec2:US East (N. Virginia):123456789012:capacity-reservation/
cr-12345678",
      "accountId": "123456789012",
      "type": "AWS::EC2::CapacityReservation"
    }
  ],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "123456789012",
  "serviceEventDetails": {
      "capacityReservationId": "cr-12345678",
      "capacityReservationState": "failed"
      }
 }
```

CapacityBlockExpired

```
{
  "eventVersion": "1.05",
```

```
  "userIdentity": {
    "accountId": "123456789012",
    "invokedBy": "AWS Internal;"
  },
  "eventTime": "2023-10-02T00:06:08Z",
  "eventSource": "ec2.amazonaws.com",
  "eventName": "CapacityBlockExpired",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.25",
  "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a1b2c3d4-EXAMPLE",
  "readOnly": false,
  "resources": [
    {
      "ARN": "arn:aws:ec2:US East (N. Virginia):123456789012:capacity-reservation/
cr-12345678",
      "accountId": "123456789012",
      "type": "AWS::EC2::CapacityReservation"
    }
  ],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "123456789012",
  "serviceEventDetails": {
      "capacityReservationId": "cr-12345678",
      "capacityReservationState": "expired"
      }
 }
```

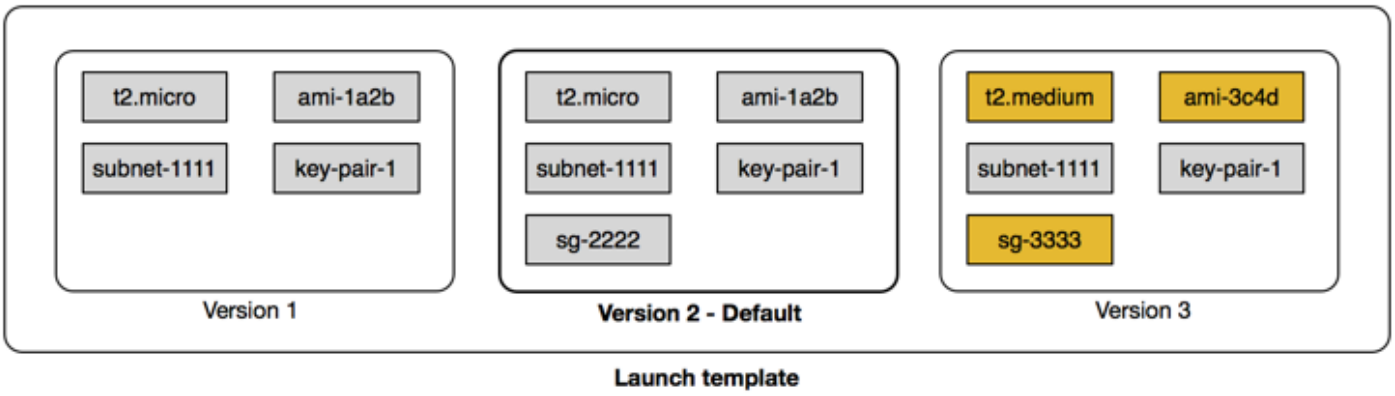

Amazon EC2 시작 템플릿에 인스턴스 시작 파라미터 저장

Amazon EC2 인스턴스를 시작할 때마다 지정할 필요가 없도록 Amazon EC2 시작 템플릿을 사용하여 인스턴스 시작 파라미터를 저장할 수 있습니다. 예를 들어, 일반적으로 인스턴스를 시작하는 데 사용하는 AMI ID, 인스턴스 유형 및 네트워크 설정을 저장하는 시작 템플릿을 생성할 수 있습니다. 파라미터를 다시 입력하는 대신 Amazon EC2 콘솔, AWS SDK 또는 명령줄 도구를 사용하여 인스턴스를 시작할 때 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 사용

할 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

- Amazon EC2 시작 템플릿에 대한 제한 사항
- Amazon EC2 시작 템플릿에 필요한 IAM 권한
- Amazon EC2 시작 템플릿을 사용하여 Amazon EC2 인스턴스 시작 제어
- Amazon EC2 시작 템플릿 생성
- 시작 템플릿 수정(시작 템플릿 버전 관리)
- 시작 템플릿 또는 시작 템플릿 버전 삭제

Amazon EC2 시작 템플릿에 대한 제한 사항

다음 제한 사항은 시작 템플릿과 시작 템플릿 버전에 적용됩니다.

- 할당량 - 시작 템플릿과 시작 템플릿 버전의 할당량을 보려면 Service Quotas 할당량 콘솔을 열거나 list-service-quotas AWS CLI 명령을 사용합니다. 각 AWS 계정별로, 리전당 최대 5,000개의 시작 템플릿과 시작 템플릿당 최대 1만개의 버전을 생성할 수 있습니다. 계정의 경과 시간과 사용 내역에 따라 할당량이 다를 수 있습니다.

- 파라미터는 선택 사항 - 시작 템플릿 파라미터는 선택 사항입니다. 그러나 인스턴스 시작 요청에 필요한 모든 파라미터가 포함되도록 해야 합니다. 예를 들어, 시작 템플릿에 AMI ID가 포함되어 있지 않으면 이 시작 템플릿을 사용하여 인스턴스를 시작할 때 AMI ID를 지정해야 합니다.
- 파라미터는 검증되지 않음 – 시작 템플릿을 생성할 때 시작 템플릿 파라미터가 완전히 검증되지 않습니다. 잘못된 값을 지정하거나 지원되지 않는 파라미터 조합을 사용하면 이 시작 템플릿을 사용하여 인스턴스를 시작할 수 없습니다. 문제를 방지하려면 올바른 값을 지정하고 지원되는 파라미터 조합을 사용하는지 확인합니다. 예를 들어 배치 그룹에서 인스턴스를 시작하려면 지원되는 인스턴스 유형을 지정해야 합니다.
- 태그 - 시작 템플릿에 태그를 지정할 수 있지만, 시작 템플릿 버전에는 태그를 지정할 수 없습니다.
- 변경 불가능 - 시작 템플릿은 변경할 수 없습니다. 시작 템플릿을 수정하려면 시작 템플릿의 새 버전을 만들어야 합니다.
- 버전 번호 - 시작 템플릿 버전은 생성한 순서대로 번호가 지정됩니다. 시작 템플릿 버전을 생성할 때 버전 번호를 자체적으로 지정할 수 없습니다.

Amazon EC2 시작 템플릿에 필요한 IAM 권한

IAM 권한을 사용하여 사용자가 시작 템플릿이나 시작 템플릿 버전 나열, 보기, 생성 또는 삭제를 수행할 수 있는지 제어할 수 있습니다.

> **Important**
>
> 사용자가 시작 템플릿이나 시작 템플릿 버전을 생성할 때 시작 템플릿에서 지정할 수 있는 리소스를 제한하기 위해 리소스 수준의 권한을 사용할 수 없습니다. 따라서 신뢰할 수 있는 관리자에게만 시작 템플릿과 시작 템플릿 버전을 생성할 수 있는 권한을 부여해야 합니다.

시작 템플릿을 사용하는 모든 사용자에게 시작 템플릿에 지정된 리소스를 사용하고 생성할 수 있는 권한을 부여해야 합니다. 예시:

- 공유 프라이빗 Amazon Machine Image(AMI)에서 인스턴스를 시작하려면 사용자에게 AMI에 대한 시작 권한이 있어야 합니다.
- 기존 스냅샷의 태그를 사용하여 EBS 볼륨을 생성하려면 사용자에게 스냅샷에 대한 읽기 권한과 볼륨을 생성하고 태그를 지정할 수 있는 권한이 있어야 합니다.

내용

- ec2:CreateLaunchTemplate

- ec2:DescribeLaunchTemplates
- ec2:DescribeLaunchTemplateVersions
- ec2:DeleteLaunchTemplate
- 버전 관리 권한 제어
- 시작 템플릿의 태그에 대한 액세스 제어

ec2:CreateLaunchTemplate

시작 템플릿을 콘솔에서 생성하거나 API를 사용하여 생성하려면 보안 주체에게 IAM 정책에서 `ec2:CreateLaunchTemplate` 권한이 있어야 합니다. 가능하면 태그를 사용하여 계정의 시작 템플릿에 대한 액세스를 제어합니다.

예를 들어 다음 IAM 정책 문에서는 템플릿이 지정된 태그(*purpose*=*testing*)를 사용하는 경우에만 보안 주체에게 시작 템플릿을 생성할 수 있는 권한을 부여합니다.

```
{
    "Sid": "IAMPolicyForCreatingTaggedLaunchTemplates",
    "Action": "ec2:CreateLaunchTemplate",
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/purpose": "testing"
        }
    }
}
```

시작 템플릿을 생성하는 보안 주체에게는 다음과 같은 몇 가지 관련 권한이 필요할 수도 있습니다.

- ec2:CreateTags - `CreateLaunchTemplate` 작업 중에 시작 템플릿에 태그를 추가하려면 `CreateLaunchTemplate` 호출자는 IAM 정책에서 `ec2:CreateTags` 권한이 있어야 합니다.
- ec2:RunInstances - 생성한 시작 템플릿에서 EC2 인스턴스를 시작하려면 보안 주체는 IAM 정책에서 `ec2:RunInstances` 권한도 있어야 합니다.

태그를 적용하는 리소스 생성 작업의 경우 사용자는 `ec2:CreateTags` 권한이 있어야 합니다. 다음 IAM 정책 문은 `ec2:CreateAction` 조건 키를 사용하여 사용자가 `CreateLaunchTemplate`의 컨텍스트에서만 태그를 생성하도록 허용합니다. 사용자는 기존 시작 템플릿이나 다른 어떤 리소스에도 태

그를 지정할 수 없습니다. 자세한 내용은 생성 시 Amazon EC2 리소스 태그 지정에 대한 권한 부여 섹션을 참조하세요.

```
{
    "Sid": "IAMPolicyForTaggingLaunchTemplatesOnCreation",
    "Action": "ec2:CreateTags",
    "Effect": "Allow",
    "Resource": "arn:aws:ec2:region:account-id:launch-template/*",
    "Condition": {
        "StringEquals": {
            "ec2:CreateAction": "CreateLaunchTemplate"
        }
    }
}
```

시작 템플릿을 생성한 IAM 사용자에게는 자신이 생성한 시작 템플릿을 사용할 수 있는 권한이 자동으로 부여되지 않습니다. 다른 보안 주체와 마찬가지로 시작 템플릿 생성자는 IAM 정책을 통해 권한을 받아야 합니다. IAM 사용자가 시작 템플릿에서 EC2 인스턴스를 시작하려는 경우 `ec2:RunInstances` 권한이 있어야 합니다. 이러한 권한을 부여할 때 사용자가 특정 태그 또는 특정 ID의 시작 템플릿만 사용할 수 있도록 지정할 수 있습니다. 또한 `RunInstances` 호출에 대한 리소스 수준 권한을 지정하여 시작 템플릿을 사용하는 모든 사용자가 인스턴스를 시작할 때 참조하고 사용할 수 있는 AMI 및 기타 리소스를 제어할 수도 있습니다. 예시 정책은 시작 템플릿 섹션을 참조하세요.

ec2:DescribeLaunchTemplates

계정의 시작 템플릿을 나열하고 보려면 보안 주체에게 IAM 정책에 `ec2:DescribeLaunchTemplates` 권한이 있어야 합니다. `Describe` 작업은 리소스 수준 권한을 지원하지 않으므로 조건 없이 지정해야 하며 정책의 리소스 요소 값은 `"*"`여야 합니다.

예를 들어 다음 IAM 정책 문은 보안 주체에게 계정의 모든 시작 템플릿을 나열하고 볼 수 있는 권한을 부여합니다.

```
{
    "Sid": "IAMPolicyForDescribingLaunchTemplates",
    "Action": "ec2:DescribeLaunchTemplates",
    "Effect": "Allow",
    "Resource": "*"
}
```

ec2:DescribeLaunchTemplateVersions

시작 템플릿을 나열하고 보는 보안 주체에게는 시작 템플릿을 구성하는 전체 속성 집합을 검색할 수 있는 `ec2:DescribeLaunchTemplateVersions` 권한도 있어야 합니다.

계정에 시작 템플릿 버전을 나열하고 보려면 보안 주체에게 IAM 정책에서 `ec2:DescribeLaunchTemplateVersions` 권한이 있어야 합니다. `Describe` 작업은 리소스 수준 권한을 지원하지 않으므로 조건 없이 지정해야 하며 정책의 리소스 요소 값은 `"*"`여야 합니다.

예를 들어 다음 IAM 정책 문은 보안 주체에게 계정에서 모든 시작 템플릿 버전을 나열하고 볼 수 있는 권한을 부여합니다.

```
{
    "Sid": "IAMPolicyForDescribingLaunchTemplateVersions",
    "Effect": "Allow",
    "Action": "ec2:DescribeLaunchTemplateVersions",
    "Resource": "*"
}
```

ec2:DeleteLaunchTemplate

> **Important**
>
> 보안 주체에게 리소스를 삭제할 수 있는 권한을 부여할 때는 주의해야 합니다. 시작 템플릿을 삭제하면 그 시작 템플릿에 의존하는 AWS 리소스에 오류가 발생할 수도 있습니다.

시작 템플릿을 삭제하려면 보안 주체에게 IAM 정책에서 `ec2:DeleteLaunchTemplate` 권한이 있어야 합니다. 가능하면 태그 기반 조건 키를 사용하여 사용 권한을 제한합니다.

예를 들어 다음 IAM 정책 문에서는 템플릿에 지정된 태그(*purpose*=*testing*)가 있는 경우에만 보안 주체에게 시작 템플릿을 삭제할 수 있는 권한을 부여합니다.

```
{
    "Sid": "IAMPolicyForDeletingLaunchTemplates",
    "Action": "ec2:DeleteLaunchTemplate",
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
```

```
            "aws:ResourceTag/purpose": "testing"
        }
    }
}
```

또는 ARN을 사용하여 IAM 정책이 적용되는 시작 템플릿을 식별할 수 있습니다.

시작 템플릿에는 다음과 같은 ARN이 있습니다.

```
"Resource": "arn:aws:ec2:region:account-id:launch-template/lt-09477bcd97b0d310e"
```

여러 ARN을 목록으로 묶어 지정하거나 Condition 요소 없이 Resource 값 "*"를 지정하여 보안 주체가 계정의 모든 시작 템플릿을 삭제하도록 허용할 수 있습니다.

버전 관리 권한 제어

신뢰할 수 있는 관리자에게는 다음 예와 유사한 IAM 정책을 사용하여 시작 템플릿의 버전을 생성 및 삭제하고 시작 템플릿의 기본 버전을 변경할 수 있는 액세스 권한을 부여할 수 있습니다.

Important

보안 주체에게 시작 템플릿 버전을 생성하거나 시작 템플릿을 수정할 수 있는 권한을 부여할 때는 주의해야 합니다.

- 시작 템플릿 버전을 생성하면 Amazon EC2가 사용자를 대신하여 Latest 버전으로 인스턴스를 시작할 수 있는 AWS 리소스에 영향을 줍니다.
- 시작 템플릿을 수정하면 Default인 버전을 변경할 수 있으므로 Amazon EC2가 사용자를 대신하여 이 수정된 버전으로 인스턴스를 시작할 수 있는 AWS 리소스에 영향을 줍니다.

또한 Latest 또는 Default 시작 템플릿 버전과 상호 작용하는 AWS 리소스(예: EC2 플릿 및 스팟 플릿)를 처리하는 방법에 주의해야 합니다. Latest 또는 Default에 다른 시작 템플릿 버전이 사용되는 경우 AWS 리소스와 사용자 상호 작용이 없기 때문에 Amazon EC2는 플릿의 목표 용량을 충족하기 위해 새 인스턴스를 시작할 때 완료해야 할 작업에 대한 권한을 다시 확인하지 않습니다. 사용자에게 CreateLaunchTemplateVersion 및 ModifyLaunchTemplate API를 호출할 수 있는 권한을 부여하면 플릿이 인스턴스 프로파일(IAM 역할에 대한 컨테이너)이 포함된 다른 시작 템플릿 버전을 가리키는 경우에도 사용자에게 사실상 iam:PassRole 권한이 부여됩니다. 따라서 사용자는 iam:PassRole 권한이 없더라도 잠재적으로 시작 템플릿을 업데이트하여 IAM 역할을 인스턴스에 전달할 수 있습니다. 시

작 템플릿 버전을 생성하고 관리할 수 있는 사람에게 권한을 부여할 때 주의를 기울이면 이러한 위험을 관리할 수 있습니다.

ec2:CreateLaunchTemplateVersion

시작 템플릿의 새 버전을 생성하려면 보안 주체에게 IAM 정책에서 시작 템플릿에 대한 `ec2:CreateLaunchTemplateVersion` 권한이 있어야 합니다.

예를 들어 다음 IAM 정책 문에서는 버전이 지정된 태그(*environment*=*production*)를 사용하는 경우에만 보안 주체에게 시작 템플릿 버전을 생성할 수 있는 권한을 부여합니다. 또는 하나 이상의 시작 템플릿 ARN을 지정하거나 `Condition` 요소 없이 `Resource` 값 `"*"`를 지정하여 보안 주체가 계정에 있는 모든 시작 템플릿의 버전을 생성하도록 허용할 수 있습니다.

```
{
    "Sid": "IAMPolicyForCreatingLaunchTemplateVersions",
    "Action": "ec2:CreateLaunchTemplateVersion",
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/environment": "production"
        }
    }
}
```

ec2:DeleteLaunchTemplateVersion

> **Important**
>
> 보안 주체에 리소스를 삭제할 수 있는 권한을 부여할 때는 항상 주의해야 합니다. 시작 템플릿 버전을 삭제하면 그 시작 템플릿 버전에 의존하는 AWS 리소스에 오류가 발생할 수도 있습니다.

시작 템플릿 버전을 삭제하려면 보안 주체에게 IAM 정책에서 시작 템플릿에 대한 `ec2:DeleteLaunchTemplateVersion` 권한이 있어야 합니다.

예를 들어 다음 IAM 정책 문에서는 버전이 지정된 태그(*environment*=*production*)를 사용하는 경우에만 보안 주체에게 시작 템플릿 버전을 삭제할 수 있는 권한을 부여합니다. 또는 하나 이상의 템플

릿 ARN을 지정하거나 Condition 요소 없이 Resource 값 "*"를 지정하여 보안 주체가 계정에 있는 모든 시작 템플릿의 버전을 삭제하도록 허용할 수 있습니다.

```
{
    "Sid": "IAMPolicyForDeletingLaunchTemplateVersions",
    "Action": "ec2:DeleteLaunchTemplateVersion",
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/environment": "production"
        }
    }
}
```

ec2:ModifyLaunchTemplate

시작 템플릿과 연결된 Default 버전을 변경하려면 보안 주체에게 IAM 정책에서 시작 템플릿에 대한 ec2:ModifyLaunchTemplate 권한이 있어야 합니다.

예를 들어 다음 IAM 정책 문에서는 시작 템플릿이 지정된 태그(*environment*=*production*)를 사용하는 경우에만 보안 주체에게 시작 템플릿을 수정할 수 있는 권한을 부여합니다. 또는 하나 이상의 시작 템플릿 ARN을 지정하거나 Condition 요소 없이 Resource 값 "*"를 지정하여 보안 주체가 계정에 있는 모든 시작 템플릿을 수정하도록 허용할 수 있습니다.

```
{
    "Sid": "IAMPolicyForModifyingLaunchTemplates",
    "Action": "ec2:ModifyLaunchTemplate",
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/environment": "production"
        }
    }
}
```

시작 템플릿의 태그에 대한 액세스 제어

리소스가 시작 템플릿인 경우 조건 키를 사용하여 태그 지정 권한을 제한할 수 있습니다. 예를 들어 다음 IAM 정책은 지정된 계정과 리전의 시작 템플릿에서 *temporary* 키가 있는 태그만 제거하도록 허용합니다.

```
{
    "Sid": "IAMPolicyForDeletingTagsOnLaunchTemplates",
    "Action": "ec2:DeleteTags",
    "Effect": "Allow",
    "Resource": "arn:aws:ec2:region:account-id:launch-template/*",
    "Condition": {
        "ForAllValues:StringEquals": {
            "aws:TagKeys": ["temporary"]
        }
    }
}
```

Amazon EC2 리소스에 적용할 수 있는 태그 키 및 값을 제어하는 데 사용할 수 있는 조건 키에 대한 자세한 내용은 특정 태그에 대한 액세스 제어를 참조하세요.

Amazon EC2 시작 템플릿을 사용하여 Amazon EC2 인스턴스 시작 제어

사용자가 시작 템플릿을 사용하는 경우에만 인스턴스를 시작할 수 있고, 특정 시작 템플릿만 사용할 수 있도록 지정하여 Amazon EC2 인스턴스의 구성을 제어할 수 있습니다. 또한 시작 템플릿과 시작 템플릿 버전을 생성, 수정, 설명 및 삭제할 수 있는 사용자를 제어할 수 있습니다.

시작 템플릿을 사용하여 시작 파라미터 제어

시작 템플릿에는 시작 시 인스턴스를 구성하는 데 필요한 모든 파라미터 또는 일부파라미터가 포함될 수 있습니다. 그러나 시작 템플릿을 사용해 인스턴스를 시작할 때 시작 템플릿에 지정된 파라미터를 재정의할 수 있습니다. 또는 시작 템플릿에 없는 추가 파라미터를 지정할 수 있습니다.

> **Note**
>
> 시작 중에는 시작 템플릿 파라미터를 제거할 수 없습니다(예를 들어 파라미터에 대해 null 값을 지정할 수 없습니다). 파라미터를 제거하려면 파라미터 없이 새로운 버전의 시작 템플릿을 생성하고 이 버전을 사용하여 인스턴스를 시작합니다.

인스턴스를 시작하려면 사용자에게 `ec2:RunInstances` 작업을 사용할 수 있는 권한이 있어야 합니다. 사용자는 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 있습

니다. 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. 예시 정책은 시작 템플릿 섹션을 참조하세요.

시작 템플릿 사용 제어

기본적으로 사용자에게는 시작 템플릿 사용 권한이 없습니다. 사용자에게 시작 템플릿과 시작 템플릿 버전을 생성, 수정, 설명 및 삭제할 수 있는 권한을 부여하는 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 예: 시작 템플릿 작업 정책 예제를 참조하세요.

`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.

EC2 플릿 또는 스팟 플릿에서 시작 템플릿을 사용할 때의 중요한 보안 문제

시작 템플릿을 사용하려면 사용자에게 시작 템플릿과 시작 템플릿 버전을 생성, 수정, 설명, 삭제할 수 있는 권한을 부여해야 합니다. `ec2:CreateLaunchTemplate` 및 `ec2:CreateLaunchTemplateVersion` 작업에 대한 액세스를 제어하여 시작 템플릿 및 시작 템플릿 버전을 생성할 수 있는 사용자를 제어할 수 있습니다. 또한 `ec2:ModifyLaunchTemplate` 작업에 대한 액세스를 제어하여 시작 템플릿을 수정할 수 있는 사용자를 제어할 수 있습니다.

> **Important**
>
> EC2 플릿 또는 스팟 플릿이 최신 또는 기본 시작 템플릿 버전을 사용하도록 구성된 경우 플릿은 나중에 최신 또는 기본값이 다른 시작 템플릿 버전을 가리키도록 변경되었는지 여부를 인식하지 못합니다. 최신 또는 기본에 다른 시작 템플릿 버전이 사용되는 경우 Amazon EC2는 플릿의 목표 용량을 충족하기 위해 새 인스턴스를 시작할 때 완료해야 할 작업에 대한 권한을 다시 확인하지 않습니다. 이는 시작 템플릿 버전을 만들고 관리할 수 있는 사람에게 권한을 부여할 때 중요한 고려 사항이며, 특히 사용자에게 기본 시작 템플릿 버전 변경을 허용하는 `ec2:ModifyLaunchTemplate` 작업의 경우 더욱 그렇습니다.

사용자에게 시작 템플릿 API에 대한 EC2 작업을 사용할 수 있는 권한을 부여하면 사용자는 인스턴스 프로파일(IAM 역할에 대한 컨테이너)이 포함된 다른 시작 템플릿 버전을 가리키도록 EC2 플릿 또는 스팟 플릿을 생성하거나 업데이트하는 경우에도 사용자에게 사실상 `iam:PassRole` 권한이 부여됩니다. 따라서 사용자는 `iam:PassRole` 권한이 없더라도 잠재적으로 시작 템플릿을 업데이트하여 IAM 역할

을 인스턴스에 전달할 수 있습니다. 자세한 정보와 IAM 정보 예시는 IAM 사용 설명서의 IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여를 참조하세요.

자세한 내용은 시작 템플릿 사용 제어 및 예: 시작 템플릿 작업(을)를 참조하세요.

Amazon EC2 시작 템플릿 생성

인스턴스 구성 파라미터의 값을 직접 지정하거나 기존 시작 템플릿 또는 Amazon EC2 인스턴스에서 값을 가져와서 Amazon EC2 시작 템플릿을 생성할 수 있습니다.

시작 템플릿의 모든 파라미터에 대해 값을 지정할 필요는 없으며, 인스턴스 구성 파라미터 하나만 지정하면 시작 템플릿을 생성할 수 있습니다. 지정하지 않기로 선택한 파라미터를 표시하려면 콘솔 사용 시 시작 템플릿에 포함하지 않음을 선택합니다. 명령줄 도구를 사용할 때는 시작 템플릿에서 파라미터를 지정하지 않기로 선택했음을 나타내는 파라미터를 포함하지 않습니다

시작 템플릿에 AMI를 지정하려면 AMI를 선택하거나 인스턴스 시작 시 AMI를 가리키는 Systems Manager 파라미터를 지정합니다.

시작 템플릿으로 인스턴스이 시작될 때 시작 템플릿에 지정된 값을 사용하여 해당 인스턴스 파라미터가 구성됩니다. 시작 템플릿에 값이 지정되지 않은 경우 해당 인스턴스 파라미터의 기본값이 사용됩니다.

업무

- 파라미터를 지정하여 시작 템플릿 생성
- 기존 시작 템플릿에서 시작 템플릿 생성
- 인스턴스에서 시작 템플릿 생성
- AMI ID 대신 Systems Manager 파라미터 사용

파라미터를 지정하여 시작 템플릿 생성

시작 템플릿을 만들려면 시작 템플릿 이름과 하나 이상의 인스턴스 구성 파라미터를 지정해야 합니다.

파라미터에 대한 설명은 Amazon EC2 인스턴스 구성 파라미터에 대한 참조 섹션을 참조하세요.

Console

시작 템플릿을 생성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 시작 템플릿을 선택한 다음 시작 템플릿 생성을 선택합니다.

3. 시작 템플릿 이름 및 설명에서 다음을 수행합니다.

   a. Launch template name에 대해 시작 템플릿의 설명이 포함된 이름을 입력하세요.

   b. [템플릿 버전 설명(Template version description)]에 시작 템플릿의 이 버전에 대한 간단한 설명을 입력합니다.

   c. 생성 시 시작 템플릿에 태그를 지정하려면 템플릿 태그를 확장하고 새 태그 추가를 선택한 다음 태그 키와 값 페어를 입력합니다. 추가할 각 추가 태그에 대해 새 태그 추가를 다시 선택합니다.

   > **Note**
   >
   > 인스턴스가 시작될 때 생성되는 리소스에 태깅하려면 리소스 태그(Resource tags)에서 태깅해야 합니다. 자세한 내용은 이 절차의 9단계를 참조하세요.

4. 애플리케이션 및 OS 이미지(Amazon Machine Image)에서 시작 템플릿에 포함하지 않음을 선택된 상태로 유지하거나 인스턴스의 운영 체제(OS)를 선택한 다음 AMI를 선택할 수 있습니다. 또는 AMI를 지정하는 대신 Systems Manager 파라미터를 지정할 수 있습니다. 자세한 내용은 AMI ID 대신 Systems Manager 파라미터 사용 섹션을 참조하세요.

   AMI는 인스턴스를 시작하는 데 필요한 운영 체제 및 소프트웨어가 포함된 템플릿입니다.

5. 인스턴스 유형에서 시작 템플릿에 포함하지 않음을 선택된 상태로 유지하거나, 인스턴스 유형을 선택하거나, 인스턴스 속성을 지정하여 Amazon EC2가 해당 속성으로 인스턴스 유형을 식별하도록 할 수 있습니다.

   > **Note**
   >
   > 인스턴스 속성 지정은 Auto Scaling 그룹, EC2 플릿 및 스팟 플릿에서 인스턴스를 시작하기 위해 시작 템플릿을 사용하는 경우에만 지원됩니다. 자세한 내용은 속성 기반 인스턴스 유형 선택을 사용하여 혼합 인스턴스 그룹 생성 및 EC2 플릿 또는 스팟 플릿의 인스턴스 유형 선택에 대한 속성 지정 섹션을 참조하세요.
   > 인스턴스 시작 마법사 또는 RunInstances API에서 시작 템플릿을 사용하려는 경우 인스턴스 유형 속성을 지정할 수 없습니다.

   인스턴스 유형은 인스턴스에 사용되는 호스트 컴퓨터의 하드웨어 구성(CPU, 메모리, 스토리지 및 네트워킹 용량) 과 크기를 결정합니다.

   어떤 인스턴스 유형을 선택할지 확실하지 않으면 다음을 수행합니다.

- 인스턴스 유형 비교를 선택하여 vCPU 수, 아키텍처, 메모리 양(GiB), 스토리지 양(GB), 스토리지 유형 및 네트워크 성능과 같은 속성을 기준으로 서로 다른 인스턴스 유형을 비교합니다.
- 조언 받기를 선택하여 Amazon Q EC2 인스턴스 유형 선택기에서 인스턴스 유형에 대한 지침과 제안을 받습니다. 자세한 내용은 EC2 인스턴스 유형 파인더에서 권장 사항 받기 섹션을 참조하세요.

> **Note**
>
> AWS 계정 사용 기간이 12개월 미만인 경우 t2.micro 인스턴스 유형 또는 t2.micro를 사용할 수 없는 리전의 경우 t3.micro 인스턴스 유형을 선택하여 프리 티어로 Amazon EC2를 사용할 수 있습니다. t3.micro 인스턴스를 시작하면 기본적으로 무제한 모드로 설정되므로 CPU 사용량에 따라 추가 요금이 부과될 수 있다는 점에 유의하세요. 인스턴스 유형이 프리 티어 사용 자격이 있으면 프리 티어 사용 가능(Free tier eligible)으로 표시됩니다.

6. 키 페어(로그인)에서 키 페어 이름에 대해 시작 템플릿에 포함하지 않음을 선택된 상태로 두거나, 기존 키 페어를 선택하거나, 새 키 페어를 생성합니다.
7. 네트워크 설정에서 시작 템플릿에 포함하지 않음을 선택된 상태로 두거나 다양한 네트워크 설정에 대한 값을 지정할 수 있습니다.
8. 스토리지 구성에서 시작 템플릿에 AMI를 지정한 경우 AMI에 루트 볼륨(볼륨 1(AMI 루트)을 포함한 하나 이상의 스토리지 볼륨이 포함됩니다. 필요에 따라 인스턴스에 연결할 추가 볼륨을 지정할 수 있습니다. 새 볼륨을 추가하려면 새 볼륨 추가를 선택합니다.
9. 리소스 태그에서 인스턴스가 시작될 때 생성되는 리소스에 태그를 지정하려면 태그 추가를 선택하고 태그 키와 값 페어를 입력합니다. 리소스 유형(Resource types)에서 생성 시 태깅할 리소스를 지정합니다. 모든 리소스에 대해 동일한 태그를 지정하거나 다른 리소스에 대해 다른 태그를 지정할 수 있습니다. 추가할 각 추가 태그에 대해 태그 추가(Add tag)를 다시 선택합니다.

   시작 템플릿을 사용할 때 생성되는 다음 리소스에 대한 태그를 지정할 수 있습니다.

   - 인스턴스
   - 볼륨
   - Elastic Graphics
   - 스팟 인스턴스 요청

- 네트워크 인터페이스

> **Note**
>
> 시작 템플릿 자체에 태깅하려면 템플릿 태그(Template tags)에서 태깅해야 합니다. 자세한 내용은 이 절차의 3단계를 참조하세요.

10. 고급 세부 정보에서 필드를 볼 수 있도록 섹션을 확장하고 필요에 따라 인스턴스를 위한 추가 파라미터를 지정합니다.
11. 요약 패널을 사용하여 시작 템플릿 구성을 검토합니다. 링크를 선택하여 원하는 섹션으로 이동한 다음, 필요에 따라 변경할 수 있습니다.
12. 시작 템플릿을 생성할 준비가 되었으면 시작 템플릿 생성(Create launch template)을 선택합니다.

AWS CLI

시작 템플릿 생성

create-launch-template 명령을 사용합니다.

```
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 인스턴스 구성에 대한 시작 템플릿 데이터를 포함하는 JSON 파일 예제입니다. 예제 명령에 표시된 대로 JSON을 파일에 저장하고 `--launch-template-data` 파라미터에 포함합니다.

```
{
    "NetworkInterfaces": [{
        "AssociatePublicIpAddress": true,
        "DeviceIndex": 0,
        "Ipv6AddressCount": 1,
        "SubnetId": "subnet-0abcdef1234567890"
    }],
    "ImageId": "ami-0abcdef1234567890",
    "InstanceType": "r5.4xlarge",
```

```
    "TagSpecifications": [{
        "ResourceType": "instance",
        "Tags": [{
            "Key":"Name",
            "Value":"webserver"
        }]
    }],
    "CpuOptions": {
        "CoreCount":4,
        "ThreadsPerCore":2
    }
}
```

출력의 예시는 다음과 같습니다.

```
{
    "LaunchTemplate": {
        "LatestVersionNumber": 1,
        "LaunchTemplateId": "lt-01238c059e3466abc",
        "LaunchTemplateName": "TemplateForWebServer",
        "DefaultVersionNumber": 1,
        "CreatedBy": "arn:aws:iam::123456789012:root",
        "CreateTime": "2017-11-27T09:13:24.000Z"
    }
}
```

PowerShell

시작 템플릿 생성

New-EC2LaunchTemplate cmdlet을 사용합니다.

```
$launchTemplateData = [Amazon.EC2.Model.RequestLaunchTemplateData]@{
    ImageId = 'ami-0abcdef1234567890'
    InstanceType = 'r5.4xlarge'
    NetworkInterfaces = @(

 [Amazon.EC2.Model.LaunchTemplateInstanceNetworkInterfaceSpecificationRequest]@{
            AssociatePublicIpAddress = $true
            DeviceIndex = 0
            Ipv6AddressCount = 1
            SubnetId = 'subnet-0abcdef1234567890'
        }
```

```
    )
    TagSpecifications = @(
        [Amazon.EC2.Model.LaunchTemplateTagSpecificationRequest]@{
            ResourceType = 'instance'
            Tags = [Amazon.EC2.Model.Tag]@{
                Key = 'Name'
                Value = 'webserver'
            }
        }
    )
    CpuOptions = [Amazon.EC2.Model.LaunchTemplateCpuOptionsRequest]@{
        CoreCount = 4
        ThreadsPerCore = 2
    }
}
$tagSpecificationData = [Amazon.EC2.Model.TagSpecification]@{
    ResourceType = 'launch-template'
    Tags = [Amazon.EC2.Model.Tag]@{
        Key = 'purpose'
        Value = 'production'
    }
}
New-EC2LaunchTemplate -LaunchTemplateName 'TemplateForWebServer' `
    -VersionDescription 'WebVersion1' `
    -LaunchTemplateData $launchTemplateData `
    -TagSpecification $tagSpecificationData
```

출력의 예시는 다음과 같습니다.

```
CreatedBy            : arn:aws:iam::123456789012:root
CreateTime           : 9/19/2023 16:57:55
DefaultVersionNumber : 1
LatestVersionNumber  : 1
LaunchTemplateId     : lt-01238c059eEXAMPLE
LaunchTemplateName   : TemplateForWebServer
Tags                 : {purpose}
```

기존 시작 템플릿에서 시작 템플릿 생성

기존 시작 템플릿을 복제하고 파라미터를 조정하여 새 시작 템플릿을 생성할 수 있습니다. 그러나 Amazon EC2 콘솔을 사용하는 경우에만 이렇게 할 수 있습니다. AWS CLI는 템플릿 복제를 지원하지

않습니다. 파라미터에 대한 설명은 Amazon EC2 인스턴스 구성 파라미터에 대한 참조 섹션을 참조하세요.

Console

기존 시작 템플릿에서 시작 템플릿을 생성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 시작 템플릿을 선택한 다음 시작 템플릿 생성을 선택합니다.
3. Launch template name에 대해 시작 템플릿의 설명이 포함된 이름을 입력하세요.
4. [템플릿 버전 설명(Template version description)]에 시작 템플릿의 이 버전에 대한 간단한 설명을 입력합니다.
5. 생성 시 시작 템플릿에 태그를 지정하려면 템플릿 태그를 확장하고 새 태그 추가를 선택한 다음 태그 키와 값 페어를 입력합니다.
6. 소스 템플릿을 확장하고 시작 템플릿 이름에 대해 새 시작 템플릿의 기준으로 사용할 시작 템플릿을 선택합니다.
7. 소스 템플릿 버전에서 새로운 시작 템플릿의 토대가 되는 시작 템플릿 버전을 선택합니다.
8. 필요에 따라 시작 파라미터를 조정하고 시작 템플릿 생성을 선택합니다.

인스턴스에서 시작 템플릿 생성

기존 Amazon EC2 인스턴스의 파라미터를 복제하고 파라미터를 조정하여 시작 템플릿을 생성할 수 있습니다. 파라미터에 대한 설명은 Amazon EC2 인스턴스 구성 파라미터에 대한 참조 섹션을 참조하세요.

Console

인스턴스에서 시작 템플릿을 생성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 작업, 이미지 및 템플릿, 인스턴스에서 템플릿 생성을 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. 시작 템플릿 생성을 선택합니다.

AWS CLI

AWS CLI를 사용하면 먼저 인스턴스에서 시작 템플릿 데이터를 가져온 다음 시작 템플릿 데이터를 사용해 시작 템플릿을 생성하여 기존 인스턴스에서 시작 템플릿을 생성할 수 있습니다.

인스턴스에서 시작 템플릿 데이터를 가져오려면

- get-launch-template-data 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 예시 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 데이터를 사용하여 시작 템플릿을 생성하려면

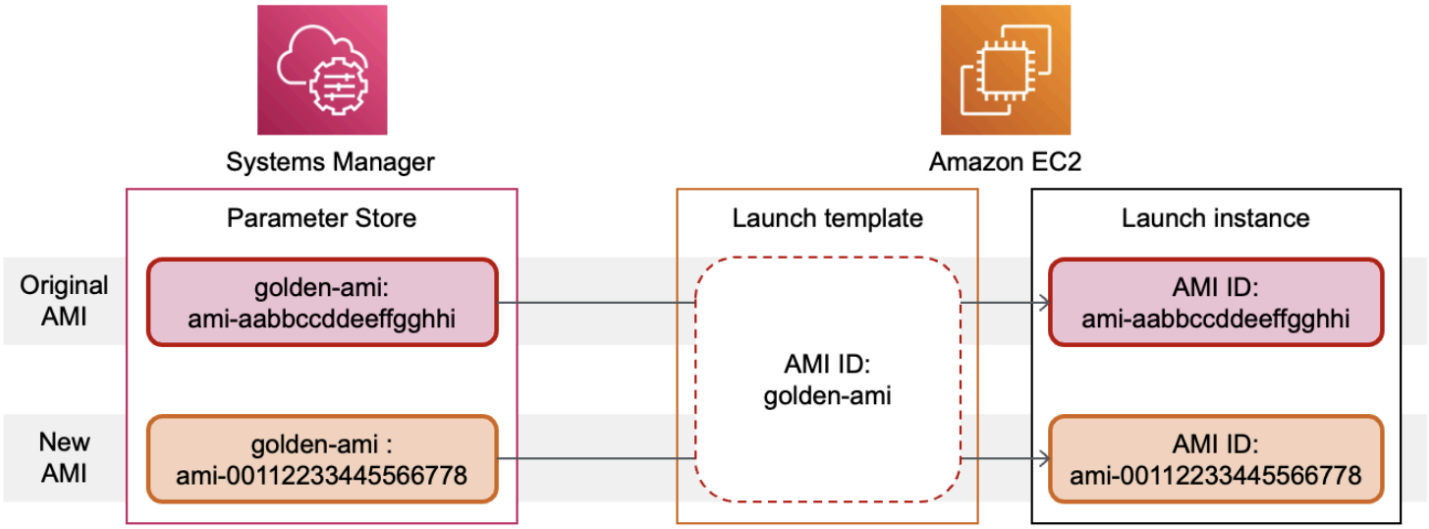
- create-launch-template 명령을 사용하여 이전 절차의 출력을 사용해 시작 템플릿을 생성합니다. AWS CLI를 사용하여 시작 템플릿을 생성하는 방법에 대한 자세한 내용은 파라미터를 지정하여 시작 템플릿 생성 섹션을 참조하세요.

AMI ID 대신 Systems Manager 파라미터 사용

시작 템플릿에 AMI ID를 지정하는 대신 AWS Systems Manager 파라미터를 지정할 수 있습니다. AMI ID가 변경되면 Systems Manager Parameter Store에서 Systems Manager 파라미터를 업데이트하여 한 곳에서 AMI ID를 업데이트할 수 있습니다. 다른 AWS 계정과 파라미터를 공유할 수도 있습니다. 하나의 계정에 AMI 파라미터를 중앙에서 저장 및 관리하고 이를 참조해야 하는 다른 모든 계정과 공유할 수 있습니다. Systems Manager 파라미터를 사용하면 한 번의 작업으로 모든 시작 템플릿을 업데이트할 수 있습니다.

Systems Manager 파라미터는 AWS Systems Manager Parameter Store에서 생성하는 사용자 정의 키-값 페어입니다. Parameter Store는 애플리케이션 구성 값을 저장할 수 있는 중앙 위치를 제공합니다.

다음 다이어그램에서 `golden-ami` 파라미터는 먼저 Parameter Store의 원래 AMI `ami-aabbccddeeffgghhi`에 매핑됩니다. 시작 템플릿에서 AMI ID의 값은 `golden-ami`입니다. 이 시작 템플릿을 사용하여 인스턴스가 시작되면 AMI ID가 `ami-aabbccddeeffgghhi`로 확인됩니다. 나중에 AMI가 업데이트되어 새 AMI ID가 생성됩니다. Parameter Store에서 `golden-ami` 파라미터는 새 `ami-00112233445566778`에 매핑됩니다. 시작 템플릿은 변경되지 않은 상태로 유지됩니다. 이 시작 템플릿을 사용하여 인스턴스가 시작되면 AMI ID가 새 `ami-00112233445566778`로 확인됩니다.



AMI ID에 대한 Systems Manager 파라미터 형식

시작 템플릿을 사용하려면 AMI ID 대신 사용 시 사용자 정의 Systems Manager 파라미터가 다음 형식을 준수해야 합니다.

- 파라미터 유형: `String`

- 파라미터 데이터 유형: `aws:ec2:image` - 입력한 값이 AMI ID에 올바른 형식인지 Parameter Store에서 검증하도록 합니다.

AMI ID에 유효한 파라미터를 생성하는 방법에 대한 자세한 내용은 AWS Systems Manager 사용 설명서의 Systems Manager 파라미터 생성을 참조하세요.

시작 템플릿의 Systems Manager 파라미터 형식

시작 템플릿에서 AMI ID 대신 Systems Manager 파라미터를 사용하려면 시작 템플릿에 파라미터를 지정할 때 다음 형식 중 하나를 사용해야 합니다.

공용 파라미터 참조:

- `resolve:ssm:public-parameter`

동일한 계정에 저장된 파라미터 참조:

- `resolve:ssm:parameter-name`
- `resolve:ssm:parameter-name:version-number` - 버전 번호 자체가 기본 레이블입니다.
- `resolve:ssm:parameter-name:label`

다른 AWS 계정에서 공유된 파라미터 참조:

- `resolve:ssm:parameter-ARN`
- `resolve:ssm:parameter-ARN:version-number`
- `resolve:ssm:parameter-ARN:label`

파라미터 버전

Systems Manager 파라미터는 버전이 지정된 리소스입니다. 파라미터를 업데이트하면 파라미터의 새 버전이 연속적으로 생성됩니다. Systems Manager는 특정 버전의 파라미터에 매핑할 수 있는 파라미터 레이블을 지원합니다.

예를 들어, `golden-ami` 파라미터에는 1, 2, 3의 세 가지 버전이 있을 수 있습니다. 버전 2에 매핑되는 파라미터 레이블 `beta`와 버전 3에 매핑되는 파라미터 레이블 `prod`를 생성할 수 있습니다.

다음 형식 중 하나를 사용하여 시작 템플릿에 `golden-ami` 파라미터의 버전 3을 지정할 수 있습니다.

- `resolve:ssm:golden-ami:3`
- `resolve:ssm:golden-ami:prod`

버전 또는 레이블 지정은 선택 사항입니다. 버전 또는 레이블이 지정되지 않은 경우 파라미터의 최신 버전이 사용됩니다.

시작 템플릿에 Systems Manager 파라미터 지정

시작 템플릿 또는 새 버전의 시작 템플릿을 생성할 때 시작 템플릿에 AMI ID 대신 Systems Manager 파라미터를 지정할 수 있습니다.

Console

시작 템플릿에 Systems Manager 파라미터 지정

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 시작 템플릿을 선택한 다음 시작 템플릿 생성을 선택합니다.
3. Launch template name에 대해 시작 템플릿의 설명이 포함된 이름을 입력하십시오.
4. Application and OS Images (Amazon Machine Image)(애플리케이션 및 OS 이미지(Amazon Machine Image))에서 Browse more AMIs(더 많은 AMI 찾아보기)를 선택하세요.
5. 검색 창 오른쪽에 있는 화살표 버튼을 선택한 다음 사용자 지정 값/Systems Manager 파라미터 지정을 선택하세요.
6. 사용자 지정 값 또는 Systems Manager 파라미터 지정 대화 상자에서 다음을 수행합니다.

   a. AMI ID 또는 Systems Manager 파라미터 문자열에 다음 형식 중 하나를 사용하여 Systems Manager 파라미터 이름을 입력합니다.

      공용 파라미터 참조:

      - **resolve:ssm:*public-parameter***

      동일한 계정에 저장된 파라미터 참조:

      - **resolve:ssm:*parameter-name***
      - **resolve:ssm:*parameter-name*:*version-number***
      - **resolve:ssm:*parameter-name*:*label***

다른 AWS 계정에서 공유된 파라미터 참조:

- **resolve:ssm:*parameter-ARN***
- **resolve:ssm:*parameter-ARN*:*version-number***
- **resolve:ssm:*parameter-ARN*:*label***

b. 저장을 선택합니다.

7. 필요에 따라 다른 시작 템플릿 파라미터를 지정하고 시작 템플릿 생성을 선택합니다.

자세한 내용은 파라미터를 지정하여 시작 템플릿 생성 섹션을 참조하세요.

AWS CLI

시작 템플릿에 Systems Manager 파라미터 지정

- create-launch-template 명령을 사용하여 시작 템플릿을 생성합니다. 사용할 AMI를 지정하려면 다음 형식 중 하나를 사용하여 Systems Manager 파라미터 이름을 입력합니다.

  공용 파라미터 참조:

  - **resolve:ssm:*public-parameter***

  동일한 계정에 저장된 파라미터 참조:

  - **resolve:ssm:*parameter-name***
  - **resolve:ssm:*parameter-name*:*version-number***
  - **resolve:ssm:*parameter-name*:*label***

  다른 AWS 계정에서 공유된 파라미터 참조:

  - **resolve:ssm:*parameter-ARN***
  - **resolve:ssm:*parameter-ARN*:*version-number***
  - **resolve:ssm:*parameter-ARN*:*label***

  아래 예제에서는 다음을 지정하는 시작 템플릿을 생성합니다.

  - 시작 템플릿의 이름(*TemplateForWebServer*)

- 시작 템플릿에 대한 태그(*purpose*=*production*)
- JSON 파일에 지정된 인스턴스 구성에 대한 데이터:
  - 사용할 AMI(resolve:ssm:*golden-ami*)
  - 시작할 인스턴스 유형(*m5.4xlarge*)
  - 인스턴스에 대한 태그(*Name*=*webserver*)

```
aws ec2 create-launch-template \
    --launch-template-name TemplateForWebServer \
    --tag-specifications 'ResourceType=launch-
template,Tags=[{Key=purpose,Value=production}]' \
    --launch-template-data file://template-data.json
```

다음은 인스턴스 구성에 대한 시작 템플릿 데이터를 포함하는 JSON 파일의 예입니다. ImageId의 값은 필수 형식 resolve:ssm:*golden-ami*로 입력된 Systems Manager 파라미터 이름입니다.

```
{"LaunchTemplateData": {
    "ImageId": "resolve:ssm:golden-ami",
    "InstanceType": "m5.4xlarge",
    "TagSpecifications": [{
        "ResourceType": "instance",
        "Tags": [{
            "Key":"Name",
            "Value":"webserver"
        }]
    }]
  }
}
```

시작 템플릿에 올바른 AMI ID가 있는지 확인

Systems Manager 파라미터를 실제 AMI ID로 확인하는 방법

describe-launch-template-versions 명령을 사용하고 --resolve-alias 파라미터를 포함합니다.

```
aws ec2 describe-launch-template-versions \
    --launch-template-name my-launch-template \
    --versions $Default \
```

```
    --resolve-alias
```

응답에는 ImageId에 대한 AMI ID가 포함됩니다. 이 예제에서 이 시작 템플릿을 사용하여 인스턴스가 시작되면 AMI ID가 ami-0ac394d6a3example로 확인됩니다.

```
{
    "LaunchTemplateVersions": [
        {
            "LaunchTemplateId": "lt-089c023a30example",
            "LaunchTemplateName": "my-launch-template",
            "VersionNumber": 1,
            "CreateTime": "2022-12-28T19:52:27.000Z",
            "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
            "DefaultVersion": true,
            "LaunchTemplateData": {
                "ImageId": "ami-0ac394d6a3example",
                "InstanceType": "t3.micro",
            }
        }
    ]
}
```

관련 리소스

Systems Manager 파라미터 작업에 대한 자세한 내용은 Systems Manager 설명서의 다음 참조 자료에서 확인할 수 있습니다.

- Amazon EC2에서 지원하는 AMI 공용 파라미터를 조회하는 방법에 대한 자세한 내용은 AMI 공용 파라미터 호출을 참조하세요.
- 다른 AWS 계정과 또는 AWS Organizations를 통해 파라미터를 공유하는 방법에 대한 자세한 내용은 Working with shared parameters를 참조하세요.
- 파라미터가 성공적으로 생성되었는지 모니터링하는 방법에 대한 자세한 내용은 Native parameter support for Amazon Machine Image IDs를 참조하세요.

제한 사항

- instant 유형의 EC2 플릿은 AMI ID 대신 Systems Manager 파라미터가 지정된 시작 템플릿 사용을 지원합니다.

- `maintain` 및 `request` 유형의 EC2 플릿과 스팟 플릿은 AMI ID 대신 Systems Manager 파라미터가 지정된 시작 템플릿을 지원하지 않습니다. `maintain` 및 `request` 유형의 EC2 플릿과 스팟 플릿의 경우 시작 템플릿에서 AMI를 지정하려면 AMI ID를 지정해야 합니다.
- EC2 플릿에서 속성 기반 인스턴스 선택을 사용하는 경우 AMI ID 대신 Systems Manager 파라미터를 지정할 수 없습니다. 속성 기반 인스턴스 선택을 사용하는 경우 AMI ID를 지정해야 합니다.
- Amazon EC2 Auto Scaling에는 다른 제한 사항이 있습니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Use AWS Systems Manager parameters instead of AMI IDs in launch templates를 참조하세요.

시작 템플릿 수정(시작 템플릿 버전 관리)

시작 템플릿은 변경할 수 없으므로 시작 템플릿을 생성한 후에는 수정할 수 없습니다. 대신 필요한 변경 사항이 포함된 새 버전의 시작 템플릿을 만들 수 있습니다.

시작 템플릿의 여러 버전을 생성하고, 기본 버전을 설정하고, 시작 템플릿을 설명하고, 더 이상 필요하지 않은 버전을 삭제할 수 있습니다.

업무

- 시작 템플릿 버전 생성
- 기본 시작 템플릿 버전 설정
- 시작 템플릿 버전 설명

시작 템플릿 버전 생성

시작 템플릿 버전을 생성할 때 새로운 시작 파라미터를 지정하거나 기존 버전을 새 버전을 위한 기본 템플릿으로 사용할 수 있습니다. 파라미터에 대한 설명은 Amazon EC2 인스턴스 구성 파라미터에 대한 참조 섹션을 참조하세요.

Console

시작 템플릿 버전을 생성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 시작 템플릿을 선택합니다.
3. 시작 템플릿을 선택한 다음 작업, 템플릿 수정(새 버전 생성)을 선택합니다.

4. [템플릿 버전 설명(Template version description)]에 시작 템플릿의 이 버전에 대한 설명을 입력합니다.
5. (선택 사항) 소스 템플릿을 확장하고 새 시작 템플릿 버전의 기반으로 사용할 시작 템플릿 버전을 선택합니다. 새 시작 템플릿 버전은 이 시작 템플릿 버전으로부터 시작 파라미터를 상속합니다.
6. 필요에 따라 시작 파라미터를 수정합니다.
7. Create launch template(시작 템플릿 생성)을 선택합니다.

AWS CLI

시작 템플릿 버전 생성

create-launch-template-version 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-0abcdef1234567890"
```

PowerShell

시작 템플릿 버전 생성

New-EC2LaunchTemplateVersion Cmdlet을 사용합니다. 새 버전의 토대가 될 소스 버전을 지정할 수 있습니다. 새 버전은 이 버전에서 시작 파라미터를 상속하며, `LaunchTemplateData`를 사용하여 파라미터를 재정의할 수 있습니다. 아래 예제에서는 시작 템플릿 버전 1을 토대로 새 버전을 생성하고 다른 AMI ID를 지정합니다.

```
New-EC2LaunchTemplateVersion `
    -LaunchTemplateId lt-0abcd290751193123 `
    -VersionDescription WebVersion2 `
    -SourceVersion 1 `
    -LaunchTemplateData (
        New-Object `
```

```
            -TypeName Amazon.EC2.Model.RequestLaunchTemplateData `
            -Property @{ImageId = 'ami-0abcdef1234567890'}
    )
```

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.

Console

기본 시작 템플릿 버전을 설정하려면

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. 템플릿 버전의 경우 기본 버전으로 설정할 버전 번호를 선택하고 기본 버전으로 설정을 선택합니다.

AWS CLI

기본 시작 템플릿 버전을 설정하려면

modify-launch-template 명령을 사용합니다.

```
aws ec2 modify-launch-template \
    --launch-template-id lt-0abcd290751193123 \
    --default-version 2
```

PowerShell

기본 시작 템플릿 버전을 설정하려면

Edit-EC2LaunchTemplate cmdlet을 사용합니다.

```
Edit-EC2LaunchTemplate `
    -LaunchTemplateId lt-0abcd290751193123 `
    -DefaultVersion 2
```

시작 템플릿 버전 설명

콘솔을 사용하여 선택한 시작 템플릿의 모든 버전을 보거나 최신 또는 기본 버전이 특정 버전 번호와 일치하는 시작 템플릿 목록을 가져올 수 있습니다. AWS CLI를 사용하여 지정된 시작 템플릿의 모든 버전, 개별 버전 또는 버전 범위를 설명할 수 있습니다. 계정에 있는 모든 시작 템플릿의 모든 최신 버전 또는 모든 기본 버전을 설명할 수도 있습니다.

Console

시작 템플릿 버전을 설명하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 시작 템플릿을 선택합니다.
3. 특정 시작 템플릿의 버전을 보거나 최신 또는 기본 버전이 특정 버전 번호와 일치하는 시작 템플릿 목록을 가져올 수 있습니다.
   - 시작 템플릿의 버전을 보려면: 시작 템플릿을 선택합니다. 버전 탭의 버전에서 세부 정보를 볼 버전을 선택합니다.
   - 최신 버전이 특정 버전 번호와 일치하는 모든 시작 템플릿 목록을 가져오려면: 검색 창에서 최신 버전을 선택한 다음 버전 번호를 선택합니다.
   - 기본 버전이 특정 버전 번호와 일치하는 모든 시작 템플릿 목록을 가져오려면: 검색 창에서 기본 버전을 선택한 다음 버전 번호를 선택합니다.

AWS CLI

시작 템플릿 버전을 설명하려면

describe-launch-template-versions 명령을 사용하고 버전 번호를 지정합니다. 다음 예에서는 버전 *1*과 *3*이 지정됩니다.

```
aws ec2 describe-launch-template-versions \
    --launch-template-id lt-0abcd290751193123 \
    --versions 1 3
```

계정의 최신 및 기본 시작 템플릿 버전을 설명하려면

describe-launch-template-versions 명령을 사용하고 `$Latest`, `$Default` 또는 둘 모두를 지정합니다. 호출에서 시작 템플릿 ID와 이름을 생략해야 합니다. 버전 번호는 지정할 수 없습니다.

```
aws ec2 describe-launch-template-versions \
```

```
    --versions "$Latest,$Default"
```

PowerShell

시작 템플릿 버전을 설명하려면

Get-EC2TemplateVersion Cmdlet을 사용하고 버전 번호를 지정합니다. 다음 예에서는 버전 *1*과 *3*이 지정됩니다.

```
Get-EC2TemplateVersion `
    -LaunchTemplateId lt-0abcd290751193123 `
    -Version 1,3
```

계정의 최신 및 기본 시작 템플릿 버전을 설명하려면

Get-EC2TemplateVersion Cmdlet을 사용하고 `$Latest`, `$Default` 또는 둘 다 지정합니다. 호출에서 시작 템플릿 ID와 이름을 생략해야 합니다. 버전 번호는 지정할 수 없습니다.

```
Get-EC2TemplateVersion `
    -Version '$Latest','$Default'
```

시작 템플릿 또는 시작 템플릿 버전 삭제

시작 템플릿이 더 이상 필요하지 않으면 이를 삭제할 수 있습니다. 시작 템플릿을 삭제하면 모든 버전이 삭제됩니다. 시작 템플릿의 특정 버전만 삭제하려는 경우 시작 템플릿의 다른 버전을 유지하면서 삭제할 수 있습니다.

시작 템플릿이나 시작 템플릿 버전을 삭제해도 시작 템플릿에서 시작한 인스턴스에는 영향이 없습니다.

시작 템플릿 및 모든 시작 템플릿 버전 삭제

모든 버전을 포함하여 시작 템플릿이 더 이상 필요하지 않은 경우 시작 템플릿을 삭제할 수 있습니다. 시작 템플릿을 삭제하면 모든 버전이 삭제됩니다.

Console

시작 템플릿과 모든 시작 템플릿 버전을 삭제하려면 다음을 수행하세요.

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. **Delete**를 입력하여 삭제를 확인한 다음 삭제를 선택합니다.

AWS CLI

시작 템플릿과 모든 시작 템플릿 버전을 삭제하려면 다음을 수행하세요.

delete-launch-template 명령을 사용하여 시작 템플릿을 지정합니다.

```
aws ec2 delete-launch-template --launch-template-id lt-01238c059e3466abc
```

PowerShell

시작 템플릿과 모든 시작 템플릿 버전을 삭제하려면 다음을 수행하세요.

Remove-EC2LaunchTemplate(AWS Tools for PowerShell) 명령을 사용하여 시작 템플릿을 지정합니다. `-Force`가 생략되면 PowerShell이 확인 메시지를 표시합니다.

```
Remove-EC2LaunchTemplate -LaunchTemplateId lt-0123456789example -Force
```

시작 템플릿 버전 삭제

시작 템플릿 버전이 더 이상 필요하지 않으면 이를 삭제할 수 있습니다.

고려 사항

- 삭제한 후에는 버전 번호를 바꿀 수 없습니다.
- 시작 템플릿의 기본 버전은 삭제할 수 없습니다. 먼저 다른 버전을 기본으로 지정해야 합니다. 기본 버전이 시작 템플릿의 유일한 버전인 경우 전체 시작 템플릿을 삭제해야 합니다.
- 콘솔을 사용하는 경우 한 번에 하나의 시작 템플릿 버전을 삭제할 수 있습니다. AWS CLI을(를) 사용하는 경우 한 번의 요청으로 최대 200개의 시작 템플릿 버전을 삭제할 수 있습니다. 한 번의 요청으로 200개 이상의 버전을 삭제하려면 시작 템플릿을 삭제하여 모든 버전을 삭제할 수도 있습니다.

Console

시작 템플릿 버전을 삭제하려면

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 삭제를 선택합니다.

AWS CLI

시작 템플릿 버전을 삭제하려면

delete-launch-template-versions 명령을 사용하여 삭제할 버전 번호를 지정합니다. 한 번의 요청으로 최대 200개의 시작 템플릿 버전을 삭제하도록 지정할 수 있습니다.

```
aws ec2 delete-launch-template-versions \
    --launch-template-id lt-0abcd290751193123 \
    --versions 1
```

PowerShell

시작 템플릿 버전 삭제

Remove-EC2TemplateVersion cmdlet을 사용하고 삭제할 버전 번호를 지정합니다. 한 번의 요청으로 최대 200개의 시작 템플릿 버전을 삭제하도록 지정할 수 있습니다.

```
Remove-EC2TemplateVersion `
    -LaunchTemplateId lt-0abcd290751193123 `
    -Version 1
```

Amazon EC2 인스턴스 시작하기

인스턴스는 AWS 클라우드의 가상 서버입니다. 인스턴스는 Amazon Machine Image(AMI)에서 시작됩니다. AMI는 운영 체제와 애플리케이션 서버, 그리고 인스턴스 사용을 위한 애플리케이션을 제공합니다.

AWS에 가입하면 AWS 프리 티어를 사용하여 Amazon EC2를 무료로 시작할 수 있습니다. 프리 티어를 사용하여 12개월 동안 무료로 `t2.micro` 인스턴스를 시작하고 사용할 수 있습니다(`t2.micro`를

사용할 수 없는 리전에서는 프리 티어에서 `t3.micro` 인스턴스를 사용할 수 있음). 인스턴스가 실행되는 동안에는 유휴 상태를 유지하더라도 해당 인스턴스 또는 프리 티어 한도에 포함되는 사용량에 대한 요금이 발생합니다. 자세한 설명은 Amazon EC2 요금을 참조하세요.

인스턴스를 시작할 때 다음 리소스 중 하나에 연결된 서브넷에서 인스턴스를 시작할 수 있습니다.

- 가용 영역 - 이 옵션이 기본값입니다.
- 로컬 영역 - 로컬 영역에서 인스턴스를 시작하려면 로컬 영역을 옵트인한 다음 로컬 영역에 서브넷을 생성해야 합니다. 자세한 내용은 로컬 영역 시작하기를 참조하세요.
- Wavelength 영역 - Wavelength 영역에서 인스턴스를 시작하려면 Wavelength 영역을 옵트인한 다음 Wavelength 영역에 서브넷을 생성해야 합니다. 파장 영역에서 인스턴스를 시작하는 방법에 대한 자세한 내용은 AWS Wavelength 시작하기를 참조하세요.
- Outposts - Outposts에서 인스턴스를 시작하려면 Outposts를 생성해야 합니다. Outpost를 생성하는 방법에 대한 자세한 내용은 AWS Outposts 시작하기를 참조하세요.

인스턴스 시작한 다음 인스턴스를 연결하여 사용할 수 있습니다. 인스턴스는 `pending` 상태로 시작됩니다. 인스턴스 부팅이 시작되면 인스턴스의 상태가 `running`로 변경됩니다. 인스턴스 연결이 가능해질 때까지 약간의 시간이 걸릴 수 있습니다. 베어 메탈 인스턴스 유형을 시작하는 데 더 오래 걸릴 수 있습니다.

인스턴스 연결 계획에 따라 인스턴스를 시작하는 동안 특정 구성을 지정해야 할 수 있습니다. 이러한 구성에는 특정 트래픽에 대한 인바운드 보안 그룹 규칙을 지정하거나 인스턴스 프로파일 역할을 연결하는 것이 포함될 수 있습니다. 연결에 사용할 수 있는 연결 방법 및 해당 요구 사항에 대한 자세한 내용은 EC2 인스턴스에 연결 섹션을 참조하세요.

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 작업을 완료한 후 불필요한 비용이 발생하지 않도록 인스턴스를 반드시 종료합니다. 자세한 내용은 Amazon EC2 인스턴스 종료 섹션을 참조하세요.

다음은 인스턴스를 시작할 수 있는 몇 가지 방법입니다.

메서드	도구	설명서
인스턴스 시작 마법사를 사용하여 시작 파라미터를 지정합니다.	Amazon EC2 콘솔	콘솔에서 인스턴스 시작 마법사를 사용하여 EC2 인스턴스 시작
시작 템플릿을 생성하고 이 시작 템플릿에서 인스턴스를 시작합니다.	Amazon EC2 콘솔	시작 템플릿을 사용하여 EC2 인스턴스 시작
기존 인스턴스를 기본 템플릿으로 사용합니다.	Amazon EC2 콘솔	기존 인스턴스의 세부 정보를 사용하여 EC2 인스턴스 시작
AWS Marketplace에서 구매한 AMI를 사용합니다.	Amazon EC2 콘솔	AWS Marketplace AMI에서 Amazon EC2 인스턴스 시작
지정한 AMI 사용	AWS CLI	AWS CLI에서 Amazon EC2 인스턴스 시작, 나열, 삭제
지정하는 AMI를 사용합니다.	AWS Tools for Windows PowerShell	Windows PowerShell을 사용하여 Amazon EC2 인스턴스 시작
EC2 플릿을 사용하여 여러 EC2 인스턴스 유형 및 가용 영역과 온디맨드 인스턴스, 예약 인스턴스 및 스팟 인스턴스 구매 옵션에 걸쳐 용량을 프로비저닝합니다.	AWS CLI	EC2 플릿 및 스팟 플릿
AWS CloudFormation 템플릿을 사용하여 인스턴스를 지정합니다.	AWS CloudFormation	AWS CloudFormation 사용 설명서의 AWS::EC2::Instance
언어별 AWS SDK를 사용하여 인스턴스를 시작합니다.	AWS SDK	AWS SDK for .NET AWS SDK for C++ AWS SDK for Go

메서드	도구	설명서
		AWS SDK for Java AWS SDK for JavaScript AWS SDK for PHP V3 AWS SDK for Python AWS SDK for Ruby V3

EC2 인스턴스 시작을 위한 자습서

Amazon EC2 인스턴스를 시작하고 구성하는 방법에는 여러 가지가 있습니다. 사용 방법과 구성은 특정 사용 사례에 따라 다릅니다.

다음 자습서를 통해 EC2 인스턴스를 시작하는 방법을 알아볼 수 있습니다. Amazon EC2를 처음 사용하는 경우에는 첫 번째 자습서부터 시작하는 것이 좋습니다. 자습서는 기본 사항 소개부터 시작하여 더 많은 구성 옵션을 소개하고 기본적인 사항을 익히는 데 도움을 줍니다.

목표	튜토리얼 링크
첫 번째 EC2 인스턴스 시작하기 Amazon EC2 인스턴스 시작 마법사의 기본 설정을 사용하여 Amazon EC2 인스턴스를 빠르게 시작하는 방법을 알아보세요. 또한 인스턴스 구성 필드를 검토하고 인스턴스를 종료하는 방법도 알아보세요. 소요 시간: 10분	자습서 1: 첫 번째 Amazon EC2 인스턴스 시작하기
테스트 EC2 인스턴스 시작 및 연결하기 테스트 목적으로 사용할 수 있는 Amazon EC2 인스턴스를 시작하는 방법을 알아보세요. 이 인스턴스에는 고급 구성이 없으며 중요한 정보를 저장하지 않습니다. 또한 필수적인 인스턴스 구	자습서 2: 테스트 EC2 인스턴스 시작 및 연결하기

목표	튜토리얼 링크
성 설정, 인스턴스에 연결하는 방법, 인스턴스를 중지하는 방법도 알아보세요. 소요 시간: 30분	

다른 자습서를 찾고 있나요?

- 자습서: AL2023에 LAMP 서버 설치
- 자습서: AL2023에서 SSL/TLS 구성
- 자습서: AL2023에서 WordPress 블로그 호스트
- 자습서: EC2 Instance Connect를 사용하여 인스턴스에 연결하는 데 필요한 구성 완료
- 자습서: Amazon RDS 데이터베이스에 Amazon EC2 인스턴스 연결

자습서 1: 첫 번째 Amazon EC2 인스턴스 시작하기

자습서 목표	Amazon EC2 인스턴스 시작 마법사의 기본 설정을 사용하여 Amazon EC2 인스턴스를 빠르게 시작하는 방법을 알아보세요. 또한 인스턴스 구성 필드를 검토하고 인스턴스를 종료하는 방법도 알아보세요.
EC2 경험	초보자
지속 시간	10분
비용	프리 티어 사용 가능 AWS에 가입하면 AWS 프리 티어를 사용하여 Amazon EC2를 시작할 수 있습니다. AWS 계정을 생성한 지 12개월 미만이고 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. 그렇게 하지 않으면 인스턴스를 시

	작한 시점부터 유휴 상태로 유지되는 경우에도 종료할 때까지 표준 Amazon EC2 사용료가 부과됩니다. 프리 티어 이용 자격 여부를 결정하는 지침은 the section called “프리 티어 사용량 추적” 섹션을 참조하세요.
사전 조건	• AWS 계정이 있어야 하고 관리자 액세스 권한이 있는 사용자를 구성한 다음 관리자 사용자를 사용하여 AWS Management Console에 로그인해야 합니다. 어떻게 해야 할지 잘 모르는 경우 AWS 환경 설정 자습서를 사용해 보세요. • AWS 콘솔에 대해 전반적으로 잘 알고 있어야 합니다. 어디서 시작해야 할지 잘 모르는 경우 AWS Management Console 시작하기 시작 가이드를 사용해 보세요.

자습서 개요

이 자습서는 Amazon EC2 사용 경험이 없는 초보자를 위해 고안되었습니다. EC2 콘솔을 사용하여 첫 번째 EC2 인스턴스를 생성하는 단계를 알려드리겠습니다. 이를 시작이라고 합니다. 인스턴스는 기본적으로 AWS 클라우드의 웹 서버입니다. 인스턴스를 시작한 후 콘솔에서 인스턴스를 찾는 방법을 보여드리겠습니다. 마지막으로 비용을 관리하는 데 도움이 되도록 인스턴스를 삭제하는 방법을 알려드리겠습니다. 이를 종료라고 합니다.

이 자습서는 다음과 같은 짧은 작업으로 나누어져 있습니다. 다음 작업으로 이동하기 전에 각 작업을 완료해야 합니다.

- 작업 1: 인스턴스 시작
- 작업 2: 인스턴스 찾기
- 작업 3: 인스턴스 구성 보기
- 작업 4: 인스턴스 종료

작업 1: 인스턴스 시작

이 작업에서는 필수 항목만 수행하여 인스턴스를 시작하는 가장 빠른 경로를 알려드립니다. 인스턴스 구성 및 시작을 위한 모든 필드를 제공하는 웹 기반 양식인 EC2 인스턴스 시작 마법사를 사용하겠습니다. 인스턴스 구성 필드에 기본값을 제공하여 프로세스를 간소화합니다.

시작하기 전에

관리자 사용자로 AWS Management Console에 로그인하는 것을 포함하여 이전 표에 나열되어 있는 사전 조건을 완료했는지 확인하세요.

인스턴스를 빠르게 시작하려면 다음 단계를 따르세요.

1. Amazon EC2 콘솔을 엽니다.

   https://console.aws.amazon.com/ec2/으로 이동합니다.
2. EC2 인스턴스 시작 마법사 열기:

   EC2 대시보드에서 인스턴스 시작을 선택합니다.

   인스턴스 시작 웹 기반 양식이 열립니다. 이것은 EC2 시작 인스턴스 마법사입니다.
3. 인스턴스 이름 지정:

   이름 및 태그의 이름에 설명이 포함된 이름(예: **`My first EC2 instance`**)을 입력합니다.

   인스턴스 이름 지정이 필수는 아니지만 나중에 인스턴스를 식별하는 데 도움이 됩니다.
4. 키 페어 없이 진행:

   키 페어(로그인) 아래 키 페어 이름에서 없이 진행(권장되지 않음)을 선택합니다.

   보안 로그인에 키 페어를 사용할 수 있습니다. 그러나 이 자습서에서는 인스턴스에 로그인하지 않으므로 지금은 키 페어가 필요하지 않습니다.
5. 인스턴스 시작:

   오른쪽의 요약(Summary) 패널에서 인스턴스 시작(Launch instance)을 선택합니다.

   Amazon EC2는 기본 설정을 사용하여 인스턴스를 빠르게 시작합니다. 성공 배너가 시작을 확인합니다.

축하합니다! 첫 번째 EC2 인스턴스를 성공적으로 시작했습니다.

작업 2: 인스턴스 찾기

이 작업에서는 EC2 콘솔에서 방금 시작한 인스턴스를 찾습니다.

EC2 콘솔에서 인스턴스를 찾으려면 다음 단계를 따르세요.

1. 인스턴스 페이지 열기:

   여전히 성공 페이지에 있는 경우 화면 상단의 브레드크럼에서 인스턴스를 선택합니다. 액세스하려면 먼저 3개의 타원을 선택해야 할 수도 있습니다.

   다른 곳으로 이동한 경우 탐색 창에서 인스턴스를 선택합니다.
2. 인스턴스 찾기:

   이름 열에서 사용자가 지정한 이름으로 인스턴스를 찾습니다.

작업 3: 인스턴스 구성 보기

이 작업에서는 인스턴스의 구성 세부 정보를 보는 방법을 알아봅니다.

인스턴스의 구성을 보려면 다음 단계를 따르세요.

1. 인스턴스 ID 찾기:

   인스턴스 ID 열에서 인스턴스의 고유 ID를 기록합니다. i–로 시작하고 그 뒤에 17개의 영숫자가 이어집니다(예: i-01aeed690c9fb5322).

   인스턴스 ID는 인스턴스가 시작될 때 자동으로 할당됩니다.
2. 인스턴스 세부 정보 페이지 열기:

   인스턴스 ID 열에서 ID 링크를 선택하여 구성을 검토할 수 있는 인스턴스 세부 정보 페이지를 엽니다.
3. 인스턴스 구성 세부 정보 탐색:

   잠시 시간을 내어 인스턴스의 구성 세부 정보를 살펴보세요. 다음 자습서에서는 구성에 대해 자세히 알아보겠습니다. 지금은 이 시간을 활용하여 인스턴스 세부 정보 페이지에 익숙해지세요.

   팁: 필드를 빠르게 찾으려면 키보드에서 Ctrl+F 또는 command+F를 누릅니다.

   a. 인스턴스 유형: 인스턴스 유형을 찾을 수 있나요? t2.micro 또는 t3.micro입니다.

b. 퍼블릭 IPv4 주소: 인스턴스에 할당된 퍼블릭 IPv4 주소를 찾을 수 있나요? 34.242.148.128과 유사한 형식입니다.

c. 인스턴스 소유자: 이 인스턴스의 소유자를 식별할 수 있나요? 바로 사용자입니다. 사용자의 AWS 계정 번호는 소유자 필드 아래에 나열되어 있습니다.

d. 인스턴스 태그: 인스턴스에 부여한 이름은 실제로는 태그입니다. 인스턴스 태그를 찾을 수 있나요? 태그 탭을 선택합니다. 키는 이름이고 값은 입력한 이름입니다.

e. 시작 시간: 인스턴스를 언제 시작했는지 알 수 있나요? 세부 정보 탭을 선택하고 시작 시간 필드를 찾습니다.

f. 인스턴스 상태: 인스턴스의 상태를 확인할 수 있나요? 실행 중이어야 합니다.

몇 분 더 시간을 내어 다른 인스턴스 구성 필드를 살펴보세요. 준비가 되면 다음 작업을 진행합니다.

작업 4: 인스턴스 종료

이 작업에서는 프리 티어 혜택을 유지하기 위해 인스턴스를 삭제합니다. EC2에서 종료는 인스턴스를 삭제하는 데 사용되는 용어입니다.

인스턴스를 종료하려면 다음 단계를 따르세요.

1. 종료 시작:

   여전히 인스턴스 세부 정보 페이지에 있는 경우 인스턴스 상태 메뉴(오른쪽 상단)를 선택한 다음 인스턴스 종료(삭제)를 선택합니다.

   다른 곳으로 이동한 경우 탐색 창에서 인스턴스를 선택합니다. 그런 다음 인스턴스 페이지에서 인스턴스 이름 옆의 확인란을 선택한 다음 인스턴스 상태 메뉴(오른쪽 상단)를 선택하고 인스턴스 종료(삭제)를 선택합니다.

2. 종료 확인:

   열리는 인스턴스 종료(삭제) 창에서 종료(삭제) 버튼을 선택하여 인스턴스를 종료할 것임을 확인합니다.

3. 인스턴스 상태 모니터링:

   인스턴스 페이지에서 인스턴스 상태 열을 확인합니다. 인스턴스 상태가 종료 중으로 변경됩니다. 전체 텍스트가 보이지 않으면 열을 넓혀 보세요.

   인스턴스가 종료되면 Amazon EC2가 인스턴스를 삭제하고 인스턴스 페이지에서 사라집니다.

핵심 고려 사항

이 자습서에서는 다음 주요 개념을 다루었습니다.

- 인스턴스는 AWS 클라우드의 Amazon EC2 웹 서버를 의미합니다.
- 시작은 EC2 인스턴스 생성을 의미합니다.
- 종료는 EC2 인스턴스 삭제를 의미합니다.
- EC2 인스턴스 시작 마법사에는 인스턴스 구성을 위한 기본값이 포함되어 있어 인스턴스를 빠르고 간편하게 시작할 수 있습니다.
- 인스턴스 ID는 인스턴스에 자동으로 할당되는 고유 식별자이고 인스턴스 이름은 더 쉽게 식별할 수 있도록 할당할 수 있는 태그(선택 사항)입니다.

다음 단계

인스턴스 시작 및 종료에 대한 자신감을 키우려면 이 자습서의 단계를 반복해 보세요. 프리 티어 혜택을 유지하려면 시작한 인스턴스를 모두 종료해야 합니다.

이러한 기본 사항에 익숙해지면 다음 자습서로 이동하여 주요 인스턴스 구성 필드에 대해 자세히 알아보세요.

자습서 2: 테스트 EC2 인스턴스 시작 및 연결하기

자습서 목표	테스트 목적으로 사용할 수 있는 Amazon EC2 인스턴스를 시작하는 방법을 알아보세요. 이 인스턴스에는 고급 구성이 없으며 중요한 정보를 저장하지 않습니다. 또한 필수적인 인스턴스 구성 설정, 인스턴스에 연결하는 방법, 인스턴스를 중지하는 방법도 알아보세요.
EC2 경험	초보자
지속 시간	30 분
비용	프리 티어 사용 가능 AWS에 가입하면 AWS 프리 티어를 사용하여 Amazon EC2를 시작할 수 있습니다. AWS 계정을 생성한 지 12개월 미만이고 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. 그렇게 하지 않으면 인스턴스를 시작한 시점부터 유휴 상태로 유지되는 경우에도 종료할 때까지 표준 Amazon EC2 사용료가 부과됩니다. 프리 티어 이용 자격 여부를 결정하는 지침은 the section called “프리 티어 사용량 추적” 섹션을 참조하세요.
사전 조건	자습서 1: 첫 번째 Amazon EC2 인스턴스 시작하기을 완료합니다.

자습서 개요

이 자습서는 테스트 목적으로 사용할 수 있는 EC2 인스턴스를 시작하려는 초보자를 위해 고안되었습니다.

주요 인스턴스 구성 필드에 대해 설명한 다음 EC2 콘솔에서 기본값을 사용하여 테스트 인스턴스를 시작하는 단계를 알려드리겠습니다. 인스턴스를 시작한 후 인스턴스에 로그인하는 방법을 보여드리겠습니다. 이를 연결이라고 합니다. 이 자습서에서는 또한 인스턴스에 연결하는 데 필요한 키 페어를 생성하는 방법도 보여드리겠습니다. 마지막으로 비용 관리를 지원하기 위해 사용 요금이 부과되지 않도록 인스턴스를 중지하는 방법을 알려드립니다.

이 자습서에서는 Linux 인스턴스를 시작합니다. 이 자습서의 단계는 다른 운영 체제에서 인스턴스를 시작하는 데 사용할 수 있지만 인스턴스에 연결하는 지침은 Linux 인스턴스에만 해당됩니다.

이 자습서는 다음과 같은 짧은 작업으로 나누어져 있습니다. 다음 작업으로 이동하기 전에 각 작업을 완료해야 합니다.

- 작업 1: 인스턴스 시작을 위한 주요 구성 요소 숙지하기
- 작업 2: 기술 다이어그램 검토
- 작업 3: 키 페어 생성
- 작업 4: 테스트 인스턴스 시작
- 작업 5: 인스턴스 찾기

- 작업 6: 인스턴스 구성 보기
- 작업 7: 인스턴스에 연결하기 위한 주요 구성 요소 숙지
- 작업 8: 인스턴스에 연결
- 작업 9: 인스턴스 중지

작업 1: 인스턴스 시작을 위한 주요 구성 요소 숙지하기

이 작업에서는 EC2 인스턴스를 시작하는 데 필요한 주요 구성 요소를 살펴봅니다. AMI, 인스턴스 유형, 키 페어, 보안 그룹, 네트워크(VPC 및 서브넷), Amazon EBS 볼륨이 여기에 해당합니다. 또한 선택적 구성 요소인 이름 태그를 살펴봅니다.

이러한 구성 요소를 시각화하는 데 도움이 되도록 인스턴스를 임대 주택처럼 생각하세요. 주택을 임대하면 부동산을 소유하고 유지 관리할 필요 없이 거주할 수 있는 공간이 제공되는 것처럼, EC2 인스턴스는 기본 인프라를 소유하고 유지 관리할 필요 없이 컴퓨팅 성능을 제공합니다.

시작할 인스턴스의 유형을 결정할 때는 주택에서 원하는 기준을 고려하는 것처럼 인스턴스의 구성 기준을 고려하게 됩니다. 이러한 비유는 대상을 단순화하지만 구성 요소에 더 익숙해질 때까지 이를 시각화하는 데 유용한 방법입니다.

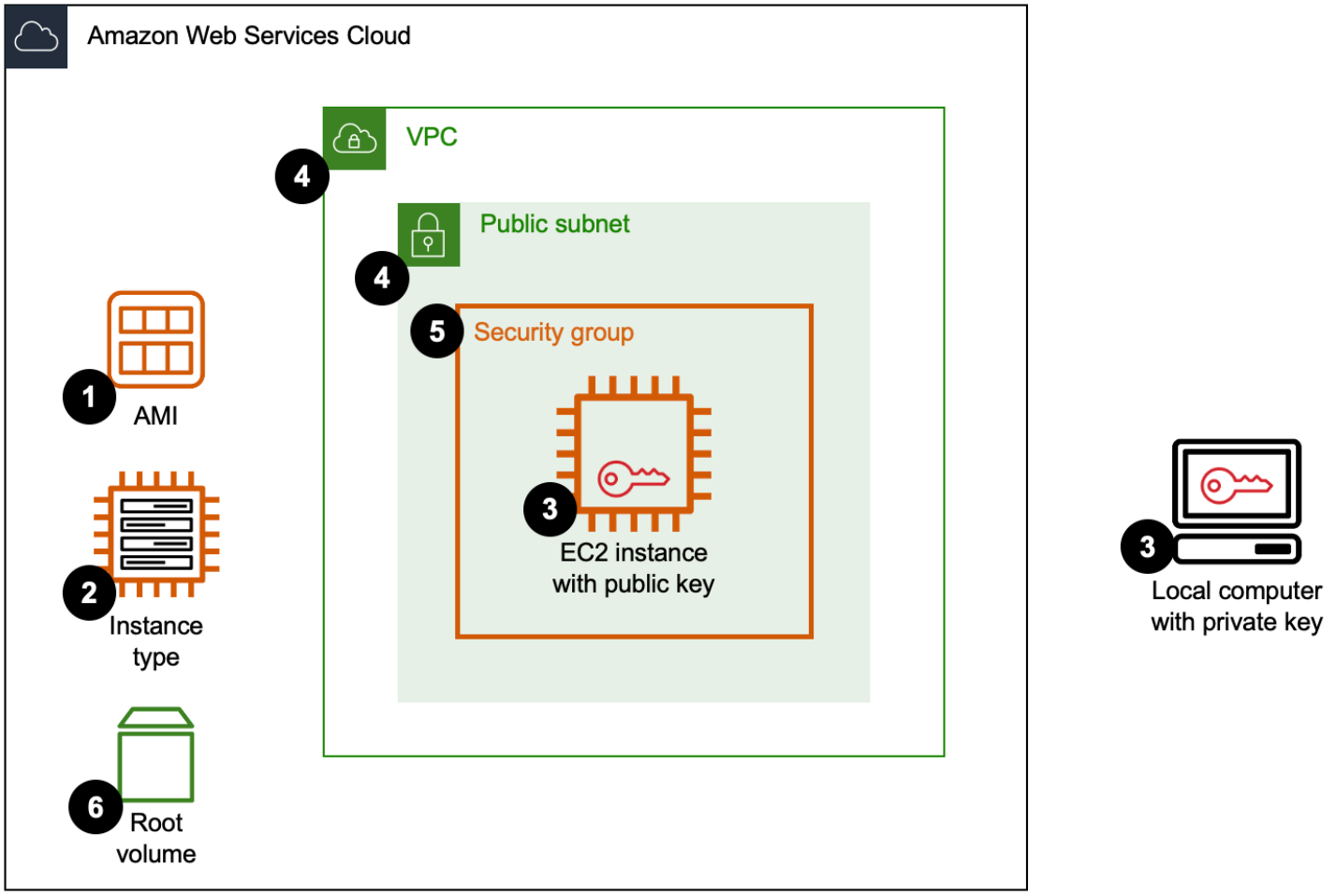
- AMI - 주택 건축 자재 및 편의 시설: Amazon Machine Image(AMI)는 인스턴스가 시작하는 운영 체제와 애플리케이션을 결정합니다. 이는 벽돌, 강철, 목재와 같은 건축 자재와 가전제품 및 가구와 같은 편의 시설을 선택하는 것과 같습니다. 기본 AMI는 기본 가전제품만 있는 미완성 주택과 같고, 소프트웨어가 사전 설치된 맞춤형 AMI는 모든 가구가 갖추어진 주택과 같습니다.
- 인스턴스 유형 - 주택 크기 및 전원: 인스턴스 유형은 주택의 크기, 방의 개수, 에너지 용량을 선택하는 것과 마찬가지로 EC2 인스턴스의 크기와 기능을 정의합니다. 각 인스턴스 유형에 따라 인스턴스의 CPU, 메모리, 스토리지, 네트워킹 용량이 결정됩니다. 선택한 AMI에 따라 선택할 수 있는 인스턴스 유형이 제한될 수 있습니다.
- 키 페어 - 프런트 도어 키: 키 페어는 주택 앞문의 잠금 장치 및 열쇠와 같습니다. 퍼블릭 키는 인스턴스의 잠금 장치 역할을 하는 반면, 프라이빗 키는 로컬 컴퓨터에 안전하게 보관해야 하는 키입니다. 현관문 열쇠를 가지고 있는 사람이 집에 들어올 수 있는 것처럼 다른 사람이 프라이빗 키를 가지고 있으면 인스턴스에 액세스할 수 있습니다.
- 네트워크(VPC 및 서브넷) - 부동산 경계, 섹션화된 영역, 주택 번호: 가상 프라이빗 클라우드(VPC)는 주택이 있는 전체 부동산과 유사하며 서브넷은 주택 주변의 구획화된 영역과 같습니다. 부동산에 여러 개의 주택(인스턴스)이 있는 경우 목적에 따라 각 주택을 별개의 영역(서로 다른 서브넷)으로 구분할 수 있습니다. 방문객이 정원을 자유롭게 돌아다닐 수 있는 주택(인터넷 접속이 가능한 퍼블릭 서브넷)이 있는 반면, 정원에 울타리를 쳐서 출입을 제한하는 주택(인터넷 접속이 불가능한 프라이빗

서브넷)도 있습니다. 각 서브넷에는 주택 번호와 마찬가지로 서브넷 내 인스턴스에 할당할 수 있는 다양한 IP 주소가 포함되어 있습니다.

- 보안 그룹 - 게이트키퍼: 보안 그룹은 게이트키퍼 역할을 하여 누가 주택을 방문할 수 있는지 제어합니다. 인스턴스에 도달할 수 있는 트래픽을 제어하는 일련의 규칙을 적용합니다. 예를 들어 특정 IP 주소로부터의 SSH 트래픽을 허용하는 규칙은 게이트키퍼가 특정 사람만 식료품을 배달할 수 있도록 허용하는 것과 같습니다. 마찬가지로 어디서나 HTTPS 트래픽을 허용하는 것은 일반 대중이 주택 외부를 살펴보도록 하는 것과 같습니다.
- Amazon EBS 볼륨 - 스토리지 단위: EBS 볼륨은 소지품을 저장할 수 있는 저장 장치와 같습니다. 각 인스턴스에는 루트 볼륨(AMI가 저장되는 장소)이 있으며 필요에 따라 언제든지 볼륨(스토리지)을 추가할 수 있습니다.
- 이름 태그 - 주택 이름: 이름 태그는 주택의 표지판과 같은 기능을 하므로 주택에 거주하는 사람을 간편하게 식별할 수 있습니다. 이름 태그를 사용하면 인스턴스를 더 간편하게 구분할 수 있지만 인스턴스를 시작할 때 반드시 필요한 것은 아닙니다.

작업 2: 기술 다이어그램 검토

이 작업에서는 AWS 설명서에서 사용하는 일반적인 기술 다이어그램에 대해 알아봅니다. 다음 다이어그램은 이 자습서에서 시작할 테스트 인스턴스의 구성을 나타냅니다. 이전 작업에서는 임대 주택의 비유를 사용하여 이러한 구성 요소를 소개했습니다. 이제 실제 EC2 구성 요소 자체에 대해 초점을 맞추겠습니다. 번호가 매겨진 레이블은 다음에 나오는 설명에 해당합니다.



1. AMI - AMI는 인스턴스를 시작할 때 선택하는 이미지입니다. 인스턴스에서 실행할 운영 체제와 소프트웨어가 포함된 템플릿입니다. 예를 들어 Linux 인스턴스를 시작하려는 경우 Amazon Linux 2023 AMI를 선택할 수 있습니다. 또는 Windows 인스턴스를 시작하려는 경우 Microsoft Windows Server 2022 기본 AMI를 선택할 수 있습니다. Amazon EC2 콘솔의 AMI 카탈로그에는 선택할 수 있는 수천 개의 이미지가 포함되어 있습니다.
2. 인스턴스 유형 - 인스턴스 유형은 인스턴스에 사용되는 호스트 컴퓨터의 CPU, 메모리, 스토리지, 네트워킹 용량을 결정하는 하드웨어입니다. Amazon EC2는 하드웨어 구성과 크기가 각각 다른 600개 이상의 인스턴스 유형 중에서 선택할 수 있도록 하므로 애플리케이션의 요구 사항에 가장 적합한 것을 선택할 수 있습니다.
3. 키 페어 - 키 페어는 인스턴스에 연결할 때 ID를 증명하는 데 사용하는 보안 자격 증명 세트입니다. 퍼블릭 키는 인스턴스에 있고 프라이빗 키는 로컬 컴퓨터에 있습니다.

   EC2에서 인스턴스에 연결한다는 것은 로컬 컴퓨터에서 인스턴스에 로그인하는 것을 의미합니다. 인스턴스에 안전하게 연결하는 다른 방법도 있지만 이 자습서에서는 키 페어를 사용합니다.

4. 네트워크 - 네트워크는 VPC와 하나 이상의 서브넷으로 구성됩니다. VPC는 AWS 클라우드 내의 가상 네트워크입니다. 모든 AWS 고객에게는 AWS 계정에 대한 전용 VPC가 있습니다. 인스턴스를 VPC의 서브넷으로 시작합니다. 서브넷은 VPC 내 IP 주소의 범위입니다. 기본 서브넷은 퍼블릭 서브넷으로, 이는 퍼블릭 IP 주소를 할당하고 Amazon 네트워크 외부에서 인스턴스에 인터넷 액세스를 제공한다는 것을 의미입니다.
5. 보안 그룹 - 보안 그룹은 인스턴스에 대한 트래픽을 제어하는 방화벽 역할을 합니다. 보안 그룹에는 특정 유형의 트래픽이 인스턴스에 들어오는 것을 허용하는 규칙이 포함되어 있습니다. 키 페어를 사용하여 로컬 컴퓨터에서 인스턴스로 SSH를 통해 연결하려면 로컬 컴퓨터에서 SSH 트래픽을 허용하는 규칙이 필요합니다.
6. EBS 볼륨 - Amazon EBS 볼륨은 실제 하드 드라이브처럼 작동하는 스토리지 디바이스입니다. 인스턴스에는 인스턴스를 부팅하는 데 필요한 운영 체제 및 소프트웨어와 함께 AMI를 저장하는 특수 EBS 볼륨인 루트 볼륨이 함께 제공됩니다. 필요에 따라 데이터 볼륨을 추가할 수 있습니다. 그러나 테스트 인스턴스에는 중요한 데이터가 저장되지 않으므로 추가로 암호화된 데이터 볼륨이 필요하지 않습니다.

축하합니다! 이 자습서의 개념 관련 작업을 완료했습니다. 다음 작업에서는 Amazon EC2 콘솔을 사용하여 이전에 학습한 구성 요소를 생성합니다.

작업 3: 키 페어 생성

이 작업에서는 키 페어를 생성합니다. 키 페어는 인스턴스에 추가할 퍼블릭 키와 인스턴스에 안전하게 연결하는 데 사용할 일치하는 프라이빗 키의 두 부분으로 구성됩니다. 다음 작업에서 인스턴스를 시작할 때 이 키 페어를 선택하면 인스턴스에 퍼블릭 키가 자동으로 추가됩니다. 프라이빗 키에 액세스할 수 있는 사람은 누구나 인스턴스에 연결할 수 있으므로 프라이빗 키를 로컬 컴퓨터에 안전하게 저장하는 것이 중요합니다.

테스트 인스턴스를 시작할 때 기존 키 페어를 사용하는 것을 선호하는 경우에는 이 작업을 건너뛰셔도 됩니다. 그렇게 하지 않으면 계속 진행하여 새 키 페어를 생성합니다.

시작하기 전에

관리자 사용자로 AWS Management Console에 로그인하는 것을 포함하여 이전 표에 나열되어 있는 사전 조건을 완료했는지 확인하세요.

키 페어를 생성하려면 다음 단계를 따르세요.

1. Amazon EC2 콘솔을 엽니다.

   https://console.aws.amazon.com/ec2/으로 이동합니다.

2. 키 페어 콘솔 페이지로 이동합니다.

   탐색 창의 네트워크 및 보안에서 키 페어를 선택합니다.

   - 이전에 키 페어를 생성한 경우 해당 키 페어가 테이블에 표시됩니다.
   - 키 페어가 없는 경우에는 테이블이 비어 있습니다.

3. 새 키 페어를 생성합니다.

   키 페어 생성 버튼(오른쪽 상단)을 선택하여 키 페어 생성 웹 기반 양식을 열고 다음과 같이 키 페어 세부 정보를 입력합니다.

   a. 키 페어 이름 지정: 이름에 키 페어를 인식하는 데 도움이 되는 이름(예: **`test-instance-key-pair`**)을 입력합니다.

      이 이름에는 최대 255자 길이의 ASCII 문자를 사용할 수 있습니다. 선행 또는 후행 공백을 포함할 수 없습니다.

   b. 키 페어 유형 선택: 키 페어 유형에서 ED25519를 선택합니다.

      Linux 인스턴스는 RSA 및 ED25519 키 유형을 모두 지원하는 반면, Windows 인스턴스는 RSA만 지원합니다. 이 자습서에서는 Linux 인스턴스를 시작할 것이므로 ED25519 키를 사용할 수 있습니다.

   c. 프라이빗 키 파일 형식 선택: 프라이빗 키 파일 형식의 경우 .pem을 선택합니다.

      이는 프라이빗 키 파일이 저장되는 형식입니다.

4. Amazon EC2에 퍼블릭 키를 저장하고 프라이빗 키를 다운로드합니다.

   키 페어 생성 버튼(오른쪽 하단)을 선택합니다.

   Amazon EC2는 퍼블릭 키를 저장하고 브라우저는 프라이빗 키 파일을 로컬 컴퓨터에 자동으로 다운로드합니다. 파일은 키 페어에 지정한 이름에 따라 이름이 지정되며 확장자는 선택한 파일 형식입니다. 프라이빗 키 파일을 컴퓨터의 안전한 위치로 옮깁니다.

   > **Important**
   > 이때가 사용자가 프라이빗 키 파일을 저장할 수 있는 유일한 기회입니다.

5. 키에 대한 권한 설정(macOS 및 Linux 사용자의 경우):

macOS 또는 Linux 컴퓨터에서 SSH를 사용하여 인스턴스에 연결하려는 경우 프라이빗 키 파일에 대한 올바른 권한을 설정해야 합니다. 터미널 창을 열고 다음 명령을 실행하여 *test-instance-key-pair*를 키 페어의 이름으로 바꿉니다.

```
chmod 400 test-instance-key-pair.pem
```

이 명령은 인스턴스에 대한 보안 연결을 설정하는 데 필요한 프라이빗 키 파일을 본인만 읽을 수 있도록 합니다. 이러한 권한이 없으면 해당 키 페어를 사용하여 연결할 수 없습니다.

축하합니다! 키 페어를 성공적으로 생성했습니다.

작업 4: 테스트 인스턴스 시작

이 작업에서는 EC2 인스턴스 시작 마법사를 사용하여 테스트 인스턴스를 빠르게 시작합니다. Linux 인스턴스에 대한 기본 인스턴스 구성 설정을 구성하고 다른 필드에는 기본값을 사용합니다.

비용 관리에 도움이 되도록 프리 티어 사용 가능 구성 요소를 선택하는 것이 좋습니다.

테스트 인스턴스를 시작하려면 다음 단계를 따르세요.

1. Amazon EC2 콘솔을 엽니다.

   https://console.aws.amazon.com/ec2/으로 이동합니다.
2. EC2 인스턴스 시작 마법사 열기:

   EC2 대시보드에서 인스턴스 시작을 선택합니다.

   인스턴스 시작 웹 기반 양식이 열립니다. 이것은 EC2 시작 인스턴스 마법사입니다.
3. 인스턴스 이름 지정:

   이름 및 태그의 이름에 설명이 포함된 이름(예: **Test instance**)을 입력합니다.

   인스턴스 이름은 태그이며, 여기서 키는 이름이고 값은 사용자가 지정하는 이름입니다.

   팁: 테스트 인스턴스의 경우 이름 태그로도 충분합니다. 그러나 프로덕션 인스턴스의 경우 모든 리소스에서 태그 지정을 표준화하는 태그 지정 정책을 설정하는 것이 좋습니다.
4. 운영 체제 및 소프트웨어인 Amazon Machine Image(AMI)를 선택합니다.

애플리케이션 및 OS 이미지(Amazon Machine Image)에서 Amazon Machine Image(AMI)의 경우 Amazon Linux 2023 AMI가 기본적으로 선택되어 있습니다. 해당 AMI는 프리 티어 사용 가능으로 표시됩니다. 이 자습서에서는 Linux 인스턴스를 실행할 예정이므로 기본 설정은 프리 티어를 그대로 유지합니다.

5. 하드웨어, 즉 인스턴스 유형을 선택합니다.

   인스턴스 유형의 인스턴스 유형에 대해 이 자습서에서는 기본 선택(t2.micro 또는 t3.micro)을 유지합니다. 기본 인스턴스 유형은 프리 티어 사용 가능이며 해당 하드웨어는 테스트 인스턴스에 적합합니다.

6. 키 페어로 보안 로그인을 준비합니다.

   키 페어(로그인)의 키 페어 이름에서 이전 작업에서 생성한 키 페어를 선택합니다. 목록에 키 페어가 보이지 않으면 새로 고침 아이콘(목록의 오른쪽에 있음)을 선택하세요.

   인스턴스가 시작되면 인스턴스에 퍼블릭 키가 배치됩니다. 인스턴스가 시작된 후 인스턴스에 연결하려면 이전 작업에서 다운로드한 해당 프라이빗 키를 사용합니다.

7. 인터넷 액세스를 활성화하도록 네트워크 설정을 구성합니다.

   네트워크 설정에서 네트워크(VPC) 및 서브넷 필드가 기본적으로 구성됩니다. 자습서의 기본 설정을 그대로 유지하면 빠르게 시작할 수 있습니다. 기본 서브넷을 수정하지 않은 경우 인스턴스는 인터넷에 액세스할 수 있습니다.

   팁: 기본 서브넷은 퍼블릭 서브넷으로, 이는 퍼블릭 IP 주소를 할당하고 Amazon 네트워크 외부에서 인스턴스에 인터넷 액세스를 제공한다는 것을 의미입니다. 테스트 인스턴스의 경우 인터넷 액세스를 제공하는 기본 서브넷 설정을 사용해도 됩니다. 그러나 프로덕션 인스턴스의 경우 반드시 필요한 경우에만 퍼블릭 IP 주소를 할당하고 인터넷에 액세스할 수 있는 서브넷을 사용하는 것이 가장 좋습니다.

8. 인스턴스 방화벽 설정(보안 그룹):

   네트워크 설정의 방화벽(보안 그룹)에서 모든 위치에서 SSH 트래픽 허용(0.0.0.0) 확인란을 선택한 상태로 유지합니다. 이렇게 하면 테스트 인스턴스에 대해 모든 IP 주소에서 SSH 트래픽을 허용하는 새 보안 그룹이 생성됩니다.

   보안 그룹은 인스턴스에 대한 트래픽을 제어하는 방화벽 역할을 합니다. 로컬 컴퓨터에서 인스턴스로 SSH를 통해 연결하려면 로컬 컴퓨터에서 SSH 트래픽을 허용하는 규칙이 필요합니다.

   팁: 인터넷 서비스 제공업체가 동적 IP 할당을 사용하는 경우 로컬 컴퓨터의 IP 주소가 시간이 지남에 따라 변경될 수 있습니다. 테스트 목적으로 인스턴스를 사용하는 경우에는 인스턴스를

사용하여 중요한 정보를 저장하지 않으므로 보안 조치가 덜 제한적일 수 있다고 가정합니다. 테스트 인스턴스의 경우 일반적으로 IP 주소가 변경되더라도 항상 연결할 수 있도록 모든 IP 주소(0.0.0.0/0)의 트래픽을 허용해도 됩니다. 그러나 프로덕션 인스턴스, 특히 중요한 데이터가 있는 인스턴스의 경우에는 알려진 IP 주소의 트래픽만 허용하는 것이 좋습니다.

9. 인스턴스 스토리지 구성:

   스토리지 구성에서 루트 볼륨(암호화됨) 필드는 기본적으로 구성되어 있습니다. 프리 티어 사용 가능 상태로 유지하려면 설정을 그대로 둡니다.

   테스트 인스턴스에는 중요한 데이터가 저장되지 않으므로 추가로 암호화된 데이터 볼륨이 필요하지 않습니다.

10. 인스턴스 구성 검토:

    오른쪽의 요약 패널에서 인스턴스를 시작하기 전에 상위 수준의 설정을 검토할 수 있습니다.

11. 인스턴스 시작:

    인스턴스를 시작할 준비가 되면 요약 패널에서 인스턴스 시작을 선택합니다.

    Amazon EC2는 사용자가 지정한 설정을 사용하여 인스턴스를 빠르게 시작합니다. 설정을 지정하지 않은 경우에는 기본값이 사용됩니다. 성공 배너가 시작을 확인합니다.

축하합니다! 테스트 인스턴스를 성공적으로 시작했습니다.

작업 5: 인스턴스 찾기

이 작업에서는 EC2 콘솔에서 방금 시작한 인스턴스를 찾습니다.

EC2 콘솔에서 인스턴스를 찾으려면 다음 단계를 따르세요.

1. 인스턴스 페이지 열기:

   여전히 성공 페이지에 있는 경우 성공 배너에서 인스턴스 ID를 선택합니다.

   다른 곳으로 이동한 경우 탐색 창에서 인스턴스를 선택합니다.

2. 인스턴스 찾기:

   이름 열에서 사용자가 지정한 이름으로 인스턴스를 찾습니다.

작업 6: 인스턴스 구성 보기

이 작업에서는 인스턴스의 구성 세부 정보를 보는 방법을 알아봅니다.

인스턴스의 구성을 보려면 다음 단계를 따르세요.

1. 인스턴스 찾기:

   이름 열에서 사용자가 지정한 이름으로 인스턴스를 찾습니다.

2. 인스턴스 세부 정보 페이지 열기:

   인스턴스 이름 옆의 확인란을 선택한 다음 작업 메뉴(오른쪽 상단)를 선택하고 세부 정보 보기를 선택하여 구성을 검토할 수 있는 인스턴스 세부 정보 페이지를 엽니다.

   이전 자습서에서는 인스턴스의 ID 링크를 선택하여 인스턴스 세부 정보 페이지를 열었습니다. EC2 콘솔에서 작업을 수행하는 방법이 여러 가지라는 점을 알게 될 것입니다.

3. 인스턴스 구성 세부 정보 탐색:

   잠시 시간을 내어 인스턴스의 구성 세부 정보를 살펴보세요.

   팁: 필드를 빠르게 찾으려면 키보드에서 Ctrl+F 또는 command+F를 누릅니다.

   a. AMI: 인스턴스를 시작하는 데 사용한 AMI를 찾을 수 있나요? 세부 정보 탭의 AMI ID 및 AMI 이름에서 정보를 찾을 수 있습니다.
   b. 인스턴스 유형: 인스턴스 유형을 찾을 수 있나요? t2.micro 또는 t3.micro입니다.
   c. 키 페어: 인스턴스를 시작할 때 선택한 키 페어를 찾을 수 있나요? 시작 시 할당된 키 페어에 대해 지정됩니다. 나중에 키 페어를 변경해도 이 값은 변경되지 않습니다.
   d. VPC: VPC의 ID를 찾을 수 있나요? 네트워킹 탭에서 모든 네트워킹 관련 구성 설정을 찾을 수 있습니다. VPC ID는 vpc-1a2b3c4d와 유사한 형식입니다.
   e. 서브넷: 인스턴스를 시작한 서브넷의 ID를 찾을 수 있나요? subnet-1a2b3c4d와 유사한 형식입니다.
   f. 퍼블릭 IPv4 주소: 인스턴스에 할당된 퍼블릭 IPv4 주소를 찾을 수 있나요? 34.242.148.128과 유사한 형식입니다.
   g. 보안 그룹: 모든 위치에서 SSH 트래픽을 허용(0.0.0.0./0)하도록 생성한 인바운드 규칙을 찾을 수 있나요? 보안 탭에서 모든 보안 관련 구성 설정을 찾을 수 있습니다.
   h. 스토리지: 이 인스턴스에 대해 생성된 볼륨을 찾을 수 있나요? 스토리지 탭에서 모든 스토리지 관련 구성 설정을 찾을 수 있습니다.

i. 인스턴스 태그: 인스턴스에 부여한 이름은 실제로는 태그입니다. 인스턴스 태그를 찾을 수 있나요? 태그 탭을 선택합니다. 키는 이름이고 값은 입력한 이름입니다.

j. 인스턴스 상태: 인스턴스의 상태를 확인할 수 있나요? 실행 중이어야 합니다.

몇 분 더 시간을 내어 다른 인스턴스 구성 필드를 살펴보세요. 준비가 되면 다음 작업을 진행합니다.

작업 7: 인스턴스에 연결하기 위한 주요 구성 요소 숙지

이 작업에서는 EC2 인스턴스에 연결하는 데 필요한 주요 구성 요소를 살펴봅니다. 연결 프로토콜, 퍼블릭 DNS, 보안 그룹, 키 페어, 인스턴스 사용자 이름이 여기에 해당합니다.

이러한 구성 요소를 시각화하기 위해 인스턴스 연결을 집에 가는 것처럼 생각해 보세요.

- 연결 프로토콜 - 이동 수단: 집으로 가는 방법을 선택하는 것과 마찬가지로 인스턴스로 이동하는 연결 프로토콜을 선택합니다. 이 자습서에서는 인터넷을 통해 컴퓨터를 인스턴스에 연결하기 위한 보안 터널을 생성하는 보안 쉘(SSH)을 사용합니다.
- 퍼블릭 DNS - 집 주소: 집에 고유한 주소가 있는 것처럼 EC2 인스턴스에는 고유한 퍼블릭 DNS 이름(예: `ec2-18-201-118-201.eu-west-1.compute.amazonaws.com`)이 있습니다. 이 퍼블릭 DNS 이름을 사용하면 SSH가 인스턴스에 직접 연결할 수 있습니다.
- 보안 그룹 - 게이트키퍼: 집에 누가 출입할 수 있는지를 제어하는 게이트키퍼가 있다고 가정해 보겠습니다. 마찬가지로 EC2 인스턴스에는 게이트키퍼와 같은 역할을 하는 보안 그룹이 있어 인스턴스 내부 또는 외부로 허용되는 네트워크 트래픽 유형을 제어합니다. 명시적으로 허용한 트래픽(예: 내 컴퓨터의 IP 주소로부터의 SSH 트래픽)만 들어올 수 있습니다.
- 프라이빗 키 - 현관문 열쇠: 인스턴스를 시작할 때 키 페어를 지정했습니다. 퍼블릭 키는 인스턴스에 배치하고 프라이빗 키는 컴퓨터에 보관합니다. 프라이빗 키는 현관문 열쇠 역할을 하며 프라이빗 키가 없으면 인스턴스에 들어갈 수 없습니다.
- 인스턴스 사용자 이름 - 거주자: 집에 도착하면 신원을 밝히고 거주자임을 증명해야 합니다. 마찬가지로 인스턴스에 연결할 때 사용자 이름을 입력합니다. 각 인스턴스는 운영 체제에 따라 기본 사용자 이름이 다릅니다. 예를 들어 Amazon Linux 인스턴스는 `ec2-user`를 기본 사용자 이름으로 사용합니다.

연결 명령

EC2 인스턴스에 연결하려면 터미널 창에서 다음 명령을 사용합니다.

```
ssh -i "test-instance-key-pair.pem" ec2-user@ec2-18-201-118-201.eu-
west-1.compute.amazonaws.com
```

다음은 해당 명령이 수행하는 작업에 대한 세부 정보입니다.

- `ssh` – 이 명령은 연결 프로토콜을 지정하여 인스턴스에 대한 보안 쉘(SSH) 연결을 시작합니다.
- `-i "test-instance-key-pair.pem"` – `-i` 플래그는 연결을 인증하는 데 필요한 프라이빗 키 파일을 나타냅니다. 이 프라이빗 키 파일은 인스턴스를 시작할 때 지정한 키 페어와 일치해야 합니다. 프라이빗 키 파일이 특정 폴더에 저장되어 있는 경우 파일의 전체 경로를 지정합니다.
- `ec2-user` - 인스턴스에 로그인할 때 사용하는 사용자 이름입니다. Amazon Linux 인스턴스의 경우 기본 사용자 이름은 `ec2-user`입니다. 다른 AMI는 Ubuntu 인스턴스를 위한 ubuntu와 같이 다른 기본 사용자 이름을 사용할 수 있습니다.
- `@` - 이 기호는 인스턴스의 주소에서 사용자 이름을 구분합니다.
- `ec2-18-201-118-201.eu-west-1.compute.amazonaws.com` - 인스턴스의 퍼블릭 주소(퍼블릭 DNS)이며 퍼블릭 IPv4 주소와 AWS 리전을 포함합니다. 인스턴스를 고유하게 식별합니다.

명령을 실행하면 발생하는 일

명령을 실행하면 SSH가 보안 터널을 설정하고 프라이빗 키로 인증합니다. 인스턴스의 보안 그룹이 트래픽을 허용하면 EC2 인스턴스에 액세스할 수 있습니다. 이제 마치 인스턴스 바로 앞에 앉아 있는 것처럼 컴퓨터에서 인스턴스를 제어할 수 있습니다. 로컬 컴퓨터에서와 마찬가지로 명령을 실행하고, 소프트웨어를 설치하고, 파일을 관리할 수 있습니다.

작업 8: 인스턴스에 연결

이 작업에서는 컴퓨터의 SSH 클라이언트를 사용하여 인스턴스에 연결합니다. 이전 작업에서는 집으로 가는 것에 비유하여 인스턴스에 연결하기 위한 구성 요소를 소개했습니다. 이제는 실제 EC2 인스턴스에 연결하는 데 초점을 맞추겠습니다.

인스턴스에 연결하는 방법에는 여러 가지가 있습니다. 연결하는 데 사용하는 방법은 인스턴스의 운영 체제에 따라 다릅니다. Linux 인스턴스를 시작했으므로 로컬 컴퓨터에서 SSH 클라이언트를 사용하게 됩니다.

먼저 컴퓨터에 SSH 클라이언트가 설치되어 있는지 확인합니다.

대부분의 컴퓨터에는 SSH 클라이언트가 사전 설치되어 있습니다. 확인하려면 컴퓨터에서 터미널 창을 열고 다음 명령을 실행합니다.

```
ssh
```

명령이 인식되면 연결할 준비가 된 것입니다.

명령이 인식되지 않는 경우에는 SSH 클라이언트를 설치해야 합니다. SSH 클라이언트 설치에 대한 지침은 이 자습서의 범위를 벗어납니다. 도움이 필요하면 이 사용 설명서의 SSH 연결 사전 조건 섹션을 참조하거나 온라인에서 운영 체제에 SSH 클라이언트를 설치하는 방법에 대한 지침을 검색하세요.

인스턴스에 연결하려면 다음 단계를 따르세요.

1. 연결 시작:

   Amazon EC2 콘솔의 인스턴스 세부 정보 페이지에 있는 경우 연결 버튼(오른쪽 상단)을 선택합니다.

   다른 곳으로 이동한 경우 탐색 창에서 인스턴스를 선택합니다. 그런 다음 인스턴스 페이지에서 인스턴스 이름 옆의 확인란을 선택하고 연결 버튼(오른쪽 상단)을 선택합니다.

   그러면 인스턴스에 연결 페이지가 열립니다.
2. 연결 방법을 선택합니다.

   인스턴스에 연결 페이지에서 SSH 클라이언트 탭을 선택합니다.

   다음에 수행할 단계이므로 잠시 시간을 내어 이 페이지의 텍스트를 검토하세요.
3. SSH 명령을 검토합니다.

   예제에 인스턴스의 세부 정보에 따라 자동으로 생성되고 사용자 지정되는 명령이 표시됩니다. 프라이빗 키 이름은 시작할 때 지정한 퍼블릭 키의 이름에서 파생됩니다.

   명령은 다음과 같이 표시됩니다.

   ```
   ssh -i "test-instance-key-pair.pem" ec2-user@ec2-18-201-118-201.eu-
   west-1.compute.amazonaws.com
   ```
4. SSH 명령 복사:

   예제 SSH 명령 옆에 있는 복사 아이콘을 선택합니다.
5. 터미널 창 열기:

   로컬 컴퓨터에서 터미널 창을 엽니다.
6. SSH 명령 붙여넣기 및 실행:

   터미널 창에 SSH 명령을 붙여넣습니다. 특정 폴더에 프라이빗 키 파일을 저장한 경우 전체 파일 경로를 포함하도록 명령을 편집합니다.

키보드에서 Enter 키를 누릅니다.

다음과 비슷한 응답이 나타납니다.

```
The authenticity of host 'ec2-18-201-118-201.eu-west-1.compute.amazonaws.com
 (18-201-118-201)' can't be established.
ED25519 key fingerprint is SHA256:examplehxj9aOr1MogvKOoMNskVVIRBQBoq0example.This
 key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

7. 연결 완료:

   키보드에서 **yes**를 입력하고 Return을 누릅니다.

   지문을 확인하는 것은 이 자습서의 범위를 벗어납니다. 자세한 내용은 (선택 사항) 인스턴스 지문 가져오기을 참조하십시오.

   연결에 성공하면 터미널 프롬프트가 인스턴스의 퍼블릭 DNS를 표시하도록 변경됩니다.

축하합니다! 인스턴스에 성공적으로 연결되었습니다.

작업 9: 인스턴스 중지

이 작업에서는 프리 티어 혜택을 유지하기 위해 인스턴스를 중지합니다. 인스턴스가 중지되면 해당 인스턴스에 대한 비용은 더 이상 발생하지 않지만 EBS 스토리지에 대한 비용은 계속 발생합니다.

인스턴스를 중지하려면 다음 단계를 따르세요.

1. 중지 시작:

   여전히 인스턴스에 연결 페이지에 있는 경우 브레드크럼에서 인스턴스를 선택합니다. 다른 곳으로 이동한 경우 탐색 창에서 인스턴스를 선택합니다.

   그런 다음 인스턴스 페이지에서 인스턴스 이름 옆의 확인란을 선택한 다음 인스턴스 상태 메뉴(오른쪽 상단)를 선택하고 인스턴스 중지를 선택합니다. 메시지가 표시되면 중지를 선택합니다.

2. 인스턴스 상태 모니터링:

   인스턴스 페이지에서 인스턴스 상태 열을 확인합니다. 인스턴스의 상태가 중지 중으로 변경된 다음 중지됨으로 변경됩니다. 전체 텍스트가 보이지 않으면 열을 넓혀 보세요.

인스턴스 상태가 중지 중에서 중지됨으로 변경되었지만 아직 표시되지 않는 경우 새로 고침 아이콘(테이블 위)을 선택하여 인스턴스 테이블을 새로 고칩니다.

핵심 고려 사항

이 자습서에서는 다음 주요 개념을 다루었습니다.

- AMI는 인스턴스를 시작하는 데 필요한 운영 체제 및 소프트웨어가 포함된 템플릿인 Amazon Machine Image를 의미합니다.
- 인스턴스 유형은 인스턴스에 사용되는 호스트 컴퓨터의 하드웨어를 의미합니다. 인스턴스의 CPU, 메모리, 스토리지, 네트워킹 용량을 결정합니다.
- 키 페어는 인스턴스에 안전하게 연결하는 데 사용할 수 있는 퍼블릭 키와 프라이빗 키 세트를 의미합니다.
- 네트워크는 VPC(AWS 클라우드 내 계정 전용 가상 프라이빗 클라우드)와 서브넷(VPC 내 IP 주소의 범위)을 의미합니다.
- 보안 그룹은 인스턴스에 도달할 수 있는 트래픽을 제어하는 일련의 규칙을 의미합니다.
- EBS 볼륨은 인스턴스의 데이터 스토리지를 의미합니다. 모든 인스턴스에는 AMI를 저장하기 위한 루트 볼륨과 하나 이상의 데이터 볼륨(선택 사항)이 있습니다.
- 태그는 인스턴스에 선택적으로 할당할 수 있는 메타데이터입니다. 인스턴스 이름은 태그이며 해당 키는 이름이고 값은 사용자가 선택합니다.
- 연결은 인터넷을 통해 인스턴스에 액세스하는 것을 의미합니다.
- SSH는 인스턴스에 연결하는 데 사용할 수 있는 보안 쉘 연결 프로토콜을 의미합니다.
- 퍼블릭 DNS는 인스턴스의 고유한 퍼블릭 주소입니다.
- 인스턴스 사용자 이름은 인스턴스의 운영 체제에 따라 결정되며 연결하는 데 필요합니다.
- 인스턴스를 중지하면 인스턴스에 대한 요금은 중지되지만 EBS 스토리지 요금은 계속 부과됩니다.

다음 단계

인스턴스 시작, 연결, 중지에 대한 자신감을 키우려면 이 자습서의 단계를 반복해 보세요. 프리 티어 혜택을 유지하려면 시작한 인스턴스를 모두 종료해야 합니다.

이러한 기본 사항에 익숙해지면 고급 자습서를 살펴볼 수 있습니다. 더 많은 자습서는 다른 자습서를 찾고 있나요? 섹션을 참조하세요.

6분 길이의 How can I avoid charges on my account when using AWS 프리 티어 services 비디오를 시청해 보세요.

Amazon EC2 인스턴스 구성 파라미터에 대한 참조

Amazon EC2 콘솔의 시작 인스턴스 마법사와 시작 템플릿은 Amazon EC2 인스턴스 구성을 위한 모든 파라미터를 제공합니다.

키 페어를 제외하고 인스턴스 시작 마법사는 각 파라미터에 대한 기본값을 제공합니다. 기본값을 일부 또는 전부 수락하거나 고유한 값으로 인스턴스를 구성할 수 있습니다. 시작 템플릿을 생성할 때 파라미터는 선택 사항입니다. 시작 템플릿을 사용하여 인스턴스를 시작하는 경우 시작 템플릿에 지정된 파라미터가 인스턴스 시작 마법사의 기본값보다 우선합니다. 시작 템플릿에 지정되지 않은 모든 파라미터는 인스턴스 시작 마법사가 제공한 값을 기본값으로 사용합니다.

파라미터는 인스턴스 시작 마법사와 시작 템플릿에서 그룹화됩니다. 콘솔의 파라미터 그룹에 따라 다음과 같은 설명이 제공됩니다.

인스턴스 구성을 위한 파라미터

- 이름 및 태그
- 애플리케이션 및 OS 이미지(Amazon Machine Image)
- 인스턴스 유형
- 키 페어(로그인)
- 네트워크 설정
- 스토리지 구성
- 고급 세부 정보
- 요약

이름 및 태그

인스턴스 이름은 태그이며, 여기서 키는 이름이고 값은 사용자가 지정하는 이름입니다. 인스턴스, 볼륨 및 네트워크 인터페이스에 태그를 지정할 수 있습니다. 스팟 인스턴스의 경우 스팟 인스턴스 요청만 태깅할 수 있습니다. 태그에 대한 자세한 내용은 Amazon EC2 리소스 태깅 단원을 참조하세요.

인스턴스 이름과 추가 태그를 지정하는 것은 선택 사항입니다.

- 이름(Name)에 인스턴스를 설명하는 이름을 입력합니다. 이름을 지정하지 않으면 인스턴스를 시작할 때 자동으로 생성되는 ID로 인스턴스를 식별할 수 있습니다.

- 태그를 추가하려면 추가 태그 추가(Add additional tags)를 선택합니다. 태그 추가(Add tag)를 선택한 다음 키와 값을 입력하고 태그를 지정할 리소스 유형을 선택합니다. 추가할 각 추가 태그에 대해 태그 추가(Add tag)를 다시 선택합니다.

인스턴스를 시작할 때만 인스턴스 이름을 지정할 수 있습니다. 시작 템플릿을 생성할 때 인스턴스 이름을 지정할 수 없지만 인스턴스 시작 시 생성되는 리소스에 태그를 추가할 수 있습니다.

애플리케이션 및 OS 이미지(Amazon Machine Image)

Amazon Machine Image(AMI)에는 인스턴스를 생성하는 데 필요한 정보가 포함되어 있습니다. 예를 들어 AMI에는 Linux, Apache, 사용자의 웹 사이트 등 웹 서버 역할을 하는 데 필요한 소프트웨어가 포함될 수 있습니다.

다음과 같이 적합한 AMI를 찾을 수 있습니다. AMI를 찾는 각 옵션에서 취소(Cancel)(오른쪽 상단)를 선택하여 AMI를 선택하지 않고 인스턴스 시작 마법사로 돌아갈 수 있습니다.

검색 창

사용 가능한 모든 AMI를 검색하려면 AMI 검색 창에 키워드를 입력한 다음 Enter 키를 누릅니다. AMI를 선택하려면 선택(Select)을 선택합니다.

최근 항목

최근에 사용한 AMI입니다.

최근 시작(Recently launched) 또는 현재 사용 중(Currently in use)을 선택한 다음 Amazon Machine Image(AMI)(Amazon Machine Image (AMI))에서 AMI를 선택합니다.

내 AMI(My AMIs)

사용자가 소유한 프라이빗 AMI 또는 공유된 프라이빗 AMI입니다.

본인 소유(Owned by me) 또는 나와 공유됨(Shared with me)을 선택한 다음 Amazon Machine Image(AMI)(Amazon Machine Image (AMI))에서 AMI를 선택합니다.

빠른 시작

AMI는 운영 체제(OS) 별로 그룹화되어 있어 빠르게 시작할 수 있습니다.

먼저 필요한 OS를 선택한 다음 Amazon Machine Image(AMI)(Amazon Machine Image (AMI))에서 AMI를 선택합니다. 프리 티어로 이용할 수 있는 AMI를 선택하려면 AMI가 프리 티어 적격(Free tier eligible)으로 표시되어 있는지 확인합니다.

더 많은 AMI 검색(Browse more AMIs)

더 많은 AMI 검색(Browse more AMIs)을 선택하여 전체 AMI 카탈로그를 검색합니다.

- 사용 가능한 모든 AMI를 검색하려면 검색 창에 키워드를 입력한 다음 Enter 키를 누릅니다.
- Systems Manager 파라미터를 사용하여 AMI를 찾으려면 시스템 검색 창 오른쪽에 있는 화살표 버튼을 선택한 다음 Search by Systems Manager parameter(Systems Manager 파라미터로 검색)를 선택하세요. 자세한 내용은 Systems Manager 파라미터를 사용하여 AMI 참조 섹션을 참조하세요.
- 범주별로 검색하려면 퀵 스타트 AMI(Quickstart AMIs), 내 AMI(My AMIs), AWS Marketplace AMI 또는 커뮤니티 AMI(Community AMIs)를 선택합니다.

  AWS Marketplace는 AMI를 비롯하여 AWS에서 실행되는 소프트웨어를 구입할 수 있는 온라인 상점입니다. AWS Marketplace에서 인스턴스를 시작하는 방법에 대한 자세한 내용은 AWS Marketplace AMI에서 Amazon EC2 인스턴스 시작을 참조하세요. 커뮤니티 AMI(Community AMIs)에서 AWS 커뮤니티 멤버가 다른 사용자가 사용할 수 있도록 설정한 AMI를 찾을 수 있습니다. Amazon 또는 검증된 파트너의 AMI는 확인된 공급 업체로 표시됩니다.
- AMI 목록을 필터링하려면 화면 왼쪽에 있는 결과 구체화(Refine results) 아래에 확인란을 하나 이상 선택합니다. 선택한 검색 범주에 따라 필터 옵션이 다릅니다.
- 각 AMI의 지원 루트 디바이스 유형 목록을 확인합니다. EBS(Amazon EBS에서 지원) 또는 인스턴스 스토어(인스턴스 스토어에서 지원) 중 필요한 유형의 AMI를 확인합니다. 자세한 내용은 루트 디바이스 유형 섹션을 참조하세요.
- 각 AMI의 지원 가상화 유형 목록을 확인합니다. hvm 또는 paravirtual 중 필요한 유형의 AMI를 확인합니다. 예를 들어 일부 인스턴스 유형은 HVM이 필요합니다. Linux 가상화 유형에 대한 자세한 내용은 가상화 유형 섹션을 참조하세요.
- 각 AMI에 대해 나열된 부팅 모드를 확인합니다. 필요한 부팅 모드(legacy-bios, uefi 또는 uefi-preferred)를 사용하는 AMI를 확인합니다. 자세한 내용은 Amazon EC2 부팅 모드를 사용한 인스턴스 시작 동작 섹션을 참조하세요.
- 용도에 적합한 AMI를 선택하고 선택 버튼을 선택합니다.

AMI 변경 시 경고

인스턴스를 시작할 때 선택한 AMI와 연결된 볼륨 또는 보안 그룹의 구성을 수정한 다음 다른 AMI를 선택하면 현재 설정 중 일부가 변경 또는 제거된다는 경고 창이 열립니다. 보안 그룹 및 볼륨에 대한 변경 사항을 검토할 수 있습니다. 또한 추가 및 삭제될 볼륨을 보거나 추가할 볼륨만 볼 수도 있습니다. 시작 템플릿을 생성할 때는 이 경고가 나타나지 않습니다.

인스턴스 유형

인스턴스 유형은 인스턴스의 하드웨어 구성과 크기를 정의합니다. 대형 인스턴스는 CPU와 메모리가 더 높습니다. 자세한 내용은 Amazon EC2 인스턴스 유형을 참조하세요.

- 인스턴스 유형: 인스턴스 유형이 지정한 AMI와 호환되는지 확인합니다. 자세한 내용은 Amazon EC2 인스턴스 유형 섹션을 참조하세요.

  프리 티어 – AWS 계정이 12개월이 아직 안 된 경우 t2.micro 인스턴스 유형 또는 t2.micro를 사용할 수 없는 리전의 경우 t3.micro 인스턴스 유형을 선택하여 프리 티어로 Amazon EC2를 사용할 수 있습니다. t3.micro 인스턴스를 시작하면 기본적으로 무제한 모드로 설정되므로 CPU 사용량에 따라 추가 요금이 부과될 수 있다는 점에 유의하세요. 인스턴스 유형이 프리 티어 사용 자격이 있으면 프리 티어 사용 가능(Free tier eligible)으로 표시됩니다.
- 인스턴스 유형 비교(Compare instance types): vCPU 수, 아키텍처, 메모리 양(GiB), 스토리지 양(GB), 스토리지 유형 및 네트워크 성능과 같은 속성을 기준으로 서로 다른 인스턴스 유형을 비교할 수 있습니다.
- 조언 받기: EC2 인스턴스 유형 파인더에서 인스턴스 유형에 대한 지침과 제안을 받을 수 있습니다. 자세한 내용은 EC2 인스턴스 유형 파인더에서 권장 사항 받기 섹션을 참조하세요.
- (시작 템플릿만 해당) 고급: 인스턴스 속성을 지정하고 Amazon EC2에서 이러한 속성으로 인스턴스 유형을 식별할 수 있도록 하려면 고급을 선택한 다음 인스턴스 유형 속성 지정을 선택합니다.
  - vCPU 수(Number of vCPUs): 컴퓨팅 요구 사항에 맞는 최소 및 최대 vCPUs 수를 입력합니다. 제한이 없음을 표시하려면 최소값 **0**을 입력하고 최대값을 비워 둡니다.
  - 메모리 용량(MiB)(Amount of memory (MiB)): 컴퓨팅 요구 사항에 맞는 최소 및 최대 메모리 양을 MiB 단위로 입력합니다. 제한이 없음을 표시하려면 최소값 **0**을 입력하고 최대값을 비워 둡니다.
  - 선택적 인스턴스 유형 특성(Optional instance type attributes)을 확장하고 속성 추가(Add attribute)를 선택하여 컴퓨팅 요구 사항을 보다 자세히 표현할 수 있습니다. 각 속성에 대한 자세한 내용은 Amazon EC2 API Reference(Amazon EC2 API 레퍼런스)의 InstanceRequirementsRequest를 참조하세요.
  - 결과 인스턴스 유형(Resulting instance types): 지정한 속성과 일치하는 인스턴스 유형을 미리 볼 수 있습니다. 인스턴스 유형을 제외하려면 속성 추가(Add attribute)를 선택하고 속성(Attribute) 목록에서 제외된 인스턴스 유형(Excluded instance types)을 선택합니다. Attribute value(속성 값) 목록에서 제외할 인스턴스 유형을 선택합니다.

키 페어(로그인)

키 페어 이름(Key pair name)에서 기존 키 페어를 선택하거나 새로운 키 페어 생성(Create new key pair)을 선택하여 새로 생성합니다. 자세한 내용은 Amazon EC2 키 페어 및 Amazon EC2 인스턴스 섹션을 참조하세요.

> **Important**
>
> 키 페어 없이 진행(Proceed without key pair)(권장하지 않음) 옵션을 선택할 경우 사용자가 다른 방법으로 로그인할 수 있도록 구성된 AMI를 선택해야만 인스턴스에 연결할 수 있습니다.

네트워크 설정

네트워크 설정은 인스턴스의 IP 주소, 보안 그룹, 네트워크 인터페이스를 정의합니다. 기본 네트워크 설정을 사용하거나 필요에 따라 구성할 수 있습니다.

- (인스턴스 시작 마법사만 해당) VPC: 인스턴스에 대한 기존 VPC를 선택합니다. 리전의 기본 VPC가 기본적으로 선택됩니다. 또는 생성한 VPC 또는 공유된 VPC를 선택할 수 있습니다. 자세한 내용은 EC2 인스턴스용 가상 프라이빗 클라우드 섹션을 참조하세요.
- 서브넷: 인스턴스의 서브넷을 선택하거나 새 서브넷 생성을 선택하고 Amazon VPC 콘솔을 사용하여 새 서브넷을 생성합니다.
  - 선택한 VPC의 가용 영역, 로컬 영역, Wavelength 영역 또는 Outpost 영역에서 서브넷을 생성할 수 있습니다.
  - IPv6 전용 서브넷에서 인스턴스를 시작하려면 인스턴스가 Nitro 기반 인스턴스여야 합니다.
- (인스턴스 시작 마법사만 해당) 퍼블릭 IP 자동 할당: 퍼블릭 IPv4 주소의 자동 할당을 활성화 또는 비활성화합니다. 인스턴스를 기본 서브넷에서 시작할 경우 기본값은 활성화입니다. 기본이 아닌 서브넷에서 인스턴스를 시작할 때 기본값은 비활성화입니다. 자세한 내용은 퍼블릭 IPv4 주소 섹션을 참조하세요.

  보조 네트워크 인터페이스를 추가하는 경우 기본값이 아닌 서브넷에 대해 이 옵션을 활성화할 수 없습니다. 자세한 내용은 the section called “시작 시 퍼블릭 IPv4 주소 할당” 섹션을 참조하세요.
- (인스턴스 시작 마법사만 해당) IPv6 IP 자동 할당: IPv6 주소의 자동 할당을 활성화 또는 비활성화합니다. 자세한 내용은 IPv6 주소 섹션을 참조하세요.
- 방화벽(보안 그룹): 기존 보안 그룹을 선택하거나 새 보안 그룹을 생성합니다. 보안 그룹에 인스턴스와 주고받는 트래픽을 허용하는 규칙이 있는지 확인합니다. 다른 모든 트래픽은 무시됩니다.

새 보안 그룹을 생성하면 SSH(Linux 인스턴스) 또는 RDP(Windows 인스턴스)를 통해 모든 IP 주소에서 인스턴스에 연결할 수 있는 인바운드 규칙이 자동으로 생성됩니다. 필요에 따라 이 규칙을 제거하거나 수정할 수 있습니다. 필요에 따라 규칙을 추가할 수 있습니다. 자세한 내용은 보안 그룹 규칙 구성 섹션을 참조하세요.

> **Warning**
>
> 모든 IP 주소가 SSH 또는 RDP를 통해 인스턴스에 액세스하도록 하는 규칙은 테스트 인스턴스를 잠시 시작하고 곧 중지하거나 종료할 경우에 허용됩니다. 프로덕션 환경에는 안전하지 않습니다. 특정 IP 주소 범위에서만 인스턴스 액세스를 허용하도록 설정해야 합니다.

이 보안 그룹은 기본 네트워크 인터페이스와 보조 네트워크 인터페이스에 추가됩니다. 네트워크 인터페이스에 대한 추가 보안 그룹을 선택할 수 있지만 여기에서 선택한 보안 그룹은 제거할 수 없습니다.

- 고급 네트워크 구성-필요에 따라 기본 네트워크 인터페이스를 구성할 수 있습니다. 보조 네트워크 인터페이스를 추가하려면 Add network interface(네트워크 인터페이스 추가)를 선택합니다. 추가할 수 있는 네트워크 인터페이스의 수는 선택한 인스턴스 유형에 따라 다릅니다. 이 섹션은 서브넷을 선택한 경우에만 사용할 수 있습니다.
  - 디바이스 인덱스: 디바이스 인덱스입니다. 기본 네트워크 인터페이스는 인덱스 0에 할당되어야 합니다.
  - 네트워크 인터페이스: 네트워크 인터페이스입니다. Amazon EC2에서 새로운 인터페이스를 생성하도록 새 인터페이스를 선택하거나 사용 가능한 기존 네트워크 인터페이스를 선택합니다. 기존 네트워크 인터페이스를 기본 네트워크 인터페이스로 선택하는 경우 기본이 아닌 서브넷에 대해 퍼블릭 IP 자동 할당을 활성화할 수 없습니다.
  - 설명: 새로운 네트워크 인터페이스에 대한 설명입니다.
  - 서브넷(Subnet): 새로운 네트워크 인터페이스를 생성할 서브넷입니다. 인스턴스는 기본 네트워크 인터페이스와 동일한 서브넷에서 시작됩니다.

    기본 네트워크 인터페이스의 서브넷과 동일한 가용 영역에서 보조 네트워크 인터페이스의 서브넷을 선택해야 합니다. 다른 VPC에서 서브넷을 선택하면 네트워크 인터페이스 옆에 다중 VPC 레이블이 나타납니다. 그러면 네트워킹 및 보안 구성이 서로 다른 여러 VPC 사이에 다중 홈 인스턴스를 만들 수 있습니다.

    EC2 인스턴스를 IPv6 전용 서브넷으로 시작하려면 Nitro 기반 인스턴스를 사용해야 합니다. IPv6 전용 인스턴스를 시작할 때 DHCPv6이 인스턴스에 IPv6 DNS 이름 서버를 즉시 제공하지 않을 수

있습니다. 이 초기 지연 중에는 인스턴스가 퍼블릭 도메인을 확인하지 못할 수 있습니다. 구성 파일을 변경하고 AMI를 다시 이미지화하여 부팅 시 파일에 IPv6 DNS 이름 서버 주소를 즉시 부여할 수 있습니다.

- 보안 그룹: 네트워크 인터페이스와 연결할 보안 그룹입니다. 네트워크 인터페이스의 서브넷과 동일한 VPC에서 보안 그룹을 선택해야 합니다.
- (시작 템플릿만 해당) 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 섹션을 참조하세요.
- 기본 IP: 서브넷 범위 중 프라이빗 IPv4 주소입니다. Amazon EC2가 자동으로 프라이빗 IPv4 주소를 선택하도록 하려면 비워 둡니다.
- 보조 IP: 서브넷 범위 중 추가적인 프라이빗 IPv4 주소입니다. 수동 할당을 선택하고 IPv4 주소를 입력합니다. IP 추가를 선택하여 다른 IPv4 주소를 추가합니다. 또는 자동 할당을 선택하고 Amazon EC2가 사용자를 위해 선택할 IPv4 주소 수를 나타내는 값을 입력합니다.
- (IPv6에만 해당) IPv6 IP: 서브넷 범위 중 IPv6 주소입니다. 수동 할당을 선택하고 IPv6 주소를 입력합니다. IP 추가를 선택하여 다른 IPv6 주소를 추가합니다. 또는 자동 할당을 선택하고 Amazon EC2가 사용자를 위해 선택할 IPv6 주소 수를 나타내는 값을 입력합니다.
- [IPv4 접두사(IPv4 Prefixes)]: 네트워크 인터페이스의 IPv4 접두사입니다. 수동 할당을 선택하고 IPv4 접두사를 입력합니다. 또는 자동 할당을 선택하고 Amazon EC2가 사용자를 위해 선택할 IPv4 접두사 수를 나타내는 값을 입력합니다.
- [IPv6 접두사(IPv4 Prefixes)]: 네트워크 인터페이스의 IPv6 접두사입니다. 수동 할당을 선택하고 IPv6 접두사를 입력합니다. 또는 자동 할당을 선택하고 Amazon EC2가 사용자를 위해 선택할 IPv6 접두사 수를 나타내는 값을 입력합니다.
- (듀얼 스택 및 IPv6 전용) 기본 IPv6 IP 할당: 듀얼 스택 또는 IPv6 전용 서브넷을 선택하는 경우 기본 IPv6 주소를 할당합니다. 이렇게 하면 인스턴스 또는 네트워크 인터페이스로 향하는 트래픽이 중단되는 것을 방지할 수 있습니다. IPv6 주소가 변경되지 않는 경우 이 옵션을 활성화합니다. 기본 IPv6 주소는 나중에 제거할 수 없습니다. IPv6 GUA 주소를 기본 IPv6로 활성화하면 인스턴스가 종료되거나 네트워크 인터페이스가 분리될 때까지 첫 번째 IPv6 GUA가 기본 IPv6 주소로 설정됩니다. 네트워크 인터페이스와 연결된 IPv6 주소가 여러 개 있고 Amazon EC2가 기본 IPv6 주소를 할당하도록 허용한 경우, 네트워크 인터페이스와 연결된 첫 번째 IPv6 GUA 주소가 기본 IPv6 주소입니다.
- 종료 시 삭제: 인스턴스가 삭제될 때 네트워크 인터페이스도 삭제되도록 할 것인지 여부를 나타냅니다.
- 인터페이스 유형: 네트워크 인터페이스 유형:

  - ENA: CPU 사용량을 최소화하면서 TCP/IP 프로토콜의 높은 처리량과 초당 패킷 속도를 처리하도록 설계된 고성능 네트워크 인터페이스입니다. 이것이 기본값입니다. ENA에 대한 자세한 내용은 Elastic Network Adapter 섹션을 참조하세요.
  - ENA 포함 EFA: SRD 기반 전송과 함께 기존 TCP/IP 기반 전송을 위해 ENA 디바이스와 EFA 디바이스를 모두 지원하는 네트워크 인터페이스입니다. EFA에 대한 자세한 내용은 Elastic Fabric Adapter 섹션을 참조하세요.
  - EFA 전용: 운영 체제 스택을 우회하면서 SRD 기반 전송을 위한 높은 처리량과 지연 시간이 짧은 노드 간 통신을 처리하도록 설계된 고성능 네트워크 인터페이스입니다. EFA 전용 네트워크 인터페이스는 IP 주소를 지원하지 않습니다. EFA에 대한 자세한 내용은 Elastic Fabric Adapter 섹션을 참조하세요.
- Elastic Fabric Adapter(EFA): 네트워크 인터페이스가 Elastic Fabric Adapter(EFA)임을 나타냅니다. 자세한 내용은 Amazon EC2의 AI/ML 및 HPC 워크로드를 위한 Elastic Fabric Adapter 섹션을 참조하세요.
- 네트워크 카드 인덱스: 네트워크 카드의 인덱스입니다. 기본 네트워크 인터페이스는 네트워크 카드 인덱스 0에 할당되어야 합니다. 일부 인스턴스 유형은 여러 네트워크 카드를 지원합니다.
- ENA Express: ENA Express는 AWS SRD(Scalable Reliable Datagram) 기술로 구동됩니다. SRD 기술은 패킷 분산 메커니즘을 사용하여 부하를 분산하고 네트워크 혼잡을 방지합니다. ENA Express를 활성화하면 지원되는 인스턴스가 가능한 경우 일반 TCP 트래픽을 기반으로 SRD를 사용하여 통신할 수 있습니다. 목록에서 활성화 또는 비활성화를 선택하지 않는 한 인스턴스 시작 마법사 또는 시작 템플릿에는 인스턴스에 대한 ENA Express 구성이 포함되지 않습니다.
- ENA Express UDP: ENA Express를 활성화한 경우 필요에 따라 UDP 트래픽에 사용할 수 있습니다. 활성화 또는 비활성화를 선택하지 않는 한 인스턴스 시작 마법사 또는 시작 템플릿에는 인스턴스에 대한 ENA Express 구성이 포함되지 않습니다.

스토리지 구성

선택한 AMI에는 루트 볼륨을 포함한 하나 이상의 스토리지 볼륨이 있습니다. 인스턴스에 연결할 추가 볼륨을 지정할 수 있습니다.

(인스턴스 시작 마법사만 해당) 단순 보기 또는 고급 보기를 사용할 수 있습니다. 간단(Simple) 보기를 통해 볼륨의 크기와 유형을 지정합니다. 모든 볼륨 파라미터를 지정하려면 고급(Advanced) 보기(카드 우측 상단)를 선택합니다.

고급(Advanced) 보기를 사용하여 다음과 같이 각 볼륨을 구성할 수 있습니다.

- 스토리지 유형(Storage type): 인스턴스에 연결할 Amazon EBS 또는 인스턴스 스토어 볼륨을 선택합니다. 목록에서 사용 가능한 볼륨 유형은 선택한 인스턴스 유형에 따라 다릅니다. 자세한 내용은 EC2 인스턴스용 인스턴스 스토어 임시 블록 스토리지 및 Amazon EBS volumes를 참조하세요.
- 디바이스 이름(Device name): 볼륨에서 사용할 디바이스 이름을 목록에서 선택합니다.
- 스냅샷(Snapshot): 볼륨 복원에 사용할 스냅샷을 입력합니다. 스냅샷(Snapshot) 필드에 텍스트를 입력하여 사용 가능한 공유 및 퍼블릭 스냅샷을 검색할 수 있습니다.
- Size(GiB)(크기(GiB)): EBS 볼륨의 경우 스토리지 크기를 지정할 수 있습니다. 선택한 AMI와 인스턴스가 프리 티어에 해당되는 경우 프리 티어 한도를 유지하려면 총 스토리지 크기를 30GiB 미만으로 유지해야 합니다.
- 볼륨 유형(Volume type): EBS 볼륨에 대한 볼륨 유형을 선택합니다. 자세한 내용은 Amazon EBS 사용 설명서의 Amazon EBS volume types를 참조하세요.
- IOPS: Provisioned IOPS SSD 볼륨 유형을 선택한 경우, 볼륨에서 지원되는 초당 I/O(IOPS) 수를 입력할 수 있습니다.
- 종료 시 삭제(Delete on termination): Amazon EBS 볼륨에서 예(Yes)를 선택하여 인스턴스가 종료될 때 볼륨을 삭제하거나, 아니요(No)를 선택하여 볼륨을 유지합니다. 자세한 내용은 인스턴스가 종료될 때 데이터 보존 섹션을 참조하세요.
- 암호화(Encrypted): 인스턴스 유형이 EBS 암호화를 지원하는 경우 예(Yes)를 선택하여 볼륨의 암호화를 활성화할 수 있습니다. 이 리전에서 기본적으로 암호화를 활성화한 경우, 사용자에 대해 암호화가 활성화됩니다. 자세한 내용은 Amazon EBS 사용 설명서의 Amazon EBS encryption을 참조하세요.
- 키(Key): 암호화(Encrypted)에 대해 예(Yes)를 선택한 경우 볼륨을 암호화하는 데 사용할 고객 관리형 키를 선택해야 합니다. 이 리전에서 기본적으로 암호화를 사용하도록 설정한 경우 기본 고객 관리형 키가 자동으로 선택됩니다. 다른 키를 선택하거나 생성한 고객 관리형 키의 ARN을 지정할 수 있습니다.
- 처리량: gp3 볼륨 유형을 선택한 경우 볼륨이 지원할 수 있는 처리량을 MiB/s 단위로 입력할 수 있습니다.
- 볼륨 초기화 속도: 스냅샷을 선택한 경우 선택적으로 스냅샷 블록을 Amazon S3에서 볼륨으로 다운로드할 볼륨 초기화 속도를 MiB/s 단위로 지정할 수 있습니다. 자세한 내용은 볼륨 초기화에 Amazon EBS 프로비저닝 속도 사용을 참조하세요. 기본 초기화 속도 또는 빠른 스냅샷 복원(선택한 스냅샷에 대해 활성화된 경우)을 사용하려면 속도를 지정하지 마세요.
- 파일 시스템: Amazon EFS 또는 Amazon FSx 파일 시스템을 인스턴스에 탑재합니다. Amazon EFS 파일 시스템 탑재에 대한 자세한 내용은 Amazon EC2 Linux 인스턴스에서 Amazon EFS 사용 섹션을 참조하세요. Amazon FSx 시스템 탑재에 대한 자세한 내용은 Amazon EC2 인스턴스에서 Amazon FSx 사용 섹션을 참조하세요.

고급 세부 정보

고급 세부 정보에서 필드를 볼 수 있도록 섹션을 확장하고 인스턴스를 위한 추가 파라미터를 지정합니다.

- (인스턴스 시작 마법사만 해당) 도메인 조인 디렉터리: 시작 후 사용자의 인스턴스가 조인된 AWS Directory Service 디렉터리(도메인)를 선택합니다. 도메인을 선택하는 경우, 필요한 권한이 있는 IAM 역할을 선택해야 합니다. 도메인 조인에 대한 자세한 내용은 Amazon EC2 Linux 인스턴스를 AWS 관리형 Microsoft AD 디렉터에 원활하게 조인(Linux 인스턴스) 및 Amazon EC2 Windows 인스턴스를 AWS 관리형 Microsoft AD 디렉터리에 원활하게 조인(Windows 인스턴스)을 참조하세요.
- IAM 인스턴스 프로파일: 인스턴스와 연결할 IAM 인스턴스 프로파일을 선택합니다. IAM 역할의 컨테이너입니다. 자세한 내용은 Amazon EC2의 IAM 역할 섹션을 참조하세요.
- 호스트 이름 유형(Hostname type): 인스턴스의 게스트 OS 호스트 이름에 리소스 이름 또는 IP 이름을 포함할지 선택합니다. 자세한 내용은 EC2 인스턴스 호스트 이름 및 도메인 섹션을 참조하세요.
- DNS 호스트 이름(DNS Hostname): 리소스 이름 또는 IP 이름(호스트 이름 유형(Hostname type)에 대해 선택한 것에 따라)에 대한 DNS 쿼리가 IPv4 주소(A 레코드), IPv6 주소(AAAA 레코드) 또는 둘 다로 응답할지 결정합니다. 자세한 내용은 EC2 인스턴스 호스트 이름 및 도메인 섹션을 참조하세요.
- 인스턴스 자동 복구: 활성화되면 시스템 상태 확인이 실패할 경우 인스턴스를 복구합니다. 이 설정은 지원되는 인스턴스 유형에 대해 시작 시 활성화됩니다. 자세한 내용은 Amazon EC2 인스턴스에서 간소화된 자동 복구 구성 섹션을 참조하세요.
- 종료 동작: 인스턴스 종료 시 적용할 인스턴스 상태(중지 또는 종료)를 선택합니다. 자세한 내용은 인스턴스가 시작하는 종료 동작 변경 섹션을 참조하세요.
- 중지 - 최대 절전 모드 동작(Stop - Hibernate behavior): 최대 절전 모드를 사용하려면 활성화(Enable)를 선택합니다. 이 필드는 인스턴스가 최대 절전 모드 필수 조건을 충족하는 경우에만 사용할 수 있습니다. 자세한 내용은 Amazon EC2 인스턴스를 최대 절전 모드로 전환 섹션을 참조하세요.
- 종료 방지(Termination protection): 실수로 인스턴스를 종료하는 일을 방지하려면 활성화(Enable)를 선택합니다. 자세한 내용은 인스턴스 종료 방지 변경 섹션을 참조하세요.
- 중지 방지(Stop protection): 우발적 중지를 방지하려면 활성화(Enable)를 선택합니다. 자세한 내용은 중지 방지 사용 설정 섹션을 참조하세요.
- 세부 CloudWatch 모니터링(Detailed CloudWatch monitoring): Amazon CloudWatch를 사용하여 인스턴스에 대한 세부 모니터링 기능을 켜려면 사용 설정(Enable)을 선택합니다. 추가 요금이 발생합니다. 자세한 내용은 CloudWatch를 사용하여 인스턴스 모니터링 섹션을 참조하세요.
- 크레딧 사양(Credit specification): 애플리케이션이 필요한 만큼 기준 이상으로 버스트하도록 하려면 무제한(Unlimited)을 선택합니다. 이 필드는 T 인스턴스에만 유효합니다. 추가 요금이 적용될 수 있습니다. 자세한 내용은 성능 순간 확장 가능 인스턴스 섹션을 참조하세요.

- 배치 그룹: 인스턴스를 시작할 배치 그룹을 지정합니다. 기존의 배치 그룹을 선택하거나 새로 생성할 수 있습니다. 모든 인스턴스 유형이 하나의 배치 그룹에서 하나의 인스턴스 시작을 지원하지 않습니다. 자세한 내용은 Amazon EC2 인스턴스에 대한 배치 그룹 섹션을 참조하세요.
- EBS 최적화 인스턴스(EBS-optimized instance): Amazon EBS에 최적화된 인스턴스는 최적화된 구성 스택을 사용하며 Amazon EBS I/O를 위한 추가 전용 용량을 제공합니다. 인스턴스 유형이 이 기능을 지원하는 경우 기능을 사용하려면 활성화(Enable)를 선택하여 활성화합니다. 추가 요금이 발생합니다. 자세한 내용은 the section called “EBS 최적화” 섹션을 참조하세요.
- 인스턴스 대역폭 구성: 네트워크 대역폭 또는 EBS 대역폭을 향상시킬 수 있습니다. 지원되는 인스턴스 유형에만 사용할 수 있습니다. 자세한 내용은 EC2 인스턴스 대역폭 가중치 구성 섹션을 참조하세요.
- 구매 옵션: 온디맨드 가격으로 제한된 스팟 가격에서 스팟 인스턴스를 요청하려면 스팟 인스턴스를 선택하고 기본 스팟 인스턴스 설정을 변경하려면 스팟 인스턴스 옵션 사용자 지정을 선택합니다. 최고 가격을 설정(권장되지 않음)하고 요청 유형, 요청 기간 및 중단 동작을 변경할 수 있습니다. 스팟 인스턴스를 요청하지 않으면 Amazon EC2는 기본적으로 온디맨드 인스턴스를 시작합니다. 자세한 내용은 스팟 인스턴스 관리 섹션을 참조하세요.
- 용량 예약(Capacity Reservation): 인스턴스를 모든 열린 용량 예약(공개(Open)), 특정 용량 예약(ID별 목표(Target by ID)) 또는 용량 예약 그룹(그룹별 목표(Target by group))으로 시작할지 여부를 지정합니다. 용량 예약을 사용하지 않도록 지정하려면 없음(None)을 선택합니다. 자세한 내용은 인스턴스를 기존 용량 예약으로 시작 섹션을 참조하세요.
- 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 Amazon EC2 전용 인스턴스 및 Amazon EC2 전용 호스트 섹션을 참조하세요.
- RAM 디스크 ID(RAM disk ID): (반가상화(PV) AMI에만 유효) 인스턴스의 RAM 디스크를 선택합니다. 커널을 선택한 경우 지원하는 드라이버가 있는 특정 RAM 디스크를 선택해야 할 수도 있습니다.
- 커널 ID(Kernel ID): (반가상화(PV) AMI에만 유효) 인스턴스의 커널을 선택합니다.
- [Nitro Enclave]: Amazon EC2 인스턴스에서 enclaves라는 격리된 실행 환경을 생성할 수 있습니다. AWS Nitro Enclaves에 대해 인스턴스를 활성화하려면 활성화(Enable)를 선택합니다. 자세한 내용은 AWS Nitro Enclaves 사용 설명서의 AWS Nitro Enclaves란 무엇입니까?를 참조하세요.
- 라이선스 구성: 지정된 라이선스 구성에 대해 인스턴스를 시작하여 라이선스 사용을 추적할 수 있습니다. 자세한 내용은 AWS License Manager 사용 설명서에서 라이선스 구성 생성을 참조하세요.
- CPU 옵션 지정: 인스턴스 시작 마법사에서는 선택한 인스턴스 유형이 CPU 옵션 지정을 지원하는 경우에만 이 필드가 표시됩니다. CPU 옵션 지정을 선택해 시작 중 vCPU 수를 사용자 지정할 수 있

습니다. CPU 코어 수와 코어당 스레드 수를 설정합니다. 자세한 내용은 Amazon EC2 인스턴스의 CPU 옵션 섹션을 참조하세요.

- 액세스 가능한 메타데이터: 인스턴스 메타데이터 서비스(IMDS)에 대한 액세스를 활성화하거나 비활성화할 수 있습니다. 자세한 내용은 새 인스턴스에 대한 인스턴스 메타데이터 옵션 구성 섹션을 참조하세요.
- 메타데이터 IPv6 엔드포인트: IMDS IPv6 주소 `[fd00:ec2::254]`를 사용하여 인스턴스 메타데이터를 검색하도록 인스턴스를 설정할 수 있습니다. 이 옵션은 Nitro 기반 인스턴스를 IPv6 지원 서브넷(듀얼 스택 또는 IPv6 전용)으로 시작하는 경우에만 사용할 수 있습니다. 인스턴스 메타데이터 검색에 대한 자세한 내용은 EC2 인스턴스에 대한 인스턴스 메타데이터에 액세스 섹션을 참조하세요.
- 메타데이터 버전: IMDS에 대한 액세스를 활성화하면 인스턴스 메타데이터를 요청할 때 인스턴스 메타데이터 서비스 버전 2를 사용하도록 요구할 수 있습니다. 자세한 내용은 새 인스턴스에 대한 인스턴스 메타데이터 옵션 구성 섹션을 참조하세요.
- 메타데이터 응답 홉 제한: IMDS를 활성화하면 메타데이터 토큰에 허용되는 네트워크 홉 수를 설정할 수 있습니다. 자세한 내용은 새 인스턴스에 대한 인스턴스 메타데이터 옵션 구성 섹션을 참조하세요.
- 메타데이터에 태그 허용(Allow tags in metadata): 다음을 선택하는 경우 사용 설정(Enable)을 선택하면 인스턴스가 메타데이터에서 모든 태그에 액세스할 수 있습니다. 값을 지정하지 않으면 기본적으로 인스턴스 메타데이터의 태그에 대한 액세스가 허용되지 않습니다. 자세한 내용은 인스턴스 메타데이터에서 태그에 대한 액세스 활성화 섹션을 참조하세요.
- 사용자 데이터: 시작 과정에서 인스턴스를 구성하거나 구성 스크립트를 실행할 때 사용할 사용자 데이터를 지정할 수 있습니다. Linux 인스턴스의 사용자 데이터에 대한 자세한 내용은 사용자 데이터 입력을 사용하여 EC2 인스턴스를 시작할 때 명령 실행 섹션을 참조하세요. Windows 인스턴스의 사용자 데이터에 대한 자세한 내용은 Amazon EC2가 Windows 인스턴스의 사용자 데이터를 처리하는 방법 섹션을 참조하세요.

요약

요약(Summary) 패널을 사용하여 시작할 인스턴스 수를 지정하고 인스턴스 구성을 검토하고 인스턴스를 시작합니다.

- 인스턴스 개수: 시작할 인스턴스의 수를 입력합니다. 모든 인스턴스는 동일한 구성으로 시작됩니다.

> **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) 인스턴스를 2개 이상 지정하는 경우 애플리케이션 수요를 처리할 인스턴스의 수를 올바르게 유지하는 데 도움이 되도록 EC2 Auto Scaling 고려(Consider EC2 Auto Scaling)를 선택하여 시작 템플릿과 Auto Scaling 그룹을 생성할 수 있습니다. Auto Scaling은 사양에 따라 그룹에서 인스턴스의 수를 조정합니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서를 참조하세요.

> **Note**
>
> Amazon EC2 Auto Scaling에서 상태가 비정상인 것으로 표시한 Auto Scaling 그룹 내 인스턴스는 자동으로 교체 예정되어 종료되고 다른 인스턴스가 시작되어 원래 인스턴스의 데이터가 손실됩니다. 사용자가 인스턴스를 중지 또는 재부팅하거나 다른 이벤트에서 인스턴스의 상태를 비정상인 것으로 표시하는 경우 인스턴스가 비정상으로 표시됩니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Auto Scaling 그룹의 인스턴스 상태 확인 섹션을 참조하세요.

- 인스턴스의 세부 정보를 검토하고 필요한 사항을 변경합니다. 요약(Summary) 패널에서 해당 링크를 선택하여 섹션으로 직접 이동할 수 있습니다.
- 인스턴스를 시작할 준비가 되면 인스턴스 시작(Launch instance)을 선택합니다.

콘솔에서 인스턴스 시작 마법사를 사용하여 EC2 인스턴스 시작

Amazon EC2 콘솔에서 인스턴스 시작 마법사를 사용하여 Amazon EC2 인스턴스를 시작할 수 있습니다. 마법사는 시작 파라미터의 기본값을 제공합니다. 이를 그대로 사용하거나 요구 사항에 맞게 수정할 수 있습니다. 지정되지 않는 유일한 파라미터는 키 페어입니다. 기본값을 수락하면 키 페어만 선택하여 인스턴스를 빠르게 시작할 수 있습니다.

> **Important**
>
> 인스턴스가 유휴 상태일 때도 `running` 상태인 동안에는 인스턴스에 대한 요금이 부과됩니다. 하지만 프리 티어에 해당한다면 요금이 부과되지 않을 수 있습니다. 자세한 내용은 Amazon EC2의 프리 티어 사용량 추적 섹션을 참조하세요.

인스턴스 시작 마법사의 각 파라미터에 대한 설명은 Amazon EC2 인스턴스 구성 파라미터에 대한 참조 섹션을 참조하세요.

주제

- 빠르게 인스턴스 시작
- 정의된 파라미터를 사용하여 인스턴스 시작

빠르게 인스턴스 시작

테스트 목적으로 인스턴스를 빠르게 설정하려면 다음 단계를 따르세요. 운영 체제와 키 페어를 선택하고 기본값을 그대로 사용합니다. 키 페어를 제외하고 인스턴스 시작 마법사는 모든 파라미터에 대한 기본값을 제공합니다. 기본값의 일부 또는 전부를 수락하거나 각 파라미터에 고유한 값을 지정하여 인스턴스를 구성할 수 있습니다.

인스턴스 시작 마법사의 각 파라미터에 대한 설명은 Amazon EC2 인스턴스 구성 파라미터에 대한 참조 섹션을 참조하세요.

인스턴스 시작 마법사를 사용하여 빠르게 인스턴스를 시작하려면 다음을 수행하세요.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 화면 상단의 탐색 모음에는 현재 AWS 리전이 표시됩니다(예: 미국 동부(오하이오)). 필요한 경우 인스턴스를 시작할 다른 리전을 선택합니다.
3. Amazon EC2 콘솔 대시보드에서 인스턴스 시작을 선택합니다.
4. (선택 사항) 이름 및 태그(Name and tags) 아래의 이름(Name)에 인스턴스를 설명하는 이름을 입력합니다.
5. 애플리케이션 및 OS 이미지(Amazon Machine Image)(Application and OS Images (Amazon Machine Image)) 아래에서 빠른 시작(Quick Start)을 선택한 다음 인스턴스의 운영 체제(OS)를 선택합니다.
6. (선택 사항) 키 페어(로그인)(Key pair (login)) 아래의 키 페어 이름(Key pair name)에서 기존 키 페어를 선택하거나 새로 생성합니다.
7. 요약(Summary) 패널에서 인스턴스 실행(Launch instance)을 선택합니다.

정의된 파라미터를 사용하여 인스턴스 시작

프로덕션에서 사용할 인스턴스를 시작하는 경우 요구 사항에 맞게 인스턴스를 구성해야 합니다. 인스턴스 시작 마법사의 각 파라미터에 대한 설명은 Amazon EC2 인스턴스 구성 파라미터에 대한 참조 섹션을 참조하세요.

인스턴스 시작 마법사로 모든 시작 파라미터를 정의하여 인스턴스를 시작하려면 다음을 수행하세요.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 화면 상단의 탐색 모음에는 현재 AWS 리전이 표시됩니다(예: 미국 동부(오하이오)). 필요한 경우 인스턴스를 시작할 다른 리전을 선택합니다.
3. Amazon EC2 콘솔 대시보드에서 인스턴스 시작을 선택합니다.
4. (선택 사항) 쉽게 추적할 수 있도록 이름 및 태그 아래의 이름에 인스턴스에 대한 설명이 포함된 이름을 입력합니다.

   인스턴스 이름은 태그이며, 여기서 키는 이름이고 값은 사용자가 지정하는 이름입니다.
5. 애플리케이션 및 OS 이미지(Amazon Machine Image)에서 인스턴스의 운영 체제(OS)를 선택한 다음 AMI를 선택합니다.

   AMI는 인스턴스를 시작하는 데 필요한 운영 체제와 소프트웨어가 들어 있는 템플릿입니다.
6. 인스턴스 유형(Instance type)에서 인스턴스 유형을 선택합니다.

   인스턴스 유형은 인스턴스에 사용되는 호스트 컴퓨터의 하드웨어 구성(CPU, 메모리, 스토리지, 네트워킹 용량)과 크기를 결정합니다.

   어떤 인스턴스 유형을 선택해야 할지 확실하지 않은 경우 다음을 수행할 수 있습니다.

   - 인스턴스 유형 비교를 선택하여 vCPU 수, 아키텍처, 메모리 양(GiB), 스토리지 양(GB), 스토리지 유형 및 네트워크 성능과 같은 속성을 기준으로 서로 다른 인스턴스 유형을 비교합니다.
   - 조언 받기를 선택하여 Amazon Q EC2 인스턴스 유형 선택기에서 인스턴스 유형에 대한 지침과 제안을 받습니다. 자세한 내용은 EC2 인스턴스 유형 파인더에서 권장 사항 받기 단원을 참조하십시오.

   > **Note**
   >
   > AWS 계정 사용 기간이 12개월 미만인 경우 t2.micro 인스턴스 유형 또는 t2.micro를 사용할 수 없는 리전의 경우 t3.micro 인스턴스 유형을 선택하여 프리 티어로 Amazon EC2를 사용할 수 있습니다. t3.micro 인스턴스를 시작하면 기본적으로 무제한 모드로 설정되므로

CPU 사용량에 따라 추가 요금이 부과될 수 있다는 점에 유의하세요. 인스턴스 유형이 프리 티어 사용 자격이 있으면 프리 티어 사용 가능(Free tier eligible)으로 표시됩니다.

7. (선택 사항) 키 페어(로그인)(Key pair (login)) 아래의 키 페어 이름(Key pair name)에서 기존 키 페어를 선택하거나 새로 생성합니다. 인스턴스에 연결하는 데 키 페어가 필요하지 않은 경우 키 페어 없이 계속 진행(권장되지 않음)을 선택할 수 있습니다.
8. 네트워크 설정에서 테스트 인스턴스를 시작하는 경우 기본값을 유지할 수 있습니다. 프로덕션 인스턴스를 시작하는 경우 정의한 네트워크 설정과 보안 그룹을 사용하여 인스턴스로 들어오고 나가는 트래픽을 제어하는 것이 가장 좋습니다.
9. 스토리지 구성에서 기본값을 유지하거나 추가 스토리지를 지정할 수 있습니다. 선택한 AMI에는 루트 볼륨을 포함한 하나 이상의 스토리지 볼륨이 있습니다. 인스턴스에 연결할 추가 볼륨을 지정할 수 있습니다.

   간단(Simple) 또는 고급(Advanced) 보기를 사용할 수 있습니다. 간단(Simple) 보기를 통해 볼륨의 크기와 유형을 지정합니다. 모든 볼륨 파라미터를 지정하려면 고급(Advanced) 보기(카드 우측 상단)를 선택합니다.
10. 고급 세부 정보에서 필드를 볼 수 있도록 섹션을 확장하고 인스턴스를 위한 추가 파라미터를 지정합니다.
11. 요약 패널에서 다음을 수행할 수 있습니다.

    a. 시작할 인스턴스 수를 지정합니다.

    b. 인스턴스 구성을 검토하고 해당 링크를 선택하여 섹션으로 직접 이동합니다.

    c. 인스턴스를 시작할 준비가 되면 인스턴스 시작(Launch instance)을 선택합니다.

   인스턴스가 시작하지 않거나 상태가 `terminated`이 아닌 `running`로 변경되는 경우, Amazon EC2 인스턴스 시작 문제 해결 섹션을 참조하세요.
12. (선택 사항) 인스턴스에 대한 결제 알림을 생성할 수 있습니다. 확인 화면의 다음 단계(Next Steps)에서 결제 알림 생성(Create billing alerts)을 선택하고 지침을 따르세요. 인스턴스를 시작하면 결제 알림도 생성할 수 있습니다. 자세한 정보는 Amazon CloudWatch 사용 설명서의 예상 AWS 요금을 모니터링하기 위한 결제 경보 생성을 참조하세요.

시작 템플릿을 사용하여 EC2 인스턴스 시작

Amazon EC2 시작 템플릿은 인스턴스 시작 파라미터를 저장하므로 인스턴스를 시작할 때마다 파라미터를 지정할 필요가 없습니다.

일부 인스턴스 시작 서비스는 인스턴스를 시작할 때 선택적으로 시작 템플릿을 사용할 수 있지만, EC2 플릿 등의 다른 서비스의 경우 시작 템플릿을 사용하지 않으면 인스턴스를 시작할 수 없습니다. 이 주제에서는 EC2 인스턴스 시작 마법사, Amazon EC2 Auto Scaling, EC2 플릿 및 스팟 플릿을 사용하여 인스턴스를 시작할 때 시작 템플릿을 사용하는 방법을 설명합니다.

시작 템플릿을 생성하는 방법을 비롯하여 시작 템플릿에 대한 자세한 내용은 Amazon EC2 시작 템플릿에 인스턴스 시작 파라미터 저장 섹션을 참조하세요.

주제

- 시작 템플릿을 사용하여 Amazon EC2 인스턴스 시작
- 시작 템플릿을 사용하여 Amazon EC2 Auto Scaling 그룹에서 인스턴스를 시작합니다.
- 시작 템플릿을 사용하여 EC2 플릿 시작
- 시작 템플릿을 사용하여 스팟 플릿 시작

시작 템플릿을 사용하여 Amazon EC2 인스턴스 시작

시작 템플릿에 포함된 파라미터를 사용하여 Amazon EC2 인스턴스를 시작할 수 있습니다. 시작 템플릿을 선택한 후 인스턴스를 시작하기 전에 시작 파라미터를 수정할 수 있습니다.

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.

Console

시작 템플릿을 사용하여 인스턴스를 시작하려면 다음을 수행하세요.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 다음 옵션 중 하나를 사용하여 시작 템플릿을 선택합니다.
   - Amazon EC2 콘솔 대시보드에서 인스턴스 시작 옆의 아래쪽 화살표를 선택하고 템플릿에서 인스턴스 시작을 선택한 다음 소스 템플릿에서 시작 템플릿을 선택합니다.
   - 탐색 창에서 시작 템플릿을 선택하고 시작 템플릿을 선택한 다음 작업, 템플릿에서 인스턴스 시작을 선택합니다.
3. 소스 템플릿 버전의 경우 사용할 시작 템플릿 버전을 선택합니다.

4. (선택 사항) 시작 파라미터의 값을 수정할 수 있습니다. 값을 수정하지 않으면 시작 템플릿에 정의한 값이 사용됩니다. 시작 템플릿에 값이 지정되지 않은 경우 파라미터의 기본값이 사용됩니다.
5. 요약 패널의 인스턴스 수에 시작할 인스턴스 수를 지정합니다.
6. 인스턴스 시작을 선택합니다.

   인스턴스가 시작하지 않거나 상태가 terminated이 아닌 running로 변경되는 경우, Amazon EC2 인스턴스 시작 문제 해결 섹션을 참조하세요.

AWS CLI

시작 템플릿에서 인스턴스를 시작하려면 다음을 수행하세요.

- run-instances 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, Amazon EC2 인스턴스 시작 문제 해결 섹션을 참조하세요.

PowerShell

AWS Tools for PowerShell를 사용하여 시작 템플릿에서 인스턴스를 시작하려면

- New-EC2Instance 명령을 사용하여 `-LaunchTemplate` 파라미터를 지정합니다. 선택에 따라 사용할 시작 템플릿 버전을 지정합니다. 버전을 지정하지 않으면 기본 버전이 사용됩니다.

```
Import-Module AWS.Tools.EC2
New-EC2Instance `
    -LaunchTemplate (
        New-Object -TypeName Amazon.EC2.Model.LaunchTemplateSpecification -
Property @{
            LaunchTemplateId = 'lt-0abcd290751193123';
            Version          = '4'
    }
)
```

- 시작 템플릿 파라미터를 재정의하려면 New-EC2Instance 명령에서 파라미터를 지정합니다. 아래 예제는 시작 템플릿(존재하는 경우)에 지정된 인스턴스 유형을 재정의합니다.

```
Import-Module AWS.Tools.EC2
New-EC2Instance `
    -InstanceType t4g.small `
    -LaunchTemplate (
        New-Object -TypeName Amazon.EC2.Model.LaunchTemplateSpecification -
Property @{
            LaunchTemplateId = 'lt-0abcd290751193123';
            Version          = '4'
    }
```

```
)
```

- 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.

```
Import-Module AWS.Tools.EC2
New-EC2Instance `
    -InstanceType t4g.small  `
    -LaunchTemplate (
        New-Object -TypeName Amazon.EC2.Model.LaunchTemplateSpecification -
Property @{
            LaunchTemplateId = 'lt-0abcd290751193123';
            Version          = '4'
        }
) `
    -TagSpecification (
        New-Object -TypeName Amazon.EC2.Model.TagSpecification -Property @{
            ResourceType = 'instance';
            Tags         = @(
                @{key = "Owner"; value = "TeamA" },
                @{key = "Department"; value = "Operations" }
            )
        }
)
```

  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.

```
Import-Module AWS.Tools.EC2
New-EC2Instance `
    -InstanceType t4g.small  `
    -LaunchTemplate (
        New-Object -TypeName Amazon.EC2.Model.LaunchTemplateSpecification -
Property @{
            LaunchTemplateId = 'lt-0abcd290751193123';
            Version          = '4'
    }
```

```
) `
    -BlockDeviceMapping  (
        New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping -Property @{
            DeviceName = '/dev/xvdb';
            EBS        = (
                New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice -Property @{
                    VolumeSize = 25;
                    VolumeType = 'gp3'
                }
            )
        }
)
```

인스턴스가 시작하지 않거나 상태가 `terminated`이 아닌 `running`로 변경되는 경우, Amazon EC2 인스턴스 시작 문제 해결 섹션을 참조하세요.

시작 템플릿을 사용하여 Amazon EC2 Auto Scaling 그룹에서 인스턴스를 시작합니다.

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.

시작 템플릿을 사용하여 Auto Scaling 그룹을 생성하려면 먼저 Auto Scaling 그룹의 인스턴스를 시작하는 데 필요한 파라미터를 포함하는 시작 템플릿을 생성해야 합니다. AMI의 ID와 같은 일부 파라미터는 필수이며, 일부 파라미터는 Auto Scaling 그룹에서 사용할 수 없습니다. 콘솔에 Amazon EC2 Auto Scaling에서 사용할 수 있는 템플릿을 생성하는 데 도움이 되는 지침이 나와 있습니다.

콘솔을 사용하여 시작 템플릿으로 Auto Scaling 그룹을 생성하려면 다음을 수행하세요.

- 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 시작 템플릿을 사용하여 Auto Scaling 그룹 생성을 참조하세요.

를 사용하여 시작 템플릿으로 Auto Scaling 생성 또는 업데이트

- create-auto-scaling-group 또는 update-auto-scaling-group 명령을 사용하여 `--launch-template` 파라미터를 지정합니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 다음 주제를 참조하세요.

- Auto Scaling 그룹에 대한 시작 템플릿 생성
- 고급 설정을 사용하여 시작 템플릿 생성
- Examples for creating and managing launch templates with the AWS Command Line Interface (AWS CLI) – 다양한 파라미터 조합으로 시작 템플릿을 생성하는 방법을 보여주는 예제를 제공합니다.
- 시작 템플릿을 사용하여 Auto Scaling 생성
- Auto Scaling 업데이트

시작 템플릿을 사용하여 EC2 플릿 시작

시작 템플릿은 EC2 플릿 요청 생성 시 필수 사항입니다. 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 섹션을 참조하세요.

AWS CLI를 사용하여 시작 템플릿으로 EC2 플릿을 생성하려면

- create-fleet 명령을 사용합니다. `--launch-template-configs` 파라미터를 사용하여 시작 템플릿과 시작 템플릿에 대한 모든 재구성을 지정합니다.

시작 템플릿을 사용하여 스팟 플릿 시작

시작 템플릿은 스팟 플릿 요청 생성 시 선택 사항입니다. 시작 템플릿을 사용하지 않는 경우 시작 파라미터 값을 수동으로 지정할 수 있습니다. 시작 템플릿을 사용하는 경우 Amazon EC2는 스팟 플릿 요청을 이행할 때 연결된 시작 템플릿에 정의된 시작 파라미터가 사용됩니다. 시작 템플릿에 지정된 일부 파라미터는 재정의가 가능합니다. 자세한 내용은 스팟 플릿 생성 섹션을 참조하세요.

시작 템플릿을 사용하여 스팟 플릿 요청을 생성하려면 다음을 수행하세요.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. 스팟 인스턴스 요청을 선택합니다.
4. Launch parameters(시작 파라미터)에서 Use a launch template(시작 템플릿 사용)을 선택합니다.
5. Launch template(시작 템플릿)에서 시작 템플릿을 선택한 다음 오른쪽 필드에서 시작 템플릿 버전을 선택합니다.
6. 이 화면에서 다른 옵션을 선택하여 스팟 플릿을 구성합니다. 이러한 옵션에 대한 자세한 내용은 정의된 파라미터를 사용하여 스팟 플릿 요청 생성 섹션을 참조하세요.

7. 스팟 플릿을 생성할 준비가 되면 Launch(시작)를 선택합니다.

시작 템플릿을 사용하여 스팟 플릿 요청을 생성하려면 다음을 수행하세요.

- request-spot-fleet 명령을 사용합니다. `LaunchTemplateConfigs` 파라미터를 사용하여 시작 템플릿과 시작 템플릿에 대한 모든 재구성을 지정합니다.

기존 인스턴스의 세부 정보를 사용하여 EC2 인스턴스 시작

Amazon EC2 콘솔은 Launch more like this(기존 인스턴스를 기반으로 시작) 옵션을 제공하여 현재 인스턴스를 기본 템플릿으로 사용하여 다른 인스턴스를 시작할 수 있도록 합니다. 이 옵션을 사용하면 Amazon EC2 인스턴스 시작 마법사에서 선택한 인스턴스의 세부적인 구성 정보가 자동으로 입력됩니다.

고려 사항

- 인스턴스를 복제하지 않고, 구성 세부 정보 중 일부만 복제합니다. 인스턴스의 사본을 만드려면 해당 인스턴스에서 AMI를 생성한 후 AMI에서 추가 인스턴스를 시작하세요. 동일한 시작 세부 정보를 사용하여 인스턴스를 시작할 수 있도록 시작 템플릿을 생성합니다.
- 현재 인스턴스는 `running` 상태여야 합니다.

복사된 세부 정보

선택한 인스턴스에서 인스턴스 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 섹션을 참조하세요.
- 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
- [Windows 인스턴스] 연결 정보. 선택한 인스턴스에 구성 파일이 연결된 경우, 새 인스턴스도 동일한 파일에 자동으로 연결됩니다. 해당 구성 파일에 조인된 도메인 구성이 포함되었다면 새 인스턴스가 동일한 도메인에 조인됩니다. 도메인 조인에 대한 자세한 내용은 AWS Directory Service 관리 안내서의 Seamlessly join a Windows EC2 instance to your AWS Managed Microsoft AD Active Directory를 참조하세요.

세부 정보가 복사되지 않음

다음 구성 세부 정보는 선택한 인스턴트에서 복사되지 않습니다. 대신 마법사는 기본 설정이나 동작을 적용합니다.

- 네트워크 인터페이스 수 - 기본값은 기본 네트워크 인터페이스(eth0)인 네트워크 인터페이스 1개입니다.
- 스토리지 - AMI와 인스턴스 유형에 따라 기본 스토리지 구성이 결정됩니다.

기존 인스턴스처럼 더 많은 인스턴스를 시작하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 작업, 이미지 및 템플릿, 기존 인스턴스를 기반으로 시작을 차례로 선택합니다.
4. 인스턴스 시작 마법사가 열립니다. 이 화면에서 다른 옵션을 선택하여 필요에 따라 인스턴스 구성을 변경할 수 있습니다.

   인스턴스를 시작할 준비가 되면 Launch instance(인스턴스 시작)를 선택합니다.
5. 인스턴스가 시작하지 않거나 상태가 `terminated`이 아닌 `running`로 변경되는 경우, Amazon EC2 인스턴스 시작 문제 해결 섹션을 참조하세요.

AWS Marketplace AMI에서 Amazon EC2 인스턴스 시작

Amazon EC2 콘솔 또는 명령줄 도구를 사용하여 AWS Marketplace AMI를 구독하고 인스턴스를 시작할 수 있습니다. AWS Marketplace AMI에 대한 자세한 내용은 Amazon EC2 인스턴스용 AWS Marketplace의 유료 AMI 섹션을 참조하세요.

시작 후 AMI 구독을 취소하려면 먼저 AMI에서 시작된 모든 인스턴스를 종료해야 합니다. 자세한 내용은 AWS Marketplace 구독 관리 단원을 참조하십시오.

Amazon EC2 콘솔을 사용하여 AWS Marketplace AMI에서 인스턴스를 시작하려면 다음을 수행하세요.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. Amazon EC2 콘솔 대시보드에서 인스턴스 시작을 선택합니다.
3. (선택 사항) 이름 및 태그(Name and tags) 아래의 이름(Name)에 인스턴스를 설명하는 이름을 입력합니다.
4. 애플리케이션 및 OS 이미지(Amazon Machine Image)(Application and OS Images (Amazon Machine Image))에서 더 많은 AMI 찾아보기(Browse more AMIs)를 선택한 다음 AWS Marketplace AMIs 탭을 선택합니다. 범주를 검색하거나 검색 기능을 사용하여 적합한 AMI를 찾습니다. 제품을 선택하려면 선택(Select)을 선택합니다.
5. 선택한 제품에 대한 개요가 표시된 창이 열립니다. 요금 정보와 공급업체에서 제공한 기타 정보를 조회할 수 있습니다. 준비가 되면 다음 버튼 중 하나를 선택합니다.
   - 인스턴스 시작 시 구독 - 10단계에서 인스턴스 시작을 선택하면 구독이 시작됩니다.
   - 지금 구독 - 구독이 즉시 시작됩니다. 구독이 진행되는 동안 이 절차의 단계를 계속 진행하여 인스턴스를 구성할 수 있습니다. 신용 카드 정보에 문제가 있는 경우 계정 세부 정보를 업데이트하라는 메시지가 나타납니다.

   > **Note**
   >
   > AMI를 사용하여 인스턴스를 시작하기 전에는 제품 사용 요금이 부과되지 않습니다. 인스턴스 유형을 선택할 때 지원되는 각 인스턴스 유형의 요금을 기록해 둡니다. 제품에 추가 세금이 적용될 수도 있습니다.
6. 인스턴스 유형(Instance type)에서 인스턴스에 대한 인스턴스 유형을 선택합니다. 인스턴스 유형은 시작할 인스턴스의 하드웨어 구성과 크기를 정의합니다.

7. (선택 사항) 키 페어(로그인)(Key pair (login)) 아래의 키 페어 이름(Key pair name)에서 기존 키 페어를 선택하거나 새로 생성합니다.
8. 네트워크 설정의 방화벽(보안 그룹)에서 벤더 사양에 따라 제품을 위해 생성된 새로운 보안 그룹을 기록하세요. 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.
9. 화면의 다른 필드를 사용하여 인스턴스를 구성하고, 스토리지를 추가하고, 태그를 추가할 수 있습니다. 구성 가능한 다른 옵션에 대한 자세한 내용을 알아보려면 Amazon EC2 인스턴스 구성 파라미터에 대한 참조 섹션을 참조하세요.
10. 요약(Summary) 패널의 소프트웨어 이미지(AMI)(Software Image (AMI))에서 인스턴스를 시작하려는 AMI의 세부 정보를 확인합니다. 또한 지정한 다른 구성 세부 정보도 확인합니다. 인스턴스를 시작할 준비가 되면 인스턴스 시작(Launch instance)을 선택합니다.
11. 구독한 제품에 따라 인스턴스를 시작하는 데 몇 분 또는 그 이상 걸릴 수 있습니다. 5단계에서 인스턴스 시작 시 구독을 선택한 경우 인스턴스를 시작하려면 먼저 제품을 구독해야 합니다. 신용 카드 정보에 문제가 있는 경우 계정 세부 정보를 업데이트하라는 메시지가 나타납니다. 시작 확인 페이지가 표시되면 모든 인스턴스 보기(View all instances)를 선택하여 인스턴스 페이지로 이동합니다.

> **Note**
>
> 유휴 상태를 포함해 인스턴스가 `running` 상태인 동안 구독 요금이 청구됩니다. 인스턴스를 중지하더라도 스토리지에 대해 요금이 부과될 수 있습니다.

12. 인스턴스가 `running` 상태일 때 인스턴스에 연결할 수 있습니다. 이렇게 하려면 목록에서 인스턴스를 선택하고 연결(Connect)을 선택하고 연결 옵션을 선택합니다. 인스턴스 연결에 대한 자세한 내용은 EC2 인스턴스에 연결 주제를 참조하세요.

> **Important**
>
> 특정 사용자 이름을 사용하여 인스턴스에 연결해야 할 수 있으므로 공급업체의 사용 지침을 주의 깊게 확인하세요. 구독 세부 정보 액세스에 대한 자세한 내용을 알아보려면 AWS Marketplace 구독 관리 섹션을 참조하세요.

13. 인스턴스가 시작하지 않거나 상태가 `terminated`이 아닌 `running`로 변경되는 경우, Amazon EC2 인스턴스 시작 문제 해결 섹션을 참조하세요.

명령줄 도구를 사용하여 AWS Marketplace AMI에서 인스턴스를 시작하려면 다음을 수행하세요.

명령줄 도구를 사용하여 AWS Marketplace 제품에서 인스턴스를 시작하려면 먼저 제품을 구독해야 합니다. 다음 방법을 사용하여 제품의 AMI ID로 인스턴스를 시작할 수 있습니다.

방법	설명서
AWS CLI	run-instances 명령을 사용합니다. 자세한 내용은 AWS Command Line Interface 사용 가이드의 인스턴스 시작을 참조하세요.
AWS Tools for Windows PowerShell	New-EC2Instance 명령을 사용합니다. 자세한 내용은 Windows PowerShell을 사용하여 Amazon EC2 시작을 참조하세요.
Query API	RunInstances 요청을 사용합니다.

EC2 인스턴스에 연결

Amazon EC2 인스턴스는 AWS 클라우드에 있는 가상 서버입니다. 인스턴스에 로그인하려면 인스턴스 연결을 설정해야 합니다. 인스턴스에 연결하는 방법은 인스턴스의 운영 체제와 인스턴스에 연결하는 데 사용하는 컴퓨터의 운영 체제에 따라 다릅니다. 다음 표에서는 각 연결 방법의 요구 사항에 대해 자세히 설명합니다.

연결 옵션	인스턴스 운영 체제	인바운드 트래픽 규칙	IAM 권한	인스턴스 프로파일 역할	인스턴스의 소프트웨어	연결 시스템의 소프트웨어	키 페어
SSH 클라이언트	Linux	예	아니요	아니요	아니요	예	예
EC2 Instance Connect	Linux	예	예	아니요	예 [1]	아니요	아니요
PuTTY	Linux	예	아니요	아니요	아니요	예	예
RDP 클라이언트	Windows	예	아니요	아니요	아니요	예	예 [2]

연결 옵션	인스턴스 운영 체제	인바운드 트래픽 규칙	IAM 권한	인스턴스 프로파일 역할	인스턴스의 소프트웨어	연결 시스템의 소프트웨어	키 페어
Fleet Manager	Windows	아니요	예	예	예 [1]	아니요	예
세션 관리자	Linux, Windows	아니요	예	예	예 [1]	아니요	아니요
EC2 Instance Connect 엔드포인트	Linux, Windows	예	예	아니요	아니요	아니요	예

[1] 필수 소프트웨어는 특정 AMI에만 사전 설치되어 있습니다. 지원되는 운영 체제에 필수 소프트웨어를 필요에 따라 수동으로 설치할 수 있습니다.

[2] 키 페어는 로컬 관리자 사용자 계정에 무작위로 생성된 비밀번호를 사용하는 경우에만 필요합니다.

자세한 내용은 사용하려는 연결 옵션에 대한 설명서를 참조하세요.

연결 옵션

- SSH 클라이언트를 사용하여 Linux 인스턴스에 연결
- PuTTY를 사용하여 Linux 인스턴스에 연결
- RDP 클라이언트를 사용하여 Windows 인스턴스에 연결
- Fleet Manager를 사용하여 Windows 인스턴스에 연결
- 세션 관리자를 사용하여 연결
- EC2 Instance Connect를 사용한 연결
- EC2 Instance Connect 엔드포인트를 사용하여 연결

일반 연결 사전 조건

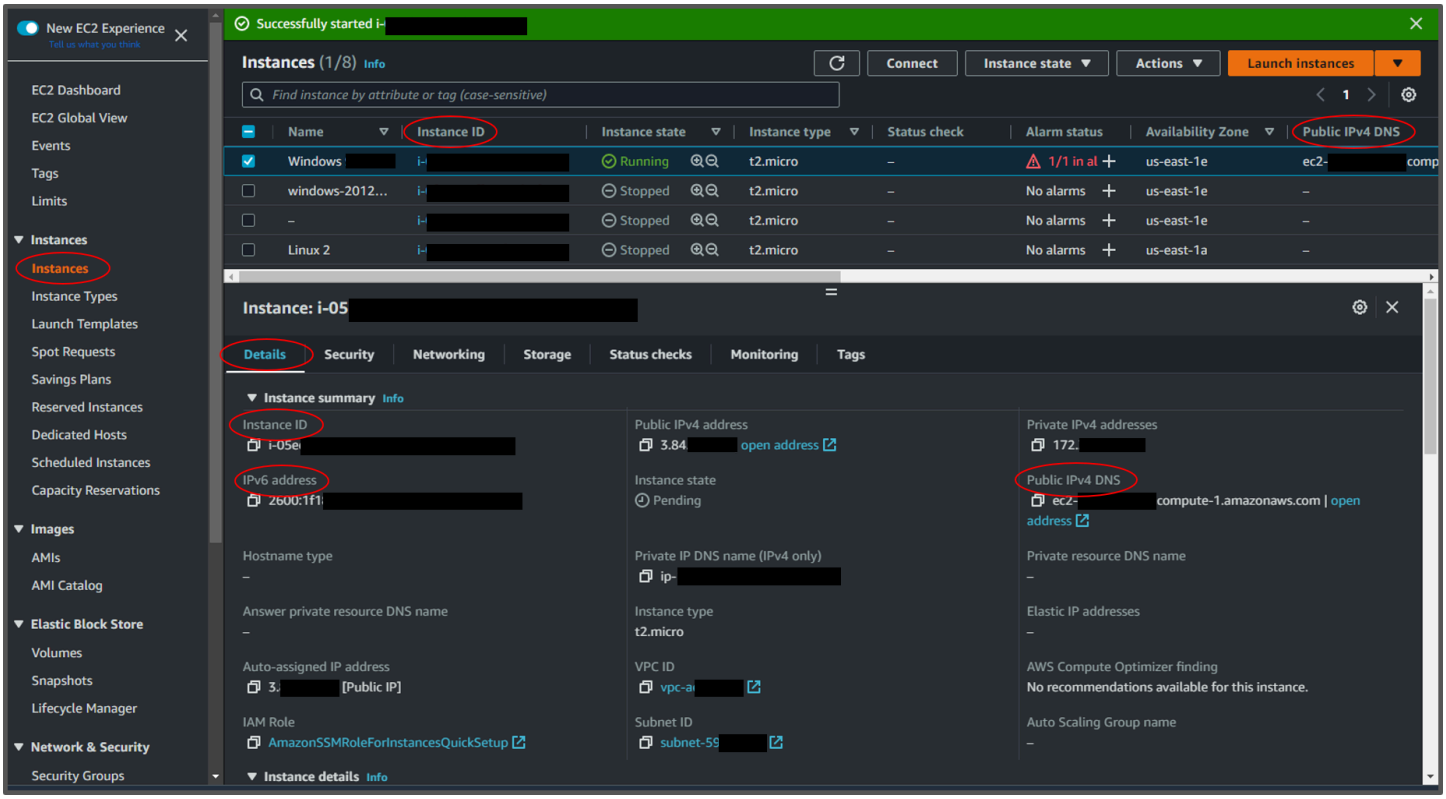
다음은 인스턴스에 연결하기 위한 일반적인 사전 조건입니다. 선택한 연결 옵션과 관련된 추가 사전 조건이 있을 수 있습니다.

일반적인 사전 요구 사항

- 인스턴스가 상태 확인을 통과했는지 확인합니다. 인스턴스가 연결 요청을 수락할 준비가 될 때까지 몇 분 정도 걸릴 수 있습니다. 자세한 내용은 상태 확인 보기 섹션을 참조하세요.
- 필수 인스턴스 세부 정보 가져오기.
- 프라이빗 키 찾기 및 권한 설정.
- (선택 사항) 인스턴스 지문 가져오기.

필수 인스턴스 세부 정보 가져오기

인스턴스 연결을 준비하려면 Amazon EC2 콘솔에서 또는 명령줄을 사용하여 다음 정보를 확인하세요.



- 인스턴스의 퍼블릭 DNS 이름을 가져옵니다.

  Amazon EC2 콘솔에서 인스턴스의 퍼블릭 DNS를 가져올 수 있습니다. 인스턴스 창의 퍼블릭 IPv4 DNS 열을 확인합니다. 이 열이 숨겨져 있는 경우 화면의 오른쪽 위에서 설정 아이콘 ( ) 을 선택하고 퍼블릭 IPv4 DNS를 선택합니다. 인스턴스 창의 인스턴스 정보 섹션에서도 퍼블릭 DNS를 찾을 수 있습니다. Amazon EC2 콘솔의 인스턴스 창에서 인스턴스를 선택하면 해당 인스턴스에 대한 정보가 페이지 하단에 표시됩니다. 세부 정보 탭에서 퍼블릭 IPv4 DNS를 찾습니다.

원하는 경우 describe-instances(AWS CLI) 또는 Get-EC2Instance(AWS Tools for Windows PowerShell) 명령을 사용할 수 있습니다.

퍼블릭 IPv4 DNS가 표시되지 않는 경우 인스턴스 상태가 실행 중이고 프라이빗 서브넷에서 인스턴스를 시작하지 않았는지 확인합니다. 인스턴스 시작 마법사를 사용하여 인스턴스를 시작한 경우 네트워크 설정에서 퍼블릭 IP 자동 할당 필드를 편집하고 값을 비활성화로 변경했을 수 있습니다. 퍼블릭 IP 자동 할당 옵션을 비활성화하면 시작 시 인스턴스에 퍼블릭 IP 주소가 할당되지 않습니다.

- (IPv6 전용 인스턴스) 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 콘솔에서 인스턴스의 IPv6 주소를 가져올 수 있습니다. 인스턴스 창의 IPv6 IP 열을 확인합니다. 또는 인스턴스 정보 섹션에서 IPv6 주소를 찾을 수 있습니다. Amazon EC2 콘솔의 인스턴스 창에서 인스턴스를 선택하면 해당 인스턴스에 대한 정보가 페이지 하단에 표시됩니다. 세부 정보 탭에서 IPv6 주소를 찾습니다.

  원하는 경우 describe-instances(AWS CLI) 또는 Get-EC2Instance(AWS Tools for Windows PowerShell) 명령을 사용할 수 있습니다. IPv6에 대한 자세한 내용은 IPv6 주소 섹션을 참조하세요.

- (Linux 인스턴스) 인스턴스의 사용자 이름을 가져옵니다.

  사용자 계정의 사용자 이름 또는 인스턴스를 시작하는 데 사용한 AMI의 기본 사용자 이름을 사용하여 인스턴스에 연결할 수 있습니다.

  - 사용자 계정의 사용자 이름을 가져옵니다.

    사용자 계정을 생성하는 방법에 대한 자세한 내용은 Amazon EC2 Linux 인스턴스에서 시스템 사용자 관리 섹션을 참조하세요.

  - 인스턴스를 시작하는 데 사용한 AMI의 기본 사용자 이름을 가져옵니다.
    - Amazon Linux – `ec2-user`
    - CentOS – `centos` 또는 `ec2-user`
    - Debian – `admin`
    - Fedora – `fedora` 또는 `ec2-user`
    - RHEL – `ec2-user` 또는 `root`
    - SUSE – `ec2-user` 또는 `root`
    - Ubuntu – `ubuntu`
    - Oracle – `ec2-user`

- Bitnami – `bitnami`
- Rocky Linux – `rocky`
- 기타 - AMI 제공업체에 문의

프라이빗 키 찾기 및 권한 설정

SSH를 사용하는 Linux 인스턴스 또는 RDP를 사용하는 Windows 인스턴스에 초기 연결을 설정하려면 프라이빗 키 파일의 위치를 알아야 합니다. SSH 연결의 경우 사용자만 프라이빗 키를 읽을 수 있도록 파일 권한을 설정해야 합니다.

Amazon EC2를 사용할 때 키 페어가 작동하는 방식에 대한 자세한 내용은 Amazon EC2 키 페어 및 Amazon EC2 인스턴스 섹션을 참조하세요.

- 프라이빗 키를 찾습니다.

  인스턴스를 시작할 때 지정한 키 페어를 찾기 위해 `.pem` 파일의 컴퓨터 상 위치에 대한 정규화된 경로를 얻습니다. 자세한 내용은 the section called “시작 시 지정된 퍼블릭 키 식별” 섹션을 참조하세요.

  프라이빗 키 파일을 찾을 수 없는 경우 프라이빗 키를 분실했습니다. 인스턴스에 연결하려면 어떻게 해야 하나요? 섹션을 참조하세요.

  (Linux 인스턴스) PuTTY를 사용하여 인스턴스에 연결하고 `.pem` 파일을 `.ppk`로 변환해야 하는 경우 PuTTYgen을 사용하여 프라이빗 키 변환의 내용을 참조하세요.
- (Linux 인스턴스) 사용자만 읽을 수 있도록 프라이빗 키의 권한을 설정하세요.
  - macOS 또는 Linux에서 연결

    macOS 또는 Linux 컴퓨터에서 SSH 클라이언트를 사용하여 Linux 인스턴스에 연결할 계획이면 사용자만 프라이빗 키 파일을 읽을 수 있도록 다음 명령으로 해당 권한을 설정합니다.

    ```
    chmod 400 key-pair-name.pem
    ```

    이러한 권한을 설정하지 않으면 이 키 페어를 사용하여 인스턴스에 연결할 수 없습니다. 자세한 내용은 오류: 보호되지 않는 프라이빗 키 파일 섹션을 참조하세요.
  - Windows에서 연결

파일 탐색기를 열고 .pem 파일을 마우스 오른쪽 버튼으로 클릭합니다. 속성 > 보안 탭을 선택하고 고급을 선택합니다. 상속 비활성화를 선택합니다. 현재 사용자를 제외한 모든 사용자에 대한 액세스 권한을 제거합니다.

(선택 사항) 인스턴스 지문 가져오기

중간자 공격으로부터 보호하려면 표시되는 지문을 확인하여 연결하려는 인스턴스의 신뢰성을 확인합니다. 지문 확인 기능은 타사가 제공한 퍼블릭 AMI에서 인스턴스를 시작하는 경우 유용합니다.

태스크 개요

먼저 인스턴스에서 인스턴스 지문을 가져옵니다. 인스턴스에 연결할 때 지문을 확인하라는 메시지가 표시되면 이 절차에서 가져온 지문을 표시된 지문을 비교합니다. 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

- 인스턴스는 pending 상태가 아니어야 합니다. 지문은 인스턴스의 첫 번째 부팅이 완료된 후에만 사용할 수 있습니다.
- 콘솔 출력을 가져오려면 인스턴스 소유자여야 합니다.
- 인스턴스 지문을 가져오는 방법은 여러 가지가 있습니다. AWS CLI를 사용하려면 먼저 로컬 컴퓨터에 설치해야 합니다. AWS CLI의 설치에 대한 자세한 내용은 AWS Command Line Interface 사용 안내서의 AWS CLI 시작하기 섹션을 참조하세요.

인스턴스 지문을 가져오려면

1단계에서는 인스턴스 지문이 포함된 콘솔 출력을 가져옵니다. 2단계에서는 콘솔 출력에서 인스턴스 지문을 찾습니다.

1. 다음 방법 중 하나를 사용하여 콘솔 출력을 가져옵니다.

   Console

   1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
   2. 왼쪽 탐색 창에서 인스턴스를 선택합니다.
   3. 인스턴스를 선택하고 작업, 모니터링 및 문제 해결, 시스템 로그 가져오기를 차례로 선택합니다.

AWS CLI

연결 중이 아닌 로컬 컴퓨터에서 get-console-output 명령을 사용합니다. 출력이 큰 경우 출력을 읽기 쉬운 텍스트 파일로 파이핑할 수 있습니다.

```
aws ec2 get-console-output \
    --instance-id i-1234567890abcdef0 \
    --query Output \
    --output text > temp.txt
```

PowerShell

로컬 컴퓨터에서 다음 Get-EC2ConsoleOutput cmdlet을 사용합니다.

```
$encodedOutput = (Get-EC2ConsoleOutput -InstanceId i-1234567890abcdef0).Output
[System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($encodedOutput
```

2. 콘솔 출력에서 BEGIN SSH HOST KEY FINGERPRINTS 아래에 있는 인스턴스(호스트) 지문을 찾습니다. 인스턴스 지문이 여러 개 있을 수 있습니다. 인스턴스에 연결하면 지문 중 하나만 표시됩니다.

   정확한 출력은 운영 체제, AMI 버전, AWS에서 키 페어를 생성했는지 여부에 따라 다를 수 있습니다. 출력의 예시는 다음과 같습니다.

```
ec2:#############################################################
ec2: -----BEGIN SSH HOST KEY FINGERPRINTS-----
ec2: 256 SHA256:l4UB/neBad9tvkgJf1QZWxheQmR59WgrgzEimCG6kZY no comment (ECDSA)
ec2: 256 SHA256:kpEa+rw/Uq3zxaYZN8KT501iBtJOIdHG52dFi66EEfQ no comment (ED25519)
ec2: 2048 SHA256:L8l6pepcA7iqW/jBecQjVZClUrKY+o2cHLI0iHerbVc no comment (RSA)
ec2: -----END SSH HOST KEY FINGERPRINTS-----
ec2: #############################################################
```

> **Note**
>
> 인스턴스에 연결할 때 이 지문을 참조합니다.

SSH를 사용하여 Linux 인스턴스에 연결

SSH를 사용하여 Linux 인스턴스에 연결하는 방법에는 여러 가지가 있습니다. 일부는 연결하는 로컬 컴퓨터의 운영 체제에 따라 달라집니다. 다른 방법은 EC2 Instance Connect 또는 AWS Systems Manager Session Manager와 같은 브라우저 기반이며 모든 컴퓨터에서 사용할 수 있습니다. SSH를 사용하여 Linux 인스턴스에 연결하고 명령을 실행하거나 SSH를 사용하여 로컬 컴퓨터와 인스턴스 간에 파일을 전송할 수 있습니다.

SSH를 사용하여 Linux 인스턴스에 연결하려면 먼저 다음 사전 조건을 완료하세요.

- 인스턴스가 상태 확인을 통과했는지 확인합니다. 인스턴스가 연결 요청을 수락할 준비가 될 때까지 몇 분 정도 걸릴 수 있습니다. 자세한 내용은 상태 확인 보기 섹션을 참조하세요.
- 인스턴스와 연관된 보안 그룹이 IP 주소로부터 들어오는 SSH 트래픽을 허용하는지 확인하세요. 자세한 내용은 컴퓨터에서 인스턴스 연결에 대한 규칙 섹션을 참조하세요.
- 필수 인스턴스 세부 정보 가져오기.
- 프라이빗 키 찾기 및 권한 설정.
- (선택 사항) 인스턴스 지문 가져오기.

그런 후에 다음 옵션 중 하나를 선택하고 SSH를 사용하여 Linux 인스턴스에 연결합니다.

- SSH 클라이언트를 사용하여 연결
- PuTTY를 사용한 연결
- SCP를 사용하여 파일 전송

인스턴스에 연결할 수 없고 문제 해결에 도움이 필요한 경우 Amazon EC2 Linux 인스턴스 연결 문제 해결의 내용을 참조하세요.

SSH 클라이언트를 사용하여 Linux 인스턴스에 연결

Secure Shell(SSH)을 사용하여 로컬 컴퓨터에서 Linux 인스턴스에 연결할 수 있습니다. 기타 옵션에 대한 자세한 내용은 EC2 인스턴스에 연결의 내용을 참조하세요.

> **Note**
>
> 인스턴스에 연결하려고 시도하는 동안 오류가 발생하는 경우 인스턴스가 모든 SSH 연결 사전 조건을(를) 충족하는지 확인하세요. 모든 사전 조건을 충족하는데도 여전히 Linux 인스턴스에 연결할 수 없는 경우 Amazon EC2 Linux 인스턴스 연결 문제 해결을(를) 참조하세요.

내용

- SSH 연결 사전 조건
- SSH 클라이언트를 사용하여 Linux 인스턴스에 연결

SSH 연결 사전 조건

SSH를 사용하여 Linux 인스턴스에 연결하려면 먼저 다음 작업을 완료하세요.

일반적인 사전 조건을 완료하세요.

- 인스턴스가 상태 확인을 통과했는지 확인합니다. 인스턴스가 연결 요청을 수락할 준비가 될 때까지 몇 분 정도 걸릴 수 있습니다. 자세한 내용은 상태 확인 보기 섹션을 참조하세요.
- 필수 인스턴스 세부 정보 가져오기.
- 프라이빗 키 찾기 및 권한 설정.
- (선택 사항) 인스턴스 지문 가져오기.

IP 주소에서의 인바운드 SSH 트래픽을 허용합니다.

인스턴스와 연관된 보안 그룹이 IP 주소로부터 들어오는 SSH 트래픽을 허용하는지 확인하세요. 자세한 내용은 컴퓨터에서 인스턴스 연결에 대한 규칙 섹션을 참조하세요.

(필요한 경우) 로컬 컴퓨터에 SSH 클라이언트를 설치합니다.

로컬 컴퓨터에 기본적으로 SSH 클라이언트가 설치되어 있을 수 있습니다. 터미널 창에 다음 명령을 입력하여 이를 확인할 수 있습니다. 컴퓨터가 명령을 인식하지 않는다면 SSH 클라이언트를 설치해야 합니다.

```
ssh
```

다음은 Windows에서 사용할 수 있는 몇 가지 옵션입니다. 컴퓨터가 다른 운영 체제를 실행하는 경우 해당 운영 체제 설명서에서 SSH 클라이언트 옵션을 참조하세요.

Windows에서 OpenSSL 설치

Windows에서 OpenSSH를 설치한 후 SSH를 사용하여 Windows 컴퓨터에서 Linux 인스턴스에 연결할 수 있습니다. 시작하기 전에 다음 요구 사항을 충족하는지 확인하세요.

Windows 버전

컴퓨터의 Windows 버전이 Windows Server 2019 이후 버전이어야 합니다.

이전 버전의 경우 Win32-OpenSSH를 다운로드하고 설치하세요.

PowerShell 요구 사항

PowerShell을 사용하여 Windows OS에 OpenSSH를 설치하려면 PowerShell 버전 5.1 이상을 실행해야 하며 계정은 기본 제공 관리자 그룹의 멤버여야 합니다. PowerShell에서 `$PSVersionTable.PSVersion`을 실행하여 PowerShell 버전을 확인합니다.

기본 제공 관리자 그룹의 멤버인지 확인하려면 다음 PowerShell 명령을 실행합니다.

```
(New-Object
 Security.Principal.WindowsPrincipal([Security.Principal.WindowsIdentity]::GetCurrent())).Is
```

기본 제공 관리자 그룹의 멤버인 경우 출력은 `True`입니다.

PowerShell을 사용하여 Windows용 OpenSSH를 설치하려면 다음 PowerShell 명령을 실행합니다.

```
Add-WindowsCapability -Online -Name OpenSSH.Client~~~~0.0.1.0
```

출력의 예시는 다음과 같습니다.

```
Path          :
Online        : True
RestartNeeded : False
```

PowerShell을 사용하여 Windows에서 OpenSSH를 제거하려면 다음 PowerShell 명령을 실행합니다.

```
Remove-WindowsCapability -Online -Name OpenSSH.Client~~~~0.0.1.0
```

출력의 예시는 다음과 같습니다.

```
Path          :
```

```
Online        : True
RestartNeeded : True
```

Windows Subsystem for Linux(WSL) 설치

Windows에서 WSL을 설치한 후 SSH 클라이언트와 같은 Linux 명령줄 도구를 사용하여 Windows 컴퓨터에서 Linux 인스턴스에 연결할 수 있습니다.

EC2 Windows 인스턴스에 Windows Subsystem for Linux 설치의 지침을 따르세요. Microsoft의 설치 설명서의 지침을 따르면 Linux의 Ubuntu 배포판이 설치됩니다. 원하는 경우 다른 Linux 배포판을 설치할 수 있습니다.

WSL 터미널 창에서 인스턴스 시작 시 지정한 키 페어에 대한 .pem 파일을 Windows에서 WSL로 복사합니다. 인스턴스에 연결할 때는 WSL에서 .pem 파일의 정규화된 경로를 확인하세요. Windows 하드 드라이브로 가는 경로의 지정 방식은 C 드라이브 액세스 방법을 참고하십시오.

```
cp /mnt/<Windows drive letter>/path/my-key-pair.pem ~/WSL-path/my-key-pair.pem
```

Windows Subsystem for Linux의 제거 방법은 WSL 배포 제거 방법을 참조하세요.

SSH 클라이언트를 사용하여 Linux 인스턴스에 연결

SSH 클라이언트를 사용하여 Linux 인스턴스에 연결하려면 다음 프로시저를 사용하세요.

SSH 클라이언트를 사용하여 인스턴스에 연결

1. 컴퓨터에서 터미널 창을 엽니다.
2. ssh 명령을 사용하여 인스턴스에 연결합니다. 사전 조건의 일부로 수집한 인스턴스에 대한 세부 정보가 필요합니다. 예를 들어 프라이빗 키(.pem 파일)의 위치, 사용자 이름 및 퍼블릭 DNS 이름 또는 IPv6 주소가 필요합니다. 다음은 예제 명령입니다.
   - (퍼블릭 DNS) 퍼블릭 DNS 이름을 사용하려면 다음 명령을 입력합니다.

     ```
     ssh -i /path/key-pair-name.pem instance-user-name@instance-public-dns-name
     ```

   - (IPv6) 또는 인스턴스에 IPv6 주소가 있는 경우 다음 명령을 입력하여 IPv6 주소를 사용합니다.

     ```
     ssh -i /path/key-pair-name.pem instance-user-name@2001:db8::1234:5678:1.2.3.4
     ```

   다음은 응답의 예입니다.

```
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com (198-51-100-1)'
 can't be established.
ECDSA key fingerprint is l4UB/neBad9tvkgJf1QZWxheQmR59WgrgzEimCG6kZY.
Are you sure you want to continue connecting (yes/no)?
```

3. (선택 사항) 보안 알림의 지문이 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. 자세한 내용은 인스턴스 지문 가져오기를 참조하세요.
4. **yes**를 입력합니다.

   다음과 같은 응답이 표시됩니다:

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (ECDSA) to
 the list of known hosts.
```

PuTTY를 사용하여 Linux 인스턴스에 연결

Windows용 무료 SSH 클라이언트인 PuTTY를 사용하여 Linux 인스턴스에 연결할 수 있습니다.

Windows Server 2019 이후 버전을 실행하는 경우 SSH 프로토콜을 사용한 원격 로그인을 위한 오픈 소스 연결 도구인 OpenSSH를 사용하는 것이 좋습니다.

> **Note**
>
> 인스턴스에 연결하려고 시도하는 동안 오류가 발생하는 경우 인스턴스가 모든 SSH 연결 사전 조건을(를) 충족하는지 확인하세요. 모든 사전 조건을 충족하는데도 여전히 Linux 인스턴스에 연결할 수 없는 경우 Amazon EC2 Linux 인스턴스 연결 문제 해결을(를) 참조하세요.

내용

- 사전 조건
- (선택 사항) PuTTYgen을 사용하여 프라이빗 키 변환
- Linux 인스턴스에 연결합니다

사전 조건

PuTTY를 사용하여 Linux 인스턴스에 연결하려면 먼저 다음 작업을 완료하세요.

일반적인 사전 조건을 완료하세요.

- 인스턴스가 상태 확인을 통과했는지 확인합니다. 인스턴스가 연결 요청을 수락할 준비가 될 때까지 몇 분 정도 걸릴 수 있습니다. 자세한 내용은 상태 확인 보기 섹션을 참조하세요.
- 필수 인스턴스 세부 정보 가져오기.
- 프라이빗 키 찾기 및 권한 설정.
- (선택 사항) 인스턴스 지문 가져오기.

IP 주소에서의 인바운드 SSH 트래픽을 허용합니다.

인스턴스와 연관된 보안 그룹이 IP 주소로부터 들어오는 SSH 트래픽을 허용하는지 확인하세요. 자세한 내용은 컴퓨터에서 인스턴스 연결에 대한 규칙 섹션을 참조하세요.

(필요한 경우) 로컬 컴퓨터에 PuTTY를 설치합니다.

PuTTY를 PuTTY 다운로드 페이지에서 다운로드하고 설치합니다. 이전 버전의 PuTTY가 이미 설치되어 있는 경우 최신 버전을 다운로드하는 것이 좋습니다. 전체 제품군을 설치해야 합니다.

PuTTYgen을 사용하여 프라이빗 키를 PPK 형식으로 변환합니다.

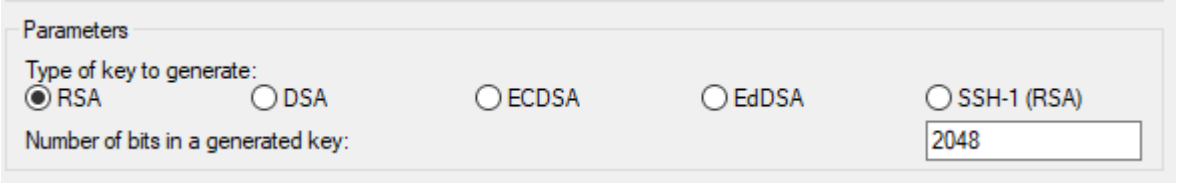
인스턴스를 시작할 때 지정한 키 페어에 대한 프라이빗 키를 지정해야 합니다. .pem 형식으로 프라이빗 키를 생성한 경우 PuTTY에서 사용할 PPK 파일로 변환해야 합니다. 프라이빗 키(.pem 파일)를 찾은 후 PuTTYgen을 사용하여 프라이빗 키 변환의 단계를 따릅니다.

(선택 사항) PuTTYgen을 사용하여 프라이빗 키 변환

PuTTY는 SSH 키의 PEM 형식을 기본적으로 지원하지 않습니다. PuTTY는 PuTTY에 필요한 PPK 형식으로 PEM 키를 변환하는 PuTTYgen이라는 도구를 제공합니다. PPK 형식 대신 PEM 형식을 사용하여 키를 생성한 경우 PuTTY에서 사용할 형식(.ppk 파일)으로 프라이빗 키(.pem 파일)를 변환해야 합니다.

프라이빗 키를 PEM에서 PPK 형식으로 변환

1. 시작 메뉴에서 모든 프로그램, PuTTY, PuTTYgen을 선택합니다.
2. Type of key to generate(생성할 키 유형)에서 RSA를 선택합니다. PuTTYgen 버전에 이 옵션이 포함되어 있지 않으면 SSH-2 RSA를 선택합니다.



3. 로드(Load)를 선택합니다. 기본적으로 PuTTYgen에는 확장명이 .ppk인 파일만 표시됩니다. .pem 파일을 찾으려면 모든 유형의 파일을 표시하는 옵션을 선택합니다.

4. 인스턴스를 시작할 때 지정한 키 페어에 대한 .pem 파일을 선택한 다음 열기를 선택합니다. PuTTYgen에 .pem 파일을 성공적으로 가져왔다는 알림이 표시됩니다. 확인을 선택합니다.
5. PuTTY에서 사용할 수 있는 형식으로 키를 저장하려면 [프라이빗 키 저장(Save private key)]을 선택합니다. 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: key-pair-name) [저장(Save)]을 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 연결합니다

PuTTY를 사용하여 Linux 인스턴스에 연결하려면 다음 프로시저를 사용하세요. 프라이빗 키에 대해 생성한 .ppk 파일이 필요합니다. 자세한 내용은 이전 섹션의 (선택 사항) PuTTYgen을 사용하여 프라이빗 키 변환를 참조하세요. 인스턴스에 연결을 시도하는 동안 오류가 발생한 경우 Amazon EC2 Linux 인스턴스 연결 문제 해결 섹션을 참조하세요.

마지막 테스트 버전 - PuTTY .78

PuTTY를 사용하여 인스턴스에 연결하려면

1. PuTTY를 시작합니다(시작 메뉴에서 PuTTY를 검색한 다음 열기 선택).
2. 범주 창에서 세션을 선택하고 다음 필드를 작성합니다.

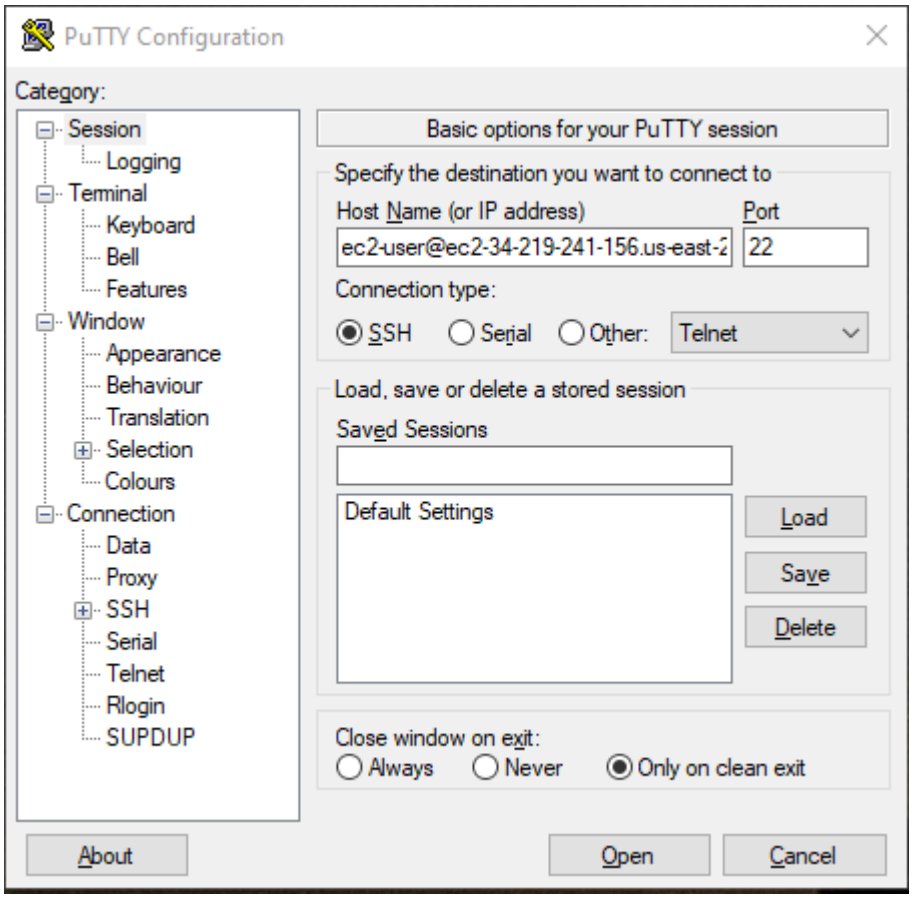
   a. 호스트 이름 상자에서 다음 중 하나를 수행합니다.

- (퍼블릭 DNS) 인스턴스의 퍼블릭 DNS 이름을 사용하여 연결하려면 *instance-user-name*@*instance-public-dns-name*를 입력합니다.
- (IPv6) 또는 인스턴스에 IPv6 주소가 있는 경우 인스턴스의 IPv6 주소를 사용하여 연결하려면 *instance-user-name*@*2001:db8::1234:5678:1.2.3.4*를 입력합니다.

인스턴스의 사용자 이름과 인스턴스의 퍼블릭 DNS 이름 또는 IPv6 주소를 가져오는 방법에 대한 자세한 내용은 필수 인스턴스 세부 정보 가져오기을(를) 참조하세요.

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 초에 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, SSH 및 Auth를 확장합니다. 자격 증명을 선택합니다.

5. 인증을 위한 프라이빗 키 파일 옆에서 찾아보기를 선택합니다. 프라이빗 키 파일 선택 대화 상자에서 키 페어에 대해 생성한 .ppk 파일을 선택합니다. 파일을 두 번 클릭하거나 프라이빗 키 파일 선택 대화 상자에서 열기를 선택할 수 있습니다.
6. (선택 사항) 이 세션 후에 이 인스턴스에 다시 연결하려는 경우 나중에 사용할 수 있도록 세션 정보를 저장할 수 있습니다. 범주 창에서 세션을 선택합니다. 저장된 세션에 세션 이름을 입력한 다음 저장을 선택합니다.
7. 인스턴스에 연결하려면 열기를 선택합니다.
8. 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 가져오기에서 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. 수락을 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 Amazon EC2 Linux 인스턴스 연결 문제 해결 섹션을 참조하세요.

SCP를 사용하여 Linux 인스턴스로 파일 전송

로컬 컴퓨터와 Linux 인스턴스 간에 파일을 전송하는 한 가지 방법은 SCP(Secure Copy Protocol)를 사용하는 것입니다. SCP는 일회성 파일 복사 등 간단한 작업에 적합한 옵션입니다. SCP는 SSH를 사용한 인스턴스 연결에 사용하는 것과 동일한 .pem 파일을 사용하여 파일 전송을 보호합니다. 파일을 동기화된 상태로 유지해야 하거나 파일이 큰 경우 rsync가 SCP보다 빠르고 효율적입니다. rsync는 기본적으로 일반 텍스트를 사용하여 데이터를 전송하므로 보안을 위해서는 SSH보다 rsync를 사용하세요.

SCP를 사용하여 Linux 인스턴스에 연결하려면 먼저 다음 작업을 완료하세요.

- 일반적인 사전 조건을 완료하세요.
  - 인스턴스가 상태 확인을 통과했는지 확인합니다. 인스턴스가 연결 요청을 수락할 준비가 될 때까지 몇 분 정도 걸릴 수 있습니다. 자세한 내용은 상태 확인 보기 섹션을 참조하세요.

  - 필수 인스턴스 세부 정보 가져오기.
  - 프라이빗 키 찾기 및 권한 설정.
  - (선택 사항) 인스턴스 지문 가져오기.
- IP 주소에서의 인바운드 SSH 트래픽을 허용합니다.

  인스턴스와 연관된 보안 그룹이 IP 주소로부터 들어오는 SSH 트래픽을 허용하는지 확인하세요. 자세한 내용은 컴퓨터에서 인스턴스 연결에 대한 규칙 섹션을 참조하세요.
- SCP 클라이언트를 설치합니다.

  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 https://www.openssh.com을 참조하세요.

다음 절차에서는 인스턴스의 퍼블릭 DNS 이름, 또는 인스턴스에 있는 경우 IPv6 주소를 사용하여 SCP를 사용하는 파일 전송 단계를 안내합니다.

SCP를 사용하여 컴퓨터와 인스턴스 사이에서 파일을 전송하려면

1. 컴퓨터의 소스 파일 위치와 인스턴스의 대상 경로를 확인합니다. 다음 예제에서 프라이빗 키 파일의 이름은 `key-pair-name.pem`, 전송할 파일은 `my-file.txt`, 인스턴스에 대한 사용자 이름은 ec2-user, 인스턴스의 퍼블릭 DNS는 `instance-public-dns-name`, 인스턴스의 IPv6 주소를 `2001:db8::1234:5678:1.2.3.4`입니다.
   - (퍼블릭 DNS) 인스턴스의 대상으로 파일을 전송하려면 컴퓨터에서 다음 명령을 입력합니다.

     ```
     scp -i /path/key-pair-name.pem /path/my-file.txt ec2-user@instance-public-dns-name:path/
     ```
   - (IPv6) 인스턴스에 IPv6 주소가 있는 경우 인스턴스의 대상으로 파일을 전송하려면 컴퓨터에서 다음 명령을 입력합니다. IPv6 주소는 이스케이프된([ ]) 대괄호(\)로 묶어야 합니다.

     ```
     scp -i /path/key-pair-name.pem /path/my-file.txt ec2-user@
     \[2001:db8::1234:5678:1.2.3.4\]:path/
     ```
2. SSH를 사용하여 인스턴스에 아직 연결하지 않은 경우 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)?
```

(선택 사항) 보안 알림의 지문이 인스턴스 지문과 일치하는지 확인할 수 있습니다. 자세한 내용은 (선택 사항) 인스턴스 지문 가져오기 섹션을 참조하세요.

yes를 입력합니다.

3. 전송이 성공한 경우 다음과 유사한 응답이 표시됩니다.

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (RSA)
to the list of known hosts.
my-file.txt                                   100%   480     24.4KB/s   00:00
```

4. 반대 방향으로(Amazon EC2 인스턴스에서 로컬 컴퓨터로) 파일을 전송하려면 호스트 파라미터의 순서를 역순으로 지정하면 됩니다. 예를 들어 다음 예시와 같이 EC2 인스턴스에서 `my-file.txt`를 로컬 컴퓨터의 대상에서 `my-file2.txt`로 전송할 수 있습니다.

   - (퍼블릭 DNS) 컴퓨터의 대상으로 파일을 전송하려면 컴퓨터에서 다음 명령을 입력합니다.

     ```
     scp -i /path/key-pair-name.pem ec2-user@instance-public-dns-name:path/my-file.txt path/my-file2.txt
     ```

   - (IPv6) 컴퓨터에 IPv6 주소가 있는 경우 인스턴스의 대상으로 파일을 전송하려면 컴퓨터에서 다음 명령을 입력합니다. IPv6 주소는 이스케이프된([ ]) 대괄호(\)로 묶어야 합니다.

     ```
     scp -i /path/key-pair-name.pem ec2-user@\[2001:db8::1234:5678:1.2.3.4\]:path/my-file.txt path/my-file2.txt
     ```

Amazon EC2 Linux 인스턴스에서 시스템 사용자 관리

각 Linux 인스턴스는 기본 Linux 시스템 사용자로 시작됩니다. 인스턴스에 사용자를 추가하고 삭제할 수 있습니다.

기본 사용자의 경우 기본 사용자 이름은 인스턴스를 시작할 때 지정된 AMI에 의해 결정됩니다.

> **Note**
>
> 기본적으로 암호 인증 및 루트 로그인은 비활성화되어 있고 sudo는 활성화되어 있습니다. 인스턴스에 로그인하려면 키 페어를 사용해야 합니다. 로그인에 대한 자세한 내용을 알아보려면 SSH를 사용하여 Linux 인스턴스에 연결 섹션을 참조하세요.

인스턴스에 암호 인증 및 루트 로그인을 사용할 수 있습니다. 자세한 내용은 운영 체제 설명서를 참조하세요.

> **Note**
>
> Linux 시스템 사용자를 IAM 사용자와 혼동해서는 안 됩니다. 자세한 내용은 IAM 사용 설명서에서 IAM 사용자를 참조하세요.

내용

- 기본 사용자 이름
- 고려 사항
- 사용자 생성
- 사용자 제거

기본 사용자 이름

EC2 인스턴스의 기본 사용자 이름은 인스턴스를 시작할 때 지정된 AMI에 의해 결정됩니다.

기본 사용자 이름은 다음과 같습니다.

- Amazon Linux AMI의 경우 사용자 이름은 `ec2-user`입니다.
- CentOS AMI의 경우 사용자 이름은 `centos` 또는 `ec2-user`입니다.
- Debian AMI의 경우 사용자 이름은 `admin`입니다.
- Fedora AMI의 경우 사용자 이름은 `fedora` 또는 `ec2-user`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.
- Oracle AMI의 경우 사용자 이름은 `ec2-user`입니다.
- Bitnami AMI의 경우 사용자 이름은 `bitnami`입니다.

> **Note**
>
> 다른 Linux 배포에 사용할 기본 사용자 이름을 찾으려면 AMI 제공업체에 문의하세요.

고려 사항

기본 사용자를 사용하면 많은 애플리케이션에 적합합니다. 그러나 개인이 자신의 파일과 작업 영역을 가질 수 있도록 사용자를 추가하도록 선택할 수 있습니다. 게다가 신규 사용자를 생성하는 방법은 사용이 미숙할 수 있는 여러 사용자에게 기본 사용자 액세스를 허용하는 방법보다 보안상 훨씬 안전합니다. 기본 사용자는 잘못 사용될 경우 시스템에 심각한 손상을 줄 수 있기 때문입니다. 자세한 내용은 EC2 인스턴스의 보안 유지를 위한 팁을 참조하세요.

Linux 시스템 사용자를 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 단원을 참조하십시오.

사용자 생성

먼저 사용자를 생성한 다음, 사용자의 연결을 허용하는 SSH 퍼블릭 키를 추가하고 인스턴스에 로그인하세요.

> **Important**
>
> 이 절차의 1단계에서는 새 키 페어를 생성합니다. 키 페어는 암호와 같은 기능을 하기 때문에 안전하게 처리하는 것이 중요합니다. 사용자를 위한 키 페어를 생성하는 경우 프라이빗 키가 안전하게 전송되도록 해야 합니다. 또는 사용자가 자체 키 페어를 생성하고 프라이빗 키를 컴퓨터에 안전하게 보관한 다음 퍼블릭 키를 보내 3단계의 절차를 완료하여 1단계와 2단계를 완료할 수 있습니다.

사용자를 생성하는 방법

1. 새 키 페어를 생성합니다. 사용자를 생성할 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 사용자를 생성하여 /etc/passwd 파일의 항목으로 시스템에 추가합니다. 이 명령은 사용자에 대한 그룹과 홈 디렉터리도 생성합니다. 이 예제에서 사용자의 이름은 *newuser*입니다.

   - AL2023 및 Amazon Linux 2

     AL2023 및 Amazon Linux 2에서는 기본적으로 암호 인증이 비활성화된 사용자가 생성됩니다.

     ```
     [ec2-user ~]$ sudo adduser newuser
     ```

   - Ubuntu

     암호 인증이 비활성화된 사용자를 생성하려면 --disabled-password 파라미터를 포함합니다.

     ```
     [ubuntu ~]$ sudo adduser newuser --disabled-password
     ```

5. 생성할 디렉터리와 파일이 정확한 소유권을 가질 수 있도록 새 사용자로 전환합니다.

   ```
   [ec2-user ~]$ sudo su - newuser
   ```

   프롬프트가 ec2-user에서 *newuser*로 바뀌며 쉘 세션이 새 사용자로 전환된 것을 나타냅니다.
6. 사용자에게 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_keys라는 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 *newuser* 파일에 추가한 퍼블릭 키에 해당하는 프라이빗 키를 사용하여 인스턴스에서 authorized_keys 사용자에게 로그인할 수 있습니다. Linux 인스턴스에 연결하는 다양한 방법에 대한 자세한 내용은 SSH를 사용하여 Linux 인스턴스에 연결 섹션을 참조하세요.

사용자 제거

사용자가 더 이상 필요하지 않은 경우 더 이상 사용할 수 없도록 사용자를 제거할 수 있습니다.

userdel 명령으로 시스템에서 사용자를 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
```

RDP를 사용하여 Windows 인스턴스에 연결

원격 데스크톱을 사용하여 대부분의 Windows Amazon Machine Image(AMI)에서 생성되는 Amazon EC2 인스턴스에 연결할 수 있습니다. 원격 데스크톱은 원격 데스크톱 프로토콜(RDP)을 사용하며 바로 앞에 있는 컴퓨터(로컬 컴퓨터)를 사용하는 것처럼 인스턴스를 연결하여 사용할 수 있도록 해줍니다. 대부분의 Windows 버전과 Mac OS에서도 사용할 수 있습니다.

Windows Server 운영 체제 라이선스는 관리 목적으로 두 개의 동시 원격 연결을 허용합니다. Windows 인스턴스 가격에는 Windows Server 라이선스가 포함됩니다. 2개를 초과하는 동시 원격 연결이 필요할 경우, 원격 데스크톱 서비스(RDS) 라이선스를 구매해야 합니다. 제3의 연결을 시도하면 오류가 발생합니다.

> **Tip**
>
> AWS Nitro 시스템을 기반으로 구축된 인스턴스의 부팅, 네트워크 구성 및 기타 문제를 해결하기 위해 인스턴스에 연결해야 하는 경우 인스턴스용 EC2 직렬 콘솔을 사용할 수 있습니다.

내용

- RDP 클라이언트를 사용하여 Windows 인스턴스에 연결
- Fleet Manager를 사용하여 Windows 인스턴스에 연결
- RDP를 사용하여 Windows 인스턴스로 파일 전송

RDP 클라이언트를 사용하여 Windows 인스턴스에 연결

다음과 같이 RDP 클라이언트를 사용하여 Windows 인스턴스에 연결할 수 있습니다.

> **Tip**
>
> Systems Manager Fleet Manager 또는 EC2 Instance Connect 엔드포인트를 사용하여 Windows 인스턴스에 연결할 수도 있습니다.

사전 조건

RDP 클라이언트를 사용하여 Windows 인스턴스에 연결하려면 다음 사전 요구 사항을 충족해야 합니다.

- 일반적인 사전 조건을 완료하세요.
  - 인스턴스가 상태 확인을 통과했는지 확인합니다. 인스턴스가 연결 요청을 수락할 준비가 될 때까지 몇 분 정도 걸릴 수 있습니다. 자세한 내용은 상태 확인 보기 단원을 참조하십시오.
  - 필수 인스턴스 세부 정보 가져오기.
  - 프라이빗 키 찾기 및 권한 설정.
  - (선택 사항) 인스턴스 지문 가져오기.
- RDP 클라이언트를 설치합니다.
  - (Windows) Windows에는 기본적으로 RDP 클라이언트가 포함되어 있습니다. 확인하려면 명령 프롬프트 창에 mstsc를 입력합니다. 컴퓨터에서이 명령을 인식하지 못하면 Microsoft Store에서 Microsoft 원격 데스크톱 앱을 다운로드합니다.
  - (macOS X) Mac App Store에서 Windows App for Mac(이전의 Microsoft 원격 데스크톱)을 다운로드합니다.
  - (Linux) Remmina를 사용합니다.
- IP 주소에서 인바운드 RDP 트래픽을 허용합니다.

  인스턴스와 연관된 보안 그룹이 IP 주소로부터 들어오는 RDP 트래픽을 허용하는지 확인하십시오. 자세한 내용은 컴퓨터에서 인스턴스 연결에 대한 규칙 단원을 참조하십시오.

관리자 암호 검색

인스턴스를 도메인에 조인한 경우 AWS Directory Service의 도메인 자격 증명을 사용하여 인스턴스에 연결할 수 있습니다. 원격 데스크톱 로그인 화면에서 로컬 컴퓨터 이름과 생성된 암호를 사용하는 대신 관리자의 정규화된 사용자 이름(예: **`corp.example.com\Admin`**)과 이 계정의 암호를 사용합니다.

RDP를 사용하여 Windows 인스턴스에 연결하려면 최초 관리자 암호를 검색한 다음 인스턴스에 연결할 때 이 암호를 입력해야 합니다. 인스턴스를 시작한 후 암호를 사용할 수 있으려면 몇 분 정도 걸립니

다. 계정에 GetPasswordData 작업을 호출할 권한이 있어야 합니다. 자세한 내용은 Amazon EC2 API에 대한 액세스를 제어하는 정책 예제 단원을 참조하십시오.

관리자 계정의 기본 사용자 이름은 AMI에 포함된 운영 체제(OS)의 언어에 따라 다릅니다. 올바른 사용자 이름을 확인하려면 운영 체제의 언어를 확인한 다음 해당 사용자 이름을 선택합니다. 예를 들어, 영어 OS의 경우 사용자 이름은 `Administrator`이고, 프랑스 OS의 경우 사용자 이름은 `Administrateur`이며, 포르투갈어 OS의 경우 사용자 이름은 `Administrador`입니다. OS의 언어 버전에 해당 언어의 사용자 이름이 없는 경우 사용자 이름 `Administrator (Other)`를 선택합니다. 자세한 내용은 Microsoft 웹 사이트의 Localized Names for Administrator Account in Windows를 참조하세요.

최초 관리자 암호 검색

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances(인스턴스)를 선택합니다.
3. 찾은 인스턴스를 선택한 다음 [Connect]를 선택합니다.
4. 인스턴스에 연결 페이지에서 RDP 클라이언트 탭을 선택합니다.
5. 관리자 계정의 기본 사용자 이름을 사용자 이름으로 선택합니다. 선택한 사용자 이름은 인스턴스를 시작하는 데 사용한 AMI에 포함된 운영 체제(OS)의 언어와 일치해야 합니다. 운영 체제와 동일한 언어의 사용자 이름이 없는 경우 관리자(기타)를 선택합니다.
6. 암호 가져오기를 선택합니다.
7. Windows 암호 가져오기 페이지에서 다음을 수행하세요.

   a. 프라이빗 키 파일 업로드를 선택하고 인스턴스를 시작할 때 지정한 프라이빗 키(`.pem`) 파일로 이동합니다. 파일을 선택하고 [열기(Open)]를 클릭하여 파일의 전체 콘텐츠를 이 창에 복사합니다.

   b. 암호 해독을 선택합니다. Windows 암호 가져오기 페이지가 닫히고 이전에 표시된 암호 가져오기 링크를 대체하는 암호 아래에 인스턴스의 기본 관리자 암호가 표시됩니다.

   c. 암호를 복사하여 안전한 장소에 저장합니다. 이 암호는 인스턴스에 연결하는 데 필요합니다.

Windows 인스턴스에 연결

다음 절차에서는 Windows용 원격 데스크톱 연결 클라이언트(MSTSC)를 사용합니다. 다른 RDP 클라이언트를 사용하는 경우 RDP 파일을 다운로드한 다음 RDP 클라이언트 설명서에서 RDP 연결 설정 단계를 참조하세요.

RDP 클라이언트를 사용하여 Windows 인스턴스에 연결

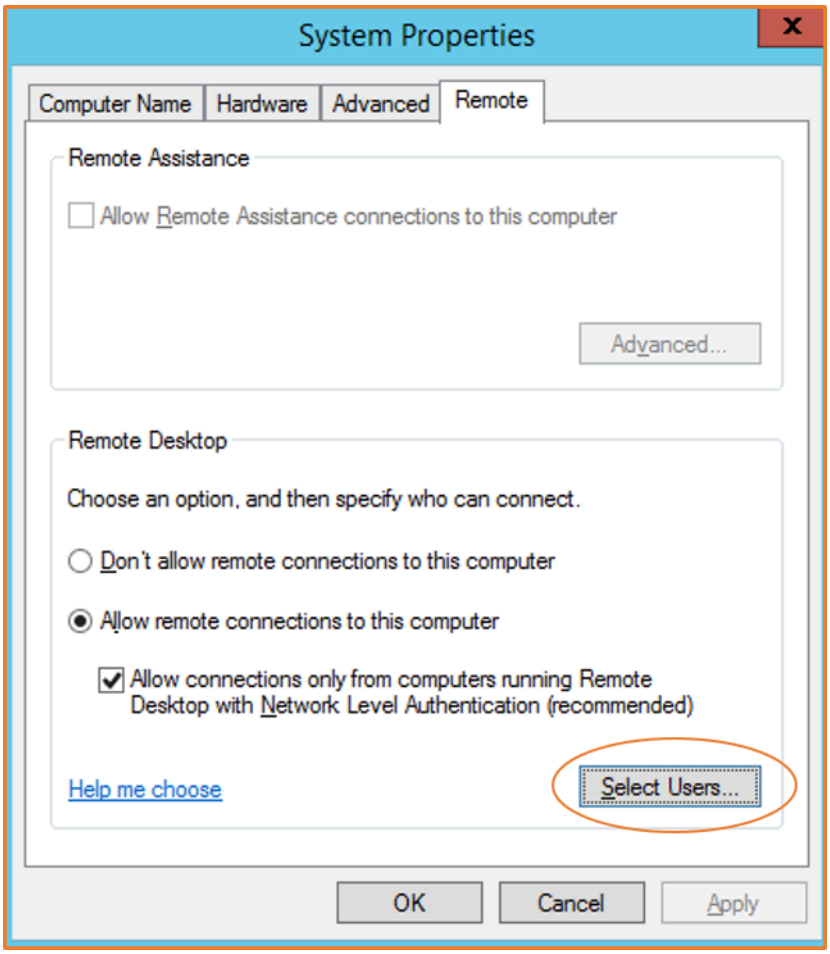
1. 인스턴스에 연결 페이지에서 원격 데스크톱 파일 다운로드를 선택합니다. 파일 다운로드가 완료되면 취소를 선택하여 인스턴스 페이지로 돌아갑니다. RDP 파일이 Downloads 폴더에 다운로드됩니다.
2. mstsc.exe를 실행하여 RDP 클라이언트를 엽니다.
3. 옵션 표시를 확장하고 열기를 선택한 다음 Downloads 폴더에서.rdp 파일을 선택합니다.
4. 기본적으로 컴퓨터는 인스턴스의 퍼블릭 IPv4 DNS 이름이고 사용자 이름은 관리자 계정입니다. 대신 IPv6를 사용하여 인스턴스에 연결하려면 인스턴스의 퍼블릭 IPv4 DNS 이름을 해당 IPv6 주소로 바꿉니다. 기본 설정을 검토하고 필요에 따라 변경합니다.
5. 연결을 선택합니다. 원격 연결 게시자를 알 수 없다는 경고가 표시되면 연결을 선택하여 계속합니다.
6. 앞서 저장한 암호를 입력한 후 확인을 선택합니다.
7. 자체 서명된 인증서의 특성으로 인해, 보안 인증서를 인증할 수 없다는 경고 메시지가 나타날 수도 있습니다. 다음 중 하나를 수행합니다.
   - 인증서를 신뢰하는 경우 예를 선택하여 인스턴스에 연결합니다.
   - [Windows] 계속하기 전에 인증서의 지문을 시스템 로그의 값과 비교하여 원격 컴퓨터의 ID를 확인합니다. 인증서 보기를 선택한 다음 세부 정보 탭에서 지문을 선택합니다. 이 값을 작업, 모니터링 및 문제 해결, 시스템 로그 가져오기의 RDPCERTIFICATE-THUMBPRINT 값과 비교합니다.
   - [Mac OS X] 계속하기 전에 인증서의 지문을 시스템 로그의 값과 비교하여 원격 컴퓨터의 ID를 확인합니다. 인증서 보기를 선택하고 세부 정보를 확장한 다음 SHA1 지문을 선택합니다. 이 값을 작업, 모니터링 및 문제 해결, 시스템 로그 가져오기의 RDPCERTIFICATE-THUMBPRINT 값과 비교합니다.
8. RDP 연결에 성공하면 RDP 클라이언트는 Windows 로그인 화면을 표시한 다음 Windows 데스크톱을 표시합니다. 오류 메시지가 표시되면 the section called “원격 데스크톱으로 원격 컴퓨터에 연결할 수 없음”의 내용을 참조하세요. RDP 연결을 완료하면 RDP 클라이언트를 닫을 수 있습니다.

사용자 계정 구성

RDP를 통해 인스턴스에 연결한 후에는 다음 작업을 수행하는 것이 좋습니다.

- 기본값으로 제공된 관리자 암호를 변경합니다. Windows Server를 실행하는 다른 컴퓨터처럼 인스턴스 자체에 로그온한 상태에서 암호를 변경할 수 있습니다.

- 인스턴스에서 관리자 권한을 사용하여 또 다른 사용자를 생성합니다. 이는 관리자 암호를 잊어버리거나 관리자 계정에 문제가 있는 경우를 위한 보호 수단입니다. 새 사용자에게 인스턴스에 원격으로 액세스할 수 있는 권한이 있어야 합니다. Windows 바탕 화면이나 파일 탐색기에서 이 PC 아이콘을 마우스 오른쪽 단추로 클릭하고 속성을 선택하여 시스템 속성을 엽니다. 원격 설정을 선택하고 사용자 선택을 선택하여 사용자를 원격 데스크톱 사용자 그룹에 추가합니다.



Fleet Manager를 사용하여 Windows 인스턴스에 연결

AWS Systems Manager의 기능인 Fleet Manager를 사용하면 RDP(Remote Desktop Protocol)를 사용하여 Windows 인스턴스에 연결하고 AWS Management Console의 동일한 페이지에 최대 4개의 Windows 인스턴스를 표시할 수 있습니다. Amazon EC2 콘솔의 인스턴스 페이지에서 Fleet Manager 원격 데스크톱의 첫 번째 인스턴스에 직접 연결할 수 있습니다. Fleet Manager에 대한 자세한 내용은 AWS Systems Manager 사용 설명서의 원격 데스크톱을 사용하여 관리형 인스턴스에 연결을 참조하세요.

Fleet Manager를 사용하여 연결하는 경우 IP 주소로부터 수신 중인 RDP 트래픽을 특별히 허용할 필요가 없습니다. Fleet Manager가 알아서 처리합니다.

사전 조건

Fleet Manager를 사용하여 인스턴스에 연결하기 전에 사용자 환경을 설정해야 합니다. 자세한 내용은 AWS Systems Manager 사용 설명서의 환경 설정을 참조하세요.

Fleet Manager를 사용하여 Windows 인스턴스에 연결

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances(인스턴스)를 선택합니다.
3. 찾은 인스턴스를 선택한 다음 [Connect]를 선택합니다.
4. RDP 클라이언트 탭의 연결 유형에서 Fleet Manager를 사용하여 연결을 선택합니다.
5. Fleet Manager 원격 데스크톱을 선택합니다. 그러면 AWS Systems Manager 콘솔에서 Fleet Manager Remote Desktop(Fleet Manager 원격 데스크톱) 페이지가 열립니다.
6. 자격 증명을 입력한 다음 연결을 선택합니다.
7. RDP 연결이 성공하면 Fleet Manager에 Windows 데스크톱이 표시됩니다. 세션을 완료하면 작업, 세션 종료를 선택합니다.

자세한 내용을 알아보려면 AWS Systems Manager 사용 설명서의 원격 데스크톱을 사용하여 Windows Server 관리 인스턴스에 연결을 참조하세요.

RDP를 사용하여 Windows 인스턴스로 파일 전송

다른 Windows 서버를 사용할 때와 똑같은 방식으로 Windows 인스턴스를 사용할 수 있습니다. 예를 들어, Microsoft 원격 데스크톱 연결(RDP) 소프트웨어의 로컬 파일 공유 기능을 사용하여 Windows 인스턴스와 로컬 컴퓨터 간에 파일을 전송할 수 있습니다. 하드 디스크 드라이브, DVD 드라이브, 휴대용 미디어 드라이브 및 매핑된 네트워크 드라이브에 있는 로컬 파일에 액세스할 수 있습니다.

Windows 인스턴스에서 로컬 파일에 액세스하려면 원격 세션 드라이브를 로컬 드라이브에 매핑하여 로컬 파일 공유 기능을 활성화해야 합니다. 로컬 컴퓨터 운영 체제가 Windows인지 macOS X인지에 따라 단계가 약간 다릅니다.

RDP를 사용하여 연결하기 위한 사전 조건에 대한 자세한 내용은 사전 조건 섹션을 참조하세요.

Windows

로컬 Windows 컴퓨터의 로컬 드라이브에 원격 세션 드라이브 매핑

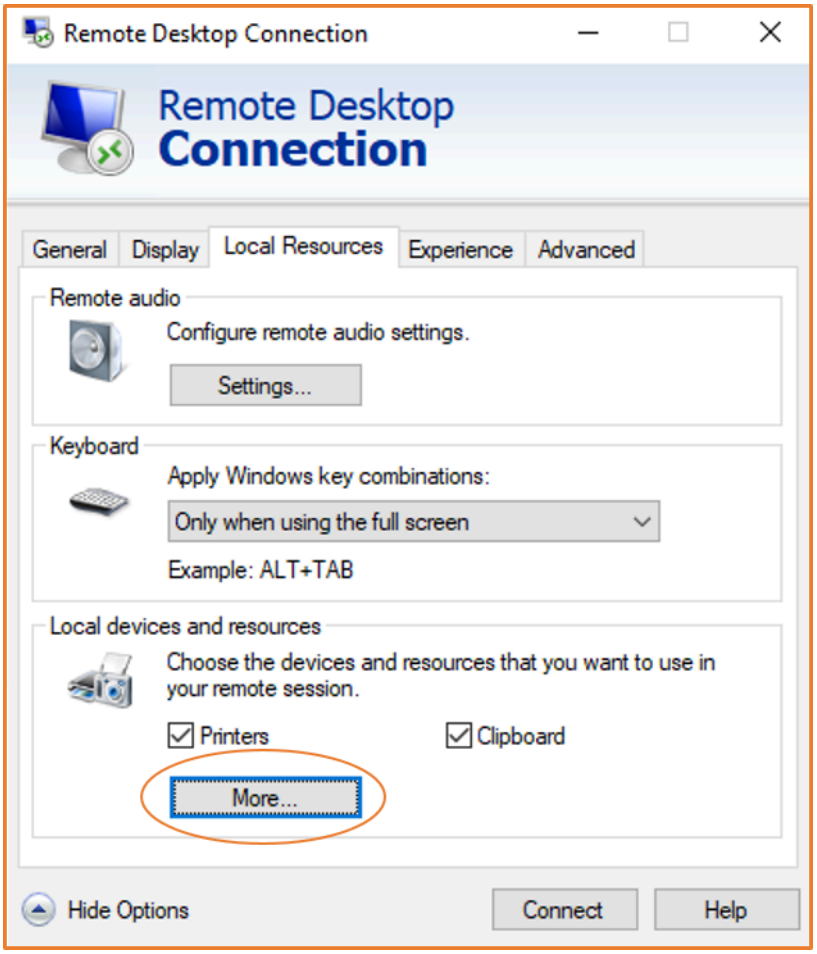
1. 원격 데스크톱 연결 클라이언트를 엽니다.

2. [옵션 보기(Show Options)]를 선택합니다.
3. 다음과 같이 컴퓨터 필드에 인스턴스 호스트 이름을 추가하고 사용자 이름 필드에 사용자 이름을 추가합니다.

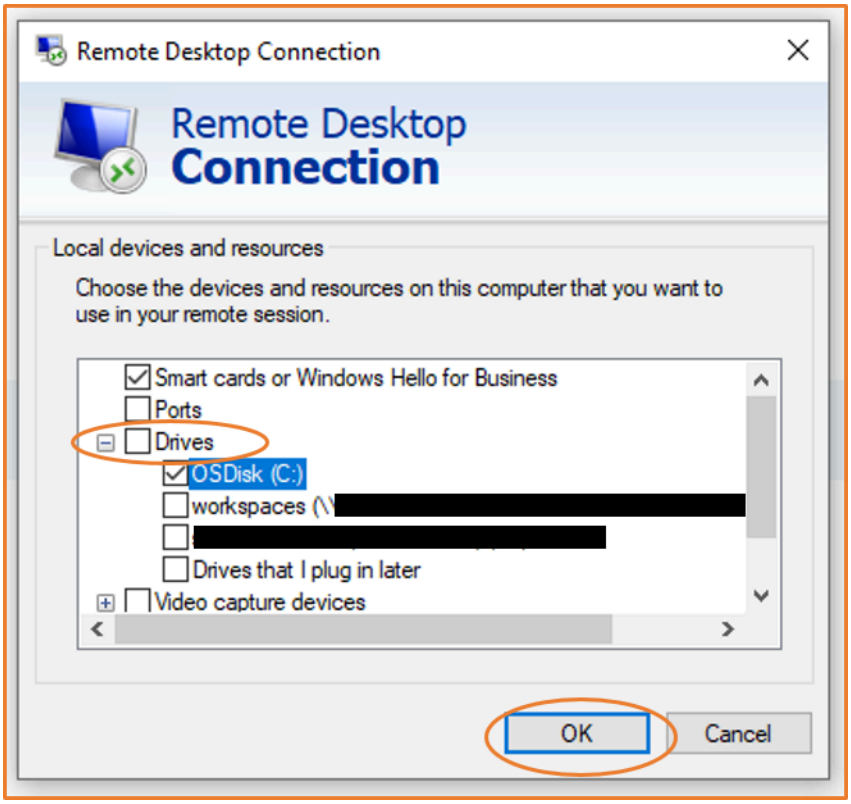
   a. 연결 설정(Connection settings)에서 열기...(Open...)를 선택하고 Amazon EC2 콘솔에서 다운로드한 RDP 바로 가기 파일을 찾습니다. 이 파일에는 인스턴스를 식별하는 퍼블릭 IPv4 DNS 호스트 이름과 관리자 사용자 이름이 포함되어 있습니다.

   b. 파일과 열기(Open)를 차례로 선택합니다. 컴퓨터(Computer) 및 사용자 이름(User name) 필드는 RDP 바로 가기 파일의 값으로 채워집니다.

   c. Save(저장)를 선택합니다.
4. 로컬 리소스 탭을 선택합니다.
5. 로컬 디바이스 및 리소스(Local devices and resources)에서 더 보기...(More...)를 선택합니다.



6. 드라이브를 열고 Windows 인스턴스에 매핑할 로컬 드라이브를 선택합니다.
7. 확인을 선택합니다.

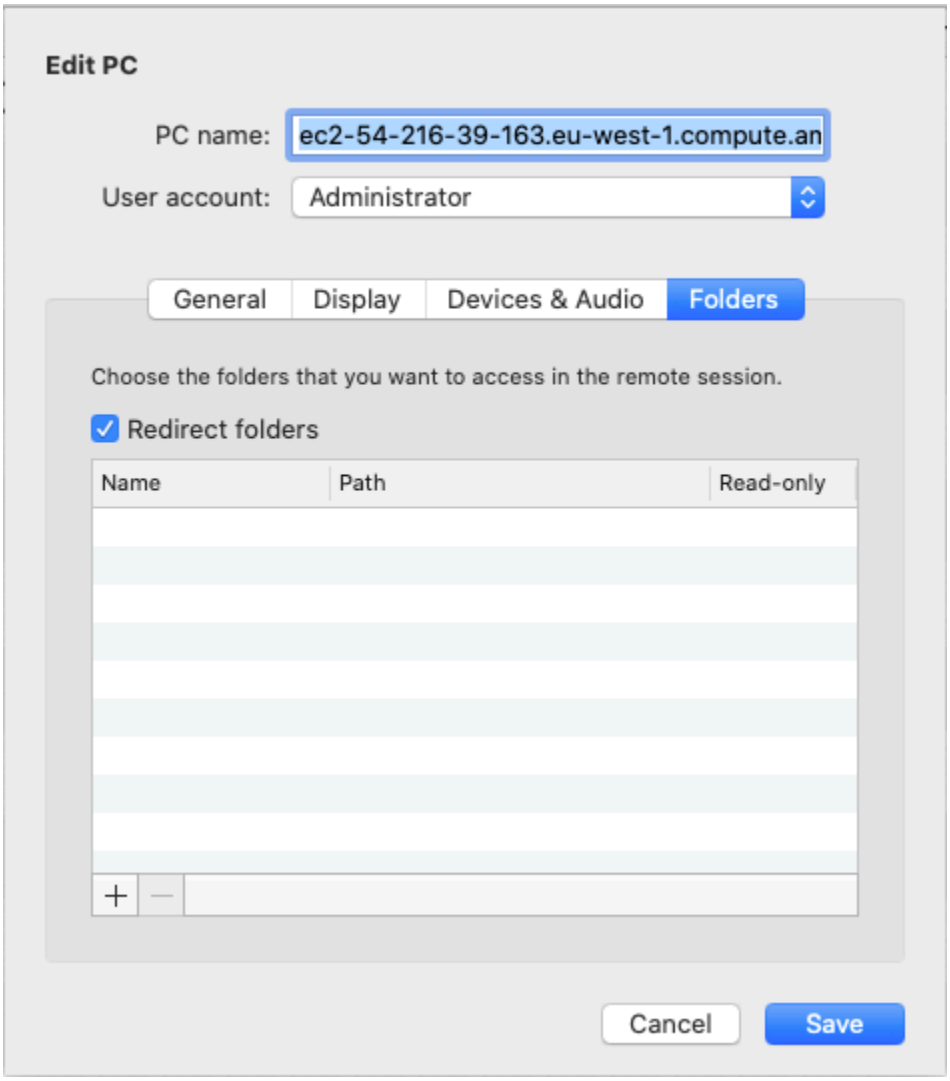


8. 연결을 선택하여 Windows 인스턴스에 연결합니다.

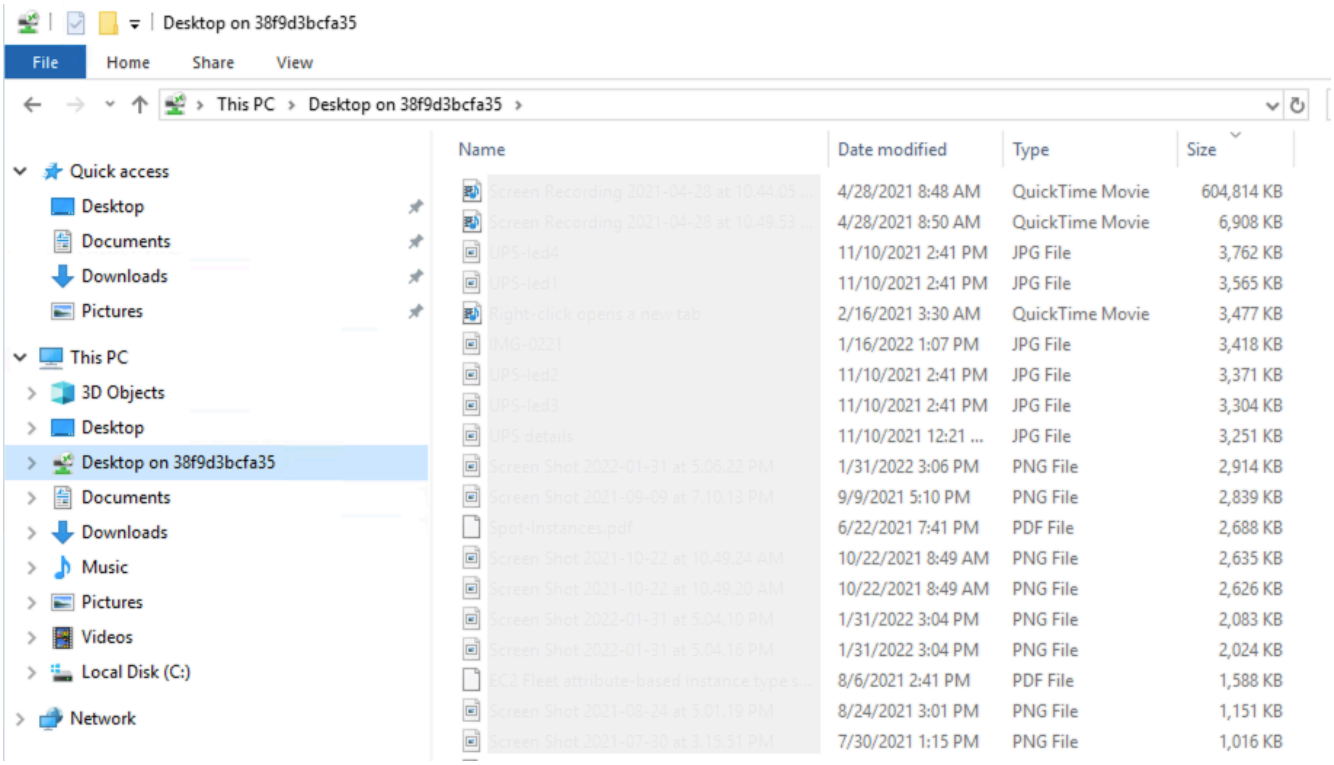
macOS X

로컬 macOS X 컴퓨터의 로컬 폴더에 원격 세션 드라이브 매핑

1. 원격 데스크톱 연결 클라이언트를 엽니다.
2. 인스턴스에 처음 연결할 때 Amazon EC2 콘솔에서 다운로드한 RDP 파일을 찾아 원격 데스크톱 연결 클라이언트로 끌어 놓습니다.
3. RDP 파일을 마우스 오른쪽 버튼으로 클릭하고 편집(Edit)을 선택합니다.
4. 폴더 탭을 선택하고 폴더 리디렉션 확인란을 선택합니다.



5. 왼쪽 하단에서 + 아이콘을 선택하고 매핑할 폴더를 찾은 다음 열기(Open)를 선택합니다. 매핑할 모든 폴더에 대해 이 단계를 반복합니다.
6. Save(저장)를 선택합니다.
7. 연결을 선택하여 Windows 인스턴스에 연결합니다. 암호를 입력하라는 메시지가 나타납니다.
8. 인스턴스의 파일 탐색기에서내 PC(This PC)를 클릭하고 로컬 파일에 액세스할 수 있는 공유 폴더를 찾습니다. 다음 스크린샷에서 로컬 컴퓨터의 바탕 화면(Desktop) 폴더는 인스턴스의 원격 세션 드라이브에 매핑되었습니다.



Mac 컴퓨터의 원격 세션에서 로컬 디바이스를 사용할 수 있도록 하는 방법에 대한 자세한 내용은 macOS 클라이언트 시작을 참조하세요.

Session Manager를 사용하여 Amazon EC2 인스턴스에 연결

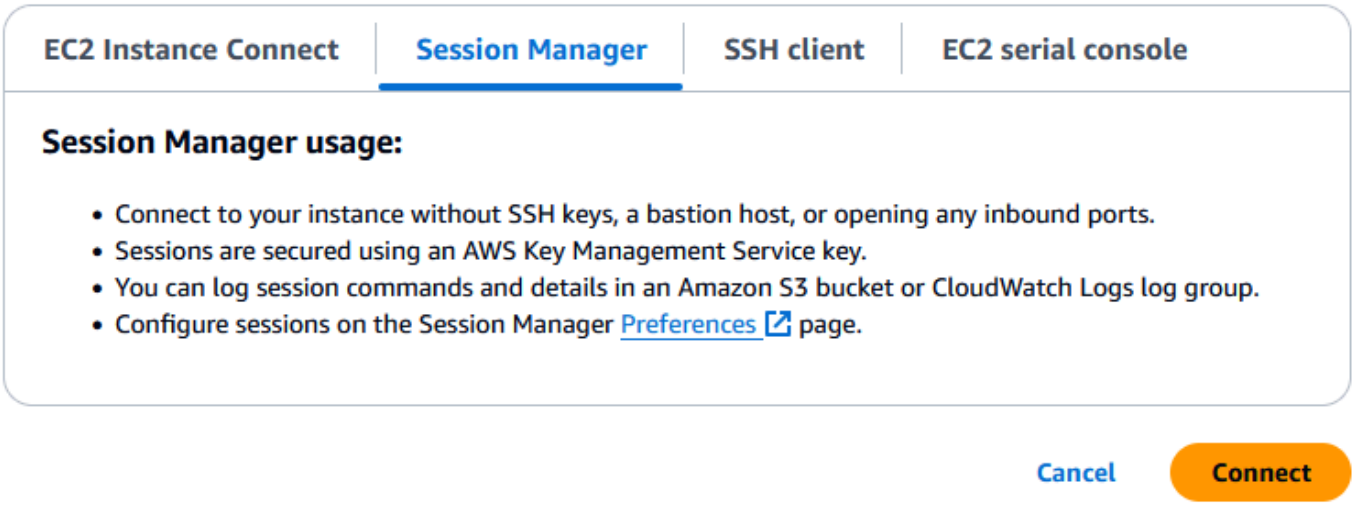
세션 관리자는 대화형 원클릭 브라우저 기반 셸 또는 AWS Systems Manager를 통해 Amazon EC2 인스턴스를 관리할 수 있는 완전관리형 AWS CLI 기능입니다. 세션 관리자를 사용하여 계정의 인스턴스와 관련된 세션을 시작할 수 있습니다. 세션이 시작된 후에는 다른 연결 유형에서와 마찬가지로 인스턴스에서 대화형 명령을 실행할 수 있습니다. 세션 관리자에 대한 자세한 내용은 AWS Systems Manager 사용자 설명서의 AWS Systems Manager 세션 관리자를 참조하세요.

사전 조건

Session Manager를 사용하여 인스턴스에 연결하기 전에 필요한 설정 단계를 완료해야 합니다. 예를 들어 인스턴스는 SSM에서 관리하고 AmazonSSMManagedInstanceCore 정책과 IAM 역할을 연결해야 합니다. 자세한 내용은 Session Manager 설정을 참조하세요.

Amazon EC2 콘솔의 Session Manager를 사용하여 Amazon EC2 인스턴스에 연결하려면 다음을 수행합니다.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances(인스턴스)를 선택합니다.
3. 인스턴스를 선택한 다음 연결을 선택합니다.
4. 연결 방법으로 Session Manager를 선택합니다.
5. 연결을 클릭하여 세션을 선택합니다.



문제 해결

하나 이상의 Systems Manager 작업(`ssm:command-name`)을 수행할 권한이 없다는 오류가 표시되면 Amazon EC2 콘솔에서 세션을 시작할 수 있도록 정책을 업데이트해야 합니다. 자세한 내용과 지침은 AWS Systems Manager 사용 설명서의 Session Manager에 대한 빠른 시작 기본 IAM 정책을 참조하세요.

EC2 Instance Connect를 사용하여 Linux 인스턴스에 연결

Amazon EC2 Instance Connect는 Secure Shell(SSH)을 사용하여 Linux 인스턴스에 연결할 수 있는 안전한 방법을 제공합니다. 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.

EC2 Instance Connect를 사용하면 Amazon EC2 콘솔 또는 직접 선택한 SSH 클라이언트로 인스턴스에 연결할 수 있습니다.

EC2 Instance Connect를 사용하여 인스턴스에 연결할 때 EC2 Instance Connect API는 SSH 퍼블릭 키를 인스턴스 메타데이터에 푸시하여 60초 동안 유지합니다. 사용자에게 연결된 IAM 정책은 퍼블릭 키를 인스턴스 메타데이터로 푸시하도록 사용자에게 권한을 부여합니다. SSH 대몬은 EC2 Instance Connect가 설치될 때 구성된 `AuthorizedKeysCommand` 및 `AuthorizedKeysCommandUser`를 사용하여 인증을 위해 인스턴스 메타데이터에서 퍼블릭 키를 찾고 사용자를 인스턴스에 연결합니다.

> **Tip**
>
> EC2 Instance Connect는 Linux 인스턴스에 연결할 수 있는 옵션 중 하나입니다. 다른 옵션은 SSH를 사용하여 Linux 인스턴스에 연결을(를) 참조하세요. Windows 인스턴스에 연결하려면 RDP를 사용하여 Windows 인스턴스에 연결 섹션을 참조하세요.

요금

EC2 Instance Connect는 추가 비용 없이 제공됩니다.

리전 가용성

EC2 인스턴스 연결은 아시아 태평양(태국)와 멕시코(중부) 리전을 제외한 모든 AWS 리전에서 사용할 수 있습니다. 로컬 영역에서는 지원되지 않습니다.

내용

- 자습서: EC2 Instance Connect를 사용하여 인스턴스에 연결하는 데 필요한 구성 완료
- EC2 Instance Connect 사전 조건
- EC2 Instance Connect에 대한 IAM 권한 부여
- EC2 인스턴스에 EC2 Instance Connect를 설치합니다.
- EC2 Instance Connect를 사용하여 Linux 인스턴스에 연결
- EC2 Instance Connect 제거

EC2 Instance Connect를 사용하여 Bastion Host의 보안을 개선하는 방법은 블로그 게시물 Securing your bastion hosts with Amazon EC2 Instance Connect(Amazon EC2 Instance Connect를 사용하여 Bastion Host의 보안 유지)를 참조하세요.

자습서: EC2 Instance Connect를 사용하여 인스턴스에 연결하는 데 필요한 구성 완료

Amazon EC2 콘솔에서 EC2 Instance Connect를 사용하여 인스턴스에 연결하려면 먼저 인스턴스에 성공적으로 연결할 수 있는 사전 필수 구성 작업을 완료해야 합니다. 이 자습서의 목적은 사전 필수 구성을 완료하는 작업을 안내하는 것입니다.

자습서 개요

이 자습서에서는 다음 작업을 완료합니다.

- 작업 1: EC2 Instance Connect 사용에 필요한 권한 부여

  먼저 퍼블릭 키를 인스턴스 메타데이터에 푸시할 수 있는 IAM 권한이 포함된 IAM 정책을 생성합니다. IAM 자격 증명에서 이 권한을 얻을 수 있도록 IAM 자격 증명(사용자, 사용자 그룹 또는 역할)에 이 정책을 연결합니다.
- 작업 2: EC2 Instance Connect 서비스에서 인스턴스로의 인바운드 트래픽 허용

  그런 다음, EC2 Instance Connect 서비스에서 인스턴스로 트래픽을 허용하는 보안 그룹을 생성합니다. 이는 Amazon EC2 콘솔에서 EC2 Instance Connect를 사용하여 인스턴스에 연결할 때 필요합니다.
- 작업 3: 인스턴스 시작

  그런 다음, EC2 Instance Connect에 사전 설치된 AMI를 사용하여 EC2 인스턴스를 시작하고 이전 단계에서 생성한 보안 그룹을 추가합니다.
- 작업 4: 인스턴스에 연결

  마지막으로 Amazon EC2 콘솔에서 EC2 Instance Connect를 사용하여 인스턴스에 연결합니다. 연결할 수 있다면 작업 1, 2, 3에서 완료한 사전 조건 구성이 성공한 것입니다.

작업 1: EC2 Instance Connect 사용에 필요한 권한 부여

EC2 Instance Connect를 사용하여 인스턴스에 연결할 때 EC2 Instance Connect API는 SSH 퍼블릭 키를 인스턴스 메타데이터에 푸시하여 60초 동안 유지합니다. 퍼블릭 키를 인스턴스 메타데이터로 푸시하는 데 필요한 권한을 부여하려면 IAM 자격 증명(사용자, 사용자 그룹 또는 역할)에 연결된 IAM 정책이 필요합니다.

작업 목표

퍼블릭 키를 인스턴스에 푸시하도록 권한을 부여하는 IAM 정책을 생성합니다. 허용할 특정 작업은 `ec2-instance-connect:SendSSHPublicKey`입니다. 또한 Amazon EC2 콘솔에서 인스턴스를 보고 선택할 수 있도록 `ec2:DescribeInstances` 작업을 허용해야 합니다.

정책을 생성한 후에 IAM 자격 증명에서 이 권한을 얻을 수 있도록 IAM 자격 증명(사용자, 사용자 그룹 또는 역할)에 정책을 연결합니다.

다음과 같이 구성된 정책을 생성합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": "ec2-instance-connect:SendSSHPublicKey",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeInstances",
            "Resource": "*"
        }
    ]
}
```

Important

이 자습서에서 생성한 IAM 정책은 매우 허용적인 정책으로, 모든 AMI 사용자 이름을 사용하여 모든 인스턴스에 연결할 수 있습니다. 자습서를 단순하게 유지하고 이 자습서에서 학습할 특정 구성에 초점을 맞추고자 매우 허용적인 이 정책을 사용합니다. 하지만 프로덕션 환경에서는 최소 권한을 제공하도록 IAM 정책을 구성하는 것이 좋습니다. 예제 IAM 정책은 EC2 Instance Connect에 대한 IAM 권한 부여 섹션을 참조하세요.

EC2 Instance Connect를 사용하여 인스턴스에 연결할 수 있게 하는 IAM 정책을 생성 및 연결하는 방법

1. 먼저 IAM 정책 생성

   a. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.

   b. 탐색 창에서 Policies를 선택합니다.

c. 정책 생성을 선택합니다.

d. 권한 지정 페이지에서 다음을 수행합니다.

i. 서비스에서 EC2 Instance Connect를 선택합니다.

ii. 허용되는 작업의 검색 필드에 **send**를 입력하기 시작하면 관련 작업이 표시됩니다. SendSSHPublicKey를 선택합니다.

iii. 리소스 아래에서 모두를 선택합니다. 프로덕션 환경에서는 ARN으로 인스턴스를 지정하는 것이 좋습니다. 단, 이 자습서에서는 모든 인스턴스를 허용합니다.

iv. Add More Permissions(더 많은 권한 추가)를 선택합니다.

v. 서비스에서 EC2를 선택합니다.

vi. 허용되는 작업의 검색 필드에 **describein**을 입력하기 시작하면 관련 작업이 표시됩니다. DescribeInstances를 선택합니다.

vii. 다음을 선택합니다.

e. 검토 및 생성 페이지에서 다음을 수행합니다.

i. [정책 이름(Policy name)]에 정책의 이름을 입력합니다.

ii. 정책 생성을 선택합니다.

2. 그리고 자격 증명에 정책 연결

a. IAM 콘솔의 탐색 창에서 정책(Policies)을 선택합니다.

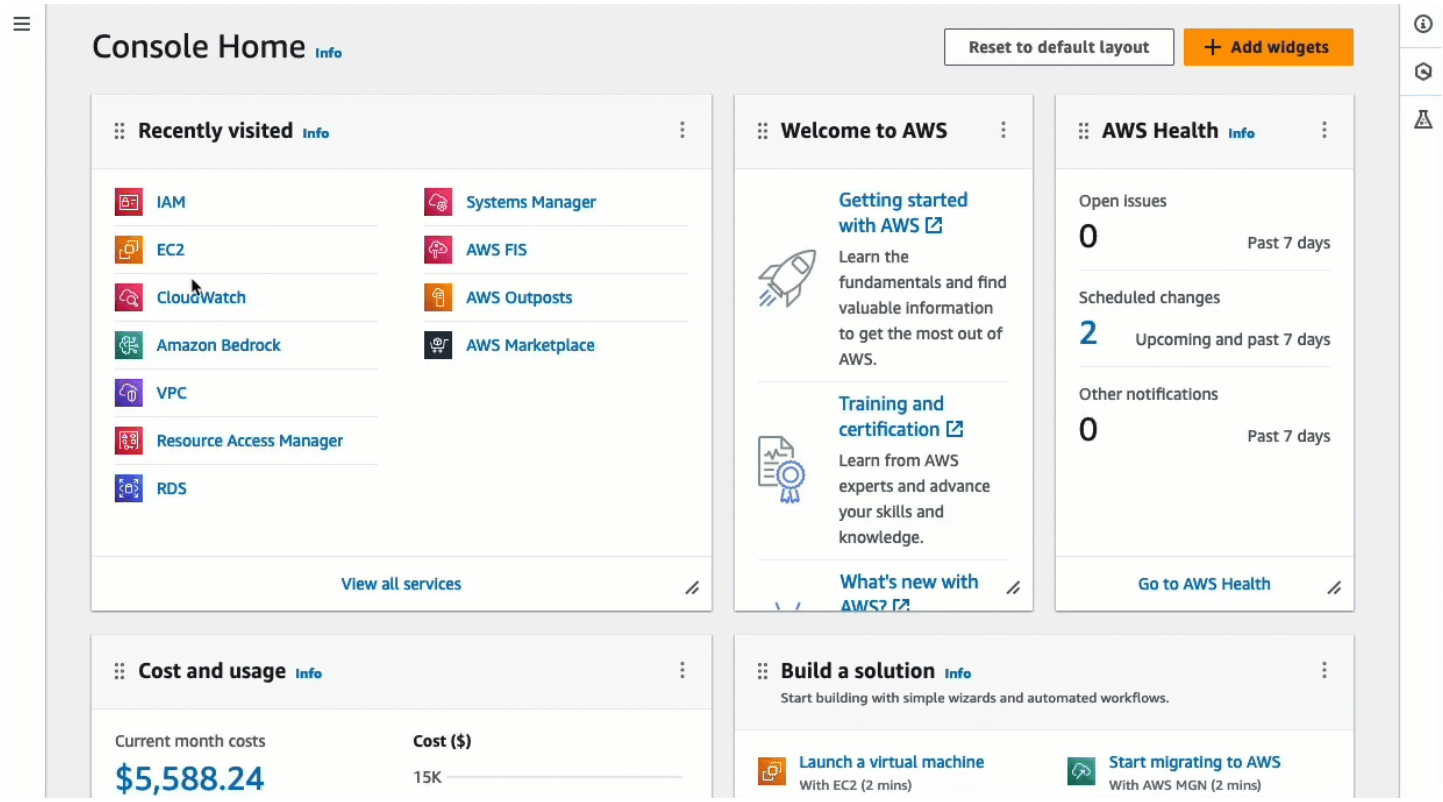
b. 정책 목록에서 연결할 정책 이름 옆의 옵션 버튼을 선택합니다. 검색 상자를 사용하여 정책 목록을 필터링할 수 있습니다.

c. 작업], 연결을 선택합니다.

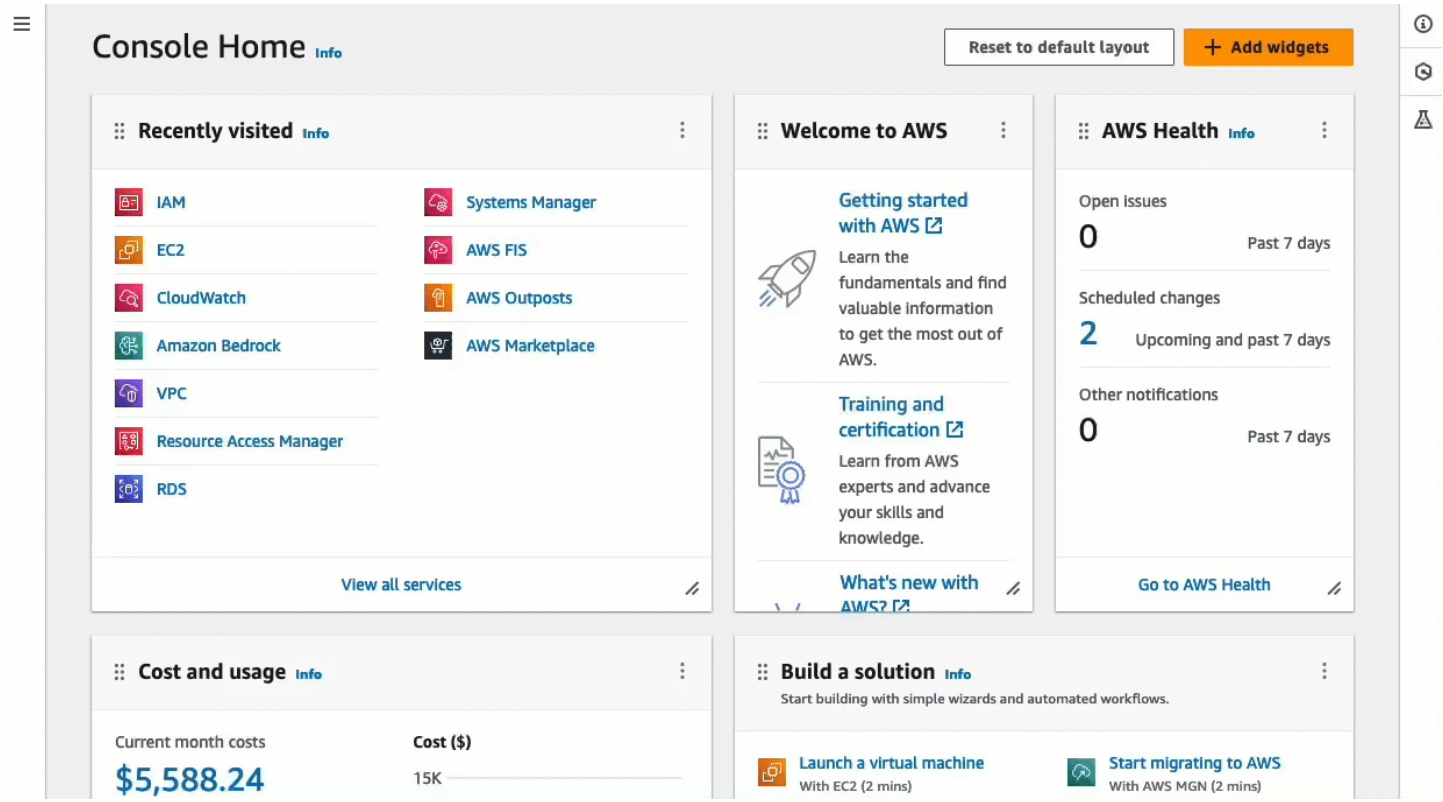
d. IAM 개체에서 ID(사용자, 사용자 그룹 또는 역할) 옆의 확인란을 선택합니다. 검색 상자를 사용하여 개체 목록을 필터링할 수 있습니다.

e. 정책 연결을 선택합니다.

애니메이션 보기: IAM 정책 생성



애니메이션 보기: IAM 정책 연결



작업 2: EC2 Instance Connect 서비스에서 인스턴스로의 인바운드 트래픽 허용

Amazon EC2 콘솔에서 EC2 Instance Connect를 사용하여 인스턴스에 연결할 경우, 인스턴스에 도달하기 위해 허용해야 하는 트래픽은 EC2 Instance Connect 서비스에서 생성된 트래픽입니다. 이는 로컬 컴퓨터에서 인스턴스로 연결하는 작업과는 다릅니다. 이 경우 로컬 컴퓨터에서 인스턴스로의 트래픽을 허용해야 합니다. EC2 Instance Connect 서비스의 트래픽을 허용하려면 EC2 Instance Connect 서비스의 IP 주소 범위에서 생성되는 인바운드 SSH 트래픽을 허용하는 보안 그룹을 생성해야 합니다.

AWS는 접두사 목록을 사용하여 IP 주소 범위를 관리합니다. EC2 Instance Connect 접두사 목록의 이름은 다음과 같으며, *##*은 리전 코드로 대체됩니다.

- IPv4 접두사 목록 이름: `com.amazonaws.region.ec2-instance-connect`
- IPv6 접두사 목록 이름: `com.amazonaws.region.ipv6.ec2-instance-connect`

작업 목표

인스턴스가 있는 리전에서 IPv4 접두사 목록의 포트 22 상의 인바운드 SSH 트래픽을 허용하는 보안 그룹을 생성합니다.

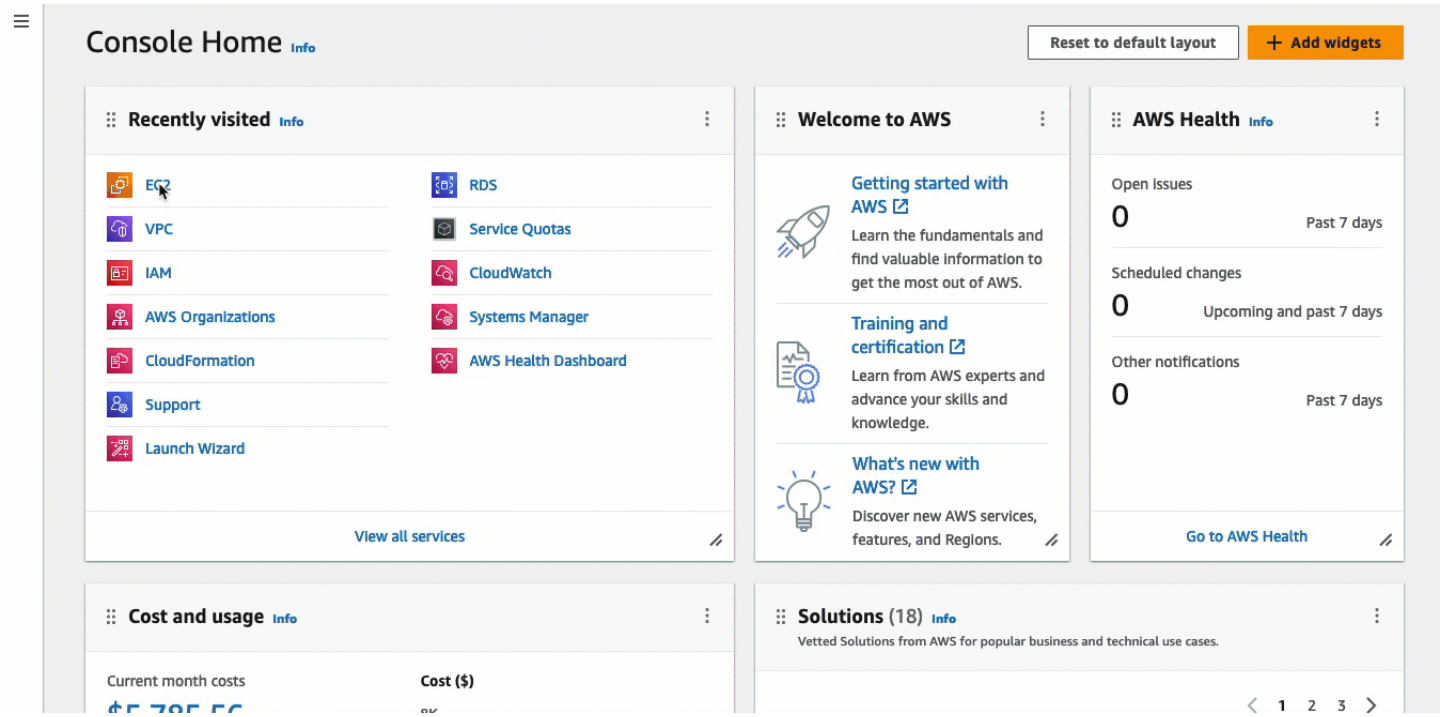
EC2 Instance Connect 서비스에서 인스턴스로 인바운드 트래픽을 허용하는 보안 그룹을 생성하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다.
3. 보안 그룹 생성을 선택합니다.
4. 기본 세부 정보에서 다음을 수행합니다.
   a. 보안 그룹 이름에 보안 그룹의 의미 있는 이름을 입력합니다.
   b. 설명에 보안 그룹의 의미 있는 설명을 입력합니다.
5. 인바운드 규칙에서 다음을 수행합니다.
   a. 규칙 추가를 선택합니다.
   b. Type(유형)에서 SSH를 선택합니다.
   c. 소스에서 사용자 지정을 그대로 둡니다.
   d. 소스 옆의 필드에서 EC2 Instance Connect의 접두사 목록을 선택합니다.

예를 들어, 인스턴스가 미국 동부(버지니아 북부)(us-east-1) 리전에 있고 사용자가 퍼블릭 IPv4 주소에 연결할 경우, com.amazonaws.us-east-1.ec2-instance-connect 접두사 목록을 선택합니다.

6. 보안 그룹 생성을 선택합니다.

애니메이션 보기: 보안 그룹 생성



작업 3: 인스턴스 시작

인스턴스를 시작할 때 인스턴스를 시작하는 데 필요한 정보가 포함된 AMI를 지정해야 합니다. EC2 Instance Connect가 사전 설치되어 있거나 설치되지 않은 상태에서 인스턴스를 시작하도록 선택할 수 있습니다. 이 작업에서는 EC2 Instance Connect가 사전 설치된 AMI를 지정합니다.

EC2 Instance Connect가 사전 설치되지 않은 상태에서 인스턴스를 시작하고 EC2 Instance Connect를 사용하여 인스턴스에 연결하려는 경우 추가 구성 단계를 수행해야 합니다. 이 단계는 본 자습서에서 다루지 않습니다.

작업 목표

EC2 Instance Connect가 사전 설치된 Amazon Linux 2023 AMI에서 인스턴스를 시작합니다. 또한 Amazon EC2 콘솔에서 EC2 Instance Connect를 사용하여 인스턴스에 연결할 수 있도록 이전에 생성

한 보안 그룹을 지정해야 합니다. EC2 Instance Connect를 사용하여 인스턴스에 연결하면 퍼블릭 키를 인스턴스의 메타데이터로 푸시하므로 인스턴스를 시작할 때 SSH 키를 지정하지 않아도 됩니다.

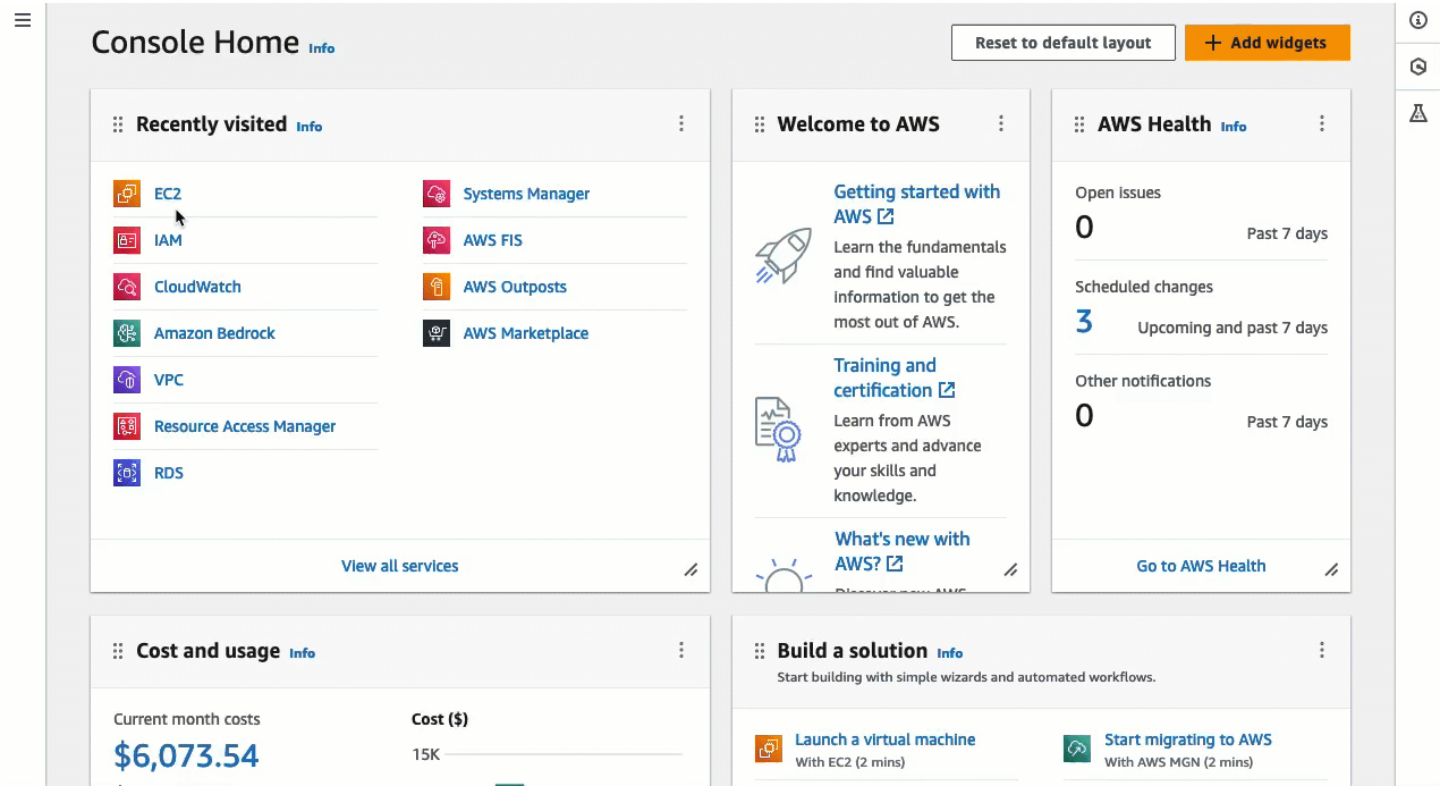
Amazon EC2 콘솔에서 EC2 Instance Connect를 사용하여 연결할 인스턴스를 시작하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 화면 상단의 탐색 모음에는 현재 AWS 리전이 표시됩니다(예: 아일랜드). 인스턴스를 시작할 리전을 선택합니다. 특정 리전의 트래픽을 허용하는 보안 그룹을 생성했기 때문에 이 선택이 중요합니다. 인스턴스를 시작할 리전과 동일한 리전을 선택해야 하기 때문입니다.
3. Amazon EC2 콘솔 대시보드에서 인스턴스 시작을 선택합니다.
4. (선택 사항) 이름 및 태그(Name and tags) 아래의 이름(Name)에 인스턴스를 설명하는 이름을 입력합니다.
5. 애플리케이션 및 OS 이미지(Amazon Machine Image)에서 빠른 시작을 선택합니다. Amazon Linux가 기본적으로 선택됩니다. Amazon Machine Image(AMI) 아래에서 Amazon Linux 2023 AMI가 기본적으로 선택됩니다. 이 작업에 대한 기본 선택을 그대로 유지합니다.
6. 인스턴스 유형 아래 인스턴스 유형에서 기본 선택을 유지하거나 다른 인스턴스 유형을 선택합니다.
7. 키 페어(로그인) 아래 키 페어 이름에서 없이 진행(권장되지 않음)을 선택합니다. EC2 Instance Connect를 사용하여 인스턴스에 연결하는 경우 EC2 Instance Connect는 키 페어를 인스턴스의 메타데이터에 푸시하며, 이 키 페어가 연결에 사용됩니다.
8. Network settings(네트워크 설정)에서 다음을 수행합니다.
   a. 퍼블릭 IP 자동 할당에서 활성화를 그대로 둡니다.

      > **Note**
      >
      > Amazon EC2 콘솔에서 EC2 Instance Connect를 사용하여 인스턴스에 연결하려면 인스턴스에 퍼블릭 IPv4 또는 퍼블릭 IPv6 주소가 있어야 합니다.

   b. 방화벽(보안 그룹)에서 기존 보안 그룹 선택을 선택합니다.
   c. 일반 보안 그룹에서 이전에 생성한 보안 그룹을 선택합니다.
9. 요약(Summary) 패널에서 인스턴스 실행(Launch instance)을 선택합니다.

애니메이션 보기: 인스턴스 시작



작업 4: 인스턴스에 연결

EC2 Instance Connect를 사용하여 인스턴스에 연결할 때 EC2 Instance Connect API는 SSH 퍼블릭 키를 인스턴스 메타데이터에 푸시하여 60초 동안 유지합니다. SSH 대몬은 `AuthorizedKeysCommand` 및 `AuthorizedKeysCommandUser`를 사용하여 인증을 위해 인스턴스 메타데이터에서 퍼블릭 키를 찾고 사용자를 인스턴스에 연결합니다.

작업 목표

이 작업에서는 Amazon EC2 콘솔에서 EC2 Instance Connect를 사용하여 인스턴스에 연결합니다. 필수 작업 1, 2, 3을 완료하면 연결에 성공합니다.

인스턴스에 연결하는 단계

다음 단계를 사용하여 인스턴스에 연결합니다. 이 단계의 애니메이션을 보려면 애니메이션 보기: 인스턴스에 연결 섹션을 참조하세요.

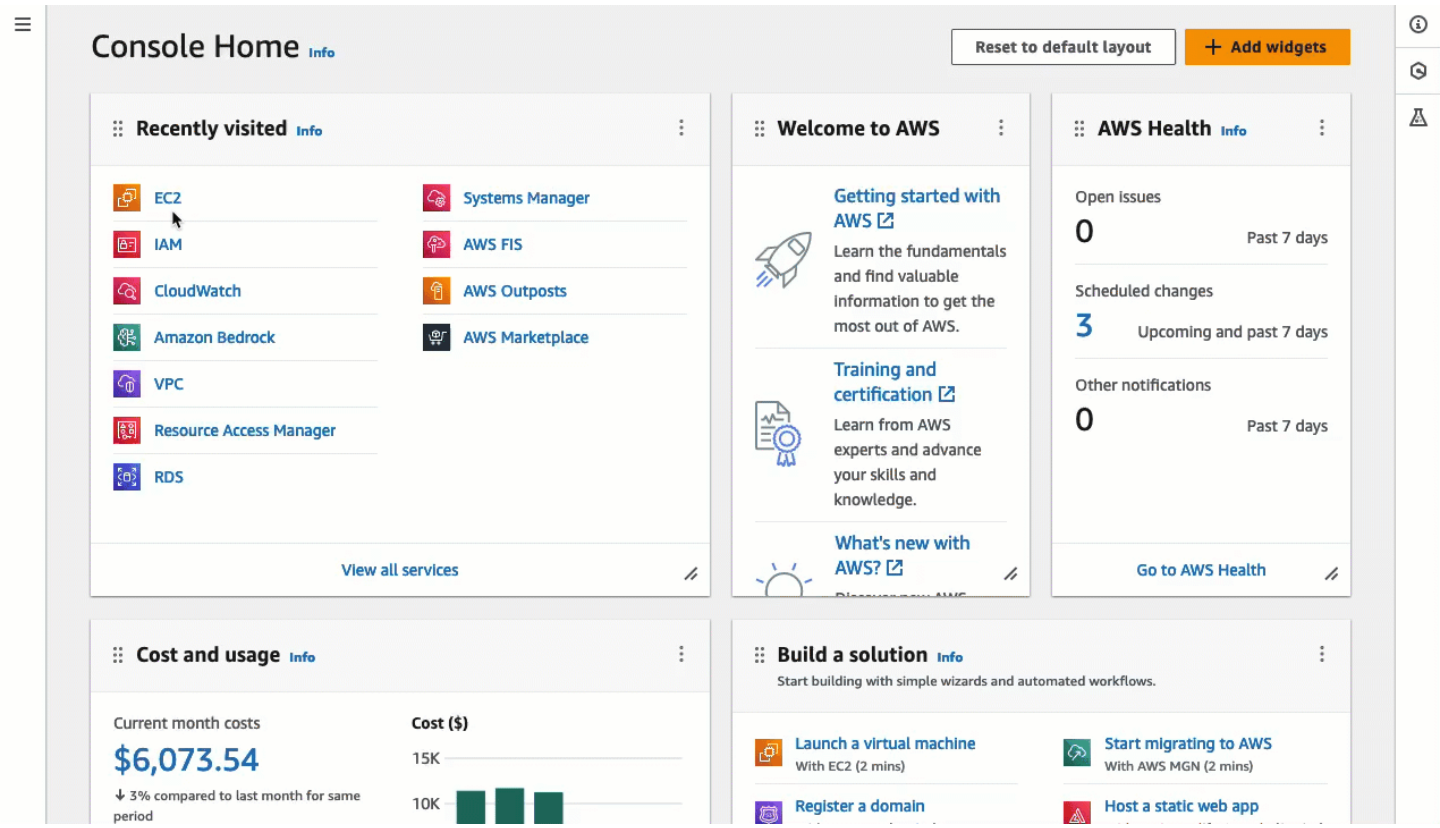
Amazon EC2 콘솔에서 EC2 Instance Connect를 사용하여 인스턴스를 연결하는 방법

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

2. 화면 상단의 탐색 모음에는 현재 AWS 리전이 표시됩니다(예: 아일랜드). 인스턴스가 있는 리전을 선택합니다.
3. 탐색 창에서 인스턴스를 선택합니다.
4. 인스턴스를 선택하고 연결을 선택합니다.
5. EC2 Instance Connect 탭을 선택합니다.
6. 연결 유형에서 EC2 Instance Connect를 사용하여 연결을 선택합니다.
7. 연결을 선택합니다.

   터미널 창이 브라우저에서 열리고 인스턴스에 연결됩니다.

애니메이션 보기: 인스턴스에 연결



EC2 Instance Connect 사전 조건

다음은 EC2 Instance Connect 설치 및 사용을 위한 사전 조건입니다.

- EC2 Instance Connect 설치
- 네트워크 연결 확인
- 인바운드 SSH 트래픽 허용

- 권한 부여
- 로컬 컴퓨터에 SSH 클라이언트 설치
- 사용자 이름 요구 사항 충족

EC2 Instance Connect 설치

EC2 Instance Connect를 사용하여 인스턴스에 연결하려면 인스턴스에 EC2 Instance Connect가 설치되어 있어야 합니다. EC2 Instance Connect가 사전 설치된 AMI를 사용하여 인스턴스를 시작하거나 지원되는 AMI로 시작된 인스턴스에 EC2 Instance Connect를 설치할 수 있습니다. 자세한 내용은 EC2 인스턴스에 EC2 Instance Connect를 설치합니다. 섹션을 참조하세요.

네트워크 연결 확인

사용자가 인터넷이나 인스턴스의 프라이빗 IP 주소를 통해 인스턴스에 연결할 수 있도록 인스턴스를 구성할 수 있습니다. 사용자가 EC2 Instance Connect를 사용하여 인스턴스에 연결하는 방법에 따라 다음 네트워크 액세스를 구성해야 합니다.

- 사용자가 인터넷을 통해 인스턴스에 연결하는 경우 인스턴스는 퍼블릭 IPv4 또는 퍼블릭 IPv6 주소를 보유해야 하고 인터넷 경로가 있는 퍼블릭 서브넷에 있어야 합니다. 기본 퍼블릭 서브넷을 수정하지 않은 경우 IPv4의 인터넷 연결 경로만 포함되고 IPv6는 포함되지 않습니다. 자세한 내용은 Amazon VPC 사용 설명서의 인터넷 게이트웨이를 사용하여 VPC 인터넷 액세스 활성화를 참조하세요.
- 사용자가 인스턴스의 프라이빗 IPv4 주소를 통해 인스턴스에 연결하는 경우 사용자가 인스턴스의 프라이빗 IP 주소에 도달할 수 있도록 AWS Direct Connect, AWS Site-to-Site VPN 또는 VPC 피어링을 사용하여 VPC에 대한 프라이빗 네트워크 연결을 설정해야 합니다.

인스턴스에 퍼블릭 IPv4 또는 퍼블릭 IPv6 주소가 없고 위에서 설명한 대로 네트워크 액세스를 구성하지 않으려는 경우 EC2 Instance Connect 대신 EC2 Instance Connect 엔드포인트를 고려할 수 있습니다. EC2 Instance Connect 엔드포인트를 사용하면 인스턴스에 퍼블릭 IPv4 또는 퍼블릭 IPv6 주소가 없어도 SSH 또는 RDP를 통해 인스턴스에 연결할 수 있습니다. 자세한 내용은 Amazon EC2 콘솔을 사용하여 Linux 인스턴스에 연결 섹션을 참조하세요.

인바운드 SSH 트래픽 허용

Amazon EC2 콘솔을 사용하여 인스턴스에 연결할 때

사용자가 Amazon EC2 콘솔을 사용하여 인스턴스에 연결하는 경우, 인스턴스에 도달하기 위해 허용해야 하는 트래픽은 EC2 Instance Connect 서비스에서 생성된 트래픽입니다. 서비스는 특정 IP 주소 범

위로 식별되며, 이는 AWS에서 접두사 목록을 통해 관리됩니다. EC2 Instance Connect 서비스에서 인바운드 SSH 트래픽을 허용하는 보안 그룹을 생성해야 합니다. 이를 구성하려면 인바운드 규칙의 경우 소스 옆의 필드에서 EC2 Instance Connect 접두사 목록을 선택합니다.

AWS에서는 각 리전에 대한 IPv4 및 IPv6 주소 서로 다른 관리 접두사 목록을 제공합니다. EC2 Instance Connect 접두사 목록의 이름은 다음과 같으며, *##*은 리전 코드로 대체됩니다.

- IPv4 접두사 목록 이름: `com.amazonaws.region.ec2-instance-connect`
- IPv6 접두사 목록 이름: `com.amazonaws.region.ipv6.ec2-instance-connect`

보안 그룹 생성에 대한 자세한 내용은 작업 2: EC2 Instance Connect 서비스에서 인스턴스로의 인바운드 트래픽 허용 섹션을 참조하세요. 자세한 내용은 Amazon VPC 사용 설명서의 사용 가능한 AWS 관리형 접두사 목록을 참조하세요.

CLI 또는 SSH를 사용하여 인스턴스에 연결할 때

인스턴스와 연관된 보안 그룹이 IP 주소 또는 네트워크에서 포트 22를 통한 인바운드 SSH 트래픽을 허용하는지 확인하세요. VPC의 기본 보안 그룹은 기본적으로 수신 SSH 트래픽을 허용하지 않습니다. 인스턴스 시작 마법사에서 생성한 보안 그룹은 기본적으로 수신 SSH 트래픽을 허용합니다. 자세한 내용은 컴퓨터에서 인스턴스 연결에 대한 규칙 섹션을 참조하세요.

권한 부여

EC2 Instance Connect를 사용하여 인스턴스에 연결할 모든 IAM 사용자에게 필요한 권한을 부여해야 합니다. 자세한 내용은 EC2 Instance Connect에 대한 IAM 권한 부여 섹션을 참조하세요.

로컬 컴퓨터에 SSH 클라이언트 설치

사용자가 SSH를 사용하여 연결하는 경우 로컬 컴퓨터에 SSH 클라이언트가 있어야 합니다.

사용자의 로컬 컴퓨터에는 대개 기본적으로 SSH 클라이언트가 설치되어 있습니다. 사용자는 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 참조하세요. Windows 10에 SSH 클라이언트를 설치하는 방법은 OpenSSH in Windows를 참조하세요.

사용자가 Amazon EC2 콘솔만을 사용하여 인스턴스에 연결하는 경우 로컬 컴퓨터에 SSH 클라이언트를 설치할 필요가 없습니다.

사용자 이름 요구 사항 충족

EC2 Instance Connect를 사용하여 인스턴스에 연결하는 경우 사용자 이름은 다음 요구 사항을 충족해야 합니다.

- 첫 번째 문자: 문자(A-Z, a-z), 숫자(0-9) 또는 밑줄(_)이어야 함
- 다음 문자: 문자(A-Z, a-z), 숫자(0-9) 또는 다음 문자를 사용할 수 있습니다. @ . _ -
- 최소 길이: 1자
- 최대 길이: 31자

EC2 Instance Connect에 대한 IAM 권한 부여

EC2 Instance Connect를 사용하여 인스턴스에 연결하려면 다음 작업 및 조건에 대한 사용자 권한을 부여하는 IAM 정책을 만들어야 합니다.

- `ec2-instance-connect:SendSSHPublicKey` 작업 - 퍼블릭 키를 인스턴스에 푸시할 수 있는 권한을 부여합니다.
- `ec2:osuser` 조건 - 퍼블릭 키를 인스턴스로 푸시할 수 있는 OS 사용자의 이름을 지정합니다. 인스턴스를 시작하는 데 사용한 AMI의 기본 사용자 이름을 사용합니다. AL2023 및 Amazon Linux 2의 기본 사용자 이름은 `ec2-user`이며 Ubuntu의 경우 `ubuntu`입니다.
- `ec2:DescribeInstances` 작업 - 래퍼가 이 작업을 호출하기 때문에 EC2 콘솔을 사용할 때 필요합니다. 사용자는 이미 다른 정책에서 이 작업을 호출할 권한을 보유할 수도 있습니다.
- `ec2:DescribeVpcs` 조치 — IPv6 주소에 연결할 때 필요합니다.

특정 EC2 인스턴스에 대한 액세스 제한을 고려하세요. 그렇지 않은 경우, `ec2-instance-connect:SendSSHPublicKey` 작업에 대한 권한이 있는 모든 IAM 보안 주체가 모든 EC2 인스턴스에 연결할 수 있습니다. 리소스 ARN을 지정하거나 리소스 태그를 조건 키로 사용하여 액세스를 제한할 수 있습니다.

자세한 내용은 Amazon EC2 Instance Connect에 사용되는 작업, 리소스 및 조건 키를 참조하세요.

IAM 정책 생성에 대한 자세한 내용은 IAM 사용 설명서의 IAM 정책 생성을 참조하세요.

사용자가 특정 인스턴스에 연결하도록 허용

다음 IAM 정책은 리소스 ARN으로 식별되는 특정 인스턴스에 연결할 권한을 부여합니다.

다음 예제 IAM 정책에서 다음 작업 및 조건이 지정되어 있습니다.

- `ec2-instance-connect:SendSSHPublicKey` 작업은 사용자에게 리소스 ARN으로 확인된 두 개의 인스턴스에 연결할 수 있는 권한을 부여합니다. 사용자에게 모든 EC2 인스턴스에 연결할 수 있는 권한을 부여하려면 리소스 ARN을 * 와일드카드로 바꿉니다.
- `ec2:osuser` 조건은 연결할 때 *ami-username*이 지정된 경우에만 인스턴스에 연결할 수 있는 권한을 부여합니다.
- `ec2:DescribeInstances` 작업은 콘솔을 사용하여 인스턴스에 연결할 사용자에게 권한을 부여하도록 지정됩니다. 사용자가 SSH 클라이언트만을 사용하여 인스턴스에 연결하는 경우 `ec2:DescribeInstances`를 생략할 수 있습니다. `ec2:Describe*` API 작업은 리소스 수준 권한을 지원하지 않습니다. 따라서 `Resource` 요소에 * 와일드카드가 필요합니다.
- `ec2:DescribeVpcs` 작업은 콘솔을 사용하여 IPv6 주소를 사용하여 인스턴스에 연결할 사용자에게 권한을 부여하도록 지정됩니다. 사용자가 공용 IPv4 주소만 사용할 경우 `ec2:DescribeVpcs`를 생략할 수 있습니다. `ec2:Describe*` API 작업은 리소스 수준 권한을 지원하지 않습니다. 따라서 `Resource` 요소에 * 와일드카드가 필요합니다.

```
{
    "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",
                "ec2:DescribeVpcs"
            ],
            "Resource": "*"
        }
    ]
}
```

사용자가 특정 태그가 있는 인스턴스에 연결하도록 허용

ABAC(속성 기반 액세스 제어)는 사용자 및 AWS 리소스에 연결할 수 있는 태그를 기반으로 권한을 정의하는 권한 부여 전략입니다. 리소스 태그를 사용하여 인스턴스에 대한 액세스를 제어할 수 있습니다. 태그를 사용하여 AWS 리소스에 대한 액세스를 제어하는 방법에 대한 자세한 내용은 IAM 사용 설명서에서 AWS 리소스에 대한 액세스 제어를 참조하세요.

다음 예제 IAM 정책에서 `ec2-instance-connect:SendSSHPublicKey` 작업은 인스턴스에 키=`tag-key` 및 값=`tag-value`인 리소스 태그가 있는 조건에서 임의 인스턴스(리소스 ARN에 * 와일드카드로 표시됨)에 연결할 수 있는 권한을 사용자에게 부여합니다.

`ec2:DescribeInstances` 작업은 콘솔을 사용하여 인스턴스에 연결할 사용자에게 권한을 부여하도록 지정됩니다. 사용자가 SSH 클라이언트만을 사용하여 인스턴스에 연결하는 경우 `ec2:DescribeInstances`를 생략할 수 있습니다. `ec2:Describe*` API 작업은 리소스 수준 권한을 지원하지 않습니다. 따라서 `Resource` 요소에 * 와일드카드가 필요합니다.

`ec2:DescribeVpcs` 작업은 콘솔을 사용하여 IPv6 주소를 사용하여 인스턴스에 연결할 사용자에게 권한을 부여하도록 지정됩니다. 사용자가 공용 IPv4 주소만 사용할 경우 `ec2:DescribeVpcs`를 생략할 수 있습니다. `ec2:Describe*` API 작업은 리소스 수준 권한을 지원하지 않습니다. 따라서 `Resource` 요소에 * 와일드카드가 필요합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": "ec2-instance-connect:SendSSHPublicKey",
            "Resource": "arn:aws:ec2:region:account-id:instance/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/tag-key": "tag-value"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeVpcs"
            ],
            "Resource": "*"
        }
    ]
```

```
}
```

EC2 인스턴스에 EC2 Instance Connect를 설치합니다.

EC2 Instance Connect를 사용하여 Linux 인스턴스에 연결하려면 인스턴스에 EC2 Instance Connect가 설치되어 있어야 합니다. EC2 Instance Connect를 설치하면 인스턴스에 SSH 대몬(daemon)이 구성됩니다.

EC2 Instance Connect 패키지에 대한 자세한 내용은 GitHub 웹 사이트의 aws/aws-ec2-instance-connect-config를 참조하세요.

> **Note**
>
> SSH 인증을 위한 `AuthorizedKeysCommand` 및 `AuthorizedKeysCommandUser` 설정을 구성했다면 EC2 Instance Connect 설치 시 이들 항목이 업데이트가 되지 않습니다. 따라서 EC2 Instance Connect를 사용할 수 없습니다.

사전 조건 설치

EC2 Instance Connect를 설치하기 전에 다음 사전 조건을 충족해야 합니다.

- 인스턴스가 다음 중 하나를 사용하는지 확인:
  - 버전 2.0.20190618 이전의 Amazon Linux 2
  - AL2023 최소 AMI 또는 Amazon ECS 최적화 AMI
  - CentOS Stream 8 및 9
  - 14.2.1 이전의 macOS Sonoma, 13.6.3 이전의 Ventura 및 12.7.2 이전의 Monterey
  - Red Hat Enterprise Linux(RHEL) 8 및 9
  - Ubuntu 16.04 및 18.04

  > **Tip**
  >
  > 인스턴스가 Amazon Linux, macOS Sonoma, macOS Ventura, macOS Monterey 또는 Ubuntu의 최신 버전을 사용하여 시작된 경우 EC2 Instance Connect가 사전 설치되어 있으므로 직접 설치할 필요가 없습니다.

- EC2 Instance Connect에 대한 일반 사전 조건을 확인합니다.

자세한 내용은 EC2 Instance Connect 사전 조건 섹션을 참조하세요.

- 로컬 시스템에서 SSH 클라이언트를 사용하여 인스턴스에 연결하기 위한 사전 조건을 확인합니다.

  자세한 내용은 SSH를 사용하여 Linux 인스턴스에 연결 섹션을 참조하세요.

- 인스턴스의 ID 보기.

  Amazon EC2 콘솔을 사용하여(인스턴스 ID(Instance ID) 열에서) 인스턴스의 ID를 가져올 수 있습니다. 원하는 경우 describe-instances(AWS CLI) 또는 Get-EC2Instance(AWS Tools for Windows PowerShell) 명령을 사용할 수 있습니다.

EC2 Instance Connect 수동 설치

> **Note**
>
> 다음 AMI 중 하나를 사용하여 인스턴스를 시작한 경우 EC2 Instance Connect가 사전 설치되어 있으므로 이 절차를 건너뛸 수 있습니다.
>
> - AL2023 표준 AMI
> - Amazon Linux 2 2.0.20190618 이상
> - macOS Sonoma 14.2.1 이상
> - macOS Ventura 13.6.3 이상
> - macOS Monterey 12.7.2 이상
> - Ubuntu 20.04 이상

인스턴스의 운영 체제에 따라 다음 절차 중 하나를 사용하여 EC2 Instance Connect를 설치합니다.

Amazon Linux 2

EC2 Instance Connect로 시작된 인스턴스에 Amazon Linux 2를 설치하려면

1. SSH를 사용하여 인스턴스에 연결합니다.

   다음 명령에서 예제 값을 사용자의 값으로 바꿉니다. 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.

```
$ 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 SSH 클라이언트를 사용하여 Linux 인스턴스에 연결 주제를 참조하세요.

2. 인스턴스에 EC2 Instance Connect 패키지를 설치합니다.

```
[ec2-user ~]$ sudo yum install ec2-instance-connect
```

/opt/aws/bin/ 폴더에 3개의 새 스크립트가 표시됩니다.

```
eic_curl_authorized_keys
eic_parse_authorized_keys
eic_run_authorized_keys
```

3. (선택 사항) EC2 Instance Connect가 인스턴스에 성공적으로 설치되었는지 확인합니다.

```
[ec2-user ~]$ sudo less /etc/ssh/sshd_config
```

AuthorizedKeysCommand 및 AuthorizedKeysCommandUser 행에 다음 값이 포함되어 있으면 EC2 Instance Connect가 성공적으로 설치된 것입니다.

```
AuthorizedKeysCommand /opt/aws/bin/eic_run_authorized_keys %u %f
AuthorizedKeysCommandUser ec2-instance-connect
```

- AuthorizedKeysCommand는 eic_run_authorized_keys 스크립트를 설정하여 인스턴스 메타데이터에서 키를 찾습니다
- AuthorizedKeysCommandUser는 시스템 사용자를 로 설정합니다.ec2-instance-connect

> **Note**
>
> 이전에 AuthorizedKeysCommand 및 AuthorizedKeysCommandUser를 구성한 경우 EC2 Instance Connect 설치가 값을 변경하지 않으며 EC2 Instance Connect를 사용할 수 없습니다.

CentOS

CentOS에서 시작된 인스턴스에 EC2 Instance Connect를 설치하는 방법

1. SSH로 인스턴스에 연결합니다.

   다음 명령에서 예제 값을 사용자의 값으로 바꿉니다. 인스턴스를 시작할 때 인스턴스에 할당된 SSH 키 페어 및 인스턴스를 시작하는 데 사용한 AMI의 기본 사용자 이름을 사용합니다. CentOS의 경우 기본 사용자 이름은 `centos` 또는 `ec2-user`입니다.

   ```
   $ ssh -i my_ec2_private_key.pem centos@ec2-a-b-c-d.us-
   west-2.compute.amazonaws.com
   ```

   인스턴스 연결에 대한 자세한 내용은 SSH 클라이언트를 사용하여 Linux 인스턴스에 연결 주제를 참조하세요.

2. HTTP 또는 HTTPS 프록시를 사용하는 경우 현재 셸 세션에서 `http_proxy` 또는 `https_proxy` 환경 변수를 설정해야 합니다.

   프록시를 사용하지 않는 경우 이 단계를 건너뛸 수 있습니다.

   - HTTP 프록시 서버의 경우 다음 명령을 실행합니다.

     ```
     $ export http_proxy=http://hostname:port
     $ export https_proxy=http://hostname:port
     ```

   - HTTPS 프록시 서버의 경우 다음 명령을 실행합니다.

     ```
     $ export http_proxy=https://hostname:port
     $ export https_proxy=https://hostname:port
     ```

3. 다음 명령을 실행하여 인스턴스에 EC2 Instance Connect 패키지를 설치합니다.

   CentOS용 EC2 Instance Connect 구성 파일은 Intel/AMD(x86_64) 또는 ARM(AArch64)에서 실행되는 인스턴스 유형에 대해 CentOS 8 및 CentOS 9의 경우 다양한 RPM 패키지와 함께 Red Hat Package Manager(RPM) 패키지로 제공됩니다.

   운영 체제 및 CPU 아키텍처에 맞는 명령 블록을 사용합니다.

   - CentOS 8

     Intel/AMD(x86_64)

```
[ec2-user ~]$ mkdir /tmp/ec2-instance-connect
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-
west-2.s3.us-west-2.amazonaws.com/latest/linux_amd64/ec2-instance-
connect-2.0.0-3.rhel8.x86_64.rpm -o /tmp/ec2-instance-connect/ec2-instance-
connect.rpm
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-
west-2.amazonaws.com/latest/linux_amd64/ec2-instance-connect-
selinux-2.0.0-3.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-
selinux.rpm
[ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-
connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
```

ARM(AArch64)

```
[ec2-user ~]$ mkdir /tmp/ec2-instance-connect
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-
west-2.s3.us-west-2.amazonaws.com/latest/linux_arm64/ec2-instance-
connect-2.0.0-3.rhel8.aarch64.rpm -o /tmp/ec2-instance-connect/ec2-instance-
connect.rpm
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-
west-2.amazonaws.com/latest/linux_arm64/ec2-instance-connect-
selinux-2.0.0-3.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-
selinux.rpm
[ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-
connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
```

- CentOS 9

  Intel/AMD(x86_64)

```
[ec2-user ~]$ mkdir /tmp/ec2-instance-connect
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-
west-2.s3.us-west-2.amazonaws.com/latest/linux_amd64/ec2-instance-
connect-2.0.0-3.rhel9.x86_64.rpm -o /tmp/ec2-instance-connect/ec2-instance-
connect.rpm
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-
west-2.amazonaws.com/latest/linux_amd64/ec2-instance-connect-
selinux-2.0.0-3.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-
selinux.rpm
[ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-
connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
```

ARM(AArch64)

```
[ec2-user ~]$ mkdir /tmp/ec2-instance-connect
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-
west-2.s3.us-west-2.amazonaws.com/latest/linux_arm64/ec2-instance-
connect-2.0.0-3.rhel9.aarch64.rpm -o /tmp/ec2-instance-connect/ec2-instance-
connect.rpm
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-
west-2.amazonaws.com/latest/linux_arm64/ec2-instance-connect-
selinux-2.0.0-3.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-
selinux.rpm
[ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-
connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
```

`/opt/aws/bin/` 폴더에 다음과 같은 새 스크립트가 표시됩니다.

```
eic_run_authorized_keys
```

4. (선택 사항) EC2 Instance Connect가 인스턴스에 성공적으로 설치되었는지 확인합니다.

   - CentOS 8의 경우:

     ```
     [ec2-user ~]$ sudo less /lib/systemd/system/sshd.service.d/ec2-instance-
     connect.conf
     ```

   - CentOS 9의 경우:

     ```
     [ec2-user ~]$ sudo less /etc/ssh/sshd_config.d/60-ec2-instance-connect.conf
     ```

   `AuthorizedKeysCommand` 및 `AuthorizedKeysCommandUser` 행에 다음 값이 포함되어 있으면 EC2 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` 스크립트를 설정하여 인스턴스 메타데이터에서 키를 찾습니다

- `AuthorizedKeysCommandUser`는 시스템 사용자를 로 설정합니다.`ec2-instance-connect`

> **Note**
>
> 이전에 `AuthorizedKeysCommand` 및 `AuthorizedKeysCommandUser`를 구성한 경우 EC2 Instance Connect 설치가 값을 변경하지 않으며 EC2 Instance Connect를 사용할 수 없습니다.

macOS

macOS에서 시작된 인스턴스에 EC2 Instance Connect를 설치하는 방법

1. SSH로 인스턴스에 연결합니다.

   다음 명령에서 예제 값을 사용자의 값으로 바꿉니다. 인스턴스를 시작할 때 인스턴스에 할당된 SSH 키 페어 및 인스턴스를 시작하는 데 사용한 AMI의 기본 사용자 이름을 사용합니다. macOS 인스턴스의 경우 기본 사용자 이름은 `ec2-user`입니다.

   ```
   $ 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 SSH 클라이언트를 사용하여 Linux 인스턴스에 연결 주제를 참조하세요.

2. 다음 명령을 사용하여 Homebrew를 업데이트합니다. 업데이트에는 Homebrew에서 알고 있는 소프트웨어가 나열됩니다. EC2 Instance Connect 패키지는 macOS 인스턴스에서 Homebrew를 통해 제공됩니다. 자세한 내용은 Amazon EC2 인스턴스에서 운영 체제 및 소프트웨어 업데이트 단원을 참조하세요.

   ```
   [ec2-user ~]$ brew update
   ```

3. 인스턴스에 EC2 Instance Connect 패키지를 설치합니다. 그러면 소프트웨어가 설치되고 이를 사용하도록 sshd가 구성됩니다.

   ```
   [ec2-user ~]$ brew install ec2-instance-connect
   ```

/opt/aws/bin/ 폴더에 다음과 같은 새 스크립트가 표시됩니다.

```
eic_run_authorized_keys
```

4. (선택 사항) EC2 Instance Connect가 인스턴스에 성공적으로 설치되었는지 확인합니다.

```
[ec2-user ~]$ sudo less /etc/ssh/sshd_config.d/60-ec2-instance-connect.conf
```

AuthorizedKeysCommand 및 AuthorizedKeysCommandUser 행에 다음 값이 포함되어 있으면 EC2 Instance Connect가 성공적으로 설치된 것입니다.

```
AuthorizedKeysCommand /opt/aws/bin/eic_run_authorized_keys %u %f
AuthorizedKeysCommandUser ec2-instance-connect
```

- AuthorizedKeysCommand는 eic_run_authorized_keys 스크립트를 설정하여 인스턴스 메타데이터에서 키를 찾습니다
- AuthorizedKeysCommandUser는 시스템 사용자를 로 설정합니다.ec2-instance-connect

> **Note**
>
> 이전에 AuthorizedKeysCommand 및 AuthorizedKeysCommandUser를 구성한 경우 EC2 Instance Connect 설치가 값을 변경하지 않으며 EC2 Instance Connect를 사용할 수 없습니다.

RHEL

Red Hat Enterprise Linux(RHEL)에서 시작된 인스턴스에 EC2 Instance Connect를 설치하는 방법

1. SSH로 인스턴스에 연결합니다.

   다음 명령에서 예제 값을 사용자의 값으로 바꿉니다. 인스턴스를 시작할 때 인스턴스에 할당된 SSH 키 페어 및 인스턴스를 시작하는 데 사용한 AMI의 기본 사용자 이름을 사용합니다. RHEL의 경우 기본 사용자 이름은 ec2-user 또는 root입니다.

```
$ 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 SSH 클라이언트를 사용하여 Linux 인스턴스에 연결 주제를 참조하세요.

2. HTTP 또는 HTTPS 프록시를 사용하는 경우 현재 셸 세션에서 `http_proxy` 또는 `https_proxy` 환경 변수를 설정해야 합니다.

   프록시를 사용하지 않는 경우 이 단계를 건너뛸 수 있습니다.

   - HTTP 프록시 서버의 경우 다음 명령을 실행합니다.

     ```
     $ export http_proxy=http://hostname:port
     $ export https_proxy=http://hostname:port
     ```

   - HTTPS 프록시 서버의 경우 다음 명령을 실행합니다.

     ```
     $ export http_proxy=https://hostname:port
     $ export https_proxy=https://hostname:port
     ```

3. 다음 명령을 실행하여 인스턴스에 EC2 Instance Connect 패키지를 설치합니다.

   RHEL용 EC2 Instance Connect 구성 파일은 Intel/AMD(x86_64) 또는 ARM(AArch64)에서 실행되는 인스턴스 유형에 대해 RHEL 8 및 RHEL 9의 경우 다양한 RPM 패키지와 함께 Red Hat Package Manager(RPM) 패키지로 제공됩니다.

   운영 체제 및 CPU 아키텍처에 맞는 명령 블록을 사용합니다.

   - RHEL 8

     Intel/AMD(x86_64)

     ```
     [ec2-user ~]$ mkdir /tmp/ec2-instance-connect
     [ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-
     west-2.s3.us-west-2.amazonaws.com/latest/linux_amd64/ec2-instance-
     connect-2.0.0-3.rhel8.x86_64.rpm -o /tmp/ec2-instance-connect/ec2-instance-
     connect.rpm
     [ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-
     west-2.amazonaws.com/latest/linux_amd64/ec2-instance-connect-
     ```

```
selinux-2.0.0-3.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-
selinux.rpm
[ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-
connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
```

ARM(AArch64)

```
[ec2-user ~]$ mkdir /tmp/ec2-instance-connect
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-
west-2.s3.us-west-2.amazonaws.com/latest/linux_arm64/ec2-instance-
connect-2.0.0-3.rhel8.aarch64.rpm -o /tmp/ec2-instance-connect/ec2-instance-
connect.rpm
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-
west-2.amazonaws.com/latest/linux_arm64/ec2-instance-connect-
selinux-2.0.0-3.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-
selinux.rpm
[ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-
connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
```

- RHEL 9

  Intel/AMD(x86_64)

```
[ec2-user ~]$ mkdir /tmp/ec2-instance-connect
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-
west-2.s3.us-west-2.amazonaws.com/latest/linux_amd64/ec2-instance-
connect-2.0.0-3.rhel9.x86_64.rpm -o /tmp/ec2-instance-connect/ec2-instance-
connect.rpm
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-
west-2.amazonaws.com/latest/linux_amd64/ec2-instance-connect-
selinux-2.0.0-3.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-
selinux.rpm
[ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-
connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
```

  ARM(AArch64)

```
[ec2-user ~]$ mkdir /tmp/ec2-instance-connect
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-
west-2.s3.us-west-2.amazonaws.com/latest/linux_arm64/ec2-instance-
connect-2.0.0-3.rhel9.aarch64.rpm -o /tmp/ec2-instance-connect/ec2-instance-
connect.rpm
```

```
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-
west-2.amazonaws.com/latest/linux_arm64/ec2-instance-connect-
selinux-2.0.0-3.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-
selinux.rpm
[ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-
connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
```

/opt/aws/bin/ 폴더에 다음과 같은 새 스크립트가 표시됩니다.

```
eic_run_authorized_keys
```

4. (선택 사항) EC2 Instance Connect가 인스턴스에 성공적으로 설치되었는지 확인합니다.

   - RHEL 8의 경우:

     ```
     [ec2-user ~]$ sudo less /lib/systemd/system/sshd.service.d/ec2-instance-
     connect.conf
     ```

   - RHEL 9의 경우:

     ```
     [ec2-user ~]$ sudo less /etc/ssh/sshd_config.d/60-ec2-instance-connect.conf
     ```

   AuthorizedKeysCommand 및 AuthorizedKeysCommandUser 행에 다음 값이 포함되어 있으면 EC2 Instance Connect가 성공적으로 설치된 것입니다.

   ```
   AuthorizedKeysCommand /opt/aws/bin/eic_run_authorized_keys %u %f
   AuthorizedKeysCommandUser ec2-instance-connect
   ```

   - AuthorizedKeysCommand는 eic_run_authorized_keys 스크립트를 설정하여 인스턴스 메타데이터에서 키를 찾습니다
   - AuthorizedKeysCommandUser는 시스템 사용자를 로 설정합니다.ec2-instance-connect

> **Note**
>
> 이전에 `AuthorizedKeysCommand` 및 `AuthorizedKeysCommandUser`를 구성한 경우 EC2 Instance Connect 설치가 값을 변경하지 않으며 EC2 Instance Connect를 사용할 수 없습니다.

Ubuntu

Ubuntu 16.04 이상으로 시작된 인스턴스에 EC2 Instance Connect를 설치하려면

1. SSH로 인스턴스에 연결합니다.

   다음 명령에서 예제 값을 사용자의 값으로 바꿉니다. 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.

   ```
   $ ssh -i my_ec2_private_key.pem ubuntu@ec2-a-b-c-d.us-
   west-2.compute.amazonaws.com
   ```

   인스턴스 연결에 대한 자세한 내용은 SSH 클라이언트를 사용하여 Linux 인스턴스에 연결 주제를 참조하세요.

2. (선택 사항) 인스턴스에 최신 Ubuntu AMI가 있는지 확인합니다.

   다음 명령을 실행하여 인스턴스의 모든 패키지를 업데이트합니다.

   ```
   ubuntu:~$ sudo apt-get update
   ```

   ```
   ubuntu:~$ sudo apt-get upgrade
   ```

3. 인스턴스에 EC2 Instance Connect 패키지를 설치합니다.

   ```
   ubuntu:~$ sudo apt-get install ec2-instance-connect
   ```

   `/usr/share/ec2-instance-connect/` 폴더에 3개의 새 스크립트가 표시됩니다.

   ```
   eic_curl_authorized_keys
   ```

```
eic_parse_authorized_keys
eic_run_authorized_keys
```

4. (선택 사항) EC2 Instance Connect가 인스턴스에 성공적으로 설치되었는지 확인합니다.

```
ubuntu:~$ sudo less /lib/systemd/system/ssh.service.d/ec2-instance-connect.conf
```

AuthorizedKeysCommand 및 AuthorizedKeysCommandUser 행에 다음 값이 포함되어 있으면 EC2 Instance Connect가 성공적으로 설치된 것입니다.

```
AuthorizedKeysCommand /usr/share/ec2-instance-connect/eic_run_authorized_keys %
%u %%f
AuthorizedKeysCommandUser ec2-instance-connect
```

- AuthorizedKeysCommand는 eic_run_authorized_keys 스크립트를 설정하여 인스턴스 메타데이터에서 키를 찾습니다
- AuthorizedKeysCommandUser는 시스템 사용자를 로 설정합니다.ec2-instance-connect

> **Note**
>
> 이전에 AuthorizedKeysCommand 및 AuthorizedKeysCommandUser를 구성한 경우 EC2 Instance Connect 설치가 값을 변경하지 않으며 EC2 Instance Connect를 사용할 수 없습니다.

EC2 Instance Connect를 사용하여 Linux 인스턴스에 연결

다음 지침에서는 Amazon EC2 콘솔, AWS CLI 또는 SSH 클라이언트를 통해 EC2 Instance Connect를 사용하여 Linux 인스턴스에 연결하는 방법을 설명합니다.

EC2 Instance Connect를 사용하여 콘솔 또는 AWS CLI를 통해 인스턴스에 연결할 때 EC2 Instance Connect API는 SSH 퍼블릭 키를 자동으로 인스턴스 메타데이터에 푸시하여 60초 동안 유지합니다. 사용자에게 연결된 IAM 정책이 이 작업을 승인합니다. 자체 SSH 키를 사용하는 것을 선호하는 경우 SSH 클라이언트를 사용하고 EC2 Instance Connect를 사용하여 SSH 키를 인스턴스로 명시적으로 푸시할 수 있습니다.

고려 사항

EC2 Instance Connect를 사용하여 인스턴스에 연결한 후 SSH 세션이 종료될 때까지 연결이 유지됩니다. 연결 기간은 IAM 자격 증명 기간에 따라 결정되지 않습니다. IAM 자격 증명이 만료되더라도 연결은 계속 유지됩니다. EC2 Instance Connect 콘솔 환경을 사용할 때 IAM 자격 증명이 만료되면 브라우저 페이지를 닫아 연결을 종료합니다. 자체 SSH 클라이언트와 EC2 Instance Connect를 사용하여 키를 푸시할 때 SSH 제한 시간 값을 설정하여 SSH 세션을 자동으로 종료할 수 있습니다.

요구 사항

시작하기 전에 사전 조건을 검토해야 합니다.

연결 옵션

- Amazon EC2 콘솔을 사용하여 연결
- AWS CLI를 사용하여 연결
- 자체 키 및 SSH 클라이언트를 사용하여 연결
- 문제 해결

Amazon EC2 콘솔을 사용하여 연결

Amazon EC2 콘솔을 통해 EC2 Instance Connect를 사용하여 인스턴스를 연결할 수 있습니다.

요구 사항

Amazon EC2 콘솔을 사용하여 연결하려면 인스턴스에 퍼블릭 IPv4 또는 퍼블릭 IPv6 주소가 있어야 합니다. 인스턴스에 프라이빗 IPv4 주소만 있는 경우 ec2-instance-connect AWS CLI를 사용하여 연결할 수 있습니다.

Amazon EC2 콘솔을 사용하여 인스턴스에 연결

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 탭을 선택합니다.
5. 연결 유형에서 EC2 Instance Connect를 사용하여 연결을 선택합니다.
6. 선택권이 있는 경우 연결할 IP 주소를 선택하세요. 그렇게 하지 않으면 IP 주소가 자동으로 선택됩니다.
7. 사용자 이름에서 사용자 이름을 확인합니다.

8. 연결을 선택하여 연결을 설정합니다. 브라우저 내 터미널 창이 열립니다.

AWS CLI를 사용하여 연결

ec2-instance-connect AWS CLI를 사용하여 SSH 클라이언트로 인스턴스에 연결할 수 있습니다. EC2 Instance Connect는 지정된 연결 유형에 따라 사전 정의된 순서로 사용 가능한 IP 주소를 사용하여 연결을 설정하려고 시도합니다. IP 주소를 사용할 수 없는 경우 자동으로 그 다음 IP 주소를 순서대로 시도합니다.

연결 유형

`auto`(기본값)

EC2 Instance Connect는 다음 순서와 해당 연결 유형으로 인스턴스의 IP 주소를 사용하여 연결을 시도합니다.

1. 퍼블릭 IPv4: `direct`
2. 프라이빗 IPv4: `eice`
3. 퍼블릭 IPv6: `direct`

`direct`

EC2 Instance Connect는 다음 순서로 인스턴스의 IP 주소를 사용하여 연결을 시도합니다.

1. 퍼블릭 IPv4
2. 퍼블릭 IPv6
3. 프라이빗 IPv4(EC2 Instance Connect 엔드포인트를 통해 연결되지 않음)

`eice`

EC2 Instance Connect는 인스턴스의 프라이빗 IPv4 주소와 EC2 Instance Connect 엔드포인트를 사용하여 연결을 시도합니다.

> **Note**
>
> `auto` 연결 유형의 동작은 향후 변경될 수도 있습니다. 원하는 연결 유형이 사용되도록 `--connection-type`을 `direct` 또는 `eice`로 명시적으로 설정하는 것이 좋습니다.
> ec2-instance-connect AWS CLI를 사용할 때는 프라이빗 IPv6 주소에 연결할 수 없습니다.

요구 사항

AWS CLI 버전 2를 사용해야 합니다. 자세한 내용은 최신 버전의 AWS CLI 설치 또는 업데이트를 참조하세요.

인스턴스 ID를 사용하여 인스턴스에 연결

인스턴스 ID만 알고 있고, 인스턴스에 연결할 때 사용할 연결 유형을 EC2 Instance Connect가 결정하도록 설정하려면 ec2-instance-connect CLI를 사용하고 ssh 명령과 인스턴스 ID를 지정합니다.

```
aws ec2-instance-connect ssh --instance-id i-1234567890example
```

인스턴스 ID 및 EC2 Instance Connect 엔드포인트를 사용하여 인스턴스에 연결

EC2 Instance Connect 엔드포인트를 통해 인스턴스에 연결하려면 앞선 명령을 사용하고 --connection-type 파라미터를 eice 값으로 지정합니다.

```
aws ec2-instance-connect ssh --instance-id i-1234567890example --connection-type eice
```

인스턴스 ID와 자체 프라이빗 키 파일을 사용하여 인스턴스에 연결

자체 프라이빗 키를 사용하여 EC2 Instance Connect 엔드포인트를 통해 인스턴스에 연결하려면 인스턴스 ID와 프라이빗 키 파일의 경로를 지정합니다. 경로에 *file://*을 포함하지 마세요. 예를 들어 *file:///path/to/key*를 사용하면 실패합니다.

```
aws ec2-instance-connect ssh --instance-id i-1234567890example --private-key-file /
path/to/key.pem
```

> **Tip**
>
> 이 명령을 사용할 때 오류가 발생하면 AWS CLI 버전 2를 사용하고 있는지 확인해야 합니다. ssh 명령은 이 메이저 버전에서만 사용 가능하기 때문입니다. 또한 최신 기능에 액세스하려면 AWS CLI 버전 2의 최신 마이너 버전으로 정기적으로 업데이트하는 것이 좋습니다. 자세한 내용은 AWS Command Line Interface 사용 설명서의 AWS CLI 버전 2 정보를 참조하세요.

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 EC2 Instance Connect 기능에서 퍼블릭 키를 인스턴스로 푸시할 수 있습니다. 이 연결 방법은 퍼블릭 및 프라이빗 IP 주소가 있는 인스턴스에서 작동합니다.

요구 사항

- 키 페어에 대한 요구 사항
  - 지원되는 유형: RSA(OpenSSH 및 SSH2) 및 ED25519
  - 지원되는 길이: 2,048 및 4,096
  - 자세한 내용은 서드 파티 도구를 사용하여 키 페어를 생성하고 Amazon EC2로 퍼블릭 키 가져오기 섹션을 참조하세요.
- 프라이빗 IP 주소만 있는 인스턴스에 연결하는 경우 SSH 세션을 시작하는 로컬 컴퓨터에서 EC2 Instance Connect 서비스 엔드포인트(인스턴스에 SSH 퍼블릭 키를 푸시하는 경우)에 연결하고 인스턴스의 프라이빗 IP 주소에 네트워크로 연결하여 SSH 세션을 설정해야 합니다. 인터넷 또는 AWS Direct Connect 퍼블릭 가상 인터페이스를 통해 EC2 Instance Connect 서비스 엔드포인트에 연결할 수 있습니다. 인스턴스의 프라이빗 IP 주소에 연결하려면 AWS Direct Connect, AWS Site-to-Site VPN 또는 VPC 피어링과 같은 서비스를 활용하면 됩니다.

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_key` 및 `my_key.pub`을 생성할 수 있습니다.

   ```
   ssh-keygen -t rsa -f my_key
   ```

2. SSH 퍼블릭 키를 인스턴스에 푸시합니다.

   send-ssh-public-key 명령을 사용하여 SSH 퍼블릭 키를 인스턴스에 푸시합니다. AL2023 또는 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 \
       --region us-west-2 \
       --availability-zone us-west-2b \
       --instance-id i-001234a4bf70dec41EXAMPLE \
       --instance-os-user ec2-user \
       --ssh-public-key file://my_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 DNS 이름을 지정합니다(프라이빗 네트워크를 통해 연결하는 경우 프라이빗 DNS 이름 또는 IP 주소 지정). ssh 구성의 파일과 지정된 키만 연결에 사용되도록 하려면 `IdentitiesOnly=yes` 옵션을 추가합니다.

   ```
   ssh -o "IdentitiesOnly=yes" -i my_key ec2-
   user@ec2-198-51-100-1.compute-1.amazonaws.com
   ```

   다음 예제에서는 `timeout 3600`을 사용하여 SSH 세션이 1시간 후에 종료되도록 설정합니다. 세션 중에 시작된 프로세스는 세션이 종료된 후에도 인스턴스에서 계속 실행될 수 있습니다.

   ```
   timeout 3600 ssh -o “IdentitiesOnly=yes” -i my_key ec2-
   user@ec2-198-51-100-1.compute-1.amazonaws.com
   ```

문제 해결

인스턴스에 연결을 시도하는 동안 오류가 발생한 경우 다음 섹션을 참조하세요.

- Amazon EC2 Linux 인스턴스 연결 문제 해결
- EC2 Instance Connect를 사용하여 EC2 인스턴스에 연결할 때 발생하는 문제를 해결하려면 어떻게 해야 하나요?

EC2 Instance Connect 제거

EC2 Instance Connect를 비활성화하려면 Linux 인스턴스에 연결하여 OS에 설치된 `ec2-instance-connect` 패키지를 제거하세요. `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

EC2 Instance Connect이 사전 구성된 AL2023 및 Amazon Linux 2 2.0.20190618 이상에서 EC2 Instance Connect를 제거할 수 있습니다.

Amazon Linux를 사용하여 시작된 인스턴스에서 EC2 Instance Connect 제거

1. SSH로 인스턴스에 연결합니다. AL2023 또는 Amazon Linux 2 AMI의 기본 사용자 이름(ec2-user) 및 인스턴스를 시작할 때 인스턴스에서 사용한 SSH 키 페어를 지정합니다.

   예를 들어 다음 ssh 명령은 키 페어 ec2-a-b-c-d.us-west-2.compute.amazonaws.com를 사용하여 퍼블릭 DNS 이름 my_ec2_private_key.pem에 연결됩니다.

   ```
   $ ssh -i my_ec2_private_key.pem ec2-user@ec2-a-b-c-d.us-
   west-2.compute.amazonaws.com
   ```

2. ec2-instance-connect 명령을 사용하여 yum 패키지를 제거합니다.

   ```
   [ec2-user ~]$ sudo yum remove ec2-instance-connect
   ```

Ubuntu

Ubuntu AMI를 사용하여 시작된 인스턴스에서 EC2 Instance Connect 제거

1. SSH로 인스턴스에 연결합니다. 인스턴스를 시작할 때 사용한 SSH 키 페어와 Ubuntu AMI의 기본 사용자 이름(ubuntu)을 지정합니다.

   예를 들어 다음 ssh 명령은 키 페어 ec2-a-b-c-d.us-west-2.compute.amazonaws.com를 사용하여 퍼블릭 DNS 이름 my_ec2_private_key.pem에 연결됩니다.

   ```
   $ ssh -i my_ec2_private_key.pem ubuntu@ec2-a-b-c-d.us-
   west-2.compute.amazonaws.com
   ```

2. ec2-instance-connect 명령을 사용하여 apt-get 패키지를 제거합니다.

   ```
   ubuntu:~$ sudo apt-get remove ec2-instance-connect
   ```

EC2 Instance Connect 엔드포인트를 사용하여 인스턴스에 연결

EC2 Instance Connect 엔드포인트를 사용하면 Bastion Host를 사용하거나 Virtual Private Cloud(VPC)에서 인터넷에 직접 연결하지 않고도 인터넷에서 인스턴스에 안전하게 연결할 수 있습니다.

이점

- 인스턴스에 퍼블릭 IPv4 주소가 없어도 인스턴스에 연결할 수 있습니다. AWS는 실행 중인 인스턴스 및 탄력적 IP 주소와 연결된 퍼블릭 IPv4 주소를 포함하여 모든 퍼블릭 IPv4 주소에 대해 요금을 청구합니다. 자세한 내용은 Amazon VPC 요금 페이지의 퍼블릭 IPv4 주소 탭을 참조하세요.
- 인터넷 게이트웨이를 통해 VPC에서 인터넷에 직접 연결하지 않고도 인터넷에서 인스턴스에 연결할 수 있습니다.
- IAM 정책 및 권한으로 인스턴스에 연결하기 위해 EC2 Instance Connect 엔드포인트의 생성 및 사용에 대한 액세스를 제어할 수 있습니다.
- 인스턴스에 연결하려는 모든 시도(성공 및 실패)는 CloudTrail에 기록됩니다.

요금

EC2 Instance Connect 엔드포인트를 사용하는 데 추가 비용은 없습니다. EC2 Instance Connect 엔드포인트를 사용하여 서로 다른 가용 영역의 인스턴스에 연결하는 경우 가용 영역 간 데이터 전송에 대한 추가 요금이 청구됩니다.

내용

- 작동 방법
- 고려 사항
- EC2 Instance Connect 엔드포인트를 사용할 수 있는 권한 부여
- EC2 Instance Connect 엔드포인트 보안 그룹
- EC2 Instance Connect 엔드포인트 생성
- EC2 인스턴스 연결 엔드포인트를 사용하여 Amazon EC2 인스턴스에 연결
- EC2 Instance Connect 엔드포인트를 통해 설정된 연결 기록
- EC2 Instance Connect 엔드포인트 삭제
- EC2 Instance Connect 엔드포인트에 대한 서비스 연결 역할
- EC2 Instance Connect 엔드포인트의 할당량

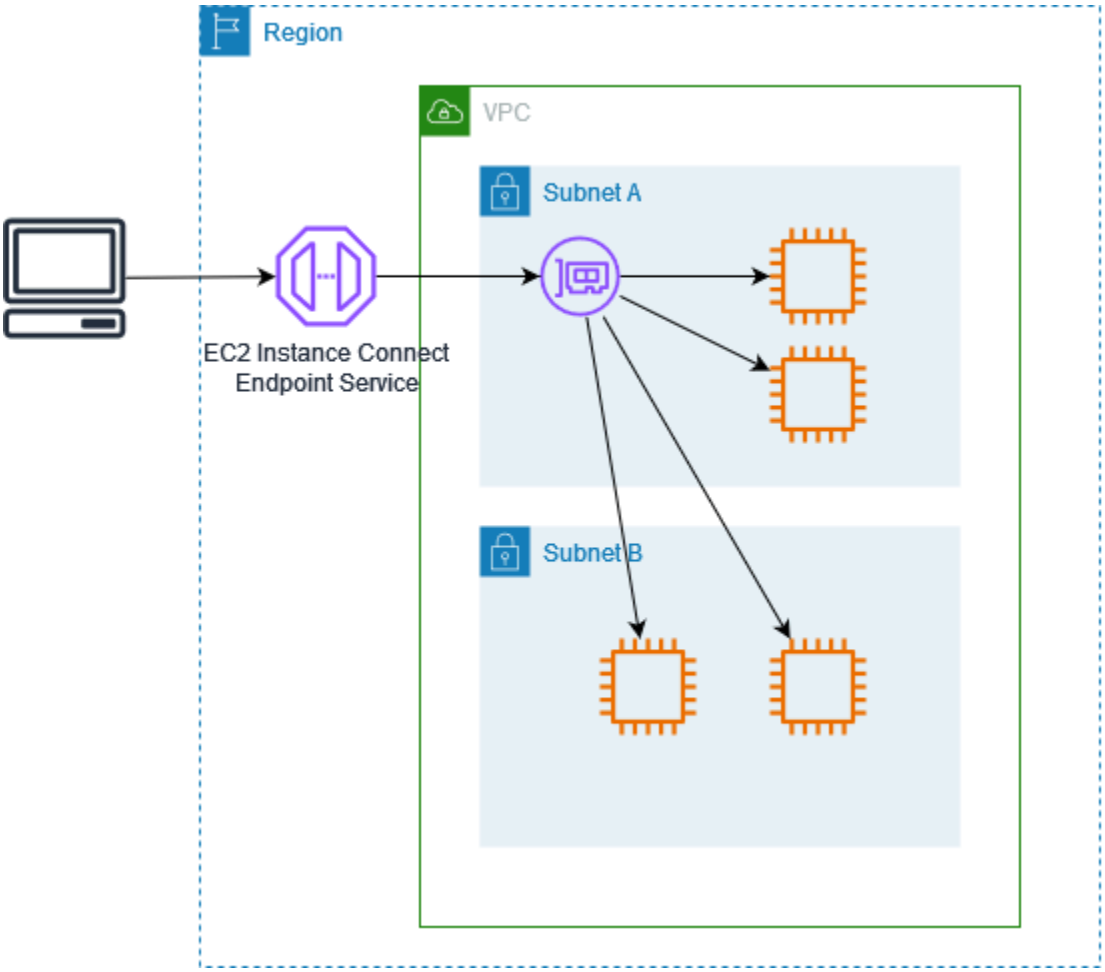
작동 방법

EC2 Instance Connect 엔드포인트는 자격 증명을 인식하는 TCP 프록시입니다. EC2 Instance Connect 엔드포인트 서비스는 IAM 엔터티의 자격 증명을 사용하여 컴퓨터에서 엔드포인트로 연결되는 프라이빗 터널을 설정합니다. 트래픽은 VPC에 도달하기 전에 인증 및 승인됩니다.

인스턴스로의 인바운드 트래픽을 제한하는 추가 보안 그룹 규칙을 구성할 수 있습니다. 예를 들어 인바운드 규칙을 사용하여 EC2 Instance Connect 엔드포인트에서 생성되는 트래픽만 관리 포트에서 허용할 수 있습니다.

엔드포인트가 VPC의 모든 서브넷에 있는 모든 인스턴스에 연결할 수 있도록 라우팅 테이블 규칙을 구성할 수 있습니다.

다음 다이어그램은 사용자가 EC2 Instance Connect 엔드포인트를 사용하여 인터넷에서 인스턴스에 연결하는 방법을 보여줍니다. 먼저 서브넷 A에서 EC2 Instance Connect 엔드포인트를 생성합니다. 서브넷의 엔드포인트에 대한 네트워크 인터페이스를 생성합니다. 그러면 VPC의 인스턴스로 향하는 트래픽의 진입점 역할을 합니다. 서브넷 B의 라우팅 테이블이 서브넷 A에서 생성되는 트래픽을 허용하는 경우 엔드포인트를 사용하여 서브넷 B의 인스턴스에 도달할 수 있습니다.



고려 사항

시작하기 전에 다음을 고려합니다.

- EC2 Instance Connect 엔드포인트는 대용량 데이터 전송이 아닌 관리 트래픽 사용 사례를 위해 특별히 고안되었습니다. 대용량 데이터 전송이 제한됩니다.
- 인스턴스에 IPv4 주소(프라이빗 또는 퍼블릭)가 있어야 합니다. EC2 Instance Connect 엔드포인트에서는 IPv6 주소를 사용하는 연결을 지원하지 않습니다.
- (Linux 인스턴스) 자체 키 페어를 사용하는 경우 모든 Linux AMI를 사용할 수 있습니다. 그렇지 않으면 인스턴스에 EC2 Instance Connect를 설치해야 합니다. EC2 Instance Connect를 포함하는 AMI 및 지원되는 다른 AMI에 EC2 Instance Connect를 설치하는 방법에 대한 자세한 내용은 EC2 Instance Connect 설치 섹션을 참조하세요.
- EC2 Instance Connect 엔드포인트를 생성할 때 이 엔드포인트에 보안 그룹을 할당할 수 있습니다. 그렇지 않으면 VPC에 대한 기본 보안 그룹을 사용합니다. EC2 Instance Connect 엔드포인트의 보안 그룹은 대상 인스턴스로의 아웃바운드 트래픽을 허용해야 합니다. 자세한 내용은 EC2 Instance Connect 엔드포인트 보안 그룹 섹션을 참조하세요.
- 요청을 인스턴스로 라우팅할 때 클라이언트의 소스 IP 주소를 보존하도록 EC2 Instance Connect 엔드포인트를 구성할 수 있습니다. 그렇지 않으면 네트워크 인터페이스의 IP 주소는 모든 수신 트래픽에 대한 클라이언트 IP 주소가 됩니다.
  - 클라이언트 IP 보존을 켜면 인스턴스의 보안 그룹에서 클라이언트의 트래픽을 허용해야 합니다. 또한 인스턴스는 EC2 Instance Connect 엔드포인트와 동일한 VPC에 있어야 합니다.
  - 클라이언트 IP 보존을 끄면 인스턴스의 보안 그룹에서 VPC의 트래픽을 허용해야 합니다. 이 값이 기본값입니다.
  - 다음 인스턴스 유형, C1, CG1, CG2, G1, HI1, M1, M2, M3, T1은 클라이언트 IP 보존을 지원하지 않습니다. 클라이언트 IP 보존을 켜고 EC2 Instance Connect 엔드포인트를 사용하여 이러한 인스턴스 유형 중 하나의 인스턴스에 연결하려고 하면 연결에 실패합니다.
  - 트래픽이 전송 게이트웨이를 통해 라우팅되면 클라이언트 IP 보존이 지원되지 않습니다.
- EC2 Instance Connect 엔드포인트를 생성할 때 AWS Identity and Access Management(IAM)의 Amazon EC2 서비스에 대해 서비스 연결 역할이 자동으로 생성됩니다. Amazon EC2는 서비스 연결 역할을 사용하여 계정에 네트워크 인터페이스를 프로비저닝하며, 이 인터페이스는 EC2 Instance Connect 엔드포인트를 생성할 때 필요합니다. 자세한 내용은 EC2 Instance Connect 엔드포인트에 대한 서비스 연결 역할 섹션을 참조하세요.
- VPC 및 서브넷당 1개의 EC2 Instance Connect 엔드포인트만 생성할 수 있습니다. 자세한 내용은 EC2 Instance Connect 엔드포인트의 할당량 섹션을 참조하세요. 동일한 VPC 내의 다른 가용 영역에 다른 EC2 Instance Connect 엔드포인트를 생성해야 하는 경우 먼저 기존 EC2 Instance Connect 엔드포인트를 삭제해야 합니다. 그렇지 않으면 할당량 오류가 발생합니다.
- 각 EC2 인스턴스 연결 엔드포인트는 최대 20개의 동시 연결을 지원할 수 있습니다.

- 설정된 TCP 연결의 최대 지속 시간은 1시간(3,600초)입니다. IAM 정책에서 허용되는 최대 지속 시간을 지정할 수 있습니다(최대 3,600초). 자세한 내용은 인스턴스에 연결하기 위해 EC2 Instance Connect 엔드포인트를 사용할 수 있는 권한 섹션을 참조하세요.

  연결 기간은 IAM 자격 증명 기간에 따라 결정되지 않습니다. IAM 자격 증명이 만료된 경우 지정된 최대 기간에 도달할 때까지 연결이 계속 유지됩니다. EC2 Instance Connect 엔드포인트 콘솔 환경을 사용하여 인스턴스에 연결할 때 최대 터널 기간(초)을 IAM 자격 증명 기간보다 작은 값으로 설정합니다. IAM 자격 증명이 조기에 만료되는 경우 브라우저 페이지를 닫아 인스턴스에 대한 연결을 종료합니다.

EC2 Instance Connect 엔드포인트를 사용할 수 있는 권한 부여

기본적으로 IAM 엔터티에는 EC2 Instance Connect 엔드포인트를 생성, 설명 또는 수정할 수 있는 권한이 없습니다. IAM 관리자는 필요한 리소스에서 특정 작업을 수행하는 데 필요한 권한을 부여하는 IAM 정책을 생성할 수 있습니다.

IAM 정책 생성에 대한 자세한 내용은 IAM 사용 설명서의 IAM 정책 생성을 참조하세요.

다음 예제 정책에서는 EC2 Instance Connect 엔드포인트에 대해 필요한 권한을 제어하는 방법을 보여줍니다.

예시

- EC2 Instance Connect 엔드포인트 생성, 설명 및 삭제할 수 있는 권한
- 인스턴스에 연결하기 위해 EC2 Instance Connect 엔드포인트를 사용할 수 있는 권한
- 특정 IP 주소 범위에서만 연결할 수 있는 권한

EC2 Instance Connect 엔드포인트 생성, 설명 및 삭제할 수 있는 권한

EC2 Instance Connect 엔드포인트를 생성하려면 사용자에게 다음 작업에 대한 권한이 필요합니다.

- `ec2:CreateInstanceConnectEndpoint`
- `ec2:CreateNetworkInterface`
- `ec2:CreateTags`
- `iam:CreateServiceLinkedRole`

EC2 Instance Connect 엔드포인트를 설명 및 삭제하려면 사용자에게 다음 작업에 대한 권한이 필요합니다.

- ec2:DescribeInstanceConnectEndpoints
- ec2:DeleteInstanceConnectEndpoint

모든 서브넷에서 EC2 Instance Connect 엔드포인트의 생성, 설명, 삭제 권한을 부여하는 정책을 생성할 수 있습니다. 또는 서브넷 ARN을 허용된 Resource로 지정하거나 ec2:SubnetID 조건 키를 사용하여 지정된 서브넷에 대한 작업만 제한할 수 있습니다. aws:ResourceTag 조건 키를 사용하여 특정 태그를 사용한 엔드포인트 생성을 명시적으로 허용하거나 거부할 수도 있습니다. 자세한 내용은 IAM 사용 설명서에서 IAM의 정책 및 권한을 참조하세요.

IAM 정책 예제

다음 IAM 정책 예제에서 Resource 섹션은 별표(*)로 지정된 모든 서브넷에서 엔드포인트를 생성하고 삭제할 수 있는 권한을 부여합니다. ec2:Describe* API 작업은 리소스 수준 권한을 지원하지 않습니다. 따라서 Resource 요소에 * 와일드카드가 필요합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Sid": "GrantAllActionsInAllSubnets",
            "Action": [
                "ec2:CreateInstanceConnectEndpoint",
                "ec2:DeleteInstanceConnectEndpoint",
                "ec2:CreateNetworkInterface",
                "ec2:CreateTags",
                "iam:CreateServiceLinkedRole"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:ec2:region:account-id:subnet/*"
        },
        {
            "Action": [
                "ec2:CreateNetworkInterface"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:ec2:::security-group/*"
        },
        {
            "Sid": "DescribeInstanceConnectEndpoints",
            "Action": [
                "ec2:DescribeInstanceConnectEndpoints"
            ],
```

```
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

인스턴스에 연결하기 위해 EC2 Instance Connect 엔드포인트를 사용할 수 있는 권한

`ec2-instance-connect:OpenTunnel` 작업은 EC2 Instance Connect 엔드포인트를 통해 연결하도록 인스턴스에 대한 TCP 연결을 설정하는 권한을 부여합니다. 사용할 EC2 Instance Connect 엔드포인트를 지정할 수 있습니다. 또는 별표(*)가 있는 `Resource`를 사용하면 사용 가능한 모든 EC2 Instance Connect 엔드포인트를 사용할 수 있습니다. 조건 키로서 리소스 태그의 존재 여부에 따라 인스턴스에 대한 액세스를 제한할 수도 있습니다.

조건

- `ec2-instance-connect:remotePort` - TCP 연결 설정에 사용할 수 있는 인스턴스의 포트. 이 조건 키가 사용되는 경우 정책에 지정된 포트가 아닌 다른 포트에서 인스턴스에 연결하려고 하면 오류가 발생합니다.
- `ec2-instance-connect:privateIpAddress` - TCP 연결을 설정하려는 인스턴스와 연결된 대상 프라이빗 IP 주소. 단일 IP 주소(예: `10.0.0.1/32`)를 지정하거나 CIDR을 통해 IP 범위(예: `10.0.1.0/28`)를 지정할 수 있습니다. 이 조건 키가 사용될 때 프라이빗 IP 주소가 다르거나 CIDR 범위를 벗어난 인스턴스에 연결하려고 하면 오류가 발생합니다.
- `ec2-instance-connect:maxTunnelDuration` - 설정된 TCP 연결의 최대 지속 시간. 단위는 초이며, 지속 시간은 최소 1초에서 최대 3,600초(1시간)입니다. 조건이 지정되지 않은 경우 기본 지속 시간은 3,600초(1시간)로 설정됩니다. IAM 정책에 지정된 지속 시간보다 오래 또는 기본 최대값보다 긴 지속 시간 동안 인스턴스에 연결하려고 시도하면 오류가 발생합니다. 지정된 지속 시간이 지나면 연결이 해제됩니다.

  `maxTunnelDuration`이 IAM 정책에 지정되고 지정된 값이 3,600초(기본값) 미만인 경우 인스턴스에 연결할 때 명령에 `--max-tunnel-duration`을 지정해야 합니다. 인스턴스 연결 방법에 대한 자세한 내용은 EC2 인스턴스 연결 엔드포인트를 사용하여 Amazon EC2 인스턴스에 연결 섹션을 참조하세요.

또한 EC2 Instance Connect 엔드포인트의 리소스 태그 존재 여부에 따라 사용자에게 인스턴스에 대한 연결을 설정할 권한을 부여할 수 있습니다. 자세한 내용은 IAM 사용자 가이드에서 IAM의 정책 및 권한을 참조하세요.

Linux 인스턴스의 경우 `ec2-instance-connect:SendSSHPublicKey` 작업은 인스턴스에 공개 키를 푸시할 수 있는 권한을 부여합니다. `ec2:osuser` 조건은 퍼블릭 키를 인스턴스에 푸시할 수 있는 OS(운영 체제) 사용자의 이름을 지정합니다. 인스턴스를 시작하는 데 사용한 AMI의 기본 사용자 이름을 사용합니다. 자세한 내용은 EC2 Instance Connect에 대한 IAM 권한 부여 섹션을 참조하세요.

IAM 정책 예제

다음 IAM 정책 예제는 IAM 보안 주체가 지정된 엔드포인트 ID `eice-123456789abcdef`로 식별되는 지정된 EC2 인스턴스 연결 엔드포인트만 사용하여 인스턴스에 연결할 수 있도록 허용합니다. 모든 조건이 충족되는 경우에만 연결이 성공적으로 설정됩니다.

> **Note**
>
> `ec2:Describe*` API 작업은 리소스 수준 권한을 지원하지 않습니다. 따라서 `Resource` 요소에 `*` 와일드카드가 필요합니다.

Linux

이 예에서는 인스턴스에 대한 연결이 -포트 22(SSH)에서 설정되었는지, 인스턴스의 프라이빗 IP 주소가 `10.0.1.0/31` 범위(`10.0.1.0~10.0.1.1` 사이)에 있는지와 `maxTunnelDuration`이 `3600`초 이하인지 평가합니다. `3600`초(1시간) 후에 연결이 해제됩니다.

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Sid": "EC2InstanceConnect",
            "Action": "ec2-instance-connect:OpenTunnel",
            "Effect": "Allow",
            "Resource": "arn:aws:ec2:region:account-id:instance-connect-
endpoint/eice-123456789abcdef",
            "Condition": {
                "NumericEquals": {
                    "ec2-instance-connect:remotePort": "22"
                },
                "IpAddress": {
                    "ec2-instance-connect:privateIpAddress": "10.0.1.0/31"
                },
                "NumericLessThanEquals": {
                    "ec2-instance-connect:maxTunnelDuration": "3600"
                }
```

```
            }
        },
        {
            "Sid": "SSHPublicKey",
            "Effect": "Allow",
            "Action": "ec2-instance-connect:SendSSHPublicKey",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ec2:osuser": "ami-username"
                }
            }
        },
        {
            "Sid": "Describe",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceConnectEndpoints"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Windows

이 예에서는 인스턴스에 대한 연결이 포트 3389(RDP)에서 설정되었는지, 인스턴스의 프라이빗 IP 주소가 10.0.1.0/31 범위(10.0.1.0~10.0.1.1)에 있는지와 maxTunnelDuration이 3600초 이하인지 평가합니다. 3600초(1시간) 후에 연결이 해제됩니다.

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Sid": "EC2InstanceConnect",
            "Action": "ec2-instance-connect:OpenTunnel",
            "Effect": "Allow",
            "Resource": "arn:aws:ec2:region:account-id:instance-connect-
endpoint/eice-123456789abcdef",
            "Condition": {
                "NumericEquals": {
                    "ec2-instance-connect:remotePort": "3389"
                },
```

```
                "IpAddress": {
                    "ec2-instance-connect:privateIpAddress": "10.0.1.0/31"
                },
                "NumericLessThanEquals": {
                    "ec2-instance-connect:maxTunnelDuration": "3600"
                }
            }
        },
        {
            "Sid": "Describe",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceConnectEndpoints"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

특정 IP 주소 범위에서만 연결할 수 있는 권한

다음 IAM 정책 예제는 정책에 지정된 IP 주소 범위 내의 IP 주소에서 연결하는 조건으로 IAM 보안 주체의 인스턴스 연결을 허용합니다. IAM 보안 주체가 192.0.2.0/24에 없는 IP 주소(예: 이 정책의 IP 주소 범위)에서 OpenTunnel을 직접 호출하는 경우 응답은 Access Denied입니다. 자세한 내용은 IAM 사용 설명서에서 aws:SourceIp 섹션을 참조하세요.

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": "ec2-instance-connect:OpenTunnel",
            "Resource": "arn:aws:ec2:region:account-id:instance-connect-
endpoint/eice-123456789abcdef",
            "Condition": {
                "IpAddress": {
                    "aws:SourceIp": "192.0.2.0/24"
                },
                "NumericEquals": {
                    "ec2-instance-connect:remotePort": "22"
                }
            }
```

```
        },
        {
            "Sid": "SSHPublicKey",
            "Effect": "Allow",
            "Action": "ec2-instance-connect:SendSSHPublicKey",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ec2:osuser": "ami-username"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceConnectEndpoints"
            ],
            "Resource": "*"
        }
    ]
}
```

EC2 Instance Connect 엔드포인트 보안 그룹

보안 그룹은 연결된 리소스에 도달하고 나갈 수 있는 트래픽을 제어합니다. 예를 들어, Amazon EC2 인스턴스에 연결된 보안 그룹에서 특별히 허용하지 않는 한, 해당 인스턴스에서 송수신되는 트래픽을 거부합니다.

다음 예제에서는 EC2 Instance Connect 엔드포인트 및 대상 인스턴스에 대한 보안 그룹 규칙을 구성하는 방법을 보여줍니다.

예시

- EC2 Instance Connect 엔드포인트 보안 그룹 규칙
- 대상 인스턴스 보안 그룹 규칙

EC2 Instance Connect 엔드포인트 보안 그룹 규칙

EC2 Instance Connect 엔드포인트의 보안 그룹 규칙을 통해 대상 인스턴스를 대상으로 하는 아웃바운드 트래픽이 엔드포인트를 떠날 수 있습니다. VPC의 IPv4 주소 범위 또는 인스턴스 보안 그룹을 대상으로 지정할 수 있습니다.

엔드포인트로 향하는 트래픽은 EC2 Instance Connect 엔드포인트 서비스에서 시작되며, 엔드포인트 보안 그룹의 인바운드 규칙과 상관없이 허용됩니다. EC2 Instance Connect 엔드포인트를 사용하여 인스턴스에 연결할 수 있는 사용자를 제어하려면 IAM 정책을 사용합니다. 자세한 내용은 인스턴스에 연결하기 위해 EC2 Instance Connect 엔드포인트를 사용할 수 있는 권한 섹션을 참조하세요.

아웃바운드 규칙 예제: 보안 그룹 참조

다음 예제에서는 보안 그룹 참조를 사용합니다. 즉, 대상은 대상 인스턴스와 연결된 보안 그룹입니다. 이 규칙은 엔드포인트에서 이 보안 그룹을 사용하는 모든 인스턴스로의 아웃바운드 트래픽을 허용합니다.

프로토콜	대상	포트 범위	설명
TCP	*#### ## ### ID*	22	인스턴스 보안 그룹과 연결된 모든 인스턴스로의 아웃바운드 SSH 트래픽 허용

아웃바운드 규칙 예제: IPv4 주소 범위

다음 예제에서는 지정된 IPv4 주소 범위로의 아웃바운드 트래픽을 허용합니다. 인스턴스의 IPv4 주소는 서브넷에서 할당되므로 VPC의 IPv4 주소 범위를 사용할 수 있습니다.

프로토콜	대상	포트 범위	설명
TCP	*VPC IPv4 CIDR*	22	VPC로의 아웃바운드 SSH 트래픽 허용

대상 인스턴스 보안 그룹 규칙

대상 인스턴스의 보안 그룹 규칙에서는 EC2 Instance Connect 엔드포인트에서 생성되는 인바운드 트래픽을 허용해야 합니다. 엔드포인트 보안 그룹 또는 IPv4 주소 범위를 소스로 지정할 수 있습니다. IPv4 주소 범위를 지정하는 경우 소스는 클라이언트 IP 보존이 켜졌는지, 꺼졌는지에 따라 달라집니다. 자세한 내용은 고려 사항 섹션을 참조하세요.

보안 그룹은 상태를 저장하므로, 인스턴스 보안 그룹의 아웃바운드 규칙과 상관없이 응답 트래픽이 VPC를 떠날 수 있습니다.

인바운드 규칙 예제: 보안 그룹 참조

다음 예제에서는 보안 그룹 참조를 사용합니다. 즉, 소스는 엔드포인트와 연결된 보안 그룹입니다. 이 규칙에서는 클라이언트 IP 보존이 켜졌는지, 꺼졌는지에 상관없이 엔드포인트에서 이 보안 그룹을 사용하는 모든 인스턴스로의 인바운드 SSH 트래픽을 허용합니다. SSH에 대한 다른 인바운드 보안 그룹 규칙이 없는 경우 인스턴스는 엔드포인트에서의 SSH 트래픽만 수락합니다.

프로토콜	소스	포트 범위	설명
TCP	*##### ## ## ID*	22	엔드포인트 보안 그룹과 연결된 리소스의 인바운드 SSH 트래픽 허용

인바운드 규칙 예제: 클라이언트 IP 보존 꺼짐

다음 예제에서는 지정된 IPv4 주소 범위의 인바운드 SSH 트래픽을 허용합니다. 클라이언트 IP 보존이 꺼져 있으므로 소스 IPv4 주소는 엔드포인트 네트워크 인터페이스의 주소입니다. 엔드포인트 네트워크 인터페이스의 주소는 해당 서브넷에서 할당되므로 VPC의 IPv4 주소 범위를 사용하여 VPC의 모든 인스턴스에 대한 연결을 허용할 수 있습니다.

프로토콜	소스	포트 범위	설명
TCP	*VPC IPv4 CIDR*	22	VPC의 인바운드 SSH 트래픽 허용

인바운드 규칙 예제: 클라이언트 IP 보존 켜짐

다음 예제에서는 지정된 IPv4 주소 범위의 인바운드 SSH 트래픽을 허용합니다. 클라이언트 IP 보존이 켜져 있으므로 소스 IPv4 주소는 클라이언트의 주소입니다.

프로토콜	소스	포트 범위	설명
TCP	*### IPv4 ## ##*	22	지정된 클라이언트 IPv4 주소 범위에서의 인바운드 트래픽 허용

EC2 Instance Connect 엔드포인트 생성

인스턴스에 대한 보안 연결을 허용하는 EC2 Instance Connect 엔드포인트를 생성할 수 있습니다.

EC2 인스턴스 연결 엔드포인트를 생성한 후에는 수정할 수 없습니다. 대신 EC2 Instance Connect 엔드포인트를 삭제하고 필요한 설정으로 새 엔드포인트를 생성해야 합니다.

사전 조건

EC2 Instance Connect 엔드포인트를 생성하려면 필요한 IAM 권한이 있어야 합니다. 자세한 내용은 EC2 Instance Connect 엔드포인트 생성, 설명 및 삭제할 수 있는 권한 섹션을 참조하세요.

공유 서브넷

공유된의 서브넷에 EC2 Instance Connect 엔드포인트를 생성합니다. VPC 소유자가 공유된 서브넷에서 생성한 EC2 Instance Connect 엔드포인트는 사용할 수 없습니다.

Console

EC2 Instance Connect 엔드포인트 생성

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 엔드포인트를 선택합니다.
3. 엔드포인트 생성을 선택하고 다음과 같이 엔드포인트 설정을 지정합니다.
   a. (선택 사항) 이름 태그에 엔드포인트의 이름을 입력합니다.
   b. 유형에서 EC2 Instance Connect 엔드포인트를 선택합니다.
   c. VPC의 네트워크 설정에서 대상 인스턴스가 있는 VPC를 선택합니다.
   d. (선택 사항) 클라이언트 IP 주소를 보존하려면 추가 설정을 확장하고 클라이언트 IP 보존 확인란을 선택합니다. 그렇지 않으면 기본적으로 엔드포인트 네트워크 인터페이스를 클라이언트 IP 주소로 사용합니다.
   e. (선택 사항) 보안 그룹의 경우 엔드포인트와 연결할 보안 그룹을 선택합니다. 그렇지 않으면 기본적으로 VPC에 대한 기본 보안 그룹이 사용됩니다. 자세한 내용은 EC2 Instance Connect 엔드포인트 보안 그룹 섹션을 참조하세요.
   f. 서브넷에서 엔드포인트를 생성할 서브넷을 선택합니다.
   g. (선택 사항) 태그를 추가하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
4. 설정을 검토하고 엔드포인트 생성을 선택합니다.

   엔드포인트의 초기 상태는 보류 중입니다. 이 엔드포인트를 사용하여 인스턴스에 연결하기 전에 엔드포인트 상태가 사용 가능이 될 때까지 기다려야 합니다. 몇 분 정도 소요될 수 있습니다.
5. 엔드포인트를 사용하여 인스턴스에 연결하려면 인스턴스에 연결 섹션을 참조하세요.

AWS CLI

EC2 Instance Connect 엔드포인트 생성

create-instance-connect-endpoint 명령을 사용합니다.

```
aws ec2 create-instance-connect-endpoint --subnet-id subnet-0123456789example
```

출력의 예시는 다음과 같습니다.

```
{
        "OwnerId": "111111111111",
        "InstanceConnectEndpointId": "eice-0123456789example",
        "InstanceConnectEndpointArn": "arn:aws:ec2:us-east-1:111111111111:instance-
connect-endpoint/eice-0123456789example",
        "State": "create-complete",
        "StateMessage": "",
        "DnsName": "eice-0123456789example.0123abcd.ec2-instance-connect-
endpoint.us-east-1.amazonaws.com",
        "FipsDnsName": "eice-0123456789example.0123abcd.fips.ec2-instance-connect-
endpoint.us-east-1.amazonaws.com",
        "NetworkInterfaceIds": [
            "eni-0123abcd"
        ],
        "VpcId": "vpc-0123abcd",
        "AvailabilityZone": "us-east-1a",
        "CreatedAt": "2023-04-07T15:43:53.000Z",
        "SubnetId": "subnet-0123abcd",
        "PreserveClientIp": false,
        "SecurityGroupIds": [
            "sg-0123abcd"
        ],
        "Tags": []
}
```

생성 상태를 모니터링하는 방법

State 필드의 초기 값은 create-in-progress입니다. 이 엔드포인트를 사용하여 인스턴스에 연결하기 전에 상태가 create-complete이(가) 될 때까지 기다리세요. describe-instance-connect-endpoints 명령을 사용하여 EC2 Instance Connect 엔드포인트의 상태를 모니터링합니다. --query 파라미터는 결과를 State 필드로 필터링합니다.

```
aws ec2 describe-instance-connect-endpoints --instance-connect-endpoint-
ids eice-0123456789example --query InstanceConnectEndpoints[*].State --output text
```

출력의 예시는 다음과 같습니다.

```
create-complete
```

PowerShell

EC2 Instance Connect 엔드포인트를 생성하려면 다음을 수행하세요.

New-EC2InstanceConnectEndpoint cmdlet을 사용하세요.

```
New-EC2InstanceConnectEndpoint -SubnetId subnet-0123456789example
```

출력의 예시는 다음과 같습니다.

```
OwnerId                     : 111111111111
InstanceConnectEndpointId   : eice-0123456789example
InstanceConnectEndpointArn  : arn:aws:ec2:us-east-1:111111111111:instance-connect-
endpoint/eice-0123456789example
State                       : create-complete
StateMessage                :
DnsName                     : eice-0123456789example.0123abcd.ec2-instance-connect-
endpoint.us-east-1.amazonaws.com
FipsDnsName                 : eice-0123456789example.0123abcd.fips.ec2-instance-
connect-endpoint.us-east-1.amazonaws.com
NetworkInterfaceIds         : {eni-0123abcd}
VpcId                       : vpc-0123abcd
AvailabilityZone            : us-east-1a
CreatedAt                   : 4/7/2023 3:43:53 PM
SubnetId                    : subnet-0123abcd
PreserveClientIp            : False
SecurityGroupIds            : {sg-0123abcd}
Tags                        : {}
```

생성 상태를 모니터링하는 방법

State 필드의 초기 값은 create-in-progress입니다. 이 엔드포인트를 사용하여 인스턴스에 연결하기 전에 상태가 create-complete이(가) 될 때까지 기다리세요. Get-

EC2InstanceConnectEndpoint cmdlet을 사용하여 EC2 인스턴스 연결 엔드포인트의 상태를 모니터링합니다. `.State.Value`는 결과를 `State` 필드로 필터링합니다.

```
(Get-EC2InstanceConnectEndpoint -InstanceConnectEndpointId
 "eice-0123456789example").State.Value
```

출력의 예시는 다음과 같습니다.

```
create-complete
```

EC2 인스턴스 연결 엔드포인트를 사용하여 Amazon EC2 인스턴스에 연결

EC2 인스턴스 연결 엔드포인트를 사용하여 SSH 또는 RDP를 지원하는 Amazon EC2 인스턴스에 연결할 수 있습니다.

사전 조건

- EC2 Instance Connect 엔드포인트에 연결하려면 필요한 IAM 권한이 있어야 합니다. 자세한 내용은 인스턴스에 연결하기 위해 EC2 Instance Connect 엔드포인트를 사용할 수 있는 권한 섹션을 참조하세요.
- EC2 Instance Connect 엔드포인트가 사용 가능(콘솔) 또는 `create-complete`(AWS CLI) 상태여야 합니다. VPC에 대한 EC2 Instance Connect 엔드포인트가 없는 경우 새로 생성할 수 있습니다. 자세한 내용은 EC2 Instance Connect 엔드포인트 생성 섹션을 참조하세요.
- 인스턴스에 IPv4 주소(프라이빗 또는 퍼블릭)가 있어야 합니다. EC2 Instance Connect 엔드포인트에서는 IPv6 주소를 사용하는 연결을 지원하지 않습니다.
- (Linux 인스턴스) Amazon EC2 콘솔을 사용하여 인스턴스에 연결하거나, CLI를 사용하여 연결하고 EC2 Instance Connect에서 임시 키를 처리하도록 하려면 인스턴스에 EC2 Instance Connect가 설치되어 있어야 합니다. 자세한 내용은 EC2 Instance Connect 설치 섹션을 참조하세요.
- 인스턴스의 보안 그룹이 EC2 Instance Connect 엔드포인트에서의 인바운드 SSH 트래픽을 허용하는지 확인합니다. 자세한 내용은 대상 인스턴스 보안 그룹 규칙 섹션을 참조하세요.

연결 옵션

- Amazon EC2 콘솔을 사용하여 Linux 인스턴스에 연결
- SSH를 사용하여 Linux 인스턴스에 연결
- AWS CLI를 사용하여 Linux 인스턴스에 연결

- RDP를 사용하여 Windows 인스턴스에 연결
- 문제 해결

Amazon EC2 콘솔을 사용하여 Linux 인스턴스에 연결

다음과 같이 Amazon EC2 콘솔(브라우저 기반 클라이언트)을 사용하여 인스턴스에 연결할 수 있습니다.

Amazon EC2 콘솔을 사용하여 인스턴스에 연결

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 탭을 선택합니다.
5. 연결 유형에서 EC2 Instance Connect 엔드포인트를 사용하여 연결을 선택합니다.
6. EC2 Instance Connect 엔드포인트의 경우 EC2 Instance Connect 엔드포인트 ID를 선택합니다.
7. 인스턴스를 시작하는 데 사용한 AMI가 `ec2-user` 이외의 사용자 이름을 사용하는 경우 사용자 이름에 올바른 사용자 이름을 입력합니다.
8. 최대 터널 지속 시간(초)에 SSH 연결의 허용되는 최대 지속 시간을 입력합니다.

   지속 시간은 IAM 정책에 지정된 `maxTunnelDuration` 조건을 준수해야 합니다. IAM 정책에 대한 액세스 권한이 없는 경우 관리자에게 문의하세요.
9. 연결을 선택합니다. 그러면 인스턴스의 터미널 창이 열립니다.

SSH를 사용하여 Linux 인스턴스에 연결

SSH를 사용하여 Linux 인스턴스에 연결하고, `open-tunnel` 명령을 사용하여 프라이빗 터널을 설정할 수 있습니다. 단일 연결 또는 다중 연결 모드에서 `open-tunnel`을 사용할 수 있습니다.

SSH로 인스턴트에 연결하기 위한 AWS CLI 사용에 대한 자세한 내용은 AWS CLI를 사용하여 연결의 내용을 참조하세요.

다음 예제에서는 OpenSSH를 사용합니다. 프록시 모드를 지원하는 다른 SSH 클라이언트를 사용할 수 있습니다.

단일 연결

SSH 및 **`open-tunnel`** 명령을 사용하여 인스턴스에 단일 연결만 허용

ssh 및 open-tunnel AWS CLI 명령을 다음과 같이 실행합니다. -o 프록시 명령은 인스턴스에 대한 프라이빗 터널을 생성하는 open-tunnel 명령을 포함합니다.

```
ssh -i my-key-pair.pem ec2-user@i-1234567890abcdef0 \
    -o ProxyCommand='aws ec2-instance-connect open-tunnel --instance-
id i-1234567890abcdef0'
```

여기에서:

- `-i` - 인스턴스를 시작하는 데 사용되었던 키 페어를 지정합니다.
- *ec2-user*@*i-1234567890abcdef0* - 인스턴스를 시작하는 데 사용되었던 AMI의 사용자 이름과 인스턴스 ID를 지정합니다.
- `--instance-id` - 연결할 인스턴스의 ID를 지정합니다. 또는 %h를 지정하여 사용자로부터 인스턴스 ID를 추출할 수도 있습니다.

다중 연결

인스턴스에 다중 연결을 허용하려면 먼저 open-tunnel AWS CLI 명령을 실행하여 새 TCP 연결 수신을 시작하고, ssh를 사용하여 새 TCP 연결과 인스턴스에 대한 프라이빗 터널을 생성합니다.

SSH 및 **`open-tunnel`** 명령을 사용하여 인스턴스에 다중 연결 허용

1. 다음 명령을 실행하여 로컬 시스템의 지정된 포트에서 새 TCP 연결 수신을 시작합니다.

   ```
   aws ec2-instance-connect open-tunnel \
       --instance-id i-1234567890abcdef0 \
       --local-port 8888
   ```

   예상 결과

   ```
   Listening for connections on port 8888.
   ```

2. 새 터미널 창에서 다음 ssh 명령을 실행하여 인스턴스에 대한 새 TCP 연결 및 프라이빗 터널을 생성합니다.

   ```
   ssh -i my-key-pair.pem ec2-user@localhost -p 8888
   ```

   예상 출력 - 첫 번째 터미널 창에 다음이 표시됩니다.

```
[1] Accepted new tcp connection, opening websocket tunnel.
```

다음이 표시될 수도 있습니다.

```
[1] Closing tcp connection.
```

AWS CLI를 사용하여 Linux 인스턴스에 연결

인스턴스 ID만 아는 경우 ec2-instance-connect AWS CLI 명령을 사용하여 SSH 클라이언트로 인스턴트에 연결할 수 있습니다. ec2-instance-connect 명령 사용에 대한 자세한 내용은 AWS CLI를 사용하여 연결 섹션을 참조하세요.

사전 조건

AWS CLI 버전 2를 설치하고 자격 증명을 사용하여 구성합니다. 자세한 내용은 AWS Command Line Interface 사용 설명서의 Install or update to the latest version of the AWS CLI 및 Configure the AWS CLI를 참조하세요. 또는 사전 인증된 쉘에서 AWS CloudShell을 열고 AWS CLI 명령을 실행합니다.

인스턴스 ID 및 EC2 Instance Connect 엔드포인트를 사용하여 인스턴스에 연결

인스턴스 ID만 아는 경우 ec2-instance-connect CLI 명령을 사용하고 `ssh` 명령, 인스턴스 ID, `eice` 값이 있는 `--connection-type` 파라미터를 지정합니다.

```
aws ec2-instance-connect ssh --instance-id i-1234567890example --os-user ec2-user --
connection-type eice
```

> **Tip**
>
> 이 명령을 사용할 때 오류가 발생하면 AWS CLI 버전 2를 사용하고 있는지 확인합니다. `ssh` 파라미터는 AWS CLI 버전 2에서만 사용할 수 있습니다. 자세한 내용은 AWS Command Line Interface 사용 설명서의 AWS CLI 버전 2 정보를 참조하세요.

RDP를 사용하여 Windows 인스턴스에 연결

EC2 Instance Connect 엔드포인트를 통한 원격 데스크톱 프로토콜(RDP)을 사용하여 퍼블릭 IPv4 주소 또는 퍼블릭 DNS 이름 없이 Windows 인스턴스에 연결할 수 있습니다.

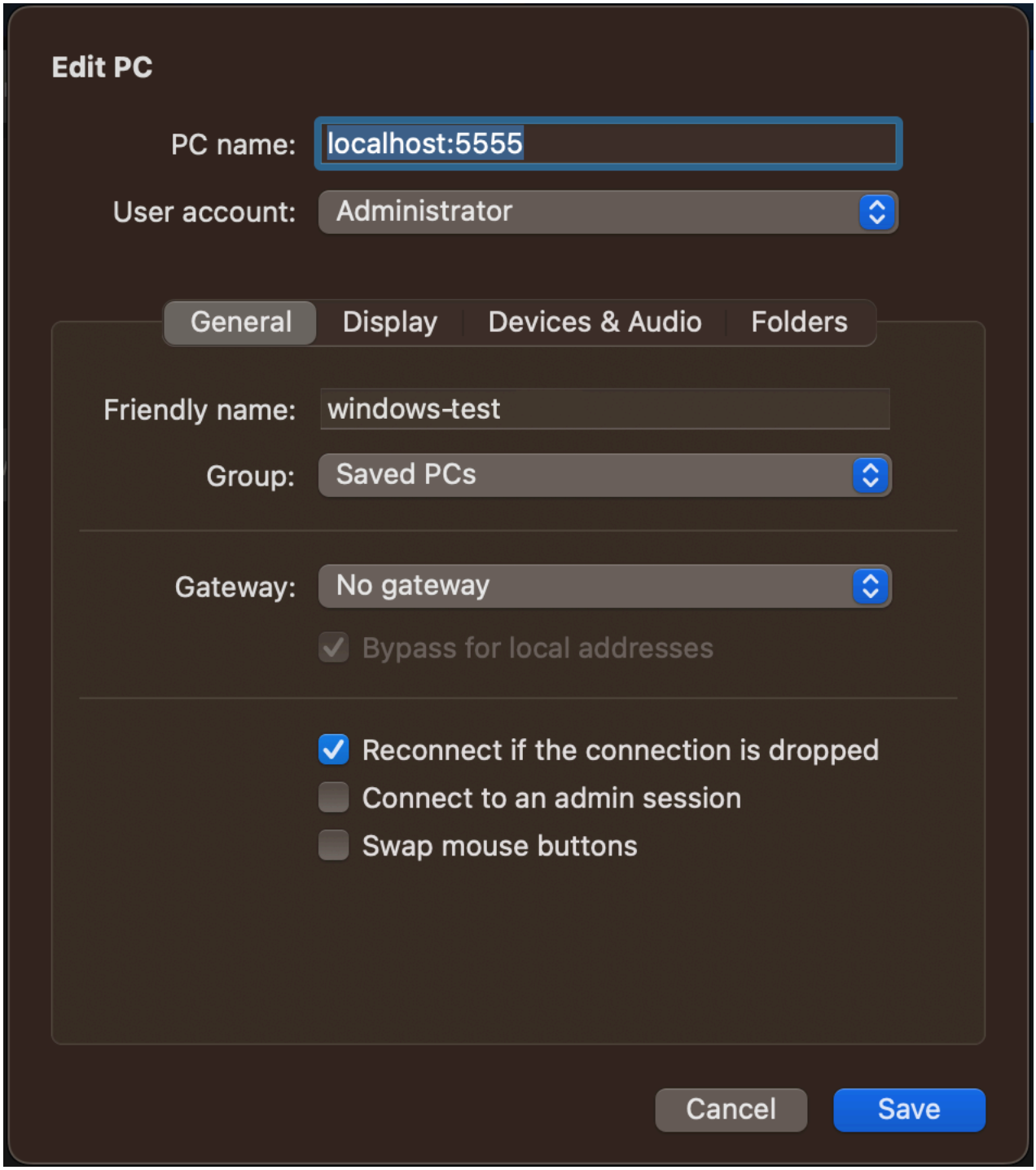
RDP 클라이언트로 Windows 인스턴스 연결

1. RDP를 사용하여 Windows 인스턴스에 연결의 1~8단계를 완료합니다. 8단계에서 RDP 데스크톱 파일을 다운로드한 이후 연결할 수 없음 메시지가 표시되는데, 이는 인스턴스에 퍼블릭 IP 주소가 없기 때문일 수 있습니다.
2. 다음 명령을 실행하여 인스턴스가 위치한 VPC에 대한 프라이빗 터널을 설정합니다. RDP가 기본적으로 포트 3389를 사용하기 때문에 `--remote-port`는 3389여야 합니다.

```
aws ec2-instance-connect open-tunnel \
    --instance-id i-1234567890abcdef0 \
    --remote-port 3389 \
    --local-port any-port
```

3. 다운로드 폴더에서 다운로드한 RDP 데스크톱 파일을 찾아 RDP 클라이언트 창으로 끌어다 놓습니다.
4. RDP 데스크톱 파일을 마우스 오른쪽 버튼으로 클릭하고 편집을 선택합니다.
5. PC 편집 창의 PC 이름(연결할 인스턴스)에 `localhost:local-port`를 입력합니다. 여기에서 `local-port`는 2단계에서 지정한 것과 동일한 값을 사용합니다. 입력 후 저장을 선택합니다.

   PC 편집 창의 다음 스크린샷은 Mac의 Microsoft Remote Desktop 스크린샷입니다. Windows 클라이언트를 사용하는 경우 창이 다를 수 있습니다.



6. RDP 클라이언트에서 방금 구성한 PC를 마우스 오른쪽 버튼으로 클릭하고 연결을 선택하여 인스턴스에 연결합니다.
7. 프롬프트에서 관리자 계정의 해독된 암호를 입력합니다.

문제 해결

다음 정보를 사용하면 EC2 Instance Connect 엔드포인트로 인스턴스를 연결할 때 발생할 수 있는 문제를 진단하고 수정하는 데 도움이 됩니다.

인스턴스에 연결할 수 없음

다음은 인스턴스에 연결할 수 없는 일반적인 이유입니다.

- 보안 그룹 - EC2 Instance Connect 엔드포인트와 인스턴스에 할당된 보안 그룹을 확인합니다. 필수 보안 그룹 규칙에 대한 자세한 내용은 EC2 Instance Connect 엔드포인트 보안 그룹 섹션을 참조하세요.
- 인스턴스 상태 - 인스턴스가 `running` 상태인지 확인합니다.
- 키 페어 - 연결하는 데 사용하는 명령에 프라이빗 키가 필요한 경우 인스턴스에 퍼블릭 키가 있고 해당 프라이빗 키가 있는지 확인합니다.
- IAM 권한 - 필요한 IAM 권한이 있는지 확인합니다. 자세한 내용은 EC2 Instance Connect 엔드포인트를 사용할 수 있는 권한 부여 섹션을 참조하세요.

Linux 인스턴스의 문제 해결 팁에 대한 자세한 내용은 Amazon EC2 Linux 인스턴스 연결 문제 해결 섹션을 참조하세요. Windows 인스턴스의 문제 해결 팁에 대한 자세한 내용은 the section called “Windows 인스턴스 RDP 문제” 섹션을 참조하세요.

ErrorCode: AccessDeniedException

`AccessDeniedException` 오류가 발생하고, `maxTunnelDuration` 조건이 IAM 정책에 지정된 경우 인스턴스에 연결할 때 `--max-tunnel-duration` 파라미터를 지정해야 합니다. 이 파라미터에 대한 자세한 내용은 AWS CLI 명령 참조의 open-tunnel을 참조하세요.

EC2 Instance Connect 엔드포인트를 통해 설정된 연결 기록

EC2 Instance Connect 엔드포인트를 통해 설정된 리소스 작업 및 감사 연결을 AWS CloudTrail 로그로 기록할 수 있습니다.

Amazon EC2에서 AWS CloudTrail을 사용하는 방법에 대한 자세한 내용은 AWS CloudTrail을 사용하여 Amazon EC2 API 호출 로깅 섹션을 참조하세요.

AWS CloudTrail로 EC2 Instance Connect 엔드포인트 API 호출 기록

EC2 Instance Connect 엔드포인트 리소스 작업은 CloudTrail에 관리 이벤트로 기록됩니다. 다음 API 호출이 이루어지면 활동이 이벤트 기록에 CloudTrail 이벤트로 기록됩니다.

- `CreateInstanceConnectEndpoint`
- `DescribeInstanceConnectEndpoints`
- `DeleteInstanceConnectEndpoint`

AWS 계정에서 최신 이벤트를 확인, 검색 및 다운로드할 수 있습니다. 자세한 내용은 AWS CloudTrail 사용 설명서에서 CloudTrail 이벤트 기록을 사용하여 이벤트 보기를 참조하세요.

AWS CloudTrail을 사용하여 EC2 Instance Connect 엔드포인트를 사용하여 인스턴스에 연결하는 사용자 감사

EC2 Instance Connect 엔드포인트를 통한 인스턴스 연결 시도는 이벤트 기록의 CloudTrail에 기록됩니다. EC2 Instance Connect 엔드포인트를 통해 인스턴스 연결이 시작되면 연결은 `OpenTunnel`의 `eventName`을 사용하여 CloudTrail 관리 이벤트로 기록됩니다.

CloudTrail 이벤트를 대상으로 라우팅하는 Amazon EventBridge 규칙을 생성할 수 있습니다. 자세한 내용은 Amazon EventBridge 사용 설명서를 참조하세요.

다음은 CloudTrail에 기록된 `OpenTunnel` 관리 이벤트 예제입니다.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "ABCDEFGONGNOMOOCB6XYTQEXAMPLE",
        "arn": "arn:aws:iam::1234567890120:user/IAM-friendly-name",
        "accountId": "123456789012",
        "accessKeyId": "ABCDEFGUKZHNAW4OSN2AEXAMPLE",
        "userName": "IAM-friendly-name"
    },
    "eventTime": "2023-04-11T23:50:40Z",
    "eventSource": "ec2-instance-connect.amazonaws.com",
    "eventName": "OpenTunnel",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "1.2.3.4",
    "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
    "requestParameters": {
        "instanceConnectEndpointId": "eici-0123456789EXAMPLE",
        "maxTunnelDuration": "3600",
        "remotePort": "22",
        "privateIpAddress": "10.0.1.1"
    },
```

```
    "responseElements": null,
    "requestID": "98deb2c6-3b3a-437c-a680-03c4207b6650",
    "eventID": "bbba272c-8777-43ad-91f6-c4ab1c7f96fd",
    "readOnly": false,
    "resources": [{
        "accountId": "123456789012",
        "type": "AWS::EC2::InstanceConnectEndpoint",
        "ARN": "arn:aws:ec2:us-east-1:123456789012:instance-connect-endpoint/
eici-0123456789EXAMPLE"
    }],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```

EC2 Instance Connect 엔드포인트 삭제

EC2 Instance Connect 엔드포인트 관련 작업을 마치면 해당 엔드포인트를 삭제할 수 있습니다.

EC2 Instance Connect 엔드포인트를 생성하려면 필요한 IAM 권한이 있어야 합니다. 자세한 내용은 EC2 Instance Connect 엔드포인트 생성, 설명 및 삭제할 수 있는 권한 섹션을 참조하세요.

콘솔을 사용하여 EC2 Instance Connect 엔드포인트를 삭제하면 삭제 중 상태가 됩니다. 삭제에 성공하면 삭제된 엔드포인트는 더 이상 표시되지 않습니다. 삭제에 실패하면 `delete-failed` 상태가 되고 상태 메시지에서 실패 이유를 제공합니다.

AWS CLI를 사용하여 EC2 Instance Connect 엔드포인트를 삭제하면 `delete-in-progress` 상태가 됩니다. 삭제에 성공하면 `delete-complete` 상태가 됩니다. 삭제에 실패하면 `delete-failed` 상태가 되고 `StateMessage`에서 실패 이유를 제공합니다.

Console

EC2 Instance Connect 엔드포인트 삭제

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 엔드포인트를 선택합니다.
3. 엔드포인트를 선택합니다.
4. 작업(Actions), VPC 엔드포인트 삭제(Delete VPC endpoints)를 차례로 선택합니다.
5. 확인 메시지가 표시되면 **`delete`**를 입력합니다.
6. 삭제를 선택합니다.

AWS CLI

EC2 Instance Connect 엔드포인트 삭제

delete-instance-connect-endpoint 명령을 사용하고 삭제할 EC2 Instance Connect 엔드포인트의 ID를 지정합니다.

```
aws ec2 delete-instance-connect-endpoint --instance-connect-endpoint-
id eice-03f5e49b83924bbc7
```

출력의 예시는 다음과 같습니다.

```
{
    "InstanceConnectEndpoint": {
        "OwnerId": "111111111111",
        "InstanceConnectEndpointId": "eice-0123456789example",
        "InstanceConnectEndpointArn": "arn:aws:ec2:us-east-1:111111111111:instance-
connect-endpoint/eice-0123456789example",
        "State": "delete-in-progress",
        "StateMessage": "",
        "NetworkInterfaceIds": [],
        "VpcId": "vpc-0123abcd",
        "AvailabilityZone": "us-east-1d",
        "CreatedAt": "2023-02-07T12:05:37+00:00",
        "SubnetId": "subnet-0123abcd"
    }
}
```

PowerShell

EC2 Instance Connect 엔드포인트 삭제

Remove-EC2InstanceConnectEndpoint cmdlet을 사용하고 삭제할 EC2 Instance Connect 엔드포인트의 ID를 지정합니다.

```
Remove-EC2InstanceConnectEndpoint -InstanceConnectEndpointId eice-03f5e49b83924bbc7
```

출력의 예시는 다음과 같습니다.

```
@{
    InstanceConnectEndpoint = @{
        OwnerId = "111111111111"
```

```
        InstanceConnectEndpointId = "eice-0123456789example"
        InstanceConnectEndpointArn = "arn:aws:ec2:us-east-1:111111111111:instance-
connect-endpoint/eice-0123456789example"
        State = "delete-in-progress"
        StateMessage = ""
        NetworkInterfaceIds = @()
        VpcId = "vpc-0123abcd"
        AvailabilityZone = "us-east-1d"
        CreatedAt = "2023-02-07T12:05:37+00:00"
        SubnetId = "subnet-0123abcd"
    }
}
```

EC2 Instance Connect 엔드포인트에 대한 서비스 연결 역할

Amazon EC2는 AWS Identity and Access Management(IAM) 서비스 연결 역할을 사용합니다. 서비스 연결 역할은 Amazon EC2에 직접 연결된 고유한 유형의 IAM 역할입니다. 서비스 연결 역할은 Amazon EC2에서 사전 정의되며, Amazon EC2에서 다른 AWS 서비스를 자동으로 직접 호출할 수 있도록 필요한 모든 권한을 포함합니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할을 참조하세요.

EC2 Instance Connect 엔드포인트에 대한 서비스 연결 역할 권한

Amazon EC2는 AWSServiceRoleforec2InstanceConnect를 사용하여 EC2 Instance Connect 엔드포인트에 필요한 네트워크 인터페이스를 계정에서 생성하고 관리합니다.

AWSServiceRoleForEC2InstanceConnect 서비스 연결 역할은 역할을 수임하기 위해 다음 서비스를 신뢰합니다.

- `ec2-instance-connect.amazonaws.com`

AWSServiceRoleForEC2InstanceConnect 서비스 연결 역할에서는 관리형 정책 Ec2InstanceConnectEndpoint를 사용합니다. 이 정책의 권한을 보려면 AWS 관리형 정책 참조의 Ec2InstanceConnectEndpoint를 참조하세요.

IAM 엔터티(사용자, 그룹, 역할 등)가 서비스 링크 역할을 생성하고 편집하거나 삭제할 수 있도록 권한을 구성할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 권한을 참조하세요.

EC2 Instance Connect 엔드포인트에 대한 서비스 연결 역할 생성

서비스 연동 역할을 수동으로 생성하지 않아도 됩니다. EC2 Instance Connect 엔드포인트를 생성할 때 Amazon EC2에서 서비스 연결 역할을 자동으로 생성합니다.

EC2 Instance Connect 엔드포인트에 대한 서비스 연결 역할 편집

EC2 Instance Connect 엔드포인트에서는 AWSServiceRoleForEC2Instance 서비스 연결 역할 편집을 허용하지 않습니다.

EC2 Instance Connect 엔드포인트에 대한 서비스 연결 역할 삭제

EC2 Instance Connect 엔드포인트를 더 이상 사용할 필요가 없는 경우 AWSServiceRoleForEC2InstanceConnect 서비스 연결 역할을 삭제하는 것이 좋습니다.

서비스 연결 역할을 삭제하기 전에 모든 EC2 Instance Connect 엔드포인트 리소스를 삭제해야 합니다.

서비스 연결 역할을 삭제하려면 IAM 사용 설명서의 서비스 연결 역할 삭제를 참조하세요.

EC2 Instance Connect 엔드포인트의 할당량

AWS 계정에는 각 AWS 서비스에 대한 기본 할당량(이전에는 제한이라고 함)이 있습니다. 다르게 표시되지 않는 한, 지역별로 각 할당량이 적용됩니다.

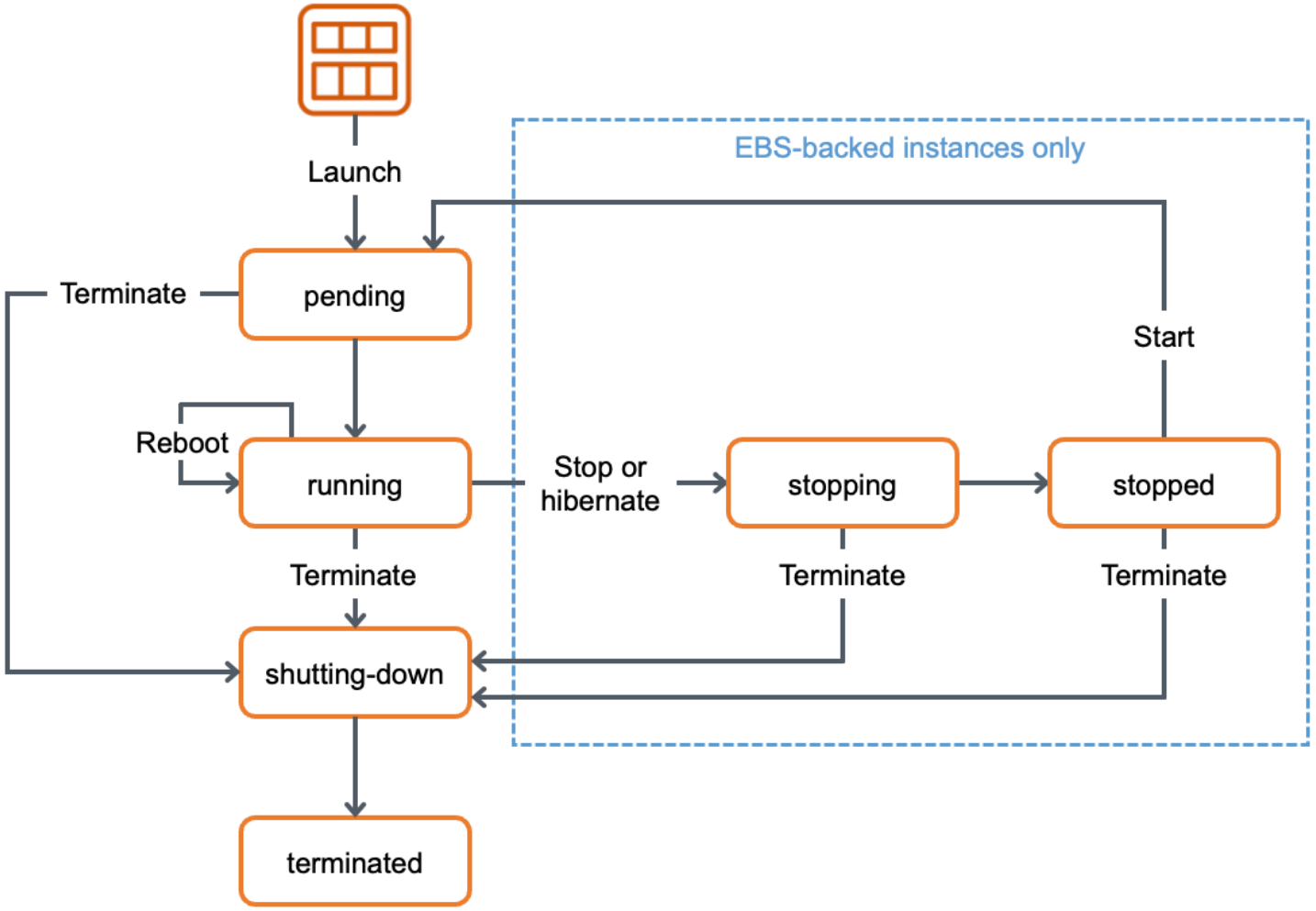
AWS 계정에는 EC2 Instance Connect 엔드포인트와 관련된 다음과 같은 할당량이 있습니다.

명칭	기본값	조정 가능
AWS 리전별 AWS 계정당 최대 EC2 Instance Connect 엔드포인트 수	5	아니요
VPC당 최대 EC2 Instance Connect 엔드포인트 수	1	아니요
서브넷당 최대 EC2 Instance Connect 엔드포인트 수	1	아니요
EC2 인스턴스 연결 엔드포인트당 최대 동시 연결 수	20	아니요

Amazon EC2 인스턴스 상태 변경

Amazon EC2 인스턴스는 시작한 순간부터 종료될 때까지 다양한 상태로 전환됩니다.

다음 그림은 인스턴스 상태 간 전환을 나타냅니다.



인스턴스 상태가 변경될 때 알림을 받을 수 있습니다. 자세한 내용은 the section called “상태 변경 이벤트” 섹션을 참조하세요.

인스턴스 상태별 청구

다음 테이블에서는 각 인스턴스 상태에 대한 간략한 설명과 인스턴스 사용량에 대한 비용 청구 여부를 제공합니다. Amazon EBS 볼륨 및 탄력적 IP 주소와 같은 일부 AWS 리소스는 인스턴스의 상태와 상관없이 요금이 발생합니다. 자세한 내용은 AWS Billing 사용 설명서에서 예기치 않은 요금 이해를 참조하세요.

인스턴스 상태	설명	인스턴스 사용 요금
pending	인스턴스는 running 상태로 될 준비를 하고 있습니다. 인스턴스는 시작되거나 stopped 상태 이후에 시작되면 pending 상태로 들어갑니다.	미청구
running	인스턴스를 실행하고 사용할 준비가 되었습니다.	청구
stopping	인스턴스를 중지할 준비를 하고 있습니다.	미청구 **Note** 인스턴스를 최대 절전 모드로 전환하면 인스턴스가 stopping 상태인 동안 요금이 청구됩니다.
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 Amazon EC2용 예약 인스턴스 개요 섹션을 참조하세요.

보류 중인 인스턴스

인스턴스를 시작하면 인스턴스가 pending 상태로 전환됩니다. 시작 시 지정한 인스턴스 유형에 따라 인스턴스에 사용되는 호스트 컴퓨터의 하드웨어가 결정됩니다. 인스턴스는 시작 시 사용자가 지정한 Amazon Machine Image(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.

중지된 인스턴스

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. 인스턴스가 stopped인 경우에는 인스턴스에 대한 사용 또는 데이터 전송 요금이 청구되지 않습니다. 모든 Amazon EBS 볼륨의 스토리지에는 요금이 부과됩니다. 인스턴스가 stopped상태인 경우 인스턴스 유형을 비롯하여 인스턴스의 특정 속성을 수정할 수 있습니다.

인스턴스를 시작하면 인스턴스가 pending 상태로 전환되며, 인스턴스가 새 호스트 컴퓨터로 이동됩니다(경우에 따라 현재 호스트에 남아 있음). 인스턴스를 중지했다가 시작하면 이전 호스트 컴퓨터에 연결된 인스턴스 스토어 볼륨에 있는 데이터가 모두 손실됩니다.

인스턴스에서 프라이빗 IPv4 주소가 유지됩니다. 즉, 프라이빗 IPv4 주소 또는 네트워크 인터페이스와 연결된 탄력적 IP 주소와 인스턴스가 연결된 상태로 유지된다는 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.

인스턴스의 중지 및 시작에 대한 자세한 내용은 Amazon EC2 인스턴스 중지 및 시작 단원을 참조하세요.

최대 절전 모드로 전환된 인스턴스

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 중지한 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, 인스턴스가 새 호스트 컴퓨터로 이동됩니다(경우에 따라 현재 호스트에 남아 있음).

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 Amazon EC2 인스턴스를 최대 절전 모드로 전환 섹션을 참조하세요.

재부팅 중인 인스턴스

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 시간이 새로 시작되지 않으며, 최소 1분 요금 부과 없이 초 단위 결제가 계속됩니다.

자세한 내용은 Amazon EC2 인스턴스 재부팅 섹션을 참조하세요.

종료된 인스턴스

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. 종료한 인스턴스에 연결하거나 복구할 수 없습니다.

각각의 Amazon EBS 기반 인스턴스는 `InstanceInitiatedShutdownBehavior` 속성을 지원하는데, 이러한 속성은 인스턴스 자체 내에서 종료를 시작할 때 인스턴스가 중지되는지 또는 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 Amazon EC2 인스턴스 종료 섹션을 참조하세요.

인스턴스 상태 간 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.	인스턴스가 새 호스트 컴퓨터로 이동됩니다(경우에 따라 현재 호스트에 남아 있음).	인스턴스가 새 호스트 컴퓨터로 이동됩니다(경우에 따라 현재 호스트에 남아 있음).	없음
프라이빗 IPv4 주소	인스턴스가 관련 프라이빗 IPv4 주소를 유지합니다.	인스턴스가 관련 프라이빗 IPv4 주소를 유지합니다.	인스턴스가 관련 프라이빗 IPv4 주소를 유지합니다.	없음
퍼블릭 IPv4 주소	인스턴스가 퍼블릭 IPv4 주소를 유지합니다.	인스턴스에 탄력적 IP 주소와 연결된 보조 네트워크 인터페이스 또는 보조 프라이빗 IPv4 주소가 없는 경우 인스턴스는	인스턴스에 탄력적 IP 주소와 연결된 보조 네트워크 인터페이스 또는 보조 프라이빗 IPv4 주소가 없는 경우 인스턴스는	없음

특성	재부팅	중지/시작(Amazon EBS 기반 인스턴스에만 해당)	최대 절전 모드(Amazon EBS 지원 인스턴스에만 해당)	Terminate
		새 퍼블릭 IPv4 주소를 가져옵니다.	새 퍼블릭 IPv4 주소를 가져옵니다.	
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	인스턴스가 관련 IPv6 주소를 유지합니다.	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.

특성	재부팅	중지/시작(Amazon EBS 기반 인스턴스에만 해당)	최대 절전 모드(Amazon EBS 지원 인스턴스에만 해당)	Terminate
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 기간이 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 기간이 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 시작하는 종료 동작 변경 섹션을 참조하세요.

Amazon EC2 인스턴스 중지 및 시작

Amazon EBS 볼륨을 루트 디바이스로 사용하는 인스턴스를 중지했다가 다시 시작할 수 있습니다. 인스턴스를 중지하면 인스턴스가 종료됩니다. 인스턴스를 시작하면 일반적으로 새 기본 호스트 컴퓨터로 마이그레이션되고 새 퍼블릭 IPv4 주소가 할당됩니다.

인스턴스 중지는 사용자가 시작하거나(인스턴스를 수동으로 중지하는 경우), AWS에 의해 시작될 수 있습니다(AWS가 인스턴스의 기본 호스트에서 복구할 수 없는 장애를 감지한 경우 예약된 중지 이벤트에 대한 응답으로).

사용자가 시작한 중지의 경우 인스턴스에서 운영 체제 중지 명령을 실행하는 대신 Amazon EC2 콘솔, CLI 또는 API를 사용하는 것이 좋습니다. Amazon EC2를 사용하는 경우 인스턴스가 몇 분 안에 완전히 종료되지 않으면 Amazon EC2가 하드 종료를 수행합니다. 또한 AWS CloudTrail에서는 인스턴스가 중지된 시점의 API 레코드를 생성합니다.

이 주제에서는 사용자가 시작하는 중지를 수행하는 방법을 설명합니다. AWS에서 수행하는 중지에 대한 자세한 내용은 중지 또는 사용 중지하도록 Amazon EC2 인스턴스 관리 섹션을 참조하세요.

인스턴스를 중지해도 해당 인스턴스는 삭제되지 않습니다. 더 이상 필요 없는 인스턴스는 종료할 수 있습니다. 자세한 내용은 Amazon EC2 인스턴스 종료 섹션을 참조하세요. 인스턴스를 최대 절전 모드로 전환하여 인스턴스 메모리(RAM)의 콘텐츠를 저장하려면 Amazon EC2 인스턴스를 최대 절전 모드로 전환 단원을 참조하세요. 인스턴스 수명 주기 작업 간의 차이점에 대해서는 인스턴스 상태 간 차이 단원을 참조하세요.

내용

- EC2 인스턴스 중지 및 시작 작동 방식
- 인스턴스 수동 중지 및 시작
- 인스턴스 자동 중지 및 시작
- 실행 중인 인스턴스와 중지된 인스턴스 모두 찾기
- 초기 및 가장 최근 시작 시간 찾기
- EC2 인스턴스에 대한 중지 방지 활성화

EC2 인스턴스 중지 및 시작 작동 방식

Amazon EC2 인스턴스를 중지하면 변경 내용이 인스턴스의 운영 체제(OS) 수준에 등록되며 일부 리소스는 손실되고 일부는 지속됩니다. 인스턴스를 시작하면 인스턴스 수준에서 변경 사항이 등록됩니다.

주제

- 인스턴스 중지 시 발생하는 상황
- 인스턴스 시작 시 발생하는 상황
- 중지 및 시작에 대한 애플리케이션 응답 테스트
- 인스턴스 시작 및 중지와 관련된 비용

인스턴스 중지 시 발생하는 상황

OS 수준에서 등록된 변경 내용

- API 요청은 버튼 누름 이벤트를 게스트로 전송합니다.
- 버튼 누름 이벤트로 인해 다양한 시스템 서비스가 중지됩니다. 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 몇 분 이내에 완전히 종료되지 않으면 하드 종료가 수행됩니다.
- 인스턴스 실행을 중지합니다.
- 인스턴스 상태가 `stopping`으로 바뀌었다가 다시 `stopped`로 바뀝니다.
- [Auto Scaling] 인스턴스가 Auto Scaling 그룹에 있는 경우, 인스턴스가 `running` 이외의 Amazon EC2 상태이거나 상태 확인의 상태가 `impaired`가 되면 Amazon EC2 Auto Scaling은 인스턴스를 비정상으로 간주하여 대체합니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Auto Scaling 그룹의 인스턴스 상태 확인 섹션을 참조하세요.
- [Windows 인스턴스] Windows 인스턴스를 중지하고 시작할 때 시작 에이전트가 연결된 Amazon EBS 볼륨의 드라이브 문자를 변경하는 등 인스턴스에 대한 태스크를 수행합니다. 이러한 기본값과 변경 방법에 대한 자세한 내용은 the section called “EC2Launch v2” 섹션을 참조하세요.

리소스 손실

- RAM에 저장된 데이터.
- 인스턴스 스토어 볼륨에 저장된 데이터.
- 시작 시 Amazon EC2가 인스턴스에 자동으로 할당한 퍼블릭 IPv4 주소. 변경되지 않는 퍼블릭 IPv4 주소를 유지하려면 탄력적 IP 주소를 인스턴스와 연결할 수 있습니다.

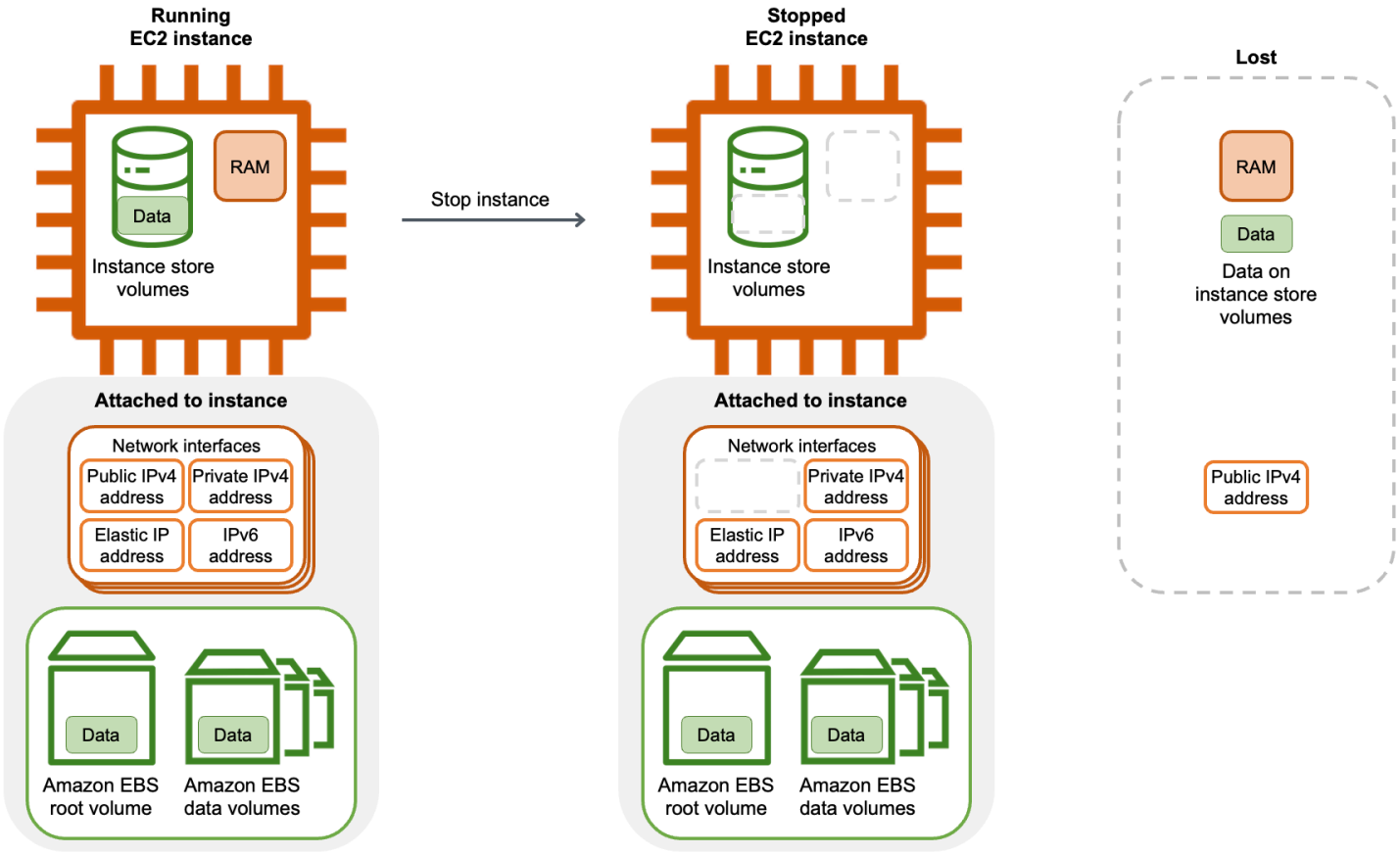
지속되는 리소스

- 연결된 모든 Amazon EBS 루트 및 데이터 볼륨.
- Amazon EBS 볼륨에 저장된 데이터.
- 연결된 모든 네트워크 인터페이스.

  네트워크 인터페이스에는 다음과 같은 리소스가 포함되며, 이러한 리소스도 지속됩니다.
  - 프라이빗 IPv4 주소.
  - IPv6 주소.
  - 인스턴스와 연결된 탄력적 IP 주소. 인스턴스가 중지되면 연결된 탄력적 IP 주소에 대한 요금이 부과됩니다.

다음 다이어그램에서는 Amazon EC2 인스턴스가 중지되었을 때 지속되는 항목과 손실되는 항목을 보여줍니다. 다이어그램은 세 부분으로 나뉩니다. EC2 인스턴스 실행 중이라는 레이블이 지정된 첫 번째

부분에서는 `running` 상태의 인스턴스를 리소스와 함께 보여줍니다. 중지된 EC2 인스턴스라는 레이블이 지정된 두 번째 부분에서는 `stopped` 상태의 인스턴스를 지속되는 리소스와 함께 보여줍니다. 손실이라는 레이블이 지정된 세 번째 부분에서는 인스턴스가 중지될 때 손실된 리소스를 보여줍니다.



Mac 인스턴스를 중지하면 어떤 일이 발생하는지에 대한 자세한 내용은 Amazon EC2 Mac 인스턴스를 중지 또는 종료 섹션을 참조하세요.

인스턴스 시작 시 발생하는 상황

- 대부분의 경우 인스턴스는 새로운 기본 호스트 컴퓨터로 마이그레이션됩니다(전용 호스트 구성에서 인스턴스가 호스트에 할당된 경우와 같은 일부 경우에는 현재 호스트에 남아 있음).
- Amazon EC2는 인스턴스에 탄력적 IP 주소와 연결된 보조 네트워크 인터페이스 또는 보조 프라이빗 IPv4 주소가 없는 한 인스턴스가 퍼블릭 IPv4 주소를 수신하도록 구성된 경우 인스턴스에 새 퍼블릭 IPv4 주소를 할당합니다.

중지 및 시작에 대한 애플리케이션 응답 테스트

AWS Fault Injection Service를 사용하여 인스턴스가 중지 및 시작될 때 애플리케이션이 어떻게 반응하는지 테스트할 수 있습니다. 자세한 내용은 AWS Fault Injection Service 사용 설명서를 참조하세요.

인스턴스 시작 및 중지와 관련된 비용

인스턴스 중지 및 시작과 관련된 비용은 다음과 같습니다.

중지 - 인스턴스 상태가 `shutting-down` 또는 `terminated`으로 변경되는 즉시 해당 인스턴스에 대한 요금은 더 이상 발생하지 않습니다. 중지된 인스턴스에 대한 사용 또는 데이터 전송 요금은 청구되지 않습니다. Amazon EBS 스토리지 볼륨을 저장하려면 요금이 발생합니다.

시작 - 중지된 인스턴스를 시작할 때마다 최소 1분의 사용량에 대해 요금이 부과됩니다. 1분 이후에는 사용한 시간(초)에 대해서만 요금이 부과됩니다. 예를 들어 인스턴스를 20초 동안 실행한 다음 중지하면 1분 사용 요금이 부과됩니다. 인스턴스를 3분 40초간 실행한 경우 사용한 3분 40초에 대한 요금이 부과됩니다.

인스턴스 수동 중지 및 시작

Amazon EBS 지원 인스턴스(EBS 루트 볼륨이 있는 인스턴스)를 중지하고 시작할 수 있습니다. 인스턴스 스토어 루트 볼륨이 있는 인스턴스는 중지하거나 시작할 수 없습니다.

> **Warning**
>
> 인스턴스를 중지하면 인스턴스 스토어 볼륨의 데이터가 삭제됩니다. 인스턴스를 중지하기 전에 필요한 데이터를 인스턴스 스토어 볼륨에서 영구 스토리지(예: Amazon EBS 또는 Amazon S3)로 복사했는지 확인합니다.

[Linux 인스턴스] 인스턴스에서 OS halt 명령을 사용해도 종료가 시작되지 않습니다. halt 명령을 사용하는 경우 인스턴스는 종료되지 않습니다. 대신 CPU를 HLT 안으로 배치하여 CPU 작업이 일시 중단됩니다. 인스턴스는 계속 실행됩니다.

OS shutdown 또는 poweroff 명령을 사용하여 종료를 시작할 수 있습니다. OS 명령을 사용하면 기본적으로 인스턴스가 중지됩니다. 이 동작을 변경할 수 있습니다. 자세한 내용은 인스턴스가 시작하는 종료 동작 변경 섹션을 참조하세요.

Console

Amazon EBS 기반 인스턴스 중지 및 시작

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 인스턴스를 선택한 다음, 인스턴스를 선택합니다.

3. 인스턴스 상태, 인스턴스 중지를 차례로 선택합니다. 이 옵션이 비활성화되어 있으면 해당 인스턴스가 이미 중지되었거나 해당 루트 디바이스가 인스턴스 스토어 볼륨인 것입니다.
4. 확인 메시지가 표시되면 [Stop]을 선택합니다. 인스턴스가 중지하는 데 몇 분 정도 걸릴 수 있습니다.
5. 중지된 인스턴스를 시작하려면 인스턴스를 선택하고 인스턴스 상태, 인스턴스 시작을 차례로 선택합니다.
6. 인스턴스가 `running` 상태가 되는 데 몇 분 정도 걸릴 수 있습니다.

AWS CLI

인스턴스 중지

stop-instances 명령을 사용합니다.

```
aws ec2 stop-instances --instance-ids i-1234567890abcdef0
```

인스턴스 시작

아래와 같이 start-instances 명령을 사용합니다.

```
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

PowerShell

인스턴스 중지

Stop-EC2Instance cmdlet을 사용합니다.

```
Stop-EC2Instance -InstanceId i-1234567890abcdef0
```

인스턴스 시작

Start-EC2Instance cmdlet을 사용합니다.

```
Start-EC2Instance -InstanceId i-1234567890abcdef0
```

중지한 Amazon EBS 기반 인스턴스가 `stopping` 상태에서 "멈춘" 것으로 나타나는 경우 해당 인스턴스를 강제로 중지할 수 있습니다. 자세한 내용은 Amazon EC2 인스턴스 중지 문제 해결 섹션을 참조하세요.

인스턴스 자동 중지 및 시작

다음 서비스를 사용하여 인스턴스 중지 및 시작을 자동화할 수 있습니다.

AWS의 인스턴스 스케줄러

AWS에서 인스턴스 스케줄러를 사용하여 EC2 인스턴스의 시작 및 중지를 자동화할 수 있습니다. 자세한 내용은 인스턴스 스케줄러를 CloudFormation과 함께 사용하여 EC2 인스턴스를 예약하려면 어떻게 해야 합니까?를 참조하세요. 추가 요금이 적용됩니다.

AWS Lambda 및 Amazon EventBridge 규칙

Lambda 및 EventBridge 규칙을 사용하여 예약에 따라 인스턴스를 중지하고 시작할 수 있습니다. 자세한 내용은 Lambda를 사용하여 Amazon EC2 인스턴스를 정기적으로 중지하고 시작하려면 어떻게 해야 하나요?를 참조하세요.

Amazon EC2 Auto Scaling

애플리케이션 로드를 처리하는 데 사용할 수 있는 Amazon EC2 인스턴스의 정확한 수를 확보하려면 Auto Scaling을 생성합니다. Amazon EC2 Auto Scaling에서는 애플리케이션이 항상 트래픽 수요를 처리할 수 있는 적절한 용량을 갖고 필요할 때만 인스턴스를 시작하여 비용을 절감하도록 보장합니다. Amazon EC2 Auto Scaling에서는 불필요한 인스턴스를 중지하는 것이 아니라 종료합니다. Auto Scaling을 설정하려면 Amazon EC2 Auto Scaling 시작하기를 참조하세요.

실행 중인 인스턴스와 중지된 인스턴스 모두 찾기

Amazon EC2 Global View를 사용하여 단일 페이지의 모든 AWS 리전에서 실행 중인 모든 인스턴스와 중지된 인스턴스를 모두 찾을 수 있습니다. 이 기능은 인벤토리를 작성하고 잊어버린 인스턴스를 찾는 데 특히 유용합니다. 글로벌 보기 사용 방법에 대한 자세한 내용은 Amazon EC2 Global View를 사용하여 리전 간 리소스 보기 섹션을 참조하세요.

또는 인스턴스가 있는 각 리전에서 명령 또는 cmdlet을 실행할 수 있습니다.

AWS CLI

리전의 EC2 인스턴스 수를 가져오는 방법

다음 describe-instances 명령을 사용하여 현재 리전의 인스턴스를 계산합니다. 인스턴스가 있는 각 리전에서 이 명령을 실행해야 합니다.

```
aws ec2 describe-instances \
    --region us-east-2 \
```

```
    --query "length(Reservations[].Instances[])"
```

출력의 예시는 다음과 같습니다.

```
27
```

리전의 EC2 인스턴스에 대한 요약 정보를 가져오는 방법

다음 describe-instances 명령을 사용합니다. 인스턴스가 있는 각 리전에서 이 명령을 실행해야 합니다.

```
aws ec2 describe-instances \
    --region us-east-2 \
    --query "Reservations[].Instances[].[InstanceId,InstanceType,PrivateIpAddress]"
 \
    --output table
```

출력의 예시는 다음과 같습니다.

```
-------------------------------------------------------------
|                   DescribeInstances                       |
+---------------------+---------------+-----------------+
|  i-0e3e777f4362f1bf7|  t2.micro     |  10.0.12.9      |
|  i-09453945dcf1529e9|  t2.micro     |  10.0.143.213   |
|  i-08fd74f3f1595fdbd|  m7i.4xlarge  |  10.0.1.103     |
+---------------------+---------------+-----------------+
```

PowerShell

리전의 EC2 인스턴스 수를 가져오는 방법

Get-EC2Instance cmdlet을 사용합니다.

```
(Get-EC2Instance -Region us-east-2).Instances.Length
```

출력의 예시는 다음과 같습니다.

```
27
```

리전의 EC2 인스턴스에 대한 요약 정보를 가져오는 방법

Get-EC2Instance cmdlet을 사용합니다. 인스턴스가 있는 각 리전에서 이 명령을 실행해야 합니다.

```
(Get-EC2Instance).Instances | Select InstanceId, InstanceType, PrivateIpAddress
```

출력의 예시는 다음과 같습니다.

```
InstanceId          InstanceType PrivateIpAddress
----------          ------------ ----------------
i-0e3e777f4362f1bf7 t2.micro     10.0.12.9
i-09453945dcf1529e9 t2.micro     10.0.143.213
i-08fd74f3f1595fdbd m7i.4xlarge  10.0.1.103
```

초기 및 가장 최근 시작 시간 찾기

인스턴스를 설명할 때 인스턴스의 시작 시간은 가장 최근 시작 시간입니다. 인스턴스를 중지하고 시작하면 시작 시간에 새 인스턴스 시작 시간이 반영됩니다. 인스턴스의 초기 시작 시간을 찾으려면 인스턴스를 중지했다가 시작한 후에도 기본 네트워크 인터페이스가 인스턴스에 연결된 시간을 확인합니다.

Console

최신 시작 시간을 찾는 방법

인스턴스를 선택하고 세부 정보 탭의 인스턴스 세부 정보에서 시작 시간을 찾습니다.

초기 시작 시간을 찾는 방법

인스턴스를 선택하고 네트워킹 탭의 네트워크 인터페이스에서 기본 네트워크 인터페이스(디바이스 인덱스가 0)를 찾습니다.

AWS CLI

초기 및 가장 최신 시작 시간을 찾는 방법

describe-instances 명령을 실행하여 지정된 인스턴스의 초기 시작 시간과 최신 시작 시간을 모두 표시합니다.

```
aws ec2 describe-instances \
    --instance-id i-1234567890abcdef0 \
    --query 'Reservations[].Instances[].
{InstanceID:InstanceId,InitialLaunch:NetworkInterfaces[0].Attachment.AttachTime,LastLaunch:L
```

출력의 예시는 다음과 같습니다.

```
[
    {
        "InstanceID": "i-1234567890abcdef0",
        "InitialLaunch": "2024-04-19T00:47:08+00:00",
        "LastLaunch": "2024-05-27T06:24:06+00:00"
    }
]
```

PowerShell

최신 시작 시간을 찾는 방법

Get-EC2Instance cmdlet을 사용합니다.

```
(Get-EC2Instance -InstanceId i-1234567890abcdef0).Instances.LaunchTime
```

출력의 예시는 다음과 같습니다.

```
Monday, May 27, 2024 6:24:06 AM
```

초기 시작 시간을 찾는 방법

Get-EC2Instance cmdlet을 사용합니다.

```
(Get-EC2Instance -
InstanceId i-1234567890abcdef0).Instances.NetworkInterfaces.Attachment.AttachTime
```

출력의 예시는 다음과 같습니다.

```
Friday, April 19, 2024 12:47:08 AM
```

EC2 인스턴스에 대한 중지 방지 활성화

인스턴스의 우발적 중지를 방지하기 위해 해당 인스턴스에 대한 중지 방지를 사용 설정할 수 있습니다. 또한 중지 방지는 인스턴스를 우발적인 종료로부터 보호합니다.

Amazon EC2 `ModifyInstanceAttribute` API의 `DisableApiStop` 속성은 Amazon EC2 콘솔, AWS CLI 또는 Amazon EC2 API를 사용하여 인스턴스 중지 여부를 제어합니다. 인스턴스를 실행할 때 또는 인스턴스가 실행 중이거나 인스턴스가 중지되어 있을 때 이 속성의 값을 설정할 수 있습니다.

고려 사항

- 중지 방지 기능을 활성화해도 운영 체제 명령(shutdown 또는 poweroff)을 사용하여 인스턴스 종료를 시작하는 방식으로 인스턴스 종료를 방지하지 않습니다.
- 중지 방지를 활성화해도 인스턴스를 중지하는 예약 이벤트가 있는 경우 AWS에서 인스턴스 중지를 방지하지 않습니다.
- 중지 보호를 활성화해도 인스턴스가 비정상일 때 또는 스케일 인(scale-in) 이벤트 중에 Amazon EC2 Auto Scaling에서 인스턴스를 종료합니다. 인스턴스 스케일 인 보호를 사용하여 스케일 인할 때 Auto Scaling이 특정 인스턴스를 종료할 수 있는지 여부를 제어할 수 있습니다.
- 보호 중지는 인스턴스의 우발적 중지를 방지할 뿐만 아니라 콘솔, AWS CLI 또는 API를 사용한 우발적 종료도 방지합니다. 그러나 `DisableApiTermination` 속성을 자동으로 설정하지는 않습니다. `DisableApiStop` 속성이 `false`로 설정되면 `DisableApiTermination` 속성 설정에 따라 콘솔, AWS CLI, 또는 API를 사용하여 인스턴스를 종료할 수 있는지 여부가 결정됩니다. 자세한 내용은 Amazon EC2 인스턴스 종료을 참조하세요.
- 인스턴스 스토어 지원 인스턴스에 대해 중지 방지를 활성화할 수 없습니다.
- 스팟 인스턴스에 대한 중지 방지는 활성화할 수 없습니다.
- Amazon EC2 API는 중지 방지를 사용 설정하거나 사용 중지할 때 최종 일관성 모델을 따릅니다. 이는 중지 방지 속성을 설정하기 위해 명령을 실행한 결과가 실행하는 모든 후속 명령에 즉시 표시되지 않을 수 있음을 의미합니다. 자세한 내용은 Amazon EC2 개발자 안내서의 Eventual consistency를 참조하세요.

중지 방지 작업

- 시작 시 인스턴스에 대한 중지 방지 사용 설정
- 실행 중이거나 중지된 인스턴스에 대한 중지 방지 사용 설정
- 실행 중이거나 중지된 인스턴스에 대한 중지 방지 사용 중지

시작 시 인스턴스에 대한 중지 방지 사용 설정

인스턴스를 시작할 때 인스턴스에 대한 중지 방지를 활성화할 수 있습니다.

Console

시작 시 인스턴스에 대한 중지 방지 사용 설정

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 대시보드에서 인스턴스 시작을 선택합니다.

3. 새 인스턴스 시작 마법사에서 인스턴스를 구성합니다.
4. 마법사에서 고급 세부 정보 아래의 중지 방지에 대해 활성화를 선택하여 중지 방지를 활성화합니다.

AWS CLI

시작 시 인스턴스에 대한 중지 방지 사용 설정

run-instances 명령을 사용하여 인스턴스를 시작합니다. 다음 파라미터를 추가합니다.

```
--disable-api-stop
```

PowerShell

시작 시 인스턴스에 대한 중지 방지 사용 설정

New-EC2Instance cmdlet을 사용하여 인스턴스를 시작합니다. 다음 파라미터를 추가합니다.

```
-DisableApiStop $true
```

실행 중이거나 중지된 인스턴스에 대한 중지 방지 사용 설정

인스턴스가 실행 중이거나 중지된 동안 인스턴스에 대한 중지 방지를 활성화할 수 있습니다.

Console

인스턴스에 대한 중지 방지를 활성화하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 작업>인스턴스 설정>중지 방지 변경을 선택합니다.
4. 활성화 확인란을 선택한 다음 저장을 선택합니다.

AWS CLI

인스턴스에 대한 중지 방지를 활성화하는 방법

modify-instance-attribute 명령을 사용합니다.

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --disable-api-stop
```

PowerShell

인스턴스에 대한 중지 방지를 활성화하는 방법

Edit-EC2InstanceAttribute cmdlet을 사용합니다.

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -DisableApiStop $true
```

실행 중이거나 중지된 인스턴스에 대한 중지 방지 사용 중지

다음 방법 중 하나를 사용하여 실행 중이거나 중지된 인스턴스에 대한 중지 방지를 사용 중지할 수 있습니다.

Console

실행 중인 또는 중단된 인스턴스에 대한 중지 방지 비활성화

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 작업(Actions), 인스턴스 설정(Instance Settings), 중지 방지 변경(Change Stop Protection)을 선택합니다.
4. 활성화 확인란을 선택 취소한 다음 저장을 선택합니다.

AWS CLI

실행 중인 또는 중단된 인스턴스에 대한 중지 방지 비활성화

modify-instance-attribute 명령을 사용합니다.

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --no-disable-api-stop
```

PowerShell

인스턴스에 대한 중지 방지를 비활성화하는 방법

Edit-EC2InstanceAttribute cmdlet을 사용합니다.

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -DisableApiStop $false
```

Amazon EC2 인스턴스를 최대 절전 모드로 전환

인스턴스를 최대 절전 모드로 전환하면 Amazon EC2에서 최대 절전 모드(디스크 일시 중단)를 수행하도록 운영 체제에 신호를 보냅니다. 최대 절전 모드는 인스턴스 메모리(RAM)의 내용을 Amazon Elastic Block Store(Amazon EBS) 루트 볼륨에 저장합니다. Amazon EC2는 인스턴스의 EBS 루트 볼륨과 연결된 모든 EBS 데이터 볼륨을 유지합니다. 인스턴스 시작 시:

- 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 활성화되어 있고 최대 절전 모드 사전 조건을 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 워밍할 수 있습니다. 인스턴스를 사전 예열하려면 다음을 수행하세요.

1. 최대 절전 모드가 활성화된 상태에서 시작하세요.
2. 원하는 상태로 만듭니다.
3. 최대 절전 모드로 전환하여 필요할 때마다 원하는 상태로 다시 시작할 수 있습니다.

인스턴스가 `stopped` 상태일 때 최대 절전 모드로 전환된 인스턴스의 인스턴스 사용량에 대한 비용이나 RAM의 콘텐츠를 EBS 루트 볼륨으로 전송할 때 데이터 전송에 대한 비용은 부과되지 않습니다. RAM 콘텐츠에 대한 스토리지를 포함하여 모든 EBS 볼륨의 스토리지에 대해서는 요금이 부과됩니다.

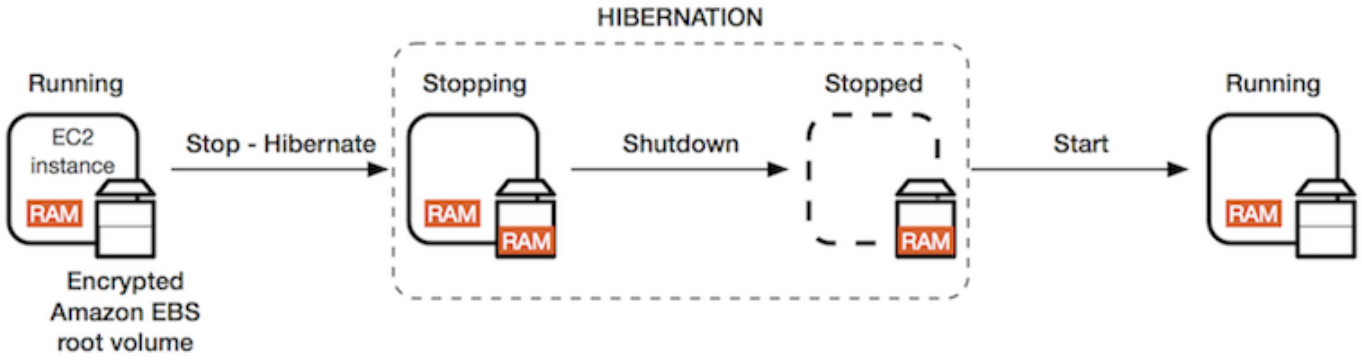
인스턴스가 더 이상 필요하지 않을 경우 stopped(최대 절전 모드) 상태인 경우를 포함해 언제든지 인스턴스를 종료할 수 있습니다. 자세한 내용은 Amazon EC2 인스턴스 종료 섹션을 참조하세요.

내용

- Amazon EC2 인스턴스의 최대 절전 모드 작동 방식
- EC2 인스턴스 최대 절전 모드를 위한 사전 조건
- 최대 절전 모드를 지원하도록 Linux AMI 구성
- Amazon EC2 인스턴스에서 최대 절전 모드 활성화
- 인스턴스에서 KASLR 비활성화(Ubuntu만 해당)
- Amazon EC2 인스턴스를 최대 절전 모드로 전환
- 최대 절전 모드로 전환된 Amazon EC2 인스턴스 시작
- Amazon EC2 인스턴스 최대 절전 모드 문제 해결

Amazon EC2 인스턴스의 최대 절전 모드 작동 방식

다음 다이어그램에서는 EC2 인스턴스의 최대 절전 모드 프로세스에 대한 기본 개요를 보여줍니다.



인스턴스를 최대 절전 모드로 전환할 때 상황

인스턴스를 최대 절전 모드로 전환하면 다음과 같은 상황이 나타납니다.

- 인스턴스는 stopping 상태로 이동합니다. Amazon EC2는 운영 체제에 최대 절전 모드(디스크 일시 중단)를 수행하라는 신호를 보냅니다. 최대 절전 모드를 실행하면 모든 프로세스가 동결되고, RAM의 콘텐츠를 EBS 루트 볼륨에 저장한 다음 일상적인 종료를 수행합니다.
- 종료가 완료되면 인스턴스가 stopped 상태가 됩니다.
- 모든 EBS 볼륨이 인스턴스에 연결된 상태로 유지되고 저장된 RAM 콘텐츠를 포함해 볼륨의 데이터도 유지됩니다.

- 인스턴스에 연결된 Amazon EC2 인스턴스 스토어 볼륨이 있는 경우 인스턴스 스토어 볼륨상의 데이터는 손실됩니다.
- 대부분의 경우 인스턴스가 시작되면 새로운 기본 호스트 컴퓨터로 마이그레이션됩니다. 인스턴스를 중지했다가 시작할 때도 이렇게 됩니다.
- 인스턴스가 시작되면 인스턴스가 부팅되고, 프로세스가 동결 해제되어 상태가 재개되기 전에 운영 체제에서 EBS 루트 볼륨의 RAM 내용을 읽습니다.
- 인스턴스는 프라이빗 IPv4 주소와 모든 IPv6 주소를 유지합니다. 인스턴스가 시작되면 인스턴스에서는 프라이빗 IPv4 주소와 모든 IPv6 주소를 계속 유지합니다.
- Amazon EC2는 퍼블릭 IPv4 주소를 해제합니다. 인스턴스가 시작되면 Amazon EC2에서는 인스턴스에 새 퍼블릭 IPv4 주소를 할당합니다.
- 인스턴스가 연결된 탄력적 IP 주소를 유지합니다. 최대 절전 모드 인스턴스와 연결된 모든 탄력적 IP 주소에 대한 요금이 부과됩니다.

최대 절전 모드과 재부팅, 중지 및 종료 간의 차이점은 인스턴스 상태 간 차이 섹션을 참조하세요.

제한 사항

- 인스턴스를 최대 절전 모드로 전환하면 인스턴스 스토어 볼륨의 데이터가 삭제됩니다.
- (Linux 인스턴스) RAM이 150GiB를 초과하는 Linux 인스턴스는 최대 절전 모드로 전환할 수 없습니다.
- (Windows 인스턴스) RAM이 16GiB를 초과하는 Windows 인스턴스는 최대 절전 모드로 전환할 수 없습니다.
- 인스턴스가 최대 절전 모드로 전환된 동안에는 인스턴스를 수정할 수 없습니다. 이는 인스턴스를 중지했지만 최대 절전 모드가 아닌 경우와는 다릅니다. 최대 절전 모드가 아닌 중지된 인스턴스에서는 인스턴스 유형이나 크기 같은 특정 속성을 수정할 수 있습니다.
- 최대 절전 모드이거나 최대 절전 모드가 사용 설정된 인스턴스에서 스냅샷 또는 AMI를 생성하는 경우, AMI 또는 스냅샷으로 생성된 AMI에서 시작된 새 인스턴스에 연결하지 못할 수 있습니다.
- (스팟 인스턴스만 해당) Amazon EC2에서 스팟 인스턴스를 최대 절전 모드로 전환하는 경우 Amazon EC2에서만 인스턴스를 재개할 수 있습니다. 본인이 스팟 인스턴스를 최대 절전 모드로 전환하는 경우(사용자가 시작한 최대 절전 모드 전환) 본인이 인스턴스를 재개할 수 있습니다. 최대 절전 모드로 전환된 스팟 인스턴스는 용량을 사용할 수 있고 스팟 가격이 지정된 최대 가격 이하인 경우에만 재개될 수 있습니다.
- Auto Scaling 그룹에 속하거나 Amazon ECS에서 사용하는 인스턴스는 최대 절전 모드로 전환할 수 없습니다. 인스턴스가 Auto Scaling 그룹에 있을 때 최대 절전 모드로 전환하려고 하면, Amazon EC2 Auto Scaling 서비스가 중지된 인스턴스를 비정상으로 간주해 이를 종료하고 대체 인스턴스를

시작할 수 있습니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Auto Scaling 그룹의 인스턴스 상태 확인 섹션을 참조하세요.

- UEFI 보안 부팅이 활성화된 상태에서 UEFI 모드로 부팅하도록 구성된 인스턴스는 최대 절전 모드로 전환할 수 없습니다.
- 용량 예약으로 시작된 인스턴스를 최대 절전 모드로 전환하면 최대 절전 모드 인스턴스를 시작하려고 한 후 용량 예약에서 해당 인스턴스를 다시 시작할 수 있도록 보장하지 않습니다.
- FIPS(Federal Information Processing Standard) 모드를 사용하는 경우 5.10 이하의 커널을 사용하는 인스턴스를 최대 절전 모드로 전환할 수 없습니다.
- 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 대해 알려 드립니다.

스팟 인스턴스 최대 절전 모드 전환 시 고려 사항

- 직접 스팟 인스턴스를 최대 절전 모드로 전환하는 경우 용량을 사용할 수 있고 스팟 가격이 지정된 최대 가격 이하이면 다시 시작할 수 있습니다.
- Amazon EC2에서 스팟 인스턴스를 최대 절전 모드로 전환하는 경우:
  - Amazon EC2에서만 인스턴스를 재개할 수 있습니다.
  - 지정된 최대 가격 이하인 스팟 가격으로 용량을 사용할 수 있게 되면 최대 절전 모드로 전환된 스팟 인스턴스를 Amazon EC2에서 재개합니다.
  - Amazon EC2에서 스팟 인스턴스를 최대 절전 모드로 전환하기 전에 최대 절전 모드 시작 2분 전에 중단 고지가 수신됩니다.

  자세한 내용은 스팟 인스턴스 중단 섹션을 참조하세요.

EC2 인스턴스 최대 절전 모드를 위한 사전 조건

온디맨드 인스턴스 또는 스팟 인스턴스를 시작할 때 최대 절전 모드 지원을 활성화할 수 있습니다. 기존 인스턴스(실행 중 또는 중지됨 상태)에서는 최대 절전 모드를 활성화할 수 없습니다. 자세한 내용은 인스턴스의 최대 절전 모드 활성화 섹션을 참조하세요.

인스턴스를 최대 절전 모드로 전환하기 위한 요구 사항

- AWS 리전
- AMI
- 인스턴스 패밀리
- 인스턴스 RAM 크기
- 루트 볼륨 유형
- 루트 볼륨 크기
- 루트 볼륨 암호화
- EBS 볼륨 유형
- 스팟 인스턴스 요청

AWS 리전

모든 AWS 리전에서 최대 절전 모드를 사용할 수 있습니다.

AMI

최대 절전 모드를 지원하는 HVM AMI를 사용해야 합니다. 다음 AMI는 최대 절전 모드를 지원합니다.

Linux AMI

Intel 및 AMD 인스턴스 유형용 AMI

- 2023년 9월 20일 또는 그 이후에 릴리스된 AL2023 AMI [1]
- 2019년 8월 29일 또는 그 이후에 릴리스된 Amazon Linux 2 AMI
- 2018년 11월 16일 또는 그 이후에 릴리스된 Amazon Linux AMI 2018.03
- CentOS 버전 8 AMI [2](추가 구성 필요)
- Fedora 버전 34 이상 AMI [2](추가 구성 필요)
- Red Hat Enterprise Linux(RHEL) 9 AMI [2](추가 구성 필요)
- Red Hat Enterprise Linux(RHEL) 8 AMI [2](추가 구성 필요)
- 일련 번호 20230303 이상으로 릴리스된 Ubuntu 22.04.2 LTS (Jammy Jellyfish) AMI [3]
- 일련 번호 20210820 이상으로 릴리스된 Ubuntu 20.04 LTS(Focal Fossa) [3]
- 일련 번호 20190722.1 이상으로 릴리스된 Ubuntu 18.04 LTS(Bionic Beaver) AMI [3] [5]
- Ubuntu 16.04 LTS(Xenial Xerus) AMI [3] [4] [5](추가 구성 필요)

Graviton 인스턴스 유형용 AMI

- AL2023 AMI(64비트 Arm), 2024년 7월 1일 이후 릴리스 [1]
- Amazon Linux 2 AMI(64비트 Arm), 2024년 6월 20일 이후 릴리스
- Ubuntu 22.04.2 LTS(64비트 Arm)(Jammy Jellyfish) AMI, 일련 번호 20240701 이후에 릴리스 [3]
- Ubuntu 20.04 LTS(64비트 Arm)(Focal Fossa) AMI, 일련 번호 20240701 이상으로 릴리스 [3]

[1] AL2023 최소 AMI의 경우 추가 구성이 필요합니다.

[2] CentOS, Fedora 및 Red Hat Enterprise Linux의 경우 최대 절전 모드는 Nitro 기반 인스턴스에서만 지원됩니다.

[3] Ubuntu 22.04.2 LTS(Jammy Jellyfish), Ubuntu 20.04 LTS(Focal Fossa), Ubuntu 18.04 LTS(Bionic Beaver), Ubuntu 16.04 LTS(Xenial Xerus)가 있는 인스턴스에서는 KASLR을 비활성화하는 것이 좋습니다. 자세한 내용은 인스턴스에서 KASLR 비활성화(Ubuntu만 해당) 섹션을 참조하세요.

[4] Ubuntu 16.04 LTS(Xenial Xerus) AMI의 경우 `t3.nano` 인스턴스 유형에서 최대 절전 모드가 지원되지 않습니다. Ubuntu(Xenial Xerus)는 2021년 4월에 지원을 종료했기 때문에 패치가 제공되지 않습니다. `t3.nano` 인스턴스 유형을 사용하려면 Ubuntu 22.04.2 LTS (Jammy Jellyfish), Ubuntu 20.04 LTS(Focal Fossa) 또는 Ubuntu 18.04 LTS(Bionic Beaver) AMI로 업그레이드하는 것이 좋습니다.

[5] Ubuntu 18.04 LTS(Bionic Beaver)와 Ubuntu 16.04 LTS(Xenial Xerus)에 대한 지원이 종료되었습니다.

최대 절전 모드를 지원하도록 자체 AMI를 구성하려면 최대 절전 모드를 지원하도록 Linux AMI 구성 섹션을 참조하세요.

다른 버전의 Ubuntu 및 기타 운영 체제에 대한 지원도 곧 제공될 예정입니다.

Windows AMI

- 2023년 9월 13일 이후에 릴리스된 Windows Server 2022 AMI 이상
- 2019년 9월 11일 또는 그 이후에 릴리스된 Windows Server 2019 AMI 이상
- 2019년 9월 11일 또는 그 이후에 릴리스된 Windows Server 2016 AMI 이상
- 2019년 9월 11일 또는 그 이후에 릴리스된 Windows Server 2012 R2 AMI 이상
- 2019년 9월 11일 또는 그 이후에 릴리스된 Windows Server 2012 AMI 이상

인스턴스 패밀리

최대 절전 모드를 지원하는 인스턴스 패밀리를 사용해야 합니다. 단, 베어 메탈 인스턴스는 지원되지 않습니다.

- 범용: M3, M4, M5, M5a, M5ad, M5d, M6a, M6g, M6gd, M6i, M6id, M6idn, M6in, M7a, M7g, M7gd, M7i, M7i-flex, M8g, M8gd, T2, T3, T3a, T4g
- 컴퓨팅 최적화: C3, C4, C5, C5d, C6a, C6g, C6gd, C6gn, C6i, C6id, C6in, C7a, C7g, C7gd, C7gn, C7i, C7i-flex, C8g, C8gd
- 메모리 최적화: R3, R4, R5, R5a, R5ad, R5d, R6a, R6g, R6gd, R6idn, R6in, R7a, R7g, R7gd, R7i, R7iz, R8g, R8gd, X2gd
- 스토리지 최적화: I3, I3en, I4g, I7ie, I8g, Im4gn, Is4gen

Console

최대 절전 모드를 지원하는 인스턴스 유형을 가져오려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스 유형을 선택합니다.
3. On-Demand Hibernation support = true 필터를 추가합니다.
4. (선택 사항) 필터를 추가하여 관심 있는 특정 인스턴스 유형에 범위를 추가로 지정합니다.

AWS CLI

최대 절전 모드를 지원하는 인스턴스 유형을 가져오려면

describe-instance-types 명령을 사용합니다. 사용 가능한 인스턴스 유형은 리전마다 다릅니다.

```
aws ec2 describe-instance-types \
    --filters Name=hibernation-supported,Values=true \
    --query "InstanceTypes[*].[InstanceType]" \
    --output text | sort
```

PowerShell

최대 절전 모드를 지원하는 인스턴스 유형을 가져오려면

Get-EC2InstanceType cmdlet을 사용합니다. 사용 가능한 인스턴스 유형은 리전마다 다릅니다.

```
(Get-EC2InstanceType `
    -Filter @{Name="hibernation-supported"; Values="true"}).InstanceType | Sort-
Object
```

인스턴스 RAM 크기

Linux 인스턴스 - 150GiB 미만이어야 합니다.

Windows 인스턴스 - 16GiB 이하여야 합니다. T3 또는 T3a Windows 인스턴스를 최대 절전 모드로 전환하려면 최소 1GiB의 RAM을 사용하는 것이 좋습니다.

루트 볼륨 유형

루트 볼륨은 인스턴스 스토어 볼륨이 아니라 EBS 볼륨이어야 합니다.

루트 볼륨 크기

루트 볼륨 크기는 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.

루트 볼륨 암호화

최대 절전 모드일 때 메모리에 있는 중요한 콘텐츠를 보호할 수 있도록 루트 볼륨을 암호화해야 합니다. RAM 데이터가 EBS 루트 볼륨으로 이전하면 항상 암호화됩니다. 루트 볼륨 암호화는 인스턴스 시작 시 적용됩니다.

루트 볼륨이 암호화된 EBS 볼륨인지 확인하려면 다음 세 가지 옵션 중 하나를 사용합니다.

- EBS 암호화 기본 지원 - 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 기본으로 암호화를 참조하세요.
- EBS “단일 단계” 암호화 - 암호화되지 않은 AMI에서 암호화된 EBS 지원 EC2 인스턴스를 시작하고, 그와 동시에 최대 절전 모드를 활성화할 수도 있습니다. 자세한 내용은 EBS-backed AMI에서 암호화 사용 섹션을 참조하세요.
- 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 및 AMI 복사 섹션을 참조하세요.

EBS 볼륨 유형

EBS 볼륨은 다음 EBS 볼륨 유형 중 하나를 사용해야 합니다.

- 범용 SSD(gp2 및 gp3)
- 프로비저닝된 IOPS SSD(io1 및 io2)

프로비저닝된 IOPS SSD 볼륨 유형을 선택한 경우 최대 절전 모드에서 최적의 성능을 얻으려면 적절한 IOPS로 EBS 볼륨을 프로비저닝해야 합니다. 자세한 내용은 Amazon EBS 사용 설명서의 Amazon EBS volume types를 참조하세요.

스팟 인스턴스 요청

스팟 인스턴스에는 다음 요구 사항이 적용됩니다.

- 스팟 인스턴스 요청 유형은 persistent여야 합니다.
- 스팟 인스턴스 요청에서 시작 그룹을 지정할 수 없습니다.

최대 절전 모드를 지원하도록 Linux AMI 구성

다음 Linux AMI는 이 섹션에서 설명하는 추가 구성 단계를 완료하면 Amazon EC2 인스턴스의 최대 절전 모드를 지원할 수 있습니다.

다음에 대한 추가 구성이 필요합니다.

- 2023년 9월 20일 이후 릴리스된 AL2023 최소 AMI
- 2019년 8월 29일 이후에 릴리스된 Amazon Linux 2 최소 AMI
- 2019년 8월 29일 이전에 릴리스된 Amazon Linux 2
- 2018년 11월 16일 이전에 릴리스된 Amazon Linux
- CentOS 버전 8 이상
- Fedora 버전 34 이상
- Red Hat Enterprise Linux 버전 8 또는 9
- 일련 번호 20210820 이전에 릴리스된 Ubuntu 20.04 LTS(Focal Fossa)
- 일련 번호 20190722.1 이전에 릴리스된 Ubuntu 18.04(Bionic Beaver)
- Ubuntu 16.04(Xenial Xerus)

최대 절전 모드를 지원하고 추가 구성이 필요하지 않은 Linux 및 Windows AMI는 AMI의 내용을 참조하세요.

자세한 내용은 Amazon Linux 2 인스턴스에서 인스턴스 소프트웨어 업데이트를 참조하세요.

2023년 9월 20일 이후 릴리스된 AL2023 최소 AMI

최대 절전 모드를 지원하도록 2023년 9월 20일 이후에 릴리스된 AL2023 최소 AMI를 구성하려면 다음과 같이 하세요.

1. 리포지토리에서 `ec2-hibinit-agent` 패키지를 설치합니다.

```
[ec2-user ~]$ sudo dnf install ec2-hibinit-agent
```

2. 서비스를 다시 시작합니다.

```
[ec2-user ~]$ sudo systemctl start hibinit-agent
```

2019년 8월 29일 이후에 릴리스된 Amazon Linux 2 최소 AMI

최대 절전 모드를 지원하도록 2019년 8월 29일 이후에 릴리스된 Amazon Linux 2 최소 AMI 구성

1. 리포지토리에서 `ec2-hibinit-agent` 패키지를 설치합니다.

```
[ec2-user ~]$ sudo yum install ec2-hibinit-agent
```

2. 서비스를 다시 시작합니다.

```
[ec2-user ~]$ sudo systemctl start hibinit-agent
```

2019년 8월 29일 이전에 릴리스된 Amazon Linux 2

최대 절전 모드를 지원하도록 2019년 8월 29일 이전에 릴리스된 Amazon Linux 2 AMI를 구성하려면

1. 커널을 `4.14.138-114.102` 이상으로 업데이트합니다.

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

2. 리포지토리에서 `ec2-hibinit-agent` 패키지를 설치합니다.

```
[ec2-user ~]$ sudo yum install ec2-hibinit-agent
```

3. 인스턴스를 재부팅합니다.

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

4. 커널 버전이 `4.14.138-114.102` 이상으로 업데이트되었는지 확인합니다.

```
[ec2-user ~]$ uname -a
```

5. 인스턴스를 중지하고 AMI를 생성합니다. 자세한 내용은 Amazon EBS 지원 AMI 생성 섹션을 참조하세요.

2018년 11월 16일 이전에 릴리스된 Amazon Linux

최대 절전 모드를 지원하도록 2018년 11월 16일 이전에 릴리스된 Amazon Linux AMI를 구성하려면

1. 커널을 `4.14.77-70.59` 이상으로 업데이트합니다.

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

2. 리포지토리에서 `ec2-hibinit-agent` 패키지를 설치합니다.

```
[ec2-user ~]$ sudo yum install ec2-hibinit-agent
```

3. 인스턴스를 재부팅합니다.

```
[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 Amazon EBS 지원 AMI 생성 섹션을 참조하세요.

CentOS 버전 8 이상

최대 절전 모드를 지원하도록 CentOS 버전 8 이상 AMI를 구성하려면

1. 커널을 `4.18.0-305.7.1.el8_4.x86_64` 이상으로 업데이트합니다.

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

2. Fedora Extra Packages for Enterprise Linux(EPEL) 리포지토리를 설치합니다.

```
[ec2-user ~]$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

3. 리포지토리에서 `ec2-hibinit-agent` 패키지를 설치합니다.

```
[ec2-user ~]$ sudo yum install ec2-hibinit-agent
```

4. 부팅 시 최대 절전 모드 에이전트가 시작되도록 활성화합니다.

```
[ec2-user ~]$ sudo systemctl enable hibinit-agent.service
```

5. 인스턴스를 재부팅합니다.

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

6. 커널 버전이 `4.18.0-305.7.1.el8_4.x86_64` 이상으로 업데이트되었는지 확인합니다.

```
[ec2-user ~]$ uname -a
```

Fedora 버전 34 이상

최대 절전 모드를 지원하도록 Fedora 버전 34 이상 AMI를 구성하려면

1. 커널을 `5.12.10-300.fc34.x86_64` 이상으로 업데이트합니다.

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

2. 리포지토리에서 `ec2-hibinit-agent` 패키지를 설치합니다.

```
[ec2-user ~]$ sudo dnf install ec2-hibinit-agent
```

3. 부팅 시 최대 절전 모드 에이전트가 시작되도록 활성화합니다.

```
[ec2-user ~]$ sudo systemctl enable hibinit-agent.service
```

4. 인스턴스를 재부팅합니다.

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

5. 커널 버전이 `5.12.10-300.fc34.x86_64` 이상으로 업데이트되었는지 확인합니다.

```
[ec2-user ~]$ uname -a
```

Red Hat Enterprise Linux 버전 8 또는 9

최대 절전 모드를 지원하도록 Red Hat Enterprise Linux 8 또는 9 AMI 구성 방법

1. 커널을 `4.18.0-305.7.1.el8_4.x86_64` 이상으로 업데이트합니다.

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

2. Fedora Extra Packages for Enterprise Linux(EPEL) 리포지토리를 설치합니다.

   RHEL 버전 8:

```
[ec2-user ~]$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

   RHEL 버전 9:

```
[ec2-user ~]$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm
```

3. 리포지토리에서 `ec2-hibinit-agent` 패키지를 설치합니다.

```
[ec2-user ~]$ sudo yum install ec2-hibinit-agent
```

4. 부팅 시 최대 절전 모드 에이전트가 시작되도록 활성화합니다.

```
[ec2-user ~]$ sudo systemctl enable hibinit-agent.service
```

5. 인스턴스를 재부팅합니다.

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

6. 커널 버전이 4.18.0-305.7.1.el8_4.x86_64 이상으로 업데이트되었는지 확인합니다.

```
[ec2-user ~]$ uname -a
```

일련 번호 20210820 이전에 릴리스된 Ubuntu 20.04 LTS(Focal Fossa)

최대 절전 모드를 지원하도록 일련 번호 20210820 이전에 릴리스된 Ubuntu 20.04 LTS(Focal Fossa) AMI를 구성하는 법

1. Linux-aws-kernel을 5.8.0-1038.40 이상으로 업데이트하고 grub2를 2.04-1ubuntu26.13 이상으로 업데이트합니다.

```
[ec2-user ~]$ sudo apt update
[ec2-user ~]$ sudo apt dist-upgrade
```

2. 인스턴스를 재부팅합니다.

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

3. 커널 버전이 5.8.0-1038.40 이상으로 업데이트되었는지 확인합니다.

```
[ec2-user ~]$ uname -a
```

4. grub2 버전이 2.04-1ubuntu26.13 이상으로 업데이트되었는지 확인합니다.

```
[ec2-user ~]$ dpkg --list | grep grub2-common
```

일련 번호 20190722.1 이전에 릴리스된 Ubuntu 18.04(Bionic Beaver)

최대 절전 모드를 지원하도록 일련 번호 20190722.1 이전에 릴리스된 Ubuntu 18.04 LTS AMI를 구성하려면

1. 커널을 4.15.0-1044 이상으로 업데이트합니다.

```
[ec2-user ~]$ sudo apt update
```

```
[ec2-user ~]$ sudo apt dist-upgrade
```

2. 리포지토리에서 `ec2-hibinit-agent` 패키지를 설치합니다.

```
[ec2-user ~]$ sudo apt install ec2-hibinit-agent
```

3. 인스턴스를 재부팅합니다.

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

4. 커널 버전이 `4.15.0-1044` 이상으로 업데이트되었는지 확인합니다.

```
[ec2-user ~]$ uname -a
```

Ubuntu 16.04(Xenial Xerus)

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.

> **Important**
>
> `linux-aws-hwe` 커널 패키지는 Canonical을 통해 지원됩니다. Ubuntu 16.04 LTS에 대한 표준 지원은 2021년 4월에 종료되었으며 패키지는 더 이상 정기 업데이트를 받지 않습니다. 그러나 연장된 보안 유지 관리 지원이 2024년에 종료될 때까지 추가 보안 업데이트를 받게 됩니다. 자세한 내용은 Canonical Ubuntu 블로그의 Ubuntu 16.04 LTS용 Amazon EC2 Hibernation, 이제 사용 가능을 참조하세요.
> Ubuntu 20.04 LTS(Focal Fossa) AMI 또는 Ubuntu 18.04 LTS(Bionic Beaver) AMI로 업그레이드하는 것이 좋습니다.

최대 절전 모드를 지원하도록 Ubuntu 16.04 LTS AMI를 구성하려면

1. 커널을 `4.15.0-1058-aws` 이상으로 업데이트합니다.

```
[ec2-user ~]$ sudo apt update
[ec2-user ~]$ sudo apt install linux-aws-hwe
```

2. 리포지토리에서 `ec2-hibinit-agent` 패키지를 설치합니다.

```
[ec2-user ~]$ sudo apt install ec2-hibinit-agent
```

3. 인스턴스를 재부팅합니다.

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

4. 커널 버전이 `4.15.0-1058-aws` 이상으로 업데이트되었는지 확인합니다.

```
[ec2-user ~]$ uname -a
```

Amazon EC2 인스턴스에서 최대 절전 모드 활성화

인스턴스를 최대 절전 모드로 전환하려면 먼저 인스턴스를 시작하는 동안 최대 절전 모드로 전환하도록 활성화해야 합니다.

> **Important**
> 인스턴스를 시작한 후에는 인스턴스에 대해 최대 절전 모드를 활성화하거나 비활성화할 수 없습니다.

업무

- 온디맨드 인스턴스 최대 절전 모드 활성화
- 스팟 인스턴스 최대 절전 모드 활성화
- 인스턴스의 최대 절전 모드가 활성화되었는지 보기

온디맨드 인스턴스 최대 절전 모드 활성화

온디맨드 인스턴스에서 최대 절전 모드를 활성화할 수 있습니다.

Console

온디맨드 인스턴스 최대 절전 모드를 활성화하는 방법

1. 절차에 따라 인스턴스를 시작하되 다음 단계를 완료하여 최대 절전 모드를 활성화할 때까지 인스턴스를 시작하지 마세요.
2. 최대 절전 모드를 활성화하려면 인스턴스 시작 마법사에서 다음 필드를 구성합니다.

a. Application and OS Images (Amazon Machine Image)(애플리케이션 및 OS 이미지 (Amazon Machine Image))에서 최대 절전 모드를 지원하는 AMI를 선택합니다. 자세한 내용은 AMI 섹션을 참조하세요.

b. 인스턴스 유형(Instance type)에서 지원되는 인스턴스 유형을 선택합니다. 자세한 내용은 인스턴스 패밀리 섹션을 참조하세요.

c. Configure storage,(스토리지 구성)에서 Advanced(고급)(오른쪽)를 선택하고 루트 볼륨에 대해 다음 정보를 지정합니다.

- 크기(GiB)(Size (GiB))에 EBS 루트 볼륨 크기를 입력합니다. 볼륨은 RAM 내용을 저장하고 예상 사용량을 수용할 수 있을 정도로 커야 합니다.
- Volume type(볼륨 유형)에서 지원되는 EBS 볼륨 유형(범용 SSD(gp2 및 gp3) 또는 프로비저닝된 IOPS SSD(io1 또는 io2))을 선택합니다.
- Encrypted(암호화)에서 Yes(예)를 선택합니다. 이 AWS 리전에서 기본적으로 암호화를 활성화한 경우 Yes(예)가 선택됩니다.
- KMS key(KMS 키)에서 볼륨의 암호화 키를 선택합니다. 이 AWS 리전에서 기본적으로 암호화를 활성화한 경우 기본 암호화 키가 선택됩니다.

루트 볼륨의 사전 조건에 대한 자세한 내용은 EC2 인스턴스 최대 절전 모드를 위한 사전 조건 섹션을 참조하세요.

d. Advanced details(고급 세부 정보)를 확장하고 Stop - Hibernate behavior(중지 - 최대 절전 모드 동작)에 대해 Enable(활성화)을 선택합니다.

3. Summary(요약) 패널에서 인스턴스 구성을 검토한 다음 Launch instance(인스턴스 시작)를 선택합니다. 자세한 내용은 콘솔에서 인스턴스 시작 마법사를 사용하여 EC2 인스턴스 시작 섹션을 참조하세요.

AWS CLI

온디맨드 인스턴스 최대 절전 모드를 활성화하는 방법

run-instances 명령을 사용하여 인스턴스를 시작합니다. `--block-device-mappings file://mapping.json` 파라미터를 사용하여 EBS 루트 볼륨 파라미터를 지정하고, `--hibernation-options Configured=true` 파라미터를 사용하여 최대 절전 모드를 활성화합니다.

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type m5.large \
```

```
    --block-device-mappings file://mapping.json \
    --hibernation-options Configured=true \
    --count 1 \
    --key-name MyKeyPair
```

mapping.json에서 다음을 지정합니다.

```
[
    {
        "DeviceName": "/dev/xvda",
        "Ebs": {
            "VolumeSize": 30,
            "VolumeType": "gp2",
            "Encrypted": true
        }
    }
]
```

DeviceName의 값은 AMI와 연결된 루트 디바이스 이름과 일치해야 합니다. 루트 디바이스 이름을 찾으려면 describe-images 명령을 사용합니다.

```
aws ec2 describe-images --image-id ami-0abcdef1234567890
```

이 AWS 리전에서 기본적으로 암호화를 활성화한 경우 "Encrypted": true를 생략할 수 있습니다.

PowerShell

온디맨드 인스턴스 최대 절전 모드를 활성화하는 방법

New-EC2Instance 명령을 사용하여 인스턴스를 시작합니다. 먼저 블록 디바이스 매핑을 정의한 다음 -BlockDeviceMappings 파라미터를 사용하여 명령에 추가하여 EBS 루트 볼륨을 지정합니다. -HibernationOptions_Configured $true 파라미터를 사용하여 최대 절전 모드를 활성화합니다.

```
$ebs_encrypt = New-Object Amazon.EC2.Model.BlockDeviceMapping
$ebs_encrypt.DeviceName = "/dev/xvda"
$ebs_encrypt.Ebs = New-Object Amazon.EC2.Model.EbsBlockDevice
$ebs_encrypt.Ebs.VolumeSize = 30
$ebs_encrypt.Ebs.VolumeType = "gp2"
$ebs_encrypt.Ebs.Encrypted = $true
```

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType m5.large `
    -BlockDeviceMappings $ebs_encrypt `
    -HibernationOptions_Configured $true `
    -MinCount 1 `
    -MaxCount 1 `
    -KeyName MyKeyPair
```

DeviceName의 값은 AMI와 연결된 루트 디바이스 이름과 일치해야 합니다. 루트 디바이스 이름을 찾으려면 Get-EC2Image 명령을 사용합니다.

```
Get-EC2Image -ImageId ami-0abcdef1234567890
```

이 AWS 리전에서 기본적으로 암호화를 활성화한 경우 블록 디바이스 매핑에서 Encrypted = $true를 생략할 수 있습니다.

스팟 인스턴스 최대 절전 모드 활성화

스팟 인스턴스에서 최대 절전 모드를 활성화할 수 있습니다. 중단 시 스팟 인스턴스 최대 절전 모드 전환에 대한 자세한 내용은 스팟 인스턴스 중단 섹션을 참조하세요.

Console

스팟 인스턴스 최대 절전 모드를 활성화하는 방법

1. 절차에 따라 인스턴스 시작 마법사를 사용하여 스팟 인스턴스를 요청하되 다음과 같은 단계를 완료하여 최대 절전 모드를 활성화할 때까지 인스턴스를 시작하지 마세요.
2. 최대 절전 모드를 활성화하려면 인스턴스 시작 마법사에서 다음 필드를 구성합니다.
   a. Application and OS Images (Amazon Machine Image)(애플리케이션 및 OS 이미지(Amazon Machine Image))에서 최대 절전 모드를 지원하는 AMI를 선택합니다. 자세한 내용은 AMI 섹션을 참조하세요.
   b. 인스턴스 유형(Instance type)에서 지원되는 인스턴스 유형을 선택합니다. 자세한 내용은 인스턴스 패밀리 섹션을 참조하세요.
   c. Configure storage,(스토리지 구성)에서 Advanced(고급)(오른쪽)를 선택하고 루트 볼륨에 대해 다음 정보를 지정합니다.

- 크기(GiB)(Size (GiB))에 EBS 루트 볼륨 크기를 입력합니다. 볼륨은 RAM 내용을 저장하고 예상 사용량을 수용할 수 있을 정도로 커야 합니다.
- Volume type(볼륨 유형)에서 지원되는 EBS 볼륨 유형(범용 SSD(gp2 및 gp3) 또는 프로비저닝된 IOPS SSD(io1 또는 io2))을 선택합니다.
- Encrypted(암호화)에서 Yes(예)를 선택합니다. 이 AWS 리전에서 기본적으로 암호화를 활성화한 경우 Yes(예)가 선택됩니다.
- KMS key(KMS 키)에서 볼륨의 암호화 키를 선택합니다. 이 AWS 리전에서 기본적으로 암호화를 활성화한 경우 기본 암호화 키가 선택됩니다.

루트 볼륨의 사전 조건에 대한 자세한 내용은 EC2 인스턴스 최대 절전 모드를 위한 사전 조건 섹션을 참조하세요.

d. 고급 세부 정보를 확장하고 스팟 인스턴스를 구성하는 필드 외에 다음을 수행합니다.

i. 요청 유형에 지속적을 선택합니다.

ii. 인터럽트 방식에 최대 절전 모드를 선택합니다. 또는 중지 - 최대 절전 모드 동작에 활성화를 선택합니다. 두 필드에서 모두 스팟 인스턴스에 대한 최대 절전 모드가 활성화됩니다. 둘 중 하나만 구성하면 됩니다.

3. Summary(요약) 패널에서 인스턴스 구성을 검토한 다음 Launch instance(인스턴스 시작)를 선택합니다. 자세한 내용은 콘솔에서 인스턴스 시작 마법사를 사용하여 EC2 인스턴스 시작 섹션을 참조하세요.

AWS CLI

스팟 인스턴스 최대 절전 모드를 활성화하는 방법

run-instances 명령을 사용하여 스팟 인스턴스를 요청합니다. `--block-device-mappings file://mapping.json` 파라미터를 사용하여 EBS 루트 볼륨 파라미터를 지정하고, `--hibernation-options Configured=true` 파라미터를 사용하여 최대 절전 모드를 활성화합니다. 스팟 요청 유형(SpotInstanceType)은 persistent여야 합니다.

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type c4.xlarge \
    --block-device-mappings file://mapping.json \
    --hibernation-options Configured=true \
    --count 1 \
```

```
    --key-name MyKeyPair
    --instance-market-options
        {
           "MarketType":"spot",
           "SpotOptions":{
              "MaxPrice":"1",
              "SpotInstanceType":"persistent"
            }
        }
```

다음과 같이 mapping.json의 EBS 루트 볼륨 파라미터를 지정합니다.

```
[
    {
        "DeviceName": "/dev/xvda",
        "Ebs": {
            "VolumeSize": 30,
            "VolumeType": "gp2",
            "Encrypted": true
        }
    }
]
```

DeviceName의 값은 AMI와 연결된 루트 디바이스 이름과 일치해야 합니다. 루트 디바이스 이름을 찾으려면 describe-images 명령을 사용합니다.

```
aws ec2 describe-images --image-id ami-0abcdef1234567890
```

이 AWS 리전에서 기본적으로 암호화를 활성화한 경우 "Encrypted": true를 생략할 수 있습니다.

PowerShell

스팟 인스턴스 최대 절전 모드를 활성화하는 방법

New-EC2Instance 명령을 사용하여 스팟 인스턴스를 요청합니다. 먼저 블록 디바이스 매핑을 정의한 다음 -BlockDeviceMappings 파라미터를 사용하여 명령에 추가하여 EBS 루트 볼륨을 지정합니다. -HibernationOptions_Configured $true 파라미터를 사용하여 최대 절전 모드를 활성화합니다.

```
$ebs_encrypt = New-Object Amazon.EC2.Model.BlockDeviceMapping
```

```
$ebs_encrypt.DeviceName = "/dev/xvda"
$ebs_encrypt.Ebs = New-Object Amazon.EC2.Model.EbsBlockDevice
$ebs_encrypt.Ebs.VolumeSize = 30
$ebs_encrypt.Ebs.VolumeType = "gp2"
$ebs_encrypt.Ebs.Encrypted = $true

New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType m5.large `
    -BlockDeviceMappings $ebs_encrypt `
    -HibernationOptions_Configured $true `
    -MinCount 1 `
    -MaxCount 1 `
    -KeyName MyKeyPair `
    -InstanceMarketOption @(
        MarketType = spot;
        SpotOptions @{
        MaxPrice = 1;
        SpotInstanceType = persistent}
    )
```

`DeviceName`의 값은 AMI와 연결된 루트 디바이스 이름과 일치해야 합니다. 루트 디바이스 이름을 찾으려면 Get-EC2Image 명령을 사용합니다.

```
Get-EC2Image -ImageId ami-0abcdef1234567890
```

이 AWS 리전에서 기본적으로 암호화를 활성화한 경우 블록 디바이스 매핑에서 `Encrypted = $true`를 생략할 수 있습니다.

인스턴스의 최대 절전 모드가 활성화되었는지 보기

인스턴스에 최대 절전 모드가 활성화되어 있는지 확인할 수 있습니다.

Console

인스턴스에 대해 최대 절전 모드가 활성화되어 있는지 여부 확인

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 세부 정보 탭의 인스턴스 세부 정보 섹션에서 중지 - 최대 절전 모드 동작을 살펴봅니다. 활성은 인스턴스에 대해 최대 절전 모드가 활성화되어 있음을 나타냅니다.

AWS CLI

인스턴스에 대해 최대 절전 모드가 활성화되어 있는지 여부 확인

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
}
```

PowerShell

인스턴스에 대해 최대 절전 모드가 활성화되어 있는지 여부 확인

Get-EC2Instance cmdlet을 사용하여 최대 절전 모드가 활성화된 인스턴스를 필터링합니다.

```
(Get-EC2Instance `
    -Filter @{Name="hibernation-options.configured"; Values="true"}).Instances
```

인스턴스에서 KASLR 비활성화(Ubuntu만 해당)

Ubuntu 16.04 LTS(Xenial Xerus), 일련 번호 20190722.1 이후에 릴리스된 Ubuntu 18.04 LTS(Bionic Beaver) 또는 일련 번호 20210820 이후에 릴리스된 Ubuntu 20.04 LTS(Focal Fossa)가 있는 새로 시작된 인스턴스에서 최대 절전 모드를 실행하려면 KASLR(커널 주소 스페이스 레이아웃 무작위화)을 비활성화하는 것이 좋습니다. Ubuntu 16.04 LTS, Ubuntu 18.04 LTS 또는 Ubuntu 20.04 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 참조하세요.

Ubuntu로 시작된 인스턴스에서 KASLR을 비활성화하려면

1. SSH를 사용하여 인스턴스에 연결합니다. 자세한 내용은 SSH를 사용하여 Linux 인스턴스에 연결 섹션을 참조하세요.
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 편집하세요.

```
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.

```
sudo update-grub
```

5. 인스턴스를 재부팅합니다.

```
sudo reboot
```

6. 다음 명령을 실행하여 `nokaslr`이 추가되었는지 확인합니다.

```
cat /proc/cmdline
```

명령의 출력에는 `nokaslr` 옵션이 포함되어야 합니다.

Amazon EC2 인스턴스를 최대 절전 모드로 전환

인스턴스가 EBS 지원 인스턴스이고, 최대 절전 모드 활성화됨이며, 최대 절전 모드 사전 조건이 충족되면 온디맨드 인스턴스 또는 스팟 인스턴스에 대한 최대 절전 모드를 시작할 수 있습니다. 인스턴스를 최대 절전 모드로 전환할 수 없는 경우 정상 종료가 진행됩니다.

Console

인스턴스를 최대 절전 모드로 전환하는 방법

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

3. 인스턴스를 선택하고 인스턴스 상태, 인스턴스를 최대 절전 모드로 전환을 차례로 선택합니다. 인스턴스를 최대 절전 모드로 전환이 비활성화되어 있으면 해당 인스턴스가 이미 최대 절전 모드로 전환 또는 중지되었거나 최대 절전 모드로 전환할 수 없는 것입니다. 자세한 내용은 EC2 인스턴스 최대 절전 모드를 위한 사전 조건 섹션을 참조하세요.
4. 확인 메시지가 나타나면 최대 절전 모드로 전환을 선택합니다. 인스턴스가 최대 절전 모드로 전환하는 데 몇 분 정도 걸릴 수 있습니다. 인스턴스가 최대 절전 모드 상태인 경우 인스턴스 상태가 먼저 중지 중(Stopping)으로 바뀐 후 중지됨(Stopped)으로 바뀝니다.

AWS CLI

인스턴스를 최대 절전 모드로 전환하는 방법

stop-instances 명령을 사용하여 `--hibernate` 파라미터를 지정합니다.

```
aws ec2 stop-instances \
    --instance-ids i-1234567890abcdef0 \
    --hibernate
```

PowerShell

인스턴스를 최대 절전 모드로 전환하는 방법

Stop-EC2Instance cmdlet을 사용합니다.

```
Stop-EC2Instance `
    -InstanceId i-1234567890abcdef0 `
    -Hibernate $true
```

인스턴스에서 최대 절전 모드가 시작되었는지 확인할 수 있습니다.

Console

인스턴스에 대해 최대 절전 모드가 시작되었는지 확인

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 세부 정보 탭의 인스턴스 세부 정보 섹션에서 상태 전환 메시지의 값을 확인합니다.

Client.UserInitiatedHibernate: User initiated hibernate는 온디맨드 인스턴스 또는 스팟 인스턴스에 대한 최대 절전 모드를 시작했다는 것을 나타냅니다.

AWS CLI

인스턴스에 대해 최대 절전 모드가 시작되었는지 확인

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 필드에 온디맨드 인스턴스 또는 스팟 인스턴스에 대한 최대 절전 모드가 시작되었다는 것이 표시됩니다.

```
"StateReason": {
    "Code": "Client.UserInitiatedHibernate"
}
```

PowerShell

인스턴스에 대해 최대 절전 모드가 시작되었는지 확인

Get-EC2Instance cmdlet을 사용하고 `state-reason-code` 필터를 지정하여 최대 절전 모드가 시작된 인스턴스를 확인합니다.

```
Get-EC2Instance `
    -Filter @{Name="state-reason-code";Value="Client.UserInitiatedHibernate"}
```

최대 절전 모드로 전환된 Amazon EC2 인스턴스 시작

중지된 인스턴스를 다시 시작하는 것과 같은 방법으로 최대 절전 모드 인스턴스를 시작합니다.

스팟 인스턴스의 경우 Amazon EC2에서 인스턴스를 최대 절전 모드로 전환했으면 Amazon EC2에서만 재개할 수 있습니다. 본인이 최대 절전 모드로 전환한 경우에만 최대 절전 모드로 전환된 스팟 인스턴스를 본인이 재개할 수 있습니다. 스팟 인스턴스는 용량을 사용할 수 있고 스팟 가격이 지정된 최대 가격 이하인 경우에만 재개될 수 있습니다.

Console

최대 절전 모드로 전환된 인스턴스 시작

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 최대 절전 모드로 전환된 인스턴스를 선택하고 인스턴스 상태, 인스턴스 시작을 차례로 선택합니다. 인스턴스가 `running` 상태가 되는 데 몇 분 정도 걸릴 수 있습니다. 이 과정에서 인스턴스 상태 확인에는 그 인스턴스가 시작될 때까지 실패 상태가 표시됩니다.

AWS CLI

최대 절전 모드로 전환된 인스턴스 시작

아래와 같이 start-instances 명령을 사용합니다.

```
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

PowerShell

최대 절전 모드로 전환된 인스턴스 시작

Start-EC2Instance cmdlet을 사용합니다.

```
Start-EC2Instance -InstanceId i-1234567890abcdef0
```

Amazon EC2 인스턴스 최대 절전 모드 문제 해결

이 정보를 사용하여 인스턴스를 최대 절전 모드로 전환할 때 발생할 수 있는 문제를 진단 및 수정합니다.

최대 절전 모드 문제

- 시작 직후 최대 절전 모드로 전환할 수 없음
- stopping에서 stopped로 전환하는 데 너무 오래 걸리고 시작 후 메모리 상태가 복원되지 않음
- 인스턴스가 stopping 상태에 멈춰 있음
- 최대 절전 모드로 전환 후 즉시 스팟 인스턴스를 시작할 수 없음

- 스팟 인스턴스 재개 실패

시작 직후 최대 절전 모드로 전환할 수 없음

인스턴스를 시작한 후 너무 빨리 인스턴스를 최대 절전 모드로 전환하려고 하면 오류 메시지가 표시됩니다.

시작 후 Linux 인스턴스는 약 2분, Windows 인스턴스는 약 5분을 기다려야 최대 절전 모드로 전환할 수 있습니다.

stopping에서 stopped로 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.

Linux 인스턴스

인스턴스 시스템 로그를 확인하고 최대 절전 모드와 관련된 메시지를 살펴보세요. 시스템 로그에 액세스하려면 인스턴스에 연결하거나 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. 지원 AMI에 대한 내용은 EC2 인스턴스 최대 절전 모드를 위한 사전 조건 단원을 참조하세요. 자체 Linux AMI를 사용하는 경우 최대 절전 모드를 지원하도록 Linux AMI 구성의 지침을 따라야 합니다.

Windows Server 2016 이상

EC2 시작 로그를 확인하고 최대 절전 모드와 관련된 메시지를 살펴보세요. EC2 시작 로그에 액세스하려면, 인스턴스에 연결하고 텍스트 편집기에서 `C:\ProgramData\Amazon\EC2-Windows\Launch\Log\Ec2Launch.log` 파일을 엽니다. EC2Launch v2를 사용하는 경우 `C:\ProgramData\Amazon\EC2Launch\log\agent.log`를 엽니다.

> **Note**
>
> 기본적으로 Windows는 파일과 폴더를 `C:\ProgramData` 아래에 숨깁니다. EC2 디렉터리와 파일을 보려면 Windows 탐색기에 경로를 입력하거나 숨겨진 파일과 폴더를 표시하도록 폴더 속성을 변경합니다.

최대 절전 모드에 대한 로그 줄을 찾습니다. 긴 줄에 실패라고 표시되거나 로그 줄이 없는 경우 시작 시 최대 절전 모드 구성에 실패했을 가능성이 큽니다.

예를 들어, 다음 메시지는 최대 절전 모드를 구성하지 못했음을 나타냅니다. `Message: Failed to enable hibernation.` 오류 메시지에 십진수 ASCII 값이 포함된 경우 전체 오류 메시지를 읽으려면 ASCII 값을 일반 텍스트로 변환합니다.

의 로그 줄이 `HibernationEnabled: true`를 포함하면 최대 절전 모드가 성공적으로 구성된 것입니다.

Windows Server 2012 R2 및 이전

EC2 구성 로그를 확인하고 최대 절전 모드와 관련된 메시지를 살펴보세요. EC2 구성 로그에 액세스하려면, 인스턴스에 연결하고 텍스트 편집기에서 `C:\Program Files\Amazon\Ec2ConfigService\Logs\Ec2ConfigLog.txt` 파일을 엽니다. `SetHibernateOnSleep`에 대한 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. `SetHibernateOnSleep: Failed to enable hibernation: Hibernation failed with the following error: There is not enough space on the disk.`

의 로그 줄이 `SetHibernateOnSleep: HibernationEnabled: true`이면 최대 절전 모드가 성공적으로 구성된 것입니다.

Windows 인스턴스 크기

RAM이 1GiB 미만인 T3 또는 T3a Windows 인스턴스를 사용하는 경우 인스턴스의 크기를 RAM이 1GiB 이상인 인스턴스로 늘려보세요.

인스턴스가 stopping 상태에 멈춰 있음

인스턴스를 최대 절전 모드로 전환했는데 `stopping` 상태에 "멈춰" 있으면 강제로 중지할 수 있습니다. 자세한 내용은 Amazon EC2 인스턴스 중지 문제 해결 섹션을 참조하세요.

최대 절전 모드로 전환 후 즉시 스팟 인스턴스를 시작할 수 없음

스팟 인스턴스를 최대 절전 모드로 전환한 후 2분 이내에 시작하려고 하면 다음 오류가 발생할 수 있습니다.

```
You failed to start the Spot Instance because the associated Spot Instance request is not in an appropriate state to support start.
```

Linux 인스턴스는 약 2분, Windows 인스턴스는 약 5을 기다린 후 인스턴스를 다시 시작하세요.

스팟 인스턴스 재개 실패

스팟 인스턴스가 성공적으로 최대 절전 모드로 전환되었지만 재개에 실패하고 대신 재부팅된 경우(최대 절전 모드로 전환된 상태가 유지되지 않는 새로 다시 시작) 사용자 데이터에 다음 스크립트가 포함되어 있기 때문일 수 있습니다.

```
/usr/bin/enable-ec2-spot-hibernation
```

시작 템플릿의 사용자 데이터 필드에서 이 스크립트를 제거한 다음 새 스팟 인스턴스를 요청합니다.

단, 인스턴스가 재개되지 않더라도 최대 절전 모드로 전환된 상태가 유지되지 않으면 인스턴스를 `stopped` 상태에서 시작하는 것과 같은 방식으로 시작할 수 있습니다.

Amazon EC2 인스턴스 재부팅

인스턴스 재부팅은 운영 체제 재부팅과 같습니다. 대부분의 경우 인스턴스를 재부팅하는 데는 몇 분 밖에 걸리지 않습니다.

인스턴스를 재부팅할 때 다음이 유지됩니다.

- 퍼블릭 DNS 이름(IPv4)
- 프라이빗 IPv4 주소
- 퍼블릭 IPv4 주소
- IPv6 주소(해당하는 경우)
- 인스턴스 스토어 볼륨의 모든 데이터

인스턴스를 중지했다가 다시 시작할 때(최소 1분 요금을 포함하는 새 청구 기간이 시작됨)와는 달리, 인스턴스를 재부팅해도 새 인스턴스 청구 기간이 시작되지 않습니다.

인스턴스 재부팅은 사용자가 시작하거나(인스턴스를 수동으로 재부팅하는 경우) AWS에 의해 시작될 수 있습니다(자동 인스턴스 복구를 위해 또는 재부팅이 필요한 업데이트를 적용하는 등 필요한 유지 관리를 위해 예약된 재부팅 이벤트에 대한 응답으로).

사용자가 시작한 재부팅의 경우 인스턴스에서 운영 체제 재부팅 명령을 실행하는 대신 Amazon EC2 콘솔, CLI 또는 API를 사용하는 것이 좋습니다. Amazon EC2를 사용하는 경우 인스턴스가 몇 분 안에 완전히 종료되지 않으면 Amazon EC2가 하드 재부팅을 수행합니다. 또한 AWS CloudTrail에서는 인스턴스가 재부팅된 시점의 API 레코드를 생성합니다.

이 주제에서는 사용자가 시작하는 재부팅을 수행하는 방법을 설명합니다. AWS에서 수행하는 재부팅에 대한 자세한 내용은 자동 인스턴스 복구 및 재부팅하도록 예약된 Amazon EC2 인스턴스 관리 섹션을 참조하세요.

Windows 인스턴스

Windows가 인스턴스에 업데이트를 설치할 경우 모든 업데이트가 설치될 때까지는 Amazon EC2 콘솔 또는 명령줄을 사용하여 인스턴스를 재부팅하거나 종료하지 않는 것이 좋습니다. Amazon EC2 콘솔 또는 명령줄을 사용하여 인스턴스를 재부팅하거나 종료할 경우 인스턴스가 하드 재부팅할 위험이 있습니다. 업데이트가 설치되는 도중의 하드 재부팅은 인스턴스 상태를 불안정하게 만들 수 있습니다.

Console

인스턴스 재부팅

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스(Instances)를 선택합니다.
3. 인스턴스를 선택하고 인스턴스 상태(Instance state), 인스턴스 재부팅(Reboot instance)을 차례로 선택합니다.

   또는 인스턴스를 선택한 다음 작업(Actions), 인스턴스 상태 관리(Manage instance state)를 선택합니다. 열리는 화면에서 재부팅(Reboot) 및 상태 변경(Change state)을 차례로 선택합니다.
4. 확인 메시지가 표시되면 재부팅(Reboot)을 선택합니다.

   인스턴스는 `running` 상태로 유지됩니다.

AWS CLI

인스턴스 재부팅

reboot-instances 명령을 사용합니다.

```
aws ec2 reboot-instances --instance-ids i-1234567890abcdef0
```

PowerShell

인스턴스 재부팅

Restart-EC2Instance cmdlet을 사용합니다.

```
Restart-EC2Instance -InstanceId i-1234567890abcdef0
```

통제된 오류 주입 실험을 실행하려면

AWS Fault Injection Service를 사용하여 인스턴스가 재부팅될 때 애플리케이션이 어떻게 반응하는지 테스트할 수 있습니다. 자세한 내용은 AWS Fault Injection Service 사용 설명서를 참조하세요.

Amazon EC2 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 새 인스턴스를 실행할 수 있습니다.

대신 인스턴스를 중지하거나 최대 절전 모드로 전환하는 경우 Amazon EC2 인스턴스 중지 및 시작 또는 Amazon EC2 인스턴스를 최대 절전 모드로 전환 섹션을 참조하세요. 자세한 내용은 인스턴스 상태 간 차이 섹션을 참조하세요.

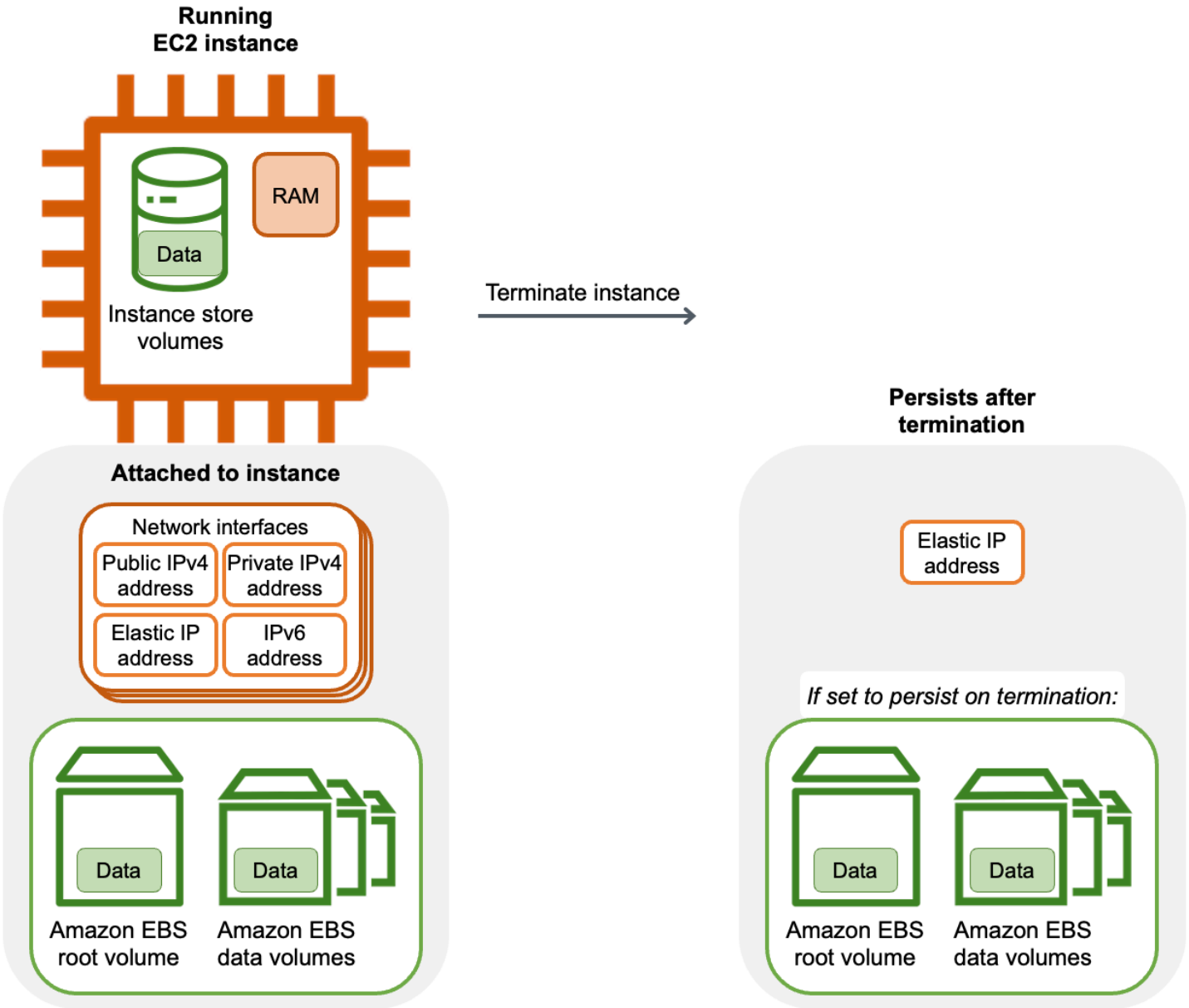
내용

- 인스턴스 종료 작동 방식
- 인스턴스 종료
- 인스턴스 종료 문제 해결
- 인스턴스 종료 방지 변경
- 인스턴스가 시작하는 종료 동작 변경
- 인스턴스가 종료될 때 데이터 보존

인스턴스 종료 작동 방식

인스턴스를 종료하면 변경 내용이 인스턴스의 운영 체제(OS) 수준에 등록되며 일부 리소스는 손실되고 일부는 지속됩니다.

다음 다이어그램은 Amazon EC2 인스턴스가 종료될 때 손실되는 항목과 지속되는 항목을 나타냅니다. 인스턴스가 종료되면 인스턴스 저장소 볼륨의 데이터와 인스턴스 RAM에 저장된 데이터가 지워집니다. 인스턴스와 연결된 탄력적 IP 주소가 분리됩니다. Amazon EBS 루트 볼륨과 데이터 볼륨의 경우 결과는 각 볼륨의 종료 시 삭제 설정에 따라 달라집니다.



고려 사항

- 데이터 지속성
  - 인스턴스 저장소 볼륨: 인스턴스가 종료될 때 모든 데이터가 영구적으로 삭제됩니다.
  - EBS 루트 볼륨
    - 시작 시 연결된 경우 인스턴스가 종료될 때 기본적으로 삭제됩니다.
    - 시작 후 연결된 경우 인스턴스가 종료될 때 기본적으로 유지됩니다.
  - EBS 데이터 볼륨:
    - 콘솔을 사용하여 시작 시 연결된 경우: 인스턴스가 종료될 때 기본적으로 유지됩니다.
    - CLI를 사용하여 시작 시 연결된 경우: 인스턴스가 종료될 때 기본적으로 삭제됩니다.
    - 콘솔 또는 CLI를 사용하여 시작 후 연결된 경우: 인스턴스가 종료될 때 기본적으로 유지됩니다.

> **Note**
>
> 인스턴스 종료 시 삭제되지 않은 볼륨에는 계속 요금이 부과됩니다. 인스턴스 종료 시 볼륨이 삭제되거나 유지되도록 설정을 변경할 수 있습니다. 자세한 내용은 인스턴스가 종료될 때 데이터 보존 섹션을 참조하세요.

- 우발적인 종료로부터 보호
  - 인스턴스를 실수로 종료하지 않도록 하기 위해 인스턴스에 대한 종료 방지를 활성화합니다.
  - 인스턴스에서 종료를 시작할 때 인스턴스의 중지 또는 종료 여부를 제어하려면 인스턴스 시작 종료 동작을 변경합니다.
- 종료 스크립트 - 인스턴스 종료에 대한 스크립트를 실행하는 경우 종료 스크립트의 실행을 보장할 방법이 없기 때문에 인스턴스가 비정상적으로 종료될 수 있습니다. Amazon EC2는 인스턴스를 완전히 종료하고 시스템 종료 스크립트를 실행하려고 시도합니다. 그러나 하드웨어 장애와 같은 특정 이벤트로 인해 이러한 시스템 종료 스크립트가 실행되지 않을 수 있습니다.
- 베어 메탈 인스턴스 - x86 베어 메탈 인스턴스는 협력적 종료를 지원하지 않습니다.

인스턴스 종료 시 발생하는 상황

OS 수준에서 등록된 변경 내용

- API 요청은 버튼 누름 이벤트를 게스트로 전송합니다.

- 버튼 누름 이벤트로 인해 다양한 시스템 서비스가 중지됩니다. 시스템의 정상 종료는 systemd(Linux) 또는 시스템 프로세스(Windows)에 의해 제공됩니다. 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. 인스턴스는 잠시 동안 콘솔에 표시되며 그 이후 항목이 자동으로 삭제됩니다.

리소스 손실

- 인스턴스 스토어 볼륨에 저장된 데이터.
- `DeleteOnTermination` 속성이 `true`로 설정된 경우 EBS 루트 볼륨
- `DeleteOnTermination` 속성이 `true`로 설정된 경우 EBS 데이터 볼륨(시작 시 또는 시작 후 연결됨)

지속되는 리소스

- `DeleteOnTermination` 속성이 `false`로 설정된 경우 EBS 루트 볼륨
- `DeleteOnTermination` 속성이 `false`로 설정된 경우 EBS 데이터 볼륨(시작 시 또는 시작 후 연결됨)

인스턴스 종료에 대한 애플리케이션 응답 테스트

AWS Fault Injection Service를 사용하여 인스턴스가 종료될 때 애플리케이션이 어떻게 반응하는지 테스트할 수 있습니다. 자세한 내용은 AWS Fault Injection Service 사용 설명서를 참조하세요.

인스턴스 종료

언제든지 인스턴스를 종료할 수 있습니다.

Console

인스턴스를 종료하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 인스턴스 상태, 인스턴스 종료(삭제)를 선택합니다.
4. 확인 메시지가 나타나면 종료(삭제)를 선택합니다.

5. 인스턴스를 종료한 후에도 인스턴스는 잠깐 동안 terminated의 상태로 표시됩니다.

   종료에 실패하거나 종료된 인스턴스가 몇 시간 이상 표시될 경우 종료된 인스턴스가 계속 표시됨 섹션을 참조하세요.

AWS CLI

인스턴스를 종료하려면

terminate-instances 명령을 사용합니다.

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0
```

PowerShell

인스턴스를 종료하려면

Remove-EC2Instance cmdlet을 사용합니다.

```
Remove-EC2Instance -InstanceId i-1234567890abcdef0
```

인스턴스 종료 문제 해결

요청자는 ec2:TerminateInstances를 직접 호출할 수 있는 권한이 있어야 합니다. 자세한 내용은 인스턴스 작업을 위한 정책 예제를 참조하세요.

인스턴스를 종료했을 때 다른 인스턴스가 시작될 경우 EC2 집합 또는 Amazon EC2 Auto Scaling 같은 기능을 통해 자동 조정을 구성했을 가능성이 큽니다. 자세한 내용은 인스턴스가 자동으로 시작되거나 종료됨 섹션을 참조하세요.

종료 방지가 활성화된 경우 인스턴스를 종료할 수 없습니다. 자세한 내용은 종료 보호 섹션을 참조하세요.

인스턴스가 shutting-down 상태에 일반적인 경우보다 장기간 머물러 있는 경우, 해당 인스턴스는 Amazon EC2 서비스 내 자동화된 과정에 의해 클린업(종료)됩니다. 자세한 내용은 지연된 인스턴스 종료 섹션을 참조하세요.

인스턴스 종료 방지 변경

TerminateInstances를 직접적으로 호출하거나 Amazon EC2 콘솔과 같은 다른 인터페이스를 사용하거나 관계없이 Amazon EC2 API를 사용하여 인스턴스가 실수로 종료되는 것을 방지하려면 인스턴

스에 대한 종료 방지를 활성화합니다. `DisableApiTermination` 속성에서는 인스턴스 종료 여부를 제어합니다. 기본 설정상 인스턴스에 대한 종료 보호 기능은 비활성화되어 있습니다. 인스턴스를 시작할 때 또는 인스턴스가 실행 중이거나 인스턴스가 중지된 동안 이 속성의 값을 설정할 수 있습니다.

`DisableApiTermination` 속성은 `InstanceInitiatedShutdownBehavior` 속성이 `terminate`로 설정된 경우 인스턴스에서 종료를 시작하는 방식(예: 시스템 종료를 위해 운영 체제 명령 사용)으로 인스턴스 종료를 방지하지 않습니다. 자세한 내용은 인스턴스가 시작하는 종료 동작 변경 섹션을 참조하세요.

고려 사항

- 종료 방지를 활성화해도 인스턴스를 종료하는 예약 이벤트가 있는 경우 AWS에서 인스턴스 종료를 방지하지 않습니다.
- 종료 방지를 활성화해도 인스턴스가 비정상일 때 또는 스케일 인 이벤트 중에 Amazon EC2 Auto Scaling에서 인스턴스를 종료합니다. 인스턴스 스케일 인 보호를 사용하여 스케일 인할 때 Auto Scaling이 특정 인스턴스를 종료할 수 있는지 여부를 제어할 수 있습니다. ReplaceUnhealthy 조정 프로세스를 일시 중지하여 Auto Scaling에서 비정상 인스턴스의 종료 여부를 제어할 수 있습니다.
- 스팟 인스턴스에 대한 종료 방지 기능은 활성화할 수 없습니다.

Console

시작 시 인스턴스에 대한 종료 방지를 활성화하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 대시보드에서 인스턴스 시작을 선택합니다.
3. Advanced details(고급 세부 정보)를 확장합니다. 종료 방지에서 활성화를 선택합니다.
4. 인스턴스에 대한 세부 정보 지정을 마치면 인스턴스 시작을 선택합니다.

인스턴스에 대한 종료 방지를 변경하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [인스턴스(Instances)]를 선택합니다.
3. 인스턴스를 선택합니다.
4. 작업, 인스턴스 설정, 종료 보호 변경을 선택합니다.
5. 종료 방지에서 활성화를 선택하거나 선택 취소합니다.
6. 저장을 선택합니다.

AWS CLI

인스턴스에 대한 종료 방지를 활성화하는 방법

modify-instance-attribute 명령을 사용합니다.

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --disable-api-termination
```

인스턴스에 대한 종료 방지를 비활성화하려면

modify-instance-attribute 명령을 사용합니다.

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --no-disable-api-termination
```

PowerShell

인스턴스에 대한 종료 방지를 활성화하는 방법

Edit-EC2InstanceAttribute cmdlet을 사용합니다.

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -DisableApiTermination $true
```

인스턴스에 대한 종료 방지를 비활성화하려면

Edit-EC2InstanceAttribute cmdlet을 사용합니다.

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -DisableApiTermination $false
```

종료 방지를 사용하여 여러 인스턴스 종료

동일한 요청에서 여러 가용 영역에 있는 여러 인스턴스를 종료할 때 지정한 인스턴스 중 하나 이상에 종료 방지가 활성화된 경우 요청이 실패하고 다음과 같은 결과가 발생합니다.

- 보호된 인스턴스와 동일한 가용 영역에 있는 지정된 인스턴스가 종료되지 않습니다.
- 지정된 인스턴스 중 보호된 인스턴스가 없는 다른 가용 영역의 지정된 인스턴스는 성공적으로 종료됩니다.

예제

두 가용 영역에 다음 네 개의 인스턴스가 있다고 가정합니다.

Instance	가용 영역	종료 방지
인스턴스 1	AZ A	Disabled
인스턴스 2		Disabled
인스턴스 3	AZ B	Enabled
인스턴스 4		Disabled

동일한 요청에서 이러한 모든 인스턴스를 종료하려고 하면 요청이 실패하고 다음과 같은 결과가 나타납니다.

- 인스턴스 1 및 인스턴스 2는 두 인스턴스에서 종료 방지가 활성화되지 않았기 때문에 종료됩니다.
- 인스턴스 3 및 인스턴스 4는 인스턴스 3에서 종료 방지가 활성화되었기 때문에 종료되지 않습니다.

인스턴스가 시작하는 종료 동작 변경

기본적으로 shutdown 또는 poweroff 등의 명령을 사용하여 Amazon EBS 지원 인스턴스에서 종료를 시작하면 인스턴스가 중지됩니다. 인스턴스에 대한 `InstanceInitiatedShutdownBehavior` 속성을 변경하여 대신 이 인스턴스스가 종료되도록 이 동작을 변경할 수 있습니다. 인스턴스가 실행 중이거나 중단된 상태에 있을 때 이 속성을 변경할 수 있습니다.

halt 명령은 종료를 시작하지 않습니다. 이 기능을 사용하는 경우 인스턴스가 종료되지 않고, 대신 CPU를 HLT 상태로 두고 인스턴스는 계속 실행됩니다.

> **Note**
>
> `InstanceInitiatedShutdownBehavior` 속성은 인스턴스 자체의 운영 체제를 종료를 수행하는 경우에만 적용됩니다. `StopInstances` API 또는 Amazon EC2 콘솔을 사용하는 인스턴스를 중지하는 경우에는 적용되지 않습니다.

Console

인스턴스 시작 종료 동작 변경 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다.
4. 작업, 인스턴스 설정, 종료 동작 변경을 차례로 선택합니다.

   종료 동작은 현재 동작을 표시합니다.
5. 동작을 변경하려면 종료 동작에서 중지 또는 종료를 선택합니다.
6. 저장을 선택합니다.

AWS CLI

인스턴스 시작 종료 동작 변경 방법

modify-instance-attribute 명령을 사용합니다.

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --instance-initiated-shutdown-behavior terminate
```

PowerShell

인스턴스 시작 종료 동작 변경 방법

Edit-EC2InstanceAttribute cmdlet을 사용합니다.

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -InstanceInitiatedShutdownBehavior terminate
```

인스턴스가 종료될 때 데이터 보존

Amazon EC2 인스턴스가 종료될 때 인스턴스 저장소 볼륨 또는 Amazon EBS 볼륨의 데이터를 보존할 수 있습니다. 이 주제에서는 인스턴스 종료 이후에도 데이터가 유지되도록 하는 방법을 설명합니다.

인스턴스 종료가 루트 및 데이터 볼륨에 미치는 영향

인스턴스 스토어 볼륨

인스턴스가 종료될 때 인스턴스 저장소 볼륨이 자동으로 삭제되고 데이터가 손실됩니다. 인스턴스 수명 이후에도 이 데이터를 보존하려면 인스턴스를 종료하기 전에 Amazon EBS 볼륨, Amazon S3 버킷 또는 Amazon EFS 파일 시스템과 같은 영구적인 스토리지에 해당 데이터를 수동으로 복사합니다. 자세한 내용은 Amazon EC2 인스턴스의 스토리지 옵션 섹션을 참조하세요.

Amazon EBS 볼륨

인스턴스가 종료될 때 각 볼륨에 대한 `DeleteOnTermination` 속성 값에 따라 EBS 볼륨이 삭제되거나 유지됩니다.

- 예(콘솔)/`true`(CLI) - 인스턴스가 종료될 때 볼륨이 삭제됩니다.
- 아니요(콘솔)/`false`(CLI) - 인스턴스가 종료될 때 볼륨이 유지됩니다. 보존된 볼륨에는 계속 요금이 부과됩니다.

> **Note**
>
> 인스턴스가 종료된 후에 유지된 볼륨의 스냅샷을 만들거나 다른 인스턴스에 연결할 수 있습니다. 요금이 발생하지 않도록 하려면 볼륨을 삭제해야 합니다.

EBS 볼륨의 기본 삭제 동작

기본 `DeleteOnTermination` 값은 볼륨 유형, 볼륨이 시작 시 연결되었는지 또는 시작 후 연결되었는지, 볼륨을 연결하는 데 사용된 방법(콘솔 또는 CLI)에 따라 달라집니다.

볼륨 유형	연결 시기	연결 방법	인스턴스 종료 시 기본 동작
루트 볼륨	시작 시	콘솔 또는 CLI	삭제
루트 볼륨	시작한 후	콘솔 또는 CLI	Preserve

볼륨 유형	연결 시기	연결 방법	인스턴스 종료 시 기본 동작
데이터 볼륨	시작 시	콘솔	Preserve
데이터 볼륨	시작 시	CLI	삭제
데이터 볼륨	시작한 후	콘솔 및 CLI	Preserve

볼륨 지속성 설정 확인

EBS 볼륨의 시작 시 기본값은 AMI에 설정된 `DeleteOnTermination` 속성에 의해 결정됩니다. 인스턴스 시작 시 값을 변경하여 AMI 설정을 재정의할 수 있습니다. 인스턴스를 시작한 후에는 `DeleteOnTermination` 속성에 대한 기본 설정을 확인하는 것이 좋습니다.

인스턴스 종료 시 Amazon EBS 볼륨이 삭제되는지 확인하려면 다음을 수행하세요.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다.
4. 스토리지 탭을 선택합니다.
5. 블록 디바이스에서 오른쪽으로 스크롤하여 종료 시 삭제 열을 확인합니다.
   - 예를 선택하면 인스턴스가 종료될 때 볼륨이 삭제됩니다.
   - 아니요를 선택하면 인스턴스가 종료될 때 볼륨이 삭제되지 않습니다. 삭제되지 않은 볼륨에는 계속 요금이 부과됩니다.

시작 시 루트 볼륨을 지속하도록 변경

인스턴스를 시작할 때 EBS 루트 볼륨의 `DeleteOnTermination` 속성을 변경할 수 있습니다. 데이터 볼륨에 대해 다음 절차를 사용할 수도 있습니다.

Console

실행 시 인스턴스의 루트 볼륨이 유지되도록 변경하려면 다음을 수행하세요.

1. 절차에 따라 인스턴스를 시작하고 다음 단계를 완료하여 지속하도록 루트 볼륨을 변경한 후에만 인스턴스를 시작합니다.

2. 스토리지 구성 창에서 고급을 선택합니다.
3. EBS 볼륨에서 루트 볼륨 정보를 확장합니다.
4. 종료 시 삭제에서 예를 선택합니다.
5. Summary(요약) 패널에서 인스턴스 구성을 검토한 다음 Launch instance(인스턴스 시작)를 선택합니다. 자세한 내용은 콘솔에서 인스턴스 시작 마법사를 사용하여 EC2 인스턴스 시작 섹션을 참조하세요.

AWS CLI

실행 시 인스턴스의 루트 볼륨이 유지되도록 변경하려면 다음을 수행하세요.

run-instances 명령을 사용하여 블록 디바이스 매핑에서 `DeleteOnTermination` 값을 변경합니다.

`--block-device-mappings` 옵션을 추가합니다.

```
--block-device-mappings file://mapping.json
```

`mapping.json`에서 디바이스 이름(예: `/dev/sda1` 또는 `/dev/xvda`)을 지정하고 `DeleteOnTermination`에 대해 `false`를 지정합니다.

```
[
  {
    "DeviceName": "device_name",
    "Ebs": {
      "DeleteOnTermination": false
    }
  }
]
```

PowerShell

실행 시 인스턴스의 루트 볼륨이 유지되도록 변경하려면 다음을 수행하세요.

New-EC2Instance cmdlet을 사용하여 블록 디바이스 매핑에서 `DeleteOnTermination` 값을 변경합니다.

`-BlockDeviceMapping` 옵션을 추가합니다.

```
-BlockDeviceMapping $bdm
```

bdm에서 디바이스 이름(예: /dev/sda1 또는 /dev/xvda)을 지정하고 DeleteOnTermination에 대해 false를 지정합니다.

```
$ebd = New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice
$ebd.DeleteOnTermination = false
$bdm = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$bdm.DeviceName = "/dev/sda1"
$bdm.Ebs = $ebd
```

실행 중인 인스턴스의 루트 볼륨이 지속되도록 변경

실행 중인 인스턴스의 EBS 루트 볼륨이 지속되도록 변경할 수 있습니다. 데이터 볼륨에 대해 다음 절차를 사용할 수도 있습니다.

AWS CLI

루트 볼륨이 지속되도록 변경하려면 다음을 수행하세요.

modify-instance-attribute 명령을 사용합니다.

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

mapping.json에서 디바이스 이름(예: /dev/sda1 또는 /dev/xvda)을 지정하고 --DeleteOnTermination에 대해 false를 지정합니다.

```
[
  {
    "DeviceName": "device_name",
    "Ebs": {
      "DeleteOnTermination": false
    }
  }
]
```

PowerShell

루트 볼륨이 지속되도록 변경하려면 다음을 수행하세요.

Edit-EC2InstanceAttribute cmdlet을 사용합니다.

`-BlockDeviceMapping` 옵션을 추가합니다.

```
-BlockDeviceMapping $bdm
```

bdm에서 디바이스 이름(예: `/dev/sda1` 또는 `/dev/xvda`)을 지정하고 `DeleteOnTermination`에 대해 `false`를 지정합니다.

```
$ebd = New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice
$ebd.DeleteOnTermination = false
$bdm = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$bdm.DeviceName = "/dev/sda1"
$bdm.Ebs = $ebd
```

인스턴스 만료

AWS에서 인스턴스를 호스팅하는 기본 하드웨어의 복구 불가능한 장애가 검색되는 경우 인스턴스가 만료 대상으로 예약됩니다. 인스턴스 루트 디바이스는 인스턴스 폐기 동작을 결정합니다.

- 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 인스턴스는 종료되며 다시 사용할 수 없습니다.

인스턴스 이벤트 유형에 대한 자세한 내용은 Amazon EC2 인스턴스의 예약된 이벤트 섹션을 참조하세요.

목차

- 만료 예약된 인스턴스 식별
- 만료 예약된 EBS 지원 인스턴스에 대해 수행할 작업
- 만료 예약된 인스턴스 스토어 지원 인스턴스에 대해 수행할 작업

만료 예약된 인스턴스 식별

인스턴스에 대한 만료가 예약되어 있는 경우 만료 이벤트가 발생하기 전에 인스턴스 ID와 만료 날짜가 포함된 이메일이 수신됩니다. 사용 중지될 예정인 인스턴스도 확인할 수 있습니다.

> **Important**
>
> 인스턴스가 사용 중지될 예정인 경우 이미 인스턴스에 연결하지 못할 수 있으므로 가능한 한 빨리 조치를 취하는 것이 좋습니다. 자세한 내용은 Check if your instance is reachable 섹션을 참조하세요.

사용 중지될 예정인 인스턴스를 식별하는 옵션

- 계정 연락처에 대한 이메일 모니터링
- 인스턴스 확인

계정 연락처에 대한 이메일 모니터링

인스턴스가 사용 중지될 예정인 경우 이벤트 전에 계정의 기본 연락처와 운영 연락처에 이메일이 수신됩니다. 이 이메일에는 인스턴스 ID와 예정된 사용 중지 날짜가 포함됩니다. 자세한 내용은 AWS Account Management 참조 가이드의 AWS 계정의 기본 연락처 업데이트와 AWS 계정의 대체 연락처 업데이트를 참조하세요.

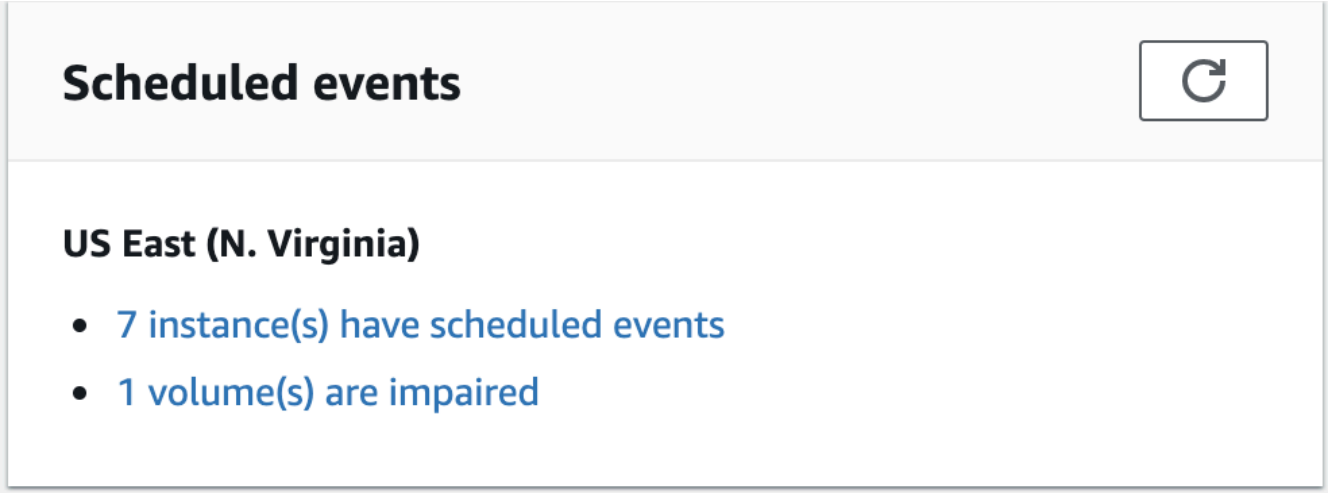
인스턴스 확인

정기적으로 확인하지 않는 이메일 계정을 사용하는 경우 인스턴스 사용 중지 알림을 놓칠 수 있습니다. 언제든지 인스턴스가 사용 중지될 예정인지 확인할 수 있습니다.

Console

사용 중지될 예정인 인스턴스를 식별하는 방법

1. https://console.aws.amazon.com/ec2/에서 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 이벤트가 연결되어 있는 모든 리소스가 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.

AWS CLI

사용 중지될 예정인 인스턴스를 찾는 방법

아래와 같이 describe-instance-status 명령을 사용합니다. 인스턴스를 실행 중인 각 리전에서 반복합니다.

```
aws ec2 describe-instance-status --filters Name=event.code,Values=instance-
retirement
```

PowerShell

사용 중지될 예정인 인스턴스를 찾는 방법

Get-EC2InstanceStatus cmdlet을 사용합니다. 인스턴스를 실행 중인 각 리전에서 반복합니다.

```
Get-EC2InstanceStatus -Filter @{Name="event.code"; Values="instance-retirement"}
```

만료 예약된 EBS 지원 인스턴스에 대해 수행할 작업

만료되는 인스턴스의 데이터를 보존하려면 다음 작업 중 하나를 수행할 수 있습니다. 예기치 않은 중단 시간 및 데이터 손실을 방지하려면 인스턴스 만료 날짜 전에 이 작업을 수행해야 합니다.

Linux 인스턴스의 경우 인스턴스가 EBS 또는 인스턴스 스토어에 의해 백업되는지 확실하지 않는다면 Amazon EC2 인스턴스용 루트 볼륨 섹션을 참조하세요.

인스턴스에 연결할 수 있는지 확인

인스턴스에 대한 만료가 예약되어 있다는 알림을 받으면 가능한 한 빨리 다음 작업을 수행하는 것이 좋습니다.

- 인스턴스에 연결하거나 인스턴스에 대해 ping을 실행하여 인스턴스에 연결할 수 있는지 확인합니다.
- 인스턴스에 연결할 수 있는 경우 예약된 만료 날짜 이전에 영향이 가장 적은 적합한 시간에 인스턴스를 중지/시작하도록 계획해야 합니다. 인스턴스 중지 및 시작과 인스턴스 중지 시 발생하는 결과(예: 인스턴스와 연결된 퍼블릭, 프라이빗 및 탄력적 IP 주소에 대한 영향)에 대한 자세한 내용은 Amazon EC2 인스턴스 중지 및 시작 섹션을 참조하세요. 인스턴스를 중지하고 시작하면 인스턴스 스토어 볼륨의 데이터가 손실됩니다.
- 인스턴스에 연결할 수 없는 경우 즉시 작업을 수행하고 중지/시작을 수행하여 인스턴스를 복구해야 합니다.
- 또는 인스턴스를 종료하려는 경우 가능한 한 빨리 종료하여 인스턴스에 대한 요금 발생을 중지할 수 있습니다.

인스턴스의 백업 생성

인스턴스에서 EBS 지원 AMI를 생성하면 백업이 생깁니다. 데이터 무결성을 보장하려면 AMI를 생성하기 전에 인스턴스를 중지합니다. 예약된 만료 날짜까지 기다리거나(인스턴스가 중지되는 경우), 만료 날짜 전에 인스턴스를 중지합니다. 언제든지 인스턴스를 시작할 수 있습니다. 자세한 내용은 Amazon EBS 지원 AMI 생성 섹션을 참조하세요.

대체 인스턴스 시작

인스턴스에서 AMI를 생성한 후 AMI를 사용하여 대체 인스턴스를 시작할 수 있습니다. Amazon EC2 콘솔에서 새 AMI를 선택한 다음 AMI로 인스턴스 시작을 선택합니다. 인스턴스의 파라미터를 구성한 다음 인스턴스 시작을 선택합니다. 각 필드에 대한 자세한 내용은 콘솔에서 인스턴스 시작 마법사를 사용하여 EC2 인스턴스 시작 섹션을 참조하세요.

만료 예약된 인스턴스 스토어 지원 인스턴스에 대해 수행할 작업

만료되는 인스턴스의 데이터를 보존하려면 다음 작업 중 하나를 수행할 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 관계없이, 볼륨이 EBS 기반 인스턴스에 연결되어 있더라도 인스턴스가 만료되면 인스턴스 스토어 볼륨의 데이터는 손실됩니다.

인스턴스에 연결할 수 있는지 확인

인스턴스에 대한 만료가 예약되어 있다는 알림을 받으면 가능한 한 빨리 다음 작업을 수행하는 것이 좋습니다.

- 인스턴스에 연결하거나 인스턴스에 대해 ping을 실행하여 인스턴스에 연결할 수 있는지 확인합니다.
- 인스턴스에 연결할 수 없는 경우 인스턴스를 복구하기 위해 수행할 수 있는 작업이 거의 없을 수 있습니다. 자세한 내용은 연결할 수 없는 Amazon EC2 인스턴스 문제 해결 단원을 참조하세요. AWS은(는) 예약된 만료 날짜에 인스턴스를 종료하므로, 연결할 수 없는 인스턴스의 경우 인스턴스를 즉시 종료할 수 있습니다.

대체 인스턴스 시작

인스턴스 스토어 지원 AMI 생성의 설명에 따라 AMI 도구를 사용하여 인스턴스에서 인스턴스 스토어 지원 AMI를 생성합니다.. Amazon EC2 콘솔에서 새 AMI를 선택한 다음 AMI로 인스턴스 시작을 선택합니다. 인스턴스의 파라미터를 구성한 다음 인스턴스 시작을 선택합니다. 각 필드에 대한 자세한 내용은 콘솔에서 인스턴스 시작 마법사를 사용하여 EC2 인스턴스 시작 섹션을 참조하세요.

인스턴스를 EBS 지원 인스턴스로 변환

데이터를 EBS 볼륨에 전송하고, 볼륨의 스냅샷을 만든 후 스냅샷에서 AMI를 생성합니다. 새 AMI에서 대체 인스턴스를 시작할 수 있습니다. 자세한 내용은 EBS 지원 AMI로 인스턴스 스토어 지원 AMI 변환 섹션을 참조하세요.

자동 인스턴스 복구

> **Important**
>
> 이 섹션에서는 EC2 인스턴스에서 복구 메커니즘을 사전에 구성하는 방법을 설명합니다. 이러한 복구 메커니즘은 AWS가 시스템 상태 확인 실패를 유발하는 기본 하드웨어 또는 소프트웨어 결함을 감지할 때 인스턴스 가용성을 복원하도록 설계되었습니다. 현재 인스턴스에 액세스하는 데 문제가 있는 경우 EC2 인스턴스 문제 해결을 참조하세요.

AWS에서 기본 하드웨어 또는 소프트웨어 문제로 인해 인스턴스를 사용할 수 없다고 판단하는 경우 인스턴스 가용성을 자동으로 복원할 수 있는 두 가지 메커니즘, 즉 간소화된 자동 복구와 Amazon CloudWatch 작업 기반 복구가 있습니다. 인스턴스 가용성 복구는 인스턴스 복구라고도 합니다.

인스턴스 복구 프로세스 중에 AWS는 기본 하드웨어 또는 소프트웨어 문제가 있는 호스트에서 다른 호스트로 인스턴스를 이동하려고 시도합니다. 인스턴스 복구 프로세스가 성공하면 인스턴스가 계획되지 않은 재부팅을 한 것으로 표시됩니다. 인스턴스 복구가 발생했는지 확인할 수 있습니다.

복구 프로세스가 실패하면 기본 하드웨어 또는 소프트웨어 문제가 있는 호스트에서 인스턴스가 계속 실행될 수 있습니다. 이 경우 수동 개입이 필요합니다. 인스턴스에 더 이상 연결할 수 없거나 인스턴스 시스템 상태 확인 실패가 계속되는 경우 인스턴스를 수동으로 중지 및 시작하는 것이 좋습니다. 인스턴스를 시작하면 일반적으로 새 기본 호스트 컴퓨터로 마이그레이션됩니다. 그러나 인스턴스가 퍼블릭 IPv4 주소를 유지하는 자동 인스턴스 복구와 달리 다시 시작된 인스턴스는 탄력적 IP 주소가 없는 한 새 퍼블릭 IPv4 주소를 수신합니다.

자동 복구 메커니즘의 이점을 활용하려면 시스템 상태 확인이 실패하기 전에 인스턴스에서 미리 구성해야 합니다. 기본적으로 인스턴스를 시작하는 동안 해당 인스턴스에 대해 간소화된 자동 복구가 활성화됩니다. 선택적으로 시작 후 Amazon CloudWatch 작업 기반 복구를 구성할 수 있습니다. 이러한 메커니즘 중 하나를 구성하면 인스턴스의 복원력이 향상됩니다.

간소화된 자동 복구 및 Amazon CloudWatch 작업 기반 복구는 지원되는 인스턴스에서만 사용할 수 있습니다. 자세한 내용은 간소화된 자동 복구 활성화 요구 사항 및 CloudWatch 작업 기반 복구 활성화에 대한 요구 사항(을)를 참조하세요.

> **Warning**
>
> AWS가 기본 하드웨어 또는 소프트웨어 문제로 인해 인스턴스를 복구하는 경우 휘발성 메모리(RAM)에 저장된 데이터가 손실되고 운영 체제의 가동 시간이 0에서 다시 시작됩니다. 또한 CloudWatch 작업 기반 복구를 사용하면 인스턴스 저장소 볼륨의 데이터도 손실됩니다. 데이터

손실을 방지하려면 중요한 데이터의 백업을 정기적으로 생성하는 것이 좋습니다. EC2 인스턴스의 백업 및 복구 모범 사례에 대한 자세한 내용은 Amazon EC2 모범 사례를 참조하세요. 자동 인스턴스 복구 메커니즘은 개별 인스턴스에 맞게 설계되었습니다. 복원력이 높은 시스템 구축에 대한 지침은 복원력 있는 시스템 구축 섹션을 참조하세요.

주제

- 자동 인스턴스 복구의 주요 개념
- 간소화된 자동 복구와 CloudWatch 작업 기반 복구의 차이점
- 복원력 있는 시스템 구축
- 자동 인스턴스 복구가 발생했는지 확인
- Amazon EC2 인스턴스에서 간소화된 자동 복구 구성
- EC2 인스턴스에서 CloudWatch 작업 기반 복구 구성

자동 인스턴스 복구의 주요 개념

자동 인스턴스 복구는 기본 하드웨어 또는 소프트웨어 장애가 발생할 때 인스턴스 가용성을 자동으로 복원하여 EC2 인스턴스의 복원력과 신뢰성을 높이는 Amazon EC2 기능입니다.

다음은 자동 인스턴스 복구의 주요 개념입니다.

구성 옵션

두 가지 메커니즘을 구성하여 자동 인스턴스 복구를 지원할 수 있습니다.

- 간소화된 자동 복구: 기본적으로 지원되는 인스턴스에서 활성화됩니다.
- CloudWatch 작업 기반 복구: 지원되는 인스턴스에 대한 수동 구성이 필요합니다.

시스템 상태 확인

시스템 상태 확인은 EC2 인스턴스가 실행되는 AWS 인프라를 자동으로 모니터링합니다.

- 시스템 상태 확인에 실패하면 AWS는 영향을 받는 인스턴스를 다른 하드웨어로 마이그레이션하려고 시도하는 자동 인스턴스 복구를 시작합니다.
- 실패한 시스템 상태 확인은 인스턴스 자체의 문제가 아니라 호스트의 하드웨어 또는 소프트웨어에 문제가 있음을 나타냅니다. 자동 인스턴스 복구는 시스템 상태 확인에 실패한 인스턴스를 복구할 수 있습니다. 그러나 인스턴스 상태 확인만 실패하면 자동 인스턴스 복구가 작동하지 않습니다.

- 인스턴스 상태 확인과 시스템 상태 확인의 차이점은 상태 확인 유형을 참조하세요.

기본 하드웨어 또는 소프트웨어 문제의 예

시스템 상태 확인이 실패할 수 있는 하드웨어 또는 소프트웨어 문제에는 네트워크 연결 손실, 시스템 전원 손실, 물리적 호스트의 소프트웨어 문제, 네트워크 연결성에 영향을 미치는 물리적 호스트의 하드웨어 문제 등이 있습니다.

복구된 인스턴스의 특성

복구된 인스턴스는 손실된 요소를 제외하고 원래 인스턴스와 동일합니다.

보존되는 요소:

- 인스턴스 ID
- 퍼블릭, 프라이빗 및 탄력적 IP 주소
- 인스턴스 메타데이터
- 배치 그룹
- 연결된 EBS 볼륨
- 가용 영역

손실되는 요소:

- 휘발성 메모리(RAM)에 저장된 데이터
- 인스턴스 저장소 볼륨에 저장된 데이터(CloudWatch 작업 기반 복구에만 해당)
- 운영 체제 가동 시간이 0으로 재설정됨

CloudWatch를 사용하여 시스템 상태 확인 모니터링

CloudWatch의 StatusCheckFailed_System 지표는 시스템 상태 확인의 통과 또는 실패 여부를 나타냅니다.

지표 값:

- 0 - 시스템 상태 확인을 통과했습니다.
- 1 - 시스템 상태 확인에 실패했습니다.

AWS Health Dashboard의 이벤트

자동 인스턴스 복구 시도 중에는 AWS가 구성된 복구 메커니즘과 그 결과에 따라 AWS Health Dashboard에 이벤트를 보냅니다.

- 간소화된 자동 복구
  - 성공 이벤트: `AWS_EC2_SIMPLIFIED_AUTO_RECOVERY_SUCCESS`
  - 실패 이벤트: `AWS_EC2_SIMPLIFIED_AUTO_RECOVERY_FAILURE`
- CloudWatch 작업 기반 복구
  - 성공 이벤트: `AWS_EC2_INSTANCE_AUTO_RECOVERY_SUCCESS`
  - 실패 이벤트: `AWS_EC2_INSTANCE_AUTO_RECOVERY_FAILURE`

간소화된 자동 복구와 CloudWatch 작업 기반 복구의 차이점

다음 표에서는 간소화된 자동 복구와 CloudWatch 작업 기반 복구의 주요 차이점을 비교합니다.

비교 요점	간소화된 자동 복구	CloudWatch 작업 기반 복구
구성	지원되는 인스턴스에서 기본적으로 활성화됨	CloudWatch 경보 및 작업의 수동 구성 필요
유연성	AWS에서 관리하는 복구 동작 수정	사용자 지정 가능한 작업 및 조건
Notification	AWS Health Dashboard를 통한 기본 알림	SNS를 통한 사용자 지정 가능 알림
메탈 인스턴스 크기	제외됨	포함
인스턴스 스토어 볼륨이 시작 시에 연결됨	시작 시 인스턴스 저장소 볼륨을 연결하는 인스턴스에는 지원되지 않음	일부 인스턴스 유형에서 지원됨. 인스턴스 복구 중 인스턴스 볼륨 볼륨의 데이터가 손실됨.
복구 시간	표준 복구 시도	간소화된 자동 복구보다 빠른 복구 시도
마이그레이션 중 호스트 문제 해결	마이그레이션이 취소될 수 있으며 인스턴스는 원래 호스트에 남아 있음	계속 새 호스트로 마이그레이션이 진행됨
비용	추가 요금 없음	CloudWatch 요금이 발생할 수 있음

복원력 있는 시스템 구축

간소화된 자동 복구 및 CloudWatch 작업 기반 복구는 개별 인스턴스 가용성을 유지하는 데 효과적이지만 AWS에서는 정상 인스턴스로 유입되는 트래픽을 장애 조치할 수 있는 고가용성 아키텍처를 구현할 것을 권장합니다.

이를 위해서는 Elastic Load Balancing(수신 트래픽을 여러 EC2 인스턴스에 분산) 및 Amazon EC2 Auto Scaling(수요 및 상태에 따라 인스턴스 수를 자동으로 조정)과 같은 AWS 서비스를 사용하는 것이 좋습니다.

EC2 인스턴스를 사용하여 복원력이 뛰어나고 내결함성을 갖춘 시스템을 구축하는 방법에 대한 자세한 내용은 다음 리소스를 참조하세요.

- AWS YouTube 채널의 Back to Basics: Designing for Failure with EC2
- AWS 아키텍처 블로그 사이트의 Disaster Recovery (DR) Architecture on AWS, Part I: Strategies for Recovery in the Cloud
- Application Load Balancer 사용 설명서
- Amazon EC2 Auto Scaling 사용 설명서
- Reliability Pillar AWS Well-Architected Framework의 REL11-BP02 정상 리소스로 장애 조치

자동 인스턴스 복구가 발생했는지 확인

인스턴스가 오프라인으로 표시되고 예기치 않게 재부팅된 경우 기본 하드웨어 또는 소프트웨어 문제에 대응하여 자동 인스턴스 복구가 수행되었을 수 있습니다. AWS Health Dashboard에서 자동 인스턴스 복구 이벤트를 확인하여 발생 여부를 확인할 수 있습니다. StatusCheckFailed_System Amazon CloudWatch 지표를 확인하여 인스턴스에 대한 기본 하드웨어 또는 소프트웨어 문제가 감지되었는지 확인할 수도 있습니다.

AWS Health Dashboard에서 이벤트 확인

자동 인스턴스 복구 시도가 발생하면 AWS는 이벤트를 AWS Health Dashboard에 전송합니다. 이러한 이벤트는 구성된 복구 메커니즘과 시도 성공 또는 실패 여부에 따라 달라집니다.

AWS Health Dashboard에서 자동 인스턴스 복구 이벤트를 확인하려면

1. https://phd.aws.amazon.com/phd/home#/에서 AWS Health Dashboard를 엽니다.
2. 자동 인스턴스 복구와 관련된 이벤트를 찾습니다. 이러한 이벤트가 있으면 자동 인스턴스 복구 시도가 발생했는지 여부와 그 결과를 확인할 수 있습니다.

- 간소화된 자동 복구
  - 성공 이벤트: `AWS_EC2_SIMPLIFIED_AUTO_RECOVERY_SUCCESS`
  - 실패 이벤트: `AWS_EC2_SIMPLIFIED_AUTO_RECOVERY_FAILURE`
- CloudWatch 작업 기반 복구
  - 성공 이벤트: `AWS_EC2_INSTANCE_AUTO_RECOVERY_SUCCESS`
  - 실패 이벤트: `AWS_EC2_INSTANCE_AUTO_RECOVERY_FAILURE`

CloudWatch를 사용하여 시스템 상태 확인 모니터링

CloudWatch에서 StatusCheckFailed_System 지표를 확인하여 인스턴스에 대한 기본 하드웨어 또는 소프트웨어 문제가 감지되었는지 확인할 수 있습니다. 지표 값은 시스템 상태 확인을 통과했는지(하드웨어 또는 소프트웨어 문제가 없음) 또는 실패했는지(하드웨어 또는 소프트웨어 문제가 있음)를 나타냅니다.

기본 하드웨어 또는 소프트웨어 문제가 감지되었는지 확인하려면

1. https://console.aws.amazon.com/cloudwatch/home?#metricsV2에서 CloudWatch 콘솔 지표 페이지를 엽니다.
2. EC2 인스턴스와 동일한 리전에 있는지 확인합니다.
3. 지표 검색 필드에 다음 지표를 붙여 넣고 Enter 키를 누릅니다.

   ```
   StatusCheckFailed_System
   ```

4. EC2 > 인스턴스별 지표를 선택합니다.
5. 테이블에서 확인하려는 인스턴스 옆의 확인란을 선택합니다.
6. 쿼리 기간을 복구 이벤트가 발생한 것으로 의심되는 시간으로 변경합니다.
7. 그래프 지표 탭을 선택하고 StatusCheckFailed_System에서 다음을 수행합니다.
   a. 통계에서 평균, 최대 또는 최소를 선택합니다.
   b. 기간에서 1분을 선택합니다.
8. StatusCheckFailed_System 값을 확인합니다.
   - 값 0: 시스템 상태 확인을 통과했으며 기본 하드웨어 또는 소프트웨어 문제가 없음을 나타냅니다.

- 값 1: 시스템 상태 확인에 실패했으며 기본 하드웨어 또는 소프트웨어 문제가 있음을 나타냅니다.

자세한 내용은 자동 인스턴스 복구 섹션을 참조하세요.

Amazon EC2 인스턴스에서 간소화된 자동 복구 구성

> **Important**
>
> 이 섹션에서는 EC2 인스턴스에서 복구 메커니즘을 사전에 구성하는 방법을 설명합니다. 이러한 복구 메커니즘은 AWS가 시스템 상태 확인 실패를 유발하는 기본 하드웨어 또는 소프트웨어 결함을 감지할 때 인스턴스 가용성을 복원하도록 설계되었습니다. 현재 인스턴스에 액세스하는 데 문제가 있는 경우 EC2 인스턴스 문제 해결을 참조하세요.

AWS가 기본 하드웨어 또는 소프트웨어 문제로 인해 인스턴스가 사용할 수 없는 상태임을 감지하면 간소화된 자동 복구는 기본 문제가 있는 호스트에서 다른 호스트로 인스턴스를 이동하여 인스턴스 가용성을 자동으로 복원할 수 있습니다.

간소화된 자동 복구가 발생하는 경우 AWS는 결과에 따라 다음 이벤트 중 하나를 AWS Health Dashboard에 전송합니다.

- 성공 이벤트: `AWS_EC2_SIMPLIFIED_AUTO_RECOVERY_SUCCESS`
- 실패 이벤트: `AWS_EC2_SIMPLIFIED_AUTO_RECOVERY_FAILURE`

이러한 이벤트에 대한 알림을 받으려면 알림을 구성할 수 있습니다. 자세한 내용은 AWS 사용자 알림 사용 설명서의 AWS 사용자 알림에서 첫 번째 알림 구성 만들기를 참조하세요. 또한 Amazon EventBridge 규칙을 사용하여 간소화된 자동 복구 이벤트를 모니터링할 수 있습니다.

간소화된 자동 복구는 기본적으로 인스턴스 시작 중 지원되는 인스턴스에서 활성화됩니다. 그러나 인스턴스가 `running` 상태이고 AWS Health Dashboard에 나열된 서비스 이벤트가 없으며 인스턴스 유형에 사용 가능한 용량이 있는 경우에만 작동합니다. 심각한 중단과 같은 일부 상황에서는 용량 제약으로 인해 복구 시도가 실패할 수 있습니다. 자세한 내용은 the section called “간소화된 자동 복구 실패 문제 해결” 섹션을 참조하세요.

시작 중 또는 시작 후 간소화된 자동 복구를 비활성화하고 필요한 경우 나중에 다시 활성화할 수 있습니다.

> **Warning**
>
> AWS가 기본 하드웨어 또는 소프트웨어 문제로 인해 인스턴스를 복구하는 경우 휘발성 메모리(RAM)에 저장된 데이터가 손실되고 운영 체제의 가동 시간이 0에서 다시 시작됩니다. 데이터 손실을 방지하려면 중요한 데이터의 백업을 정기적으로 생성하는 것이 좋습니다. EC2 인스턴스의 백업 및 복구 모범 사례에 대한 자세한 내용은 Amazon EC2 모범 사례를 참조하세요. 자동 인스턴스 복구 메커니즘은 개별 인스턴스에 맞게 설계되었습니다. 복원력이 높은 시스템 구축에 대한 지침은 복원력 있는 시스템 구축 섹션을 참조하세요.

내용

- 간소화된 자동 복구 활성화 요구 사항
- 간소화된 자동 복구 구성
- 간소화된 자동 복구 실패 문제 해결

간소화된 자동 복구 활성화 요구 사항

다음 기준을 충족하는 인스턴스에서 간소화된 자동 복구를 활성화할 수 있습니다.

인스턴스 타입

- 범용: A1, M3, M4, M5, M5a, M5n, M5zn, M6a, M6g, M6i, M6in, M7a, M7g, M7i, M7i-flex, M8g, T1, T2, T3, T3a, T4g
- 컴퓨팅 최적화: C3, C4, C5, C5a, C5n, C6a, C6g, C6gn, C6i, C6in, C7a, C7g, C7gn, C7i, C7i-flex, C8g
- 메모리 최적화: R3, R4, R5, R5a, R5b, R5n, R6a, R6g, R6i, R6in, R7a, R7g, R7i, R7iz, R8g, U-3tb1, U-6tb1, U-9tb1, U-12tb1, U-18tb1, U-24tb1, U7i-6tb, U7i-8tb, U7i-12tb, U7in-16tb, U7in-24tb, U7in-32tb, U7inh-32tb, X1, X1e, X2iezn, X8g
- 가속 컴퓨팅: G3, G5g, Inf1, P3, VT1
- 고성능 컴퓨팅: Hpc6a, Hpc7a, Hpc7g

Tenancy

- 공유
- Dedicated Instance

  자세한 내용은 Amazon EC2 전용 인스턴스 섹션을 참조하세요.

제한 사항

간소화된 자동 복구는 다음과 같은 특성이 있는 인스턴스에서는 지원되지 않습니다.

- 인스턴스 크기: `metal` 인스턴스
- 테넌시: 전용 호스트. 전용 호스트의 경우 전용 호스트 자동 복구를 대신 사용합니다.
- 스토리지: 인스턴스 저장소 볼륨이 있는 인스턴스
- 네트워킹: Elastic Fabric Adapter를 사용하는 인스턴스
- Auto Scaling: Auto Scaling 그룹에 포함된 인스턴스
- 유지 관리: 현재 예약된 유지 관리 이벤트가 진행 중인 인스턴스

간소화된 자동 복구 구성

기본적으로 지원되는 인스턴스가 시작될 때 해당 인스턴스에 대해 간소화된 자동 복구가 활성화됩니다. 인스턴스 시작 중 또는 이후 자동 복구 동작을 `disabled`로 설정할 수 있습니다.

`default` 구성에서는 지원되지 않는 인스턴스에 대해 간소화된 자동 복구가 활성화되지 않습니다.

Console

시작 시 간소화된 자동 복구 사용 중지

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스(Instances)를 선택한 후 인스턴스 시작(Launch instance)을 선택합니다.
3. 고급 세부 정보 섹션에서 인스턴스 자동 복구에 대해 비활성화됨을 선택합니다.
4. 필요에 따라 나머지 인스턴스 시작 설정을 구성한 다음 인스턴스를 시작합니다.

시작 후 간소화된 자동 복구를 비활성화하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택한 다음에 작업(Actions), 인스턴스 설정(Instance settings), 자동 복구 동작 변경(Change auto-recovery behavior)을 선택합니다.
4. 해제(Off)를 선택한 다음 저장(Save)을 선택합니다.

시작 후 간소화된 자동 복구를 활성화하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택한 다음에 작업(Actions), 인스턴스 설정(Instance settings), 자동 복구 동작 변경(Change auto-recovery behavior)을 선택합니다.
4. 기본값을 선택한 다음 저장을 선택합니다.

AWS CLI

시작 시 간소화된 자동 복구 사용 중지

run-instances 명령을 `--maintenance-options` 옵션과 함께 사용합니다.

```
--maintenance-options AutoRecovery=Disabled
```

시작 후 간소화된 자동 복구를 비활성화하는 방법

modify-instance-maintenance-options 명령을 사용합니다.

```
aws ec2 modify-instance-maintenance-options \
    --instance-id i-1234567890abcdef0 \
    --auto-recovery disabled
```

시작 후 간소화된 자동 복구를 활성화하는 방법

modify-instance-maintenance-options 명령을 사용합니다.

```
aws ec2 modify-instance-maintenance-options \
    --instance-id i-1234567890abcdef0 \
    --auto-recovery default
```

PowerShell

시작 시 간소화된 자동 복구 사용 중지

New-EC2Instance cmdlet을 사용합니다.

```
-MaintenanceOptions_AutoRecovery Disabled
```

시작 후 간소화된 자동 복구를 비활성화하는 방법

Edit-EC2InstanceMaintenanceOption cmdlet을 사용합니다.

```
Edit-EC2InstanceMaintenanceOption `
    -InstanceId i-1234567890abcdef0 `
    -AutoRecovery Disabled
```

시작 후 간소화된 자동 복구를 활성화하는 방법

Edit-EC2InstanceMaintenanceOption cmdlet을 사용합니다.

```
Edit-EC2InstanceMaintenanceOption `
    -InstanceId i-1234567890abcdef0 `
    -AutoRecovery Enabled
```

간소화된 자동 복구 실패 문제 해결

간소화된 자동 복구에서 인스턴스 복구에 실패할 경우 다음 사항을 고려하세요.

- AWS 서비스 이벤트 실행 중

  간소화된 자동 복구는 AWS Health Dashboard에서 서비스 이벤트가 발생하는 동안에는 작동하지 않습니다. 이 이벤트에 대한 복구 실패 알림을 받지 못할 수도 있습니다. 최신 서비스 가용성 정보에 대해서는 서비스 상태 상태 페이지를 참조하세요.

- 용량 부족

  인스턴스를 마이그레이션할 대체 하드웨어가 일시적으로 부족합니다.

- 최대 일일 복구 시도 횟수 도달

  인스턴스 복구 시도가 하루 최대 허용 횟수에 도달했습니다. 자동 복구가 실패하고 원래 시스템 상태 확인 실패의 근본 원인이 하드웨어 성능 저하로 확인되는 경우 이후에 인스턴스가 종료될 수 있습니다.

여러 번의 복구 시도에도 불구하고 인스턴스의 시스템 상태 검사 실패가 계속되는 경우 상태 검사에 실패한 인스턴스 문제 해결에서 추가 지침을 참조하세요.

EC2 인스턴스에서 CloudWatch 작업 기반 복구 구성

> **Important**
>
> 이 섹션에서는 EC2 인스턴스에서 복구 메커니즘을 사전에 구성하는 방법을 설명합니다. 이러한 복구 메커니즘은 AWS가 시스템 상태 확인 실패를 유발하는 기본 하드웨어 또는 소프트웨어 결함을 감지할 때 인스턴스 가용성을 복원하도록 설계되었습니다. 현재 인스턴스에 액세스하는 데 문제가 있는 경우 EC2 인스턴스 문제 해결을 참조하세요.

AWS가 기본 하드웨어 또는 소프트웨어 문제로 인해 인스턴스가 사용할 수 없는 상태임을 감지하면 CloudWatch 작업 기반 복구는 기본 문제가 있는 호스트에서 다른 호스트로 인스턴스를 이동하여 인스턴스 가용성을 자동으로 복원할 수 있습니다.

CloudWatch 작업 기반 복구가 발생하는 경우 AWS는 결과에 따라 다음 이벤트 중 하나를 AWS Health Dashboard에 전송합니다.

- 성공 이벤트: `AWS_EC2_INSTANCE_AUTO_RECOVERY_SUCCESS`
- 실패 이벤트: `AWS_EC2_INSTANCE_AUTO_RECOVERY_FAILURE`

CloudWatch 작업 기반 복구를 구성하면 Amazon CloudWatch 경보에 복구 작업을 추가할 수 있습니다. CloudWatch 작업 기반 복구는 `StatusCheckFailed_System` 지표와 함께 작동합니다. CloudWatch 작업 기반 복구에서는 최첨단 복구 응답 시간 세분성과 복구 작업 및 결과에 대한 Amazon Simple Notification Service(Amazon SNS) 알림이 제공됩니다. 이러한 구성 옵션을 사용하면 간소화된 자동 복구에 비해 시스템 상태 검사 실패 이벤트 응답을 더 세밀하게 제어하여 복구 시도를 더 빠르게 수행할 수 있습니다. 사용 가능한 CloudWatch 옵션에 대한 자세한 내용은 인스턴스 상태 검사를 참조하세요.

그러나 CloudWatch 작업 기반 복구는 인스턴스가 `running` 상태이고 AWS Health Dashboard에 나열된 서비스 이벤트가 없으며 인스턴스 유형에 사용 가능한 용량이 있는 경우에만 작동합니다. 심각한 중단과 같은 일부 상황에서는 용량 제약으로 인해 복구 시도가 실패할 수 있습니다. 자세한 내용은 the section called “문제 해결” 섹션을 참조하세요.

> **Warning**
>
> AWS가 기본 하드웨어 또는 소프트웨어 문제로 인해 인스턴스를 복구하는 경우 휘발성 메모리(RAM)와 인스턴스 저장소 볼륨에 저장된 데이터가 손실되고 운영 체제의 가동 시간이 0에서 다시 시작됩니다. 데이터 손실을 방지하려면 중요한 데이터의 백업을 정기적으로 생성하는 것

이 좋습니다. EC2 인스턴스의 백업 및 복구 모범 사례에 대한 자세한 내용은 Amazon EC2 모범 사례를 참조하세요.
자동 인스턴스 복구 메커니즘은 개별 인스턴스에 맞게 설계되었습니다. 복원력이 높은 시스템 구축에 대한 지침은 복원력 있는 시스템 구축 섹션을 참조하세요.

내용

- CloudWatch 작업 기반 복구 활성화에 대한 요구 사항
- CloudWatch 작업 기반 복구 구성
- CloudWatch 작업 기반 복구 실패 문제 해결

CloudWatch 작업 기반 복구 활성화에 대한 요구 사항

CloudWatch 작업 기반 복구는 다음 기준을 충족하는 인스턴스에서 활성화할 수 있습니다.

인스턴스 타입

- 범용: A1, M3, M4, M5, M5a, M5n, M5zn, M6a, M6g, M6i, M6in, M7a, M7g, M7i, M7i-flex, M8g, T1, T2, T3, T3a, T4g
- 컴퓨팅 최적화: C3, C4, C5, C5a, C5n, C6a, C6g, C6gn, C6i, C6in, C7a, C7g, C7gn, C7i, C7i-flex, C8g
- 메모리 최적화: R3, R4, R5, R5a, R5b, R5n, R6a, R6g, R6i, R6in, R7a, R7g, R7i, R7iz, R8g, U-3tb1, U-6tb1, U-9tb1, U-12tb1, U-18tb1, U-24tb1, U7i-6tb, U7i-8tb, U7i-12tb, U7in-16tb, U7in-24tb, U7in-32tb, U7inh-32tb, X1, X1e, X2idn, X2iedn, X2iezn, X8g
- 가속 컴퓨팅: G3, G5g, Inf1, P3, VT1
- 고성능 컴퓨팅: Hpc6a, Hpc7a, Hpc7g
- 베탈 인스턴스: 베탈 인스턴스 크기를 가진 위의 모든 인스턴스 유형
- 시작 시 인스턴스 저장소 볼륨이 추가되는 경우: M3, C3, R3, X1, X1e, X2idn, X2iedn 인스턴스 유형만 지원

Tenancy

- 공유
- Dedicated Instance

자세한 내용은 Amazon EC2 전용 인스턴스 섹션을 참조하세요.

제한 사항

CloudWatch 작업 기반 복구는 다음과 같은 특성이 있는 인스턴스에서는 지원되지 않습니다.

- 테넌시: 전용 호스트. 전용 호스트의 경우 전용 호스트 자동 복구를 대신 사용합니다.
- 네트워킹: Elastic Fabric Adapter를 사용하는 인스턴스
- Auto Scaling: Auto Scaling 그룹에 포함된 인스턴스
- 유지 관리: 현재 예약된 유지 관리 이벤트가 진행 중인 인스턴스

지원되는 인스턴스 유형 찾기

CloudWatch 작업 기반 복구를 지원하는 인스턴스 유형을 볼 수 있습니다.

Console

CloudWatch 작업 기반 복구를 지원하는 인스턴스 유형을 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 인스턴스 유형(Instance Types)을 선택합니다.
3. 필터 막대에 Auto Recovery support = true 필터를 추가합니다. 인스턴스 유형 테이블에는 CloudWatch 작업 기반 복구를 지원하는 모든 인스턴스 유형이 표시됩니다.
4. (선택 사항) 필터를 추가하여 관심 있는 특정 인스턴스 유형에 범위를 추가로 지정합니다.

AWS CLI

CloudWatch 작업 기반 복구를 지원하는 인스턴스 유형을 보려면

describe-instance-types 명령을 `auto-recovery-supported` 필터와 함께 사용합니다.

```
aws ec2 describe-instance-types \
    --filters Name=auto-recovery-supported,Values=true \
    --query "InstanceTypes[*].[InstanceType]" \
    --output text | sort
```

PowerShell

CloudWatch 작업 기반 복구를 지원하는 인스턴스 유형을 보려면

Get-EC2InstanceType cmdlet을 `auto-recovery-supported` 필터와 함께 사용합니다.

```
Get-EC2InstanceType `
    -Filter @{Name="auto-recovery-supported";Values="true"} | `
    Select InstanceType | Sort-Object InstanceType
```

CloudWatch 작업 기반 복구 구성

EC2 인스턴스에 대한 CloudWatch 작업 기반 복구를 구성하려면 지정된 인스턴스의 `StatusCheckFailed_System` 지표를 모니터링하는 CloudWatch 경보를 생성합니다. 지표 값이 1(시스템 상태 확인 실패)일 때 경보가 트리거되도록 설정합니다. 트리거될 때 인스턴스를 자동으로 복구하도록 경보 작업을 구성합니다.

Amazon EC2 콘솔 또는 CloudWatch 콘솔을 사용하여 경보를 구성할 수 있습니다. 지침은이 본 사용 설명서 Amazon CloudWatch 경보에 복구 작업 추가 섹션 또는 Amazon CloudWatch 사용 설명서의 Amazon CloudWatch 경보에 복구 작업 추가하기를 참조하세요.

CloudWatch 작업 기반 복구 실패 문제 해결

CloudWatch 작업 기반 복구가 인스턴스 복구에 실패할 경우 다음 사항을 고려하세요.

- AWS 서비스 이벤트 실행 중

  CloudWatch 작업 기반 복구는 AWS Health Dashboard에서 서비스 이벤트가 발생하는 동안에는 작동하지 않습니다. 이 이벤트에 대한 복구 실패 알림을 받지 못할 수도 있습니다. 최신 서비스 가용성 정보에 대해서는 서비스 상태 상태 페이지를 참조하세요.

- 용량 부족

  인스턴스를 마이그레이션할 대체 하드웨어가 일시적으로 부족합니다.

- 최대 일일 복구 시도 횟수 도달

  인스턴스 복구 시도가 하루 최대 허용 횟수에 도달했습니다. 자동 복구가 실패하고 원래 시스템 상태 확인 실패의 근본 원인이 하드웨어 성능 저하로 확인되는 경우 이후에 인스턴스가 종료될 수 있습니다.

여러 번의 복구 시도에도 불구하고 인스턴스의 시스템 상태 검사 실패가 계속되는 경우 상태 검사에 실패한 인스턴스 문제 해결에서 추가 지침을 참조하세요.

인스턴스 메타데이터를 사용하여 EC2 인스턴스를 관리합니다.

인스턴스 메타데이터는 실행 중인 인스턴스를 구성 또는 관리하는 데 사용될 수 있는 인스턴스 관련 데이터입니다. 인스턴스 메타데이터에는 다음이 포함됩니다.

인스턴스 메타데이터 속성

인스턴스 메타데이터 속성은 호스트 이름, 이벤트, 보안 그룹 등의 범주로 분류됩니다.

동적 데이터

동적 데이터는 인스턴스가 시작될 때 생성되는 메타데이터(예: 인스턴스 ID 문서)입니다. 자세한 내용은 동적 데이터 카테고리 섹션을 참조하세요.

사용자 데이터

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 수도 있습니다. 예를 들어 다양한 소규모 비즈니스를 위해 웹 서버를 운영하는 경우 모두 동일한 일반 AMI를 사용하고 시작 시 사용자 데이터에 지정한 Amazon S3 버킷에서 콘텐츠를 검색할 수 있습니다. 언제든지 새 고객을 추가하려면 고객에 대한 버킷을 생성하고, 콘텐츠를 추가한 다음, 사용자 데이터에서 코드에 제공된 고유의 버킷 이름으로 AMI를 시작합니다. 동일한 `RunInstances` 호출을 사용하여 여러 인스턴스를 시작하면 해당 예약의 모든 인스턴스에서 사용자 데이터를 사용할 수 있습니다. 동일한 예약에 속하는 각 인스턴스에는 고유한 `ami-launch-index` 번호가 있으므로 인스턴스의 기능을 제어하는 코드를 작성할 수 있습니다. 예를 들어, 첫 번째 호스트는 클러스터의 원래 노드로 자체 선택될 수 있습니다. 자세한 AMI 시작 예는 단원을 참조하십시오.단일 요청으로 시작된 각 인스턴스를 식별합니다.

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.

내용

- 인스턴스 메타데이터 카테고리
- 동적 데이터 카테고리
- EC2 인스턴스에 대한 인스턴스 메타데이터에 액세스
- 인스턴스 메타데이터 서비스 옵션 구성
- 사용자 데이터 입력을 사용하여 EC2 인스턴스를 시작할 때 명령 실행
- 단일 요청으로 시작된 각 인스턴스를 식별합니다.

인스턴스 메타데이터 카테고리

인스턴스 메타데이터 속성은 몇 가지 범주로 분류됩니다. 인스턴스 메타데이터 속성을 검색하려면 요청에서 범주를 지정하여 응답에서 메타데이터를 반환합니다.

새 카테고리가 릴리스되면 새 버전 번호로 새 인스턴스 메타데이터 빌드가 생성됩니다. 다음 표에서 카테고리가 릴리스된 버전(Version when category was released) 열은 인스턴스 메타데이터 카테고리가 릴리스될 때 빌드 버전을 지정합니다. Amazon EC2가 새 인스턴스 메타데이터 빌드를 릴리스할 때마다 코드를 업데이트하지 않으려면 메타데이터 요청에서 버전 번호 대신 `latest`를 사용하세요. 자세한 내용은 인스턴스 메타데이터의 사용 가능한 버전 가져오기 섹션을 참조하세요.

Amazon EC2에서 새 인스턴스 메타데이터 범주를 릴리스하면 새 범주에 대한 인스턴스 메타데이터를 기존 인스턴스에서 사용하지 못할 수 있습니다. Nitro 기반 인스턴스를 사용하면 시작 시 사용할 수 있었던 범주에 대해서만 인스턴스 메타데이터를 검색할 수 있습니다. Xen 하이퍼바이저가 있는 인스턴스의 경우 인스턴스를 중지한 후 시작하여 인스턴스에 사용 가능한 범주를 업데이트할 수 있습니다.

다음 표는 인스턴스 메타데이터의 카테고리를 목록으로 표시합니다. 범주 이름 중 일부에는 해당 인스턴스에만 있는 데이터 자리 표시자가 포함되어 있습니다. 예를 들어, *mac*은 네트워크 인터페이스의 MAC 주소를 나타냅니다. 인스턴스 메타데이터를 가져올 때 이 자리 표시자를 실제 값으로 바꿔야 합니다.

범주	설명	카테고리 릴리스 버전
`ami-id`	인스턴스를 시작하기 위해 사용된 AMI ID.	1.0
`ami-launch-index`	동일한 `RunInstances` 호출을 사용하여 여러 인스턴스를 시작하는 경우 이 값은 각 인스턴스의 시작 순서를 나타냅니다. 첫 번째 인스턴	1.0

범주	설명	카테고리 릴리스 버전
	스의 값은 0입니다. Auto Scaling 또는 EC2 플릿을 사용하여 인스턴스를 시작하는 경우 이 값은 항상 0입니다.	
ami-manifest-path	Amazon S3에 위치한 AMI 매니페스트 파일 경로. Amazon EBS 지원 AMI를 사용하여 인스턴스를 시작한 경우 반환되는 결과는 unknown입니다.	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
autoscaling/target-lifecycle-state	Auto Scaling 인스턴스가 전환 중인 대상 Auto Scaling 수명 주기 상태를 보여주는 값입니다. 2022년 3월 10일 이후에 인스턴스가 대상 수명 주기 상태 중 하나로 전환될 때 표시됩니다. 가능한 값은 Detached | InService | Standby | Terminated | Warmed:Hibernated | Warmed:Running | Warmed:Stopped | Warmed:Terminated 입니다. Amazon EC2 Auto Scaling 사용 설명서의 인스턴스 메타데이터를 통해 대상 수명 주기 상태 검색을 참조하세요.	2021-07-15
block-device-mapping/ami	루트/부트 파일 시스템을 포함하는 가상 디바이스.	2007-12-15

범주	설명	카테고리 릴리스 버전
block-device-mapping/ebsN	Amazon EBS 볼륨과 연결된 가상 디바이스입니다. Amazon EBS 볼륨은 시작 시 존재하는 경우 또는 인스턴스를 마지막으로 시작한 시점에만 메타데이터에서 사용할 수 있습니다. N은 Amazon EBS 볼륨의 색인을 나타냅니다(ebs1 또는 ebs2 등).	2007-12-15
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	swap와 연결된 가상 디바이스. 항상 존재하는 것은 아님.	2007-12-15

범주	설명	카테고리 릴리스 버전
`events/maintenance/history`	인스턴스에 대해 완료되거나 취소한 유지 관리 이벤트가 있다면, 이벤트에 관한 정보가 있는 JSON 문자열이 포함됩니다.	2018-08-17
`events/maintenance/scheduled`	인스턴스에 대해 활성화된 유지 관리 이벤트가 있다면, 이벤트에 관한 정보가 있는 JSON 문자열이 포함됩니다. 자세한 내용은 Amazon EC2 인스턴스에 영향을 미치는 예약된 이벤트 보기 섹션을 참조하세요.	2018-08-17
`events/recommendations/rebalance`	인스턴스에 대해 EC2 인스턴스 리밸런싱 권고 알림이 생성되는 대략적인 시간(UTC)입니다. 다음은 이 범주에 대한 메타데이터의 예입니다 `{"noticeTime": "2020-11-05T08:22:00Z"}` . 이 범주는 알림이 생성된 후에만 사용할 수 있습니다. 자세한 내용은 EC2 인스턴스 리밸런싱 권고 섹션을 참조하세요.	2020-10-27

범주	설명	카테고리 릴리스 버전
`hostname`	EC2 인스턴스가 IP 기반 이름 지정(IPBN)을 사용하는 경우 인스턴스의 프라이빗 IPv4 DNS 호스트 이름입니다. EC2 인스턴스가 리소스 기반 이름 지정(RBN)을 사용하는 경우 RBN입니다. 다중 네트워크 인터페이스가 존재하는 경우 eth0 디바이스를 의미함(디바이스 번호가 0인 디바이스). IPBN 및 RBN에 대한 자세한 내용은 EC2 인스턴스 호스트 이름 및 도메인 섹션을 참조하세요.	1.0
`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 참조). 그렇지 않을 경우 제공되지 않습니다.	2012-01-12
`identity-credentials/ec2/info`	`identity-credentials/ec2/security-credentials/ec2-instance` 의 보안 인증에 대한 정보입니다.	2018-05-23

범주	설명	카테고리 릴리스 버전
identity-credentials/ec2/security-credentials/ec2-instance	인스턴스 내 소프트웨어가 EC2 Instance Connect 및 AWS Systems Manager 기본 호스트 관리 구성과 같은 기능을 지원하기 위해 AWS에 자신을 식별할 수 있도록 하는 인스턴스 ID 역할에 대한 보안 인증 정보입니다. 이러한 자격 증명에는 정책이 연결되어 있지 않으므로 AWS 기능에 대한 인스턴스를 식별하는 것 외에는 추가 AWS API 권한이 없습니다. 자세한 내용은 Amazon EC2 인스턴스의 인스턴스 자격 증명 역할 섹션을 참조하세요.	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-life-cycle	이 인스턴스의 구매 옵션입니다. 자세한 내용은 Amazon EC2 결제 및 구매 옵션 섹션을 참조하세요.	2019년 10월 1일
instance-type	인스턴스 유형. 자세한 내용은 Amazon EC2 인스턴스 유형 섹션을 참조하세요.	2007-08-29

범주	설명	카테고리 릴리스 버전
ipv6	인스턴스의 IPv6 주소. 여러 네트워크 인터페이스가 있는 경우 이는 eth0 디바이스(디바이스 번호가 0인 디바이스) 네트워크 인터페이스와 할당된 첫 번째 IPv6 주소를 나타냅니다. 네트워크 인터페이스[0]에 IPv6 주소가 없으면 이 항목이 설정되지 않고 HTTP 404 응답이 발생합니다.	2021-01-03
kernel-id	이 인스턴스와 함께 시작한 커널 ID(해당하는 경우).	2008-02-01
local-hostname	다중 네트워크 인터페이스가 존재하는 경우 eth0 디바이스를 의미함(디바이스 번호가 0인 디바이스). EC2 인스턴스가 IP 기반 이름 지정(IPBN)을 사용하는 경우 인스턴스의 프라이빗 IPv4 DNS 호스트 이름입니다. EC2 인스턴스가 리소스 기반 이름 지정(RBN)을 사용하는 경우 RBN입니다. IPBN, RBN 및 EC2 인스턴스 이름 지정에 대한 자세한 내용은 EC2 인스턴스 호스트 이름 및 도메인 섹션을 참조하세요.	2007-01-19
local-ipv4	인스턴스의 프라이빗 IPv4 주소. 다중 네트워크 인터페이스가 존재하는 경우 eth0 디바이스를 의미함(디바이스 번호가 0인 디바이스). IPv6 전용 인스턴스인 경우 이 항목이 설정되지 않고 HTTP 404 응답이 발생합니다.	1.0

범주	설명	카테고리 릴리스 버전
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
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 필드에 해당합니다.	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	인터페이스에 할당된 IPv6 주소	2016-06-30
network/interfaces/macs/mac/ipv6-prefix	네트워크 인터페이스에 할당된 IPv6 접두사	

범주	설명	카테고리 릴리스 버전
network/interfaces/macs/mac/local-hostname	인스턴스의 프라이빗 IPv4 DNS 호스트 이름. 다중 네트워크 인터페이스가 존재하는 경우 eth0 디바이스를 의미함(디바이스 번호가 0인 디바이스). IPv6 전용 인스턴스인 경우 리소스 기반 이름입니다. IPBN 및 RBN에 대한 자세한 내용은 EC2 인스턴스 호스트 이름 및 도메인 섹션을 참조하세요.	2007-01-19
network/interfaces/macs/mac/local-ipv4s	프라이빗 IPv4 주소는 인터페이스와 연결됩니다. IPv6 전용 네트워크 인터페이스인 경우 이 항목이 설정되지 않고 HTTP 404 응답이 발생합니다.	2011-01-01
network/interfaces/macs/mac/mac	인스턴스의 MAC 주소.	2011-01-01
network/interfaces/macs/*mac*/network-card	네트워크 카드의 인덱스입니다. 일부 인스턴스 유형은 여러 네트워크 카드를 지원합니다.	2020-11-01
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

범주	설명	카테고리 릴리스 버전
network/interfaces/macs/mac/public-hostname	인터페이스의 퍼블릭 DNS(IPv4). 이 범주는 enableDnsHostnames 속성이 true로 설정된 경우에만 반환됩니다. 자세한 내용을 알아보려면 Amazon VPC 사용 설명서의 VPC에 대한 DNS 속성을 참조하세요. 인스턴스에 퍼블릭 IPv6 주소만 있고 퍼블릭 IPv4 주소가 없는 경우 이 항목이 설정되지 않고 HTTP 404 응답이 발생합니다.	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
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

범주	설명	카테고리 릴리스 버전
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
placement/availability-zone-id	인스턴스가 시작된 정적 가용 영역 ID입니다. 가용 영역 ID는 계정 간에 일관성이 있습니다. 그러나 가용 영역과는 다를 수 있으며, 가용 영역은 계정에 따라 다를 수 있습니다.	2019년 10월 1일
placement/group-name	인스턴스가 시작된 배치 그룹의 이름입니다.	2020-08-24
placement/host-id	인스턴스가 시작된 호스트의 ID입니다. 전용 호스트에만 해당됩니다.	2020-08-24
placement/partition-number	인스턴스가 시작된 파티션의 번호입니다.	2020-08-24
placement/region	인스턴스가 시작된 AWS 리전입니다.	2020-08-24

범주	설명	카테고리 릴리스 버전
product-codes	AWS Marketplace인스턴스에 연결된 제품 코드(해당되는 경우).	2007-03-01
public-hostname	인스턴스의 퍼블릭 DNS(IPv4). 이 범주는 enableDnsHostnames 속성이 true로 설정된 경우에만 반환됩니다. 자세한 내용을 알아보려면 Amazon VPC 사용 설명서의 VPC에 대한 DNS 속성을 참조하세요. 인스턴스에 퍼블릭 IPv6 주소만 있고 퍼블릭 IPv4 주소가 없는 경우 이 항목이 설정되지 않고 HTTP 404 응답이 발생합니다.	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

범주	설명	카테고리 릴리스 버전
services/domain	리전의 AWS 리소스에 대한 도메인입니다.	2014-02-25
services/partition	리소스가 있는 파티션. 표준 AWS 리전에서 파티션은 aws입니다. 리소스가 다른 파티션에 있는 경우 파티션은 aws-*partitionname* 입니다. 예를 들어 중국(베이징) 리전에 있는 리소스의 파티션은 aws-cn입니다.	2015-10-20
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 섹션을 참조하세요.	2016-11-15
spot/termination-time	스팟 인스턴스의 운영 체제가 종료 신호를 수신하는 UTC 기준 예상 시간. 스팟 인스턴스가 Amazon EC2의 종료 대상으로 표시된 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 섹션을 참조하세요.	2014-11-05

범주	설명	카테고리 릴리스 버전
tags/instance	인스턴스와 연결된 인스턴스 태그입니다. 인스턴스 메타데이터의 태그에 대한 액세스를 명시적으로 허용하는 경우에만 사용할 수 있습니다. 자세한 내용은 인스턴스 메타데이터에서 태그에 대한 액세스 활성화 섹션을 참조하세요.	2021-03-23

동적 데이터 카테고리

다음 표는 동적 데이터의 카테고리를 목록으로 표시합니다.

범주	설명	카테고리 릴리스 버전
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. Amazon EC2 인스턴스의 인스턴스 자격 증명 문서을(를) 참조하세요.	2009-04-04
instance-identity/pkcs7	문서의 신뢰성 및 서명 내용을 검증하는 데 사용됩니다. Amazon EC2 인스턴스의 인스턴스 자격 증명 문서을(를) 참조하세요.	2009-04-04
instance-identity/signature	출처 및 신뢰성을 검증하기 위해 다른 사용자가 사용할 수 있는 데이터. Amazon EC2 인스턴스의 인스턴스 자격 증명 문서을(를) 참조하세요.	2009-04-04

EC2 인스턴스에 대한 인스턴스 메타데이터에 액세스

인스턴스 자체 내에서 또는 EC2 콘솔, API, SDK 또는 AWS CLI에서 EC2 인스턴스 메타데이터에 액세스할 수 있습니다. 콘솔 또는 명령줄에서 인스턴스의 현재 인스턴스 메타데이터 설정을 가져오려면 기존 인스턴스에 대한 인스턴스 메타데이터 옵션 쿼리 섹션을 참조하세요.

EBS 루트 볼륨이 있는 인스턴스의 사용자 데이터를 수정할 수도 있습니다. 인스턴스가 중지 상태여야 합니다. 콘솔 지침은 인스턴스 사용자 데이터 업데이트 섹션을 참조하세요. AWS CLI를 사용하는 Linux의 예는 modify-instance-attribute를 참조하세요. Windows PowerShell용 도구를 사용하는 Windows의 예는 the section called “사용자 데이터 및 Tools for Windows PowerShell” 섹션을 참조하세요.

> **Note**
>
> 인스턴스 메타데이터 및 사용자 데이터를 가져오기 위해 사용되는 HTTP 요청 비용은 청구되지 않습니다.

인스턴스 메타데이터 액세스 고려 사항

인스턴스 메타데이터 검색 문제를 방지하려면 다음을 고려하세요.

명령 형식

명령 형식은 인스턴스 메타데이터 서비스 버전 1(IMDSv1)을 사용하는지, 인스턴스 메타데이터 서비스 버전 2(IMDSv2)를 사용하는지에 따라 다릅니다. 기본적으로 인스턴스 메타데이터 서비스의 두 버전을 모두 사용할 수 있습니다. IMDSv2를 사용해야 하도록 설정하려면 인스턴스 메타데이터 서비스를 사용하여 인스턴스 메타데이터에 액세스 섹션을 참조하세요.

IMDSv2가 필요한 경우 IMDSv1이 작동하지 않습니다.

IMDSv1을 사용하고 응답이 없는 경우 IMDSv2가 필요할 수 있습니다. IMDSv2가 필수인지 확인하려면 세부 정보를 확인할 인스턴스를 선택합니다. IMDSv2 값은 필수(IMDSv2를 사용해야 함)이거나 선택 사항(IMDSv2 또는 IMDSv1을 사용할 수 있음)임을 나타냅니다.

(IMDSv2) /latest/api/token을 사용하여 토큰 검색

버전별 경로(예: `/2021-03-23/api/token`)에 대해 PUT 요청을 실행하면 메타데이터 서비스에서 403 Forbidden 오류가 반환됩니다. 이는 의도된 동작입니다.

메타데이터 버전

Amazon EC2가 새 인스턴스 메타데이터 빌드를 릴리스할 때마다 코드를 업데이트하지 않으려면 버전 번호가 아니라, 경로에서 `latest`를 사용하는 것이 좋습니다.

IPv6 지원

IPv6 주소를 사용하여 인스턴스 메타데이터를 검색하려면 IPv4 주소 `169.254.169.254` 대신 IMDS `[fd00:ec2::254]`의 IPv6 주소를 활성화하고 사용해야 합니다. 인스턴스는 IPv6을 지원하는 서브넷에서 시작된 Nitro 기반 인스턴스여야 합니다.

(Windows) Windows Sysprep을 사용하여 사용자 지정 AMI 생성

사용자 지정 Windows AMI에서 인스턴스를 시작할 때 IMDS가 작동하도록 하려면 AMI가 Windows Sysprep으로 만든 표준화된 이미지여야 합니다. 그렇지 않으면 IMDS가 작동하지 않습니다. 자세한 내용은 Windows Sysprep을 사용하여 Amazon EC2 AMI 생성 섹션을 참조하세요.

컨테이너 환경에서는 재구성을 고려하거나 홉 제한을 2로 늘리는 것이 좋습니다.

AWS SDK는 기본적으로 IMDSv2 호출을 사용합니다. IMDSv2 호출에 응답이 없으면 일부 AWS SDK는 호출을 다시 시도하고 여전히 실패하면 IMDSv1을 사용합니다. 이로 인해 특히 컨테이너 환경에서 지연이 발생할 수 있습니다. IMDSv2가 필요한 AWS SDK의 경우 컨테이너 환경에서 홉 제한이 1이면 호출에 응답이 전혀 없을 수 있습니다. 컨테이너로의 이동이 추가 네트워크 홉으로 간주되기 때문입니다.

컨테이너 환경에서 이러한 문제를 완화하려면 설정(예: AWS 리전)을 컨테이너에 직접 전달하도록 구성을 변경하거나 홉 제한을 2로 늘리는 것이 좋습니다. 홉 제한의 영향에 대한 자세한 내용은 EC2 인스턴스 메타데이터 서비스의 향상된 기능을 통해 개방형 방화벽, 역방향 프록시 및 SSRF 취약성에 대한 심층적인 방어 기능 추가를 참조하세요. 홉 제한 변경에 대한 자세한 내용은 PUT 응답 홉 제한 변경 섹션을 참조하세요.

초당 패킷 수(PPS) 제한

링크-로컬 주소를 사용하는 서비스에는 초당 1024 패킷(PPS) 제한이 있습니다. 이 제한에는 Route 53 Resolver DNS 쿼리, 인스턴스 메타데이터 서비스(IMDS) 요청, Amazon Time Service 네트워크 타임 프로토콜(NTP) 요청, Windows 라이선스 서비스(Microsoft Windows 기반 인스턴스용) 요청의 집계가 포함됩니다.

사용자 데이터 액세스에 대한추가 고려 사항

- 사용자 데이터는 불투명 데이터로 취급됩니다. 검색 시 지정한 항목만 가져옵니다. 사용자 데이터 해석과 그에 따른 조치 수행은 인스턴스에 따라 다릅니다.

- 사용자 데이터는 base64로 인코딩해야 합니다. 사용 중인 도구 또는 SDK에 따라 base64 인코딩이 자동으로 수행될 수 있습니다. 예시:
  - Amazon EC2 콘솔은 base64 인코딩을 수행하거나 base64로 인코딩된 입력을 수락할 수 있습니다.
  - AWS CLI 버전 2는 기본적으로 바이너리 파라미터의 base64 인코딩을 자동으로 수행합니다. AWS CLI 버전 1은 `--user-data` 파라미터의 base64 인코딩을 자동으로 수행합니다.
  - AWS SDK for Python (Boto3)는 `UserData` 파라미터의 base64 인코딩을 자동으로 수행합니다.
- 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.
- 인스턴스를 중지하고 사용자 데이터를 수정한 다음 인스턴스를 시작할 경우 인스턴스를 시작할 때 업데이트된 사용자 데이터가 자동으로 실행되지 않습니다. Windows 인스턴스에서는 인스턴스를 시작할 때 한 번만 또는 인스턴스를 재부팅하거나 시작할 때마다 업데이트된 사용자 데이터 스크립트가 실행되도록 설정을 구성할 수 있습니다.
- 사용자 데이터는 인스턴스 속성입니다. 인스턴스에서 AMI를 생성하는 경우 인스턴스 사용자 데이터는 AMI에 포함되지 않습니다.

EC2 인스턴스 내에서 인스턴스 메타데이터에 액세스

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.

다음은 모두 인스턴스 메타데이터로 간주되지만 액세스 방법은 다릅니다. 액세스하려는 인스턴스 메타데이터의 유형을 나타내는 탭을 선택하면 자세한 정보를 볼 수 있습니다.

Metadata

인스턴스 메타데이터 속성은 몇 가지 범주로 분류됩니다. 각 인스턴스 메타데이터 범주에 대한 설명은 인스턴스 메타데이터 카테고리 섹션을 참조하세요.

실행 중인 인스턴스 내에서 인스턴스 메타데이터 속성에 액세스하려면 다음 IPv4 또는 IPv6 URI에서 데이터를 가져옵니다. 이러한 IP 주소는 링크-로컬 주소이며 인스턴스에서만 유효합니다. 자세한 내용은 링크-로컬 주소 섹션을 참조하세요.

IPv4

```
http://169.254.169.254/latest/meta-data/
```

IPv6

```
http://[fd00:ec2::254]/latest/meta-data/
```

Dynamic data

실행 중인 인스턴스 내에서 동적 데이터를 검색하려면 다음 URI 중 하나를 사용합니다.

IPv4

```
http://169.254.169.254/latest/dynamic/
```

IPv6

```
http://[fd00:ec2::254]/latest/dynamic/
```

예: cURL로 액세스

다음 예제에서는 cURL을 사용하여 개요 수준의 인스턴스 ID 범주를 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" 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
```

예: PowerShell로 액세스

다음 예제는 PowerShell을 사용하여 개요 수준의 인스턴스 ID 범주를 검색합니다.

IMDSv2

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/dynamic/instance-identity/
document
rsa2048
pkcs7
signature
```

IMDSv1

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/dynamic/instance-identity/
document
rsa2048
pkcs7
signature
```

동적 데이터 및 가져오기 방법의 예제에 대한 자세한 내용은 Amazon EC2 인스턴스의 인스턴스 자격 증명 문서 섹션을 참조하세요.

User data

인스턴스에서 사용자 데이터를 검색하려면 다음 URI 중 하나를 사용합니다. IPv6 주소를 사용하여 사용자 데이터를 검색하려면 해당 주소를 사용하도록 설정해야 하며 인스턴스는 IPv6를 지원하는 서브넷에 있는 Nitro 기반 인스턴스여야 합니다.

IPv4

```
http://169.254.169.254/latest/user-data
```

IPv6

```
http://[fd00:ec2::254]/latest/user-data
```

사용자 데이터를 요청하면 데이터 자체(콘텐츠 유형 application/octet-stream)가 반환됩니다. 인스턴스에 사용자 데이터가 없는 경우 요청에서 404 - Not Found를 반환합니다.

예: cURL로 액세스하여 쉼표로 구분된 텍스트 검색

다음 예제는 cURL을 사용하여 쉼표로 구분된 텍스트로 지정된 사용자 데이터를 검색합니다.

IMDSv2

```
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" http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

IMDSv1

```
curl http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

예: PowerShell로 액세스하여 쉼표로 구분된 텍스트 검색

다음 예제는 PowerShell을 사용하여 쉼표로 구분된 텍스트로 지정된 사용자 데이터를 검색합니다.

IMDSv2

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method GET -Uri http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

IMDSv1

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} `
```

```
-Method PUT -Uri http://169.254.169.254/latest/api/token} -Method GET -uri
 http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

예: cURL로 액세스하여 스크립트 검색

다음 예제는 cURL을 사용하여 스크립트로 지정된 사용자 데이터를 검색합니다.

IMDSv2

```
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" http://169.254.169.254/latest/user-
data
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

IMDSv1

```
curl http://169.254.169.254/latest/user-data
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

예: PowerShell로 액세스하여 스크립트 검색

다음 예제는 PowerShell을 사용하여 스크립트로 지정된 사용자 데이터를 검색합니다.

IMDSv2

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds"
 = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method GET -Uri
 http://169.254.169.254/latest/user-data
<powershell>
$file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
New-Item $file -ItemType file
```

```
</powershell>
<persist>true</persist>
```

IMDSv1

```
Invoke-RestMethod -uri http://169.254.169.254/latest/user-data
<powershell>
$file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

기존 인스턴스에 대한 인스턴스 메타데이터 옵션 쿼리

기존 인스턴스에 대한 인스턴스 메타데이터 옵션을 쿼리할 수 있습니다.

Console

기존 인스턴스의 인스턴스 메타데이터 옵션을 쿼리하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 다음 필드를 확인합니다.
   - IMDSv2 – 값은 필수 또는 선택입니다.
   - 인스턴스 메타데이터의 태그 허용 - 값은 활성화됨 또는 비활성화됨입니다.
4. 인스턴스를 선택한 상태로 작업, 인스턴스 설정, 인스턴스 메타데이터 옵션 수정을 차례로 선택합니다.

   대화 상자에는 선택한 인스턴스에 대해 인스턴스 메타데이터 서비스가 활성화 또는 비활성화되었는지 여부가 표시됩니다.

AWS CLI

기존 인스턴스의 인스턴스 메타데이터 옵션을 쿼리하려면

아래와 같이 describe-instances 명령을 사용합니다.

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

```
    --instance-id i-1234567898abcdef0 \
    --query 'Reservations[].Instances[].MetadataOptions'
```

PowerShell

Tools for PowerShell를 사용하여 기존 인스턴스의 인스턴스 메타데이터 옵션을 쿼리하는 방법

Get-EC2Instance Cmdlet을 사용하세요.

```
(Get-EC2Instance `
    -InstanceId i-1234567898abcdef0).Instances.MetadataOptions
```

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.

IMDSv1 요청에 응답이 없는 경우 IMDSv2가 필요할 수 있습니다.

IMDSv2를 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 IMDS가 비활성화되어 있습니다.
- `404 - Not Found` - 리소스를 사용할 수 없거나 이러한 리소스가 없습니다.
- `503` - 요청을 완료할 수 없습니다. 요청을 다시 시도하세요.

IMDS가 오류를 반환하면 curl은 출력에 오류 메시지를 인쇄하고 성공 상태 코드를 반환합니다. 오류 메시지는 `TOKEN` 변수에 저장되며 이로 인해 토큰을 사용하는 curl 명령이 실패합니다. `-f` 옵션을 사용하여 curl을 호출하면 HTTP 서버 오류가 발생할 경우 오류 상태 코드가 반환됩니다. 오류 처리를 활성화하면 쉘이 오류를 포착하여 스크립트를 중지할 수 있습니다.

쿼리 조절

쿼리는 인스턴스당 IMDS로 제한되고, 한 인스턴스에서 IMDS로의 동시 연결 수에도 제한이 있습니다.

IMDS를 사용하여 AWS 보안 인증 정보를 가져올 경우 모든 트랜잭션 중에 또는 많은 스레드나 프로세스에서 동시에 자격 증명을 쿼리하지 마세요. 이렇게 하면 제한이 발생할 수 있습니다. 자격 증명 만료일이 다가오기 전까지는 자격 증명을 캐시에 저장하는 것이 좋습니다. IAM 역할 및 해당 역할과 연결된 보안 자격 증명에 대한 자세한 내용은 인스턴스 메타데이터에서 보안 자격 증명 검색 섹션을 참조하세요.

IMDS에 액세스하는 동안 제한이 발생하면 지수 백오프 전략으로 쿼리를 다시 시도하세요.

인스턴스 메타데이터 서비스를 사용하여 인스턴스 메타데이터에 액세스

다음 방법 중 하나를 사용하여 실행 중인 인스턴스에서 인스턴스 메타데이터에 액세스할 수 있습니다.

- 인스턴스 메타데이터 서비스 버전 2(IMDSv2) – 세션 지향 방법

  예시는 IMDSv2 예제 섹션을 참조하세요.
- 인스턴스 메타데이터 서비스 버전 1(IMDSv1) – 요청/응답 방법

  예시는 IMDSv1 예제 섹션을 참조하세요.

기본적으로 IMDSv1 또는 IMDSv2를 사용하거나 둘 다 사용할 수 있습니다.

로컬 코드 또는 사용자가 IMDSv2를 사용해야 하도록 각 인스턴스에서 인스턴스 메타데이터 서비스(IMDS)를 구성할 수 있습니다. IMDSv2를 사용해야 하도록 지정하면 IMDSv1는 더 이상 작동하지 않습니다. 인스턴스에서 IMDSv2를 사용하도록 구성하는 방법에 대한 자세한 내용은 인스턴스 메타데이터 서비스 옵션 구성 섹션을 참조하세요.

PUT 또는 GET 헤더는 IMDSv2에만 있습니다. 요청에 이러한 헤더가 있는 경우 요청은 IMDSv2를 위한 것입니다. 헤더가 없는 경우 요청은 IMDSv1을 위한 것으로 간주됩니다.

IMDSv2에 대한 자세한 내용은 EC2 인스턴스 메타데이터 서비스의 향상된 기능을 통해 개방형 방화벽, 역방향 프록시, SSRF 취약성에 대한 심층적인 방어 기능 추가를 참조하세요.

주제

- 인스턴스 메타데이터 서비스 버전 2 작동 방식
- 지원되는 AWS SDK 사용
- IMDSv2 예제

- IMDSv1 예제

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.

> **Note**
>
> 이 섹션의 예에서는 인스턴스 메타데이터 서비스(IMDS)의 IPv4 주소(`169.254.169.254`)를 사용합니다. IPv6 주소를 통해 EC2 인스턴스의 인스턴스 메타데이터를 검색하는 경우, 대신 IPv6 주소(`[fd00:ec2::254]`)를 활성화하고 사용해야 합니다. IMDS의 IPv6 주소는 IMDSv2 명령과 호환됩니다. IPv6 주소는 IPv6 지원 서브넷(듀얼 스택 또는 IPv6 전용)에 있는 Nitro 기반 인스턴스에서만 액세스할 수 있습니다.

다음 예에서는 쉘 스크립트와 IMDSv2를 사용하여 최상위 인스턴스 메타데이터 항목을 검색합니다. 각 예시:

- PUT 요청을 사용하여 6시간(21,600초) 동안 지속되는 세션 토큰을 생성합니다.
- 세션 토큰 헤더를 `TOKEN`(Linux 인스턴스) 또는 `token`(Windows 인스턴스)이라는 변수에 저장합니다.
- 토큰을 사용하여 최상위 메타데이터 항목을 요청합니다.

Linux 예

별도의 두 명령을 실행하거나 둘을 결합할 수 있습니다.

별도의 명령

먼저 다음 명령을 사용하여 토큰을 생성합니다.

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"`
```

그런 다음 해당 토큰을 사용하여 다음 명령으로 최상위 메타데이터 항목을 생성합니다.

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/
meta-data/
```

결합된 명령

토큰을 저장하고 명령을 결합할 수 있습니다. 다음 예는 위의 두 명령을 결합하고 TOKEN이라는 변수에 세션 토큰 헤더를 저장합니다.

> **Note**
>
> 토큰이 유효하지 않고, 토큰을 만드는 데 오류가 발생하면 오류 메시지가 변수에 저장되고 명령이 작동하지 않습니다.

```
[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" 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 $TOKEN에 저장한 토큰을 재사용합니다.

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/
meta-data/ami-id
```

Windows 예

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method GET -
Uri 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 $token에 저장한 토큰을 재사용합니다.

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} `
 -Method GET -uri 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. IMDS에 대한 모든 GET 요청에 토큰을 포함합니다. 토큰 사용이 `required`로 설정되면 유효한 토큰이 없거나 만료된 토큰이 있는 요청은 `401 - Unauthorized` HTTP 오류 코드를 수신합니다.
   - 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 PUT를 사용하여 새 세션을 생성해야 합니다.
   - 토큰을 재사용하거나 모든 요청에 새 토큰을 생성하도록 선택할 수 있습니다. 요청 수가 적은 경우 IMDS에 액세스해야 할 때마다 토큰을 생성하고 즉시 사용하는 것이 더 간편할 수 있습니다. 하지만 효율성을 향상하려면 인스턴스 메타데이터를 요청해야 할 때마다 PUT 요청을 작성하는 대신 토큰에 더 긴 기간을 지정하고 토큰을 재사용할 수 있습니다. 동시 토큰 수에는 실질적인 제한이 없으며 각각은 자체 세션을 나타냅니다. 그러나 IMDSv2에는 표준 IMDS 연결 및 조절 제한이 여전히 적용됩니다. 자세한 내용은 쿼리 조절 섹션을 참조하세요.

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 1의 응답 홉 제한(TTL(Time to Live))이 있습니다. 더 큰 홉 제한이 필요한 경우 modify-instance-metadata-options AWS CLI 명령을 사용하여 조정할 수 있습니다. 예를 들어 인스턴스에서 실행 중인 컨테이너 서비스가 있는 경우 이전 버전과의 호환성을 위해 더 큰 홉 제한이 필요할 수 있습니다. 자세한 내용은 기존 인스턴스에 대한 인스턴스 메타데이터 옵션 수정 섹션을 참조하세요.

지원되는 AWS SDK 사용

IMDSv2를 사용하려면 EC2 인스턴스에서 IMDSv2 사용을 지원하는 AWS SDK 버전을 사용해야 합니다. 모든 AWS SDK의 최신 버전에서는 IMDSv2 사용을 지원합니다.

> **Important**
>
> 최신 기능, 보안 업데이트 및 기본 종속성을 지원하려면 SDK 릴리스를 최신 상태로 유지하는 것이 좋습니다. 지원되지 않는 SDK 버전을 계속 사용하는 것은 권장되지 않으며 그에 따른 책

임은 사용자에게 있습니다. 자세한 내용은 AWS SDK 및 도구 참조 가이드에서 AWS SDK 및 도구 유지 관리 정책을 참조하세요.

다음은 IMDSv2 사용을 지원하는 최소 버전입니다.

- AWS CLI - 1.16.289
- AWS Tools for Windows PowerShell – 4.0.1.0
- AWS SDK for .NET – 3.3.634.1
- AWS SDK for C++ – 1.7.229
- AWS SDK for Go – 1.25.38
- AWS Go v2용 SDK – 0.19.0
- AWS SDK for Java – 1.11.678
- AWS SDK for Java 2.x – 2.10.21
- AWS Node.js의 JavaScript용 SDK – 2.722.0
- AWS SDK for Kotlin-1.1.4
- AWS SDK for PHP – 3.147.7
- AWS SDK for Python(Botocore) – 1.13.25
- AWS SDK for Python (Boto3) – 1.12.6
- AWS SDK for Ruby – 3.79.0

IMDSv2 예제

Amazon EC2 인스턴스에서 다음 예제를 실행하여 IMDSv2의 인스턴스 메타데이터를 검색합니다.

Windows 인스턴스에서 Windows PowerShell을 사용하거나 cURL 또는 wget을 설치할 수 있습니다. Windows 인스턴스에 서드 파티 도구를 설치하는 경우 직접 호출과 출력이 여기 설명된 것과 다를 수 있으므로 부속 문서를 주의 깊게 정독해야 합니다.

예시

- 인스턴스 메타데이터의 사용 가능한 버전 가져오기
- 최고 수준 메타데이터 항목 가져오기
- 메타데이터 항목의 값 가져오기
- 사용 가능한 퍼블릭 키 목록 가져오기

- 퍼블릭 키 0을 이용할 수 있는 형식 표시
- 퍼블릭 키 0(OpenSSH 키 형식) 가져오기
- 인스턴스에 대한 서브넷 ID 가져오기
- 인스턴스에 대한 인스턴스 태그 가져오기

인스턴스 메타데이터의 사용 가능한 버전 가져오기

이 예를 통해 이용 가능한 인스턴스 메타데이터 버전을 가져올 수 있습니다. 각 버전은 새 인스턴스 메타데이터 카테고리가 릴리스될 때 인스턴스 메타데이터 빌드를 참조합니다. 인스턴스 메타데이터 빌드 버전은 Amazon EC2 API 버전과 상관관계가 없습니다. 이전 버전의 구조 및 정보를 사용하는 스크립트인 경우 이전 버전을 사용할 수 있습니다.

cURL

```
[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" 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
...
latest
```

PowerShell

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri 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
...
latest
```

최고 수준 메타데이터 항목 가져오기

이 예제는 최고 수준 메타데이터 항목을 가져옵니다. 응답의 항목에 대한 자세한 내용은 인스턴스 메타데이터 카테고리 섹션을 참조하세요.

액세스를 허용한 경우에만 이 출력에 태그가 포함됩니다. 자세한 내용은 the section called “인스턴스 메타데이터에서 태그에 대한 액세스 활성화” 섹션을 참조하세요.

cURL

```
[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" 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-life-cycle
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/
tags/
```

PowerShell

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
hostname
iam/
instance-action
instance-id
instance-life-cycle
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
```

```
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/
tags/
```

메타데이터 항목의 값 가져오기

이 예에서는 앞의 예에서 얻은 최상위 메타데이터 항목 중 일부의 값을 가져옵니다. 이 요청은 이전 예제의 명령을 사용하여 생성한 저장된 토큰을 사용합니다. 토큰이 만료되지 않아야 합니다.

cURL

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/
latest/meta-data/ami-id
ami-0abcdef1234567890
```

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/
latest/meta-data/reservation-id
r-0efghijk987654321
```

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/
latest/meta-data/local-hostname
ip-10-251-50-12.ec2.internal
```

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/
latest/meta-data/public-hostname
ec2-203-0-113-25.compute-1.amazonaws.com
```

PowerShell

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/ami-id
ami-0abcdef1234567890
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/reservation-id
```

```
r-0efghijk987654321
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/local-hostname
ip-10-251-50-12.ec2.internal
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri 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.

cURL

```
[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" http://169.254.169.254/latest/meta-
data/public-keys/
0=my-public-key
```

PowerShell

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/public-keys/
0=my-public-key
```

퍼블릭 키 0을 이용할 수 있는 형식 표시

이 예제는 퍼블릭 키 0을 이용할 수 있는 형식을 보여줍니다.

cURL

```
[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" http://169.254.169.254/latest/meta-data/public-keys/0/
openssh-key
```

PowerShell

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
openssh-key
```

퍼블릭 키 0(OpenSSH 키 형식) 가져오기

이 예제에서는 퍼블릭 키 0(OpenSSH 키 형식)을 획득합니다.

cURL

```
[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" 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
```

PowerShell

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri 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.

cURL

```
[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" http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:29:96:8f:6a:2d/subnet-id
subnet-be9b61d7
```

PowerShell

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -
Method GET -Uri http://169.254.169.254/latest/meta-data/network/interfaces/
macs/02:29:96:8f:6a:2d/subnet-id
subnet-be9b61d7
```

인스턴스에 대한 인스턴스 태그 가져오기

인스턴스 메타데이터의 인스턴스 태그에 대한 액세스가 설정된 경우 인스턴스 메타데이터에서 인스턴스의 태그를 가져올 수 있습니다. 자세한 내용은 인스턴스 메타데이터에서 태그 검색 섹션을 참조하세요.

IMDSv1 예제

Amazon EC2 인스턴스에서 다음 예제를 실행하여 IMDSv1의 인스턴스 메타데이터를 검색합니다.

Windows 인스턴스에서 Windows PowerShell을 사용하거나 cURL 또는 wget을 설치할 수 있습니다. Windows 인스턴스에 서드 파티 도구를 설치하는 경우 직접 호출과 출력이 여기 설명된 것과 다를 수 있으므로 부속 문서를 주의 깊게 정독해야 합니다.

예시

- 인스턴스 메타데이터의 사용 가능한 버전 가져오기
- 최고 수준 메타데이터 항목 가져오기
- 메타데이터 항목의 값 가져오기
- 사용 가능한 퍼블릭 키 목록 가져오기
- 퍼블릭 키 0을 이용할 수 있는 형식 표시
- 퍼블릭 키 0(OpenSSH 키 형식) 가져오기
- 인스턴스에 대한 서브넷 ID 가져오기
- 인스턴스에 대한 인스턴스 태그 가져오기

인스턴스 메타데이터의 사용 가능한 버전 가져오기

이 예를 통해 이용 가능한 인스턴스 메타데이터 버전을 가져올 수 있습니다. 각 버전은 새 인스턴스 메타데이터 카테고리가 릴리스될 때 인스턴스 메타데이터 빌드를 참조합니다. 인스턴스 메타데이터 빌드 버전은 Amazon EC2 API 버전과 상관관계가 없습니다. 이전 버전의 구조 및 정보를 사용하는 스크립트인 경우 이전 버전을 사용할 수 있습니다.

cURL

```
[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
...
latest
```

PowerShell

```
PS C:\> Invoke-RestMethod -uri 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
...
latest
```

최고 수준 메타데이터 항목 가져오기

이 예제는 최고 수준 메타데이터 항목을 가져옵니다. 응답의 항목에 대한 자세한 내용은 인스턴스 메타데이터 카테고리 섹션을 참조하세요.

액세스를 허용한 경우에만 이 출력에 태그가 포함됩니다. 자세한 내용은 the section called “인스턴스 메타데이터에서 태그에 대한 액세스 활성화” 섹션을 참조하세요.

cURL

```
[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/
tags/
```

PowerShell

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
```

```
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/
tags/
```

메타데이터 항목의 값 가져오기

이 예제는 이전 예제에서 얻은 최상위 메타데이터 항목 중 일부의 값을 가져옵니다.

cURL

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/ami-id
ami-0abcdef1234567890
```

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/reservation-id
r-0efghijk987654321
```

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/local-hostname
ip-10-251-50-12.ec2.internal
```

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-hostname
ec2-203-0-113-25.compute-1.amazonaws.com
```

PowerShell

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/ami-id
ami-0abcdef1234567890
```

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/reservation-id
r-0efghijk987654321
```

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/local-hostname
ip-10-251-50-12.ec2.internal
```

```
PS C:\> Invoke-RestMethod -uri 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.

cURL

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/
0=my-public-key
```

PowerShell

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/public-keys/ 0=my-public-key
```

퍼블릭 키 0을 이용할 수 있는 형식 표시

이 예제는 퍼블릭 키 0을 이용할 수 있는 형식을 보여줍니다.

cURL

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/0/
```

```
openssh-key
```

PowerShell

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/public-
keys/0/openssh-key
openssh-key
```

퍼블릭 키 0(OpenSSH 키 형식) 가져오기

이 예제에서는 퍼블릭 키 0(OpenSSH 키 형식)을 획득합니다.

cURL

```
[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
```

PowerShell

```
PS C:\> Invoke-RestMethod -uri 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.

cURL

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/network/interfaces/
macs/02:29:96:8f:6a:2d/subnet-id
subnet-be9b61d7
```

PowerShell

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/network/
interfaces/macs/02:29:96:8f:6a:2d/subnet-id
subnet-be9b61d7
```

인스턴스에 대한 인스턴스 태그 가져오기

인스턴스 메타데이터의 인스턴스 태그에 대한 액세스가 설정된 경우 인스턴스 메타데이터에서 인스턴스의 태그를 가져올 수 있습니다. 자세한 내용은 인스턴스 메타데이터에서 태그 검색 섹션을 참조하세요.

인스턴스 메타데이터 서비스 버전 2 사용으로 전환

로컬 코드 또는 사용자가 인스턴스 메타데이터 서비스 버전 2(IMDSv2)를 사용해야 하도록 인스턴스를 마이그레이션하려면 다음 도구 및 전환 경로를 사용하는 것이 좋습니다.

주제

- IMDSv2로 전환하는 데 도움이 되는 도구
- IMDSv2를 요구하는 권장 경로

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 CLI 및 AWS SDK는 IMDSv2를 지원합니다. IMDSv2를 사용하려면 EC2 인스턴스에 최신 버전의 CLI 및 SDK가 있는지 확인합니다. CLI 업데이트에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 최신 버전의 AWS CLI 설치 또는 업데이트를 참조하세요.

모든 Amazon Linux 2 및 Amazon Linux 2023 소프트웨어 패키지에서 IMDSv2를 지원합니다. Amazon Linux 2023에서는 IMDSv1은 기본적으로 비활성화되어 있습니다.

IMDsv2를 지원하는 최소 AWS SDK 버전은 지원되는 AWS SDK 사용 섹션을 참조하세요.

IMDS 패킷 분석기

IMDS 패킷 분석기는 인스턴스의 부팅 단계에서 IMDSv1 호출을 식별하고 기록하는 오픈 소스 도구입니다. 이는 EC2 인스턴스에서 IMDSv1 호출을 수행하는 소프트웨어를 식별하는 데 도움이 되며, 이를 통해 인스턴스가 IMDSv2만 사용할 준비가 되도록 업데이트해야 하는 항목을 정확히 찾아낼 수 있습니다. 명령줄에서 IMDS 패킷 분석기를 실행하거나 서비스로 설치할 수 있습니다. 자세한 내용은 GitHub의 AWS ImdsPacketAnalyzer를 참조하세요.

CloudWatch

IMDSv2는 토큰 지원 세션을 사용하지만 IMDSv1은 사용하지 않습니다. `MetadataNoToken` CloudWatch 지표는 IMDSv1을 사용하는 인스턴스 메타데이터 서비스(IMDS)에 대한 호출 수를 추적합니다. 이 지표를 0까지 추적하면 모든 소프트웨어가 IMDSv2를 사용하도록 업그레이드되었는지 여부와 업그레이드된 시간을 확인할 수 있습니다.

IMDSv1을 비활성화한 후 `MetadataNoTokenRejected` CloudWatch 지표를 사용하여 IMDsv1 직접 호출이 시도되었지만 거부된 횟수를 추적할 수 있습니다. 이 지표를 추적하면 IMDSv2를 사용하기 위해 소프트웨어를 업데이트해야 하는지 여부를 확인할 수 있습니다.

자세한 내용은 인스턴스 지표 섹션을 참조하세요.

EC2 API 및 CLI 업데이트

새 인스턴스의 경우 RunInstances API를 사용하여 IMDSv2를 사용해야 하는 새 인스턴스를 시작할 수 있습니다. 자세한 내용은 새 인스턴스에 대한 인스턴스 메타데이터 옵션 구성 섹션을 참조하세요.

기존 인스턴스의 경우 ModifyInstanceMetadataOptions API를 사용하여 IMDSv2를 사용하도록 할 수 있습니다. 자세한 내용은 기존 인스턴스에 대한 인스턴스 메타데이터 옵션 수정 섹션을 참조하세요.

Auto Scaling 그룹에서 시작한 모든 새 인스턴스에서 IMDSv2를 사용해야 하는 경우 Auto Scaling 그룹에서 시작 템플릿 또는 시작 구성을 사용할 수 있습니다. 시작 템플릿을 생성하거나 시작 구성을 생성할 때 IMDSv2를 반드시 사용하도록 `MetadataOptions` 파라미터를 구성해야 합니다. Auto Scaling 그룹은 새 시작 템플릿 또는 시작 구성을 사용하여 새 인스턴스를 시작하지만 기존 인스턴스는 영향을 받지 않습니다. Auto Scaling 그룹의 기존 인스턴스의 경우 ModifyInstanceMetadataOptions API를 사용하여 기존 인스턴스에서 IMDSv2를 사용하도록 요구하거나 인스턴스를 종료하면 Auto Scaling 그룹이 새 시작 템플릿 또는 시작 구성에 정의된 인스턴스 메타데이터 옵션 설정으로 새 대체 인스턴스를 시작합니다.

기본적으로 IMDSv2를 구성하는 AMI 사용

인스턴스를 시작할 때 `v2.0`으로 설정된 `ImdsSupport` 파라미터로 구성한 AMI로 인스턴스를 시작하여 기본적으로 IMDSv2를 사용하도록 인스턴스를 자동 구성할 수 있습니다(`HttpTokens` 파라미터는 `required`로 설정됨). register-image CLI 명령을 사용하여 AMI를 등록할 때 `ImdsSupport` 파라미터를 `v2.0`로 설정하거나 modify-image-attribute CLI 명령을 사용하여 기존 AMI를 수정할 수 있습니다. 자세한 내용은 AMI 구성 섹션을 참조하세요.

IAM 정책 및 SCP

IAM 정책 또는 AWS Organizations 서비스 제어 정책(SCP)을 사용하여 다음과 같이 사용자를 제어할 수 있습니다.

- 인스턴스가 IMDSv2를 사용하도록 구성되어 있지 않으면 RunInstances API를 사용하여 인스턴스를 시작할 수 없습니다.
- IMDSv1을 다시 활성화하기 위해 ModifyInstanceMetadataOptions API를 사용하여 실행 중인 인스턴스를 수정할 수 없습니다.

IAM 정책 또는 SCP에 다음 IAM 조건 키가 포함되어야 합니다.

- `ec2:MetadataHttpEndpoint`
- `ec2:MetadataHttpPutResponseHopLimit`
- `ec2:MetadataHttpTokens`

API 또는 CLI 호출의 파라미터가 조건 키가 포함된 정책에 지정된 상태와 일치하지 않는 경우 API 또는 CLI 호출은 `UnauthorizedOperation` 응답과 함께 실패합니다.

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 `ec2:RoleDelivery`인 조건 키 `2.0`을 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 섹션을 참조하세요. SCP에 대한 자세한 내용을 알아보려면 AWS Organizations 사용 설명서의 서비스 제어 정책을 참조하세요.

IMDSv2를 요구하는 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 IMDSv2와 호환되는 버전으로 업데이트합니다. CLI 업데이트에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 최신 버전의 AWS CLI 설치 또는 업데이트를 참조하세요.

그런 다음, IMDSv2 요청을 사용하여 인스턴스 메타데이터에 직접 액세스하는(다시 말해서, SDK를 사용하지 않는) 소프트웨어를 변경합니다. IMDS 패킷 분석기로 IMDSv2 요청 사용을 위해 변경해야 하는 소프트웨어를 식별할 수 있습니다.

2단계: 전환 진행률 추적

CloudWatch 지표 `MetadataNoToken`을 사용하여 전환 진행률을 추적합니다. 이 지표는 인스턴스에서 IMDS에 대한 IMDSv1 호출 수를 표시합니다. 자세한 내용은 인스턴스 지표 섹션을 참조하세요.

3단계: IMDSv1 사용량이 0인 경우

CloudWatch 지표 `MetadataNoToken`이 IMDSv1 사용량을 0으로 기록하면 인스턴스가 IMDSv2 사용으로 완전히 전환할 준비가 된 것입니다. 이 단계에서 다음 작업을 수행할 수 있습니다.

- 계정 기본값

  IMDSv2를 반드시 계정 기본값으로 사용하도록 설정할 수 있습니다. 인스턴스가 시작되면 인스턴스 구성이 계정 기본값으로 자동 설정됩니다.

  계정 기본값을 설정하려면 다음을 수행합니다.

  - Amazon EC2 콘솔: EC2 대시보드의 계정 속성, 데이터 보호 및 보안 아래에서 IMDS 기본값에 대해 인스턴스 메타데이터 서비스를 활성화됨으로 설정하고 메타데이터 버전을 V2 전용(토큰 필요)으로 설정합니다. 자세한 내용은 IMDSv2를 계정 기본값으로 설정 섹션을 참조하세요.
  - AWS CLI: modify-instance-metadata-defaults CLI 명령을 사용하고 `--http-tokens required` 및 `--http-put-response-hop-limit 2`를 지정합니다.
- 새 인스턴스

  새 인스턴스를 시작할 때 다음을 수행할 수 있습니다.

  - Amazon EC2 콘솔: 인스턴스 시작 마법사에서 Metadata accessible(액세스 가능한 메타데이터)을 Enabled(사용)로 설정하고 Metadata version(메타데이터 버전)을 V2 only (token required)(V2 전용(토큰 필요))로 설정합니다. 자세한 내용은 시작 시 인스턴스 구성 섹션을 참조하세요.
  - AWS CLI: run-instances 명령을 사용하여 IMDSv2를 필수로 지정합니다.
- 기존 인스턴스

  기존 인스턴스의 경우 다음 작업을 수행할 수 있습니다.

  - Amazon EC2 콘솔: 인스턴스 페이지에서 인스턴스를 선택하고 작업, 인스턴스 설정, 인스턴스 메타데이터 수정 옵션을 선택하고 IMDsv2의 경우 필수를 선택합니다. 자세한 내용은 IMDSv2의 사용 요구 섹션을 참조하세요.
  - AWS CLI: modify-instance-metadata-options CLI 명령을 사용하여 IMDSv2만 사용하도록 지정합니다.

  실행 중인 인스턴스에서 인스턴스 메타데이터 옵션을 수정할 수 있으며 인스턴스 메타데이터 옵션을 수정한 후 인스턴스를 다시 시작할 필요가 없습니다.

4단계: 인스턴스가 IMDSv2로 전환되었는지 확인

인스턴스가 아직 IMDSv2를 사용하도록 구성되지 않았는지, 즉 IMDSv2가 여전히 `optional`로 구성되었는지 확인할 수 있습니다. 인스턴스가 여전히 `optional`로 구성된 경우 이전 3단계를 반복하여 인스턴스 메타데이터 옵션을 수정하여 IMDSv2 `required`를 만들 수 있습니다.

인스턴스를 필터링하려면 다음을 수행합니다.

- Amazon EC2 콘솔: 인스턴스 페이지에서 IMDSv2 = 선택 사항 필터를 사용하여 인스턴스를 필터링합니다. 필터링에 대한 자세한 내용은 콘솔을 사용하여 리소스 필터링 섹션을 참조하세요. 또한 각 인스턴스에 대해 IMDSv2가 필수인지 선택 사항인지 확인할 수 있습니다. 기본 설정 창에서 IMDSv2를 켜서 IMDSv2 열을 인스턴스 테이블에 추가하세요.

- AWS CLI: describe-instance 명령을 사용하고 다음과 같이 `metadata-options.http-tokens = optional`로 필터링합니다.

```
aws ec2 describe-instances --filters "Name=metadata-options.http-
tokens,Values=optional" --query "Reservations[*].Instances[*].[InstanceId]" --output
 text
```

5단계: 모든 인스턴스가 IMDSv2로 전환될 때

`ec2:MetadataHttpTokens`, `ec2:MetadataHttpPutResponseHopLimit`, `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 섹션을 참조하세요.

또한 IMDSv1을 비활성화한 후 `MetadataNoTokenRejected` CloudWatch 지표를 사용하여 IMDSv1 직접 호출이 시도되었지만 거부된 횟수를 추적할 수 있습니다. IMDSv1을 비활성화한 후 소프트웨어가 제대로 작동하지 않고 `MetadataNoTokenRejected` 지표에서 IMDSv1 직접 호출을 기록하는 경우 IMDSv2를 사용하려면 이 소프트웨어를 업데이트해야 할 수 있습니다.

인스턴스 메타데이터 서비스에 대한 액세스 제한

로컬 방화벽 규칙을 사용하여 인스턴스 메타데이터 서비스(IMDS)에 대한 일부 또는 모든 프로세스의 액세스를 비활성화할 수 있습니다.

Nitro 기반 인스턴스의 경우 가상 라우터와 같은 VPC 내의 네트워크 어플라이언스가 IMDS 주소로 패킷을 전달하고 인스턴스의 기본 원본/대상 확인이 비활성화되어 있으면 자체 네트워크에서 IMDS에 연결할 수 있습니다. VPC 외부 소스에서 IMDS에 연결할 수 없도록 하려면 대상 IPv4 주소가 IMDS `169.254.169.254`이고 IPv6 엔드포인트를 활성화한 경우 IPv6 주소가 IMDS `[fd00:ec2::254]`인 패킷을 삭제하도록 네트워크 어플라이언스의 구성을 수정하는 것이 좋습니다.

Linux 인스턴스에 대한 IMDS 액세스 제한

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 소프트웨어나 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 IMDS에 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 예에서는 IMDS에 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.

Windows 인스턴스에 대한 IMDS 액세스 제한

Windows 방화벽을 사용하여 액세스 제한

다음 PowerShell 예제에서는 기본 제공 Windows 방화벽을 사용하여 Internet Information Server 웹 서버(기본 설치 사용자 ID NT AUTHORITY\IUSR 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.

```
PS C:\> $blockPrincipal = New-Object -TypeName System.Security.Principal.NTAccount ("NT
 AUTHORITY\IUSR")
PS C:\> $BlockPrincipalSID =
 $blockPrincipal.Translate([System.Security.Principal.SecurityIdentifier]).Value
PS C:\> $BlockPrincipalSDDL = "D:(A;;CC;;;$BlockPrincipalSID)"
PS C:\> New-NetFirewallRule -DisplayName "Block metadata service from IIS" -Action
 block -Direction out `
-Protocol TCP -RemoteAddress 169.254.169.254 -LocalUser $BlockPrincipalSDDL
```

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 `blockPrincipal`에서 지정한 프로세스(이 예제에서는 `Everyone`라는 그룹)를 제외하고, 변수 `exceptionPrincipal`에서 지정한 OS 그룹으로 실행하는 모든 프로세스(이 예제에서는 Windows 그룹 `trustworthy-users`)가 인스턴스 메타데이터에 액세스할 수 없도록 설정합니다.

Linux iptables의 ! `--uid-owner trustworthy-user` 규칙과 달리 Windows 방화벽은 다른 모든 보안 주체를 거부하여 특정 보안 주체만 허용하는 바로 가기 메커니즘을 제공하지 않으므로, 보안 주체 거부 및 허용을 모두 지정해야 합니다.

```
PS C:\> $blockPrincipal = New-Object -TypeName System.Security.Principal.NTAccount
 ("Everyone")
PS C:\> $BlockPrincipalSID =
 $blockPrincipal.Translate([System.Security.Principal.SecurityIdentifier]).Value
PS C:\> $exceptionPrincipal = New-Object -TypeName System.Security.Principal.NTAccount
 ("trustworthy-users")
PS C:\> $ExceptionPrincipalSID =
 $exceptionPrincipal.Translate([System.Security.Principal.SecurityIdentifier]).Value
PS C:\> $PrincipalSDDL = "O:LSD:(D;;CC;;;$ExceptionPrincipalSID)(A;;CC;;;
$BlockPrincipalSID)"
PS C:\> New-NetFirewallRule -DisplayName "Block metadata service for
 $($blockPrincipal.Value), exception: $($exceptionPrincipal.Value)" -Action block -
Direction out `
-Protocol TCP -RemoteAddress 169.254.169.254 -LocalUser $PrincipalSDDL
```

> **Note**
>
> 로컬 방화벽 규칙을 사용하려면 이전 예제 명령을 필요에 맞게 조정해야 합니다.

netsh 규칙을 사용하여 액세스 제한

`netsh` 규칙을 사용하여 모든 소프트웨어를 차단할 수 있지만, 이러한 규칙은 훨씬 더 유연합니다.

```
C:\> netsh advfirewall firewall add rule name="Block metadata service altogether"
 dir=out protocol=TCP remoteip=169.254.169.254 action=block
```

> **Note**
>
> - 로컬 방화벽 규칙을 사용하려면 이전 예제 명령을 필요에 맞게 조정해야 합니다.
> - `netsh` 규칙은 승격된 명령 프롬프트에서 설정해야 하며, 특정 보안 주체를 거부하거나 허용하도록 설정할 수 없습니다.

인스턴스 메타데이터 서비스 옵션 구성

인스턴스 메타데이터 서비스(IMDS)는 모든 EC2 인스턴스에서 로컬로 실행됩니다. 인스턴스 메타데이터 옵션은 EC2 인스턴스에서 IMDS의 액세스 가능성과 동작을 제어하는 일련의 구성입니다.

각 인스턴스에서 다음 인스턴스 메타데이터 옵션을 구성할 수 있습니다.

인스턴스 메타데이터 서비스(IMDS): `enabled` | `disabled`

인스턴스에서 IMDS를 활성화 또는 비활성화할 수 있습니다. 비활성화하면 사용자 또는 다른 코드에서 인스턴스의 인스턴스 메타데이터에 액세스할 수 없습니다.

IMDS에는 인스턴스에 IPv4(`169.254.169.254`) 및 IPv6(`[fd00:ec2::254]`)이라는 두 개의 엔드포인트가 있습니다. IMDS를 활성화하면 IPv4 엔드포인트가 자동으로 활성화됩니다. IPv6 엔드포인트를 활성화하려면 명시적으로 활성화해야 합니다.

IMDS IPv6 엔드포인트: `enabled` | `disabled`

인스턴스에서 IPv6 IMDS 엔드포인트를 명시적으로 활성화할 수 있습니다. IPv6 엔드포인트가 활성화된 경우 IPv4 엔드포인트는 활성화된 상태로 유지됩니다. IPv6 엔드포인트는 IPv6 지원 서브넷(듀얼 스택 또는 IPv6 전용)의 Nitro 기반 인스턴스에서만 지원됩니다.

메타데이터 버전: `IMDSv1 or IMDSv2 (token optional)` | `IMDSv2 only (token required)`

인스턴스 메타데이터를 요청할 때 IMDSv2를 직접 호출하려면 토큰이 필요합니다. IMDSv1 직접 호출에는 토큰이 필요하지 않습니다. IMDSv1 또는 IMDSv2 직접 호출(토큰은 선택 사항)을 허용하거나 IMDSv2 직접 호출만(토큰은 필수) 허용하도록 인스턴스를 구성할 수 있습니다.

메타데이터 응답 홉 제한: 1–64

홉 제한은 PUT 응답에 허용된 네트워크 홉 수입니다. 홉 제한을 최소 1 및 최대 64로 설정할 수 있습니다. 컨테이너 환경에서는 홉 제한 1이 문제를 일으킬 수 있습니다. 이러한 문제를 완화하는 방법에 대한 자세한 내용은 인스턴스 메타데이터 액세스 고려 사항 아래의 컨테이너 환경 정보를 참조하세요.

인스턴스 메타데이터에서 태그에 대한 액세스: `enabled` | `disabled`

인스턴스 메타데이터에서 인스턴스의 태그에 대한 액세스를 활성화하거나 비활성화할 수 있습니다. 자세한 내용은 인스턴스 메타데이터를 사용하여 EC2 인스턴스의 태그 보기 섹션을 참조하세요.

인스턴스의 현재 구성을 보려면 기존 인스턴스에 대한 인스턴스 메타데이터 옵션 쿼리 섹션을 참조하세요.

인스턴스 메타데이터 옵션을 구성하는 위치

인스턴스 메타데이터 옵션은 다음과 같이 다양한 수준에서 구성할 수 있습니다.

- 계정 - 각 AWS 리전의 계정 수준에서 인스턴스 메타데이터 옵션의 기본값을 설정할 수 있습니다. 인스턴스가 시작되면 인스턴스 메타데이터 옵션이 계정 수준 값으로 자동 설정됩니다. 이러한 값은 시작할 때 변경할 수 있습니다. 계정 수준의 기본값은 기존 인스턴스에 영향을 주지 않습니다.
- AMI - AMI를 등록하거나 수정할 때 `imds-support` 파라미터를 `v2.0`으로 설정할 수 있습니다. 이 AMI로 인스턴스를 시작하면 인스턴스 메타데이터 버전이 자동으로 IMDSv2로 설정되고 홉 제한은 2로 설정됩니다.
- 인스턴스 - 시작할 때 인스턴스의 모든 인스턴스 메타데이터 옵션을 변경하여 기본 설정을 재정의할 수 있습니다. 실행 중이거나 중지된 인스턴스에서 시작 후 인스턴스 메타데이터 옵션을 변경할 수도 있습니다. 단, IAM 또는 SCP 정책에 따라 변경이 제한될 수 있습니다.

자세한 내용은 새 인스턴스에 대한 인스턴스 메타데이터 옵션 구성 및 기존 인스턴스에 대한 인스턴스 메타데이터 옵션 수정(을)를 참조하세요.

인스턴스 메타데이터 옵션의 우선순위

각 인스턴스 메타데이터 옵션의 값은 계층적 우선순위에 따라 인스턴스를 시작할 때 결정됩니다. 계층 구조는 다음과 같습니다(맨 위의 우선순위가 가장 높음).

- 우선순위 1: 시작 시 인스턴스 구성 - 시작 템플릿 또는 인스턴스 구성에서 값을 지정할 수 있습니다. 여기에 지정된 모든 값은 계정 수준 또는 AMI에서 지정된 값을 재정의합니다.
- 우선순위 2: 계정 설정 - 인스턴스 시작 시 값을 지정하지 않은 경우 계정 수준 설정(각 AWS 리전에 대해 설정됨)에 따라 값이 결정됩니다. 계정 수준 설정은 각 메타데이터 옵션의 값을 포함하거나 기본 설정이 없음을 나타냅니다.
- 우선순위 3: AMI 구성 - 인스턴스 시작 시 또는 계정 수준에서 값이 지정되지 않은 경우 AMI 구성에 따라 값이 결정됩니다. 이것은 `HttpTokens` 및 `HttpPutResponseHopLimit`에만 적용됩니다.

각 메타데이터 옵션은 개별적으로 평가됩니다. 인스턴스는 직접 인스턴스 구성, 계정 수준 기본값 및 AMI의 구성을 혼합하여 구성할 수 있습니다.

IAM 또는 SCP 정책에 의해 변경이 제한되지 않는 한, 실행 중이거나 중지된 인스턴스에서 시작 후 모든 메타데이터 옵션의 값을 변경할 수 있습니다.

메타데이터 옵션 값 결정 - 예 1

이 예에서는 계정 수준에서 HttpPutResponseHopLimit가 1로 설정된 리전에서 EC2 인스턴스가 시작됩니다. 지정된 AMI에서 ImdsSupport가 v2.0으로 설정되었습니다. 시작할 때 인스턴스에서 직접 메타데이터 옵션이 지정되지 않습니다. 인스턴스는 다음 메타데이터 옵션으로 시작됩니다.

```
"MetadataOptions": {
    ...
    "HttpTokens": "required",
    "HttpPutResponseHopLimit": 1,
    ...
```

이러한 값은 다음과 같이 결정됩니다.

- 시작 시 지정된 메타데이터 옵션 없음: 인스턴스를 시작하는 동안 인스턴스 시작 파라미터나 시작 템플릿에 메타데이터 옵션의 특정 값이 제공되지 않습니다.
- 계정 설정이 다음 우선순위: 시작 시 지정된 특정 값이 없는 경우 리전 내 계정 수준의 설정이 우선됩니다. 즉, 계정 수준에서 구성된 기본값이 적용됩니다. 이 경우에는 HttpPutResponseHopLimit가 1로 설정되었습니다.
- AMI 설정이 마지막 우선순위: 시작 시 또는 HttpTokens에 대한 계정 수준(인스턴스 메타데이터 버전)에서 특정 값이 지정되지 않은 경우 AMI 설정이 적용됩니다. 이 경우 AMI 설정 ImdsSupport: v2.0은 HttpTokens가 required로 설정되었음을 확인했습니다. AMI 설정 ImdsSupport: v2.0은 HttpPutResponseHopLimit: 2를 설정하도록 설계되었지만 우선순위가 더 높은 계정 수준 설정 HttpPutResponseHopLimit: 1로 재정의되었습니다.

메타데이터 옵션 값 결정 - 예 2

이 예에서 EC2 인스턴스는 이전 예 1과 동일한 설정으로 시작되지만 시작 시 인스턴스에서 HttpTokens가 optional로 직접 설정되어 있습니다. 인스턴스는 다음 메타데이터 옵션으로 시작됩니다.

```
"MetadataOptions": {
    ...
    "HttpTokens": "optional",
    "HttpPutResponseHopLimit": 1,
    ...
```

HttpPutResponseHopLimit의 값은 예 1과 같은 방식으로 결정됩니다. 하지만 HttpTokens의 값은 다음과 같이 결정됩니다. 시작할 때 인스턴스에서 구성된 메타데이터 옵션이 우선 적용됩니다. AMI

가 ImdsSupport: v2.0(즉, HttpTokens가 required로 설정됨)으로 구성되어 있더라도 시작 시 인스턴스에 지정된 값(HttpTokens가 optional로 설정됨)이 우선됩니다.

인스턴스 메타데이터 버전 설정

인스턴스 시작 시 인스턴스 메타데이터 버전 값은 IMDSv1 or IMDSv2 (token optional) 또는 IMDSv2 only (token required)입니다.

인스턴스를 시작할 때 메타데이터 버전의 값을 수동으로 지정하거나 기본값을 사용할 수 있습니다. 값을 수동으로 지정하면 해당 값이 기본값을 재정의합니다. 값을 수동으로 지정하지 않는 경우 다음 테이블에 설명된 대로 기본 설정의 조합에 따라 값이 결정됩니다.

이 테이블에서는 시작 시 인스턴스의 메타데이터 버전(4번째 열의 결과 인스턴스 구성으로 표시됨)이 다양한 구성 수준의 설정에 따라 어떻게 결정되는지를 보여줍니다. 우선순위는 왼쪽에서 오른쪽 순서이며, 다음과 같이 첫 번째 열이 우선순위가 가장 높습니다.

- 열 1: 시작 파라미터 - 시작 시 수동으로 지정하는 인스턴스의 설정을 나타냅니다.
- 열 2: 계정 수준 기본값 - 계정 설정을 나타냅니다.
- 열 3: AMI 기본값 - AMI의 설정을 나타냅니다.

시작 파라미터	계정 수준 기본값	AMI 기본값	결과 인스턴스 구성
V2 전용(토큰 필요)	기본 설정 없음	V2 전용	V2 전용
V2 전용(토큰 필요)	V2 전용	V2 전용	V2 전용
V2 전용(토큰 필요)	V1 또는 V2	V2 전용	V2 전용
V1 또는 V2(토큰 선택 사항)	기본 설정 없음	V2 전용	V1 또는 V2
V1 또는 V2(토큰 선택 사항)	V2 전용	V2 전용	V1 또는 V2
V1 또는 V2(토큰 선택 사항)	V1 또는 V2	V2 전용	V1 또는 V2
설정되지 않음	기본 설정 없음	V2 전용	V2 전용

시작 파라미터	계정 수준 기본값	AMI 기본값	결과 인스턴스 구성
설정되지 않음	V2 전용	V2 전용	V2 전용
설정되지 않음	V1 또는 V2	V2 전용	V1 또는 V2
V2 전용(토큰 필요)	기본 설정 없음	null	V2 전용
V2 전용(토큰 필요)	V2 전용	null	V2 전용
V2 전용(토큰 필요)	V1 또는 V2	null	V2 전용
V1 또는 V2(토큰 선택 사항)	기본 설정 없음	null	V1 또는 V2
V1 또는 V2(토큰 선택 사항)	V2 전용	null	V1 또는 V2
V1 또는 V2(토큰 선택 사항)	V1 또는 V2	null	V1 또는 V2
설정되지 않음	기본 설정 없음	null	V1 또는 V2
설정되지 않음	V2 전용	null	V2 전용
설정되지 않음	V1 또는 V2	null	V1 또는 V2

IAM 조건 키를 사용하여 인스턴스 메타데이터 옵션 제한

다음과 같이 IAM 정책 또는 SCP에서 IAM 조건 키를 사용할 수 있습니다.

- IMDSv2를 사용해야 하도록 구성된 경우에만 인스턴스를 시작하도록 허용
- 허용된 홉 수 제한
- 인스턴스 메타데이터에 대한 액세스 비활성화

업무

- 새 인스턴스에 대한 인스턴스 메타데이터 옵션 구성
- 기존 인스턴스에 대한 인스턴스 메타데이터 옵션 수정

> **Note**
>
> 작업을 조심스럽게 진행해야 하며 무엇이든 변경하기 전에 세심하게 테스트해야 합니다. 다음에 유의하세요.
>
> - 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.
> - IMDSv2의 경우 토큰을 검색할 때 `/latest/api/token`을 사용해야 합니다.
> - (Windows만 해당) PowerShell 버전이 4.0 이전 버전인 경우 IMDSv2를 사용하려면 Windows Management Framework 4.0으로 업데이트해야 합니다.

새 인스턴스에 대한 인스턴스 메타데이터 옵션 구성

새 인스턴스에서 다음과 같은 인스턴스 메타데이터 옵션을 구성할 수 있습니다.

옵션

- IMDSv2의 사용 요구
- IMDS IPv4 및 IPv6 엔드포인트 활성화
- 인스턴스 메타데이터에 대한 액세스 비활성화
- 인스턴스 메타데이터의 태그에 대한 액세스 허용

> **Note**
>
> 이러한 옵션에 대한 설정은 계정 수준에서 직접 구성되거나 선언적 정책을 사용하여 구성됩니다. 이 설정은 인스턴스 메타데이터 옵션을 구성하려는 각 AWS 리전에서 구성해야 합니다. 선언적 정책을 사용하면 여러 리전과 여러 계정에 동시에 설정을 적용할 수 있습니다. 선언적 정책을 사용 중인 경우 계정 내에서 직접 설정을 수정할 수 없습니다. 이 주제에서는 계정 내에서 직접 설정을 구성하는 방법을 설명합니다. 선언적 정책 사용에 대한 자세한 내용은 AWS Organizations 사용 설명서의 선언적 정책을 참조하세요.

IMDSv2의 사용 요구

다음 방법을 사용하여 새 인스턴스에서 IMDSv2 사용을 요구할 수 있습니다.

IMDSv2를 사용하도록 설정

- IMDSv2를 계정 기본값으로 설정
- 시작 시 인스턴스 구성
- AMI 구성
- IAM 정책 사용

IMDSv2를 계정 기본값으로 설정

각 AWS 리전의 계정 수준에서 인스턴스 메타데이터 서비스(IMDS)의 기본 버전을 설정할 수 있습니다. 즉, 새 인스턴스가 시작되면 인스턴스 메타데이터 버전이 자동으로 계정 수준 기본값으로 설정됩니다. 하지만 시작 시 또는 시작한 후에 값을 수동으로 재정의할 수 있습니다. 계정 수준 설정 및 수동 재정의가 인스턴스에 미치는 영향에 대한 자세한 내용은 인스턴스 메타데이터 옵션의 우선순위 섹션을 참조하세요.

> **Note**
>
> 계정 수준 기본값을 설정해도 기존 인스턴스는 재설정되지 않습니다. 예를 들어, 계정 수준 기본값을 IMDSv2로 설정하는 경우 IMDSv1로 설정된 기존 인스턴스는 영향을 받지 않습니다. 기존 인스턴스의 값을 변경하려면 인스턴스 자체의 값을 수동으로 변경해야 합니다.

계정의 모든 새 인스턴스가 필요한 IMDSv2로 시작되도록(IMDSv1은 비활성화됨) 인스턴스 메타데이터 버전의 계정 기본값을 IMDSv2로 설정할 수 있습니다. 이 계정 기본값을 사용하면 인스턴스를 시작할 때 인스턴스의 기본값은 다음과 같습니다.

- 콘솔: 메타데이터 버전은 V2 전용(토큰 필요)으로 설정되고 메타데이터 응답 홉 제한은 2로 설정됩니다.
- AWS CLI: `HttpTokens`는 `required`로 설정되고 `HttpPutResponseHopLimit`는 2로 설정됩니다.

> **Note**
>
> 계정 기본값을 IMDSv2로 설정하기 전에 인스턴스가 IMDSv1에 종속되지 않았는지 확인합니다. 자세한 내용은 IMDSv2를 요구하는 권장 경로 섹션을 참조하세요.

Console

IMDSv2를 지정된 리전의 계정 기본값으로 설정하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. AWS 리전을(를) 변경하려면 페이지의 오른쪽 상단 모서리에 있는 리전 선택기를 사용합니다.
3. 탐색 창에서 EC2 대시보드를 선택합니다.
4. 계정 속성에서 데이터 보호 및 보안을 선택합니다.
5. IMDS 기본값 옆에서 관리를 선택합니다.
6. IMDS 기본값 관리 페이지에서 다음을 수행합니다.
   a. 인스턴스 메타데이터 서비스에서 활성화됨을 선택합니다.
   b. 메타데이터 버전(Metadata version)에 V2만 해당(토큰 필요)(V2 only (token required))를 선택합니다.
   c. 메타데이터 응답 홉 제한에서 인스턴스가 컨테이너를 호스팅하는 경우 2를 지정합니다. 그렇지 않으면 기본 설정 없음을 선택합니다. 기본 설정이 지정되지 않은 경우 AMI에 IMDSv2가 필요하면 시작 시 기본값은 2이고, 그렇지 않으면 기본값은 1입니다.
   d. 업데이트를 선택합니다.

AWS CLI

IMDSv2를 지정된 리전의 계정 기본값으로 설정하는 방법

modify-instance-metadata-defaults 명령을 사용하고 IMDS 계정 수준 설정을 수정할 리전을 지정합니다. 인스턴스가 컨테이너를 호스팅할 경우 `--http-tokens`(`required`로 설정됨) 및 `--http-put-response-hop-limit`(`2`로 설정됨)를 포함합니다. 그렇지 않으면 기본 설정 없음을 표시하도록 `-1`을 지정합니다. `-1`(기본 설정 없음)이 지정된 경우 AMI에 IMDSv2가 필요하면 시작 시 기본값은 `2`이고, 그렇지 않으면 `1`입니다.

```
aws ec2 modify-instance-metadata-defaults \
    --region us-east-1 \
```

```
    --http-tokens required \
    --http-put-response-hop-limit 2
```

출력의 예시는 다음과 같습니다.

```
{
    "Return": true
}
```

지정된 리전의 인스턴스 메타데이터 옵션에 대한 기본 계정 설정을 보는 방법

get-instance-metadata-defaults 명령을 사용하고 리전을 지정합니다.

```
aws ec2 get-instance-metadata-defaults --region us-east-1
```

출력의 예시는 다음과 같습니다.

```
{
    "AccountLevel": {
        "HttpTokens": "required",
        "HttpPutResponseHopLimit": 2
    },
    "ManagedBy": "account"
}
```

ManagedBy 필드는 설정을 구성한 엔터티를 나타냅니다. 이 예제의 account는 해당 설정이 계정에서 직접 구성되었음을 나타냅니다. 값이 declarative-policy이면 설정이 선언적 정책에 의해 구성되었음을 의미합니다. 자세한 내용은 AWS Organizations 사용 설명서의 선언적 정책을 참조하세요.

IMDSv2를 모든 리전의 계정 기본값으로 설정하려면 다음을 수행하세요.

modify-instance-metadata-defaults 명령을 사용하여 모든 리전의 IMDS 계정 수준 설정을 수정합니다. 인스턴스가 컨테이너를 호스팅할 경우 --http-tokens(required로 설정됨) 및 --http-put-response-hop-limit(2로 설정됨)를 포함합니다. 그렇지 않으면 기본 설정 없음을 표시하도록 -1을 지정합니다. -1(기본 설정 없음)이 지정된 경우 AMI에 IMDSv2가 필요하면 시작 시 기본값은 2이고, 그렇지 않으면 1입니다.

```
echo -e "Region          \t Modified" ; \
echo -e "--------------  \t ---------" ; \
```

```
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
        aws ec2 modify-instance-metadata-defaults \
            --region $region \
            --http-tokens required \
            --http-put-response-hop-limit 2 \
            --output text)
        echo -e "$region        \t $output"
    );
done
```

출력의 예시는 다음과 같습니다.

```
Region                  Modified
--------------          ---------
ap-south-1              True
eu-north-1              True
eu-west-3               True
...
```

모든 리전의 인스턴스 메타데이터 옵션에 대한 기본 계정 설정을 보려면 다음을 수행하세요.

get-instance-metadata-defaults 명령을 사용합니다.

```
echo -e "Region   \t Level          Hops    HttpTokens" ; \
echo -e "-------------- \t ------------   ----    ----------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
        aws ec2 get-instance-metadata-defaults \
            --region $region \
            --output text)
        echo -e "$region \t $output"
    );
```

```
done
```

출력의 예시는 다음과 같습니다.

```
Region           Level          Hops    HttpTokens
--------------   ------------   ----    ----------
ap-south-1       ACCOUNTLEVEL   2       required
eu-north-1       ACCOUNTLEVEL   2       required
eu-west-3        ACCOUNTLEVEL   2       required
...
```

PowerShell

IMDSv2를 지정된 리전의 계정 기본값으로 설정하는 방법

Edit-EC2InstanceMetadataDefault 명령을 사용하고 IMDS 계정 수준 설정을 수정할 리전을 지정합니다. 인스턴스가 컨테이너를 호스팅할 경우 -HttpToken(required로 설정됨) 및 -HttpPutResponseHopLimit(2로 설정됨)를 포함합니다. 그렇지 않으면 기본 설정 없음을 표시하도록 -1을 지정합니다. -1(기본 설정 없음)이 지정된 경우 AMI에 IMDSv2가 필요하면 시작 시 기본값은 2이고, 그렇지 않으면 1입니다.

```
Edit-EC2InstanceMetadataDefault `
    -Region us-east-1 `
    -HttpToken required `
    -HttpPutResponseHopLimit 2
```

출력의 예시는 다음과 같습니다.

```
True
```

지정된 리전의 인스턴스 메타데이터 옵션에 대한 기본 계정 설정을 보는 방법

Get-EC2InstanceMetadataDefault 명령을 사용하고 리전을 지정합니다.

```
Get-EC2InstanceMetadataDefault -Region us-east-1 | Format-List
```

출력의 예시는 다음과 같습니다.

```
HttpEndpoint            :
```

```
HttpPutResponseHopLimit : 2
HttpTokens              : required
InstanceMetadataTags    :
```

IMDSv2를 모든 리전의 계정 기본값으로 설정하려면 다음을 수행하세요.

Edit-EC2InstanceMetadataDefault Cmdlet을 사용하여 모든 리전의 IMDS 계정 수준 설정을 수정합니다. 인스턴스가 컨테이너를 호스팅할 경우 -HttpToken(required로 설정됨) 및 -HttpPutResponseHopLimit(2로 설정됨)를 포함합니다. 그렇지 않으면 기본 설정 없음을 표시하도록 -1을 지정합니다. -1(기본 설정 없음)이 지정된 경우 AMI에 IMDSv2가 필요하면 시작 시 기본값은 2이고, 그렇지 않으면 1입니다.

```
(Get-EC2Region).RegionName | `
    ForEach-Object {
    [PSCustomObject]@{
        Region   = $_
        Modified = (Edit-EC2InstanceMetadataDefault `
                -Region $_ `
                -HttpToken required `
                -HttpPutResponseHopLimit 2)
    }
} | `
Format-Table Region, Modified -AutoSize
```

예상 결과

```
Region         Modified
------         --------
ap-south-1         True
eu-north-1         True
eu-west-3          True
...
```

모든 리전의 인스턴스 메타데이터 옵션에 대한 기본 계정 설정을 보려면 다음을 수행하세요.

Get-EC2InstanceMetadataDefault Cmdlet을 사용합니다.

```
(Get-EC2Region).RegionName | `
    ForEach-Object {
    [PSCustomObject]@{
        Region = $_
```

```
        HttpPutResponseHopLimit = (Get-EC2InstanceMetadataDefault -Region
 $_).HttpPutResponseHopLimit
        HttpTokens              = (Get-EC2InstanceMetadataDefault -Region
 $_).HttpTokens
    }
} | `
Format-Table -AutoSize
```

출력 예시

```
Region         HttpPutResponseHopLimit HttpTokens
------         ----------------------- ----------
ap-south-1                           2 required
eu-north-1                           2 required
eu-west-3                            2 required
...
```

시작 시 인스턴스 구성

인스턴스를 시작할 때 다음 필드를 구성하여 IMDSv2를 사용하도록 인스턴스를 구성할 수 있습니다.

- Amazon EC2 콘솔: Metadata version(메타데이터 버전)을 V2 only (token required)(V2 전용(토큰 필요))로 설정합니다.
- AWS CLI: `HttpTokens`를 `required`로 설정합니다.

IMDSv2를 사용하도록 지정하면 메타데이터 액세스 가능을 활성화됨(콘솔)로 설정하거나 `HttpEndpoint`를 `enabled`(AWS CLI)로 설정하여 인스턴스 메타데이터 서비스(IMDS) 엔드포인트도 활성화해야 합니다.

컨테이너 환경에서는 IMDSv2가 필요한 경우 홉 제한을 2로 설정하는 것이 좋습니다. 자세한 내용은 인스턴스 메타데이터 액세스 고려 사항 섹션을 참조하세요.

Console

새 인스턴스에서 IMDSv2를 사용해야 하도록 설정하려면

- Amazon EC2 콘솔에서 새 인스턴스를 시작할 때 고급 세부 정보를 확장하고 다음을 수행합니다.
  - 액세스 가능한 메타데이터(Metadata accessible)에 활성화(Enabled)를 선택합니다.

- 메타데이터 버전(Metadata version)에 V2만 해당(토큰 필요)(V2 only (token required))를 선택합니다.
- (컨테이너 환경) 메타데이터 응답 홉 제한의 경우 2를 선택합니다.

자세한 내용은 고급 세부 정보 섹션을 참조하세요.

AWS CLI

새 인스턴스에서 IMDSv2를 사용해야 하도록 설정하려면

다음 run-instances 예에서는 `c6i.large`를 `--metadata-options`로 설정하여 `HttpTokens=required` 인스턴스를 시작합니다. 또한 `HttpTokens`의 값을 지정할 때 `HttpEndpoint`를 `enabled`로 설정해야 합니다. 메타데이터 검색 요청에 대해 보안 토큰 헤더가 `required`로 설정되어 있으므로 인스턴스 메타데이터를 요청할 때 인스턴스가 IMDSv2를 사용해야 합니다.

컨테이너 환경에서는 IMDSv2가 필요한 경우 `HttpPutResponseHopLimit=2`를 사용하여 홉 제한을 2로 설정하는 것이 좋습니다.

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type c6i.large \
 ...
    --metadata-options
 "HttpEndpoint=enabled,HttpTokens=required,HttpPutResponseHopLimit=2"
```

PowerShell

새 인스턴스에서 IMDSv2를 사용해야 하도록 설정하려면

다음 New-EC2Instance Cmdlet 예에서는 `MetadataOptions_HttpEndpoint`를 `enabled`로 설정하고 `MetadataOptions_HttpTokens` 파라미터를 `required`로 설정하여 `c6i.large` 인스턴스를 시작합니다. 또한 `HttpTokens`의 값을 지정할 때 `HttpEndpoint`를 `enabled`로 설정해야 합니다. 메타데이터 검색 요청에 대해 보안 토큰 헤더가 `required`로 설정되어 있으므로 인스턴스 메타데이터를 요청할 때 인스턴스가 IMDSv2를 사용해야 합니다.

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
```

```
    -InstanceType c6i.large `
    -MetadataOptions_HttpEndpoint enabled `
    -MetadataOptions_HttpTokens required
```

AWS CloudFormation

AWS CloudFormation을 사용하여 인스턴스에 대한 메타데이터 옵션을 지정하려면 AWS CloudFormation 사용 설명서의 AWS::EC2::LaunchTemplate MetadataOptions 속성을 참조하세요.

AMI 구성

새 AMI를 등록하거나 기존 AMI를 수정할 때 `imds-support` 파라미터를 `v2.0`로 설정할 수 있습니다. 이 AMI에서 시작된 인스턴스에 대해 메타데이터 버전은 V2 전용(토큰 필요)(콘솔)으로 설정되거나 `HttpTokens`가 `required`(AWS CLI)로 설정됩니다. 이러한 설정을 사용하면 인스턴스에서 인스턴스 메타데이터를 요청할 때 IMDSv2를 사용하도록 지정됩니다.

`imds-support`를 `v2.0`으로 설정하면 이 AMI에서 시작된 인스턴스의 Metadata response hop limit(메타데이터 응답 홉 제한)(콘솔) 또는 `http-put-response-hop-limit`(AWS CLI)도 2로 설정됩니다.

Important

AMI 소프트웨어가 IMDSv2를 지원하지 않는 한 이 파라미터를 사용하지 마세요. 값을 `v2.0`으로 설정한 후에는 이를 실행 취소할 수 없습니다. AMI를 "재설정"하는 유일한 방법은 기본 스냅샷에 새 AMI를 생성하는 것입니다.

IMDSv2를 위해 새 AMI 구성

다음 방법 중 하나를 사용하여 IMDsv2에 대한 새 AMI를 구성합니다.

AWS CLI

다음 register-image 예제는 EBS 루트 볼륨의 지정된 스냅샷을 디바이스 `/dev/xvda`로 사용하여 AMI를 등록합니다. 이 AMI에서 시작된 인스턴스가 인스턴스 메타데이터를 요청할 때 IMDSv2를 사용하도록 `imds-support` 파라미터를 `v2.0`으로 지정합니다.

```
aws ec2 register-image \
    --name my-image \
```

```
    --root-device-name /dev/xvda \
    --block-device-mappings DeviceName=/dev/
xvda,Ebs={SnapshotId=snap-0123456789example} \
    --architecture x86_64 \
    --imds-support v2.0
```

PowerShell

다음 Register-EC2Image Cmdlet 예제는 EBS 루트 볼륨의 지정된 스냅샷을 디바이스 /dev/xvda로 사용하여 AMI를 등록합니다. 이 AMI에서 시작된 인스턴스가 인스턴스 메타데이터를 요청할 때 IMDSv2를 사용하도록 ImdsSupport 파라미터를 v2.0으로 지정합니다.

```
Register-EC2Image `
    -Name 'my-image' `
    -RootDeviceName /dev/xvda `
    -BlockDeviceMapping  (
    New-Object `
        -TypeName Amazon.EC2.Model.BlockDeviceMapping `
        -Property @{
        DeviceName = '/dev/xvda';
        EBS        = (New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice -Property
 @{
                SnapshotId = 'snap-0123456789example'
                VolumeType = 'gp3'
                } )
        }  ) `
    -Architecture X86_64 `
    -ImdsSupport v2.0
```

IMDSv2를 위해 기존 AMI 구성

다음 방법 중 하나를 사용하여 IMDSv2용 기존 AMI를 구성합니다.

AWS CLI

다음 modify-image-attribute 예제는 IMDSv2에 대해서만 기존 AMI를 수정합니다. 이 AMI에서 시작된 인스턴스가 인스턴스 메타데이터를 요청할 때 IMDSv2를 사용하도록 imds-support 파라미터를 v2.0으로 지정합니다.

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
```

```
    --imds-support v2.0
```

PowerShell

다음 Edit-EC2ImageAttribute Cmdlet 예제는 IMDSv2에 대해서만 기존 AMI를 수정합니다. 이 AMI에서 시작된 인스턴스가 인스턴스 메타데이터를 요청할 때 IMDSv2를 사용하도록 `imds-support` 파라미터를 `v2.0`으로 지정합니다.

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -ImdsSupport 'v2.0'
```

IAM 정책 사용

사용자가 새 인스턴스에서 IMDSv2를 사용하도록 지정하지 않는 경우 새 인스턴스를 시작하지 못하게 하는 IAM 정책을 생성할 수 있습니다.

IAM 정책을 사용하여 모든 새 인스턴스에서 IMDSv2를 사용해야 하도록 설정

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 섹션을 참조하세요.

IMDS IPv4 및 IPv6 엔드포인트 활성화

IMDS에는 인스턴스에 IPv4(`169.254.169.254`) 및 IPv6(`[fd00:ec2::254]`)이라는 두 개의 엔드포인트가 있습니다. IMDS를 활성화하면 IPv4 엔드포인트가 자동으로 활성화됩니다. IPv6 전용 서브넷으로 인스턴스를 시작하는 경우에도 IPv6 엔드포인트는 비활성화된 상태로 유지됩니다. IPv6 엔드포인트를 활성화하려면 명시적으로 활성화해야 합니다. IPv6 엔드포인트가 활성화되면 IPv4 엔드포인트는 활성화된 상태로 유지됩니다.

인스턴스 시작 시 또는 이후에 IPv6 엔드포인트를 활성화할 수 있습니다.

IPv6 엔드포인트 활성화 요구

- 선택한 인스턴스 유형은 Nitro 기반 인스턴스입니다.
- 선택한 서브넷은 IPv6를 지원합니다. 여기서 서브넷은 이중 스택 또는 IPv6 전용입니다.

다음 방법 중 하나를 사용하여 IMDS IPv6 엔드포인트가 활성화된 상태로 인스턴스를 시작합니다.

Console

인스턴스 시작 시 IMDS IPv6 엔드포인트를 활성화하는 방법

- Advanced details(고급 세부 정보)에서 다음을 지정하여 Amazon EC2 콘솔에서 인스턴스를 시작합니다.
  - 메타데이터 IPv6 엔드포인트에 대해 활성화됨을 선택합니다.

자세한 내용은 고급 세부 정보 섹션을 참조하세요.

AWS CLI

인스턴스 시작 시 IMDS IPv6 엔드포인트를 활성화하는 방법

다음 run-instances 예제는 IMDS에 대해 활성화된 IPv6 엔드포인트로 `c6i.large` 인스턴스를 시작합니다. IPv6 엔드포인트를 활성화하려면 `--metadata-options` 파라미터에 대해 `HttpProtocolIpv6=enabled`를 지정합니다. 또한 `HttpProtocolIpv6`의 값을 지정할 때 `HttpEndpoint`를 `enabled`로 설정해야 합니다.

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type c6i.large \
    ...
    --metadata-options "HttpEndpoint=enabled,HttpProtocolIpv6=enabled"
```

PowerShell

인스턴스 시작 시 IMDS IPv6 엔드포인트를 활성화하는 방법

다음 New-EC2Instance Cmdlet 예제는 IMDS에 대해 활성화된 IPv6 엔드포인트로 `c6i.large` 인스턴스를 시작합니다. IPv6 엔드포인트를 활성화하려면 `MetadataOptions_HttpProtocolIpv6`를 `enabled`로 지정합니다. 또한 `MetadataOptions_HttpProtocolIpv6`의 값을 지정할 때 `MetadataOptions_HttpEndpoint`를 `enabled`로 설정해야 합니다.

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType c6i.large `
    -MetadataOptions_HttpEndpoint enabled `
    -MetadataOptions_HttpProtocolIpv6 enabled
```

인스턴스 메타데이터에 대한 액세스 비활성화

인스턴스를 시작할 때 IMDS를 비활성화하여 인스턴스 메타데이터에 대한 액세스를 끌 수 있습니다. 나중에 IMDS를 다시 활성화하여 액세스를 켤 수 있습니다. 자세한 내용은 인스턴스 메타데이터에 대한 액세스 활성화 섹션을 참조하세요.

> **Important**
>
> 시작 시 또는 시작한 후에 IMDS를 비활성할 수 있습니다. 시작 시 IMDS를 비활성화하면 다음 사항이 작동하지 않을 수 있습니다.
>
> - 인스턴스에 대한 SSH 액세스 권한이 없을 수 있습니다. 인스턴스의 퍼블릭 SSH 키인 `public-keys/0/openssh-key`는 일반적으로 키가 제공되고 EC2 인스턴스 메타데이터에서 액세스되기 때문에 액세스할 수 없습니다.
> - EC2 사용자 데이터는 사용할 수 없으며 인스턴스 시작 시 실행되지 않습니다. EC2 사용자 데이터는 IMDS에서 호스팅됩니다. IMDS를 비활성화하면 사용자 데이터에 대한 액세스가 사실상 꺼집니다.
>
> 이 기능을 사용하려면, 시작 후 IMDS를 다시 활성화할 수 있습니다.

Console

시작 시 인스턴스 메타데이터에 대한 액세스 끄기

- Advanced details(고급 세부 정보)에서 다음을 지정하여 Amazon EC2 콘솔에서 인스턴스를 시작합니다.
  - 액세스 가능한 메타데이터(Metadata accessible)에 비활성화(Disabled)를 선택합니다.

자세한 내용은 고급 세부 정보 섹션을 참조하세요.

AWS CLI

시작 시 인스턴스 메타데이터에 대한 액세스 끄기

`--metadata-options`를 `HttpEndpoint=disabled`로 설정하여 인스턴스를 시작합니다.

```
aws ec2 run-instances \
```

```
    --image-id ami-0abcdef1234567890 \
    --instance-type c6i.large \
    ...
    --metadata-options "HttpEndpoint=disabled"
```

PowerShell

시작 시 인스턴스 메타데이터에 대한 액세스 끄기

다음 New-EC2Instance Cmdlet 예에서는 MetadataOptions_HttpEndpoint를 disabled로 설정하여 인스턴스를 시작합니다.

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType c6i.large `
    -MetadataOptions_HttpEndpoint disabled
```

AWS CloudFormation

AWS CloudFormation을 사용하여 인스턴스에 대한 메타데이터 옵션을 지정하려면 AWS CloudFormation 사용 설명서의 AWS::EC2::LaunchTemplate MetadataOptions 속성을 참조하세요.

인스턴스 메타데이터의 태그에 대한 액세스 허용

기본적으로 인스턴스 태그는 인스턴스 메타데이터에서 액세스할 수 없습니다. 각 인스턴스에 대해 명시적으로 액세스를 허용해야 합니다. 액세스가 허용되면 인스턴스 태그 키는 특정 문자 제한을 준수해야 합니다. 그렇지 않으면 인스턴스 시작이 실패합니다. 자세한 내용은 인스턴스 메타데이터에서 태그에 대한 액세스 활성화 섹션을 참조하세요.

기존 인스턴스에 대한 인스턴스 메타데이터 옵션 수정

기존 인스턴스에 대한 인스턴스 메타데이터 옵션을 수정할 수 있습니다.

사용자가 기존 인스턴스의 인스턴스 메타데이터 옵션을 수정하지 못하도록 하는 IAM 정책을 생성할 수도 있습니다. 인스턴스 메타데이터 옵션을 수정할 수 있는 사용자를 제어하려면 지정된 역할을 가진 사용자 이외의 모든 사용자가 ModifyInstanceMetadataOptions API를 사용하지 못하게 하는 정책을 지정합니다. IAM 정책 예제는 인스턴스 메타데이터 작업 섹션을 참조하세요.

> **Note**
>
> 선언적 정책을 사용하여 인스턴스 메타데이터 옵션을 구성한 경우 계정 내에서 직접 수정할 수 없습니다. 자세한 내용은 AWS Organizations 사용 설명서의 선언적 정책을 참조하세요.

IMDSv2의 사용 요구

인스턴스 메타데이터를 요청할 때 IMDSv2를 사용하도록 하려면 다음 방법 중 하나를 사용하여 기존 인스턴스의 인스턴스 메타데이터 옵션을 수정합니다. IMDSv2가 필요한 경우 IMDSv1을 사용할 수 없습니다.

> **Note**
>
> IMDSv2를 사용하도록 요구하기 전에 인스턴스가 IMDSv1을 직접 호출하지 않는지 확인합니다. `MetadataNoToken` CloudWatch 지표는 IMDSv1 직접 호출을 추적합니다. `MetadataNoToken`에서 인스턴스의 IMDSv1 사용량을 0으로 기록하면 해당 인스턴스에서 IMDSv2를 요구할 준비가 된 것입니다.

Console

기존 인스턴스에서 IMDSv2를 사용해야 하도록 설정하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다.
4. 작업, 인스턴스 설정, 인스턴스 메타데이터 옵션 수정을 차례로 선택합니다.
5. 인스턴스 메타데이터 옵션 수정 대화 상자에서 다음을 수행합니다.
   a. 인스턴스 메타데이터 서비스에서 활성화를 선택합니다.
   b. IMDSv2의 경우 필수를 선택합니다.
   c. 저장을 선택합니다.

AWS CLI

기존 인스턴스에서 IMDSv2를 사용해야 하도록 설정하려면

modify-instance-metadata-options CLI 명령을 사용하고 `http-tokens` 파라미터를 `required`로 설정합니다. 또한 `http-tokens`의 값을 지정할 때 `http-endpoint`를 `enabled`로 설정해야 합니다.

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567890abcdef0 \
    --http-tokens required \
    --http-endpoint enabled
```

PowerShell

기존 인스턴스에서 IMDSv2를 사용해야 하도록 설정하려면

Edit-EC2InstanceMetadataOption Cmdlet을 사용하고 `HttpTokens` 파라미터를 `required`(으)로 설정합니다. 또한 `HttpTokens`의 값을 지정할 때 `HttpEndpoint`를 `enabled`로 설정해야 합니다.

```
(Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -HttpTokens required `
    -HttpEndpoint enabled).InstanceMetadataOptions
```

IMDSv1 사용 복원

IMDSv2가 필수인 경우 인스턴스 메타데이터를 요청할 때 IMDSv1이 작동하지 않습니다. IMDSv2가 선택 사항인 경우 IMDSv2와 IMDSv1이 모두 작동합니다. 따라서 IMDSv1을 복원하려면 다음 방법 중 하나를 사용하여 IMDSv2를 선택 사항으로 지정합니다.

Console

인스턴스에서 IMDSv1 사용을 복원하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다.
4. 작업, 인스턴스 설정, 인스턴스 메타데이터 옵션 수정을 차례로 선택합니다.
5. 인스턴스 메타데이터 옵션 수정 대화 상자에서 다음을 수행합니다.
   a. 인스턴스 메타데이터 서비스의 경우 활성화가 선택되어 있는지 확인합니다.
   b. IMDSv2의 경우 선택 사항을 선택합니다.

c. 저장을 선택합니다.

AWS CLI

인스턴스에서 IMDSv1 사용을 복원하려면

인스턴스 메타데이터를 요청할 때 `http-tokens`가 `optional`로 설정된 modify-instance-metadata-options CLI를 사용하여 IMDSv1의 사용을 복원할 수 있습니다.

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567890abcdef0 \
    --http-tokens optional \
    --http-endpoint enabled
```

PowerShell

인스턴스에서 IMDSv1 사용을 복원하려면

Edit-EC2InstanceMetadataOption Cmdlet을 사용하되 `HttpTokens`을(를) `optional`(으)로 설정하여 인스턴스 메타데이터를 요청할 때 IMDSv1 사용을 복원할 수 있습니다.

```
(Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -HttpTokens optional `
    -HttpEndpoint enabled).InstanceMetadataOptions
```

PUT 응답 홉 제한 변경

기존 인스턴스의 경우 PUT 응답 홉 제한에 대한 설정을 변경할 수 있습니다.

현재는 AWS CLI 및 AWS SDK만 PUT 응답 홉 제한 변경을 지원합니다.

AWS CLI

PUT 응답 홉 제한을 변경하려면

modify-instance-metadata-options CLI 명령을 사용하고 `http-put-response-hop-limit` 파라미터를 필요한 홉 수로 설정합니다. 다음 예제에서는 옵 제한이 3으로 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-1234567890abcdef0 \
    --http-put-response-hop-limit 3 \
    --http-endpoint enabled
```

PowerShell

PUT 응답 홉 제한을 변경하려면

Edit-EC2InstanceMetadataOption Cmdlet을 사용하고 HttpPutResponseHopLimit 매개 변수를 필요한 홉 수로 설정합니다. 다음 예제에서는 옵 제한이 3으로 설정됩니다. 또한 HttpPutResponseHopLimit의 값을 지정할 때 HttpEndpoint를 enabled로 설정해야 합니다.

```
(Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -HttpPutResponseHopLimit 3 `
    -HttpEndpoint enabled).InstanceMetadataOptions
```

IMDS IPv4 및 IPv6 엔드포인트 활성화

IMDS에는 인스턴스에 IPv4(169.254.169.254) 및 IPv6([fd00:ec2::254])이라는 두 개의 엔드포인트가 있습니다. IMDS를 활성화하면 IPv4 엔드포인트가 자동으로 활성화됩니다. IPv6 전용 서브넷으로 인스턴스를 시작하는 경우에도 IPv6 엔드포인트는 비활성화된 상태로 유지됩니다. IPv6 엔드포인트를 활성화하려면 명시적으로 활성화해야 합니다. IPv6 엔드포인트가 활성화되면 IPv4 엔드포인트는 활성화된 상태로 유지됩니다.

인스턴스 시작 시 또는 이후에 IPv6 엔드포인트를 활성화할 수 있습니다.

IPv6 엔드포인트 활성화 요구

- 선택한 인스턴스 유형은 Nitro 기반 인스턴스입니다.
- 선택한 서브넷은 IPv6를 지원합니다. 여기서 서브넷은 이중 스택 또는 IPv6 전용입니다.

현재는 AWS CLI 및 AWS SDK만 인스턴스 후 IPv6 엔드포인트 활성화를 지원합니다.

AWS CLI

인스턴스에 대해 IMDS IPv6 엔드포인트를 활성화하는 방법

modify-instance-metadata-options CLI 명령을 사용하고 `http-protocol-ipv6` 파라미터를 `enabled`로 설정합니다. 또한 `http-protocol-ipv6`의 값을 지정할 때 `http-endpoint`를 `enabled`로 설정해야 합니다.

```
aws ec2 modify-instance-metadata-options \
 --instance-id i-1234567890abcdef0 \
 --http-protocol-ipv6 enabled \
 --http-endpoint enabled
```

PowerShell

인스턴스에 대해 IMDS IPv6 엔드포인트를 활성화하는 방법

Edit-EC2InstanceMetadataOption Cmdlet을 사용하고 `HttpProtocolIpv6` 파라미터를 `enabled`(으)로 설정합니다. 또한 `HttpProtocolIpv6`의 값을 지정할 때 `HttpEndpoint`를 `enabled`로 설정해야 합니다.

```
(Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -HttpProtocolIpv6 enabled `
    -HttpEndpoint enabled).InstanceMetadataOptions
```

인스턴스 메타데이터에 대한 액세스 활성화

사용 중인 IMDS 버전에 관계없이 인스턴스에서 IMDS의 HTTP 엔드포인트를 활성화하여 인스턴스 메타데이터에 대한 액세스를 사용 설정할 수 있습니다. HTTP 엔드포인트를 비활성화하여 언제든지 이 변경을 되돌릴 수 있습니다.

인스턴스에서 인스턴스 메타데이터에 대한 액세스를 사용 설정하려면 다음 방법 중 하나를 사용합니다.

Console

인스턴스 메타데이터에 대한 액세스를 활성화하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다.
4. 작업, 인스턴스 설정, 인스턴스 메타데이터 옵션 수정을 차례로 선택합니다.

5. 인스턴스 메타데이터 옵션 수정 대화 상자에서 다음을 수행합니다.

   a. 인스턴스 메타데이터 서비스에서 활성화를 선택합니다.

   b. 저장을 선택합니다.

AWS CLI

인스턴스 메타데이터에 대한 액세스를 활성화하려면

modify-instance-metadata-options CLI 명령을 사용하고 `http-endpoint` 파라미터를 `enabled`로 설정합니다.

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567890abcdef0 \
    --http-endpoint enabled
```

PowerShell

인스턴스 메타데이터에 대한 액세스를 활성화하려면

Edit-EC2InstanceMetadataOption Cmdlet을 사용하고 `HttpEndpoint` 파라미터를 `enabled`(으)로 설정합니다.

```
(Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -HttpEndpoint enabled).InstanceMetadataOptions
```

인스턴스 메타데이터에 대한 액세스 비활성화

사용 중인 IMDS 버전에 관계없이 인스턴스에서 IMDS의 HTTP 엔드포인트를 비활성화하여 인스턴스 메타데이터에 대한 액세스를 비활성화할 수 있습니다. HTTP 엔드포인트를 활성화하여 언제든지 이 변경을 되돌릴 수 있습니다.

인스턴스에서 인스턴스 메타데이터에 대한 액세스를 비활성화하려면 다음 방법 중 하나를 사용합니다.

Console

인스턴스 메타데이터에 대한 액세스를 끄려면

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

2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다.
4. 작업, 인스턴스 설정, 인스턴스 메타데이터 옵션 수정을 차례로 선택합니다.
5. 인스턴스 메타데이터 옵션 수정 대화 상자에서 다음을 수행합니다.
   a. 인스턴스 메타데이터 서비스의 경우 활성화를 선택 취소합니다.
   b. 저장을 선택합니다.

AWS CLI

인스턴스 메타데이터에 대한 액세스를 끄려면

modify-instance-metadata-options CLI 명령을 사용하고 `http-endpoint` 파라미터를 `disabled`로 설정합니다.

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567890abcdef0 \
    --http-endpoint disabled
```

PowerShell

인스턴스 메타데이터에 대한 액세스를 끄려면

Edit-EC2InstanceMetadataOption Cmdlet을 사용하고 `HttpEndpoint` 파라미터를 `disabled`(으)로 설정합니다.

```
(Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -HttpEndpoint disabled).InstanceMetadataOptions
```

인스턴스 메타데이터의 태그에 대한 액세스 허용

실행 중이거나 중지된 인스턴스에서 인스턴스 메타데이터의 태그에 대한 액세스를 허용할 수 있습니다. 각 인스턴스에 대해 명시적으로 액세스를 허용해야 합니다. 액세스가 허용되면 인스턴스 태그 키는 특정 문자 제한을 준수해야 합니다. 그렇지 않으면 오류가 발생합니다. 자세한 내용은 인스턴스 메타데이터에서 태그에 대한 액세스 활성화 섹션을 참조하세요.

사용자 데이터 입력을 사용하여 EC2 인스턴스를 시작할 때 명령 실행

Amazon EC2 인스턴스를 시작할 때 사용자 데이터를 인스턴스에 전달하여 자동화된 구성 태스크를 수행하는 데 사용할 수 있고, 인스턴스가 시작된 후에 스크립트를 실행하는 데 사용할 수도 있습니다.

더욱 복잡한 자동화 시나리오를 원하는 경우 AWS CloudFormation을 고려할 수 있습니다. 자세한 정보는 AWS CloudFormation 사용 설명서의 AWS CloudFormation을 사용하여 Amazon EC2에 애플리케이션 배포를 참조하세요.

Linux 인스턴스에서는 Amazon EC2에 쉘 스크립트 및 cloud-init 명령이라는 두 가지 유형의 사용자 데이터를 전달할 수 있습니다. 이 데이터를 인스턴스 시작 마법사에 일반 텍스트, 파일(명령줄 도구를 사용하여 인스턴스를 시작하는 데 유용) 또는 base64 인코딩 텍스트(API 호출용)로 전달할 수도 있습니다.

Windows 인스턴스에서는 시작 에이전트가 사용자 데이터 스크립트를 처리합니다.

고려 사항

- 사용자 데이터는 불투명 데이터로 취급됨: 제공한 것만을 살펴볼 수 있습니다. 해석은 인스턴스에 따라 다릅니다.
- 사용자 데이터는 base64로 인코딩해야 합니다. Amazon EC2 콘솔은 base64 인코딩을 수행하거나 base64 인코딩된 입력을 수락할 수 있습니다. 인스턴스 메타데이터 또는 콘솔을 사용하여 사용자 데이터를 가져오는 경우 데이터는 자동으로 base64 디코딩됩니다.
- 사용자 데이터는 base64로 인코딩되기 전에 원시 16KB 형식으로 제한됩니다. base64 인코딩 이후 n 길이의 문자열 크기는 ceil(n/3)*4입니다.
- 사용자 데이터는 인스턴스 속성입니다. 인스턴스에서 AMI를 생성하는 경우 인스턴스 사용자 데이터는 AMI에 포함되지 않습니다.

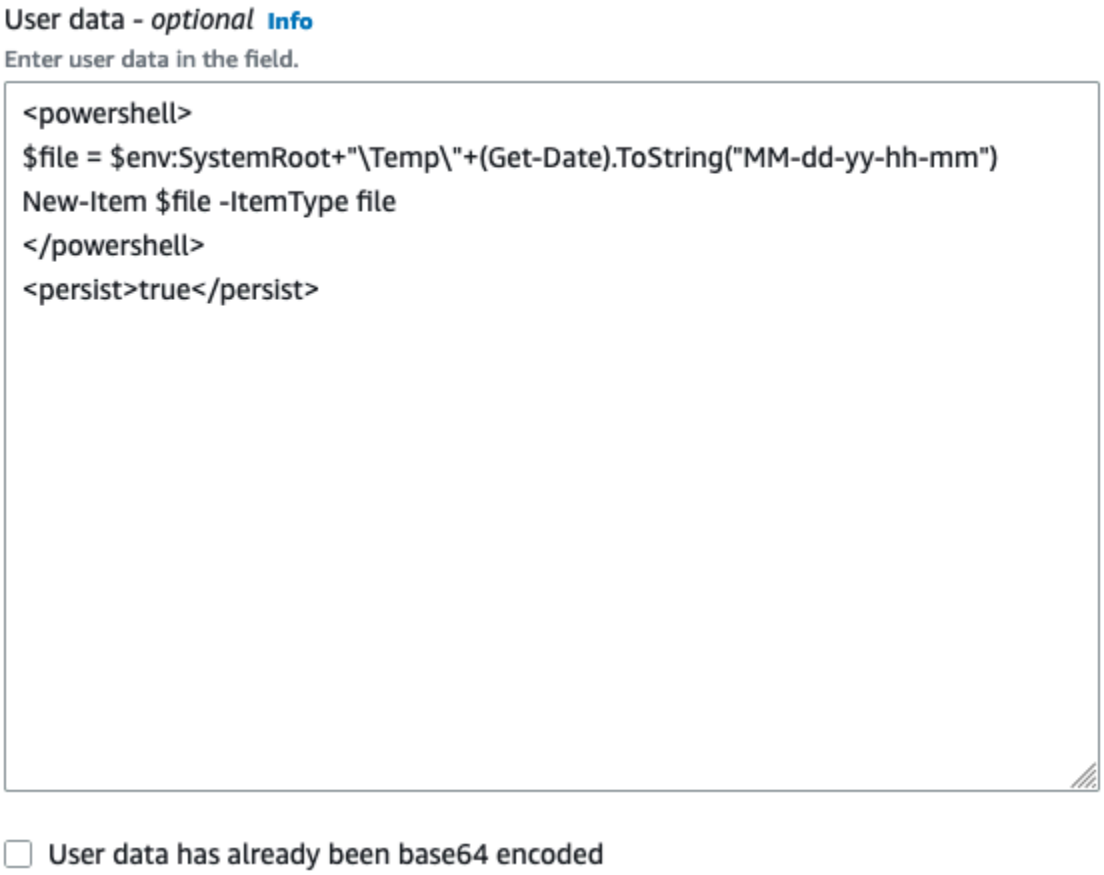
AWS Management Console의 사용자 데이터

인스턴스를 시작할 때 인스턴스 사용자 데이터를 지정할 수 있습니다. 인스턴스의 루트 볼륨이 EBS 볼륨이면 인스턴스를 중지하고 사용자 데이터를 업데이트할 수도 있습니다.

Launch Wizard를 사용하여 시작 시 인스턴스 사용자 데이터 지정

EC2 콘솔에서 Launch Wizard를 사용하여 인스턴스를 시작할 때 사용자 데이터를 지정할 수 있습니다. 시작 시 사용자 데이터를 지정하려면 인스턴스 시작 절차를 따르세요. 사용자 데이터(User data) 필드는 인스턴스 시작 마법사의 고급 세부 정보 섹션에 있습니다. 사용자 데이터 필드에 PowerShell 스크립트를 입력한 다음 인스턴스 시작 절차를 완료합니다.

다음 사용자 데이터 필드 스크린샷의 예제 스크립트는 파일 이름에 현재 날짜 및 시간을 사용하여 Windows 임시 폴더에 파일을 생성합니다. `<persist>true</persist>`를 포함할 경우 인스턴스를 재부팅하거나 시작할 때마다 스크립트가 실행됩니다. 사용자 데이터가 이미 base64로 인코딩됨 확인란을 비워 두면 Amazon EC2 콘솔이 base64 인코딩을 수행합니다.



자세한 내용은 Launch Wizard를 사용하여 시작 시 인스턴스 사용자 데이터 지정 섹션을 참조하세요. AWS CLI를 사용하는 Linux의 예는 the section called “사용자 데이터 및 AWS CLI” 섹션을 참조하세요. Windows PowerShell용 도구를 사용하는 Windows의 예는 the section called “사용자 데이터 및 Tools for Windows PowerShell” 섹션을 참조하세요.

인스턴스 사용자 데이터 보기 및 업데이트

모든 인스턴스의 인스턴스 사용자 데이터를 볼 수 있으며, 중지된 인스턴스의 인스턴스 사용자 데이터를 업데이트할 수 있습니다.

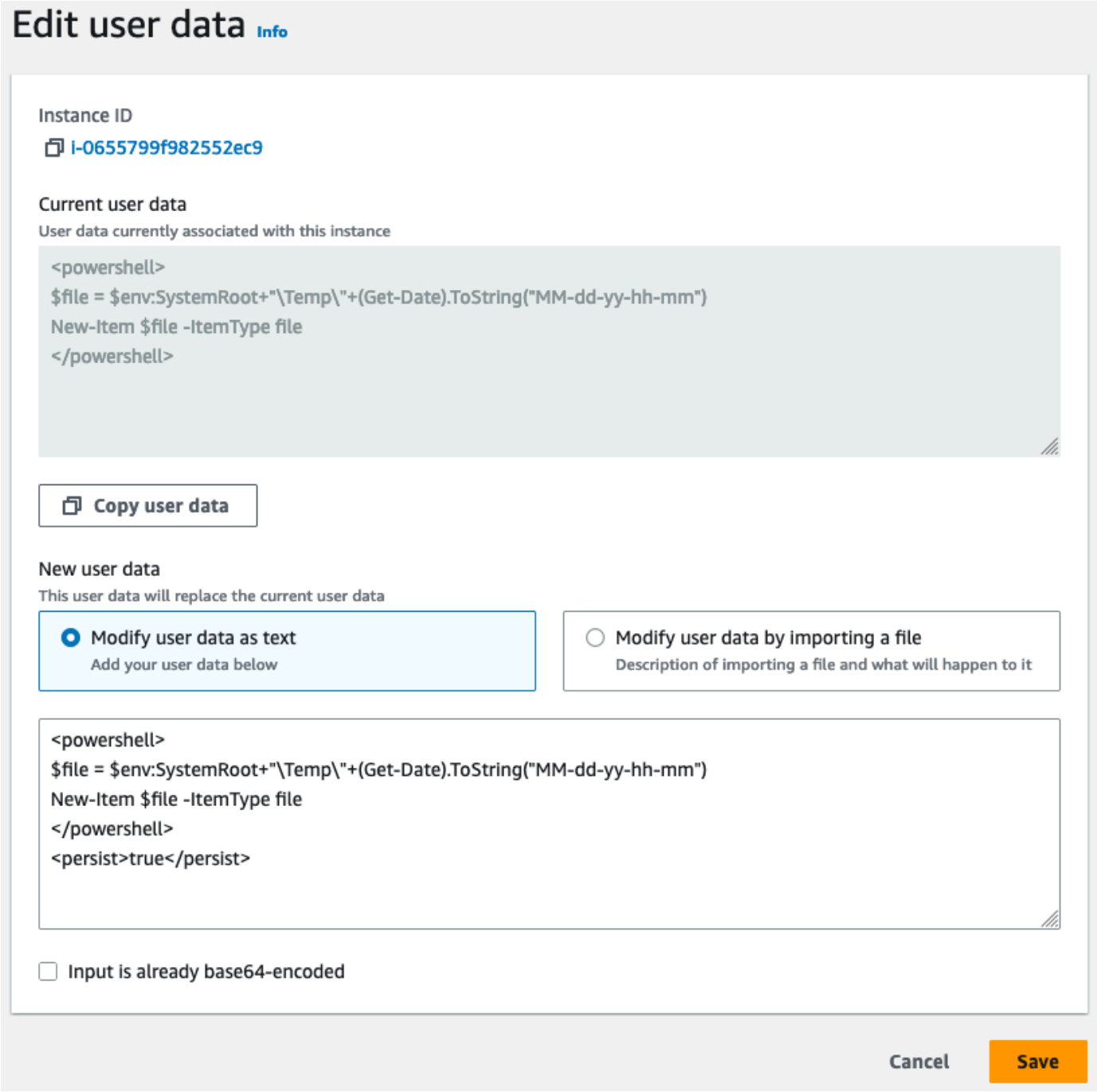
콘솔을 사용하여 인스턴스의 사용자 데이터를 업데이트하려면

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

3. 인스턴스를 선택하고 작업, 인스턴스 상태, 인스턴스 중지를 차례로 선택합니다.

   > **Warning**
   >
   > 인스턴스를 중지하면 인스턴스 스토어 볼륨의 데이터가 삭제됩니다. 인스턴스 스토어 볼륨의 데이터를 유지하려면 영구 스토리지에 백업하세요.

4. 확인 메시지가 표시되면 [Stop]을 선택합니다. 인스턴스가 중지하는 데 몇 분 정도 걸릴 수 있습니다.
5. 인스턴스가 선택된 상태에서 작업(Actions), 인스턴스 설정(Instance settings), 사용자 데이터 편집(Edit user data)을 차례로 선택합니다. 인스턴스가 실행 중일 때는 사용자 데이터를 변경할 수 없습니다. 하지만 볼 수는 있습니다.
6. [사용자 데이터 편집(Edit user data)] 대화 상자에서 사용자 데이터를 업데이트하고 [저장(Save)]을 선택합니다. 매번 인스턴스를 재부팅하거나 시작할 때마다 사용자 데이터 스크립트를 실행하려면 다음 예제와 같이 `<persist>true</persist>`를 추가합니다.



7. 인스턴스를 시작합니다. 이후의 재부팅 또는 시작에 대해 사용자 데이터 실행을 활성화한 경우 인스턴스 시작 프로세스의 일부로서 업데이트된 사용자 데이터 스크립트가 실행됩니다.

Amazon EC2가 Linux 인스턴스의 사용자 데이터를 처리하는 방법

다음 예제에서는 Amazon Linux 2에 LAMP 서버 설치의 명령을 인스턴스 시작 시 실행되는 shell 스크립트 및 cloud-init 명령 세트로 변환합니다. 각 예제에서는 사용자 데이터에 따라 다음 태스크를 수행합니다.

- 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
- 사용자 데이터 및 shell 스크립트
- 인스턴스 사용자 데이터 업데이트
- 사용자 데이터 및 cloud-init 명령
- 사용자 데이터 및 AWS CLI
- 셸 스크립트와 cloud-init 지시문 결합

사전 조건

이 주제의 예에서는 다음과 같이 가정합니다.

- 사용자 인스턴스에 인터넷에서 접속 가능한 퍼블릭 DNS 이름이 있습니다.
- 인스턴스와 연결된 보안 그룹은 SSH(포트 22) 트래픽을 허용하도록 구성되어 있으므로 인스턴스에 연결하여 출력 로그 파일을 볼 수 있습니다.
- 인스턴스는 Amazon Linux 2 AMI를 사용하여 시작됩니다. 이러한 명령은 Amazon Linux 2에서만 사용해야 합니다. 다른 Linux 배포에서는 명령이 작동하지 않을 수 있습니다. 다른 배포에 대한 cloud-init 지원 등의 자세한 내용은 해당 설명서를 참조하세요.

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 지식 센터에서 사용자 데이터를 활용하여 Amazon EC2 Linux 인스턴스를 다시 시작할 때마다 스크립트를 자동으로 실행하려면 어떻게 해야 하나요?를 참조하세요.

사용자 데이터 shell 스크립트는 #! 문자 및 스크립트를 읽을 인터프리터의 경로(일반적으로 /bin/bash))로 시작되어야 합니다. 쉘 스크립팅에 대한 소개는 GNU 운영 체제 웹 사이트의 Bash 참조 매뉴얼을 참조하세요.

사용자 데이터로 입력된 스크립트는 루트 사용자 권한으로 실행되므로 스크립트에 sudo 명령을 사용하지 마세요. 생성하는 모든 파일의 소유권은 루트 사용자에게 있습니다. 루트 이외의 사용자에게 파일 액세스를 허용하려면 스크립트에서 권한을 적절히 수정해야 합니다. 또한 스크립트는 대화형으로 실행되지 않으므로 사용자의 입력이 필요한 명령(예: yum update 플래그 없는 -y)은 포함할 수 없습니다.

사용자 데이터 스크립트에서 AWS CLI를 포함한 AWS API를 사용하는 경우 인스턴스를 시작할 때 인스턴스 프로파일을 사용해야 합니다. 인스턴스 프로필은 사용자 데이터 스크립트에서 API 호출을 실행하는 데 필요한 적절한 AWS 자격 증명을 제공합니다. 자세한 내용은 IAM 사용 설명서의 인스턴스 프로파일 사용을 참조하세요. IAM 역할에 할당하는 권한은 API를 사용하여 호출하는 서비스에 따라 다릅니다. 자세한 내용은 Amazon EC2의 IAM 역할 섹션을 참조하세요.

cloud-init 출력 로그 파일이 콘솔 출력을 캡처하므로 시작 후 인스턴스가 의도한 대로 동작하지 않더라도 스크립트를 손쉽게 디버깅할 수 있습니다. 로그 파일을 보려면 인스턴스에 연결하여 `/var/log/cloud-init-output.log`를 엽니다.

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 필요하다

고 언급했습니다. 그렇지 않은 경우, 스크립트는 AMI에서 시작된 어떤 인스턴스에서든 이 디렉터리에 존재합니다.

인스턴스 사용자 데이터 업데이트

인스턴스 사용자 데이터를 업데이트하려면 먼저 인스턴스를 중단해야 합니다. 인스턴스가 실행 중인 경우 사용자 데이터를 볼 수 있지만 수정할 수는 없습니다.

> **Warning**
> 인스턴스를 중지하면 인스턴스 스토어 볼륨의 데이터가 삭제됩니다. 인스턴스 스토어 볼륨의 데이터를 유지하려면 영구 스토리지에 백업하세요.

인스턴스 사용자 데이터를 수정하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 인스턴스 상태, 인스턴스 중지를 차례로 선택합니다. 이 옵션이 비활성화되어 있으면 해당 인스턴스가 이미 중지되었거나 해당 루트 디바이스가 인스턴스 스토어 볼륨인 것입니다.
4. 확인 메시지가 표시되면 [Stop]을 선택합니다. 인스턴스가 중지하는 데 몇 분 정도 걸릴 수 있습니다.
5. 인스턴스가 선택된 상태에서 작업(Actions), 인스턴스 설정(Instance settings), 사용자 데이터 편집(Edit user data)을 차례로 선택합니다.
6. 필요에 따라 사용자 데이터를 수정한 다음 저장(Save)을 선택합니다.
7. 인스턴스를 시작합니다. 새 사용자 데이터는 인스턴스를 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. cloud-init 패키지가 Amazon Linux 인스턴스에 대해 수행하는 구성 태스크에 대한 자세한 내용은 Amazon Linux 2 사용 설명서의 Amazon Linux 2에서 cloud-init 사용하기를 참조하세요.

cloud-init 사용자 명령을 인스턴스 시작 시에 전달하는 방법은 스크립트를 전달하는 방법과 동일하지만 구문은 서로 다릅니다. cloud-init에 대한 자세한 내용은 https://cloudinit.readthedocs.org/en/latest/index.html을 참조하세요.

> **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 지식 센터에서 사용자 데이터를 활용하여 Amazon EC2 Linux 인스턴스를 다시 시작할 때마다 스크립트를 자동으로 실행하려면 어떻게 해야 하나요?를 참조하세요.

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. 인스턴스 시작 절차를 따릅니다. 사용자 데이터(User data) 필드는 인스턴스 시작 마법사의 고급 세부 정보 섹션에 있습니다. 사용자 데이터(User data) 필드에 cloud-init 지시문 텍스트를 입력한 다음 인스턴스 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 태스크를 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 확인하세요. 자세한 내용은 보안 그룹 규칙 구성 섹션을 참조하세요.

3. (선택 사항) 명령에서 의도한 작업을 완료하지 못했거나 명령이 오류 없이 완료되었는지 여부를 확인하려는 경우 인스턴스에 연결하여 출력 로그 파일(/var/log/cloud-init-output.log)을 조사하고 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.log로 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 EC2 인스턴스에 대한 인스턴스 메타데이터에 액세스 섹션을 참조하세요.

Windows에서 AWS Tools for Windows PowerShell를 사용하는 대신 AWS CLI을 사용할 수 있습니다. 자세한 정보는 사용자 데이터 및 Tools for Windows PowerShell을 참조하세요.

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: 중지된 인스턴스의 사용자 데이터 삭제

기존 사용자 데이터를 삭제하려면 다음과 같이 modify-instance-attribute 명령을 사용합니다.

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --user-data Value=
```

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
  ```

출력의 예시는 다음과 같습니다.

```
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

셸 스크립트와 cloud-init 지시문 결합

기본적으로 사용자 데이터에는 한 번에 하나의 콘텐츠 유형만 포함할 수 있습니다. 그러나 MIME 멀티 파트 파일에서 `text/cloud-config` 및 `text/x-shellscript` 콘텐츠 유형을 사용하여 사용자 데이터에 셸 스크립트와 cloud-init 지시문을 모두 포함할 수 있습니다.

다음은 MIME 멀티 파트 형식을 보여줍니다.

```
Content-Type: multipart/mixed; boundary="//"
MIME-Version: 1.0

--//
Content-Type: text/cloud-config; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename="cloud-config.txt"

#cloud-config
cloud-init directives

--//
Content-Type: text/x-shellscript; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename="userdata.txt"

#!/bin/bash
shell script commands
--//--
```

예를 들어, 다음 사용자 데이터에는 cloud-init 지시문과 bash 셸 스크립트가 포함됩니다. cloud-init 지시문은 파일(`/test-cloudinit/cloud-init.txt`)을 생성하고 해당 파일에 `Created by cloud-init`를 씁니다. bash 셸 스크립트는 파일(`/test-userscript/userscript.txt`)을 생성하고 해당 파일에 `Created by bash shell script`를 씁니다.

```
Content-Type: multipart/mixed; boundary="//"
MIME-Version: 1.0

--//
Content-Type: text/cloud-config; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename="cloud-config.txt"

#cloud-config
runcmd:
- [ mkdir, /test-cloudinit ]
write_files:
- path: /test-cloudinit/cloud-init.txt
content: Created by cloud-init

--//
Content-Type: text/x-shellscript; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename="userdata.txt"

#!/bin/bash
mkdir test-userscript
touch /test-userscript/userscript.txt
echo "Created by bash shell script" >> /test-userscript/userscript.txt
--//--
```

Amazon EC2가 Windows 인스턴스의 사용자 데이터를 처리하는 방법

Windows 인스턴스에서는 시작 에이전트가 사용자 데이터 관련 작업을 수행합니다. 자세한 내용은 다음 자료를 참조하세요.

- EC2Launch v2
- EC2Launch
- EC2Config 서비스

`UserData` 템플릿의 AWS CloudFormation 속성 어셈블리 예제는 Base64 인코딩 UserData 속성 및 AccessKey 및 SecretKey를 사용하는 Base64 인코딩 UserData 속성을 참조하세요.

수명 주기 후크와 함께 작동하는 Auto Scaling 그룹 내의 인스턴스에서 명령을 실행하는 예제는 Amazon EC2 Auto Scaling 사용 설명서에서 자습서: 인스턴스 메타데이터를 통해 대상 수명 주기 상태를 검색하도록 사용자 데이터 구성을 참조하세요.

내용

- 사용자 데이터 스크립트
- 압축된 사용자 데이터
- 사용자 데이터 실행
- 사용자 데이터 및 Tools for Windows PowerShell

사용자 데이터 스크립트

EC2Config 또는 EC2Launch에서 스크립트를 실행하려면 사용자 데이터에 추가할 때 해당 스크립트를 특수 태그 안에 포함해야 합니다. 사용하는 태그는 명령 프롬프트 창에서 명령이 실행되는지(배치 명령) 아니면 Windows PowerShell을 사용하는지에 따라 달라집니다.

배치 스크립트와 Windows PowerShell 스크립트를 둘 다 지정할 경우, 인스턴스 사용자 데이터에 나타나는 순서와 관계 없이, 배치 스크립트가 먼저 실행되고 Windows PowerShell 스크립트가 다음에 실행됩니다.

사용자 데이터 스크립트에서 AWS을(를) 포함한 AWS CLI API를 사용하는 경우 인스턴스를 시작할 때 인스턴스 프로파일을 사용해야 합니다. 인스턴스 프로필은 사용자 데이터 스크립트에서 API 호출을 실행하는 데 필요한 적절한 AWS 자격 증명을 제공합니다. 자세한 내용은 인스턴스 프로파일 섹션을 참조하세요. IAM 역할에 할당하는 권한은 API를 사용하여 호출하는 서비스에 따라 다릅니다. 자세한 내용은 Amazon EC2의 IAM 역할 섹션을 참조하세요.

스크립트 유형

- 배치 스크립트 구문
- Windows PowerShell 스크립트 구문
- YAML 구성 스크립트 구문
- Base64 인코딩

배치 스크립트 구문

script 태그를 사용하여 배치 스크립트를 지정합니다. 다음 예제와 같이 줄 바꿈을 사용하여 명령을 구분합니다.

```
<script>
    echo Current date and time >> %SystemRoot%\Temp\test.log
    echo %DATE% %TIME% >> %SystemRoot%\Temp\test.log
</script>
```

기본적으로 사용자 데이터 스크립트는 인스턴스를 시작할 때 한 번만 실행됩니다. 인스턴스를 재부팅하거나 시작할 때마다 사용자 데이터 스크립트를 실행하려면 <persist>true</persist>를 사용자 데이터에 추가합니다.

```
<script>
    echo Current date and time >> %SystemRoot%\Temp\test.log
    echo %DATE% %TIME% >> %SystemRoot%\Temp\test.log
</script>
<persist>true</persist>
```

EC2Launch v2 에이전트

UserData 단계에서 EC2Launch v2 executeScript 태스크와 함께 XML 사용자 데이터 스크립트를 분리된 프로세스로 실행하려면 사용자 데이터에 <detach>true</detach>를 추가하세요.

> **Note**
>
> detach 태그는 이전 시작 에이전트에서 지원되지 않습니다.

```
<script>
    echo Current date and time >> %SystemRoot%\Temp\test.log
    echo %DATE% %TIME% >> %SystemRoot%\Temp\test.log
</script>
<detach>true</detach>
```

Windows PowerShell 스크립트 구문

AWS Windows AMI에는 AWS Tools for Windows PowerShell이(가) 포함되어 있으므로 사용자 데이터에서 이러한 cmdlet을 지정할 수 있습니다. IAM 역할을 인스턴스와 연결하는 경우, 인스턴스에서 실행되는 애플리케이션이 역할의 자격 증명을 사용하여 AWS 리소스(예: Amazon S3 버킷 등)에 액세스할 수 있기 때문에 cmdlet에 대한 자격 증명을 지정할 필요가 없습니다.

<powershell> 태그를 사용하여 Windows PowerShell 스크립트를 지정합니다. 줄 바꿈을 사용하여 명령을 구분합니다. <powershell> 태그는 대/소문자를 구분합니다.

예시:

```
<powershell>
    $file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
    New-Item $file -ItemType file
</powershell>
```

기본적으로 사용자 데이터 스크립트는 인스턴스를 시작할 때 한 번만 실행됩니다. 인스턴스를 재부팅하거나 시작할 때마다 사용자 데이터 스크립트를 실행하려면 `<persist>true</persist>`를 사용자 데이터에 추가합니다.

```
<powershell>
    $file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
    New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

`<powershellArguments>` 태그를 사용하여 하나 이상의 PowerShell 인수를 지정할 수 있습니다. 인수가 전달되지 않는 경우 EC2Launch 및 EC2Launch v2는 기본적으로 `-ExecutionPolicy Unrestricted` 인수를 추가합니다.

예시:

```
<powershell>
    $file = $env:SystemRoot + "\Temp" + (Get-Date).ToString("MM-dd-yy-hh-mm")
    New-Item $file -ItemType file
</powershell>
<powershellArguments>-ExecutionPolicy Unrestricted -NoProfile -NonInteractive</
powershellArguments>
```

EC2Launch v2 에이전트

`UserData` 단계에서 EC2Launch v2 executeScript 태스크와 함께 XML 사용자 데이터 스크립트를 분리된 프로세스로 실행하려면 사용자 데이터에 `<detach>true</detach>`를 추가하세요.

> **Note**
>
> detach 태그는 이전 시작 에이전트에서 지원되지 않습니다.

```
<powershell>
    $file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
    New-Item $file -ItemType file
</powershell>
<detach>true</detach>
```

YAML 구성 스크립트 구문

EC2Launch v2를 사용하여 스크립트를 실행하는 경우 YAML 형식을 사용할 수 있습니다. EC2Launch v2에 대한 구성 작업, 세부 정보 및 예제를 보려면 EC2Launch v2 태스크 구성 섹션을 참조하세요.

`executeScript` 작업을 통해 YAML 스크립트를 지정합니다.

PowerShell 스크립트를 실행하는 YAML 구문 예제

```
version: 1.0
    tasks:
    - task: executeScript
      inputs:
      - frequency: always
        type: powershell
        runAs: localSystem
        content: |-
          $file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
          New-Item $file -ItemType file
```

배치 스크립트를 실행하는 YAML 구문 예제

```
version: 1.1
    tasks:
    - task: executeScript
      inputs:
      - frequency: always
        type: batch
        runAs: localSystem
        content: |-
          echo Current date and time >> %SystemRoot%\Temp\test.log
          echo %DATE% %TIME% >> %SystemRoot%\Temp\test.log
```

Base64 인코딩

Amazon EC2 API 또는 사용자 데이터의 base64 인코딩을 수행하지 않는 도구를 사용하는 경우, 직접 사용자 데이터를 인코딩해야 합니다. 그렇지 않을 경우, 실행할 `script` 또는 `powershell` 태그를 찾을 수 없다는 오류가 기록됩니다. 다음은 Windows PowerShell을 사용하여 인코딩하는 예제입니다.

```
$UserData =
 [System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($Script))
```

다음은 PowerShell을 사용하여 디코딩하는 예제입니다.

```
$Script =
 [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($UserData))
```

base64 인코딩에 대한 자세한 내용은 https://www.ietf.org/rfc/rfc4648.txt을 참조하세요.

압축된 사용자 데이터

EC2Launch v2는 IMDS에서 부과한 16KB 제한보다 큰 사용자 데이터를 제출하는 방법으로 압축된 사용자 데이터를 지원합니다. 이 기능을 사용하려면 사용자 데이터 스크립트를 `.zip` 아카이브로 압축하여 EC2 인스턴스에 전달합니다. EC2Launch v2는 압축된 사용자 데이터를 감지하면 압축된 사용자 데이터 스크립트의 압축을 자동으로 풀고 실행합니다.

표준 사용자 데이터와 마찬가지로 Amazon EC2 API 또는 사용자 데이터의 base64 인코딩을 수행하지 않는 도구를 사용하는 경우, 직접 사용자 데이터를 인코딩해야 합니다. 사용자 데이터 크기 제한 및 base64 인코딩에 대한 자세한 내용은 EC2 인스턴스에 대한 인스턴스 메타데이터에 액세스 섹션을 참조하세요.

사용자 데이터 실행

기본적으로 모든 AWS Windows AMI는 초기 실행에 대해 활성화되는 사용자 데이터 실행 기능을 포함하고 있습니다. 다음 번에 인스턴스가 재부팅하거나 재시작할 때 사용자 데이터 스크립트가 실행되도록 지정할 수 있습니다. 또는 인스턴스가 재부팅하거나 재시작할 때마다 사용자 데이터 스크립트가 실행되도록 지정할 수도 있습니다.

> **Note**
>
> 초기 실행 후에는 사용자 데이터가 기본적으로 실행되도록 설정되어 있지 않습니다. 인스턴스를 재부팅하거나 시작할 때 사용자 데이터가 실행되도록 설정하려면 이후 재부팅 또는 시작 시 스크립트 실행 섹션을 참조하세요.

사용자 데이터 스크립트는 임의 암호가 생성될 때 로컬 관리자 계정에서 실행됩니다. 그렇게 하지 않으면 사용자 데이터 스크립트가 시스템 계정에서 실행됩니다.

인스턴스 시작 스크립트

인스턴스 사용자 데이터의 스크립트는 인스턴스 초기 시작 중에 실행됩니다. `persist` 태그가 있는 경우 사용자 데이터 실행이 후속 재부팅이나 시작에 대해 활성화됩니다. EC2Launch v2, EC2Launch 및 EC2Config용 로그 파일에는 표준 출력 및 표준 오류 스트림의 출력이 들어 있습니다.

EC2Launch v2

EC2Launch v2의 로그 파일은 `C:\ProgramData\Amazon\EC2Launch\log\agent.log`입니다.

> **Note**
>
> `C:\ProgramData` 폴더를 숨길 수 있습니다. 폴더를 보려면 숨겨진 파일과 폴더를 표시해야 합니다.

다음 정보는 사용자 데이터가 실행될 때 로깅됩니다.

- `Info: Converting user-data to yaml format` - 사용자 데이터가 XML 형식으로 제공된 경우
- `Info: Initialize user-data state` - 사용자 데이터 실행의 시작
- `Info: Frequency is: always` - 부팅할 때마다 사용자 데이터 태스크가 실행되는 경우
- `Info: Frequency is: once` - 사용자 데이터 태스크가 한 번만 실행되는 경우
- `Stage: postReadyUserData execution completed` - 사용자 데이터 실행의 끝

EC2Launch

EC2Launch에 대한 로그 파일은 `C:\ProgramData\Amazon\EC2-Windows\Launch\Log\UserdataExecution.log`입니다.

`C:\ProgramData` 폴더를 숨길 수 있습니다. 폴더를 보려면 숨겨진 파일과 폴더를 표시해야 합니다.

다음 정보는 사용자 데이터가 실행될 때 로깅됩니다.

- `Userdata execution begins` - 사용자 데이터 실행의 시작
- `<persist> tag was provided: true` - persist 태그가 있는 경우

- `Running userdata on every boot` - persist 태그가 있는 경우
- `<powershell> tag was provided.. running powershell content` - powershell 태그가 있는 경우
- `<script> tag was provided.. running script content` - script 태그가 있는 경우
- `Message: The output from user scripts` - 사용자 데이터 스크립트가 실행되는 경우 해당 출력이 로깅됨

EC2Config

EC2Config의 로그 파일은 `C:\Program Files\Amazon\Ec2ConfigService\Logs\Ec2Config.log`입니다. 다음 정보는 사용자 데이터가 실행될 때 로깅됩니다.

- `Ec2HandleUserData: Message: Start running user scripts` - 사용자 데이터 실행의 시작
- `Ec2HandleUserData: Message: Re-enabled userdata execution` - persist 태그가 있는 경우
- `Ec2HandleUserData: Message: Could not find <persist> and </persist>` - persist 태그가 없는 경우
- `Ec2HandleUserData: Message: The output from user scripts` - 사용자 데이터 스크립트가 실행되는 경우 해당 출력이 로깅됨

이후 재부팅 또는 시작 시 스크립트 실행

인스턴스 사용자 데이터를 업데이트하는 경우 다음에 인스턴스를 재부팅하거나 시작할 때 업데이트된 사용자 데이터 콘텐츠가 인스턴스 메타데이터에 자동으로 반영됩니다. 그러나 설치된 시작 에이전트에 따라 후속 재부팅 또는 시작 시 실행하도록 사용자 데이터 스크립트를 구성하려면 추가 구성이 필요할 수 있습니다.

Sysprep으로 종료 옵션을 선택하는 경우 후속 재부팅 또는 시작시 사용자 데이터 실행을 활성화하지 않았더라도 다음에 인스턴스가 재부팅되거나 시작될 때 사용자 데이터 스크립트가 실행됩니다.

사용자 데이터 실행을 활성화하는 지침을 보려면 시작 에이전트와 일치하는 탭을 선택합니다.

EC2Launch v2

EC2Launch v1과 달리 EC2Launch v2는 부팅할 때마다 사용자 데이터 작업을 평가합니다. 사용자 데이터 작업을 수동으로 예약하지 않아도 됩니다. 사용자 데이터는 포함된 빈도 또는 지속 옵션에 따라 실행됩니다.

XML 사용자 데이터 스크립트의 경우

부팅할 때마다 사용자 데이터 스크립트를 실행하려면 사용자 데이터에 <persist>true</persist> 플래그를 추가하세요. 지속 플래그가 포함되지 않은 경우 사용자 데이터 스크립트는 초기 부팅 시에만 실행됩니다.

YAML 사용자 데이터의 경우

- 초기 부팅 시 사용자 데이터에서 작업을 실행하려면 `frequency` 작업을 `once`로 설정하세요.
- 부팅할 때마다 사용자 데이터에서 작업을 실행하려면 `frequency` 작업을 `always`로 설정하세요.

EC2Launch

1. Windows 인스턴스에 연결합니다.
2. PowerShell 명령 창을 열고 다음 명령 중 하나를 실행하세요.

   한 번 실행

   다음 부팅 시 사용자 데이터를 한 번 실행하려면 `-Schedule` 플래그를 사용하세요.

   ```
   C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -
   Schedule
   ```

   모든 후속 부팅에서 실행

   모든 후속 부팅 시 사용자 데이터를 실행하려면 `-SchedulePerBoot` 플래그를 사용하세요.

   ```
   C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -
   SchedulePerBoot
   ```

3. Windows 인스턴스에서 연결을 해제합니다. 다음 번에 인스턴스가 시작될 때 업데이트된 스크립트를 실행하려면 인스턴스를 중지하고 사용자 데이터를 업데이트합니다.

EC2Config

1. Windows 인스턴스에 연결합니다.
2. Open `C:\Program Files\Amazon\Ec2ConfigService\Ec2ConfigServiceSetting.exe`.

3. 사용자 데이터에서 Enable UserData execution for next service start(다음 서비스 시작 시 사용자 데이터 실행 활성화)를 선택합니다.
4. Windows 인스턴스에서 연결을 해제합니다. 다음 번에 인스턴스가 시작될 때 업데이트된 스크립트를 실행하려면 인스턴스를 중지하고 사용자 데이터를 업데이트합니다.

사용자 데이터 및 Tools for Windows PowerShell

Tools for Windows PowerShell을 사용하여 인스턴스의 사용자 데이터를 지정, 수정 및 확인할 수 있습니다. 인스턴스 메타데이터를 사용하여 인스턴스의 사용자 데이터를 보는 방법에 대한 자세한 내용은 EC2 인스턴스에 대한 인스턴스 메타데이터에 액세스 섹션을 참조하세요. 사용자 데이터 및 AWS CLI에 대한 자세한 내용은 사용자 데이터 및 AWS CLI 섹션을 참조하세요.

예시: 시작 시 인스턴스 사용자 데이터 지정

인스턴스 사용자 데이터를 포함하는 텍스트 파일을 생성합니다. 매번 인스턴스를 재부팅하거나 시작할 때마다 사용자 데이터 스크립트를 실행하려면 다음 예제와 같이 `<persist>true</persist>`를 추가합니다.

```
<powershell>
    $file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
    New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

인스턴스를 시작할 때 인스턴스 사용자 데이터를 지정하려면 New-EC2Instance 명령을 사용합니다. 이 명령은 사용자 데이터의 base64 인코딩을 수행하지 않습니다. 다음 명령을 사용하여 사용자 데이터를 `script.txt` 이름의 텍스트 파일로 인코딩합니다.

```
PS C:\> $Script = Get-Content -Raw script.txt
PS C:\> $UserData =
 [System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($Script))
```

`-UserData` 파라미터를 사용하여 사용자 데이터를 New-EC2Instance 명령으로 전달합니다.

```
PS C:\> New-EC2Instance -ImageId ami-abcd1234 -MinCount 1 -MaxCount 1 -
InstanceType m3.medium \
    -KeyName my-key-pair -SubnetId subnet-12345678 -SecurityGroupIds sg-1a2b3c4d \
    -UserData $UserData
```

예시: 중지된 인스턴스에 대한 인스턴스 사용자 데이터 업데이트

Edit-EC2InstanceAttribute 명령을 사용하여 중지된 인스턴스의 사용자 데이터를 수정할 수 있습니다.

새 스크립트를 포함하는 텍스트 파일을 생성합니다. 다음 명령을 사용하여 사용자 데이터를 new-script.txt 이름의 텍스트 파일로 인코딩합니다.

```
PS C:\> $NewScript = Get-Content -Raw new-script.txt
PS C:\> $NewUserData =
 [System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($NewScript))
```

-UserData 및 -Value 파라미터를 사용하여 사용자 데이터를 지정합니다.

```
PS C:\> Edit-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -Attribute userData -
Value $NewUserData
```

예시: 인스턴스 사용자 데이터 보기

인스턴스의 사용자 데이터를 가져오려면 Get-EC2InstanceAttribute 명령을 사용합니다.

```
PS C:\> (Get-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -Attribute
 userData).UserData
```

다음은 예시 출력입니다. 사용자 데이터가 인코딩됩니다.

```
PHBvd2Vyc2hlbGw
+DQpSZW5hbWUtQ29tcHV0ZXIgLU5ld05hbWUgdXNlci1kYXRhLXRlc3QNCjwvcG93ZXJzaGVsbD4=
```

다음 명령을 사용하여 인코딩된 사용자 데이터를 변수에 저장한 다음 디코딩합니다.

```
PS C:\> $UserData_encoded = (Get-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -
Attribute userData).UserData
PS C:
\> [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($UserData_encoded)
```

다음은 예시 출력입니다.

```
<powershell>
    $file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
    New-Item $file -ItemType file
</powershell>
```

```
<persist>true</persist>
```

예시: 태그 값과 일치하도록 인스턴스의 이름 바꾸기

Get-EC2Tag 명령을 사용하여 태그 값을 읽고 처음 부팅할 때 태그 값과 일치하도록 인스턴스 이름을 바꾼 다음 재부팅할 수 있습니다. 이 명령을 성공적으로 실행하려면 API 호출로 태그 정보를 가져오기 때문에 인스턴스에 `ec2:DescribeTags` 권한이 연결된 역할이 있어야 합니다. IAM 역할을 사용하여 권한을 설정하는 방법에 대한 자세한 내용은 IAM 역할을 인스턴스에 연결 섹션을 참조하세요.

IMDSv2

```
<powershell>
    [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} -Method PUT -Uri 'http://169.254.169.254/latest/api/token' -
UseBasicParsing
    $instanceId = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" =
 $token} -Method GET -Uri 'http://169.254.169.254/latest/meta-data/instance-id' -
UseBasicParsing
 $nameValue = (Get-EC2Tag -Filter @{Name="resource-id";Value=
$instanceid},@{Name="key";Value="Name"}).Value
 $pattern = "^(?![0-9]{1,15}$)[a-zA-Z0-9-]{1,15}$"
 #Verify Name Value satisfies best practices for Windows hostnames
 If ($nameValue -match $pattern)
     {Try
         {Rename-Computer -NewName $nameValue -Restart -ErrorAction Stop}
     Catch
         {$ErrorMessage = $_.Exception.Message
         Write-Output "Rename failed: $ErrorMessage"}}
 Else
     {Throw "Provided name not a valid hostname. Please ensure Name value is between
 1 and 15 characters in length and contains only alphanumeric or hyphen characters"}
</powershell>
```

IMDSv1

```
<powershell>
 $instanceId = (Invoke-WebRequest http://169.254.169.254/latest/meta-data/instance-
id -UseBasicParsing).content
 $nameValue = (Get-EC2Tag -Filter @{Name="resource-id";Value=
$instanceid},@{Name="key";Value="Name"}).Value
 $pattern = "^(?![0-9]{1,15}$)[a-zA-Z0-9-]{1,15}$"
 #Verify Name Value satisfies best practices for Windows hostnames
```

```
 If ($nameValue -match $pattern)
     {Try
         {Rename-Computer -NewName $nameValue -Restart -ErrorAction Stop}
     Catch
         {$ErrorMessage = $_.Exception.Message
         Write-Output "Rename failed: $ErrorMessage"}}
 Else
     {Throw "Provided name not a valid hostname. Please ensure Name value is between
 1 and 15 characters in length and contains only alphanumeric or hyphen characters"}
</powershell>
```

인스턴스가 인스턴스 메타데이터의 태그에 액세스하도록 구성된 경우 인스턴스 메타데이터의 태그를 사용하여 인스턴스 이름을 바꿀 수도 있습니다. 자세한 내용은 인스턴스 메타데이터를 사용하여 EC2 인스턴스의 태그 보기 섹션을 참조하세요.

IMDSv2

```
<powershell>
    [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} -Method PUT -Uri 'http://169.254.169.254/latest/api/token' -
UseBasicParsing
 $nameValue = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token}
 -Method GET -Uri 'http://169.254.169.254/latest/meta-data/tags/instance/Name' -
UseBasicParsing
 $pattern = "^(?![0-9]{1,15}$)[a-zA-Z0-9-]{1,15}$"
 #Verify Name Value satisfies best practices for Windows hostnames
 If ($nameValue -match $pattern)
     {Try
         {Rename-Computer -NewName $nameValue -Restart -ErrorAction Stop}
     Catch
         {$ErrorMessage = $_.Exception.Message
         Write-Output "Rename failed: $ErrorMessage"}}
 Else
     {Throw "Provided name not a valid hostname. Please ensure Name value is between
 1 and 15 characters in length and contains only alphanumeric or hyphen characters"}
</powershell>
```

IMDSv1

```
<powershell>
 $nameValue = Get-EC2InstanceMetadata -Path /tags/instance/Name
 $pattern = "^(?![0-9]{1,15}$)[a-zA-Z0-9-]{1,15}$"
```

```
 #Verify Name Value satisfies best practices for Windows hostnames
 If ($nameValue -match $pattern)
     {Try
         {Rename-Computer -NewName $nameValue -Restart -ErrorAction Stop}
     Catch
         {$ErrorMessage = $_.Exception.Message
         Write-Output "Rename failed: $ErrorMessage"}}
 Else
     {Throw "Provided name not a valid hostname. Please ensure Name value is between
 1 and 15 characters in length and contains only alphanumeric or hyphen characters"}
</powershell>
```

단일 요청으로 시작된 각 인스턴스를 식별합니다.

이 예제는 사용자 데이터 및 인스턴스 메타데이터를 사용하여 Amazon EC2 인스턴스를 구성하는 방법을 보여줍니다.

> **Note**
>
> 이 섹션의 예에서는 IMDS의 IPv4 주소(`169.254.169.254`)를 사용합니다. IPv6 주소를 통해 EC2 인스턴스의 인스턴스 메타데이터를 검색하는 경우, 대신 IPv6 주소(`[fd00:ec2::254]`)를 활성화하고 사용해야 합니다. IMDS의 IPv6 주소는 IMDSv2 명령과 호환됩니다. IPv6 주소는 IPv6 지원 서브넷(듀얼 스택 또는 IPv6 전용)의 Nitro 기반 인스턴스에서만 액세스할 수 있습니다.

Alice는 데이터베이스 AMI 인스턴스 4개를 시작하여 그 중 첫 번째 인스턴스는 원래 인스턴스의 역할을 하고 나머지 3개는 복제본의 역할을 하도록 하려고 합니다. 그러한 인스턴스는 시작되었을 때 각 복제본의 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. 1개 이상의 인스턴스를 동시에 시작하는 경우 `ami-launch-index`는 인스턴스가 시작되는 순서를 나타냅니다. 첫 번째 인스턴스의 값은 `0`입니다.

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 정의하는 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 다음 표와 같이 각 인스턴스에는 고유한 메타데이터가 있습니다.

메타데이터	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

메타데이터	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

메타데이터	값
instance-id	i-0ee992212549ce0e7
ami-launch-index	2
public-hostname	ec2-67-202-51-225.compute-1.amazonaws.com
public-ipv4	67.202.51.225
local-hostname	ip-10-251-50-37.ec2.internal
local-ipv4	10.251.50.37

메타데이터	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

메타데이터	값
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" 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"
    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"
 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
```

호스트가 EC2 인스턴스인지 감지

특히 혼합 컴퓨팅 환경이 있는 경우 애플리케이션 또는 웹 사이트가 EC2 인스턴스에서 실행되고 있는지 알아야 할 수 있습니다. 다음 옵션 중 하나를 사용하여 애플리케이션 또는 웹 사이트의 호스트가 EC2 인스턴스인지 확인할 수 있습니다.

옵션

- 인스턴스 자격 증명 문서 검사
- 시스템 UUID 검사
- 시스템 가상 머신 생성 식별자 검사

인스턴스 자격 증명 문서 검사

각 인스턴스에는 암호화 방식으로 확인할 수 있는 서명된 인스턴스 ID 문서가 있습니다. 인스턴스 메타데이터 서비스(IMDS)를 사용하여 이러한 문서를 찾을 수 있습니다.

자세한 내용은 인스턴스 자격 증명 문서 섹션을 참조하세요.

시스템 UUID 검사

시스템 UUID를 가져와 UUID의 첫 번째 8진수에서 EC2(Linux에서는 소문자 ec2일 수 있음)를 찾을 수 있습니다. 이 방법은 빠르지만 EC2 인스턴스가 아닌 시스템에 이러한 문자로 시작하는 UUID가 있을 가능성이 적기 때문에 잠재적으로는 부정확할 수 있습니다. 또한 일부 버전의 SMBIOS는 UUID의 시작 부분에 EC2가 포함되지 않는 little-endian 형식을 사용합니다. 이는 Windows용 SMBIOS 2.4를 사용하는 EC2 인스턴스 또는 자체 SMBIOS 구현이 있는 Amazon Linux 이외의 Linux 배포판의 경우에 해당할 수 있습니다.

Linux 예: DMI에서 UUID 가져오기(HVM AMI만 해당)

데스크톱 관리 인터페이스(DMI)를 사용하여 UUID를 가져오려면 다음 명령을 사용합니다.

```
[ec2-user ~]$ sudo dmidecode --string system-uuid
```

다음 출력 예에서 UUID가 "EC2"로 시작합니다. 이는 시스템이 아마도 EC2 인스턴스라는 것을 나타냅니다.

```
EC2E1916-9099-7CAF-FD21-012345ABCDEF
```

다음 예시 출력에서 UUID는 리틀 엔디안 형식으로 표시됩니다.

```
45E12AEC-DCD1-B213-94ED-012345ABCDEF
```

또는 Nitro 시스템에 구축된 인스턴스의 경우 다음 명령을 사용할 수 있습니다.

```
[ec2-user ~]$ cat /sys/devices/virtual/dmi/id/board_asset_tag
```

출력이 인스턴스 ID인 경우 다음 예시 출력과 같이 시스템은 EC2 인스턴스입니다.

```
i-0af01c0123456789a
```

Linux 예: 하이퍼바이저에서 UUID 가져오기(PV AMI만 해당)

다음 명령을 사용하여 하이퍼바이저에서 UUID를 가져옵니다.

```
[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
```

Windows 예: WMI 또는 Windows PowerShell을 사용하여 UUID 가져오기

다음과 같이 Windows Management Instrumentation 명령줄(WMIC)을 사용합니다.

```
wmic path win32_computersystemproduct get uuid
```

또는 Windows PowerShell을 사용하는 경우 다음과 같이 Get-WmiObject cmdlet을 사용합니다.

```
PS C:\> Get-WmiObject -query "select uuid from Win32_ComputerSystemProduct" | Select
 UUID
```

다음 출력 예에서 UUID가 "EC2"로 시작합니다. 이는 시스템이 아마도 EC2 인스턴스라는 것을 나타냅니다.

```
EC2AE145-D1DC-13B2-94ED-012345ABCDEF
```

SMBIOS 2.4를 사용하는 인스턴스의 경우 UUID를 리틀 엔디안 형식으로 나타낼 수 있습니다. 예를 들면 다음과 같습니다.

```
45E12AEC-DCD1-B213-94ED-012345ABCDEF
```

시스템 가상 머신 생성 식별자 검사

가상 머신 생성 식별자는 암호화 임의의 정수 식별자로 해석되는 128비트의 고유 버퍼로 구성됩니다. 가상 머신 생성 식별자를 검색하여 Amazon Elastic Compute Cloud 인스턴스를 식별할 수 있습니다. 생성 식별자는 ACPI 테이블 항목을 통해 인스턴스의 게스트 운영 체제 내에 노출됩니다. 머신을 복제 또는 복사하거나 VM Import/Export 등을 통해 AWS로 가져오는 경우 값이 변경됩니다.

예: Linux에서 가상 머신 생성 식별자 검색

다음과 같은 명령을 사용하여 Linux를 실행하는 인스턴스에서 가상 머신 생성 식별자를 검색할 수 있습니다.

Amazon Linux 2

1. 필요하면 다음과 같은 명령을 사용하여 기존 소프트웨어 패키지를 업데이트합니다.

```
sudo yum update
```

2. 필요한 경우 다음과 같은 명령을 사용하여 busybox 패키지를 소싱합니다.

```
sudo curl https://www.rpmfind.net/linux/epel/next/8/Everything/x86_64/Packages/
b/busybox-1.35.0-2.el8.next.x86_64.rpm --output busybox.rpm
```

3. 필요한 경우 다음과 같은 명령을 사용하여 필수 구성 요소 패키지를 설치합니다.

```
sudo yum install busybox.rpm iasl -y
```

4. 다음과 같은 `iasl` 명령을 실행하여 ACPI 테이블에서 출력을 생산합니다.

```
sudo iasl -p ./SSDT2 -d /sys/firmware/acpi/tables/SSDT2
```

5. 다음과 같은 명령을 실행하여 `iasl` 명령의 출력을 검토합니다.

```
cat SSDT2.dsl
```

출력에서는 가상 머신 생성 식별자를 검색하는 데 필요한 주소 공간이 산출되어야 합니다.

```
Intel ACPI Component Architecture
ASL+ Optimizing Compiler/Disassembler version 20190509
Copyright (c) 2000 - 2019 Intel Corporation

File appears to be binary: found 32 non-ASCII characters, disassembling
Binary file appears to be a valid ACPI table, disassembling
Input file /sys/firmware/acpi/tables/SSDT2, Length 0x7B (123) bytes
ACPI: SSDT 0x0000000000000000 00007B (v01 AMAZON AMZNSSDT 00000001 AMZN
 00000001)
Pass 1 parse of [SSDT]
Pass 2 parse of [SSDT]
Parsing Deferred Opcodes (Methods/Buffers/Packages/Regions)

Parsing completed
Disassembly completed
ASL Output:    ./SSDT2.dsl - 1065 bytes
$
```

```
/*
* Intel ACPI Component Architecture
* AML/ASL+ Disassembler version 20190509 (64-bit version)
* Copyright (c) 2000 - 2019 Intel Corporation
*
* Disassembling to symbolic ASL+ operators
*
* Disassembly of /sys/firmware/acpi/tables/SSDT2, Tue Mar 29 16:15:14 2022
*
* Original Table Header:
*     Signature        "SSDT"
*     Length           0x0000007B (123)
*     Revision         0x01
*     Checksum         0xB8
*     OEM ID           "AMAZON"
*     OEM Table ID     "AMZNSSDT"
*     OEM Revision     0x00000001 (1)
*     Compiler ID      "AMZN"
*     Compiler Version 0x00000001 (1)
*/
DefinitionBlock ("", "SSDT", 1, "AMAZON", "AMZNSSDT", 0x00000001)
{
Scope (\_SB)
{
    Device (VMGN)
    {
        Name (_CID, "VM_Gen_Counter")  // _CID: Compatible ID
        Name (_DDN, "VM_Gen_Counter")  // _DDN: DOS Device Name
        Name (_HID, "AMZN0000")  // _HID: Hardware ID
        Name (ADDR, Package (0x02)
        {
            0xFED01000,
            Zero
        })
    }
}
}
```

6. (선택 사항) 다음과 같은 명령으로 나머지 단계에 대한 터미널 권한을 승격합니다.

```
sudo -s
```

7. 다음과 같은 명령을 사용하여 이전에 수집한 주소 공간을 저장합니다.

```
VMGN_ADDR=0xFED01000
```

8. 다음과 같은 명령을 사용하여 주소 공간을 반복하고 가상 머신 생성 식별자를 구축합니다.

```
for offset in 0x0 0x4 0x8 0xc; do busybox devmem $(($VMGN_ADDR + $offset)) | sed
 's/0x//' | sed -z '$ s/\n$//' >> vmgenid; done
```

9. 다음과 같은 명령으로 출력 파일에서 가상 머신 생성 식별자를 검색합니다.

```
cat vmgenid ; echo
```

다음과 유사하게 출력되어야 합니다.

```
EC2F335D979132C4165896753E72BD1C
```

Ubuntu

1. 필요하면 다음과 같은 명령을 사용하여 기존 소프트웨어 패키지를 업데이트합니다.

```
sudo apt update
```

2. 필요한 경우 다음과 같은 명령을 사용하여 필수 구성 요소 패키지를 설치합니다.

```
sudo apt install busybox iasl -y
```

3. 다음과 같은 `iasl` 명령을 실행하여 ACPI 테이블에서 출력을 생산합니다.

```
sudo iasl -p ./SSDT2 -d /sys/firmware/acpi/tables/SSDT2
```

4. 다음과 같은 명령을 실행하여 `iasl` 명령의 출력을 검토합니다.

```
cat SSDT2.dsl
```

출력에서는 가상 머신 생성 식별자를 검색하는 데 필요한 주소 공간이 산출되어야 합니다.

```
Intel ACPI Component Architecture
ASL+ Optimizing Compiler/Disassembler version 20190509
Copyright (c) 2000 - 2019 Intel Corporation
```

```
File appears to be binary: found 32 non-ASCII characters, disassembling
Binary file appears to be a valid ACPI table, disassembling
Input file /sys/firmware/acpi/tables/SSDT2, Length 0x7B (123) bytes
ACPI: SSDT 0x0000000000000000 00007B (v01 AMAZON AMZNSSDT 00000001 AMZN
 00000001)
Pass 1 parse of [SSDT]
Pass 2 parse of [SSDT]
Parsing Deferred Opcodes (Methods/Buffers/Packages/Regions)

Parsing completed
Disassembly completed
ASL Output:    ./SSDT2.dsl - 1065 bytes
$
/*
* Intel ACPI Component Architecture
* AML/ASL+ Disassembler version 20190509 (64-bit version)
* Copyright (c) 2000 - 2019 Intel Corporation
*
* Disassembling to symbolic ASL+ operators
*
* Disassembly of /sys/firmware/acpi/tables/SSDT2, Tue Mar 29 16:15:14 2022
*
* Original Table Header:
*     Signature        "SSDT"
*     Length           0x0000007B (123)
*     Revision         0x01
*     Checksum         0xB8
*     OEM ID           "AMAZON"
*     OEM Table ID     "AMZNSSDT"
*     OEM Revision     0x00000001 (1)
*     Compiler ID      "AMZN"
*     Compiler Version 0x00000001 (1)
*/
DefinitionBlock ("", "SSDT", 1, "AMAZON", "AMZNSSDT", 0x00000001)
{
Scope (\_SB)
{
    Device (VMGN)
    {
        Name (_CID, "VM_Gen_Counter")  // _CID: Compatible ID
        Name (_DDN, "VM_Gen_Counter")  // _DDN: DOS Device Name
        Name (_HID, "AMZN0000")  // _HID: Hardware ID
        Name (ADDR, Package (0x02)
```

```
        {
            0xFED01000,
            Zero
        })
    }
}
}
```

5. (선택 사항) 다음과 같은 명령으로 나머지 단계에 대한 터미널 권한을 승격합니다.

```
sudo -s
```

6. 다음과 같은 명령을 사용하여 이전에 수집한 주소 공간을 저장합니다.

```
VMGN_ADDR=0xFED01000
```

7. 다음과 같은 명령을 사용하여 주소 공간을 반복하고 가상 머신 생성 식별자를 구축합니다.

```
for offset in 0x0 0x4 0x8 0xc; do busybox devmem $(($VMGN_ADDR + $offset)) | sed
 's/0x//' | sed -z '$ s/\n$//' >> vmgenid; done
```

8. 다음과 같은 명령으로 출력 파일에서 가상 머신 생성 식별자를 검색합니다.

```
cat vmgenid ; echo
```

다음과 유사하게 출력되어야 합니다.

```
EC2F335D979132C4165896753E72BD1C
```

예: Windows에서 가상 머신 생성 식별자 검색

샘플 애플리케이션을 생성하여 Windows를 실행하는 인스턴스에서 가상 머신 생성 식별자를 검색할 수 있습니다. 자세한 내용은 Microsoft 설명서의 가상 머신 생성 식별자 얻기 섹션을 참조하세요.

Amazon EC2 인스턴스의 인스턴스 자격 증명 문서

시작하는 각 인스턴스에는 인스턴스 자체에 대한 정보를 제공하는 인스턴스 자격 증명 문서가 있습니다. 인스턴스 자격 증명 문서를 사용하여 인스턴스의 속성을 확인할 수 있습니다.

인스턴스 자격 증명 문서는 인스턴스를 중지했다가 시작하거나 다시 시작하거나 시작할 때 생성됩니다. 인스턴스 메타데이터 서비스(IMDS)를 통해 인스턴스에 대한 인스턴스 ID 문서에 액세스할 수 있습니다. 해당 지침은 인스턴스 자격 증명 문서 검색을(를) 참조하십시오.

인스턴스 ID 문서는 일반 텍스트 JSON 형식을 사용합니다. 여기에는 다음 정보가 포함됩니다.

Data	설명
accountId	인스턴스를 시작한 AWS 계정의 ID.
architecture	인스턴스를 시작하는 데 사용된 AMI의 아키텍처(i386 | x86_64 | arm64).
availabilityZone	인스턴스가 실행 중인 가용 영역.
billingProducts	인스턴스의 결제 제품입니다.
devpayProductCodes	사용되지 않음.
imageId	인스턴스를 시작하는 데 사용된 AMI의 ID.
instanceId	인스턴스의 ID
instanceType	인스턴스의 인스턴스 유형입니다.
kernelId	인스턴스에 연결된 커널의 ID(해당되는 경우).
marketplaceProductCodes	인스턴스를 시작하는 데 사용된 AMI의 AWS Marketplace 제품 코드.
pendingTime	인스턴스가 시작된 날짜와 시간.
privateIp	인스턴스의 프라이빗 IPv4 주소.
ramdiskId	인스턴스에 연결된 RAM 디스크의 ID(해당되는 경우).
region	인스턴스가 실행 중인 리전.
version	인스턴스 자격 증명 문서 형식의 버전.

EC2 인스턴스에 대한 인스턴스 ID 문서 검색

Amazon EC2 인스턴스에 대한 인스턴스 ID 문서는 일반 텍스트 JSON 형식을 사용합니다. 인스턴스 ID 문서의 내용에 대한 설명은 the section called “인스턴스 자격 증명 문서” 섹션을 참조하세요.

인스턴스 ID 문서는 `instance-identity/document` 동적 데이터 범주에서 인스턴스에 대한 인스턴스 메타데이터에 저장됩니다. 인스턴스에 연결하고 인스턴스 메타데이터에서 검색하여 인스턴스에 대한 인스턴스 ID 문서에 액세스할 수 있습니다.

IPv4 주소 169.254.169.254 또는 IPv6 주소 fd00:ec2::254를 사용하여 인스턴스 메타데이터에 액세스할 수 있습니다. 링크-로컬 주소이므로 인스턴스에서만 액세스할 수 있습니다. 이 페이지의 예제에서는 IMDS의 IPv4 주소(169.254.169.254)를 사용합니다. IPv6를 통해 EC2 인스턴스에 대한 인스턴스 메타데이터를 검색하려면 fd00:ec2::254를 사용합니다.

검색한 후 인스턴스 ID 문서의 신뢰성을 확인하려면 인스턴스 자격 증명 문서 확인을 참조하세요.

IMDSv2

Linux

Linux 인스턴스에서 다음 명령을 실행하여 인스턴스 ID 문서를 검색합니다.

```
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" http://169.254.169.254/latest/dynamic/instance-identity/document
```

Windows

Windows 인스턴스에서 다음 cmdlet을 실행하여 인스턴스 ID 문서를 검색합니다.

```
[string]$token = (Invoke-WebRequest -Headers @{'X-aws-ec2-metadata-token-ttl-seconds' = '21600'} `
    http://169.254.169.254/latest/api/token).Content
```

```
(Invoke-WebRequest -Headers @{'X-aws-ec2-metadata-token' = $Token} `
    http://169.254.169.254/latest/dynamic/instance-identity/document).Content
```

IMDSv1

Linux

Linux 인스턴스에서 다음 명령을 실행하여 인스턴스 ID 문서를 검색합니다.

```
curl http://169.254.169.254/latest/dynamic/instance-identity/document
```

Windows

Windows 인스턴스에서 다음 cmdlet을 실행하여 인스턴스 ID 문서를 검색합니다.

```
(Invoke-WebRequest http://169.254.169.254/latest/dynamic/instance-identity/
document).Content
```

출력의 예시는 다음과 같습니다.

```
{
    "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"
}
```

Amazon EC2 인스턴스의 인스턴스 자격 증명 문서 확인

중요한 목적으로 인스턴스 자격 증명 문서의 내용을 사용하려는 경우 사용 전에 해당 내용과 신뢰성을 확인해야 합니다.

일반 텍스트 인스턴스 자격 증명 문서에는 해시되고 암호화된 3개의 서명이 함께 제공됩니다. 이러한 서명을 사용하여 인스턴스 자격 증명 문서의 출처 및 신뢰성과 포함된 정보를 확인할 수 있습니다. 다음과 같은 서명이 제공됩니다.

- Base64 인코딩 서명—RSA 키 페어를 사용하여 암호화된 인스턴스 자격 증명 문서의 Base64로 인코딩된 SHA256 해시입니다.
- PKCS7 서명—DSA 키 페어를 사용하여 암호화된 인스턴스 자격 증명 문서의 SHA1 해시입니다.
- RSA-2048 서명—RSA-2048 키 페어를 사용하여 암호화된 인스턴스 자격 증명 문서의 SHA256 해시입니다.

각 서명은 인스턴스 메타데이터의 서로 다른 엔드포인트에서 사용할 수 있습니다. 해시 및 암호화 요구 사항에 따라 이러한 서명 중 하나를 사용할 수 있습니다. 서명을 확인하려면 해당 AWS 퍼블릭 인증서를 사용해야 합니다.

옵션

- 옵션 1: PKCS7 서명을 사용하여 인스턴스 자격 증명 문서 확인
- 옵션 2: Base64 인코딩 서명을 사용하여 인스턴스 자격 증명 문서 확인
- 옵션 3: RSA-2048 서명을 사용하여 인스턴스 자격 증명 문서 확인

옵션 1: PKCS7 서명을 사용하여 인스턴스 자격 증명 문서 확인

이 주제에서는 PKCS7 서명 및 AWS DSA 퍼블릭 인증서를 사용하여 인스턴스 자격 증명 문서를 확인하는 방법을 설명합니다.

Linux 인스턴스

PKCS7 서명 및 AWS DSA 퍼블릭 인증서를 사용하여 인스턴스 자격 증명 문서를 확인하려면

1. 인스턴스에 연결합니다.
2. 인스턴스 메타데이터에서 PKCS7 서명을 검색하여 pkcs7이라는 새 파일에, 필요한 헤더 및 푸터와 함께 추가합니다. 인스턴스에서 사용하는 IMDS 버전에 따라 다음 명령 중 하나를 사용합니다.

   IMDSv2

```
$ echo "-----BEGIN PKCS7-----" >> pkcs7 \
 && 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" http://169.254.169.254/latest/dynamic/instance-identity/pkcs7 >> pkcs7 \
 && echo "" >> pkcs7 \
 && echo "-----END PKCS7-----" >> pkcs7
```

IMDSv1

```
$ echo "-----BEGIN PKCS7-----" >> pkcs7 \
 && curl -s http://169.254.169.254/latest/dynamic/instance-identity/pkcs7
 >> pkcs7 \
 && echo "" >> pkcs7 \
 && echo "-----END PKCS7-----" >> pkcs7
```

3. 인스턴스 ID 문서 서명을 위한 AWS 퍼블릭 인증서에서 리전의 DSA 퍼블릭 인증서를 찾고 콘텐츠를 새 `certificate` 파일에 추가합니다.
4. OpenSSL smime 명령을 사용하여 서명을 확인합니다. 서명을 확인해야 함을 나타내는 `-verify` 옵션과 인증서를 확인할 필요가 없음을 나타내는 `-noverify` 옵션을 포함합니다.

```
$ openssl smime -verify -in pkcs7 -inform PEM -certfile certificate -noverify | tee
 document
```

서명이 유효하면 `Verification successful` 메시지가 나타납니다.

또한 이 명령은 인스턴스 ID 문서의 내용을 document라는 새 파일에 씁니다. 다음 명령을 사용하여 인스턴스 메타데이터의 인스턴스 ID 문서 내용을 이 파일의 내용과 비교할 수 있습니다.

```
$ openssl dgst -sha256 < document
```

```
$ curl -s -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/
dynamic/instance-identity/document | openssl dgst -sha256
```

서명을 확인할 수 없는 경우 지원에 문의하세요.

Windows 인스턴스

사전 조건

이 절차를 수행하려면 `System.Security` Microsoft .NET Core 클래스가 필요합니다. PowerShell 세션에 클래스를 추가하려면 다음 명령을 실행합니다.

```
PS C:\> Add-Type -AssemblyName System.Security
```

> **Note**
>
> 이 명령은 현재 PowerShell 세션에만 클래스를 추가합니다. 새 세션을 시작하는 경우 명령을 다시 실행해야 합니다.

PKCS7 서명 및 AWS DSA 퍼블릭 인증서를 사용하여 인스턴스 자격 증명 문서를 확인하려면

1. 인스턴스에 연결합니다.
2. 인스턴스 메타데이터에서 PKCS7 서명을 검색하여 바이트 배열로 변환한 다음 `$Signature`라는 변수에 추가합니다. 인스턴스에 사용되는 IMDS 버전에 따라 다음 명령 중 하나를 사용합니다.

   IMDSv2

   ```
   PS C:\> [string]$token = (Invoke-WebRequest -Method Put -Headers @{'X-aws-
   ec2-metadata-token-ttl-seconds' = '21600'} http://169.254.169.254/latest/api/
   token).Content
   ```

   ```
   PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest -Headers
    @{'X-aws-ec2-metadata-token' = $Token} http://169.254.169.254/latest/dynamic/
   instance-identity/pkcs7).Content)
   ```

   IMDSv1

   ```
   PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest
    http://169.254.169.254/latest/dynamic/instance-identity/pkcs7).Content)
   ```

3. 인스턴스 메타데이터에서 일반 텍스트 인스턴스 자격 증명 문서를 검색하여 바이트 배열로 변환한 다음 `$Document`라는 변수에 추가합니다. 인스턴스에서 사용하는 IMDS 버전에 따라 다음 명령 중 하나를 사용합니다.

   IMDSv2

   ```
   PS C:\> $Document = [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest -Headers
    @{'X-aws-ec2-metadata-token' = $Token} http://169.254.169.254/latest/dynamic/
   instance-identity/document).Content)
   ```

IMDSv1

```
PS C:\> $Document =  [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest
 http://169.254.169.254/latest/dynamic/instance-identity/document).Content)
```

4. 인스턴스 ID 문서 서명을 위한 AWS 퍼블릭 인증서에서 리전의 DSA 퍼블릭 인증서를 찾고 콘텐츠를 새 `certificate.pem` 파일에 추가합니다.
5. 인증서 파일에서 인증서를 추출하여 `$Store`라는 변수에 저장합니다.

```
PS C:\> $Store =
 [Security.Cryptography.X509Certificates.X509Certificate2Collection]::new([Security.Cryptog
Path certificate.pem)))
```

6. 서명을 확인합니다.

```
PS C:\> $SignatureDocument = [Security.Cryptography.Pkcs.SignedCms]::new()
```

```
PS C:\> $SignatureDocument.Decode($Signature)
```

```
PS C:\> $SignatureDocument.CheckSignature($Store, $true)
```

서명이 유효하면 명령에서 출력이 반환되지 않으며, 서명을 확인할 수 없으면 명령에서 `Exception calling "CheckSignature" with "2" argument(s): "Cannot find the original signer`가 반환됩니다. 서명을 확인할 수 없는 경우 AWS Support에 문의하세요.

7. 인스턴스 자격 증명 문서의 내용을 확인합니다.

```
PS C:
\> [Linq.Enumerable]::SequenceEqual($SignatureDocument.ContentInfo.Content, $Document)
```

인스턴스 자격 증명 문서의 내용이 유효하면 명령에서 `True`가 반환됩니다. 인스턴스 ID 문서를 검증할 수 없는 경우 AWS Support에 문의하세요.

옵션 2: Base64 인코딩 서명을 사용하여 인스턴스 자격 증명 문서 확인

이 주제에서는 base64 인코딩 서명 및 AWS RSA 퍼블릭 인증서를 사용하여 인스턴스 자격 증명 문서를 확인하는 방법을 설명합니다.

Linux 인스턴스

base64 인코딩 서명 및 AWS RSA 퍼블릭 인증서를 사용하여 인스턴스 자격 증명 문서를 확인하려면

1. 인스턴스에 연결합니다.
2. 인스턴스 메타데이터에서 base64 인코딩 서명을 검색하여 이진수로 변환한 다음 `signature`라는 변수에 추가합니다. 인스턴스에 사용되는 IMDS 버전에 따라 다음 명령 중 하나를 사용합니다.

   IMDSv2

   ```
   $ 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" http://169.254.169.254/latest/dynamic/instance-identity/signature | base64 -d >> signature
   ```

   IMDSv1

   ```
   $ curl -s http://169.254.169.254/latest/dynamic/instance-identity/signature |
    base64 -d >> signature
   ```

3. 인스턴스 메타데이터에서 일반 텍스트 인스턴스 자격 증명 문서를 검색하여 `document`라는 파일에 추가합니다. 인스턴스에서 사용하는 IMDS 버전에 따라 다음 명령 중 하나를 사용합니다.

   IMDSv2

   ```
   $ 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" http://169.254.169.254/latest/dynamic/instance-identity/document >> document
   ```

   IMDSv1

   ```
   $ curl -s http://169.254.169.254/latest/dynamic/instance-identity/document
    >> document
   ```

4. 인스턴스 ID 문서 서명을 위한 AWS 퍼블릭 인증서에서 리전의 RSA 퍼블릭 인증서를 찾고 콘텐츠를 새 `certificate` 파일에 추가합니다.
5. AWS RSA 퍼블릭 인증서에서 퍼블릭 키를 추출하여 `key`라는 파일에 저장합니다.

   ```
   $ openssl x509 -pubkey -noout -in certificate >> key
   ```

6. OpenSSL dgst 명령을 사용하여 인스턴스 자격 증명 문서를 확인합니다.

```
$ openssl dgst -sha256 -verify key -signature signature document
```

서명이 유효하면 `Verification successful` 메시지가 나타납니다.

또한 이 명령은 인스턴스 ID 문서의 내용을 document라는 새 파일에 씁니다. 다음 명령을 사용하여 인스턴스 메타데이터의 인스턴스 ID 문서 내용을 이 파일의 내용과 비교할 수 있습니다.

```
$ openssl dgst -sha256 < document
```

```
$ curl -s -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/
dynamic/instance-identity/document | openssl dgst -sha256
```

서명을 확인할 수 없는 경우 지원에 문의하세요.

Windows 인스턴스

base64 인코딩 서명 및 AWS RSA 퍼블릭 인증서를 사용하여 인스턴스 자격 증명 문서를 확인하려면

1. 인스턴스에 연결합니다.
2. 인스턴스 메타데이터에서 base64 인코딩 서명을 검색하여 바이트 배열로 변환한 다음 $Signature라는 변수에 추가합니다. 인스턴스에 사용되는 IMDS 버전에 따라 다음 명령 중 하나를 사용합니다.

   IMDSv2

```
PS C:\> [string]$token = (Invoke-WebRequest -Method Put -Headers @{'X-aws-
ec2-metadata-token-ttl-seconds' = '21600'} http://169.254.169.254/latest/api/
token).Content
```

```
PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest -Headers
 @{'X-aws-ec2-metadata-token' = $Token} http://169.254.169.254/latest/dynamic/
instance-identity/signature).Content)
```

IMDSv1

```
PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest
 http://169.254.169.254/latest/dynamic/instance-identity/signature).Content)
```

3. 인스턴스 메타데이터에서 일반 텍스트 인스턴스 자격 증명 문서를 검색하여 바이트 배열로 변환한 다음 $Document라는 변수에 추가합니다. 인스턴스에서 사용하는 IMDS 버전에 따라 다음 명령 중 하나를 사용합니다.

   IMDSv2

   ```
   PS C:\> $Document = [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest -Headers
    @{'X-aws-ec2-metadata-token' = $Token} http://169.254.169.254/latest/dynamic/
   instance-identity/document).Content)
   ```

   IMDSv1

   ```
   PS C:\> $Document =  [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest
    http://169.254.169.254/latest/dynamic/instance-identity/document).Content)
   ```

4. 인스턴스 ID 문서 서명을 위한 AWS 퍼블릭 인증서에서 리전의 RSA 퍼블릭 인증서를 찾고 콘텐츠를 새 certificate.pem 파일에 추가합니다.
5. 인스턴스 자격 증명 문서를 확인합니다.

   ```
   PS C:\> [Security.Cryptography.X509Certificates.X509Certificate2]::new((Resolve-
   Path certificate.pem)).PublicKey.Key.VerifyData($Document, 'SHA256', $Signature)
   ```

   서명이 유효하면 명령에서 True가 반환됩니다. 서명을 확인할 수 없는 경우 지원에 문의하세요.

옵션 3: RSA-2048 서명을 사용하여 인스턴스 자격 증명 문서 확인

이 주제에서는 RSA-2048 서명 및 AWS RSA-2048 퍼블릭 인증서를 사용하여 인스턴스 자격 증명 문서를 확인하는 방법을 설명합니다.

Linux 인스턴스

RSA-2048 서명 및 AWS RSA-2048 퍼블릭 인증서를 사용하여 인스턴스 자격 증명 문서를 확인하려면

1. 인스턴스에 연결합니다.

2. 인스턴스 메타데이터에서 RSA-2048 서명을 검색하여 `rsa2048`이라는 파일에 필요한 헤더 및 푸터와 함께 추가합니다. 인스턴스에서 사용하는 IMDS 버전에 따라 다음 명령 중 하나를 사용합니다.

   IMDSv2

   ```
   $ echo "-----BEGIN PKCS7-----" >> rsa2048 \
    && 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" http://169.254.169.254/latest/
   dynamic/instance-identity/rsa2048 >> rsa2048 \
    && echo "" >> rsa2048 \
    && echo "-----END PKCS7-----" >> rsa2048
   ```

   IMDSv1

   ```
   $ echo "-----BEGIN PKCS7-----" >> rsa2048 \
    && curl -s http://169.254.169.254/latest/dynamic/instance-identity/rsa2048
    >> rsa2048 \
    && echo "" >> rsa2048 \
    && echo "-----END PKCS7-----" >> rsa2048
   ```

3. 인스턴스 ID 문서 서명을 위한 AWS 퍼블릭 인증서에서 리전의 RSA-2048 퍼블릭 인증서를 찾고 콘텐츠를 새 `certificate` 파일에 추가합니다.

4. OpenSSL smime 명령을 사용하여 서명을 확인합니다. 서명을 확인해야 함을 나타내는 `-verify` 옵션과 인증서를 확인할 필요가 없음을 나타내는 `-noverify` 옵션을 포함합니다.

   ```
   $ openssl smime -verify -in rsa2048 -inform PEM -certfile certificate -noverify |
    tee document
   ```

   서명이 유효하면 `Verification successful` 메시지가 나타납니다. 서명을 확인할 수 없는 경우 지원에 문의하세요.

Windows 인스턴스

사전 조건

이 절차를 수행하려면 `System.Security` Microsoft .NET Core 클래스가 필요합니다. PowerShell 세션에 클래스를 추가하려면 다음 명령을 실행합니다.

```
PS C:\> Add-Type -AssemblyName System.Security
```

> **Note**
>
> 이 명령은 현재 PowerShell 세션에만 클래스를 추가합니다. 새 세션을 시작하는 경우 명령을 다시 실행해야 합니다.

RSA-2048 서명 및 AWS RSA-2048 퍼블릭 인증서를 사용하여 인스턴스 자격 증명 문서를 확인하려면

1. 인스턴스에 연결합니다.
2. 인스턴스 메타데이터에서 RSA-2048 서명을 검색하여 바이트 배열로 변환한 다음 `$Signature`라는 변수에 추가합니다. 인스턴스에 사용되는 IMDS 버전에 따라 다음 명령 중 하나를 사용합니다.

   IMDSv2

   ```
   PS C:\> [string]$token = (Invoke-WebRequest -Method Put -Headers @{'X-aws-
   ec2-metadata-token-ttl-seconds' = '21600'} http://169.254.169.254/latest/api/
   token).Content
   ```

   ```
   PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest -Headers
    @{'X-aws-ec2-metadata-token' = $Token} http://169.254.169.254/latest/dynamic/
   instance-identity/rsa2048).Content)
   ```

   IMDSv1

   ```
   PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest
    http://169.254.169.254/latest/dynamic/instance-identity/rsa2048).Content)
   ```

3. 인스턴스 메타데이터에서 일반 텍스트 인스턴스 자격 증명 문서를 검색하여 바이트 배열로 변환한 다음 `$Document`라는 변수에 추가합니다. 인스턴스에서 사용하는 IMDS 버전에 따라 다음 명령 중 하나를 사용합니다.

IMDSv2

```
PS C:\> $Document = [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest -Headers
 @{'X-aws-ec2-metadata-token' = $Token} http://169.254.169.254/latest/dynamic/
instance-identity/document).Content)
```

IMDSv1

```
PS C:\> $Document =  [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest
 http://169.254.169.254/latest/dynamic/instance-identity/document).Content)
```

4. 인스턴스 ID 문서 서명을 위한 AWS 퍼블릭 인증서에서 리전의 RSA-2048 퍼블릭 인증서를 찾고 콘텐츠를 새 `certificate.pem` 파일에 추가합니다.
5. 인증서 파일에서 인증서를 추출하여 `$Store`라는 변수에 저장합니다.

```
PS C:\> $Store =
 [Security.Cryptography.X509Certificates.X509Certificate2Collection]::new([Security.Cryptog
Path certificate.pem)))
```

6. 서명을 확인합니다.

```
PS C:\> $SignatureDocument = [Security.Cryptography.Pkcs.SignedCms]::new()
```

```
PS C:\> $SignatureDocument.Decode($Signature)
```

```
PS C:\> $SignatureDocument.CheckSignature($Store, $true)
```

서명이 유효하면 명령에서 출력이 반환되지 않으며, 서명을 확인할 수 없으면 명령에서 `Exception calling "CheckSignature" with "2" argument(s): "Cannot find the original signer`가 반환됩니다. 서명을 확인할 수 없는 경우 AWS Support에 문의하세요.

7. 인스턴스 자격 증명 문서의 내용을 확인합니다.

```
PS C:
\> [Linq.Enumerable]::SequenceEqual($SignatureDocument.ContentInfo.Content, $Document)
```

인스턴스 자격 증명 문서의 내용이 유효하면 명령에서 `True`가 반환됩니다. 인스턴스 ID 문서를 검증할 수 없는 경우 AWS Support에 문의하세요.

인스턴스 ID 문서 서명을 위한 AWS 퍼블릭 인증서

인스턴스 자격 증명 문서 확인에서 설명하는 것과 같이 다음 AWS 퍼블릭 인증서를 사용하여 인스턴스의 인스턴스 자격 증명 문서 콘텐츠를 확인할 수 있습니다.

리전 및 사용 중인 확인 절차에 맞는 인증서를 사용하는지 확인합니다. PKCS7 서명을 확인하는 경우 DSA 인증서를 사용합니다. base64로 인코딩된 서명을 확인하는 경우에는 RSA 인증서를 사용합니다. RSA-2048 서명을 확인하는 경우 RSA-2048 인증서를 사용합니다.

리전별 인증서를 보려면 아래 각 리전을 확장합니다.

미국 동부(버지니아 북부) - us-east-1

DSA

```
-----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-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUE1y2NIKCU+Rg4uu4u32koG9QEYIwDQYJKoZIhvcNAQEL
```

```
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE3MzQwMVoXDTI5MDQyODE3MzQwMVowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUE1y2NIKC
U+Rg4uu4u32koG9QEYIwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQAlxSmwcWnhT4uAeSinJuz+1BTcKhVSWb5jT8pYjQb8ZoZkXXRGb09mvYeU
NeqOBr27rvRAnaQ/9LUQf72+SahDFuS4CMI8nwowytqbmwquqFr4dxA/SDADyRiF
ea1UoMuNHTY49J/1vPomqsVn7mugTp+TbjqCfOJTpu0temHcFA==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJALFpzEAVWaQZMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA4MTQw
ODU5MTJaGA8yMTk1MDExNzA4NTkxMlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAjS2vqZu9mEOhOq+0bRpAbCUiapbZMFNQqRg7kTlr7Cf+gDqXKpHPjsng
SfNz+JHQd8WPI+pmNs+q0Z2aTe23klmf2U52KH9/j1k8RlIbap/yFibFTSedmegX
E5r447GbJRsHUmuIIfZTZ/oRlpuIIO5/Vz7SOj22tdkdY2ADp7caZkNxhSP915fk
2jJMTBUOzyXUS2rBU/ulNHbTTeePjcEkvzVYPahD30TeQ+/A+uWUu89bHSQOJR8h
Um4cFApzZgN3aD5j2LrSMu2pctkQwf9CaWyVznqrsGYjYOY66LuFzSCXwqSnFBfv
fFBAFsjCgY24G2DoMyYkF3MyZlu+rwIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUrynSPp4uqSECwy+PiO4qyJ8TWSkwgY4GA1UdIwSBhjCBg4AUrynS
Pp4uqSECwy+PiO4qyJ8TWSmhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJALFpzEAVWaQZMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBADW/s8lXijwdP6NkEoH1m9XLrvK4YTqkNfR6
er/uRRgTx2QjFcMNrx+g87gAml11z+D0crAZ5LbEhDMs+JtZYR3ty0HkDk6SJM85
haoJNAFF7EQ/zCp1EJRIkLLsC7bcDL/Eriv1swt78/BB4RnC9W9kSp/sxd5svJMg
N9a6FAplpNRsWAnbP8JBlAP93oJzblX2LQXgykTghMkQO7NaY5hg/H5o4dMPclTK
lYGqlFUCH6A2vdrxmpKDLmTn5//5pujdD2MN0df6sZWtxwZ0osljV4rDjm9Q3VpA
NWIsDEcp3GUB4proOR+C7PNkY+VGODitBOw09qBGosCBstwyEqY=
```

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

미국 동부(오하이오) - us-east-2

DSA

```
-----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-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUVJTc+hOU+8Gk3JlqsX438Dk5c58wDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE3MTE0OVoXDTI5MDQyODE3MTE0OVowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUVJTc+hOU
+8Gk3JlqsX438Dk5c58wEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
```

```
AAOBgQAywJQaVNWJqW0R0T0xVOSoN1GLk9x9kKEuN67RN9CLin4dA97qa7Mr5W4P
FZ6vnh5CjOhQBRXV9xJUeYSdqVItNAUFK/fEzDdjf1nUfPlQ3OJ49u6CV01NoJ9m
usvY9kWcV46dqn2bk2MyfTTgvmeqP8fiMRPxxnVRkSzlldP5Fg==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAM07oeX4xevdMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNjA2MTAx
MjU4MThaGA8yMTk1MTExNDEyNTgxOFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA6v6kGMnRmFDLxBEqXzP4npnL65OO0kmQ7w8YXQygSdmNIoScGSU5wfh9
mZdcvCxCdxgALFsFqPvH8fqiE9ttI0fEfuZvHOs8wUsIdKr0Zz0MjSx3cik4tKET
ch0EKfMnzKOgDBavraCDeX1rUDU0Rg7HFqNAOry3uqDmnqtk00XC9GenS3z/7ebJ
fIBEPAam5oYMVFpX6M6St77WdNE8wEU8SuerQughiMVx9kMB07imeVHBiELbMQ0N
lwSWRL/61fA02keGSTfSp/0m3u+lesf2VwVFhqIJs+JbsEscPxOkIRlzy8mGd/JV
ONb/DQpTedzUKLgXbw7KtO3HTG9iXQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU2CTGYE5fTjx7gQXzdZSGPEWAJY4wgY4GA1UdIwSBhjCBg4AU2CTG
YE5fTjx7gQXzdZSGPEWAJY6hYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAM07oeX4xevdMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBANdqkIpVypr2PveqUsAKke1wKCOSuw1UmH9k
xX1/VRoHbrI/UznrXtPQOPMmHA2LKSTedwsJuorUn3cFH6qNs8ixBDrl8pZwfKOY
IBJcTFBbI1xBEFkZoO3wczzo5+8vPQ60RVqAaYb+iCa1HFJpccC3Ovajfa4GRdNb
n6FYnluIcDbmpcQePoVQwX7W3oOYLB1QLN7fE6H1j4TBIsFdO3OuKzmaifQlwLYt
DVxVCNDabpOr6Uozd5ASm4ihPPoEoKo7Ilp0fOT6fZ41U2xWA4+HF/89UoygZSo7
K+cQ90xGxJ+gmlYbLFR5rbJOLfjrgDAb2ogbFy8LzHo2ZtSe60M=
-----END CERTIFICATE-----
```

미국 서부(캘리포니아 북부) - us-west-1

DSA

```
-----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-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUK2zmY9PUSTR7rc1k2OwPYu4+g7wwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE3MDI0M1oXDTI5MDQyODE3MDI0M1owXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUK2zmY9PU
STR7rc1k2OwPYu4+g7wwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQA1Ng4QmN4n7iPh5CnadSOc0ZfM7by0dBePwZJyGvOHdaw6P6E/vEk76KsC
Q8p+akuzVzVPkU4kBK/TRqLp19wEWoVwhhTaxHjQ1tTRHqXIVlrkw4JrtFbeNM21
GlkSLonuzmNZdivn9WuQYeGe7nUD4w3q9GgiF3CPorJe+UxtbA==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJANNPkIpcyEtIMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTEwMjkw
OTAzMDdaGA8yMTk1MDQwMzA5MDMwN1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
```

```
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEApHQGvHvq3SVCzDrC7575BW7GWLzcj8CLqYcL3YY7Jffupz7OjcftO57Z
4fo5Pj0CaS8DtPzh8+8vdwUSMbiJ6cDd3ooio3MnCq6DwzmsY+pY7CiI3UVG7KcH
4TriDqr1Iii7nB5MiPJ8wTeAqX89T3SYaf6Vo+4GCb3LCDGvnkZ9TrGcz2CHkJsj
AIGwgopFpwhIjVYm7obmuIxSIUv+oNH0wXgDL029Zd98SnIYQd/njiqkzE+lvXgk
4h4Tu17xZIKBgFcTtWPky+POGu81DYFqiWVEyR2JKKm2/iR1dL1YsT39kbNg47xY
aR129sS4nB5Vw3TRQA2jL0ToTIxzhQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUgepyiONs8j+q67dmcWu+mKKDa+gwgY4GA1UdIwSBhjCBg4AUgepy
iONs8j+q67dmcWu+mKKDa+ihYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJANNPkIpcyEtIMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAGLFWyutf1u0xcAc+kmnMPqtc/Q6b79VIX0E
tNoKMI2KR8lcV8ZElXDb0NC6v8UeLpe1WBKjaWQtEjL1ifKg9hdY9RJj4RXIDSK7
33qCQ8juF4vep2U5TTBd6hfWxt1Izi88xudjixmbpUU4YKr8UPbmixldYR+BEx0u
B1KJi9l1lxvuc/Igy/xeHOAZEjAXzVvHp8Bne33VVwMiMxWECZCiJxE4I7+Y6fqJ
pLLSFFJKbNaFyXlDiJ3kXyePEZSc1xiWeyRB2ZbTi5eu7vMG4i3AYWuFVLthaBgu
lPfHafJpj/JDcqt2vKUKfur5edQ6j1CGdxqqjawhOTEqcN8m7us=
-----END CERTIFICATE-----
```

미국 서부(오레곤) - us-west-2

DSA

```
-----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-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUFx8PxCkbHwpD31bOyCtyz3GclbgwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE3MjM1OVoXDTI5MDQyODE3MjM1OVowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUFx8PxCkb
HwpD31bOyCtyz3GclbgwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQBzOl+9Xy1+UsbUBI95HO9mbbdnuX+aMJXgG9uFZNjgNEbMcvx+h8P9IMko
z7PzFdheQQ1NLjsHH9mSR1SyC4m9ja6BsejH5nLBWyCdjfdP3muZM4O5+r7vUa1O
dWU+hP/T7DUrPAIVMOE7mpYa+WPWJrN6BlRwQkKQ7twm9kDalA==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJALZL3lrQCSTMMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA4MTQw
OTAxMzJaGA8yMTk1MDExNzA5MDEzMlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA02Y59qtAA0a6uzo7nEQcnJ26OKF+LRPwZfixBH+EbEN/Fx0gYy1jpjCP
s5+VRNg6/WbfqAsV6X2VSjUKN59ZMnMY9ALA/Ipz0n00Huxj38EBZmX/NdNqKm7C
qWu1q5kmIvYjKGiadfboU8wLwLcHo8ywvfgI6FiGGsEO9VMC56E/hL6Cohko11LW
dizyvRcvg/IidazVkJQCN/4zC9PUOVyKdhW33jXy8BTg/QH927QuNk+ZzD7HH//y
tIYxDhR6TIZsSnRjz3bOcEHxt1nsidc65mY0ejQty4hy7ioSiapw316mdbtE+RTN
fcH9FPIFKQNBpiqfAW5Ebp3Lal3/+wIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU7coQx8Qnd75qA9XotSWT3IhvJmowgY4GA1UdIwSBhjCBg4AU7coQ
x8Qnd75qA9XotSWT3IhvJmqhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJALZL3lrQCSTMMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAFZ1e2MnzRaXCaLwEC1pW/f0oRG8nHrlPZ9W
OYZEWbh+QanRgaikBNDtVTwARQcZm3z+HWSkaIx3cyb6vM0DSkZuiwzm1LJ9rDPc
```

```
aBm03SEt5v8mcc7sXWvgFjCnUpzosmky6JheCD4O1Cf8k0olZ93FQnTrbg62OK0h
83mGCDeVKU3hLH97FYoUq+3N/IliWFDhvibAYYKFJydZLhIdlCiiB99AM6Sg53rm
oukS3csyUxZyTU2hQfdjyo1nqW9yhvFAKjnnggiwxNKTTPZzstKW8+cnYwiiTwJN
QpVoZdt0SfbuNnmwRUMi+QbuccXweav29QeQ3ADqjgB0CZdSRKk=
-----END CERTIFICATE-----
```

아프리카(케이프타운) – af-south-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7DCCAqwCCQCncbCtQbjuyzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xOTA2MDQxMjQ4MDVaFw00
NTA2MDQxMjQ4MDVaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbYwggErBgcqhkjOOAQBMIIBHgKBgQC12Nr1gMrHcFSZ7S/A
pQBSCMHWmn2qeoQTMVWqe50fnTd0zGFxDdIjKxUK58/8zjWG5uR4TXRzmZpGpmXB
bSufAR6BGqud2LnT/HIWGJAsnX2uOtSyNfCoJigqwhea5w+CqZ6I7iBDdnB4TtTw
qO6TlnExHFVj8LMkylZgiaE1CQIVAIhdobse4K0QnbAhCL6R2euQzloXAoGAV/21
WUuMz/79Ga0JvQcz1FNy1sT0pU9rU4TenqLQIt5iccn/7EIfNtvVO5TZKulIKq7J
gXZr0x/KIT8zsNweetLOaGehPIYRMPX0vunMMR7hN7qA7W17WZv/76adywIsnDKq
ekfe15jinaX8MsKUdyDK7Y+ifCG4PVhoM4+W2XwDgYQAAoGAIxOKbVgwLxbn6Pi2
6hBOihFv16jKxAQI0hHzXJLV0Vyv9QwnqjJJRfOCy3dB0zicLXiIxeIdYfvqJr+u
hlN8rGxEZYYJjEUKMGvsc0DW85jonXz0bNfcP0aaKH0lKKVjL+OZi5n2kn9wgdo5
F3CVnMl8BUra8A1Tr2yrrE6TVZ4wCQYHKoZIzjgEAwMvADAsAhQfa7MCJZ+/TEY5
AUr0J4wm8VzjoAIUSYZVu2NdRJ/ERPmDfhW5EsjHlCA=
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICNjCCAZ+gAwIBAgIJAKumfZiRrNvHMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xOTExMjcw
NzE0MDVaGA8yMTk5MDUwMjA3MTQwNVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQDFd571nUzVtke3rPyRkYfvs3jh0C0EMzzG72boyUNjnfw1+m0TeFraTLKb9T6F
7TuB/ZEN+vmlYqr2+5Va8U8qLbPF0bRH+FdaKjhgWZdYXxGzQzU3ioy5W5ZM1VyB
7iUsxEAlxsybC3ziPYaHI42UiTkQNahmoroNeqVyHNnBpQIDAQABMA0GCSqGSIb3
DQEBCwUAA4GBAAJLylWyElEgOpW4B1XPyRVD4pAds8Guw2+krgqkY0HxLCdjosuH
```

```
RytGDGN+q75aAoXzW5a7SGpxLxk6Hfv0xp3RjDHsoeP0i1d8MD3hAC5ezxS4oukK
s5gbPOnokhKTMPXbTdRn5ZifCbWlx+bYN/mTYKvxho7b5SVg2o1La9aK
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAIFI+O5A6/ZIMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xOTA2MDQx
MjQ4MDRaGA8yMTk4MTEwNzEyNDgwNFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAy7/WHBBHOrk+20aumT07g8rxrSM0UXgki3eYgKauPCG4Xx//vwQbuZwI
oeVmR9nqnhfij2wOcQdbLandh0EGtbxerete3IoXzd1KXJb11PVmzrzyu5SPBPuP
iCeV4qdjjkXo2YWM6t9YQ911hcG96YSp89TBXFYUh3KLxfqAdTVhuC0NRGhXpyii
j/czo9njofHhqhTr7UEyPun8NVS2QWctLQ86N5zWR3Q0GRoVqqMrJs0cowHTrVw2
9Qr7QBjjBOVbyYmtYxm/DtiKprYV/e6bCAVok015X1sZDd3oCOQNoGlv5XbHJe2o
JFD8GRRy2rkWO/lNwVFDcwec6zC3QwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQCE
goqzjpCpmMgCpszFHwvRaSMbspKtK7wNImUjrSBOfBJsfFulyg1Zgn2nDCK7kQhx
jMJmNIvXbps3yMqQ2cHUkKcKf5t+WldfeT4Vk1Rz6HSA8sd0kgVcIesIaoy2aaXU
VEB/oQziRGyKdN1d4TGYVZXG44CkrzSDvlbmfiTq5tL+kAieznVF3bzHgPZW6hKP
EXC3G/IXrXicFEe6YyE1Rakl62VncYSXiGe/i2XvsiNH3Qlmnx5XS7W0SCN0oAxW
EH9twibauv82DVg1WOkQu8EwFw8hFde9X0Rkiu0qVcuU8lJgFEvPWMDFU5sGB6ZM
gkEKTzMvlZpPbBhg99Jl
-----END CERTIFICATE-----
```

아시아 태평양(홍콩) - ap-east-1

DSA

```
-----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-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICSzCCAbQCCQDtQvkVxRvK9TANBgkqhkiG9w0BAQsFADBqMQswCQYDVQQGEwJV
UzETMBEGA1UECBMKV2FzaGluZ3RvbjEQMA4GA1UEBxMHU2VhdHRsZTEYMBYGA1UE
ChMPQW1hem9uLmNvbSBJbmMuMRowGAYDVQQDExFlYzIuYW1hem9uYXdzLmNvbTAe
Fw0xOTAyMDMwMzAwMDZaFw0yOTAyMDIwMzAwMDZaMGoxCzAJBgNVBAYTAlVTMRMw
EQYDVQQIEwpXYXNoaW5ndG9uMRAwDgYDVQQHEwdTZWF0dGxlMRgwFgYDVQQKEw9B
bWF6b24uY29tIEluYy4xGjAYBgNVBAMTEWVjMi5hbWF6b25hd3MuY29tMIGfMA0G
CSqGSIb3DQEBAQUAA4GNADCBiQKBgQC1kkHXYTfc7gY5Q55JJhjTieHAgacaQkiR
Pity9QPDE3b+NXDh4UdP1xdIw73JcIIG3sG9RhWiXVCHh6KkuCTqJfPUknIKk8vs
M3RXflUpBe8Pf+P92pxqPMCz1Fr2NehS3JhhpkCZVGxxwLC5gaG0Lr4rFORubjYY
Rh84dK98VwIDAQABMA0GCSqGSIb3DQEBCwUAA4GBAA6xV9f0HMqXjPHuGILDyaNN
dKcvplNFwDTydVg32MNubAGnecoEBtUPtxBsLoVYXCOb+b5/ZMDubPF9tU/vSXuo
TpYM5Bq57gJzDRaBOntQbX9bgHiUxw6XZWaTS/6xjRJDT5p3S1E0mPI3lP/eJv4o
Ezk5zb3eIf10/sqt4756
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAMoxixvs3YssMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xODA3MjAw
ODQ0NDRaGA8yMTk3MTIyMzA4NDQ0NFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA4T1PNsOg0FDrGlWePoHeOSmOJTA3HCRy5LSbYD33GFU2eBrOIxoU/+SM
rInKu3GghAMfH7WxPW3etIAZiyTDDU5RLcUq2Qwdr/ZpXAWpYocNc/CEmBFtfbxF
z4uwBIN3/drM0RSbe/wP9EcgmNUGQMMZWeAji8sMtwpOblNWAP9BniUG0Flcz6Dp
uPovwDTLdAYT3TyhzlohKL3f6O48TR5yTaV+3Ran2SGRhyJjfh3FRpP4VC+z5LnT
WPQHN74Kdq35UgrUxNhJraMGCzznolUuoR/tFMwR93401GsM9fVA7SW3jjCGF81z
PSzjy+ArKyQqIpLW1YGWDFk3sf08FQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQDK
2/+C3nPMgtyOFX/I3Cyk+Pui44IgOwCsIdNGwuJysdqp5VIfnjegEu2zIMWJSKGO
lMZoQXjffkVZZ97J7RNDW06oB7kj3WVE8a7U4WEOfnO/CbMUf/x99CckNDwpjgW+
```

```
K8V8SzAsQDvYZs2KaE+18GFfLVF1TGUYK2rPSZMHyX+v/TIlc/qUceBycrIQ/kke
jDFsihUMLqgmOV2hXKUpIsmiWMGrFQV4AeV0iXP8L/ZhcepLf1t5SbsGdUA3AUY1
3If8s81uTheiQjwY5t9nM0SY/1Th/tL3+RaEI79VNEVfG1FQ8mgqCK0ar4m0oZJl
tmmEJM7xeURdpBBx36Di
-----END CERTIFICATE-----
```

아시아 태평양(하이데라바드) - ap-south-2

DSA

```
-----BEGIN CERTIFICATE-----
MIIC8DCCArCgAwIBAgIGAXjrQ4+XMAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24g
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/
hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYUAAoGBAJCKGBBoxIUxqBk94JHhwZZbgvbP0DAOoHENQWxp/98lI7/
YOfYJOVMJS22aCnHDurofmo5rvNIkgXi7Rztbhu
+lko9rK6DgpmpUwBU0WZtf34aZ2IWNBwHaVhHvWAQf9/46u18dMa2YucK1Wi+Vc+M
+KldrvgXmhym6ErNlzhJyMAkGByqGSM44BAMDLwAwLAIUaaPKxaOHoYvwz709xXpsQueIq+UCFFa/
GpzoDOSokl1057NU/2hnsiW4
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICMzCCAZygAwIBAgIGAXjwLj9CMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIDBBXYXNoaW5n
+sFcobrjvcAYm0PNRD8f4R1jAzvoLt2+qGeOTAyO1Httj6cmsYN3AP1hN5iYuppFiYsl2eNPa/
CD0Vg0BAfDFlV5rzjpA0j7TJabVh4kj7JvtD+xYMi6wEQA4x6SPONY4OeZ2+8o/
HS8nucpWDVdPRO6ciWUlMhjmDmwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAAy6sgTdRkTqELHBeWj69q60xHyUmsWqHAQ
TGGbYP0yP2qfMl0cCImzRI5W0gn8gogdervfeT7nH5ih0TWEy/QDWfkQ601L4erm4yh4YQq8vcqAPSkf04N
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAIVWfPw/X82fMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMjA3MDQx
NDMwMjhaGA8yMjAxMTIwODE0MzAyOFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
```

```
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAg29QEFriG+qFEjYW/v62nN70lMJY/Hevx5TtmU/VIYBPQa3HUGTBAbbI
2Tmy8UMpa8kZeaYeI3RAfiQWt0Ws7wUrBuO2Pdp518WDPaJUH7RWEuu1BDDkyZRW
NAMNPCn3ph7Od243IFcLGku7HVekel5poqRpSfojrMasjlf+CvixUeAJbmFoxUHK
kh5unzG2sZyO4wHXcJPQkRf5a8zSTPe9YZP1kXPPEv4p/jTSggaYPxXyS6QVaTlV
zLeLFZ0fesLPMeil3KYQtV7IKLQiEA2F6dxWnxNWQlyMHtdq6PucfEmVx17i/Xza
yNBRo0azY8WUNVkEXrRhp/pU8Nh3GQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU9AO1aZk9RLXk2ZvRVoUxYvQy9uwwgY4GA1UdIwSBhjCBg4AU9AO1
aZk9RLXk2ZvRVoUxYvQy9uyhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAIVWfPw/X82fMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBADexluMRQRftqViahCnauEWGdMvLCBr8A+Yr
6hJq0guoxEk/lahxR137DnfMPuSbi1Rx5QKo7oBrWfG/zsgQUnF2IwHTzwD+i/2m
XCane6FiS5RpK3lGdILq8ZmlhQk+6iI8yoZLr0LCfTh+CLgIKHOknfR5lFzgzAiF
SI8/Q9mm+uvYtSTZECI6Zh57QZPoETAG/yl+9jiOy2lAelqa/k1i+Qo8gMfOc+Pm
dwY7o6fV+oucgRlsdey6VM45LeyILQqv0RXtVzjuowanzmCCFMjgqiO9oZAWu4Oh
+F3unijELo01vZJs8s2N3KGlo3/jtUFTX6RTKShZlAPLwBi5GMI=
-----END CERTIFICATE-----
```

아시아 태평양(자카르타) – ap-southeast-3

DSA

```
-----BEGIN CERTIFICATE-----
MIIC8DCCArCgAwIBAgIGAXbVDEikMAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24g
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/
hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYUAAoGBAPjuiEx05N3JQ6cVwntJie67D8OuNo4jGRn
+crEtL7YO0jSVB9zGE1ga
+UgRPIaYETL293S8rTJTVgXAqdpBwfaHC6NUzre8U8iJ8FMNnlP9Gw1oUIlgQBjORyynVJexoB31TDZM
+/52g9O/bpq1QqNyKbeIgyBBlc1dAtr1QLnsMAkGByqGSM44BAMDLwAwLAIUK8E6RDIRtwK+9qnaTOBhvO/
njuQCFFocyT1OxK+UDR888oNsdgtif2Sf
-----END CERTIFICATE-----
```

RSA

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

```
MIICMzCCAZygAwIBAgIGAXbVDG2yMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIDBBXYXNoaW5n
Vbt0gQ1ebWcur2hSO7PnJifE4OPxQ7RgSAlc4/spJp1sDP+ZrS0LO1ZJfKhXf1R9S3AUwLnsc7b
+IuVXdY5LK9RKqu64nyXP5dx170zoL8loEyCSuRR2fs+04i2QsWBVP+KFNAn7P5L1EHRjkgTO8kjNKviwRV
+OkP9ab5wIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAI4WUy6+DKh0JDSzQEZNyBgNlSoSuC2owtMxCwGB6nBfzzfcekWvs
+87w/g9lNwUnUt0ZHYyh2tuBG6hVJuUEwDJ/z3wDd6wQviLOTF3MITawt9P8siR1hXqLJNxpjRQFZrgHqi
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAMtdyRcH51j9MA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMjA0MDgx
MjM5MTZaGA8yMjAxMDkxMjEyMzkxNlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAvUsKCxoH6KXRYJLeYTWAQfaBQeCwhJaR56mfUeFHJE4g8aFjWkiN4uc1
TvOyYNnIZKTHWmzmulmdinWNbwP0GiROHb/i7ro0Hhvnptyyc Gt8ag8affiIbx5X
7ohdwSN2KJ6G0IKflIx7f2NEI0oAMM/9k+T1eVF+MVWzpZoiDp8frLNkqp8+RAgz
ScZsbRfwv3u/if5xJAvdg2nCkIWDMSHEVPoz0lJo7v0ZuDtWWsL1LHnL5ozvsKEk
+ZJyEi23r+U1hIT1NTBdp4yoigNQexedtwCSr7q36oOdDwvZpqYlkLi3uxZ4ta+a
01pzOSTwMLgQZSbKWQrpMvsIAPrxoQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU1GgnGdNpbnL3lLF30Jomg7Ji9hYwgY4GA1UdIwSBhjCBg4AU1Ggn
GdNpbnL3lLF30Jomg7Ji9hahYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAMtdyRcH51j9MBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBACVl00qQlatBKVeiWMrhpczsJroxDxlZTOba
6wTMZk7c3akb6XMOSZFbGaifkebPZqTHEhDlrClM2j9AIlYcCx6YCrTf4cuhn2mD
gcJN33143eOWSaeRY3ee4j+V9ne98y3kO2wLz95VrRgclPFR8po2iWGzGhwUi+FG
q8dXeCH3N0DZgQsSgQWwmdNQXZZej6RHLU/8In5trHKLY0ppnLBjn/UZQbeTyW5q
RJB3GaveXjfgFUWj2qOcDuRGaikdS+dYaLsi5z9cA3FolHzWxx9MOs8io8vKqQzV
XUrLTNWwuhZy88cOlqGPxnoRbw7TmifwPw/cunNrsjUUOgs6ZTk=
-----END CERTIFICATE-----
```

아시아 태평양(말레이시아) - ap-southeast-5

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq4CCQC5X6U+vgoLEDAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDAxMDMxMjU3NTRaGA8y
```

```
MDUwMDEwMzEyNTc1NFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0
b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIg
U2VydmljZXMgTExDMIIBtjCCASsGByqGSM44BAEwggEeAoGBAIzEMPCIPFfOYCg4
BCjKGyl60wOfmwHPzSOXZ3Z2wS/LYNHUtHGwtVNePSTyCu/CuZF6gC9n/wBORtQp
+SSkn+weGc/BmUA1mp/vrN7v+aSCgKJoO+sgpa1PPOgNvUaMw6O5odsZWQCMSjkU
6RTo/PL2v/tMfiCocF4ghvyRC6hvAhUA0VoObKC2IXzXgVvRRupo4qHbcm8CgYAe
bbNuawh3rAxkFvUs9FPzW5E+xllG16Z//61PENKqonmk+zBiBdiIlS1F6ZqmTqkI
z5+qfSt1m3pb3j2W0NT7lEDFvy8Gr6Y2vohCHmL+T1u1Yy4PeqbgrFwcen7y7WoO
/KCV7Y9/ODQMMyuAzT3h5wJNweT7L5MUN8JYpZSi3QOBhAACgYBqaDuG2u6V91Qj
K2wEAElxaaRaNo/ewg/wWDmHYqoeHOR0HfuFCYgASE9f7ULqYtX1VURcgcjw9XN4
BDmPilXvfi04INPTnw4IxFJKDzzC0kVH7esVas982Po8v3megH32H9Rl87r7UG1c
ZEbkSkKVX6YKYg1PR3rfjXgdwVZv/zAJBgcqhkjOOAQDAzAAMC0CFFWeRe2fYW2i
6mMd26Wzbx87Y0DXAhUAoPCnF+5hGJw0jT9aL7QsgcfLi9Y=
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICNjCCAZ+gAwIBAgIJAMuBl6rhZCJKMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDAxMDMx
MjU3NTRaGA8yMjAzMDYwOTEyNTc1NFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQDFuKydxZsordNH7bLwIluEGOkX7/CdLdpeqkDKEhQkFwzpRxaX4EAlkGh2/o7D
8qneC9cGQhqSG5WVVBrmZG7sfkFOM4m1AtY++kfv+MYto1VFgLk1xJbkpq1r4YeQ
Ul+ZsJYsZpyX/t+g8s7rW0OVcBsYx4L75bf34z38mwK8PQIDAQABMA0GCSqGSIb3
DQEBCwUAA4GBADD9C4pWL8RUvF1CJW8kExj35xmozlFlmrKs8Zpi8+Eg6q+W9dgd
xMdH95tgZtmVMDqlvVR+DK0iO1BNpqPjrqWkk2tTLivpS+sGzCE/jCl18Q28Rk71
/A3gLD7Rtbq5TKNvuFCHwYMjrTDHI6aBjIaAlDm4e2/j/OxVtHyZGTre
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJANC3xtbPhQ2GMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDAxMDMx
MjU3NTRaGA8yMjAzMDYwOTEyNTc1NFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAt3aMy7Hsp4ySG3mlfi+pdTcZw6H6XNU1Y36fNdi4c+MzinQQbnqMPyt7
QLgU+XCWmcWsVo7GQF6n9N01Rh+UXXUZU4jcXlFocQPCWf9O+IIIIPXkd67kFMUV
```

```
HAxCELjfxHbC+I8e7dwOJhmdF4Bfi52Ty8zz0HdE8JDypPkTDlXuGvTgDyW7NP56
I/vlQaXLoYSbcQe5pv2a9gyBaaCM1QoeqwAHAeCNXb9Nuj9ZX3GHGJb3TuqAeKCD
5i9TscCB9XjY6Fx+zfSAobjBZwgLEtLOwJhbZnKmx4gJMaaNFipAjVT2FSS3+yev
eTYBoa1dvhk0ivQyQIPpHmihrmkWuwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBC
fdgyI8GjmCqiHAlh+L1bj0LdNql9zl7RXmOEzsuRdtdMumkxYXX88UtROy3fdi1i
VaEwHdAK8ThzRkesgHza/cXzqCMewaYxujSI6p6G7x99FFeGif1x0FJdj8AoeTL7
4h9bmS/6l4/NL7DJI9G7ovES/hoUA9v9TDhv+vauxXlgfrp0MPecprxBYlrc+DH2
adGCdKCp2lQ2YDKOD9TCEjYIli8XSoyevoWHUjfDYrCrCp8l4s/p7HOgYr8fJBAs
EuVy8211LVz1/X4EMBRNtNjXK9sklsxAOX14NDfBFSS0toxl3K6Tf9t/PviBl95d
hncyDAcFgDCK4w8LLlVW
-----END CERTIFICATE-----
```

아시아 태평양(멜버른) – ap-southeast-4

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq
+gAwIBAgIGAXjWF7P2MAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxED
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/
hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYQAAoGAPRXSsQP9E3dw8QXKlrgBgEVCprLHdK/bbrMas0XMu1EhOD
+q
+0PcTr8+iwbtoXlY5MCeatWIp1GrXQjVqsF8vQqxlEuRuYKbR3nq4mWwaeGlx9AG5EjQHRa3GQ44wWHOdof0M3NRI1MP
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICMzCCAZygAwIBAgIGAXjSh40SMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIDBBXYXNoaW5n
+qWTGAbGsPeMX4hBMjAJUKys2NIRcRZaLM/BCew2FIPVjNtlaj6Gwn9ipU4Mlz3zIwAMWi1AvGMSreppt
+wV6MRtfOjh0Dvj/veJe88aEZJMozNgkJFRS
+WFWsckQeL56tf6kY6QTlNo8V/0CsQIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAF7vpPghH0FRo5gu49EArRNPrIvW1egM
wcgkqIwwuXYj+1rhlL+/
iMpQWjdVGEqIZSeXn5fLmdx50eegFCwND837r9e8XYTiQS143Sxt9+Yi6BZ7U7YD8kK9NBWoJxFqUeHdpRCs0O7COjT3
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAN4GTQ64zVs8MA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMjA3MTMx
MzMzMDBaGA8yMjAxMTIxNzEzMzMwMFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA2BYgeCr+Rk/jIAEDOHS7wJql62vc83QEwjuzk0qOFEReIZzlN1fBRNXK
g0T178Kd3gLYcE59wEFbTe/X5yOA1Lo95x1anSAo7R+Cisf9C2HQuJp+gVb+zx71
lniPF7gHziGpm0M8DdAU/IW+wkZwGbP4z7Hq9+bJ0P2ltvPJ5yxSgkFuDsI9VBHa
CLoprHsCHh2VdP8KcMgQQMmHe1NmBpyTk0ul/aLmQkCQEX6ZIRGOeq228fwlh/t+
Ho+jv87duihVKic6MrL32SlD+maX0LSDUydWdaOLLTGkh7oV7+bFuH6msrXUu+Ur
ZEP1r/MidCWMhfgrFzeTBzOHA97qxQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUcHMd1cHqzmsQ5hpUK3EMLhHdsi4wgY4GA1UdIwSBhjCBg4AUcHMd
1cHqzmsQ5hpUK3EMLhHdsi6hYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAN4GTQ64zVs8MBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAI4PFyVN+7EGSObioiPnv0LL0f70SSzUZJ8p
XO90d4rWea7jIbgZ2AKb+ErynkU9xVg7XQQ5k6KDWgp/4jYFL2dqnt/YAY4PS0un
RSrYElawxLT0BcLn4rcSDC79vQe1xGC5//wDdV6b399COAHRAK6axWYy5w32u9PL
uw0cIp3Ch8JoNwcgTHKRRGzePmBeR4PNqhHTArG4/dJk6/aUO4OpX0WzI6L67CGY
6Nex3dau+gkLCK93dTEkrXtyXHu4wB0J9zd1w+iQ0SEa9eKc78/NjEsF/FZdGrWC
t57lIMOOXJhQ1kRgSwNeZdQWV1dRakvO6sfcvVYkfj1wAvZvvAw=
-----END CERTIFICATE-----
```

아시아 태평양(뭄바이) - ap-south-1

DSA

```
-----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-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUDLA+x6tTAP3LRTr0z6nOxfsozdMwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE0MTMwMVoXDTI5MDQyODE0MTMwMVowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUDLA+x6tT
AP3LRTr0z6nOxfsozdMwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQAZ7rYKoAwwiiH1M5GJbrT/BEk3OO2VrEPw8ZxgpqQ/EKlzMlOs/0Cyrmp7
UYyUgYFQe5nq37Z94rOUSeMgv/WRxaMwrLlLqD78cuF9DSkXaZIX/kECtVaUnjk8
BZx0QhoIHOpQocJUSlm/dLeMuE0+0A3HNR6JVktGsUdv9ulmKw==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAPRYyD8TtmC0MA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNjAzMDcx
MDQ1MDFaGA8yMTk1MDgxMTEwNDUwMVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA0LSS5I/eCT2PM0+qusorBx67QL26BIWQHd/yF6ARtHBb/1DdFLRqE5Dj
07Xw7eENC+T79mOxOAbeWg91KaODOzw6i9I/2/HpK0+NDEdD6sPKDA1d45jRra+v
CqAjI+nV9Vw91wv7HjMk3RcjWGziM8/hw+3YNIutt7aQzZRwIWlBpcqx3/AFd8Eu
2UsRMSHgkGUW6UzUF+h/U8218XfrauKNGmNKDYUhtmyBrHT+k6J0hQ4pN7fe6h+Z
```

```
w9RVHm24BGhlLxLHLmsOIxvbrF277uX9Dxu1HfKfu5D2kimTY7xSZDNLR2dt+kNY
/+iWdIeEFpPT0PLSILt52wP6stF+3QIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBI
E6w+WWC2gCfoJO6c9HMyGLMFEpqZmz1n5IcQt1h9iyO7Vkm1wkJiZsMhXpk73zXf
TPxuXEacTX3SOEa07OIMCFwkusO5f6leOyFTynHCzBgZ3U0UkRVZA3WcpbNB6Dwy
h7ysVlqyT9WZd7EOYm5j5oue2G2xdei+6etgn5UjyWm6liZGrcOF6WPTdmzqa6WG
ApEqanpkQd/HM+hUYex/ZS6zEhd4CCDLgYkIjlrFbFb3pJ1OVLztIfSN5J4Oolpu
JVCfIq5u1NkpzL7ys/Ub8eYipbzI6P+yxXiUSuF0v9b98ymczMYjrSQXIf1e8In3
OP2CclCHoZ8XDQcvvKAh
-----END CERTIFICATE-----
```

아시아 태평양(오사카) – ap-northeast-3

DSA

```
-----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-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUHTRhxHhBZFOGvTFKxHoy9+f5Hl8wDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE2NTQwN1oXDTI5MDQyODE2NTQwN1owXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
```

```
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUHTRhxHhB
ZFOGvTFKxHoy9+f5Hl8wEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQAUZx7DcYbhWNTD4BNGhr5beruT2OUoGHH9J73UKxwdqeb9bH1LIWhIZO0X
/1mjn3bWBgCwfoS8gjZwsVB6fZbNBRy8urdBZJ87xF/4JPBjt7S9oGx/zthDUYrC
yK0Y0v4G0PgiS81CvYLg09LpmYhLSJbXENlkC04v5yxdKxZxyg==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAMn1yPk22ditMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNzA3MTkx
MTEyNThaGA8yMTk2MTIyMjExMTI1OFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEArznEYef8IjhrJoazI0QGZkmlmHm/4rEbyQbMNifxjsDE8YWtHNwaM91z
zmyK6Sk/tKlWxcnl3g31iq305ziyFPEewe5Qbwf1iz2cMsvfNBcTh/E6u+mBPH3J
gvGanqUJt6c4IbipdEouIjjnynyVWd4D6erLl/ENijeR1OxVpaqSW5SBK7jms49E
pw3wtbchEl3qsE42Ip4IYmWxqjgaxB7vps91n4kfyzAjUmklcqTfMfPCkzmJCRgp
Vh1C79vRQhmriVKD6BXwfZ8tG3a7mijeDn7kTsQzgO07Z2SAE63PIO48JK8HcObH
tXORUQ/XF1jzi/SIaUJZT7kq3kWl8wIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBj
ThtO9dLvU2QmKuXAhxXjsIdlQgGG3ZGh/Vke4If1ymgLx95v2Vj9Moxk+gJuUSRL
BzFte3TT6b3jPolbECgmAorjj8NxjC17N8QAAI1d0S0gI8kqkG7V8iRyPIFekv+M
pcai1+cIv5IV5qAz8QOMGYfGdYkcoBjsgiyvMJu/2N2UbZJNGWvcEGkdjGJUYYOO
NaspCAFm+6HA/K7BD9zXB1IKsprLgqhiIUgEaW3UFEbThJT+z8UfHG9fQjzzfN/J
nT6vuY/0RRu1xAZPyh2gr5okN/s6rnmh2zmBHU1n8cbCc64MVfXe2g3EZ9Glq/9n
izPrI09hMypJDP04ugQc
-----END CERTIFICATE-----
```

아시아 태평양(서울) - ap-northeast-2

DSA

```
-----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-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUbBSn2UIO6vYk4iNWV0RPxJJtHlgwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTEzMzg0NloXDTI5MDQyODEzMzg0NlowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUbBSn2UIO
6vYk4iNWV0RPxJJtHlgwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQAmjTjalG8MGLqWTC2uYqEM8nzI3px1eo0ArvFRsyqQ3fgmWcQpxExqUqRy
l3+2134Kv8dFab04Gut5wlfRtc2OwPKKicmv/IXGN+9bKFnQFjTqif08NIzrDZch
aFT/uvxrIiM+oN2YsHq66GUhO2+xVRXDXVxM/VObFgPERbJpyA==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJANuCgCcHtOJhMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
```

```
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA5MTQx
NTU3NDRaGA8yMTk1MDIxNzE1NTc0NFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA66iNv6pJPmGM20W8HbVYJSlKcAg2vUGx8xeAbzZIQdpGfkabVcUHGB6m
Gy59VXDMDlrJckDDk6dxUOhmcX9z785TtVZURq1fua9QosdbTzX4kAgHGdp4xQEs
mO6QZqg5qKjBP6xr3+PshfQ1rB8Bmwg0gXEm22CC7o77+7N7Mu2sWzWbiUR7vil4
9FjWS8XmMNwFTlShp4l1TDTevDWW/uYmC30RThM9S4QPvTZ0rAS18hHVam8BCTxa
LHaVCH/Yy52rsz0hM/FlghnSnK105ZKj+b+KIp3adBL8OMCjgc/Pxi0+j3HQLdYE
32+FaXWU84D2iP2gDT28evnstzuYTQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQC1
mA4q+12pxy7By6g3nBk1s34PmWikNRJBwOqhF8ucGRv8aiNhRRye9lokcXomwo8r
KHbbqvtK85l0xUZp/Cx4sm4aTgcMvfJP29jGLclDzeqADIvkWEJ4+xncxSYVlS9x
+78TvF/+8h9U2LnSl64PXaKdxHy2IsHIVRN4GtoaP2Xhpa1S0M328Jykq/571nfN
1WRD1c/fQf1edgzRjhQ4whcAhv7WRRF+qTbfQJ/vDxy8lkiOsvU9XzUaZ0fZSfXX
wXxZamQbONvFcxVHY/0PSiM8nQoUmkkBQuKleDwRWvkoJKYKyr3jvXK7HIWtMrO4
jmXe0aMy3thyK6g5sJVg
-----END CERTIFICATE-----
```

아시아 태평양(싱가포르) - ap-southeast-1

DSA

```
-----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-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUSqP6ih+++5KF07NXngrWf26mhSUwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE0MzAxNFoXDTI5MDQyODE0MzAxNFowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUSqP6ih++
+5KF07NXngrWf26mhSUwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQAw13BxW11U/JL58j//Fmk7qqtrZTqXmaz1qm2WlIpJpW750MOcP4ux1uPy
eM0RdVZ4jHSMv5gtLAv/PjExBfw9n6vNCk+5GZG4Xec5DoapBZHXmfMo93sjxBFP
4x9rWn0GuwAVO9ukjYPevq2RerilrqSVvppHtbATVNY2qecXDA==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAJVMGw5SHkcvMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTEwMjkw
ODU3MTlaGA8yMTk1MDQwMzA4NTcxOVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAlaSSLfBl7OgmikjLReHuNhVuvM20dCsVzptUyRbut+KmIEEc24wd/xVy
2RMIrydGedkW4tUjkUyOyfET5OAyT43jTzDPHZTkRSVkYjBdcYbe9o/0Q4P7IVS3
XlvwrUu0qo9nSID0mxMnOoF1l8KAqnn10tQ0W+lNSTkasW7QVzcb+3okPEVhPAOq
MnlY3vkMQGI8zX4iOKbEcSVIzf6wuIffXMGHVC/JjwihJ2USQ8fq6oy686g54P4w
ROg415kLYcodjqThmGJPNUpAZ7MOc5Z4pymFuCHgNAZNvjhZDA842Ojecqm62zcm
Tzh/pNMNeGCRYq2EQX0aQtYOIj7bOQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU6SSB+3qALorPMVNjToM1Bj3oJMswgY4GA1UdIwSBhjCBg4AU6SSB
+3qALorPMVNjToM1Bj3oJMuhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAJVMGw5SHkcvMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAF/0dWqkIEZKg5rca8o0P0VS+tolJJE/FRZO
atHOeaQbWzyac6NEwjYeeV2kY63skJ+QPuYbSuIBLM8p/uTRIvYM4LZYImLGUvoO
```

```
IdtJ8mAzq8CZ3ipdMs1hRqF5GRp8lg4w2QpX+PfhnW47iIOBiqSAUkIr3Y3BDaDn
EjeXF6qS4iPIvBaQQ0cvdddNh/pE33/ceghbkZNTYkrwMyBkQlRTTVKXFN7pCRUV
+L9FuQ9y8mP0BYZa5e1sdkwebydU+eqVzsil98ntkhpjvRkaJ5+Drs8TjGaJWlRw
5WuOr8unKj7YxdL1bv7//RtVYVVi296ldoRUYv4SCvJF11z0OdQ=
-----END CERTIFICATE-----
```

아시아 태평양(시드니) - ap-southeast-2

DSA

```
-----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-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUFxWyAdk4oiXIOC9PxcgjYYh71mwwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE1MjE0M1oXDTI5MDQyODE1MjE0M1owXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
```

```
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUFxWyAdk4
oiXIOC9PxcgjYYh71mwwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQByjeQe6lr7fiIhoGdjBXYzDfkX0lGGvMIhRh57G1bbceQfaYdZd7Ptc0jl
bpycKGaTvhUdkpMOiV2Hi9dOOYawkdhyJDstmDNKu6P9+b6Kak8He5z3NU1tUR2Y
uTwcz7Ye8Nldx//ws3raErfTI7D6s9m63OX8cAJ/f8bNgikwpw==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAL2bOgb+dq9rMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTEwMjkw
OTAwNTdaGA8yMTk1MDQwMzA5MDA1N1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAmRcyLWraysQS8yDC1b5Abs3TUaJabjqWu7d5gHik5Icd6dKl8EYpQSeS
vz6pLhkgO4xBbCRGlgE8LS/OijcZ5HwdrxBiKbicR1YvIPaIyEQQvF5sX6UWkGYw
Ma5IRGj4YbRmJkBybw+AAV9Icb5LJNOMWPi34OWM+2tMh+8L234v/JA6ogpdPuDr
sM6YFHMZ0NWo58MQ0FnEj2D7H58Ti//vFPl0TaaPWaAIRF85zBiJtKcFJ6vPidqK
f2/SDuAvZmyHC8ZBHg1moX9bR5FsU3QazfbW+c+JzAQWHj2AaQrGSCITxCMlS9sJ
l51DeoZBjnx8cnRe+HCaC4YoRBiqIQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU/wHIo+r5U31VIsPoWoRVsNXGxowwgY4GA1UdIwSBhjCBg4AU/wHI
o+r5U31VIsPoWoRVsNXGxoyhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAL2bOgb+dq9rMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBACobLvj8IxlQyORTz/9q7/VJL509/p4HAeve
92riHp6+Moi0/dSEYPeFTgdWB9W3YCNc34Ss9TJq2D7t/zLGGlbI4wYXU6VJjL0S
hCjWeIyBXUZOZKFCb0DSJeUElsTRSXSFuVrZ9EAwjLvHni3BaC9Ve34iP71ifr75
8Tpk6PEj0+JwiijFH8E4GhcV5chB0/iooU6ioQqJrMwFYnwo1cVZJD5v6D0mu9bS
TMIJLJKv4QQQqPsNdjiB7G9bfkB6trP8fUVYLHLsVlIy5lGx+tgwFEYkG1N8IOO/
2LCawwaWm8FYAFd3IZl04RImNs/IMG7VmH1bf4swHOBHgCN1uYo=
-----END CERTIFICATE-----
```

아시아 태평양(타이베이) - ap-east-2

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7jCCAq4CCQDrb+HArMxssTAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
```

```
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDA1MjQxNDU2MDZaGA8y
MDUwMDUyNTE0NTYwNlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0
b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIg
U2VydmljZXMgTExDMIIBtjCCASsGByqGSM44BAEwggEeAoGBAPFb5R5TBUs2N0YW
aJcOjxr4UCDrpVT3x6EeqoGFNAc1yiZw9iv++ff6wsXm0VG5XV1QkWMQcjPfK/2I
TvXh9XEOhsG7vaBHoWkb2gmIxE6C2u8TfoZu0ZmiMceFK7aYz5w/q2J9CmRs38dT
I9qgTq0LYX+VkiNYOQnn7sQXQ8CjAhUAp2Lue9SfYTmubbpH+3j91Sivc+cCgYAi
DiyGU5R5VgiNV3mhuAyiHCft3LEmTIxos/hx/rHzgm+vCMH55tYejIrv9n7YGrIZ
MejOt8bGYxqSf0Ir/aoj3ZfHihxUzwyfyzwH7D8kAodXJNsPUTKi0O0pGl/e+nVD
JhTIFV3AwC8Tt0r2hljbjhOiyhdYy5hxWHa/WokLAQOBhAACgYAkGGHXbFr/JXol
itpS1qfiETWSRkXGHrdSmUkRumgti2y7c57TgXPwCyNlP1sTr1duy/kYDpwSwyqi
qEYfko0b/xWpDEQQPsWBgolpEYP7RIT85IxrTCrC9ohz85n7v6m9oF6BiBQGtZdk
GBN7iqemg85UaF6mXfHvl3tUYjsB2jAJBgcqhkjOOAQDAy8AMCwCFED9Ybn6fEVF
wLNjDvULmki5bPBWAhR64pxsHFRHz0ZQi+SYMWdIpEs5RQ==
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICNjCCAZ+gAwIBAgIJAJp99iy2CTXhMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDA1MjQx
NDU2MDZaGA8yMjAzMTAyOTE0NTYwNlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQDVdclgpcIIsyGG51ag8srK1UMfZ+Ioh9E4PLP+cIWQcdh31fv2kV/whKaqfmif
NefBIuHrTPl6L+mJQbsfeA+TiEx1Jamg4lwQfhgQKQCmjZVMaHtK+AY57TVN6FBf
HXxT2MXzktsKxAQRGeSTFQYrHRzSYvu+GkA+37lqyWRfcwIDAQABMA0GCSqGSIb3
DQEBCwUAA4GBABM+WL9MSb7z3bNOTMkM7ufPucvj+LQ+zEAJiuG0rcB8GI2DhGB5
8ohDPUGlq1cP7vYmDkWG/FrLqx7qgIzkbE3zfb62JpdOuX73oHgCmikMfq8ifoS1
CUxMQ36PDZBFNPzGZxv5kEORqhmGBkYCoIUCQ43jh3aALR/A/rfzfij8
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAJt5hgMuWbpMMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDA1MjQx
NDU2MDZaGA8yMjAzMTAyOTE0NTYwNlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
```

```
CgKCAQEA91FXs9Wk9kMI3lkNhjC69C6RTBpSvdLWSohPDlJKXFoaGYIDtyW1ViwO
FORAKz0koarVBx1D4fNKCxf+mhYAf//4Pxt31GzjehHNs2YVGimwhq/SNr+3AuIO
PibA+qScXamaiknE1Pxb4jisbBu2PMAG9OxU2dESNwiqBMvdJvg30YNTUITZ6r0e
MPjR1Y/EFVx/SLHW6wdvwqhNi9hz8HHdThzjCbCQPvmoLBJIKXIfosKqvauSezt1
KJSoS2jMx1OjJB2bBlQTZC8vF0WFNPoiH4jXyJ6+wMUMNSSd5b3uL5stVTMzuIz4
39Hd5YW41rmAy+6jmpGp2eXn6Td0rQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQAs
gcgv+Xr67Cj08jKdAUT4GuCoGxveOFAjSDjs5y0cD+kXIT/Yj/+HKFOx5cwThT8F
TxNQ/oVuo3bGeuFEX799H0rT9Jupy0WiE1Vx81mk2M/1EDtns437edM4ofiWR8tR
Q1VNiRgz6Crr0nFgNoQa9HmR6/M+bImxevHJkWB3skvnxtA+lpKGwJs8jo6fA7AW
jr00opK0r731iBd1z/5hjIjkHn5NizAdiJEh3lVY/A3WS82aygW8LvYnKZeaBfZJ
J4NgxeflxocLFKBJD+hyf5u1hhvl7Mn/aGr5VI8/0F2oUqYo+xsHLT3keNghywO4
NNcyLcD9u16HVXhRhGvo
-----END CERTIFICATE-----
```

아시아 태평양(태국) - ap-southeast-7

DSA

```
-----BEGIN CERTIFICATE-----
MIIC8TCCAq8CCQCOEEMWiJIJpTAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDA0MTIxNTI4NTZaGA8y
MDUwMDQxMzE1Mjg1NlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0
b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIg
U2VydmljZXMgTExDMIIBtzCCASsGByqGSM44BAEwggEeAoGBAJWKfrIlz9+/U4wm
NDJPX00avUTdksBKX5RwHkY4o0///2G2HWwiBbxw8F1U+RzCXr80gSoe4+/SCSg3
uDuhJyzN5r4jr7c590CAgsiGdWERs7l4gLQCP504feU4TEjJoq9A8MfRZeqMj7Ug
fT5InTUW7S1/r98ddG/rBHMsJwEbAhUAneIckZ1YeyGcCeL32lujMhj+g68CgYA+
RsyTXCy3Tug5aHun51IfcG2d+pn5K/tv/N3WUR18RplVctrLhwIOUoAsDWPWtxNV
sODetezAyo759CK43JAmYGzXKbRFUHm3n46jP88tSdhuSeJhc/D4l5/0+2L7ndXp
L3+W4N05NdiKSGI8e8t452wTZv/RODivPZ3RtuCDywOBhQACgYEAjVS1Huwzdn1J
+kpd2Rcbe1BAGkmvk5sUu0KhyttqIB1kxTxeWZgsdO8REZSC5gJOwkcGFvXnb3DY
+Ms45r0e0s0rx2FFYjqMLwyRpK9wUjJfSXeJMa9iLQEXuyzBz6zPgfemXbS3zNq/
eoJ9ztIwjB9DMoKL+E1vSLsTGhehqRowCQYHKoZIzjgEAwMxADAuAhUAmljDM3c9
hf0j4XbMjjpnzrx1xhkCFQCASR9pgFNgLK8y6Kojj+P1KJkrSQ==
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICNjCCAZ+gAwIBAgIJAIuIHAhL0xWcMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
```

```
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDA0MTIx
NTI4NTZaGA8yMjAzMDkxNzE1Mjg1NlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQCajgAeOauwvqGDLrHvxujnZ1BnkMzwjrycMUTkj8jqNtWoDQWUJVNPZJILosEU
VwK2I3oNkEsx/ryl9XfXcNNceoYfVEPzkTzozrZyuOG66FWtUU1LKeJ7h9/rX0Zd
9lZEokrdr6dLPt9FsHWaK5ExlUnWBjNltcQLkkKqoeYaFwIDAQABMA0GCSqGSIb3
DQEBCwUAA4GBAE4G5G+FvKTsX3T7BEcId7f5LSCc2J9gZRDiWn2oTr40CrBMOzJT
KsWr9W89YXW3gaGWltzc0WCwYQbJZgAkuEAZItJjbhdnns87ZbsFO+NZhc6gDtjA
WC3dPlSB9b6rfVoVW9O6Xwa7iNXZo8ddYVJ/ZOIv/totUz9qJt4DmmKk
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJANaQIrYcijxaMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDA0MTIx
NTI4NTZaGA8yMjAzMDkxNzE1Mjg1NlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAxtMGK4r6WerVtMfpRRCY3DMP9Q/s4jnRIqolPaxeT99BAFp03HXyOrTz
WjggEHArRGGeDowFkL7Vj+QNY7ran2lnYVlAQ7Ow6OLl6WttT6lMsgsLUIcsED6G
eJ1Ko+ovT1qvmuush1U5R2pHkcCJ1IT2s6uQff85l066K7dCFpaoA1QHUk9zACHj
i0DtWuTAWBXirAZSGlmtP6uj1Aw6y5kACDTjIvZJeW/kmswfApTKaJ56eNkJsUBU
NpKwt3um1BMZduy0cjUv0Sc3dIbLNyfOdyPn8tX5u5ck0EPdtBB5WjEh71IXdZJD
oaBREV7AHg5ERPQsm0BqSvMQt09KiwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQDG
qJyZbv/GUQnm8dEyVYgilQr3Yu35n8sfTwwZ1OJzIVce7NBPWePMfYii/tvGoKNp
KwFa/Vugbn1EwqpTEKqkPE1dOZvnUa97XxhYkoW5U7sgdKCANi1KWjywn7MiJ3Eg
j4gdqYK8wxvHi21ppqr572UO77uA8YMBoBT/CyQzWYUSmbqwKnnzaQBAZeO2iAk
VHuNU+9UsntNB676gRl9ag3Wfxq3yx5Ee1CeQf+US3HJn/pKk1H8dExXmBHvHwO6
GKUVNPN1rxfjTiaSt8wuO80uElAnyHIM3VOR8rJO7PKsobyEeJV0WI1hURO+wxpL
h3IsW7iBrFVvhX5xx7ZU
-----END CERTIFICATE-----
```

아시아 태평양(도쿄) - ap-northeast-1

DSA

```
-----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-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIULgwDh7TiDrPPBJwscqDwiBHkEFQwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTEyMjMxMFoXDTI5MDQyODEyMjMxMFowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IULgwDh7Ti
DrPPBJwscqDwiBHkEFQwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQBtjAglBde1t4F9EHCZOj4qnY6Gigy07Ou54i+lR77MhbpzE8V28Li9l+YT
QMIn6SzJqU3/fIycIro1OVY1lHmaKYgPGSEZxBenSBHfzwDLRmC9oRp4QMe0BjOC
gepj1lUoiN7OA6PtA+ycNlsP0oJvdBjhvayLiuM3tUfLTrgHbw==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAL9KIB7Fgvg/MA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
```

```
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA4MTQw
OTAwMjVaGA8yMTk1MDExNzA5MDAyNVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAz0djWUcmRW85C5CiCKPFiTIvj6y2OuopFxNE5d3Wtab10bm06vnXVKXu
tz3AndG+Dg0zIL0gMlU+QmrSR0PH2PfV9iejfLak9iwdm1WbwRrCEAj5VxPe0Q+I
KeznOtxzqQ5Wo5NLE9bA61sziUAFNVsTFUzphEwRohcekYyd3bBC4v/RuAjCXHVx
40z6AIksnAOGN2VABMlTeMNvPItKOCIeRLlllSqXX1gbtL1gxSW40JWdF3WPB68E
e+/1U3F7OEr7XqmNODOL6yh92QqZ8fHjG+afOL9Y2Hc4g+P1nk4w4iohQOPABqzb
MPjK7B2Rze0f9OEc51GBQu13kxkWWQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU5DS5IFdU/QwYbikgtWvkU3fDwRgwgY4GA1UdIwSBhjCBg4AU5DS5
IFdU/QwYbikgtWvkU3fDwRihYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAL9KIB7Fgvg/MBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAG/N7ua8IE9IMyno0n5T57erBvLTOQ79fIJN
Mf+mKRM7qRRsdg/eumFft0rLOKo54pJ+Kim2cngCWNhkzctRHBV567AJNt4+ZDG5
hDgV0IxWO1+eaLE4qzqWP/9VrO+p3reuumgFZLVpvVpwXBBeBFUf2drUR14aWfI2
L/6VGINXYs7uP8v/2VBS7r6XZRnPBUy/R4hv5efYXnjwA9gq8+a3stC2ur8m5ySl
faKSwE4H320yAyaZWH4gpwUdbUlYgPHtm/ohRtiWPrN7KEG5Wq/REzMIjZCnxOfS
6KR6PNjlhxBsImQhmBvz6j5PLQxOxBZIpDoiK278e/1Wqm9LrBc=
-----END CERTIFICATE-----
```

캐나다(중부) - ca-central-1

DSA

```
-----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-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUIrLgixJJB5C4G8z6pZ5rB0JU2aQwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE1MzU0M1oXDTI5MDQyODE1MzU0M1owXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUIrLgixJJ
B5C4G8z6pZ5rB0JU2aQwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQBHiQJmzyFAaSYs8SpiRijIDZW2RIo7qBKb/pI3rqK6yOWDlPuMr6yNI81D
IrKGGftg4Z+2KETYU4x76HSf0s//vfH3QA57qFaAwddhKYy4BhteFQl/Wex3xTlX
LiwI07kwJvJy3mS6UfQ4HcvZy219tY+0iyOWrz/jVxwq7TOkCw==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAJNKhJhaJOuMMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNjA3Mjkx
MTM3MTdaGA8yMTk2MDEwMjExMzcxN1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAhDUh6j1ACSt057nSxAcwMaGr8Ez87VA2RW2HyY8l9XoHndnxmP50Cqld
+26AJtltlqHpI1YdtnZ6OrVgVhXcVtbvte0lZ3ldEzC3PMvmISBhHs6A3SWHA9ln
InHbToLX/SWqBHLOX78HkPRaG2k0COHpRy+fG9gvz8HCiQaXCbWNFDHZev9OToNI
xhXBVzIa3AgUnGMalCYZuh5AfVRCEeALG60kxMMC8IoAN7+HG+pMdqAhJxGUcMO0
LBvmTGGeWhi04MUZWfOkwn9JjQZuyLg6B1OD4Y6s0LB2P1MovmSJKGY4JcF8Qu3z
xxUbl7Bh9pvzFR5gJN1pjM2n3gJEPwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQAJ
UNKM+gIIHNk0G0tzv6vZBT+o/vt+tIp8lEoZwaPQh1121iw/I7ZvhMLAigx7eyvf
IxUt9/nf8pxWaeGzi98RbSmbap+uxYRynqe1p5rifTamOsguuPrhVpl12OgRWLcT
rjg/K60UMXRsmg2w/cxV45pUBcyVb5h6Op5uEVAVq+CVns13ExiQL6kk3guG4+Yq
```

```
LvP1p4DZfeC33a2Rfre2IHLsJH5D4SdWcYqBsfTpf3FQThH0l0KoacGrXtsedsxs
9aRd7OzuSEJ+mBxmzxSjSwM84Ooh78DjkdpQgv967p3d+8NiSLt3/n7MgnUy6WwB
KtDujDnB+ttEHwRRngX7
-----END CERTIFICATE-----
```

캐나다 서부(캘거리) - ca-west-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq
+gAwIBAgIGAYPouptUMAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxED
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYQAAoGAMITzTJUa6cBsIfdHN69zW/
aHjUB4r1ZfKb1FMhIp9EZtEf5n+06oXjUG2+dKRSlFQeEK333ehNZsPd6uqey6TYKtHpFb5XRLS8BpqB
+7gnbAd0CBZM5o4NWesSQ1GLnTdQcGZkYG/
QESkbadoCXQTifCujJE682hTDLIVt1d4ewwCQYHKoZIzjgEAwMvADAsAhRJc4gRS/HWTkCR2MESaQEe/
jOMNQIUNoTwLvuPrmGPupPlGiHe0veZiO8=
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICMzCCAZygAwIBAgIGAYPou9weMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIDBBXYXNoaW5n
v4XBVH13ZCMgqlRHMqV8AWI5iO6gFn2A9sN3AZXTMqwtZeiDdebq3k6Wt7ieYvpXTg0qvgsjQIovRZWaBDBJy9x8C2hW
+w9lMQjFHkJ7Jy/
PHCJ69EzebQIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAGe9Snkz1A6rHBH6/5kDtYvtPYwhx2sXNxztbhkXErFk40Nw5l4
gvDVtWG7qyb6fAqgoisyAbk8K9LzxSim2S1nmT9vD84B/t/VvwQBylc
+ej8kRxMH7fquZLp7IXfmtBzyUqu6Dpbne+chG2
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJALyTn5IHrIZjMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMzEyMDcx
NTM3MDFaGA8yMjAzMDUxMzE1MzcwMVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
```

```
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA1GP5os424BJmGPCKOSgOclP7lzUiB85du03M4hfjzSOszsBpmBGFDLz1
owYHtIxlq3+Vi1Lt5Qlx3id/ov1QyaBPFWXVeklHVXy9vieCcI3TdjGjTl1W/8MM
m3X26QPcsnHM/Kk2wJ7s186MrqmdSsp3SCPpxv4vEG2Q9yR2bXY41hpc2rWlW8qU
D0JGXluvmmAdFnto20l1XWZ6xFen1h60DRugek/ufCbN+ljKy0xLqPoavH0Ybjsb
UpsAsBs7phaoN+X/5hIERfbp5LfVnqq54pNG5KNu4KynfW9+kA/WS4cJ6FTTN5t+
y0P1HvcL+BL2RuDy6T2bB21xw5WqtQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQURTvu/Dd4zDnmS5G5CfVLnmUBNOswgY4GA1UdIwSBhjCBg4AURTvu
/Dd4zDnmS5G5CfVLnmUBNOuhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJALyTn5IHrIZjMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAFt523A3Aug6/F8xxyITgA8gkU0btFhlXNSP
U4U2OQ9n0tWI9WqnKNWH3KBxwY5EPitU6b3LM4xc9lDWpz7h2Pto+WhxP9LVKe6f
r8r7teTLCVZ7cfYZHzHg+flZjVpAgzE5BVfrRlj3QKpv0hYT3J1wMtI++Vorq5Nf
aPjzedehJLhmZVALwnfqfLrgv6/gmraP9Vmoa8U4D6AljNiQGYaLwyoPoRm3bUs2
v1Mh9GkEQ1b9+1pFXcqgzJJTGRuiPCyPbECI79FAnx5JM/CkGJV8HlOmjIW1qkK1
Y2qT7wzErrKLJyB53Pw15BdIM1onbDAQreZb0yZQLdoEl/tx7Uk=
-----END CERTIFICATE-----
```

중국(베이징) - cn-north-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIDNjCCAh4CCQD3yZ1w1AVkTzANBgkqhkiG9w0BAQsFADBcMQswCQYDVQQGEwJV
UzEZMBcGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEg
MB4GA1UEChMXQW1hem9uIFdlYiBTZXJ2aWNlcyBMTEMwIBcNMTUwNTEzMDk1OTE1
WhgPMjE5NDEwMTYwOTU5MTVaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNo
aW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24g
V2ViIFNlcnZpY2VzIExMQzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
AMWk9vyppSmDU3AxZ2Cy2bvKeK3F1UqNpMuyeriizi+NTsZ8tQqtNloaQcqhto/l
gsw9+QSnEJeYWnmivJWOBdn9CyDpN7cpHVmeGgNJL2fvImWyWe2f2Kq/BL9l7N7C
P2ZT52/sH9orlck1n2zO8xPi7MItgPHQwu3OxsGQsAdWucdxjHGtdchulpo1uJ31
jsTAPKZ3p1/sxPXBBAgBMatPHhRBqhwHO/Twm4J3GmTLWN7oVDds4W3bPKQfnw3r
vtBj/SM4/IgQ3xJslFcl90TZbQbgxIi88R/gWTbs7GsyT2PzstU30yLdJhKfdZKz
/aIzraHvoDTWFaOdy0+OOaECAwEAATANBgkqhkiG9w0BAQsFAAOCAQEAdSzN2+0E
V1BfR3DPWJHWRf1b7zl+1X/ZseW2hYE5r6YxrLv+1VPf/L5I6kB7GEtqhZUqteY7
zAceoLrVu/7OynRyfQetJVGichaaxLNM3lcr6kcxOowb+WQQ84cwrB3keykH4gRX
KHB2rlWSxta+2panSEO1JX2q5jhcFP90rDOtZjlpYv57N/Z9iQ+dvQPJnChdq3BK
5pZlnIDnVVxqRike7BFy8tKyPj7HzoPEF5mh9Kfnn1YoSVu+61lMVv/qRjnyKfS9
c96nE98sYFj0ZVBzXw8Sq4Gh8FiVmFHbQp1peGC19idOUqxPxWsasWxQXO0azYsP
```

```
9RyWLHKxH1dMuA==
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDCzCCAnSgAwIBAgIJALSOMbOoU2svMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0yMzA3MDQw
ODM1MzlaFw0yODA3MDIwODM1MzlaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA
uhhUNlqAZdcWWB/OSDVDGk3OA99EFzOn/mJlmciQ/Xwu2dFJWmSCqEAE6gjufCjQ
q3voxAhC2CF+elKtJW/C0Sz/LYo60PUqd6iXF4h+upB9HkOOGuWHXsHBTsvgkgGA
1CGgel4U0Cdq+23eANr8N8m28UzljjSnTlrYCHtzN4sCAwEAAaOB1DCB0TALBgNV
HQ8EBAMCB4AwHQYDVR0OBBYEFBkZu3wT27NnYgrfH+xJz4HJaNJoMIGOBgNVHSME
gYYwgYOAFBkZu3wT27NnYgrfH+xJz4HJaNJooWCkXjBcMQswCQYDVQQGEwJVUzEZ
MBcGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEgMB4G
A1UEChMXQW1hem9uIFdlYiBTZXJ2aWNlcyBMTEOCCQC0jjGzqFNrLzASBgNVHRMB
Af8ECDAGAQH/AgEAMA0GCSqGSIb3DQEBCwUAA4GBAECji43p+oPkYqmzll7e8Hgb
oADS0ph+YUz5P/bUCm61wFjlxaTfwKcuTR3ytj7bFLoW5Bm7Sa+TCl3lOGb2taon
2h+9NirRK6JYk87LMNvbS40HGPFumJL2NzEsGUeK+MRiWu+Oh5/lJGii3qw4YByx
SUDlRyNy1jJFstEZjOhs
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAOtrM5XLDSjCMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA4MTQx
MDAxNDJaGA8yMTk1MDExNzEwMDE0MlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAvVBz+WQNdPiM9S+aUULOQEriTmNDUrjLWLr7SfaOJScBzis5D5ju0jh1
+qJdkbuGKtFX5OTWTm8pWhInX+hIOoS3exC4BaANoa1A3o6quoG+Rsv72qQf8LLH
sgEi6+LMlCN9TwnRKOToEabmDKorss4zFl7VSsbQJwcBSfOcIwbdRRaW9Ab6uJHu
79L+mBR3Ea+G7vSDrVIA8goAPkae6jY9WGw9KxsOrcvNdQoEkqRVtHo4bs9fMRHU
Etphj2gh4ObXlFN92VtvzD6QBs3CcoFWgyWGvzg+dNG5VCbsiiuRdmii3kcijZ3H
Nv1wCcZoEAqH72etVhsuvNRC/xAP8wIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQA8
ezx5LRjzUU9EYWYhyYIEShFlP1qDHs7F4L46/5lc4pL8FPoQm5CZuAF31DJhYi/b
fcV7i3n++/ymQbCLC6kAg8DUB7NrcROll5ag8d/JXGzcTCnlDXLXx1905fPNa+jI
0q5quTmdmiSi0taeaKZmyUdhrB+a7ohWdSdlokEIOtbH1P+g5yll3bI2leYE6Tm8
```

```
LKbyfK/532xJPqO9abx4Ddn89ZEC6vvWVNDgTsxERg992Wi+/xoSw3XxkgAryIv1
zQ4dQ6irFmXwCWJqc6kHg/M5W+z60S/94+wGTXmp+19U6Rkq5jVMLh16XJXrXwHe
4KcgIS/aQGVgjM6wivVA
-----END CERTIFICATE-----
```

중국(닝샤) - cn-northwest-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIDNjCCAh4CCQD3yZ1w1AVkTzANBgkqhkiG9w0BAQsFADBcMQswCQYDVQQGEwJV
UzEZMBcGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEg
MB4GA1UEChMXQW1hem9uIFdlYiBTZXJ2aWNlcyBMTEMwIBcNMTUwNTEzMDk1OTE1
WhgPMjE5NDEwMTYwOTU5MTVaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNo
aW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24g
V2ViIFNlcnZpY2VzIExMQzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
AMWk9vyppSmDU3AxZ2Cy2bvKeK3F1UqNpMuyeriizi+NTsZ8tQqtNloaQcqhto/l
gsw9+QSnEJeYWnmivJWOBdn9CyDpN7cpHVmeGgNJL2fvImWyWe2f2Kq/BL9l7N7C
P2ZT52/sH9orlck1n2zO8xPi7MItgPHQwu3OxsGQsAdWucdxjHGtdchulpo1uJ31
jsTAPKZ3p1/sxPXBBAgBMatPHhRBqhwHO/Twm4J3GmTLWN7oVDds4W3bPKQfnw3r
vtBj/SM4/IgQ3xJslFcl90TZbQbgxIi88R/gWTbs7GsyT2PzstU30yLdJhKfdZKz
/aIzraHvoDTWFaOdy0+OOaECAwEAATANBgkqhkiG9w0BAQsFAAOCAQEAdSzN2+0E
V1BfR3DPWJHWRf1b7zl+1X/ZseW2hYE5r6YxrLv+1VPf/L5I6kB7GEtqhZUqteY7
zAceoLrVu/7OynRyfQetJVGichaaxLNM3lcr6kcxOowb+WQQ84cwrB3keykH4gRX
KHB2rlWSxta+2panSEO1JX2q5jhcFP90rDOtZjlpYv57N/Z9iQ+dvQPJnChdq3BK
5pZlnIDnVVxqRike7BFy8tKyPj7HzoPEF5mh9Kfnn1YoSVu+61lMVv/qRjnyKfS9
c96nE98sYFj0ZVBzXw8Sq4Gh8FiVmFHbQp1peGC19idOUqxPxWsasWxQXO0azYsP
9RyWLHKxH1dMuA==
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDCzCCAnSgAwIBAgIJALSOMbOoU2svMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0yMzA3MDQw
ODM1MzlaFw0yODA3MDIwODM1MzlaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA
uhhUNlqAZdcWWB/OSDVDGk3OA99EFzOn/mJlmciQ/Xwu2dFJWmSCqEAE6gjufCjQ
q3voxAhC2CF+elKtJW/C0Sz/LYo60PUqd6iXF4h+upB9HkOOGuWHXsHBTsvgkgGA
1CGgel4U0Cdq+23eANr8N8m28UzljjSnTlrYCHtzN4sCAwEAAaOB1DCB0TALBgNV
```

```
HQ8EBAMCB4AwHQYDVR0OBBYEFBkZu3wT27NnYgrfH+xJz4HJaNJoMIGOBgNVHSME
gYYwgYOAFBkZu3wT27NnYgrfH+xJz4HJaNJooWCkXjBcMQswCQYDVQQGEwJVUzEZ
MBcGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEgMB4G
A1UEChMXQW1hem9uIFdlYiBTZXJ2aWNlcyBMTEOCCQC0jjGzqFNrLzASBgNVHRMB
Af8ECDAGAQH/AgEAMA0GCSqGSIb3DQEBCwUAA4GBAECji43p+oPkYqmzll7e8Hgb
oADS0ph+YUz5P/bUCm61wFjlxaTfwKcuTR3ytj7bFLoW5Bm7Sa+TCl3lOGb2taon
2h+9NirRK6JYk87LMNvbS40HGPFumJL2NzEsGUeK+MRiWu+Oh5/lJGii3qw4YByx
SUDlRyNy1jJFstEZjOhs
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAPu4ssY3BlzcMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTEyMDMy
MTI5MzJaGA8yMTk1MDUwODIxMjkzMlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAsOiGi4A6+YTLzCdIyP8b8SCT2M/6PGKwzKJ5XbSBoL3gsnSWiFYqPg9c
uJPNbiy9wSA9vlyfWMd90qvTfiNrT6vewP813QdJ3EENZOx4ERcf/Wd22tV72kxD
yw1Q3I1OMH4bOItGQAxU5OtXCjBZEEUZooOkU8RoUQOU2Pql4NTiUpzWacNutAn5
HHS7MDc4lUlsJqbN+5QW6fFrcNG/0Mrib3JbwdFUNhrQ5j+Yq5h78HarnUivnX/3
Ap+oPbentv1qd7wvPJu556LZuhfqI0TohiIT1Ah+yUdN5osoaMxTHKKtf/CsSJ1F
w3qXqFJQA0VWsqjFyHXFI32I/GOupwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQCn
Um00QHvUsJSN6KATbghowLynHn3wZSQsuS8E0COpcFJFxP2SV0NYkERbXu0n/Vhi
yq5F8v4/bRA2/xpedLWmvFs7QWlomuXhSnYFkd33Z5gnXPb9vRkLwiMSw4uXls35
qQraczUJ9EXDhrv7VmngIk9H3YsxYrlDGEqh/oz4Ze4ULOgnfkauanHikk+BUEsg
/jsTD+7e+niEzJPihHdsvKFDlud5pakEzyxovHwNJ1GS2I//yxrJFIL91mehjqEk
RLPdNse7N6UvSnuXcOokwu6l6kfzigGkJBxkcq4gre3szZFdCQcUioj7Z4xtuTL8
YMqfiDtN5cbD8R8ojw9Y
-----END CERTIFICATE-----
```

유럽(프랑크푸르트) - eu-central-1

DSA

```
-----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-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUFD5GsmkxRuecttwsCG763m3u63UwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE1NTUyOVoXDTI5MDQyODE1NTUyOVowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUFD5Gsmkx
RuecttwsCG763m3u63UwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQBBh0WaXlBsW56Hqk588MmJxsOrvcKfDjF57RgEDgnGnQaJcStCVWDO9UYO
JX2tdsPw+E7AjDqjsuxYaotLn3Mr3mK0sNOXq9BljBnWD4pARg89KZnZI8FN35HQ
O/LYOVHCknuPL123VmVRNs51qQA9hkPjvw21UzpDLxaUxt9Z/w==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAKD+v6LeR/WrMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA4MTQw
OTA4MTlaGA8yMTk1MDExNzA5MDgxOVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
```

```
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAka8FLhxs1cSJGK+Q+q/vTf8zVnDAPZ3U6oqppOW/cupCtpwMAQcky8DY
Yb62GF7+C6usniaq/9W6xPn/3o//wti0cNt6MLsiUeHqNl5H/4U/Q/fR+GA8pJ+L
npqZDG2tFi1WMvvGhGgIbScrjR4VO3TuKy+rZXMYvMRk1RXZ9gPhk6evFnviwHsE
jV5AEjxLz3duD+u/SjPp1vloxe2KuWnyC+EKInnka909sl4ZAUh+qIYfZK85DAjm
GJP4W036E9wTJQF2hZJrzsiB1MGyC1WI9veRISd30izZZL6VVXLXUtHwVHnVASrS
zZDVpzj+3yD5hRXsvFigGhY0FCVFnwIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUxC2l6pvJaRflgu3MUdN6zTuP6YcwgY4GA1UdIwSBhjCBg4AUxC2l
6pvJaRflgu3MUdN6zTuP6YehYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAKD+v6LeR/WrMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAIK+DtbUPppJXFqQMv1f2Gky5/82ZwgbbfXa
HBeGSii55b3tsyC3ZW5ZlMJ7Dtnr3vUkiWbV1EUaZGOUlndUFtXUMABCb/coDndw
CAr53XTv7UwGVNe/AFO/6pQDdPxXn3xBhF0mTKPrOGdvYmjZUtQMSVb9lbMWCFfs
w+SwDLnm5NF4yZchIcTs2fdpoyZpOHDXy0xgxO1gWhKTnYbaZOxkJvEvcckxVAwJ
obF8NyJla0/pWdjhlHafEXEN8lyxyTTyOa0BGTuYOBD2cTYYynauVKY4fqHUkr3v
Z6fboaHEd4RFamShM8uvSu6eEFD+qRmvqlcodbpsSOhuGNLzhOQ=
-----END CERTIFICATE-----
```

유럽(아일랜드) - eu-west-1

DSA

```
-----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-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUakDaQ1Zqy87Hy9ESXA1pFC116HkwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE2MTgxMFoXDTI5MDQyODE2MTgxMFowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUakDaQ1Zq
y87Hy9ESXA1pFC116HkwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQADIkn/MqaLGPuK5+prZZ5Ox4bBZLPtreO2C7r0pqU2kPM2lVPyYYydkvP0
lgSmmsErGu/oL9JNztDe2oCA+kNy17ehcsf8cw0uP861czNFKCeU8b7FgBbL+sIm
qi33rAq6owWGi/5uEcfCR+JP7W+oSYVir5r/yDmWzx+BVH5S/g==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAOrmqHuaUt0vMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTEwMjkw
OTA2MTlaGA8yMTk1MDQwMzA5MDYxOVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAjE7nVu+aHLtzp9FYV25Qs1mvJ1JXD7J0iQ1Gs/RirW9a5ZECCtc4ssnf
zQHq2JRVr0GRchvDrbm1HaP/avtFQR/Thvfltwu9AROVT22dUOTvERdkNzveoFCy
hf52Rqf0DMrLXG8ZmQPPXPDFAv+sVMWCDftcChxRYZ6mP9O+TpgYNT1krD5PdvJU
7HcXrkNHDYqbsg8A+Mu2hzl0QkvUET83Csg1ibeK54HP9w+FsD6F5W+6ZSHGJ88l
FI+qYKs7xsjJQYgXWfEt6bbckWs1kZIaIOyMzYdPF6ClYzEec/UhIe/uJyUUNfpT
VIsI5OltBbcPF4c7Y20jOIwwI2SgOQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUF2DgPUZivKQR/Zl8mB/MxIkjZDUwgY4GA1UdIwSBhjCBg4AUF2Dg
PUZivKQR/Zl8mB/MxIkjZDWhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAOrmqHuaUt0vMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAGm6+57W5brzJ3+T8/XsIdLTuiBSe5ALgSqI
qnO5usUKAeQsa+kZIJPyEri5i8LEodh46DAF1RlXTMYgXXxl0YggX88XPmPtok17
```

```
l4hib/D9/lu4IaFIyLzYNSzsETYWKWoGVe7ZFz60MTRTwY2u8YgJ5dec7gQgPSGj
avB0vTIgoW41G58sfw5b+wjXCsh0nROon79RcQFFhGnvup0MZ+JbljyhZUYFzCli
31jPZiKzqWa87xh2DbAyvj2KZrZtTe2LQ48Z4G8wWytJzxEeZdREe4NoETf+Mu5G
4CqoaPR05KWkdNUdGNwXewydb3+agdCgfTs+uAjeXKNdSpbhMYg=
-----END CERTIFICATE-----
```

유럽(런던) - eu-west-2

DSA

```
-----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-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUCgCV/DPxYNND/swDgEKGiC5I+EwwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE2MjkxNFoXDTI5MDQyODE2MjkxNFowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
```

```
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUCgCV/DPx
YNND/swDgEKGiC5I+EwwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQATPu/sOE2esNa4+XPEGKlEJSgqzyBSQLQc+VWo6FAJhGG9fp7D97jhHeLC
5vwfmtTAfnGBxadfAOT3ASkxnOZhXtnRna460LtnNHm7ArCVgXKJo7uBn6ViXtFh
uEEw4y6p9YaLQna+VC8Xtgw6WKq2JXuKzuhuNKSFaGGw9vRcHg==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJANBx0E2bOCEPMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNjA4MTEx
NDU2NDJaGA8yMTk2MDExNTE0NTY0MlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEArYS3mJLGaMrh2DmiPLbqr4Z+xWXTzBWCjOwpsuHE9H6dWUUyl2Bgnu+Z
d8QvW306Yleec45M4F2RA3J4hWHtShzsMlOJVRt+YulGeTf9OCPr26QmIFfs5nD4
fgsJQEry2MBSGA9Fxq3Cw6qkWcrOPsCR+bHOU0XykdKl0MnIbpBf0kTfciAupQEA
dEHnM2J1L2iI0NTLBgKxy5PXLH9weX20BFauNmHH9/J07OpwL20SN5f8TxcM9+pj
Lbk8h1V4KdIwVQpdWkbDL9BCGlYjyadQJxSxz1J343NzrnDM0M4h4HtVaKOS7bQo
Bqt2ruopLRCYgcuFHck/1348iAmbRQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBG
wujwU1Otpi3iBgmhjMClgZyMMn0aQIxMigoFNqXMUNx1Mq/e/Tx+SNaOEAu0n2FF
aiYjvY0/hXOx75ewzZvM7/zJWIdLdsgewpUqOBH4DXFhbSk2TxggSPb0WRqTBxq5
Ed7F7+7GRIeBbRzdLqmISDnfqey8ufW0ks51XcQNomDIRG5s9XZ5KHviDCar8FgL
HngBCdFI04CMagM+pwTO9XN1Ivt+NzUj208ca3oP1IwEAd5KhIhPLcihBQA5/Lpi
h1s3170z1JQ1HZbDrH1pgp+8hSI0DwwDVb3IIH8kPR/J0Qn+hvOl2HOpaUg2Ly0E
pt1RCZe+W7/dF4zsbqwK
-----END CERTIFICATE-----
```

유럽(밀라노) – eu-south-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAqwCCQCMElHPdwG37jAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xOTA0MjkyMDM1MjJaFw00
NTA0MjkyMDM1MjJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
```

```
cnZpY2VzIExMQzCCAbYwggErBgcqhkjOOAQBMIIBHgKBgQDAkoL4YfdMI/MrQ0oL
NPfeEk94eiCQA5xNOnU7+2eVQtEqjFbDADFENh1p3sh9Q9OoheLFH8qpSfNDWn/0
ktCS909ApTY6Esx1ExjGSeQq/U+SC2JSuuTT4WFMKJ63a/czMtFkEPPnVIjJJJmT
HJSKSsVUgpdDIRvJXuyB0zdB+wIVALQ3OLaVGdlPMNfS1nD/Yyn+32wnAoGAPBQ3
7XHg5NLOS4326eFRUT+4ornQFjJjP6dp3pOBEzpImNmZTtkCNNUKE4Go9hv5T4lh
R0pODvWv0CBupMAZVBP9ObplXPCyEIZtuDqVa7ukPOUpQNgQhLLAqkigTyXVOSmt
ECBj9tu5WNP/x3iTZTHJ+g0rhIqpgh012UwJpKADgYQAAoGAV1OEQPYQUg5/M3xf
6vE7jKTxxyFWEyjKfJK7PZCzOIGrE/swgACy4PYQW+AwcUweSlK/Hx2OaZVUKzWo
wDUbeu65DcRdw2rSwCbBTU342sitFo/iGCV/Gjf+BaiAJtxniZze7J1ob8vOBeLv
uaMQmgOYeZ5e0fl04GtqPl+lhcQwCQYHKoZIzjgEAwMwADAtAhQdoeWLrkm0K49+
AeBK+j6m2h9SKQIVAIBNhS2a8cQVABDCQXVXrc0tOmO8
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICNjCCAZ+gAwIBAgIJAOZ3GEIaDcugMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xOTEwMjQx
NTE5MDlaGA8yMTk5MDMyOTE1MTkwOVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQCjiPgW3vsXRj4JoA16WQDyoPc/eh3QBARaApJEc4nPIGoUolpAXcjFhWplo2O+
ivgfCsc4AU9OpYdAPha3spLey/bhHPRi1JZHRNqScKP0hzsCNmKhfnZTIEQCFvsp
DRp4zr91/WS06/flJFBYJ6JHhp0KwM81XQG59lV6kkoW7QIDAQABMA0GCSqGSIb3
DQEBCwUAA4GBAGLLrY3P+HH6C57dYgtJkuGZGT2+rMkk2n81/abzTJvsqRqGRrWv
XRKRXlKdM/dfiuYGokDGxiC0Mg6TYy6wvsR2qRhtXW1OtZkiHWcQCnOttz+8vpew
wx8JGMvowtuKB1iMsbwyRqZkFYLcvH+Opfb/Aayi20/ChQLdI6M2R5VU
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAO/+DgYF78KwMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xOTA0Mjky
MDM1MjJaGA8yMTk4MTAwMjIwMzUyMlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAv1ZLV+Z/P6INq+R1qLkzETBg7sFGKPiwHekbpuB6lrRxKHhj8V9vaReM
lnv1Ur5LAPpMPYDsuJ4WoUbPYAqVqyMAo7ikJHCCM1cXgZJefgN6z9bpS+uA3YVh
V/0ipHh/X2hc2S9wvxKWiSHu6Aq9GVpqL035tJQD+NJuqFd+nXrtcw4yGtmvA6wl
5Bjn8WdsP3xOTKjrByYY1BhXpP/f1ohU9jE9dstsRXLa+XTgTPWcWdCS2oRTWPGR
```

```
c5Aeh47nnDsyQfP9gLxHeYeQItV/BD9kU/2Hn6mnRg/B9/TYH8qzlRTzLapXp4/5
iNwusrTNexGl8BgvAPrfhjDpdgYuTwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQB7
5ya11K/hKgvaRTvZwVV8GlVZt0CGPtNvOi4AR/UN6TMm51BzUB5nurB4z0R2MoYO
Uts9sLGvSFALJ4otoB77hyNpH3drttU1CVVwal/yK/RQLSon/IoUkaGEbqalu+mH
nYad5IG4tEbmepX456XXcO58MKmnczNbPyw3FRzUZQtI/sf94qBwJ1Xo6XbzPKMy
xjL57LHIZCssD+XPifXay69OFlsCIgLim11HgPkRIHEOXLSf3dsW9r+4CjoZqB/Z
jj/P4TLCxbYCLkvglwaMjgEWF40Img0fhx7yT2X92MiSrs3oncv/IqfdVTiN8OXq
jgnq1bf+EZEZKvb6UCQV
-----END CERTIFICATE-----
```

유럽(파리) - eu-west-3

DSA

```
-----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-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUaC9fX57UDr6u1vBvsCsECKBZQyIwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE2MzczOFoXDTI5MDQyODE2MzczOFowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
```

```
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUaC9fX57U
Dr6u1vBvsCsECKBZQyIwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQCARv1bQEDaMEzYI0nPlu8GHcMXgmgA94HyrXhMMcaIlQwocGBs6VILGVhM
TXP2r3JFaPEpmXSQNQHvGA13clKwAZbni8wtzv6qXb4L4muF34iQRHF0nYrEDoK7
mMPR8+oXKKuPO/mv/XKo6XAV5DDERdSYHX5kkA2R9wtvyZjPnQ==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJALWSfgHuT/ARMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNzA1MzEx
MTE4MTZaGA8yMTk2MTEwMzExMTgxNlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAy5V7KDqnEvF3DrSProFcgu/oL+QYD62b1U+Naq8aPuljJe127Sm9WnWA
EBdOSASkOaQ9fzjCPoG5SGgWKxYoZjsevHpmzjVv9+Ci+F57bSuMbjgUbvbRIFUB
bxQojVoXQPHgK5v433ODxkQ4sjRyUbf4YV1AFdfU7zabC698YgPVOExGhXPlTvco
8mlc631ubw2g52j0lzaozUkHPSbknTomhQIvO6kUfX0e0TDMH4jLDG2ZIrUB1L4r
OWKG4KetduFrRZyDHF6ILZu+s6ywiMicUd+2UllDFC6oas+a8D11hmO/rpWU/ieV
jj4rWAFrsebpn+Nhgy96iiVUGS2LuQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQDE
iYv6FQ6knXCg+svlcaQG9q59xUC5z8HvJZ1+SxzPKKC4PKQdKvIIfE8GxVXqlZG1
cl5WKTFDMapnzb9RV/DTaVzWx3cMYT77vm1Hl1XGjhx611CGcENH1egI3lOTILsa
+KfopuJEQQ9TDMAIkGjhA+KieU/U5Ctv9fdej6d0GC6OEuwKkTNzPWue6UMq8d4H
2xqJboWsE1t4nybEosvZfQJcZ8jyIYcYBnsG13vCLM+ixjuU5MVVQNMY/gBJzqJB
V+U0QiGiuT5cYgY/QihxdHt99zwGaE0ZBC7213NKrlNuLSrqhDI2NLu8NsExqOFy
OmY0v/xVmQUQl26jJXaM
-----END CERTIFICATE-----
```

유럽(스페인) - eu-south-2

DSA

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

```
MIIC8DCCAq
+gAwIBAgIGAXjwLk46MAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxED
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYQAAoGAGG2m8EKmaf5qQqj3Z
+rzSaTaXE3B/R/4A2VuGqRYR7MljPtwdmU6/3CPjCACcZmTIcOAKbFiDHqadQgBZXfzGpzw8Zo
+eYmmk5fXycgnj57PYH1dIWU6I7mCbAah5MZMcmHaTmIsomGrhcnWB8d8qOU7oZ0UWK4lbiAQs1MihoUwCQYHKoZIzjg
WmbaU7YM5GwCFCvIJOesO5hZ8PHC52dAR8WWC6oe
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICMzCCAZygAwIBAgIGAXjwLkiaMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIDBBXYXNoaW5n
VvR1+45Aey5zn3vPk6xBm5o9grSDL6D2iAuprQnfVXn8CIbSDbWFhA3fi5ippjKkh3sl8VyCvCOUXKdOaNrYBrPRkrdH
+3m/
rxIUZ2IK1fDlC6sWAjddf6sBrV2w2a78H0H8EwuwiSgttURBjwJ7KPPJCqaqrQIDAQABMA0GCSqGSIb3DQEBBQUAA4GB
+FzqQDzun/
iMMzcFucmLMl5BxEblrFXOz7IIuOeiGkndmrqUeDCykztLku45s7hxdNy4ltTuVAaE5aNBdw5J8U1mRvsKvHLy2ThH6h
+hBgiphYp84DUbWVYeP8YqLEJSqscKscWC
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJALWSmO6DvSpQMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMjA3MTgx
MzU4NDNaGA8yMjAxMTIyMjEzNTg0M1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAuAAhuSpsHC00/fD2zNlBDpNLRndi9qbHsNeuz3WqN7Samj2aSrM2hS+i
hUxx0BspZj0tZCOsbpPZ+i74NOEQtFeqQoEGvKhB1nJiF4y5I81HDhs5qHvoIivm
7rbbik3zgm1PqS/DmDjVQaXPcD31Rd9ILwBmWEwJqHigyNVlxYtCzTQcrlBrvNZM
dnNgCDAdX/HBEFxx9O12xeu0bSt0s+PJWZ1RTbYrNe7LIH6ntUqHxP/ziQ5trXEZ
uqy7aWk1L8uK4jmyNph0lbaqBa3Y6pYmU1nC27UE4i3fnPB0LSiAr+SrwVvX1g4z
ilo8kr+tbIF+JmcgYLBvO8Jwp+EUqQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUwvGzKJL9A5LReJ4Fxo5K6I20xcowgY4GA1UdIwSBhjCBg4AUwvGz
KJL9A5LReJ4Fxo5K6I20xcqhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
```

```
b24gV2ViIFNlcnZpY2VzIExMQ4IJALWSmO6DvSpQMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAJAZd31jyoTGLawAD2+v/vQsaB9vZIx5EImi
G8YGkd61uFWeNhAmtrwyE/i6FDSIphDrMHBkvw/D3BsqK+Ev/JOK/VYuaYDx/8fp
H4cwp9jC57CXzdIDREWNf6M9PsHFg2WA9XNNtClOZL5WJiJwel8eDSg+sqJUxE0l
MW+QChq/20F6niyaRK4bXrZq14as7h+F9u3A9xHEOVP7Zk9C2ehrBXzCMLSDt3GV
fEuMea2RxMhozwz34Hkdb6j18qoCfygubulovRNQjKw/cEmgPR16KfZPP5caILVt
9qkYPvePmbiVswZDee73cDymJYxLqILpOZwyXvUH8StiH42FHZQ=
-----END CERTIFICATE-----
```

유럽(스톡홀름) - eu-north-1

DSA

```
-----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-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUN1c9U6U/xiVDFgJcYKZB4NkH1QEwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE2MDYwM1oXDTI5MDQyODE2MDYwM1owXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
```

```
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUN1c9U6U/
xiVDFgJcYKZB4NkH1QEwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQBTIQdoFSDRHkpqNPUbZ9WXR2O5v/9bpmHojMYZb3Hw46wsaRso7STiGGX/
tRqjIkPUIXsdhZ3+7S/RmhFznmZc8e0bjU4n5vi9CJtQSt+1u4E17+V2bF+D3h/7
wcfE0l3414Q8JaTDtfEf/aF3F0uyBvr4MDMd7mFvAMmDmBPSlA==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJALc/uRxg++EnMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xODA0MTAx
NDAwMTFaGA8yMTk3MDkxMzE0MDAxMVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAzwCGJEJIxqtr2PD2a1mA6LhRzKhTBa1AZsg3eYfpETXIVlrpojMfvVoN
qHvGshWLgrGTT6os/3gsaADheSaJKavxwX3X6tJA8fvEGqr3a1C1MffH9hBWbQqC
LbfUTAbkwis4GdTUwOwPjT1Cm3u9R/VzilCNwkj7iQ65AFAI8Enmsw3UGldEsop4
yChKB3KW3WI0FTh0+gD0YtjrqqYJxpGOYBpJp5vwdd3fZ4t1vidmDMs7liv4f9Bx
p0oSmUobU4GUlFhBchK1DukICVQdnOVzdMonYm7s+HtpFbVHR8yf6QoixBKGdSal
mBf7+y0ixjCn0pnC0VLVooGo4mi17QIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQDG
4ONZiixgk2sjJctwbyD5WKLTH6+mxYcDw+3y/F0fWz561YORhP2FNnPOmEkf0Sl/
Jqk4svzJbCbQeMzRoyaya/46d7UioXMHRZam5IaGBhOdQbi97R4VsQjwQj0RmQsq
yDueDyuKTwWLK9KnvI+ZA6e6bRkdNGflK4N8GGKQ+fBhPwVELkbT9f16OJkezeeN
S+F/gDADGJgmPXfjogICb4Kvshq0H5Lm/xZlDULF2g/cYhyNY6EOI/eS5m1I7R8p
D/m6WoyZdpInxJfxW616OMkxQMRVsruLTNGtby3u1g6ScjmpFtvAMhYejBSdzKG4
FEyxIdEjoeO1jhTsck3R
-----END CERTIFICATE-----
```

유럽(취리히) - eu-central-2

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq
+gAwIBAgIGAXjXiKJnMAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxED
```

```
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYQAAoGAYNjaCNg/
cfgQOl1BUj5ClUulqwZ9Q+SfDzPZh9D2C0VbiRANiZoxrV8RdgmzzC5T7VcriVwjwvta2Ch//
b+sZ86E5h0XWWr+BeEjD9cu3eDj12XB5sWEbNHNx49p5Tmtu5r2LDtlL8X/
Rpfalu2Z2OJgjFJWGf7hRwxe456n
+lowCQYHKoZIzjgEAwMvADAsAhRChsLcj4U5CVb2cp5M0RE1XbXmhAIUeGSnH+aiUQIWmPEFja+itWDufIk=
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICMzCCAZygAwIBAgIGAXjSGFGiMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIDBBXYXNoaW5n
opKZAUusJx2hpgU3pUHhlp9ATh/VeVD582jTd9IY
+8t5MDa6Z3fGliByEiXz0LEHdi8MBacLREu1TwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAILlpoE3k9o7KdALAxsFJNit
+g3RMzdbiFM+7MA63Nv5fsf+0xgcjSNBElvPCDKFvTJl4QQhToy056llO5GvdS9RK
+H8xrP2mrqngApoKTApv93vHBixgFSn5KrczRO0YSm3OjkqbydU7DFlmkXXR7GYE+5jbHvQHYiT1J5sMu
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJALvT012pxTxNMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMjA3MTgx
NTEyMDdaGA8yMjAxMTIyMjE1MTIwN1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAyn+Lsnq1ykrfYlZkk6aAAYNReNd9Iw8AUwCBkgOr2eBiBBepYxHwU85N
++moQ+j0EV2VaahBeTLShGZZS1HsyK8+cYT2QzpghIoamcYhrPXyIxlWiRQlaqSg
OFiE9bsqL3rCF5Vz+tOiTe5W/7ojfOFls6++g7ZpobwJlpMbuJepqyeHMPyjvO5A
age81lJewc4bxo2ntaW0HCqNksqfYB78j6X6kn3PFpX7FaYAwZA+Xx6C7UCY7rNi
UdQzfAo8htfJi4chz7frpUdQ9kl3IOQrsLshBB5fFUjl09NiFipCGBwi+8ZMeSn1
5qwBI01BWXPfG7WX6Owyjhmh6JtE1wIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU8HN4vvJrsZgPQeksMBgJb9xR1yYwgY4GA1UdIwSBhjCBg4AU8HN4
vvJrsZgPQeksMBgJb9xR1yahYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJALvT012pxTxNMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAGlHYDtcHpfBvdHx9HeQE8HgNugJUPdEqxun
t9U33p8VFrs+uLPtrOd9HDJEGvvs5h84EUie/oGJxRt7V1Vlid1PvHf6cRmpjgqY
```

```
YdggAVkZtY/PnFVmzf2bMVlSQPrqCl7U0zaw2Kvnj4zgX0rZyCetgrRZSUSxotyp
978Wy9ccXwVSeYG/YAr5rJpS6ZH7eRQvUY0IzwFNeaOPgOTEVpcjWlV6+MQEvsEx
W85q+s6AVr49eppEx8SLJsl0C23yB+L+t32tAveQImRWtJMpzZ5cxh/sYgDVeoC0
85H1NK/7H9fAzT1cPu1oHSnB0xYzzHGOAmXmusMfwUk8fL1RQkE=
-----END CERTIFICATE-----
```

이스라엘(텔아비브) – il-central-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq+gAwIBAgIGAX0QPi
+9MAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcMB1NlYX
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/
hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYQAAoGAbazCL5XXyPmcw3+oMYQUF5/9YogW6D0FZbYuyPgjOoUwWdl6fjlzWca
pq+l1ezuK2DFOzNTEyPEwwCQYHKoZIzjgEAwMvADAsAhRt1jkpXsvrS
+xTo2M9h2s2uLAhEQIUOZ2FcnTSrshF2EIdixZZwtNv66Q=
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICMzCCAZygAwIBAgIGAX0QQGVLMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIDBBXYXNoaW5n
+S8vOy5hpLoRe4RkOrY0cM3bN07GdEMlin5mU0y1t8y3ct4YewvmkgT42kTyMM
+t1K4S0xsqjXxxS716uGYh7eWtkxrCihj8AbXN/6pa095h
+7TZyl2n83keiNUzM2KoqQVMwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBADwA6VVEIIZD2YL00F12po40xDLzIc9XvqFPS
FmU7H8s62/jD6cOR1A1cClIyZUe1yT1ZbPySCs43J+Thr8i8FSRxzDBSZZi5foW
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAOVp1h2I9wW7MA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMjA3MTUx
MjQ0MTJaGA8yMjAxMTIxOTEyNDQxMlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
```

```
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA13PkyWvl61iV/SYfO1UFO76UpDfPm2SF/Rz/o33cm699X++EYPxTnoEc
vmWeS0I7eDXc4OCUiToG0sEx0klE0CX1Z1tK6qJ+zgWQLZ9SZEC9H0NsSA6LhrHu
Nq0dzeK3LjhdfcX46/4GqdiptpdTuM4m/hOQ5yx4JMQ/n1sdpv4M5VLRWwWW9Lem
ufb79Id7O9SispxgRsz1KXIjp7N9S4BY7itSXz97uSyzTqEjWZ6mDUhTu3t21GKC
6flALGTTrG2yghEhz53rkvLsvwzjPSSlT6LIf0mrRPzHaf+EdaKoasElE1SHh+ZH
9mI81HywpE+HZ+W+5hBCvjYp90Y1fwIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU58tN2JO+yEGq5JbIXxGi4vRVPyIwgY4GA1UdIwSBhjCBg4AU58tN
2JO+yEGq5JbIXxGi4vRVPyKhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAOVp1h2I9wW7MBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBANBN0e1EqNy4+IU2yQzMJ+Wy5ZIOtTP6GSBR
7muVY1bDeAwtNTEOpwgrZVlC7/xq5Q0LC1y0Z70hHXEf8au7qStaAoUtxzvhTAZI
NC0lwoFU56UFw4N0vZII17iqEfoqRC4PpI3OxqEJHFy0VLlvAzJoKB4QLLqDAYVA
LXCiOLoVT+y9tRYsxw5MyO0Bi6fxQIIADl2bE9xkunTN1Jkkwqo3LxNy/ryz4QWR
8K7jHUItifv4h/hxBKpHEquN8CkdvM9oeGl7I8PFrSFEpGr1euDXY0euZzzYiDBV
m6GpTJgzpVsEuIX52dPcPemwQncoIfZyhWDW85MJUnby2WTEcFo=
-----END CERTIFICATE-----
```

멕시코(중부)-mx-central-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq8CCQD4QwfTExrgxTAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDExMTQwOTQ4MjRaGA8y
MDUwMTExNTA5NDgyNFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0
b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIg
U2VydmljZXMgTExDMIIBtzCCASwGByqGSM44BAEwggEfAoGBAK+pJEOtD2RFlCbf
oAzONffZa42FB8G6OA/30DT81xWvB2Hnmgw7LfiELYqgfuck3YSDffNDcmiFTJuX
AU9u0NtbbxcOytcdUq4HInpHiB85I6WWWKYB66aGbvYowUPpqBZkASb1i/pYAh5W
nNYvx408008tTzqpMfZDchJHzWqBAhUA0ogoJzTw2/4pKhz9aqTsRCzRVPsCgYEA
qklRUcuU0du/bT/M6kWxYvrTGhO9KXQe+7RtbaIq4dWRsCBnO4smDY/GmI9H8ew1
LRJ9AcLGmxDm795CvVZkHNcht7PDAREagWmz2YvhLA+ev6U0RpfdlBXCck2p1CxQ
LMtoFO7DThksHIQBtnlQdHDvguKEZQz/Iobhne6Kb3wDgYQAAoGAQSTlOqFq6RtR
Jvp406XhGl+eO9SQUpevpzG3Dzbdy6EQ8PBJDOHHJvxbebvpnQssh0CnQfTkSw26
jwPy9V5zRC0ZezwnRTyGSJwiErUKDGeVekEAoNJL1USy8jKkgBajlskZrR+0JHDN
Jv3UwYIPplc4ZS2f2E7btrtlaWt/P70wCQYHKoZIzjgEAwMvADAsAhRXljPWKWYf
61DkDVdgPPHsS2/LuwIUejcGV9WuS3uPvJ6lmn4opxUGBZw=
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICPzCCAaigAwIBAgIUCmzpTTMBQYItpMC2VDYsZfIAS7IwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcMB1NlYXR0bGUxIDAeBgNVBAoMF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQwMjEyNTAzNloXDTI5MDQwMTEyNTAzNlowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcMB1NlYXR0bGUxIDAe
BgNVBAoMF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQDfvOnCzm1iN58Nm7k6ehoy6v0lnnFI617D6CY3bfuq01RCdEQL
96+pYawJieTH8JAQKj02CAa3AeaqdXTE/pDhI/YKLreeMb4K68WMn24Wjjs6oxjB
bAmsKXtt9ihKHGBFNUhgFrNFYyA2i7ieJviwpHjQ/XgXiG2u1/t/4VydUwIDAQAB
MA0GCSqGSIb3DQEBCwUAA4GBAL5+vvj4lhaE+J5tuCqV3XJzDd97lsD4le2O2uGw
P0sGdUcRAdxzU3Bwq/hhtzNWnfwo0aCEQKmLM7xyd3nUa0VvKXLq+DDuayipWINr
OATnNxFRe99d38qHTR1dgkjZdKbbtnl6O4fgM57tVEuQJd/N4IL19jaRcJ5Ip+9t
3y5t
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJALxuE00HoJomMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDEwMzEx
MTE0MzNaGA8yMjA0MDQwNjExMTQzM1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA13xWucALo8M/TXbZJgHrqFqFON91dSLPE/eLbmeIJbP1pb9ICd33qKAx
HlKSrXI9b9YS3U1Pl0bF3ZgfeE/x4Y0KDDZwzpf07H8IgRittULJoNLYVKCJXWPq
Ky1qvDJX3653dUbUu9eAdVCTRgk7eKpPBLAmW27+pgAGzEYrVV3u2AvqNTonvfTU
sPgEVnAL1J490pNM85KtFynxFTWGigHkd3BHidxmLrTH4I4eRxNZ9q/3gsDW+zKt
jQlpM7JzZa2OqxsF5YQDh1fF52Emqsr+ufLeGqDLOgT1QWcqpz57AX8GqZpgZULo
itCRNXbQDzZY9FxIgpiFJv3y/qYYDQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQCt
MTeH4EgqjJEjb1qm5tzXYurLprVrEVQ+PhGXJFJd3xAZyeaDVYy7kio08E2xhmHd
HtkBDty2KnOHsTQmeAJCci7d4tYXZ/1qe341wwm9OoFcO8jhIndx6FXJCgQUY4dL
AAr9HQJFWG5dMZgbi1Zuhxdio3sSoOBjL2p7QIsGNkITvCDIs/H0/szpJnyyyIqu
wmUhSel5hdy5MwOsyUKVGnAdaS5Vd9oL4kLszS9nBZ7ny6BC9odIkFAdGqQ5vM4z
vcbfOq14hjatQmJgJhksN/0Dpl78Gheq0pIhP8LTkAOEG2832nQLzCa3oxSk8otG
GJXkzzyQjse+13r8+yNJ
-----END CERTIFICATE-----
```

중동(바레인) - me-south-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7jCCAq4CCQCVWIgSmP8RhTAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xOTAyMDUxMzA2MjFaFw00
NTAyMDUxMzA2MjFaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbgwggEsBgcqhkjOOAQBMIIBHwKBgQDcwojQfgWdV1QliO0B
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-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDPDCCAqWgAwIBAgIJAMl6uIV/zqJFMA0GCSqGSIb3DQEBCwUAMHIxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIDApXYXNoaW5ndG9uMRAwDgYDVQQHDAdTZWF0dGxlMSAw
HgYDVQQKDBdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzEaMBgGA1UEAwwRZWMyLmFt
YXpvbmF3cy5jb20wIBcNMTkwNDI2MTQzMjQ3WhgPMjE5ODA5MjkxNDMyNDdaMHIx
CzAJBgNVBAYTAlVTMRMwEQYDVQQIDApXYXNoaW5ndG9uMRAwDgYDVQQHDAdTZWF0
dGxlMSAwHgYDVQQKDBdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzEaMBgGA1UEAwwR
ZWMyLmFtYXpvbmF3cy5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBALVN
CDTZEnIeoX1SEYqq6k1BV0ZlpY5y3KnoOreCAE589TwS4MX5+8Fzd6AmACmugeBP
Qk7Hm6b2+g/d4tWycyxLaQlcq81DB1GmXehRkZRgGeRge1ePWd1TUA0I8P/QBT7S
gUePm/kANSFU+P7s7u1NNl+vynyi0wUUrw7/wIZTAgMBAAGjgdcwgdQwHQYDVR0O
BBYEFILtMd+T4YgH1cgc+hVsVOV+480FMIGkBgNVHSMEgZwwgZmAFILtMd+T4YgH
1cgc+hVsVOV+480FoXakdDByMQswCQYDVQQGEwJVUzETMBEGA1UECAwKV2FzaGlu
Z3RvbjEQMA4GA1UEBwwHU2VhdHRsZTEgMB4GA1UECgwXQW1hem9uIFdlYiBTZXJ2
aWNlcyBMTEMxGjAYBgNVBAMMEWVjMi5hbWF6b25hd3MuY29tggkAyXq4hX/OokUw
DAYDVR0TBAUwAwEB/zANBgkqhkiG9w0BAQsFAAOBgQBhkNTBIFgWFd+ZhC/LhRUY
4OjEiykmbEp6hlzQ79T0Tfbn5A4NYDI2icBP0+hmf6qSnIhwJF6typyd1yPK5Fqt
NTpxxcXmUKquX+pHmIkK1LKDO8rNE84jqxrxRsfDi6by82fjVYf2pgjJW8R1FAw+
```

```
mL5WQRFexbfB5aXhcMo0AA==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJANZkFlQR2rKqMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xOTAyMDUx
MzA2MjBaGA8yMTk4MDcxMTEzMDYyMFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAy4Vnit2eBpEjKgOKBmyupJzJAiT4fr74tuGJNwwa+Is2vH12jMZn9Il1
UpvvEUYTIboIgISpf6SJ5LmV5rCv4jT4a1Wm0kjfNbiIlkUi8SxZrPypcw24m6ke
BVuxQZrZDs+xDUYIZifTmdgD50u5YE+TLg+YmXKnVgxBU6WZjbuK2INohi71aPBw
2zWUR7Gr/ggIpf635JLU3KIBLNEmrkXCVSnDFlsK4eeCrB7+UNak+4BwgpuykSGG
Op9+2vsuNqFeU1l9daQeG9roHR+4rIWSPa0opmMxv5nctgypOrE6zKXx2dNXQldd
VULv+WH7s6Vm4+yBeG8ctPYH5GOo+QIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBs
ZcViiZdFdpcXESZP/KmZNDxB/kktlIEIhsQ+MNn29jayE5oLmtGjHj5dtA3XNKlr
f6PVygVTKbtQLQqunRT83e8+7iCZMKI5ev7pITUQVvTUwI+Fc01JkYZxRFlVBuFA
WGZO+98kxCS4n6tTwVt+nSuJr9BJRVC17apfHBgSS8c5OWna0VU/Cc9ka4eAfQR4
7pYSDU3wSRE01cs30q34lXZ629IyFirSJ5TTOIc0osNL7vwMQYj8HOn4OBYqxKy8
ZJyvfXsIPh0Na76PaBIs6ZlqAOflLrjGzxBPiwRM/XrGmF8ze4KzoUqJEnK13O6A
KHKgfiigQZ1+gv5FlyXH
-----END CERTIFICATE-----
```

중동(UAE) - me-central-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq
+gAwIBAgIGAXjXhqnnMAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxED
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYQAAoGAW+csuHsWp/7/
pv8CTKFwxsYudxuR6rbWaHCykIeAydXL9AWnphK6yp1ODEMBFl68Xq8Hp23sOWyf8moOhqCom9+0+ovuUFdpvCie86bp
TOZU568Ty1ff3dDWbdRzeNQRHodRG+XEQSizMkAreeWt4kBa+PUwCQYHKoZIzjgEAwMvADAsAhQD3Z
+XGmzKmgaLgGcVX/Qf1+Tn4QIUH1cgksBSVKbWj81tovBMJeKgdYo=
```

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

RSA

```
-----BEGIN CERTIFICATE-----
MIICMzCCAZygAwIBAgIGAXjRrnDjMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIDBBXYXNoaW5n
KyA6zyruJQrYy00a6wqLA7eeUzk3bMiTkLsTeDQfrkaZMfBAjGaaOymRo1C3qzE4rIenmahvUplu9ZmLwL1idWXMRX2R
+d2SeoKOKQWoc2UOFZMHYxDue7zkyk1CIRaBukTeY13/
RIrlc6X61zJ5BBtZXlHwayjQIDAQABMA0GCSqGSIb3DQEBBQUAA4GBABTqTy3R6RXKPW45FA+cgo7YZEj/
Cnz5YaoUivRRdX2A83BHuBTvJE2+WX00FTEj4hRVjameE1nENoO8Z7fUVloAFDlDo69fhkJeSvn51D1WRrPnoWGgEfr1
B+Wqm3kVEz/QNcz6npmA6
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAM4h7b1CVhqqMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMjA0MTEx
MDE1MDNaGA8yMjAxMDkxNTEwMTUwM1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEApybTWfMOhSoMpqPo72eqAmnn1dXGZM+G8EoZXzwHwT/+IHEXNB4q5N6k
tudYLre1bJxuzEw+iProSHjmb9bB9YscRTofjVhBlt35Fc+i8BaMeH94SR/eE8QO
mll8gnLNW3d62lyuhzuyv1e5wVlRqzYw+X2zRH4/wRDOCOpzjKoHIgyPKsMgwsw5
aTZhNMsGxZN9dbkf0iCGeQLDytwU/JTh/HqvSr3VfUOapTJJiyAxoCtZWgp1/7wC
RvOCSMRJobpUqxZgl/VsttwNkikSFz1wGkcYeSQvk+odbnYQckA8tdddoVI56eD4
qtREQvfpMAX5v7fcqLexl5d5vH8uZQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUOadrbTs+OhzwoAgUJ7RqQNdwufkwgY4GA1UdIwSBhjCBg4AUOadr
bTs+OhzwoAgUJ7RqQNdwufmhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAM4h7b1CVhqqMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAICTdAOGEOnII8HaGCpCB8us/hGFaLptJaAf
D5SJAyVy66/mdfjGzE1BKkKxnbxemEVUIzbRidOnyilB+pKwN3edAjTZtWdpVA0V
R/G/qQPmcVljtycBz4VC6SuOUYflGzLH1GZ6GJWbuDtFzw8r7HGdRN1wrEPe3UF2
sMpuVezqnRUdvVRoVQP4jFgNsE7kNvtn2NiPhb/CtrxpcwIQ7r6YeoHcBSheuV1Z
xZDHynC3KUprQGx1+Z9QqPrDf180MaoqAlTl4+W6Pr2NJYrVUFGS/ivYshMg574l
CPU6r4wWZSKwEUXq4BInYX6z6iclp/p/J5QnJp2mAwyi6M+I13Y=
-----END CERTIFICATE-----
```

남아메리카(상파울루) - sa-east-1

DSA

```
-----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-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUX4Bh4MQ86Roh37VDRRX1MNOB3TcwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE2NDYwOVoXDTI5MDQyODE2NDYwOVowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUX4Bh4MQ8
6Roh37VDRRX1MNOB3TcwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQBnhocfH6ZIX6F5K9+Y9V4HFk8vSaaKL5ytw/P5td1h9ej94KF3xkZ5fyjN
URvGQv3kNmNJBoNarcP9I7JIMjsNPmVzqWawyCEGCZImoARxSS3Fc5EAs2PyBfcD
9nCtzMTaKO09Xyq0wqXVYn1xJsE5d5yBDsGrzaTHKjxo61+ezQ==
```

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

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAMcyoxx4U0xxMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA4MTQw
ODU4MDJaGA8yMTk1MDExNzA4NTgwMlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAw45IhGZVbQcy1fHBqzROhO8CsrDzxj/WP4cRbJo/2DAnimVrCCDs5O86
FA39Zo1xsDuJHDlwMKqeXYXkJXHYbcPWc6EYYAnR+PlLG+aNSOGUzsy202S03hT0
B20hWPCqpPp39itIRhG4id6nbNRJOzLm6evHuepMAHR4/OV7hyGOiGaV/v9zqiNA
pMCLhbh2xk0PO35HCVBuWt3HUjsgeks2eEsu9Ws6H3JXTCfiqp0TjyRWapM29OhA
cRJfJ/d/+wBTz1fkWOZ7TF+EWRIN5ITEadlDTPnF1r8kBRuDcS/lIGFwrOOHLo4C
cKoNgXkhTqDDBDu6oNBb2rS0K+sz3QIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUqBy7D847Ya/w321Dfr+rBJGsGTwwgY4GA1UdIwSBhjCBg4AUqBy7
D847Ya/w321Dfr+rBJGsGTyhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAMcyoxx4U0xxMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBACOoWSBf7b9AlcNrl4lr3QWWSc7k90/tUZal
PlT0G3Obl2x9T/ZiBsQpbUvs0lfotG0XqGVVHcIxF38EbVwbw9KJGXbGSCJSEJkW
vGCtc/jYMHXfhx67Szmftm/MTYNvnzsyQQ3v8y3Rdah+xe1NPdpFrwmfL6xe3pFF
cY33KdHA/3PNLdn9CaEsHmcmj3ctaaXLFIzZhQyyjtsrgGfTLvXeXRokktvsLDS/
YgKedQ+jFjzVJqgr4NjfY/Wt7/8kbbdhzaqlB5pCPjLLzv0zp/XmO6k+JvOePOGh
JzGk5t1QrSju+MqNPFk3+1O7o910Vrhqw1QRB0gr1ExrviLbyfU=
-----END CERTIFICATE-----
```

AWS GovCloud(미국 동부) - us-gov-east-1

DSA

```
-----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-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIULVyrqjjwZ461qelPCiShB1KCCj4wDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDUwNzE1MjIzNloXDTI5MDUwNjE1MjIzNlowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCpohwYUVPH9I7Vbkb3WMe/JB0Y/bmfVj3VpcK445YBRO9K80al
esjgBc2tAX4KYg4Lht4EBKccLHTzaNi51YEGX1aLNrSmxhz1+WtzNLNUsyY3zD9z
vwX/3k1+JB2dRA+m+Cpwx4mjzZyAeQtHtegVaAytkmqtxQrSCexBxvqRqQIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQU1ZXneBYnPvYXkHVlVjg7918V
gE8wgZkGA1UdIwSBkTCBjoAU1ZXneBYnPvYXkHVlVjg7918VgE+hYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IULVyrqjjw
Z461qelPCiShB1KCCj4wEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQBfAL/YZv0y3zmVbXjyxQCsDloeDCJjFKIu3ameEckeIWJbST9LMto0zViZ
puIAf05x6GQiEqfBMk+YMxJfcTmJB4Ebaj4egFlslJPSHyC2xuydHlr3B04INOH5
Z2oCM68u6GGbj0jZjg7GJonkReG9N72kDva/ukwZKgq8zErQVQ==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJALPB6hxFhay8MA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xODA0MTAx
MjMyNDlaGA8yMTk3MDkxMzEyMzI0OVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAva9xsI9237KYb/SPWmeCVzi7giKNron8hoRDwlwwMC9+uHPd53UxzKLb
pTgtJWAPkZVxEdl2Gdhwr3SULoKcKmkqE6ltVFrVuPT33La1UufguT9k8ZDDuO9C
```

```
hQNHUdSVEuVrK3bLjaSsMOS7Uxmnn7lYT990IReowvnBNBsBlcabfQTBV04xfUG0
/m0XUiUFjOxDBqbNzkEIblW7vK7ydSJtFMSljga54UAVXibQt9EAIF7B8k9l2iLa
mu9yEjyQy+ZQICTuAvPUEWe6va2CHVY9gYQLA31/zU0VBKZPTNExjaqK4j8bKs1/
7dOV1so39sIGBz21cUBec1o+yCS5SwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBt
hO2W/Lm+Nk0qsXW6mqQFsAou0cASc/vtGNCyBfoFNX6aKXsVCHxq2aq2TUKWENs+
mKmYu1lZVhBOmLshyllh3RRoL3Ohp3jCwXytkWQ7ElcGjDzNGc0FArzB8xFyQNdK
MNvXDi/ErzgrHGSpcvmGHiOhMf3UzChMWbIr6udoDlMbSIO7+8F+jUJkh4Xl1lKb
YeN5fsLZp7T/6YvbFSPpmbn1YoE2vKtuGKxObRrhU3h4JHdp1Zel1pZ6lh5iM0ec
SD11SximGIYCjfZpRqI3q50mbxCd7ckULz+UUPwLrfOds4VrVVSj+x0ZdY19Plv2
9shw5ez6Cn7E3IfzqNHO
-----END CERTIFICATE-----
```

AWS GovCloud(미국 서부) - us-gov-west-1

DSA

```
-----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-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUe5wGF3jfb7lUHzvDxmM/ktGCLwwwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDUwNzE3MzAzMloXDTI5MDUwNjE3MzAzMlowXDELMAkGA1UEBhMCVVMx
```

```
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCpohwYUVPH9I7Vbkb3WMe/JB0Y/bmfVj3VpcK445YBRO9K80al
esjgBc2tAX4KYg4Lht4EBKccLHTzaNi51YEGX1aLNrSmxhz1+WtzNLNUsyY3zD9z
vwX/3k1+JB2dRA+m+Cpwx4mjzZyAeQtHtegVaAytkmqtxQrSCexBxvqRqQIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQU1ZXneBYnPvYXkHVlVjg7918V
gE8wgZkGA1UdIwSBkTCBjoAU1ZXneBYnPvYXkHVlVjg7918VgE+hYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUe5wGF3jf
b7lUHzvDxmM/ktGCLwwwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQCbTdpx1Iob9SwUReY4exMnlwQlmkTLyA8tYGWzchCJOJJEPfsW0ryy1A0H
YIuvyUty3rJdp9ib8h3GZR71BkZnNddHhy06kPs4p8ewF8+d8OWtOJQcI+ZnFfG4
KyM4rUsBrljpG2aOCm12iACEyrvgJJrS8VZwUDZS6mZEnn/lhA==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJANCOF0Q6ohnuMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA5MTAx
OTQyNDdaGA8yMTk1MDIxMzE5NDI0N1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAzIcGTzNqie3f1olrrqcfzGfbymSM2QfbTzDIOG6xXXeFrCDAmOq0wUhi
3fRCuoeHlKOWAPu76B9os71+zgF22dIDEVkpqHCjBrGzDQZXXUwOzhm+PmBUI8Z1
qvbVD4ZYhjCujWWzrsX6Z4yEK7PEFjtf4M4W8euw0RmiNwjy+knIFa+VxK6aQv94
lW98URFP2fD84xedHp6ozZlr3+RZSIFZsOiyxYsgiwTbesRMI0Y7LnkKGCIHQ/XJ
OwSISWaCddbu59BZeADnyhl4f+pWaSQpQQ1DpXvZAVBYvCH97J1oAxLfH8xcwgSQ
/se3wtn095VBt5b7qTVjOvy6vKZazwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQA/
S8+a9csfASkdtQUOLsBynAbsBCH9Gykq2m8JS7YE4TGvqlpnWehz78rFTzQwmz4D
fwq8byPkl6DjdF9utqZ0JUo/Fxelxom0h6oievtBlSkmZJNbgc2WYm1zi6ptViup
Y+4S2+vWZyg/X1PXD7wyRWuETmykk73uEyeWFBYKCHWsO9sI+62O4Vf8Jkuj/cie
1NSJX8fkervfLrZSHBYhxLbL+actVEo00tiyZz8GnhgWx5faCY38D/k4Y/j5Vz99
7lUX/+fWHT3+lTL8ZZK7fOQWh6NQpI0wTP9KtWqfOUwMIbgFQPoxkP00TWRmdmPz
WOwTObEf9ouTnjG9OZ20
-----END CERTIFICATE-----
```

EC2 인스턴스의 정밀 클럭 및 시간 동기화

Amazon EC2 인스턴스에서 일관되고 정확한 시간 참조는 많은 서버 작업과 프로세스에 매우 중요합니다. 시스템 로그의 타임스탬프는 문제가 발생한 시기와 이벤트의 연대순을 식별하는 데 중요한 역할

을 합니다. AWS CLI 또는 AWS SDK를 사용하면 인스턴스에서 요청하는 경우 이러한 도구가 사용자를 대신하여 요청에 서명합니다. 인스턴스의 날짜 및 시간 설정이 정확하지 않아 서명 날짜와 요청 날짜 불일치로 인해 AWS가 요청을 거부할 수 있습니다.

이러한 중요한 측면을 해결하기 위해 Amazon은 모든 EC2 인스턴스에서 액세스할 수 있고 다양한 AWS 서비스 인스턴스에서 사용되는 Amazon Time Sync Service를 제공합니다. 서비스는 AWS 리전의 위성 연결 및 원자 기준 시계 집합을 사용하여 협정 세계시(UTC) 세계 표준의 정확한 현재 시간 판독을 제공합니다.

최상의 성능을 위해서는 EC2 인스턴스에서 로컬 Amazon Time Sync Service를 사용하는 것이 좋습니다. 인스턴스의 로컬 Amazon Time Sync Service에 백업하거나 Amazon EC2 외부의 리소스를 Amazon Time Sync Service에 연결하려면 `time.aws.com`에 있는 퍼블릭 Amazon Time Sync Service를 사용하면 됩니다. 로컬 Amazon Time Sync Service와 마찬가지로 퍼블릭 Amazon Time Sync Service는 UTC에 추가된 윤초를 자동으로 제거합니다. 퍼블릭 Amazon Time Sync Service는 각 AWS 리전의 위성 연결 및 원자 기준 시계 플릿을 사용하여 전 세계적으로 지원됩니다.

하드웨어 패킷 타임스탬핑

인스턴스에서 하드웨어 패킷 타임스탬핑을 활성화하여 모든 수신 네트워크 패킷에 64비트 나노초 정밀도 타임스탬프를 추가할 수 있습니다. 하드웨어 패킷 타임스탬핑은 패킷이 커널, 소켓 또는 애플리케이션 계층에 도달하기 전에 하드웨어 수준에서 발생하므로 소프트웨어 타임스탬핑으로 추가된 지연을 우회합니다. 하드웨어 타임스탬핑을 위한 기본 참조 클럭은 Amazon Time Sync Service PTP 하드웨어 클럭입니다.

이점

하드웨어 패킷 타임스탬핑은 다음과 같은 이점을 제공합니다.

- 이벤트 순서를 개선합니다. 이 순서는 패킷이 EC2 인스턴스에 도착하는 실제 순서를 결정하는 데도 사용할 수 있으므로 공정한 패킷 처리가 보장됩니다.
- 단방향 네트워크 지연 시간을 측정합니다.
- 대부분의 온프레미스 솔루션에 비해 더 높은 정밀도 및 정확도로 분산 트랜잭션 속도를 높입니다.

사전 조건 및 구성

하드웨어 패킷 타임스탬핑을 활성화하려면 인스턴스가 다음 사전 조건을 충족해야 합니다.

- Linux 인스턴스여야 합니다.
- PTP 하드웨어 클럭을 지원하기 위한 요구 사항을 충족합니다.

구성 지침은 GitHub의 Linux kernel driver for Elastic Network Adapter (ENA) family에서 Hardware Packet Timestamping 페이지를 참조하세요.

윤초

1972년에 도입된 윤초는 국제 원자시(TAI)와 태양시(Ut1) 간의 차이를 수용하기 위해 지구 자전의 불규칙성을 고려하여 UTC 시간을 1초씩 조정하는 것입니다. 고객을 대신하여 윤초를 관리하기 위해 Amazon Time Sync Service 내에서 윤초 스미어링(leap smearing)을 설계했습니다. 자세한 내용은 Look Before You Leap – The Coming Leap Second and AWS를 참조하세요.

윤초는 사라지고 있으며, Amzaon은 2035년 또는 그 이전에 윤초를 없애기로 한 제27차 도량형 총회 결정을 전적으로 지지합니다.

이러한 전환을 지원하기 위해 로컬 NTP 연결 또는 퍼블릭 NTP 풀(`time.aws.com`)을 통해 Amazon Time Sync Service에 액세스할 때 윤초 발생 시 시간 스미어링(smearing)을 계획하고 있습니다. 그러나 PTP 하드웨어 클럭은 스미어링한 시간 옵션을 제공하지 않습니다. 윤초가 발생하는 경우 PTP 하드웨어 클럭은 UTC 표준에 따라 윤초를 추가합니다. 대부분의 경우 윤초를 스미어링한 시간 소스와 윤초 시간 소스는 동일합니다. 그러나 윤초 발생 시에는 다르므로 시간 클라이언트 구성에서 스미어링한 시간 소스와 스미어링하지 않은 시간 소스 모두 사용하지 않는 것이 좋습니다.

주제

- 로컬 Amazon Time Sync Service를 사용하도록 EC2 인스턴스의 시간 참조 설정
- 퍼블릭 Amazon Time Sync Service를 사용하도록 EC2 인스턴스 또는 인터넷 연결 디바이스의 시간 참조 설정
- Linux 인스턴스의 타임스탬프 비교
- 인스턴스의 시간대 변경

관련 리소스

- AWS 컴퓨팅 블로그: It's About Time: Microsecond-Accurate Clocks on Amazon EC2 Instances
- AWS 클라우드 운영 및 마이그레이션 블로그: Amazon Time Sync Service 및 Amazon CloudWatch를 사용하여 Amazon EC2 인스턴스 클럭 정확도 관리 - 1부
- (Linux) https://chrony-project.org/

로컬 Amazon Time Sync Service를 사용하도록 EC2 인스턴스의 시간 참조 설정

로컬 Amazon Time Sync Service는 NTP(Network Time Protocol)를 사용하거나 지원되는 인스턴스에서 로컬 PTP(Precision Time Protocol) 하드웨어 클럭을 제공합니다. PTP 하드웨어 클럭은 NTP 연결(Linux 및 Windows 인스턴스) 또는 직접 PTP 연결(Linux 인스턴스만 해당)을 지원합니다. NTP 및 직접 PTP 연결은 매우 정확한 동일한 시간 소스를 사용하지만 직접 PTP 연결이 NTP 연결보다 더 정확합니다. Amazon Time Sync Service에 대한 NTP 연결은 윤초 스미어링(leap smearing)을 지원하는 반면, PTP 하드웨어 클럭에 대한 PTP 연결은 시간을 스미어링하지 않습니다. 자세한 내용은 윤초 섹션을 참조하세요.

인스턴스는 다음과 같이 로컬 Amazon Time Sync Service에 액세스할 수 있습니다.

- 다음 IP 주소 엔드포인트에서 NTP를 통해
  - IPv4: `169.254.169.123`
  - IPv6: `fd00:ec2::123`(Nitro 기반 인스턴스에서만 액세스할 수 있습니다.)
- (Linux만 해당) 로컬 PTP 하드웨어 클록에 연결하기 위한 직접 PTP 연결을 통해:
  - `PHC0`

Amazon Linux AMI, Windows AMI와 대부분의 파트너 AMI는 기본적으로 NTP IPv4 엔드포인트를 사용하도록 인스턴스를 구성합니다. 이는 대부분의 고객 워크로드에 권장되는 설정입니다. IPv6 엔드포인트를 사용하거나 PTP 하드웨어 클럭에 직접 연결하려는 경우가 아니면 이러한 AMI에서 시작된 인스턴스에는 추가 구성이 필요하지 않습니다.

NTP 및 PTP 연결에는 VPC 구성 변경이 필요하지 않으며 인스턴스에 인터넷 액세스가 필요하지 않습니다.

고려 사항

- 링크-로컬 주소를 사용하는 서비스에는 초당 1024 패킷(PPS) 제한이 있습니다. 이 제한에는 Route 53 Resolver DNS 쿼리, 인스턴스 메타데이터 서비스(IMDS) 요청, Amazon Time Service 네트워크 타임 프로토콜(NTP) 요청, Windows 라이선스 서비스(Microsoft Windows 기반 인스턴스용) 요청의 집계가 포함됩니다.
- Linux 인스턴스만 직접 PTP 연결을 사용하여 로컬 PTP 하드웨어 클록에 연결할 수 있습니다. Windows 인스턴스는 NTP를 사용하여 로컬 PTP 하드웨어 클록에 연결합니다.

내용

- Amazon Time Sync Service의 IPv4 엔드포인트에 연결합니다.
- Amazon Time Sync Service의 IPv6 엔드포인트에 연결합니다.
- PTP 하드웨어 클럭에 연결

Amazon Time Sync Service의 IPv4 엔드포인트에 연결합니다.

AMI는 기본적으로 Amazon Time Sync Service를 이미 구성했을 수 있습니다. 그렇지 않으면 다음 절차를 사용하여 IPv4 엔드포인트를 통해 로컬 Amazon Time Sync Service를 사용하도록 인스턴스를 구성합니다.

문제 해결에 도움이 필요하면 Linux 인스턴스의 NTP 동기화 문제 해결 또는 Windows 인스턴스의 시간 문제 해결을 참조하세요.

Amazon Linux

AL2023 및 최신 버전의 Amazon Linux 2는 기본적으로 Amazon Time Sync Service IPv4 엔드포인트를 사용하도록 구성되어 있습니다. 인스턴스가 이미 구성되었음을 확인하면 다음 절차를 건너뛸 수 있습니다.

IPv4 엔드포인트를 사용하도록 chrony가 구성되어 있는지 확인하는 방법

다음 명령을 실행합니다. 출력에서 ^*로 시작되는 줄은 기본 시간 소스를 나타냅니다.

```
chronyc sources -v | grep -F ^*
^* 169.254.169.123               3   4   377    14    +12us[+9653ns] +/-  290us
```

이전 버전의 Amazon Linux 2에서 IPv4 엔드포인트에 연결하도록 chrony를 구성하는 방법

1. 인스턴스를 연결하고 NTP 서비스를 제거합니다.

   ```
   [ec2-user ~]$ sudo yum erase 'ntp*'
   ```

2. chrony 패키지를 설치합니다.

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

3. /etc/chrony.conf 또는 vim과 같은 텍스트 편집기를 사용하여 nano 파일을 엽니다. 파일에 이미 존재할 수 있는 server 또는 pool 문 앞에 다음 라인을 추가하고 변경 사항을 저장합니다.

```
server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
```

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. 각 시스템 부팅 시 chronyd가 시작되도록 구성하려면 chkconfig 명령을 사용합니다.

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

6. chrony가 169.254.169.123 IPv4 엔드포인트를 사용하여 시간을 동기화하고 있는지 확인합니다.

```
[ec2-user ~]$ chronyc sources -v | grep -F ^*
```

출력에서 ^*는 기본 시간 소스를 나타냅니다.

```
^* 169.254.169.123               3   6    17    43    -30us[ -226us] +/-  287us
```

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
```

Ubuntu

Ubuntu에서 IPv4 엔드포인트에 연결하도록 chrony를 구성하는 방법

1. 인스턴스를 연결해 apt를 사용하여 chrony 패키지를 설치합니다.

   ```
   ubuntu:~$ sudo apt install chrony
   ```

   > **Note**
   >
   > 필요할 경우 `sudo apt update`를 실행하여 먼저 인스턴스를 업데이트합니다.

2. /etc/chrony/chrony.conf 또는 vim과 같은 텍스트 편집기를 사용하여 nano 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
   ```

   ```
   Restarting chrony (via systemctl): chrony.service.
   ```

4. chrony가 169.254.169.123 IPv4 엔드포인트를 사용하여 시간을 동기화하고 있는지 확인합니다.

   ```
   ubuntu:~$ chronyc sources -v | grep -F ^*
   ```

   출력에서 ^*로 시작되는 줄은 기본 시간 소스를 나타냅니다.

   ```
   ^* 169.254.169.123               3   6    17    12    +15us[  +57us] +/-  320us
   ```

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 Linux

SUSE Linux Enterprise Server 15부터 chrony는 NTP의 기본 구현입니다.

SUSE Linux에서 IPv4 엔드포인트에 연결하도록 chrony를 구성하는 방법

1. /etc/chrony.conf 또는 vim과 같은 텍스트 편집기를 사용하여 nano 파일을 엽니다.
2. 파일이 다음 라인을 포함하고 있는지 확인합니다.

```
server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
```

   이 라인이 표시되지 않는 경우, 추가합니다.
3. 다른 서버 혹은 풀 라인 설명
4. YaST를 열고 chrony 서비스를 활성화하세요.

Windows

Windows AMI는 2018년 8월 릴리스부터 기본적으로 Amazon Time Sync Service를 사용합니다. 이러한 AMI에서 시작된 인스턴스에는 추가 구성이 필요하지 않으며 다음 절차를 건너뛰어도 됩니다.

기본적으로 Amazon Time Sync Service가 구성되지 않은 AMI를 사용하는 경우 먼저 현재 NTP 구성을 확인하세요. 인스턴스가 이미 Amazon Time Sync Service의 IPv4 엔드포인트를 사용하고 있는 경우 추가 구성이 필요하지 않습니다. 인스턴스가 Amazon Time Sync Service를 사용하지 않는 경우 Amazon Time Sync Service를 사용하도록 NTP 서버를 변경하는 절차를 완료합니다.

NTP 구성을 확인하려면

1. 인스턴스에서 명령 프롬프트 창을 엽니다.
2. 다음 명령을 입력하여 현재 NTP 구성을 가져옵니다.

   ```
   w32tm /query /configuration
   ```

   이 명령은 Windows 인스턴스에 대한 현재 구성 설정을 반환하고 Amazon Time Sync Service에 연결되어 있는지 보여줍니다.
3. (선택 사항) 다음 명령을 입력하여 현재 구성의 상태를 가져옵니다.

   ```
   w32tm /query /status
   ```

   이 명령은 인스턴스를 NTP 서버와 마지막으로 동기화한 시간, 폴링 간격 등의 정보를 반환합니다.

Amazon Time Sync Service를 사용하도록 NTP 서버를 변경하려면

1. 명령 프롬프트 창에서 다음 명령을 실행합니다.

   ```
   w32tm /config /manualpeerlist:169.254.169.123 /syncfromflags:manual /update
   ```
2. 다음 명령을 사용하여 새 설정을 확인합니다.

   ```
   w32tm /query /configuration
   ```

   반환된 출력에서 `NtpServer`에 `169.254.169.123` IPv4 엔드포인트가 표시되는지 확인합니다.

Amazon Windows AMI에 대한 기본 NTP 설정

일반적으로 Amazon Machine Image(AMI)는 EC2 인프라에서 함수 변경이 필요한 경우를 제외하고는 내장된 기본 설정을 그대로 유지합니다. 다음 설정은 가상 환경에서 잘 작동할 뿐 아니라 클럭 드리프트를 1초 이내의 정확도로 유지하는 것으로 판명되었습니다.

- 업데이트 간격 - 시간 서비스가 시스템 시간의 정확도를 조정하는 빈도를 제어합니다. AWS는 업데이트 간격을 2분에 한 번으로 구성합니다.
- NTP 서버 – AMI는 2018년 8월 릴리스부터 기본적으로 Amazon Time Sync Service를 사용합니다. 이 시간 서비스는 169.254.169.123 IPv4 엔드포인트에 있는 모든 AWS 리전에서 액세스할 수 있습니다. 뿐만 아니라 0x9 플래그는 시간 서비스가 클라이언트 역할을 하면서 `SpecialPollInterval`을 사용하여 구성된 시간 서버에 체크인하는 빈도를 파악함을 나타냅니다.
- 유형 – “NTP”는 서비스가 도메인 일부의 역할을 수행하는 대신 독립 실행형 NTP 클라이언트의 역할을 수행할 것임을 뜻합니다.
- 활성화 및 InputProvider – 시간 서비스가 활성화되어 운영 체제에 시간을 제공합니다.
- 특수 폴링 간격 – 900초(15분)마다 구성된 NTP 서버를 확인합니다.

레지스트리 경로	키 이름	Data
HKLM:\System\CurrentControlSet\services\w32time\Config	UpdateInterval	120
HKLM:\System\CurrentControlSet\services\w32time\Parameters	NtpServer	169.254.169.123,0x9
HKLM:\System\CurrentControlSet\services\w32time\Parameters	유형	NTP
HKLM:\System\CurrentControlSet\services\w32time\TimeProviders\NtpClient	활성	1

레지스트리 경로	키 이름	Data
HKLM:\System\CurrentControlSet\services\w32time\TimeProviders\NtpClient	InputProvider	1
HKLM:\System\CurrentControlSet\services\w32time\TimeProviders\NtpClient	SpecialPollInterval	900

Amazon Time Sync Service의 IPv6 엔드포인트에 연결합니다.

이 섹션에서는 IPv6 엔드포인트를 통해 로컬 Amazon Time Sync Service를 사용하도록 인스턴스를 구성하는 경우 Amazon Time Sync Service의 IPv4 엔드포인트에 연결합니다.에 설명된 단계가 어떻게 다른지 설명합니다. 전체 Amazon Time Sync Service 구성 프로세스를 설명하지는 않습니다.

IPv6 엔드포인트는 Nitro 기반 인스턴스에서만 액세스할 수 있습니다.

IPv4 및 IPv6 엔드포인트 항목을 함께 사용하는 것은 권장하지 않습니다. IPv4 및 IPv6 NTP 패킷은 인스턴스의 동일한 로컬 서버에서 가져옵니다. IPv4 엔드포인트와 IPv6 엔드포인트를 모두 구성할 필요는 없으며, 이렇게 해도 인스턴스의 시간 정확도가 향상되지 않습니다.

Linux

사용 중인 Linux 배포에 따라 `chrony.conf` 파일을 편집하는 단계에 도달하면 IPv4 엔드포인트(`169.254.169.123`) 대신 Amazon Time Sync Service(`fd00:ec2::123`)의 IPv6 엔드포인트를 사용합니다.

```
server fd00:ec2::123 prefer iburst minpoll 4 maxpoll 4
```

파일을 저장하고 chrony가 `fd00:ec2::123` IPv6 엔드포인트를 사용하여 시간을 동기화하는지 확인합니다.

```
[ec2-user ~]$ chronyc sources -v
```

출력에 `fd00:ec2::123` IPv6 엔드포인트가 표시될 경우 구성이 완료된 것입니다.

Windows

Amazon Time Sync Service를 사용하도록 NTP 서버를 변경하는 단계에 도달하면 IPv4 엔드포인트(169.254.169.123)가 아닌 Amazon Time Sync Service(fd00:ec2::123)의 IPv6 엔드포인트를 사용하게 됩니다.

```
w32tm /config /manualpeerlist:fd00:ec2::123 /syncfromflags:manual /update
```

새 설정이 fd00:ec2::123 IPv6 엔드포인트를 사용하여 시간을 동기화하고 있는지 확인합니다.

```
w32tm /query /configuration
```

출력에서 NtpServer가 fd00:ec2::123 IPv6 엔드포인트를 표시하는지 확인합니다.

PTP 하드웨어 클럭에 연결

PTP 하드웨어 클럭은 AWS Nitro 시스템의 일부이므로 고객 리소스를 사용하지 않고도 지원되는 베어 메탈 및 가상화된 EC2 인스턴스에서 직접 액세스할 수 있습니다.

PTP 하드웨어 클럭에 대한 NTP 엔드포인트는 일반 Amazon Time Sync Service의 것과 동일합니다. 인스턴스에 PTP 하드웨어 클럭이 있고 IPv4 또는 IPv6 엔드포인트에 대한 NTP 연결을 구성한 경우 인스턴스 시간은 NTP를 통해 PTP 하드웨어 클럭에서 자동으로 생성됩니다.

Linux 인스턴스의 경우 NTP 연결보다 더 정확한 시간을 제공하는 직접 PTP 연결을 구성할 수 있습니다. Windows 인스턴스는 PTP 하드웨어 클럭에 대한 NTP 연결만 지원합니다.

요구 사항

다음 요구 사항이 충족되면 인스턴스에서 PTP 하드웨어 클럭을 사용할 수 있습니다.

- 지원되는 AWS 리전: 미국 동부(버지니아 북부), 미국 동부(오하이오), 아시아 태평양(말레이시아), 아시아 태평양(태국), 아시아 태평양(도쿄), 유럽(스톡홀름)
- 지원되는 로컬 영역: 미국 동부(뉴욕)
- 지원되는 인스턴스 패밀리:
  - 범용: M7a, M7g, M7gd, M7i, M8g, M8gd
  - 컴퓨팅 최적화: C7a, C7gd, C7i, C8g, C8gd
  - 메모리 최적화: R7a, R7g, R7gd, R7i, R8g, R8gd, U7i-6tb, U7i-8tb, U7i-12tb, U7in-16tb, U7in-24tb, U7in-32tb, X8g

  - 스토리지 최적화: I7i, I7ie, I8g
  - 고성능 컴퓨팅: Hpc7a
- (Linux만 해당) 지원되는 운영 체제에 ENA 드라이버 버전 2.10.0 이상이 설치되어 있어야 합니다. 지원되는 운영 체제에 대한 자세한 내용은 GitHub의 드라이버 사전 조건을 참조하세요.

(Linux만 해당) PTP 하드웨어 클럭에 대한 직접 PTP 연결 구성

이 섹션에서는 PTP 직접 연결을 사용하여 PTP 하드웨어 클럭을 통해 로컬 Amazon Time Sync Service를 사용하도록 Linux 인스턴스를 구성하는 방법을 설명합니다. `chrony` 구성 파일에서 PTP 하드웨어 클럭에 대한 서버 항목을 추가해야 합니다.

PTP 하드웨어 클럭에 대한 직접 PTP 연결을 구성하려면(Linux 인스턴스만 해당)

1. 사전 조건 설치

   Linux 인스턴스에 연결하고 다음을 수행합니다.

   a. Elastic Network Adapter(ENA) 버전 2.10.0 이상용 Linux 커널 드라이버를 설치합니다.

   b. PTP 하드웨어 클럭을 활성화합니다.

   설치 지침은 GitHub의 Linux kernel driver for Elastic Network Adapter (ENA) family를 참조하세요.
2. ENA PTP 디바이스 확인

   인스턴스에 ENA PTP 하드웨어 클럭 디바이스가 표시되는지 확인하세요.

   ```
   [ec2-user ~]$ for file in /sys/class/ptp/*; do echo -n "$file: "; cat "$file/
   clock_name"; done
   ```

   예상 결과

   ```
   /sys/class/ptp/ptp<index>: ena-ptp-<PCI slot>
   ```

   위치:

   - *index*는 커널에 등록된 PTP 하드웨어 클럭 인덱스입니다.
   - *PCI slot*은 ENA 이더넷 컨트롤러 PCI 슬롯입니다. `lspci | grep ENA`에 표시된 것과 동일한 슬롯입니다.

출력 예시

```
/sys/class/ptp/ptp0: ena-ptp-05
```

ena-ptp-*<PCI slot>*이 출력에 없으면 ENA 드라이버가 제대로 설치되지 않은 것입니다. 드라이버 설치에 대한 이 절차의 1단계를 검토합니다.

3. PTP symlink 구성

PTP 디바이스 이름은 하드웨어 초기화 순서에 따라 인덱스가 지정되며 일반적으로 /dev/ptp0, /dev/ptp1 등입니다. symlink를 생성하면 chrony와 같은 애플리케이션이 인덱스 변경에 관계없이 올바른 디바이스를 일관되게 참조할 수 있습니다.

최신 Amazon Linux 2023 AMI에는 ENA 호스트에 연결된 올바른 /dev/ptp 항목을 가리키는 /dev/ptp_ena symlink를 생성하는 udev 규칙이 포함되어 있습니다.

먼저 다음 명령을 실행하여 symlink가 있는지 확인하세요.

```
[ec2-user ~]$ ls -l /dev/ptp*
```

출력 예시

```
crw------- 1 root root 245, 0 Jan 31 2025 /dev/ptp0
lrwxrwxrwx 1 root root      4 Jan 31 2025 /dev/ptp_ena -> ptp0
```

위치:

- /dev/ptp*<index>*는 PTP 디바이스에 대한 경로입니다.
- /dev/ptp_ena는 동일한 PTP 디바이스를 가리키는 상수 symlink입니다.

/dev/ptp_ena symlink가 있는 경우 이 절차의 4단계로 건너뛰세요. 누락된 경우 다음 중 하나를 수행하세요.

a. 다음 udev 규칙을 추가하세요.

```
[ec2-user ~]$ echo "SUBSYSTEM==\"ptp\", ATTR{clock_name}==\"ena-ptp-*\",
 SYMLINK += \"ptp_ena\"" | sudo tee -a /etc/udev/rules.d/53-ec2-network-
interfaces.rules
```

b. 인스턴스를 재부팅하거나 다음 명령을 실행하여 udev 규칙을 다시 로드하세요.

```
[ec2-user ~]$ sudo udevadm control --reload-rules && udevadm trigger
```

4. chrony 구성

/dev/ptp*<index>*를 직접 참조하는 대신 /dev/ptp_ena symlink를 사용하도록 chrony를 구성해야 합니다.

a. 텍스트 편집기를 사용하여 /etc/chrony.conf를 편집하고 파일에 다음 줄을 추가합니다.

```
refclock PHC /dev/ptp_ena poll 0 delay 0.000010 prefer
```

b. chrony를 다시 시작합니다.

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

5. chrony 구성 확인

chrony가 PTP 하드웨어 클럭을 사용하여 이 인스턴스의 시간을 동기화하고 있는지 확인합니다.

```
[ec2-user ~]$ chronyc sources
```

예상 결과

```
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
#* PHC0                          0   0    377    1   +2ns[ +1ns] +/-   5031ns
```

반환된 출력에서 *는 기본 시간 소스를 나타냅니다. PHC0은 PTP 하드웨어 클럭에 해당합니다. chrony를 다시 시작한 후 별표가 나타나려면 몇 초 정도 기다려야 할 수 있습니다.

퍼블릭 Amazon Time Sync Service를 사용하도록 EC2 인스턴스 또는 인터넷 연결 디바이스의 시간 참조 설정

인터넷을 통해 `time.aws.com`에 액세스할 수 있는 퍼블릭 Amazon Time Sync Service를 사용하도록 인스턴스나 로컬 컴퓨터, 온프레미스 서버 등 인터넷에 연결된 모든 디바이스를 설정할 수 있습니다. 퍼블릭 Amazon Time Sync Service를 로컬 Amazon Time Sync Service의 백업으로 사용하고 AWS 외부의 리소스를 Amazon Time Sync Service에 연결할 수 있습니다.

> **Note**
>
> 최상의 성능을 위해 인스턴스에서는 로컬 Amazon Time Sync Service를 사용하고 백업으로만 퍼블릭 Amazon Time Sync Service를 사용하는 것이 좋습니다.

인스턴스 또는 디바이스의 운영 체제에 대한 지침을 따르세요.

Linux

chrony 또는 ntpd로 퍼블릭 Amazon Time Sync Service를 사용하도록 Linux 인스턴스 또는 디바이스 설정

1. 다음과 같은 텍스트 편집기를 사용하여 `/etc/chrony.conf`(chrony를 사용하는 경우) 또는 `/etc/ntp.conf`(ntpd를 사용하는 경우)를 편집합니다.

   a. 인스턴스나 디바이스에서 스미어링한 서버와 스미어링하지 않은 서버를 혼용하지 않도록 하려면 로컬 Amazon Time Sync Service에 대한 기존 연결을 제외하고 `server`로 시작하는 줄을 제거하거나 주석 처리합니다.

   > **Important**
   >
   > 퍼블릭 Amazon Time Sync Service에 연결하도록 EC2 인스턴스를 설정하는 경우 로컬 Amazon Time Sync Service에 연결하도록 인스턴스를 설정하는 다음 줄을 제거하지 마세요. 로컬 Amazon Time Sync Service는 보다 직접적인 연결이며 더 나은 클럭 정확도를 제공합니다. 퍼블릭 Amazon Time Sync Service는 백업으로만 사용해야 합니다.
   >
   > ```
   > server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
   > ```

b. 퍼블릭 Amazon Time Sync Service에 연결하려면 다음 줄을 추가합니다.

```
pool time.aws.com iburst
```

2. 다음 명령 중 하나를 사용하여 대몬(daemon)을 다시 시작합니다.

- chrony

```
sudo service chronyd force-reload
```

- ntpd

```
sudo service ntp reload
```

macOS

퍼블릭 Amazon Time Sync Service를 사용하도록 macOS 인스턴스 또는 디바이스 설정

1. 시스템 환경 설정를 엽니다.
2. Date & Time(날짜 및 시간)을 선택한 다음 Date & Time(날짜 및 시간) 탭을 선택합니다.
3. 변경하려면 잠금 아이콘을 선택하고 암호를 묻는 메시지가 표시되면 암호를 입력합니다.
4. Set date and time automatically(날짜 및 시간 자동 설정)에 **time.aws.com**을 입력합니다.

Windows

퍼블릭 Amazon Time Sync Service를 사용하도록 Windows 인스턴스 또는 디바이스 설정

1. 제어판을 엽니다.
2. 날짜 및 시간 아이콘을 선택합니다.
3. 인터넷 시간 탭을 선택합니다. PC가 도메인에 속해 있는 경우에는 이 탭을 사용할 수 없습니다. 이 경우 도메인 컨트롤러와 시간이 동기화됩니다. 퍼블릭 Amazon Time Sync Service를 사용하도록 컨트롤러를 구성할 수 있습니다.
4. 설정 변경을 선택합니다.
5. 인터넷 시간 서버와 동기화 확인란을 선택합니다.
6. 서버 옆에 **time.aws.com**을 입력합니다.

퍼블릭 Amazon Time Sync Service를 사용하도록 Windows Server 인스턴스 또는 디바이스 설정

- Microsoft의 지침에 따라 레지스트리를 업데이트합니다.

Linux 인스턴스의 타임스탬프 비교

Amazon Time Sync Service를 사용하는 경우 Amazon EC2 Linux 인스턴스의 타임스탬프를 ClockBound와 비교하여 이벤트의 실제 시간을 결정할 수 있습니다. ClockBound는 EC2 인스턴스의 클록 정확도를 측정하고 지정된 타임스탬프가 인스턴스의 현재 클록과 관련하여 과거인지 미래인지 확인할 수 있도록 합니다. 이 정보는 각 인스턴스의 지리적 위치에 관계없이 EC2 인스턴스 전체에서 이벤트 및 트랜잭션의 순서와 일관성을 결정하는 데 유용합니다.

ClockBond는 오픈 소스 데몬과 라이브러리입니다. 설치 지침을 포함하여 ClockBound에 대한 자세한 내용은 GitHub의 ClockBound를 참조하세요.

ClockBound는 Linux 인스턴스에서만 지원됩니다.

PTP 하드웨어 클럭에 대한 직접 PTP 연결을 사용하는 경우 chrony와 같은 시간 대몬(daemon)은 클럭 오차 범위를 과소평가합니다. 이는 PTP 하드웨어 클럭이 NTP와 같은 방식으로 올바른 오차 범위 정보를 chrony에 전달하지 않기 때문입니다. 따라서 시계 동기화 대몬(daemon)은 클럭이 UTC에 정확하다고 가정하므로 오차 범위가 `0`입니다. 전체 오차 범위를 측정하기 위해 Nitro System은 PTP 하드웨어 클럭의 오차 범위를 계산하고 이를 ENA 드라이버 `sysfs` 파일 시스템을 통해 EC2 인스턴스에서 사용할 수 있도록 지원합니다. 나노초 단위의 값으로 직접 읽을 수 있습니다.

PTP 하드웨어 클럭 오류 범위를 검색하는 방법

1. 먼저 다음 명령 중 하나를 사용하여 PTP 하드웨어 클록 디바이스의 올바른 위치를 가져옵니다. 명령의 경로는 인스턴스를 시작할 때 사용한 AMI에 따라 달라집니다.

   - 대상 Amazon Linux 2:

     ```
     cat /sys/class/net/eth0/device/uevent | grep PCI_SLOT_NAME
     ```

   - Amazon Linux 2023의 경우:

     ```
     cat /sys/class/net/ens5/device/uevent | grep PCI_SLOT_NAME
     ```

   출력은 PTP 하드웨어 클럭 디바이스의 위치인 PCI 슬롯 이름입니다. 이 예제에서 위치는 `0000:00:03.0`입니다.

```
PCI_SLOT_NAME=0000:00:03.0
```

2. PTP 하드웨어 클럭 오차 범위를 검색하려면 다음 명령을 실행합니다. 이전 단계의 PCI 슬롯 이름을 포함합니다.

```
cat /sys/bus/pci/devices/0000:00:03.0/phc_error_bound
```

출력은 PTP 하드웨어 클럭의 클럭 오차 범위(나노초)입니다.

PTP 하드웨어 클럭에 대한 직접 PTP 연결을 사용할 때 특정 시점의 올바른 클럭 오차 범위를 계산하려면 chrony가 PTP 하드웨어 클럭을 폴링하는 시간에 chrony 또는 ClockBound에서 바인딩된 클럭 오차를 추가해야 합니다. 클럭 정확도 측정 및 모니터링에 대한 자세한 내용은 Manage Amazon EC2 instance clock accuracy using Amazon Time Sync Service and Amazon CloudWatch – Part 1을 참조하세요.

인스턴스의 시간대 변경

Amazon EC2 인스턴스는 기본적으로 UTC(협정 세계시) 표준 시간대로 설정됩니다. 인스턴스의 시간을 현지 시간대나 네트워크의 다른 시간대로 변경할 수 있습니다.

인스턴스 운영 체제에 대한 지침을 사용하세요.

Linux

> **Important**
>
> 이 정보는 Amazon Linux에 적용됩니다. 기타 배포에 대한 자세한 내용은 해당 설명서를 참조하세요.

Amazon Linux의 표준 시간대를 변경하려면

1. 시스템의 현재 표준 시간대 설정을 확인합니다.

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

2. 사용 가능한 표준 시간대를 나열합니다.

```
[ec2-user ~]$ timedatectl list-timezones
```

3. 선택한 표준 시간대를 설정합니다.

```
[ec2-user ~]$ sudo timedatectl set-timezone America/Vancouver
```

4. (선택 사항) timedatectl 명령을 다시 실행하여 현재 표준 시간대가 새 표준 시간대로 업데이트되는지 확인합니다.

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

Windows

Windows 인스턴스의 시간대 변경

1. 인스턴스에서 명령 프롬프트 창을 엽니다.
2. 인스턴스에서 사용할 표준 시간대를 식별합니다. 표준 시간대 목록을 가져오려면 다음 명령을 사용합니다.

```
tzutil /l
```

이 명령은 사용 가능한 모든 시간대의 목록을 다음 형식으로 반환합니다.

```
display name
time zone ID
```

3. 인스턴스에 배정할 표준 시간대 ID를 찾습니다.
4. 예: UTC 표준 시간대 할당:

```
tzutil /s "UTC"
```

예: 태평양 표준시 할당:

```
tzutil /s "Pacific Standard Time"
```

Windows 인스턴스의 시간대를 변경할 경우 시스템을 다시 시작해도 시간대가 계속 유지되는지 확인해야 합니다. 그렇게 하지 않으면 인스턴스가 다시 시작될 때 UTC 시간으로 다시 돌아갑니다. RealTimeIsUniversal 레지스트리 키를 추가하여 시간대 설정을 계속 유지할 수 있습니다. 이 키는 모든 현재 세대 인스턴스에서 기본적으로 설정됩니다. RealTimeIsUniversal 레지스트리 키가 설정되어 있는지 확인하려면 다음 절차의 3단계를 참조하세요. 키가 설정되지 않은 경우 처음부터 다음 단계를 수행하세요.

RealTimeIsUniversal 레지스트리 키를 설정하려면

1. 인스턴스에서 명령 프롬프트 창을 엽니다.
2. 다음 명령을 사용하여 레지스트리 키를 추가합니다.

```
reg add "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\TimeZoneInformation" /
v RealTimeIsUniversal /d 1 /t REG_DWORD /f
```

3. (선택 사항) 다음 명령을 사용하여 인스턴스에서 키를 성공적으로 저장했는지 확인합니다.

```
reg query "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control
\TimeZoneInformation" /s
```

이 명령은 TimeZoneInformation 레지스트리 키에 대한 하위 키를 반환합니다. 목록의 맨 아래에 RealTimeIsUniversal 키가 다음과 비슷하게 표시되어야 합니다.

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\TimeZoneInformation
    Bias                            REG_DWORD     0x1e0
    DaylightBias                    REG_DWORD     0xfffffc4
    DaylightName                    REG_SZ        @tzres.dll,-211
    DaylightStart                   REG_BINARY    00000300020002000000000000000000
    StandardBias                    REG_DWORD     0x0
    StandardName                    REG_SZ        @tzres.dll,-212
    StandardStart                   REG_BINARY    00000B0001000200000000000000000
    TimeZoneKeyName                 REG_SZ        Pacific Standard Time
    DynamicDaylightTimeDisabled     REG_DWORD     0x0
    ActiveTimeBias                  REG_DWORD     0x1a4
    RealTimeIsUniversal             REG_DWORD     0x1
```

EC2 인스턴스의 디바이스 드라이버 관리

디바이스 드라이버는 Amazon EC2 인스턴스의 가상화된 하드웨어와 통신하는 소프트웨어 구성 요소입니다. 시스템 오류, 성능 문제 및 기타 예상치 못한 동작을 방지하려면 드라이버를 최신 상태로 유지하는 것이 중요합니다. 네트워킹, 그래픽, 스토리지 드라이버와 같이 사용량에 따라 시스템 성능에 큰 영향을 미칠 수 있는 드라이버의 경우 특히 최신 상태로 유지해야 합니다. 새 드라이버 릴리스에는 결함 수정이 포함되거나 현재 실행 중인 인스턴스에 활용할 수 있는 기능 확장이 도입될 수 있습니다.

네트워크 드라이버

Linux 배포판은 Elastic Network Adapter(ENA) 또는 Elastic Fabric Adapter(EFA) 같은 네트워킹 기능을 커널 내에 통합할 수 있습니다. 하지만 배포판마다 커널 드라이버 기능을 구현하는 시기가 달라질 수 있습니다.

ENA 및 EFA Linux 커널 드라이버는 Amazon 드라이버 GitHub 리포지토리에 제공됩니다. 사용 가능한 드라이버에 대한 자세한 내용 및 링크는 GitHub의 Amazon Drivers를 참조하세요.

ENA 드라이버에 대한 자세한 내용은 EC2 인스턴스에서 ENA로 향상된 네트워킹 활성화의 내용을 참조하세요. EFA 드라이버에 대한 자세한 내용은 이 설명서의 Amazon EC2의 AI/ML 및 HPC 워크로드를 위한 Elastic Fabric Adapter 섹션에 있는 시작하기 주제를 참조하세요.

Windows 인스턴스의 네트워킹 드라이버를 설치하거나 업데이트하려면 다음 주제를 참조하세요.

- Windows에 ENA 드라이버 설치
- 최신 AWS PV 드라이버 설치

  자세한 내용은 Windows 인스턴스의 반가상화 드라이버 섹션을 참조하세요.

> **Note**
>
> Windows 인스턴스에서는 EFA가 지원되지 않습니다.

그래픽 드라이버

그래픽 드라이버를 설치하거나 업데이트하려면 다음 주제를 참조하세요.

- EC2 인스턴스용 AMD 드라이버
- Amazon EC2 인스턴스용 NVIDIA 드라이버

스토리지 디바이스 드라이버

스토리지 드라이버를 설치하거나 업데이트하려면 다음 주제를 참조하세요.

- Linux 인스턴스의 경우 Amazon EBS 사용 설명서의 NVMe 드라이버 설치 또는 업그레이드를 참조하세요.
- Windows 인스턴스의 경우 AWS NVMe 드라이버의 내용을 참조하세요.

EC2 인스턴스용 AMD 드라이버

G4ad 인스턴스와 같이 연결된 AMD GPU가 있는 인스턴스에는 적절한 AMD 드라이버가 설치되어 있어야 합니다. 요구 사항에 따라 드라이버가 사전 설치된 AMI를 사용하거나 Amazon S3에서 드라이버를 다운로드할 수 있습니다.

G4dn 인스턴스와 같이 연결된 NVIDIA GPU가 있는 인스턴스에 NVIDIA 드라이버를 설치하려면 NVIDIA 드라이버 섹션을 참조하세요.

목차

- AMD Radeon Pro Software for Enterprise Driver
- AMD 드라이버가 설치된 AMI
- AMD 드라이버 다운로드

AMD Radeon Pro Software for Enterprise Driver

AMD Radeon Pro Software for Enterprise Driver는 전문가 등급의 그래픽 사용 사례를 지원하도록 구축되었습니다. 이 드라이버를 사용하면 GPU당 2개의 4K 디스플레이로 인스턴스를 구성할 수 있습니다.

지원되는 API

- OpenGL, OpenCL
- Vulkan
- AMD Advanced Media Framework
- Video Acceleration API
- DirectX 9 이상

- Microsoft Hardware Media Foundation Transform

AMD 드라이버가 설치된 AMI

AWS는 AMD 드라이버가 설치되어 있는 다양한 Amazon Machine Image(AMI)를 제공합니다. AMD 드라이버가 포함된 Marketplace 제품을 엽니다.

AMD 드라이버 다운로드

AMD 드라이버가 설치된 AMI를 사용하지 않는 경우 AMD 드라이버를 다운로드하여 인스턴스에 설치할 수 있습니다. 다음 운영 체제 버전만 AMD 드라이버를 지원합니다.

- 커널 버전 5.4가 포함된 Amazon Linux 2
- Ubuntu 20.04
- Ubuntu 22.04
- Ubuntu 24.04
- Windows Server 2016
- Windows Server 2019
- Windows Server 2022

이러한 다운로드는 AWS 고객만 사용할 수 있습니다. 드라이버를 다운로드하면 AMD Radeon Pro V520 하드웨어와 함께 사용할 목적으로 AMI를 개발하는 데에만 다운로드한 소프트웨어를 사용한다는 것에 동의하게 됩니다. 소프트웨어를 설치하면 AMD 소프트웨어 최종 사용자 라이선스 계약의 약관이 적용됩니다.

Amazon Linux 2 Linux 인스턴스에 AMD 드라이버 설치

1. Linux 인스턴스에 연결합니다.
2. Linux 인스턴스에 AWS CLI를 설치하고 기본 자격 증명을 구성합니다. 자세한 내용은 AWS Command Line Interface 사용 설명서에서 AWS CLI 설치를 참조하세요.

> **Important**
>
> 사용자 또는 역할에 AmazonS3ReadOnlyAccess 정책이 포함된 권한이 부여되어야 합니다. 자세한 내용을 알아보려면 Amazon Simple Storage Service 사용 설명서의 AWS 관리형 정책: AmazonS3ReadOnlyAccess를 참조하세요.

3. 커널 5.4 설치

```
$ sudo amazon-linux-extras disable kernel-5.10
$ sudo amazon-linux-extras enable kernel-5.4
$ sudo yum install -y kernel
```

4. gcc 및 make를 설치합니다(아직 설치되지 않은 경우).

```
$ sudo yum install gcc make
```

5. 패키지 캐시를 업데이트하고 인스턴스에 대한 패키지 업데이트를 가져옵니다.

```
$ sudo amazon-linux-extras install epel -y
$ sudo yum update -y
```

6. 인스턴스를 재부팅합니다.

```
$ sudo reboot
```

7. 재부팅된 후 인스턴스에 다시 연결합니다.
8. 최신 AMD 드라이버를 다운로드합니다.

```
$ aws s3 cp --recursive s3://ec2-amd-linux-drivers/latest/ .
```

9. 파일의 압축을 풉니다.

```
$ tar -xf amdgpu-pro-*rhel*.tar.xz
```

10. 압축을 푼 드라이버의 폴더로 변경합니다.
11. 자체 설치 스크립트를 실행하여 전체 그래픽 스택을 설치합니다.

```
$ ./amdgpu-pro-install -y --opencl=pal,legacy
```

12. 인스턴스를 재부팅합니다.

```
$ sudo reboot
```

13. 드라이버가 작동하는지 확인합니다.

```
$ sudo dmesg | grep amdgpu
```

응답은 다음과 같아야 합니다.

```
Initialized amdgpu
```

Ubuntu Linux 인스턴스에 AMD 드라이버 설치

1. Linux 인스턴스에 연결합니다.
2. 패키지 캐시를 업데이트하고 인스턴스에 대한 패키지 업데이트를 가져옵니다.

```
$ sudo apt-get update --fix-missing && sudo apt-get upgrade -y
```

3. gcc 및 make를 설치합니다(아직 설치되지 않은 경우).

```
$ sudo apt install build-essential -y
```

4. Linux 펌웨어 및 커널 모듈 설치

```
$ sudo apt install linux-firmware linux-modules-extra-aws -y
```

5. 인스턴스 재부팅

```
$ sudo reboot
```

6. 재부팅된 후 인스턴스에 다시 연결합니다.
7. AMD Linux 드라이버 패키지 설치

   - Ubuntu 20.04의 경우:

```
$ wget https://repo.radeon.com/.preview/afe3e25b8f1beff0bb312e27924d63b5/amdgpu-install/5.4.02.01/ubuntu/focal/amdgpu-install_5.4.02.01.50402-1_all.deb
$ sudo dpkg --add-architecture i386
$ sudo apt install ./amdgpu-install_5.4.02.01.50402-1_all.deb
```

   - 이후 Ubuntu 버전은 Linux® Drivers for AMD Radeon™ Graphics로 이동하고 최신 Ubuntu 패키지를 다운로드하여 설치합니다.

```
$ sudo apt install ./amdgpu-install_{version-you-downloaded}.deb
```

8. 자체 설치 스크립트를 실행하여 전체 그래픽 스택을 설치합니다.

```
$ amdgpu-install --usecase=workstation --vulkan=pro -y
```

9. 인스턴스를 재부팅합니다.

```
$ sudo reboot
```

10. 드라이버가 작동하는지 확인합니다.

```
$ sudo dmesg | grep amdgpu
```

응답은 다음과 같아야 합니다.

```
Initialized amdgpu
```

Windows 인스턴스에 AMD 드라이버 설치

1. Windows 인스턴스에 연결하고 PowerShell 창을 엽니다.
2. Windows 인스턴스에서 AWS Tools for Windows PowerShell에 대한 기본 자격 증명을 구성합니다. 자세한 내용은 AWS Tools for Windows PowerShell 사용 설명서에서 AWS Tools for Windows PowerShell 시작하기를 참조하세요.

> **Important**
>
> 사용자 또는 역할에 AmazonS3ReadOnlyAccess 정책이 포함된 권한이 부여되어야 합니다. 자세한 내용을 알아보려면 Amazon Simple Storage Service 사용 설명서의 AWS 관리형 정책: AmazonS3ReadOnlyAccess를 참조하세요.

3. 사용 중인 Windows 버전에 따라 키 접두사를 설정합니다.

   - Windows 10 및 Windows 11

   ```
   $KeyPrefix = "latest/AMD_GPU_WINDOWS10"
   ```

   - Windows Server 2016

   ```
   $KeyPrefix = "archives"
   ```

   - Windows Server 2019

```
$KeyPrefix = "latest/AMD_GPU_WINDOWS_2K19" # use "archives" for Windows Server
 2016
```

- Windows Server 2022

```
$KeyPrefix = "latest/AMD_GPU_WINDOWS_2K22"
```

4. 다음 PowerShell 명령을 사용하여 Amazon S3 드라이버를 데스크톱에 다운로드합니다.

```
$Bucket = "ec2-amd-windows-drivers"
$LocalPath = "$home\Desktop\AMD"
$Objects = Get-S3Object -BucketName $Bucket -KeyPrefix $KeyPrefix -Region us-east-1
foreach ($Object in $Objects) {
$LocalFileName = $Object.Key
if ($LocalFileName -ne '' -and $Object.Size -ne 0) {
    $LocalFilePath = Join-Path $LocalPath $LocalFileName
    Copy-S3Object -BucketName $Bucket -Key $Object.Key -LocalFile $LocalFilePath -
Region us-east-1
    }
}
```

5. 다운로드한 드라이버 파일의 압축을 풀고 다음 PowerShell 명령을 사용하여 설치 프로그램을 실행합니다.

```
Expand-Archive $LocalFilePath -DestinationPath "$home\Desktop\AMD\$KeyPrefix" -
Verbose
```

다음으로, 새 디렉터리의 콘텐츠를 확인합니다. 디렉터리 이름은 `Get-ChildItem` PowerShell 명령을 사용하여 검색할 수 있습니다.

```
Get-ChildItem "$home\Desktop\AMD\$KeyPrefix"
```

다음과 유사하게 출력됩니다.

```
Directory: C:\Users\Administrator\Desktop\AMD\latest

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----       10/13/2021  12:52 AM                210414a-365562C-Retail_End_User.2
```

드라이버 설치:

```
pnputil /add-driver $home\Desktop\AMD\$KeyPrefix\*.inf /install /subdirs
```

6. 안내에 따라 드라이버를 설치하고 필요에 따라 인스턴스를 재부팅합니다.
7. GPU가 제대로 작동하는지 확인하려면 장치 관리자를 확인합니다. "AMD Radeon Pro V520 MxGPU"가 디스플레이어 어댑터로 나열됩니다.
8. 최대 4K 해상도의 디스플레이 4개를 활용하는 데 도움이 되도록 고성능 디스플레이 프로토콜인 Amazon DCV를 설정합니다.

Amazon EC2 인스턴스용 NVIDIA 드라이버

P3 또는 G4dn 인스턴스와 같이 연결된 NVIDIA 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.

G4ad 인스턴스와 같이 연결된 AMD GPU가 있는 인스턴스에 AMD 드라이버를 설치하려면 AMD 드라이버 섹션을 참조하세요.

목차

- NVIDIA 드라이버의 유형
- 인스턴스 유형별로 사용 가능한 드라이버
- 설치 옵션
  - 옵션 1: NVIDIA 드라이버가 설치된 AMI
  - 옵션 2: 퍼블릭 NVIDIA 드라이버
  - 옵션 3: GRID 드라이버(G6, Gr6, G6e, G5, G4dn, G3 인스턴스)
  - 옵션 4: NVIDIA 게임 드라이버(G4dn, G5, G6 및 G6e 인스턴스)
- CUDA의 추가 버전 설치

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.

구성된 모드

Windows에서 Tesla 드라이버는 TCC(Tesla 컴퓨팅 클러스터) 모드에서 실행되도록 구성됩니다. GRID 및 게임 드라이버는 WDDM(Windows 디스플레이 드라이버 모델) 모드에서 실행되도록 구성됩니다. TCC 모드에서 이 카드는 컴퓨팅 워크로드에 전용됩니다. WDDM 모드에서 이 카드는 컴퓨팅 워크로드와 그래픽 워크로드를 모두 지원합니다.

NVIDIA 제어판

NVIDIA 제어판은 GRID 및 게임 드라이버에서 지원됩니다. Tesla 드라이버에서는 지원되지 않습니다.

Tesla, GRID, 게임 드라이버에 대해 지원되는 API

- OpenCL, OpenGL 및 Vulkan
- NVIDIA CUDA 및 관련 라이브러리(예: cuDNN, TensorRT, nvJPEG, cuBLAS)
- 비디오 인코딩용 NVENC 및 비디오 디코딩용 NVDEC
- Windows 전용 API: DirectX, Direct2D, DirectX Video Acceleration, DirectX Raytracing

인스턴스 유형별로 사용 가능한 드라이버

다음 표에는 각 GPU 인스턴스 유형에 지원되는 NVIDIA 드라이버가 요약되어 있습니다.

인스턴스 유형	Tesla 드라이버	GRID 드라이버	게임 드라이버
G3	예	예	아니요
G4dn	예	예	예
G5	예	예	예
G5g	예 [1]	아니요	아니요
G6	예	예	예
G6e	예	예	예
Gr6	예	예	아니요
P2	예	아니요	아니요
P3	예	아니요	아니요
P4d	예	아니요	아니요
P4de	예	아니요	아니요
P5	예	아니요	아니요
P5e	예	아니요	아니요
P5en	예	아니요	아니요
P6-B200	예	아니요	아니요

[1] 이 Tesla 드라이버는 ARM64 플랫폼과 관련된 최적화된 그래픽 어플리케이션도 지원합니다.

[2] Marketplace AMI만 사용

설치 옵션

다음 옵션 중 하나를 사용하여 GPU 인스턴스에 필요한 NVIDIA 드라이버를 가져옵니다.

옵션

- 옵션 1: NVIDIA 드라이버가 설치된 AMI
- 옵션 2: 퍼블릭 NVIDIA 드라이버
- 옵션 3: GRID 드라이버(G6, Gr6, G6e, G5, G4dn, G3 인스턴스)
- 옵션 4: NVIDIA 게임 드라이버(G4dn, G5, G6 및 G6e 인스턴스)

옵션 1: NVIDIA 드라이버가 설치된 AMI

AWS와 NVIDIA는 NVIDIA 드라이버가 설치된 상태로 제공되는 다양한 Amazon Machine Image(AMI)를 제공합니다.

- Tesla 드라이버와 함께 제공되는 Marketplace 상품
- GRID 드라이버와 함께 제공되는 Marketplace 상품
- 게임 드라이버와 함께 제공되는 Marketplace 상품

운영 체제(OS) 플랫폼별 고려 사항을 검토하려면 AMI에 해당하는 탭을 선택합니다.

Linux

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.

Windows

AWS Marketplace 오퍼링 중 하나를 사용하여 사용자 지정 Windows AMI를 만드는 경우 GRID 드라이버가 작동하도록 하기 위해서는 AMI는 Windows Sysprep으로 만든 표준화된 이미지여야 합니다. 자세한 내용은 Windows Sysprep을 사용하여 Amazon EC2 AMI 생성 섹션을 참조하세요.

옵션 2: 퍼블릭 NVIDIA 드라이버

AWS에서 제공하는 옵션은 드라이버에 필요한 라이선스와 함께 제공됩니다. 또는 퍼블릭 드라이버를 설치하고 기존 보유 라이선스를 사용할 수도 있습니다. 퍼블릭 드라이버를 설치하려면 여기에 설명된 대로 NVIDIA 사이트에서 다운로드합니다.

또는 퍼블릭 드라이버 대신 AWS에서 제공하는 옵션을 사용할 수 있습니다. P3 인스턴스에서 GRID 드라이버를 사용하려면 옵션 1에 설명된 대로 AWS Marketplace AMI를 사용합니다. G6, G6e, Gr6, G5, G4dn 또는 G3 인스턴스에서 GRID 드라이버를 사용하려면 옵션 1에 설명된 대로 AWS Marketplace AMI를 사용하거나 옵션 3: GRID 드라이버(G6, Gr6, G6e, G5, G4dn, G3 인스턴스)에 설명된 대로 AWS에서 제공하는 NVIDIA 드라이버를 설치합니다.

퍼블릭 NVIDIA 드라이버를 다운로드하려면

인스턴스에 로그인하고 http://www.nvidia.com/Download/Find.aspx에서 인스턴스 유형에 적합한 64비트 NVIDIA 드라이버를 다운로드할 수 있습니다. 제품 유형, 제품 시리즈 및 제품에 다음 표의 옵션을 사용합니다.

인스턴스	제품 유형	제품 시리즈	제품	최소 드라이버 버전
G3	Tesla	M-Class	M60	--
G4dn	Tesla	T 시리즈	T4	--
G5	Tesla	A 시리즈	A10	470.00 이상
G5g [1]	Tesla	T 시리즈	NVIDIA T4G	470.82.01 이상
G6	Tesla	L 시리즈	L4	525.0 이상
G6e	Tesla	L 시리즈	L40S	535.0 이상
Gr6	Tesla	L 시리즈	L4	525.0 이상
P2	Tesla	K 시리즈	K80	--
P3	Tesla	V 시리즈	V100	--
P4d	Tesla	A 시리즈	A100	--

인스턴스	제품 유형	제품 시리즈	제품	최소 드라이버 버전
P4de	Tesla	A 시리즈	A100	--
P5	Tesla	H-시리즈	H100	530 이상
P5e	Tesla	H-시리즈	H200	550 이상
P5en	Tesla	H-시리즈	H200	550 이상
P6-B200	Tesla	B 시리즈	B200	570 이상

[1] G5g 인스턴스의 운영 체제는 Linux aarch64입니다.

Linux 운영 체제에 NVIDIA 드라이버를 설치하려면 NVIDIA 드라이버 설치 빠른 시작 가이드를 참조하세요.

Windows에 NVIDIA 드라이버를 설치하려면 다음 단계를 수행합니다.

1. 드라이버를 다운로드한 폴더를 열고 설치 파일을 실행합니다. 안내에 따라 드라이버를 설치하고 필요에 따라 인스턴스를 재부팅합니다.
2. 디바이스 관리자를 사용하여 경고 아이콘이 표시된 Microsoft Basic Display Adapter라는 디스플레이 어댑터를 비활성화합니다. Windows 기능인 미디어 파운데이션 및 qWave(Quality Windows Audio Video Experience)를 설치합니다.

   > **Important**
   >
   > Microsoft Remote Display Adapter라는 디스플레이 어댑터를 비활성화하지 마세요. Microsoft Remote Display Adapter가 비활성화된 경우 연결이 중단되고 재부팅된 후 인스턴스에 연결하려는 시도가 실패할 수 있습니다.

3. GPU가 올바르게 작동하는지 확인하려면 장치 관리자를 확인합니다.
4. GPU에서 최상의 성능을 얻으려면 Amazon EC2 인스턴스의 GPU 설정 최적화의 최적화 단계를 완료합니다.

옵션 3: GRID 드라이버(G6, Gr6, G6e, G5, G4dn, G3 인스턴스)

이러한 다운로드는 AWS 고객만 사용할 수 있습니다. 이 소프트웨어를 다운로드할 경우 NVIDIA GRID Cloud 최종 사용자 라이선스 계약(EULA)에 언급된 AWS 솔루션의 요구 사항을 준수하기 위해 NVIDIA L4, NVIDIA A10G, NVIDIA Tesla T4 또는 NVIDIA Tesla M60 하드웨어와 함께 사용할 AMI를 개발하는 용도로만 다운로드한 소프트웨어를 사용하는 데 동의한 것으로 간주됩니다. 소프트웨어를 설치하면 NVIDIA GRID 클라우드 최종 사용자 라이선스 계약의 약관이 적용됩니다. 운영 체제의 NVIDIA GRID 드라이버 버전에 대한 자세한 내용은 NVIDIA 웹 사이트의 NVIDIA 가상 GPU(vGPU) 소프트웨어를 참조하세요.

고려 사항

- G6e 인스턴스에는 GRID 17.4 이상이 필요합니다.
- G6 및 Gr6 인스턴스에는 GRID 17.1 이상이 필요합니다.
- G5 인스턴스에는 GRID 13.1 이상이나 GRID 12.4 이상이 필요합니다.
- Grid 라이선싱이 작동하려면 AWS에서 제공하는 DNS 확인이 G3 인스턴스에 필요합니다.
- IMDSv2는 NVIDIA 드라이버 버전 14.0 이상에서만 지원됩니다.
- Windows 인스턴스의 경우 사용자 지정 Windows AMI에서 인스턴스를 시작하면 GRID 드라이버 작동을 보장하기 위해 AMI가 Windows Sysprep으로 생성된 표준화된 이미지여야 합니다. 자세한 내용은 Windows Sysprep을 사용하여 Amazon EC2 AMI 생성 섹션을 참조하세요.
- GRID 17.0 이상은 Windows Server 2019를 지원하지 않습니다.
- GRID 14.2 이상은 Windows Server 2016을 지원하지 않습니다.
- GRID 17.0 이상은 G3 인스턴스에서 지원되지 않습니다.
- Linux 인스턴스의 경우 NVIDIA 설치 관리자가 오류 메시지와 함께 실패하면 gcc와 같은 패키지를 설치하거나 업데이트해야 할 수 있습니다. 세부 정보는 운영 체제의 버전 및 커널에 따라 다릅니다. 자세한 내용은 NVIDIA Enterprise Support 포털을 참조하세요.

사전 조건

- (Linux) 인스턴스에 AWS CLI가 설치되고 기본 자격 증명으로 구성되어 있는지 확인합니다. 자세한 내용을 알아보려면 AWS Command Line Interface 사용자 가이드에서 AWS CLI 설치를 참조하세요.
- (Windows) 인스턴스에서 AWS Tools for Windows PowerShell에 대한 기본 자격 증명을 구성합니다. 자세한 내용은 AWS Tools for Windows PowerShell 사용 설명서에서 AWS Tools for Windows PowerShell 시작하기를 참조하세요.
- 사용자 또는 역할에 AmazonS3ReadOnlyAccess 정책이 포함된 권한이 부여되어야 합니다.

Amazon Linux 2023

인스턴스에 NVIDIA GRID 드라이버 설치

1. 인스턴스에 연결합니다. 패키지 캐시를 업데이트하고 인스턴스에 대한 패키지 업데이트를 가져옵니다.

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

2. gcc 및 make를 설치합니다(아직 설치되지 않은 경우).

```
[ec2-user ~]$ sudo dnf install gcc make
```

3. 인스턴스를 재부팅하여 최신 커널 버전을 로드합니다.

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

4. 재부팅이 끝난 후 인스턴스에 다시 연결합니다.
5. 커널 헤더 패키지를 설치합니다.

```
[ec2-user ~]$ sudo dnf install -y kernel-devel kernel-modules-extra
```

6. 다음 명령을 사용하여 GRID 드라이버 설치 유틸리티를 다운로드합니다.

```
[ec2-user ~]$ aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/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/
```

7. 다음 명령을 사용하여 드라이버 설치 유틸리티를 실행할 수 있는 권한을 추가합니다.

```
[ec2-user ~]$ chmod +x NVIDIA-Linux-x86_64*.run
```

8. 다음과 같이 자체 설치 스크립트를 실행하여 다운로드한 GRID 드라이버를 설치합니다. 예:

```
[ec2-user ~]$ sudo /bin/sh ./NVIDIA-Linux-x86_64*.run
```

메시지가 표시되면 라이선스 계약에 동의하고 필요에 따라 설치 옵션을 지정합니다. 기본 옵션을 사용해도 됩니다.

9. 드라이버가 작동하는지 확인합니다. 다음 명령의 응답에는 설치된 NVIDIA 드라이버 버전과 GPU에 대한 세부 정보가 나열됩니다.

```
[ec2-user ~]$ nvidia-smi -q | head
```

10. G4dn, G5 또는 G5g 인스턴스에서 NVIDIA vGPU 소프트웨어 버전 14.x 이상을 사용하는 경우 다음 명령으로 GSP를 비활성화합니다. 이것이 필요한 이유에 대한 자세한 내용은 NVIDIA 설명서를 참조하세요.

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

11. 인스턴스를 재부팅합니다.

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

12. (선택 사항) 사용 사례에 따라 다음과 같은 선택적 단계를 완료할 수 있습니다. 이 기능이 필요하지 않으면 다음 단계를 완료하지 마세요.

    a. 최대 4K 해상도의 디스플레이 4개를 활용하는 데 도움이 되도록 고성능 디스플레이 프로토콜인 Amazon DCV를 설정합니다.

    b. NVIDIA Quadro 가상 워크스테이션 모드는 기본적으로 활성화되어 있습니다. RDSH 애플리케이션 호스팅 기능을 위해 GRID 가상 애플리케이션을 활성화하려면 Amazon EC2 GPU 기반 인스턴스에서 NVIDIA GRID 가상 애플리케이션 활성화의 GRID 가상 애플리케이션 활성화 단계를 완료하세요.

Amazon Linux 2

인스턴스에 NVIDIA GRID 드라이버 설치

1. 인스턴스에 연결합니다. 패키지 캐시를 업데이트하고 인스턴스에 대한 패키지 업데이트를 가져옵니다.

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

2. gcc 및 make를 설치합니다(아직 설치되지 않은 경우).

```
[ec2-user ~]$ sudo yum install gcc make
```

3. 인스턴스를 재부팅하여 최신 커널 버전을 로드합니다.

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

4. 재부팅이 끝난 후 인스턴스에 다시 연결합니다.
5. 실행 중인 커널의 버전에 맞는 커널 헤더 패키지를 설치합니다.

```
[ec2-user ~]$ sudo yum install -y kernel-devel-$(uname -r)
```

6. 다음 명령을 사용하여 GRID 드라이버 설치 유틸리티를 다운로드합니다.

```
[ec2-user ~]$ aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/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/
```

7. 다음 명령을 사용하여 드라이버 설치 유틸리티를 실행할 수 있는 권한을 추가합니다.

```
[ec2-user ~]$ chmod +x NVIDIA-Linux-x86_64*.run
```

8. 다음과 같이 자체 설치 스크립트를 실행하여 다운로드한 GRID 드라이버를 설치합니다. 예:

```
[ec2-user ~]$ sudo /bin/sh ./NVIDIA-Linux-x86_64*.run
```

커널 버전 5.10과 함께 Amazon Linux 2를 사용하는 경우 다음 명령을 사용하여 GRID 드라이버를 설치합니다.

```
[ec2-user ~]$ sudo CC=/usr/bin/gcc10-cc ./NVIDIA-Linux-x86_64*.run
```

메시지가 표시되면 라이선스 계약에 동의하고 필요에 따라 설치 옵션을 지정합니다. 기본 옵션을 사용해도 됩니다.

9. 드라이버가 작동하는지 확인합니다. 다음 명령의 응답에는 설치된 NVIDIA 드라이버 버전과 GPU에 대한 세부 정보가 나열됩니다.

```
[ec2-user ~]$ nvidia-smi -q | head
```

10. G4dn, G5 또는 G5g 인스턴스에서 NVIDIA vGPU 소프트웨어 버전 14.x 이상을 사용하는 경우 다음 명령으로 GSP를 비활성화합니다. 이것이 필요한 이유에 대한 자세한 내용은 NVIDIA 설명서를 참조하세요.

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

11. 인스턴스를 재부팅합니다.

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

12. (선택 사항) 사용 사례에 따라 다음과 같은 선택적 단계를 완료할 수 있습니다. 이 기능이 필요하지 않으면 다음 단계를 완료하지 마세요.

    a. 최대 4K 해상도의 디스플레이 4개를 활용하는 데 도움이 되도록 고성능 디스플레이 프로토콜인 Amazon DCV를 설정합니다.

    b. NVIDIA Quadro 가상 워크스테이션 모드는 기본적으로 활성화되어 있습니다. RDSH 애플리케이션 호스팅 기능을 위해 GRID 가상 애플리케이션을 활성화하려면 Amazon EC2 GPU 기반 인스턴스에서 NVIDIA GRID 가상 애플리케이션 활성화의 GRID 가상 애플리케이션 활성화 단계를 완료하세요.

CentOS 7 및 Red Hat Enterprise Linux 7

인스턴스에 NVIDIA GRID 드라이버 설치

1. 인스턴스에 연결합니다. 패키지 캐시를 업데이트하고 인스턴스에 대한 패키지 업데이트를 가져옵니다.

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

2. gcc 및 make를 설치합니다(아직 설치되지 않은 경우).

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

3. 인스턴스를 재부팅하여 최신 커널 버전을 로드합니다.

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

4. 재부팅이 끝난 후 인스턴스에 다시 연결합니다.
5. 실행 중인 커널의 버전에 맞는 커널 헤더 패키지를 설치합니다.

```
[ec2-user ~]$ sudo yum install -y kernel-devel-$(uname -r)
```

6. NVIDIA 그래픽 카드용 nouveau 오픈 소스 드라이버를 비활성화합니다.

   a. nouveau를 `/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.

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

7. 다음 명령을 사용하여 GRID 드라이버 설치 유틸리티를 다운로드합니다.

```
[ec2-user ~]$ aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/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/
```

8. 다음 명령을 사용하여 드라이버 설치 유틸리티를 실행할 수 있는 권한을 추가합니다.

```
[ec2-user ~]$ chmod +x NVIDIA-Linux-x86_64*.run
```

9. 다음과 같이 자체 설치 스크립트를 실행하여 다운로드한 GRID 드라이버를 설치합니다. 예:

```
[ec2-user ~]$ sudo /bin/sh ./NVIDIA-Linux-x86_64*.run
```

메시지가 표시되면 라이선스 계약에 동의하고 필요에 따라 설치 옵션을 지정합니다. 기본 옵션을 사용해도 됩니다.

10. 드라이버가 작동하는지 확인합니다. 다음 명령의 응답에는 설치된 NVIDIA 드라이버 버전과 GPU에 대한 세부 정보가 나열됩니다.

```
[ec2-user ~]$ nvidia-smi -q | head
```

11. G4dn, G5 또는 G5g 인스턴스에서 NVIDIA vGPU 소프트웨어 버전 14.x 이상을 사용하는 경우 다음 명령으로 GSP를 비활성화합니다. 이것이 필요한 이유에 대한 자세한 내용은 NVIDIA 설명서를 참조하세요.

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

12. 인스턴스를 재부팅합니다.

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

13. (선택 사항) 사용 사례에 따라 다음과 같은 선택적 단계를 완료할 수 있습니다. 이 기능이 필요하지 않으면 다음 단계를 완료하지 마세요.

    a. 최대 4K 해상도의 디스플레이 4개를 활용하는 데 도움이 되도록 고성능 디스플레이 프로토콜인 Amazon DCV를 설정합니다.

b. NVIDIA Quadro 가상 워크스테이션 모드는 기본적으로 활성화되어 있습니다. RDSH 애플리케이션 호스팅 기능을 위해 GRID 가상 애플리케이션을 활성화하려면 Amazon EC2 GPU 기반 인스턴스에서 NVIDIA GRID 가상 애플리케이션 활성화의 GRID 가상 애플리케이션 활성화 단계를 완료하세요.

c. GUI 데스크탑/워크스테이션 패키지를 설치합니다.

```
[ec2-user ~]$ sudo yum groupinstall -y "Server with GUI"
```

CentOS Stream 8 및 Red Hat Enterprise Linux 8

인스턴스에 NVIDIA GRID 드라이버 설치

1. 인스턴스에 연결합니다. 패키지 캐시를 업데이트하고 인스턴스에 대한 패키지 업데이트를 가져옵니다.

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

2. gcc 및 make를 설치합니다(아직 설치되지 않은 경우).

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

3. 인스턴스를 재부팅하여 최신 커널 버전을 로드합니다.

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

4. 재부팅이 끝난 후 인스턴스에 다시 연결합니다.
5. 실행 중인 커널의 버전에 맞는 커널 헤더 패키지를 설치합니다.

```
[ec2-user ~]$ sudo dnf install -y elfutils-libelf-devel libglvnd-devel kernel-
devel-$(uname -r)
```

6. 다음 명령을 사용하여 GRID 드라이버 설치 유틸리티를 다운로드합니다.

```
[ec2-user ~]$ aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/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/
```

7. 다음 명령을 사용하여 드라이버 설치 유틸리티를 실행할 수 있는 권한을 추가합니다.

```
[ec2-user ~]$ chmod +x NVIDIA-Linux-x86_64*.run
```

8. 다음과 같이 자체 설치 스크립트를 실행하여 다운로드한 GRID 드라이버를 설치합니다. 예:

```
[ec2-user ~]$ sudo /bin/sh ./NVIDIA-Linux-x86_64*.run
```

   메시지가 표시되면 라이선스 계약에 동의하고 필요에 따라 설치 옵션을 지정합니다. 기본 옵션을 사용해도 됩니다.

9. 드라이버가 작동하는지 확인합니다. 다음 명령의 응답에는 설치된 NVIDIA 드라이버 버전과 GPU에 대한 세부 정보가 나열됩니다.

```
[ec2-user ~]$ nvidia-smi -q | head
```

10. G4dn, G5 또는 G5g 인스턴스에서 NVIDIA vGPU 소프트웨어 버전 14.x 이상을 사용하는 경우 다음 명령으로 GSP를 비활성화합니다. 이것이 필요한 이유에 대한 자세한 내용은 NVIDIA 설명서를 참조하세요.

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

11. 인스턴스를 재부팅합니다.

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

12. (선택 사항) 사용 사례에 따라 다음과 같은 선택적 단계를 완료할 수 있습니다. 이 기능이 필요하지 않으면 다음 단계를 완료하지 마세요.

    a. 최대 4K 해상도의 디스플레이 4개를 활용하는 데 도움이 되도록 고성능 디스플레이 프로토콜인 Amazon DCV를 설정합니다.

    b. NVIDIA Quadro 가상 워크스테이션 모드는 기본적으로 활성화되어 있습니다. RDSH 애플리케이션 호스팅 기능을 위해 GRID 가상 애플리케이션을 활성화하려면 Amazon EC2 GPU 기

반 인스턴스에서 NVIDIA GRID 가상 애플리케이션 활성화의 GRID 가상 애플리케이션 활성화 단계를 완료하세요.

c. GUI 워크스테이션 패키지를 설치합니다.

```
[ec2-user ~]$ sudo dnf groupinstall -y workstation
```

Rocky Linux 8

Linux 인스턴스에서 NVIDIA GRID 드라이버를 설치하려면

1. 인스턴스에 연결합니다. 패키지 캐시를 업데이트하고 인스턴스에 대한 패키지 업데이트를 가져옵니다.

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

2. gcc 및 make를 설치합니다(아직 설치되지 않은 경우).

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

3. 인스턴스를 재부팅하여 최신 커널 버전을 로드합니다.

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

4. 재부팅이 끝난 후 인스턴스에 다시 연결합니다.
5. 실행 중인 커널의 버전에 맞는 커널 헤더 패키지를 설치합니다.

```
[ec2-user ~]$ sudo dnf install -y elfutils-libelf-devel libglvnd-devel kernel-
devel-$(uname -r)
```

6. 다음 명령을 사용하여 GRID 드라이버 설치 유틸리티를 다운로드합니다.

```
[ec2-user ~]$ aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/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/
```

7. 다음 명령을 사용하여 드라이버 설치 유틸리티를 실행할 수 있는 권한을 추가합니다.

```
[ec2-user ~]$ chmod +x NVIDIA-Linux-x86_64*.run
```

8. 다음과 같이 자체 설치 스크립트를 실행하여 다운로드한 GRID 드라이버를 설치합니다. 예:

```
[ec2-user ~]$ sudo /bin/sh ./NVIDIA-Linux-x86_64*.run
```

메시지가 표시되면 라이선스 계약에 동의하고 필요에 따라 설치 옵션을 지정합니다. 기본 옵션을 사용해도 됩니다.

9. 드라이버가 작동하는지 확인합니다. 다음 명령의 응답에는 설치된 NVIDIA 드라이버 버전과 GPU에 대한 세부 정보가 나열됩니다.

```
[ec2-user ~]$ nvidia-smi -q | head
```

10. G4dn, G5 또는 G5g 인스턴스에서 NVIDIA vGPU 소프트웨어 버전 14.x 이상을 사용하는 경우 다음 명령으로 GSP를 비활성화합니다. 이것이 필요한 이유에 대한 자세한 내용은 NVIDIA 설명서를 참조하세요.

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

11. 인스턴스를 재부팅합니다.

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

12. (선택 사항) 사용 사례에 따라 다음과 같은 선택적 단계를 완료할 수 있습니다. 이 기능이 필요하지 않으면 다음 단계를 완료하지 마세요.

    a. 최대 4K 해상도의 디스플레이 4개를 활용하는 데 도움이 되도록 고성능 디스플레이 프로토콜인 Amazon DCV를 설정합니다.

    b. NVIDIA Quadro 가상 워크스테이션 모드는 기본적으로 활성화되어 있습니다. RDSH 애플리케이션 호스팅 기능을 위해 GRID 가상 애플리케이션을 활성화하려면 Amazon EC2 GPU 기반 인스턴스에서 NVIDIA GRID 가상 애플리케이션 활성화의 GRID 가상 애플리케이션 활성화 단계를 완료하세요.

Ubuntu 및 Debian

인스턴스에 NVIDIA GRID 드라이버 설치

1. 인스턴스에 연결합니다. 패키지 캐시를 업데이트하고 인스턴스에 대한 패키지 업데이트를 가져옵니다.

```
$ sudo apt-get update -y
```

2. gcc 및 make를 설치합니다(아직 설치되지 않은 경우).

```
[ec2-user ~]$ sudo apt-get install -y gcc make
```

3. (Ubuntu) `linux-aws` 패키지를 업그레이드하여 최신 버전을 받습니다.

```
$ sudo apt-get upgrade -y linux-aws
```

(Debian) 패키지를 업그레이드하여 최신 버전을 받습니다.

```
$ sudo apt-get upgrade -y
```

4. 인스턴스를 재부팅하여 최신 커널 버전을 로드합니다.

```
$ sudo reboot
```

5. 재부팅이 끝난 후 인스턴스에 다시 연결합니다.
6. 현재 실행 중인 커널의 버전에 맞는 커널 헤더 패키지를 설치합니다.

```
$ sudo apt-get install -y linux-headers-$(uname -r)
```

7. NVIDIA 그래픽 카드용 nouveau 오픈 소스 드라이버를 비활성화합니다.

   a. nouveau를 `/etc/modprobe.d/blacklist.conf` 블랙리스트 파일에 추가합니다. 다음 코드 블록을 복사하여 터미널에 붙여 넣습니다.

```
$ 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.

```
$ sudo update-grub
```

8. 다음 명령을 사용하여 GRID 드라이버 설치 유틸리티를 다운로드합니다.

```
$ aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/latest/ .
```

여러 버전의 GRID 드라이버가 이 버킷에 저장되어 있습니다. 다음 명령을 사용하여 사용 가능한 모든 버전을 볼 수 있습니다.

```
$ aws s3 ls --recursive s3://ec2-linux-nvidia-drivers/
```

9. 다음 명령을 사용하여 드라이버 설치 유틸리티를 실행할 수 있는 권한을 추가합니다.

```
$ chmod +x NVIDIA-Linux-x86_64*.run
```

10. 다음과 같이 자체 설치 스크립트를 실행하여 다운로드한 GRID 드라이버를 설치합니다. 예:

```
$ sudo /bin/sh ./NVIDIA-Linux-x86_64*.run
```

메시지가 표시되면 라이선스 계약에 동의하고 필요에 따라 설치 옵션을 지정합니다. 기본 옵션을 사용해도 됩니다.

11. 드라이버가 작동하는지 확인합니다. 다음 명령의 응답에는 설치된 NVIDIA 드라이버 버전과 GPU에 대한 세부 정보가 나열됩니다.

```
$ nvidia-smi -q | head
```

12. G4dn, G5 또는 G5g 인스턴스에서 NVIDIA vGPU 소프트웨어 버전 14.x 이상을 사용하는 경우 다음 명령으로 GSP를 비활성화합니다. 이것이 필요한 이유에 대한 자세한 내용은 NVIDIA 설명서를 참조하세요.

```
$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /etc/
modprobe.d/nvidia.conf
```

13. 인스턴스를 재부팅합니다.

```
$ sudo reboot
```

14. (선택 사항) 사용 사례에 따라 다음과 같은 선택적 단계를 완료할 수 있습니다. 이 기능이 필요하지 않으면 다음 단계를 완료하지 마세요.

    a. 최대 4K 해상도의 디스플레이 4개를 활용하는 데 도움이 되도록 고성능 디스플레이 프로토콜인 Amazon DCV를 설정합니다.

    b. NVIDIA Quadro 가상 워크스테이션 모드는 기본적으로 활성화되어 있습니다. RDSH 애플리케이션 호스팅 기능을 위해 GRID 가상 애플리케이션을 활성화하려면 Amazon EC2 GPU 기반 인스턴스에서 NVIDIA GRID 가상 애플리케이션 활성화의 GRID 가상 애플리케이션 활성화 단계를 완료하세요.

    c. GUI 데스크탑/워크스테이션 패키지를 설치합니다.

```
$ sudo apt-get install -y lightdm ubuntu-desktop
```

Windows 운영 체제

Windows 인스턴스에서 NVIDIA GRID 드라이버를 설치하려면

1. Windows 인스턴스에 연결하고 PowerShell 창을 엽니다.
2. 다음 PowerShell 명령을 사용하여 Amazon S3에서 드라이버와 NVIDIA GRID 클라우드 최종 사용자 라이선스 계약을 데스크톱에 다운로드합니다.

```
$Bucket = "ec2-windows-nvidia-drivers"
$KeyPrefix = "latest"
$LocalPath = "$home\Desktop\NVIDIA"
$Objects = Get-S3Object -BucketName $Bucket -KeyPrefix $KeyPrefix -Region us-east-1
foreach ($Object in $Objects) {
$LocalFileName = $Object.Key
if ($LocalFileName -ne '' -and $Object.Size -ne 0) {
    $LocalFilePath = Join-Path $LocalPath $LocalFileName
    Copy-S3Object -BucketName $Bucket -Key $Object.Key -LocalFile $LocalFilePath -
Region us-east-1
```

```
}
}
```

여러 버전의 NVIDIA GRID 드라이버가 이 버킷에 저장되어 있습니다. `-KeyPrefix $KeyPrefix` 옵션을 제거하면 버킷에서 사용 가능한 모든 Windows 버전을 다운로드할 수 있습니다. 운영 체제의 NVIDIA GRID 드라이버 버전에 대한 자세한 내용은 NVIDIA 웹 사이트의 NVIDIA 가상 GPU(vGPU) 소프트웨어를 참조하세요.

GRID 버전 11.0부터는 `latest`에서 G3 및 G4dn 인스턴스 모두에 드라이버를 사용할 수 있습니다. 11.0 이후 버전은 `g4/latest`에 추가되지 않지만 G4dn에 해당하는 11.0 이하 버전은 `g4/latest`에 유지됩니다.

G5 인스턴스에는 GRID 13.1 이상이나 GRID 12.4 이상이 필요합니다.

3. 바탕 화면으로 이동하여 설치 파일을 두 번 클릭하여 시작합니다(인스턴스 OS 버전에 해당하는 드라이버 버전 선택). 안내에 따라 드라이버를 설치하고 필요에 따라 인스턴스를 재부팅합니다. GPU가 제대로 작동하는지 확인하려면 장치 관리자를 확인합니다.
4. (선택 사항) 다음 명령으로 제어판에서 라이선싱 페이지를 비활성화하여 사용자가 실수로 제품 유형을 변경하는 것을 방지합니다(NVIDIA GRID 가상 워크스테이션은 기본적으로 활성화되어 있음). 자세한 내용은 GRID 라이선싱 사용 설명서를 참조하세요.

   PowerShell

   다음 PowerShell 명령을 실행하여 제어판에서 라이선스 페이지를 비활성화하는 레지스트리 값을 생성합니다. 기본적으로 AWS Tools for PowerShell Windows AMI의 AWS은 32비트 버전으로 설정되어 있어서 이 명령이 실패합니다. 대신 운영 체제에 포함된 64비트 버전의 PowerShell을 사용하세요.

   ```
   New-Item -Path "HKLM:\SOFTWARE\NVIDIA Corporation\Global" -Name GridLicensing
   New-ItemProperty -Path "HKLM:\SOFTWARE\NVIDIA Corporation\Global\GridLicensing" -
   Name "NvCplDisableManageLicensePage" -PropertyType "DWord" -Value "1"
   ```

   명령 프롬프트

   다음 레지스트리 명령을 실행하여 제어판에서 라이선스 페이지를 비활성화하는 레지스트리 값을 생성합니다. 명령 프롬프트 창 또는 64비트 버전의 PowerShell을 사용하여 실행할 수 있습니다.

   ```
   reg add "HKLM\SOFTWARE\NVIDIA Corporation\Global\GridLicensing" /v
    NvCplDisableManageLicensePage /t REG_DWORD /d 1
   ```

5. (선택 사항) 사용 사례에 따라 다음과 같은 선택적 단계를 완료할 수 있습니다. 이 기능이 필요하지 않으면 다음 단계를 완료하지 마세요.

   a. 최대 4K 해상도의 디스플레이 4개를 활용하는 데 도움이 되도록 고성능 디스플레이 프로토콜인 Amazon DCV를 설정합니다.

   b. NVIDIA Quadro 가상 워크스테이션 모드는 기본적으로 활성화되어 있습니다. RDSH 애플리케이션 호스팅 기능을 위해 GRID 가상 애플리케이션을 활성화하려면 Amazon EC2 GPU 기반 인스턴스에서 NVIDIA GRID 가상 애플리케이션 활성화의 GRID 가상 애플리케이션 활성화 단계를 완료하세요.

옵션 4: NVIDIA 게임 드라이버(G4dn, G5, G6 및 G6e 인스턴스)

이러한 드라이버는 AWS 고객만 사용할 수 있습니다. 드라이버를 다운로드하면 NVIDIA A10G 및 NVIDIA Tesla T4 하드웨어와 함께 사용하기 위해 AMI를 개발하는 용도로만 다운로드한 소프트웨어를 사용한다는 것에 동의하는 것입니다. 소프트웨어를 설치하면 NVIDIA GRID 클라우드 최종 사용자 라이선스 계약의 약관이 적용됩니다.

고려 사항

- Grid 라이선싱이 작동하려면 AWS에서 제공하는 DNS 확인이 G3 인스턴스에 필요합니다.
- IMDSv2는 NVIDIA 드라이버 버전 495.x 이상에서만 지원됩니다.

사전 조건

- (Linux) 인스턴스에 AWS CLI가 설치되고 기본 자격 증명으로 구성되어 있는지 확인합니다. 자세한 내용은 AWS Command Line Interface 사용 설명서에서 AWS CLI 설치를 참조하세요.
- 사용자 또는 역할에 AmazonS3ReadOnlyAccess 정책이 포함된 권한이 부여되어야 합니다.

Amazon Linux 2023

인스턴스에 NVIDIA 게임 드라이버 설치

1. 인스턴스에 연결합니다. 패키지 캐시를 업데이트하고 인스턴스에 대한 패키지 업데이트를 가져옵니다.

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

2. gcc 및 make를 설치합니다(아직 설치되지 않은 경우).

```
[ec2-user ~]$ sudo dnf install gcc make
```

3. 인스턴스를 재부팅하여 최신 커널 버전을 로드합니다.

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

4. 재부팅이 끝나면 인스턴스에 다시 연결합니다.
5. 커널 헤더 패키지를 설치합니다.

```
[ec2-user ~]$ sudo dnf install -y kernel-devel kernel-modules-extra kernel-devel-
$(uname -r) kernel-headers-$(uname -r) dkms
```

6. 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/
```

7. 다운로드한 `.zip` 아카이브에서 게임 드라이버 설치 유틸리티의 압축을 풉니다.

```
[ec2-user ~]$ unzip latest-driver-name.zip -d nvidia-drivers
```

8. 다음 명령을 사용하여 드라이버 설치 유틸리티를 실행할 수 있는 권한을 추가합니다.

```
[ec2-user ~]$ chmod +x nvidia-drivers/NVIDIA-Linux-x86_64*-grid.run
```

9. 다음 명령을 사용하여 설치 프로그램을 실행합니다.

```
[ec2-user ~]$ sudo ./nvidia-drivers/NVIDIA-Linux-x86_64*.run
```

메시지가 표시되면 라이선스 계약에 동의하고 필요에 따라 설치 옵션을 지정합니다. 기본 옵션을 사용해도 됩니다.

10. 다음 명령을 사용하여 필수 구성 파일을 생성합니다.

```
[ec2-user ~]$ cat << EOF | sudo tee -a /etc/nvidia/gridd.conf
vGamingMarketplace=2
```

```
EOF
```

11. 인증 파일을 다운로드하고 이름을 바꾸려면 다음 명령을 사용하세요.

    - 버전 460.39 이상:

      ```
      [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2024_02_22.cert"
      ```

    - 버전 440.68에서 445.48:

      ```
      [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2020_04.cert"
      ```

    - 이전 버전:

      ```
      [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2019_09.cert"
      ```

12. G4dn, G5 또는 G5g 인스턴스에서 NVIDIA 드라이버 버전 510.x 이상을 사용하는 경우 다음 명령으로 GSP를 비활성화합니다. 이것이 필요한 이유에 대한 자세한 내용은 NVIDIA 설명서를 참조하세요.

    ```
    [ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
    ```

    ```
    [ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /etc/modprobe.d/nvidia.conf
    ```

13. 인스턴스를 재부팅합니다.

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

14. 다음 명령을 사용하여 NVIDIA 게임 라이선스를 확인합니다.

    ```
    [ec2-user ~]$ nvidia-smi.exe -q
    ```

    출력에서 `vGPU Software Licensed Product`를 검색합니다.

15. (선택 사항) 최대 4K 해상도의 단일 디스플레이를 활용하는 데 도움이 되도록 고성능 디스플레이 프로토콜인 Amazon DCV를 설정합니다.

Amazon Linux 2

인스턴스에 NVIDIA 게임 드라이버 설치

1. 인스턴스에 연결합니다. 패키지 캐시를 업데이트하고 인스턴스에 대한 패키지 업데이트를 가져옵니다.

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

2. gcc 및 make를 설치합니다(아직 설치되지 않은 경우).

```
[ec2-user ~]$ sudo yum install gcc make
```

3. 인스턴스를 재부팅하여 최신 커널 버전을 로드합니다.

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

4. 재부팅이 끝나면 인스턴스에 다시 연결합니다.
5. 현재 실행 중인 커널의 버전에 맞는 커널 헤더 패키지를 설치합니다.

```
[ec2-user ~]$ sudo yum install -y kernel-devel-$(uname -r)
```

6. 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/
```

7. 다운로드한 `.zip` 아카이브에서 게임 드라이버 설치 유틸리티의 압축을 풉니다.

```
[ec2-user ~]$ unzip latest-driver-name.zip -d nvidia-drivers
```

8. 다음 명령을 사용하여 드라이버 설치 유틸리티를 실행할 수 있는 권한을 추가합니다.

```
[ec2-user ~]$ chmod +x nvidia-drivers/NVIDIA-Linux-x86_64*-grid.run
```

9. 다음 명령을 사용하여 설치 프로그램을 실행합니다.

```
[ec2-user ~]$ sudo ./nvidia-drivers/NVIDIA-Linux-x86_64*.run
```

커널 버전 5.10과 함께 Amazon Linux 2를 사용하는 경우 다음 명령을 사용하여 NVIDIA 게이밍 드라이버를 설치합니다.

```
[ec2-user ~]$ sudo CC=/usr/bin/gcc10-cc ./NVIDIA-Linux-x86_64*.run
```

메시지가 표시되면 라이선스 계약에 동의하고 필요에 따라 설치 옵션을 지정합니다. 기본 옵션을 사용해도 됩니다.

10. 다음 명령을 사용하여 필수 구성 파일을 생성합니다.

```
[ec2-user ~]$ cat << EOF | sudo tee -a /etc/nvidia/gridd.conf
vGamingMarketplace=2
EOF
```

11. 인증 파일을 다운로드하고 이름을 바꾸려면 다음 명령을 사용하세요.
    - 버전 460.39 이상:

      ```
      [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2024_02_22.cert"
      ```

    - 버전 440.68에서 445.48:

      ```
      [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2020_04.cert"
      ```

    - 이전 버전:

      ```
      [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2019_09.cert"
      ```

12. G4dn, G5 또는 G5g 인스턴스에서 NVIDIA 드라이버 버전 510.x 이상을 사용하는 경우 다음 명령으로 GSP를 비활성화합니다. 이것이 필요한 이유에 대한 자세한 내용은 NVIDIA 설명서를 참조하세요.

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

13. 인스턴스를 재부팅합니다.

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

14. 다음 명령을 사용하여 NVIDIA 게임 라이선스를 확인합니다.

```
[ec2-user ~]$ nvidia-smi.exe -q
```

출력에서 `vGPU Software Licensed Product`를 검색합니다.

15. (선택 사항) 최대 4K 해상도의 단일 디스플레이를 활용하는 데 도움이 되도록 고성능 디스플레이 프로토콜인 Amazon DCV를 설정합니다.

CentOS 7 및 Red Hat Enterprise Linux 7

인스턴스에 NVIDIA 게임 드라이버 설치

1. Linux 인스턴스에 연결합니다. gcc 및 make를 설치합니다(아직 설치되지 않은 경우).

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

2. 패키지 캐시를 업데이트하고 인스턴스에 대한 패키지 업데이트를 가져옵니다.

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

3. 인스턴스를 재부팅하여 최신 커널 버전을 로드합니다.

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

4. 재부팅이 끝난 후 인스턴스에 다시 연결합니다.
5. 현재 실행 중인 커널의 버전에 맞는 커널 헤더 패키지를 설치합니다.

```
[ec2-user ~]$ sudo yum install -y unzip kernel-devel-$(uname -r)
```

6. NVIDIA 그래픽 카드용 nouveau 오픈 소스 드라이버를 비활성화합니다.

a. nouveau를 /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.

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

7. 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/
```

8. 다운로드한 .zip 아카이브에서 게임 드라이버 설치 유틸리티의 압축을 풉니다.

```
[ec2-user ~]$ unzip *Gaming-Linux-Guest-Drivers.zip -d nvidia-drivers
```

9. 다음 명령을 사용하여 드라이버 설치 유틸리티를 실행할 수 있는 권한을 추가합니다.

```
[ec2-user ~]$ chmod +x nvidia-drivers/NVIDIA-Linux-x86_64*-grid.run
```

10. 다음 명령을 사용하여 설치 프로그램을 실행합니다.

```
[ec2-user ~]$ sudo nvidia-drivers/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 사용하세요.

    - 버전 460.39 이상:

    ```
    [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
    gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2024_02_22.cert"
    ```

    - 버전 440.68에서 445.48:

    ```
    [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
    gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2020_04.cert"
    ```

    - 이전 버전:

    ```
    [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
    gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2019_09.cert"
    ```

13. G4dn, G5 또는 G5g 인스턴스에서 NVIDIA 드라이버 버전 510.x 이상을 사용하는 경우 다음 명령으로 GSP를 비활성화합니다. 이것이 필요한 이유에 대한 자세한 내용은 NVIDIA 설명서를 참조하세요.

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

14. 인스턴스를 재부팅합니다.

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

15. (선택 사항) 최대 4K 해상도의 단일 디스플레이를 활용하는 데 도움이 되도록 고성능 디스플레이 프로토콜인 Amazon DCV를 설정합니다. 이 기능이 필요하지 않으면 다음 단계를 완료하지 마세요.

CentOS Stream 8 및 Red Hat Enterprise Linux 8

인스턴스에 NVIDIA 게임 드라이버 설치

1. Linux 인스턴스에 연결합니다. gcc 및 make를 설치합니다(아직 설치되지 않은 경우).

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

2. 패키지 캐시를 업데이트하고 인스턴스에 대한 패키지 업데이트를 가져옵니다.

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

3. 인스턴스를 재부팅하여 최신 커널 버전을 로드합니다.

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

4. 재부팅이 끝난 후 인스턴스에 다시 연결합니다.
5. 현재 실행 중인 커널의 버전에 맞는 커널 헤더 패키지를 설치합니다.

```
[ec2-user ~]$ sudo yum install -y unzip kernel-devel-$(uname -r)
```

6. 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/
```

7. 다운로드한 `.zip` 아카이브에서 게임 드라이버 설치 유틸리티의 압축을 풉니다.

```
[ec2-user ~]$ unzip *Gaming-Linux-Guest-Drivers.zip -d nvidia-drivers
```

8. 다음 명령을 사용하여 드라이버 설치 유틸리티를 실행할 수 있는 권한을 추가합니다.

```
[ec2-user ~]$ chmod +x nvidia-drivers/NVIDIA-Linux-x86_64*-grid.run
```

9. 다음 명령을 사용하여 설치 프로그램을 실행합니다.

```
[ec2-user ~]$ sudo nvidia-drivers/NVIDIA-Linux-x86_64*.run
```

메시지가 표시되면 라이선스 계약에 동의하고 필요에 따라 설치 옵션을 지정합니다. 기본 옵션을 사용해도 됩니다.

10. 다음 명령을 사용하여 필수 구성 파일을 생성합니다.

```
[ec2-user ~]$ cat << EOF | sudo tee -a /etc/nvidia/gridd.conf
vGamingMarketplace=2
EOF
```

11. 인증 파일을 다운로드하고 이름을 바꾸려면 다음 명령을 사용하세요.

- 버전 460.39 이상:

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2024_02_22.cert"
```

- 버전 440.68에서 445.48:

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2020_04.cert"
```

- 이전 버전:

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2019_09.cert"
```

12. G4dn, G5 또는 G5g 인스턴스에서 NVIDIA 드라이버 버전 510.x 이상을 사용하는 경우 다음 명령으로 GSP를 비활성화합니다. 이것이 필요한 이유에 대한 자세한 내용은 NVIDIA 설명서를 참조하세요.

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

13. 인스턴스를 재부팅합니다.

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

14. (선택 사항) 최대 4K 해상도의 단일 디스플레이를 활용하는 데 도움이 되도록 고성능 디스플레이 프로토콜인 Amazon DCV를 설정합니다.

Rocky Linux 8

인스턴스에 NVIDIA 게임 드라이버 설치

1. Linux 인스턴스에 연결합니다. gcc 및 make를 설치합니다(아직 설치되지 않은 경우).

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

2. 패키지 캐시를 업데이트하고 인스턴스에 대한 패키지 업데이트를 가져옵니다.

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

3. 인스턴스를 재부팅하여 최신 커널 버전을 로드합니다.

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

4. 재부팅이 끝난 후 인스턴스에 다시 연결합니다.
5. 현재 실행 중인 커널의 버전에 맞는 커널 헤더 패키지를 설치합니다.

```
[ec2-user ~]$ sudo dnf install -y unzip elfutils-libelf-devel libglvnd-devel
 kernel-devel-$(uname -r)
```

6. 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/
```

7. 다운로드한 `.zip` 아카이브에서 게임 드라이버 설치 유틸리티의 압축을 풉니다.

```
[ec2-user ~]$ unzip *Gaming-Linux-Guest-Drivers.zip -d nvidia-drivers
```

8. 다음 명령을 사용하여 드라이버 설치 유틸리티를 실행할 수 있는 권한을 추가합니다.

```
[ec2-user ~]$ chmod +x nvidia-drivers/NVIDIA-Linux-x86_64*-grid.run
```

9. 다음 명령을 사용하여 설치 프로그램을 실행합니다.

```
[ec2-user ~]$ sudo nvidia-drivers/NVIDIA-Linux-x86_64*.run
```

   메시지가 표시되면 라이선스 계약에 동의하고 필요에 따라 설치 옵션을 지정합니다. 기본 옵션을 사용해도 됩니다.

10. 다음 명령을 사용하여 필수 구성 파일을 생성합니다.

```
[ec2-user ~]$ cat << EOF | sudo tee -a /etc/nvidia/gridd.conf
vGamingMarketplace=2
EOF
```

11. 인증 파일을 다운로드하고 이름을 바꾸려면 다음 명령을 사용하세요.
    - 버전 460.39 이상:

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2024_02_22.cert"
```

    - 버전 440.68에서 445.48:

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2020_04.cert"
```

    - 이전 버전:

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2019_09.cert"
```

12. G4dn, G5 또는 G5g 인스턴스에서 NVIDIA 드라이버 버전 510.x 이상을 사용하는 경우 다음 명령으로 GSP를 비활성화합니다. 이것이 필요한 이유에 대한 자세한 내용은 NVIDIA 설명서를 참조하세요.

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

13. 인스턴스를 재부팅합니다.

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

14. (선택 사항) 최대 4K 해상도의 단일 디스플레이를 활용하는 데 도움이 되도록 고성능 디스플레이 프로토콜인 Amazon DCV를 설정합니다.

Ubuntu 및 Debian

인스턴스에 NVIDIA 게임 드라이버 설치

1. Linux 인스턴스에 연결합니다. gcc 및 make를 설치합니다(아직 설치되지 않은 경우).

```
$ sudo apt-get install -y gcc make build-essential
```

2. 패키지 캐시를 업데이트하고 인스턴스에 대한 패키지 업데이트를 가져옵니다.

```
$ sudo apt-get update -y
```

3. `linux-aws` 패키지를 업그레이드하여 최신 버전을 받습니다.

```
$ sudo apt-get upgrade -y linux-aws
```

4. 인스턴스를 재부팅하여 최신 커널 버전을 로드합니다.

```
$ sudo reboot
```

5. 재부팅이 끝난 후 인스턴스에 다시 연결합니다.
6. 현재 실행 중인 커널의 버전에 맞는 커널 헤더 패키지를 설치합니다.

```
$ sudo apt install -y unzip dkms linux-headers-$(uname -r)
```

7. NVIDIA 그래픽 카드용 nouveau 오픈 소스 드라이버를 비활성화합니다.

   a. nouveau를 `/etc/modprobe.d/blacklist.conf` 블랙리스트 파일에 추가합니다. 다음 코드 블록을 복사하여 터미널에 붙여 넣습니다.

```
$ 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.

```
$ sudo update-grub
```

8. 다음 명령을 사용하여 게임 드라이버 설치 유틸리티를 다운로드합니다.

```
$ 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.

```
$ aws s3 ls --recursive s3://nvidia-gaming/linux/
```

9. 다운로드한 .zip 아카이브에서 게임 드라이버 설치 유틸리티의 압축을 풉니다.

```
$ unzip *Gaming-Linux-Guest-Drivers.zip -d nvidia-drivers
```

10. 다음 명령을 사용하여 드라이버 설치 유틸리티를 실행할 수 있는 권한을 추가합니다.

```
$ chmod +x nvidia-drivers/NVIDIA-Linux-x86_64*-grid.run
```

11. 다음 명령을 사용하여 설치 프로그램을 실행합니다.

```
$ sudo nvidia-drivers/NVIDIA-Linux-x86_64*.run
```

메시지가 표시되면 라이선스 계약에 동의하고 필요에 따라 설치 옵션을 지정합니다. 기본 옵션을 사용해도 됩니다.

12. 다음 명령을 사용하여 필수 구성 파일을 생성합니다.

```
$ cat << EOF | sudo tee -a /etc/nvidia/gridd.conf
vGamingMarketplace=2
EOF
```

13. 인증 파일을 다운로드하고 이름을 바꾸려면 다음 명령을 사용하세요.

    - 버전 460.39 이상:

```
$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2024_02_22.cert"
```

    - 버전 440.68에서 445.48:

```
$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2020_04.cert"
```

    - 이전 버전:

```
$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2019_09.cert"
```

14. G4dn, G5 또는 G5g 인스턴스에서 NVIDIA 드라이버 버전 510.x 이상을 사용하는 경우 다음 명령으로 GSP를 비활성화합니다. 이것이 필요한 이유에 대한 자세한 내용은 NVIDIA 설명서를 참조하세요.

```
$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /etc/
modprobe.d/nvidia.conf
```

15. 인스턴스를 재부팅합니다.

```
$ sudo reboot
```

16. (선택 사항) 최대 4K 해상도의 단일 디스플레이를 활용하는 데 도움이 되도록 고성능 디스플레이 프로토콜인 Amazon DCV를 설정합니다. 이 기능이 필요하지 않으면 다음 단계를 완료하지 마세요.

Windows 운영 체제

인스턴스에 NVIDIA 게임 드라이버를 설치하기 전에 먼저 모든 게임 드라이버에서 명시한 고려 사항 외에도 다음 사전 조건을 충족하는지 확인해야 합니다.

- 사용자 지정 Windows AMI를 사용하여 Windows 인스턴스를 시작하는 경우 게임 드라이버가 작동하려면 AMI가 Windows Sysprep으로 생성된 표준화된 이미지여야 합니다. 자세한 내용은 Windows Sysprep을 사용하여 Amazon EC2 AMI 생성 섹션을 참조하세요.
- Windows 인스턴스에서 AWS Tools for Windows PowerShell에 대한 기본 자격 증명을 구성합니다. 자세한 내용은 AWS Tools for Windows PowerShell 사용 설명서에서 AWS Tools for Windows PowerShell 시작하기를 참조하세요.

Windows 인스턴스에서 NVIDIA 게임 드라이버를 설치하려면

1. Windows 인스턴스에 연결하고 PowerShell 창을 엽니다.
2. 다음 PowerShell 명령을 사용하여 게임 드라이버를 다운로드하고 설치합니다.

```
$Bucket = "nvidia-gaming"
$KeyPrefix = "windows/latest"
$LocalPath = "$home\Desktop\NVIDIA"
$Objects = Get-S3Object -BucketName $Bucket -KeyPrefix $KeyPrefix -Region us-east-1
foreach ($Object in $Objects) {
$LocalFileName = $Object.Key
if ($LocalFileName -ne '' -and $Object.Size -ne 0) {
    $LocalFilePath = Join-Path $LocalPath $LocalFileName
    Copy-S3Object -BucketName $Bucket -Key $Object.Key -LocalFile $LocalFilePath -
Region us-east-1
}
}
```

   여러 버전의 NVIDIA GRID 드라이버가 이 S3 버킷에 저장되어 있습니다. `$KeyPrefix` 변수의 값을 "windows/latest"에서 "windows"로 변경하면 버킷에서 사용 가능한 모든 버전을 다운로드할 수 있습니다.
3. 바탕 화면으로 이동하여 설치 파일을 두 번 클릭하여 시작합니다(인스턴스 OS 버전에 해당하는 드라이버 버전 선택). 안내에 따라 드라이버를 설치하고 필요에 따라 인스턴스를 재부팅합니다. GPU가 제대로 작동하는지 확인하려면 Device Manager를 확인합니다.
4. 다음과 같은 방법 중 하나를 사용하여 드라이버를 등록합니다.

Version 527.27 or above

PowerShell 64비트 버전 또는 명령 프롬프트 창을 통해 다음과 같은 레지스트리 키를 생성합니다.

키: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\nvlddmkm\Global

이름: vGamingMarketplace

유형: DWord

값: 2

PowerShell

다음과 같은 PowerShell 명령을 사용하여 이 레지스트리 값을 생성합니다. 기본적으로 AWS Tools for PowerShell Windows AMI의 AWS은 32비트 버전으로 설정되어 있어서 이 명령이 실패합니다. 대신 운영 체제에 포함된 64비트 버전의 PowerShell을 사용하세요.

```
New-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\nvlddmkm\Global"
 -Name "vGamingMarketplace" -PropertyType "DWord" -Value "2"
```

명령 프롬프트

다음과 같은 레지스트리 명령을 사용하여 이 레지스트리 값을 생성합니다. 명령 프롬프트 창 또는 64비트 버전의 PowerShell을 사용하여 실행할 수 있습니다.

```
reg add "HKLM\SYSTEM\CurrentControlSet\Services\nvlddmkm\Global" /v
 vGamingMarketplace /t REG_DWORD /d 2
```

Earlier versions

PowerShell 64비트 버전 또는 명령 프롬프트 창을 통해 다음과 같은 레지스트리 키를 생성합니다.

키: HKEY_LOCAL_MACHINE\SOFTWARE\NVIDIA Corporation\Global

이름: vGamingMarketplace

유형: DWord

값: 2

PowerShell

다음과 같은 PowerShell 명령을 사용하여 이 레지스트리 값을 생성합니다. 기본적으로 AWS Tools for PowerShell Windows AMI의 AWS은 32비트 버전으로 설정되어 있어서 이 명령이 실패합니다. 대신 운영 체제에 포함된 64비트 버전의 PowerShell을 사용하세요.

```
New-ItemProperty -Path "HKLM:\SOFTWARE\NVIDIA Corporation\Global" -Name
 "vGamingMarketplace" -PropertyType "DWord" -Value "2"
```

명령 프롬프트

다음과 같은 레지스트리 명령을 실행하여 명령 프롬프트 창을 통해 이 레지스트리 키를 생성합니다. PowerShell 64비트 버전에서도 이 명령을 사용할 수 있습니다.

```
reg add "HKLM\SOFTWARE\NVIDIA Corporation\Global" /v vGamingMarketplace /t
 REG_DWORD /d 2
```

5. PowerShell에서 다음 명령을 실행합니다. 이는 인증 파일을 다운로드하고, `GridSwCert.txt` 파일 이름을 바꾸고, 파일을 시스템 드라이브의 공용 문서 폴더로 이동시킵니다. 일반적으로 폴더 경로는 `C:\Users\Public\Documents`입니다.
   - 버전 460.39 이상:

     ```
     Invoke-WebRequest -Uri "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-
     Archive/GridSwCertWindows_2024_02_22.cert" -OutFile "$Env:PUBLIC\Documents
     \GridSwCert.txt"
     ```

   - 버전 445.87:

     ```
     Invoke-WebRequest -Uri "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-
     Archive/GridSwCert-Windows_2020_04.cert" -OutFile "$Env:PUBLIC\Documents
     \GridSwCert.txt"
     ```

   - 이전 버전:

```
Invoke-WebRequest -Uri "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-
Archive/GridSwCert-Windows_2019_09.cert" -OutFile "$Env:PUBLIC\Documents
\GridSwCert.txt"
```

Windows Server 2016 또는 이전 버전을 사용 중이고 파일을 다운로드할 때 오류가 발생하는 경우 PowerShell 터미널에서 TLS 1.2를 활성화해야 할 수 있습니다. 다음 명령을 사용하여 현재 PowerShell 세션에 대해 TLS 1.2를 활성화한 다음 다시 시도해보세요.

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
```

6. 인스턴스를 재부팅합니다.
7. 인스턴스에서 `nvidia-smi.exe` 파일을 찾습니다.

```
Get-ChildItem -Path C:\ -Recurse -Filter "nvidia-smi.exe"
```

다음 명령을 사용하여 NVIDIA 게임 라이선스를 확인합니다. 이전 명령의 출력에서 *path*를 폴더의 이름으로 바꿉니다.

```
C:\Windows\System32\DriverStore\FileRepository\path\nvidia-smi.exe -q
```

다음과 같이 출력됩니다

```
vGPU Software Licensed Product
Product Name                : NVIDIA Cloud Gaming
License Status              : Licensed (Expiry: N/A)
```

8. (선택 사항) 최대 4K 해상도의 단일 디스플레이를 활용하는 데 도움이 되도록 고성능 디스플레이 프로토콜인 Amazon DCV를 설정합니다. 이 기능이 필요하지 않으면 다음 단계를 완료하지 마세요.

CUDA의 추가 버전 설치

인스턴스에 NVIDIA 그래픽 드라이버를 설치한 후 그래픽 드라이버와 함께 번들로 제공되는 버전이 아닌 CUDA 버전을 설치할 수 있습니다. 다음 절차에서는 인스턴스에서 여러 버전의 CUDA를 구성하는 방법을 보여줍니다.

Linux에 CUDA 도구 키트 설치

Linux에 CUDA 도구 키트를 설치하려면 다음 단계를 수행합니다.

1. Linux 인스턴스에 연결합니다.
2. NVIDIA 웹 사이트를 열고 필요한 CUDA 버전을 선택합니다.
3. 인스턴스의 운영 체제에 대한 아키텍처, 배포 및 버전을 선택합니다. 설치 프로그램 유형에서 runfile(로컬)을 선택합니다.
4. 지침에 따라 설치 스크립트를 다운로드합니다.
5. 다음 명령을 사용하여 다운로드한 설치 스크립트에 실행 권한을 추가합니다.

```
[ec2-user ~]$ chmod +x downloaded_installer_file
```

6. 다음과 같이 설치 스크립트를 실행하여 CUDA 도구 키트를 설치하고 CUDA 버전 번호를 도구 키트 경로에 추가합니다.

```
[ec2-user ~]$ sudo sh downloaded_installer_file --silent --override --toolkit --
samples --toolkitpath=/usr/local/cuda-version --samplespath=/usr/local/cuda --no-
opengl-libs
```

7. (선택 사항) 다음과 같이 기본 CUDA 버전을 설정합니다.

```
[ec2-user ~]$ sudo ln -s /usr/local/cuda-version /usr/local/cuda
```

Windows에 CUDA 도구 키트 설치

Windows에 CUDA 도구 키트를 설치하려면 다음 단계를 수행합니다.

CUDA 도구 키트를 설치하려면

1. Windows 인스턴스에 연결합니다.
2. NVIDIA 웹 사이트를 열고 필요한 CUDA 버전을 선택합니다.
3. 설치 프로그램 유형에서 exe(로컬)를 선택한 다음 다운로드를 선택합니다.
4. 브라우저를 사용하여 다운로드한 설치 파일을 실행합니다. 지침에 따라 CUDA 도구 키트를 설치합니다. 인스턴스를 재부팅해야 할 수 있습니다.

EC2 Windows 인스턴스에 ENA 드라이버 설치

인스턴스가 Amazon에서 제공하는 최신 Windows Amazon Machine Image(AMI) 중 하나를 기반으로 하지 않는 경우 다음 절차에 따라 인스턴스에 최신 ENA 드라이버를 설치하세요. 인스턴스를 재부팅하기 편리한 시간에 이 업데이트를 수행하세요. 설치 스크립트가 인스턴스를 자동으로 재부팅하지 않는 경우 마지막 단계로 인스턴스를 재부팅하는 것이 좋습니다.

인스턴스가 실행되는 동안 인스턴스 스토어 볼륨을 사용하여 데이터를 저장하는 경우 인스턴스를 중지하면 해당 데이터가 지워집니다. 인스턴스를 중지하기 전에 필요한 데이터를 인스턴스 스토어 볼륨에서 영구 스토리지(예: Amazon EBS 또는 Amazon S3)로 복사했는지 확인하세요.

사전 조건

ENA 드라이버를 설치 또는 업그레이드하려면 Windows 인스턴스가 다음 사전 조건을 충족해야 합니다.

- PowerShell 버전 3.0 이상이 설치되어 있어야 합니다.
- 이 섹션에 표시된 명령은 64비트 버전의 PowerShell에서 실행되어야 합니다. x86 버전의 PowerShell을 사용하지 마세요. 이는 32비트 버전 쉘이며 이러한 명령에는 지원되지 않습니다.

1단계: 데이터 백업

장치 관리자를 통해 변경 사항을 롤백할 수 없는 경우를 대비하여 백업 AMI를 생성하는 것이 좋습니다. AWS Management Console을 사용하여 백업 AMI를 생성하려면 다음 단계를 따르세요.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances(인스턴스)를 선택합니다.
3. 드라이버 업그레이드가 필요한 인스턴스를 선택하고 인스턴스 상태 메뉴에서 인스턴스 중지를 선택합니다.
4. 인스턴스가 중지되면 인스턴스를 다시 선택합니다. 백업을 생성하려면 작업 메뉴에서 이미지 및 템플릿을 선택한 다음 이미지 생성을 선택합니다.
5. 인스턴스를 다시 시작하려면 인스턴스 상태 메뉴에서 인스턴스 시작을 선택합니다.

2단계: ENA 드라이버 설치 또는 업그레이드

AWS Systems Manager 배포자 또는 PowerShell cmdlet을 사용하여 ENA 드라이버를 설치하거나 업그레이드할 수 있습니다. 자세한 내용을 보려면 사용하고자 하는 방법에 해당하는 탭을 선택하세요.

Systems Manager Distributor

Systems Manager Distributor 기능을 사용하여 Systems Manager 관리 노드에 패키지를 배포할 수 있습니다. Systems Manager Distributor와 함께 ENA 드라이버 패키지는 한 번 또는 예약된 업데이트와 함께 설치할 수 있습니다. Systems Manager 배포자를 사용하여 ENA 드라이버 패키지(`AwsEnaNetworkDriver`)를 설치하는 방법에 대한 자세한 내용은 AWS Systems Manager 사용 설명서의 패키지 설치 또는 업데이트를 참조하세요.

PowerShell

이 섹션에서는 PowerShell cmdlet을 사용하여 인스턴스에 ENA 드라이버 패키지를 다운로드하고 설치하는 방법을 설명합니다.

옵션 1: 최신 버전 다운로드 및 추출

1. 인스턴스 연결 후 로컬 관리자로 로그인합니다.
2. invoke-webrequest cmdlet을 사용하여 최신 드라이버 패키지를 다운로드합니다.

```
PS C:\> invoke-webrequest https://ec2-windows-drivers-
downloads.s3.amazonaws.com/ENA/Latest/AwsEnaNetworkDriver.zip -
outfile $env:USERPROFILE\AwsEnaNetworkDriver.zip
```

> **Note**
>
> Windows Server 2016 또는 이전 버전을 사용 중이고 파일을 다운로드할 때 오류가 발생하는 경우 PowerShell 터미널에서 TLS 1.2를 활성화해야 할 수 있습니다. 다음 명령을 사용하여 현재 PowerShell 세션에 대해 TLS 1.2를 활성화한 다음 다시 시도해보세요.
>
> ```
> [Net.ServicePointManager]::SecurityProtocol =
>  [Net.SecurityProtocolType]::Tls12
> ```
>
> 또는 인스턴스의 브라우저 창에서 최신 드라이버 패키지를 다운로드할 수도 있습니다.

3. expand-archive cmdlet을 사용하여 인스턴스에 다운로드한 zip 아카이브를 추출합니다.

```
PS C:\> expand-archive $env:userprofile\AwsEnaNetworkDriver.zip -
DestinationPath $env:userprofile\AwsEnaNetworkDriver
```

옵션 2: 특정 버전 다운로드 및 추출

1. 인스턴스 연결 후 로컬 관리자로 로그인합니다.
2. ENA Windows 드라이버 버전 내역 표의 버전 링크에서 원하는 특정 버전의 ENA 드라이버 패키지를 다운로드합니다.
3. ZIP 아카이브를 인스턴스로 추출합니다.

PowerShell을 사용하여 ENA 드라이버 설치

설치 단계는 최신 드라이버를 다운로드했든 특정 버전을 다운로드했든 동일합니다. ENA 드라이버를 설치하려면 다음 단계를 따르세요.

1. 드라이버를 설치할 때에는 인스턴스의 `AwsEnaNetworkDriver` 디렉토리에서 `install.ps1` PowerShell 스크립트를 실행합니다. 오류가 발생하면 PowerShell 3.0 이상을 사용하고 있는지 확인합니다.
2. 설치 프로그램이 인스턴스를 자동으로 재부팅하지 않는 경우 Restart-Computer PowerShell cmdlet을 실행합니다.

```
PS C:\> Restart-Computer
```

3단계(선택 사항): 설치 후 ENA 드라이버 버전 확인

ENA 드라이버 패키지가 인스턴스에 성공적으로 설치되었는지 확인하려면 다음과 같이 새 버전을 확인할 수 있습니다.

1. 인스턴스 연결 후 로컬 관리자로 로그인합니다.
2. Windows 장치 관리자를 열려면 실행(Run) 상자에 `devmgmt.msc`를 입력합니다.
3. 확인을 선택합니다. 그러면 장치 관리자 창이 열립니다.
4. 네트워크 어댑터(Network adapters) 왼쪽의 화살표를 선택하여 목록을 확장합니다.
5. 이름을 선택하거나 Amazon Elastic Network Adapter의 컨텍스트 메뉴를 연 다음 속성(Properties)을 선택합니다. 그러면 Amazon Elastic Network Adapter 속성 대화 상자가 열립니다.

> **Note**
>
> ENA 어댑터는 모두 동일한 드라이버를 사용합니다. ENA 어댑터가 여러 개 있는 경우, 그 중 하나를 선택하여 모든 ENA 어댑터의 드라이버를 업데이트할 수 있습니다.

6. 설치된 현재 버전을 확인하려면 드라이버 탭을 열고 드라이버 버전을 확인합니다. 현재 버전이 대상 버전과 일치하지 않는 경우 Elastic Network Adapter Windows 드라이버 문제 해결을 참조하세요.

ENA 드라이버 설치 롤백

설치에 문제가 생기면 드라이버를 롤백해야 할 수도 있습니다. 인스턴스에 설치된 ENA 드라이버의 이전 버전으로 롤백하려면 다음 단계를 따르세요.

1. 인스턴스 연결 후 로컬 관리자로 로그인합니다.
2. Windows 장치 관리자를 열려면 실행(Run) 상자에 `devmgmt.msc`를 입력합니다.
3. 확인을 선택합니다. 그러면 장치 관리자 창이 열립니다.
4. 네트워크 어댑터(Network adapters) 왼쪽의 화살표를 선택하여 목록을 확장합니다.
5. 이름을 선택하거나 Amazon Elastic Network Adapter의 컨텍스트 메뉴를 연 다음 속성(Properties)을 선택합니다. 그러면 Amazon Elastic Network Adapter 속성 대화 상자가 열립니다.

> **Note**
>
> ENA 어댑터는 모두 동일한 드라이버를 사용합니다. ENA 어댑터가 여러 개 있는 경우, 그 중 하나를 선택하여 모든 ENA 어댑터의 드라이버를 업데이트할 수 있습니다.

6. 드라이버를 롤백하려면 드라이버 탭을 열고 드라이버 롤백을 선택합니다. 그러면 드라이버 패키지 롤백 창이 열립니다.

> **Note**
>
> 드라이버 탭에 드라이버 롤백 작업이 표시되지 않거나 작업을 사용할 수 없는 경우 인스턴스의 드라이버 스토어에 이전에 설치한 드라이버 패키지가 포함되어 있지 않다는 뜻입니다. 이 문제를 해결하려면 예기치 않은 ENA 드라이버 버전 설치 섹션에서 문제 해결 시나리오을 참조하여 확장하세요. 장치 드라이버 패키지 선택 프로세스에 대한 자세한 내용은 Microsoft 설명서 웹 사이트에서 Windows 디바이스에 대한 드라이버 패키지를 선택하는 방법을 참조하세요.

ENA Windows 드라이버 버전 릴리스 추적

Windows AMI에는 향상된 네트워킹을 활성화하기 위한 ENA Windows 드라이버가 포함되어 있습니다.

Windows Server 버전 2016 이상의 경우 최신 드라이버 버전을 사용하는 것이 좋습니다. 이전 버전의 Windows Server의 경우 다음 표를 참조하여 사용할 ENA 드라이버 버전을 결정합니다.

Windows Server 버전	ENA 드라이버 버전
Windows Server 2012 R2	2.6.0 이하
Windows Server 2012	2.6.0 이하
Windows Server 2008 R2	2.2.3 이하

ENA Windows 드라이버 버전 내역

다음 표에는 각 릴리스에 대한 변경 사항이 요약되어 있습니다.

드라이버 버전	세부 정보	릴리스 날짜
2.9.0	새로운 기능 • 디바이스에서 시작한 비동기 재설정 요청에 대한 지원이 추가되었습니다. • 디바이스에서 제공하는 최대 대형 LLQ 깊이 값 처리에 대한 지원이 추가되었습니다. • 디바이스 구성 오류로 인한 예상치 못한 전원 상태 전환에 대한 가시성을 높이기 위해 Windows 이벤트 뷰어에 이벤트 ID 58001을 추가했습니다. 버그 수정 • 디바이스 초기화 중 예기치 않은 재부팅을 방지하기 위해 메모리 할당 실패의 부적절한 처리를 수정했습니다. •	2024년 12월 12일

드라이버 버전	세부 정보	릴리스 날짜
	디바이스가 중지되는 동안 DPC를 대기열에 추가하여 서비스 중단 루틴의 문제를 해결하여 예기치 않은 재부팅을 방지했습니다.	
2.8.0	버그 수정 • 송신 네트워크 버퍼 목록(NBL) 처리의 전체 흐름에서 경합 상태를 수정했으며, 이는 이미 릴리스된 NBL을 해제하려고 하면 발생할 수 있는 메모리 손상으로 이어질 수 있습니다. • 모든 LSO 및 체크섬 오프로드를 비활성화할 때 예기치 않은 동작이 발생할 수 있는 L3 프로토콜의 잘못된 탐지를 수정했습니다.	2024년 9월 30일

드라이버 버전	세부 정보	릴리스 날짜
2.7.0	새로운 기능 • 제거됨: Windows Server 2012(Windows 8) 및 Windows Server 2012 R2(Windows 8.1)에 대한 지원. 이 운영 체제 버전은 AWS에서 지원이 종료되었습니다. Windows Server 2012 및 이전 버전에서는 드라이버 설치가 실패합니다. • IPv6 Tx 체크섬 계산을 디바이스로 오프로드하기 위한 지원이 추가되었습니다. • 광범위한 저지연 대기열(LLQ) 지원이 추가되었습니다. 이것은 기기 권장 사항에 따라 동적으로 활성화됩니다. 새 “WideLLQ” 레지스트리 키로 이 설정을 재정의할 수 있습니다. • Rx 오버런으로 인한 패킷 삭제 보고를 추가했습니다. 이는 수신 패킷을 위한 Rx 링의 공간이 충분하지 않음을 나타냅니다. • 디바이스의 최적이 아닌 구성 알림에 대한 지원이 추가되었습니다. Windows 이벤트 뷰어에서 이벤트 ID `59000`를 참조하십시오. 버그 수정 • 헤더가 최대 LLQ(Low Latency Queuing) 헤더 크기를 초과하는 헤더가 포함된 Tx 패킷으로 인해 발생하는 불필요한 장치 재설정을 방지하십시오.	2024년 5월 1일

드라이버 버전	세부 정보	릴리스 날짜

2.6.0	새로운 기능
	• ENA Express를 지원하는 인스턴스 유형에 대해 다음과 같은 네트워크 성능 메트릭을 추가합니다.
	  • ena_srd_mode
	  • ena_srd_tx_pkts
	  • ena_srd_eligible_tx_pkts
	  • ena_srd_rx_pkts
	  • ena_srd_resource_utilization
	• Nitro 기반 인스턴스 유형에 대해 conntrack_allowance_available 네트워크 성능 메트릭을 추가합니다.
	• RX 데이터 손상 감지로 인한 새 어댑터 재설정 사유를 추가합니다.
	• 드라이버 로깅 인프라를 업데이트합니다.
	
	버그 수정
	• CPU 부족으로 인해 네트워크 성능 메트릭 업데이트가 실패하는 경우 어댑터 재설정을 방지합니다.
	• 장치 하트비트 중단의 오탐을 방지합니다.
	• 다운그레이드 작업을 지원하도록 드라이버 설치 스크립트를 수정합니다.
		2023년 6월 20일

드라이버 버전	세부 정보	릴리스 날짜
	• 수신 오류 수 통계를 수정합니다.	
2.5.0	관련 공지 사항 ENA Windows 드라이버 버전 2.5.0은 Windows 도메인 컨트롤러에서 초기화하지 못해 롤백되었습니다. Windows 클라이언트와 Windows 서버는 영향을 받지 않습니다.	2023년 2월 17일
2.4.0	새로운 기능 • Windows Server 2022에 대한 지원을 추가합니다. • Windows Server 2008 R2에 대한 지원을 제거합니다. • 6세대 Amazon EC2 인스턴스의 성능을 개선하려면 짧은 지연 시간 큐잉(LLQ)을 항상 켜기로 설정합니다. 버그 수정 • PCW(Windows용 성능 카운터) 시스템에 네트워크 성능 지표를 게시하지 못하는 문제를 수정합니다. • 레지스트리 키 읽기 작업 중 메모리 누수를 수정합니다. • 어댑터 재설정 프로세스 중에 복구할 수 없는 오류가 발생하는 경우 무한 재설정 루프를 방지합니다.	2022년 4월 28일

드라이버 버전	세부 정보	릴리스 날짜

2.2.4	관련 공지 사항
	
	ENA Windows 드라이버 버전 2.2.4는 6세대 EC2 인스턴스의 잠재적인 성능 저하로 인해 롤백되었습니다. 다음 방법 중 하나를 사용하여 드라이버를 다운그레이드하는 것이 좋습니다.
	
	• 이전 버전 설치
	
	1. 이 테이블의 링크에서 이전 버전 패키지를 다운로드합니다(버전 2.2.3).
	2. install.ps1 PowerShell 설치 스크립트를 실행합니다.
	
	사전 및 사후 설치 단계에 대한 자세한 내용은 Windows에서 향상된 네트워킹 활성화 섹션을 참조하세요.
	
	대량 업데이트에 Amazon EC2 Systems Manager 사용
	
	• 다음 파라미터를 사용하여 SSM 문서 `AWS-ConfigureAWSPackage` 를 통해 대량 업데이트를 수행합니다.
	  • 이름: AwsEnaNetworkDriver
	  • 버전: 2.2.3
		2021년 10월 26일

드라이버 버전	세부 정보	릴리스 날짜
2.2.3	새로운 기능 • 최대 400Gbps의 인스턴스 네트워킹을 제공하는 새로운 Nitro Card에 대한 지원이 추가됩니다. 버그 수정 • 시스템 시간 변경과 시스템 시간 쿼리 ENA 드라이버의 시스템 시간 쿼리 간의 경쟁 조건을 수정하여 HW 무응답의 거짓 탐지를 방지했습니다. Windows ENA 드라이버 버전 2.2.3은 Windows Server 2008 R2를 지원하는 최종 버전입니다. ENA를 사용하는 현재 사용 가능한 인스턴스 유형은 Windows Server 2008 R2에서 계속 지원되며 드라이버를 다운로드하여 사용할 수 있습니다. 이후 인스턴스 유형은 Windows Server 2008 R2를 지원하지 않으며 Windows Server 2008 R2 이미지를 이후 인스턴스 유형으로 시작, 가져오기 또는 마이그레이션할 수 없습니다.	2021년 3월 25일
2.2.2	새로운 기능 • CloudWatch 및 Windows 소비자용 성능 카운터를 사용하여 네트워크 어댑터 성능 지표를 쿼리하는 지원을 추가합니다. 버그 수정 • 베어메탈 인스턴스의 성능 문제를 해결합니다.	2020년 12월 21일

드라이버 버전	세부 정보	릴리스 날짜
2.2.1	새로운 기능 • 호스트가 Elastic Network Adapter에 네트워크 성능 지표를 쿼리할 수 있도록 하는 메서드를 추가합니다.	2020년 10월 1일
2.2.0	새로운 기능 • 차세대 하드웨어 유형에 대한 지원을 추가합니다. • 중지-최대 절전 모드에서 재개한 후 인스턴스 시작 시간을 개선하고 오탐지 ENA 오류 메시지를 제거합니다. 성능 최적화 • 인바운드 트래픽의 처리를 최적화합니다. • 낮은 리소스 환경에서 공유 메모리 관리를 개선합니다. 버그 수정 • 드문 경우이지만 드라이버를 재설정하지 못하는 경우 ENA 디바이스 제거 시 시스템 충돌을 방지합니다.	2020년 8월 12일
2.1.5	버그 수정 • 베어 메탈 인스턴스에서 가끔 발생하는 네트워크 어댑터 초기화 실패를 해결합니다.	2020년 6월 23일

드라이버 버전	세부 정보	릴리스 날짜
2.1.4	버그 수정 • 네트워크 스택에서 도착하는 손상된 LSO 패킷 메타데이터로 인해 발생하는 연결 문제를 방지합니다. • 이미 릴리스된 패킷 메모리에 액세스하게 되는 드문 교착 상태로 인해 발생하는 시스템 충돌을 방지합니다.	2019년 11월 25일
2.1.2	새로운 기능 • OS에서 MAC 기반 UUID를 생성할 수 있도록 공급업체 ID 보고서에 대한 지원이 추가되었습니다. 버그 수정 • 초기화 시 DHCP 네트워크 구성 성능이 향상되었습니다. • 최대 전송 단위(MTU)가 4K를 초과하는 경우 인바운드 IPv6 트래픽에서 L4 체크섬을 적절히 계산합니다. • 드라이버 안정성 및 사소한 버그 수정에 대한 전반적인 개선 사항입니다.	2019년 11월 4일
2.1.1	버그 수정 • 운영 체제에서 매우 조각화된 TCP LSO 패킷이 떨어지는 것을 방지합니다. • IPv6 네트워크의 IPSec 내에서 캡슐화 보안 페이로드(ESP) 프로토콜을 적절히 처리합니다.	2019년 9월 16일

드라이버 버전	세부 정보	릴리스 날짜

2.1.0	ENA Windows 드라이버 v2.1은 새로운 ENA 디바이스 기능을 도입하고 성능 향상을 제공하며 새로운 기능을 추가하고 여러 안정성 개선 기능을 포함합니다.
	• 새로운 기능
	  • 점보 프레임 구성에 표준화된 Windows 레지스트리 키를 사용합니다.
	  • ENA 드라이버 속성 GUI를 통한 VLAN ID 설정을 허용합니다.
	  • 복구 흐름이 개선되었습니다.
	  • 결함 식별 메커니즘이 개선되었습니다.
	  • 튜닝 가능한 복구 파라미터에 대한 지원이 추가되었습니다.
	  • vCPU가 8개 이상인 최신 EC2 인스턴스의 경우 최대 32개의 I/O 대기열을 지원합니다.
	  • 드라이버 메모리 공간 90%까지 절감
	• 성능 최적화
	  • 전송 경로 지연 시간 감소
	  • 수신 체크섬 오프로드를 지원합니다.
	  • 과다 로드된 시스템(잠금 메커니즘의 사용 최적화)을 위한 성능 최적화
	  • CPU 사용률을 줄이고 로드 시 시스템 응답 속도를 개선하는 추가 향상 기능
		2019년 7월 1일

드라이버 버전	세부 정보	릴리스 날짜

	• 버그 수정
	  • 불연속 Tx 헤더의 유효하지 않은 구문 분석으로 인한 충돌을 수정했습니다.
	  • Bare Metal 인스턴스에서 탄력적 네트워크 인터페이스 분리 중 드라이버 v1.5 충돌 문제를 수정했습니다.
	  • IPv6에 대한 LSO 의사 헤더 체크섬 계산 오류를 수정했습니다.
	  • 초기화 실패 시 잠재적인 메모리 리소스 유출을 수정했습니다.
	  • IPv4 조각에 대한 TCP/UDP 체크섬 오프로드를 비활성화했습니다.
	  • VLAN 구성에 대해 수정했습니다. VLAN 우선순위만 비활성화해야 한 경우 VLAN이 잘못 비활성화되었습니다.
	  • 이벤트 뷰어로 사용자 지정 드라이버 메시지의 올바른 구문 분석을 활성화했습니다.
	  • 잘못된 타임스탬프 처리로 인한 드라이버 초기화 실패 문제를 수정했습니다.
	  • 데이터 처리 및 ENA 디바이스 비활성화 사이의 교착 상태를 수정했습니다.	

드라이버 버전	세부 정보	릴리스 날짜
1.5.0	• 안정성 및 성능 수정 사항이 개선되었습니다. • 이제, ENA NIC의 고급 속성에서 수신 버퍼를 최대 8192의 값으로 구성할 수 있습니다. • 기본 수신 버퍼는 1k입니다.	2018년 10월 4일
1.2.3	안정성 수정 사항이 포함되고, Windows Server 2008 R2부터 Windows Server 2016에 이르는 지원을 통합합니다.	2018년 2월 13일
1.0.8	최초 릴리스입니다. Windows Server 2008 R2, Windows Server 2012 RTM, Windows Server 2012 R2 및 Windows Server 2016용 AMI에 포함됩니다.	2016년 7월

Amazon SNS에서 ENA Windows 드라이버 릴리스 알림 구독

새로운 EC2 Windows Driver 버전이 릴리스되면 이를 알리도록 Amazon SNS를 설정할 수 있습니다. 알림을 받으려면 다음 절차를 수행합니다.

EC2 알림 구독

1. https://console.aws.amazon.com/sns/v3/home에서 Amazon SNS 콘솔을 엽니다.
2. 필요한 경우 탐색 모음에서 리전을 미국 동부(버지니아 북부)로 변경합니다. 구독을 신청하는 SNS 알림이 이 지역에 있기 때문에 이 지역을 선택해야 합니다.
3. 탐색 창에서 구독을 선택합니다.
4. 구독 생성을 선택합니다.
5. 구독 생성 대화 상자에서 다음 작업을 수행합니다.
   a. TopicARN의 경우, 다음 Amazon 리소스 이름(ARN)을 복사합니다.

      arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers
   b. 프로토콜에서 `Email`을 선택합니다.
   c. 엔드포인트에 알림을 보낼 이메일 주소를 입력합니다.

d. Create subscription을 선택합니다.

6. 확인 이메일이 발송됩니다. 이메일을 열고 지침에 따라 구독을 완료합니다.

새 EC2 Windows 드라이버가 릴리스될 때마다 구독자에게 알림이 전송됩니다. 이런 알림을 더 이상 받지 않기를 원하는 경우, 다음 절차를 수행해서 구독을 해제하세요.

Amazon EC2 Windows 드라이버 알림 구독 해제

1. https://console.aws.amazon.com/sns/v3/home에서 Amazon SNS 콘솔을 엽니다.
2. 탐색 창에서 구독을 선택합니다.
3. 구독 확인란을 선택한 후 작업, 구독 삭제를 선택합니다. 확인 메시지가 나타나면 Delete(삭제)를 선택합니다.

Windows 인스턴스의 반가상화 드라이버

Windows AMI는 가상화 하드웨어에 대한 액세스를 허용하는 드라이버 세트를 포함하고 있습니다. 이 드라이버는 Amazon EC2에 의해 사용되어 인스턴스 스토어 및 Amazon EBS 볼륨을 해당 디바이스로 매핑합니다. 다음 표는 드라이버 간의 주요 차이점을 보여줍니다.

	Red Hat PV	Citrix PV	AWS PV
인스턴스 유형	모든 인스턴스 유형에서 지원되는 것은 아님. 사용자가 지원되지 않는 인스턴스를 지정한 경우 인스턴스가 손상됩니다.	Xen 인스턴스 유형에 지원됩니다.	Xen 인스턴스 유형에 지원됩니다.
연결된 볼륨	최대 16개 볼륨 연결 지원.	16개 이상 볼륨 연결 지원.	16개 이상 볼륨 연결 지원.
네트워크	이 드라이버에는 부하가 높은 경우(예: 빠른 FTP 파일 전송) 네트워크 연결이 초기화되는 알려진 문제가 있습니다.		호환되는 인스턴스 유형인 경우 드라이버는 네트워크 어댑터에서 점보 프레임을 자동으로 구성

	Red Hat PV	Citrix PV	AWS PV
			합니다. 인스턴스가 클러스터 배치 그룹에 있는 경우 클러스터 배치 그룹에 있는 인스턴스 간에 더 나은 네트워크 성능을 제공합니다. 자세한 내용은 Amazon EC2 인스턴스에 대한 배치 그룹 섹션을 참조하세요.

다음 표에서는 Amazon EC2의 각 Windows Server 버전에서 어떤 PV 드라이버를 실행해야 하는지를 보여줍니다.

Windows Server 버전	PV 드라이버 버전
Windows Server 2025	지원되지 않음
Windows Server 2022	AWS PV 최신 버전
Windows Server 2019	AWS PV 최신 버전
Windows Server 2016	AWS PV 최신 버전
Windows Server 2012 R2	AWS PV 버전 8.4.3
Windows Server 2012	AWS PV 버전 8.4.3

Windows Server 버전	PV 드라이버 버전
Windows Server 2008 R2	AWS PV 버전 8.3.5
Windows Server 2008	Citrix PV 5.9
Windows Server 2003	Citrix PV 5.9

목차

- AWSPV 드라이버
- Citrix PV 드라이버
- Red Hat PV 드라이버
- 알림 구독
- EC2 Windows 인스턴스의 PV 드라이버 업그레이드
- Windows 인스턴스에서 PV 드라이버 문제 해결

AWSPV 드라이버

AWS PV 드라이버는 `%ProgramFiles%\Amazon\Xentools` 디렉터리에 저장됩니다. 이 디렉터리에는 퍼블릭 기호 및 `xenstore_client.exe` 명령줄 도구가 포함되어 사용자는 XenStore의 항목에 액세스할 수 있습니다. 예를 들어, 다음 PowerShell 명령은 하이퍼바이저에서 현재 시간을 반환합니다.

```
PS C:\> [DateTime]::FromFileTimeUTC((gwmi -n root\wmi -cl
 AWSXenStoreBase).XenTime).ToString("hh:mm:ss")
11:17:00
```

AWS PV 드라이버 구성 요소는 `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services` 아래 Windows 레지스트리에 나열됩니다. 이러한 드라이버 구성 요소로는 xenbus, xeniface, xennet, xenvbd 및 xenvif 등이 있습니다.

AWS또한 PV 드라이버에는 사용자 모드에서 실행되는 LiteAgent라는 Windows 서비스가 있습니다. 이 서비스는 Xen 세대 인스턴스의 AWS API에서 이벤트 종료 및 재시작과 같은 작업을 수행합니다. 사용자는 명령줄에서 `Services.msc`를 실행하여 서비스에 액세스하고 관리할 수 있습니다. Nitro 세대 인스턴스에서 실행할 때는 AWS PV 드라이버가 사용되지 않으며, 드라이버 버전 8.2.4에서부터는 LiteAgent 서비스가 자동 중지됩니다. 또한 최신 AWS PV 드라이버로 업데이트하면 LiteAgent가 업데이트되고 모든 인스턴스 세대에서 신뢰성을 높일 수 있습니다.

최신 AWS PV 드라이버 설치

Amazon Windows AMI는 가상화 하드웨어에 대한 액세스를 허용하는 드라이버 세트를 포함하고 있습니다. 이 드라이버는 Amazon EC2에 의해 사용되어 인스턴스 스토어 및 Amazon EBS 볼륨을 해당 디바이스로 매핑합니다. EC2 Windows 인스턴스의 안정성과 성능을 향상하려면 최신 드라이버를 설치하는 것이 좋습니다.

설치 옵션

- AWS Systems Manager를 사용하여 PV 드라이버를 자동으로 업데이트합니다. 자세한 내용은 AWS Systems Manager 사용 설명서에서 연습: EC2 Windows 인스턴스에서 PV 드라이버 자동 업데이트를 참조하세요.
- 다운로드 드라이버 패키지를 하고 설치 프로그램을 수동으로 실행합니다. `readme.txt` 파일에서 시스템 요구 사항을 확인하세요. AWS PV 드라이버 다운로드 및 설치 또는 도메인 컨트롤러 업그레이드에 대한 자세한 내용은 수동으로 Windows Server 인스턴스 업그레이드(AWS PV 업그레이드) 섹션을 참조하세요.

AWS PV 드라이버 패키지 기록

다음 표에서는 각 드라이버 릴리스에 대한 AWS PV 드라이버의 변경 사항을 보여 줍니다.

패키지 버전	세부 정보	릴리스 날짜
8.6.0	• XenStore 상호 작용 안정성 수정.	2025년 5월 27일
8.5.0	• 네트워크 디바이스 분리 중 드물게 발생하는 충돌을 해결하기 위해 안정성 수정. • EBS 볼륨 분리 중 드물게 발생하는 충돌을 해결하기 위해 안정성 수정. • 패키지 설치 프로그램의 버그를 수정했습니다. • `Pnputil`을 사용하도록 PV 설치 관리자를 업데이트했습니다.	2024년 10월 31일
8.4.3	업그레이드 환경을 개선하기 위해 패키지 설치 관리자의 버그를 수정했습니다. Windows Server 2012 및 2012 R2에서 실행할 수 있는 마지막 버전입니다. 이 버전은 다운로드할 수 있지만	2023년 1월 24일

패키지 버전	세부 정보	릴리스 날짜
	Windows Server 2012 및 2012 R2의 수명이 종료되어 더 이상 지원되지 않습니다.	
8.4.2	교착 상태가 해결되도록 안정성이 수정됩니다.	2022년 4월 13일
8.4.1	패키지 설치 프로그램이 개선되었습니다.	2022년 1월 7일
8.4.0	• 드물게 발생하는 디스크 IO 장애를 해결하기 위해 안정성 수정. • EBS 볼륨 분리 중 드물게 발생하는 충돌을 해결하기 위해 안정성 수정. • 20,000 IOPS 이상을 활용하고 병목 현상으로 인한 성능 저하를 경험하는 워크로드를 위해 여러 코어에 부하를 분산하는 기능이 추가되었습니다. 이 기능을 사용하려면 20,000 디스크 IOPS 이상을 활용하는 워크로드에서 CPU 병목 현상으로 인해 성능 저하 발생 섹션을 참조하세요.	2021년 3월 2일
8.3.5	패키지 설치 프로그램이 개선되었습니다. Windows Server 2008 R2에서 실행할 수 있는 마지막 버전입니다. 이 버전은 다운로드할 수 있지만 더 이상 지원되지 않습니다. Windows Server 2008 R2는 수명이 종료되어 더 이상 Microsoft에서 지원하지 않습니다.	2022년 1월 7일
8.3.4	네트워크 디바이스 연결의 안정성이 향상되었습니다.	2020년 8월 4일

패키지 버전	세부 정보	릴리스 날짜
8.3.3	• 오류 처리 경로 중 버그 검사를 방지하기 위해 XenStore 지향 구성 요소로 업데이트합니다. • 잘못된 SRB가 제출될 때 충돌을 방지하기 위해 스토리지 구성 요소로 업데이트합니다. Windows Server 2008 R2 인스턴스에서 이 드라이버를 업데이트하려면 먼저 Microsoft 보안 공지 사항(Microsoft Security Advisory 3033929)을 해결하기 위해 적절한 패치가 설치되어 있는지 확인해야 합니다.	2020년 2월 4일
8.3.2	네트워킹 구성 요소 안정성이 향상되었습니다.	2019년 7월 30일
8.3.1	스토리지 구성 요소의 성능과 견고성을 개선했습니다.	2019년 6월 12일
8.2.7	최신 세대 인스턴스 유형으로의 마이그레이션을 지원하는 향상된 효율성.	2019년 5월 20일
8.2.6	충돌 덤프 경로의 효율성을 개선했습니다.	2019년 1월 15일
8.2.5	추가 보안 개선사항 이제 패키지에서 PowerShell 설치 관리자를 이용할 수 있습니다.	2018년 12월 12일
8.2.4	안정성 개선.	2018년 10월 2일
8.2.3	버그 수정 및 성능 향상. EBS 볼륨 ID를 EBS 볼륨의 디스크 일련 번호로 보고합니다. 이렇게 하면 S2D 같은 클러스터 시나리오가 활성화됩니다.	2018년 5월 29일

패키지 버전	세부 정보	릴리스 날짜
8.2.1	네트워크 및 스토리지 성능 개선 및 향상을 위한 다양한 수정. 이 버전이 설치되어 있는지 확인하려면 Windows 레지스트리 값 `HKLM\Software\Amazon\PVDriver\Version` `8.2.1`을 참조하세요.	2018년 3월 8일
7.4.3	Windows Server 2016에 대한 지원 추가됨. 지원되는 모든 Windows OS 버전에 대한 안정성 수정. *AWS PV 드라이버 버전 7.4.3의 서명은 2019년 3월 29일 만료됩니다. 최신 AWS PV 드라이버 업데이트를 권장합니다.	2016년 11월 18일
7.4.2	X1 인스턴스 유형의 지원에 대한 안정성 수정.	2016년 8월 2일
7.4.1	• AWS PV 스토리지 드라이버의 성능 향상. • AWS PV 스토리지 드라이버의 안정성 수정: 인스턴스가 버그 검사 코드 0x0000DEAD로 시스템 충돌을 일으키는 문제 수정됨. • AWS PV 네트워크 드라이버의 안정성 수정. • Windows Server 2008R2에 대한 지원 추가됨.	2016년 7월 12일
7.3.2	• 로깅 및 진단 개선됨. • AWS PV 스토리지 드라이버의 안정성 수정. 경우에 따라 인스턴스에 디스크를 다시 연결한 후 디스크가 Windows에서 표시되지 않을 수 있습니다. • Windows Server 2012에 대한 지원 추가됨.	2015년 6월 24일
7.3.1	TRIM 업데이트: TRIM 요청과 관련하여 수정이 이루어졌습니다. 이 업데이트는 많은 수의 TRIM 요청을 관리할 때 인스턴스를 안정화하고 인스턴스 성능을 높입니다.	

패키지 버전	세부 정보	릴리스 날짜
7.3.0	TRIM 지원: 이제 AWS PV 드라이버가 TRIM 요청을 하이퍼바이저에 전송합니다. 기본 스토리지에서 TRIM(SSD)을 지원할 경우 휘발성 디스크가 TRIM 요청을 제대로 처리합니다. 2015년 3월을 기준으로 EBS 기반 스토리지에서 TRIM을 지원하지 않습니다.	
7.2.5	• AWS PV 스토리지 드라이버의 안정성 수정: 경우에 따라 AWS PV 드라이버가 유효하지 않은 메모리를 역참조하고 시스템 오류를 유발할 수 있습니다. • 충돌 덤프를 생성하는 중 안정성 수정: 경우에 따라 AWS PV 드라이버가 충돌 덤프를 작성할 때 경합 상태로 멈출 수 있습니다. 이 릴리스 전에는 드라이버를 강제로 중지하고 다시 시작하여 문제를 해결할 수 있었지만 메모리 덤프가 손실되었습니다.	
7.2.4	디바이스 ID 지속성: 이 드라이버 수정은 플랫폼 PCI 디바이스 ID를 숨기고 인스턴스가 이동된 경우에도 시스템이 항상 동일한 디바이스 ID를 표시하도록 강제 적용합니다. 이러한 수정 사항은 대체로 하이퍼바이저가 가상 디바이스를 표시하는 방법에 영향을 미치며, AWS PV 드라이버의 공동 설치 관리자에 대한 수정 사항도 포함하므로 시스템이 매핑된 가상 디바이스를 유지합니다.	
7.2.2	• DSRM(디렉터리 서비스 복원 모드) 모드에서 AWS PV 드라이버 로드:디렉터리 서비스 복원 모드는 Windows 서버 도메인 컨트롤러에 사용할 수 있는 안전 모드 부팅 옵션입니다. • 네트워크 어댑터 디바이스가 다시 연결된 경우 영구 디바이스 ID: 이 수정 사항은 시스템이 MAC 주소 매핑을 확인하고 디바이스 ID를 유지하도록 강제 적용합니다. 또한 이를 통해 어댑터가 다시 연결된 경우 해당 정적 설정을 유지할 수 있습니다.	

패키지 버전	세부 정보	릴리스 날짜
7.2.1	• 안전 모드에서 실행: 드라이버가 안전 모드에서 로드되지 않는 문제를 해결했습니다. 이전에는 AWS PV 드라이버가 정상 실행 중인 시스템에서만 인스턴스화했습니다. • Microsoft Windows 스토리지 풀에 디스크 추가: 이전에는 페이지 83 쿼리를 합성했으며, 수정 사항으로 페이지 83 지원이 비활성화됐습니다. 이는 PV 디스크가 유효한 클러스터 디스크가 아니므로 클러스터 환경에서 사용되는 스토리지 풀에 영향을 미치지 않습니다.	
7.2.0	기본: AWS PV 기본 버전입니다.	

Citrix PV 드라이버

Citrix PV 드라이버는 `%ProgramFiles%\Citrix\XenTools`(32비트 인스턴트의 경우) 또는 `%ProgramFiles(x86)%\Citrix\XenTools`(64비트 인스턴스의 경우) 디렉터리에 저장됩니다.

Citrix PV 드라이버 구성 요소는 Windows 레지스트리의 `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services` 아래에 나열됩니다. 이러한 드라이버 구성요소로는 xenevtchn, xeniface, xennet, Xennet6, xensvc, xenvbd 및 xenvif 등이 있습니다.

또한, Citrix에는 Windows 서비스를 구동하는 드라이버 구성요소인 XenGuestAgent가 있습니다. LiteAgent는 API 이벤트 종료 및 재시작과 같은 작업을 수행합니다. 사용자는 명령줄에서 `Services.msc`를 실행하여 서비스에 액세스하고 관리할 수 있습니다.

특정 워크로드 실행 시 네트워크 오류가 발생한 경우 Citrix PV 드라이버에서 TCP 오프로딩 기능을 비활성화해야 합니다. 자세한 내용은 TCP 오프로드 섹션을 참조하세요.

Red Hat PV 드라이버

Red Hat 드라이버는 레거시 인스턴스에 사용할 수 있도록 지원되지만, 드라이버 제한 사항으로 인해 RAM이 12GB 이상인 새로운 인스턴스에서는 사용하지 않는 것이 좋습니다. Red Hat 드라이버를 실행 중인 RAM이 12GB보다 큰 인스턴스는 부팅에 실패하고 액세스할 수 없는 상태가 될 수 있습니다. Red Hat 드라이버를 Citrix PV 드라이버로 업그레이드한 다음 Citrix PV 드라이버를 AWS PV 드라이버로 업그레이드하는 것이 좋습니다.

Red Hat 드라이버의 소스 파일은 `%ProgramFiles%\RedHat`(32비트 인스턴스의 경우) 또는 `%ProgramFiles(x86)%\RedHat`(64비트 인스턴스의 경우) 디렉터리에 저장됩니다. 드라이버

로는 Red Hat 반가상화 네트워크 드라이버인 `rhelnet`과 Red Hat SCSI 미니포트 드라이버인 `rhelscsi`의 두 가지가 있습니다.

알림 구독

새로운 EC2 Windows Driver 버전이 릴리스되면 이를 알리도록 Amazon SNS를 설정할 수 있습니다. 이러한 알림을 구독할 수 있습니다.

> **Note**
>
> SNS 주제가 생성된 리전을 지정해야 합니다.

새 EC2 Windows 드라이버가 릴리스될 때마다 구독자에게 알림이 전송됩니다. 이런 알림을 더 이상 받지 않기를 원하는 경우 구독을 취소할 수 있습니다. 자세한 내용은 SNS 주제 및 구독 삭제를 참조하세요.

Console

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 변경합니다. 구독을 신청하는 SNS 알림이 이 지역에 있기 때문에 이 지역을 선택해야 합니다.
3. 탐색 창에서 구독을 선택합니다.
4. 구독 생성을 선택합니다.
5. 구독 생성 대화 상자에서 다음 작업을 수행합니다.
   a. TopicARN의 경우, 다음 Amazon 리소스 이름(ARN)을 복사합니다.

      arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers
   b. 프로토콜에서 `Email`을 선택합니다.
   c. 엔드포인트에서 알림을 받을 이메일 주소를 입력합니다.
   d. Create subscription을 선택합니다.
6. 확인 이메일이 발송됩니다. 이메일을 열고 지침에 따라 구독을 완료합니다.

AWS CLI

알림을 구독하려면

다음 명령을 사용합니다.

```
aws sns subscribe \
    --topic-arn arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers \
    --region us-east-1 \
    --protocol email \
    --notification-endpoint YourUserName@YourDomainName.ext
```

PowerShell

알림을 구독하려면

다음 명령을 사용합니다.

```
Connect-SNSNotification `
    -TopicArn 'arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers' `
    -Region us-east-1 `
    -Protocol email `
    -Endpoint "YourUserName@YourDomainName.ext"
```

EC2 Windows 인스턴스의 PV 드라이버 업그레이드

EC2 Windows 인스턴스의 안정성과 성능을 향상하려면 최신 PV 드라이버를 설치하는 것이 좋습니다. 이 페이지의 지침은 드라이버 패키지를 다운로드하고 설치 프로그램을 실행하는 데 도움이 됩니다.

Windows 인스턴스에서 사용하는 드라이버를 확인하려면

디바이스 관리자를 열고 네트워크 어댑터를 표시합니다. PV 드라이버가 다음 중 하나에 해당하는지 확인합니다.

- AWS PV 네트워크 디바이스
- Citrix PV 이더넷 어댑터
- Red Hat PV NIC 드라이버

시스템 요구 사항

다운로드의 `readme.txt` 파일에서 시스템 요구 사항을 확인하세요.

내용

- Distributor를 사용하여 Windows Server 인스턴스 업그레이드(AWS PV 업그레이드)
- 수동으로 Windows Server 인스턴스 업그레이드(AWS PV 업그레이드)
- 도메인 컨트롤러 업그레이드(AWS PV 업그레이드)
- Windows Server 2008 및 2008 R2 인스턴스 업그레이드(Red Hat에서 Citrix PV로 업그레이드)
- Citrix Xen 게스트 에이전트 서비스 업그레이드

Distributor를 사용하여 Windows Server 인스턴스 업그레이드(AWS PV 업그레이드)

AWS Systems Manager의 기능인 Distributor를 사용하여 AWS PV 드라이버 패키지를 설치하거나 업그레이드할 수 있습니다. 설치 또는 업그레이드는 한 번 수행하거나 일정에 따라 설치 또는 업데이트할 수 있습니다. 이 Distributor 패키지에 대해서는 설치 유형에 대한 `In-place update` 옵션이 지원되지 않습니다.

> **Important**
>
> 인스턴스가 도메인 컨트롤러인 경우 도메인 컨트롤러 업그레이드(AWS PV 업그레이드) 섹션을 참조하세요. 도메인 컨트롤러 인스턴스의 업그레이드 프로세스는 Windows의 표준 버전과 다릅니다.

1. 변경 내용을 롤백해야 하는 경우를 대비하여 백업을 생성하는 것이 좋습니다.

   > **Tip**
   >
   > Amazon EC2 콘솔에서 AMI를 생성하는 대신 Systems Manager Automation을 사용하여 `AWS-CreateImage` 런북으로 AMI를 생성할 수 있습니다. 자세한 내용은 AWS Systems Manager 자동화 런북 레퍼런스 사용 설명서의 AWS-CreateImage 섹션을 참조하세요.

   a. 인스턴스를 중지하면 인스턴스 스토어 볼륨의 데이터가 삭제됩니다. 인스턴스를 중지하기 전에 필요한 데이터를 인스턴스 스토어 볼륨에서 영구 스토리지(예: Amazon EBS 또는 Amazon S3)로 복사했는지 확인합니다.

   b. 탐색 창에서 인스턴스를 선택합니다.

   c. 드라이버 업그레이드가 필요한 인스턴스를 선택하고 [인스턴스 상태(Instance state)], [인스턴스 중지(Stop instances)]를 선택합니다.

d. 인스턴스가 중지되면 [작업(Actions)], [이미지 및 템플릿(Image and templates)] 및 [이미지 생성(Create image)]을 차례로 선택합니다.

e. 인스턴스 상태, 인스턴스 시작을 차례로 선택합니다.

2. 원격 데스크톱을 사용하여 인스턴스에 연결합니다. 자세한 내용은 the section called “RDP 클라이언트를 사용하여 연결” 섹션을 참조하세요.
3. 이 업그레이드를 수행하기 전에 모든 비 시스템 디스크를 오프라인으로 하고 모든 드라이브 문자 매핑을 디스크 관리의 보조 디스크에 기록하는 것이 좋습니다. AWS PV 드라이버의 현재 위치 업그레이드를 수행할 경우에는 이 단계가 필요하지 않습니다. 또한 서비스 콘솔에서 필수적이지 않은 서비스를 수동 시작으로 설정하는 것이 좋습니다.
4. Distributor를 사용하여 AWS PV 드라이버 패키지를 설치하거나 업그레이드하는 방법에 대한 지침은 AWS Systems Manager 사용 설명서의 Install or update packages 절차를 참조하세요.
5. 이름에서 AWSPVDriver를 선택합니다.
6. 설치 유형에서 제거 및 다시 설치를 선택합니다.
7. 필요에 따라 패키지의 다른 파라미터를 구성하고 Step 4에서 참조된 절차를 사용하여 설치 또는 업그레이드를 실행합니다.

   Distributor 패키지를 실행하면 인스턴스가 자동으로 재부팅되고 드라이버를 업그레이드합니다. 최대 15분 동안 인스턴스를 사용할 수 없습니다.
8. 업그레이드가 완료되고 인스턴스가 Amazon EC2 콘솔에서 두 상태 확인을 모두 통과하면 원격 데스크톱을 사용하여 인스턴스에 연결하고 새 드라이버가 설치되었는지 확인합니다.
9. 연결이 완료되면 다음 PowerShell 명령을 실행합니다.

```
Get-ItemProperty HKLM:\SOFTWARE\Amazon\PVDriver
```

10. 드라이버 버전이 드라이버 버전 기록 표에 나열된 최신 버전과 동일한지 확인합니다. 자세한 내용은 AWS PV 드라이버 패키지 기록 개방형 디스크 관리를 참조하여 모든 오프라인 보조 볼륨을 검토하고 Step 3에서 기록한 드라이버 문자에 따라 이를 온라인으로 전환합니다.

Citrix PV 드라이버용 Netsh를 사용하여 이전에 TCP 오프로드을 비활성화한 경우 AWS PV 드라이버로 업그레이드한 후 이 기능을 다시 활성화하는 것이 좋습니다. Citrix 드라이버의 TCP 오프로딩 문제가 AWS PV 드라이버에는 없습니다. 따라서 TCP 오프로딩은 AWS PV 드라이버를 사용할 때 더 우수한 성능을 제공합니다.

이전에 네트워크 인터페이스에 정적 IP 주소 또는 DNS 구성을 적용한 경우 AWS PV 드라이버 업그레이드 이후 네트워크 인터페이스에 정적 IP 주소 또는 DNS 구성을 다시 적용해야 할 수 있습니다.

수동으로 Windows Server 인스턴스 업그레이드(AWS PV 업그레이드)

Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016, Windows Server 2019 또는 Windows Server 2022에서 AWS PV 드라이버의 인플레이스 업그레이드를 수행하거나 Citrix PV 드라이버에서 AWS PV 드라이버로 업그레이드하려면 다음 절차를 사용하세요. 이 업그레이드는 Red Hat 드라이버 또는 다른 버전의 Windows Server에서는 제공되지 않습니다.

일부 이전 버전의 Windows Server에서는 최신 드라이버를 사용할 수 없습니다. 운영 체제에 사용할 드라이버 버전을 확인하려면 Windows 인스턴스의 반가상화 드라이버페이지의 드라이버 버전 표를 참조하세요.

> **Important**
>
> 인스턴스가 도메인 컨트롤러인 경우 도메인 컨트롤러 업그레이드(AWS PV 업그레이드) 섹션을 참조하세요. 도메인 컨트롤러 인스턴스의 업그레이드 프로세스는 Windows의 표준 버전과 다릅니다.

수동으로 AWS PV 드라이버 업그레이드

1. 변경 내용을 롤백해야 하는 경우를 대비하여 백업을 생성하는 것이 좋습니다.

   > **Tip**
   >
   > Amazon EC2 콘솔에서 AMI를 생성하는 대신 Systems Manager Automation을 사용하여 `AWS-CreateImage` 런북으로 AMI를 생성할 수 있습니다. 자세한 내용은 AWS Systems Manager 자동화 런북 레퍼런스 사용 설명서의 AWS-CreateImage 섹션을 참조하세요.

   a. 인스턴스를 중지하면 인스턴스 스토어 볼륨의 데이터가 삭제됩니다. 인스턴스를 중지하기 전에 필요한 데이터를 인스턴스 스토어 볼륨에서 영구 스토리지(예: Amazon EBS 또는 Amazon S3)로 복사했는지 확인합니다.
   b. 탐색 창에서 인스턴스를 선택합니다.
   c. 드라이버 업그레이드가 필요한 인스턴스를 선택하고 [인스턴스 상태(Instance state)], [인스턴스 중지(Stop instances)]를 선택합니다.
   d. 인스턴스가 중지되면 [작업(Actions)], [이미지 및 템플릿(Image and templates)] 및 [이미지 생성(Create image)]을 차례로 선택합니다.
   e. 인스턴스 상태, 인스턴스 시작을 차례로 선택합니다.

2. 원격 데스크톱을 사용하여 인스턴스에 연결합니다.
3. 이 업그레이드를 수행하기 전에 모든 비 시스템 디스크를 오프라인으로 하고 모든 드라이브 문자 매핑을 디스크 관리의 보조 디스크에 기록하는 것이 좋습니다. AWS PV 드라이버의 현재 위치 업그레이드를 수행할 경우에는 이 단계가 필요하지 않습니다. 또한 서비스 콘솔에서 필수적이지 않은 서비스를 수동 시작으로 설정하는 것이 좋습니다.
4. 다음 옵션 중 하나를 사용하여 드라이버를 인스턴스에 다운로드합니다.
   - 브라우저 - 다운로드 인스턴스에 최신 드라이버 패키지를 하고 zip 아카이브를 추출합니다.
   - PowerShell - 다음 명령을 실행합니다.

     ```
     Invoke-WebRequest https://s3.amazonaws.com/ec2-windows-drivers-downloads/AWSPV/
     Latest/AWSPVDriver.zip -outfile $env:USERPROFILE\pv_driver.zip
     Expand-Archive $env:userprofile\pv_driver.zip -DestinationPath $env:userprofile
     \pv_drivers
     ```

     Windows Server 2016 또는 이전 버전을 사용 중이고 파일을 다운로드할 때 오류가 발생하는 경우 PowerShell 터미널에서 TLS 1.2를 활성화해야 할 수 있습니다. 다음 명령을 사용하여 현재 PowerShell 세션에 대해 TLS 1.2를 활성화한 다음 다시 시도해보세요.

     ```
     [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
     ```
5. `AWSPVDriverSetup.msi`을(를) 실행합니다.

MSI를 실행하면 인스턴스가 자동으로 재부팅되고 드라이버를 업그레이드합니다. 최대 15분 동안 인스턴스를 사용할 수 없습니다. 업그레이드를 완료하고 인스턴스가 Amazon EC2 콘솔에서 두 상태 확인을 모두 통과하면 원격 데스크톱을 사용하여 인스턴스에 연결한 다음, PowerShell 명령을 실행하여 새 드라이버가 설치되었는지 확인할 수 있습니다.

```
Get-ItemProperty HKLM:\SOFTWARE\Amazon\PVDriver
```

드라이버 버전이 드라이버 버전 기록 표에 나열된 최신 버전과 동일한지 확인합니다. 자세한 내용은 AWS PV 드라이버 패키지 기록 개방형 디스크 관리를 참조하여 모든 오프라인 보조 볼륨을 검토하고 Step 3에서 기록한 드라이버 문자에 따라 이를 온라인으로 전환합니다.

Citrix PV 드라이버용 Netsh를 사용하여 이전에 TCP 오프로드을 비활성화한 경우 AWS PV 드라이버로 업그레이드한 후 이 기능을 다시 활성화하는 것이 좋습니다. Citrix 드라이버의 TCP 오프로딩 문제가 AWS PV 드라이버에는 없습니다. 따라서 TCP 오프로딩은 AWS PV 드라이버를 사용할 때 더 우수한 성능을 제공합니다.

이전에 네트워크 인터페이스에 정적 IP 주소 또는 DNS 구성을 적용한 경우 AWS PV 드라이버 업그레이드 이후 네트워크 인터페이스에 정적 IP 주소 또는 DNS 구성을 다시 적용해야 할 수 있습니다.

도메인 컨트롤러 업그레이드(AWS PV 업그레이드)

도메인 컨트롤러에서 다음 절차를 사용하여 AWS PV 드라이버의 현재 위치 업그레이드를 수행하거나 Citrix PV 드라이버를 AWS PV 드라이버로 업그레이드합니다. 업그레이드 중에 FSMO 역할이 계속 작동하도록 하려면 업그레이드를 시작하기 전에 해당 역할을 다른 도메인 컨트롤러로 이전하는 것이 좋습니다. 자세한 내용은 Microsoft Learn 웹 사이트의 FSMO 역할을 보고 전송하는 방법을 참조하세요.

도메인 컨트롤러를 업그레이드하려면

1. 변경 내용을 롤백해야 하는 경우를 대비하여 도메인 컨트롤러의 백업을 생성하는 것이 좋습니다. AMI를 백업으로 사용하는 것은 지원되지 않습니다. 자세한 내용은 Microsoft 설명서의 백업 및 복원 고려 사항을 참조하세요.
2. 다음 명령을 실행하여 Windows를 Directory Services Restore Mode(DSRM)로 부팅하도록 구성합니다.

   > **Warning**
   >
   > 이 명령을 실행하기 전에 DSRM 암호를 알고 있는지 확인합니다. 업그레이드가 완료되고 인스턴스가 자동으로 재부팅된 후 인스턴스에 로그인할 수 있도록 이 정보가 필요합니다.

   ```
   bcdedit /set {default} safeboot dsrepair
   ```

   PowerShell:

   ```
   PS C:\> bcdedit /set "{default}" safeboot dsrepair
   ```

   시스템을 DSRM으로 부팅해야 합니다. 업그레이드 유틸리티가 AWS PV 드라이버를 설치하기 위해 Citrix PV 스토리지 드라이버를 제거하기 때문입니다. 따라서 모든 드라이버 문자 및 폴더 매핑을 디스크 관리의 보조 디스크에 기록하는 것이 좋습니다. Citrix PV 스토리지 드라이버가 존재하지 않으면 보조 드라이브가 검색되지 않습니다. 보조 디스크가 검색되지 않으므로 보조 드라이브의 NTDS 폴더를 사용하는 도메인 컨트롤러가 부팅되지 않습니다.

> **Warning**
>
> 이 명령을 실행한 후 시스템을 수동으로 재부팅하지 마세요. Citrix PV 드라이버는 DSRM을 지원하지 않으므로 시스템에 접속할 수 없습니다.

3. 다음 명령을 실행하여 **DisableDCCheck**를 레지스트리에 추가합니다.

```
reg add HKLM\SOFTWARE\Wow6432Node\Amazon\AWSPVDriverSetup /v DisableDCCheck /t
 REG_SZ /d true
```

4. 다운로드 인스턴스에 최신 드라이버 패키지를 하고 zip 아카이브를 추출합니다.
5. `AWSPVDriverSetup.msi`을(를) 실행합니다.

   MSI를 실행하면 인스턴스가 자동으로 재부팅되고 드라이버를 업그레이드합니다. 최대 15분 동안 인스턴스를 사용할 수 없습니다.
6. 업그레이드를 완료하고 인스턴스가 Amazon EC2 콘솔에서 두 상태 확인을 모두 통과하면 원격 데스크톱을 사용하여 인스턴스에 연결합니다. 개방형 디스크 관리에서 모든 오프라인 보조 볼륨을 검토하고 이전에 기록한 드라이버 문자와 폴더 매핑에 따라 이를 온라인으로 전환합니다.

   사용자 이름을 hostname\administrator 형식으로 지정하여 인스턴스와 연결해야 합니다. 형식(예: Win2k12TestBox\administrator)으로 지정하여 인스턴스에 연결해야 합니다.
7. 다음 명령을 실행하여 DSRM 부팅 구성을 제거합니다.

```
bcdedit /deletevalue safeboot
```

8. 인스턴스를 재부팅합니다.
9. 업그레이드 프로세스를 완료하려면 새 드라이버가 설치되었는지 확인합니다. 디바이스 관리자(Device Manager)의 스토리지 컨트롤러(Storage Controllers) 아래에서 AWS PV Storage Host Adapter를 찾습니다. 드라이버 버전이 드라이버 버전 기록 표에 나열된 최신 버전과 동일한지 확인합니다. 자세한 내용은 AWS PV 드라이버 패키지 기록 섹션을 참조하세요.
10. 다음 명령을 실행하여 레지스트리에서 **DisableDCCheck**를 삭제합니다.

```
reg delete HKLM\SOFTWARE\Wow6432Node\Amazon\AWSPVDriverSetup /v DisableDCCheck
```

> **Note**
>
> Citrix PV 드라이버용 Netsh를 사용하여 이전에 TCP 오프로드을 비활성화한 경우 AWS PV 드라이버로 업그레이드한 후 이 기능을 다시 활성화하는 것이 좋습니다. Citrix 드라이버의 TCP 오프로딩 문제가 AWS PV 드라이버에는 없습니다. 따라서 TCP 오프로딩은 AWS PV 드라이버를 사용할 때 더 우수한 성능을 제공합니다.

Windows Server 2008 및 2008 R2 인스턴스 업그레이드(Red Hat에서 Citrix PV로 업그레이드)

Red Hat 드라이버를 Citrix PV 드라이버로 업그레이드하기 전에 다음을 수행해야 합니다.

- 최신 버전의 EC2Config 서비스를 설치합니다. 자세한 내용은 최신 버전의 EC2Config 설치 섹션을 참조하세요.
- Windows PowerShell 3.0이 설치되었는지 확인합니다. 설치된 버전을 확인하려면 PowerShell 창에서 다음 명령을 실행합니다.

  ```
  PS C:\> $PSVersionTable.PSVersion
  ```

  Windows PowerShell 3.0은 Windows Management Framework(WMF) 버전 3.0 설치 패키지 번들에 포함됩니다. Windows PowerShell 3.0을 설치해야 하는 경우 Microsoft 다운로드 센터에서 Windows Management Framework 3.0을 참조하세요.
- 중요한 정보를 인스턴스에 백업하거나 인스턴스에서 AMI를 생성합니다. AMI 생성에 대한 자세한 내용은 Amazon EBS 지원 AMI 생성 섹션을 참조하세요.

  > **Tip**
  >
  > Amazon EC2 콘솔에서 AMI를 생성하는 대신 Systems Manager Automation을 사용하여 `AWS-CreateImage` 런북으로 AMI를 생성할 수 있습니다. 자세한 내용은 AWS Systems Manager 자동화 런북 레퍼런스 사용 설명서의 AWS-CreateImage 섹션을 참조하세요.

  AMI를 생성하는 경우 다음을 수행해야 합니다.

  - 암호를 입력합니다.
  - Sysprep 도구를 직접 실행하거나 EC2Config 서비스를 사용하지 마세요.
  - 이더넷 어댑터를 설정하여 DHCP를 통해 IP 주소를 자동으로 할당받습니다.

Red Hat 드라이버를 업그레이드하려면

1. 인스턴스 연결 후 로컬 관리자로 로그인합니다. 인스턴스 연결에 대한 자세한 내용은 RDP를 사용하여 Windows 인스턴스에 연결 주제를 참조하세요.
2. 인스턴스에서 Citrix PV 업그레이드 패키지를 다운로드합니다.
3. 원하는 위치에서 업그레이드 패키지 콘텐츠의 압축을 풉니다.
4. Upgrade.bat 파일을 두 번 클릭합니다. 보안 경고가 나타나면 실행(Run)을 선택합니다.
5. 드라이버 업그레이드(Upgrade Drivers) 대화 상자에서 정보를 확인한 다음 업그레이드를 시작하려면 예(Yes)를 선택합니다.
6. Red Hat Paravirtualized Xen Drivers for Windows 설치 제거 프로그램(Red Hat Paravirtualized Xen Drivers for Windows uninstaller) 대화 상자에서 예(Yes)를 선택하여 Red Hat 소프트웨어를 제거합니다. 그러면 인스턴스가 재부팅됩니다.

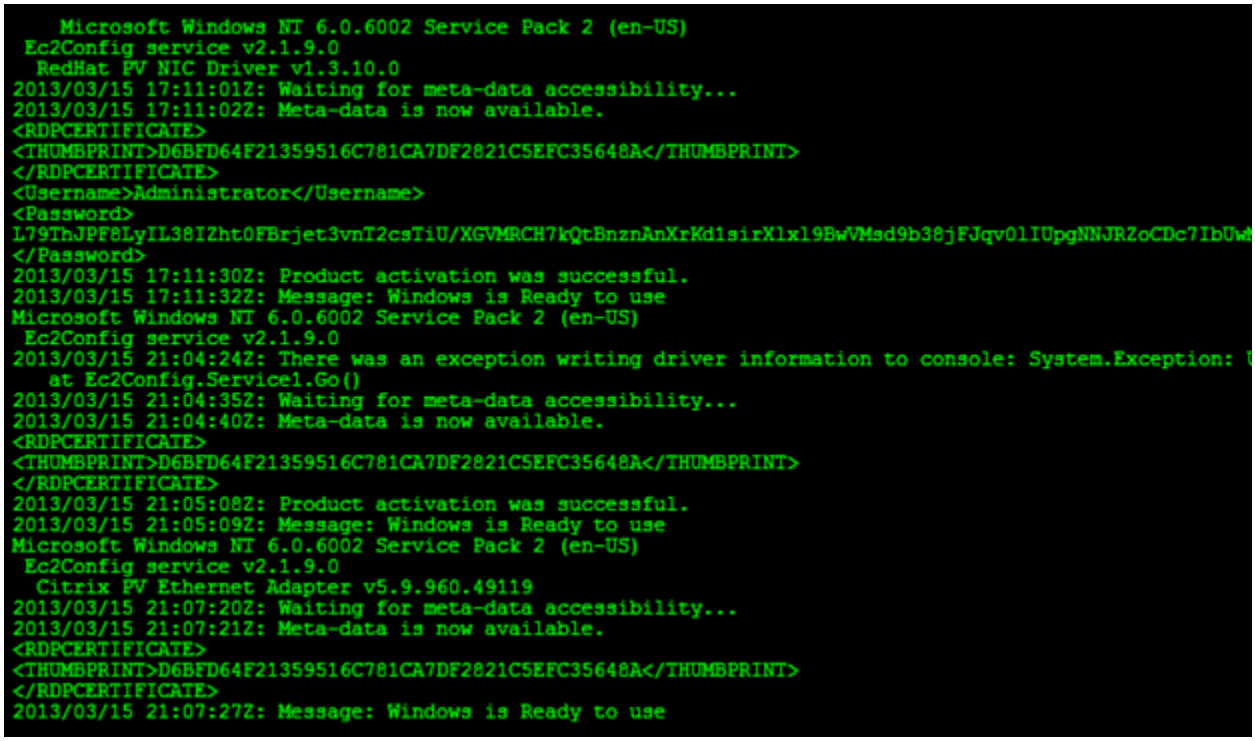
   > **Note**
   >
   > 제거 대화 상자가 나타나지 않으면 Windows 작업 표시줄에서 Red Hat Paravirtualize를 선택합니다.

7. 인스턴스가 재부팅되고 사용할 준비가 되었는지 확인합니다.

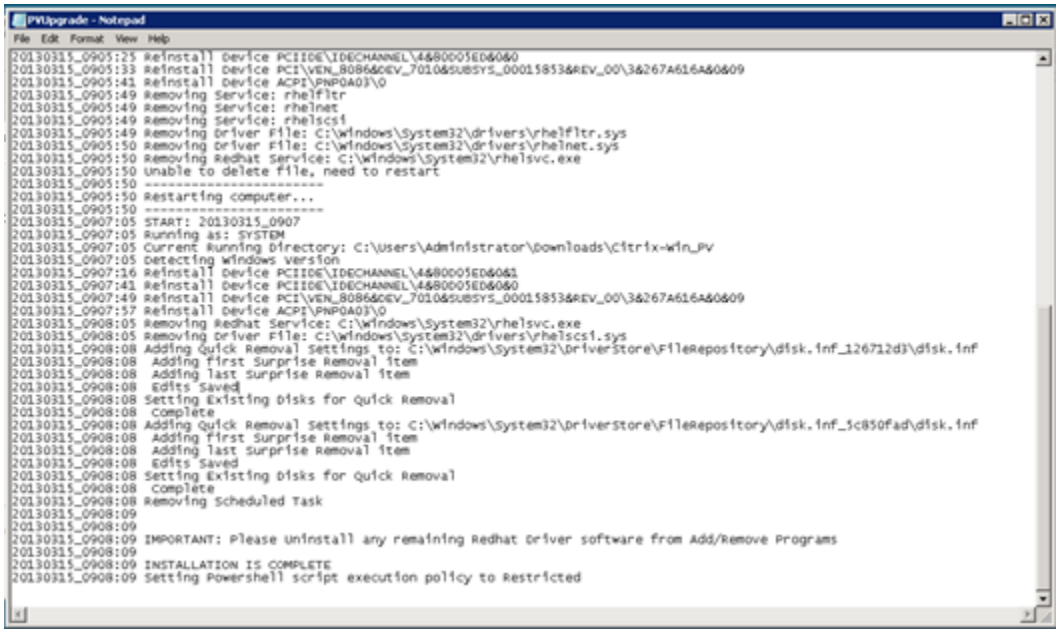
   a. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.

   b. [인스턴스(Instances)] 페이지에서 [작업(Actions)], [모니터링 및 문제 해결(Monitor and troubleshoot)] 및 [시스템 로그 가져오기(Get system log)]를 차례로 선택합니다.

   c. 그러면 업그레이드 작업이 수행되며 서버가 3회 또는 4회 재시작됩니다. 이 작업은 로그 파일에 표시되는 횟수 `Windows is Ready to use`로 확인할 수 있습니다.



8. 인스턴스 연결 후 로컬 관리자로 로그인합니다.
9. Red Hat Paravirtualized Xen Drivers for Windows 설치 제거 프로그램(Red Hat Paravirtualized Xen Drivers for Windows uninstaller) 대화 상자를 닫습니다.
10. 설치가 완료되었는지 확인합니다. 이전에 압축을 푼 `Citrix-WIN_PV` 폴더로 이동한 다음 `PVUpgrade.log` 파일을 열고 `INSTALLATION IS COMPLETE`라는 텍스트가 있는지 확인합니다.



Citrix Xen 게스트 에이전트 서비스 업그레이드

Windows Server에서 Citrix PV 드라이버를 사용하는 경우 Citrix Xen 게스트 에이전트 서비스를 업그레이드할 수 있습니다. 이 Windows 서비스는 API의 종료 및 재시작 이벤트와 같은 작업을 처리합니다. 인스턴스에서 Citrix PV 드라이버를 실행하는 경우 Windows Server의 모든 버전에서 이 업그레이드 패키지를 실행할 수 있습니다.

> **Important**
>
> Windows Server 2008 R2 이상의 경우 게스트 에이전트 업데이트가 포함된 AWS PV 드라이버로 업그레이드하는 것이 좋습니다.

드라이버 업그레이드를 시작하기 전, 중요한 정보를 인스턴스에 백업하거나 인스턴스에서 AMI를 생성합니다. AMI 생성에 대한 자세한 내용은 Amazon EBS 지원 AMI 생성 섹션을 참조하세요.

> **Tip**
>
> Amazon EC2 콘솔에서 AMI를 생성하는 대신 Systems Manager Automation을 사용하여 `AWS-CreateImage` 런북으로 AMI를 생성할 수 있습니다. 자세한 내용은 AWS Systems Manager 자동화 런북 레퍼런스 사용 설명서의 AWS-CreateImage 섹션을 참조하세요.

AMI를 생성하는 경우 다음을 수행해야 합니다.

- EC2Config 서비스에서 Sysprep 도구를 활성화하지 마세요.
- 암호를 입력합니다.
- 이더넷 어댑터를 DHCP로 설정합니다.

Citrix Xen 게스트 에이전트 서비스를 업그레이드하려면

1. 인스턴스 연결 후 로컬 관리자로 로그인합니다. 인스턴스 연결에 대한 자세한 내용은 RDP를 사용하여 Windows 인스턴스에 연결 주제를 참조하세요.
2. 인스턴스에서 Citrix 업그레이드 패키지를 다운로드합니다.
3. 원하는 위치에서 업그레이드 패키지 콘텐츠의 압축을 풉니다.
4. Upgrade.bat 파일을 두 번 클릭합니다. 보안 경고가 나타나면 실행(Run)을 선택합니다.

5. 드라이버 업그레이드(Upgrade Drivers) 대화 상자에서 정보를 확인한 다음 업그레이드를 시작하려면 예(Yes)를 선택합니다.
6. 업그레이드가 완료되면 PVUpgrade.log라는 텍스트가 있는 UPGRADE IS COMPLETE 파일이 열립니다.
7. 인스턴스를 재부팅합니다.

Windows 인스턴스에서 PV 드라이버 문제 해결

다음은 기존 Amazon EC2 이미지 및 PV 드라이버에 발생할 수 있는 문제에 대한 해결 방법입니다.

목차

- 인스턴스를 재부팅한 후 Windows Server 2012 R2에서 네트워크와 스토리지 연결이 끊김
- TCP 오프로드
- 시간 동기화
- 20,000 디스크 IOPS 이상을 활용하는 워크로드에서 CPU 병목 현상으로 인해 성능 저하 발생

인스턴스를 재부팅한 후 Windows Server 2012 R2에서 네트워크와 스토리지 연결이 끊김

> **Important**
>
> 이 문제는 2014년 9월 이전에 제공된 AMI에만 발생합니다

2014년 9월 10일 이전에 제공된 Windows Server 2012 R2 Amazon Machine Image(AMI)에서는 인스턴스 재부팅 후 네트워크와 스토리지 연결이 끊길 수 있습니다. AWS Management Console 시스템 로그의 오류는 "콘솔 출력에 대한 PV 드라이버 세부 정보를 감지하는 중 장애 발생"으로 표시됩니다. 플러그-앤-플레이 클린업 기능으로 인해 연결이 끊깁니다. 이 기능은 30일마다 비활성 시스템 디바이스를 검색하고 비활성화합니다. 기능이 EC2 네트워크 디바이스를 비활성 상태로 잘못 식별하고 시스템에서 제거합니다. 이러한 경우 인스턴스가 재부팅 후 네트워크 연결이 끊깁니다.

주의 대상 시스템은 이 문제에 의해 영향을 받을 수 있으며, 현재 위치 드라이버 업그레이드를 다운로드하고 실행할 수 있습니다. 현재 위치 드라이버 업그레이드를 수행할 수 없는 경우에는 헬퍼 스크립트를 실행할 수 있습니다. 스크립트가 인스턴스가 영향을 받는지 여부를 판별합니다. 영향을 받고 있고 Amazon EC2 네트워크 디바이스가 제거되지 않은 경우 스크립트가 플러그-앤-플레이 클린업 스캔을 비활성화합니다. 네트워크 디바이스가 제거되지 않은 경우 스크립트는 디바이스를 복구하고, 플러그-

앤-플레이 클린업 스캔을 비활성화하며, 네트워크 연결이 활성화된 상태로 인스턴스가 재부팅되도록 허용합니다.

목차

- 문제 해결 방법 선택
- 방법 1 - 향상된 네트워킹 기능
- 방법 2 - 레지스트리 구성
- 수정 스크립트 실행

문제 해결 방법 선택

문제가 발생한 인스턴스에 대한 네트워크 및 스토리지 연결을 복구하는 방법은 두 가지가 있습니다. 다음 방법 중 한 가지를 선택하세요.

방법	사전 조건	절차 개요
방법 1 - 향상된 네트워킹 기능	향상된 네트워킹 기능은 C3 인스턴스 유형을 사용하는 Virtual Private Cloud(VPC)에서만 가능합니다. 서버에서 현재 사용하는 유형이 C3 인스턴스가 아닐 때는 잠시 C3 인스턴스로 변경해야 합니다.	서버 인스턴스 유형을 C3 인스턴스로 변경합니다. 그러면 향상된 네트워킹 기능을 통해 문제가 발생한 인스턴스에 연결하여 문제를 해결할 수 있습니다. 문제를 해결한 후에는 원래 인스턴스 유형으로 다시 바꿀 수 있습니다. 일반적으로 이 방법은 방법 2보다 빠를 뿐만 아니라 사용자 오류의 가능성이 낮습니다. C3 인스턴스가 실행 중일 때는 변경 사항이 추가로 발생하기 마련입니다.
방법 2 - 레지스트리 구성	보조 서버 생성 및 액세스 기능. 레지스트리 설정 변경 기능.	문제가 발생한 인스턴스에서 루트 볼륨을 분리하여 다른 인스턴스에 연결한 다음 레지스트리 설정을 변경합니다. 보조 서버가 실행 중일 때는 변경 사항이 추가로 발생하기 마련입니다. 이 방법은 방법 1보다 느

방법	사전 조건	절차 개요
		리지만 방법 1로 문제를 해결하지 못하는 상황에서 효과가 있었습니다.

방법 1 - 향상된 네트워킹 기능

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 문제가 발생한 인스턴스를 찾습니다. 인스턴스를 선택하고 [인스턴스 상태(Instance state)]를 선택한 다음 [인스턴스 중지(Stop instance)]를 선택합니다.

   > **Warning**
   >
   > 인스턴스를 중지하면 인스턴스 스토어 볼륨의 데이터가 삭제됩니다. 인스턴스 스토어 볼륨의 데이터를 유지하려면 영구 스토리지에 백업하세요.

4. 인스턴스가 중지되면 백업을 생성합니다. 인스턴스를 선택하고 [작업(Actions)], [이미지 및 템플릿(Image and templates)] 및 [이미지 생성(Create image)]을 차례로 선택합니다.
5. 인스턴스 유형을 C3 인스턴스로 변경합니다.
6. 인스턴스를 시작합니다.
7. 원격 데스크톱을 사용하여 인스턴스에 연결한 다음 AWS PV 드라이버 업그레이드 패키지를 인스턴스에 다운로드합니다.
8. 폴더의 내용 압축을 풀고 `AWSPVDriverSetup.msi`를 실행합니다.

   MSI를 실행하면 인스턴스가 자동으로 재부팅되고 드라이버를 업그레이드합니다. 최대 15분 동안 인스턴스를 사용할 수 없습니다.
9. 업그레이드를 완료하고 인스턴스가 Amazon EC2 콘솔에서 두 상태 확인을 모두 통과하면 원격 데스크톱을 사용하여 인스턴스에 연결하고 새 드라이버가 설치되었는지 확인합니다. 디바이스 관리자(Device Manager)의 스토리지 컨트롤러(Storage Controllers) 아래에서 AWS PV Storage Host Adapter를 찾습니다. 드라이버 버전이 드라이버 버전 기록 표에 나열된 최신 버전과 동일한지 확인합니다. 자세한 내용은 AWS PV 드라이버 패키지 기록 섹션을 참조하세요.
10. 인스턴스를 중단하고 원래 인스턴스 유형으로 다시 바꿉니다.
11. 원래 인스턴스를 시작하여 정상적인 사용을 재개합니다.

방법 2 - 레지스트리 구성

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 문제가 발생한 인스턴스를 찾습니다. 인스턴스를 선택하고 [인스턴스 상태(Instance state)]를 선택한 다음 [인스턴스 중지(Stop instance)]를 선택합니다.

   > **Warning**
   >
   > 인스턴스를 중지하면 인스턴스 스토어 볼륨의 데이터가 삭제됩니다. 인스턴스 스토어 볼륨의 데이터를 유지하려면 영구 스토리지에 백업하세요.

4. [인스턴스 시작(Launch instances)]을 선택하고 문제가 발생한 인스턴스와 동일한 가용 영역에 임시 Windows Server 2008 또는 Windows Server 2012 인스턴스를 생성합니다. Windows Server 2012 R2 인스턴스를 생성하지 마세요.

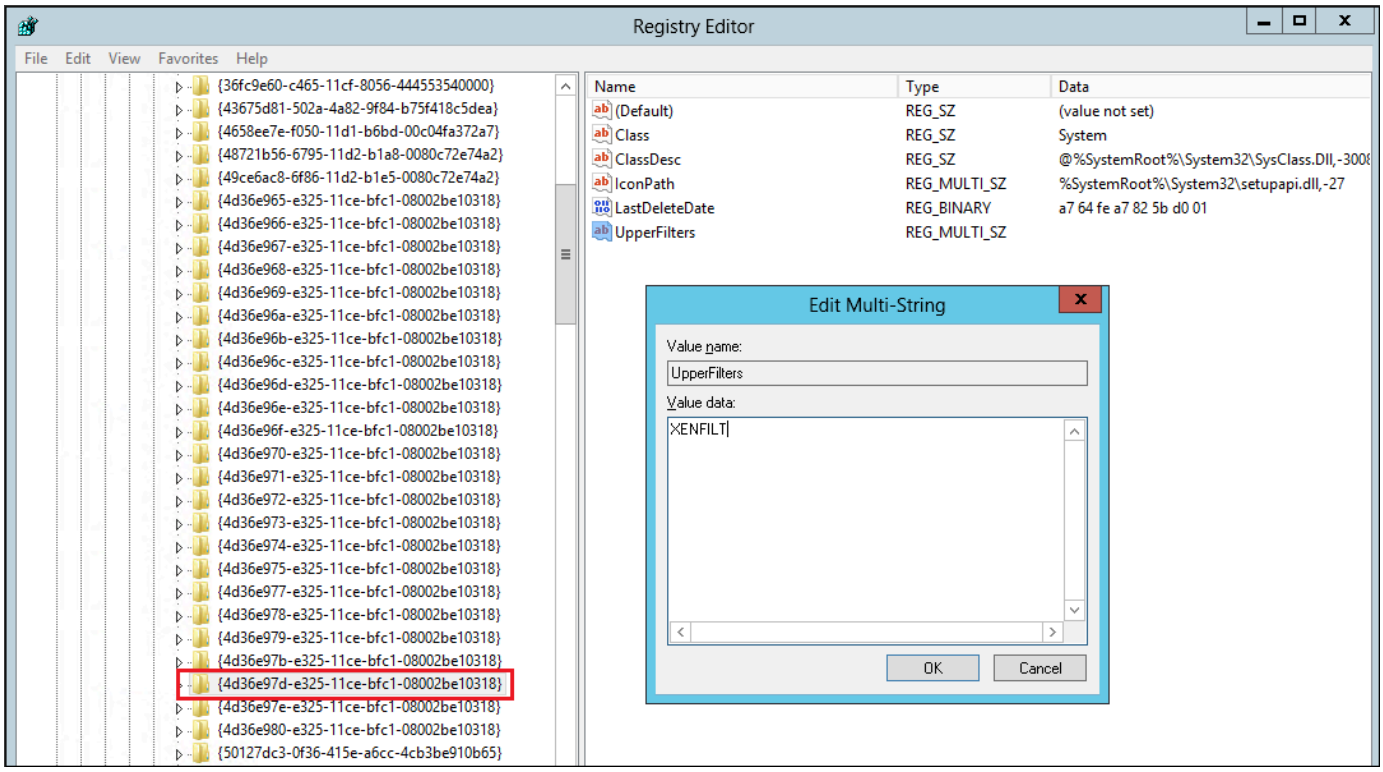
   > **Important**
   >
   > 문제가 발생한 인스턴스와 동일한 가용 영역에서 인스턴스를 생성하지 않는 경우에는 문제가 발생한 인스턴스의 루트 볼륨을 새 인스턴스에 연결할 수 없습니다.

5. 탐색 창에서 볼륨을 선택합니다.
6. 문제가 발생한 인스턴스의 루트 볼륨을 찾습니다. 볼륨을 분리하고 이전에 생성한 임시 인스턴스에 볼륨을 연결합니다. 기본 디바이스 이름(xvdf)으로 연결합니다.
7. 원격 데스크톱을 사용하여 임시 인스턴스에 연결한 후 디스크 관리 유틸리티를 사용하여 볼륨을 사용할 수 있도록 지정합니다.
8. 임시 인스턴스에서 실행 대화 상자를 열고 **`regedit`**를 입력한 다음 Enter 키를 누릅니다.
9. 레지스트리 편집기 탐색 창에서 HKEY_Local_Machine을 선택한 후 파일(File) 메뉴에서 Hive 로드(Load Hive)를 선택합니다.
10. Hive 로드(Load Hive) 대화 상자에서 Affected Volume\Windows\System32\config\System을 탐색하고 키 이름(Key Name) 대화 상자에 임시 이름을 입력합니다. 예를 들어, OldSys를 입력합니다.
11. 레지스트리 편집기의 탐색 창에서 다음 키를 찾습니다.

    HKEY_LOCAL_MACHINE***`your_temporary_key_name`**\ControlSet001\Control\Class\4d36e97d-e325-11ce-bfc1-08002be10318

HKEY_LOCAL_MACHINE***your_temporary_key_name**\ControlSet001\Control\Class\4d36e96a-e325-11ce-bfc1-08002be10318

12. 각각의 키에 대해 UpperFilters를 두 번 클릭하고 XENFILT 값을 입력한 후 확인(OK)을 선택합니다.



13. 다음 키를 찾습니다.

HKEY_LOCAL_MACHINE***your_temporary_key_name**\ControlSet001\Services\XENBUS\Parameters

14. ActiveDevice 이름 및 다음 값을 사용하여 새 문자열(REG_SZ)을 생성합니다.

PCI\VEN_5853&DEV_0001&SUBSYS_00015853&REV_01

15. 다음 키를 찾습니다.

HKEY_LOCAL_MACHINE***your_temporary_key_name**\ControlSet001\Services\XENBUS

16. 수(Count)를 0에서 1로 변경합니다.
17. 다음 키를 찾아 삭제합니다.

HKEY_LOCAL_MACHINE***your_temporary_key_name**\ControlSet001\Services\xenvbd\StartOverride

HKEY_LOCAL_MACHINE *your_temporary_key_name*\ControlSet001\Services\xenfilt \StartOverride

18. 레지스트리 편집기 탐색 창에서 처음 레지스트리 편집기를 열 때 생성한 임시 키를 선택합니다.
19. 파일(File) 메뉴에서 Hive 언로드(Unload Hive)를 선택합니다.
20. 디스크 관리(Disk Management) 유틸리티에서 이전에 연결한 드라이브를 선택하고 컨텍스트(오른쪽 클릭) 메뉴를 열고 오프라인(Offline)을 선택합니다.
21. Amazon EC2 콘솔에서 임시 인스턴스로부터 문제가 발생한 볼륨을 분리하고 디바이스 이름 /dev/sda1을 사용하여 Windows Server 2012 R2 인스턴스에 다시 연결합니다. 볼륨을 루트 볼륨으로 지정하려면 이 디바이스 이름을 지정해야 합니다.
22. 인스턴스를 시작합니다.
23. 원격 데스크톱을 사용하여 인스턴스에 연결한 다음 AWS PV 드라이버 업그레이드 패키지를 인스턴스에 다운로드합니다.
24. 폴더의 내용 압축을 풀고 `AWSPVDriverSetup.msi`를 실행합니다.

    MSI를 실행하면 인스턴스가 자동으로 재부팅되고 드라이버를 업그레이드합니다. 최대 15분 동안 인스턴스를 사용할 수 없습니다.
25. 업그레이드를 완료하고 인스턴스가 Amazon EC2 콘솔에서 두 상태 확인을 모두 통과하면 원격 데스크톱을 사용하여 인스턴스에 연결하고 새 드라이버가 설치되었는지 확인합니다. 디바이스 관리자(Device Manager)의 스토리지 컨트롤러(Storage Controllers) 아래에서 AWS PV Storage Host Adapter를 찾습니다. 드라이버 버전이 드라이버 버전 기록 표에 나열된 최신 버전과 동일한지 확인합니다. 자세한 내용은 AWS PV 드라이버 패키지 기록 섹션을 참조하세요.
26. 앞에서 임시로 생성한 인스턴스는 이 절차에서 삭제하거나 중단합니다.

수정 스크립트 실행

현재 위치 드라이버 업그레이드를 수행하거나 새로운 인스턴스를 마이그레이션할 수 없는 경우 수정 스크립트를 실행하여 플러그-앤-플레이 클린업 작업에서 발생한 문제를 수정할 수 있습니다.

수정 스크립트를 실행하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 수정 스크립트를 실행할 인스턴스를 선택합니다. [인스턴스 상태(Instance state)]를 선택한 다음 [인스턴스 중지(Stop instance)]를 선택합니다.

> **Warning**
>
> 인스턴스를 중지하면 인스턴스 스토어 볼륨의 데이터가 삭제됩니다. 인스턴스 스토어 볼륨의 데이터를 유지하려면 영구 스토리지에 백업하세요.

4. 인스턴스가 중지되면 백업을 생성합니다. 인스턴스를 선택하고 [작업(Actions)], [이미지 및 템플릿(Image and templates)] 및 [이미지 생성(Create image)]을 차례로 선택합니다.
5. [인스턴스 상태(Instance state)]를 선택한 다음 [인스턴스 시작(Start instance)]을 선택합니다.
6. 원격 데스크톱을 사용하여 인스턴스에 연결하고 RemediateDriverIssue.zip 폴더를 인스턴스에 다운로드합니다.
7. 폴더 내용을 추출합니다.
8. Readme.txt 파일의 지침에 따라 수정 스크립트를 실행합니다. 파일은 RemediateDriverIssue.zip을 추출한 폴더에 있습니다.

TCP 오프로드

> **Important**
>
> AWS PV 또는 Intel 네트워크 드라이버를 실행하는 인스턴스에는 이 문제가 적용되지 않습니다.

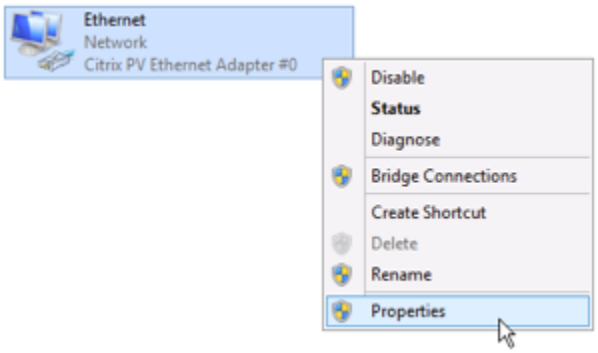
기본적으로, Windows AMI의 Citrix PV 드라이버에서는 TCP 오프로드가 활성화됩니다. 예를 들어, 특정 SQL 워크로드 등 전송 수준 오류 또는 패킷 전송 오류(Windows 성능 모니터에서 확인 가능)가 발생한 경우 이 기능을 비활성화해야 합니다.

> **Warning**
>
> TCP 오프로드를 비활성화하면 인스턴스의 네트워크 성능이 감소합니다.

Windows Server 2012 및 2008에서 TCP 오프로드를 비활성화하려면,

1. 인스턴스 연결 후 로컬 관리자로 로그인합니다.
2. Windows Server 2012의 경우 Ctrl+Esc를 눌러 시작 화면에 액세스한 다음 제어판을 선택합니다. Windows Server 2008의 경우 시작 화면에 액세스한 다음 제어판을 선택합니다.

3. 네트워크 및 인터넷, 그 다음 네트워크 및 공유 센터를 선택합니다.
4. 어댑터 설정 변경을 선택합니다.
5. Citrix PV Ethernet Adapter #0를 마우스 오른쪽 단추로 클릭한 다음 속성을 선택합니다.



6. 로컬 영역 연결 속성 대화 상자에서 구성을 선택하여 Citrix PV Ethernet Adapter #0 속성 대화 상자를 엽니다.
7. 고급 탭에서 TCP/UDP 체크섬 값 수정 외에 각 속성을 해제합니다. 속성을 해제하려면 속성 중에서 선택하여 값에서 비활성화를 선택합니다.
8. 확인을 선택합니다.
9. 명령 프롬프트 창에서 다음 명령을 실행합니다.

```
netsh int ip set global taskoffload=disabled
netsh int tcp set global chimney=disabled
netsh int tcp set global rss=disabled
netsh int tcp set global netdma=disabled
```

10. 인스턴스를 재부팅합니다.

시간 동기화

2013.02.13 Windows AMI 릴리스 이전에는 Citrix Xen 게스트 에이전트에서 시스템 시간이 올바르지 않게 설정될 수 있었습니다. 이로 인해 DHCP 임대 만료가 발생했습니다. 인스턴스 연결에 문제가 있으면 이 에이전트를 업데이트하세요.

Citrix Xen 게스트 에이전트의 업데이트 여부를 확인하려면 `C:\Program Files\Citrix\XenGuestAgent.exe` 파일의 날짜가 2013년 3월인지 확인합니다. 이 파일의 날짜가 그 이전인 경우 Citrix Xen 게스트 에이전트 서비스를 업데이트합니다. 자세한 내용은 Citrix Xen 게스트 에이전트 서비스 업그레이드 섹션을 참조하세요.

20,000 디스크 IOPS 이상을 활용하는 워크로드에서 CPU 병목 현상으로 인해 성능 저하 발생

20,000 IOPS 이상을 활용하는 AWS PV 드라이버를 실행하는 Windows 인스턴스를 사용하고 있고 버그 확인 코드 `0x9E: USER_MODE_HEALTH_MONITOR`이(가) 발생하는 경우 이 문제의 영향을 받을 수 있습니다.

AWS PV 드라이버의 디스크 읽기 및 쓰기(IO)는 IO 준비 및 IO 완료의 두 단계로 발생합니다. 기본적으로 준비 단계는 단일한 임의 코어에서 실행됩니다. 완료 단계는 코어 0에서 실행됩니다. IO를 처리하는 데 필요한 계산량은 크기 및 기타 속성에 따라 다릅니다. 일부 IO는 준비 단계에서 더 많은 계산을 사용하고 다른 IO는 완료 단계에서 더 많은 계산을 사용합니다. 인스턴스가 20,000 IOPS를 초과할 경우 준비 또는 완료 단계에 따라 병목 현상이 발생할 수 있습니다. 이 경우 인스턴스가 실행되는 CPU가 100% 용량이 됩니다. 준비 단계나 완료 단계가 병목 현상이 되는지 여부는 애플리케이션에서 사용하는 IO의 속성에 따라 다릅니다.

AWS PV 드라이버 8.4.0부터 준비 단계 및 완료 단계의 부하를 여러 코어에 분산하여 병목 현상을 제거할 수 있습니다. 각 애플리케이션은 서로 다른 IO 속성을 사용합니다. 따라서 다음 구성 중 하나를 적용하면 애플리케이션의 성능이 향상 또는 낮아지거나 성능에 영향을 주지 않을 수 있습니다. 이러한 구성을 적용한 후 애플리케이션을 모니터링하여 원하는 성능을 충족하는지 확인합니다.

1. 사전 조건

   이 문제 해결 절차를 시작하기 전에 다음과 같은 사전 요구 사항을 확인하세요.

   - 인스턴스에서 AWS PV 드라이버 버전 8.4.0 이상을 사용합니다. 업그레이드하려면 EC2 Windows 인스턴스의 PV 드라이버 업그레이드을 참조하세요.
   - 인스턴스에 대한 RDP 액세스 권한이 있습니다. RDP를 사용하여 Windows 인스턴스에 연결하는 단계에 대해서는 RDP 클라이언트를 사용하여 Windows 인스턴스에 연결 섹션을 참조하세요.
   - 인스턴스에 대한 관리자 권한이 있습니다.

2. 인스턴스에 대한 CPU 부하 관찰

   Windows 작업 관리자를 사용하여 각 CPU의 부하를 보고 디스크 IO에 잠재적인 병목 현상을 확인할 수 있습니다.

   1. 애플리케이션이 실행 중이고 프로덕션 워크로드와 유사한 트래픽을 처리하고 있는지 확인합니다.
   2. RDP를 사용하여 인스턴스에 연결합니다.
   3. 인스턴스에서 [시작] 메뉴를 선택합니다.

4. [시작] 메뉴에 Task Manager를 입력하여 작업 관리자를 엽니다.
5. 작업 관리자가 요약 보기를 표시하는 경우 [자세히]를 클릭하여 세부 보기를 확장합니다.
6. 성능 탭을 선택합니다.
7. 왼쪽 창에서 [CPU]를 선택합니다.
8. 기본 창에서 그래프를 마우스 오른쪽 버튼으로 클릭하고[그래프 변경]>[논리 프로세서]를 클릭하여 각 개별 코어를 표시합니다.
9. 인스턴스에 있는 코어 수에 따라, 시간 경과에 따른 CPU 부하를 표시하는 줄이 나타나거나 숫자만 나타날 수 있습니다.
   - 시간 경과에 따른 부하를 표시하는 그래프가 나타나면 상자가 거의 완전히 음영 처리된 CPU를 찾습니다.
   - 각 코어에 숫자가 나타나면 95% 이상을 일관되게 표시하는 코어를 찾습니다.

   10코어 0 또는 다른 코어가 과부하를 겪고 있다는 점에 유의합니다.

3. 적용할 구성 선택

구성 이름	이 구성을 적용해야 하는 경우	참고
Default configuration	워크로드가 20,000 IOPS 미만을 구동하거나 다른 구성으로 인해 성능 또는 안정성이 향상되지 않았습니다.	이 구성의 경우 IO는 몇 개의 코어에서 발생하므로 캐시 지역성을 높이고 컨텍스트 전환을 줄임으로써 더 작은 워크로드에 도움이 될 수 있습니다.
Allow driver to choose whether to distribute completion	워크로드가 20,000 IOPS 이상을 구동하고 코어 0에서 중간 또는 높은 부하가 관찰됩니다.	이 구성은 문제 발생 여부에 관계없이 PV 8.4.0 이상을 사용하고 20,000 IOPS 이상을 활용하는 모든 Xen 인스턴스에 권장됩니다.
Distribute both preparation and completion	워크로드가 20,000 IOPS 이상을 구동하고 있으며 드라이버가 배포를 선택할 수 있도록 허용해도 성능이 향상되지 않았거나0 이외의 코어에서	이 구성을 통해 IO 준비 및 IO 완료 모두를 분산시킬 수 있습니다.

구성 이름	이 구성을 적용해야 하는 경우	참고
	높은 부하가 발생하고 있습니다.	

> **Note**
>
> 준비 단계가 병렬로 실행될 때 완료 단계가 준비 단계의 과부하에 민감하기 때문에 IO 완료(DpcRedirection 설정 없이 NotifierDistributed 설정)를 분산시키지 않고 IO 준비를 분산시키는 것은 좋지 않습니다.

레지스트리 키 값

- NotifierDistributed

  0 값 또는 존재하지 않음 — 완료 단계는 코어0에서 실행됩니다.

  1 값 — 드라이버가 완료 단계나 코어 0 또는 연결된 디스크당 하나의 추가 코어를 실행하도록 선택합니다.

  2 값 — 드라이버가 연결된 디스크당 하나의 추가 코어에서 완료 단계를 실행합니다.
- DpcRedirection

  0 값 또는 존재하지 않음 — 준비 단계가 하나의 임의의 코어에서 실행됩니다.

  1 값 — 준비 단계가 여러 코어에 분산됩니다.

기본 구성

8.4.0 이전의 AWS PV 드라이버 버전에 또는 이 섹션의 다른 구성 중 하나를 적용한 후 성능 또는 안정성 저하가 관찰되는 경우 기본 구성을 적용하세요.

1. RDP를 사용하여 인스턴스에 연결합니다.
2. 관리자 권한으로 새 PowerShell 명령 프롬프트를 엽니다.

3. 다음 명령을 실행하여 NotifierDistributed 및 DpcRedirection 레지스트리 키를 제거합니다.

```
Remove-ItemProperty -Path HKLM:\System\CurrentControlSet\Services\xenvbd
\Parameters -Name NotifierDistributed
```

```
Remove-ItemProperty -Path HKLM:\System\CurrentControlSet\Services\xenvbd
\Parameters -Name DpcRedirection
```

4. 인스턴스를 재부팅합니다.

드라이버에서 완료를 배포할지 여부를 선택하도록 허용

PV 스토리지 드라이버가 IO 완료를 분산시킬지 여부를 선택할 수 있도록 NotiferDistributed 레지스트리 키를 설정합니다.

1. RDP를 사용하여 인스턴스에 연결합니다.
2. 관리자 권한으로 새 PowerShell 명령 프롬프트를 엽니다.
3. 다음 명령을 실행하여 NotiferDistributed 레지스트리 키를 설정합니다.

```
Set-ItemProperty -Type DWORD -Path HKLM:\System\CurrentControlSet\Services\xenvbd
\Parameters -Value 0x00000001 -Name NotifierDistributed
```

4. 인스턴스를 재부팅합니다.

준비와 완료를 모두 배포

항상 준비 단계와 완료 단계를 모두 분산시키도록 NotifierDistributed 및DpcRedirection레지스트리 키를 설정합니다.

1. RDP를 사용하여 인스턴스에 연결합니다.
2. 관리자 권한으로 새 PowerShell 명령 프롬프트를 엽니다.
3. 다음 명령을 실행하여 NotifierDistributed 및 DpcRedirection 레지스트리 키를 설정합니다.

```
Set-ItemProperty -Type DWORD -Path HKLM:\System\CurrentControlSet\Services\xenvbd
\Parameters -Value 0x00000002 -Name NotifierDistributed
```

```
Set-ItemProperty -Type DWORD -Path HKLM:\System\CurrentControlSet\Services\xenvbd
\Parameters -Value 0x00000001 -Name DpcRedirection
```

4. 인스턴스를 재부팅합니다.

AWS NVMe 드라이버

Amazon EBS 볼륨 및 인스턴스 스토어 볼륨은 Nitro 기반 인스턴스에서 NVMe 블록 디바이스로 표시됩니다. NVMe 블록 디바이스로 노출된 볼륨에 대해 Amazon EBS 기능의 성능과 기능을 완전히 활용하려면 인스턴스에 AWS NVMe 드라이버가 설치되어 있어야 합니다. 모든 최신 AWS Windows AMI에는 기본적으로 AWS NVMe 드라이버가 설치되어 있습니다.

EBS 및 NVMe에 대한 자세한 내용은 Amazon EBS 사용 설명서의 Amazon EBS and NVMe를 참조하세요. SSD 인스턴스 스토어 및 NVMe에 대한 자세한 내용은 EC2 인스턴스용 SSD 인스턴스 스토어 볼륨 섹션을 참조하세요.

Linux 인스턴스

다음 AMI는 필수 NVMe 드라이버를 포함합니다.

- Amazon Linux 2
- Amazon Linux AMI 2018.03
- Ubuntu 14.04 이상(`linux-aws` 커널 포함)

> **Note**
>
> Ubuntu 18.04 이상(`linux-aws` 커널 포함)이 필요한 AWS Graviton 기반 인스턴스 유형

- 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 NVMe 드라이버가 있는지 확인하려면

다음 명령을 사용하여 인스턴스에 NVMe 드라이버가 있는지 확인할 수 있습니다.

- Amazon Linux, RHEL, CentOS 및 SUSE Linux Enterprise Server

```
$ modinfo nvme
```

  인스턴스에 NVMe 드라이버가 있는 경우 명령은 드라이버에 대한 정보를 반환합니다.

- Amazon Linux 2 및 Ubuntu

```
$ ls /sys/module/ | grep nvme
```

  인스턴스에 NVMe 드라이버가 있는 경우 명령은 설치된 드라이버를 반환합니다.

NVMe 드라이버를 업데이트하려면

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.

Windows 인스턴스

PowerShell

인스턴스를 Amazon이 제공하는 최신 AWS Windows AMI 중 하나에서 시작하지 않은 경우 다음 절차에 따라 인스턴스에 현재 AWS NVMe 드라이버를 설치합니다. 이 설치에는 재부팅이 필요합니다. 설치 스크립트에 따라 인스턴스가 재부팅되거나 최종 단계로 인스턴스를 재부팅해야 합니다.

사전 조건

- PowerShell 버전 3.0 이상이 설치되어 있어야 합니다.
- 이 섹션에 표시된 명령은 64비트 버전의 PowerShell에서 실행되어야 합니다. x86 버전의 PowerShell을 사용하지 마세요. 이는 32비트 버전 쉘이며 이러한 명령에는 지원되지 않습니다.

최신 AWS NVMe 드라이버를 다운로드하고 설치하려면

1. 변경 사항을 롤백해야 하는 경우를 대비하여 다음과 같이 AMI를 백업으로 생성하는 것이 좋습니다.

   a. 인스턴스를 중지하면 인스턴스 스토어 볼륨의 데이터가 삭제됩니다. 인스턴스를 중지하기 전에 필요한 데이터를 인스턴스 스토어 볼륨에서 영구 스토리지(예: Amazon EBS 또는 Amazon S3)로 복사했는지 확인합니다.
   b. 탐색 창에서 인스턴스를 선택합니다.
   c. 드라이버 업그레이드가 필요한 인스턴스를 선택하고 [인스턴스 상태(Instance state)], [인스턴스 중지(Stop instances)]를 선택합니다.
   d. 인스턴스가 중지되면 [작업(Actions)], [이미지 및 템플릿(Image and templates)] 및 [이미지 생성(Create image)]을 차례로 선택합니다.
   e. 인스턴스 상태, 인스턴스 시작을 차례로 선택합니다.
2. 인스턴스 연결 후 로컬 관리자로 로그인합니다.

3. 다음 옵션 중 하나를 사용하여 드라이버를 인스턴스에 다운로드합니다.
   - 브라우저 - 다운로드 인스턴스에 최신 드라이버 패키지를 하고 zip 아카이브를 추출합니다.
   - PowerShell - 다음 명령을 실행합니다.

     ```
     Invoke-WebRequest https://s3.amazonaws.com/ec2-windows-drivers-downloads/
     NVMe/Latest/AWSNVMe.zip -outfile $env:USERPROFILE\nvme_driver.zip
     Expand-Archive $env:userprofile\nvme_driver.zip -DestinationPath
      $env:userprofile\nvme_driver
     ```

     Windows Server 2016 또는 이전 버전을 사용 중이고 파일을 다운로드할 때 오류가 발생하는 경우 PowerShell 터미널에서 TLS 1.2를 활성화해야 할 수 있습니다. 다음 명령을 사용하여 현재 PowerShell 세션에 대해 TLS 1.2를 활성화한 다음 다시 시도해보세요.

     ```
     [Net.ServicePointManager]::SecurityProtocol =
      [Net.SecurityProtocolType]::Tls12
     ```

4. nvme_driver 디렉터리(.\install.ps1)에서 install.ps1 PowerShell 스크립트를 실행하여 인스턴스에 드라이버를 설치합니다. 오류가 발생하면 PowerShell 3.0 이상을 사용하고 있는지 확인합니다.

   a. (선택 사항) AWS NVMe 버전 1.5.0부터 Windows Server 2016 이상에서 SCSI(Small Computer System Interface) 영구 예약이 지원됩니다. 이 기능은 공유 Amazon EBS 스토리지를 통한 Windows Server 장애 조치 클러스터링에 대한 지원을 추가합니다. 기본적으로 이 기능은 설치 중에 활성화되지 않습니다.

   EnableSCSIPersistentReservations 파라미터를 값 $true(으)로 지정하여 드라이버를 설치하기 위해 install.ps1 스크립트를 실행할 때 이 기능을 활성화할 수 있습니다.

   ```
   PS C:\> .\install.ps1 -EnableSCSIPersistentReservations $true
   ```

   EnableSCSIPersistentReservations 파라미터를 값 $false(으)로 지정하여 드라이버를 설치하기 위해 install.ps1 스크립트를 실행할 때 이 기능을 비활성화할 수 있습니다.

   ```
   PS C:\> .\install.ps1 -EnableSCSIPersistentReservations $false
   ```

b. AWS NVMe `1.5.0`부터 `install.ps1` 스크립트는 항상 드라이버와 함께 `ebsnvme-id` 도구를 설치합니다.

(선택 사항) 버전 `1.4.0`, `1.4.1` 및 `1.4.2`의 경우 `install.ps1` 스크립트를 사용하여 `ebsnvme-id` 도구를 드라이버와 함께 설치할지 여부를 지정할 수 있습니다.

i. `ebsnvme-id` 도구를 설치하려면 `InstallEBSNVMeIdTool 'Yes'`를 지정합니다.

ii. 도구를 설치하지 않으려는 경우 `InstallEBSNVMeIdTool 'No'`을(를) 지정하세요.

`InstallEBSNVMeIdTool`을 지정하지 않으면 `C:\ProgramData\Amazon\Tools`에 도구가 이미 있는 경우 패키지는 기본적으로 도구를 업그레이드합니다. 도구가 없는 경우 `install.ps1`은 기본적으로 도구를 업그레이드하지 않습니다.

도구를 패키지의 일부로 설치하지 않고 나중에 설치하려는 경우, 드라이버 패키지에서 최신 버전이나 도구를 찾을 수 있습니다. 또는 Amazon S3에서 버전 `1.0.0`을 다운로드할 수 있습니다.

`ebsnvme-id` 도구를 다운로드하세요.

5. 설치 관리자에서 인스턴스가 재부팅되지 않는 경우 인스턴스를 재부팅합니다.

Distributor

AWS Systems Manager의 기능인 Distributor를 사용하여 NVMe 드라이버 패키지를 일회성으로 설치하거나 예약된 업데이트와 함께 설치할 수 있습니다.

최신 AWS NVMe 드라이버를 설치하려면

1. Distributor를 사용하여 NVMe 드라이버 패키지를 설치하는 방법에 대한 지침은 Amazon EC2 Systems Manager 사용 설명서의 패키지 설치 또는 업데이트 절차를 참조하세요.
2. 설치 유형에서 제거 및 다시 설치를 선택합니다.
3. 이름에서 AWSNVMe를 선택합니다.
4. (선택 사항) 추가 인수에서 값을 지정하여 설치를 사용자 지정할 수 있습니다. 유효한 JSON 구문을 사용하여 값의 형식을 지정해야 합니다. `aws configure` 패키지에 대한 추가 인수를 전달하는 방법의 예는 Command 문서 플러그인 참조를 참조하세요.

a. AWS NVMe `1.5.0`부터 드라이버는 Windows Server 2016 이상의 SCSI 영구 예약을 지원합니다. 기본적으로 이 기능은 설치 중에 활성화되지 않습니다.

- 이 기능을 활성화하려면 {"SSM_EnableSCSIPersistentReservations": "true"}를 지정하세요.
- 이 기능을 활성화하지 않으려면 {"SSM_EnableSCSIPersistentReservations": "false"}를 지정하세요.

b. AWS NVMe 1.5.0부터 install.ps1 스크립트는 항상 ebsnvme-id 도구를 설치합니다.

(선택 사항) 버전 1.4.0, 1.4.1 및 1.4.2의 경우 install.ps1 스크립트를 사용하여 ebsnvme-id 도구를 드라이버와 함께 설치할지 여부를 지정할 수 있습니다.

- ebsnvme-id 도구를 설치하려면 {"SSM_InstallEBSNVMeIdTool": "Yes"}를 지정하세요.
- 도구를 설치하지 않으려는 경우 {"SSM_InstallEBSNVMeIdTool": "No"}을(를) 지정하세요.

추가 인수(Additional Arguments)에 대해 SSM_InstallEBSNVMeIdTool이 지정되지 않았으며 C:\ProgramData\Amazon\Tools에 도구가 이미 있는 경우 패키지는 기본적으로 도구를 업그레이드합니다. 도구가 없는 경우 패키지는 기본적으로 도구를 업그레이드하지 않습니다.

도구를 패키지의 일부로 설치하지 않고 나중에 설치하려는 경우, 드라이버 패키지에서 최신 버전이나 도구를 찾을 수 있습니다. 또는 Amazon S3에서 버전 1.0.0을 다운로드할 수 있습니다.

ebsnvme-id 도구를 다운로드하세요.

5. 설치 관리자에서 인스턴스가 재부팅되지 않는 경우 인스턴스를 재부팅합니다.

Windows 인스턴스용 SCSI 영구 예약 구성

AWS NVMe 드라이버 버전 1.5.0 이상을 설치한 후에는 Windows Server 2016 이상의 Windows 레지스트리를 사용하여 SCSI 영구 예약을 활성화하거나 비활성화할 수 있습니다. 이러한 레지스트리 변경 사항을 적용하려면 인스턴스를 재부팅해야 합니다.

1의 값을 EnableSCSIPersistentReservations(으)로 설정하는 다음 명령을 통해 SCSI 영구 예약을 활성화할 수 있습니다.

```
PS C:\> $registryPath = "HKLM:\SYSTEM\CurrentControlSet\Services\AWSNVMe\Parameters
\Device"
Set-ItemProperty -Path $registryPath -Name EnableSCSIPersistentReservations -Value 1
```

`0`의 값을 `EnableSCSIPersistentReservations`(으)로 설정하는 다음 명령을 통해 SCSI 영구 예약을 비활성화할 수 있습니다.

```
PS C:\> $registryPath = "HKLM:\SYSTEM\CurrentControlSet\Services\AWSNVMe\Parameters
\Device"
Set-ItemProperty -Path $registryPath -Name EnableSCSIPersistentReservations -Value 0
```

AWS NVMe Windows 드라이버 버전 내역

다음 표에서는 Amazon EC2의 각 Windows Server 버전에서 실행되는 AWS NVMe 드라이버를 보여줍니다.

Windows Server 버전	AWS NVMe 드라이버 버전
Windows Server 2025	최신 버전
Windows Server 2022	최신 버전
Windows Server 2019	최신 버전
Windows Server 2016	최신 버전
Windows Server 2012 R2	버전 1.5.1 이하
Windows Server 2012	버전 1.5.1 이하
Windows Server 2008 R2	버전 1.3.2 이하
Windows Server 2008	버전 1.3.2 이하

다음 표에서는 AWS NVMe 드라이버의 릴리스 버전에 대해 설명합니다.

패키지 버전	드라이버 버전	세부 정보	릴리스 날짜
1.6.0	1.6.0	• PnPUtil을 사용하도록 설치 스크립트를 업데이트했습니다. • NVMe IOCTL을 사용하도록 ebsnvme-id.exe를 업데이트했습니다.	2024년 10월 25일
1.5.1	1.5.0	`ebsnvme-id` 도구에 대한 폴더가 없는 경우 생성하도록 설치 스크립트가 수정되었습니다.	2023년 11월 17일
1.5.0	1.5.0	Windows Server 2016 이상을 실행하는 인스턴스에 대한 SCSI(Small Computer System Interface) 영구 예약에 대한 지원이 추가되었습니다. 이제 기본적으로 ebsnvme-id 도구(`ebsnvme-id.exe`)가 설치됩니다.	2023년 8월 31일
1.4.2	1.4.2	D3 인스턴스에서 AWS NVMe 드라이버가 인스턴스 스토어 볼륨을 지원하지 않는 버그가 수정되었습니다.	2023년 3월 16일
1.4.1	1.4.1	이 선택적 NVMe 기능을 지원하는 EBS 볼륨에 대한 NPGW(Namespace Preferred Write Granularity)를 보고합니다. 자세한 내용은 NVMe Base Specification, version 1.4(NVMe 기본 사양, 버전 1.4)의 섹션 8.25, "Improving Performance through I/O Size and Alignment Adherence"(I/O 크기 및 정렬 준수를 통한 성능 향상)를 참조하세요.	2022년 5월 20일
1.4.0	1.4.0	• 애플리케이션이 NVMe 디바이스와 상호 작용할 수 있도록 하는 IOCTL에 대한 지원이 추가되었습니다. 이 지원을 통해 애플리케이션은 NVMe 디바이스에서 `IdentifyController` , `IdentifyNamespace` 및 `NameSpace` 목록을 가져올 수 있습니다. 자세한 내용은 Microsoft 설명서의 프로토콜별 쿼리를 참조하세요. •	2021년 11월 23일

패키지 버전	드라이버 버전	세부 정보	릴리스 날짜
		1.4.0 드라이버 버전과 최신 `ebsnvme-id` 도구(`ebsnvme-id.exe`)가 단일 패키지에 결합되어 있습니다. 이 조합을 사용하면 단일 패키지에서 드라이버와 도구를 모두 설치할 수 있습니다. 자세한 내용은 AWS NVMe 드라이버 섹션을 참조하세요. • 버그 수정 및 안정성 향상.	
1.3.2	1.3.2	IO를 처리하는 EBS 볼륨 수정과 관련된 문제가 수정되어 데이터가 손상될 수 있습니다. 온라인 EBS 볼륨을 수정(예: 크기 조정 또는 유형 변경)하지 않는 고객은 영향을 받지 않습니다. Windows Server 2008 및 2008 R2에서 실행할 수 있는 마지막 버전입니다. 이 버전은 다운로드할 수 있지만 더 이상 지원되지 않습니다. Windows Server 2008 및 2008 R2는 수명이 종료되어 더 이상 Microsoft에서 지원하지 않습니다.	2019년 9월 10일
1.3.1	1.3.1	안정성 개선.	2019년 5월 21일
1.3.0	1.3.0	디바이스 최적화 개선.	2018년 8월 31일
1.2.0	1.2.0	베어 메탈 인스턴스를 포함하여 지원되는 모든 인스턴스의 AWS NVMe 디바이스에 대한 성능과 안정성을 개선했습니다.	2018년 6월 13일
>1.0.0	>1.0.0	Windows Server를 실행하는 지원 인스턴스 유형에 대한 AWS NVMe 드라이버	2018년 2월 12일

알림 구독

새로운 EC2 Windows Driver 버전이 릴리스되면 이를 알리도록 Amazon SNS를 설정할 수 있습니다. 알림을 받으려면 다음 절차를 수행합니다.

콘솔에서 EC2 알림을 받으려면

1. https://console.aws.amazon.com/sns/v3/home에서 Amazon SNS 콘솔을 엽니다.
2. 필요한 경우 탐색 모음에서 리전을 미국 동부(버지니아 북부)로 변경합니다. 구독을 신청하는 SNS 알림이 이 지역에 있기 때문에 이 지역을 선택해야 합니다.
3. 탐색 창에서 구독을 선택합니다.
4. 구독 생성을 선택합니다.
5. 구독 생성 대화 상자에서 다음 작업을 수행합니다.
   a. TopicARN의 경우, 다음 Amazon 리소스 이름(ARN)을 복사합니다.

      arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers
   b. 프로토콜에서 `Email`을 선택합니다.
   c. 엔드포인트에서 알림을 받을 이메일 주소를 입력합니다.
   d. Create subscription을 선택합니다.
6. 확인 이메일이 발송됩니다. 이메일을 열고 지침에 따라 구독을 완료합니다.

새 EC2 Windows 드라이버가 릴리스될 때마다 구독자에게 알림이 전송됩니다. 이런 알림을 더 이상 받지 않기를 원하는 경우, 다음 절차를 수행해서 구독을 해제하세요.

Amazon EC2 Windows 드라이버 알림을 구독 해제하려면

1. https://console.aws.amazon.com/sns/v3/home에서 Amazon SNS 콘솔을 엽니다.
2. 탐색 창에서 구독을 선택합니다.
3. 구독 확인란을 선택한 후 작업, 구독 삭제를 선택합니다. 확인 메시지가 나타나면 삭제를 선택합니다.

AWS CLI를 사용하여 EC2 알림을 구독하려면

AWS CLI를 사용하여 EC2 알림을 구독하려면 다음 명령을 사용합니다.

```
aws sns subscribe --topic-arn arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers --
protocol email --notification-endpoint YourUserName@YourDomainName.ext
```

AWS Tools for Windows PowerShell을(를) 사용하여 EC2 알림을 구독하는 방법

AWS Tools for Windows PowerShell를 사용하여 EC2 알림을 구독하려면 다음 명령을 사용합니다.

```
Connect-SNSNotification -TopicArn 'arn:aws:sns:us-east-1:801119661308:ec2-windows-
drivers' -Protocol email -Region us-east-1 -Endpoint 'YourUserName@YourDomainName.ext'
```

Amazon EC2 Windows 인스턴스 구성

Windows 인스턴스를 시작한 후에는 관리자로 로그인하여 Windows 기능과 시스템 설정에 대한 추가 구성을 수행할 수 있습니다. EC2 Windows 문제 해결 유틸리티는 인스턴스 관련 문제를 해결하는 데 도움이 될 수 있습니다.

다음과 같이 Windows 시작 에이전트와 기타 Windows 관련 기능을 구성할 수 있습니다.

Windows 시작 에이전트

각 AWS Windows AMI와 AWS Marketplace에서 제공되는 다른 많은 AMI에는 기본 설정으로 사전 구성된 Windows 시작 에이전트가 포함되어 있습니다. 시작 에이전트는 인스턴스 시작 중 태스크를 수행하고, 인스턴스가 중지되었다가 나중에 시작되거나 다시 시작되면 실행됩니다.

Windows용 EC2 Fast Launch

모든 Amazon EC2 Windows 인스턴스는 여러 번의 재부팅이 포함된 표준 Windows 운영 체제(OS) 시작 단계를 거쳐야 하며, 완료하는 데 15분 이상이 걸리는 경우가 많습니다. EC2 Fast Launch 기능이 활성화된 Amazon EC2 Windows Server AMI는 이러한 단계와 재부팅 중 일부를 미리 완료하여 인스턴스를 시작하는 데 걸리는 시간을 줄입니다.

Windows 관련 시스템 설정

다음 목록에는 Windows 운영 체제에만 적용되는 일부 시스템 설정이 포함되어 있습니다.

Windows 관리자 암호 변경

Windows 인스턴스에 연결할 경우 인스턴스에 대한 액세스 권한이 있는 사용자 계정과 암호를 지정해야 합니다. 인스턴스에 처음 연결할 때 관리자 계정을 사용하고 기본 암호를 입력해야 합니다. 인스턴스에 처음 연결했을 때 관리자 암호를 기본값과 다르게 변경하는 것이 좋습니다.

Windows 시스템 구성 요소 추가

Windows Server 운영 체제에는 여러 선택 구성 요소가 포함됩니다. 각 AWS Windows Server AMI에 모든 선택 구성 요소를 포함시키는 것은 유용하지 않습니다. 대신 설치 미디어 EBS 스냅샷을 제공하는데 여기에는 Windows 인스턴스에 구성 요소를 구성 또는 설치하는 데 필요한 파일이 있습니다.

Windows에 WSL 설치

Windows Subsystem for Linux(WSL)는 Windows 인스턴스에 무료로 다운로드하여 설치할 수 있습니다. WSL을 설치하면 네이티브 Linux 명령줄 도구를 Windows 인스턴스에서 직접 실행할 수 있고 기존 Windows 데스크톱과 함께 스크립트용 Linux 도구도 사용할 수 있습니다. 단일 Windows 인스턴스에서 Linux와 Windows 간에 쉽게 전환할 수 있으므로 개발 환경에서 유용할 수도 있습니다.

Windows 인스턴스의 AWS 디바이스 드라이버

Windows 인스턴스의 AWS 디바이스 드라이버를 업데이트할 수 있습니다. 자세한 내용은 the section called “디바이스 드라이버 관리” 섹션을 참조하세요.

다음 표에는 Windows 버전별로 Nitro 기반 인스턴스에 지원되는 드라이버가 요약되어 있습니다.

버전	스토리지 드라이버	향상된 네트워킹 드라이버
Windows Server 2025	AWS NVMe 최신 버전	ENA 최신 버전
Windows Server 2022	AWS NVMe 최신 버전	ENA 최신 버전
Windows Server 2019	AWS NVMe 최신 버전	ENA 최신 버전
Windows Server 2016	AWS NVMe 최신 버전	ENA 최신 버전
Windows Server 2012 R2	AWS NVMe 버전 1.5.1	ENA 버전 2.6.0
Windows Server 2008 R2	AWS NVMe 버전 1.3.2	ENA 버전 2.2.3

다음 표에는 Windows 버전별로 Xen 기반 인스턴스에 지원되는 드라이버가 요약되어 있습니다.

버전	스토리지 드라이버	향상된 네트워킹 드라이버
Windows Server 2022	AWS PV 최신 버전	

버전	스토리지 드라이버	향상된 네트워킹 드라이버
		• ENA 최신 버전 [1] • Intel VF [2] • AWS PV 최신 버전 [3]
Windows Server 2019	AWS PV 최신 버전	• ENA 최신 버전 [1] • Intel VF [2] • AWS PV 최신 버전 [3]
Windows Server 2016	AWS PV 최신 버전	• ENA 최신 버전 [1] • Intel VF [2] • AWS PV 최신 버전 [3]
Windows Server 2012 R2	AWS PV 버전 8.4.3	• ENA 버전 2.6.0 [1] • Intel VF [2] • AWS PV 버전 8.4.3 [3]
Windows Server 2008 R2	AWS PV 버전 8.3.5	• ENA 버전 2.2.3 [1] • Intel VF [2] • AWS PV 버전 8.3.5 [3]

[1] 인스턴스 유형 G3, H1, I3, `m4.16xlarge`, P2, P3, P3dn, R4의 경우.

[2] 인스턴스 유형 C3, C4, D2, I2, M4(`m4.16xlarge` 제외), R3의 경우.

[3] 인스턴스 유형 C1, M1, M2, M3, T1, T2, X1, X1e의 경우.

Amazon EC2 Windows 인스턴스의 Windows 시작 에이전트

각 AWS Windows AMI에는 기본 설정으로 사전 구성된 Windows 시작 에이전트가 포함되어 있습니다. 시작 에이전트는 인스턴스 시작 중 태스크를 수행하고, 인스턴스가 중지되었다가 나중에 시작되거나 다시 시작되면 실행됩니다. 특정 에이전트에 대한 자세한 내용은 다음 목록의 세부 정보 페이지를 참조하세요.

AWS Windows AMI에 대한 자세한 내용은 AWS Windows AMI 참조를 참조하세요.https://docs.aws.amazon.com/ec2/latest/windows-ami-reference/windows-amis.html

- EC2Launch v2 에이전트를 사용하여 EC2 Windows 인스턴스 시작 중 태스크 수행
- EC2Launch v1 에이전트를 사용하여 EC2 Windows 인스턴스 시작 중 태스크 수행
- EC2Config 서비스를 사용하여 EC2 레거시 Windows 운영 체제 인스턴스를 시작하는 동안 태스크를 수행합니다.

내용

- Amazon EC2 시작 에이전트 비교
- EC2 Windows 시작 에이전트용 DNS 접미사 구성
- EC2 Windows 시작 에이전트 알림 구독
- Windows 인스턴스용 EC2Launch v2로 마이그레이션
- EC2Launch v2 및 EC2Config 에이전트에 대한 Windows 서비스 관리

Amazon EC2 시작 에이전트 비교

다음 표에서는 EC2Config, EC2Launch v1, EC2Launch v2 간의 주요 기능 차이점을 보여줍니다.

기능	EC2Config	EC2Launch v1	EC2Launch v2
실행 방식	Windows 서비스	PowerShell 스크립트	Windows 서비스
지원	레거시 OS 전용	Windows Server 버전: • 2016 • 2019(LTSC 및 SAC)	Windows Server 버전: • 2016 • 2019(LTSC 및 SAC) • 2022 • 2025
구성 파일	XML	JSON	JSON/YAML
관리자 사용자 이름 설정	아니요	아니요	예

기능	EC2Config	EC2Launch v1	EC2Launch v2
압축된 사용자 데이터	아니요	아니요	예
AMI에서 베이크된 로컬 사용자 데이터	아니요	아니요	예, 구성 가능
사용자 데이터의 작업 구성	아니요	아니요	예
구성 가능한 월페이퍼(wallpaper)	아니요	아니요	예
태스크 실행 순서 사용자 지정	아니요	아니요	예
구성 가능한 작업 수	15	9	20(시작 시)
Windows 이벤트 뷰어 지원	예	아니요	예
이벤트 뷰어 이벤트 유형 수	2	0	30

> **Note**
>
> EC2Config 설명서는 기록 참조용으로만 제공됩니다. 실행되는 운영 체제 버전은 Microsoft에서 더 이상 지원되지 않습니다. 최신 시작 서비스로 업그레이드하는 것이 좋습니다.

EC2 Windows 시작 에이전트용 DNS 접미사 구성

Amazon EC2 시작 에이전트를 사용하면 Windows 인스턴스가 도메인 이름 확인에 사용하는 DNS 접미사 목록을 구성할 수 있습니다. 시작 에이전트는 DNS 접미사 검색 목록에 다음 값을 추가하여 `System\CurrentControlSet\Services\Tcpip\Parameters\SearchList` 레지스트리 키의 표준 Windows 설정을 재정의합니다.

- 인스턴스의 도메인

- 인스턴스 도메인의 권한 승계에 따른 접미사
- NV 도메인
- 각 네트워크 인터페이스 카드에서 지정한 도메인

모든 시작 에이전트는 DNS 접미사 구성을 지원합니다. 자세한 내용은 특정 시작 에이전트 버전을 참조하세요.

- `setDnsSuffix` 작업 및 EC2Launch v2에서 DNS 접미사를 구성하는 방법에 대한 자세한 내용은 setDnsSuffix 를 참조하세요.
- DNS 접미사 목록 설정 및 EC2Launch v1에 대한 권한 승계를 활성화하거나 비활성화하는 방법에 대한 자세한 내용은 Windows 인스턴스에서 EC2Launch v1 에이전트 구성 섹션을 참조하세요.
- DNS 접미사 목록 설정 및 EC2Config에 대한 권한 승계를 활성화하거나 비활성화하는 방법에 대한 자세한 내용은 EC2Config 설정 파일 섹션을 참조하세요.

도메인 이름 권한 승계

도메인 이름 권한 승계는 하위 도메인의 컴퓨터가 정규화된 도메인 이름을 사용하지 않고도 상위 도메인의 리소스에 액세스할 수 있도록 하는 Active Directory 동작입니다. 기본적으로 도메인 이름 권한 승계는 도메인 이름 진행 과정에서 노드가 2개만 남을 때까지 계속됩니다.

인스턴스가 도메인에 연결된 경우 시작 에이전트는 도메인 이름에 대한 권한 승계를 수행하고 **`System\CurrentControlSet\Services\Tcpip\Parameters\SearchList`** 레지스트리 키에 유지되는 DNS 접미사 검색 목록에 결과를 추가합니다. 에이전트는 다음 레지스트리 키의 설정을 사용하여 권한 승계 동작을 결정합니다.

- **`System\CurrentControlSet\Services\Tcpip\Parameters\UseDomainNameDevolution`**
  - 설정하지 않으면 권한 승계가 비활성화됨
  - `1`로 설정하면 권한 승계가 활성화됨(기본값)
  - `0`으로 설정하면 권한 승계가 비활성화됨
- **`System\CurrentControlSet\Services\Dnscache\Parameters\DomainNameDevolutionLevel`**
  - 설정하지 않으면 수준 2 사용(기본값)
  - 3 이상으로 설정하면 값을 사용하여 수준 설정

권한 승계를 비활성화하거나 승계 설정을 더 높은 수준으로 변경하면 System\CurrentControlSet\Services\Tcpip\Parameters\SearchList 레지스트리 키에 이전에 추가된 접미사가 포함됩니다. 자동으로 제거되지는 않습니다. 목록을 수동으로 업데이트하거나, 목록을 지우고 에이전트가 새 목록을 설정하는 프로세스를 실행하도록 할 수 있습니다.

> **Note**
>
> 레지스트리에서 DNS 접미사 목록을 지우려면 다음 명령을 실행하세요.
>
> ```
> PS C:\> Invoke-CimMethod -ClassName Win32_NetworkAdapterConfiguration -
> MethodName "SetDNSSuffixSearchOrder" -Arguments @{ DNSDomainSuffixSearchOrder =
>  $null } | Out-Null
> ```

권한 승계 예제

다음 예제는 도메인 이름 권한 승계 프로세스의 진행 상황을 보여줍니다.

corp.example.com

- example.com으로 진행

locale.region.corp.example.com

1. region.corp.example.com으로 진행
2. corp.example.com으로 진행
3. example.com으로 진행

locale.region.corp.example.com(DomainNameDevolutionLevel=3 설정)

1. region.corp.example.com으로 진행
2. corp.example.com으로 진행 수준 설정으로 인해 진행이 여기서 중지됩니다.

EC2 Windows 시작 에이전트 알림 구독

Amazon SNS는 새로운 EC2 시작 에이전트 버전이 릴리스되면 알려줍니다. 알림을 받으려면 다음 절차를 수행합니다.

EC2Config 알림 구독

1. https://console.aws.amazon.com/sns/v3/home에서 Amazon SNS 콘솔을 엽니다.
2. 필요한 경우 탐색 모음에서 리전을 미국 동부(버지니아 북부)로 변경합니다. 구독하는 SNS 알림이 이 리전에 생성되었기 때문에 이 리전을 선택해야 합니다.
3. 탐색 창에서 Subscriptions를 선택합니다.
4. 구독 생성을 선택합니다.
5. 구독 생성 대화 상자에서 다음 작업을 수행합니다.
   a. 주제 ARN에 알림을 받으려는 에이전트와 일치하는 다음 Amazon 리소스 이름(ARN)을 사용합니다.
      - EC2Launch v2:

        ```
        arn:aws:sns:us-east-1:309726204594:amazon-ec2launch-v2
        ```

      - EC2Launch 또는 EC2Config:

        ```
        arn:aws:sns:us-east-1:801119661308:ec2-windows-ec2config
        ```

   b. 프로토콜에서 `Email`을 선택합니다.
   c. 엔드포인트에 알림을 받을 이메일 주소를 입력합니다.
   d. Create subscription을 선택합니다.
6. 구독을 확인하도록 요청하는 전자 메일이 전송되면 이메일을 열고 지침에 따라 구독을 완료합니다.

새 시작 에이전트 버전이 릴리스될 때마다 구독자에게 알림이 전송됩니다. 이런 알림을 더 이상 받지 않기를 원하는 경우, 다음 절차를 수행해서 구독을 해제하세요.

시작 에이전트 알림 구독 해제

1. Amazon SNS 콘솔을 엽니다.
2. 탐색 창에서 구독을 선택합니다.
3. 구독을 선택한 후 작업, 구독 삭제를 선택합니다. 확인 메시지가 나타나면 삭제를 선택합니다.

Windows 인스턴스용 EC2Launch v2로 마이그레이션

EC2Launch 마이그레이션 도구는 설치된 시작 에이전트(EC2Config 및 EC2Launch v1)를 제거하고 EC2Launch v2를 설치하여 에이전트를 업그레이드합니다. 이전 시작 서비스의 해당하는 구성이 자동으로 새 서비스로 마이그레이션됩니다. 마이그레이션 도구는 EC2Launch v1 스크립트에 연결된 예약 태스크를 검색하지 않으므로 EC2Launch v2에서 이러한 태스크를 자동으로 설정하지 않습니다. 이러한 태스크를 구성하려면 `agent-config.yml` 파일을 편집하거나 EC2Launch v2 설정 대화 상자를 사용합니다. 예를 들어 인스턴스에 `InitializeDisks.ps1`을 실행하는 예약 태스크가 있는 경우 마이그레이션 도구를 실행한 후 EC2Launch v2 설정 대화 상자에서 초기화할 볼륨을 지정해야 합니다. EC2Launch v2 설정 대화 상자를 사용하여 설정 변경에 대한 절차의 6단계를 참조하세요.

마이그레이션 도구를 다운로드하거나 SSM RunCommand 문서를 사용하여 설치할 수 있습니다.

다음 위치에서 도구를 다운로드할 수 있습니다.

- https://s3.amazonaws.com/amazon-ec2launch-v2-utils/MigrationTool/windows/amd64/latest/EC2LaunchMigrationTool.zip

> **Note**
>
> 관리자 권한으로 EC2Launch v2 마이그레이션 도구를 실행해야 합니다. 마이그레이션 도구를 실행하면 EC2Launch v2가 서비스로 설치됩니다. 즉시 실행되지 않습니다. 기본적으로 인스턴스 시작 중에 실행되며 인스턴스가 중지된 후 나중에 시작되거나 다시 시작될 때 실행됩니다.

`AWSEC2Launch-RunMigration` SSM 문서를 사용하여 SSM Run Command로 최신 EC2Launch v2 버전으로 마이그레이션합니다. 이 문서를 사용하는 데는 파라미터가 필요 없습니다. SSM Run Command 사용에 대한 자세한 내용은 AWS Systems Manager Run Command를 참조하세요.

마이그레이션 도구는 EC2Config 의 다음 구성을 EC2Launch v2에 적용합니다.

- `Ec2DynamicBootVolumeSize`가 `false`로 설정된 경우 EC2Launch v2 `boot` 스테이지 제거
- `Ec2SetPassword`가 `Enabled`로 설정된 경우 EC2Launch v2 암호 유형을 `random`으로 설정
- `Ec2SetPassword`가 `Disabled`로 설정된 경우 EC2Launch v2 암호 유형을 `donothing`으로 설정
- `SetDnsSuffixList`가 `false`로 설정된 경우 EC2Launch v2 `setDnsSuffix` 태스크 제거
- `EC2SetComputerName`이 true로 설정된 경우 `yaml` 구성에 EC2Launch v2 `setHostName` 태스크 추가

마이그레이션 도구는 EC2Launch v1의 다음 구성을 EC2Launch v2에 적용합니다.

- `ExtendBootVolumeSize`가 `false`로 설정된 경우 EC2Launch v2 `boot` 스테이지 제거
- `AdminPasswordType`가 `Random`로 설정된 경우 EC2Launch v2 암호 유형을 `random`으로 설정
- `AdminPasswordType`이 `Specify`로 설정된 경우 EC2Launch v2 암호 유형을 `static`으로 설정하고 암호 데이터를 `AdminPassword`에 지정된 암호로 설정
- `SetWallpaper`가 `false`로 설정된 경우 EC2Launch v2 `setWallpaper` 태스크 제거
- `AddDnsSuffixList`가 `false`로 설정된 경우 EC2Launch v2 `setDnsSuffix` 태스크 제거
- `SetComputerName`이 `true`로 설정된 경우 EC2Launch v2 `setHostName` 태스크 추가

EC2Launch v2 및 EC2Config 에이전트에 대한 Windows 서비스 관리

관리 권한이 있는 사용자로 인스턴스에 로그인한 경우 다른 Windows 서비스와 마찬가지로 EC2Launch v2 및 EC2Config 실행 에이전트를 관리할 수 있습니다. EC2Launch v1은 기본적으로 예약된 태스크를 통해 관리되는 PowerShell 스크립트 세트입니다. 이 섹션에서는 EC2Launch v2 및 EC2Config에 대한 서비스 관리를 다룹니다.

인스턴스에 업데이트된 설정을 적용하려면 서비스용 Microsoft Management Console(MMC) 인터페이스에서 EC2Launch v2 에이전트나 EC2Config 서비스 실행 에이전트를 중지했다가 다시 시작합니다. 마찬가지로 새 버전의 시작 에이전트를 설치할 때는 먼저 에이전트를 중지한 다음 설치가 완료되면 다시 시작해야 합니다.

> **Note**
>
> 이러한 작업을 선택하려면 관리자 권한으로 MMC 서비스 인터페이스를 열어야 합니다. 이렇게 하려면 컨텍스트 메뉴에서 관리자 권한으로 실행을 선택합니다. 또는 키보드를 사용하여 인터페이스를 열려면 다음 단계를 따르세요.
>
> 1. `Tab` 키 또는 화살표 키를 사용하여 관리 도구 메뉴에서 서비스 메뉴 항목을 선택합니다.
> 2. `Ctrl` + `Shift` + `Enter` 키보드 조합을 사용하여 관리자 권한으로 엽니다.

다음 절차에는 인스턴스에서 시작 에이전트를 중지하고 시작하는 단계가 나와 있습니다.

시작 에이전트 중지

1. 실행을 시작해서 Windows 인스턴스에 연결합니다.

2. Windows 시작 메뉴에서 관리 도구를 선택합니다.
3. 이 섹션 시작 부분의 설명에 따라 관리자 권한으로 서비스 콘솔을 엽니다.
4. 서비스 목록에서 인스턴스에서 실행 중인 에이전트(EC2Launch 또는 EC2Config)를 선택한 다음 작업 메뉴에서 중지를 선택합니다. 또는 컨텍스트 메뉴를 사용하여 에이전트를 중지합니다.

시작 에이전트 재시작

1. 실행을 시작해서 Windows 인스턴스에 연결합니다.
2. Windows 시작 메뉴에서 관리 도구를 선택합니다.
3. 이 섹션 시작 부분의 설명에 따라 관리자 권한으로 서비스 콘솔을 엽니다.
4. 서비스 목록에서 인스턴스에서 실행 중인 에이전트(EC2Launch 또는 EC2Config)를 선택한 다음 작업 메뉴에서 시작 또는 다시 시작을 선택합니다. 또는 컨텍스트 메뉴를 사용하여 에이전트를 재시작할 수 있습니다.

구성 설정을 업데이트하거나, 자체 AMI를 생성하거나, AWS Systems Manager를 사용할 필요가 없는 경우에는 시작 에이전트를 삭제하고 제거할 수 있습니다.

삭제

서비스를 삭제하면 등록 서브키도 제거됩니다.

제거

서비스를 설치 제거하면 파일, 등록 서브키, 서비스 바로가기도 제거됩니다.

시작 에이전트 삭제

1. 실행을 시작해서 Windows 인스턴스에 연결합니다.
2. Windows 명령 프롬프트 창을 시작합니다.
3. 다음 명령 중 하나를 실행하여 시작 에이전트를 삭제합니다.
   - 다음 명령을 실행하여 EC2Launch 또는 EC2Launch v2를 삭제합니다.

     ```
     sc delete ec2launch
     ```

   - EC2Config 서비스를 삭제하려면 다음 명령을 실행합니다.

```
sc delete ec2config
```

시작 에이전트 제거

1. 실행을 시작해서 Windows 인스턴스에 연결합니다.
2. Windows 시스템을 선택한 다음 Windows 시작 메뉴에서 제어판을 선택합니다.
3. 프로그램 및 기능을 선택하여 인스턴스에 설치된 프로그램 목록을 엽니다.
4. 목록에서 시작 에이전트(Amazon EC2Launch 또는 EC2ConfigService)를 선택하고 파일 메뉴에서 제거를 선택합니다. 또는 컨텍스트 메뉴를 사용할 수 있습니다.

> **Note**
>
> 버전 열에서 설치된 시작 에이전트 버전을 확인할 수 있습니다.

EC2Launch v2 에이전트를 사용하여 EC2 Windows 인스턴스 시작 중 태스크 수행

AWS Windows Server 2022 및 Windows Server 2025 AMI에서 시작되는 Amazon EC2의 지원되는 모든 인스턴스에는 기본적으로 EC2Launch v2 시작 에이전트(`EC2Launch.exe`)가 포함됩니다. 또한 EC2Launch v2를 기본 시작 에이전트로 설치한 Windows Server 2016 및 2019 AMI를 제공합니다. 이러한 AMI는 EC2Launch v1을 포함하는 Windows Server 2016 및 2019 AMI와 함께 제공됩니다. Amazon EC2 콘솔에서 AMI 페이지의 검색에 `EC2LaunchV2-Windows_Server-*` 접두사를 입력하여 기본적으로 EC2Launch v2를 포함하는 Windows AMI를 검색할 수 있습니다.

시작 에이전트 버전 기능을 비교하려면 Amazon EC2 시작 에이전트 비교 섹션을 참조하세요.

EC2Launch v2는 인스턴스 시작 시 태스크를 수행하며 인스턴스가 중지된 후 나중에 시작되거나 재시작된 경우 실행되는 서비스입니다. EC2Launch v2는 태스크를 온디맨드로 수행할 수도 있습니다. 이러한 작업 중 일부는 자동으로 활성화되고, 나머지는 수동으로 활성화해야 합니다. EC2Launch v2 서비스는 모든 EC2Config 및 EC2Launch 기능을 지원합니다.

이 서비스는 구성 파일을 사용하여 작업을 제어합니다. 그래픽 도구를 사용하거나 단일 .yml 파일(`agent-config.yml`)로 직접 편집하여 구성 파일을 업데이트할 수 있습니다. 파일 위치에 대한 자세한 내용은 EC2Launch v2 디렉터리 구조 섹션을 참조하세요.

EC2Launch v2에서는 오류 문제를 해결하고 트리거를 설정하는 데 도움이 되는 Windows 이벤트 로그를 게시합니다. 자세한 내용은 Windows 이벤트 로그 섹션을 참조하세요.

지원되는 OS 버전

EC2Launch v2 에이전트는 다음 Windows Server 운영 체제(OS) 버전을 지원합니다.

- Windows Server 2025
- Windows Server 2022
- Windows Server 2019(장기 서비스 채널 및 연 2회 채널)
- Windows Server 2016

기본적으로 실행되는 태스크

EC2Launch v2 에이전트는 초기 인스턴스 시작 중에 기본적으로 다음 태스크를 한 번만 실행합니다. 태스크는 시작 단계에서 실행되는 순서에 따라 구성되어 있습니다.

`Boot` 단계

- extendRootPartition

`PreReady` 단계

- activateWindows
- setDnsSuffix
- setAdminAccount
- setWallpaper

`PostReady` 단계

- startSsm

EC2Launch v2 개념

EC2Launch v2를 고려하는 경우 다음 개념을 이해하는 것이 좋습니다.

agent-config

`agent-config`는 EC2Launch v2의 구성 폴더에 있는 파일입니다. 부팅, 네트워크, 사전 준비 및 사후 준비 스테이지에 대한 구성이 포함되어 있습니다. 이 파일은 AMI가 처음 부팅되거나 이후에 부팅될 때 실행되어야 하는 태스크의 인스턴스 구성을 지정하는 데 사용됩니다.

기본적으로 EC2Launch v2 설치 시 표준 Amazon Windows AMI에서 사용되는 권장 구성을 포함한 `agent-config` 파일이 설치됩니다. 구성 파일을 업데이트하여 EC2Launch v2에서 지정되는 AMI의 기본 부팅 환경을 변경할 수 있습니다. 파일 위치에 대한 자세한 내용은 EC2Launch v2 디렉터리 구조 섹션을 참조하세요.

Frequency(주파수)

태스크 빈도는 부팅 컨텍스트에 따라 태스크 실행 일정을 결정합니다. 대부분의 태스크에는 허용되는 빈도가 하나만 있습니다. `executeScript` 태스크의 빈도를 지정할 수 있습니다.

EC2Launch v2 태스크 구성에 다음과 같은 빈도가 표시됩니다.

- 한 번 – AMI가 처음 부팅될 때 태스크가 한 번 실행됩니다(Sysprep이 완료됨).
- 항상 – 시작 에이전트가 실행될 때마다 태스크가 실행됩니다. 시작 에이전트가 실행되는 경우:
  - 인스턴스 시작 또는 재시작
  - EC2Launch 서비스 실행
  - `EC2Launch.exe run` 호출

단계

단계는 EC2Launch v2 에이전트가 실행하는 태스크를 논리적으로 묶은 것입니다. 일부 작업은 특정 스테이지에서만 실행할 수 있고, 일부 작업은 여러 스테이지로 실행할 수 있습니다. `agent-config.yml`을 사용할 때 단계 목록과 각 단계 내에서 실행할 태스크 목록을 지정해야 합니다.

서비스는 다음 단계로 실행합니다.

1단계: 부팅

2단계: 네트워크

3단계: PreReady

Windows가 준비됨

PreReady 단계가 완료되면 서비스에서 Amazon EC2 콘솔로 `Windows is ready` 메시지를 전송합니다.

4단계: PostReady

사용자 데이터는 PostReady 단계에서 실행됩니다. 다음과 같이 일부 스크립트 버전은 `agent-config.yml` 파일 PostReady 단계 이전에 실행되고 일부는 해당 단계 이후에 실행됩니다.

`agent-config.yml` 이전

- YAML 사용자 데이터 버전 1.1

- XML 사용자 데이터

`agent-config.yml` 이후

- YAML 사용자 데이터 버전 1.0(이전 버전과의 호환성을 위한 레거시 버전)

예제 단계 및 태스크는 예: `agent-config.yml` 섹션을 참조하세요.

사용자 데이터를 사용하는 경우 실행 에이전트가 실행할 태스크 목록을 지정해야 합니다. 단계는 암시됩니다. 예제 태스크는 예: 사용자 데이터 섹션을 참조하세요.

EC2Launch v2는 `agent-config.yml` 및 사용자 데이터에 지정하는 순서대로 태스크 목록을 실행합니다. 단계는 순차적으로 실행됩니다. 이전 단계가 완료된 후 다음 단계가 시작됩니다. 태스크도 순차적으로 실행됩니다.

Task

태스크를 호출하여 인스턴스에 대한 작업을 수행할 수 있습니다. `agent-config.yml` 파일에서 또는 사용자 데이터를 통해 태스크를 구성할 수 있습니다. EC2Launch v2에 사용 가능한 태스크 목록은 EC2Launch v2 태스크를 참조하세요. 태스크 구성 스키마와 세부 정보는 EC2Launch v2 태스크 구성 섹션을 참조하세요.

사용자 데이터

사용자 데이터는 인스턴스를 시작할 때 구성 가능한 데이터입니다. 사용자 데이터를 업데이트하여 사용자 지정 AMI 또는 퀵 스타트 AMI 구성 방식을 동적으로 변경할 수 있습니다. EC2Launch v2는 60kB의 사용자 데이터 입력 길이를 지원합니다. 사용자 데이터에는 UserData 스테이지만 포함되므로 `agent-config` 파일 이후에 실행됩니다. 인스턴스 시작 마법사를 사용하여 인스턴스를 시작할 때 사용자 데이터를 입력하거나 EC2 콘솔에서 사용자 데이터를 수정할 수 있습니다. 사용자 데이터 작업에 대한 자세한 내용은 Amazon EC2가 Windows 인스턴스의 사용자 데이터를 처리하는 방법 섹션을 참조하세요.

EC2Launch v2 태스크 개요

EC2Launch v2는 부팅될 때마다 다음 태스크를 수행할 수 있습니다.

- 인스턴스에 대한 정보를 렌더링하는 새로운/선택적으로 사용자 지정된 월페이퍼(wallpaper)를 설정합니다.
- 로컬 시스템에 생성된 관리자 계정의 속성을 설정합니다.
- 검색 접미사 목록에 DNS 접미사를 추가합니다. 아직 존재하지 않는 접미사만 목록에 추가됩니다.
- 추가 볼륨에 대해 드라이브 문자를 설정하고 사용 가능한 공간을 사용하도록 확장합니다.
- 구성의 파일을 디스크에 씁니다.

- EC2Launch v2 구성 파일 또는 user-data에서 지정된 스크립트를 실행합니다. user-data의 스크립트는 일반 텍스트로 제공하거나 압축하여 base64 형식으로 제공할 수 있습니다.
- 주어진 인수를 사용해 프로그램을 실행합니다.
- 컴퓨터 이름을 설정합니다.
- 인스턴스 정보를 Amazon EC2 콘솔에 전송합니다.
- RDP 인증서 지문을 Amazon EC2 콘솔에 전송합니다.
- 파티션 처리되지 않은 공간을 포함시키기 위한 운영 시스템 파티션을 동적으로 확장.
- 사용자 데이터를 실행합니다. 사용자 데이터 지정에 대한 자세한 내용은 EC2Launch v2 태스크 구성 섹션을 참조하세요.
- 비영구 정적 경로를 설정하여 메타데이터 서비스 및 AWS KMS 서버에 도달합니다.
- 부팅 파티션이 아닌 파티션을 mbr 또는 gpt.로 설정합니다.
- Sysprep 이후에 Systems Manager 서비스를 시작합니다.
- ENA 설정을 최적화합니다.
- 최신 Windows 버전을 위해 OpenSSH를 활성화합니다.
- 점보 프레임을 활성화합니다.
- EC2Launch v2에서 Sysprep이 실행되도록 설정합니다.
- Windows 이벤트 로그를 게시합니다.

EC2Launch v2 디렉터리 구조

EC2Launch v2는 다음 디렉터리에 설치되어야 합니다.

- 서비스 바이너리: `%ProgramFiles%\Amazon\EC2Launch`
- 서비스 데이터(설정, 로그 파일 및 상태 파일): `%ProgramData%\Amazon\EC2Launch`

> **Note**
>
> 기본적으로 Windows는 파일과 폴더를 `C:\ProgramData` 아래에 숨깁니다. EC2Launch v2 디렉터리와 파일을 보려면 Windows 탐색기에 경로를 입력하거나 숨겨진 파일과 폴더를 표시하도록 폴더 속성을 변경해야 합니다.

`%ProgramFiles%\Amazon\EC2Launch` 디렉터리에는 바이너리와 지원 라이브러리가 들어 있습니다. 여기에는 다음과 같은 하위 디렉터리가 포함됩니다.

- settings
  - EC2LaunchSettingsUI.exe - agent-config.yml 파일을 수정하기 위한 사용자 인터페이스
  - YamlDotNet.dll - 사용자 인터페이스에서 일부 작업을 지원하기 위한 DLL
- tools
  - ebsnvme-id.exe - 인스턴스에서 EBS 볼륨의 메타데이터를 검사하기 위한 도구
  - AWSAcpiSpcrReader.exe - 사용할 올바른 COM 포트를 결정하기 위한 도구
  - EC2LaunchEventMessage.dll - EC2Launch에 대한 Windows 이벤트 로깅을 지원하기 위한 DLL
- service
  - EC2LaunchService.exe — 시작 에이전트가 서비스로 실행될 때 시작되는 Windows 서비스 실행 파일.
- EC2Launch.exe - 주요 EC2Launch 실행 파일
- EC2LaunchAgentAttribution.txt - EC2Launch 내에서 사용되는 코드의 저작권 표시

%ProgramData%\Amazon\EC2Launch 디렉터리에는 다음 하위 디렉터리가 포함됩니다. 로그, 구성 및 상태를 포함하여 서비스에서 생성된 모든 데이터가 이 디렉터리에 저장됩니다.

- config - 구성

  서비스 구성 파일은 이 디렉터리에 agent-config.yml로 저장됩니다. 서비스에서 실행되는 기본 태스크를 수정, 추가 또는 제거하도록 이 파일을 업데이트할 수 있습니다. 이 디렉터리에 파일을 생성할 권한은 권한 에스컬레이션을 방지하기 위해 관리자 계정으로 제한됩니다.

- log - 인스턴스 로그

  서비스(agent.log), 콘솔(console.log), 성능(bench.log), 오류(err.log), 텔레메트리(telemetry.log) 로그가 이 디렉터리에 저장됩니다. 서비스의 후속 실행 시 로그 파일이 추가됩니다.

- state - 서비스 상태 데이터

  서비스가 실행할 작업을 결정하는 데 사용하는 상태가 여기에 저장됩니다. Sysprep 이후에 서비스가 이미 실행되었는지 여부를 나타내는 .run-once 파일이 있습니다(따라서 빈도가 한 번인 태스크는 다음 실행 시 건너뜀). 이 하위 디렉터리에는 각 태스크의 상태를 추적하는 state.json 및 previous-state.json이 포함되어 있습니다.

- sysprep - Sysprep

이 디렉터리에는 재사용할 수 있는 사용자 지정 Windows AMI를 만들 때 Sysprep에서 수행할 작업을 결정하는 데 사용되는 파일이 포함되어 있습니다.

- `wallpaper` - 월페이퍼

  이 월페이퍼 이미지는 이 디렉터리에 저장됩니다.

원격 측정

원격 측정 데이터는 AWS가 사용자의 요구 사항을 더 잘 이해하고, 문제를 진단하고, AWS 서비스의 경험을 개선할 기능을 제공하는 데 도움이 되는 추가 정보입니다.

EC2Launchv2 버전 `2.0.592` 이상은 사용량 지표 및 오류와 같은 원격 측정 데이터를 수집합니다. 이 데이터는 EC2Launch v2가 실행되는 Amazon EC2 인스턴스에서 수집됩니다. 여기에는 AWS가 소유한 모든 Windows AMI가 포함됩니다.

EC2Launch v2에서 수집되는 원격 측정 데이터의 유형은 다음과 같습니다.

- 사용량 정보 - 에이전트 명령, 설치 방법 및 예약된 실행 빈도입니다.
- 오류 및 진단 정보 - 에이전트 설치 오류 코드, 실행 오류 코드 및 오류 호출 스택입니다.

수집되는 데이터의 예:

```
2021/07/15 21:44:12Z: EC2LaunchTelemetry: IsAgentScheduledPerBoot=true
2021/07/15 21:44:12Z: EC2LaunchTelemetry: IsUserDataScheduledPerBoot=true
2021/07/15 21:44:12Z: EC2LaunchTelemetry: AgentCommandCode=1
2021/07/15 21:44:12Z: EC2LaunchTelemetry: AgentCommandErrorCode=5
2021/07/15 21:44:12Z: EC2LaunchTelemetry: AgentInstallCode=2
2021/07/15 21:44:12Z: EC2LaunchTelemetry: AgentInstallErrorCode=0
```

원격 측정은 기본적으로 활성화됩니다. 언제든지 원격 측정 데이터 수집을 비활성화할 수 있습니다. 원격 측정이 활성화되면 EC2Launch v2는 별도로 고객에게 알리지 않고 원격 측정 데이터를 전송합니다.

원격 측정 가시성

원격 측정이 활성화되면 Amazon EC2 콘솔 출력에 다음과 같이 표시됩니다.

```
2021/07/15 21:44:12Z: Telemetry: <Data>
```

인스턴스의 원격 측정 비활성화

단일 인스턴스에 대한 원격 측정을 비활성화하려면 시스템 환경 변수를 설정하거나 MSI를 사용하여 설치 파일을 수정하면 됩니다.

시스템 환경 변수를 설정하여 원격 측정을 비활성화하려면 관리자로 다음 명령을 실행합니다.

```
setx /M EC2LAUNCH_TELEMETRY 0
```

MSI를 사용하여 원격 측정을 비활성화하려면 MSI를 다운로드한 후 다음 명령을 실행합니다.

```
msiexec /i ".\AmazonEC2Launch.msi" Remove="Telemetry" /q
```

EC2Launch v2에 대한 더 많은 주제

- 최신 버전의 EC2Launch v2 설치
- Windows 인스턴스에 대한 EC2Launch v2 설정 구성
- EC2Launch v2 시작 태스크에 대한 태스크 정의
- EC2Launch v2 에이전트와 관련된 문제 해결
- EC2Launch v2 버전 기록

최신 버전의 EC2Launch v2 설치

다음 방법 중 하나를 사용하여 EC2 인스턴스에 EC2Launch v2 에이전트를 설치할 수 있습니다.

- Amazon S3에서 에이전트를 다운로드하고 Windows PowerShell을 사용하여 설치합니다. 다운로드 URL은 Amazon S3에서 EC2Launch v2 다운로드을(를) 참조하세요.
- SSM Distributor를 사용하여 설치합니다.
- 사용자 지정 이미지를 생성할 때 EC2 Image Builder 구성 요소에서 설치합니다.
- EC2Launch v2가 사전 설치된 AMI에서 인스턴스를 시작합니다.

> **Warning**
>
> AmazonEC2Launch.msi는 EC2Launch(v1) 및 EC2Config와 같은 EC2 시작 서비스의 이전 버전을 제거합니다.

설치 단계를 보려면 원하는 방법과 일치하는 탭을 선택하세요.

PowerShell

Windows PowerShell을 사용하여 최신 버전의 EC2Launch v2 에이전트를 설치하려면 다음 단계를 따릅니다.

1. 로컬 디렉터리를 생성합니다.

```
New-Item -Path "$env:USERPROFILE\Desktop\EC2Launchv2" -ItemType Directory
```

2. 다운로드 위치의 URL을 설정합니다. 사용할 Amazon S3 URL로 다음 명령을 실행합니다. 다운로드 URL은 Amazon S3에서 EC2Launch v2 다운로드을(를) 참조하세요

```
$Url = "Amazon S3 URL/AmazonEC2Launch.msi"
```

3. 다음 복합 명령을 사용하여 에이전트를 다운로드하고 설치합니다

```
$DownloadFile = "$env:USERPROFILE\Desktop\EC2Launchv2\" + $(Split-Path -Path
 $Url -Leaf)
Invoke-WebRequest -Uri $Url -OutFile $DownloadFile
msiexec /i "$DownloadFile"
```

> **Note**
>
> Windows Server 2016 또는 이전 버전을 사용 중이고 파일을 다운로드할 때 오류가 발생하는 경우 PowerShell 터미널에서 TLS 1.2를 활성화해야 할 수 있습니다. 다음 명령을 사용하여 현재 PowerShell 세션에 대해 TLS 1.2를 활성화한 다음 다시 시도해보세요.
>
> ```
> [Net.ServicePointManager]::SecurityProtocol =
>  [Net.SecurityProtocolType]::Tls12
> ```

4. 이 msiexec 명령은 Windows Server 인스턴스의 다음 위치에 EC2Launch v2 버전을 설치합니다. `%ProgramFiles%\Amazon\EC2Launch` 인스턴스의 로컬 파일 시스템을 검사하여 설치가 실행되었는지 확인할 수 있습니다.

AWS Systems Manager Distributor

AWS Systems Manager 빠른 설정을 사용하여 EC2Launch v2의 자동 업데이트를 구성하려면 Distributor 빠른 설정을 사용하여 자동 설치 및 업데이트 섹션을 참조하세요.

AWS Systems Manager Distributor의 `AWSEC2Launch-Agent` 패키지를 한 번만 설치할 수도 있습니다. Systems Manager Distributor에서 패키지를 설치하는 방법에 대한 지침은 AWS Systems Manager 사용 설명서의 패키지 설치 또는 업데이트를 참조하세요.

EC2 Image Builder component

EC2 Image Builder 사용하여 사용자 지정 이미지를 빌드할 때 `ec2launch-v2-windows` 구성 요소를 설치할 수 있습니다. EC2 Image Builder 사용하여 사용자 지정 이미지를 빌드하는 방법에 대한 지침은 EC2 Image Builder 사용 설명서에서 EC2 Image Builder 콘솔 마법사를 사용하여 이미지 파이프라인 생성을 참조하세요.

AMI

EC2Launch v2는 다음 Windows Server 2022 이상 운영 체제의 AMI에 기본적으로 사전 설치되어 있습니다.

- Windows_Server-*version*-English-Full-Base
- Windows_Server-*version*-English-Core-Base
- Windows_Server-*version*-English-Core-EKS_Optimized
- 다른 모든 언어의 Windows Server *version* AMI
- SQL이 설치된 Windows Server *version* AMI

또한 EC2Launch v2는 다음 Windows Server AMI에도 사전 설치되어 있습니다. Amazon EC2 콘솔에서 또는 AWS CLI에서 검색 접두사 `EC2LaunchV2-`을(를) 사용하여 이러한 AMI를 찾을 수 있습니다.

- EC2LaunchV2-Windows_Server-2019-English-Core-Base
- EC2LaunchV2-Windows_Server-2019-English-Full-Base
- EC2LaunchV2-Windows_Server-2016-English-Core-Base
- EC2LaunchV2-Windows_Server-2016-English-Full-Base

AWS Systems Manager Distributor 빠른 설정을 사용하여 EC2Launch v2 자동 설치 및 업데이트

AWS Systems Manager Distributor 빠른 설정을 사용하면 EC2Launch v2의 자동 업데이트를 설정할 수 있습니다. 다음 프로세스는 지정한 빈도에 따라 EC2Launch v2 에이전트를 자동으로 업데이트하는 Systems Manager 연결을 인스턴스에 설정합니다. Distributor 빠른 설치가 생성하는 연결에는 AWS 계정 및 리전 내의 인스턴스 또는 AWS 조직 내의 인스턴스가 포함될 수 있습니다. 조직 설정에 대한 자세한 내용은 AWS Organizations 사용 설명서의 자습서: 조직 생성 및 구성을 참조하세요.

시작하기 전에 인스턴스가 모든 사전 조건을 충족하는지 확인합니다.

사전 조건

Distributor 빠른 설정으로 자동 업데이트를 설정하려면 인스턴스가 다음 사전 조건을 충족해야 합니다.

- EC2Launch v2를 지원하는 하나 이상의 실행 중인 인스턴스가 있습니다. EC2Launch v2에 지원되는 운영 체제를 확인합니다.
- 인스턴스에서 Systems Manager 설치 작업을 수행했습니다. 자세한 내용은 AWS Systems Manager 사용 설명서의 Systems Manager 설정을 참조하세요.
- EC2Launch v2가 인스턴스에 설치된 유일한 시작 에이전트여야 합니다. 시작 에이전트가 두 개 이상 설치된 경우 Distributor 빠른 설치 구성에 실패합니다. Distributor 빠른 설정을 사용하여 EC2Launch v2를 구성하기 전에 EC2Config 또는 EC2Launch v1 시작 에이전트가 있다면 제거합니다.

EC2Launch v2용 Distributor 빠른 설정 구성

Distributor 빠른 설정을 사용하여 EC2Launch v2의 구성을 생성하려면 Distributor 패키지 배포 단계를 완료한 후 다음 설정을 사용합니다.

- 소프트웨어 패키지: Amazon EC2Launch v2 에이전트입니다.
- 업데이트 빈도: 목록에서 빈도를 선택합니다.
- 대상: 사용 가능한 배포 옵션 중에서 선택합니다.

구성 상태를 확인하려면 AWS Management Console에서 Systems Manager 빠른 설정 구성 탭으로 이동합니다.

1. AWS Systems Manager 콘솔(https://console.aws.amazon.com/systems-manager/)을 엽니다.
2. 탐색 창에서 빠른 설정을 선택합니다.
3. 구성 탭에서 생성한 구성과 관련된 행을 선택합니다. 구성 탭에는 구성이 나열되며 리전, 배포 상태, 연결 상태와 같은 주요 세부 정보가 요약되어 있습니다.

   > **Note**
   >
   > 모든 EC2Launch v2 Distributor 구성의 연결 이름은 `AWS-QuickSetup-Distributor-EC2Launch-Agent-` 접두사로 시작합니다.

4. 세부 정보를 보려면 구성을 선택하고 세부 정보 보기를 선택합니다.

자세한 내용과 문제 해결 단계는 AWS Systems Manager 사용 설명서의 빠른 설정 결과 문제 해결을 참조하세요.

Amazon S3에서 EC2Launch v2 다운로드

EC2Launch v2의 최신 버전을 설치하려면 다음 위치에서 설치 관리자를 다운로드합니다.

- https://s3.amazonaws.com/amazon-ec2launch-v2/windows/amd64/latest/AmazonEC2Launch.msi

설치 옵션 구성

EC2Launch v2를 설치하거나 업그레이드할 때 EC2Launch v2 설치 대화 상자 또는 명령줄 쉘에서 msiexec 명령을 사용하여 설치 옵션을 구성할 수 있습니다.

EC2Launch v2 설치 프로그램은 인스턴스에서 처음 실행될 때 다음과 같이 인스턴스의 시작 에이전트 설정을 초기화합니다.

- 로컬 경로를 생성하고 해당 경로에 시작 에이전트 파일을 기록합니다. 이를 새로 설치라고도 합니다.
- EC2LAUNCH_TELEMETRY 환경 변수가 아직 존재하지 않는 경우 생성하고 구성에 따라 설정합니다.

구성 세부 정보를 보려면 사용할 구성 방법과 일치하는 탭을 선택하세요.

Amazon EC2Launch Setup dialog

EC2Launch v2를 설치하거나 업그레이드할 때 EC2Launch v2 설치 대화 상자를 통해 다음 설치 옵션을 구성할 수 있습니다.

기본 설치 옵션

원격 측정 전송

설치 대화 상자에서 이 기능을 포함하면 설치 프로그램은 EC2LAUNCH_TELEMETRY 환경 변수를 값 1(으)로 설정합니다. 원격 측정 전송을 비활성화하면 설치 프로그램은 환경 변수를 값 0(으)로 설정합니다.

EC2Launch v2 에이전트가 실행되면 EC2LAUNCH_TELEMETRY 환경 변수를 읽어 원격 측정 데이터를 업로드할지 여부를 결정합니다. 값이 1이면 데이터를 업로드합니다. 그렇게 하지 않으면 업로드하지 않습니다.

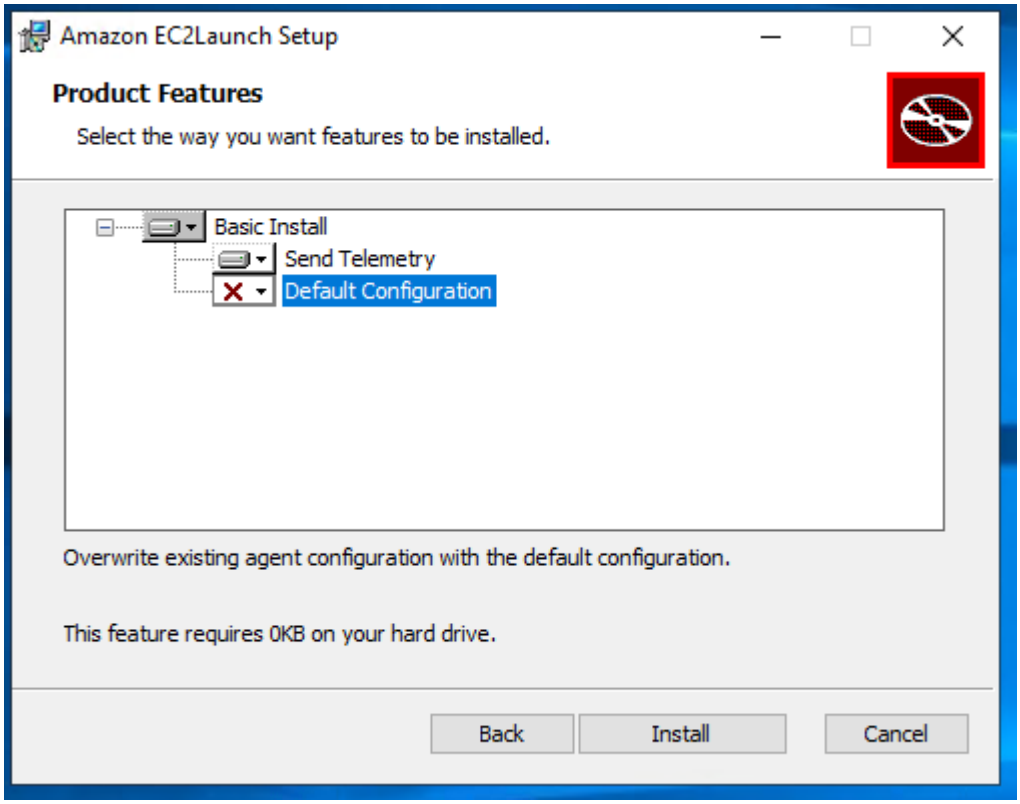
기본 구성

EC2Launch v2의 기본 구성은 로컬 시작 에이전트가 이미 존재하는 경우 이를 덮어쓰는 것입니다. 인스턴스에서 처음 설치를 실행하면 기본 구성은 새로 설치를 수행합니다. 처음 설치할 때 기본 구성을 비활성화하면 설치가 실패합니다.

인스턴스에서 설치를 다시 실행하는 경우 기본 구성을 비활성화하여 `%ProgramData%/Amazon/EC2Launch/config/agent-config.yml` 파일을 바꾸지 않는 업그레이드를 수행할 수 있습니다.

예: 원격 측정을 사용하는 EC2Launch v2 업그레이드

다음 예는 현재 설치를 업그레이드하고 원격 측정을 활성화하도록 구성된 EC2Launch v2 설치 대화 상자를 보여줍니다. 이 구성은 에이전트 구성 파일을 바꾸지 않고 설치를 수행하며 `EC2LAUNCH_TELEMETRY` 환경 변수를 값 1(으)로 설정합니다.



Command line

EC2Launch v2를 설치하거나 업그레이드할 때 명령줄 쉘에서 msiexec 명령을 사용하여 설치 옵션을 구성할 수 있습니다.

ADDLOCAL 파라미터 값

Basic(필수)

시작 에이전트를 설치합니다. 이 값이 ADDLOCAL 파라미터에 없으면 설치가 종료됩니다.

친환경

ADDLOCAL 파라미터에 Clean 값을 포함하면 설치 프로그램은 에이전트 구성 파일을 %ProgramData%/Amazon/EC2Launch/config/agent-config.yml 위치에 기록합니다. 에이전트 구성 파일이 이미 존재하는 경우 해당 파일을 덮어씁니다.

ADDLOCAL 파라미터에서 Clean 값을 제외하면 설치 프로그램은 에이전트 구성 파일을 바꾸지 않는 업그레이드를 수행합니다.

원격 측정

ADDLOCAL 파라미터에 Telemetry 값을 포함하면 설치 프로그램은 EC2LAUNCH_TELEMETRY 환경 변수를 값 1(으)로 설정합니다.

ADDLOCAL 파라미터에서 Telemetry 값을 제외하면 설치 프로그램은 환경 변수를 값 0(으)로 설정합니다.

EC2Launch v2 에이전트가 실행되면 EC2LAUNCH_TELEMETRY 환경 변수를 읽어 원격 측정 데이터를 업로드할지 여부를 결정합니다. 값이 1이면 데이터를 업로드합니다. 그렇게 하지 않으면 업로드하지 않습니다.

예: 원격 측정을 사용하는 EC2Launch v2 설치

```
& msiexec /i "C:\Users\Administrator\Desktop\EC2Launchv2\AmazonEC2Launch.msi"
 ADDLOCAL="Basic,Clean,Telemetry" /q
```

EC2Launch v2 버전 확인

다음 절차 중 하나를 사용하여 인스턴스에 설치된 EC2Launch v2의 버전을 확인합니다.

PowerShell

다음과 같이 Windows PowerShell을 사용하여 설치된 EC2Launch v2 버전을 확인합니다.

1. AMI에서 인스턴스를 실행해서 여기에 연결합니다.
2. PowerShell에서 다음 명령을 실행하여 설치된 EC2Launch v2 버전을 확인합니다.

```
& "C:\Program Files\Amazon\EC2Launch\EC2Launch.exe" version
```

Windows Control Panel

다음과 같이 Windows 제어판에서 설치된 EC2Launch v2 버전을 확인합니다.

1. AMI에서 인스턴스를 실행해서 여기에 연결합니다.
2. Windows 제어판을 연 다음 프로그램 및 기능을 선택합니다.
3. 설치된 프로그램 목록에서 `Amazon EC2Launch`을(를) 찾습니다. 버전 번호가 버전 열에 표시됩니다.

AWS Windows AMI의 최신 업데이트를 보려면 AWS Windows AMI 참조의 Windows AMI version history를 참조하세요.

최신 버전의 EC2Launch v2는 EC2Launch v2 버전 기록 섹션을 참조하세요.

EC2Launch v2 마이그레이션 도구의 최신 버전은 EC2Launch v2 마이그레이션 도구 버전 기록 섹션을 참조하세요.

새로운 EC2Launch v2 서비스 버전이 릴리스되면 알림을 받을 수 있습니다. 자세한 내용은 EC2 Windows 시작 에이전트 알림 구독 섹션을 참조하세요.

Windows 인스턴스에 대한 EC2Launch v2 설정 구성

이 섹션에서는 EC2Launch v2에 대한 설정을 구성하는 방법을 설명합니다.

포함된 주제:

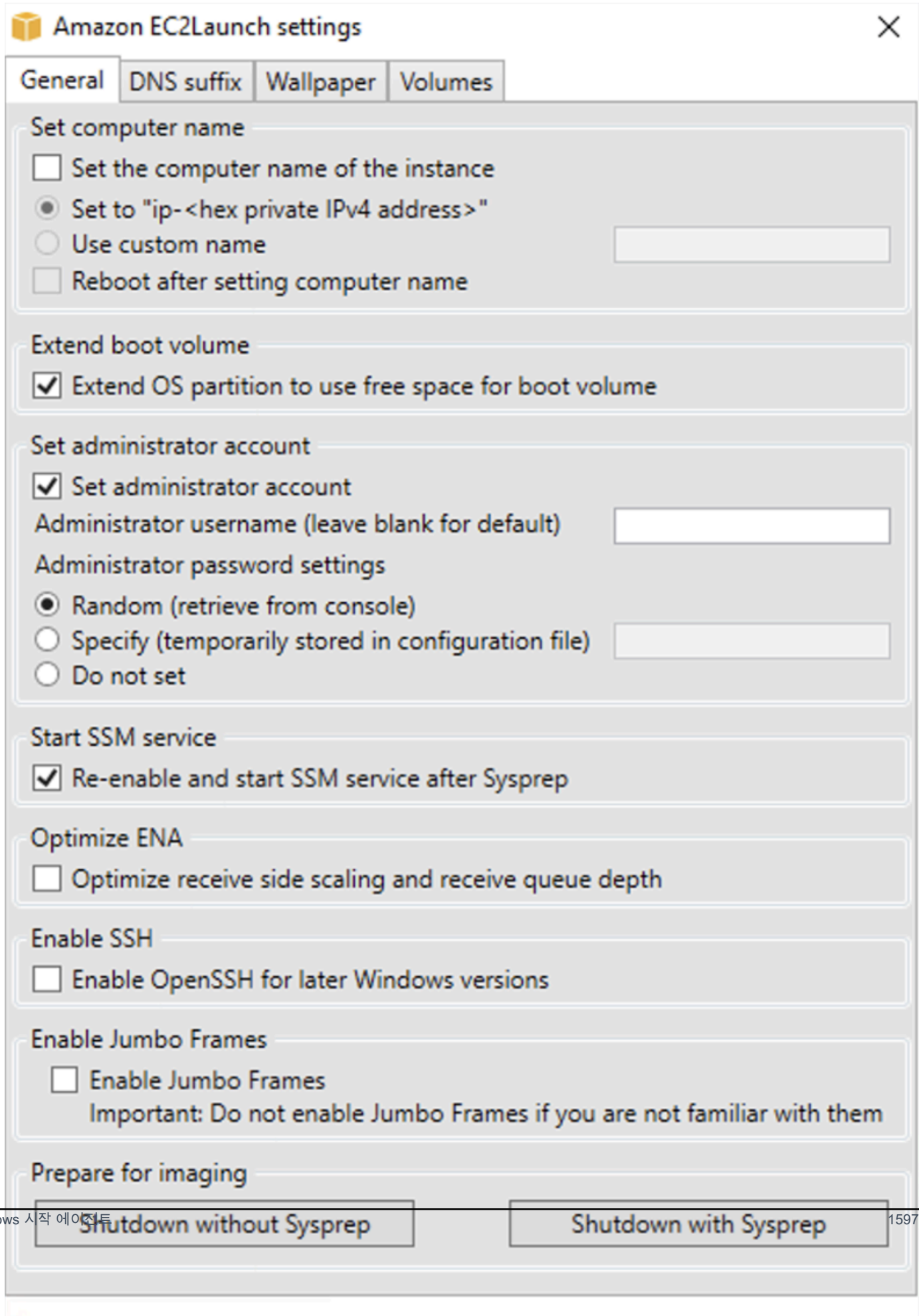
- EC2Launch v2 설정 대화 상자를 사용하여 설정 변경
- CLI를 사용하여 EC2Launch v2 구성
- EC2Launch v2 태스크 구성
- EC2Launch v2 종료 코드 및 재부팅
- EC2Launch v2 및 Sysprep

EC2Launch v2 설정 대화 상자를 사용하여 설정 변경

다음 절차는 EC2Launch v2 설정 대화 상자를 사용하여 설정을 사용하거나 비활성화하는 방법을 설명합니다.

> **Note**
>
> agent-config.yml 파일에서 사용자 정의 태스크를 잘못 구성하고 Amazon EC2Launch 설정 대화 상자를 열려고 하는 경우 오류가 발생합니다. 스키마 예제는 예: `agent-config.yml` 섹션을 참조하세요.

1. 실행을 시작해서 Windows 인스턴스에 연결합니다.
2. 시작 메뉴에서 모든 프로그램을 선택한 다음 EC2Launch 설정으로 이동합니다.


Amazon EC2Launch settings
General
DNS suffix
Wallpaper
Volumes
Set computer name
Set the computer name of the instance
Set to "ip-<hex private IPv4 address>"
Use custom name
Reboot after setting computer name
Extend boot volume
Extend OS partition to use free space for boot volume
Set administrator account
Set administrator account
Administrator username (leave blank for default)
Administrator password settings
Random (retrieve from console)
Specify (temporarily stored in configuration file)
Do not set
Start SSM service
Re-enable and start SSM service after Sysprep
Optimize ENA
Optimize receive side scaling and receive queue depth
Enable SSH
Enable OpenSSH for later Windows versions
Enable Jumbo Frames
Enable Jumbo Frames
Important: Do not enable Jumbo Frames if you are not familiar with them
Prepare for imaging
Shutdown without Sysprep
Shutdown with Sysprep

3. EC2Launch 설정 대화 상자의 일반 탭에서 다음 설정을 활성화 또는 비활성화할 수 있습니다.

   a. 컴퓨터 이름 설정

      이 설정을 활성화하면(기본적으로 비활성화됨) 부팅 때마다 현재 호스트 이름이 원하는 호스트 이름과 비교됩니다. 호스트 이름이 일치하지 않는 경우 호스트 이름이 재설정되고, 선택적으로 시스템이 재부팅되어 새 호스트 이름을 선택합니다. 사용자 지정 호스트 이름을 지정하지 않으면 16진수 형식의 프라이빗 IPv4 주소(예: `ip-AC1F4E6`)를 사용하여 생성됩니다. 기존 호스트 이름이 변경되는 것을 방지하려면 이 설정을 활성화하지 마세요.

   b. 부트 볼륨 확장

      이 설정은 파티션 처리되지 않은 공간을 모두 포함하도록 동적으로 `Disk 0/Volume 0`을 확장합니다. 이는 인스턴스가 사용자 설정 크기를 가진 루트 디바이스 볼륨에서 부팅될 때 유용할 수 있습니다.

   c. 관리자 계정 설정

      이 기능이 활성화되면 로컬 시스템에 생성된 관리자 계정의 사용자 이름 및 암호 속성을 설정할 수 있습니다. 이 기능이 활성화되지 않으면 Sysprep 이후에 시스템에서 관리자 계정이 생성되지 않습니다. `adminPassword`이 `adminPasswordtype`인 경우에만 `Specify`에 암호를 입력합니다.

      암호 유형은 다음과 같이 정의됩니다.

      i. `Random`

         EC2Launch는 암호를 생성하고 사용자의 키를 사용하여 암호를 암호화합니다. 인스턴스가 재부팅 또는 중지되었다가 시작된 경우 이 암호가 그대로 유지되도록 시스템은 인스턴스가 시작된 후 이 설정을 비활성화합니다.

      ii. `Specify`

         에 지정한 암호가 EC2Launch에 사용됩니다`adminPassword` 암호가 시스템 요구 사항에 맞지 않으면 EC2Launch에서 임의의 암호를 대신 생성합니다. 암호는 `agent-config.yml`에 일반 텍스트로 저장되며 Sysprep에서 관리자 암호를 설정한 후에 삭제됩니다. EC2Launch는 사용자의 키를 사용하여 암호를 암호화합니다.

      iii. `Do not set`

         unattend.xml 파일에 지정한 암호가 EC2Launch에 사용됩니다. unattend.xml에 암호를 지정하지 않으면 관리자 계정이 비활성화됩니다.

d. SSM 서비스 시작

선택하면 Systems Manager 서비스가 사용되어 다음 Sysprep이 시작됩니다. EC2Launch v2는 앞서 설명한 모든 작업을 수행하며 SSM Agent는 Run Command 및 State Manager와 같은 Systems Manager 기능에 대한 요청을 처리합니다.

Run Command를 사용하여 기존 인스턴스가 최신 버전의 EC2Launch v2 서비스와 SSM Agent를 사용하도록 업그레이드할 수 있습니다. 자세한 내용은 AWS Systems Manager 사용 설명서에서 Run Command를 사용하여 SSM Agent 업데이트를 참조하세요.

e. ENA 최적화

이 옵션을 선택하면 ENA 수신측 조정 및 수신 대기열 깊이 설정이 AWS에 맞게 최적화되도록 ENA 설정이 구성됩니다. 자세한 내용은 수신 측 조정 CPU 선호도 구성 섹션을 참조하세요.

f. SSH 활성화

이 설정을 사용하면 최신 Windows 버전에서 원격 시스템 관리가 가능하도록 OpenSSH가 활성화됩니다.

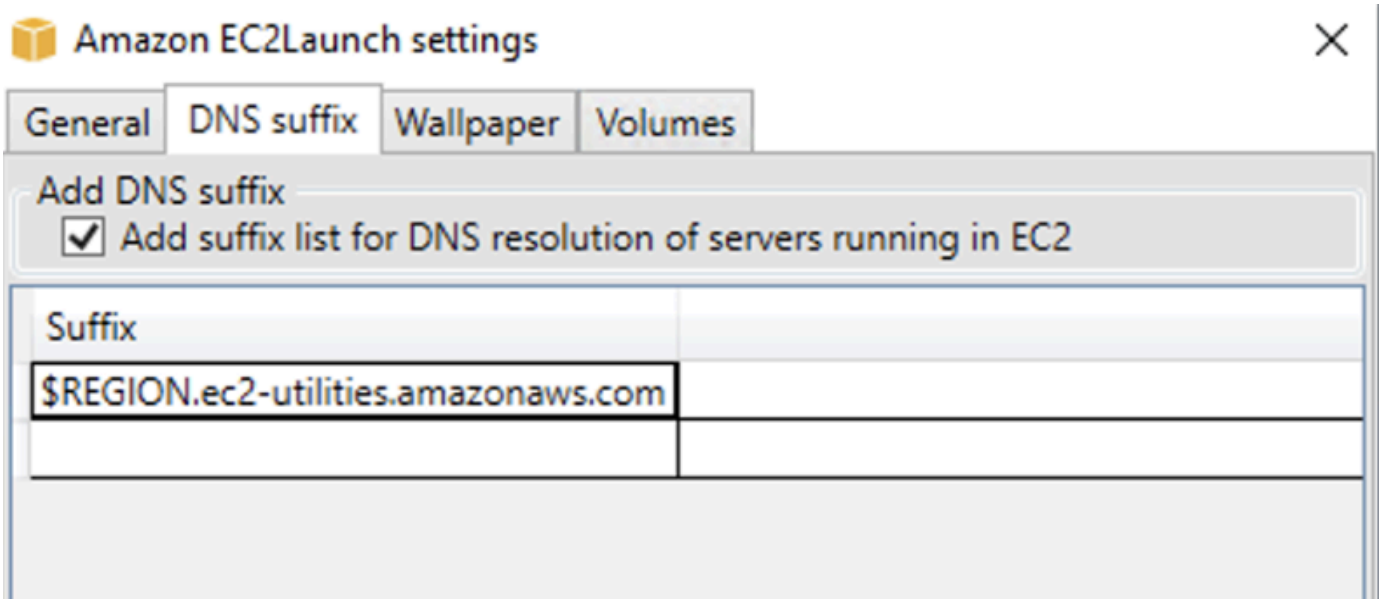
g. 점보 프레임 활성화

점보 프레임을 활성화하려면 선택합니다. 점보 프레임은 네트워크 통신에 의도하지 않은 영향을 줄 수 있으므로 점보 프레임을 활성화하기 전에 시스템에 미칠 수 있는 영향을 파악해야 합니다. 점보 프레임에 대한 자세한 내용은 점보 프레임(9001 MTU) 섹션을 참조하세요.

h. 이미징 준비

EC2 인스턴스를 종료할 때 Sysprep을 실행할지 아니면 실행하지 않을지를 선택합니다. EC2Launch v2에서 Sysprep을 실행하려면 [Sysprep을 실행하여 종료(Shutdown with Sysprep)]를 선택합니다.

4. DNS 접미사 탭에서, 정규화된 도메인 이름을 제공하지 않고 EC2에서 실행되는 서버의 DNS를 확인할 수 있도록 DNS 접미사 목록의 추가 여부를 선택할 수 있습니다. DNS 접미사는 변수 `$REGION` 및 `$AZ`를 포함할 수 있습니다. 아직 존재하지 않는 접미사만 목록에 추가됩니다.

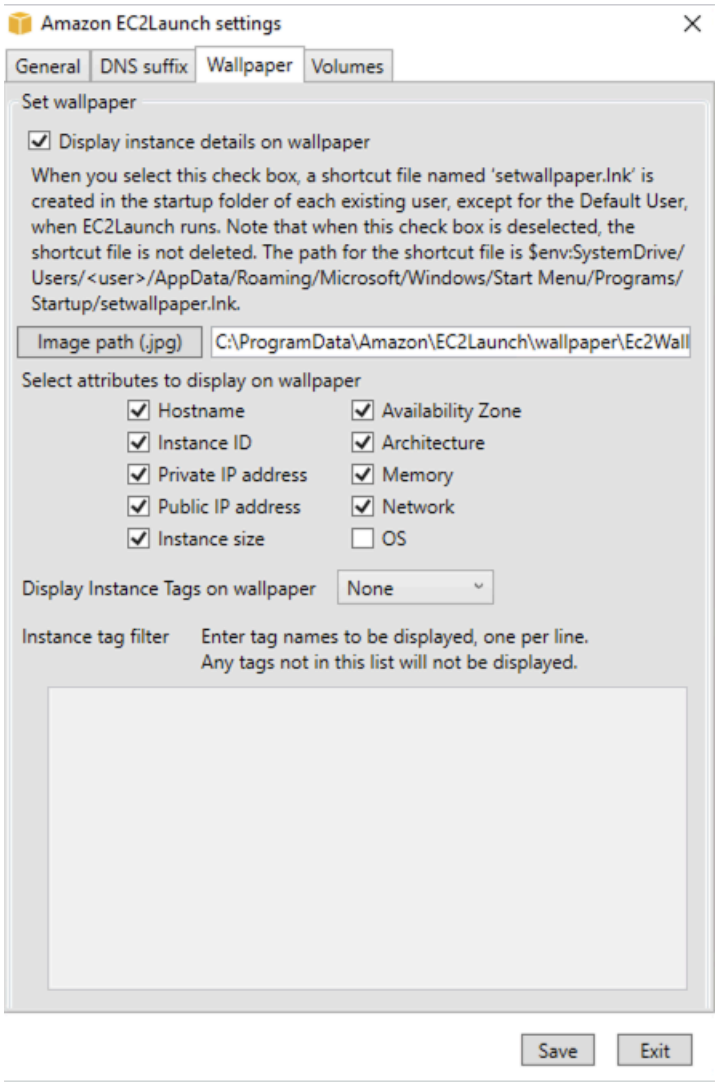


5. 월페이퍼 탭에서 배경 이미지로 인스턴스 월페이퍼를 구성하고 표시할 월페이퍼에 대한 인스턴스 세부 정보를 지정할 수 있습니다. Amazon EC2는 로그인할 때마다 세부 정보를 생성합니다.

   다음 컨트롤로 월페이퍼를 구성할 수 있습니다.

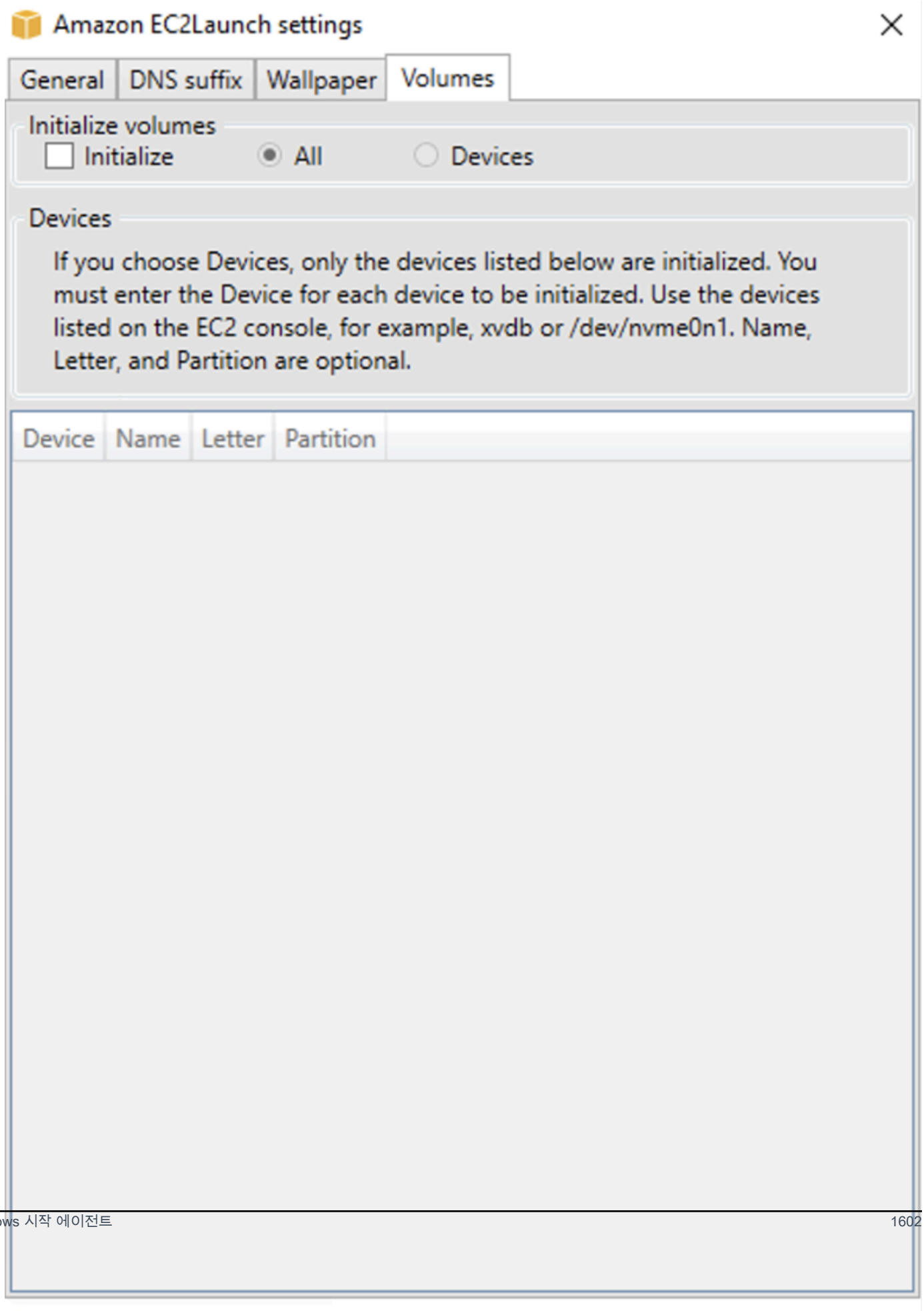
   - 월페이퍼에 인스턴스 세부 정보 표시 - 이 확인란은 월페이퍼의 인스턴스 세부 정보 표시를 활성화하거나 비활성화합니다.
   - 이미지 경로(.jpg) - 월페이퍼 배경으로 사용할 이미지의 경로를 지정합니다.
   - 월페이퍼에 표시할 속성 선택 - 월페이퍼에 표시하려는 인스턴스 세부 정보의 확인란을 선택합니다. 월페이퍼에서 인스턴스 세부 정보를 제거하려면 확인란 선택을 취소합니다.
   - 월페이퍼에 인스턴스 태그 표시 - 월페이퍼에 인스턴스 태그를 표시하려면 다음 설정 중 하나를 선택합니다.
     - 없음 - 월페이퍼에 인스턴스 태그를 표시하지 않습니다.
     - 모두 표시 - 월페이퍼에 모든 인스턴스 태그를 표시합니다.
     - 필터링된 항목 표시 - 월페이퍼에 지정된 인스턴스 태그를 표시합니다. 이 설정을 선택하면 인스턴스 태그 필터 상자에서 배경화면에 표시할 인스턴스 태그를 추가할 수 있습니다.

     > **Note**
     >
     > 월페이퍼에 태그를 표시하려면 메타데이터에서 태그를 활성화해야 합니다. 인스턴스 태그 및 메타데이터에 대한 자세한 내용은 인스턴스 메타데이터를 사용하여 EC2 인스턴스의 태그 보기 섹션을 참조하세요.



6. 볼륨 탭에서 인스턴스에 연결된 볼륨을 초기화할지 여부를 선택합니다. 활성화하면 추가 볼륨의 드라이브 문자가 설정되고 사용 가능한 공간을 사용하도록 확장됩니다. 모두를 선택하면 모든 스토리지 볼륨이 초기화됩니다. 디바이스를 선택하면 목록에 지정된 디바이스만 초기화됩니다. 초기화할 각 디바이스에 대해 디바이스를 입력해야 합니다. EC2 콘솔에 나열된 디바이스(예: `xvdb` 또는 `/dev/nvme0n1`)를 사용합니다. 드롭다운 목록에는 인스턴스에 연결된 스토리지 볼륨이 표시됩니다. 인스턴스에 연결되지 않은 디바이스를 입력하려면 텍스트 필드에 디바이스를 입력합니다.

   이름, 문자 및 파티션은 선택적인 필드입니다. 파티션에 값을 지정하지 않으면 2TB보다 큰 스토리지 볼륨은 `gpt` 파티션 유형으로 초기화되고 2TB보다 작은 스토리지 볼륨은 `mbr` 파티션 유형으로 초기화됩니다. 디바이스가 구성되고 NTFS 이외의 디바이스에 파티션 테이블이 포함되어 있거나 디스크의 처음 4KB에 데이터가 포함된 경우 디스크를 건너뛰고 작업이 기록됩니다.


Amazon EC2Launch settings
General
DNS suffix
Wallpaper
Volumes
Initialize volumes
Initialize
All
Devices
Devices
If you choose Devices, only the devices listed below are initialized. You must enter the Device for each device to be initialized. Use the devices listed on the EC2 console, for example, xvdb or /dev/nvme0n1. Name, Letter, and Partition are optional.
Device
Name
Letter
Partition

다음은 EC2Launch 대화 상자에 입력한 설정에서 생성된 구성 YAML 파일의 예입니다.

```
version: 1.0
config:
  - stage: boot
 tasks:
      - task: extendRootPartition
  - stage: preReady
    tasks:
      - task: activateWindows
        inputs:
          activation:
            type: amazon
      - task: setDnsSuffix
        inputs:
          suffixes:
            - $REGION.ec2-utilities.amazonaws.com
      - task: setAdminAccount
        inputs:
          password:
            type: random
      - task: setWallpaper
        inputs:
          path: C:\ProgramData\Amazon\EC2Launch\wallpaper\Ec2Wallpaper.jpg
          attributes:
            - hostName
            - instanceId
            - privateIpAddress
            - publicIpAddress
            - instanceSize
            - availabilityZone
            - architecture
            - memory
            - network
  - stage: postReady
    tasks:
      - task: startSsm
```

CLI를 사용하여 EC2Launch v2 구성

CLI(명령줄 인터페이스)를 사용하여 EC2Launch 설정을 구성하고 서비스를 관리할 수 있습니다. 다음 섹션에서는 EC2Launch v2를 관리하는 데 사용할 수 있는 CLI 명령에 대한 설명과 사용법 정보를 제공합니다.

명령

- collect-logs
- get-agent-config
- list-volumes
- reset
- run
- status
- sysprep
- 검증
- version
- wallpaper

collect-logs

EC2Launch에 대한 로그 파일을 수집하고, 파일을 압축하여 지정된 디렉터리에 배치합니다.

예제

```
ec2launch collect-logs -o C:\Mylogs.zip
```

사용량

`ec2launch collect-logs [flags]`

Flags

`-h, --help`

`collect-logs`에 대한 도움말

`-o, --output string`

압축된 출력 로그 파일 경로

get-agent-config

`agent-config.yml`을 지정된 형식(JSON 또는 YAML)으로 인쇄합니다. 형식이 지정되지 않은 경우 `agent-config.yml`은 이전에 지정한 형식으로 인쇄됩니다.

예제

```
ec2launch get-agent-config -f json
```

사용량

`ec2launch get-agent-config [flags]`

Flags

`-h, --help`

`get-agent-config`에 대한 도움말

`-f, --format string`

`agent-config` 파일의 출력 형식: `json`, `yaml`

list-volumes

휘발성 및 EBS 볼륨을 포함하여 인스턴스에 연결된 모든 스토리지 볼륨을 나열합니다.

예제

```
ec2launch list-volumes
```

사용량

`ec2launch list-volumes`

Flags

`-h, --help`

`list-volumes`에 대한 도움말

reset

이 태스크의 주요 목표는 에이전트가 다음에 실행될 때 에이전트를 재설정하는 것입니다. 이 목표를 위해 reset 명령은 로컬 `EC2Launch` 디렉터리에서 EC2Launch v2의 모든 에이전트 상태 데이터를 삭제합니다(EC2Launch v2 디렉터리 구조 단원 참조) 재설정할 경우 서비스 및 Sysprep 로그가 선택적으로 삭제됩니다.

스크립트 동작은 에이전트가 스크립트를 실행하는 모드 (인라인 모드 또는 분리 모드)에 따라 달라집니다.

인라인(기본값)

EC2Launch v2 에이전트는 한 번에 하나씩(`detach: false`) 스크립트를 실행합니다. 이것이 기본 설정입니다.

> **Note**
>
> 인라인 스크립트가 reset 또는 sysprep 명령을 내리면 해당 명령이 즉시 실행되고 에이전트가 재설정됩니다. 현재 태스크가 끝나면 에이전트가 추가 태스크를 실행하지 않고 종료됩니다.
> 예를 들어 명령을 실행하는 태스크 뒤에 `startSsm` 태스크(사용자 데이터 실행 후 기본적으로 포함됨)가 뒤따르는 경우 해당 태스크가 실행되지 않고 Systems Manager 서비스도 시작되지 않습니다.

분리됨

EC2Launch v2 에이전트는 다른 태스크(`detach: true`)와 동시에 스크립트를 실행합니다.

> **Note**
>
> 분리된 스크립트가 reset 또는 sysprep 명령을 내리면 해당 명령은 실행되기 전에 에이전트가 완료될 때까지 기다립니다. ExecuteScript 이후의 태스크는 계속 실행됩니다.

예제

```
ec2launch reset -c
```

사용량

```
ec2launch reset [flags]
```

Flags

`-c`, `--clean`

reset 전에 인스턴스 로그 정리

-h, --help

reset에 대한 도움말

run

EC2Launch v2를 실행합니다.

예제

```
ec2launch run
```

사용량

ec2launch run [flags]

Flags

-h, --help

run에 대한 도움말

status

EC2Launch v2 에이전트의 상태를 가져옵니다. 필요한 경우 에이전트가 완료될 때까지 프로세스를 차단합니다. 에이전트 상태는 프로세스 종료 코드에 따라 결정됩니다.

- 0 – 에이전트가 실행되었고 성공했습니다.
- 1 – 에이전트가 실행되었지만 실패했습니다.
- 2 – 에이전트가 여전히 실행 중입니다.
- 3 – 에이전트가 알 수 없는 상태입니다. 에이전트가 실행 중이 아니거나 중지된 상태입니다.
- 4 – 에이전트 상태를 검색하는 동안 오류가 발생했습니다.
- 5 – 에이전트가 실행 중이 아니며 마지막으로 알려진 실행 상태를 알 수 없습니다. 이는 다음 중 하나를 의미할 수 있습니다.
  - state.json 및 previous-state.json 모두 삭제되었습니다.
  - previous-state.json이 손상되었습니다.

reset 명령을 실행한 후의 에이전트 상태입니다.

예시:

```
ec2launch status -b
```

사용량

```
ec2launch status [flags]
```

Flags

`-b,--block`

에이전트 실행이 완료될 때까지 프로세스를 차단합니다.

`-h,--help`

`status`에 대한 도움말

sysprep

이 태스크의 주요 목표는 에이전트가 다음에 실행될 때 에이전트를 재설정하는 것입니다. 이를 위해 sysprep 명령은 에이전트 상태를 재설정하고, `unattend.xml` 파일을 업데이트하고, RDP를 비활성화하고, Sysprep을 실행합니다.

스크립트 동작은 에이전트가 스크립트를 실행하는 모드 (인라인 모드 또는 분리 모드)에 따라 달라집니다.

인라인(기본값)

EC2Launch v2 에이전트는 한 번에 하나씩(`detach: false`) 스크립트를 실행합니다. 이것이 기본 설정입니다.

> **Note**
>
> 인라인 스크립트가 reset 또는 sysprep 명령을 내리면 해당 명령이 즉시 실행되고 에이전트가 재설정됩니다. 현재 태스크가 끝나면 에이전트가 추가 태스크를 실행하지 않고 종료됩니다.

예를 들어 명령을 실행하는 태스크 뒤에 `startSsm` 태스크(사용자 데이터 실행 후 기본적으로 포함됨)가 뒤따르는 경우 해당 태스크가 실행되지 않고 Systems Manager 서비스도 시작되지 않습니다.

분리됨

EC2Launch v2 에이전트는 다른 태스크(`detach: true`)와 동시에 스크립트를 실행합니다.

> **Note**
>
> 분리된 스크립트가 reset 또는 sysprep 명령을 내리면 해당 명령은 실행되기 전에 에이전트가 완료될 때까지 기다립니다. ExecuteScript 이후의 태스크는 계속 실행됩니다.

예시:

```
ec2launch sysprep
```

사용량

`ec2launch sysprep [flags]`

Flags

`-c,--clean`

`sysprep` 전에 인스턴스 로그 정리

`-h,--help`

Sysprep에 대한 도움말

`-s,--shutdown`

`sysprep` 후에 인스턴스 종료

검증

`agent-config` 파일 `C:\ProgramData\Amazon\EC2Launch\config\agent-config.yml`을 검증합니다

예제

```
ec2launch validate
```

사용량

`ec2launch validate [flags]`

Flags

-h , `--help`

`validate`에 대한 도움말

version

실행 가능한 버전을 가져옵니다.

예제

```
ec2launch version
```

사용량

`ec2launch version [flags]`

Flags

`-h, --help`

`version`에 대한 도움말

wallpaper

제공된 월페이퍼 경로(.jpg 파일)로 새 월페이퍼를 설정하고 선택한 인스턴스 세부 정보를 표시합니다.

구문

```
ec2launch wallpaper ^
--path="C:\ProgramData\Amazon\EC2Launch\wallpaper\Ec2Wallpaper.jpg" ^
--all-tags ^
--
attributes=hostName,instanceId,privateIpAddress,publicIpAddress,instanceSize,availabilityZone,a
```

입력

파라미터

--allowed-tags [***tag-name-1***, ***tag-name-n***]

(선택 사항) 월페이퍼에 표시할 인스턴스 태그 이름의 Base64로 인코딩된 JSON 배열입니다. 이 태그 또는 `--all-tags`를 사용할 수 있지만 둘 다 사용할 수는 없습니다.

--attributes ***attribute-string-1***, ***attribute-string-n***

(선택 사항) 월페이퍼에 설정을 적용하기 위한 쉼표로 구분된 `wallpaper` 속성 문자열 목록입니다.

[--path | -p] ***path-string***

(필수) `wallpaper` 배경 이미지 파일 경로를 지정합니다.

플래그

--all-tags

(선택 사항) 월페이퍼에 모든 인스턴스 태그를 표시합니다. 이 태그 또는 `--allowed-tags`를 사용할 수 있지만 둘 다 사용할 수는 없습니다.

[--help | -h]

wallpaper 명령에 대한 도움말을 표시합니다.

EC2Launch v2 태스크 구성

이 섹션에는 `agent-config.yml` 및 사용자 데이터에 대한 구성 스키마, 작업, 세부 정보, 예시가 나와 있습니다.

작업 및 예시

- 스키마: agent-config.yml
- 시작 또는 재부팅 중 실행되는 EC2Launch v2 사용자 데이터 스크립트 구성

스키마: **`agent-config.yml`**

`agent-config.yml` 파일의 구조는 아래에 나와 있습니다. 동일한 단계에서는 작업을 반복할 수 없습니다. 작업 속성에 대해서는 다음 작업 설명을 참조하세요.

문서 구조: agent-config.yml

JSON

```
{
 "version": "1.1",
 "config": [
  {
   "stage": "string",
   "tasks": [
    {
     "task": "string",
     "inputs": {
      ...
     }
    },
    ...
   ]
  },
  ...
 ]
}
```

YAML

```
version: 1.1
config:
- stage: string
  tasks:
  - task: string
 inputs:
   ...
  ...
...
```

예: **agent-config.yml**

다음 예에서는 agent-config.yml 구성 파일에 대한 설정을 보여줍니다.

```
version: 1.1
config:
- stage: boot
```

```
  tasks:
  - task: extendRootPartition
- stage: preReady
  tasks:
  - task: activateWindows
    inputs:
      activation:
        type: amazon
  - task: setDnsSuffix
    inputs:
      suffixes:
      - $REGION.ec2-utilities.amazonaws.com
  - task: setAdminAccount
    inputs:
      password:
        type: random
  - task: setWallpaper
    inputs:
      path: C:\ProgramData\Amazon\EC2Launch\wallpaper\Ec2Wallpaper.jpg
      attributes:
      - hostName
      - instanceId
      - privateIpAddress
      - publicIpAddress
      - instanceSize
      - availabilityZone
      - architecture
      - memory
      - network
- stage: postReady
  tasks:
  - task: startSsm
```

시작 또는 재부팅 중 실행되는 EC2Launch v2 사용자 데이터 스크립트 구성

다음 JSON 및 YAML 예는 사용자 데이터의 문서 구조를 보여줍니다. Amazon EC2는 문서에서 지정한 `tasks` 배열에 이름이 지정된 각 작업을 구문 분석합니다. 각 작업에는 일련의 자체 속성 및 요구 사항이 있습니다. 자세한 내용은 EC2Launch v2 시작 태스크에 대한 태스크 정의를 참조하세요.

> **Note**
>
> 태스크는 사용자 데이터 태스크 배열에 한 번만 나타나야 합니다.

문서 구조: 사용자 데이터

JSON

```
{
 "version": "1.1",
 "tasks": [
  {
   "task": "string",
   "inputs": {
    ...
   },
  },
  ...
 ]
}
```

YAML

```
version: 1.1
tasks:
- task: string
  inputs:
    ...
...
```

예: 사용자 데이터

사용자 데이터에 대한 자세한 내용은 Amazon EC2가 Windows 인스턴스의 사용자 데이터를 처리하는 방법 섹션을 참조하세요.

다음 YAML 문서 예제는 EC2Launch v2가 파일을 생성하기 위해 사용자 데이터로 실행하는 PowerShell 스크립트를 보여줍니다.

```
version: 1.1
tasks:
- task: executeScript
  inputs:
  - frequency: always
    type: powershell
    runAs: localSystem
    content: |-
```

```
      New-Item -Path 'C:\PowerShellTest.txt' -ItemType File
```

이전 버전의 시작 에이전트와 호환되는 사용자 데이터에 XML 형식을 사용할 수 있습니다. EC2Launch v2는 스크립트를 UserData 단계에서 executeScript 태스크로 실행합니다. EC2Launch v1 및 EC2Config 동작을 준수하기 위해 사용자 데이터 스크립트는 기본적으로 첨부/인라인 프로세스로 실행됩니다.

스크립트 실행 방식을 사용자 지정하기 위해 선택적 태그를 추가할 수 있습니다. 예를 들어 인스턴스가 시작될 때와 인스턴스가 재부팅될 때 사용자 데이터 스크립트를 실행하려면 다음 태그를 사용할 수 있습니다.

<persist>true</persist>

예시:

```
<powershell>
  $file = $env:SystemRoot + "\Temp" + (Get-Date).ToString("MM-dd-yy-hh-mm")
  New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

<powershellArguments> 태그를 사용하여 하나 이상의 PowerShell 인수를 지정할 수 있습니다. 인수가 전달되지 않은 경우 EC2Launch v2는 기본적으로 -ExecutionPolicy Unrestricted 인수를 추가합니다.

예시:

```
<powershell>
  $file = $env:SystemRoot + "\Temp" + (Get-Date).ToString("MM-dd-yy-hh-mm")
  New-Item $file -ItemType file
</powershell>
<powershellArguments>-ExecutionPolicy Unrestricted -NoProfile -NonInteractive</
powershellArguments>
```

XML 사용자 데이터 스크립트를 분리된 프로세스로 실행하려면 사용자 데이터에 다음 태그를 추가하세요.

<detach>true</detach>

예시:

```
<powershell>
  $file = $env:SystemRoot + "\Temp" + (Get-Date).ToString("MM-dd-yy-hh-mm")
  New-Item $file -ItemType file
</powershell>
<detach>true</detach>
```

> **Note**
>
> 분리 태그는 이전 시작 에이전트에서 지원되지 않습니다.

변경 로그: 사용자 데이터

다음 표에는 사용자 데이터에 대한 변경 사항이 나열되어 있으며 해당 변경 사항이 적용되는 EC2Launch v2 에이전트 버전과 상호 참조합니다.

사용자 데이터 버전	세부 사항	소개
1.1	• 사용자 데이터 작업은 에이전트 구성 파일의 `PostReady` 단계 이전에 실행됩니다. • Systems Manager Agent를 시작하기 전에 사용자 데이터를 실행합니다(EC2Launch v1 및 EC2Config와 동일한 동작).*	EC2Launch v2 버전 2.0.1245
1.0	• 사용 중지됩니다. • 사용자 데이터 작업은 에이전트 구성 파일의 `PostReady` 단계 이후에 실행됩니다. 이것은 EC2Launch v1과 하위 호환되지 않습니다. • Systems Manager Agent 시작 및 사용자 데이터 작업 간의 경합 상태에 영향을 받았습니다.	EC2Launch v2 버전 2.0.0

* 기본 `agent-config.yml` 파일과 함께 사용하는 경우.

EC2Launch v2 종료 코드 및 재부팅

EC2Launch v2를 사용하여 스크립트에서 종료 코드를 처리하는 방법을 정의할 수 있습니다. 기본적으로 스크립트에서 실행된 마지막 명령의 종료 코드는 전체 스크립트의 종료 코드로 보고됩니다. 예를 들어 스크립트에 3개의 명령이 포함되어 있는 경우 첫 번째 명령이 실패하고 다음 명령이 성공하면 최종 명령이 성공했기 때문에 실행 상태가 `success`로 보고됩니다.

스크립트로 인스턴스를 재부팅하려면 스크립트의 마지막 단계가 재부팅인 경우에도 스크립트에 `exit 3010`을 지정해야 합니다. `exit 3010`은 EC2Launch v2에 `3010`이 아닌 종료 코드가 반환되거나 최대 재부팅 횟수에 도달할 때까지 인스턴스를 재부팅하고 스크립트를 다시 호출하도록 지시합니다. EC2Launch v2는 태스크당 최대 5번의 재부팅을 허용합니다. `Restart-Computer`와 같은 다른 메커니즘을 사용하여 스크립트에서 인스턴스를 재부팅하려고 하면 스크립트 실행 상태가 일관되지 않습니다. 예를 들어 다시 시작 루프에서 멈추거나 다시 시작을 수행하지 않을 수 있습니다.

이전 에이전트와 호환되는 XML 사용자 데이터 형식을 사용하는 경우 사용자 데이터가 의도한 것보다 더 많이 실행될 수 있습니다. 자세한 내용은 문제 해결 섹션의 서비스가 사용자 데이터를 두 번 이상 실행함 섹션을 참조하세요.

EC2Launch v2 및 Sysprep

EC2Launch v2 서비스에서는 재사용할 수 있는 사용자 정의된 Windows AMI를 생성하는 데 사용할 수 있는 Microsoft 도구인 Sysprep을 실행합니다. EC2Launch v2가 Sysprep을 호출하면 Sysprep은 `%ProgramData%\Amazon\EC2Launch` 안의 파일을 사용하여 어느 작업을 수행할지 결정합니다. 이러한 파일은 EC2Launch 설정 대화 상자를 사용하여 간접적으로 편집하거나, YAML 편집기 또는 텍스트 편집기를 사용하여 직접 편집할 수 있습니다. 그러나 EC2Launch 설정 대화 상자에서 사용할 수 없는 몇 가지 고급 설정이 있으며, 이러한 항목은 직접 편집해야 합니다.

설정을 업데이트한 후에 인스턴스에서 AMI를 생성하면 새로운 설정은 그 새로운 AMI에서 실행하는 모든 인스턴스에 적용됩니다. AMI 생성에 대한 자세한 내용은 Amazon EBS 지원 AMI 생성 섹션을 참조하세요.

EC2Launch v2 시작 태스크에 대한 태스크 정의

EC2Launch v2가 시작 중 실행하는 태스크마다 자체 속성과 요구 사항 세트가 있습니다. 태스크 세부 정보에는 태스크가 실행되는 빈도(한 번 또는 항상), 에이전트 부팅 프로세스의 어느 단계에서 태스크가 실행되는지, 구문 및 YAML 문서 예에 대한 설정이 포함됩니다. 자세한 내용은 이 참조에 표시된 태스크 세부 정보를 검토하세요.

EC2Launch v2 태스크

- activateWindows
- enableJumboFrames
- enableOpenSsh
- executeProgram
- executeScript
- extendRootPartition
- initializeVolume
- optimizeEna
- setAdminAccount
- setDnsSuffix
- setHostName
- setWallpaper
- startSsm
- sysprep
- writeFile

activateWindows

AWS KMS 서버 집합에 대해 Windows를 활성화합니다. 인스턴스가 기존 보유 라이선스 사용(BYOL)으로 감지되면 활성화를 건너뜁니다.

Frequency - once

AllowedStages - `[PreReady]`

Inputs -

`activation`: (맵)

`type`: (문자열) 사용할 활성화 유형, 으로 설정`amazon`

예

```
task: activateWindows
```

```
  inputs:
    activation:
    type: amazon
```

enableJumboFrames

네트워크 어댑터의 최대 전송 단위(MTU)를 늘리는 점보 프레임을 활성화합니다. 자세한 내용은 점보 프레임(9001 MTU) 섹션을 참조하세요.

Frequency — always

AllowedStages - `[PostReady, UserData]`

Inputs - none

예

```
task: enableJumboFrames
```

enableOpenSsh

Windows OpenSSH를 활성화하고 인스턴스의 퍼블릭 키를 인증된 키 폴더에 추가합니다.

Frequency - once

AllowedStages - `[PreReady, UserData]`

Inputs - none

예

다음 예에서는 인스턴스에서 OpenSSH를 활성화하고 인스턴스의 퍼블릭 키를 인증된 키 폴더에 추가하는 방법을 보여줍니다. 이 구성은 Windows Server 2019 이상 버전을 실행하는 인스턴스에서만 작동합니다.

```
task: enableOpenSsh
```

executeProgram

선택적 인수와 지정된 빈도를 사용하여 프로그램을 실행합니다.

단계: PreReady, PostReady, UserData 단계에서 executeProgram 작업을 실행할 수 있습니다.

주파수: 구성 가능, 입력 참조.

입력

이 섹션에는 executeProgram 태스크가 실행할 하나 이상의 프로그램이 들어 있습니다(입력). 각 입력에는 다음과 같은 구성 가능한 속성이 포함될 수 있습니다.

주파수(문자열)

(필수) 다음 값 중 정확히 하나를 지정할 수 있습니다.

- once
- always

경로(문자열)

(필수) 실행할 실행 파일 경로입니다.

인수(문자열 목록)

(선택 사항) 프로그램에 입력으로 제공할 쉼표로 구분된 인수 목록입니다.

runAs(문자열)

(필수) localSystem로 설정해야 합니다.

출력

모든 작업은 로그 파일 항목을 agent.log 파일에 기록합니다. executeProgram작업의 추가 출력은 다음과 같이 동적으로 이름이 지정된 폴더에 별도로 저장됩니다.

%LocalAppData%\Temp*EC2Launch#########**outputfilename.tmp*

출력 파일의 정확한 경로가 agent.log 파일에 포함됩니다. 예를 들면 다음과 같습니다.

```
Program file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\ExecuteProgramInputs.tmp
Output file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\Output.tmp
Error file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\Err.tmp
```

executeProgram 작업의 출력 파일

ExecuteProgramInputs.tmp

실행 파일의 경로와 executeProgram 작업이 실행될 때 전달되는 모든 입력 매개 변수를 포함합니다.

Output.tmp

executeProgram 작업이 실행되는 프로그램의 런타임 출력을 포함합니다.

Err.tmp

executeProgram 작업이 실행되는 프로그램의 런타임 오류 메시지를 포함합니다.

예시

다음 예는 executeProgram 작업이 있는 인스턴스의 로컬 디렉터리에서 실행 파일을 실행하는 방법을 보여줍니다.

예 1: 한 인수로 실행 파일 설정

이 예에서는 설치 실행 파일을 자동 모드로 실행하는 executeProgram 작업을 보여줍니다.

```
task: executeProgram
  inputs:
    - frequency: always
      path: C:\Users\Administrator\Desktop\setup.exe
      arguments: ['-quiet']
```

예 2: 두 인수를 사용한 VLC 실행 파일

이 예에서는 입력 매개 변수로 전달된 두 인수를 사용하여 VLC 실행 파일을 실행하는 executeProgram 작업을 보여줍니다.

```
task: executeProgram
  inputs:
    - frequency: always
      path: C:\vlc-3.0.11-win64.exe
      arguments: ['/L=1033','/S']
      runAs: localSystem
```

executeScript

선택적 인수와 지정된 빈도를 사용하여 스크립트를 실행합니다. 스크립트 동작은 에이전트가 스크립트를 실행하는 모드 (인라인 모드 또는 분리 모드)에 따라 달라집니다.

인라인(기본값)

EC2Launch v2 에이전트는 한 번에 하나씩(`detach: false`) 스크립트를 실행합니다. 이것이 기본 설정입니다.

> **Note**
>
> 인라인 스크립트가 reset 또는 sysprep 명령을 내리면 해당 명령이 즉시 실행되고 에이전트가 재설정됩니다. 현재 태스크가 끝나면 에이전트가 추가 태스크를 실행하지 않고 종료됩니다.
> 예를 들어 명령을 실행하는 태스크 뒤에 `startSsm` 태스크(사용자 데이터 실행 후 기본적으로 포함됨)가 뒤따르는 경우 해당 태스크가 실행되지 않고 Systems Manager 서비스도 시작되지 않습니다.

분리됨

EC2Launch v2 에이전트는 다른 태스크(`detach: true`)와 동시에 스크립트를 실행합니다.

> **Note**
>
> 분리된 스크립트가 reset 또는 sysprep 명령을 내리면 해당 명령은 실행되기 전에 에이전트가 완료될 때까지 기다립니다. ExecuteScript 이후의 태스크는 계속 실행됩니다.

단계: `PreReady`, `PostReady`, `UserData` 단계에서 `executeScript` 작업을 실행할 수 있습니다.

주파수: 구성 가능, 입력 참조.

입력

이 섹션에는 executeScript 태스크가 실행할 하나 이상의 스크립트가 들어 있습니다(입력). 각 입력에는 다음과 같은 구성 가능한 속성이 포함될 수 있습니다.

주파수(문자열)

(필수) 다음 값 중 정확히 하나를 지정할 수 있습니다.

- once
- always

유형(문자열)

(필수) 다음 값 중 정확히 하나를 지정할 수 있습니다.

- batch
- powershell

인수(문자열 목록)

(선택 사항) 쉘에 전달할 문자열 인수 목록입니다(PowerShell 스크립트에 전달하지 않음). type: batch에는 이 매개변수가 지원되지 않습니다. 인수가 전달되지 않은 경우 EC2Launch v2는 기본적으로 -ExecutionPolicy Unrestricted 인수를 추가합니다.

내용(문자열)

(필수) 스크립트 콘텐츠.

runAs(문자열)

(필수) 다음 값 중 정확히 하나를 지정할 수 있습니다.

- admin
- localSystem

분리(부울)

(선택 사항) EC2Launch v2 에이전트는 기본적으로 스크립트를 한 번에 하나씩(detach: false) 실행합니다. 스크립트를 다른 작업과 동시에 실행하려면 값을 true(detach: true)로 설정합니다.

> **Note**
>
> 스크립트 종료 코드(3010 포함)는 detach가 true로 설정되면 아무런 효과가 없습니다.

출력

모든 작업은 로그 파일 항목을 agent.log 파일에 기록합니다. executeScript 작업이 실행되는 스크립트의 추가 출력은 다음과 같이 동적으로 이름이 지정된 폴더에 별도로 저장됩니다.

%LocalAppData%\Temp*EC2Launch#########**outputfilename.ext*

출력 파일의 정확한 경로가 agent.log 파일에 포함됩니다. 예를 들면 다음과 같습니다.

```
Program file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\UserScript.ps1
Output file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\Output.tmp
Error file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\Err.tmp
```

executeScript 작업의 출력 파일

UserScript.*ext*

executeScript 작업이 실행한 스크립트를 포함합니다. 파일 확장자는 다음과 같이 executeScript 태스크의 type 파라미터에 지정한 스크립트 유형에 따라 달라집니다.

- 유형이 batch인 경우 파일 확장자는 .bat입니다.
- 유형이 powershell인 경우 파일 확장자는 .ps1입니다.

Output.tmp

executeScript 작업이 실행하는 스크립트의 런타임 출력을 포함합니다.

Err.tmp

executeScript 작업이 실행하는 스크립트의 런타임 오류 메시지를 포함합니다.

예시

다음 예에서는 executeScript 작업에서 인라인 스크립트를 실행하는 방법을 보여줍니다.

예 1: Hello world 출력 텍스트 파일

이 예에서는 PowerShell 스크립트를 실행하여 C: 드라이브에 “Hello world” 텍스트 파일을 만드는 executeScript 작업을 보여줍니다.

```
task: executeScript
  inputs:
    - frequency: always
      type: powershell
      runAs: admin
      content: |-
        New-Item -Path 'C:\PowerShellTest.txt' -ItemType File
        Set-Content 'C:\PowerShellTest.txt' "Hello world"
```

예 2: 두 스크립트 실행

이 예에서는 executeScript 작업이 둘 이상의 스크립트를 실행할 수 있으며 스크립트 유형이 반드시 일치할 필요는 없음을 보여줍니다.

첫 번째 스크립트(type: powershell)는 현재 인스턴스에서 실행 중인 프로세스의 요약을 C: 드라이브에 있는 텍스트 파일에 기록합니다.

두 번째 스크립트(batch)는 시스템 정보를 Output.tmp 파일에 기록합니다.

```
task: executeScript
  inputs:
    - frequency: always
      type: powershell
      runAs: localSystem
      content: |
        Get-Process | Out-File -FilePath C:\Process.txt
    - frequency: always
      type: batch
      runAs: localSystem
      content: |
        systeminfo
```

예 3: 재부팅을 통한 멱등성 시스템 구성

이 예에서는 멱등성 스크립트 실행을 통해 각 단계 사이에 재부팅하여 다음 시스템 구성을 수행하는 executeScript 작업을 보여 줍니다.

- 컴퓨터 이름을 변경합니다.
- 도메인에 컴퓨터를 결합합니다.
- Telnet을 활성화합니다.

스크립트는 각 작업이 한 번만 실행되도록 합니다. 이렇게 하면 재부팅 루프가 방지되고 스크립트가 멱등성을 갖습니다.

```
task: executeScript
  inputs:
    - frequency: always
      type: powershell
      runAs: localSystem
      content: |-
        $name = $env:ComputerName
```

```
        if ($name -ne $desiredName) {
          Rename-Computer -NewName $desiredName
          exit 3010
        }
        $domain = Get-ADDomain
        if ($domain -ne $desiredDomain)
        {
          Add-Computer -DomainName $desiredDomain
          exit 3010
        }
        $telnet = Get-WindowsFeature -Name Telnet-Client
        if (-not $telnet.Installed)
        {
          Install-WindowsFeature -Name "Telnet-Client"
          exit 3010
        }
```

extendRootPartition

루트 볼륨을 확장하여 디스크에서 사용 가능한 공간을 모두 사용합니다.

Frequency - once

AllowedStages - `[Boot]`

Inputs - none

예

```
task: extendRootPartition
```

initializeVolume

인스턴스에 연결된 빈 볼륨을 활성화 및 분할되도록 초기화합니다. 시작 에이전트에서는 볼륨이 비어 있지 않은 것이 감지되면 초기화를 건너뜁니다. 볼륨의 처음 4KiB가 비어 있거나 볼륨에 Windows에서 인식 가능한 드라이브 레이아웃이 없으면 볼륨이 비어 있는 것으로 간주합니다.

`letter` 입력 파라미터는 드라이브가 이미 초기화되었는지 여부와 관계없이 이 태스크를 실행할 때 항상 적용됩니다.

`initializeVolume`에서는 다음과 같은 작업을 수행합니다.

- 디스크의 `offline` 및 `readonly` 속성을 false로 설정합니다.
- 파티션을 생성합니다. `partition` 입력 파라미터에 파티션 유형이 지정되지 않으면 다음과 같은 기본값이 적용됩니다.
  - 디스크 크기가 2TB 미만이면 파티션 유형을 `mbr`로 설정합니다.
  - 디스크 크기가 2TB 이상이면 파티션 유형을 `gpt`로 설정합니다.
- 볼륨 형식을 NTFS로 지정합니다.
- 볼륨 레이블을 다음과 같이 설정합니다.
  - `name` 입력 파라미터의 값을 사용합니다(지정된 경우).
  - 임시 볼륨이고 지정된 이름이 없으면 볼륨 레이블을 `Temporary Storage Z`로 설정합니다.
- 임시 볼륨(Amazon EBS가 아닌 SSD 또는 HDD)이면 다음과 같은 내용으로 볼륨 루트에 `Important.txt` 파일을 생성합니다.

```
This is an 'Instance Store' disk and is provided at no additional charge.

*This disk offers increased performance since it is local to the host
*The number of Instance Store disks available to an instance vary by instance type
*DATA ON THIS DRIVE WILL BE LOST IN CASES OF IMPAIRMENT OR STOPPING THE INSTANCE.
 PLEASE ENSURE THAT ANY IMPORTANT DATA IS BACKED UP FREQUENTLY

For more information, please refer to: EC2 ##### #### ### ## ## ####.
```

- 드라이브 문자를 `letter` 입력 파라미터에 지정된 값으로 설정합니다.

스테이지: `PostReady` 및 `UserData` 스테이지 동안 `initializeVolume` 태스크를 실행할 수 있습니다.

빈도: 항상입니다.

입력

다음과 같이 런타임 매개변수를 구성할 수 있습니다.

디바이스(맵 목록)

(조건부) 시작 에이전트에서 초기화하는 각 디바이스의 구성입니다. `initialize` 입력 파라미터가 `devices`로 설정되는 경우에 필요합니다.

- 디바이스(문자열, 필수 사항) – 인스턴스 생성 중 디바이스를 식별합니다. 예: `xvdb`, `xvdf` 또는 `\dev\nvme0n1`.

- 문자(문자열, 선택 사항) – 1자입니다. 할당할 드라이브 문자입니다.
- 이름(문자열, 선택 사항) - 할당할 볼륨 이름입니다.
- 파티션(문자열, 선택 사항) - 생성할 파티션 유형에 다음과 같은 값 중 하나를 지정하거나 볼륨 크기에 따라 시작 에이전트에서 기본값을 설정하도록 합니다.
  - mbr
  - gpt

초기화(문자열)

(필수) 다음 값 중 정확히 하나를 지정할 수 있습니다.

- `all`
- `devices`

예시

다음 예에서는 `initializeVolume` 태스크의 샘플 입력 구성을 보여줍니다.

예 1: 인스턴스에 대한 2개 볼륨 초기화

이 예에서는 인스턴스에 대한 2개의 보조 볼륨을 초기화하는 `initializeVolume` 태스크를 보여줍니다. 예에서 이름이 `DataVolume2`인 디바이스는 임시 디바이스입니다.

```
task: initializeVolume
inputs:
  initialize: devices
  devices:
  - device: xvdb
    name: DataVolume1
    letter: D
    partition: mbr
  - device: /dev/nvme0n1
    name: DataVolume2
    letter: E
    partition: gpt
```

예 2: 인스턴스에 연결된 EBS 볼륨 초기화

이 예에서는 인스턴스에 연결된 모든 빈 EBS 볼륨을 초기화하는 `initializeVolume` 태스크를 보여줍니다.

```
task: initializeVolume
inputs:
  initialize: all
```

optimizeEna

현재 인스턴스 유형에 따라 ENA 설정을 최적화합니다. 인스턴스를 재부팅할 수 있습니다.

Frequency — always

AllowedStages - [PostReady, UserData]

Inputs - none

예

```
task: optimizeEna
```

setAdminAccount

로컬 시스템에 생성된 기본 관리자 계정의 속성을 설정합니다.

Frequency - once

AllowedStages - [PreReady]

Inputs -

name: (문자열) 관리자 계정의 이름

password: (맵)

type: (문자열) 암호 설정 전략. static, random 또는 doNothing

data: (문자열) type 필드가 정적인 경우 데이터 저장

예

```
task: setAdminAccount
inputs:
  name: Administrator
```

```
  password:
  type: random
```

setDnsSuffix

검색 접미사 목록에 DNS 접미사를 추가합니다. 아직 존재하지 않는 접미사만 목록에 추가됩니다. 시작 에이전트가 DNS 접미사를 설정하는 방법에 대한 자세한 내용은 EC2 Windows 시작 에이전트용 DNS 접미사 구성 섹션을 참조하세요.

Frequency — always

AllowedStages - `[PreReady]`

Inputs -

`suffixes`: (문자열 목록) 하나 이상의 유효한 DNS 접미사 목록. 유효한 대체 변수는 `$REGION` 및 `$AZ`

예

```
task: setDnsSuffix
inputs:
  suffixes:
  - $REGION.ec2-utilities.amazonaws.com
```

setHostName

컴퓨터의 호스트 이름을 사용자 지정 문자열, 또는 `hostName`이 지정되지 않은 경우 프라이빗 IPv4 주소로 설정합니다.

Frequency — always

AllowedStages - `[PostReady, UserData]`

Inputs -

`hostName`: (문자열) 선택적 호스트 이름입니다. 이 이름은 다음과 같은 형식이어야 합니다.

- 15자 이하여야 합니다.
- 영숫자(a-z, A-Z, 0-9) 및 하이픈(-) 문자만 포함해야 합니다.
- 전체가 숫자 문자로 구성되어서는 안 됩니다.

reboot: (부울) 호스트 이름이 변경될 때 재부팅이 허용되는지 여부를 나타냅니다.

예

```
task: setHostName
inputs:
  reboot: true
```

setWallpaper

setwallpaper.lnk를 제외한 각 기존 사용자의 시작 폴더에 Default User 바로 가기 파일을 생성합니다. 이 바로 가기 파일은 사용자가 인스턴스를 부팅한 후 처음으로 로그인할 때 실행됩니다. 이 파일은 인스턴스 속성을 표시하는 사용자 지정 월페이퍼를 사용해 인스턴스를 설정합니다.

바로 가기 파일 경로는 다음과 같습니다.

```
$env:SystemDrive/Users/<user>/AppData/Roaming/Microsoft/Windows/Start Menu/Programs/
Startup/setwallpaper.lnk
```

> **Note**
>
> setWallpaper 태스크를 제거해도 이 바로 가기 파일은 삭제되지 않습니다. 자세한 내용은 setWallpaper 태스크를 사용하지 않지만 재부팅 시 월페이퍼 재설정 단원을 참조하십시오.

스테이지: PreReady 및 UserData 스테이지 중에 배경화면을 구성할 수 있습니다.

빈도: always

월페이퍼 구성

다음 설정을 사용하여 월페이퍼를 구성할 수 있습니다.

입력

제공하는 입력 파라미터와 월페이퍼를 구성하기 위해 설정할 수 있는 속성:

속성(문자열 목록)

(선택 사항) 월페이퍼에 다음 속성 중 하나 이상을 추가할 수 있습니다.

- architecture

- availabilityZone
- hostName
- instanceId
- instanceSize
- memory
- network
- privateIpAddress
- publicIpAddress

instanceTags

(선택 사항) 이 설정에 대해 다음 옵션 중 정확히 하나를 사용할 수 있습니다.

- AllTags(문자열) - 월페이퍼에 모든 인스턴스 태그를 추가합니다.

```
instanceTags: AllTags
```

- instanceTags(문자열 목록) – 월페이퍼에 추가할 인스턴스 태그 이름의 목록을 지정합니다. 예시:

```
instanceTags:
  - Tag 1
  - Tag 2
```

경로(문자열)

(필수) 월페이퍼 이미지에 사용할 로컬.jpg 형식 이미지 파일의 파일 이름 경로입니다.

예제

다음 예제에서는 Tag 1 및 Tag 2라는 인스턴스 태그와 함께 월페이퍼 배경 이미지의 파일 경로를 설정하는 월페이퍼 구성 입력과 인스턴스의 호스트 이름, 인스턴스 ID, 프라이빗 및 퍼블릭 IP 주소를 포함하는 속성을 보여줍니다.

```
task: setWallpaper
inputs:
  path: C:\ProgramData\Amazon\EC2Launch\wallpaper\Ec2Wallpaper.jpg
  attributes:
  - hostName
  - instanceId
```

```
  - privateIpAddress
  - publicIpAddress
instanceTags:
  - Tag 1
  - Tag 2
```

> **Note**
>
> 웜페이퍼에 태그를 표시하려면 메타데이터에서 태그를 활성화해야 합니다. 인스턴스 태그 및 메타데이터에 대한 자세한 내용은 인스턴스 메타데이터를 사용하여 EC2 인스턴스의 태그 보기 섹션을 참조하세요.

startSsm

Sysprep 이후에 Systems Manager(SSM) 서비스를 시작합니다.

Frequency — always

AllowedStages - `[PostReady, UserData]`

Inputs - none

예

```
task: startSsm
```

sysprep

서비스 상태를 재설정하고, `unattend.xml`을 업데이트하고, RDP를 비활성화하고, Sysprep을 실행합니다. 이 태스크는 다른 모든 태스크가 완료된 후에만 실행됩니다.

Frequency - once

AllowedStages - `[UserData]`

Inputs -

`clean`: (부울) Sysprep을 실행하기 전에 인스턴스 로그 정리

`shutdown`: (부울) Sysprep을 실행 한 후 인스턴스 종료

예

```
task: sysprep
inputs:
clean: true
shutdown: true
```

writeFile

대상에 파일을 씁니다.

Frequency - Inputs 참조

AllowedStages - [PostReady, UserData]

Inputs -

frequency: (문자열) once 또는 always 중 하나

destination: (문자열) 콘텐츠를 쓸 경로

content: (문자열) 대상에 쓸 텍스트

예제

```
task: writeFile
inputs:
  - frequency: once
  destination: C:\Users\Administrator\Desktop\booted.txt
  content: Windows Has Booted
```

EC2Launch v2 에이전트와 관련된 문제 해결

이 섹션에서는 EC2Launch v2에 대한 일반적인 문제 해결 시나리오, Windows 이벤트 로그 보기에 대한 정보 및 콘솔 로그 출력과 메시지를 보여줍니다.

주제 문제 해결

- 일반적인 문제 해결 시나리오
- Windows 이벤트 로그
- EC2Launch v2 콘솔 로그 출력

일반적인 문제 해결 시나리오

이 섹션에서는 일반적인 문제 해결 시나리오와 해결 단계를 보여 줍니다.

시나리오

- 서비스에서 월페이퍼를 설정하지 못함
- 서비스에서 사용자 데이터를 실행하지 못함
- 서비스가 태스크를 한 번만 실행함
- 서비스에서 작업을 실행하지 못함
- 서비스가 사용자 데이터를 두 번 이상 실행함
- EC2Launch v2로 마이그레이션한 후 EC2Launch v1의 예약된 태스크가 실행되지 않음
- 서비스가 비어 있지 않은 EBS 볼륨을 초기화함
- setWallpaper 태스크를 사용하지 않지만 재부팅 시 월페이퍼 재설정
- 서비스가 실행 중 상태에서 멈춤
- 잘못된 agent-config.yml 때문에 EC2Launch v2 설정 대화 상자가 열리지 않음
- task:executeScript should be unique and only invoked once

서비스에서 월페이퍼를 설정하지 못함

해결 방법

1. `%AppData%\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\setwallpaper.lnk`가 있는지 확인합니다.
2. `%ProgramData%\Amazon\EC2Launch\log\agent.log`를 검토해 오류가 발생했는지 확인합니다.

서비스에서 사용자 데이터를 실행하지 못함

가능한 원인: 사용자 데이터를 실행하기 전에 서비스가 실패했을 수 있습니다.

해결 방법

1. `%ProgramData%\Amazon\EC2Launch\state\previous-state.json`을 검토합니다.
2. `boot`, `network`, `preReady` 및 `postReadyLocalData`가 모두 성공으로 표시되었는지 확인합니다.

3. 스테이지 중 하나가 실패한 경우 `%ProgramData%\Amazon\EC2Launch\log\agent.log`에서 구체적인 오류를 확인합니다.

서비스가 태스크를 한 번만 실행함

해결 방법

1. 작업 빈도를 확인합니다.
2. 서비스가 Sysprep 이후에 이미 실행되었고 작업 빈도가 `once`로 설정된 경우 작업이 다시 실행되지 않습니다.
3. EC2Launch v2가 실행될 때마다 태스크를 실행하려는 경우 태스크 빈도를 `always`로 설정합니다.

서비스에서 작업을 실행하지 못함

해결 방법

1. `%ProgramData%\Amazon\EC2Launch\log\agent.log`에서 최근 항목을 확인합니다.
2. 오류가 발생하지 않은 경우 `"%ProgramFiles%\Amazon\EC2Launch\EC2Launch.exe" run`에서 수동으로 서비스를 실행하여 작업이 성공했는지 확인합니다.

서비스가 사용자 데이터를 두 번 이상 실행함

해결 방법

사용자 데이터는 EC2Launch v1과 EC2Launch v2에서 다르게 처리됩니다. `persist`가 `true`로 설정된 경우 EC2Launch v1은 인스턴스에서 예약 작업으로 사용자 데이터를 실행합니다. `persist`가 `false`로 설정된 경우에는 재부팅 후 종료되거나 실행 중에 중단된 경우에도 태스크가 예약되지 않습니다.

EC2Launch v2는 사용자 데이터를 에이전트 태스크로 실행하고 실행 상태를 추적합니다. 사용자 데이터가 컴퓨터 다시 시작을 실행하거나 사용자 데이터가 실행 중에 중단된 경우 실행 상태는 `pending`으로 유지되고 다음 인스턴스 부팅 시 사용자 데이터가 다시 실행됩니다. 사용자 데이터 스크립트가 두 번 이상 실행되지 않도록 하려면 스크립트를 멱등성으로 만듭니다.

다음 예제 멱등성 스크립트는 컴퓨터 이름을 설정하고 도메인에 연결합니다.

```
<powershell>
  $name = $env:computername
  if ($name -ne $desiredName) {
```

```
 Rename-Computer -NewName $desiredName
  }
  $domain = Get-ADDomain
  if ($domain -ne $desiredDomain)
  {
 Add-Computer -DomainName $desiredDomain
  }
  $telnet = Get-WindowsFeature -Name Telnet-Client
  if (-not $telnet.Installed)
  {
 Install-WindowsFeature -Name "Telnet-Client"
  }
</powershell>
<persist>false</persist>
```

EC2Launch v2로 마이그레이션한 후 EC2Launch v1의 예약된 태스크가 실행되지 않음

해결 방법

마이그레이션 도구는 EC2Launch v1 스크립트에 연결된 예약 태스크를 검색하지 않으므로 EC2Launch v2에서 이러한 태스크를 자동으로 설정하지 않습니다. 이러한 태스크를 구성하려면 `agent-config.yml` 파일을 편집하거나 EC2Launch v2 설정 대화 상자를 사용합니다. 예를 들어 인스턴스에 `InitializeDisks.ps1`을 실행하는 예약 태스크가 있는 경우 마이그레이션 도구를 실행한 후 EC2Launch v2 설정 대화 상자에서 초기화할 볼륨을 지정해야 합니다. EC2Launch v2 설정 대화 상자를 사용하여 설정 변경에 대한 절차의 6단계를 참조하세요.

서비스가 비어 있지 않은 EBS 볼륨을 초기화함

해결 방법

EC2Launch v2는 볼륨을 초기화하기 전에 볼륨이 비어 있는지 여부를 감지합니다. 볼륨이 비어 있지 않으면 초기화를 건너뜁니다. 비어 있지 않은 것으로 감지된 볼륨은 초기화되지 않습니다. 볼륨의 처음 4KiB가 비어 있거나 볼륨에 Windows 인식 가능한 드라이브 레이아웃이 없는 경우 볼륨은 비어 있는 것으로 간주됩니다. Linux 시스템에서 초기화되고 포맷된 볼륨에는 MBR 또는 GPT와 같은 Windows 인식 가능한 드라이브 레이아웃이 없습니다. 따라서 비어 있고 초기화된 것으로 간주됩니다. 이 데이터를 보존하려면 EC2Launch v2 빈 드라이브 감지를 사용하지 마세요. 대신 EC2Launch v2 설정 대화 상자(6단계 참조) 또는 `agent-config.yml`에서 초기화할 볼륨을 지정합니다.

`setWallpaper` 태스크를 사용하지 않지만 재부팅 시 월페이퍼 재설정

`setWallpaper` 태스크는 `Default User`를 제외한 각 기존 사용자의 시작 폴더에 `setwallpaper.lnk` 바로 가기 파일을 생성합니다. 이 바로 가기 파일은 사용자가 인스턴스를 부팅한

후 처음으로 로그인할 때 실행됩니다. 이 파일은 인스턴스 속성을 표시하는 사용자 지정 월페이퍼를 사용해 인스턴스를 설정합니다. setWallpaper 태스크를 제거하더라도 이 바로 가기 파일은 삭제되지 않습니다. 이 파일을 수동으로 삭제하거나 스크립트를 사용하여 삭제해야 합니다.

바로 가기 경로는 다음과 같습니다.

$env:SystemDrive/Users/<user>/AppData/Roaming/Microsoft/Windows/Start Menu/Programs/Startup/setwallpaper.lnk

해결 방법

이 파일을 수동으로 삭제하거나 스크립트를 사용하여 삭제합니다.

바로 가기 파일을 삭제하는 PowerShell 스크립트 예

```
foreach ($userDir in (Get-ChildItem "C:\Users" -Force -Directory).FullName)
{
 $startupPath = Join-Path $userDir -ChildPath "AppData\Roaming\Microsoft\Windows\Start
 Menu\Programs\Startup"
 if (Test-Path $startupPath)
 {
  $wallpaperSetupPath = Join-Path $startupPath -ChildPath "setwallpaper.lnk"
  if (Test-Path $wallpaperSetupPath)
  {
   Remove-Item $wallpaperSetupPath -Force -Confirm:$false
  }
 }
}
```

서비스가 실행 중 상태에서 멈춤

설명

EC2Launch v2는 다음과 유사한 로그 메시지(agent.log)로 차단됩니다.

```
2022-02-24 08:08:58 Info:
 ********************************************************************************
2022-02-24 08:08:58 Info: EC2Launch Service starting
2022-02-24 08:08:58 Info: Windows event custom log exists: Amazon EC2Launch
2022-02-24 08:08:58 Info: ACPI SPCR table not supported. Bailing Out
2022-02-24 08:08:58 Info: Serial port is in use. Waiting for Serial Port...
2022-02-24 08:09:00 Info: ACPI SPCR table not supported. Use default console port.
2022-02-24 08:09:02 Info: ACPI SPCR table not supported. Use default console port.
```

```
2022-02-24 08:09:04 Info: ACPI SPCR table not supported. Use default console port.
2022-02-24 08:09:06 Info: ACPI SPCR table not supported. Use default console port.
```

가능한 원인

SAC가 사용 설정되어 직렬 포트를 사용합니다. 자세한 내용은 SAC를 사용하여 Windows 인스턴스 문제 해결을 참조하세요.

해결 방법

다음 단계에 따라 문제를 해결하세요.

- 직렬 포트를 사용 중인 서비스를 사용 중지합니다.
- 서비스에서 직렬 포트를 계속 사용하려면 사용자 지정 스크립트를 작성하여 시작 에이전트 작업을 수행하고 예약된 작업으로 호출합니다.

잘못된 **agent-config.yml** 때문에 EC2Launch v2 설정 대화 상자가 열리지 않음

설명

EC2Launch v2 설정은 대화 상자를 열기 전에 `agent-config.yml` 파일 구문 분석을 시도합니다. YAML 구성 파일이 지원되는 스키마를 따르지 않으면 대화 상자에 다음 오류가 표시됩니다.

```
Unable to parse configuration file agent-config.yml. Review configuration file. Exiting application.
```

해결 방법

1. 구성 파일이 지원되는 스키마를 따르는지 확인합니다.
2. 처음부터 시작하려면 기본 구성 파일을 `agent-config.yml`에 복사하세요. 작업 구성 섹션의 `agent-config.yml` 예시를 사용할 수 있습니다.
3. `agent-config.yml`을 삭제하여 다시 시작할 수도 있습니다. EC2Launch v2 설정은 빈 구성 파일을 생성합니다.

task:executeScript should be unique and only invoked once

설명

동일한 단계에서는 작업을 반복할 수 없습니다.

해결 방법

executeScript 및 executeProgram과 같은 일부 작업은 배열로 입력해야 합니다. 스크립트를 배열로 작성하는 방법의 예제는 executeScript를 참조하세요.

Windows 이벤트 로그

EC2Launch v2는 서비스 시작, Windows 준비, 태스크 성공 및 실패와 같은 중요한 이벤트에 대한 Windows 이벤트 로그를 게시합니다. 이벤트 식별자는 특정 이벤트를 고유하게 식별합니다. 각 이벤트에는 스테이지, 작업 및 레벨 정보와 설명이 포함됩니다. 이벤트 식별자를 사용하여 특정 이벤트에 대한 트리거를 설정할 수 있습니다.

이벤트 ID는 이벤트에 대한 정보를 제공하고 일부 이벤트를 고유하게 식별합니다. 이벤트 ID의 최하위 자릿수는 이벤트의 심각도를 나타냅니다.

Event	최하위 자릿수
Success	. . .0
Informational	. . .1
Warning	. . .2
Error	. . .3

서비스가 시작되거나 중지될 때 생성되는 서비스 관련 이벤트에는 한 자리의 이벤트 식별자가 포함됩니다.

Event	한 자리의 식별자
Success	0
Informational	1
Warning	2
Error	3

EC2LaunchService.exe 이벤트에 대한 이벤트 메시지는 Service:로 시작합니다. EC2Launch.exe 이벤트에 대한 이벤트 메시지는 Service:로 시작하지 않습니다.

네 자리의 이벤트 ID에는 이벤트의 단계, 작업 및 심각도에 대한 정보가 포함됩니다.

주제

- 이벤트 ID 형식
- 이벤트 ID 예
- Windows 이벤트 로그 스키마

이벤트 ID 형식

다음 표에는 EC2Launch v2 이벤트 식별자의 형식이 나와 있습니다.

3	2 1	0
S	T	L

표의 문자와 숫자는 다음과 같은 이벤트 유형 및 정의를 나타냅니다.

이벤트 유형	정의
S(스테이지)	0 - 서비스 수준 메시지 1 - 부팅 2 - 네트워크 3 - 사전 준비 5 - Windows 준비 6 - 사후 준비 7 - 사용자 데이터
T(작업)	해당 두 값으로 표시되는 작업은 각 스테이지마다 다릅니다. 전체 이벤트 목록을 보려면 Windows 이벤트 로그 스키마를 참조하세요.

이벤트 유형	정의
L(이벤트 수준)	0 - 성공 1 - 정보 2 - 경고 3 - 오류

이벤트 ID 예

다음은 이벤트 ID의 예입니다.

- 5000 - Windows를 사용할 준비가 됨
- 3010 - 사전 준비 스테이지에서 Windows 활성화 작업 성공
- 6013 - 사후 준비 로컬 데이터 스테이지에서 월페이퍼 설정 작업에 오류 발생

Windows 이벤트 로그 스키마

메시지 ID/이벤트 ID	이벤트 메시지
. . .0	Success
. . .1	Informational
. . .2	Warning
. . .3	Error
x	EC2Launch service-level logs
0	EC2Launch service exited successfully
1	EC2Launch service informational logs

메시지 ID/이벤트 ID	이벤트 메시지
2	EC2Launch service warning logs
3	EC2Launch service error logs
10	Replace state.json with prev ious-state.json
100	Serial Port
200	Sysprep
300	PrimaryNic
400	Metadata
x000	Stage (1 digit), Task (2 digits), Status (1 digit)
1000	Boot
1010	Boot - extend_root_partition
2000	Network
2010	Network - add_routes
3000	PreReady
3010	PreReady - activate_windows
3020	PreReady - install_egpu_manager
3030	PreReady - set_monitor_on
3040	PreReady - set_hibernation
3050	PreReady - set_admin_account
3060	PreReady - set_dns_suffix

메시지 ID/이벤트 ID	이벤트 메시지
3070	PreReady - set_wallpaper
3080	PreReady - set_update_schedule
3090	PreReady - output_log
3100	PreReady - enable_open_ssh
5000	Windows is Ready to use
6000	PostReadyLocalData
7000	PostReadyUserData
6010/7010	PostReadyLocal/UserData - set_wallpaper
6020/7020	PostReadyLocal/UserData - set_update_schedule
6030/7030	PostReadyLocal/UserData - set_hostname
6040/7040	PostReadyLocal/UserData - execute_program
6050/7050	PostReadyLocal/UserData - execute_script
6060/7060	PostReadyLocal/UserData - manage_package
6070/7070	PostReadyLocal/UserData - initialize_volume
6080/7080	PostReadyLocal/UserData - write_file

메시지 ID/이벤트 ID	이벤트 메시지
6090/7090	PostReadyLocal/UserData - start_ssm
7100	PostReadyUserData - enable_open_ssh
6110/7110	PostReadyLocal/UserData - enable_jumbo_frames

EC2Launch v2 콘솔 로그 출력

이 섹션에는 EC2Launch v2에 대한 샘플 콘솔 로그 출력이 포함되어 있으며, 문제를 해결하는 데 도움이 되는 모든 EC2Launch v2 콘솔 로그 오류 메시지가 나열됩니다. 인스턴스 콘솔 출력 및 액세스 방법에 대한 자세한 내용은 the section called “인스턴스 콘솔 출력” 섹션을 참조하세요.

출력

- EC2Launch v2 콘솔 로그 출력
- EC2Launch v2 콘솔 로그 메시지

EC2Launch v2 콘솔 로그 출력

다음은 EC2Launch v2에 대한 샘플 콘솔 로그 출력입니다.

```
2023/11/30 20:18:53Z: Windows sysprep configuration complete.
2023/11/30 20:18:57Z: Message: Waiting for access to metadata...
2023/11/30 20:18:57Z: Message: Meta-data is now available.
2023/11/30 20:18:57Z: AMI Origin Version: 2023.11.15
2023/11/30 20:18:57Z: AMI Origin Name: Windows_Server-2022-English-Full-Base
2023/11/30 20:18:58Z: OS: Microsoft Windows NT 10.0.20348
2023/11/30 20:18:58Z: OsVersion: 10.0
2023/11/30 20:18:58Z: OsProductName: Windows Server 2022 Datacenter
2023/11/30 20:18:58Z: OsBuildLabEx: 20348.1.amd64fre.fe_release.210507-1500
2023/11/30 20:18:58Z: OsCurrentBuild: 20348
2023/11/30 20:18:58Z: OsReleaseId: 2009
2023/11/30 20:18:58Z: Language: en-US
2023/11/30 20:18:58Z: TimeZone: UTC
2023/11/30 20:18:58Z: Offset: UTC +0000
```

```
2023/11/30 20:18:58Z: Launch: EC2 Launch v2.0.1643
2023/11/30 20:18:58Z: AMI-ID: ami-1234567890abcdef1
2023/11/30 20:18:58Z: Instance-ID: i-1234567890abcdef0
2023/11/30 20:18:58Z: Instance Type: c5.large
2023/11/30 20:19:00Z: Driver: AWS NVMe Driver v1.5.0.33
2023/11/30 20:19:00Z: SubComponent: AWS NVMe Driver v1.5.0.33;
 EnableSCSIPersistentReservations: 0
2023/11/30 20:19:00Z: Driver: AWS PV Driver Package v8.4.3
2023/11/30 20:19:01Z: Driver: Amazon Elastic Network Adapter v2.6.0.0
2023/11/30 20:19:01Z: RDPCERTIFICATE-SUBJECTNAME: EC2AMAZ-SO1T009
2023/11/30 20:19:01Z: RDPCERTIFICATE-THUMBPRINT:
 1234567890ABCDEF1234567890ABCDEF1234567890
2023/11/30 20:19:09Z: SSM: Amazon SSM Agent v3.2.1705.0
2023/11/30 20:19:13Z: Username: Administrator
2023/11/30 20:19:13Z: Password: <Password>
1234567890abcdef1EXAMPLEPASSWORD
</Password>
2023/11/30 20:19:14Z: User data format: no_user_data
2023/11/30 20:19:14Z: EC2LaunchTelemetry: IsTelemetryEnabled=true
2023/11/30 20:19:14Z: EC2LaunchTelemetry: AgentOsArch=windows_amd64
2023/11/30 20:19:14Z: EC2LaunchTelemetry: IsAgentScheduledPerBoot=true
2023/11/30 20:19:14Z: EC2LaunchTelemetry: AgentCommandErrorCode=0
2023/11/30 20:19:14Z: Message: Windows is Ready to use
```

EC2Launch v2 콘솔 로그 메시지

다음은 모든 EC2Launch v2 콘솔 로그 메시지 목록입니다.

```
Message: Error EC2Launch service is stopping. {error message}
 Error setting up EC2Launch agent folders
 See instance logs for detail
 Error stopping service
 Error initializing service
Message: Windows sysprep configuration complete
Message: Invalid administrator username: {invalid username}
Message: Invalid administrator password
Username: {username}
Password: <Password>{encrypted password}</Password>
AMI Origin Version: {amiVersion}
AMI Origin Name: {amiName}
Microsoft Windows NT {currentVersion}.{currentBuildNumber}
OsVersion: {currentVersion}
OsProductName: {productName}
OsBuildLabEx: {buildLabEx}
```

```
OsCurrentBuild: {currentBuild}
OsReleaseId: {releaseId}
Language: {language}
TimeZone: {timeZone}
Offset: UTC {offset}
Launch agent: EC2Launch {BuildVersion}
AMI-ID: {amiId}
Instance-ID: {instanceId}
Instance Type: {instanceType}
RDPCERTIFICATE-SUBJECTNAME: {certificate subject name}
RDPCERTIFICATE-THUMBPRINT: {thumbprint hash}
SqlServerBilling: {sql billing}
SqlServerInstall: {sql patch leve, edition type}
Driver: AWS NVMe Driver {version}
Driver: Inbox NVMe Driver {version}
Driver: AWS PV Driver Package {version}
Microsoft-Hyper-V is installed.
Unable to get service status for vmms
Microsoft-Hyper-V is {status}
SSM: Amazon SSM Agent {version}
AWS VSS Version: {version}
Message: Windows sysprep configuration complete
Message: Windows is being configured. SysprepState is {state}
Windows is still being configured. SysprepState is {state}
Message: Windows is Ready to use
Message: Waiting for meta-data accessibility...
Message: Meta-data is now available.
Message: Still waiting for meta-data accessibility...
Message: Failed to find primary network interface...retrying...
User data format: {format}
```

EC2Launch v2 버전 기록

버전 기록

- EC2Launch v2 버전 기록
- EC2Launch v2 마이그레이션 도구 버전 기록

EC2Launch v2 버전 기록

다음 표에서는 EC2Launch v2의 릴리스 버전에 대해 설명합니다.

버전	세부 정보	릴리스 날짜
2.1.1	• IPv6에 대한 전체 지원이 추가되었습니다.	2025년 5월 14일
2.0.2107	• 인터페이스에서 IPv4 또는 IPv6 주소를 사용할 수 없는 상황을 처리하기 위해 경로 추가 기능이 향상되었습니다.	2025년 3월 27일
2.0.2081	• RDP 인증서 정보가 제대로 검색되지 않거나 검증되지 않는 문제를 해결했습니다. 필요한 경우 원격 데스크톱 서비스를 자동으로 시작하는 기능이 추가되었습니다. • 서비스 상태를 쿼리할 때 발생하는 문제를 해결하도록 EC2Launch v2 서비스 권한을 조정했습니다.	2025년 2월 4일
2.0.2046	• 기본 운영 체제 바탕 화면 경로를 사용하도록 `agent-config.yml` 파일의 바탕 화면 경로를 업데이트했습니다. • 에이전트 오류가 발생한 위치를 모니터링하기 위해 원격 측정을 추가했습니다. • 에이전트 로그 메시지를 업데이트했습니다.	2024년 10월 3일
2.0.1981	• 관리자가 아닌 사용자에 대한 `EC2Launch.exe` CLI 명령 오류 메시지가 업데이트되었습니다.	2024년 8월 6일
2.0.1948	• 관리자 암호 옵션의 사용을 모니터링하기 위해 텔레메트리가 추가되었습니다. • EC2Launch 권한이 수정되었습니다.	2024년 7월 1일
2.0.1924	• EC2Launch 설정 UI를 업데이트했습니다. • 배경화면 CLI 명령을 업데이트했습니다.	2024년 6월 10일

버전	세부 정보	릴리스 날짜
	• EC2Launch 설치 프로그램을 업데이트했습니다.	
2.0.1914	• 지정되지 않은 게이트웨이 주소로 경로를 추가합니다(IPv4의 경우 `0.0.0.0` 또는 IPv6의 경우 `::`). • 항상 IPv4와 IPv6 경로를 모두 추가하세요. • `Administrator` 사용자 이름이 지정되지 않은 경우 `agent-config.yml` 파일에 사용자 이름이 추가되는 문제를 수정했습니다. • EC2Launch v2 권한이 수정되었습니다.	2024년 6월 5일
2.0.1881	• `setAdminAccount` 작업에 암호화된 암호 옵션을 추가했습니다. • agent-config.yml에서 정적 암호를 암호화하는 CLI 명령을 추가했습니다. • XML 사용자 데이터를 관리자 권한으로 실행할 때 PowerShell 인수를 추가하지 않는 문제를 수정했습니다. 자세한 내용은 Amazon EC2가 Windows 인스턴스의 사용자 데이터를 처리하는 방법을 참조하세요. • `executeScript` 작업 및 사용자 데이터 스크립트가 `LocalSystem` 권한을 사용하여 실행될 때의 PowerShell 인수를 조정했습니다. 인수가 비어 있는 경우 에이전트는 기본값 `-ExecutionPolicy Unrestricted` 을 사용합니다. • 콘솔 로그에 중복된 드라이버 버전을 인쇄하지 못했습니다.	2024년 5월 8일

버전	세부 정보	릴리스 날짜
2.0.1815	• sysprep 이전의 중요한 설치 문제에서 실패하도록 오류 처리를 조정했습니다. • 기본 네트워크 인터페이스에 여러 IP 주소가 할당된 인스턴스에서 월페이퍼 및 호스트 이름 작업이 잘못된 IP 주소를 사용할 수 있는 문제를 수정했습니다. • 월페이퍼 및 호스트 이름 작업이 먼저 IMDS에서 프라이빗 IP를 가져오고, IMDS가 비활성화된 경우 WMI로 페일백되도록 변경되었습니다. • 일시적 오류로 인해 `sc1` 볼륨을 초기화하지 못했던 `initializeVolume` 작업 문제를 수정했습니다.	2024년 3월 6일
2.0.1739	• Windows 관리자 사용자로 실행한 `executeScript` 작업에서 종료 코드가 캡처되지 않는 문제 수정됨	2024년 1월 17일
2.0.1702	• `Telemetry.log` 권한은 표준 사용자의 `read-execute` 로만 제한됩니다. • 시작 실패 시 EC2Launch Windows 서비스를 다시 시작하도록 구성되었습니다. • `route.exe` `stderr` 출력을 로깅하여 `add-routes` 장애를 실행 가능한 항목으로 설정했습니다. • 경로 지표가 [1, 9999] 범위를 벗어날 때 발생하는 문제가 수정되었습니다. • 여러 새 인스턴스 유형에 배경 화면 지원이 추가되었습니다. • Windows 관리자로 실행되어 출력을 `stderr`에 보내는 사용자 데이터 스크립트로 인해 발생하던 문제가 수정되었습니다.	2024년 1월 4일

버전	세부 정보	릴리스 날짜
2.0.1643	• `ebsnvme-id.exe` 도구를 버전 1.1.0.7로 업데이트했습니다. • 'metal-*'로 시작하는 메탈 인스턴스 유형(예: metal-48x1)에서 RSS(수신 측 조정) 및 수신 대기열 깊이 설정과 관련된 문제를 수정했습니다. • 에이전트를 차단하는 XML userdata 명령을 보고하는 원격 측정 이벤트를 제거했습니다. • 레지스트리 항목 `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Dnscache\Parameters\DomainNameDevolutionLevel` 을 기준으로 도메인 이름 권한 승계를 제한하도록 `setDnsSuffix` 작업을 업데이트했습니다. • 네트워크 경로를 추가하는 퍼블릭 작업 및 CLI를 추가했습니다. • 참고 - Windows Server 2012를 공식적으로 지원하는 마지막 버전입니다. • 참고 - 32비트 운영 체제를 공식적으로 지원하는 마지막 버전입니다.	2023년 10월 4일
2.0.1580	• 로그 파일 권한을 수정할 때 시작 에이전트에서 오류를 처리하는 방식을 변경했습니다. • 직렬 포트 연결에 대한 제한 시간을 추가했습니다. 제한 시간을 사용하면 직렬 포트가 사용 중인 경우에도 시작 에이전트가 계속 실행될 수 있습니다.	2023년 9월 5일

버전	세부 정보	릴리스 날짜

2.0.1521	• `EC2Launch.exe` reset 및 sysprep 명령의 `–block` 플래그가 더 이상 사용되지 않습니다.
	• `EC2Launch.exe` 에서 인라인 `executeScript` 태스크에 사용되는 reset 및 sysprep 명령을 감지하고 처리하도록 업데이트했습니다. 이러한 명령은 `executeScript` 태스크에서 해당 명령들을 실행한 후 에이전트 실행이 중지하도록 합니다.
	• XML 사용자 데이터 스크립트가 기본적으로 인라인으로 실행되도록 업데이트했습니다.
	• XML 사용자 데이터 스크립트를 활성화하고 새 `detach` 태그로 분리되어 실행되도록 합니다. 자세한 내용은 사용자 데이터 스크립트을 참조하세요.
	• 에이전트 로그에 대해 다음과 같은 변경 사항이 발생했습니다.
	  • 에이전트 로그 메시지를 업데이트했습니다.
	  • 에이전트 로그에서 `executeScript` 콘텐츠 및 출력을 제거했습니다.
	  • 에이전트 로그에서 `executeProgram` 인수 및 출력을 제거했습니다.
	• 콘솔 로그에 대해 다음과 같은 변경 사항이 발생했습니다.
	  • 콘솔 로그에 `EnableSCSIPersistentReservations` 값을 추가했습니다.
		2023년 7월 3일

버전	세부 정보	릴리스 날짜
2.0.1303	• 네트워크 경로를 추가할 때 추가 오류 처리 및 로그 라인을 추가했습니다. • PreReady 단계에서 `executeScript` 와 `executeProgram` 작업을 허용했습니다. • executeScript 작업의 출력과 유사한 출력 파일을 생성하도록 `executeProgram` 작업을 업데이트했습니다. 자세한 내용은 executeProgram 섹션을 참조하세요. • XML 사용자 데이터에서 차단 에이전트 명령의 사용을 모니터링하는 원격 측정이 추가되었습니다.	2023년 5월 3일
2.0.1245	• 충돌 호출 스택을 명확한 텍스트로 기록하여 충돌에 대한 가시성을 개선했습니다. • Amazon EC2Launch 서비스가 EventLog 서비스보다 빠르게 시작될 때 발생하는 충돌을 해결하기 위해 EventLog 서비스를 시작 종속성으로 추가했습니다. • 에이전트 구성 파일(예: EC2Launch v1 및 EC2Config)에서 PostReady 단계 이전에 XML 사용자 데이터를 실행하도록 했습니다. • 에이전트 구성 파일에서 PostReady 단계 이전에 사용자 데이터를 실행하도록 YAML 사용자 데이터 버전 1.1을 추가했습니다(YAML 사용자 데이터 버전 1.0은 에이전트 구성 파일의 PostReady 단계 이후에 실행됨).	2023년 3월 8일

버전	세부 정보	릴리스 날짜
2.0.1173	• 월페이퍼에 인스턴스 태그를 표시하는 선택적 기능을 추가합니다. 자세한 내용은 setWallpaper 섹션을 참조하세요. • Elastic Graphics에 대한 보안 그룹이 제대로 설정되지 않은 경우 오류 처리를 추가합니다. • 인스턴스 메타데이터 서비스가 활성화되지 않은 경우 제한 시간을 수정합니다.	2023년 2월 6일
2.0.1121	• 퍼블릭 IPv4 주소가 할당되지 않은 경우 배경 화면에 404 오류가 표시되는 문제를 해결합니다. • 디바이스의 드라이브 문자가 D로 설정된 경우 볼륨의 파일 시스템이 NTFS 대신 RAW로 포맷되는 문제를 수정합니다. • NVMe SSD 볼륨이 EBS 볼륨으로 잘못 식별되는 문제를 수정합니다. • IMDS가 비활성화된 경우 Windows를 활성화할 때 발생하는 오류를 수정합니다.	2023년 1월 4일

버전	세부 정보	릴리스 날짜
2.0.1082	• IMDS가 비활성화된 경우 `setWallpaper` : `privateIpAddress` 필드가 비어 있는 문제를 해결합니다. • IMDS가 비활성화된 경우 호스트 이름을 프라이빗 IPv4 주소로 설정할 때 발생하는 문제를 해결합니다. • Windows Server 2012에서 볼륨을 초기화할 때 발생하는 문제를 해결합니다. • 점보 프레임 설정 관련 문제를 해결합니다. • 인스턴스 시작 시 SSH 키가 지정되지 않은 경우에 발생하는 오류를 해결합니다. • Windows에 'ReleaseId' 레지스트리 키가 없을 때 발생하는 Windows Server 2012의 오류를 해결합니다.	2022년 12월 7일
2.0.1011	• PnPDeviceID가 비어 있을 때 네트워크 어댑터를 찾는 로직을 수정합니다.	2022년 11월 11일
2.0.1009	• PCI 세그먼트 정보를 사용하여 콘솔 포트를 선택합니다.	2022년 11월 8일
2.0.982	• RDP 정보를 가져오기 위한 재시도 논리를 추가합니다. • `d2.8xlarge` 인스턴스에서 볼륨 초기화 중 발생하는 오류를 수정합니다. • 재부팅 후 잘못된 네트워크 어댑터가 선택될 수 있는 문제를 수정합니다. • ACPI SPCR을 사용할 수 없을 때 나타나는 잘못된 경보 오류 메시지를 제거합니다.	2022년 10월 31일

버전	세부 정보	릴리스 날짜
2.0.863	• IMDSv2 요청만 수행하도록 IMDS 대기 로직을 업데이트합니다. • 이미 초기화되었지만 마운트되지 않은 볼륨에 드라이브 문자를 할당하는 논리를 추가합니다. • 키 쌍 유형이 지원되지 않는 경우 보다 구체적인 오류 메시지를 인쇄합니다. • 3010 재부팅 코드 버그를 수정합니다. • 잘못된 base64 인코딩 사용자 데이터에 대한 검사를 추가합니다.	2022년 7월 6일
2.0.698	• 스크립트를 실행할 때 로그 출력의 오타를 수정합니다.	2022년 1월 30일
2.0.674	• 원격 측정은 사용/비활성화된 개인 정보 제어를 업로드합니다. • `index out of bounds` 버그를 수정합니다. • `sysprep` 중 배경 화면 바로 가기를 제거합니다.	2021년 11월 15일
2.0.651	• EC2Launch v2 설치 시에 레거시 에이전트를 제거하는 로직를 추가합니다. • 루트 볼륨이 볼륨 0으로 나열되지 않은 경우에 발생하는 `list-volume` CLI 문제를 수정합니다.	2021년 10월 7일
2.0.592	• 단계 상태를 올바르게 보고하도록 버그가 수정되었습니다. • 로그 파일이 닫힐 때 잘못된 경보 오류 메시지를 제거합니다. • 원격 측정을 추가합니다.	2021년 8월 31일

버전	세부 정보	릴리스 날짜
2.0.548	• 16진수 IP 호스트 이름에 선행 0을 추가합니다. • `enableOpenSsh` 태스크에 대한 파일 권한 작업을 수행합니다. • sysprep 명령 충돌을 수정합니다.	2021년 8월 4일
2.0.470	• DHCP가 인스턴스에 IP를 할당할 때까지 기다리는 네트워크 단계의 버그를 수정합니다. • `SearchList` 레지스트리 키가 존재하지 않는 경우 `setDnsSuffix` 로 버그를 수정합니다. • `setDnsSuffix` 로 DNS 권한 승계 논리의 버그를 수정합니다. • 중간 재부팅 후 네트워크 경로를 추가합니다. • `initializeVolume` 을 허용하여 기존 볼륨의 문자를 다시 지정합니다. • 버전 하위 명령에서 추가 정보를 제거합니다.	2021년 7월 20일
2.0.285	• 분리된 프로세스에서 사용자 스크립트를 실행하는 옵션을 추가합니다. • 이제 기존 사용자 데이터(XML 사용자 데이터)는 분리된 프로세스에서 실행되며, 이는 이전 시작 에이전트와 유사한 동작입니다. • CLI 플래그를 `sysprep` 및`reset` 명령에 추가하여 서비스가 중지될 때까지 해당 명령을 차단할 수 있습니다. • 구성 폴더 권한을 제한합니다.	2021년 3월 8일

버전	세부 정보	릴리스 날짜
2.0.207	• hostName 작업에 선택적 setHostName 필드를 추가합니다. • 재부팅 버그를 수정합니다. executeScript 작업을 재부팅하고 executeProgram 은 실행 중으로 표시됩니다. • 상태 명령에 더 많은 반환 코드를 추가합니다. • t2.nano 인스턴스 유형에서 실행할 때 시작 문제를 해결하기 위해 부트스트랩 서비스를 추가합니다. • 새로 설치 모드를 수정하여 설치 프로그램이 추적하지 않는 파일을 제거합니다.	2021년 2월 2일
2.0.160	• 잘못된 스테이지 이름을 감지하는 validate 명령을 수정합니다. • w32tm resync 태스크에 addroutes 명령을 추가합니다. • DNS 접미사 검색 순서 변경 관련 문제가 수정되었습니다. • 잘못된 사용자 데이터의 보고를 개선하는 확인 조건을 추가합니다.	2020년 12월 4일
2.0.153	UserData에 Sysprep 기능을 추가합니다.	2020년 11월 3일

버전	세부 정보	릴리스 날짜
2.0.146	• 영어 이외의 AMI에서 RootExtend와 관련된 문제를 해결합니다. • 사용자 그룹에 로그 파일에 대한 쓰기 권한을 부여합니다. • GPT 볼륨에 대한 MS 예약 파티션을 생성합니다. • Amazon EC2Launch 설정에 list-volumes 명령 및 볼륨 드롭다운을 추가합니다. • yaml 또는 json 형식으로 agent-config.yml 파일을 출력하기 위한 get-agent-config 명령을 추가합니다. • 퍼블릭 키가 감지되지 않으면 정적 암호를 지웁니다.	2020년 10월 6일
2.0.124	• 배경 화면에 OS 버전을 표시하는 옵션을 추가합니다. • 암호화된 EBS 볼륨을 초기화합니다. • 로컬 DNS 이름이 없는 VPC에 대한 경로를 추가합니다.	2020년 9월 10일
2.0.104	• DNS 접미사 검색 목록이 존재하지 않으면 만듭니다. • 요청받지 않으면 최대 절전 모드를 건너뜁니다.	2020년 8월 12일
2.0.0	최초 릴리스.	2020년 6월 30일

EC2Launch v2 마이그레이션 도구 버전 기록

다음 표에서는 EC2Launch v2 마이그레이션 도구의 릴리스 버전에 대해 설명합니다.

새로운 EC2Launch v2 에이전트 버전이 릴리스되면 알림을 받을 수 있습니다. 자세한 내용은 EC2 Windows 시작 에이전트 알림 구독 섹션을 참조하세요.

버전	세부 정보	릴리스 날짜
1.0.440	• EC2Launch v2 에이전트의 최신 버전인 2.0.2046으로 마이그레이션 도구를 업데이트합니다.	2024년 10월 28일
1.0.413	• EC2Launch v2 에이전트의 최신 버전인 2.0.1981로 마이그레이션 도구를 업데이트합니다.	2024년 8월 9일
1.0.412	• EC2Launch v2 에이전트의 최신 버전인 2.0.1948로 마이그레이션 도구를 업데이트합니다.	2024년 8월 7일
1.0.396	• EC2Launch v2 에이전트의 최신 버전인 2.0.1924로 마이그레이션 도구를 업데이트합니다.	2024년 6월 11일
1.0.394	• EC2Launch v2 에이전트의 최신 버전인 2.0.1914로 마이그레이션 도구를 업데이트합니다.	2024년 6월 6일
1.0.384	• EC2Launch v2 에이전트의 최신 버전인 2.0.1881로 마이그레이션 도구를 업데이트합니다.	2024년 5월 8일
1.0.358	• EC2Launch v2 에이전트의 최신 버전인 2.0.1815로 마이그레이션 도구를 업데이트합니다.	2024년 3월 8일
1.0.345	• EC2Launch v2 에이전트의 최신 버전인 2.0.1739로 마이그레이션 도구를 업데이트.	2024년 1월 18일
1.0.342	• EC2Launch v2 에이전트의 최신 버전인 2.0.1702로 마이그레이션 도구를 업데이트.	2024년 1월 5일
1.0.331	• EC2Launch v2 에이전트의 최신 버전인 2.0.1643으로 마이그레이션 도구를 업데이트. •	2023년 11월 3일

버전	세부 정보	릴리스 날짜
	`.Install.ps1 -DryRun` 실행 중 발생하는 오류를 수정합니다. • EC2Config에서 마이그레이션하는 동안 암호 구성이 `random`으로 잘못 설정되는 문제를 수정합니다. • EC2Launch에서 마이그레이션하는 동안 `setWallpaper` 를 `False`로 설정할 경우 발생하는 오류를 수정합니다.	
1.0.303	EC2Launch v2 에이전트의 최신 버전인 2.0.1580으로 마이그레이션 도구를 업데이트.	2023년 9월 14일
1.0.286	EC2Launch v2 에이전트의 최신 버전인 2.0.1521로 마이그레이션 도구를 업데이트.	2023년 7월 14일
1.0.272	EC2Launch v2 에이전트의 최신 버전인 2.0.1303으로 마이그레이션 도구를 업데이트.	2023년 5월 3일
1.0.262	EC2Launch v2 에이전트의 최신 버전인 2.0.1245로 마이그레이션 도구를 업데이트.	2023년 3월 9일
1.0.241	EC2Launch v2 에이전트의 버전 번호가 2.0.1011로 높아집니다.	2022년 12월 7일
1.0.218	• 인스턴스 메타데이터에서 검색된 리전 값을 검증합니다. • 언어 팩의 마이그레이션 실패 버그를 수정합니다. • EC2Launch v2 에이전트의 버전 번호가 2.0.863으로 높아집니다.	2022년 9월 3일
1.0.162	• 레거시 에이전트를 제거하는 로직을 EC2Launch v2 MSI로 이동합니다. • EC2Launch v2 에이전트의 버전 번호가 2.0.698로 높아집니다.	2022년 3월 18일

버전	세부 정보	릴리스 날짜
1.0.136	EC2Launch v2 에이전트의 버전 번호가 2.0.651로 높아집니다.	2021년 10월 13일
1.0.130	EC2Launch v2 에이전트의 버전 번호가 2.0.548로 높아집니다.	2021년 8월 5일
1.0.113	IMDSv1 대신 IMDSv2를 사용합니다.	2021년 6월 4일
1.0.101	EC2Launch v2 에이전트의 버전 번호가 2.0.285로 높아집니다.	2021년 3월 12일
1.0.86	EC2Launch v2 에이전트의 버전 번호가 2.0.207로 높아집니다.	2021년 2월 3일
1.0.76	EC2Launch v2 에이전트의 버전 번호가 2.0.160으로 높아집니다.	2020년 12월 4일
1.0.69	EC2Launch v2 에이전트의 버전 번호가 2.0.153으로 높아집니다.	2020년 11월 5일
1.0.65	EC2Launch v2 에이전트의 버전 번호가 2.0.146으로 높아집니다.	2020년 10월 9일
1.0.60	EC2Launch v2 에이전트의 버전 번호가 2.0.124로 높아집니다.	2020년 9월 10일
1.0.54	• 에이전트가 설치되지 않은 경우 EC2Launch v2를 설치합니다. • EC2Launch v2 에이전트의 버전 번호가 2.0.104로 높아집니다. • SSM Agent를 분리합니다.	2020년 8월 12일
1.0.50	NuGet 종속성을 제거합니다.	2020년 8월 10일

버전	세부 정보	릴리스 날짜
1.0.0	최초 릴리스.	2020년 6월 30일

EC2Launch v1 에이전트를 사용하여 EC2 Windows 인스턴스 시작 중 태스크 수행

Windows Server 2016 및 2019용 Amazon 관리형 AMI에는 EC2Launch라는 Windows Powershell 스크립트 세트가 포함되어 있습니다. EC2Launch는 초기 인스턴스 부팅 중에 작업을 수행합니다. AWS Windows AMI에 포함된 EC2Launch 버전에 대한 자세한 내용은 AWS Windows AMI 참조를 참조하세요.

> **Note**
>
> Windows Server 2016 이상 운영 체제용 최신 시작 에이전트는 EC2Config와 EC2Launch를 모두 대체하며 AWS Windows Server 2016 및 2019 AMI에 `EC2LaunchV2-Windows_Server-*`로 시작하는 이름으로 사전 설치되어 제공됩니다. 마이그레이션 도구를 사용하여 EC2Launch v2로 마이그레이션하거나, Windows Server 2016 및 2019에 에이전트를 수동으로 설치하고 구성할 수도 있습니다.
> IMDSv2에서 EC2Launch를 사용하려면 버전이 1.3.2002730 이상이어야 합니다.

다음 Windows PowerShell 명령을 사용하여 설치된 EC2Launch 버전을 확인할 수 있습니다.

```
Test-ModuleManifest -Path "C:\ProgramData\Amazon\EC2-Windows\Launch\Module
\Ec2Launch.psd1" | Select Version
```

EC2Launch 작업

EC2Launch는 초기 인스턴스 부팅 중에 기본적으로 다음 작업을 수행합니다.

- 인스턴스에 대한 정보를 렌더링하는 새로운 월페이퍼(wallpaper)를 설정합니다.
- 컴퓨터 이름을 인스턴스의 프라이빗 IPv4 주소로 설정합니다.
- 인스턴스 정보를 Amazon EC2 콘솔에 전송합니다.
- RDP 인증서 지문을 EC2 콘솔에 전송합니다.
- 관리자 계정에 대한 무작위 암호를 설정합니다.

- DNS 접미사를 추가합니다.
- 분할되지 않은 공간을 포함하도록 운영 체제 파티션을 동적으로 확장합니다.
- 사용자 데이터를 실행합니다(지정된 경우). 사용자 데이터 지정에 대한 자세한 내용은 사용자 데이터 입력을 사용하여 EC2 인스턴스를 시작할 때 명령 실행 섹션을 참조하세요.
- 영구 정적 경로를 설정하여 메타데이터 서비스 및 AWS KMS 서버에 도달합니다.

> **Important**
>
> 사용자 지정 AMI를 이 인스턴스에서 생성한 경우 이 경로는 OS 구성의 일부로서 캡처되며 AMI에서 시작한 새로운 인스턴스는 서브넷 배치와 상관없이 동일한 경로를 유지합니다. 경로를 업데이트하려면 사용자 지정 AMI 시작 시 Server 2016 이후에 대한 메타데이터/KMS 경로 업데이트 섹션을 참조하세요.

다음 작업은 EC2Config 서비스와 역방향 호환성을 유지하는 데 도움이 됩니다. 스타트업 중에 이러한 작업을 수행하도록 EC2Launch를 구성할 수도 있습니다.

- 둘째 EBS 볼륨을 초기화합니다.
- Windows Event 로그를 EC2 콘솔 로그에 전송합니다.
- Windows is ready to use 메시지를 EC2 콘솔에 전송합니다.

EC2Launch 디렉터리 구조

EC2Launch는 기본적으로 루트 디렉터리 `C:\ProgramData\Amazon\EC2-Windows\Launch`의 Windows Server 2016 이후 AMI에 설치됩니다.

> **Note**
>
> 기본적으로 Windows는 파일과 폴더를 `C:\ProgramData` 아래에 숨깁니다. EC2Launch 디렉터리와 파일을 보려면 Windows 탐색기에 경로를 입력하거나 숨겨진 파일과 폴더를 표시하도록 폴더 속성을 변경해야 합니다.

`Launch` 디렉터리에는 다음 하위 디렉터리가 포함됩니다.

- `Scripts` - EC2Launch를 구성하는 PowerShell 스크립트가 포함됩니다.
- `Module` - Amazon EC2와 관련된 스크립트를 빌드하기 위한 모듈이 포함됩니다.

- `Config` - 사용자가 사용자 지정할 수 있는 스크립트 구성 파일이 포함됩니다.
- `Sysprep` - Sysprep 리소스가 포함됩니다.
- `Settings` - Sysprep 그래픽 사용자 인터페이스용 애플리케이션이 포함됩니다.
- `Library` - EC2 시작 에이전트용 공유 라이브러리를 포함합니다.
- `Logs` - 스크립트가 생성하는 로그 파일이 포함됩니다.

원격 측정

원격 측정 데이터는 AWS가 사용자의 요구 사항을 더 잘 이해하고, 문제를 진단하고, AWS 서비스의 경험을 개선할 기능을 제공하는 데 도움이 되는 추가 정보입니다.

EC2Launch 버전 `1.3.2003498` 이상은 사용량 지표 및 오류와 같은 원격 측정 데이터를 수집합니다. 이 데이터는 EC2Launch가 실행되는 Amazon EC2 인스턴스에서 수집됩니다. 여기에는 AWS가 소유한 모든 Windows AMI가 포함됩니다.

EC2Launch에서 수집되는 원격 측정 데이터의 유형은 다음과 같습니다.

- 사용량 정보 - 에이전트 명령, 설치 방법 및 예약된 실행 빈도입니다.
- 오류 및 진단 정보 - 에이전트 설치 및 실행 오류 코드입니다.

수집되는 데이터의 예:

```
2021/07/15 21:44:12Z: EC2LaunchTelemetry: IsAgentScheduledPerBoot=true
2021/07/15 21:44:12Z: EC2LaunchTelemetry: IsUserDataScheduledPerBoot=true
2021/07/15 21:44:12Z: EC2LaunchTelemetry: AgentCommandCode=1
2021/07/15 21:44:12Z: EC2LaunchTelemetry: AgentCommandErrorCode=5
2021/07/15 21:44:12Z: EC2LaunchTelemetry: AgentInstallCode=2
2021/07/15 21:44:12Z: EC2LaunchTelemetry: AgentInstallErrorCode=0
```

원격 측정은 기본적으로 활성화됩니다. 언제든지 원격 측정 데이터 수집을 비활성화할 수 있습니다. 원격 측정이 활성화되면 EC2Launch는 별도로 고객에게 알리지 않고 원격 측정 데이터를 전송합니다.

원격 분석을 사용하거나 비활성화하는 선택 사항이 수집됩니다.

원격 측정 수집을 옵트인 또는 옵트아웃할 수 있습니다. 원격 측정 옵션을 준수하기 위해 원격 측정 옵트인 또는 옵트아웃 선택이 수집됩니다.

원격 측정 가시성

원격 측정이 활성화되면 Amazon EC2 콘솔 출력에 다음과 같이 표시됩니다.

```
2021/07/15 21:44:12Z: Telemetry: <Data>
```

인스턴스의 원격 측정 비활성화

시스템 환경 변수를 설정하여 원격 측정을 비활성화하려면 관리자로 다음 명령을 실행합니다.

```
setx /M EC2LAUNCH_TELEMETRY 0
```

설치 중 원격 측정을 비활성화하려면 다음과 같이 `install.ps1`을 실행합니다.

```
. .\install.ps1 -EnableTelemetry:$false
```

EC2Launch에 대한 더 많은 주제

- 최신 버전의 EC2Launch 설치
- Windows 인스턴스에서 EC2Launch v1 에이전트 구성
- EC2Launch 버전 기록

최신 버전의 EC2Launch 설치

다음 절차를 이용해 인스턴스에 최신 버전의 EC2Launch를 다운로드하여 설치합니다.

최신 버전의 EC2Launch 다운로드하여 설치하기

1. 이미 인스턴스에 EC2Launch를 설치하여 구성한 경우 EC2Launch 구성 파일의 백업을 만듭니다. 설치 프로세스는 이 파일에 변경 사항을 보존하지 않습니다. 기본적으로 `C:\ProgramData\Amazon\EC2-Windows\Launch\Config` 디렉터리에 파일이 위치합니다.
2. 인스턴스의 디렉터리로 EC2-Windows-Launch.zip 파일을 다운로드합니다.
3. `EC2-Windows-Launch.zip` 파일을 다운로드한 동일한 디렉터리에 install.ps1을 다운로드합니다.
4. `install.ps1` 실행
5. EC2Launch 구성 파일의 백업을 만든 경우 `C:\ProgramData\Amazon\EC2-Windows\Launch\Config` 디렉터리에 복사합니다.

PowerShell을 사용하여 최신 버전의 EC2Launch 다운로드하여 설치하기

이미 인스턴스에 EC2Launch를 설치하여 구성한 경우 EC2Launch 구성 파일의 백업을 만듭니다. 설치 프로세스는 이 파일에 변경 사항을 보존하지 않습니다. 기본적으로 `C:\ProgramData\Amazon\EC2-Windows\Launch\Config` 디렉터리에 파일이 위치합니다.

PowerShell을 사용하여 최신 버전의 EC2Launch를 설치하려면 관리자 권한으로 PowerShell 창에서 다음 명령을 실행합니다.

```
mkdir $env:USERPROFILE\Desktop\EC2Launch
$Url = "https://s3.amazonaws.com/ec2-downloads-windows/EC2Launch/latest/EC2-Windows-Launch.zip"
$DownloadZipFile = "$env:USERPROFILE\Desktop\EC2Launch\" + $(Split-Path -Path $Url -Leaf)
Invoke-WebRequest -Uri $Url -OutFile $DownloadZipFile
$Url = "https://s3.amazonaws.com/ec2-downloads-windows/EC2Launch/latest/install.ps1"
$DownloadZipFile = "$env:USERPROFILE\Desktop\EC2Launch\" + $(Split-Path -Path $Url -Leaf)
Invoke-WebRequest -Uri $Url -OutFile $DownloadZipFile
& $env:USERPROFILE\Desktop\EC2Launch\install.ps1
```

> **Note**
>
> Windows Server 2016을 사용 중이고 파일을 다운로드할 때 오류가 발생하는 경우 PowerShell 터미널에서 TLS 1.2를 활성화해야 할 수 있습니다. 다음 명령을 사용하여 현재 PowerShell 세션에 대해 TLS 1.2를 활성화한 다음 다시 시도해보세요.
>
> ```
> [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
> ```

시작 에이전트를 확인하여 설치를 확인합니다. 관리자 권한으로 PowerShell 창에서 다음 명령을 실행합니다.

```
Import-Module C:\ProgramData\Amazon\EC2-Windows\Launch\Module\Ec2Launch.psm1
Import-LocalizedData -BaseDirectory C:\ProgramData\Amazon\EC2-Windows\Launch\Module\ -FileName 'Ec2Launch.psd1' -BindingVariable moduleManifest
$moduleManifest.Get_Item('ModuleVersion')
```

Windows 인스턴스에서 EC2Launch v1 에이전트 구성

인스턴스가 처음으로 초기화된 후에는 다시 실행하고 다른 스타트업 작업을 수행하도록 EC2Launch를 구성할 수 있습니다.

작업

- 초기화 작업 구성
- 매 부팅마다 실행하도록 EC2Launch 예약
- 드라이브 및 맵 드라이브 문자 초기화
- EC2 콘솔에 Windows 이벤트 로그 전송
- 부팅 성공 후 Windows is ready 메시지 전송

초기화 작업 구성

`LaunchConfig.json` 구성 파일에서 설정을 지정하여 다음과 같은 초기화 작업을 수행 혹은 해제합니다.

- 컴퓨터 이름을 인스턴스 프라이빗 IPv4 주소로 설정합니다.
- 모니터가 항상 켜져 있도록 설정합니다.
- 새로운 월페이퍼(wallpaper)를 설정합니다.
- DNS 접미사 목록을 추가합니다.

> **Note**
>
> 이렇게 하면 다음 도메인에 대한 DNS 접미사 조회가 추가되고 다른 표준 접미사가 구성됩니다. 시작 에이전트가 DNS 접미사를 설정하는 방법에 대한 자세한 내용은 EC2 Windows 시작 에이전트용 DNS 접미사 구성 섹션을 참조하세요.
>
> ```
> region.ec2-utilities.amazonaws.com
> ```

- 부팅 볼륨 크기를 확장합니다.
- 관리자 암호를 설정합니다.

초기화 설정을 구성하려면

1. 구성할 인스턴스에서 텍스트 편집기에 `C:\ProgramData\Amazon\EC2-Windows\Launch\Config\LaunchConfig.json` 파일을 엽니다.
2. 다음 설정을 필요에 따라 업데이트하고 변경 내용을 저장합니다. `adminPassword`이 `adminPasswordtype`인 경우에만 `Specify`에 암호를 입력합니다.

```
{
 "setComputerName": false,
 "setMonitorAlwaysOn": true,
 "setWallpaper": true,
 "addDnsSuffixList": true,
 "extendBootVolumeSize": true,
 "handleUserData": true,
 "adminPasswordType": "Random | Specify | DoNothing",
 "adminPassword": "password that adheres to your security policy (optional)"
}
```

암호 유형은 다음과 같이 정의됩니다.

`Random`

EC2Launch는 암호를 생성하고 사용자의 키를 사용하여 암호를 암호화합니다. 인스턴스가 재부팅 또는 중지되었다가 시작된 경우 이 암호가 그대로 유지되도록 시스템은 인스턴스가 시작된 후 이 설정을 비활성화합니다.

`Specify`

`adminPassword`에 지정한 암호가 EC2Launch에 사용됩니다. 암호가 시스템 요구 사항에 맞지 않으면 EC2Launch에서 임의의 암호를 대신 생성합니다. 암호는 `LaunchConfig.json`에 일반 텍스트로 저장되며 Sysprep에서 관리자 암호를 설정한 후에 삭제됩니다. EC2Launch는 사용자의 키를 사용하여 암호를 암호화합니다.

`DoNothing`

`unattend.xml` 파일에 지정한 암호가 EC2Launch에 사용됩니다. `unattend.xml`에 암호를 지정하지 않으면 관리자 계정이 비활성화됩니다.

3. Windows PowerShell에서 다음 명령을 실행하여 스크립트가 Windows 예약된 작업으로 실행되도록 예약합니다. 스크립트는 다음 부팅 중에 이 작업을 한 번 실행한 다음 이 작업이 다시 실행되지 않도록 비활성화합니다.

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -Schedule
```

매 부팅마다 실행하도록 EC2Launch 예약

처음 부팅할 때만이 아니라 매 부팅마다 실행하도록 EC2Launch를 예약할 수 있습니다.

매 부팅마다 실행하도록 EC2Launch를 활성화하는 방법:

1. Windows PowerShell을 열어 다음 명령을 실행합니다.

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -
SchedulePerBoot
```

2. 또는 다음 명령을 사용하여 실행 파일을 실행합니다.

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Settings\Ec2LaunchSettings.exe
```

   그런 다음 `Run EC2Launch on every boot`를 선택합니다. EC2 인스턴스 `Shutdown without Sysprep` 또는 `Shutdown with Sysprep`을 지정할 수 있습니다.

> **Note**
>
> 부팅 시마다 실행하도록 EC2Launch를 활성화할 경우 다음 번에 EC2Launch를 실행할 때 다음과 같이 됩니다.
>
> - `AdminPasswordType`이 여전히 `Random`인 경우 EC2Launch가 다음 부팅 시 새 암호를 생성합니다. EC2Launch가 후속 부팅에서 새 암호를 생성하지 않도록, 부팅 후 `AdminPasswordType`이 자동으로 `DoNothing`으로 설정됩니다. EC2Launch가 최초 부팅 시 새 암호를 생성하지 않도록 하려면 `AdminPasswordType`을 `DoNothing`으로 설정합니다.
> - 사용자 데이터에서 `HandleUserData`를 `false`로 설정한 경우가 아니면 `persist`가 `true`로 다시 설정됩니다. 자세한 내용은 the section called “사용자 데이터 스크립트” 섹션을 참조하세요.

드라이브 및 맵 드라이브 문자 초기화

`DriveLetterMappingConfig.json` 파일에서 설정을 지정하여 드라이브를 초기화 및 포맷하고 드라이브 문자를 EC2 인스턴스의 볼륨에 매핑합니다. 이 스크립트는 아직 초기화 및 파티셔닝되지 않은 드라이브를 초기화합니다. Windows에서 볼륨 세부 정보를 가져오는 방법에 대한 자세한 내용은 Microsoft 설명서에서 Get-Volume을 참조하세요.

드라이브 문자를 볼륨에 매핑하려면

1. 텍스트 에디터에서 C:\ProgramData\Amazon\EC2-Windows\Launch\Config\DriveLetterMappingConfig.json 파일을 엽니다.
2. 다음 볼륨 설정을 지정하고 변경 내용을 저장합니다.

```
{
 "driveLetterMapping": [
  {
   "volumeName": "sample volume",
   "driveLetter": "H"
  }
 ]
}
```

3. 디스크를 초기화할 EC2Launch 스크립트를 수행하기 위해서 Windows PowerShell을 열고 다음 명령을 실행합니다.

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1
```

인스턴스를 부팅할 때마다 디스크를 초기화하려면 다음과 같이 -Schedule 플래그를 추가하세요.

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1 -Schedule
```

EC2 콘솔에 Windows 이벤트 로그 전송

EventLogConfig.json 구성 파일에서 설정을 지정하여 Windows Event 로그를 EC2 콘솔 로그에 전송합니다.

Windows Event 로그를 전송하도록 설정을 구성하려면

1. 인스턴스의 텍스트 편집기에서 C:\ProgramData\Amazon\EC2-Windows\Launch\Config\EventLogConfig.json 파일을 엽니다.
2. 다음 로그 설정을 구성하고 변경 내용을 저장합니다.

```
{
 "events": [
  {
```

```
    "logName": "System",
    "source": "An event source (optional)",
    "level": "Error | Warning | Information",
    "numEntries": 3
   }
 ]
}
```

3. Windows PowerShell에서 다음 명령을 실행합니다. 그러면 시스템에서 인스턴스가 부팅될 때마다 스크립트가 Windows 예약된 작업으로 실행되도록 예약됩니다.

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\SendEventLogs.ps1 -Schedule
```

로그가 EC2 콘솔 로그에 나타나려면 3분 이상 걸릴 수 있습니다.

부팅 성공 후 Windows is ready 메시지 전송

EC2Config 서비스는 매번 부팅 후에 "Windows is ready" 메시지를 EC2 콘솔에 전송했습니다. EC2Launch는 초기 부팅 후에만 이 메시지를 전송합니다. EC2Config 서비스와 역방향 호환성을 위해 매번 부팅한 후에 이 메시지를 전송하도록 EC2Launch를 예약할 수 있습니다. 인스턴스에서 Windows PowerShell을 열고 다음 명령을 실행합니다. 시스템에서 이 스크립트는 Windows 예약된 작업으로 실행되도록 예약됩니다.

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\SendWindowsIsReady.ps1 -Schedule
```

EC2Launch 버전 기록

> **Important**
>
> 2025년 1월 1일부터는 2개의 최신 버전만 지원됩니다. 새 버전이 릴리스되면 이전에 지원된 가장 오래된 버전이 자동으로 비공개로 표시되며 더 이상 다운로드할 수 없습니다.

최신 버전의 EC2Launch를 다운로드하여 설치하려면 최신 버전의 EC2Launch 설치 섹션을 참조하세요.

새로운 EC2Launch 에이전트 버전이 릴리스되면 알림을 받을 수 있습니다. 자세한 내용은 EC2 Windows 시작 에이전트 알림 구독 섹션을 참조하세요.

다음 EC2Launch 에이전트 버전이 지원되며 다운로드할 수 있습니다.

버전	세부 정보	릴리스 날짜
1.3.2005119	• 파라미터 없이 직접적으로 호출하는 경우 `Invoke-Userdata` 가 실패하는 문제를 수정했습니다.	2025년 2월 11일
1.3.2005065	• RDP 인증서 정보가 제대로 검색되지 않거나 검증되지 않는 문제를 해결했습니다. 필요한 경우 원격 데스크톱 서비스를 자동으로 시작하는 기능이 추가되었습니다.	2024년 10월 22일

다음과 같은 이전 버전의 EC2Launch는 더 이상 다운로드할 수 없습니다.

버전	세부 정보	릴리스 날짜
1.3.2005008	• 기본 배경 화면 이미지를 찾을 수 없는 경우 단색 배경으로 돌아가도록 `Set-Wallpaper` 를 업데이트했습니다.	2024년 8월 6일
1.3.2004959	• Windows Server 2025 이상에서 지원되지 않는 설치를 방지하도록 설치 프로그램 로직이 업데이트되었습니다.	2024년 7월 2일
1.3.2004891	• `HandleUserData` 가 예상대로 `false`로 설정되지 않은 문제를 수정했습니다. • `LaunchConfig.json` 에 `Encrypted` 암호 옵션을 추가했습니다. • 기본적으로 사용자 지정 암호를 암호화하도록 `Settings UI` 동작이 변경되었습니다. • 에이전트 구성 파일에서 `Specify` 암호 옵션을 `Encrypted` 암호 옵션으로 변환하도록 `SetAdminPasswordConfig.ps1` 을 추가했습니다.	2024년 5월 31일
1.3.2004617	• 벽지 설정 시 오류 수정됨	2024년 1월 15일

버전	세부 정보	릴리스 날짜
1.3.2004592	• `%ProgramData%\Amazon\EC2-Windows\Launch` 에서 install.ps1로 설정된 액세스 권한이 업데이트되었습니다. • EC2Launch 폴더 및 파일 액세스가 표준 사용자 계정에서 읽기 실행으로만 제한됩니다. • 인스턴스에 대한 인스턴스 메타데이터 서비스(IMDS)가 활성화되지 않은 경우 IMDS가 초기화될 때까지 기다리지 않도록 에이전트가 변경되었습니다. • IMDS 초기화를 기다릴 때 5분의 제한 시간이 추가되었습니다. • 인스턴스 콘솔 로그에서 `Windows is Ready` 메시지 이후가 아닌 이전에 원격 측정 데이터를 쓰도록 에이전트가 변경되었습니다. • 여러 새 인스턴스 유형에 배경 화면 지원이 추가되었습니다. EC2Launch 디렉터리의 액세스 권한 및 사용자 계정 권한에 대한 자세한 내용은 the section called “EC2Launch 디렉터리 구조” 섹션을 참조하세요.	2024년 1월 2일
1.3.2004491	• 관리자 암호 지정 옵션의 사용을 모니터링하기 위해 텔레메트리가 추가되었습니다.	2023년 11월 9일
1.3.2004462	• 직렬 콘솔에 쓸 때마다 플러시를 추가했습니다.	2023년 10월 18일

버전	세부 정보	릴리스 날짜
1.3.2004438	• 레지스트리 항목을 기준으로 도메인 이름 권한 승계 제한: `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Dnscache\Parameters\DomainNameDevolutionLevel` . • `UserdataExecution.log` 권한을 `Administrators` 전용으로만 제한됩니다. • 로그 초기화 실패 시 Windows 이벤트 로그에 오류 메시지를 추가했습니다.	2023년 10월 4일
1.3.2004256	• 콘솔 로그에 `EnableSCSIPersistentReservations` 값을 추가했습니다. • Get-ConsolePort에 대한 재시도 기능이 추가되었습니다.	2023년 7월 7일
1.3.2004052	• 인스턴스 시작 시 SSH 키가 지정되지 않은 경우에 발생하는 오류를 해결합니다. • 실패 시 AmazonSSMAgent Windows 서비스를 다시 시작하도록 업데이트되었습니다. • Sysprep.cmd가 0이 아닌 종료 코드로 실패하는 경우 SysprepInstance.ps1이 실패하도록 업데이트되었습니다.	2023년 3월 8일
1.3.2003975	• `SysprepInstance.ps1` 에서 1의 `$LastErrorCode` 가 반환되어 Packer AMI 빌드에 영향을 주는 문제를 해결했습니다.	2022년 12월 24일

버전	세부 정보	릴리스 날짜
1.3.2003961	• 빠르게 시작되는 인스턴스에서 명시적으로 지정된 관리자 암호를 임의의 암호로 덮어쓰는 문제가 해결되었습니다. • SSM Agent가 더 작은 인스턴스 유형에서 시작되지 않는 문제가 해결되었습니다. • 인스턴스 콘솔 로그에 유효한 RDP 인증서 지문 값 대신 `RDPCERTIFICATE-THUMBPRINT: 0000000 0000000000000000` 이 포함되는 문제가 해결되었습니다.	2022년 12월 6일
1.3.2003923	• PnPDeviceID가 비어 있을 때 네트워크 어댑터를 찾는 로직을 수정합니다.	2022년 11월 9일
1.3.2003919	• PCI 세그먼트 정보를 사용하도록 GET-ConsolePort가 업데이트되었습니다. • 재부팅 후 잘못된 네트워크 어댑터가 선택될 수 있는 문제를 수정했습니다. • start-SSM-Agent 시간 초과 로직을 수정했습니다. • Send-AdminCredentials 함수 별칭에 대한 이전 버전과의 호환성을 수정했습니다.	2022년 11월 8일
1.3.2003857	• 기본 네트워크 어댑터를 선택할 때 기본 게이트웨이를 사용하는 어댑터의 우선 순위를 지정합니다. • 인 메모리 암호 암호화가 확장되었습니다.	2022년 10월 3일

버전	세부 정보	릴리스 날짜
1.3.2003824	• `setComputerName` 중 오류를 수정했습니다. • BYOL 결제 코드가 감지되면 Windows 활성화를 건너뛰는 로직이 추가되었습니다. • 인 메모리 암호 암호화가 추가되었습니다. • `m6id.4xlarge` 에서 볼륨 초기화 중 오류를 수정했습니다.	2022년 8월 30일
1.3.2003691	• IMDSv2 요청만 수행하도록 IMDS 대기 로직을 업데이트했습니다. • eGPU 설치에 영향을 미치는 버그를 수정했습니다.	2022년 6월 21일
1.3.2003639	• 초기화 전에 사용하지 못하도록 네트워크 어댑터 대기 로직을 추가했습니다. • 사소한 문제를 수정했습니다.	2022년 5월 10일
1.3.2003498	• 원격 측정이 추가되었습니다. • 설정 UI의 바로 가기가 추가되었습니다. • PowerShell 스크립트의 형식이 지정되었습니다. • BeforeSysprep.cmd가 완료되기 전에 발생하는 종료 문제가 수정되었습니다.	2022년 1월 31일
1.3.2003411	복잡도가 낮은 암호를 제외하도록 암호 생성 논리를 변경했습니다.	2021년 8월 4일
1.3.2003364	IMDSv2를 지원하는 업데이트된 Install-EgpuManager.	2021년 6월 7일

버전	세부 정보	릴리스 날짜
1.3.2003312	• `setMonitorAlwaysOn` 설정 앞과 뒤에 로그 줄이 추가되었습니다. • AWS Nitro Enclaves 패키지 버전이 콘솔 로그에 추가되었습니다.	2021년 5월 4일
1.3.2003284	사용자 데이터의 저장 위치를 `LocalAppData` 로 업데이트하여 권한 모델을 개선했습니다.	2021년 3월 23일
1.3.2003236	• `Set-AdminAccount` 및 `Randomize-LocalAdminPassword` 에서 사용자 암호를 설정하는 방법이 업데이트되었습니다. • `InitializeDisks` 수정을 통해 디스크가 쓰기 가능으로 설정되기 전에 읽기 전용으로 설정되어 있는지 여부를 확인합니다.	2021년 2월 11일
1.3.2003210	`install.ps1` 에 대한 지역화 수정	2021년 1월 7일
1.3.2003205	`install.ps1` 디렉터리에 대한 권한을 업데이트하는 `%ProgramData%AmazonEC2-Windows\LaunchModuleScripts` 에 대한 보안 수정입니다.	2020년 12월 28일
1.3.2003189	라우팅 추가 후에 `w32tm resync`가 추가되었습니다.	2020년 12월 4일
1.3.2003155	인스턴스 유형 정보가 업데이트되었습니다.	2020년 8월 25일
1.3.2003150	`OsCurrentBuild` 및 `OsReleaseId` 가 콘솔 출력에 추가되었습니다.	2020년 4월 22일
1.3.2003040	IMDS 버전 1 대체 로직이 수정했습니다.	2020년 4월 7일

버전	세부 정보	릴리스 날짜
1.3.2002730	IMDS V2에 대한 지원이 추가되었습니다.	2020년 3월 3일
1.3.2002240	사소한 문제를 수정했습니다.	2019년 10월 31일
1.3.2001660	처음 Sysprep을 실행한 후 암호가 없는 사용자의 자동 로그인 문제를 수정했습니다.	2019년 7월 2일
1.3.2001360	사소한 문제를 수정했습니다.	2019년 3월 27일
1.3.2001220	모든 PowerShell 스크립트는 서명이 된 것입니다.	2019년 2월 28일
1.3.2001200	Windows Server 장애 조치 클러스터의 노드에서 스크립트를 실행하면 드라이브 문자가 로컬 드라이브 문자와 일치하는 원격 노드의 드라이브가 포맷되는 InitializeDisks.ps1 문제를 수정했습니다.	2019년 2월 27일
1.3.2001160	Windows 2019에서 월페이퍼(wallpaper) 누락 문제를 수정했습니다.	2019년 2월 22일
1.3.2001040	• ACPI 문제 해결을 위해 모니터가 절대로 꺼지지 않게 설정하는 플러그인을 추가했습니다. • SQL Server 에디션과 버전을 콘솔에 기록했습니다.	2019년 1월 21일
1.3.2000930	ipv6 사용 ENI에서 메타데이터로 가는 경로를 추가하는 수정사항이 적용되었습니다.	2019년 1월 2일
1.3.2000760	• RSS에 대한 기본 구성과 ENA 디바이스에 대한 수신 대기열 설정을 추가했습니다. • Sysprep 도중 최대 절전 모드를 비활성화했습니다.	2018년 12월 5일

버전	세부 정보	릴리스 날짜
1.3.2000630	• DNS 서버에 대한 라우트 169.254.169.253/32를 추가했습니다. • 관리자 사용자 설정 필터를 추가했습니다. • 인스턴스 최대 절전 모드를 개선했습니다. • 매 부팅마다 실행하도록 EC2Launch를 예약하는 옵션을 추가했습니다.	2018년 11월 9일
1.3.2000430.0	• AMZN 시간 서비스에 라우트 169.254.169.123/32 추가. • GRID 라이선스 서비스에 라우트 169.254.169.249/32 추가. • Systems Manager 시작 시도 시 25초의 시간 제한 추가.	2018년 9월 19일
1.3.200039.0	• EBS NVME 볼륨의 부적절한 드라이브 문자 수정. • NVME 드라이버 버전에 대한 추가 로깅 추가.	2018년 8월 15일
1.3.2000080	사소한 문제를 수정했습니다.	
1.3.610	사용자 데이터에서 파일로 출력 및 오류 리디렉션 문제를 수정했습니다.	
1.3.590	• 월페이퍼(wallpaper)에 누락된 인스턴스 유형을 추가했습니다. • 드라이브 문자 매핑 및 디스크 설치 문제를 수정했습니다.	
1.3.580	• 웹 요청에 기본 시스템 프록시 설정을 사용하도록 `Get-Metadata` 를 수정했습니다. • 디스크 설치에서 NVMe 특수 사례를 추가했습니다. • 사소한 문제를 수정했습니다.	

버전	세부 정보	릴리스 날짜
1.3.550	종료 없이 Sysprep을 활성화하는 `-NoShutdown` 옵션을 추가했습니다.	
1.3.540	사소한 문제를 수정했습니다.	
1.3.530	사소한 문제를 수정했습니다.	
1.3.521	사소한 문제를 수정했습니다.	
1.3.0	• 컴퓨터 이름 변경에 발생한 16진수 길이 문제를 수정했습니다. • 컴퓨터 이름 변경에 발생 가능한 재부팅 루프를 수정했습니다. • 월페이퍼(wallpaper) 설정 문제를 수정했습니다.	
1.2.0	• 설치된 운영 체제(OS)에 대한 정보가 EC2 시스템 로그에 표시되도록 업데이트합니다. • EC2 시스템 로그에 EC2Launch 및 SSM Agent 버전이 표시되도록 업데이트합니다. • 사소한 문제를 수정했습니다.	

버전	세부 정보	릴리스 날짜
1.1.2	• EC2 시스템 로그에 ENA 드라이버 정보가 표시되도록 업데이트합니다. • 기본 NIC 필터 로직에서 Hyper-V를 제외하도록 업데이트합니다. • KMS 정품 인증을 위한 레지스트리 키에 AWS KMS 서버와 포트를 추가했습니다. • 여러 사용자에 대한 월페이퍼(wallpaper)를 개선했습니다. • 영구 저장소에서 경로를 지우도록 업데이트합니다. • DNS 접미사 목록의 가용 영역에서 z를 제거하도록 업데이트합니다. • 사용자 데이터에서 <runAsLocalSystem> 태그로 문제를 해결하도록 업데이트합니다.	
1.1.1	최초 릴리스.	

EC2Config 서비스를 사용하여 EC2 레거시 Windows 운영 체제 인스턴스를 시작하는 동안 태스크를 수행합니다.

> **Note**
>
> EC2Config가 지원 종료 단계에 도달했습니다. 실행되는 운영 체제 버전은 Microsoft에서 더 이상 지원되지 않습니다. 최신 시작 에이전트로 업그레이드하는 것이 좋습니다.
> Windows Server 2022 이상 운영 체제용 최신 시작 에이전트는 EC2Config와 EC2Launch를 모두 대체하며 AWS Windows Server 2022 및 2025 AMI에 사전 설치되어 제공되는 EC2Launch v2입니다. 마이그레이션 도구를 사용하여 EC2Launch v2로 마이그레이션하거나, Windows Server 2016 및 2019에 에이전트를 수동으로 설치하고 구성할 수도 있습니다.

Windows Server 2016 이전의 Windows Server 버전용 Windows AMI는 EC2Config 서비스(EC2Config.exe)라는 선택적 서비스를 포함합니다. 시작 중에 인스턴스가 부팅되고 작업을 수행할 때와 인스턴스를 중지하거나 시작할 때마다 EC2Config가 시작됩니다. 또한 EC2Config는 요청 시 작업을 수행할 수 있습니다. 이러한 작업 중 일부는 자동으로 활성화되고, 나머지는 수동으로 활성화해야 합니다. 선택 사항이긴 하지만, 이 서비스는 다른 방식으로는 사용할 수 없는 고급 기능에 대한 액세스를 제공합니다. 이 서비스는 LocalSystem 계정에서 실행됩니다.

EC2Config 서비스에서는 재사용할 수 있는 사용자 정의된 Windows AMI를 생성하는 데 사용할 수 있는 Microsoft 도구인 Sysprep을 실행합니다. EC2Config가 Sysprep을 호출하면 Sysprep은 %ProgramFiles%\Amazon\EC2ConfigService\Settings 안의 파일을 사용하여 어느 작업을 수행할지 결정합니다. 이러한 파일은 EC2 서비스 속성 시스템 대화 상자를 사용하여 간접적으로 편집하거나, XML 편집기 또는 텍스트 편집기를 사용하여 직접적으로 편집할 수 있습니다. 그러나 Ec2 서비스 속성 시스템 대화 상자에서 사용할 수 없는 몇 가지 고급 설정이 있으며, 이러한 항목은 직접 편집해야 합니다.

설정을 업데이트한 후에 인스턴스에서 AMI를 생성하면 새로운 설정은 그 새로운 AMI에서 실행하는 모든 인스턴스에 적용됩니다. AMI 생성에 대한 자세한 내용은 Amazon EBS 지원 AMI 생성 섹션을 참조하세요.

EC2Config에서 설정 파일을 사용하여 해당 작업을 제어합니다. 이러한 설정 파일은 그래픽 도구를 사용하거나 XML 파일을 직접 편집하여 업데이트할 수 있습니다. 서비스 바이너리 및 추가 파일은 %ProgramFiles%\Amazon\EC2ConfigService 디렉터리에 저장됩니다.

내용

- EC2Config 및 AWS Systems Manager
- EC2Config 태스크
- EC2Config 설정 파일
- 최신 버전의 EC2Config 설치
- EC2Config 서비스의 .NET 프록시 설정 구성
- EC2 Windows 인스턴스의 시스템 대화 상자에서 EC2Config 서비스 속성 설정
- EC2Config 시작 에이전트와 관련된 문제 해결
- EC2Config 버전 기록

EC2Config 및 AWS Systems Manager

EC2Config 서비스는 2016년 11월 이전에 발표된 Windows Server 2016 이전 버전의 Windows Server용 AMI에서 생성된 인스턴스에 대한 Systems Manager 요청을 처리합니다.

2016년 11월 이후에 발표된 Windows Server 2016 이전 버전의 Windows Server용 AMI에서 생성된 인스턴스에는 EC2Config 서비스 및 SSM Agent가 포함되어 있습니다. EC2Config는 앞서 설명한 모든 작업을 수행하고 SSM Agent는 명령 실행 및 상태 관리자 같은 Systems Manager 기능에 대한 요청을 처리합니다.

Run Command를 사용하여 기존 인스턴스가 최신 버전의 EC2Config 서비스와 SSM Agent를 사용하도록 업그레이드할 수 있습니다. 자세한 내용은 AWS Systems Manager 사용 설명서에서 Run Command를 사용하여 SSM Agent 업데이트를 참조하세요.

EC2Config 태스크

EC2Config는 인스턴스가 처음 시작할 때 초기 시작 작업을 실행하고 이후에 그 작업을 비활성화합니다. 이런 작업을 다시 실행하기 위해서는 인스턴스를 종료하기 전에 이를 명시적으로 활성화하거나 Sysprep을 수동으로 실행하는 방법을 사용해야 합니다. 이 작업은 다음과 같습니다.

- 관리자 계정에 대한 무작위의 암호화된 암호 설정.
- Remote Desktop Connection(원격 데스크톱 연결)에 사용되는 호스트 인증서의 생성 및 설치.
- 파티션 처리되지 않은 공간을 포함시키기 위한 운영 시스템 파티션을 동적으로 확장.
- 지정된 사용자 데이터 실행(설치된 경우는 Cloud-Init도 실행). 사용자 데이터 지정에 대한 자세한 내용은 사용자 데이터 입력을 사용하여 EC2 인스턴스를 시작할 때 명령 실행 섹션을 참조하세요.

EC2Config는 다음 작업을 인스턴스가 시작할 때마다 실행합니다.

- 16진수 표기법으로 프라이빗 IP 주소와 일치하도록 호스트 이름 변경(이 작업은 기본적으로 비활성화되어 있으며 인스턴스가 시작할 때 실행하려면 활성화해야 함)
- AWS KMS(핵심 관리 서버) 구성, Windows 정품 인증 상태 확인, 필요한 경우 Windows 정품 인증을 활성화.
- 모든 Amazon EBS 볼륨 및 인스턴스 스토어 볼륨을 마운트하고 볼륨 이름을 드라이브 문자로 매핑합니다.
- 문제 해결을 돕기 위해 이벤트 로그 항목을 콘솔에 기록(이 작업은 기본적으로 비활성화되어 있으며 인스턴스가 시작할 때 실행될 수 있도록 활성화되어야 함)
- Windows가 준비된 상태를 콘솔에 기록.
- 단일 NIC 또는 복수의 NIC가 연결될 때 다음 IP 주소(`169.254.169.250`, `169.254.169.251`, `169.254.169.254`)를 활성화하기 위해 기본 네트워크 어댑터에 사용자 정의 라우팅을 추가합니다. 이 주소들은 Windows 정품 인증을 할 때와 인스턴스 메타데이터에 액세스할 때 사용됩니다.

> **Note**
>
> Windows OS가 IPv4를 사용하도록 구성된 경우 이러한 IPv4 링크 로컬 주소를 사용할 수 있습니다. Windows OS에서 IPv4 네트워크 프로토콜 스택이 비활성화되어 있고 IPv6을 대신 사용하는 경우 `169.254.169.250` 및 `169.254.169.251` 대신 `[fd00:ec2::250]`을 추가합니다. 그런 다음 `169.254.169.254` 대신 `[fd00:ec2::254]`를 추가합니다.

EC2Config는 다음 작업을 사용자가 로그인할 때마다 실행합니다.

- 바탕 화면 배경에 월페이퍼(wallpaper) 정보 표시

인스턴스가 실행 중일 때 사용자는 EC2Config가 다음 작업을 필요할 때 수행하도록 요청할 수 있습니다.

- AMI을 인스턴스에서 생성할 수 있도록 Sysprep을 실행하고 인스턴스를 종료. 자세한 내용은 Windows Sysprep을 사용하여 Amazon EC2 AMI 생성 섹션을 참조하세요.

EC2Config 설정 파일

본 설정 파일은 EC2Config 서비스의 작동을 제어합니다. 이 파일은 `C:\Program Files\Amazon\Ec2ConfigService\Settings` 디렉터리에 위치합니다.

- `ActivationSettings.xml` 핵심 관리 서버()를 사용해서 제품의 정품 인증을 제어합니다.AWS KMS
- `AWS.EC2.Windows.CloudWatch.json` - CloudWatch에 전송할 성능 카운터 및 CloudWatch Logs에 전송할 로그가 무엇인지를 제어합니다.
- `BundleConfig.xml` - EC2Config가 AMI 생성을 위한 인스턴스 스토어 지원 인스턴스의 준비 방법을 제어합니다.
- `Config.xml` - 기본 설정을 제어합니다.
- `DriveLetterConfig.xml` - 드라이브 문자 매핑을 제어합니다.
- `EventLogConfig.xml` - 인스턴스가 부팅 중인 동안 콘솔에 표시되는 이벤트 로그 정보를 제어합니다.
- `WallpaperSettings.xml` - 바탕 화면 배경에 표시되는 정보를 제어합니다.

ActivationSettings.xml

이 파일은 제품 정품 인증을 제어하는 설정을 포함하고 있습니다. Windows가 부팅될 때 EC2Config 서비스는 Windows가 이미 정품 인증되었는지를 검사합니다. Windows가 아직 정품 인증되지 않은 경우, EC2Config 서비스는 지정된 AWS KMS 서버를 검색해서 Windows 정품 인증을 시도합니다.

- `SetAutodiscover` - AWS KMS를 자동으로 감지할지 여부를 지정합니다.
- `TargetKMSServer` 의 프라이빗 IP 주소를 저장합니다.AWS KMS AWS KMS은(는) 인스턴스와 동일한 리전에 위치해 있어야 합니다.
- `DiscoverFromZone` - 지정된 DNS 영역에서 AWS KMS 서버를 검색합니다.
- `ReadFromUserData` - UserData에서 AWS KMS 서버를 가져옵니다.
- `LegacySearchZones` - 지정된 DNS 영역에서 AWS KMS 서버를 검색합니다.
- `DoActivate` - 해당 섹션의 지정된 설정을 사용해서 정품 인증을 시도합니다. 이 값은 `true` 또는 `false`일 수 있습니다.
- `LogResultToConsole` - 결과를 콘솔에 표시합니다.

BundleConfig.xml

이 파일은 EC2Config가 AMI 생성을 위한 인스턴스 스토어 지원 인스턴스의 준비 방법을 제어하는 설정을 포함하고 있습니다.

- `AutoSysprep`- Sysprep을 자동적으로 사용할지 여부를 지정합니다. `Yes`로 값을 변경해서 Sysprep을 사용하도록 합니다.
- `SetRDPCertificate` - 자체 서명된 인증서를 원격 데스크톱 서버로 설정합니다. 이를 통해 안전하게 RDP를 인스턴스에 연결시킬 수 있습니다. 새 인스턴스가 인증서를 보유해야 하는 경우는 이 값을 `Yes`로 변경합니다.

  Windows Server 2016 이전의 운영 체제 버전을 사용하는 인스턴스는 자신의 고유한 인증서를 생성할 수 있으므로 이 설정은 이 인스턴스에서는 사용되지 않습니다.
- `SetPasswordAfterSysprep` - 새로 실행된 인스턴스에 무작위 암호를 설정하고 이를 사용자 실행 키로 암호화하고 암호화된 암호를 콘솔에 출력합니다. 새 인스턴스를 무작위의 암호화된 암호로 설정하지 않아야 하는 경우는 설정의 값을 `No`로 변경합니다.

Config.xml

플러그인

- Ec2SetPassword - 인스턴스를 실행할 때마다 무작위의 암호화된 암호를 생성합니다. 이 기능은 첫 실행 후 기본적으로 비활성화되어 해당 인스턴스의 재부팅으로 인해 사용자가 설정한 암호가 변경되지 않도록 합니다. 이 설정을 Enabled로 변경하면 인스턴스를 실행할 때마다 계속 새로운 암호를 생성합니다.

  이 설정은 인스턴스에서 AMI를 생성하려는 경우 필요할 수 있습니다.
- Ec2SetComputerName - 인스턴스의 호스트 이름을 인스턴스의 IP 주소에 기반한 고유의 이름으로 변경하고 인스턴스를 재부팅합니다. 자신의 호스트 이름을 설정하거나 기존 호스트 이름이 변경되는 것을 방지하려면 이 설정을 비활성화해야 합니다.
- Ec2InitializeDrives - 시작 시에 모든 볼륨을 초기화하고 포맷합니다. 이 기능은 기본적으로 활성화되어 있습니다.
- Ec2EventLog - 콘솔에서 이벤트 로그 항목을 표시합니다. 기본적으로 시스템 이벤트 로그에서 가장 최근의 오류 항목 3개가 표시됩니다. 표시할 이벤트 로그 항목을 지정하려면 EventLogConfig.xml 디렉터리에 있는 EC2ConfigService\Settings 파일을 편집합니다. 아 파일의 설정에 대한 자세한 내용은 Eventlog Key 섹션을 참조하세요.
- Ec2ConfigureRDP - 인스턴스에 자체 서명된 인증서를 설정해서 사용자가 원격 데스크톱을 사용하여 안전하게 인스턴스에 액세스할 수 있도록 합니다. Windows Server 2016 이전의 운영 체제 버전을 사용하는 인스턴스는 자신의 고유한 인증서를 생성할 수 있으므로 이 설정은 이 인스턴스에서는 사용되지 않습니다.
- Ec2OutputRDPCert - 원격 데스크톱 인증서 정보를 사용자가 그 지문에 대해 검증할 수 있도록 상기 정보를 콘솔로 표시합니다.
- Ec2SetDriveLetter - 사용자가 지정한 설정에 기반하는 마운트된 볼륨의 드라이브 문자를 설정합니다. 기본적으로 Amazon EBS 볼륨이 인스턴스에 연결되면 인스턴스의 드라이브 문자를 사용하여 상기 볼륨을 마운트할 수 있습니다. 드라이브 문자 매핑을 지정하려면 DriveLetterConfig.xml에 있는 EC2ConfigService\Settings 파일을 편집합니다.
- Ec2WindowsActivate - 플러그인이 Windows 정품 인증을 처리합니다. Windows가 정품 인증되었는지 확인합니다. 인증되지 않은 경우 AWS KMS 클라이언트 설정을 업데이트한 후 Windows를 정품 인증합니다.

  AWS KMS 설정을 변경하려면 ActivationSettings.xml 디렉터리에 위치한 EC2ConfigService\Settings 파일을 편집합니다.
- Ec2DynamicBootVolumeSize - 파티션 처리되지 않은 모든 공간을 포함하도록 디스크 0/볼륨 0을 확장합니다.
- Ec2HandleUserData—Sysprep이 실행된 후에 인스턴스를 처음 실행했을 때 사용자가 생성한 스크립트를 생성하여 실행시킵니다. 스크립트 태그에 둘러싸인 명령은 배치 파일로 저장되고,

PowerShell 태그로 둘러싸인 명령은 .ps1 파일로 저장됩니다. 이는 Ec2 서비스 속성 시스템 대화 상자의 사용자 데이터 확인란에 해당되는 기능입니다.

- `Ec2ElasticGpuSetup` - 인스턴스가 탄력적 GPU와 연결되어 있으면 탄력적 GPU 소프트웨어 패키지를 설치합니다.
- `Ec2FeatureLogging` - Windows 기능 설치 및 이에 상응하는 서비스 상태를 콘솔로 전송합니다. Microsoft Hyper-V 기능 및 이에 상응하는 vmms 서비스에 대해서만 지원됩니다.

글로벌 설정

- `ManageShutdown` - 인스턴스 스토어 지원 AMI에서 실행한 인스턴스가 Sysprep 실행 중에 종료되지 않도록 합니다.
- `SetDnsSuffixList` - Amazon EC2에 대한 네트워크 어댑터의 DNS 접미사를 설정합니다. 이를 통해 정규화된 도메인 이름을 제공하지 않고도 Amazon EC2에서 실행 중인 서버의 DNS 확인을 할 수 있습니다.

> **Note**
>
> 이렇게 하면 다음 도메인에 대한 DNS 접미사 조회가 추가되고 다른 표준 접미사가 구성됩니다. 시작 에이전트가 DNS 접미사를 설정하는 방법에 대한 자세한 내용은 EC2 Windows 시작 에이전트용 DNS 접미사 구성 섹션을 참조하세요.
>
> ```
> region.ec2-utilities.amazonaws.com
> ```

- `WaitForMetaDataAvailable` - 부팅이 시작되기 전에 메타데이터의 액세스와 네트워크의 사용이 가능해질 때까지 EC2Config 서비스가 대기하도록 설정합니다. 이를 통해 EC2Config가 정품 인증 및 기타 플러그인에 대한 메타데이터를 얻을 수 있는지를 검사할 수 있습니다.
- `ShouldAddRoutes` - 복수 NIC가 연결될 때 다음 IP 주소(169.254.169.250, 169.254.169.251, 169.254.169.254)를 활성화하기 위해 기본 네트워크 어댑터에 사용자 지정 라우팅 추가. 이 주소들은 Windows 정품 인증을 할 때와 인스턴스 메타데이터에 액세스할 때 사용됩니다.
- `RemoveCredentialsfromSyspreponStartup` - 다음에 서비스가 시작할 때 `Sysprep.xml`에서 관리자 암호를 제거합니다. 암호가 유지되도록 하려면 이 설정을 편집합니다.

DriveLetterConfig.xml

이 파일은 드라이브 문자 매핑을 제어하는 설정을 포함하고 있습니다. 기본적으로 볼륨은 사용 가능한 모든 드라이브 문자에 매핑할 수 있습니다. 다음과 같은 특정 드라이브에 볼륨을 마운트할 수 있습니다.

```
<?xml version="1.0" standalone="yes"?>
<DriveLetterMapping>
  <Mapping>
    <VolumeName></VolumeName>
    <DriveLetter></DriveLetter>
  </Mapping>
  . . .
  <Mapping>
    <VolumeName></VolumeName>
    <DriveLetter></DriveLetter>
  </Mapping>
</DriveLetterMapping>
```

- `VolumeName` - 볼륨 레이블. 예: *`My Volume`*. 인스턴스 스토리지 볼륨에 대한 매핑을 지정하려면 라벨 `Temporary Storage X`를 사용합니다(`X`는 0-25 범위 내 숫자).
- `DriveLetter` - 드라이브 문자. 예: *`M:`*. 그러나 드라이브 문자가 이미 사용 중인 경우 드라이브 문자 매핑은 실패하게 됩니다.

EventLogConfig.xml

이 파일은 인스턴스가 부팅 중인 동안 콘솔에 표시되는 이벤트 로그 정보를 제어하는 설정을 포함하고 있습니다. 기본적으로 시스템 이벤트 로그에서 가장 최근의 오류 항목 3개가 표시됩니다.

- `Category` - 모니터링할 이벤트 로그 키.
- `ErrorType` - 이벤트 유형(예:`Error` ,`Warning` ,`Information`)
- `NumEntries` - 해당 카테고리에 대해 저장된 이벤트 수.
- `LastMessageTime` - 동일한 메시지가 반복적으로 푸시되는 것을 방지하기 위해 서비스는 이 값을 메시지를 푸시할 때마다 업데이트합니다.
- `AppName` - 이벤트를 기록한 이벤트 소스 또는 애플리케이션.

WallpaperSettings.xml

이 파일은 바탕 화면 배경에 표시되는 정보를 제어하는 설정을 포함하고 있습니다. 기본적으로 다음 정보가 표시됩니다.

- `Hostname` - 컴퓨터 이름을 표시합니다.
- `Instance ID` - 인스턴스의 ID를 표시합니다.
- `Public IP Address` - 인스턴스의 퍼블릭 IP 주소를 표시합니다.
- `Private IP Address` - 인스턴스의 프라이빗 IP 주소를 표시합니다.
- `Availability Zone` - 인스턴스가 실행 중인 가용 영역을 표시합니다.
- `Instance Size` - 인스턴스의 유형을 표시합니다.
- `Architecture` - PROCESSOR_ARCHITECTURE 환경 변수의 설정을 표시합니다.

기본적으로 표시되는 정보를 해당 항목을 삭제하여 제거할 수 있습니다. 다음과 같이 표시할 추가 인스턴스 메타데이터를 추가할 수 있습니다.

```
<WallpaperInformation>
  <name>display_name</name>
  <source>metadata</source>
  <identifier>meta-data/path</identifier>
</WallpaperInformation>
```

다음과 같이 표시할 시스템 환경 변수를 추가할 수 있습니다.

```
<WallpaperInformation>
  <name>display_name</name>
  <source>EnvironmentVariable</source>
  <identifier>variable-name</identifier>
</WallpaperInformation>
```

InitializeDrivesSettings.xml

이 파일에는 EC2Config의 드라이브 초기화 방식을 제어하는 설정이 포함되어 있습니다.

기본적으로 EC2Config는 운영 체제를 통해 온라인 상태가 되지 않은 드라이브를 초기화합니다. 다음과 같이 플러그인을 사용자 지정할 수 있습니다.

```
<InitializeDrivesSettings>
```

```
    <SettingsGroup>setting</SettingsGroup>
</InitializeDrivesSettings>
```

설정 그룹을 사용하여 드라이브 초기화 방식을 지정합니다:

FormatWithTRIM

드라이브 포맷 시 TRIM 명령을 활성화합니다. 드라이브가 포맷되고 초기화된 후 시스템이 TRIM 구성을 복원합니다.

EC2Config 버전 3.18부터 기본적으로 디스크 포맷 작업 중에 TRIM 명령이 비활성화됩니다. 이를 통해 포맷 시간이 단축되었습니다. 이 설정을 사용하여 EC2Config 버전 3.18 이상에서 디스크 포맷 작업 중에 TRIM을 활성화합니다.

FormatWithoutTRIM

드라이브 포맷 시 TRIM 명령을 비활성화하고 Windows 포맷 시간을 단축합니다. 드라이브가 포맷되고 초기화된 후 시스템이 TRIM 구성을 복원합니다.

DisableInitializeDrives

새 드라이브의 포맷을 비활성화합니다. 이 설정을 사용하여 드라이브를 수동으로 초기화합니다.

최신 버전의 EC2Config 설치

> **Note**
>
> Windows Server 2022 이상 운영 체제용 최신 시작 에이전트는 EC2Config와 EC2Launch를 모두 대체하는 EC2Launch v2입니다. EC2Launch v2는 AWS Windows Server 2022 및 2025 AMI에 사전 설치되어 있습니다. 마이그레이션 도구를 사용하여 EC2Launch v2로 마이그레이션하거나 Windows Server 2016 및 2019에 에이전트를 수동으로 설치하고 구성할 수도 있습니다.

EC2Config 업데이트 알림을 받는 방법에 대한 자세한 내용은 EC2 Windows 시작 에이전트 알림 구독 섹션을 참조하세요. 각 버전의 변경 사항에 대한 자세한 내용은 EC2Config 버전 기록 섹션을 참조하세요.

시작하기 전에

- .NET framework 3.5 SP1 이상이 설치되어 있는지 확인합니다.

- 기본적으로 설치는 사용자의 설정 파일을 기본 설정 파일로 교체하고 설치가 완료되면 EC2Config 서비스를 재시작합니다. EC2Config 서비스 설정을 변경한 경우 `config.xml` 디렉터리에서 `%Program Files%\Amazon\Ec2ConfigService\Settings` 파일을 복사합니다. EC2Config 서비스를 업데이트한 후 이 파일을 복원하여 구성 변경을 유지할 수 있습니다.

EC2Config 버전 확인

다음 절차를 사용하여 인스턴스에 설치된 EC2Config의 버전을 확인합니다.

설치된 EC2Config 버전을 확인하려면

1. AMI에서 인스턴스를 실행해서 여기에 연결합니다.
2. 제어판에서 프로그램 및 기능을 선택합니다.
3. 설치된 프로그램 목록에서 `Ec2ConfigService`를 찾습니다. 버전 번호가 버전 열에 표시됩니다.

EC2Config 업데이트

다음 절차를 사용하여 인스턴스에 최신 버전의 EC2Config를 다운로드하고 설치합니다.

최신 버전의 EC2Config를 다운로드하고 설치하려면

1. EC2Config 설치 관리자를 다운로드하고 압축을 풉니다.
2. `EC2Install.exe`를 실행합니다. 전체 옵션 목록을 보려면 `EC2Install` 옵션을 포함해 `/?` 파일을 실행합니다. 기본적으로 설치는 프롬프트를 표시합니다. 프롬프트 없이 명령을 실행하려면 `/quiet` 옵션을 사용합니다.

   > **Important**
   >
   > 저장한 `config.xml` 파일의 사용자 지정 설정을 그대로 유지하려면 `EC2Install` 옵션으로 `/norestart` 파일을 실행하고 사용자의 설정을 복원한 다음에 EC2Config 서비스를 수동으로 재시작합니다.

3. EC2Config 버전 4.0 이상을 실행하는 경우 Microsoft Services 스냅인 인스턴스에서 SSM Agent를 재시작해야 합니다.

> **Note**
>
> 인스턴스를 재부팅하거나 중지하고 시작할 때까지 업데이트된 EC2Config 버전 정보는 인스턴스 시스템 로그 또는 Trusted Advisor 검사에 나타나지 않습니다.

PowerShell을 사용하여 최신 버전의 EC2Config 다운로드하여 설치하기

PowerShell을 사용하여 최신 버전의 EC2Config를 다운로드하고 압축 해제하고 설치하려면 PowerShell 창에서 다음 명령을 실행합니다.

```
$Url = "https://s3.amazonaws.com/ec2-downloads-windows/EC2Config/EC2Install.zip"
$DownloadZipFile = "$env:USERPROFILE\Desktop\" + $(Split-Path -Path $Url -Leaf)
$ExtractPath = "$env:USERPROFILE\Desktop\"
Invoke-WebRequest -Uri $Url -OutFile $DownloadZipFile
$ExtractShell = New-Object -ComObject Shell.Application
$ExtractFiles = $ExtractShell.Namespace($DownloadZipFile).Items()
$ExtractShell.NameSpace($ExtractPath).CopyHere($ExtractFiles)
Start-Process $ExtractPath
Start-Process `
    -FilePath $env:USERPROFILE\Desktop\EC2Install.exe `
    -ArgumentList "/S"
```

> **Note**
>
> Windows Server 2016 또는 이전 버전을 사용 중이고 파일을 다운로드할 때 오류가 발생하는 경우 PowerShell 터미널에서 TLS 1.2를 활성화해야 할 수 있습니다. 다음 명령을 사용하여 현재 PowerShell 세션에 대해 TLS 1.2를 활성화한 다음 다시 시도해보세요.
>
> ```
> [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
> ```

`C:\Program Files\Amazon\`에서 `Ec2ConfigService` 디렉터리를 확인하여 설치를 확인입니다.

EC2Config 서비스의 .NET 프록시 설정 구성

AWS SDK for .NET, `system.net` 요소 또는 Microsoft Group Policy 및 Internet Explorer를 사용하여 프록시를 통해 통신하도록 EC2Config 서비스를 구성할 수 있습니다. AWS SDK for .NET을 사용하면 로그인 자격 증명을 지정할 수 있어 좋습니다.

메서드

- AWS SDK for .NET을 사용하여 프록시 설정 구성(권장)
- system.net 요소를 사용하여 프록시 설정 구성
- Microsoft Group Policy 및 Microsoft Internet Explorer를 사용하여 프록시 설정 구성

AWS SDK for .NET을 사용하여 프록시 설정 구성(권장)

`proxy` 파일에 `Ec2Config.exe.config` 요소를 지정하여 EC2Config 서비스에 대한 프록시 설정을 구성할 수 있습니다. 자세한 내용은 AWS SDK for .NET에 대한 구성 파일 참조를 참조하세요.

Ec2Config.exe.config에서 프록시 요소를 지정하는 방법

1. EC2Config 서비스가 프록시를 통해 통신하게 하려는 인스턴스에서 `Ec2Config.exe.config` 파일을 편집합니다. 기본적으로 이 파일은 `%ProgramFiles%\Amazon\Ec2ConfigService` 디렉터리에 위치합니다.
2. 다음 `aws` 요소를 `configSections`에 추가합니다. 기존의 `sectionGroups`에 추가하면 안 됩니다.

   EC2Config 버전 3.17 이하의 경우

   ```
   <configSections>
      <section name="aws" type="Amazon.AWSSection, AWSSDK"/>
   </configSections>
   ```

   EC2Config 버전 3.18 이상의 경우

   ```
   <configSections>
        <section name="aws" type="Amazon.AWSSection, AWSSDK.Core"/>
   </configSections>
   ```

3. 다음 `aws` 요소를 `Ec2Config.exe.config` 파일에 추가합니다.

   ```
   <aws>
      <proxy
        host="string value"
        port="string value"
        username="string value"
        password="string value" />
   </aws>
   ```

4. 변경 내용을 저장합니다.

system.net 요소를 사용하여 프록시 설정 구성

`system.net` 파일의 `Ec2Config.exe.config` 요소에 프록시 설정을 지정할 수 있습니다. 자세한 내용은 defaultProxy 요소(네트워크 설정) 섹션을 참조하세요.

Ec2Config.exe.config에서 system.net 요소를 지정하는 방법

1. EC2Config 서비스가 프록시를 통해 통신하게 하려는 인스턴스에서 `Ec2Config.exe.config` 파일을 편집합니다. 기본적으로 이 파일은 `%ProgramFiles%\Amazon\Ec2ConfigService` 디렉터리에 위치합니다.
2. `defaultProxy` 항목을 `system.net`에 추가합니다. 자세한 내용은 defaultProxy 요소(네트워크 설정) 섹션을 참조하세요.

   예를 들어, 다음과 같이 구성하면 프록시를 우회하는 메타데이터 및 라이선스 트래픽을 제외하고 모든 트래픽이 현재 Internet Explorer에 구성된 프록시를 사용하도록 라우팅됩니다.

   ```
   <defaultProxy>
       <proxy usesystemdefault="true" />
       <bypasslist>
           <add address="169.254.169.250" />
           <add address="169.254.169.251" />
           <add address="169.254.169.254" />
           <add address="[fd00:ec2::250]" />
           <add address="[fd00:ec2::254]" />
       </bypasslist>
   </defaultProxy>
   ```

3. 변경 내용을 저장합니다.

Microsoft Group Policy 및 Microsoft Internet Explorer를 사용하여 프록시 설정 구성

EC2Config 서비스는 로컬 시스템 사용자 계정으로 실행됩니다. 인스턴스에서 Group Policy 설정을 변경한 후 Internet Explorer에서 이 계정에 대한 인스턴스 전역 프록시 설정을 지정할 수 있습니다.

Group Policy 및 Internet Explorer를 사용하여 프록시 설정을 구성하려면

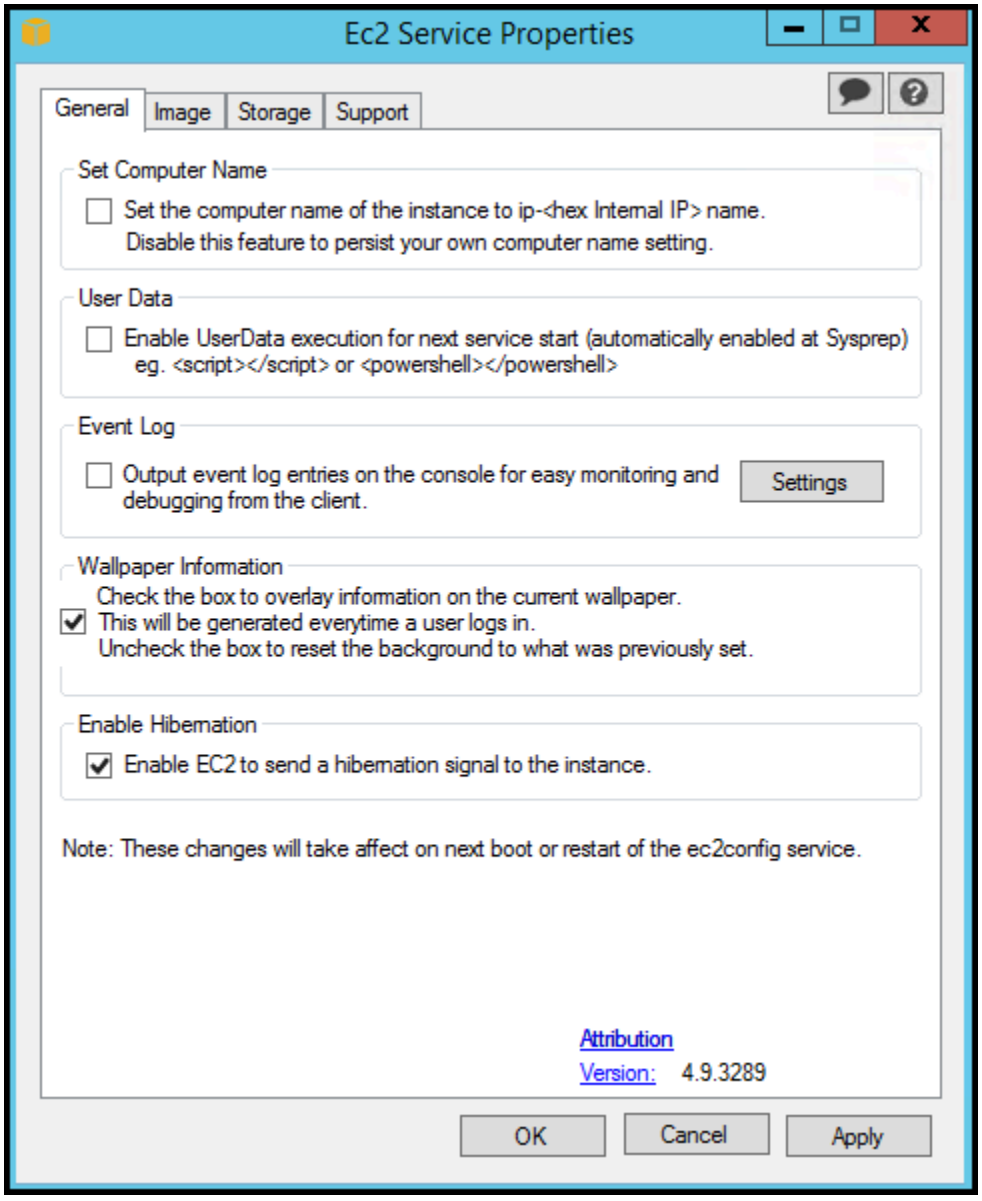
1. EC2Config 서비스를 프록시를 통해 통신하게 하려는 인스턴스에서 관리자 자격으로 명령 프롬프트를 열고 **`gpedit.msc`**를 입력한 후 Enter를 누릅니다.

2. 로컬 그룹 정책 편집기의 로컬 컴퓨터 정책에서 컴퓨터 구성, 관리 템플릿, Windows 구성 요소, Internet Explorer를 선택합니다.
3. 오른쪽 창에서 사용자 단위보다는 컴퓨터 단위로 프록시 설정 만들기를 선택한 후 정책 설정 편집을 선택합니다.
4. 사용을 선택한 후 적용을 선택합니다.
5. Internet Explorer를 열고 도구 버튼을 선택합니다.
6. 인터넷 옵션을 선택한 후 연결 탭을 선택합니다.
7. LAN 설정을 선택합니다.
8. 프록시 서버에서 LAN에 프록시 서버 사용 옵션을 선택합니다.
9. 주소와 포트 정보를 지정한 후 확인을 선택합니다.

EC2 Windows 인스턴스의 시스템 대화 상자에서 EC2Config 서비스 속성 설정

다음 절차는 EC2 서비스 속성 시스템 대화 상자를 사용하여 설정을 활성화 또는 비활성화하는 방법을 설명합니다.

1. 실행을 시작해서 Windows 인스턴스에 연결합니다.
2. 시작 메뉴에서 모든 프로그램을 클릭하고 EC2ConfigService Settings를 클릭합니다.



3. EC2 서비스 속성 시스템 대화 상자의 일반 탭에서 다음과 같은 설정을 활성화하거나 비활성화할 수 있습니다.

   컴퓨터 이름 설정

   이 설정이 활성화된 경우(기본 설정상 비활성화되어 있음), 매 부팅마다 호스트 이름은 현재 IP 주소와 비교됩니다. 호스트 이름과 IP 주소가 일치하지 않는 경우, 호스트 이름은 내부 IP 주소를 포함하도록 재설정되며 이후 시스템은 재부팅되어 새로운 호스트 이름을 가지게 됩니다. 자신의 호스트 이름을 설정하거나 기존 호스트 이름이 변경되는 것을 방지하려면 이 설정을 활성화하지 마세요.

사용자 데이터

사용자 데이터 실행을 통해 인스턴스 메타데이터에서 스크립트를 지정할 수 있습니다. 기본적으로 이러한 스크립트는 초기 실행 중에 실행됩니다. 또한 다음 번에 인스턴스를 재부팅하거나 시작할 때 또는 매번 인스턴스를 재부팅하거나 시작할 때마다 실행하도록 구성할 수도 있습니다.

크기가 큰 스크립트가 있는 경우, 사용자 데이터를 사용해서 스크립트를 다운로드하고 이를 실행하는 것이 권장됩니다.

자세한 내용은 사용자 데이터 실행 섹션을 참조하세요.

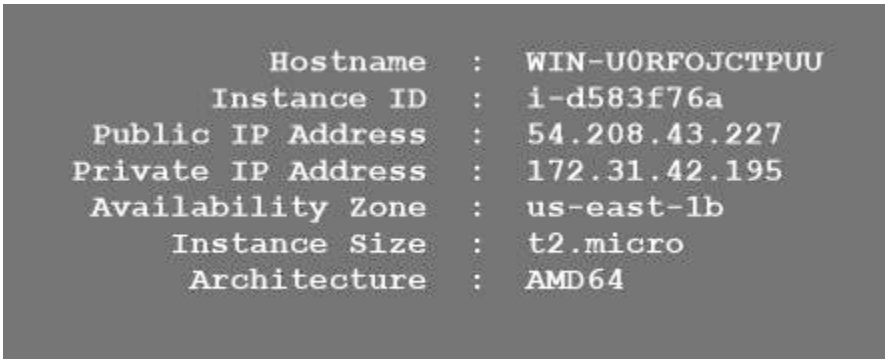
Event Log(이벤트 로그)

이 설정을 사용해서 모니터링 및 디버깅을 손쉽게 할 수 있도록 부팅 동안 콘솔의 이벤트 로그 항목을 표시합니다.

설정을 클릭해서 콘솔에 전송되는 로그 항목에 대한 필터를 지정합니다. 기본 필터는 시스템 이벤트 로그에서 콘솔로 가장 최근의 오류 항목 3개를 전송합니다.

월페이퍼 정보

이 설정을 사용해서 바탕 화면 배경에 시스템 정보를 표시합니다. 다음은 바탕 화면 배경에 표시되는 정보의 예입니다.

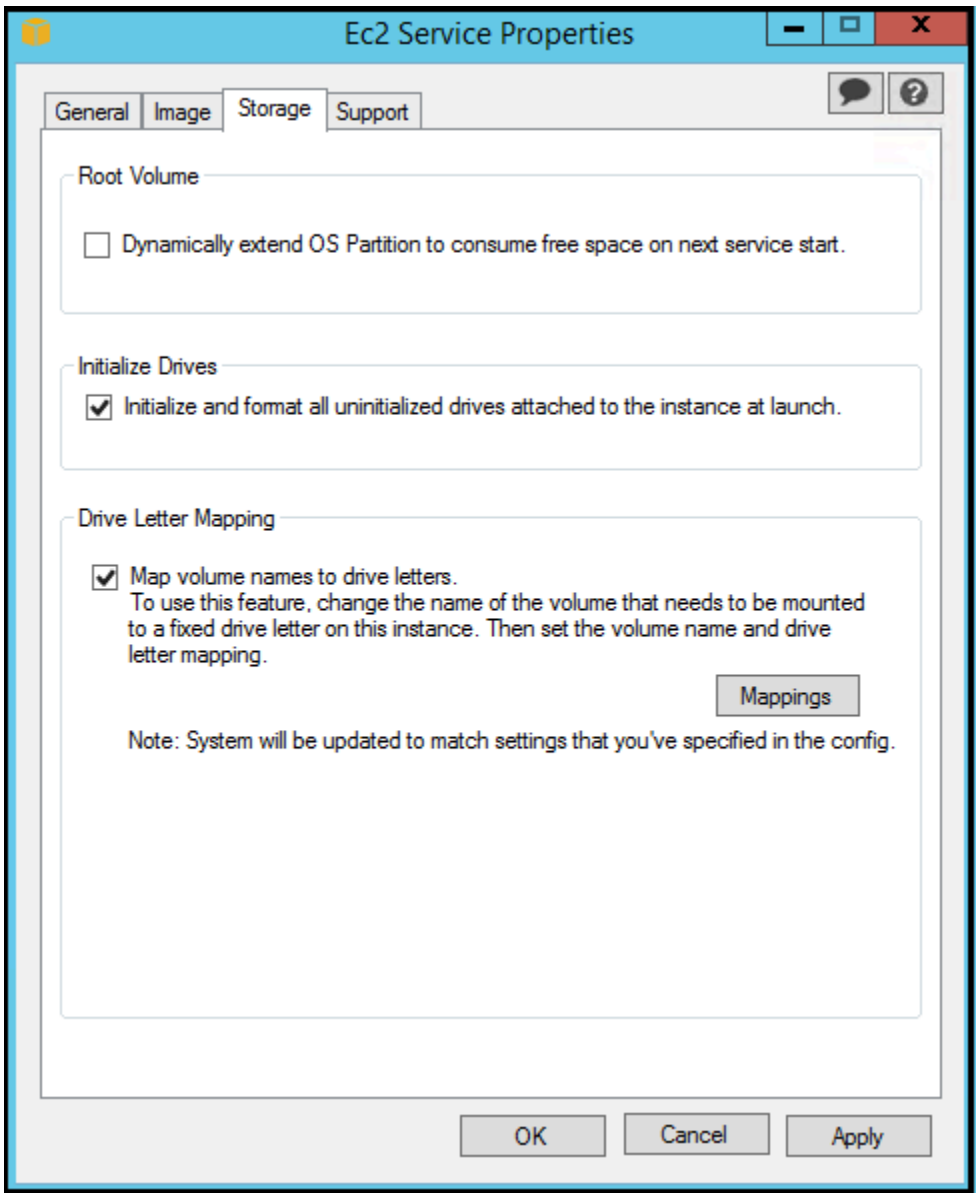


바탕 화면 배경에 표시되는 정보는 설정 파일 `EC2ConfigService\Settings\WallpaperSettings.xml`에 의해 제어됩니다.

최대 절전 모드 활성화

EC2에서 최대 절전 모드를 수행하도록 운영 체제에 신호를 보낼 수 있도록 하려면 이 설정을 사용합니다.

4. 스토리지 탭을 클릭합니다. 다음 설정을 활성화 또는 비활성활 수 있습니다.



Root Volume(루트 볼륨)

이 설정은 파티션 처리되지 않은 공간을 모두 포함하도록 동적으로 디스크 0/볼륨 0을 확장합니다. 이는 인스턴스가 사용자 설정 크기를 가진 루트 디바이스 볼륨에서 부팅될 때 유용할 수 있습니다.

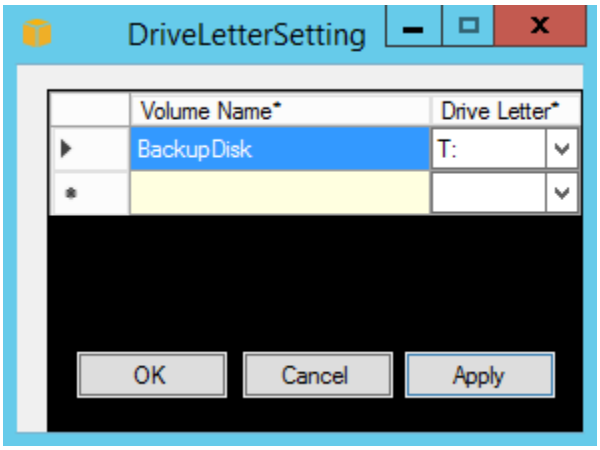
드라이브 초기화

이 설정은 시작 동안 인스턴스에 연결된 모든 볼륨을 포맷하고 마운트합니다.

드라이브 문자 매핑

시스템은 인스턴스에 연결된 볼륨을 드라이브 문자로 매핑합니다. Amazon EBS 볼륨의 경우, 기본 설정은 D:부터 Z:까지 드라이브 문자를 배정합니다. 인스턴스 스토어 볼륨의 경우, 기본값은 드라이버에 따라 다릅니다. AWS PV 드라이버 및 Citrix PV 드라이버는 인스턴스 스토어 볼륨에 Z:부터 A:까지의 드라이브 문자를 할당합니다. Red Hat 드라이버는 인스턴스 스토어 볼륨에 D:부터 Z:까지 드라이브 문자를 배정합니다.

볼륨에 대한 드라이브 문자를 선택하려면 매핑(Mappings)을 클릭합니다. DriveLetterSetting 대화 상자에서 각 볼륨에 대한 볼륨 이름(Volume Name)과 드라이브 문자(Drive Letter) 값을 지정하고 적용을 클릭한 후 확인을 클릭합니다. 사용할 가능성 있는 드라이브 문자와 충돌하지 않는 드라이브 문자(예: 알파벳 중간 위치 정도에 있는 문자)를 선택하는 것이 권장됩니다.



드라이브 문자 매핑을 지정하고 지정한 볼륨 이름과 동일한 라벨에 볼륨을 연결합니다. 그러나 드라이브 문자가 이미 사용 중인 경우 드라이브 문자 매핑은 실패하게 됩니다. 드라이브 문자 매핑을 지정한 경우 EC2Config는 이미 마운트된 볼륨의 드라이브 문자를 변경하지는 않는다는 점을 알아두시기 바랍니다.

5. 설정을 저장하고 나중에 작업을 계속하려면 확인을 클릭하여 EC2 서비스 속성 시스템 대화 상자를 닫습니다. 인스턴스의 사용자 지정을 완료하고 그 인스턴스에서 AMI를 생성하고 싶다면 Windows Sysprep을 사용하여 Amazon EC2 AMI 생성 섹션을 참조하세요.

EC2Config 시작 에이전트와 관련된 문제 해결

다음 정보는 EC2Config 서비스와 관련된 문제를 해결하는 데 도움이 될 수 있습니다.

접속 불가 인스턴스에서 EC2Config 업데이트

원격 데스크톱을 사용하여 액세스할 수 없는 Windows Server 인스턴스에서 EC2Config 서비스를 업데이트하려면 다음 절차를 사용합니다.

연결할 수 없는 Amazon EBS 지원 Windows 인스턴스에서 EC2Config 업데이트 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 문제가 발생한 인스턴스를 찾습니다. 인스턴스를 선택하고 [인스턴스 상태(Instance state)]를 선택한 다음 [인스턴스 중지(Stop instance)]를 선택합니다.

   > **Warning**
   >
   > 인스턴스를 중지하면 인스턴스 스토어 볼륨의 데이터가 삭제됩니다. 인스턴스 스토어 볼륨의 데이터를 유지하려면 영구 스토리지에 백업하세요.

4. [인스턴스 시작(Launch instances)]을 선택하고 문제가 발생한 인스턴스와 동일한 가용 영역에 임시 `t2.micro` 인스턴스를 생성합니다. 문제가 발생한 인스턴스를 시작하는 데 사용한 것과 다른 AMI를 사용합니다.

   > **Important**
   >
   > 문제가 발생한 인스턴스와 동일한 가용 영역에서 인스턴스를 생성하지 않는 경우에는 문제가 발생한 인스턴스의 루트 볼륨을 새 인스턴스에 연결할 수 없습니다.

5. EC2 콘솔에서 볼륨을 선택합니다.
6. 문제가 발생한 인스턴스의 루트 볼륨을 찾습니다. 볼륨을 분리하고 이전에 생성한 임시 인스턴스에 볼륨을 연결합니다. 기본 디바이스 이름(xvdf)으로 연결합니다.
7. 원격 데스크톱을 사용하여 임시 인스턴스에 연결한 후 디스크 관리 유틸리티를 사용하여 볼륨을 사용할 수 있도록 지정합니다.
8. 최신 버전의 EC2Config 서비스를 다운로드합니다. 연결한 드라이브의 `.zip` 디렉터리에 `Temp` 파일의 압축을 풉니다.
9. 임시 인스턴스에서 실행 대화 상자를 열고 **`regedit`**를 입력한 다음 Enter 키를 누릅니다.
10. 를 선택합니다`HKEY_LOCAL_MACHINE` 파일 메뉴에서 Hive 로드를 선택합니다. 드라이브를 선택한 다음 해당 드라이브로 이동하여 `Windows\System32\config\SOFTWARE` 파일을 엽니다. 메시지가 나타나면 키 이름을 지정합니다.

11. 방금 로드한 키를 선택하고 Microsoft\Windows\CurrentVersion 경로로 이동합니다. RunOnce 키를 선택합니다. 이 키가 없는 경우 컨텍스트(오른쪽 클릭) 메뉴에서 CurrentVersion을 선택하고 새로 생성을 선택한 다음 키를 선택합니다. 키 이름을 RunOnce로 지정합니다.
12. 컨텍스트(오른쪽 클릭) 메뉴에서 RunOnce 키를 선택하고 새로 생성을 선택한 다음 문자열 값(String Value)을 선택합니다. Ec2Install을 이름으로, C:\Temp\Ec2Install.exe /quiet를 데이터로 입력합니다.
13. HKEY_LOCAL_MACHINE*specified key name*\Microsoft\Windows NT\CurrentVersion\Winlogon 키를 선택합니다. 컨텍스트(오른쪽 클릭) 메뉴에서 새로 생성을 선택한 다음 문자열 값(String Value)을 선택합니다. **AutoAdminLogon**을 이름으로, **1**를 값 데이터로 입력합니다.
14. HKEY_LOCAL_MACHINE*specified key name*\Microsoft\Windows NT\CurrentVersion\Winlogon> 키를 선택합니다. 컨텍스트(오른쪽 클릭) 메뉴에서 새로 생성을 선택한 다음 문자열 값(String Value)을 선택합니다. **DefaultUserName**을 이름으로, **Administrator**를 값 데이터로 입력합니다.
15. HKEY_LOCAL_MACHINE*specified key name*\Microsoft\Windows NT\CurrentVersion\Winlogon 키를 선택합니다. 컨텍스트(오른쪽 클릭) 메뉴에서 새로 생성을 선택한 다음 문자열 값(String Value)을 선택합니다. **DefaultPassword**를 이름으로 입력하고 값 데이터에 암호를 입력합니다.
16. 레지스트리 편집기 탐색 창에서 처음 레지스트리 편집기를 열 때 생성한 임시 키를 선택합니다.
17. 파일(File) 메뉴에서 Hive 언로드(Unload Hive)를 선택합니다.
18. 디스크 관리 유틸리티에서 이전에 연결한 드라이브를 선택하고 마우스 오른쪽 버튼을 클릭하여 컨텍스트 메뉴를 열고 오프라인을 선택합니다.
19. Amazon EC2 콘솔에서 임시 인스턴스로부터 문제가 발생한 볼륨을 분리하고 디바이스 이름 /dev/sda1을 사용하여 인스턴스에 이를 다시 연결합니다. 볼륨을 루트 볼륨으로 지정하려면 이 디바이스 이름을 지정해야 합니다.
20. Amazon EC2 인스턴스 중지 및 시작 인스턴스.
21. 인스턴스 시작 후에 시스템 로그를 검사하여 Windows is ready to use 메시지를 확인합니다.
22. 레지스트리 편집기를 열고 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon을 선택합니다. 앞에서 만든 문자열 값 키 AutoAdminLogon, DefaultUserName 및 DefaultPassword를 삭제합니다.
23. 앞에서 임시로 생성한 인스턴스는 이 절차에서 삭제하거나 중단합니다.

EC2Config 버전 기록

다음 표에서는 EC2Config의 릴리스 버전에 대해 설명합니다. SSM Agent 업데이트에 대한 자세한 내용은 Systems Manager SSM Agent 출시 정보를 참조하세요.

> **Important**
>
> EC2Config가 지원 종료 단계에 도달했습니다. 최신 버전의 EC2Config 에이전트만 다운로드할 수 있습니다. 이전 버전은 비공개로 표시됩니다.

버전	세부 정보	릴리스 날짜
4.9.5777	• 일부 인스턴스 유형에서 RSS 구성이 잘못 설정되는 문제가 해결되었습니다. • 새 버전의 SSM 에이전트 `3.3.484.0` .	2024년 6월 17일
4.9.5554	• 레지스트리 항목을 기준으로 도메인 이름 권한 승계 제한: `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Dnscache\Parameters\DomainNameDevolutionLevel` . • 새 버전의 SSM 에이전트 `3.2.1630.0` .	2023년 10월 4일
4.9.5467	• 콘솔 포트 검색을 위한 재시도 기능을 추가했습니다. • 새 버전의 SSM 에이전트 `3.1.2282.0` .	2023년 8월 1일
4.9.5288	• AWS 코어 SDK를 버전 `3.7.103.23` 으로 업데이트했습니다. • IMDSv2만 사용하도록 설정된 인스턴스에서 `AWS-UpdateEC2Config` SSM 문서가 `EC2Config` 를 업데이트하지 못하는 문제가 해결되었습니다. • 새 버전의 SSM 에이전트 `3.1.2144.0` .	2023년 3월 8일

버전	세부 정보	릴리스 날짜
4.9.5231	• 새로운 SSM Agent 버전 3.1.1927.0.	2023년 2월 14일
4.9.5103	• r5d 및 i4i 인스턴스 패밀리에서 휘발성 볼륨이 잘못 식별되는 문제가 해결되었습니다. • 새로운 SSM Agent 버전 3.1.1856.0	2022년 12월 5일
4.9.5064	• PCI 세그먼트 정보를 사용하여 콘솔 포트를 선택하도록 업데이트되었습니다. • PowerShell 스크립트에 서명하고 저작권 헤더를 추가했습니다. • 기본 네트워크 어댑터 선택 로직을 수정했습니다. • 새로운 SSM Agent 버전 3.1.1732.0	2022년 11월 16일
4.9.4588	• IMDSv2 요청만 수행하도록 IMDS 대기 로직을 업데이트했습니다. • libec2launch.dll 시작 에이전트 공유 라이브러리가 추가되었습니다. • 새로운 SSM Agent 버전 3.1.1188.0	2022년 5월 31일
4.9.4556	• 사용하기 전에 NIC를 완전히 초기화하도록 대기 로직을 추가했습니다. • Log4Net 2.0.14.0의 새 버전이 보안 패치를 선택합니다. • SSM Agent 3.1.1045.0의 새 버전이 보안 패치를 선택합니다.	2022년 3월 1일
4.9.4536	• 임시 폴더가 없을 때 사용자 데이터가 충돌하는 문제가 수정되었습니다. • 새로운 SSM Agent 버전 3.1.804.0	2022년 1월 31일

버전	세부 정보	릴리스 날짜
4.9.4508	• diskpart 스크립트 경로를 올바르게 계산하도록 문제가 해결되었습니다. • 새로운 SSM Agent 버전 3.1.338.0	2021년 10월 6일
4.9.4500	• IMDS v2를 지원하도록 `Install-EgpuManagerConfig` 를 업데이트했습니다. • https를 사용하도록 웹 링크를 업데이트했습니다. • 새로운 SSM Agent 버전 3.1.282.0	2021년 9월 7일
4.9.4419	• IMDS 버전 1 대체 로직을 수정했습니다. • Windows 임시 디렉터리의 모든 사용을 EC2Config 임시 디렉터리로 업데이트 • 새로운 SSM Agent 버전 3.0.1124.0	2021년 6월 2일
4.9.4381	• EC2ConfigUpdater의 SSM 문서 스키마 버전 2.2에 대한 지원 추가 • AWS Nitro Enclaves 패키지 버전이 콘솔 로그에 추가됨 • 새로운 새로운 SSM Agent 3.0.529.0	2021년 5월 4일
4.9.4326	• 설정 UI의 모든 링크를 제거했음 • Windows Server 2008을 지원하는 마지막 EC2Config 버전입니다.	2021년 3월 3일
4.9.4279	• `Ec2ConfigMonitor` 예약 태스크와 관련된 보안 문제가 수정됨 • 드라이브 문자 매핑 문제 및 잘못된 휘발성 디스크 수가 수정됨 • `OsCurrentBuild` 및 `OsReleaseId` 가 콘솔 출력에 추가됨 • 새로운 SSM Agent 버전 2.3.871.0	2020년 12월 11일

버전	세부 정보	릴리스 날짜
4.9.4222	• IMDS 버전 1 대체 로직을 수정했습니다. • 새로운 SSM Agent 버전 2.3.842.0	2020년 4월 7일
4.9.4122	• IMDS v2에 대한 지원을 추가했습니다. • 새로운 SSM Agent 버전 2.3.814.0	2020년 3월 4일
4.9.3865	• 메탈 인스턴스에서 Windows Server 2008 R2에 대한 COM 포트 감지 문제를 해결했습니다. • 새로운 SSM Agent 버전 2.3.722.0	2019년 10월 31일
4.9.3519	• 새로운 SSM Agent 버전 2.3.634.0	2019년 6월 18일
4.9.3429	• 새로운 SSM Agent 버전 2.3.542.0	2019년 4월 25일
4.9.3289	• 새로운 SSM Agent 버전 2.3.444.0	2019년 2월 11일
4.9.3270	• ACPI 문제 해결을 위해 모니터가 절대로 꺼지지 않게 설정하는 플러그인을 추가했습니다 • SQL Server 에디션과 버전을 콘솔에 기록했습니다 • 새로운 SSM Agent 버전 2.3.415.0	2019년 1월 22일
4.9.3230	• 기능에 좀 더 부합하도록 드라이브 문자 매핑 설명을 업데이트했습니다. • 새로운 SSM Agent 버전 2.3.372.0	2019년 1월 10일
4.9.3160	• 기본 NIC의 대기 시간이 증가했습니다. • RSS에 대한 기본 구성과 ENA 디바이스에 대한 수신 대기열 설정을 추가했습니다. • Sysprep 도중 최대 절전 모드를 비활성화했습니다. • 새로운 SSM Agent 버전 2.3.344.0 • AWS SDK가 3.3.29.13으로 업그레이드	2018년 12월 15일

버전	세부 정보	릴리스 날짜
4.9.3067	• 인스턴스 최대 절전 모드를 개선했습니다. • 새로운 SSM Agent 버전 2.3.235.0	2018년 11월 8일
4.9.3034	• DNS 서버에 대한 라우트 169.254.169.253/32를 추가했습니다. • 새로운 SSM Agent 버전 2.3.193.0	2018년 10월 24일
4.9.2986	• 모든 EC2Config 관련에 대한 서명을 추가했습니다. • 새로운 SSM Agent 버전 2.3.136.0	2018년 10월 11일
4.9.2953	새로운 SSM Agent 버전(2.3.117.0)	2018년 10월 2일
4.9.2926	새로운 SSM Agent 버전(2.3.68.0)	2018년 9월 18일
4.9.2905	• 새로운 SSM Agent 버전(2.3.50.0) • AMZN 시간 서비스에 라우트 169.254.169.123/32 추가. • GRID 라이선스 서비스에 라우트 169.254.169.249/32 추가. • EBS NVMe 볼륨이 임시로 표시되는 문제 해결.	2018년 9월 17일
4.9.2854	새로운 SSM Agent 버전(2.3.13.0)	2018년 8월 17일
4.9.2831	새로운 SSM Agent 버전(2.2.916.0)	2018년 8월 7일
4.9.2818	새로운 SSM Agent 버전(2.2.902.0)	2018년 7월 31일
4.9.2756	새로운 SSM Agent 버전(2.2.800.0)	2018년 6월 27일
4.9.2688	새로운 SSM Agent 버전(2.2.607.0)	2018년 5월 25일

버전	세부 정보	릴리스 날짜
4.9.2660	새로운 SSM Agent 버전(2.2.546.0)	2018년 5월 11일
4.9.2644	새로운 SSM Agent 버전(2.2.493.0)	2018년 4월 26일
4.9.2586	새로운 SSM Agent 버전(2.2.392.0)	2018년 3월 28일
4.9.2565	• 새로운 SSM Agent 버전(2.2.355.0) • M5 및 C5 인스턴스의 문제 수정(PV 드라이버 검색 불가) • 인스턴스 유형, 최신 PV 드라이버, NVMe 드라이버에 대한 콘솔 로깅 추가	2018년 3월 13일
4.9.2549	새로운 SSM Agent 버전(2.2.325.0)	2018년 3월 8일
4.9.2461	새로운 SSM Agent 버전(2.2.257.0)	2018년 2월 15일
4.9.2439	새로운 SSM Agent 버전(2.2.191.0)	2018년 2월 6일
4.9.2400	새로운 SSM Agent 버전(2.2.160.0)	2018년 1월 16일
4.9.2327	• 새로운 SSM Agent 버전(2.2.120.0) • Amazon EC2 베어 메탈 인스턴스에 COM 포트 검색을 추가했습니다. • Amazon EC2 베어 메탈 인스턴스에 Hyper-V 상태 로깅을 추가했습니다.	2018년 1월 2일
4.9.2294	새로운 SSM Agent 버전(2.2.103.0)	2017년 12월 4일

버전	세부 정보	릴리스 날짜
4.9.2262	새로운 SSM Agent 버전(2.2.93.0)	2017년 11월 15일
4.9.2246	새로운 SSM Agent 버전(2.2.82.0)	2017년 11월 11일
4.9.2218	새로운 SSM Agent 버전(2.2.64.0)	2017년 10월 29일
4.9.2212	새로운 SSM Agent 버전(2.2.58.0)	2017년 10월 23일
4.9.2203	새로운 SSM Agent 버전(2.2.45.0)	2017년 10월 19일
4.9.2188	새로운 SSM Agent 버전(2.2.30.0)	2017년 10월 10일
4.9.2180	• 새로운 SSM Agent 버전(2.2.24.0) • GPU 인스턴스용 탄력적 GPU 플러그인을 추가했습니다.	2017년 10월 5일
4.9.2143	새로운 SSM Agent 버전(2.2.16.0)	2017년 10월 1일
4.9.2140	새로운 SSM Agent 버전(2.1.10.0)	
4.9.2130	새로운 SSM Agent 버전(2.1.4.0)	
4.9.2106	새로운 SSM Agent 버전(2.0.952.0)	
4.9.2061	새로운 SSM Agent 버전(2.0.922.0)	
4.9.2047	새로운 SSM Agent 버전(2.0.913.0)	
4.9.2031	새로운 SSM Agent 버전(2.0.902.0)	

버전	세부 정보	릴리스 날짜
4.9.2016	• 새로운 SSM Agent 버전(2.0.879.0) • Windows Server 2003용 CloudWatch Logs 디렉터리 경로를 수정했습니다.	
4.9.1981	• 새로운 SSM Agent 버전(2.0.847.0) • EBS 볼륨에서 생성되는 `important.txt` 와 관련된 문제를 해결했습니다.	
4.9.1964	새로운 SSM Agent 버전(2.0.842.0)	
4.9.1951	• 새로운 SSM Agent 버전(2.0.834.0) • 휘발성 드라이브에 대해 Z:에서 매핑되지 않는 드라이브 문자 문제를 해결했습니다.	
4.9.1925	• 새로운 SSM Agent 버전(2.0.822.0) • [버그] 이 버전은 SSM Agent v4.9.1775에서 업데이트할 수 있는 대상 버전이 아닙니다.	
4.9.1900	새로운 SSM Agent 버전(2.0.805.0)	
4.9.1876	• 새로운 SSM Agent 버전(2.0.796.0) • 관리 사용자 데이터 실행에 대한 출력/오류 리디렉션 문제를 해결했습니다.	

버전	세부 정보	릴리스 날짜
4.9.1863	• 새로운 SSM Agent 버전(2.0.790.0) • 여러 EBS 볼륨을 Amazon EC2 인스턴스에 연결할 때 발생하는 문제를 해결했습니다. • 이전 버전과의 호환성을 위해 구성 경로를 가져오도록 CloudWatch를 개선했습니다.	
4.9.1791	새로운 SSM Agent 버전(2.0.767.0)	
4.9.1775	새로운 SSM Agent 버전(2.0.761.0)	
4.9.1752	새로운 SSM Agent 버전(2.0.755.0)	
4.9.1711	새로운 SSM Agent 버전(2.0.730.0)	
4.8.1676	새로운 SSM Agent 버전(2.0.716.0)	
4.7.1631	새로운 SSM Agent 버전(2.0.682.0)	
4.6.1579	• 새로운 SSM Agent 버전(2.0.672.0) • v4.3, v4.4, v4.5의 에이전트 업데이트 문제를 수정했습니다.	
4.5.1534	새로운 SSM Agent 버전(2.0.645.1)	
4.4.1503	새로운 SSM Agent 버전(2.0.633.0)	
4.3.1472	새로운 SSM Agent 버전(2.0.617.1)	
4.2.1442	새로운 SSM Agent 버전(2.0.599.0)	
4.1.1378	새로운 SSM Agent 버전(2.0.558.0)	

버전	세부 정보	릴리스 날짜
4.0.1343	• Run Command, State Manager, CloudWatch 에이전트, 도메인 조인 지원이 SSM Agent라고 하는 다른 에이전트로 이전되었습니다. SSM Agent는 EC2Config 업그레이드의 일부로 설치됩니다. 자세한 내용은 EC2Config 및 AWS Systems Manager 섹션을 참조하세요. • EC2Config에 프록시를 설정해 놓은 경우 업그레이드하기 전에 SSM Agent의 프록시 설정을 업데이트해야 합니다. 프록시 설정을 업데이트하지 않으면 Run Command를 사용하여 인스턴스를 관리할 수 없습니다. 이러한 문제를 방지하려면 최신 버전으로 업데이트하기 전에 AWS Systems Manager 사용 설명서에서 Windows 인스턴스에서 SSM Agent 설치 및 구성을 참조하세요. • 이전에 로컬 구성 파일(`AWS.EC2.Windows.CloudWatch.json`)을 사용하여 인스턴스에서 CloudWatch 통합을 사용하도록 설정한 경우 SSM Agent로 작업하도록 이 파일을 구성해야 합니다.	
3.19.1153	• 이전 AWS KMS 구성을 사용하여 인스턴스에 대한 활성화 플러그인을 다시 활성화했습니다. BYOL 사용자의 경우 활성화를 건너뛰세요. • 디스크 포맷 작업 중에 비활성화할 기본 TRIM 행동을 변경하고, 사용자 데이터로 InitializeDisks 플러그인을 재정의하기 위해 FormatWithTRIM을 추가했습니다.	
3.18.1118	• 기본 네트워크 어댑터에 대한 경로를 안정적으로 추가하도록 수정했습니다. • AWS 서비스 지원 향상을 위한 업데이트.	

버전	세부 정보	릴리스 날짜
3.17.1032	• 필터를 동일한 범주로 설정한 경우에 표시되는 중복 시스템 로그를 수정했습니다. • 디스크 초기화 중에 중지되는 문제를 수정했습니다.	
3.16.930	시작할 때 Windows 이벤트 로그에 "Window is Ready to use" 이벤트를 기록하도록 지원을 추가했습니다.	
3.15.880	'.' 문자를 포함하는 S3 버킷 이름에 대한 Systems Manager Run Command 출력을 업로드할 수 있도록 수정했습니다.	
3.14.786	InitializeDisks 플러그인 설정을 재정의하도록 지원을 추가했습니다. 예: SSD 디스크 초기화 속도를 높이려면 사용자 데이터에서 다음을 지정하여 TRIM을 일시적으로 비활성화할 수 있습니다. <InitializeDrivesSettings><SettingsGroup>FormatWithoutTRIM</SettingsGroup></InitializeDrivesSettings	
3.13.727	Systems Manager Run Command - Windows가 재부팅된 후 명령을 안정적으로 처리하도록 수정했습니다.	
3.12.649	• 명령/스크립트를 실행할 때 재부팅을 정상적으로 처리하도록 수정했습니다. • 실행 중인 명령을 안정적으로 취소할 수 있도록 수정했습니다. • Systems Manager Run Command를 통해 애플리케이션을 설치할 때 MSI 로그를 S3에 업로드(선택 사항)하기 위한 지원을 추가했습니다.	

버전	세부 정보	릴리스 날짜
3.11.521	• Windows Server 2003에 대한 RDP 지문 생성을 활성화하도록 수정했습니다. • EC2Config 로그 줄에 시간대와 UTC 오프셋을 포함하도록 수정했습니다. • Systems Manager는 병렬 Run Command 실행을 지원합니다. • 분할된 디스크를 온라인으로 전환하기 위해 이전 변경 사항을 롤백합니다.	
3.10.442	• MSI 애플리케이션을 설치할 때 발생하는 Systems Manager 구성 장애를 해결했습니다. • 스토리지 디스크를 온라인으로 안정적으로 전환하도록 수정했습니다. • AWS 서비스 지원 향상을 위한 업데이트.	
3.9.359	• Sysprep 이후 스크립트에서 Windows 업데이트 구성을 기본 상태로 유지하도록 수정했습니다. • GPO 암호 정책 설정을 가져올 때 안정성을 향상하기 위해 암호 생성 플러그인을 수정했습니다. • EC2Config/SSM 로그 폴더 권한을 로컬 Administrators 그룹으로 제한합니다. • AWS 서비스 지원 향상을 위한 업데이트.	

버전	세부 정보	릴리스 날짜
3.8.294	• 기본 드라이브에 있지 않은 경우 로그가 업로드되지 않는 CloudWatch 문제를 해결했습니다. • 재시도 로직을 추가하여 디스크 초기화 프로세스를 개선했습니다. • AMI 생성 중에 SetPassword 플러그인이 실패할 경우에 발생하는 오류 처리 성능을 개선했습니다. • AWS 서비스 지원 향상을 위한 업데이트.	
3.7.308	• 인스턴스 내에서 구성을 테스트하고 문제를 해결하기 위해 ec2config-cli 유틸리티를 개선했습니다. • OpenVPN 어댑터에서 AWS KMS 및 메타데이터 서비스에 대한 정적 경로 추가를 방지합니다. • 사용자 데이터 실행 중에 "persist" 태그를 인식하지 못하는 문제를 해결했습니다. • EC2 콘솔 로깅을 사용할 수 없을 때의 오류 처리 성능을 개선했습니다. • AWS 서비스 지원 향상을 위한 업데이트.	
3.6.269	• AWS KMS을(를) 통해 Windows를 정품 인증할 때 링크 로컬 주소 169.254.0.250/251을 먼저 사용하도록 Windows 정품 인증 안정성을 수정했습니다. • Systems Manager, Windows 정품 인증 및 도메인 조인 시나리오에 대한 프록시 처리 성능을 개선했습니다. • Sysprep 응답 파일에 사용자 계정 줄이 중복으로 추가되는 문제를 해결했습니다.	

버전	세부 정보	릴리스 날짜
3.5.228	• CloudWatch 플러그인이 Windows 이벤트 로그를 읽을 때 CPU와 메모리를 과도하게 사용하는 문제를 해결했습니다. • EC2Config 설정 UI에 CloudWatch 구성 문서에 대한 링크를 추가했습니다.	
3.4.212	• VM Import와 함께 사용될 때 발생하는 EC2Config 문제를 해결했습니다. • WiX 설치 관리자의 서비스 이름 지정 문제를 해결했습니다.	
3.3.174	• Systems Manager 및 도메인 조인 장애에 대한 예외 처리 성능을 개선했습니다. • Systems Manager SSM 스키마 버전 관리를 지원하도록 변경했습니다. • Win2K3의 휘발성 디스크 포맷을 수정했습니다. • 2TB보다 큰 디스크 크기 구성을 지원하도록 변경했습니다. • GC 모드를 기본값으로 설정하여 가상 메모리 사용을 축소했습니다. • `aws:psModule` 및 `aws:application` 플러그 인의 UNC 경로에서 아티팩트 다운로드를 지원합니다. • Windows 정품 인증 플러그인에 대한 로깅을 개선했습니다.	

버전	세부 정보	릴리스 날짜
3.2.97	• Systems Manager SSM 어셈블리를 지연 로드하여 성능을 개선했습니다. • 형식이 잘못된 sysprep2008.xml에 대한 예외 처리를 개선했습니다. • Systems Manager 'Apply' 구성에 대한 명령줄 지원을 추가했습니다. • 컴퓨터 이름 바꾸기가 보류 중일 때 도메인 조인을 지원하도록 변경했습니다. • `aws:applications` 플러그인의 선택적 파라미터를 지원합니다. • `aws:psModule` 플러그인의 명령 어레이를 지원합니다.	
3.0.54	• Systems Manager 지원 활성화 • Systems Manager를 통해 EC2 Windows 인스턴스를 AWS 디렉터리에 자동으로 도메인 조인합니다. • Systems Manager를 통해 CloudWatch Logs/지표를 구성하고 업로드합니다. • Systems Manager를 통해 PowerShell 모듈을 설치합니다. • Systems Manager를 통해 MSI 애플리케이션을 설치합니다.	

버전	세부 정보	릴리스 날짜
2.4.233	• 서비스 시작 실패로부터 EC2Config를 복구하도록 예약된 작업을 추가했습니다. • 콘솔 로그 오류 메시지를 개선했습니다. • AWS 서비스 지원 향상을 위한 업데이트.	
2.3.313	• CloudWatch Logs 기능을 활성화할 때 많은 메모리가 사용되는 문제를 해결했습니다. • 업그레이드 버그를 수정하여 이제 ec2config 2.1.19 이하 버전에서 최신 버전으로 업그레이드할 수 있습니다. • 로그에서 COM 포트 열기 예외를 보다 친숙하고 유용하게 업데이트했습니다. • Ec2configServiceSettings UI 크기 조정을 비활성화하고 UI의 특성 및 버전 표시 위치를 수정했습니다.	
2.2.12	• 종종 null을 반환하는 Windows Sysprep 상태를 결정하기 위해 레지스트리 키를 쿼리할 때 발생하는 NullPointerException 예외를 해결했습니다. • 마지막 블록에서 관리되지 않는 리소스를 확보했습니다.	
2.2.11	CloudWatch 플러그인에서 빈 로그 줄 처리 문제를 해결했습니다.	
2.2.10	• UI를 통한 CloudWatch Logs 설정 구성을 제거했습니다. • 사용자는 `%ProgramFiles%\Amazon\Ec2ConfigService\Settings\AWS.EC2.Windows.CloudWatch.json` 파일에서 CloudWatch Logs 설정을 정의하여 추가 개선을 허용할 수 있습니다.	

버전	세부 정보	릴리스 날짜
2.2.9	처리되지 않는 예외를 해결하고 로깅을 추가했습니다.	
2.2.8	• Windows Server 2003 SP1 이상을 지원하도록 EC2Config 설치 관리자에서 Windows OS 버전 검사를 수정했습니다. • Sysprep 구성 파일과 관련된 레지스트리 키를 읽을 때 null 값 처리를 수정했습니다.	
2.2.7	• Windows 2008 이상에 대해 Sysprep을 실행하는 중에 EC2Config를 실행하도록 지원을 추가했습니다. • 향상된 진단을 위해 예외 처리 및 로깅을 개선했습니다.	
2.2.6	• 로그 이벤트를 업로드할 때 인스턴스 및 CloudWatch Logs에 대한 부하를 축소했습니다. • CloudWatch Logs 플러그인이 항상 활성화된 상태로 유지되지 않는 업그레이드 문제를 해결했습니다.	
2.2.5	• CloudWatch Logs 서비스에 대한 로그 업로드 지원을 추가했습니다. • Ec2OutputRDPCert 플러그인에서 경합 문제를 해결했습니다. • TakeNoAction에서 재시작하도록 EC2Config 서비스 복구 옵션을 변경했습니다. • EC2Config가 충돌하는 경우에 더 많은 예외 정보를 추가했습니다.	

버전	세부 정보	릴리스 날짜
2.2.4	• PostSysprep.cmd에서 오타를 수정했습니다. • EC2Config가 OS2012+의 시작 메뉴에 고정되지 않는 버그를 수정했습니다.	
2.2.3	• 설치 시 서비스를 즉시 시작하지 않고 EC2Config를 설치하기 위한 옵션을 추가했습니다. 사용하려면 명령 프롬프트에서 'Ec2Install.exe start=false'를 실행합니다. • 월페이퍼(wallpaper) 플러그인에 월페이퍼 추가/제거를 제어하기 위한 파라미터를 추가했습니다. 사용하려면 명령 프롬프트에서 'Ec2WallpaperInfo.exe set' 또는 'Ec2WallpaperInfo.exe revert'를 실행합니다. • RealTimeIsUniveral 레지스트리 키의 잘못된 설정을 콘솔에 출력하도록 RealTimeIsUniversal 키 검사를 추가했습니다. • Windows temp 폴더에 대한 EC2Config 종속성을 제거했습니다. • .Net 3.5에 대한 사용자 데이터 실행 종속성을 제거했습니다.	
2.2.2	• 리소스가 릴리스 중인지 확인하도록 서비스 중지 동작에 대한 검사 기능을 추가했습니다. • 도메인에 조인될 때 실행 시간이 오래 걸리는 문제를 해결했습니다.	

버전	세부 정보	릴리스 날짜
2.2.1	• 이전 버전에서의 업그레이드를 허용하도록 설치 관리자를 업데이트했습니다. • .Net4.5 전용 환경에서 Ec2WallpaperInfo 버그를 해결했습니다. • 간헐적인 드라이버 감지 버그를 해결했습니다. • 자동 설치 옵션을 추가했습니다. '-q' 옵션을 사용하여 Ec2Install.exe 실행(예: 'Ec2Install.exe -q')	
2.2.0	• .Net4 및 .Net4.5 전용 환경에 대한 지원을 추가했습니다. • 설치 관리자를 업데이트했습니다.	
2.1.19	• intel 네트워크 드라이버(예: C3 인스턴스 유형)를 사용하는 경우의 휘발성 디스크 레이블 지정 자세한 내용은 Amazon EC2 인스턴스의 향상된 네트워킹 섹션을 참조하세요. • 콘솔 출력에 AMI 원본 버전 및 AMI 원본 이름 지원을 추가했습니다. • 일관된 서식/구문 분석을 위해 콘솔 출력을 변경했습니다. • 도움말 파일을 업데이트했습니다.	
2.1.18	• 완료 알림에 대한 EC2Config WMI 객체(-Namespace root\Amazon -Class EC2_ConfigService)를 추가했습니다. • 대용량 이벤트 로그를 사용하여 초기 실행 중에 오래 동안 높은 CPU를 사용하는 시작 WMI 쿼리 성능을 개선했습니다.	

버전	세부 정보	릴리스 날짜
2.1.17	• 표준 출력 및 표준 오류 버퍼 채우기를 통해 사용자 데이터 실행 문제를 해결했습니다. • w2k8 이상 OS에 대한 콘솔 출력에 나타나는 잘못된 RDP 지문을 수정했습니다. • 이제 Windows 2008 이상 버전의 콘솔 출력에는 머신 이름 값을 포함하는 'RDPCERTIFICATE-SubjectName:'이 포함되어 있습니다. • [D:\ to Drive Letter Mapping] 드롭다운을 추가했습니다. • 도움말 버튼을 오른쪽 위로 이동하고 모양/느낌을 변경했습니다. • Feedback Survey 링크를 오른쪽 위에 추가했습니다.	

버전	세부 정보	릴리스 날짜
2.1.16	• [General] 탭에 새 버전의 EC2Config 다운로드 페이지에 대한 링크가 포함되어 있습니다. • MyDoc 리디렉션을 지원하기 위해 이제 바탕 화면 월페이퍼(Wallpaper) 오버레이가 [내 문서] 대신 Users Local Appdata 폴더에 저장됩니다. • Sysprep 이후 스크립트에서 MSSQLServer 이름이 시스템과 동기화되었습니다(2008 이상). • 애플리케이션 폴더 순서를 변경했습니다(파일을 Plugin 디렉터리로 이동하고 중복 파일 제거). • 시스템 로그 출력(콘솔)을 변경했습니다. • *쉬운 구문 분석을 위해 날짜, 이름, 값 형식으로 전환했습니다. 새 형식에 대한 종속성 마이그레이션을 시작하세요. • *'Ec2SetPassword' 플러그인 상태를 추가했습니다. • *Sysprep 시작 및 종료 시간을 추가했습니다. • 영어 이외 언어로 된 운영 체제에서 휘발성 디스크가 'Temporary Storage'로 레이블링되지 않는 문제를 해결했습니다. • Sysprep을 실행한 이후의 EC2Config 제거 오류를 해결했습니다.	

버전	세부 정보	릴리스 날짜
2.1.15	• 메타데이터 서비스에 대한 요청을 최적화했습니다. • 메타데이터는 이제 프록시 설정을 우회합니다. • 휘발성 디스크가 'Temporary Storage'로 레이블 지정되고 Important.txt가 볼륨에 배치됩니다(있는 경우)(Citrix PV 드라이버에만 해당). 자세한 내용은 EC2 Windows 인스턴스의 PV 드라이버 업그레이드 섹션을 참조하세요. • 휘발성 디스크에는 A부터 Z 사이의 드라이브 문자가 할당됩니다(Citrix PV 드라이버에만 해당). 드라이브 문자 매핑 플러그인을 사용하여 볼륨 레이블 'Temporary Storage X'로 할당을 덮어쓸 수 있습니다. 여기서 x는 0-25 사이의 숫자입니다. • 이제 [Windows가 준비됨(Windows is Ready)] 상태 전환 후 즉시 UserData가 실행됩니다.	
2.1.14	바탕 화면 월페이퍼(wallpaper)를 수정했습니다.	
2.1.13	• 바탕 화면 월페이퍼(wallpaper)에 기본적으로 호스트 이름이 표시됩니다. • Windows 시간 서비스에 대한 종속성을 제거했습니다. • 단일 인터페이스에 여러 IP가 할당되는 경우의 경로를 추가했습니다.	
2.1.11	• Ec2Activation 플러그인을 변경했습니다. • -정품 인증 상태를 30일마다 확인합니다. • -유예 기간이 180일 중에서 90일이 남으면 정품 인증을 다시 시도합니다.	

버전	세부 정보	릴리스 날짜
2.1.10	• Sysprep 또는 [Shutdown without Sysprep] 시 바탕 화면 월페이퍼(wallpaper) 오버레이가 더 이상 지속되지 않습니다. • UserData 옵션을 사용하여 <persist>true</persist>로 시작되는 모든 서비스를 실행 • /DisableWinUpdate.cmd의 위치와 이름을 /Scripts/PostSysprep.cmd로 변경했습니다. • /Scripts/PostSysprep.cmd에서 관리자 암호를 기본적으로 만료되지 않도록 설정했습니다. • 설치 제거하면 EC2Config PostSysprep 스크립트가 c:\windows\setup\script\CommandComplete.cmd에서 제거됩니다. • [Add Route]에서 사용자 지정 인터페이스 지표를 지원합니다.	
2.1.9	사용자 데이터 실행이 더 이상 3851자로 제한되지 않습니다.	

버전	세부 정보	릴리스 날짜
2.1.7	• OS 버전과 언어 식별자가 콘솔에 기록됩니다. • EC2Config 버전이 콘솔에 기록됩니다. • PV 드라이버 버전이 콘솔에 기록됩니다. • 버그를 검사한 후 버그가 있을 경우 다음에 부팅할 때 콘솔에 출력합니다. • Sysprep 자격 증명을 유지하도록 config.xml에 옵션을 추가했습니다. • 시작 시 ENI를 사용할 수 없을 경우 경로 재시도 로직을 추가했습니다. • 사용자 데이터 실행 PID를 콘솔에 기록했습니다. • 최소 생성 암호 길이를 GPO에서 검색합니다. • 3회 재시도하도록 서비스 시작을 설정했습니다. • S3_DownloadFile.ps1 및 S3_Upload file.ps1 예를 /Scripts 폴더에 추가했습니다.	

버전	세부 정보	릴리스 날짜
2.1.6	• 버전 정보를 [General] 탭에 추가했습니다. • [Bundle] 탭의 이름을 [Image]로 변경했습니다. • 암호 지정 프로세스를 간소화하고 암호 관련 UI를 [General] 탭에서 [Image] 탭으로 이동했습니다. • [Disk Settings] 탭의 이름을 [Storage]로 변경했습니다. • 문제 해결을 위한 일반 도구가 들어 있는 [Support] 탭을 추가했습니다. • OS 파티션을 기본적으로 확장하도록 Windows Server 2003 `sysprep.ini` 를 설정했습니다. • 월페이퍼(wallpaper)에 대한 프라이빗 IP 주소를 추가했습니다. • 프라이빗 IP 주소가 월페이퍼(wallpaper)에 표시됩니다. • 콘솔 출력에 대한 재시도 로직을 추가했습니다. • 메타데이터 액세스 가능성에 대한 COM 포트 예외(콘솔 출력이 표시되기 이전에 EC2Config가 종료되는 문제)를 수정했습니다. • 부팅할 때마다 정품 인증 상태를 확인하여 필요 시 정품 인증을 수행합니다. • 시작 폴더에서 월페이퍼(wallpaper) 바로 가기를 수동으로 실행할 때 Administrator/logs를 가리키는 상대 경로 문제를 해결했습니다. • Windows Server 2003 사용자(관리자 이외)에 대한 기본 배경색을 수정했습니다.	

버전	세부 정보	릴리스 날짜
2.1.2	• UTC(Zulu) 기반 콘솔 타임스탬프 • [Sysprep] 탭에서 하이퍼링크 모양을 제거했습니다. • Windows 2008 이상을 처음으로 부팅할 때 루트 볼륨을 동적으로 확장하는 기능을 추가했습니다. • [Set-Password]를 활성화하면 이제 암호 설정을 위해 EC2Config가 자동으로 활성화됩니다. • EC2Config는 Sysprep을 실행하기 이전에 정품 인증 상태를 확인합니다(정품 인증되지 않은 경우 경고 표시). • 이제 Windows Server 2003 `Sysprep.xml` 은 기본적으로 태평양 시간대 대신 UTC 시간대로 설정됩니다. • 임의 정품 인증 서버 • [Drive Mapping] 탭의 이름을 [Disk Settings]로 변경했습니다. • [Initialize Drives] UI 항목을 [General] 탭에서 [Disk Settings] 탭으로 이동했습니다. • 도움말 버튼이 이제 HTML 도움말 파일을 가리킵니다. • 변경 사항으로 HTML 도움말 파일을 업데이트했습니다. • 드라이브 문자 매핑에 대한 'Note' 텍스트를 업데이트했습니다. • Sysprep 이전에 패치 및 정리를 자동화하기 위해 InstallUpdates.ps1을 /Scripts 폴더에 추가했습니다.	

버전	세부 정보	릴리스 날짜
2.1.0	• 처음으로 로그온할 때(연결을 끊었다가 다시 연결할 때 아님) 바탕 화면 월페이퍼(wallpaper)에 기본적으로 인스턴스 정보가 표시됩니다. • 코드를 <powershell></powershell>로 묶어서 사용자 데이터에서 PowerShell을 실행할 수 있음	

Windows 인스턴스에 EC2 Fast Launch 사용

EC2 Fast Launch을 위해 Windows Server AMI를 구성하면 Amazon EC2는 다음과 같이 더 빠른 시작을 위해 사용할 사전 프로비저닝된 스냅샷 세트를 생성합니다.

1. Amazon EC2는 설정에 따라 임시 t3 인스턴스 세트를 시작합니다.
2. 각 임시 인스턴스가 표준 시작 단계를 완료하면 Amazon EC2는 사전 프로비저닝된 인스턴스 스냅샷을 생성합니다. 이 스냅샷을 Amazon S3 버킷에 저장합니다.
3. 스냅샷이 준비되면 Amazon EC2는 리소스 비용을 최대한 낮게 유지하기 위해 관련 t3 인스턴스를 종료합니다.
4. 다음 번에 Amazon EC2가 EC2 Fast Launch가 활성화된 AMI에서 인스턴스를 시작할 때 이러한 스냅샷 중 하나를 사용하여 시작하는 데 걸리는 시간을 상당히 줄입니다.

Amazon EC2는 준비된 스냅샷을 사용하여 EC2 Fast Launch가 활성화된 AMI의 인스턴스를 시작할 때 자동으로 이러한 스냅샷을 보충합니다.

EC2 Fast Launch가 활성화된 AMI에 액세스할 수 있는 모든 계정은 시작 시간 단축의 이점을 누릴 수 있습니다. AMI 소유자가 인스턴스를 시작할 수 있는 액세스 권한을 부여하면 사전 프로비저닝된 스냅샷이 AMI 소유자 계정에서 제공됩니다.

EC2 Fast Launch를 지원하는 AMI를 공유하는 경우 공유 AMI에서 직접 빠른 시작을 활성화하거나 비활성화할 수 있습니다. EC2 Fast Launch를 위해 공유 AMI를 활성화하면 Amazon EC2는 계정에 직접 사전 프로비저닝된 스냅샷을 생성합니다. 본인 계정의 스냅샷이 고갈되어도 AMI 소유자 계정의 스냅샷을 계속 사용할 수 있습니다.

> **Note**
>
> EC2 Fast Launch는 시작과 함께 사용되는 사전 프로비저닝된 스냅샷을 삭제하여 스토리지 비용을 최소화하고 재사용을 방지합니다. 하지만 삭제된 스냅샷이 보관 규칙에 해당하는 경우 휴지통에 자동으로 보관됩니다. 이런 일이 발생하지 않도록 휴지통 보관 규칙의 범위를 검토하는 것이 좋습니다. 자세한 내용은 Amazon EBS 사용 설명서의 휴지통을 참조하세요.
> 이 기능은 EBS 빠른 스냅샷 복원과 동일하지 않습니다. EBS 빠른 스냅샷 복원은 스냅샷별로 명시적으로 활성화되어야 하며 자체 관련 비용이 있습니다.

AWS에서 EC2 Windows 인스턴스를 최대 65% 더 빠르게 시작 비디오에서는 Windows AMI를 더 빠르게 시작할 수 있도록 구성하는 방법과 관련 주요 용어 및 정의에 대해 간략하게 소개합니다.

리소스 비용

EC2 Fast Launch을 위해 Windows AMI를 구성하는 데 드는 서비스 요금은 없습니다. 그러나 Amazon EC2가 사용하는 모든 기본 AWS 리소스에는 표준 요금이 적용됩니다. 관련 리소스 비용과 이를 관리하는 방법에 대해 자세히 알아보려면 EC2 Fast Launch의 기본 리소스 비용 관리 섹션을 참조하세요.

내용

- 주요 용어
- Windows용 EC2 Fast Launch 사전 조건
- Amazon EC2 Windows Server AMI를 위한 EC2 Fast Launch를 구성합니다.
- EC2 Fast Launch가 활성화된 Windows AMI 보기
- EC2 Fast Launch의 기본 리소스 비용 관리
- EC2 Fast Launch 모니터링
- EC2 Fast Launch를 위한 서비스 연결 역할
- Windows EC2 Fast Launch 문제 해결

주요 용어

EC2 Fast Launch 기능에서는 다음과 같은 주요 용어를 사용합니다.

사전 프로비저닝된 스냅샷

EC2 Fast Launch가 활성화된 Windows AMI에서 시작된 인스턴스의 스냅샷으로, 다음의 Windows 시작 단계를 완료하여 필요에 따라 재부팅합니다.

- Sysprep 특수화
- Windows Out of Box Experience(OOBE)

이러한 단계가 완료되면 EC2 Fast Launch은 인스턴스를 중지하고 사용자의 구성에 따라 나중에 AMI에서 더 빠르게 시작하는 데 사용되는 스냅샷을 생성합니다.

실행 빈도

Amazon EC2가 지정된 기간 내에 시작할 수 있는 사전 프로비저닝된 스냅샷 수를 제어합니다. AMI에 대해 EC2 Fast Launch를 활성화하면 Amazon EC2가 백그라운드에서 사전 프로비저닝된 스냅샷의 초기 세트를 생성합니다. 예를 들어, 시작 빈도가 시간당 5개의 기본값으로 설정된 경우 EC2 Fast Launch은 사전 프로비저닝된 스냅샷 5개로 구성된 초기 집합을 생성합니다.

Amazon EC2가 EC2 Fast Launch가 활성화된 AMI에서 인스턴스를 시작할 때 사전 프로비저닝된 스냅샷 중 하나를 사용하여 시작 시간을 줄입니다. 스냅샷이 사용되면 시작 빈도에 지정된 숫자까지 자동으로 보충됩니다.

AMI에서 시작되는 인스턴스 수가 급증할 것으로 예상되는 경우(예: 특별 이벤트 중) 필요한 추가 인스턴스를 처리하기 위해 미리 시작 빈도를 늘릴 수 있습니다. 시작 비율이 정상으로 돌아오면 빈도를 다시 낮출 수 있습니다.

예상보다 더 많은 수의 시작이 발생하면 사용 가능한 사전 프로비저닝된 스냅샷을 모두 사용할 수 있습니다. 이로 인해 실행이 실패하지는 않습니다. 그러나 스냅샷을 보충할 수 있을 때까지 일부 인스턴스가 표준 시작 프로세스를 거치게 될 수 있습니다.

대상 리소스 수

EC2 Fast Launch가 활성화된 Amazon EC2 Windows Server AMI를 위해 보유하는 사전 프로비저닝된 스냅샷 수입니다.

최대 병렬 시작

Amazon EC2가 EC2 Fast Launch에 사용할 사전 프로비저닝된 스냅샷을 생성하기 위해 동시에 시작할 수 있는 인스턴스의 수를 제어합니다. 대상 리소스 수가 구성한 최대 병렬 시작 수보다 많은 경우 Amazon EC2는 최대 병렬 시작에서 지정한 수의 인스턴스를 시작하여 스냅샷 생성을 시작합니다. 이러한 인스턴스가 프로세스를 완료하면 Amazon EC2가 스냅샷을 가져오고 인스턴스를 중지합니다. 그런 다음 사용 가능한 총 스냅샷 수가 대상 리소스 수에 도달할 때까지 계속해서 더 많은 인스턴스를 시작합니다. 최대 병렬 시작 값은 6 이상이어야 합니다.

Windows용 EC2 Fast Launch 사전 조건

EC2 Fast Launch를 설정하기 전에 AWS 계정에서 AMI에 대한 스냅샷을 생성하는 데 필요한 다음과 같은 필수 조건이 충족되었는지 확인하세요.

- EC2 Fast Launch를 구성할 때 사용자 지정 시작 템플릿을 제공하는 경우 서비스는 시작 템플릿에서 정의한 VPC와 기타 구성 설정을 사용합니다. 자세한 내용은 EC2 Fast Launch을 설정할 때 시작 템플릿 사용 섹션을 참조하세요.
- 사용자 지정 시작 템플릿을 사용하여 설정을 구성하지 않는 경우 EC2 Fast Launch를 활성화하기 전에 EC2FastLaunchFullAccess 정책을 현재 IAM 역할에 연결해야 합니다. 그런 다음 서비스는 AWS 계정에 다음 리소스가 포함된 AWS CloudFormation 스택을 자동으로 생성합니다.
  - Virtual Private Cloud(VPC)
  - 여러 가용 영역의 프라이빗 서브넷
  - 인스턴스 메타데이터 서비스 버전 2(IMDSv2)로 구성된 시작 템플릿
  - 인바운드나 아웃바운드 규칙이 없는 보안 그룹
- 프라이빗 EC2 Fast Launch AMI는 사용자 데이터 스크립트 실행을 지원해야 합니다.
- AMI에 EC2 Fast Launch를 구성하려면 종료 옵션과 함께 Sysprep을 사용하여 생성해야 합니다. EC2 Fast Launch 기능은 현재 실행 중인 인스턴스에서 생성된 AMI를 지원하지 않습니다.

  Sysprep를 사용하여 AMI를 생성하려면 Windows Sysprep을 사용하여 Amazon EC2 AMI 생성 섹션을 참조하세요.
- 암호화를 위해 고객 관리형 키를 사용하는 암호화된 AMI에 대해 EC2 Fast Launch를 활성화하려면 EC2 Fast Launch에 대한 서비스 연결 역할에 CMK를 사용할 수 있는 권한을 부여해야 합니다. 자세한 내용은 the section called “고객 관리형 키에 대한 액세스” 섹션을 참조하세요.
- AWS 계정의 모든 AMI에 대한 최대 병렬 시작의 기본 할당량은 리전당 40개입니다. 다음과 같이 계정에 대한 Service Quotas 증가를 요청할 수 있습니다.

  1. https://console.aws.amazon.com/servicequotas/에서 Service Quotas 콘솔을 엽니다.
  2. 탐색 창에서 AWS 서비스를 선택합니다.
  3. 검색 창에 **`EC2 Fast Launch`**를 입력하고 결과를 선택합니다.
  4. 병렬 인스턴스 시작 링크를 선택하여 서비스 할당량 세부 정보 페이지를 엽니다.
  5. 계정 수준에서 증가 요청을 선택합니다.

  자세한 내용은 Service Quotas 사용 설명서의 할당량 증가 요청을 참조하십시오.

Amazon EC2 Windows Server AMI를 위한 EC2 Fast Launch를 구성합니다.

AWS Management Console, API, SDK, CloudFormation 또는 AWS Command Line Interface(AWS CLI)에서 사용자가 소유한 Windows AMI 또는 사용자와 공유하는 AMI에 대해 EC2 Fast Launch를 구성할 수 있습니다. EC2 Fast Launch를 구성하기 전에 AMI가 사전 프로비저닝된 스냅샷을 생성하는 데 필요한 모든 사전 요구 사항을 충족하는지 확인하세요. 자세한 내용은 Windows용 EC2 Fast Launch 사전 조건 섹션을 참조하세요.

Windows 인스턴스에서 빠른 시작을 활성화하면 Amazon EC2는 암호화된 AMI에 대한 권한을 포함하여 지정된 AMI 및 시작 템플릿(제공된 경우)에서 인스턴스를 시작하는 데 필요한 권한이 있는지 확인합니다. 인스턴스 시작 프로세스 중 오류를 방지하기 위해 서비스는 EC2 Fast Launch를 활성화하기 전에 권한을 검증합니다. 필요한 권한이 없는 경우 서비스는 오류를 반환하고 EC2 Fast Launch를 활성화하지 않습니다.

EC2 Fast Launch은 EC2 Image Builder와 통합되어 EC2 Fast Launch가 활성화된 사용자 지정 이미지를 생성할 수 있도록 도와줍니다. 자세한 내용은 EC2 Image Builder 사용 설명서에서 EC2 빠른 시작이 활성화된 Windows AMI에 대한 배포 설정 생성(AWS CLI)을 참조하세요.

EC2 Fast Launch 활성화

이러한 설정을 변경하기 전에 AMI와 실행하는 리전이 Windows용 EC2 Fast Launch 사전 조건을 모두 충족하는지 확인하세요.

Console

EC2 Fast Launch를 활성화하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창의 이미지(Images) 아래에서 AMI를 선택합니다.
3. 이름(Name) 옆의 확인란을 선택하여 업데이트할 AMI를 선택합니다.
4. AMI 목록 위에 있는 작업 메뉴에서 빠른 시작 구성을 선택합니다. 그러면 EC2 Fast Launch의 설정을 구성할 수 있는 빠른 시작 구성 페이지가 열립니다.
5. 사전 프로비저닝된 스냅샷을 사용하여 Windows AMI에서 인스턴스를 더 빠르게 시작하려면 Windows용 빠른 시작 활성화 확인란을 선택합니다.
6. 예상 시작 빈도 설정(Set anticipated launch frequency) 드롭다운 목록에서 값을 선택하여 예상 인스턴스 시작 볼륨을 처리하기 위해 생성 및 유지 관리되는 스냅샷 수를 지정합니다.
7. 변경 작업을 마치면 변경 사항 저장(Save changes)을 선택합니다.

> **Note**
>
> VPC를 지정하거나 IMDSv2에 대한 메타데이터 설정을 구성하기 위해 시작 템플릿을 사용해야 하는 경우 EC2 Fast Launch을 설정할 때 시작 템플릿 사용를 참조하세요.

AWS CLI

EC2 Fast Launch를 활성화하는 방법

다음 enable-fast-launch 명령을 사용하여 지정된 AMI에 대한 EC2 Fast Launch를 활성화하며, 사전 프로비저닝을 위한 6개의 병렬 인스턴스를 시작합니다.

```
aws ec2 enable-fast-launch \
    --image-id ami-0abcdef1234567890  \
    --max-parallel-launches 6 \
    --resource-type snapshot
```

출력의 예시는 다음과 같습니다.

```
{
 "ImageId": "ami-0abcdef1234567890",
 "ResourceType": "snapshot",
 "SnapshotConfiguration": {
     "TargetResourceCount": 10
 },
 "LaunchTemplate": {},
 "MaxParallelLaunches": 6,
 "OwnerId": "0123456789123",
 "State": "enabling",
 "StateTransitionReason": "Client.UserInitiated",
 "StateTransitionTime": "2022-01-27T22:16:03.199000+00:00"
}
```

PowerShell

EC2 Fast Launch를 활성화하는 방법

Enable-EC2FastLaunch cmdlet을 사용하여 지정된 AMI에 대한 EC2 Fast Launch를 활성화하며, 사전 프로비저닝을 위한 6개의 병렬 인스턴스를 시작합니다.

```
Enable-EC2FastLaunch `
 -ImageId ami-0abcdef1234567890 `
 -MaxParallelLaunch 6 `
 -Region us-west-2 `
 -ResourceType snapshot
```

출력의 예시는 다음과 같습니다.

```
ImageId               : ami-0abcdef1234567890
 LaunchTemplate       :
 MaxParallelLaunches  : 6
 OwnerId              : 0123456789123
 ResourceType         : snapshot
 SnapshotConfiguration : Amazon.EC2.Model.FastLaunchSnapshotConfigurationResponse
 State                : enabling
 StateTransitionReason : Client.UserInitiated
 StateTransitionTime  : 2/25/2022 12:24:11 PM
```

EC2 Fast Launch 비활성화

이러한 설정을 변경하기 전에 AMI와 실행하는 리전이 Windows용 EC2 Fast Launch 사전 조건을 모두 충족하는지 확인하세요.

Console

EC2 Fast Launch를 비활성화하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창의 이미지(Images) 아래에서 AMI를 선택합니다.
3. 이름(Name) 옆의 확인란을 선택하여 업데이트할 AMI를 선택합니다.
4. AMI 목록 위에 있는 작업 메뉴에서 빠른 시작 구성을 선택합니다. 그러면 EC2 Fast Launch의 설정을 구성할 수 있는 빠른 시작 구성 페이지가 열립니다.
5. EC2 Fast Launch를 비활성화하고 사전 프로비저닝된 스냅샷을 제거하려면 Windows용 빠른 시작 활성화 확인란 선택을 취소합니다. 그러면 앞으로 AMI가 각 인스턴스에 대해 표준 시작 프로세스를 사용하게 됩니다.

> **Note**
>
> Windows 이미지 최적화를 비활성화하면 기존의 사전 프로비저닝된 스냅샷이 자동으로 삭제됩니다. 기능을 다시 사용하기 시작하려면 이 단계를 완료해야 합니다.

6. 변경 작업을 마치면 변경 사항 저장(Save changes)을 선택합니다.

AWS CLI

EC2 Fast Launch를 비활성화하는 방법

다음 disable-fast-launch 명령을 사용하여 지정된 AMI에서 EC2 Fast Launch를 비활성화하고 사전 프로비저닝된 기존 스냅샷을 정리합니다.

```
aws ec2 disable-fast-launch --image-id ami-01234567890abcedf
```

출력의 예시는 다음과 같습니다.

```
{
    "ImageId": "ami-01234567890abcedf",
    "ResourceType": "snapshot",
    "SnapshotConfiguration": {},
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-01234567890abcedf",
        "LaunchTemplateName": "EC2FastLaunchDefaultResourceCreation-
a8c6215d-94e6-441b-9272-dbd1f87b07e2",
        "Version": "1"
    },
    "MaxParallelLaunches": 6,
    "OwnerId": "0123456789123",
    "State": "disabling",
    "StateTransitionReason": "Client.UserInitiated",
    "StateTransitionTime": "2022-01-27T22:47:29.265000+00:00"
}
```

PowerShell

EC2 Fast Launch를 비활성화하는 방법

Disable-EC2FastLaunch cmdlet을 사용하여 지정된 AMI에서 EC2 Fast Launch를 비활성화하고 사전 프로비저닝된 기존 스냅샷을 정리합니다.

```
Disable-EC2FastLaunch -ImageId ami-0abcdef1234567890
```

출력의 예시는 다음과 같습니다.

```
ImageId               : ami-0abcdef1234567890
 LaunchTemplate        :
 Amazon.EC2.Model.FastLaunchLaunchTemplateSpecificationResponse
 MaxParallelLaunches   : 6
 OwnerId               : 0123456789123
 ResourceType          : snapshot
 SnapshotConfiguration :
 State                 : disabling
 StateTransitionReason : Client.UserInitiated
 StateTransitionTime   : 2/25/2022 1:10:08 PM
```

EC2 Fast Launch을 설정할 때 시작 템플릿 사용

시작 템플릿을 사용하면 Amazon EC2가 해당 템플릿에서 인스턴스를 시작할 때마다 사용하는 시작 파라미터 세트를 구성할 수 있습니다. 기본 이미지, 인스턴스 유형, 스토리지, 네트워크 설정 등에 사용할 AMI와 같은 항목을 지정할 수 있습니다.

빠른 시작을 구성할 때 Windows AMI에 대한 시작 템플릿을 사용해야 하는 다음과 같은 특정 경우를 제외하고 시작 템플릿은 선택 사항입니다.

- Windows AMI에 대한 기존 VPC를 지정하려면 시작 템플릿을 사용해야 합니다. AWS 계정에 기본 VPC를 사용하는 경우에는 적용되지 않습니다.
- 계정에 Amazon EC2 인스턴스용 IMDSv2를 적용하는 정책이 포함되어 있는 경우, IMDSv2를 적용하도록 메타데이터 구성을 지정하는 시작 템플릿을 생성해야 합니다.

  EC2 콘솔에서 혹은 AWS CLI에서 enable-fast-launch 명령을 실행하거나 EnableFastLaunch API 작업을 호출하는 경우 메타데이터 구성을 포함한 시작 템플릿을 사용합니다.

Amazon EC2 EC2 Fast Launch은 시작 템플릿을 사용할 때 다음 구성을 지원하지 않습니다. EC2 Fast Launch용 시작 템플릿을 사용하는 경우 다음을 지정해서는 안 됩니다.

- 사용자 데이터 스크립트
- 종료 방지
- 비활성화된 메타데이터

- 스팟 옵션
- 인스턴스를 종료하는 종료 동작
- 네트워크 인터페이스, 탄력적 그래픽 또는 스팟 인스턴스 요청에 대한 리소스 태그

VPC 지정

1단계: 시작 템플릿 생성

Windows 인스턴스에 대한 다음과 같은 세부 정보를 지정하는 시작 템플릿을 생성합니다.

- VPC 서브넷.
- `t3.xlarge` 유형의 인스턴스.

자세한 내용은 Amazon EC2 시작 템플릿 생성 섹션을 참조하세요.

2단계: EC2 Fast Launch AMI에 대한 시작 템플릿 지정

Console

EC2 Fast Launch AMI에 대한 시작 템플릿을 지정하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창의 이미지(Images) 아래에서 AMI를 선택합니다.
3. 이름(Name) 옆의 확인란을 선택하여 업데이트할 AMI를 선택합니다.
4. AMI 목록 위에 있는 작업 메뉴에서 빠른 시작 구성을 선택합니다. 그러면 EC2 Fast Launch의 설정을 구성할 수 있는 빠른 시작 구성 페이지가 열립니다.
5. 이 시작 템플릿 상자는 현재 리전의 계정에서 입력한 텍스트와 일치하는 시작 템플릿을 찾는 필터링된 검색을 수행합니다. 상자에 시작 템플릿 이름 또는 ID의 전부 또는 일부를 지정하여 일치하는 시작 템플릿 목록을 표시합니다. 예를 들어 `fast`를 상자에 입력하면, Amazon EC2는 현재 리전의 계정에서 이름에 “fast”가 있는 모든 시작 템플릿을 찾습니다.

   시작 템플릿을 생성하려면 시작 템플릿 생성(Create launch template)을 선택합니다.
6. 시작 템플릿을 선택하면 Amazon EC2는 해당 템플릿의 기본 버전을 소스 템플릿 버전 상자에 표시합니다. 다른 버전을 지정하려면 대체할 기본 버전을 강조 표시하고 상자에 원하는 버전 번호를 입력합니다.
7. 변경 작업을 마치면 변경 사항 저장(Save changes)을 선택합니다.

AWS CLI

EC2 Fast Launch AMI에 대한 시작 템플릿을 지정하는 방법

enable-fast-launch 명령을 `--launch-template` 옵션과 함께 사용하여 시작 템플릿의 이름 또는 ID를 지정합니다.

```
--launch-template LaunchTemplateName=my-launch-template
```

PowerShell

EC2 Fast Launch AMI에 대한 시작 템플릿을 지정하는 방법

Enable-EC2FastLaunch cmdlet을 `-LaunchTemplate_LaunchTemplateId` 또는 `-LaunchTemplate_LaunchTemplateName` 파라미터와 함께 사용합니다.

```
-LaunchTemplate_LaunchTemplateName my-launch-template
```

EC2 시작 템플릿에 대한 자세한 내용은 Amazon EC2 시작 템플릿에 인스턴스 시작 파라미터 저장를 참조하세요.

EC2 Fast Launch가 활성화된 Windows AMI 보기

AWS CLI에서 describe-fast-launch-images 명령 또는 Get-EC2FastLaunchImage Tools for PowerShell Cmdlet를 사용하여 EC2 Fast Launch가 활성화된 Windows AMI에 대한 세부 정보를 얻을 수 있습니다.

Amazon EC2 결과에 반환되는 각 Windows AMI에 대해 다음과 같은 세부 정보를 제공합니다.

- EC2 Fast Launch가 활성화된 AMI의 이미지 ID.
- 연결된 Windows AMI를 사전 프로비저닝하는 데 사용되는 리소스 유형. 지원되는 값: `snapshot`.
- 스냅샷을 사용하여 연결된 Windows AMI의 사전 프로비저닝을 구성하는 파라미터 그룹인 스냅샷 구성.
- 연결된 AMI가 사전 프로비저닝된 스냅샷에서 Window 인스턴스를 시작할 때 사용하는 시작 템플릿의 ID, 이름 및 버전을 포함한 시작 템플릿 정보.
- 리소스 생성을 위해 동시에 시작할 수 있는 최대 병렬 인스턴스 수.
- 연결된 AMI의 소유자 ID. 본인과 공유되는 AMI의 경우에는 채워지지 않습니다.

- 연결된 AMI의 EC2 Fast Launch 현재 상태. 지원되는 값에는 `enabling | enabling-failed | enabled | enabled-failed | disabling | disabling-failed`가 포함됩니다.

> **Note**
>
> 또한 EC2 콘솔의 이미지 최적화 관리(Manage image optimization) 페이지에 이미지 최적화 상태로 표시된 현재 상태를 볼 수 있습니다.

- 연결된 AMI의 EC2 Fast Launch가 현재 상태로 변경된 이유.
- 연결된 AMI의 EC2 Fast Launch가 현재 상태로 변경된 시각.

AWS CLI

EC2 Fast Launch용으로 구성된 AMI를 찾는 방법

다음 describe-fast-launch-images 명령을 사용하여 EC2 Fast Launch용으로 구성된 계정의 각 AMI에 대한 세부 정보를 설명합니다. 이 예에서는 EC2 Fast Launch을 위해 계정의 AMI가 하나만 구성되어 있습니다.

```
aws ec2 describe-fast-launch-images
```

출력의 예시는 다음과 같습니다.

```
{
    "FastLaunchImages": [
        {
            "ImageId": "ami-01234567890abcedf",
            "ResourceType": "snapshot",
            "SnapshotConfiguration": {},
            "LaunchTemplate": {
                "LaunchTemplateId": "lt-01234567890abcedf",
                "LaunchTemplateName": "EC2FastLaunchDefaultResourceCreation-
a8c6215d-94e6-441b-9272-dbd1f87b07e2",
                "Version": "1"
            },
            "MaxParallelLaunches": 6,
            "OwnerId": "0123456789123",
            "State": "enabled",
            "StateTransitionReason": "Client.UserInitiated",
            "StateTransitionTime": "2022-01-27T22:20:06.552000+00:00"
        }
```

```
    ]
}
```

PowerShell

EC2 Fast Launch용으로 구성된 AMI를 찾는 방법

다음 Get-EC2FastLaunchImage cmdlet을 사용하여 EC2 Fast Launch용으로 구성된 계정의 각 AMI에 대한 세부 정보를 설명합니다. 이 예에서는 EC2 Fast Launch을 위해 계정의 AMI가 하나만 구성되어 있습니다.

```
Get-EC2FastLaunchImage -ImageId ami-0abcdef1234567890
```

출력의 예제는 다음과 같습니다.

```
ImageId               : ami-0abcdef1234567890
LaunchTemplate        :
 Amazon.EC2.Model.FastLaunchLaunchTemplateSpecificationResponse
MaxParallelLaunches   : 6
OwnerId               : 012345678912
ResourceType          : snapshot
SnapshotConfiguration :
State                 : enabled
StateTransitionReason : Client.UserInitiated
StateTransitionTime   : 2/25/2022 12:54:43 PM
```

EC2 Fast Launch의 기본 리소스 비용 관리

EC2 Fast Launch을 위해 Windows AMI를 구성하는 데 드는 서비스 요금은 없습니다. 그러나 Amazon EC2 Windows AMI에 대해 EC2 Fast Launch를 활성화하면 Amazon EC2에서 사전 프로비저닝된 스냅샷을 준비하고 저장하는 데 사용하는 기본 AWS 리소스에는 표준 요금이 적용됩니다. 비용 할당 태그를 구성하면 EC2 Fast Launch 리소스와 관련된 비용을 추적하고 관리하는 데 도움이 될 수 있습니다. 비용 할당 태그를 구성하는 방법에 대한 자세한 내용은 청구서에서 EC2 Fast Launch 비용 추적 섹션을 참조하세요.

다음 예에서는 EC2 Fast Launch 스냅샷 비용과 관련된 비용을 할당하는 방법을 보여줍니다.

예시 시나리오: AtoZ Example 회사에는 50GiB EBS 루트 볼륨이 있는 Windows AMI가 있습니다. AMI에 대한 EC2 Fast Launch를 더 빠르게 시작하고 대상 리소스 수를 5로 설정합니다. 한 달 동안 AMI에 대해 EC2 Fast Launch를 사용하는 데 약 5.00 USD가 소요되며 비용 내역은 다음과 같습니다.

1. AtoZ 예시를 사용하여 EC2 Fast Launch이 가능하면 Amazon EC2는 5개의 스몰 인스턴스를 시작합니다. 각 인스턴스는 Sysprep 및 OOBE Windows 시작 단계를 통해 실행되며 필요에 따라 재부팅됩니다. 각 인스턴스에 대해 몇 분 정도 걸립니다(시간은 해당 리전 또는 가용 영역(AZ)의 사용 빈도와 AMI 크기에 따라 달라질 수 있음).

   비용

   - 인스턴스 런타임 비용(또는 해당하는 경우 최소 런타임): 5개 인스턴스
   - 볼륨 비용: 5개의 EBS 루트 볼륨

2. 사전 프로비저닝 프로세스가 완료되면 Amazon EC2는 Simple Storage Service(Amazon S3)에 저장하는 인스턴스의 스냅샷을 생성합니다. 스냅샷은 일반적으로 시작에 사용되기 전에 4-8시간 동안 저장됩니다. 이 경우 비용은 스냅샷당 약 0.02~0.05 USD입니다.

   비용

   - 스냅샷 스토리지(Amazon S3): 스냅샷 5개

3. Amazon EC2가 스냅샷을 생성한 후 인스턴스를 중지합니다. 이 시점에서 인스턴스는 더 이상 비용을 발생시키지 않습니다. 그러나 EBS 거래량 비용은 계속 발생합니다.

   비용

   - EBS 볼륨: 관련 EBS 루트 볼륨에 대한 비용은 계속 발생합니다.

> **Note**
>
> 여기에 표시된 비용은 데모용일 뿐입니다. 비용은 AMI 구성 및 요금제에 따라 달라집니다.

청구서에서 EC2 Fast Launch 비용 추적

비용 할당 태그를 사용하여 EC2 Fast Launch와 관련된 비용을 반영하도록 AWS 청구서를 구성할 수 있습니다. Amazon EC2가 EC2 Fast Launch를 위해 사전 프로비저닝된 스냅샷을 준비하고 저장할 때 생성하는 리소스에 추가하는 다음 태그를 사용할 수 있습니다.

태그 키: `CreatedBy`, 값: `EC2 Fast Launch`

Billing and Cost Management 콘솔에서 태그를 활성화하고 세부 결제 보고서를 설정하면 보고서에 `user:CreatedBy` 열이 나타납니다. 이 열에는 모든 서비스의 값이 포함됩니다. 그러나 CSV 파일을 다운로드하면 데이터를 스프레드시트로 가져오고 값에서 `EC2 Fast Launch`를 필터링할 수 있습니다. 이 정보는 태그가 활성화되면 AWS Cost and Usage Report에도 나타납니다.

1단계: 사용자 정의 비용 할당 태그 활성화

비용 보고서에 리소스 태그를 포함하려면 먼저 Billing and Cost Management 콘솔에서 태그를 활성화해야 합니다. 태그 활성화에 대한 자세한 내용은 AWS 결제 및 비용 관리 사용 설명서에서 사용자 정의 비용 할당 태그 활성화를 참조하세요.

> **Note**
>
> 활성화에는 최대 24시간이 소요될 수 있습니다.

2단계: 비용 보고서 설정

비용 보고서를 이미 설정한 경우 활성화가 완료된 후 다음에 보고서를 실행할 때 태그 열이 나타납니다. 비용 보고서를 처음으로 설정하려면 다음 중 하나를 선택합니다.

- 자세한 내용은 AWS 결제 및 비용 관리 사용 설명서의 월별 비용 할당 보고서 설정을 참조하세요.
- AWS Cost and Usage Report 사용 설명서의 비용 및 사용 보고서 생성을 참조하세요.

> **Note**
>
> AWS가 S3 버킷에 보고서 전송을 시작하는 데 최대 24시간이 걸릴 수 있습니다.

AWS CLI의 Amazon EC2 콘솔, API, SDK, CloudFormation 또는 ec2 명령에서 사용자가 소유한 Windows AMI 또는 사용자와 공유하는 AMI에 대해 EC2 Fast Launch를 구성할 수 있습니다. 다음 섹션에서는 Amazon EC2 콘솔 및 AWS CLI에 대한 구성 단계를 다룹니다.

또한 EC2 Image Builder를 사용하여 EC2 Fast Launch를 위해 구성된 사용자 지정 Windows AMI를 생성할 수 있습니다. 자세한 내용은 EC2 Fast Launch가 활성화된 Windows AMI에 대한 배포 설정 생성(AWS CLI)을 참조하세요.

EC2 Fast Launch 모니터링

이 섹션에서는 EC2 Fast Launch가 활성화된 계정에서 Amazon EC2 Windows Server AMI를 모니터링하는 방법을 설명합니다.

EventBridge를 사용하여 EC2 Fast Launch 상태 변경 모니터링

EC2 Fast Launch가 활성화된 Windows AMI의 상태가 변경되면 Amazon EC2가 `EC2 Fast Launch State-change Notification` 이벤트를 생성합니다. 그런 다음 Amazon EC2는 상태 변경 이벤트를 Amazon EventBridge(이전에는 Amazon CloudWatch Events)로 전송합니다.

상태 변경 이벤트에 대한 응답으로 하나 이상의 작업을 트리거하는 EventBridge 규칙을 생성할 수 있습니다. 예를 들어 EC2 Fast Launch가 활성화된 시점을 감지하고 다음 작업을 수행하는 EventBridge 규칙을 생성할 수 있습니다.

- 구독자에게 알리는 메시지를 Amazon SNS 토픽으로 보냅니다.
- 일부 작업을 수행하는 Lambda 함수를 호출합니다.
- 분석을 위해 상태 변경 데이터를 Amazon Data Firehose로 전송합니다.

자세한 내용을 알아보려면 Amazon EventBridge 사용 설명서의 이벤트에 응답하는 Amazon EventBridge 규칙 생성을 참조하세요.

상태 변경 이벤트

EC2 Fast Launch 기능은 최선의 노력을 기준으로 JSON 형식의 상태 변경 이벤트를 내보냅니다. Amazon EC2는 거의 실시간으로 이벤트를 EventBridge로 전송합니다. 이 섹션에서는 이벤트 필드를 설명하고 이벤트 형식의 예를 보여줍니다.

`EC2 Fast Launch State-change Notification`

imageId

EC2 Fast Launch 기능 상태 변경을 통해 AMI를 식별합니다.

resourceType

사전 프로비저닝에 사용할 리소스 유형입니다. 지원되는 값: `snapshot`. 기본값은 `snapshot`입니다.

state

지정된 AMI에 대한 EC2 Fast Launch 기능의 현재 상태입니다. 유효한 값은 다음과 같습니다.

- enabling – AMI에 대한 EC2 Fast Launch 기능을 활성화했으며 Amazon EC2에서 사전 프로비저닝 프로세스를 위한 스냅샷을 생성하기 시작했습니다.

- enabling-failed – 문제가 발생하여 AMI에 대해 EC2 Fast Launch 기능을 처음 사용하도록 설정할 때 사전 프로비저닝 프로세스가 실패했습니다. 이는 사전 프로비저닝 프로세스 중에 언제든지 발생할 수 있습니다.
- enabled – EC2 Fast Launch 기능이 활성화되었습니다. Amazon EC2가 새로 활성화된 EC2 Fast Launch AMI를 위해 사전 프로비저닝된 첫 번째 스냅샷을 생성하는 즉시 상태가 `enabled`로 변경됩니다. AMI가 이미 활성화되어 있고 사전 프로비저닝을 다시 거치면 상태가 즉시 변경됩니다.
- enabled-failed – 이 상태는 EC2 Fast Launch AMI가 사전 프로비저닝 프로세스를 거치는 것이 처음이 아닌 경우에만 적용됩니다. 이는 EC2 Fast Launch 기능을 사용하지 않도록 설정한 후 나중에 다시 사용하도록 설정하거나 사전 프로비저닝을 처음 완료한 후 구성이 변경되거나 기타 오류가 발생한 경우에 발생할 수 있습니다.
- disabling – AMI 소유자가 AMI에 대한 EC2 Fast Launch 기능을 해제했으며 Amazon EC2에서 정리 프로세스를 시작했습니다.
- disabled – EC2 Fast Launch 기능이 비활성화되었습니다. Amazon EC2가 정리 프로세스를 완료하자마자 상태가 `disabled`로 변경됩니다.
- disabling-fail – 문제가 발생하여 정리 프로세스가 실패했습니다. 즉, 사전 프로비저닝된 일부 스냅샷은 여전히 계정에 남아 있을 수 있습니다.

stateTransitionReason

EC2 Fast Launch AMI 상태가 변경된 이유입니다.

> **Note**
>
> 이 이벤트 메시지의 모든 필드는 필수입니다.

다음 예는 사전 프로비저닝 프로세스를 시작하는 첫 번째 인스턴스를 시작한 새로 활성화된 EC2 Fast Launch AMI를 보여줍니다. 현재 상태는 `enabling`과 같습니다. Amazon EC2에서 사전 프로비저닝된 첫 번째 스냅샷을 생성하면 상태가 `enabled`로 변경됩니다.

```
{
 "version": "0",
 "id": "01234567-0123-0123-0123-012345678901",
 "detail-type": "EC2 Fast Launch State-change Notification",
 "source": "aws.ec2",
 "account": "123456789012",
 "time": "2022-08-31T20:30:12Z",
```

```
 "region": "us-east-1",
 "resources": [
  "arn:aws:ec2:us-east-1:123456789012:image/ami-123456789012"
 ],
 "detail": {
  "imageId": "ami-123456789012",
  "resourceType": "snapshot",
  "state": "enabling",
  "stateTransitionReason": "Client.UserInitiated"
 }
}
```

CloudWatch를 사용하여 EC2 Fast Launch 지표 모니터링

EC2 Fast Launch가 활성화된 Amazon EC2 AMI는 Amazon CloudWatch로 지표를 전송합니다. AWS Management Console, AWS CLI 또는 API를 사용하여 EC2 Fast Launch가 CloudWatch로 전송하는 지표를 나열할 수 있습니다. AWS/EC2 네임스페이스에는 다음과 같은 EC2 Fast Launch 지표가 포함되어 있습니다.

지표	설명
NumberOfAvailableFastLaunchSnapshots	EC2 Fast Launch를 지원하는 AMI당 사용할 수 있는 사전 프로비저닝된 스냅샷 수입니다.
NumberOfInstancesFastLaunched	사전 프로비저닝된 스냅샷에서 시작되었으며 EC2 Fast Launch를 지원하는 AMI당 인스턴스 수입니다.
NumberOfInstancesNotFastLaunched	시작 시 사용 가능한 사전 프로비저닝된 스냅샷이 부족하여 콜드 부팅이 발생한 EC2 Fast Launch를 지원하는 AMI당 인스턴스 수입니다.
FastLaunchSnapshotUsedToRefillStartTime	기존 스냅샷을 사용한 후 다른 스냅샷을 생성하기 위해 Amazon EC2가 EC2 Fast Launch를 지원하는 AMI에서 새 이미지를 시작한 시점의 타임스탬프입니다.
FastLaunchSnapshotCreationTime	Amazon EC2가 인스턴스를 시작하고 EC2 Fast Launch를 지원하는 AMI용 스냅샷을 생성하는 데 걸린 시간을 측정합니다.

EC2 Fast Launch를 위한 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 수임할 수 있으므로 서비스 연결 역할은 AWS 서비스에 권한을 위임하는 안전한 방법을 제공합니다. Amazon EC2가 서비스 연결 역할을 포함하여 IAM 역할을 사용하는 방법에 대한 자세한 내용은 Amazon EC2의 IAM 역할 섹션을 참조하세요.

Amazon EC2는 AWSServiceRoleForEC2FastLaunch라는 서비스 연결 역할을 사용하여 Windows AMI에서 인스턴스를 시작하는 데 걸리는 시간을 줄이는 사전 프로비저닝된 스냅샷 세트를 생성하고 관리합니다.

AWSServiceRoleForEC2FastLaunch으로 부여된 권한

AWSServiceRoleForEC2FastLaunch 서비스 연결 역할은 그 역할을 위임하기 위해 다음 서비스를 신뢰합니다.

- `ec2fastlaunch.amazonaws.com`

Amazon EC2에서는 EC2FastLaunchServiceRolePolicy 관리형 정책을 사용하여 다음 작업을 완료할 수 있습니다.

- AWS CloudFormation - EC2 Fast Launch가 연결된 CloudFormation 스택에 대한 설명을 가져오도록 허용합니다.
- Amazon CloudWatch - EC2 Fast Launch와 관련된 지표 데이터를 Amazon EC2 네임스페이스에 게시합니다.
- Amazon EC2 - EC2 Fast Launch가 다음 작업을 수행할 수 있는 액세스 권한이 부여됩니다.
  - 프로비저닝 단계를 수행하기 위해 EC2 Fast Launch가 활성화된 Amazon EC2 Windows Server AMI에서 인스턴스를 시작합니다. 또한 License Manager와 연결된 AMI에 `ec2:RunInstances`를 허용하는 리소스 패턴을 지정합니다.
  - 사전 프로비저닝된 스냅샷을 생성한 후 EC2 Fast Launch에서 시작한 인스턴스를 중지하고 종료합니다.
  - EC2 Fast Launch가 활성화된 Amazon EC2 Windows Server AMI에서 인스턴스를 시작하고 해당 인스턴스에서 스냅샷을 생성하는 데 사용되는 이미지와 인스턴스 유형 리소스를 설명합니다.
  - 시작 템플릿 리소스와 시작 템플릿의 시작 인스턴스를 설명합니다.
  - 인스턴스, 인스턴스 속성, 인스턴스 상태를 설명합니다
  - 스냅샷과 시작 템플릿을 포함하여 EC2 Fast Launch가 생성한 리소스를 삭제합니다.

  - EC2 Fast Launch가 시작과 사전 프로비저닝, Windows 인스턴스를 위해 생성하는 리소스에 태그를 지정하고 최종 시작 프로세스가 사용할 스냅샷을 생성합니다.
- Amazon EventBridge - EventBridge 이벤트 규칙을 생성하고 생성한 규칙에 대한 세부 정보를 검색하거나 삭제할 수 있는 액세스 권한을 포함합니다. EC2 Fast Launch는 이벤트 규칙에 따라 전달되는 EC2 Fast Launch 이벤트를 수신하는 대상 서비스 목록을 가져오고 생성된 이벤트 규칙에 대상 서비스를 추가하거나 제거할 수도 있습니다.
- IAM - EC2 Fast Launch가 `EC2FastLaunchServiceRolePolicy` 서비스 연결 역할을 생성하고, 이름에 `ec2fastlaunch`가 포함된 인스턴스 프로파일을 가져오고 사용하며, 시작 템플릿의 인스턴스 프로파일을 사용하여 사용자를 대신하여 인스턴스를 시작하도록 허용합니다.
- AWS KMS - 권한을 생성할 수 있는 권한과 EC2 Fast Launch에서 생성한 권한 중 사용 중지할 수 있는 권한을 나열할 수 있는 권한을 포함합니다. 또한 EC2 Fast Launch가 생성하는 인스턴스에 연결된 볼륨을 암호화하거나 해독할 목적으로 키를 설명하거나 사용하고 일반 텍스트가 아닌 데이터 키를 생성합니다.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 EC2FastLaunchServiceRolePolicy를 참조하세요.

Amazon EC2에 관리형 정책 사용에 대한 자세한 내용은 Amazon EC2에 대한 AWS 관리형 정책 섹션을 참조하세요.

서비스 연결 역할 생성

이 서비스 연결 역할을 수동으로 생성할 필요가 없습니다. AMI에 EC2 Fast Launch를 사용하기 시작하면 Amazon EC2가 서비스 연결 역할을 생성합니다(아직 없는 경우).

서비스 연결 역할이 계정에서 삭제되면 다른 Windows AMI가 계정에 역할을 다시 생성할 수 있도록 EC2 Fast Launch를 활성화할 수 있습니다. 또는 현재 AMI에 대해 EC2 Fast Launch를 비활성화했다가 다시 활성화할 수 있습니다. 그러나 이 기능을 비활성화하면 AMI가 모든 새 인스턴스에 표준 시작 프로세스를 사용하게 되고 Amazon EC2는 사전 프로비저닝된 스냅샷을 모두 제거합니다. 사전 프로비저닝된 모든 스냅샷이 없어지면 AMI를 위한 EC2 Fast Launch를 다시 활성화할 수 있습니다.

고객 관리형 키에 대한 액세스

암호화를 위해 고객 관리형 키를 사용하는 암호화된 AMI에 대해 EC2 Fast Launch를 활성화하려면 AWSServiceRoleForEC2FastLaunch 역할에 CMK를 사용할 수 있는 권한을 부여해야 합니다. 이렇게 하려면 create-grant 명령을 직접적으로 호출합니다. `--grantee-principal`에서 계정의 AWSServiceRoleForEC2FastLaunch 역할에 대한 ARN을 지정합니다. `--operations`에서 `CreateGrant`를 지정합니다.

```
aws kms create-grant \
    --key-id arn:aws:kms:region:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/AWSServiceRoleForEC2FastLaunch \
    --operations CreateGrant
```

서비스 연결 역할 편집

Amazon EC2에서는 AWSServiceRoleForEC2FastLaunch 서비스 연결 역할을 편집하도록 허용하지 않습니다. 서비스 연결 역할을 생성한 후에는 다양한 개체가 역할을 참조할 수 있기 때문에 역할 이름을 변경할 수 없습니다. 그러나 IAM을 사용하여 역할의 설명을 편집할 수 있습니다. 자세한 정보는 IAM 사용 설명서의 서비스 연결 역할 편집을 참조하세요.

서비스 연결 역할 삭제

먼저 관련 리소스를 모두 삭제한 후에만 서비스 연결 역할을 삭제할 수 있습니다. 이렇게 하면 리소스에 대한 액세스 권한을 실수로 제거할 수 없으므로 EC2 Fast Launch가 활성화된 Amazon EC2 Windows Server AMI와 연결된 Amazon EC2 리소스가 보호됩니다.

IAM 콘솔, AWS CLI 또는 AWS API를 사용하여 AWSServiceRoleForEC2FastLaunch 서비스 연결 역할을 삭제합니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 삭제를 참조하세요.

지원되는 리전

Amazon EC2는 Amazon EC2 서비스를 사용할 수 있는 모든 리전에서 EC2 Fast Launch를 지원합니다.

Windows EC2 Fast Launch 문제 해결

문제 해결 시나리오

다음 시나리오를 참고하여 EC2 Fast Launch를 활성화하려고 할 때 발생할 수 있는 일반적인 문제를 진단하고 해결할 수 있습니다.

스냅샷 생성을 위해 인스턴스를 중지할 수 없음

설명

EC2 Fast Launch를 활성화하면 서비스가 사전 프로비저닝된 스냅샷을 생성하는 데 사용되는 인스턴스 세트를 시작합니다. 각 인스턴스는 프로세스를 완료하는 데 30분이 소요됩니다. 인스턴스 중 하나라도 성공적으로 완료되면 서비스는 AMI의 Fast Launch 상태를 `Enabled`로 설정합니다. 그러나 인스턴스가 할당된 시간 내에 프로세스를 완료하지 못하고 다른 인스턴스도 모두 프로세스를 완료하지 못한

경우 서비스는 모든 인스턴스를 종료하고 AMI의 Fast Launch 상태를 `enabling_failed`로 설정하고 Fast Launch 상태 이유를 다음과 같이 설정합니다.

```
Unable to stop instance ID=i-1234567890abcdef0 for snapshot creation.
```

원인

이는 실행 중인 인스턴스에서 생성된 Windows AMI 또는 모든 EC2 Fast Launch 사전 조건를 충족하지 않는 AMI에 대해 EC2 Fast Launch를 활성화하려고 할 때 발생하는 경우가 많습니다.

Solution

사용하는 AMI가 모든 EC2 Fast Launch 사전 조건를 충족하는지 확인합니다.

AMI에 EC2 Fast Launch를 구성하려면 종료 옵션과 함께 Sysprep을 사용하여 생성해야 합니다. 자세한 내용은 Windows Sysprep을 사용하여 Amazon EC2 AMI 생성 섹션을 참조하세요.

VPC 한도에 도달

설명

시작 템플릿을 사용하여 기존 VPC를 지정하지 않고 계정에 기본 VPC가 정의되지 않은 경우 서비스는 EC2 Fast Launch 사전 조건에 설명된 대로 VPC 및 기타 리소스를 포함하는 AWS CloudFormation 스택을 자동으로 생성합니다.

원인

리전에 대해 AWS 계정에서 허용되는 최대 VPC 수에 도달했으며 EC2 Fast Launch에서 사용할 기존 VPC를 지정하지 않았습니다. 이로 인해 프로세스가 실패합니다.

Solution

다음 옵션 중 하나를 사용하여 문제를 해결할 수 있습니다.

- 할당량 증가 요청 가능
- 기존 VPC를 지정하는 시작 템플릿 제공 가능

계정에서 리전별로 정의할 수 있는 VPC 수를 늘리도록 요청하려면 다음 단계를 따르세요.

1. https://console.aws.amazon.com/servicequotas/에서 Service Quotas 콘솔을 엽니다.

2. 서비스 콘솔 대시보드에서 Amazon Virtual Private Cloud(Amazon VPC)를 선택합니다. 그러면 VPC 서비스 할당량이 열립니다.
3. `VPCs per Region`을 필터링하여 할당량으로 바로 이동합니다.
4. 리전당 VPC를 선택하고 계정 수준에서 증가 요청을 선택합니다.

긴급 할당량 요청이 있거나 할당량 증가 요청이 거부된 경우 지원팀에 문의하여 지원을 받으세요. 자세한 내용은 Service Quotas 사용 설명서의 할당량 증가 요청을 참조하세요.

EC2 Fast Launch를 활성화할 수 있는 권한 부족

설명

시작 템플릿을 지정하지 않고 EC2 Fast Launch를 처음 활성화하면 EC2 Fast Launch는 서비스 기본 리소스가 있는 서비스 소유 AWS CloudFormation 스택을 생성합니다. 그러나 IAM 위탁자(역할 또는 사용자)에 필요한 권한이 없는 경우 CloudFormation 템플릿이 배포되지 않습니다.

로그 메시지가 다음과 같을 수 있습니다.

```
Can't enable EC2 Fast Launch. The IAM credentials that you are using do not have
 sufficient permissions. Attach EC2FastLaunchFullAccess in the IAM console.
```

원인

IAM 사용자 또는 역할에 EC2 Fast Launch를 활성화하는 데 필요한 권한이 없습니다.

Solution

EC2 Fast Launch를 활성화하는 IAM 위탁자(사용자 또는 역할)에 `EC2FastLaunchFullAccess` 정책이 연결되어 있는지 확인합니다. 이 AWS 관리형 정책은 모든 EC2 Fast Launch 리소스에 대한 전체 액세스 권한을 부여합니다. 이 정책의 권한을 보려면 AWS 관리형 정책 참조의 EC2FastLaunchFullAccess를 참조하세요.

Amazon EC2 인스턴스에 대한 Windows 관리자 암호 변경

AWS Windows AMI에서 인스턴스를 시작하는 경우 사전 설치된 시작 에이전트는 다음과 같이 기본 암호를 설정합니다.

- Windows Server 2022 이상의 경우 EC2Launch v2가 기본 암호를 생성합니다.
- Windows Server 2016 및 2019의 경우 EC2Launch 에이전트가 기본 암호를 생성합니다.
- Windows Server 2012 R2 이하의 경우 EC2Config 서비스가 기본 암호를 생성합니다.

> **Note**
>
> Windows Server 2016 이상 AMI의 경우 로컬 관리자에 대해 `Password never expires`가 비활성화됩니다. Windows Server 2016 이전 AMI 버전의 경우 로컬 관리자에 대해 `Password never expires`가 활성화됩니다.

연결 후 관리자 암호 변경

인스턴스에 처음 연결했을 때 관리자 암호를 기본값과 다르게 변경하는 것이 좋습니다. 다음 절차에 따라 Windows 인스턴스에 대한 관리자 암호를 변경합니다.

> **Important**
>
> 새 암호를 안전한 위치에 저장합니다. Amazon EC2 콘솔을 사용하여 새 암호를 검색할 수 없습니다. 콘솔에서는 기본 암호만 검색할 수 있습니다. 암호를 변경한 이후에 기본 암호를 사용하여 인스턴스에 연결할 경우 "Your credentials did not work."라는 오류가 발생합니다.

로컬 관리자 암호를 변경하려면

1. 인스턴스에 연결하고 명령 프롬프트를 엽니다.
2. 다음 명령을 실행합니다. 새 암호에 특수 문자가 포함된 경우 암호를 큰따옴표로 묶습니다.

```
net user Administrator "new_password"
```

3. 새 암호를 안전한 위치에 저장합니다.

분실 또는 만료된 암호 변경

암호가 기억나지 않거나 만료된 경우 새 암호를 생성할 수 있습니다. 암호 재설정 절차는 Amazon EC2 Windows 인스턴스의 Windows 관리자 암호 재설정을 참조하세요.

Amazon EC2 Windows 인스턴스에 선택적 Windows Server 구성 요소 추가

선택 구성 요소에 액세스하고 설치하려면 현재 Windows Server 버전의 올바른 EBS 스냅샷을 찾고 스냅샷에서 볼륨을 생성하고 인스턴스에 볼륨을 연결해야 합니다.

시작하기 전에

AWS Management Console 또는 명령줄 도구를 사용하여 인스턴스의 인스턴스 ID 및 가용 영역을 가져올 수 있습니다. 인스턴스와 동일한 가용 영역에서 EBS 볼륨을 생성해야 합니다.

다음 절차 중 하나를 사용하여 인스턴스에 Windows 구성 요소를 추가합니다.

Console

인스턴스에 Windows 구성 요소를 추가하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Snapshots]를 선택합니다.
3. 필터(Filter) 막대에서 퍼블릭 스냅샷(Public snapshots)을 선택합니다.
4. 소유자 별칭(Owner Alias) 필터를 추가하고 amazon을 선택합니다.
5. 설명 필터를 추가하고 **Windows**를 입력합니다.
6. Enter를 누릅니다
7. 시스템 아키텍처 및 언어 기본 설정과 일치하는 스냅샷을 선택합니다. 예를 들어 현재 인스턴스가 Windows Server 2019를 실행한다면 Windows 2019 English Installation Media를 선택합니다.
8. 작업(Actions), 스냅샷에서 볼륨 생성(Create volume from snapshot)을 선택합니다.
9. 가용 영역(Availability Zone)에서 Windows 인스턴스와 일치하는 가용 영역을 선택합니다. 태그 추가(Add tag)를 선택하고 태그 키에 **Name**을, 태그 값에 설명이 포함된 이름을 입력합니다. 볼륨 생성을 선택합니다.
10. 볼륨이 생성됨(Successfully created volume) 메시지(녹색 배너)에서 방금 생성한 볼륨을 선택합니다.
11. 작업(Actions), 볼륨 연결(Attach volume)을 선택합니다.
12. 인스턴스(Instance)에서 인스턴스 ID를 선택합니다.
13. 디바이스 이름(Device name)에서 첨부 파일의 디바이스 이름을 입력합니다. 디바이스 이름과 관련하여 도움이 필요하면 Amazon EC2 인스턴스의 볼륨 디바이스 이름 섹션을 참조하세요.
14. 볼륨 연결(Attach Volume)을 선택합니다.
15. 인스턴스에 연결하고 볼륨을 사용 가능하도록 만듭니다. 자세한 내용은 Amazon EBS 사용 설명서의 Make an Amazon EBS volume available for use를 참조하세요.

> **Important**
>
> 볼륨을 초기화하지 마세요.

16. 제어판을 열고 프로그램 및 기능을 엽니다. Windows 기능 사용/사용 안 함을 선택합니다. 설치할 매체를 선택하라는 메시지가 뜨면 EBS 볼륨을 지정합니다.
17. (선택 사항) 설치 미디어에 대한 작업을 마쳤으면 볼륨을 분리할 수 있습니다. 볼륨을 분리한 후 삭제할 수 있습니다.

AWS CLI

인스턴스에 Windows 구성 요소를 추가하려면

1. describe-snapshots 명령을 사용하고 `owner-ids` 파라미터 및 `description` 필터를 적용하여 사용 가능한 설치 미디어 스냅샷의 목록을 가져옵니다.

```
aws ec2 describe-snapshots \
    --owner-ids amazon \
    --filters Name=description,Values=Windows*
```

2. 출력에서 시스템 아키텍처 및 언어 기본 설정과 일치하는 스냅샷의 ID를 적어 둡니다. 예:

```
{
    "Snapshots": [
    ...
        {
            "OwnerAlias": "amazon",
            "Description": "Windows 2019 English Installation Media",
            "Encrypted": false,
            "VolumeId": "vol-be5eafcb",
            "State": "completed",
            "VolumeSize": 6,
            "Progress": "100%",
            "StartTime": "2019-10-25T20:00:47.000Z",
            "SnapshotId": "snap-22da283e",
            "OwnerId": "123456789012"
        },
    ...
   ]
}
```

3. create-volume 명령을 사용하여 스냅샷에서 볼륨을 만듭니다. 인스턴스와 동일한 가용 영역을 지정합니다.

```
aws ec2 create-volume \
```

```
    --snapshot-id snap-0abcdef1234567890 \
    --volume-type gp2 \
    --availability-zone us-east-1a
```

4. 볼륨 ID를 출력에 기록해 둡니다.

```
{
    "AvailabilityZone": "us-east-1a",
    "Encrypted": false,
    "VolumeType": "gp2",
    "VolumeId": "vol-01234567890abcdef",
    "State": "creating",
    "Iops": 100,
    "SnapshotId": "snap-0abcdef1234567890",
    "CreateTime": "2017-04-18T10:33:10.940Z",
    "Size": 6
}
```

5. attach-volume 명령을 사용하여 이 볼륨을 인스턴스에 연결합니다.

```
aws ec2 attach-volume \
    --volume-id vol-0c98b37f30bcbc290 \
    --instance-id i-01474ef662b89480 \
    --device xvdg
```

6. 인스턴스에 연결하고 볼륨을 사용 가능하도록 만듭니다. 자세한 내용은 Amazon EBS 사용 설명서의 Make an Amazon EBS volume available for use를 참조하세요.

> **Important**
>
> 볼륨을 초기화하지 마세요.

7. 제어판을 열고 프로그램 및 기능을 엽니다. Windows 기능 사용/사용 안 함을 선택합니다. 설치할 매체를 선택하라는 메시지가 뜨면 EBS 볼륨을 지정합니다.
8. (선택 사항) 설치 미디어에 대한 작업을 마쳤으면 detach-volume 명령을 사용하여 인스턴스에서 볼륨을 분리합니다. 볼륨을 분리한 후 delete-volume 명령을 사용하여 볼륨을 삭제할 수 있습니다.

PowerShell

인스턴스에 Windows 구성 요소를 추가하려면

1. Get-EC2Snapshot cmdlet와 Owner 및 description 필터를 사용하여 사용 가능한 설치 미디어 스냅샷의 목록을 가져옵니다.

```
Get-EC2Snapshot `
    -Owner amazon `
    -Filter @{ Name="description"; Values="Windows*" }
```

2. 출력에서 시스템 아키텍처 및 언어 기본 설정과 일치하는 스냅샷의 ID를 적어 둡니다. 예:

```
...
DataEncryptionKeyId :
Description         : Windows 2019 English Installation Media
Encrypted           : False
KmsKeyId            :
OwnerAlias          : amazon
OwnerId             : 123456789012
Progress            : 100%
SnapshotId          : snap-0abcdef1234567890
StartTime           : 10/25/2019 8:00:47 PM
State               : completed
StateMessage        :
Tags                : {}
VolumeId            : vol-01234567890abcdef
VolumeSize          : 6
...
```

3. New-EC2Volume cmdlet을 사용하여 스냅샷에서 볼륨을 만듭니다. 인스턴스와 동일한 가용 영역을 지정합니다.

```
New-EC2Volume `
    -AvailabilityZone us-east-1a `
    -VolumeType gp2 `
    -SnapshotId snap-0abcdef1234567890
```

4. 볼륨 ID를 출력에 기록해 둡니다.

```
Attachments      : {}
AvailabilityZone : us-east-1a
```

```
CreateTime       : 4/18/2017 10:50:25 AM
Encrypted        : False
Iops             : 100
KmsKeyId         :
Size             : 6
SnapshotId       : snap-0abcdef1234567890
State            : creating
Tags             : {}
VolumeId         : vol-01234567890abcdef
VolumeType       : gp2
```

5. Add-EC2Volume cmdlet을 사용하여 인스턴스에 볼륨을 연결합니다.

```
Add-EC2Volume `
    -InstanceId i-1234567890abcdef0 `
    -VolumeId vol-01234567890abcdef `
    -Device xvdh
```

6. 인스턴스에 연결하고 볼륨을 사용 가능하도록 만듭니다. 자세한 내용은 Amazon EBS 사용 설명서의 Make an Amazon EBS volume available for use를 참조하세요.

> **Important**
>
> 볼륨을 초기화하지 마세요.

7. 제어판을 열고 프로그램 및 기능을 엽니다. Windows 기능 사용/사용 안 함을 선택합니다. 설치할 매체를 선택하라는 메시지가 뜨면 EBS 볼륨을 지정합니다.
8. (선택 사항) 설치 미디어에 대한 작업을 마쳤으면 Dismount-EC2Volume cmdlet을 사용하여 인스턴스에서 볼륨을 분리합니다. 볼륨을 분리한 후 Remove-EC2Volume cmdlet을 사용하여 볼륨을 삭제할 수 있습니다.

EC2 Windows 인스턴스에 Windows Subsystem for Linux 설치

인스턴스 유형과 인스턴스 운영 체제에 따라 설치할 수 있는 Windows Subsystem for Linux(WSL)에는 WSL 1과 WSL 2의 두 가지 버전이 있습니다. `.metal` 인스턴스 유형의 경우 WSL 1 또는 WSL 2를 설치할 수 있습니다. 다른 모든 인스턴스 유형에는 다음 요구 사항이 적용됩니다.

- 가상화된 EC2 인스턴스의 경우 WSL 1을 설치해야 합니다.
- Windows Server를 실행하는 인스턴스의 경우 WSL을 설치하려면 운영 체제 버전이 다음 중 하나여야 합니다.

- Windows Server 2019
- Windows Server 2022
- Windows Server 2025

자세한 내용은 Microsoft Build 웹 사이트에서 Windows Subsystem for Linux 설명서를 참조하세요.

WSL 설치

다음 지침에서는 Windows Server 2022를 실행하는 EC2 인스턴스에 WSL을 설치합니다. Windows Server 2019를 실행하는 EC2 인스턴스에 WSL을 설치하기 위한 지침은 Microsoft 웹 사이트에서 이전 버전의 Windows Server에 WSL 설치를 참조하세요. 해당 지침을 따른 후 아래 지침의 3단계를 사용하여 WSL 1을 사용하도록 WSL을 구성할 수 있습니다.

WSL 1 설치

1. WSL을 설치하려면 EC2 인스턴스에서 다음 표준 설치 명령을 실행하되 `--enable-wsl1`을 포함하여 WSL 1을 활성화해야 합니다. 기본적으로 WSL 2가 설치됩니다. 가상화된 인스턴스 유형을 사용하여 인스턴스를 시작한 경우 이 절차의 3단계를 완료하여 버전을 WSL 1로 설정해야 합니다.

   ```
   wsl --install --enable-wsl1 --no-launch
   ```

2. EC2 인스턴스를 재시작합니다.

   ```
   shutdown -r -t 20
   ```

3. WSL 1을 사용하도록 WSL을 구성하려면 인스턴스에서 다음 명령을 실행합니다. WSL 버전 설정에 대한 자세한 내용은 Microsoft Build 웹 사이트에서 이전 버전 WSL의 수동 설치 단계를 참조하세요.

   ```
   wsl --set-default-version 1
   ```

4. 기본 배포를 설치합니다.

   ```
   wsl --install
   ```

WSL 2 설치

- WSL을 설치하려면 EC2 인스턴스에서 다음 표준 설치 명령을 실행합니다. 기본적으로 WSL 2가 설치됩니다. `.metal` 인스턴스에 WSL을 설치하는 경우 이 단계만 수행하면 됩니다.

  ```
  wsl --install
  ```

  자세한 내용은 Microsoft Build 웹 사이트에서 WSL을 사용하여 Windows에 Linux 설치하기를 참조하세요.

EC2 Windows 문제 해결 유틸리티

`EC2WinUtil` 드라이버는 Windows 인스턴스에 대해 다음과 같은 유형의 문제 해결 지원을 제공합니다.

충돌 호출 스택

`EC2WinUtil`은 인스턴스에서 기본 충돌 정보를 수집하여 직렬 콘솔에 기록합니다. 다음 목록에는 유틸리티가 콘솔에 기록하는 몇 가지 주요 세부 정보가 포함되어 있습니다.

- 장애를 일으킨 모듈의 식별.
- 이벤트와 연결된 Windows 오류 코드.
- 가장 최근 호출의 스택 트레이스.

이러한 세부 정보를 사용하여 초기 근본 원인 분석을 수행하고 추가 분석이 필요한지 여부를 판단할 수 있습니다. 또한 직렬 콘솔로 출력하면 AWS가 Amazon EC2 드라이버의 충돌 추세를 추적하고 대규모 충돌 이벤트를 진단할 수 있습니다.

> **Note**
>
> `EC2WinUtil`은 충돌 호출 스택에서 고객 데이터를 수집하지 않습니다.

최대 절전 모드/재개 안정성

`EC2WinUtil`은 최대 절전 모드/재개 주기 전반에 걸쳐 인스턴스의 가상화 설정을 추적합니다. 이는 최대 절전 모드를 활성화한 인스턴스의 장기 안정성을 개선하는 데 도움이 됩니다.

드라이버 릴리스 정보는 EC2 Windows 유틸리티 드라이버 버전 기록 섹션을 참조하세요.

EC2 Windows 유틸리티 드라이버 버전 기록

다음 표에서는 Amazon EC2의 각 Windows Server 버전에서 실행되는 `EC2WinUtil` 드라이버를 보여줍니다. 이전 버전의 운영 체제에서는 인스턴스가 시작된 AWS Windows Server AMI에 사전 설치된 드라이버를 사용합니다. AWS Marketplace를 통해 공유되거나 구독하는 AMI에는 드라이버가 사전 설치되어 있지 않습니다.

Windows Server 버전	EC2WinUtil 드라이버 버전
Windows Server 2025	최신 버전
Windows Server 2022	최신 버전
Windows Server 2019	최신 버전
Windows Server 2016	최신 버전

> **Note**
>
> 드라이버 버전 3.0.0 이전에는 수동 설치를 위해 `EC2WinUtil` 드라이버를 다운로드할 수 없었습니다. 이전 버전은 AWS Windows AMI용 사전 설치된 드라이버로만 제공되었습니다.

다음 표에서는 `EC2WinUtil` 드라이버의 릴리스 버전에 대해 설명합니다.

패키지 다운로드 링크	드라이버 버전	세부 정보	릴리스 날짜
3.0.0	3.0.0	Windows 10용 드라이버를 현대화하고 기본 드라이버로 설치하기 위한 지원을 추가했습니다.	2024년 6월 13일
이 버전에서는 다운로드할 수 없습니다.	2.0.0	메탈 인스턴스 유형의 MMIO 직렬 포트 출력 지원이 추가되었습니다. 또한 충돌 구분 분석이 개선되고 출력 형식이 업데이트되었습니다.	2018년 8월 23일

패키지 다운로드 링크	드라이버 버전	세부 정보	릴리스 날짜
이 버전에서는 다운로드할 수 없습니다.	1.0.1	Amazon Inspector와의 네임스페이스 충돌로 인해 드라이버 이름을 `EC2WinUtil` 로 변경했습니다. 몇 가지 버그 수정이 포함되어 있습니다.	2018년 3월 1일
이 버전에서는 다운로드할 수 없습니다.	1.0.0	최초 릴리스. 처음에 드라이버의 이름은 `AwsAgent`였습니다.	2017년 11월 28일

EC2 Windows 인스턴스를 새 버전의 Windows Server로 업그레이드

EC2 Windows 인스턴스의 Windows Server 운영 체제를 이전 버전에서 업그레이드해야 하는 경우 다음 방법 중 하나를 사용할 수 있습니다.

현재 위치 업그레이드

인플레이스 업그레이드는 기존 인스턴스에서 작동합니다. 이 프로세스 중에는 운영 체제 파일만 영향을 받고, 설정, 서버 역할 및 데이터는 그대로 유지됩니다.

마이그레이션(병렬 업그레이드라고도 함)

마이그레이션은 설정, 구성 및 데이터를 캡처하고 이를 새로운 EC2 Windows 인스턴스의 최신 운영 체제로 포팅합니다. 구독한 퍼블릭 또는 프라이빗 Windows AMI, AWS Marketplace 또는 공유된 AMI에서 인스턴스를 시작할 수 있습니다. 또한 EC2 Image Builder를 사용하여 사용자 지정 AMI를 생성할 수 있습니다. 자세한 내용은 Image Builder 사용 설명서를 참조하세요.

> **Note**
>
> AWS에서는 EC2 인스턴스에서 실행되는 Windows Server 버전에 대해 공개적으로 사용 가능한 Amazon Machine Image(AMI) 세트를 제공합니다. 이러한 AMI는 월 단위로 업데이트됩니다. 최신 Windows AMI에 대한 정보는 AWS Windows AMI 참조를 확인하세요.

Microsoft는 전통적으로 Windows Server를 인플레이스 업그레이드하는 대신 새 버전으로 마이그레이션하는 것을 권장해 왔습니다. 마이그레이션은 업그레이드 오류 또는 문제가 줄어들 수 있지만, 새 인스턴스를 프로비저닝하고 애플리케이션을 계획 및 포팅하고 새 인스턴스에서 구성 설정을 조정해야 하므로 현재 위치 업그레이드보다 시간 더 걸릴 수 있습니다. 현재 위치 업그레이드가 더 빠를 수 있지만 소프트웨어 호환성 문제 때문에 오류가 발생할 수도 있습니다.

내용

- EC2 Windows 인스턴스에서 인플레이스 업그레이드 수행
- Automation 런북을 사용하여 EC2 Windows 인스턴스 업그레이드
- EC2 Windows 인스턴스를 Nitro 기반 인스턴스 유형으로 마이그레이션
- EC2 Windows 인스턴스의 운영 체제 업그레이드 문제 해결

EC2 Windows 인스턴스에서 인플레이스 업그레이드 수행

현재 위치 업그레이드를 수행하기 전에 인스턴스에서 어느 네트워크 드라이버가 실행되고 있는지 확인해야 합니다. PV 네트워크 드라이버는 사용자가 원격 데스크톱을 사용하여 인스턴스에 액세스할 수 있게 해줍니다. 인스턴스는 AWS PV, intel Network Adapter 또는 Enhanced Networking 드라이버를 사용합니다. 자세한 내용은 Windows 인스턴스의 반가상화 드라이버 섹션을 참조하세요.

인플레이스(In-Place) 업그레이드를 시작하기 전에

다음 작업을 완료하고 인 플레이스 업그레이드를 시작하기 전에 다음과 같은 중요 세부 정보를 기록합니다.

- 업그레이드 요구 사항, 알려진 문제점 및 제약 조건을 파악할 수 있도록 Microsoft 설명서를 읽습니다. 공식적인 업그레이드 지침도 검토해야 합니다.
  - Windows Server 2012용 업그레이드 옵션
  - Windows Server 2012 R2용 업그레이드 옵션
  - Windows Server 2016 이상의 업그레이드 및 전환 옵션
  - Windows Server 업그레이드
- 2개 이상의 vCPU와 4GB 이상의 RAM을 이용하는 인스턴스에서 운영 체제 업그레이드를 수행하는 것이 좋습니다. 필요하다면 인스턴스를 같은 유형의 더 큰 인스턴스로 변경하고(예: t2.small에서 t2.large로), 업그레이드를 수행한 다음 원래 크기로 다시 변경할 수도 있습니다. 인스턴스 크기를 유지해야 한다면, 인스턴스 콘솔 스크린샷을 사용하여 진행 상황을 모니터링할 수 있습니다. 자세한 내용은 Amazon EC2 인스턴스 유형 변경 섹션을 참조하세요.

- Windows 인스턴스의 루트 볼륨에 사용 가능한 디스크 공간이 충분히 있는지 확인합니다. Windows 설치 프로세스에서 디스크 공간 부족에 대한 경고 메시지를 표시하지 않을 수도 있습니다. 특정 운영 체제를 업그레이드하는 데 필요한 디스크 공간에 대한 정보는 Microsoft 설명서를 참조하세요. 볼륨에 공간이 부족한 경우 확장할 수 있습니다. 자세한 내용은 Amazon EBS 사용 설명서의 Amazon EBS Elastic Volumes를 참조하세요.
- 업그레이드 경로를 확인합니다. 운영 체제를 동일한 아키텍처로 업그레이드해야 합니다. 예를 들어, 32비트 시스템을 32비트 시스템으로 업그레이드해야 합니다. Windows Server 2008 R2 이상은 64비트 전용입니다.
- 안티바이러스와 안티스파이웨어 소프트웨어 및 방화벽을 비활성화합니다. 이러한 유형의 소프트웨어는 업그레이드 프로세스와 충돌할 수 있습니다. 업그레이드를 마친 후에는 안티바이러스와 안티스파이웨어 소프트웨어 및 방화벽을 다시 활성화합니다.
- EC2 Windows 인스턴스를 Nitro 기반 인스턴스 유형으로 마이그레이션에 나온 방법에 따라 최신 드라이버로 업데이트합니다.
- 업그레이드 헬퍼 서비스는 Citrix PV 드라이버를 실행하는 인스턴스만 지원합니다. 인스턴스가 Red Hat 드라이버를 실행하는 경우에는 먼저 수동으로 이러한 드라이버를 업그레이드해야 합니다.

AWS PV, 인텔 Network Adapter 또는 향상된 네트워킹 드라이버를 사용하여 인스턴스 인 플레이스 업그레이드

다음 절차를 통해 AWS PV, intel Network Adapter 또는 Enhanced Networking 네트워크 드라이버를 사용하여 Windows Server 인스턴스를 업그레이드합니다.

인 플레이스 업그레이드를 수행하려면

1. 백업 또는 테스트를 위해 업그레이드할 시스템의 AMI를 생성합니다. 그런 다음 사본에서 업그레이드를 수행하여 테스트 환경을 시뮬레이션합니다. 업그레이드가 완료되면 거의 가동 중단 없이 트래픽을 이 인스턴스로 전환할 수 있습니다. 업그레이드에 실패할 경우에는 백업으로 되돌릴 수 있습니다. 자세한 내용은 Amazon EBS 지원 AMI 생성 섹션을 참조하세요.
2. Windows Server 인스턴스가 최신 네트워크 드라이버를 사용하고 있는지 확인합니다.
    a. AWS PV 드라이버를 업데이트하려면 EC2 Windows 인스턴스의 PV 드라이버 업그레이드 섹션을 참조하세요.
    b. ENA 드라이버를 업데이트하려면 EC2 Windows 인스턴스에 ENA 드라이버 설치 섹션을 참조하세요.
    c. Intel 드라이버를 업데이트하려면 Intel 82599 VF 인터페이스를 통한 향상된 네트워킹 섹션을 참조하세요.

3. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
4. 탐색 창에서 인스턴스를 선택합니다. 인스턴스를 찾습니다. 인스턴스 ID 및 인스턴스의 Availability Zone을 기록해 둡니다. 이 정보는 이 절차의 뒷부분에서 필요합니다.
5. Windows Server 2012 또는 2012 R2에서 Windows Server 2016 이상으로 업그레이드할 경우 계속하기 전에 인스턴스에서 다음을 수행하세요.

   a. EC2Config 서비스를 제거합니다. 자세한 내용은 EC2Launch v2 및 EC2Config 에이전트에 대한 Windows 서비스 관리 섹션을 참조하세요.

   b. EC2Launch v1 또는 EC2Launch v2 에이전트를 설치합니다. 자세한 내용은 EC2Launch v1 에이전트를 사용하여 EC2 Windows 인스턴스 시작 중 태스크 수행와 EC2Launch v2 에이전트를 사용하여 EC2 Windows 인스턴스 시작 중 태스크 수행 섹션을 참조하세요.

   c. AWS Systems Manager SSM Agent 설치 자세한 내용은 AWS Systems Manager 사용 설명서의 Windows Server용 Amazon EC2에서 수동으로 SSM Agent 설치를 참조하세요.
6. Windows Server 설치 미디어 스냅샷에서 새 볼륨을 생성합니다.

   a. 왼쪽 탐색 창의 Elastic Block Store에서 스냅샷을 선택합니다.

   b. 필터 표시줄에서 퍼블릭 스냅샷을 선택합니다.

   c. 검색 표시줄에서 다음 필터를 지정합니다.

      - 소유자 별칭, =, amazon을 차례로 선택합니다.
      - 설명을 선택하고 **`Windows`**를 입력하기 시작합니다. 업그레이드하려는 시스템 아키텍처 및 언어 기본 설정과 일치하는 Windows 필터를 선택합니다. 예를 들어 Windows Server 2019로 업그레이드하려면 Windows 2019 English Installation Media를 선택합니다.

   d. 업그레이드하려는 시스템 아키텍처 및 언어 기본 설정과 일치하는 스냅샷 옆의 확인란을 선택하고 작업, 스냅샷에서 볼륨 생성을 선택합니다.

   e. 볼륨 생성 페이지에서 Windows 인스턴스와 일치하는 가용 영역을 선택하고 볼륨 생성을 선택합니다.
7. 페이지 상단의 볼륨을 새성함 vol-***`1234567890example`*** 배너에서 방금 생성한 볼륨 ID를 선택합니다.
8. 작업(Actions), 볼륨 연결(Attach volume)을 선택합니다.
9. 볼륨 연결 페이지의 인스턴스에서 Windows 인스턴스의 인스턴스 ID를 선택하고 볼륨 연결을 선택합니다.
10. Amazon EBS 볼륨을 사용할 수 있도록 하기의 단계에 따라 새 볼륨을 사용할 수 있도록 합니다.

> **Important**
>
> 디스크를 초기화하면 기존 데이터가 삭제되므로 디스크를 초기화하지 마세요.

11. Windows PowerShell에서 새 볼륨 드라이브로 전환합니다. 인스턴스에 연결한 설치 미디어 볼륨을 열어 업그레이드를 시작합니다.

    a. Windows Server 2016 이상으로 업그레이드한다면 다음을 실행합니다.

    ```
    .\setup.exe /auto upgrade /dynamicupdate disable
    ```

    > **Note**
    >
    > `/dynamicupdate` 옵션을 비활성으로 설정하고 setup.exe를 실행하면 Windows Server 업그레이드 프로세스 중에 Windows에서 업데이트를 설치하지 못하게 됩니다. 업그레이드 중에 업데이트를 설치하면 오류가 발생할 수 있기 때문입니다. 업그레이드가 완료된 후 Windows 업데이트를 사용하여 업데이트를 설치할 수 있습니다.

    Windows Server의 이전 버전으로 업그레이드하는 경우 다음을 실행합니다.

    ```
    Sources\setup.exe
    ```

    b. 설치하고자 하는 운영 체제 선택에서 Windows Server 인스턴스의 전체 설치 옵션을 선택하고 다음을 선택합니다.

    c. 어떤 설치 유형으로 하시겠습니까?(Which type of installation do you want?)에서 업그레이드를 선택합니다.

    d. 마법사를 완료합니다.

Windows Server 설치 프로그램이 파일을 복사하고 처리합니다. 몇 분 후 원격 데스크톱 세션이 닫힙니다. 업그레이드하는 데 걸리는 시간은 Windows Server 인스턴스에서 실행하는 애플리케이션 및 서버 역할의 수에 따라 달라집니다. 업그레이드 프로세스는 최소한 40분 또는 몇 시간이 걸릴 수도 있습니다. 업그레이드 프로세스 중에는 인스턴스가 두 상태 확인 중 하나에 실패합니다. 업그레이드가 완료되면 두 상태 확인이 모두 통과됩니다. 시스템 로그에서 콘솔 출력을 확인하거나 디스크 또는 CPU 활동에 대한 Amazon CloudWatch 측정치를 사용하여 업그레이드가 진행되지 않는지 확인할 수 있습니다.

> **Note**
>
> Windows Server 2019로 업그레이드할 경우, 업그레이드가 완료된 후 필요하면 바탕 화면 배경을 직접 변경하여 이전 운영 체제 이름을 제거할 수 있습니다.

몇 시간 후에도 인스턴스가 두 상태 확인을 모두 통과하지 못한 경우에는 EC2 Windows 인스턴스의 운영 체제 업그레이드 문제 해결 섹션을 참조하세요.

업그레이드 이후 작업

1. 인스턴스에 로그인하여 .NET Framework의 업그레이드를 시작하고 메시지가 나타나면 시스템을 재부팅합니다.
2. 이전 단계에서 EC2Launch v1 또는 EC2Launch v2 에이전트를 아직 설치하지 않은 경우 설치합니다. 자세한 내용은 EC2Launch v1 에이전트를 사용하여 EC2 Windows 인스턴스 시작 중 태스크 수행 및 EC2Launch v2 에이전트를 사용하여 EC2 Windows 인스턴스 시작 중 태스크 수행를 참조하세요.
3. Windows Server 2012 R2로 업그레이드한 경우에는 PV 드라이버를 AWS PV 드라이버로 업그레이드하는 것이 좋습니다. Nitro 기반 인스턴스로 업그레이드한 경우 NVME 및 ENA 드라이버를 설치하거나 업그레이드하는 것이 좋습니다. 자세한 내용은 AWS NVMe 드라이버 또는 Windows에서 향상된 네트워킹 활성화을 참조하세요.
4. 안티바이러스와 안티스파이웨어 소프트웨어 및 방화벽을 다시 활성화합니다.

Automation 런북을 사용하여 EC2 Windows 인스턴스 업그레이드

AWS Systems Manager Automation 런북을 사용하여 AWS에서 Windows 및 SQL Server 인스턴스의 자동 업그레이드를 수행할 수 있습니다.

내용

- 관련 서비스
- 실행 옵션
- Windows Server 업그레이드
- SQL Server 업그레이드

관련 서비스

다음 AWS 서비스는 자동 업그레이드 프로세스에 사용됩니다.

- AWS Systems Manager. AWS Systems Manager은 강력한 통합 인터페이스로서 AWS 리소스를 중앙 집중식으로 관리합니다. 자세한 내용은 AWS Systems Manager 사용 설명서를 참조하세요.
- AWS Systems Manager 에이전트(SSM Agent)는 Amazon EC2 인스턴스, 온프레미스 서버 또는 가상 머신(VM)에 설치 및 구성할 수 있는 Amazon 소프트웨어입니다. SSM Agent를 사용하면 Systems Manager가 이러한 리소스를 업데이트, 관리 및 구성할 수 있습니다. 에이전트는 AWS 클라우드에서 Systems Manager 서비스의 요청을 처리하고 요청에 지정된 대로 실행합니다. 자세한 내용은 AWS Systems Manager 사용 설명서에서 SSM Agent 작업을 참조하세요.
- AWS Systems Manager SSM 런북. SSM 런북은 Systems Manager가 관리형 인스턴스에서 실행하는 작업을 정의합니다. SSM 런북은 JSON(JavaScript Object Notation) 또는 YAML을 사용하며 사용자가 지정하는 단계와 파라미터를 포함합니다. 이 주제에서는 자동화를 위해 2개의 Systems Manager SSM 런북을 사용합니다. 자세한 내용은 AWS Systems Manager 사용 설명서의 AWS Systems Manager Automation 실행서 참조를 참조하세요.

실행 옵션

Systems Manager 콘솔에서 자동화를 선택할 때 실행을 선택합니다. 자동화 문서를 선택한 후 자동화 실행 옵션을 선택하라는 메시지가 표시됩니다. 다음 옵션 중 하나를 선택합니다. 이 주제에서 소개하는 경로에서는 각 단계에 간편 실행 옵션을 사용합니다.

간편 실행

단일 인스턴스를 업데이트하지만 각각의 자동화 단계를 거쳐 결과를 감사하지는 않으려면 이 옵션을 선택합니다. 이 옵션은 이어지는 업그레이드 단계에서 자세히 설명됩니다.

속도 제어

둘 이상의 인스턴스에 업그레이드를 적용하려면 이 옵션을 선택합니다. 다음 설정을 정의합니다.

- 파라미터

  이 설정은 다중 계정 및 리전 설정에도 적용되어 있는데, 자동화의 브랜칭 아웃 방법을 정의합니다.
- 대상

  자동화를 적용하려는 대상을 선택합니다. 또한 이 설정은 다중 계정 및 리전 설정에도 적용됩니다.

- 파라미터 값

  자동화 문서 파라미터에 정의된 값을 사용합니다.

- 리소스 그룹

  AWS에서 리소스란 작업할 수 있는 엔터티입니다. 그 예로는 Amazon EC2 인스턴스, AWS CloudFormation 스택, 또는 Amazon S3 버킷이 포함됩니다. 다수의 리소스를 사용할 경우, 작업을 할 때마다 AWS 서비스 사이를 이동하는 것보다는 해당 리소스를 그룹으로 관리하는 것이 더 유용할 수 있습니다. 몇몇 경우에는 하나의 애플리케이션 계층을 구성하는 EC2 인스턴스 등의 관련 리소스를 다수로 관리해야 할 수 있습니다. 이런 경우에는 해당 리소스에 대한 일괄 작업을 한 번에 수행해야 할 수 있습니다.

- 태그

  태그를 사용하면 AWS 리소스를 용도별, 소유자별 또는 환경별 등 다양한 방식으로 분류할 수 있습니다. 이처럼 분류를 해 놓으면 동일한 유형의 리소스가 많을 때 유용합니다. 할당된 태그를 사용하여 특정 리소스를 신속하게 식별할 수 있습니다.

- 속도 제어

  또한 속도 제어는 다중 계정 및 리전 설정에도 적용됩니다. 속도 제어 파라미터를 설정하면 대상 개수 또는 백분율로 자동화를 적용하려는 플릿의 수를 정의합니다.

다중 계정 및 리전

속도 제어에서 지정되고 다중 계정 및 리전 설정에도 사용되는 파라미터에 더해 추가로 살펴볼 설정이 두 가지 더 있습니다.

- 계정 및 OU(조직 단위)

  자동화를 실행할 여러 계정을 지정합니다.

- AWS 리전

  자동화를 실행할 여러 AWS 리전을 지정합니다.

수동 실행

이 옵션은 간편 실행과 비슷하지만 각 자동화 단계를 거치고 결과를 감사할 수 있다는 데서 차이를 보입니다.

Windows Server 업그레이드

AWSEC2-CloneInstanceAndUpgradeWindows 런북은 계정의 Windows Server 인스턴스에서 Amazon Machine Image(AMI)를 생성하고 이 AMI를 선택한 지원되는 버전으로 업그레이드합니다. 이 다단계 프로세스는 완료까지 최대 2시간이 소요됩니다.

자동 업그레이드 프로세스에는 두 개의 AMI가 포함됩니다.

- 현재 실행 중인 인스턴스: 첫 번째 AMI는 현재 실행 중인 인스턴스로서 업그레이드되지 않았습니다. 이 AMI는 다른 인스턴스를 시작하여 현재 위치 업그레이드를 실행하는 데 쓰입니다. 프로세스가 완료되면 이 AMI는 원본 인스턴스를 유지하도록 특별히 요청하지 않는 한 계정에서 삭제됩니다. 이 설정은 파라미터 KeepPreUpgradeImageBackUp에서 처리합니다(기본값은 false이며, 즉 AMI가 기본적으로 삭제됨).
- 업그레이드된 AMI: 이 AMI는 자동화 프로세스의 결과물입니다.

최종 결과물은 하나의 AMI로서, 업그레이드된 AMI 인스턴스입니다.

업그레이드가 완료되면 Amazon VPC에서 새 AMI를 시작하여 애플리케이션 기능을 테스트할 수 있습니다. 테스트 이후와 다른 업그레이드를 수행하기 전에는 업그레이드된 인스턴스로 완전히 전환하기에 앞서 애플리케이션 중단 시간을 예약하세요.

사전 조건

AWS Systems Manager Automation 문서로 Windows Server 업그레이드를 자동화하려면 다음 태스크를 수행해야 합니다.

- 지정한 IAM 정책으로 IAM 규칙 만들기를 하여 Systems Manager가 Amazon EC2 인스턴스에서 자동화 작업을 수행하고 사용자가 필수적인 Systems Manager 사용 조건에 부합하는지 확인합니다. 자세한 내용은 AWS Identity and Access Management 사용 설명서의 AWS 서비스에 대한 권한을 위임할 역할 생성을 참조하세요.
- 원하는 자동화 실행 방법에 대한 옵션을 선택합니다. 실행 옵션으로는 간편 실행(Simple execution), 속도 제어(Rate control), 다중 계정 및 리전(Multi-account and Region), 수동 실행(Manual execution)이 있습니다. 이러한 옵션에 대한 자세한 내용은 실행 옵션 섹션을 참조하세요.
- 인스턴스에 SSM Agent가 설치되었는지 확인합니다. 자세한 내용은 Windows Server용 Amazon EC2 인스턴스에 SSM Agent 설치 및 구성을 참조하세요.
- 인스턴스에 Windows PowerShell 3.0 이상이 설치되어 있어야 합니다.
- Microsoft Active Directory 도메인에 조인된 인스턴스의 경우, 호스트 이름 충돌을 피하기 위해 도메인 컨트롤러에 연결되지 않은 SubnetId를 지정하는 것이 좋습니다.

- 인스턴스 서브넷에는 Amazon S3와 같은 AWS 서비스에 대한 액세스와 Microsoft에서 패치를 다운로드할 수 있는 액세스를 제공하는 인터넷에 대한 아웃바운드 연결이 있어야 합니다. 서브넷이 퍼블릭 서브넷이고 인스턴스에 퍼블릭 IP 주소가 있거나, 서브넷이 퍼블릭 NAT 디바이스에 인터넷 트래픽을 전송하는 경로가 있는 프라이빗 서브넷인 경우에 이 요구 사항이 충족됩니다.
- 이 자동화는 Windows Server 2008 R2, Windows Server 2012 R2, Windows Server 2016 및 Windows Server 2019를 실행하는 인스턴스에서 작동합니다.
- 인스턴스의 부트 디스크에 20GB의 사용 가능한 디스크 공간이 있는지 확인합니다.
- 인스턴스에서 AWS 제공 Windows 라이선스를 사용하지 않는 경우 Windows Server 2012 R2 설치 미디어를 포함하는 Amazon EBS 스냅샷 ID를 지정합니다. 방법:
  1. Amazon EC2 인스턴스에서 Windows Server 2012 이상을 실행 중인지 확인합니다.
  2. 인스턴스가 실행 중인 동일 가용 영역에서 6GB Amazon EBS 볼륨을 생성합니다. 볼륨을 인스턴스에 연결합니다. 예를 들면 드라이브 D에 탑재합니다.
  3. ISO를 마우스 오른쪽 버튼으로 클릭하고 인스턴스(예를 들면 드라이브 E)에 탑재합니다.
  4. 드라이브 E:\에서 드라이브 D:\로 ISO의 내용을 복사합니다.
  5. 위의 2단계에서 생성한 6GB 볼륨의 Amazon EBS 스냅샷을 생성합니다.

Windows Server 업그레이드 제한 사항

이 자동화는 Windows 도메인 컨트롤러, 클러스터 또는 Windows 데스크톱 운영 체제 업그레이드를 지원하지 않습니다. 또한 이 자동화는 다음 역할이 설치된 Windows Server용 Amazon EC2 인스턴스를 지원하지 않습니다.

- 원격 데스크톱 세션 호스트(RDSH)
- 원격 데스크톱 연결 브로커(RDCB)
- 원격 데스크톱 가상화 호스트(RDVH)
- 원격 데스크톱 웹 액세스(RDWA)

Windows Server의 자동 업그레이드를 수행하는 단계

다음 단계에 따라 AWSEC2-CloneInstanceAndUpgradeWindows 자동화 런북을 사용하여 Windows Server 인스턴스를 업그레이드합니다.

1. AWS 관리 콘솔에서 Systems Manager를 엽니다.
2. 왼쪽 탐색 창의 변경 관리(Change Management) 아래에서 자동화(Automation)를 선택합니다.
3. 자동화 실행(Execute automation)을 선택합니다.

4. `AWSEC2-CloneInstanceAndUpgradeWindows`라는 이름의 자동화 문서를 찾습니다.
5. 문서 이름이 보이면 선택합니다. 해당 이름을 선택하면 문서의 세부 정보가 표시됩니다.
6. 자동화 실행(Execute automation)을 선택하여 이 문서의 파라미터를 입력합니다. 페이지 상단에서 간편 실행(Simple execution)이 선택된 채로 둡니다.
7. 요청된 파라미터를 다음 지침을 따라 입력합니다.

   - `InstanceID`

     Type: 문자열

     (필수) SSM Agent가 설치된 상태에서 Windows Server 2008 R2, 2012 R2, 2016 또는 2019를 실행하는 인스턴스입니다.

   - `InstanceProfile`.

     유형: 문자열

     (필수) IAM 인스턴스 프로파일입니다. Amazon EC2 인스턴스 및 AWS AMI에 대해 Systems Manager 자동화를 수행하는 데 사용되는 IAM 역할입니다. 자세한 내용은 AWS Systems Manager 사용 설명서의 EC2 인스턴스 권한 구성을 참조하세요.

   - `TargetWindowsVersion`

     Type: 문자열

     (필수) 대상 Windows 버전을 선택합니다.

   - `SubnetId`

     Type: 문자열

     (필수) 업그레이드 프로세스를 위한 서브넷으로, 소스 EC2 인스턴스가 상주합니다. 서브넷에서 (패치를 다운로드할) AWS 서비스(Amazon S3 포함) 및 Microsoft로의 아웃바운드 연결이 설정되었는지 확인합니다.

   - `KeepPreUpgradedBackUp`

     Type: 문자열

     (선택 사항) 이 파라미터가 `true`로 설정된 경우, 인스턴스에서 생성된 이미지는 자동화 과정에서 그대로 유지됩니다. 기본 설정은 `false`입니다.

   - `RebootInstanceBeforeTakingImage`

Type: 문자열

(선택 사항) 기본값은 `false`입니다(재부팅 안 함). 이 파라미터가 `true`로 설정된 경우, Systems Manager에서는 업그레이드를 위해 AMI를 만들기 전에 인스턴스를 재부팅합니다.

8. 파라미터를 입력한 다음 Execute(실행)를 선택합니다. 자동화가 시작되면 실행 진척도를 모니터링할 수 있습니다.
9. 자동화가 완료되면 AMI ID를 볼 수 있습니다. AMI를 시작하여 Windows OS의 업그레이드 여부를 확인할 수 있습니다.

> **Note**
>
> 모든 단계를 자동화로 실행할 필요는 없습니다. 해당 단계는 자동화와 인스턴스의 동작에 따라 달라질 수 있습니다. Systems Manager에서는 필요하지 않은 단계를 생략할 수도 있습니다.
> 또한 몇몇 단계는 시간이 초과될 수 있습니다. Systems Manager는 모든 최신 패치의 업그레이드와 설치를 시도합니다. 그러나 때에 따라서는 특정 단계의 정의 가능 제한 시간 설정에 따라 패치의 시간이 초과됩니다. 이 경우에는 내부 OS가 대상 Windows Server 버전으로 업그레이드될 수 있도록 Systems Manager 자동화가 다음 단계까지 계속됩니다.

10. 자동화가 완료되고 나면 AMI ID로 Amazon EC2 인스턴스를 실행하여 업그레이드를 검토할 수 있습니다. AWS AMI에서 Amazon EC2 인스턴스를 만드는 방법에 대한 자세한 내용은 사용자 지정 AMI에서 EC2 인스턴스를 시작하려면 어떻게 해야 합니까?를 참조하세요.

SQL Server 업그레이드

AWSEC2-CloneInstanceAndUpgradeSQLServer 스크립트는 계정 내에서 SQL Server를 실행하는 Amazon EC2 인스턴스에서 AMI를 생성한 다음 해당 AMI를 이후 버전의 SQL Server로 업그레이드합니다. 이 다단계 프로세스는 완료까지 최대 2시간이 소요됩니다.

이 워크플로우에서는 자동화를 통해 인스턴스에서 AMI를 생성한 다음 사용자가 제공하는 서브넷에서 새 AMI를 실행합니다. 그런 다음 자동화는 SQL Server의 현재 위치 업그레이드를 수행합니다. 이 자동화는 업그레이드 완료 후 업그레이드된 인스턴스를 종료하기 전에 새 AMI를 생성합니다.

자동 업그레이드 프로세스에는 두 개의 AMI가 포함됩니다.

- 현재 실행 중인 인스턴스: 첫 번째 AMI는 현재 실행 중인 인스턴스로서 업그레이드되지 않았습니다. 이 AMI는 다른 인스턴스를 시작하여 현재 위치 업그레이드를 실행하는 데 쓰입니다. 프로세스가 완

료되면 이 AMI는 원본 인스턴스를 유지하도록 특별히 요청하지 않는 한 계정에서 삭제됩니다. 이 설정은 파라미터 KeepPreUpgradeImageBackUp에서 처리합니다(기본값은 false이며, 즉 AMI가 기본적으로 삭제됨).

- 업그레이드된 AMI: 이 AMI는 자동화 프로세스의 결과물입니다.

최종 결과물은 하나의 AMI로서, 업그레이드된 AMI 인스턴스입니다.

업그레이드가 완료되면 Amazon VPC에서 새 AMI를 시작하여 애플리케이션 기능을 테스트할 수 있습니다. 테스트 이후와 다른 업그레이드를 수행하기 전에는 업그레이드된 인스턴스로 완전히 전환하기에 앞서 애플리케이션 중단 시간을 예약하세요.

사전 조건

AWS Systems Manager Automation 문서로 SQL Server 업그레이드를 자동화하려면 다음 태스크를 수행해야 합니다.

- 지정한 IAM 정책으로 IAM 규칙 만들기를 하여 Systems Manager가 Amazon EC2 인스턴스에서 자동화 작업을 수행하고 사용자가 필수적인 Systems Manager 사용 조건에 부합하는지 확인합니다. 자세한 내용은 AWS Identity and Access Management 사용 설명서의 AWS 서비스에 대한 권한을 위임할 역할 생성을 참조하세요.
- 원하는 자동화 실행 방법에 대한 옵션을 선택합니다. 실행 옵션으로는 간편 실행(Simple execution), 속도 제어(Rate control), 다중 계정 및 리전(Multi-account and Region), 수동 실행(Manual execution)이 있습니다. 이러한 옵션에 대한 자세한 내용은 실행 옵션 섹션을 참조하세요.
- Amazon EC2 인스턴스는 Windows Server 2008 R2 이상과 SQL Server 2008 이상을 사용해야 합니다.
- 인스턴스에 SSM Agent가 설치되었는지 확인합니다. 자세한 내용은 Windows Server용 Amazon EC2 인스턴스에서 SSM Agent 사용을 참조하세요.
- 인스턴스에 dudb 디스크 공간이 충분한지 확인합니다.
  - Windows Server 2008 R2에서 2012 R2로 업그레이드하거나 Windows Server 2012 R2에서 이후 운영 체제로 업그레이드하는 경우 인스턴스 부팅 디스크에 20GB의 여유 디스크 공간이 있는지 확인합니다.
  - Windows Server 2008 R2에서 2016 이상으로 업그레이드하는 경우 인스턴스 부팅 디스크에 40GB의 여유 디스크 공간이 있는지 확인합니다.
- 기존 보유 라이선스 사용(BYOL) SQL Server 버전을 사용하는 인스턴스의 경우 다음 추가 사전 조건이 적용됩니다.
  - 대상 SQL Server 설치 미디어가 포함된 Amazon EBS 스냅샷 ID를 제공합니다. 방법:

1. Amazon EC2 인스턴스에서 Windows Server 2008 R2 이상을 실행 중인지 확인합니다.
2. 인스턴스가 실행 중인 동일 가용 영역에서 6GB Amazon EBS 볼륨을 생성합니다. 볼륨을 인스턴스에 연결합니다. 예를 들면 드라이브 D에 탑재합니다.
3. ISO를 마우스 오른쪽 버튼으로 클릭하고 인스턴스(예를 들면 드라이브 E)에 탑재합니다.
4. 드라이브 E:\에서 드라이브 D:\로 ISO의 내용을 복사합니다.
5. 2단계에서 생성한 6GB 볼륨의 Amazon EBS 스냅샷을 생성합니다.

SQL Server 자동 업그레이드 제한

AWSEC2-CloneInstanceAndUpgradeSQLServer 런북을 사용하여 자동 업그레이드를 수행할 때 다음 제한 사항이 적용됩니다.

- 이 업그레이드는 Windows 인증을 사용하여 SQL Server에서만 수행할 수 있습니다.
- 인스턴스에서 대기 중인 보안 패치 업데이트가 없는지 확인합니다. 제어판을 열고 나서 업데이트 확인을 선택합니다.
- HA 및 미러링 모드의 SQL 서버 배포는 지원되지 않습니다.

SQL Server의 자동 업그레이드를 수행하는 단계

다음 단계에 따라 AWSEC2-CloneInstanceAndUpgradeSQLServer 자동화 런북을 사용하여 SQL Server 인스턴스를 업그레이드합니다.

1. 아직 확인이 되지 않은 경우에는 SQL Server 2016을 .iso 파일로 다운로드하고 이를 소스 서버에 마운트합니다.
2. .iso 파일 마운팅 후에는 구성 요소 파일을 모두 복사하고 원하는 볼륨에 배치합니다.
3. 볼륨의 Amazon EBS 스냅샷을 만들고 스냅샷의 ID를 이후에 사용할 클립보드에 복사합니다. 자세한 내용은 Amazon EBS 사용 설명서의 Amazon EBS 스냅샷 생성을 참조하세요.
4. 인스턴스 프로파일을 Amazon EC2 소스 인스턴스에 연결합니다. 그러면 Systems Manager가 EC2 인스턴스와 통신하고 해당 인스턴스가 AWS Systems Manager 서비스에 추가된 후에 명령을 실행할 수 있습니다. 이 예에서는 역할을 `SSM-EC2-Profile-Role`로 명명했으며 해당 역할에는 `AmazonSSMManagedInstanceCore` 정책이 연결되었습니다.
5. AWS Systems Manager 콘솔의 왼쪽 탐색 창에서 Managed Instances(관리형 인스턴스)를 선택합니다. EC2 인스턴스가 관리형 인스턴스 목록에 있는지 확인합니다. 몇 분 후에도 인스턴스가 표시되지 않으면 AWS Systems Manager 사용 설명서에서 내 인스턴스는 어디에 있나요?를 참조하세요.

6. 왼쪽 탐색 창의 변경 관리(Change Management) 아래에서 자동화(Automation)를 선택합니다.
7. 자동화 실행(Execute automation)을 선택합니다.
8. `AWSEC2-CloneInstanceAndUpgradeSQLServer`라는 이름의 자동화 문서를 찾습니다.
9. `AWSEC2-CloneInstanceAndUpgradeSQLServer` SSM 문서를 선택하고 다음(Next)을 선택합니다.
10. 간편 실행(Simple execution) 옵션이 선택되었는지 확인합니다.
11. 요청된 파라미터를 다음 지침을 따라 입력합니다.

    - `InstanceId`

      Type: 문자열

      (필수) SQL Server 2008 R2(또는 이후 버전) 실행 인스턴스입니다.

    - `IamInstanceProfile`

      Type: 문자열

      (필수) IAM 인스턴스 프로파일입니다.

    - `SQLServerSnapshotId`

      유형: 문자열

      (필수) 대상 SQL Server 설치 미디어의 스냅샷 ID입니다. 이 파라미터는 SQL Server 라이선스 포함 인스턴스에 필요하지 않습니다.

    - `SubnetId`

      Type: 문자열

      (필수) 업그레이드 프로세스를 위한 서브넷으로, 소스 EC2 인스턴스가 상주합니다. 서브넷에서 (패치를 다운로드할) AWS 서비스(Amazon S3 포함) 및 Microsoft로의 아웃바운드 연결이 설정되었는지 확인합니다.

    - `KeepPreUpgradedBackUp`

      Type: 문자열

      (선택 사항) 이 파라미터가 `true`로 설정된 경우, 인스턴스에서 생성된 이미지는 자동화 과정에서 그대로 유지됩니다. 기본 설정은 `false`입니다.

    - `RebootInstanceBeforeTakingImage`

Type: 문자열

(선택 사항) 기본값은 `false`입니다(재부팅 안 함). 이 파라미터가 `true`로 설정된 경우, Systems Manager에서는 업그레이드를 위해 AMI를 만들기 전에 인스턴스를 재부팅합니다.

- `TargetSQLVersion`

  유형: 문자열

  (선택 사항) 대상 SQL Server 버전입니다. 기본값은 `2016`입니다.

12. 파라미터를 입력한 다음 실행(Execute)을 선택합니다. 자동화가 시작되면 실행 진척도를 모니터링할 수 있습니다.
13. 실행 상태(Execution Status)가 성공(Success)으로 표시되면 출력(Outputs)을 선택하여 AMI 정보를 봅니다. AMI ID를 사용하여 원하는 VPC에 대해 SQL Server 인스턴스를 시작할 수 있습니다.
14. Amazon EC2 콘솔을 엽니다. 왼쪽 탐색 창에서 AMI를 선택합니다. 그러면 새 AMI가 보일 것입니다.
15. 새 SQL Server 버전이 성공적으로 설치되었는지 확인하려면 새 AMI를 선택하고 시작(Launch)을 선택합니다.
16. 배포하려는 AMI, VPC 및 서브넷, 그리고 사용하려는 스토리지에 대해 인스턴스 유형을 선택합니다. AMI에서 새 인스턴스를 시작하므로 해당 볼륨은 사용자가 시작하는 EC2 인스턴스 내에 포함할 옵션으로 표시됩니다. 이 볼륨은 제거할 수 있으며, 반대로 추가할 수도 있습니다.
17. 인스턴스 식별에 도움이 되도록 태그를 추가합니다.
18. 보안 그룹을 인스턴스에 추가합니다.
19. 인스턴스 시작을 선택합니다.
20. 인스턴스의 이름을 선택하고 작업 드롭다운 메뉴 아래에서 연결을 선택합니다.
21. 새 SQL Server 버전이 새 인스턴스의 데이터베이스 엔진인지 확인합니다.

EC2 Windows 인스턴스를 Nitro 기반 인스턴스 유형으로 마이그레이션

AWS Windows AMI는 Microsoft 설치 미디어에서 사용하는 기본 설정 및 일부 사용자 지정으로 구성됩니다. 사용자 지정에는 M5 및 C5와 같은 Nitro 기반 인스턴스를 지원하는 드라이버와 구성이 포함됩니다.

베어 메탈 인스턴스를 포함하여 Xen 기반 인스턴스에서 Nitro 기반 인스턴스로 마이그레이션하는 경우 다음 사례에 해당한다면 이 항목의 단계를 따르는 것이 좋습니다.

- 사용자 지정 Windows AMI에서 인스턴스를 시작하는 경우
- 2018년 8월 이전에 생성된, Amazon에서 제공한 Windows AMI에서 인스턴스를 시작하는 경우

또는 `AWSSupport-UpgradeWindowsAWSDrivers` 자동화 문서를 사용하여 1부, 2부 및 3부에서 설명한 절차를 자동화할 수 있습니다. 자동화된 절차 사용을 선택한 경우 다음((대안) AWS를 사용하여 AWS Systems Manager PV, ENA 및 NVMe 드라이버 업그레이드)을 참조하고 4부와 5부로 계속 진행합니다.

자세한 내용은 Amazon EC2 업데이트 - 추가 인스턴스 유형, Nitro 시스템 및 CPU 옵션을 참조하세요.

> **Note**
>
> 다음 마이그레이션 절차는 Windows Server 버전 2016 이상에서 수행할 수 있습니다. 수명이 종료된 이전 운영 체제 버전은 테스트되지 않으므로 최신 인스턴스 유형과 호환되지 않을 수 있습니다.
> Linux 인스턴스를 마이그레이션하려면 the section called “인스턴스 유형 변경” 섹션을 참조하세요.

목차

- 1부: AWS PV 드라이버 설치 및 업그레이드
- 2부: ENA 설치 및 업그레이드
- 3부: AWS NVMe 드라이버 업그레이드
- 4부: EC2Config 및 EC2Launch 업데이트
- 5부: 베어 메탈 인스턴스를 위한 직렬 포트 설치
- 6부: 전원 관리 설정 업데이트
- 7부: 새 인스턴스 유형에 대한 인텔 칩셋 드라이버 업데이트
- (대안) AWS를 사용하여 AWS Systems Manager PV, ENA 및 NVMe 드라이버 업그레이드

시작하기 전에

이 절차에서는 M4 또는 C4와 같은 Xen 기반 인스턴스가 있고 Nitro 기반 인스턴스로 마이그레이션한다고 가정합니다.

업그레이드를 성공적으로 수행하려면 PowerShell 버전 3.0 이상을 사용해야 합니다.

> **Note**
>
> 마이그레이션하는 경우 해당 인스턴스가 새 향상된 네트워킹 어댑터 디바이스로 기본 설정되어 있으므로 기존 네트워크 인터페이스 카드에 대한 정적 IP 또는 사용자 지정 DNS 네트워크 설정이 손실될 수 있습니다.

이 절차의 단계를 수행하기 전에 인스턴스를 백업해 놓는 것이 좋습니다. EC2 콘솔에서 마이그레이션 필요한 인스턴스를 선택하고, 바로 가기(마우스 오른쪽 버튼 클릭) 메뉴를 열고 인스턴스 상태를 선택한 후 중지를 선택합니다.

> **Warning**
>
> 인스턴스를 중지하면 인스턴스 스토어 볼륨의 데이터가 삭제됩니다. 인스턴스 스토어 볼륨의 데이터를 보존하기 위해 영구 스토리지에 데이터를 백업해야 합니다.

EC2 콘솔에서 인스턴스의 바로 가기(마우스 오른쪽 버튼 클릭) 메뉴를 열고 이미지를 선택한 후 이미지 생성을 선택합니다.

> **Note**
>
> 이 지침의 4부와 5부는 인스턴스 유형을 마이그레이션하거나 변경한 후에 완료할 수 있습니다. 그러나 특히 베어 메탈 인스턴스 유형으로 마이그레이션하는 경우에는 마이그레이션하기 전에 이 작업을 완료하는 것이 좋습니다.

1부: AWS PV 드라이버 설치 및 업그레이드

Nitro 시스템에서 AWS PV 드라이버를 사용하지 않더라도 이전 버전의 Citrix PV 또는 AWS PV를 사용하는 경우 업그레이드해야 합니다. 최신 AWS PV 드라이버는 이전 버전 드라이버에서 Nitro 시스템을 사용하거나 Xen 기반 인스턴스로 다시 마이그레이션해야 할 경우 발생할 수 있는 버그를 해결했습니다. 항상 AWS의 Windows 인스턴스용 최신 드라이버로 업데이트하는 것이 가장 좋습니다.

다음 절차에 따라 AWS PV 드라이버의 현재 위치 업그레이드를 수행하거나, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016 또는 Windows Server 2019의 Citrix PV 드라이버에서 AWS PV 드라이버로 업그레이드할 수 있습니다. 자세한 내용은 EC2 Windows 인스턴스의 PV 드라이버 업그레이드 섹션을 참조하세요.

도메인 컨트롤러를 업그레이드하려면 도메인 컨트롤러 업그레이드(AWS PV 업그레이드) 섹션을 참조하세요.

AWS PV 드라이버의 업그레이드를 수행하려면

1. 원격 데스크톱을 사용하여 인스턴스에 연결하고 인스턴스를 업그레이드할 준비를 합니다. 업그레이드를 수행하기 전에 시스템 디스크가 아닌 모든 디스크를 오프라인으로 전환합니다. AWS PV 드라이버의 현재 위치 업그레이드를 수행할 경우에는 이 단계가 필요하지 않습니다. 또한 서비스 콘솔에서 필수적이지 않은 서비스를 수동 시작으로 설정합니다.
2. 최신 드라이버 패키지를 인스턴스로 다운로드합니다.
3. 폴더의 내용 압축을 풀고 `AWSPVDriverSetup.msi`를 실행합니다.

MSI를 실행하면 인스턴스가 자동으로 재부팅되고 드라이버를 업그레이드합니다. 최대 15분 동안 인스턴스를 사용할 수 없습니다.

업그레이드를 완료하고 인스턴스가 Amazon EC2 콘솔에서 두 상태 확인을 모두 통과하면 원격 데스크톱을 사용하여 인스턴스에 연결하고 새 드라이버가 설치되었는지 확인합니다. 디바이스 관리자(Device Manager)의 스토리지 컨트롤러(Storage Controllers) 아래에서 AWS PV Storage Host Adapter를 찾습니다. 드라이버 버전이 드라이버 버전 기록 표에 나열된 최신 버전과 동일한지 확인합니다. 자세한 내용은 AWS PV 드라이버 패키지 기록 섹션을 참조하세요.

2부: ENA 설치 및 업그레이드

모든 네트워크 기능을 지원할 수 있도록 최신 Elastic Network Adapter 드라이버로 업그레이드합니다. 인스턴스를 시작했는데 향상된 네트워킹 기능이 활성화되어 있지 않은 경우에는 인스턴스에 필요한 네트워크 어댑터 드라이버를 다운로드하고 설치해야 합니다. enaSupport 인스턴스 속성을 설정하여 향상된 네트워킹을 활성화합니다. 이 속성은 지원되는 인스턴스 유형 및 ENA 드라이버가 설치된 경우에만 활성화할 수 있습니다. 자세한 내용은 EC2 인스턴스에서 ENA로 향상된 네트워킹 활성화 섹션을 참조하세요.

1. 최신 드라이버를 인스턴스로 다운로드합니다. 이전 버전의 드라이버가 필요한 경우 ENA Windows 드라이버 버전 내역 섹션을 참조하세요.
2. ZIP 아카이브를 추출합니다.
3. 압축 파일을 푼 폴더에서 `install.ps1` PowerShell 스크립트를 실행하여 드라이버를 설치합니다.

> **Note**
>
> 설치 오류를 피하려면 `install.ps1` 스크립트를 관리자로 실행하세요.

4. AMI에 enaSupport가 활성화되었는지 확인합니다. 그렇지 않은 경우 EC2 인스턴스에서 ENA로 향상된 네트워킹 활성화의 설명서를 따라 계속하세요.

3부: AWS NVMe 드라이버 업그레이드

AWS NVMe 드라이버는 성능 개선을 위해 Nitro 시스템에 NVMe 블록 디바이스로 표시되는 Amazon EBS 및 SSD 인스턴스 스토어 볼륨과 상호 작용하는 데 사용됩니다.

> **Important**
>
> 다음 지침은 인스턴스를 Nitro 기반 인스턴스로 마이그레이션하려는 목적으로 Xen 기반 인스턴스에 AWS NVMe를 설치하거나 업그레이드하는 경우에 맞도록 특별히 수정되었습니다.

1. 최신 드라이버 패키지를 인스턴스로 다운로드합니다.

   이전 버전의 드라이버가 필요한 경우 지원되는 버전에 대한 NVMe Windows 드라이버 릴리스 섹션을 참조하세요.
2. ZIP 아카이브를 추출합니다.
3. `Readme.txt`의 설명에 따라 드라이버를 설치합니다.
4. PowerShell 세션을 열고 다음 명령을 실행합니다.

```
PS C:\> start rundll32.exe sppnp.dll,Sysprep_Generalize_Pnp -wait
```

> **Note**
>
> 명령을 적용하려면 PowerShell 세션을 관리자로 실행해야 합니다. PowerShell(x86) 버전에서는 오류가 발생합니다.
> 이 명령은 디바이스 드라이버에서만 sysprep을 실행합니다. 전체 sysprep 준비를 실행하지는 않습니다.

5. Windows Server 2008 R2 및 Windows Server 2012의 경우 인스턴스를 종료하고 인스턴스 유형을 변경한 후 인스턴스를 시작한 다음 4부로 계속 진행합니다. Nitro 기반 인스턴스 유형으로 마이그레이션하기 전에 Xen 기반 인스턴스 유형에서 인스턴스를 다시 시작하면 인스턴스가 부팅되지 않습니다. 지원되는 다른 Windows AMI를 위해, 디바이스에서 sysprep을 수행한 후 언제든지 인스턴스 유형을 변경할 수 있습니다.

4부: EC2Config 및 EC2Launch 업데이트

Windows 인스턴스의 경우, 최신 EC2Config와 EC2Launch 유틸리티는 EC2 Bare Metal을 비롯한 Nitro 시스템에서 실행할 때 추가 기능과 정보를 제공합니다. 기본적으로 EC2Config 서비스는 Windows Server 2016 이전의 AMI에 포함되어 있습니다. EC2Launch는 Windows Server 2016 이상 AMI의 EC2Config를 대체합니다.

EC2Config 서비스와 EC2Launch 서비스가 업데이트되면 새 AWS Windows AMI에 최신 버전의 서비스가 포함됩니다. 그러나 자체 Windows AMI 및 인스턴스는 별도로 최신 버전의 EC2Config 및 EC2Launch로 업데이트해야 합니다.

EC2Config를 설치 또는 업데이트하려면

1. EC2Config 설치 관리자를 다운로드하고 압축을 풉니다.
2. `EC2Install.exe`를 실행합니다. 전체 옵션 목록을 보려면 `EC2Install` 옵션을 포함해 `/?` 파일을 실행합니다. 기본적으로 설치는 프롬프트를 표시합니다. 프롬프트 없이 명령을 실행하려면 `/quiet` 옵션을 사용합니다.

자세한 내용은 최신 버전의 EC2Config 설치 섹션을 참조하세요.

EC2Launch를 설치 또는 업데이트하려면

1. 이미 인스턴스에 EC2Launch를 설치하여 구성한 경우 EC2Launch 구성 파일의 백업을 만듭니다. 설치 프로세스는 이 파일에 변경 사항을 보존하지 않습니다. 기본적으로 `C:\ProgramData\Amazon\EC2-Windows\Launch\Config` 디렉터리에 파일이 위치합니다.
2. 인스턴스의 디렉터리로 EC2-Windows-Launch.zip 파일을 다운로드합니다.
3. `EC2-Windows-Launch.zip` 파일을 다운로드한 동일한 디렉터리에 install.ps1을 다운로드합니다.
4. `install.ps1`를 실행합니다.

> **Note**
>
> 설치 오류를 피하려면 `install.ps1` 스크립트를 관리자로 실행하세요.

5. EC2Launch 구성 파일의 백업을 만든 경우 `C:\ProgramData\Amazon\EC2-Windows\Launch\Config` 디렉터리에 복사합니다.

자세한 내용은 EC2Launch v1 에이전트를 사용하여 EC2 Windows 인스턴스 시작 중 태스크 수행 섹션을 참조하세요.

5부: 베어 메탈 인스턴스를 위한 직렬 포트 설치

`i3.metal` 인스턴스 유형은 I/O 포트 기반 직렬 디바이스가 아닌 PCI 기반 직렬 디바이스를 사용합니다. 최신 Windows AMI는 PCI 기반 직렬 디바이스를 자동으로 사용하며 직렬 포트 드라이버가 설치되어 있습니다. Amazon에서 제공한 2018.04.11일자 또는 그 이전 Windows AMI에서 시작한 인스턴스를 사용하지 않는 경우, 직렬 포트 드라이버를 설치하여 직렬 디바이스에서 EC2 기능(암호 생성 및 콘솔 출력 등)을 사용할 수 있도록 해야 합니다. 또한 최신 EC2Config와 EC2Launch 유틸리티는 i3.metal을 지원하고 추가 기능을 제공합니다. 아직 수행하지 않은 경우 4부의 단계를 수행할 것을 권장합니다.

직렬 시리얼 포트 드라이버를 설치하려면

1. 직렬 드라이버 패키지를 인스턴스로 다운로드합니다.
2. 폴더 내용의 압축을 풀고 `aws_ser.INF`의 컨텍스트 메뉴를 열고(마우스 오른쪽 버튼으로 클릭) [설치(install)]를 선택합니다.
3. 확인을 선택합니다.

6부: 전원 관리 설정 업데이트

전원 관리 설정에 대한 다음 업데이트는 디스플레이가 꺼지지 않도록 설정해 Nitro 시스템에서 OS를 정상적으로 종료할 수 있도록 합니다. Amazon에서 2018년 11월 28일 이후 제공한 모든 Windows AMI에는 이러한 기본 구성이 이미 적용되어 있습니다.

1. 명령 프롬프트 또는 PowerShell 세션을 엽니다.
2. 다음 명령을 실행합니다:

```
powercfg /setacvalueindex 381b4222-f694-41f0-9685-ff5bb260df2e 7516b95f-
f776-4464-8c53-06167f40cc99 3c0bc021-c8a8-4e07-a973-6b14cbcb2b7e 0
```

```
powercfg /setacvalueindex 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c 7516b95f-
f776-4464-8c53-06167f40cc99 3c0bc021-c8a8-4e07-a973-6b14cbcb2b7e 0
powercfg /setacvalueindex a1841308-3541-4fab-bc81-f71556f20b4a 7516b95f-
f776-4464-8c53-06167f40cc99 3c0bc021-c8a8-4e07-a973-6b14cbcb2b7e 0
```

7부: 새 인스턴스 유형에 대한 인텔 칩셋 드라이버 업데이트

`u-6tb1.metal`, `u-9tb1.metal` 및 `u-12tb1.metal` 인스턴스 유형은 이전에 Windows AMI에 설치되지 않은 칩셋 드라이버가 필요한 하드웨어를 사용합니다. Amazon에서 제공한 2018.11.19일자 또는 그 이전 Windows AMI에서 시작한 인스턴스를 사용하지 않는 경우, 인텔 Chipset INF 유틸리티를 사용하여 드라이버를 설치해야 합니다.

칩셋 드라이버를 설치하려면

1. 인스턴스에 대한 Chipset INF 유틸리티입니다.
2. 파일의 압축을 풉니다.
3. `SetupChipset.exe`를 실행합니다.
4. 인텔 소프트웨어 라이선스 계약에 동의하고 칩셋 드라이버를 설치합니다.
5. 인스턴스를 재부팅합니다.

(대안) AWS를 사용하여 AWS Systems Manager PV, ENA 및 NVMe 드라이버 업그레이드

`AWSSupport-UpgradeWindowsAWSDrivers` 자동화 문서는 1부, 2부 및 3부에서 설명한 단계를 자동화합니다. 이 방법은 또한 드라이버 업그레이드 실패 시 인스턴스를 복구할 수 있습니다.

`AWSSupport-UpgradeWindowsAWSDrivers` 자동화 문서는 지정된 EC2 인스턴스의 스토리지 및 네트워크 AWS 드라이버를 업그레이드 또는 복구합니다. 문서는 AWS 에이전트(SSM Agent)를 호출하여 온라인으로 AWS Systems Manager 드라이버의 최신 버전 설치를 시도합니다. SSM Agent에 접촉할 수 없다면 문서는 명시적 요청된 경우 AWS 드라이버의 오프라인 설치를 수행할 수 있습니다.

> **Note**
>
> 이 절차는 도메인 컨트롤러에는 수행하지 못합니다. 도메인 컨트롤러의 드라이버를 업데이트하려면 도메인 컨트롤러 업그레이드(AWS PV 업그레이드)을 참조하세요.

AWS을 사용하여 AWS Systems Manager PV, ENA 및 NVMe 드라이버를 자동으로 업그레이드하려면

1. https://console.aws.amazon.com/systems-manager에서 Systems Manager 콘솔을 엽니다.
2. 자동화를 선택한 다음, 자동화 실행(Execute automation)을 선택합니다.
3. AWSSupport-UpgradeWindowsAWSDrivers 자동화 문서를 검색하여 선택한 후 자동화 실행을 선택합니다.
4. 입력 파라미터 섹션에서 다음 옵션을 구성합니다.

   인스턴스 ID

   업그레이드할 인스턴스의 고유 ID를 입력합니다.

   AllowOffline

   (선택 사항) 다음 옵션 중 하나를 선택합니다.

   - `True` - 이 옵션을 선택하여 오프라인 설치를 수행합니다. 업그레이드 프로세스 도중에는 인스턴스가 중지 및 재시작됩니다.

     > **Warning**
     > 인스턴스를 중지하면 인스턴스 스토어 볼륨의 데이터가 삭제됩니다. 인스턴스 스토어 볼륨의 데이터를 보존하기 위해 영구 스토리지에 데이터를 백업해야 합니다.

   - `False` - (기본) 온라인 설치를 수행하려면 이 옵션을 선택한 채로 둡니다. 업그레이드 프로세스 도중에는 인스턴스가 재시작됩니다.

     > **Important**
     > 온라인 및 오프라인 업그레이드에서 업그레이드 작업 전에 AMI가 생성됩니다. AMI는 자동화 완료 이후에도 지속됩니다. AMI에 대한 액세스에 보안 조치를 취하거나 더 이상 필요하지 않은 경우 삭제합니다.

   SubnetId

   (선택 사항) 다음 값 중 하나를 입력합니다.

   - `SelectedInstanceSubnet` — (기본) 업그레이드 프로세스에서 도우미 인스턴스를 업그레이드해야 하는 인스턴스로 동일한 서브넷에서 시작합니다. 서브넷은 Systems Manager 엔드포인트(`ssm.*`)와의 통신을 허용해야 합니다.

- `CreateNewVPC` — 업그레이드 프로세스가 새 VPC에 도우미 인스턴스를 시작합니다. 대상 인스턴스의 서브넷이 `ssm.*` 엔드포인트와의 통신을 허용하는지 불확실한 경우 이 옵션을 사용합니다. 사용자에게 VPC 생성 권한이 있어야 합니다.
- 특정 서브넷 ID — 도우미 인스턴스를 시작할 특정 서브넷의 ID를 지정합니다. 인스턴스를 업그레이드해야 하므로 서브넷은 동일한 가용 영역에 있어야 하고, `ssm.*` 엔드포인트와의 통신을 허용해야 합니다.

5. 실행을 선택합니다.
6. 업그레이드 완료를 허용합니다. 온라인 업그레이드를 완료하려면 최대 10분이 걸리고, 오프라인 업그레이드를 완료하려면 최대 25분이 걸릴 수 있습니다.

EC2 Windows 인스턴스의 운영 체제 업그레이드 문제 해결

AWS는 Citrix PV 드라이버가 관련된 현재 위치 업그레이드를 수행하는 사용자를 돕는 AWS 유틸리티인 업그레이드 헬퍼 서비스를 통해 업그레이드 문제에 대한 지원을 제공합니다.

업그레이드 후 인스턴스에서 .NET 런타임 최적화 서비스가 .NET 프레임워크를 최적화하는 동안 일시적으로 CPU 사용률이 평균보다 높게 나타날 수 있습니다. 이는 예상된 동작입니다.

몇 시간 후에도 인스턴스가 두 상태 확인을 모두 통과하지 못한 경우에는 다음을 점검하세요.

- Windows Server 2008로 업그레이드한 후 몇 시간이 지나도 두 상태 확인에 실패하는 경우 업그레이드되지 않고 확인 클릭(Click OK)을 수행하여 롤백을 확인하라는 메시지가 표시될 수 있습니다. 이 상태에서는 콘솔에 액세스할 수 없기 때문에 버튼을 클릭할 수 있는 방법이 없습니다. 이 문제를 해결하려면 Amazon EC2 콘솔 또는 API를 통해 재부팅을 수행하세요. 재부팅은 시작하는 데 10분 이상 걸리고, 25분이 지나면 인스턴스를 사용할 수 있습니다.
- 서버에서 애플리케이션 또는 서버 역할을 제거하고 다시 시도합니다.

서버에서 애플리케이션 또는 서버 역할을 제거한 후에도 인스턴스가 두 상태 확인을 통과하지 못하는 경우 다음을 수행하세요.

- 인스턴스를 중지하고 루트 볼륨을 다른 인스턴스에 연결합니다. 자세한 내용은 "Waiting for the metadata service"에서 루트 볼륨을 중지하고 다른 인스턴스에 연결하는 방법에 대한 설명을 참조하세요.
- 장애에 대한 Windows 설치 로그 파일 및 이벤트 로그를 분석합니다.

운영 체제 업그레이드 또는 마이그레이션과 관련된 기타 모든 문제에 대해서는 인플레이스(In-Place) 업그레이드를 시작하기 전에에 수록된 도움말을 확인하는 것이 좋습니다.

자습서: Amazon RDS 데이터베이스에 Amazon EC2 인스턴스 연결

자습서 목표

이 자습서의 목표는 AWS Management Console을 사용하여 Amazon EC2 인스턴스와 Amazon RDS 데이터베이스 간의 보안 연결을 구성하는 방법을 알아보는 데 있습니다.

연결을 구성하기 위한 다양한 옵션이 있습니다. 이 자습서에서는 다음과 같은 세 가지 옵션을 살펴봅니다.

- 옵션 1: EC2 콘솔을 사용하여 자동으로 RDS 데이터베이스에 인스턴스 연결

  EC2 콘솔에서 자동 연결 기능을 사용하여 EC2 인스턴스와 RDS 데이터베이스 간의 트래픽을 허용하도록 EC2 인스턴스와 RDS 데이터베이스 간의 연결을 자동으로 구성합니다.
- 옵션 2: RDS 콘솔을 사용하여 자동으로 RDS 데이터베이스에 인스턴스 연결

  RDS 콘솔에서 자동 연결 기능을 사용하여 EC2 인스턴스와 RDS 데이터베이스 간의 트래픽을 허용하도록 EC2 인스턴스와 RDS 데이터베이스 간의 연결을 자동으로 구성합니다.
- 옵션 3: 보안 그룹을 생성하여 수동으로 RDS 데이터베이스에 인스턴스 연결

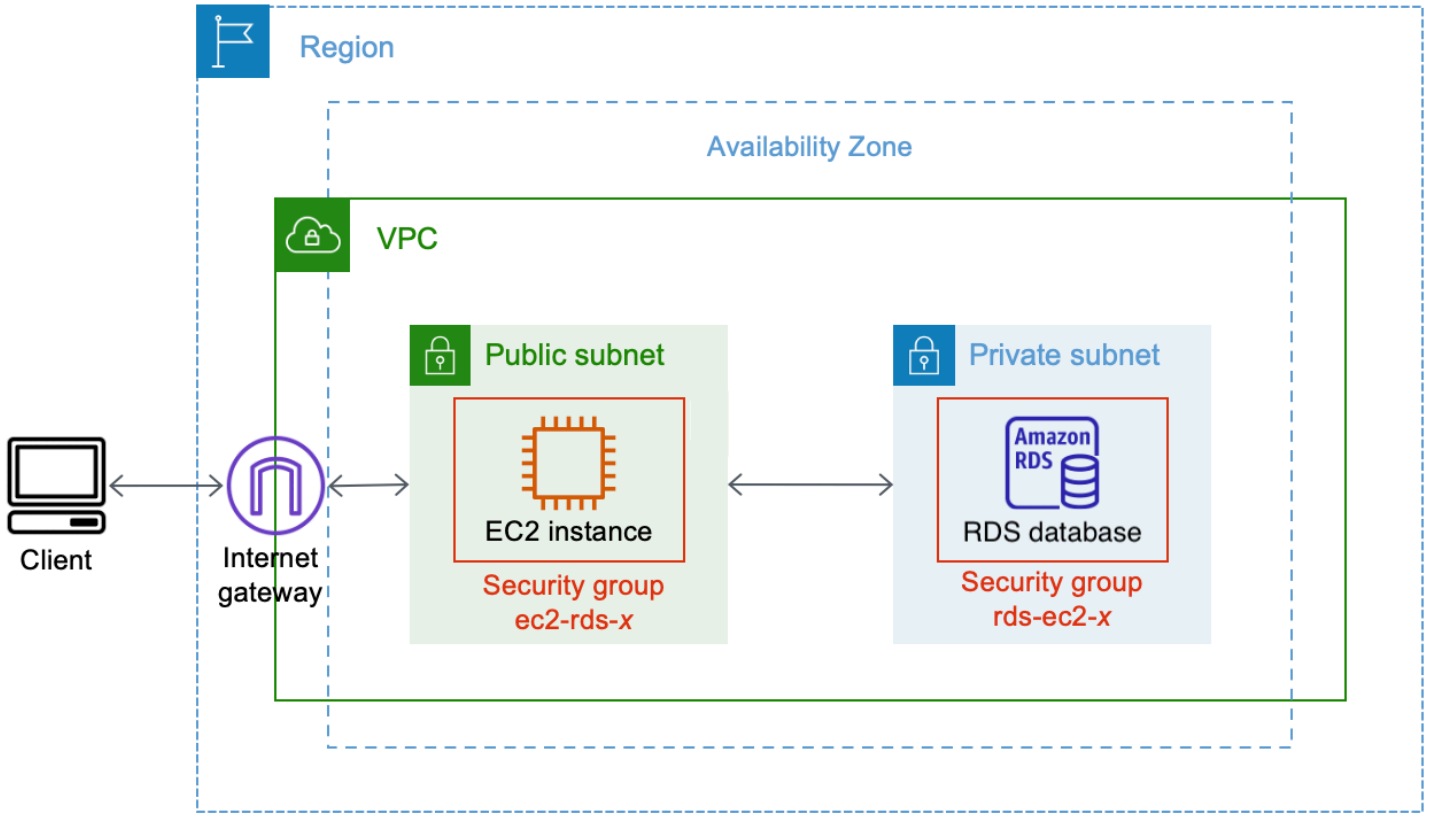
  옵션 1 및 옵션 2의 자동 연결 기능에 의해 자동으로 생성된 구성을 재현하도록 보안 그룹을 수동으로 구성하고 할당하여 EC2 인스턴스와 RDS 데이터베이스 간의 연결을 구성합니다.

컨텍스트

EC2 인스턴스와 RDS 데이터베이스 간의 연결을 구성하려는 이유에 대한 컨텍스트로, 다음 시나리오를 살펴보겠습니다. 즉, 웹 사이트에서 사용자가 작성할 양식을 제시합니다. 이 경우 데이터베이스에 양식 데이터를 캡처해야 합니다. 웹 서버로 구성된 EC2 인스턴스에 웹 사이트를 호스팅할 수 있으며, RDS 데이터베이스에 양식 데이터를 캡처할 수 있습니다. 양식 데이터가 EC2 인스턴스에서 RDS 데이터베이스로 이동할 수 있도록 EC2 인스턴스와 RDS 데이터베이스는 서로 연결되어야 합니다. 이 자습서에서는 해당 연결을 구성하는 방법을 설명합니다. 이는 EC2 인스턴스와 RDS 데이터베이스를 연결하기 위한 사용 사례의 한 예일 뿐입니다.

아키텍처

다음 다이어그램은 생성된 리소스와 이 자습서의 모든 단계를 완료한 결과의 아키텍처 구성을 보여 줍니다.



다이어그램에서는 다음과 같이 생성할 리소스를 보여 줍니다.

- 동일한 AWS 리전, VPC 및 가용 영역에 EC2 인스턴스와 RDS 데이터베이스를 생성합니다.
- 퍼블릭 서브넷에 EC2 인스턴스를 생성합니다.
- 프라이빗 서브넷에 RDS 데이터베이스를 생성합니다.

  RDS 콘솔을 사용하여 RDS 데이터베이스를 생성하고 EC2 인스턴스를 자동으로 연결하면 데이터베이스에 대한 VPC, DB 서브넷 그룹 및 퍼블릭 액세스 설정이 자동으로 선택됩니다. RDS 데이터베이스는 EC2 인스턴스와 동일한 VPC 내의 프라이빗 서브넷에 자동으로 생성됩니다.
- 인터넷 사용자는 인터넷 게이트웨이를 통해 SSH 또는 HTTP/HTTPS를 사용하여 EC2 인스턴스에 연결할 수 있습니다.
- 인터넷 사용자는 RDS 데이터베이스에 직접 연결할 수 없습니다. EC2 인스턴스만 RDS 데이터베이스에 연결됩니다.

- 자동 연결 기능을 사용하여 EC2 인스턴스와 RDS 데이터베이스 간의 트래픽을 허용할 때 다음 보안 그룹이 자동으로 생성되고 추가됩니다.
  - 보안 그룹 ec2-rds-*x*가 생성되어 EC2 인스턴스에 추가됩니다. 이 보안 그룹에는 rds-ec2-*x* 보안 그룹을 대상으로 참조하는 아웃바운드 규칙이 하나 있습니다. 이를 통해 EC2 인스턴스의 트래픽은 rds-ec2-*x* 보안 그룹이 있는 RDS 데이터베이스에 도달할 수 있습니다.
  - 보안 그룹 rds-ec2-*x*가 생성되어 RDS 데이터베이스에 추가됩니다. 이 보안 그룹에는 ec2-rds-*x* 보안 그룹을 소스로 참조하는 인바운드 규칙이 하나 있습니다. 이를 통해 ec2-rds-*x* 보안 그룹이 있는 EC2 인스턴스의 트래픽이 RDS 데이터베이스에 도달할 수 있습니다.

  별도의 보안 그룹(EC2 인스턴스용 보안 그룹과 RDS 데이터베이스용 보안 그룹)을 사용하면 인스턴스와 데이터베이스의 보안을 더 적절히 제어할 수 있습니다. 인스턴스와 데이터베이스 모두에서 동일한 보안 그룹을 사용하고 데이터베이스에만 적합하도록 보안 그룹을 수정한 경우 이러한 수정은 인스턴스와 데이터베이스 모두에 영향을 미칩니다. 다시 말해서 하나의 보안 그룹을 사용하는 경우 보안 그룹이 리소스(인스턴스 또는 데이터베이스)에 연결되어 있다는 사실을 잊어버려서 의도치 않게 리소스의 보안을 수정할 수 있습니다.

  또한 자동으로 생성되는 보안 그룹은 워크로드별 보안 그룹 쌍을 생성하여 데이터베이스 포트에서 이 워크로드에 대한 상호 연결만 허용하므로 최소 권한을 준수합니다.

고려 사항

이 자습서에서 작업을 완료할 때 다음을 고려하세요.

- Two consoles(두 개의 콘솔) - 이 자습서에서는 다음과 같은 두 개의 콘솔을 사용합니다.
  - Amazon EC2 콘솔 - EC2 콘솔을 사용하여 인스턴스를 시작하고, EC2 인스턴스를 RDS 데이터베이스에 자동으로 연결하며, 수동 옵션의 경우 보안 그룹을 생성해 연결을 구성합니다.
  - Amazon RDS 콘솔 - RDS 콘솔을 사용하여 RDS 데이터베이스를 생성하고, EC2 인스턴스를 RDS 데이터베이스에 자동으로 연결합니다.
- One VPC(하나의 VPC) - 자동 연결 기능을 사용하려면 EC2 인스턴스와 RDS 데이터베이스가 동일한 VPC에 있어야 합니다.

  EC2 인스턴스와 RDS 데이터베이스 간의 연결을 수동으로 구성하는 경우 한 VPC에서 EC2 인스턴스를 시작하고 다른 VPC에서 RDS 데이터베이스를 시작할 수 있습니다. 그러나 라우팅 및 VPC 구성을 추가로 설정해야 합니다. 이 자습서에서는 이 시나리오를 다루지 않습니다.
- 하나의 AWS 리전 - EC2 인스턴스와 RDS 데이터베이스가 동일한 리전에 있어야 합니다.

- Two security groups(두 개의 보안 그룹) - EC2 인스턴스와 RDS 데이터베이스 간의 연결은 EC2 인스턴스용 보안 그룹과 RDS 데이터베이스용 보안 그룹이라는 두 개의 보안 그룹에 의해 구성됩니다.

  EC2 콘솔 또는 RDS 콘솔에서 자동 연결 기능을 사용하여 연결을 구성하면(이 자습서의 옵션 1 및 옵션 2) 보안 그룹이 자동으로 생성되어 EC2 인스턴스 및 RDS 데이터베이스에 할당됩니다.

  자동 연결 기능을 사용하지 않는 경우 보안 그룹을 수동으로 생성하고 할당해야 합니다. 이 자습서의 옵션 3에서 이 작업을 수행합니다.

자습서를 완료하는 데 걸리는 시간

30 분

자습서를 전체적으로 한 번에 완료하거나 한번에 한 작업씩 완료할 수 있습니다.

비용

이 튜토리얼을 완료하면 생성하는 AWS 리소스에 대해 비용이 발생할 수 있습니다.

AWS 계정을 생성한 지 12개월 미만이고 프리 티어 요구 사항에 따라 리소스를 구성하는 경우 프리 티어로 Amazon EC2를 사용할 수 있습니다.

EC2 인스턴스와 RDS 데이터베이스가 서로 다른 가용 영역에 있는 경우 데이터 전송 요금이 발생합니다. 이러한 요금이 발생되지 않도록 하려면 EC2 인스턴스와 RDS 데이터베이스가 동일한 가용 영역에 있어야 합니다. 데이터 전송 요금에 관한 자세한 내용은 Amazon EC2 온디맨드 요금 페이지에서 데이터 전송을 참조하세요.

자습서를 완료한 후 비용이 발생되지 않도록 하려면 더 이상 필요하지 않은 리소스를 삭제해야 합니다. 리소스를 삭제하는 구체적인 단계는 작업 4(선택 사항): 정리 단원을 참조하세요.

옵션 1: EC2 콘솔을 사용하여 자동으로 RDS 데이터베이스에 인스턴스 연결

옵션 1의 목표는 EC2 인스턴스에서 RDS 데이터베이스로의 트래픽을 허용하도록 EC2 인스턴스와 RDS 데이터베이스 간의 연결을 자동으로 구성하는 EC2 콘솔의 자동 연결 기능을 살펴보는 데 있습니다. 연결을 수동으로 구성하는 방법은 옵션 3에서 알아봅니다.

업무

- 시작하기 전 준비 사항
- 작업 1(선택 사항): RDS 데이터베이스 생성
- 작업 2(선택 사항): EC2 인스턴스 시작

- 작업 3: EC2 인스턴스를 RDS 데이터베이스에 자동으로 연결
- 작업 4: 연결 구성 확인
- 작업 5(선택 사항): 정리

시작하기 전 준비 사항

이 자습서를 완료하려면 다음이 필요합니다.

- EC2 인스턴스와 동일한 VPC에 있는 RDS 데이터베이스 - 기존 RDS 데이터베이스를 사용하거나 작업 1의 단계에 따라 새 RDS 데이터베이스를 생성할 수 있습니다.
- RDS 데이터베이스와 동일한 VPC에 있는 EC2 인스턴스입니다. 기존 EC2 인스턴스를 사용하거나 작업 2의 단계에 따라 새 EC2 인스턴스를 생성할 수 있습니다.
- 다음 작업을 호출할 수 있는 권한:
  - `ec2:AssociateRouteTable`
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:CreateRouteTable`
  - `ec2:CreateSecurityGroup`
  - `ec2:CreateSubnet`
  - `ec2:DescribeInstances`
  - `ec2:DescribeNetworkInterfaces`
  - `ec2:DescribeRouteTables`
  - `ec2:DescribeSecurityGroups`
  - `ec2:DescribeSubnets`
  - `ec2:ModifyNetworkInterfaceAttribute`
  - `ec2:RevokeSecurityGroupEgress`

작업 1(선택 사항): RDS 데이터베이스 생성

> **Note**
>
> Amazon RDS 데이터베이스 생성은 이 자습서에서 중점적으로 다루는 사항이 아닙니다. RDS 데이터베이스가 이미 있고 이 자습서에서 해당 데이터베이스를 사용하려는 경우 이 작업을 건너뛸 수 있습니다.

기존 RDS 데이터베이스를 사용하는 경우 자동 연결 기능을 사용할 수 있도록 해당 데이터베이스가 EC2 인스턴스와 동일한 VPC에 있는지 확인하세요.

이 작업의 목표는 EC2 인스턴스와 RDS 데이터베이스 간의 연결을 구성하는 작업 3을 완료할 수 있도록 RDS 데이터베이스를 생성하는 데 있습니다. 이 작업의 단계에서는 다음과 같이 RDS 데이터베이스를 구성합니다.

- 엔진 유형: MySQL
- 템플릿: 프리 티어
- DB 인스턴스 식별자: **`tutorial-database-1`**
- DB 인스턴스 클래스: `db.t3.micro`

Important

프로덕션 환경에서는 특정 요구를 충족하도록 데이터베이스를 구성해야 합니다.

MySQL RDS 데이터베이스 생성

1. https://console.aws.amazon.com/rds/에서 Amazon RDS 콘솔을 엽니다.
2. 리전 선택기(오른쪽 상단)에서 AWS 리전을 선택합니다. EC2 콘솔에서 자동 연결 기능을 사용하기 위해서는 데이터베이스와 EC2 인스턴스가 동일한 리전에 있어야 합니다.
3. 대시보드에서 Create database(데이터베이스 생성)를 선택합니다.
4. Choose a database creation method(데이터베이스 생성 방법 선택)에서 Standard create(표준 생성)가 선택되어 있는지 확인합니다. Easy create(손쉬운 생성)를 선택한 경우 VPC 선택기를 사용할 수 없습니다. EC2 콘솔에서 자동 연결 기능을 사용하려면 데이터베이스가 EC2 인스턴스와 동일한 VPC에 있어야 합니다.
5. Engine options(엔진 옵션)에서 Engine type(엔진 유형)으로 MySQL을 선택합니다.
6. Templates(템플릿)에서 요구에 맞는 샘플 템플릿을 선택합니다. 이 자습서에서는 Free tier(프리 티어)를 선택하여 무료로 RDS 데이터베이스를 생성합니다. 그러나 프리 티어는 계정을 생성한 지 12개월 미만인 경우에만 사용할 수 있습니다. 그리고 다른 제한 사항이 적용됩니다. Free tier(프리 티어) 상자에서 Info(정보) 링크를 선택하면 자세한 내용을 확인할 수 있습니다.
7. 설정에서 다음을 수행합니다.

a. DB instance identifier(DB 인스턴스 식별자)에 데이터베이스 이름을 입력합니다. 이 자습서에서는 **tutorial-database-1**을 입력합니다.

b. Master username(마스터 사용자 이름)의 경우 기본 이름인 **admin**을 그대로 둡니다.

c. Master password(마스터 암호)에 이 자습서에서 기억할 수 있는 암호를 입력한 다음, Confirm password(암호 확인)에 암호를 다시 입력합니다.

8. 인스턴스 구성의 DB 인스턴스 클래스에서 기본값인 db.t3.micro를 그대로 둡니다. 계정이 12개월이 아직 안 된 경우 이 데이터베이스 클래스를 무료로 사용할 수 있습니다. 그리고 다른 제한 사항이 적용됩니다. 자세한 내용은 AWS 프리 티어 섹션을 참조하세요.

9. Connectivity(연결)에서 Compute resource(컴퓨팅 리소스)에 대해 Don't connect to an EC2 compute resource(EC2 컴퓨팅 리소스에 연결하지 않음)를 선택합니다. 나중에 작업 3에서 EC2 인스턴스와 RDS 데이터베이스를 연결하기 때문입니다.

   (나중에 이 자습서의 옵션 2에서 Connect to an EC2 compute resource(EC2 컴퓨팅 리소스에 연결)를 선택하여 RDS 콘솔의 자동 연결 기능을 사용해 봅니다.)

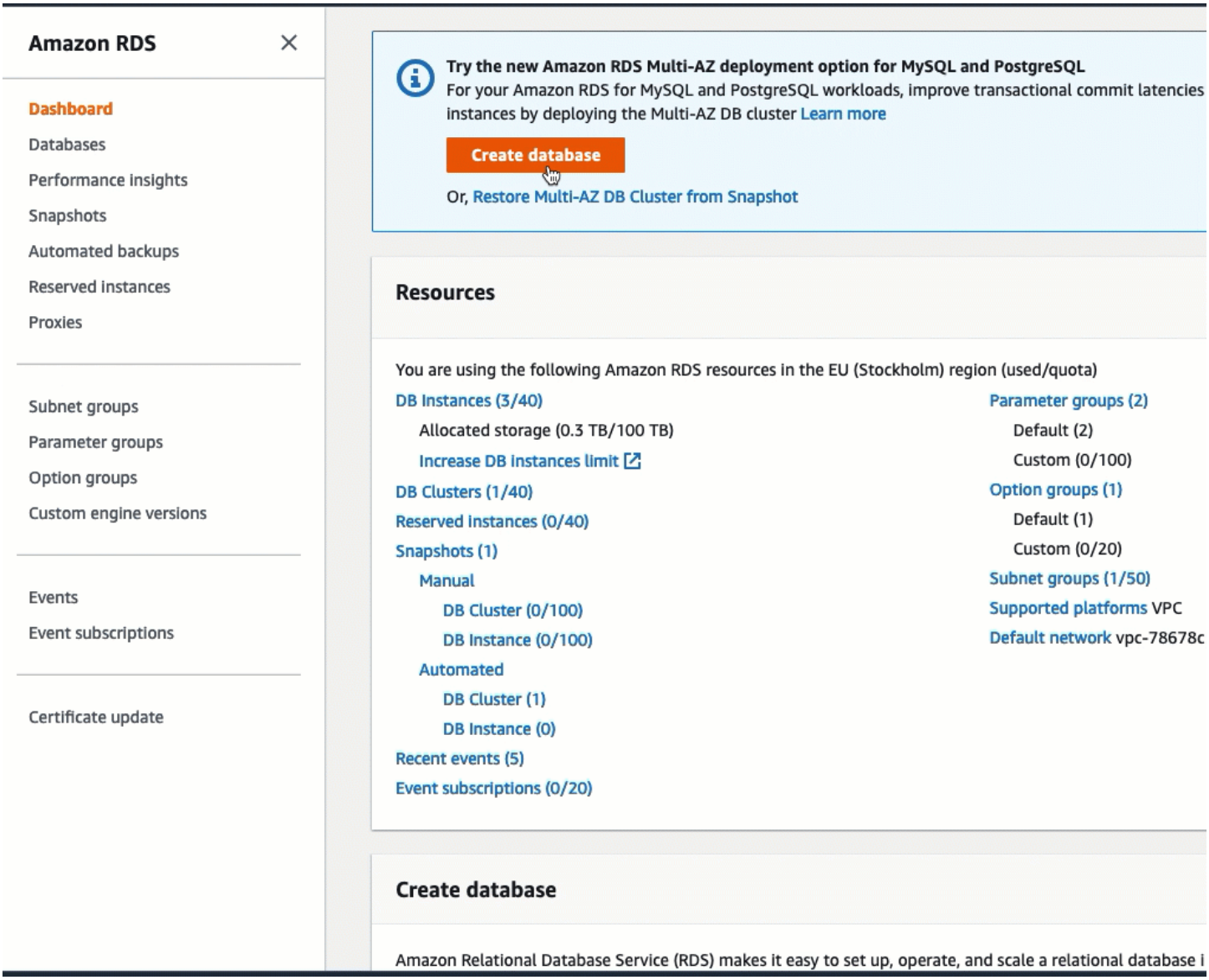
10. Virtual private cloud (VPC)에서 VPC를 선택합니다. VPC에는 DB 서브넷 그룹이 있어야 합니다. 자동 연결 기능을 사용하려면 EC2 인스턴스와 RDS 데이터베이스가 동일한 VPC에 있어야 합니다.

11. 이 페이지의 다른 필드에 대한 기본값을 모두 그대로 유지합니다.

12. 데이터베이스 생성를 선택합니다.

    Databases(데이터베이스) 화면에서 데이터베이스를 사용할 준비가 될 때까지 새 데이터베이스의 Status(상태)는 Creating(생성 중)입니다. 상태가 Available(사용 가능)로 변경되면 데이터베이스에 연결할 수 있습니다. 데이터베이스 클래스 및 스토리지 양에 따라 새 데이터베이스를 사용할 수 있게 되기까지 최대 20분이 걸릴 수 있습니다.

애니메이션 보기: RDS 데이터베이스 생성



작업 2(선택 사항): EC2 인스턴스 시작

> **Note**
>
> 인스턴스 시작은 이 자습서에서 중점적으로 다루는 사항이 아닙니다. Amazon EC2 인스턴스가 이미 있고 이 자습서에서 해당 인스턴스를 사용하려는 경우 이 작업을 건너뛸 수 있습니다. 기존 EC2 인스턴스를 사용하는 경우 자동 연결 기능을 사용할 수 있도록 해당 인스턴스가 RDS 데이터베이스와 동일한 VPC에 있는지 확인하세요.

이 작업의 목표는 EC2 인스턴스와 Amazon RDS 데이터베이스 간의 연결을 구성하는 작업 3을 완료할 수 있도록 EC2 인스턴스를 시작하는 데 있습니다. 이 작업의 단계에서는 다음과 같이 EC2 인스턴스를 구성합니다.

- 인스턴스 이름: **`tutorial-instance-1`**
- AMI: Amazon Linux 2
- 인스턴스 유형: `t2.micro`
- 퍼블릭 IP 자동 할당: 활성화됨
- 다음과 같은 세 가지 규칙이 있는 보안 그룹:
  - IP 주소에서 SSH 허용
  - 어디에서 들어오든 HTTPS 트래픽 허용
  - 어디에서 들어오든 HTTP 트래픽 허용

> **Important**
>
> 프로덕션 환경에서는 특정 요구를 충족하도록 인스턴스를 구성해야 합니다.

EC2 인스턴스 시작

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 리전 선택기(오른쪽 상단)에서 AWS 리전을 선택합니다. EC2 콘솔에서 자동 연결 기능을 사용하기 위해서는 인스턴스와 RDS 데이터베이스가 동일한 리전에 있어야 합니다.
3. EC2 Dashboard(EC2 대시보드)에서 Launch instance(인스턴스 시작)를 선택하세요.
4. Name and tags(이름 및 태그)에서 Name(이름)에 인스턴스를 식별하는 이름을 입력합니다. 이 자습서에서는 인스턴스 이름을 **`tutorial-instance-1`**로 지정합니다. 인스턴스 이름을 반드시 입력해야 하는 것은 아니지만, EC2 콘솔에서 인스턴스를 선택할 때 해당 이름을 통해 인스턴스를 쉽게 식별할 수 있습니다.
5. Application and OS Images(애플리케이션 및 OS 이미지)에서 웹 서버 요구에 맞는 AMI를 선택합니다. 이 자습서에서는 Amazon Linux 2를 사용합니다.
6. Instance type(인스턴스 유형)에서 Instance type(인스턴스 유형)으로 웹 서버 요구에 맞는 인스턴스 유형을 선택합니다. 이 자습서에서는 `t2.micro`를 사용합니다.

> **Note**
>
> AWS 계정을 생성한 지 12개월 미만이고 `t2.micro` 인스턴스 유형 또는 `t2.micro`를 사용할 수 없는 리전에서는 `t3.micro`를 선택한 경우 프리 티어로 Amazon EC2를 사용할 수 있습니다. t3.micro 인스턴스를 시작하면 기본적으로 무제한 모드로 설정되므로 CPU 사용량에 따라 추가 요금이 부과될 수 있다는 점에 유의하세요.

7. Key pair (login)(키 페어(로그인))에서 Key pair name(키 페어 이름)에 대해 키 페어를 선택합니다.
8. Network settings(네트워크 설정)에서 다음을 수행합니다.

   a. 기본 VPC 또는 서브넷을 변경하지 않은 경우 Network(네트워크) 및 Subnet(서브넷)의 기본 설정을 유지할 수 있습니다.

      기본 VPC 또는 서브넷을 변경한 경우 다음을 확인합니다.

      i. 자동 연결 기능을 사용하려면 인스턴스가 RDS 데이터베이스와 동일한 VPC에 있어야 합니다. 기본적으로 VPC는 하나만 있습니다.

      ii. 인스턴스를 시작하는 VPC에는 인터넷 게이트웨이가 연결되어 있어야 합니다. 그래야 인터넷에서 웹 서버에 액세스할 수 있습니다. 기본 VPC는 인터넷 게이트웨이를 통해 자동으로 설정됩니다.

      iii. 인스턴스가 퍼블릭 IP 주소를 수신하도록 하려면 Auto-assign public IP(퍼블릭 IP 자동 할당)에 대해 Enable(활성화)이 선택되어 있는지 확인합니다. Disable(비활성화)이 선택된 경우 Edit(편집)(Network Settings(네트워크 설정) 오른쪽에 있음)를 선택한 다음, Auto-assign public IP(퍼블릭 IP 자동 할당)에서 Enable(활성화)을 선택합니다.

   b. SSH를 사용하여 인스턴스에 연결하려면 컴퓨터의 퍼블릭 IPv4 주소로부터의 SSH(Linux) 또는 RDP(Windows) 트래픽을 승인하는 보안 그룹 규칙이 필요합니다. 기본적으로 인스턴스를 시작하면 어디에서 들어오든 인바운드 SSH 트래픽을 허용하는 규칙을 사용하여 새 보안 그룹이 생성됩니다.

      자체 IP 주소만 인스턴스에 연결할 수 있도록 하려면 방화벽(보안 그룹) 아래의 다음으로부터의 SSH 트래픽 허용 확인란 옆에 있는 드롭다운 목록에서 내 IP를 선택합니다.

   c. 인터넷에서 인스턴스로의 트래픽을 허용하려면 다음 확인란을 선택합니다.

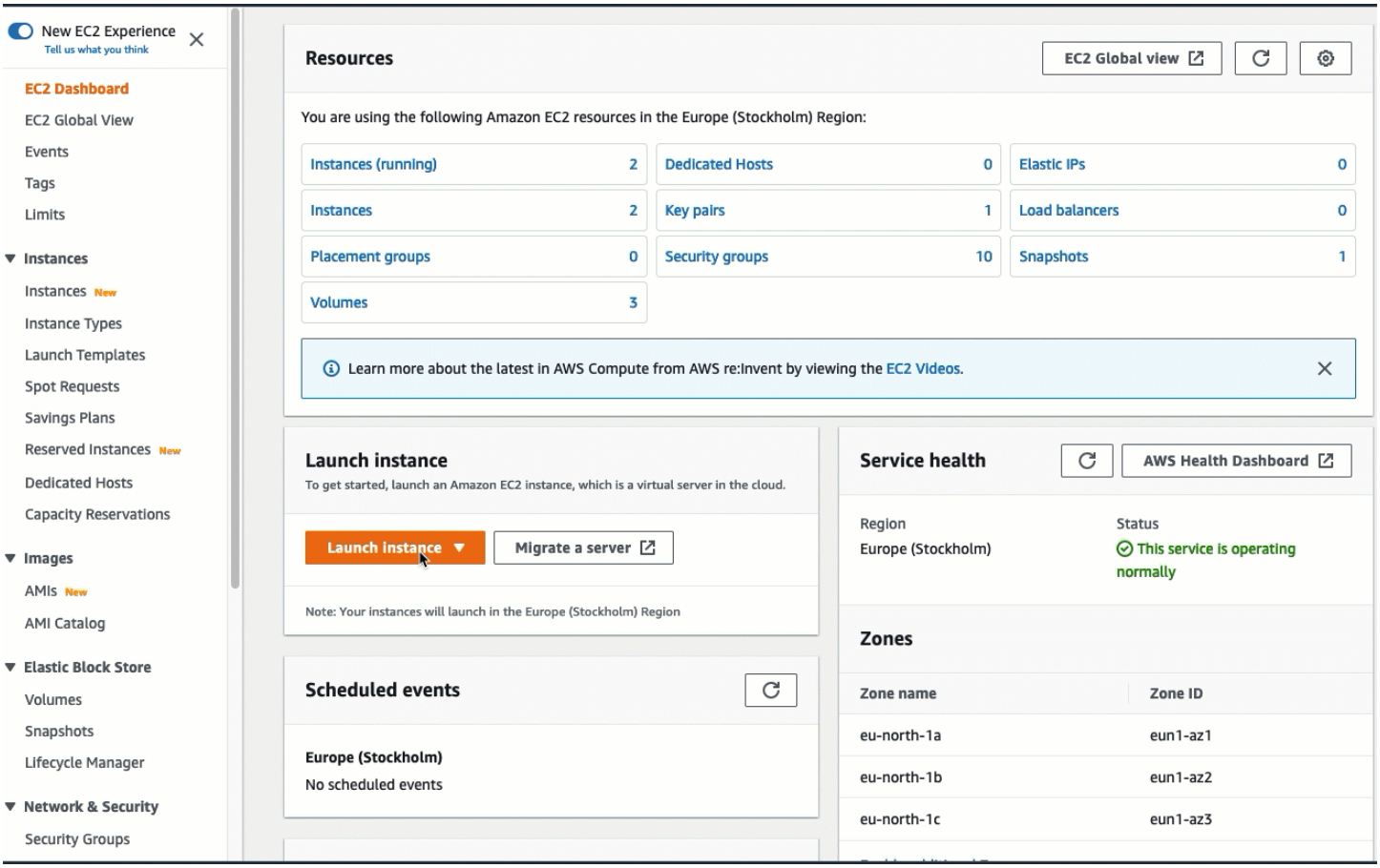
      - Allow HTTPs traffic from the internet(인터넷에서 오는 HTTPS 트래픽 허용)
      - Allow HTTP traffic from the internet(인터넷에서 오는 HTTP 트래픽 허용)

9. Summary(요약) 패널에서 인스턴스 구성을 검토한 다음, Launch instance(인스턴스 시작)를 선택합니다.
10. 확인 페이지를 연 상태로 유지합니다. 인스턴스를 데이터베이스에 자동으로 연결할 때 이는 다음 작업에 필요합니다.

    인스턴스가 시작하지 않거나 상태가 `terminated`이 아닌 `running`로 변경되는 경우, Amazon EC2 인스턴스 시작 문제 해결 섹션을 참조하세요.

인스턴스 시작에 관한 자세한 내용은 콘솔에서 인스턴스 시작 마법사를 사용하여 EC2 인스턴스 시작 섹션을 참조하세요.

애니메이션 보기: EC2 인스턴스 시작



작업 3: EC2 인스턴스를 RDS 데이터베이스에 자동으로 연결

이 작업의 목표는 EC2 콘솔에서 자동 연결 기능을 사용하여 EC2 인스턴스와 RDS 데이터베이스 간의 연결을 자동으로 구성하는 데 있습니다.

EC2 콘솔을 사용하여 EC2 인스턴스를 RDS 데이터베이스에 자동으로 연결하는 방법

1. 인스턴스 시작 확인 페이지(이전 작업에서 열어 둔 상태여야 함)에서 Connect an RDS database(RDS 데이터베이스 연결)를 선택합니다.

   확인 페이지를 닫은 경우 다음 단계를 따르세요.

   a. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
   b. 탐색 창에서 Instances(인스턴스)를 선택합니다.
   c. 방금 생성한 EC2 인스턴스를 선택한 다음, Actions(작업), Networking(네트워킹), Connect RDS database(RDS 데이터베이스 연결)를 차례로 선택합니다.

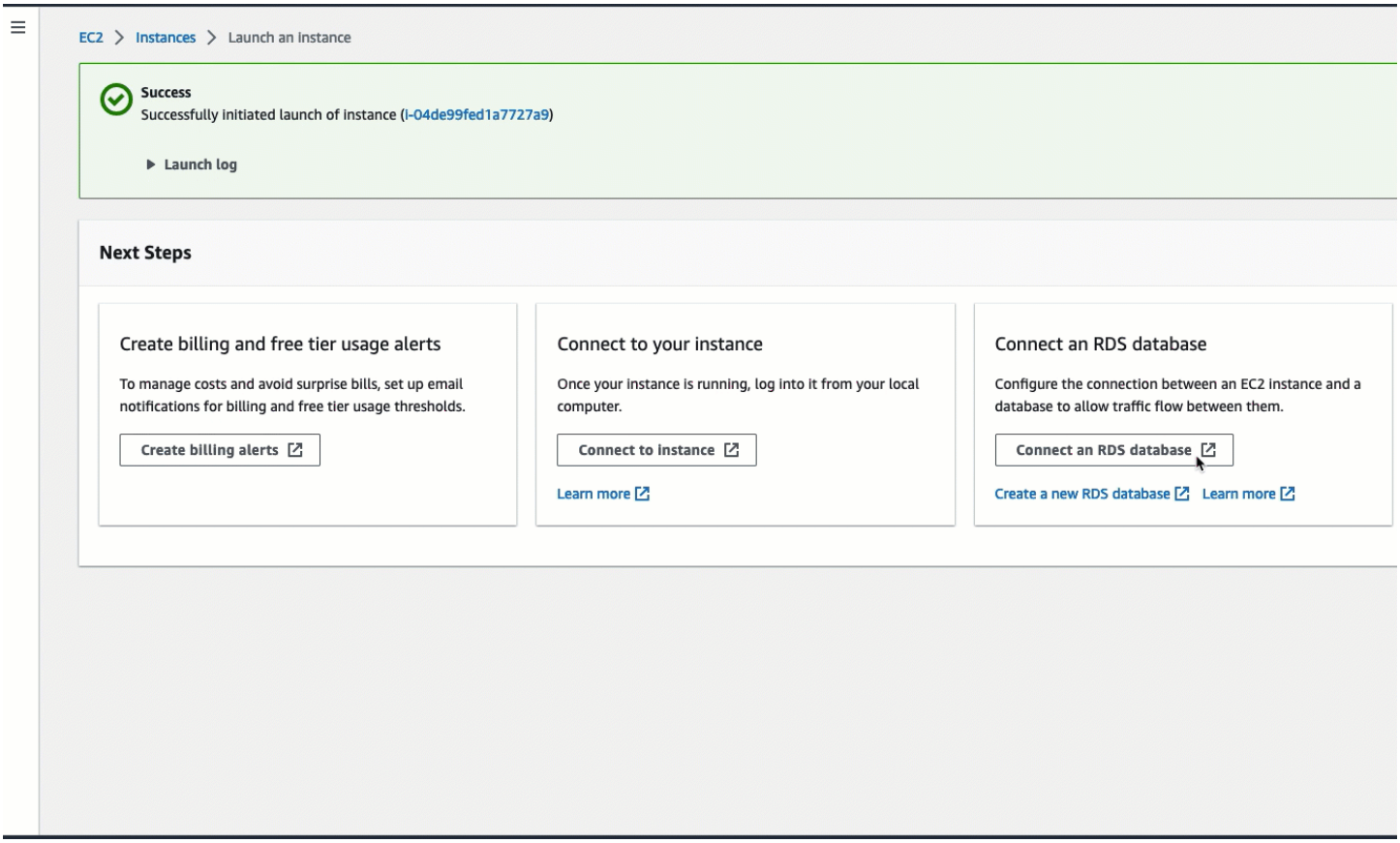
      Connect RDS database(RDS 데이터베이스 연결)를 사용할 수 없는 경우 EC2 인스턴스가 Running(실행 중) 상태인지 확인하세요.

2. Database role(데이터베이스 역할)에서 Instance(인스턴스)를 선택합니다. 이 경우 인스턴스는 데이터베이스 인스턴스를 나타냅니다.
3. RDS database(RDS 데이터베이스)의 경우 작업 1에서 생성했던 RDS 데이터베이스를 선택합니다.

   > **Note**
   >
   > EC2 인스턴스와 RDS 데이터베이스를 서로 연결하기 위해서는 이들이 동일한 VPC에 있어야 합니다.

4. 연결을 선택합니다.

애니메이션 보기: 새로 시작된 EC2 인스턴스를 RDS 데이터베이스에 자동으로 연결



작업 4: 연결 구성 확인

이 작업의 목표는 두 보안 그룹이 생성되어 인스턴스 및 데이터베이스에 할당되었는지 확인하는 데 있습니다.

콘솔에서 자동 연결 기능을 사용하여 연결을 구성하면 다음과 같이 보안 그룹이 자동으로 생성되어 인스턴스 및 데이터베이스에 할당됩니다.

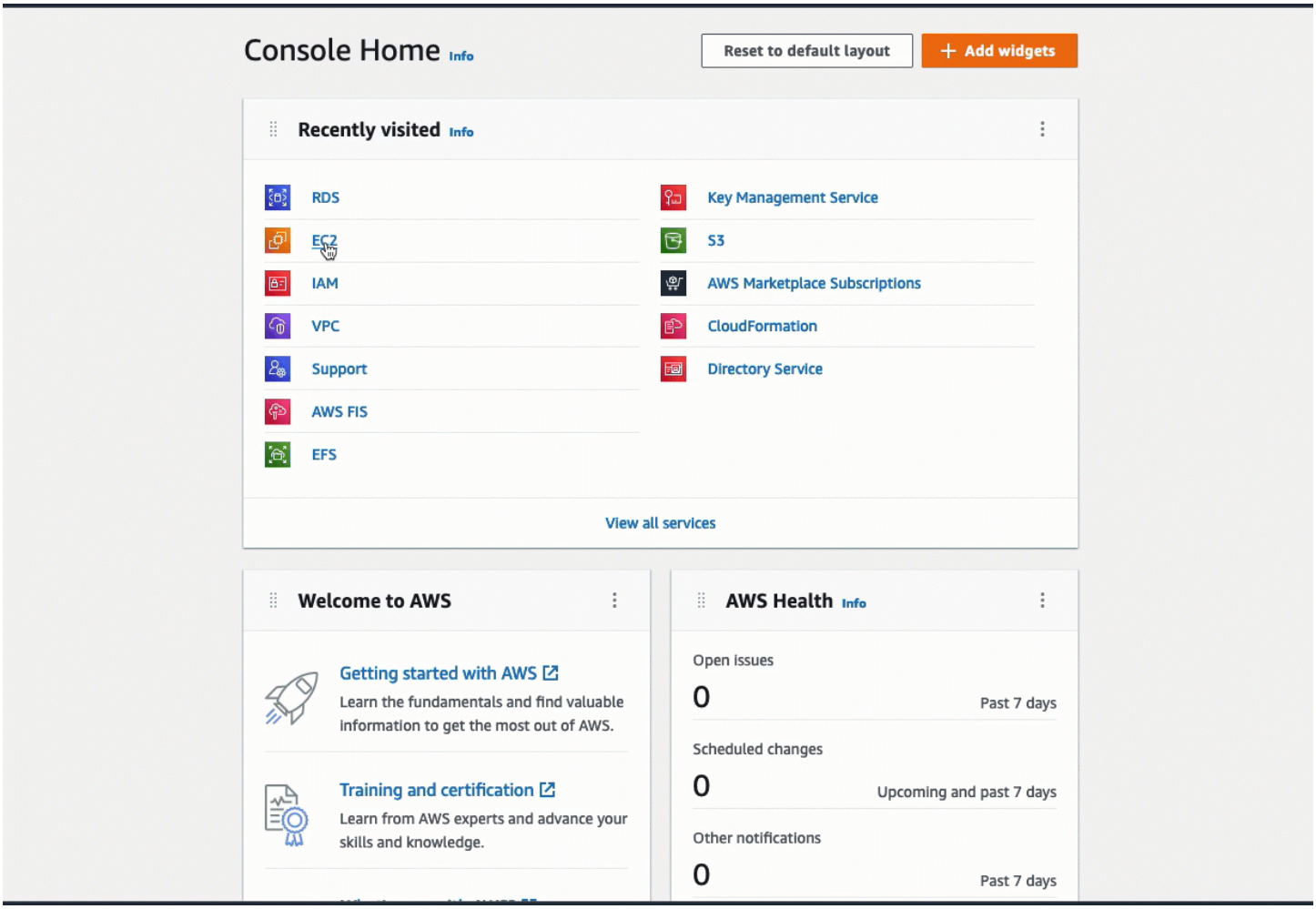
- 보안 그룹 rds-ec2-*x*가 생성되어 RDS 데이터베이스에 추가됩니다. 이 보안 그룹에는 ec2-rds-*x* 보안 그룹을 소스로 참조하는 인바운드 규칙이 하나 있습니다. 이를 통해 ec2-rds-*x* 보안 그룹이 있는 EC2 인스턴스의 트래픽이 RDS 데이터베이스에 도달할 수 있습니다.
- 보안 그룹 ec2-rds-*x*가 생성되어 EC2 인스턴스에 추가됩니다. 이 보안 그룹에는 rds-ec2-*x* 보안 그룹을 대상으로 참조하는 아웃바운드 규칙이 하나 있습니다. 이를 통해 EC2 인스턴스의 트래픽은 rds-ec2-*x* 보안 그룹이 있는 RDS 데이터베이스에 도달할 수 있습니다.

콘솔을 사용하여 연결 구성을 확인하려면 다음과 같이 하세요.

1. https://console.aws.amazon.com/rds/에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 페이지에서 Databases(데이터베이스)를 선택합니다.
3. 이 자습서용으로 생성한 RDS 데이터베이스를 선택합니다.
4. Connectivity & security(연결 & 보안) 탭의 Security(보안), VPC security groups(VPC 보안 그룹)에서 rds-ec2-*x*라는 보안 그룹이 표시되는지 확인합니다.
5. rds-ec2-*x* 보안 그룹을 선택합니다. EC2 콘솔에 Security Groups(보안 그룹) 화면이 열립니다.
6. rds-ec2-*x* 보안 그룹을 선택하여 엽니다.
7. 인바운드 규칙 탭을 선택합니다.
8. 다음과 같은 보안 그룹 규칙이 있는지 확인합니다.
   - 유형: MYSQL/Aurora
   - 포트 범위: 3306
   - 소스: ***sg-0987654321example*** / ec2-rds-*x* - 이전 단계에서 확인한 EC2 인스턴스에 할당된 보안 그룹입니다.
   - 설명: ***sg-1234567890example***이 연결된 EC2 인스턴스에서의 연결을 허용하는 규칙
9. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
10. 탐색 창에서 Instances(인스턴스)를 선택합니다.
11. 이전 작업에서 RDS 데이터베이스에 연결하기 위해 선택한 EC2 인스턴스를 선택하고 Security(보안) 탭을 선택합니다.
12. Security details(보안 세부 정보), Security groups(보안 그룹)에서 ec2-rds-*x*라는 보안 그룹이 목록에 있는지 확인합니다. 여기서 *x*는 숫자입니다.
13. ec2-rds-*x* 보안 그룹을 선택하여 엽니다.
14. Outbound rules(아웃바운드 규칙) 탭을 선택합니다.
15. 다음과 같은 보안 그룹 규칙이 있는지 확인합니다.
    - 유형: MYSQL/Aurora
    - 포트 범위: 3306
    - 대상: ***sg-1234567890example*** / rds-ec2-*x*
    - 설명: 이 보안 그룹이 연결된 인스턴스에서 **database-tutorial**로의 연결을 허용하는 규칙

이러한 보안 그룹 및 보안 그룹 규칙이 있는지와 본 절차에 설명된 대로 RDS 데이터베이스 및 EC2 인스턴스에 할당되었는지 확인함으로써 자동 연결 기능을 사용하여 연결이 자동으로 구성되었는지 확인할 수 있습니다.

애니메이션 보기: 연결 구성 확인



이 자습서의 옵션 1을 완료했습니다. 이제 RDS 콘솔을 사용하여 EC2 인스턴스를 RDS 데이터베이스에 자동으로 연결하는 방법을 알려 주는 옵션 2를 완료하거나, 옵션 1에서 자동으로 생성된 보안 그룹을 수동으로 구성하는 방법을 알려 주는 옵션 3을 완료할 수 있습니다.

작업 5(선택 사항): 정리

이제 자습서를 완료했으므로 더 이상 사용하지 않을 리소스를 정리(삭제)하는 것이 좋습니다. AWS 리소스를 정리하면 계정에 추가 요금이 발생되지 않도록 방지할 수 있습니다.

이 자습서를 위해 특별히 EC2 인스턴스를 시작한 경우 해당 인스턴스를 종료하여 관련 요금이 발생되지 않도록 할 수 있습니다.

콘솔을 사용한 인스턴스 종료 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances(인스턴스)를 선택합니다.
3. 이 자습서용으로 생성한 인스턴스를 선택하고 Instance state(인스턴스 상태), Terminate instance(인스턴스 종료)를 선택합니다.
4. 확인 메시지가 나타나면 종료를 선택합니다.

이 자습서를 위해 특별히 RDS 데이터베이스를 생성한 경우 해당 데이터베이스를 삭제하여 관련 요금이 발생되지 않도록 할 수 있습니다.

콘솔을 사용하여 RDS 데이터베이스 삭제

1. https://console.aws.amazon.com/rds/에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택합니다.
3. 이 자습서용으로 생성한 RDS 데이터베이스를 선택하고 Actions(작업), Delete(삭제)를 선택합니다.
4. 상자에 **`delete me`**를 입력한 다음, Delete(삭제)를 선택합니다.

옵션 2: RDS 콘솔을 사용하여 자동으로 RDS 데이터베이스에 인스턴스 연결

옵션 2의 목표는 EC2 인스턴스에서 RDS 데이터베이스로의 트래픽을 허용하도록 EC2 인스턴스와 RDS 데이터베이스 간의 연결을 자동으로 구성하는 RDS 콘솔의 자동 연결 기능을 살펴보는 데 있습니다. 연결을 수동으로 구성하는 방법은 옵션 3에서 알아봅니다.

업무

- 시작하기 전 준비 사항
- 작업 1(선택 사항): EC2 인스턴스 시작
- 작업 2: RDS 데이터베이스를 생성하여 EC2 인스턴스에 자동으로 연결
- 작업 3: 연결 구성 확인
- 작업 4(선택 사항): 정리

시작하기 전 준비 사항

이 자습서를 완료하려면 다음이 필요합니다.

- RDS 데이터베이스와 동일한 VPC에 있는 EC2 인스턴스입니다. 기존 EC2 인스턴스를 사용하거나 작업 1의 단계에 따라 새 인스턴스를 생성할 수 있습니다.
- 다음 작업을 호출할 수 있는 권한:
  - `ec2:AssociateRouteTable`
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:CreateRouteTable`
  - `ec2:CreateSecurityGroup`
  - `ec2:CreateSubnet`
  - `ec2:DescribeInstances`
  - `ec2:DescribeNetworkInterfaces`
  - `ec2:DescribeRouteTables`
  - `ec2:DescribeSecurityGroups`
  - `ec2:DescribeSubnets`
  - `ec2:ModifyNetworkInterfaceAttribute`
  - `ec2:RevokeSecurityGroupEgress`

작업 1(선택 사항): EC2 인스턴스 시작

> **Note**
>
> 인스턴스 시작은 이 자습서에서 중점적으로 다루는 사항이 아닙니다. Amazon EC2 인스턴스가 이미 있고 이 자습서에서 해당 인스턴스를 사용하려는 경우 이 작업을 건너뛸 수 있습니다.

이 작업의 목표는 EC2 인스턴스와 Amazon RDS 데이터베이스 간의 연결을 구성하는 작업 2를 완료할 수 있도록 EC2 인스턴스를 시작하는 데 있습니다. 이 작업의 단계에서는 다음과 같이 EC2 인스턴스를 구성합니다.

- 인스턴스 이름: **`tutorial-instance-2`**
- AMI: Amazon Linux 2
- 인스턴스 유형: `t2.micro`
- 퍼블릭 IP 자동 할당: 활성화됨
- 다음과 같은 세 가지 규칙이 있는 보안 그룹:

- IP 주소에서 SSH 허용
- 어디에서 들어오든 HTTPS 트래픽 허용
- 어디에서 들어오든 HTTP 트래픽 허용

> **Important**
>
> 프로덕션 환경에서는 특정 요구를 충족하도록 인스턴스를 구성해야 합니다.

EC2 인스턴스 시작

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. EC2 Dashboard(EC2 대시보드)에서 Launch instance(인스턴스 시작)를 선택하세요.
3. Name and tags(이름 및 태그)에서 Name(이름)에 인스턴스를 식별하는 이름을 입력합니다. 이 자습서에서는 인스턴스 이름을 **`tutorial-instance-2`**로 지정합니다. 인스턴스 이름을 반드시 입력해야 하는 것은 아니지만, RDS 콘솔에서 인스턴스를 선택할 때 해당 이름을 통해 인스턴스를 쉽게 식별할 수 있습니다.
4. Application and OS Images(애플리케이션 및 OS 이미지)에서 웹 서버 요구에 맞는 AMI를 선택합니다. 이 자습서에서는 Amazon Linux를 사용합니다.
5. Instance type(인스턴스 유형)에서 Instance type(인스턴스 유형)으로 웹 서버 요구에 맞는 인스턴스 유형을 선택합니다. 이 자습서에서는 `t2.micro`를 사용합니다.

   > **Note**
   >
   > AWS 계정을 생성한 지 12개월 미만이고 `t2.micro` 인스턴스 유형 또는 `t2.micro`를 사용할 수 없는 리전에서는 `t3.micro`를 선택한 경우 프리 티어로 Amazon EC2를 사용할 수 있습니다. t3.micro 인스턴스를 시작하면 기본적으로 무제한 모드로 설정되므로 CPU 사용량에 따라 추가 요금이 부과될 수 있다는 점에 유의하세요.

6. Key pair (login)(키 페어(로그인))에서 Key pair name(키 페어 이름)에 대해 키 페어를 선택합니다.
7. Network settings(네트워크 설정)에서 다음을 수행합니다.

   a. 기본 VPC 또는 서브넷을 변경하지 않은 경우 Network(네트워크) 및 Subnet(서브넷)의 기본 설정을 유지할 수 있습니다.

      기본 VPC 또는 서브넷을 변경한 경우 다음을 확인합니다.

i. 자동 연결 구성을 사용하려면 인스턴스가 RDS 데이터베이스와 동일한 VPC에 있어야 합니다. 기본적으로 VPC는 하나만 있습니다.

ii. 인스턴스를 시작하는 VPC에는 인터넷 게이트웨이가 연결되어 있어야 합니다. 그래야 인터넷에서 웹 서버에 액세스할 수 있습니다. 기본 VPC는 인터넷 게이트웨이를 통해 자동으로 설정됩니다.

iii. 인스턴스가 퍼블릭 IP 주소를 수신하도록 하려면 Auto-assign public IP(퍼블릭 IP 자동 할당)에 대해 Enable(활성화)이 선택되어 있는지 확인합니다. Disable(비활성화)이 선택된 경우 Edit(편집)(Network Settings(네트워크 설정) 오른쪽에 있음)를 선택한 다음, Auto-assign public IP(퍼블릭 IP 자동 할당)에서 Enable(활성화)을 선택합니다.

b. SSH를 사용하여 인스턴스에 연결하려면 컴퓨터의 퍼블릭 IPv4 주소로부터의 SSH(Linux) 또는 RDP(Windows) 트래픽을 승인하는 보안 그룹 규칙이 필요합니다. 기본적으로 인스턴스를 시작하면 어디에서 들어오든 인바운드 SSH 트래픽을 허용하는 규칙을 사용하여 새 보안 그룹이 생성됩니다.

   자체 IP 주소만 인스턴스에 연결할 수 있도록 하려면 방화벽(보안 그룹) 아래의 다음으로부터의 SSH 트래픽 허용 확인란 옆에 있는 드롭다운 목록에서 내 IP를 선택합니다.

c. 인터넷에서 인스턴스로의 트래픽을 허용하려면 다음 확인란을 선택합니다.

   - Allow HTTPs traffic from the internet(인터넷에서 오는 HTTPS 트래픽 허용)
   - Allow HTTP traffic from the internet(인터넷에서 오는 HTTP 트래픽 허용)

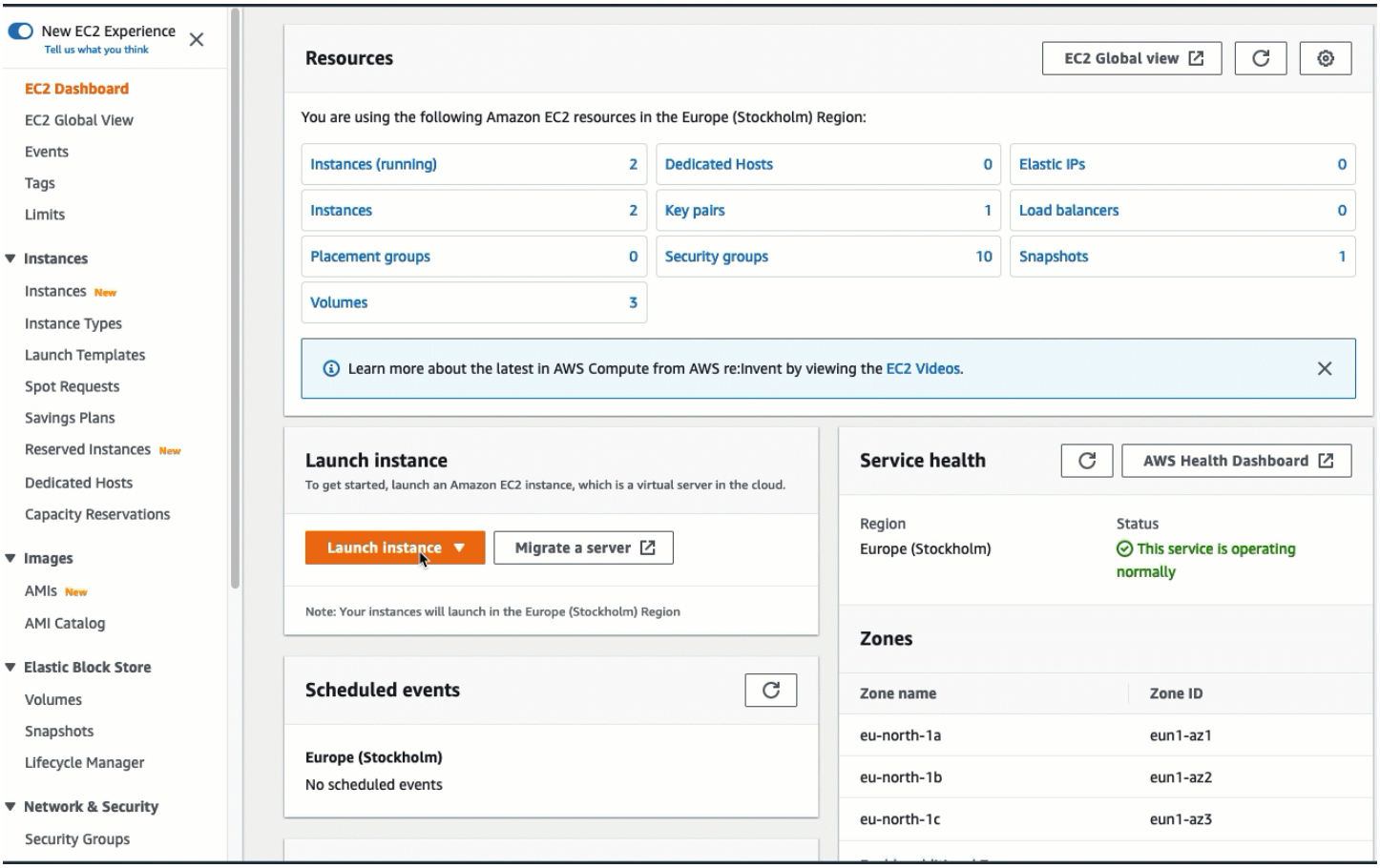
8. Summary(요약) 패널에서 인스턴스 구성을 검토한 다음, Launch instance(인스턴스 시작)를 선택합니다.

9. 모든 인스턴스 보기(View all instances)를 선택하여 확인 페이지를 닫고 콘솔로 돌아갑니다. 인스턴스는 먼저 `pending` 상태에 있다가 `running` 상태가 됩니다.

   인스턴스가 시작하지 않거나 상태가 `terminated`이 아닌 `running`로 변경되는 경우, Amazon EC2 인스턴스 시작 문제 해결 섹션을 참조하세요.

인스턴스 시작에 관한 자세한 내용은 콘솔에서 인스턴스 시작 마법사를 사용하여 EC2 인스턴스 시작 섹션을 참조하세요.

애니메이션 보기: EC2 인스턴스 시작



작업 2: RDS 데이터베이스를 생성하여 EC2 인스턴스에 자동으로 연결

이 작업의 목표는 RDS 콘솔에서 RDS 데이터베이스를 생성하고 자동 연결 기능을 사용하여 EC2 인스턴스와 RDS 데이터베이스 간의 연결을 자동으로 구성하는 데 있습니다. 이 작업의 단계에서는 다음과 같이 DB 인스턴스를 구성합니다.

- 엔진 유형: MySQL
- 템플릿: 프리 티어
- DB 인스턴스 식별자: **`tutorial-database`**
- DB 인스턴스 클래스: `db.t3.micro`

> **Important**
>
> 프로덕션 환경에서는 특정 요구를 충족하도록 인스턴스를 구성해야 합니다.

RDS 데이터베이스를 생성하여 EC2 인스턴스에 자동으로 연결하는 방법

1. https://console.aws.amazon.com/rds/에서 Amazon RDS 콘솔을 엽니다.
2. 리전 선택기(오른쪽 상단)에서 EC2 인스턴스를 생성한 AWS 리전을 선택합니다. EC2 인스턴스와 RDS 데이터베이스가 동일한 리전에 있어야 합니다.
3. 대시보드에서 Create database(데이터베이스 생성)를 선택합니다.
4. Choose a database creation method(데이터베이스 생성 방법 선택)에서 Standard create(표준 생성)가 선택되어 있는지 확인합니다. Easy create(손쉬운 생성)를 선택한 경우 자동 연결 기능을 사용할 수 없습니다.
5. Engine options(엔진 옵션)에서 Engine type(엔진 유형)으로 MySQL을 선택합니다.
6. Templates(템플릿)에서 요구에 맞는 샘플 템플릿을 선택합니다. 이 자습서에서는 Free tier(프리 티어)를 선택하여 무료로 RDS 데이터베이스를 생성합니다. 그러나 프리 티어는 계정을 생성한 지 12개월 미만인 경우에만 사용할 수 있습니다. 그리고 다른 제한 사항이 적용됩니다. Free tier(프리 티어) 상자에서 Info(정보) 링크를 선택하면 자세한 내용을 확인할 수 있습니다.
7. 설정에서 다음을 수행합니다.

   a. DB instance identifier(DB 인스턴스 식별자)에 데이터베이스 이름을 입력합니다. 이 자습서에서는 **`tutorial-database`**을 입력합니다.

   b. Master username(마스터 사용자 이름)의 경우 기본 이름인 **`admin`**을 그대로 둡니다.

   c. Master password(마스터 암호)에 이 자습서에서 기억할 수 있는 암호를 입력한 다음, Confirm password(암호 확인)에 암호를 다시 입력합니다.
8. 인스턴스 구성에서 DB 인스턴스 클래스의 기본값인 db.t3.micro를 그대로 둡니다. 계정을 생성한 지 12개월 미만인 경우 이 인스턴스를 무료로 사용할 수 있습니다. 그리고 다른 제한 사항이 적용됩니다. 자세한 내용은 AWS 프리 티어 섹션을 참조하세요.
9. Connectivity(연결)에서 Compute resource(컴퓨팅 리소스)에 대해 Connect to an EC2 compute resource(EC2 컴퓨팅 리소스에 연결)를 선택합니다. 이는 RDS 콘솔의 자동 연결 기능입니다.
10. EC2 instance(EC2 인스턴스)에서 연결하려는 EC2 인스턴스를 선택합니다. 이 자습서에서는 이전 작업에서 생성했던 인스턴스(이름을 **`tutorial-instance`**로 지정함)를 선택하거나 기존의 다른 인스턴스를 선택할 수 있습니다. 목록에 인스턴스가 표시되지 않으면 Connectivity(연결) 오른쪽에 있는 새로 고침 아이콘을 선택합니다.

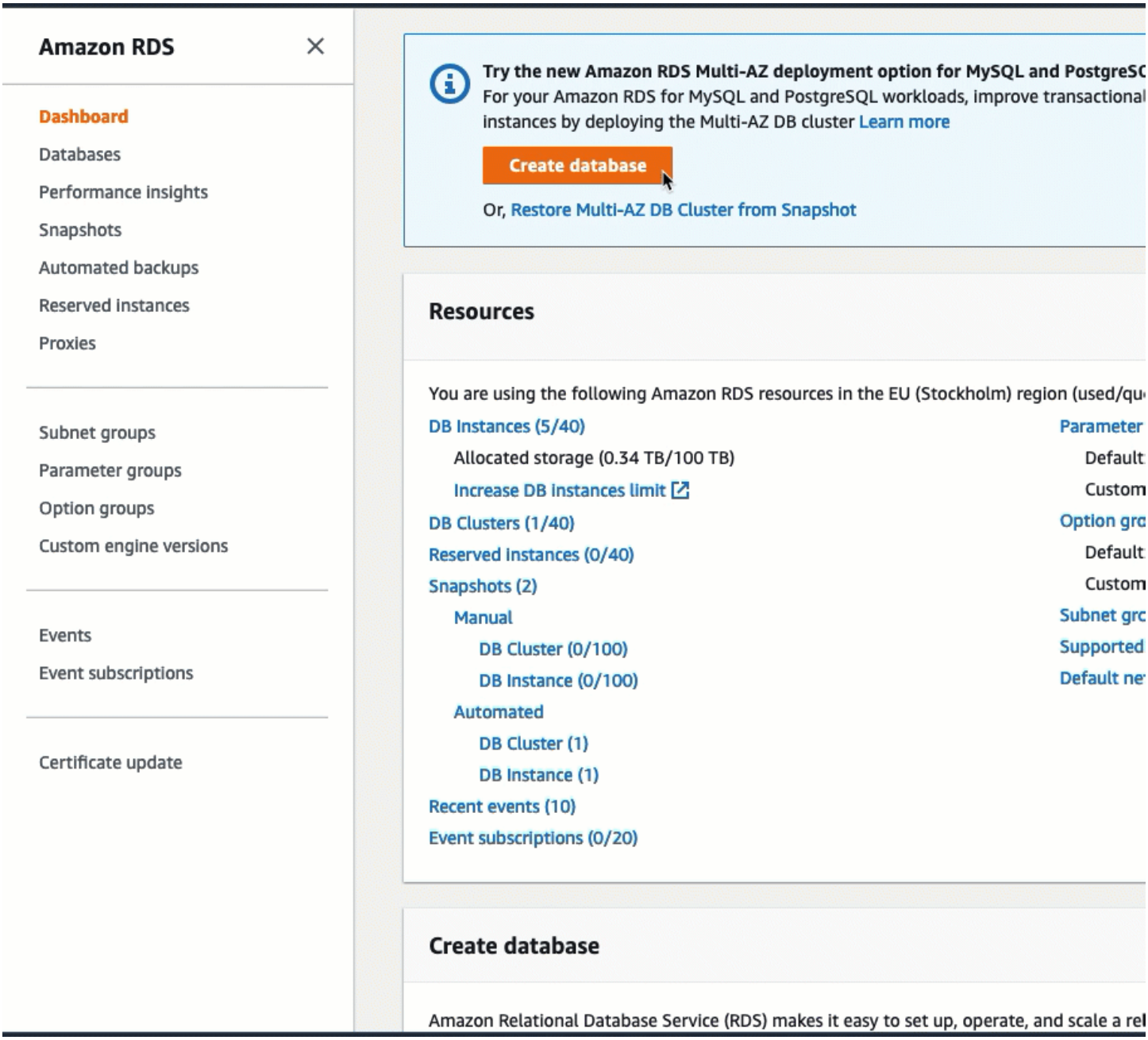
    자동 연결 기능을 사용하면 이 EC2 인스턴스에 보안 그룹이 추가되고 RDS 데이터베이스에 다른 보안 그룹이 추가됩니다. 보안 그룹은 EC2 인스턴스와 RDS 데이터베이스 간의 트래픽을 허용하도록 자동으로 구성됩니다. 다음 작업에서는 보안 그룹이 생성되어 EC2 인스턴스 및 RDS 데이터베이스에 할당되었는지 확인합니다.

11. 데이터베이스 생성를 선택합니다.

    Databases(데이터베이스) 화면에서 데이터베이스를 사용할 준비가 될 때까지 새 데이터베이스의 Status(상태)는 Creating(생성 중)입니다. 상태가 Available(사용 가능)로 변경되면 데이터베이스에 연결할 수 있습니다. 데이터베이스 클래스 및 스토리지 양에 따라 새 데이터베이스를 사용할 수 있게 되기까지 최대 20분이 걸릴 수 있습니다.

자세히 알아보려면 Amazon RDS 사용 설명서의 EC2 인스턴스와의 자동 네트워크 연결 구성을 참조하세요.

애니메이션 보기: RDS 데이터베이스를 생성하여 EC2 인스턴스에 자동으로 연결



작업 3: 연결 구성 확인

이 작업의 목표는 두 보안 그룹이 생성되어 인스턴스 및 데이터베이스에 할당되었는지 확인하는 데 있습니다.

콘솔에서 자동 연결 기능을 사용하여 연결을 구성하면 다음과 같이 보안 그룹이 자동으로 생성되어 인스턴스 및 데이터베이스에 할당됩니다.

- 보안 그룹 rds-ec2-*x*가 생성되어 RDS 데이터베이스에 추가됩니다. 이 보안 그룹에는 ec2-rds-*x* 보안 그룹을 소스로 참조하는 인바운드 규칙이 하나 있습니다. 이를 통해 ec2-rds-*x* 보안 그룹이 있는 EC2 인스턴스의 트래픽이 RDS 데이터베이스에 도달할 수 있습니다.
- 보안 그룹 ec2-rds-*x*가 생성되어 EC2 인스턴스에 추가됩니다. 이 보안 그룹에는 rds-ec2-*x* 보안 그룹을 대상으로 참조하는 아웃바운드 규칙이 하나 있습니다. 이를 통해 EC2 인스턴스의 트래픽은 rds-ec2-*x* 보안 그룹이 있는 RDS 데이터베이스에 도달할 수 있습니다.

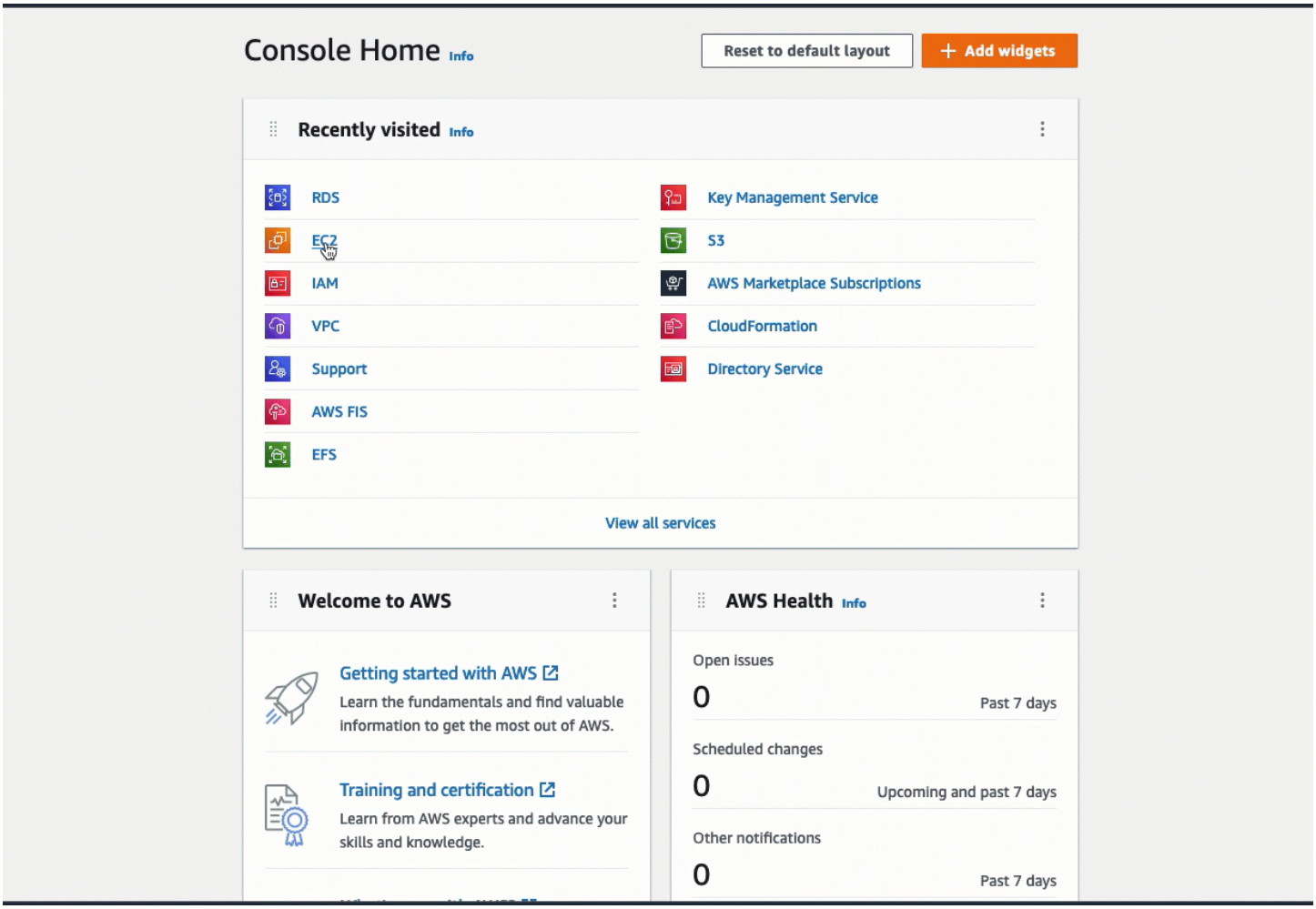
콘솔을 사용하여 연결 구성을 확인하려면·다음과·같이·하세요.

1. https://console.aws.amazon.com/rds/에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 페이지에서 Databases(데이터베이스)를 선택합니다.
3. 이 자습서용으로 생성한 RDS 데이터베이스를 선택합니다.
4. Connectivity & security(연결 & 보안) 탭의 Security(보안), VPC security groups(VPC 보안 그룹)에서 rds-ec2-*x*라는 보안 그룹이 표시되는지 확인합니다.
5. rds-ec2-*x* 보안 그룹을 선택합니다. EC2 콘솔에 Security Groups(보안 그룹) 화면이 열립니다.
6. rds-ec2-*x* 보안 그룹을 선택하여 엽니다.
7. 인바운드 규칙 탭을 선택합니다.
8. 다음과 같은 보안 그룹 규칙이 있는지 확인합니다.
   - 유형: MYSQL/Aurora
   - 포트 범위: 3306
   - 소스: ***sg-0987654321example*** / ec2-rds-*x* - 이전 단계에서 확인한 EC2 인스턴스에 할당된 보안 그룹입니다.
   - 설명: ***sg-1234567890example***이 연결된 EC2 인스턴스에서의 연결을 허용하는 규칙
9. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
10. 탐색 창에서 Instances(인스턴스)를 선택합니다.
11. 이전 작업에서 RDS 데이터베이스에 연결하기 위해 선택한 EC2 인스턴스를 선택하고 Security(보안) 탭을 선택합니다.
12. Security details(보안 세부 정보), Security groups(보안 그룹)에서 ec2-rds-*x*라는 보안 그룹이 목록에 있는지 확인합니다. 여기서 *x*는 숫자입니다.
13. ec2-rds-*x* 보안 그룹을 선택하여 엽니다.
14. Outbound rules(아웃바운드 규칙) 탭을 선택합니다.
15. 다음과 같은 보안 그룹 규칙이 있는지 확인합니다.

- 유형: MYSQL/Aurora
- 포트 범위: 3306
- 대상: ***sg-1234567890example*** / rds-ec2-***x***
- 설명: 이 보안 그룹이 연결된 인스턴스에서 **database-tutorial**로의 연결을 허용하는 규칙

이러한 보안 그룹 및 보안 그룹 규칙이 있는지와 본 절차에 설명된 대로 RDS 데이터베이스 및 EC2 인스턴스에 할당되었는지 확인함으로써 자동 연결 기능을 사용하여 연결이 자동으로 구성되었는지 확인할 수 있습니다.

애니메이션 보기: 연결 구성 확인



이 자습서의 옵션 2를 완료했습니다. 이제 옵션 2에서 자동으로 생성된 보안 그룹을 수동으로 구성하는 방법을 알려 주는 옵션 3을 완료할 수 있습니다.

작업 4(선택 사항): 정리

이제 자습서를 완료했으므로 더 이상 사용하지 않을 리소스를 정리(삭제)하는 것이 좋습니다. AWS 리소스를 정리하면 계정에 추가 요금이 발생되지 않도록 방지할 수 있습니다.

이 자습서를 위해 특별히 EC2 인스턴스를 시작한 경우 해당 인스턴스를 종료하여 관련 요금이 발생되지 않도록 할 수 있습니다.

콘솔을 사용한 인스턴스 종료 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances(인스턴스)를 선택합니다.
3. 이 자습서용으로 생성한 인스턴스를 선택하고 Instance state(인스턴스 상태), Terminate instance(인스턴스 종료)를 선택합니다.
4. 확인 메시지가 나타나면 종료를 선택합니다.

이 자습서를 위해 특별히 RDS 데이터베이스를 생성한 경우 해당 데이터베이스를 삭제하여 관련 요금이 발생되지 않도록 할 수 있습니다.

콘솔을 사용하여 RDS 데이터베이스 삭제

1. https://console.aws.amazon.com/rds/에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택합니다.
3. 이 자습서용으로 생성한 RDS 데이터베이스를 선택하고 Actions(작업), Delete(삭제)를 선택합니다.
4. 상자에 **`delete me`**를 입력한 다음, Delete(삭제)를 선택합니다.

옵션 3: 보안 그룹을 생성하여 수동으로 RDS 데이터베이스에 인스턴스 연결

옵션 3의 목표는 자동 연결 기능의 구성을 수동으로 재현하여 EC2 인스턴스와 RDS 데이터베이스 간의 연결을 수동으로 구성하는 방법을 알아보는 데 있습니다.

업무

- 시작하기 전 준비 사항
- 작업 1(선택 사항): EC2 인스턴스 시작
- 작업 2(선택 사항): RDS 데이터베이스 생성

- 작업 3: 보안 그룹을 생성하여 인스턴스에 할당함으로써 EC2 인스턴스를 RDS 데이터베이스에 수동으로 연결
- 작업 4(선택 사항): 정리

시작하기 전 준비 사항

이 자습서를 완료하려면 다음이 필요합니다.

- RDS 데이터베이스와 동일한 VPC에 있는 EC2 인스턴스입니다. 기존 EC2 인스턴스를 사용하거나 작업 1의 단계에 따라 새 인스턴스를 생성할 수 있습니다.
- EC2 인스턴스와 동일한 VPC에 있는 RDS 데이터베이스 - 기존 RDS 데이터베이스를 사용하거나 작업 2의 단계에 따라 새 데이터베이스를 생성할 수 있습니다.
- 다음 작업을 호출할 수 있는 권한:
  - `ec2:AssociateRouteTable`
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:CreateRouteTable`
  - `ec2:CreateSecurityGroup`
  - `ec2:CreateSubnet`
  - `ec2:DescribeInstances`
  - `ec2:DescribeNetworkInterfaces`
  - `ec2:DescribeRouteTables`
  - `ec2:DescribeSecurityGroups`
  - `ec2:DescribeSubnets`
  - `ec2:ModifyNetworkInterfaceAttribute`
  - `ec2:RevokeSecurityGroupEgress`

작업 1(선택 사항): EC2 인스턴스 시작

> **Note**
>
> 인스턴스 시작은 이 자습서에서 중점적으로 다루는 사항이 아닙니다. Amazon EC2 인스턴스가 이미 있고 이 자습서에서 해당 인스턴스를 사용하려는 경우 이 작업을 건너뛸 수 있습니다.

이 작업의 목표는 EC2 인스턴스와 Amazon RDS 데이터베이스 간의 연결을 구성하는 작업 3을 완료할 수 있도록 EC2 인스턴스를 시작하는 데 있습니다. 이 작업의 단계에서는 다음과 같이 EC2 인스턴스를 구성합니다.

- 인스턴스 이름: **`tutorial-instance`**
- AMI: Amazon Linux 2
- 인스턴스 유형: `t2.micro`
- 퍼블릭 IP 자동 할당: 활성화됨
- 다음과 같은 세 가지 규칙이 있는 보안 그룹:
  - IP 주소에서 SSH 허용
  - 어디에서 들어오든 HTTPS 트래픽 허용
  - 어디에서 들어오든 HTTP 트래픽 허용

> **Important**
>
> 프로덕션 환경에서는 특정 요구를 충족하도록 인스턴스를 구성해야 합니다.

EC2 인스턴스 시작

1. AWS Management Console에 로그인하고 https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. EC2 Dashboard(EC2 대시보드)에서 Launch instance(인스턴스 시작)를 선택하세요.
3. Name and tags(이름 및 태그)에서 Name(이름)에 인스턴스를 식별하는 이름을 입력합니다. 이 자습서에서는 인스턴스 이름을 **`tutorial-instance-manual-1`**로 지정합니다. 인스턴스 이름을 반드시 입력해야 하는 것은 아니지만, 해당 이름을 통해 인스턴스를 쉽게 식별할 수 있습니다.
4. Application and OS Images(애플리케이션 및 OS 이미지)에서 웹 서버 요구에 맞는 AMI를 선택합니다. 이 자습서에서는 Amazon Linux를 사용합니다.
5. Instance type(인스턴스 유형)에서 Instance type(인스턴스 유형)으로 웹 서버 요구에 맞는 인스턴스 유형을 선택합니다. 이 자습서에서는 `t2.micro`를 사용합니다.

   > **Note**
   >
   > AWS 계정을 생성한 지 12개월 미만이고 `t2.micro` 인스턴스 유형 또는 `t2.micro`를 사용할 수 없는 리전에서는 `t3.micro`를 선택한 경우 프리 티어로 Amazon EC2를 사용할

수 있습니다. t3.micro 인스턴스를 시작하면 기본적으로 무제한 모드로 설정되므로 CPU 사용량에 따라 추가 요금이 부과될 수 있다는 점에 유의하세요.

6. Key pair (login)(키 페어(로그인))에서 Key pair name(키 페어 이름)에 대해 키 페어를 선택합니다.
7. Network settings(네트워크 설정)에서 다음을 수행합니다.
   a. 기본 VPC 또는 서브넷을 변경하지 않은 경우 Network(네트워크) 및 Subnet(서브넷)의 기본 설정을 유지할 수 있습니다.

      기본 VPC 또는 서브넷을 변경한 경우 다음을 확인합니다.

      i. 인스턴스는 RDS 데이터베이스와 동일한 VPC에 있어야 합니다. 기본적으로 VPC는 하나만 있습니다.
      ii. 인스턴스를 시작하는 VPC에는 인터넷 게이트웨이가 연결되어 있어야 합니다. 그래야 인터넷에서 웹 서버에 액세스할 수 있습니다. 기본 VPC는 인터넷 게이트웨이를 통해 자동으로 설정됩니다.
      iii. 인스턴스가 퍼블릭 IP 주소를 수신하도록 하려면 Auto-assign public IP(퍼블릭 IP 자동 할당)에 대해 Enable(활성화)이 선택되어 있는지 확인합니다. Disable(비활성화)이 선택된 경우 Edit(편집)(Network Settings(네트워크 설정) 오른쪽에 있음)를 선택한 다음, Auto-assign public IP(퍼블릭 IP 자동 할당)에서 Enable(활성화)을 선택합니다.
   b. SSH를 사용하여 인스턴스에 연결하려면 컴퓨터의 퍼블릭 IPv4 주소로부터의 SSH(Linux) 또는 RDP(Windows) 트래픽을 승인하는 보안 그룹 규칙이 필요합니다. 기본적으로 인스턴스를 시작하면 어디에서 들어오든 인바운드 SSH 트래픽을 허용하는 규칙을 사용하여 새 보안 그룹이 생성됩니다.

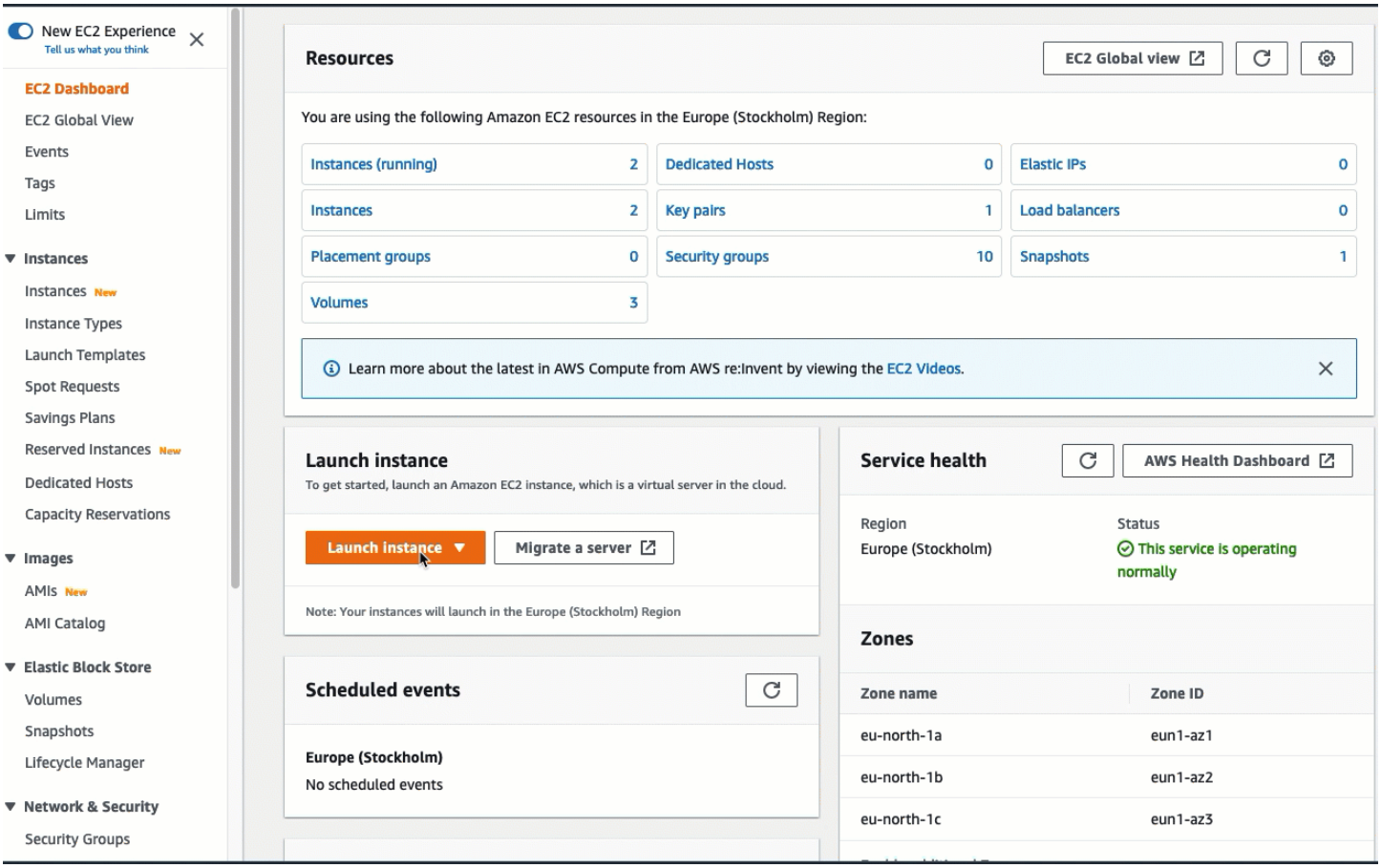
      자체 IP 주소만 인스턴스에 연결할 수 있도록 하려면 방화벽(보안 그룹) 아래의 다음으로부터의 SSH 트래픽 허용 확인란 옆에 있는 드롭다운 목록에서 내 IP를 선택합니다.
   c. 인터넷에서 인스턴스로의 트래픽을 허용하려면 다음 확인란을 선택합니다.

      - Allow HTTPs traffic from the internet(인터넷에서 오는 HTTPS 트래픽 허용)
      - Allow HTTP traffic from the internet(인터넷에서 오는 HTTP 트래픽 허용)
8. Summary(요약) 패널에서 인스턴스 구성을 검토한 다음, Launch instance(인스턴스 시작)를 선택합니다.
9. 모든 인스턴스 보기(View all instances)를 선택하여 확인 페이지를 닫고 콘솔로 돌아갑니다. 인스턴스는 먼저 `pending` 상태에 있다가 `running` 상태가 됩니다.

인스턴스가 시작하지 않거나 상태가 `terminated`이 아닌 `running`로 변경되는 경우, Amazon EC2 인스턴스 시작 문제 해결 섹션을 참조하세요.

인스턴스 시작에 관한 자세한 내용은 콘솔에서 인스턴스 시작 마법사를 사용하여 EC2 인스턴스 시작 섹션을 참조하세요.

애니메이션 보기: EC2 인스턴스 시작



작업 2(선택 사항): RDS 데이터베이스 생성

> **Note**
>
> RDS 데이터베이스 생성은 자습서의 이 부분에서 중점적으로 다루는 사항이 아닙니다. RDS 데이터베이스가 이미 있고 이 자습서에서 해당 데이터베이스를 사용하려는 경우 이 작업을 건너뛸 수 있습니다.

이 작업의 목표는 RDS 데이터베이스를 생성하는 데 있습니다. 작업 3에서 EC2 인스턴스에 연결할 때 이 인스턴스를 사용합니다. 이 작업의 단계에서는 다음과 같이 RDS 데이터베이스를 구성합니다.

- 엔진 유형: MySQL
- 템플릿: 프리 티어
- DB 인스턴스 식별자: **`tutorial-database-manual`**
- DB 인스턴스 클래스: `db.t3.micro`

> **Important**
>
> 프로덕션 환경에서는 특정 요구를 충족하도록 인스턴스를 구성해야 합니다.

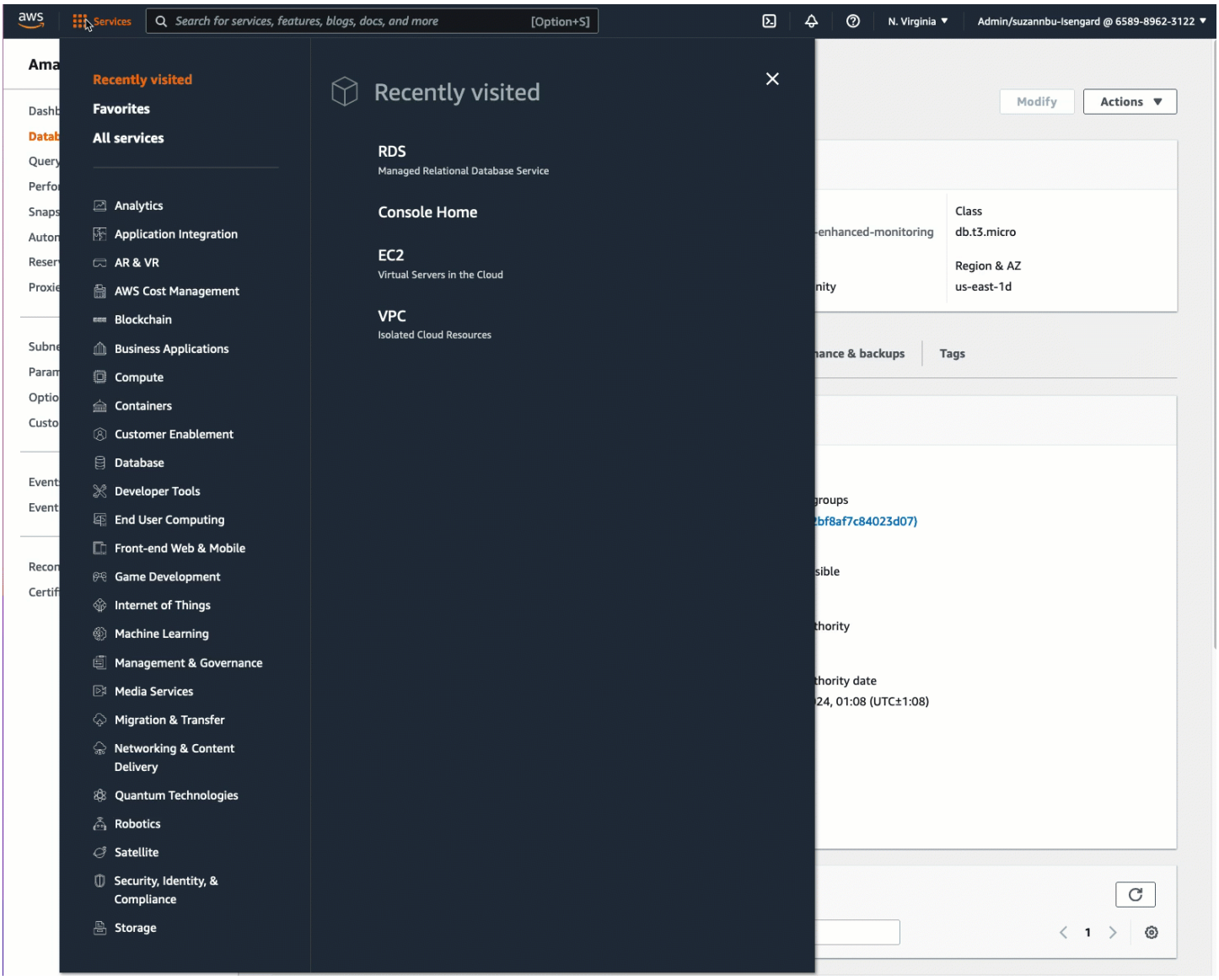
MySQL DB 인스턴스를 만들려면

1. https://console.aws.amazon.com/rds/에서 Amazon RDS 콘솔을 엽니다.
2. 리전 선택기(오른쪽 상단)에서 EC2 인스턴스를 생성한 AWS 리전을 선택합니다. EC2 인스턴스와 DB 인스턴스가 동일한 리전에 있어야 합니다.
3. 대시보드에서 Create database(데이터베이스 생성)를 선택합니다.
4. Choose a database creation method(데이터베이스 생성 방법 선택)에서 Easy create(손쉬운 생성)를 선택합니다. 이 옵션을 선택하면 연결을 자동으로 구성하는 자동 연결 기능을 사용할 수 없습니다.
5. Engine options(엔진 옵션)에서 Engine type(엔진 유형)으로 MySQL을 선택합니다.
6. DB instance size(DB 인스턴스 크기)에서 프리 티어를 선택합니다.
7. DB instance identifier(DB 인스턴스 식별자)에 RDS 데이터베이스 이름을 입력합니다. 이 자습서에서는 **`tutorial-database-manual`**을 입력합니다.
8. Master username(마스터 사용자 이름)의 경우 기본 이름인 **`admin`**을 그대로 둡니다.
9. Master password(마스터 암호)에 이 자습서에서 기억할 수 있는 암호를 입력한 다음, Confirm password(암호 확인)에 암호를 다시 입력합니다.
10. 데이터베이스 생성를 선택합니다.

    Databases(데이터베이스) 화면에서 DB 인스턴스를 사용할 준비가 될 때까지 새 DB 인스턴스의 Status(상태)는 Creating(생성 중)입니다. 상태가 Available(사용 가능)로 변경되면 DB 인스턴스에

연결할 수 있습니다. DB 인스턴스 클래스와 스토리지의 양에 따라 새 인스턴스를 사용할 수 있을 때까지 최대 20분이 걸릴 수 있습니다.

애니메이션 보기: DB 인스턴스 생성



작업 3: 보안 그룹을 생성하여 인스턴스에 할당함으로써 EC2 인스턴스를 RDS 데이터베이스에 수동으로 연결

이 작업의 목표는 두 개의 새 보안 그룹을 생성한 다음, EC2 인스턴스 및 RDS 데이터베이스에 각각 보안 그룹을 추가하는 작업을 수동으로 수행하여 자동 연결 기능의 연결 구성을 재현하는 데 있습니다.

두 개의 새 보안 그룹을 생성하고 EC2 인스턴스 및 RDS 데이터베이스에 각각 하나씩 할당하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 먼저, 다음과 같이 EC2 인스턴스에 추가할 보안 그룹을 생성합니다.
   a. 탐색 창에서 보안 그룹을 선택합니다.
   b. 보안 그룹 생성을 선택합니다.
   c. Security group name(보안 그룹 이름)에 보안 그룹을 설명하는 이름을 입력합니다. 이 자습서에서는 **ec2-rds-manual-configuration**을 입력합니다.
   d. Description(설명)에 간략한 설명을 입력합니다. 이 자습서에서는 **EC2 instance security group to allow EC2 instance to securely connect to RDS database**을 입력합니다.
   e. 보안 그룹 생성을 선택합니다. RDS 데이터베이스 보안 그룹을 생성한 후 이 보안 그룹으로 돌아와 아웃바운드 규칙을 추가합니다.
3. 이제 다음과 같이 RDS 데이터베이스에 추가할 보안 그룹을 생성합니다.
   a. 탐색 창에서 보안 그룹을 선택합니다.
   b. 보안 그룹 생성을 선택합니다.
   c. Security group name(보안 그룹 이름)에 보안 그룹을 설명하는 이름을 입력합니다. 이 자습서에서는 **rds-ec2-manual-configuration**을 입력합니다.
   d. Description(설명)에 간략한 설명을 입력합니다. 이 자습서에서는 **RDS database security group to allow EC2 instance to securely connect to RDS database**을 입력합니다.
   e. Inbound rules(인바운드 규칙)에서 Add rule(규칙 추가)을 선택하고 다음을 수행합니다.
      i. Type(유형)에서 MYSQL/Aurora를 선택합니다.
      ii. Source(소스)에서 본 절차의 2단계에서 생성한 EC2 인스턴스 보안 그룹 ec2-rds-manual-configuration을 선택합니다.
   f. 보안 그룹 생성을 선택합니다.
4. 다음과 같이 EC2 인스턴스 보안 그룹을 편집하여 아웃바운드 규칙을 추가합니다.
   a. 탐색 창에서 보안 그룹을 선택합니다.
   b. EC2 인스턴스 보안 그룹(이름을 **ec2-rds-manual-configuration**으로 지정함)을 선택하고 Outbound rules(아웃바운드 규칙) 탭을 선택합니다.
   c. Edit outbound rules(아웃바운드 규칙 편집)를 선택합니다.

d. Add rule(규칙 추가)을 선택하고 다음을 수행합니다.

   i. Type(유형)에서 MYSQL/Aurora를 선택합니다.

   ii. 대상에서 본 절차의 3단계에서 생성한 RDS 데이터베이스 보안 그룹 rds-ec2-manual-configuration을 선택합니다.

   iii. 규칙 저장을 선택합니다.

5. 다음과 같이 EC2 인스턴스 보안 그룹을 EC2 인스턴스에 추가합니다.

   a. 탐색 창에서 Instances(인스턴스)를 선택합니다.

   b. EC2 인스턴스를 선택하고 Actions(작업), Security(보안), Change security groups(보안 그룹 변경)를 선택합니다.

   c. Associated security groups(연결된 보안 그룹)에서 Select security groups(보안 그룹 선택) 필드를 선택하고 앞서 생성한 ec2-rds-manual-configuration을 선택한 다음, Add security group(보안 그룹 추가)을 선택합니다.

   d. Save(저장)를 선택합니다.

6. 다음과 같이 RDS 데이터베이스 보안 그룹을 RDS 데이터베이스에 추가합니다.

   a. https://console.aws.amazon.com/rds/에서 Amazon RDS 콘솔을 엽니다.

   b. 탐색 창에서 Databases(데이터베이스)를 선택하고 데이터베이스를 선택합니다.

   c. 수정을 선택합니다.

   d. Connectivity(연결)에서 Security group(보안 그룹)에 대해 앞서 생성한 rds-ec2-manual-configuration을 선택한 다음, Continue(계속)를 선택합니다.

   e. Scheduling of modifications(수정 사항 예약)에서 Apply immediately(즉시 적용)를 선택합니다.

   f. Modify DB instance(DB 인스턴스 수정)를 선택합니다.

   이제 자동 연결 기능 사용 시 수행되는 자동 단계를 모방하는 수동 단계를 완료했습니다.

이 자습서의 옵션 3을 완료했습니다. 옵션 1, 2 및 3을 완료했으며 이 자습서에서 생성한 리소스가 더 이상 필요하지 않은 경우 불필요한 비용이 발생하지 않도록 해당 리소스를 삭제해야 합니다. 자세한 내용은 작업 4(선택 사항): 정리 단원을 참조하십시오.

작업 4(선택 사항): 정리

이제 자습서를 완료했으므로 더 이상 사용하지 않을 리소스를 정리(삭제)하는 것이 좋습니다. AWS 리소스를 정리하면 계정에 추가 요금이 발생되지 않도록 방지할 수 있습니다.

이 자습서를 위해 특별히 EC2 인스턴스를 시작한 경우 해당 인스턴스를 종료하여 관련 요금이 발생되지 않도록 할 수 있습니다.

콘솔을 사용한 인스턴스 종료 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances(인스턴스)를 선택합니다.
3. 이 자습서용으로 생성한 인스턴스를 선택하고 Instance state(인스턴스 상태), Terminate instance(인스턴스 종료)를 선택합니다.
4. 확인 메시지가 나타나면 종료를 선택합니다.

이 자습서를 위해 특별히 RDS 데이터베이스를 생성한 경우 해당 데이터베이스를 삭제하여 관련 요금이 발생되지 않도록 할 수 있습니다.

콘솔을 사용하여 RDS 데이터베이스 삭제

1. https://console.aws.amazon.com/rds/에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택합니다.
3. 이 자습서용으로 생성한 RDS 데이터베이스를 선택하고 Actions(작업), Delete(삭제)를 선택합니다.
4. 상자에 **`delete me`**를 입력한 다음, Delete(삭제)를 선택합니다.

EC2 플릿 및 스팟 플릿

EC2 플릿 및 스팟 플릿은 단일 작업으로 수십, 수백 또는 수천 개의 Amazon EC2 인스턴스를 효과적으로 시작할 수 있도록 설계되었습니다. 플릿의 각 인스턴스는 시작 템플릿 또는 시작 시 수동으로 구성한 시작 파라미터 세트에 따라 구성됩니다.

주제

- 기능 및 이점
- 어떤 플릿 방법을 사용하는 것이 최선입니까?
- EC2 플릿 또는 스팟 플릿에 대한 구성 옵션
- EC2 플릿 사용
- 스팟 플릿 사용
- EC2 플릿 또는 스팟 플릿 모니터링
- EC2 플릿에 대한 튜토리얼
- EC2 플릿에 대한 CLI 구성 예제
- 예제 CLI 구성 스팟 플릿
- EC2 플릿 및 스팟 플릿 할당량

기능 및 이점

플릿은 다음 기능과 이점을 제공하고, 이를 통해 여러 EC2 인스턴스에서 애플리케이션을 실행할 때 비용 절감을 극대화하고 가용성과 성능을 최적화할 수 있습니다.

다양한 인스턴스 유형

> 플릿은 단일 인스턴스 유형의 가용성에 의존하지 않도록 여러 인스턴스 유형을 시작할 수 있습니다. 이렇게 하면 플릿 내 인스턴스의 전체 가용성이 증가합니다.

여러 가용 영역에서 인스턴스 분산

> 플릿은 고가용성을 위해 인스턴스를 여러 가용 영역에서 균등하게 자동으로 분산하려고 시도합니다. 이를 통해 가용 영역을 사용할 수 없는 경우 복원성을 지원합니다.

다양한 구매 옵션

플릿은 여러 구매 옵션(스팟 및 온디맨드 인스턴스)을 시작할 수 있으므로 스팟 인스턴스 사용을 통해 비용을 최적화할 수 있습니다. 플릿의 온디맨드 인스턴스와 함께 사용하여 예약 인스턴스 및 절감형 플랜의 할인 혜택을 받을 수도 있습니다.

스팟 인스턴스의 자동 교체

플릿에 스팟 인스턴스가 포함된 경우, 스팟 인스턴스가 중단되면 스팟 용량을 자동으로 교체하도록 요청할 수 있습니다. 또한 플릿은 용량 리밸런싱을 통해 중단될 위험이 높은 스팟 인스턴스를 모니터링하고 사전에 교체할 수 있습니다.

온디맨드 용량 예약

플릿은 온디맨드 용량 예약을 사용하여 온디맨드 용량을 예약할 수 있습니다. 플릿에 ML용 용량 블록이 포함되어 향후에 GPU 인스턴스를 예약하여 단기간의 ML(기계 학습) 워크로드를 지원할 수 있습니다.

어떤 플릿 방법을 사용하는 것이 최선입니까?

일반적으로 Amazon EC2 Auto Scaling을 사용하여 스팟 및 온디맨드 인스턴스 플릿을 시작하는 것이 좋습니다. 플릿을 관리하는 데 사용할 수 있는 추가 기능을 제공하기 때문입니다. 추가 기능 목록에는 스팟 및 온디맨드 인스턴스 모두에 대한 자동 상태 확인 교체, 애플리케이션 기반 상태 확인, 애플리케이션 트래픽을 정상 인스턴스로 균등하게 분산하기 위한 Elastic Load Balancing과의 통합이 포함됩니다. Amazon ECS, Amazon EKS(자체 관리형 노드 그룹) 및 Amazon VPC Lattice와 같은 AWS 서비스를 사용할 때도 Auto Scaling을 사용할 수 있습니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서를 참조하세요.

Amazon EC2 Auto Scaling을 사용할 수 없는 경우 EC2 플릿 또는 스팟 플릿 사용을 고려해 볼 수 있습니다. EC2 플릿과 스팟 플릿은 동일한 핵심 기능을 제공합니다. 하지만 EC2 플릿은 명령줄을 통해서만 사용할 수 있으며 콘솔 지원은 제공하지 않습니다. 스팟 플릿은 콘솔 지원을 제공하지만 계획된 투자가 없는 레거시 API를 기반으로 합니다.

다음 표를 사용하여 사용할 플릿 방법을 결정하세요.

플릿 방법	언제 사용해야 하나요?	사용 사례
Amazon EC2 Auto Scaling	• 단일 구성 또는 혼합 구성을 사용하는 여러 인스턴스가 필요합니다.	원하는 인스턴스 수를 유지하면서 인스턴스의 수명 주기를 관리하는 Auto Scaling 그룹을 생

플릿 방법	언제 사용해야 하나요?	사용 사례
	• 인스턴스의 수명 주기 관리를 자동화하려고 합니다.	성합니다. 지정된 최소 및 최대 한도 사이의 수평적 크기 조정(인스턴스 추가)을 지원합니다.
EC2 플릿	• 단일 구성 또는 혼합 구성을 사용하는 여러 인스턴스가 필요합니다. • 인스턴스 수명 주기를 자체 관리하려고 합니다. • Auto Scaling이 필요하지 않은 경우 `instant` 유형 EC2 플릿을 사용하는 것을 권장합니다.	인스턴스 유형, AMI, 가용 영역 또는 서브넷에 따라 바뀌는 여러 출시 사양을 사용하여 단일 작업으로 온디맨드 인스턴스와 스팟 인스턴스의 `instant` 플릿을 생성합니다. 스팟 인스턴스 할당 전략의 기본값은 단위당 `lowest-price` 이지만, `price-capacity-optimized` 로 변경하는 것이 좋습니다.
스팟 플릿	• 계획된 투자가 없는 레거시 API이므로 스팟 플릿을 사용하지 않는 것이 좋습니다. • 인스턴스 수명 주기를 관리하려면 EC2 플릿을 사용하세요. • 인스턴스 수명 주기를 관리하지 않으려면 Auto Scaling 그룹을 사용하세요.	EC2 플릿을 사용할 경우의 사용 사례에 대한 콘솔 지원이 필요한 경우에만 스팟 플릿을 사용하세요.

EC2 플릿 또는 스팟 플릿에 대한 구성 옵션

EC2 플릿 또는 스팟 플릿을 계획할 때 플릿 구성 방법 결정 시 다음 옵션을 고려하는 것이 좋습니다.

구성 옵션	질문	설명서
플릿 요청 유형	원하는 목표 용량을 위한 일회성 요청을 제출하는 플릿을 생성합니까? 아니면 시간 경과에 따라 목표 용량을 유지하는 플릿을 생성합니까?	EC2 플릿 및 스팟 플릿 요청 유형
스팟 인스턴스	플릿에 스팟 인스턴스를 포함할 계획입니까? 가능한 최저 가격으로 인스턴스를 프로비저닝할 수 있도록 플릿을 계획할 때 스팟 모범 사례를 검토하고 사용하세요.	Amazon EC2 스팟 모범 사례
플릿의 지출 한도	시간당 플릿에 지불할 요금을 제한하고 싶으신가요?	EC2 플릿 또는 스팟 플릿에 지출 한도 설정
인스턴스 유형 및 속성 기반 인스턴스 유형 선택	플릿의 인스턴스 유형을 지정하시겠습니까? 아니면 Amazon EC2에서 애플리케이션 요구 사항을 충족하는 인스턴스 유형을 선택하도록 하시겠습니까?	EC2 플릿 또는 스팟 플릿의 인스턴스 유형 선택에 대한 속성 지정
인스턴스 가중치 부여	Amazon EC2가 원하는 목표 용량을 충족하기 위해 사용 가능한 인스턴스 유형 조합을 선택할 수 있도록 각 인스턴스 유형에 컴퓨팅 용량과 성능에 해당하는 가중치를 할당하시겠습니까?	인스턴스 가중치를 사용하여 EC2 플릿 또는 스팟 플릿의 비용 및 성능 관리
할당 전략	플릿의 스팟 인스턴스와 온디맨드 인스턴스에 사용할 가용 용량, 가격 또는 인스턴스 유형을 최적화할지 여부를 결정하시겠습니까?	할당 전략을 사용하여 EC2 플릿 또는 스팟 플릿이 스팟 및 온디맨드 용량을 충족하는 방법 결정
용량 재조정	플릿에서 위험에 처한 스팟 인스턴스를 자동으로 교체하기를 원하십니까?	EC2 플릿 및 스팟 플릿의 용량 리밸런싱을 사용하여 위험이 있는 스팟 인스턴스 교체
온디맨드 용량 예약	플릿의 온디맨드 인스턴스 용량을 예약하시겠습니까?	용량 예약을 사용하여 EC2 플릿의 온디맨드 용량 예약

EC2 플릿 및 스팟 플릿 요청 유형

EC2 플릿 또는 스팟 플릿에 대한 요청 유형에 따라 요청이 동기식 또는 비동기식인지, 원하는 목표 용량을 일회성으로 요청할지 또는 시간의 흐름에 따라 목표 용량을 유지하는 작업을 지속적으로 할지 여부를 정의합니다. 플릿을 구성할 때 요청 유형을 지정해야 합니다.

EC2 플릿 및 스팟 플릿 모두 두 가지 요청 유형(`request` 및`maintain`)을 제공합니다. 추가로 EC2 플릿은 `instant`라는 세 번째 요청 유형을 제공합니다.

플릿 요청 유형

`instant`(EC2 플릿만 해당)

요청 유형을 `instant`로 구성하면 EC2 집합이 원하는 용량을 얻기 위한 동기식 일회성 요청을 합니다. API 응답에서 시작된 인스턴스가 반환되고 시작할 수 없는 인스턴스에 대한 오류가 제공됩니다. 자세한 내용은 instant 유형의 EC2 플릿 구성 단원을 참조하십시오.

`request`

요청 유형을 `request`로 구성하면 플릿이 원하는 용량을 얻기 위한 비동기식 일회성 요청을 합니다. 스팟 중단으로 인해 용량이 감소할 경우 플릿은 스팟 인스턴스를 보충하려고 하지 않으며 용량을 사용할 수 없는 경우 대체 스팟 용량에서 요청을 제출하지 않습니다. 콘솔을 사용하여 `request` 유형의 스팟 플릿을 생성하는 경우 목표 용량 유지 확인란을 선택 취소합니다.

`maintain`(기본값)

요청 유형을 `maintain`으로 구성하면 플릿은 원하는 용량을 얻기 위한 비동기식 요청을 하고 중단된 모든 스팟 인스턴스를 자동으로 보충해 용량을 유지합니다. 콘솔을 사용하여 `maintain` 유형의 스팟 플릿을 생성하는 경우 목표 용량 유지 확인란을 선택합니다.

instant 유형의 EC2 플릿 구성

인스턴트 유형의 EC2 플릿은 원하는 용량을 시작하는 시도를 단 한 번만 하는 동기식 일회성 요청입니다. API 응답에는 시작할 수 없는 인스턴스에 대한 오류와 함께 시작된 인스턴스가 나열됩니다. 인스턴스 유형의 EC2 플릿을 사용하면 몇 가지 이점이 있으며, 이 문서에서는 그러한 이점을 설명합니다. 예제 구성은 문서 끝 부분에 나와 있습니다.

EC2 인스턴스를 시작하기 위해 시작 전용 API가 필요한 워크로드의 경우 RunInstances API를 사용할 수 있습니다. 하지만 RunInstances를 사용할 경우 온디맨드 인스턴스 또는 스팟 인스턴스만 시작할 수 있으며 동일한 요청으로 둘 다 시작할 수는 없습니다. 또한 RunInstances를 사용하여 스팟 인스턴스를

시작할 경우 하나의 인스턴스 유형과 하나의 가용 영역으로 스팟 인스턴스 요청이 제한됩니다. 이는 단일 스팟 용량 풀(인스턴스 유형과 가용 영역이 동일한 미사용 인스턴스의 집합)을 대상으로 합니다. 스팟 용량 풀에 요청을 처리하기에 충분한 스팟 인스턴스 용량이 없는 경우 RunInstances 호출이 실패합니다.

RunInstances를 사용하여 스팟 인스턴스를 시작하는 대신, `type` 파라미터를 `instant`로 설정하여 CreateFleet API를 사용하면 다음과 같은 이점을 얻을 수 있습니다.

- 한 번의 요청으로 온디맨드 인스턴스와 스팟 인스턴스를 모두 시작합니다. EC2 플릿은 온디맨드 인스턴스, 스팟 인스턴스 또는 둘 모두를 시작할 수 있습니다. 스팟 인스턴스에 대한 요청은 요청의 시간당 최대 가격이 현재 스팟 가격을 초과하고 사용 가능한 용량이 있으면 수행됩니다.
- 스팟 인스턴스의 가용성이 높아집니다. `instant` 유형의 EC2 플릿을 사용하여 스팟 모범 사례에 따라 스팟 인스턴스를 시작하면 다음과 같은 이점이 있습니다.
  - 스팟 모범 사례: 인스턴스 유형 및 가용 영역에 대한 유연성 유지.

    이점: 여러 인스턴스 유형 및 가용 영역을 지정하면 스팟 용량 풀 수가 늘어납니다. 이렇게 하면 스팟 서비스에서 원하는 스팟 컴퓨팅 용량을 찾고 할당하게 될 가능성이 더 높아집니다. 일반적으로 각 워크로드에 대해 최소 10개의 인스턴스 유형을 유연하게 선택할 수 있어야 하고, 모든 가용 영역이 VPC에 사용하도록 구성되어야 합니다.
  - 스팟 모범 사례: price-capacity-optimized할당 전략 사용

    이점: `price-capacity-optimized` 할당 전략은 가장 적합한 스팟 용량 풀에서 인스턴스를 식별한 후 이 풀 중에서 가격이 가장 낮은 것에서 인스턴스를 자동으로 프로비저닝합니다. 스팟 인스턴스 용량이 최적의 용량을 가진 풀에서 소싱되기 때문에 Amazon EC2에서 해당 용량을 다시 필요로 하여 스팟 인스턴스가 중지될 가능성이 줄어듭니다.
- 더욱 다양한 기능 세트에 액세스할 수 있습니다. 시작 전용 API가 필요한 워크로드의 경우나 EC2 플릿이 인스턴스를 관리하도록 하지 않고 인스턴스의 수명 주기를 직접 관리하려는 경우, RunInstances API 대신 `instant` 유형의 EC2 플릿을 사용합니다. 다음 예에서 보듯이 EC2 플릿은 RunInstances보다 다양한 기능 세트를 제공합니다. 다른 모든 워크로드의 경우, ELB 지원 애플리케이션, 컨테이너화된 워크로드, 대기열 처리 작업 등 다양한 워크로드를 위한 보다 포괄적인 기능 세트를 제공하는 Amazon EC2 Auto Scaling을 사용해야 합니다.

유형이 인스턴트인 EC2 플릿을 사용하여 인스턴스를 용량 블록으로 내보낼 수 있습니다. 자세한 내용은 튜토리얼: 인스턴스를 용량 블록으로 시작하도록 EC2 플릿을 구성 섹션을 참조하세요.

Amazon EC2 Auto Scaling 및 Amazon EMR과 같은 AWS 서비스는 인스턴트 유형의 EC2 플릿을 사용하여 EC2 인스턴스를 시작합니다.

인스턴트 유형 EC2 플릿의 사전 조건

EC2 플릿 생성을 위한 사전 조건은 EC2 집합 사전 조건 섹션을 참조하세요.

인스턴트 EC2 플릿의 작동 방식

`instant` 유형의 EC2 플릿으로 작업할 때 발생하는 이벤트 순서는 다음과 같습니다.

1. 구성: CreateFleet 요청 유형을 `instant`로 구성합니다. 자세한 내용은 EC2 집합 생성 섹션을 참조하세요. API 호출을 한 후에는 수정할 수 없습니다.
2. 요청: API 직접 호출 시 Amazon EC2가 원하는 용량을 얻기 위한 동기식 일회성 요청을 합니다.
3. 응답: API 응답에는 시작할 수 없는 인스턴스에 대한 오류와 함께 시작된 인스턴스가 나열됩니다.
4. 설명: EC2 플릿을 설명하고, EC2 플릿과 연결된 인스턴스를 나열하고, EC2 플릿의 기록을 확인할 수 있습니다.
5. 인스턴스 종료: 언제든지 인스턴스를 종료할 수 있습니다.
6. 플릿 요청 삭제: 플릿 요청은 수동 또는 자동으로 삭제할 수 있습니다.
   - 수동: 인스턴스가 시작된 후 플릿 요청을 삭제할 수 있습니다.

     참고: 삭제된 `instant` 플릿에서 인스턴스를 실행하는 것은 지원되지 않습니다. `instant` 플릿을 삭제하면 Amazon EC2가 모든 인스턴스를 자동으로 종료합니다. 인스턴스가 1,000개 이상인 플릿의 경우 삭제 요청이 실패할 수 있습니다. 플릿에 인스턴스가 1,000개 이상인 경우 먼저 인스턴스를 대부분 수동으로 종료하여 1,000개 이하를 남깁니다. 그런 다음 플릿을 삭제하면 나머지 인스턴스가 자동으로 종료됩니다.
   - 자동: Amazon EC2는 다음 중 하나 이후에 플릿 요청을 삭제합니다.
     - 모든 인스턴스가 종료됩니다.
     - 플릿이 인스턴스를 시작하지 못합니다.

예시

다음 예에서는 다양한 사용 사례에서 `instant` 유형의 EC2 플릿을 사용하는 방법을 보여줍니다. EC2 CreateFleet API 파라미터 사용에 대한 자세한 내용은 Amazon EC2 API 참조에서 CreateFleet을 참조하세요.

예시

- 예 1: 용량 최적화 할당 전략을 사용하여 여러 스팟 인스턴스 시작
- 예 2: 용량 최적화 할당 전략을 사용하여 단일 스팟 인스턴스 시작
- 예 3: 인스턴스 가중치를 사용하여 스팟 인스턴스 시작

- 예제 4: 단일 가용 영역 내에서 스팟 인스턴스 시작
- 예 5: 단일 가용 영역 내에서 단일 인스턴스 유형의 스팟 인스턴스 시작
- 예 6: 최소 목표 용량을 시작할 수 있는 경우에만 스팟 인스턴스 시작
- 예 7: 단일 가용 영역에서 동일한 인스턴스 유형으로 최소 목표 용량을 시작할 수 있는 경우에만 스팟 인스턴스 시작
- 예 8: 여러 시작 템플릿을 사용하여 인스턴스 시작
- 예 9: 기본 온디맨드 인스턴스를 사용하여 스팟 인스턴스 시작
- 예 10: 용량 예약 및 우선 순위별 할당 전략을 사용한 온디맨드 인스턴스를 기반으로 용량 최적화 할당 전략을 사용하여 스팟 인스턴스 시작
- 예 11: capacity-optimized-prioritized 할당 전략을 사용하여 스팟 인스턴스 시작
- 예제 12: AMI ID 대신 Systems Manager 파라미터 지정

예 1: 용량 최적화 할당 전략을 사용하여 여러 스팟 인스턴스 시작

다음 예에서는 `instant` 유형의 EC2 플릿에 필요한 파라미터, 즉 시작 템플릿, 목표 용량, 기본 구매 옵션, 시작 템플릿 재정의를 지정합니다.

- 시작 템플릿은 시작 템플릿 이름과 버전 번호로 식별됩니다.
- 12개의 시작 템플릿 재정의는 각각 별도의 가용 영역에 있는 4개의 서로 다른 인스턴스 유형과 3개의 서브넷을 지정합니다. 각 인스턴스 유형 및 서브넷 조합은 스팟 용량 풀을 정의하여 12개의 스팟 용량 풀을 생성합니다.
- 플릿의 목표 용량은 인스턴스 20개입니다.
- 기본 구매 옵션은 `spot`으로, 이 옵션을 사용하면 플릿은 시작하는 인스턴스의 수에 대한 최적의 용량을 가진 스팟 용량 풀로 스팟 인스턴스 20개를 시작하려고 시도합니다.

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized"
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
```

```
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5.large",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-e7188bab"
            },
```

```
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-49e41922"
            }
         ]
      }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

예 2: 용량 최적화 할당 전략을 사용하여 단일 스팟 인스턴스 시작

TotalTargetCapacity를 1로 설정하여 instant 유형의 여러 EC2 플릿에 대해 API 호출을 수행하면 한 번에 하나의 스팟 인스턴스를 최적의 용량으로 시작할 수 있습니다.

다음 예에서는 인스턴트 유형의 EC2 플릿에 필요한 파라미터, 즉 시작 템플릿, 목표 용량, 기본 구매 옵션 및 시작 템플릿 재정의를 지정합니다. 시작 템플릿은 시작 템플릿 이름과 버전 번호로 식별됩니다. 12개의 시작 템플릿 재정의는 각각 별도의 가용 영역에 있는 4개의 서로 다른 인스턴스 유형과 3개의 서브넷을 사용합니다. 플릿의 목표 용량이 인스턴스 1개이고 기본 구매 옵션은 스팟입니다. 이 경우 플릿이 용량 최적화 할당 전략에 따라 12개의 스팟 용량 풀 중 하나에서 스팟 인스턴스를 시작함으로써 최적의 용량 풀에서 스팟 인스턴스를 시작하려고 시도합니다.

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized"
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
```

```
            "InstanceType":"c5.large",
            "SubnetId":"subnet-e7188bab"
         },
         {
            "InstanceType":"c5.large",
            "SubnetId":"subnet-49e41922"
         },
         {
            "InstanceType":"c5d.large",
            "SubnetId":"subnet-fae8c380"
         },
         {
            "InstanceType":"c5d.large",
            "SubnetId":"subnet-e7188bab"
         },
         {
            "InstanceType":"c5d.large",
            "SubnetId":"subnet-49e41922"
         },
         {
            "InstanceType":"m5.large",
            "SubnetId":"subnet-fae8c380"
         },
         {
            "InstanceType":"m5.large",
            "SubnetId":"subnet-e7188bab"
         },
         {
            "InstanceType":"m5.large",
            "SubnetId":"subnet-49e41922"
         },
         {
            "InstanceType":"m5d.large",
            "SubnetId":"subnet-fae8c380"
         },
         {
            "InstanceType":"m5d.large",
            "SubnetId":"subnet-e7188bab"
         },
         {
            "InstanceType":"m5d.large",
            "SubnetId":"subnet-49e41922"
         }
      ]
```

```
      }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 1,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

예 3: 인스턴스 가중치를 사용하여 스팟 인스턴스 시작

다음 예제에서는 인스턴스 가중치를 사용하는데, 이는 곧 인스턴스 시간이 기준이 아니라 단위 시간을 기준으로 한 가격이라는 의미입니다. 각 시작 구성에는 한 단위의 워크로드에 15GB의 메모리와 4개의 vCPU가 필요하다고 가정할 때 인스턴스에서 실행할 수 있는 워크로드 단위 수에 따라 서로 다른 인스턴스 유형과 가중치가 나열됩니다. 예를 들어 m5.xlarge(vCPU 4개와 16GB 메모리)는 한 단위를 실행할 수 있으며 가중치가 1이고, m5.2xlarge(vCPU 8개와 32GB 메모리)는 두 단위를 실행할 수 있으며 가중치가 2인 식입니다. 총 목표 용량은 40단위로 설정됩니다. 기본 구매 옵션은 스팟이며 할당 전략은 용량 최적화입니다. 이 경우 용량 최적화 할당 전략에 따라 40개의 m5.xlarge(40 나누기 1), 20개의 m5.2xlarge(40 나누기 2), 10개의 m5.4xlarge(40 나누기 4), 5개의 m5.8xlarge(40 나누기 8) 또는 원하는 용량까지 가중치가 적용된 인스턴스 유형의 조합이 사용됩니다.

자세한 내용은 인스턴스 가중치를 사용하여 EC2 플릿 또는 스팟 플릿의 비용 및 성능 관리 섹션을 참조하세요.

```
{
   "SpotOptions":{
      "AllocationStrategy":"capacity-optimized"
   },
   "LaunchTemplateConfigs":[
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"m5.xlarge",
               "SubnetId":"subnet-fae8c380",
               "WeightedCapacity":1
            },
            {
               "InstanceType":"m5.xlarge",
```

```
            "SubnetId":"subnet-e7188bab",
            "WeightedCapacity":1
         },
         {
            "InstanceType":"m5.xlarge",
            "SubnetId":"subnet-49e41922",
            "WeightedCapacity":1
         },
         {
            "InstanceType":"m5.2xlarge",
            "SubnetId":"subnet-fae8c380",
            "WeightedCapacity":2
         },
         {
            "InstanceType":"m5.2xlarge",
            "SubnetId":"subnet-e7188bab",
            "WeightedCapacity":2
         },
         {
            "InstanceType":"m5.2xlarge",
            "SubnetId":"subnet-49e41922",
            "WeightedCapacity":2
         },
         {
            "InstanceType":"m5.4xlarge",
            "SubnetId":"subnet-fae8c380",
            "WeightedCapacity":4
         },
         {
            "InstanceType":"m5.4xlarge",
            "SubnetId":"subnet-e7188bab",
            "WeightedCapacity":4
         },
         {
            "InstanceType":"m5.4xlarge",
            "SubnetId":"subnet-49e41922",
            "WeightedCapacity":4
         },
         {
            "InstanceType":"m5.8xlarge",
            "SubnetId":"subnet-fae8c380",
            "WeightedCapacity":8
         },
         {
```

```
               "InstanceType":"m5.8xlarge",
               "SubnetId":"subnet-e7188bab",
               "WeightedCapacity":8
            },
            {
               "InstanceType":"m5.8xlarge",
               "SubnetId":"subnet-49e41922",
               "WeightedCapacity":8
            }
         ]
      }
   ],
   "TargetCapacitySpecification":{
      "TotalTargetCapacity":40,
      "DefaultTargetCapacityType":"spot"
   },
   "Type":"instant"
}
```

예제 4: 단일 가용 영역 내에서 스팟 인스턴스 시작

스팟 옵션 SingleAvailabilityZone을 true로 설정하여 단일 가용 영역에서 모든 인스턴스를 시작하도록 플릿을 구성할 수 있습니다.

12개의 시작 템플릿 재정의는 각각 별도의 가용 영역에 있는 서로 다른 인스턴스 유형과 서브넷을 사용하지만 가중치가 적용된 용량은 동일합니다. 총 목표 용량은 인스턴스 20개이고, 기본 구매 옵션은 스팟이며, 스팟 할당 전략은 용량 최적화입니다. EC2 플릿은 시작 사양을 사용하여 최적의 용량을 가진 스팟 용량 풀에서 단일 AZ의 스팟 인스턴스 20개를 모두 시작합니다.

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized",
        "SingleAvailabilityZone": true
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.4xlarge",
```

```
            "SubnetId":"subnet-fae8c380"
         },
         {
            "InstanceType":"c5.4xlarge",
            "SubnetId":"subnet-e7188bab"
         },
         {
            "InstanceType":"c5.4xlarge",
            "SubnetId":"subnet-49e41922"
         },
         {
            "InstanceType":"c5d.4xlarge",
            "SubnetId":"subnet-fae8c380"
         },
         {
            "InstanceType":"c5d.4xlarge",
            "SubnetId":"subnet-e7188bab"
         },
         {
            "InstanceType":"c5d.4xlarge",
            "SubnetId":"subnet-49e41922"
         },
         {
            "InstanceType":"m5.4xlarge",
            "SubnetId":"subnet-fae8c380"
         },
         {
            "InstanceType":"m5.4xlarge",
            "SubnetId":"subnet-e7188bab"
         },
         {
            "InstanceType":"m5.4xlarge",
            "SubnetId":"subnet-49e41922"
         },
         {
            "InstanceType":"m5d.4xlarge",
            "SubnetId":"subnet-fae8c380"
         },
         {
            "InstanceType":"m5d.4xlarge",
            "SubnetId":"subnet-e7188bab"
         },
         {
            "InstanceType":"m5d.4xlarge",
```

```
                "SubnetId":"subnet-49e41922"
            }
         ]
      }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

예 5: 단일 가용 영역 내에서 단일 인스턴스 유형의 스팟 인스턴스 시작

SpotOptions SingleInstanceType을 true로 설정하고, SingleAvailabilityZone을 true로 설정하여 단일 가용 영역에서 동일한 인스턴스 유형의 인스턴스를 모두 시작하도록 플릿을 구성할 수 있습니다.

12개의 시작 템플릿 재정의는 각각 별도의 가용 영역에 있는 서로 다른 인스턴스 유형과 서브넷을 사용하지만 가중치가 적용된 용량은 동일합니다. 총 목표 용량은 인스턴스 20개이고, 기본 구매 옵션은 스팟이며, 스팟 할당 전략은 용량 최적화입니다. EC2 플릿은 시작 사양을 사용하여 최적의 용량을 가진 스팟 인스턴스 풀에서 단일 AZ의 인스턴스 유형이 동일한 스팟 인스턴스 20개를 시작합니다.

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized",
        "SingleInstanceType": true,
        "SingleAvailabilityZone": true
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
```

```
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5d.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5d.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5d.4xlarge",
               "SubnetId":"subnet-49e41922"
            }
         ]
      }
   ],
   "TargetCapacitySpecification": {
```

```
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

예 6: 최소 목표 용량을 시작할 수 있는 경우에만 스팟 인스턴스 시작

스팟 옵션 MinTargetCapacity를 한 번에 시작할 최소 목표 용량으로 설정함으로써 최소 목표 용량을 시작할 수 있는 경우에만 인스턴스를 시작하도록 플릿을 구성할 수 있습니다.

MinTargetCapacity를 지정할 때는 SingleInstanceType 또는 SingleAvailabilityZone 파라미터 중 하나 이상을 지정해야 합니다. 이 예제에서는 SingleInstanceType이 지정되므로 20개 인스턴스 모두 동일한 인스턴스 유형을 사용해야 합니다.

12개의 시작 템플릿 재정의는 각각 별도의 가용 영역에 있는 서로 다른 인스턴스 유형과 서브넷을 사용하지만 가중치가 적용된 용량은 동일합니다. 총 목표 용량과 최소 목표 용량이 모두 인스턴스 20개이고, 기본 구매 옵션은 스팟이며, 스팟 할당 전략은 용량 최적화입니다. EC2 플릿은 20개의 인스턴스를 동시에 시작할 수 있는 경우에만 시작 템플릿 재정의를 사용하여 최적의 용량을 가진 스팟 용량 풀에서 20개의 스팟 인스턴스를 시작합니다.

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized",
        "SingleInstanceType": true,
        "MinTargetCapacity": 20
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
```

```
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5d.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5d.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5d.4xlarge",
               "SubnetId":"subnet-49e41922"
            }
         ]
      }
   ],
   "TargetCapacitySpecification": {
```

```
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

예 7: 단일 가용 영역에서 동일한 인스턴스 유형으로 최소 목표 용량을 시작할 수 있는 경우에만 스팟 인스턴스 시작

스팟 옵션 MinTargetCapacity를 한 번에 시작할 최소 목표 용량으로 설정하고 SingleInstanceType 및 SingleAvailabilityZone 옵션을 사용함으로써 단일 가용 영역에서 단일 인스턴스 유형으로 최소 목표 용량을 시작할 수 있는 경우에만 인스턴스를 시작하도록 플릿을 구성할 수 있습니다.

시작 템플릿을 재정의하는 12개의 시작 사양은 각각 별도의 가용 영역에 있는 서로 다른 인스턴스 유형과 서브넷을 사용하지만 가중치가 적용된 용량은 동일합니다. 총 목표 용량과 최소 목표 용량이 모두 인스턴스 20개이고, 기본 구매 옵션은 스팟이며, 스팟 할당 전략은 용량 최적화이고, SingleInstanceType과 SingleAvailabilityZone이 true입니다. EC2 플릿은 20개의 인스턴스를 동시에 시작할 수 있는 경우에만 시작 사양을 사용하여 최적의 용량을 가진 스팟 용량 풀에서 단일 AZ의 인스턴스 유형이 동일한 스팟 인스턴스 20개를 시작합니다.

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized",
        "SingleInstanceType": true,
        "SingleAvailabilityZone": true,
        "MinTargetCapacity": 20
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
```

```
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5d.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5d.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5d.4xlarge",
               "SubnetId":"subnet-49e41922"
            }
         ]
      }
   ],
   "TargetCapacitySpecification": {
```

```
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

예 8: 여러 시작 템플릿을 사용하여 인스턴스 시작

여러 시작 템플릿을 지정하여 인스턴스 유형 또는 인스턴스 유형 그룹에 따라 서로 다른 시작 사양으로 인스턴스를 시작하도록 플릿을 구성할 수 있습니다. 이 예에서는 인스턴스 유형에 따라 서로 다른 EBS 볼륨 크기를 원하며, 이는 시작 템플릿 ec2-fleet-lt-4xl, ec2-fleet-lt-9xl, ec2-fleet-lt-18xl에 구성되어 있습니다.

이 예에서는 크기에 따라 3개의 인스턴스 유형에 3개의 서로 다른 시작 템플릿을 사용하고 있습니다. 모든 시작 템플릿의 시작 사양 재정의에는 인스턴스 유형의 vCPU 수에 따라 인스턴스 가중치를 사용합니다. 총 목표 용량은 144 단위이고, 기본 구매 옵션은 스팟이며, 스팟 할당 전략은 용량 최적화입니다. EC2 플릿은 용량 최적화 할당 전략에 따라 시작 템플릿 ec2-fleet-4xl을 사용하여 9개의 c5n.4xlarge(144 나누기 16)를 시작하거나, 시작 템플릿 ec2-fleet-9xl을 사용하여 4개의 c5n.9xlarge(144 나누기 36)를 시작하거나, 시작 템플릿 ec2-fleet-18xl을 사용하여 2개의 c5n.18xlarge(144 나누기 72)를 시작하거나, 원하는 용량까지 가중치가 적용된 인스턴스 유형의 조합을 시작할 수 있습니다.

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized"
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt-18xl",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5n.18xlarge",
               "SubnetId":"subnet-fae8c380",
               "WeightedCapacity":72
            },
            {
               "InstanceType":"c5n.18xlarge",
               "SubnetId":"subnet-e7188bab",
               "WeightedCapacity":72
```

```
            },
            {
               "InstanceType":"c5n.18xlarge",
               "SubnetId":"subnet-49e41922",
               "WeightedCapacity":72
            }
         ]
      },
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt-9xl",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5n.9xlarge",
               "SubnetId":"subnet-fae8c380",
               "WeightedCapacity":36
            },
            {
               "InstanceType":"c5n.9xlarge",
               "SubnetId":"subnet-e7188bab",
               "WeightedCapacity":36
            },
            {
               "InstanceType":"c5n.9xlarge",
               "SubnetId":"subnet-49e41922",
               "WeightedCapacity":36
            }
         ]
      },
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt-4xl",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5n.4xlarge",
               "SubnetId":"subnet-fae8c380",
               "WeightedCapacity":16
            },
            {
               "InstanceType":"c5n.4xlarge",
```

```
                "SubnetId":"subnet-e7188bab",
                "WeightedCapacity":16
            },
            {
                "InstanceType":"c5n.4xlarge",
                "SubnetId":"subnet-49e41922",
                "WeightedCapacity":16
            }
         ]
      }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 144,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

예 9: 기본 온디맨드 인스턴스를 사용하여 스팟 인스턴스 시작

다음 예제에서는 총 목표 용량인 인스턴스 20개를 플릿에 지정하고 목표 용량은 온디맨드 인스턴스 5개로 지정합니다. 기본 구매 옵션은 스팟입니다. 지정한 대로 플릿은 온디맨드 인스턴스 5개를 시작하지만 총 목표 용량을 충족하려면 인스턴스를 15개 더 시작해야 합니다. 차이에 대한 구매 옵션이 TotalTargetCapacity – OnDemandTargetCapacity = DefaultTargetCapacityType으로 계산되어 플릿이 용량 최적화 할당 전략에 따라 12개의 스팟 용량 풀 중 하나에서 15개의 스팟 인스턴스를 시작합니다.

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized"
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5.large",
```

```
            "SubnetId":"subnet-e7188bab"
         },
         {
            "InstanceType":"c5.large",
            "SubnetId":"subnet-49e41922"
         },
         {
            "InstanceType":"c5d.large",
            "SubnetId":"subnet-fae8c380"
         },
         {
            "InstanceType":"c5d.large",
            "SubnetId":"subnet-e7188bab"
         },
         {
            "InstanceType":"c5d.large",
            "SubnetId":"subnet-49e41922"
         },
         {
            "InstanceType":"m5.large",
            "SubnetId":"subnet-fae8c380"
         },
         {
            "InstanceType":"m5.large",
            "SubnetId":"subnet-e7188bab"
         },
         {
            "InstanceType":"m5.large",
            "SubnetId":"subnet-49e41922"
         },
         {
            "InstanceType":"m5d.large",
            "SubnetId":"subnet-fae8c380"
         },
         {
            "InstanceType":"m5d.large",
            "SubnetId":"subnet-e7188bab"
         },
         {
            "InstanceType":"m5d.large",
            "SubnetId":"subnet-49e41922"
         }
      ]
   }
```

```
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "OnDemandTargetCapacity": 5,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

예 10: 용량 예약 및 우선 순위별 할당 전략을 사용한 온디맨드 인스턴스를 기반으로 용량 최적화 할당 전략을 사용하여 스팟 인스턴스 시작

용량 예약에 대한 사용 전략을 use-capacity-reservations-first로 설정하여 기본 목표 용량 유형이 스팟인 온디맨드 인스턴스의 기반을 시작할 때 온디맨드 용량 예약부터 사용하도록 플릿을 구성할 수 있습니다. 그리고 여러 인스턴스 풀에 미사용 용량 예약이 있는 경우, 선택한 온디맨드 할당 전략이 적용됩니다. 이 예에서 온디맨드 할당 전략은 우선 순위별입니다.

이 예에서는 사용할 수 있는 미사용 용량 예약이 6개 있습니다. 이는 플릿의 목표 온디맨드 용량인 온디맨드 인스턴스 10개보다 적습니다.

이 계정은 2개의 풀에 다음과 같은 미사용 용량 예약 6개를 가지고 있습니다. 각 풀의 용량 예약 수는 AvailableInstanceCount로 표시됩니다.

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "m5.large",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 3,
    "InstanceMatchCriteria": "open",
    "State": "active"
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "c5.large",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 3,
    "InstanceMatchCriteria": "open",
    "State": "active"
}
```

다음 플릿 구성에서는 이 예제와 관련된 구성만 보여 줍니다. 온디맨드 할당 전략은 우선 순위별이고, 용량 예약의 사용 전략은 use-capacity-reservations-first입니다. 스팟 할당 전략은 용량 최적화입니다. 총 목표 용량은 20이고, 온디맨드 목표 용량은 10이며, 기본 목표 용량 유형은 스팟입니다.

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized"
    },
    "OnDemandOptions":{
       "CapacityReservationOptions": {
         "UsageStrategy": "use-capacity-reservations-first"
       },
       "AllocationStrategy":"prioritized"
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-fae8c380",
               "Priority": 1.0
            },
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-e7188bab",
               "Priority": 2.0
            },
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-49e41922",
               "Priority": 3.0
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-fae8c380",
               "Priority": 4.0
            },
            {
               "InstanceType":"c5d.large",
```

```
            "SubnetId":"subnet-e7188bab",
            "Priority": 5.0
         },
         {
            "InstanceType":"c5d.large",
            "SubnetId":"subnet-49e41922",
            "Priority": 6.0
         },
         {
            "InstanceType":"m5.large",
            "SubnetId":"subnet-fae8c380",
            "Priority": 7.0
         },
         {
            "InstanceType":"m5.large",
            "SubnetId":"subnet-e7188bab",
            "Priority": 8.0
         },
         {
            "InstanceType":"m5.large",
            "SubnetId":"subnet-49e41922",
            "Priority": 9.0
         },
         {
            "InstanceType":"m5d.large",
            "SubnetId":"subnet-fae8c380",
            "Priority": 10.0
         },
         {
            "InstanceType":"m5d.large",
            "SubnetId":"subnet-e7188bab",
            "Priority": 11.0
         },
         {
            "InstanceType":"m5d.large",
            "SubnetId":"subnet-49e41922",
            "Priority": 12.0
         }
      ]
   }
],
"TargetCapacitySpecification": {
    "TotalTargetCapacity": 20,
    "OnDemandTargetCapacity": 10,
```

```
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

이전 구성을 사용하여 인스턴트 플릿을 생성하면 목표 용량을 충족하기 위해 다음 20개의 인스턴스가 시작됩니다.

- us-east-1a의 c5.large 온디맨드 인스턴스 7개 – us-east-1a의 c5.large가 첫 번째 우선 순위이고 사용 가능한 미사용 c5.large 용량 예약은 3개입니다. 먼저 용량 예약을 사용하여 온디맨드 인스턴스 3개를 시작하고, 온디맨드 할당 전략(이 예에서는 우선 순위별)에 따라 4개의 온디맨드 인스턴스를 추가로 시작합니다.
- us-east-1a의 m5.large 온디맨드 인스턴스 3개 – us-east-1a의 m5.large가 두 번째 우선 순위이고 사용 가능한 미사용 c3.large 용량 예약은 3개입니다.
- 용량 최적화 할당 전략에 따라 최적의 용량을 가진 12개의 스팟 용량 풀 중 하나에서 10개의 스팟 인스턴스를 시작합니다.

플릿이 시작된 후, describe-capacity-reservations를 실행하여 미사용 용량 예약이 몇 개나 남아 있는지 확인할 수 있습니다. 이 예에서는 c5.large 및 m5.large 용량 예약이 모두 사용되었음을 보여주는 다음과 같은 응답이 나타납니다.

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "m5.large",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "c5.large",
    "AvailableInstanceCount": 0
}
```

예 11: capacity-optimized-prioritized 할당 전략을 사용하여 스팟 인스턴스 시작

다음 예에서는 인스턴트 유형의 EC2 플릿에 필요한 파라미터, 즉 시작 템플릿, 목표 용량, 기본 구매 옵션 및 시작 템플릿 재정의를 지정합니다. 시작 템플릿은 시작 템플릿 이름과 버전 번호로 식별됩니다. 시작 템플릿을 재정의하는 12개의 시작 사양은 각각 별도의 가용 영역에 있는 4개의 서로 다른 인스턴스 유형(우선 순위가 서로 다름)과 3개의 서브넷을 사용합니다. 플릿의 목표 용량이 20개의 인스턴스이

고 기본 구매 옵션은 스팟이므로, 플릿이 최선을 다해 우선 순위를 구현하지만 용량을 최우선으로 하는 capacity-optimized-prioritized 할당 전략에 따라 12개의 스팟 용량 풀 중 하나에서 20개의 스팟 인스턴스를 시작하려고 시도합니다.

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized-prioritized"
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-fae8c380",
               "Priority": 1.0
            },
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-e7188bab",
               "Priority": 1.0
            },
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-49e41922",
               "Priority": 1.0
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-fae8c380",
               "Priority": 2.0
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-e7188bab",
               "Priority": 2.0
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-49e41922",
```

```
                "Priority": 2.0
            },
            {
                "InstanceType":"m5.large",
                "SubnetId":"subnet-fae8c380",
                "Priority": 3.0
            },
            {
                "InstanceType":"m5.large",
                "SubnetId":"subnet-e7188bab",
                "Priority": 3.0
            },
            {
                "InstanceType":"m5.large",
                "SubnetId":"subnet-49e41922",
                "Priority": 3.0
            },
            {
                "InstanceType":"m5d.large",
                "SubnetId":"subnet-fae8c380",
                "Priority": 4.0
            },
            {
                "InstanceType":"m5d.large",
                "SubnetId":"subnet-e7188bab",
                "Priority": 4.0
            },
            {
                "InstanceType":"m5d.large",
                "SubnetId":"subnet-49e41922",
                "Priority": 4.0
            }
         ]
      }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

예제 12: AMI ID 대신 Systems Manager 파라미터 지정

다음 예제에서는 시작 템플릿을 사용하여 플릿의 인스턴스에 대한 구성을 지정합니다. 이 예제에서 `ImageId`의 경우 AMI ID를 지정하는 대신 AMI가 Systems Manager 파라미터로 참조됩니다. 인스턴스 시작 시 Systems Manager 파라미터는 AMI ID로 확인됩니다.

이 예제에서 Systems Manager 파라미터는 유효한 `resolve:ssm:golden-ami` 형식으로 지정됩니다. Systems Manager 파라미터의 경우 다른 유효한 형식도 있습니다. 자세한 내용은 AMI ID 대신 Systems Manager 파라미터 사용 섹션을 참조하세요.

> **Note**
>
> 플릿 유형은 `instant`여야 합니다. 다른 플릿 유형에서는 AMI ID 대신 Systems Manager 파라미터 지정을 지원하지 않습니다.

```
{
    "LaunchTemplateData": {
        "ImageId": "resolve:ssm:golden-ami",
        "InstanceType": "m5.4xlarge",
        "TagSpecifications": [{
            "ResourceType": "instance",
            "Tags": [{
                "Key": "Name",
                "Value": "webserver"
            }]
        }]
    }
}
```

EC2 플릿 또는 스팟 플릿에 지출 한도 설정

EC2 플릿 또는 스팟 플릿에 시간당 지출할 금액을 제한할 수 있습니다. 지출 한도에 도달하면 플릿은 목표 용량에 도달하지 않은 경우에도 인스턴스 실행을 중지합니다.

온디맨드 인스턴스와 스팟 인스턴스에는 별도의 지출 한도가 있습니다.

EC2 플릿의 온디맨드 인스턴스 및 스팟 인스턴스에 대한 지출 한도 구성

create-fleet 명령과 다음 파라미터를 사용합니다.

- 온디맨드 인스턴스의 경우: `OnDemandOptions` 구조의 `MaxTotalPrice` 필드에 지출 한도를 지정합니다.
- 스팟 인스턴스의 경우: `SpotOptions` 구조의 `MaxTotalPrice` 필드에 지출 한도를 지정합니다.

스팟 플릿의 온디맨드 인스턴스 및 스팟 인스턴스에 대한 지출 한도 구성

Amazon EC2 콘솔 또는 AWS CLI를 사용하여 지출 한도를 구성할 수 있습니다.

(콘솔) 스팟 플릿을 생성할 때 스팟 인스턴스 최대 비용 설정 확인란을 선택한 다음 최대 비용 설정(시간당)에 값을 입력합니다. 자세한 내용은 정의된 파라미터를 사용하여 스팟 플릿 요청 생성의 6.e 단계를 참조하세요.

(AWS CLI) request-spot-fleet 명령과 다음 파라미터를 사용합니다.

- 온디맨드 인스턴스의 경우: `OnDemandMaxTotalPrice` 필드에 지출 한도를 지정합니다.
- 스팟 인스턴스의 경우: `SpotMaxTotalPrice` 필드에 지출 한도를 지정합니다.

예시

다음 예제와 같이 이 작업을 두 가지 시나리오로 수행할 수 있습니다. 첫 번째 예제에서 플릿은 온디맨드 인스턴스(`OnDemandTargetCapacity`)에 설정된 목표 용량에 도달하면 온디맨드 인스턴스 시작을 중지합니다. 두 번째 예제에서 플릿은 온디맨드 인스턴스(`MaxTotalPrice`)에 대해 시간당 지불할 최대 금액에 도달하면 온디맨드 인스턴스 실행을 중지합니다.

예제: 목표 용량에 도달할 때 온디맨드 인스턴스 실행 중지

다음과 같은 `m4.large` 온디맨드 인스턴스 요청 시:

- 온디맨드 가격: 시간당 0.10 USD
- `OnDemandTargetCapacity`: 10
- `MaxTotalPrice`: 1.50 USD

플릿은 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.

예제: 최대 총 가격에 도달할 때 온디맨드 인스턴스 실행 중지

다음과 같은 `m4.large` 온디맨드 인스턴스 요청 시:

- 온디맨드 가격: 시간당 0.10 USD
- `OnDemandTargetCapacity`: 10
- `MaxTotalPrice`: 0.80 USD

플릿이 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 플릿은 8개의 온디맨드 인스턴스(온디맨드 대상 용량 미만)만 실행합니다. 더 많이 실행하면 온디맨드 인스턴스의 `MaxTotalPrice`를 초과할 수 있기 때문입니다.

성능 순간 확장 가능 인스턴스

버스트 가능 성능 인스턴스 유형을 사용하여 스팟 인스턴스를 시작하고 CPU 크레딧 발생에 대한 유휴 시간 없이 즉시 짧은 기간 동안 버스트 가능 성능 스팟 인스턴스를 사용할 계획인 경우 표준 모드로 시작하여 높은 비용 지불을 방지하는 것이 좋습니다. 버스팅 가능 성능 스팟 인스턴스를 무제한 모드로 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.

무제한 모드는 버스팅에 대한 CPU 크레딧이 발생할 정도로 인스턴스 실행이 긴 경우에만 버스팅 가능 성능 스팟 인스턴스에 적합합니다. 그렇게 하지 않으면 잉여 크레딧 비용을 지불하면 버스트 가능 성능 스팟 인스턴스가 다른 인스턴스를 사용하는 것보다 비용이 많이 듭니다. 자세한 내용은 무제한 모드 대 고정 CPU 사용 시기 섹션을 참조하세요.

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 스팟 인스턴스에 무제한 모드를 사용하거나 전용 CPU를 포함한 인스턴스 유형을 사용할 수 있습니다.

EC2 플릿 또는 스팟 플릿의 인스턴스 유형 선택에 대한 속성 지정

EC2 플릿 또는 스팟 플릿을 생성할 때 플릿에서 온디맨드 인스턴스 및 스팟 인스턴스를 구성하기 위해 하나 이상의 인스턴스 유형을 지정해야 합니다. 인스턴스 유형을 수동으로 지정하는 작업 대신 인스턴스에 있어야 하는 속성을 지정하면 Amazon EC2는 해당 속성으로 모든 인스턴스 유형을 식별합니다. 이를 속성 기반 인스턴스 유형 선택이라고 합니다. 예를 들어 인스턴스에 필요한 최소 및 최대 vCPU 수를 지정할 수 있으며, 플릿은 해당 vCPU 요구 사항을 충족하는 사용 가능한 인스턴스 유형을 사용하여 인스턴스를 시작합니다.

속성 기반 인스턴스 유형 선택은 컨테이너 또는 웹 플릿 실행, 빅 데이터 처리, 지속적 통합 및 배포(CI/CD) 도구 구현 등 사용할 인스턴스 유형을 유연하게 처리하는 워크로드 및 프레임워크에 이상적입니다.

장점

속성 기반 인스턴스 유형을 선택하면 다음과 같은 이점이 있습니다.

- 쉽게 올바른 인스턴스 유형 사용 - 사용 가능한 인스턴스 유형이 너무 많기 때문에 워크로드에 적합한 인스턴스 유형을 찾는 데 시간이 많이 걸릴 수 있습니다. 인스턴스 속성을 지정하면 인스턴스 유형에는 워크로드에 필요한 속성이 자동으로 포함됩니다.
- 단순화된 구성 - 플릿에 대해 여러 인스턴스 유형을 수동으로 지정하려면 각 인스턴스 유형에 대해 별도의 시작 템플릿 재정의를 생성해야 합니다. 그러나 속성 기반 인스턴스 유형을 선택할 경우 여러 인스턴스 유형을 제공하려면 시작 템플릿 또는 시작 템플릿 재정의에서 인스턴스 속성만 지정하면 됩니다.
- 새 인스턴스 유형의 자동 사용 - 인스턴스 유형이 아닌 인스턴스 속성을 지정하면 플릿이 릴리스될 때 새로운 세대의 인스턴스 유형을 사용하여 플릿의 구성을 '나중에 교정'할 수 있습니다.
- 인스턴스 유형 유연성 - 인스턴스 유형이 아닌 인스턴스 속성을 지정하면 플릿은 스팟 인스턴스를 시작하기 위해 다양한 인스턴스 유형 중에서 선택할 수 있으며, 이때 인스턴스 유형 유연성에 대한 스팟 모범 사례를 따릅니다.

주제

- 속성 기반 인스턴스 유형 선택 작동 방법
- 가격 보호
- 성능 보호
- 고려 사항
- 속성 기반 인스턴스 유형 선택으로 EC2 플릿 생성
- 속성 기반 인스턴스 유형 선택으로 스팟 플릿 생성
- 유효한 EC2 플릿 구성과 유효하지 않은 구성 예제
- 유효한 스팟 플릿 구성과 유효하지 않은 구성 예제
- 지정한 속성을 가진 인스턴스 유형 미리 보기

속성 기반 인스턴스 유형 선택 작동 방법

플릿 구성에서 속성 기반 인스턴스 유형 선택을 사용하려면 인스턴스 유형 목록을 인스턴스에 필요한 인스턴스 속성 목록으로 바꿉니다. EC2 플릿 또는 스팟 플릿은 지정된 인스턴스 속성을 가진 사용 가능한 인스턴스 유형에서 인스턴스를 시작합니다.

주제

- 인스턴스 속성 유형
- 속성 기반 인스턴스 유형 선택을 구성하는 곳
- 플릿을 프로비저닝할 때 EC2 플릿 또는 스팟 플릿이 속성 기반 인스턴스 유형 선택을 사용하는 방법

인스턴스 속성 유형

컴퓨팅 요구 사항을 표현하기 위해 지정할 수 있는 다음과 같은 몇 가지 인스턴스 속성이 있습니다.

- vCPU 수 – 인스턴스당 최소 및 최대 vCPU 수입니다.
- 메모리 - 인스턴스당 최소 및 최대 메모리 GiB입니다.
- 로컬 스토리지 - 로컬 스토리지에 EBS를 사용할지 아니면 인스턴스 스토어 볼륨을 사용할지입니다.
- 성능 버스트 기능 – T4g, T3a, T3 및 T2 유형을 포함한 T 인스턴스 패밀리를 사용할지 여부입니다.

각 속성 및 기본값에 대한 설명은 Amazon EC2 API Reference(Amazon EC2 API 레퍼런스)의 InstanceRequirements를 참조하세요.

속성 기반 인스턴스 유형 선택을 구성하는 곳

콘솔을 사용하는지 아니면 AWS CLI를 사용하는지에 따라 속성 기반 인스턴스 유형 선택에 대한 인스턴스 속성을 다음과 같이 지정할 수 있습니다.

콘솔에서 다음 플릿 구성 요소에 있는 인스턴스 속성을 지정할 수 있습니다.

- 시작 템플릿 및 플릿 요청의 시작 템플릿 참조에서
- (스팟 플릿만 해당) 플릿 요청에서

AWS CLI에서 다음 플릿 구성 요소 중 하나 또는 모두에 인스턴스 속성을 지정할 수 있습니다.

- 시작 템플릿 및 플릿 요청의 시작 템플릿 참조에서
- 시작 템플릿 재정의에서

다른 AMI를 사용하는 인스턴스를 혼합하려면 여러 시작 템플릿 재정의에서 인스턴스 속성을 지정할 수 있습니다. 예를 들어 다른 인스턴스 유형은 x86 및 ARM 기반 프로세서를 사용할 수 있습니다.

- (스팟 플릿만 해당) 시작 사양에서

플릿을 프로비저닝할 때 EC2 플릿 또는 스팟 플릿이 속성 기반 인스턴스 유형 선택을 사용하는 방법

EC2 플릿 또는 스팟 플릿은 다음과 같은 방식으로 플릿을 프로비저닝합니다.

- 지정한 속성을 가진 인스턴스 유형을 식별합니다.
- 가격 보호를 사용하여 제외할 인스턴스 유형을 결정합니다.
- 인스턴스 유형이 일치하는 AWS 리전 또는 가용 영역을 기반으로 인스턴스 시작을 고려할 용량 풀을 결정합니다.
- 지정된 할당 전략을 적용하여 인스턴스를 시작할 용량 풀을 결정합니다.

  속성 기반 인스턴스 유형 선택은 플릿을 프로비저닝할 용량 풀을 선택하지 않습니다. 이는 할당 전략의 작업입니다.

  할당 전략을 지정하면 플릿은 지정된 할당 전략에 따라 인스턴스를 시작합니다.
  - 스팟 인스턴스의 경우 속성 기반 인스턴스 유형 선택은 가격 용량 최적화, 용량 최적화 및 최저 가격을 지원합니다. 최저 가격 스팟 할당 전략은 스팟 인스턴스의 중단 위험이 가장 높기 때문에 권장하지 않습니다.
  - 온디맨드 인스턴스의 경우 속성 기반 인스턴스 유형 선택은 최저 가격 할당 전략을 지원합니다.
- 지정된 인스턴스 속성을 가진 인스턴스 유형에 대한 용량이 없으면 인스턴스를 시작할 수 없으며 플릿이 오류를 반환합니다.

가격 보호

가격 보호는 EC2 플릿 또는 스팟 플릿이 사용자가 지정한 속성에 적합하더라도 너무 비싼 인스턴스 유형을 사용하지 못하도록 방지하는 기능입니다. 가격 보호를 사용하려면 기준 금액을 설정합니다. 그런 다음, Amazon EC2가 해당 속성이 있는 인스턴스 유형을 선택하면 임곗값 이상의 가격이 책정된 인스턴스 유형을 제외합니다.

Amazon EC2가 기준 금액을 계산하는 방법은 다음과 같습니다.

- Amazon EC2는 먼저 속성과 일치하는 인스턴스 유형 중에서 가장 저렴한 인스턴스 유형을 식별합니다.

- 그러면 Amazon EC2는 가격 보호 파라미터에 지정한 값(백분율로 표시)을 가져와 식별된 인스턴스 유형의 가격과 곱합니다. 결과는 기준 금액으로 사용되는 가격입니다.

온디맨드 인스턴스와 스팟 인스턴스에는 별도의 기준 금액이 있습니다.

속성 기반 인스턴스 유형 선택으로 플릿을 생성하면 기본적으로 가격 보호가 사용됩니다. 기본값을 유지할 수도 있고 직접 지정할 수도 있습니다.

가격 보호를 끌 수도 있습니다. 가격 보호 기준이 없음을 나타내려면 999999와 같은 높은 백분율 값을 지정합니다.

주제

- 최저 가격의 인스턴스 유형을 식별하는 방법
- 온디맨드 인스턴스 가격 보호
- 스팟 인스턴스 가격 보호
- 가격 보호 임곗값 지정

최저 가격의 인스턴스 유형을 식별하는 방법

Amazon EC2는 지정된 속성과 일치하는 인스턴스 유형 중 가격이 가장 낮은 인스턴스 유형을 식별하여 기준 금액의 기준이 될 가격을 결정합니다. 이는 다음과 같은 방식으로 수행됩니다.

- 먼저 속성과 일치하는 현재 세대 C, M 또는 R 인스턴스 유형을 살펴봅니다. 일치하는 항목이 발견되면 가격이 가장 낮은 인스턴스 유형을 식별합니다.
- 일치하는 항목이 없으면 속성과 일치하는 현재 세대 인스턴스 유형을 모두 찾습니다. 일치하는 항목이 발견되면 가격이 가장 낮은 인스턴스 유형을 식별합니다.
- 일치하는 항목이 없으면 속성과 일치하는 이전 세대 인스턴스 유형을 찾아 가격이 가장 낮은 인스턴스 유형을 식별합니다.

온디맨드 인스턴스 가격 보호

온디맨드 인스턴스 유형의 가격 보호 기준은 식별된 가격이 가장 낮은 온디맨드 인스턴스 유형(`OnDemandMaxPricePercentageOverLowestPrice`)보다 높은 백분율로 계산됩니다. 지불할 의사가 있는 비율을 더 높게 지정합니다. 이 파라미터를 지정하지 않으면 식별된 가격보다 20% 더 높은 가격 보호 기준을 계산하는 데 기본값 `20`이 사용됩니다.

예를 들어 식별된 온디맨드 인스턴스 가격이 0.4271이고 25를 지정하는 경우 가격 기준 금액은 0.4271보다 25% 높습니다. 이는 0.4271 * 1.25 = 0.533875로 계산됩니다. 계산된 가격은 온디맨드 인스턴스에 대해 지불할 의사가 있는 최대 가격이며, 이 예제에서 Amazon EC2는 비용이 0.533875를 초과하는 모든 온디맨드 인스턴스 유형을 제외합니다.

스팟 인스턴스 가격 보호

기본적으로 Amazon EC2는 다양한 인스턴스 유형 중에서 일관되게 선택할 수 있도록 최적의 스팟 인스턴스 가격 보호를 자동으로 적용합니다. 가격 보호를 직접 수동으로 설정할 수도 있습니다. 하지만 Amazon EC2가 자동으로 수행하면 스팟 용량이 충족될 가능성을 높일 수 있습니다.

다음 옵션 중 하나를 사용하여 가격 보호를 수동으로 지정할 수 있습니다. 가격 보호를 수동으로 설정하는 경우 첫 번째 옵션을 사용하는 것이 좋습니다.

- 식별된 최저 가격의 온디맨드 인스턴스 유형 비율[MaxSpotPriceAsPercentageOfOptimalOnDemandPrice]

  예를 들어 식별된 온디맨드 인스턴스 가격이 0.4271이고 60을 지정하는 경우 가격 기준 금액은 0.4271의 60%입니다. 이는 0.4271 * 0.60 = 0.25626으로 계산됩니다. 계산된 가격은 스팟 인스턴스에 대해 지불할 의사가 있는 최대 가격이며, 이 예제에서 Amazon EC2는 비용이 0.25626를 초과하는 모든 스팟 인스턴스 유형을 제외합니다.
- 식별된 최저 가격보다 높은 가격의 스팟 인스턴스 유형 비율[SpotMaxPricePercentageOverLowestPrice]

  예를 들어 식별된 스팟 인스턴스 유형 가격이 0.1808이고 25를 지정하는 경우 가격 기준 금액은 0.1808보다 25% 높습니다. 이는 0.1808 * 1.25 = 0.226으로 계산됩니다. 계산된 가격은 스팟 인스턴스에 대해 지불할 의사가 있는 최대 가격이며, 이 예제에서 Amazon EC2는 비용이 0.266를 초과하는 모든 스팟 인스턴스 유형을 제외합니다. 스팟 가격은 변동될 수 있으므로 가격 보호 기준도 변동될 수 있으므로 이 파라미터를 사용하지 않는 것이 좋습니다.

가격 보호 임곗값 지정

AWS CLI를 사용하여 가격 보호 임곗값 지정

AWS CLI를 사용하여 EC2 플릿 또는 스팟 플릿을 생성하는 동안 속성 기반 인스턴스 유형 선택을 위해 플릿을 구성하고 다음을 수행합니다.

- 온디맨드 인스턴스 가격 보호 임곗값을 지정하려면 JSON 구성 파일의 `InstanceRequirements` 구조에서 `OnDemandMaxPricePercentageOverLowestPrice`에 대해 가격 보호 임곗값을 백분율로 입력합니다.
- 스팟 인스턴스 가격 보호 기준을 지정하려면 JSON 구성 파일의 `InstanceRequirements` 구조에서 다음 파라미터 중 하나를 지정합니다.
  - `MaxSpotPriceAsPercentageOfOptimalOnDemandPrice`에 가격 보호 기준을 백분율로 입력합니다.
  - `SpotMaxPricePercentageOverLowestPrice`에 가격 보호 기준을 백분율로 입력합니다.

자세한 내용은 속성 기반 인스턴스 유형 선택으로 EC2 플릿 생성 또는 속성 기반 인스턴스 유형 선택으로 스팟 플릿 생성을 참조하세요.

(스팟 플릿만 해당) 콘솔을 사용하여 가격 보호 임곗값 지정

콘솔에서 스팟 플릿을 생성하는 동안 속성 기반 인스턴스 유형 선택을 위해 플릿을 구성하고 다음을 수행합니다.

- 온디맨드 인스턴스 가격 보호 임곗값을 지정하려면 추가 인스턴스 속성에서 온디맨드 가격 보호를 선택하고, 속성 추가를 선택한 다음 가격 보호 임곗값을 백분율로 입력합니다.
- 스팟 인스턴스 가격 보호 임곗값을 지정하려면 추가 인스턴스 속성에서 스팟 가격 보호를 선택하고, 속성 추가를 선택하고 가격 기준이 될 기준 값을 선택한 다음 가격 보호 임곗값을 백분율로 입력합니다.

> **Note**
>
> 플릿을 생성할 때 `TargetCapacityUnitType`을 `vcpu` 또는 `memory-mib`로 설정하면 가격 보호 임곗값이 인스턴스당 가격 대신 vCPU당 또는 메모리당 가격을 기준으로 적용됩니다.

성능 보호

성능 보호는 EC2 플릿 또는 스팟 플릿이 지정된 성능 기준과 비슷하거나 초과하는 인스턴스 유형을 사용하도록 하는 기능입니다. 성능 보호를 사용하려면 인스턴스 패밀리를 기준 참조로 지정합니다. 지정된 인스턴스 패밀리의 기능에 따라 허용 가능한 최저 수준의 성능이 정해집니다. Amazon EC2는 플릿의 인스턴스 유형을 선택할 때 사용자가 지정한 속성과 성능 기준을 고려합니다. 성능 기준보다 낮은 인스턴스 유형은 사용자가 지정한 다른 속성과 일치하더라도 선택에서 자동으로 제외됩니다. 이렇게

하면 선택한 모든 인스턴스 유형이 지정된 인스턴스 패밀리에서 설정한 기준과 비슷하거나 더 나은 성능을 제공할 수 있습니다. Amazon EC2는 이 기준을 사용하여 인스턴스 유형 선택을 안내하지만 선택한 인스턴스 유형이 모든 애플리케이션의 기준을 항상 초과한다는 보장은 없습니다.

현재 이 기능은 CPU 성능만 기준 성능 요소로 지원합니다. 지정된 인스턴스 패밀리의 CPU 프로세서의 CPU 성능은 성능 기준 역할을 하며 선택한 인스턴스 유형이 이 기준과 비슷하거나 이를 초과하도록 합니다. CPU 프로세서가 동일한 인스턴스 패밀리는 네트워크 또는 디스크 성능이 다르더라도 필터링 결과가 동일합니다. 예를 들어 `c6in` 또는 `c6i`를 기준 참조로 지정하면 두 인스턴스 패밀리가 동일한 CPU 프로세서를 사용하기 때문에 동일한 성능 기반 필터링 결과가 생성됩니다.

지원되지 않는 인스턴스 패밀리

다음 인스턴스 패밀리는 성능 보호에서 지원되지 않습니다.

- `c1`
- `g3 | g3s`
- `hpc7g`
- `m1 | m2`
- `mac1 | mac2 | mac2-m1ultra | mac2-m2 | mac2-m2pro`
- `p3dn | p4d | p5`
- `t1`
- `u-12tb1 | u-18tb1 | u-24tb1 | u-3tb1 | u-6tb1 | u-9tb1 | u7i-12tb | u7in-16tb | u7in-24tb | u7in-32tb`

지원되는 인스턴스 패밀리를 지정하여 성능 보호를 활성화하면 위의 지원되지 않는 인스턴스 패밀리는 반환되는 인스턴스 유형에서 제외됩니다.

지원되지 않는 인스턴스 패밀리를 기준 성능 값으로 지정하면 API는 GetInstanceTypesFromInstanceRequirements에 대한 빈 응답과 CreateFleet, RequestSpotFleet, ModifyFleet, ModifySpotFleetRequest에 대한 예외를 반환합니다.

예: CPU 성능 기준 설정

다음 예제에서 인스턴스 요구 사항은 `c6i` 인스턴스 패밀리만큼 성능이 뛰어난 CPU 코어가 있는 인스턴스 유형으로 시작하는 것입니다. 이렇게 하면 vCPU 수 등 지정된 다른 인스턴스 요구 사항을 충족하더라도 CPU 성능이 이보다 낮은 인스턴스 유형은 필터링됩니다. 예를 들어 지정된 인스턴스 속성에

vCPU 4개와 메모리 16GB가 포함된 경우 이러한 속성이 있지만 CPU 성능이 c6i보다 낮은 인스턴스 유형은 선택에서 제외됩니다.

```
"BaselinePerformanceFactors": {
        "Cpu": {
            "References": [
                {
                    "InstanceFamily": "c6i"
                }
            ]
        }
```

고려 사항

- EC2 플릿 또는 스팟 플릿에서 인스턴스 유형 또는 인스턴스 속성을 지정할 수 있지만 둘 다 동시에 지정할 수는 없습니다.

  CLI를 사용할 때 시작 템플릿 재정의가 시작 템플릿을 재정의합니다. 예를 들어 시작 템플릿에 인스턴스 유형이 포함되어 있고 시작 템플릿 재정의에 인스턴스 속성이 포함되어 있는 경우 인스턴스 속성으로 식별되는 인스턴스는 시작 템플릿의 인스턴스 유형을 재정의합니다.
- CLI를 사용하고 인스턴스 속성을 재정의로 지정할 때 가중치나 우선순위를 지정할 수도 없습니다.
- 요청 구성에서 최대 4개의 InstanceRequirements 구조를 지정할 수 있습니다.

속성 기반 인스턴스 유형 선택으로 EC2 플릿 생성

속성 기반 인스턴스 유형 선택을 사용하도록 EC2 플릿을 구성할 수 있습니다. Amazon EC2 콘솔을 사용하여 EC2 플릿을 생성할 수는 없습니다.

속성 기반 인스턴스 유형 선택을 위한 속성은 InstanceRequirements 구조에 지정되어 있습니다. InstanceRequirements가 플릿 구성에 포함되어 있으면 InstanceType 및 WeightedCapacity는 인스턴스 속성과 동시에 플릿 구성을 결정할 수 없으므로 제외되어야 합니다.

AWS CLI 및 PowerShell 예제에서는 다음 속성이 지정됩니다.

- VCpuCount - 최소 2개의 vCPU와 최대 4개의 vCPU. 최대 한도가 필요하지 않은 경우 최댓값을 생략할 수 있습니다.
- MemoryMiB - 최소 8GiB의 메모리와 최대 16GiB의 메모리. 최대 한도가 필요하지 않은 경우 최댓값을 생략할 수 있습니다.

이 구성은 2~4개의 vCPU와 8~16GiB의 메모리를 탑재한 인스턴스 유형을 모두 식별합니다. 단, EC2 플릿에서 플릿을 제공하는 경우 가격 보호 및 할당 전략에서 일부 인스턴스 유형이 제외될 수 있습니다.

AWS CLI

속성 기반 인스턴스 유형 선택으로 EC2 플릿을 생성하려면

create-fleet 명령을 사용하여 EC2 플릿을 생성합니다. JSON 파일에 플릿 구성을 지정합니다.

```
aws ec2 create-fleet \
    --region us-east-1 \
    --cli-input-json file://file_name.json
```

다음 예제 *file_name*.json 파일에는 속성 기반 인스턴스 유형 선택 방식을 사용하도록 EC2 플릿을 구성하는 파라미터가 포함되어 있습니다.

```
{
    "SpotOptions": {
        "AllocationStrategy": "price-capacity-optimized"
    },
    "LaunchTemplateConfigs": [{
        "LaunchTemplateSpecification": {
            "LaunchTemplateName": "my-launch-template",
            "Version": "1"
        },
        "Overrides": [{
            "InstanceRequirements": {
                "VCpuCount": {
                    "Min": 2,
                    "Max": 4
                },
                "MemoryMiB": {
                    "Min": 8192,
                    "Max": 16384
                }
            }
        }]
    }],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
```

```
    },
    "Type": "instant"
}
```

PowerShell

속성 기반 인스턴스 유형 선택으로 EC2 플릿을 생성하려면

New-EC2Fleet cmdlet을 사용합니다.

```
$vcpuCount = New-Object Amazon.EC2.Model.VCpuCountRangeRequest
$vcpuCount.Min = 2
$vcpuCount.Max = 4
$memoryMiB = New-Object Amazon.EC2.Model.MemoryMiBRequest
$memoryMiB.Min = 8192
$memoryMiB.Max = 16384
$instanceRequirements = New-Object Amazon.EC2.Model.InstanceRequirementsRequest
$instanceRequirements.VCpuCount = $vcpuCount
$instanceRequirements.MemoryMiB = $memoryMiB

$launchTemplateSpec = New-Object
 Amazon.EC2.Model.FleetLaunchTemplateSpecificationRequest
$launchTemplateSpec.LaunchTemplateName = "my-launch-template"
$launchTemplateSpec.Version = "1"
$override = New-Object Amazon.EC2.Model.FleetLaunchTemplateOverridesRequest
$override.InstanceRequirements = $instanceRequirements
$launchTemplateConfig = New-Object Amazon.EC2.Model.FleetLaunchTemplateConfigRequest
$launchTemplateConfig.LaunchTemplateSpecification = $launchTemplateSpec
$launchTemplateConfig.Overrides = @($override)

New-EC2Fleet `
    -SpotOptions_AllocationStrategy "price-capacity-optimized" `
    -LaunchTemplateConfig @($launchTemplateConfig) `
    -TargetCapacitySpecification_DefaultTargetCapacityType "spot" `
    -TargetCapacitySpecification_TotalTargetCapacity 20 `
    -Type "instant"
```

속성 기반 인스턴스 유형 선택으로 스팟 플릿 생성

속성 기반 인스턴스 유형 선택을 사용하도록 플릿을 구성할 수 있습니다.

속성 기반 인스턴스 유형 선택을 위한 속성은 `InstanceRequirements` 구조에 지정되어 있습니다. `InstanceRequirements`가 플릿 구성에 포함되어 있으면 `InstanceType` 및 `WeightedCapacity`는 인스턴스 속성과 동시에 플릿 구성을 결정할 수 없으므로 제외되어야 합니다.

AWS CLI 및 PowerShell 예제에서는 다음 속성이 지정됩니다.

- `VCpuCount` - 최소 2개의 vCPU와 최대 4개의 vCPU. 최대 한도가 필요하지 않은 경우 최댓값을 생략할 수 있습니다.
- `MemoryMiB` - 최소 8GiB의 메모리와 최대 16GiB의 메모리. 최대 한도가 필요하지 않은 경우 최댓값을 생략할 수 있습니다.

이 구성은 2~4개의 vCPU와 8~16GiB의 메모리를 탑재한 인스턴스 유형을 모두 식별합니다. 단, 스팟 플릿에서 플릿을 제공하는 경우 가격 보호 및 할당 전략에서 일부 인스턴스 유형이 제외될 수 있습니다.

Console

속성 기반 인스턴스 유형 선택을 위해 스팟 플릿을 구성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청(Spot Requests)을 선택한 다음 스팟 인스턴스 요청(Request Spot Instance)을 선택합니다.
3. 단계에 따라 스팟 플릿을 생성합니다. 자세한 내용은 정의된 파라미터를 사용하여 스팟 플릿 요청 생성 섹션을 참조하세요.

   스팟 플릿을 생성하는 동안 다음과 같이 속성 기반 인스턴스 유형 선택을 위해 플릿을 구성합니다.

   a. 인스턴스 유형 요구 사항(Instance type requirements)에서 컴퓨팅 요구 사항에 맞는 인스턴스 속성 지정(Specify instance attributes that match your compute requirements)을 선택합니다.

   b. vCPU(vCPUs)에 원하는 최소 및 최대 vCPU 수를 입력합니다. 무한을 지정하려면 최소 없음(No minimum), 최대 없음(No maximum) 또는 둘 다 선택합니다.

   c. 메모리(GiB)(Memory (GiB))에 원하는 최소 및 최대 메모리 양을 입력합니다. 무한을 지정하려면 최소 없음(No minimum), 최대 없음(No maximum) 또는 둘 다 선택합니다.

d. (선택 사항) 추가 인스턴스 속성(Additional instance attributes)에서 필요에 따라 하나 이상의 속성을 지정하여 컴퓨팅 요구 사항을 더 자세히 표현할 수 있습니다. 각 추가 속성은 요청에 제약 조건을 추가합니다.

e. (선택 사항) 지정한 속성을 가진 인스턴스 유형을 보려면 일치하는 인스턴스 유형 미리 보기(Preview matching instance types)를 확장합니다.

AWS CLI

속성 기반 인스턴스 유형 선택을 위해 스팟 플릿을 구성하려면

request-spot-fleet 명령을 사용하여 스팟 플릿을 생성합니다. JSON 파일에 플릿 구성을 지정합니다.

```
aws ec2 request-spot-fleet \
    --region us-east-1 \
    --spot-fleet-request-config file://file_name.json
```

다음 예제 *file_name*.json 파일에는 속성 기반 인스턴스 유형 선택 방식을 사용하도록 스팟 플릿을 구성하는 파라미터가 포함되어 있습니다.

```
{
    "AllocationStrategy": "priceCapacityOptimized",
    "TargetCapacity": 20,
    "Type": "request",
    "LaunchTemplateConfigs": [{
        "LaunchTemplateSpecification": {
            "LaunchTemplateName": "my-launch-template",
            "Version": "1"
        },
        "Overrides": [{
            "InstanceRequirements": {
                "VCpuCount": {
                    "Min": 2,
                    "Max": 4
                },
                "MemoryMiB": {
                    "Min": 8192,
                    "Max": 16384
                }
            }
        }]
```

```
    }]
}
```

PowerShell

속성 기반 인스턴스 유형 선택을 위해 스팟 플릿을 구성하려면

Request-EC2SpotFleet cmdlet을 사용합니다.

```
$vcpuCount = New-Object Amazon.EC2.Model.VCpuCountRange
$vcpuCount.Min = 2
$vcpuCount.Max = 4
$memoryMiB = New-Object Amazon.EC2.Model.MemoryMiB
$memoryMiB.Min = 8192
$memoryMiB.Max = 16384
$instanceRequirements = New-Object Amazon.EC2.Model.InstanceRequirements
$instanceRequirements.VCpuCount = $vcpuCount
$instanceRequirements.MemoryMiB = $memoryMiB

$launchTemplateSpec = New-Object Amazon.EC2.Model.FleetLaunchTemplateSpecification
$launchTemplateSpec.LaunchTemplateName = "my-launch-template"
$launchTemplateSpec.Version = "1"
$override = New-Object Amazon.EC2.Model.LaunchTemplateOverrides
$override.InstanceRequirements = $instanceRequirements
$launchTemplateConfig = New-Object Amazon.EC2.Model.LaunchTemplateConfig
$launchTemplateConfig.LaunchTemplateSpecification = $launchTemplateSpec
$launchTemplateConfig.Overrides = @($override)

Request-EC2SpotFleet `
    -SpotFleetRequestConfig_AllocationStrategy "PriceCapacityOptimized" `
    -SpotFleetRequestConfig_TargetCapacity 20 `
    -SpotFleetRequestConfig_Type "Request" `
    -SpotFleetRequestConfig_LaunchTemplateConfig $launchTemplateConfig
```

유효한 EC2 플릿 구성과 유효하지 않은 구성 예제

AWS CLI를 사용하여 EC2 플릿을 생성하려면 플릿 구성이 유효한지 확인해야 합니다. 다음 예에서는 유효한 구성과 유효하지 않은 구성을 보여줍니다.

다음을 포함하는 구성은 유효하지 않은 것으로 간주됩니다.

- 둘 다 `InstanceRequirements` 및 `InstanceType`인 단일 `Overrides` 구조

- 하나는 InstanceRequirements이고 다른 하나는 InstanceType인 2개의 Overrides 구조
- 동일한 LaunchTemplateSpecification 내에서 겹치는 속성 값을 갖는 2개의 InstanceRequirements 구조

구성의 예

- 유효한 구성: 재정의가 있는 단일 시작 템플릿
- 유효한 구성: 여러 InstanceRequirements가 있는 단일 시작 템플릿
- 유효한 구성: 각각 재정의가 있는 2개의 시작 템플릿
- 유효한 구성: InstanceRequirements만 지정, 겹치는 속성 값 없음
- 구성이 유효하지 않음: Overrides가 InstanceRequirements 및 InstanceType 포함
- 구성이 유효하지 않음: 2개의 Overrides가 InstanceRequirements 및 InstanceType 포함
- 구성이 유효하지 않음: 속성 값 겹침

유효한 구성: 재정의가 있는 단일 시작 템플릿

다음 구성은 유효합니다. 여기에는 하나의 시작 템플릿과 하나의 InstanceRequirements 구조를 갖는 하나의 Overrides 구조가 포함됩니다. 예제 구성에 대한 텍스트 설명은 다음과 같습니다.

```
{
        "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "My-launch-template",
                "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 2,
                            "Max": 8
                        },
                        "MemoryMib": {
                            "Min": 0,
                            "Max": 10240
                        },
                        "MemoryGiBPerVCpu": {
                            "Max": 10000
```

```
                        },
                        "RequireHibernateSupport": true
                    }
                }
            ]
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 5000,
        "DefaultTargetCapacityType": "spot",
        "TargetCapacityUnitType": "vcpu"
        }
    }
}
```

InstanceRequirements

속성 기반 인스턴스 선택을 사용하려면 플릿 구성에 InstanceRequirements 구조를 포함하고 플릿의 인스턴스에 대해 원하는 속성을 지정합니다.

앞의 예제에서 다음과 같은 인스턴스 속성이 지정됩니다.

- VCpuCount - 인스턴스 유형에 최소 2개, 최대 8개의 vCPU가 있어야 합니다.
- MemoryMiB - 인스턴스 유형에 최대 10,240MiB의 메모리가 있어야 합니다. 최솟값 0은 최소 제한이 없음을 나타냅니다.
- MemoryGiBPerVCpu - 인스턴스 유형에 vCPU당 최대 10,000GiB의 메모리가 있어야 합니다. Min 파라미터는 선택 항목입니다. 생략하면 최소 제한이 없음을 나타냅니다.

TargetCapacityUnitType

TargetCapacityUnitType 파라미터는 목표 용량의 단위를 지정합니다. 이 예에서 목표 용량은 5000이고 목표 용량 단위 유형은 vcpu입니다. 이들은 함께 원하는 목표 용량 vCPU 5,000개를 지정합니다. EC2 플릿은 플릿의 총 vCPU 수가 5,000개가 되도록 충분한 인스턴스를 시작합니다.

유효한 구성: 여러 InstanceRequirements가 있는 단일 시작 템플릿

다음 구성은 유효합니다. 하나의 시작 템플릿과 2개의 InstanceRequirements 구조를 갖는 하나의 Overrides 구조가 포함됩니다. InstanceRequirements에 지정된 속성은 값이 겹치지 않기 때문에 유효합니다. 첫 번째 InstanceRequirements 구조는 vCPU 0~2개의 VCpuCount를 지정하고 두 번째 InstanceRequirements 구조는 vCPU 4~8개를 지정합니다.

```
{
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                },
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 4,
                            "Max": 8
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
            }
        ],
        "TargetCapacitySpecification": {
            "TotalTargetCapacity": 1,
            "DefaultTargetCapacityType": "spot"
        }
    }
}
```

유효한 구성: 각각 재정의가 있는 2개의 시작 템플릿

다음 구성은 유효합니다. 각각 하나의 InstanceRequirements 구조를 포함하는 하나의 Overrides 구조가 있는 2개의 시작 템플릿이 포함됩니다. 이 구성은 동일한 플릿에서 arm 및 x86 아키텍처 지원에 유용합니다.

```
{
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "armLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                },
                {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "x86LaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
```

```
            }
        ],
         "TargetCapacitySpecification": {
            "TotalTargetCapacity": 1,
            "DefaultTargetCapacityType": "spot"
        }
    }
}
```

유효한 구성: **InstanceRequirements**만 지정, 겹치는 속성 값 없음

다음 구성은 유효합니다. 각각 시작 템플릿이 있고 InstanceRequirements 구조가 포함된 Overrides 구조가 있는 2개의 LaunchTemplateSpecification 구조가 포함됩니다. InstanceRequirements에 지정한 속성은 값이 겹치지 않기 때문에 유효합니다. 첫 번째 InstanceRequirements 구조는 vCPU 0~2개의 VCpuCount를 지정하고 두 번째 InstanceRequirements 구조는 vCPU 4~8개를 지정합니다.

```
{
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
            },
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyOtherLaunchTemplate",
                    "Version": "1"
                },
```

```
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 4,
                            "Max": 8
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
            }
        ],
        "TargetCapacitySpecification": {
            "TotalTargetCapacity": 1,
            "DefaultTargetCapacityType": "spot"
        }
    }
}
```

구성이 유효하지 않음: **Overrides**가 **InstanceRequirements** 및 **InstanceType** 포함

다음 구성은 유효하지 않습니다. Overrides 구조에 InstanceRequirements 및 InstanceType이 모두 포함됩니다. Overrides에서 InstanceRequirements 또는 InstanceType 중 하나를 지정할 수 있지만 둘 다 지정할 수는 없습니다.

```
{
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
```

```
                            "Min": 0
                        }
                    }
                },
                {
                    "InstanceType": "m5.large"
                }
            ]
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 1,
        "DefaultTargetCapacityType": "spot"
    }
}
}
```

구성이 유효하지 않음: 2개의 **Overrides**가 **InstanceRequirements** 및 **InstanceType** 포함

다음 구성은 유효하지 않습니다. Overrides 구조에 InstanceRequirements 및 InstanceType이 모두 포함됩니다. InstanceRequirements 또는 InstanceType 중 하나를 지정할 수 있지만 다른 Overrides 구조라고 하더라도 둘 다 지정할 수는 없습니다.

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "MyLaunchTemplate",
                "Version": "1"
            },
            "Overrides": [
            {
                "InstanceRequirements": {
                    "VCpuCount": {
                        "Min": 0,
                        "Max": 2
                    },
                    "MemoryMiB": {
                        "Min": 0
                    }
                }
            }
        ]
```

```
            },
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyOtherLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceType": "m5.large"
                }
              ]
            }
        ],
         "TargetCapacitySpecification": {
            "TotalTargetCapacity": 1,
            "DefaultTargetCapacityType": "spot"
        }
    }
}
```

구성이 유효하지 않음: 속성 값 겹침

다음 구성은 유효하지 않습니다. 2개의 InstanceRequirements 구조는 각각 "VCpuCount": {"Min": 0, "Max": 2}를 포함합니다. 이러한 속성의 값이 겹치므로 용량 풀이 중복됩니다.

```
{
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    },
```

```
                    {
                      "InstanceRequirements": {
                          "VCpuCount": {
                              "Min": 0,
                              "Max": 2
                          },
                          "MemoryMiB": {
                              "Min": 0
                          }
                      }
                  }
                }
              ]
            }
        ],
         "TargetCapacitySpecification": {
            "TotalTargetCapacity": 1,
            "DefaultTargetCapacityType": "spot"
        }
    }
}
```

유효한 스팟 플릿 구성과 유효하지 않은 구성 예제

AWS CLI를 사용하여 스팟 플릿을 생성하려면 플릿 구성이 유효한지 확인해야 합니다. 다음 예에서는 유효한 구성과 유효하지 않은 구성을 보여줍니다.

다음을 포함하는 구성은 유효하지 않은 것으로 간주됩니다.

- 둘 다 `InstanceRequirements` 및 `InstanceType`인 단일 `Overrides` 구조
- 하나는 `InstanceRequirements`이고 다른 하나는 `InstanceType`인 2개의 `Overrides` 구조
- 동일한 `LaunchTemplateSpecification` 내에서 겹치는 속성 값을 갖는 2개의 `InstanceRequirements` 구조

구성의 예

- 유효한 구성: 재정의가 있는 단일 시작 템플릿
- 유효한 구성: 여러 InstanceRequirements가 있는 단일 시작 템플릿
- 유효한 구성: 각각 재정의가 있는 2개의 시작 템플릿
- 유효한 구성: InstanceRequirements만 지정, 겹치는 속성 값 없음

- 구성이 유효하지 않음: Overrides가 InstanceRequirements 및 InstanceType 포함
- 구성이 유효하지 않음: 2개의 Overrides가 InstanceRequirements 및 InstanceType 포함
- 구성이 유효하지 않음: 속성 값 겹침

유효한 구성: 재정의가 있는 단일 시작 템플릿

다음 구성은 유효합니다. 여기에는 하나의 시작 템플릿과 하나의 `InstanceRequirements` 구조를 갖는 하나의 `Overrides` 구조가 포함됩니다. 예제 구성에 대한 텍스트 설명은 다음과 같습니다.

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "My-launch-template",
                "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 2,
                            "Max": 8
                        },
                        "MemoryMib": {
                            "Min": 0,
                            "Max": 10240
                        },
                        "MemoryGiBPerVCpu": {
                            "Max": 10000
                        },
                        "RequireHibernateSupport": true
                    }
                }
            ]
        }
    ],
        "TargetCapacity": 5000,
```

```
            "OnDemandTargetCapacity": 0,
            "TargetCapacityUnitType": "vcpu"
    }
}
```

InstanceRequirements

속성 기반 인스턴스 선택을 사용하려면 플릿 구성에 InstanceRequirements 구조를 포함하고 플릿의 인스턴스에 대해 원하는 속성을 지정합니다.

앞의 예제에서 다음과 같은 인스턴스 속성이 지정됩니다.

- VCpuCount - 인스턴스 유형에 최소 2개, 최대 8개의 vCPU가 있어야 합니다.
- MemoryMiB - 인스턴스 유형에 최대 10,240MiB의 메모리가 있어야 합니다. 최솟값 0은 최소 제한이 없음을 나타냅니다.
- MemoryGiBPerVCpu - 인스턴스 유형에 vCPU당 최대 10,000GiB의 메모리가 있어야 합니다. Min 파라미터는 선택 항목입니다. 생략하면 최소 제한이 없음을 나타냅니다.

TargetCapacityUnitType

TargetCapacityUnitType 파라미터는 목표 용량의 단위를 지정합니다. 이 예에서 목표 용량은 5000이고 목표 용량 단위 유형은 vcpu입니다. 이들은 함께 원하는 목표 용량 vCPU 5,000개를 지정합니다. 스팟 플릿은 플릿의 총 vCPU 수가 5,000개가 되도록 충분한 인스턴스를 시작합니다.

유효한 구성: 여러 InstanceRequirements가 있는 단일 시작 템플릿

다음 구성은 유효합니다. 하나의 시작 템플릿과 2개의 InstanceRequirements 구조를 갖는 하나의 Overrides 구조가 포함됩니다. InstanceRequirements에 지정된 속성은 값이 겹치지 않기 때문에 유효합니다. 첫 번째 InstanceRequirements 구조는 vCPU 0~2개의 VCpuCount를 지정하고 두 번째 InstanceRequirements 구조는 vCPU 4~8개를 지정합니다.

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
```

```
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                },
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 4,
                            "Max": 8
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
            }
        ],
        "TargetCapacity": 1,
        "OnDemandTargetCapacity": 0,
        "Type": "maintain"
    }
}
```

유효한 구성: 각각 재정의가 있는 2개의 시작 템플릿

다음 구성은 유효합니다. 각각 하나의 InstanceRequirements 구조를 포함하는 하나의 Overrides 구조가 있는 2개의 시작 템플릿이 포함됩니다. 이 구성은 동일한 플릿에서 arm 및 x86 아키텍처 지원에 유용합니다.

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
```

```
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "armLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                },
                {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "x86LaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
            }
        ],
        "TargetCapacity": 1,
        "OnDemandTargetCapacity": 0,
        "Type": "maintain"
```

```
    }
}
```

유효한 구성: **InstanceRequirements**만 지정, 겹치는 속성 값 없음

다음 구성은 유효합니다. 각각 시작 템플릿이 있고 InstanceRequirements 구조가 포함된 Overrides 구조가 있는 2개의 LaunchTemplateSpecification 구조가 포함됩니다. InstanceRequirements에 지정한 속성은 값이 겹치지 않기 때문에 유효합니다. 첫 번째 InstanceRequirements 구조는 vCPU 0~2개의 VCpuCount를 지정하고 두 번째 InstanceRequirements 구조는 vCPU 4~8개를 지정합니다.

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
            },
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyOtherLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
```

```
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 4,
                            "Max": 8
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
            }
        ],
        "TargetCapacity": 1,
        "OnDemandTargetCapacity": 0,
        "Type": "maintain"
    }
}
```

구성이 유효하지 않음: **Overrides**가 **InstanceRequirements** 및 **InstanceType** 포함

다음 구성은 유효하지 않습니다. Overrides 구조에 InstanceRequirements 및 InstanceType이 모두 포함됩니다. Overrides에서 InstanceRequirements 또는 InstanceType 중 하나를 지정할 수 있지만 둘 다 지정할 수는 없습니다.

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
```

```
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                },
                {
                    "InstanceType": "m5.large"
                }
              ]
            }
        ],
        "TargetCapacity": 1,
        "OnDemandTargetCapacity": 0,
        "Type": "maintain"
    }
}
```

구성이 유효하지 않음: 2개의 **Overrides**가 **InstanceRequirements** 및 **InstanceType** 포함

다음 구성은 유효하지 않습니다. Overrides 구조에 InstanceRequirements 및 InstanceType이 모두 포함됩니다. InstanceRequirements 또는 InstanceType 중 하나를 지정할 수 있지만 다른 Overrides 구조라고 하더라도 둘 다 지정할 수는 없습니다.

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
```

```
                    },
                    "MemoryMiB": {
                        "Min": 0
                    }
                }
            }
          ]
        },
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "MyOtherLaunchTemplate",
                "Version": "1"
            },
            "Overrides": [
            {
                "InstanceType": "m5.large"
            }
          ]
        }
    ],
    "TargetCapacity": 1,
    "OnDemandTargetCapacity": 0,
    "Type": "maintain"
  }
}
```

구성이 유효하지 않음: 속성 값 겹침

다음 구성은 유효하지 않습니다. 2개의 InstanceRequirements 구조는 각각 "VCpuCount": {"Min": 0, "Max": 2}를 포함합니다. 이러한 속성의 값이 겹치므로 용량 풀이 중복됩니다.

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
```

```
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    },
                    {
                      "InstanceRequirements": {
                          "VCpuCount": {
                              "Min": 0,
                              "Max": 2
                          },
                          "MemoryMiB": {
                              "Min": 0
                          }
                      }
                  }
                }
              ]
            }
        ],
        "TargetCapacity": 1,
        "OnDemandTargetCapacity": 0,
        "Type": "maintain"
    }
}
```

지정한 속성을 가진 인스턴스 유형 미리 보기

get-instance-types-from-instance-requirements 명령을 사용하여 지정한 속성과 일치하는 인스턴스 유형을 미리 봅니다. 이 기능은 인스턴스를 시작하지 않고 요청 구성에서 지정할 속성을 계산할 때 특히 유용합니다. 이 명령은 사용 가능한 용량을 고려하지 않습니다.

AWS CLI로 속성을 지정하여 인스턴스 유형 목록 미리 보기

1. (선택 사항) 지정할 수 있는 모든 가능한 속성을 생성하려면 get-instance-types-from-instance-requirements 명령과 `--generate-cli-skeleton` 파라미터를 사용합니다. 선택적으로 `input > attributes.json`을 사용하여 출력을 파일로 지정하여 저장할 수 있습니다.

```
aws ec2 get-instance-types-from-instance-requirements \
    --region us-east-1 \
    --generate-cli-skeleton input > attributes.json
```

예상 결과

```
{
    "DryRun": true,
    "ArchitectureTypes": [
        "i386"
    ],
    "VirtualizationTypes": [
        "hvm"
    ],
    "InstanceRequirements": {
        "VCpuCount": {
            "Min": 0,
            "Max": 0
        },
        "MemoryMiB": {
            "Min": 0,
            "Max": 0
        },
        "CpuManufacturers": [
            "intel"
        ],
        "MemoryGiBPerVCpu": {
            "Min": 0.0,
            "Max": 0.0
        },
        "ExcludedInstanceTypes": [
            ""
        ],
        "InstanceGenerations": [
            "current"
        ],
        "SpotMaxPricePercentageOverLowestPrice": 0,
        "OnDemandMaxPricePercentageOverLowestPrice": 0,
        "BareMetal": "included",
        "BurstablePerformance": "included",
        "RequireHibernateSupport": true,
        "NetworkInterfaceCount": {
```

```
            "Min": 0,
            "Max": 0
        },
        "LocalStorage": "included",
        "LocalStorageTypes": [
            "hdd"
        ],
        "TotalLocalStorageGB": {
            "Min": 0.0,
            "Max": 0.0
        },
        "BaselineEbsBandwidthMbps": {
            "Min": 0,
            "Max": 0
        },
        "AcceleratorTypes": [
            "gpu"
        ],
        "AcceleratorCount": {
            "Min": 0,
            "Max": 0
        },
        "AcceleratorManufacturers": [
            "nvidia"
        ],
        "AcceleratorNames": [
            "a100"
        ],
        "AcceleratorTotalMemoryMiB": {
            "Min": 0,
            "Max": 0
        },
        "NetworkBandwidthGbps": {
            "Min": 0.0,
            "Max": 0.0
        },
        "AllowedInstanceTypes": [
            ""
        ]
    },
    "MaxResults": 0,
    "NextToken": ""
}
```

2. 이전 단계의 출력을 사용하여 JSON 구성 파일을 생성하고 다음과 같이 구성합니다.

> **Note**
>
> `ArchitectureTypes`, `VirtualizationTypes`, `VCpuCount` 및 `MemoryMiB`의 값을 입력해야 합니다. 다른 속성을 생략할 수 있으며 생략 시 기본값이 사용됩니다.
> 각 속성과 기본값에 대한 설명은 get-instance-types-from-instance-requirements를 참조하세요.

   a. `ArchitectureTypes`에 대해 하나 이상의 프로세서 아키텍처 유형을 지정합니다.

   b. `VirtualizationTypes`에 대해 하나 이상의 가상화 유형을 지정합니다.

   c. `VCpuCount`에 대해 최소 및 최대 vCPU 수를 지정합니다. 최소 제한을 지정하지 않으려면 `Min`에 대해 `0`을 지정합니다. 최대 제한을 지정하지 않으려면 `Max` 파라미터를 생략합니다.

   d. `MemoryMiB`에 대해 최소 및 최대 메모리 양(MiB)을 지정합니다. 최소 제한을 지정하지 않으려면 `Min`에 대해 `0`을 지정합니다. 최대 제한을 지정하지 않으려면 `Max` 파라미터를 생략합니다.

   e. 선택적으로 하나 이상의 다른 속성을 지정하여 반환되는 인스턴스 유형 목록을 추가로 제한할 수 있습니다.

3. JSON 파일에서 지정한 속성이 있는 인스턴스 유형을 미리 보려면 get-instance-types-from-instance-requirements 명령을 사용하고 `--cli-input-json` 파라미터를 통해 JSON 파일의 이름과 경로를 지정합니다. 필요에 따라 테이블 형식으로 표시되도록 출력 형식을 지정할 수 있습니다.

```
aws ec2 get-instance-types-from-instance-requirements \
    --cli-input-json file://attributes.json \
    --output table
```

예제 *attributes.json* 파일

이 예에서는 필수 속성이 JSON 파일에 포함되어 있습니다. 이들은 `ArchitectureTypes`, `VirtualizationTypes`, `VCpuCount` 및 `MemoryMiB`입니다. 또한 선택적인 `InstanceGenerations` 속성도 포함되어 있습니다. 참고로 `MemoryMiB`에 대해 `Max` 값을 생략하여 제한이 없음을 나타낼 수 있습니다.

```
{
```

```
    "ArchitectureTypes": [
        "x86_64"
    ],
    "VirtualizationTypes": [
        "hvm"
    ],
    "InstanceRequirements": {
        "VCpuCount": {
            "Min": 4,
            "Max": 6
        },
        "MemoryMiB": {
            "Min": 2048
        },
        "InstanceGenerations": [
            "current"
        ]
    }
}
```

출력 예시

```
-----------------------------------------
|GetInstanceTypesFromInstanceRequirements|
+---------------------------------------+
||             InstanceTypes             ||
|+-------------------------------------+|
||             InstanceType              ||
|+-------------------------------------+|
||  c4.xlarge                          ||
||  c5.xlarge                          ||
||  c5a.xlarge                         ||
||  c5ad.xlarge                        ||
||  c5d.xlarge                         ||
||  c5n.xlarge                         ||
||  d2.xlarge                          ||
...
```

4. 필요에 맞는 인스턴스 유형을 식별한 후 플릿 요청을 구성할 때 사용할 수 있도록 사용한 인스턴스 속성을 기록해 둡니다.

인스턴스 가중치를 사용하여 EC2 플릿 또는 스팟 플릿의 비용 및 성능 관리

인스턴스 가중치를 사용하면 EC2 플릿 또는 스팟 플릿의 각 인스턴스 유형에 가중치를 할당하여 서로 상대적인 컴퓨팅 용량과 성능을 반영합니다. 가중치를 기반으로 플릿은 원하는 목표 용량을 충족할 수 있는 한 지정된 인스턴스 유형을 원하는 대로 조합하여 사용할 수 있습니다. 이는 플릿의 비용과 성능 관리에 도움이 될 수 있습니다.

가중치는 인스턴스 유형이 총 목표 용량을 구성하는 용량 단위를 나타냅니다.

예제: 성능 관리를 위한 인스턴스 가중치 사용

플릿에 두 개의 인스턴스 유형이 있고, 다음과 같이 동일한 성능을 달성하기 위해 각 인스턴스 유형에 필요한 가중치를 반영하여 각 인스턴스 유형에 다른 가중치를 할당한다고 가정해 보겠습니다.

- `m5.large` – 가중치: 1
- `m5.2xlarge` – 가중치: 4

이 가중치를 할당하면 1개의 `m5.2xlarge`와 동일한 성능을 달성하려면 `m5.large` 인스턴스 4개가 필요하다는 뜻입니다.

주어진 목표 용량에 필요한 각 인스턴스 유형의 인스턴스 수를 계산하려면 다음 공식을 사용합니다.

```
target capacity / weight = number of instances
```

목표 용량 단위가 8개인 경우 플릿은 다음과 같이 `m5.large` 또는 `m5.2xlarge`, 아니면 두 가지를 혼합하여 목표 용량을 충족할 수 있습니다.

- `m5.large` 인스턴스 8개(용량 8/가중치 1 = 인스턴스 8개)
- `m5.2xlarge` 인스턴스 2개(용량 8/가중치 4 = 인스턴스 2개)
- `m5.large` 4개 및 `m5.2xlarge` 1개

예제: 비용 관리를 위한 인스턴스 가중치 사용

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 유닛 수로 나누어 계산할 수 있습니다. 플릿은 목표 용량을 인스턴스 가중치로 나누어 시작할 인스턴스의 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 플릿의 단위당 가격을 결정하기 위한 계산의 예제가 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)

플릿 인스턴스 가중치를 사용하여 다음과 같이 원하는 목표 용량을 충족 시점의 단위당 최저 가격으로 풀에서 프로비저닝합니다.

1. 플릿의 목표 용량을 인스턴스(기본값) 또는 선택한 단위(예: vCPU 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 플릿 요청을 고려하세요.

- 목표 용량은 24
- 인스턴스 유형이 r3.2xlarge이고 가중치가 6인 시작 사양
- 인스턴스 유형이 c3.xlarge이고 가중치가 5인 시작 사양

가중치는 목표 용량에 대하여 인스턴스 유형이 나타내는 단위 수를 의미합니다. 첫 번째 시작 사양에서 단위당 최저 가격(인스턴스 시간당 r3.2xlarge의 가격을 6으로 나눈 값)을 제공하는 경우 플릿은 이 인스턴스 중 4개(24를 6으로 나눈 값)를 시작합니다.

두 번째 시작 사양에서 단위당 최저 가격(인스턴스 시간당 c3.xlarge에 대한 가격을 5로 나눈 값)을 제공하는 경우 플릿은 이들 인스턴스 중 5개(24를 5로 나눈 결과를 올림한 값)를 시작합니다.

인스턴스 가중치 부여 및 할당 전략

다음과 같은 구성의 플릿 요청을 고려하세요.

- 목표 용량 스팟 인스턴스 30개
- 인스턴스 유형이 c3.2xlarge이고 가중치가 8인 시작 사양
- 인스턴스 유형이 m3.xlarge이고 가중치가 8인 시작 사양
- 인스턴스 유형이 r3.xlarge이고 가중치가 8인 시작 사양

플릿은 4개의 인스턴스(30을 8로 나눈 결과를 올림한 값)를 시작합니다. diversified 전략 사용 시 플릿은 풀 3개에서 각각 인스턴스 1개를 시작하고 풀 3개 중 어디에 있든 네 번째 인스턴스가 단위당 최저 가격을 제공합니다.

할당 전략을 사용하여 EC2 플릿 또는 스팟 플릿이 스팟 및 온디맨드 용량을 충족하는 방법 결정

EC2 플릿 또는 스팟 플릿에서 여러 용량 풀(각각 인스턴스 유형과 가용 영역으로 구성)을 사용하는 경우 할당 전략을 사용하여 Amazon EC2가 해당 용량 풀에서 스팟 및 온디맨드 용량을 충족하는 방식을 관리할 수 있습니다. 할당 전략을 통해 가용 용량, 가격 및 사용할 인스턴스 유형을 최적화할 수 있습니다. 스팟 인스턴스 및 온디맨드 인스턴스에 대한 할당 전략은 서로 다릅니다.

주제

- 스팟 인스턴스를 위한 할당 전략
- 온디맨드 인스턴스 할당 전략
- 적합한 스팟 할당 전략 선택
- 스팟 인스턴스의 목표 용량 유지
- 온디맨드 용량에 대한 인스턴스 유형 우선순위 지정

스팟 인스턴스를 위한 할당 전략

시작 구성은 EC2 플릿 또는 스팟 플릿이 스팟 인스턴스를 시작할 수 있는 모든 스팟 용량 풀(인스턴스 유형, 가용 영역)을 결정합니다. 하지만 인스턴스를 시작할 때 플릿은 지정한 할당 전략을 사용하여 가능한 풀에서 특정 풀을 선택합니다.

> **Note**
>
> (Linux 인스턴스만 해당) AMD SEV-SNP가 켜진 상태에서 스팟 인스턴스를 시작하도록 구성하면 선택한 인스턴스 유형의 온디맨드 시간당 요금의 10%에 해당하는 시간당 사용 요금이 추가로 부과됩니다. 할당 전략에서 가격을 입력으로 사용하는 경우 플릿에는 이 추가 요금이 포함되지 않습니다. 스팟 가격만 사용됩니다.

스팟 인스턴스에는 다음 할당 전략 중 하나를 지정할 수 있습니다.

가격 용량 최적화(권장)

플릿은 시작하는 인스턴스의 수에 맞추어 용량 가용성이 가장 높은 풀을 가져옵니다. 즉, 가까운 시일 내에 중단될 가능성이 가장 낮다고 판단되는 풀에서 스팟 인스턴스를 요청합니다. 그러면 플릿이 해당 풀에서 가장 가격이 낮은 스팟 인스턴스를 요청합니다.

가격 용량 최적화 할당 전략은 컨테이너화된 상태 비저장 애플리케이션, 마이크로서비스, 웹 애플리케이션, 데이터 및 분석 작업, 배치 처리와 같은 대부분의 스팟 워크로드에 가장 적합합니다.

AWS CLI를 사용하는 경우 파라미터 이름은 EC2 플릿의 경우 `price-capacity-optimized`, 스팟 플릿의 경우 `priceCapacityOptimized`입니다.

용량 최적화

플릿은 시작하는 인스턴스의 수에 맞추어 용량 가용성이 가장 높은 풀을 가져옵니다. 즉, 가까운 시일 내에 중단될 가능성이 가장 낮다고 판단되는 풀에서 스팟 인스턴스를 요청합니다. 플릿의 각 인스턴스 유형에 대해 우선순위를 설정할 수 있습니다. 이 경우 플릿은 용량을 먼저 최적화하지만 최선의 노력을 기준으로 인스턴스 유형 우선순위를 고려합니다.

스팟 인스턴스에서 요금은 시간이 지나면서 수요 및 공급의 장기 추세에 따라 서서히 변화하지만 용량은 실시간으로 변동합니다. 용량 최적화 전략은 실시간 용량 데이터를 기준으로 가장 가용성이 높은 풀을 예측하여 자동으로 스팟 인스턴스를 가장 가용성이 높은 풀로 시작합니다. 이 전략은 작업 재시작 시 중단으로 인한 비용이 상대적으로 높을 수 있는 워크로드에 유용합니다. 이 전략은 긴 지속적 통합(CI), 이미지 및 미디어 렌더링, 딥 러닝, 고성능 컴퓨팅(HPC)과 같이 작업 재시작 비용이 상대적으로 높을 수 있는 워크로드에 유용합니다. 용량 최적화 전략은 중단을 줄일 수 있는 가능성을 제공함으로써 전체 워크로드 비용을 낮출 수 있습니다.

사용자는 용량 최적화 우선 할당 전략과 우선순위 파라미터를 사용하여 우선순위가 높은 순서에서 낮은 순서로 인스턴스 유형 우선순위를 지정할 수 있습니다. 여러 인스턴스 유형에 동일한 우선순위를 설정할 수 있습니다. 플릿은 용량을 우선으로 최적화하지만 최선의 노력을 기준으로 인스턴스

유형 우선순위를 따릅니다. 예를 들어 플릿에서 최적 용량으로 프로비저닝하는 데 우선순위가 큰 영향을 미치지 않을 수 있습니다. 이 옵션은 중단 가능성을 최소화해야 하고 특정 인스턴스 유형에 대한 선호도가 중요한 워크로드에 적합합니다. 스팟 용량의 인스턴스 유형에 대한 우선순위를 설정할 때 온디맨드 할당 전략이 우선순위로 설정되어 있으면 온디맨드 인스턴스에도 동일한 우선순위가 적용됩니다. 스팟 플릿의 경우 우선순위 사용은 플릿이 시작 템플릿을 사용하는 경우에만 지원됩니다.

AWS CLI를 사용하는 경우 파라미터 이름은 EC2 플릿의 경우 `capacity-optimized`와 `capacity-optimized-prioritized`, 스팟 플릿의 경우 `capacityOptimized`와 `capacityOptimizedPrioritized`입니다.

다각화

스팟 인스턴스는 모든 스팟 용량 풀에 걸쳐 분산됩니다. AWS CLI를 사용하는 경우 파라미터 이름은 EC2 플릿과 스팟 플릿 모두에서 `diversified`입니다.

최저 가격(권장되지 않음)

> **Warning**
>
> 최저 가격 할당 전략은 스팟 인스턴스의 중단 위험이 가장 높기 때문에 권장하지 않습니다.

스팟 인스턴스는 용량이 있는 최저 가격 풀에서 제공됩니다. AWS CLI를 사용하는 경우에는 기본 전략입니다. 하지만 가격 용량 최적화 할당 전략을 지정하여 기본값을 재정의하는 것이 좋습니다.

최저 가격 전략을 사용하면 가장 낮은 가격의 풀에 사용 가능한 용량이 없는 경우 스팟 인스턴스는 사용 가능한 용량이 있는 다음으로 가장 낮은 가격의 풀에서 제공됩니다. 목표 용량을 충족하기 전에 풀에 용량이 부족해질 경우 플릿은 다음으로 가격이 낮은 풀에서 끌어와 요청을 계속 이행합니다. 목표 용량이 충족되었는지 확인하기 위해 여러 풀에서 스팟 인스턴스를 받을 수도 있습니다.

이 전략은 인스턴스 가격만 고려하고 용량 가용성은 고려하지 않기 때문에 중단률이 높아질 수 있습니다.

최저 가격 할당 전략은 AWS CLI를 사용하는 경우에만 제공됩니다. 파라미터 이름은 EC2 플릿의 경우 `lowest-price`, 스팟 플릿의 경우 `lowestPrice`입니다.

사용할 풀 수

대상 스팟 용량을 할당할 스팟 풀 수입니다. 스팟 할당 전략이 최저 가격으로 설정된 경우에만 유효합니다. 플릿에서 가격이 가장 낮은 스팟 풀을 선택하고 지정한 스팟 풀 수에 걸쳐 대상 스팟 용량을 균등하게 할당합니다.

플릿은 가능한 한도에서 지정한 풀 수에서 스팟 인스턴스를 끌어오려고 합니다. 목표 용량을 충족하기 전에 풀에 스팟 용량이 부족해질 경우 플릿은 다음으로 가격이 낮은 풀에서 끌어와 요청을 계속 이행합니다. 목표 용량이 충족되도록 하기 위해 지정한 풀 수보다 많은 수의 스팟 인스턴스를 받게 될 수 있습니다. 마찬가지로 대부분의 풀에 스팟 용량이 없는 경우 지정한 풀 수보다 적은 수의 풀에서 전체 목표 용량을 받을 수 있습니다.

이 파라미터는 최저 가격 할당 전략을 지정하고 AWS CLI를 사용하는 경우에만 제공됩니다. 파라미터 이름은 EC2 플릿과 스팟 플릿 모두에서 `InstancePoolsToUseCount`입니다.

온디맨드 인스턴스 할당 전략

시작 구성은 EC2 플릿 또는 스팟 플릿이 온디맨드 인스턴스를 시작할 수 있는 모든 용량 풀(인스턴스 유형, 가용 영역)을 결정합니다. 하지만 인스턴스를 시작할 때 플릿은 지정한 할당 전략을 사용하여 가능한 풀에서 특정 풀을 선택합니다.

온디맨드 인스턴스에는 다음 할당 전략 중 하나를 지정할 수 있습니다.

최저 가격

온디맨드 인스턴스는 용량이 있는 최저 가격 풀에서 제공됩니다. 이는 기본 전략입니다.

가장 낮은 가격의 풀에 사용 가능한 용량이 없는 경우 온디맨드 인스턴스는 사용 가능한 용량이 있는 다음으로 가장 낮은 가격의 풀에서 제공됩니다.

목표 용량을 충족하기 전에 풀에 용량이 부족해질 경우 플릿은 다음으로 가격이 낮은 풀에서 끌어와 요청을 계속 이행합니다. 목표 용량이 충족되었는지 확인하기 위해 여러 풀에서 온디맨드 인스턴스를 받을 수도 있습니다.

우선순위

플릿은 각 시작 템플릿 재정의에 할당한 우선순위를 사용하여 최우선순위 순서로 인스턴스 유형을 시작합니다. 이 전략은 속성 기반 인스턴스 유형 선택과 함께 사용할 수 없습니다. 이 할당 전략을 사용하는 방법에 대한 예제는 온디맨드 용량에 대한 인스턴스 유형 우선순위 지정의 내용을 참조하세요.

적합한 스팟 할당 전략 선택

적절한 스팟 할당 전략을 선택하여 사용 사례에 따라 플릿을 최적화할 수 있습니다.

최저 가격과 용량 가용성의 균형

가장 가격이 낮은 스팟 용량 풀과 용량 가용성이 가장 높은 스팟 용량 풀 간의 균형을 맞추려면 가격 용량 최적화 할당 전략을 사용하는 것이 좋습니다. 이 전략은 풀 가격과 해당 풀에 있는 스팟 인스턴스의 용량 가용성을 기준으로 스팟 인스턴스를 요청할 풀을 결정합니다. 즉, 가격을 고려하는 동시에 가까운 시일 내에 중단될 가능성이 가장 낮다고 판단되는 풀에서 스팟 인스턴스를 요청합니다.

플릿이 컨테이너화된 애플리케이션, 마이크로서비스, 웹 애플리케이션, 데이터 및 분석 작업, 배치 처리 등 복원력이 뛰어난 상태 비저장 워크로드를 실행하는 경우, 가격 용량 최적화 할당 전략을 사용하여 최적의 비용과 용량 가용성을 확보하세요.

플릿에서 작업 재시작 시 중단으로 인한 비용이 상대적으로 높을 수 있는 워크로드를 실행하는 경우, 애플리케이션이 중단되었을 때 해당 시점부터 다시 시작할 수 있도록 체크포인트를 구현해야 합니다. 체크포인트를 사용하면 가격이 가장 낮으면서 스팟 인스턴스 중단률도 낮은 풀에서 용량이 할당되므로, 가격 용량 최적화 할당 전략이 이러한 워크로드에도 적합하게 될 수 있습니다.

가격 용량 최적화 할당 전략을 사용하는 JSON 구성의 예제는 다음을 참조하세요.

- EC2 플릿 - 예제 10: price-capacity-optimized 플릿에서 스팟 인스턴스 시작
- 스팟 플릿 - 예제 11: priceCapacityOptimized 플릿에서 스팟 인스턴스 시작

워크로드의 중단 비용이 높은 경우

비슷한 가격의 인스턴스 유형을 사용하는 워크로드를 실행하거나 중단 비용이 너무 높아 중단이 미미하게 증가하는 데도 비용 절감 효과가 너무 적은 워크로드를 실행하는 경우, 용량 최적화 전략을 선택적으로 사용할 수 있습니다. 이 전략은 중단을 줄일 수 있는 가능성을 제공하는 가용성이 가장 높은 스팟 용량 풀에서 용량을 할당하므로, 전체 워크로드 비용을 낮출 수 있습니다.

중단 가능성을 최소화해야 하지만 특정 인스턴스 유형을 우선적으로 사용해야 하는 경우, 용량 최적화 우선 할당 전략을 사용한 다음 사용할 인스턴스 유형의 순서를 가장 높은 우선순위에서 가장 낮은 우선순위로 설정하여 풀 우선순위를 표현할 수도 있습니다.

용량 최적화 우선에 대한 우선순위를 설정할 때 온디맨드 할당 전략이 우선순위로 설정되어 있으면 온디맨드 인스턴스에도 동일한 우선순위가 적용됩니다. 또한 스팟 플릿의 경우 우선순위 사용은 플릿이 시작 템플릿을 사용하는 경우에만 지원됩니다.

용량 최적화 할당 전략을 사용하는 JSON 구성의 예제는 다음을 참조하세요.

- EC2 플릿 - 예제 8: 용량 최적화 플릿에서 스팟 인스턴스 시작

- 스팟 플릿 - 예제 9: 용량 최적화 플릿에서 스팟 인스턴스 시작

용량 최적화 우선 할당 전략을 사용하는 JSON 구성의 예제는 다음을 참조하세요.

- EC2 플릿 - 예제 9: 용량 최적화 플릿에서 우선순위를 사용하여 스팟 인스턴스 시작
- 스팟 플릿 - 예제 10: 용량 최적화 플릿에서 우선순위를 사용하여 스팟 인스턴스 시작

시간이 유동적이고 용량 가용성이 중요하지 않은 워크로드의 경우

플릿이 작거나 짧은 시간 동안 실행될 경우, 가격 용량 최적화를 사용하여 가용성을 고려하면서 비용 절감 효과를 극대화할 수 있습니다.

플릿 규모가 크거나 장시간 실행되는 경우

플릿이 크거나 장시간 실행될 경우 다각화 전략을 사용하여 여러 풀에 걸쳐 스팟 인스턴스를 분산하여 플릿의 가용성을 높일 수 있습니다. 예를 들어 플릿이 풀 10개와 인스턴스 100개의 목표 용량을 지정하는 경우 플릿은 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. 다각화 전략을 사용하면 플릿은 온디맨드 가격보다 높거나 이 가격과 동일한 스팟 가격의 풀로 스팟 인스턴스를 시작하지 않습니다.

스팟 인스턴스의 목표 용량 유지

스팟 가격 또는 스팟 용량 풀의 가용 용량 변화로 인해 스팟 인스턴스가 종료된 후 `maintain` 유형의 플릿은 대체 스팟 인스턴스를 시작합니다. 이 할당 전략은 다음과 같이 대체 인스턴스를 시작할 풀을 결정합니다.

- 가격 용량 최적화 할당 전략을 사용하는 경우, 플릿은 스팟 인스턴스 용량 가용성이 가장 높은 풀에서 대체 인스턴스를 시작하면서 동시에 가격도 고려합니다. 즉, 용량 가용성이 높으면서 가격이 가장 낮은 풀을 식별합니다.
- 할당 전략이 용량 최적화인 경우, 플릿은 스팟 인스턴스 용량의 가용성이 가장 높은 풀에서 대체 인스턴스를 시작합니다.
- 할당 전략이 다각화인 경우 플릿은 나머지 풀에 대체 스팟 인스턴스를 배포합니다.

온디맨드 용량에 대한 인스턴스 유형 우선순위 지정

EC2 플릿 또는 스팟 플릿이 온디맨드 용량을 채우려고 시도하는 경우 기본적으로 최저 가격의 인스턴스 유형을 먼저 시작합니다. 온디맨드 할당 전략이 우선순위로 설정된 경우 플릿은 온디맨드 용량을 충

족할 때 먼저 사용할 인스턴스 유형을 우선순위를 사용하여 결정합니다. 시작 템플릿 재정의에 우선순위를 할당하고 우선순위가 가장 높은 것을 먼저 시작합니다.

예: 인스턴스 유형 우선순위

예를 들어, 서로 다른 인스턴스 유형을 각각 지닌 3개의 시작 템플릿 재정의를 구성했다고 가정해 보겠습니다.

인스턴스 유형에 대한 온디맨드 요금 가격 범위. 다음은 이 예제에서 사용된 인스턴스 유형이며, 가격이 가장 낮은 인스턴스 유형부터 가격 순서대로 나열되어 있습니다.

- `m4.large` - 가장 낮은 가격
- `m5.large`
- `m5a.large`

우선순위를 사용해 순서를 결정하지 않는 경우 플릿은 가격이 가장 낮은 인스턴스 유형으로 시작하여 온디맨드 용량을 채웁니다.

하지만 가장 먼저 사용하려는 `m5.large` 예약 인스턴스를 사용하지 않았다고 가정해 보겠습니다. 다음과 같이 인스턴스 유형이 우선순위에 따라 사용되도록 시작 템플릿 재정의 우선순위를 설정할 수 있습니다.

- `m5.large` - 우선순위 1
- `m4.large` - 우선순위 2
- `m5a.large` - 우선순위 3

EC2 플릿 및 스팟 플릿의 용량 리밸런싱을 사용하여 위험이 있는 스팟 인스턴스 교체

용량 리밸런싱을 통해 EC2 플릿 또는 스팟 플릿은 중단 위험이 있는 스팟 인스턴스를 사전에 교체하여 원하는 스팟 용량을 유지할 수 있습니다. 스팟 인스턴스의 중단 위험이 높아질 때 Amazon EC2는 재조정 권고를 전송합니다. 용량 리밸런싱이 활성화된 경우 재조정 권고에 따라 위험이 있는 인스턴스가 중단되기 전에 새 스팟 인스턴스가 시작됩니다.

용량 리밸런싱을 사용하면 실행 중인 인스턴스가 Amazon EC2에 의해 중단되기 전에 미리 새 스팟 인스턴스로 플릿을 보강할 수 있으므로 워크로드 가용성을 유지하는 데 도움이 됩니다.

대체 스팟 인스턴스를 시작하도록 용량 리밸런싱을 사용하여 EC2 플릿 구성

create-fleet 명령과 MaintenanceStrategies 구조의 관련 파라미터를 사용합니다. 예제 JSON 구성은 예제 7: 대체 스팟 인스턴스를 시작하도록 용량 리밸런싱 구성의 내용을 참조하세요.

대체 스팟 인스턴스를 시작하도록 용량 리밸런싱을 사용하여 스팟 플릿 구성

Amazon EC2 콘솔 또는 AWS CLI를 사용하여 용량 리밸런싱을 구성할 수 있습니다.

(콘솔) 스팟 플릿을 생성할 때 용량 재조정 확인란을 선택합니다. 자세한 내용은 정의된 파라미터를 사용하여 스팟 플릿 요청 생성의 6.d 단계를 참조하세요.

(AWS CLI) request-spot-fleet 명령과 SpotMaintenanceStrategies 구조의 관련 파라미터를 사용합니다. 예제 JSON 구성은 예제 8: 대체 스팟 인스턴스를 시작하도록 용량 리밸런싱 구성의 내용을 참조하세요.

주제

- 제한 사항
- 구성 옵션
- 고려 사항

제한 사항

- 용량 재분배는 maintain 유형의 플릿에만 사용할 수 있습니다.
- 플릿이 실행 중인 경우 용량 리밸런싱 설정을 수정할 수 없습니다. 용량 리밸런싱 설정을 변경하려면 플릿을 삭제하고 새 플릿을 생성해야 합니다.

구성 옵션

EC2 플릿 및 스팟 플릿에 대한 ReplacementStrategy는 다음 두 값을 지원합니다.

launch-before-terminate

Amazon EC2는 새로운 대체 스팟 인스턴스가 시작된 후 리밸런싱 알림을 받는 스팟 인스턴스를 종료합니다. launch-before-terminate를 지정하면 termination-delay에 대한 값도 함께 지정해야 합니다. 새 대체 인스턴스가 시작된 후 Amazon EC2는 termination-delay 기간 동안 기다린 다음 이전 인스턴스를 종료합니다. termination-delay에 대해 최솟값은 120초(2분)이고 최댓값은 7,200초(2시간)입니다.

인스턴스 종료 절차가 완료되는 데 걸리는 시간을 예측할 수 있는 경우에만 launch-before-terminate를 사용하는 것이 좋습니다. 이렇게 하면 종료 절차가 완료된 후에만 이전 인스턴스가

종료됩니다. Amazon EC2는 `termination-delay` 전 2분 경고를 통해 기존 인스턴스를 중단할 수 있습니다.

중단 위험이 높은 교체 스팟 인스턴스를 피하기 위해 `launch-before-terminate`와 함께 `lowest-price`(EC2 플릿) 또는 `lowestPrice`(스팟 플릿) 할당 전략을 사용하지 않는 것이 좋습니다.

`launch`

Amazon EC2는 기존 스팟 인스턴스에 대해 리밸런싱 알림이 전송될 때 대체 스팟 인스턴스를 시작합니다. Amazon EC2는 리밸런싱 알림을 수신하는 인스턴스를 종료하지 않습니다. 이전 인스턴스를 종료하거나 실행 중인 상태로 둘 수 있습니다. 두 인스턴스가 실행되는 동안에는 두 인스턴스에 대해 요금이 청구됩니다.

고려 사항

용량 리밸런싱을 위해 EC2 플릿 또는 스팟 플릿을 구성하는 경우 다음을 고려하세요.

요청에 가능한 한 많은 스팟 용량 풀 제공

여러 인스턴스 유형 및 가용 영역을 사용하도록 플릿을 구성합니다. 이렇게 하면 다양한 스팟 용량 풀의 스팟 인스턴스를 유연하게 시작할 수 있습니다. 자세한 내용은 인스턴스 유형 및 가용 영역에 대한 유연성 유지 단원을 참조하십시오.

대체 스팟 인스턴스의 중단 위험 증가 방지

서비스 중단 위험이 높아지는 것을 방지하려면 `capacity-optimized` 또는 `capacity-optimized-prioritized` 할당 전략을 사용하는 것이 좋습니다. 이러한 전략을 통해 대체 스팟 인스턴스가 최적의 스팟 용량 풀에서 시작되므로 가까운 시일 내에 중단될 가능성이 줄어듭니다. 자세한 내용은 가격 및 용량 최적화 할당 전략 사용 단원을 참조하십시오.

`lowest-price` 할당 전략을 사용하는 경우, 교체 스팟 인스턴스가 중단될 위험이 높아질 수도 있습니다. 이는 교체 스팟 인스턴스가 시작된 직후 중단될 가능성이 높더라도 Amazon EC2는 항상 그 순간에 가용 용량이 있는 최저 가격의 풀에서 인스턴스를 시작하기 때문입니다.

Amazon EC2는 가용성이 동일하거나 더 나은 경우에만 새 인스턴스를 시작함

용량 리밸런싱의 목표 중 하나는 스팟 인스턴스의 가용성을 개선하는 것입니다. 기존 스팟 인스턴스에 대한 리밸런싱 권장 사항이 있는 경우, Amazon EC2는 새 인스턴스가 기존 인스턴스와 동일하거나 더 나은 가용성을 제공하는 경우에만 새 인스턴스를 시작합니다. 새 인스턴스의 중단 위험이 기존 인스턴스보다 더 높은 경우 Amazon EC2는 새 인스턴스를 시작하지 않습니다. 하지만 Amazon EC2는 스팟 용량 풀을 계속 평가하여 가용성이 향상되면 새 인스턴스를 시작합니다.

Amazon EC2가 사전에 새 인스턴스를 시작하지 않은 채로 기존 인스턴스가 중단될 수 있습니다. 이 경우 Amazon EC2는 새 인스턴스가 중단될 위험이 높은지 여부에 관계없이 새 인스턴스를 시작하려고 시도합니다.

용량 리밸런싱은 스팟 인스턴스의 간섭 속도를 증가시키지 않음

용량 리밸런싱을 활성화하면 스팟 인스턴스 중단 속도(Amazon EC2가 용량을 회수해야 할 때 회수되는 스팟 인스턴스의 수)가 증가하지 않습니다. 그러나 용량 리밸런싱에서 중단 위험이 있는 인스턴스를 탐지할 경우, Amazon EC2가 즉시 새로운 인스턴스를 시작하려고 시도합니다. 위험한 인스턴스가 중단된 후 Amazon EC2가 시작되기를 기다리면 새 인스턴스를 시작하는 것보다 더 많은 인스턴스가 교체될 수 있습니다.

용량 리밸런싱을 활성화하면 더 많은 인스턴스를 교체하게 될 수도 있지만, 인스턴스가 중단되기 전까지 조치를 취할 시간 여유가 늘어나 미리 대비할 수 있어 도움이 됩니다. 스팟 인스턴스 중단 알림이 오면 일반적으로 최대 2분 이내에 인스턴스를 적절히 종료해야 합니다. 용량 리밸런싱이 미리 새 인스턴스를 시작할 경우, 기존 프로세스에서 위험한 인스턴스를 완료하여 인스턴스 종료 절차를 시작할 기회가 커지므로 위험한 인스턴스에 새 작업이 예약되지 않도록 방지할 수 있습니다. 또한, 새로 시작한 인스턴스가 애플리케이션을 가져가도록 준비를 시작할 수도 있습니다. 용량 리밸런싱에서 미리 인스턴스를 교체하기 때문에 적절한 연속성을 누릴 수 있습니다.

용량 리밸런싱을 사용할 때의 위험과 장점을 보여주는 이론적 예시로써 다음의 시나리오를 보여드리겠습니다.

- 오후 2:00 – 인스턴스-A에 대한 리밸런싱 권고를 받고, Amazon EC2가 즉시 교체 인스턴스-B를 시작하려고 시도하므로 종료 절차를 시작할 시간이 마련됩니다.*
- 오후 2:30 – 인스턴스-B에 대한 리밸런싱 권고를 받고 인스턴스-C로 교체하므로 종료 절차를 시작할 시간이 마련됩니다.*
- 오후 2:32 – 용량 리밸런싱을 활성화하지 않았고 인스턴스-A에 대한 스팟 인스턴스 간섭 알림을 오후 2:32에 받지 않은 경우, 조치를 취할 시간이 2분에 불과하지만 인스턴스-A는 이 시점까지 계속 실행 중입니다.

* `launch-before-terminate`를 지정한 경우, 교체 인스턴스가 실행된 후 Amazon EC2가 위험한 인스턴스를 종료합니다.

Amazon EC2는 이행 용량이 목표 용량의 두 배가 되기 전까지 새 대체 스팟 인스턴스를 시작할 수 있음

플릿에 용량 리밸런싱이 구성된 경우 리밸런싱 권고를 수신하는 모든 스팟 인스턴스에 대해 새로운 대체 스팟 인스턴스를 시작합니다. 재분배 권고를 수신하는 스팟 인스턴스는 더 이상 이행 용량의 일부로 계산되지 않습니다. 대체 전략에 따라 Amazon EC2는 사전 구성된 종료 지연 후에 인스턴스

를 종료하거나 실행 중인 상태로 둡니다. 그러면 인스턴스에서 리밸런싱 작업을 수행할 수 있습니다.

플릿이 목표 용량의 두 배에 도달하면 대체 인스턴스 자체가 리밸런싱 권고를 수신하더라도 새 대체 인스턴스의 시작이 중지됩니다.

예를 들어 100개의 스팟 인스턴스를 목표 용량으로 하는 플릿을 생성할 수 있습니다. 모든 스팟 인스턴스가 재분배 권고를 수신하고 Amazon EC2가 100개의 대체 스팟 인스턴스를 시작합니다. 그러면 이행된 스팟 인스턴스의 수가 200으로 증가하여 목표 용량의 두 배가 됩니다. 대체 인스턴스 중 일부는 리밸런싱 권고를 수신하지만, 플릿이 목표 용량의 두 배를 초과할 수 없기 때문에 더 이상 대체 인스턴스가 시작되지 않습니다.

인스턴스가 실행되는 동안에는 모든 인스턴스에 대해 요금이 청구됩니다.

리밸런싱 권고를 수신하는 스팟 인스턴스를 종료하는 스팟 플릿을 구성하는 것이 권장됨

용량 리밸런싱을 플릿을 구성하는 경우 인스턴스 종료 절차가 완료되는 데 걸리는 시간을 예측할 수 있는 경우에만 적절한 종료 지연으로 `launch-before-terminate`를 선택합니다. 이렇게 하면 종료 절차가 완료된 후에만 이전 인스턴스가 종료됩니다.

재분배를 위해 권장되는 인스턴스를 직접 종료하도록 선택하는 경우 플릿의 스팟 인스턴스에 수신되는 재분배 권고 신호를 모니터링하는 것이 좋습니다. 신호를 모니터링하면 Amazon EC2에서 중단하기 전에 영향을 받는 인스턴스에 대해 리밸런싱 작업을 신속하게 수행한 다음 수동으로 종료할 수 있습니다. 인스턴스를 종료하지 않으면 인스턴스가 실행되는 동안 계속 비용을 지불하게 됩니다. Amazon EC2는 리밸런싱 권고를 수신하는 인스턴스를 자동으로 종료하지 않습니다.

Amazon EventBridge 또는 인스턴스 메타데이터를 사용하여 알림을 설정할 수 있습니다. 자세한 내용은 리밸런싱 권고 신호 모니터링 단원을 참조하십시오.

플릿은 확장 또는 축소 중에 이행 용량을 계산할 때 리밸런싱 권고를 수신하는 인스턴스를 계산하지 않음

플릿에 용량 리밸런싱이 구성되어 있고 목표 용량을 축소 또는 확장으로 변경하는 경우 다음과 같이 플릿은 리밸런싱으로 표시된 인스턴스를 이행 용량의 일부로 계산하지 않습니다.

- 스케일 인 – 원하는 목표 용량을 줄이는 경우 Amazon EC2는 원하는 용량에 도달할 때까지 리밸런싱으로 표시되지 않은 인스턴스를 종료합니다. 리밸런싱으로 표시된 인스턴스는 이행 용량으로 계산되지 않습니다.

  예를 들어 스팟 인스턴스 100개의 목표 용량으로 플릿을 생성하는 경우 10개의 인스턴스에서 리밸런싱 권고를 수신하면 Amazon EC2가 10개의 새로운 대체 인스턴스를 시작하므로 결과적으로

이행 용량은 110개 인스턴스가 됩니다. 그런 다음 목표 용량을 50으로 줄이면(스케일 인) 이행 용량은 실제로 60개 인스턴스가 됩니다. Amazon EC2가 리밸런싱으로 표시된 10개 인스턴스가 종료되지 않기 때문입니다. 이러한 인스턴스를 수동으로 종료하거나 실행 상태로 둘 수 있습니다.

- 스케일 아웃 – 원하는 목표 용량을 늘리면 원하는 용량에 도달할 때까지 Amazon EC2가 새 인스턴스를 시작합니다. 리밸런싱으로 표시된 인스턴스는 이행 용량으로 계산되지 않습니다.

  예를 들어 스팟 인스턴스 100개의 목표 용량으로 플릿을 생성하는 경우 10개의 인스턴스에서 리밸런싱 권고를 수신하면 플릿이 10개의 새로운 대체 인스턴스를 시작하므로 결과적으로 이행 용량은 110개 인스턴스가 됩니다. 그런 다음 목표 용량을 200으로 늘리면(확장) 이행 용량은 실제로 210개 인스턴스가 됩니다. 플릿에서 리밸런싱으로 표시된 10개 인스턴스가 목표 용량의 일부로 계산되지 않기 때문입니다. 이러한 인스턴스를 수동으로 종료하거나 실행 상태로 둘 수 있습니다.

용량 예약을 사용하여 EC2 플릿의 온디맨드 용량 예약

온디맨드 용량 예약을 사용하면 특정 가용 영역의 온디맨드 인스턴스에 대해 원하는 기간만큼 컴퓨팅 용량을 예약할 수 있습니다. 먼저 온디맨드 인스턴스를 시작할 때 용량 예약을 사용하도록 EC2 플릿을 구성할 수 있습니다.

온디맨드 용량 예약은 요청 유형이 `instant`로 설정된 EC2 플릿에만 사용할 수 있습니다.

용량 예약은 `open` 또는 `targeted`로 구성됩니다. EC2 플릿은 다음과 같이 온디맨드 인스턴스를 `open` 또는 `targeted` 용량 예약으로 시작할 수 있습니다.

- 용량 예약이 `open`이면 일치하는 속성이 있는 온디맨드 인스턴스는 예약 용량으로 자동 실행됩니다.
- 용량 예약이 `targeted`이면 인스턴스를 예약 용량으로 실행하도록 대상을 구체적으로 지정해야 합니다. 이 기능은 특정 용량 예약을 사용하거나 특정 용량 예약을 사용할 시기를 제어할 때 유용합니다.

EC2 플릿의 `targeted` 용량 예약을 사용하는 경우, 목표 온디맨드 용량을 처리할 수 있는 충분한 용량 예약이 있어야 합니다. 그렇게 하지 않으면 시작이 실패합니다. 시작 실패를 방지하려면 `targeted` 용량 예약을 리소스 그룹에 추가한 후 리소스 그룹을 대상으로 지정합니다. 리소스 그룹에 충분한 용량 예약이 필요하지 않습니다. 목표 온디맨드 용량이 처리되기 전에 용량 예약이 부족하면 플릿이 나머지 목표 용량을 일반 온디맨드 용량으로 시작할 수 있습니다.

EC2 플릿에서 용량 예약을 사용하려면

1. 플릿을 `instant` 유형으로 구성합니다. 다른 유형의 플릿에는 용량 예약을 사용할 수 없습니다.

2. 용량 예약에 대한 사용 전략을 `use-capacity-reservations-first`로 구성합니다.
3. 시작 템플릿의 용량 예약에서 열기(Open) 또는 그룹별 대상(Target by group)을 선택합니다. 그룹별 대상(Target by group)을 선택한 경우 용량 예약 리소스 그룹 ID를 지정합니다.

플릿이 온디맨드 용량을 처리하려고 할 때 여러 인스턴스 풀에 일치하는 용량 예약이 사용되지 않는 것으로 확인되면 온디맨드 할당 전략(`lowest-price` 또는 `prioritized`)에 따라 온디맨드 인스턴스를 시작할 풀을 결정합니다.

관련 리소스

- 온디맨드 용량을 처리하기 위해 용량 예약을 사용하도록 플릿을 구성하는 방법의 CLI 예제는 EC2 플릿에 대한 CLI 구성 예제 섹션, 특히 예제 5~7을 참조하세요.
- 용량 예약을 생성하고, 플릿에서 용량 예약을 사용하고, 남은 용량 예약 수를 확인하는 단계를 안내하는 튜토리얼은 튜토리얼: 대상으로 지정된 용량 예약을 사용하여 온디맨드 인스턴스를 시작하도록 EC2 플릿 구성 섹션을 참조하세요.
- 용량 예약 구성에 대한 자세한 내용은 EC2 온디맨드 용량 예약으로 컴퓨팅 용량 예약 및 온디맨드 용량 예약 FAQ를 참조하세요.

EC2 플릿 사용

EC2 플릿 사용을 시작하려면 총 목표 용량, 온디맨드 용량, 스팟 용량 및 플릿의 인스턴스 구성을 지정하는 시작 템플릿을 포함하는 요청을 생성합니다. 추가 파라미터를 지정하거나 플릿에서 기본값을 사용하도록 할 수도 있습니다. 또한 플릿을 생성할 때 플릿 요청과 해당 인스턴스 및 볼륨에 태그를 지정할 수 있습니다.

플릿은 가용 용량이 있을 때 온디맨드 인스턴스를 시작하며, 최고 가격이 스팟 가격을 초과하고 가용 용량이 있을 때 스팟 인스턴스를 시작합니다.

플릿이 시작되면 플릿 요청, 플릿의 인스턴스, 모든 플릿 이벤트를 설명할 수 있습니다. 필요에 따라 추가 태그를 할당할 수도 있습니다.

총 대상 용량 등의 플릿 파라미터를 변경해야 하는 경우 용량을 유지하도록 구성된 플릿만 수정하면 됩니다. 일회성 요청이 제출된 후에는 해당 요청의 용량을 수정할 수 없습니다.

플릿 요청은 요청이 만료되거나 삭제될 때까지 활성 상태로 유지됩니다. 플릿 요청을 삭제할 때 인스턴스를 종료하거나 실행 중인 상태로 둘 수 있습니다 실행 중인 상태로 둘 경우 온디맨드 인스턴스는 사용자가 종료할 때까지 실행되고 스팟 인스턴스는 중단되거나 사용자가 종료할 때까지 실행됩니다.

주제

- EC2 집합 요청 상태
- EC2 집합 사전 조건
- EC2 집합 생성
- 신규 또는 기존 EC2 플릿 요청과 이를 통해 시작되는 인스턴스 및 볼륨 태그 지정
- EC2 플릿, 해당 인스턴스 및 해당 이벤트 설명
- EC2 집합 수정
- EC2 플릿 요청 및 플릿의 인스턴스 삭제

EC2 집합 요청 상태

EC2 플릿 요청 상태는 다음 중 하나일 수 있으며, 각 상태는 요청 수명 주기 및 인스턴스 관리의 서로 다른 단계를 나타냅니다.

EC2 집합 요청은 다음 상태 중 하나일 수 있습니다.

`submitted`

EC2 집합 요청을 평가 중이며 Amazon EC2에서 목표 개수의 인스턴스를 시작하기 위해 준비 중입니다. 요청이 플릿 제한을 초과하면 즉시 해당 요청이 삭제됩니다.

`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` 플릿 유형만 수정할 수 있습니다. 이 상태는 다른 요청 유형에는 적용되지 않습니다.

`deleted_running`

EC2 플릿 요청이 삭제되었고 추가 스팟 인스턴스를 시작하지 않습니다. 수동으로 중단되거나 종료될 때까지 기존 인스턴스가 계속 실행됩니다. 그 요청은 모든 인스턴스가 중단 또는 종료될 때까지 계속 이 상태로 유지됩니다. EC2 집합 요청이 삭제된 후에는 `maintain` 또는 `request` 유형의 EC2 집합에서만 인스턴스를 실행할 수 있습니다. 삭제된 `instant` 플릿에서 인스턴스를 실행하는 것은 지원되지 않습니다. 이 상태는 `instant` 플릿에는 적용되지 않습니다.

`deleted_terminating`

EC2 집합 요청이 삭제되었고 해당 인스턴스를 종료하는 중입니다. 그 요청은 모든 인스턴스가 종료될 때까지 계속 이 상태로 유지됩니다.

`deleted`

EC2 집합 요청이 삭제되었고 실행 중인 인스턴스가 없습니다. 인스턴스가 종료되고 2일 후 요청이 삭제됩니다.

EC2 집합 사전 조건

EC2 집합을 생성하려면 다음 사전 조건이 갖춰져야 합니다.

- 시작 템플릿
- EC2 집합의 서비스 연결 역할
- 암호화된 AMI 및 EBS 스냅샷에 사용할 고객 관리형 키에 대한 액세스 권한 부여
- EC2 플릿 사용자의 권한

시작 템플릿

시작 템플릿은 인스턴스 유형과 가용 영역 등 시작할 인스턴스에 대한 구성 정보를 지정합니다. 시작 템플릿에 대한 자세한 내용은 Amazon EC2 시작 템플릿에 인스턴스 시작 파라미터 저장의 내용을 참조하세요.

EC2 집합의 서비스 연결 역할

`AWSServiceRoleForEC2Fleet` 역할은 EC2 플릿에 사용자 대신 인스턴스를 요청, 시작, 종료 및 태깅할 수 있는 권한을 부여합니다. Amazon EC2는 이 서비스 연결 역할을 사용하여 다음 작업을 완료합니다.

- `ec2:RunInstances` – 인스턴스를 시작합니다.
- `ec2:RequestSpotInstances` – 스팟 인스턴스 요청.
- `ec2:TerminateInstances` – 인스턴스를 종료합니다.
- `ec2:DescribeImages` - 인스턴스에 대한 Amazon Machine Image(AMI) 설명
- `ec2:DescribeInstanceStatus` - 인스턴스 상태 설명
- `ec2:DescribeSubnets` - 인스턴스의 서브넷 설명
- `ec2:CreateTags` – EC2 집합, 인스턴스 및 볼륨에 태그를 추가합니다.

AWS CLI 또는 API를 사용하여 EC2 플릿을 생성하려면 먼저 이 역할이 있어야 합니다.

> **Note**
>
> `instant` EC2 집합에는 이 역할이 필요하지 않습니다.

역할을 생성하려면 다음과 같이 IAM 콘솔을 사용하세요.

EC2 집합에 대한 AWSServiceRoleForEC2Fleet 역할 생성

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 역할을 선택합니다.
3. 역할 생성(Create role)을 선택합니다.
4. 신뢰할 수 있는 엔터티 선택 페이지에서 다음을 수행합니다.
   a. 신뢰할 수 있는 엔터티 유형에서 AWS 서비스를 선택합니다.
   b. 사용 사례에서 서비스 또는 사용 사례로 EC2 - 플릿을 선택합니다.

      > **Tip**
      >
      > EC2 - 플릿을 선택해야 합니다. EC2를 선택하면 EC2 - 플릿 사용 사례가 사용 사례 목록에 나타나지 않습니다. EC2 - 플릿 사용 사례는 필요한 IAM 권한으로 정책을 자동으로 생성하고 AWSServiceRoleForEC2Fleet을 역할 이름으로 제안합니다.

   c. 다음을 선택합니다.
5. 권한 추가 페이지에서 다음을 선택합니다.
6. 이름 지정, 검토 및 생성 페이지에서 역할 생성을 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 역할을 참조하세요.

암호화된 AMI 및 EBS 스냅샷에 사용할 고객 관리형 키에 대한 액세스 권한 부여

EC2 플릿에 암호화된 AMI 또는 암호화된 Amazon EBS 스냅샷을 지정하고 암호화에 AWS KMS 키를 사용하는 경우 Amazon EC2에서 자동으로 인스턴스를 시작하려면 고객 관리형 키를 사용할 권한을

AWSServiceRoleForEC2Fleet 역할에 부여해야 합니다. 이렇게 하려면 다음 절차에 표시된 바와 같이 고객 관리형 키에 대한 권한 부여를 추가해야 합니다.

권한을 제공할 때 권한 부여는 키 정책을 대체합니다. 자세한 내용은 AWS Key Management Service 개발자 안내서에서 권한 부여 사용 및 AWS KMS의 키 정책 사용을 참조하세요.

AWSServiceRoleForEC2Fleet 역할에 고객 관리형 키를 사용할 수 있는 권한을 부여하려면

- create-grant 명령을 사용하여 고객 관리형 키에 대한 권한 부여를 추가하고 허용된 작업을 수행할 수 있는 권한이 부여된 보안 주체(AWSServiceRoleForEC2Fleet 서비스 연결 역할)를 지정합니다. 고객 관리형 키는 `key-id` 파라미터와 고객 관리형 키의 ARN으로 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 플릿 사용자의 권한

EC2 플릿을 생성하거나 관리하는 사용자에게 필요한 권한을 부여해야 합니다.

EC2 플릿에 대한 정책 생성

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 Policies를 선택합니다.
3. [Create policy]를 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":"arn:aws:iam::123456789012:role/DevTeam*"
        }
    ]
}
```

`ec2:*`는 모든 Amazon EC2 API 작업을 호출할 수 있는 사용자 권한을 부여합니다. 사용자를 특정 Amazon EC2 API 작업으로 제한하려면 해당 작업을 대신 지정하세요.

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) 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. 액세스 권한을 제공하려면 사용자, 그룹 또는 역할에 권한을 추가하세요:

   - AWS IAM Identity Center의 사용자 및 그룹:

권한 세트를 생성합니다. AWS IAM Identity Center 사용 설명서의 권한 세트 생성의 지침을 따릅니다.

- 보안 인증 공급자를 통해 IAM에서 관리되는 사용자:

  ID 페더레이션을 위한 역할을 생성합니다. IAM 사용 설명서의 Create a role for a third-party identity provider (federation)의 지침을 따릅니다.

- IAM 사용자:
  - 사용자가 맡을 수 있는 역할을 생성합니다. IAM 사용 설명서에서 Create a role for an IAM user의 지침을 따릅니다.
  - (권장되지 않음)정책을 사용자에게 직접 연결하거나 사용자를 사용자 그룹에 추가합니다. IAM 사용 설명서에서 사용자(콘솔)에 권한 추가의 지침을 따르세요.

EC2 집합 생성

EC2 플릿을 생성하려면 create-fleet 명령을 사용하여 JSON 파일에서 플릿 구성을 정의하고 파일을 참조합니다. JSON 파일에서 플릿의 총 목표 용량, 스팟 인스턴스 및 온디맨드 인스턴스에 대한 개별 목표 용량, AMI, 인스턴스 유형, 서브넷 또는 가용 영역, 하나 이상의 보안 그룹과 같은 구성을 정의하는 시작 템플릿을 지정해야 합니다. 시작 템플릿 구성을 재정의하는 파라미터, EC2 용량 풀에서 스팟 인스턴스 및 온디맨드 인스턴스를 선택하기 위한 할당 전략, 플릿에 대해 지불하려는 최대 금액 등의 추가 구성을 지정할 수도 있습니다. 자세한 내용은 EC2 플릿 또는 스팟 플릿에 대한 구성 옵션 섹션을 참조하세요.

EC2 플릿은 가용 용량이 있을 때 온디맨드 인스턴스를 시작하고, 최고 가격이 스팟 가격을 초과하고 가용 용량이 있을 때 스팟 인스턴스를 시작합니다.

플릿에 스팟 인스턴스가 포함되어 있고 유형이 `maintain`인 경우 Amazon EC2에서 스팟 인스턴스가 중단되어도 플릿 목표 용량을 유지하려고 합니다.

EC2 집합 제한 사항

EC2 집합에는 다음과 같은 제한이 적용됩니다.

- EC2 플릿 생성은 Amazon EC2 API, AWS CLI, AWS SDK, AWS CloudFormation을 통해서만 사용 가능합니다.
- EC2 플릿 요청을 AWS 리전으로 확장할 수 없습니다. 리전마다 따로 EC2 집합을 생성해야 합니다.
- EC2 집합 요청으로 동일한 가용 영역의 서로 다른 서브넷을 확장할 수 없습니다.

EC2 집합 생성

EC2 플릿을 사용하여 인스턴스 플릿을 시작하려면 플릿 요청에서 다음 파라미터만 지정하면 되고, 플릿은 다른 파라미터에 대한 기본값을 사용합니다.

- `LaunchTemplateId` 또는 `LaunchTemplateName` - 사용할 시작 템플릿을 지정합니다(인스턴스 유형 및 가용 영역 등 시작할 인스턴스에 대한 파라미터 포함).
- `TotalTargetCapacity` - 플릿의 총 목표 용량을 지정합니다.
- `DefaultTargetCapacityType` - 기본 구매 옵션이 온디맨드 또는 스팟인지 여부를 지정합니다.

시작 템플릿에 지정된 파라미터를 재정의하기 위해 하나 이상의 재정의를 지정할 수 있습니다. 각각은 인스턴스 유형, 가용 영역, 서브넷, 최고 가격에 따라 달라질 수 있으며 다른 가중치 용량을 포함할 수 있습니다. 인스턴스 유형을 지정하는 대신 인스턴스에 있어야 하는 속성을 지정하면 Amazon EC2는 해당 속성으로 모든 인스턴스 유형을 식별할 수 있습니다. 자세한 내용은 EC2 플릿 또는 스팟 플릿의 인스턴스 유형 선택에 대한 속성 지정 단원을 참조하세요.

`instant` 유형의 EC2 플릿의 경우 AMI ID 대신 Systems Manager 파라미터를 지정할 수 있습니다. 재정의 또는 시작 템플릿에서 Systems Manager 파라미터를 지정할 수 있습니다. 자세한 내용은 AMI ID 대신 Systems Manager 파라미터 사용 섹션을 참조하세요.

JSON 파일의 플릿 파라미터를 지정할 수 있습니다. 지정할 수 있는 모든 파라미터에 대한 내용은 모든 EC2 플릿 구성 옵션 보기의 내용을 참조하세요.

플릿 구성 예제는 EC2 플릿에 대한 CLI 구성 예제의 내용을 참조하세요.

현재 콘솔에서 EC2 플릿 생성은 지원되지 않습니다.

EC2 플릿 생성

create-fleet 명령을 사용하여 플릿을 생성하고 플릿 구성 파라미터를 포함하는 JSON 파일을 지정합니다.

```
aws ec2 create-fleet --cli-input-json file://file_name.json
```

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"
      ]
```

```
  ]
}
```

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": ""
    },
  ],
  "Instances": [
    {
      "LaunchTemplateAndOverrides": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
          "Version": "1"
        },
        "Overrides": {
          "InstanceType": "c5.large",
          "AvailabilityZone": "us-east-1a"
        }
      },
      "Lifecycle": "on-demand",
      "InstanceIds": [
        "i-1234567890abcdef0",
        "i-9876543210abcdef9"
      ]
  ]
```

```
}
```

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": ""
    },
    {
      "LaunchTemplateAndOverrides": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
          "Version": "1"
        },
        "Overrides": {
          "InstanceType": "c5.large",
          "AvailabilityZone": "us-east-1a",
        }
      },
      "Lifecycle": "on-demand",
      "ErrorCode": "InsufficientCapacity",
      "ErrorMessage": ""
    },
  ],
  "Instances": []
}
```

비정상 스팟 인스턴스를 교체하는 EC2 플릿 생성

EC2 집합은 분마다 플릿의 인스턴스 상태를 확인합니다. 인스턴스의 상태는 `healthy` 또는 `unhealthy`입니다.

EC2 집합은 Amazon EC2에서 제공하는 상태 확인을 사용하여 인스턴스의 상태를 판단합니다. 세 번의 연속 상태 확인에서 인스턴스 상태 또는 시스템 상태가 `unhealthy`인 경우, 해당 인스턴스의 상태는 `impaired`으로 확인됩니다. 자세한 내용은 Amazon EC2 인스턴스 상태 확인 섹션을 참조하세요.

플릿을 구성하여 비정상 스팟 인스턴스를 교체할 수 있습니다. `ReplaceUnhealthyInstances`를 `true`로 설정한 이후 스팟 인스턴스는 `unhealthy`로 보고될 때 교체됩니다. 플릿은 비정상 스팟 인스턴스가 교체되는 동안 최대 몇 분간 목표 용량을 밑돌 수 있습니다.

요구 사항

- 상태 확인 교체는 목표 용량을 유지하는 EC2 집합(`maintain` 유형 플릿)에만 지원되지만 `request` 또는 `instant` 유형 플릿에서는 지원되지 않습니다.
- 상태 확인 교체는 스팟 인스턴스에 대해서만 지원됩니다. 이 기능은 온디맨드 인스턴스에 대해 지원되지 않습니다.
- 비정상 인스턴스를 생성할 경우에만 이를 교체하도록 EC2 집합을 구성할 수 있습니다.
- 사용자는 `ec2:DescribeInstanceStatus` 작업을 호출할 권한이 있는 경우에만 상태 확인 대체를 사용할 수 있습니다.

비정상 스팟 인스턴스를 교체하도록 EC2 집합을 구성하려면

1. EC2 집합 생성에서 EC2 플릿 생성을 위한 정보를 사용합니다.
2. 비정상 스팟 인스턴스를 교체하도록 플릿을 구성하려면 JSON 파일에서 `ReplaceUnhealthyInstances`에 대해 `true`를 지정합니다.

모든 EC2 플릿 구성 옵션 보기

EC2 플릿 구성 파라미터의 전체 목록을 보려면 JSON 파일을 생성할 수 있습니다. 각 파라미터에 대한 설명은 create-fleet 섹션을 참조하세요.

가능한 모든 EC2 플릿 파라미터가 포함된 JSON 파일 생성

create-fleet(AWS CLI) 명령 및 `--generate-cli-skeleton` 파라미터를 사용하여 EC2 플릿 JSON 파일을 생성하고 파일에 출력을 향하게 하여 저장합니다.

```
aws ec2 create-fleet \
    --generate-cli-skeleton input > ec2createfleet.json
```

출력의 예시는 다음과 같습니다.

```
{
    "DryRun": true,
    "ClientToken": "",
    "SpotOptions": {
        "AllocationStrategy": "price-capacity-optimized",
        "MaintenanceStrategies": {
            "CapacityRebalance": {
                "ReplacementStrategy": "launch"
            }
        },
        "InstanceInterruptionBehavior": "hibernate",
        "InstancePoolsToUseCount": 0,
        "SingleInstanceType": true,
        "SingleAvailabilityZone": true,
        "MinTargetCapacity": 0,
        "MaxTotalPrice": ""
    },
    "OnDemandOptions": {
        "AllocationStrategy": "prioritized",
        "CapacityReservationOptions": {
            "UsageStrategy": "use-capacity-reservations-first"
        },
        "SingleInstanceType": true,
        "SingleAvailabilityZone": true,
        "MinTargetCapacity": 0,
        "MaxTotalPrice": ""
    },
    "ExcessCapacityTerminationPolicy": "termination",
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "",
                "LaunchTemplateName": "",
                "Version": ""
            },
            "Overrides": [
                {
                    "InstanceType": "r5.metal",
```

```
                    "MaxPrice": "",
                    "SubnetId": "",
                    "AvailabilityZone": "",
                    "WeightedCapacity": 0.0,
                    "Priority": 0.0,
                    "Placement": {
                        "AvailabilityZone": "",
                        "Affinity": "",
                        "GroupName": "",
                        "PartitionNumber": 0,
                        "HostId": "",
                        "Tenancy": "dedicated",
                        "SpreadDomain": "",
                        "HostResourceGroupArn": ""
                    },
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 0
                        },
                        "MemoryMiB": {
                            "Min": 0,
                            "Max": 0
                        },
                        "CpuManufacturers": [
                            "amd"
                        ],
                        "MemoryGiBPerVCpu": {
                            "Min": 0.0,
                            "Max": 0.0
                        },
                        "ExcludedInstanceTypes": [
                            ""
                        ],
                        "InstanceGenerations": [
                            "previous"
                        ],
                        "SpotMaxPricePercentageOverLowestPrice": 0,
                        "OnDemandMaxPricePercentageOverLowestPrice": 0,
                        "BareMetal": "included",
                        "BurstablePerformance": "required",
                        "RequireHibernateSupport": true,
                        "NetworkInterfaceCount": {
                            "Min": 0,
```

```
                        "Max": 0
                    },
                    "LocalStorage": "excluded",
                    "LocalStorageTypes": [
                        "ssd"
                    ],
                    "TotalLocalStorageGB": {
                        "Min": 0.0,
                        "Max": 0.0
                    },
                    "BaselineEbsBandwidthMbps": {
                        "Min": 0,
                        "Max": 0
                    },
                    "AcceleratorTypes": [
                        "inference"
                    ],
                    "AcceleratorCount": {
                        "Min": 0,
                        "Max": 0
                    },
                    "AcceleratorManufacturers": [
                        "amd"
                    ],
                    "AcceleratorNames": [
                        "a100"
                    ],
                    "AcceleratorTotalMemoryMiB": {
                        "Min": 0,
                        "Max": 0
                    }
                }
            }
        ]
    }
],
"TargetCapacitySpecification": {
    "TotalTargetCapacity": 0,
    "OnDemandTargetCapacity": 0,
    "SpotTargetCapacity": 0,
    "DefaultTargetCapacityType": "on-demand",
    "TargetCapacityUnitType": "memory-mib"
},
"TerminateInstancesWithExpiration": true,
```

```
    "Type": "instant",
    "ValidFrom": "1970-01-01T00:00:00",
    "ValidUntil": "1970-01-01T00:00:00",
    "ReplaceUnhealthyInstances": true,
    "TagSpecifications": [
        {
            "ResourceType": "fleet",
            "Tags": [
                {
                    "Key": "",
                    "Value": ""
                }
            ]
        }
    ],
    "Context": ""
}
```

신규 또는 기존 EC2 플릿 요청과 이를 통해 시작되는 인스턴스 및 볼륨 태그 지정

EC2 플릿 요청과 이를 통해 시작되는 인스턴스와 볼륨을 분류 및 관리하는 데 도움이 되도록 사용자 지정 메타데이터로 이 요청에 태그를 지정할 수 있습니다. EC2 집합 요청을 만들 때 또는 만든 후 요청에 태그를 지정할 수 있습니다. 마찬가지로 플릿에서 인스턴스와 볼륨이 시작될 때 또는 그 이후에 인스턴스와 볼륨에 태그를 할당할 수 있습니다.

플릿 요청에 태그를 지정할 때 플릿에서 시작한 인스턴스 및 볼륨에는 태그가 자동으로 지정되지 않습니다. 플릿에서 시작한 인스턴스 및 볼륨에 명시적으로 태그를 지정해야 합니다. 플릿 요청에만, 플릿에서 시작한 인스턴스에만, 플릿에서 시작한 인스턴스에 연결된 볼륨에만, 또는 모두에 태그를 할당하도록 선택할 수 있습니다.

> **Note**
>
> `instant` 플릿 유형의 경우 온디맨드 인스턴스 및 스팟 인스턴스에 연결된 볼륨에 태그를 지정할 수 있습니다. `request` 또는 `maintain` 플릿 유형의 경우 온디맨드 인스턴스에 연결된 볼륨에만 태그를 지정할 수 있습니다.

태그 작동 방식에 대한 자세한 내용은 Amazon EC2 리소스 태깅 섹션을 참조하세요.

사전 조건

사용자에게 리소스에 태그를 지정할 수 있는 권한을 부여합니다. 자세한 내용은 예: 태그 리소스 섹션을 참조하세요.

사용자에게 리소스에 태그를 지정할 수 있는 권한 부여

다음을 포함하는 IAM 정책을 만듭니다.

- `ec2:CreateTags` 작업 사용자에게 태그 생성 권한이 부여됩니다.
- `ec2:CreateFleet` 작업 사용자에게 EC2 플릿 요청 생성 권한이 부여됩니다.
- `Resource`의 경우 `"*"`를 지정하는 것이 좋습니다. 이를 통해 사용자는 모든 리소스 유형에 태그를 지정할 수 있습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "TagEC2FleetRequest",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags",
                "ec2:CreateFleet"
            ],
            "Resource": "*"
}
```

Important

`create-fleet` 리소스에 대한 리소스 수준 권한은 현재 지원되지 않습니다. `create-fleet`을 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:CreateFleet"
    ],
    "Resource": "arn:aws:ec2:us-east-1:111122223333:create-fleet/*"
}
```

액세스 권한을 제공하려면 사용자, 그룹 또는 역할에 권한을 추가하세요:

- AWS IAM Identity Center의 사용자 및 그룹:

  권한 세트를 생성합니다. AWS IAM Identity Center 사용 설명서의 권한 세트 생성의 지침을 따릅니다.
- 보안 인증 공급자를 통해 IAM에서 관리되는 사용자:

  ID 페더레이션을 위한 역할을 생성합니다. IAM 사용 설명서의 Create a role for a third-party identity provider (federation)의 지침을 따릅니다.
- IAM 사용자:
  - 사용자가 맡을 수 있는 역할을 생성합니다. IAM 사용 설명서에서 Create a role for an IAM user의 지침을 따릅니다.
  - (권장되지 않음)정책을 사용자에게 직접 연결하거나 사용자를 사용자 그룹에 추가합니다. IAM 사용 설명서에서 사용자(콘솔)에 권한 추가의 지침을 따르세요.

새 EC2 집합 요청에 태그를 지정하려면

EC2 집합 요청 생성 시 요청에 태그를 지정하려면 플릿을 생성하는 데 사용되는 JSON 파일에 키-값 페어를 지정하십시오. `ResourceType`의 값은 `fleet`이어야 합니다. 다른 값을 지정하면 집합 요청이 실패합니다.

EC2 집합에서 시작한 인스턴스 및 볼륨에 태그를 지정하려면

플릿이 시작하는 해당 인스턴스 및 볼륨에 태그를 지정하려면 EC2 집합 요청에서 참조되는 시작 템플릿에서 태그를 지정하십시오.

> **Note**
>
> `request` 또는 `maintain` 플릿 유형에서 시작한 스팟 인스턴스에 연결된 볼륨에는 태그를 지정할 수 없습니다.

기존 EC2 플릿 요청, 인스턴스 및 볼륨에 태그 지정

create-tags 명령을 사용하여 기존 리소스에 태그를 지정합니다.

```
aws ec2 create-tags \
```

```
    --resources fleet-12a34b55-67cd-8ef9-
ba9b-9208dEXAMPLE i-1234567890abcdef0 vol-1234567890EXAMPLE \
    --tags Key=purpose,Value=test
```

EC2 플릿, 해당 인스턴스 및 해당 이벤트 설명

EC2 플릿 구성, EC2 플릿의 인스턴스 및 EC2 플릿의 이벤트 기록을 설명할 수 있습니다.

업무

- EC2 플릿 설명
- EC2 플릿의 모든 인스턴스 설명
- EC2 플릿의 이벤트 기록 설명

EC2 플릿 설명

AWS CLI

EC2 플릿을 설명하려면

describe-fleets 명령을 사용합니다.

```
aws ec2 describe-fleets \
    --fleet-ids fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

출력의 예시는 다음과 같습니다.

```
{
    "Fleets": [
        {
            "ActivityStatus": "fulfilled",
            "CreateTime": "2022-02-09T03:35:52+00:00",
            "FleetId": "fleet-364457cd-3a7a-4ed9-83d0-7b63e51bb1b7",
            "FleetState": "active",
            "ExcessCapacityTerminationPolicy": "termination",
            "FulfilledCapacity": 2.0,
            "FulfilledOnDemandCapacity": 0.0,
            "LaunchTemplateConfigs": [
                {
                    "LaunchTemplateSpecification": {
                        "LaunchTemplateName": "my-launch-template",
```

```
                    "Version": "$Latest"
                }
            }
        ],
        "TargetCapacitySpecification": {
            "TotalTargetCapacity": 2,
            "OnDemandTargetCapacity": 0,
            "SpotTargetCapacity": 2,
            "DefaultTargetCapacityType": "spot"
        },
        "TerminateInstancesWithExpiration": false,
        "Type": "maintain",
        "ReplaceUnhealthyInstances": false,
        "SpotOptions": {
            "AllocationStrategy": "capacity-optimized",
            "InstanceInterruptionBehavior": "terminate"
        },
        "OnDemandOptions": {
            "AllocationStrategy": "lowestPrice"
        }
    }
]
}
```

PowerShell

EC2 플릿을 설명하려면

Get-EC2FleetList cmdlet을 사용합니다.

```
Get-EC2FleetList `
    -FleetId fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

EC2 플릿의 모든 인스턴스 설명

반환되는 실행 중 인스턴스 목록은 주기적으로 새로 고쳐지며 최신 상태가 아닐 수도 있습니다.

AWS CLI

지정된 EC2 플릿의 인스턴스를 설명하려면

describe-fleet-instances 명령을 사용합니다.

```
aws ec2 describe-fleet-instances \
    --fleet-id fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

출력의 예시는 다음과 같습니다.

```
{
    "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"
}
```

PowerShell

지정된 EC2 플릿의 인스턴스를 설명하려면

Get-EC2FleetInstanceList cmdlet을 사용합니다.

```
Get-EC2FleetInstanceList `
    -FleetId fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

EC2 플릿의 이벤트 기록 설명

이벤트 기록의 이벤트에 대한 자세한 내용은 EC2 집합 이벤트 유형 섹션을 참조하세요.

AWS CLI

지정된 EC2 플릿에 대한 이벤트를 설명하려면

describe-fleet-history 명령을 사용합니다.

```
aws ec2 describe-fleet-history \
    --fleet-id fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --start-time 2020-06-01T00:00:00Z
```

출력의 예시는 다음과 같습니다.

```
{
    "HistoryRecords": [
        {
            "EventInformation": {
                "EventSubType": "submitted"
            },
            "EventType": "fleetRequestChange",
            "Timestamp": "2020-09-01T18:26:05.000Z"
        },
        {
            "EventInformation": {
                "EventSubType": "active"
            },
            "EventType": "fleetRequestChange",
            "Timestamp": "2020-09-01T18:26:15.000Z"
        },
        {
            "EventInformation": {
                "EventDescription": "t2.small, ami-07c8bc5c1ce9598c3, ...",
                "EventSubType": "progress"
            },
            "EventType": "fleetRequestChange",
            "Timestamp": "2020-09-01T18:26:17.000Z"
        },
        {
            "EventInformation": {
                "EventDescription": "{\"instanceType\":\"t2.small\", ...}",
                "EventSubType": "launched",
                "InstanceId": "i-083a1c446e66085d2"
            },
            "EventType": "instanceChange",
            "Timestamp": "2020-09-01T18:26:17.000Z"
        },
        {
            "EventInformation": {
                "EventDescription": "{\"instanceType\":\"t2.small\", ...}",
                "EventSubType": "launched",
```

```
                "InstanceId": "i-090db02406cc3c2d6"
            },
            "EventType": "instanceChange",
            "Timestamp": "2020-09-01T18:26:17.000Z"
        }
    ],
    "FleetId": "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
    "LastEvaluatedTime": "1970-01-01T00:00:00.000Z",
    "StartTime": "2020-06-01T00:00:00.000Z"
}
```

PowerShell

지정된 EC2 플릿에 대한 이벤트를 설명하려면

Get-EC2FleetHistory cmdlet을 사용합니다.

```
Get-EC2FleetHistory `
    -FleetId fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE `
    -UtcStartTime 2020-06-01T00:00:00Z
```

EC2 집합 수정

EC2 플릿의 총 목표 용량, 스팟 용량 및 온디맨드 용량을 수정할 수 있습니다. 총 목표 용량이 플릿의 현재 크기보다 작아지면 실행 중인 인스턴스를 종료할지 여부를 수정할 수도 있습니다.

고려 사항

EC2 플릿 수정 시 고려할 사항:

- 플릿 유형 - 유형이 `maintain`인 EC2 플릿만 수정할 수 있습니다. 유형이 `request` 또는 `instant`인 EC2 플릿은 수정할 수 없습니다.
- 플릿 파라미터 - EC2 플릿의 다음 파라미터를 수정할 수 있습니다.
  - `target-capacity-specification` – 다음의 목표 용량 증가 또는 감소:
    - `TotalTargetCapacity`
    - `OnDemandTargetCapacity`
    - `SpotTargetCapacity`
  - `excess-capacity-termination-policy` – EC2 집합의 총 목표 용량이 플릿의 현재 크기보다 작아지면 실행 중인 인스턴스를 종료할지 여부입니다. 유효한 값은 다음과 같습니다.

  - `no-termination`
  - `termination`
- 총 목표 용량 증가 시 플릿 동작 - 총 목표 용량을 늘리면 EC2 플릿은 `DefaultTargetCapacityType`에 대해 지정된 인스턴스 구매 옵션(온디맨드 인스턴스 또는 스팟 인스턴스), 그리고 지정된 할당 전략에 따라 추가 인스턴스를 시작합니다.
- 스팟 목표 용량 감소 시 플릿 동작 - 스팟 목표 용량을 줄이면 EC2 플릿이 새 목표 용량을 초과하는 모든 열린 요청을 삭제합니다. 플릿의 크기가 새 목표 용량에 도달할 때까지 플릿에서 스팟 인스턴스를 종료하도록 요청할 수 있습니다. 목표 용량이 줄어 EC2 플릿이 스팟 인스턴스를 종료하면 해당 인스턴스는 스팟 인스턴스 중단 공지를 받습니다.

  인스턴스는 할당 전략에 따라 종료 대상이 선택됩니다.

  - `capacity-optimized` - 사용 가능한 용량이 가장 적은 풀에서 인스턴스를 종료합니다.
  - `price-capacity-optimized` - 가격과 사용 가능한 용량의 조합을 사용합니다. 즉, 사용 가능한 용량이 가장 적은 풀에 속하고 해당 풀에서 가격이 가장 높은 인스턴스를 종료합니다.
  - `diversified` - 모든 풀에서 인스턴스를 종료합니다.
  - `lowest-price` - 최고 가격의 풀에서 인스턴스를 종료합니다.

  또는 EC2 집합이 플릿을 현재 크기로 유지하되, 중단되거나 수동으로 종료하는 스팟 인스턴스를 교체하지 않도록 요청할 수 있습니다.
- 플릿 상태 - `submitted` 또는 `active` 상태인 EC2 플릿을 수정할 수 있습니다. 플릿을 수정할 때 플릿은 `modifying` 상태가 됩니다.

EC2 플릿 수정 명령

AWS CLI

EC2 플릿의 총 목표 용량 수정

modify-fleet 명령을 사용합니다.

```
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 이전 예제를 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
```

PowerShell

EC2 플릿의 총 목표 용량 수정

Edit-EC2Fleet cmdlet을 사용합니다.

```
Edit-EC2Fleet `
    -FleetId "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TargetCapacitySpecification_TotalTargetCapacity 20
```

목표 용량을 줄이고 플릿은 현재 크기로 유지하려는 경우 다음과 같이 이전 예제를 수정할 수 있습니다.

```
Edit-EC2Fleet `
    -FleetId "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TargetCapacitySpecification_TotalTargetCapacity 20 `
    -ExcessCapacityTerminationPolicy "NoTermination"
```

EC2 플릿 요청 및 플릿의 인스턴스 삭제

EC2 플릿 요청이 더 이상 필요하지 않으면 삭제할 수 있습니다. 플릿 요청을 삭제하면 플릿과 연결된 모든 스팟 요청이 취소되어 새 스팟 인스턴스가 시작되지 않습니다.

EC2 플릿 요청을 삭제하는 경우 해당 인스턴스도 모두 종료할지 여부를 지정해야 합니다. 여기에는 온디맨드 인스턴스와 스팟 인스턴스가 모두 포함됩니다. `instant` 플릿의 경우, EC2 플릿은 플릿이 삭제될 때 인스턴스를 종료해야 합니다. 삭제된 `instant` 플릿에서 인스턴스를 실행하는 것은 지원되지 않습니다.

플릿 요청이 삭제되면 인스턴스가 종료되도록 지정할 경우 플릿 요청이 `deleted_terminating` 상태가 됩니다. 그렇게 하지 않으면 `deleted_running` 상태가 되어 인스턴스가 중단되거나 수동으로 종료될 때까지 계속 실행됩니다.

제한 사항

- 단일 작업으로 `instant` 유형의 플릿을 25개까지 삭제할 수 있습니다.

- 단일 작업으로 `maintain` 또는 `request` 유형의 플릿을 100개까지 삭제할 수 있습니다.
- 위에서 지정한 대로 각 플릿 유형의 할당량을 초과하지 않는 경우 단일 작업으로 최대 125개의 플릿을 삭제할 수 있습니다.
- 삭제할 플릿의 지정된 수를 초과하면 플릿이 삭제되지 않습니다.
- 삭제된 `instant` 플릿에서 인스턴스를 실행하는 것은 지원되지 않습니다. `instant` 플릿을 삭제하면 Amazon EC2가 모든 인스턴스를 자동으로 종료합니다. 인스턴스가 1,000개 이상인 `instant` 플릿의 경우 삭제 요청이 실패할 수 있습니다. 플릿에 인스턴스가 1,000개 이상인 경우 먼저 인스턴스를 대부분 수동으로 종료하여 1,000개 이하를 남깁니다. 그런 다음 플릿을 삭제하면 나머지 인스턴스가 자동으로 종료됩니다.

AWS CLI

EC2 플릿 요청을 삭제하고 해당 인스턴스를 종료하려면

delete-fleets 명령을 `--terminate-instances` 옵션과 함께 사용합니다.

```
aws ec2 delete-fleets \
    --fleet-ids fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --terminate-instances
```

출력의 예시는 다음과 같습니다.

```
{
    "UnsuccessfulFleetDeletions": [],
    "SuccessfulFleetDeletions": [
        {
            "CurrentFleetState": "deleted_terminating",
            "PreviousFleetState": "active",
            "FleetId": "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
        }
    ]
}
```

인스턴스 종료 없이 EC2 플릿 요청을 삭제하려면

`--no-terminate-instances` 옵션을 대신 사용하여 이전 예제를 수정합니다. `--no-terminate-instances`는 `instant` 플릿에 지원되지 않습니다.

```
aws ec2 delete-fleets \
```

```
    --fleet-ids fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --no-terminate-instances
```

출력의 예시는 다음과 같습니다.

```
{
    "UnsuccessfulFleetDeletions": [],
    "SuccessfulFleetDeletions": [
        {
            "CurrentFleetState": "deleted_running",
            "PreviousFleetState": "active",
            "FleetId": "fleet-4b8aaae8-dfb5-436d-a4c6-3dafa4c6b7dcEXAMPLE"
        }
    ]
}
```

PowerShell

EC2 플릿 요청을 삭제하고 해당 인스턴스를 종료하려면

Remove-EC2Fleet cmdlet을 `-TerminateInstance` 파라미터와 함께 사용합니다.

```
Remove-EC2Fleet `
    -FleetId "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TerminateInstance $true
```

인스턴스 종료 없이 EC2 플릿 요청을 삭제하려면

`-TerminateInstance` 파라미터 값을 변경하여 이전 예제를 수정합니다.

```
Remove-EC2Fleet `
    -FleetId "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TerminateInstance $false
```

플릿 삭제 실패 시 문제 해결

EC2 플릿 요청이 삭제에 실패하는 경우 출력의 `UnsuccessfulFleetDeletions`에서 EC2 플릿의 ID, 오류 코드 및 오류 메시지를 반환합니다.

오류 코드는 다음과 같습니다.

- ExceededInstantFleetNumForDeletion
- fleetIdDoesNotExist
- fleetIdMalformed
- fleetNotInDeletableState
- NoTerminateInstancesNotSupported
- UnauthorizedOperation
- unexpectedError

ExceededInstantFleetNumForDeletion 문제 해결

단일 요청에서 25개 이상의 instant 플릿을 삭제하려고 하면
ExceededInstantFleetNumForDeletion 오류가 반환됩니다. 다음은 이 오류의 예제 출력입니다.

```
{
    "UnsuccessfulFleetDeletions": [
     {
          "FleetId": " fleet-5d130460-0c26-bfd9-2c32-0100a098f625",
          "Error": {
                  "Message": "Can’t delete more than 25 instant fleets in a single
 request.",
                  "Code": "ExceededInstantFleetNumForDeletion"
           }
     },
     {
           "FleetId": "fleet-9a941b23-0286-5bf4-2430-03a029a07e31",
           "Error": {
                  "Message": "Can’t delete more than 25 instant fleets in a single
 request.",
                  "Code": "ExceededInstantFleetNumForDeletion"
            }
     }
     .
     .
     .
     ],
     "SuccessfulFleetDeletions": []
}
```

NoTerminateInstancesNotSupported 문제 해결

플릿을 삭제할 때 instant 플릿의 인스턴스를 종료하지 않도록 지정하면 NoTerminateInstancesNotSupported 오류가 반환됩니다. --no-terminate-instances는 instant 플릿에 대해 지원되지 않습니다. 다음은 이 오류의 예제 출력입니다.

```
{
      "UnsuccessfulFleetDeletions": [
            {
                  "FleetId": "fleet-5d130460-0c26-bfd9-2c32-0100a098f625",
                  "Error": {
                          "Message": "NoTerminateInstances option is not supported for
 instant fleet",
                          "Code": "NoTerminateInstancesNotSupported"
                   }
            }
       ],
       "SuccessfulFleetDeletions": []
}
```

UnauthorizedOperation 문제 해결

인스턴스 종료 권한이 없는 경우 인스턴스를 종료해야 하는 플릿을 삭제할 때 UnauthorizedOperation 오류가 발생합니다. 다음은 오류 응답입니다.

```
<Response><Errors><Error><Code>UnauthorizedOperation</Code><Message>You are not
 authorized to perform this
operation. Encoded authorization failure message: VvuncIxj7Z_CPGNYXWqnuFV-
YjByeAU66Q9752NtQ-I3-qnDLWs6JLFd
KnSMMiq5s6cGqjjPtEDpsnGHzzyHasFHOaRYJpaDVravoW25azn6KNkUQQlFwhJyujt2dtNCdduJfrqcFYAjlEiRMkfDHt7
BHturzDK6A560Y2nDSUiMmAB1y9UNtqaZJ9SNe5sNxKMqZaqKtjRbk02RZu5V2vn9VMk6fm2aMVHbY9JhLvGypLcMUjtJ76
VPiU5v2s-
UgZ7h0p2yth6ysUdhlONg6dBYu8_y_HtEI54invCj4CoK0qawqzMNe6rcmCQHvtCxtXsbkgyaEbcwmrm2m01-
EMhekLFZeJLr
DtYOpYcEl4_nWFX1wtQDCnNNCmxnJZAoJvb3VMDYpDTsxjQv1PxODZuqWHs23YXWVywzgnLtHeRf2o4lUhGBw17mXsS07k7
PT9vrHtQiILor5VVTsjSPWg7edj__1rsnXhwPSu8gI48ZLRGrPQqFq0RmKO_QIE8N8s6NWzCK4yoX-9gDcheurOGpkprPIC
</Message></Error></Errors><RequestID>89b1215c-7814-40ae-a8db-41761f43f2b0</
RequestID></Response>
```

이 오류를 해결하려면 다음 예제와 같이 IAM 정책에 ec2:TerminateInstances 작업을 추가해야 합니다.

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Sid": "DeleteFleetsAndTerminateInstances",
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteFleets"
                "ec2:TerminateInstances"
            ],
            "Resource": "*"
        }
    ]
}
```

스팟 플릿 사용

스팟 플릿 사용을 시작하기 위해 선택적 온디맨드 부분인 스팟 인스턴스의 총 대상 용량을 포함하는 요청을 생성하고 AMI와 키 페어를 수동으로 지정하거나 플릿의 인스턴스에 대한 구성을 포함하는 시작 템플릿을 지정합니다. 추가 파라미터를 지정하거나 플릿에서 기본값을 사용하도록 할 수도 있습니다. 또한 플릿을 생성할 때 플릿 요청과 해당 인스턴스 및 볼륨에 태그를 지정할 수 있습니다.

플릿은 가용 용량이 있을 때 온디맨드 인스턴스를 시작하며, 최고 가격이 스팟 가격을 초과하고 가용 용량이 있을 때 스팟 인스턴스를 시작합니다.

플릿이 시작되면 플릿 요청, 플릿의 인스턴스, 모든 플릿 이벤트를 설명할 수 있습니다. 필요에 따라 추가 태그를 할당할 수도 있습니다.

총 대상 용량 등의 플릿 파라미터를 변경해야 하는 경우 용량을 유지하도록 구성된 플릿만 수정하면 됩니다. 일회성 요청이 제출된 후에는 해당 요청의 용량을 수정할 수 없습니다.

플릿 요청은 요청이 만료되거나 취소(삭제)될 때까지 활성 상태로 유지됩니다. 플릿 요청을 취소할 때 인스턴스를 종료하거나 실행 중인 상태로 둘 수 있습니다 실행 중인 상태로 둘 경우 온디맨드 인스턴스는 사용자가 종료할 때까지 실행되고 스팟 인스턴스는 중단되거나 사용자가 종료할 때까지 실행됩니다.

주제

- 스팟 플릿 요청 상태
- 스팟 플릿 권한
- 스팟 플릿 생성
- 신규 또는 기존 스팟 플릿 요청과 이를 통해 시작되는 인스턴스 및 볼륨 태그 지정

- 스팟 플릿 요청, 해당 인스턴스, 이벤트 기록 설명
- 스팟 플릿 요청 수정
- 스팟 플릿 요청 취소(삭제)
- 스팟 플릿의 자동 규모 조정 이해

스팟 플릿 요청 상태

스팟 플릿 요청 상태는 다음 중 하나일 수 있으며, 각 상태는 요청 수명 주기 및 인스턴스 관리의 서로 다른 단계를 나타냅니다.

스팟 플릿 요청은 다음 상태 중 하나일 수 있습니다.

`submitted`

스팟 플릿 요청을 평가 중이며 Amazon EC2에서 목표 개수의 인스턴스를 시작하기 위해 준비 중입니다. 요청이 스팟 플릿 할당량을 초과하면 해당 요청이 즉시 취소됩니다.

`active`

스팟 플릿이 검증되었으며 Amazon EC2가 실행 중인 스팟 인스턴스의 목표 개수를 유지하려고 시도하고 있습니다. 그 요청은 수정 또는 취소될 때까지 계속 이 상태로 유지됩니다.

`modifying`

스팟 집합 요청이 수정되고 있습니다. 그 요청은 수정이 완전히 처리될 때까지 또는 요청이 취소될 때까지 계속 이 상태로 유지됩니다. `maintain` 플릿 유형만 수정할 수 있습니다. 이 상태는 일회성 `request` 플릿 유형에 적용되지 않습니다.

`cancelled_running`

스팟 플릿이 취소(삭제)되었고 추가 스팟 인스턴스가 시작되지 않습니다. 수동으로 중단되거나 종료될 때까지 기존 인스턴스가 계속 실행됩니다. 그 요청은 모든 인스턴스가 중단 또는 종료될 때까지 계속 이 상태로 유지됩니다.

`cancelled_terminating`

스팟 플릿이 취소(삭제)되었고 해당 인스턴스를 종료하는 중입니다. 그 요청은 모든 인스턴스가 종료될 때까지 계속 이 상태로 유지됩니다.

`cancelled`

스팟 플릿이 취소(삭제)되었고 실행 중인 인스턴스가 없습니다. 인스턴스가 종료되고 2일 후 요청이 삭제됩니다.

스팟 플릿 권한

사용자가 스팟 플릿을 생성하거나 관리하는 경우 필요한 권한을 부여해야 합니다.

Amazon EC2 콘솔을 사용하여 스팟 플릿을 생성하는 경우 `AWSServiceRoleForEC2SpotFleet` 및 `AWSServiceRoleForEC2Spot`이라는 서비스 연결 역할 2개와 스팟 플릿에 사용자 대신 리소스를 요청, 시작, 종료 및 태깅할 수 있는 권한을 부여하는 `aws-ec2-spot-fleet-tagging-role`이라는 역할이 생성됩니다. AWS CLI 또는 API를 사용하는 경우 이러한 역할이 존재하는지 확인해야 합니다.

다음 지침에 따라 필요한 권한을 부여하고 역할을 생성합니다.

사용 권한 및 역할

- 스팟 플릿의 사용자에게 권한 부여
- 스팟 플릿의 서비스 연결 역할
- 스팟 인스턴스에 대한 서비스 연결 역할
- 스팟 플릿 태깅을 위한 IAM 역할

스팟 플릿의 사용자에게 권한 부여

스팟 플릿을 생성하거나 관리하는 사용자에게 필요한 권한을 부여해야 합니다.

스팟 플릿에 대한 정책 생성

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 선택하고 텍스트를 다음과 같이 바꿉니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances",
                "ec2:CreateTags",
                "ec2:RequestSpotFleet",
                "ec2:ModifySpotFleetRequest",
                "ec2:CancelSpotFleetRequests",
                "ec2:DescribeSpotFleetRequests",
```

```
                "ec2:DescribeSpotFleetInstances",
                "ec2:DescribeSpotFleetRequestHistory"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/aws-ec2-spot-fleet-tagging-role"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole",
                "iam:ListRoles",
                "iam:ListInstanceProfiles"
            ],
            "Resource": "*"
        }
    ]
}
```

위의 예제 정책은 사용자에게 대부분의 스팟 플릿 사용 사례에 필요한 권한을 부여합니다. 사용자를 특정 API 작업으로 제한하려면 대신 해당 작업만 지정하세요.

필수 EC2 및 IAM API

다음 API가 정책에 포함되어야 합니다.

- `ec2:RunInstances` - 스팟 플릿에서 인스턴스를 시작하는 데 필요합니다.
- `ec2:CreateTags` - 스팟 플릿 요청, 인스턴스 또는 볼륨을 태깅하는 데 필요합니다.
- `iam:PassRole` - 스팟 플릿 역할을 지정하는 데 필요합니다.
- `iam:CreateServiceLinkedRole` - 서비스 연결 역할을 생성하는 데 필요합니다.
- `iam:ListRoles` - 기존 IAM 역할을 열거하는 데 필요합니다.
- `iam:ListInstanceProfiles` - 기존 인스턴스 프로파일을 열거하는 데 필요합니다.

> **Important**
>
> 시작 사양 또는 시작 템플릿에서 IAM 인스턴스 프로파일에 대한 역할을 지정하는 경우, 사용자에게 역할을 서비스에 전달할 수 있는 권한을 부여해야 합니다. 이를 위해

IAM 정책에 "arn:aws:iam::*:role/*IamInstanceProfile-role*" 작업을 위한 iam:PassRole 리소스를 포함하세요. 자세한 내용은 IAM 사용 설명서에서 사용자에게 AWS 서비스 역할을 전달할 수 있는 권한 부여를 참조하세요.

스팟 플릿 API

필요에 따라 다음 스팟 플릿 API 작업을 정책에 추가합니다.

- ec2:RequestSpotFleet
- ec2:ModifySpotFleetRequest
- ec2:CancelSpotFleetRequests
- ec2:DescribeSpotFleetRequests
- ec2:DescribeSpotFleetInstances
- ec2:DescribeSpotFleetRequestHistory

선택적 IAM API

(선택 사항) 사용자가 IAM 콘솔을 사용하여 역할 또는 인스턴스 프로파일을 생성할 수 있도록 하려면 정책에 다음 작업을 추가해야 합니다.

- iam:AddRoleToInstanceProfile
- iam:AttachRolePolicy
- iam:CreateInstanceProfile
- iam:CreateRole
- iam:GetRole
- iam:ListPolicies

4. 정책 검토를 선택합니다.
5. 정책 검토 페이지에 정책 이름과 설명을 입력한 다음 정책 생성을 선택합니다.
6. 액세스 권한을 제공하려면 사용자, 그룹 또는 역할에 권한을 추가하세요:

   - AWS IAM Identity Center의 사용자 및 그룹:

     권한 세트를 생성합니다. AWS IAM Identity Center 사용 설명서의 권한 세트 생성의 지침을 따릅니다.

- 보안 인증 공급자를 통해 IAM에서 관리되는 사용자:

  ID 페더레이션을 위한 역할을 생성합니다. IAM 사용 설명서의 Create a role for a third-party identity provider (federation)의 지침을 따릅니다.
- IAM 사용자:
  - 사용자가 맡을 수 있는 역할을 생성합니다. IAM 사용 설명서에서 Create a role for an IAM user의 지침을 따릅니다.
  - (권장되지 않음)정책을 사용자에게 직접 연결하거나 사용자를 사용자 그룹에 추가합니다. IAM 사용 설명서에서 사용자(콘솔)에 권한 추가의 지침을 따르세요.

스팟 플릿의 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 역할을 참조하세요.

Amazon EC2는 AWSServiceRoleForEC2Spot이라는 이름의 서비스 연결 역할을 사용하여 사용자 대신 인스턴스를 시작하고 관리합니다.

> **Important**
>
> 스팟 플릿에서 암호화된 AMI 또는 암호화된 Amazon EBS 스냅샷을 지정하는 경우 Amazon EC2에서 자동으로 인스턴스를 시작하려면 CMK를 사용할 권한을 AWSServiceRoleForEC2SpotFleet 역할에 부여해야 합니다. 자세한 내용은 암호화된 AMI 및 EBS 스냅샷에 사용할 CMK에 대한 액세스 권한 부여 섹션을 참조하세요.

AWSServiceRoleForEC2SpotFleet에서 부여된 권한

AWSServiceRoleForEC2SpotFleet 역할은 스팟 플릿에 사용자 대신 인스턴스를 요청, 시작 및 종료하고 태그를 지정할 수 있는 권한을 부여합니다. Amazon EC2는 이 서비스 연결 역할을 사용하여 다음 작업을 완료합니다.

- `ec2:RequestSpotInstances` - 스팟 인스턴스 요청
- `ec2:RunInstances` - 인스턴스 시작
- `ec2:TerminateInstances` - 인스턴스 종료

- `ec2:DescribeImages` - 인스턴스에 대한 Amazon Machine Image(AMI) 설명
- `ec2:DescribeInstanceStatus` - 인스턴스 상태 설명
- `ec2:DescribeSubnets` - 인스턴스의 서브넷 설명
- `ec2:CreateTags` - 스팟 플릿 요청, 인스턴스 및 볼륨에 태그 추가
- `elasticloadbalancing:RegisterInstancesWithLoadBalancer` - 지정된 로드 밸런서에 지정된 인스턴스 추가
- `elasticloadbalancing:RegisterTargets` - 지정된 대상 그룹에 지정된 대상 등록

서비스 연결 역할 생성

대부분의 상황에서는 서비스 연결 역할을 수동으로 생성할 필요가 없습니다. 사용자가 콘솔을 사용하여 처음으로 스팟 플릿을 생성하면 Amazon EC2가 AWSServiceRoleForEC2SpotFleet 서비스 연결 역할을 생성합니다.

Amazon EC2가 이 서비스 연결 역할을 지원하기 시작한 2017년 10월 이전에 활성 스팟 플릿 요청을 보유한 경우 Amazon EC2에는 사용자의 AWS 계정에 AWSServiceRoleForEC2SpotFleet 역할이 이미 생성되어 있습니다. 자세한 내용은 IAM 사용 설명서에서 내 AWS 계정에 표시되는 새 역할을 참조하세요.

AWS CLI 또는 API를 사용하여 스팟 플릿을 생성하는 경우 먼저 이 역할이 있는지 확인해야 합니다.

콘솔을 사용하여 스팟 플릿에 대한 AWSServiceRoleForEC2SpotFleet 역할을 생성하려면 다음을 수행하세요.

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 역할을 선택합니다.
3. 역할 생성(Create role)을 선택합니다.
4. 신뢰할 수 있는 엔터티 선택 페이지에서 다음을 수행합니다.
    a. 신뢰할 수 있는 엔터티 유형에서 AWS 서비스를 선택합니다.
    b. 사용 사례에서 서비스 또는 사용 사례로 EC2를 선택합니다.
    c. 사용 사례에서 EC2 - 스팟 플릿을 선택합니다.

> **Note**
>
> EC2 - 스팟 플릿 사용 사례는 필요한 IAM 권한으로 정책을 자동으로 생성하고 AWSEC2SpotFleetServiceRolePolicy를 역할 이름으로 제안합니다.

d. 다음을 선택합니다.

5. 권한 추가 페이지에서 다음을 선택합니다.
6. 이름 지정, 검토 및 생성 페이지에서 역할 생성을 선택합니다.

AWS CLI를 사용하여 스팟 플릿에 대한 AWSServiceRoleForEC2SpotFleet 역할을 생성하려면 다음을 수행하세요.

다음과 같이 create-service-linked-role 명령을 사용합니다.

```
aws iam create-service-linked-role --aws-service-name spotfleet.amazonaws.com
```

스팟 플릿이 더 이상 필요 없으면 AWSServiceRoleForEC2SpotFleet 역할을 삭제하는 것이 좋습니다. 계정에서 이 역할이 삭제된 후 콘솔을 사용하여 스팟 플릿을 요청하면 Amazon EC2가 다시 해당 역할을 생성합니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 삭제를 참조하십시오.

암호화된 AMI 및 EBS 스냅샷에 사용할 CMK에 대한 액세스 권한 부여

스팟 플릿 요청에서 암호화된 AMI 또는 암호화된 Amazon EBS 스냅샷을 지정하고 암호화에 고객 관리형 키를 사용하는 경우 Amazon EC2에서 자동으로 인스턴스를 시작하려면 CMK를 사용할 권한을 AWSServiceRoleForEC2SpotFleet 역할에 부여해야 합니다. 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 개발자 안내서에서 권한 부여 사용 및 AWS KMS의 키 정책 사용을 참조하세요.

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.

```
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/aws-service-role/
spotfleet.amazonaws.com/AWSServiceRoleForEC2SpotFleet \
    --operations "Decrypt" "Encrypt" "GenerateDataKey"
 "GenerateDataKeyWithoutPlaintext" "CreateGrant" "DescribeKey" "ReEncryptFrom"
 "ReEncryptTo"
```

스팟 인스턴스에 대한 서비스 연결 역할

Amazon EC2는 AWSServiceRoleForEC2Spot이라는 이름의 서비스 연결 역할을 사용하여 사용자 대신 스팟 인스턴스를 시작하고 관리합니다. 자세한 내용은 스팟 인스턴스 요청에 대한 서비스 연결 역할 섹션을 참조하세요.

스팟 플릿 태깅을 위한 IAM 역할

`aws-ec2-spot-fleet-tagging-role` IAM 역할은 스팟 플릿 요청, 인스턴스 및 볼륨에 태깅할 수 있는 권한을 스팟 플릿에 부여합니다. 자세한 내용은 신규 또는 기존 스팟 플릿 요청과 이를 통해 시작되는 인스턴스 및 볼륨 태그 지정 섹션을 참조하세요.

> **Important**
>
> 플릿의 인스턴스를 태깅하고 목표 용량을 유지하기로 선택하면(스팟 플릿 요청은 `maintain` 유형) 사용자 설정 권한 차이로 인해 `IamFleetRole`은 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 `aws-ec2-spot-fleet-tagging-role`에 `IamFleetRole` 역할을 사용하는 것이 좋습니다. 또는 기존 역할을 사용하려면 `AmazonEC2SpotFleetTaggingRole` AWS 관리형 정책을 기존 역할에 연결합니다. 그렇게 하지 않으면 기존 정책에 `CreateTags` 권한을 수동으로 추가해야 합니다.

스팟 플릿 태깅을 위한 IAM 역할을 생성하려면

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 역할을 선택합니다.
3. 역할 생성(Create role)을 선택합니다.

4. 신뢰할 수 있는 엔터티 선택 페이지의 신뢰할 수 있는 엔터티 유형(Trusted entity type) 아래에서 서비스(AWS service)를 선택합니다.
5. 사용 사례(Use case)의 기타 AWS 서비스 사용 사례(Use cases for other services)에서 EC2를 선택한 다음 EC2 - 스팟 플릿 태깅(EC2 - Spot Fleet Tagging)을 선택합니다.
6. 다음을 선택합니다.
7. 권한 추가 페이지에서 다음을 선택합니다.
8. 이름, 검토 및 생성 페이지에서 역할 이름(Role name)에서 역할 이름을 입력합니다(예: **aws-ec2-spot-fleet-tagging-role**).
9. 페이지에서 정보를 검토한 다음 역할 생성(Create role)을 선택합니다.

교차 서비스 혼동된 대리인 방지

혼동된 대리자 문제는 작업을 수행할 권한이 없는 엔터티가 권한이 더 많은 엔터티에 작업을 수행하도록 강요할 수 있는 보안 문제입니다. 리소스에 다른 서비스를 제공하는 스팟 플릿 권한을 제한하려면 `aws-ec2-spot-fleet-tagging-role` 신뢰 정책에서 `aws:SourceArn` 및 `aws:SourceAccount` 글로벌 조건 컨텍스트 키를 사용하는 것이 좋습니다.

aws:SourceArn 및 aws:SourceAccount 조건 키 **aws-ec2-spot-fleet-tagging-role** 신뢰 정책 추가

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 역할을 선택합니다.
3. 이전에 생성한 `aws-ec2-spot-fleet-tagging-role`을 찾아 링크(확인란 아님)를 선택합니다.
4. 요약(Summary)에서 신뢰 관계(Trust relationships) 탭을 선택한 후 신뢰 정책 편집(Edit trust policy)을 선택합니다.
5. JSON 문에 혼동된 대리자 문제(confused deputy problem)를 방지하는 `aws:SourceAccount` 및 `aws:SourceArn` 전역 조건 컨텍스트 키 등의 `Condition` 요소를 추가합니다.

```
"Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:ec2:us-east-1:account_id:spot-fleet-request/sfr-
*"
      },
      "StringEquals": {
        "aws:SourceAccount": "account_id"
```

```
      }
```

> **Note**
>
> aws:SourceArn 값에 계정 ID가 포함되고 두 전역 조건 컨텍스트 키를 사용하는 경우, aws:SourceAccount 값 및 aws:SourceArn 값의 계정은 동일한 정책 문에서 사용될 경우 반드시 같은 계정 ID를 사용해야 합니다.

최종 신뢰 정책은 다음과 같습니다.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "spotfleet.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:ec2:us-east-1:account_id:spot-fleet-request/sfr-
*"
      },
      "StringEquals": {
        "aws:SourceAccount": "account_id"
      }
    }
  }
}
```

6. 정책 업데이트(Update policy)를 선택합니다.

다음 표에는 다양한 정도의 특정성을 가진 aws-ec2-spot-fleet-tagging-role 범위를 제한하기 위해 aws:SourceArn에 대한 잠재적 값을 제공합니다.

API 작업	호출된 서비스	범위	**aws:SourceArn**
RequestSpotFleet	AWS STS (AssumeRole)	지정된 계정에서 스팟 플릿 요청에 대해 aws-ec2-spot-fleet-tagging-role 로 AssumeRole 기능을 제한합니다.	arn:aws:ec2:*: *123456789012* :spot-fleet-request/sfr-*
RequestSpotFleet	AWS STS (AssumeRole)	지정된 계정 및 지정된 리전에서 스팟 플릿 요청에 대해 aws-ec2-spot-fleet-tagging-role 로 AssumeRole 기능을 제한합니다. 이 역할은 다른 리전에서는 사용할 수 없습니다.	arn:aws:ec2: *us-east-1* :*123456789012* :spot-fleet-request/sfr-*
RequestSpotFleet	AWS STS (AssumeRole)	플릿 sfr-11111111-1111-1111-1111-111111111111에 영향을 미치는 작업에 대해서만 aws-ec2-spot-fleet-tagging-role 로 AssumeRole 기능을 제한합니다. 이 역할은 다른 스팟 플릿에서는 사용하지 못할 수 있습니다. 또한 이 역할은 요청 스팟 플릿을 통해 새로운 스팟 플릿을 시작하는 데 사용할 수 없습니다.	arn:aws:ec2: *us-east-1* :*123456789012* :spot-fleet-request/sfr-*11111111-1111-1111-1111-111111111111*

스팟 플릿 생성

AWS Management Console을 사용하여 AMI와 원하는 총 대상 용량만 선택하면 스팟 플릿 요청을 빠르게 생성할 수 있습니다. Amazon EC2는 스팟 모범 사례를 따르며 요구 사항을 가장 잘 충족하는 플릿을 구성합니다. 또는 어떠한 기본 설정이든 수정할 수 있습니다.

플릿에 온디맨드 인스턴스를 포함하려면 요청에서 시작 템플릿을 지정하고 원하는 온디맨드 용량을 지정해야 합니다.

플릿은 가용 용량이 있을 때 온디맨드 인스턴스를 시작하고, 최고 가격이 스팟 가격을 초과하고 가용 용량이 있을 때 스팟 인스턴스를 시작합니다.

플릿에 스팟 인스턴스가 포함되어 있고 유형이 `maintain`인 경우 Amazon EC2에서 스팟 인스턴스가 중단되어도 플릿 목표 용량을 유지하려고 합니다.

필수 권한

자세한 내용은 the section called “스팟 플릿 권한” 섹션을 참조하세요.

업무

- 스팟 플릿 요청을 빠르게 생성
- 정의된 파라미터를 사용하여 스팟 플릿 요청 생성
- 비정상 스팟 인스턴스를 교체하는 스팟 플릿 생성

스팟 플릿 요청을 빠르게 생성

다음과 같은 단계에 따라 Amazon EC2 콘솔을 사용하여 빠르게 스팟 플릿 요청을 생성합니다.

권장되는 설정을 사용하여 스팟 플릿 요청을 생성하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. 스팟을 처음 사용하는 경우 시작 페이지가 표시되면 시작하기를 선택합니다. 그렇지 않으면 스팟 플릿 요청 생성을 선택합니다.
4. 시작 파라미터(Launch parameters) 아래에서 시작 파라미터 수동 구성(Manually configure launch parameters)을 선택합니다.
5. AMI에서 AMI를 선택합니다.

6. 목표 용량(Target capacity)의 총 목표 용량(Total target capacity)에 요청할 단위 수를 지정합니다. 단위 유형에서 인스턴스, vCPU 또는 메모리(GiB)를 선택할 수 있습니다.
7. 플릿 요청 살펴보기에서 플릿 구성을 검토하고 시작을 선택합니다.

정의된 파라미터를 사용하여 스팟 플릿 요청 생성

정의한 파라미터를 사용하여 스팟 플릿을 생성할 수 있습니다.

Console

정의된 파라미터를 사용하여 스팟 플릿 요청을 생성하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. 스팟을 처음 사용하는 경우 시작 페이지가 표시되면 시작하기를 선택합니다. 그렇지 않으면 스팟 플릿 요청 생성을 선택합니다.
4. 시작 파라미터의 경우 다음과 같이 시작 파라미터를 수동으로 구성하거나 시작 템플릿을 사용할 수 있습니다.

   a. [수동 구성] Amazon EC2 콘솔에서 시작 파라미터를 정의하려면 시작 파라미터 수동 구성을 선택하고 다음을 수행합니다.

      i. AMI의 경우 AWS가 제공하는 기본 AMI 중 하나를 선택하거나 AMI 검색(Search for AMI)을 선택하여 사용자 커뮤니티, AWS Marketplace 또는 자체 AMI를 사용합니다.

      > **Note**
      >
      > 시작 파라미터에 지정된 AMI가 등록 취소되거나 비활성화된 경우 AMI에서 새 인스턴스를 시작할 수 없습니다. 목표 용량을 유지하도록 설정된 플릿의 경우 목표 용량이 유지되지 않습니다.

      ii. (선택 사항) 키 페어 이름에서 기존 키 페어를 선택하거나 새로 생성합니다.

      [기존 키 페어] 키 페어를 선택합니다.

      [새 키 페어] 새 키 페어 생성을 선택하여 키 페어 페이지로 이동합니다. 마친 후에 스팟 요청(Spot Requests) 페이지로 돌아가고 목록을 새로 고칩니다.

iii. (선택 사항) 추가 시작 파라미터(Additional launch parameters)를 확장하고 다음을 수행합니다.

A. (선택 사항) Amazon EBS 최적화를 사용하려면 EBS 최적화(EBS-optimized)에서 EBS 최적 인스턴스 시작(Launch EBS-optimized instances)을 선택합니다.

B. (선택 사항) 인스턴스에 대해 임시 블록 스토리지를 추가하려면 인스턴스 스토어에 대해 시작 시 연결을 선택합니다.

C. (선택 사항) 스토리지를 추가하려면 새로운 볼륨 추가(Add new volume)를 선택하고 인스턴스 유형에 따라 추가 인스턴스 스토어 볼륨이나 Amazon EBS 볼륨을 지정합니다.

D. (선택 사항) 기본적으로 인스턴스에 대해 기본 모니터링 기능이 활성화됩니다. 세부 모니터링을 사용하려면 모니터링(Monitoring)에서 CloudWatch 세부 모니터링 활성화(Enable CloudWatch detailed monitoring)를 선택합니다.

E. (선택 사항) 전용 스팟 인스턴스를 실행하려면 [테넌시(Tenancy)]에서 전용 - 전용 인스턴스 실행(Dedicated - run a dedicated instance)]을 선택합니다.

F. (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 생성(Create new security group)을 선택하여 보안 그룹(Security Groups) 페이지로 이동합니다. 마친 후에 스팟 요청(Spot Requests) 페이지로 돌아가고 목록을 새로 고칩니다.

G. (선택 사항) 인터넷에서 인스턴스에 연결할 수 있도록 하려면 IPv4 퍼블릭 IP 자동 할당에 대해 활성화를 선택합니다.

H. (선택 사항) IAM 역할로 스팟 인스턴스를 시작하려면 IAM 인스턴스 프로파일에서 역할을 선택합니다.

I. (선택 사항) 시작 스크립트를 실행하려면 해당 스크립트를 사용자 데이터에 복사합니다.

J. (선택 사항) 태그를 추가하려면 태그 생성(Create tag)을 선택하고 해당 태그에 대한 키와 값을 입력한 다음 생성(Create)을 선택합니다. 각 태그에 대해 반복합니다.

각 태그에서 인스턴스와 스팟 플릿 요청에 같은 태그를 사용하여 태깅하려면 인스턴스(Instance)와 플릿(Fleet)이 모두 선택되어 있는지 확인합니다. 플릿에서

시작한 인스턴스만 태깅하려면 플릿(Fleet)을 선택 취소합니다. 스팟 플릿 요청만 태깅하려면 인스턴스(Instance) 선택을 취소합니다.

b. [시작 템플릿] 시작 템플릿에서 생성한 구성을 사용하려면 시작 템플릿 사용을 선택하고, 시작 템플릿에서 시작 템플릿을 선택합니다.

> **Note**
>
> 스팟 플릿에서 온디맨드 용량을 사용하려면 시작 템플릿을 지정해야 합니다.

5. (선택 사항) 추가 요청 세부 정보(Additional request details)에서 다음을 수행합니다.

   a. 추가 요청 세부 정보를 검토합니다. 변경하려면 기본값 적용(Apply defaults)의 선택을 취소합니다.

   b. (선택 사항) IAM 플릿 역할(IAM fleet role)에서 기본 역할을 사용하거나 다른 역할을 선택할 수 있습니다. 역할을 변경한 후 기본 역할을 사용하려면 기본 역할 사용을 선택합니다.

   c. (선택 사항) 특정 기간 동안만 유효한 요청을 생성하려면 요청 유효 시작 시간 및 요청 유효 종료 시간(Request valid until)을 편집합니다.

   d. (선택 사항) 기본적으로 스팟 플릿 요청이 만료되면 Amazon EC2에서 스팟 인스턴스를 종료합니다. 요청 만료 후에도 계속 실행하려면 요청 만료 시 인스턴스 종료(Terminate the instances when the request expires)를 선택 취소합니다.

   e. (선택 사항) 로드 밸런서에 스팟 인스턴스를 등록하려면 하나 이상의 로드 밸런서에서 트래픽 수신을 선택하고 하나 이상의 Classic Load Balancer나 대상 그룹을 선택합니다.

6. 목표 용량(Target capacity)에서 다음을 수행합니다.

   a. 총 목표 용량(Total target capacity)에서 요청할 단위 수를 지정합니다. 단위 유형에서 인스턴스(Instances), vCPU(vCPUs) 또는 메모리(MiB)(Memory (MiB))를 선택할 수 있습니다. 나중에 용량을 추가할 수 있도록 목표 용량을 0으로 지정하려면 목표 용량 유지를 선택해야 합니다.

   b. (선택 사항) 온디맨드 기반 용량 포함(Include On-Demand base capacity)에서 요청할 온디맨드 단위 수를 지정합니다. 이 수는 [총 목표 용량(Total target capacity)]보다 작아야 합니다. Amazon EC2는 차이를 계산하고 스팟 단위에 요청할 차이를 할당합니다.

> **Important**
>
> 선택적 온디맨드 용량을 지정하려면 먼저 시작 템플릿을 선택해야 합니다.

c. (선택 사항) 기본적으로 Amazon EC2는 스팟 인스턴스가 중단되면 스팟 인스턴스를 종료합니다. 목표 용량을 유지하려면 목표 용량 유지(Maintain target capacity)를 선택합니다. 그런 다음, 스팟 인스턴스가 중단되면 Amazon EC2는 해당 스팟 인스턴스를 종료하거나 중지하거나 최대 절전 모드로 전환합니다. 이를 위해 인터럽트 방식에서 해당 옵션을 선택합니다.

> **Note**
>
> 시작 파라미터에 지정된 AMI가 등록 취소되거나 비활성화된 경우 AMI에서 새 인스턴스를 시작할 수 없습니다. 이 경우 목표 용량을 유지하도록 설정된 플릿의 목표 용량이 유지되지 않습니다.

d. (선택 사항) 플릿의 기존 스팟 인스턴스에 대해 인스턴스 재분배 알림이 생성될 때 스팟 플릿에서 대체 스팟 인스턴스를 시작할 수 있도록 하려면 용량 재분배(Capacity rebalance)를 선택한 다음 인스턴스 대체 전략을 선택합니다. 종료 전 시작을 선택하는 경우 Amazon EC2에서 이전 인스턴스를 종료하기 전의 지연 시간(초)을 지정합니다. 자세한 내용은 EC2 플릿 및 스팟 플릿의 용량 리밸런싱을 사용하여 위험이 있는 스팟 인스턴스 교체 섹션을 참조하세요.

e. (선택 사항) 플릿의 모든 스팟 인스턴스에 대해 시간당 지불하는 금액을 관리하려면 스팟 인스턴스의 최대 비용 설정(Set maximum cost for Spot Instances)을 선택한 다음 시간당 지불할 최대 총 금액을 입력합니다. 최대 총 금액에 도달하면 스팟 플릿은 목표 용량을 충족하지 않은 경우에도 스팟 인스턴스 시작을 중지합니다. 자세한 내용은 EC2 플릿 또는 스팟 플릿에 지출 한도 설정 섹션을 참조하세요.

7. 네트워크(Network)에서 다음을 수행합니다.

a. 네트워크에서 기존 VPC를 선택하거나 새로 생성합니다.

[기존 VPC] VPC를 선택합니다.

[새 VPC] Amazon VPC 콘솔로 이동하려면 새 VPC 생성을 선택합니다. 완료되면 이 화면으로 돌아와 목록을 새로 고칩니다.

b. (선택 사항) 가용 영역의 경우 Amazon EC2에서 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. 완료되면 이 화면으로 돌아와 목록을 새로 고칩니다.

8. 인스턴스 유형 요구 사항(Instance type requirements)에서 인스턴스 속성을 지정하고 Amazon EC2가 해당 속성으로 인스턴스 유형을 식별하도록 하거나 인스턴스 목록을 지정할 수 있습니다. 자세한 내용은 EC2 플릿 또는 스팟 플릿의 인스턴스 유형 선택에 대한 속성 지정 섹션을 참조하세요.

   a. 컴퓨팅 요구 사항에 맞는 인스턴스 속성 지정(Specify instance attributes that match your compute requirements)을 선택하는 경우 다음과 같이 인스턴스 속성을 지정합니다.

      i. vCPU(vCPUs)에 원하는 최소 및 최대 vCPU 수를 입력합니다. 무한을 지정하려면 최소 없음, 최대 없음 또는 두 가지 모두를 선택합니다.

      ii. 메모리(GiB)(Memory (GiB))에 원하는 최소 및 최대 메모리 양을 입력합니다. 무한을 지정하려면 최소 없음, 최대 없음 또는 두 가지 모두를 선택합니다.

      iii. (선택 사항) 추가 인스턴스 속성에서 필요에 따라 하나 이상의 속성을 지정하여 컴퓨팅 요구 사항을 더 자세히 표현할 수 있습니다. 각 추가 속성은 요청에 추가 제약 조건을 추가합니다. 추가 속성을 생략할 수 있으며 생략 시 기본값이 사용됩니다. 각 속성과 기본값에 대한 설명은 get-spot-placement-scores를 참조하세요.

      iv. (선택 사항) 지정한 속성을 가진 인스턴스 유형을 보려면 일치하는 인스턴스 유형 미리 보기(Preview matching instance types)를 확장합니다. 요청에 사용되는 인스턴스 유형을 제외하려면 인스턴스를 선택한 다음 선택한 인스턴스 유형 제외(Exclude selected instance types)를 선택합니다.

   b. 수동으로 인스턴스 유형 선택(Manually select instance types)을 선택하는 경우 스팟 플릿은 기본 인스턴스 유형 목록을 제공합니다. 인스턴스 유형을 더 많이 선택하려면 인스턴스 유형 추가(Add instance types)를 선택하고 요청에 사용할 인스턴스 유형을 선택한 다음 선택(Select)을 선택합니다. 인스턴스 유형을 삭제하려면 인스턴스 유형을 선택하고 삭제(Delete)를 선택합니다.

9. 할당 전략의 경우 필요에 맞는 스팟 할당 전략과 온디맨드 할당 전략을 선택합니다. 자세한 내용은 할당 전략을 사용하여 EC2 플릿 또는 스팟 플릿이 스팟 및 온디맨드 용량을 충족하는 방법 결정 섹션을 참조하세요.

10. 플릿 요청 한눈에 보기(Your fleet request at a glance)에서 플릿 구성을 검토하고 필요한 경우 조정합니다.

11. (선택 사항) AWS CLI용 시작 구성의 복사본을 다운로드하려면 JSON 구성을 선택합니다.

12. 스팟 플릿을 시작할 준비가 되면 시작을 선택합니다.

    스팟 플릿 요청 유형은 `fleet`입니다. 요청이 이행되면 `instance` 유형의 요청이 추가되며, 그 상태는 `active`이고 상황은 `fulfilled`입니다.

AWS CLI

스팟 플릿 요청을 생성하는 방법

request-spot-fleet 명령을 사용합니다.

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

구성 파일에 대한 예시는 예제 CLI 구성 스팟 플릿 섹션을 참조하세요.

PowerShell

스팟 플릿 요청을 생성하는 방법

Request-EC2SpotFleet cmdlet을 사용합니다. 다음 예제는 용량 최적화 플릿에서 스팟 인스턴스를 시작합니다.

```
Request-EC2SpotFleet `
    -SpotFleetRequestConfig_TargetCapacity 50 `
    -SpotFleetRequestConfig_AllocationStrategy "CapacityOptimized" `
    -SpotFleetRequestConfig_IamFleetRole "arn:aws:iam::123456789012:role/my-spot-
fleet-role" `
    -SpotFleetRequestConfig_LaunchTemplateConfig @($launchConfig)
```

다음과 같이 시작 구성을 정의하여 시작 템플릿을 설정하고 필요한 속성을 재정의합니다. 구성에 대한 예시는 예제 CLI 구성 스팟 플릿 섹션을 참조하세요.

```
$lcSpec = Amazon.EC2.Model.FleetLaunchTemplateSpecification
# To do - Set FleetLaunchTemplateSpecification properties
$lcOverrides = New-Object Amazon.EC2.Model.LaunchTemplateOverrides
# To do - Set LaunchTemplateOverrides properties
$launchConfig = New-Object Amazon.EC2.Model.LaunchTemplateConfig
$launchConfig.LaunchTemplateSpecification $lcSpec
$launchConfig.Overrides @($lcOverrides)
```

비정상 스팟 인스턴스를 교체하는 스팟 플릿 생성

스팟 플릿은 2분마다 플릿의 스팟 인스턴스 상태를 확인합니다. 인스턴스의 상태는 `healthy` 또는 `unhealthy`입니다.

스팟 플릿은 Amazon EC2에서 제공하는 상태 확인을 사용하여 인스턴스의 상태를 판단합니다. 세 번의 연속 상태 확인에서 인스턴스 상태 또는 시스템 상태가 unhealthy인 경우, 해당 인스턴스의 상태는 impaired로 확인됩니다. 자세한 내용은 Amazon EC2 인스턴스 상태 확인 섹션을 참조하세요.

플릿을 구성하여 비정상 스팟 인스턴스를 교체할 수 있습니다. 상태 확인 교체를 사용하도록 설정한 이후 스팟 인스턴스는 unhealthy로 보고될 때 교체됩니다. 플릿은 비정상 스팟 인스턴스가 교체되는 동안 최대 몇 분간 목표 용량을 밑돌 수 있습니다.

요구 사항

- 상태 확인 대체는 목표 용량을 유지하는스팟 집합(maintain 유형 플릿)에만 지원되며 일회성 스팟 집합(request 유형 플릿)에는 지원되지 않습니다.
- 상태 확인 교체는 스팟 인스턴스에 대해서만 지원됩니다. 이 기능은 온디맨드 인스턴스에 대해 지원되지 않습니다.
- 비정상 인스턴스를 생성할 경우에만 이를 교체하도록 스팟 플릿을 구성할 수 있습니다.
- 사용자는 ec2:DescribeInstanceStatus 작업을 호출할 권한이 있는 경우에만 상태 확인 대체를 사용할 수 있습니다.

Console

비정상 스팟 인스턴스를 교체하도록 스팟 플릿을 구성하는 방법

1. 정의된 파라미터를 사용하여 스팟 플릿 요청 생성의 스팟 플릿 생성 단계를 따릅니다.
2. 비정상 스팟 인스턴스를 교체하도록 플릿을 구성하려면 추가 시작 파라미터를 확장하고 상태 확인에서 비정상 인스턴스 교체를 선택합니다. 이 옵션을 활성화하려면 먼저 대상 용량 유지를 선택해야 합니다.

AWS CLI

비정상 스팟 인스턴스를 교체하도록 스팟 플릿을 구성하는 방법

ReplaceUnhealthyInstances 속성이 SpotFleetRequestConfig인 request-spot-fleet 명령을 사용합니다.

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "lowestPrice",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
```

```
        "TargetCapacity": 10,
        "ReplaceUnhealthyInstances": true
    }
}
```

PowerShell

비정상 스팟 인스턴스를 교체하도록 스팟 플릿 요청을 구성하는 방법

Request-EC2SpotFleet cmdlet을 -SpotFleetRequestConfig_ReplaceUnhealthyInstance 파라미터와 함께 사용합니다.

```
-SpotFleetRequestConfig_ReplaceUnhealthyInstance $true
```

신규 또는 기존 스팟 플릿 요청과 이를 통해 시작되는 인스턴스 및 볼륨 태그 지정

스팟 플릿 요청과 이를 통해 시작되는 인스턴스와 볼륨을 분류 및 관리하는 데 도움이 되도록 사용자 지정 메타데이터로 이 요청에 태그를 지정할 수 있습니다. 스팟 플릿 요청을 생성할 때 또는 생성한 후 요청을 태깅할 수 있습니다. 마찬가지로 플릿에서 인스턴스와 볼륨이 시작될 때 또는 그 이후에 인스턴스와 볼륨에 태그를 할당할 수 있습니다.

플릿 요청에 태그를 지정할 때 플릿에서 시작한 인스턴스 및 볼륨에는 태그가 자동으로 지정되지 않습니다. 플릿에서 시작한 인스턴스 및 볼륨에 명시적으로 태그를 지정해야 합니다. 플릿 요청에만, 플릿에서 시작한 인스턴스에만, 플릿에서 시작한 인스턴스에 연결된 볼륨에만, 또는 모두에 태그를 할당하도록 선택할 수 있습니다.

> **Note**
>
> 온디맨드 인스턴스에 연결된 볼륨에만 태그를 지정할 수 있습니다. 스팟 인스턴스에 연결된 볼륨에는 태그를 지정할 수 없습니다.

Amazon EC2 콘솔이나 명령줄 도구를 사용하여 태그를 지정할 수 있습니다.

태그 작동 방식에 대한 자세한 내용은 Amazon EC2 리소스 태깅 섹션을 참조하세요.

내용

- 전제 조건

- 새 스팟 플릿과 해당 플릿에서 시작되는 인스턴스 및 볼륨 태깅
- 기존 스팟 플릿 태깅
- 스팟 플릿 요청 태그 보기

전제 조건

사용자에게 리소스에 태그를 지정할 수 있는 권한을 부여합니다. 자세한 내용은 예: 태그 리소스 섹션을 참조하세요.

사용자에게 리소스에 태그를 지정할 수 있는 권한 부여

다음을 포함하는 IAM 정책을 생성합니다.

- `ec2:CreateTags` 작업 사용자에게 태그 생성 권한이 부여됩니다.
- `ec2:RequestSpotFleet` 작업 사용자에게 스팟 플릿 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`을 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/*"
}
```

액세스 권한을 제공하려면 사용자, 그룹 또는 역할에 권한을 추가하세요:

- AWS IAM Identity Center의 사용자 및 그룹:

  권한 세트를 생성합니다. AWS IAM Identity Center 사용 설명서의 권한 세트 생성의 지침을 따릅니다.
- 보안 인증 공급자를 통해 IAM에서 관리되는 사용자:

  ID 페더레이션을 위한 역할을 생성합니다. IAM 사용 설명서의 Create a role for a third-party identity provider (federation)의 지침을 따릅니다.
- IAM 사용자:
  - 사용자가 맡을 수 있는 역할을 생성합니다. IAM 사용 설명서에서 Create a role for an IAM user의 지침을 따릅니다.
  - (권장되지 않음)정책을 사용자에게 직접 연결하거나 사용자를 사용자 그룹에 추가합니다. IAM 사용 설명서에서 사용자(콘솔)에 권한 추가의 지침을 따르세요.

새 스팟 플릿과 해당 플릿에서 시작되는 인스턴스 및 볼륨 태깅

콘솔을 사용하여 새 스팟 플릿 요청 및 해당 스팟 플릿이 시작하는 인스턴스와 볼륨에 태그를 지정하려면 다음을 수행하세요.

1. 정의된 파라미터를 사용하여 스팟 플릿 요청 생성의 절차를 따르세요.
2. 태그를 추가하는 방법은 플릿을 수동으로 구성했는지, 시작 템플릿을 사용했는지에 따라 달라집니다.

   - 수동으로 플릿을 구성한 경우 다음을 수행합니다.

태그를 추가하려면 추가 시작 파라미터를 확장하고 태그 생성을 선택한 다음 태그의 키와 값을 입력합니다. 각 태그에 대해 반복합니다.

각 태그에 대해 동일한 태그로 스팟 플릿 요청과 인스턴스를 태깅할 수 있습니다. 인스턴스와 요청에 모두 태그를 지정하려면 인스턴스와 플릿이 모두 선택되어 있는지 확인합니다. 스팟 플릿 요청만 태깅하려면 인스턴스(Instance) 선택을 취소합니다. 플릿에서 시작한 인스턴스만 태깅하려면 플릿(Fleet)을 선택 취소합니다.

> **Note**
>
> 수동으로 플릿을 구성하는 경우 볼륨에 태그를 지정하는 옵션이 없습니다. 볼륨 태그는 온디맨드 인스턴스에 연결된 볼륨에 대해서만 지원됩니다. 수동으로 플릿을 구성하는 경우 온디맨드 인스턴스를 지정할 수 없습니다.

- 시작 템플릿을 사용한 경우 다음을 수행합니다.

  플릿 요청에 태그를 추가하려면 태그에서 태그 생성을 선택하고 태그 키와 값을 입력합니다. 각 태그에 대해 반복합니다.

  플릿의 리소스에 태그를 지정하려면 시작 템플릿에서 태그를 지정해야 합니다.

AWS CLI를 사용하여 새 스팟 플릿 요청 및 해당 스팟 플릿이 시작하는 인스턴스 및 볼륨에 태깅

스팟 플릿 요청을 생성할 때 태깅하고 해당 스팟 플릿에서 시작할 때 인스턴스 및 볼륨을 태깅하려면 스팟 플릿 요청 구성을 다음과 같이 구성합니다.

스팟 플릿 요청 태그:

- `SpotFleetRequestConfig`에서 스팟 플릿 요청에 대한 태그를 지정합니다.
- `ResourceType`에 `spot-fleet-request`을 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 `instance`을 지정합니다. 다른 값을 지정하면 플릿 요청이 실패합니다.

- Tags에 대해 키-값 페어를 지정합니다. 둘 이상의 키-값 페어를 지정할 수 있습니다.

  또는 스팟 플릿 요청에서 참조되는 시작 템플릿에서 인스턴스에 대한 태그를 지정할 수 있습니다.

볼륨 태그:

- 스팟 플릿 요청에서 참조되는 시작 템플릿의 볼륨에 대한 태그를 지정합니다. LaunchSpecifications에서의 볼륨 태그 지정은 지원되지 않습니다.

다음 예에서 스팟 플릿 요청은 Key=Environment 및 Value=Production, Key=Cost-Center 및 Value=123이라는 2개의 태그로 태깅됩니다. 플릿에서 시작한 인스턴스는 스팟 플릿 요청의 태그 중 하나와 동일한 Key=Cost-Center 및 Value=123 태그 1개로만 태깅됩니다.

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "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 플릿에서 시작한 인스턴스에 태깅

플릿에서 시작되는 인스턴스를 태깅하려면 스팟 플릿 요청에서 참조되는 시작 템플릿에서 태그를 지정하거나 다음과 같이 스팟 플릿 요청 구성에서 태그를 지정할 수 있습니다.

- `LaunchSpecifications`의 인스턴스에 대한 태그를 지정합니다.
- `ResourceType`에 `instance`을 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": "priceCapacityOptimized",
        "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 스팟 플릿에서 시작한 온디맨드 인스턴스에 연결된 볼륨에 태깅

플릿에서 생성하는 볼륨을 태깅하려면 스팟 플릿 요청에서 참조되는 시작 템플릿에서 태그를 지정합니다.

> **Note**
>
> 볼륨 태그는 온디맨드 인스턴스에 연결된 볼륨에 대해서만 지원됩니다. 스팟 인스턴스에 연결된 볼륨에는 태그를 지정할 수 없습니다.
> `LaunchSpecifications`에서의 볼륨 태그 지정은 지원되지 않습니다.

기존 스팟 플릿 태깅

콘솔을 사용하여 기존 스팟 플릿 요청을 태깅하려면

스팟 플릿 요청을 생성한 후 콘솔을 사용하여 플릿 요청에 태그를 추가할 수 있습니다.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. 스팟 플릿 요청을 선택합니다.
4. 태그 탭을 선택하고 태그 생성을 선택합니다.

AWS CLI를 사용하여 기존 스팟 플릿 요청에 태깅

create-tags 명령을 사용해 기존 리소스에 태그를 지정할 수 있습니다. 다음 예에서, 기존 스팟 플릿 요청은 Key=purpose 및 Value=test로 태깅됩니다.

```
aws ec2 create-tags \
    --resources sfr-11112222-3333-4444-5555-66666EXAMPLE \
    --tags Key=purpose,Value=test
```

스팟 플릿 요청 태그 보기

콘솔을 사용하여 스팟 플릿 요청 태그를 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. 스팟 플릿 요청을 선택하고 [태그(Tags)] 탭을 선택합니다.

스팟 플릿 요청 태그를 설명하려면

describe-tags 명령을 사용하여 지정된 리소스에 대한 태그를 표시합니다. 다음 예제에서는 지정된 스팟 플릿 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 플릿 요청을 설명하여 스팟 플릿 요청의 태그를 볼 수도 있습니다.

describe-spot-fleet-requests 명령을 사용하여 지정된 스팟 플릿 요청의 구성을 볼 수 있습니다. 여기에는 플릿 요청에 대해 지정된 태그가 모두 포함됩니다.

```
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 플릿 요청, 해당 인스턴스, 이벤트 기록 설명

스팟 플릿 구성, 스팟 플릿의 인스턴스 및 스팟 플릿의 이벤트 기록을 설명할 수 있습니다.

Console

스팟 플릿을 설명하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. 스팟 플릿 요청을 선택합니다. ID는 sfr- 시작합니다. 구성 세부 정보를 보려면 설명을 선택합니다.
4. 스팟 플릿에 대한 스팟 인스턴스를 나열하려면 [인스턴스(Instances)]를 선택합니다.
5. 스팟 플릿에 대한 기록을 보려면 [기록(History)]을 선택합니다.

AWS CLI

스팟 플릿 요청을 설명하려면

describe-spot-fleet-requests 명령을 사용합니다.

```
aws ec2 describe-spot-fleet-requests \
    --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

지정된 스팟 플릿 요청에 대한 실행 중인 인스턴스를 설명하려면

describe-spot-fleet-instances 명령을 사용합니다.

```
aws ec2 describe-spot-fleet-instances \
    --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

지정된 스팟 플릿 요청에 대한 이벤트 기록을 설명하려면

describe-spot-fleet-request-history 명령을 사용합니다.

```
aws ec2 describe-spot-fleet-request-history \
    --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --start-time 2024-05-18T00:00:00Z
```

PowerShell

스팟 플릿 요청을 설명하려면

Get-EC2SpotFleetRequest cmdlet을 사용합니다.

```
Get-EC2SpotFleetRequest
```

지정된 스팟 플릿 요청에 대한 실행 중인 인스턴스를 설명하려면

Get-EC2SpotFleetInstance cmdlet을 사용합니다.

```
Get-EC2SpotFleetInstance `
    -SpotFleetRequestId "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
```

지정된 스팟 플릿 요청에 대한 이벤트 기록을 설명하려면

Get-EC2SpotFleetRequestHistory cmdlet을 사용합니다.

```
Get-EC2SpotFleetRequestHistory `
    -SpotFleetRequestId "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -UtcStartTime 2024-05-18T00:00:00Z
```

스팟 플릿 요청 수정

다음 태스크를 완료하기 위해 활성 스팟 플릿 요청을 수정할 수 있습니다.

- 총 목표 용량 및 온디맨드 부분 증가
- 총 목표 용량 및 온디맨드 부분 감소

총 목표 용량을 늘리면 스팟 플릿이 스팟 플릿 요청에 대한 할당 전략에 따라 추가 스팟 인스턴스를 시작합니다. 온디맨드 부분을 늘리면 스팟 플릿이 추가 온디맨드 인스턴스를 시작합니다.

총 목표 용량을 줄이면 스팟 플릿이 새 목표 용량을 초과하는 모든 열린 요청을 취소합니다. 플릿의 크기가 새 목표 용량에 도달할 때까지 스팟 플릿에서 스팟 인스턴스를 종료하도록 요청할 수 있습니다. 할당 전략이 `diversified`이면 스팟 플릿이 풀 전체의 인스턴스를 종료합니다. 또는 스팟 플릿에서 플릿을 현재 크기로 유지하되 중단되거나 수동으로 종료한 스팟 인스턴스는 교체하지 않도록 요청할 수 있습니다.

고려 사항

- 일회성 스팟 플릿 요청은 수정할 수 없습니다. 스팟 플릿 요청을 생성할 때 [목표 용량 유지(Maintain target capacity)]를 선택한 경우에만 스팟 플릿 요청을 수정할 수 있습니다.
- 목표 용량이 줄어 스팟 플릿이 인스턴스를 종료하면 해당 인스턴스는 스팟 인스턴스 중단 공지를 받습니다.

Console

스팟 플릿 요청을 수정하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. 스팟 플릿 요청을 선택합니다.
4. 작업을 선택한 다음, Modify target capacity(목표 용량 수정)을 선택합니다.
5. 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 CLI

스팟 플릿 요청을 수정하려면

modify-spot-fleet-request 명령을 사용하여 지정된 스팟 플릿 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 플릿의 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
```

PowerShell

스팟 플릿 요청을 수정하려면

Edit-EC2SpotFleetRequest cmdlet을 사용하여 지정된 스팟 플릿 요청의 목표 용량을 업데이트합니다.

```
Edit-EC2SpotFleetRequest `
    -SpotFleetRequestId "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TargetCapacity 20
```

다음과 같이 이전 명령을 수정하여 결과적으로 어떤 스팟 인스턴스도 종료하지 않고 지정된 스팟 플릿의 목표 용량을 줄일 수 있습니다.

```
Edit-EC2SpotFleetRequest `
    -SpotFleetRequestId "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TargetCapacity 20 `
    -ExcessCapacityTerminationPolicy "NoTermination"
```

스팟 플릿 요청 취소(삭제)

스팟 플릿이 더 이상 필요하지 않은 경우 요청을 스팟 플릿 요청을 취소할 수 있습니다. 그러면 요청이 삭제됩니다. 플릿 요청을 취소하면 플릿과 연결된 모든 스팟 요청도 취소되어 새 스팟 인스턴스가 시작되지 않습니다.

스팟 플릿 요청을 취소하는 경우 해당 인스턴스도 모두 종료할지 여부를 지정해야 합니다. 여기에는 온디맨드 인스턴스와 스팟 인스턴스가 모두 포함됩니다.

플릿 요청이 취소되면 인스턴스가 종료되도록 지정할 경우 플릿 요청이 `cancelled_terminating` 상태가 됩니다. 그렇게 하지 않으면 `cancelled_running` 상태가 되어 인스턴스가 중단되거나 수동으로 종료될 때까지 계속 실행됩니다.

제한 사항

- 단일 요청으로 최대 100개의 플릿을 취소할 수 있습니다. 지정된 수를 초과하면 플릿이 취소되지 않습니다.

Console

스팟 플릿 요청을 취소(삭제)하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. 스팟 플릿 요청을 선택합니다.
4. 작업을 선택한 후, 요청 취소를 선택합니다.
5. 스팟 요청 취소 대화 상자에서 다음을 수행합니다.
   a. 스팟 플릿 요청을 취소함과 동시에 연결된 인스턴스를 종료하려면 인스턴스 종료 확인란을 선택된 상태로 둡니다. 연결된 인스턴스를 종료하지 않고 스팟 플릿 요청을 취소하려면 인스턴스 종료 확인란 선택을 취소합니다.
   b. 확인을 선택합니다.

AWS CLI

스팟 플릿 요청을 취소(삭제)하고 해당 인스턴스를 종료하려면

cancel-spot-fleet-requests 명령을 `--terminate-instances` 옵션과 함께 사용합니다.

```
aws ec2 cancel-spot-fleet-requests \
    --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --terminate-instances
```

출력의 예시는 다음과 같습니다.

```
{
    "SuccessfulFleetRequests": [
        {
            "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
            "CurrentSpotFleetRequestState": "cancelled_terminating",
            "PreviousSpotFleetRequestState": "active"
        }
    ],
    "UnsuccessfulFleetRequests": []
}
```

해당 인스턴스를 종료하지 않고 스팟 플릿 요청을 취소(삭제)하려면

`--no-terminate-instances` 옵션을 대신 사용하여 이전 예제를 수정합니다.

```
aws ec2 cancel-spot-fleet-requests \
    --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --no-terminate-instances
```

출력의 예시는 다음과 같습니다.

```
{
    "SuccessfulFleetRequests": [
        {
            "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
            "CurrentSpotFleetRequestState": "cancelled_running",
            "PreviousSpotFleetRequestState": "active"
        }
    ],
    "UnsuccessfulFleetRequests": []
}
```

PowerShell

스팟 플릿 요청을 취소(삭제)하고 해당 인스턴스를 종료하려면

Stop-EC2SpotFleetRequest cmdlet을 `-TerminateInstance` 파라미터와 함께 사용합니다.

```
Stop-EC2SpotFleetRequest `
    -SpotFleetRequestId "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TerminateInstance $true
```

해당 인스턴스를 종료하지 않고 스팟 플릿 요청을 취소(삭제)하려면

`-TerminateInstance` 파라미터 값을 변경하여 이전 예제를 수정합니다.

```
Stop-EC2SpotFleetRequest `
    -SpotFleetRequestId "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TerminateInstance $false
```

스팟 플릿의 자동 규모 조정 이해

자동 규모 조정을 사용하면 수요에 따라 스팟 플릿의 대상 용량을 늘리거나 줄일 수 있습니다. 자동 규모 조정을 사용하면 스팟 플릿은 하나 이상의 조정 정책에 따라 지정된 범위 내에서 인스턴스를 시작(스케일 아웃)하거나 인스턴스를 종료(스케일 인)할 수 있습니다.

스팟 플릿의 자동 크기 조정은 Amazon EC2, Amazon CloudWatch 및 Application Auto Scaling API의 조합을 통해 수행됩니다. 스팟 플릿은 Amazon EC2를 통해 생성됩니다. CloudWatch는 경보를 생성합니다. 크기 조정 정책은 Application Auto Scaling을 통해 생성됩니다.

자동 규모 조정 유형

스팟 플릿은 다음과 같은 유형의 자동 크기 조정을 지원합니다.

- 대상 추적 조정 - 특정 지표에 대한 값을 대상으로 지정하여 플릿의 현재 용량을 늘리거나 줄입니다. 이 작업은 온도 조절기가 집안 온도를 유지하는 방식과 비슷합니다. 사용자가 원하는 온도만 선택하면 나머지는 온도 조절기가 알아서 합니다.
- 단계 크기 조정 - 일련의 크기 조정 조절값(즉, 경보 위반의 크기에 따라 달라지는 단계 조절값)에 따라 플릿의 현재 용량을 늘리거나 줄입니다.
- 예약 크기 조정 - 날짜 및 시간을 기준으로 플릿의 현재 용량을 늘리거나 줄입니다.

고려 사항

스팟 플릿에 자동 규모 조정을 사용하는 경우 다음 고려 사항을 참조하세요.

- 인스턴스 가중치 - 인스턴스 가중치를 사용하는 경우 스팟 플릿에서 필요에 따라 대상 용량을 초과할 수 있다는 점에 주의하세요. 이행된 용량은 부동 소수점 숫자일 수 있으나 목표 용량은 정수여야 하므로 스팟 플릿은 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 가정하세요. 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 플릿이 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.

- 휴지 기간 - 스팟 플릿에 대해 생성된 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.
- 세부 모니터링 사용 - 인스턴스 지표를 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 EC2 인스턴스에 대한 세부 모니터링 관리 및 정의된 파라미터를 사용하여 스팟 플릿 요청 생성(을)를 참조하세요.
- AWS CLI - 스팟 플릿의 규모 조정을 구성하기 위해 AWS CLI를 사용하는 경우 application-autoscaling 명령을 사용하게 됩니다.

스팟 플릿 자동 크기 조정에 필요한 IAM 권한

스팟 플릿의 자동 크기 조정은 Amazon EC2, Amazon CloudWatch 및 Application Auto Scaling API의 조합을 통해 수행됩니다. 스팟 플릿은 Amazon EC2를 통해 생성됩니다. CloudWatch는 경보를 생성합니다. 크기 조정 정책은 Application Auto Scaling을 통해 생성됩니다. 플릿 규모 조정 설정에 액세스하는 사용자에게는 스팟 플릿 및 Amazon EC2를 사용하는 데 필요한 IAM 권한에 더해 자동 규모 조정을 지원하는 서비스에 대한 적절한 권한이 있어야 합니다.

스팟 플릿에 자동 규모 조정을 사용하려면 사용자에게는 다음 예제 정책에 나온 태스크를 수행할 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 내용은 Application Auto Scaling 사용 설명서의 Identity and Access Management for Application Auto Scaling을 참조하세요.

Application Auto Scaling 서비스에는 스팟 플릿 및 CloudWatch 경보를 설명할 수 있는 권한과 사용자 대신 스팟 플릿 목표 용량을 수정할 수 있는 권한도 필요합니다. 스팟 플릿에 자동 크기 조정을 사용하면 `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 Scaling의 관리형 스팟 플릿 역할은 원래 `aws-ec2-spot-fleet-autoscale-role`이었지만 더 이상 필요하지 않습니다. 서비스 연결 역할은 Application Auto Scaling의 기본 역할입니다. 자세한 정보는 Application Auto Scaling 사용 설명서의 Application Auto Scaling 서비스 연결 역할을 참조하세요.

대상 추적 조정: 특정 지표에 대한 값을 대상으로 지정하여 스팟 플릿 규모 조정

대상 추적 조정을 사용하면 지표를 선택하고 대상 값을 설정하여 대상 추적 조정 정책을 생성할 수 있습니다. 스팟 플릿은 조정 정책을 트리거하고 선택한 지표 및 목표 값에 따라 조정 조절값을 계산하는

CloudWatch 경보를 생성하고 관리합니다. 조정 정책은 지표를 지정된 대상 값에 가깝게 유지하기 위해 필요에 따라 인스턴스를 추가하거나 제거하여 용량을 조정합니다. 대상 추적 정책은 지표를 대상 값에 가깝게 유지할 뿐만 아니라 로드 패턴의 변동으로 인한 지표의 변동에 맞춰 조정하고 급격한 용량 변동을 최소화합니다.

각 정책이 다른 지표를 사용한다는 전제하에 스팟 플릿에 대해 다수의 대상 추적 조정 정책을 생성할 수 있습니다. 플릿은 최대 플릿 용량을 지정하는 정책에 따라 조정됩니다. 이를 통해 여러 시나리오에 대응하여 애플리케이션 워크로드에 충분한 용량을 확보할 수 있습니다.

애플리케이션 가용성을 보장하기 위해 스팟 플릿은 측정치에 비례하여 가능한 신속하게 확장되지만, 축소는 점진적으로 이루어집니다.

대상 용량이 줄어 스팟 플릿이 스팟 인스턴스를 종료하면 해당 인스턴스는 스팟 인스턴스 중단 공지를 받습니다.

> **Note**
>
> 스팟 플릿의 대상 추적 조정 정책에서 관리되는 CloudWatch 경보는 편집하거나 삭제하지 마세요. 대상 추적 조정 정책을 삭제하면 스팟 플릿에서 경보가 자동으로 삭제됩니다.

사전 조건

- 스팟 플릿 요청에는 `maintain` 유형의 요청이 있어야 합니다. `request` 유형의 요청에는 자동 조정이 지원되지 않습니다.
- 스팟 플릿 자동 크기 조정에 필요한 IAM 권한 구성.
- 고려 사항을(를) 검토합니다.

대상 추적 조정 정책을 구성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. 스팟 플릿 요청을 선택합니다.
4. 화면 하단 근처의 Auto Scaling 탭을 선택합니다. 스팟 플릿에 대한 링크를 선택한 경우에는 탭이 없습니다. 대신 Auto Scaling 섹션이 나타날 때까지 아래로 스크롤합니다.
5. 자동 조정이 구성되어 있지 않으면 구성을 선택합니다.

6. 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.
7. [정책 이름(Policy name)]에 정책의 이름을 입력합니다.
8. 대상 지표를 선택합니다.
9. 측정치에 대한 대상 값을 입력합니다.
10. 휴지 기간의 경우 새 값(초)을 지정하거나 기본값을 유지합니다.
11. (선택 사항) 현재 구성에 따라 스케일 인 정책 생성을 생략하려면 축소 비활성화를 선택합니다. 다른 구성을 사용하여 축소 정책을 생성할 수 있습니다.
12. 저장을 선택합니다.

AWS CLI를 사용하여 대상 추적 정책을 구성하려면

1. register-scalable-target 명령을 사용하여 스팟 플릿 요청을 확장 가능한 대상으로 등록합니다.
2. put-scaling-policy 명령을 사용하여 조정 정책을 생성합니다.

단계 조정: 단계 조정 정책을 사용하여 스팟 플릿 규모 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 단위 또는 현재 용량의 지정된 비율까지 플릿의 목표 용량을 늘립니다.
- 제거 - 지정된 수의 용량 단위 또는 현재 용량의 지정된 비율까지 플릿의 목표 용량을 줄입니다.
- 설정 - 플릿의 목표 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 가정하세요. 경보가 트리거되면 자동 크기 조정 프로세스에서 10에 1을 추가하여 11이 되므로 스팟 플릿은 1개 인스턴스를 시작합니다.

대상 용량이 줄어 스팟 플릿이 스팟 인스턴스를 종료하면 해당 인스턴스는 스팟 인스턴스 중단 공지를 받습니다.

사전 조건

- 스팟 플릿 요청에는 `maintain` 유형의 요청이 있어야 합니다. `request` 유형의 요청에는 자동 조정이 지원되지 않습니다.
- 스팟 플릿 자동 크기 조정에 필요한 IAM 권한 구성.
- 어떤 CloudWatch 지표가 애플리케이션에 중요한지 생각하세요. AWS에서 제공하는 지표 또는 사용자 지정 지표를 기반으로 CloudWatch 경보를 생성할 수 있습니다.
- 크기 조정 정책에 사용할 AWS 지표에 대한 CloudWatch 지표 수집이 지표를 제공하는 서비스에서 기본적으로 사용되지 않은 경우 지표 수집을 사용하도록 설정합니다.
- 고려 사항을(를) 검토합니다.

CloudWatch 경보를 생성하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 경보를 확장하고 모든 경보를 선택합니다.
3. 경보 생성을 선택합니다.
4. 지표 및 조건 지정(Specify metric and conditions) 페이지에서 지표 선택(Select metric)을 선택합니다.
5. EC2 스팟, 플릿 요청 지표를 선택하고, 지표(예: TargetCapacity)를 선택한 다음 지표 선택을 선택합니다.

   선택한 지표에 대한 그래프와 기타 정보가 표시된 Specify metric and conditions(지표 및 조건 지정) 페이지가 나타납니다.
6. 기간에 예컨대, 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 참조하세요.

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 설명은 Amazon CloudWatch 사용자 가이드의 CloudWatch 경보가 누락된 데이터를 처리하는 방법 구성을 참조하세요.
10. 다음을 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. 다음을 선택합니다.
13. 이름 및 설명 추가에 경보의 이름과 설명을 입력하고 다음을 선택합니다.
14. 경보 생성(Create alarm)을 선택하세요.

스팟 플릿에 대한 단계 조정 정책을 구성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. 스팟 플릿 요청을 선택합니다.
4. 화면 하단 근처의 Auto Scaling 탭을 선택합니다. 스팟 플릿에 대한 링크를 선택한 경우에는 탭이 없습니다. 대신 Auto Scaling 섹션이 나타날 때까지 아래로 스크롤합니다.
5. 자동 조정이 구성되어 있지 않으면 구성을 선택합니다.
6. 다음 사이로 용량 조정(Scale capacity between)을 사용하여 플릿에 대한 최소 및 최대 용량을 설정합니다. 조정 정책에서는 최소 용량 미만이나 최대 용량을 초과하여 플릿을 조정하지 않습니다.
7. 조정 정책 아래의 정책 유형에서 단계 조정 정책을 선택합니다.
8. 처음에 조정 정책에는 단계 조정 정책(ScaleUp 및 ScaleDown)이 포함되어 있습니다. 이러한 정책을 완료하거나 정책 제거(Remove policy)를 선택하여 삭제할 수 있습니다. 또한 정책 추가를 선택하여 정책을 추가할 수도 있습니다.
9. 정책을 정의하려면 다음을 수행합니다.

    a. [정책 이름(Policy name)]에 정책의 이름을 입력합니다.

    b. 정책 트리거에서 기존 경보를 선택하거나 새 경보 생성을 선택하여 Amazon CloudWatch 콘솔을 열고 경보를 생성합니다.

    c. 용량 수정에서 조정할 크기 및 단계 조정의 상한과 하한을 정의합니다. 특정 인스턴스 개수나는 기존 플릿 크기의 백분율을 추가 또는 제거하거나 플릿을 정확한 크기로 설정할 수 있습니다.

예를 들어 플릿의 용량을 30% 늘리는 단계 조정 정책을 생성하려면 추가를 선택하고 다음 필드에 30을 입력한 후 퍼센트를 선택합니다. 기본적으로 추가 정책의 하한은 경보 임계값이고 상한은 양(+)의 무한대입니다. 기본적으로 제거 조정의 상한은 경보 임계값이고 하한은 음(-)의 무한대입니다.

d. (선택 사항) 다른 단계를 추가하려면 단계 추가를 선택합니다.

e. 휴지 기간의 경우 새 값(초)을 지정하거나 기본값을 유지합니다.

10. 저장을 선택합니다.

AWS CLI를 사용하여 스팟 플릿에 대한 단계 크기 조정 정책 구성

1. register-scalable-target 명령을 사용하여 스팟 플릿 요청을 확장 가능한 대상으로 등록합니다.
2. put-scaling-policy 명령을 사용하여 조정 정책을 생성합니다.
3. put-metric-alarm 명령을 사용하여 조정 정책을 트리거하는 경보를 생성합니다.

예약된 규모 조정: 일정에 따라 스팟 플릿 규모 조정

일정을 기반으로 플릿 규모를 조정하면 수요에 따른 로드 변경에 맞게 애플리케이션 규모를 조정할 수 있습니다. 예약된 작업을 생성하여 스팟 플릿에서 특정 시간에 규모 조정 작업을 수행하도록 할 수 있습니다. 예약 작업을 생성하려면 기존 스팟 플릿, 규모 조정 활동이 발생해야 하는 시간, 원하는 최소 용량 및 최대 용량을 지정해야 합니다. 한 번 또는 반복되는 일정에 따라 크기가 조정되도록 예약된 작업을 구성할 수 있습니다. 변경이 필요한 경우 예약된 작업을 편집하거나 삭제할 수 있습니다.

사전 조건

- 기존 스팟 플릿에 대해서만 예약된 작업을 생성할 수 있습니다. 스팟 플릿을 생성할 때는 예약된 작업을 생성할 수 없습니다.
- 스팟 플릿 요청에는 `maintain` 유형의 요청이 있어야 합니다. `request` 유형의 요청에는 자동 조정이 지원되지 않습니다.
- 스팟 플릿 자동 크기 조정에 필요한 IAM 권한 구성.
- 고려 사항을(를) 검토합니다.

1회성 예약된 작업을 만들려면

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

3. 스팟 플릿 요청을 선택합니다.
4. 화면 하단 근처의 예약된 조정 탭을 선택합니다. 스팟 플릿에 대한 링크를 선택한 경우에는 탭이 없습니다. 대신 예약된 조정 섹션이 나타날 때까지 아래로 스크롤합니다.
5. 예약된 작업 생성을 선택합니다.
6. 이름에서 예약된 작업의 이름을 지정합니다.
7. 최소 용량, 최대 용량 또는 두 가지 모두 입력합니다.
8. 반복에서 1회(Once)를 선택합니다.
9. (선택 사항) 시작 시간, 종료 시간 또는 두 가지 모두에 대해 날짜와 시간을 선택합니다.
10. 생성(Create)을 선택합니다.

반복되는 예약된 작업을 생성하려면 다음을 수행하세요.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스팟 요청을 선택합니다.
3. 스팟 플릿 요청을 선택합니다.
4. 화면 하단 근처의 예약된 조정 탭을 선택합니다. 스팟 플릿에 대한 링크를 선택한 경우에는 탭이 없습니다. 대신 예약된 조정 섹션이 나타날 때까지 아래로 스크롤합니다.
5. 이름에서 예약된 작업의 이름을 지정합니다.
6. 최소 용량, 최대 용량 또는 두 가지 모두 입력합니다.
7. [반복(Recurrence)]에서 사전 정의된 일정(예: [매일(Every day)]) 중 하나를 선택하거나 [사용자 지정(Custom)]을 선택하고 cron 표현식을 입력합니다. 예약된 조정에서 지원하는 cron 표현식에 대한 자세한 내용은 Amazon EventBridge 사용자 가이드의 Cron expressions를 참조하세요.
8. (선택 사항) 시작 시간, 종료 시간 또는 두 가지 모두에 대해 날짜와 시간을 선택합니다.
9. 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 플릿 요청을 선택합니다.
4. 화면 하단 근처의 예약된 조정 탭을 선택합니다. 스팟 플릿에 대한 링크를 선택한 경우에는 탭이 없습니다. 대신 예약된 조정 섹션이 나타날 때까지 아래로 스크롤합니다.

5. 예약된 작업을 선택한 다음, 작업, 편집을 선택합니다.
6. 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 플릿 요청을 선택합니다.
4. 화면 하단 근처의 예약된 조정 탭을 선택합니다. 스팟 플릿에 대한 링크를 선택한 경우에는 탭이 없습니다. 대신 예약된 조정 섹션이 나타날 때까지 아래로 스크롤합니다.
5. 예약된 작업을 선택한 다음, 작업, 삭제를 선택합니다.
6. 확인 메시지가 나타나면 삭제를 선택합니다.

AWS CLI를 사용하여 예약된 조정을 관리하려면

다음 명령을 사용합니다.

- put-scheduled-action
- describe-scheduled-actions
- delete-scheduled-action

EC2 플릿 또는 스팟 플릿 모니터링

최적의 성능을 유지하고 신뢰성을 보장하려면 EC2 플릿 또는 스팟 플릿을 효과적으로 모니터링하는 것이 필수적입니다. 이 주제에서 다루는 Amazon CloudWatch와 Amazon EventBridge를 비롯하여 이를 달성하는 데 도움이 되는 다양한 도구가 있습니다.

CloudWatch를 사용하여 지표를 수집 및 추적하고, 경보를 설정하고, 플릿 상태의 변화에 자동으로 대응할 수 있습니다.

EventBridge를 사용하면 플릿에서 발생하는 이벤트를 모니터링하고 프로그래밍 방식으로 대응할 수 있습니다. EventBridge에서 규칙을 정의하면 인스턴스 종료 또는 플릿 상태 변경과 같은 특정 플릿 이벤트에 대한 응답을 자동화하여 운영 효율성을 개선할 수 있습니다.

주제

- CloudWatch를 사용하여 EC2 플릿 또는 스팟 플릿 모니터링

- Amazon EventBridge를 사용하여 EC2 플릿 또는 스팟 플릿에서 발생하는 이벤트를 모니터링하고 프로그래밍 방식으로 대응하세요.

CloudWatch를 사용하여 EC2 플릿 또는 스팟 플릿 모니터링

이 섹션에 설명하는 Amazon CloudWatch 지표를 사용하여 EC2 플릿 또는 스팟 플릿을 모니터링할 수 있습니다.

> **Important**
>
> 정확성을 보장하기 위해, 이 측정치를 사용할 때는 세부 모니터링을 활성화하는 것이 좋습니다. 자세한 내용은 EC2 인스턴스에 대한 세부 모니터링 관리 섹션을 참조하세요.

CloudWatch 사용에 관한 자세한 내용은 CloudWatch를 사용하여 인스턴스 모니터링의 내용을 참조하세요.

EC2 플릿 및 스팟 플릿 지표

AWS/EC2Spot 네임스페이스에는 플릿에 대한 다음 지표와 플릿의 스팟 인스턴스에 대한 CloudWatch 지표가 포함되어 있습니다. 자세한 내용은 인스턴스 지표 섹션을 참조하세요.

측정치	설명
AvailableInstancePoolsCount	플릿 요청에 지정된 스팟 용량 풀입니다. 단위: 개
BidsSubmittedForCapacity	Amazon EC2가 플릿 요청을 제출한 용량입니다. 단위: 개
EligibleInstancePoolCount	플릿 요청에 지정되어 Amazon EC2가 요청을 이행할 수 있는 스팟 용량 풀입니다. 사용자가 지불하려는 스팟 인스턴스 최대 가격이 스팟 가격보다 낮거나 스팟 가격이 온디맨드 인스턴스 가격보다 높은 풀에서는 Amazon EC2가 요청을 이행하지 않습니다. 단위: 개수

측정치	설명
FulfilledCapacity	Amazon EC2가 달성한 용량. 단위: 개
MaxPercentCapacityAllocation	플릿 요청에 지정된 모든 플릿 풀에 걸친 PercentCapacityAllocation 의 최댓값입니다. 단위: 백분율
PendingCapacity	TargetCapacity 와 FulfilledCapacity 의 차이점. 단위: 개수
PercentCapacityAllocation	지정된 차원의 스팟 용량 풀에 할당된 용량. 모든 스팟 용량 풀에 기록된 최댓값을 얻으려면 MaxPercentCapacityAllocation 을 사용하세요. 단위: 백분율
TargetCapacity	플릿 요청의 목표 용량입니다. 단위: 개
TerminatingCapacity	프로비저닝된 용량이 목표 용량보다 커서 종료되는 용량입니다. 단위: 개수

수치 측정 단위가 Count(수)인 경우, 가장 유용한 통계는 Average(평균)입니다.

EC2 플릿 및 스팟 플릿 차원

플릿에 대한 데이터를 필터링하려면 다음 차원을 사용합니다.

Dimensions	설명
AvailabilityZone	가용 영역별로 데이터를 필터링합니다.

Dimensions	설명
FleetRequestId	플릿 요청별로 데이터를 필터링합니다.
InstanceType	인스턴스 유형별로 데이터를 필터링합니다.

EC2 플릿 또는 스팟 플릿에 대한 CloudWatch 지표 보기

Amazon CloudWatch 콘솔을 사용하여 플릿에 대한 CloudWatch 지표를 볼 수 있습니다. 이 측정치들은 모니터링 그래프로 표시됩니다. 플릿이 활성 상태가 되면 이 그래프에 데이터 포인트가 표시됩니다.

측정치는 먼저 네임스페이스별로 그룹화된 다음, 각 네임스페이스 내에서 다양한 차원 조합별로 그룹화됩니다. 예를 들어 모든 플릿 지표를 보거나 플릿 요청 ID, 인스턴스 유형 또는 가용 영역별로 그룹화된 플릿 지표를 볼 수 있습니다.

플릿 지표 보기

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표를 확장한 다음 모든 지표를 선택합니다.
3. EC2 스팟 네임스페이스를 선택합니다.

   > **Note**
   >
   > EC2 스팟 네임스페이스가 표시되지 않는 경우 두 가지 이유가 있습니다. 리전에서 EC2 플릿 또는 스팟 플릿을 사용하지 않았습니다. 사용 중인 AWS 서비스만 Amazon CloudWatch에 지표를 보냅니다. 또는 리전에서 EC2 플릿 또는 스팟 플릿을 사용한 적이 있지만 지난 2주 동안 사용하지 않은 경우 네임스페이스가 표시되지 않습니다.

4. 측정치를 차원을 기준으로 필터링하려면 다음 중 하나를 선택하세요.
   - 플릿 요청 지표 – 플릿 요청으로 그룹화
   - 가용 영역 기준 – 플릿 요청 및 가용 영역으로 그룹화
   - 인스턴스 유형 기준 - 플릿 요청 및 인스턴스 유형으로 그룹화
   - 가용 영역/인스턴스 유형 기준 - 플릿 요청, 가용 영역 및 인스턴스 유형으로 그룹화
5. 지표의 데이터를 보려면 지표 옆의 확인란을 선택합니다.

Amazon EventBridge를 사용하여 EC2 플릿 또는 스팟 플릿에서 발생하는 이벤트를 모니터링하고 프로그래밍 방식으로 대응하세요.

EC2 플릿 또는 스팟 플릿 상태가 변경되면 플릿에서 알림을 보냅니다. 알림은 Amazon EventBridge로 전송되는 이벤트(이전의 Amazon CloudWatch Events) 형태로 제공됩니다. 이벤트는 최선의 작업을 기반으로 발생됩니다.

Amazon EventBridge를 사용하여 이벤트에 대한 응답으로 프로그래밍 동작을 트리거하는 규칙을 생성할 수 있습니다. 예를 들어, 두 개의 EventBridge 규칙을 생성할 수 있습니다. 하나는 플릿 상태가 변경될 때 트리거되고, 다른 하나는 플릿의 인스턴스가 종료될 때 트리거됩니다. 이 예제에서는 플릿 상태가 변경되면 규칙이 SNS 주제를 호출하여 사용자에게 이메일 알림을 보내도록 첫 번째 규칙을 구성할 수 있습니다. 플릿의 인스턴스가 종료되면 두 번째 규칙이 Lambda 함수를 호출하여 새 인스턴스를 시작하도록 두 번째 규칙을 구성할 수 있습니다.

> **Note**
>
> `maintain` 및 `request` 유형의 플릿에서만 이벤트가 생성됩니다. `instant` 유형 플릿은 동기식 일회성 요청을 제출하고 응답에서 플릿 상태를 즉시 알 수 있으므로 이벤트가 생성되지 않습니다. Amazon EventBridge를 사용하여 플릿 이벤트를 모니터링하려면 요청 유형이 `maintain` 또는 `request`여야 합니다.

플릿의 이벤트 기록을 설명하는 방법에 대한 지침은 EC2 플릿의 이벤트 기록 설명 섹션을 참조하세요.

주제

- EC2 플릿 또는 스팟 플릿 이벤트를 모니터링하는 Amazon EventBridge 규칙 생성
- EC2 집합 이벤트 유형
- 스팟 플릿 이벤트 유형

EC2 플릿 또는 스팟 플릿 이벤트를 모니터링하는 Amazon EventBridge 규칙 생성

EC2 플릿 또는 스팟 플릿에 대한 상태 변경 알림이 생성되면 해당 알림은 JSON 파일 형식의 이벤트로 Amazon EventBridge에 전송됩니다. EventBridge에서 규칙에 정의된 패턴과 일치하는 이벤트 패턴이 감지되는 경우, EventBridge는 규칙에 정의된 대상을 호출합니다.

일치하는 이벤트 패턴을 기반으로 작업을 자동화하는 EventBridge 규칙을 작성할 수 있습니다.

다음과 같은 이벤트 필드는 규칙에 정의된 이벤트 패턴을 형성합니다.

`"source": "aws.ec2fleet"`

EC2 집합에서 시작된 이벤트를 식별합니다.

`"detail-type": "EC2 Fleet State Change"`

이벤트 유형을 식별합니다.

`"detail": { "sub-type": "submitted" }`

이벤트 하위 유형을 식별합니다.

EC2 플릿 또는 스팟 플릿 이벤트 목록 및 이벤트 데이터 예는 EC2 집합 이벤트 유형 및 스팟 플릿 이벤트 유형 섹션을 참조하세요.

예시

- 알림을 보내는 EventBridge 규칙 생성
- Lambda 함수를 트리거하는 EventBridge 규칙 생성

알림을 보내는 EventBridge 규칙 생성

다음 예제에서는 Amazon EC2에서 EC2 플릿 상태 변경 알림이 생성될 때마다 이메일, 문자 메시지 또는 모바일 푸시 알림을 보내는 EventBridge 규칙을 생성합니다. 이 예제의 신호는 규칙에 정의된 작업을 트리거하는 `EC2 Fleet State Change` 이벤트로 생성됩니다.

전제 조건

EventBridge 규칙을 생성하기 전에 이메일, 문자 메시지 또는 모바일 푸시 알림에 대한 Amazon SNS 주제를 생성해야 합니다.

EC2 집합 상태가 변경될 때 알림을 보내는 EventBridge 규칙을 생성하려면

1. https://console.aws.amazon.com/events/에서 Amazon EventBridge 콘솔을 엽니다.
2. [규칙 생성(Create rule)]을 선택합니다.
3. 규칙 세부 정보 정의(Define rule detail)에 대해 다음을 수행하십시오:
   a. 규칙의 이름을 입력하고 선택적으로 설명을 입력합니다.

      규칙은 동일한 지역과 동일한 이벤트 버스의 다른 규칙과 동일한 이름을 가질 수 없습니다.

b. 이벤트 버스(Event bus)에서 기본값(default)을 선택합니다. 계정의 AWS 서비스가 이벤트를 생성하면 항상 계정의 기본 이벤트 버스로 이동합니다.

c. 규칙 유형(Rule type)에서 이벤트 패턴이 있는 규칙(Rule with an event pattern)을 생성합니다.

d. 다음을 선택합니다.

4. 이벤트 패턴 빌드(Build event pattern)에서 다음을 수행하십시오:

a. 이벤트 소스(Event source)에서 AWS 이벤트 또는 EventBridge 파트너 이벤트(events or EventBridge partner events)를 선택합니다.

b. 이벤트 패턴(Event pattern)의 경우 이 예에서는 `EC2 Fleet Instance Change` 이벤트와 일치하도록 다음 이벤트 패턴을 지정합니다.

```
{
 "source": ["aws.ec2fleet"],
 "detail-type": ["EC2 Fleet Instance Change"]
}
```

이벤트 패턴을 추가하려면 다음과 같이 이벤트 패턴 양식(Event pattern form)을 선택하여 템플릿을 사용하거나 사용자 정의 패턴(JSON 편집기)(Custom pattern (JSON editor))을 선택하여 고유한 패턴을 지정할 수 있습니다.

i. 템플릿을 사용하여 이벤트 패턴을 생성하려면 다음을 수행하세요.

A. 이벤트 패턴 양식(Event pattern form)을 선택합니다.

B. 이벤트 소스(Event source)에서 AWS 서비스(services)를 선택합니다.

C. AWS 서비스(Service)에서 EC2 플릿(EC2 Fleet)을 선택합니다.

D. 이벤트 유형(Event type)에서 EC2 플릿 인스턴스 변경(EC2 Fleet Instance Change)을 선택합니다.

E. 템플릿을 사용자 지정하려면 패턴 편집(Edit pattern)을 선택하고 예시 이벤트 패턴과 일치하도록 변경합니다.

ii. (대안) 사용자 정의 이벤트 패턴을 지정하려면 다음을 수행하세요.

A. 사용자 정의 패턴(JSON 편집기)을 선택합니다.

B. 이벤트 패턴(Event pattern) 상자에서 이 예시의 이벤트 패턴을 추가합니다.

c. 다음을 선택합니다.

5. 대상 선택(Select target(s))에서 다음을 수행합니다.

a. 대상 유형(Target types)에서 AWS 서비스(service)를 선택합니다.

b. 대상 선택(Select a target)에서 SNS 주제(SNS topic)를 선택하여 이벤트가 발생할 때 이메일, 문자 메시지 또는 모바일 푸시 알림을 보내도록 합니다.

c. [주제(Topic)]에서 기존 주제를 선택합니다. 먼저 Amazon SNS 콘솔을 사용하여 Amazon SNS 주제를 생성해야 합니다. 자세한 내용은 Amazon Simple Notification Service 개발자 안내서에서 A2P(Application-to-Person) 메시징에 Amazon SNS 사용을 참조하세요.

d. (선택 사항) 추가 설정(Additional settings)에서 선택적으로 추가 설정을 구성할 수 있습니다. 자세한 설명은 Amazon EventBridge 사용자 가이드의 Creating Amazon EventBridge rules that react to events(이벤트에 응답하는 Amazon EventBridge 규칙 생성)(16단계)를 참조하세요.

e. 다음을 선택합니다.

6. (선택 사항) 태그(Tags)에서 선택적으로 하나 이상의 태그를 규칙에 할당하고 다음(Next)을 선택할 수 있습니다.

7. 검토 및 생성(Review and create)에서 다음을 수행합니다.

a. 규칙의 세부 정보를 검토하고 필요에 따라 수정합니다.

b. 규칙 생성을 선택합니다.

자세한 내용은 Amazon EventBridge 사용 설명서에서 Amazon EventBridge 규칙 및 Amazon EventBridge 이벤트 패턴을 참조하세요.

Lambda 함수를 트리거하는 EventBridge 규칙 생성

다음 예제에서는 Amazon EC2에서 인스턴스가 시작되는 경우 EC2 플릿 인스턴스 변경 알림이 생성될 때마다 Lambda 함수를 트리거하는 EventBridge 규칙을 생성합니다. 이 예제의 신호는 규칙에서 정의한 작업을 트리거하는 하위 유형 `launched`인 `EC2 Fleet Instance Change` 이벤트로 전송됩니다.

EventBridge 규칙을 생성하기 전에 Lambda 함수를 생성해야 합니다.

EventBridge 규칙에서 사용 Lambda 함수 생성

1. https://console.aws.amazon.com/lambda/에서 AWS Lambda 콘솔을 엽니다.

2. 함수 생성(Create function)을 선택합니다.

3. 함수 이름을 입력하고 코드를 구성한 다음 [함수 생성(Create function)]을 선택합니다.

자세한 내용은 AWS Lambda 개발자 안내서의 첫 Lambda 함수 생성을 참조하세요.

EC2 집합의 인스턴스 상태가 변경될 때 Lambda 함수를 트리거하는 EventBridge 규칙을 생성하려면

1. https://console.aws.amazon.com/events/에서 Amazon EventBridge 콘솔을 엽니다.
2. [규칙 생성(Create rule)]을 선택합니다.
3. 규칙 세부 정보 정의(Define rule detail)에 대해 다음을 수행하십시오:
   a. 규칙의 이름을 입력하고 선택적으로 설명을 입력합니다.

      규칙은 동일한 지역과 동일한 이벤트 버스의 다른 규칙과 동일한 이름을 가질 수 없습니다.
   b. 이벤트 버스(Event bus)에서 기본값(default)을 선택합니다. 계정의 AWS 서비스가 이벤트를 생성하면 항상 계정의 기본 이벤트 버스로 이동합니다.
   c. 규칙 유형(Rule type)에서 이벤트 패턴이 있는 규칙(Rule with an event pattern)을 생성합니다.
   d. 다음을 선택합니다.
4. 이벤트 패턴 빌드(Build event pattern)에서 다음을 수행하십시오:
   a. 이벤트 소스(Event source)에서 AWS 이벤트 또는 EventBridge 파트너 이벤트(events or EventBridge partner events)를 선택합니다.
   b. 이벤트 패턴(Event pattern)의 경우 이 예에서는 `EC2 Fleet Instance Change` 이벤트 및 `launched` 하위 유형과 일치하도록 다음 이벤트 패턴을 지정합니다.

      ```
      {
       "source": ["aws.ec2fleet"],
       "detail-type": ["EC2 Fleet Instance Change"],
       "detail": {
         "sub-type": ["launched"]
      }
      ```

      이벤트 패턴을 추가하려면 다음과 같이 이벤트 패턴 양식(Event pattern form)을 선택하여 템플릿을 사용하거나 사용자 정의 패턴(JSON 편집기)(Custom pattern (JSON editor))을 선택하여 고유한 패턴을 지정할 수 있습니다.

      i. 템플릿을 사용하여 이벤트 패턴을 생성하려면 다음을 수행하세요.

         A. 이벤트 패턴 양식(Event pattern form)을 선택합니다.

B. 이벤트 소스(Event source)에서 AWS 서비스(services)를 선택합니다.

C. AWS 서비스(Service)에서 EC2 플릿(EC2 Fleet)을 선택합니다.

D. 이벤트 유형(Event type)에서 EC2 플릿 인스턴스 변경(EC2 Fleet Instance Change)을 선택합니다.

E. 패턴 편집(Edit pattern)을 선택하고 예제 이벤트 패턴과 일치하도록 `"detail": {"sub-type": ["launched"]`를 추가합니다. 적절한 JSON 형식을 위해 앞의 대괄호(]) 뒤에 쉼표(,)를 삽입합니다.

ii. (대안) 사용자 정의 이벤트 패턴을 지정하려면 다음을 수행하세요.

A. 사용자 정의 패턴(JSON 편집기)을 선택합니다.

B. 이벤트 패턴(Event pattern) 상자에서 이 예시의 이벤트 패턴을 추가합니다.

c. 다음을 선택합니다.

5. 대상 선택(Select target(s))에서 다음을 수행합니다.

a. 대상 유형(Target types)에서 AWS 서비스(service)를 선택합니다.

b. 대상 선택(Select a target)에서 SNS 주제(SNS topic)를 선택하여 이벤트가 발생할 때 이메일, 문자 메시지 또는 모바일 푸시 알림을 보내도록 합니다.

c. 주제(Target)에서 Lambda 함수(Lambda function)를 선택하고 함수(Function)에서 이벤트 발생 시 응답을 위해 생성한 함수를 선택합니다.

d. (옵션) 추가 설정(Additional settings)에서 선택적으로 추가 설정을 구성할 수 있습니다. 자세한 설명은 Amazon EventBridge 사용자 가이드의 Creating Amazon EventBridge rules that react to events(이벤트에 응답하는 Amazon EventBridge 규칙 생성)(16단계)를 참조하세요.

e. 다음을 선택합니다.

6. (선택 사항) 태그(Tags)에서 선택적으로 하나 이상의 태그를 규칙에 할당하고 다음(Next)을 선택할 수 있습니다.

7. 검토 및 생성(Review and create)에서 다음을 수행합니다.

a. 규칙의 세부 정보를 검토하고 필요에 따라 수정합니다.

b. 규칙 생성을 선택합니다.

Lambda 함수를 생성하고, Lambda 함수를 실행하는 EventBridge 규칙을 생성하는 방법에 대한 자습서는 AWS Lambda 개발자 안내서에서 자습서: EventBridge를 사용하여 Amazon EC2 인스턴스의 상태 로깅을 참조하세요.

EC2 집합 이벤트 유형

EC2 집합 이벤트 유형은 5가지입니다. 각 이벤트 유형에는 몇 가지 하위 유형이 있습니다.

이벤트 유형

- EC2 플릿 상태 변경
- EC2 플릿 스팟 인스턴스 요청 변경
- EC2 플릿 인스턴스 변경
- EC2 플릿 정보
- EC2 플릿 오류

EC2 플릿 상태 변경

EC2 집합 상태가 변경될 때 EC2 집합은 `EC2 Fleet State Change` 이벤트를 Amazon EventBridge로 전송합니다.

다음은 이 이벤트의 예제 데이터입니다.

```
{
    "version": "0",
    "id": "715ed6b3-b8fc-27fe-fad6-528c7b8bf8a2",
    "detail-type": "EC2 Fleet State Change",
    "source": "aws.ec2fleet",
    "account": "123456789012",
    "time": "2020-11-09T09:00:20Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:fleet/fleet-598fb973-87b7-422d-
be4d-6b0809bfff0a"
    ],
    "detail": {
        "sub-type": "active"
    }
}
```

`sub-type`에 대해 가능한 값은 다음과 같습니다.

`active`

EC2 집합 요청이 확인되었으며 Amazon EC2가 실행 중인 인스턴스를 목표 개수만큼 유지하려고 시도하고 있습니다.

`deleted`

EC2 집합 요청이 삭제되었고 실행 중인 인스턴스가 없습니다. EC2 집합은 인스턴스가 종료되고 2일 후에 삭제됩니다.

`deleted_running`

EC2 집합 요청이 삭제되었고 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 삭제되었고 해당 인스턴스를 종료하는 중입니다. 그 요청은 모든 인스턴스가 종료될 때까지 계속 이 상태로 유지됩니다.

`expired`

EC2 집합 요청이 만료되었습니다. 요청이 `TerminateInstancesWithExpiration` 세트로 생성된 경우 후속 `terminated` 이벤트는 인스턴스가 종료되었음을 나타냅니다.

`modify_in_progress`

EC2 집합 요청을 수정하고 있습니다. 그 요청은 수정이 완전히 처리될 때까지 계속 이 상태로 유지됩니다.

`modify_succeeded`

EC2 집합 요청이 수정되었습니다.

`submitted`

EC2 집합 요청을 평가 중이며 Amazon EC2에서 목표 개수의 인스턴스를 시작하기 위해 준비 중입니다.

`progress`

EC2 집합 요청이 이행되는 중입니다.

EC2 플릿 스팟 인스턴스 요청 변경

플릿의 스팟 인스턴스 요청 상태가 변경될 때 EC2 집합은 `EC2 Fleet Spot Instance Request Change` 이벤트를 Amazon EventBridge로 보냅니다.

다음은 이 이벤트의 예제 데이터입니다.

```
{
    "version": "0",
    "id": "19331f74-bf4b-a3dd-0f1b-ddb1422032b9",
    "detail-type": "EC2 Fleet Spot Instance Request Change",
    "source": "aws.ec2fleet",
    "account": "123456789012",
    "time": "2020-11-09T09:00:05Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:fleet/
fleet-83fd4e48-552a-40ef-9532-82a3acca5f10"
    ],
    "detail": {
        "spot-instance-request-id": "sir-rmqske6h",
        "description": "SpotInstanceRequestId sir-rmqske6h, PreviousState:
 cancelled_running",
        "sub-type": "cancelled"
    }
}
```

`sub-type`에 대해 가능한 값은 다음과 같습니다.

`active`

스팟 인스턴스 요청이 이행되었으며 요청에 연결된 스팟 인스턴스가 있습니다.

`cancelled`

스팟 인스턴스 요청을 취소했거나 스팟 인스턴스 요청이 만료되었습니다.

`disabled`

스팟 인스턴스를 중지했습니다.

`submitted`

스팟 인스턴스 요청을 제출했습니다.

EC2 플릿 인스턴스 변경

플릿의 인스턴스가 상태가 변경될 때 EC2 집합은 `EC2 Fleet Instance Change` 이벤트를 Amazon EventBridge로 보냅니다.

다음은 이 이벤트의 예제 데이터입니다.

```
{
    "version": "0",
    "id": "542ce428-c8f1-0608-c015-e8ed6522c5bc",
    "detail-type": "EC2 Fleet Instance Change",
    "source": "aws.ec2fleet",
    "account": "123456789012",
    "time": "2020-11-09T09:00:23Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:fleet/fleet-598fb973-87b7-422d-
be4d-6b0809bfff0a"
    ],
    "detail": {
        "instance-id": "i-0c594155dd5ff1829",
        "description": "{\"instanceType\":\"c5.large\",\"image\":\"ami-6057e21a\",
\"productDescription\":\"Linux/UNIX\",\"availabilityZone\":\"us-east-1d\"}",
        "sub-type": "launched"
    }
}
```

`sub-type`에 대해 가능한 값은 다음과 같습니다.

`launched`

새 인스턴스가 시작되었습니다.

`terminated`

인스턴스가 종료되었습니다.

`termination_notified`

스케일 다운 중에 Amazon EC2에 의해 스팟 인스턴스가 종료될 때 플릿의 목표 용량이 감소(예: 목표 용량 4에서 목표 용량 3으로)되었을 때 인스턴스 종료 알림이 전송되었습니다.

EC2 플릿 정보

이행 중에 오류가 발생할 때 EC2 집합은 `EC2 Fleet Information` 이벤트를 Amazon EventBridge로 보냅니다. 정보 이벤트는 플릿에서 목표 용량을 이행하려는 시도를 차단하지 않습니다.

다음은 이 이벤트의 예제 데이터입니다.

```
{
    "version": "0",
    "id": "76529817-d605-4571-7224-d36cc1b2c0c4",
    "detail-type": "EC2 Fleet Information",
    "source": "aws.ec2fleet",
    "account": "123456789012",
    "time": "2020-11-09T08:17:07Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:fleet/fleet-8becf5fe-
bb9e-415d-8f54-3fa5a8628b91"
    ],
    "detail": {
        "description": "c4.xlarge, ami-0947d2ba12ee1ff75, Linux/UNIX, us-east-1a,
 Spot price in either SpotFleetRequestConfigData or SpotFleetLaunchSpecification or
 LaunchTemplate or LaunchTemplateOverrides is less than Spot market price $0.0619",
        "sub-type": "launchSpecUnusable"
    }
}
```

`sub-type`에 대해 가능한 값은 다음과 같습니다.

`fleetProgressHalted`

모든 시작 사양의 요금이 스팟 요금보다 낮기 때문에 모든 시작 사양의 요금이 유효하지 않습니다(모든 시작 사양이 `launchSpecUnusable` 이벤트 생성). 스팟 요금이 변경되면 시작 사양이 유효해질 수 있습니다.

`launchSpecTemporarilyBlacklisted`

구성이 유효하지 않으며 인스턴스 시작하려는 시도가 여러 번 실패했습니다. 자세한 내용은 이벤트 설명을 참조하세요.

`launchSpecUnusable`

시작 사양의 요금이 스팟 요금보다 낮기 때문에 시작 사양의 요금이 유효하지 않습니다.

`registerWithLoadBalancersFailed`

로드 밸런서에 인스턴스를 등록하지 못했습니다. 자세한 내용은 이벤트 설명을 참조하세요.

EC2 플릿 오류

이행 중에 오류가 발생할 때 EC2 집합은 `EC2 Fleet Error` 이벤트를 Amazon EventBridge로 보냅니다. 오류 이벤트는 플릿에서 목표 용량을 이행하려는 시도를 차단합니다.

다음은 이 이벤트의 예제 데이터입니다.

```
{
    "version": "0",
    "id": "69849a22-6d0f-d4ce-602b-b47c1c98240e",
    "detail-type": "EC2 Fleet Error",
    "source": "aws.ec2fleet",
    "account": "123456789012",
    "time": "2020-10-07T01:44:24Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:fleet/fleet-9bb19bc6-60d3-4fd2-ae47-
d33e68eafa08"
    ],
    "detail": {
        "description": "m3.large, ami-00068cd7555f543d5, Linux/UNIX: IPv6 is not
 supported for the instance type 'm3.large'. ",
        "sub-type": "spotFleetRequestConfigurationInvalid"
    }
}
```

`sub-type`에 대해 가능한 값은 다음과 같습니다.

`iamFleetRoleInvalid`

EC2 플릿에 인스턴스를 시작하거나 종료하는 데 필요한 권한이 없습니다.

`allLaunchSpecsTemporarilyBlacklisted`

유효한 구성이 없으며 인스턴스를 시작하려는 시도가 여러 번 실패했습니다. 자세한 내용은 이벤트 설명을 참조하세요.

`spotInstanceCountLimitExceeded`

시작할 수 있는 스팟 인스턴스 수가 한도에 도달했습니다.

`spotFleetRequestConfigurationInvalid`

구성이 유효하지 않습니다. 자세한 내용은 이벤트 설명을 참조하세요.

스팟 플릿 이벤트 유형

스팟 플릿 이벤트 유형은 5가지입니다. 각 이벤트 유형에는 몇 가지 하위 유형이 있습니다.

이벤트 유형

- EC2 스팟 플릿 상태 변경
- EC2 스팟 플릿 스팟 인스턴스 요청 변경
- EC2 스팟 플릿 인스턴스 변경
- EC2 스팟 플릿 정보
- EC2 스팟 플릿 오류

EC2 스팟 플릿 상태 변경

스팟 플릿의 상태가 변경될 때 스팟 플릿은 `EC2 Spot Fleet State Change` 이벤트를 Amazon EventBridge로 보냅니다.

다음은 이 이벤트의 예제 데이터입니다.

```
{
    "version": "0",
    "id": "d1af1091-6cc3-2e24-203a-3b870e455d5b",
    "detail-type": "EC2 Spot Fleet State Change",
    "source": "aws.ec2spotfleet",
    "account": "123456789012",
    "time": "2020-11-09T08:57:06Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:spot-fleet-request/sfr-4b6d274d-0cea-4b2c-
b3be-9dc627ad1f55"
    ],
    "detail": {
        "sub-type": "submitted"
    }
}
```

sub-type에 대해 가능한 값은 다음과 같습니다.

active

스팟 플릿 요청이 확인되었으며 Amazon EC2가 실행 중인 인스턴스의 목표 개수를 유지하려고 시도하고 있습니다.

cancelled

스팟 플릿 요청이 취소되었고 실행 중인 인스턴스가 없습니다. 스팟 플릿은 인스턴스가 종료되고 2일 후에 삭제됩니다.

cancelled_running

스팟 플릿 요청이 취소되었고 추가 인스턴스를 시작하지 않습니다. 중단되거나 종료될 때까지 기존 인스턴스가 계속 실행됩니다. 그 요청은 모든 인스턴스가 중단 또는 종료될 때까지 계속 이 상태로 유지됩니다.

cancelled_terminating

스팟 플릿 요청이 취소되었고 해당 인스턴스를 종료하는 중입니다. 그 요청은 모든 인스턴스가 종료될 때까지 계속 이 상태로 유지됩니다.

expired

스팟 플릿 요청이 만료되었습니다. 요청이 TerminateInstancesWithExpiration 세트로 생성된 경우 후속 terminated 이벤트는 인스턴스가 종료되었음을 나타냅니다.

modify_in_progress

스팟 집합 요청이 수정되고 있습니다. 그 요청은 수정이 완전히 처리될 때까지 계속 이 상태로 유지됩니다.

modify_succeeded

스팟 플릿 요청이 수정되었습니다.

submitted

스팟 플릿 요청을 평가 중이며 Amazon EC2에서 목표 개수의 인스턴스를 시작하기 위해 준비 중입니다.

progress

스팟 플릿 요청이 이행되는 중입니다.

EC2 스팟 플릿 스팟 인스턴스 요청 변경

스팟 플릿은 플릿의 스팟 인스턴스 요청 상태가 변경될 때 `EC2 Spot Fleet Spot Instance Request Change` 이벤트를 Amazon EventBridge로 보냅니다.

다음은 이 이벤트의 예제 데이터입니다.

```
{
    "version": "0",
    "id": "cd141ef0-14af-d670-a71d-fe46e9971bd2",
    "detail-type": "EC2 Spot Fleet Spot Instance Request Change",
    "source": "aws.ec2spotfleet",
    "account": "123456789012",
    "time": "2020-11-09T08:53:21Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:spot-fleet-request/sfr-
a98d2133-941a-47dc-8b03-0f94c6852ad1"
    ],
    "detail": {
        "spot-instance-request-id": "sir-a2w9gc5h",
        "description": "SpotInstanceRequestId sir-a2w9gc5h, PreviousState:
 cancelled_running",
        "sub-type": "cancelled"
    }
}
```

`sub-type`에 대해 가능한 값은 다음과 같습니다.

`active`

스팟 인스턴스 요청이 이행되었으며 요청에 연결된 스팟 인스턴스가 있습니다.

`cancelled`

스팟 인스턴스 요청을 취소했거나 스팟 인스턴스 요청이 만료되었습니다.

`disabled`

스팟 인스턴스를 중지했습니다.

`submitted`

스팟 인스턴스 요청을 제출했습니다.

EC2 스팟 플릿 인스턴스 변경

플릿의 인스턴스가 상태가 변경될 때 스팟 플릿은 `EC2 Spot Fleet Instance Change` 이벤트를 Amazon EventBridge로 보냅니다.

다음은 이 이벤트의 예제 데이터입니다.

```
{
    "version": "0",
    "id": "11591686-5bd7-bbaa-eb40-d46529c2710f",
    "detail-type": "EC2 Spot Fleet Instance Change",
    "source": "aws.ec2spotfleet",
    "account": "123456789012",
    "time": "2020-11-09T07:25:02Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:spot-fleet-request/sfr-c8a764a4-bedc-4b62-
af9c-0095e6e3ba61"
    ],
    "detail": {
        "instance-id": "i-08b90df1e09c30c9b",
        "description": "{\"instanceType\":\"r4.2xlarge\",\"image\":
\"ami-032930428bf1abbff\",\"productDescription\":\"Linux/UNIX\",\"availabilityZone\":
\"us-east-1a\"}",
        "sub-type": "launched"
    }
}
```

`sub-type`에 대해 가능한 값은 다음과 같습니다.

`launched`

새 인스턴스가 시작되었습니다.

`terminated`

인스턴스가 종료되었습니다.

`termination_notified`

스케일 다운 중에 Amazon EC2에 의해 스팟 인스턴스가 종료될 때 플릿의 목표 용량이 감소(예: 목표 용량 4에서 목표 용량 3으로)되었을 때 인스턴스 종료 알림이 전송되었습니다.

EC2 스팟 플릿 정보

이행 중에 오류가 발생할 때 스팟 플릿은 `EC2 Spot Fleet Information` 이벤트를 Amazon EventBridge로 보냅니다. 정보 이벤트는 플릿에서 목표 용량을 이행하려는 시도를 차단하지 않습니다.

다음은 이 이벤트의 예제 데이터입니다.

```
{
    "version": "0",
    "id": "73a60f70-3409-a66c-635c-7f66c5f5b669",
    "detail-type": "EC2 Spot Fleet Information",
    "source": "aws.ec2spotfleet",
    "account": "123456789012",
    "time": "2020-11-08T20:56:12Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:spot-fleet-request/sfr-2531ea06-
af18-4647-8757-7d69c94971b1"
    ],
    "detail": {
        "description": "r3.8xlarge, ami-032930428bf1abbff, Linux/UNIX, us-east-1a, Spot
 bid price is less than Spot market price $0.5291",
        "sub-type": "launchSpecUnusable"
    }
}
```

`sub-type`에 대해 가능한 값은 다음과 같습니다.

`fleetProgressHalted`

모든 시작 사양의 요금이 스팟 요금보다 낮기 때문에 모든 시작 사양의 요금이 유효하지 않습니다(모든 시작 사양이 `launchSpecUnusable` 이벤트 생성). 스팟 요금이 변경되면 시작 사양이 유효해질 수 있습니다.

`launchSpecTemporarilyBlacklisted`

구성이 유효하지 않으며 인스턴스 시작하려는 시도가 여러 번 실패했습니다. 자세한 내용은 이벤트 설명을 참조하세요.

`launchSpecUnusable`

시작 사양의 요금이 스팟 요금보다 낮기 때문에 시작 사양의 요금이 유효하지 않습니다.

`registerWithLoadBalancersFailed`

로드 밸런서에 인스턴스를 등록하지 못했습니다. 자세한 내용은 이벤트 설명을 참조하세요.

EC2 스팟 플릿 오류

이행 중에 오류가 발생할 때 스팟 플릿은 `EC2 Spot Fleet Error` 이벤트를 Amazon EventBridge로 보냅니다. 오류 이벤트는 플릿에서 목표 용량을 이행하려는 시도를 차단합니다.

다음은 이 이벤트의 예제 데이터입니다.

```
{
    "version": "0",
    "id": "10adc4e7-675c-643e-125c-5bfa1b1ba5d2",
    "detail-type": "EC2 Spot Fleet Error",
    "source": "aws.ec2spotfleet",
    "account": "123456789012",
    "time": "2020-11-09T06:56:07Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:spot-fleet-request/
sfr-38725d30-25f1-4f30-83ce-2907c56dba17"
    ],
    "detail": {
        "description": "r4.2xlarge, ami-032930428bf1abbff, Linux/UNIX: The
 associatePublicIPAddress parameter can only be specified for the network interface
 with DeviceIndex 0. ",
        "sub-type": "spotFleetRequestConfigurationInvalid"
    }
}
```

`sub-type`에 대해 가능한 값은 다음과 같습니다.

`iamFleetRoleInvalid`

스팟 플릿에 인스턴스를 시작하거나 종료하는 데 필요한 권한이 없습니다.

`allLaunchSpecsTemporarilyBlacklisted`

유효한 구성이 없으며 인스턴스를 시작하려는 시도가 여러 번 실패했습니다. 자세한 내용은 이벤트 설명을 참조하세요.

`spotInstanceCountLimitExceeded`

시작할 수 있는 스팟 인스턴스 수가 한도에 도달했습니다.

`spotFleetRequestConfigurationInvalid`

구성이 유효하지 않습니다. 자세한 내용은 이벤트 설명을 참조하세요.

EC2 플릿에 대한 튜토리얼

EC2 플릿을 구성하는 방법에는 여러 가지가 있습니다. 선택하는 구성은 구체적인 사용 사례에 따라 다릅니다.

다음 튜토리얼에서는 몇 가지 가능한 사용 사례를 다루고 이를 구현하는 데 필요한 태스크를 제공합니다.

사용 사례	튜토리얼 링크
인스턴스 가중치를 사용하여 EC2 플릿의 가용성과 성능을 관리할 수 있습니다. 인스턴스 가중치를 사용하면 EC2 플릿의 각 인스턴스 유형에 가중치를 할당하여 서로 상대적인 컴퓨팅 용량과 성능을 반영합니다. 가중치를 기반으로 플릿은 원하는 목표 용량을 충족할 수 있는 한 지정된 인스턴스 유형을 원하는 대로 조합하여 사용할 수 있습니다.	튜토리얼: 인스턴스 가중치를 사용하도록 EC2 플릿 구성
온디맨드 용량을 사용하여 피크 기간 동안 가용성을 보장하면서 저렴한 비용으로 스팟 용량을 추가로 활용할 수 있습니다. 온디맨드 인스턴스를 기본 용량으로 사용하도록 EC2 플릿을 구성하여 피크 기간 동안 가용 용량을 확보합니다. 또한 스팟 인스턴스에 일부 용량을 할당하여 할인된 가격 혜택을 받을 수 있습니다. 단, Amazon EC2에 용량이 다시 필요할 경우 스팟 인스턴스가 중단될 수 있다는 점에 유의하세요.	튜토리얼: 온디맨드 인스턴스를 기본 용량으로 사용하도록 EC2 플릿 구성

사용 사례	튜토리얼 링크
용량 예약을 사용하여 온디맨드 인스턴스용 컴퓨팅 용량을 예약할 수 있습니다. 먼저 온디맨드 인스턴스를 시작할 때 `targeted` 용량 예약을 사용하도록 EC2 플릿을 구성합니다. 용량 요구 사항이 엄격하고 특정 수준의 장기 또는 단기 용량 보증이 요구되는 비즈니스 크리티컬 워크로드를 실행하는 경우, 필요할 때 언제든지 필요한 기간 동안 Amazon EC2 용량에 액세스할 수 있도록 용량 예약을 생성하는 것이 좋습니다.	튜토리얼: 대상으로 지정된 용량 예약을 사용하여 온디맨드 인스턴스를 시작하도록 EC2 플릿 구성
용량 블록을 사용하여 인기가 높은 GPU 인스턴스를 ML 워크로드에 맞게 예약할 수 있습니다. 인스턴스를 용량 블록으로 시작하도록 EC2 플릿을 구성합니다.	튜토리얼: 인스턴스를 용량 블록으로 시작하도록 EC2 플릿을 구성

튜토리얼: 인스턴스 가중치를 사용하도록 EC2 플릿 구성

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 리소

스를 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(RAM 60GB, vCPU 8개)을 고려하여 기본 용량의 20배가 요구에 적합하겠다고 결정하고 그래서 EC2 플릿 요청의 대상 용량을 20개으로 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(244.0GB, 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 USD	1	0.7 USD
r3.4xLarge	1.4 USD	2	0.7 USD
r3.8xLarge	2.8 USD	4	0.7 USD

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 확인

Example Corp은 EC2 집합을 생성하기 전에 필요한 권한을 가진 IAM 역할이 있는지 확인합니다. 자세한 내용은 EC2 집합 사전 조건 섹션을 참조하세요.

시작 템플릿 생성

그런 다음 Example Corp에서 시작 템플릿을 생성합니다. 시작 템플릿 ID는 다음 단계에서 사용됩니다. 자세한 내용은 Amazon EC2 시작 템플릿 생성 섹션을 참조하세요.

EC2 집합 생성

Example Corp은 EC2 집합에 대해 다음 구성으로 `config.json` 파일을 생성합니다. 다음 예제에서는 리소스 식별자를 사용자 고유의 리소스 식별자로 바꿉니다.

```
{
    "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 섹션을 참조하세요.

이행

할당 전략에서는 스팟 용량 풀이 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.

튜토리얼: 온디맨드 인스턴스를 기본 용량으로 사용하도록 EC2 플릿 구성

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 달성하는 것을 목표로 합니다.

계획

ABC Online은 피크 시간을 처리하기 위해 고정된 용량이 필요하지만, 더 낮은 비용으로 추가 용량을 이용하고자 합니다. 이 회사는 EC2 플릿에 대해 다음 요구 사항을 결정합니다.

- 온디맨드 인스턴스 용량 - ABC Online은 온디맨드 인스턴스 15개가 있어야 피크 시간의 트래픽을 수용할 수 있습니다.
- 스팟 인스턴스 용량 - 성능을 높이면서도 가격을 낮추기 위해 ABC Online은 5개의 스팟 인스턴스를 프로비저닝할 계획입니다.

권한 확인

ABC Online은 EC2 집합을 생성하기 전에 필요한 권한을 가진 IAM 역할이 있는지 확인합니다. 자세한 내용은 EC2 집합 사전 조건 섹션을 참조하세요.

시작 템플릿 생성

그런 다음 ABC Online에서 시작 템플릿을 생성합니다. 시작 템플릿 ID는 다음 단계에서 사용됩니다. 자세한 내용은 Amazon EC2 시작 템플릿 생성 섹션을 참조하세요.

EC2 집합 생성

ABC Online은 EC2 집합에 대해 다음 구성으로 `config.json` 파일을 생성합니다. 다음 예제에서는 리소스 식별자를 사용자 고유의 리소스 식별자로 바꿉니다.

```
{
    "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 섹션을 참조하세요.

이행

할당 전략에 따라 온디맨드 용량은 항상 충족되고 남아 있는 대상 용량이 사용 가능한 용량이 있는 경우 스팟으로 충족됩니다.

튜토리얼: 대상으로 지정된 용량 예약을 사용하여 온디맨드 인스턴스를 시작하도록 EC2 플릿 구성

이 자습서에서는 EC2 플릿이 `targeted` 용량 예약으로 온디맨드 인스턴스를 시작하기 위해 수행해야 하는 모든 단계를 안내합니다.

먼저 온디맨드 인스턴스를 시작할 때 `targeted` 온디맨드 용량 예약을 사용하도록 플릿을 구성하는 방법에 대해 학습합니다. 또한 총 온디맨드 목표 용량이 사용 가능한 미사용 용량 예약 수를 초과할 경우 플릿이 지정된 할당 전략을 사용하여 나머지 목표 용량을 시작할 인스턴스 풀을 선택하도록 플릿을 구성하는 방법도 학습합니다.

EC2 플릿 구성

이 튜토리얼에서 플릿은 다음과 같이 구성됩니다.

- 목표 용량: 온디맨드 인스턴스 10개
- 총 미사용 `targeted` 용량 예약: 6개(플릿의 온디맨드 목표 용량인 온디맨드 인스턴스 10개보다 적음)

- 용량 예약 풀 수: 2개(us-east-1a 및 us-east-1b)
- 풀당 용량 예약 수: 3개
- 온디맨드 할당 전략: lowest-price(미사용 용량 예약 수가 온디맨드 목표 용량보다 적을 경우 플릿에서 온디맨드 할당 전략을 기반으로 나머지 온디맨드 용량을 시작할 풀 결정)

  lowest-price 할당 전략 대신 prioritized 할당 전략을 사용할 수도 있습니다.

온디맨드 인스턴스를 targeted 용량 예약으로 시작하려면 다음과 같이 여러 단계를 수행해야 합니다.

- 1단계: 용량 예약 생성
- 2단계: 용량 예약 리소스 그룹 생성
- 3단계: 용량 예약을 용량 예약 리소스 그룹에 추가
- (선택 사항) 4단계: 리소스 그룹에서 용량 예약 확인
- 5단계: 용량 예약이 특정 리소스 그룹을 대상으로 하도록 지정하는 시작 템플리트 생성
- (선택 사항) 6단계: 시작 템플릿 설명
- 7단계: EC2 플릿 생성
- (선택사항) 8단계: 나머지 미사용 용량 예약 수 확인

1단계: 용량 예약 생성

create-capacity-reservation 명령을 사용하여 us-east-1a에 대해 용량 예약 3개, us-east-1b에 대해 또 다른 용량 예약 3개를 생성합니다. 가용 영역을 제외하고 용량 예약의 다른 속성은 동일합니다.

us-east-1a의 용량 예약 3개

```
aws ec2 create-capacity-reservation \
    --availability-zone us-east-1a \
    --instance-type c5.xlarge \
    --instance-platform Linux/UNIX \
    --instance-count 3 \
    --instance-match-criteria targeted
```

결과 용량 예약 ID의 예

```
cr-1234567890abcdef1
```

us-east-1b의 용량 예약 3개

```
aws ec2 create-capacity-reservation \
    --availability-zone us-east-1b \
    --instance-type c5.xlarge \
    --instance-platform Linux/UNIX \
    --instance-count 3 \
    --instance-match-criteria targeted
```

결과 용량 예약 ID의 예

```
cr-54321abcdef567890
```

2단계: 용량 예약 리소스 그룹 생성

`resource-groups` 서비스 및 create-group 명령을 사용하여 용량 예약 리소스 그룹을 생성합니다. 이 예제에서 리소스 그룹의 이름은 `my-cr-group`입니다. 리소스 그룹을 생성해야 하는 이유에 대한 자세한 내용은 용량 예약을 사용하여 EC2 플릿의 온디맨드 용량 예약 섹션을 참조하세요.

```
aws resource-groups create-group \
    --name my-cr-group \
    --configuration '{"Type":"AWS::EC2::CapacityReservationPool"}'
 '{"Type":"AWS::ResourceGroups::Generic", "Parameters": [{"Name": "allowed-resource-
types", "Values": ["AWS::EC2::CapacityReservation"]}]}'
```

3단계: 용량 예약을 용량 예약 리소스 그룹에 추가

`resource-groups` 서비스 및 group-resources 명령을 사용하여 1단계에서 생성한 용량 예약을 용량 예약 리소스 그룹에 추가합니다. 해당 ARN을 기준으로 온디맨드 용량 예약을 참조해야 합니다.

```
aws resource-groups group-resources \
    --group my-cr-group \
    --resource-arns \
      arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-1234567890abcdef1 \
      arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-54321abcdef567890
```

출력 예시

```
{
   "Failed": [],
   "Succeeded": [
```

```
    "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-1234567890abcdef1",
    "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-54321abcdef567890"
    ]
}
```

(선택 사항) 4단계: 리소스 그룹에서 용량 예약 확인

`resource-groups` 서비스 및 list-group-resources 명령을 사용하여 선택적으로 리소스 그룹을 설명하여 해당 용량 예약을 확인합니다.

```
aws resource-groups list-group-resources --group my-cr-group
```

출력 예시

```
{
    "ResourceIdentifiers": [
        {
            "ResourceType": "AWS::EC2::CapacityReservation",
            "ResourceArn": "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/
cr-1234567890abcdef1"
        },
        {
            "ResourceType": "AWS::EC2::CapacityReservation",
            "ResourceArn": "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/
cr-54321abcdef567890"
        }
    ]
}
```

5단계: 용량 예약이 특정 리소스 그룹을 대상으로 하도록 지정하는 시작 템플리트 생성

Create-launch-template 명령을 사용하여, 사용할 용량 예약을 지정할 시작 템플릿을 생성합니다. 이 예에서 플릿은 리소스 그룹에 추가된 `targeted` 용량 예약을 사용합니다. 따라서 시작 템플릿 데이터는 용량 예약이 특정 리소스 그룹을 대상으로 하도록 지정합니다. 이 예제에서 시작 템플릿의 이름은 `my-launch-template`입니다.

```
aws ec2 create-launch-template \
    --launch-template-name my-launch-template \
    --launch-template-data \
        '{"ImageId": "ami-0123456789example",
          "CapacityReservationSpecification":
```

```
            {"CapacityReservationTarget":
                { "CapacityReservationResourceGroupArn": "arn:aws:resource-groups:us-
east-1:123456789012:group/my-cr-group" }
            }
        }'
```

(선택 사항) 6단계: 시작 템플릿 설명

describe-launch-template-versions 명령을 사용하여 구성을 확인할 시작 템플릿을 선택적으로 설명합니다.

```
aws ec2 describe-launch-template-versions --launch-template-name my-launch-template
```

출력 예시

```
{
    "LaunchTemplateVersions": [
        {
            "LaunchTemplateId": "lt-01234567890example",
            "LaunchTemplateName": "my-launch-template",
            "VersionNumber": 1,
            "CreateTime": "2021-01-19T20:50:19.000Z",
            "CreatedBy": "arn:aws:iam::123456789012:user/Admin",
            "DefaultVersion": true,
            "LaunchTemplateData": {
                "ImageId": "ami-0947d2ba12ee1ff75",
                "CapacityReservationSpecification": {
                    "CapacityReservationTarget": {
                        "CapacityReservationResourceGroupArn": "arn:aws:resource-
groups:us-east-1:123456789012:group/my-cr-group"
                    }
                }
            }
        }
    ]
}
```

7단계: EC2 플릿 생성

시작할 인스턴스에 대한 구성 정보를 지정하는 EC2 플릿을 생성합니다. 다음 EC2 플릿 구성에서는 이 예제와 관련된 구성만 보여 줍니다. 시작 템플릿 `my-launch-template`은 5단계에서 생성한 시

작 템플릿입니다. 두 개의 인스턴스 풀이 있으며, 각각 인스턴스 유형(c5.xlarge)은 동일하지만 가용 영역(us-east-1a 및 us-east-1b)은 다릅니다. 인스턴스 풀의 가격은 가용 영역이 아닌 리전에 대해 정의되기 때문에 동일합니다. 총 목표 용량은 10이고 기본 목표 용량 유형은 on-demand입니다. 온디맨드 할당 전략은 lowest-price입니다. 용량 예약에 대한 사용 전략은 use-capacity-reservations-first입니다.

> **Note**
>
> 플릿 유형은 instant여야 합니다. 다른 플릿 유형에서는 use-capacity-reservations-first를 지원하지 않습니다.

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1"
            },
            "Overrides": [
               {
                   "InstanceType": "c5.xlarge",
                   "AvailabilityZone": "us-east-1a"
               },
               {
                    "InstanceType": "c5.xlarge",
                    "AvailabilityZone": "us-east-1b"
               }
            ]
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 10,
        "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 10개의 인스턴스가 시작됩니다.

- 용량 예약은 먼저 다음과 같이 6개의 온디맨드 인스턴스를 시작하는 데 사용됩니다.
  - 3개의 온디맨드 인스턴스는 `us-east-1a`에서 3개의 `c5.xlarge` `targeted` 용량 예약으로 시작됩니다.
  - 3개의 온디맨드 인스턴스는 `us-east-1b`에서 3개의 `c5.xlarge` `targeted` 용량 예약으로 시작됩니다.
- 목표 용량을 충족하기 위해 4개의 추가 온디맨드 인스턴스가 온디맨드 할당 전략에 따라 일반 온디맨드 용량으로 시작됩니다(이 예에서는 `lowest-price`). 그러나 풀의 가격이 동일하기 때문에(가용 영역이 아닌 리전별로 가격이 책정되기 때문에) 플릿은 나머지 4개의 온디맨드 인스턴스를 두 풀 중 하나로 시작합니다.

(선택사항) 8단계: 나머지 미사용 용량 예약 수 확인

플릿이 시작된 후, 선택적으로 describe-capacity-reservations를 실행하여 미사용 용량 예약이 몇 개나 남아 있는지 확인할 수 있습니다. 이 예에서는 모든 풀의 용량 예약이 모두 사용되었음을 보여 주는 다음 응답이 나타납니다.

```
{ "CapacityReservationId":  "cr-111",
     "InstanceType":  "c5.xlarge",
     "AvailableInstanceCount":  0
}

 { "CapacityReservationId":  "cr-222",
     "InstanceType":  "c5.xlarge",
     "AvailableInstanceCount":  0
}
```

튜토리얼: 인스턴스를 용량 블록으로 시작하도록 EC2 플릿을 구성

이 자습서에서는 EC2 플릿에서 용량 블록으로 인스턴스를 내보내려면 수행해야 하는 단계를 안내합니다.

대부분의 경우 EC2 플릿 요청의 대상 용량은 목표로 하는 용량 블록 예약의 가용 용량 이하여야 합니다. 용량 블록 예약의 한도를 초과하는 목표 용량 요청은 충족되지 않습니다. 대상 용량 요청이 용량 블

록 예약 한도를 초과하면 용량 블록 예약 한도를 초과하는 용량에 대한 `Insufficient Capacity Exception`이 발생합니다.

> **Note**
>
> 용량 블록의 경우 EC2 플릿이 대체되어 원하는 목표 용량의 나머지에 대한 온디맨드 인스턴스를 시작하지 않습니다.

가용 용량 블록 예약에서 요청된 목표 용량을 EC2 플릿에서 충족할 수 없으면 EC2 플릿에서는 최대한 많은 용량을 충족하고, 시작할 수 있었던 인스턴스를 반환합니다. 모든 인스턴스가 프로비저닝될 때까지 EC2 플릿 직접 호출을 다시 반복할 수 있습니다.

EC2 플릿 요청 구성 후에는 용량 블록 예약 시작 날짜까지 기다려야 합니다. 아직 시작되지 않은 용량 블록으로 내보내도록 EC2 플릿에 요청하면 `Insufficient Capacity Error`가 발생합니다.

용량 블록 예약이 활성화되면 EC2 플릿 API를 직접적으로 호출하고 선택한 파라미터에 따라 인스턴스를 용량 블록에 프로비저닝할 수 있습니다. 용량 블록에서 실행 중인 인스턴스는 인스턴스를 수동으로 중지 또는 종료할 때까지 또는 용량 블록 예약이 종료될 때 Amazon EC2에서 인스턴스를 종료할 때까지 계속 실행됩니다.

용량 블록에 대한 자세한 내용은 ML용 용량 블록 섹션을 참조하세요.

고려 사항

- 용량 블록으로 인스턴스를 시작하는 데는 `instant` 유형의 EC2 플릿 요청만 지원됩니다. 자세한 내용은 instant 유형의 EC2 플릿 구성 섹션을 참조하세요.
- 동일한 EC2 플릿 요청의 여러 용량 블록은 지원되지 않습니다.
- `capacity-block`을 `DefaultTargetCapacity`로 설정하는 동안 `OnDemandTargetCapacity` 또는 `SpotTargetCapacity`를 사용하는 것도 지원되지 않습니다.
- `DefaultTargetCapacityType`이 `capacity-block`으로 설정되었으면 `OnDemandOptions::CapacityReservationOptions`를 제공할 수 없습니다. 예외가 발생합니다.

용량 블록으로 인스턴스를 시작하도록 EC2 플릿을 구성하려면 다음을 수행하세요.

1. 시작 템플릿을 생성합니다.

   시작 템플릿에서 다음을 수행합니다.

- `InstanceMarketOptionsRequest`의 경우 `MarketType`을 `capacity-block`으로 설정합니다.
- 용량 블록 예약을 대상으로 하려면 `CapacityReservationID`에 대해 용량 블록 예약 ID를 지정합니다.

시작 템플릿 이름과 버전을 기록해 둡니다. 다음 단계에서 이 정보를 사용합니다.

시작 템플릿 생성에 대한 자세한 내용은 Amazon EC2 시작 템플릿 생성 섹션을 참조하세요.

2. EC2 플릿을 구성합니다.

   다음 EC2 플릿 구성으로 `config.json` 파일을 생성합니다. 다음 예제에서는 리소스 식별자를 사용자 고유의 리소스 식별자로 바꿉니다.

   EC2 플릿 구성에 대한 자세한 내용은 EC2 집합 생성 섹션을 참조하세요.

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "CBR-launch-template",
                "Version": "1"
            },
            "Overrides": [
                {
                    "InstanceType": "p5.48xlarge",
                    "AvailabilityZone": "us-east-1a"
                },
            ]
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 10,
        "DefaultTargetCapacityType": "capacity-block"
    },
    "Type": "instant"
}
```

3. 플릿을 시작합니다.

   다음 create-fleet 명령을 사용합니다.

```
aws ec2 create-fleet --cli-input-json file://config.json
```

자세한 내용은 EC2 집합 생성 섹션을 참조하세요.

EC2 플릿에 대한 CLI 구성 예제

다음과 같이 JSON 파일에서 EC2 플릿 구성을 정의하고, create-fleet 명령으로 해당 파일을 참조하여 플릿을 생성할 수 있습니다.

```
aws ec2 create-fleet --cli-input-json file://file_name.json
```

다음 예제는 다양한 EC2 플릿 사용 사례의 시작 구성을 보여줍니다. 구성 파라미터에 대한 자세한 내용은 create-fleet을 참조하세요.

예시

- 예 1: 스팟 인스턴스를 기본 구입 옵션으로 시작
- 예 2: 온디맨드 인스턴스를 기본 구입 옵션으로 시작
- 예 3: 온디맨드 인스턴스를 기본 용량으로 시작
- 예제 4: 여러 용량 예약을 사용하여 온디맨드 인스턴스 시작
- 예제 5: 총 목표 용량이 미사용 용량 예약 수보다 많은 경우 용량 예약을 사용하여 온디맨드 인스턴스 시작
- 예제 6: 대상으로 지정된 용량 예약을 사용하여 온디맨드 인스턴스 시작
- 예제 7: 대체 스팟 인스턴스를 시작하도록 용량 리밸런싱 구성
- 예제 8: 용량 최적화 플릿에서 스팟 인스턴스 시작
- 예제 9: 용량 최적화 플릿에서 우선순위를 사용하여 스팟 인스턴스 시작
- 예제 10: price-capacity-optimized 플릿에서 스팟 인스턴스 시작
- 예제 11: 속성 기반 인스턴스 유형 선택 구성

유형 `instant` 플릿에 대한 추가 CLI 예제는 instant 유형의 EC2 플릿 구성의 내용을 참조하세요.

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 spot이므로 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-demand이므로 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 spot입니다. 지정한 대로 플릿은 온디맨드 인스턴스 1개를 시작하지만 총 목표 용량을 충족하려면 인스턴스를 하나 더 시작해야 합니다. 차이에 대한 구매 옵션이 TotalTargetCapacity – OnDemandTargetCapacity = DefaultTargetCapacityType으로 계산되므로 플릿에서 스팟 인스턴스 1개를 시작합니다.

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e8c754449b27161c",
                "Version": "1"
            }

        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
        "OnDemandTargetCapacity": 1,
        "DefaultTargetCapacityType": "spot"
    }
}
```

예제 4: 여러 용량 예약을 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. 이 예에서는 플릿이 목표 용량을 처리하는 데 필요한 용량 예약보다 많을 때 사용할 용량 예약을 선택하는 방법을 보여 줍니다.

이 예에서 플릿 구성은 다음과 같습니다.

- 목표 용량: 온디맨드 인스턴스 12개
- 총 미사용 용량 예약: 15개(플릿의 목표 용량인 온디맨드 인스턴스 12개보다 많음)
- 용량 예약 풀 수: 3개(m5.large, m4.xlarge 및 m4.2xlarge)

- 풀당 용량 예약 수: 5개
- 온디맨드 할당 전략: `lowest-price`(여러 인스턴스 풀에 미사용 용량 예약이 여러 개 있는 경우 플릿은 온디맨드 할당 전략에 따라 온디맨드 인스턴스를 시작할 풀 결정)

  `lowest-price` 할당 전략 대신 `prioritized` 할당 전략을 사용할 수도 있습니다.

용량 예약

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"
}
```

플릿 구성

다음 플릿 구성에서는 이 예제와 관련된 구성만 보여 줍니다. 총 목표 용량은 12이고, 기본 목표 용량 유형은 on-demand입니다. 온디맨드 할당 전략은 lowest-price입니다. 용량 예약에 대한 사용 전략은 use-capacity-reservations-first입니다.

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 instant여야 합니다. 다른 플릿 유형에서는 use-capacity-reservations-first를 지원하지 않습니다.

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-abc1234567example",
                "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.large`가 최저가이며, 사용 가능한 미사용 `m5.large` 용량 예약 5개
- us-east-1a의 `m4.xlarge` 온디맨드 인스턴스 5개 – us-east-1a의 `m4.xlarge`가 최저가이며, 사용 가능한 미사용 `m4.xlarge` 용량 예약 5개
- us-east-1a의 `m4.2xlarge` 온디맨드 인스턴스 2개 – us-east-1a의 `m4.2xlarge`가 세 번째 최저가이고 사용 가능한 미사용 `m4.2xlarge` 용량 예약은 5개이며, 그중 2개만 있으면 목표 용량 충족 가능

플릿이 시작된 후, describe-capacity-reservations를 실행하여 미사용 용량 예약이 몇 개나 남아 있는지 확인할 수 있습니다. 이 예에서는 `m5.large` 및 `m4.xlarge` 용량 예약이 모두 사용되었고 `m4.2xlarge` 용량 예약 3개는 미사용 상태로 남아 있음을 보여주는 다음과 같은 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
}
```

예제 5: 총 목표 용량이 미사용 용량 예약 수보다 많은 경우 용량 예약을 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. 이 예에서는 총 목표 용량이 사용 가능한 미사용 용량 예약 수를 초과할 때 온 디맨드 인스턴스를 시작할 인스턴스 풀을 플릿이 선택하는 방법을 보여 줍니다.

이 예에서 플릿 구성은 다음과 같습니다.

- 목표 용량: 온디맨드 인스턴스 16개
- 총 미사용 용량 예약: 15개(플릿의 목표 용량인 온디맨드 인스턴스 16개보다 적음)
- 용량 예약 풀 수: 3개(m5.large, m4.xlarge 및 m4.2xlarge)
- 풀당 용량 예약 수: 5개
- 온디맨드 할당 전략: lowest-price(미사용 용량 예약 수가 온디맨드 목표 용량보다 적을 경우 플릿에서 온디맨드 할당 전략을 기반으로 나머지 온디맨드 용량을 시작할 풀 결정)

  lowest-price 할당 전략 대신 prioritized 할당 전략을 사용할 수도 있습니다.

용량 예약

이 계정은 3개의 풀에 다음과 같은 미사용 용량 예약 15개를 가지고 있습니다. 각 풀의 용량 예약 수는 AvailableInstanceCount로 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"
}
```

플릿 구성

다음 플릿 구성에서는 이 예제와 관련된 구성만 보여 줍니다. 총 목표 용량은 16이고 기본 목표 용량 유형은 `on-demand`입니다. 온디맨드 할당 전략은 `lowest-price`입니다. 용량 예약에 대한 사용 전략은 `use-capacity-reservations-first`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 유형에서는 `use-capacity-reservations-first`를 지원하지 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 최저가이며, 사용 가능한 미사용 m5.large 용량 예약 5개 용량 예약은 먼저 5개의 온디맨드 인스턴스를 시작하는 데 사용됩니다. 나머지 m4.xlarge 및 m4.2xlarge 용량 예약을 사용한 후, 목표 용량을 충족하기 위해 추가 온디맨드 인스턴스가 온디맨드 할당 전략에 따라 시작됩니다(이 예에서는 lowest-price).
- us-east-1a의 m4.xlarge 온디맨드 인스턴스 5개 – us-east-1a의 m4.xlarge가 두 번째 최저가이며, 사용 가능한 미사용 m4.xlarge 용량 예약 5개
- us-east-1a의 m4.2xlarge 온디맨드 인스턴스 5개 – us-east-1a의 m4.2xlarge가 세 번째 최저가이며, 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
}
```

예제 6: 대상으로 지정된 용량 예약을 사용하여 온디맨드 인스턴스 시작

용량 예약에 대한 사용 전략을 use-capacity-reservations-first로 설정하여 온디맨드 인스턴스를 시작할 때 targeted 온디맨드 용량 예약부터 사용하도록 플릿을 구성할 수 있습니다. 이 예에서는 온디맨드 인스턴스를 targeted 용량 예약으로 시작하는 방법을 보여 줍니다. 여기서 용량 예약의 속성은 가용 영역(us-east-1a 및 us-east-1b)을 제외하고 동일합니다. 또한 총 목표 용량이 사용

가능한 미사용 용량 예약 수를 초과할 때 온디맨드 인스턴스를 시작할 인스턴스 풀을 플릿이 선택하는 방법을 보여 줍니다.

이 예에서 플릿 구성은 다음과 같습니다.

- 목표 용량: 온디맨드 인스턴스 10개
- 총 미사용 `targeted` 용량 예약: 6개(플릿의 온디맨드 목표 용량인 온디맨드 인스턴스 10개보다 적음)
- 용량 예약 풀 수: 2개(`us-east-1a` 및 `us-east-1b`)
- 풀당 용량 예약 수: 3개
- 온디맨드 할당 전략: `lowest-price`(미사용 용량 예약 수가 온디맨드 목표 용량보다 적을 경우 플릿에서 온디맨드 할당 전략을 기반으로 나머지 온디맨드 용량을 시작할 풀 결정)

  `lowest-price` 할당 전략 대신 `prioritized` 할당 전략을 사용할 수도 있습니다.

이 예제를 완료하기 위해 수행해야 하는 절차에 대한 시연은 튜토리얼: 대상으로 지정된 용량 예약을 사용하여 온디맨드 인스턴스를 시작하도록 EC2 플릿 구성 섹션을 참조하세요.

용량 예약

이 계정은 2개의 풀에 다음과 같은 미사용 용량 예약 6개를 가지고 있습니다. 이 예에서 풀은 가용 영역에 따라 다릅니다. 각 풀의 용량 예약 수는 `AvailableInstanceCount`로 표시됩니다.

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "c5.xlarge",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 3,
    "InstanceMatchCriteria": "open",
    "State": "active"
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "c5.xlarge",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1b",
    "AvailableInstanceCount": 3,
    "InstanceMatchCriteria": "open",
```

```
    "State": "active"
}
```

플릿 구성

다음 플릿 구성에서는 이 예제와 관련된 구성만 보여 줍니다. 총 목표 용량은 10이고 기본 목표 용량 유형은 on-demand입니다. 온디맨드 할당 전략은 lowest-price입니다. 용량 예약에 대한 사용 전략은 use-capacity-reservations-first입니다.

이 예에서 us-east-1의 c5.xlarge에 대한 온디맨드 인스턴스 가격은 시간당 0.17 USD입니다.

> **Note**
>
> 플릿 유형은 instant여야 합니다. 다른 플릿 유형에서는 use-capacity-reservations-first를 지원하지 않습니다.

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1"
            },
            "Overrides": [
               {
                   "InstanceType": "c5.xlarge",
                   "AvailabilityZone": "us-east-1a"
               },
               {
                    "InstanceType": "c5.xlarge",
                    "AvailabilityZone": "us-east-1b"
               }
            ]
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 10,
        "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 10개의 인스턴스가 시작됩니다.

- 용량 예약은 먼저 다음과 같이 6개의 온디맨드 인스턴스를 시작하는 데 사용됩니다.
  - 3개의 온디맨드 인스턴스는 `us-east-1a`에서 3개의 `c5.xlarge` `targeted` 용량 예약으로 시작됩니다.
  - 3개의 온디맨드 인스턴스는 `us-east-1b`에서 3개의 `c5.xlarge` `targeted` 용량 예약으로 시작됩니다.
- 목표 용량을 충족하기 위해 4개의 추가 온디맨드 인스턴스가 온디맨드 할당 전략에 따라 일반 온디맨드 용량으로 시작됩니다(이 예에서는 `lowest-price`). 그러나 풀의 가격이 동일하기 때문에(가용 영역이 아닌 리전별로 가격이 책정되기 때문에) 플릿은 나머지 4개의 온디맨드 인스턴스를 두 풀 중 하나로 시작합니다.

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": "c5.xlarge",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "c5.xlarge",
    "AvailableInstanceCount": 0
}
```

예제 7: 대체 스팟 인스턴스를 시작하도록 용량 리밸런싱 구성

다음 예제에서는 Amazon EC2에서 플릿의 스팟 인스턴스에 대한 리밸런싱 권고가 생성될 때 대체 스팟 인스턴스를 시작하도록 EC2 플릿을 구성합니다. 스팟 인스턴스의 자동 대체를 구성하려면 `ReplacementStrategy`에 `launch-before-terminate`를 지정합니다. 새 대체 스팟 인스턴스가 시작되는 시점부터 이전 스팟 인스턴스가 자동으로 삭제될 때까지의 시간 지연을 구성하려면 `termination-delay`에 대해 초 단위로 값을 지정합니다. 자세한 내용은 구성 옵션 섹션을 참조하세요.

> **Note**
>
> 이러한 절차가 완료된 후에만 이전 인스턴스가 종료되도록 인스턴스 종료 프로시저가 완료되는 데 걸리는 시간을 예측할 수 있는 경우에만 `launch-before-terminate`를 사용하는 것이 좋습니다. 두 인스턴스가 실행되는 동안에는 두 인스턴스에 대해 요금이 청구됩니다.

용량 리밸런싱 전략의 효과는 EC2 집합 요청에 지정된 스팟 용량 풀 수에 따라 달라집니다. 다양한 인스턴스 유형 및 가용 영역 세트로 플릿을 구성하고 `AllocationStrategy`에 대해 `capacity-optimized`를 지정하는 것이 좋습니다. 용량 리밸런싱에 대해 EC2 집합을 구성할 때 고려해야 할 사항에 대한 자세한 내용은 EC2 플릿 및 스팟 플릿의 용량 리밸런싱을 사용하여 위험이 있는 스팟 인스턴스 교체 섹션을 참조하세요.

```
{
    "ExcessCapacityTerminationPolicy": "termination",
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "LaunchTemplate",
                "Version": "1"
            },
                 "Overrides": [
                       {
                           "InstanceType": "c3.large",
                           "WeightedCapacity": 1,
                            "Placement": {
                               "AvailabilityZone": "us-east-1a"
                           }
                       },
                       {
                           "InstanceType": "c4.large",
```

```
                            "WeightedCapacity": 1,
                             "Placement": {
                                "AvailabilityZone": "us-east-1a"
                            }
                        },
                        {
                            "InstanceType": "c5.large",
                            "WeightedCapacity": 1,
                             "Placement": {
                                "AvailabilityZone": "us-east-1a"
                            }
                        }
                ]
          }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 5,
        "DefaultTargetCapacityType": "spot"
    },
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized",
        "MaintenanceStrategies": {
            "CapacityRebalance": {
                "ReplacementStrategy": "launch-before-terminate",
                "TerminationDelay": "720"
            }
        }
    }
}
```

예제 8: 용량 최적화 플릿에서 스팟 인스턴스 시작

다음 예제에서는 용량을 최적화하는 스팟 할당 전략을 사용하여 EC2 플릿을 구성하는 방법을 보여줍니다. 용량을 최적화하려면 `AllocationStrategy`를 `capacity-optimized`로 설정해야 합니다.

다음 예제에서는 3개의 시작 사양으로 3개의 스팟 용량 풀을 지정합니다. 목표 용량은 스팟 인스턴스 50개입니다. EC2 플릿은 시작하는 인스턴스의 수에 대한 최적의 용량을 가진 스팟 용량 풀로 스팟 인스턴스 50개를 시작하려고 시도합니다.

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized",
```

```
        },
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1"
            },
                 "Overrides": [
                       {
                           "InstanceType": "r4.2xlarge",
                           "Placement": {
                               "AvailabilityZone": "us-west-2a"
                           },
                      },
                       {
                           "InstanceType": "m4.2xlarge",
                           "Placement": {
                               "AvailabilityZone": "us-west-2b"
                           },
                       },
                       {
                           "InstanceType": "c5.2xlarge",
                           "Placement": {
                               "AvailabilityZone": "us-west-2b"
                           }
                       }
                  ]
            }
    ],
    "TargetCapacitySpecification": {
            "TotalTargetCapacity": 50,
            "DefaultTargetCapacityType": "spot"

    }
}
```

예제 9: 용량 최적화 플릿에서 우선순위를 사용하여 스팟 인스턴스 시작

다음 예제에서는 최선의 노력을 기준으로 우선순위를 사용하는 동안 용량을 최적화하는 스팟 할당 전략을 사용하여 EC2 플릿을 구성하는 방법을 보여줍니다.

`capacity-optimized-prioritized` 할당 전략을 사용할 때 `Priority` 파라미터를 사용하여 스팟 인스턴스 풀의 우선순위를 지정할 수 있습니다. 여기서 숫자가 낮을수록 우선순위는 높습니다. 여러

스팟 용량 풀에 선호도가 동일한 경우 동일한 우선순위를 설정해도 됩니다. 우선순위가 설정되지 않은 풀은 우선순위 측면에서 마지막으로 고려됩니다.

스팟 용량 풀의 우선순위를 지정하려면 AllocationStrategy를 capacity-optimized-prioritized로 설정해야 합니다. EC2 플릿은 용량을 우선으로 최적화하지만 최선의 노력을 기준으로 우선순위를 따릅니다. 예를 들어 EC2 플릿에서 최적 용량으로 프로비저닝하는 데 우선순위가 큰 영향을 미치지 않을 수 있습니다. 이 옵션은 중단 가능성을 최소화해야 하고 특정 인스턴스 유형에 대한 선호도가 중요한 워크로드에 적합합니다.

다음 예제에서는 3개의 시작 사양으로 3개의 스팟 용량 풀을 지정합니다. 각 풀의 우선순위가 지정되며, 여기서 숫자가 낮을수록 우선순위가 높습니다. 목표 용량은 스팟 인스턴스 50개입니다. EC2 플릿은 가장 높은 우선순위의 스팟 용량 풀로 50개의 스팟 인스턴스를 시작하려고 시도하지만 먼저 용량을 최적화합니다.

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized-prioritized"
        },
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1"
            },
                 "Overrides": [
                        {
                            "InstanceType": "r4.2xlarge",
                            "Priority": 1,
                            "Placement": {
                                "AvailabilityZone": "us-west-2a"
                            },
                       },
                        {
                            "InstanceType": "m4.2xlarge",
                            "Priority": 2,
                            "Placement": {
                                "AvailabilityZone": "us-west-2b"
                            },
                        },
                        {
                            "InstanceType": "c5.2xlarge",
                            "Priority": 3,
```

```
                            "Placement": {
                                "AvailabilityZone": "us-west-2b"
                            }
                        }
                    ]
                }
    ],
    "TargetCapacitySpecification": {
            "TotalTargetCapacity": 50,
            "DefaultTargetCapacityType": "spot"
}
```

예제 10: price-capacity-optimized 플릿에서 스팟 인스턴스 시작

다음 예제에서는 용량과 가격을 모두 최적화하는 스팟 할당 전략을 사용하여 EC2 플릿을 구성하는 방법을 보여줍니다. 가격을 고려하면서 용량을 최적화하려면 스팟 `AllocationStrategy`를 `price-capacity-optimized`로 설정해야 합니다.

다음 예제에서는 3개의 시작 사양으로 3개의 스팟 용량 풀을 지정합니다. 목표 용량은 스팟 인스턴스 50개입니다. EC2 플릿은 시작하는 인스턴스의 수에 대한 최적의 용량을 가진 스팟 용량 풀로 스팟 인스턴스 50개를 시작하는 동시에 가격이 가장 낮은 풀을 선택하려고 시도합니다.

```
{
    "SpotOptions": {
        "AllocationStrategy": "price-capacity-optimized",
        "MinTargetCapacity": 2,
        "SingleInstanceType": true
    },
    "OnDemandOptions": {
        "AllocationStrategy": "lowest-price"
    },
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1"
            },
                 "Overrides": [
                        {
                            "InstanceType": "r4.2xlarge",
                            "Placement": {
                                "AvailabilityZone": "us-west-2a"
```

```
                        },
                    },
                     {
                        "InstanceType": "m4.2xlarge",
                        "Placement": {
                            "AvailabilityZone": "us-west-2b"
                        },
                     },
                     {
                        "InstanceType": "c5.2xlarge",
                        "Placement": {
                            "AvailabilityZone": "us-west-2b"
                        }
                     }
                ]
            }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 50,
        "OnDemandTargetCapacity":0,
        "SpotTargetCapacity":50,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

예제 11: 속성 기반 인스턴스 유형 선택 구성

다음 예에서는 인스턴스 유형을 식별하는 데 속성 기반 인스턴스 유형 선택 방식을 사용하도록 EC2 플릿을 구성하는 방법을 보여줍니다. 필요한 인스턴스 속성을 지정하려면 `InstanceRequirements` 구조에서 해당 속성을 지정합니다.

다음 예제에서는 2개의 인스턴스 속성이 지정됩니다.

- `VCpuCount` - vCPU가 최소 2개로 지정되었습니다. 최댓값이 지정되지 않았으므로 최대 한도는 없습니다.
- `MemoryMiB` - 메모리가 최소 4MiB로 지정되었습니다. 최댓값이 지정되지 않았으므로 최대 한도는 없습니다.

vCPU가 2개 이상이고 메모리가 4MiB 이상인 모든 인스턴스 유형이 식별됩니다. 단, EC2 플릿에서 플릿을 제공하는 경우 가격 보호 및 할당 전략에서 일부 인스턴스 유형이 제외될 수 있습니다.

지정 가능한 모든 속성의 목록과 설명은 Amazon EC2 API 참조에서 InstanceRequirements를 참조하세요.

```
{
 "SpotOptions": {
  "AllocationStrategy": "price-capacity-optimized"
 },
 "LaunchTemplateConfigs": [{
  "LaunchTemplateSpecification": {
   "LaunchTemplateName": "my-launch-template",
   "Version": "1"
  },
  "Overrides": [{
   "InstanceRequirements": {
    "VCpuCount": {
     "Min": 2
    },
    "MemoryMiB": {
     "Min": 4
    }
   }
  }]
 }],
 "TargetCapacitySpecification": {
  "TotalTargetCapacity": 20,
  "DefaultTargetCapacityType": "spot"
 },
 "Type": "instant"
}
```

예제 CLI 구성 스팟 플릿

다음과 같이 JSON 파일에서 스팟 플릿 구성을 정의하고, request-spot-fleet AWS CLI 명령으로 해당 파일을 참조하여 플릿을 생성할 수 있습니다.

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://file_name.json
```

다음 예제는 다양한 스팟 플릿 사용 사례의 시작 구성을 보여줍니다. 구성 파라미터에 대한 자세한 내용은 request-spot-fleet을 참조하세요. 스팟 플릿 생성에 대한 자세한 내용은 스팟 플릿 생성의 내용을 참조하세요.

> **Note**
>
> 스팟 플릿의 경우 시작 템플릿 또는 시작 사양에 네트워크 인터페이스 ID를 지정할 수 없습니다. 시작 템플릿 또는 시작 사양에서 `NetworkInterfaceID` 파라미터를 생략하세요.

예시

- 예제 1: 리전에서 최저 가격의 가용 영역을 사용하여 스팟 인스턴스 시작
- 예제 2: 목록에서 최저 가격의 가용 영역을 사용하여 스팟 인스턴스 시작
- 예제 3: 목록에서 최저 가격의 인스턴스 유형을 사용하여 스팟 인스턴스 시작
- 예 4. 요청에 대한 가격 재정의
- 예제 5: 다각화된 할당 전략을 사용하여 스팟 플릿 시작
- 예제 6: 인스턴스 가중치를 사용하여 스팟 플릿 시작
- 예제 7: 온디맨드 용량을 사용하여 스팟 플릿 시작
- 예제 8: 대체 스팟 인스턴스를 시작하도록 용량 리밸런싱 구성
- 예제 9: 용량 최적화 플릿에서 스팟 인스턴스 시작
- 예제 10: 용량 최적화 플릿에서 우선순위를 사용하여 스팟 인스턴스 시작
- 예제 11: priceCapacityOptimized 플릿에서 스팟 인스턴스 시작
- 예제 12: 속성 기반 인스턴스 유형 선택 구성

예제 1: 리전에서 최저 가격의 가용 영역을 사용하여 스팟 인스턴스 시작

다음 예는 가용 영역이나 서브넷이 없는 단일 시작 사양을 지정합니다. 스팟 플릿은 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"
        }
    }
  ]
}
```

예제 2: 목록에서 최저 가격의 가용 영역을 사용하여 스팟 인스턴스 시작

다음 예는 가용 영역이나 서브넷은 다르지만 인스턴스 유형과 AMI는 같은 두 개의 시작 사양을 지정합니다.

가용 영역

스팟 플릿은 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 플릿 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 할당하세요. 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::123456789012:instance-profile/my-iam-role"
        }
    }
...
```

예제 3: 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 플릿은 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": "c5.4xlarge",
        "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": "c5.4xlarge",
            "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 플릿은 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"
        }
    ]
}
```

예제 5: 다각화된 할당 전략을 사용하여 스팟 플릿 시작

다음 예제에서는 diversified 할당 전략을 사용합니다. 시작 사양의 인스턴스 유형은 다르지만 AMI와 가용 영역 또는 서브넷은 같습니다. 스팟 플릿은 3개의 시작 사양에 걸쳐 각 유형의 인스턴스가 10개씩 있도록 30개의 인스턴스를 분산합니다. 자세한 내용은 할당 전략을 사용하여 EC2 플릿 또는 스팟 플릿이 스팟 및 온디맨드 용량을 충족하는 방법 결정 섹션을 참조하세요.

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"
        }
    ]
}
```

예제 6: 인스턴스 가중치를 사용하여 스팟 플릿 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 플릿은 단위 시간당 최저 가격의 인스턴스 유형을 선택합니다. 스팟 플릿은 목표 용량을 인스턴스 가중치로 나누어 시작할 스팟 인스턴스의 수를 계산합니다. 결과가 정수가 아닌 경우 스팟 플릿은 결과를 다음 정수로 반올림하므로 플릿 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 인스턴스 가중치를 사용하여 EC2 플릿 또는 스팟 플릿의 비용 및 성능 관리 섹션을 참조하세요.

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
      }
  ]
}
```

예제 7: 온디맨드 용량을 사용하여 스팟 플릿 시작

항상 인스턴스 용량을 사용할 수 있도록 스팟 플릿 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 지정되지 않습니다. 스팟 용량은 목표 용량에서 온디맨드 용량을 뺀 잔액으로 암시됩니다. Amazon EC2는 온디맨드로 5개의 용량 단위를 시작하고 사용 가능한 Amazon EC2 용량(10-5=5) 및 가용성이 있는 경우 5개의 용량 단위를 스팟으로 시작합니다.

```
{
  "IamFleetRole": "arn:aws:iam::123456789012: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"
        }
      ]
    }
  ]
}
```

예제 8: 대체 스팟 인스턴스를 시작하도록 용량 리밸런싱 구성

다음 예제에서는 Amazon EC2에서 플릿의 스팟 인스턴스에 대한 리밸런싱 권고가 생성될 때 대체 스팟 인스턴스를 시작하도록 스팟 플릿을 구성합니다. 스팟 인스턴스의 자동 대체를 구성하려면 `ReplacementStrategy`에 `launch-before-terminate`를 지정합니다. 새 대체 스팟 인스턴스가 시작되는 시점부터 이전 스팟 인스턴스가 자동으로 삭제될 때까지의 시간 지연을 구성하려면 `termination-delay`에 대해 초 단위로 값을 지정합니다. 자세한 내용은 구성 옵션 섹션을 참조하세요.

> **Note**
>
> 인스턴스 종료 절차가 완료되는 데 걸리는 시간을 예측할 수 있는 경우에만 `launch-before-terminate`를 사용하는 것이 좋습니다. 이렇게 하면 종료 절차가 완료된 후에만 이전 인스턴스가 종료됩니다. 두 인스턴스가 실행되는 동안에는 두 인스턴스에 대해 요금이 청구됩니다.

용량 리밸런싱 전략의 효과는 스팟 플릿 요청에 지정된 스팟 용량 풀 수에 따라 달라집니다. 다양한 인스턴스 유형 및 가용 영역 세트로 플릿을 구성하고 `AllocationStrategy`에 대해 `capacityOptimized`를 지정하는 것이 좋습니다. 용량 리밸런싱에 대해 스팟 플릿을 구성할 때 고려해야 할 사항에 대한 자세한 내용은 EC2 플릿 및 스팟 플릿의 용량 리밸런싱을 사용하여 위험이 있는 스팟 인스턴스 교체 섹션을 참조하세요.

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "capacityOptimized",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "LaunchTemplate",
                    "Version": "1"
                },
                 "Overrides": [
                       {
                           "InstanceType": "c3.large",
                           "WeightedCapacity": 1,
                           "Placement": {
                               "AvailabilityZone": "us-east-1a"
                           }
```

```
                    },
                    {
                        "InstanceType": "c4.large",
                        "WeightedCapacity": 1,
                        "Placement": {
                            "AvailabilityZone": "us-east-1a"
                        }
                    },
                    {
                        "InstanceType": "c5.large",
                        "WeightedCapacity": 1,
                        "Placement": {
                            "AvailabilityZone": "us-east-1a"
                        }
                    }
                ]
            }
    ],
        "TargetCapacity": 5,
        "SpotMaintenanceStrategies": {
            "CapacityRebalance": {
                "ReplacementStrategy": "launch-before-terminate",
                "TerminationDelay": "720"
            }
        }
    }
}
```

예제 9: 용량 최적화 플릿에서 스팟 인스턴스 시작

다음 예제에서는 용량을 최적화하는 스팟 할당 전략을 사용하여 스팟 플릿을 구성하는 방법을 보여줍니다. 용량을 최적화하려면 `AllocationStrategy`를 `capacityOptimized`로 설정해야 합니다.

다음 예제에서는 3개의 시작 사양으로 3개의 스팟 용량 풀을 지정합니다. 목표 용량은 스팟 인스턴스 50개입니다. 스팟 플릿은 시작하는 인스턴스의 수에 대한 최적의 용량을 가진 스팟 용량 풀로 스팟 인스턴스 50개를 시작하려고 시도합니다.

```
{
    "TargetCapacity": "50",
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "capacityOptimized",
    },
    "LaunchTemplateConfigs": [
```

```
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1"
            },
            "Overrides": [
                {
                    "InstanceType": "r4.2xlarge",
                    "AvailabilityZone": "us-west-2a"
                },
                {
                    "InstanceType": "m4.2xlarge",
                    "AvailabilityZone": "us-west-2b"
                },
                {
                    "InstanceType": "c5.2xlarge",
                    "AvailabilityZone": "us-west-2b"
                }
            ]
        }
    ]
}
```

예제 10: 용량 최적화 플릿에서 우선순위를 사용하여 스팟 인스턴스 시작

다음 예제에서는 최선의 노력을 기준으로 우선순위를 사용하는 동안 용량을 최적화하는 스팟 할당 전략을 사용하여 스팟 플릿을 구성하는 방법을 보여줍니다.

`capacityOptimizedPrioritized` 할당 전략을 사용할 때 `Priority` 파라미터를 사용하여 스팟 인스턴스 풀의 우선순위를 지정할 수 있습니다. 여기서 숫자가 낮을수록 우선순위는 높습니다. 여러 스팟 용량 풀에 선호도가 동일한 경우 동일한 우선순위를 설정해도 됩니다. 우선순위가 설정되지 않은 풀은 우선순위 측면에서 마지막으로 고려됩니다.

스팟 용량 풀의 우선순위를 지정하려면 `AllocationStrategy`를 `capacityOptimizedPrioritized`로 설정해야 합니다. 스팟 플릿은 용량을 우선으로 최적화하지만 최선의 노력을 기준으로 우선순위를 따릅니다. 예를 들어 스팟 플릿에서 최적 용량으로 프로비저닝하는 데 우선순위가 큰 영향을 미치지 않을 수 있습니다. 이 옵션은 중단 가능성을 최소화해야 하고 특정 인스턴스 유형에 대한 선호도가 중요한 워크로드에 적합합니다.

다음 예제에서는 3개의 시작 사양으로 3개의 스팟 용량 풀을 지정합니다. 각 풀의 우선순위가 지정되며, 여기서 숫자가 낮을수록 우선순위가 높습니다. 목표 용량은 스팟 인스턴스 50개입니다. 스팟 플릿

은 가장 높은 우선순위의 스팟 용량 풀로 50개의 스팟 인스턴스를 시작하려고 시도하지만 먼저 용량을 최적화합니다.

```
{
    "TargetCapacity": "50",
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "capacityOptimizedPrioritized"
    },
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1"
            },
            "Overrides": [
                {
                    "InstanceType": "r4.2xlarge",
                   "Priority": 1,
                    "AvailabilityZone": "us-west-2a"
                },
                {
                           "InstanceType": "m4.2xlarge",
                           "Priority": 2,
                           "AvailabilityZone": "us-west-2b"
                },
                {
                           "InstanceType": "c5.2xlarge",
                           "Priority": 3,
                           "AvailabilityZone": "us-west-2b"
                }
            ]
        }
    ]
}
```

예제 11: priceCapacityOptimized 플릿에서 스팟 인스턴스 시작

다음 예제에서는 용량과 가격을 모두 최적화하는 스팟 할당 전략을 사용하여 스팟 플릿을 구성하는 방법을 보여줍니다. 가격을 고려하면서 용량을 최적화하려면 스팟 `AllocationStrategy`를 `priceCapacityOptimized`로 설정해야 합니다.

다음 예제에서는 3개의 시작 사양으로 3개의 스팟 용량 풀을 지정합니다. 목표 용량은 스팟 인스턴스 50개입니다. 스팟 플릿은 시작하는 인스턴스의 수에 대한 최적의 용량을 가진 스팟 용량 풀로 스팟 인스턴스 50개를 시작하는 동시에 가격이 가장 낮은 풀을 선택하려고 시도합니다.

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "OnDemandAllocationStrategy": "lowestPrice",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::111111111111:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateId": "lt-0123456789example",
                    "Version": "1"
                },
                "Overrides": [
                     {
                          "InstanceType": "r4.2xlarge",
                          "AvailabilityZone": "us-west-2a"
                     },
                     {
                          "InstanceType": "m4.2xlarge",
                          "AvailabilityZone": "us-west-2b"
                     },
                     {
                          "InstanceType": "c5.2xlarge",
                          "AvailabilityZone": "us-west-2b"
                     }
                ]
            }
        ],
        "TargetCapacity": 50,
        "Type": "request"
    }
}
```

예제 12: 속성 기반 인스턴스 유형 선택 구성

다음 예에서는 인스턴스 유형을 식별하는 데 속성 기반 인스턴스 유형 선택 방식을 사용하도록 스팟 플릿을 구성하는 방법을 보여줍니다. 필요한 인스턴스 속성을 지정하려면 `InstanceRequirements` 구조에서 해당 속성을 지정합니다.

다음 예제에서는 2개의 인스턴스 속성이 지정됩니다.

- `VCpuCount` - vCPU가 최소 2개로 지정되었습니다. 최댓값이 지정되지 않았으므로 최대 한도는 없습니다.
- `MemoryMiB` - 메모리가 최소 4MiB로 지정되었습니다. 최댓값이 지정되지 않았으므로 최대 한도는 없습니다.

vCPU가 2개 이상이고 메모리가 4MiB 이상인 모든 인스턴스 유형이 식별됩니다. 단, 스팟 플릿에서 플릿을 제공하는 경우 가격 보호 및 할당 전략에서 일부 인스턴스 유형이 제외될 수 있습니다.

지정 가능한 모든 속성의 목록과 설명은 Amazon EC2 API 참조에서 InstanceRequirements를 참조하세요.

```
{
 "AllocationStrategy": "priceCapacityOptimized",
 "TargetCapacity": 20,
 "Type": "request",
 "LaunchTemplateConfigs": [{
  "LaunchTemplateSpecification": {
   "LaunchTemplateName": "my-launch-template",
   "Version": "1"
  },
  "Overrides": [{
   "InstanceRequirements": {
    "VCpuCount": {
     "Min": 2
    },
    "MemoryMiB": {
     "Min": 4
    }
   }
  }]
 }]
}
```

EC2 플릿 및 스팟 플릿 할당량

AWS 계정에는 각 AWS 서비스에 대한 기본 할당량(이전에는 제한이라고 함)이 있습니다. 다르게 표시되지 않는 한, 지역별로 각 할당량이 적용됩니다.

EC2 플릿 또는 스팟 플릿에서 시작되는 인스턴스에는 스팟 인스턴스 제한 및 볼륨 제한과 같은 일반적인 Amazon EC2 할당량이 적용됩니다.

추가로 AWS 계정에는 EC2 플릿 및 스팟 플릿과 관련된 할당량이 있습니다.

할당량 설명	할당량
리전당 `active`, `deleted_running` , `cancelled_running` 상태인 `maintain` 및 `request` 유형의 EC2 플릿 및 스팟 플릿 수	1,000 [1] [2] [3]
`instant` 유형의 EC2 플릿 수	무제한
`maintain` 및 `request` 유형의 EC2 플릿 및 스팟 플릿에 대한 스팟 용량 풀 수(인스턴스 유형과 서브넷의 고유한 조합)	300 [1]
`instant` 유형의 EC2 플릿에 대한 스팟 용량 풀 수(인스턴스 유형과 서브넷의 고유한 조합)	무제한
시작 사양의 사용자 데이터 크기	16KB [2]
EC2 플릿 또는 스팟 플릿당 목표 용량	10,000개
리전 내 모든 EC2 플릿 및 스팟 플릿의 목표 용량	100,000 [1]
스팟 플릿 요청 또는 스팟 플릿 요청으로 리전을 확장할 수 없습니다.	
스팟 플릿 요청 또는 스팟 플릿 요청으로 동일한 가용 영역의 서로 다른 서브넷을 확장할 수 없습니다.	

[1] 이러한 할당량은 사용자의 EC2 플릿 및 스팟 플릿 모두에 적용됩니다.

[2] 하드 할당량입니다. 사용자는 이러한 할당량의 증가를 요청할 수 없습니다.

[3] EC2 플릿을 삭제하거나 스팟 플릿 요청을 취소한 후, 그리고 사용자가 요청을 삭제하거나 취소 시 플릿이 스팟 인스턴스를 종료하지 않아야 한다고 지정한 경우, 플릿 요청은 `deleted_running`(EC2 플릿) 또는 `cancelled_running`(스팟 플릿) 상태가 되고 인스턴스는 중단되거나 사용자가 수동으로 종료하지 않는 한 계속 실행됩니다. 만약 인스턴스를 종료하면 플릿 요청은 `deleted_terminating`(EC2 플릿) 또는 `cancelled_terminating`(스팟 플릿) 상태가 되며 이러한 할당량에 포함되지 않습니다. 자세한 내용은 EC2 플릿 요청 및 플릿의 인스턴스 삭제 및 스팟 플릿 요청 취소(삭제)(을)를 참조하세요.

목표 용량의 할당량 증가 요청

목표 용량의 기본 할당량 이상이 필요한 경우 할당량 증가를 요청할 수 있습니다.

목표 용량의 할당량 증가 요청

1. 지원 센터 사례 생성(Create case) 양식을 엽니다.
2. 서비스 한도 증가(Service Limit increase)를 선택합니다.
3. 제한 유형(Limit type)에서 EC2 플릿(EC2 Fleet)을 선택합니다.
4. 리전에서 할당량 증가를 요청할 AWS 리전을 선택합니다.
5. 한도(Limit)에서, 증가시킬 할당량에 따라 플릿당 목표 플릿 용량(단위)(Target Fleet Capacity per Fleet (in units)) 또는 리전당 목표 플릿 용량(단위)(Target Fleet Capacity per Region (in units))을 선택합니다.
6. 새 한도 값(New limit value)에 새 할당량 값을 입력합니다.
7. 다른 할당량 증가를 요청하려면 다른 요청 추가(Add another request)를 선택하고 4~6단계를 반복합니다.
8. 사용 사례 설명(Use case description)에서 할당량 증가를 요청하는 이유를 입력합니다.
9. 문의 옵션(Contact options)에서 선호하는 문의 언어와 문의 방법을 지정합니다.
10. 제출을 선택합니다.

Amazon EC2의 네트워킹

Amazon VPC를 사용하면 Virtual Private Cloud(VPC)라는 AWS 계정 전용 가상 네트워크로 Amazon EC2 인스턴스와 같은 AWS 리소스를 시작할 수 있습니다. 인스턴스를 시작할 때 VPC에서 서브넷을 선택할 수 있습니다. 인스턴스는 논리적 가상 네트워크 카드인 기본 네트워크 인터페이스로 구성됩니다. 인스턴스는 서브넷의 IPv4 주소에서 기본 프라이빗 IP 주소를 수신하며, 이 주소는 기본 네트워크 인터페이스에 할당됩니다.

인스턴스가 Amazon의 퍼블릭 IP 주소 풀에서 퍼블릭 IP 주소를 수신하는지 여부를 제어할 수 있습니다. 인스턴스의 퍼블릭 IP 주소는 인스턴스가 중지되거나 종료될 때까지만 인스턴스와 연결됩니다. 영구 퍼블릭 IP 주소가 필요한 경우 AWS 계정에 탄력적 IP 주소를 할당하고 이를 인스턴스 또는 네트워크 인터페이스에 연결할 수 있습니다. 탄력적 IP 주소는 사용자가 해제할 때까지 AWS 계정과 연결되며, 필요한 경우 하나의 인스턴스에서 다른 이스턴스로 이동할 수 있습니다. 자신의 IP 주소 범위를 AWS 계정에 가져와서 주소 풀로 표시한 다음 주소 풀에서 탄력적 IP 주소를 할당할 수 있습니다.

네트워크 성능을 높이고 지연 시간을 줄이기 위해 배치 그룹으로 인스턴스를 시작할 수 있습니다. 향상된 네트워킹을 사용하여 PPS(초당 패킷) 성능을 크게 높일 수 있습니다. 지원되는 인스턴스 유형에 연결할 수 있는 네트워크 디바이스인 Elastic Fabric Adapter(EFA)를 사용하여 고성능 컴퓨팅 및 기계 학습 애플리케이션을 가속화할 수 있습니다.

특성

- 리전 및 영역
- Amazon EC2 인스턴스 IP 주소 지정
- EC2 인스턴스 호스트 이름 및 도메인
- Amazon EC2로 고유 IP 주소 가져오기(BYOIP)
- 탄력적 IP 주소
- 탄력적 네트워크 인터페이스
- Amazon EC2 인스턴스 네트워크 대역폭
- Amazon EC2 인스턴스의 향상된 네트워킹
- Amazon EC2의 AI/ML 및 HPC 워크로드를 위한 Elastic Fabric Adapter
- Amazon EC2 인스턴스 토폴로지
- Amazon EC2 인스턴스에 대한 배치 그룹
- EC2 인스턴스에 대한 네트워크 MTU(최대 전송 단위)
- EC2 인스턴스용 가상 프라이빗 클라우드

리전 및 영역

Amazon EC2는 전 세계의 여러 곳에서 호스팅되고 있습니다. 이 위치는 AWS 리전, 가용 영역, Local Zones, AWS Outposts 및 Wavelength Zone으로 구성됩니다.

- 리전은 별도의 지리적 영역입니다.
- 가용 영역은 각 리전 내에 있는 여러 격리된 위치입니다.
- 로컬 영역에서는 최종 사용자와 가까운 여러 위치에 컴퓨팅, 스토리지 등의 리소스를 배치할 수 있는 기능을 제공합니다.
- Wavelength Zone에서는 5G 디바이스 및 최종 사용자에게 매우 짧은 지연 시간을 지원하는 애플리케이션을 빌드하는 기능을 제공합니다. Wavelength는 표준 AWS 컴퓨팅 및 스토리지 서비스를 통신 사업자의 5G 네트워크 엣지에 배포합니다.
- AWS Outposts는 네이티브 AWS 서비스, 인프라 및 운영 모델을 사실상 모든 데이터 센터, 콜로케이션 공간 또는 온프레미스 시설로 옮길 수 있습니다.

AWS은 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.

자세한 내용은 AWS 글로벌 인프라를 참조하세요.

내용

- 리전
- 가용 영역
- 로컬 영역
- Wavelength Zone
- AWS Outposts

리전

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

인스턴스를 시작할 때 특정 고객과 가까운 곳에 인스턴스를 배치하거나 법률 또는 기타 요구 사항을 충족하는 리전을 선택합니다. 여러 리전에서 인스턴스를 시작할 수 있습니다.

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

사용 가능한 리전

사용 가능한 리전 목록은 AWS 리전을 참조하세요.

Amazon EC2에 대한 리전별 엔드포인트

명령줄 인터페이스 또는 API 작업을 사용해서 인스턴스로 작업할 때는 리전 엔드포인트를 지정해야 합니다. Amazon EC2의 리전과 엔드포인트에 대한 자세한 내용은 Amazon EC2 개발자 안내서의 Amazon EC2 service endpoints를 참조하세요.

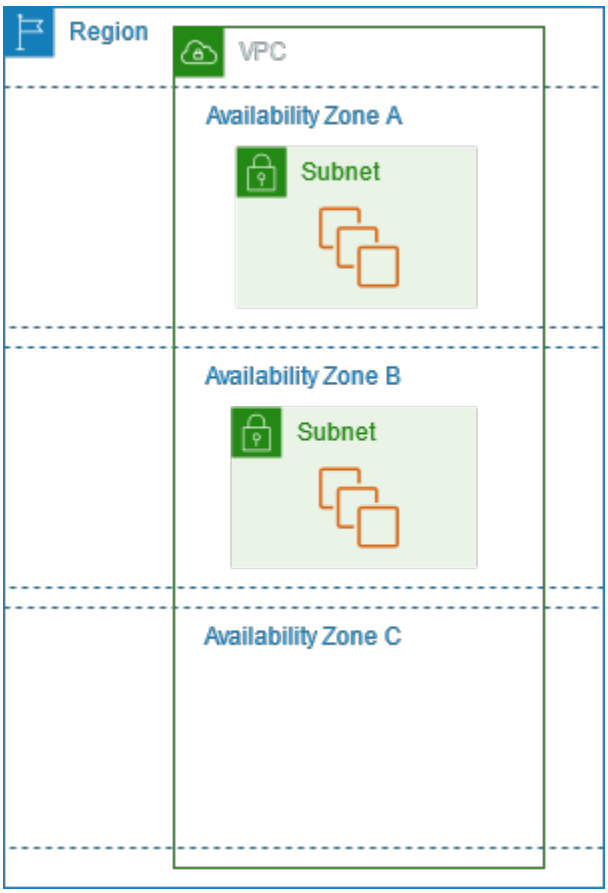
자세한 내용은 AWS 리전 and Availability Zones User Guide의 AWS 리전를 참조하세요.

가용 영역

각 리전은 가용 영역이라고 알려진 격리된 위치를 여러 개 가지고 있습니다. 가용 영역의 코드는 리전 코드와 문자 식별자를 조합한 것입니다. 예를 들어 `us-east-1a`입니다.

여러 가용 영역에서 EC2 인스턴스를 시작함으로써 리전의 단일 위치에서 장애가 발생할 경우 애플리케이션을 보호할 수 있습니다.

다음 다이어그램은 AWS 리전의 여러 가용 영역을 보여줍니다. 가용 영역 A 및 가용 영역 B에는 각각 하나의 서브넷이 있고 각 서브넷에는 EC2 인스턴스가 있습니다. 가용 영역 C에는 서브넷이 없으므로 이 가용 영역으로 인스턴스를 시작할 수 없습니다.



자세한 내용은 EC2 인스턴스용 가상 프라이빗 클라우드 섹션을 참조하세요.

리전별 가용 영역

리전별 가용 영역 목록은 AWS 가용 영역을 참조하세요.

가용 영역의 인스턴스

인스턴스를 시작할 때 리전 및 가상 프라이빗 클라우드(VPC)를 선택합니다. 그런 다음 사용자가 가용 영역 중 하나에서 서브넷을 선택하거나 사용자를 대신해 자동으로 서브넷을 선택할 수 있습니다. 초기 인스턴스를 시작하는 경우 시스템 상태 및 가용 용량에 따라 사용자를 대신해 가용 영역을 선택하도록 하는 것이 좋습니다. 추가 인스턴스를 시작하는 경우 새 인스턴스가 기존 인스턴스와 가까이 있거나 분리되어 있어야 하는 경우에만 가용 영역을 지정합니다.

여러 가용 영역에 걸쳐 인스턴스를 배포했을 때 하나의 인스턴스에서 장애가 발생한 경우 다른 가용 영역의 인스턴스가 대신 요청을 처리하도록 애플리케이션을 설계할 수 있습니다.

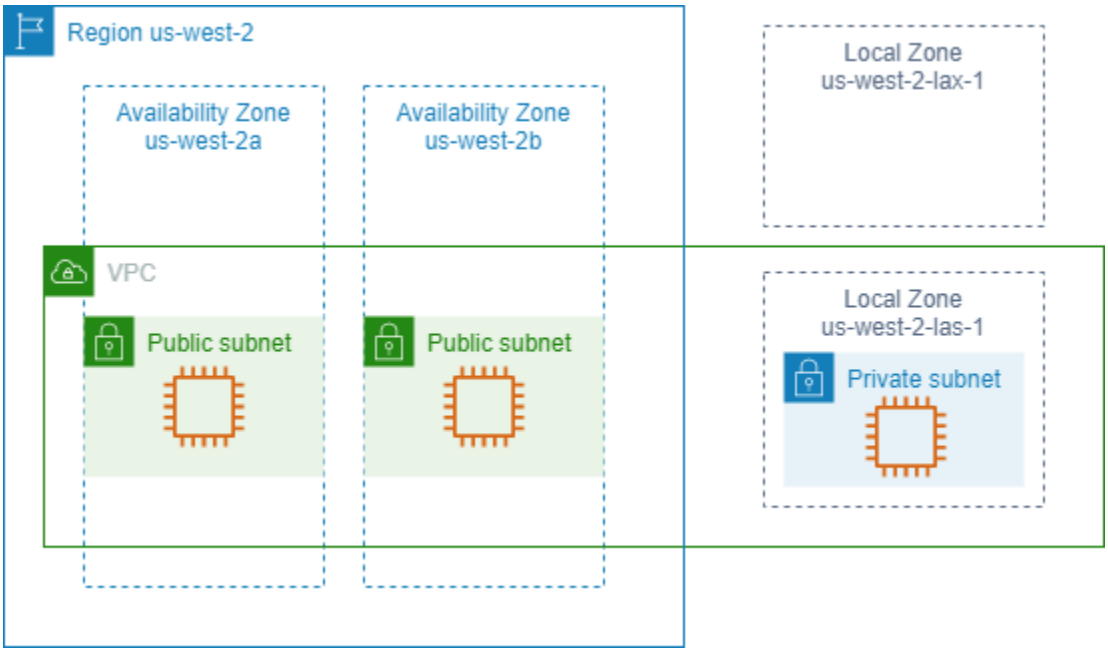
자세한 내용은 AWS 리전 and Availability Zones User Guide의 AWS Availability Zones를 참조하세요.

로컬 영역

Local Zone은 사용자와 지리적으로 근접한 AWS 리전의 확장입니다. Local Zones는 자체 연결을 통해 인터넷에 연결되며 AWS Direct Connect를 지원하므로 Local Zone에 생성된 리소스를 짧은 지연 시간의 통신을 통해 로컬 사용자에게 제공할 수 있습니다. 자세한 내용은 AWS Local Zones 사용 설명서의 What is AWS Local Zones?를 참조하세요.

로컬 영역에 대한 코드는 물리적 위치를 나타내는 식별자가 뒤에 붙는 리전 코드입니다. 로스앤젤레스의 `us-west-2-lax-1`을 예로 들 수 있습니다.

다음 다이어그램에서는 AWS 리전 `us-west-2`, 가용 영역 2개, Local Zones 2개를 보여줍니다. VPC는 가용 영역과 Local Zones 중 하나에 걸쳐 있습니다. VPC의 각 영역에는 하나의 서브넷이 있고 각 서브넷에는 인스턴스가 있습니다.



사용 가능한 로컬 영역

사용 가능한 로컬 영역 목록은 AWS Local Zones User Guide의 Available Local Zones를 참조하세요. 발표된 로컬 영역 목록은 AWS Local Zones locations를 참조하세요.

로컬 영역의 인스턴스

Local Zone을 사용하려면 먼저 사용하도록 설정해야 합니다. 그런 다음 로컬 영역에서 서브넷을 생성합니다. 인스턴스를 시작할 때 로컬 영역 서브넷을 지정하여 로컬 영역의 로컬 영역 서브넷에 배치할 수 있습니다.

로컬 영역에서 인스턴스를 시작할 때 네트워크 경계 그룹의 IP 주소도 할당합니다. 네트워크 경계 그룹은 AWS가 IP 주소를 알릴 때 사용하는 가용 영역, Local Zones 또는 Wavelength Zone의 고유한 집합(예: `us-west-2-lax-1a`)입니다. 네트워크 경계 그룹에서 다음 IP 주소를 할당할 수 있습니다.

- Amazon에서 제공하는 탄력적 IPv4 주소
- Amazon 제공 IPv6 VPC 주소(로스앤젤레스 영역에서만 사용 가능)

로컬 영역에서 인스턴스를 시작하는 방법에 대한 자세한 내용은 AWS 로컬 영역 사용 설명서에서 AWS 로컬 영역 시작하기를 참조하세요.

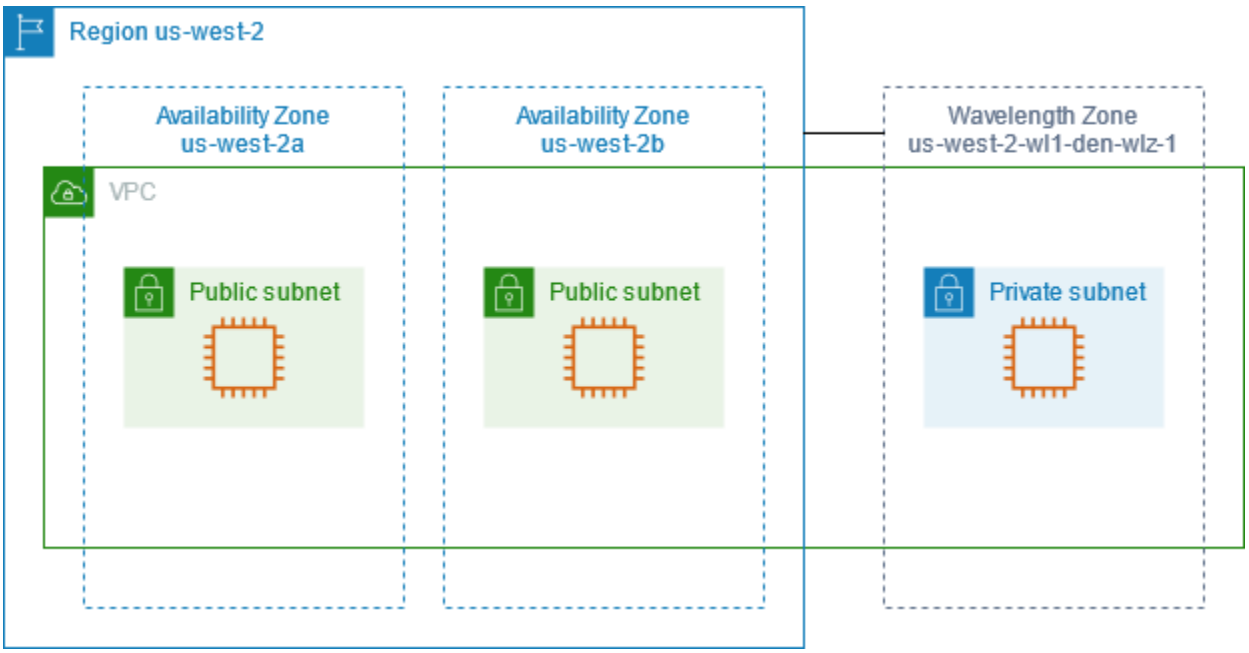
Wavelength Zone

AWS Wavelength을(를) 사용하면 개발자는 모바일 디바이스 및 최종 사용자에게 매우 짧은 지연 시간을 제공하는 애플리케이션을 빌드할 수 있습니다. Wavelength는 표준 AWS 컴퓨팅 및 스토리지 서비스를 통신 사업자의 5G 네트워크 엣지에 배포합니다. 개발자는 Virtual Private Cloud(VPC)를 하나 이상의 Wavelength Zone으로 확장한 다음, Amazon EC2 인스턴스와 같은 AWS 리소스를 사용하여 매우 짧은 지연 시간으로 리전의 AWS 서비스에 연결해야 하는 애플리케이션을 실행할 수 있습니다.

Wavelength Zone은 Wavelength 인프라가 배포된 통신 사업자 위치의 격리된 영역입니다. Wavelength Zone은 리전에 연결되어 있습니다. Wavelength Zone은 리전의 논리적 확장이며, 리전의 제어 플레인에 의해 관리됩니다.

Wavelength 영역에 대한 코드는 물리적 위치를 나타내는 식별자가 뒤에 붙는 리전 코드입니다. 보스턴의 `us-east-1-wl1-bos-wlz-1`을 예로 들 수 있습니다.

다음 다이어그램에서는 AWS 리전 `us-west-2`, 가용 영역 2개, Wavelength 영역 1개를 보여줍니다. VPC는 가용 영역과 Wavelength 영역에 걸쳐 있습니다. VPC의 각 영역에는 하나의 서브넷이 있고 각 서브넷에는 인스턴스가 있습니다.



Wavelength Zone은 일부 리전에서 사용할 수 없습니다. Wavelength Zone을 지원하는 리전에 대한 자세한 내용은 AWS Wavelength 개발자 안내서의 사용 가능한 Wavelength Zone을 참조하세요.

사용 가능한 Wavelength 영역

사용 가능한 Wavelength 영역 목록은 AWS Wavelength Guide의 Available Wavelength Zones를 참조하세요.

Wavelength 영역의 인스턴스

Wavelength Zone을 사용하려면 먼저 Wavelength Zone을 옵트인해야 합니다. 그런 다음 Wavelength 영역에서 서브넷을 생성합니다. 인스턴스를 시작할 때 Wavelength 서브넷을 지정할 수 있습니다. 또한 네트워크 경계 그룹에서 통신 사업자 IP 주소를 할당합니다. 네트워크 경계 그룹은 AWS가 IP 주소를 알리는 가용 영역, Local Zones 또는 Wavelength Zone의 고유한 집합(예: `us-east-1-wl1-bos-wlz-1`)입니다.

Wavelength 영역에서 인스턴스를 시작하는 단계별 지침은 AWS Wavelength Developer Guide의 Get started with AWS Wavelength를 참조하세요.

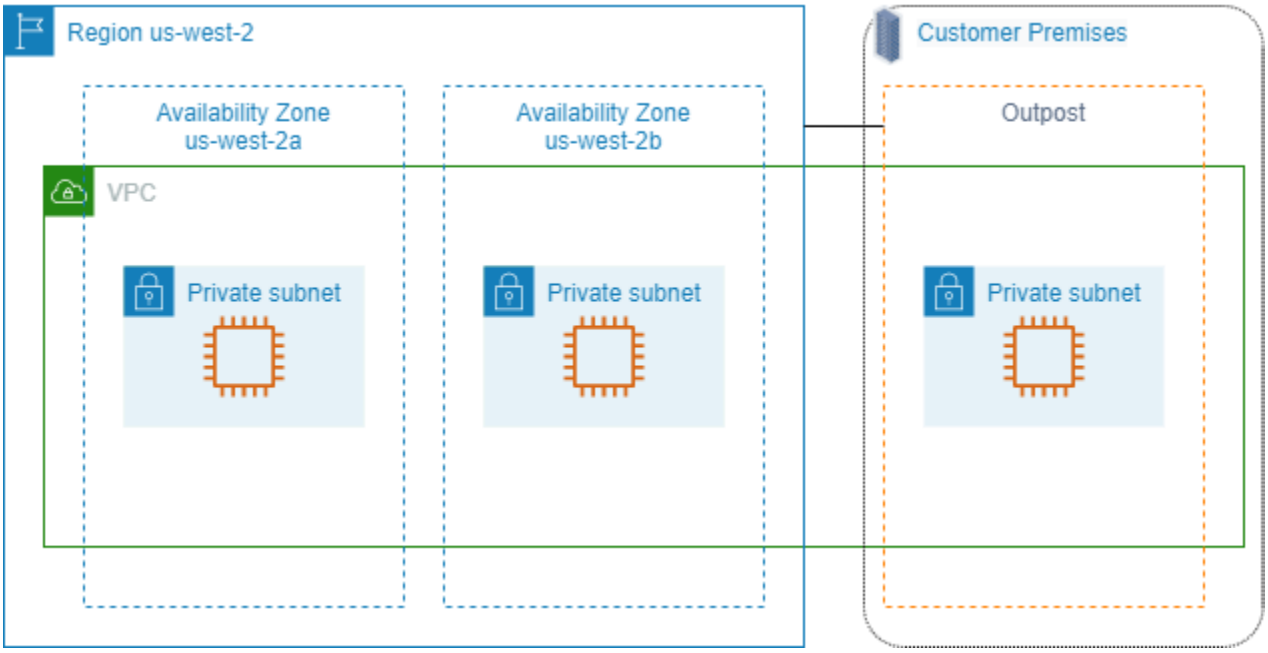
AWS Outposts

AWS Outposts은(는) AWS 인프라, 서비스, API 및 도구를 고객 온프레미스로 확장하는 완전관리형 서비스입니다. AWS 관리형 인프라에 대한 로컬 액세스를 제공하는 AWS Outposts을(를) 통해 고객은 AWS 리전에서 사용하는 것과 동일한 프로그래밍 인터페이스를 사용해 온프레미스에서 애플리케이션

을 구축하고 실행할 수 있으며, 짧은 지연 시간과 로컬 데이터 처리가 필요한 경우에 로컬 컴퓨팅 및 스토리지 리소스를 사용할 수 있습니다.

Outpost는 고객 사이트에 배포된 AWS의 컴퓨팅 및 스토리지 용량 풀입니다. AWS는 이 용량을 AWS 리전의 일부로 운영, 모니터링 및 관리합니다. AWS 리소스를 생성할 때 Outpost에 서브넷을 생성하고 해당 서브넷을 지정할 수 있습니다. Outpost 서브넷의 인스턴스는 프라이빗 IP 주소를 사용하여 AWS 리전의 다른 인스턴스와 통신합니다(모두 동일한 VPC에 있음).

다음 다이어그램에서는 AWS 리전 `us-west-2`, 가용 영역 2개, Outpost 1개를 보여줍니다. VPC는 가용 영역과 Outpost에 걸쳐 있습니다. Outpost는 온프레미스 고객 데이터 센터에 있습니다. VPC의 각 영역에는 하나의 서브넷이 있고 각 서브넷에는 인스턴스가 있습니다.



Outpost의 인스턴스

AWS Outposts를 사용하려면 Outpost를 만들고 Outpost 용량을 주문해야 합니다. AWS Outposts는 Outposts 랙과 Outposts 서버라는 두 가지 폼 팩터를 제공합니다. Outposts 구성에 대한 자세한 내용은 AWS Outposts 패밀리를 참조하세요. Outposts 장비를 설치한 후 Outpost에서 EC2 인스턴스를 시작할 때 컴퓨팅 및 스토리지 용량을 사용할 수 있습니다.

EC2 인스턴스를 시작하려면 Outpost 서브넷을 생성해야 합니다. 보안 그룹은 가용 영역 서브넷의 인스턴스와 마찬가지로 Outpost 서브넷의 인스턴스에 대한 인바운드 및 아웃바운드 트래픽을 제어합니다. SSH를 사용하여 Outpost 서브넷에서 인스턴스에 연결하려면 가용 영역 서브넷의 인스턴스에서와 마찬가지로 인스턴스를 시작할 때 키 페어를 지정합니다.

자세한 내용은 Outposts 랙 시작하기 또는 Outposts 서버 시작하기를 참조하세요.

Outpost 랙의 볼륨

Outpost 랙에 Outposts 컴퓨팅 용량이 있으면 생성한 Outpost 서브넷에 EBS 볼륨을 만들 수 있습니다. 볼륨을 만들 때 Outposts의 Amazon 리소스 이름(ARN)을 지정합니다.

다음 create-volume 명령은 지정된 Outposts에 빈 50GB 볼륨을 만듭니다.

```
aws ec2 create-volume --availability-zone us-east-2a --outpost-arn arn:aws:outposts:us-east-2:123456789012:outpost/op-03e6fecad652a6138 --size 50
```

Amazon EBS gp2 볼륨을 분리하지 않고 해당 볼륨의 크기를 동적으로 수정할 수 있습니다. 볼륨을 분리하지 않고 수정하는 방법에 대한 자세한 내용은 Amazon EBS 사용 설명서의 EBS 볼륨에 대한 수정 요청을 참조하세요.

Outpost 랙에서 인스턴스의 루트 볼륨은 30GiB 이하로 제한하는 것이 좋습니다. AMI 또는 인스턴스의 블록 디바이스 매핑에서 데이터 볼륨을 지정하여 추가 스토리지를 제공할 수 있습니다. 부팅 볼륨에서 사용되지 않는 블록을 트리밍하려면 AWS 파트너 네트워크 블로그에서 Sparse EBS 볼륨 구축 방법을 참조하세요.

루트 볼륨에 대한 NVMe 제한 시간을 늘리는 것이 좋습니다. 자세한 내용은 Amazon EBS 사용 설명서의 I/O 작업 제한 시간을 참조하세요.

Outposts 서버의 볼륨

Outpost 서버의 인스턴스는 인스턴스 스토어 볼륨을 제공하지만 EBS 볼륨은 지원하지 않습니다. 하나의 EBS 스냅샷만 있는 Amazon EBS 지원 AMI를 선택합니다. 애플리케이션의 요구 사항을 충족하기에 충분한 인스턴스 스토리지를 갖춘 인스턴스 크기를 선택합니다. 자세한 내용은 인스턴스 저장소 볼륨 제한 섹션을 참조하세요.

Amazon EC2 인스턴스 IP 주소 지정

Amazon EC2와 Amazon VPC는 IPv4 및 IPv6 주소 지정 프로토콜을 모두 지원합니다. 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 VPC에 할당하고 그 블록에 속한 IPv6 주소를 서브넷의 인스턴스에 할당할 수도 있습니다.

EC2 인스턴스를 시작할 때 VPC 및 서브넷을 지정할 수 있습니다. 인스턴스에는 서브넷 CIDR 범위 내의 프라이빗 IPv4 주소가 할당됩니다. 인스턴스를 퍼블릭 IPv4 주소 및 IPv6 주소로 선택적으로 구성

할 수 있습니다. 서로 다른 VPC의 EC2 인스턴스가 퍼블릭 IP 주소를 사용하여 통신하는 경우 트래픽은 AWS 프라이빗 글로벌 네트워크에 유지되고 퍼블릭 인터넷을 통과하지 않습니다.

내용

- 프라이빗 IPv4 주소
- 퍼블릭 IPv4 주소
- 퍼블릭 IPv4 주소 최적화
- IPv6 주소
- 다중 IP 주소
- EC2 인스턴스 호스트 이름
- 링크-로컬 주소
- EC2 인스턴스에 대한 IPv4 주소 관리
- EC2 인스턴스에 대한 IPv6 주소 관리
- EC2 인스턴스에 대한 보조 IP 주소
- Windows 인스턴스의 보조 프라이빗 IPv4 주소 구성

프라이빗 IPv4 주소

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 참조하세요. 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.

VPC 서브넷은 다음 유형 중 하나일 수 있습니다.

- IPv4 전용 서브넷 - 이러한 서브넷에는 IPv4 주소가 할당된 리소스만 생성할 수 있습니다.
- IPv6 전용 서브넷 - 이러한 서브넷에는 IPv6 주소가 할당된 리소스만 생성할 수 있습니다.
- IPv4 및 IPv6 서브넷 - 이러한 서브넷에는 IPv4 또는 IPv6 주소가 할당된 리소스만 생성할 수 있습니다.

EC2 인스턴스를 IPv4 전용 또는 이중 스택(IPv4 및 IPv6) 서브넷에서 시작할 경우, 인스턴스는 서브넷의 IPv4 주소 범위에서 기본 프라이빗 IP 주소를 수신합니다. 자세한 내용은 Amazon VPC 사용 설명서의 IP 주소 지정을 참조하세요. 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(index 0)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 섹션을 참조하세요.

프라이빗 IPv4 주소는 기본 주소인지 보조 주소인지와 관계없이 인스턴스가 중지되었다가 시작될 때 또는 최대 절전 모드로 전환되었다가 시작될 때 네트워크 인터페이스와 연결이 유지되고 인스턴스가 종료되면 릴리스됩니다.

퍼블릭 IPv4 주소

퍼블릭 IP 주소는 인터넷을 통해 연결할 수 있는 IPv4 주소입니다. 퍼블릭 주소는 인스턴스와 인터넷의 상호 통신을 위해 사용될 수 있습니다.

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 속성을 수정합니다. 자세한 내용은 Amazon VPC 사용 설명서의 서브넷의 퍼블릭 IPv4 주소 지정 속성 수정을 참조하세요.
- 시작 시 퍼블릭 IP 주소 지정 기능을 활성화 또는 비활성화합니다. 이는 서브넷의 퍼블릭 IP 주소 지정 속성을 재정의합니다. 자세한 내용은 시작 시 퍼블릭 IPv4 주소 할당 섹션을 참조하세요.
- 시작 후 인스턴스에서 퍼블릭 IP 주소 할당을 취소합니다. 자세한 내용은 the section called “IP 주소 관리” 섹션을 참조하세요.

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.

다음과 같은 경우에 인스턴스에서 퍼블릭 IP 주소를 해제하거나 새 퍼블릭 IP 주소를 할당합니다.

- 인스턴스가 중지되거나 최대 절전 모드로 전환되거나 종료되면 퍼블릭 IP 주소가 해제됩니다. 사용자가 중지되거나 최대 절전 모드로 전환된 인스턴스를 시작하면 새 퍼블릭 IP 주소가 할당됩니다.
- 사용자가 탄력적 IP 주소를 인스턴스와 연결하면 퍼블릭 IP 주소가 해제됩니다. 사용자가 인스턴스에서 탄력적 IP 주소의 연결을 해제하면 새 퍼블릭 IP 주소가 할당됩니다.
- 인스턴스의 퍼블릭 IP 주소가 해제되고 보조 네트워크 인터페이스가 있는 경우 새 퍼블릭 IP 주소가 할당되지 않습니다.
- 인스턴스의 퍼블릭 IP 주소가 해제되고 탄력적 IP 주소와 연결된 보조 프라이빗 IP 주소가 있는 경우 새 퍼블릭 IP 주소가 할당되지 않습니다.

필요에 따라 인스턴스 간에 연결할 수 있는 영구 퍼블릭 IP 주소가 필요한 경우 탄력적 IP 주소를 대신하여 사용합니다.

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 섹션을 참조하세요.

Amazon VPC IP 주소 관리자(IPAM)를 사용하는 경우, AWS에서 퍼블릭 IPv4 주소의 연속 블록을 가져오고 AWS 리소스에 탄력적 IP 주소를 할당하는 데 사용할 수 있습니다. 연속적 IPv4 주소 블록을 사용하면 보안 액세스 제어 목록에 대한 관리 오버헤드를 크게 줄이고 AWS의 규모를 조정하는 기업의 IP 주소 할당 및 추적을 단순화할 수 있습니다. 자세한 내용은 Amazon VPC IPAM 사용 설명서의 IPAM 풀에서 순차적 탄력적 IP 주소 할당을 참조하세요.

고려 사항

- AWS에서는 탄력적 IP 주소 및 실행 중인 인스턴스에 연결된 퍼블릭 IPv4 주소를 포함하여 모든 퍼블릭 IPv4 주소에 요금을 부과합니다. 자세한 내용은 Amazon VPC 요금 페이지의 퍼블릭 IPv4 주소 탭을 참조하세요.
- 인스턴스가 동일 리전에 존재하는지 여부에 따라 퍼블릭 NAT IP 주소를 통해 다른 인스턴스에 액세스하는 인스턴스에는 리전별 또는 인터넷 데이터 전송 비용이 청구됩니다.

퍼블릭 IPv4 주소 최적화

AWS에서는 탄력적 IP 주소 및 실행 중인 인스턴스에 연결된 퍼블릭 IPv4 주소를 포함하여 모든 퍼블릭 IPv4 주소에 요금을 부과합니다. 자세한 내용은 Amazon VPC 요금 페이지의 퍼블릭 IPv4 주소 탭을 참조하세요.

다음 목록에는 사용하는 퍼블릭 IPv4 주소 수를 최적화하기 위해 취할 수 있는 조치가 포함되어 있습니다.

- Elastic Load Balancer를 사용하여 EC2 인스턴스에 대한 트래픽을 로드 밸런싱하고 인스턴스에 할당된 기본 ENI에서 퍼블릭 IP 자동 할당을 비활성화하세요. 로드 밸런서는 단일 퍼블릭 IPv4 주소를 사용하므로 퍼블릭 IPv4 주소 수가 줄어듭니다. 기존 로드 밸런서를 통합하여 퍼블릭 IPv4 주소 수를 더 줄일 수도 있습니다.
- NAT 게이트웨이를 사용하는 유일한 이유가 유지 관리 또는 긴급 상황을 위해 프라이빗 서브넷의 EC2 인스턴스에 SSH로 연결하는 것이라면 EC2 Instance Connect 엔드포인트를 대신 사용하는 것이 좋습니다. EC2 Instance Connect 엔드포인트를 사용하면 인스턴스에 퍼블릭 IPv4 주소가 없어도 인터넷에서 인스턴스에 연결할 수 있습니다.
- EC2 인스턴스가 퍼블릭 IP 주소가 할당된 퍼블릭 서브넷에 있는 경우 인스턴스를 프라이빗 서브넷으로 이동하고, 퍼블릭 IP 주소를 제거하고, 퍼블릭 NAT 게이트웨이를 사용하여 EC2 인스턴스와의 액세스를 허용하는 것이 좋습니다. NAT 게이트웨이 사용 시 비용 고려 사항이 있습니다. 이 계산 방법을 사용하여 NAT 게이트웨이가 비용 효율적인지 결정하세요. AWS 결제 비용 및 사용 보고서를 생성하면 이 계산에 필요한 `Number of public IPv4 addresses` 정보를 얻을 수 있습니다.

  ```
  NAT gateway per hour + NAT gateway public IPs + NAT gateway transfer / Existing
   public IP cost
  ```

  위치:

  - `NAT gateway per hour = $0.045 * 730 hours in a month * Number of Availability Zones the NAT gateways are in`
  - `NAT gateway public IPs = $0.005 * 730 hours in a month * Number of IPs associated with your NAT gateways`
  - `NAT gateway transfer = $0.045 * Number of GBs that will go through the NAT gateway in a month`
  - `Existing public IP cost = $0.005 * 730 hours in a month * Number of public IPv4 addresses`

  합계가 1보다 작은 경우 NAT 게이트웨이가 퍼블릭 IPv4 주소보다 저렴합니다.
- 퍼블릭 IPv4 주소 및 인터넷 게이트웨이를 사용하는 대신 AWS PrivateLink를 사용하여 AWS 서비스 또는 다른 AWS 계정에서 호스팅되는 서비스에 비공개로 연결합니다.
- Amazon 소유의 퍼블릭 IPv4 주소를 사용하는 대신 AWS로 고유 IP 주소 범위를 가져와서(BYOIP) 퍼블릭 IPv4 주소에 대한 범위를 사용합니다.

- 서브넷에서 시작된 인스턴스에 대한 퍼블릭 IPv4 주소 자동 할당을 끕니다 이 옵션은 일반적으로 서브넷을 생성할 때 VPC에 대해 기본적으로 비활성화되지만 기존 서브넷은 비활성화되어 있는지 확인해야 합니다.
- 퍼블릭 IPv4 주소가 필요하지 않은 EC2 인스턴스가 있는 경우 인스턴스에 연결된 네트워크 인터페이스에서 퍼블릭 IP 자동 할당이 비활성화되어 있는지 확인하세요.
- 프라이빗 서브넷의 EC2 인스턴스에 대해 AWS Global Accelerator에서 액셀러레이터 엔드포인트를 구성하면 퍼블릭 IP 주소 없이도 인터넷 트래픽이 VPC의 엔드포인트로 직접 전달되게 할 수 있습니다. 또한 AWS Global Accelerator로 고유 주소를 가져와 액셀러레이터의 고정 IP 주소로 고유 IPv4 주소를 사용할 수 있습니다.

IPv6 주소

IPv6 주소는 전역적으로 고유하며 프라이빗으로 유지되거나 인터넷으로 접속하도록 구성할 수 있습니다. AWS에서는 퍼블릭 및 프라이빗 IPv6 주소를 모두 사용할 수 있습니다:

- 프라이빗 IPv6: AWS는 AWS에서 인터넷을 통해 광고되지 않고 광고할 수 없는 주소를 프라이빗 IPv6 주소로 간주합니다.
- 퍼블릭 IPv6: AWS는 AWS에서 인터넷에 광고하는 주소를 퍼블릭 IPv6 주소로 간주합니다.

퍼블릭 및 프라이빗 IPv6 주소에 대한 자세한 내용은 Amazon VPC 사용 설명서의 IPv6 주소를 참조하세요.

C1, M1, M2, M3, T1을 제외한 모든 인스턴스 유형은 IPv6 주소를 지원합니다.

IPv6 CIDR 블록이 VPC 및 서브넷과 연결되어 있고 다음 중 하나에 해당하는 경우 EC2 인스턴스는 IPv6 주소를 받습니다:

- 서브넷은 시작 중인 인스턴스에 IPv6 주소를 자동으로 할당하도록 구성됩니다. 자세한 내용은 서브넷의 IP 주소 지정 속성 수정을 참조하세요.
- 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 주 네트워크 인터페이스(index 0)와 연결됩니다. 다음 방법으로 인스턴스 기본 네트워크 인터페이스에 대한 IPv6 주소를 관리할 수 있습니다.

- 네트워크 인터페이스에서 IPv6 주소를 할당 및 할당 해제합니다. 네트워크 인터페이스에 할당할 수 있는 IPv6 주소의 개수, 그리고 인스턴스에 연결할 수 있는 네트워크 인터페이스의 개수는 인스턴스 유형에 따라 달라집니다. 자세한 내용은 네트워크 인터페이스당 최대 IP 주소 섹션을 참조하세요.
- 기본 IPv6 주소를 활성화합니다. 기본 IPv6 주소를 사용하면 인스턴스나 ENI에 대한 트래픽 중단을 방지할 수 있습니다. 자세한 내용은 EC2 인스턴스용 네트워크 인터페이스 생성 또는 네트워크 인터페이스의 IP 주소 관리을 참조하세요.

인스턴스를 중지했다가 시작할 때 또는 최대 절전 모드로 전환했다가 시작할 때에는 IPv6 주소가 지속되다가 인스턴스를 종료하면 릴리스됩니다. IPv6 주소는 다른 네트워크 인터페이스에 할당되는 동안에는 재할당할 수 없으므로 먼저 할당을 해제해야 합니다.

서브넷에 대한 라우팅을 제어하거나 보안 그룹 및 네트워크 ACL 규칙을 사용함으로써 인스턴스가 IPv6 주소를 통해 접속이 가능하도록 할지 여부를 제어할 수 있습니다. 자세한 내용은 Amazon VPC 사용 설명서의 인터네트워크 트래픽 개인 정보 보호를 참조하세요.

예약된 IPv6 주소 범위에 대한 자세한 정보는 IANA IPv6 Special-Purpose Address Registry 및 RFC4291을 참조하세요.

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 네트워크 인터페이스당 최대 IP 주소 섹션을 참조하세요.

사용 사례

- 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.

다중 IP 주소 동작 방법

- 사용자는 모든 네트워크 인터페이스에 보조 프라이빗 IPv4 주소를 할당할 수 있습니다.
- 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.

자세한 내용은 the section called “보조 IP 주소” 섹션을 참조하세요.

EC2 인스턴스 호스트 이름

EC2 인스턴스를 생성할 때 AWS는 해당 인스턴스에 대한 호스트 이름을 만듭니다. 호스트 이름 유형 및 AWS가 프로비저닝하는 방법에 대한 자세한 내용은 EC2 인스턴스 호스트 이름 및 도메인 단원을 참조하세요. Amazon은 Amazon이 제공한 호스트 이름을 IPv4 및 IPv6 주소로 변환하는 DNS 서버를 제공합니다. Amazon DNS 서버는 사용자 VPC 네트워크 범위 +2의 범위에 위치합니다. 자세한 내용을 알아보려면 Amazon VPC 사용 설명서의 VPC에 대한 DNS 속성을 참조하세요.

링크-로컬 주소

링크-로컬 주소는 잘 알려진 라우팅할 수 없는 IP 주소입니다. Amazon EC2는 링크-로컬 주소 공간의 주소를 사용하여 EC2 인스턴스에서만 액세스할 수 있는 서비스를 제공합니다. 이러한 서비스는 인스

턴스가 아닌 기본 호스트에서 실행됩니다. 이러한 서비스의 링크-로컬 주소에 액세스하면 Xen 하이퍼바이저 또는 Nitro 컨트롤러와 통신합니다.

링크-로컬 주소 범위

- IPv4 – 169.254.0.0/16(169.254.0.0~169.254.255.255)
- IPv6 – fe80::/10

링크-로컬 주소를 사용하여 액세스하는 서비스

- 인스턴스 메타데이터 서비스
- Amazon Route 53 Resolver(Amazon DNS 서버라고도 함)
- Amazon Time Sync Service
- AWS KMS 서버

EC2 인스턴스에 대한 IPv4 주소 관리

인스턴스를 시작할 때 퍼블릭 IPv4 주소를 인스턴스에 할당할 수 있습니다. 인스턴스(Instances) 페이지 또는 네트워크 인터페이스(Network Interfaces) 페이지를 통해 콘솔에서 인스턴스의 IPv4 주소를 볼 수 있습니다.

업무

- 시작 시 퍼블릭 IPv4 주소 할당
- 시작 시 프라이빗 IPv4 주소 할당
- 기본 IPv4 주소 보기
- 인스턴스 메타데이터를 사용하여 IPv4 주소 보기

시작 시 퍼블릭 IPv4 주소 할당

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 0인 네트워크 인터페이스에 할당됩니다. 이 기능은 인스턴스 시작 시점의 특정 조건에 따라 달라집니다.

고려 사항

- 네트워크 인터페이스와 연결된 IP 주소를 관리하여 시작 후 인스턴스에서 퍼블릭 IP 주소 할당을 취소할 수 있습니다. 퍼블릭 IPv4 주소에 대한 자세한 내용은 퍼블릭 IPv4 주소 섹션을 참조하세요.
- 네트워크 인터페이스를 2개 이상 지정하면 퍼블릭 IP 주소를 자동 할당할 수 없습니다. 또한 디바이스 색인 0에 대해 기존 네트워크 인터페이스를 지정하면 퍼블릭 IP 자동 할당 기능을 사용하여 서브넷 설정을 재정의할 수 없습니다.
- 시작 도중에 퍼블릭 IP 주소를 인스턴스에 할당하는지의 여부에 관계없이 시작 후에는 인스턴스와 탄력적 IP 주소를 연결할 수 있습니다. 자세한 내용은 탄력적 IP 주소 섹션을 참조하세요. 또한, 사용자는 서브넷의 퍼블릭 IPv4 주소 지정 동작을 변경할 수 있습니다. 자세한 내용은 서브넷의 퍼블릭 IPv4 주소 지정 속성 수정을 참조하세요.

Console

시작 시 퍼블릭 IPv4 주소를 할당하는 방법

절차에 따라 인스턴스를 시작하고 Network Settings(네트워크 설정)를 구성할 때 Auto-assign Public IP(퍼블릭 IP 자동 할당) 옵션을 선택합니다.

AWS CLI

시작 시 퍼블릭 IPv4 주소를 할당하는 방법

run-instances 명령을 `--associate-public-ip-address` 옵션과 함께 사용합니다.

```
--associate-public-ip-address
```

PowerShell

시작 시 퍼블릭 IPv4 주소를 할당하는 방법

`-AssociatePublicIp` 파라미터와 함께 New-EC2Instance cmdlet을 사용합니다.

```
-AssociatePublicIp $true
```

시작 시 프라이빗 IPv4 주소 할당

서브넷의 IPv4 주소 범위 중에서 IPv4 주소를 지정하거나 Amazon EC2가 선택하도록 할 수 있습니다. 이 주소는 기본 네트워크 인터페이스에 할당됩니다.

시작 시 IPv4 주소를 할당하는 방법은 the section called “인스턴스에 보조 IP 주소 할당” 단원을 참조하세요.

Console

시작 시 프라이빗 IPv4 주소를 할당하는 방법

인스턴스 시작 절차를 따릅니다. 네트워크 설정을 구성할 때 고급 네트워크 구성을 확장하고 기본 IP에 대한 값을 입력합니다.

AWS CLI

시작 시 프라이빗 IPv4 주소를 할당하는 방법

run-instances 명령을 `--private-ip-address` 옵션과 함께 사용합니다.

```
--private-ip-addresses 10.251.50.12
```

Amazon EC2가 IP 주소를 선택하도록 하려면 이 옵션을 생략합니다.

PowerShell

시작 시 프라이빗 IPv4 주소를 할당하는 방법

`-PrivateIpAddress` 파라미터와 함께 New-EC2Instance cmdlet을 사용합니다.

```
-PrivateIpAddress 10.251.50.12
```

Amazon EC2가 IP 주소를 선택하도록 하려면 이 파라미터를 생략합니다.

기본 IPv4 주소 보기

퍼블릭 IPv4 주소는 콘솔에서 네트워크 인터페이스의 속성으로 표시되지만 NAT를 통해 주 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.

Console

인스턴스에 대한 IPv4 주소를 보는 방법

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

2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다.
4. 네트워킹 탭에서 퍼블릭 IPv4 주소와 프라이빗 IPv4 주소를 찾습니다.
5. (선택 사항) 네트워킹 탭에는 네트워크 인터페이스와 인스턴스에 대한 탄력적 IP 주소도 있습니다.

AWS CLI

인스턴스에 대한 기본 IPv4 주소를 보는 방법

아래와 같이 describe-instances 명령을 사용합니다.

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query "Reservations[].Instances[].PrivateIpAddress" \
    --output text
```

출력의 예시는 다음과 같습니다.

```
10.251.50.12
```

PowerShell

인스턴스에 대한 기본 IPv4 주소를 보는 방법

Get-EC2Instance cmdlet을 사용합니다.

```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances.PrivateIpAddress
```

출력의 예시는 다음과 같습니다.

```
10.251.50.12
```

인스턴스 메타데이터를 사용하여 IPv4 주소 보기

인스턴스 메타데이터를 검색하여 인스턴스에 대한 IPv4 주소를 가져올 수 있습니다. 자세한 내용은 인스턴스 메타데이터를 사용하여 EC2 인스턴스를 관리합니다. 섹션을 참조하세요.

인스턴스 메타데이터를 사용하여 IPv4 주소를 보는 방법

1. 인스턴스에 연결합니다. 자세한 내용은 EC2 인스턴스에 연결 섹션을 참조하세요.
2. 다음 명령 중 하나를 실행합니다.

IMDSv2

Linux

Linux 인스턴스에서 다음 명령을 실행합니다.

```
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" http://169.254.169.254/latest/
meta-data/local-ipv4
```

Windows

Windows 인스턴스에서 다음 명령을 실행합니다.

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/local-ipv4
```

IMDSv1

Linux

Linux 인스턴스에서 다음 명령을 실행합니다.

```
curl http://169.254.169.254/latest/meta-data/local-ipv4
```

Windows

Windows 인스턴스에서 다음 명령을 실행합니다.

```
Invoke-RestMethod http://169.254.169.254/latest/meta-data/local-ipv4
```

3. 다음과 같은 명령 중 하나를 사용하여 퍼블릭 IP 주소에 액세스합니다. 인스턴스와 연결된 탄력적 IP 주소가 있으면 명령을 통해 탄력적 IP 주소가 반환됩니다.

   IMDSv2

   Linux

   Linux 인스턴스에서 다음 명령을 실행합니다.

   ```
   [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" http://169.254.169.254/latest/
   meta-data/public-ipv4
   ```

   Windows

   Windows 인스턴스에서 다음 명령을 실행합니다.

   ```
   [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
   seconds" = "21600"} `
       -Method PUT -Uri http://169.254.169.254/latest/api/token
   ```

   ```
   Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} `
       -Method GET -Uri http://169.254.169.254/latest/meta-data/public-ipv4
   ```

   IMDSv1

   Linux

   Linux 인스턴스에서 다음 명령을 실행합니다.

   ```
   curl http://169.254.169.254/latest/meta-data/public-ipv4
   ```

   Windows

   Windows 인스턴스에서 다음 명령을 실행합니다.

   ```
   Invoke-RestMethod http://169.254.169.254/latest/meta-data/public-ipv4
   ```

EC2 인스턴스에 대한 IPv6 주소 관리

VPC와 서브넷에 연결된 IPv6 CIDR 블록이 있는 경우, 시작 중 또는 시작 후 인스턴스에 IPv6 주소를 할당할 수 있습니다. 인스턴스 페이지 또는 네트워크 인터페이스 페이지를 통해 콘솔에서 인스턴스의 IPv6 주소를 볼 수 있습니다.

업무

- 인스턴스에 IPv6 주소 할당
- 인스턴스에 대한 IPv6 주소 보기:
- 인스턴스 메타데이터를 사용하여 IPv6 주소 보기
- 인스턴스에서 IPv6 주소 할당 해제

인스턴스에 IPv6 주소 할당

서브넷의 IPv6 주소 범위 중에서 IPv6 주소를 지정하거나 Amazon EC2가 선택하도록 할 수 있습니다. 이 주소는 기본 네트워크 인터페이스에 할당됩니다. C1, M1, M2, M3, T1 인스턴스 유형은 IPv6 주소를 지원하지 않습니다.

Console

시작 시 IPv6 주소를 할당하는 방법

인스턴스 시작 절차를 따릅니다. 네트워크 설정을 구성할 때 IPv6 IP 자동 할당 옵션을 선택합니다. 이 옵션이 표시되지 않으면 선택한 서브넷과 연결된 IPv6 CIDR 블록이 없는 것입니다.

시작 후에 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 차례로 선택합니다.
4. 네트워크 인터페이스를 확장합니다. IPv6 주소에서 새 IP 주소 할당을 선택합니다.
5. 서브넷 범위 중에서 IPv6 주소를 입력하거나 필드를 비워 두어 Amazon EC2가 IPv6 주소를 선택하도록 합니다. 이 옵션이 표시되지 않으면 인스턴스 서브넷과 연결된 IPv6 CIDR 블록이 없는 것입니다.
6. 저장을 선택합니다.

AWS CLI

시작 시 IPv6 주소를 할당하는 방법

run-instances 명령을 `--ipv6-addresses` 옵션과 함께 사용합니다. 다음 예제에서는 2개의 IPv6 주소를 할당합니다.

```
--ipv6-addresses Ipv6Address=2001:db8::1234:5678:1.2.3.4
 Ipv6Address=2001:db8::1234:5678:5.6.7.8
```

Amazon EC2가 IPv6 주소를 선택하도록 하려면 대신 `--ipv6-address-count` 옵션을 사용합니다. 다음 예제에서는 2개의 IPv6 주소를 할당합니다.

```
--ipv6-address-count 2
```

시작 후에 IPv6 주소를 할당하려면

assign-ipv6-addresses 명령을 사용합니다. 다음 예제에서는 2개의 IPv6 주소를 할당합니다.

```
aws ec2 assign-ipv6-addresses \
    --network-interface-id eni-1234567890abcdef0 \
    --ipv6-addresses 2001:db8::1234:5678:1.2.3.4 2001:db8::1234:5678:5.6.7.8
```

Amazon EC2가 IPv6 주소를 선택하도록 하려면 대신 `--ipv6-address-count` 옵션을 사용합니다. 다음 예제에서는 2개의 IPv6 주소를 할당합니다.

```
aws ec2 assign-ipv6-addresses \
    --network-interface-id eni-1234567890abcdef0 \
    --ipv6-address-count 2
```

PowerShell

시작 시 IPv6 주소를 할당하는 방법

`-Ipv6Address` 파라미터와 함께 New-EC2Instance cmdlet을 사용합니다. 다음 예제에서는 2개의 IPv6 주소를 할당합니다.

```
-Ipv6Address $ipv6addr1,$ipv6addr2
```

다음과 같이 IPv6 주소를 정의합니다.

```
$ipv6addr1 = New-Object Amazon.EC2.Model.InstanceIpv6Address
$ipv6addr1.Ipv6Address = "2001:db8::1234:5678:1.2.3.4"
$ipv6addr2 = New-Object Amazon.EC2.Model.InstanceIpv6Address
$ipv6addr2.Ipv6Address = "2001:db8::1234:5678:5.6.7.8"
```

Amazon EC2가 IPv6 주소를 선택하도록 하려면 대신 `-Ipv6AddressCount` 파라미터를 사용합니다. 다음 예제에서는 2개의 IPv6 주소를 할당합니다.

```
-Ipv6AddressCount 2
```

시작 후에 IPv6 주소를 할당하려면

Register-EC2Ipv6AddressList cmdlet을 사용합니다. 다음 예제에서는 2개의 IPv6 주소를 할당합니다.

```
Register-EC2Ipv6AddressList `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Ipv6Address "2001:db8::1234:5678:1.2.3.4","2001:db8::1234:5678:5.6.7.8"
```

Amazon EC2가 IPv6 주소를 선택하도록 하려면 대신 `-Ipv6AddressCount` 파라미터를 사용합니다. 다음 예제에서는 2개의 IPv6 주소를 할당합니다.

```
Register-EC2Ipv6AddressList `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Ipv6AddressCount 2
```

인스턴스에 대한 IPv6 주소 보기:

인스턴스에 대한 IPv6 주소를 볼 수 있습니다.

Console

인스턴스에 대한 IPv6 주소를 보는 방법

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

4. 네트워킹 탭에서 IPv6 주소를 찾습니다.

AWS CLI

인스턴스에 대한 IPv6 주소를 보는 방법

아래와 같이 describe-instances 명령을 사용합니다.

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query "Reservations[*].Instances[].Ipv6Address" \
    --output text
```

출력의 예시는 다음과 같습니다.

```
2001:db8::1234:5678:1.2.3.4
```

PowerShell

인스턴스에 대한 IPv6 주소를 보는 방법

Get-EC2Instance cmdlet을 사용합니다.

```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances.Ipv6Address
```

출력의 예시는 다음과 같습니다.

```
2001:db8::1234:5678:1.2.3.4
```

인스턴스 메타데이터를 사용하여 IPv6 주소 보기

인스턴스에 연결한 후 인스턴스 메타데이터를 사용하여 IPv6 주소를 검색할 수 있습니다. 먼저, `http://169.254.169.254/latest/meta-data/network/interfaces/macs/`에서 인스턴스의 MAC 주소를 가져와야 합니다.

IMDSv2

Linux

Linux 인스턴스에서 다음 명령을 실행합니다.

```
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" http://169.254.169.254/latest/meta-data/network/interfaces/macs/mac-address/ipv6s
```

Windows

Windows 인스턴스에서 다음과 같은 cmdlet을 실행합니다.

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/network/interfaces/macs/mac-address/ipv6s
```

IMDSv1

Linux

Linux 인스턴스에서 다음 명령을 실행합니다.

```
curl http://169.254.169.254/latest/meta-data/network/interfaces/macs/mac-address/ipv6s
```

Windows

Windows 인스턴스에서 다음 cmdlet을 실행합니다.

```
Invoke-RestMethod -Uri http://169.254.169.254/latest/meta-data/network/interfaces/macs/mac-address/ipv6s
```

인스턴스에서 IPv6 주소 할당 해제

언제든지 인스턴스에서 IPv6 주소 할당을 해제할 수 있습니다.

Console

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 차례로 선택합니다.
4. 네트워크 인터페이스를 확장합니다. IPv6 주소에서 해당 IPv6 주소 옆의 할당 해제를 선택합니다.
5. 저장을 선택합니다.

AWS CLI

인스턴스에서 IPv6 주소 할당 해제

unassign-ipv6-addresses 명령을 사용합니다.

```
aws ec2 unassign-ipv6-addresses \
    --network-interface-id eni-1234567890abcdef0 \
    --ipv6-addresses 2001:db8::1234:5678:1.2.3.4
```

PowerShell

인스턴스에서 IPv6 주소 할당 해제

Unregister-EC2Ipv6AddressList cmdlet을 사용합니다.

```
Unregister-EC2Ipv6AddressList `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Ipv6Address 2001:db8::1234:5678:1.2.3.4
```

EC2 인스턴스에 대한 보조 IP 주소

네트워크 인터페이스에 할당된 첫 번째 IPv4 주소를 기본 IP 주소라고 합니다. 보조 IP 주소는 네트워크 인터페이스에 할당된 추가 IPv4 주소입니다. 자세한 내용은 the section called “다중 IP 주소” 섹션을 참조하세요.

인스턴스에 여러 프라이빗 IPv6 주소를 할당할 수도 있습니다. 자세한 내용은 the section called “IPv6 주소” 섹션을 참조하세요.

업무

- 인스턴스에 보조 IP 주소 할당
- 보조 IP 주소를 사용하도록 운영 체제 구성
- 인스턴스에서 보조 프라이빗 IP 주소 할당을 해제하는 방법

인스턴스에 보조 IP 주소 할당

인스턴스 시작 시 또는 인스턴스 실행 후 인스턴스에 대한 네트워크 인터페이스에 보조 IP 주소를 할당할 수 있습니다.

Console

시작 시 보조 IP 주소를 할당하는 방법

1. 인스턴스 시작 절차를 따릅니다. 네트워크 설정을 구성할 때 고급 네트워크 구성을 확장합니다.
2. 보조 IP에는 자동 할당을 선택하고 Amazon EC2에서 할당할 IP 주소 수를 입력합니다. 또는 수동 할당을 선택하고 IPv4를 입력합니다.
3. 나머지 단계를 완료하여 인스턴스를 시작합니다.

시작 후 보조 IP 주소를 할당하는 방법

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 차례로 선택합니다.
4. 네트워크 인터페이스를 확장합니다.
5. IPv4 주소를 추가하려면 IPv4 주소에서 새 IP 주소 할당을 선택합니다. 서브넷 범위 중에서 IPv4 주소를 입력하거나 필드를 비워 두어 Amazon EC2가 선택하도록 합니다.
6. 저장을 선택합니다.

AWS CLI

시작 시 보조 IP 주소를 할당하는 방법

run-instances 명령을 `--secondary-private-ip-addresses` 옵션과 함께 사용합니다.

```
--secondary-private-ip-addresses 10.251.50.12
```

Amazon EC2가 IP 주소를 선택하도록 하려면 대신 `--secondary-private-ip-address-count` 옵션을 사용합니다. 다음 예제에서는 1개의 보조 IP 주소를 할당합니다.

```
--secondary-private-ip-address-count 1
```

또는 네트워크 인터페이스를 생성할 수도 있습니다. 자세한 내용은 the section called “네트워크 인터페이스 생성” 섹션을 참조하세요.

시작 후 보조 IP 주소를 할당하는 방법

assign-private-ip-addresses 명령을 `--private-ip-addresses` 옵션과 함께 사용합니다.

```
aws ec2 assign-private-ip-addresses \
    --network-interface-ids eni-1234567890abcdef0 \
    --private-ip-addresses 10.251.50.12
```

Amazon EC2가 IPv4 주소를 선택하도록 하려면 대신 `--secondary-private-ip-address-count` 파라미터를 사용합니다. 다음 예제에서는 1개의 IPv4 주소를 할당합니다.

```
aws ec2 assign-private-ip-addresses \
    --network-interface-ids eni-1234567890abcdef0 \
    --secondary-private-ip-address-count 1
```

PowerShell

시작 시 보조 IP 주소를 할당하는 방법

네트워크 인터페이스를 생성해야 합니다. 자세한 내용은 the section called “네트워크 인터페이스 생성” 섹션을 참조하세요.

시작 후 보조 IP 주소를 할당하는 방법

`-PrivateIpAddress` 파라미터와 함께 Register-EC2PrivateIpAddress cmdlet을 사용합니다.

```
Register-EC2PrivateIpAddress `
    -NetworkInterfaceId eni-1234567890abcdef0 `
```

```
    -PrivateIpAddress 10.251.50.12
```

Amazon EC2가 IPv4 주소를 선택하도록 하려면 대신 `-SecondaryPrivateIpAddressCount` 파라미터를 사용합니다. 다음 예제에서는 1개의 IPv4 주소를 할당합니다.

```
Register-EC2PrivateIpAddress `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -SecondaryPrivateIpAddressCount 1
```

보조 IP 주소를 사용하도록 운영 체제 구성

인스턴스에 보조 프라이빗 IP 주소를 할당한 후 추가 프라이빗 IPv4 주소가 인식되도록 인스턴스의 운영 체제를 구성해야 합니다.

Linux 인스턴스

- 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 Configure your network interface using ec2-net-utils를 참조하세요.
- 다른 Linux 배포판을 사용하는 경우 해당 Linux 배포판에서 제공된 문서를 참조하세요. 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 인스턴스

자세한 내용은 Windows 인스턴스의 보조 프라이빗 IPv4 주소 구성 섹션을 참조하세요.

인스턴스에서 보조 프라이빗 IP 주소 할당을 해제하는 방법

보조 프라이빗 IP 주소가 더는 필요하지 않으면 인스턴스 또는 네트워크 인터페이스에서 할당을 해제할 수 있습니다. 보조 프라이빗 IPv4 주소가 네트워크 인터페이스에서 할당이 해제된 경우 탄력적 IP 주소(존재하는 경우)도 또한 연결이 해제됩니다.

Console

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. 네트워크 인터페이스를 확장합니다. IPv4 주소에서 할당을 해제할 IPv4 주소에 대해 할당 해제를 선택합니다.
5. 저장을 선택합니다.

AWS CLI

보조 프라이빗 IP 주소 할당 해제 방법

unassign-private-ip-addresses 명령을 사용합니다.

```
aws ec2 unassign-private-ip-addresses \
    --network-interface eni-1234567890abcdef0\
    --private-ip-addresses 10.251.50.12
```

PowerShell

보조 프라이빗 IP 주소 할당 해제 방법

Unregister-EC2PrivateIpAddress cmdlet을 사용합니다.

```
Unregister-EC2PrivateIpAddress `
    -NetworkInterface eni-1234567890abcdef0 `
    -PrivateIpAddress 10.251.50.12
```

Windows 인스턴스의 보조 프라이빗 IPv4 주소 구성

인스턴스에서 다중 프라이빗 IPv4 주소를 지정할 수 있습니다. 인스턴스에 보조 프라이빗 IPv4 주소를 배정한 후에는 보조 프라이빗 IPv4 주소를 인식하도록 인스턴스의 운영 체제를 구성해야 합니다.

> **Note**
>
> 이러한 지침은 Windows Server 2022에 기초합니다. 이러한 단계의 구현은 Windows 인스턴스의 운영 체제에 따라 다를 수 있습니다.

업무

- 사전 조건
- 1단계: 인스턴스에서 고정 IP 주소 지정 구성
- 2단계: 인스턴스에 대한 보조 프라이빗 IP 주소 구성
- 3단계: 보조 프라이빗 IP 주소를 사용하도록 애플리케이션 구성

사전 조건

1. 보조 프라이빗 IPv4 주소를 인스턴스의 네트워크 인터페이스에 할당합니다. 인스턴스 시작 시 또는 인스턴스가 실행된 다음 보조 프라이빗 IPv4 주소를 할당할 수 있습니다. 자세한 내용은 인스턴스에 보조 IP 주소 할당 섹션을 참조하세요.
2. 탄력적 IP 주소를 할당하고 보조 프라이빗 IPv4 주소에 연결합니다. 자세한 내용은 탄력적 IP 주소 할당 섹션을 참조하세요.

1단계: 인스턴스에서 고정 IP 주소 지정 구성

Windows 인스턴스에서 여러 IP 주소를 사용할 수 있도록 하려면 DHCP 보다는 고정 IP 주소 지정 방식을 사용하도록 인스턴스를 구성해야 합니다.

> **Important**
>
> 인스턴스에서 고정 IP 주소 지정을 구성하는 경우 IP 주소가 콘솔, CLI 또는 API에 표시된 것과 정확히 일치해야 합니다. 이러한 IP 주소를 잘못 입력할 경우 인스턴스가 연결하지 못할 수 있습니다.

Windows 인스턴스에서 고정 IP 주소 지정을 구성하려면

1. 인스턴스에 연결합니다.

2. 먼저 다음 단계를 수행하여 인스턴스에 대한 IP 주소, 서브넷 마스크 및 기본 게이트웨이 주소를 찾습니다.

   - PowerShell에서 다음 명령을 실행합니다.

     ```
     ipconfig /all
     ```

     출력을 검토하고 네트워크 인터페이스에 대한 IPv4 주소, 서브넷 마스크, 기본 게이트웨이, DNS 서버 값을 기록해 둡니다. 출력은 다음 예제와 비슷합니다.

     ```
     ...

     Ethernet adapter Ethernet 4:

        Connection-specific DNS Suffix  . : us-west-2.compute.internal
        Description . . . . . . . . . . . : Amazon Elastic Network Adapter #2
        Physical Address. . . . . . . . . : 02-9C-3B-FC-8E-67
        DHCP Enabled. . . . . . . . . . . : Yes
        Autoconfiguration Enabled . . . . : Yes
        Link-local IPv6 Address . . . . . : fe80::f4d1:a773:5afa:cd1%7(Preferred)
        IPv4 Address. . . . . . . . . . . : 10.200.0.128(Preferred)
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Lease Obtained. . . . . . . . . . : Monday, April 8, 2024 12:19:29 PM
        Lease Expires . . . . . . . . . . : Monday, April 8, 2024 4:49:30 PM
        Default Gateway . . . . . . . . . : 10.200.0.1
        DHCP Server . . . . . . . . . . . : 10.200.0.1
        DHCPv6 IAID . . . . . . . . . . . : 151166011
        DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2D-67-AC-FC-12-34-9A-BE-A5-
     E7
        DNS Servers . . . . . . . . . . . : 10.200.0.2
        NetBIOS over Tcpip. . . . . . . . : Enabled
     ```

3. PowerShell에서 다음 명령을 실행하여 네트워킹 및 공유 센터를 엽니다.

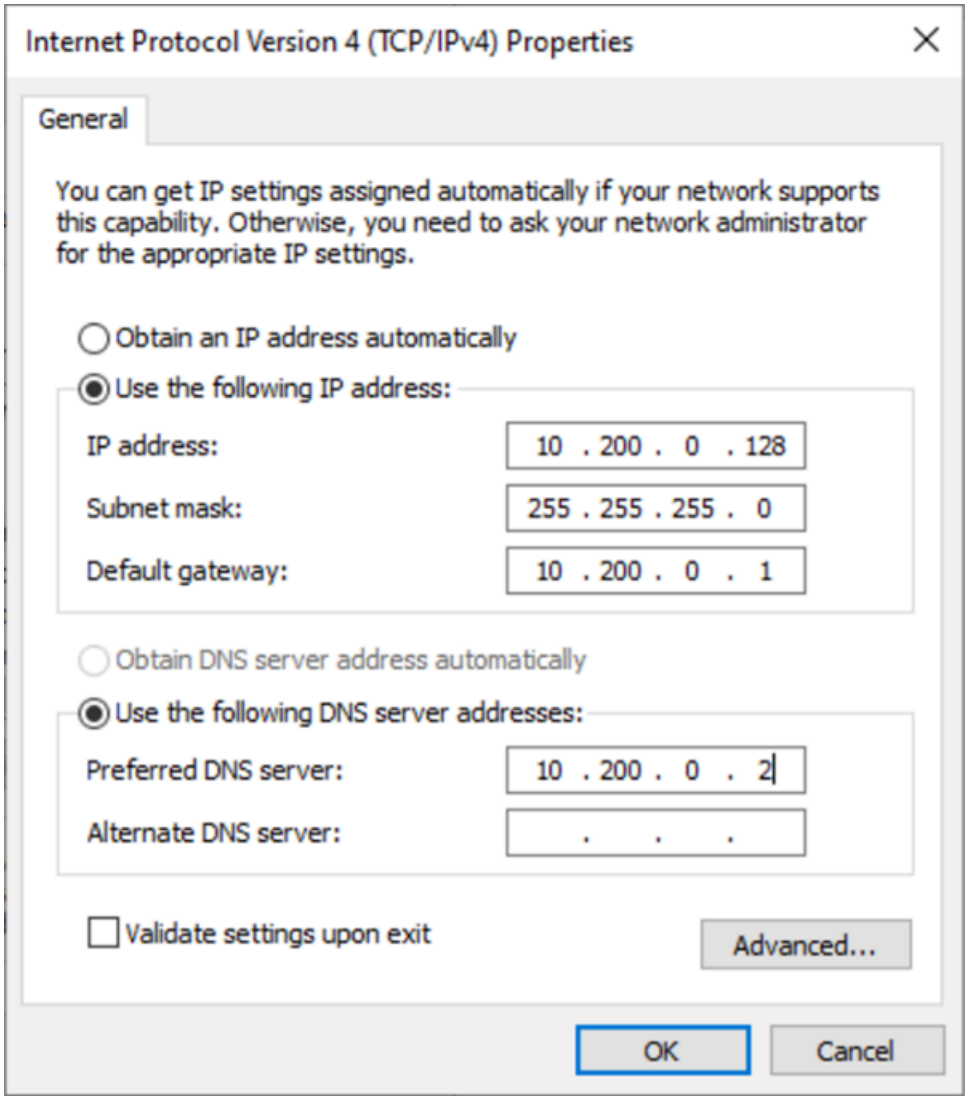
   ```
   & $env:SystemRoot\system32\control.exe ncpa.cpl
   ```

4. 네트워크 인터페이스(로컬 영역 연결 또는 이더넷)에 대한 컨텍스트 메뉴를 열고(마우스 오른쪽 버튼 클릭) 속성을 선택합니다.
5. 인터넷 프로토콜 버전 4(TCP/IPv4)를 선택하고 속성을 클릭합니다.
6. 인터넷 프로토콜 버전 4(TCP/IPv4) 속성 대화 상자에서 다음 IP 주소 사용을 선택하고 다음 값을 입력한 다음 확인을 클릭합니다.

필드	값
IP 주소	위의 2단계에서 얻은 IPv4 주소입니다.
서브넷 마스크	위의 2단계에서 얻은 서브넷 마스크입니다.
기본 게이트웨이	위의 2단계에서 얻은 기본 게이트웨이 주소입니다.
기본 설정 DNS 서버	위의 2단계에서 얻은 DNS 서버입니다.
보조 DNS 서버	위의 2단계에서 얻은 보조 DNS 서버입니다. 보조 DNS 서버가 나열되지 않은 경우 이 필드를 비워 둡니다.

Important

IP 주소를 현재 IP 주소가 아닌 다른 값으로 설정할 경우 인스턴스와의 연결이 끊어집니다.



인스턴스가 DHCP 사용을 고정 주소 지정으로 변환하는 몇 초 동안은 Windows 인스턴스와 RDP 간의 연결이 끊어집니다. 인스턴스가 전과 동일한 IP 주소 정보를 유지하지만, 지금은 이 정보가 고정 정보이며 DHCP에 의해 관리되지 않습니다.

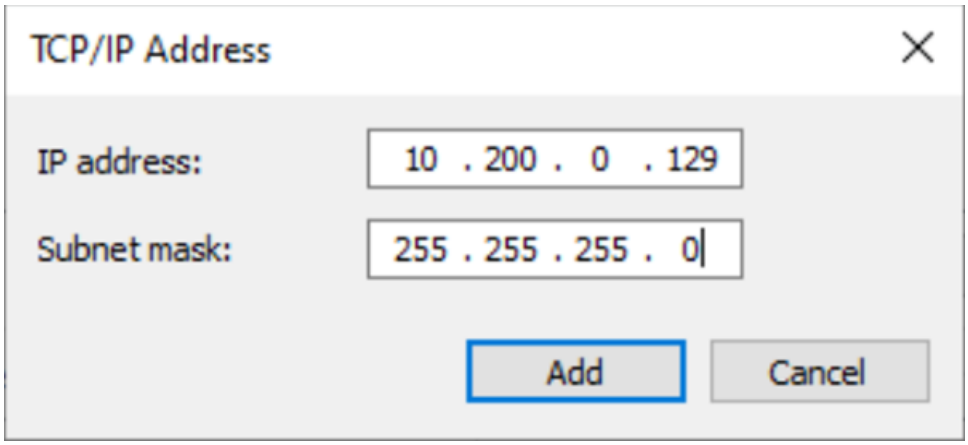
2단계: 인스턴스에 대한 보조 프라이빗 IP 주소 구성

Windows 인스턴스에서 고정 IP 주소 지정을 설정하고 나면 두 번째 프라이빗 IP 주소를 준비할 수 있습니다.

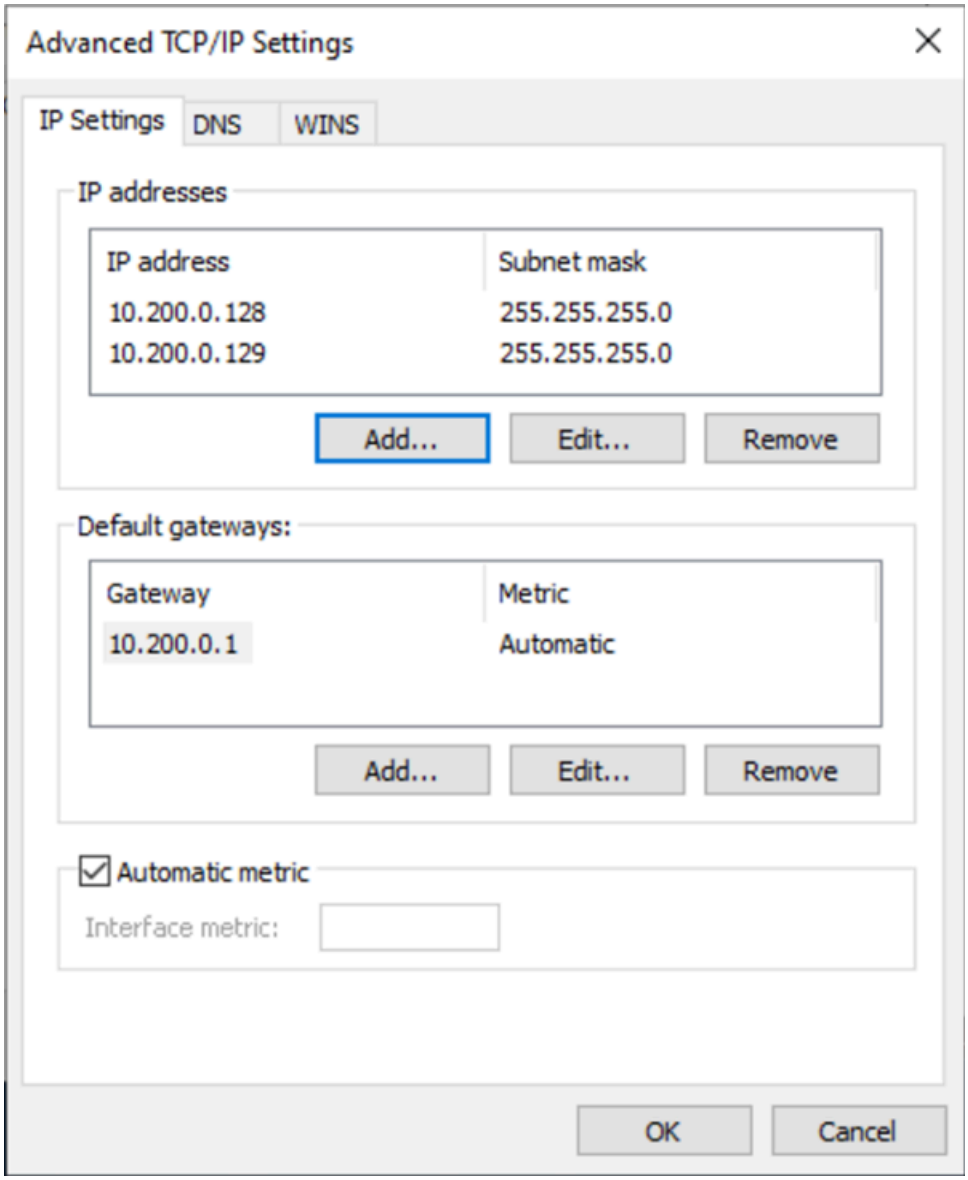
보조 IP 주소를 구성하려면

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

2. 탐색 창에서 [인스턴스(Instances)]를 선택하고 인스턴스를 선택합니다.
3. [네트워킹(Networking)]에서 보조 IP 주소를 기록합니다.
4. 인스턴스에 연결합니다.
5. Windows 인스턴스에서 시작, 제어판을 선택합니다.
6. 네트워크 및 인터넷, 네트워크 및 공유 센터를 선택합니다.
7. 네트워크 인터페이스(로컬 영역 연결 또는 이더넷)를 선택한 다음, 속성을 선택합니다.
8. 로컬 영역 연결 속성 페이지에서 인터넷 프로토콜 버전 4(TCP/IPv4), 속성, 고급을 선택합니다.
9. [추가]를 선택합니다.
10. TCP/IP 주소 대화 상자의 IP 주소에 보조 프라이빗 IP 주소를 입력합니다. 서브넷 마스크 필드에 1단계: 인스턴스에서 고정 IP 주소 지정 구성의 주 프라이빗 IP 주소로 입력한 것과 동일한 서브넷 마스크를 입력하고 추가를 선택합니다.



11. IP 주소 설정을 확인하고 확인을 선택합니다.



12. 확인, 닫기를 선택합니다.
13. 보조 IP 주소가 운영 체제에 추가되었는지 확인하려면 PowerShell에서 `ipconfig /all` 명령을 실행합니다. 출력은 다음과 같을 것입니다.

```
Ethernet adapter Ethernet 4:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Amazon Elastic Network Adapter #2
   Physical Address. . . . . . . . . : 02-9C-3B-FC-8E-67
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::f4d1:a773:5afa:cd1%7(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.200.0.128(Preferred)
```

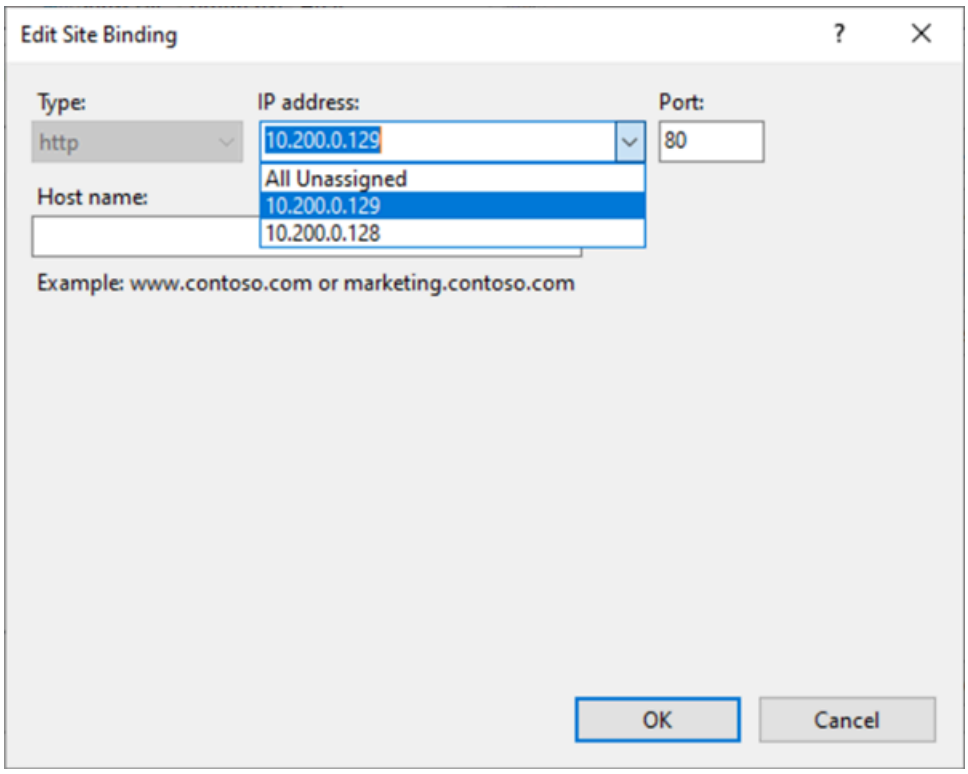
```
Subnet Mask . . . . . . . . . . . : 255.255.255.0
IPv4 Address. . . . . . . . . . . : 10.200.0.129(Preferred)
Subnet Mask . . . . . . . . . . . : 255.255.255.0
Default Gateway . . . . . . . . . : 10.200.0.1
DHCPv6 IAID . . . . . . . . . . . : 151166011
DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2D-67-AC-FC-12-34-9A-BE-A5-E7
DNS Servers . . . . . . . . . . . : 10.200.0.2
NetBIOS over Tcpip. . . . . . . . : Enabled
```

3단계: 보조 프라이빗 IP 주소를 사용하도록 애플리케이션 구성

부 프라이빗 IP 주소를 사용하도록 애플리케이션을 구성할 수 있습니다. 예를 들어, 인스턴스에서 IIS의 웹 사이트를 실행 중인 경우 부 프라이빗 IP 주소를 사용하도록 IIS를 구성할 수 있습니다.

부 프라이빗 IP 주소를 사용하도록 IIS를 구성하려면

1. 인스턴스에 연결합니다.
2. IIS(인터넷 정보 서비스) 관리자를 엽니다.
3. 연결 창에서 사이트를 확장합니다.
4. 웹 사이트에 대한 컨텍스트 메뉴를 열고(마우스 오른쪽 버튼 클릭) 바인딩 편집을 선택합니다.
5. 사이트 바인딩 대화 상자의 유형에서 http, 편집을 선택합니다.
6. 사이트 바인딩 편집 대화 상자의 IP 주소에서 보조 프라이빗 IP 주소를 선택합니다. 기본적으로 각 웹 사이트는 모든 IP 주소의 HTTP 요청을 허용합니다.



7. 확인, 닫기를 선택합니다.

EC2 인스턴스 호스트 이름 및 도메인

Amazon EC2 인스턴스를 효과적으로 관리하고 액세스하려면 Amazon EC2 인스턴스 호스트 이름과 도메인을 이해하는 것이 중요합니다. 각 EC2 인스턴스는 용도가 서로 다르고 특정 이름 지정 규칙을 따르는 서로 다른 호스트 이름 유형(프라이빗 및 퍼블릭)을 가질 수 있습니다.

이 주제에서는 EC2 인스턴스 호스트 이름의 구조, 구성 방법, 사용 가능한 서로 다른 호스트 이름 유형에 대해 설명합니다. 호스트 이름 설정을 보고 수정하는 방법을 알아보고, 각 유형을 사용해야 하는 경우를 이해하고, AWS 환경에서 호스트 이름을 관리하는 모범 사례를 알아봅니다.

내용

- EC2 인스턴스 호스트 이름 및 도메인 이해
- 호스트 이름 유형

EC2 인스턴스 호스트 이름 및 도메인 이해

EC2 인스턴스 주소는 다양한 구성 요소로 구성됩니다. 다음은 인스턴스의 프라이빗 IPv4 주소를 사용하는 EC2 인스턴스 주소의 예입니다.

```
   IP address         Domain name
   #--------# #-----------------------#
ip-10-24-34-0.us-west-2.compute.internal
#-----------#
  Hostname
#--------------------------------------#
    Fully qualified domain name (FQDN)
```

위치:

- IP 주소: 인스턴스에 연결된 기본 네트워크 인터페이스의 기본 IPv4 주소입니다.
- 호스트 이름: 특정 EC2 인스턴스의 로컬 이름으로, 운영 체제와 로컬 네트워크 식별에 사용됩니다.
- 도메인 이름: FQDN 중에서 AWS가 제공하는 부분입니다.
- 정규화된 도메인 이름(FQDN): 호스트 이름과 도메인 이름을 모두 포함하는 전체 주소입니다. FQDN은 네트워크에서 인스턴스에 연결하는 데 사용되는 전역적으로 고유한 전체 식별자입니다.

인스턴스 또는 인스턴스에 연결된 기본 네트워크 인터페이스에 선택한 호스트 이름 유형에 따라 호스트 이름 및 도메인 이름 형식이 위의 예제와 달라집니다. 이 섹션에서는 호스트 이름 유형 옵션을 설명합니다.

호스트 이름 유형

AWS는 프라이빗과 퍼블릭이라는 두 가지 유형의 호스트 이름을 제공합니다. 다음 표에서는 확인 방법, 구성 방법, 각 유형을 사용하는 경우 등 프라이빗 호스트 이름과 퍼블릭 호스트 이름의 주요 차이점을 비교합니다.

	프라이빗 호스트 이름	퍼블릭 호스트 이름
DNS 확인	프라이빗 호스트 이름을 사용하면 퍼블릭 인터넷에서 액세스할 수 없는 프라이빗 FQDN을 사용할 수 있습니다. 프라이빗 호스트 이름은 VPC 내의 프라이빗 IPv4 및 IPv6 GUA 주소로 확인되는 요청만 허용합니다.	퍼블릭 호스트 이름을 사용하면 퍼블릭 인터넷에서 액세스할 수 있는 퍼블릭 FQDN을 사용할 수 있습니다. 퍼블릭 호스트 이름을 사용하면 요청을 VPC 내의 프라이빗 IPv4 및 IPv6 GUA와 인터넷의 퍼블릭 IP로 확인할 수 있습니다(분할-수평 DNS).

	프라이빗 호스트 이름	퍼블릭 호스트 이름
구성	프라이빗 호스트 이름은 인스턴스 수준에서 구성됩니다.	퍼블릭 호스트 이름은 네트워크 인터페이스 수준에서 구성됩니다.
사용해야 하는 경우	• VPC 내에서만 확인되는 호스트 이름의 경우 내부 통신을 위해 • 내부 이름 지정 체계를 프라이빗으로 유지하기 위해 • 외부 DNS 고려 사항 없이 내부 리소스 관리를 간소화하기 위해	• 인터넷에서 IPv4 및 IPv6를 통해 인스턴스에 액세스하기 위해 • IPv4 환경에서 IPv6 환경으로 원활하게 마이그레이션하기 위해

내용

- 퍼블릭 호스트 이름
- 프라이빗 호스트 이름

퍼블릭 호스트 이름

퍼블릭 호스트 이름을 사용하면 인스턴스의 퍼블릭 IPv4 또는 IPv6 주소로 확인되고 IPv6로 쉽게 전환할 수 있는 호스트 이름을 사용하여 EC2 인스턴스에 액세스할 수 있습니다.

퍼블릭 호스트 이름을 사용하면 다음을 수행할 수 있습니다.

- IPv4 및 IPv6를 통해 EC2 인스턴스에 액세스하여 리소스에 더 유연하게 연결할 수 있습니다.
- 원하는 속도로 IPv4 환경에서 IPv6 환경으로 마이그레이션합니다. 예를 들어 데이터베이스 마이그레이션과 애플리케이션 마이그레이션을 분리하여 복잡성과 위험을 줄일 수 있습니다.
- 적절한 IP 주소로 자동 확인되는 여러 호스트 이름 옵션(IPv4 전용, IPv6 전용, 듀얼 스택)을 사용합니다.
- VPC 내에서 쿼리할 때 호스트 이름을 프라이빗 IP 주소로 확인하는 분할-수평 DNS를 통한 보안 향상의 이점을 누릴 수 있습니다.

내용

- 퍼블릭 호스트 이름 유형 및 사용 시기
- 퍼블릭 호스트 이름 보기

- 퍼블릭 호스트 이름 유형 수정

퍼블릭 호스트 이름 유형 및 사용 시기

퍼블릭 호스트 이름을 사용하려면 기존 네트워크 인터페이스를 수정해야 합니다. 이 섹션은 세 가지 퍼블릭 호스트 이름 유형 옵션을 설명하며, 사용할 옵션을 결정하는 데 도움이 됩니다.

- 듀얼 스택 - IP 기반 이름
  - IPv4에서 IPv6로 마이그레이션하거나 마이그레이션할 계획인 경우에 가장 좋은 옵션입니다. 이 옵션을 사용하면 IPv4를 통한 연결과 IPv6를 통한 연결이 모두 가능하므로 클라이언트가 두 프로토콜 중 하나를 유연하게 사용할 수 있고 IPv6로 마이그레이션하는 동안 클라이언트가 동일한 호스트 이름을 유지할 수 있습니다.
  - VPC 내에서의 요청은 네트워크 인터페이스의 프라이빗 IPv4 주소와 IPv6 글로벌 유니캐스트 주소(GUA)로 모두 확인됩니다. 인터넷에서의 요청은 네트워크 인터페이스의 퍼블릭 IPv4 주소와 IPv6 GUA 주소로 확인됩니다.
  - 예제
    - 이 옵션을 선택하면 이 네트워크 인터페이스의 듀얼 스택 FQDN이 생성됩니다. 다음은 생성되는 FQDN의 예입니다.
      - f5lnz-0khrm-nt2u3-gyqqt-nbdl5-q3cdpO.ap-southeast-2.ip.aws
    - 위치:
      - f5lnz-0khrm-nt2u3-gyqqt-nbdl5-q3cdpO는 네트워크 인터페이스 기본 퍼블릭 IPv6 주소의 base36 표현(f5lnz-0khrm-nt2u3-gyqt-nbdl5)과 함께 네트워크 인터페이스 기본 퍼블릭 IPv4 주소의 base36 표현(q3cdpO)이 있는 호스트 이름입니다.
      - f5lnz-0khrm-nt2u3-gyqqt-nbdl5는 Amazon DNS 리졸버에 의해 IPv6 GUA 주소 FFFF:1407:4:f000:81d:2689:1066:4489로 확인됩니다. 이 주소는 네트워크 인터페이스에 할당된 첫 번째 IPv6 GUA입니다.
      - q3cdpO는 IPv4 주소 52.54.55.56으로 확인됩니다. 이 주소는 기본 네트워크 인터페이스에 연결된 퍼블릭 IPv4 주소입니다.
      - ap-southeast-2는 네트워크 인터페이스가 있는 서브넷의 리전입니다.
      - ip.aws는 AWS가 제공하는 도메인입니다.
- IPv6 - IP 기반 이름
  - 이 옵션은 이미 IPv6로 마이그레이션했고 IPv6를 통해서만 연결해야 하는 경우에 적합합니다.
  - VPC 내부에서의 요청이나 인터넷에서의 요청은 네트워크 인터페이스의 IPv6 GUA로 확인됩니다.
  - 예제

    - 이 옵션을 선택하면 이 네트워크 인터페이스의 FQDN이 생성됩니다. 다음은 생성되는 FQDN의 예입니다.
        - f5lnz-0khrm-nt2u3-gyqqt-nbdl5.ap-southeast-2.ip.aws
    - 위치:
        - f5lnz-0khrm-nt2u3-gyqqt-nbdl5는 네트워크 인터페이스의 기본 퍼블릭 IPv6 주소를 base36으로 표현한 호스트 이름입니다.
        - f5lnz-0khrm-nt2u3-gyqqt-nbdl5는 Amazon DNS 리졸버에 의해 IPv6 GUA 주소 FFFF:1407:4:f000:81d:2689:1066:4489로 확인됩니다. 이 주소는 네트워크 인터페이스에 할당된 첫 번째 IPv6 GUA입니다.
        - ap-southeast-2는 네트워크 인터페이스가 있는 서브넷의 리전입니다.
        - ip.aws는 AWS가 제공하는 도메인입니다.
- IPv4 - IP 기반 이름
    - 이 옵션은 IPv6로 전환하는 동안 이 네트워크 인터페이스를 사용하는 인스턴스가 IPv4 액세스를 유지해야 하거나 인스턴스에서 실행되는 애플리케이션이나 시스템이 IPv4만 지원하는 경우에 적합합니다. 이 옵션은 IPv4 연결만 유지하면 되고 워크로드에 IPv6 지원이 필요하지 않은 경우에 가장 적합합니다. 예를 들어 IPv6로 마이그레이션하는 경우 일부 애플리케이션은 IPv4에 유지하고 일부 애플리케이션은 IPv6로 이동하도록 결정할 수 있습니다.
    - VPC 내에서의 요청은 네트워크 인터페이스의 프라이빗 기본 IPv4 주소로 확인됩니다. 인터넷에서의 요청은 네트워크 인터페이스의 퍼블릭 IPv4 주소로 확인됩니다.
    - 예제
        - 이 옵션을 선택하면 이 네트워크 인터페이스의 IPv4-기반 퍼블릭 호스트 이름이 생성됩니다. 다음은 생성되는 DNS 이름의 예입니다.
            - ec2-52-54-55-66.ap-southeast-2.compute.amazonaws.com
        - 위치:
            - ec2-52-54-55-66은 네트워크 인터페이스의 기본 퍼블릭 IPv4 주소를 base36으로 표현한 호스트 이름입니다.
            - ec2-52-54-55-66은 IPv4 주소 52.54.55.56으로 확인됩니다. 이 주소는 기본 네트워크 인터페이스에 연결된 퍼블릭 IPv4 주소입니다.
            - ap-southeast-2는 네트워크 인터페이스가 있는 서브넷의 리전입니다.
            - ip.aws는 AWS가 제공하는 도메인입니다.

> **Important**
>
> 위 예제에서 호스트 이름을 생성하는 데 IP 주소가 사용되는 것을 확인할 수 있습니다. 기본 프라이빗 IPv4 주소 또는 네트워크 인터페이스에 할당된 첫 번째 IPv6 GUA를 변경하면 IP 주소로 변환되는 호스트 이름 부분이 변경되고 이전에 생성된 퍼블릭 호스트 이름은 더 이상 유효하지 않습니다. 또한 기본 IPv4 퍼블릭 주소를 변경하면 Amazon EC2 사용 설명서의 인스턴스 메타데이터 서비스(IMDS)가 강제로 다운스트림에서 새로 고쳐지므로 EC2 인스턴스 IMDS 메타데이터가 자동으로 업데이트됩니다.

퍼블릭 호스트 이름 보기

네트워크 인터페이스가 있는 VPC에 EnableDnsHostnames와 EnableDnsSupport 둘 다 활성화되어 있지 않으면 호스트 이름 유형이 정의되거나 생성되지 않은 것입니다.

Console

인스턴스 또는 프라이머리 네트워크 인터페이스의 퍼블릭 호스트 이름을 볼 수 있습니다.

인스턴스의 호스트 이름 유형과 DNS 이름을 보려면 다음을 수행하세요.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스에 대한 확인란을 선택합니다.
4. 네트워크 탭의 호스트 이름 및 DNS에서 다음을 찾습니다.
   - 퍼블릭 호스트 이름 유형
   - 퍼블릭 DNS
   - IPv4 전용 - IP 기반 이름
   - IPv6 전용 - IP 기반 이름
   - 듀얼 스택 - IP 기반 이름

네트워크 인터페이스의 호스트 이름 유형과 DNS 이름을 보려면 다음을 수행하세요.

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

3. 검색 필드에 인스턴스의 ID를 입력합니다. 네트워크 인터페이스의 ID를 선택하여 세부 정보 페이지를 엽니다.
4. 호스트 이름 및 DNS에서 다음을 찾습니다.

   - 퍼블릭 호스트 이름 유형
   - 공개 DNS 이름
   - 퍼블릭 IPv4 DNS 이름
   - 퍼블릭 IPv6 DNS 이름
   - 퍼블릭 듀얼 스택 DNS 이름

AWS CLI

네트워크 인터페이스의 호스트 이름 유형과 DNS 이름을 보려면 다음을 수행하세요.

describe-network-interfaces 명령을 사용합니다.

```
aws ec2 describe-network-interfaces \
    --network-interface-id eni-1234567890abcdef0 \
    --query NetworkInterfaces[].PublicIpDnsNameOptions
```

출력의 예시는 다음과 같습니다. 호스트 이름 유형이 `public-dual-stack-dns-name`이므로 DNS 호스트 이름은 `PublicDualStackDnsName`과 동일합니다.

```
[
     {
          "DnsHostnameType": "public-dual-stack-dns-name",
          "PublicIpv4DnsName": "ec2-52-54-55-66.ap-
southeast-2.compute.amazonaws.com",
          "PublicIpv6DnsName": "f5lnz-0khrm-nt2u3-gyqqt-nbdl5.ap-
southeast-2.ip.aws",
          "PublicDualStackDnsName": "f5lnz-0khrm-nt2u3-gyqqt-nbdl5-q3cdp0.ap-
southeast-2.ip.aws"
     }
]
```

PowerShell

네트워크 인터페이스의 호스트 이름 유형과 DNS 이름을 보려면 다음을 수행하세요.

Get-EC2NetworkInterface cmdlet을 사용합니다.

```
(Get-EC2NetworkInterface `
    -NetworkInterfaceId eni-1234567890abcdef0).PublicIpDnsNameOptions
```

출력의 예시는 다음과 같습니다. 호스트 이름 유형이 public-dual-stack-dns-name이므로 DNS 호스트 이름은 PublicDualStackDnsName과 동일합니다.

```
DnsHostnameType        : public-dual-stack-dns-name
PublicDualStackDnsName : f5lnz-0khrm-nt2u3-gyqqt-nbdl5-q3cdp0.ap-southeast-2.ip.aws
PublicIpv4DnsName      : ec2-52-54-55-66.ap-southeast-2.compute.amazonaws.com
PublicIpv6DnsName      : f5lnz-0khrm-nt2u3-gyqqt-nbdl5.ap-southeast-2.ip.aws
```

퍼블릭 호스트 이름 유형 수정

퍼블릭 호스트 이름 유형 옵션은 네트워크 인터페이스에 연결된 IP 주소에 따라 다릅니다.

- 네트워크 인터페이스에 퍼블릭 IPv4 주소만 있는 경우 호스트 이름 유형은 IPv4 - IP 기반 이름이어야 합니다.
- 네트워크 인터페이스에 퍼블릭 IPv6 주소만 있는 경우 호스트 이름 유형은 IPv6 - IP 기반 이름이어야 합니다.
- 네트워크 인터페이스에 퍼블릭 IPv4 주소와 IPv6 주소가 모두 있는 경우 호스트 이름 유형은 듀얼 스택 - IP 기반 이름일 수 있습니다.

사전 조건

- 네트워크 인터페이스에 연결된 퍼블릭 IPv4 주소 또는 IPv6 주소가 있어야 합니다.
- 네트워크 인터페이스가 있는 VPC에서 EnableDnsHostnames와 EnableDnsSupport가 활성화되어 있어야 합니다. Amazon VPC 사용 설명서의 VPC에 대한 DNS 속성 보기 및 업데이트를 참조하세요.

Console

퍼블릭 호스트 이름 유형을 수정하려면 다음을 수행하세요.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.
3. 검색 필드에 인스턴스의 ID를 입력합니다. 네트워크 인터페이스의 확인란을 선택합니다.

또는 인스턴스 세부 정보 페이지에서 네트워킹 탭을 선택하고 디바이스 인덱스 0에 대한 네트워크 인터페이스의 ID를 선택합니다.

4. 작업, 퍼블릭 호스트 이름 유형 수정을 선택합니다.
5. 다음 옵션을 선택합니다.

   - 듀얼 스택 - IP 기반 이름: 네트워크 인터페이스의 듀얼 스택 퍼블릭 호스트 이름입니다. VPC 내에서의 요청은 네트워크 인터페이스의 프라이빗 IPv4 주소와 IPv6 글로벌 유니캐스트 주소로 모두 확인됩니다. 인터넷에서의 요청은 네트워크 인터페이스의 퍼블릭 IPv4 주소와 IPv6 GUA 주소로 확인됩니다.
   - IPv4 - IP 기반 이름: 네트워크 인터페이스의 IPv4 기반 퍼블릭 호스트 이름입니다. VPC 내에서의 요청은 네트워크 인터페이스의 프라이빗 기본 IPv4 주소로 확인됩니다. 인터넷에서의 요청은 네트워크 인터페이스의 퍼블릭 IPv4 주소로 확인됩니다.
   - IPv6 - IP 기반 이름: 네트워크 인터페이스의 IPv6 기반 퍼블릭 호스트 이름입니다. VPC 내부에서의 요청이나 인터넷에서의 요청은 네트워크 인터페이스의 IPv6 GUA로 확인됩니다.
6. 수정을 선택합니다.

AWS CLI

퍼블릭 호스트 이름 유형을 수정하려면 다음을 수행하세요.

modify-public-ip-dns-name-options 명령을 사용합니다.

```
aws ec2 modify-public-ip-dns-name-options \
    --network-interface-id eni-1234567890abcdef0 \
    --hostname-type public-dual-stack-dns-name
```

출력의 예시는 다음과 같습니다.

```
{
    "Successful": true
}
```

PowerShell

퍼블릭 호스트 이름 유형을 수정하려면 다음을 수행하세요.

Edit-EC2PublicIpDnsNameOption cmdlet을 사용합니다.

```
Edit-EC2PublicIpDnsNameOption `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -HostNameType public-dual-stack-dns-name
```

프라이빗 호스트 이름

이 섹션에서는 VPC 서브넷에서 인스턴스를 시작할 때 사용할 수 있는 Amazon EC2 인스턴스 프라이빗 호스트 이름에 대해 설명합니다.

프라이빗 호스트 이름으로 네트워크에서 EC2 인스턴스를 구별할 수 있습니다. 예를 들어 네트워크의 일부 또는 모든 인스턴스와 통신하기 위해 스크립트를 실행하려는 경우 인스턴스의 프라이빗 호스트 이름을 사용할 수 있습니다.

내용

- 프라이빗 호스트 이름 유형
- 리소스 이름 및 IP 이름을 찾을 수 있는 위치
- 리소스 이름과 IP 이름 중에서 선택
- Amazon EC2에 대한 리소스 기반 이름 지정 옵션 변경

프라이빗 호스트 이름 유형

EC2 인스턴스가 VPC에서 시작될 때 게스트 OS 호스트 이름에는 두 가지 프라이빗 호스트 이름 유형이 있습니다.

- IP 이름(IP name): 레거시 이름 지정 체계로 인스턴스를 시작할 때 인스턴스의 프라이빗 IPv4 주소가 인스턴스의 호스트 이름에 포함됩니다. EC2 인스턴스의 수명 동안 IP 이름이 존재합니다. 프라이빗 DNS 호스트 이름으로 사용될 경우 프라이빗 IPv4 주소(A 레코드)만 반환합니다.
- 리소스 이름(Resource name): 인스턴스를 시작할 때 EC2 인스턴스 ID는 인스턴스의 호스트 이름에 포함됩니다. EC2 인스턴스의 수명 동안 리소스 이름이 존재합니다. 프라이빗 DNS 호스트 이름으로 사용되는 경우 프라이빗 IPv4 주소(A 레코드) 또는 IPv6 글로벌 유니캐스트 주소(AAAA 레코드)를 모두 반환할 수 있습니다.

EC2 인스턴스 게스트 OS 호스트 이름 유형은 서브넷 설정에 따라 다릅니다.

- 인스턴스가 IPv4 전용 서브넷으로 시작되면 IP 이름 또는 리소스 이름 중 하나를 선택할 수 있습니다.

- 인스턴스가 듀얼 스택(IPv4 + IPv6) 서브넷으로 시작되면 IP 이름 또는 리소스 이름 중 하나를 선택할 수 있습니다.
- 인스턴스가 IPv6 전용 서브넷으로 시작되면 리소스 이름이 자동으로 사용됩니다.

내용

- IP 이름
- 리소스 이름
- IP 이름과 리소스 이름의 차이점

IP 이름

IP 이름(IP name)의 호스트 이름 유형(Hostname type)을 사용하여 EC2 인스턴스를 시작하면 게스트 OS 호스트 이름이 프라이빗 IPv4 주소를 사용하도록 구성됩니다.

- us-east-1의 인스턴스 형식: *private-ipv4-address*.ec2.internal
- 예시: *ip-10-24-34-0*.ec2.internal
- 다른 AWS 리전의 인스턴스의 형식: *private-ipv4-address.region*.compute.internal
- 예시: *ip-10-24-34-0.us-west-2*.compute.internal

리소스 이름

IPv6 전용 서브넷에서 EC2 인스턴스를 시작할 때 리소스 이름(Resource name)의 호스트 이름 유형(Hostname type)이 기본적으로 선택됩니다. IPv4 전용 또는 듀얼 스택(IPv4 + IPv6) 서브넷에서 인스턴스를 시작할 때 리소스 이름(Resource name)이 선택할 수 있는 옵션입니다. 인스턴스를 시작한 후에 호스트 이름 구성을 관리할 수 있습니다. 자세한 내용은 Amazon EC2에 대한 리소스 기반 이름 지정 옵션 변경 섹션을 참조하세요.

리소스 이름(Resource name)의 호스트 이름 유형(Hostname type)으로 EC2 인스턴스를 시작하면 게스트 OS 호스트 이름이 EC2 인스턴스 ID를 사용하도록 구성됩니다.

- us-east-1의 인스턴스 형식: *ec2-instance-id*.ec2.internal
- 예시: *i-0123456789abcdef*.ec2.internal
- 다른 AWS 리전의 인스턴스의 형식: *ec2-instance-id.region*.compute.internal
- 예시: *i-0123456789abcdef.us-west-2*.compute.internal

IP 이름과 리소스 이름의 차이점

IP 이름과 리소스 이름에 대한 DNS 쿼리가 공존하여 이전 버전과의 호환성을 보장하고 호스트 이름의 IP 기반 이름 지정에서 리소스 기반 이름 지정으로 마이그레이션할 수 있습니다. IP 이름 기반 프라이빗 DNS 호스트 이름의 경우 인스턴스에 대한 DNS A 레코드 쿼리가 응답할지 여부를 구성할 수 없습니다. DNS A 레코드 쿼리는 게스트 OS 호스트 이름 설정과 관계없이 항상 응답합니다. 반면 리소스 이름 기반 프라이빗 DNS 호스트 이름의 경우 인스턴스에 대한 DNS A 및/또는 DNS AAAA 쿼리에 응답할지 여부를 구성할 수 있습니다. 인스턴스를 시작하거나 서브넷을 수정할 때 응답 동작을 구성합니다. 자세한 내용은 Amazon EC2에 대한 리소스 기반 이름 지정 옵션 변경 섹션을 참조하세요.

리소스 이름 및 IP 이름을 찾을 수 있는 위치

Amazon EC2 콘솔에서 호스트 이름 유형, 리소스 이름 및 IP 이름을 확인할 수 있습니다.

내용

- EC2 인스턴스를 생성하는 곳
- 기존 EC2 인스턴스의 세부 정보를 볼 때

EC2 인스턴스를 생성하는 곳

EC2 인스턴스를 생성할 때 선택한 서브넷 유형에 따라 리소스 이름(Resource name)의 호스트 이름 유형(Hostname type)을 사용할 수 있거나, 선택되어 수정할 수 없을 수도 있습니다. 이 섹션에서는 호스트 이름 유형 리소스 이름과 IP 이름이 표시되는 시나리오를 설명합니다.

시나리오 1

마법사에서 EC2 인스턴스를 생성하고(콘솔에서 인스턴스 시작 마법사를 사용하여 EC2 인스턴스 시작 참조) 세부 정보를 구성할 때 IPv6 전용으로 구성한 서브넷을 선택합니다.

이 경우 리소스 이름(Resource name)의 호스트 이름 유형(Hostname type)이 자동으로 선택되며 수정할 수 없습니다. IP 이름 IPv4(A 레코드) DNS 요청 사용 설정(Enable IP name IPv4 (A record) DNS requests)의 DNS 호스트 이름(DNS Hostname) 옵션 및 리소스 기반 IPv4(A 레코드) DNS 요청 사용 설정(Enable resource-based IPv4 (A record) DNS requests)이 자동으로 선택 해제되고 수정할 수 없습니다. 리소스 기반 IPv6 (AAAA 레코드) DNS 요청 사용 설정(Enable resource-based IPv6 (AAAA record) DNS requests)이 기본적으로 선택되지만 수정할 수 있습니다. 이 옵션을 선택하면 리소스 이름에 대한 DNS 요청은 이 EC2 인스턴스의 IPv6 주소(AAAA 레코드)로 확인됩니다.

시나리오 2

마법사에서 EC2 인스턴스를 만들고(콘솔에서 인스턴스 시작 마법사를 사용하여 EC2 인스턴스 시작 참조) 세부 정보를 구성할 때 IPv4 CIDR 블록 또는 IPv4 및 IPv6 CIDR 블록('이중 스택')으로 구성된 서브넷을 선택합니다.

이 경우, IP 이름 IPv4(A 레코드) DNS 요청 사용 설정(Enable IP name IPv4 (A record) DNS requests)이 자동으로 선택되며, 변경할 수 없습니다. 즉, IP 이름에 대한 요청은 이 EC2 인스턴스의 IPv4 주소(A 레코드)로 확인됩니다.

이 옵션은 기본적으로 서브넷 구성을 사용하지만 서브넷 설정에 따라 이 인스턴스에 대한 옵션을 수정할 수 있습니다.

- 호스트 이름 유형(Hostname type): EC2 인스턴스의 게스트 OS 호스트 이름을 리소스 이름으로 지정할 것인지 아니면 IP 이름으로 지정할 것인지 결정합니다. 기본값은 IP 이름(IP name)입니다.
- 리소스 기반 IPV4(A 레코드) DNS 요청 활성화(Enable resource-based IPV4 (A record) DNS requests): 리소스 이름에 대한 요청이 이 EC2 인스턴스의 프라이빗 IPv4 주소(A 레코드)로 확인되는지 여부를 결정합니다. 이 옵션은 기본적으로 설정되어 있지 않습니다.
- 리소스 기반 IPv6(AAAA 레코드) DNS 요청 활성화(Enable resource-based IPv6 (AAAA record) DNS requests): 리소스 이름에 대한 요청이 이 EC2 인스턴스의 IPv6 주소(AAAA 레코드)로 확인되는지 여부를 결정합니다. 이 옵션은 기본적으로 설정되어 있지 않습니다.

기존 EC2 인스턴스의 세부 정보를 볼 때

EC2 인스턴스에 대한 세부 정보(Details) 탭에서 기존 EC2 인스턴스의 호스트 이름 값을 확인할 수 있습니다.

- 호스트 이름 유형(Hostname type): IP 이름 또는 리소스 이름 형식의 호스트 이름입니다.
- 프라이빗 IP DNS 이름(IPv4에만 해당)(Private IP DNS name (IPv4 only)): 항상 인스턴스의 프라이빗 IPv4 주소로 확인되는 IP 이름입니다.
- 프라이빗 리소스 DNS 이름(Private resource DNS name): 이 인스턴스에 대해 선택한 DNS 레코드로 확인할 수 있는 리소스 이름입니다.
- 프라이빗 리소스 DNS 이름 응답(Answer private resource DNS name): 리소스 이름은 IPv4(A), IPv6(AAAA) 또는 IPv4 및 IPv6(A 및 AAAA) DNS 레코드로 확인됩니다.

또한 SSH를 통해 EC2 인스턴스에 직접 연결하고 `hostname` 명령을 입력하면 호스트 이름이 IP 이름 또는 리소스 이름 형식으로 표시됩니다.

리소스 이름과 IP 이름 중에서 선택

EC2 인스턴스를 시작할 때(콘솔에서 인스턴스 시작 마법사를 사용하여 EC2 인스턴스 시작 참조) 리소스 이름(Resource name)의 호스트 이름 유형(Hostname type)을 선택하면 EC2 인스턴스가 리소스 이름 형식의 호스트 이름으로 시작됩니다. 이 경우 이 EC2 인스턴스의 DNS 레코드는 리소스 이름을 가리킬 수도 있습니다. 따라서 호스트 이름이 인스턴스의 IPv4 주소, IPv6 주소 또는 IPv4 및 IPv6 주소로 확인되는지 여부를 유연하게 선택할 수 있습니다. 향후 IPv6를 사용할 계획이거나 현재 듀얼 스택 서브넷을 사용하는 경우, 리소스 이름(Resource name)의 호스트 이름 유형(Hostname type)을 사용하여 DNS 레코드 자체를 변경하지 않고 인스턴스의 호스트 이름에 대한 DNS 확인을 변경할 수 있습니다. 리소스 이름을 사용하면 EC2 인스턴스에서 IPv4 및 IPv6 DNS 확인을 추가 및 제거할 수 있습니다.

대신 IP 이름(IP name)의 호스트 이름 유형(Hostname type)을 선택하고 이를 DNS 호스트 이름으로 사용하면 인스턴스의 IPv4 주소로만 확인됩니다. 인스턴스에 연결된 IPv4 주소와 IPv6 주소가 모두 있더라도 인스턴스의 IPv6 주소로 확인되지 않습니다.

Amazon EC2에 대한 리소스 기반 이름 지정 옵션 변경

서브넷에 대한 호스트 이름 유형과 DNS 호스트 이름 구성을 변경할 수 있으며, 이는 해당 주체의 모든 후속 인스턴스 시작에 영향을 줍니다. 또는 EC2 인스턴스 시작 후 해당 인스턴스에 대한 호스트 이름 유형과 DNS 호스트 이름 구성을 변경할 수 있습니다.

리소스 기반 이름 지정 옵션

- 호스트 이름 유형: 서브넷에서 시작된 EC2 인스턴스의 게스트 OS 호스트 이름의 기본 설정을 결정합니다. 이 설정은 리소스 이름 또는 IP 이름입니다.
- DNS 호스트 IPv4(A 레코드) 요청 활성화: 리소스 이름에 대한 DNS 요청/쿼리가 EC2 인스턴스의 프라이빗 IPv4 주소(A 레코드)로 확인되는지 여부를 결정합니다.
- DNS 호스트 IPv6(AAAA 레코드) 요청 활성화: 리소스 이름에 대한 DNS 요청/쿼리가 EC2 인스턴스의 IPv6 주소(AAAA 레코드)로 확인되는지 여부를 결정합니다.

서브넷

서브넷 설정을 변경해도 서브넷에서 이미 시작된 EC2 인스턴스의 구성은 변경되지 않습니다.

Console

서브넷에 대한 옵션을 수정하려면

Amazon VPC 콘솔을 열고 서브넷을 선택합니다. 작업, 서브넷 설정 편집을 차례로 선택합니다. 필요에 따라 설정을 수정한 다음 변경 사항을 저장합니다.

AWS CLI

서브넷에 대한 옵션을 수정하려면

modify-subnet-attribute 명령을 사용합니다.

```
aws ec2 modify-subnet-attribute \
    --subnet-id subnet-0abcdef1234567890 \
    --private-dns-hostname-type-on-launch resource-name \
    --enable-resource-name-dns-a-record-on-launch \
    --enable-resource-name-dns-aaaa-record-on-launch
```

PowerShell

서브넷에 대한 옵션을 수정하려면

Edit-EC2SubnetAttribute cmdlet을 사용합니다.

```
Edit-EC2SubnetAttribute `
    -SubnetId subnet-0abcdef1234567890 `
    -PrivateDnsHostnameTypeOnLaunch ResourceName `
    -EnableResourceNameDnsAAAARecordOnLaunch $true `
    -EnableResourceNameDnsARecordOnLaunch $true
```

EC2 인스턴스

고려 사항

- 호스트 이름 유형을 변경하려면 먼저 인스턴스를 중지해야 합니다. 다른 두 옵션은 변경하기 위해 인스턴스를 중지할 필요가 없습니다.
- 인스턴스 저장소 지원 인스턴스는 중지할 수 없으므로 인스턴스 시작 시에만 호스트 이름 유형 및 DNS 호스트 이름을 구성할 수 있습니다. 다음과 같은 인스턴스 유형만 인스턴스 저장소 볼륨을 루트 볼륨으로 지원합니다. C1, C3, D2, I2, M1, M2, M3, R3, X1.

Console

인스턴스의 호스트 이름 유형 및 DNS 호스트 이름 옵션을 수정하려면

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

2. 리소스 기반 이름 지정을 게스트 OS 호스트 이름으로 사용(Use resource based naming as guest OS hostname) 설정을 변경하려면 먼저 EC2 인스턴스를 중지합니다. 그렇지 않은 경우 이 단계를 건너뜁니다.

   인스턴스를 중지하려면 인스턴스를 선택하고 인스턴스 상태(Instance state), 인스턴스 중지(Stop instance)를 차례로 선택합니다.
3. 인스턴스를 선택한 다음 작업, 인스턴스 설정, 리소스 기반 이름 지정 옵션 변경을 차례로 선택합니다.

   - 리소스 기반 이름 지정을 게스트 OS 호스트 이름으로 사용: EC2 인스턴스의 게스트 OS 호스트 이름을 리소스 이름으로 지정할지 아니면 IP 이름으로 지정할지 결정합니다.
   - DNS 호스트 IPv4(A 레코드) 요청에 응답(Answer DNS hostname IPv4 (A record) requests): 리소스 이름에 대한 DNS 요청/쿼리가 이 EC2 인스턴스의 프라이빗 IPv4 주소로 확인되는지 여부를 결정합니다.
   - DNS 호스트 IPv6(AAAA 레코드) 요청에 응답: 리소스 이름에 대한 DNS 요청/쿼리가 이 EC2 인스턴스의 IPv6 주소(AAAA 레코드)로 확인되는지 여부를 결정합니다.
4. 저장을 선택합니다.
5. 인스턴스를 중지한 후 다시 시작할 수 있습니다.

AWS CLI

인스턴스의 호스트 이름 유형 및 DNS 호스트 이름 옵션을 수정하려면

modify-private-dns-name-options 명령을 사용합니다.

```
aws ec2 modify-private-dns-name-options \
    --instance-id i-1234567890abcdef0 \
    --private-dns-hostname-type resource-name \
    --enable-resource-name-dns-a-record \
    --enable-resource-name-dns-aaaa-record
```

PowerShell

인스턴스의 호스트 이름 유형 및 DNS 호스트 이름 옵션을 수정하려면

Edit-EC2PrivateDnsNameOption cmdlet을 사용합니다.

```
Edit-EC2PrivateDnsNameOption `
```

```
    -InstanceId i-1234567890abcdef0 `
    -PrivateDnsHostnameType ResourceName `
    -EnableResourceNameDnsAAAARecord $true`
    -EnableResourceNameDnsARecord $true
```

Amazon EC2로 고유 IP 주소 가져오기(BYOIP)

온프레미스 네트워크에서 공개적으로 라우팅 가능한 모든 IPv4 또는 IPv6 주소 범위의 일부 또는 전부를 AWS 계정으로 가져올 수 있습니다. 주소 범위를 계속해서 제어할 수 있으며 AWS를 통해 인터넷에 주소 범위를 알릴 수 있습니다. 주소 범위를 Amazon EC2로 가져오면 이러한 주소가 AWS 계정에 주소 풀로 나타납니다.

> **Note**
>
> 이 설명서에서는 Amazon EC2에서만 사용할 고유 IP 주소 범위를 가져오는 방법을 설명합니다. AWS Global Accelerator에서 사용할 자체 IP 주소 범위를 가져오는 방법은 AWS Global Accelerator 개발자 안내서에서 자체 IP 주소 가져오기(BYOIP)를 참조하세요. Amazon VPC IP Address Manager와 함께 사용할 고유한 IP 주소 범위를 가져오려면 Amazon VPC IPAM 사용 설명서에서 튜토리얼: IPAM으로 IP 주소 가져오기를 참조하세요.

IP 주소 범위를 AWS로 가져오면 AWS에서 IP 주소 범위를 제어할 수 있는지 검증합니다. 범위를 제어한다는 것을 보여주기 위해 두 가지 방법을 사용할 수 있습니다.

- IP 주소 범위가 RDAP를 지원하는 인터넷 레지스트리(예: ARIN, RIPE, APNIC)에 등록된 경우 이 페이지의 프로세스를 사용하여 X.509 인증서로 도메인 제어를 확인할 수 있습니다. 인증서는 프로비전 프로세스 기간 동안만 유효해야 합니다. 프로비저닝 단계가 완료된 후 RIR 기록에서 인증서를 제거할 수 있습니다.
- 인터넷 레지스트리가 RDAP를 지원하는지 여부와 무관하게 Amazon VPC IPAM을 사용하여 DNS TXT 레코드로 도메인의 제어를 확인할 수 있습니다. 이 프로세스는 Amazon VPC IPAM 사용 설명서의 자습서: IPAM으로 IP 주소 가져오기에 설명되어 있습니다.

자세한 내용은 AWS 온라인 테크 톡 Deep Dive on Bring Your Own IP를 참조하세요.

내용

- BYOIP 정의
- 요구 사항 및 할당량

- 리전별 가용성
- 로컬 영역 가용성
- Amazon EC2에서 BYOIP를 위한 사전 조건
- Amazon EC2에서 사용할 주소 범위 온보딩
- Amazon EC2에서 BYOIP 주소 범위 사용

BYOIP 정의

- X.509 자체 서명 인증서 - 네트워크 내에서 데이터를 암호화하고 인증하는 데 가장 일반적으로 사용되는 인증서 표준입니다. RDAP 레코드에서 IP 공간에 대한 제어를 검증하기 위해 AWS에서 사용하는 인증서입니다. X.509 인증서에 대한 자세한 내용은 RFC 3280을 참조하세요
- Autonomous System Number(ASN) - IP 접두사 그룹을 정의하는 글로벌 고유 식별자로, 명확하게 정의된 단일 라우팅 정책을 유지 관리하는 하나 이상의 네트워크 운영자에 의해 실행됩니다.
- 리전 인터넷 레지스트리(RIR) - 전 세계의 한 리전 내에서 IP 주소 및 ASN의 할당, 등록을 관리하는 조직입니다.
- RDAP(Registry Data Access Protocol) - RIR 내에서 현재 등록 데이터를 쿼리하는 읽기 전용 프로토콜입니다. 쿼리된 RIR 데이터베이스 내의 항목을 “RDAP 레코드”라고 합니다. 특정 레코드 유형은 RIR 제공 메커니즘을 통해 고객이 업데이트해야 합니다. 이러한 레코드는 AWS에서 쿼리하여 RIR의 주소 공간 제어를 확인합니다.
- ROA(Route Origin Authorization) - 고객이 특정 자율 시스템에서 IP 알림을 인증하기 위해 RIR에서 생성한 객체입니다. 개요는 ARIN 웹 사이트의 Route Origin Authorizations (ROAs)(ROA(Route Origin Authorization))를 참조하세요.
- 로컬 인터넷 레지스트리(LIR) - 고객을 위해 RIR에서 IP 주소 블록을 할당하는 인터넷 서비스 제공업체와 같은 조직입니다.

요구 사항 및 할당량

- 주소 범위는 리전 인터넷 레지스트리(RIR)에 등록되어 있어야 합니다. 지리적 리전과 관련된 모든 정책은 RIR을 참조하세요. BYOIP는 현재 미국 인터넷 번호 등록 협회(ARIN), Réseaux IP Européens Network Coordination Centre(RIPE) 또는 아시아 태평양 지역 네트워크 정보센터(APNIC) 등록을 지원합니다. 주소 범위를 기업 또는 기관에 등록해야 하며 개인에게는 등록할 수 없습니다.
- 가져올 수 있는 가장 구체적인 IPv4 주소 범위는 /24입니다.
- 가져올 수 있는 가장 구체적인 IPv6 주소 범위는 공개적으로 알려지는 CIDR의 경우 /48이고, 공개적으로 알려지지 않는 CIDR의 경우 /60입니다.

- 공개적으로 알려지지 않은 CIDR 범위에는 ROA가 필요하지 않지만 RDAP 레코드는 여전히 업데이트해야 합니다.
- 한 번에 하나의 AWS 리전으로 각 주소 범위를 가져올 수 있습니다.
- AWS 리전당 총 5개의 BYOIP IPv4 및 IPv6 주소 범위를 AWS 계정으로 가져올 수 있습니다. Service Quotas 콘솔을 사용하여 BYOIP CIDR에 대한 할당량을 조정할 수는 없지만, AWS 일반 참조의 AWS 서비스 할당량에 설명된 대로 AWS 지원 센터에 문의하여 할당량 증가를 요청할 수 있습니다.
- Amazon VPC IP Address Manager(IPAM)를 사용하고 IPAM을 AWS Organizations과 통합하지 않는 한 AWS RAM을 사용하여 IP 주소 범위를 다른 계정과 공유할 수 없습니다. 자세한 내용은 Amazon VPC IPAM 사용 설명서의 AWS Organizations와 IPAM 통합을 참조하세요.
- 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 권리를 보유합니다.
- 리전 인터넷 레지스트리(RIR) 시스템이 구성되기 전에 IANA(인터넷 할당 번호 기관)의 중앙 레지스트리에서 배포한 IPv4 주소 스페이스인 레거시 주소 스페이스에는 여전히 해당 ROA 객체가 필요합니다.
- LIR의 경우 수동 프로세스를 사용하여 레코드를 업데이트하는 것이 일반적입니다. LIR에 따라 배포하는 데 며칠이 걸릴 수 있습니다.
- 큰 CIDR 블록에는 단일 ROA 객체와 RDAP 레코드가 필요합니다. 단일 객체와 레코드를 사용하여 여러 AWS 리전에서도 해당 범위에서 AWS로 여러 개의 작은 CIDR 블록을 가져올 수 있습니다.
- BYOIP는 Wavelength 영역 또는 AWS Outposts에서 지원되지 않습니다.
- RADb 또는 기타 IRR에서 BYOIP를 수동으로 변경하지 마세요. BYOIP에서 RADb를 자동으로 업데이트합니다. BYOIP ASN 등을 수동으로 변경하면 BYOIP 프로비저닝 작업에 실패하게 됩니다.
- 사용하여 AWS로 IPv4 주소 범위를 가져오면 첫 번째 주소(네트워크 주소)와 마지막 주소(브로드캐스트 주소)를 포함하여 범위 내의 IP 주소를 모두 사용할 수 있습니다.

리전별 가용성

BYOIP 기능은 현재 중국 리전을 제외한 모든 상용 AWS 리전에서 사용할 수 있습니다.

로컬 영역 가용성

Local Zone은 사용자와 지리적으로 근접한 AWS 리전의 확장입니다. 로컬 영역은 '네트워크 경계 그룹'으로 그룹화됩니다. AWS에서 네트워크 경계 그룹은 AWS에서 퍼블릭 IP 주소를 알리는 가용 영역(AZ), 로컬 영역 또는 Wavelength 영역의 모음입니다. 로컬 영역은 AWS 네트워크와 해당 영역의 리소

스에 액세스하는 고객 사이에서 지연 시간 또는 물리적 거리를 최소화하기 위해 AWS 리전의 AZ와 다른 네트워크 경계 그룹을 보유할 수 있습니다.

`--network-border-group` 옵션을 사용하여 BYOIPv4 주소 범위를 프로비저닝하고 다음 로컬 영역 네트워크 경계 그룹에 알릴 수 있습니다.

- af-south-1-los-1
- ap-northeast-1-tpe-1
- ap-south-1-ccu-1
- ap-south-1-del-1
- ap-southeast-1-bkk-1
- ap-southeast-1-mnl-1
- ap-southeast-2-akl-1
- ap-southeast-2-per-1
- eu-central-1-ham-1
- eu-central-1-waw-1
- eu-north-1-cph-1
- eu-north-1-hel-1
- me-south-1-mct-1
- us-east-1-atl-2
- us-east-1-bos-1
- us-east-1-bue-1
- us-east-1-chi-2
- us-east-1-dfw-2
- us-east-1-iah-2
- us-east-1-lim-1
- us-east-1-mci-1
- us-east-1-mia-2
- us-east-1-msp-1
- us-east-1-nyc-1
- us-east-1-nyc-2
- us-east-1-phl-1

- us-east-1-qro-1
- us-east-1-scl-1
- us-west-2-den-1
- us-west-2-hnl-1
- us-west-2-las-1
- us-west-2-lax-1
- us-west-2-pdx-1
- us-west-2-phx-2
- us-west-2-sea-1

로컬 영역을 활성화한 경우(로컬 영역 활성화 참조) BYOIPv4 CIDR을 프로비저닝하고 알릴 때 로컬 영역에 대한 네트워크 경계 그룹을 선택할 수 있습니다. EIP와 연결된 AWS 리소스는 동일한 네트워크 경계 그룹에 있어야 하므로 네트워크 경계 그룹을 신중하게 선택하세요.

> **Note**
>
> 지금은 로컬 영역에서 BYOIPv6 주소 범위를 프로비저닝하거나 알릴 수 없습니다.

Amazon EC2에서 BYOIP를 위한 사전 조건

BYOIP에 대한 온보딩 프로세스는 3개의 하위 단계를 포함하는 2단계로 이루어집니다. 이러한 단계는 다음 다이어그램에 설명된 단계에 해당합니다. 이 설명서에는 수동 단계가 포함되어 있지만 RIR에서 이러한 단계를 지원하는 관리형 서비스를 제공할 수 있습니다.

> **Tip**
>
> 이 섹션의 작업을 수행하려면 Linux 터미널이 필요하며, Linux, AWS CloudShell 또는 Windows Subsystem for Linux를 사용하여 수행할 수 있습니다.

내용

- 개요
- 프라이빗 키 생성 및 X.509 인증서 생성
- RIR의 RDAP 레코드에 X.509 인증서 업로드

- RIR에 ROA 객체 생성

개요

준비 단계

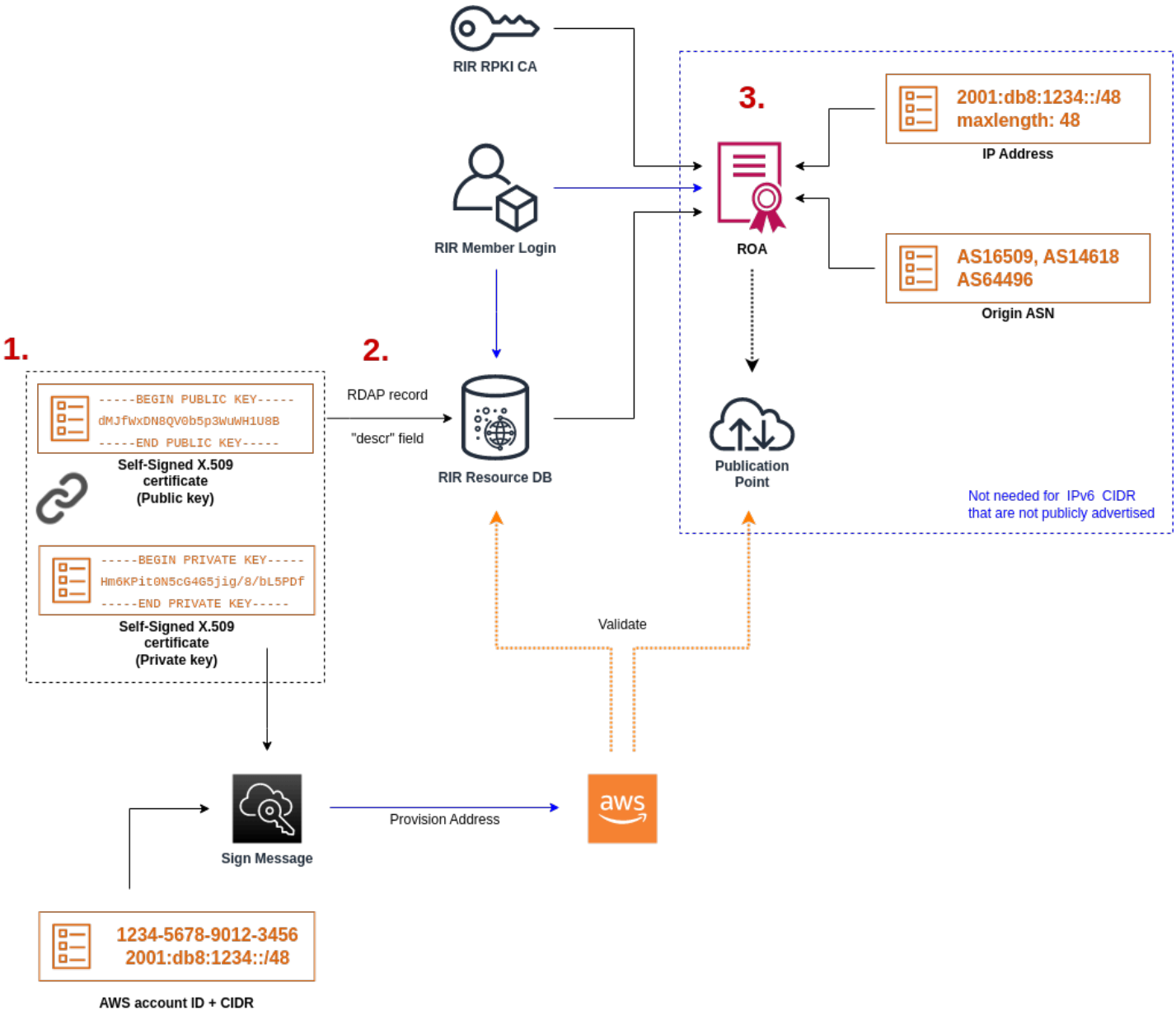
[1] 프라이빗 키를 생성하고 이를 사용하여 인증 목적으로 자체 서명된 X.509 인증서를 생성합니다. 이 인증서는 프로비저닝 단계에서만 사용됩니다. 프로비저닝 단계가 완료된 후 RIR 기록에서 인증서를 제거할 수 있습니다.

RIR 구성 단계

[2] RDAP 레코드 주석에 자체 서명된 인증서를 업로드합니다.

[3] RIR에서 ROA 객체를 생성합니다. ROA는 원하는 주소 범위, 주소 범위를 알릴 수 있는 자율 시스템 번호(ASN) 및 RIR의 리소스 퍼블릭 키 인프라(RPKI)에 등록할 만료 날짜를 정의합니다.

> **Note**
>
> 비공개적으로 알려진 IPv6 주소 공간에는 ROA가 필요하지 않습니다.



여러 불연속 주소 범위를 가져오려면 각 주소 범위에 대해 이 프로세스를 반복해야 합니다. 그러나 연속 블록을 여러 AWS 리전에 걸쳐 분할하는 경우 준비 및 RIR 구성 단계를 반복할 필요가 없습니다.

주소 범위를 가져오더라도 이전에 가져온 주소 범위에는 영향을 주지 않습니다.

프라이빗 키 생성 및 X.509 인증서 생성

다음 절차에 따라 자체 서명된 X.509 인증서를 생성하여 RIR의 RDAP 레코드에 추가합니다. 이 키 페어는 RIR로 주소 범위를 인증하는 데 사용됩니다. openssl 명령에는 OpenSSL 버전 1.0.2 이상이 필요합니다.

다음 명령을 복사하고 자리 표시자 값(색상이 있는 기울임꼴 텍스트)만 바꿉니다.

이 절차는 프라이빗 RSA 키의 암호화 및 액세스 시 암호 필수 사용에 대한 모범 사례를 따릅니다.

1. 다음과 같이 표시된 대로 RSA 2048비트 프라이빗 키를 생성합니다.

```
$ openssl genpkey -aes256 -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -out
 private-key.pem
```

-aes256 파라미터는 프라이빗 키 암호화에 사용되는 알고리즘을 지정합니다. 이 명령은 암호 설정 프롬프트를 포함한 다음 출력을 반환합니다.

```
......+++
.+++
Enter PEM pass phrase: xxxxxxx
Verifying - Enter PEM pass phrase: xxxxxxx
```

다음 명령을 사용하여 키를 검사할 수 있습니다.

```
$ openssl pkey -in private-key.pem -text
```

다음과 유사한 암호 프롬프트와 키 내용이 반환됩니다.

```
Enter pass phrase for private-key.pem: xxxxxxx
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQDFBXHRI4HVKAhh
3seiciooizCRTbJe1+YsxNTja4XyKypVGIFWDGhZs44FCHlPOOSVJ+NqP74w96oM
7DPS3xo9kaQyZBFn2YEp2EBq5vf307KHNRmZZUmkn0zHOSEpNmY2fMxISBxewlxR
FAniwmSd/8TDvHJMY9FvAIvWuTsv5l0tJKk+a91K4+tO3UdDR7Sno5WXExfsBrW3
g1ydo3TBsx8i5/YiVOcNApy7ge2/FiwY3aCXJB6r6nuF6H8mRgI4r4vkMRsOlAhJ
DnZPNeweboo+K3Q3lwbgbmOKD/z9svk8N/+hUTBtIX0fRtbG+PLIw3xWRHGrMSn2
BzsPVuDLAgMBAAECggEACiJUj2hfJkKv47Dc3es3Zex67A5uDVjXmxfox2Xhdupn
fAcNqAptV6fXt0SPUNbhUxbBKNbshoJGufFwXPli1SXnpzvkdU4Hyco4zgbhXFsE
RNYjYfOGzTPwdBLpNMB6k3Tp4RHse6dNrlH0jDhpioL8cQEBdBJyVF5X0wymEbmV
mC0jgH/MxsBAPWW6ZKicg9ULMlWiAZ3MRAZPjHHgpYkAAsUWKAbCBwVQcVjGO59W
jfZjzTX5pQtVVH68ruciH88DTZCwjCkjBhxg+OIkJBLE5wkh82jIHSivZ63flwLw
z+E0+HhELSZJrn2MY6Jxmik3qNNUOF/Z+3msdj2luQKBgQDjwlC/3jxp8zJy6P8o
JQKv7TdvMwUj4VSWOHZBHLv4evJaaia0uQjIo1UDa8AYitqhX1NmCCehGH8yuXj/
v6V3CzMKDkmRr1NrONnSz5QsndQ04Z6ihAQlPmJ96g4wKtgoC7AYpyP0g1a+4/sj
b1+o3YQI4pD/F71c+qaztH7PRwKBgQDdc23yNmT3+Jyptf0fKjEvONK+xwUKzi9c
L/OzBq5yOIC1Pz2T85gOe1i8kwZws+xlpG6uBT6lmIJELd0k59FyupNu4dPvX5SD
6GGqdx4jk9KvI74usGeOBohmF0phTHkrWKBxXiyT0oS8zjnJlEn8ysIpGgO28jjr
LpaHNZ/MXQKBgQDfLNcnS0LzpsS2aK0tzyZU8SMyqVHOGMxj7quhneBq2T6FbiLD
```

```
T9TVlYaGNZ0j71vQaLI19qOubWymbautH0Op5KV8owdf4+bf1/NJaPIOzhDUSIjD
Qo01WW31Z9XDSRhKFTnWzmCjBdeIcajyzf10YKsycaAW9lItu8aBrMndnQKBgQDb
nNp/JyRwqjOrNljk7DHEs+SD39kHQzzCfqd+dnTPv2sc06+cpym3yulQcbokULpy
fmRo3bin/pvJQ3aZX/Bdh9woTXqhXDdrrSwWInVYMQPyPk8f/D9mIOJp5FUWMwHD
U+whIZSxsEeE+jtixlWtheKRYkQmzQZXbWdIhYyI3QKBgD+F/6wcZ85QW8nAUykA
3WrSIx/3cwDGdm4NRGct8ZOZjTHjiy9ojMOD1L7iMhRQ/3k3hUsin5LDMp/ryWGG
x4uIaLat40kiC7T4I66DM7P59euqdz3w0PD+VU+h7GSivvsFDdySUt7bNK0AUVLh
dMJfWxDN8QV0b5p3WuWH1U8B
-----END PRIVATE KEY-----
Private-Key: (2048 bit)
modulus:
    00:c5:05:71:d1:23:81:d5:28:08:61:de:c7:a2:72:
    2a:28:8b:30:91:4d:b2:5e:d7:e6:2c:c4:d4:e3:6b:
    85:f2:2b:2a:55:18:81:56:0c:68:59:b3:8e:05:08:
    79:4f:38:e4:95:27:e3:6a:3f:be:30:f7:aa:0c:ec:
    33:d2:df:1a:3d:91:a4:32:64:11:67:d9:81:29:d8:
    40:6a:e6:f7:f7:d3:b2:87:35:19:99:65:49:a4:9f:
    4c:c7:39:21:29:36:66:36:7c:cc:48:48:1c:5e:c2:
    5c:51:14:09:e2:c2:64:9d:ff:c4:c3:bc:72:4c:63:
    d1:6f:00:8b:d6:b9:3b:2f:e6:5d:2d:24:a9:3e:6b:
    dd:4a:e3:eb:4e:dd:47:43:47:b4:a7:a3:95:97:13:
    17:ec:06:b5:b7:83:5c:9d:a3:74:c1:b3:1f:22:e7:
    f6:22:54:e7:0d:02:9c:bb:81:ed:bf:16:2c:18:dd:
    a0:97:24:1e:ab:ea:7b:85:e8:7f:26:46:02:38:af:
    8b:e4:31:1b:0e:94:08:49:0e:76:4f:35:ec:1e:6e:
    8a:3e:2b:74:37:97:06:e0:6e:63:8a:0f:fc:fd:b2:
    f9:3c:37:ff:a1:51:30:6d:21:7d:1f:46:d6:c6:f8:
    f2:c8:c3:7c:56:44:71:ab:31:29:f6:07:3b:0f:56:
    e0:cb
publicExponent: 65537 (0x10001)
privateExponent:
    0a:22:54:8f:68:5f:26:42:af:e3:b0:dc:dd:eb:37:
    65:ec:7a:ec:0e:6e:0d:58:d7:9b:17:e8:c7:65:e1:
    76:ea:67:7c:07:0d:a8:0a:6d:57:a7:d7:b7:44:8f:
    50:d6:e1:53:16:c1:28:d6:ec:86:82:46:b9:f1:70:
    5c:f9:62:d5:25:e7:a7:3b:e4:75:4e:07:c9:ca:38:
    ce:06:e1:5c:5b:04:44:d6:23:61:f3:86:cd:33:f0:
    74:12:e9:34:c0:7a:93:74:e9:e1:11:ec:7b:a7:4d:
    ae:51:f4:8c:38:69:8a:82:fc:71:01:01:74:12:72:
    54:5e:57:d3:0c:a6:11:b9:95:98:2d:23:80:7f:cc:
    c6:c0:40:3d:65:ba:64:a8:9c:83:d5:0b:32:55:a2:
    01:9d:cc:44:06:4f:8c:71:e0:a5:89:00:02:c5:16:
    28:06:c2:07:05:50:71:58:c6:3b:9f:56:8d:f6:63:
    cd:35:f9:a5:0b:55:54:7e:bc:ae:e7:22:1f:cf:03:
```

```
    4d:90:b0:8c:29:23:06:1c:60:f8:e2:24:24:12:c4:
    e7:09:21:f3:68:c8:1d:28:af:67:ad:df:97:02:f0:
    cf:e1:34:f8:78:44:2d:26:49:ae:7d:8c:63:a2:71:
    9a:29:37:a8:d3:54:38:5f:d9:fb:79:ac:76:3d:a5:
    b9
prime1:
    00:e3:c2:50:bf:de:3c:69:f3:32:72:e8:ff:28:25:
    02:af:ed:37:6f:33:05:23:e1:54:96:38:76:41:1c:
    bb:f8:7a:f2:5a:6a:26:b4:b9:08:c8:a3:55:03:6b:
    c0:18:8a:da:a1:5f:53:66:08:27:a1:18:7f:32:b9:
    78:ff:bf:a5:77:0b:33:0a:0e:49:91:af:53:6b:38:
    d9:d2:cf:94:2c:9d:d4:34:e1:9e:a2:84:04:25:3e:
    62:7d:ea:0e:30:2a:d8:28:0b:b0:18:a7:23:f4:83:
    56:be:e3:fb:23:6f:5f:a8:dd:84:08:e2:90:ff:17:
    bd:5c:fa:a6:b3:b4:7e:cf:47
prime2:
    00:dd:73:6d:f2:36:64:f7:f8:9c:a9:b5:fd:1f:2a:
    31:2f:38:d2:be:c7:05:0a:ce:2f:5c:2f:f3:b3:06:
    ae:72:38:80:b5:3f:3d:93:f3:98:0e:7b:58:bc:93:
    06:70:b3:ec:65:a4:6e:ae:05:3e:a5:98:82:44:2d:
    dd:24:e7:d1:72:ba:93:6e:e1:d3:ef:5f:94:83:e8:
    61:aa:77:1e:23:93:d2:af:23:be:2e:b0:67:8e:06:
    88:66:17:4a:61:4c:79:2b:58:a0:71:5e:2c:93:d2:
    84:bc:ce:39:c9:94:49:fc:ca:c2:29:1a:03:b6:f2:
    38:eb:2e:96:87:35:9f:cc:5d
exponent1:
    00:df:2c:d7:27:4b:42:f3:a6:c4:b6:68:ad:2d:cf:
    26:54:f1:23:32:a9:51:ce:18:cc:63:ee:ab:a1:9d:
    e0:6a:d9:3e:85:6e:22:c3:4f:d4:d5:95:86:86:35:
    9d:23:ef:5b:d0:68:b2:35:f6:a3:ae:6d:6c:a6:6d:
    ab:ad:1f:43:a9:e4:a5:7c:a3:07:5f:e3:e6:df:d7:
    f3:49:68:f2:0e:ce:10:d4:48:88:c3:42:8d:35:59:
    6d:f5:67:d5:c3:49:18:4a:15:39:d6:ce:60:a3:05:
    d7:88:71:a8:f2:cd:fd:74:60:ab:32:71:a0:16:f6:
    52:2d:bb:c6:81:ac:c9:dd:9d
exponent2:
    00:db:9c:da:7f:27:24:70:aa:33:ab:36:58:e4:ec:
    31:c4:b3:e4:83:df:d9:07:43:3c:c2:7e:a7:7e:76:
    74:cf:bf:6b:1c:d3:af:9c:a7:29:b7:ca:e9:50:71:
    ba:24:50:ba:72:7e:64:68:dd:b8:a7:fe:9b:c9:43:
    76:99:5f:f0:5d:87:dc:28:4d:7a:a1:5c:37:6b:ad:
    2c:16:22:75:58:31:03:f2:3e:4f:1f:fc:3f:66:20:
    e2:69:e4:55:16:33:01:c3:53:ec:21:21:94:b1:b0:
    47:84:fa:3b:62:c6:55:ad:85:e2:91:62:44:26:cd:
```

```
    06:57:6d:67:48:85:8c:88:dd
coefficient:
    3f:85:ff:ac:1c:67:ce:50:5b:c9:c0:53:29:00:dd:
    6a:d2:23:1f:f7:73:00:c6:76:6e:0d:44:67:2d:f1:
    93:99:8d:31:e3:8b:2f:68:8c:c3:83:d4:be:e2:32:
    14:50:ff:79:37:85:4b:22:9f:92:c3:32:9f:eb:c9:
    61:86:c7:8b:88:68:b6:ad:e3:49:22:0b:b4:f8:23:
    ae:83:33:b3:f9:f5:eb:aa:77:3d:f0:d0:f0:fe:55:
    4f:a1:ec:64:a2:be:fb:05:0d:dc:92:52:de:db:34:
    ad:00:51:52:e1:74:c2:5f:5b:10:cd:f1:05:74:6f:
    9a:77:5a:e5:87:d5:4f:01
```

프라이빗 키를 사용하지 않을 때는 안전한 위치에 보관하세요.

2. 이전 단계에서 생성한 프라이빗 키를 사용하여 X.509 인증서를 생성합니다. 이 예제에서 인증서는 365일이 지나면 만료되므로, 이 기간이 지난 후에는 신뢰할 수 없습니다. 따라서 만료 날짜를 적절하게 설정해야 합니다. 인증서는 프로비전 프로세스 기간 동안만 유효해야 합니다. 프로비저닝 단계가 완료된 후 RIR 기록에서 인증서를 제거할 수 있습니다. `tr -d "\n"` 명령은 출력에서 줄 바꿈 문자를 제거합니다. 메시지가 표시되면 일반 이름을 제공해야 하지만 다른 필드는 비워 둘 수 있습니다.

```
$ openssl req -new -x509 -key private-key.pem -days 365 | tr -d "\n" >
 certificate.pem
```

결과로 다음과 유사한 출력이 반환됩니다.

```
Enter pass phrase for private-key.pem: xxxxxxx
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) []:
State or Province Name (full name) []:
Locality Name (eg, city) []:
Organization Name (eg, company) []:
Organizational Unit Name (eg, section) []:
Common Name (eg, fully qualified host name) []:example.com
Email Address []:
```

> **Note**
>
> AWS 프로비저닝에는 일반 이름이 필요하지 않습니다. 내부 또는 퍼블릭 도메인 이름일 수 있습니다.

다음 명령을 사용하여 인증서를 검사할 수 있습니다.

```
$ cat certificate.pem
```

출력은 줄 바꿈이 없는 긴 PEM 인코딩 문자로, `-----BEGIN CERTIFICATE-----` 접두사가 있고 `-----END CERTIFICATE-----`가 뒤에 나옵니다.

RIR의 RDAP 레코드에 X.509 인증서 업로드

RIR에 대한 RDAP 레코드에 이전에 생성되었던 인증서를 추가합니다. 인코딩된 부분 앞과 뒤에 `-----BEGIN CERTIFICATE-----` 및 `-----END CERTIFICATE-----` 문자열을 포함해야 합니다. 이 모든 내용은 하나의 긴 줄에 있어야 합니다. RDAP를 업데이트하는 절차는 RIR에 따라 다릅니다.

- ARIN의 경우, 계정 관리자 포털을 사용하여 주소 범위를 나타내는 '네트워크 정보' 객체의 '공개 주석' 섹션에 인증서를 추가합니다. 조직의 주석 섹션에 추가하지 않습니다.
- RIPE의 경우, 주소 범위를 나타내는 'inetnum' 또는 'inet6num' 객체에 새 'descr' 필드로서 인증서를 추가합니다. 대체로 RIPE 데이터베이스 포털의 "내 리소스" 섹션에서 찾을 수 있습니다. 조직의 주석 섹션이나 위 객체의 '비고' 필드에 추가하지 않습니다.
- APNIC의 경우, 이메일을 통해 인증서를 helpdesk@apnic.net로 전송하여 주소 범위의 '비고' 필드에 수동으로 추가합니다. IP 주소의 APNIC 공인 연락처를 사용하여 이메일을 전송합니다.

아래 프로비저닝 단계가 완료된 후 RIR 레코드에서 인증서를 제거할 수 있습니다.

RIR에 ROA 객체 생성

ROA 객체를 생성하여 현재 주소 범위를 알리도록 권한을 부여받은 ASN뿐만 아니라 Amazon ASN 16509 및 14618이 주소 범위를 알리도록 권한을 부여합니다. AWS GovCloud (US) Regions의 경우 16509 및 14618 대신 ASN 8987을 승인하세요. 가져올 CIDR의 크기로 최대 길이를 설정해야 합니다. 가져올 수 있는 가장 구체적인 IPv4 접두사는 /24입니다. 가져올 수 있는 가장 구체적인 IPv6 주소 범위는 공개적으로 알려지는 CIDR의 경우 /48이고, 공개적으로 알려지지 않는 CIDR의 경우 /60입니다.

> **Important**
>
> Amazon VPC IP Address Manager(IPAM)에 대한 ROA 객체를 생성하는 경우 ROA를 생성할 때 IPv4 CIDR에 대해 IP 주소 접두사의 최대 길이를 /24로 설정해야 합니다. IPv6 CIDR의 경우 알릴 수 있는 풀에 해당 CIDR을 추가하면 IP 주소 접두사의 최대 길이는 /48여야 합니다. 이렇게 하면 퍼블릭 IP 주소를 AWS 리전 전체에 걸쳐 충분히 유연하게 분할할 수 있습니다. IPAM은 사용자가 설정한 최대 길이를 적용합니다. IPAM에 대한 BYOIP 주소와 관련한 자세한 내용은 Amazon VPC IPAM 사용 설명서의 자습서: BYOIP 주소 CIDR을 IPAM으로 단원을 참조하세요.

Amazon이 ROA를 사용할 수 있을 때까지 최대 24시간이 걸릴 수 있습니다. 자세한 내용은 RIR에 문의하세요.

- ARIN - ROA 요청
- RIPE - ROA 관리
- APNIC — 라우팅 관리

온프레미스 워크로드에서 AWS로 광고를 마이그레이션하는 경우 Amazon의 ASN에 대한 ROA를 생성하기 전에 기존 ASN에 대한 ROA를 생성해야 합니다. 그렇지 않으면 기존 라우팅 및 광고에 영향을 줄 수 있습니다.

> **Important**
>
> Amazon에서 IP 주소 범위를 알린 후 이후에도 알리려면 Amazon ASN에서 ROA는 위의 지침을 준수해야 합니다. ROA가 유효하지 않거나 위의 규정을 준수하지 않는 경우 Amazon은 IP 주소 범위에 대한 알림을 중단할 권리를 보유합니다.

> **Note**
>
> 비공개적으로 알려진 IPv6 주소 공간에는 이 단계가 필요하지 않습니다.

Amazon EC2에서 사용할 주소 범위 온보딩

BYOIP의 온보딩 프로세스에는 필요에 따라 다음과 같은 태스크가 포함됩니다.

업무

- AWS에서 공개적으로 알려진 주소 범위 프로비저닝
- 공개적으로 알려지지 않는 IPv6 주소 범위 프로비저닝
- AWS을(를) 통해 주소 범위 알리기
- 주소 범위 프로비저닝 취소
- BYOIP 검증

AWS에서 공개적으로 알려진 주소 범위 프로비저닝

AWS에서 사용할 수 있도록 주소 범위를 프로비저닝하는 경우 주소 범위를 제어하고 있는지 확인하고 Amazon에 해당 주소 범위를 알릴 수 있는 권한을 부여하고 있는 것입니다. 또한 서명된 권한 부여 메시지를 통해 주소 범위를 제어하고 있음을 확인합니다. 이 메시지는 X.509 인증서로 RDAP 레코드를 업데이트할 때 사용한 자체 서명 X.509 키 페어로 서명됩니다. AWS에서는 암호화 방식으로 서명된 권한 부여 메시지를 RIR에 제출해야 합니다. RIR은 RDAP에 추가된 인증서를 기준으로 서명을 인증하고 ROA를 기준으로 권한 부여 세부 정보를 확인합니다.

주소 범위를 프로비저닝하려면

1. 메시지 작성

   일반 텍스트 권한 부여 메시지를 작성합니다. 메시지 형식은 다음과 같으며, 여기서 날짜는 메시지의 만료 날짜입니다.

   ```
   1|aws|account|cidr|YYYYMMDD|SHA256|RSAPSS
   ```

   계정 번호, 주소 범위 및 만료 날짜를 해당하는 값으로 바꾸어 다음과 같은 메시지를 생성합니다.

   ```
   text_message="1|aws|0123456789AB|198.51.100.0/24|20211231|SHA256|RSAPSS"
   ```

   형태가 비슷한 ROA 메시지와 혼동해서는 안 됩니다.

2. 메시지 서명

   이전에 생성한 프라이빗 키를 사용하여 일반 텍스트 메시지에 서명합니다. 이 명령에 의해 반환된 서명은 다음 단계에서 사용해야 하는 긴 문자열입니다.

> **Important**
>
> 이 명령을 복사하여 붙여넣는 것이 좋습니다. 메시지 내용을 제외하고 값을 수정하거나 바꾸지 마세요.

```
signed_message=$( echo -n $text_message | openssl dgst -sha256 -sigopt
 rsa_padding_mode:pss -sigopt rsa_pss_saltlen:-1 -sign private-key.pem -keyform PEM
 | openssl base64 | tr -- '+=/' '-_~' | tr -d "\n")
```

3. 주소 프로비저닝

   주소 범위를 프로비저닝하려면 AWS CLI provision-byoip-cidr 명령을 사용합니다. `--cidr-authorization-context` 옵션은 이전에 생성한 메시지 및 서명 문자열을 사용합니다.

   > **Important**
   >
   > Configure the AWS CLI `Default region name`과 다른 경우 BYOIP 범위를 프로비저닝할 AWS 리전을 지정해야 합니다.

   ```
   aws ec2 provision-byoip-cidr --cidr address-range --cidr-authorization-context
    Message="$text_message",Signature="$signed_message" --region us-east-1
   ```

   주소 범위 프로비저닝은 비동기 작업이므로, 호출이 즉시 반환되지만 주소 범위는 상태가 `pending-provision`에서 `provisioned`로 변경되어야 사용할 준비가 된 것입니다.

4. 진행률 모니터링

   대부분의 프로비저닝은 2시간 내에 완료되지만 공개적으로 알려진 범위에 대한 프로비저닝 프로세스를 완료하려면 최대 1주가 걸릴 수 있습니다. 진행률을 모니터링하려면 이 예제에서와 같이 describe-byoip-cidrs 명령을 사용합니다.

   ```
   aws ec2 describe-byoip-cidrs --max-results 5 --region us-east-1
   ```

   프로비저닝 중에 문제가 발생하고 상태가 `failed-provision`으로 전환되면 문제가 해결된 후 `provision-byoip-cidr` 명령을 다시 실행해야 합니다.

공개적으로 알려지지 않는 IPv6 주소 범위 프로비저닝

기본적으로 주소 범위는 인터넷에 공개적으로 알려지도록 프로비저닝됩니다. 공개적으로 알려지지 않는 IPv6 주소 범위를 프로비저닝할 수 있습니다. 공개적으로 알릴 수 없는 라우팅의 경우 프로비저닝 프로세스는 일반적으로 몇 분 이내에 완료됩니다. 비공개 주소 범위의 IPv6 CIDR 블록을 VPC와 연결하는 경우 IPv6 CIDR은 AWS Direct Connect, AWS Site-to-Site VPN 또는 Amazon VPC Transit Gateway와 같이 IPv6를 지원하는 하이브리드 연결 옵션을 통해서만 액세스할 수 있습니다.

ROA는 비공개 주소 범위를 프로비저닝하는 데 필요하지 않습니다.

> **Important**
>
> - 프로비저닝 중에 주소 범위를 공개적으로 알릴지 여부만 지정할 수 있습니다. 알리기 상태를 나중에 변경할 수는 없습니다.
> - Amazon VPC는 고유 로컬 주소(ULA) CIDR을 지원하지 않습니다. 모든 VPC에는 고유한 IPv6 CIDR이 있어야 합니다. 두 VPC는 IPv6 CIDR 범위가 같으면 안 됩니다.

공개적으로 알려지지 않는 IPv6 주소 범위를 프로비저닝하려면 다음 provision-byoip-cidr 명령을 사용합니다.

```
aws ec2 provision-byoip-cidr --cidr address-range --cidr-authorization-context
 Message="$text_message",Signature="$signed_message" --no-publicly-advertisable --
region us-east-1
```

AWS을(를) 통해 주소 범위 알리기

주소 범위가 프로비저닝되면 알릴 준비가 된 것입니다. 프로비저닝한 정확한 주소 범위를 알려야 합니다. 프로비저닝한 주소 범위의 일부만 알릴 수 없습니다.

공개적으로 알려지지 않는 IPv6 주소 범위를 프로비저닝한 경우 이 단계를 완료할 필요가 없습니다.

AWS을 통해 알리기 전에 다른 위치에서 주소 범위 또는 그 범위의 일부분 알리기를 중지하는 것이 좋습니다. 다른 위치에서 IP 주소 범위 또는 그 일부분 알리기를 계속하면 당사가 해당 주소 범위를 안정적으로 지원하지 못하거나 문제를 제대로 해결하지 못할 수 있습니다. 특히, 해당 주소 범위 또는 그 일부분으로의 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 AWS을(를) 통해 주소 알리기를 시작할 수

있습니다. 주소 풀의 탄력적 IP 주소 할당에 대한 자세한 내용은 탄력적 IP 주소 할당 섹션을 참조하세요.

제한 사항

- 매번 다른 주소 범위를 지정하더라도 최소 10초마다 advertise-byoip-cidr 명령을 실행할 수 있습니다.
- 매번 다른 주소 범위를 지정하더라도 최소 10초마다 withdraw-byoip-cidr 명령을 실행할 수 있습니다.

주소 범위를 알리려면 다음 provision-byoip-cidr 명령을 사용합니다.

```
aws ec2 advertise-byoip-cidr --cidr address-range --region us-east-1
```

주소 범위 알리기를 중지하려면 다음 withdraw-byoip-cidr 명령을 사용합니다.

```
aws ec2 withdraw-byoip-cidr --cidr address-range --region us-east-1
```

주소 범위 프로비저닝 취소

AWS에서 주소 범위 사용을 중지하려면 먼저 탄력적 IP 주소를 해제하고 여전히 주소 풀에서 할당된 IPv6 CIDR 블록의 연결을 해제합니다. 그런 다음 주소 범위 알리기를 중지하고 마지막으로 주소 범위 프로비저닝을 취소하세요.

주소 범위의 일부에 대해 프로비저닝을 취소할 수는 없습니다. AWS에서 보다 구체적인 주소 범위를 사용하려면 전체 주소 범위의 프로비저닝을 취소하고 보다 구체적인 주소 범위를 프로비저닝하세요.

(IPv4) 각 탄력적 IP 주소를 해제하려면 다음 release-address 명령을 사용합니다.

```
aws ec2 release-address --allocation-id eipalloc-12345678abcabcabc --region us-east-1
```

(IPv6) IPv6 CIDR 블록의 연결을 해제하려면 다음 disassociate-vpc-cidr-block 명령을 사용합니다.

```
aws ec2 disassociate-vpc-cidr-block --association-id vpc-cidr-assoc-12345abcd1234abc1
 --region us-east-1
```

주소 범위 알리기를 중지하려면 다음 withdraw-byoip-cidr 명령을 사용합니다.

```
aws ec2 withdraw-byoip-cidr --cidr address-range --region us-east-1
```

주소 범위 프로비저닝을 취소하려면 다음 deprovision-byoip-cidr 명령을 사용합니다.

```
aws ec2 deprovision-byoip-cidr --cidr address-range --region us-east-1
```

주소 범위의 프로비저닝을 해제하는 데 최대 하루가 걸릴 수 있습니다.

BYOIP 검증

1. 자체 서명된 x.509 키 페어 검증

   whois 명령을 통해 인증서가 업로드되었고 유효한지 검증합니다.

   ARIN의 경우 `whois -h whois.arin.net r + 2001:0DB8:6172::/48`을 사용하여 주소 범위에 대한 RDAP 레코드를 조회합니다. 명령 출력에서 NetRange(네트워크 범위)에 대한 Public Comments 섹션을 확인합니다. 인증서는 주소 범위에 대한 Public Comments 섹션에 추가되어야 합니다.

   다음 명령을 사용하여 인증서가 포함된 Public Comments를 검사할 수 있습니다.

   ```
   whois -h whois.arin.net r + 2001:0DB8:6172::/48 | grep Comments | grep BEGIN
   ```

   그러면 다음과 유사한 키 내용이 포함된 출력이 반환됩니다.

   ```
   Public Comments:
   -----BEGIN CERTIFICATE-----
   MIID1zCCAr+gAwIBAgIUBkRPNSLrPqbRAFP8RDAHSP+I1TowDQYJKoZIhvcNAQE
   LBQAwezELMAkGA1UEBhMCTloxETAPBgNVBAgMCEF1Y2tsYW5kMREwDwYDVQQHDA
   hBdWNrbGFuZDEcMBoGA1UECgwTQW1hem9uIFdlYiBTZXJ2aWNlczETMBEGA1UEC
   wwKQllPSVAgRGVtbzETMBEGA1UEAwwKQllPSVAgRGVtbzAeFw0yMTEyMDcyMDI0
   NTRaFw0yMjEyMDcyMDI0NTRaMHsxCzAJBgNVBAYTAk5aMREwDwYDVQQIDAhBdWN
   rbGFuZDERMA8GA1UEBwwIQXVja2xhbmQxHDAaBgNVBAoME0FtYXpvbiBXZWIgU2
   VydmljZXMxEzARBgNVBAsMCkJZT0lQIERlbW8xEzARBgNVBAMMCkJZT0lQIERlb
   W8wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCfmacvDp0wZ0ceiXXc
   R/q27mHI/U5HKt7SST4X2eAqufR9wXkfNanAEskgAseyFypwEEQr4CJijI/5hp9
   prh+jsWHWwkFRoBRR9FBtwcU/45XDXLga7D3stsI5QesHVRwOaXUdprAnndaTug
   mDPkD0vrl475JWDSIm+PUxGWLy+60aBqiaZq35wU/x+wXlAqBXg4MZK2KoUu27k
   Yt2zhmy0S7Ky+oRfRJ9QbAiSu/RwhQbh5Mkp1ZnVIc7NqnhdeIW48QaYjhMlUEf
   xdaqYUinzz8KpjfADZ4Hvqj9jWZ/eXo/9b2rGlHWkJsbhr0VEUyAGu1bwkgcdww
   3A7NjOxQbAgMBAAGjUzBRMB0GA1UdDgQWBBStFyujN6SYBr2glHpGt0XGF7GbGT
   AfBgNVHSMEGDAWgBStFyujN6SYBr2glHpGt0XGF7GbGTAPBgNVHRMBAf8EBTADA
   QH/MA0GCSqGSIb3DQEBCwUAA4IBAQBX6nn6YLhz521lfyVfxY0t6o3410bQAeAF
   08ud+ICtmQ4IO4A4B7zV3zIVYr0clrOOaFyLxngwMYN0XY5tVhDQqk4/gmDNEKS
   ```

```
Zy2QkX4Eg0YUWVzOyt6fPzjOvJLcsqc1hcF9wySL507XQz76Uk5cFypBOzbnk35
UkWrzA9KK97cXckfIESgK/k1N4ecwxwG6VQ8mBGqVpPpey+dXpzzzv1iBKN/VY4
ydjgH/LBfdTsVarmmy2vtWBxwrqkFvpdhSGCvRDl/qdO/GIDJi77dmZWkh/ic90
MNk1f38gs1jrCj8lThoar17Uo9y/Q5qJIsoNPyQrJRzqFU9F3FBjiPJF
-----END CERTIFICATE-----
```

RIPE의 경우 `whois -r -h whois.ripe.net` *`2001:0DB8:7269::/48`*을 사용하여 주소 범위에 대한 RDAP 레코드를 조회합니다. 명령 출력에서 `inetnum` 객체(네트워크 범위)에 대한 `descr` 섹션을 확인합니다. 인증서는 주소 범위에 대한 새 `descr` 필드로 추가되어야 합니다.

다음 명령을 사용하여 인증서가 포함된 `descr`를 검사할 수 있습니다.

```
whois -r -h whois.ripe.net 2001:0DB8:7269::/48 | grep descr | grep BEGIN
```

그러면 다음과 유사한 키 내용이 포함된 출력이 반환됩니다.

```
descr:
-----BEGIN CERTIFICATE-----MIID1zCCAr+gAwIBAgIUBkRPNSLrPqbRAFP8
RDAHSP+I1TowDQYJKoZIhvcNAQELBQAwezELMAkGA1UEBhMCTloxETAPBgNVBAg
MCEF1Y2tsYW5kMREwDwYDVQQHDAhBdWNrbGFuZDEcMBoGA1UECgwTQW1hem9uIF
dlYiBTZXJ2aWNlczETMBEGA1UECwwKQllPSVAgRGVtbzETMBEGA1UEAwwKQllPS
VAgRGVtbzAeFw0yMTEyMDcyMDI0NTRaFw0yMjEyMDcyMDI0NTRaMHsxCzAJBgNV
BAYTAk5aMREwDwYDVQQIDAhBdWNrbGFuZDERMA8GA1UEBwwIQXVja2xhbmQxHDA
aBgNVBAoME0FtYXpvbiBXZWIgU2VydmljZXMxEzARBgNVBAsMCkJZT0lQIERlbW
8xEzARBgNVBAMMCkJZT0lQIERlbW8wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwg
gEKAoIBAQCfmacvDp0wZ0ceiXXcR/q27mHI/U5HKt7SST4X2eAqufR9wXkfNanA
EskgAseyFypwEEQr4CJijI/5hp9prh+jsWHWwkFRoBRR9FBtwcU/45XDXLga7D3
stsI5QesHVRwOaXUdprAnndaTugmDPkD0vrl475JWDSIm+PUxGWLy+60aBqiaZq
35wU/x+wXlAqBXg4MZK2KoUu27kYt2zhmy0S7Ky+oRfRJ9QbAiSu/RwhQbh5Mkp
1ZnVIc7NqnhdeIW48QaYjhMlUEfxdaqYUinzz8KpjfADZ4Hvqj9jWZ/eXo/9b2r
GlHWkJsbhr0VEUyAGu1bwkgcdww3A7NjOxQbAgMBAAGjUzBRMB0GA1UdDgQWBBS
tFyujN6SYBr2glHpGt0XGF7GbGTAfBgNVHSMEGDAWgBStFyujN6SYBr2glHpGt0
XGF7GbGTAPBgNVHRMBAf8EBTADAQH/MA0GCSqGSIb3DQEBCwUAA4IBAQBX6nn6Y
Lhz521lfyVfxY0t6o3410bQAeAF08ud+ICtmQ4IO4A4B7zV3zIVYr0clrOOaFyL
xngwMYN0XY5tVhDQqk4/gmDNEKSZy2QkX4Eg0YUWVzOyt6fPzjOvJLcsqc1hcF9
wySL507XQz76Uk5cFypBOzbnk35UkWrzA9KK97cXckfIESgK/k1N4ecwxwG6VQ8
mBGqVpPpey+dXpzzzv1iBKN/VY4ydjgH/LBfdTsVarmmy2vtWBxwrqkFvpdhSGC
vRDl/qdO/GIDJi77dmZWkh/ic90MNk1f38gs1jrCj8lThoar17Uo9y/Q5qJIsoN
PyQrJRzqFU9F3FBjiPJF
-----END CERTIFICATE-----
```

APNIC의 경우 `whois -h whois.apnic.net` *2001:0DB8:6170::/48*을 사용하여 BYOIP 주소 범위에 대한 RDAP 레코드를 조회합니다. 명령 출력에서 `inetnum` 객체(네트워크 범위)에 대한 `remarks` 섹션을 확인합니다. 인증서는 주소 범위에 대한 새 `remarks` 필드로 추가되어야 합니다.

다음 명령을 사용하여 인증서가 포함된 `remarks`를 검사할 수 있습니다.

```
whois -h whois.apnic.net 2001:0DB8:6170::/48 | grep remarks | grep BEGIN
```

그러면 다음과 유사한 키 내용이 포함된 출력이 반환됩니다.

```
remarks:
-----BEGIN CERTIFICATE-----
MIID1zCCAr+gAwIBAgIUBkRPNSLrPqbRAFP8RDAHSP+I1TowDQYJKoZIhvcNAQE
LBQAwezELMAkGA1UEBhMCTloxETAPBgNVBAgMCEF1Y2tsYW5kMREwDwYDVQQHDA
hBdWNrbGFuZDEcMBoGA1UECgwTQW1hem9uIFdlYiBTZXJ2aWNlczETMBEGA1UEC
wwKQllPSVAgRGVtbzETMBEGA1UEAwwKQllPSVAgRGVtbzAeFw0yMTEyMDcyMDI0
NTRaFw0yMjEyMDcyMDI0NTRaMHsxCzAJBgNVBAYTAk5aMREwDwYDVQQIDAhBdWN
rbGFuZDERMA8GA1UEBwwIQXVja2xhbmQxHDAaBgNVBAoME0FtYXpvbiBXZWIgU2
VydmljZXMxEzARBgNVBAsMCkJZT0lQIERlbW8xEzARBgNVBAMMCkJZT0lQIERlb
W8wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCfmacvDp0wZ0ceiXXc
R/q27mHI/U5HKt7SST4X2eAqufR9wXkfNanAEskgAseyFypwEEQr4CJijI/5hp9
prh+jsWHWwkFRoBRR9FBtwcU/45XDXLga7D3stsI5QesHVRwOaXUdprAnndaTug
mDPkD0vrl475JWDSIm+PUxGWLy+60aBqiaZq35wU/x+wXlAqBXg4MZK2KoUu27k
Yt2zhmy0S7Ky+oRfRJ9QbAiSu/RwhQbh5Mkp1ZnVIc7NqnhdeIW48QaYjhMlUEf
xdaqYUinzz8KpjfADZ4Hvqj9jWZ/eXo/9b2rGlHWkJsbhr0VEUyAGu1bwkgcdww
3A7NjOxQbAgMBAAGjUzBRMB0GA1UdDgQWBBStFyujN6SYBr2glHpGt0XGF7GbGT
AfBgNVHSMEGDAWgBStFyujN6SYBr2glHpGt0XGF7GbGTAPBgNVHRMBAf8EBTADA
QH/MA0GCSqGSIb3DQEBCwUAA4IBAQBX6nn6YLhz521lfyVfxY0t6o3410bQAeAF
08ud+ICtmQ4IO4A4B7zV3zIVYr0clrOOaFyLxngwMYN0XY5tVhDQqk4/gmDNEKS
Zy2QkX4Eg0YUWVzOyt6fPzjOvJLcsqc1hcF9wySL507XQz76Uk5cFypBOzbnk35
UkWrzA9KK97cXckfIESgK/k1N4ecwxwG6VQ8mBGqVpPpey+dXpzzzv1iBKN/VY4
ydjgH/LBfdTsVarmmy2vtWBxwrqkFvpdhSGCvRDl/qdO/GIDJi77dmZWkh/ic90
MNk1f38gs1jrCj8lThoar17Uo9y/Q5qJIsoNPyQrJRzqFU9F3FBjiPJF
-----END CERTIFICATE-----
```

2. ROA 객체 생성 검증

RIPEstat Data API를 사용하여 ROA 객체가 성공적으로 생성되었는지 확인합니다. Amazon ASN 16509 및 14618과 현재 주소 범위를 알리도록 승인된 ASN에 대해 주소 범위를 테스트해야 합니다.

다음 명령을 사용하여 주소 범위가 있는 여러 Amazon ASN의 ROA 객체를 검사할 수 있습니다.

```
curl --location --request GET "https://stat.ripe.net/data/rpki-validation/data.json?
resource=ASN&prefix=CIDR
```

이 예제 출력에서 응답의 결과는 Amazon ASN 16509에 대해 `"status": "valid"`입니다. 이는 주소 범위에 대한 ROA 객체가 성공적으로 생성되었음을 나타냅니다.


```
{
    "messages": [],
    "see_also": [],
    "version": "0.3",
    "data_call_name": "rpki-validation",
    "data_call_status": "supported",
    "cached": false,
    "data": {
        "validating_roas": [
            {
                "origin": "16509",
                "prefix": "2001:0DB8::/32",
                "max_length": 48,
                "validity": "valid"
            },
            {
                "origin": "14618",
                "prefix": "2001:0DB8::/32",
                "max_length": 48,
                "validity": "invalid_asn"
            },
            {
                "origin": "64496",
                "prefix": "2001:0DB8::/32",
                "max_length": 48,
                "validity": "invalid_asn"
            }
        ],
        "status": "valid",
        "validator": "routinator",
        "resource": "16509",
        "prefix": "2001:0DB8::/32"
    },
    "query_id": "20230224152430-81e6384e-21ba-4a86-852a-31850787105f",
    "process_time": 58,
    "server_id": "app116",
```

```
    "build_version": "live.2023.2.1.142",
    "status": "ok",
    "status_code": 200,
    "time": "2023-02-24T15:24:30.773654"
}
```

"unknown" 상태는 주소 범위에 대한 ROA 객체가 생성되지 않았음을 나타냅니다. "invalid_asn" 상태는 주소 범위에 대한 ROA 객체가 성공적으로 생성되지 않았음을 나타냅니다.

Amazon EC2에서 BYOIP 주소 범위 사용

계정에서 프로비저닝한 IPv4 및 IPv6 주소 범위를 보고 사용할 수 있습니다. 자세한 내용은 the section called "주소 범위 온보딩" 섹션을 참조하세요.

IPv4 주소 범위

IPv4 주소 풀에서 탄력적 IP 주소를 생성하여 AWS 리소스(예: EC2 인스턴스, NAT 게이트웨이, Network Load Balancer)와 함께 사용할 수 있습니다.

계정에서 프로비저닝한 IPv4 주소 풀에 대한 정보를 보려면 다음 describe-public-ipv4-pools 명령을 사용합니다.

```
aws ec2 describe-public-ipv4-pools --region us-east-1
```

IPv4 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.

IPv6 주소 범위

계정에서 프로비저닝한 IPv6 주소 풀에 대한 정보를 보려면 다음 describe-ipv6-pools 명령을 사용합니다.

```
aws ec2 describe-ipv6-pools --region us-east-1
```

VPC를 생성하고 IPv6 주소 풀에서 IPv6 CIDR을 지정하려면 다음 create-vpc 명령을 사용합니다. Amazon이 IPv6 주소 풀에서 IPv6 CIDR을 선택하도록 하려면 `--ipv6-cidr-block` 옵션을 생략합니다.

```
aws ec2 create-vpc --cidr-block 10.0.0.0/16 --ipv6-cidr-block ipv6-cidr --ipv6-
pool pool-id --region us-east-1
```

IPv6 주소 풀의 IPv6 CIDR 블록을 VPC와 연결하려면 다음 associate-vpc-cidr-block 명령을 사용합니다. Amazon이 IPv6 주소 풀에서 IPv6 CIDR을 선택하도록 하려면 `--ipv6-cidr-block` 옵션을 생략합니다.

```
aws ec2 associate-vpc-cidr-block --vpc-id vpc-123456789abc123ab --ipv6-cidr-block ipv6-
cidr --ipv6-pool pool-id --region us-east-1
```

VPC 및 연결된 IPv6 주소 풀에 대한 정보를 보려면 describe-vpcs 명령을 사용합니다. 특정 IPv6 주소 풀에서 연결된 IPv6 CIDR 블록에 대한 정보를 보려면 다음 get-associated-ipv6-pool-cidrs 명령을 사용합니다.

```
aws ec2 get-associated-ipv6-pool-cidrs --pool-id pool-id --region us-east-1
```

VPC에서 IPv6 CIDR 블록의 연결을 해제하면 해당 블록이 IPv6 주소 풀로 다시 해제됩니다.

탄력적 IP 주소

탄력적 IP 주소는 동적 클라우드 컴퓨팅을 위해 고안된 정적 IPv4 주소입니다. 탄력적 IP 주소는 AWS 계정에 할당되며 릴리스할 때까지 할당된 상태로 유지됩니다. 탄력적 IP 주소를 사용하면 주소를 계정의 다른 인스턴스에 신속하게 다시 매핑하여 인스턴스나 소프트웨어의 오류를 마스킹할 수 있습니다. 또는 도메인이 인스턴스를 가리키도록 도메인에 대한 DNS 레코드에 탄력적 IP 주소를 지정할 수 있습니다. 자세한 내용은 도메인 등록자에 대한 문서를 참조하세요.

탄력적 IP 주소는 인터넷에서 연결 가능한 퍼블릭 IPv4 주소입니다. 퍼블릭 IPv4 주소가 없는 인스턴스에 연결해야 하는 경우 탄력적 IP 주소를 인스턴스에 연결하여 인터넷 통신을 활성화할 수 있습니다.

내용

- 탄력적 IP 주소 요금
- 탄력적 IP 주소 기본 사항
- 탄력적 IP 주소 할당량
- 인스턴스와 탄력적 IP 주소 연결
- AWS 계정 간에 탄력적 IP 주소 전송
- 탄력적 IP 주소 릴리스
- Amazon EC2에서 이메일에 대한 역방향 DNS 레코드 생성

탄력적 IP 주소 요금

사용 중(EC2 인스턴스와 같은 리소스에 할당됨)이거나 유휴 상태(계정에서 생성되었지만 할당되지 않음)이거나 관계없이 모든 탄력적 IP 주소에 대해 요금이 부과됩니다.

AWS에서는 탄력적 IP 주소 및 실행 중인 인스턴스에 연결된 퍼블릭 IPv4 주소를 포함하여 모든 퍼블릭 IPv4 주소에 요금을 부과합니다. 자세한 내용은 Amazon VPC 요금 페이지의 퍼블릭 IPv4 주소 탭을 참조하세요.

탄력적 IP 주소 기본 사항

탄력적 IP 주소의 기본 특성은 다음과 같습니다.

- 탄력적 IP 주소는 정적이며 시간이 지남에 따라 변경되지 않습니다.
- 탄력적 IP 주소는 특정 리전에서만 사용할 수 있으며 다른 리전으로 이전할 수 없습니다.
- IPv4 주소의 Amazon 풀 또는 AWS 계정으로 가져온 사용자 지정 IPv4 주소 풀에서 탄력적 IP 주소를 할당할 수 있습니다. IPv6에 대한 탄력적 IP 주소는 지원하지 않습니다.
- 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 연결할 때 인스턴스에 이미 연결된 퍼블릭 IPv4 주소가 있는 경우 해당 퍼블릭 IPv4 주소는 Amazon의 퍼블릭 IPv4 주소 풀로 다시 해제되고 대신 탄력적 IP 주소가 인스턴스와 연결됩니다. 이전에 인스턴스와 연결된 퍼블릭 IPv4 주소를 재사용할 수 없으며 해당 퍼블릭 IPv4 주소를 탄력적 IP 주소로 변환할 수 없습니다. 자세한 내용은 퍼블릭 IPv4 주소 섹션을 참조하세요.
- 탄력적 IP 주소는 리소스에서 연결 해제했다가 다른 리소스와 다시 연결할 수 있습니다. 예기치 않은 동작을 방지하려면 변경하기 전에 기존 연결에 이름이 지정된 리소스에 대한 모든 활성 연결이 닫혀 있는지 확인합니다. 탄력적 IP 주소를 다른 리소스에 연결한 후 새로 연결된 리소스에 대한 연결을 다시 열 수 있습니다.
- 연결 해제한 탄력적 IP 주소는 명시적으로 릴리스할 때까지 계정에 할당되어 있습니다. 인스턴스와의 연결 여부와 관계없이 계정의 모든 탄력적 IP 주소에 대해 요금이 부과됩니다. 자세한 내용은 Amazon VPC 요금 페이지의 퍼블릭 IPv4 주소 탭을 참조하세요.
- 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. 자세한 내용은 탄력적 IP 주소 할당량 섹션을 참조하세요.
- 탄력적 IP 주소를 할당할 때 탄력적 IP 주소를 네트워크 경계 그룹에 연결할 수 있습니다. 이는 CIDR 블록을 공고하는 위치입니다. 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 주소는 특정 네트워크 경계 그룹에서만 사용할 수 있습니다.

탄력적 IP 주소 할당량

퍼블릭(IPv4) 인터넷 주소는 흔치 않은 퍼블릭 리소스이기 때문에 기본적으로 모든 AWS 계정은 리전당 5개의 탄력적 IP 주소가 할당됩니다. 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.

아키텍처에서 추가 탄력적 IP 주소를 보증한다고 생각되면 Service Quotas 콘솔에서 직접 할당량 증가를 요청할 수 있습니다. 할당량 증가를 요청하려면 계정 수준에서 증가 요청을 선택합니다. 자세한 내용은 Amazon EC2 서비스 할당량 섹션을 참조하세요.

인스턴스와 탄력적 IP 주소 연결

탄력적 IP 주소를 할당한 후 EC2 인스턴스, NAT 게이트웨이 또는 Network Load Balancer와 같은 AWS 리소스와 연결할 수 있습니다. 나중에 탄력적 IP 주소를 다른 AWS 리소스와 연결하려면 현재 리소스에서 연결을 해제한 다음 새 리소스와 연결하면 됩니다.

업무

- 탄력적 IP 주소 할당
- 탄력적 IP 주소 연결
- 탄력적 IP 주소 연결 해제

탄력적 IP 주소 할당

리전에서 사용할 탄력적 IP 주소를 할당할 수 있습니다. 사용 중(EC2 인스턴스와 같은 리소스에 연결됨)이거나 유휴 상태(계정에서 생성되었지만 연결되지 않음)이거나 관계없이 모든 탄력적 IP 주소에 대해 요금이 부과됩니다.

Console

탄력적 IP 주소 할당

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [네트워크 및 보안], [탄력적 IP]를 선택합니다.
3. Allocate Elastic IP address(탄력적 IP 주소 할당)를 선택합니다.
4. (선택 사항) 탄력적 IP 주소(EIP)를 할당할 때는 EIP를 할당할 네트워크 경계 그룹을 선택합니다. 네트워크 경계 그룹은 AWS가 퍼블릭 IP 주소를 알리는 가용 영역, Local Zone 또는 Wavelength Zone의 집합입니다. Local Zone 및 Wavelength Zone은 AWS 네트워크와 해당 영역의 리소스에 액세스하는 고객 간의 지연 시간 또는 물리적 거리를 최소화하기 위해 리전의 AZ와 다른 네트워크 경계 그룹을 가질 수 있습니다.

   > **Important**
   >
   > EIP와 연결될 AWS 리소스와 동일한 네트워크 경계 그룹에 EIP를 할당해야 합니다. 한 네트워크 경계 그룹의 EIP는 해당 네트워크 경계 그룹의 영역에서만 알릴 수 있으며 다른 네트워크 경계 그룹이 나타내는 다른 영역에서는 알릴 수 없습니다.

   Local Zone 또는 Wavelength Zone을 활성화한 경우(자세한 내용은 Local Zone 활성화 또는 Wavelength Zone 활성화 참조) AZ, Local Zone 또는 Wavelength Zone에 대한 네트워크 경계 그룹을 선택할 수 있습니다. EIP와 연결된 AWS 리소스는 동일한 네트워크 경계 그룹에 있어야 하므로 네트워크 경계 그룹을 신중하게 선택하세요. EC2 콘솔을 사용하여 가용 영역, Local Zones 또는 Wavelength Zones가 속해 있는 네트워크 경계 그룹을 볼 수 있습니다. 일반적으로 한 리전의 모든 가용 영역은 동일한 네트워크 경계 그룹에 속하지만 Local Zone 또는 Wavelength Zone은 별도의 자체 네트워크 경계 그룹에 속합니다.

   Local Zone 또는 Wavelength Zone이 활성화되지 않은 경우 EIP를 할당하면 해당 리전의 모든 AZ를 나타내는 네트워크 경계 그룹(예:`us-west-2`)이 미리 정의되어 있으며 변경할 수 없습니다. 즉, 이 네트워크 경계 그룹에 할당한 EIP는 현재 속한 리전의 모든 AZ에 광고됩니다.
5. 퍼블릭 IPv4 주소 풀에서 다음 중 하나를 선택합니다.

- [Amazon의 IP 주소 풀(Amazon's pool of IPv4 addresses)]—IPv4 주소를 Amazon의 IPv4 주소 풀에서 할당하려는 경우.
- AWS 계정으로 가져오는 퍼블릭 IPv4 주소-AWS 계정으로 가져온 IP 주소 풀에서 불연속적(비순차적) 퍼블릭 IPv4 주소를 할당하려는 경우. IP 주소 풀이 없는 경우에는 이 옵션을 사용할 수 없습니다. AWS 계정으로 고유한 IP 주소 범위 가져오기에 대한 자세한 내용은 Amazon EC2로 고유 IP 주소 가져오기(BYOIP) 섹션을 참조하세요.
- [고객 소유 IPv4 주소 풀(Customer owned pool of IPv4 addresses)] - AWS Outpost를 통해 사용할 온프레미스 네트워크에서 생성된 풀에서 IPv4 주소를 할당하려는 경우. AWS Outposts가 없는 경우 이 옵션이 비활성화됩니다.
- IPAM IPv4 풀을 사용하여 할당: IPAM 풀의 연속적 퍼블릭 IPv4 블록에서 순차적 탄력적 IP 주소를 할당하려는 경우. 순차적 탄력적 IP 주소를 할당하면 보안 액세스 제어 목록에 대한 관리 오버헤드를 크게 줄이고 AWS의 규모를 조정하는 기업의 IP 주소 할당 및 추적을 단순화할 수 있습니다. 자세한 내용은 Amazon VPC IPAM 사용 설명서의 IPAM 풀에서 순차적 탄력적 IP 주소 할당을 참조하세요.

6. (선택 사항) 태그를 추가하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.

AWS CLI

탄력적 IP 주소 할당

allocate-address AWS CLI 명령을 사용합니다.

```
aws ec2 allocate-address
```

PowerShell

탄력적 IP 주소 할당

New-EC2Address cmdlet을 사용합니다.

```
New-EC2Address -Domain Vpc
```

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 게이트웨이를 사용하여 VPC 인터넷 액세스 활성화를 참조하세요.

Console

인스턴스와 엘라스틱 IP 주소를 연결하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Elastic IPs]를 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. 연결(Associate)을 선택합니다.

네트워크 인터페이스와 탄력적 IP 주소를 연결하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Elastic IPs]를 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. 연결(Associate)을 선택합니다.

AWS CLI

탄력적 IP 주소를 연결하려면

associate-address AWS CLI 명령을 사용합니다.

```
aws ec2 associate-address --instance-id i-0b263919b6498b123 --allocation-
id eipalloc-64d5890a
```

PowerShell

탄력적 IP 주소를 연결하려면

Register-EC2Address cmdlet을 사용합니다.

```
Register-EC2Address `
    -InstanceId i-0b263919b6498b123 `
    -AllocationId eipalloc-64d5890a
```

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 연결할 수 있습니다.

Console

탄력적 IP 주소의 연결을 해제하고 다시 연결하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Elastic IPs]를 선택합니다.
3. 연결을 해제할 탄력적 IP 주소를 선택하고 작업, 탄력적 IP 주소 연결 해제를 선택합니다.
4. 연결 해제를 선택합니다.

AWS CLI

엘라스틱 IP 주소를 연결 해제하려면

disassociate-address AWS CLI 명령을 사용합니다.

```
aws ec2 disassociate-address --association-id eipassoc-12345678
```

PowerShell

탄력적 IP 주소를 연결 해제하려면

Unregister-EC2Address cmdlet을 사용합니다.

```
Unregister-EC2Address -AssociationId eipassoc-12345678
```

AWS 계정 간에 탄력적 IP 주소 전송

AWS 계정 간에 탄력적 IP 주소를 전송할 수 있습니다. 이는 다음과 같은 상황에서 도움이 될 수 있습니다.

- 재해 복구 - 비상 상황 발생 시 대중에게 공개된 인터넷 워크로드에 대한 IP 주소를 빠르게 다시 매핑합니다.
- 조직 구조 조정 - AWS 계정 간에 워크로드를 이동합니다. 주소 전송을 이용하면 보안 그룹과 네트워크 ACL에서 새로운 탄력적 IP 주소를 허용할 때까지 기다릴 필요가 없습니다.
- 중앙 집중식 보안 관리 - 중앙 집중식 AWS 보안 계정을 사용하여 보안 규정 준수를 위해 검증된 탄력적 IP 주소를 추적하고 전송합니다.

요금

탄력적 IP 주소 전송에는 요금이 부과되지 않습니다.

업무

- 탄력적 IP 주소 전송 활성화
- 전송된 탄력적 IP 주소 수락
- 탄력적 IP 주소 전송 비활성화

탄력적 IP 주소 전송 활성화

이 섹션에서는 전송된 탄력적 IP 주소를 수락하는 방법을 설명합니다. 전송을 위한 탄력적 IP 주소 활성화와 관련해 다음과 같은 제한 사항에 유의하세요.

- 모든 AWS 계정(소스 계정)의 탄력적 IP 주소를 동일한 AWS 리전의 다른 AWS 계정(전송 계정)으로 전송할 수 있습니다. 탄력적 IP 주소를 다른 리전으로 전송할 수 없습니다.
- 탄력적 IP 주소를 전송할 때 AWS 계정 간에 2단계 핸드셰이크가 발생합니다. 소스 계정에서 전송이 시작되면 전송 계정은 7일 내에 탄력적 IP 주소 전송을 수락해야 합니다. 이 7일 동안 소스 계정은 대기 중인 전송을 볼 수 있습니다(예: AWS 콘솔에서 보거나 describe-address-transfers 명령 사용). 7일이 지나면 전송이 만료되고 탄력적 IP 주소의 소유권이 소스 계정으로 반환됩니다.
- 수락된 전송은 전송이 수락된 후 14일 동안 소스 계정에서 볼 수 있습니다(예: AWS 콘솔에서 보거나 describe-address-transfers 명령 사용).

- AWS에서는 보류 중인 탄력적 IP 주소 전송 요청에 대해 전송 계정에 알리지 않습니다. 소스 계정의 소유자는 반드시 수락해야 하는 탄력적 IP 주소 전송 요청이 있음을 전송 계정 소유자에게 알려야 합니다.
- 전송되는 탄력적 IP 주소와 연결된 모든 태그는 전송이 완료된 후에 재설정됩니다.
- AWS 계정에 가져온 퍼블릭 IPv4 주소 풀(일반적으로 고유 IP 주소 가져오기(BYOIP) 주소 풀이라고 함)에서 할당된 탄력적 IP 주소는 전송할 수 없습니다.
- Amazon에서 제공하는 연속적 퍼블릭 IPv4 Amazon VPC IP 주소 관리자(IPAM) 풀에서 할당된 탄력적 IP 주소는 전송할 수 없습니다. 대신 IPAM을 사용하면 IPAM을 AWS 조직과 통합하고 AWS RAM을 사용하여 AWS 계정 간에 IPAM 풀을 공유할 수 있습니다. 자세한 내용은 Amazon VPC IPAM 사용 설명서의 IPAM 풀에서 순차적 탄력적 IP 주소 할당을 참조하세요.
- 역방향 DNS 레코드가 연결되어 있는 탄력적 IP 주소를 전송하려고 할 경우 전송 프로세스를 시작할 수 있지만, 연결된 DNS 레코드가 제거될 때까지 전송 계정이 전송을 수락할 수 없습니다.
- AWS Outposts를 활성화하고 구성한 경우 고객 소유의 IP 주소 풀(COIP)에서 탄력적 IP 주소를 할당했을 수 있습니다. CoIP에서 할당된 탄력적 IP 주소는 전송할 수 없습니다. 하지만 AWS RAM을 사용하여 다른 계정과 CoIP를 공유할 수 있습니다. 자세한 내용은 AWS Outposts 사용 설명서에서 고객 소유 IP 주소를 참조하세요.
- Amazon VPC IPAM을 사용하여 AWS Organizations의 조직 내 계정으로 탄력적 IP 주소 전송을 추적할 수 있습니다. 자세한 내용은 IP 주소 기록 보기를 참조하세요. 그러나 탄력적 IP 주소가 조직 외부의 AWS 계정으로 전송되는 경우 탄력적 IP 주소에 대한 IPAM 감사 기록은 손실됩니다.

소스 계정으로 이 단계를 수행해야 합니다.

Console

탄력적 IP 주소 전송 활성화

1. 소스 AWS 계정을 사용하고 있는지 확인합니다.
2. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
3. 탐색 창에서 Elastic IPs를 선택합니다.
4. 전송을 활성화할 탄력적 IP 주소를 하나 이상 선택하고 Actions(작업), Enable transfer(전송 활성화)를 선택합니다.
5. 탄력적 IP 주소를 여러 개 전송하는 경우 전송 유형(Transfer type) 옵션이 표시됩니다. 다음 옵션 중 하나를 선택하세요.

- 탄력적 IP 주소를 단일 AWS 계정으로 전송하려는 경우 Single account(단일 계정)를 선택합니다.
- 탄력적 IP 주소를 여러 AWS 계정으로 전송하려는 경우 Multiple accounts(다중 계정)를 선택합니다.

6. Transfer account ID(전송 계정 ID)에 탄력적 IP 주소를 전송하려는 AWS 계정의 ID를 입력합니다.
7. 텍스트 상자에 **enable**을 입력하여 전송을 확인합니다.
8. 제출을 선택합니다.
9. 전송을 수락하려면 전송된 탄력적 IP 주소 수락 섹션을 참조하세요. 전송을 비활성화하려면 탄력적 IP 주소 전송 비활성화 섹션을 참조하세요.

AWS CLI

탄력적 IP 주소 전송 활성화

enable-address-transfer 명령을 사용합니다.

```
aws ec2 enable-address-transfer \
    --allocation-id eipalloc-09ad461b0d03f6aaf \
    --transfer-account-id 123456789012
```

PowerShell

탄력적 IP 주소 전송 활성화

Enable-EC2AddressTransfer cmdlet을 사용합니다.

```
Enable-EC2AddressTransfer `
    -AllocationId eipalloc-09ad461b0d03f6aaf `
    -TransferAccountId 123456789012
```

전송된 탄력적 IP 주소 수락

이 섹션에서는 전송된 탄력적 IP 주소를 수락하는 방법을 설명합니다.

탄력적 IP 주소를 전송할 때 AWS 계정 간에 2단계 핸드셰이크가 발생합니다. 소스 계정에서 전송이 시작되면 전송 계정은 7일 내에 탄력적 IP 주소 전송을 수락해야 합니다. 이 7일 동안 소스 계정은 대기

중인 전송을 볼 수 있습니다(예: AWS 콘솔에서 보거나 describe-address-transfers 명령 사용). 7일이 지나면 전송이 만료되고 탄력적 IP 주소의 소유권이 소스 계정으로 반환됩니다.

전송을 수락할 때 발생할 수 있는 다음 예외 사항과 이에 대한 해결 방법을 참고하세요.

- AddressLimitExceeded: 전송 계정이 탄력적 IP 주소 할당량을 초과한 경우, 소스 계정에서 탄력적 IP 주소 전송을 활성화할 수 있지만 전송 계정이 전송을 수락하려고 하면 이 예외가 발생합니다. 기본적으로 모든 AWS 계정은 리전당 탄력적 IP 주소 5개로 제한됩니다. 한도 증가에 대한 지침은 탄력적 IP 주소 할당량 섹션을 참조하세요.
- InvalidTransfer.addressCustomPtrSet: 귀하 또는 조직 내 다른 사용자가 귀하가 전송하려는 탄력적 IP 주소를 역방향 DNS 조회를 사용하도록 구성한 경우, 소스 계정에서 탄력적 IP 주소 전송을 활성화할 수 있지만 전송 계정이 전송을 수락하려고 하면 이 예외가 발생합니다. 이 문제를 해결하기 위해서는 소스 계정에서 탄력적 IP 주소의 DNS 레코드를 제거해야 합니다. 자세한 내용은 Amazon EC2에서 이메일에 대한 역방향 DNS 레코드 생성 섹션을 참조하세요.
- InvalidTransfer.addressAssociated: 탄력적 IP 주소가 ENI 또는 EC2 인스턴스와 연결된 경우, 소스 계정에서 탄력적 IP 주소 전송을 활성화할 수 있지만 전송 계정이 전송을 수락하려고 하면 이 예외가 발생합니다. 이 문제를 해결하기 위해서는 탄력적 IP 주소의 연결을 해제해야 합니다. 자세한 내용은 탄력적 IP 주소 연결 해제 섹션을 참조하세요.

기타 예외 사항은 지원에 문의하세요.

전송 계정으로 이 단계를 수행해야 합니다.

Console

탄력적 IP 주소 전송 수락

1. 전송 계정을 사용 중인지 확인합니다.
2. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
3. 탐색 창에서 탄력적 IP(Elastic IPs)를 선택합니다.
4. Actions(작업), Accept transfer(전송 수락)를 선택합니다.
5. 전송을 수락하면 전송 중인 탄력적 IP 주소와 연결된 태그가 탄력적 IP 주소로 전송되지 않습니다. 수락하려는 탄력적 IP 주소의 Name(이름) 태그를 정의하려는 경우 Create a tag with a key of 'Name' and a value that you specify('Name' 키와 지정한 값으로 태그 생성)를 선택합니다.
6. 전송할 탄력적 IP 주소를 입력합니다.
7. 전송된 탄력적 IP 주소를 여러 개 수락 중인 경우 Add address(주소 추가)를 선택하여 추가 탄력적 IP 주소를 입력합니다.

8. 제출을 선택합니다.

AWS CLI

탄력적 IP 주소 전송 수락

accept-address-transfer 명령을 사용합니다.

```
aws ec2 accept-address-transfer --address 100.21.184.216
```

PowerShell

탄력적 IP 주소 전송 수락

Approve-EC2AddressTransfer cmdlet을 사용합니다.

```
Approve-EC2AddressTransfer -Address 100.21.184.216
```

탄력적 IP 주소 전송 비활성화

이 섹션에서는 탄력적 IP 전송을 활성화한 후 이를 비활성화하는 방법을 설명합니다.

전송을 활성화한 소스 계정으로 다음 단계를 수행해야 합니다.

Console

탄력적 IP 주소 전송 비활성화

1. 소스 AWS 계정을 사용하고 있는지 확인합니다.
2. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
3. 탐색 창에서 탄력적 IP(Elastic IPs)를 선택합니다.
4. 탄력적 IP의 리소스 목록에서 Transfer status(전송 상태) 열을 표시하는 속성이 활성화되어 있는지 확인합니다.
5. Transfer status(전송 상태)가 Pending(보류 중)인 탄력적 IP 주소를 하나 이상 선택하고 Actions(작업), Disable transfer(전송 비활성화)를 선택합니다.
6. 텍스트 상자에 **disable**을 입력하여 확인합니다.
7. 제출을 선택합니다.

AWS CLI

탄력적 IP 주소 전송 비활성화

disable-address-transfer 명령을 사용합니다.

```
aws ec2 disable-address-transfer --allocation-id eipalloc-09ad461b0d03f6aaf
```

PowerShell

탄력적 IP 주소 전송 비활성화

Disable-EC2AddressTransfer cmdlet을 사용합니다.

```
Disable-EC2AddressTransfer -AllocationId eipalloc-09ad461b0d03f6aaf
```

탄력적 IP 주소 릴리스

탄력적 IP 주소가 더 이상 필요하지 않으면 해당 주소를 릴리스하는 것이 좋습니다. 해제할 탄력적 IP 주소가 AWS 리소스와 현재 연결되어 있지 않아야 합니다.

Console

탄력적 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 선택한 후, 작업에서 탄력적 IP 주소 릴리스를 선택합니다
4. 릴리스를 선택합니다.

AWS CLI

탄력적 IP 주소를 해제합니다

release-address AWS CLI 명령을 사용합니다.

```
aws ec2 release-address --allocation-id eipalloc-64d5890a
```

PowerShell

탄력적 IP 주소를 해제합니다

Remove-EC2Address cmdlet을 사용합니다.

```
Remove-EC2Address -AllocationId eipalloc-64d5890a
```

탄력적 IP 주소를 해제한 후 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 없습니다.

AWS CLI

탄력적 IP 주소를 복구하려면

allocate-address 명령을 사용합니다.

```
aws ec2 allocate-address \
    --domain vpc \
    --address 203.0.113.3
```

PowerShell

탄력적 IP 주소를 복구하려면

New-EC2Address cmdlet을 사용합니다.

```
New-EC2Address `
    -Address 203.0.113.3 `
    -Domain vpc `
    -Region us-east-1
```

Amazon EC2에서 이메일에 대한 역방향 DNS 레코드 생성

EC2 인스턴스에서 제3자에게 이메일을 보내려는 경우 하나 이상의 탄력적 IP 주소를 프로비저닝하고 이메일을 보내는 데 사용하는 탄력적 IP 주소에 정적 역방향 DNS 레코드를 할당하는 것이 좋습니

다. 이렇게 하면 일부 스팸 방지 조직에서 이메일에 스팸 플래그를 지정하는 것을 방지할 수 있습니다. AWS는 ISP 및 인터넷 스팸 방지 조직과 협력하여 이러한 주소에서 보내는 이메일에 스팸으로 플래그가 지정될 가능성을 줄이고 있습니다.

고려 사항

- 역방향 DNS 레코드를 생성하기 전에 탄력적 IP 주소를 가리키는 정방향 DNS 레코드(레코드 유형 A)를 설정해야 합니다.
- 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.
- 지원에 연락하여 탄력적 IP 주소의 역방향 DNS를 설정한 경우 역방향 DNS를 제거할 수는 있지만 탄력적 IP 주소는 지원에 의해 잠겨 있어서 해제할 수 없습니다. 탄력적 IP 주소를 해제하려면 AWS Support에 문의하세요. 탄력적 IP 주소 잠금이 해제된 후에 탄력적 IP 주소를 해제할 수 있습니다.
- [AWS GovCloud (US) Region] 역방향 DNS 레코드를 생성할 수 없습니다. AWS가 정적 역방향 DNS 레코드를 할당해야 합니다. 지원 사례를 열어 역방향 DNS 및 이메일 전송 제한을 제거합니다. 탄력적 IP 주소와 역방향 DNS 레코드를 제공해야 합니다.

역방향 DNS 레코드 생성

다음과 같이 탄력적 IP 주소에 대한 역방향 DNS 레코드를 생성할 수 있습니다.

Console

역방향 DNS 레코드를 생성하려면 다음을 수행하세요.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [탄력적 IP(Elastic IPs)]를 선택합니다.
3. 탄력적 IP 주소를 선택하고 [작업], [역방향 DNS 업데이트]를 선택합니다.
4. 역방향 DNS 도메인 이름(Reverse DNS domain name)에 도메인 이름을 입력합니다.
5. **update**를 입력하여 확인합니다.
6. 업데이트(Update)를 선택합니다.

AWS CLI

역방향 DNS 레코드를 생성하려면 다음을 수행하세요.

modify-address-attribute 명령을 사용합니다.

```
aws ec2 modify-address-attribute \
    --allocation-id eipalloc-abcdef01234567890 \
    --domain-name example.com
```

출력의 예시는 다음과 같습니다.

```
{
    "Addresses": [
        {
            "PublicIp": "192.0.2.0",
            "AllocationId": "eipalloc-abcdef01234567890",
            "PtrRecord": "example.net.",
            "PtrRecordUpdate": {
                "Value": "example.com.",
                "Status": "PENDING"
            }
        }
    ]
}
```

PowerShell

역방향 DNS 레코드를 생성하려면 다음을 수행하세요.

Edit-EC2AddressAttribute cmdlet을 사용하세요.

```
Edit-EC2AddressAttribute `
    -AllocationId 'eipalloc-abcdef01234567890' `
    -DomainName 'example.com' |
Format-List `
    AllocationId, PtrRecord, PublicIp,
    @{Name='PtrRecordUpdate';Expression={$_.PtrRecordUpdate | Format-List | Out-
String}}
```

출력의 예시는 다음과 같습니다.

```
AllocationId    : eipalloc-abcdef01234567890
PtrRecord       : example.net.
PublicIp        : 192.0.2.0
PtrRecordUpdate :
                  Reason :
                  Status : PENDING
```

```
                    Value  : example.com.
```

역방향 DNS 레코드 제거

다음과 같이 탄력적 IP 주소에서 역방향 DNS 레코드를 제거할 수 있습니다.

다음과 같은 오류가 수신되면 지원을 위해 지원에 이메일 전송 제한 제거 요청을 제출할 수 있습니다.

```
The address cannot be released because it is locked to your account.
```

Console

역방향 DNS 레코드를 제거하려면 다음을 수행하세요.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [탄력적 IP(Elastic IPs)]를 선택합니다.
3. 탄력적 IP 주소를 선택하고 [작업], [역방향 DNS 업데이트]를 선택합니다.
4. 역방향 DNS 도메인 이름(Reverse DNS domain name)에서 도메인 이름을 지웁니다.
5. **update**를 입력하여 확인합니다.
6. 업데이트(Update)를 선택합니다.

AWS CLI

역방향 DNS 레코드를 제거하려면 다음을 수행하세요.

reset-address-attribute 명령을 사용합니다.

```
aws ec2 reset-address-attribute \
    --allocation-id eipalloc-abcdef01234567890 \
    --attribute domain-name
```

출력의 예시는 다음과 같습니다.

```
{
    "Addresses": [
        {
            "PublicIp": "192.0.2.0",
            "AllocationId": "eipalloc-abcdef01234567890",
```

```
            "PtrRecord": "example.com.",
            "PtrRecordUpdate": {
                "Value": "example.net.",
                "Status": "PENDING"
            }
        }
    ]
}
```

PowerShell

역방향 DNS 레코드를 제거하려면 다음을 수행하세요.

Reset-EC2AddressAttribute cmdlet을 사용하세요.

```
Reset-EC2AddressAttribute `
    -AllocationId 'eipalloc-abcdef01234567890' `
    -Attribute domain-name |
Format-List `
    AllocationId, PtrRecord, PublicIp,
    @{Name='PtrRecordUpdate';Expression={$_.PtrRecordUpdate | Format-List | Out-
String}}
```

출력의 예제는 다음과 같습니다.

```
AllocationId    : eipalloc-abcdef01234567890
PtrRecord       : example.com.
PublicIp        : 192.0.2.0
PtrRecordUpdate :
                  Reason :
                  Status : PENDING
                  Value  : example.net.
```

탄력적 네트워크 인터페이스

탄력적 네트워크 인터페이스는 VPC에서 가상 네트워크 카드를 나타내는 논리적 네트워킹 구성 요소입니다. 네트워크 인터페이스를 생성 및 구성하고 동일한 가용 영역에서 시작하는 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 원본 인스턴스에서 새 인스턴스로 리디렉션됩니다.

이 AWS 리소스를 AWS Management Console 및 Amazon EC2 API에서는 네트워크 인터페이스라고 합니다. 따라서 이 설명서에서는 “탄력적 네트워크 인터페이스” 대신 “네트워크 인터페이스”를 사용합니다. 이 설명서에서 “네트워크 인터페이스”라는 용어는 항상 “탄력적 네트워크 인터페이스”를 의미합니다.

네트워크 인터페이스 속성

네트워크 인터페이스에는 다음 속성이 포함될 수 있습니다.

- 서브넷의 IPv4 주소 범위 중 기본 프라이빗 IPv4 주소
- 서브넷의 IPv6 주소 범위 중 기본 IPv6 주소
- 서브넷의 IPv4 주소 범위의 보조 프라이빗 IPv4 주소
- 프라이빗 IPv4 주소당 하나의 탄력적 IP 주소(IPv4)
- 한 개의 퍼블릭 IPv4 주소
- 보조 IPv6 주소
- 보안 그룹
- MAC 주소
- 원본/대상 확인 플래그
- 설명

트래픽 모니터링

네트워크 인터페이스에서 VPC 흐름 로그를 활성화하여 네트워크 인터페이스로 주고 받는 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 참조하세요.

내용

- 네트워크 인터페이스 개념
- 네트워크 카드
- 네트워크 인터페이스당 최대 IP 주소
- EC2 인스턴스용 네트워크 인터페이스 생성
- EC2 인스턴스용 네트워크 인터페이스 연결
- 네트워크 인터페이스의 IP 주소 관리
- 네트워크 인터페이스 속성 수정
- Amazon EC2 인스턴스를 위한 여러 네트워크 인터페이스

- 요청자 관리 네트워크 인터페이스
- Amazon EC2 네트워크 인터페이스에 대한 접두사 위임
- 네트워크 인터페이스 삭제

네트워크 인터페이스 개념

다음은 네트워크 인터페이스 사용을 시작하려면 알아야 하는 중요한 개념입니다.

기본 네트워크 인터페이스

각 인스턴스는 기본 네트워크 인터페이스라는 기본 네트워크 인터페이스를 갖습니다. 주 네트워크 인터페이스는 인스턴스에서 분리할 수 없습니다.

보조 네트워크 인터페이스

보조 네트워크 인터페이스를 생성하고 인스턴스에 연결할 수 있습니다. 최대 네트워크 인터페이스 수는 인스턴스 유형에 따라 다릅니다. 자세한 내용은 네트워크 인터페이스당 최대 IP 주소 섹션을 참조하세요.

네트워크 인터페이스용 IPv4 주소

EC2 인스턴스를 IPv4 전용 또는 이중 스택 서브넷에서 시작할 경우, 인스턴스는 서브넷의 IPv4 주소 범위에서 기본 프라이빗 IP 주소를 수신합니다. 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.

네트워크 인터페이스용 퍼블릭 IPv4 주소

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 서브넷 설정을 참조하세요. 인스턴스를 시작하면 기본 네트워크 인터페이스에 IP 주소가 할당됩니다. 인스턴스를 시작할 때 기존 네트워크 인터페이스를 기본 네트워크 인터페이스로 지정하면 퍼블릭 IPv4 주소는 이 네트워크 인터페이스에 의해 결정됩니다.

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.

인스턴스가 중지되거나 최대 절전 모드로 전환되거나 종료되면 퍼블릭 IP 주소가 해제됩니다. 보조 네트워크 인터페이스 또는 탄력적 IP 주소와 연결된 보조 프라이빗 IPv4 주소가 없는 경우 중지되거나 최대 절전 모드로 전환된 인스턴스를 시작할 때 새 퍼블릭 IP 주소가 할당됩니다.

네트워크 인터페이스용 IPv6 주소

IPv6 CIDR 블록을 VPC 및 서브넷에 연결하고 서브넷 범위에 속하는 IPv6 주소를 네트워크 인터페이스에 할당할 수 있습니다. 각 IPv6 주소는 하나의 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. 인스턴스를 시작하면 기본 네트워크 인터페이스에 IPv6 주소가 할당됩니다.

네트워크 인터페이스의 탄력적 IP 주소

네트워크 인터페이스의 프라이빗 IPv4 주소 중 하나와 탄력적 IP 주소를 연결할 수 있습니다. 한 탄력적 IP 주소를 각 프라이빗 IPv4 주소와 연결할 수 있습니다. 네트워크 인터페이스에서 탄력적 IP 주소를 연결 해제하면 해당 주소를 해제하거나 다른 인스턴스에 연결할 수 있습니다.

종료 동작

인스턴스에 연결된 네트워크 인터페이스의 종료 동작을 설정할 수 있습니다. 연결된 인스턴스를 종료할 때 네트워크 인터페이스를 자동으로 삭제할지를 지정할 수 있습니다.

원본/대상 확인

원본/대상 확인을 활성화 또는 비활성화할 수 있습니다. 이를 통해 인스턴스가 수신되는 트래픽의 원본 또는 대상인지 확인할 수 있습니다. 원본/대상 확인은 기본적으로 활성화됩니다. 인스턴스에서 네트워크 주소 변환, 라우팅 또는 방화벽과 같은 서비스를 실행하는 경우 소스/대상 확인을 비활성화해야 합니다.

요청자 관리 네트워크 인터페이스

이러한 네트워크 인터페이스는 사용자가 일부 리소스와 서비스를 사용할 수 있도록 AWS 서비스에서 생성하고 관리합니다. 이러한 네트워크 인터페이스는 사용자가 직접 관리할 수 없습니다. 자세한 내용은 요청자 관리 네트워크 인터페이스 섹션을 참조하세요.

접두사 위임

접두사는 인스턴스와 연결된 네트워크 인터페이스에 자동 또는 수동으로 할당하기 위해 할당하는 예약된 프라이빗 IPv4 또는 IPv6 CIDR 범위입니다. 위임된 접두사를 사용하면 IP 주소 범위를 단일 접두사로 할당하여 서비스를 더 빠르게 시작할 수 있습니다.

관리형 네트워크 인터페이스

관리형 네트워크 인터페이스는 Amazon EKS Auto Mode와 같은 서비스 제공업체가 관리합니다. 관리형 네트워크 인터페이스의 설정은 직접 수정할 수 없습니다. 관리형 네트워크 인터페이스는 관리형 필드에서 true 값으로 식별됩니다. 자세한 내용은 Amazon EC2 관리형 인스턴스 섹션을 참조하세요.

네트워크 카드

대부분의 인스턴스 유형은 하나의 네트워크 카드를 지원합니다. 여러 네트워크 카드를 지원하는 인스턴스 유형은 100Gbps 이상의 대역폭 기능과 향상된 패킷 속도 성능을 포함하여 더 높은 네트워크 성능을 제공합니다. 여러 네트워크 카드를 지원하는 인스턴스에 네트워크 인터페이스를 연결할 때 네트워크 인터페이스의 네트워크 카드를 선택할 수 있습니다. 기본 네트워크 인터페이스는 네트워크 카드 인덱스 0에 할당되어야 합니다.

EFA 및 EFA 전용 네트워크 인터페이스는 네트워크 인터페이스로 간주됩니다. 네트워크 카드당 하나의 EFA 또는 EFA 전용 네트워크 인터페이스만 할당할 수 있습니다. EFA 전용 네트워크 인터페이스는 기본 네트워크 인터페이스로 사용할 수 없습니다.

다음 인스턴스 유형은 여러 네트워크 카드를 지원합니다. 인스턴스 유형이 지원하는 네트워크 인터페이스 수에 대한 자세한 내용은 네트워크 인터페이스당 최대 IP 주소 섹션을 참조하세요.

인스턴스 유형	네트워크 카드 수
c6in.32xlarge	2
c6in.metal	2
dl1.24xlarge	4
g6e.24xlarge	2
g6e.48xlarge	4
hpc6id.32xlarge	2
hpc7a.12xlarge	2
hpc7a.24xlarge	2
hpc7a.48xlarge	2
hpc7a.96xlarge	2
m6idn.32xlarge	2
m6idn.metal	2

인스턴스 유형	네트워크 카드 수
m6in.32xlarge	2
m6in.metal	2
p4d.24xlarge	4
p4de.24xlarge	4
p5.48xlarge	32
p5e.48xlarge	32
p5en.48xlarge	16
p6-b200.48xlarge	8
r6idn.32xlarge	2
r6idn.metal	2
r6in.32xlarge	2
r6in.metal	2
trn1.32xlarge	8
trn1n.32xlarge	16
trn2.48xlarge	16
trn2u.48xlarge	16
u7in-16tb.224xlarge	2
u7in-24tb.224xlarge	2
u7in-32tb.224xlarge	2
u7inh-32tb.480xlarge	2

네트워크 인터페이스당 최대 IP 주소

각 인스턴스 유형은 최대 네트워크 인터페이스 수, 네트워크 인터페이스당 최대 프라이빗 IPv4 주소, 네트워크 인터페이스당 최대 IPv6 주소 수를 지원합니다. IPv6 주소 제한은 네트워크 인터페이스당 프라이빗 IPv4 주소 제한과 별개입니다. C1, M1, M2, M3, T1을 제외한 모든 인스턴스 유형은 IPv6 주소 지정을 지원합니다.

사용 가능한 네트워크 인터페이스

Amazon EC2 인스턴스 유형 안내서에서는 각 인스턴스 유형에 사용할 수 있는 네트워크 인터페이스에 대한 정보를 제공합니다. 자세한 내용은 다음 자료를 참조하세요.

- 네트워크 사양 - 범용
- 네트워크 사양 - 컴퓨팅 최적화
- 네트워크 사양 - 메모리 최적화
- 네트워크 사양 - 스토리지 최적화
- 네트워크 사양 - 가속 컴퓨팅
- 네트워크 사양 - 고성능 컴퓨팅
- 네트워크 사양 - 이전 세대

Console

최대 네트워크 인터페이스를 검색하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스 유형을 선택합니다.
3. 필터를 추가하여 인스턴스 유형(Instance type=c5.12xlarge) 또는 인스턴스 패밀리(Instance family=c5)를 지정합니다.
4. (선택 사항) 기본 설정 아이콘을 클릭한 다음에 최대 네트워크 인터페이스 수를 켭니다. 이 열은 각 인스턴스 유형에 대한 최대 네트워크 인터페이스 수를 나타냅니다.
5. (선택 사항) 인스턴스 유형을 선택합니다. 네트워킹 탭에서 최대 네트워크 인터페이스 수를 찾습니다.

AWS CLI

최대 네트워크 인터페이스를 검색하는 방법

describe-instance-types 명령을 사용하여 지원되는 네트워크 인터페이스 및 인터페이스당 IP 주소와 같은 인스턴스 유형 관련 정보를 표시할 수 있습니다. 다음 예시에서는 모든 C5 인스턴스에 대해 이 정보를 표시합니다.

```
aws ec2 describe-instance-types \
    --filters "Name=instance-type,Values=c5.*" \
    --query "InstanceTypes[].{ \
        Type: InstanceType, \
        MaxENI: NetworkInfo.MaximumNetworkInterfaces, \
        IPv4addr: NetworkInfo.Ipv4AddressesPerInterface}" \
    --output table
```

출력의 예시는 다음과 같습니다.

```
-----------------------------------------
|        DescribeInstanceTypes        |
+----------+----------+---------------+
| IPv4addr | MaxENI   |     Type      |
+----------+----------+---------------+
|  30      |  8       |  c5.4xlarge   |
|  50      |  15      |  c5.24xlarge  |
|  15      |  4       |  c5.xlarge    |
|  30      |  8       |  c5.12xlarge  |
|  10      |  3       |  c5.large     |
|  15      |  4       |  c5.2xlarge   |
|  50      |  15      |  c5.metal     |
|  30      |  8       |  c5.9xlarge   |
|  50      |  15      |  c5.18xlarge  |
+----------+----------+---------------+
```

PowerShell

최대 네트워크 인터페이스를 검색하는 방법

Get-EC2InstanceType PowerShell 명령을 사용하여 지원되는 네트워크 인터페이스 및 인터페이스당 IP 주소와 같은 인스턴스 유형 관련 정보를 표시할 수 있습니다. 다음 예시에서는 모든 C5 인스턴스에 대해 이 정보를 표시합니다.

```
Get-EC2InstanceType -Filter @{Name="instance-type"; Values="c5.*"} | `
Select-Object `
    @{Name='Ipv4AddressesPerInterface';
 Expression={($_.Networkinfo.Ipv4AddressesPerInterface)}},
```

```
    @{Name='MaximumNetworkInterfaces';
 Expression={($_.Networkinfo.MaximumNetworkInterfaces)}},
    InstanceType | `
Format-Table -AutoSize
```

출력의 예시는 다음과 같습니다.

```
Ipv4AddressesPerInterface MaximumNetworkInterfaces InstanceType
------------------------- ----------------------- ------------
                       30                       8 c5.4xlarge
                       15                       4 c5.xlarge
                       30                       8 c5.12xlarge
                       50                      15 c5.24xlarge
                       30                       8 c5.9xlarge
                       50                      15 c5.metal
                       15                       4 c5.2xlarge
                       10                       3 c5.large
                       50                      15 c5.18xlarge
```

EC2 인스턴스용 네트워크 인터페이스 생성

EC2 인스턴스에서 사용할 네트워크 인터페이스를 생성할 수 있습니다. 네트워크 인터페이스를 생성할 때 네트워크 인터페이스를 생성할 서브넷을 지정합니다. 네트워크 인터페이스는 일단 생성되고 나면 다른 서브넷으로 옮길 수 없습니다. 네트워크 인터페이스는 동일한 가용 영역의 인스턴스에 연결해야 합니다. 보조 네트워크 인터페이스를 인스턴스에서 분리한 다음 동일한 가용 영역의 다른 인스턴스에 연결할 수 있습니다. 주 네트워크 인터페이스는 인스턴스에서 분리할 수 없습니다. 자세한 내용은 the section called “네트워크 인터페이스 연결” 섹션을 참조하세요.

Console

네트워크 인터페이스를 생성하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.
3. 네트워크 인터페이스 생성을 선택합니다.
4. (선택 사항) 설명에 설명적인 이름을 입력합니다.
5. 서브넷에서 서브넷을 선택합니다. 다음 단계에서 사용할 수 있는 옵션은 선택한 서브넷 유형(IPv4 전용, IPv6 전용 또는 이중 스택 (IPv4 및 IPv6))에 따라 달라집니다.

6. 인터페이스 유형에서 다음 중 하나를 선택합니다.

   - ENA: CPU 사용량을 최소화하면서 TCP/IP 프로토콜의 높은 처리량과 초당 패킷 속도를 처리하도록 설계된 고성능 네트워크 인터페이스입니다. 이것이 기본값입니다. ENA에 대한 자세한 내용은 Elastic Network Adapter 섹션을 참조하세요.
   - ENA 포함 EFA: SRD 기반 전송과 함께 기존 TCP/IP 기반 전송을 위해 ENA 디바이스와 EFA 디바이스를 모두 지원하는 네트워크 인터페이스입니다. ENA가 포함된 EFA를 선택하는 경우 연결하려는 인스턴스가 EFA를 지원해야 합니다. EFA에 대한 자세한 내용은 Elastic Fabric Adapter 섹션을 참조하세요.
   - EFA 전용: 운영 체제 스택을 우회하면서 SRD 기반 전송을 위한 높은 처리량과 지연 시간이 짧은 노드 간 통신을 처리하도록 설계된 고성능 네트워크 인터페이스입니다. 이 옵션을 선택하는 경우 연결하려는 인스턴스가 EFA를 지원해야 합니다. EFA 전용 네트워크 인터페이스는 IP 주소를 지원하지 않습니다. EFA에 대한 자세한 내용은 Elastic Fabric Adapter 섹션을 참조하세요.

7. 프라이빗 IPv4 주소에 대해 다음 중 하나를 수행합니다.

   - 자동 할당을 선택하여 Amazon EC2가 서브넷에서 IPv4 주소를 선택하도록 허용합니다.
   - 사용자 지정을 선택하고 서브넷에서 선택한 IPv4 주소를 입력합니다.

8. (IPv6 주소가 있는 서브넷에만 해당) IPv6 주소에서 다음 중 하나를 수행합니다.

   - 네트워크 인터페이스에 IPv6 주소를 할당하지 않으려는 경우 없음을 선택합니다.
   - 자동 할당을 선택하여 Amazon EC2가 서브넷에서 IPv6 주소를 선택하도록 허용합니다.
   - 사용자 지정을 선택하고 서브넷에서 선택한 IPv6 주소를 입력합니다.

9. (선택 사항) 듀얼 스택 또는 IPv6 전용 서브넷에서 네트워크 인터페이스를 생성하는 경우 기본 IPv6 IP를 할당할 수 있습니다. 이렇게 하면 네트워크 인터페이스에 기본 IPv6 글로벌 유니캐스트 주소(GUA)가 할당됩니다. 기본 IPv6 주소를 할당하면 인스턴스나 ENI에 대한 트래픽 중단을 방지할 수 있습니다. 이 ENI가 연결될 인스턴스가 변경되지 않는 IPv6 주소를 사용하는 경우 활성화를 선택하세요. AWS은(는) 인스턴스에 연결된 ENI와 연결된 IPv6 주소를 기본 IPv6 주소로 자동 할당합니다. IPv6 GUA 주소를 기본 IPv6로 활성화한 후에는 비활성화할 수 없습니다. IPv6 GUA 주소를 기본 IPv6로 활성화하면 인스턴스가 종료되거나 네트워크 인터페이스가 분리될 때까지 첫 번째 IPv6 GUA가 기본 IPv6 주소로 설정됩니다. 인스턴스에 연결된 ENI와 연결된 IPv6 주소가 여러 개 있고 기본 IPv6 주소를 활성화한 경우 ENI와 연결된 첫 번째 IPv6 GUA 주소가 기본 IPv6 주소가 됩니다.

10. (선택 사항) Elastic Fabric Adapter(EFA)를 생성하려면 Elastic Fabric Adapter(EFA),활성화를 선택합니다.

11. (선택 사항) 고급 설정에서 선택적으로 IP 접두사 위임을 설정할 수 있습니다. 자세한 내용은 접두사 위임 섹션을 참조하세요.
    - 자동 할당-AWS가 서브넷의 IPv4 또는 IPv6 CIDR 블록에서 접두사를 선택하고 네트워크 인터페이스에 할당합니다.
    - 사용자 지정-사용자가 서브넷의 IPv4 또는 IPv6 CIDR 블록에서 접두사를 지정하고, AWS는 접두사를 네트워크 인터페이스에 할당하기 전에 접두사가 다른 리소스에 할당되지 않았는지 확인합니다.
12. (선택 사항) 고급 설정에서 유휴 연결 추적 제한 시간에 대해 기본 유휴 연결 제한 시간을 수정합니다. 자세한 내용은 유휴 연결 추적 제한 시간 섹션을 참조하세요.
    - TCP 설정 제한 시간: 설정된 상태의 유휴 TCP 연결에 대한 제한 시간(초)입니다. 최솟값: 60초. 최댓값: 43만 2,000초(5일). 기본값: 43만 2,000초. 권장값: 43만 2,000초 미만.
    - UDP 제한 시간: 단일 방향 또는 단일 요청-응답 트랜잭션의 트래픽만 확인한 유휴 UDP 흐름의 제한 시간(초)입니다. 최솟값: 30초. 최댓값: 60초. 기본값: 30초.
    - UDP 스트림 제한 시간: 둘 이상의 요청-응답 트랜잭션을 확인한 스트림으로 분류된 유휴 UDP 흐름의 제한 시간(초)입니다. 최솟값: 60초. 최댓값: 180초(3분) 기본값: 180초
13. 보안 그룹에서 하나 이상의 보안 그룹을 선택합니다.
14. (선택 사항) 각 태그에 대해 [새 태그 추가(Add new tag)]를 선택하고 태그 키와 선택적 태그 값을 입력합니다.
15. 네트워크 인터페이스 생성을 선택합니다.

AWS CLI

예제 1: Amazon EC2에서 선택한 IP 주소로 네트워크 인터페이스를 생성하는 방법

다음 create-network-interface 명령을 사용합니다. 이 예제에서는 Amazon EC2에서 선택한 퍼블릭 IPv4 주소와 IPv6 주소로 네트워크 인터페이스를 생성합니다.

```
aws ec2 create-network-interface \
    --subnet-id subnet-0abcdef1234567890 \
    --description "my dual-stack network interface" \
    --ipv6-address-count 1 \
    --groups sg-1234567890abcdef0
```

예제 2: 특정 IP 주소로 네트워크 인터페이스를 생성하는 방법

다음 create-network-interface 명령을 사용합니다.

```
aws ec2 create-network-interface \
    --subnet-id subnet-0abcdef1234567890 \
    --description "my dual-stack network interface" \
    --private-ip-address 10.251.50.12 \
    --ipv6-addresses 2001:db8::1234:5678:1.2.3.4 \
    --groups sg-1234567890abcdef0
```

예제 3: 여러 보조 IP 주소로 네트워크 인터페이스를 생성하는 방법

다음 create-network-interface 명령을 사용합니다. 이 예제의 Amazon EC2에서는 기본 IP 주소와 보조 IP 주소를 모두 선택합니다.

```
aws ec2 create-network-interface \
    --subnet-id subnet-0abcdef1234567890 \
    --description "my network interface" \
    --secondary-private-ip-address-count 2 \
    --groups sg-1234567890abcdef0
```

예제 4: 특정 보조 IP 주소로 네트워크 인터페이스를 생성하는 방법

다음 create-network-interface 명령을 사용합니다. 이 예제에서는 기본 IP 주소와 보조 IP 주소를 지정합니다.

```
aws ec2 create-network-interface \
    --subnet-id subnet-0abcdef1234567890 \
    --description "my network interface" \
    --private-ip-addresses PrivateIpAddress=10.0.1.30,Primary=true \
                           PrivateIpAddress=10.0.1.31,Primary=false
    --groups sg-1234567890abcdef0
```

PowerShell

예제 1: Amazon EC2에서 선택한 IP 주소로 네트워크 인터페이스를 생성하는 방법

New-EC2NetworkInterface cmdlet을 사용합니다. 이 예제에서는 Amazon EC2에서 선택한 퍼블릭 IPv4 주소와 IPv6 주소로 네트워크 인터페이스를 생성합니다.

```
New-EC2NetworkInterface `
    -SubnetId subnet-0abcdef1234567890 `
    -Description "my dual-stack network interface" `
    -Ipv6AddresCount 1 `
```

```
    -Group sg-1234567890abcdef0
```

예제 2: 특정 IP 주소로 네트워크 인터페이스를 생성하는 방법

New-EC2NetworkInterface cmdlet을 사용합니다.

```
New-EC2NetworkInterface `
    -SubnetId subnet-0abcdef1234567890 `
    -Description "my dual-stack network interface" `
    -PrivateIpAddress 10.251.50.12 `
    -Ipv6Address $ipv6addr `
    -Group sg-1234567890abcdef0
```

다음과 같이 IPv6 주소를 정의합니다.

```
$ipv6addr = New-Object Amazon.EC2.Model.InstanceIpv6Address
$ipv6addr1.Ipv6Address = "2001:db8::1234:5678:1.2.3.4"
```

예제 3: 여러 보조 IP 주소로 네트워크 인터페이스를 생성하는 방법

New-EC2NetworkInterface cmdlet을 사용합니다. 이 예제의 Amazon EC2에서는 기본 IP 주소와 보조 IP 주소를 모두 선택합니다.

```
New-EC2NetworkInterface `
    -SubnetId subnet-0abcdef1234567890 `
    -Description "my network interface" `
    -SecondaryPrivateIpAddressCount 2 `
    -Group sg-1234567890abcdef0
```

예제 4: 특정 보조 IP 주소로 네트워크 인터페이스를 생성하는 방법

New-EC2NetworkInterface cmdlet을 사용합니다. 이 예제에서는 기본 IP 주소와 보조 IP 주소를 지정합니다.

```
New-EC2NetworkInterface `
    -SubnetId subnet-0abcdef1234567890 `
    -Description "my network interface" `
    -PrivateIpAddresses @($primary, $secondary) `
    -Group sg-1234567890abcdef0
```

다음과 같이 보조 주소를 정의합니다.

```
$primary = New-Object Amazon.EC2.Model.PrivateIpAddressSpecification
$primary.PrivateIpAddress = "10.0.1.30"
$primary.Primary = $true
$secondary = New-Object Amazon.EC2.Model.PrivateIpAddressSpecification
$secondary.PrivateIpAddress = "10.0.1.31"
$secondary.Primary = $false
```

EC2 인스턴스용 네트워크 인터페이스 연결

기본 또는 보조 네트워크 인터페이스로 EC2 인스턴스에서 사용할 네트워크 인터페이스를 생성할 수 있습니다. 네트워크 인터페이스는 네트워크 인터페이스와 동일한 가용 영역의 EC2 인스턴스에 연결해야 합니다. 인스턴스의 인스턴스 유형에 따라 인스턴스에 연결할 수 있는 네트워크 인터페이스 수가 결정됩니다. 자세한 내용은 the section called “네트워크 인터페이스당 IP 주소” 섹션을 참조하세요.

고려 사항

- 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 인터페이스는 분리할 수 없습니다.
- 보조 네트워크 인터페이스를 인스턴스에서 분리하고 다른 인스턴스에 연결할 수 있습니다.
- CLI, API 또는 SDK에서 인스턴스를 시작할 때 기본 네트워크 인터페이스 및 추가 네트워크 인터페이스를 지정할 수 있습니다. 시작 중에 보조 네트워크 인터페이스를 추가하는 경우 퍼블릭 IPv4 주소의 자동 할당을 활성화할 수 없습니다.
- 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 또는 핫 연결을 사용하려면 수동으로 두 번째 인터페이스를 열고 프라이빗 IPv4 주소를 구성하고 그에 따라 라우팅 테이블을 수정해야 할 수 있습니다. Amazon Linux 또는 Windows Server를 실행하는 인스턴스는 웜 또는 핫 연결을 자동으로 인식하여 자체적으로 구성됩니다.
- 이중 홈 인스턴스로 송/수신되는 네트워크 대역폭을 높이거나 두 배로 늘리기 위해 인스턴스에 다른 네트워크 인터페이스를 연결(예: NIC 팀 구성)할 수는 없습니다.
- 동일한 서브넷의 여러 네트워크 인터페이스를 한 인스턴스에 연결하면 비대칭 라우팅 같은 네트워킹 문제가 발생할 수 있습니다. 가능하면 대신 기본 네트워크 인터페이스에서 보조 프라이빗 IPv4 주소를 추가하세요.

- IPv6 전용 서브넷에 있는 EC2 인스턴스의 경우 보조 네트워크 인터페이스를 연결하면 보조 네트워크 인터페이스의 프라이빗 DNS 호스트 이름이 기본 네트워크 인터페이스의 기본 IPv6 주소로 확인됩니다.
- [Windows 인스턴스] - 여러 개의 네트워크 인터페이스를 추가하는 경우 정적 라우팅을 사용하도록 네트워크 인터페이스를 구성해야 합니다.

네트워크 인터페이스 연결

Amazon EC2 콘솔의 [인스턴스(Instances)] 또는 [네트워크 인터페이스(Network Interfaces)] 페이지를 사용하여 네트워크 인터페이스와 동일한 가용 영역의 인스턴스에 네트워크 인터페이스를 연결할 수 있습니다. 또는 인스턴스를 시작할 때 기존 네트워크 인터페이스를 지정할 수 있습니다.

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 단원을 참조하세요.

Console

인스턴스 페이지를 사용하여 네트워크 인터페이스를 연결하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스에 대한 확인란을 선택합니다.
4. 작업, 네트워킹, 네트워크 인터페이스 연결을 차례로 선택합니다.
5. VPC를 선택합니다. 네트워크 인터페이스는 인스턴스와 동일한 가용 영역에 있는 한, 인스턴스와 동일한 VPC 또는 사용자가 소유한 다른 VPC에 있을 수 있습니다. 그러면 네트워킹 및 보안 구성이 서로 다른 여러 VPC 사이에 다중 홈 인스턴스를 만들 수 있습니다.
6. 네트워크 인터페이스를 선택합니다. 인스턴스가 여러 네트워크 카드를 지원하는 경우 네트워크 카드를 선택할 수 있습니다.
7. 연결을 선택합니다.

네트워크 인터페이스 페이지를 사용하여 네트워크 인터페이스를 연결하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.
3. 네트워크 인터페이스의 확인란을 선택합니다.

4. [작업], [연결]을 선택합니다.
5. 인스턴스를 선택합니다. 인스턴스가 여러 네트워크 카드를 지원하는 경우 네트워크 카드를 선택할 수 있습니다.
6. 연결을 선택합니다.

AWS CLI

네트워크 인터페이스를 연결하는 방법

다음 attach-network-interface 명령을 사용합니다.

```
aws ec2 attach-network-interface \
    --network-interface-id eni-1234567890abcdef0 \
    --instance-id i-1234567890abcdef0 \
    --device-index 1
```

PowerShell

네트워크 인터페이스를 연결하는 방법

Add-EC2NetworkInterface cmdlet을 사용합니다.

```
Add-EC2NetworkInterface `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -InstanceId i-1234567890abcdef0 `
    -DeviceIndex 1
```

네트워크 인터페이스 분리

Amazon EC2 콘솔의 인스턴스 또는 네트워크 인터페이스 페이지를 사용하면 언제든지 EC2 인스턴스에 연결된 보조 네트워크 인터페이스를 분리할 수 있습니다.

Elastic Load Balancing 로드 밸런서, Lambda 함수, WorkSpace 또는 NAT 게이트웨이와 같은 다른 서비스에서 리소스에 연결된 네트워크 인터페이스를 분리하려고 하면, 리소스에 액세스할 권한이 없다는 오류가 발생합니다. 네트워크 인터페이스에 연결된 리소스를 생성한 서비스를 찾으려면 네트워크 인터페이스에 대한 설명을 확인하세요. 리소스를 삭제하면 해당 네트워크 인터페이스가 삭제됩니다.

Console

인스턴스 페이지를 사용하여 네트워크 인터페이스를 분리하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스에 대한 확인란을 선택합니다. 네트워킹 탭의 네트워크 인터페이스섹션을 확인하여 네트워크 인터페이스가 인스턴스에 보조 네트워크 인터페이스로 연결되어 있는지 알아봅니다.
4. 작업, 네트워킹, 네트워크 인터페이스 분리를 차례로 선택합니다.
5. 네트워크 인터페이스를 선택하고 분리를 선택합니다.

네트워크 인터페이스 페이지를 사용하여 인스턴스를 분리하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.
3. 네트워크 인터페이스의 확인란을 선택합니다. [세부 정보] 탭의 [인스턴스 세부 정보] 섹션을 확인하여 네트워크 인터페이스가 인스턴스에 보조 네트워크 인터페이스로 연결되어 있는지 알아봅니다.
4. [작업], [삭제]를 선택합니다.
5. 확인 메시지가 나타나면 [Detach]를 선택합니다.
6. 네트워크 인터페이스가 인스턴스에서 분리되지 않으면 [강제 분리], [활성화]를 선택하고 다시 시도합니다. 분리 강제는 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.

AWS CLI

네트워크 인터페이스를 분리하는 방법

다음 detach-network-interface 명령을 사용합니다.

```
aws ec2 detach-network-interface --attachment-id eni-attach-016c93267131892c9
```

PowerShell

네트워크 인터페이스를 분리하는 방법

Dismount-EC2NetworkInterface cmdlet을 사용합니다.

```
Dismount-EC2NetworkInterface -AttachmentId eni-attach-016c93267131892c9
```

네트워크 인터페이스의 IP 주소 관리

네트워크 인터페이스에 대해 다음 IP 주소를 관리할 수 있습니다.

- 탄력적 IP 주소(프라이빗 IPv4 주소당 하나)
- IPv4 주소
- IPv6 주소

Console

네트워크 인터페이스의 IP 주소를 관리하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.
3. 네트워크 인터페이스의 확인란을 선택합니다.
4. IPv4 및 IPv6 주소를 관리하려면 다음을 수행합니다.
   a. [작업], [IP 주소 관리]를 선택합니다.
   b. 네트워크 인터페이스를 확장합니다.
   c. IPv4 주소는 필요에 따라 IP 주소를 편집합니다. 추가 IPv4 주소를 할당하려면 새 IP 주소 할당을 선택한 다음에 서브넷 범위에서 IPv4 주소를 지정하거나 AWS가 선택하도록 합니다.
   d. (듀얼 스택 또는 IPv6만 해당) IPv6 주소는 필요에 따라 IP 주소를 편집합니다. 추가 IPv6 주소를 할당하려면 새 IP 주소 할당을 선택한 다음에 서브넷 범위에서 IPv6 주소를 지정하거나 AWS가 선택하도록 합니다.
   e. 네트워크 인터페이스에 퍼블릭 IPv4 주소를 할당하거나 할당 해제하려면 퍼블릭 IP 자동 할당을 선택합니다. 모든 네트워크 인터페이스에 대해 이 기능을 활성화하거나 비활성화할 수 있지만, 기본 네트워크 인터페이스에만 영향을 미칩니다.
   f. (듀얼 스택 또는 IPv6 전용) 기본 IPv6 IP 할당은 활성화를 선택하여 기본 IPv6 주소를 할당합니다. 네트워크 인터페이스와 연결된 첫 번째 GUA가 기본 IPv6 주소로 선택됩니다. 기

본 IPv6 주소를 할당한 후에는 변경할 수 없습니다. 이 주소는 인스턴스가 종료되거나 네트워크 인터페이스가 분리될 때까지 기본 IPv6 주소입니다.

g. 저장을 선택합니다.

5. 탄력적 IP 주소를 연결하려면 다음을 수행합니다.

a. [작업], [주소 연결]을 선택합니다.

b. [주소]에서 [탄력적 IP 주소]를 선택합니다.

c. [프라이빗 IP 주소]에 대해 탄력적 IP 주소와 연결할 프라이빗 IPv4 주소를 선택합니다.

d. (선택 사항) 네트워크 인터페이스가 현재 다른 인스턴스 또는 네트워크 인터페이스와 연결되어 있는 경우 [탄력적 IP 주소를 재연결하도록 허용]을 선택합니다.

e. 연결을 선택합니다.

6. 탄력적 IP 주소를 연결 해제하려면 다음을 수행합니다.

a. [작업(Actions)], [주소 연결 해제(Disassociate address)]를 선택합니다.

b. [주소]에서 [탄력적 IP 주소]를 선택합니다.

c. 연결 해제를 선택합니다.

AWS CLI

IPv4 주소를 관리하는 방법

다음 assign-private-ip-addresses 명령을 사용하여 IPv4 주소를 할당합니다.

```
aws ec2 assign-private-ip-addresses \
    --network-interface-id eni-1234567890abcdef0 \
    --private-ip-addresses 10.0.0.82
```

다음 unassign-private-ip-addresses 명령을 사용하여 IPv4 주소 할당을 해제합니다.

```
aws ec2 unassign-private-ip-addresses \
    --network-interface-id eni-1234567890abcdef0 \
    --private-ip-addresses 10.0.0.82
```

IPv6 주소를 관리하는 방법

다음 assign-ipv6-addresses 명령을 사용하여 IPv6 주소를 할당합니다.

```
aws ec2 assign-ipv6-addresses \
```

```
    --network-interface-id eni-1234567890abcdef0 \
    --ipv6-addresses 2001:db8:1234:1a00:9691:9503:25ad:1761
```

다음 unassign-ipv6-addresses 명령을 사용하여 IPv6 주소 할당을 해제합니다.

```
aws ec2 unassign-ipv6-addresses \
    --network-interface-id eni-1234567890abcdef0 \
    --ipv6-addresses 2001:db8:1234:1a00:9691:9503:25ad:1761
```

기본 프라이빗 IPv4 주소에 대한 탄력적 IP 주소를 관리하는 방법

다음 associate-address 명령을 사용하여 탄력적 IP 주소를 기본 프라이빗 IPv4 주소와 연결합니다.

```
aws ec2 associate-address \
    --allocation-id eipalloc-0b263919b6EXAMPLE \
    --network-interface-id eni-1234567890abcdef0
```

다음 disassociate-address 명령을 사용하여 기본 프라이빗 IPv4 주소에서 탄력적 IP 주소 연결을 해제합니다.

```
aws ec2 disassociate-address --association-id eipassoc-2bebb745a1EXAMPLE
```

PowerShell

IPv4 주소를 관리하는 방법

Register-EC2PrivateIpAddress cmdlet을 사용하여 IPv4 주소를 할당합니다.

```
Register-EC2PrivateIpAddress `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -PrivateIpAddress 10.0.0.82
```

Unregister-EC2PrivateIpAddress cmdlet을 사용하여 IPv4 주소 할당을 해제합니다.

```
Unregister-EC2PrivateIpAddress `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -PrivateIpAddress 10.0.0.82
```

IPv6 주소를 관리하는 방법

Register-EC2Ipv6AddressList cmdlet을 사용하여 IPv6 주소를 할당합니다.

```
Register-EC2Ipv6AddressList `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Ipv6Address 2001:db8:1234:1a00:9691:9503:25ad:1761
```

Unregister-EC2Ipv6AddressList cmdlet을 사용하여 IPv6 주소 할당을 해제합니다.

```
Unregister-EC2Ipv6AddressList `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Ipv6Address 2001:db8:1234:1a00:9691:9503:25ad:1761
```

기본 프라이빗 IPv4 주소에 대한 탄력적 IP 주소를 관리하는 방법

Register-EC2Address cmdlet을 사용하여 탄력적 IP 주소를 기본 프라이빗 IPv4 주소와 연결합니다.

```
Register-EC2Address `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -AllocationId eipalloc-0b263919b6EXAMPLE
```

Unregister-EC2Address cmdlet을 사용하여 기본 프라이빗 IPv4 주소에서 탄력적 IP 주소 연결을 해제합니다.

```
Unregister-EC2Address -AssociationId eipassoc-2bebb745a1EXAMPLE
```

네트워크 인터페이스 속성 수정

다음 네트워크 인터페이스 속성을 변경할 수 있습니다.

- 설명
- 보안 그룹
- 종료 시 삭제
- 원본/대상 확인
- 유휴 연결 추적 제한 시간

고려 사항

요청자 관리형 네트워크 인터페이스의 속성은 변경할 수 없습니다.

Console

네트워크 인터페이스 속성을 수정하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.
3. 네트워크 인터페이스의 확인란을 선택합니다.
4. 설명을 변경하려면 다음을 수행합니다.
   a. [작업], [설명 변경]을 선택합니다.
   b. 설명에 설명을 입력합니다.
   c. 저장을 선택합니다.
5. 보안 그룹을 변경하려면 다음을 수행합니다.
   a. [작업], [보안 그룹 변경]을 선택합니다.
   b. 연결된 보안 그룹은 필요에 따라 보안 그룹을 추가하고 제거합니다. 동일한 VPC에 대해 보안 그룹 및 네트워크 인터페이스를 생성해야 합니다.
   c. 저장을 선택합니다.
6. 종료 동작을 변경하려면 다음을 수행합니다.
   a. [작업], [종료 방식 변경]을 선택합니다.
   b. 종료 시 삭제, 활성화를 선택하거나 지웁니다.
   c. 저장을 선택합니다.
7. 소스/대상 확인을 변경하려면 다음을 수행합니다.
   a. [작업], [원본/대상 확인 변경(Change source/dest check)]을 선택합니다.
   b. 소스/대상 확인, 활성화를 선택하거나 지웁니다.
   c. 저장을 선택합니다.
8. 유휴 연결 추적 제한 시간을 변경하려면 다음을 수행합니다.
   a. 작업, 유휴 연결 추적 제한 시간 수정을 선택합니다.
   b. 필요에 따라 제한 시간 값을 수정합니다. 자세한 내용은 유휴 연결 추적 제한 시간 섹션을 참조하세요.
      - TCP 설정 제한 시간: 설정된 상태의 유휴 TCP 연결에 대한 제한 시간(초)입니다. 최솟값: 60초. 최댓값: 43만 2,000초(5일). 기본값: 43만 2,000초. 권장값: 43만 2,000초 미만.

- UDP 제한 시간: 단일 방향 또는 단일 요청-응답 트랜잭션의 트래픽만 확인한 유휴 UDP 흐름의 제한 시간(초)입니다. 최솟값: 30초. 최댓값: 60초. 기본값: 30초.
- UDP 스트림 제한 시간: 둘 이상의 요청-응답 트랜잭션을 확인한 스트림으로 분류된 유휴 UDP 흐름의 제한 시간(초)입니다. 최솟값: 60초. 최댓값: 180초(3분) 기본값: 180초

c. 저장을 선택합니다.

AWS CLI

Example 예제: 설명을 수정하는 방법

다음 modify-network-interface-attribute 명령을 사용합니다.

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --description "my updated description"
```

Example 예제: 보안 그룹을 수정하는 방법

다음 modify-network-interface-attribute 명령을 사용합니다.

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --groups sg-1234567890abcdef0
```

Example 예제: 종료 동작을 수정하는 방법

다음 modify-network-interface-attribute 명령을 사용합니다.

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --attachment AttachmentId=eni-attach-43348162abEXAMPLE,DeleteOnTermination=false
```

Example 예제: 소스/대상 확인을 활성화하는 방법

다음 modify-network-interface-attribute 명령을 사용합니다.

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
```

```
    --source-dest-check
```

Example 예제: 유휴 연결 추적 제한 시간을 수정하는 방법

다음 modify-network-interface-attribute 명령을 사용합니다. 자세한 내용은 유휴 연결 추적 제한 시간 섹션을 참조하세요.

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --connection-tracking-specification
 TcpEstablishedTimeout=172800,UdpStreamTimeout=90,UdpTimeout=60
```

PowerShell

Example 예제: 설명을 수정하는 방법

Edit-EC2NetworkInterfaceAttribute cmdlet을 사용합니다.

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Description "my updated description"
```

Example 예제: 보안 그룹을 수정하는 방법

Edit-EC2NetworkInterfaceAttribute cmdlet을 사용합니다.

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Group sg-1234567890abcdef0
```

Example 예제: 종료 동작을 수정하는 방법

Edit-EC2NetworkInterfaceAttribute cmdlet을 사용합니다.

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Attachment_AttachmentId eni-attach-43348162abEXAMPLE `
    -Attachment_DeleteOnTermination $false
```

Example 예제: 소스/대상 확인을 활성화하는 방법

Edit-EC2NetworkInterfaceAttribute cmdlet을 사용합니다.

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -SourceDestCheck $true
```

Example 예제: 유휴 연결 추적 제한 시간 수정을 수정하는 방법

Edit-EC2NetworkInterfaceAttribute cmdlet을 사용합니다. 자세한 내용은 유휴 연결 추적 제한 시간 섹션을 참조하세요.

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -ConnectionTrackingSpecification_TcpEstablishedTimeout 172800 `
    -ConnectionTrackingSpecification_UdpStreamTimeout 90 `
    -ConnectionTrackingSpecification_UdpTimeout 60
```

Amazon EC2 인스턴스를 위한 여러 네트워크 인터페이스

다음이 필요한 경우 여러 네트워크 인터페이스를 하나의 인스턴스에 연결하면 유용합니다.

- 관리 네트워크
- 네트워크 및 보안 어플라이언스.
- 서로 다른 서브넷 또는 VPC에 워크로드가 있는 이중 홈 인스턴스.
- 저예산 고가용성 솔루션

네트워크 관리

다음 개요에서는 여러 네트워크 인터페이스를 사용하여 생성한 관리 네트워크에 대해 설명합니다.

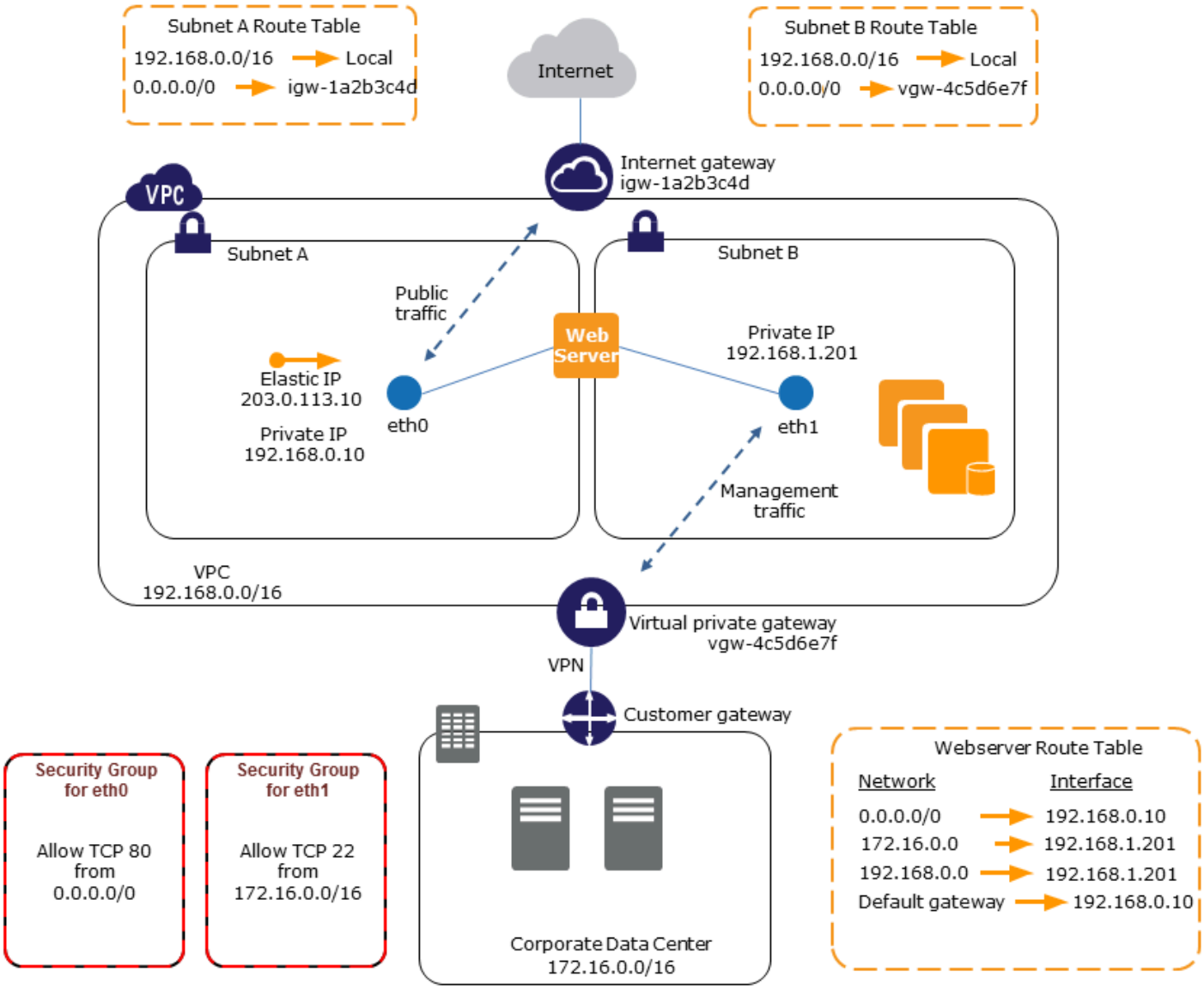
기준

- 인스턴스의 기본 네트워크 인터페이스(예: eth0)는 퍼블릭 트래픽을 처리합니다.
- 인스턴스의 보조 네트워크 인터페이스(eth1)는 백엔드 관리 트래픽을 처리합니다. 이 네트워크 인터페이스는 액세스 제어가 더 제한적인 별도의 서브넷에 연결되며, 기본 네트워크 인터페이스와 동일한 가용 영역 내에 있습니다.

설정

- 로드 밸런서 뒤에 있을 수도 있고 그렇지 않을 수도 있는 기본 네트워크 인터페이스에는 인터넷에서 서버에 액세스할 수 있는 연결된 보안 그룹이 있습니다. 예를 들어, 0.0.0.0/0 또는 로드 밸런서에서 TCP 포트 80 및 443을 허용합니다.
- 보조 네트워크 인터페이스에는 다음과 같은 위치 중 하나에서 시작된 SSH 액세스만 허용되는 연결된 보안 그룹이 있습니다.
  - VPC 내부 또는 인터넷에서 허용되는 IP 주소 범위입니다.
  - 기본 네트워크 인터페이스와 동일한 가용 영역 내의 프라이빗 서브넷입니다.
  - 가상 프라이빗 게이트웨이입니다.

> **Note**
>
> 장애 조치 기능을 유지하려면 네트워크 인터페이스에서 유입 트래픽에 대해 보조 프라이빗 IPv4 사용을 고려해 보세요. 인스턴스 장애 발생 시 인터페이스 및/또는 보조 프라이빗 IPv4 주소를 스탠바이 인스턴스로 이동할 수 있습니다.



네트워크 및 보안 어플라이언스

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 생성 및 연결한 후 이 추가 인터페이스를 고유의 퍼블릭 및 프라이빗 IP 주소, 보안 그룹 및 원본/대상 확인으로 구성할 수 있습니다.

서로 다른 서브넷에 워크로드가 있는 이중 홈 인스턴스

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.

같은 계정의 서로 다른 서브넷에 워크로드가 있는 이중 홈 인스턴스

네트워크 인터페이스가 EC2 인스턴스와 동일한 가용 영역에 있는 한, VPC에서 해당 인스턴스를 시작하고 다른 VPC에서 보조 ENI를 연결할 수 있습니다. 그러면 네트워킹 및 보안 구성이 서로 다른 여러 VPC 사이에 다중 홈 인스턴스를 만들 수 있습니다. 서로 다른 AWS 계정의 여러 VPC에 걸쳐 멀티홈 인스턴스를 생성할 수 없습니다.

다음과 같은 사용 사례에서는 여러 VPC 사이에 이중 홈 인스턴스를 사용할 수 있습니다.

- 함께 피어링할 수 없는 두 VPC 간 CIDR 중첩 해결: VPC의 보조 CIDR를 활용하여 겹치지 않는 두 IP 범위 간 통신을 인스턴스에 허용할 수 있습니다.
- 단일 계정 내 여러 VPC 연결: 일반적으로 VPC 경계로 구분되는 개별 리소스 간 통신이 활성화됩니다.

저예산 고가용성 솔루션

특정 기능을 제공하는 인스턴스 중 하나에 장애가 발생할 경우 서비스를 신속하게 복구하기 위해 관련 네트워크 인터페이스를 동일한 역할로 미리 구성된 대체 또는 핫 스탠바이 인스턴스에 연결할 수 있습니다. 예를 들어, 데이터베이스 인스턴스 또는 NAT 인스턴스와 같은 중요한 서비스에 대한 기본 또는 보조 네트워크 인터페이스로 네트워크 인터페이스를 사용할 수 있습니다. 인스턴스가 작동하지 않는 경우 사용자 또는 사용자를 대신하는 실행 중인 코드는 네트워크 인터페이스를 핫 스탠바이 인스턴스에 연결할 수 있습니다. 인터페이스에서 프라이빗 IP 주소, 탄력적 IP 주소 및 MAC 주소를 관리하므로 네트워크 인터페이스를 대체 인스턴스에 연결하자마자 네트워크 트래픽이 스탠바이 인스턴스로 전달되기 시작합니다. 인스턴스에 장애가 발생한 시간과 네트워크 인터페이스가 대기 인스턴스에 연결되는 시간 사이에 잠시 연결이 끊기지만 라우팅 테이블 또는 DNS 서버에 대해 어떠한 변경도 수행할 필요가 없습니다.

요청자 관리 네트워크 인터페이스

요청자 관리형 네트워크 인터페이스는 AWS 서비스가 사용자를 대신하여 VPC에 생성하는 네트워크 인터페이스입니다. 네트워크 인터페이스는 Amazon RDS의 DB 인스턴스, NAT 게이트웨이, AWS PrivateLink의 인터페이스 VPC 엔드포인트 등의 다른 서비스에 대한 리소스와 연결됩니다.

고려 사항

- 계정에서 요청자 관리형 네트워크 인터페이스를 볼 수 있습니다. 태그를 추가하거나 제거할 수 있지만, 요청자 관리형 네트워크 인터페이스의 기타 속성은 변경할 수 없습니다.
- 요청자 관리형 네트워크 인터페이스는 분리할 수 없습니다.
- 요청자 관리형 네트워크 인터페이스와 연결된 리소스를 삭제하면 AWS 서비스에서 네트워크 인터페이스를 분리하고 삭제합니다. 서비스가 네트워크 인터페이스를 분리했지만 삭제하지 않은 경우 분리된 네트워크 인터페이스를 삭제할 수 있습니다.

Console

요청자 관리형 네트워크 인터페이스를 보는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 네트워크 및 보안(Network & Security) 네트워크 인터페이스(Network Interfaces)를 선택합니다.
3. 네트워크 인터페이스의 ID를 선택하여 세부 정보 페이지를 엽니다.
4. 다음은 네트워크 인터페이스의 용도를 결정하는 데 사용할 수 있는 주요 필드입니다.
   - 설명(Description): 인터페이스를 생성한 AWS 서비스에서 제공하는 설명입니다. "VPC 엔드포인트 인터페이스 vpce 089f2123488812123"을 예로 들 수 있습니다.
   - 요청자 관리형(Requester-managed): 네트워크 인터페이스가 AWS에서 관리되는지 여부를 나타냅니다.
   - 요청자 ID(Requester ID): 네트워크 인터페이스를 생성한 보안 주체 또는 서비스의 별칭 또는 AWS 계정 ID입니다. 네트워크 인터페이스를 생성한 경우, 사용자의 AWS 계정 ID입니다. 그렇게 하지 않으면 다른 보안 주체 또는 서비스가 생성했습니다.

AWS CLI

요청자 관리형 네트워크 인터페이스를 보는 방법

다음과 같이 describe-network-interfaces 명령을 사용합니다.

```
aws ec2 describe-network-interfaces \
    --filters "Name=requester-managed,Values=true" \
    --query "NetworkInterfaces[*].[Description, InterfaceType]" \
    --output table
```

다음은 네트워크 인터페이스의 용도를 결정하는 데 사용할 수 있는 주요 필드(Description 및 InterfaceType)를 보여주는 예제 출력입니다.

```
-------------------------------------------------------------------------------
|                          DescribeNetworkInterfaces                          |
+-----------------------------------------------------+-------------------------+
|  VPC Endpoint Interface: vpce-0f00567fa8477a1e6     |  interface              |
|  VPC Endpoint Interface vpce-0d8ddce4be80e4474      |  interface              |
|  VPC Endpoint Interface vpce-078221a1e27d1ea5b      |  vpc_endpoint           |
|  Resource Gateway Interface rgw-0bba03f3d56060135   |  interface              |
|  VPC Endpoint Interface: vpce-0cc199f605eaeace7     |  interface              |
|  VPC Endpoint Interface vpce-019b90d6f16d4f958      |  interface              |
+-----------------------------------------------------+-------------------------+
```

PowerShell

요청자 관리형 네트워크 인터페이스를 보는 방법

다음과 같이 Get-EC2NetworkInterface cmdlet을 사용합니다.

```
Get-EC2NetworkInterface -Filter @{Name="requester-managed"; Values="true"} | Select
 Description, InterfaceType
```

다음은 네트워크 인터페이스의 용도를 결정하는 데 사용할 수 있는 주요 필드(Description 및 InterfaceType)를 보여주는 예제 출력입니다.

```
Description                                       InterfaceType
-----------                                       -------------
VPC Endpoint Interface: vpce-0f00567fa8477a1e6    interface
VPC Endpoint Interface vpce-0d8ddce4be80e4474     interface
VPC Endpoint Interface vpce-078221a1e27d1ea5b     vpc_endpoint
Resource Gateway Interface rgw-0bba03f3d56060135  interface
VPC Endpoint Interface: vpce-0cc199f605eaeace7    interface
VPC Endpoint Interface vpce-019b90d6f16d4f958     interface
```

Amazon EC2 네트워크 인터페이스에 대한 접두사 위임

프라이빗 IPv4 또는 IPv6 CIDR 범위를 자동 또는 수동으로 네트워크 인터페이스에 할당할 수 있습니다. 접두사를 할당하면 인스턴스에 여러 IP 주소가 필요한 컨테이너 및 네트워킹 애플리케이션을 포함

하여 애플리케이션을 확장하고 관리할 수 있습니다. IPv4 및 IPv6 주소에 대한 자세한 내용은 Amazon EC2 인스턴스 IP 주소 지정 섹션을 참조하세요.

다음과 같은 할당 옵션을 사용할 수 있습니다.

- 자동 할당(Automatic assignment) - AWS가 VPC 서브넷의 IPv4 또는 IPv6 CIDR 블록에서 접두사를 선택하고 네트워크 인터페이스에 할당합니다.
- 수동 할당(Manual Assignment) - 사용자가 VPC 서브넷의 IPv4 또는 IPv6 CIDR 블록에서 접두사를 지정하고, AWS는 접두사를 네트워크 인터페이스에 할당하기 전에 접두사가 다른 리소스에 할당되지 않았는지 확인합니다.

접두사를 할당하면 다음과 같은 이점이 있습니다.

- 네트워크 인터페이스에서 IP 주소 수 증가 - 접두사를 사용하면 개별 IP 주소가 아닌 IP 주소 블록을 할당합니다. 이렇게 하면 네트워크 인터페이스의 IP 주소 수가 늘어납니다.
- 컨테이너에 대한 VPC 관리 간소화 - 컨테이너 애플리케이션의 각 컨테이너는 고유한 IP 주소가 필요합니다. 인스턴스에 접두사를 할당하면 개별 IP 할당을 위해 Amazon EC2 API를 호출할 필요 없이 컨테이너를 시작하고 종료할 수 있으므로 VPC 관리가 간소화됩니다.

내용

- 기본 사항
- 고려 사항
- 네트워크 인터페이스의 접두사 관리

기본 사항

- 새 네트워크 인터페이스나 기존 네트워크 인터페이스에 접두사를 할당할 수 있습니다.
- 접두사를 사용하려면 네트워크 인터페이스에 접두사를 할당하고 네트워크 인터페이스를 인스턴스에 연결한 다음, 운영 체제를 구성합니다.
- 접두사를 지정하는 옵션을 선택할 때는 접두사는 다음 요구 사항을 반드시 충족해야 합니다.
  - 지정할 수 있는 IPv4 접두사는 /28입니다.
  - 지정할 수 있는 IPv6 접두사는 /80입니다.
  - 접두사는 네트워크 인터페이스의 서브넷 CIDR에 있으며 서브넷의 기존 리소스에 할당된 다른 접두사 또는 IP 주소와 겹치지 않습니다.
- 기본 네트워크 인터페이스나 보조 네트워크 인터페이스에 접두사를 할당할 수 있습니다.

- 접두사가 할당된 네트워크 인터페이스에 탄력적 IP 주소를 할당할 수 있습니다.
- 할당된 접두사의 IP 주소 부분에 탄력적 IP 주소를 할당할 수도 있습니다.
- 인스턴스의 프라이빗 DNS 호스트 이름을 기본 프라이빗 IPv4 주소로 확인합니다.
- 접두사의 주소를 포함하여 네트워크 인터페이스에 각 프라이빗 IPv4 주소를 다음과 같은 형식을 사용하여 할당합니다.
  - us-east-1 리전

```
ip-private-ipv4-address.ec2.internal
```

  - 기타 모든 리전

```
ip-private-ipv4-address.region.compute.internal
```

고려 사항

접두사를 사용할 때는 다음 사항을 고려하세요.

- 접두사가 있는 네트워크 인터페이스는 Nitro 기반 인스턴스에서 지원됩니다.
- 네트워크 인터페이스의 접두사는 IPv6 주소 및 프라이빗 IPv4 주소로 제한됩니다.
- 네트워크 인터페이스에 할당할 수 있는 IP 주소의 최대 개수는 인스턴스 유형에 따라 다릅니다. 네트워크 인터페이스에 할당하는 각 접두사는 하나의 IP 주소로 계산됩니다. 예를 들어 c5.large 인스턴스에는 네트워크 인터페이스당 10 개의 IPv4 주소 제한이 있습니다. 이 인스턴스의 각 네트워크 인터페이스에는 기본 IPv4 주소가 있습니다. 네트워크 인터페이스에 보조 IPv4 주소가 없는 경우 네트워크 인터페이스에 최대 9개의 접두사를 할당할 수 있습니다. 네트워크 인터페이스에 할당한 각 추가 IPv4 주소의 경우 네트워크 인터페이스에 접두사를 하나 더 적게 할당할 수 있습니다. 자세한 내용은 네트워크 인터페이스당 최대 IP 주소 섹션을 참조하세요.
- 접두사는 원본/대상 확인에 포함됩니다.
- 접두사가 있는 네트워크 인터페이스와 작동하도록 운영 체제를 구성해야 합니다. 다음 사항에 유의하세요.
  - 일부 Amazon Linux AMI는 AWS에서 설치한 추가 스크립트(ec2-net-utils)를 포함합니다. 이러한 스크립트는 네트워크 인터페이스의 구성을 선택적으로 구성합니다. Amazon Linux에서만 사용할 수 있습니다.
  - 컨테이너의 경우 Kubernetes 플러그인용 컨테이너 네트워크 인터페이스(CNI)를 사용합니다. 또는 Docker를 사용하여 컨테이너를 관리하는 경우 dockerd를 사용할 수 있습니다.

네트워크 인터페이스의 접두사 관리

다음과 같이 네트워크 인터페이스에 접두사를 관리할 수 있습니다.

업무

- 네트워크 인터페이스 생성 중에 접두사 할당
- 기존 네트워크 인터페이스에 접두사 할당
- 네트워크 인터페이스에서 접두사 제거

네트워크 인터페이스 생성 중에 접두사 할당

네트워크 인터페이스를 생성할 때 자동 또는 사용자 지정 접두사를 할당할 수 있습니다.

Console

네트워크 인터페이스 생성 중에 자동 접두사를 할당하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.
3. 네트워크 인터페이스 생성을 선택합니다.
4. 네트워크 인터페이스에 대한 설명을 입력하고 네트워크 인터페이스를 생성할 서브넷을 선택한 다음 프라이빗 IPv4 및 IPv6 주소를 구성합니다.
5. Advanced settings(고급 설정)를 확장합니다.
6. IPv4 접두사 위임에 대해 다음 중 하나를 수행합니다.
   - IPv4 접두사를 자동으로 할당하려면 자동 할당을 선택합니다. IPv4 접두사 수에 할당할 접두사 수를 입력합니다.
   - 특정 IPv4 접두사를 할당하려면 사용자 지정을 선택합니다. 새 접두사 추가를 선택하고 접두사를 입력합니다.
7. IPv6 접두사 위임에 대해 다음 중 하나를 수행합니다.
   - IPv6 접두사를 자동으로 할당하려면 자동 할당을 선택합니다. IPv6 접두사 수에 할당할 접두사 수를 입력합니다.
   - 특정 IPv6 접두사를 할당하려면 사용자 지정을 선택합니다. 새 접두사 추가를 선택하고 접두사를 입력합니다.

> **Note**
>
> IPv6 접두사 위임(IPv6 prefix delegation)은 선택한 서브넷이 IPv6에 대해 사용되는 경우에만 표시됩니다.

8. 필요한 경우 네트워크 인터페이스와 연결할 보안 그룹을 선택하고 리소스 태그를 할당합니다.
9. 네트워크 인터페이스 생성을 선택합니다.

AWS CLI

네트워크 인터페이스 생성 중에 자동 IPv4 접두사를 할당하려면

create-network-interface 명령을 사용하여 `--ipv4-prefix-count`를 AWS에서 할당하려는 IPv4 접두사 수로 설정합니다. 다음 예제에서 AWS가 IPv4 접두사를 하나 할당합니다.

```
aws ec2 create-network-interface \
--subnet-id subnet-047cfed18eEXAMPLE \
--description "IPv4 automatic example" \
--ipv4-prefix-count 1
```

네트워크 인터페이스 생성 중에 특정 IPv4 접두사를 할당하려면

create-network-interface 명령을 사용하여 `--ipv4-prefixes`를 접두사로 설정합니다. AWS는 이 범위에서 IPv4 주소를 선택합니다. 다음 예제에서 접두사 CIDR은 10.0.0.208/28입니다.

```
aws ec2 create-network-interface \
    --subnet-id subnet-047cfed18eEXAMPLE \
    --description "IPv4 manual example" \
    --ipv4-prefixes Ipv4Prefix=10.0.0.208/28
```

네트워크 인터페이스 생성 중에 자동 IPv6 접두사를 할당하려면

create-network-interface 명령을 사용하여 `--ipv6-prefix-count`를 AWS에서 할당하려는 IPv6 접두사 수로 설정합니다. 다음 예제에서 AWS가 IPv6 접두사를 하나 할당합니다.

```
aws ec2 create-network-interface \
--subnet-id subnet-047cfed18eEXAMPLE \
--description "IPv6 automatic example" \
--ipv6-prefix-count 1
```

네트워크 인터페이스 생성 중에 특정 IPv6 접두사를 할당하려면

create-network-interface 명령을 사용하여 `--ipv6-prefixes`를 접두사로 설정합니다. AWS는 이 범위에서 IPv6 주소를 선택합니다. 다음 예제에서 접두사 CIDR은 2600:1f13:fc2:a700:1768::/80입니다.

```
aws ec2 create-network-interface \
    --subnet-id subnet-047cfed18eEXAMPLE \
    --description "IPv6 manual example" \
    --ipv6-prefixes Ipv6Prefix=2600:1f13:fc2:a700:1768::/80
```

PowerShell

네트워크 인터페이스 생성 중에 자동 IPv4 접두사를 할당하려면

New-EC2NetworkInterface cmdlet을 사용하고 `Ipv4PrefixCount`를 AWS에서 할당하려는 IPv4 접두사 수로 설정합니다. 다음 예제에서 AWS가 IPv4 접두사를 하나 할당합니다.

```
New-EC2NetworkInterface `
    -SubnetId 'subnet-047cfed18eEXAMPLE' `
    -Description 'IPv4 automatic example' `
    -Ipv4PrefixCount 1
```

네트워크 인터페이스 생성 중에 특정 IPv4 접두사를 할당하려면

New-EC2NetworkInterface cmdlet을 사용하고 `Ipv4Prefix`를 접두사로 설정합니다. AWS는 이 범위에서 IPv4 주소를 선택합니다. 다음 예제에서 접두사 CIDR은 10.0.0.208/28입니다.

```
Import-Module AWS.Tools.EC2
New-EC2NetworkInterface `
    -SubnetId 'subnet-047cfed18eEXAMPLE' `
    -Description 'IPv4 manual example' `
    -Ipv4Prefix (New-Object `
        -TypeName Amazon.EC2.Model.Ipv4PrefixSpecificationRequest `
        -Property @{Ipv4Prefix = '10.0.0.208/28'})
```

네트워크 인터페이스 생성 중에 자동 IPv6 접두사를 할당하려면

New-EC2NetworkInterface cmdlet을 사용하고 `Ipv6PrefixCount`를 AWS에서 할당하려는 IPv6 접두사 수로 설정합니다. 다음 예제에서 AWS가 IPv6 접두사를 하나 할당합니다.

```
New-EC2NetworkInterface `
```

```
    -SubnetId 'subnet-047cfed18eEXAMPLE' `
    -Description 'IPv6 automatic example' `
    -Ipv6PrefixCount 1
```

네트워크 인터페이스 생성 중에 특정 IPv6 접두사를 할당하려면

New-EC2NetworkInterface cmdlet을 사용하고 `Ipv6Prefixes`를 접두사로 설정합니다. AWS는 이 범위에서 IPv6 주소를 선택합니다. 다음 예제에서 접두사 CIDR은 2600:1f13:fc2:a700:1768::/80입니다.

```
Import-Module AWS.Tools.EC2
New-EC2NetworkInterface `
    -SubnetId 'subnet-047cfed18eEXAMPLE' `
    -Description 'IPv6 manual example' `
    -Ipv6Prefix (New-Object `
        -TypeName Amazon.EC2.Model.Ipv6PrefixSpecificationRequest `
        -Property @{Ipv6Prefix = '2600:1f13:fc2:a700:1768::/80'})
```

기존 네트워크 인터페이스에 접두사 할당

기존 네트워크 인터페이스에 자동 또는 사용자 지정 접두사를 할당할 수 있습니다.

Console

기존 네트워크 인터페이스에 자동 접두사를 할당하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.
3. 접두사를 할당할 네트워크 인터페이스를 선택하고 작업(Actions), 접두사 관리(Manage prefixes)를 선택합니다.
4. IPv4 접두사 위임에 대해 다음 중 하나를 수행합니다.
   - IPv4 접두사를 자동으로 할당하려면 자동 할당을 선택합니다. IPv4 접두사 수에 할당할 접두사 수를 입력합니다.
   - 특정 IPv4 접두사를 할당하려면 사용자 지정을 선택합니다. 새 접두사 추가를 선택하고 접두사를 입력합니다.
5. IPv6 접두사 위임에 대해 다음 중 하나를 수행합니다.
   - IPv6 접두사를 자동으로 할당하려면 자동 할당을 선택합니다. IPv6 접두사 수에 할당할 접두사 수를 입력합니다.

- 특정 IPv6 접두사를 할당하려면 사용자 지정을 선택합니다. 새 접두사 추가를 선택하고 접두사를 입력합니다.

> **Note**
>
> IPv6 접두사 위임(IPv6 prefix delegation)은 선택한 서브넷이 IPv6에 대해 사용되는 경우에만 표시됩니다.

6. 저장을 선택합니다.

AWS CLI

assign-ipv6-addresses 명령을 사용하여 IPv6 접두사를 할당하고 assign-private-ip-address 명령을 사용하여 IPv4 접두사를 기존 네트워크 인터페이스에 할당합니다.

기존 네트워크 인터페이스에 자동 IPv4 접두사를 할당하려면

assign-private-ip-address 명령을 사용하여 `--ipv4-prefix-count`를 AWS에서 할당하려는 IPv4 접두사 수로 설정합니다. 다음 예제에서 AWS가 IPv4 접두사를 하나 할당합니다.

```
aws ec2 assign-private-ip-addresses \
--network-interface-id eni-081fbb4095EXAMPLE \
--ipv4-prefix-count 1
```

기존 네트워크 인터페이스에 특정 IPv4 접두사를 할당하려면 다음을 수행하세요.

assign-private-ip-address 명령을 사용하여 `--ipv4-prefixes`를 접두사로 설정합니다. AWS는 이 범위에서 IPv4 주소를 선택합니다. 다음 예제에서 접두사 CIDR은 10.0.0.208/28입니다.

```
aws ec2 assign-private-ip-addresses \
--network-interface-id eni-081fbb4095EXAMPLE \
--ipv4-prefixes 10.0.0.208/28
```

기존 네트워크 인터페이스에 자동 IPv6 접두사를 할당하려면

assign-ipv6-addresses 명령을 사용하여 `--ipv6-prefix-count`를 AWS에서 할당하려는 IPv6 접두사 수로 설정합니다. 다음 예제에서 AWS가 IPv6 접두사를 하나 할당합니다.

```
aws ec2 assign-ipv6-addresses \
--network-interface-id eni-00d577338cEXAMPLE \
```

```
--ipv6-prefix-count 1
```

기존 네트워크 인터페이스에 특정 IPv6 접두사를 할당하려면 다음을 수행하세요.

assign-ipv6-addresses 명령을 사용하여 `--ipv6-prefixes`를 접두사로 설정합니다. AWS는 이 범위에서 IPv6 주소를 선택합니다. 다음 예제에서 접두사 CIDR은 2600:1f13:fc2:a700:18bb::/80입니다.

```
aws ec2 assign-ipv6-addresses \
--network-interface-id eni-00d577338cEXAMPLE \
--ipv6-prefixes 2600:1f13:fc2:a700:18bb::/80
```

PowerShell

기존 네트워크 인터페이스에 자동 IPv4 접두사를 할당하려면

Register-EC2PrivateIpAddress cmdlet을 사용하고 `Ipv4PrefixCount`를 AWS에서 할당하려는 IPv4 접두사 수로 설정합니다. 다음 예제에서 AWS가 IPv4 접두사를 하나 할당합니다.

```
Register-EC2PrivateIpAddress `
    -NetworkInterfaceId 'eni-00d577338cEXAMPLE' `
    -Ipv4PrefixCount 1
```

기존 네트워크 인터페이스에 특정 IPv4 접두사를 할당하려면 다음을 수행하세요.

Register-EC2PrivateIpAddress cmdlet을 사용하고 `Ipv4Prefix`를 접두사로 설정합니다. AWS는 이 범위에서 IPv4 주소를 선택합니다. 다음 예제에서 접두사 CIDR은 10.0.0.208/28입니다.

```
Register-EC2PrivateIpAddress `
    -NetworkInterfaceId 'eni-00d577338cEXAMPLE' `
    -Ipv4Prefix '10.0.0.208/28'
```

기존 네트워크 인터페이스에 자동 IPv6 접두사를 할당하려면

Register-EC2Ipv6AddressList cmdlet을 사용하고 `Ipv6PrefixCount`를 AWS에서 할당하려는 IPv4 접두사 수로 설정합니다. 다음 예제에서 AWS가 IPv6 접두사를 하나 할당합니다.

```
Register-EC2Ipv6AddressList `
    -NetworkInterfaceId 'eni-00d577338cEXAMPLE' `
    -Ipv6PrefixCount 1
```

기존 네트워크 인터페이스에 특정 IPv6 접두사를 할당하려면 다음을 수행하세요.

Register-EC2Ipv6AddressList cmdlet을 사용하고 `Ipv6Prefix`를 접두사로 설정합니다. AWS는 이 범위에서 IPv6 주소를 선택합니다. 다음 예제에서 접두사 CIDR은 2600:1f13:fc2:a700:18bb::/80입니다.

```
Register-EC2Ipv6AddressList `
    -NetworkInterfaceId 'eni-00d577338cEXAMPLE' `
    -Ipv6Prefix '2600:1f13:fc2:a700:18bb::/80'
```

네트워크 인터페이스에서 접두사 제거

기존 네트워크 인터페이스에서 접두사를 제거할 수 있습니다.

Console

네트워크 인터페이스에서 접두사를 제거하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.
3. 네트워크 인터페이스를 선택합니다.
4. 작업, 접두사 관리를 선택합니다.
5. IPv4 접두사 위임에서 특정 접두사를 제거하려면 제거할 접두사 옆에 있는 할당 취소를 선택합니다. 모든 접두사를 제거하려면 할당하지 않음을 선택합니다.
6. IPv6 접두사 위임에서 특정 접두사를 제거하려면 제거할 접두사 옆에 있는 할당 취소를 선택합니다. 모든 접두사를 제거하려면 할당하지 않음을 선택합니다.

   > **Note**
   >
   > IPv6 접두사 위임(IPv6 prefix delegation)은 선택한 서브넷이 IPv6에 대해 사용되는 경우에만 표시됩니다.

7. 저장을 선택합니다.

AWS CLI

unassign-ipv6-addresses 명령을 사용하여 IPv6 접두사를 제거하고 unassign-private-ip-address 명령을 사용하여 기존 네트워크 인터페이스에서 IPv4 접두사를 제거합니다.

네트워크 인터페이스에서 IPv4 접두사를 제거하려면

unassign-private-ip-address 명령을 사용하고 `--ipv4-prefix`를 제거하려는 접두사 CIDR로 설정합니다.

```
aws ec2 unassign-private-ip-addresses \
--network-interface-id eni-081fbb4095EXAMPLE \
--ipv4-prefixes 10.0.0.176/28
```

네트워크 인터페이스에서 IPv6 접두사를 제거하려면

unassign-ipv6-address 명령을 사용하고 `--ipv6-prefix`를 제거하려는 접두사 CIDR로 설정합니다.

```
aws ec2 unassign-ipv6-addresses \
--network-interface-id eni-00d577338cEXAMPLE \
--ipv6-prefix 2600:1f13:fc2:a700:18bb::/80
```

PowerShell

네트워크 인터페이스에서 IPv4 접두사를 제거하려면

Unregister-EC2PrivateIpAddress cmdlet을 사용하고 `Ipv4Prefix`를 제거하려는 접두사 CIDR로 설정합니다.

```
Unregister-EC2PrivateIpAddress `
    -NetworkInterfaceId 'eni-00d577338cEXAMPLE' `
    -Ipv4Prefix '10.0.0.208/28'
```

네트워크 인터페이스에서 IPv6 접두사를 제거하려면

Unregister-EC2Ipv6AddressList cmdlet을 사용하고 `Ipv6Prefix`를 제거하려는 접두사 CIDR로 설정합니다.

```
Unregister-EC2Ipv6AddressList `
    -NetworkInterfaceId 'eni-00d577338cEXAMPLE' `
    -Ipv6Prefix '2600:1f13:fc2:a700:18bb::/80'
```

네트워크 인터페이스 삭제

네트워크 인터페이스를 삭제하면 인터페이스와 연결된 모든 속성이 해제되고 다른 인스턴스에서 사용할 수 있도록 프라이빗 IP 주소나 탄력적 IP 주소가 해제됩니다.

사용 중인 네트워크 인터페이스는 삭제할 수 없습니다. 먼저 네트워크 인터페이스를 분리해야 합니다.

Console

네트워크 인터페이스 삭제

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.
3. 네트워크 인터페이스에 대한 확인란을 선택한 후 [작업], [삭제]를 선택합니다.
4. 확인 메시지가 나타나면 삭제를 선택합니다.

AWS CLI

네트워크 인터페이스 삭제

다음 delete-network-interface 명령을 사용합니다.

```
aws ec2 delete-network-interface --network-interface-id eni-1234567890abcdef0
```

PowerShell

네트워크 인터페이스 삭제

Remove-EC2NetworkInterface cmdlet을 사용합니다.

```
Remove-EC2NetworkInterface -NetworkInterfaceId eni-1234567890abcdef0
```

Amazon EC2 인스턴스 네트워크 대역폭

인스턴스 대역폭 사양은 인스턴스의 인바운드 트래픽과 아웃바운드 트래픽에 모두 적용됩니다. 예를 들어 인스턴스가 최대 10Gbps의 대역폭을 지정하는 경우 인바운드 트래픽에 대해 최대 10Gbps의 대역폭 및 이와 동시에 아웃바운드 트래픽에 대해 최대 10Gbps의 대역폭이 있음을 의미합니다. EC2 인스턴스에서 사용 가능한 네트워크 대역폭은 다음과 같이 몇 가지 요인에 따라 달라집니다.

다중 흐름 트래픽

다중 흐름 트래픽의 대역폭은 32개 이상의 vCPU 또는 5Gbps 중 더 큰 쪽을 사용하는 인스턴스의 경우 인터넷 게이트웨이 또는 로컬 게이트웨이를 통과하는 트래픽에 사용할 수 있는 대역폭의 50%로 제한됩니다. vCPUs가 32개 미만인 인스턴스의 경우 대역폭은 5Gbps로 제한됩니다.

단일 흐름 트래픽

단일 흐름 트래픽의 대역폭은 인스턴스가 동일한 클러스터 배치 그룹에 없을 때 5Gbps로 제한됩니다. 지연 시간을 줄이고 단일 흐름 대역폭을 늘리려면 다음 중 하나를 시도해 보세요.

- 클러스터 배치 그룹을 사용하여 동일한 배치 그룹 내의 인스턴스에 대해 최대 10Gbps의 대역폭을 확보할 수 있습니다.
- 다중 경로 TCP(MPTCP)로 두 엔드포인트 간에 여러 경로를 설정하여 더 높은 대역폭을 확보합니다.
- 동일한 가용 영역 내의 적격 인스턴스에 대해 해당 인스턴스 간에 최대 25Gbps를 확보하도록 ENA Express를 구성합니다.

> **Note**
>
> 단일 흐름은 고유한 5-튜플 TCP 또는 UDP 흐름으로 간주됩니다. GRE 또는 IPsec과 같이 IP 헤더 뒤에 오는 다른 프로토콜의 경우 소스 IP, 대상 IP, 다음 프로토콜의 3-튜플이 흐름 정의에 사용됩니다.

사용 가능한 인스턴스 대역폭

인스턴스의 사용 가능한 네트워크 대역폭은 인스턴스가 보유한 vCPU 수에 따라 달라집니다. 예를 들어,`m5.8xlarge` 인스턴스는 32개의 vCPU 및 10Gbps의 네트워크 대역폭을 보유하고 `m5.16xlarge` 인스턴스는 64개의 vCPU 및 20Gbps의 네트워크 대역폭을 보유합니다. 하지만 초당 패킷 수 또는 추적된 연결 수와 같은 인스턴스 수준의 네트워크 허용량을 초과하는 경우 인스턴스에서 이러한 대역폭을 달성하지 못할 수 있습니다. 트래픽에 활용할 수 있는 대역폭의 양은 vCPU 수와 대상에 따라 다릅니다. 예를 들어 `m5.16xlarge` 인스턴스에는 64개의 vCPU가 있으므로 리전의 다른 인스턴스로 전송되는 트래픽은 전체 가용 대역폭(20Gbps)을 활용할 수 있습니다. 하지만 인터넷 게이트웨이 또는 로컬 게이트웨이를 통과하는 트래픽에는 가용 대역폭의 50%(10Gbps)만 활용할 수 있습니다.

일반적으로 vCPU가 16개 이하인 인스턴스(크기: `4xlarge` 이하)는 지정된 대역폭을 '최대' 보유하는 것으로 문서화됩니다(예: '최대 10Gbps'). 이러한 인스턴스에는 기준 대역폭이 있습니다. 추가 요구 사항을 충족하려면 네트워크 I/O 크레딧 메커니즘을 통해 기준 대역폭을 초과하여 확장할 수 있습니다.

인스턴스는 인스턴스 크기에 따라 일반적으로 5분에서 60분까지 제한된 시간 동안 버스트 대역폭을 사용할 수 있습니다.

인스턴스는 시작할 때 최대 개수의 네트워크 I/O 크레딧을 받습니다. 인스턴스가 네트워크 I/O 크레딧을 모두 사용할 경우 기준 대역폭으로 돌아갑니다. 실행 중인 인스턴스는 기준 대역폭보다 적은 네트워크 대역폭을 사용할 때마다 네트워크 I/O 크레딧을 얻습니다. 중지된 인스턴스는 네트워크 I/O 크레딧을 얻지 못합니다. 버스트 대역폭은 공유 리소스이므로 인스턴스에 사용 가능한 크레딧이 있더라도 인스턴스 버스트는 가능한 범위에서 최대한 지원됩니다.

인바운드 트래픽과 아웃바운드 트래픽에 별도의 네트워크 I/O 크레딧 버킷이 있습니다.

기본 및 버스트 네트워크 성능

Amazon EC2 인스턴스 유형 안내서에서는 각 인스턴스 유형의 네트워크 성능과 버스트 대역폭을 사용할 수 있는 인스턴스의 사용 가능한 기준 네트워크 대역폭에 대해 설명합니다. 자세한 내용은 다음 자료를 참조하세요.

- 네트워크 사양 - 범용
- 네트워크 사양 - 컴퓨팅 최적화
- 네트워크 사양 - 메모리 최적화
- 네트워크 사양 - 스토리지 최적화
- 네트워크 사양 - 가속 컴퓨팅
- 네트워크 사양 - 고성능 컴퓨팅
- 네트워크 사양 - 이전 세대

다른 방식으로는 이 명령줄 도구를 사용하여 이 정보를 가져올 수 있습니다. Amazon EC2 콘솔에는 인스턴스 유형의 기준 네트워크 대역폭이 표시되지 않습니다.

AWS CLI

describe-instance-types 명령을 사용하여 인스턴스 유형에 대한 정보를 표시할 수 있습니다. 다음 예에서는 모든 C5 인스턴스에 대한 네트워크 성능 정보를 표시합니다.

```
aws ec2 describe-instance-types \
    --filters "Name=instance-type,Values=c5.*" \
    --query "InstanceTypes[].[InstanceType, NetworkInfo.NetworkPerformance,
 NetworkInfo.NetworkCards[0].BaselineBandwidthInGbps] | sort_by(@,&[2])" \
    --output table
```

출력의 예시는 다음과 같습니다. 출력이 기준 대역폭에 맞지 않는 경우 AWS CLI를 최신 버전으로 업데이트합니다.

```
-----------------------------------------------
|           DescribeInstanceTypes             |
+--------------+--------------------+---------+
|  c5.large    |  Up to 10 Gigabit  |  0.75 |
|  c5.xlarge   |  Up to 10 Gigabit  |  1.25 |
|  c5.2xlarge  |  Up to 10 Gigabit  |  2.5  |
|  c5.4xlarge  |  Up to 10 Gigabit  |  5.0  |
|  c5.9xlarge  |  12 Gigabit        |  12.0 |
|  c5.12xlarge |  12 Gigabit        |  12.0 |
|  c5.18xlarge |  25 Gigabit        |  25.0 |
|  c5.24xlarge |  25 Gigabit        |  25.0 |
|  c5.metal    |  25 Gigabit        |  25.0 |
+--------------+--------------------+---------+
```

PowerShell

Get-EC2InanceType PowerShell 명령을 사용하여 인스턴스 유형에 대한 정보를 표시할 수 있습니다. 다음 예에서는 모든 C5 인스턴스에 대한 네트워크 성능 정보를 표시합니다.

```
Get-EC2InstanceType -Filter @{Name = "instance-type"; Values = "c5.*" } | `
    Select-Object `
    InstanceType,
    @{Name = 'NetworkPerformance'; Expression =
 {($_.Networkinfo.NetworkCards.NetworkPerformance)}},
    @{Name = 'BaselineBandwidthInGbps'; Expression =
 {($_.Networkinfo.NetworkCards.BaselineBandwidthInGbps)}} | `
Format-Table -AutoSize
```

출력의 예시는 다음과 같습니다.

```
InstanceType NetworkPerformance BaselineBandwidthInGbps
------------ ------------------ ----------------------
c5.4xlarge   Up to 10 Gigabit                    5.00
c5.xlarge    Up to 10 Gigabit                    1.25
c5.12xlarge  12 Gigabit                         12.00
c5.9xlarge   12 Gigabit                         12.00
c5.24xlarge  25 Gigabit                         25.00
c5.metal     25 Gigabit                         25.00
c5.2xlarge   Up to 10 Gigabit                    2.50
```

```
c5.large     Up to 10 Gigabit                          0.75
c5.18xlarge  25 Gigabit                               25.00
```

EC2 인스턴스 대역폭 가중치 구성

일부 인스턴스 유형은 구성 가능한 대역폭 가중치를 지원하므로 네트워크 처리 또는 EBS 작업에 유리한 기준 대역폭 가중치를 선택할 수 있습니다. 기준 대역폭의 기본 설정은 인스턴스 유형에 따라 결정됩니다. 시작 중에 대역폭 가중치를 구성하거나 다음 가중치 기본 설정을 사용하여 인스턴스 설정을 수정할 수 있습니다.

- 기본값 - 이 옵션은 선택한 인스턴스 유형에 표준 대역폭 구성을 사용합니다.
- vpc-1 - 이 옵션은 네트워킹에 사용할 수 있는 기준 대역폭을 늘리고 EBS 작업에 사용할 기준 대역폭을 줄입니다.
- ebs-1 - 이 옵션은 EBS 작업에 사용할 수 있는 기준 대역폭을 늘리고 네트워킹에 사용할 기준 대역폭을 줄입니다.

대역폭 가중치 고려 사항

다음은 대역폭 가중치 전략에 영향을 미칠 수 있는 몇 가지 고려 사항입니다.

- 대역폭 가중치 기본 설정을 지정하면 대역폭 사양에만 영향을 미칩니다. 초당 네트워크 패킷(PPS) 및 초당 EBS 입출력 작업량(IOPS) 사양은 변경되지 않습니다.
- 네트워킹과 EBS 간의 결합된 대역폭 사양은 변경되지 않습니다. 대역폭 가중치 구성을 선택하면 선택한 옵션에 사용할 수 있는 기준 대역폭이 증가하고 나머지 옵션의 기준 대역폭이 동일한 절대 양만큼 줄어듭니다. 사용 가능한 버스트 대역폭은 선택한 옵션에는 동일하게 유지되고 나머지 옵션에는 줄어듭니다.
- 대역폭 할당의 변화가 EBS의 I/O 성능에 어떤 영향을 미칠 수 있는지 이해하는 것이 중요합니다. `vpc-1` 구성(네트워크 대역폭 증가)이 있는 EC2 인스턴스의 경우 IOPS 제한에 도달하기 전에 EBS 대역폭 제한에 도달하면 EBS 볼륨의 IOPS가 낮아질 수 있습니다. IOPS 저하는 I/O 크기가 클수록 더 두드러집니다.

  예를 들어 일반적으로 16KiB I/O 크기의 240,000 IOPS를 지원하는 인스턴스 유형에서 `vpc-1` 가중치를 선택하면 조정된 EBS 기준 대역폭 제한으로 인해 달성 가능한 IOPS가 낮아질 수 있습니다.

  워크로드를 계획할 때는 I/O 크기와 패턴을 고려하세요. I/O 크기가 작을수록 대역폭 제한의 영향을 덜 받을 수 있지만 I/O 크기나 순차적 워크로드가 클수록 대역폭 변경으로 인해 더 많은 영향을 받을

수 있습니다. 선택한 구성에서 최적의 성능을 유지하기 위해 항상 사용하려는 워크로드를 테스트하세요.

- 인터넷 게이트웨이 또는 로컬 게이트웨이를 통과하는 트래픽에 대한 네트워킹 다중 흐름 대역폭 사양은 구성된 옵션의 기준 대역폭의 50% 또는 해당하는 경우 5Gbps로 조정됩니다. 자세한 내용은 Amazon EC2 인스턴스 네트워크 대역폭 섹션을 참조하세요.

  다음 예제는 기본 기준 대역폭이 40Gbps이고 기본 경계 대역폭이 20Gbps인 인스턴스 유형을 기반으로 합니다. 이 인스턴스에 대해 `vpc-1` 대역폭 가중치를 선택하면 가중치 기준 대역폭이 50Gbps로 변경되고 경계 대역폭이 25Gbps로 변경됩니다.
- 이 기능은 EC2 인스턴스 가용성 및 지원에 따라 모든 상용 리전에서 사용할 수 있습니다.
- 이 기능을 사용할 경우 EC2 인스턴스에 추가 비용이 발생하지 않습니다.

대역폭 가중치가 지원되는 인스턴스 유형

다음 인스턴스 패밀리의 인스턴스 유형에서는 구성 가능한 대역폭 가중치가 지원됩니다.

- 범용: M8g, M8gd
- 컴퓨팅 최적화: C8g, C8gd
- 메모리 최적화: R8g, R8gd, X8g

현재 대역폭 설정 확인

인스턴스의 현재 대역폭 설정을 보려면 지침에 대한 탭 중 하나를 선택합니다.

Console

인스턴스의 대역폭 설정을 가져오는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 목록에서 확인할 인스턴스를 선택하고 네트워킹 탭으로 이동합니다. 현재 설정이 구성된 대역폭 필드에 표시됩니다. Amazon EC2는 대역폭이 특정 값으로 설정되지 않은 경우 인스턴스 유형에 기본 설정을 사용합니다.

AWS CLI

인스턴스의 대역폭 설정을 가져오는 방법

describe-instances 명령을 사용합니다.

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query Reservations[].Instances[].NetworkPerformanceOptions.BandwidthWeighting
 \
    --output text
```

출력의 예시는 다음과 같습니다.

```
default
```

이 예제에서는 더 높은 네트워킹 대역폭을 위해 대역폭 가중치 기본 설정이 vpc-1로 설정된 모든 인스턴스를 나열합니다.

```
aws ec2 describe-instances \
    --filters "Name=network-performance-options.bandwidth-weighting,Values=vpc-1" \
    --query Reservations[].Instances[].InstanceId \
    --output text
```

PowerShell

인스턴스의 대역폭 설정을 가져오는 방법

Get-EC2Instance cmdlet을 사용합니다.

```
(Get-EC2Instance `
    -
InstanceId i-1234567890abcdef0).Instances.NetworkPerformanceOptions.BandwidthWeighting.Value
```

출력의 예시는 다음과 같습니다.

```
default
```

이 예제에서는 더 높은 네트워킹 대역폭을 위해 대역폭 가중치 기본 설정이 vpc-1로 설정된 모든 인스턴스를 나열합니다.

```
(Get-EC2Instance `
    -Filter @{Name="network-performance-options.bandwidth-
weighting";Values="vpc-1"}).Instances.InstanceId
```

인스턴스의 대역폭 가중치 구성

시작 시 대역폭 가중치를 구성하거나 EC2 콘솔, API/SDKs 또는 CLI의 기존 인스턴스를 수정하여 대역폭 가중치를 구성할 수 있습니다.

인스턴스를 시작할 때 대역폭 가중치 구성

인스턴스를 시작할 때 대역폭 설정을 구성하려면 지침에 대한 탭 중 하나를 선택합니다.

시작 템플릿에서 대역폭 가중치를 지정할 수도 있습니다. 시작 템플릿을 생성하려면 Amazon EC2 시작 템플릿 생성을 참조하세요. 설정할 파라미터는 콘솔에서 직접 인스턴스를 시작하기 위한 파라미터와 동일한 위치에 있습니다. 고급 세부 정보 섹션을 확장하고 인스턴스 대역폭 구성을 설정합니다.

시작 템플릿으로 인스턴스를 시작하려면 시작 템플릿을 사용하여 EC2 인스턴스 시작 섹션을 참조하세요.

Console

구성 가능한 대역폭 가중치로 인스턴스를 시작하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스 시작(Launch Instances)을 선택합니다. 인스턴스 시작 대화 상자가 열립니다. 기본 설정에 따라 시작 대화 상자로 이동할 수 있는 몇 가지 추가 방법이 있습니다. 예를 들어 AMI 또는 Amazon EC2 대시보드 자체에서 직접 인스턴스를 시작할 수 있습니다.
4. 시작하려는 Amazon Machine Image(AMI)는 `Arm` 아키텍처를 기반으로 해야 합니다. 많은 빠른 시작 이미지는 `x86` 및 `Arm` 아키텍처를 모두 지원합니다. 인스턴스의 운영 체제를 선택한 후 아키텍처 목록에서 `Arm` 옵션을 선택합니다.
5. 이 기능의 인스턴스 유형은 지원되는 인스턴스 유형 중 하나여야 합니다.
6. 고급 세부 정보 섹션을 확장한 다음 아래로 스크롤하여 인스턴스 대역폭 구성 설정을 찾을 수 있습니다. 인스턴스의 대역폭 구성 옵션을 선택합니다.
7. 평소처럼 인스턴스에 대한 다른 모든 설정을 구성하고 인스턴스 시작을 선택합니다.

AWS CLI

구성 가능한 대역폭 가중치로 인스턴스를 시작하는 방법

run-instances 명령을 다음 옵션과 함께 사용하여 더 높은 네트워크 대역폭 가중치를 위해 구성된 인스턴스를 시작합니다.

```
--network-performance-options BandwidthWeighting=vpc-1
```

run-instances 명령을 다음 옵션과 함께 사용하여 더 높은 EBS 대역폭 가중치를 위해 구성된 인스턴스를 시작합니다.

```
--network-performance-options BandwidthWeighting=ebs-1
```

PowerShell

구성 가능한 대역폭 가중치로 인스턴스를 시작하는 방법

New-EC2Instance cmdlet을 다음 파라미터와 함께 사용하여 더 높은 네트워크 대역폭 가중치를 위해 구성된 인스턴스를 시작합니다.

```
-NetworkPerformanceOptions_BandwidthWeighting vpc-1
```

New-EC2Instance cmdlet을 다음 파라미터와 함께 사용하여 더 높은 EBS 대역폭 가중치를 위해 구성된 인스턴스를 시작합니다.

```
-NetworkPerformanceOptions_BandwidthWeighting ebs-1
```

기존 인스턴스의 대역폭 가중치 업데이트

기존 인스턴스의 대역폭 가중치를 업데이트하려면 인스턴스가 Stopped 상태여야 합니다.

Console

대역폭 가중치를 업데이트하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 목록에서 업데이트할 인스턴스를 선택합니다.
4. 대역폭 구성을 변경하기 전에 인스턴스가 Stopped 상태여야 합니다. 인스턴스가 실행 중인 경우 인스턴스 상태 메뉴에서 인스턴스 중지를 선택합니다.
5. 작업 > 네트워킹 메뉴에서 대역폭 관리를 선택합니다. 그러면 대역폭 관리 대화 상자가 열립니다.

> **Note**
>
> 인스턴스 유형이 대역폭 가중치 구성을 지원하지 않는 경우 해당 메뉴 항목이 비활성화됩니다.

6. 옵션을 선택하여 인스턴스를 업데이트하고 변경을 선택하여 설정을 저장합니다.

AWS CLI

대역폭 가중치를 업데이트하는 방법

modify-instance-network-performance-options 명령을 사용하여 지정된 인스턴스에 대해 더 높은 네트워크 대역폭 가중치를 구성합니다.

```
aws ec2 modify-instance-network-performance-options \
    --instance-id i-1234567890abcdef0 \
    --bandwidth-weighting=vpc-1
```

다음 예제에서는 지정된 인스턴스에 대해 더 높은 EBS 대역폭 가중치를 구성합니다.

```
aws ec2 modify-instance-network-performance-options \
    --instance-id i-1234567890abcdef0 \
    --bandwidth-weighting=ebs-1
```

PowerShell

대역폭 가중치를 업데이트하는 방법

Edit-EC2InstanceNetworkPerformanceOption cmdlet을 사용하여 지정된 인스턴스에 대해 더 높은 네트워크 대역폭 가중치를 구성합니다.

```
Edit-EC2InstanceNetworkPerformanceOption `
    -InstanceId i-1234567890abcdef0 `
    -BandwidthWeighting vpc-1
```

다음 예제에서는 지정된 인스턴스에 대해 더 높은 EBS 대역폭 가중치를 구성합니다.

```
Edit-EC2InstanceNetworkPerformanceOption `
    -InstanceId i-1234567890abcdef0 `
```

```
    -BandwidthWeighting ebs-1
```

네트워킹에 대한 대역폭 가중치 영향

다음 표는 대역폭 가중치가 지원되는 인스턴스 패밀리의 네트워킹 대역폭에 미치는 영향을 보여줍니다.

인스턴스 크기	기본 대역폭(Gbps) 기준/버스트	vpc-1 기준/버스트	ebs-1 기준/버스트
.medium	0.52 / 12.5	0.65 / 12.5	0.4 / 10
.large	0.94 / 12.5	1.17 / 12.5	0.8 / 10
.xlarge	1.88 / 12.5	2.35 / 12.5	1.6 / 10
.2xlarge	3.75 / 15	4.69 / 15	3.1 / 12.5
.4xlarge	7.5 / 15	9.38 / 15	6.3 / 12.5
.8xlarge	15	18.75	12.5
.12xlarge	22.5	28.13	18.8
.16xlarge	30	37.5	25
.24xlarge	40	50	32.5
.48xlarge	50	62.5	40
.metal-24xl	40	50	32.5
.metal-48xl	50	62.5	40

EBS에 대한 대역폭 가중치 영향

다음 표는 대역폭 가중치가 EBS 작업에 사용할 수 있는 지원되는 인스턴스 패밀리의 대역폭에 미치는 영향을 보여줍니다.

인스턴스 크기	기본 대역폭(Gbps) 기준/버스트	vpc-1 기준/버스트	ebs-1 기준/버스트
.medium	0.3 / 10	0.2 / 6.3	0.4 / 10
.large	0.6 / 10	0.4 / 6.3	0.8 / 10
.xlarge	1.3 / 10	0.8 / 6.3	1.6 / 10
.2xlarge	2.5 / 10	1.6 / 6.3	3.1 / 10
.4xlarge	5.0 / 10	3.1 / 6.3	6.3 / 10
.8xlarge	10	6.3	12.5
.12xlarge	15	9.4	18.8
.16xlarge	20	12.5	25
.24xlarge	30	20	37.5
.48xlarge	40	27.5	50
.metal-24xl	30	20	37.5
.metal-48xl	40	27.5	50

인스턴스 대역폭 모니터링

CloudWatch 지표를 사용하여 인스턴스 네트워크 대역폭과 송수신 패킷을 모니터링할 수 있습니다. Elastic Network Adapter(ENA) 드라이버에서 제공하는 네트워크 성능 지표를 사용하여 트래픽이 Amazon EC2 인스턴스 수준에서 정의한 네트워크 허용량을 초과하는 시기를 모니터링할 수 있습니다.

Amazon EC2에서 1분 또는 5분 기간을 사용하여 인스턴스에 대한 지표 데이터를 CloudWatch로 전송할지 여부를 구성할 수 있습니다. CloudWatch 인스턴스 지표와 달리 네트워크 성능 지표에는 허용량이 초과되었고 패킷이 삭제되었다고 표시될 수 있습니다. 네트워크 리소스 수요에 단기 스파이크(마이크로버스트)가 발생하지만 CloudWatch 지표가 이러한 마이크로초 스파이크를 반영하도록 충분히 세분화되지 않은 경우 이러한 현상이 발생할 수 있습니다.

자세히 알아보기

- 인스턴스 지표
- 네트워크 성능 모니터링

Amazon EC2 인스턴스의 향상된 네트워킹

향상된 네트워킹에서는 지원되는 인스턴스 유형에서 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.

각 인스턴스 유형에 대해 지원되는 네트워크 속도에 대한 자세한 내용은 Amazon EC2 인스턴스 유형을 참조하세요.

다음 메커니즘 중 하나를 사용하여 향상된 네트워킹을 활성화할 수 있습니다.

ENA(Elastic Network Adapter)

탄력적 네트워크 어댑터(ENA)는 지원되는 인스턴스 유형에 대해 최대 100Gbps의 네트워크 속도를 지원합니다.

모든 Nitro 기반 인스턴스는 향상된 네트워킹을 위해 ENA를 사용합니다. 또한 H1, I3, G3, `m4.16xlarge`, P2, P3, P3dn, R4와 같은 Xen 기반 인스턴스도 ENA를 사용합니다.

자세한 내용은 EC2 인스턴스에서 ENA로 향상된 네트워킹 활성화 섹션을 참조하세요.

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 인스턴스 유형은 향상된 네트워킹을 위해 Intel 82599 VF 인터페이스를 사용합니다.

자세한 내용은 Intel 82599 VF 인터페이스를 통한 향상된 네트워킹 섹션을 참조하세요.

내용

- EC2 인스턴스에서 ENA로 향상된 네트워킹 활성화

- ENA Express로 EC2 인스턴스 간 네트워크 성능 향상
- Intel 82599 VF 인터페이스를 통한 향상된 네트워킹
- EC2 인스턴스의 ENA 설정에 대한 네트워크 성능 모니터링
- Linux에서 ENA 커널 드라이버 문제 해결
- Elastic Network Adapter Windows 드라이버 문제 해결
- Linux 기반 EC2 인스턴스의 네트워크 지연 시간 개선
- 성능 튜닝을 위한 Nitro 시스템 고려 사항
- EC2 Windows 인스턴스에서 네트워크 성능 최적화

EC2 인스턴스에서 ENA로 향상된 네트워킹 활성화

Amazon EC2는 ENA(Elastic Network Adapter)를 통해 향상된 네트워킹을 제공합니다. 향상된 네트워킹을 사용하려면 필수 ENA 드라이버가 포함된 AMI를 사용하거나 이를 수동으로 설치해야 합니다. 그러면 인스턴스에서 ENA 지원을 활성화할 수 있습니다.

ENA 드라이버의 릴리스 정보나 설치 지침을 검토하려면 인스턴스 운영 체제 플랫폼과 일치하는 탭을 참조하세요.

Linux

GitHub에서 다음 문서를 검토할 수 있습니다.

- GitHub에서 ENA Linux kernel driver release notes를 검토하세요.
- 설치 지침을 포함한 ENA Linux 커널 드라이버 개요는 GitHub에서 Linux kernel driver for Elastic Network Adapter (ENA) family를 참조하세요.

Windows

이 안내서의 디바이스 드라이버 관리 섹션에서 다음 문서를 검토할 수 있습니다.

- ENA Windows 드라이버 버전 릴리스 추적.
- EC2 Windows 인스턴스에 ENA 드라이버 설치.

Nitro 기반 인스턴스의 경우 향상된 네트워킹 기능은 인스턴스 유형이 구현하는 Nitro 버전에 따라 다릅니다.

인스턴스의 네트워크 사양을 검토하려면 인스턴스 유형에 맞는 인스턴스 패밀리 링크를 선택하세요. 어떤 인스턴스 패밀리가 적용되는지 확실하지 않은 경우 Amazon EC2 인스턴스 유형 안내서의 이름 지정 규칙을 참조하세요.

- 가속 컴퓨팅 인스턴스의 네트워크 사양
- 컴퓨팅 최적화 인스턴스의 네트워크 사양
- 범용 인스턴스의 네트워크 사양
- 고성능 컴퓨팅 인스턴스의 네트워크 사양
- 메모리 최적화 인스턴스의 네트워크 사양
- 스토리지 최적화 인스턴스의 네트워크 사양

내용

- ENA를 사용한 향상된 네트워킹의 사전 조건
- 향상된 네트워킹 기능 활성화 여부 테스트
- 인스턴스에서 향상된 네트워킹 기능 활성화

ENA를 사용한 향상된 네트워킹의 사전 조건

ENA를 사용하여 향상된 네트워킹을 준비하려면 인스턴스를 다음과 같이 설정하세요.

- Nitro 기반 인스턴스를 시작합니다.
- 인스턴스가 인터넷에 연결되어 있는지 확인합니다.
- 인스턴스에 보존해야 할 중요한 데이터가 있는 경우 인스턴스에서 AMI를 만들어 데이터를 백업해야 합니다. ENA 커널 드라이버를 업데이트하고 `enaSupport` 속성을 활성화하면 호환되지 않는 인스턴스나 운영 체제에 접속할 수 없게 됩니다. 최신 백업을 확보하면 이 경우에도 데이터를 보존할 수 있습니다.
- Linux 인스턴스 – 지원되는 Linux 커널 버전과 지원되는 배포판을 사용하여 인스턴스를 시작하면 인스턴스에 대해 ENA 향상된 네트워킹이 자동으로 사용 설정됩니다. 자세한 내용은 ENA Linux Kernel Driver 릴리스 정보를 참조하세요.
- Windows 인스턴스 - 인스턴스가 Windows Server 2008 R2 SP1을 실행 중인 경우 SHA-2 코드 서명 지원 업데이트가 있는지 확인합니다.
- AWS Management Console에서 AWS CloudShell을(를) 사용하거나 선택한 컴퓨터에 AWS CLI 또는 AWS Tools for Windows PowerShell을(를) 설치하고 구성합니다(로컬 데스크톱/노트북 권장). 자

세한 내용은 Amazon EC2 액세스 또는 AWS CloudShell 사용 설명서를 참조하세요. Amazon EC2 콘솔에서는 향상된 네트워킹을 관리할 수 없습니다.

향상된 네트워킹 기능 활성화 여부 테스트

인스턴스 또는 AMI에서 향상된 네트워킹이 활성화되어 있는지 테스트할 수 있습니다.

인스턴스 속성

`enaSupport` 인스턴스 속성의 값을 확인합니다.

AWS CLI

describe-instances 명령을 사용합니다.

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

향상된 네트워킹이 활성화된 경우 출력은 다음과 같습니다.

```
[
    true
]
```

PowerShell

Get-EC2Instance cmdlet을 사용하세요.

```
(Get-EC2Instance -InstanceId i-1234567890abcdef0).Instances.EnaSupport
```

향상된 네트워킹이 활성화된 경우 출력은 다음과 같습니다.

```
True
```

이미지 속성

`enaSupport` 이미지 속성의 값을 확인합니다.

AWS CLI

describe-images 명령을 사용합니다.

```
aws ec2 describe-images \
    --image-id ami-0abcdef1234567890 \
    --query "Images[].EnaSupport"
```

향상된 네트워킹이 활성화된 경우 출력은 다음과 같습니다.

```
[
    true
]
```

PowerShell

Get-EC2Image cmdlet을 사용하세요.

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).EnaSupport
```

향상된 네트워킹이 활성화된 경우 출력은 다음과 같습니다.

```
True
```

Linux 네트워크 인터페이스 드라이버

다음 명령과 확인하고자 하는 인터페이스 이름을 사용하여 해당 인터페이스에서 ena 커널 드라이버가 사용되고 있는지를 확인할 수 있습니다. 단일 인터페이스를 사용하는 경우(기본 설정), eth0으로 표시됩니다. Linux 배포가 예측 가능한 네트워크 이름을 지원하는 경우 이는 ens5 등의 이름일 수 있습니다. 자세한 내용을 보려면 인스턴스에서 향상된 네트워킹 기능 활성화에서 RHEL, SUSE 및 CentOS 섹션을 확장하세요.

다음 예제에서는 vif가 드라이버로 표시되어, ena 커널 드라이버가 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 예제에서 ena 커널 드라이버가 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
```

인스턴스에서 향상된 네트워킹 기능 활성화

사용하는 절차는 인스턴스의 운영 체제에 따라 달라집니다.

Amazon Linux

Amazon Linux 2 및 최신 버전의 Amazon Linux AMI에는 ENA가 설치된 향상된 네트워킹에 필요한 커널 드라이버가 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 테스트 섹션을 참조하세요.

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 커널 드라이버로 업데이트합니다.

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

3. 로컬 컴퓨터에서 Amazon EC2 콘솔을 사용하거나 reboot-instances(AWS CLI) 또는 Restart-EC2Instance(AWS Tools for Windows PowerShell) 명령 중 하나를 사용하여 인스턴스를 재부팅합니다.
4. 인스턴스에 다시 연결하고 ena에서 modinfo ena 명령을 사용하여 향상된 네트워킹 기능 활성화 여부 테스트 커널 드라이버가 설치되어 있고 최소 권장 버전 요건을 충족하는지를 확인합니다.
5. [EBS 지원 인스턴스] 로컬 컴퓨터에서 Amazon EC2 콘솔을 사용하거나 stop-instances(AWS CLI) 또는 Stop-EC2Instance(AWS Tools for Windows PowerShell) 명령 중 하나를 사용하여 인스턴스를 중지합니다.

   [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))로 넘어가세요.
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 i-1234567890abcdef0 --ena-support
     ```

   - Edit-EC2InstanceAttribute(Windows PowerShell용 도구)

     ```
     Edit-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -EnaSupport $true
     ```
7. (선택 사항) 의 설명에 따라 인스턴스에서 AMI를 생성합니다Amazon EBS 지원 AMI 생성 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 컴퓨터에서 Amazon EC2 콘솔을 사용하거나 start-instances(AWS CLI) 또는 Start-EC2Instance(AWS Tools for Windows PowerShell) 명령 중 하나를 사용하여 인스턴스를 시작합니다.
9. 인스턴스에 연결하고 ena에서 ethtool -i eth*n* 명령을 사용하여 향상된 네트워킹 기능 활성화 여부 테스트 커널 드라이버가 설치되어 있고 네트워크 인터페이스에 로드되었는지 확인합니다.

   향상된 네트워킹을 활성화한 이후에 인스턴스에 연결할 수 없는 경우 Linux에서 ENA 커널 드라이버 문제 해결 단원을 참조하세요.

Amazon Linux AMI에서 향상된 네트워킹을 활성화하려면(인스턴스 스토어 지원 인스턴스)

이전 절차에서 인스턴스를 중지한 단계까지 진행합니다. 인스턴스 스토어 지원 AMI 생성에 설명된 것처럼 새 AMI를 생성하고, AMI를 등록할 때 향상된 네트워킹 속성을 활성화합니다.

- register-image (AWS CLI)

```
aws ec2 register-image --ena-support ...
```

- Register-EC2Image (AWS Tools for Windows PowerShell)

```
Register-EC2Image -EnaSupport $true ...
```

Ubuntu

최신 Ubuntu HVM AMI에는 ENA가 설치된 향상된 네트워킹에 필요한 커널 드라이버가 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 테스트 섹션을 참조하세요.

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 패키지를 설치하려면(Ubuntu 16.04 이상)

Ubuntu 16.04와 18.04는 Ubuntu 사용자 지정 커널(`linux-aws` 커널 패키지)과 함께 제공됩니다. 다른 커널을 사용하려면 지원에 문의하세요.

`linux-aws` 커널 패키지를 설치하려면(Ubuntu Trusty 14.04)

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 콘솔을 사용하거나 stop-instances(AWS CLI) 또는 Stop-EC2Instance(AWS Tools for Windows PowerShell) 명령 중 하나를 사용하여 인스턴스를 중지합니다.

[인스턴스 스토어 지원 인스턴스] 속성을 수정하기 위해 인스턴스를 중지할 수 없습니다. 그 대신 이 절차(Ubuntu에서 확장 네트워킹 기능을 사용하려면(인스턴스 스토어 지원 인스턴스))로 넘어가세요.

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 i-1234567890abcdef0 --ena-support
   ```

   - Edit-EC2InstanceAttribute(Windows PowerShell용 도구)

   ```
   Edit-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -EnaSupport $true
   ```

5. (선택 사항) 의 설명에 따라 인스턴스에서 AMI를 생성합니다Amazon EBS 지원 AMI 생성 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 컴퓨터에서 Amazon EC2 콘솔을 사용하거나 start-instances(AWS CLI) 또는 Start-EC2Instance(AWS Tools for Windows PowerShell) 명령 중 하나를 사용하여 인스턴스를 시작합니다.

Ubuntu에서 확장 네트워킹 기능을 사용하려면(인스턴스 스토어 지원 인스턴스)

이전 절차에서 인스턴스를 중지한 단계까지 진행합니다. 인스턴스 스토어 지원 AMI 생성에 설명된 것처럼 새 AMI를 생성하고, AMI를 등록할 때 향상된 네트워킹 속성을 활성화합니다.

- register-image (AWS CLI)

```
aws ec2 register-image --ena-support ...
```

- Register-EC2Image (AWS Tools for Windows PowerShell)

```
Register-EC2Image -EnaSupport $true ...
```

RHEL, SUSE, CentOS

Red Hat Enterprise Linux, SUSE Linux Enterprise Server 및 CentOS용 최신 AMI에는 ENA가 포함된 향상된 네트워킹에 필요한 커널 드라이버가 포함되어 있으며 ENA 지원이 활성화되어 있습니다. 따라서 지원되는 인스턴스 유형에 최신 AMI를 사용하여 인스턴스를 시작하면 향상된 네트워킹이 이미 해

당 인스턴스에서 활성화된 상태입니다. 자세한 내용은 향상된 네트워킹 기능 활성화 여부 테스트 섹션을 참조하세요.

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 참조하세요.

Linux에서 향상된 네트워킹을 활성화하려면

1. 인스턴스에 연결합니다.
2. https://github.com/amzn/amzn-drivers의 GitHub에서 인스턴스의 ena 커널 드라이버에 대한 소스 코드를 복제합니다. (SUSE Linux Enterprise Server 12 SP2 이상에는 기본적으로 ENA 2.02가 포함되므로 ENA 드라이버를 다운로드하고 컴파일할 필요가 없습니다. SUSE Linux Enterprise Server 12 SP2 이상의 경우 원하는 드라이버 버전을 스톡 커널에 추가하기 위한 요청을 제출해야 합니다.)

   ```
   git clone https://github.com/amzn/amzn-drivers
   ```

3. 인스턴스에 ena 커널 드라이버를 컴파일하고 설치합니다. 이러한 단계는 Linux 배포판에 따라 달라집니다. Red Hat Enterprise Linux에서 커널 드라이버를 컴파일하는 방법에 대한 자세한 내용은 RHEL을 실행하는 Amazon EC2 Instance에서 강화된 네트워크 지원을 받기 위해 최신 ENS 드라이버를 설치하려면 어떻게 해야 합니까?를 참조하세요.
4. 커널 드라이버 종속성을 업데이트하려면 sudo depmod 명령을 실행합니다.
5. 인스턴스에서 `initramfs`를 업데이트하여 부팅 시 새 커널 드라이버가 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 참조하세요.

   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 콘솔을 사용하거나 stop-instances(AWS CLI) 또는 Stop-EC2Instance(AWS Tools for Windows PowerShell) 명령 중 하나를 사용하여 인스턴스를 중지합니다.

   [인스턴스 스토어 지원 인스턴스] 속성을 수정하기 위해 인스턴스를 중지할 수 없습니다. 그 대신 이 절차(Linux에서 향상된 네트워킹을 사용하려면(인스턴스 스토어 지원 인스턴스))로 넘어가세요.

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 i-1234567890abcdef0 --ena-support
   ```

   - Edit-EC2InstanceAttribute(Windows PowerShell용 도구)

   ```
   Edit-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -EnaSupport $true
   ```

9. (선택 사항) 의 설명에 따라 인스턴스에서 AMI를 생성합니다Amazon EBS 지원 AMI 생성 생성된 AMI는 인스턴스의 향상된 네트워킹 `enaSupport` 속성을 상속합니다. 따라서 이 AMI를 사용하여 기본적으로 향상된 네트워킹 기능이 활성화된 상태로 다른 인스턴스를 시작할 수 있습니다.

   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, eth0이 실패하여 부팅 문제가 발생할 수 있습니다. 이 파일은 다음 부팅 주기에 생성되고 AMI에서 시작된 모든 인스턴스가 자체 버전의 파일을 생성합니다.

10. 로컬 컴퓨터에서 Amazon EC2 콘솔을 사용하거나 start-instances(AWS CLI) 또는 Start-EC2Instance(AWS Tools for Windows PowerShell) 명령 중 하나를 사용하여 인스턴스를 시작합니다.
11. (선택 사항) 인스턴스에 연결하여 커널 드라이버의 설치 여부를 확인합니다.

    향상된 네트워킹을 활성화한 이후에 인스턴스에 연결할 수 없는 경우 Linux에서 ENA 커널 드라이버 문제 해결 단원을 참조하세요.

Linux에서 향상된 네트워킹을 사용하려면(인스턴스 스토어 지원 인스턴스)

이전 절차에서 인스턴스를 중지한 단계까지 진행합니다. 인스턴스 스토어 지원 AMI 생성에 설명된 것처럼 새 AMI를 생성하고, AMI를 등록할 때 향상된 네트워킹 속성을 활성화합니다.

- register-image (AWS CLI)

```
aws ec2 register-image --ena-support ...
```

- Register-EC2Image (AWS Tools for Windows PowerShell)

```
Register-EC2Image -EnaSupport ...
```

DKMS가 포함된 Ubuntu

이 방법은 테스트 및 피드백 목적으로만 사용됩니다. 프로덕션 배포에 사용하기 위한 것이 아닙니다. 프로덕션 배포는 Ubuntu를 참조하세요.

> **Important**
>
> DKMS를 사용하면 구독에 대한 지원 계약이 무효화됩니다. 프로덕션 배포에는 사용할 수 없습니다.

Ubuntu에서 ENA를 사용하여 향상된 네트워킹 기능을 활성화하려면(EBS 지원 인스턴스)

1. Ubuntu의 1과 2단계를 따르세요.

2. 커널 드라이버를 컴파일하도록 build-essential 패키지를 설치하고 커널을 업데이트할 때마다 ena 커널 드라이버가 다시 빌드되도록 dkms 패키지를 설치합니다.

```
ubuntu:~$ sudo apt-get install -y build-essential dkms
```

3. https://github.com/amzn/amzn-drivers의 GitHub에서 인스턴스의 ena 커널 드라이버에 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 커널 드라이버를 추가 및 빌드하고 설치합니다.

   커널 드라이버를 DKMS에 추가합니다.

```
ubuntu:~$ sudo dkms add -m amzn-drivers -v 1.0.0
```

dkms 명령을 사용하여 커널 드라이버를 빌드합니다.

```
ubuntu:~$ sudo dkms build -m amzn-drivers -v 1.0.0
```

dkms를 사용하여 커널 드라이버를 설치합니다.

```
ubuntu:~$ sudo dkms install -m amzn-drivers -v 1.0.0
```

7. 부팅 시 올바른 커널 드라이버가 로드되도록 `initramfs`를 다시 빌드합니다.

```
ubuntu:~$ sudo update-initramfs -u -k all
```

8. ena의 modinfo ena 명령을 사용하여 향상된 네트워킹 기능 활성화 여부 테스트 커널 드라이버가 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. Ubuntu의 3단계를 계속하세요.

Windows에서 향상된 네트워킹 활성화

확장 네트워크를 설정하지 않은 상태로 인스턴스를 시작한 경우에는 인스턴스에 필요한 네트워크 어댑터 드라이버를 다운로드하여 설치한 다음 `enaSupport` 인스턴스 속성을 설정하여 확장 네트워크를 활성화해야 합니다.

향상된 네트워킹을 활성화하려면

1. 인스턴스 연결 후 로컬 관리자로 로그인합니다.
2. [Windows Server 2016 및 2019만 해당] 다음 EC2Launch PowerShell 스크립트를 실행하여 드라이버가 설치된 후의 인스턴스를 구성합니다.

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -
Schedule
```

3. 다음과 같이 인스턴스 상에서 드라이버를 설치합니다.
   a. 최신 드라이버를 인스턴스로 다운로드합니다.
   b. ZIP 아카이브를 추출합니다.
   c. `install.ps1` PowerShell 스크립트를 실행하여 드라이버를 설치합니다.

> **Note**
>
> 실행 정책 오류가 발생하면 정책을 `Unrestricted`(기본값으로 `Restricted` 또는 `RemoteSigned`로 설정되어 있음)로 설정합니다. 명령 줄에서 `Set-ExecutionPolicy -ExecutionPolicy Unrestricted`를 실행한 다음 `install.ps1` PowerShell 스크립트를 다시 실행하세요.

4. 로컬 컴퓨터에서 Amazon EC2 콘솔을 사용하거나 stop-instances(AWS CLI) 또는 Stop-EC2Instance(AWS Tools for Windows PowerShell) 명령 중 하나를 사용하여 인스턴스를 중지합니다.
5. 다음과 같이 인스턴스에서 ENA 지원을 활성화합니다.

a. 로컬 컴퓨터에서 다음 명령 중 하나를 실행하여 해당 인스턴스의 EC2 인스턴스 ENA 지원 속성을 확인합니다. 이 속성이 활성 상태가 아니면 "[]" 또는 공백이 출력됩니다. 기본적으로 `EnaSupport`는 `false`로 설정됩니다.

- describe-instances (AWS CLI)

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

- Get-EC2Instance(Windows PowerShell용 도구)

```
(Get-EC2Instance -InstanceId i-1234567890abcdef0).Instances.EnaSupport
```

b. ENA 지원을 활성화하려면 다음 명령 중 하나를 실행합니다.

- modify-instance-attribute (AWS CLI)

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --ena-
support
```

- Edit-EC2InstanceAttribute (AWS Tools for Windows PowerShell)

```
Edit-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -EnaSupport $true
```

인스턴스를 재시작할 때 문제가 발생하는 경우 다음 명령 중 하나를 사용하여 ENA 지원을 비활성화할 수도 있습니다.

- modify-instance-attribute (AWS CLI)

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --no-ena-
support
```

- Edit-EC2InstanceAttribute (AWS Tools for Windows PowerShell)

```
Edit-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -EnaSupport $false
```

c. 이전 설명과 같이 `true` 또는 describe-instances를 사용하여 속성이 Get-EC2Instance로 설정되어 있는지 확인합니다. 이제 다음 결과가 표시됩니다.

```
[
 true
]
```

6. 로컬 컴퓨터에서 Amazon EC2 콘솔을 사용하거나 start-instances(AWS CLI) 또는 Start-EC2Instance(AWS Tools for Windows PowerShell) 명령 중 하나를 사용하여 인스턴스를 시작합니다.
7. 인스턴스에서 다음과 같이 ENA 드라이버가 설치되고 활성화되어 있는지 확인합니다.
   a. 네트워크 아이콘을 마우스 오른쪽 버튼으로 클릭하고 네트워크 및 공유 센터 열기(Open Network and Sharing Center)를 선택합니다.
   b. 이더넷 어댑터(예: Ethernet 2)를 선택합니다.
   c. 세부 정보를 선택합니다. 네트워크 연결 세부 정보(Network Connection Details)에서 설명(Description)이 Amazon Elastic Network Adapter인지 확인합니다.
8. (선택 사항) 인스턴스에서 AMI를 만듭니다. 생성된 AMI는 인스턴스의 `enaSupport` 속성을 상속합니다. 따라서 이 AMI를 사용하여 기본적으로 ENA가 활성화된 상태로 다른 인스턴스를 시작할 수 있습니다.

ENA Express로 EC2 인스턴스 간 네트워크 성능 향상

ENA Express는 AWS Scalable Reliable Datagram(SRD) 기술로 구동됩니다. SRD는 동적 라우팅을 사용하여 스루풋을 늘리고 테일 지연 시간을 최소화하는 고성능 네트워크 전송 프로토콜입니다. ENA Express를 사용하면 동일한 가용 영역에 있는 두 EC2 인스턴스 간에 통신할 수 있습니다.

ENA Express의 이점

- 가용 영역 내에서 단일 흐름에 사용할 수 있는 최대 대역폭을 5Gbps에서 25Gbps로 집계 인스턴스 제한까지 늘립니다.
- 특히 네트워크 부하가 높은 기간 동안, EC2 인스턴스 간 네트워크 트래픽의 테일 지연 시간을 줄입니다.
- 혼잡한 네트워크 경로를 감지하고 피합니다.
- 수신 측의 패킷 재정렬 작업이나 필요한 대부분의 재전송 작업과 같은 일부 작업을 네트워크 계층에서 직접 처리합니다. 따라서 애플리케이션 계층을 다른 작업에 사용할 수 있습니다.

> **Note**
>
> - 애플리케이션이 초당 많은 양의 패킷을 송수신하고 대부분의 시간, 특히 네트워크에 정체가 없는 기간 동안 지연 시간을 최적화해야 하는 경우 향상된 네트워킹이 네트워크에 더 적합할 수 있습니다.
> - ENA Express 트래픽은 로컬 영역으로 전송할 수 없습니다.

인스턴스에서 네트워크 인터페이스 연결에 대해 ENA Express를 활성화하면, 전송 인스턴스가 수신 인스턴스와의 통신을 시작하고 SRD는 ENA Express가 전송 인스턴스와 수신 인스턴스 모두에서 작동하는지 감지합니다. ENA Express가 작동 중인 경우 통신에 SRD 전송이 사용될 수 있습니다. ENA Express가 작동하지 않는 경우 통신이 표준 ENA 전송으로 폴백됩니다.

네트워크 트래픽이 적은 시간 동안 패킷이 ENA Express를 사용하면 패킷 지연 시간(수십 마이크로초)이 약간 증가할 수 있습니다. 이 시간 동안 특정 네트워크 성능 특성을 우선시하는 애플리케이션은 ENA Express를 통해 다음과 같은 이점을 얻을 수 있습니다.

- 프로세스는 동일한 가용 영역 내에서 5Gbps에서 25Gbps(집계 인스턴스 제한)로 늘어난 최대 단일 흐름 대역폭의 이점을 누릴 수 있습니다. 예를 들어, 특정 인스턴스 유형이 최대 12.5Gbps를 지원하는 경우 단일 흐름 대역폭도 12.5Gbps로 제한됩니다.
- 더 오래 실행되는 프로세스의 경우 네트워크 정체 기간 동안 테일 지연 시간이 감소해야 합니다.
- 네트워크 응답 시간에 대해 보다 원활하고 표준적인 배포에 따른 이점이 있습니다.

주제

- ENA Express의 작동 방식
- ENA Express를 지원하는 인스턴스 유형
- Linux Windows 인스턴스의 사전 조건
- Linux 인스턴스의 ENA Express 설정에 대한 성능 튜닝
- EC2 인스턴스의 ENA Express 설정 검토
- EC2 인스턴스의 ENA Express 설정 구성

ENA Express의 작동 방식

ENA Express는 AWS Scalable Reliable Datagram(SRD) 기술로 구동됩니다. 각 네트워크 흐름의 패킷을 서로 다른 AWS 네트워크 경로에 분산하고, 혼잡 징후가 감지되면 분포를 동적으로 조정합니다. 또한 수신 측에서 패킷 재정렬을 관리합니다.

ENA Express가 의도한 대로 네트워크 트래픽을 관리하려면, 전송 및 수신 인스턴스와 해당 인스턴스 간 통신이 다음 요구 사항을 모두 충족해야 합니다.

- 전송 인스턴스 유형과 수신 인스턴스 유형이 모두 지원되어야 합니다. 자세한 내용은 ENA Express를 지원하는 인스턴스 유형 표를 참조하세요.
- 전송 인스턴스와 수신 인스턴스 모두에 ENA Express가 구성되어 있어야 합니다. 구성에 차이가 있는 경우, 트래픽이 표준 ENA 전송으로 기본 설정되는 상황이 발생할 수 있습니다. 다음 시나리오는 이 경우에 어떤 일이 발생하는지 보여줍니다.

  시나리오: 구성의 차이

Instance	ENA Express가 활성화됨	UDP가 ENA Express 사용
인스턴스 1	예	예
인스턴스 2	예	아니요

  이 경우 두 인스턴스 모두에 ENA Express가 활성화되어 있으므로 두 인스턴스 간의 TCP 트래픽에 ENA Express를 사용할 수 있습니다. 하지만 인스턴스 중 하나는 UDP 트래픽에 ENA Express를 사용하지 않으므로, UDP를 통한 이 두 인스턴스 간의 통신에는 표준 ENA 전송이 사용됩니다.
- 전송 인스턴스와 수신 인스턴스가 동일한 가용 영역에서 실행되어야 합니다.
- 인스턴스 간 네트워크 경로에 미들웨어 박스가 포함되지 않아야 합니다. ENA Express는 현재 미들웨어 박스를 지원하지 않습니다.
- (Linux 인스턴스만 해당) 대역폭의 잠재력을 최대한 활용하려면 드라이버 버전 2.2.9 이상을 사용하세요.
- (Linux 인스턴스만 해당) 지표를 생성하려면 드라이버 버전 2.8 이상을 사용합니다.

요구 사항이 하나라도 충족되지 않을 경우, 인스턴스는 표준 TCP/UDP 프로토콜을 사용하지만 SRD 없이 통신합니다.

인스턴스 네트워크 드라이버가 최적의 성능을 발휘하도록 구성되었는지 확인하려면, ENA 드라이버에 대한 권장 모범 사례를 검토하세요. 이러한 모범 사례는 ENA Express에도 적용됩니다. 자세한 내용은 GitHub 웹 사이트에서 ENA Linux Driver Best Practices and Performance Optimization Guide(ENA Linux 드라이버 모범 사례 및 성능 최적화 가이드)를 참조하세요.

> **Note**
>
> Amazon EC2는 인스턴스와 연결(attachment)로서 해당 인스턴스에 연결된 네트워크 인터페이스 간의 관계를 참조합니다. ENA Express 설정이 이 연결에 적용됩니다. 네트워크 인터페이스가 인스턴스에서 분리되면, 이 연결은 더 이상 존재하지 않으며 해당 연결에 적용된 ENA Express 설정도 더 이상 적용되지 않습니다. 네트워크 인터페이스가 남아 있을 수 있지만, 인스턴스가 종료될 때도 마찬가지입니다.

전송 인스턴스와 수신 인스턴스 모두에서 네트워크 인터페이스 연결에 대해 ENA Express를 활성화한 후에는 ENA Express 지표를 사용하여 인스턴스에서 SRD 기술이 제공하는 성능 개선의 이점을 최대한 활용할 수 있습니다. ENA Express 지표에 대한 자세한 내용은 ENA Express의 지표 섹션을 참조하세요.

ENA Express를 지원하는 인스턴스 유형

다음 인스턴스 유형은 ENA Express를 지원합니다.

General purpose

인스턴스 유형	아키텍처
m6a.12xlarge	x86_64
m6a.16xlarge	x86_64
m6a.24xlarge	x86_64
m6a.32xlarge	x86_64
m6a.48xlarge	x86_64
m6a.metal	x86_64
m6i.8xlarge	x86_64

인스턴스 유형	아키텍처
m6i.12xlarge	x86_64
m6i.16xlarge	x86_64
m6i.24xlarge	x86_64
m6i.32xlarge	x86_64
m6i.metal	x86_64
m6id.8xlarge	x86_64
m6id.12xlarge	x86_64
m6id.16xlarge	x86_64
m6id.24xlarge	x86_64
m6id.32xlarge	x86_64
m6id.metal	x86_64
m6idn.8xlarge	x86_64
m6idn.12xlarge	x86_64
m6idn.16xlarge	x86_64
m6idn.24xlarge	x86_64
m6idn.32xlarge	x86_64
m6idn.metal	x86_64
m6in.8xlarge	x86_64
m6in.12xlarge	x86_64
m6in.16xlarge	x86_64

인스턴스 유형	아키텍처
m6in.24xlarge	x86_64
m6in.32xlarge	x86_64
m6in.metal	x86_64
m7a.12xlarge	x86_64
m7a.16xlarge	x86_64
m7a.24xlarge	x86_64
m7a.32xlarge	x86_64
m7a.48xlarge	x86_64
m7a.metal-48xl	x86_64
m7g.12xlarge	arm64
m7g.16xlarge	arm64
m7g.metal	arm64
m7gd.12xlarge	arm64
m7gd.16xlarge	arm64
m7gd.metal	arm64
m7i.12xlarge	x86_64
m7i.16xlarge	x86_64
m7i.24xlarge	x86_64
m7i.48xlarge	x86_64
m7i.metal-24xl	x86_64

인스턴스 유형	아키텍처
m7i.metal-48xl	x86_64
m8g.12xlarge	arm64
m8g.16xlarge	arm64
m8g.24xlarge	arm64
m8g.48xlarge	arm64
m8g.metal-24xl	arm64
m8g.metal-48xl	arm64
m8gd.12xlarge	arm64
m8gd.16xlarge	arm64
m8gd.24xlarge	arm64
m8gd.48xlarge	arm64
m8gd.metal-24xl	arm64
m8gd.metal-48xl	arm64

Compute optimized

인스턴스 유형	아키텍처
c6a.12xlarge	x86_64
c6a.16xlarge	x86_64
c6a.24xlarge	x86_64
c6a.32xlarge	x86_64

인스턴스 유형	아키텍처
c6a.48xlarge	x86_64
c6a.metal	x86_64
c6gn.4xlarge	arm64
c6gn.8xlarge	arm64
c6gn.12xlarge	arm64
c6gn.16xlarge	arm64
c6i.8xlarge	x86_64
c6i.12xlarge	x86_64
c6i.16xlarge	x86_64
c6i.24xlarge	x86_64
c6i.32xlarge	x86_64
c6i.metal	x86_64
c6id.8xlarge	x86_64
c6id.12xlarge	x86_64
c6id.16xlarge	x86_64
c6id.24xlarge	x86_64
c6id.32xlarge	x86_64
c6id.metal	x86_64
c6in.8xlarge	x86_64
c6in.12xlarge	x86_64

인스턴스 유형	아키텍처
c6in.16xlarge	x86_64
c6in.24xlarge	x86_64
c6in.32xlarge	x86_64
c6in.metal	x86_64
c7a.12xlarge	x86_64
c7a.16xlarge	x86_64
c7a.24xlarge	x86_64
c7a.32xlarge	x86_64
c7a.48xlarge	x86_64
c7a.metal-48xl	x86_64
c7g.12xlarge	arm64
c7g.16xlarge	arm64
c7g.metal	arm64
c7gd.12xlarge	arm64
c7gd.16xlarge	arm64
c7gd.metal	arm64
c7gn.4xlarge	arm64
c7gn.8xlarge	arm64
c7gn.12xlarge	arm64
c7gn.16xlarge	arm64

인스턴스 유형	아키텍처
c7gn.metal	arm64
c7i.12xlarge	x86_64
c7i.16xlarge	x86_64
c7i.24xlarge	x86_64
c7i.48xlarge	x86_64
c7i.metal-24xl	x86_64
c7i.metal-48xl	x86_64
c8g.12xlarge	arm64
c8g.16xlarge	arm64
c8g.24xlarge	arm64
c8g.48xlarge	arm64
c8g.metal-24xl	arm64
c8g.metal-48xl	arm64
c8gd.12xlarge	arm64
c8gd.16xlarge	arm64
c8gd.24xlarge	arm64
c8gd.48xlarge	arm64
c8gd.metal-24xl	arm64
c8gd.metal-48xl	arm64

Memory optimized

인스턴스 유형	아키텍처
r6a.12xlarge	x86_64
r6a.16xlarge	x86_64
r6a.24xlarge	x86_64
r6a.32xlarge	x86_64
r6a.48xlarge	x86_64
r6a.metal	x86_64
r6i.8xlarge	x86_64
r6i.12xlarge	x86_64
r6i.16xlarge	x86_64
r6i.24xlarge	x86_64
r6i.32xlarge	x86_64
r6i.metal	x86_64
r6idn.8xlarge	x86_64
r6idn.12xlarge	x86_64
r6idn.16xlarge	x86_64
r6idn.24xlarge	x86_64
r6idn.32xlarge	x86_64
r6idn.metal	x86_64
r6in.8xlarge	x86_64

인스턴스 유형	아키텍처
r6in.12xlarge	x86_64
r6in.16xlarge	x86_64
r6in.24xlarge	x86_64
r6in.32xlarge	x86_64
r6in.metal	x86_64
r6id.8xlarge	x86_64
r6id.12xlarge	x86_64
r6id.16xlarge	x86_64
r6id.24xlarge	x86_64
r6id.32xlarge	x86_64
r6id.metal	x86_64
r7a.12xlarge	x86_64
r7a.16xlarge	x86_64
r7a.24xlarge	x86_64
r7a.32xlarge	x86_64
r7a.48xlarge	x86_64
r7a.metal-48xl	x86_64
r7g.12xlarge	arm64
r7g.16xlarge	arm64
r7g.metal	arm64

인스턴스 유형	아키텍처
r7gd.12xlarge	arm64
r7gd.16xlarge	arm64
r7gd.metal	arm64
r7i.12xlarge	x86_64
r7i.16xlarge	x86_64
r7i.24xlarge	x86_64
r7i.48xlarge	x86_64
r7i.metal-24xl	x86_64
r7i.metal-48xl	x86_64
r8g.12xlarge	arm64
r8g.16xlarge	arm64
r8g.24xlarge	arm64
r8g.48xlarge	arm64
r8g.metal-24xl	arm64
r8g.metal-48xl	arm64
r8gd.12xlarge	arm64
r8gd.16xlarge	arm64
r8gd.24xlarge	arm64
r8gd.48xlarge	arm64
r8gd.metal-24xl	arm64

인스턴스 유형	아키텍처
r8gd.metal-48xl	arm64
u7i-6tb.112xlarge	x86_64
u7i-8tb.112xlarge	x86_64
u7i-12tb.224xlarge	x86_64
u7in-16tb.224xlarge	x86_64
u7in-24tb.224xlarge	x86_64
u7in-32tb.224xlarge	x86_64
u7inh-32tb.480xlarge	x86_64
x2idn.16xlarge	x86_64
x2idn.24xlarge	x86_64
x2idn.32xlarge	x86_64
x2idn.metal	x86_64
x2iedn.8xlarge	x86_64
x2iedn.16xlarge	x86_64
x2iedn.24xlarge	x86_64
x2iedn.32xlarge	x86_64
x2iedn.metal	x86_64
x8g.12xlarge	arm64
x8g.16xlarge	arm64
x8g.24xlarge	arm64

인스턴스 유형	아키텍처
x8g.48xlarge	arm64
x8g.metal-24xl	arm64
x8g.metal-48xl	arm64

Accelerated computing

인스턴스 유형	아키텍처
g6.48xlarge	x86_64
g6e.12xlarge	x86_64
g6e.24xlarge	x86_64
g6e.48xlarge	x86_64
p5.48xlarge	x86_64
p5e.48xlarge	x86_64
p5en.48xlarge	x86_64
p6-b200.48xlarge	x86_64

Storage optimized

인스턴스 유형	아키텍처
i4g.4xlarge	arm64
i4g.8xlarge	arm64
i4g.16xlarge	arm64
i4i.8xlarge	x86_64

인스턴스 유형	아키텍처
i4i.12xlarge	x86_64
i4i.16xlarge	x86_64
i4i.24xlarge	x86_64
i4i.32xlarge	x86_64
i4i.metal	x86_64
i7i.12xlarge	x86_64
i7i.16xlarge	x86_64
i7i.24xlarge	x86_64
i7i.48xlarge	x86_64
i7i.metal-24xl	x86_64
i7i.metal-48xl	x86_64
i7ie.12xlarge	x86_64
i7ie.18xlarge	x86_64
i7ie.24xlarge	x86_64
i7ie.48xlarge	x86_64
i7ie.metal-24xl	x86_64
i7ie.metal-48xl	x86_64
i8g.12xlarge	arm64
i8g.16xlarge	arm64
i8g.24xlarge	arm64

인스턴스 유형	아키텍처
i8g.48xlarge	arm64
i8g.metal-24xl	arm64
im4gn.4xlarge	arm64
im4gn.8xlarge	arm64
im4gn.16xlarge	arm64

Linux Windows 인스턴스의 사전 조건

ENA Express가 효과적으로 작동할 수 있도록 하려면 다음과 같이 Linux 인스턴스 설정을 업데이트하세요.

- 인스턴스에서 점보 프레임을 사용하는 경우, 다음 명령을 실행하여 최대 전송 단위(MTU)를 8900으로 설정합니다.

```
[ec2-user ~]$ sudo ip link set dev eth0 mtu 8900
```

- 다음과 같이 수신기(Rx) 링 크기를 늘립니다.

```
[ec2-user ~]$ ethtool -G device rx 8192
```

- ENA Express 대역폭을 최대화하려면 다음과 같이 TCP 대기열 제한을 구성합니다.

  1. TCP 작은 대기열 제한을 1MB 이상으로 설정합니다. 이렇게 하면 소켓에서 전송을 위해 대기열에 등록되는 데이터 양이 증가합니다.

     ```
     sudo sh -c 'echo 1048576 > /proc/sys/net/ipv4/tcp_limit_output_bytes'
     ```

  2. eth 디바이스에서 바이트 대기열 제한이 Linux 배포에 대해 활성화되어 있는 경우 비활성화합니다. 이렇게 하면 장치 대기열에 전송을 위해 등록되는 데이터가 증가합니다.

     ```
     sudo sh -c 'for txq in /sys/class/net/eth0/queues/tx-*; do echo max > ${txq}/
     byte_queue_limits/limit_min; done'
     ```

> **Note**
>
> Amazon Linux 배포용 ENA 드라이버는 기본적으로 바이트 대기열 제한을 비활성화합니다.

Linux 인스턴스의 ENA Express 설정에 대한 성능 튜닝

최적의 ENA Express 성능을 위해 Linux 인스턴스 구성을 확인하려면 Amazon GitHub 리포지토리에서 사용 가능한 다음 스크립트를 실행할 수 있습니다.

https://github.com/amzn/amzn-ec2-ena-utilities/blob/main/ena-express/check-ena-express-settings.sh

이 스크립트는 일련의 테스트를 실행하고 권장 및 필수 구성 변경을 모두 제안합니다.

EC2 인스턴스의 ENA Express 설정 검토

인스턴스 또는 네트워크 인터페이스별로 ENA Express 설정을 확인할 수 있습니다. ENA Express 설정을 업데이트하려면 the section called “인스턴스 설정 구성” 섹션을 참조하세요.

Console

네트워크 인터페이스의 ENA Express 설정을 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network interfaces(네트워크 인터페이스)를 선택합니다.
3. 해당 인스턴스의 세부 정보를 보려면 네트워크 인터페이스를 선택합니다. Network interface ID(네트워크 인터페이스 ID) 링크를 선택하여 세부 정보 페이지를 열거나, 목록 왼쪽의 확인란을 선택하여 페이지 하단의 세부 정보 창에서 세부 정보를 볼 수 있습니다.
4. Details(세부 정보) 탭 또는 세부 정보 페이지의 Network interface attachment(네트워크 인터페이스 연결) 섹션에서 ENA Express 및 ENA Express UDP의 설정을 검토합니다.

인스턴스의 ENA Express 설정을 보려면

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

3. 해당 인스턴스의 세부 정보를 보려면 인스턴스를 선택합니다. Instance ID(인스턴스 ID) 링크를 선택하여 세부 정보 페이지를 열거나, 목록 왼쪽의 확인란을 선택하여 페이지 하단의 세부 정보 창에서 세부 정보를 볼 수 있습니다.
4. Networking(네트워킹) 탭의 Network interfaces(네트워크 인터페이스) 섹션에서 오른쪽으로 스크롤하여 ENA Express 및 ENA Express UDP의 설정을 검토합니다.

AWS CLI

인스턴스의 ENA Express 설정을 가져오려면

describe-instances 명령을 사용합니다. 이 명령 예제는 `--instance-ids` 파라미터로 지정된 실행 중인 각 인스턴스에 연결된 네트워크 인터페이스의 ENA Express 구성 목록을 반환합니다.

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 i-0598c7d356eba48d7 \
    --query 'Reservations[*].Instances[*].[InstanceId,
 NetworkInterfaces[*].Attachment.EnaSrdSpecification]'
```

출력의 예시는 다음과 같습니다.

```
[
    [
        [
            "i-1234567890abcdef0",
            [
                {
                    "EnaSrdEnabled": true,
                    "EnaSrdUdpSpecification": {
                        "EnaSrdUdpEnabled": false
                    }
                }
            ]
        ]
    ],
    [
        [
            "i-0598c7d356eba48d7",
            [
            {
                    "EnaSrdEnabled": true,
                    "EnaSrdUdpSpecification": {
```

```
                        "EnaSrdUdpEnabled": false
                    }
                }
            ]
        ]
    ]
```

네트워크 인터페이스의 ENA Express 설정을 가져오려면

describe-network-interfaces 명령을 사용합니다.

```
aws ec2 describe-network-interfaces \
    --network-interface-ids eni-1234567890abcdef0 \
    --query NetworkInterfaces[].[NetworkInterfaceId,Attachment.EnaSrdSpecification]
```

출력의 예시는 다음과 같습니다.

```
[
    [
        "eni-1234567890abcdef0",
        {
            "EnaSrdEnabled": true,
            "EnaSrdUdpSpecification": {
                "EnaSrdUdpEnabled": false
            }
        }
    ]
]
```

PowerShell

네트워크 인터페이스의 ENA Express 설정을 가져오려면

Get-EC2NetworkInterface cmdlet을 사용하세요.

```
Get-EC2NetworkInterface `
    -NetworkInterfaceId eni-1234567890abcdef0 | `
Select-Object `
    Association,
    NetworkInterfaceId,
```

```
    OwnerId,
    @{Name = 'AttachTime'; Expression = { $_.Attachment.AttachTime } },
    @{Name = 'AttachmentId'; Expression = { $_.Attachment.AttachmentId } },
    @{Name = 'DeleteOnTermination'; Expression =
{ $_.Attachment.DeleteOnTermination } },
    @{Name = 'NetworkCardIndex'; Expression = { $_.Attachment.NetworkCardIndex } },
    @{Name = 'InstanceId'; Expression = { $_.Attachment.InstanceId } },
    @{Name = 'InstanceOwnerId'; Expression = { $_.Attachment.InstanceOwnerId } },
    @{Name = 'Status'; Expression = { $_.Attachment.Status } },
    @{Name = 'EnaSrdEnabled'; Expression =
{ $_.Attachment.EnaSrdSpecification.EnaSrdEnabled } },
    @{Name = 'EnaSrdUdpEnabled'; Expression =
{ $_.Attachment.EnaSrdSpecification.EnaSrdUdpSpecification.EnaSrdUdpEnabled } }
```

출력의 예시는 다음과 같습니다.

```
Association         :
NetworkInterfaceId  : eni-0d1234e5f6a78901b
OwnerId             : 111122223333
AttachTime          : 6/11/2022 1:13:11 AM
AttachmentId        : eni-attach-0d1234e5f6a78901b
DeleteOnTermination : True
NetworkCardIndex    : 0
InstanceId          : i-1234567890abcdef0
InstanceOwnerId     : 111122223333
Status              : attached
EnaSrdEnabled       : True
EnaSrdUdpEnabled    : False
```

EC2 인스턴스의 ENA Express 설정 구성

추가 소프트웨어를 설치하지 않고도 지원되는 EC2 인스턴스 유형에 대해 ENA Express를 구성할 수 있습니다. 자세한 내용은 the section called “ENA Express를 지원하는 인스턴스 유형” 섹션을 참조하세요.

Console

네트워크 인터페이스의 ENA Express를 관리하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network interfaces(네트워크 인터페이스)를 선택합니다.

3. 인스턴스에 연결된 네트워크 인터페이스를 선택합니다. Network interface ID(네트워크 인터페이스 ID) 링크를 선택하여 세부 정보 페이지를 열거나, 목록 왼쪽의 확인란을 선택할 수 있습니다.
4. 페이지 오른쪽 상단의 Action(작업) 메뉴에서 Manage ENA Express(ENA Express 관리)를 선택합니다. 그러면 선택한 네트워크 인터페이스 ID와 현재 설정이 표시되어 있는 Manage ENA Express(ENA Express 관리) 대화 상자가 열립니다.

   선택한 네트워크 인터페이스가 인스턴스에 연결되지 않은 경우 이 작업은 메뉴에 나타나지 않습니다.
5. ENA Express를 사용하려면 활성화 확인란을 선택합니다.
6. ENA Express가 활성화되면 UDP 설정을 구성할 수 있습니다. ENA Express UDP를 사용하려면 활성화 확인란을 선택합니다.
7. 설정을 저장하려면 Save(저장)를 선택합니다.

인스턴스의 ENA Express를 관리하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 인스턴스를 선택합니다.
3. 관리할 인스턴스를 선택합니다. Instance ID(인스턴스 ID)를 선택하여 세부 정보 페이지를 열거나, 목록 왼쪽의 확인란을 선택할 수 있습니다.
4. 인스턴스에 대해 구성할 Network interface(네트워크 인터페이스)를 선택합니다.
5. 페이지 오른쪽 상단의 Action(작업) 메뉴에서 Manage ENA Express(ENA Express 관리)를 선택합니다.
6. 인스턴스에 연결된 네트워크 인터페이스에 대해 ENA Express를 구성하려면 Network interface(네트워크 인터페이스) 목록에서 해당 인터페이스를 선택합니다.
7. 선택한 네트워크 인터페이스 연결에 ENA Express를 사용하려면 활성화 확인란을 선택합니다.
8. ENA Express가 활성화되면 UDP 설정을 구성할 수 있습니다. ENA Express UDP를 사용하려면 활성화 확인란을 선택합니다.
9. 설정을 저장하려면 Save(저장)를 선택합니다.

네트워크 인터페이스를 연결할 때 ENA Express를 구성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network interfaces(네트워크 인터페이스)를 선택합니다.

3. 인스턴스에 연결되지 않은 네트워크 인터페이스(Status(상태)가 Available(사용 가능)인 네트워크 인터페이스)를 선택합니다. Network interface ID(네트워크 인터페이스 ID) 링크를 선택하여 세부 정보 페이지를 열거나, 목록 왼쪽의 확인란을 선택할 수 있습니다.
4. 연결할 Instance(인스턴스)를 선택합니다.
5. 인스턴스에 네트워크 인터페이스를 연결한 후 ENA Express를 사용하려면 활성화 확인란을 선택합니다.
6. ENA Express가 활성화되면 UDP 설정을 구성할 수 있습니다. ENA Express UDP를 사용하려면 활성화 확인란을 선택합니다.
7. 네트워크 인터페이스를 인스턴스에 연결하고 ENA Express 설정을 저장하려면 Attach(연결)를 선택합니다.

AWS CLI

네트워크 인터페이스를 연결할 때 ENA Express를 구성하려면

다음 예제와 같이 attach-network-interface 명령을 사용합니다.

예제 1: TCP 트래픽에 ENA Express를 사용하고 UDP 트래픽에는 사용 안 함

이 예제는 `EnaSrdEnabled`를 `true`로 구성하고 `EnaSrdUdpEnabled`를 기본적으로 `false`로 설정되도록 합니다.

```
aws ec2 attach-network-interface \
    --network-interface-id eni-1234567890abcdef0 \
    --instance-id i-1234567890abcdef0 \
    --device-index 1 \
    --ena-srd-specification 'EnaSrdEnabled=true'
```

예 2: TCP 트래픽과 UDP 트래픽 모두에 ENA Express 사용

이 예제는 `EnaSrdEnabled` 및 `EnaSrdUdpEnabled`를 모두 `true`로 구성합니다.

```
aws ec2 attach-network-interface \
    --network-interface-id eni-1234567890abcdef0 \
    --instance-id i-1234567890abcdef0 \
    --device-index 1 \
    --ena-srd-specification
 'EnaSrdEnabled=true,EnaSrdUdpSpecification={EnaSrdUdpEnabled=true}'
```

네트워크 인터페이스 연결의 ENA Express 설정을 업데이트하려면

다음 예제와 같이 modify-network-interface-attribute 명령을 사용합니다.

예제 1: TCP 트래픽에 ENA Express를 사용하고 UDP 트래픽에는 사용 안 함

이 예제는 EnaSrdEnabled를 true로 구성하고, 이전에 설정하지 않은 경우 EnaSrdUdpEnabled가 기본적으로 false로 설정되도록 합니다.

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --ena-srd-specification 'EnaSrdEnabled=true'
```

예 2: TCP 트래픽과 UDP 트래픽 모두에 ENA Express 사용

이 예제는 EnaSrdEnabled 및 EnaSrdUdpEnabled를 모두 true로 구성합니다.

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --ena-srd-specification
 'EnaSrdEnabled=true,EnaSrdUdpSpecification={EnaSrdUdpEnabled=true}'
```

예 3: UDP 트래픽의 ENA Express 사용 중지

이 예제는 EnaSrdUdpEnabled를 false로 구성합니다.

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --ena-srd-specification 'EnaSrdUdpSpecification={EnaSrdUdpEnabled=false}'
```

PowerShell

네트워크 인터페이스를 연결할 때 ENA Express를 구성하려면

다음 예제와 같이 Add-EC2NetworkInterface cmdlet을 사용합니다.

예제 1: TCP 트래픽에 ENA Express를 사용하고 UDP 트래픽에는 사용 안 함

이 예제는 EnaSrdEnabled를 true로 구성하고 EnaSrdUdpEnabled를 기본적으로 false로 설정되도록 합니다.

```
Add-EC2NetworkInterface `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -InstanceId i-1234567890abcdef0 `
```

```
    -DeviceIndex 1 `
    -EnaSrdSpecification_EnaSrdEnabled $true
```

예 2: TCP 트래픽과 UDP 트래픽 모두에 ENA Express 사용

이 예제는 EnaSrdEnabled 및 EnaSrdUdpEnabled를 모두 true로 구성합니다.

```
Add-EC2NetworkInterface `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -InstanceId i-1234567890abcdef0 `
    -DeviceIndex 1 `
    -EnaSrdSpecification_EnaSrdEnabled $true `
    -EnaSrdUdpSpecification_EnaSrdUdpEnabled $true
```

네트워크 인터페이스 연결의 ENA Express 설정을 구성하려면

다음 예제와 같이 Edit-EC2NetworkInterfaceAttribute cmdlet을 사용합니다.

예제 1: TCP 트래픽에 ENA Express를 사용하고 UDP 트래픽에는 사용 안 함

이 예제는 EnaSrdEnabled를 true로 구성하고, 이전에 설정하지 않은 경우 EnaSrdUdpEnabled가 기본적으로 false로 설정되도록 합니다.

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -EnaSrdSpecification_EnaSrdEnabled $true ;
Get-EC2NetworkInterface -NetworkInterfaceId eni-0123f4567890a1b23 | `
Select-Object `
    NetworkInterfaceId,
    @{Name = 'EnaSrdEnabled'; Expression =
 { $_.Attachment.EnaSrdSpecification.EnaSrdEnabled }},
    @{Name = 'EnaSrdUdpEnabled'; Expression =
 { $_.Attachment.EnaSrdSpecification.EnaSrdUdpSpecification.EnaSrdUdpEnabled }} | `
Format-List
```

예 2: TCP 트래픽과 UDP 트래픽 모두에 ENA Express 사용

이 예제는 EnaSrdEnabled 및 EnaSrdUdpEnabled를 모두 true로 구성합니다.

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -EnaSrdSpecification_EnaSrdEnabled $true `
    -EnaSrdSpecification_EnaSrdUdpSpecification_EnaSrdUdpEnabled $true ;
```

```
Get-EC2NetworkInterface -NetworkInterfaceId eni-1234567890abcdef0 | `
Select-Object `
    NetworkInterfaceId,
    @{Name = 'EnaSrdEnabled'; Expression =
 { $_.Attachment.EnaSrdSpecification.EnaSrdEnabled }},
    @{Name = 'EnaSrdUdpEnabled'; Expression =
 { $_.Attachment.EnaSrdSpecification.EnaSrdUdpSpecification.EnaSrdUdpEnabled }} | `
Format-List
```

예 3: UDP 트래픽의 ENA Express 사용 중지

이 예제는 EnaSrdUdpEnabled를 false로 구성합니다.

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-0123f4567890a1b23 `
    -EnaSrdSpecification_EnaSrdUdpSpecification_EnaSrdUdpEnabled $false ;
Get-EC2NetworkInterface -NetworkInterfaceId eni-0123f4567890a1b23 | `
Select-Object `
    NetworkInterfaceId,
    @{Name = 'EnaSrdEnabled'; Expression =
 { $_.Attachment.EnaSrdSpecification.EnaSrdEnabled }},
    @{Name = 'EnaSrdUdpEnabled'; Expression =
 { $_.Attachment.EnaSrdSpecification.EnaSrdUdpSpecification.EnaSrdUdpEnabled }} | `
Format-List
```

시작 시 ENA Express 구성

인스턴스를 시작할 때 다음 방법 중 하나를 사용하여 ENA Express를 구성할 수 있습니다. 지정된 링크는 이러한 방법에 대한 AWS Management Console 지침을 참조합니다.

- 인스턴스 시작 마법사 - 인스턴스 시작 마법사로 시작 시 ENA Express를 구성할 수 있습니다. 자세한 내용은 인스턴스 시작 마법사의 네트워크 설정에서 고급 네트워크 구성을 참조하세요.
- 시작 템플릿 - 시작 템플릿을 사용하는 경우 시작 시 ENA Express를 구성할 수 있습니다. 자세한 내용은 Amazon EC2 시작 템플릿 생성 페이지를 참조하고, 네트워크 설정 섹션을 확장하고 고급 네트워크 구성을 검토하세요.

Intel 82599 VF 인터페이스를 통한 향상된 네트워킹

Xen 기반 인스턴스의 경우 Intel 82599 VF(Virtual Function) 인터페이스로 향상된 네트워킹 기능을 제공합니다. 인터페이스는 Intel ixgbevf 드라이버를 사용합니다.

다음 탭은 인스턴스 운영 체제에 설치된 네트워크 어댑터 드라이버를 확인하는 방법을 보여줍니다.

Linux

Linux 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 이름을 지원하는 경우 이는 `ens5`와 같은 이름일 수 있습니다.

다음 예시에서는 `vif`가 드라이버로 표시되어, `ixgbevf` 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
```

Windows

Windows 네트워크 어댑터

드라이버의 설치 여부를 확인하려면 인스턴스에 연결한 뒤 Device Manager를 실행합니다. 네트워크 어댑터에 `Intel(R) 82599 Virtual Function`이 표시되어야 합니다.

내용

- 인스턴스에 향상된 네트워킹 준비
- 향상된 네트워킹 기능 활성화 여부 테스트
- 인스턴스에서 향상된 네트워킹 기능 활성화
- 연결 문제 해결

인스턴스에 향상된 네트워킹 준비

intel 82599 VF 인터페이스를 사용하여 향상된 네트워킹을 준비하려면 인스턴스를 다음과 같이 설정하세요.

- 인스턴스 유형이 C3, C4, D2, I2, M4(`m4.16xlarge` 제외), R3 중 하나인지 확인합니다.
- 인스턴스가 인터넷에 연결되어 있는지 확인합니다.
- 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.
- Linux 인스턴스 – 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, 향상된 네트워킹을 지원하는 인스턴스를 시작하는 경우 인스턴스에 대해 향상된 네트워킹을 사용하도록 이미 설정되어 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.

- Windows 인스턴스 – 64비트 HVM AMI에서 인스턴스를 시작합니다. Windows Server 2008에서는 향상된 네트워킹을 사용하도록 설정할 수 없습니다. Windows Server 2012 R2와 Windows Server 2016 이상 AMI에서는 확장 네트워킹이 이미 활성화되어 있습니다. Windows Server 2012 R2에는

intel 드라이버 1.0.15.3이 포함되어 있으므로 Pnputil.exe 유틸리티를 사용하여 해당 드라이버를 최신 버전으로 업그레이드하는 것이 좋습니다.

- AWS Management Console에서 AWS CloudShell을(를) 사용하거나 선택한 컴퓨터에 AWS CLI 또는 AWS Tools for Windows PowerShell을(를) 설치하고 구성합니다(로컬 데스크톱/노트북 권장). 자세한 내용은 Amazon EC2 액세스 또는 AWS CloudShell 사용 설명서를 참조하세요. Amazon EC2 콘솔에서는 향상된 네트워킹을 관리할 수 없습니다.

향상된 네트워킹 기능 활성화 여부 테스트

`sriovNetSupport` 속성이 인스턴스 또는 이미지에 설정되어 있는지 확인합니다.

AWS CLI

인스턴스 속성(sriovNetSupport)을 확인하는 방법

다음 describe-instance-attribute 명령을 사용합니다. 속성이 설정되어 있으면 값은 `simple`입니다.

```
aws ec2 describe-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --attribute sriovNetSupport
```

이미지 속성(sriovNetSupport)을 확인하는 방법

다음 describe-images 명령을 사용합니다. 속성이 설정되어 있으면 값은 `simple`입니다.

```
aws ec2 describe-images \
    --image-id ami-0abcdef1234567890 \
    --query "Images[].SriovNetSupport"
```

PowerShell

인스턴스 속성(sriovNetSupport)을 확인하는 방법

Get-EC2InstanceAttribute cmdlet을 사용하세요. 속성이 설정되어 있으면 값은 `simple`입니다.

```
Get-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -Attribute sriovNetSupport
```

이미지 속성(sriovNetSupport)을 확인하는 방법

다음 describe-images 명령을 사용합니다. 속성이 설정되어 있으면 값은 simple입니다.

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).SriovNetSupport
```

인스턴스에서 향상된 네트워킹 기능 활성화

사용하는 절차는 인스턴스의 운영 체제에 따라 달라집니다.

> **Warning**
> 향상된 네트워킹 속성을 활성화한 다음에는 다시 비활성화할 수 없습니다.

Amazon Linux

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 테스트 섹션을 참조하세요.

구형 Amazon Linux AMI를 사용하여 인스턴스를 시작했는데 아직 확장 네트워크 기능이 활성화되어 있지 않다면 다음 절차에 따라 확장 네트워크를 활성화할 수 있습니다.

향상된 네트워킹을 활성화하려면

1. 인스턴스에 연결합니다.
2. 인스턴스 상에서 다음 명령을 사용하여 인스턴스를 ixgbevf를 포함한 최신 커널과 커널 모듈로 업데이트합니다.

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

3. 로컬 컴퓨터에서 Amazon EC2 콘솔을 사용하거나 reboot-instances(AWS CLI) 또는 Restart-EC2Instance(AWS Tools for Windows PowerShell) 명령 중 하나를 사용하여 인스턴스를 재부팅합니다.
4. 인스턴스에 다시 연결하고 ixgbevf에서 modinfo ixgbevf 명령을 사용하여 향상된 네트워킹 기능 활성화 여부 테스트 모듈이 설치되어 있고 최소 권장 버전 요건을 충족하는지를 확인합니다.

5. [EBS 지원 인스턴스] 로컬 컴퓨터에서 Amazon EC2 콘솔을 사용하거나 stop-instances(AWS CLI) 또는 Stop-EC2Instance(AWS Tools for Windows PowerShell) 명령 중 하나를 사용하여 인스턴스를 중지합니다.

   [인스턴스 스토어 지원 인스턴스] 속성을 수정하기 위해 인스턴스를 중지할 수 없습니다. 대신 다음 절차로 진행합니다.

6. 사용자의 로컬 컴퓨터에서 다음 명령 중 하나를 사용하여 확장 네트워크 속성을 활성화합니다.

   AWS CLI

   다음과 같이 modify-instance-attribute 명령을 실행합니다.

   ```
   aws ec2 modify-instance-attribute \
       --instance-id i-1234567890abcdef0 \
       --sriov-net-support simple
   ```

   PowerShell

   다음과 같이 Edit-EC2InstanceAttribute cmdlet을 사용합니다.

   ```
   Edit-EC2InstanceAttribute `
       -InstanceId i-1234567890abcdef0 `
       -SriovNetSupport "simple"
   ```

7. (선택 사항) 의 설명에 따라 인스턴스에서 AMI를 생성합니다Amazon EBS 지원 AMI 생성 생성된 AMI는 인스턴스의 확장 네트워크 속성을 상속합니다. 따라서 이 AMI를 사용하여 기본적으로 향상된 네트워킹 기능이 활성화된 상태로 다른 인스턴스를 시작할 수 있습니다.

8. 로컬 컴퓨터에서 Amazon EC2 콘솔을 사용하거나 start-instances(AWS CLI) 또는 Start-EC2Instance(AWS Tools for Windows PowerShell) 명령 중 하나를 사용하여 인스턴스를 시작합니다.

9. 인스턴스에 연결하고 `ixgbevf`에서 ethtool -i eth*n* 명령을 사용하여 향상된 네트워킹 기능 활성화 여부 테스트 모듈이 설치되어 있고 네트워크 인터페이스에 로드되었는지 확인합니다.

향상된 네트워킹 기능을 활성화하려면(인스턴스 스토어 지원 인스턴스)

이전 절차에서 인스턴스를 중지한 단계까지 진행합니다. 인스턴스 스토어 지원 AMI 생성에 설명된 것처럼 새 AMI를 생성하고, AMI를 등록할 때 향상된 네트워킹 속성을 활성화합니다.

AWS CLI

다음과 같이 register-image 명령을 실행합니다.

```
aws ec2 register-image --sriov-net-support simple ...
```

PowerShell

다음과 같이 Register-EC2Image를 사용합니다.

```
Register-EC2Image -SriovNetSupport "simple" ...
```

Ubuntu

시작하기 전에 인스턴스에서 향상된 네트워킹이 이미 활성화되어 있는지 확인합니다.

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.

기타 Linux 배포

시작하기 전에 인스턴스에서 향상된 네트워킹이 이미 활성화되어 있는지 확인합니다. 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 참조하세요.

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 참조하세요.

   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 콘솔을 사용하거나 stop-instances(AWS CLI) 또는 Stop-EC2Instance(AWS Tools for Windows PowerShell) 명령 중 하나를 사용하여 인스턴스를 중지합니다.

   [인스턴스 스토어 지원 인스턴스] 속성을 수정하기 위해 인스턴스를 중지할 수 없습니다. 대신 다음 절차로 진행합니다.

8. 사용자의 로컬 컴퓨터에서 다음 명령 중 하나를 사용하여 확장 네트워크 속성을 활성화합니다.

   AWS CLI

   다음과 같이 modify-instance-attribute 명령을 실행합니다.

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 -\
    -sriov-net-support simple
```

   PowerShell

   다음과 같이 Edit-EC2InstanceAttribute cmdlet을 사용합니다.

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -SriovNetSupport "simple"
```

9. (선택 사항) 의 설명에 따라 인스턴스에서 AMI를 생성합니다Amazon EBS 지원 AMI 생성 생성된 AMI는 인스턴스의 확장 네트워크 속성을 상속합니다. 따라서 이 AMI를 사용하여 기본적으로 향상된 네트워킹 기능이 활성화된 상태로 다른 인스턴스를 시작할 수 있습니다.

   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 컴퓨터에서 Amazon EC2 콘솔을 사용하거나 start-instances(AWS CLI) 또는 Start-EC2Instance(AWS Tools for Windows PowerShell) 명령 중 하나를 사용하여 인스턴스를 시작합니다.
11. (선택 사항) 인스턴스에 연결하여 모듈의 설치 여부를 확인합니다.

향상된 네트워킹 기능을 사용하려면(인스턴스 스토어 지원 인스턴스)

이전 절차에서 인스턴스를 중지한 단계까지 진행합니다. 인스턴스 스토어 지원 AMI 생성에 설명된 것처럼 새 AMI를 생성하고, AMI를 등록할 때 향상된 네트워킹 속성을 활성화합니다.

AWS CLI

다음과 같이 register-image 명령을 실행합니다.

```
aws ec2 register-image --sriov-net-support simple ...
```

PowerShell

다음과 같이 Register-EC2Image를 사용합니다.

```
Register-EC2Image -SriovNetSupport "simple" ...
```

Windows

확장 네트워크를 설정하지 않은 상태로 인스턴스를 시작한 경우에는 인스턴스에 필요한 네트워크 어댑터 드라이버를 다운로드하여 설치한 다음 `sriovNetSupport` 인스턴스 속성을 설정하여 확장 네트워크를 활성화해야 합니다. 이 속성 또는 지원되는 인스턴스 유형만 사용할 수 있습니다. 자세한 내용은 Amazon EC2 인스턴스의 향상된 네트워킹 섹션을 참조하세요.

> **Important**
>
> Windows AMI의 최신 드라이버 업데이트를 보려면 AWS Windows AMI 참조의 Windows AMI version history를 참조하세요.

향상된 네트워킹을 활성화하려면

1. 인스턴스 연결 후 로컬 관리자로 로그인합니다.
2. [Windows Server 2016 이상] 다음 EC2 Launch PowerShell 스크립트를 실행하여 드라이버가 설치된 후의 인스턴스를 구성합니다.

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -
Schedule
```

> **Important**
>
> 초기화 인스턴스 EC2 Launch 스크립트를 활성화하면 관리자 암호가 재설정됩니다. 관리자 암호 재설정은 구성 파일을 수정하여 비활성화할 수 있는데 초기화 작업에 대한 설정에서 이를 지정하면 됩니다.

3. 인스턴스에서 운영 체제에 대한 인텔 네트워크 어댑터 드라이버를 다운로드합니다.

   - Windows Server 2022

     다운로드 페이지로 이동하고 `Wired_driver_version_x64.zip`을 다운로드합니다.
   - Windows Server 2019(Server 버전 1809 이상 포함)*

     다운로드 페이지로 이동하고 `Wired_driver_version_x64.zip`을 다운로드합니다.
   - Windows Server 2016(Server 버전 1803 이하 포함)*

     다운로드 페이지로 이동하고 `Wired_driver_version_x64.zip`을 다운로드합니다.
   - Windows Server 2012 R2

     다운로드 페이지로 이동하고 `Wired_driver_version_x64.zip`을 다운로드합니다.
   - Windows Server 2012

     다운로드 페이지로 이동하고 `Wired_driver_version_x64.zip`을 다운로드합니다.
   - Windows Server 2008 R2

     다운로드 페이지로 이동하고 `PROWinx64Legacy.exe`을 다운로드합니다.

*Server 버전 1803 및 이전 버전과 1809 및 이후 버전은 Intel 드라이버 및 소프트웨어 페이지에 구체적으로 설명되지 않았습니다.

4. 운영 체제에 대한 인텔 네트워크 어댑터 드라이버를 설치합니다.

   - Windows Server 2008 R2

     1. 다운로드(Downloads) 폴더에서 PROWinx64Legacy.exe 파일을 찾고 이름을 PROWinx64Legacy.zip으로 바꿉니다.
     2. PROWinx64Legacy.zip 파일 내용의 압축을 풉니다.
     3. 명령줄을 열고 압축을 푼 폴더로 이동한 후 다음 명령을 실행하여 pnputil 유틸리티를 통해 INF 파일을 드라이버 스토어에 추가하고 설치합니다.

     ```
     C:\> pnputil -a PROXGB\Winx64\NDIS62\vxn62x64.inf
     ```

   - Windows Server 2022, Windows Server 2019, Windows Server 2016, Windows Server 2012 R2 및 Windows Server 2012

     1. 다운로드(Downloads) 폴더에서 Wired_driver_*version*_x64.zip 파일 내용의 압축을 풉니다.
     2. 명령줄을 열고 압축을 푼 폴더로 이동한 후 다음 명령 중 하나를 실행하여 pnputil 유틸리티를 통해 INF 파일을 드라이버 스토어에 추가하고 설치합니다.

        - Windows Server 2022

        ```
        pnputil -i -a PROXGB\Winx64\NDIS68\vxn68x64.inf
        ```

        - Windows Server 2019

        ```
        pnputil -i -a PROXGB\Winx64\NDIS68\vxn68x64.inf
        ```

        - Windows Server 2016

        ```
        pnputil -i -a PROXGB\Winx64\NDIS65\vxn65x64.inf
        ```

        - Windows Server 2012 R2

        ```
        pnputil -i -a PROXGB\Winx64\NDIS64\vxn64x64.inf
        ```

        - Windows Server 2012

        ```
        pnputil -i -a PROXGB\Winx64\NDIS63\vxn63x64.inf
        ```

5. 사용자의 로컬 컴퓨터에서 다음 명령 중 하나를 사용하여 확장 네트워크 속성을 활성화합니다.

AWS CLI

다음과 같이 modify-instance-attribute 명령을 실행합니다.

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --sriov-net-support simple
```

PowerShell

다음과 같이 Edit-EC2InstanceAttribute cmdlet을 사용합니다.

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -SriovNetSupport "simple"
```

6. (선택 사항) 의 설명에 따라 인스턴스에서 AMI를 생성합니다Amazon EBS 지원 AMI 생성 생성된 AMI는 인스턴스의 확장 네트워크 속성을 상속합니다. 따라서 이 AMI를 사용하여 기본적으로 향상된 네트워킹 기능이 활성화된 상태로 다른 인스턴스를 시작할 수 있습니다.
7. 로컬 컴퓨터에서 Amazon EC2 콘솔을 사용하거나 start-instances(AWS CLI) 또는 Start-EC2Instance(AWS Tools for Windows PowerShell) 명령 중 하나를 사용하여 인스턴스를 시작합니다.

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 켜고 구성하려면 어떻게 해야 합니까?를 참조하세요.

EC2 인스턴스의 ENA 설정에 대한 네트워크 성능 모니터링

Elastic Network Adapter(ENA) 드라이버는 드라이버가 활성화된 인스턴스의 네트워크 성능 지표를 게시합니다. 이러한 지표를 사용하여 인스턴스 성능 문제를 해결하고, 워크로드에 적합한 인스턴스 크기

를 선택하며, 조정 작업을 사전 예방적으로 계획하고, 애플리케이션을 벤치마킹하여 인스턴스의 가용 성능을 최대화할지 여부를 결정할 수 있습니다.

Amazon EC2는 인스턴스 수준에서 네트워크 최대값을 정의하여 인스턴스 크기 전반에 걸쳐 일관된 네트워크 성능을 제공하는 등 고품질 네트워킹 환경을 보장합니다. AWS는 각 인스턴스에 대해 다음에 대한 최대값을 제공합니다.

- 대역폭 기능 - 각 EC2 인스턴스에는 인스턴스 유형 및 크기에 따라 집계 인바운드 및 아웃바운드 트래픽에 대한 최대 대역폭이 있습니다. 일부 인스턴스는 네트워크 I/O 크레딧 메커니즘을 사용하여 평균 대역폭 사용률을 기준으로 네트워크 대역폭을 할당합니다. Amazon EC2에는 AWS Direct Connect 및 인터넷에 대한 트래픽에도 최대 대역폭이 있습니다. 자세한 내용은 Amazon EC2 인스턴스 네트워크 대역폭 섹션을 참조하세요.
- 초당 패킷(PPS) 성능 - 각 EC2 인스턴스에는 인스턴스 유형 및 크기에 따라 최대 PPS 성능이 있습니다.
- 연결 추적 - 보안 그룹은 설정된 각 연결을 추적하여 반환 패킷이 예상대로 전달되는지 확인합니다. 인스턴스당 추적할 수 있는 최대 연결 수가 있습니다. 자세한 내용은 Amazon EC2 보안 그룹 연결 추적 섹션을 참조하세요.
- 링크-로컬 서비스 액세스 - Amazon EC2는 Amazon DNS 서비스, Instance Metadata Service 및 Amazon Time Sync Service와 같은 로컬 프록시 서비스에 대한 트래픽에서 네트워크 인터페이스당 최대 PPS를 제공합니다.

인스턴스의 네트워크 트래픽이 최대값을 초과하면 AWS은(는) 네트워크 패킷을 대기열에 넣은 다음 삭제하여 최대값을 초과하는 트래픽을 처리합니다. 네트워크 성능 지표를 사용하여 트래픽이 최대값을 초과하는 시기를 모니터링할 수 있습니다. 이러한 지표는 네트워크 트래픽에 미치는 영향과 가능한 네트워크 성능 문제를 실시간으로 알려줍니다.

내용

- 요구 사항
- ENA 드라이버에 대한 지표
- 인스턴스에 대한 네트워크 성능 지표 보기
- ENA Express의 지표
- ENA용 DPDK 드라이버를 통한 네트워크 성능 지표
- FreeBSD를 실행하는 인스턴스의 지표

요구 사항

Linux 인스턴스

- ENA 드라이버 버전 2.2.10 이상을 설치합니다. 설치된 버전을 확인하려면 ethtool 명령을 사용합니다. 다음 예에서는 버전이 최소 요구 사항을 충족합니다.

```
[ec2-user ~]$ ethtool -i eth0 | grep version
version: 2.2.10
```

  ENA 드라이버를 업그레이드하려면 향상된 네트워킹을 참조하세요.
- 이러한 지표를 Amazon CloudWatch로 가져오려면 CloudWatch 에이전트를 설치합니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 네트워크 성능 지표 수집을 참조하세요.
- `conntrack_allowance_available` 지표를 지원하려면 ENA 드라이버 버전 2.8.1 이상을 설치하세요.
- 송신 조각 PPS 제한인 1024를 재정의하려면 ENA 드라이버 버전 2.13.3 이상을 설치하세요.

Windows 인스턴스

- ENA 드라이버 버전 2.2.2 이상을 설치합니다. 설치된 버전을 확인하려면 다음과 같이 디바이스 관리자를 사용합니다.
  1. `devmgmt.msc`를 실행하여 디바이스 관리자를 엽니다.
  2. [네트워크 어댑터(Network Adapters)]를 확장합니다.
  3. [Amazon Elastic Network Adapter], [속성(Properties)]을 선택합니다.
  4. [드라이버(Driver)] 탭에서 [드라이버 버전(Driver Version)]을 찾습니다.

  ENA 드라이버를 업그레이드하려면 향상된 네트워킹을 참조하세요.
- 이러한 지표를 Amazon CloudWatch로 가져오려면 CloudWatch 에이전트를 설치합니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 고급 네트워크 지표 수집을 참조하세요.

ENA 드라이버에 대한 지표

ENA 드라이버는 인스턴스에 다음과 같은 지표를 실시간으로 전달합니다. 마지막 드라이버 재설정 이후 각 네트워크 인터페이스에서 대기열에 추가되거나 손실된 누적 패킷 수를 제공합니다.

지표	설명	지원
bw_in_allowance_exceeded	인바운드 집계 대역폭이 인스턴스의 최댓값을 초과하여 대기열에 추가되거나 손실된 패킷 수입니다.	모든 인스턴스 유형
bw_out_allowance_exceeded	아웃바운드 집계 대역폭이 인스턴스의 최댓값을 초과하여 대기열에 추가되거나 손실된 패킷 수입니다.	모든 인스턴스 유형
conntrack_allowance_exceeded	연결 추적에서 인스턴스의 최댓값이 초과되어 새 연결을 설정하지 못했기 때문에 손실된 패킷 수입니다. 이로 인해 인스턴스의 수신 또는 송신 트래픽에 대한 패킷 손실이 발생할 수 있습니다.	모든 인스턴스 유형
conntrack_allowance_available	해당 인스턴스 유형의 연결 추적 허용량에 도달하기 전에 인스턴스에서 설정할 수 있는 추적된 연결 수입니다.	Nitro 기반 인스턴스 전용
linklocal_allowance_exceeded	로컬 프록시 서비스에 대한 트래픽의 PPS가 네트워크 인터페이스의 최댓값을 초과하여 손실된 패킷 수입니다. 이는 Amazon DNS 서비스, Instance Metadata Service 및 Amazon Time Sync Service에 대한 트래픽에 영향을 주지만, 사용자 지정 DNS 해석기에 대한 트래픽에는 영향을 주지 않습니다.	모든 인스턴스 유형

지표	설명	지원
pps_allowance_exceeded	양방향 PPS가 인스턴스의 최댓값을 초과하여 대기열에 추가되거나 손실된 패킷 수입니다. *	모든 인스턴스 유형

* ENA Linux 드라이버 v2.13.3 이상에 대한 조각 프록시 모드 설정에 따라 이 제한에는 네트워크 인터페이스에 대해 1024PPS를 초과하는 송신 조각 드롭도 포함될 수 있습니다. Linux 드라이버에 대해 조각 프록시 모드가 활성화된 경우 송신 조각 드롭은 일반적으로 적용되고 표준 PPS 허용량 내에서 계산되는 1024PPS 제한을 우회합니다. 조각 프록시 모드는 기본적으로 비활성화되어 있습니다.

인스턴스에 대한 네트워크 성능 지표 보기

사용하는 절차는 인스턴스의 운영 체제에 따라 달라집니다.

Linux 인스턴스

자주 사용하는 도구에 지표를 게시하여 지표 데이터를 시각화할 수 있습니다. 예를 들어 CloudWatch 에이전트를 사용하여 Amazon CloudWatch에 지표를 게시할 수 있습니다. 이 에이전트를 사용하면 개별 지표를 선택하고 게시를 제어할 수 있습니다.

ethtool을 사용하여 다음과 같이 eth0과 같은 각 네트워크 인터페이스에 대한 지표를 검색할 수도 있습니다.

```
[ec2-user ~]$ ethtool -S eth0
     bw_in_allowance_exceeded: 0
     bw_out_allowance_exceeded: 0
     pps_allowance_exceeded: 0
     conntrack_allowance_exceeded: 0
     linklocal_allowance_exceeded: 0
     conntrack_allowance_available: 136812
```

Windows 인스턴스

Windows 성능 카운터의 소비자를 사용하여 지표를 볼 수 있습니다. EnaPerfCounters 매니페스트에 따라 데이터를 구문 분석할 수 있습니다. EnaPerfCounters 매니페스트는 성능 카운터 공급자 및 해당 카운터 세트를 정의하는 XML 파일입니다.

매니페스트를 다운로드하려면 다음을 수행합니다.

ENA 드라이버 2.2.2 이상이 포함된 AMI를 사용하여 인스턴스를 시작했거나 ENA 드라이버 2.2.2의 드라이버 패키지의 설치 스크립트를 사용한 경우 매니페스트가 이미 설치되어 있습니다. 매니페스트를 수동으로 설치하려면 다음 단계를 수행합니다.

1. 다음 명령을 사용하여 기존 매니페스트를 제거합니다.

```
unlodctr /m:EnaPerfCounters.man
```

2. 드라이버 설치 패키지에서 매니페스트 파일 `EnaPerfCounters.man`을(를) `%SystemRoot%\System32\drivers`(으)로 복사합니다.
3. 다음 명령을 사용하여 새 매니페스트를 설치합니다.

```
lodctr /m:EnaPerfCounters.man
```

성능 모니터를 사용하여 지표를 보려면 다음을 수행합니다.

1. 성능 모니터를 엽니다.
2. Ctrl+N을 눌러 새 카운터를 추가합니다.
3. 목록에서 [ENA 패킷 셰이핑(ENA Packets Shaping)]을 선택합니다.
4. 모니터링할 인스턴스를 선택하고 [추가(Add)]를 선택합니다.
5. 확인을 선택합니다.

ENA Express의 지표

ENA Express는 AWS Scalable Reliable Datagram(SRD) 기술로 구동됩니다. SRD는 동적 라우팅을 사용하여 스루풋을 늘리고 테일 지연 시간을 최소화하는 고성능 네트워크 전송 프로토콜입니다. 전송 인스턴스와 수신 인스턴스 모두에서 네트워크 인터페이스 연결에 대해 ENA Express를 활성화한 경우 ENA Express 지표를 사용하여 인스턴스에서 SRD 기술이 제공하는 성능 개선의 이점을 최대한 활용할 수 있습니다. 예시:

- 리소스를 평가하여 추가로 SRD 연결을 설정할 용량이 충분한지 확인합니다.
- 적절한 전송 패킷의 SRD 사용을 방해하는 잠재적 문제가 있는 위치를 식별합니다.
- 인스턴스에서 SRD를 사용하는 전송 트래픽의 비율을 계산합니다.
- 인스턴스에서 SRD를 사용하는 수신 트래픽의 비율을 계산합니다.

> **Note**
>
> 지표를 생성하려면 드라이버 버전 2.8 이상을 사용합니다.

ENA Express에 대해 필터링된 Linux 인스턴스의 지표 목록을 보려면 네트워크 인터페이스(이 예제에서는 `eth0`으로 표시됨)에 대해 다음 ethtool 명령을 실행합니다. `ena_srd_mode` 지표의 값을 기록해 둡니다.

```
[ec2-user ~]$ ethtool -S eth0 | grep ena_srd
NIC statistics:
 ena_srd_mode: 1
 ena_srd_tx_pkts: 0
 ena_srd_eligible_tx_pkts: 0
 ena_srd_rx_pkts: 0
 ena_srd_resource_utilization: 0
```

다음 지표는 ENA Express가 활성화된 모든 인스턴스에 제공됩니다.

ena_srd_mode

어떤 ENA Express 기능이 활성화되어 있는지 설명합니다. 값은 다음과 같습니다.

- `0` = ENA Express 꺼짐, UDP 꺼짐
- 1 = ENA Express 켜짐, UDP 꺼짐
- 2 = ENA Express 꺼짐, UDP 켜짐

> **Note**
>
> ENA Express가 원래 활성화되어 있고, UDP가 ENA Express를 사용하도록 구성된 경우에만 이렇게 나타납니다. UDP 트래픽에 대한 이전 값이 유지됩니다.

- `3` = ENA Express 켜짐, UDP 켜짐

ena_srd_eligible_tx_pkts

네트워크 번호는 다음과 같습니다.

- 전송 인스턴스 유형과 수신 인스턴스 유형이 모두 지원되어야 합니다. 자세한 내용은 ENA Express를 지원하는 인스턴스 유형 표를 참조하세요.
- 전송 인스턴스와 수신 인스턴스 모두에 ENA Express가 구성되어 있어야 합니다.

- 전송 인스턴스와 수신 인스턴스가 동일한 가용 영역에서 실행되어야 합니다.
- 인스턴스 간 네트워크 경로에 미들웨어 박스가 포함되지 않아야 합니다. ENA Express는 현재 미들웨어 박스를 지원하지 않습니다.

> **Note**
>
> ENA Express 적합성 지표는 소스 및 대상 요구 사항과 두 엔드포인트 간의 네트워크를 보여줍니다. 이미 카운팅되었더라도 적격 패킷이 적합하지 않을 수 있습니다. 예를 들어 적격 패킷이 최대 전송 단위(MTU) 한도를 초과하는 경우, 해당 패킷이 적격 패킷으로 카운터에 반영되더라도 표준 ENA 전송으로 폴백됩니다.

ena_srd_tx_pkts

일정 기간 동안 전송한 SRD 패킷의 수입니다.

ena_srd_rx_pkts

일정 기간 동안 수신한 SRD 패킷의 수입니다.

ena_srd_resource_utilization

동시 SRD 연결에 허용된 최대 메모리 사용률 중 인스턴스에 사용된 비율입니다.

패킷 전송에 SRD가 사용되는지 확인하려면, 일정 기간 동안 전송된 SRD 패킷 수(`ena_srd_tx_pkts` 지표)와 적격 패킷 수(`ena_srd_eligible_tx_pkts` 지표)를 비교해보면 됩니다.

송신 트래픽(전송 패킷)

송신 트래픽이 예상대로 SRD를 사용하는지 확인하려면, 일정 기간 동안 전송된 SRD 패킷 수(`ena_srd_eligible_tx_pkts`)와 SRD 적격 패킷 수(`ena_srd_tx_pkts`)를 비교합니다.

적격 패킷 수와 전송된 SRD 패킷 수 간의 큰 차이는 리소스 사용률 문제로 인해 발생하는 경우가 많습니다. 인스턴스에 연결된 네트워크 카드가 최대 리소스를 모두 사용하거나 패킷이 MTU 한도를 초과하면, 적격 패킷이 SRD를 통해 전송되지 않으며 표준 ENA 전송으로 폴백해야 합니다. 라이브 마이그레이션이나 라이브 서버 업데이트 중에도 패킷에서 이러한 차이가 발생할 수 있습니다. 근본 원인을 파악하려면 추가적인 문제 해결 프로세스가 필요합니다.

> **Note**
>
> 적격 패킷 수와 SRD 패킷 수 간의 사소한 차이는 무시해도 됩니다. 예를 들어 인스턴스가 SRD 트래픽을 전송하기 위해 다른 인스턴스에 연결할 때 이 문제가 발생할 수 있습니다.

일정 기간 동안 총 송신 트래픽 중 SRD를 사용한 트래픽의 비율을 확인하려면, 전송된 SRD 패킷 수(`ena_srd_tx_pkts`)를 해당 기간 동안 인스턴스에 전송된 총 패킷 수(`NetworkPacketOut`)와 비교합니다.

수신 트래픽(수신 패킷)

SRD를 사용한 수신 트래픽의 비율을 확인하려면, 일정 기간 동안 수신된 SRD 패킷 수(`ena_srd_rx_pkts`)를 해당 기간 동안 인스턴스에 수신된 총 패킷 수(`NetworkPacketIn`)와 비교합니다.

리소스 사용률

리소스 사용률은 일정 시간에 단일 인스턴스가 보유할 수 있는 동시 SRD 연결 수를 기준으로 합니다. 리소스 사용률 지표(`ena_srd_resource_utilization`)는 인스턴스의 현재 사용률을 추적합니다. 사용률이 100%에 가까워지면 성능 문제가 발생할 수 있습니다. ENA Express가 SRD에서 표준 ENA 전송으로 폴백되므로 패킷이 손실될 가능성이 커집니다. 높은 리소스 사용률은 인스턴스를 스케일 아웃하여 네트워크 성능을 개선해야 할 때임을 나타내는 징후입니다.

> **Note**
>
> 인스턴스의 네트워크 트래픽이 최대값을 초과하면 AWS은(는) 네트워크 패킷을 대기열에 넣은 다음 삭제하여 최대값을 초과하는 트래픽을 처리합니다.

Persistence

송신 및 수신 지표는 인스턴스에 대해 ENA Express가 활성화되어 있는 동안 누적됩니다. ENA Express가 비활성화되면 지표 누적이 중단되지만, 인스턴스가 실행되는 동안에는 계속 유지됩니다. 인스턴스가 재부팅 또는 종료되거나 네트워크 인터페이스가 인스턴스에서 분리되면 지표가 재설정됩니다.

ENA용 DPDK 드라이버를 통한 네트워크 성능 지표

ENA 드라이버 버전 2.2.0 이상은 네트워크 지표 보고를 지원합니다. DPDK 20.11에는 ENA 드라이버 2.2.0이 포함되어 있으며 이 기능을 지원하는 최초의 DPDK 버전입니다.

DPDK 드라이버 v25.03 이상은 조각 프록시 모드를 지원합니다. DPDK 드라이버에 대해 조각 프록시 모드가 활성화된 경우 송신 조각 드롭은 일반적으로 적용되고 표준 PPS 허용량 내에서 계산되는 1024PPS 제한을 우회합니다. 조각 프록시 모드는 기본적으로 비활성화되어 있습니다.

예제 애플리케이션을 사용하여 DPDK 통계를 볼 수 있습니다. 예제 애플리케이션의 대화형 버전을 시작하려면 다음 명령을 실행합니다.

```
./app/dpdk-testpmd -- -i
```

이 대화형 세션 내에서 포트에 대한 확장된 통계를 검색하는 명령을 입력할 수 있습니다. 다음 예제 명령은 포트 0에 대한 통계를 검색합니다.

```
show port xstats 0
```

다음은 DPDK 예제 애플리케이션과의 대화형 세션에 대한 예입니다.

```
[root@ip-192.0.2.0 build]# ./app/dpdk-testpmd -- -i
        EAL: Detected 4 lcore(s)
        EAL: Detected 1 NUMA nodes
        EAL: Multi-process socket /var/run/dpdk/rte/mp_socket
        EAL: Selected IOVA mode 'PA'
        EAL: Probing VFIO support...
        EAL:   Invalid NUMA socket, default to 0
        EAL:   Invalid NUMA socket, default to 0
        EAL: Probe PCI driver: net_ena (1d0f:ec20) device: 0000:00:06.0
(socket 0)
        EAL: No legacy callbacks, legacy socket not created
        Interactive-mode selected

        Port 0: link state change event
        testpmd: create a new mbuf pool <mb_pool_0>: n=171456,
size=2176, socket=0
        testpmd: preferred mempool ops selected: ring_mp_mc

        Warning! port-topology=paired and odd forward ports number, the
last port will pair with itself.
```

```
        Configuring Port 0 (socket 0)
        Port 0: 02:C7:17:A2:60:B1
        Checking link statuses...
        Done
        Error during enabling promiscuous mode for port 0: Operation
not supported - ignore
        testpmd> show port xstats 0
        ###### NIC extended statistics for port 0
        rx_good_packets: 0
        tx_good_packets: 0
        rx_good_bytes: 0
        tx_good_bytes: 0
        rx_missed_errors: 0
        rx_errors: 0
        tx_errors: 0
        rx_mbuf_allocation_errors: 0
        rx_q0_packets: 0
        rx_q0_bytes: 0
        rx_q0_errors: 0
        tx_q0_packets: 0
        tx_q0_bytes: 0
        wd_expired: 0
        dev_start: 1
        dev_stop: 0
        tx_drops: 0
        bw_in_allowance_exceeded: 0
        bw_out_allowance_exceeded: 0
        pps_allowance_exceeded: 0
        conntrack_allowance_exceeded: 0
        linklocal_allowance_exceeded: 0
        rx_q0_cnt: 0
        rx_q0_bytes: 0
        rx_q0_refill_partial: 0
        rx_q0_bad_csum: 0
        rx_q0_mbuf_alloc_fail: 0
        rx_q0_bad_desc_num: 0
        rx_q0_bad_req_id: 0
        tx_q0_cnt: 0
        tx_q0_bytes: 0
        tx_q0_prepare_ctx_err: 0
        tx_q0_linearize: 0
        tx_q0_linearize_failed: 0
        tx_q0_tx_poll: 0
        tx_q0_doorbells: 0
```

```
        tx_q0_bad_req_id: 0
        tx_q0_available_desc: 1023
        testpmd>
```

예제 애플리케이션 및 확장 통계 검색에 이 애플리케이션을 사용하는 방법에 대한 자세한 내용은 DPDK 설명서의 Testpmd 애플리케이션 사용 설명서를 참조하세요.

FreeBSD를 실행하는 인스턴스의 지표

버전 2.3.0부터 ENA FreeBSD 드라이버는 FreeBSD를 실행하는 인스턴스에 대한 네트워크 성능 지표 수집을 지원합니다. FreeBSD 지표 수집을 활성화하려면 다음 커밋을 입력하고 *interval*을 1에서 3600 사이의 값으로 설정합니다. FreeBSD 지표를 수집하는 빈도(초)를 지정합니다.

```
sysctl dev.ena.network_interface.eni_metrics.sample_interval=interval
```

예를 들어 다음 명령은 10초마다 네트워크 인터페이스 1에서 FreeBSD 지표를 수집하도록 드라이버를 설정합니다.

```
sysctl dev.ena.1.eni_metrics.sample_interval=10
```

FreeBSD 지표 수집을 끄려면 앞의 명령을 실행하고 *interval*을 0으로 지정하면 됩니다.

FreeBSD 지표 수집을 활성화한 후에는 다음 명령을 실행하여 수집된 지표의 최신 세트를 검색할 수 있습니다.

```
sysctl dev.ena.network_interface.eni_metrics
```

Linux에서 ENA 커널 드라이버 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 섹션에서 시작하세요.

6세대 인스턴스 유형으로 마이그레이션한 후 성능 저하가 발생하는 경우 지식 센터의 최대 네트워크 성능을 얻으려면 EC2 인스턴스를 6세대 인스턴스로 마이그레이션하기 전에 어떤 작업이 필요한가요?를 참조하세요.

인스턴스에 연결할 수 있는 경우 이 주제의 이후 섹션에서 다루는 장애 탐지 및 복구 메커니즘을 사용하여 진단 정보를 수집할 수 있습니다.

목차

- 연결 문제 해결
- 연결 유지 메커니즘
- 레지스터 읽기 시간 초과
- Statistics
- syslog의 드라이버 오류 로그
- 최적이 아닌 구성 알림

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 컴퓨터에서 Amazon EC2 콘솔, stop-instances 명령(AWS CLI) 또는 Stop-EC2Instance cmdlet(AWS Tools for PowerShell)을 사용하여 인스턴스를 중지합니다.
2. modify-instance-attribute 명령을 로컬 컴퓨터에서 `--no-ena-support` 옵션과 함께 사용하거나 Edit-EC2InstanceAttribute cmdlet을 `-EnaSupport $false` 파라미터와 함께 사용하여 향상된 네트워킹 속성을 비활성화합니다.
3. 로컬 컴퓨터에서 Amazon EC2 콘솔, start-instances 명령, 또는 Start-EC2Instance cmdlet을 사용하여 인스턴스를 시작합니다.
4. (선택 사항) 인스턴스에 연결한 후 ena의 단계에 따라 현재 커널 버전으로 EC2 인스턴스에서 ENA로 향상된 네트워킹 활성화 모듈을 다시 설치해 보십시오.

ENA를 사용하여 향상된 네트워킹을 비활성화하려면(인스턴스 스토어 지원 인스턴스)

1. 인스턴스 스토어 지원 AMI 생성에 설명된 대로 새 AMI를 생성합니다.
2. AMI를 등록할 때는 stop-instances 명령(AWS CLI)에 `--no-ena-support` 옵션을 포함하거나 Register-EC2Image cmdlet에 `-EnaSupport $false` 파라미터를 포함해야 합니다.

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, 그렇게 하지 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 나머지 절차에 대한 예제입니다.

```
[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
```

Statistics

네트워크 성능이 저하되거나 지연 시간 문제가 발생할 경우 디바이스 통계를 불러온 후 확인해야 합니다. 다음과 같이 ethtool을 사용하여 이러한 통계를 가져올 수 있습니다.

```
[ec2-user ~]$ ethtool -S ethN
NIC statistics:
 tx_timeout: 0
 suspend: 0
 resume: 0
 wd_expired: 0
 interface_up: 1
 interface_down: 0
 admin_q_pause: 0
 bw_in_allowance_exceeded: 0
 bw_out_allowance_exceeded: 0
 pps_allowance_exceeded: 0
 conntrack_allowance_available: 450878
 conntrack_allowance_exceeded: 0
 linklocal_allowance_exceeded: 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.

`suspend`: *N*

드라이버가 일시 중지 작업을 수행한 횟수입니다.

`resume`: *N*

드라이버가 다시 시작 작업을 수행한 횟수입니다.

`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*

실행 중인 상태의 관리 대기열을 찾지 못한 횟수입니다.

`bw_in_allowance_exceeded`: *N*

인바운드 집계 대역폭이 인스턴스의 최댓값을 초과하여 대기열에 추가되거나 손실된 패킷 수입니다.

`bw_out_allowance_exceeded`: *N*

아웃바운드 집계 대역폭이 인스턴스의 최댓값을 초과하여 대기열에 추가되거나 손실된 패킷 수입니다.

`pps_allowance_exceeded`: *N*

양방향 PPS가 인스턴스의 최댓값을 초과하여 대기열에 추가되거나 손실된 패킷 수입니다. *

`conntrack_allowance_available`: *N*

해당 인스턴스 유형의 연결 추적 허용량에 도달하기 전에 인스턴스에서 설정할 수 있는 추적된 연결 수입니다. Nitro 기반 인스턴스에 대해서만 사용 가능합니다. FreeBSD 인스턴스 또는 DPDK 환경에서는 지원되지 않습니다.

`conntrack_allowance_exceeded`: *N*

연결 추적에서 인스턴스의 최댓값이 초과되어 새 연결을 설정하지 못했기 때문에 손실된 패킷 수입니다. 이로 인해 인스턴스의 수신 또는 송신 트래픽에 대한 패킷 손실이 발생할 수 있습니다.

`linklocal_allowance_exceeded`: *N*

로컬 프록시 서비스에 대한 트래픽의 PPS가 네트워크 인터페이스의 최댓값을 초과하여 손실된 패킷 수입니다. 이는 Amazon DNS 서비스, Instance Metadata Service 및 Amazon Time Sync Service에 대한 트래픽에 영향을 주지만, 사용자 지정 DNS 해석기에 대한 트래픽에는 영향을 주지 않습니다.

`queue_`*N*`_tx_cnt`: *N*

이 대기열의 전송된 패킷 수입니다.

queue_*N*_tx_bytes: *N*

이 대기열의 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_linearize: *N*

이 대기열에 대해 SKB 선형화가 시도된 횟수입니다.

queue_*N*_tx_linearize_failed: *N*

이 대기열에 대해 SKB 선형화가 실패한 횟수입니다.

queue_*N*_tx_napi_comp: *N*

napi 핸들러가 이 대기열에 대해 napi_complete을 호출한 횟수입니다.

queue_*N*_tx_tx_poll: *N*

napi 핸들러가 이 대기열에 대해 예약된 횟수입니다.

queue_*N*_tx_doorbells: *N*

이 대기열의 전송 도어벨 수입니다.

queue_*N*_tx_prepare_ctx_err: *N*

이 대기열에 대해 ena_com_prepare_tx가 실패한 횟수입니다.

queue_*N*_tx_bad_req_id: *N*

이 대기열에 유효하지 않은 req_id입니다. 유효한 req_id는 0, -queue_size, -1입니다.

queue_*N*_tx_llq_buffer_copy: *N*

헤더 크기가 이 대기열의 llq 항목보다 큰 패킷 수입니다.

queue_*N*_tx_missed_tx: *N*

이 대기열에 대해 완료되지 않은 상태로 남은 패킷 수입니다.

`queue_N_tx_unmask_interrupt`: *N*

이 대기열에 대해 tx 인터럽트가 마스크되지 않은 횟수입니다.

`queue_N_rx_cnt`: *N*

이 대기열에 대해 수신된 패킷 수입니다.

`queue_N_rx_bytes`: *N*

이 대기열에 대해 수신된 바이트 수입니다.

`queue_N_rx_rx_copybreak_pkt`: *N*

rx 대기열이 이 대기열의 rx_copybreak 패킷 크기보다 작은 패킷을 수신한 횟수입니다.

`queue_N_rx_csum_good`: *N*

체크섬이 확인되었고 이 대기열에 대해 올바른 패킷을 rx 대기열이 수신한 횟수입니다.

`queue_N_rx_refil_partial`: *N*

드라이버가 rx 대기열의 빈 부분을 이 대기열에 대한 버퍼로 리필하는 데 실패한 횟수입니다. 이 값이 0이 아니면 메모리 리소스가 부족한 것입니다.

`queue_N_rx_bad_csum`: *N*

`rx` 대기열에 이 대기열에 대한 잘못된 체크섬이 포함된 횟수입니다(rx 체크섬 오프로드가 지원되는 경우에만 해당).

`queue_N_rx_page_alloc_fail`: *N*

이 대기열에 대해 페이지 할당이 실패한 횟수입니다. 이 값이 0이 아니면 메모리 리소스가 부족한 것입니다.

`queue_N_rx_skb_alloc_fail`: *N*

이 대기열에 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_bad_req_id: *N*

이 대기열의 req_id가 유효하지 않습니다. 유효한 req_id는 [0, queue_size - 1]입니다.

queue_*N*_rx_empty_rx_ring: *N*

이 대기열에 대해 rx 대기열이 빈 횟수입니다.

queue_*N*_rx_csum_unchecked: *N*

이 대기열에 대해 체크섬이 확인되지 않은 패킷을 rx 대기열이 받은 횟수입니다.

queue_*N*_rx_xdp_aborted: *N*

XDP 패킷이 XDP_ABORT로 분류된 횟수입니다.

queue_*N*_rx_xdp_drop: *N*

XDP 패킷이 XDP_DROP으로 분류된 횟수입니다.

queue_*N*_rx_xdp_pass: *N*

XDP 패킷이 XDP_PASS로 분류된 횟수입니다.

queue_*N*_rx_xdp_tx: *N*

XDP 패킷이 XDP_TX로 분류된 횟수입니다.

queue_*N*_rx_xdp_invalid: *N*

패킷에 대한 XDP 반환 코드가 올바르지 않았던 횟수입니다.

queue_*N*_rx_xdp_redirect: *N*

XDP 패킷이 XDP_REDIRECT로 분류된 횟수입니다.

queue_*N*_xdp_tx_cnt: *N*

이 대기열의 전송된 패킷 수입니다.

queue_*N*_xdp_tx_bytes: *N*

이 대기열의 전송된 바이트 수입니다.

queue_*N*_xdp_tx_queue_stop: *N*

이 대기열이 가득 차서 중지된 횟수입니다.

queue_*N*_xdp_tx_queue_wakeup: *N*

이 대기열이 중지되었다가 재개된 횟수입니다.

queue_*N*_xdp_tx_dma_mapping_err: *N*

직접 메모리 액세스 오류 수입니다. 이 값이 0이 아니면 시스템 리소스가 부족한 것입니다.

queue_*N*_xdp_tx_linearize: *N*

이 대기열에 대해 XDP 버퍼 선형화가 시도된 횟수입니다.

queue_*N*_xdp_tx_linearize_failed: *N*

이 대기열에 대해 XDP 버퍼 선형화가 실패한 횟수입니다.

queue_*N*_xdp_tx_napi_comp: *N*

napi 핸들러가 이 대기열에 대해 napi_complete를 호출한 횟수입니다.

queue_*N*_xdp_tx_tx_poll: *N*

napi 핸들러가 이 대기열에 대해 예약된 횟수입니다.

queue_*N*_xdp_tx_doorbells: *N*

이 대기열의 전송 도어벨 수입니다.

queue_*N*_xdp_tx_prepare_ctx_err: *N*

이 대기열에 대해 ena_com_prepare_tx가 실패한 횟수입니다. 이 값은 항상 0이어야 합니다. 그렇지 않은 경우 드라이버 로그를 참조하세요.

queue_*N*_xdp_tx_bad_req_id: *N*

이 대기열의 req_id가 유효하지 않습니다. 유효한 req_id는 [0, queue_size - 1]입니다.

queue_*N*_xdp_tx_llq_buffer_copy: *N*

이 대기열에 대해 llq 버퍼 복사를 사용하여 헤더를 복사한 패킷 수입니다.

queue_*N*_xdp_tx_missed_tx: *N*

tx 대기열 항목이 이 대기열의 완료 제한 시간을 놓친 횟수입니다.

queue_*N*_xdp_tx_unmask_interrupt: *N*

이 대기열에 대해 tx 인터럽트가 마스크되지 않은 횟수입니다.

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 디바이스에서 지원되지 않습니다.
- 12: RSS 간접 테이블 구성은 이 디바이스에서 지원되지 않습니다.
- 18: RSS 해시 입력 구성은 이 디바이스에서 지원되지 않습니다.
- 20: 인터럽트 조절은 이 디바이스에서 지원되지 않습니다.
- 27: Elastic Network Adapte(ENA) 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.

최적이 아닌 구성 알림

ENA 디바이스는 드라이버에서 변경할 수 있는 최적이 아닌 구성 설정을 탐지합니다. 디바이스는 ENA 드라이버에 알리고 콘솔에 경고를 기록합니다. 다음 예는 경고 메시지의 형식을 보여줍니다.

```
Sub-optimal configuration notification code: 1. Refer to AWS ENA documentation for
 additional details and mitigation options.
```

다음 목록은 최적이 아닌 구성 결과에 대한 알림 코드 세부 정보와 권장 조치를 보여줍니다.

- 코드 1: 넓은 LLQ 구성을 갖춘 ENA Express는 권장되지 않음

  ENA Express ENI는 넓은 LLQ로 구성되어 있습니다. 해당 구성은 최적이 아니며 ENA Express의 성능에 영향을 미칠 수 있습니다. ENA Express ENI를 사용할 때는 다음과 같이 넓은 LLQ 설정을 비활성화하는 것이 좋습니다.

  ```
  sudo rmmod ena && sudo modprobe ena force_large_llq_header=0
  ```

  ENA Express의 최적 구성에 대한 자세한 내용은 ENA Express로 EC2 인스턴스 간 네트워크 성능 향상을 참조하세요.
- 코드 2: Tx 대기열 깊이가 최적이 아닌 ENA Express ENI는 권장되지 않음

  ENA Express ENI가 최적이 아닌 Tx 대기열 깊이로 구성되어 있습니다. 해당 구성은 ENA Express의 성능에 영향을 미칠 수 있습니다. ENA Express ENI를 사용하는 경우 다음과 같이 모든 Tx 대기열을 네트워크 인터페이스의 최댓값으로 확장하는 것이 좋습니다.

  다음 ethtool 명령을 실행하여 LLQ 크기를 조정할 수 있습니다. 넓은 LLQ를 제어, 쿼리 및 활성화하는 방법에 대한 자세한 내용은 Amazon Drivers GitHub 리포지토리의 ENA 설명서에서 Linux 커널 드라이버에 대한 Large Low-Latency Queue (Large LLQ) 주제를 참조하세요.

  ```
  ethtool -g interface
  ```

  최대 깊이로 Tx 대기열 설정:

  ```
  ethtool -G interface tx depth
  ```

  ENA Express의 최적 구성에 대한 자세한 내용은 ENA Express로 EC2 인스턴스 간 네트워크 성능 향상을 참조하세요.
- 코드 3: 일반 LLQ 크기 및 Tx 패킷 트래픽을 사용하는 ENA에서 헤더 지원 최대 크기를 초과함

  기본적으로 ENA LLQ는 Tx 패킷 헤더 크기를 최대 96바이트까지 지원합니다. 패킷 헤더 크기가 96바이트보다 크면 패킷이 삭제됩니다. 이 문제를 완화하려면 지원되는 Tx 패킷 헤더 크기를 최대 224바이트로 늘리는 넓은 LLQ를 활성화하는 것이 좋습니다.

하지만 넓은 LLQ를 활성화하면 최대 Tx 링 크기가 1,000개 항목에서 512개 항목으로 줄어듭니다. 넓은 LLQ는 모든 Nitro v4 이상의 인스턴스 유형에서 기본적으로 활성화됩니다.

- Nitro v4 인스턴스 유형에서 기본 최대 넓은 LLQ Tx 링 크기는 512개 항목이며, 변경할 수 없습니다.
- Nitro v5 인스턴스 유형에서 기본 넓은 LLQ Tx 링 크기는 512개 항목이며, 최대 1,000개까지 늘릴 수 있습니다.

다음 ethtool 명령을 실행하여 LLQ 크기를 조정할 수 있습니다. 넓은 LLQ를 제어, 쿼리 및 활성화하는 방법에 대한 자세한 내용은 Amazon Drivers GitHub 리포지토리의 ENA 설명서에서 Linux 커널 드라이버에 대한 Large Low-Latency Queue (Large LLQ) 주제를 참조하세요.

Tx 대기열의 최대 깊이 찾기:

```
ethtool -g interface
```

최대 깊이로 Tx 대기열 설정:

```
ethtool -G interface tx depth
```

Elastic Network Adapter Windows 드라이버 문제 해결

Elastic Network Adapter(ENA)는 운영 체제 상태를 개선하고 Windows 인스턴스의 작동을 방해할 수 있는 예기치 않은 하드웨어 동작이나 오류를 줄이도록 설계되었습니다. ENA 아키텍처는 디바이스 또는 드라이버 장애가 운영 체제에 영향을 주지 않도록 최대한 보호합니다.

인스턴스에 대한 진단 정보 수집

Windows 운영 체제(OS) 도구를 여는 단계는 인스턴스에 설치된 OS 버전에 따라 다릅니다. 다음 섹션에서는 실행(Run) 대화 상자를 사용하여 모든 OS 버전에서 동일하게 작동하는 도구를 엽니다. 그러나 원하는 방법을 사용하여 이러한 도구에 액세스할 수 있습니다.

실행(Run) 대화 상자에 액세스

- Windows 로고 키 조합 사용: `Windows + R`
- 검색 창 사용:
  - 검색줄에 `run`를 입력합니다.
  - 검색 결과에서 실행(Run) 애플리케이션을 선택합니다.

일부 단계에서는 속성 또는 상황에 맞는 작업에 액세스하기 위해 컨텍스트 메뉴가 필요합니다. OS 버전과 하드웨어에 따라 여러 가지 방법이 있습니다.

컨텍스트 메뉴에 액세스

- 마우스 사용: 항목을 마우스 오른쪽 버튼으로 클릭하여 컨텍스트 메뉴를 표시합니다.
- 키보드 사용:
  - OS 버전에 따라 `Shift+F10` 또는 `Ctrl+Shift+F10`을 사용합니다.
  - 키보드에 컨텍스트 키가 있는 경우(상자에 가로줄 3개) 원하는 항목을 선택한 다음 컨텍스트 키를 누릅니다.

인스턴스에 연결할 수 있는 경우 다음 기술을 사용하여 문제 해결을 위한 진단 정보를 수집합니다.

ENA 디바이스 상태 확인

Windows 장치 관리자를 사용하여 ENA Windows 드라이버의 상태를 확인하려면 다음 단계를 따르세요.

1. 이전 섹션에서 설명한 방법 중 하나를 사용하여 실행(Run) 대화 상자를 엽니다.
2. Windows 장치 관리자를 열려면 실행(Run) 상자에 `devmgmt.msc`를 입력합니다.
3. 확인을 선택합니다. 그러면 장치 관리자 창이 열립니다.
4. 네트워크 어댑터(Network adapters) 왼쪽의 화살표를 선택하여 목록을 확장합니다.
5. 이름을 선택하거나 Amazon Elastic Network Adapter의 컨텍스트 메뉴를 연 다음 속성(Properties)을 선택합니다. 그러면 Amazon Elastic Network Adapter 속성 대화 상자가 열립니다.
6. 일반 탭의 메시지가 “이 장치는 제대로 작동하고 있습니다”라고 표시되는지 확인합니다.

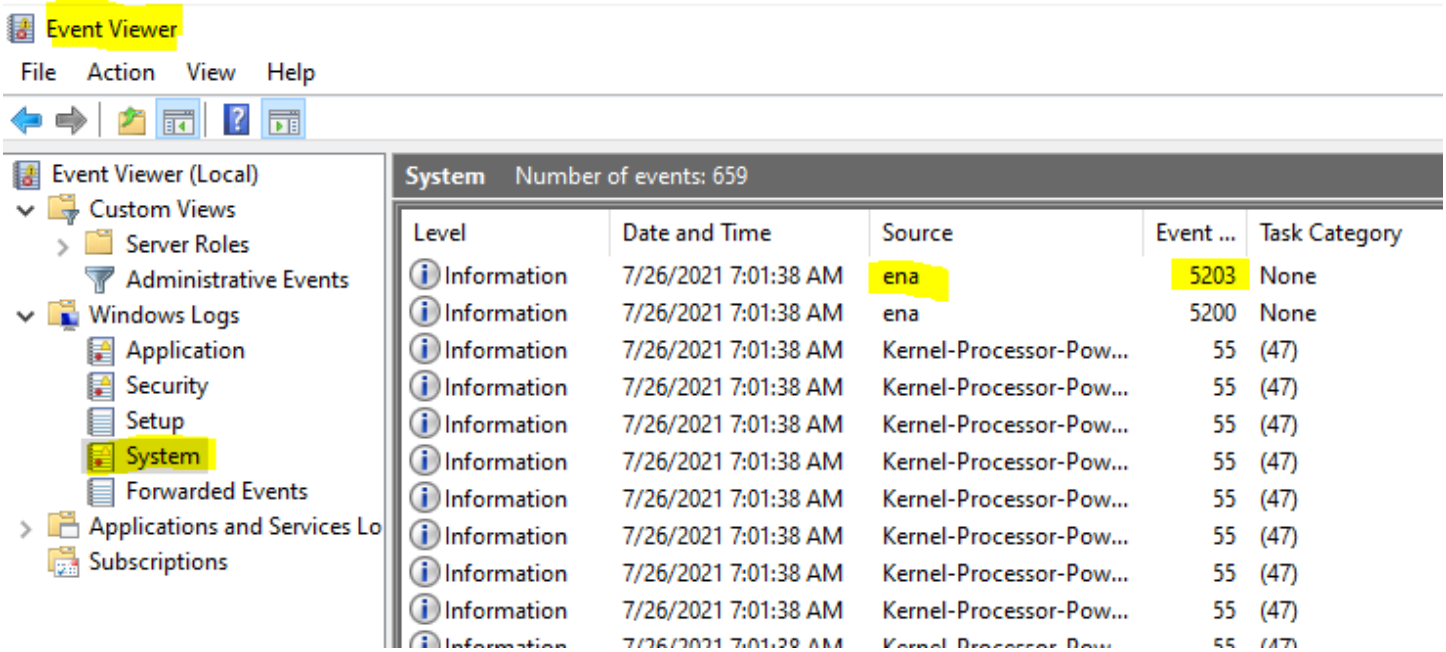
드라이버 이벤트 메시지 조사

Windows 이벤트 뷰어를 사용하여 ENA Windows 드라이버 이벤트 로그를 검토하려면 다음 단계를 따르세요.

1. 이전 섹션에서 설명한 방법 중 하나를 사용하여 실행(Run) 대화 상자를 엽니다.
2. Windows 이벤트 뷰어를 열려면 실행(Run) 상자에 `eventvwr.msc`를 입력합니다.
3. 확인을 선택합니다. 이벤트 뷰어(Event Viewer) 창이 열립니다.
4. Windows 로그(Windows Logs) 메뉴를 확장한 다음 시스템(System)을 선택합니다.

5. 오른쪽 상단 패널의 작업(Actions) 아래에서 현재 로그 필터링(Filter Current Log)을 선택합니다. 필터링 대화 상자가 표시됩니다.
6. 이벤트 원본(Event sources) 상자에 ena를 입력합니다. 이는 결과를 ENA Windows 드라이버에 의해 생성된 이벤트로 제한합니다.
7. 확인을 선택합니다. 창의 세부 정보 섹션에 필터링된 이벤트 로그 결과가 표시됩니다.
8. 세부 정보로 드릴다운하려면 목록에서 이벤트 메시지를 선택합니다.

다음 예제에서는 Windows 이벤트 뷰어 시스템 이벤트 목록의 ENA 드라이버 이벤트를 보여줍니다.



이벤트 메시지 요약

다음 표에는 ENA Windows 드라이버가 생성하는 이벤트 메시지가 나와 있습니다.

입력

이벤트 ID	ENA 드라이버 이벤트 설명	유형
5001	하드웨어에 리소스가 부족합니다.	오류
5002	어댑터가 하드웨어 오류를 감지했습니다.	오류

이벤트 ID	ENA 드라이버 이벤트 설명	유형
5005	적시에 완료되지 않은 NDIS 작업에서 어댑터가 시간 초과되었습니다.	오류
5032	어댑터가 디바이스를 재설정하지 못했습니다.	오류
5200	어댑터가 초기화되었습니다.	정보
5201	어댑터가 중지되었습니다.	정보
5202	어댑터가 일시 중지되었습니다.	정보
5203	어댑터가 다시 시작되었습니다.	정보
5204	어댑터가 종료되었습니다.	정보
5205	어댑터가 재설정되었습니다.	오류
5206	어댑터가 갑자기 제거되었습니다.	오류
5208	어댑터 초기화 루틴이 실패했습니다.	오류
5210	어댑터에 내부 문제가 발생했으며 성공적으로 복구했습니다.	오류

성능 지표 검토

ENA Windows 드라이버는 지표가 활성화된 인스턴스의 네트워크 성능 지표를 게시합니다. 기본 성능 모니터 애플리케이션을 사용하여 인스턴스에 대한 지표를 보고 사용할 수 있습니다. ENA Windows 드라이버가 생성하는 지표에 대한 자세한 내용은 EC2 인스턴스의 ENA 설정에 대한 네트워크 성능 모니터링 섹션을 참조하세요.

ENA 지표가 활성화되고 Amazon CloudWatch 에이전트가 설치된 인스턴스에서 CloudWatch는 Windows 성능 모니터의 카운터와 연결된 지표 및 ENA에 대한 몇 가지 고급 지표를 수집합니다. 이러한 지표가 EC2 인스턴스에서 기본적으로 활성화되어 있는 지표와 함께 수집됩니다. 이러한 지표에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 에이전트가 수집하는 지표를 참조하세요.

> **Note**
>
> 성능 지표는 ENA 드라이버 버전 2.4.0 이상(버전 2.2.3에도 해당)에서 사용할 수 있습니다. ENA 드라이버 버전 2.2.4는 6세대 EC2 인스턴스의 잠재적인 성능 저하로 인해 롤백되었습니다. 최신 업데이트를 받으려면 최신 버전의 드라이버로 업그레이드하는 것이 좋습니다.

다음은 성능 지표를 사용할 수 있는 몇 가지 방법입니다.

- 인스턴스 성능 문제를 해결합니다.
- 워크로드에 적합한 인스턴스 크기를 선택합니다.
- 크기 조정 활동을 사전에 계획합니다.
- 애플리케이션을 벤치마킹하여 인스턴스에서 사용 가능한 성능을 극대화하는지 확인합니다.

새로 고침 속도

기본적으로 드라이버는 1초 간격으로 지표를 새로 고칩니다. 그러나 지표를 검색하는 애플리케이션은 폴링에 다른 간격을 사용할 수 있습니다. 드라이버의 고급 속성을 사용하여 장치 관리자에서 새로 고침 간격을 변경할 수 있습니다.

ENA Windows 드라이버에 대한 지표 새로 고침 간격을 변경하려면 다음 단계를 따르세요.

1. 이전 섹션에서 설명한 방법 중 하나를 사용하여 실행(Run) 대화 상자를 엽니다.
2. Windows 장치 관리자를 열려면 실행(Run) 상자에 `devmgmt.msc`를 입력합니다.
3. 확인을 선택합니다. 그러면 장치 관리자 창이 열립니다.
4. 네트워크 어댑터(Network adapters) 왼쪽의 화살표를 선택하여 목록을 확장합니다.
5. 이름을 선택하거나 Amazon Elastic Network Adapter의 컨텍스트 메뉴를 연 다음 속성(Properties)을 선택합니다. 그러면 Amazon Elastic Network Adapter 속성 대화 상자가 열립니다.
6. 팝업 창에서 고급(Advanced) 탭을 엽니다.
7. (속성)Property 목록에서 지표 새로 고침 간격(Metrics Refresh Interval)을 선택하여 값을 변경합니다.

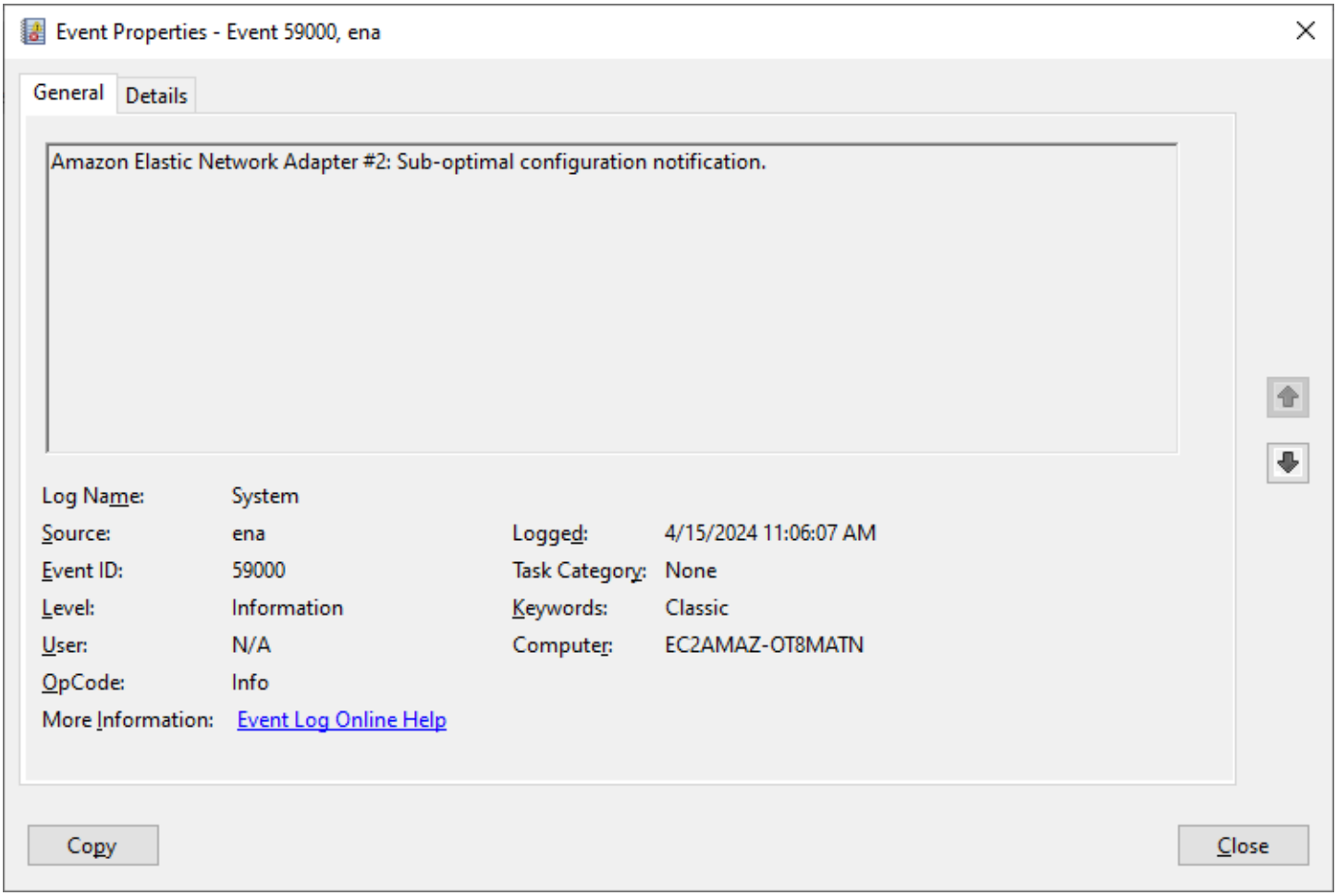
8. 완료했으면 확인(OK)을 선택합니다.

최적이 아닌 구성 알림 조사

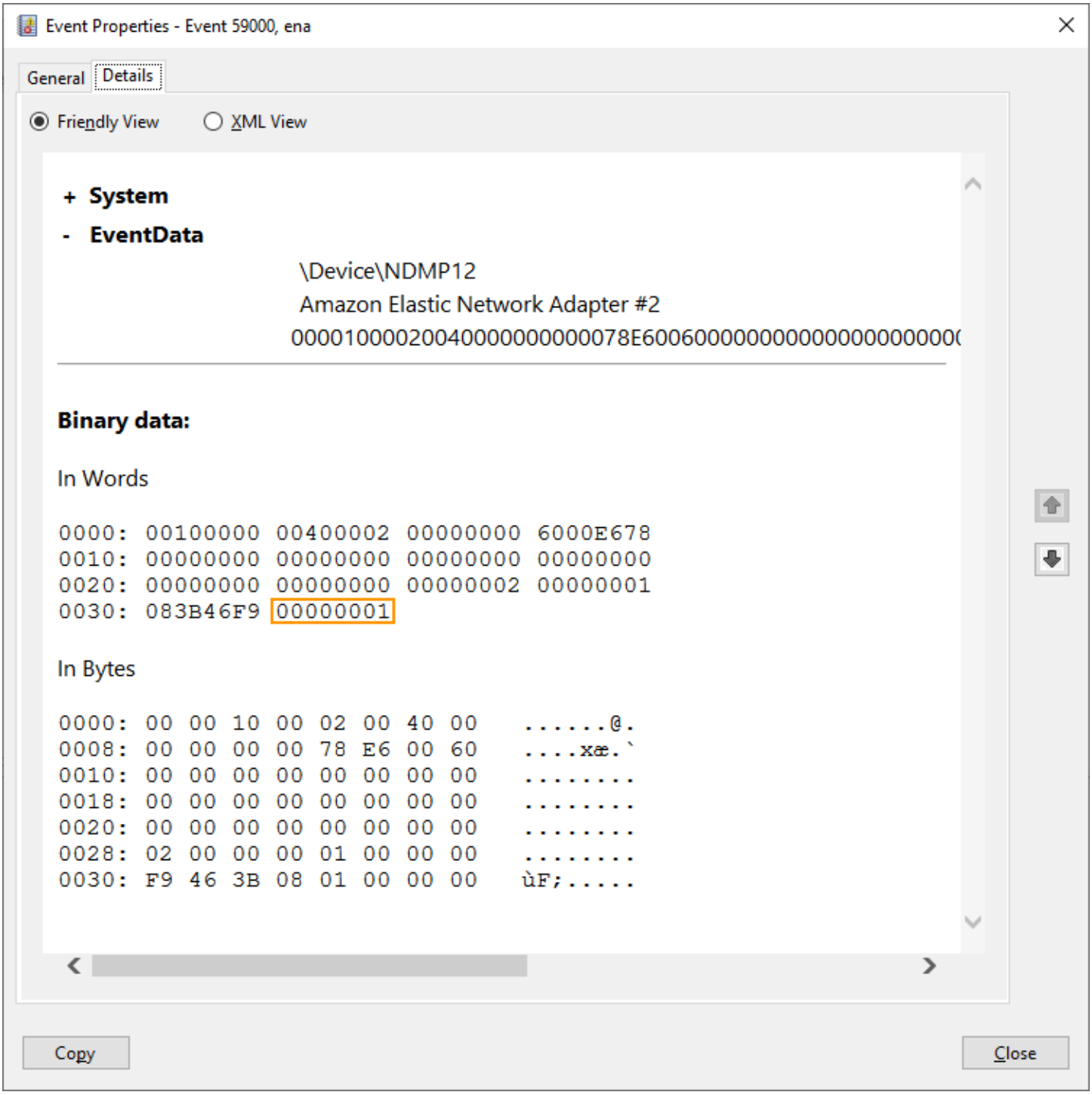
ENA 디바이스는 드라이버에서 변경할 수 있는 최적이 아닌 구성 설정을 탐지합니다. 디바이스는 ENA 드라이버에 알리고 이벤트 알림을 로깅합니다. Windows 이벤트 뷰어에서 최적화되지 않은 이벤트를 검토하려면

1. 이전 섹션에서 설명한 방법 중 하나를 사용하여 실행(Run) 대화 상자를 엽니다.
2. Windows 이벤트 뷰어를 열려면 실행(Run) 상자에 `eventvwr.msc`를 입력합니다.
3. 확인을 선택합니다. 이벤트 뷰어(Event Viewer) 창이 열립니다.
4. Windows 로그(Windows Logs) 메뉴를 확장한 다음 시스템(System)을 선택합니다.
5. 오른쪽 상단 패널의 작업(Actions) 아래에서 현재 로그 필터링(Filter Current Log)을 선택합니다. 필터링 대화 상자가 표시됩니다.
6. 이벤트 원본(Event sources) 상자에 `ena`를 입력합니다. 이는 결과를 ENA Windows 드라이버에 의해 생성된 이벤트로 제한합니다.
7. 확인을 선택합니다. 창의 세부 정보 섹션에 필터링된 이벤트 로그 결과가 표시됩니다.

ID `5900`0인 이벤트는 최적화되지 않은 구성 결과를 알려줍니다. 이벤트를 마우스 오른쪽 버튼으로 클릭하고 이벤트 속성을 선택하여 세부 정보 보기를 열거나 보기 메뉴에서 미리 보기 패널을 선택하여 동일한 세부 정보를 확인합니다.



세부 정보(Details) 탭을 열어 이벤트 코드를 확인합니다. 바이너리 데이터: 단어(Binary Data: In words) 섹션에서 마지막 단어는 코드입니다.



다음 목록은 최적이 아닌 구성 결과에 대한 알림 코드 세부 정보와 권장 조치를 보여줍니다.

- 코드 **1**: 넓은 LLQ 구성을 갖춘 ENA Express는 권장되지 않습니다

ENA Express ENI는 넓은 LLQ로 구성되어 있습니다. 해당 구성은 최적이 아니며 ENA Express의 성능에 영향을 미칠 수 있습니다. ENA Express ENI를 사용할 때는 다음과 같이 넓은 LLQ 설정을 비활성화하는 것이 좋습니다.

1. Windows 장치 관리자를 열려면 실행(Run) 상자에 `devmgmt.msc`를 입력합니다.
2. 확인을 선택합니다. 그러면 장치 관리자 창이 열립니다.
3. 네트워크 어댑터(Network adapters) 왼쪽의 화살표를 선택하여 목록을 확장합니다.
4. `Amazon Elastic Network Adapter`의 디바이스 속성을 엽니다.
5. 여기에서 변경하려면 고급 탭을 엽니다.
6. LLQ 헤더 크기 정책(LLQ Header Size Policy) 속성을 선택하고 값을 `Normal (128 Bytes)`로 설정합니다.
7. 확인을 선택하여 변경 사항을 저장합니다.

- 코드 **2**: Tx 대기열 깊이가 최적이 아닌 ENA Express ENI는 권장되지 않음

  ENA Express ENI가 최적이 아닌 Tx 대기열 깊이로 구성되어 있습니다. 해당 구성은 ENA Express의 성능에 영향을 미칠 수 있습니다. ENA Express ENI를 사용하는 경우 다음과 같이 모든 Tx 대기열을 네트워크 인터페이스의 최댓값으로 확장하는 것이 좋습니다.

  다음 단계에 따라 Tx 대기열을 최대 깊이로 확대합니다.

  1. Windows 장치 관리자를 열려면 실행(Run) 상자에 `devmgmt.msc`를 입력합니다.
  2. 확인을 선택합니다. 그러면 장치 관리자 창이 열립니다.
  3. 네트워크 어댑터(Network adapters) 왼쪽의 화살표를 선택하여 목록을 확장합니다.
  4. `Amazon Elastic Network Adapter`의 디바이스 속성을 엽니다.
  5. 여기에서 변경하려면 고급 탭을 엽니다.
  6. 전송 버퍼(Transmit Buffers) 속성을 선택하고 해당 값을 지원되는 최대값으로 설정합니다.
  7. 확인을 선택하여 변경 사항을 저장합니다.

ENA 어댑터 재설정

ENA Windows 드라이버가 어댑터에서 오류를 감지하고 어댑터를 비정상으로 표시하면 재설정 프로세스가 시작됩니다. 드라이버는 자체적으로 재설정할 수 없으므로 운영 체제에 따라 어댑터 상태를 확인하고 ENA Windows 드라이버에 대한 재설정 핸들을 호출합니다. 재설정 프로세스로 인해 잠시 동안 트래픽 손실이 발생할 수 있습니다. 그러나 TCP 연결은 복구할 수 있어야 합니다.

ENA 어댑터는 연결 유지 알림을 보내지 못해 간접적으로 디바이스 재설정 절차를 요청할 수도 있습니다. 예를 들어 ENA 어댑터가 복구 불가능한 구성을 로드한 후 알 수 없는 상태에 도달하면 연결 유지 알림 전송을 중지할 수 있습니다.

ENA 어댑터 재설정의 일반적인 원인

- 연결 유지 메시지 없음

  ENA 어댑터는 고정된 속도(일반적으로 1초당 한 번)로 연결 유지 이벤트를 게시합니다. ENA Windows 드라이버는 감시 메커니즘을 구현하여 이러한 연결 유지 메시지가 있는지를 주기적으로 확인합니다. 마지막으로 확인한 이후에 하나 이상의 새 메시지를 감지하면 성공적인 결과를 기록합니다. 그렇지 않으면 드라이버는 디바이스에 오류가 발생했다고 결론짓고 재설정 시퀀스를 시작합니다.

- 패킷이 전송 대기열에서 멈춤

  ENA 어댑터는 패킷이 예상대로 전송 대기열을 통해 흐르고 있는지 확인합니다. ENA Windows 드라이버는 패킷이 멈춘 경우 이를 감지하고 재설정 시퀀스를 시작합니다.

- MMIO(Memory Mapped I/O) 레지스터에 대한 읽기 시간 초과

  I/O(MMIO) 읽기 작업을 제한하기 위해 ENA Windows 드라이버는 초기화 및 재설정 프로세스 중에만 MMIO 레지스터에 액세스합니다. 드라이버가 시간 초과를 감지하면 실행 중인 프로세스에 따라 다음 작업 중 하나를 수행합니다.

  - 초기화 중 시간 초과가 감지되면 흐름이 실패하여 드라이버가 Windows 장치 관리자에서 ENA 어댑터 옆에 노란색 느낌표를 표시합니다.
  - 재설정 중 시간 초과가 감지되면 흐름이 실패합니다. 그러면 OS가 ENA 어댑터의 갑작스러운 제거를 시작하고 제거된 어댑터를 중지했다가 시작하여 복구합니다. 네트워크 인터페이스 카드(NIC)의 갑작스러운 제거에 대한 자세한 내용은 Microsoft Windows 하드웨어 개발자 설명서의 NIC의 갑작스러운 제거 처리를 참조하세요.

문제 해결 시나리오

다음 시나리오는 ENA Windows 드라이버에서 발생할 수 있는 문제를 해결하는 데 도움이 될 수 있습니다. 최신 버전이 없는 경우 ENA 드라이버 업그레이드부터 시작하는 것이 좋습니다. 사용 중인 Windows OS 버전에 대한 최신 드라이버를 찾으려면 ENA Windows 드라이버 버전 릴리스 추적 섹션을 참조하세요.

예상치 않은 ENA 드라이버 버전 설치

설명

단계를 수행하여 특정 버전의 ENA 드라이버를 설치하면 Windows 장치 관리자에 Windows에서 다른 버전의 ENA 드라이버를 설치했다고 표시됩니다.

원인

드라이버 패키지 설치를 실행하면 Windows는 시작 전에 로컬 드라이버 저장소의 해당 장치에 유효한 모든 드라이버 패키지의 순위를 매깁니다. 그런 다음 순위 값이 가장 낮은 패키지를 가장 잘 맞는 패키지로 선택합니다. 설치하려는 패키지와 다를 수도 있습니다. 장치 드라이버 패키지 선택 프로세스에 대한 자세한 내용은 Microsoft 설명서 웹 사이트에서 Windows 디바이스에 대한 드라이버 패키지를 선택하는 방법을 참조하세요.

Solution

Windows에서 선택한 드라이버 패키지 버전을 설치하도록 PnPUtil 명령줄 도구를 사용하여 드라이버 저장소에서 순위가 낮은 드라이버 패키지를 제거할 수 있습니다.

ENA 드라이버를 업데이트하려면 다음 단계를 따르세요.

1. 인스턴스 연결 후 로컬 관리자로 로그인합니다.
2. ENA 디바이스 상태 확인 섹션에 설명된 대로 장치 관리자 속성 창을 엽니다. 그러면 Amazon Elastic Network Adapter 속성 창의 일반 탭이 열립니다.
3. 드라이버(Driver) 탭을 엽니다.
4. 드라이버 업데이트(Update Driver)를 선택합니다. 그러면 드라이버 소프트웨어 업데이트 — Amazon Elastic Network Adapter 대화 상자가 열립니다.
    a. 드라이버 소프트웨어는 어떻게 검색합니까? 페이지에서 컴퓨터에서 드라이버 소프트웨어 찾아보기를 선택합니다.
    b. 컴퓨터에서 드라이버 소프트웨어 찾아보기 페이지에서 검색 창 아래에 있는 컴퓨터의 장치 드라이버 목록에서 직접 선택을 선택합니다.
    c. 설치할 하드웨어 장치 드라이버를 선택하십시오 페이지에서 디스크 있음...을 선택합니다.
    d. 디스크에서 설치 창에서 드롭다운 목록의 파일 위치 옆에 있는 찾아보기...를 선택합니다.
    e. 대상 ENA 드라이버 패키지를 다운로드한 위치로 이동합니다. `ena.inf` 파일을 선택하고 열기를 선택합니다.
    f. 설치를 시작하려면 확인을 선택한 후 다음을 선택합니다.

5. 설치 프로그램이 인스턴스를 자동으로 재부팅하지 않는 경우 Restart-Computer PowerShell cmdlet을 실행합니다.

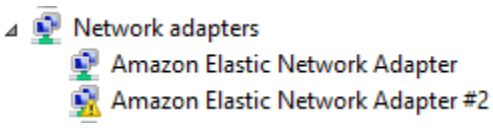
```
PS C:\> Restart-Computer
```

ENA 드라이버에 대한 디바이스 경고

설명

장치 관리자 네트워크 어댑터(Network adapters) 섹션의 ENA 어댑터 아이콘은 경고 기호(내부에 느낌표가 있는 노란색 삼각형)를 표시합니다.

다음 예에서는 Windows 장치 관리자에서 경고 아이콘이 있는 ENA 어댑터를 보여줍니다.



원인

이 디바이스 경고는 대개 더 많은 연구가 필요할 수 있는 환경 문제로 인해 발생하며 근본 원인을 확인하기 위해 제거 프로세스가 필요한 경우가 많습니다. 전체 디바이스 오류 목록은 Microsoft 설명서의 장치 관리자 오류 메시지를 참조하세요.

Solution

이 디바이스 경고의 해결 방법은 근본 원인에 따라 다릅니다. 여기에 설명된 제거 프로세스에는 간단한 방법으로 해결할 수 있는 가장 일반적인 문제를 식별하고 해결하는 데 도움이 되는 몇 가지 기본 단계가 포함되어 있습니다. 이러한 단계로 문제가 해결되지 않으면 추가 근본 원인 분석이 필요합니다.

다음 단계에 따라 일반적인 문제를 식별하고 해결할 수 있습니다.

1. 디바이스 중지 및 시작

   ENA 디바이스 상태 확인 섹션에 설명된 대로 장치 관리자 속성 창을 엽니다. 그러면 장치 상태(Device status)에 오류 코드와 짧은 메시지가 표시되는 Amazon Elastic Network Adapter 속성(Amazon Elastic Network Adapter Properties) 창의 일반(General) 탭이 열립니다.

   a. 드라이버(Driver) 탭을 엽니다.
   b. 장치 사용 안 함(Disable Device)을 선택하고 표시되는 경고 메시지에 예(Yes)라고 응답합니다.

c. 장치 사용(Enable Device)을 선택합니다.

2. EC2 인스턴스 중지 및 시작

   장치 관리자에 여전히 경고 아이콘이 표시되는 경우 다음 단계는 EC2 인스턴스를 중지했다가 시작하는 것입니다. 이렇게 하면 대부분의 경우 다른 하드웨어에서 인스턴스가 다시 시작됩니다.

3. 가능한 인스턴스 리소스 문제 조사

   EC2 인스턴스를 중지했다가 시작했는데 문제가 지속되면 메모리 부족과 같은 인스턴스의 리소스 문제가 있는 것일 수 있습니다.

어댑터 재설정으로 인한 연결 시간 초과(오류 코드 5007, 5205)

설명

Windows 이벤트 뷰어는 ENA 어댑터에 대해 함께 발생하는 어댑터 시간 초과 이벤트와 재설정 이벤트를 보여줍니다. 메시지는 다음 예와 유사합니다.

- 이벤트 ID 5007: Amazon Elastic Network Adapter: 작업 중 시간이 초과되었습니다.
- 이벤트 ID 5205: Amazon Elastic Network Adapter: 어댑터 재설정이 시작되었습니다.

어댑터 재설정은 트래픽 중단을 최소화합니다. 재설정이 여러 번 수행되더라도 심각한 네트워크 중단이 발생하는 경우는 드뭅니다.

원인

이 일련의 이벤트는 ENA Windows 드라이버가 응답하지 않는 ENA 어댑터에 대한 재설정을 시작했음을 나타냅니다. 그러나 장치 드라이버가 이 문제를 감지하는 데 사용하는 메커니즘은 CPU 0 기아로 인한 오탐의 위험이 있습니다.

Solution

이러한 오류 조합이 자주 발생하는 경우 리소스 할당을 확인하여 조정이 도움이 될 수 있는 부분을 확인하세요.

1. 이전 섹션에서 설명한 방법 중 하나를 사용하여 실행(Run) 대화 상자를 엽니다.
2. Windows 리소스 모니터를 열려면 실행(Run) 상자에 `resmon`을 입력합니다.
3. 확인을 선택합니다. 그러면 리소스 모니터 창이 열립니다.
4. CPU 탭을 엽니다. 리소스 모니터 창의 오른쪽에 CPU당 사용량 그래프가 표시됩니다.

5. CPU 0의 사용 수준이 너무 높은지 확인합니다.

더 큰 인스턴스 유형(vCPU 16개 이상)에서 ENA 어댑터에 대해 CPU 0을 제외하도록 RSS를 구성하는 것이 좋습니다. 더 작은 인스턴스 유형의 경우 RSS를 구성하면 환경이 향상될 수 있지만 사용 가능한 코어 수가 적기 때문에 CPU 코어 제한이 성능에 부정적인 영향을 주지 않도록 테스트해야 합니다.

다음 예와 같이 Set-NetAdapterRss 명령을 사용하여 ENA 어댑터에 대한 RSS를 구성합니다.

```
Set-NetAdapterRss -name (Get-NetAdapter | Where-Object {$_.InterfaceDescription -like
 "*Elastic*"}).Name -Baseprocessorgroup 0 -BaseProcessorNumber 1
```

6세대 인스턴스 인프라로 마이그레이션하면 성능 또는 연결에 영향이 있습니다.

설명

6세대 EC2 인스턴스로 마이그레이션하는 경우 ENA Windows 드라이버 버전을 업데이트하지 않으면 성능 저하 또는 ENA 연결 실패가 발생할 수 있습니다.

원인

6세대 EC2 인스턴스 유형에는 인스턴스 운영 체제(OS)에 따라 다음과 같은 최소 버전의 ENA Windows 드라이버가 필요합니다.

최소 버전

Windows Server 버전	ENA 드라이버 버전
Windows Server 2008 R2	2.2.3 또는 2.4.0
Windows Server 2012 이상	2.2.3 이상
Windows 워크스테이션	2.2.3 이상

Solution

6세대 EC2 인스턴스로 업그레이드하기 전에 시작하는 AMI에 이전 표와 같이 인스턴스 OS 기반의 호환 가능한 드라이버가 있는지 확인합니다. 자세한 내용을 알아보려면 AWS re:Post 지식 센터의 최대

네트워크 성능을 얻으려면 EC2 인스턴스를 6세대 인스턴스로 마이그레이션하기 전에 어떤 작업이 필요한가요?를 참조하세요.

탄력적 네트워크 인터페이스에 대한 최적이 아닌 성능

설명

ENA 인터페이스가 예상대로 작동하지 않습니다.

원인

성능 문제에 대한 근본 원인 분석은 제거 프로세스입니다. 일반적인 원인을 설명하기에는 너무 많은 변수가 관련되어 있습니다.

Solution

근본 원인 분석의 첫 번째 단계는 예상대로 수행되지 않는 인스턴스에 대한 진단 정보를 검토하여 문제를 일으킬 수 있는 오류가 있는지 확인하는 것입니다. 자세한 내용은 인스턴스에 대한 진단 정보 수집(을)를 참조하세요.

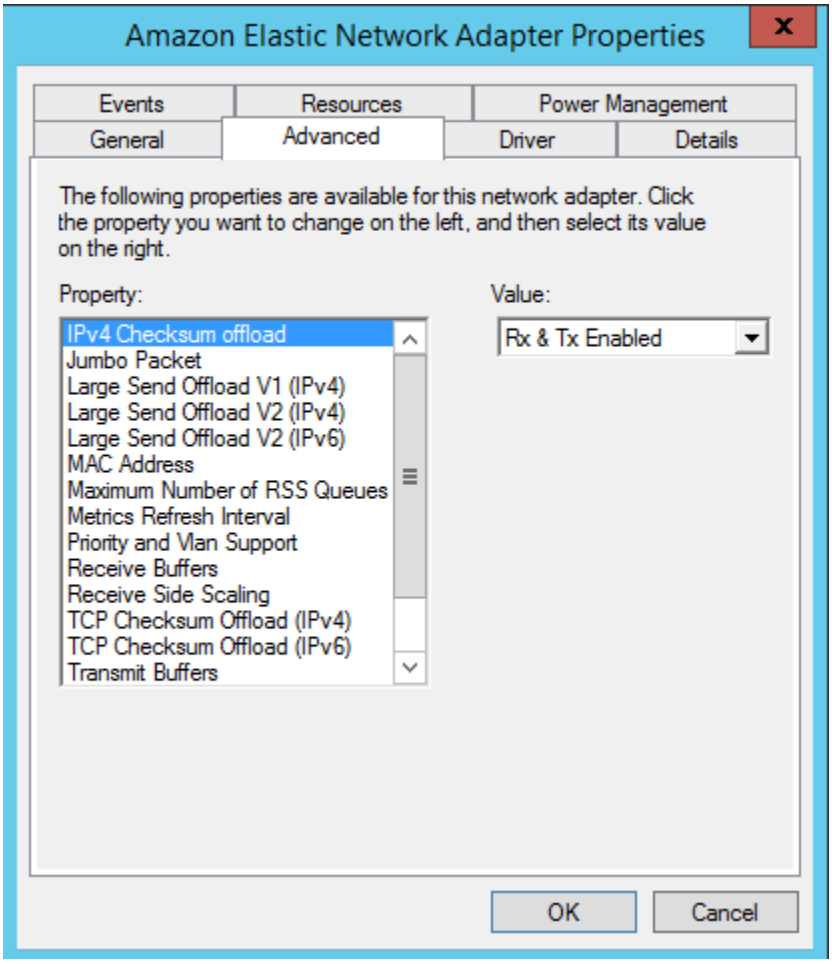
향상된 네트워킹이 있는 인스턴스에서 네트워크 성능을 최대화하려면 기본 운영 체제 구성을 수정해야 할 수 있습니다. 체크섬 오프로딩 켜기 및 RSS 사용과 같은 일부 최적화는 공식 Windows AMI에서 기본적으로 구성됩니다. ENA 어댑터에 적용할 수 있는 기타 최적화는 ENA 어댑터 성능 조정에 표시된 성능 조정을 참조하세요.

주의해서 진행하고 이 섹션에 나열된 항목 또는 AWS Support 팀에서 권장하는 특정 변경 사항으로 디바이스 속성 조정을 제한하는 것이 좋습니다.

ENA 어댑터 속성을 변경하려면 다음 단계를 따르세요.

1. 이전 섹션에서 설명한 방법 중 하나를 사용하여 실행(Run) 대화 상자를 엽니다.
2. Windows 장치 관리자를 열려면 실행(Run) 상자에 `devmgmt.msc`를 입력합니다.
3. 확인을 선택합니다. 그러면 장치 관리자 창이 열립니다.
4. 네트워크 어댑터(Network adapters) 왼쪽의 화살표를 선택하여 목록을 확장합니다.
5. 이름을 선택하거나 Amazon Elastic Network Adapter의 컨텍스트 메뉴를 연 다음 속성(Properties)을 선택합니다. 그러면 Amazon Elastic Network Adapter 속성 대화 상자가 열립니다.
6. 변경하려면 고급 탭을 엽니다.
7. 작업을 마쳤으면 확인을 선택하여 변경 사항을 저장합니다.

다음 예에서는 Windows 장치 관리자에서 ENA 어댑터 속성을 보여줍니다.



ENA 어댑터 성능 조정

다음 표에는 ENA 인터페이스의 성능 향상을 위해 조정할 수 있는 속성이 포함되어 있습니다.

입력

속성	설명	기본값	조절
수신 버퍼(Receive Buffers)	소프트웨어 수신 대기열의 항목 수를 제어합니다.	1024	최대 8,192개까지 늘릴 수 있습니다.
수신측 배율(RSS)(Receive Side Scaling (RSS))	다중 프로세서 시스템의 여러 CPU에 네트워크 수신 처리를 효율적	활성화됨	여러 프로세서에 로드를 분산할 수 있습니다. 자세한 내용은 EC2 Windows 인스턴

속성	설명	기본값	조절
	으로 배포할 수 있습니다.		스에서 네트워크 성능 최적화을 참조하세요.
최대 RSS 대기열 수 (Maximum Number of RSS Queues)	RSS가 활성화된 경우 허용되는 최대 RSS 대기열 수를 설정합니다.	32	RSS 대기열 수는 드라이버 초기화 중 결정되며 다음과 같은 제한 사항을 포함합니다. • 이 속성에 의해 설정된 RSS 대기열 제한 • 인스턴스 제한 (vCPU 수) • 하드웨어 생성 제한 (ENAv1에서 최대 8개의 RSS 대기열, ENAv2에서 최대 32개의 RSS 대기열) 인스턴스 및 하드웨어 생성 제한에 따라 1~32의 값을 설정할 수 있습니다. 자세한 내용은 EC2 Windows 인스턴스에서 네트워크 성능 최적화을 참조하세요.

속성	설명	기본값	조절
Jumbo 패킷(Jumbo packet)	점보 이더넷 프레임(1,500바이트 이상의 페이로드)을 사용할 수 있습니다.	사용 안 함(페이로드를 1500바이트 이하로 제한)	값을 최대 9015로 설정할 수 있으며 이는 9,001바이트의 페이로드로 변환됩니다. 이는 점보 이더넷 프레임의 최대 페이로드입니다. 점보 이더넷 프레임 사용에 대한 고려 사항 섹션을 참조하세요.

점보 이더넷 프레임 사용에 대한 고려 사항

점보 프레임에서는 패킷당 페이로드 크기를 늘려 1,500바이트 이상의 데이터가 허용됩니다. 그 결과, 패킷 오버헤드에 해당하지 않는 패킷의 비율의 늘어납니다. 같은 양의 사용 가능한 데이터를 보내더라도 더 적은 수의 패킷만 있으면 됩니다. 단, 다음과 같은 경우에는 트래픽의 MTU가 최대 1,500으로 제한됩니다.

- EC2 Classic에 대해 지정된 AWS 리전 외부의 트래픽
- 단일 VPC 외부의 트래픽
- 리전 간 VPC 피어링 연결을 통한 트래픽
- VPN 연결을 통한 트래픽
- 인터넷 게이트웨이를 통한 트래픽

> **Note**
>
> 1,500바이트를 초과하는 패킷은 조각화됩니다. IP 헤더에 `Don't Fragment` 플래그가 설정되어 있으면 이러한 패킷은 삭제됩니다.
> 인터넷 트래픽이나 VPC를 벗어나는 트래픽에 점보 프레임을 사용할 때는 주의가 필요합니다. 중간 시스템에서 패킷이 단편화되면서 트래픽이 느려지기 때문입니다. VPC에서 나가는 아웃바운드 트래픽에 영향을 주지 않고 VPC 내부에서 점보 프레임을 사용하려면 다음 옵션 중 하나를 사용해 봅니다.
>
> - 경로별로 MTU 크기를 구성합니다.

- MTU 크기와 경로가 다른 여러 네트워크 인터페이스를 사용합니다.

점보 프레임의 권장 사용 사례

점보 프레임은 VPC 내부 및 VPC 간의 트래픽에 유용할 수 있습니다. 다음과 같은 사용 사례에 점보 프레임을 사용하면 좋습니다.

- 클러스터 배치 그룹 내부에 배치된 인스턴스의 경우 점보 프레임은 가능한 최대 네트워크 처리량을 달성하는 데 도움이 됩니다. 자세한 내용은 Amazon EC2 인스턴스에 대한 배치 그룹 단원을 참조하십시오.
- AWS Direct Connect를 통한 VPC와 온프레미스 네트워크 간의 트래픽에 점보 프레임을 사용할 수 있습니다. AWS Direct Connect 사용 및 점보 프레임 기능 확인에 대한 자세한 내용은 AWS Direct Connect 사용 설명서의 프라이빗 가상 인터페이스 또는 전송 가상 인터페이스에 대한 MTU을 참조하세요.
- 전송 게이트웨이에 대해 지원되는 MTU 크기에 대한 자세한 내용은 Amazon VPC Transit Gateway의 Transit Gateway에 대한 할당량을 참조하세요.

Linux 기반 EC2 인스턴스의 네트워크 지연 시간 개선

네트워크 지연 시간은 데이터 패킷이 소스에서 대상으로 전송되는 데 걸리는 시간입니다. 네트워크를 통해 데이터를 전송하는 애플리케이션에서 긍정적인 사용자 경험을 제공하려면 적시에 응답해야 합니다. 네트워크 지연 시간이 길면 다음과 같은 다양한 문제가 발생할 수 있습니다.

- 느린 웹 페이지 로드 시간
- 비디오 스트리밍 지연
- 온라인 리소스에 액세스하기가 어려움

이 섹션에서는 Linux에서 실행되는 Amazon EC2 인스턴스의 네트워크 지연 시간을 개선하기 위해 취할 수 있는 단계를 간략하게 설명합니다. 최적의 지연 시간을 실현하려면 다음에서 설명하는 단계에 따라 인스턴스, 커널 및 ENA 드라이버 설정을 구성하세요. 추가 구성 지침은 GitHub의 ENA Linux Driver Best Practices and Performance Optimization Guide(ENA Linux 드라이버 모범 사례 및 성능 최적화 가이드)를 참조하세요.

> **Note**
>
> 단계와 설정은 특정 네트워크 하드웨어, 인스턴스를 시작한 AMI, 애플리케이션 사용 사례에 따라 조금씩 다를 수 있습니다. 변경하기 전에 네트워크 성능을 철저히 테스트하고 모니터링하여 원하는 결과를 얻을 수 있는지 확인하세요.

데이터 패킷의 네트워크 홉 수 감소

데이터 패킷이 라우터 간을 이동할 때 거치는 각 홉은 네트워크 지연 시간을 늘립니다. 일반적으로 트래픽은 여러 홉을 거쳐 대상에 도달합니다. Amazon EC2 인스턴스의 네트워크 홉을 줄이는 방법은 다음 두 가지가 있습니다.

- 클러스터 배치 그룹 - 클러스터 배치 그룹을 지정하면 Amazon EC2가 더 엄격한 패킹을 통해 물리적으로 동일한 가용 영역(AZ) 내에서 서로 가까이에 있는 인스턴스를 시작합니다. 그룹 내 인스턴스가 물리적으로 가깝기 때문에 고속 연결을 활용할 수 있고, 지연 시간이 짧아지면서 단일 흐름 처리량이 높아집니다.
- 전용 호스트 – 전용 호스트는 고객 전용 물리적 서버입니다. 전용 호스트를 사용하면 여러 인스턴스를 동일한 물리적 서버에서 실행할 수 있습니다. 동일한 전용 호스트에서 실행되는 인스턴스 간의 통신은 추가 네트워크 홉 없이 이루어질 수 있습니다.

Linux 커널 구성이 지연 시간에 미치는 영향

Linux 커널 구성은 네트워크 지연 시간을 늘리거나 줄일 수 있습니다. 지연 시간 최적화 목표를 달성하려면 워크로드의 특정 요구 사항에 따라 Linux 커널 구성을 미세 조정하는 것이 중요합니다.

Linux 커널에는 네트워크 지연 시간을 줄이는 데 도움이 될 만한 많은 구성 옵션이 있습니다. 가장 큰 영향을 미치는 옵션은 다음과 같습니다.

- 사용 중 폴링 모드 활성화 - 사용 중 폴링 모드는 네트워크 수신 경로의 지연 시간을 줄입니다. 사용 중 폴링 모드를 활성화하면 소켓 계층 코드가 네트워크 디바이스의 수신 대기열을 직접 폴링할 수 있습니다. 비지 폴링의 단점은 긴밀한 루프에서 새 데이터를 폴링할 때 호스트의 CPU 사용량이 증가한다는 것입니다. 모든 인터페이스의 패킷 대기 시간을 마이크로초 단위로 제어하는 두 가지 글로벌 설정이 있습니다.

`busy_read`

소켓 읽기의 지연 시간이 짧은 사용 중 폴링 제한 시간입니다. 이는 소켓 계층이 디바이스 대기열의 패킷을 읽을 때까지 대기하는 시간을 마이크로초 단위로 제어합니다. sysctl 명령을 사용하여 이 기능을 전역적으로 활성화하려면 Linux Kernel 조직에서 50마이크로초의 값을 사용하는 것이 좋습니다. 자세한 내용은 Linux 커널 사용자 및 관리자 안내서의 busy_read를 참조하세요.

```
[ec2-user ~]$ sudo sysctl -w net.core.busy_read=50
```

`busy_poll`

폴링 및 선택의 지연 시간이 짧은 사용 중 폴링 제한 시간입니다. 이는 이벤트를 대기하는 시간을 마이크로초 단위로 제어합니다. 권장 값은 50~100마이크로초이며 폴링하는 소켓 수에 따라 다릅니다. 소켓을 더 많이 추가하면 시간이 증가합니다.

```
[ec2-user ~]$ sudo sysctl -w net.core.busy_poll=50
```

- CPU 성능 상태(C 상태) 구성 - C 상태는 비활성 상태일 때 코어가 전환되는 절전 수준을 제어합니다. C 상태를 제어하여 시스템의 지연 시간과 성능 조합을 미세 조정할 수 있습니다. C 상태 심화에서 CPU는 기본적으로 '비활동 상태'이며 다시 활성 상태로 전환될 때까지 요청에 응답할 수 없습니다. 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 중단을 처리하는 코어가 비활동 상태인 경우 인터럽트 처리가 지연될 수 있습니다. C 상태 심화를 사용하지 않도록 시스템을 구성할 수 있습니다. 이 구성을 사용하여 프로세서 반응 지연 시간을 줄일 수 있지만, Turbo Boost를 위해 다른 코어에서 사용할 수 있는 여유 용량이 줄어듭니다.

  C 상태 심화를 제한하면 프로세서 반응 지연 시간을 줄일 수 있습니다. 자세한 내용은 Amazon Linux 2 사용 설명서의 C 상태 심화 제한을 통한 고성능 및 저지연 시간을 참조하세요.

인터럽트 조정

ENA 네트워크 드라이버는 인스턴스와 네트워크 간의 통신을 가능케 합니다. 이 드라이버는 네트워크 패킷을 처리하고 네트워크 스택이나 Nitro Card로 전달합니다. 네트워크 패킷이 수신되면 CPU가 소프트웨어에 이벤트에 대해 알릴 수 있도록 Nitro Card가 인터럽트를 생성합니다.

인터럽트

인터럽트는 디바이스 또는 애플리케이션이 프로세서로 보내는 신호입니다. 인터럽트는 프로세서에 이벤트가 발생했거나 즉각적인 주의가 필요한 조건에 해당함을 알립니다. 인터럽트는 네트워크 인터페이스에서의 데이터 수신, 하드웨어 이벤트 처리, 다른 디바이스의 요청 처리 등 시간에 민감한 작업을 처리할 수 있습니다.

인터럽트 조정

인터럽트 조정은 인터럽트를 집계하거나 지연시켜 디바이스에서 생성하는 인터럽트의 수를 줄이는 기술입니다. 인터럽트 조정의 목적은 대량의 인터럽트 처리에 따른 오버헤드를 줄여 시스템 성능을 개선하는 것입니다. 인터럽트가 너무 많으면 CPU 사용량이 증가하여 처리량에 부정적인 영향을 미치고, 인터럽트가 너무 적으면 지연 시간이 증가합니다.

동적 인터럽트 조정

동적 인터럽트 조정은 현재 시스템 부하 및 트래픽 패턴에 따라 인터럽트 비율을 동적으로 조정하는 개선된 형태의 인터럽트 조정입니다. 인터럽트 오버헤드와 초당 패킷 수 또는 대역폭 감소 사이의 균형을 맞추는 것을 목표로 합니다.

> **Note**
>
> 동적 인터럽트 조정은 일부 AMI에서 기본적으로 활성화되어 있으며, 모든 AMI에서 활성화 또는 비활성화할 수 있습니다.

네트워크 지연 시간을 최소화하려면 인터럽트 조정을 비활성화해야 할 수 있습니다. 하지만 이 경우 인터럽트 처리의 오버헤드가 증가할 수도 있습니다. 지연 시간을 줄이는 것과 오버헤드를 최소화하는 것 사이에서 적절한 균형을 찾는 것이 중요합니다. `ethtool` 명령은 인터럽트 조정을 구성하는 데 도움이 될 수 있습니다. 기본적으로 `rx-usecs`는 `20`으로 설정되고 `tx-usecs`는 `64`로 설정됩니다.

현재 인터럽트 수정 구성을 가져오려면 다음 명령을 사용합니다.

```
[ec2-user ~]$ ethtool -c interface | egrep "rx-usecs:|tx-usecs:|Adaptive RX"
Adaptive RX: on  TX: off
rx-usecs: 20
tx-usecs: 64
```

인터럽트 조정 및 동적 인터럽트 조정을 비활성화하려면 다음 명령을 사용합니다.

```
[ec2-user ~]$ sudo ethtool -C interface adaptive-rx off rx-usecs 0 tx-usecs 0
```

성능 튜닝을 위한 Nitro 시스템 고려 사항

Nitro 시스템은 우수한 성능과 고가용성, 철저한 보안을 지원하기 위해 AWS가 구축한 하드웨어 및 소프트웨어 구성 요소의 모음입니다. Nitro System은 가상화 오버헤드를 없애고 호스트 하드웨어에 대한 모든 액세스 권한이 필요한 워크로드를 지원하는 베어 메탈과 유사한 기능을 제공합니다. 자세한 내용은 AWS Nitro 시스템을 참조하세요.

현재 세대의 모든 EC2 인스턴스 유형은 EC2 Nitro Card에서 네트워크 패킷 처리를 수행합니다. 이 주제에서는 Nitro Card에서 높은 수준의 패킷 처리, 패킷 처리 성능에 영향을 미치는 네트워크 아키텍처 및 구성의 일반적인 측면, Nitro 기반 인스턴스의 최대 성능을 얻기 위해 취할 수 있는 조치에 대해 다룹니다.

Nitro Card는 Virtual Private Cloud(VPC)에 필요한 인터페이스와 같은 모든 입출력(I/O) 인터페이스를 처리합니다. 네트워크를 통해 정보를 송수신하는 모든 구성 요소에서 Nitro Card는 고객 워크로드가 실행되는 시스템 메인 보드와 물리적으로 분리된 I/O 트래픽을 위한 독립형 컴퓨팅 디바이스 역할을 합니다.

Nitro Card의 네트워크 패킷 흐름

Nitro 시스템에 구축된 EC2 인스턴스에는 초당 패킷 수(PPS) 처리량 속도로 측정할 때 더 빠른 패킷 처리를 지원하는 하드웨어 가속 기능이 있습니다. Nitro Card가 새 흐름에 대한 초기 평가를 수행하면 보안 그룹, 액세스 제어 목록, 라우팅 테이블 항목 등 흐름의 모든 패킷에 대해 동일한 정보가 저장됩니다. 동일한 흐름에 대해 추가 패킷을 처리할 때 저장된 정보를 사용하여 해당 패킷의 오버헤드를 줄일 수 있습니다.

연결 속도는 초당 연결 수(CPS) 지표로 측정됩니다. 새로 연결할 때마다 추가 처리 오버헤드가 필요하며, 이 오버헤드는 워크로드 용량 추정에 포함되어야 합니다. 워크로드를 설계할 때는 CPS와 PPS 지표를 모두 고려하는 것이 중요합니다.

연결 설정 방법

Nitro 기반 인스턴스와 다른 엔드포인트 간에 연결이 설정되면 Nitro Card는 두 엔드포인트 간에 전송 또는 수신되는 첫 번째 패킷의 전체 흐름을 평가합니다. 동일한 흐름의 후속 패킷의 경우 일반적으로 전체 재평가가 필요하지 않습니다. 하지만 예외는 있습니다. 예외에 대한 자세한 내용은 하드웨어 가속을 사용하지 않는 패킷 섹션을 참조하세요.

다음 속성은 두 엔드포인트와 두 엔드포인트 간 패킷 흐름을 정의합니다. 이 다섯 가지 속성을 합쳐서 5튜플 흐름이라고 합니다.

- 소스 IP

- 원본 포트
- 목적지 IP
- 대상 포트
- 통신 프로토콜

패킷 흐름의 방향을 수신(인바운드) 및 송신(아웃바운드)이라고 합니다. 다음은 포괄적인 네트워크 패킷 흐름을 요약한 개괄적인 설명입니다.

- 수신 - Nitro Card가 인바운드 네트워크 패킷을 처리할 때 상태 저장 방화벽 규칙 및 액세스 제어 목록을 기준으로 패킷을 평가합니다. 연결을 추적하고, 측정하며, 필요에 따라 기타 작업을 수행합니다. 그런 다음, 패킷을 호스트 CPU의 대상으로 전달합니다.
- 송신 - Nitro Card가 아웃바운드 네트워크 패킷을 처리할 때 원격 인터페이스 대상을 찾고, 다양한 VPC 기능을 평가하au, 속도 제한을 적용하고, 적용되는 기타 작업을 수행합니다. 그런 다음, 패킷을 네트워크의 다음 홉 대상으로 전달합니다.

최적의 성능을 위한 네트워크 설계

Nitro 시스템의 성능 기능을 활용하려면 네트워크 처리 요구 사항을 이해하고, 이러한 요구 사항이 Nitro 리소스의 워크로드에 어떤 영향을 미치는지 이해해야 합니다. 그러면 네트워크 환경에 맞는 최적의 성능을 지원하도록 설계할 수 있습니다. 인프라 설정 및 애플리케이션 워크로드 설계와 구성은 패킷 처리 및 연결 속도 모두에 영향을 미칠 수 있습니다. 예를 들어 DNS 서비스, 방화벽 또는 가상 라우터와 같이 애플리케이션의 연결 설정 비율이 높으면 연결이 설정된 후에만 적용되는 하드웨어 가속을 활용할 기회가 줄어듭니다.

워크로드를 간소화하고 네트워크 성능을 개선하도록 애플리케이션 및 인프라 설정을 구성할 수 있습니다. 그러나 모든 패킷이 가속에 적합한 것은 아닙니다. Nitro 시스템은 새 연결과 가속에 적합하지 않은 패킷에 대해 전체 네트워크 흐름을 사용합니다.

이 섹션의 나머지 부분에서는 패킷 흐름이 최대한 가속 경로 내에서 진행되도록 보장하는 애플리케이션 및 인프라 설계 고려 사항에 초점을 맞춥니다.

Nitro 시스템의 네트워크 설계 고려 사항

인스턴스의 네트워크 트래픽을 구성할 때는 PPS 성능에 영향을 미칠 수 있는 여러 측면을 고려해야 합니다. 흐름이 설정된 후에는 정기적으로 수신되거나 송신되는 대부분의 패킷이 가속에 적합할 수 있습니다. 그러나 인프라 설계 및 패킷 흐름이 프로토콜 표준을 계속 충족하도록 하기 위한 예외가 존재합니다.

Nitro Card를 최대한 활용하려면 인프라 및 애플리케이션에 대한 다음 구성 세부 정보의 장단점을 신중하게 고려해야 합니다.

인프라 고려 사항

인프라 구성은 패킷 흐름과 처리 효율성에 영향을 미칠 수 있습니다. 다음 목록에는 중요한 몇 가지 고려 사항이 나와 있습니다.

비대칭성이 있는 네트워크 인터페이스 구성

보안 그룹은 연결 추적을 사용해 인스턴스에서 송수신되는 트래픽에 대한 정보를 추적합니다. 트래픽이 한 네트워크 인터페이스를 통해 인스턴스로 수신되고 다른 네트워크 인터페이스를 통해 송신되는 비대칭 라우팅을 사용하면 흐름을 추적하는 경우 인스턴스에서 달성할 수 있는 최고 성능을 줄일 수 있습니다. 보안 그룹 연결 추적, 추적되지 않은 연결 및 자동 추적된 연결에 대한 자세한 내용은 Amazon EC2 보안 그룹 연결 추적 섹션을 참조하세요.

네트워크 드라이버

네트워크 드라이버는 정기적으로 업데이트 및 릴리스됩니다. 오래된 드라이버인 경우 성능이 크게 저하될 수 있습니다. 최신 패치를 사용하고 최신 세대 드라이버에서만 사용할 수 있는 가속 경로 기능과 같은 성능 개선 기능을 활용할 수 있도록 드라이버를 최신 상태로 유지합니다. 이전 드라이버는 가속 경로 기능을 지원하지 않습니다.

가속 경로 기능을 활용하려면 인스턴스에 최신 ENA 드라이버를 설치하는 것이 좋습니다.

Linux 인스턴스 - ENA Linux 드라이버 2.2.9 이상. Amazon 드라이버 GitHub 리포지토리에서 ENA Linux 드라이버를 설치하거나 업데이트하려면 readme 파일의 Driver compilation 섹션을 참조하세요.

Windows 인스턴스 - ENA Windows 드라이버 2.0.0 이상. ENA Windows 드라이버를 설치하거나 업데이트하려면 EC2 Windows 인스턴스에 ENA 드라이버 설치 섹션을 참조하세요.

엔드포인트 간 거리

동일한 가용 영역에 있는 두 인스턴스 간의 연결은 애플리케이션 계층에서 TCP 윈도우 기능(지정된 시점에 전송할 수 있는 데이터의 양을 결정함)의 결과로 리전 간 연결보다 초당 더 많은 패킷을 처리할 수 있습니다. 인스턴스 간 거리가 멀면 지연 시간이 늘어나고 엔드포인트에서 처리할 수 있는 패킷 수가 줄어듭니다.

바이트 대기열 제한(BQL)

BQL은 대기열을 줄이기 위해 Nitro Card에 전달되는 바이트 수를 제한하는 기능입니다. BQL은 ENA 드라이버, Amazon Linux 운영 체제 및 대부분의 Linux 배포판에서 기본적으로 비활성화되어

있습니다. BQL과 조각 프록시 재정의를 모두 활성화하면 모든 조각이 처리되기 전에 Nitro에 전달되는 바이트 수가 제한되어 성능 제한이 발생할 수 있습니다.

애플리케이션 설계 고려 사항

애플리케이션 설계 및 구성에는 처리 효율성에 영향을 줄 수 있는 여러 측면이 있습니다. 다음 목록에는 중요한 몇 가지 고려 사항이 나와 있습니다.

패킷 크기

패킷 크기가 클수록 인스턴스가 네트워크에서 송수신할 수 있는 데이터의 처리량이 증가할 수 있습니다. Amazon EC2는 9001바이트의 점보 프레임을 지원하지만 다른 서비스에서는 다른 제한을 적용할 수 있습니다. 패킷 크기가 작을수록 패킷 처리 속도가 증가할 수 있지만 패킷 수가 PPS 허용치를 초과할 경우 얻을 수 있는 최대 대역폭이 줄어들 수 있습니다.

패킷 크기가 네트워크 홉의 최대 전송 단위(MTU)를 초과하는 경우 경로를 따라 라우터에서 패킷을 조각화할 수 있습니다. 그 결과 패킷 조각은 예외로 간주되며, 일반적으로 표준 속도(가속화되지 않음)로 처리됩니다. 이로 인해 성능이 달라질 수 있습니다. 그러나 조각 프록시 모드 설정을 사용하여 아웃바운드 조각화된 패킷의 표준 동작을 재정의할 수 있습니다. 자세한 내용은 Nitro 시스템의 네트워크 성능 극대화 섹션을 참조하세요. MTU를 구성할 때 토폴로지를 평가하는 것이 좋습니다.

프로토콜 절충

TCP와 같은 신뢰할 수 있는 프로토콜은 UDP와 같은 신뢰할 수 없는 프로토콜보다 오버헤드가 더 큽니다. UDP 전송 프로토콜은 오버헤드가 낮고 네트워크 처리가 단순화되므로 PPS 속도는 높지만 패킷 전달의 안정성이 떨어질 수 있습니다. 애플리케이션에 안정적인 패킷 전송이 중요하지 않은 경우 UDP를 사용하는 것이 좋습니다.

마이크로 버스팅

마이크로 버스팅은 트래픽이 균등하게 분배되지 않고 짧은 시간 동안 허용량을 초과할 때 발생합니다. 일반적으로 마이크로초 단위로 발생합니다.

예를 들어 인스턴스에서 최대 10Gbps를 전송할 수 있고, 애플리케이션이 0.5초 안에 10Gb 전체를 전송한다고 가정합니다. 이 마이크로 버스팅은 처음 0.5초 동안 허용량을 초과하고 나머지 0.5초 동안 아무 것도 남기지 않습니다. 1초 동안 10Gb를 전송했지만 처음 0.5초의 허용량 때문에 패킷이 삭제되거나 대기 상태가 될 수 있습니다.

Linux Traffic Control과 같은 네트워크 스케줄러를 사용하면 처리량을 조절하고 마이크로 버스팅으로 인해 패킷이 대기 상태가 되거나 삭제되는 것을 방지할 수 있습니다.

흐름 수

단일 흐름은 최대 10Gbps를 지원하는 클러스터 배치 그룹 내에 있지 않거나 최대 25Gbps를 지원하는 ENA Express를 사용하는 경우가 아니면 5Gbps로 제한됩니다.

마찬가지로 Nitro Card는 단일 흐름을 사용하는 것보다 여러 흐름에서 더 많은 패킷을 처리할 수 있습니다. 인스턴스당 최대 패킷 처리 속도를 달성하려면 총 대역폭이 100Gbps 이상인 인스턴스에서 최소 100개의 흐름을 사용하는 것이 좋습니다. 총 대역폭 용량이 증가하면 최대 처리 속도를 달성하는 데 필요한 흐름 수도 증가합니다. 벤치마킹은 네트워크에서 최고 속도를 달성하는 데 필요한 구성을 결정하는 데 도움이 됩니다.

Elastic Network Adapter(ENA) 대기열

ENA(Elastic Network Adapter)는 여러 수신(Rx) 및 전송(Tx) 대기열(ENA 대기열)을 사용하여 EC2 인스턴스의 네트워크 성능과 확장성을 개선합니다. 이러한 대기열은 사용할 수 있는 대기열 사이에 송수신된 데이터를 로드 밸런싱하여 네트워크 트래픽을 효율적으로 관리합니다.

자세한 내용은 the section called “ENA 대기열” 섹션을 참조하세요.

특성 처리 오버헤드

트래픽 미러링 및 ENA Express와 같은 특성은 처리 오버헤드를 증가시켜 절대 패킷 처리 성능을 감소시킬 수 있습니다. 특성 사용을 제한하거나 특성을 비활성화하여 패킷 처리 속도를 높일 수 있습니다.

연결 추적으로 상태 유지

보안 그룹은 연결 추적을 사용해 인스턴스에서 송수신되는 트래픽에 대한 정보를 저장합니다. 연결 추적은 네트워크 트래픽의 개별 흐름에 규칙을 적용하여 해당 트래픽의 허용 또는 거부를 결정합니다. Nitro Card는 흐름 추적을 사용하여 흐름의 상태를 유지 관리합니다. 더 많은 보안 그룹 규칙이 적용되면 흐름을 평가하는 데 더 많은 작업이 필요합니다.

> **Note**
>
> 모든 트래픽 흐름을 추적하지는 않습니다. 추적되지 않는 연결에서 보안 그룹 규칙이 구성된 경우 유효한 응답 경로가 여러 개 있을 때 대칭 라우팅을 보장하기 위해 자동으로 추적되는 연결을 제외하고는 추가 작업이 필요하지 않습니다.

하드웨어 가속을 사용하지 않는 패킷

모든 패킷이 하드웨어 가속을 이용할 수 있는 것은 아닙니다. 이러한 예외를 처리하려면 네트워크 흐름의 상태를 보장하는 데 필요한 일부 처리 오버헤드가 수반됩니다. 네트워크 흐름은 프로토콜 표준을 안

정적으로 충족하고, VPC 설계의 변경 사항을 준수하며, 허용된 대상으로만 패킷을 라우팅해야 합니다. 하지만 오버헤드로 인해 성능이 저하됩니다.

패킷 조각

애플리케이션 고려 사항에서 언급한 바와 같이, 네트워크 MTU를 초과하는 패킷으로 인해 발생하는 패킷 조각은 일반적으로 예외로 처리되며 하드웨어 가속을 활용할 수 없습니다. 하지만 드라이버 버전에 따라 조각 프록시 모드를 사용하여 송신 조각 제한을 우회할 수 있습니다. 자세한 내용은 Nitro 시스템의 네트워크 성능 극대화 섹션에서 수행할 수 있는 작업을 참조하세요.

유휴 연결

연결이 제한 시간에 도달하지 않았더라도 한동안 연결에 활동이 없으면 시스템에서 우선순위를 낮출 수 있습니다. 연결 우선순위가 낮아진 후에 데이터가 수신된 경우 다시 연결하려면 시스템에서 이를 예외로 처리해야 합니다.

연결을 관리하기 위해 연결 추적 제한 시간을 사용하여 유휴 연결을 종료할 수 있습니다. 또한 TCP Keepalive를 사용하여 유휴 연결을 열린 상태로 유지할 수 있습니다. 자세한 내용은 유휴 연결 추적 제한 시간 섹션을 참조하세요.

VPC 변형

보안 그룹, 라우팅 테이블, 액세스 제어 목록에 대한 모든 업데이트를 처리 경로에서 재평가하여 라우팅 항목과 보안 그룹 규칙이 여전히 예상대로 적용되는지 확인해야 합니다.

ICMP 플로우

ICMP(인터넷 제어 메시지 프로토콜)는 네트워크 장치가 네트워크 통신 문제를 진단하는 데 사용하는 네트워크 계층 프로토콜입니다. 이러한 패킷은 항상 전체 흐름을 사용합니다.

비대칭 L2 흐름

NitroV3 및 이전 플랫폼은 동일한 서브넷에서 한 ENI는 기본 게이트웨이 라우터를 사용하고 다른 ENI는 기본 게이트웨이 라우터를 사용하지 않는 경우 두 ENI 간의 트래픽에 하드웨어 가속화를 사용하지 않습니다. 이 시나리오에서 NitroV4 이후 플랫폼은 하드웨어 가속화를 활용합니다. NitroV3 또는 이전 플랫폼에서 성능을 향상시키려면 사용되는 기본 게이트웨이 라우터가 두 ENI 간에 일치하는지 또는 해당 ENI가 서로 다른 서브넷에 있는지 확인합니다.

Nitro 시스템의 네트워크 성능 극대화

네트워크 설정을 조정하여 Nitro 시스템에서 네트워크 성능을 극대화할 수 있습니다.

주제

- 고려 사항
- 튜닝 PPS 성능
- Linux 인스턴스에서 성능 모니터링
- ENA 대기열

고려 사항

최상의 결과를 얻으려면 설계 결정을 내리거나 인스턴스의 네트워크 설정을 조정하기 전에 다음 단계를 수행하는 것이 좋습니다.

1. Nitro 시스템의 네트워크 설계 고려 사항 검토를 통해 성능 개선을 위해 수행할 수 있는 작업의 장단점을 이해합니다.

   Linux에서 인스턴스 구성을 위한 추가 고려 사항과 모범 사례는 GitHub의 ENA Linux Driver Best Practices and Performance Optimization Guide를 참조하세요.
2. 최대 활성 흐름 수로 워크로드를 벤치마킹하여 애플리케이션 성능의 기준선을 결정합니다. 성능 기준선을 통해 특히 스케일 업 또는 스케일 아웃을 계획하는 경우 설정 또는 애플리케이션 설계의 변화를 테스트하여 어떤 고려 사항이 가장 큰 영향을 미치는지 파악할 수 있습니다.

튜닝 PPS 성능

다음 목록에는 시스템 요구 사항에 따라 PPS 성능을 조정하기 위해 수행할 수 있는 작업이 포함되어 있습니다.

- 두 인스턴스 간 물리적 거리를 줄입니다. 송신 및 수신 인스턴스가 동일한 가용 영역에 있거나 클러스터 배치 그룹을 사용하는 경우 패킷이 한 엔드포인트에서 다른 엔드포인트로 이동하는 데 필요한 홉 수를 줄일 수 있습니다.
- 추적되지 않는 연결를 사용합니다.
- 네트워크 트래픽에 UDP 프로토콜을 사용합니다.
- 총 대역폭이 100Gbps 이상인 EC2 인스턴스의 경우 100개 이상의 개별 흐름에 워크로드를 분산하여 Nitro Card 전체에 작업을 고르게 분산합니다.
- EC2 인스턴스에서 송신 조각 PPS 제한을 극복하기 위해 조각 프록시 모드를 활성화할 수 있습니다(드라이버 버전에 따라 다름). 이 설정을 사용하면 조각화된 패킷을 처리 경로에서 평가할 수 있으므로 1024의 송신 PPS 제한을 극복할 수 있습니다. 드라이버를 로드할 때 다음 명령 중 하나를 실행하여 조각 프록시 모드를 활성화하거나 비활성화합니다.

조각 프록시 모드 활성화

```
sudo insmod ena.ko enable_frag_bypass=1
```

조각 프록시 모드 비활성화

```
sudo insmod ena.ko enable_frag_bypass=0
```

Linux 인스턴스에서 성능 모니터링

Linux 인스턴스에서 Ethtool 지표를 사용하여 대역폭, 패킷 속도, 연결 추적과 같은 인스턴스 네트워킹 성능 지표를 모니터링할 수 있습니다. 자세한 내용은 EC2 인스턴스의 ENA 설정에 대한 네트워크 성능 모니터링 섹션을 참조하세요.

ENA 대기열

ENA 대기열은 인스턴스 유형과 크기에 따라 기본 정적 제한이 있는 네트워크 인터페이스에 할당됩니다. 지원되는 인스턴스 유형에서는 이러한 대기열을 탄력적 네트워크 인터페이스(ENI)에 동적으로 할당할 수 있습니다. 인스턴스당 총 대기열 수는 유형과 크기에 따라 다르지만 ENI 및 인스턴스의 최대 대기열 수를 충족할 때까지 ENA 대기열로 여러 ENI를 구성할 수 있습니다.

유연한 ENA 대기열 할당은 리소스 배포를 최적화하여 vCPU 사용률을 극대화합니다. 높은 네트워크 성능 워크로드에는 일반적으로 여러 개의 ENA 대기열이 필요합니다. 특정 워크로드 요구 사항에 따라 대기열 수를 조정하여 네트워크 성능과 초당 패킷 수(PPS)를 미세 조정할 수 있습니다. 예를 들어 네트워크 집약적인 애플리케이션은 CPU 집약적인 애플리케이션에 비해 더 많은 대기열이 필요할 수 있습니다.

주제

- 지원되는 인스턴스
- ENA 대기열 가용성
- 대기열 수 수정

지원되는 인스턴스

다음 인스턴스는 여러 ENA 대기열의 동적 할당을 지원합니다.

- 범용: M6i, M6id, M6idn, M6in

- 컴퓨팅 최적화: C6i, C6id, C6in
- 메모리 최적화: R6i, R6in

다음 명령을 사용하여 인스턴스가 ENA 대기열의 동적 할당을 지원하는지 확인할 수 있습니다.

```
aws ec2 describe-instance-types --filters Name=network-info.flexible-ena-queues-
support,Values=supported
```

Amazon EC2 베어 메탈 인스턴스는 유연한 ENA 대기열을 지원하지 않습니다.

ENA 대기열 가용성

사용 가능한 ENA 대기열 수는 인스턴스 유형과 크기에 기반합니다. 다음 명령을 사용하여 사용 가능한 대기열 수를 찾습니다.

```
aws ec2 describe-instance-types --filters "Name=network-info.flexible-
ena-queues-support,Values=supported" --query "InstanceTypes[*].
[InstanceType,NetworkInfo.NetworkCards[*].DefaultEnaQueueCountPerInterface,NetworkInfo.NetworkC
 --output json
```

인스턴스의 모든 ENI에서 사용 가능한 `DefaultEnaQueueCountPerInterface`, `MaximumEnaQueueCountPerInterface`, `MaximumEnaQueueCount`를 볼 수 있습니다.

대기열 수 수정

AWS Management Console 또는 AWS CLI를 사용하여 ENA 대기열 수를 수정할 수 있습니다. AWS Management Console에서 ENA 대기열 구성은 각 네트워크 인터페이스 설정에서 사용할 수 있습니다.

AWS CLI를 사용하여 ENA 대기열 수를 수정하려면 다음 명령 중 하나를 사용합니다. 대기열 수를 수정하기 전에 다음 명령을 사용하여 현재 대기열 수를 확인합니다.

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

> **Note**
>
> - ENA 대기열 수를 수정하기 전에 인스턴스를 중지해야 합니다.
> - ENA 대기열의 값은 1, 2, 4, 8, 16, 32 등과 같이 2의 거듭제곱이어야 합니다.
> - 단일 ENI에 할당된 대기열 수는 인스턴스에서 사용 가능한 vCPU 수를 초과할 수 없습니다.

attach-network-interface

다음 예제에서는 ENI에 32개의 ENA 대기열이 구성됩니다.

```
aws ec2 attach-network-interface \
  --network-interface-id eni-001aa1bb223cdd4e4 \
  --instance-id i-1234567890abcdef0 \
  --device-index 1 \
  --ena-queue-count 32
```

run-instances

다음 예제에서는 3개의 ENI에 2개의 ENI 대기열이 구성됩니다.

```
aws ec2 run-instances \
  --image-id ami-12ab3c30 \
  --instance-type c6i.large \
  --min-count 1 \
  --max-count 1 \
  --network-interfaces \
    "[{\"DeviceIndex\":0,\"SubnetId\":\"subnet-123456789012a345a\",\"EnaQueueCount
\":2},
      {\"DeviceIndex\":1,\"SubnetId\":\"subnet-123456789012a345a\",\"EnaQueueCount
\":2},
      {\"DeviceIndex\":2,\"SubnetId\":\"subnet-123456789012a345a\",\"EnaQueueCount
\":2}]"
```

modify-network-interface-attribute

다음 예제에서는 ENI에 32개의 ENA 대기열이 구성됩니다.

```
aws ec2 modify-network-interface-attribute \
--network-interface-id eni-1234567890abcdef0 \
--attachment AttachmentId=eni-attach-12345678,EnaQueueCount=32
```

다음 예제에서는 ENA 수가 기본값으로 재설정됩니다.

```
aws ec2 modify-network-interface-attribute \
--network-interface-id eni-1234567890abcdef0 \
--attachment AttachmentId=eni-attach-12345678,DefaultEnaQueueCount=true
```

EC2 Windows 인스턴스에서 네트워크 성능 최적화

향상된 네트워킹을 지원하는 Windows 인스턴스에서 네트워크 성능을 최대화하려는 경우 기본 운영 체제 구성을 수정해야 할 수도 있습니다. 높은 네트워크 성능이 필요한 애플리케이션의 경우 다음과 같이 구성을 변경하는 것이 좋습니다. 다른 최적화(예: 체크섬 오프로드 켜기 및 RSS 활성화)는 공식 Windows AMI에 이미 구성되어 있습니다.

> **Note**
>
> 대부분 용도에서는 TCP Chimney 오프로드를 비활성화해야 하며, Windows Server 2016부터는 더 이상 사용되지 않았습니다.

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) 섹션을 참조하세요.

AWS은(는) 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 System 기반 인스턴스에 최신 버전의 ENA 드라이버를 사용하는 것이 좋습니다.

수신 측 조정 CPU 선호도 구성

네트워크 트래픽 CPU 로드를 여러 프로세서로 분산하는 데 수신 측 조정(RSS)이 사용됩니다. 기본적으로 공식 Amazon Windows AMI는 RSS가 활성화된 상태로 구성됩니다. ENA 탄력적 네트워크 인터페이스는 최대 8개의 RSS 대기열을 제공합니다. RSS 대기열 및 다른 시스템 프로세스에 대한 CPU 선호도를 정의하면, 다중 코어 시스템으로 CPU 로드를 분산해 더 많은 네트워크 트래픽을 처리할 수 있습니다. vCPU를 16개 이상 사용하는 인스턴스 유형의 경우 `Set-NetAdapterRSS` PowerShell cmdlet 사용을 권장합니다. 다양한 시스템 구성 요소의 경합을 방지하기 위해 모든 탄력적 네트워크 인터페이스의 RSS 구성에서 부트 프로세서(하이퍼 스레딩 사용 시 논리 프로세서 0 및 1)를 수동으로 제외합니다.

Windows는 하이퍼 스레드를 인식하며, 단일 NIC(네트워크 인터페이스 카드)의 RSS 대기열이 언제나 다른 물리적 코어에 위치하게 합니다. 따라서 다른 NIC와의 경합 방지를 위해 하이퍼 스레딩을 사용 중지하지 않는다면, 각 NIC의 RSS 구성을 16개의 논리적 프로세서에 분산합니다. `Set-NetAdapterRss` cmdlet을 사용하면 BaseProcessorGroup, BaseProcessorNumber, MaxProcessingGroup, MaxProcessorNumber 및 NumaNode(선택 사항) 값을 정의하여 유효한 논리적

프로세서의 NIC당 범위를 정의할 수 있습니다. 물리적 코어가 부족해 NIC간 경합을 완전히 제거할 수 없다면, 중첩되는 범위를 최소화하거나 탄력적 네트워크 인터페이스의 예상 워크로드에 따라 인터페이스 범위에 있는 논리적 프로세서의 수를 줄여보세요(적은 볼륨의 관리자 네트워크 인터페이스는 할당된 RSS 대기열보다 적은 대기열이 필요할 수 있습니다). 또한 위에서 언급했듯이 다양한 구성 요소를 CPU 0에서 실행해야 하므로 충분한 vCPU를 사용할 수 있다면 모든 RSS 구성에서 이를 제외하는 것이 좋습니다.

예를 들어 하이퍼 스레딩이 활성화된 NUMA 노드 2개가 있는 72 vCPU 인스턴스에 탄력적 네트워크 인터페이스 3개가 있다면, 다음 명령은 중첩되거나 코어 0 사용을 완전히 방지하지 않고도 네트워크 로드를 두 CPU에 분산합니다.

```
Set-NetAdapterRss -Name NIC1 -BaseProcessorGroup 0 -BaseProcessorNumber 2 -
MaxProcessorNumber 16
Set-NetAdapterRss -Name NIC2 -BaseProcessorGroup 1 -BaseProcessorNumber 0 -
MaxProcessorNumber 14
Set-NetAdapterRss -Name NIC3 -BaseProcessorGroup 1 -BaseProcessorNumber 16 -
MaxProcessorNumber 30
```

이러한 설정은 각 네트워크 어댑터에 영구적으로 적용됩니다. 인스턴스가 vCPU 숫자가 다른 인스턴스로 크기 조정된다면, 활성화된 각 탄력적 네트워크 인터페이스에 대해 RSS 구성을 재평가해야 합니다. cmdlet에 대한 전체 Microsoft 설명서는 Set-NetAdapterRss에서 확인할 수 있습니다.

SQL 워크로드 전용 참고 사항: 같은 CPU에 대한 I/O와 네트워크 경합을 최소화할 수 있도록, I/O 스레드 선호도 설정과 탄력적 네트워크 인터페이스 RSS 구성을 검토해 보세요. 선호도 마스크 서버 구성 옵션을 확인하세요.

Amazon EC2의 AI/ML 및 HPC 워크로드를 위한 Elastic Fabric Adapter

Elastic Fabric Adapter(EFA)는 Amazon EC2 인스턴스에 연결하여 인공 지능(AI), 기계 학습(ML), 고성능 컴퓨팅(HPC) 애플리케이션을 가속화할 수 있는 네트워크 디바이스입니다. EFA를 사용하면 AWS 클라우드가 제공하는 확장성, 유연성, 탄력성을 통해 온프레미스 AI/ML 또는 HPC 클러스터의 애플리케이션 성능을 구현할 수 있습니다.

EFA는 전통적으로 클라우드 기반 HPC 시스템에서 사용하는 TCP 전송보다 지연율이 낮고 일정하며 더 높은 처리량을 제공합니다. 또한 대규모 AI/ML 및 HPC 애플리케이션에서 중요한 인스턴스 간 통신 성능을 확장합니다. 이는 기존 AWS 네트워크 인프라에서 작업하도록 최적화되어 애플리케이션 요구량에 따라 크기를 변경합니다.

EFA는 Libfabric 1.7.0 이상과 통합되며 AI 및 ML 애플리케이션을 위한 NCCL(Nvidia Collective Communications Library)과 HPC 애플리케이션을 위한 Open MPI 4.1 이상, Intel MPI 2019 업데이트 5 이상을 지원합니다.

EFA는 Nitro 버전 4 이상이 설치된 지원되는 대부분의 인스턴스 유형에서 RDMA(Remote Direct Memory Access) 쓰기를 지원합니다. RDMA 읽기는 Nitro 버전 4 이상의 모든 인스턴스에서 지원됩니다. 자세한 내용은 지원되는 인스턴스 유형 섹션을 참조하세요.

내용

- EFA 기본 사항
- 지원되는 인터페이스 및 라이브러리
- 지원되는 인스턴스 유형
- 지원되는 운영 체제
- EFA 제한 사항
- EFA 요금
- Amazon EC2에서 HPC 워크로드를 위한 EFA 및 MPI 시작하기
- Amazon EC2에서 ML 워크로드를 위한 EFA 및 NCCL 시작하기
- 여러 네트워크 카드를 사용하여 Amazon EC2 인스턴스의 네트워크 대역폭 극대화
- Elastic Fabric Adapter를 생성하여 Amazon EC2 인스턴스에 연결
- Amazon EC2 인스턴스에서 EFA 분리 및 삭제
- Amazon EC2에서 Elastic Fabric Adapter 모니터링
- 체크섬을 사용하여 EFA 설치 프로그램 확인
- Elastic Fabric Adapter 릴리스 정보

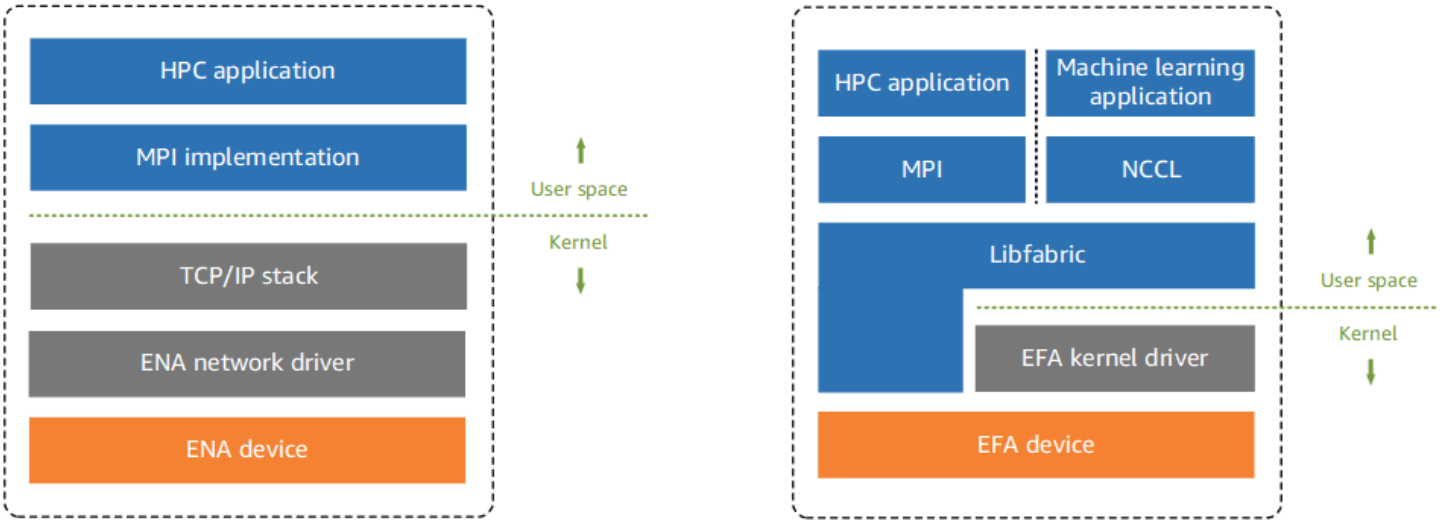
EFA 기본 사항

EFA 디바이스는 두 가지 방법으로 EC2 인스턴스에 연결할 수 있습니다.

1. EFA 디바이스와 ENA 디바이스를 모두 생성하는 기존 EFA 인터페이스(ENA 포함 EFA라고도 함)를 사용합니다.
2. EFA 디바이스만 생성하는 EFA 전용 인터페이스를 사용합니다.

EFA 디바이스는 SRD(Scalable Reliable Datagram) 프로토콜을 통해 내장 OS 바이패스 및 혼잡 제어와 같은 기능을 제공합니다. EFA 디바이스 기능을 사용하면 지연 시간이 짧고 안정적인 전송 기능을

통해 EFA 인터페이스가 Amazon EC2의 HPC 및 ML 애플리케이션에 더 나은 애플리케이션 성능을 제공할 수 있습니다. ENA 디바이스는 기존 IP 네트워킹을 제공합니다.



전통적으로 AI/ML 애플리케이션은 NCCL을 사용하고 HPC 애플리케이션은 MPI(Message Passing Interface)를 사용하여 시스템의 네트워크 전송과 인터페이스합니다. AWS 클라우드에서 NCCL 또는 MPI를 사용하는 애플리케이션 인터페이스를 의미하며 이는 인스턴스 간 네트워크 통신을 위해 운영 체제의 TCP/IP 스택과 ENA 디바이스 드라이버를 사용한다는 의미입니다.

기존 EFA(ENA 포함 EFA) 또는 EFA 전용 인터페이스와 함께, AI/ML 애플리케이션은 NCCL을 사용하고 HPC 애플리케이션은 MPI를 사용하여 Libfabric API와 직접 인터페이스합니다. Libfabric API는 운영 체제 커널을 우회하여 EFA 디바이스와 직접 통신을 통해 네트워크에 패킷을 전송합니다. 이를 통해 오버헤드를 줄이고 AI/ML 및 HPC 애플리케이션을 더욱 효율적으로 실행할 수 있습니다.

> **Note**
>
> Libfabric은 OFI(OpenFabrics Interface) 프레임워크의 핵심 구성 요소로서 OFI의 사용자 공간 API를 정의하고 내보냅니다. 자세한 내용은 Libfabric OpenFabrics 웹 사이트를 참조하세요.

ENA, EFA 및 EFA 전용 네트워크 인터페이스 간의 차이점

Amazon EC2는 두 가지 유형의 네트워크 인터페이스를 제공합니다.

- ENA 인터페이스는 VPC를 위한 IP 네트워킹을 지원하는 데 필요한 모든 기존 IP 네트워킹 및 라우팅 기능을 제공합니다. 자세한 내용은 EC2 인스턴스에서 ENA로 향상된 네트워킹 활성화 섹션을 참조하세요.

- EFA(ENA 포함 EFA) 인터페이스는 IP 네트워킹을 위한 ENA 디바이스와 짧은 지연 시간, 높은 처리량 통신을 위한 EFA 디바이스를 모두 제공합니다.
- EFA 전용 인터페이스는 기존 IP 네트워킹을 위한 ENA 디바이스 없이 EFA 디바이스 기능만 지원합니다.

다음 표에서는 ENA, EFA, EFA 전용 네트워크 인터페이스를 비교합니다.

	ENA	EFA(ENA 포함 EFA)	EFA 전용
IP 네트워킹 기능 지원	예	예	아니요
IPv4 또는 IPv6 주소 할당 가능	예	예	아니요
인스턴스의 기본 네트워크 인터페이스로 사용 가능	예	예	아니요
인스턴스의 ENI 연결 한도에 가산	예	예	예
인스턴스 유형 지원	모든 Nitro 기반 인스턴스 유형에서 지원	지원되는 인스턴스 유형	지원되는 인스턴스 유형
EC2 API의 파라미터 이름 지정	`interface`	`efa`	`efa-only`
EC2 콘솔의 필드 이름 지정	선택 사항 없음	ENA 포함 EFA	EFA 전용

지원되는 인터페이스 및 라이브러리

EFA는 다음 인터페이스 및 라이브러리를 지원합니다.

- MPI 4.1 이상 열기
- 인텔 MPI 2019 업데이트 5 이상
- NCCL(Nvidia Collective Communications Library) 2.4.2 이상
- AWS Neuron SDK 버전 2.3 이상

지원되는 인스턴스 유형

다음 인스턴스 유형 모두 EFA를 지원합니다. 또한 표에는 인스턴스 유형에 대한 RDMA 읽기 및 RDMA 쓰기 지원이 표시되어 있습니다.

Nitro v6

인스턴스 유형	RDMA 읽기 지원	RDMA 쓰기 지원
가속 컴퓨팅		
p6-b200.48xlarge		

Nitro v5

인스턴스 유형	RDMA 읽기 지원	RDMA 쓰기 지원
범용		
m8g.24xlarge		
m8g.48xlarge		
m8g.metal-24xl		
m8g.metal-48xl		
m8gd.24xlarge		

인스턴스 유형	RDMA 읽기 지원	RDMA 쓰기 지원
m8gd.48xlarge		
m8gd.metal-24xl		
m8gd.metal-48xl		
컴퓨팅 최적화		
c7gn.16xlarge		
c7gn.metal		
c8g.24xlarge		
c8g.48xlarge		
c8g.metal-24xl		
c8g.metal-48xl		
c8gd.24xlarge		
c8gd.48xlarge		
c8gd.metal-24xl		
c8gd.metal-48xl		
메모리 최적화		
r8g.24xlarge		
r8g.48xlarge		
r8g.metal-24xl		
r8g.metal-48xl		
r8gd.24xlarge		
r8gd.48xlarge		

인스턴스 유형	RDMA 읽기 지원	RDMA 쓰기 지원
r8gd.metal-24xl		
r8gd.metal-48xl		
x8g.24xlarge		
x8g.48xlarge		
x8g.metal-24xl		
x8g.metal-48xl		
스토리지 최적화		
i7ie.48xlarge		
i7ie.metal-48xl		
i8g.48xlarge		
가속 컴퓨팅		
p5en.48xlarge		
trn2.48xlarge		
trn2u.48xlarge		
고성능 컴퓨팅		
hpc7g.4xlarge		
hpc7g.8xlarge		
hpc7g.16xlarge		

Nitro v4

인스턴스 유형	RDMA 읽기 지원	RDMA 쓰기 지원
범용		
m6a.48xlarge		
m6a.metal		
m6i.32xlarge		
m6i.metal		
m6id.32xlarge		
m6id.metal		
m6idn.32xlarge		
m6idn.metal		
m6in.32xlarge		
m6in.metal		
m7a.48xlarge		
m7a.metal-48xl		
m7g.16xlarge		
m7g.metal		
m7gd.16xlarge		
m7gd.metal		
m7i.48xlarge		
m7i.metal-48xl		
컴퓨팅 최적화		

인스턴스 유형	RDMA 읽기 지원	RDMA 쓰기 지원
c6a.48xlarge		
c6a.metal		
c6gn.16xlarge		
c6i.32xlarge		
c6i.metal		
c6id.32xlarge		
c6id.metal		
c6in.32xlarge		
c6in.metal		
c7a.48xlarge		
c7a.metal-48xl		
c7g.16xlarge		
c7g.metal		
c7gd.16xlarge		
c7gd.metal		
c7i.48xlarge		
c7i.metal-48xl		
메모리 최적화		
r6a.48xlarge		
r6a.metal		
r6i.32xlarge		

인스턴스 유형	RDMA 읽기 지원	RDMA 쓰기 지원
r6i.metal		
r6idn.32xlarge		
r6idn.metal		
r6in.32xlarge		
r6in.metal		
r6id.32xlarge		
r6id.metal		
r7a.48xlarge		
r7a.metal-48xl		
r7g.16xlarge		
r7g.metal		
r7gd.16xlarge		
r7gd.metal		
r7i.48xlarge		
r7i.metal-48xl		
r7iz.32xlarge		
r7iz.metal-32xl		
u7i-6tb.112xlarge		
u7i-8tb.112xlarge		
u7i-12tb.224xlarge		
u7in-16tb.224xlarge		

인스턴스 유형	RDMA 읽기 지원	RDMA 쓰기 지원
u7in-24tb.224xlarge		
u7in-32tb.224xlarge		
u7inh-32tb.480xlarge		
x2idn.32xlarge		
x2idn.metal		
x2iedn.32xlarge		
x2iedn.metal		
스토리지 최적화		
i4g.16xlarge		
i4i.32xlarge		
i4i.metal		
i7i.24xlarge		
i7i.48xlarge		
i7i.metal-48xl		
im4gn.16xlarge		
가속 컴퓨팅		
f2.48xlarge		
g6.8xlarge		
g6.12xlarge		
g6.16xlarge		
g6.24xlarge		

인스턴스 유형	RDMA 읽기 지원	RDMA 쓰기 지원
g6.48xlarge		
g6e.8xlarge		
g6e.12xlarge		
g6e.16xlarge		
g6e.24xlarge		
g6e.48xlarge		
gr6.8xlarge		
p5.48xlarge		
p5e.48xlarge		
trn1.32xlarge		
trn1n.32xlarge		
고성능 컴퓨팅		
hpc6a.48xlarge		
hpc6id.32xlarge		
hpc7a.12xlarge		
hpc7a.24xlarge		
hpc7a.48xlarge		
hpc7a.96xlarge		

Nitro v3

인스턴스 유형	RDMA 읽기 지원	RDMA 쓰기 지원
범용		
m5dn.24xlarge		
m5dn.metal		
m5n.24xlarge		
m5n.metal		
m5zn.12xlarge		
m5zn.metal		
컴퓨팅 최적화		
c5n.9xlarge		
c5n.18xlarge		
c5n.metal		
메모리 최적화		
r5dn.24xlarge		
r5dn.metal		
r5n.24xlarge		
r5n.metal		
x2iezn.12xlarge		
x2iezn.metal		
스토리지 최적화		
i3en.12xlarge		

인스턴스 유형	RDMA 읽기 지원	RDMA 쓰기 지원
i3en.24xlarge		
i3en.metal		
가속 컴퓨팅		
dl1.24xlarge		
dl2q.24xlarge		
g4dn.8xlarge		
g4dn.12xlarge		
g4dn.16xlarge		
g4dn.metal		
g5.8xlarge		
g5.12xlarge		
g5.16xlarge		
g5.24xlarge		
g5.48xlarge		
inf1.24xlarge		
p3dn.24xlarge		
p4d.24xlarge		
p4de.24xlarge		
vt1.24xlarge		

특정 리전에서 EFA를 지원하는 사용 가능한 인스턴스 유형 확인

사용 가능한 인스턴스 유형은 리전마다 다릅니다. 리전에서 EFA를 지원하는 사용 가능한 인스턴스 유형을 확인하려면 describe-instance-types 명령을 `--region` 파라미터와 함께 사용합니다. EFA를 지원하는 인스턴스 유형으로 결과 범위를 지정하려면 `--filters` 파라미터를 포함하고 `InstanceType` 값으로 출력 범위를 지정하려면 `--query` 파라미터를 포함합니다.

```
aws ec2 describe-instance-types \
    --region us-east-1  \
    --filters Name=network-info.efa-supported,Values=true \
    --query "InstanceTypes[*].[InstanceType]"  \
    --output text | sort
```

지원되는 운영 체제

운영 체제 지원은 프로세서 유형에 따라 다릅니다. 다음 표는 지원되는 운영 체제를 보여 줍니다.

운영 체제	Intel/AMD(**x86_64**) 인스턴스 유형	AWS Graviton(**arm64**) 인스턴스 유형
Amazon Linux 2023	✓	✓
Amazon Linux 2	✓	✓
RHEL 8 및 9	✓	✓
Debian 11 및 12	✓	✓
Rocky Linux 8과 9	✓	✓
Ubuntu 22.04 및 24.04	✓	✓
SUSE Linux Enterprise 15 SP2 이상	✓	✓
OpenSUSE Leap 15.5 이상	✓	

> **Note**
>
> 나열된 운영 체제 중 일부는 Intel MPI에서 지원되지 않을 수 있습니다. Intel MPI를 사용하는 경우 Intel MPI 설명서를 참조하여 운영 체제에 대한 지원을 확인하세요.

EFA 제한 사항

EFA에는 다음과 같은 제한 사항이 있습니다.

> **Note**
>
> EFA 트래픽은 EFA(ENA 포함 EFA) 또는 EFA 전용 인터페이스의 EFA 디바이스를 통해 전송되는 트래픽을 말합니다.

- RDMA 쓰기는 일부 인스턴스 유형에서 지원되지 않습니다. 자세한 내용은 지원되는 인스턴스 유형 섹션을 참조하세요.
- P4d/P4de/DL1 인스턴스와 다른 인스턴스 유형 간의 EFA 트래픽은 현재 지원되지 않습니다.
- 여러 네트워크 카드를 지원하는 인스턴스 유형은 네트워크 카드당 하나의 EFA로 구성할 수 있습니다. 지원되는 다른 모든 인스턴스 유형은 인스턴스당 하나의 EFA만 지원합니다.
- `c7g.16xlarge`, `m7g.16xlarge` 및 `r7g.16xlarge` 전용 인스턴스 및 전용 호스트의 경우 EFA 연결 시 지원되지 않습니다.
- EFA 트래픽은 가용 영역 또는 VPC를 통과할 수 없습니다. 이는 EFA 인터페이스의 ENA 디바이스에서 발생하는 일반 IP 트래픽에는 적용되지 않습니다.
- EFA 트래픽은 라우팅할 수 없습니다. EFA 인터페이스의 ENA 디바이스에서 발생하는 일반 IP 트래픽은 라우팅 가능한 상태로 유지됩니다.
- EFA는 AWS Outposts에서 지원되지 않습니다.
- EFA(ENA 포함 EFA) 인터페이스의 EFA 디바이스는 AWS Cloud Digital Interface 소프트웨어 개발 키트(AWS CDI SDK) 기반 애플리케이션용 Windows 인스턴스에서만 지원됩니다. CDI SDK 기반이 아닌 애플리케이션의 Windows 인스턴스에 EFA(ENA 포함 EFA) 인터페이스를 연결하면 추가된 EFA 디바이스 기능 없이 ENA 인터페이스로 작동합니다. EFA 전용 인터페이스는 Windows 또는 Linux에서 실행되는 AWS CDI 기반 애플리케이션에서 지원되지 않습니다. 자세한 내용은 AWS Cloud Digital Interface 소프트웨어 개발 키트(AWS CDI SDK) 사용 설명서를 참조하세요.

EFA 요금

EFA는 추가 비용 없이 지원되는 모든 인스턴스에서 활성화할 수 있는 선택적 Amazon EC2 네트워킹 기능으로 사용할 수 있습니다.

Amazon EC2에서 HPC 워크로드를 위한 EFA 및 MPI 시작하기

이 자습서는 HPC 워크로드를 위한 MPI 지원 인스턴스와 EFA를 시작하는 데 도움이 됩니다.

> **Note**
>
> `u7i-12tb.224xlarge`, `u7in-16tb.224xlarge`, `u7in-24tb.224xlarge`, `u7in-32tb.224xlarge` 인스턴스는 Open MPI를 사용하는 최대 128개의 병렬 MPI 프로세스 또는 Intel MPI를 사용하는 최대 256개의 병렬 MPI 프로세스를 실행할 수 있습니다.

업무

- 1단계: EFA를 사용한 보안 그룹 준비
- 2단계: 임시 인스턴스 시작
- 3단계: EFA 소프트웨어 설치
- 4단계: (선택 사항) Open MPI 5 활성화
- 5단계: (선택 사항) 인텔 MPI 설치
- 6단계: ptrace 보호 비활성화
- 7단계. 설치 확인
- 8단계: HPC 애플리케이션 설치
- 9단계: EFA 지원 AMI 생성
- 10단계: 클러스터 배치 그룹으로 EFA 지원 인스턴스 시작
- 11단계: 임시 인스턴스 종료
- 12단계: 암호 없는 SSH 활성화

1단계: EFA를 사용한 보안 그룹 준비

EFA에는 보안 그룹 자체 내의 모든 인바운드 및 아웃바운드 트래픽을 허용하는 보안 그룹이 필요합니다. 다음 절차에서는 자체적으로 들어오고 나가는 모든 인바운드 및 아웃바운드 트래픽을 허용하고 SSH 연결을 위해 모든 IPv4 주소의 인바운드 SSH 트래픽을 허용하는 보안 그룹을 생성합니다.

> **Important**
>
> 이 보안 그룹은 테스트 목적으로만 사용됩니다. 프로덕션 환경에서는 컴퓨터의 IP 주소 또는 로컬 네트워크의 IP 주소 범위와 같이 연결하려는 IP 주소로부터의 트래픽만 허용하는 인바운드 SSH 규칙을 생성하는 것이 좋습니다.

다른 시나리오는 다양한 사용 사례에 대한 보안 그룹 규칙 섹션을 참조하세요.

EFA 사용 보안 그룹을 생성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 보안 그룹(Security Groups)을 선택한 다음, 보안 그룹 생성(Create security group)을 선택합니다.
3. 보안 그룹 생성(Create security group) 창에서 다음을 수행합니다.
   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 EFA 사용 인스턴스를 시작하려는 VPC를 선택합니다.
   d. 보안 그룹 생성을 선택합니다.
4. 생성한 보안 그룹을 선택하고 세부 정보(Details) 탭에서 보안 그룹 ID(Security group ID)를 복사합니다.
5. 보안 그룹을 선택한 상태에서 작업(Actions), 인바운드 규칙 편집(Edit inbound rules)을 선택한 후 다음을 수행합니다.
   a. [Add another rule]을 선택합니다.
   b. 유형(Type)에서 모든 트래픽(All traffic)을 선택합니다.
   c. 소스 유형(Source type)에서 사용자 지정(Custom)을 선택하고 복사한 보안 그룹 ID를 필드에 붙여넣습니다.
   d. 규칙 추가를 선택합니다.
   e. Type(유형)에서 SSH를 선택합니다.
   f. 소스 유형(Source type)에서 어디서나 - IPv4(Anywhere - IPv4)를 선택합니다.
   g. 규칙 저장을 선택합니다.

6. 보안 그룹을 선택한 상태에서 작업(Actions), 아웃바운드 규칙 편집(Edit outbound rules)을 선택한 후 다음을 수행합니다.

   a. [Add another rule]을 선택합니다.

   b. 유형(Type)에서 모든 트래픽(All traffic)을 선택합니다.

   c. 대상 유형(Destination type)에서 사용자 지정(Custom)을 선택하고 복사한 보안 그룹 ID를 필드에 붙여넣습니다.

   d. 규칙 저장을 선택합니다.

2단계: 임시 인스턴스 시작

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. 탐색 창에서 Instances(인스턴스)를 선택한 다음에 Launch Instances(인스턴스 시작)를 선택하여 새 인스턴스 시작 마법사를 엽니다.
3. (선택 사항)이름 및 태그(Name and tags) 섹션에서 인스턴스의 이름(예: `EFA-instance`)을 제공합니다. 이름은 인스턴스에 리소스 태그(`Name=`*`EFA-instance`*)로 할당됩니다.
4. Application and OS Images(애플리케이션 및 OS 이미지) 섹션에서 지원되는 운영 체제 중 하나에 해당하는 AMI를 선택합니다.
5. 인스턴스 유형(Instance type) 섹션에서 지원되는 인스턴스 유형(supported instance type)을 선택합니다.
6. 키 페어(Key pair) 섹션에서 인스턴스에 사용할 키 페어를 선택합니다.
7. 네트워크 설정(Network settings) 섹션에서 편집(Edit)을 선택하고 다음과 같이 수행합니다.

   a. 서브넷(Subnet)에서 인스턴스를 시작할 서브넷을 선택합니다. 서브넷을 선택하지 않으면 EFA에 대해 인스턴스를 활성화할 수 없습니다.

   b. 방화벽(보안 그룹)의 경우 기존 보안 그룹 선택(Select existing security group)을 선택한 다음에 이전 단계에서 생성한 보안 그룹을 선택합니다.

   c. 고급 네트워크 구성 섹션을 확장합니다.

      네트워크 인터페이스 1에서 네트워크 카드 인덱스 = 0, 디바이스 인덱스 = 0 및 인터페이스 유형 = ENA 포함 EFA를 선택합니다.

(선택 사항) p4d.24xlarge 또는 p5.48xlarge와 같은 다중 카드 인스턴스 유형을 사용하는 경우, 필요한 추가 네트워크 인터페이스마다 네트워크 인터페이스 추가를 선택하고 네트워크 카드 인덱스에서 다음 미사용 인덱스를 선택한 다음, 디바이스 인덱스 = 1 및 인터페이스 유형 = ENA 포함 EFA 또는 EFA 전용을 선택합니다.

8. 스토리지(Storage) 섹션에서 필요에 따라 볼륨을 구성합니다.
9. 오른쪽의 요약(Summary) 패널에서 인스턴스 시작(Launch instance)을 선택합니다.

> **Note**
>
> IMDSv2를 이미 계정의 기본값으로 설정하지 않은 경우 9단계에서 생성할 AMI뿐만 아니라 임시 인스턴스에도 IMDSv2를 사용하도록 요구하는 것을 고려해 보세요. IMDSv2 구성 단계에 대한 자세한 내용은 새 인스턴스에 대한 인스턴스 메타데이터 옵션 구성 섹션을 참조하세요.

3단계: EFA 소프트웨어 설치

임시 인스턴스에서 EFA를 지원하는 데 필요한 EFA 사용 커널, EFA 드라이버, Libfabric 및 Open MPI 스택을 설치합니다.

이 단계는 EFA를 Open MPI에서 사용할지, Intel MPI에서 사용할지 또는 Open MPI 및 Intel MPI에서 사용할지에 따라 다릅니다.

> **Note**
>
> 일부 운영 체제는 Intel MPI에서 지원되지 않을 수 있습니다. Intel MPI를 사용하는 경우 Intel MPI 설명서를 참조하여 운영 체제에 대한 지원을 확인하세요.

EFA 소프트웨어를 설치하려면

1. 시작한 인스턴스에 연결합니다. 자세한 내용은 SSH를 사용하여 Linux 인스턴스에 연결 섹션을 참조하세요.
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 2023, Amazon Linux 2, RHEL 8/9, Rocky Linux 8/9

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

- Ubuntu 및 Debian

```
$ sudo apt-get update && sudo apt-get upgrade -y
```

- SUSE Linux Enterprise

```
$ sudo zypper update -y
```

3. 인스턴스를 재부팅하고 다시 연결합니다.
4. EFA 소프트웨어 설치 파일을 다운로드합니다. 소프트웨어 설치 파일은 압축 tarball(`.tar.gz`) 파일로 패키징되어 있습니다. 최신 안정 버전을 다운로드하려면 다음 명령을 사용하십시오.

   또한 위의 명령에서 버전 번호를 `latest`로 바꾸면 최신 버전을 얻을 수 있습니다.

```
$ curl -O https://efa-installer.amazonaws.com/aws-efa-installer-1.42.0.tar.gz
```

5. (선택 사항) EFA tarball(`.tar.gz`) 파일의 신뢰성 및 무결성을 확인합니다.

   이 작업을 수행하여 소프트웨어 게시자의 자격 증명을 확인하고 파일이 게시된 이후 변경되거나 손상되지 않았는지 확인하는 것이 좋습니다. tarball 파일을 확인하지 않으려면 이 단계를 건너뜁니다.

   > **Note**
   >
   > 또는 MD5 또는 SHA256 체크섬을 대신 사용하여 tarball 파일을 확인하려면 체크섬을 사용하여 EFA 설치 프로그램 확인 섹션을 참조하세요.

   a. 퍼블릭 GPG 키를 다운로드하고 키 링으로 가져옵니다.

```
$ wget https://efa-installer.amazonaws.com/aws-efa-installer.key && gpg --
import aws-efa-installer.key
```

      명령에서 키 값이 반환됩니다. 다음 단계에서 필요하므로 키 값을 기록해 둡니다.

   b. GPG 키의 지문을 확인합니다. 다음 명령을 실행하고 이전 단계의 키 값을 지정합니다.

```
$ gpg --fingerprint key_value
```

명령에서 4E90 91BC BB97 A96B 26B1 5E59 A054 80B1 DD2D 3CCC와 동일한 지문이 반환되어야 합니다. 지문이 일치하지 않으면 EFA 설치 스크립트를 실행하지 말고 지원에 문의하세요.

c. 서명 파일을 다운로드하고 EFA tarball 파일의 서명을 확인합니다.

```
$ wget https://efa-installer.amazonaws.com/aws-efa-installer-1.42.0.tar.gz.sig
 && gpg --verify ./aws-efa-installer-1.42.0.tar.gz.sig
```

다음은 출력의 예입니다.

```
gpg: Signature made Wed 29 Jul 2020 12:50:13 AM UTC using RSA key ID DD2D3CCC
gpg: Good signature from "Amazon EC2 EFA <ec2-efa-maintainers@amazon.com>"
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: 4E90 91BC BB97 A96B 26B1  5E59 A054 80B1 DD2D 3CCC
```

결과에 Good signature가 있고 이전 단계에서 반환된 지문과 지문이 일치하면 다음 단계로 진행합니다. 그렇지 않은 경우 EFA 설치 스크립트를 실행하지 말고 지원에 문의하세요.

6. 압축 .tar.gz 파일에서 파일을 추출하고 추출된 디렉터리로 이동하십시오.

```
$ tar -xf aws-efa-installer-1.42.0.tar.gz && cd aws-efa-installer
```

7. EFA 소프트웨어를 설치합니다. 사용 사례에 따라 다음 중 하나를 수행합니다.

> **Note**
>
> EFA는 SUSE 리눅스에서 NVIDIA GPUDirect를 지원하지 않습니다. SUSE Linux를 사용하는 경우 kmod 설치를 방지하려면 --skip-kmod 옵션을 추가로 지정해야 합니다. 기본적으로 SUSE Linux는 트리 외부 커널 모듈을 허용하지 않습니다.

Open MPI and Intel MPI

Open MPI 및 Intel MPI에서 EFA를 사용하려는 경우 Libfabric 및 Open MPI와 함께 EFA 소프트웨어를 설치해야 하며, 5단계: Intel MPI 설치를 완료해야 합니다.

Libfabric 및 Open MPI와 함께 EFA 소프트웨어를 설치하려면 다음 명령을 실행합니다.

> **Note**
>
> EFA 1.30.0부터 Open MPI 4.1 및 Open MPI 5 모두 기본적으로 설치됩니다. 설치하려는 Open MPI 버전을 선택적으로 지정할 수 있습니다. Open MPI 4.1만 설치하려면 `--mpi=openmpi4`를 포함합니다. Open MPI 5만 설치하려면 `--mpi=openmpi5`를 포함합니다. 둘 다 설치하려면 `--mpi` 옵션을 생략합니다.

```
$ sudo ./efa_installer.sh -y
```

Libfabric이 `/opt/amazon/efa`에 설치됩니다. Open MPI 4.1이 `/opt/amazon/openmpi`에 설치됩니다. Open MPI 5가 `/opt/amazon/openmpi5`에 설치됩니다.

Open MPI only

Open MPI에서만 EFA를 사용하려는 경우 Libfabric 및 Open MPI와 함께 EFA 소프트웨어를 설치해야 하며, 5단계: Intel MPI 설치를 건너뛰어야 합니다. Libfabric 및 Open MPI와 함께 EFA 소프트웨어를 설치하려면 다음 명령을 실행합니다.

> **Note**
>
> EFA 1.30.0부터 Open MPI 4.1 및 Open MPI 5 모두 기본적으로 설치됩니다. 설치하려는 Open MPI 버전을 선택적으로 지정할 수 있습니다. Open MPI 4.1만 설치하려면 `--mpi=openmpi4`를 포함합니다. Open MPI 5만 설치하려면 `--mpi=openmpi5`를 포함합니다. 둘 다 설치하려면 `--mpi` 옵션을 생략합니다.

```
$ sudo ./efa_installer.sh -y
```

Libfabric이 `/opt/amazon/efa`에 설치됩니다. Open MPI 4.1이 `/opt/amazon/openmpi`에 설치됩니다. Open MPI 5가 `/opt/amazon/openmpi5`에 설치됩니다.

Intel MPI only

Intel MPI에서만 EFA를 사용하려는 경우 Libfabric 및 Open MPI 없이 EFA 소프트웨어를 설치할 수 있습니다. 이 경우 Intel MPI는 임베디드 Libfabric을 사용합니다. 이렇게 하려면 5단계: Intel MPI 설치를 완료해야 합니다.

Libfabric 및 Open MPI없이 EFA 소프트웨어를 설치하려면 다음 명령을 실행합니다.

```
$ sudo ./efa_installer.sh -y --minimal
```

8. EFA 설치 프로그램에서 인스턴스를 재부팅하라는 메시지를 표시하면 인스턴스를 재부팅한 다음 인스턴스에 다시 연결합니다. 그렇게 하지 않으면 인스턴스에서 로그아웃한 다음 다시 로그인하여 설치를 완료합니다.
9. 압축되지 않은 tarball과 tarball 자체를 삭제합니다. 그렇지 않으면 생성한 EFA 지원 AMI에 포함되므로 크기가 늘어납니다.

4단계: (선택 사항) Open MPI 5 활성화

> **Note**
>
> Open MPI 5를 사용하려는 경우에만 이 단계를 수행합니다.

EFA 1.30.0부터 Open MPI 4.1 및 Open MPI 5 모두 기본적으로 설치됩니다. 또는 Open MPI 4.1 또는 Open MPI 5만 설치하도록 선택할 수도 있습니다.

3단계: EFA 소프트웨어 설치에서 Open MPI 5를 설치하기로 선택한 후 이를 사용하려는 경우 다음 단계를 수행하여 활성화해야 합니다.

Open MPI 5를 활성화하는 방법

1. Open MPI 5를 PATH 환경 변수에 추가합니다.

```
$ module load openmpi5
```

2. Open MPI 5를 사용할 수 있도록 활성화했는지 확인합니다.

```
$ which mpicc
```

이 명령은 Open MPI 5 설치 디렉터리(/opt/amazon/openmpi5)를 반환해야 합니다.

3. (선택 사항) 인스턴스가 시작될 때마다 Open MPI 5가 PATH 환경 변수에 추가되도록 하려면 다음을 수행합니다.

   bash shell

   `module load openmpi5`를 /home/*username*/.bashrc 및 /home/*username*/.bash_profile에 추가합니다.

   csh and tcsh shells

   `module load openmpi5`을 /home/*username*/.cshrc에 추가합니다.

PATH 환경 변수에서 Open MPI 5를 제거해야 하는 경우 다음 명령을 실행하고 쉘 시작 스크립트에서 명령을 제거합니다.

```
$ module unload openmpi5
```

5단계: (선택 사항) 인텔 MPI 설치

> **Important**
>
> Intel MPI를 사용하려는 경우에만 이 단계를 수행하세요. Open MPI만 사용하려면 이 단계를 건너뜁니다.

Intel MPI에는 추가 설치 및 환경 변수 구성이 필요합니다.

전제 조건

다음 단계를 수행하는 사용자가 sudo 권한을 가지고 있는지 확인하세요.

Intel MPI를 설치하려면

1. 인텔 MPI 설치 스크립트를 다운로드하려면 다음을 수행하세요.

   a. 인텔 웹 사이트를 방문합니다.

   b. 웹 페이지의 인텔 MPI 라이브러리(Intel MPI Library) 섹션에서 Intel MPI Library for Linux 오프라인(Offline) 설치 관리자에 대한 링크를 선택합니다.

2. 이전 단계에서 다운로드한 설치 스크립트를 실행합니다.

```
$ sudo bash installation_script_name.sh
```

3. 설치 관리자에서 수락 및 설치(Accept & install)를 선택합니다.
4. 인텔 개선 프로그램을 읽고 적절한 옵션을 선택한 다음 설치 시작(Begin Installation)을 선택합니다.
5. 설치가 완료되면 [닫기(Close)]를 선택합니다.
6. 기본적으로 인텔 MPI는 임베디드(내부) Libfabric을 사용합니다. 대신 EFA 설치 프로그램과 함께 제공되는 Libfabric을 사용하도록 인텔 MPI를 구성할 수 있습니다. 일반적으로 EFA 설치 프로그램은 인텔 MPI보다 최신 버전의 Libfabric과 함께 제공됩니다. EFA 설치 프로그램과 함께 제공되는 Libfabric이 인텔 MPI보다 성능이 더 좋은 경우도 있습니다. EFA 설치 프로그램과 함께 제공되는 Libfabric을 사용하도록 인텔 MPI를 구성하려면 셸에 따라 다음 중 하나를 수행합니다.

   bash shells

   /home/*username*/.bashrc 및 /home/*username*/.bash_profile에 다음 명령문을 추가합니다.

```
export I_MPI_OFI_LIBRARY_INTERNAL=0
```

   csh and tcsh shells

   /home/*username*/.cshrc에 다음 명령문을 추가합니다.

```
setenv I_MPI_OFI_LIBRARY_INTERNAL 0
```

7. 셸 스크립트에 다음 source 명령을 추가하여 설치 디렉터리에서 vars.sh 스크립트를 가져와 인스턴스가 시작될 때마다 컴파일러 환경을 설정합니다. 셸에 따라 다음 중 하나를 수행합니다.

   bash shells

   /home/*username*/.bashrc 및 /home/*username*/.bash_profile에 다음 명령문을 추가합니다.

```
source /opt/intel/oneapi/mpi/latest/env/vars.sh
```

csh and tcsh shells

/home/*username*/.cshrc에 다음 명령문을 추가합니다.

```
source /opt/intel/oneapi/mpi/latest/env/vars.csh
```

8. 기본적으로 잘못된 구성으로 인해 EFA를 사용할 수 없는 경우 인텔 MPI는 기본적으로 TCP/IP 네트워크 스택을 사용하므로 애플리케이션 성능이 느려질 수 있습니다. I_MPI_OFI_PROVIDER를 efa로 설정하여 이를 방지할 수 있습니다. 그러면 EFA를 사용할 수 없는 경우 인텔 MPI가 다음 오류와 함께 실패합니다.

```
Abort (XXXXXX) on node 0 (rank 0 in comm 0): Fatal error in PMPI_Init: OtherMPI
 error,
MPIR_Init_thread (XXX).......:
MPID_Init (XXXX).............:
MPIDI_OFI_mpi_init_hook (XXXX):
open_fabric (XXXX)...........:
find_provider (XXXX).........:
OFI fi_getinfo() failed (ofi_init.c:2684:find_provider:
```

셸에 따라 다음 중 하나를 수행합니다.

bash shells

/home/*username*/.bashrc 및 /home/*username*/.bash_profile에 다음 명령문을 추가합니다.

```
export I_MPI_OFI_PROVIDER=efa
```

csh and tcsh shells

/home/*username*/.cshrc에 다음 명령문을 추가합니다.

```
setenv I_MPI_OFI_PROVIDER efa
```

9. 기본적으로 인텔 MPI는 디버깅 정보를 인쇄하지 않습니다. 디버깅 정보를 제어하기 위해 다양한 세부 사항 수준을 지정할 수 있습니다. 가능한 값(제공하는 세부 정보의 양순으로)은 0(기본값), 1, 2, 3, 4, 5입니다. 레벨 1 이상은 libfabric version과 libfabric provider를 인쇄합니다. libfabric version을 사용하여 인텔 MPI가 내부 Libfabric을 사용하고 있는지 아니면 EFA 설치 프로그램과 함께 제공되는 Libfabric을 사용하고 있는지 확인합니다. 내부 Libfabric을 사용하는

경우 버전에 impi가 접미사로 붙습니다. libfabric provider를 사용하여 인텔 MPI가 EFA를 사용하고 있는지 아니면 TCP/IP 네트워크를 사용하고 있는지 확인합니다. EFA를 사용하고 있는 경우 값은 efa입니다. TCP/IP를 사용하고 있는 경우 값은 tcp;ofi_rxm입니다.

디버깅 정보를 활성화하려면 셸에 따라 다음 중 하나를 수행합니다.

bash shells

/home/*username*/.bashrc 및 /home/*username*/.bash_profile에 다음 명령문을 추가합니다.

```
export I_MPI_DEBUG=value
```

csh and tcsh shells

/home/*username*/.cshrc에 다음 명령문을 추가합니다.

```
setenv I_MPI_DEBUG value
```

10. 기본적으로 인텔 MPI는 노드 내 통신을 위해 운영 체제의 공유 메모리(shm)를 사용하고 노드 간 통신에만 Libfabric(ofi)을 사용합니다. 일반적으로 이 구성은 최상의 성능을 제공합니다. 그러나 경우에 따라 인텔 MPI shm 패브릭으로 인해 특정 애플리케이션이 무기한 중단될 수 있습니다.

    이 문제를 해결하려면 인텔 MPI가 노드 내 통신과 노드 간 통신 모두에 Libfabric을 사용하도록 합니다. 이렇게 하려면 셸에 따라 다음 중 하나를 수행합니다.

    bash shells

    /home/*username*/.bashrc 및 /home/*username*/.bash_profile에 다음 명령문을 추가합니다.

    ```
    export I_MPI_FABRICS=ofi
    ```

    csh and tcsh shells

    /home/*username*/.cshrc에 다음 명령문을 추가합니다.

    ```
    setenv I_MPI_FABRICS ofi
    ```

> **Note**
>
> EFA Libfabric 제공업체는 노드 내 통신을 위해 운영 체제의 공유 메모리를 사용합니다. 즉, `I_MPI_FABRICS`를 `ofi`로 설정하면 기본 `shm:ofi` 구성과 유사한 성능을 얻을 수 있습니다.

11. 인스턴스에서 로그아웃한 후 다시 로그인합니다.

Intel MPI를 더 이상 사용하지 않으려면 셸 시작 스크립트에서 환경 변수를 제거하세요.

6단계: 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 수행하세요.

- 테스트 목적으로 ptrace 보호를 일시적으로 비활성화하려면 다음 명령을 실행하세요.

```
$ 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.

7단계. 설치 확인

설치 성공 확인

1. 다음 명령을 실행하여 MPI가 성공적으로 설치되었는지 확인:

```
$ which mpicc
```

- Open MPI의 경우 반환되는 경로에 /opt/amazon/가 포함됩니다.
- Intel MPI의 경우 반환되는 경로에 /opt/intel/이 포함됩니다. 예상한 출력이 나오지 않는 경우 Intel MPI vars.sh 스크립트를 소싱했는지 확인합니다.

2. EFA 소프트웨어 구성 요소와 Libfabric 가 성공적으로 설치되었는지 확인하려면 다음 명령을 실행합니다.

```
$ fi_info -p efa -t FI_EP_RDM
```

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
```

8단계: HPC 애플리케이션 설치

임시 인스턴스에 HPC 애플리케이션을 설치합니다. 설치 절차는 HPC 애플리케이션에 따라 다릅니다. 자세한 내용은 Amazon Linux 2 사용 설명서의 AL2 인스턴스의 소프트웨어 관리를 참조하십시오.

> **Note**
>
> 설치 지침은 HPC 애플리케이션의 설명서를 참조하세요.

9단계: EFA 지원 AMI 생성

필수 소프트웨어 구성 요소를 설치한 뒤 AMI를 사용하여 EFA 사용 인스턴스를 재사용하고 시작합니다.

임시 인스턴스에서 AMI를 생성하려면

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

3. 생성한 임시 인스턴스를 선택하고 [작업(Actions)], [이미지(Image)], [이미지 생성(Create image)]을 선택합니다.
4. [이미지 생성(Create image)]에서 다음을 수행합니다.
   a. [이미지 이름(Image name)]에 AMI를 설명하는 이름을 입력합니다.
   b. (선택 사항) [이미지 설명(Image description)]에 AMI의 용도에 대한 간략한 설명을 입력합니다.
   c. 이미지 생성(Create image)을 선택합니다.
5. 탐색 창에서 AMI를 선택합니다.
6. 목록에서 생성한 AMI를 찾습니다. 상태가 `pending`에서 `available`로 바뀔 때까지 기다린 후 다음 단계를 계속합니다.

10단계: 클러스터 배치 그룹으로 EFA 지원 인스턴스 시작

7단계에서 생성한 EFA이 지원되는 AMI를 사용한 클러스터 배치 그룹과 1단계에서 생성한 EFA를 지원하는 보안 그룹에 EFA를 사용한 인스턴스를 시작합니다.

> **Note**
>
> - 클러스터 placementgroup으로 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 인스턴스를 확장할 때 용량을 사용할 수 있도록 하려면 클러스터 배치 그룹에 대한 용량 예약을 생성할 수 있습니다. 자세한 내용은 클러스터 배치 그룹에 용량 예약 생성 섹션을 참조하세요.

인스턴스 시작

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances(인스턴스)를 선택한 다음에 Launch Instances(인스턴스 시작)를 선택하여 새 인스턴스 시작 마법사를 엽니다.
3. (선택 사항)이름 및 태그(Name and tags) 섹션에서 인스턴스의 이름(예: `EFA-instance`)을 제공합니다. 이름은 인스턴스에 리소스 태그(`Name=`*`EFA-instance`*)로 할당됩니다.

4. 애플리케이션 및 OS 이미지(Application and OS Images) 섹션에서 내 AMI(My AMIs)를 선택한 다음에 이전 단계에서 생성한 AMI를 선택합니다.
5. 인스턴스 유형(Instance type) 섹션에서 지원되는 인스턴스 유형(supported instance type)을 선택합니다.
6. 키 페어(Key pair) 섹션에서 인스턴스에 사용할 키 페어를 선택합니다.
7. 네트워크 설정(Network settings) 섹션에서 편집(Edit)을 선택하고 다음과 같이 수행합니다.

   a. 서브넷(Subnet)에서 인스턴스를 시작할 서브넷을 선택합니다. 서브넷을 선택하지 않으면 EFA에 대해 인스턴스를 활성화할 수 없습니다.

   b. 방화벽(보안 그룹)의 경우 기존 보안 그룹 선택(Select existing security group)을 선택한 다음에 이전 단계에서 생성한 보안 그룹을 선택합니다.

   c. 고급 네트워크 구성 섹션을 확장합니다.

      네트워크 인터페이스 1에서 네트워크 카드 인덱스 = 0, 디바이스 인덱스 = 0 및 인터페이스 유형 = ENA 포함 EFA를 선택합니다.

      (선택 사항) `p4d.24xlarge` 또는 `p5.48xlarge`와 같은 다중 카드 인스턴스 유형을 사용하는 경우, 필요한 추가 네트워크 인터페이스마다 네트워크 인터페이스 추가를 선택하고 네트워크 카드 인덱스에서 다음 미사용 인덱스를 선택한 다음, 디바이스 인덱스 = 1 및 인터페이스 유형 = ENA 포함 EFA 또는 EFA 전용을 선택합니다.
8. (선택 사항) 스토리지(Storage) 섹션에서 필요에 따라 볼륨을 구성합니다.
9. 고급 세부 정보(Advanced details) 섹션에서 배치 그룹 이름(Placement group name)의 경우 인스턴스를 시작할 클러스터 배치 그룹을 선택합니다. 새 클러스터 배치 그룹을 생성해야 하는 경우 새 배치 그룹 생성(Create new placement group)을 선택합니다.
10. 오른쪽의 요약(Summary) 패널에서 인스턴스 개수(Number of instances)의 경우 시작하려는 EFA 사용 인스턴스 개수를 입력한 다음에 인스턴스 시작(Launch instance)을 선택합니다.

11단계: 임시 인스턴스 종료

이 단계에서는 2단계에서 시작한 인스턴스가 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. 생성한 임시 인스턴스를 선택한 다음 작업, 인스턴스 상태, 인스턴스 종료(삭제)를 선택합니다.
4. 확인 메시지가 표시되면 종료(삭제)를 선택합니다.

12단계: 암호 없는 SSH 활성화

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 ~/.ssh/id_rsa
   ```

   키 페어는 `$HOME/.ssh/` 디렉터리에 생성됩니다.
4. 리더 노드에서 프라이빗 키의 사용 권한을 변경합니다.

   ```
   $ chmod 600 ~/.ssh/id_rsa
   chmod 600 ~/.ssh/config
   ```

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.

Amazon EC2에서 ML 워크로드를 위한 EFA 및 NCCL 시작하기

NVIDIA Collective Communications Library(NCCL)는 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 참조하세요.

다음 단계는 지원되는 운영 체제 중 하나의 기본 AMI를 사용하여 EFA와 NCCL을 시작하는 데 도움이 됩니다.

> **Note**
>
> - `p3dn.24xlarge`, `p4d.24xlarge` 및 `p5.48xlarge` 인스턴스 유형만 지원됩니다.
> - Amazon Linux 2 및 Ubuntu 22.04 기본 AMI만 지원됩니다.
> - NCCL 2.4.2 이상만 EFA에서 지원됩니다.
> - AWS Deep Learning AMIs를 사용하여 EFA 및 NCCL로 기계 학습 워크로드를 실행하는 방법에 대한 자세한 내용은 AWS Deep Learning AMIs 개발자 안내서의 Using EFA on the DLAMI를 참조하세요.

단계

- 1단계: EFA를 사용한 보안 그룹 준비
- 2단계: 임시 인스턴스 시작
- 3단계: Nvidia GPU 드라이버, Nvidia CUDA 툴킷 및 cuDNN 설치
- 4단계: GDRCopy 설치
- 5단계: EFA 소프트웨어 설치
- 6단계: NCCL 설치
- 7단계: NCCL 테스트 설치
- 8단계: EFA 및 NCCL 구성 테스트
- 9단계: 기계 학습 애플리케이션 설치

- 10단계: EFA 및 NCCL 지원 AMI 생성
- 11단계: 임시 인스턴스 종료
- 12단계: 클러스터 배치 그룹에 EFA 및 NCCL 지원 인스턴스 시작
- 13단계: 암호 없는 SSH 사용

1단계: EFA를 사용한 보안 그룹 준비

EFA에는 보안 그룹 자체 내의 모든 인바운드 및 아웃바운드 트래픽을 허용하는 보안 그룹이 필요합니다. 다음 절차에서는 자체적으로 들어오고 나가는 모든 인바운드 및 아웃바운드 트래픽을 허용하고 SSH 연결을 위해 모든 IPv4 주소의 인바운드 SSH 트래픽을 허용하는 보안 그룹을 생성합니다.

> **Important**
>
> 이 보안 그룹은 테스트 목적으로만 사용됩니다. 프로덕션 환경에서는 컴퓨터의 IP 주소 또는 로컬 네트워크의 IP 주소 범위와 같이 연결하려는 IP 주소로부터의 트래픽만 허용하는 인바운드 SSH 규칙을 생성하는 것이 좋습니다.

다른 시나리오는 다양한 사용 사례에 대한 보안 그룹 규칙 섹션을 참조하세요.

EFA 사용 보안 그룹을 생성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 보안 그룹(Security Groups)을 선택한 다음, 보안 그룹 생성(Create security group)을 선택합니다.
3. 보안 그룹 생성(Create security group) 창에서 다음을 수행합니다.
   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 EFA 사용 인스턴스를 시작하려는 VPC를 선택합니다.
   d. 보안 그룹 생성을 선택합니다.
4. 생성한 보안 그룹을 선택하고 세부 정보(Details) 탭에서 보안 그룹 ID(Security group ID)를 복사합니다.
5. 보안 그룹을 선택한 상태에서 작업(Actions), 인바운드 규칙 편집(Edit inbound rules)을 선택한 후 다음을 수행합니다.

a. [Add another rule]을 선택합니다.
b. 유형(Type)에서 모든 트래픽(All traffic)을 선택합니다.
c. 소스 유형(Source type)에서 사용자 지정(Custom)을 선택하고 복사한 보안 그룹 ID를 필드에 붙여넣습니다.
d. 규칙 추가를 선택합니다.
e. Type(유형)에서 SSH를 선택합니다.
f. 소스 유형(Source type)에서 어디서나 - IPv4(Anywhere - IPv4)를 선택합니다.
g. 규칙 저장을 선택합니다.

6. 보안 그룹을 선택한 상태에서 작업(Actions), 아웃바운드 규칙 편집(Edit outbound rules)을 선택한 후 다음을 수행합니다.

a. [Add another rule]을 선택합니다.
b. 유형(Type)에서 모든 트래픽(All traffic)을 선택합니다.
c. 대상 유형(Destination type)에서 사용자 지정(Custom)을 선택하고 복사한 보안 그룹 ID를 필드에 붙여넣습니다.
d. 규칙 저장을 선택합니다.

2단계: 임시 인스턴스 시작

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. 탐색 창에서 Instances(인스턴스)를 선택한 다음에 Launch Instances(인스턴스 시작)를 선택하여 새 인스턴스 시작 마법사를 엽니다.
3. (선택 사항)이름 및 태그(Name and tags) 섹션에서 인스턴스의 이름(예: `EFA-instance`)을 제공합니다. 이름은 인스턴스에 리소스 태그(`Name=`*`EFA-instance`*)로 할당됩니다.
4. Application and OS Images(애플리케이션 및 OS 이미지) 섹션에서 지원되는 운영 체제 중 하나에 해당하는 AMI를 선택합니다. Amazon Linux 2 및 Ubuntu 22.04만 지원됩니다.
5. 인스턴스 유형 섹션에서 `p3dn.24xlarge`, `p4d.24xlarge` 또는 `p5.48xlarge`를 선택합니다.
6. 키 페어(Key pair) 섹션에서 인스턴스에 사용할 키 페어를 선택합니다.
7. 네트워크 설정(Network settings) 섹션에서 편집(Edit)을 선택하고 다음과 같이 수행합니다.

a. 서브넷(Subnet)에서 인스턴스를 시작할 서브넷을 선택합니다. 서브넷을 선택하지 않으면 EFA에 대해 인스턴스를 활성화할 수 없습니다.

b. 방화벽(보안 그룹)의 경우 기존 보안 그룹 선택(Select existing security group)을 선택한 다음에 이전 단계에서 생성한 보안 그룹을 선택합니다.

c. 고급 네트워크 구성 섹션을 확장합니다.

네트워크 인터페이스 1에서 네트워크 카드 인덱스 = 0, 디바이스 인덱스 = 0 및 인터페이스 유형 = ENA 포함 EFA를 선택합니다.

(선택 사항) `p4d.24xlarge` 또는 `p5.48xlarge`와 같은 다중 카드 인스턴스 유형을 사용하는 경우, 필요한 추가 네트워크 인터페이스마다 네트워크 인터페이스 추가를 선택하고 네트워크 카드 인덱스에서 다음 미사용 인덱스를 선택한 다음, 디바이스 인덱스 = 1 및 인터페이스 유형 = ENA 포함 EFA 또는 EFA 전용을 선택합니다.

8. 스토리지(Storage) 섹션에서 필요에 따라 볼륨을 구성합니다.

> **Note**
>
> Nvidia CUDA 도구 키트에 사용할 10~20GiB의 스토리지를 추가로 프로비저닝해야 합니다. 스토리지를 충분히 프로비저닝하지 않으면 Nvidia 드라이버 및 CUDA 도구 키트를 설치하려고 할 때 `insufficient disk space`오류가 발생합니다.

9. 오른쪽의 요약(Summary) 패널에서 인스턴스 시작(Launch instance)을 선택합니다.

3단계: Nvidia GPU 드라이버, Nvidia CUDA 툴킷 및 cuDNN 설치

Amazon Linux 2

Nvidia GPU 드라이버 및 Nvidia CUDA 툴킷 및 cuDNN을 설치하려면

1. 모든 소프트웨어 패키지가 최신 상태로 업데이트되어 있는지 확인하기 위해, 인스턴스에서 퀵 소프트웨어 업데이트를 실행합니다.

```
$ sudo yum upgrade -y && sudo reboot
```

인스턴스가 재부팅되면 다시 연결합니다.

2. Nvidia GPU 드라이버 및 Nvidia CUDA 도구 키트를 설치하는 데 필요한 유틸리티를 설치합니다.

```
$ sudo yum groupinstall 'Development Tools' -y
```

3. nouveau 오픈 소스 드라이버를 사용 중단합니다.

   a. 현재 실행 중인 커널의 버전에 필요한 유틸리티와 커널 헤더 패키지를 설치합니다.

   ```
   $ sudo yum install -y wget kernel-devel-$(uname -r) kernel-headers-$(uname -
   r)
   ```

   b. nouveau를 /etc/modprobe.d/blacklist.conf 거부 목록 파일에 추가합니다.

   ```
   $ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
   blacklist vga16fb
   blacklist nouveau
   blacklist rivafb
   blacklist nvidiafb
   blacklist rivatv
   EOF
   ```

   c. grub 파일에 GRUB_CMDLINE_LINUX="rdblacklist=nouveau"를 추가하고 Grub 구성을 다시 구축합니다.

   ```
   $ echo 'GRUB_CMDLINE_LINUX="rdblacklist=nouveau"' | sudo tee -a /etc/
   default/grub \
   && sudo grub2-mkconfig -o /boot/grub2/grub.cfg
   ```

4. 인스턴스를 재부팅하고 다시 연결합니다.

5. 필요한 리포지토리 준비

   a. EPEL 리포지토리를 활성화하고 배포를 rhel7으로 설정합니다.

   ```
   $ sudo amazon-linux-extras install epel \
   && distribution='rhel7'
   ```

   b. CUDA 네트워크 리포지토리를 설정하고 리포지토리 캐시를 업데이트합니다.

   ```
   $ ARCH=$( /bin/arch ) \
   && sudo yum-config-manager --add-repo http://developer.download.nvidia.com/
   compute/cuda/repos/$distribution/${ARCH}/cuda-$distribution.repo \
   && sudo yum clean expire-cache
   ```

c. (커널 버전 5.10에만 해당) 커널 버전 5.10과 함께 Amazon Linux 2를 사용하는 경우에만 이 단계를 수행합니다. 커널 버전 4.12와 함께 Amazon Linux 2를 사용하는 경우 다음 단계를 건너뜁니다. 커널 버전을 확인하려면 uname -r을 실행합니다.

i. /etc/dkms/nvidia.conf라는 이름의 Nvidia 드라이버 구성 파일을 생성합니다.

```
$ sudo mkdir -p /etc/dkms \
&& echo "MAKE[0]=\"'make' -j2 module SYSSRC=\${kernel_source_dir}
 IGNORE_XEN_PRESENCE=1 IGNORE_PREEMPT_RT_PRESENCE=1 IGNORE_CC_MISMATCH=1
 CC=/usr/bin/gcc10-gcc\"" | sudo tee /etc/dkms/nvidia.conf
```

ii. (p4d.24xlarge 및 p5.48xlarge만 해당) Nvidia 드라이버 구성 파일을 복사합니다.

```
$ sudo cp /etc/dkms/nvidia.conf /etc/dkms/nvidia-open.conf
```

6. Nvidia GPU 드라이버, Nvidia CUDA 도구 키트 및 cuDNN을 설치합니다.

```
$ sudo yum clean all \
&& sudo yum -y install nvidia-driver-latest-dkms \
&& sudo yum -y install cuda-drivers-fabricmanager cuda libcudnn8-devel
```

7. 인스턴스를 재부팅하고 다시 연결합니다.

8. (p4d.24xlarge 및 p5.48xlarge만 해당) Nvidia Fabric Manager 서비스를 시작하고 인스턴스가 시작될 때 자동으로 시작되는지 확인합니다. Nvidia 패브릭 관리자는 NV 스위치 관리에 필요합니다.

```
$ sudo systemctl enable nvidia-fabricmanager && sudo systemctl start nvidia-
fabricmanager
```

9. 인스턴스가 시작될 때마다 CUDA 경로가 설정되는지 확인합니다.

- bash 셸의 경우 /home/*username*/.bashrc 및 /home/*username*/.bash_profile에 다음 명령문을 추가합니다.

```
export PATH=/usr/local/cuda/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/
lib64:$LD_LIBRARY_PATH
```

- tcsh 셸의 경우 /home/*username*/.cshrc에 다음 명령문을 추가합니다.

```
setenv PATH=/usr/local/cuda/bin:$PATH
setenv LD_LIBRARY_PATH=/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/
lib64:$LD_LIBRARY_PATH
```

10. 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.

Ubuntu 22.04

Nvidia GPU 드라이버 및 Nvidia CUDA 툴킷 및 cuDNN을 설치하려면

1. 모든 소프트웨어 패키지가 최신 상태로 업데이트되어 있는지 확인하기 위해, 인스턴스에서 퀵 소프트웨어 업데이트를 실행합니다.

```
$ sudo apt-get update && sudo apt-get upgrade -y
```

2. Nvidia GPU 드라이버 및 Nvidia CUDA 도구 키트를 설치하는 데 필요한 유틸리티를 설치합니다.

```
$ sudo apt-get update && sudo apt-get install build-essential -y
```

3. Nvidia GPU 드라이버를 사용하려면, 우선 nouveau 오픈 소스 드라이버를 비활성화해야 합니다.

   a. 현재 실행 중인 커널의 버전에 필요한 유틸리티와 커널 헤더 패키지를 설치합니다.

```
$ sudo apt-get install -y gcc make linux-headers-$(uname -r)
```

   b. nouveau를 `/etc/modprobe.d/blacklist.conf` 거부 목록 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/grub를 열고 다음을 추가합니다.

```
GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
```

d. Grub 구성을 다시 빌드합니다.

```
$ sudo update-grub
```

4. 인스턴스를 재부팅하고 다시 연결합니다.
5. CUDA 리포지토리를 추가하고 Nvidia GPU 드라이버, NVIDIA CUDA 도구 키트 및 cuDNN을 설치합니다.

- p3dn.24xlarge

```
$ sudo apt-key adv --fetch-keys http://developer.download.nvidia.com/compute/
machine-learning/repos/ubuntu2004/x86_64/7fa2af80.pub \
&& wget -O /tmp/deeplearning.deb http://developer.download.nvidia.com/compute/
machine-learning/repos/ubuntu2004/x86_64/nvidia-machine-learning-repo-
ubuntu2004_1.0.0-1_amd64.deb \
&& sudo dpkg -i /tmp/deeplearning.deb \
&& wget -O /tmp/cuda.pin https://developer.download.nvidia.com/compute/cuda/
repos/ubuntu2004/x86_64/cuda-ubuntu2004.pin \
&& sudo mv /tmp/cuda.pin /etc/apt/preferences.d/cuda-repository-pin-600 \
&& sudo apt-key adv --fetch-keys https://developer.download.nvidia.com/
compute/cuda/repos/ubuntu2004/x86_64/3bf863cc.pub \
&& sudo add-apt-repository 'deb http://developer.download.nvidia.com/compute/
cuda/repos/ubuntu2004/x86_64/ /' \
&& sudo apt update \
&& sudo apt install nvidia-dkms-535 \
&& sudo apt install -o Dpkg::Options::='--force-overwrite' cuda-drivers-535
 cuda-toolkit-12-3 libcudnn8 libcudnn8-dev -y
```

- p4d.24xlarge 및 p5.48xlarge

```
$ sudo apt-key adv --fetch-keys http://developer.download.nvidia.com/compute/
machine-learning/repos/ubuntu2004/x86_64/7fa2af80.pub \
&& wget -O /tmp/deeplearning.deb http://developer.download.nvidia.com/compute/
machine-learning/repos/ubuntu2004/x86_64/nvidia-machine-learning-repo-
ubuntu2004_1.0.0-1_amd64.deb \
```

```
&& sudo dpkg -i /tmp/deeplearning.deb \
&& wget -O /tmp/cuda.pin https://developer.download.nvidia.com/compute/cuda/
repos/ubuntu2004/x86_64/cuda-ubuntu2004.pin \
&& sudo mv /tmp/cuda.pin /etc/apt/preferences.d/cuda-repository-pin-600 \
&& sudo apt-key adv --fetch-keys https://developer.download.nvidia.com/
compute/cuda/repos/ubuntu2004/x86_64/3bf863cc.pub \
&& sudo add-apt-repository 'deb http://developer.download.nvidia.com/compute/
cuda/repos/ubuntu2004/x86_64/ /' \
&& sudo apt update \
&& sudo apt install nvidia-kernel-open-535 \
&& sudo apt install -o Dpkg::Options::='--force-overwrite' cuda-drivers-535
 cuda-toolkit-12-3 libcudnn8 libcudnn8-dev -y
```

6. 인스턴스를 재부팅하고 다시 연결합니다.
7. (p4d.24xlarge 및 p5.48xlarge만 해당) Nvidia Fabric Manager를 설치합니다.

   a. 이전 단계에서 설치한 Nvidia 커널 모듈의 버전과 일치하는 Nvidia Fabric Manager 버전을 설치해야 합니다.

      다음 명령을 실행하여 Nvidia 커널 모듈의 버전을 확인합니다.

      ```
      $ cat /proc/driver/nvidia/version | grep "Kernel Module"
      ```

      출력의 예시는 다음과 같습니다.

      ```
      NVRM version: NVIDIA UNIX x86_64 Kernel Module  450.42.01  Tue Jun 15
       21:26:37 UTC 2021
      ```

      위 예제에서 커널 모듈의 메이저 버전 450이 설치되었습니다. 즉, Nvidia Fabric Manager 버전 450을 설치해야 합니다.

   b. Nvidia Fabric Manager를 설치합니다. 다음 명령을 실행하고 이전 단계에서 식별된 메이저 버전을 지정합니다.

      ```
      $ sudo apt install -o Dpkg::Options::='--force-overwrite' nvidia-
      fabricmanager-major_version_number
      ```

      예를 들어, 커널 모듈의 메이저 버전 450이 설치된 경우 다음 명령을 사용하여 일치하는 버전의 Nvidia Fabric Manager를 설치합니다.

```
$ sudo apt install -o Dpkg::Options::='--force-overwrite' nvidia-
fabricmanager-450
```

c. 서비스를 시작하고 인스턴스가 시작될 때 서비스가 자동으로 시작되는지 확인합니다. Nvidia 패브릭 관리자는 NV 스위치 관리에 필요합니다.

```
$ sudo systemctl start nvidia-fabricmanager && sudo systemctl enable nvidia-
fabricmanager
```

8. 인스턴스가 시작될 때마다 CUDA 경로가 설정되는지 확인합니다.

   - bash 셸의 경우 /home/*username*/.bashrc 및 /home/*username*/.bash_profile에 다음 명령문을 추가합니다.

```
export PATH=/usr/local/cuda/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/
lib64:$LD_LIBRARY_PATH
```

   - tcsh 셸의 경우 /home/*username*/.cshrc에 다음 명령문을 추가합니다.

```
setenv PATH=/usr/local/cuda/bin:$PATH
setenv LD_LIBRARY_PATH=/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/
lib64:$LD_LIBRARY_PATH
```

9. 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.

4단계: GDRCopy 설치

GDRCopy를 설치하여 Libfabric의 성능을 향상합니다. GDRCopy에 관한 자세한 내용은 GDRCopy 리포지토리를 참조하세요.

Amazon Linux 2

GDRCopy 설치 방법

1. 필요한 종속 항목을 설치합니다.

```
$ sudo yum -y install dkms rpm-build make check check-devel subunit subunit-
devel
```

2. GDRCopy 패키지를 다운로드하여 압축을 풉니다.

```
$ wget https://github.com/NVIDIA/gdrcopy/archive/refs/tags/v2.4.tar.gz \
&& tar xf v2.4.tar.gz ; cd gdrcopy-2.4/packages
```

3. GDRCopy RPM 패키지를 빌드합니다.

```
$ CUDA=/usr/local/cuda ./build-rpm-packages.sh
```

4. GDRCopy RPM 패키지를 설치합니다.

```
$ sudo rpm -Uvh gdrcopy-kmod-2.4-1dkms.noarch*.rpm \
&& sudo rpm -Uvh gdrcopy-2.4-1.x86_64*.rpm \
&& sudo rpm -Uvh gdrcopy-devel-2.4-1.noarch*.rpm
```

Ubuntu 22.04

GDRCopy 설치 방법

1. 필요한 종속 항목을 설치합니다.

```
$ sudo apt -y install build-essential devscripts debhelper check libsubunit-dev
 fakeroot pkg-config dkms
```

2. GDRCopy 패키지를 다운로드하여 압축을 풉니다.

```
$ wget https://github.com/NVIDIA/gdrcopy/archive/refs/tags/v2.4.tar.gz \
&& tar xf v2.4.tar.gz \
&& cd gdrcopy-2.4/packages
```

3. GDRCopy RPM 패키지를 빌드합니다.

```
$ CUDA=/usr/local/cuda ./build-deb-packages.sh
```

4. GDRCopy RPM 패키지를 설치합니다.

```
$ sudo dpkg -i gdrdrv-dkms_2.4-1_amd64.*.deb \
&& sudo dpkg -i libgdrapi_2.4-1_amd64.*.deb \
&& sudo dpkg -i gdrcopy-tests_2.4-1_amd64.*.deb \
&& sudo dpkg -i gdrcopy_2.4-1_amd64.*.deb
```

5단계: EFA 소프트웨어 설치

인스턴스에서 EFA를 지원하는 데 필요한 EFA 사용 커널, EFA 드라이버, Libfabric, aws-ofi-nccl 플러그인 및 Open MPI 스택을 설치합니다.

EFA 소프트웨어를 설치하려면

1. 시작한 인스턴스에 연결합니다. 자세한 내용은 SSH를 사용하여 Linux 인스턴스에 연결 섹션을 참조하세요.
2. EFA 소프트웨어 설치 파일을 다운로드합니다. 소프트웨어 설치 파일은 압축 tarball(`.tar.gz`) 파일로 패키징되어 있습니다. 최신 안정 버전을 다운로드하려면 다음 명령을 사용하십시오.

   또한 위의 명령에서 버전 번호를 `latest`로 바꾸면 최신 버전을 얻을 수 있습니다.

```
$ curl -O https://efa-installer.amazonaws.com/aws-efa-installer-1.42.0.tar.gz
```

3. (선택 사항) EFA tarball(`.tar.gz`) 파일의 신뢰성 및 무결성을 확인합니다.

   이 작업을 수행하여 소프트웨어 게시자의 자격 증명을 확인하고 파일이 게시된 이후 변경되거나 손상되지 않았는지 확인하는 것이 좋습니다. tarball 파일을 확인하지 않으려면 이 단계를 건너뜁니다.

> **Note**
>
> 또는 MD5 또는 SHA256 체크섬을 대신 사용하여 tarball 파일을 확인하려면 체크섬을 사용하여 EFA 설치 프로그램 확인 섹션을 참조하세요.

   a. 퍼블릭 GPG 키를 다운로드하고 키 링으로 가져옵니다.

```
$ wget https://efa-installer.amazonaws.com/aws-efa-installer.key && gpg --
import aws-efa-installer.key
```

명령에서 키 값이 반환됩니다. 다음 단계에서 필요하므로 키 값을 기록해 둡니다.

b. GPG 키의 지문을 확인합니다. 다음 명령을 실행하고 이전 단계의 키 값을 지정합니다.

```
$ gpg --fingerprint key_value
```

명령에서 4E90 91BC BB97 A96B 26B1 5E59 A054 80B1 DD2D 3CCC와 동일한 지문이 반환되어야 합니다. 지문이 일치하지 않으면 EFA 설치 스크립트를 실행하지 말고 지원에 문의하세요.

c. 서명 파일을 다운로드하고 EFA tarball 파일의 서명을 확인합니다.

```
$ wget https://efa-installer.amazonaws.com/aws-efa-installer-1.42.0.tar.gz.sig
 && gpg --verify ./aws-efa-installer-1.42.0.tar.gz.sig
```

다음은 출력의 예입니다.

```
gpg: Signature made Wed 29 Jul 2020 12:50:13 AM UTC using RSA key ID DD2D3CCC
gpg: Good signature from "Amazon EC2 EFA <ec2-efa-maintainers@amazon.com>"
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: 4E90 91BC BB97 A96B 26B1  5E59 A054 80B1 DD2D 3CCC
```

결과에 Good signature가 있고 이전 단계에서 반환된 지문과 지문이 일치하면 다음 단계로 진행합니다. 그렇지 않은 경우 EFA 설치 스크립트를 실행하지 말고 지원에 문의하세요.

4. 압축 .tar.gz 파일에서 파일을 추출하고 추출된 디렉터리로 이동하십시오.

```
$ tar -xf aws-efa-installer-1.42.0.tar.gz && cd aws-efa-installer
```

5. EFA 소프트웨어 설치 스크립트를 실행합니다.

> **Note**
>
> EFA 1.30.0부터 Open MPI 4.1 및 Open MPI 5 모두 기본적으로 설치됩니다. Open MPI 5가 필요하지 않은 경우 Open MPI 4.1만 설치하는 것이 좋습니다. 다음 명령은 Open MPI

4.1만 설치합니다. Open MPI 4.1 및 Open MPI 5를 설치하려면 `--mpi=openmpi4`를 제거합니다.

```
$ sudo ./efa_installer.sh -y --mpi=openmpi4
```

Libfabric은 `/opt/amazon/efa` 디렉터리에 설치됩니다. aws-ofi-nccl 플러그인은 `/opt/amazon/ofi-nccl` 디렉터리에 설치됩니다. Open MPI는 `/opt/amazon/openmpi` 디렉터리에 설치됩니다.

6. EFA 설치 프로그램에서 인스턴스를 재부팅하라는 메시지를 표시하면 인스턴스를 재부팅한 다음 인스턴스에 다시 연결합니다. 그렇게 하지 않으면 인스턴스에서 로그아웃한 다음 다시 로그인하여 설치를 완료합니다.
7. EFA 소프트웨어 구성 요소가 성공적으로 설치되었는지 확인합니다.

```
$ fi_info -p efa -t FI_EP_RDM
```

이 명령은 Libfabric EFA 인스턴스에 대한 정보를 반환합니다. 다음은 해당 명령 출력을 보여주는 예제입니다.

- `p3dn.24xlarge` 및 단일 네트워크 인터페이스

```
provider: efa
fabric: EFA-fe80::94:3dff:fe89:1b70
domain: efa_0-rdm
version: 2.0
type: FI_EP_RDM
protocol: FI_PROTO_EFA
```

- `p4d.24xlarge` 및 `p5.48xlarge`와 여러 네트워크 인터페이스

```
provider: efa
fabric: EFA-fe80::c6e:8fff:fef6:e7ff
domain: efa_0-rdm
version: 111.0
type: FI_EP_RDM
protocol: FI_PROTO_EFA
provider: efa
fabric: EFA-fe80::c34:3eff:feb2:3c35
domain: efa_1-rdm
```

```
version: 111.0
type: FI_EP_RDM
protocol: FI_PROTO_EFA
provider: efa
fabric: EFA-fe80::c0f:7bff:fe68:a775
domain: efa_2-rdm
version: 111.0
type: FI_EP_RDM
protocol: FI_PROTO_EFA
provider: efa
fabric: EFA-fe80::ca7:b0ff:fea6:5e99
domain: efa_3-rdm
version: 111.0
type: FI_EP_RDM
protocol: FI_PROTO_EFA
```

6단계: NCCL 설치

NCCL을 설치합니다. NCCL에 대한 자세한 내용은 NCCL 리포지토리를 참조하세요.

NCCL을 설치하려면

1. /opt 디렉터리로 이동합니다.

```
$ cd /opt
```

2. 공식 NCCL 리포지토리를 인스턴스에 복제하고 복제된 로컬 리포지토리로 이동합니다.

```
$ sudo git clone https://github.com/NVIDIA/nccl.git -b v2.23.4-1 && cd nccl
```

3. NCCL을 빌드 및 설치하고 CUDA 설치 디렉터리를 지정합니다.

```
$ sudo make -j src.build CUDA_HOME=/usr/local/cuda
```

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 참조하세요.

NCCL 테스트를 설치하려면

1. 홈 디렉터리로 이동합니다.

```
$ cd $HOME
```

2. 공식 nccl-tests 리포지토리를 인스턴스에 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 2

```
$ export LD_LIBRARY_PATH=/opt/amazon/efa/lib64:$LD_LIBRARY_PATH
```

   - Ubuntu

```
$ export LD_LIBRARY_PATH=/opt/amazon/efa/lib:$LD_LIBRARY_PATH
```

4. NCCL 테스트를 설치하고 MPI, NCCL 및 CUDA 설치 디렉터리를 지정합니다.

```
$ make MPI=1 MPI_HOME=/opt/amazon/openmpi NCCL_HOME=/opt/nccl/build CUDA_HOME=/usr/
local/cuda
```

8단계: EFA 및 NCCL 구성 테스트

테스트를 실행하여 임시 인스턴스가 EFA 및 NCCL에 대해 올바르게 구성되었는지 확인합니다.

EFA 및 NCCL 구성을 테스트하려면

1. 테스트를 실행할 호스트를 지정하는 호스트 파일을 생성합니다. 다음 명령은 인스턴스 자체에 대한 참조를 포함하는 my-hosts라는 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_EFA_USE_DEVICE_RDMA=1 - (p4d.24xlarge만 해당) 단방향 및 양방향 전송을 위해 디바이스의 RDMA 기능을 사용합니다.
   - NCCL_DEBUG=INFO - 세부 디버깅 출력을 사용합니다. 또한 VERSION을 지정하여 테스트 시작 시 NCCL 버전만 인쇄하거나 WARN을 지정하여 오류 메시지만 수신할 수 있습니다.

   NCCL 테스트 알고리즘에 대한 자세한 내용은 공식 nccl-tests 리포지토리에서 NCCL Tests README를 참조하세요.

   - p3dn.24xlarge

   ```
   $ /opt/amazon/openmpi/bin/mpirun \
   -x LD_LIBRARY_PATH=/opt/nccl/build/lib:/usr/local/cuda/lib64:/opt/amazon/efa/lib:/opt/amazon/openmpi/lib:/opt/amazon/ofi-nccl/lib:$LD_LIBRARY_PATH \
   -x NCCL_DEBUG=INFO \
   --hostfile my-hosts -n 8 -N 8 \
   --mca pml ^cm --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
   ```

   - p4d.24xlarge 및 p5.48xlarge

   ```
   $ /opt/amazon/openmpi/bin/mpirun \
   -x FI_EFA_USE_DEVICE_RDMA=1 \
   -x LD_LIBRARY_PATH=/opt/nccl/build/lib:/usr/local/cuda/lib64:/opt/amazon/efa/lib:/opt/amazon/openmpi/lib:/opt/amazon/ofi-nccl/lib:$LD_LIBRARY_PATH \
   -x NCCL_DEBUG=INFO \
   --hostfile my-hosts -n 8 -N 8 \
   --mca pml ^cm --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
   ```

3. NCCL_DEBUG 로그가 인쇄될 때 EFA가 NCCL의 기본 공급자로 활성화되어 있는지 확인할 수 있습니다.

```
ip-192-168-2-54:14:14 [0] NCCL INFO NET/OFI Selected Provider is efa*
```

p4d.24xlarge 인스턴스를 사용할 때 다음과 같은 추가 정보가 표시됩니다.

```
ip-192-168-2-54:14:14 [0] NCCL INFO NET/OFI Running on P4d platform, Setting
 NCCL_TOPO_FILE environment variable to /home/ec2-user/install/plugin/share/aws-
ofi-nccl/xml/p4d-24xl-topo.xml
```

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 내용은 Amazon Linux 2 인스턴스에서 소프트웨어 관리를 참조하세요.

> **Note**
>
> 설치 지침은 기계 학습 애플리케이션의 설명서를 참조하세요.

10단계: EFA 및 NCCL 지원 AMI 생성

필수 소프트웨어 구성 요소를 설치한 뒤 AMI를 사용하여 EFA 사용 인스턴스를 재사용하고 시작합니다.

임시 인스턴스에서 AMI를 생성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 생성한 임시 인스턴스를 선택하고 [작업(Actions)], [이미지(Image)], [이미지 생성(Create image)]을 선택합니다.
4. [이미지 생성(Create image)]에서 다음을 수행합니다.
   a. [이미지 이름(Image name)]에 AMI를 설명하는 이름을 입력합니다.
   b. (선택 사항) [이미지 설명(Image description)]에 AMI의 용도에 대한 간략한 설명을 입력합니다.

c. 이미지 생성(Create image)을 선택합니다.

5. 탐색 창에서 AMI를 선택합니다.
6. 목록에서 생성한 AMI를 찾습니다. 상태가 `pending`에서 `available`로 바뀔 때까지 기다린 후 다음 단계를 계속합니다.

11단계: 임시 인스턴스 종료

이 단계에서는 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. 생성한 임시 인스턴스를 선택하고 작업(Actions), 인스턴스 상태(Instance state), 인스턴스 종료(Terminate instance)를 선택합니다.
4. 확인 메시지가 나타나면 종료(Terminate)를 선택합니다.

12단계: 클러스터 배치 그룹에 EFA 및 NCCL 지원 인스턴스 시작

앞서 생성한 EFA 사용 AMI 및 EFA 사용 보안 그룹을 사용하여 클러스터 배치 그룹으로 EFA 및 NCCL 사용 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 인스턴스를 확장할 때 용량을 사용할 수 있도록 하려면 클러스터 배치 그룹에 대한 용량 예약을 생성할 수 있습니다. 자세한 내용은 클러스터 배치 그룹에 용량 예약 생성 섹션을 참조하세요.

New console

임시 인스턴스를 실행합니다.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances(인스턴스)를 선택한 다음에 Launch Instances(인스턴스 시작)를 선택하여 새 인스턴스 시작 마법사를 엽니다.
3. (선택 사항)이름 및 태그(Name and tags) 섹션에서 인스턴스의 이름(예: `EFA-instance`)을 제공합니다. 이름은 인스턴스에 리소스 태그(`Name=`*`EFA-instance`*)로 할당됩니다.
4. 애플리케이션 및 OS 이미지(Application and OS Images) 섹션에서 내 AMI(My AMIs)를 선택한 다음에 이전 단계에서 생성한 AMI를 선택합니다.
5. 인스턴스 유형(Instance type) 섹션에서.`p3dn.24xlarge` 또는 `p4d.24xlarge`를 선택합니다
6. 키 페어(Key pair) 섹션에서 인스턴스에 사용할 키 페어를 선택합니다.
7. 네트워크 설정(Network settings) 섹션에서 편집(Edit)을 선택하고 다음과 같이 수행합니다.

   a. 서브넷(Subnet)에서 인스턴스를 시작할 서브넷을 선택합니다. 서브넷을 선택하지 않으면 EFA에 대해 인스턴스를 활성화할 수 없습니다.

   b. 방화벽(보안 그룹)의 경우 기존 보안 그룹 선택(Select existing security group)을 선택한 다음에 이전 단계에서 생성한 보안 그룹을 선택합니다.

   c. 고급 네트워크 구성 섹션을 확장합니다.

      네트워크 인터페이스 1에서 네트워크 카드 인덱스 = 0, 디바이스 인덱스 = 0 및 인터페이스 유형 = ENA 포함 EFA를 선택합니다.

      (선택 사항) `p4d.24xlarge` 또는 `p5.48xlarge`와 같은 다중 카드 인스턴스 유형을 사용하는 경우, 필요한 추가 네트워크 인터페이스마다 네트워크 인터페이스 추가를 선택하고 네트워크 카드 인덱스에서 다음 미사용 인덱스를 선택한 다음, 디바이스 인덱스 = 1 및 인터페이스 유형 = ENA 포함 EFA 또는 EFA 전용을 선택합니다.
8. (선택 사항) 스토리지(Storage) 섹션에서 필요에 따라 볼륨을 구성합니다.
9. 고급 세부 정보(Advanced details) 섹션에서 배치 그룹 이름(Placement group name)의 경우 인스턴스를 시작할 클러스터 배치 그룹을 선택합니다. 새 클러스터 배치 그룹을 생성해야 하는 경우 새 배치 그룹 생성(Create new placement group)을 선택합니다.
10. 오른쪽의 요약(Summary) 패널에서 인스턴스 개수(Number of instances)의 경우 시작하려는 EFA 사용 인스턴스 개수를 입력한 다음에 인스턴스 시작(Launch instance)을 선택합니다.

Old console

클러스터 배치 그룹으로 EFA 및 NCCL 사용 인스턴스를 시작하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 인스턴스 시작을 선택합니다.
3. [AMI 선택(Choose an AMI)] 페이지에서 [내 AMI(My AMIs)]를 선택하고 앞서 생성한 AMI를 찾은 다음 [선택(Select)]을 선택합니다.
4. 인스턴스 유형 선택 페이지에서 p3dn.24xlarge와 다음: 인스턴스 세부 정보 구성을 차례대로 선택합니다.
5. [Configure Instance Details] 페이지에서 다음을 수행합니다.
   a. [인스턴스 수(Number of instances)]에 시작하려는 EFA 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. 다음: 스토리지 추가를 선택합니다.
6. [스토리지 추가(Add Storage)] 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 시작(Review and Launch)]를 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 ~/.ssh/id_rsa
   ```

   키 페어는 `$HOME/.ssh/` 디렉터리에 생성됩니다.
4. 리더 노드에서 프라이빗 키의 사용 권한을 변경합니다.

   ```
   $ chmod 600 ~/.ssh/id_rsa
   chmod 600 ~/.ssh/config
   ```

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.

여러 네트워크 카드를 사용하여 Amazon EC2 인스턴스의 네트워크 대역폭 극대화

EFA를 지원하는 많은 인스턴스 유형에는 여러 네트워크 카드도 있습니다. 자세한 내용은 네트워크 카드 섹션을 참조하세요. 이러한 인스턴스 유형 중 하나에서 EFA를 사용할 계획인 경우 다음과 같은 기본 구성을 권장합니다.

- 기본 네트워크 인터페이스(네트워크 카드 인덱스 0, 디바이스 인덱스 0)에 대해 EFA(ENA 포함 EFA) 인터페이스를 생성합니다. EFA 전용 네트워크 인터페이스를 기본 네트워크 인터페이스로 사용할 수 없습니다.
- 추가 네트워크 인터페이스마다 ENA 대역폭 요구 사항이나 IP 주소 공간과 같이 사용자의 사용 사례에 따라 다음 미사용 네트워크 카드 인덱스, 디바이스 인덱스 1, ENA(ENA 포함 EFA) 또는 EFA 전용 네트워크 인터페이스를 사용합니다. 사용 사례 예시는 P5 인스턴스에 대한 EFA 구성을 참조하세요.

> **Note**
>
> P5 인스턴스는 최대 네트워크 대역폭을 활성화하려면 네트워크 인터페이스를 특정 방식으로 구성해야 합니다. 자세한 내용은 P5 인스턴스에 대한 EFA 구성 섹션을 참조하세요.

다음 예시에서는 이러한 권장 사항에 따라 인스턴스를 시작하는 방법을 보여줍니다.

Instance launch

인스턴스 시작 마법사를 사용하여 인스턴스를 시작 중에 EFA를 지정하려면

1. 네트워크 설정 섹션에서 편집을 선택합니다.
2. 고급 네트워크 구성을 확장합니다.
3. 기본 네트워크 인터페이스(네트워크 인터페이스 1)에서 네트워크 카드 인덱스 = 0, 디바이스 인덱스 = 0 및 인터페이스 유형 = ENA 포함 EFA를 선택합니다.
4. 필요한 추가 네트워크 인터페이스마다 네트워크 인터페이스 추가를 선택합니다. 네트워크 카드 인덱스에서 다음 미사용 인덱스를 선택한 다음 디바이스 인덱스 = 1 및 인터페이스 유형 = ENA 포함 EFA 또는 EFA 전용을 선택합니다.

run-instances 명령을 사용하여 인스턴스 시작 중에 EFA를 지정하려면

--network-interfaces에서 필요한 네트워크 인터페이스의 수를 지정합니다. 기본 네트워크 인터페이스에서 NetworkCardIndex=0, DeviceIndex=0, InterfaceType=efa를 지정합니다. 추가 네트워크 인터페이스에서 NetworkCardIndex에 대해 다음 미사용 인덱스, DeviceIndex=1, InterfaceType=efa 또는 efa-only을 지정합니다.

다음 예시 명령 조각은 32개의 EFA 디바이스와 1개의 ENA 디바이스가 포함된 요청을 보여줍니다.

```
$ aws --region $REGION ec2 run-instances \
 --instance-type p5.48xlarge \
 --count 1 \
 --key-name key_pair_name \
 --image-id ami-0abcdef1234567890 \
 --network-interfaces
 "NetworkCardIndex=0,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
 \
"NetworkCardIndex=1,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
efa-only" \
"NetworkCardIndex=2,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
efa-only" \
"NetworkCardIndex=3,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
efa-only" \
"NetworkCardIndex=4,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
efa-only" \
"NetworkCardIndex=5,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
efa-only" \
"NetworkCardIndex=6,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
efa-only" \
"NetworkCardIndex=7,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
efa-only" \
"NetworkCardIndex=8,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
efa-only" \
"NetworkCardIndex=9,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
efa-only" \
"NetworkCardIndex=10,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=11,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=12,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=13,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
```

```
"NetworkCardIndex=14,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=15,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=16,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=17,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=18,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=19,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=20,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=21,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=22,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=23,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=24,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=25,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=26,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=27,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=28,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=29,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=30,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=31,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only"
...
```

Launch templates

Amazon EC2 콘솔을 사용하여 시작 템플릿에 EFA를 추가하려면

1. 네트워크 설정 섹션에서 고급 네트워크 구성을 확장합니다.

2. 기본 네트워크 인터페이스(네트워크 인터페이스 1)를 추가하려면 네트워크 인터페이스 추가를 선택한 다음 네트워크 카드 인덱스 = 0, 디바이스 인덱스 = 0 및 인터페이스 유형 = ENA 포함 EFA를 선택합니다.
3. 추가 네트워크 인터페이스를 추가하려면 네트워크 인터페이스 추가를 선택합니다. 네트워크 카드 인덱스에서 다음 미사용 인덱스를 선택한 다음 디바이스 인덱스 = 1 및 인터페이스 유형 = ENA 포함 EFA 또는 EFA 전용을 선택합니다.

create-launch-template 명령을 사용하여 시작 템플릿에 EFA를 추가하려면

`NetworkInterfaces`에서 필요한 네트워크 인터페이스의 수를 지정합니다. 기본 네트워크 인터페이스에서 `NetworkCardIndex=0`, `DeviceIndex=0`, `InterfaceType=efa`를 지정합니다. 추가 네트워크 인터페이스에서 `NetworkCardIndex`에 대해 다음 미사용 인덱스, `DeviceIndex=1`, `InterfaceType=efa` 또는 `efa-only`을 지정합니다.

다음 코드 조각은 사용 가능한 32개의 네트워크 인터페이스 중 3개의 네트워크 인터페이스를 사용하는 예를 보여줍니다.

```
"NetworkInterfaces":[
{
  "NetworkCardIndex":0,
  "DeviceIndex":0,
  "InterfaceType": "efa",
  "AssociatePublicIpAddress":false,
  "Groups":[
    "security_group_id"
  ],
  "DeleteOnTermination":true
},
{
  "NetworkCardIndex": 1,
  "DeviceIndex": 1,
  "InterfaceType": "efa|efa-only",
  "AssociatePublicIpAddress":false,
  "Groups":[
    "security_group_id"
  ],
  "DeleteOnTermination":true
},
{
  "NetworkCardIndex": 2,
  "DeviceIndex": 1,
```

```
  "InterfaceType": "efa|efa-only",
  "AssociatePublicIpAddress":false,
  "Groups":[
    "security_group_id"
  ],
  "DeleteOnTermination":true
},
{
  "NetworkCardIndex": 3,
  "DeviceIndex": 1,
  "InterfaceType": "efa|efa-only",
  "AssociatePublicIpAddress":false,
  "Groups":[
    "security_group_id"
  ],
  "DeleteOnTermination":true
}
...
```

P5 인스턴스에 대한 EFA 구성

P5 인스턴스의 총 네트워크 대역폭 용량은 3,200Gbps이며, 이 중 최대 800Gbps를 IP 네트워크 트래픽에 활용할 수 있습니다. EFA와 IP 네트워크 트래픽은 동일한 기본 리소스를 공유하므로 한 쪽에서 대역폭을 사용하면 다른 쪽에서 사용할 수 있는 대역폭이 줄어듭니다. 즉, 총 대역폭이 3,200Gbps를 초과하지 않으며 IP 대역폭이 800Gbps를 초과하지 않는 한 EFA 트래픽과 IP 트래픽 간에 네트워크 대역폭을 원하는 대로 조합하여 배포할 수 있습니다.

사용 사례 1: IP 주소 저장 및 잠재적 Linux IP 문제 방지

이 구성은 하나의 프라이빗 IP 주소를 통해 최대 3,200Gbps의 EFA 네트워킹 대역폭과 최대 100Gbps의 IP 네트워킹 대역폭을 제공합니다. 이 구성은 인스턴스에 여러 네트워크 인터페이스가 있는 경우 발생할 수 있는 퍼블릭 IP 주소의 자동 할당 거부 및 IP 라우팅 문제(호스트 이름과 IP 주소 매핑 문제 및 소스 IP 주소 불일치)와 같은 잠재적인 Linux IP 문제를 방지하는 데도 도움이 됩니다. 예를 들어 400Gbps 또는 IP 대역폭을 사용하는 경우 동시에 최대 2,800Gbps의 EFA 대역폭을 구현할 수 있습니다.

- 기본 네트워크 인터페이스(네트워크 카드 인덱스 0, 디바이스 인덱스 0)의 경우 EFA(ENA 포함 EFA) 네트워크 인터페이스를 사용합니다.
- 나머지 네트워크 인터페이스(네트워크 카드 인덱스 1~31, 디바이스 인덱스 1)에는 EFA 전용 네트워크 인터페이스를 사용합니다.

사용 사례 2: 최대 EFA 및 IP 네트워크 대역폭

이 구성은 8개의 프라이빗 IP 주소를 통해 최대 3,200Gbps의 EFA 네트워킹 대역폭과 최대 800Gbps의 IP 네트워킹 대역폭을 제공합니다. 이 구성에서는 퍼블릭 IP 주소를 자동 할당할 수 없습니다. 하지만 시작한 후 인터넷 연결을 위해 기본 네트워크 인터페이스(네트워크 카드 인덱스 0, 디바이스 인덱스 0)에 탄력적 IP 주소를 연결할 수 있습니다.

- 기본 네트워크 인터페이스(네트워크 카드 인덱스 0, 디바이스 인덱스 0)의 경우 EFA(ENA 포함 EFA) 네트워크 인터페이스를 사용합니다.
- 나머지 인터페이스의 경우 다음을 수행합니다.
  - 네트워크 카드 인덱스 1, 2, 3에서 EFA 전용 네트워크 인터페이스를 지정하고 디바이스 인덱스 1을 사용합니다.
  - 다음 네트워크 카드 인덱스 하위 집합 각각에서 하나의 EFA(ENA 포함 EFA) 네트워크 인터페이스와 세 개의 EFA 전용 네트워크 인터페이스를 지정하고 디바이스 인덱스 1을 사용합니다.
    - [4,5,6,7]
    - [8,9,10,11]
    - [12,13,14,15]
    - [16,17,18,19]
    - [20,21,22,23]
    - [24,25,26,27]
    - [28,29,30,31]

다음 예시에서는 이 구성을 보여줍니다.

```
$ aws --region $REGION ec2 run-instances \
 --instance-type p5.48xlarge \
 --count 1 \
 --key-name key_pair_name \
 --image-id ami_id \
 --network-interfaces
 "NetworkCardIndex=0,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
 \
"NetworkCardIndex=1,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=2,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
```

```
"NetworkCardIndex=3,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=4,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
 \
"NetworkCardIndex=5,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=6,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=7,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=8,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
 \
"NetworkCardIndex=9,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=10,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=11,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=12,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
 \
"NetworkCardIndex=13,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=14,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=15,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=16,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
 \
"NetworkCardIndex=17,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=18,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=19,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=20,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
 \
"NetworkCardIndex=21,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=22,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=23,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=24,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
 \
```

```
"NetworkCardIndex=25,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=26,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=27,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=28,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
 \
"NetworkCardIndex=29,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=30,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=31,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only"
...
```

Elastic Fabric Adapter를 생성하여 Amazon EC2 인스턴스에 연결

Amazon EC2의 다른 탄력적 네트워크 인터페이스와 마찬가지로 EFA를 생성하여 Amazon EC2 인스턴스에 연결할 수 있습니다. 그러나 탄력적 네트워크 인터페이스와 달리 EFA는 `running` 상태에서 인스턴스에 연결하거나 연결 해제할 수 없습니다.

고려 사항

- 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. 자세한 내용은 1단계: EFA를 사용한 보안 그룹 준비 섹션을 참조하세요.

  탄력적 네트워크 인터페이스와 연결된 보안 그룹을 변경한 것과 동일한 방법을 사용하여 EFA에 연결된 보안 그룹을 변경할 수 있습니다. 자세한 내용은 the section called “네트워크 인터페이스 속성 수정” 섹션을 참조하세요.
- 탄력적 네트워크 인터페이스에 IP 주소를 할당하는 것과 동일한 방법을 사용하여 EFA(ENA 포함 EFA) 네트워크 인터페이스에 탄력적 IP(IPv4) 및 IPv6 주소를 할당할 수 있습니다. 자세한 내용은 IP 주소 관리를 참조하세요.

  EFA 전용 네트워크 인터페이스에는 IP 주소를 할당할 수 없습니다.

업무

- EFA 생성
- 중지된 인스턴스에 EFA 연결

- 인스턴스를 시작할 때 EFA 연결
- 시작 템플릿에 EFA 추가

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.

Console

EFA(ENA 포함 EFA 또는 EFA 전용) 네트워크 인터페이스를 생성하려면 다음을 수행하세요.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 네트워크 인터페이스(Network Interfaces)를 선택한 후 네트워크 인터페이스 생성(Create network interface)을 선택합니다.
3. 설명에 EFA를 설명하는 이름을 입력합니다.
4. 서브넷에서 EFA를 생성하려는 서브넷을 선택합니다.
5. 인터페이스 유형에서 다음 옵션 중 하나를 선택합니다.
   - ENA 포함 EFA - ENA 디바이스와 EFA 디바이스를 모두 지원하는 네트워크 인터페이스를 생성합니다.
   - EFA 전용 - EFA 디바이스로만 네트워크 인터페이스를 생성합니다.
6. (ENA 포함 EFA에만 해당) 네트워크 인터페이스에 대한 IP 주소와 접두사 할당을 구성합니다. 선택한 서브넷에 따라 할당할 수 있는 IP 주소와 접두사의 유형이 달라집니다. IPv4 전용 서브넷의 경우 IPv4 IP 주소와 접두사만 할당할 수 있습니다. IPv6 전용 서브넷의 경우 IPv6 IP 주소와 접두사만 할당할 수 있습니다. 듀얼 스택 서브넷의 경우 IPv4 및 IPv6 IP 주소와 접두사를 모두 할당할 수 있습니다.

   > **Note**
   > EFA 전용 네트워크 인터페이스에는 IP 주소를 할당할 수 없습니다.

   a. 프라이빗 IPv4 주소 및/또는 IPv6 주소의 경우 자동 할당을 선택하여 Amazon EC2가 선택한 서브넷에서 IP 주소를 자동으로 할당하도록 하거나 사용자 지정을 선택하여 할당할 IP 주소를 수동으로 지정합니다.

b. IPv6 주소를 할당하는 경우 선택적으로 프라이머리 IPv6 IP 할당을 활성화할 수 있습니다. 이렇게 하면 네트워크 인터페이스에 프라이머리 IPv6 글로벌 유니캐스트 주소(GUA)가 할당됩니다. 기본 IPv6 주소를 할당하면 인스턴스나 ENI에 대한 트래픽 중단을 방지할 수 있습니다. 자세한 내용은 IPv6 주소를 참조하세요.

c. IPv4 접두사 위임 및/또는 IPv6 접두사 위임의 경우 자동 할당을 선택하여 Amazon EC2가 서브넷의 CIDR 블록에서 접두사를 자동으로 할당하도록 하거나 사용자 지정을 선택하여 서브넷의 CIDR 블록에서 접두사를 수동으로 지정합니다. 접두사를 지정하면 AWS에서 해당 접두사가 다른 리소스에 이미 할당되지 않았는지 확인합니다. 자세한 내용은 Amazon EC2 네트워크 인터페이스에 대한 접두사 위임 섹션을 참조하세요.

d. (선택 사항) 유휴 연결 추적 제한 시간 설정을 구성합니다. 자세한 내용은 유휴 연결 추적 제한 시간 섹션을 참조하세요.

- TCP 설정 제한 시간 - 설정된 상태의 유휴 TCP 연결에 대한 제한 기간(초)입니다. 최솟값: 60초. 최댓값: 43만 2,000초(5일). 기본값: 43만 2,000초. 권장값: 43만 2,000초 미만.
- UDP 제한 시간 - 단일 방향 또는 단일 요청-응답 트랜잭션의 트래픽만 확인한 유휴 UDP 흐름에 대한 제한 기간(초)입니다. 최솟값: 30초. 최댓값: 60초. 기본값: 30초.
- UDP 스트림 제한 시간 - 둘 이상의 요청-응답 트랜잭션을 확인한 스트림으로 분류된 유휴 UDP 흐름에 대한 제한 기간(초)입니다. 최솟값: 60초. 최댓값: 180초(3분) 기본값: 180초

7. 보안 그룹(Security groups)에서 하나 이상의 보안 그룹을 선택합니다.
8. 네트워크 인터페이스 생성을 선택합니다.

AWS CLI

EFA를 생성하려면

create-network-interface 명령을 사용합니다. `--interface-type`에는 EFA 네트워크 인터페이스의 경우 `efa`, EFA 전용 네트워크 인터페이스의 경우 `efa-only`를 지정합니다.

```
aws ec2 create-network-interface \
    --subnet-id subnet-0abcdef1234567890 \
    --interface-type efa \
    --description "my efa"
```

PowerShell

EFA를 생성하려면

New-EC2NetworkInterface cmdlet을 사용합니다. `-InterfaceType`에는 EFA 네트워크 인터페이스의 경우 `efa`, EFA 전용 네트워크 인터페이스의 경우 `efa-only`를 지정합니다.

```
New-EC2NetworkInterface `
    -SubnetId subnet-0abcdef1234567890 `
    -InterfaceType efa `
    -Description "my efa"
```

중지된 인스턴스에 EFA 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 유형를 참조하세요.

네트워크 인터페이스를 인스턴스에 연결하는 방법을 사용해 EFA를 연결할 수 있습니다. 자세한 내용은 네트워크 인터페이스 연결 섹션을 참조하세요.

인스턴스를 시작할 때 EFA 연결

AWS CLI

인스턴스를 시작할 때 기존 EFA를 연결하려면

run-instances 명령을 `--network-interfaces` 옵션과 함께 사용합니다. 기본 네트워크 인터페이스에 대해 EFA 네트워크 인터페이스와 `NetworkCardIndex=0`, `DeviceIndex=0`을 지정합니다. 여러 EFA 네트워크 인터페이스를 연결하려면 네트워크 대역폭 극대화 섹션을 참조하세요.

```
--network-interfaces "NetworkCardIndex=0, \
    DeviceIndex=0, \
    NetworkInterfaceId=eni-1234567890abcdef0, \
    Groups=sg-1234567890abcdef0, \
    SubnetId=subnet-0abcdef1234567890"
```

인스턴스를 시작할 때 새 EFA를 연결하려면

run-instances 명령을 --network-interfaces 옵션과 함께 사용합니다. 기본 네트워크 인터페이스의 경우 NetworkCardIndex=0, DeviceIndex=0, InterfaceType=efa을 사용합니다. 여러 EFA 네트워크 인터페이스를 연결하는 경우 네트워크 대역폭 극대화를 참조하세요.

```
--network-interfaces "NetworkCardIndex=0, \
    DeviceIndex=0, \
    InterfaceType=efa, \
    Groups=sg-1234567890abcdef0, \
    SubnetId=subnet-0abcdef1234567890"
```

PowerShell

인스턴스를 시작할 때 기존 EFA를 연결하려면

-NetworkInterfaces 파라미터와 함께 New-EC2Instance cmdlet을 사용합니다.

```
-NetworkInterface $networkInterface
```

다음과 같이 네트워크 인터페이스를 정의합니다.

```
$networkInterface = New-Object
 Amazon.EC2.Model.InstanceNetworkInterfaceSpecification
$networkInterface.DeviceIndex = 0
$networkInterface.NetworkInterfaceId = "eni-1234567890abcdef0"
$networkInterface.Groups = @("sg-1234567890abcdef0")
$networkInterface.SubnetId = "subnet-0abcdef1234567890"
```

인스턴스를 시작할 때 새 EFA를 연결하려면

-NetworkInterfaces 파라미터와 함께 New-EC2Instance cmdlet을 사용합니다.

```
-NetworkInterface $networkInterface
```

다음과 같이 네트워크 인터페이스를 정의합니다.

```
$networkInterface = New-Object
 Amazon.EC2.Model.InstanceNetworkInterfaceSpecification
$networkInterface.DeviceIndex = 0
$networkInterface.InterfaceType = "efa"
$networkInterface.Groups = @("sg-1234567890abcdef0")
$networkInterface.SubnetId = "subnet-0abcdef1234567890"
```

시작 템플릿에 EFA 추가

EFA 지원 인스턴스 시작에 필요한 구성 정보가 포함된 시작 템플릿을 생성할 수 있습니다. 시작 템플릿에서 EFA 및 EFA 전용 네트워크 인터페이스를 모두 지정할 수 있습니다. EFA 지원 시작 템플릿을 생성하려면 새 시작 템플릿을 생성하고 지원되는 인스턴스 유형과 EFA 지원 AMI, EFA 지원 보안 그룹을 지정합니다. `NetworkInterfaces`에는 연결할 EFA 네트워크 인터페이스를 지정합니다. 기본 네트워크 인터페이스의 경우 `NetworkCardIndex=0`, `DeviceIndex=0`, `InterfaceType=efa`을 사용합니다. 여러 EFA 네트워크 인터페이스를 연결하는 경우 여러 네트워크 카드를 사용하여 Amazon EC2 인스턴스의 네트워크 대역폭 극대화를 참조하세요.

시작 템플릿을 활용하여 다른 AWS 서비스(예: AWS Batch 또는 AWS ParallelCluster)와 함께 EFA 사용 인스턴스를 시작할 수 있습니다.

시작 템플릿 생성에 대한 자세한 내용은 Amazon EC2 시작 템플릿 생성 섹션을 참조하세요.

Amazon EC2 인스턴스에서 EFA 분리 및 삭제

Amazon EC2의 다른 탄력적 네트워크 인터페이스와 동일한 방식으로 Amazon EC2 인스턴스에서 EFA를 분리하고 삭제할 수 있습니다.

EFA 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 네트워크 인터페이스 분리 섹션을 참조하세요.

EFA 삭제

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 섹션을 참조하세요.

Amazon EC2에서 Elastic Fabric Adapter 모니터링

다음 기능을 사용해 Elastic Fabric Adapter(EFA)의 성능을 모니터링할 수 있습니다.

주제

- Amazon EC2 인스턴스의 EFA 드라이버 메트릭
- Amazon VPC 흐름 로그
- Amazon CloudWatch

Amazon EC2 인스턴스의 EFA 드라이버 메트릭

Elastic Fabric Adapter(EFA) 드라이버는 EFA 인터페이스가 연결된 인스턴스의 여러 메트릭을 게시합니다. 이러한 지표를 사용하여 애플리케이션 성능 문제를 해결하고, 워크로드에 적합한 클러스터 크기를 선택하며, 조정 작업을 사전 예방적으로 계획하고, 애플리케이션을 벤치마킹하여 인스턴스의 가용 EFA 성능을 최대화할지 여부를 결정할 수 있습니다.

주제

- 사용 가능한 EFA 드라이버 메트릭
- 인스턴스의 EFA 드라이버 메트릭을 검색합니다.

사용 가능한 EFA 드라이버 메트릭

EFA 드라이버는 인스턴스에 다음과 같은 지표를 실시간으로 게시합니다. 인스턴스 시작 또는 마지막 드라이버 재설정 이후 연결된 EFA 디바이스에서 전송, 수신 또는 삭제된 오류 및 패킷 또는 바이트의 누적 수를 제공합니다.

지표	설명
`tx_bytes`	전송된 바이트 수입니다. 단위: 바이트
`rx_bytes`	수신한 바이트 수입니다. 단위: 바이트
`tx_pkts`	전송된 패킷 수입니다. 단위: 수
`rx_pkts`	

지표	설명
	수신된 패킷 수입니다. 단위: 수
rx_drops	수신 후 삭제된 패킷 수입니다. 단위: 수
send_bytes	전송 작업을 사용하여 전송된 바이트 수입니다. 단위: 바이트
recv_bytes	전송 작업을 사용하여 수신된 바이트 수입니다. 단위: 바이트
send_wrs	전송 작업을 사용하여 전송된 패킷 수입니다. 단위: 수
recv_wrs	전송 작업을 사용하여 수신된 패킷 수입니다. 단위: 수
rdma_write_wrs	완료된 rdma 쓰기 작업 수입니다. 단위: 수
rdma_read_wrs	완료된 rdma 읽기 작업 수입니다. 단위: 수

지표	설명
rdma_write_bytes	rdma 쓰기 작업 수를 사용하여 다른 인스턴스에 의해 기록된 바이트 수입니다. 단위: 바이트
rdma_read_bytes	rdma 읽기 작업 수를 사용하여 수신된 바이트 수입니다. 단위: 바이트
rdma_write_wr_err	로컬 또는 원격 오류가 발생한 rdma 쓰기 작업 수입니다. 단위: 수
rdma_read_wr_err	로컬 또는 원격 오류가 발생한 rdma 읽기 작업 수입니다. 단위: 수
rdma_read_resp_bytes	rdma 읽기 작업 수에 대한 응답으로 전송된 바이트 수입니다. 단위: 바이트
rdma_write_recv_bytes	rdma 쓰기 작업을 사용하여 수신된 바이트 수입니다. 단위: 바이트

인스턴스의 EFA 드라이버 메트릭을 검색합니다.

rdma-tool 명령줄 도구를 사용하여 다음과 같이 인스턴스에 연결된 모든 EFA 인터페이스의 메트릭을 검색할 수 있습니다.

```
$ rdma -p statistic show
link rdmap0s31/1
    tx_bytes 0
    tx_pkts 0
    rx_bytes 0
```

```
    rx_pkts 0
    rx_drops 0
    send_bytes 0
    send_wrs 0
    recv_bytes 0
    recv_wrs 0
    rdma_read_wrs 0
    rdma_read_bytes 0
    rdma_read_wr_err 0
    rdma_read_resp_bytes 0
    rdma_write_wrs 0
    rdma_write_bytes 0
    rdma_write_wr_err 0
```

또는 다음 명령을 사용하여 sys 파일에서 인스턴스에 연결된 각 EFA 인터페이스의 메트릭을 검색할 수 있습니다.

```
$ more /sys/class/infiniband/device_number/ports/port_number/hw_counters/* | cat
```

예

```
$ more /sys/class/infiniband/rdmap0s31/ports/1/hw_counters/* | cat
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/lifespan
::::::::::::::
12
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_read_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_read_resp_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_read_wr_err
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_read_wrs
::::::::::::::
0
::::::::::::::
```

```
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_write_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_write_recv_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_write_wr_err
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_write_wrs
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/recv_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/recv_wrs
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rx_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rx_drops
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rx_pkts
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/send_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/send_wrs
::::::::::::::
0
::::::::::::::
```

```
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/tx_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/tx_pkts
::::::::::::::
0
```

Amazon VPC 흐름 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 내용은 Amazon VPC 사용 설명서의 VPC 흐름 로그를 참조하세요.

탄력적 네트워크 인터페이스에 플로우 로그를 생성한 것과 동일한 방법으로 EFA에 플로우 로그를 생성할 수 있습니다. 자세한 내용은 Amazon VPC 사용 설명서의 플로우 로그 생성을 참조하세요.

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 EKS 클러스터에서 EFA를 사용하는 경우 CloudWatch 컨테이너 인사이트를 사용하여 EFA를 모니터링할 수 있습니다. 자세한 정보는 Amazon CloudWatch 사용 설명서의 Amazon EKS 및 Kubernetes Container Insights 지표를 참조하세요.

체크섬을 사용하여 EFA 설치 프로그램 확인

선택적으로 MD5 또는 SHA256 체크섬을 사용하여 EFA tarball(`.tar.gz` 파일)을 확인할 수 있습니다. 이 작업을 수행하여 소프트웨어 게시자의 자격 증명을 확인하고 애플리케이션이 게시된 이후 변경되거나 손상되지 않았는지 확인하는 것이 좋습니다.

tarball을 확인하려면

MD5 체크섬에 md5sum 유틸리티를 사용하거나 SHA256 체크섬에 sha256sum 유틸리티를 사용하고 tarball 파일 이름을 지정합니다. tarball 파일을 저장한 디렉터리에서 명령을 실행해야 합니다.

- MD5

```
$  md5sum tarball_filename.tar.gz
```

- SHA256

```
$  sha256sum tarball_filename.tar.gz
```

명령은 다음 형식으로 체크섬 값을 반환해야 합니다.

```
checksum_value tarball_filename.tar.gz
```

명령에서 반환한 체크섬 값을 아래 표에 제공된 체크섬 값과 비교합니다. 체크섬이 일치하면 설치 스크립트를 실행하는 것이 안전합니다. 체크섬이 일치하지 않으면 설치 스크립트를 실행하지 말고 지원에 문의하세요.

예를 들어, 다음 명령은 SHA256 체크섬을 사용하여 EFA 1.9.4 tarball을 확인합니다.

```
$  sha256sum aws-efa-installer-1.9.4.tar.gz
```

출력 예시:

```
1009b5182693490d908ef0ed2c1dd4f813cc310a5d2062ce9619c4c12b5a7f14 aws-efa-
installer-1.9.4.tar.gz
```

다음 표에는 최신 버전의 EFA에 대한 체크섬이 나열되어 있습니다.

버전	체크섬
EFA 1.42.0	MD5: 94b2b1db09da1dde08ec049db1f24370 SHA256: 4114fe612905ee05083ae5cb391a00a012510f3abfecc642d86c9a5ae4be9008

버전	체크섬
EFA 1.41.0	MD5: 086181c3ee3f8da512fc6e1c795e8936 SHA256: 3506354cdfbe31ff552fe75f5d0d9bb7efd29cf79bd99457347d29c751c38f9f
EFA 1.40.0	MD5: f3ec6f73fbeaccba082327507581157c SHA256: 30491b0fe7c3470d4439594538855c981b05fa69862d74f8c05eb9b97912368a
EFA 1.39.0	MD5: c223d5954a85a7fbcd248c942b866e43 SHA256: 2cbc028c03064633bb990782b47c36156637769e2f48704417a9c700a7a32101
EFA 1.38.1	MD5: f112569e828ab65187777f794bab542c SHA256: 83923374afd388b1cfcf4b3a21a2b1ba7cf46a01a587f7b519b8386cb95e4f81
EFA 1.38.0	MD5: 43a2a446b33a2506f40853d55059f1ea SHA256: 4f436954f35ad53754b4d005fd8d0be63de3b4184de41a695b504bdce0fecb22

버전	체크섬
EFA 1.37.0	MD5: 6328070192bae920eca45797ad4c1db1 SHA256: 2584fc3c8bb99f29b3285e275747ff09d67c18e162c2a652e36c976b72154bfb
EFA 1.36.0	MD5: 1bec83180fbffb23452ab6469ca21dfa SHA256: de183f333cfb58aeb7908a67bf9106985ba3ccb7f8638b851d2a0d8dbfacaec4
EFA 1.35.0	MD5: 252f03c978dca5f8e8d9f34e488b256e SHA256: 432b6ad4368ba0cd8b902729d14a908a97be7a3dcc5239422ea994a47f35a5e1
EFA 1.34.0	MD5: 5cd4b28d27a31677c16139b54c9acb45 SHA256: bd68839e741b0afd3ec2e37d50603803cfa7a279c120f0a736cc57c2ff2d7fdc
EFA 1.33.0	MD5: e2f61fccbcaa11e2ccfddd3660522276 SHA256: 0372877b87c6a7337bb7791d255e1053b907d030489fb2c3732ba70069185fce

버전	체크섬
EFA 1.32.0	MD5: db8d65cc028d8d08b5a9f2d88881c1b1 SHA256: 5f7233760be57f6fee6de8c09acbfbf59238de848e06048dc54d156ef578fc66
EFA 1.31.0	MD5: 856352f12bef2ccbadcd75e35aa52aaf SHA256: 943325bd37902a4300ac9e5715163537d56ecb4e7b87b37827c3e547aa1897bf
EFA 1.30.0	MD5: 31f48e1a47fe93ede8ebd273fb747358 SHA256: 876ab9403e07a0c3c91a1a34685a52eced890ae052df94857f6081c5f6c78a0a
EFA 1.29.1	MD5: e1872ca815d752c1d7c2b5c175e52a16 SHA256: 178b263b8c25845b63dc93b25bcdff5870df5204ec509af26f43e8d283488744
EFA 1.29.0	MD5: 39d06a002154d94cd982ed348133f385 SHA256: 836655f87015547e733e7d9f7c760e4e24697f8bbc261bb5f3560abd4206bc36

버전	체크섬
EFA 1.28.0	MD5: 9dc13b7446665822605e66febe074035 SHA256: 2e625d2d6d3e073b5178e8e861891273d896b66d03cb1a32244fd56789f1c435
EFA 1.27.0	MD5: 98bfb515ea3e8d93f554020f3837fa15 SHA256: 1d49a97b0bf8d964d91652a79ac851f2550e33a5bf9d0cf86ec9357ff6579aa3
EFA 1.26.1	MD5: 884e74671fdef4725501f7cd2d451d0c SHA256: c616994c924f54ebfabfab32b7fe8ac56947fae00a0ff453d975e298d174fc96
EFA 1.26.0	MD5: f8839f12ff2e3b9ba09ae8a82b30e663 SHA256: bc1abc1f76e97d204d3755d2a9ca307fc423e51c63141f798c2f15be3715aa11
EFA 1.25.1	MD5: 6d876b894547847a45bb8854d4431f18 SHA256: d2abc553d22b89a4ce92882052c1fa6de450d3a801fe005da718b7d4b9602b06

버전	체크섬
EFA 1.25.0	MD5: 1993836ca749596051da04694ea0d00c SHA256: 98b7b26ce031a2d6a93de2297cc71b03af647194866369ca53b60d82d45ad342
EFA 1.24.1	MD5: 211b249f39d53086f3cb0c07665f4e6f SHA256: 120cfeec233af0955623ac7133b674143329f9561a9a8193e473060f596aec62
EFA 1.24.0	MD5: 7afe0187951e2dd2c9cc4b572e62f924 SHA256: 878623f819a0d9099d76ecd41cf4f569d4c3aac0c9bb7ba9536347c50b6bf88e
EFA 1.23.1	MD5: 22491e114b6ee7160a8290145dca0c28 SHA256: 5ca848d8e0ff4d1571cd443c36f8d27c8cdf2a0c97e9068ebf000c303fc40797
EFA 1.23.0	MD5: 38a6d7c1861f5038dba4e441ca7683ca SHA256: 555d497a60f22e3857fdeb3dfc53aa86d05926023c68c916d15d2dc3df6525bd

버전	체크섬
EFA 1.22.1	MD5: 600c0ad7cdbc06e8e846cb763f92901b SHA256: f90f3d5f59c031b9a964466b5401e86fd0429272408f6c207c3f9048254e9665
EFA 1.22.0	MD5: 8f100c93dc8ab519c2aeb5dab89e98f8 SHA256: f329e7d54a86a03ea51da6ea9a5b68fb354fbae4a57a02f9592e21fce431dc3a
EFA 1.21.0	MD5: 959ccc3a4347461909ec02ed3ba7c372 SHA256: c64e6ca34ccfc3ebe8e82d08899ae8442b3ef552541cf5429c43d11a04333050
EFA 1.20.0	MD5: 7ebfbb8e85f1b94709df4ab3db47913b SHA256: aeefd2681ffd5c4c631d1502867db5b831621d6eb85b61fe3ec80df983d1dcf0
EFA 1.19.0	MD5: 2fd45324953347ec5518da7e3fefa0ec SHA256: 99b77821b9e72c8dea015cc92c96193e8db307deee05b91a58094cc331f16709

버전	체크섬
EFA 1.18.0	MD5: `fc2571a72f5d3c7b7b576ce2de38d91e` SHA256: `acb18a0808aedb9a5e485f1469225b9ac97f21db9af78e4cd6939700debe1cb6`
EFA 1.17.3	MD5: `0517df4a190356ab559235147174cafd` SHA256: `5130998b0d2883bbae189b21ab215ecbc1b01ae0231659a9b4a17b0a33ebc6ca`
EFA 1.17.2	MD5: `a329dedab53c4832df218a24449f4c9a` SHA256: `bca1fdde8b32b00346e175e597ffab32a09a08ee9ab136875fb38283cc4cd099`
EFA 1.17.1	MD5: `733ae2cfc9d14b52017eaf0a2ab6b0ff` SHA256: `f29322640a88ae9279805993cb836276ea240623820848463ca686c8ce02136f`
EFA 1.17.0	MD5: `d430fc841563c11c3805c5f82a4746b1` SHA256: `75ab0cee4fb6bd38889dce313183f5d3a83bd233e0a6ef6205d8352821ea901d`

버전	체크섬
EFA 1.16.0	MD5: 399548d3b0d2e812d74dd67937b696b4 SHA256: cecec36495a1bc6fdc82f97761a541e4fb6c9a3cbf3cfcb145acf25ea5dbd45b
EFA 1.15.2	MD5: 955fea580d5170b05823d51acde7ca21 SHA256: 84df4fbc1b3741b6c073176287789a601a589313accc8e6653434e8d4c20bd49
EFA 1.15.1	MD5: c4610267039f72bbe4e35d7bf53519bc SHA256: be871781a1b9a15fca342a9d169219260069942a8bda7a8ad06d4baeb5e2efd7
EFA 1.15.0	MD5: 9861694e1cc00d884fadac07d22898be SHA256: b329862dd5729d2d098d0507fb486bf859d7c70ce18b61c302982234a3a5c88f
EFA 1.14.1	MD5: 50ba56397d359e57872fde1f74d4168a SHA256: c7b1b48e86fe4b3eaa4299d3600930919c4fe6d88cc6e2c7e4a408a3f16452c7

버전	체크섬
EFA 1.14.0	MD5: 40805e7fd842c36ececb9fd7f921b1ae SHA256: 662d62c12de85116df33780d40e0533ef7dad92709f4f613907475a7a1b60a97
EFA 1.13.0	MD5: c91d16556f4fd53becadbb345828221e SHA256: ad6705eb23a3fce44af3afc0f7643091595653a723ad0374084f4f2b715192e1
EFA 1.12.3	MD5: 818aee81f097918cfaebd724eddea678 SHA256: 2c225321824788b8ca3fbc118207b944cdb096b847e1e0d1d853ef2f0d727172
EFA 1.12.2	MD5: 956bb1fc5ae0d6f0f87d2e481d49fccf SHA256: 083a868a2c212a5a4fcf3e4d732b685ce39cceb3ca7e5d50d0b74e7788d06259
EFA 1.12.1	MD5: f5bfe52779df435188b0a2874d0633ea SHA256: 5665795c2b4f09d5f3f767506d4d4c429695b36d4a17e5758b27f033aee58900

버전	체크섬
EFA 1.12.0	MD5: d6c6b49fafb39b770297e1cc44fe68a6 SHA256: 28256c57e9ecc0b0778b41c1f777a9982b4e8eae782343dfe1246079933dca59
EFA 1.11.2	MD5: 2376cf18d1353a4551e35c33d269c404 SHA256: a25786f98a3628f7f54f7f74ee2b39bc6734ea9374720507d37d3e8bf8ee1371
EFA 1.11.1	MD5: 026b0d9a0a48780cc7406bd51997b1c0 SHA256: 6cb04baf5ffc58ddf319e956b5461289199c8dd805fe216f8f9ab8d102f6d02a
EFA 1.11.0	MD5: 7d9058e010ad65bf2e14259214a36949 SHA256: 7891f6d45ae33e822189511c4ea1d14c9d54d000f6696f97be54e915ce2c9dfa
EFA 1.10.1	MD5: 78521d3d668be22976f46c6fecc7b730 SHA256: 61564582de7320b21de319f532c3a677d26cc46785378eb3b95c636506b9bcb4

버전	체크섬
EFA 1.10.0	MD5: 46f73f5a7afe41b4bb918c81888fefa9 SHA256: 136612f96f2a085a7d98296da0afb6fa807b38142e2fc0c548fa986c41186282
EFA 1.9.5	MD5: 95edb8a209c18ba8d250409846eb6ef4 SHA256: a4343308d7ea4dc943ccc21bcebed913e8868e59bfb2ac93599c61a7c87d7d25
EFA 1.9.4	MD5: f26dd5c350422c1a985e35947fa5aa28 SHA256: 1009b5182693490d908ef0ed2c1dd4f813cc310a5d2062ce9619c4c12b5a7f14
EFA 1.9.3	MD5: 95755765a097802d3e6d5018d1a5d3d6 SHA256: 46ce732d6f3fcc9edf6a6e9f9df0ad136054328e24675567f7029edab90c68f1
EFA 1.8.4	MD5: 85d594c41e831afc6c9305263140457e SHA256: 0d974655a09b213d7859e658965e56dc4f23a0eee2dc44bb41b6d039cc5bab45

Elastic Fabric Adapter 릴리스 정보

다음 표에서는 Elastic Fabric Adapter 소프트웨어의 버전 기록과 변경 로그를 설명합니다.

버전	변경 사항	릴리스 날짜
1.42.0	• efa 드라이버 2.15.3으로 업그레이드 • efa-nv-peermem 1.2.1로 업그레이드 • rdma-core 57.amzn1로 업그레이드 • libfabric 2.1.0amzn3.0으로 업그레이드 • libnccl-ofi 1.15.0으로 업그레이드	2025년 6월 6일
1.41.0	• rdma-core 57.amzn0으로 업그레이드	2025년 5월 16일
1.40.0	• libfabric 2.1.0amzn2.0으로 업그레이드 • rdma-core 56.0으로 업그레이드 • efa-config 1.18로 업그레이드했습니다. • Debian 10에 대한 지원 중단 • libnccl-ofi 1.14.2로 업그레이드 • Ubuntu 및 Amazon Linux 2023용 ARM의 Libfabric에서 CUDA 지원 활성화 • Ubuntu 및 Amazon Linux 2023용 ARM에 libnccl-ofi 추가	2025년 5월 2일
1.39.0	• libfabric 2.1.0으로 업그레이드 • Open MPI 5.0.6으로 업그레이드 • libnccl-ofi 1.14.1로 업그레이드 • efa 드라이버 2.15.0으로 업그레이드	2025년 4월 16일
1.38.1	• `Libfabric 1.22.0amzn5.0` 으로 업데이트	2025년 3월 3일
1.38.0	• `libnccl-ofi 1.13.2-1` 추가	2025년 1월 8일

버전	변경 사항	릴리스 날짜
1.37.0	• `libfabric 1.22.0amzn4.0` 로 업그레이드 • `rdma-core 54.amzn0`로 업그레이드	2024년 11월 18일
1.36.0	• Debian 12에 대한 지원 추가 • `/opt/amazon/efa/bin` 에서 유틸리티 스크립트로 `efa_test.sh` 배포 • efa 드라이버 2.13.0으로 업그레이드 • `libfabric 1.22.0amzn3.0` 로 업그레이드 • `rdma-core 54.0`로 업그레이드 • open mpi 4.1.7로 업그레이드	2024년 11월 7일
1.35.0	• Open MPI 5.0.5로 업그레이드 • PRRTE 3.0.6으로 업그레이드하고 RPM 패키지의 이름을 prrte-aws로 변경 • OpenPMIx RPM 패키지의 이름을 pmix-aws로 변경 • dpkg 및 RPM 빌드에 대한 빌드 옵션 구성 • `Libfabric 1.22.0amzn2.0` 로 업그레이드 • efa 드라이버 2.12.1로 업그레이드	2024년 10월 14일
1.34.0	• CentOS 7 및 RHEL 7에 대한 지원 중단 • `Libfabric 1.22.0amzn1.0` 수집 • efa-config 패키지를 1.17로 업데이트	2024년 8월 6일
1.33.0	• efa 드라이버 2.10.0으로 업그레이드 • `rdma-core 52.0`로 업그레이드 • PRRTE 3.0.5로 업그레이드 • Open MPI 5.0.3으로 업그레이드 • Ubuntu 24.04 LTS에 대한 지원 추가 • OpenPMIx, PRRTE, Open MPI HTML 설명서를 제거했습니다. 사용자는 공식 웹 사이트를 참조해야 합니다. • openSUSE 15.4 이상에 대한 지원 중단	2024년 6월 20일

버전	변경 사항	릴리스 날짜
1.32.0	• efa 드라이버 2.8.0으로 업그레이드 • efa-nv-peermem 드라이버 1.1.1로 업그레이드 • efa-config 패키지를 1.16으로 업그레이드 • efa-profile 패키지를 1.7로 업그레이드 • `rdma-core 50.0`로 업그레이드 • `libfabric 1.21.0amzn1.0` 로 업그레이드 • `fi_pingpong` 포트 선택으로 `efa_test.sh` 향상 • 필요한 경우 시스템에서 최신 rdma-core 설치	2024년 4월 18일
1.31.0	• OpenPMIx 4.2.8로 업그레이드 • PRRTE 3.0.3으로 업그레이드 • Open MPI 5.0.2로 업그레이드하고 MCA 구성 요소를 DSO로 변환. • `Libfabric 1.20.1amzn1.0` 로 업그레이드	2024년 3월 7일
1.30.0	• 기본적으로 `/opt/amazon/pmix` 에 설치된 OpenPMIx 4.2.7 포함 • 기본적으로 `/opt/amazon/prrte` 에 설치된 PRRTE 3.0.2 포함 • 기본적으로 `/opt/amazon/openmpi5` 에 설치된 Open MPI 5.0.0 포함 • efa-profile을 1.6으로 업데이트	2023년 12월
1.29.1	• `libfabric 1.19.0amzn4.0` 수집	2023년 12월
1.29.0	• efa 커널 드라이버 2.6.0 수집 • `libfabric 1.19.0amzn3.0` 수집	2023년 11월
1.28.0	• Open MPI를 4.1.6으로 업그레이드 • `libfabric 1.19.0amzn2.0` 수집	2023년 10월
1.27.0	• 설치 프로그램에 `Libfabric 1.19.0amzn1.0` 추가 • Open MPI 4의 GCC 내장 원자 연산 활성화	2023년 9월

버전	변경 사항	릴리스 날짜
1.26.1	• `efa_test.sh` : 무한 재시도 버그 수정	2023년 9월
1.26.0	• RHEL9 지원 추가 • 여러 배포의 디버그 빌드 수정 • `libfabric 1.18.2amzn1.0` 수집 • `efa_test.sh` : `fi_pingpong` 에 대한 재시도 추가	2023년 9월
1.25.1	• `libfabric 1.18.1amzn1.0` 수집 • EFA 설치 프로그램 스크립트의 제한 시간 업데이트	2023년 9월
1.25.0	• Amazon Linux 2023에 대한 지원 추가 • 설치 후 테스트의 버그 수정	2023년 7월
1.24.1	• libfabric을 1.18.1로 업그레이드 • efa 드라이버를 2.5.0으로 업그레이드	2023년 7월
1.24.0	• `rdma-core 46.0` 수집 • efa 드라이버 2.4.1 수집 • Debian 11 지원	2023년 6월
1.23.1	• `libfabric 1.18.0amzn2.0` 수집	2023년 6월
1.23.0	• Debian 10에 대한 지원 추가 • Ubuntu 18.04 LTS에 대한 지원 중단 • efa-config 패키지를 1.14로 업그레이드 • `libfabric 1.18.0amzn1.0` 수집	2023년 5월
1.22.1	• libfabric을 1.17.1로 업그레이드	2023년 3월
1.22.0	• Open MPI를 4.1.5로 업그레이드 • libfabric을 1.17.0으로 업그레이드 • efa-config 패키지를 1.13으로 업그레이드	2023년 2월

버전	변경 사항	릴리스 날짜
1.21.0	• Rocky Linux 9 OS에 대한 지원 추가 • efa 드라이버 2.1.1 수집 • `libfabric 1.16.1amzn3.0` 수집 • efa-config 패키지를 1.12로 업그레이드	2022년 12월
1.20.0	• Rocky Linux 8 OS에 대한 지원을 추가했습니다. • efa 드라이버 2.1.0을 수집했습니다. • `rdma-core 43.0` 을 수집했습니다. • `libfabric 1.16.1amzn1.0` 수집	2022년 11월
1.19.0	• `libfabric 1.16.0` 수집 • `--enable-orterun-prefix-by-default` 를 사용하여 Open MPI 빌드	2022년 10월
1.18.0	• Ubuntu22.04에 대한 지원 추가	2022년 8월
1.17.3	• libfabric을 `1.16.0~amzn4.0` 으로 업데이트했습니다. ~는 libfabric 1.16.0의 시험판 버전임을 나타냅니다. • 설치 후 pingpong 테스트 제한 시간을 20초로 연장	2022년 8월
1.17.2	• libfabric을 `1.16.0~amzn3.0` 으로 업데이트했습니다. ~는 libfabric 1.16.0의 시험판 버전임을 나타냅니다.	2022년 7월
1.17.1	• libfabric을 `1.16.0~amzn2.0` 으로 업데이트했습니다. ~는 libfabric 1.16.0의 시험판 버전임을 나타냅니다. • libfabric을 빌드할 때 실험 네트워크 공급업체 비활성화	2022년 7월
1.17.0	• `rdma-core` 를 v41.0으로 업데이트 • Open MPI를 4.1.4로 업데이트 • libfabric을 `1.16.0~amzn1.0` 으로 업데이트했습니다. ~는 libfabric 1.16.0의 시험판 버전임을 나타냅니다.	2022년 7월

버전	변경 사항	릴리스 날짜
1.16.0	• libfabric을 `1.15.1amzn1.0` 으로 업데이트, 뉴런 라이브러리 이름 변경 포함 • efa-config를 1.10으로 업그레이드 • libfabric 빌드에서 opx 및 rxd 공급업체 제외	2022년 6월
1.15.2	• libfabric을 1.14.1로 업데이트	2022년 5월
1.15.1	• libfabric을 `1.14.0amzn1.0` 으로 업데이트	2022년 3월
1.15.0	• Open SuSE 15.3에서 설치 실패를 유발하는 버그 수정 • Open SuSE 15.2에 대한 지원 중단(Open SuSE 15.2 수명 종료) • CentOS 8에 대한 지원 중단(CentOS 8 수명 종료) • libfabric을 1.14.0으로 업데이트 • efa 커널 드라이버를 1.16.0으로 업데이트 • `rdma-core` 를 v39.0으로 업데이트 • Open MPI를 버전 4.1.2로 업데이트했습니다.	2022년 2월
1.14.1	• libfabric을 `1.13.2amzn1.0` 으로 업데이트했습니다.	2021년 10월
1.14.0	• efa 커널 드라이버 1.14.2 수집했습니다. • 최신 efa 커널 드라이버가 기본적으로 GDR 지원을 활성화하므로 `efa_installer.sh` 의 `-g, --enable-gdr` 을 no-op 옵션으로 설정했습니다. • `rdma-core v37.0` 을 수집했습니다. • `libfabric 1.13.2` 을 수집했습니다. • 설치 중 알 수 없는 패키지의 설치를 방지하기 위해 패키지 목록을 추가하고 RPM/DEB 목록과 비교하도록 수정했습니다. • EFA 드라이버를 다시 로드한 후 udev 규칙이 적용될 때까지 기다리도록 설치 프로그램 스크립트에 지연을 추가했습니다.	2021년 10월

버전	변경 사항	릴리스 날짜
1.13.0	• `rdma-core` 를 v35.0으로 업데이트했습니다. • `libfabric` 을 v1.13.0amzn1.0으로 업데이트했습니다. • Gravition2 플랫폼의 CentOS/RHEL 8에 대한 EFA 지원을 추가했습니다. • 시스템에 더 높은 버전이 이미 설치되어 있는 경우 로컬 패키지 설치를 건너뛰도록 설치 스크립트에 버전 비교 로직을 추가했습니다.	2021년 8월
1.12.3	• EFA 커널 모듈을 1.13.0으로 업데이트했습니다. • efa-config 패키지를 버전 1.9로 업데이트했습니다. 기본값으로 설정된 대규모의 대용량 페이지 크기를 처리하기 위해 대용량 페이지 예약 계산을 개선했습니다.	2021년 7월
1.12.2	• EFA 커널 모듈을 1.12.3으로 업데이트했습니다. • `--with-libevent=external` 및 `--with-hwloc=external` 을 사용하여 Open MPI Debian 패키지를 빌드했습니다. • Open MPI rpm 빌드 ID를 2로 범프하여 CentOS 8에서 HWLOC의 이전 버전과의 호환성 문제를 해결했습니다. • SLES15SP2 및 openSUSE 15.2에서 커널 개발과 커널 소스 패키지 설치를 제거했습니다.	2021년 6월
1.12.1	• Libfabric을 버전 1.11.2amzon1.1로 업데이트했습니다. • EFA 커널 모듈을 버전 1.12.1로 업데이트햇습니다.	2021년 5월

버전	변경 사항	릴리스 날짜
1.12.0	• Open MPI를 버전 4.1.1로 업데이트했습니다. • Libfabric을 버전 1.11.2amzn1.0으로 업데이트했습니다. • AL2 rdma-core와 동일한 패키징 구성을 사용하여 Amazon Linux 2용 rdma-core를 빌드했습니다. • Open MPI RPM 빌드 시 `-Wl,--enable-new-dtags` 옵션을 강제하지 않도록 변경했습니다. • hwloc 및 libevent용 시스템 라이브러리를 사용하여 Open MPI를 빌드했습니다. • EFA 커널 모듈을 버전 1.12.0으로 업데이트 • efa-config 패키지를 버전 1.8로 업데이트했습니다. 수명이 긴 인스턴스에 대한 방대한 페이지 예약 계산을 개선했습니다. • efa-profile 패키지를 버전 1.5로 업데이트했습니다. P4d에서 발생하던 Open MPI 4.1.0 중단 현상을 해결하기 위해 사용된 Open MPI 집합 연산 튜닝 파일을 제거했습니다. • `rdma-core` 를 v32.1로 업데이트했습니다. • Amazon Linux 1 및 Ubuntu 16.04에 대한 지원을 중단했습니다.	2021년 5월
1.11.2	• 구성 파일을 통해 MPI_BARRIER를 구현하는 데 사용되는 기본 알고리즘을 변경하여 P4d에서 Open MPI를 사용할 때 Open MPI가 중단되는 문제를 해결했습니다. • `--disable-builtin-atomics` 를 통해 ARM의 Open MPI에서 기본 제공 원자 연산 사용을 비활성화하여 컴파일러 문제를 해결했습니다.	2021년 2월
1.11.1	• Open MPI를 버전 4.1.0으로 업데이트했습니다. • efa-config 패키지를 버전 1.7로 업데이트했습니다. 방대한 페이지 예약 수 계산을 개선했습니다. • efa-profile 패키지를 버전 1.3으로 업데이트했습니다. 이제 Open MPI 4.1.0이 사용되므로 불필요한 집합 연산 결정 파일을 제거했습니다.	2020년 12월

버전	변경 사항	릴리스 날짜
1.11.0	• Gravition2 플랫폼에 대한 지원을 추가했습니다. • `rdma-core` 를 버전 31.2amzn으로 업데이트했습니다. • Libfabric을 버전 1.11.1amzn1.0으로 업데이트했습니다. • efa-config를 버전 1.6으로 업데이트했습니다. • efa-profile을 버전 1.2로 업데이트했습니다.	2020년 12월
1.10.1	• CentOS/RHEL 8에 대한 지원을 추가했습니다. • Ubuntu 20.04에 대한 지원을 추가했습니다. • SUSE Linux Enterprise 15에 대한 지원을 추가했습니다.	2020년 11월
1.10.0	• P4d 플랫폼에 대한 GPUDirect RDMA 지원을 추가했습니다. GDR-aware 커널 모듈과 사용자 공간을 설치하기 위해 `--enable-gdr` 설치 프로그램 옵션을 사용하도록 했습니다. • EFA 커널 모듈을 버전 1.10.2로 업데이트했습니다. • `rdma-core` 를 버전 31.amzn0으로 업데이트했습니다. • Libfabric을 버전 1.11.1로 업데이트했습니다. • Open MPI를 버전 4.0.5로 업데이트했습니다. • efa-config를 버전 1.5로 업데이트했습니다. • efa-profile을 버전 1.1로 업데이트했습니다. Open MPI 집합 연산 결정 파일을 개선하여 포함했습니다.	2020년 10월
1.9.5	• efa-config를 버전 1.4로 업데이트했습니다. Open MPI 집합 연산 결정 파일의 버그를 수정했습니다.	2020년 9월
1.9.4	• Open MPI를 버전 4.0.3으로 업데이트했습니다. • Libfabric을 버전 1.10.1amazon1.1로 업데이트했습니다. • `rdma-core` 를 버전 28.amzn0으로 업데이트했습니다.	2020년 7월

버전	변경 사항	릴리스 날짜
1.9.3	• EFA 커널 모듈을 버전 1.6.0으로 업데이트했습니다. • `rdma-core` 를 버전 28.amzn0으로 업데이트했습니다. • Libfabric을 버전 1.10.1amzn1.1로 업데이트했습니다. • efa-config를 버전 1.3으로 업데이트했습니다. Open MPI 집합 연산 튜닝 파일을 추가했습니다. • 이미 설치된 경우 DKMS 설치를 건너뛰도록 변경했습니다. • `--skip-kmod` 설치 모드가 실제로 작동하도록 수정했습니다.	2020년 6월
1.8.4	• 구성 파일을 efa-config 및 efa-profile 패키지로 이동하여 운영 체제의 패키지 관리자가 이를 추적할 수 있도록 변경했습니다. • Open MPI를 버전 4.0.3으로 업데이트했습니다.	2020년 4월
1.8.3	• EFA 커널 모듈을 버전 1.5.1로 업데이트했습니다. • 일부 플랫폼에서 설치 신뢰성을 높이기 위해 EPEL 리포지토리에 의존하는 대신 DKMS를 플랫폼에 직접 배포하도록 변경했습니다. • RHEL 7에서는 Amazon Linux 2에 빌드된 RPM 대신 CentOS 7에서 빌드된 RPM을 설치하도록 변경했습니다.	2020년 2월
1.8.2	• 커널 모듈과 `rdma-core` 간의 디바이스 이름 지정 불일치로 인해 `rdma-core` 를 버전 25로 되돌렸습니다.	2020년 1월
1.8.1	• Libfabric을 버전 1.9.0amzn1.1로 업데이트했습니다.	2020년 1월
1.8.0	• `rdma-core` 를 버전 27.0으로 업데이트했습니다. • EFA 커널 모듈을 버전 1.5.0으로 업데이트했습니다. • Libfabric을 버전 1.9.0amzn1.0으로 업데이트했습니다. • EFA 커널 모듈 및 `rdma-core` 만 설치하는 `--minimal` 옵션을 추가했습니다.	2019년 12월
1.7.1	• Libfabric을 버전 1.8.1amzn1.3으로 업데이트했습니다.	2019년 12월

버전	변경 사항	릴리스 날짜
1.7.0	• Libfabric 모듈 파일을 추가했습니다. • Libfabric을 버전 1.8.1amzn1.1로 업데이트했습니다.	2019년 11월
1.6.2	• Open MPI를 버전 1.6.2로 업데이트했습니다.	2019년 10월
1.6.1	• Libfabric을 버전 1.8.1amzn1.0으로 업데이트했습니다. • Open MPI를 버전 4.0.1로 업데이트했습니다. • `rdma-core` 를 버전 26.0으로 업데이트했습니다.	2019년 10월
1.5.4	• EFA 커널 모듈을 버전 1.4.1로 업데이트했습니다.	2019년 9월
1.5.3	• EFA 커널 모듈을 버전 1.3.1로 업데이트했습니다. • 커널 드라이버를 설치하는 경우가 아니면 `kernel-devel` 또는 `linux-headers` 패키지가 설치되지 않도록 변경했습니다.	2019년 9월
1.5.1	• EFA 디바이스가 있는 경우에만 대용량 페이지를 구성하도록 변경했습니다.	2019년 8월
1.5.0	• Libfabric을 버전 1.8.0amzn1.1로 업데이트했습니다. • `rmda-core` 를 버전 25.0으로 업데이트했습니다.	2019년 8월
1.4.1	• Open MPI 및 Libfabric 공유 라이브러리를 올바르게 찾을 수 있도록 Libfabric 및 Open MPI 라이브러리 경로(`/opt/amazon/efa/lib64` 및 `/opt/amazon/efa/openmpi/lib`)를 `/etc/ld.so.conf.d/efa.conf` 에 추가했습니다.	2019년 7월
1.4.0	• EFA 커널 모듈을 버전 1.3.0으로 업데이트했습니다. • Libfabric을 버전 1.8.0amzn1.0으로 업데이트 • Intel MPI 2019 Update 4를 지원하는 최초 릴리스입니다.	2019년 7월

Amazon EC2 인스턴스 토폴로지

인스턴스 토폴로지를 설명하면 Amazon EC2 인스턴스 간 상대적 근접성의 계층적 보기가 제공됩니다. 이 정보를 사용하여 고성능 컴퓨팅(HPC) 및 기계 학습(ML) 컴퓨팅 인프라를 대규모로 관리하는 동시

에 작업 배치를 최적화할 수 있습니다. HPC 및 ML 작업은 지연 시간과 처리량에 민감합니다. 인스턴스 토폴로지를 사용하여 인스턴스의 위치를 감지한 다음, 이 정보로 물리적으로 서로 더 가까운 인스턴스에서 HPC 및 ML 작업을 실행하여 최적화할 수 있습니다.

인스턴스 토폴로지를 사용하여 기존 인스턴스의 위치를 감지할 수 있지만 기존 인스턴스와 물리적으로 가까운 곳에서 새 인스턴스를 시작하도록 선택하는 데 인스턴스 토폴로지를 사용할 수 없습니다. 인스턴스 배치에 영향을 미치기 위해 클러스터 배치 그룹에서 용량 예약을 생성할 수 있습니다.

고려 사항

- 인스턴스 토폴로지 보기는 `running` 상태의 인스턴스에서만 사용할 수 있습니다.
- 각 인스턴스 토폴로지 뷰는 계정별로 고유합니다.
- AWS Management Console에서는 인스턴스 토폴로지 보기를 지원하지 않습니다.

요금

인스턴스 토폴로지를 설명하는 데 추가 비용은 들지 않습니다.

내용

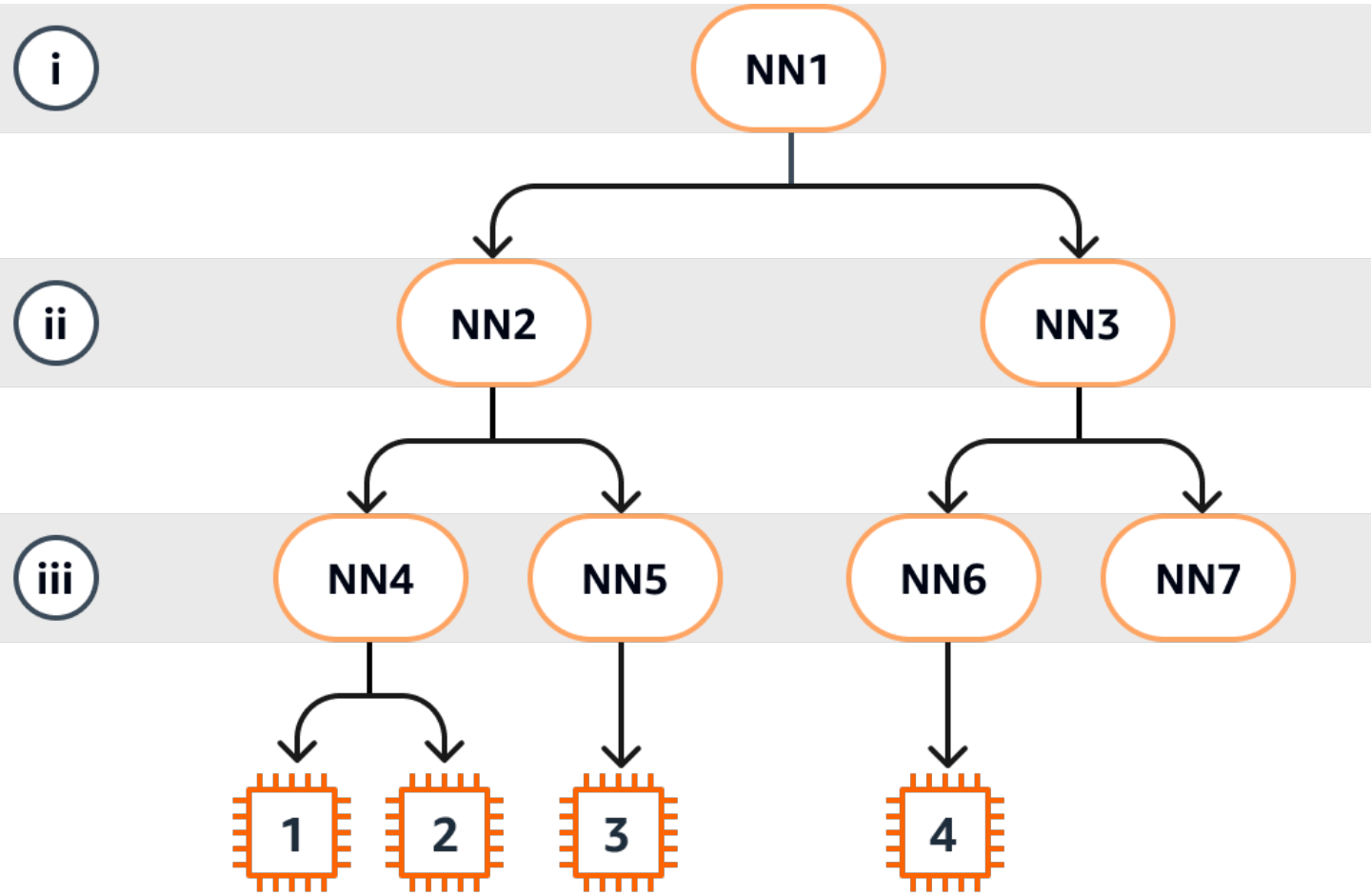
- Amazon EC2 인스턴스 토폴로지 작동 방식
- Amazon EC2 인스턴스 토폴로지의 사전 조건
- Amazon EC2 인스턴스 토폴로지의 예제

Amazon EC2 인스턴스 토폴로지 작동 방식

AWS 네트워크는 계층 구조로 정렬됩니다. EC2 인스턴스는 인스턴스 유형에 따라 세 번째 계층 또는 그 아래의 네트워크에 연결됩니다. 인스턴스의 토폴로지는 네트워크의 각 계층에 하나의 노드가 있는 노드 집합으로 설명됩니다. DescribeInstanceTopology API 응답에 설정된 노드는 인스턴스에 연결된 하단 노드와 함께 네트워크 계층 구조의 하향 보기를 제공합니다.

> **Note**
>
> 일부 인스턴스 유형은 네트워크의 4개 계층을 나타내는 노드 세트의 네트워크 노드 4개로 구성되고, 다른 인스턴스 유형은 네트워크의 3개 계층을 나타내는 네트워크 노드 3개로 구성됩니다. 지원되는 인스턴스 유형은 인스턴스 타입 섹션을 참조하세요.

다음 다이어그램은 인스턴스 토폴로지를 이해하는 데 사용할 수 있는 시각적 표현을 제공합니다. 네트워크 노드는 NN1~NN7로 식별됩니다. 숫자 i, ii, iii는 네트워크 계층을 식별합니다. 숫자 1, 2, 3, 4는 EC2 인스턴스를 식별합니다. 인스턴스는 다음 다이어그램의 iii로 식별되는 하단 계층의 노드에 연결됩니다. 둘 이상의 인스턴스가 동일한 노드에 연결할 수 있습니다.



이 예시에서는 다음이 적용됩니다.

- 인스턴스 1은 계층 iii의 네트워크 노드 4(NN4)에 연결됩니다. 이 예제에서 NN4는 계층 ii의 네트워크 노드 2(NN2)에 연결되고, NN2는 네트워크 계층의 최상위인 계층 i의 네트워크 노드 1(NN1)에 연결됩니다. 네트워크 노드 세트는 상위 계층에서 하위 계층까지 계층적으로 표현되는 NN1, NN2, NN4로 구성됩니다.
- 인스턴스 2는 네트워크 노드 4(NN4)에도 연결됩니다. 인스턴스 1과 인스턴스 2는 동일한 네트워크 노드 세트(NN1, NN2, NN4)를 공유합니다.
- 인스턴스 3은 네트워크 노드 5(NN5)에 연결됩니다. NN5는 NN2에 연결되고 NN2는 NN1에 연결됩니다. 인스턴스 3의 네트워크 노드 세트는 NN1, NN2, NN5입니다.
- 인스턴스 4는 네트워크 노드 6(NN6)에 연결됩니다. 네트워크 노드 세트는 NN1, NN3, NN6입니다.

인스턴스 1, 2, 3의 근접성을 고려할 때 인스턴스 1과 2는 동일한 네트워크 노드(NN4)에 연결되기 때문에 서로 더 가깝고, 인스턴스 3은 다른 네트워크 노드(NN5)에 연결되기 때문에 더 멀리 떨어져 있습니다.

이 다이어그램에 있는 모든 인스턴스의 근접성을 고려할 때 인스턴스 1, 2, 3은 네트워크 노드 세트에서 NN2를 공유하므로 인스턴스 4보다 서로 더 가깝습니다.

일반적으로 두 인스턴스에 연결된 네트워크 노드가 같으면 인스턴스 1과 2의 경우처럼 이러한 인스턴스는 물리적으로 서로 가깝습니다. 또한 네트워크 노드 간 홉 수가 적을수록 인스턴스는 서로 더 가까워집니다. 예를 들어, 인스턴스 1과 인스턴스 3은 인스턴스 4와 공통으로 사용하는 네트워크 노드(NN1)보다 공통 네트워크 노드(NN2)에 대한 홉 수가 적기 때문에 인스턴스 4보다 서로 더 가깝습니다.

이 예제에서는 네트워크 노드 7(NN7)에서 실행되는 인스턴스가 없으므로 API 출력에 NN7이 포함되지 않습니다.

출력을 해석하는 방법

DescribeInstanceTopology API를 사용하여 인스턴스 토폴로지 정보를 가져옵니다. 출력에서는 인스턴스의 기본 네트워크 토폴로지에 대한 계층 보기를 제공합니다.

다음 예제 출력은 위 다이어그램에 있는 4개 인스턴스의 네트워크 토폴로지 정보에 해당합니다. 이 예제의 목적을 위해 예제 출력에 설명이 포함되어 있습니다.

출력의 다음 정보에 유의해야 합니다.

- `NetworkNodes`는 인스턴스의 네트워크 노드 세트를 설명합니다.
- 각 네트워크 노드 세트에서 네트워크 노드는 위에서 아래의 계층적 순서로 나열됩니다.
- 인스턴스에 연결된 네트워크 노드는 목록의 마지막 네트워크 노드(맨 아래 계층)입니다.
- 어떤 인스턴스가 서로 가까운지 알아보려면 먼저 맨 아래 계층에서 공통 네트워크 노드를 찾습니다. 하위 계층에 공통 네트워크 노드가 없는 경우 상위 계층에서 공통 네트워크 노드를 찾습니다.

다음 예제 출력에서 `i-1111111111example`과 `i-2222222222example`은 맨 아래 계층에 공통된 네트워크 노드 `nn-4444444444example`이 있기 때문에 이 예제의 다른 인스턴스에 비해 서로 가장 가깝게 위치합니다.

> **Note**
>
> 응답에는 3개 이상의 네트워크 노드가 포함됩니다. 지원되는 각 인스턴스 유형에 대한 응답의 네트워크 노드 수에 대한 자세한 내용은 인스턴스 타입 섹션을 참조하세요.

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",  //Corresponds to instance 1
            "InstanceType": "p4d.24xlarge",
            "GroupName": "ML-group",
            "NetworkNodes": [
                "nn-1111111111example",           //Corresponds to NN1 in layer i
                "nn-2222222222example",           //Corresponds to NN2 in layer ii
                "nn-4444444444example"            //Corresponds to NN4 in layer iii -
 bottom layer, connected to the instance
            ],
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-2222222222example",  //Corresponds to instance 2
            "InstanceType": "p4d.24xlarge",
            "NetworkNodes": [
                "nn-1111111111example",           //Corresponds to NN1 - layer i
                "nn-2222222222example",           //Corresponds to NN2 - layer ii
                "nn-4444444444example"            //Corresponds to NN4 - layer iii -
 connected to instance
            ],
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-3333333333example",  //Corresponds to instance 3
            "InstanceType": "trn1.32xlarge",
            "NetworkNodes": [
                "nn-1111111111example",           //Corresponds to NN1 - layer i
                "nn-2222222222example",           //Corresponds to NN2 - layer ii
                "nn-5555555555example"            //Corresponds to NN5 - layer iii -
 connected to instance
            ],
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-444444444example",  //Corresponds to instance 4
            "InstanceType": "trn1.2xlarge",
            "NetworkNodes": [
```

```
                "nn-1111111111example",          //Corresponds to NN1 - layer i
                "nn-3333333333example",          //Corresponds to NN3 - layer ii
                "nn-6666666666example"           //Corresponds to NN6 - layer iii -
 connected to instance
            ],
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

Amazon EC2 인스턴스 토폴로지의 사전 조건

인스턴스의 인스턴스 토폴로지를 설명하기 전에 인스턴스가 다음 요구 사항을 충족하는지 확인합니다.

인스턴스 토폴로지를 설명하기 위한 요구 사항

- AWS 리전
- 인스턴스 타입
- 인스턴스 상태
- IAM 권한

AWS 리전

지원되는 AWS 리전:

- 미국 동부(버지니아 북부), 미국 동부(오하이오), 미국 서부(캘리포니아 북부), 미국 서부(오레곤)
- 아시아 태평양(뭄바이), 아시아 태평양(서울), 아시아 태평양(싱가포르), 아시아 태평양(시드니), 아시아 태평양(도쿄)
- 캐나다(중부)
- 유럽(프랑크푸르트), 유럽(아일랜드), 유럽(런던), 유럽(파리), 유럽(스페인), 유럽(스톡홀름)
- 이스라엘(텔아비브)
- 남아메리카(상파울루)

인스턴스 타입

지원되는 인스턴스 유형

- 응답에 네트워크 노드 3개 반환
  - `hpc6a.48xlarge` | `hpc6id.32xlarge` | `hpc7a.12xlarge` | `hpc7a.24xlarge` | `hpc7a.48xlarge` | `hpc7a.96xlarge` | `hpc7g.4xlarge` | `hpc7g.8xlarge` | `hpc7g.16xlarge`
  - `p3dn.24xlarge` | `p4d.24xlarge` | `p4de.24xlarge` | `p5.48xlarge` | `p5e.48xlarge` | `p5en.48xlarge`
  - `trn1.2xlarge` | `trn1.32xlarge` | `trn1n.32xlarge` | `trn2.48xlarge` | `trn2u.48xlarge`
- 응답에 네트워크 노드 4개 반환
  - `p6-b200.48xlarge`

사용 가능한 인스턴스 유형은 리전마다 다릅니다. 자세한 내용은 리전별 Amazon EC2 인스턴스 유형을 참조하세요.

인스턴스 상태

인스턴스는 `running` 상태여야 합니다. 다른 상태에 있는 인스턴스의 인스턴스 토폴로지 정보는 가져올 수 없습니다.

IAM 권한

IAM 자격 증명(사용자, 사용자 그룹, 역할)에는 다음 IAM 권한이 필요합니다.

- `ec2:DescribeInstanceTopology`

Amazon EC2 인스턴스 토폴로지의 예제

describe-instance-topology 명령을 사용하여 EC2 인스턴스의 인스턴스 토폴로지를 설명할 수 있습니다.

파라미터나 필터 없이 `describe-instance-topology` 명령을 사용하면 지정된 리전에서 이 명령에 지원되는 인스턴스 유형과 일치하는 모든 인스턴스가 응답에 포함됩니다. `--region` 파라미터를 포함하거나 기본 리전을 설정하여 리전을 지정할 수 있습니다. 기본 리전 설정에 대한 자세한 내용은 Amazon EC2 리소스에 대한 리전 선택 섹션을 참조하세요.

지정된 인스턴스 ID 또는 배치 그룹 이름과 일치하는 인스턴스를 반환하는 파라미터를 포함할 수 있습니다. 지정된 인스턴스 유형이나 인스턴스 패밀리와 일치하는 인스턴스 또는 지정된 가용 영역 또는 로컬 영역의 인스턴스를 반환하는 필터를 포함할 수도 있습니다. 단일 파라미터 또는 필터나 파라미터와 필터 조합을 포함할 수 있습니다.

출력은 기본적으로 페이지당 최대 20개의 인스턴스로 페이지가 매겨집니다. `--max-results` 파라미터를 사용하여 페이지당 최대 100개의 인스턴스를 지정할 수 있습니다.

자세한 내용은 describe-instance-topology 섹션을 참조하세요.

필요한 권한

인스턴스 토폴로지를 설명하려면 다음 권한이 필요합니다.

- `ec2:DescribeInstanceTopology`

예시

- 예제 1 - 인스턴스 ID
- 예제 2 - 배치 그룹 이름 파라미터
- 예제 3 - 인스턴스 유형 필터
  - 예제 3a - 지정된 인스턴스 유형에 대한 정확히 일치 필터
  - 예제 3b - 인스턴스 패밀리의 와일드카드 필터
  - 예제 3c - 조합된 인스턴스 패밀리와 정확히 일치 필터
- 예제 4 - 영역 ID 필터
  - 예제 4a - 가용 영역 필터
  - 예제 4b - 로컬 영역 필터
  - 예제 4c - 조합된 가용 영역 필터 및 로컬 영역 필터
- 예제 5 - 인스턴스 유형 필터 및 영역 ID 필터

예제 1 - 인스턴스 ID

AWS CLI

특정 인스턴스의 토폴로지를 설명하려면

`--instance-ids` 파라미터와 함께 describe-instance-topology 명령을 사용합니다. 출력에는 지정된 인스턴스 ID와 일치하는 인스턴스만 포함됩니다.

```
aws ec2 describe-instance-topology \
    --region us-west-2 \
    --instance-ids i-1111111111example i-2222222222example
```

출력의 예시는 다음과 같습니다.

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",
            "InstanceType": "p4d.24xlarge",
            "GroupName": "ML-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "GroupName": "HPC-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3214313214example"
            ],
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

특정 인스턴스의 토폴로지를 설명하려면

Get-EC2InstanceTopology cmdlet을 사용합니다.

```
Get-EC2InstanceTopology `
```

```
    -InstanceId i-1111111111example, i-2222222222example
```

예제 2 - 배치 그룹 이름 파라미터

AWS CLI

지정된 배치 그룹에 포함된 인스턴스의 토폴로지를 설명하려면

`group-names` 파라미터와 함께 describe-instance-topology 명령을 사용합니다. 출력에는 지정된 배치 그룹 중 하나에 있는 인스턴스만 포함됩니다.

```
aws ec2 describe-instance-topology \
    --region us-west-2 \
    --group-names ML-group HPC-group
```

출력의 예시는 다음과 같습니다.

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",
            "InstanceType": "p4d.24xlarge",
            "GroupName": "ML-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "GroupName": "HPC-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3214313214example"
            ],
            "ZoneId": "usw2-az2",
```

```
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

지정된 배치 그룹에 포함된 인스턴스의 토폴로지를 설명하려면

Get-EC2InstanceTopology cmdlet을 사용합니다.

```
Get-EC2InstanceTopology `
    -GroupName ML-group, HPC-group
```

예제 3 - 인스턴스 유형 필터

지정된 인스턴스 유형(정확히 일치)을 기준으로 필터링하거나 인스턴스 패밀리(와일드카드 사용)를 기준으로 필터링할 수 있습니다. 지정된 인스턴스 유형 필터와 인스턴스 패밀리 필터를 조합할 수도 있습니다.

예제 3a - 지정된 인스턴스 유형에 대한 정확히 일치 필터

AWS CLI

특정 인스턴스 유형을 사용하여 인스턴스의 토폴로지를 설명하려면

`instance-type` 필터와 함께 describe-instance-topology 명령을 사용합니다. 출력에는 지정된 인스턴스 유형의 인스턴스만 포함됩니다.

```
aws ec2 describe-instance-topology \
    --region us-west-2 \
    --filters Name=instance-type,Values=trn1n.32xlarge
```

출력의 예시는 다음과 같습니다.

```
{
    "Instances": [
        {
            "InstanceId": "i-2222222222example",
```

```
            "InstanceType": "trn1n.32xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

특정 인스턴스 유형을 사용하여 인스턴스의 토폴로지를 설명하려면

Get-EC2InstanceTopology cmdlet을 사용합니다.

```
Get-EC2InstanceTopology `
    -Filter @{Name="instance-type"; Values="trn1n.32xlarge"}
```

예제 3b - 인스턴스 패밀리의 와일드카드 필터

AWS CLI

특정 인스턴스 패밀리를 사용하여 인스턴스의 토폴로지를 설명하려면

`instance-type` 필터와 함께 describe-instance-topology 명령을 사용합니다. 출력에는 지정된 인스턴스 패밀리의 인스턴스만 포함됩니다.

```
aws ec2 describe-instance-topology \
    --region us-west-2 \
    --filters Name=instance-type,Values=trn1*
```

출력의 예시는 다음과 같습니다.

```
{
    "Instances": [
        {
            "InstanceId": "i-2222222222example",
```

```
            "InstanceType": "trn1n.32xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-3333333333example",
            "InstanceType": "trn1.32xlarge",
            "NetworkNodes": [
                "nn-1212121212example",
                "nn-1211122211example",
                "nn-1311133311example"
            ],
            "ZoneId": "usw2-az4",
            "AvailabilityZone": "us-west-2d"
        },
        {
            "InstanceId": "i-444444444example",
            "InstanceType": "trn1.2xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-5434334334example",
                "nn-1235301234example"
            ],
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

특정 인스턴스 패밀리를 사용하여 인스턴스의 토폴로지를 설명하려면

Get-EC2InstanceTopology cmdlet을 사용합니다.

```
Get-EC2InstanceTopology `
    -Filter @{Name="instance-type"; Values="trn1*"}
```

예제 3c - 조합된 인스턴스 패밀리와 정확히 일치 필터

AWS CLI

특정 인스턴스 패밀리 또는 인스턴스 유형을 사용하여 인스턴스의 토폴로지를 설명하려면

`instance-type` 필터와 함께 describe-instance-topology 명령을 사용합니다. 출력에는 지정된 기준을 충족하는 인스턴스만 포함됩니다.

```
aws ec2 describe-instance-topology \
    --region us-west-2 \
    --filters "Name=instance-type,Values=p4d*,trn1n.32xlarge"
```

출력의 예시는 다음과 같습니다.

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",
            "InstanceType": "p4d.24xlarge",
            "GroupName": "ML-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-4343434343example"
            ],
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

특정 인스턴스 패밀리 또는 인스턴스 유형을 사용하여 인스턴스의 토폴로지를 설명하려면

Get-EC2InstanceTopology cmdlet을 사용합니다.

```
Get-EC2InstanceTopology `
    -Filter @{Name="instance-type"; Values="p4d*", "trn1n.32xlarge"}
```

예제 4 - 영역 ID 필터

zone-id 필터를 사용하여 가용 영역 또는 로컬 영역을 기준으로 필터링할 수 있습니다. 가용 영역 필터와 로컬 영역 필터를 조합할 수도 있습니다.

예제 4a - 가용 영역 필터

AWS CLI

특정 가용 영역에 있는 인스턴스의 토폴로지를 설명하려면

zone-id 필터와 함께 describe-instance-topology 명령을 사용합니다. 출력에는 지정된 가용 영역의 인스턴스만 포함됩니다.

```
aws ec2 describe-instance-topology \
    --region us-east-1 \
    --filters Name=zone-id,Values=use1-az1
```

출력의 예시는 다음과 같습니다.

```
{
    "Instances": [
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3214313214example"
            ],
            "ZoneId": "use1-az1",
```

```
            "AvailabilityZone": "us-east-1a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

특정 가용 영역에 있는 인스턴스의 토폴로지를 설명하려면

Get-EC2InstanceTopology cmdlet을 사용합니다.

```
Get-EC2InstanceTopology `
    -Filter @{Name="zone-id"; Values="use1-az1"}
```

예제 4b - 로컬 영역 필터

AWS CLI

특정 로컬 영역에 있는 인스턴스의 토폴로지를 설명하려면

zone-id 필터와 함께 describe-instance-topology 명령을 사용합니다. 출력에는 지정된 로컬 영역의 인스턴스만 포함됩니다.

```
aws ec2 describe-instance-topology \
    --region us-east-1 \
    --filters Name=zone-id,Values=use1-atl2-az1
```

출력의 예시는 다음과 같습니다.

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",
            "InstanceType": "p4d.24xlarge",
            "GroupName": "ML-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
```

```
            ],
            "ZoneId": "use1-atl2-az1",
            "AvailabilityZone": "us-east-1-atl-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

특정 로컬 영역에 있는 인스턴스의 토폴로지를 설명하려면

Get-EC2InstanceTopology cmdlet을 사용합니다.

```
Get-EC2InstanceTopology `
    -Filter @{Name="zone-id"; Values="use1-atl2-az1"}
```

예제 4c - 조합된 가용 영역 필터 및 로컬 영역 필터

AWS CLI

특정 영역에 있는 인스턴스의 토폴로지를 설명하려면

zone-id 필터와 함께 describe-instance-topology 명령을 사용합니다. 출력에는 지정된 영역 중 하나에 있는 인스턴스만 포함됩니다.

```
aws ec2 describe-instance-topology \
    --region us-east-1 \
    --filters Name=zone-id,Values=use1-az1,use1-atl2-az1
```

출력의 예시는 다음과 같습니다.

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",
            "InstanceType": "p4d.24xlarge",
            "GroupName": "ML-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
```

```
                "nn-3333333333example"
            ],
            "ZoneId": "use1-atl2-az1",
            "AvailabilityZone": "us-east-1-atl-2a"
        },
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3214313214example"
            ],
            "ZoneId": "use1-az1",
            "AvailabilityZone": "us-east-1a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

특정 영역에 있는 인스턴스의 토폴로지를 설명하려면

Get-EC2InstanceTopology cmdlet을 사용합니다.

```
Get-EC2InstanceTopology `
    -Filter @{Name="zone-id"; Values="use1-az1", "use1-atl2-az1"}
```

예제 5 - 인스턴스 유형 필터 및 영역 ID 필터

단일 명령으로 필터를 조합할 수 있습니다.

AWS CLI

특정 인스턴스 유형, 인스턴스 패밀리 및 영역을 사용하여 인스턴스의 토폴로지를 설명하려면

`instance-type` 및 `zone-id` 필터와 함께 describe-instance-topology 명령을 사용합니다. 응답에는 지정된 인스턴스 유형에 해당하고 지정된 영역 중 하나에 있는 모든 인스턴스가 포함됩니다.

```
aws ec2 describe-instance-topology \
```

```
    --region us-east-1 \
    --filters "Name=instance-type,Values=p4d*,trn1n.32xlarge" \
              "Name=zone-id,Values=use1-az1,use1-atl2-az1"
```

출력의 예시는 다음과 같습니다.

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",
            "InstanceType": "p4d.24xlarge",
            "GroupName": "ML-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "ZoneId": "use1-atl2-az1",
            "AvailabilityZone": "us-east-1-atl-2a"
        },
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3214313214example"
            ],
            "ZoneId": "use1-az1",
            "AvailabilityZone": "us-east-1a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

특정 인스턴스 유형, 인스턴스 패밀리 및 영역을 사용하여 인스턴스의 토폴로지를 설명하려면

Get-EC2InstanceTopology cmdlet을 사용합니다.

```
Get-EC2InstanceTopology `
    -Filter @{Name="instance-type"; Values="p4d*", "trn1n.32xlarge"} `
```

```
@{Name="zone-id"; Values="use1-az1", "use1-atl2-az1"}
```

Amazon EC2 인스턴스에 대한 배치 그룹

워크로드 요구 사항을 충족하기 위해 상호 의존적인 EC2 인스턴스 그룹을 배치 그룹으로 시작하여 배치에 영향을 미칠 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 고성능 컴퓨팅(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 그룹은 선택 사항입니다. 인스턴스를 배치 그룹으로 시작하지 않으면 EC2는 모든 인스턴스가 기본 하드웨어 전반에 분산되어 상호 관련 오류를 최소화하는 방식으로 인스턴스를 배치하려고 합니다.

요금

배치 그룹 생성은 무료입니다.

규칙 및 제한 사항

배치 그룹을 사용하기 전에 다음 규칙에 유의해야 합니다.

- 인스턴스는 한 번에 하나의 배치 그룹에 배치할 수 있으며, 여러 배치 그룹에 배치할 수 없습니다.
- 여러 배치 그룹을 병합할 수는 없습니다.
- 온디맨드 용량 예약 및 영역 예약 인스턴스는 가용 영역의 EC2 인스턴스에 용량을 예약할 수 있도록 지원합니다. 인스턴스를 시작할 때 인스턴스 속성이 온디맨드 용량 예약 또는 영역 예약 인스턴스에서 지정한 속성과 일치하면 인스턴스는 예약 용량을 자동으로 사용합니다. 이는 인스턴스를 배치 그룹으로 시작하는 경우에도 마찬가지입니다.
- 배치 그룹에서는 전용 호스트를 시작할 수 없습니다.
- 배치 그룹에서는 인터럽트 시 중지하거나 최대 절전 모드로 전환하도록 구성된 스팟 인스턴스를 시작할 수 없습니다.

내용

- 배치 그룹을 위한 배치 전략
- EC2 인스턴스에 대한 배치 그룹 생성
- EC2 인스턴스의 배치 변경
- 배치 그룹 삭제
- 공유된 배치 그룹
- AWS Outposts에서의 배치 그룹

배치 그룹을 위한 배치 전략

다음 배치 전략 중 하나를 사용하여 EC2 인스턴스에 대해 배치 그룹을 생성할 수 있습니다.

배치 전략

- 클러스터 배치 그룹
- 파티션 배치 그룹
- 분산형 배치 그룹

클러스터 배치 그룹

클러스터 배치 그룹은 단일 가용 영역 내에 있는 인스턴스의 논리적 그룹입니다. 인스턴스는 단일 랙에 격리되지 않습니다. 클러스터 배치 그룹은 동일한 리전의 피어링된 가상 프라이빗 네트워크(VPC)에 걸쳐 적용될 수 있습니다. 동일한 클러스터 배치 그룹의 인스턴스는 TCP/IP 트래픽에 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 권장됩니다. 배치 그룹에 가장 짧은 지연 시간과 가장 높은 초당 패킷 네트워크 성능을 제공하려면 향상된 네트워킹을 지원하는 인스턴스 유형을 선택하세요. 자세한 내용은 향상된 네트워킹을 참조하세요.

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 보세요. 인스턴스를 시작하면 요청한 모든 인스턴스를 수용할 용량이 있는 하드웨어로 인스턴스가 마이그레이션될 수 있습니다.

규칙 및 제한 사항

클러스터 배치 그룹에는 다음 규칙이 적용됩니다.

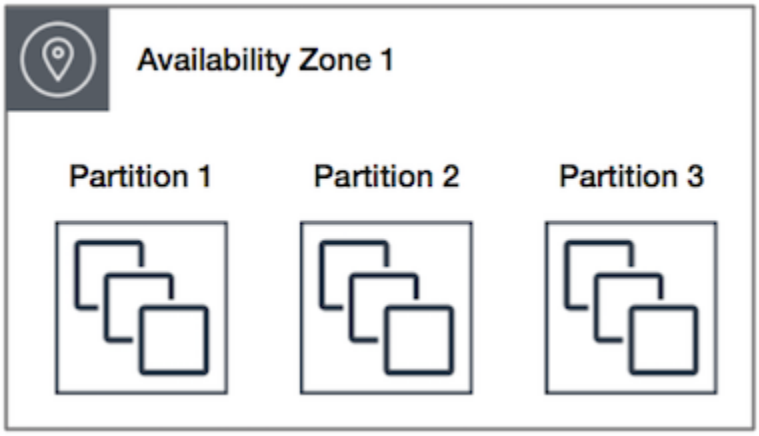
- 다음 인스턴스 유형이 지원됩니다.
  - 성능 버스트 가능 인스턴스(예: T2), Mac1 인스턴스 및 M7i-flex 인스턴스를 제외한 현재 세대 인스턴스.
  - A1, C3, C4, I2, M4, R3, R4와 같은 이전 세대 인스턴스.
- 클러스터 배치 그룹은 여러 가용 영역을 포괄할 수 없습니다.
- 두 인스턴스의 속도가 느려지면 한 클러스터 배치 그룹에 있는 두 인스턴스 간에 트래픽의 최대 네트워크 처리 속도도 느려집니다. 많은 양을 처리해야 하는 애플리케이션의 경우, 요구 사항을 충족하는 네트워크 연결을 지원하는 인스턴스 유형을 선택하세요.
- 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.
- 클러스터 배치 그룹에서 온디맨드 용량 예약을 생성하여 클러스터 배치 그룹에서 명시적으로 용량을 예약하는 것이 좋습니다. 영역별 예약 인스턴스는 배치 그룹에서 명시적으로 용량을 예약할 수 없기 때문에 이를 사용하여 용량을 예약할 수 없습니다.
- 인터넷으로 가는 네트워크 트래픽과 AWS Direct Connect 연결을 통해 온프레미스 리소스로 가는 네트워크 트래픽은 클러스터 배치 그룹의 경우 5Gbps로 제한됩니다.

파티션 배치 그룹

파티션 배치 그룹은 애플리케이션에 대한 상관 관계가 있는 하드웨어 장애 가능성을 줄이는 데 도움이 됩니다. 파티션 배치 그룹을 사용하는 경우 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 사

용자가 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.

파티션 배치 그룹에서 하나의 인스턴스를 시작할 때 요청을 이행하기에 충분한 고유 하드웨어가 없으면 요청이 실패합니다. Amazon EC2는 시간이 지남에 따라 개별 하드웨어를 추가로 제공하므로 나중에 다시 요청을 시도할 수 있습니다.

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.
- 인스턴스가 파티션 배치 그룹으로 시작되는 경우 Amazon EC2는 전체 파티션에 인스턴스를 균일하게 배포하려고 시도합니다. Amazon EC2는 전체 파티션에 걸친 인스턴스의 균일한 배포를 보장하지 않습니다.
- 전용 인스턴스가 있는 파티션 배치 그룹은 파티션을 최대 2개까지 가질 수 있습니다.
- 용량 예약은 파티션 배치 그룹의 용량을 예약하지 않습니다.

분산형 배치 그룹

분산형 배치 그룹은 각각 고유한 하드웨어에 배치된 인스턴스 그룹입니다.

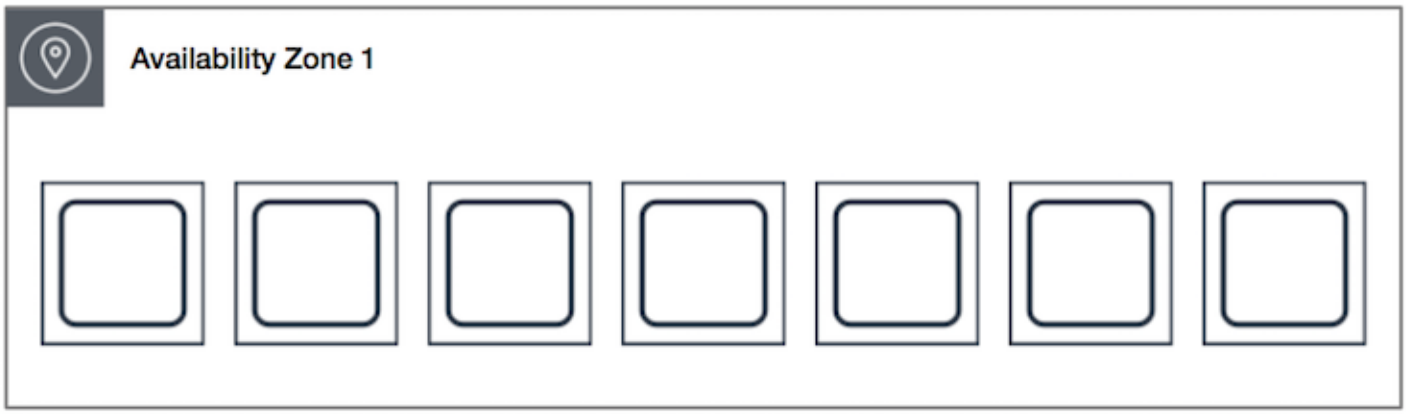
서로 떨어져 있어야 하는 중요 인스턴스의 수가 적은 애플리케이션에서는 분산형 배치 그룹이 권장됩니다. 분산형 레벨 배치 그룹에서 인스턴스를 시작하면 인스턴스가 동일한 장비를 공유할 때 장애가 동시에 발생할 수 있는 위험이 줄어듭니다. 분산형 레벨 배치 그룹은 별개의 하드웨어에 대한 액세스를 제공하기 때문에 시간 경과에 따라 인스턴스를 시작하거나 인스턴스 유형을 혼합할 때 적합합니다.

분산된 배치 그룹에서 인스턴스를 시작할 때 요청을 이행하기에 충분한 고유 하드웨어가 없으면 요청이 실패합니다. Amazon EC2는 시간이 지남에 따라 개별 하드웨어를 추가로 제공하므로 나중에 다시

요청을 시도할 수 있습니다. 배치 그룹은 랙 또는 호스트 간에 인스턴스를 분산시킬 수 있습니다. 랙 레벨 분산 배치 그룹은 AWS 리전과 AWS Outposts에서 사용할 수 있습니다. 호스트 레벨 분산 배치 그룹은 AWS Outposts에서만 사용할 수 있습니다.

랙 레벨 분산 배치 그룹

다음 이미지는 분산형 배치 그룹에 배치되는 단일 가용 영역에 있는 인스턴스 7개를 보여줍니다. 7개의 인스턴스가 7개의 서로 다른 랙에 배치되며, 랙마다 자체 네트워크 및 전원이 있습니다.



랙 레벨 분산 배치 그룹은 동일한 리전의 여러 가용 영역에 적용될 수 있습니다. 리전에서 랙 레벨 분산 배치 그룹은 그룹당 가용 영역별로 최대 7개의 실행 중인 인스턴스를 가질 수 있습니다. Oposts를 사용하면 랙 레벨 분산 배치 그룹은 Outpost 배포에 있는 랙 수만큼의 인스턴스를 보유할 수 있습니다.

호스트 레벨 분산형 배치 그룹

호스트 레벨 분산 배치 그룹은 AWS Outposts에서만 사용할 수 있습니다. 호스트 레벨 분산 배치 그룹은 Outpost 배포에 있는 호스트의 개수만큼의 인스턴스를 보유할 수 있습니다. 자세한 내용은 the section called “AWS Outposts에서의 배치 그룹” 섹션을 참조하세요.

규칙 및 제한 사항

분산형 배치 그룹에는 다음 규칙이 적용됩니다.

- 랙 분산형 배치 그룹은 각 그룹의 가용 영역당 실행 인스턴스를 최대 7개까지 지원합니다. 예를 들어 가용 영역이 3개인 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 8개 이상 있어야 하며, 분산 배치 그룹을 여러 개 사용하는 것이 좋습니다. 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 인스턴스에서 지원되지 않습니다.
- 호스트 레벨 분산형 배치 그룹은 AWS Outposts의 배치 그룹에서만 지원됩니다. 호스트 레벨 분산 배치 그룹은 Outpost 배포에 있는 호스트의 개수만큼의 인스턴스를 보유할 수 있습니다.
- 리전에서 랙 레벨 분산 배치 그룹은 그룹당 가용 영역별로 최대 7개의 실행 중인 인스턴스를 가질 수 있습니다. AWS Outposts를 사용하면 랙 레벨 분산 배치 그룹은 Outpost 배포에 있는 랙 수만큼의 인스턴스를 보유할 수 있습니다.
- 용량 예약은 스프레드 배치 그룹의 용량을 예약하지 않습니다.

EC2 인스턴스에 대한 배치 그룹 생성

배치 그룹을 사용하여 서로 관련된 인스턴스 배치를 제어할 수 있습니다. 배치 그룹을 생성한 후 배치 그룹에서 인스턴스를 시작할 수 있습니다.

제한 사항

리전당 최대 500개의 배치 그룹을 생성할 수 있습니다.

Console

배치 그룹을 생성하려면 다음을 수행하세요.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 배치 그룹을 선택합니다.
3. 배치 그룹 생성을 선택합니다.
4. 그룹의 이름을 지정합니다.
5. 그룹의 배치 전략(클러스터, 스프레드 또는 파티션)을 선택합니다.

   스프레드를 선택한 경우 스프레드 수준(랙 또는 호스트)을 선택해야 합니다.

   파티션을 선택한 경우 그룹의 파티션 수를 입력해야 합니다.
6. (선택 사항) 태그를 추가하려면 새 태그 추가를 선택하고 키와 값을 입력합니다.
7. 그룹 생성을 선택합니다.

AWS CLI

create-placement-group 명령을 사용합니다.

클러스터 배치 그룹을 생성하려면 다음을 수행하세요.

다음 예제에서는 `cluster` 배치 전략을 사용하는 배치 그룹을 생성한 후 키가 `purpose`이고 값이 `production`인 태그를 적용합니다.

```
aws ec2 create-placement-group \
    --group-name my-cluster \
    --strategy cluster \
    --tag-specifications 'ResourceType=placement-
group,Tags={Key=purpose,Value=production}'
```

파티션 배치 그룹을 생성하려면 다음을 수행하세요.

다음 예제에서는 `partition` 배치 전략을 사용하는 배치 그룹을 생성한 후 `--partition-count` 파라미터를 사용하여 5개의 파티션을 지정합니다.

```
aws ec2 create-placement-group \
    --group-name HDFS-Group-A \
    --strategy partition \
    --partition-count 5
```

PowerShell

New-EC2PlacementGroup cmdlet을 사용합니다.

클러스터 배치 그룹을 생성하려면 다음을 수행하세요.

다음 예제는 클러스터 배치 그룹을 생성합니다.

```
New-EC2PlacementGroup `
    -GroupName my-placement-group `
    -Strategy cluster
```

파티션 배치 그룹 생성

다음 예제는 파티션 배치 그룹을 생성합니다.

```
New-EC2PlacementGroup `
    -GroupName my-placement-group `
    -Strategy partition `
```

```
    -PartitionCount 5
```

EC2 인스턴스의 배치 변경

다음과 같이 인스턴스의 배치 그룹을 변경할 수 있습니다.

- 배치 그룹에 인스턴스 추가
- 한 배치 그룹에서 다른 배치 그룹으로 인스턴스 이동
- 배치 그룹에서 인스턴스 제거

요구 사항

인스턴스의 배치 그룹을 변경하려면 인스턴스가 `stopped` 상태여야 합니다.

Console

인스턴스 배치를 변경하려면 다음을 수행하세요.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다.
4. 작업, 인스턴스 설정, 인스턴스 배치 수정을 차례로 선택합니다.
5. 배치 그룹에 대해 다음 중 하나를 수행합니다.
   - 배치 그룹에 인스턴스를 추가하려면 배치 그룹을 선택합니다.
   - 배치 그룹 간에 인스턴스를 이동하려면 배치 그룹을 선택합니다.
   - 배치 그룹에서 인스턴스를 제거하려면 없음을 선택합니다.
6. 저장을 선택합니다.

AWS CLI

인스턴스를 배치 그룹으로 이동하는 방법

다음 modify-instance-placement 명령을 사용합니다.

```
aws ec2 modify-instance-placement \
    --instance-id i-0123a456700123456 \
```

```
    --group-name MySpreadGroup
```

배치 그룹에서 인스턴스를 제거하는 방법

다음 modify-instance-placement 명령을 사용합니다. 배치 그룹 이름에 빈 문자열을 지정하면 현재 배치 그룹에서 인스턴스가 제거됩니다.

```
aws ec2 modify-instance-placement \
    --instance-id i-0123a456700123456 \
    --group-name ""
```

PowerShell

인스턴스를 배치 그룹으로 이동하는 방법

배치 그룹의 이름과 함께 Edit-EC2InstancePlacement cmdlet을 사용합니다.

```
Edit-EC2InstancePlacement `
    -InstanceId i-0123a456700123456 `
    -GroupName MySpreadGroup
```

배치 그룹에서 인스턴스를 제거하는 방법

배치 그룹의 이름에 대한 빈 문자열과 함께 Edit-EC2InstancePlacement cmdlet을 사용합니다.

```
Edit-EC2InstancePlacement `
    -InstanceId i-0123a456700123456 `
    -GroupName ""
```

배치 그룹 삭제

대체해야 하거나 더 이상 필요하지 않은 배치 그룹을 삭제할 수 있습니다. 배치 그룹을 삭제하려면 배치 그룹에 인스턴스가 없어야 합니다. 인스턴스를 종료하거나, 다른 배치 그룹으로 이동하거나, 배치 그룹에서 제거할 수 있습니다.

Console

배치 그룹을 삭제하는 방법

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

2. 탐색 창에서 배치 그룹을 선택합니다.
3. 배치 그룹을 선택하고 작업, 삭제를 선택합니다.
4. 확인 메시지가 나타나면 **Delete**를 입력한 다음 삭제를 선택합니다.

AWS CLI

배치 그룹 삭제

delete-placement-group 명령을 사용합니다.

```
aws ec2 delete-placement-group --group-name my-cluster
```

PowerShell

배치 그룹 삭제

Remove-EC2PlacementGroup cmdlet을 사용합니다.

```
Remove-EC2PlacementGroup -GroupName my-cluster
```

공유된 배치 그룹

배치 그룹 공유를 통해 개별 AWS 계정이 소유한 상호 의존적인 인스턴스의 배치에 영향을 미칠 수 있습니다. 소유자는 여러 AWS 계정 간에 또는 조직 내에서 배치 그룹을 공유할 수 있습니다. 참가자는 자신의 계정에 공유되는 배치 그룹에서 인스턴스를 시작할 수 있습니다.

배치 그룹 소유자는 배치 그룹을 다음과 공유할 수 있습니다.

- 조직 내부 또는 외부의 특정 AWS 계정
- 조직 내부의 조직 단위
- 전체 조직

VPC 피어링을 사용하여 개별 AWS 계정이 소유한 인스턴스를 연결하고 공유 클러스터 배치 그룹이 제공하는 지연 시간의 이점을 누릴 수 있습니다.

내용

- 규칙 및 제한 사항
- 필수 권한
- 가용 영역 공유
- 배치 그룹 공유
- 배치 그룹 공유 취소

규칙 및 제한 사항

사용자가 배치 그룹을 공유하거나 다른 사람이 사용자와 배치 그룹을 공유하는 경우 다음 규칙과 제한 사항이 적용됩니다.

- 배치 그룹을 공유하려면 전용 호스트를 AWS 계정에 소유하고 있어야 합니다. 사용자와 공유된 배치 그룹은 공유할 수 없습니다.
- 파티션 또는 분산형 배치 그룹을 공유할 경우 배치 그룹 한도는 변경되지 않습니다. 공유 파티션 배치 그룹은 가용 영역당 실행 인스턴스를 최대 7개까지 지원하며, 공유 분산형 배치 그룹은 가용 영역당 실행 인스턴스를 최대 7개까지 지원합니다.
- 배치 그룹을 조직 또는 조직 내 조직 단위와 공유하려면 AWS Organizations와의 공유를 활성화해야 합니다. 자세한 내용은 AWS 리소스 공유를 참조하세요.
- AWS Management Console을 사용하여 인스턴스를 시작하는 경우 공유된 배치 그룹을 선택할 수 있습니다. AWS CLI를 사용하여 인스턴스를 시작하는 경우 이름이 아닌 ID별로 공유된 배치 그룹을 지정해야 합니다. 공유된 배치 그룹의 소유자인 경우에만 배치 그룹의 이름을 사용할 수 있습니다.
- 공유 배치 그룹에서 사용자가 소유한 인스턴스를 관리할 책임은 사용자에게 있습니다.
- 공유 배치 그룹과 연결되어 있지만 사용자가 소유하지 않은 인스턴스 및 용량 예약은 보거나 수정할 수 없습니다.
- 배치 그룹의 Amazon 리소스 이름(ARN)에는 배치 그룹을 소유한 계정의 ID가 포함되어 있습니다. 배치 그룹 ARN의 계정 ID 부분을 사용하여 사용자와 공유되는 배치 그룹의 소유자를 식별할 수 있습니다.

필수 권한

배치 그룹을 공유하려면 사용자에게는 다음 작업에 대한 권한이 있어야 합니다.

- `ec2:PutResourcePolicy`
- `ec2:DeleteResourcePolicy`

가용 영역 공유

리전의 가용 영역에 걸쳐 리소스가 배포될 수 있도록 각 계정의 이름에 가용 영역을 독립적으로 매핑합니다. 이로 인해 계정 전체에서 가용 영역 이름의 차이가 발생할 수 있습니다. 예를 들어 AWS 계정의 `us-east-1a` 가용 영역은 다른 AWS 계정에 대한 `us-east-1a`로 위치가 동일하지 않을 수 있습니다.

계정과 관련된 전용 호스트의 위치를 지정하려면 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 가용 영역 ID이고 모든 AWS 계정에서 동일한 위치입니다. 자세한 내용은 AZ IDs를 참조하세요.

배치 그룹 공유

배치 그룹을 공유하려면 리소스 공유에 추가해야 합니다. 리소스 공유는 여러 AWS 계정에서 리소스를 공유할 수 있게 해주는 AWS RAM 리소스입니다. 리소스 공유는 공유할 리소스와 공유 대상 소비자를 지정합니다.

AWS Organizations에서 조직에 속해 있고 조직 내에서 공유가 활성화된 경우, 공유 배치 그룹에 대한 액세스 권한이 조직의 소비자에게 부여됩니다.

배치 그룹이 조직 외부의 AWS 계정과 공유되는 경우, AWS 계정 소유자는 리소스 공유에 참여하라는 초대를 받게 됩니다. 해당 소유자는 초대를 수락한 후 공유 배치 그룹에 액세스할 수 있습니다.

AWS Resource Access Manager를 사용하여 AWS 계정에서 배치 그룹을 공유할 수 있습니다. 자세한 내용은 AWS RAM 사용 설명서의 리소스 공유 생성을 참조하세요.

배치 그룹 공유 취소

배치 그룹 소유자는 언제든지 공유 배치 그룹을 공유 해제할 수 있습니다. 공유된 배치 그룹을 공유 해제할 경우 다음 변경 사항이 적용됩니다.

- 배치 그룹이 공유된 AWS 계정이 더 이상 인스턴스를 시작하거나 용량을 예약할 수 없습니다.
- 공유 배치 그룹에서 실행 중인 모든 인스턴스는 배치 그룹과의 연결이 끊어지지만 AWS 계정에서는 계속 실행됩니다.
- 공유된 배치 그룹의 모든 용량 예약은 배치 그룹과의 연결이 끊어지지만 AWS 계정에서는 계속 사용할 수 있습니다.

자세한 내용은 AWS RAM 사용 설명서의 리소스 공유 삭제를 참조하세요.

AWS Outposts에서의 배치 그룹

AWS Outposts은(는) AWS 인프라, 서비스, API 및 도구를 고객 온프레미스로 확장하는 완전관리형 서비스입니다. AWS 관리형 인프라에 대한 로컬 액세스를 제공하는 AWS Outposts을(를) 통해 고객은 AWS 리전에서 사용하는 것과 동일한 프로그래밍 인터페이스를 사용해 온프레미스에서 애플리케이션을 구축하고 실행할 수 있으며, 짧은 지연 시간과 로컬 데이터 처리가 필요한 경우에 로컬 컴퓨팅 및 스토리지 리소스를 사용할 수 있습니다.

Outpost는 고객 사이트에 배포된 AWS의 컴퓨팅 및 스토리지 용량 풀입니다. AWS는 이 용량을 AWS 리전의 일부로 운영, 모니터링 및 관리합니다.

계정에 생성한 Outposts에서 배치 그룹을 생성할 수 있습니다. 이렇게 하면 사이트의 Outposts에서 기본 하드웨어에 인스턴스를 분산시킬 수 있습니다. 일반 가용 영역에서 배치 그룹을 생성하고 사용하는 것과 동일한 방식으로 Outposts에서 배치 그룹을 생성하고 사용합니다. Outpost에 분산 전략이 있는 배치 그룹을 생성할 때, 배치 그룹이 호스트나 랙에 인스턴스를 분산하도록 선택할 수 있습니다. 호스트 간에 인스턴스를 분산하면 단일 랙 Outpost로 분산 전략을 사용할 수 있습니다.

고려 사항

- 랙 레벨 분산 배치 그룹은 Outpost 배포에 있는 랙 수만큼의 인스턴스를 보유할 수 있습니다.
- 호스트 레벨 분산 배치 그룹은 Outpost 배포에 있는 호스트의 개수만큼의 인스턴스를 보유할 수 있습니다.

전제 조건

사이트에 Outpost가 설치되어 있어야 합니다. 자세한 내용은 AWS Outposts 사용 설명서에서 Outposts 생성 및 Outposts 용량 주문을 참조하세요.

Outpost에서 배치 그룹 사용

1. Outposts에서 서브넷을 생성합니다. 자세한 내용은 AWS Outposts 사용 설명서에서 서브넷 생성을 참조하세요.
2. Outpost의 연결된 리전에 배치 그룹을 생성합니다. 분산 전략을 사용하여 배치 그룹을 만드는 경우, 호스트 또는 랙 레벨 분산을 선택하여 그룹이 Outpost의 기본 하드웨어에 인스턴스를 분산하는 방법을 결정할 수 있습니다. 자세한 내용은 the section called “배치 그룹 생성” 섹션을 참조하세요.
3. 배치 그룹으로 인스턴스 시작 서브넷(Subnet)에서 1단계에 생성한 서브넷을 선택하고, 배치 그룹 이름(Placement group name)에서 2단계에 생성한 배치 그룹을 선택합니다. 자세한 내용은 AWS Outposts 사용 설명서에서 Outposts에서 인스턴스 시작을 참조하세요.

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 프레임은 패킷 또는 전송 중인 실제 데이터와 이를 둘러싼 네트워크 오버헤드 정보로 구성됩니다.

이더넷 프레임은 여러 가지 형식으로 제공될 수 있으며, 가장 일반적인 형식은 표준 이더넷 v2 프레임 형식입니다. 대부분의 인터넷에서 지원되는 최대 이더넷 패킷 크기인 1500MTU를 지원합니다. 인스턴스의 지원되는 최대 MTU는 인스턴스 유형에 따라 다릅니다.

모든 EC2 인스턴스 유형에서 1,500MTU가 지원됩니다.

내용

- 점보 프레임(9001 MTU)
- 경로 MTU 검색
- Amazon EC2 인스턴스의 MTU 설정
- 문제 해결

점보 프레임(9001 MTU)

점보 프레임에서는 패킷당 페이로드 크기를 상향하여 패킷 오버헤드에 해당하지 않는 패킷의 비율을 늘릴 수 있습니다. 검보 프레임은 같은 양의 사용 가능한 데이터를 보내더라도 더 적은 수의 패킷만 있으면 됩니다. 그러나 특정 유형의 트래픽에는 다음과 같은 최대 페이로드가 적용됩니다.

MTU 한도 1500바이트

- 인터넷 게이트웨이를 통한 트래픽
- VPN 연결을 통한 트래픽
- 전송 게이트웨이를 사용하지 않는 경우의 AWS 리전 간 트래픽

MTU 한도 8500바이트

- 리전 간 VPC 피어링 연결을 통한 트래픽

패킷이 해당 MTU 한도를 초과하는 경우, 조각화되거나 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 Amazon EC2 인스턴스에 대한 배치 그룹 섹션을 참조하세요.

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 프라이빗 가상 인터페이스 또는 전송 가상 인터페이스에 대한 MTU를 참조하세요.

모든 현재 세대 인스턴스 유형에서 점보 프레임이 지원됩니다. A1, C3, I2, M3, R3 등 이전 세대 인스턴스 유형에서는 점보 프레임이 지원됩니다.

관련 리소스

- NAT 게이트웨이의 경우 Amazon VPC 사용 설명서에서 NAT 게이트웨이 기본 사항을 참조하세요.
- 전송 게이트웨이의 경우 Amazon VPC 전송 게이트웨이 사용 설명서의 최대 전송 단위를 참조하세요.
- 로컬 영역의 경우 AWS 로컬 영역 사용 설명서에서 Considerations를 참조하세요.
- AWS Wavelength의 경우 AWS Wavelength 사용 설명서의 최대 전송 단위를 참조하세요.
- Outposts의 경우 AWS Outposts 사용 설명서의 서비스 링크 최대 전송 단위 요구 사항을 참조하세요.

경로 MTU 검색

경로 MTU 탐색(PMTUD)을 사용하여 두 디바이스 간의 경로 MTU를 결정할 수 있습니다. 경로 MTU는 발신 호스트와 수신 호스트 간의 경로에서 지원되는 최대 패킷 사이즈입니다. 두 호스트 간의 네트워크 MTU 크기에 차이가 있는 경우 PMTUD를 사용하면 수신 호스트가 ICMP 메시지로 발신 호스트에 응답할 수 있습니다. 이 ICMP 메시지는 발신 호스트가 네트워크 경로를 따라 최저 MTU 크기를 사용하고 요청을 재전송하도록 지시합니다. 이러한 협상이 없으면 수신 호스트가 허용할 수 없을 만큼 요청이 너무 많아져서 패킷 손실이 발생할 수 있습니다.

IPv4의 경우 호스트가 수신 호스트의 MTU 또는 경로를 따르는 디바이스의 MTU보다 큰 패킷을 전송하는 경우 수신 호스트 또는 디바이스가 패킷을 삭제한 다음 `Destination Unreachable:`

`Fragmentation Needed and Don't Fragment was Set`(유형 3, 코드 4)과 같은 ICMP 메시지를 반환합니다. 이렇게 하면 전송 호스트에 페이로드를 여러 개의 작은 패킷으로 분할한 다음 다시 전송하도록 지시합니다.

IPv6 프로토콜은 네트워크의 조각화를 지원하지 않습니다. 호스트가 수신 호스트의 MTU 또는 경로를 따르는 디바이스의 MTU보다 큰 패킷을 전송하는 경우 수신 호스트 또는 디바이스가 패킷을 삭제한 다음 `ICMPv6 Packet Too Big (PTB)`(유형 2)과 같은 ICMP 메시지를 반환합니다. 이렇게 하면 전송 호스트에 페이로드를 여러 개의 작은 패킷으로 분할한 다음 다시 전송하도록 지시합니다.

NAT 게이트웨이 및 로드 밸런서와 같은 일부 구성 요소를 통한 연결은 자동으로 추적됩니다. 이것은 아웃바운드 연결 시도에 대해 보안 그룹 추적이 자동으로 활성화된다는 의미입니다. 연결이 자동으로 추적되거나 보안 그룹 규칙이 인바운드 ICMP 트래픽을 허용하는 경우 PMTUD 응답을 받을 수 있습니다.

서브넷에 대한 ICMP 트래픽을 거부하는 네트워크 액세스 제어 목록 항목이 있는 경우와 같이 보안 그룹 수준에서 트래픽이 허용되는 경우에도 ICMP 트래픽을 차단할 수 있습니다.

> **Important**
>
> 경로 MTU 검색에서는 일부 라우터에서 점보 프레임이 삭제되지 않도록 보장하지 않습니다. VPC의 인터넷 게이트웨이는 패킷을 최대 1,500바이트까지만 전송합니다. 인터넷 트래픽에는 1,500MTU 패킷이 권장됩니다.

NAT 게이트웨이에 대한 MTU 규칙은 Amazon VPC 사용 설명서의 최대 전송 단위(MTU)를 참조하세요. 전송 게이트웨이에 대한 MTU 규칙은 AWS Transit Gateway 사용 설명서의 최대 전송 단위(MTU)를 참조하세요.

Amazon EC2 인스턴스의 MTU 설정

네트워크 연결의 MTU(최대 전송 단위)는 연결을 통해 전달할 수 있는 허용되는 최대 크기의 패킷 크기(바이트)입니다. 모든 Amazon EC2 인스턴스는 표준 프레임(1500 MTU)을 지원하고 모든 현재 세대 인스턴스 유형은 점보 프레임(9001 MTU)을 지원합니다.

Amazon EC2 인스턴스의 MTU를 보고, 인스턴스와 다른 호스트 간의 경로 MTU를 보고, 표준 또는 점보 프레임을 사용하도록 인스턴스를 구성할 수 있습니다.

업무

- 두 호스트 간 경로 MTU 확인

- 인스턴스의 MTU 확인
- 인스턴스의 MTU 설정

두 호스트 간 경로 MTU 확인

EC2 인스턴스와 다른 호스트 사이의 경로 MTU를 확인할 수 있습니다. DNS 이름 또는 IP 주소를 대상으로 지정할 수 있습니다. 대상이 다른 EC2 인스턴스인 경우에는 해당 보안 그룹이 인바운드 UDP 트래픽을 허용하는지 확인합니다.

사용하는 절차는 인스턴스의 운영 체제에 따라 달라집니다.

Linux 인스턴스

인스턴스에서 tracepath 명령을 실행하여 EC2 인스턴스와 지정된 대상 간의 경로 MTU를 확인합니다. 이 명령은 많은 Linux 배포판에서 기본적으로 제공되는 `iputils` 패키지의 일부입니다.

이 예에서는 EC2 인스턴스와 `amazon.com` 간의 경로 MTU를 확인합니다.

```
[ec2-user ~]$ tracepath amazon.com
```

이 예제 출력에서 경로 MTU는 1500입니다.

```
 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
```

Windows 인스턴스

mturoute를 사용하여 경로 MTU를 확인하려면 다음을 수행합니다.

1. https://elifulkerson.com/projects/mturoute.php에서 EC2 인스턴스로 mturoute.exe를 다운로드하세요.
2. 명령 프롬프트 창을 열고 mturoute.exe를 다운로드한 디렉터리로 변경합니다.
3. 다음 명령을 사용하여 EC2 인스턴스와 지정된 대상 간의 경로 MTU를 확인합니다. 이 예에서는 EC2 인스턴스와 `www.elifulkerson.com` 간의 경로 MTU를 확인합니다.

```
.\mturoute.exe www.elifulkerson.com
```

이 예제 출력에서 경로 MTU는 1500입니다.

```
* ICMP Fragmentation is not permitted. *
* Speed optimization is enabled. *
* Maximum payload is 10000 bytes. *
+ ICMP payload of 1472 bytes succeeded.
- ICMP payload of 1473 bytes is too big.
Path MTU: 1500 bytes.
```

인스턴스의 MTU 확인

인스턴스의 MTU 값을 확인할 수 있습니다. 일부 인스턴스는 점보 프레임을 사용하도록 구성되어 있는 반면, 표준 프레임 크기를 사용하도록 구성된 인스턴스도 있습니다.

사용하는 절차는 인스턴스의 운영 체제에 따라 달라집니다.

Linux 인스턴스

Linux 인스턴스에서 MTU 설정을 확인하려면

EC2 인스턴스에서 다음 ip 명령을 실행합니다. 기본 네트워크 인터페이스가 `eth0`이 아닌 경우 `eth0`을 네트워크 인터페이스로 교체합니다.

```
[ec2-user ~]$ ip link show eth0
```

이 예제 출력에서 *`mtu 9001`*은 인스턴스가 점보 프레임을 사용함을 나타냅니다.

```
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
```

Windows 인스턴스

사용하는 절차는 인스턴스의 드라이버에 따라 다릅니다.

ENA driver

버전 2.1.0 이상

MTU 값을 가져오려면 EC2 인스턴스에서 다음 Get-NetAdapterAdvancedProperty 명령을 사용하세요. 와일드카드(별표)를 사용하여 모든 이더넷 이름을 가져옵니다. 출력에서 인터페이스 이름 `*JumboPacket`을 확인합니다. 9015 값은 점보 프레임이 활성화되어 있음을 나타냅니다. 점보 프레임은 기본적으로 비활성화되어 있습니다.

```
Get-NetAdapterAdvancedProperty -Name "Ethernet*"
```

버전 1.5 이하

MTU 값을 가져오려면 EC2 인스턴스에서 다음 Get-NetAdapterAdvancedProperty 명령을 사용하세요. 출력에서 인터페이스 이름 `MTU`을 확인합니다. 값 9001은 점보 프레임이 활성화되었음을 나타냅니다. 점보 프레임은 기본적으로 비활성화되어 있습니다.

```
Get-NetAdapterAdvancedProperty -Name "Ethernet"
```

Intel SRIOV 82599 driver

MTU 값을 가져오려면 EC2 인스턴스에서 다음 Get-NetAdapterAdvancedProperty 명령을 사용하세요. 인터페이스 이름 `*JumboPacket`에 대한 항목을 확인합니다. 값 9014는 점보 프레임이 활성화되었음을 나타냅니다. (MTU 크기에는 헤더와 페이로드가 포함되어 있습니다.) 점보 프레임은 기본적으로 비활성화되어 있습니다.

```
Get-NetAdapterAdvancedProperty -Name "Ethernet"
```

AWS PV driver

MTU 값을 가져오려면 EC2 인스턴스에서 다음 명령을 사용하세요. 인터페이스 이름은 다를 수 있습니다. 출력에서 "Ethernet," "Ethernet 2" 또는 "Local Area Connection"이라는 이름의 항목을 찾습

니다. 점보 프레임을 활성화 또는 비활성화하려면 인터페이스 이름이 필요합니다. 값 9001은 점보 프레임이 활성화되었음을 나타냅니다.

```
netsh interface ipv4 show subinterface
```

인스턴스의 MTU 설정

VPC 내의 네트워크 트래픽에는 점보 프레임을 사용하고 인터넷 트래픽에는 표준 프레임을 사용할 수 있습니다. 어떤 사용 사례이든 인스턴스가 예상한 대로 작동하는지 확인하는 것이 좋습니다.

사용하는 절차는 인스턴스의 운영 체제에 따라 달라집니다.

Linux 인스턴스

Linux 인스턴스에서 MTU 값을 설정하려면

1. 인스턴스에서 다음 ip 명령을 실행합니다. 원하는 MTU 값을 1500으로 설정하지만 대신 9001을 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 수정하세요.

   - 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 Linux AMI의 경우, 다음 줄을 `/etc/dhcp/dhclient-eth0.conf` 파일에 추가합니다.

     ```
     interface "eth0" {
     ```

```
supersede interface-mtu 1500;
}
```

- 다른 Linux 배포의 경우, 해당 설명서를 참조하세요.

3. (선택 사항) 인스턴스를 재부팅하고 MTU 설정이 올바른지 확인합니다.

Windows 인스턴스

사용하는 절차는 인스턴스의 드라이버에 따라 다릅니다.

ENA driver

디바이스 관리자 또는 인스턴스의 Set-NetAdapterAdvancedProperty 명령을 사용하여 MTU를 변경할 수 있습니다.

버전 2.1.0 이상

점보 프레임을 사용하려면 다음 명령을 사용합니다.

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "*JumboPacket" -
RegistryValue 9015
```

점보 프레임을 사용하지 않으려면 다음 명령을 사용합니다.

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "*JumboPacket" -
RegistryValue 1514
```

버전 1.5 이하

점보 프레임을 사용하려면 다음 명령을 사용합니다.

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "MTU" -
RegistryValue 9001
```

점보 프레임을 사용하지 않으려면 다음 명령을 사용합니다.

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "MTU" -
RegistryValue 1500
```

Intel SRIOV 82599 driver

디바이스 관리자 또는 인스턴스의 Set-NetAdapterAdvancedProperty 명령을 사용하여 MTU를 변경할 수 있습니다.

점보 프레임을 사용하려면 다음 명령을 사용합니다.

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "*JumboPacket" -
RegistryValue 9014
```

점보 프레임을 사용하지 않으려면 다음 명령을 사용합니다.

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "*JumboPacket" -
RegistryValue 1514
```

AWS PV driver

인스턴스에서 netsh 명령을 사용하여 MTU를 변경할 수 있습니다. 디바이스 관리자를 사용하여 MTU 설정을 변경할 수 없습니다.

점보 프레임을 사용하려면 다음 명령을 사용합니다.

```
netsh interface ipv4 set subinterface "Ethernet" mtu=9001
```

점보 프레임을 사용하지 않으려면 다음 명령을 사용합니다.

```
netsh interface ipv4 set subinterface "Ethernet" mtu=1500
```

문제 해결

점보 프레임을 사용할 때 EC2 인스턴스와 Amazon Redshift 클러스터 사이에 연결 문제가 발생하는 경우 Amazon Redshift 클러스터 관리 가이드의 쿼리가 중단되거나, 간혹 클러스터까지 전송되지 않습니다를 참조하세요.

EC2 인스턴스용 가상 프라이빗 클라우드

Amazon Virtual Private Cloud(Amazon VPC)를 사용하면 AWS 클라우드 안에서 논리적으로 격리된 자체 영역에 virtual private cloud 또는 VPC라고 하는 가상 네트워크를 정의할 수 있습니다. Amazon EC2

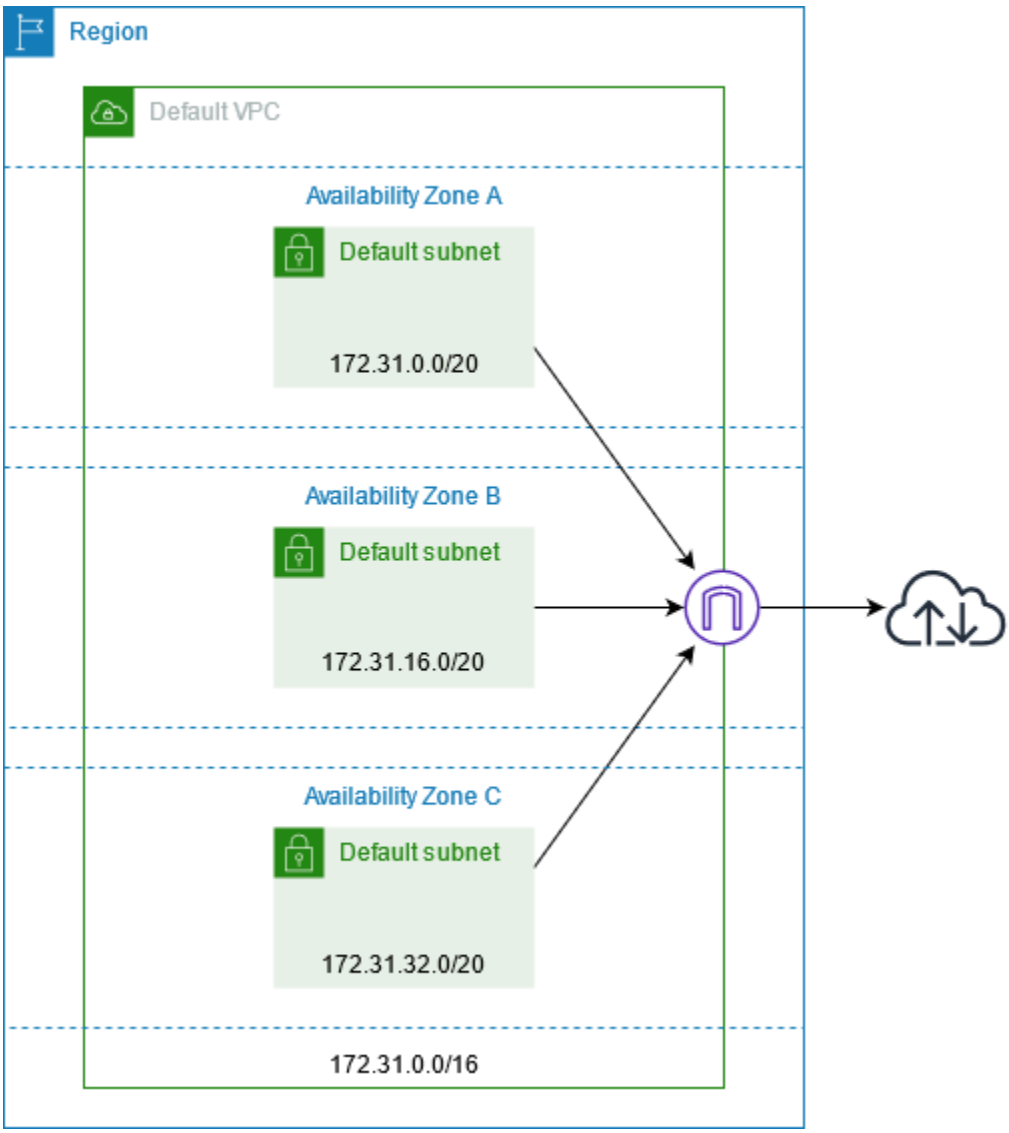
인스턴스와 같은 AWS 리소스를 VPC의 서브넷으로 실행할 수 있습니다. VPC는 고객의 자체 데이터 센터에서 운영하는 기존 네트워크와 매우 유사하지만 AWS의 확장 가능한 인프라를 사용한다는 이점을 제공합니다. 해당 IP 주소 범위를 선택하고, 서브넷을 만든 후 라우팅 테이블, 네트워크 게이트웨이 및 보안 설정을 구성하여 VPC를 구성할 수 있습니다. VPC의 인스턴스를 인터넷 또는 자체 데이터 센터에 연결합니다.

내용

- 기본 VPC
- 기본이 아닌 VPC
- 인터넷 액세스
- 공유 서브넷
- IPv6 전용 서브넷

기본 VPC

AWS 계정이 생성되면 각 리전에서 기본 VPC가 생성됩니다. 기본 VPC는 이미 구성되어 즉시 사용할 수 있는 VPC입니다. 예를 들어 각 기본 VPC의 각 가용 영역에 대한 기본 서브넷이 VPC에 연결된 인터넷 게이트웨이가 있으며, 모든 트래픽(0.0.0.0/0)을 인터넷 게이트웨이로 보내는 경로가 기본 라우팅 테이블에 있습니다. 기본 VPC의 구성을 필요에 따라 수정할 수 있습니다. 예를 들어 서브넷과 라우팅 테이블을 추가할 수 있습니다.



기본이 아닌 VPC

리소스에 기본 VPC를 사용하는 대신 Amazon VPC 사용 설명서의 VPC 생성에 설명된 대로 자체 VPC를 생성할 수 있습니다.

다음은 EC2용 VPC 인스턴스를 생성할 때 고려해야 할 몇 가지 사항입니다.

- IPv4 CIDR 블록에 대한 기본 제안을 사용하거나 애플리케이션이나 네트워크에 필요한 CIDR 블록을 입력할 수 있습니다.
- 높은 가용성을 보장하려면 여러 가용성 영역에 서브넷을 생성합니다.
- 인터넷에서 인스턴스에 액세스할 수 있어야 하는 경우 다음 중 하나를 수행합니다.

- 인스턴스가 퍼블릭 서브넷에 있을 수 있는 경우 퍼블릭 서브넷을 추가합니다. 두 DNS 옵션 모두 활성화된 상태로 유지합니다. 지금 또는 나중에 선택적으로 프라이빗 서브넷을 추가할 수 있습니다.
- 인스턴스가 프라이빗 서브넷에 있어야 하는 경우 프라이빗 서브넷만 추가합니다. NAT 게이트웨이를 추가하여 프라이빗 서브넷의 인스턴스에 대한 인터넷 액세스를 제공할 수 있습니다. 인스턴스가 가용성 영역을 통해 상당한 양의 트래픽을 송수신하는 경우 각 가용성 영역에 NAT 게이트웨이를 생성합니다. 그렇게 하지 않으면 가용성 영역 중 하나에만 NAT 게이트웨이를 생성하고 NAT 게이트웨이와 동일한 가용 영역에서 교차 영역 트래픽을 보내거나 받는 인스턴스를 시작할 수 있습니다.

인터넷 액세스

기본 VPC의 기본 서브넷으로 시작된 인스턴스는 기본 VPC가 퍼블릭 IP 주소와 DNS 호스트 이름을 할당하도록 구성되어 있고 기본 라우팅 테이블은 VPC 연결된 인터넷 게이트웨이에 대한 경로로 구성되므로 인터넷에 액세스할 수 있습니다.

기본이 아닌 서브넷과 VPC에서 시작하는 인스턴스의 경우 다음 중 하나를 수행하여 이러한 서브넷에서 시작하는 인스턴스가 인터넷에 액세스할 수 있는지 확인할 수 있습니다.

- 인터넷 게이트웨이를 구성합니다. 자세한 내용은 Amazon VPC 사용 설명서의 인터넷 게이트웨이를 사용한 인터넷 연결을 참조하세요.
- 퍼블릭 NAT 게이트웨이를 구성합니다. 자세한 내용은 Amazon VPC 사용 설명서에서 프라이빗 서브넷에서 인터넷 액세스를 참조하세요.

공유 서브넷

EC2 인스턴스를 공유 VPC 서브넷으로 시작할 경우 다음 사항에 유의하세요.

- 참가자는 공유 서브넷 ID를 지정하여 공유 서브넷에서 인스턴스를 실행할 수 있습니다. 참가자는 자신이 지정하는 모든 네트워크 인터페이스를 소유해야 합니다.
- 참가자는 공유 서브넷에서 만든 인스턴스를 시작, 중지, 종료하고 설명할 수 있습니다. 참가자는 공유 서브넷에서 VPC 소유자가 만든 인스턴스를 시작, 중지, 종료하거나 설명할 수 없습니다.
- VPC 소유자는 공유 서브넷에서 참가자들이 만든 인스턴스를 시작, 중지, 종료하고 설명할 수 없습니다.

- 참가자는 EC2 Instance Connect 엔드포인트를 사용하여 공유 서브넷의 인스턴스에 연결할 수 있습니다. 참가자는 공유 서브넷에 EC2 Instance Connect 엔드포인트를 생성해야 합니다. 참가자는 VPC 소유자가 공유 서브넷에서 생성한 EC2 인스턴스 연결 엔드포인트를 사용할 수 없습니다.

공유 Amazon EC2 리소스에 대한 자세한 내용은 다음을 참조하세요.

- the section called “조직 또는 OU와 AMI 공유 관리”
- the section called “공유 용량 예약”
- the section called “공유된 배치 그룹”
- 크로스 계정 Amazon EC2 전용 호스트 공유

공유 서브넷에 관한 자세한 내용은 Amazon VPC 사용 설명서의 다른 계정과 VPC 공유를 참조하세요.

IPv6 전용 서브넷

IPv6 전용 서브넷에서 시작된 EC2 인스턴스는 IPv6 주소를 수신하지만 IPv4 주소는 수신하지 않습니다. IPv6 전용 서브넷으로 시작하는 모든 인스턴스는 Nitro 기반 인스턴스여야 합니다.

Amazon EC2의 보안

AWS에서는 클라우드 보안을 가장 중요하게 생각합니다. AWS 고객은 보안에 가장 보안에 민감한 조직의 요구 사항에 부합하도록 빌드된 데이터 센터 및 네트워크 아키텍처의 혜택을 누릴 수 있습니다.

보안은 AWS와 사용자의 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드의 보안: AWS는 AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호할 책임이 있습니다. AWS는 안전하게 사용할 수 있는 서비스 또한 제공합니다. 타사 감사자는 AWS 규정 준수 프로그램의 일환으로 보안 효과를 정기적으로 테스트하고 검증합니다. Amazon EC2에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 규정 준수 프로그램의 범위에 속하는 AWS 서비스를 참조하세요.
- 클라우드 내부의 보안 - 고객의 책임에는 다음 영역이 포함됩니다.
  - VPC 및 보안 그룹을 구성하는 등의 작업을 통해 인스턴스에 대한 네트워크 액세스를 제어합니다. 자세한 내용은 네트워크 트래픽 제어 섹션을 참조하세요.
  - 인스턴스 연결에 사용되는 자격 증명을 관리합니다.
  - 업데이트 및 보안 패치를 포함하여 게스트 운영 체제에 배포된 게스트 운영 체제 및 소프트웨어를 관리합니다. 자세한 내용은 Amazon EC2 인스턴스에 대한 업데이트 관리 섹션을 참조하세요.
  - 인스턴스에 연결된 IAM 역할 및 해당 역할과 연결된 권한을 구성합니다. 자세한 내용은 Amazon EC2의 IAM 역할 섹션을 참조하세요.

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.

내용

- Amazon EC2의 데이터 보호
- Amazon EC2의 인프라 보안
- Amazon EC2의 복원성
- Amazon EC2의 규정 준수 확인
- Amazon EC2의 자격 증명 및 액세스 관리
- Amazon EC2 인스턴스에 대한 업데이트 관리
- Windows 인스턴스를 위한 보안 모범 사례

- Amazon EC2 키 페어 및 Amazon EC2 인스턴스
- EC2 인스턴스에 대한 Amazon EC2 보안 그룹
- Amazon EC2 인스턴스의 NitroTPM
- Windows 인스턴스용 Credential Guard
- 인터페이스 VPC 엔드포인트를 사용하여 Amazon EC2에 액세스

Amazon EC2의 데이터 보호

AWS 공동 책임 모델은 Amazon Elastic Compute Cloud의 데이터 보호에 적용됩니다. 이 모델에서 설명하는 것처럼 AWS는 모든 AWS 클라우드를 실행하는 글로벌 인프라를 보호할 책임이 있습니다. 사용자는 인프라에서 호스팅되는 콘텐츠를 관리해야 합니다. 사용하는 AWS 서비스의 보안 구성과 관리 태스크에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 대한 자세한 내용은 데이터 프라이버시 FAQ를 참조하세요. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 AWS 공동 책임 모델 및 GDPR 블로그 게시물을 참조하세요.

데이터를 보호하려면 AWS 계정 자격 증명을 보호하고 AWS IAM Identity Center 또는 AWS Identity and Access Management(IAM)를 통해 개별 사용자 계정을 설정하는 것이 좋습니다. 이렇게 하면 개별 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 다중 인증(MFA)을 사용하세요.
- SSL/TLS를 사용하여 AWS 리소스와 통신하세요. TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- AWS CloudTrail로 API 및 사용자 활동 로깅을 설정하세요. AWS 활동 캡처에 CloudTrail 추적을 사용하는 방법에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 CloudTrail 추적 작업을 참조하세요.
- AWS 암호화 솔루션을 AWS 서비스 내의 모든 기본 보안 컨트롤과 함께 사용하세요.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용하세요.
- 명령줄 인터페이스 또는 API를 통해 AWS에 액세스할 때 FIPS 140-3 검증된 암호화 모듈이 필요한 경우, FIPS 엔드포인트를 사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 Federal Information Processing Standard(FIPS) 140-3을 참조하세요.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 형식 텍스트 필드에 입력하지 않는 것이 좋습니다. 여기에는 Amazon EC2 또는 기타 AWS 서비스에서 콘솔, API,

AWS CLI 또는 AWS SDK를 사용하여 작업하는 경우가 포함됩니다. 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 자격 증명을 URL에 포함해서는 안 됩니다.

내용

- Amazon EBS 데이터 보안
- 저장 시 암호화
- 전송 중 암호화

Amazon EBS 데이터 보안

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

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

저장 시 암호화

EBS 볼륨

Amazon EBS 암호화는 EBS 볼륨과 스냅샷에 대한 암호화 솔루션입니다. 이는 AWS KMS keys을(를) 사용합니다. 자세한 내용은 Amazon EBS 사용 설명서의 Amazon EBS encryption을 참조하세요.

[Windows 인스턴스] 또한 폴더 및 파일 수준 암호화에 Microsoft EFS 및 NTFS 권한을 사용할 수도 있습니다.

인스턴스 스토어 볼륨

인스턴스의 하드웨어 모듈에 구현된 XTS-AES-256 암호를 사용하여 NVMe 인스턴스 스토어 볼륨의 데이터를 암호화합니다. 로컬로 연결된 NVMe 스토리지 디바이스에 기록된 데이터를 암호화하는 데 사용되는 키는 고객별 및 볼륨별입니다. 키는 AWS 직원이 액세스할 수 없는 하드웨어 모듈에 의해 생성되고 그 안에만 상주합니다. 인스턴스가 중지되거나 종료되면 암호화 키가 손상되어 복구가 불가능해집니다. 이 암호화를 비활성화할 수 없으며, 사용자 자신의 암호화 키를 제공할 수 없습니다.

H1, D3 및 D3en 인스턴스의 HDD 인스턴스 스토어 볼륨에 저장되는 데이터는 XTS-AES-256 및 일회용 키를 사용하여 암호화됩니다.

인스턴스를 중지하거나 최대 절전 모드로 전환하거나 종료하면 인스턴스 스토어 볼륨의 모든 스토리지 블록이 재설정됩니다. 따라서 다른 인스턴스의 인스턴스 스토어를 통해 데이터를 액세스할 수 없습니다.

메모리

메모리 암호화는 다음 인스턴스에서 활성화됩니다.

- AWS Graviton2 이상 AWS Graviton 프로세서가 있는 인스턴스는 상시 작동 메모리 암호화를 지원합니다. 암호화 키는 호스트 시스템 내에서 안전하게 생성되고 호스트 시스템을 벗어나지 않으며 호스트를 재부팅하거나 전원을 끌 때 삭제됩니다. 자세한 내용은 AWS Graviton 프로세서를 참조하세요.
- 3세대 인텔 제온 스케일러블 프로세서(Ice Lake)가 있는 인스턴스(예: M6i 인스턴스)와 4세대 인텔 제온 스케일러블 프로세서(Sapphire Rapids)가 있는 인스턴스(예: M7i 인스턴스)입니다. 이 프로세서는 인텔 전체 메모리 암호화(TME)를 사용한 상시 메모리 암호화를 지원합니다.
- 3세대 AMD EPYC 프로세서(Milan)가 있는 인스턴스(예: M6a 인스턴스)와 4세대 AMD EPYC 프로세서(Genoa)가 있는 인스턴스(예: M7a 인스턴스)입니다. 이러한 프로세서에서는 AMD SME(Secure Memory Encryption)를 사용하는 상시 메모리 암호화를 지원합니다. 3세대 AMD EPYC 프로세서(Milan)가 있는 인스턴스에서는 AMD Secure Encrypted Virtualization-Secure Nested Paging(SEV-SNP)도 지원합니다.

전송 중 암호화

물리적 계층에서의 암호화

AWS 글로벌 네트워크를 통해 AWS 리전으로 흐르는 모든 데이터는 AWS 보안 시설을 떠나기 전에 물리적 계층에서 자동으로 암호화됩니다. AZ 간에 전송되는 모든 트래픽은 암호화됩니다. 이 섹션에 나열된 암호화 계층을 비롯한 추가 암호화 계층은 추가적인 보호 기능을 제공할 수 있습니다.

Amazon VPC 피어링 및 전송 게이트웨이 리전 간 피어링에서 제공하는 암호화

Amazon VPC 피어링 및 전송 게이트웨이 피어링을 사용하는 모든 리전 간 트래픽은 리전을 종료하면 자동으로 대량 암호화됩니다. 앞서 이 섹션에서 설명한 것처럼 AWS 보안 시설에서 나가기 전에 모든 트래픽에 대한 추가 암호화 계층이 물리적 계층에서 자동으로 제공됩니다.

인스턴스 간 암호화

AWS는 모든 유형의 EC2 인스턴스 간 보안 프라이빗 연결을 제공합니다. 또한 일부 인스턴스 유형은 기본 Nitro 시스템 하드웨어의 오프로드 기능을 사용하여 인스턴스 간 전송 중 트래픽을 자동으로 암호화합니다. 이 암호화는 256비트 암호화와 함께 관련 데이터로 인증된 암호화(AEAD) 알고리즘을 사용합니다. 네트워크 성능에는 영향을 미치지 않습니다. 인스턴스 간에 이러한 전송 중 트래픽 암호화를 추가로 지원하려면 다음 요구 사항을 충족해야 합니다.

- 이러한 인스턴스는 다음 인스턴스 유형을 사용합니다.
  - 범용: M5dn, M5n, M5zn, M6a, M6i, M6id, M6idn, M6in, M7a, M7g, M7gd, M7i, M7i-flex, M8g, M8gd
  - 컴퓨팅 최적화: C5n, C6a, C6gn, C6i, C6id, C6in, C7a, C7g, C7gd, C7gn, C7i, C7i-flex, C8g, C8gd
  - 메모리 최적화: R5dn, R5n, R6a, R6i, R6idn, R6in, R6id, R7a, R7g, R7gd, R7i, R7iz, R8g, R8gd, U-3tb1, U-6tb1, U-9tb1, U-12tb1, U-18tb1, U-24tb1, U7i-6tb, U7i-8tb, U7i-12tb, U7in-16tb, U7in-24tb, U7in-32tb, U7inh-32tb, X2idn, X2iedn, X2iezn, X8g
  - 스토리지 최적화: D3, D3en, I3en, I4g, I4i, I7i, I7ie, I8g, Im4gn, Is4gen
  - 가속 컴퓨팅: DL1, DL2q, F2, G4ad, G4dn, G5, G6, G6e, Gr6, Inf1, Inf2, P3dn, P4d, P4de, P5, P5e, P5en, P6-B200, Trn1, Trn1n, Trn2, Trn2u, VT1
  - 고성능 컴퓨팅: Hpc6a, Hpc6id, Hpc7a, Hpc7g
- 인스턴스가 동일한 리전에 있습니다.
- 인스턴스가 동일한 VPC 또는 피어링된 VPC에 있으며, 트래픽이 로드 밸런서나 전송 게이트웨이 같은 가상 네트워크 디바이스 또는 서비스를 통과하지 않습니다.

앞서 이 섹션에서 설명한 것처럼 AWS 보안 시설에서 나가기 전에 모든 트래픽에 대한 추가 암호화 계층이 물리적 계층에서 자동으로 제공됩니다.

AWS CLI를 사용하여 인스턴스 간에 전송 중인 트래픽을 암호화하는 인스턴스 유형을 보려면

다음 describe-instance-types 명령을 사용합니다.

```
aws ec2 describe-instance-types \
    --filters Name=network-info.encryption-in-transit-supported,Values=true \
    --query "InstanceTypes[*].[InstanceType]" \
    --output text | sort
```

AWS Outposts로의 암호화

Outposts는 AWS 홈 리전에 대한 서비스 링크라는 특정 네트워크 연결을 생성하고 선택적으로 사용자가 지정한 VPC 서브넷에 대한 프라이빗 연결을 생성합니다. 이러한 연결을 통한 모든 트래픽은 완전히 암호화됩니다. 자세한 내용은 AWS Outposts 사용 설명서에서 서비스 링크를 통한 연결 및 전송 중 암호화를 참조하세요.

원격 액세스 암호화

SSH 및 RDP 프로토콜은 직접 또는 EC2 인스턴스 연결을 통해 인스턴스에 원격으로 액세스할 수 있는 보안 통신 채널을 제공합니다. AWS Systems Manager Session Manager 및 Run Command를 사용하는 인스턴스에 대한 원격 액세스는 TLS 1.2를 사용하여 암호화되며, 연결을 생성하기 위한 요청은 SigV4를 사용하여 서명되고 AWS Identity and Access Management에 의해 인증되고 권한이 부여됩니다.

전송 계층 보안(TLS)과 같은 암호화 프로토콜을 사용하여 클라이언트와 Amazon EC2 인스턴스 간에 전송 중인 민감한 데이터를 암호화할 책임은 사용자에게 있습니다.

(Windows 인스턴스) EC2 인스턴스와 AWS API 엔드포인트 또는 기타 중요한 원격 네트워크 서비스 간에는 암호화된 연결만 허용해야 합니다. 아웃바운드 보안 그룹 또는 Windows 방화벽 규칙을 통해 이를 적용할 수 있습니다.

Amazon EC2의 인프라 보안

관리형 서비스인 Amazon Elastic Compute Cloud는 AWS 글로벌 네트워크 보안으로 보호됩니다. AWS 보안 서비스와 AWS의 인프라 보호 방법에 대한 자세한 내용은 AWS 클라우드 보안을 참조하세요. 인프라 보안에 대한 모범 사례를 사용하여 AWS 환경을 설계하려면 보안 원칙 AWS Well-Architected Framework의 인프라 보호를 참조하세요.

AWS에서 게시한 API 호출을 사용하여 네트워크를 통해 Amazon EC2에 액세스합니다. 고객은 다음을 지원해야 합니다.

- Transport Layer Security(TLS) TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- DHE(Ephemeral Diffie-Hellman) 또는 ECDHE(Elliptic Curve Ephemeral Diffie-Hellman)와 같은 완전 전송 보안(PFS)이 포함된 암호 제품군 Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

또한 요청은 액세스 키 ID 및 IAM 주체와 관련된 시크릿 액세스 키를 사용하여 서명해야 합니다. 또는 AWS Security Token Service(AWS STS)를 사용하여 임시 자격 증명을 생성하여 요청에 서명할 수 있습니다.

자세한 내용은 보안 부문 - AWS Well-Architected 프레임워크의 인프라 보호를 참조하세요.

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.

프라이빗 IP 주소를 사용하여 VPC에서 Amazon EC2 API를 호출하려면 AWS PrivateLink을(를) 사용합니다. 자세한 내용은 인터페이스 VPC 엔드포인트를 사용하여 Amazon EC2에 액세스 단원을 참조하십시오.

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 보세요.

- 보안 그룹을 사용하여 인스턴스에 대한 액세스를 제한합니다. 필요한 최소한의 네트워크 트래픽을 허용하는 규칙을 구성합니다. 예를 들어 회사 네트워크 주소 범위의 트래픽만 허용하거나 HTTPS와

같은 특정 프로토콜에 대해서만 트래픽을 허용할 수 있습니다. Windows 인스턴스의 경우 Windows 관리 트래픽과 최소한의 아웃바운드 연결을 허용합니다.

- 보안 그룹을 Amazon EC2 인스턴스에 대한 네트워크 액세스를 제어하는 기본 메커니즘으로 활용합니다. 필요한 경우 네트워크 ACL을 거의 사용하지 않고 상태 비저장, 거친 네트워크 제어를 제공합니다. 보안 그룹은 상태 저장 패킷 필터링을 수행하고 다른 보안 그룹을 참조하는 규칙을 만들 수 있기 때문에 네트워크 ACL보다 다재다능합니다. 그러나 네트워크 ACL은 특정한 트래픽 하위 집합을 거부하거나 상위 수준의 서브넷 가드레일을 제공하는 보조적 제어 장치로 효과를 발휘할 수 있습니다. 또한 네트워크 ACL은 전체 서브넷에 적용되므로 인스턴스가 올바른 보안 그룹 없이 실수로 시작될 경우 이를 심층 방어 기능으로 사용할 수 있습니다.
- [Windows 인스턴스] Windows 방화벽 설정을 그룹 정책 개체(GPO)로 중앙에서 관리하여 네트워크 제어를 더욱 강화합니다. 고객들은 종종 네트워크 트래픽을 더 잘 파악하고 보안 그룹 필터를 보완하기 위해 Windows 방화벽을 사용하며, 고급 규칙을 만들어 특정 애플리케이션이 네트워크에 액세스하지 못하도록 차단하거나 하위 집합 IP 주소에서 오는 트래픽을 필터링하기도 합니다. 예를 들어 Windows 방화벽은 특정 사용자 또는 애플리케이션만 EC2 메타데이터 서비스 IP 주소에 액세스할 수 있도록 제한할 수 있습니다. 또는 공용 서비스에서 보안 그룹을 사용하여 트래픽을 특정 포트로 제한하고 Windows 방화벽을 사용하여 명시적으로 차단된 IP 주소의 목록을 유지 관리할 수 있습니다.
- 인터넷에서 직접 액세스하면 안 되는 경우 프라이빗 서브넷을 인스턴스에 사용합니다. 프라이빗 서브넷에 있는 인스턴스에서 인터넷에 액세스하려면 Bastion Host 또는 NAT 게이트웨이를 사용합니다.
- [Windows instances] SSL/TLS를 통한 RDP 캡슐화와 같은 보안 관리 프로토콜을 사용합니다. 원격 데스크톱 게이트웨이 빠른 시작은 SSL/TLS를 사용하도록 RDP를 구성하는 등 원격 데스크톱 게이트웨이 배포의 모범 사례를 제공합니다.
- [Windows 인스턴스] Active Directory 또는 AWS Directory Service를 사용하여 Windows 인스턴스에 대한 사용자 및 그룹의 대화식 액세스를 중앙 집중식으로 엄격하게 제어 및 모니터링하고 로컬 사용자 권한을 피할 수 있습니다. 또한 도메인 관리자를 사용하지 않고 보다 세분화된 애플리케이션별 역할 기반 계정을 만들 수 있습니다. JEA(충분한 관리)를 사용하면 대화식 액세스 또는 관리자 액세스 없이도 Windows 인스턴스의 변경 사항을 관리할 수 있습니다. 또한 JEA에서는 인스턴스 관리에 필요한 Windows PowerShell 명령의 하위 집합에 대한 관리 액세스를 잠글 수 있습니다. 자세한 내용은 AWS 보안 모범 사례 백서의 “Amazon EC2에 대한 OS 수준 액세스 관리” 섹션을 참조하세요.
- [Windows 인스턴스] 시스템 관리자는 액세스 권한이 제한된 Windows 계정을 사용하여 일상적인 작업을 수행하고 특정한 구성 변경을 수행하는 데 필요한 경우에만 액세스 권한을 상승해야 합니다. 또한 절대적으로 필요한 경우에만 Windows 인스턴스에 직접 액세스합니다. 그 대신 EC2 실행 명령, SCCM(시스템 센터 구성 관리자), Windows PowerShell DSC 또는 SSM(Amazon EC2 Systems Manager)과 같은 중앙 구성 관리 시스템을 활용하여 Windows 서버에 변경 사항을 적용합니다.

- 필요한 최소한의 네트워크 경로로 Amazon VPC 서브넷 라우팅 테이블을 구성합니다. 예를 들어, 인터넷 게이트웨이로 가는 경로가 있는 서브넷에는 인터넷에 직접 액세스해야 하는 Amazon EC2 인스턴스만 배치하고, 가상 프라이빗 게이트웨이로 가는 경로가 있는 서브넷에는 내부 네트워크에 직접 액세스해야 하는 Amazon EC2 인스턴스만 배치합니다.
- 보안 그룹 또는 네트워크 인터페이스를 추가로 사용하여 일반 애플리케이션 트래픽과 별도로 Amazon EC2 인스턴스 관리 트래픽을 제어하고 감사하는 방법을 고려해 보십시오. 이렇게 하면 변경 제어를 위한 특별한 IAM 정책을 고객이 구현할 수 있으므로 보안 그룹 규칙 또는 자동화된 규칙 확인 스크립트의 변경 사항을 감사하기가 쉬워집니다. 여러 네트워크 인터페이스를 사용하면 호스트 기반 라우팅 정책을 생성하거나 네트워크 인터페이스의 할당된 서브넷에 따라 다른 VPC 서브넷 라우팅 규칙을 활용하는 기능 등 네트워크 트래픽을 제어할 수 있는 추가 옵션도 제공됩니다.
- AWS Virtual Private Network 또는 AWS Direct Connect를 사용하여 원격 네트워크에서 VPC로 프라이빗 연결을 설정합니다. 자세한 내용은 네트워크-Amazon VPC 연결 옵션을 참조하세요.
- VPC 흐름 로그를 사용하여 인스턴스에 도달하는 트래픽을 모니터링합니다.
- GuardDuty Malware Protection을 사용하여 인스턴스에서 워크로드를 손상시키고 리소스를 악의적인 용도로 재사용하며 이터에 대한 무단으로 액세스할 수 있는 악성 소프트웨어를 나타내는 의심스러운 동작을 식별합니다.
- GuardDuty Runtime Monitoring을 사용하여 인스턴스에 대한 잠재적 위협을 식별하고 이에 대응합니다. 자세한 내용은 How Runtime Monitoring works with Amazon EC2 instances를 참조하세요.
- AWS Security Hub, Reachability Analyzer 또는 Network Access Analyzer를 사용하여 인스턴스에서 의도하지 않은 네트워크 액세스를 검사합니다.
- EC2 Instance Connect를 사용하여 SSH 키를 공유하고 관리할 필요 없이 SSH(Secure Shell)를 통해 인스턴스에 연결합니다.
- AWS Systems Manager Session Manager를 사용하면 인바운드 SSH 또는 RDP 포트를 열고 키 페어를 관리하는 대신 원격으로 인스턴스에 액세스할 수 있습니다.
- AWS Systems Manager 실행 명령을 사용하여 인스턴스에 연결하는 대신 일반적인 관리 작업을 자동화할 수 있습니다.
- [Windows 인스턴스] Windows OS 역할 및 Microsoft 비즈니스 애플리케이션은 대부분 IIS 내 IP 주소 범위 제한, Microsoft SQL Server의 TCP/IP 필터링 정책 및 Microsoft Exchange의 연결 필터 정책과 같은 향상된 기능을 제공합니다. 애플리케이션 계층 내 네트워크 제한 기능은 중요한 비즈니스 애플리케이션 서버에 대한 추가 방어 계층을 제공할 수 있습니다.

Amazon VPC는 게이트웨이, 프록시 서버, 네트워크 모니터링 옵션과 같은 추가 네트워크 보안 제어를 지원합니다. 자세한 내용은 Amazon VPC 사용 설명서의 네트워크 트래픽 제어를 참조하세요.

Amazon EC2의 복원성

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 중심으로 구축됩니다. 리전은 물리적으로 분리되고 격리된 다수의 가용 영역을 제공하며, 이러한 영역은 짧은 지연 시간, 높은 처리량 및 높은 중복성을 갖춘 네트워크를 통해 연결되어 있습니다. 가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 극복 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

더 먼 지역적 거리를 두고 데이터 또는 애플리케이션을 복제해야 하는 경우 AWS Local Zones를 사용하세요. AWS Local Zone은 사용자와 지리적으로 근접한 AWS 리전의 확장입니다. Local Zones는 인터넷에 대한 자체 연결을 가지고 있으며 AWS Direct Connect를 지원합니다. 모든 AWS 리전과 마찬가지로 AWS Local Zones는 다른 AWS 영역과 완벽히 격리되어 있습니다.

AWS Local Zones에서 데이터나 애플리케이션을 복제해야 하는 경우 AWS에서는 다음 영역 중 하나를 장애 조치 영역으로 사용하는 것이 좋습니다.

- 다른 로컬 영역
- 상위 영역이 아닌 리전의 가용 영역입니다. describe-availability-zones 명령을 사용하여 상위 영역을 볼 수 있습니다.

AWS 리전 및 가용 영역에 대한 자세한 내용은 AWS 글로벌 인프라를 참조하십시오.

AWS 글로벌 인프라 외에도 Amazon EC2는 데이터 복원력을 지원하기 위해 다음과 같은 기능을 제공합니다.

- 리전 간 AMI 복사
- 리전 간 EBS 스냅샷 복사
- Amazon Data Lifecycle Manager를 사용하여 EBS-backed AMI 자동화
- Amazon Data Lifecycle Manager를 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 규정 준수 확인

AWS 서비스가 특정 규정 준수 프로그램의 범위에 포함되는지 알아보려면 규정 준수 프로그램 제공 범위 내 AWS 서비스를 참조하고 관심 있는 규정 준수 프로그램을 선택하세요. 일반적인 정보는 AWS 규정 준수 프로그램을 참조하세요.

AWS Artifact(을)를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다. 자세한 내용은 AWS Artifact에서 보고서 다운로드를 참조하세요.

AWS 서비스 사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률과 규정에 따라 결정됩니다.AWS에서는 규정 준수를 지원할 다음과 같은 리소스를 제공합니다.

- 보안 규정 준수 및 거버넌스 - 이러한 솔루션 구현 가이드에서는 아키텍처 고려 사항을 설명하고 보안 및 규정 준수 기능을 배포하는 단계를 제공합니다.
- HIPAA 적격 서비스 참조-HIPAA 적격 서비스가 나열되어 있습니다. 모든 AWS 서비스에 HIPAA 자격이 있는 것은 아닙니다.
- AWS 규정 준수 리소스 - 고객 조직이 속한 산업 및 위치에 적용될 수 있는 워크북 및 가이드 컬렉션입니다.
- AWS 고객 규정 준수 가이드 - 규정 준수의 관점에서 공동 책임 모델을 이해합니다. 이 가이드에서는 AWS 서비스를 보호하기 위한 모범 사례를 요약하고 여러 프레임워크(미국 표준 기술 연구소(NIST), 결제 카드 산업 보안 표준 위원회(PCI), 국제 표준화기구(ISO) 등)에서 보안 컨트롤에 대한 지침을 매핑합니다.
- AWS Config 개발자 가이드의 규칙을 사용하여 리소스 평가 – AWS Config 서비스는 내부 사례, 산업 지침 및 규제에 대한 리소스 구성의 준수 상태를 평가합니다.
- AWS Security Hub – 이 AWS 서비스(은)는 AWS 내 보안 상태에 대한 포괄적인 보기를 제공합니다. Security Hub는 보안 컨트롤을 사용하여 AWS리소스를 평가하고 보안 업계 표준 및 모범 사례에 대한 규정 준수를 확인합니다. 지원되는 서비스 및 제어 목록은 Security Hub 제어 참조를 참조하세요.
- Amazon GuardDuty - 이 AWS 서비스는 의심스럽고 악의적인 활동이 있는지 환경을 모니터링하여 AWS 계정, 워크로드, 컨테이너 및 데이터에 대한 잠재적 위협을 탐지합니다. GuardDuty는 특정 규정 준수 프레임워크에서 요구하는 침입 탐지 요구 사항을 충족하여 PCI DSS와 같은 다양한 규정 준수 요구 사항을 따르는 데 도움을 줄 수 있습니다.
- AWS Audit Manager - 이 AWS 서비스은(는) AWS 사용을 지속해서 감사하여 위험을 관리하고 규정 및 업계 표준을 준수하는 방법을 간소화할 수 있도록 지원합니다.

Amazon EC2의 자격 증명 및 액세스 관리

AWS Identity and Access Management(IAM)은 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있도록 지원하는 AWS 서비스입니다. IAM 관리자는 어떤 사용자가 Amazon EC2 리소스를 사용할 수 있도록 인증(로그인)되고 권한이 부여(권한 있음)될 수 있는지 제어합니다. IAM은 추가 비용 없이 사용할 수 있는 AWS 서비스입니다.

보안 자격 증명은 AWS의 서비스에서 사용자를 식별하고 Amazon EC2 리소스와 같은 AWS 리소스의 액세스를 허가합니다. Amazon EC2 및 IAM의 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 또는 제한 사용을 허가할 수 있습니다.

개발자는 IAM 역할을 사용하여 EC2 인스턴스에서 실행하는 애플리케이션에 필요한 보안 자격 증명을 관리할 수 있습니다. 인스턴스에 IAM 역할을 연결하면 인스턴스에서 실행되는 애플리케이션이 인스턴스 메타데이터 서비스(IMDS)에서 자격 증명을 검색할 수 있습니다.

IAM을 사용하는 AWS 리소스를 보호하기 위한 모범 사례의 경우 IAM 사용 설명서의 IAM 보안 모범 사례를 참조하세요.

내용

- Amazon EC2의 ID 기반 정책
- Amazon EC2 API에 대한 액세스를 제어하는 정책 예제
- Amazon EC2 콘솔에 대한 액세스를 제어하는 정책 예제
- Amazon EC2에 대한 AWS 관리형 정책
- Amazon EC2의 IAM 역할

Amazon EC2의 ID 기반 정책

기본적으로 사용자에게는 Amazon EC2 리소스를 생성 또는 수정하거나 Amazon EC2 API, Amazon EC2 콘솔 또는 CLI를 사용하여 태스크를 수행할 권한이 없습니다. 사용자에게 리소스 생성 또는 수정 및 태스크 수행을 허용하려면 사용자에게 필요한 특정 리소스 및 API 작업을 사용할 권한을 부여하는 IAM 정책을 생성하고, 해당 권한을 필요로 하는 사용자, 그룹 또는 IAM 역할에 정책을 연결해야 합니다.

사용자, 사용자 그룹 또는 역할에 정책을 연결하면 지정된 리소스에 대해 지정된 태스크를 수행할 권한이 허용되거나 거부됩니다. IAM 정책에 대한 자세한 내용은 IAM 사용 설명서의 IAM의 정책 및 권한을 참조하세요. IAM 정책 관리 및 생성에 대한 자세한 내용은 IAM 정책 관리를 참조하세요.

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.

시작을 위해 Amazon EC2의 AWS 관리형 정책이 요구 사항을 충족하는지 확인할 수 있습니다. 아니면 고유한 사용자 지정 정책을 생성할 수 있습니다. 자세한 내용은 the section called “AWS 관리형 정책” 섹션을 참조하세요.

내용

- 정책 구문
- Amazon EC2 작업
- Amazon EC2 API 작업에 지원되는 리소스 수준 권한
- Amazon EC2의 Amazon 리소스 이름(ARN)
- Amazon EC2에 사용되는 조건 키
- 속성 기반 액세스를 사용하여 액세스 제어
- 사용자, 그룹 및 역할에 권한 부여
- 사용자에게 필요한 권한이 있는지 확인

정책 구문

IAM 정책은 하나 이상의 문으로 구성된 JSON 문서입니다. 각 명령문의 구조는 다음과 같습니다.

```
{
  "Statement":[{
    "Effect":"effect",
    "Action":"action",
    "Resource":"arn",
    "Condition":{
      "condition":{
        "key":"value"
        }
      }
    }
  ]
```

```
}
```

명령문을 이루는 요소는 다양합니다.

- 효과(Effect): 효과(effect)는 `Allow` 또는 `Deny`일 수 있습니다. 기본적으로 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 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 섹션을 참조하세요.
- Condition: Condition은 선택 사항으로서 정책이 적용되는 시점을 제어하는 데 사용할 수 있습니다. Amazon EC2에 조건을 지정하는 방법에 대한 자세한 내용은 Amazon EC2에 사용되는 조건 키 섹션을 참조하세요.

정책에 대한 자세한 내용은 IAM 사용 설명서의 IAM JSON 정책 참조를 참조하세요. Amazon EC2용 IAM 정책 명령문 예제는 Amazon EC2 API에 대한 액세스를 제어하는 정책 예제 섹션을 참조하세요.

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*"
```

> **Note**
>
> 현재 Amazon EC2 Describe* API 작업은 리소스 수준 권한을 지원하지 않습니다. Amazon EC2용 리소스 수준 권한에 대한 자세한 내용은 Amazon EC2의 ID 기반 정책 섹션을 참조하세요.

모든 Amazon EC2 API 작업을 지정하려면 다음과 같이 * 와일드카드를 사용합니다.

```
"Action": "ec2:*"
```

Amazon EC2 작업 목록을 보려면 서비스 권한 부여 참조에서 Amazon EC2에서 정의한 작업을 참조하세요.

Amazon EC2 API 작업에 지원되는 리소스 수준 권한

리소스 수준 권한이란 사용자가 작업을 수행할 수 있는 리소스를 지정하는 기능을 말합니다. 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 정보는 Amazon EC2의 Amazon 리소스 이름(ARN) 섹션을 참조하세요. 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 Amazon EC2에 사용되는 작업, 리소스 및 조건 키를 참조하세요.

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 Amazon EC2 리소스 태그 지정에 대한 권한 부여 섹션을 참조하세요.

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-id]:resourceType/resourcePath
```

service

서비스(예: ec2)입니다.

region

리소스의 리전(예: us-east-1)입니다.

account-id

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 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 Amazon EC2에서 정의한 리소스 유형을 참조하세요.

Amazon EC2에 사용되는 조건 키

정책 명령문에서 정책이 적용되는 시점을 제어하는 조건을 지정할 수 있습니다. 각 조건에는 하나 이상의 키-값 쌍이 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 Amazon EC2의 조건 키를 참조하세요. Amazon EC2는 AWS 전체 범위 조건 키도 구현합니다. 자세한 내용은 IAM 사용 설명서의 모든 요청에서 사용 가능한 정보를 참조하세요.

모든 Amazon EC2 작업은 `aws:RequestedRegion` 및 `ec2:Region` 조건 키를 지원합니다. 자세한 내용은 예: 특정 리전에 대한 액세스 제한 섹션을 참조하세요.

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. 자세한 내용은 IAM 사용 설명서의 IAM 정책 요소: 변수 및 태그 섹션을 참조하세요.

> **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.

조건 키

- ec2:Attribute 조건 키
- ec2:ResourceID 조건 키
- ec2:SourceInstanceARN 조건 키

ec2:Attribute 조건 키

`ec2:Attribute` 조건 키는 리소스의 속성으로 액세스를 필터링하는 조건에 사용할 수 있습니다.

이 조건 키는 문자열 또는 정수와 같은 기본 데이터 유형의 속성만 지원하거나 ModifyImageAttribute API 작업의 Description 또는 ImdsSupport 객체와 같이 Value 속성만 있는 복합 AttributeValue 객체를 지원합니다. 조건 키는 ModifyImageAttribute API 작업의 LaunchPermission 객체와 같이 여러 속성을 가진 복합 객체에 사용할 수 없습니다.

예를 들어 다음 정책은 ec2:Attribute/Description 조건 키를 사용하여 ModifyImageAttribute API 작업의 복잡한 Description 객체에 의한 액세스를 필터링합니다. 조건 키는 이미지의 설명을 Production 또는 Development로 수정하는 요청만 허용합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:ModifyImageAttribute",
      "Resource": "arn:aws:ec2:us-east-1::image/ami-*",
      "Condition": {
        "StringEquals": {
          "ec2:Attribute/Description": [
            "Production",
            "Development"
          ]
        }
      }
    }
  ]
}
```

다음 예제 정책은 ec2:Attribute 조건 키를 사용하여 ModifyImageAttribute API 작업의 기본 Attribute 속성으로 액세스를 필터링합니다. 조건 키는 이미지 설명을 수정하려는 모든 요청을 거부합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "ec2:ModifyImageAttribute",
      "Resource": "arn:aws:ec2:us-east-1::image/ami-*",
      "Condition": {
        "StringEquals": {
          "ec2:Attribute": "Description"
        }
      }
    }
  ]
}
```

ec2:ResourceID 조건 키

지정된 API 작업과 함께 다음 ec2:*Resource*ID 조건 키를 사용하는 경우 조건 키 값은 API 작업에 의해 생성되는 결과 리소스를 지정하는 데 사용됩니다. ec2:*Resource*ID 조건 키는 API 요청에 지정된 소스 리소스를 지정하는 데 사용할 수 없습니다. 지정된 API와 함께 다음 ec2:*Resource*ID 조건 키 중 하나를 사용하는 경우 항상 와일드카드(*)를 지정해야 합니다. 다른 값을 지정하는 경우 조건은 런타임 중에 항상 *(으)로 해석됩니다. 예를 들어 CopyImage API와 함께 ec2:ImageId 조건 키를 사용하려면 다음과 같이 조건 키를 지정해야 합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:CopyImage",
      "Resource": "arn:aws:ec2:us-east-1::image/ami-*",
      "Condition": {
        "StringEquals": {
          ec2:ImageID": "*"
        }
      }
    }
  ]
}
```

다음 API 작업에는 이러한 조건 키를 사용하지 않는 것이 좋습니다.

- ec2:DhcpOptionsID – CreateDhcpOptions
- ec2:ImageID – CopyImage, CreateImage, ImportImage, RegisterImage
- ec2:InstanceID – RunInstances 및 ImportInstance
- ec2:InternetGatewayID – CreateInternetGateway
- ec2:NetworkAclID – CreateNetworkAcl
- ec2:NetworkInterfaceID – CreateNetworkInterface
- ec2:PlacementGroupName – CreatePlacementGroup
- ec2:RouteTableID – CreateRouteTable
- ec2:SecurityGroupID – CreateSecurityGroup
- ec2:SnapshotID – CopySnapshot, CreateSnapshot, CreateSnapshots, ImportSnapshots

- ec2:SubnetID – CreateSubnet
- ec2:VolumeID – CreateVolume 및 ImportVolume
- ec2:VpcID – CreateVpc
- ec2:VpcPeeringConnectionID – CreateVpcPeeringConnection

특정 리소스 ID를 기준으로 액세스를 필터링하려면 다음과 같이 Resource 정책 요소를 사용하는 것이 좋습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:CopyImage",
      "Resource": "arn:aws:ec2:us-east-1::image/ami-01234567890abcdef"
    }
  ]
}
```

ec2:SourceInstanceARN 조건 키

ec2:SourceInstanceARN을 사용하여 요청이 이루어진 인스턴스의 ARN을 지정합니다. 이는 AWS 글로벌 조건 키이므로 Amazon EC2 이외의 서비스에서 사용할 수 있습니다. 정책 예제는 예: 특정 인스턴스에서 다른 AWS 서비스의 리소스를 보는 것을 허용을 참조하십시오.

속성 기반 액세스를 사용하여 액세스 제어

사용자에게 EC2 리소스 사용 권한을 부여하는 IAM 정책을 생성할 때 정책의 Condition 요소에 태그 정보를 포함시키면 태그를 기반으로 액세스를 제어할 수 있습니다. 이를 속성 기반 액세스 제어(ABAC)라고 합니다. ABAC를 통해 사용자는 수정, 사용 또는 삭제할 수 있는 리소스를 더욱 정확하게 제어할 수 있습니다. 자세한 내용은 AWS용 ABAC란 무엇입니까? 단원을 참조하세요.

예를 들어 사용자가 인스턴스를 종료할 수 있도록 허용하지만 인스턴스에 environment=production 태그가 있는 경우 작업을 거부하는 정책을 만들 수 있습니다. 이렇게 하려면 aws:ResourceTag 조건 키를 사용하여 리소스에 연결된 태그를 기반으로 리소스에 대한 액세스를 허용하거나 거부합니다.

```
"StringEquals": { "aws:ResourceTag/environment": "production" }
```

Amazon EC2 API 작업에서 `aws:ResourceTag` 조건 키를 사용한 액세스 제어를 지원하는지 알아보려면 Amazon EC2에 사용되는 작업, 리소스 및 조건 키를 참조하세요. `Describe` 작업은 리소스 수준 권한을 지원하지 않기 때문에 조건 없이 별도의 명령문에 지정해야 합니다.

예제 IAM 정책은 Amazon EC2 API에 대한 액세스를 제어하는 정책 예제 섹션을 참조하세요.

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.

사용자, 그룹 및 역할에 권한 부여

액세스 권한을 제공하려면 사용자, 그룹 또는 역할에 권한을 추가하세요:

- AWS IAM Identity Center의 사용자 및 그룹:

  권한 세트를 생성합니다. AWS IAM Identity Center 사용 설명서의 권한 세트 생성의 지침을 따릅니다.
- 보안 인증 공급자를 통해 IAM에서 관리되는 사용자:

  ID 페더레이션을 위한 역할을 생성합니다. IAM 사용 설명서의 Create a role for a third-party identity provider (federation)의 지침을 따릅니다.
- IAM 사용자:
  - 사용자가 맡을 수 있는 역할을 생성합니다. IAM 사용 설명서에서 Create a role for an IAM user의 지침을 따릅니다.
  - (권장되지 않음)정책을 사용자에게 직접 연결하거나 사용자를 사용자 그룹에 추가합니다. IAM 사용 설명서에서 사용자(콘솔)에 권한 추가의 지침을 따르세요.

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 사용자를 생성하고 앞서 생성한 IAM 정책을 연결하여 사용자를 테스트합니다. 그런 다음 테스트 사용자 자격으로 요청을 수행합니다.

테스트 중인 Amazon EC2 작업에서 리소스를 생성하거나 수정하는 경우 `DryRun` 파라미터를 사용하여 요청을 제출하거나 AWS CLI 명령을 `--dry-run` 옵션과 함께 실행해야 합니다. 이렇게 하면 호출 시 권한 부여 확인은 완료되지만 작업은 완료되지 않습니다. 예를 들어 인스턴스를 실제

로 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 참조의 DecodeAuthorizationMessage와 decode-authorization-message를 참조하세요.

Amazon EC2 API에 대한 액세스를 제어하는 정책 예제

IAM 정책을 사용하여 Amazon EC2를 사용하는 데 필요한 권한을 사용자에게 부여할 수 있습니다. 단계별 지침은 IAM 사용 설명서의 IAM 정책 생성을 참조하세요.

다음 예제는 사용자에게 Amazon EC2 사용 권한을 부여하는 데 사용할 수 있는 정책 명령문을 보여줍니다. 이러한 정책은 AWS CLI 또는 AWS SDK를 사용하는 요청에 맞게 설계되었습니다. 다음 예제에서는 자신의 정보로 각각의 *### ## ## ###*를 바꿉니다.

예시

- 예: 읽기 전용 액세스
- 예: 특정 리전에 대한 액세스 제한
- 인스턴스 작업
- 인스턴스 시작(RunInstances)
- 스팟 인스턴스 작업
- 예: 예약 인스턴스 작업
- 예: 태그 리소스
- 예: IAM 역할 작업
- 예: 라우팅 테이블 작업

- 예: 특정 인스턴스에서 다른 AWS 서비스의 리소스를 보는 것을 허용
- 예: 시작 템플릿 작업
- 인스턴스 메타데이터 작업
- Amazon EBS 볼륨 및 스냅샷 작업

Amazon EC2 콘솔 작업과 관련된 예제 정책은 Amazon EC2 콘솔에 대한 액세스를 제어하는 정책 예제 섹션을 참조하세요.

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 Amazon EC2에 사용되는 작업, 리소스 및 조건 키를 참조하세요.

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": "*"
    }
   ]
}
```

예: 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
- 예제: 모든 인스턴스를 설명할 수 있지만 특정 인스턴스만 중지, 시작 및 종료

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 있습니

다. API 작업이 리소스 수준 권한을 지원하지 않는 경우에도 * 와일드카드가 필요합니다. Amazon EC2 API 작업에 사용할 수 있는 ARN에 대한 자세한 내용은 Amazon EC2에 사용되는 작업, 리소스 및 조건 키를 참조하세요.

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-1의 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 Amazon EC2에 사용되는 작업, 리소스 및 조건 키를 참조하세요.

두 번째 명령문의 StopInstances 및 StartInstances 작업에는 리소스 수준 권한이 사용되었습니다. Resource 요소의 ARN에 의해 특정 인스턴스가 지정되었습니다.

세 번째 명령문은 지정된 AWS 계정에 속하는 미국 동부(버지니아 북부) 리전(us-east-1)에서 "purpose=test" 태그가 있는 모든 인스턴스를 종료할 수 있도록 합니다. 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:account-id:instance/i-1234567890abcdef0",
        "arn:aws:ec2:us-east-1:account-id:instance/i-0598c7d356eba48d7"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "ec2:TerminateInstances",
      "Resource": "arn:aws:ec2:us-east-1:account-id:instance/*",
      "Condition": {
         "StringEquals": {
            "aws:ResourceTag/purpose": "test"
         }
      }
   }

   ]
}
```

인스턴스 시작(RunInstances)

RunInstances API 작업은 하나 이상의 온디맨드 인스턴스 또는 하나 이상의 스팟 인스턴스를 시작합니다. RunInstances는 AMI를 필요로 하며 인스턴스를 생성합니다. 사용자는 요청에 키 페어와 보안 그룹을 지정할 수 있습니다. VPC로 시작하는 경우 서브넷을 입력 받아 네트워크 인터페이스를 생성합

니다. Amazon EBS 지원 AMI에서 시작하면 볼륨이 생성됩니다. 따라서 사용자가 이러한 Amazon EC2 리소스를 사용할 권한을 가지고 있어야 합니다. 사용자가 RunInstances에서 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 Amazon EC2에 사용되는 작업, 리소스 및 조건 키를 참조하세요.

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 섹션을 참조하세요.

리소스

- AMI
- 인스턴스 유형
- 서브넷
- EBS 볼륨
- Tags
- 시작 템플릿의 태그
- 탄력적 GPU
- 시작 템플릿

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-id:instance/*",
        "arn:aws:ec2:region:account-id:volume/*",
        "arn:aws:ec2:region:account-id:key-pair/*",
        "arn:aws:ec2:region:account-id:security-group/*",
        "arn:aws:ec2:region:account-id:subnet/*",
        "arn:aws:ec2:region:account-id:network-interface/*"
      ]
    }
   ]
}
```

또는 다음 정책으로 사용자가 Amazon 소유의 모든 AMI 또는 신뢰할 수 있고 확인된 특정 파트너에서 인스턴스를 시작하도록 허용할 수 있습니다. 첫 번째 명령문의 Condition 요소는 ec2:Owner가 amazon인지 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-id:instance/*",
         "arn:aws:ec2:region:account-id:subnet/*",
         "arn:aws:ec2:region:account-id:volume/*",
         "arn:aws:ec2:region:account-id:network-interface/*",
         "arn:aws:ec2:region:account-id:key-pair/*",
         "arn:aws:ec2:region:account-id:security-group/*"
         ]
      }
```

```
    ]
}
```

인스턴스 유형

다음 정책은 사용자가 t2.micro 및 t2.small 인스턴스 유형만 사용하여 인스턴스를 시작하도록 허용하므로 비용 통제에 도움이 됩니다. 첫 번째 명령문의 Condition 요소에서 ec2:InstanceType이 t2.micro 또는 t2.small인지 여부를 테스트하므로 더욱 큰 인스턴스는 시작할 수 없습니다.

```
{
   "Version": "2012-10-17",
   "Statement": [
        {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:region:account-id: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-id:subnet/*",
         "arn:aws:ec2:region:account-id:network-interface/*",
         "arn:aws:ec2:region:account-id:volume/*",
         "arn:aws:ec2:region:account-id:key-pair/*",
         "arn:aws:ec2:region:account-id: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-id: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-id:network-interface/*",
         "arn:aws:ec2:region:account-id:instance/*",
         "arn:aws:ec2:region:account-id:subnet/*",
         "arn:aws:ec2:region:account-id:volume/*",
         "arn:aws:ec2:region:account-id:key-pair/*",
         "arn:aws:ec2:region:account-id: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-id:subnet/subnet-12345678",
        "arn:aws:ec2:region:account-id:network-interface/*",
        "arn:aws:ec2:region:account-id:instance/*",
        "arn:aws:ec2:region:account-id:volume/*",
        "arn:aws:ec2:region::image/ami-*",
        "arn:aws:ec2:region:account-id:key-pair/*",
        "arn:aws:ec2:region:account-id: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-id:network-interface/*"
      ],
      "Condition": {
         "ArnNotEquals": {
            "ec2:Subnet": "arn:aws:ec2:region:account-id:subnet/subnet-12345678"
         }
      }
   },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:region::image/ami-*",
         "arn:aws:ec2:region:account-id:network-interface/*",
         "arn:aws:ec2:region:account-id:instance/*",
         "arn:aws:ec2:region:account-id:subnet/*",
         "arn:aws:ec2:region:account-id:volume/*",
         "arn:aws:ec2:region:account-id:key-pair/*",
```

```
            "arn:aws:ec2:region:account-id: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

생성 시 인스턴스에 태그 지정

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 RunInstances의 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 Amazon EC2 리소스 태그 지정에 대한 권한 부여 섹션을 참조하세요.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "ec2:RunInstances"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
         "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:us-east-1:account-id:instance/*",
      "Condition": {
         "StringEquals": {
             "ec2:CreateAction" : "RunInstances"
          }
       }
    }
  ]
}
```

생성 시 인스턴스 및 볼륨에 특정 태그를 사용하여 태그 지정

다음 정책에는 태그 aws:RequestTag 및 RunInstances를 사용하여 environment=production에 의해 생성되는 인스턴스와 볼륨에 사용자가 태그를 지정해야 하는

purpose=webserver 조건 키가 포함됩니다. 사용자가 이 특정 키들을 전달하지 않거나 태그를 전혀 지정하지 않으면 요청은 실패합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
   {
      "Effect": "Allow",
      "Action": [
         "ec2:RunInstances"
      ],
      "Resource": [
         "arn:aws:ec2:region::image/*",
         "arn:aws:ec2:region:account-id:subnet/*",
         "arn:aws:ec2:region:account-id:network-interface/*",
         "arn:aws:ec2:region:account-id:security-group/*",
         "arn:aws:ec2:region:account-id:key-pair/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
         "ec2:RunInstances"
      ],
      "Resource": [
          "arn:aws:ec2:region:account-id:volume/*",
          "arn:aws:ec2:region:account-id:instance/*"
      ],
      "Condition": {
         "StringEquals": {
             "aws:RequestTag/environment": "production" ,
             "aws:RequestTag/purpose": "webserver"
          }
       }
    },
    {
      "Effect": "Allow",
      "Action": [
         "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:region:account-id:*/*",
      "Condition": {
         "StringEquals": {
```

```
            "ec2:CreateAction" : "RunInstances"
         }
      }
   }
  ]
}
```

생성 시 인스턴스 및 볼륨에 하나 이상의 특정 태그를 사용하여 태그 지정

다음 정책은 ForAnyValue 조건에서 aws:TagKeys 변경자를 사용하여 요청에서 적어도 하나의 태그가 지정되어야 하고 태그에 키 environment 또는 webserver가 포함되어야 함을 표시합니다. 태그는 인스턴스와 볼륨에 모두 적용되어야 합니다. 요청에서 어떤 태그 값도 지정할 수 있습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
   {
      "Effect": "Allow",
      "Action": [
         "ec2:RunInstances"
      ],
      "Resource": [
         "arn:aws:ec2:region::image/*",
         "arn:aws:ec2:region:account-id:subnet/*",
         "arn:aws:ec2:region:account-id:network-interface/*",
         "arn:aws:ec2:region:account-id:security-group/*",
         "arn:aws:ec2:region:account-id:key-pair/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
          "ec2:RunInstances"
      ],
      "Resource": [
          "arn:aws:ec2:region:account-id:volume/*",
          "arn:aws:ec2:region:account-id:instance/*"
      ],
      "Condition": {
          "ForAnyValue:StringEquals": {
              "aws:TagKeys": ["environment","webserver"]
          }
       }
```

```
    },
    {
      "Effect": "Allow",
      "Action": [
          "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:region:account-id:*/*",
      "Condition": {
          "StringEquals": {
              "ec2:CreateAction" : "RunInstances"
          }
       }
    }
  ]
}
```

인스턴스를 생성할 때 태그를 지정하는 경우 특정 태그를 사용해야 함

다음 정책에서는 요청에서 태그를 지정할 필요가 없지만 지정하는 경우, 태그는 purpose=test여야 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-id:*/*",
      "Condition": {
         "StringEquals": {
             "aws:RequestTag/purpose": "test",
             "ec2:CreateAction" : "RunInstances"
          },
```

```
          "ForAllValues:StringEquals": {
              "aws:TagKeys": "purpose"
          }
       }
    }
  ]
}
```

RunInstances의 경우 생성 시 호출자가 태그를 지정할 수 없도록 하려면

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
                "arn:aws:ec2:us-east-1:*:security-group/*",
                "arn:aws:ec2:us-east-1:*:key-pair/*",
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*",
                "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
            ]
        },
        {
            "Sid": "VisualEditor0",
            "Effect": "Deny",
            "Action": "ec2:CreateTags",
            "Resource": "*"
        }
    ]
}
```

spot-instances-request에 대해서만 특정 태그를 허용합니다. 여기서도 예외적인 비일관성이 적용됩니다. 일반적인 상황에서는 태그를 지정하지 않으면 인증되지 않습니다. spot-instances-request의 경

우 spot-instances-request 태그가 없으면 이 정책이 평가되지 않으므로 태그가 지정되지 않은 Spot on Run 요청이 성공합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
                "arn:aws:ec2:us-east-1:*:security-group/*",
                "arn:aws:ec2:us-east-1:*:key-pair/*",
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*",
            ]
        },
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:us-east-1:*:spot-instances-request/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/environment": "production"
                }
            }
        }
    ]
}
```

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 인스턴스

에 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-id:launch-template/
lt-09477bcd97b0d310e"
          },
          "Bool": {
             "ec2:IsLaunchTemplateResource": "true"
          }
       }
    },
    {
      "Effect": "Allow",
      "Action": [
         "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:region:account-id: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 인스턴스가 eg1.medium 또는 eg1.large 탄력적 GPU 유형을 사용하는지 확인합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
             {
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:*:account-id:elastic-gpu/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:Region": "us-east-2",
                    "ec2:ElasticGpuType": [
                        "eg1.medium",
                        "eg1.large"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:*::image/ami-*",
                "arn:aws:ec2:*:account-id:network-interface/*",
                "arn:aws:ec2:*:account-id:instance/*",
                "arn:aws:ec2:*:account-id:subnet/*",
                "arn:aws:ec2:*:account-id:volume/*",
                "arn:aws:ec2:*:account-id:key-pair/*",
                "arn:aws:ec2:*:account-id: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-id: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-id: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 템플릿에

서만 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-id:subnet/*",
                      "arn:aws:ec2:region:account-id:network-interface/*" ],
      "Condition": {
         "ArnLike": {
             "ec2:LaunchTemplate": "arn:aws:ec2:region:account-id:launch-template/*"
          }
       }
    },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": ["arn:aws:ec2:region:account-id:subnet/*",
                   "arn:aws:ec2:region:account-id:network-interface/*" ],
      "Condition": {
         "ArnLike": {
             "ec2:LaunchTemplate": "arn:aws:ec2:region:account-id: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-id:launch-template/*",
      "Condition": {
         "ArnLike": {
             "ec2:LaunchTemplate": "arn:aws:ec2:region:account-id:launch-template/*"
          },
         "Bool": {
             "ec2:IsLaunchTemplateResource": "true"
          }
       }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": "arn:aws:ec2:region:account-id:launch-template/*",
      "Condition": {
       "StringEquals": {
           "aws:ResourceTag/Purpose": "Webservers"
        }
       }
     }
  ]
}
```

스팟 인스턴스 작업

RunInstances 작업을 사용하여 스팟 인스턴스 요청을 생성하고 생성 시 스팟 인스턴스 요청을 태깅할 수 있습니다. RunInstances에 대해 지정할 리소스는 `spot-instances-request`입니다.

`spot-instances-request` 리소스는 다음과 같이 IAM 정책에서 평가됩니다.

- 생성 시 스팟 인스턴스 요청을 태깅하지 않으면 Amazon EC2가 RunInstances 문에서 `spot-instances-request` 리소스를 평가하지 않습니다.
- 생성 시 스팟 인스턴스 요청을 태깅하면 Amazon EC2가 RunInstances 문에서 `spot-instances-request` 리소스를 평가합니다.

따라서 `spot-instances-request` 리소스의 경우 IAM 정책에 다음 규칙이 적용됩니다.

- RunInstances를 사용하여 스팟 인스턴스 요청을 생성하고 생성 시 스팟 인스턴스 요청을 태깅하지 않으려는 경우 `spot-instances-request` 리소스를 명시적으로 허용할 필요가 없습니다. 호출이 성공합니다.
- RunInstances를 사용하여 스팟 인스턴스 요청을 생성하고 생성 시 스팟 인스턴스 요청을 태깅하려는 경우 RunInstances allow 문에 `spot-instances-request` 리소스를 포함해야 합니다. 그러지 않으면 호출이 실패합니다.
- RunInstances를 사용하여 스팟 인스턴스 요청을 생성하고 생성 시 스팟 인스턴스 요청을 태깅하려는 경우 CreateTags allow 문에서 `spot-instances-request` 리소스 또는 * 와일드카드를 지정해야 합니다. 그러지 않으면 호출이 실패합니다.

RunInstances 또는 RequestSpotInstances를 사용하여 스팟 인스턴스를 요청할 수 있습니다. 다음 예제 IAM 정책은 RunInstances를 사용하여 스팟 인스턴스를 요청할 때만 적용됩니다.

예: RunInstances를 사용한 스팟 인스턴스 요청

다음 정책은 사용자가 RunInstances 작업을 사용하여 스팟 인스턴스를 요청할 수 있도록 허용합니다. RunInstances에서 생성되는 `spot-instances-request` 리소스는 스팟 인스턴스를 요청합니다.

> **Note**
>
> RunInstances를 사용하여 스팟 인스턴스 요청을 생성할 때 생성 시 스팟 인스턴스 요청을 태깅하지 않으려는 경우 `Resource` 목록에서 `spot-instances-request`를 생략할 수 있습니다. 생성 시 스팟 인스턴스 요청을 태깅하지 않으면 Amazon EC2가 RunInstances 문에서 `spot-instances-request` 리소스를 평가하지 않기 때문입니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
```

```
                "arn:aws:ec2:us-east-1:*:security-group/*",
                "arn:aws:ec2:us-east-1:*:key-pair/*",
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*",
                "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
            ]
        }
    ]
}
```

Warning

지원되지 않음 – 예: 사용자에게 RunInstances를 사용하여 스팟 인스턴스를 요청할 수 있는 권한 거부

`spot-instances-request` 리소스에는 다음 정책이 지원되지 않습니다.
다음 정책은 사용자에게 온디맨드 인스턴스를 시작할 수 있는 권한을 부여하지만 사용자에게 스팟 인스턴스를 요청할 수 있는 권한은 거부합니다. RunInstances에서 생성되는 `spot-instances-request` 리소스는 스팟 인스턴스를 요청하는 리소스입니다. 두 번째 문은 `spot-instances-request` 리소스에 대한 RunInstances 작업을 거부합니다. 그러나 생성 시 스팟 인스턴스 요청을 태깅하지 않으면 Amazon EC2가 RunInstances 문에서 `spot-instances-request` 리소스를 평가하지 않으므로 이 조건은 지원되지 않습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
                "arn:aws:ec2:us-east-1:*:security-group/*",
                "arn:aws:ec2:us-east-1:*:key-pair/*",
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*"
            ]
        },
```

```
        {
            "Sid": "DenySpotInstancesRequests - NOT SUPPORTED - DO NOT USE!",
            "Effect": "Deny",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
        }
    ]
}
```

예: 생성 시 스팟 인스턴스 요청 태깅

다음 정책은 사용자가 인스턴스 시작 중에 생성된 모든 리소스에 태그를 지정할 수 있도록 허용합니다. 첫 번째 문은 RunInstances에서 나열된 리소스를 생성할 수 있도록 허용합니다. RunInstances에서 생성되는 `spot-instances-request` 리소스는 스팟 인스턴스를 요청하는 리소스입니다. 두 번째 문은 `*` 와일드카드를 지정하여 인스턴스 시작 시 리소스가 생성될 때 모든 리소스에 태그를 지정하는 것을 허용합니다.

> **Note**
>
> 생성 시 스팟 인스턴스 요청을 태깅하면 Amazon EC2가 RunInstances 문에서 `spot-instances-request` 리소스를 평가합니다. 따라서 RunInstances 작업의 경우 `spot-instances-request` 리소스를 명시적으로 허용해야 합니다. 그러지 않으면 호출이 실패합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
                "arn:aws:ec2:us-east-1:*:security-group/*",
```

```
                "arn:aws:ec2:us-east-1:*:key-pair/*",
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*",
                "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
            ]
        },
        {
            "Sid": "TagResources",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "*"
        }
    ]
}
```

예: 스팟 인스턴스 요청에 대한 생성 시 태깅 거부

다음 정책은 사용자에게 인스턴스 시작 중에 생성된 리소스에 태그를 지정할 수 있는 권한을 거부합니다.

첫 번째 문은 RunInstances에서 나열된 리소스를 생성할 수 있도록 허용합니다. RunInstances에서 생성되는 `spot-instances-request` 리소스는 스팟 인스턴스를 요청하는 리소스입니다. 두 번째 문은 `*` 와일드카드를 제공하여 인스턴스 시작 시 리소스가 생성될 때 모든 리소스에 태그를 지정하는 것을 거부합니다. 생성 시 `spot-instances-request` 또는 다른 리소스에 태그를 지정하면 RunInstances 호출이 실패합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
                "arn:aws:ec2:us-east-1:*:security-group/*",
                "arn:aws:ec2:us-east-1:*:key-pair/*",
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*",
```

```
                "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
            ]
        },
        {
            "Sid": "DenyTagResources",
            "Effect": "Deny",
            "Action": "ec2:CreateTags",
            "Resource": "*"
        }
    ]
}
```

Warning

지원되지 않음 - 예: 특정 태그가 할당된 경우에만 스팟 인스턴스 요청을 생성하는 것을 허용 `spot-instances-request` 리소스에는 다음 정책이 지원되지 않습니다.
다음 정책은 요청이 특정 태그로 태깅된 경우에만 RunInstances에 스팟 인스턴스 요청을 생성할 수 있는 권한을 부여합니다.
첫 번째 문은 RunInstances에서 나열된 리소스를 생성할 수 있도록 허용합니다.
두 번째 문은 요청에 `environment=production` 태그가 있는 경우에만 사용자에게 스팟 인스턴스 요청을 생성할 수 있는 권한을 부여합니다. 이 조건이 RunInstances에서 생성된 다른 리소스에 적용되는 경우 태그를 지정하지 않으면 `Unauthenticated` 오류가 발생합니다. 그러나 스팟 인스턴스 요청에 태그가 지정되지 않은 경우 Amazon EC2는 RunInstances 문에서 `spot-instances-request` 리소스를 평가하지 않으므로 RunInstances에서 태깅되지 않은 스팟 인스턴스 요청이 생성됩니다.
사용자가 스팟 인스턴스 요청을 태깅하는 경우 Amazon EC2는 RunInstances 문에서 `spot-instances-request` 리소스를 평가하기 때문에 `environment=production` 이외의 다른 태그를 지정하면 `Unauthenticated` 오류가 발생합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
```

```
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
                "arn:aws:ec2:us-east-1:*:security-group/*",
                "arn:aws:ec2:us-east-1:*:key-pair/*",
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*"
            ]
        },
        {
            "Sid": "RequestSpotInstancesOnlyIfTagIs_environment=production - NOT
 SUPPORTED - DO NOT USE!",
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:us-east-1:*:spot-instances-request/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/environment": "production"
                }
            }
        },
        {
            "Sid": "TagResources",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "*"
        }

    ]
}
```

예: 특정 태그가 할당된 경우 스팟 인스턴스 요청을 생성하는 것을 거부

다음 정책은 요청이 environment=production으로 태깅된 경우 RunInstances에 스팟 인스턴스 요청을 생성할 수 있는 권한을 거부합니다.

첫 번째 문은 RunInstances에서 나열된 리소스를 생성할 수 있도록 허용합니다.

두 번째 문은 요청에 environment=production 태그가 있는 경우 사용자에게 스팟 인스턴스 요청을 생성할 수 있는 권한을 거부합니다. environment=production을 태그로 지정하면 Unauthenticated 오류가 발생합니다. 다른 태그를 지정하거나 태그를 지정하지 않으면 스팟 인스턴스 요청이 생성됩니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
                "arn:aws:ec2:us-east-1:*:security-group/*",
                "arn:aws:ec2:us-east-1:*:key-pair/*",
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*",
                "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
            ]
        },
        {
            "Sid": "DenySpotInstancesRequests",
            "Effect": "Deny",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:us-east-1:*:spot-instances-request/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/environment": "production"
                }
            }
        },
        {
            "Sid": "TagResources",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "*"
        }
    ]
}
```

예: 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": "*"
    }
  ]
}
```

예: 태그 리소스

다음 정책은 태그에 키 `CreateTags` 및 값 `environment`이 포함된 경우에만 사용자가 `production` 작업을 사용하여 인스턴스에 태그를 적용하는 것을 허용합니다. 다른 어떤 태그도 사용할 수 없으며 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-id:instance/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/environment": "production"
                }
            }
        }
    ]
}
```

다음 정책은 키가 `owner`이고 값이 username인 태그를 이미 가진 태그 지정 가능 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-id:*/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/anycompany:environment-type": ["test","prod"],
                    "aws: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-center인 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:account-id:volume/*",
      "Condition": {
        "ForAllValues:StringEquals": {
          "aws:TagKeys": ["environment","cost-center"]
        }
      }
    }
  ]
}
```

이 정책은 키가 owner이고 값이 username인 키로 리소스에 태그가 지정된 경우에 한해 어떤 리소스에서든 environment=prod 태그만 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-id:*/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/environment": "prod",
          "aws:ResourceTag/owner": "${aws:username}"
        },
        "ForAllValues:StringEquals": {
          "aws:TagKeys": ["environment"]
        }
      }
    }
  ]
}
```

예: 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.

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:us-east-1:account-id:instance/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/department":"test"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:DescribeIamInstanceProfileAssociations",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::account-id:role/DevTeam*"
    }
  ]
}
```

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 섹션을 참조하세요.

```
{
    "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-id:role/TestRole-*"
        }
    ]
}
```

예: 라우팅 테이블 작업

다음 정책은 VPC vpc-ec43eb89에만 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-id:route-table/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:Vpc": "arn:aws:ec2:region:account-id:vpc/vpc-ec43eb89"
                }
            }
        }
    ]
}
```

예: 특정 인스턴스에서 다른 AWS 서비스의 리소스를 보는 것을 허용

다음은 IAM 역할에 연결할 수 있는 정책의 예제입니다. 이 정책은 인스턴스가 다양한 AWS 서비스에서 리소스를 확인하도록 허용합니다. ec2:SourceInstanceARN 글로벌 조건 키를 사용하여 요청이 이

루어진 인스턴스가 i-093452212644b0dd6 인스턴스가 되도록 지정합니다. 동일한 IAM 역할이 다른 인스턴스와 연결된 경우에는 다른 인스턴스에서 이러한 작업을 수행할 수 없습니다.

```
{
    "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-id:instance/
i-093452212644b0dd6"
                }
            }
        }
    ]
}
```

예: 시작 템플릿 작업

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-id: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-id:launch-template/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Purpose": "Testing"
        }
      }
    }
  ]
}
```

인스턴스 메타데이터 작업

다음 정책은 사용자가 인스턴스 메타데이터 서비스 버전 2(IMDSv2)를 사용하여 인스턴스 메타데이터만 검색할 수 있도록 합니다. 다음 정책 4개를 결합하여 명령문 4개를 가진 정책 하나로 만들 수 있습니다. 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 및 인스턴스 시작(RunInstances)의 정책을 참조하세요.

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 `"autoscaling:LaunchConfigurationName"`로 지정된 `null` 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 `$latest`를 지정하여 `$default` 및 `"autoscaling:LaunchTemplateVersionSpecified": "true"`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 사용자 설명서의 Auto Scaling 그룹을 시작 템플릿으로 마이그레이션을 참조하세요.
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 update-auto-scaling-group을 참조하세요.

예시

- IMDSv2의 사용 요구
- IMDSv2 옵트아웃 거부
- 최대 홉 제한 지정
- 인스턴스 메타데이터 옵션을 수정할 수 있는 사용자 제한
- IMDSv2에서 역할 자격 증명을 검색하도록 요구

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"
                }
            }
        }
    ]
}
```

IMDSv2 옵트아웃 거부

다음 정책은 `ModifyInstanceMetadataOptions` API를 호출할 수 없으며 IMDSv1 또는 IMDSv2 옵션의 허용을 지정합니다. `ModifyInstanceMetadataOptions` API를 호출한다면 `HttpTokens` 속성은 `required`로 설정해야 합니다.

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [{
        "Sid": "DenyIMDSv1HttpTokensModification",
        "Effect": "Deny",
        "Action": "ec2:ModifyInstanceMetadataOptions",
        "Resource": "arn:aws:ec2:*:*:instance/*",
        "Condition": {
            "StringNotEquals": {
                "ec2:Attribute/HttpTokens": "required"
            },
            "Null": {
                "ec2:Attribute/HttpTokens": false
            }
        }
    }]
}
```

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 `Unauthorized0peration` 오류가 발생합니다.

```
{
    "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 `ec2-imds-admins` 역할을 가진 사용자만 인스턴스 메타데이터 옵션을 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 EBS 볼륨 및 스냅샷 작업

Amazon EBS 볼륨 및 스냅샷 작업에 관한 정책 예는 Identity-based policy examples for Amazon EBS를 참조하세요.

Amazon EC2 콘솔에 대한 액세스를 제어하는 정책 예제

IAM 정책을 사용하여 Amazon EC2를 사용하는 데 필요한 권한을 사용자에게 부여할 수 있습니다. 단계별 지침은 IAM 사용 설명서의 IAM 정책 생성을 참조하세요.

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. Amazon EC2 콘솔용 정책을 생성하는 방법에 대한 자세한 내용은 AWS 보안 블로그 게시물 Granting Users Permission to Work in the Amazon EC2 Console을 참조하세요.

다음 예제는 사용자에게 Amazon EC2 사용 권한을 부여하는 데 사용할 수 있는 정책 명령문을 보여줍니다. *`user input placeholder`*를 사용자의 정보로 바꿉니다. 이러한 정책은 AWS Management Console을 사용하는 요청에 맞게 설계되었습니다. Amazon EC2 콘솔은 단일 리소스를 표시하기 위해 여러 API 작업을 호출할 수 있으며, 사용자가 작업을 시도하고 콘솔에 오류가 표시되기 전까지는 명확하지 않을 수 있습니다. 자세한 내용은 AWS 보안 블로그 게시물 Granting Users Permission to Work in the Amazon EC2 Console을 참조하세요.

예시

- 예: 읽기 전용 액세스
- 예: EC2 시작 인스턴스 마법사 사용
- 예: 보안 그룹 작업
- 예: 탄력적 IP 주소 작업
- 예: 예약 인스턴스 작업

콘솔에서 작업을 수행하는 데 필요한 API 작업을 파악하려는 경우 AWS CloudTrail 등 호출을 로깅하는 서비스를 사용할 수 있습니다. 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.

예: 읽기 전용 액세스

사용자가 Amazon EC2 콘솔에서 모든 리소스를 조회하도록 허용하려면 다음 예제와 같은 정책을 사용합니다. 예: 읽기 전용 액세스. 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 Amazon EC2에 사용되는 작업, 리소스 및 조건 키를 참조하세요.

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 콘솔에서는 CloudWatch API를 사용하여 경보와 지표를 표시하므로 사용자에게 `cloudwatch:DescribeAlarms`, `cloudwatch:DescribeAlarmsForMetric`, `cloudwatch:ListMetrics`, `cloudwatch:GetMetricStatistics` 및 `cloudwatch:GetMetricData` 작업을 사용하는 권한을 부여해야 합니다.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "ec2:DescribeInstances",
         "ec2:DescribeInstanceTypes",
         "cloudwatch:DescribeAlarms",
         "cloudwatch:DescribeAlarmsForMetric",
         "cloudwatch:ListMetrics",
         "cloudwatch:GetMetricStatistics",
         "cloudwatch:GetMetricData"
      ],
      "Resource": "*"
   }
   ]
}
```

예: EC2 시작 인스턴스 마법사 사용

Amazon EC2 시작 인스턴스 마법사는 인스턴스를 구성하고 시작하는 옵션이 있는 화면입니다. 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 인스턴스 마법사 액세스

성공적으로 시작을 완료하려면 사용자에게 `ec2:RunInstances` API 작업 및 최소한 다음과 같은 API 작업 사용 권한을 부여해야 합니다.

- `ec2:DescribeImages`: AMI를 조회하고 선택합니다.
- `ec2:DescribeInstanceTypes`: 인스턴스 유형을 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:DescribeInstanceTypes",
                "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`: 에 의해 생성된 리소스에 태그 지정.`RunInstances` 자세한 내용은 생성 시 Amazon EC2 리소스 태그 지정에 대한 권한 부여 단원을 참조하십시오. 이 작업을 사용할 권한이 없는 사용자가 시작 인스턴스 마법사의 태그 지정 페이지에서 태그를 지정하려 시도하는 경우, 시작은 실패합니다.

  > **Important**
  >
  > 인스턴스를 시작하는 동안 Name(이름)을 지정하면 태그가 생성되고 `ec2:CreateTags` 작업이 필요합니다. 사용자에게 `ec2:CreateTags` 작업을 사용할 수 있는 권한을 부여하면 `aws:ResourceTag` 조건 키를 사용하여 다른 리소스의 사용을 제한하지 못할 수 있으므로 주의해야 합니다. 사용자가 `ec2:CreateTags` 작업을 사용할 수 있는 권한을 부여받으면 리소스의 태그를 변경하여 제한을 우회할 수 있습니다. 자세한 내용은 속성 기반 액세스를 사용하여 액세스 제어 단원을 참조하십시오.

- AMI을 선택하는 데 Systems Manager 파라미터를 사용하려면 정책에 `ssm:DescribeParameters` 및 `ssm:GetParameters`를 추가해야 합니다. `ssm:DescribeParameters`는 사용자에게 Systems Manager 파라미터를 보고 선택할 수 있는 권한을 부여하며, `ssm:GetParameters`는 사용자에게 Systems Manager 파라미터의 값을 가져올 수 있는 권한을 부여합니다. 특정 Systems Manager 파라미터에 대한 액세스를 제한할 수도 있습니다. 자세한 내용은 이 섹션의 뒷부분에 있는 특정 Systems Manager 파라미터에 대한 액세스 제한을 참조하세요.

현재 Amazon EC2 `Describe*` API 작업은 리소스별 권한을 지원하지 않으므로 사용자가 시작 인스턴스 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 및 리전에 대한 액세스 제한

다음 정책은 Amazon이 소유한 AMI를 사용하여 `t2.micro` 인스턴스를 시작하되 특정 서브넷(`subnet-1a2b3c4d`)으로만 시작하도록 허용합니다. 사용자는 지정된 리전에서만 시작할 수 있습니다. 사용자가 다른 리전을 선택하거나 시작 인스턴스 마법사에서 다른 인스턴스 유형, AMI 또는 서브넷을 선택하면 시작에 실패합니다.

첫 번째 명령문은 위 예제에 설명된 대로 사용자가 시작 인스턴스 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) 섹션을 참조하세요. 세 번째, 네 번째 명령문은 각각 인스턴스 및 AMI 리소스 사용 권한을 부여하지만 인스턴스가 t2.micro 인스턴스이고 AMI가 Amazon 또는 신뢰할 수 있고 검증된 특정 파트너의 소유인 경우로 한정합니다.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "ec2:DescribeInstances",
         "ec2:DescribeImages",
         "ec2:DescribeInstanceTypes",
         "ec2:DescribeKeyPairs",
         "ec2:CreateKeyPair",
         "ec2:DescribeVpcs",
         "ec2:DescribeSubnets", "ec2:DescribeSecurityGroups",
         "ec2:CreateSecurityGroup",
         "ec2:AuthorizeSecurityGroupIngress"
   ],
   "Resource": "*"
   },
   {
      "Effect": "Allow",
      "Action":"ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:region:111122223333:network-interface/*",
         "arn:aws:ec2:region:111122223333:volume/*",
         "arn:aws:ec2:region:111122223333:key-pair/*",
         "arn:aws:ec2:region:111122223333:security-group/*",
         "arn:aws:ec2:region:111122223333:subnet/subnet-1a2b3c4d"
      ]
   },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:region:111122223333:instance/*"
      ],
```

```
      "Condition": {
         "StringEquals": {
            "ec2:InstanceType": "t2.micro"
         }
      }
   },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
            "arn:aws:ec2:region::image/ami-*"
      ],
      "Condition": {
         "StringEquals": {
            "ec2:Owner": "amazon"
         }
      }
   }
   ]
}
```

특정 Systems Manager 파라미터에 대한 액세스 제한

다음 정책은 특정 이름의 Systems Manager 파라미터를 사용할 수 있는 액세스 권한을 부여합니다.

첫 번째 문은 시작 인스턴스 마법사에서 AMI를 선택할 때 Systems Manager 파라미터를 볼 수 있는 권한을 사용자에게 부여합니다. 두 번째 문은 사용자에게 `prod-*`라는 이름이 지정된 파라미터만 사용할 수 있는 권한을 부여합니다.

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

```
      "Resource": "arn:aws:ssm:region:123456123456:parameter/prod-*"
   }
   ]
}
```

예: 보안 그룹 작업

보안 그룹 조회와 규칙의 추가 및 삭제

다음 정책은 사용자가 Amazon EC2 콘솔에서 보안 그룹을 조회하고 인바운드 및 아웃바운드 규칙을 추가 및 제거하며 Department=Test 태그가 있는 기존 보안 그룹에 대한 규칙 설명을 나열하고 수정할 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:DescribeSecurityGroupRules",
         "ec2:DescribeTags"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
         "ec2:AuthorizeSecurityGroupIngress",
         "ec2:RevokeSecurityGroupIngress",
         "ec2:AuthorizeSecurityGroupEgress",
         "ec2:RevokeSecurityGroupEgress",
         "ec2:ModifySecurityGroupRules",
         "ec2:UpdateSecurityGroupRuleDescriptionsIngress",
         "ec2:UpdateSecurityGroupRuleDescriptionsEgress"
      ],
      "Resource": [
         "arn:aws:ec2:region:111122223333:security-group/*"
      ],
      "Condition": {
         "StringEquals": {
```

```
            "aws:ResourceTag/Department": "Test"
         }
      }
   },
   {
      "Effect": "Allow",
      "Action": [
         "ec2:ModifySecurityGroupRules"
      ],
      "Resource": [
         "arn:aws:ec2:region:111122223333:security-group-rule/*"
      ]
   }
]}
```

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 부

여하지 않으면 사용자가 잘못된 규칙이 포함된 보안 그룹을 생성하려고 할 때 규칙이 없는 보안 그룹이 생성되며, 사용자가 이후에 규칙을 추가해야 합니다.

- `ec2:UpdateSecurityGroupRuleDescriptionsIngress`: 수신(인바운드) 보안 그룹 규칙에 대한 설명을 추가하거나 업데이트합니다.
- `ec2:UpdateSecurityGroupRuleDescriptionsEgress`: 발신(아웃바운드) 보안 그룹 규칙에 대한 설명을 추가하거나 업데이트합니다.
- `ec2:ModifySecurityGroupRules`: 보안 그룹 규칙을 수정하려면
- `ec2:DescribeSecurityGroupRules`: 보안 그룹 규칙을 나열하려면

다음 정책은 보안 그룹 생성 대화 상자를 사용하고 특정 VPC(`vpc-1a2b3c4d`)에 연결된 보안 그룹에 인바운드 및 아웃바운드 규칙을 생성할 권한을 부여합니다. 사용자는 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"
         }
```

```
      }
    }
   ]
}
```

예: 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": "*"
        }
    ]
}
```

예: 예약 인스턴스 작업

다음 정책은 사용자가 계정의 예약 인스턴스를 보고 수정할 수 있을 뿐만 아니라 AWS Management Console에서 새 예약 인스턴스를 구매할 수 있도록 허용합니다.

이 정책을 통해 사용자는 계정의 모든 예약 인스턴스 및 온디맨드 인스턴스를 볼 수 있습니다. 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:DescribeInstanceTypes",
         "ec2:DescribeAvailabilityZones",
         "ec2:DescribeReservedInstancesOfferings"
      ],
      "Resource": "*"
   }
   ]
}
```

`ec2:DescribeAvailabilityZones` 작업은 Amazon EC2 콘솔이 예약 인스턴스를 구매할 수 있는 가용 영역에 대한 정보를 표시하도록 하는 데 필수적입니다. `ec2:DescribeInstances` 작업은 필수적이지는 않지만 사용자가 계정에서 인스턴스를 보고, 정확한 사양에 맞추기 위해 예약을 구매할 수 있도록 해줍니다.

API 작업을 조정해 사용자 액세스를 제한할 수 있습니다. 예를 들어 `ec2:DescribeInstances`와 `ec2:DescribeAvailabilityZones`를 제거하면 사용자가 읽기 전용 액세스 권한만을 갖게 됩니다.

Amazon EC2에 대한 AWS 관리형 정책

사용자, 그룹 또는 역할에 권한을 추가할 때 정책을 직접 작성하는 것보다 AWS관리형 정책을 사용하는 것이 더욱 편리합니다. 팀에 필요한 권한만 제공하는 IAM 고객 관리형 정책을 생성하려면 시간과 전문 지식이 필요합니다. 빨리 시작하려면 AWS관리형 정책을 사용할 수 있습니다. 이러한 정책은 일반적인 사용 사례에 적용되며 AWS계정에서 사용할 수 있습니다. AWS 관리형 정책에 대한 자세한 정보는 IAM 사용 설명서에서 AWS관리형 정책을 참조하세요.

AWS 서비스 유지 관리 및 AWS관리형 정책 업데이트입니다. AWS 관리형 정책에서는 권한을 변경할 수 없습니다. 서비스에서 때때로 추가 권한을 AWS 관리형 정책에 추가하여 새로운 기능을 지원합니다. 이 유형의 업데이트는 정책이 연결된 모든 ID(사용자, 그룹 및 역할)에 적용됩니다. 서비스는 새로운 기능이 시작되거나 새 태스크를 사용할 수 있을 때 AWS 관리형 정책에 업데이트됩니다. 서비스는 AWS관리형 정책에서 권한을 제거하지 않기 때문에 정책 업데이트로 인해 기존 권한이 손상되지 않습니다.

또한 AWS는 여러 서비스의 직무에 대한 관리형 정책을 지원합니다. 예를 들어 ReadOnlyAccess라는 이름의 AWS 관리형 정책은 모든 AWS 서비스 및 리소스에 대한 읽기 전용 액세스 권한을 제공합니다. 서비스에서 새 기능을 시작하면 AWS가 새 작업 및 리소스에 대한 읽기 전용 권한을 추가합니다. 직무 정책의 목록과 설명은 IAM 사용 설명서의 직무에 관한 AWS 관리형 정책을 참조하십시오.

AWS 관리형 정책: AmazonEC2FullAccess

`AmazonEC2FullAccess` 정책을 IAM 자격 증명에 연결할 수 있습니다. 이 정책은 Amazon EC2에 대한 전체 액세스를 허용하는 권한을 부여합니다.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 AmazonEC2FullAccess를 참조하세요.

AWS 관리형 정책: AmazonEC2ReadOnlyAccess

`AmazonEC2ReadOnlyAccess` 정책을 IAM 자격 증명에 연결할 수 있습니다. 이 정책은 Amazon EC2 대한 읽기 전용 액세스를 허용하는 권한을 부여합니다.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 AmazonEC2ReadOnlyAccess를 참조하세요.

AWS 관리형 정책: AWSEC2CapacityReservationFleetRolePolicy

이 정책은 AWSServiceRoleForEC2CapacityReservationFleet이라는 서비스 연결 역할에 연결되어 사용자를 대신해 서비스가 용량 예약 플릿에서 용량 예약을 생성, 수정 및 취소할 수 있도록 합니다. 자세한 내용은 용량 예약 플릿에 서비스 연결 역할 사용 섹션을 참조하세요.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 AWSEC2CapacityReservationFleetRolePolicy를 참조하세요.

AWS 관리형 정책: AWSEC2FleetServiceRolePolicy

이 정책은 AWSServiceRoleForEC2Fleet이라는 서비스 연결 역할에 연결되어 EC2 플릿이 사용자를 대신하여 인스턴스를 요청, 시작, 종료 및 태깅할 수 있도록 합니다. 자세한 내용은 EC2 집합의 서비스 연결 역할 섹션을 참조하세요.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 AWSEC2FleetServiceRolePolicy를 참조하세요.

AWS 관리형 정책: AWSEC2SpotFleetServiceRolePolicy

이 정책은 AWSServiceRoleForEC2SpotFleet이라는 서비스 연결 역할에 연결되어 스팟 플릿이 사용자를 대신하여 인스턴스를 시작하고 관리할 수 있도록 합니다. 자세한 내용은 스팟 플릿의 서비스 연결 역할 섹션을 참조하세요.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 AWSEC2SpotFleetServiceRolePolicy를 참조하세요.

AWS 관리형 정책: AWSEC2SpotServiceRolePolicy

이 정책은 AWSServiceRoleForEC2Spot이라는 서비스 연결 역할에 연결되어 Amazon EC2가 사용자를 대신하여 스팟 인스턴스를 시작하고 관리할 수 있도록 합니다. 자세한 내용은 스팟 인스턴스 요청에 대한 서비스 연결 역할 섹션을 참조하세요.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 AWSEC2SpotServiceRolePolicy를 참조하세요.

AWS 관리형 정책: AWSEC2VssSnapshotPolicy

Amazon EC2 Windows 인스턴스에서 사용할 IAM 인스턴스 프로파일 역할에 이 관리형 정책을 연결할 수 있습니다. 이 정책은 Amazon EC2가 자동으로 VSS 스냅샷을 생성하고 관리할 수 있는 권한을 부여합니다.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 AWSEC2VssSnapshotPolicy를 참조하세요.

AWS 관리형 정책: DeclarativePoliciesEC2Report

이 정책은 선언 정책에 대한 계정 상태 보고서를 생성하는 데 필요한 읽기 전용 API에 대한 액세스를 제공하기위해 `AWSServiceRoleForDeclarativePoliciesEC2Report`라는 서비스 연결 역할에 연결됩니다.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 DeclarativePoliciesEC2Report를 참조하세요.

AWS 관리형 정책: EC2FastLaunchFullAccess

인스턴스 프로필 또는 다른 IAM 역할에 `EC2FastLaunchFullAccess` 정책을 연결할 수 있습니다. 이 정책은 다음과 같이 EC2 Fast Launch 작업에 대한 전체 액세스 권한과 대상 권한을 부여합니다.

권한 세부 정보

- EC2 Fast Launch — 관리 액세스 권한이 부여되어 해당 역할이 EC2 Fast Launch를 활성화 또는 비활성화하고 EC2 Fast Launch 이미지를 설명할 수 있습니다.

- Amazon EC2 - Amazon EC2 RunInstances, CreateTags, Describe 작업과 시작 템플릿 생성 및 수정 작업에 대한 액세스 권한이 부여됩니다. 네트워크와 보안 그룹 리소스를 생성하고, 수신 규칙을 승인하고, EC2 Fast Launch가 생성한 리소스를 삭제할 수 있는 액세스 권한도 부여됩니다.
- IAM — 이름에 서비스 연결 역할을 생성할 수 있는 권한이 포함된 `ec2fastlaunch` 인스턴스 프로필을 가져오고 사용할 수 있는 액세스 권한이 부여됩니다. EC2FastLaunchServiceRolePolicy
- AWS CloudFormation - CloudFormation 스택을 설명 및 생성하고 생성한 스택을 삭제하기 위해 EC2 Fast Launch에 액세스할 수 있는 권한이 부여됩니다.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 EC2FastLaunchFullAccess를 참조하세요.

AWS 관리형 정책: EC2FastLaunchServiceRolePolicy

이 정책은 AWSServiceRoleForEC2FastLaunch라는 서비스 연결 역할에 연결되어 Amazon EC2가 EC2 Fast Launch가 활성화된 AMI에서 인스턴스를 시작하는 데 걸리는 시간을 줄이는 사전 프로비저닝된 스냅샷 세트를 생성하고 관리할 수 있도록 합니다. 자세한 내용은 the section called “서비스 연결 역할” 섹션을 참조하세요.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 EC2FastLaunchServiceRolePolicy를 참조하세요.

AWS 관리형 정책: Ec2InstanceConnectEndpoint

이 정책은 AWSServiceRoleForEC2InstanceConnect라는 서비스 연결 역할에 연결되어 EC2 Instance Connect 엔드포인트에서 자동으로 작업을 수행하도록 지원합니다. 자세한 내용은 EC2 Instance Connect 엔드포인트에 대한 서비스 연결 역할 섹션을 참조하세요.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 Ec2InstanceConnectEndpoint를 참조하세요.

AWS 관리형 정책에 대한 Amazon EC2 업데이트

이 서비스가 이러한 변경 내용을 추적하기 시작한 이후부터 Amazon EC2의 AWS 관리형 정책 업데이트에 대한 세부 정보를 봅니다.

변경 사항	설명	날짜
EC2FastLaunchFullAccess - 정책 업데이트	Amazon EC2는 시작 템플릿 생성 및 수정 작업을 포함하고, 네트워크 및 보안 그룹 리소스를 생성하며, 수신 규칙을 승인하고, EC2 Fast Launch가 생성	2025년 5월 14일

변경 사항	설명	날짜
	한 리소스를 삭제할 수 있도록 이 정책을 업데이트했습니다. CloudFormation 스택을 추가로 설명 및 생성하고 EC2 Fast Launch가 생성한 스택을 삭제할 수 있습니다.	
EC2FastLaunchServiceRolePolicy - 정책 업데이트	Amazon EC2는 이 정책을 업데이트하여 EC2 Fast Launch를 위한 이벤트 규칙을 생성하고 관리할 수 있도록 Amazon EventBridge 액세스 권한을 추가했습니다. 또한 EC2 Fast Launch는 이제 AWS CloudFormation 스택을 설명할 수 있고, AWS License Manager와 연결된 AMI에서 인스턴스를 시작할 수 있으며, 사용 중지할 수 있는 AWS KMS 권한 부여 목록을 가져올 수 있고, 생성한 시작 템플릿을 삭제할 수 있습니다.	2025년 5월 14일
AWSEC2CapacityReservationFleetRolePolicy - 업데이트된 권한	Amazon EC2는 `StringLike` 조건 연산자 대신 `ArnLike` 조건 연산자를 사용하도록 `AWSEC2CapacityReservationFleetRolePolicy` 관리형 정책을 업데이트했습니다.	2025년 3월 3일
AmazonEC2ReadOnlyAccess – 권한 추가	Amazon EC2는 `GetSecurityGroupsForVpc` 작업을 사용하여 보안 그룹을 검색할 수 있는 권한을 추가했습니다.	2024년 12월 27일

변경 사항	설명	날짜
EC2FastLaunchFullAccess - 새 정책	Amazon EC2는 인스턴스에서 EC2 Fast Launch 기능과 관련된 API 작업을 수행하기 위해 이 정책을 추가했습니다. 정책은 EC2 Fast Launch가 활성화된 AMI에서 시작된 인스턴스의 인스턴스 프로필에 연결할 수 있습니다.	2024년 5월 14일
AWSEC2VssSnapshotPolicy - 새 정책	Amazon EC2는 Amazon Machine Image(AMI) 및 EBS 스냅샷을 생성하고 태그를 추가할 수 있는 권한을 포함하는 AWSEC2VssSnapshotPolicy 정책을 추가했습니다.	2024년 3월 28일
EC2FastLaunchServiceRolePolicy - 새 정책	Amazon EC2는 사전 프로비저닝된 스냅샷 세트를 생성하여 Windows AMI가 인스턴스를 더 빠르게 시작할 수 있도록 EC2 Fast Launch 기능을 추가했습니다.	2021년 11월 26일
Amazon EC2 변경 사항 추적 시작	Amazon EC2 AWS관리형 정책에 대한 변경 사항 추적을 시작했습니다.	2021년 3월 1일

Amazon EC2의 IAM 역할

애플리케이션은 AWS 자격 증명으로 API 요청에 서명해야 합니다. 따라서 애플리케이션 개발자는 EC2 인스턴스에서 실행되는 인스턴스의 자격 증명을 관리할 전략을 수립해야 합니다. 예를 들어 AWS 자격 증명을 인스턴스에 안전하게 배포하여 다른 사용자로부터 보호하는 한편 해당 인스턴스의 애플리케이션이 자격 증명을 사용하여 요청에 서명하도록 할 수 있습니다. 그러나 각 인스턴스에 자격 증명을 안전하게 배포하기란 쉽지 않으며, 스팟 인스턴스와 같이 AWS에서 자동으로 생성하는 인스턴스

또는 Auto Scaling 그룹의 인스턴스에 대해서는 특히 어렵습니다. 또한 AWS 자격 증명을 교체할 때 각 인스턴스의 자격 증명을 업데이트할 수 있어야 합니다.

애플리케이션이 사용하는 보안 자격 증명을 직접 관리할 필요 없이 인스턴스의 애플리케이션에서 안전하게 API 요청을 전송할 수 있도록 IAM 역할을 설계했습니다. AWS 자격 증명을 생성하고 배포하는 대신 다음과 같이 IAM 역할을 사용하여 API 요청 전송 권한을 위임할 수 있습니다.

1. IAM 역할 생성.
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 사용자를 대상으로 정책을 생성할 때와 비슷합니다. 역할을 변경하면 모든 인스턴스에 변경 내용이 전파됩니다.

> **Note**
>
> Amazon EC2 IAM 역할 보안 인증에는 역할에 구성된 최대 세션 기간이 적용되지 않습니다. 자세한 내용은 IAM 사용 설명서의 역할 수임 방법을 참조하세요.

IAM 역할을 생성할 때, 애플리케이션에 필요한 특정 API 호출에 대한 액세스를 제한하는 최소 권한 IAM 정책을 연결합니다. Windows 간 통신의 경우 잘 정의되고 잘 문서화된 Windows 그룹 및 역할을 사용하여 Windows 인스턴스 간에 애플리케이션 수준의 액세스 권한을 부여합니다. 고객은 그룹 및 역할을 통해 최소 권한 애플리케이션 및 NTFS 폴더 수준 권한을 정의하여 애플리케이션별 요구 사항에 맞게 액세스를 제한할 수 있습니다.

인스턴스에 하나의 IAM 역할만 연결할 수 있지만, 여러 인스턴스에 동일한 역할을 연결할 수는 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 및 다음 예제: 예: IAM 역할 작업 섹션을 참조하세요.

목차

- 인스턴스 프로파일
- 사용 사례에 대한 권한
- 인스턴스 메타데이터에서 보안 자격 증명 검색
- 인스턴스에 IAM 역할을 연결할 수 있는 권한 부여
- IAM 역할을 인스턴스에 연결
- Amazon EC2 인스턴스의 인스턴스 자격 증명 역할

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. 여러 인스턴스 프로파일에 IAM 역할을 포함할 수 있습니다.

인스턴스에 대한 권한을 업데이트하려면 해당 인스턴스 프로파일을 교체합니다. 이 변경 사항이 적용되기까지 최대 1시간이 지연되므로 인스턴스 프로파일에서 역할을 제거하는 것은 권장되지 않습니다.

자세한 내용은 IAM 사용 설명서의 인스턴스 프로파일 사용을 참조하세요.

사용 사례에 대한 권한

애플리케이션에 대한 IAM 역할을 처음 생성하는 경우 간혹 필요 이상의 권한을 부여하게 될 수 있습니다. 프로덕션 환경에서 애플리케이션을 시작하기 전에 IAM 역할에 대한 액세스 활동을 기반으로 IAM 정책을 생성할 수 있습니다. IAM Access Analyzer는 사용자의 AWS CloudTrail 로그를 검토하고 지정된 날짜 범위에 역할에 의해 사용된 권한이 포함된 정책 템플릿을 생성합니다. 템플릿을 사용하여 세분화된 권한을 가진 관리형 정책을 생성한 다음 IAM 역할에 연결할 수 있습니다. 이렇게 하면 특정 사용 사례에 따라 역할이 AWS 리소스와 상호 작용하는 데 필요한 권한만 부여할 수 있습니다. 이는 최소 권한 부여 모범 사례를 준수하는 데 도움이 됩니다. 자세한 내용은 IAM 사용 설명서의 IAM Access Analyzer 정책 생성을 참조하세요.

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.

인스턴스 메타데이터에 대한 자세한 내용은 인스턴스 메타데이터를 사용하여 EC2 인스턴스를 관리합니다. 섹션을 참조하세요.

> **Warning**
>
> IAM 역할과 함께 인스턴스 메타데이터를 사용하는 서비스를 사용하는 경우 서비스에서 사용자 대신 HTTP 호출을 수행할 때 자격 증명이 노출되지 않도록 주의하세요. 자격 증명이 노출될 수 있는 서비스 유형은 HTTP 프록시, HTML/CSS 검증 서비스, XML 포함을 지원하는 XML 프로세서 등입니다.

Amazon EC2 워크로드의 경우 아래 설명된 방법을 사용하여 세션 자격 증명을 검색하는 것이 좋습니다. 이러한 자격 증명을 사용하면 `sts:AssumeRole`을 사용하여 인스턴스와 이미 연결된 동일한 역할을 맡을 필요 없이 워크로드에서 AWS API 요청을 수행할 수 있습니다. ABAC(속성 기반 액세스 제어)에 대한 세션 태그를 전달하거나 역할의 권한을 추가로 제한하기 위해 세션 정책을 전달해야 하는 경우가 아니면 이러한 역할 수임 호출은 동일한 임시 역할 세션 자격 증명의 새 세트를 생성하므로 불필요합니다.

워크로드가 역할을 사용하여 자체적으로 수임하는 경우 해당 역할이 자체적으로 수임하도록 명시적으로 허용하는 신뢰 정책을 생성해야 합니다. 신뢰 정책을 생성하지 않으면 `AccessDenied` 오류가 발생합니다. 자세한 내용은 IAM 사용 설명서의 역할 신뢰 정책 업데이트를 참조하세요.

IMDSv2

Linux

Linux 인스턴스에서 다음 명령을 실행하여 IAM 역할에 대한 보안 자격 증명을 검색합니다.

```
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" http://169.254.169.254/latest/meta-data/iam/security-credentials/role-name
```

Windows

Windows 인스턴스에서 다음 cmdlet을 실행하여 IAM 역할에 대한 보안 자격 증명을 검색합니다.

```
[string]$token = Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/iam/security-
credentials/role-name
```

IMDSv1

Linux

Linux 인스턴스에서 다음 명령을 실행하여 IAM 역할에 대한 보안 자격 증명을 검색합니다.

```
curl http://169.254.169.254/latest/meta-data/iam/security-credentials/role-name
```

Windows

Windows 인스턴스에서 다음 cmdlet을 실행하여 IAM 역할에 대한 보안 자격 증명을 검색합니다.

```
Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/iam/security-
credentials/role-name
```

출력의 예시는 다음과 같습니다. 보안 자격 증명을 검색할 수 없는 경우 IAM 사용 설명서의 제 EC2 인스턴스의 임시 보안 자격 증명에 액세스할 수 없습니다.를 참조하세요.

```
{
  "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 Tools for Windows PowerShell 명령의 경우, 임시 보안 자격 증명을 명시적으로 받지 않아도 됩니다. AWS SDK, AWS CLI 및 Tools for Windows PowerShell이 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 설명서에서 임시 보안 자격 증명을 사용하여 AWS 리소스에 대한 액세스 요청을 참조하세요.

인스턴스에 IAM 역할을 연결할 수 있는 권한 부여

IAM 사용자와 같은 AWS 계정의 ID에는 IAM 역할로 Amazon EC2 인스턴스를 시작하거나, 인스턴스에 IAM 역할을 연결하거나, 인스턴스의 IAM 역할을 바꾸거나, 인스턴스에서 IAM 역할을 분리할 수 있는 특정 권한이 있어야 합니다. 필요에 따라 다음 API 작업을 사용할 수 있는 권한을 부여해야 합니다.

- `iam:PassRole`
- `ec2:AssociateIamInstanceProfile`
- `ec2:DisassociateIamInstanceProfile`
- `ec2:ReplaceIamInstanceProfileAssociation`

> **Note**
>
> `iam:PassRole`에 대한 리소스를 *로 지정하면 인스턴스에 모든 IAM 역할을 전달할 수 있는 액세스 권한이 부여됩니다. 최소 권한의 모범 사례를 따르려면 아래 정책 예제에서와 같이 `iam:PassRole`로 특정 IAM 역할의 ARN을 지정합니다.

프로그래밍 방식 액세스를 위한 정책 예제

다음 IAM 정책은 AWS CLI 또는 Amazon EC2 API를 사용하여 IAM 역할로 인스턴스를 시작하거나, 인스턴스에 IAM 역할을 연결하거나, 인스턴스의 IAM 역할을 바꿀 수 있는 권한을 부여합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "ec2:RunInstances",
         "ec2:AssociateIamInstanceProfile",
         "ec2:DisassociateIamInstanceProfile",
```

```
            "ec2:ReplaceIamInstanceProfileAssociation"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": "iam:PassRole",
        "Resource": "arn:aws:iam::123456789012:role/DevTeam*"
    }
  ]
}
```

콘솔 액세스에 대한 추가 요구 사항

Amazon EC2 콘솔을 사용하여 동일한 태스크를 완료할 수 있는 권한을 부여하려면 `iam:ListInstanceProfiles` API 작업도 포함해야 합니다.

IAM 역할을 인스턴스에 연결

IAM 역할을 만들고 시작 도중 또는 후에 인스턴스에 연결할 수 있습니다. IAM 역할을 교체하거나 분리할 수도 있습니다.

Amazon EC2 콘솔을 사용하여 시작 시 인스턴스에 IAM 역할을 연결하려면 고급 세부 정보를 확장합니다. IAM 인스턴스 프로파일의 경우 IAM 역할을 선택합니다.

> **Note**
>
> IAM 콘솔을 사용하여 IAM 역할을 생성한 경우 인스턴스 프로파일이 생성되고 역할과 동일한 이름이 지정됩니다. AWS CLI, API 또는 AWS SDK를 사용하여 IAM 역할을 생성한 경우 인스턴스 프로파일에 역할과 다른 이름을 지정했을 수 있습니다.

실행 중이거나 중지된 인스턴스에 IAM 역할을 연결할 수 있습니다. 인스턴스에 이미 IAM 역할이 연결되어 있는 경우 새 IAM 역할로 교체해야 합니다.

Console

IAM 역할을 인스턴스에 연결하려면

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

3. 인스턴스를 선택합니다.
4. Actions(작업), Security(보안), Modify IAM role(IAM 역할 수정)을 선택합니다.
5. IAM 역할에 대해 IAM 인스턴스 프로파일을 선택합니다.
6. IAM 역할 업데이트를 선택합니다.

AWS CLI

IAM 역할을 인스턴스에 연결하려면

associate-iam-instance-profile 명령을 사용하여 인스턴스에 IAM 역할을 연결합니다. 인스턴스 프로파일을 지정할 때 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"
```

PowerShell

IAM 역할을 인스턴스에 연결하려면

Register-EC2IamInstanceProfile cmdlet을 사용합니다.

```
Register-EC2IamInstanceProfile `
    -InstanceId i-1234567890abcdef0 `
    -IamInstanceProfile_Name TestRole-1
```

이미 IAM 역할이 연결된 인스턴스에서 IAM 역할을 교체하려면 인스턴스가 실행 중이어야 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.

Console

인스턴스에서 IAM 역할을 대체하려면

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

3. 인스턴스를 선택합니다.
4. Actions(작업), Security(보안), Modify IAM role(IAM 역할 수정)을 선택합니다.
5. IAM 역할에 대해 IAM 인스턴스 프로파일을 선택합니다.
6. IAM 역할 업데이트를 선택합니다.

AWS CLI

인스턴스에서 IAM 역할을 대체하려면

1. 필요한 경우 describe-iam-instance-profile-associations 명령을 사용하여 연결 ID를 가져옵니다.

```
aws ec2 describe-iam-instance-profile-associations \
    --filters Name=instance-id,Values=i-1234567890abcdef0 \
    --query IamInstanceProfileAssociations.AssociationId
```

2. replace-iam-instance-profile-association 명령을 사용합니다. 기존 인스턴스 프로파일에 대한 연결 ID와 새 인스턴스 프로파일에 대한 ARN 또는 이름을 지정합니다.

```
aws ec2 replace-iam-instance-profile-association \
    --association-id iip-assoc-0044d817db6c0a4ba \
    --iam-instance-profile Name="TestRole-2"
```

PowerShell

인스턴스에서 IAM 역할을 대체하려면

1. 필요한 경우 Get-EC2IamInstanceProfileAssociation cmdlet을 사용하여 연결 ID를 가져옵니다.

```
(Get-EC2IamInstanceProfileAssociation -Filter @{Name="instance-id";
 Values="i-0636508011d8e966a"}).AssociationId
```

2. Set-EC2IamInstanceProfileAssociation cmdlet을 사용합니다. 기존 인스턴스 프로파일에 대한 연결 ID와 새 인스턴스 프로파일에 대한 ARN 또는 이름을 지정합니다.

```
Set-EC2IamInstanceProfileAssociation `
    -AssociationId iip-assoc-0044d817db6c0a4ba `
    -IamInstanceProfile_Name TestRole-2
```

실행 중이거나 중지된 인스턴스에서 IAM 역할을 분리할 수 있습니다.

Console

인스턴스에서 IAM 역할을 분리하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다.
4. Actions(작업), Security(보안), Modify IAM role(IAM 역할 수정)을 선택합니다.
5. IAM 역할(IAM role)에서 IAM 역할 없음(No IAM Role)을 선택합니다.
6. IAM 역할 업데이트를 선택합니다.
7. 확인 메시지가 나타나면 분리를 입력하고 분리를 선택합니다.

AWS CLI

인스턴스에서 IAM 역할을 분리하려면

1. 필요한 경우 describe-iam-instance-profile-associations를 사용하여 분리할 IAM 인스턴스 프로파일에 대한 연결 ID를 가져옵니다.

```
aws ec2 describe-iam-instance-profile-associations \
    --filters Name=instance-id,Values=i-1234567890abcdef0 \
    --query IamInstanceProfileAssociations.AssociationId
```

2. disassociate-iam-instance-profile 명령을 사용합니다.

```
aws ec2 disassociate-iam-instance-profile --association-id iip-
assoc-0044d817db6c0a4ba
```

PowerShell

인스턴스에서 IAM 역할을 분리하려면

1. 필요한 경우 Get-EC2IamInstanceProfileAssociation 을 사용하여 분리할 IAM 인스턴스 프로파일의 연결 ID를 가져옵니다.

```
(Get-EC2IamInstanceProfileAssociation -Filter @{Name="instance-id";
 Values="i-0636508011d8e966a"}).AssociationId
```

2. Unregister-EC2IamInstanceProfile cmdlet을 사용합니다.

```
Unregister-EC2IamInstanceProfile -AssociationId iip-assoc-0044d817db6c0a4ba
```

Amazon EC2 인스턴스의 인스턴스 자격 증명 역할

실행된 각 Amazon EC2 인스턴스에는 ID를 나타내는 인스턴스 ID 역할이 있습니다. 인스턴스 ID 역할은 일종의 IAM 역할입니다. 인스턴스 ID 역할을 사용하도록 통합된 AWS 서비스 및 기능은 이를 사용하여 서비스에서 인스턴스를 식별할 수 있습니다.

인스턴스 ID 역할 보안 인증 정보는 `/identity-credentials/ec2/security-credentials/ec2-instance`의 인스턴스 메타데이터 서비스(IMDS)에서 액세스할 수 있습니다. 보안 인증 정보는 AWS 임시 액세스 키 쌍과 세션 토큰으로 구성됩니다. 인스턴스 ID 역할을 사용하는 AWS 서비스에 대한 AWS Sigv4 요청에 서명하는 데 사용됩니다. 보안 인증 정보는 인스턴스 ID 역할을 사용하는 서비스 또는 기능이 인스턴스에서 활성화되었는지 여부에 관계없이 인스턴스 메타데이터에 표시됩니다.

인스턴스 ID 역할은 인스턴스가 시작될 때 자동으로 생성되며, 역할 신뢰 정책 문서가 없으며, ID 또는 리소스 정책의 적용을 받지 않습니다.

지원되는 서비스

다음 AWS 서비스는 인스턴스 ID 역할을 사용합니다.

- Amazon EC2 – EC2 Instance Connect는 인스턴스 ID 역할을 사용하여 Linux 인스턴스의 호스트 키를 업데이트합니다.
- Amazon GuardDuty - GuardDuty 런타임 모니터링은 인스턴스 ID 역할을 사용하여 런타임 에이전트가 GuardDuty VPC 엔드포인트로 보안 원격 측정을 전송할 수 있도록 합니다.
- AWS Security Token Service(AWS STS) – 인스턴스 ID 역할 보안 인증 정보를 AWS STS `GetCallerIdentity` 작업과 함께 사용할 수 있습니다.
- AWS Systems Manager— 기본 호스트 관리 구성을 사용하는 경우 AWS Systems Manager는 인스턴스 ID 역할에서 제공하는 ID를 사용하여 EC2 인스턴스를 등록합니다. 인스턴스를 식별한 후 Systems Manager는 사용자의 `AWSSystemsManagerDefaultEC2InstanceManagementRole` IAM 역할을 인스턴스에 전달할 수 있습니다.

인스턴스 ID 역할은 인스턴스 ID 역할과 통합되지 않으므로 다른 AWS 서비스 또는 기능과 함께 사용할 수 없습니다.

인스턴스 ID 역할 ARN

인스턴스 ID 역할 ARN은 다음과 같은 형식을 사용합니다.

```
arn:aws-partition:iam::account-number:assumed-role/aws:ec2-instance/instance-id
```

예시:

```
arn:aws:iam::0123456789012:assumed-role/aws:ec2-instance/i-1234567890abcdef0
```

ARN에 대한 자세한 내용은 IAM 사용 설명서의 Amazon 리소스 이름(ARN)을 참조하세요.

Amazon EC2 인스턴스에 대한 업데이트 관리

EC2 인스턴스에서 운영 체제와 애플리케이션을 정기적으로 패치, 업데이트 및 보호하는 것이 좋습니다. AWS Systems Manager Patch Manager를 사용하여 운영 체제와 애플리케이션 모두에 대해 보안 관련 업데이트를 설치하는 프로세스를 자동화할 수 있습니다.

Auto Scaling 그룹의 EC2 인스턴스의 경우 AWS-PatchAsgInstance 런북을 사용하여 패치가 진행 중인 인스턴스가 대체되는 것을 방지할 수 있습니다. 또는 자동 업데이트 서비스를 사용하거나, 애플리케이션 공급업체에서 제공하는 업데이트를 설치하기 위한 권장 프로세스를 사용할 수 있습니다.

리소스

- AL2023 - Amazon Linux 2023 사용 설명서의 AL2023 업데이트.
- AL2 - Amazon Linux 2 사용 설명서의 Amazon Linux 2 인스턴스의 소프트웨어 관리.
- Windows 인스턴스 – the section called “업데이트 관리”

Windows 인스턴스를 위한 보안 모범 사례

Windows 인스턴스에 대해 다음 보안 모범 사례를 따르는 것이 좋습니다.

내용

- 높은 수준의 보안 모범 사례
- 업데이트 관리

- 구성 관리
- 변경 관리
- Amazon EC2 Windows 인스턴스에 대한 감사 및 책임

높은 수준의 보안 모범 사례

Windows 인스턴스에 대해 다음과 같은 높은 수준의 보안 모범 사례를 준수해야 합니다.

- 최소 액세스 - 신뢰할 수 있고 예상되는 시스템 및 위치에만 액세스 권한을 부여합니다. 이는 Active Directory, Microsoft 비즈니스 생산성 서버 등 모든 Microsoft 제품 및 원격 데스크톱 서비스, 역방향 프록시 서버, IIS 웹 서버 등의 인프라 서비스에 적용됩니다. Amazon EC2 인스턴스 보안 그룹, 네트워크 액세스 제어 목록(ACL), Amazon VPC 퍼블릭/프라이빗 서브넷과 같은 AWS 기능을 사용하여 아키텍처의 여러 위치에 걸쳐 보안을 계층화합니다. 고객은 Windows 인스턴스 안에서 Windows 방화벽을 사용하여 해당 배포의 심층 방어 전략을 추가로 계층화할 수 있습니다. 시스템이 설계된 대로 작동하는 데 필요한 OS 구성 요소 및 애플리케이션만 설치하세요. IIS와 같은 인프라 서비스를 서비스 계정에서 실행하도록 구성하거나, 애플리케이션 풀 ID와 같은 기능을 사용하여 인프라 전체의 리소스에 로컬 또는 원격으로 액세스하도록 구성합니다.
- 최소 권한 - 인스턴스 및 계정이 담당하는 기능을 수행하기 위해 필요한 최소 권한 집합을 결정합니다. 이렇게 정의된 권한만 허용하도록 이러한 서버 및 사용자를 제한합니다. 역할 기반 액세스 제어와 같은 기술을 사용하여 관리 계정의 노출 영역을 줄이고, 작업 수행을 위해 가장 제한된 역할을 만듭니다. NTFS의 EFS(파일 시스템 암호화)와 같은 OS 기능을 사용하여 저장된 민감한 데이터를 암호화하고, 그 데이터에 대한 애플리케이션 및 사용자 액세스를 제어합니다.
- 구성 관리 - 바이러스 백신, 맬웨어 방지, 침입 탐지/방지 및 파일 무결성 모니터링이 포함된 호스트 기반 보호 제품군과 최신 보안 패치를 통합하는 기본 서버 구성을 생성합니다. 현재 기록된 기준선으로 각 서버를 평가하여 편차를 식별하고 플래그를 지정합니다. 각 서버가 적절한 로그 및 감사 데이터를 생성하고 안전하게 저장하도록 구성되어 있는지 확인합니다.
- 변경 관리 - 변경 프로세스의 완전 자동화를 목표로 서버 구성 기준의 변경 사항을 제어하는 프로세스를 생성합니다. 또한 Windows PowerShell DSC와 함께 JEA(충분한 관리)를 활용하여 관리 액세스를 최소한의 필수 기능만으로 제한합니다.
- 패치 관리 - EC2 인스턴스에서 운영 체제와 애플리케이션을 정기적으로 패치, 업데이트 및 보호하는 프로세스를 구현합니다.
- 감사 로그 - Amazon EC2 인스턴스에 대한 액세스 및 모든 변경 사항을 감사하여 서버 무결성을 확인하고 승인된 변경 사항만 수행되도록 합니다 IIS용 향상된 로깅 등의 기능을 활용하여 기본 로깅 기능을 향상시킵니다. VPC 흐름 로그 등 AWS 기능을 사용할 수 있으며, 허용/거부된 요청 및 API 호출 등 네트워크 액세스를 AWS CloudTrail로 각각 감사할 수도 있습니다.

업데이트 관리

Amazon EC2에서 Windows Server를 실행할 때 최상의 결과를 얻으려면 다음 모범 사례를 구현하는 것이 좋습니다.

- Configure Windows Update
- Update drivers
- Use the latest Windows AMIs
- Test performance before migration
- Update launch agents
- 업데이트를 설치한 후 Windows 인스턴스를 재부팅합니다. 자세한 내용은 Amazon EC2 인스턴스 재부팅 단원을 참조하십시오.

Windows 인스턴스를 새 버전의 Windows Server로 업그레이드하거나 마이그레이션하는 방법에 대한 자세한 내용은 EC2 Windows 인스턴스를 새 버전의 Windows Server로 업그레이드 단원을 참조하세요.

Windows 업데이트 구성

기본적으로 AWS Windows Server AMI에서 실행되는 인스턴스는 Windows 업데이트를 통해 업데이트를 받지 않습니다.

Windows 드라이버 업데이트

플릿 전체에 최신 문제 수정 및 성능 개선사항이 적용되도록 하려면 모든 Windows EC2 인스턴스에 최신 드라이버를 설치해야 합니다. 인스턴스 유형에 따라 AWS PV, Amazon ENA, AWS NVMe 드라이버를 업데이트해야 합니다.

- SNS 주제를 이용해 새로 출시된 드라이버에 관한 업데이트를 받으세요.
- AWS Systems Manager Automation 실행서 AWSSupport-UpgradeWindowsAWSDrivers를 사용하여 인스턴스 전반에 업데이트를 손쉽게 적용하세요.

최신 Windows AMI를 사용하여 인스턴스 시작

AWS는 최신 OS 패치, 드라이버, 시작 에이전트가 포함된 신규 Windows AMI를 매월 출시합니다. 새 인스턴스를 시작하거나 자체 사용자 지정 이미지를 만들 때는 최신 AMI를 활용해야 합니다.

- AWS Windows AMI의 각 릴리스에 대한 업데이트를 보려면 AWS Windows AMI 버전 기록 섹션을 참조하세요.
- 사용 가능한 최신 AMI를 이용해 이미지를 만드는 방법은 Systems Manager 파라미터 스토어를 사용한 최신 Windows AMI에 대한 쿼리 섹션을 참조하세요.
- 데이터베이스 및 규정 준수 강화 사용 사례의 인스턴스를 시작하는 데 사용할 수 있는 특수 Windows AMI에 대한 자세한 내용은 AWS Windows AMI 참조에서 특수 Windows AMI를 참조하세요.

마이그레이션하기 전 시스템/애플리케이션 성능 테스트

엔터프라이즈 애플리케이션을 AWS로 마이그레이션하는 작업은 다양한 변수와 구성을 동반할 수 있습니다. 다음을 보장하라면 EC2 솔루션의 성능을 테스트해야 합니다.

- 인스턴스 크기, 향상된 네트워킹, 테넌시(공유 또는 전용)를 포함한 인스턴스 유형이 올바르게 구성됩니다.
- 인스턴스 토폴로지는 워크로드에 적합하며, 필요 시 전용 테넌시, 배치 그룹, 인스턴스 스토어 볼륨 및 베어 메탈과 같은 고성능 기능을 활용합니다.

시작 에이전트 업데이트

플릿 전체에 최신 개선 사항이 적용되도록 EC2Launch v2 에이전트를 최신 버전으로 업데이트하세요. 자세한 내용은 the section called “EC2Launch v2로 마이그레이션” 단원을 참조하십시오.

혼합 플릿이 있거나 EC2Launch(Windows Server 2016 및 2019) 또는 EC2 Config(레거시 OS 전용) 에이전트를 계속 사용하고자 하는 경우 해당 에이전트를 최신 버전으로 업데이트하세요.

자동 업데이트는 다음 Windows Server 버전 및 시작 에이전트 조합에서 지원됩니다. Amazon EC2 시작 에이전트의 SSM Quick Setup 호스트 관리 콘솔에서 자동 업데이트를 옵트인할 수 있습니다.

Windows 버전	EC2Launch v1	EC2Launch v2
2016	✓	✓
2019	✓	✓
2022		✓

- EC2Launch v2로 업데이트하는 방법에 대한 자세한 내용은 the section called “EC2Launch v2 설치” 섹션을 참조하세요.
- EC2Config를 수동으로 업데이트하는 방법에 대한 자세한 내용은 the section called “EC2Config 설치” 섹션을 참조하세요.
- EC2Launch를 수동으로 업데이트하는 방법에 대한 자세한 내용은 the section called “EC2Launch 설치” 섹션을 참조하세요.

구성 관리

Amazon Machine Image(AMI)는 Amazon EC2 인스턴스에 대한 초기 구성을 제공합니다. 여기에는 Windows OS와 애플리케이션 및 보안 제어 같은 고객별 사용자 지정 옵션 등이 포함됩니다. 사용자 지정된 보안 구성 기준이 들어 있는 AMI 카탈로그를 생성하여 모든 Windows 인스턴스가 표준 보안 제어로 시작되도록 합니다. 보안 기준을 AMI에 베이킹하거나, EC2 인스턴스가 시작될 때 동적으로 부트스트래핑하거나, 하나의 제품으로 패키징하여 AWS Service Catalog 포트폴리오를 통해 균일하게 배포할 수 있습니다. AMI 보안 유지에 대한 자세한 내용은 AMI 작성 모범 사례를 참조하십시오.

각 Amazon EC2 인스턴스는 조직의 보안 표준을 준수해야 합니다. 필요 없는 Windows 역할 및 기능을 설치하지 말고, 악성 코드로부터 보호하고(바이러스 백신, 맬웨어 방지, 악용 완화) 호스트 무결성을 모니터링하고 침입 탐지를 수행하는 소프트웨어를 설치하세요. OS 보안 설정을 모니터링 및 유지 관리하고, 중요한 OS 파일의 무결성을 보호하며, 보안 기준에서 벗어나면 경보를 보내도록 보안 소프트웨어를 구성합니다. Microsoft, CIS(인터넷 보안 센터) 또는 NIST(미국 국립 표준 기술 연구소)에서 게시한 권장 보안 구성 벤치마크를 구현하는 것이 좋습니다. SQL Server용 모범 사례 분석기 등 특정 애플리케이션 서버를 위한 다른 Microsoft 도구를 사용하는 방법을 고려하십시오.

AWS 고객은 Amazon Inspector 평가를 실행하여 Amazon EC2 인스턴스에 배포된 애플리케이션의 보안 및 규정 준수를 개선할 수도 있습니다. Amazon Inspector는 애플리케이션의 취약성 또는 모범 사례와의 차이를 자동으로 평가하며, 공통의 보안 규정 준수 표준(예: PCI DSS)과 취약성 정의에 매핑된 수백 가지 규칙으로 구성된 기술 자료를 포함합니다. 원격 루트 로그인이 활성화되어 있는지 또는 취약한 소프트웨어 버전이 설치되어 있는지 확인하는 것이 대표적인 기본 제공 규칙입니다. 이러한 규칙은 AWS 보안 연구원이 정기적으로 업데이트합니다.

Windows 인스턴스를 보호할 때는 Active Directory 도메인 서비스를 구현하여 확장 가능하고 안전하며 관리하기 쉬운 인프라를 분산 위치에서 사용하는 것이 좋습니다. 또한 Amazon EC2 콘솔에서 인스턴스를 시작하거나 AWS CloudFormation 등의 Amazon EC2 프로비저닝 도구를 사용한 후 구성 드리프트가 발생하는 경우 Microsoft Windows PowerShell DSC와 같은 기본 OS 기능을 사용하여 구성 상태를 유지 관리하는 것이 좋습니다.

변경 관리

시작 시 Amazon EC2 인스턴스에 초기 보안 기준이 되고, 그 이후에는 진행 중인 Amazon EC2 변경 사항을 제어하여 가상 시스템의 보안을 유지합니다. AWS 리소스(예: 보안 그룹, 라우팅 테이블, 네트워크 ACL)뿐만 아니라 OS 및 애플리케이션 구성(예: Windows 또는 애플리케이션 패치 적용, 소프트웨어 업그레이드 또는 구성 파일 업데이트)에 대한 변경 사항을 승인하고 통합하는 변경 관리 프로세스를 설정합니다.

AWS는 AWS CloudTrail, AWS Config, AWS CloudFormation, AWS Elastic Beanstalk 등의 AWS 리소스 변경 사항 관리를 위한 여러 가지 도구와 시스템 센터 운영 관리자 및 시스템 센터 가상 컴퓨터 관리자를 위한 관리 팩을 제공합니다. Microsoft는 매달 두 번째 화요일에(또는 필요에 따라) Windows 패치를 릴리스하고 AWS는 Microsoft가 패치를 릴리스한 후 5일 안에 AWS가 관리하는 모든 Windows AMI를 업데이트합니다. 따라서 모든 기본 AMI를 지속적으로 패치하고, AWS CloudFormation 템플릿과 Auto Scaling 그룹 구성을 최신 AMI ID로 업데이트하고, 실행 중인 인스턴스 패치 관리를 자동화하는 도구를 구현해야 합니다.

Microsoft는 Windows OS 및 애플리케이션 변경 사항을 관리하기 위한 몇 가지 옵션을 제공합니다. 예를 들어, SCCM은 수명 주기 범위 전체에 걸쳐 환경을 수정할 수 있습니다. 비즈니스 요구 사항을 해결하고 변경 사항이 애플리케이션 SLA, 용량, 보안 및 재해 복구 절차에 미치는 영향을 제어하는 도구를 선택하세요. 수동 변경을 방지하고, 그 대신 자동화된 구성 관리 소프트웨어나 EC2 Run Command 또는 Windows PowerShell과 같은 명령줄 도구를 활용하여 반복 가능한 스크립트형 변경 프로세스를 구현하세요. 이러한 요구 사항을 지원하려면 Windows 인스턴스와의 모든 상호 작용에 향상된 로그 기능의 Bastion Host를 사용하여 모든 이벤트와 작업이 자동으로 기록되도록 하세요.

Amazon EC2 Windows 인스턴스에 대한 감사 및 책임

AWS CloudTrail, AWS Config, AWS Config 규칙은 AWS 리소스 변경 감사를 위한 감사 및 변경 추적 기능을 제공합니다. 로컬 로그 파일을 중앙 로그 관리 시스템으로 전송하도록 Windows 이벤트 로그를 구성하여 보안 및 운영 동작 분석을 위해 로그 데이터를 보존합니다. Microsoft SCOM(시스템 센터 운영 관리자)은 Windows 인스턴스에 배포된 Microsoft 애플리케이션에 대한 정보를 집계하고 애플리케이션 역할 및 서비스를 기반으로 사전 구성된 규칙 및 사용자 지정 규칙 세트를 적용합니다. 시스템 센터 관리 팩은 SCOM을 기반으로 애플리케이션별 모니터링 및 구성 지침을 제공합니다. 이러한 관리 팩은 Windows Server Active Directory, SharePoint Server 2013, Exchange Server 2013, Lync Server 2013, SQL Server 2014 등 다양한 서버와 기술을 지원합니다.

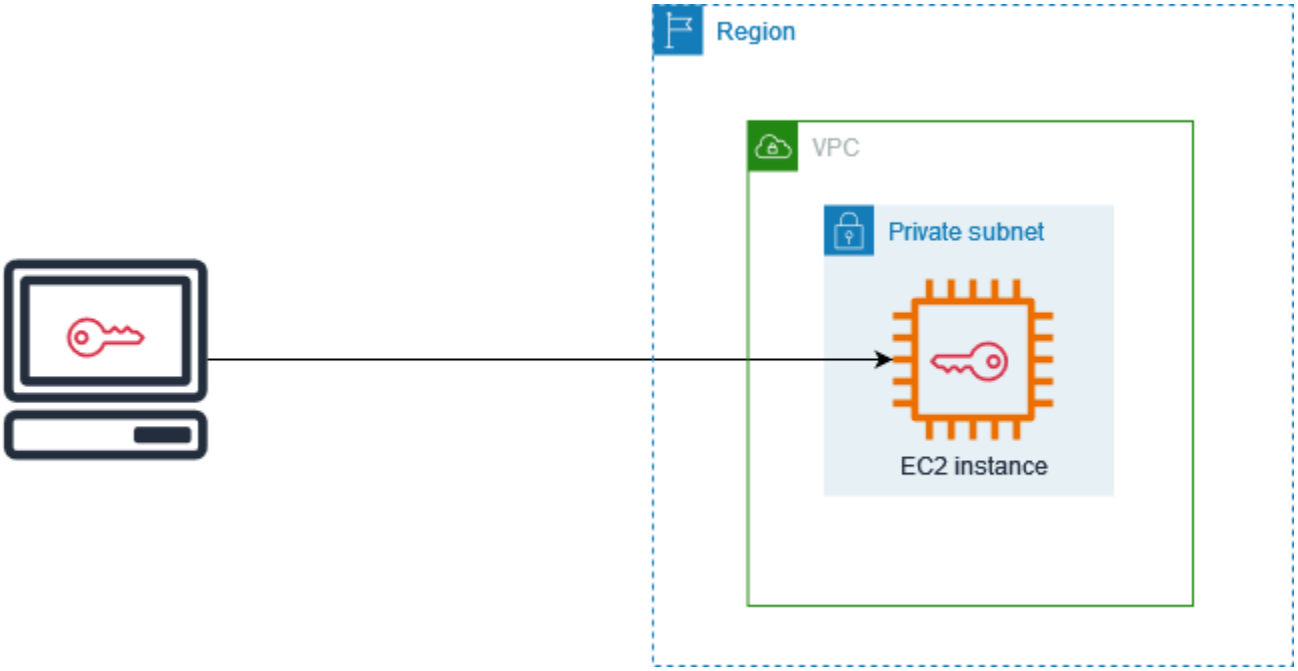
고객은 Microsoft 시스템 관리 도구 외에도 Amazon CloudWatch를 사용하여 인스턴스 CPU 사용률, 디스크 성능, 네트워크 I/O를 모니터링하고 호스트 및 인스턴스 상태 확인을 수행할 수 있습니다. EC2Config, EC2Launch, EC2Launch v2 시작 에이전트를 통해 Windows 인스턴스에 대한 고급 기능을

추가로 이용할 수 있습니다. 예를 들어 Windows 시스템, 보안, 애플리케이션 및 IIS(인터넷 정보 서비스) 로그를 CloudWatch Logs로 내보내고 이를 Amazon CloudWatch 메트릭 및 경보와 통합할 수 있습니다. 고객은 Windows 성능 카운터를 Amazon CloudWatch 사용자 지정 메트릭으로 내보내는 스크립트를 만들 수도 있습니다.

Amazon EC2 키 페어 및 Amazon EC2 인스턴스

퍼블릭 키와 프라이빗 키로 구성되는 키 페어는 Amazon EC2 인스턴스에 연결할 때 자격 증명 입증에 사용하는 보안 자격 증명 집합입니다. Linux 인스턴스의 경우 프라이빗 키를 사용하여 인스턴스에 SSH로 안전하게 연결할 수 있습니다. Windows 인스턴스의 경우 인스턴스에 연결할 때 사용하는 관리자 암호를 복호화하려면 프라이빗 키가 필요합니다.

Amazon EC2는 퍼블릭 키를 인스턴스에 저장하며, 다음 다이어그램과 같이 프라이빗 키는 사용자가 저장합니다. 프라이빗 키를 소유하는 사람은 누구나 키 페어를 사용하는 인스턴스에 연결할 수 있으므로 보안 위치에 프라이빗 키를 저장해야 합니다.



인스턴스를 시작할 때 키 페어를 지정하여 키 페어가 필요한 메서드를 사용하여 인스턴스에 연결할 수 있습니다. 보안 관리 방식에 따라 모든 인스턴스에 대해 동일한 키 페어를 지정하거나 다른 키 페어를 지정할 수 있습니다.

Linux 인스턴스의 경우 인스턴스가 처음 부팅될 때 시작 시 지정한 퍼블릭 키가 Linux 인스턴스에 `~/.ssh/authorized_keys` 내의 항목에 배치됩니다. SSH를 사용하여 Linux 인스턴스에 연결할 때 로그인하려면 퍼블릭 키에 해당하는 프라이빗 키를 지정해야 합니다.

EC2 인스턴스 연결에 대한 자세한 내용은 EC2 인스턴스에 연결 섹션을 참조하세요.

> **Important**
>
> Amazon EC2에는 프라이빗 키의 사본이 보관되지 않으므로, 프라이빗 키를 분실하면 이를 복구할 방법이 전혀 없습니다. 그러나 분실된 프라이빗 키를 사용하는 인스턴스에 연결하는 방법은 여전히 있을 수 있습니다. 자세한 내용은 프라이빗 키를 분실했습니다. 인스턴스에 연결하려면 어떻게 해야 하나요? 섹션을 참조하세요.

키 페어의 대안으로 AWS Systems Manager Session Manager를 사용하여 대화형 원클릭 브라우저 기반 셸 또는 AWS Command Line Interface(AWS CLI)로 인스턴스에 연결할 수 있습니다.

내용

- Amazon EC2 인스턴스에 대한 키 페어 생성
- 키 페어 설명
- 키 페어 삭제
- Linux 인스턴스에 대한 퍼블릭 키 추가 또는 바꾸기
- 키 페어의 지문 확인

Amazon EC2 인스턴스에 대한 키 페어 생성

Amazon EC2를 사용하여 키 페어를 생성하거나 서드 파티 도구를 사용하여 키 페어를 생성한 후에 Amazon EC2로 가져올 수 있습니다.

Amazon EC2는 Linux 및 Windows 인스턴스에 대해 2048비트 SSH-2 RSA 키를 지원합니다. Amazon EC2는 Linux 인스턴스에 대한 ED25519 키도 지원합니다.

키 페어를 생성한 후 인스턴스에 연결하는 방법에 대한 지침은 the section called “SSH를 사용하여 Linux 인스턴스에 연결” 및 the section called “RDP를 사용하여 Windows 인스턴스에 연결” 섹션을 참조하세요.

내용

- Amazon EC2를 사용하여 키 페어 생성
- AWS CloudFormation을 사용하여 키 페어 생성
- 서드 파티 도구를 사용하여 키 페어를 생성하고 Amazon EC2로 퍼블릭 키 가져오기

Amazon EC2를 사용하여 키 페어 생성

Amazon EC2 사용하여 키 페어를 생성할 때 퍼블릭 키는 Amazon EC2에 저장되며 프라이빗 키는 사용자가 저장합니다.

리전당 최대 5,000개의 키 페어를 생성할 수 있습니다. 증가를 요청하려면 지원 케이스를 생성합니다. 자세한 내용을 알아보려면 지원 사용 설명서의 지원 사례 만들기를 참조하세요.

Console

Amazon EC2를 사용하여 키 페어를 생성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창의 [Network & Security]에서 [Key Pairs]를 선택합니다.
3. Create key pair(키 페어 생성)를 선택합니다.
4. 이름에 키 페어를 설명하는 이름을 입력합니다. Amazon EC2는 사용자가 키 이름으로 지정한 이름에 퍼블릭 키를 연결합니다. 키 이름에는 최대 255자의 ASCII 문자를 포함할 수 있습니다. 선행 또는 후행 공백을 포함할 수 없습니다.
5. 운영 체제에 적합한 키 페어 유형을 선택합니다.

   (Linux 인스턴스)키 페어 유형에서 RSA 또는 ED25519를 선택합니다.

   (Windows 인스턴스) 키 페어 유형에서 RSA를 선택합니다. ED25519 키는 Windows 인스턴스에서 지원되지 않습니다.
6. 프라이빗 키 파일 형식(Private key file format)에서 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.
7. 퍼블릭 키에 태그를 추가하려면 태그 추가(Add tag)를 선택하고 해당 태그의 키와 값을 입력합니다. 각 태그에 대해 반복합니다.
8. 키 페어 생성(Create key pair)를 선택합니다.
9. 브라우저에서 프라이빗 키 파일이 자동으로 다운로드됩니다. 기본 파일 이름은 키 페어의 이름으로 지정한 이름이며, 파일 이름 확장자는 선택한 파일 형식에 따라 결정됩니다. 안전한 장소에 프라이빗 키 파일을 저장합니다.

> **Important**
>
> 이때가 사용자가 프라이빗 키 파일을 저장할 수 있는 유일한 기회입니다.

10. macOS 또는 Linux 컴퓨터에서 SSH 클라이언트를 사용하여 Linux 인스턴스에 연결할 계획이면 사용자만 프라이빗 키 파일을 읽을 수 있도록 다음 명령으로 해당 권한을 설정합니다.

```
chmod 400 key-pair-name.pem
```

이러한 권한을 설정하지 않으면 이 키 페어를 사용하여 인스턴스에 연결할 수 없습니다. 자세한 내용은 오류: 보호되지 않는 프라이빗 키 파일 섹션을 참조하세요.

AWS CLI

Amazon EC2를 사용하여 키 페어를 생성하려면

1. 다음과 같이 create-key-pair 명령을 사용하여 키 페어를 생성하고 프라이빗 키를 .pem 파일에 저장합니다. --query 옵션은 프라이빗 키 구성 요소를 출력에 인쇄합니다. --output 옵션은 지정된 파일에 프라이빗 키 구성 요소를 저장합니다. 확장명은 키 형식에 따라 .pem 또는 .ppk여야 합니다. 프라이빗 키 이름은 퍼블릭 키 이름과 다를 수 있지만, 사용상 편의를 위해 동일한 이름을 사용합니다.

```
aws ec2 create-key-pair \
    --key-name my-key-pair \
    --key-type rsa \
    --key-format pem \
    --query "KeyMaterial" \
    --output text > my-key-pair.pem
```

2. macOS 또는 Linux 컴퓨터에서 SSH 클라이언트를 사용하여 Linux 인스턴스에 연결할 계획이면 사용자만 프라이빗 키 파일을 읽을 수 있도록 다음 명령으로 해당 권한을 설정합니다.

```
chmod 400 key-pair-name.pem
```

이러한 권한을 설정하지 않으면 이 키 페어를 사용하여 인스턴스에 연결할 수 없습니다. 자세한 내용은 오류: 보호되지 않는 프라이빗 키 파일 섹션을 참조하세요.

PowerShell

Amazon EC2를 사용하여 키 페어를 생성하려면

다음과 같이 New-EC2KeyPair cmdlet을 사용하여 키를 생성하고 생성한 키를 .pem 또는 .ppk 파일에 저장합니다. Out-File cmdlet은 특정 확장자가 있는 파일에 프라이빗 키 구성 요소를 저장합니

다. 확장명은 키 형식에 따라 .pem 또는 .ppk여야 합니다. 프라이빗 키 이름은 퍼블릭 키 이름과 다를 수 있지만, 사용상 편의를 위해 동일한 이름을 사용합니다.

```
(New-EC2KeyPair `
    -KeyName "my-key-pair" `
    -KeyType "rsa" `
    -KeyFormat "pem").KeyMaterial | Out-File -Encoding ascii -FilePath C:\path\my-key-pair.pem
```

AWS CloudFormation을 사용하여 키 페어 생성

AWS CloudFormation을 사용하여 새 키 페어를 생성하면 프라이빗 키가 AWS Systems Manager 파라미터 스토어에 저장됩니다. 다음은 파라미터 이름의 형식입니다.

```
/ec2/keypair/key_pair_id
```

자세한 내용을 알아보려면 AWS Systems Manager 사용 설명서의 AWS Systems Manager Parameter Store를 참조하세요.

AWS CloudFormation을 사용하여 키 페어를 생성하려면

1. 템플릿에서 AWS::EC2::KeyPair 리소스를 사용합니다.

   ```
   Resources:
     NewKeyPair:
       Type: 'AWS::EC2::KeyPair'
       Properties:
         KeyName: new-key-pair
   ```

2. 다음과 같이 describe-key-pairs 명령을 사용하여 키 페어의 ID를 가져옵니다.

   ```
   aws ec2 describe-key-pairs --filters Name=key-name,Values=new-key-pair --query
    KeyPairs[*].KeyPairId --output text
   ```

   출력의 예시는 다음과 같습니다.

   ```
   key-05abb699beEXAMPLE
   ```

3. 다음과 같이 get-parameter 명령을 사용하여 키의 파라미터를 가져오고 키 구성 요소를 .pem 파일에 저장합니다.

```
aws ssm get-parameter --name /ec2/keypair/key-05abb699beEXAMPLE --with-decryption
 --query Parameter.Value --output text > new-key-pair.pem
```

필수 IAM 권한

AWS CloudFormation이 사용자를 대신하여 파라미터 스토어 파라미터를 관리할 수 있도록 하려면 AWS CloudFormation 또는 사용자가 수임하는 IAM 역할에 다음 권한이 있어야 합니다.

- `ssm:PutParameter` - 이 프라이빗 키 구성 요소에 대한 파라미터를 생성할 수 있는 권한을 부여합니다.
- `ssm:DeleteParameter` - 프라이빗 키 구성 요소를 저장하는 데 사용된 파라미터를 삭제할 권한을 부여합니다. 이 권한은 AWS CloudFormation에서 키 페어를 가져왔거나 생성했는지에 관계없이 필요합니다.

AWS CloudFormation은 키 페어를 생성할 때만 파라미터를 생성하고 키 페어를 가져올 때는 파라미터를 생성하지 않지만, 스택에 의해 생성되거나 가져온 키 페어를 삭제할 때 AWS CloudFormation은 권한 확인을 수행하여 사용자에게 파라미터를 삭제할 권한이 있는지 확인합니다. AWS CloudFormation은 사용자 계정의 파라미터와 일치하지 않는 조작된 파라미터 이름을 사용하여 필요한 권한을 테스트합니다. 따라서 `AccessDeniedException` 오류 메시지에 조작된 파라미터 이름이 표시될 수 있습니다.

서드 파티 도구를 사용하여 키 페어를 생성하고 Amazon EC2로 퍼블릭 키 가져오기

Amazon EC2를 사용하여 키 페어를 생성하는 대신에 서드 파티 도구를 사용하여 RSA 또는 ED25519 키 페어를 생성한 다음에 Amazon EC2로 퍼블릭 키를 가져올 수 있습니다.

키 페어에 대한 요구 사항

- 지원되는 유형:
  - (Linux 및 Windows) RSA
  - (Linux만 해당) ED25519

    > **Note**
    >
    > ED25519 키는 Windows 인스턴스에서 지원되지 않습니다.

  - Amazon EC2는 DSA 키를 허용하지 않습니다.

- 지원되는 형식:
  - OpenSSH 퍼블릭 키 형식(Linux의 경우 `~/.ssh/authorized_keys`의 형식)
  - (Linux만 해당) EC2 Instance Connect API를 사용하는 동안 SSH를 사용하여 연결하는 경우 SSH2 형식도 지원됩니다.
  - SSH 프라이빗 키 파일 형식은 PEM 또는 PPK이어야 함
  - (RSA만 해당) Base64 인코딩 DER 형식
  - (RSA만 해당) RFC 4716에 지정된 SSH 퍼블릭 키 파일 형식
- 지원되는 길이:
  - 1024, 2048, 4096.
  - (Linux만 해당) 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. 예를 들어 ssh-keygen(표준 OpenSSH 설치 시 제공되는 도구)을 사용할 수 있습니다. 또는 Java, Ruby, Python 등 각종 프로그래밍 언어에서 제공하는 표준 라이브러리를 사용하여 키 페어를 생성해도 됩니다.

   > **Important**
   >
   > 프라이빗 키는 PEM 또는 PPK 형식이어야 합니다. 예를 들어 `ssh-keygen -m PEM`을 사용하여 PEM 형식으로 OpenSSH 키를 생성합니다.

2. 퍼블릭 키는 로컬 파일에 저장합니다. 예: `~/.ssh/my-key-pair.pub`(Windows, macOS) 또는 `C:\keys\my-key-pair.pub`(Windows). 이 파일의 파일 이름 확장자는 중요하지 않습니다.
3. 프라이빗 키를 확장자가 `.pem` 또는 `.ppk`인 로컬 파일에 저장합니다. 예: `~/.ssh/my-key-pair.pem` `~/.ssh/my-key-pair.ppk`(Linux, macOS) 또는 `C:\keys\my-key-pair.pem` `C:\keys\my-key-pair.ppk`(Windows). 파일 확장자는 인스턴스에 연결하는 데 사용하는 도구에 따라 특정 파일 형식이 필요하므로 중요합니다. OpenSSH에는 `.pem` 파일이 필요한 반면, PuTTY에는 `.ppk` 파일이 필요합니다.

   > **Important**
   >
   > 프라이빗 키 파일을 안전한 장소에 저장합니다. 인스턴스를 시작할 때 퍼블릭 키의 이름을 제공하고, 인스턴스에 연결할 때마다 해당 프라이빗 키를 제공해야 합니다.

키 페어를 만든 후 다음 방법 중 하나를 사용하여 퍼블릭 키를 Amazon EC2로 가져옵니다.

Console

Amazon EC2로 퍼블릭 키를 가져오려면

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. [이름(Name)]에 퍼블릭 키를 설명하는 이름을 입력합니다. 이름에는 최대 255자의 ASCII 문자를 포함할 수 있습니다. 선행 또는 후행 공백을 포함할 수 없습니다.

   > **Note**
   >
   > EC2 콘솔에서 인스턴스에 연결할 때 콘솔은 프라이빗 키 파일의 이름으로 이 이름을 제안합니다.

5. 찾아보기를 선택하여 퍼블릭 키를 탐색하고 선택하거나 퍼블릭 키의 내용을 퍼블릭 키 내용 필드에 붙여 넣습니다.
6. 키 페어 가져오기를 선택합니다.
7. 가져온 퍼블릭 키가 키 페어 목록에 나타나는지 확인합니다.

AWS CLI

Amazon EC2로 퍼블릭 키를 가져오려면

import-key-pair 명령을 사용합니다.

```
aws ec2 import-key-pair \
    --key-name my-key-pair \
    --public-key-material fileb://path/my-key-pair.pub
```

키 페어를 성공적으로 가져왔는지 확인하려면

describe-key-pairs 명령을 사용합니다.

```
aws ec2 describe-key-pairs --key-names my-key-pair
```

PowerShell

Amazon EC2로 퍼블릭 키를 가져오려면

Import-EC2KeyPair cmdlet을 사용하세요.

```
$publickey=[Io.File]::ReadAllText("C:\Users\TestUser\.ssh\id_rsa.pub")
Import-EC2KeyPair `
    -KeyName my-key-pair `
    -PublicKey $publickey
```

키 페어를 성공적으로 가져왔는지 확인하려면

Get-EC2KeyPair cmdlet을 사용하세요.

```
Get-EC2KeyPair -KeyName my-key-pair
```

키 페어 설명

Amazon EC2에 저장된 키 페어를 설명할 수 있습니다. 퍼블릭 키 구성 요소를 검색하고 시작 시 지정되었던 공개 키를 식별할 수도 있습니다.

업무

- 키 페어 설명
- 퍼블릭 키 구성 요소 검색
- 시작 시 지정된 퍼블릭 키 식별

키 페어 설명

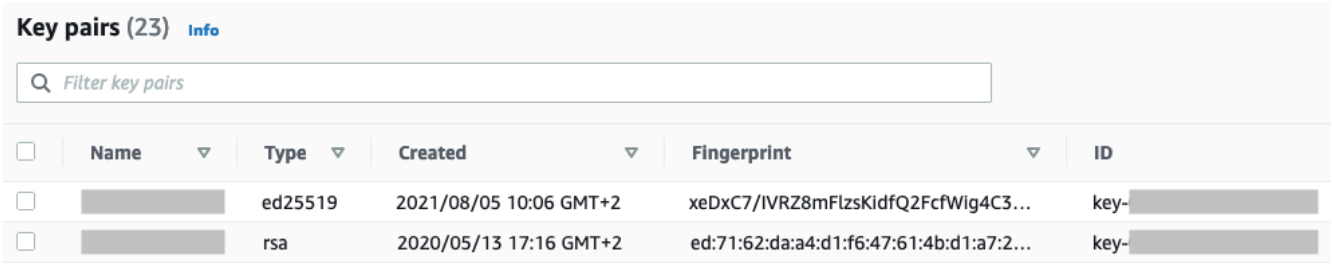
Amazon EC2 저장된 퍼블릭 키에 대한 정보를 볼 수 있으며, 이러한 정보에는 퍼블릭 키 이름, ID, 키 유형, 지문, 퍼블릭 키 구성 요소, 키가 Amazon EC2에서 생성된 날짜 및 시간(UTC 표준 시간대. 키가 서드 파티 도구에 의해 생성된 경우에는 키를 Amazon EC2로 가져온 날짜 및 시간), 퍼블릭 키와 연결된 모든 태그가 포함됩니다.

Console

키 페어에 대한 정보를 보는 방법

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

2. 왼쪽 탐색에서 키 페어(Key Pairs)를 선택합니다.
3. 키 페어(Key pairs) 테이블에서 각 퍼블릭 키에 대한 정보를 볼 수 있습니다.



4. 퍼블릭 키의 태그를 보려면 키 옆에 있는 확인란을 선택한 다음 작업, 태그 관리를 선택합니다.

AWS CLI

키 페어에 대한 정보를 보는 방법

describe-key-pairs 명령을 사용합니다.

```
aws ec2 describe-key-pairs --key-names key-pair-name
```

PowerShell

키 페어에 대한 정보를 보는 방법

Get-EC2KeyPair cmdlet을 사용하세요.

```
Get-EC2KeyPair -KeyName key-pair-name
```

퍼블릭 키 구성 요소 검색

키 페어에 대한 퍼블릭 키 구성 요소를 가져올 수 있습니다. 다음은 퍼블릭 키 예제입니다. 참고로, 가독성을 위해 줄 바꿈이 추가되어 있습니다.

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnBItntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0FzZ
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb7OzlPnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE
```

Private key

ssh-keygen을 사용하여 퍼블릭 키 구성 요소를 검색하는 방법(Linux)

로컬 Linux 또는 macOS 컴퓨터에서 ssh-keygen 명령을 사용합니다. 프라이빗 키(`.pem` 파일)를 다운로드한 경로를 지정합니다.

```
ssh-keygen -y -f /path_to_key_pair/my-key-pair.pem
```

이 ssh-keygen 명령이 작동하지 않으면 다음 chmod 명령을 실행하여 프라이빗 키 파일에 필요한 권한이 있는지 확인합니다.

```
chmod 400 key-pair-name.pem
```

PuTTYgen을 사용하여 퍼블릭 키 구성 요소를 검색하는 방법(Windows)

로컬 Windows 컴퓨터에서 PuTTYgen을 시작합니다. 로드(Load)를 선택합니다. `.ppk` 또는 `.pem` 프라이빗 키 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.

AWS CLI

퍼블릭 키 구성 요소를 검색하는 방법

다음 describe-key-pairs 명령을 사용하고 `--include-public-key` 옵션을 지정합니다.

```
aws ec2 describe-key-pairs \
    --key-names key-pair-name \
    --include-public-key \
    --query "KeyPairs[].PublicKey"
```

PowerShell

퍼블릭 키 구성 요소를 검색하는 방법

Get-EC2KeyPair cmdlet을 사용합니다.

```
(Get-EC2KeyPair -KeyName key-pair-name -IncludePublicKey $true).PublicKey
```

IMDSv2

Linux

Linux 인스턴스에서 다음과 같은 명령을 실행합니다.

```
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" http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
```

Windows

Windows 인스턴스에서 다음과 같은 cmdlet을 실행합니다.

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method GET -Uri http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
```

IMDSv1

Linux

Linux 인스턴스에서 다음 명령을 실행합니다.

```
curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
```

Windows

Windows 인스턴스에서 다음 cmdlet을 실행합니다.

```
Invoke-RestMethod -uri  http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
```

시작 시 지정된 퍼블릭 키 식별

인스턴스를 시작할 때 퍼블릭 키를 지정하면 인스턴스를 통해 퍼블릭 키 이름이 기록됩니다. 인스턴스에 대해 보고된 퍼블릭 키 이름은 인스턴스의 퍼블릭 키를 변경하거나 퍼블릭 키를 추가하더라도 변경되지 않습니다.

Console

시작 시 지정된 퍼블릭 키를 식별하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다.
4. 세부 정보 탭의 인스턴스 세부 정보에서 시작 시 할당된 키 페어를 찾습니다.

AWS CLI

시작 시 지정된 퍼블릭 키를 식별하는 방법

다음 describe-instances 명령을 사용합니다.

```
aws ec2 describe-instances \
    --instance-id i-1234567890abcdef0 \
    --query "Reservations[].Instances[].KeyName" \
    --output text
```

출력의 예시는 다음과 같습니다.

```
key-pair-name
```

PowerShell

시작 시 지정된 퍼블릭 키를 식별하는 방법

Get-EC2Instance cmdlet을 사용합니다.

```
(Get-EC2Instance -InstanceId i-1234567890abcdef0).Instances | Select KeyName
```

출력의 예시는 다음과 같습니다.

```
KeyName
-------
key-pair-name
```

키 페어 삭제

키 페어를 삭제할 수 있습니다. 그러면 Amazon EC2에 저장된 퍼블릭 키가 제거됩니다. 퍼블릭 키를 삭제해도 대응하는 프라이빗 키는 삭제되지 않습니다.

다음과 같은 방법을 사용하여 퍼블릭 키를 삭제하면 키 페어를 생성하거나 가져올 때 Amazon EC2에 저장한 퍼블릭 키만 삭제됩니다. 퍼블릭 키를 삭제하면 퍼블릭 키를 추가한 인스턴스의 퍼블릭 키는 인스턴스를 시작했을 때 또는 나중에 제거되지 않습니다. 로컬 컴퓨터의 프라이빗 키도 삭제되지 않습니다. 프라이빗 키(`.pem`) 파일이 그대로 있으면 Amazon EC2에서 삭제한 퍼블릭 키를 사용하여 시작한 인스턴스에 계속 연결할 수 있습니다.

> **Important**
>
> Auto Scaling 그룹을 사용 중인 경우(예: Elastic Beanstalk 환경에서) 연결된 시작 템플릿 또는 시작 구성에 삭제하려는 퍼블릭 키가 지정되지 않았는지 확인하세요. Amazon EC2 Auto Scaling은 비정상 인스턴스가 감지될 경우 대체 인스턴스를 시작합니다. 그러나 퍼블릭 키를 찾을 수 없으면 인스턴스가 시작되지 않습니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서에서 시작 템플릿을 참조하세요.

Console

Amazon EC2의 퍼블릭 키를 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 Actions(작업), Delete(삭제)를 차례로 선택합니다.
4. 확인 필드에 `Delete`를 입력한 다음 삭제를 선택합니다.

AWS CLI

Amazon EC2의 퍼블릭 키를 삭제하려면

delete-key-pair 명령을 사용합니다.

```
aws ec2 delete-key-pair --key-name my-key-pair
```

PowerShell

Amazon EC2의 퍼블릭 키를 삭제하려면

Remove-EC2KeyPair cmdlet을 사용하세요.

```
Remove-EC2KeyPair -KeyName my-key-pair
```

Linux 인스턴스에 대한 퍼블릭 키 추가 또는 바꾸기

프라이빗 키를 분실한 경우 키 페어를 사용하는 모든 인스턴스에 대한 액세스 권한도 사라집니다. 시작 시 지정한 키 페어와 다른 키 페어를 사용하여 인스턴스에 연결하는 방법에 대한 자세한 내용은 프라이빗 키를 분실했습니다를 참조하세요.

인스턴스를 시작할 때 키 페어를 지정할 수 있습니다. 시작 시 키 페어를 지정하면 인스턴스가 처음 부팅될 때 퍼블릭 키 구성 요소가 `~/.ssh/authorized_keys` 내 항목의 Linux 인스턴스에 배치됩니다.

인스턴스에서 새 퍼블릭 키를 추가하거나 인스턴스의 퍼블릭 키를 대체(기존 퍼블릭 키 삭제 및 새 퍼블릭 키 추가)하여 인스턴스의 기본 시스템 계정에 액세스하는 데 사용되는 키 페어를 변경할 수 있습니다. 인스턴스에서 모든 퍼블릭 키를 제거할 수도 있습니다. 키 페어를 추가하거나 교체하려면 인스턴스에 연결할 수 있어야 합니다.

다음과 같은 이유로 키 페어를 추가하거나 바꿀 수 있습니다.

- 조직 내 사용자가 별도의 키 페어를 사용하여 시스템 사용자에게 액세스해야 하는 경우 인스턴스에 퍼블릭 키 페어를 추가할 수 있습니다.
- 프라이빗 키(`.pem` 파일)의 복사본을 보유하고 있는 누군가의 인스턴스 연결을 차단하려는 경우(예: 퇴사자) 인스턴스의 퍼블릭 키를 삭제하고 새 키로 교체할 수 있습니다.
- 인스턴스에서 Linux AMI 생성하는 경우 퍼블릭 키 구성 요소가 인스턴스에서 AMI로 복사됩니다. AMI에서 인스턴스를 시작하는 경우 새 인스턴스에 원본 인스턴스의 키 페어가 포함됩니다. 프라이빗 키가 있는 사용자가 새 인스턴스에 연결할 수 없도록 AMI를 생성하기 전에 원본 인스턴스에서 퍼블릭 키를 제거할 수 있습니다.

다음 절차를 사용하여 기본 사용자(예: `ec2-user`)의 키 페어를 수정합니다. 인스턴스에 사용자를 추가하는 방법에 대한 자세한 내용은 인스턴스에 대한 운영 체제 설명서를 참조하세요.

키 페어를 추가 또는 교체하는 방법

1. Amazon EC2 콘솔 또는 타사 도구를 사용하여 새 키 페어를 만듭니다.
2. 새 키 페어에서 퍼블릭 키를 검색합니다. 자세한 내용은 퍼블릭 키 구성 요소 검색 섹션을 참조하세요.
3. 기존 프라이빗 키를 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.

> **Important**
>
> Auto Scaling 그룹을 사용 중인 경우 교체하려는 키 페어가 시작 템플릿 또는 시작 구성에 지정되지 않았는지 확인합니다. Amazon EC2 Auto Scaling은 비정상 인스턴스가 감지될 경우 대체 인스턴스를 시작합니다. 그러나 키 페어를 찾지 못하면 인스턴스 시작이 실패합니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서에서 시작 템플릿을 참조하세요.

인스턴스에서 퍼블릭 키를 제거하려면

1. 인스턴스에 연결합니다.
2. 원하는 텍스트 편집기를 사용하여 인스턴스에서 `.ssh/authorized_keys` 파일을 엽니다. 퍼블릭 키 정보를 삭제한 후 파일을 저장합니다.

> **Warning**
>
> 인스턴스에서 모든 퍼블릭 키를 제거하고 인스턴스 연결을 끊은 후에는 AMI에서 다른 로그인 방법이 제공되지 않으면 다시 연결할 수 없습니다.

키 페어의 지문 확인

키 페어의 지문을 확인하려면 Amazon EC2 콘솔의 키 페어 페이지에 표시되거나 describe-key-pairs 명령으로 반환된 지문을 로컬 컴퓨터에서 프라이빗 키를 사용하여 생성한 지문과 비교하세요. 이 지문들은 일치해야 합니다.

Amazon EC2가 지문을 계산할 때 Amazon EC2는 = 문자를 사용하여 지문에 패딩을 추가할 수 있습니다. ssh-keygen 등의 다른 도구는 이 패딩을 생략할 수 있습니다.

키 페어의 지문이 아닌 Linux EC2 인스턴스의 지문을 확인하려고 하는 경우에는 인스턴스 지문 가져오기를 참조하세요.

지문 계산 방법

Amazon EC2는 서로 다른 해시 함수를 사용하여 RSA 및 ED25519 키 페어의 지문을 계산합니다. 또한 RSA 키 페어의 경우 Amazon EC2는 키 페어가 Amazon EC2에서 생성되었는지 아니면 Amazon EC2로 가져왔는지에 따라 서로 다른 해시 함수를 사용하여 지문을 다르게 계산합니다.

다음 표에는 Amazon EC2에서 생성하여 Amazon EC2로 가져오는 RSA 및 ED25519 키 페어의 지문을 계산하는 데 사용되는 해시 함수가 나열되어 있습니다.

(Linux 인스턴스) 지문을 계산하는 데 사용되는 해시 함수

키 페어 소스	RSA 키 페어(Windows 및 Linux)	ED25519 키 페어(Linux)
Amazon EC2에서 생성	SHA-1	SHA-256
Amazon EC2로 가져오기	MD5[1]	SHA-256

Amazon EC2로 퍼블릭 RSA 키를 가져오면 지문은 MD5 해시 함수를 통해 계산됩니다. 이 방식은 서드파티 도구를 사용했든지 Amazon EC2를 사용하여 생성한 기존 프라이빗 키에서 새 퍼블릭 키를 생성했든지, 키 페어를 생성한 방법과 상관없이 동일합니다.

서로 다른 리전에서 동일한 키 페어를 사용하는 경우

동일한 키 페어를 사용하여 다른 AWS 리전의 인스턴스에 연결하려는 경우, 퍼블릭 키를 사용할 모든 리전으로 가져와야 합니다. Amazon EC2를 사용하여 키 페어를 생성하는 경우 퍼블릭 키를 다른 리전으로 가져올 수 있도록 퍼블릭 키 구성 요소 검색할 수 있습니다.

> **Note**
>
> - Amazon EC2를 사용하여 RSA 키 페어를 생성한 다음 Amazon EC2 프라이빗 키에서 퍼블릭 키를 생성하는 경우, 가져온 퍼블릭 키의 지문은 원래 퍼블릭 키와 다릅니다. 이유는 Amazon EC2를 사용하여 생성된 원래 RSA 키의 지문이 SHA-1 해시 함수를 사용하여 계산되고, 가져온 RSA 키의 지문은 MD5 해시 함수를 사용하여 계산되기 때문입니다.
> - ED25519 키 페어의 경우 지문은 동일한 SHA-256 해시 함수가 지문 계산에 사용되기 때문에 Amazon EC2에서 생성했는지 또는 Amazon EC2로 가져왔는지와 관계없이 동일합니다.

프라이빗 키에서 지문 생성

다음 명령 중 하나를 사용하여 로컬 머신의 프라이빗 키에서 지문을 생성합니다.

Windows 로컬 컴퓨터를 사용 중인 경우 Linux용 Windows 하위 시스템(WSL)을 사용하여 다음 명령을 실행할 수 있습니다. WSL을 사용하여 Windows에 Linux를 설치하는 방법의 지침을 이용하여 WSL과 Linux 배포를 설치하세요. 지침 속 사례는 Linux의 Ubuntu 배포를 설치하는 것이지만, 다른 배포의 설치에도 활용할 수 있습니다. 변경 사항을 적용하려면 컴퓨터를 다시 시작하라는 안내가 표시됩니다.

- Amazon EC2를 사용하여 키 페어를 만든 경우

  다음 예제와 같이 OpenSSL 도구를 사용하여 지문을 생성합니다.

  RSA 키 페어의 경우:

  ```
  openssl pkcs8 -in path_to_private_key -inform PEM -outform DER -topk8 -nocrypt |
   openssl sha1 -c
  ```

  (Linux 인스턴스) ED25519 키 페어의 경우:

  ```
  ssh-keygen -l -f path_to_private_key
  ```

- (RSA 키 페어만 해당) 퍼블릭 키를 Amazon EC2로 가져온 경우

  예를 들어, 타사 도구를 사용하거나 Amazon EC2로 생성된 기존 프라이빗 키에서 새 퍼블릭 키를 생성하여 키 페어를 생성한 방법에 관계없이 이 절차를 따를 수 있습니다.

  다음 예제와 같이 OpenSSL 도구를 사용하여 지문을 생성합니다.

```
openssl rsa -in path_to_private_key -pubout -outform DER | openssl md5 -c
```

- OpenSSH 7.8 이상을 사용하여 OpenSSH 키 페어를 생성하고 Amazon EC2로 퍼블릭 키를 가져온 경우

  다음 예제와 같이 ssh-keygen을 사용하여 지문을 생성합니다.

  RSA 키 페어의 경우:

```
ssh-keygen -ef path_to_private_key -m PEM | openssl rsa -RSAPublicKey_in -outform DER
 | openssl md5 -c
```

  (Linux 인스턴스) ED25519 키 페어의 경우:

```
ssh-keygen -l -f path_to_private_key
```

EC2 인스턴스에 대한 Amazon EC2 보안 그룹

보안 그룹은 EC2 인스턴스에 대한 수신 및 발신 트래픽을 제어하는 가상 방화벽 역할을 합니다. 인바운드 규칙은 인스턴스로 들어오는 트래픽을 제어하고 아웃바운드 규칙은 인스턴스에서 나가는 트래픽을 제어합니다. 인스턴스를 시작할 때 하나 이상의 보안 그룹을 지정할 수 있습니다. 보안 그룹을 지정하지 않을 경우 Amazon EC2에서 VPC에 대한 기본 보안 그룹이 사용됩니다. 인스턴스를 시작한 이후에는 해당 보안 그룹을 변경할 수 없습니다.

보안은 AWS와 사용자의 공동 책임입니다. 자세한 내용은 Amazon EC2의 보안 섹션을 참조하세요. AWS에서는 인스턴스의 보안을 유지하기 위한 도구 중 하나로 보안 그룹을 제공하며, 사용자는 보안 요구 사항에 맞게 보안 그룹을 구성해야 합니다. 보안 그룹으로 완전히 충족되지 않는 요구 사항이 있는 경우 보안 그룹을 사용하면서 인스턴스에 대한 자체 방화벽을 유지합니다.

요금

보안 그룹을 사용해도 추가 요금이 부과되지 않습니다.

내용

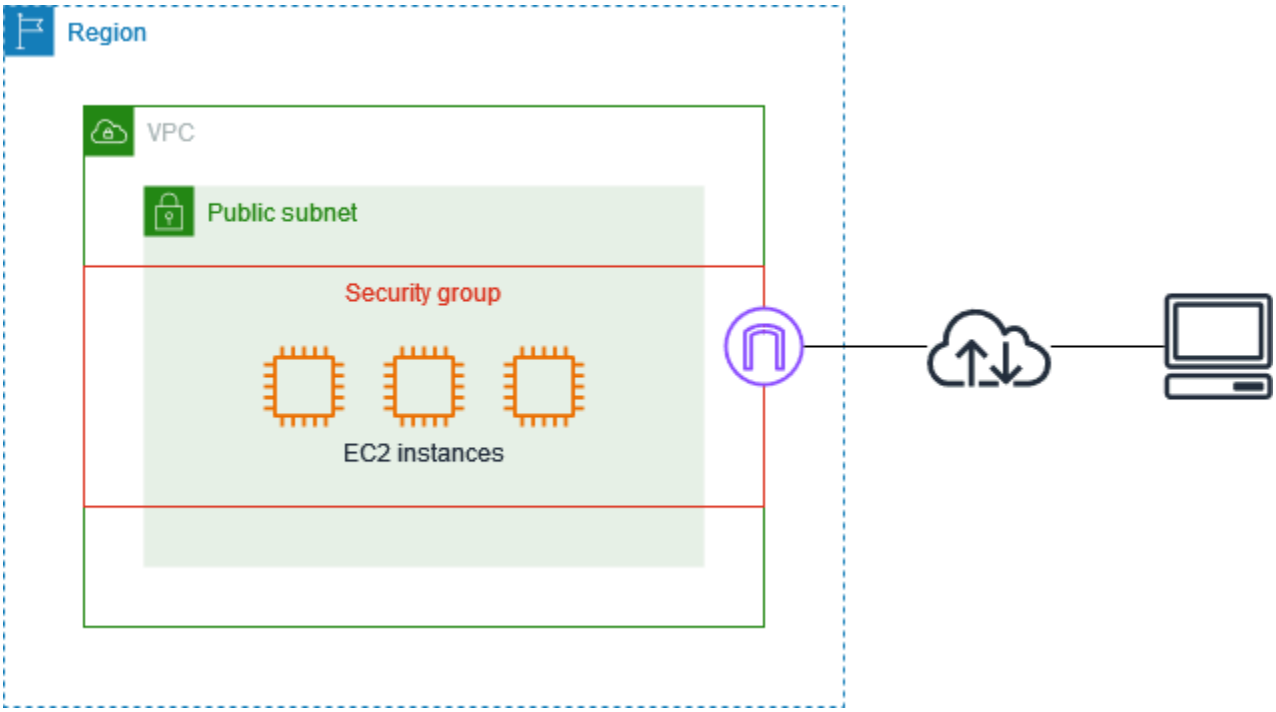
- 개요
- Amazon EC2 인스턴스에 대한 보안 그룹을 생성합니다.
- Amazon EC2 인스턴스에 대한 보안 그룹 변경

- Amazon EC2 보안 그룹 삭제
- Amazon EC2 보안 그룹 연결 추적
- 다양한 사용 사례에 대한 보안 그룹 규칙

개요

보안 그룹은 보안 그룹이 생성된 VPC에서만 사용할 수 있습니다. 각 인스턴스를 여러 보안 그룹과 연결하고, 각 보안 그룹을 여러 인스턴스와 연결할 수 있습니다. 연결된 인스턴스와 트래픽을 주고받을 수 있게 하는 규칙을 각 보안 그룹에 추가합니다. 언제든지 보안 그룹에 대한 규칙을 수정할 수 있습니다. 새 규칙 및 수정된 규칙은 보안 그룹에 연결된 모든 인스턴스에 자동으로 적용됩니다. Amazon EC2는 트래픽이 인스턴스에 도달하도록 허용할지 여부를 결정할 때 인스턴스와 연결된 모든 보안 그룹에서 모든 규칙을 평가합니다. 자세한 내용은 Amazon VPC 사용 설명서의 보안 그룹 규칙을 참조하세요.

다음 다이어그램은 서브넷, 인터넷 게이트웨이 및 보안 그룹이 있는 VPC를 보여줍니다. 서브넷에 EC2 인스턴스가 포함되어 있습니다. 보안 그룹은 인스턴스에 연결됩니다. 인스턴스에 연결되는 트래픽은 보안 그룹 규칙에서 허용되는 트래픽이 유일합니다. 예를 들어 네트워크에서 SSH 트래픽을 허용하는 규칙이 보안 그룹에 포함된 경우 SSH를 사용하여 컴퓨터에서 인스턴스로 연결할 수 있습니다. 연결된 리소스에서 모든 트래픽을 허용하는 규칙이 보안 그룹에 포함된 경우 각 인스턴스는 다른 인스턴스에서 전송된 모든 트래픽을 수신할 수 있습니다.



보안 그룹은 상태가 저장됩니다. 사용자가 인스턴스에서 요청을 전송하면 해당 요청의 응답 트래픽은 인바운드 보안 그룹 규칙에 관계없이 인바운드 흐름이 허용됩니다. 또한 아웃바운드 규칙과 무관하게 허용된 인바운드 트래픽에 대한 반응으로 외부로 나가는 흐름이 수행됩니다. 자세한 내용은 연결 추적 섹션을 참조하세요.

Amazon EC2 인스턴스에 대한 보안 그룹을 생성합니다.

보안 그룹은 연결된 인스턴스에 대한 방화벽 역할을 하여 인스턴스 수준에서 인바운드 트래픽과 아웃바운드 트래픽을 모두 제어합니다. SSH(Linux 인스턴스) 또는 RDP(Windows 인스턴스)를 사용하여 인스턴스에 연결할 수 있도록 규칙을 보안 그룹에 추가할 수 있습니다. 웹 서버로 향하는 HTTP 및 HTTPS 트래픽과 같은 클라이언트 트래픽을 허용하는 규칙을 추가할 수도 있습니다.

인스턴스를 시작할 때 인스턴스에 보안 그룹을 연결할 수 있습니다. 보안 그룹에서 연결된 규칙을 추가하거나 제거하면 해당 보안 그룹을 연결한 모든 인스턴스에 변경 내용이 자동으로 적용됩니다.

인스턴스를 시작한 이후에는 추가 보안 그룹을 연결할 수 있습니다. 자세한 내용은 Amazon EC2 인스턴스에 대한 보안 그룹 변경 섹션을 참조하세요.

보안 그룹을 생성할 때 또는 나중에 인바운드 및 아웃바운드 보안 그룹 규칙을 추가할 수 있습니다. 자세한 내용은 보안 그룹 규칙 구성 섹션을 참조하세요. 보안 그룹에 추가할 수 있는 규칙의 예제는 다양한 사용 사례에 대한 보안 그룹 규칙의 내용을 참조하세요.

고려 사항

- 기본적으로 처음에 새 보안 그룹에는 리소스에서 나가는 모든 트래픽을 허용하는 아웃바운드 규칙만 적용됩니다. 인바운드 트래픽을 사용하거나 아웃바운드 트래픽을 제한하려면 규칙을 추가해야 합니다.
- 인스턴스에 대한 SSH 또는 RDP 액세스를 허용하는 규칙의 원본을 구성할 때는 모든 위치에서의 액세스를 허용하지 마세요.인터넷의 모든 IP 주소에서 인스턴스에 액세스할 수 있게 됩니다. 테스트 환경에서 잠시 사용하는 것은 괜찮지만 프로덕션 환경에서는 안전하지 않습니다.
- 특정 포트에 대한 규칙이 여러 개 있는 경우 Amazon EC2는 최대 허용 규칙을 적용합니다. 예를 들어 IP 주소 203.0.113.1에서 TCP 포트 22(SSH)에 액세스할 수 있도록 허용하는 규칙과 어디서나 TCP 포트 22에 액세스할 수 있도록 허용하는 또 다른 규칙이 있는 경우 모든 사람이 TCP 포트 22에 액세스할 수 있습니다.
- 인스턴스에 여러 보안 그룹을 연결할 수 있습니다. 따라서 한 인스턴스에 수백 개의 규칙이 적용될 수 있습니다. 이로 인해 인스턴스에 액세스할 때 문제가 발생할 수 있습니다. 규칙을 최대한 간략하게 만드는 것이 좋습니다.

- 보안 그룹을 규칙의 소스 또는 대상으로 지정할 경우 규칙은 보안 그룹과 연결된 모든 인스턴스에 영향을 줍니다. 유입 트래픽은 퍼블릭 IP 주소 또는 탄력적 IP 주소가 아닌 원본 보안 그룹과 연결된 인스턴스의 프라이빗 IP 주소를 기반으로 허용됩니다. IP 주소에 대한 자세한 내용은 Amazon EC2 인스턴스 IP 주소 지정 주제를 참조하세요.
- Amazon EC2에서는 기본적으로 포트 25의 트래픽을 차단합니다. 자세한 내용은 포트 25를 사용하여 전송되는 이메일 관련 제한 섹션을 참조하세요.

Console

보안 그룹을 생성하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다.
3. 보안 그룹 생성을 선택합니다.
4. 보안 그룹의 설명이 포함된 이름과 간단한 설명을 입력합니다. 보안 그룹을 생성한 후에는 보안 그룹에 대한 이름과 설명을 변경할 수 없습니다.
5. VPC에서 Amazon EC2 인스턴스를 실행할 VPC를 선택합니다.
6. (선택 사항) 인바운드 규칙을 추가하려면 인바운드 규칙을 선택합니다. 각 규칙에 대해 규칙 추가를 선택하고 프로토콜, 포트 및 소스를 지정합니다. 예를 들어 SSH 트래픽을 허용하려면 유형에서 SSH를 선택하고 소스에서 컴퓨터 또는 네트워크의 퍼블릭 IPv4 주소를 지정합니다.
7. (선택 사항) 아웃바운드 규칙을 추가하려면 아웃바운드 규칙을 선택합니다. 각 규칙에 대해 규칙 추가를 선택하고 프로토콜, 포트 및 대상을 지정합니다. 아니면 모든 아웃바운드 트래픽을 허용하는 기본 규칙을 유지할 수 있습니다.
8. (선택 사항) 태그를 추가하려면 Add new tag(새 태그 추가)를 선택하고 태그 키와 태그 값을 입력합니다.
9. 보안 그룹 생성을 선택합니다.

AWS CLI

보안 그룹을 생성하는 방법

다음 create-security-group 명령을 사용합니다.

```
aws ec2 create-security-group \
    --group-name my-security-group \
    --description "my security group" \
```

```
    --vpc-id vpc-1234567890abcdef0
```

규칙을 추가하는 예제는 the section called “보안 그룹 규칙 구성” 단원을 참조하세요.

PowerShell

보안 그룹을 생성하는 방법

New-EC2SecurityGroup cmdlet을 사용합니다.

```
New-EC2SecurityGroup `
    -GroupName my-security-group `
    -Description "my security group" `
    -VpcId vpc-1234567890abcdef0
```

규칙을 추가하는 예제는 the section called “보안 그룹 규칙 구성” 단원을 참조하세요.

Amazon EC2 인스턴스에 대한 보안 그룹 변경

Amazon EC2 인스턴스를 시작할 때 해당 인스턴스에 대해 보안 그룹을 지정할 수 있습니다. 인스턴스를 시작한 이후에는 보안 그룹을 추가하거나 제거할 수 없습니다. 또한 언제든지 연결된 보안 그룹에 대한 보안 그룹 규칙을 추가, 제거 또는 편집할 수 있습니다.

보안 그룹은 네트워크 인터페이스와 연결됩니다. 보안 그룹을 추가하거나 제거하면 기본 네트워크 인터페이스와 연결된 보안 그룹이 변경됩니다. 보조 네트워크 인터페이스와 연결된 보안 그룹을 변경할 수도 있습니다. 자세한 내용은 네트워크 인터페이스 속성 수정 섹션을 참조하세요.

업무

- 보안 그룹 추가 또는 제거
- 보안 그룹 규칙 구성

보안 그룹 추가 또는 제거

인스턴스를 시작한 후 연결된 보안 그룹 목록에서 보안 그룹을 추가하거나 제거할 수 있습니다. 여러 보안 그룹을 인스턴스와 연결할 경우 각 보안 그룹의 규칙이 유효하게 결합된 단일 규칙 세트가 생성됩니다. Amazon EC2는 이 규칙 세트를 사용하여 트래픽을 허용할지 여부를 결정합니다.

요구 사항

- 인스턴스는 `running` 또는 `stopped` 상태여야 합니다.

- 보안 그룹은 VPC에 고유합니다. 하나 이상의 인스턴스와 보안 그룹을 연결할 수 있습니다.

Console

인스턴스에 대한 보안 그룹 변경

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택한 다음 [작업(Actions)], [보안(Security)], [보안 그룹 변경(Change security groups)]을 선택합니다.
4. 연결된 보안 그룹에서는 목록에서 보안 그룹을 선택하고 보안 그룹 추가를 선택합니다.

   이미 연결된 보안 그룹을 제거하려면 보안 그룹의 제거를 선택합니다.
5. 저장을 선택합니다.

AWS CLI

인스턴스에 대한 보안 그룹 변경

다음 modify-instance-attribute 명령을 사용합니다.

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --groups sg-1234567890abcdef0
```

PowerShell

인스턴스에 대한 보안 그룹 변경

Edit-EC2InstanceAttribute cmdlet을 사용합니다.

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -Group sg-1234567890abcdef0
```

보안 그룹 규칙 구성

보안 그룹을 생성한 후 해당 보안 그룹 규칙을 추가, 업데이트 및 삭제할 수 있습니다. 규칙을 추가, 업데이트 또는 삭제하면 변경 내용이 보안 그룹과 연결된 모든 리소스에 자동으로 적용됩니다.

보안 그룹에 추가할 수 있는 규칙의 예제는 다양한 사용 사례에 대한 보안 그룹 규칙의 내용을 참조하세요.

소스 및 대상

다음을 인바운드 규칙의 소스 또는 아웃바운드 규칙의 대상으로 지정할 수 있습니다.

- 사용자 지정 - IPv4 CIDR 블록, IPv6 CIDR 블록, 다른 보안 그룹 또는 접두사 목록.
- Anywhere-IPv4 – 0.0.0.0/0 IPv4 CIDR 블록.
- Anywhere-IPv6 – ::/0 IPv6 CIDR 블록.
- 내 IP: 로컬 컴퓨터의 퍼블릭 IPv4 주소.

> **Warning**
>
> 포트 22(SSH) 또는 3389(RDP)에 대한 인바운드 규칙을 추가하는 경우 인스턴스에 액세스해야 하는 특정 IP 주소 또는 주소 범위만 승인하는 것이 좋습니다. Anywhere-IPv4를 선택하면 모든 IPv4 주소의 트래픽을 허용하고 지정된 프로토콜을 사용하여 인스턴스에 액세스합니다. Anywhere-IPv6를 선택하면 모든 IPv6 주소의 트래픽을 허용하고 지정된 프로토콜을 사용하여 인스턴스에 액세스합니다.

Console

보안 그룹 규칙을 구성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 보안 그룹을 선택합니다.
3. 보안 그룹을 선택합니다.
4. 인바운드 규칙을 편집하려면 작업 또는 인바운드 규칙 탭에서 인바운드 규칙 편집을 선택합니다.

   a. 규칙을 추가하려면 규칙 추가를 선택한 다음 규칙의 유형, 프로토콜, 포트 및 소스를 입력합니다.

      유형이 TCP 또는 UDP인 경우 허용할 포트 범위를 입력해야 합니다. 사용자 지정 ICMP의 경우 프로토콜(Protocol)에서 ICMP 유형 이름을 선택하고, 해당되는 경우 포트 범위(Port range)에서 코드 이름을 선택해야 합니다. 다른 유형에 대해 프로토콜과 포트 범위가 구성됩니다.

b. 규칙을 업데이트하려면 필요에 따라 프로토콜, 설명 및 소스를 변경합니다. 하지만 소스 유형을 변경할 수는 없습니다. 예를 들어 소스가 IPv4 CIDR 블록인 경우 IPv6 CIDR 블록, 접두사 목록 또는 보안 그룹을 지정할 수 없습니다.

c. 규칙을 삭제하려면 해당 삭제 버튼을 선택합니다.

5. 아웃바운드 규칙을 편집하려면 작업 또는 아웃바운드 규칙 탭에서 아웃바운드 규칙 편집을 선택합니다.

a. 규칙을 추가하려면 규칙 추가를 선택한 다음 규칙의 유형, 프로토콜, 포트 및 대상을 입력합니다. 또한 설명을 입력할 수 있습니다(선택 사항).

유형이 TCP 또는 UDP인 경우 허용할 포트 범위를 입력해야 합니다. 사용자 지정 ICMP의 경우 프로토콜(Protocol)에서 ICMP 유형 이름을 선택하고, 해당되는 경우 포트 범위(Port range)에서 코드 이름을 선택해야 합니다. 다른 유형에 대해 프로토콜과 포트 범위가 구성됩니다.

b. 규칙을 업데이트하려면 필요에 따라 프로토콜, 설명 및 소스를 변경합니다. 하지만 소스 유형을 변경할 수는 없습니다. 예를 들어 소스가 IPv4 CIDR 블록인 경우 IPv6 CIDR 블록, 접두사 목록 또는 보안 그룹을 지정할 수 없습니다.

c. 규칙을 삭제하려면 해당 삭제 버튼을 선택합니다.

6. 규칙 저장을 선택합니다.

AWS CLI

보안 그룹 규칙을 추가하는 방법

authorize-security-group-ingress 명령을 사용하여 인바운드 규칙을 추가합니다. 다음 예제에서는 지정된 접두사 목록 중 CIDR 블록의 인바운드 SSH 트래픽을 허용합니다.

```
aws ec2 authorize-security-group-ingress \
    --group-id sg-1234567890abcdef0 \
    --ip-permissions
 'IpProtocol=tcp,FromPort=22,ToPort=22,PrefixListIds=[{PrefixListId=pl-
f8a6439156EXAMPLE}]'
```

authorize-security-group-egress 명령을 사용하여 아웃바운드 규칙을 추가합니다. 다음 예제에서는 지정된 보안 그룹이 있는 인스턴스에 포트 80의 아웃바운드 TCP 트래픽을 허용합니다.

```
aws ec2 authorize-security-group-egress \
```

```
    --group-id sg-1234567890abcdef0 \
    --ip-permissions
 'IpProtocol=tcp,FromPort=80,ToPort=80,UserIdGroupPairs=[{GroupId=sg-0aad1c26bb6EXAMPLE}]'
```

무효 보안 그룹 규칙을 제거하는 방법

다음 revoke-security-group-ingress 명령을 사용하여 인바운드 규칙을 제거합니다.

```
aws ec2 revoke-security-group-egress \
    --group id sg-1234567890abcdef0 \
    --security-group-rule-ids sgr-09ed298024EXAMPLE
```

다음 revoke-security-group-egress 명령을 사용하여 아웃바운드 규칙을 제거합니다.

```
aws ec2 revoke-security-group-ingress \
    --group id sg-1234567890abcdef0 \
    --security-group-rule-ids sgr-0352250c1aEXAMPLE
```

보안 그룹 규칙을 수정하는 방법

modify-security-group-rules 명령을 사용합니다. 다음 예제에서는 지정된 보안 그룹 규칙의 IPv4 CIDR 블록을 변경합니다.

```
aws ec2 modify-security-group-rules \
    --group id sg-1234567890abcdef0 \
    --security-group-rules
 'SecurityGroupRuleId=sgr-09ed298024EXAMPLE,SecurityGroupRule={IpProtocol=tcp,FromPort=80,To
```

PowerShell

보안 그룹 규칙을 추가하는 방법

Grant-EC2SecurityGroupIngress cmdlet을 사용하여 인바운드 규칙을 추가합니다. 다음 예제에서는 지정된 접두사 목록 중 CIDR 블록의 인바운드 SSH 트래픽을 허용합니다.

```
$plid = New-Object -TypeName Amazon.EC2.Model.PrefixListId
$plid.Id = "pl-f8a6439156EXAMPLE"
Grant-EC2SecurityGroupIngress `
    -GroupId sg-1234567890abcdef0 `
    -IpPermission @{IpProtocol="tcp"; FromPort=22; ToPort=22; PrefixListIds=$plid}
```

Grant-EC2SecurityGroupEgress cmdlet을 사용하여 아웃바운드 규칙을 추가합니다. 다음 예제에서는 지정된 보안 그룹이 있는 인스턴스에 포트 80의 아웃바운드 TCP 트래픽을 허용합니다.

```
$uigp = New-Object -TypeName Amazon.EC2.Model.UserIdGroupPair
$uigp.GroupId = "sg-0aad1c26bb6EXAMPLE"
Grant-EC2SecurityGroupEgress `
    -GroupId sg-1234567890abcdef0 `
    -IpPermission @{IpProtocol="tcp"; FromPort=80; ToPort=80; UserIdGroupPairs=
$uigp}
```

무효 보안 그룹 규칙을 제거하는 방법

Revoke-EC2SecurityGroupIngress cmdlet을 사용하여 인바운드 규칙을 제거합니다.

```
Revoke-EC2SecurityGroupIngress `
    -GroupId sg-1234567890abcdef0 `
    -SecurityGroupRuleId sgr-09ed298024EXAMPLE
```

Revoke-EC2SecurityGroupEgress cmdlet을 사용하여 아웃바운드 규칙을 제거합니다.

```
Revoke-EC2SecurityGroupEgress `
    -GroupId sg-1234567890abcdef0 `
    -SecurityGroupRuleId sgr-0352250c1aEXAMPLE
```

보안 그룹 규칙을 수정하는 방법

Edit-EC2SecurityGroupRule cmdlet을 사용합니다. 다음 예제에서는 지정된 보안 그룹 규칙의 IPv4 CIDR 블록을 변경합니다.

```
$sgrr = New-Object -TypeName Amazon.EC2.Model.SecurityGroupRuleRequest
$sgrr.IpProtocol = "tcp"
$sgrr.FromPort = 80
$sgrr.ToPort = 80
$sgrr.CidrIpv4 = "0.0.0.0/0"
$sgr = New-Object -TypeName Amazon.EC2.Model.SecurityGroupRuleUpdate
$sgr.SecurityGroupRuleId = "sgr-09ed298024EXAMPLE"
$sgr.SecurityGroupRule = $sgrr
Edit-EC2SecurityGroupRule  `
    -GroupId sg-1234567890abcdef0 `
    -SecurityGroupRule $sgr
```

Amazon EC2 보안 그룹 삭제

Amazon EC2 인스턴스에 사용할 보안 그룹 생성을 완료하면 이를 삭제할 수 있습니다.

요구 사항

- 보안 그룹은 인스턴스 또는 네트워크 인터페이스와 연결할 수 없습니다.
- 다른 보안 그룹의 규칙에서 보안 그룹을 참조할 수 없습니다.

Console

보안 그룹을 삭제하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. (선택 사항) 보안 그룹이 인스턴스와 연결되어 있지 않은지 확인하려면 다음을 수행합니다.
   a. 탐색 창에서 Security Groups를 선택합니다.
   b. 삭제할 보안 그룹의 ID를 복사합니다.
   c. 탐색 창에서 인스턴스를 선택합니다.
   d. 검색 창에서 보안 그룹 ID가 동일함 필터를 추가하고 보안 그룹의 ID를 붙여넣습니다. 결과가 없는 경우 보안 그룹이 인스턴스와 연결되지 않은 것입니다. 그렇지 않으면 보안 그룹을 삭제하기 전에 연결을 해제해야 합니다.
3. 탐색 창에서 Security Groups를 선택합니다.
4. 보안 그룹을 선택한 다음 작업, 보안 그룹 삭제를 선택합니다.
5. 보안 그룹을 둘 이상 선택하면 확인 메시지가 표시됩니다. 일부 보안 그룹을 삭제할 수 없는 경우 각 보안 그룹의 상태가 표시되어 삭제 여부를 나타냅니다. 삭제를 확인하려면 삭제를 누릅니다.
6. 삭제를 선택합니다.

AWS CLI

보안 그룹을 삭제하려면

다음 delete-security-group 명령을 사용합니다.

```
aws ec2 delete-security-group --group-id sg-1234567890abcdef0
```

PowerShell

보안 그룹을 삭제하려면

Remove-EC2SecurityGroup cmdlet을 사용합니다.

```
Remove-EC2SecurityGroup -GroupId sg-1234567890abcdef0
```

Amazon EC2 보안 그룹 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. 이 방법을 사용하면 보안 그룹의 상태가 유지됩니다. 인바운드 트래픽에 대한 응답은 아웃바운드 보안 그룹 규칙에 관계없이 인스턴스에서 나가도록 허용되며 반대의 경우도 마찬가지입니다.

예를 들어 홈 컴퓨터에서 인스턴스에 대해 netcat 또는 유사한 명령과 같은 명령을 시작하고, 인바운드 보안 그룹 규칙에서 ICMP 트래픽을 허용한다고 가정하겠습니다. 연결에 대한 정보(포트 정보 포함)가 추적됩니다. 명령에 대한 인스턴스의 응답 트래픽은 새로운 요청이 아니라 설정된 연결로 추적되며, 아웃바운드 보안 그룹 규칙이 아웃바운드 ICMP 트래픽을 제한하더라도 인스턴스에서 나가도록 허용됩니다.

TCP, UDP 또는 ICMP 이외의 프로토콜에 대해서는 IP 주소와 프로토콜 번호만 추적됩니다. 인스턴스에서 다른 호스트로 트래픽이 전송되고, 600초 이내에 동일한 트래픽 유형이 호스트에서 인스턴스로 전송되는 경우 인스턴스에 대한 보안 그룹에서는 인바운드 보안 그룹 규칙과 관계없이 해당 트래픽이 수락됩니다. 해당 트래픽이 원본 트래픽에 대한 응답 트래픽으로 간주되어 보안 그룹에서 수락됩니다.

보안 그룹 규칙을 변경하면 추적된 연결이 즉시 중단되지 않습니다. 기존 연결 시간이 초과할 때까지 보안 그룹에서 패킷이 계속 허용됩니다. 트래픽이 즉시 중단되거나 추적 상태와 관계없이 모든 트래픽에 방화벽 규칙이 적용되도록 서브넷의 네트워크 ACL을 사용할 수 있습니다. 네트워크 ACL은 상태 비저장이므로 자동으로 응답 트래픽을 허용하지 않습니다. 어느 방향이든 트래픽이 차단되는 네트워크 ACL을 추가하면 기존 연결이 끊어집니다. 자세한 내용은 Amazon VPC 사용 설명서의 네트워크 ACL 섹션을 참조하세요.

> **Note**
>
> 보안 그룹은 'VPC+2 IP 주소'(Amazon Route 53 개발자 안내서의 Amazon Route 53 Resolver란 무엇인가요? 참조) 또는 'AmazonProvidedDNS'(Amazon Virtual Private Cloud 사용 설명서의 DHCP 옵션 세트로 작업 참조)라고도 하는 Route 53 Resolver에서 들어오고 나가는 DNS 트래픽에 영향을 미치지 않습니다. Route 53 Resolver를 통해 DNS 요청을 필터링하려

면 Route 53 Resolver DNS 방화벽을 활성화하면 됩니다(Amazon Route 53 개발자 안내서의 Route 53 Resolver DNS 방화벽 참조).

추적되지 않는 연결

모든 트래픽 흐름이 추적되지는 않습니다. 모든 트래픽(0.0.0.0/0 또는 ::/0)에 대한 TCP 또는 UDP 흐름이 보안 그룹 규칙에서 허용되고 포트(0-65535)에 대한 모든 응답 트래픽(0.0.0.0/0 또는 ::/0)이 허용되는 해당 규칙이 다른 방향에 있는 경우에는 자동으로 추적되는 연결의 일부가 아니면 트래픽 흐름이 추적되지 않습니다. 추적되지 않는 흐름에 대한 응답 트래픽은 추적 정보가 아니라 응답 트래픽이 허용되는 인바운드 또는 아웃바운드 규칙에 따라 허용됩니다.

흐름을 허용하는 규칙이 제거 또는 수정될 경우 추적되지 않는 트래픽 흐름이 즉시 중단됩니다. 예를 들어 열린 (0.0.0.0/0) 아웃바운드 규칙이 있고 인스턴스에 대한 모든 (0.0.0.0/0) 인바운드 SSH(TCP 포트 22) 트래픽을 허용하는 규칙을 제거하거나 연결이 더 이상 허용되지 않도록 수정할 경우 인스턴스에 대한 기존 SSH 연결이 즉시 삭제됩니다. 이전에 연결이 추적되지 않았으므로 변경으로 인해 연결이 끊어집니다. 반면에 처음에 SSH 연결이 허용되는(즉, 연결이 추적되었음) 더 좁은 범위의 인바운드 규칙이 있지만 현재 SSH 클라이언트의 주소에서 새 연결이 더는 허용되지 않도록 해당 규칙을 변경하는 경우 기존 SSH 연결은 추적되므로 중단되지 않습니다.

자동으로 추적되는 연결

다음을 통해 이루어진 연결은 보안 그룹 구성에 추적하도록 별도로 설정되어 있지 않더라도 자동으로 추적됩니다.

- 외부 전용 인터넷 게이트웨이
- Global Accelerator 액셀러레이터
- NAT 게이트웨이
- Network Firewall 방화벽 엔드포인트
- Network Load Balancers
- AWS PrivateLink(인터페이스 VPC 엔드포인트)
- AWS Lambda(Hyperplane 탄력적 네트워크 인터페이스)

연결 추적 허용량

Amazon EC2는 인스턴스당 추적할 수 있는 최대 연결 수를 정의합니다. 최대값에 도달하면 새 연결을 설정할 수 없기 때문에 전송하거나 수신하는 패킷이 삭제됩니다. 이 경우 패킷을 전송하고 수신하는 애

플리케이션이 제대로 통신할 수 없습니다. `conntrack_allowance_available` 네트워크 성능 지표를 사용하여 해당 인스턴스 유형에 대해 여전히 사용 가능한 추적된 연결 수를 확인합니다.

인스턴스의 네트워크 트래픽이 추적할 수 있는 최대 연결 수를 초과하여 패킷이 삭제되었는지 여부를 확인하려면 `conntrack_allowance_exceeded` 네트워크 성능 지표를 사용합니다. 자세한 내용은 EC2 인스턴스의 ENA 설정에 대한 네트워크 성능 모니터링 섹션을 참조하세요.

탄력적 로드 밸런싱을 통해 인스턴스당 추적할 수 있는 최대 연결 수를 초과할 경우 로드 밸런서에 등록된 인스턴스의 수 또는 로드 밸런서에 등록된 인스턴스의 크기를 조정하는 것이 좋습니다.

연결 추적 성능 고려 사항

트래픽이 한 네트워크 인터페이스를 통해 인스턴스로 수신되고 다른 네트워크 인터페이스를 통해 송신되는 비대칭 라우팅을 사용하면 흐름을 추적하는 경우 인스턴스에서 달성할 수 있는 최고 성능을 줄일 수 있습니다.

보안 그룹에 대한 연결 추적이 활성화된 경우 최고의 성능을 유지하려면 다음 구성을 사용하는 것이 좋습니다.

- 가능하면 비대칭 라우팅 토폴로지를 사용하지 마세요.
- 필터링에 보안 그룹 대신 네트워크 ACL을 사용합니다.
- 연결 추적 기능이 있는 보안 그룹을 사용해야 하는 경우 유휴 연결 추적 제한 시간을 가능한 짧게 구성합니다. 유휴 연결 추적 제한 시간에 대한 자세한 내용은 다음 섹션을 참조하세요.

Nitro 시스템의 성능 조정에 대한 자세한 내용은 성능 튜닝을 위한 Nitro 시스템 고려 사항 섹션을 참조하세요.

유휴 연결 추적 제한 시간

보안 그룹은 설정된 각 연결을 추적하여 반환 패킷이 예상대로 전달되는지 확인합니다. 인스턴스당 추적할 수 있는 최대 연결 수가 있습니다. 유휴 상태로 유지되는 연결은 연결 추적 소진으로 이어져 연결이 추적되지 않고 패킷이 삭제될 수 있습니다. 탄력적 네트워크 인터페이스에서 유휴 연결 추적에 대한 제한 시간을 설정할 수 있습니다.

> **Note**
>
> 이 기능은 Nitro 기반 인스턴스에서만 사용할 수 있습니다.

구성 가능한 세 가지 제한 시간이 있습니다.

- TCP 설정 제한 시간: 설정된 상태의 유휴 TCP 연결에 대한 제한 시간(초)입니다. 최솟값: 60초. 최댓값: 43만 2,000초(5일). 기본값: 43만 2,000초. 권장값: 43만 2,000초 미만.
- UDP 제한 시간: 단일 방향 또는 단일 요청-응답 트랜잭션의 트래픽만 확인한 유휴 UDP 흐름의 제한 시간(초)입니다. 최솟값: 30초. 최댓값: 60초. 기본값: 30초.
- UDP 스트림 제한 시간: 둘 이상의 요청-응답 트랜잭션을 확인한 스트림으로 분류된 유휴 UDP 흐름의 제한 시간(초)입니다. 최솟값: 60초. 최댓값: 180초(3분) 기본값: 180초

다음과 같은 경우에 기본 제한 시간을 수정해야 할 수 있습니다.

- ???Amazon EC2 네트워크 성능 지표를 사용하여 추적된 연결을 모니터링하는 경우 conntrack_allowance_exceeded 및 conntrack_allowance_available 지표를 사용하면 삭제된 패킷과 추적된 연결 사용률을 모니터링하여 스케일 업 또는 아웃 작업을 통해 EC2 인스턴스 용량을 사전에 관리하여 패킷을 삭제하기 전에 네트워크 연결 수요를 충족할 수 있습니다. EC2 인스턴스에서 conntrack_allowance_exceeded 삭제가 관찰되는 경우 부적절한 클라이언트 또는 네트워크 미들 박스로 인해 발생하는 오래된 TCP/UDP 세션을 설명하기 위해 더 낮은 TCP 설정 제한 시간을 설정하는 것이 도움이 될 수 있습니다.
- 일반적으로 로드 밸런서 또는 방화벽에는 60~90분 범위의 TCP 설정 유휴 제한 시간이 있습니다. 네트워크 방화벽과 같은 어플라이언스에서 매우 많은 수(10만 개 이상)의 연결을 처리할 것으로 예상되는 워크로드를 실행하는 경우 EC2 네트워크 인터페이스에서 유사한 제한 시간을 구성하는 것이 좋습니다.
- 비대칭 라우팅 토폴로지를 사용하는 워크로드를 실행하는 경우 TCP 설정 유휴 제한 시간을 60초로 구성하는 것이 좋습니다.
- DNS, SIP, SNMP, Syslog, Radius 및 UDP를 주로 사용하여 요청을 처리하는 기타 서비스와 같이 연결 수가 많은 워크로드를 실행하는 경우 'UDP-Stream' 제한 시간을 60초로 설정하면 기존 용량의 규모/성능이 향상되고 그레이 페일(gray failure)이 발생하지 않습니다.
- 네트워크 로드 밸런서(NLB)와 Elastic Load Balancer(ELB)를 통한 TCP/UDP 연결의 경우 모든 연결이 추적됩니다. TCP 흐름의 유휴 제한 시간 값은 350초이고 UDP 흐름은 120초이며 인터페이스 수준 제한 시간 값에 따라 다릅니다. ELB/NLB의 기본값보다 제한 시간에 대한 유연성을 더 높이기 위해 네트워크 인터페이스 수준에서 제한 시간을 구성할 수 있습니다.

다음 작업을 수행할 때 연결 추적 제한 시간을 구성할 수 있는 옵션이 있습니다.

- 네트워크 인터페이스 생성
- 네트워크 인터페이스 속성 수정
- EC2 인스턴스 시작

- EC2 인스턴스 시작 템플릿 생성

예제

다음 예제의 보안 그룹에는 TCP 및 ICMP 트래픽이 허용되는 인바운드 규칙과 모든 아웃 바운드 트래픽이 허용되는 아웃 바운드 규칙이 있습니다.

인바운드

프로토콜 유형	포트 번호	소스
TCP	22 (SSH)	203.0.113.1/32
TCP	80(HTTP)	0.0.0.0/0
TCP	80(HTTP)	::/0
ICMP	모두	0.0.0.0/0

아웃바운드

프로토콜 유형	포트 번호	대상
모두	모두	0.0.0.0/0
모두	모두	::/0

인스턴스 또는 네트워크 인터페이스에 대한 직접 네트워크 연결에서는 추적 동작이 다음과 같습니다.

- 인바운드 규칙에서 모든 IP 주소(0.0.0.0/0)가 아니라 203.0.113.1/32의 트래픽만 허용되므로 포트 22(SSH)의 인바운드 및 아웃바운드 TCP 트래픽이 추적됩니다.
- 인바운드 및 아웃바운드 규칙에서 모든 TCP 트래픽이 허용되므로 포트 80(HTTP)의 인바운드 및 아웃바운드 TCP 트래픽이 추적되지 않습니다.
- ICMP 트래픽은 항상 추적됩니다.

IPv4 트래픽에 대한 아웃바운드 규칙을 제거하는 경우 포트 80(HTTP)의 트래픽을 포함하여 모든 인바운드 및 아웃바운드 IPv4 트래픽이 추적됩니다. IPv6 트래픽에 대한 아웃바운드 규칙을 제거하는 경우 IPv6 트래픽에 동일하게 적용됩니다.

다양한 사용 사례에 대한 보안 그룹 규칙

보안 그룹을 생성하고 보안 그룹과 연결된 인스턴스의 역할을 반영하는 규칙을 추가합니다. 예를 들어 웹 서버로 구성된 인스턴스에는 인바운드 HTTP 및 HTTPS 액세스를 허용하는 보안 그룹 규칙이 필요합니다. 마찬가지로 데이터베이스 인스턴스에는 데이터베이스 유형에 대한 액세스(예: MySQL의 경우 포트 3306을 통한 액세스)를 허용하는 규칙이 필요합니다.

다음은 특정한 종류의 액세스에 대해 보안 그룹에 추가할 수 있는 규칙의 종류를 예로 든 것입니다.

예시

- 웹 서버 규칙
- 데이터베이스 서버 규칙
- 컴퓨터에서 인스턴스 연결에 대한 규칙
- 보안 그룹이 동일한 인스턴스에서 인스턴스에 대한 연결 규칙
- Ping/ICMP 규칙
- DNS 서버 규칙
- Amazon EFS 규칙
- Elastic Load Balancing 규칙

지침은 보안 그룹 생성 및 the section called “보안 그룹 규칙 구성” 섹션을 참조하세요.

웹 서버 규칙

다음 인바운드 규칙에서는 임의의 IP 주소로부터 HTTP 및 HTTPS 액세스를 허용합니다. VPC가 IPv6 용으로 활성화되면 IPv6 주소에서 인바운드 HTTP 및 HTTPS 트래픽을 제어하기 위한 규칙을 추가할 수 있습니다.

프로토콜 유형	프로토콜 번호	포트	소스 IP	참고
TCP	6	80(HTTP)	0.0.0.0/0	임의의 IPv4 주소에서 인바운드 HTTP 액세스를 허용함
TCP	6	443(HTTPS)	0.0.0.0/0	임의의 IPv4 주소에서 인바운드 HTTPS 액세스를 허용함

프로토콜 유형	프로토콜 번호	포트	소스 IP	참고
TCP	6	80(HTTP)	::/0	임의의 IPv6 주소에서 인바운드 HTTP 액세스를 허용함
TCP	6	443(HTTPS)	::/0	임의의 IPv6 주소에서 인바운드 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 DB 인스턴스에 대한 자세한 내용은 Amazon RDS 사용 설명서를 참조하세요.

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)

프로토콜 유형	프로토콜 번호	포트	참고
TCP	6	1521(Oracle)	Oracle 데이터베이스 액세스를 위한 기본 포트(예: Amazon RDS 인스턴스에서)

선택적으로 데이터베이스 서버의 아웃바운드 트래픽을 제한할 수 있습니다. 예를 들어 소프트웨어 업데이트를 위해 인터넷 액세스를 허용하지만 다른 모든 종류의 트래픽은 제한할 수 있습니다. 먼저 모든 아웃바운드 트래픽을 허용하는 기본 아웃바운드 규칙을 제거해야 합니다.

프로토콜 유형	프로토콜 번호	포트	목적지 IP	참고
TCP	6	80(HTTP)	0.0.0.0/0	임의의 IPv4 주소에 대한 아웃바운드 HTTP 액세스를 허용함
TCP	6	443(HTTPS)	0.0.0.0/0	임의의 IPv4 주소에 대한 아웃바운드 HTTPS 액세스를 허용함
TCP	6	80(HTTP)	::/0	(IPv6 사용 VPC만 해당) 임의의 IPv6 주소에 대한 아웃바운드 HTTP 액세스를 허용함
TCP	6	443(HTTPS)	::/0	(IPv6 사용 VPC만 해당) 임의의 IPv6 주소에 대한 아웃바운드 HTTPS 액세스를 허용함

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. IPv6를 위해

프로토콜 유형	프로토콜 번호	포트	소스 IP
			VPC가 활성화되어 있고 인스턴스에 IPv6 주소가 있는 경우 IPv6 주소 또는 범위를 입력할 수 있습니다.
TCP	6	3389(RDP)	컴퓨터의 퍼블릭 IPv4 주소 또는 로컬 네트워크의 IP 주소 범위. IPv6를 위해 VPC가 활성화되어 있고 인스턴스에 IPv6 주소가 있는 경우 IPv6 주소 또는 범위를 입력할 수 있습니다.

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.

> **Note**
>
> 미들박스 어플라이언스를 통해 서로 다른 서브넷에 있는 두 인스턴스 간의 트래픽을 전달하도록 경로를 구성하는 경우 두 인스턴스에 대한 보안 그룹이 인스턴스 간에 트래픽이 흐르도록 허용해야 합니다. 각 인스턴스의 보안 그룹은 다른 인스턴스의 프라이빗 IP 주소 또는 다른 인스턴스가 포함된 서브넷의 CIDR 범위를 소스로 참조해야 합니다. 다른 인스턴스의 보안 그룹을 소스로 참조하면 인스턴스 간에 트래픽이 흐를 수 없습니다.

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 또는 다른 인스턴스가 포함된 서브넷의 CIDR 범위(참고 사항 참조).

Ping/ICMP 규칙

ping 명령은 ICMP 트래픽의 한 유형입니다. 인스턴스를 ping하려면 다음과 같은 인바운드 ICMP 규칙 중 하나를 추가해야 합니다.

유형	프로토콜	소스		
사용자 지정 ICMP - IPv4	에코 요청	컴퓨터의 퍼블릭 IPv4 주소, 특정 IPv4 주소 또는 위치와 관계없는 IPv4 또는 IPv6 주소입니다.		
모든 ICMP - IPv4	IPv4 ICMP(1)	컴퓨터의 퍼블릭 IPv4 주소, 특정 IPv4 주소 또는 위치와 관계없는 IPv4 또는 IPv6 주소입니다.		

ping6 명령을 사용하여 인스턴스에 대한 IPv6 주소를 ping하려면 다음 인바운드 ICMPv6 규칙을 추가해야 합니다.

유형	프로토콜	소스		
모든 ICMP - IPv6	IPv6 ICMP(58)	컴퓨터의 IPv6 주소, 특정 IPv4 주소 또는 위치와 관계없는 IPv4 또는 IPv6 주소		

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 밸런서에 EC2 인스턴스를 등록하는 경우 로드 밸런서에 연결된 보안 그룹은 인스턴스와의 통신을 허용해야 합니다. 자세한 내용은 Elastic Load Balancing 설명서에서 다음을 참조하세요.

- Application Load Balancer 보안 그룹
- Network Load Balancer의 보안 그룹
- Classic Load Balancer 보안 그룹 구성

Amazon EC2 인스턴스의 NitroTPM

NitroTPM(Nitro Trusted Platform Module)은 AWS Nitro System에서 제공하는 가상 디바이스로서 TPM 2.0 사양을 준수합니다. 인스턴스를 인증하는 데 사용되는 아티팩트(예: 암호, 인증서 또는 암호화 키)를 안전하게 저장합니다. NitroTPM은 키를 생성하여 암호화 기능(예: 해시, 서명, 암호화 및 암호 해독)에 사용할 수 있습니다.

NitroTPM은 부트로더와 운영 체제가 모든 부팅 바이너리의 암호화 해시를 생성하여 NitroTPM 내부 플랫폼 구성 레지스터(PCR)의 이전 값과 결합하는 프로세스인 측정된 부팅을 제공합니다. 측정된 부팅을 사용하면 NitroTPM에서 서명된 PCR 값을 가져와서 이를 사용하여 원격 엔터티에 인스턴스 부팅 소프트웨어의 무결성을 입증할 수 있습니다. 이를 원격 증명이라고 합니다.

NitroTPM을 사용하면 키 및 암호에 특정 PCR 값으로 태그를 지정할 수 있으므로 PCR 값 즉, 인스턴스 무결성이 변경되면 절대 액세스할 수 없습니다. 이러한 특수한 형태의 조건부 액세스 권한을 밀봉 및 밀봉 해제라고 합니다. BitLocker와 같은 운영 체제 기술은 NitroTPM을 사용하여 드라이브 암호 해독 키를 밀봉합니다. 따라서 운영 체제가 올바르게 부팅되어 정상 작동이 확인된 상태일 경우에만 드라이브의 암호를 해독할 수 있습니다.

NitroTPM을 사용하려면 NitroTPM 지원을 위해 구성된 Amazon Machine Image(AMI)를 선택한 다음 AMI를 사용하여 Nitro 기반 인스턴스를 시작해야 합니다. Amazon의 사전 빌드된 AMI 중 하나를 선택하거나 직접 생성할 수 있습니다.

요금

NitroTPM 사용에 대한 추가 비용은 없습니다. 사용하는 기본 리소스에 대해서만 비용을 지불합니다.

내용

- Amazon EC2 인스턴스에서 NitroTPM을 사용하기 위한 요구 사항
- NitroTPM용 Linux AMI 활성화

- AMI가 NitroTPM에 대해 활성화되어 있는지 확인
- Amazon EC2 인스턴스에서 NitroTPM 활성화 또는 중지
- Amazon EC2 인스턴스가 NitroTPM에 대해 활성화되어 있는지 확인
- EC2 인스턴스의 퍼블릭 인증 키 검색

Amazon EC2 인스턴스에서 NitroTPM을 사용하기 위한 요구 사항

NitroTPM이 활성화된 인스턴스를 시작하려면 다음 요구 사항이 충족되어야 합니다.

주제

- AMI
- 인스턴스 타입
- 고려 사항

AMI

AMI에서 NitroTPM이 활성화되어 있어야 합니다.

Linux AMI

사전 구성된 AMI는 없습니다. 자체 AMI를 구성해야 합니다. 자세한 내용은 NitroTPM용 Linux AMI 활성화 섹션을 참조하세요.

Windows AMI

Microsoft 키를 사용하여 NitroTPM과 UEFI 보안 부팅을 위해 사전 구성된 AWS Windows AMI를 찾으려면 AWS Windows AMIs Reference의 Find Windows Server AMIs configured with NitroTPM and UEFI Secure Boot를 참조하세요.

> **Note**
>
> 운영 체제 - AMI에는 TPM 2.0 CRB(Command Response Buffer) 드라이버가 있는 운영 체제가 포함되어야 합니다. 대부분의 최신 운영 체제에는 TPM 2.0 CRB 드라이버가 포함되어 있습니다.
> UEFI 부팅 모드 - AMI는 UEFI 부팅 모드에 맞게 구성되어야 합니다. 자세한 내용은 Amazon EC2 인스턴스의 UEFI 보안 부팅 섹션을 참조하세요.

인스턴스 타입

다음 가상화 인스턴스 유형 중 하나를 사용해야 합니다.

- 범용: M5, M5a, M5ad, M5d, M5dn, M5n, M5zn, M6a, M6g, M6gd, M6i, M6id, M6idn, M6in, M7a, M7g, M7gd, M7i, M7i-flex, M8g, M8gd, T3, T3a, T4g
- 컴퓨팅 최적화: C5, C5a, C5ad, C5d, C5n, C6a, C6g, C6gd, C6gn, C6i, C6id, C6in, C7a, C7g, C7gd, C7gn, C7i, C7i-flex, C8g, C8gd
- 메모리 최적화: R5, R5a, R5ad, R5b, R5d, R5dn, R5n, R6a, R6g, R6gd, R6i, R6idn, R6in, R6id, R7a, R7g, R7gd, R7i, R7iz, R8g, R8gd, U7i-6tb, U7i-8tb, U7i-12tb, U7in-16tb, U7in-24tb, U7in-32tb, X2idn, X2iedn, X2iezn, X8g, z1d
- 스토리지 최적화: D3, D3en, I3en, I4i, I7i, I7ie, I8g
- 가속 컴퓨팅: F2, G4dn, G5, G6, G6e, Gr6, Inf1, Inf2, P5, P5e, P5en, P6-B200, Trn2, Trn2u
- 고성능 컴퓨팅: Hpc6a, Hpc6id

고려 사항

NitroTPM 사용 시 다음 사항을 고려하세요.

- NitroTPM이 활성화된 AMI를 사용하여 인스턴스를 시작한 후 인스턴스 유형을 변경하려면 선택한 새 인스턴스 유형 역시 NitroTPM을 지원해야 합니다.
- NitroTPM 기반 키로 암호화된 BitLocker 볼륨은 원본 인스턴스에서만 사용할 수 있습니다.
- NitroTPM 상태는 Amazon EC2 콘솔에 표시되지 않습니다.
- NitroTPM 상태는 Amazon EBS 스냅샷에 포함되지 않습니다.
- NitroTPM 상태는 VM Import/Export 이미지에 포함되지 않습니다.
- NitroTPM은 AWS Outposts, 로컬 영역 또는 Wavelength 영역에서 지원되지 않습니다.

NitroTPM용 Linux AMI 활성화

인스턴스에 대해 NitroTPM을 활성화하려면 NitroTPM이 활성화된 AMI를 사용하여 인스턴스를 시작해야 합니다. 등록할 때 NitroTPM 지원 Linux AMI를 구성해야 합니다. NitroTPM 지원은 나중에 구성할 수 없습니다.

NitroTPM 지원을 위해 사전 구성된 Windows AMI 목록은 Amazon EC2 인스턴스에서 NitroTPM을 사용하기 위한 요구 사항 섹션을 참조하세요.

RegisterImage API를 사용하여 NitroTPM이 구성된 AMI를 생성해야 합니다. Amazon EC2 콘솔 또는 VM Import/Export를 사용할 수는 없습니다.

NitroTPM용 Linux AMI 활성화

1. 필요한 Linux AMI를 사용하여 임시 인스턴스를 시작합니다. 콘솔에 있는 인스턴스의 스토리지 탭에서 인스턴스의 루트 볼륨 ID를 찾을 수 있습니다.
2. 인스턴스가 `running` 상태에 도달하면 인스턴스의 루트 볼륨 스냅샷을 생성합니다. 자세한 내용은 EBS 볼륨의 스냅샷 생성을 참조하세요.
3. 생성한 스냅샷을 AMI로 등록합니다. 블록 디바이스 매핑에서 루트 볼륨에 대해 생성한 스냅샷을 지정합니다.

   다음은 예제 register-image 명령입니다. `--tpm-support`에 `v2.0`을 지정합니다. `--boot-mode`에 `uefi`을 지정합니다.

   ```
   aws ec2 register-image \
       --name my-image \
       --boot-mode uefi \
       --architecture x86_64 \
       --root-device-name /dev/xvda \
       --block-device-mappings DeviceName=/dev/
   xvda,Ebs={SnapshotId=snap-0abcdef1234567890} \
       --tpm-support v2.0
   ```

   다음은 Register-EC2Image cmdlet의 예입니다.

   ```
   $block = @{SnapshotId=snap-0abcdef1234567890}
   Register-EC2Image `
       -Name my-image `
       -Architecture "x86_64" `
       -RootDeviceName /dev/xvda `
       -BlockDeviceMapping @{DeviceName="/dev/xvda";Ebs=$block} `
       -BootMode Uefi `
       -TpmSupport V20
   ```
4. 1단계에서 시작한 임시 인스턴스를 종료합니다.

AMI가 NitroTPM에 대해 활성화되어 있는지 확인

인스턴스에 대해 NitroTPM을 활성화하려면 NitroTPM이 활성화된 AMI를 사용하여 인스턴스를 시작해야 합니다. 이미지를 설명하여 NitroTPM이 활성화되어 있는지 확인할 수 있습니다. AMI 소유자인 경우 `tpmSupport` 이미지 속성을 설명할 수 있습니다.

Amazon EC2 콘솔에는 `TpmSupport`가 표시되지 않습니다.

AWS CLI

NitroTPM이 활성화되어 있는지 확인하려면

describe-images 명령을 사용합니다.

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query Images[*].TpmSupport
```

AMI에 대해 NitroTPM이 활성화된 출력이 다음과 같습니다. TPM이 활성화되지 않은 경우 출력이 비어 있습니다.

```
[
    "v2.0"
]
```

또는 AMI 소유자인 경우 describe-image-attribute 명령을 `tpmSupport` 속성과 함께 사용합니다.

```
aws ec2 describe-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --attribute tpmSupport
```

출력의 예시는 다음과 같습니다.

```
{
    "ImageId": "ami-0abcdef1234567890",
    "TpmSupport": {
        "Value": "v2.0"
    }
}
```

NitroTPM이 활성화되어 있는 AMI를 찾으려면

다음 예제는 NitroTPM이 활성화된 상태로 소유한 AMI의 ID를 나열합니다.

```
aws ec2 describe-images \
    --owners self \
    --filters Name=tpm-support,Values=v2.0 \
    --query Images[].ImageId
```

PowerShell

NitroTPM이 활성화되어 있는지 확인하려면

Get-EC2Image cmdlet을 사용합니다.

```
Get-EC2Image `
    -ImageId ami-0abcdef1234567890 | Select TpmSupport
```

AMI에 대해 NitroTPM이 활성화된 출력이 다음과 같습니다. TPM이 활성화되지 않은 경우 출력이 비어 있습니다.

```
TpmSupport
----------
v2.0
```

또는 AMI 소유자인 경우 Get-EC2ImageAttribute cmdlet을 `tpmSupport` 속성과 함께 사용합니다.

```
Get-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute tpmSupport
```

NitroTPM이 활성화되어 있는 AMI를 찾으려면

다음 예제는 NitroTPM이 활성화된 상태로 소유한 AMI의 ID를 나열합니다.

```
Get-EC2Image `
    -Owner self `
    -Filter @{Name="tpm-support; Values="v2.0"} | Select ImageId
```

Amazon EC2 인스턴스에서 NitroTPM 활성화 또는 중지

시작 시에만 NitroTPM에 대해 Amazon EC2 인스턴스를 활성화할 수 있습니다. NitroTPM에 대해 인스턴스를 활성화하면 비활성화할 수 없습니다. 더 이상 NitroTPM을 사용할 필요가 없는 경우 사용을 중지하도록 운영 체제를 구성해야 합니다.

주제

- NitroTPM이 사용 설정된 상태로 인스턴스 시작
- 인스턴스에서 NitroTPM 사용 중지

NitroTPM이 사용 설정된 상태로 인스턴스 시작

필수 구성 요소를 사용하여 인스턴스를 시작하는 경우 NitroTPM이 인스턴스에서 자동으로 활성화됩니다. 시작 시에만 인스턴스에서 NitroTPM을 활성화할 수 있습니다. 인스턴스 시작에 대한 자세한 정보는 Amazon EC2 인스턴스 시작하기 섹션을 참조하세요.

인스턴스에서 NitroTPM 사용 중지

NitroTPM이 활성화된 상태로 인스턴스를 시작하면 인스턴스에 대해 NitroTPM을 사용 중지할 수 없습니다. 그러나 다음 도구를 통해 인스턴스에서 TPM 2.0 디바이스 드라이버를 비활성화하여, NitroTPM 사용을 중지하도록 운영 체제를 구성할 수 있습니다.

- Linux 인스턴스의 경우 tpm-tools를 사용합니다.
- Windows 인스턴스의 경우 TPM 관리 콘솔(tpm.msc)을 사용합니다.

디바이스 드라이버 사용 중지에 대한 자세한 정보는 운영 체제 설명서를 참조하세요.

Amazon EC2 인스턴스가 NitroTPM에 대해 활성화되어 있는지 확인

Amazon EC2 인스턴스에 NitroTPM이 활성화되어 있는지 확인할 수 있습니다. 인스턴스에 NitroTPM 지원이 활성화된 경우 명령이 `"v2.0"`을 반환합니다. 그렇게 하지 않으면 출력에 `TpmSupport` 필드가 없습니다.

Amazon EC2 콘솔에는 `TpmSupport` 필드가 표시되지 않습니다.

AWS CLI

인스턴스가 NitroTPM에 대해 활성화되어 있는지 확인하려면 다음을 수행하세요.

아래와 같이 describe-instances 명령을 사용합니다.

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

PowerShell

인스턴스가 NitroTPM에 대해 활성화되어 있는지 확인하려면 다음을 수행하세요.

Get-EC2Instance cmdlet을 사용합니다.

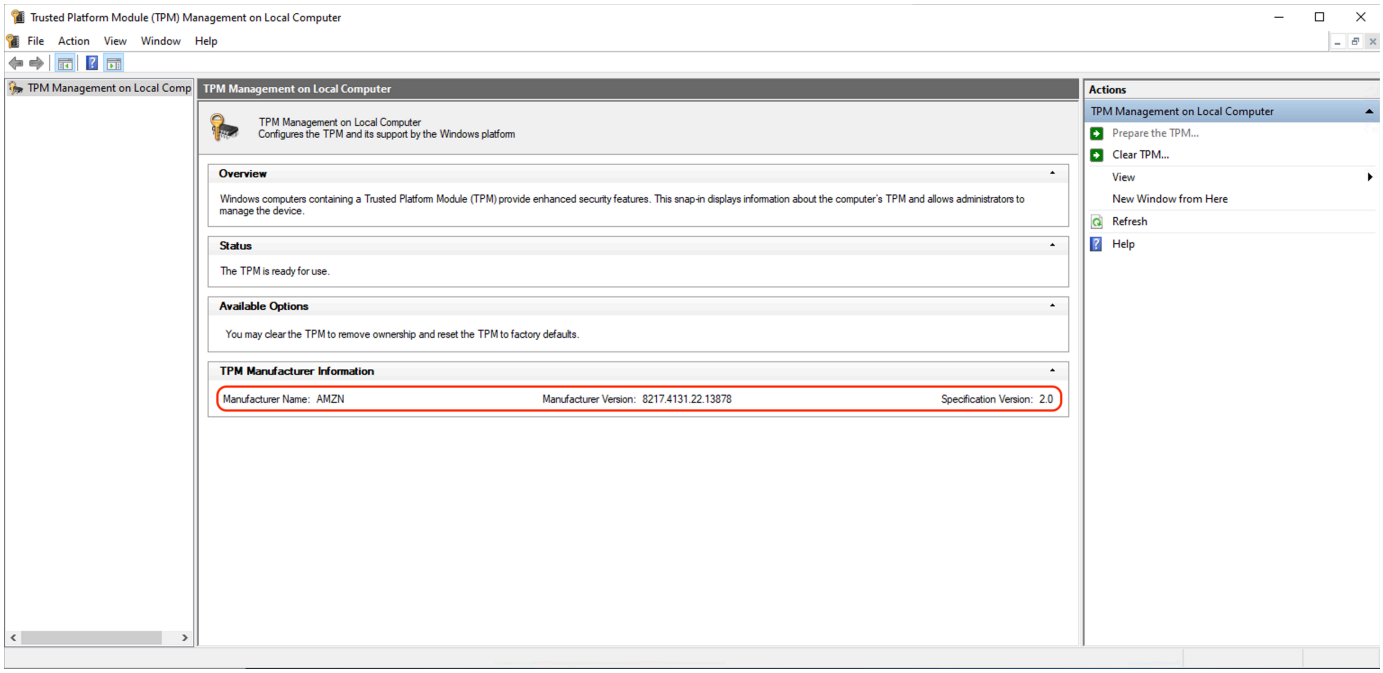
```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances.TpmSupport
```

Windows 인스턴스에서 NitroTPM 액세스 권한 확인

(Windows 인스턴스만 해당) Windows에서 NitroTPM에 액세스할 수 있는지 확인하려면 다음을 수행하세요.

1. EC2 Windows 인스턴스에 연결합니다.
2. 인스턴스에서 tpm.msc 프로그램을 실행합니다.

   로컬 컴퓨터의 TPM 관리 창이 열립니다.
3. TPM 제조업체 정보(TPM Manufacturer Information) 필드를 확인합니다. 인스턴스의 제조업체 이름과 NitroTPM 버전이 포함되어 있습니다.



EC2 인스턴스의 퍼블릭 인증 키 검색

언제든지 인스턴스의 퍼블릭 인증 키를 안전하게 검색할 수 있습니다.

AWS CLI

인스턴스의 퍼블릭 인증 키를 검색하는 방법

get-instance-tpm-ek-pub 명령을 사용합니다.

예시 1

다음 예제는 지정한 인스턴스의 `rsa-2048` 퍼블릭 인증 키(`tpmt` 형식)를 가져옵니다.

```
aws ec2 get-instance-tpm-ek-pub \
    --instance-id i-1234567890abcdef0 \
    --key-format tpmt \
    --key-type rsa-2048
```

다음은 출력 예제입니다.

```
{
    "InstanceId": "i-01234567890abcdef",
    "KeyFormat": "tpmt",
```

```
    "KeyType": "rsa-2048",
    "KeyValue": "AAEACwADALIAIINxl2dEhLEXAMPLEUal1yT9UtduBlILZPKh2hszFGmqAAYAgABDA
    EXAMPLEAAABAOiRd7WmgtdGNoV1h/AxmW+CXExblG8pEUfNm0LOLiYnEXAMPLERqApiFa/UhvEYqN4
    Z7jKMD/usbhsQaAB1gKA5RmzuhSazHQkax7EXAMPLEzDthlS7HNGuYn5eG7qnJndRcakS+iNxT8Hvf
    0S1ZtNuItMs+Yp4SO6aU28MT/JZkOKsXIdMerY3GdWbNQz9AvYbMEXAMPLEPyHfzgVO0QTTJVGdDxh
    vxtXCOu9GYf0crbjEXAMPLEd4YTbWdDdgOKWF9fjzDytJSDhrLAOUctNzHPCd/92l5zEXAMPLEOIFA
    Ss50C0/802c17W2pMSVHvCCa9lYCiAfxH/vYKovAAE="
}
```

예시 2

다음 예제는 지정한 인스턴스의 `rsa-2048` 퍼블릭 인증 키(`der` 형식)를 가져옵니다.

```
aws ec2 get-instance-tpm-ek-pub \
    --instance-id i-1234567890abcdef0 \
    --key-format der \
    --key-type rsa-2048
```

다음은 출력 예제입니다.

```
{
    "InstanceId": "i-1234567890abcdef0",
    "KeyFormat": "der",
    "KeyType": "rsa-2048",
    "KeyValue": "MIIBIjANBgEXAMPLEw0BAQEFAAOCAQ8AMIIBCgKCAQEA6JF3taEXAMPLEXWH8DGZb4
    JcTFuUbykRR82bQs4uJifaKSOv5NGoEXAMPLEG8Rio3hnuMowP+6xuGxBoAHWAoDlGbO6FJrMdEXAMP
    LEnYUHvMO2GVLsc0a5ifl4buqcmd1FxqRL6I3FPwe9/REXAMPLE0yz5inhI7ppTbwxP8lmQ4qxch0x6
    tjcZ1Zs1DP0EXAMPLERUYLQ/Id/OBU7RBNMlUZ0PGG/G1cI670Zh/RytuOdx9iEXAMPLEtZ0N2A4pYX
    1+PMPK0lIOGssA5Ry03Mc8J3/3aXnOD2/ASRQ4gUBKznQLT/zTZEXAMPLEJUe8IJr2VgKIB/Ef+9gqi
    8AAQIDAQAB"
}
```

PowerShell

인스턴스의 퍼블릭 인증 키를 검색하는 방법

Get-EC2InstanceTpmEkPub cmdlet을 사용합니다.

예시 1

다음 예제는 지정한 인스턴스의 `rsa-2048` 퍼블릭 인증 키(`tpmt` 형식)를 가져옵니다.

```
Get-EC2InstanceTpmEkPub `
```

```
    -InstanceId i-1234567890abcdef0 `
    -KeyFormat tpmt `
    -KeyType rsa-2048
```

예시 2

다음 예제는 지정한 인스턴스의 rsa-2048 퍼블릭 인증 키(der 형식)를 가져옵니다.

```
Get-EC2InstanceTpmEkPub `
    -InstanceId i-1234567890abcdef0 `
    -KeyFormat der `
    -KeyType rsa-2048
```

Windows 인스턴스용 Credential Guard

AWS Nitro System은 Amazon Elastic Compute Cloud(Amazon EC2) Windows 인스턴스용 Credential Guard를 지원합니다. Credential Guard는 Windows 커널 보호를 넘어 Windows 사용자 자격 증명 및 코드 무결성 적용과 같은 보안 자산을 보호하기 위해 격리된 환경을 생성할 수 있는 Windows VBS 기능입니다. EC2 Windows 인스턴스를 실행할 때 Credential Guard는 AWS Nitro 시스템을 사용하여 Windows 로그인 자격 증명이 운영 체제 메모리에서 추출되지 않도록 보호합니다.

내용

- 사전 조건
- 지원되는 인스턴스 시작
- 메모리 무결성 비활성화
- Credential Guard 켜기
- Credential Guard가 실행 중인지 확인

사전 조건

Credential Guard를 활용하려면 Windows 인스턴스가 다음 사전 조건을 충족해야 합니다.

Amazon 머신 이미지(AMI)

NitroTPM 및 UEFI Secure Boot를 활성화하도록 AMI를 미리 구성해야 합니다. 지원되는 AMI에 대한 자세한 내용은 the section called “요구 사항” 섹션을 참조하세요.

메모리 무결성

하이퍼바이저 보호 코드 무결성(HVCI) 또는 하이퍼바이저 적용 코드 무결성이라고도 하는 메모리 무결성은 지원되지 않습니다. Credential Guard를 켜기 전에 이 기능이 비활성화되었는지 반드시 확인해야 합니다. 자세한 내용은 메모리 무결성 비활성화 섹션을 참조하세요.

인스턴스 유형

달리 명시하지 않는 한 C5, C5d, C5n, C6i, C6id, C6in, C7i, C7i-flex, M5, M5d, M5dn, M5n, M5zn, M6i, M6id, M6idn, M6in, M7i, M7i-flex, R5, R5b, R5d, R5dn, R5n, R6i, R6id, R6idn, R6in R7i, R7iz, T3 인스턴스 유형은 모든 크기에서 Credential Guard를 지원합니다.

> **Note**
>
> - NitroTPM에는 공통적으로 필요한 몇 가지 인스턴스 유형이 있지만 Credential Guard를 지원하려면 인스턴스 유형이 앞의 인스턴스 유형 중 하나여야 합니다.
> - 다음은 Credential Guard를 지원하지 않습니다.
>   - 베어 메탈 인스턴스.
>   - C7i.48xlarge, M7i.48xlarge, R7i.48xlarge 인스턴스 유형.

인스턴스 유형에 대한 자세한 내용은 Amazon EC2 Intance Types 설명서를 참조하세요.

지원되는 인스턴스 시작

Amazon EC2 콘솔 또는 AWS Command Line Interface(AWS CLI)를 사용하여 Credential Guard를 지원할 수 있는 인스턴스를 시작할 수 있습니다. 인스턴스를 시작하려면 AWS 리전마다 고유한 호환되는 AMI ID가 필요합니다.

> **Tip**
>
> 다음 링크를 사용하여 Amazon EC2 콘솔에서 호환되는 Amazon 제공 AMI를 사용하여 인스턴스를 검색하고 시작할 수 있습니다.
> https://console.aws.amazon.com/ec2/v2/home?#Images:visibility=public-images;v=3;search=:TPM-Windows_Server;ownerAlias=amazon

Console

인스턴스 시작

지원되는 인스턴스 유형과 사전 구성된 Windows AMI를 지정하여 인스턴스를 시작하는 단계를 따르세요.

AWS CLI

인스턴스 시작

run-instances 명령을 사용하여 지원되는 인스턴스 유형과 미리 구성된 Windows AMI를 사용하여 인스턴스를 시작합니다.

```
aws ec2 run-instances \
    --image-id resolve:ssm:/aws/service/ami-windows-latest/TPM-Windows_Server-2022-English-Full-Base \
    --instance-type c6i.large \
    --region us-east-1 \
    --subnet-id subnet-0abcdef1234567890
    --key-name key-name
```

PowerShell

인스턴스 시작

New-EC2Instance 명령을 사용하여 지원되는 인스턴스 유형과 미리 구성된 Windows AMI를 사용하여 인스턴스를 시작합니다.

```
New-EC2Instance `
    -ImageId resolve:ssm:/aws/service/ami-windows-latest/TPM-Windows_Server-2022-English-Full-Base `
    -InstanceType c6i.large `
    -Region us-east-1 `
    -SubnetId subnet-0abcdef1234567890 `
    -KeyName key-name
```

메모리 무결성 비활성화

로컬 그룹 정책 편집기를 사용하여 지원되는 시나리오에서 메모리 무결성을 비활성화할 수 있습니다. 가상화 기반 코드 무결성 보호 아래 각 구성 설정에 대해 다음 지침을 적용할 수 있습니다.

- 잠금 없이 활성화 - 메모리 무결성을 비활성화하려면 설정을 비활성화됨으로 수정하세요.
- UEFI 잠금으로 활성화 – 메모리 무결성이 UEFI 잠금으로 활성화되었습니다. 메모리 무결성을 UEFI 잠금으로 활성화한 후에는 비활성화할 수 없습니다. 메모리 무결성이 비활성화된 새 인스턴스를 생성하고 지원되지 않는 인스턴스를 사용하지 않는 경우 종료하는 것이 좋습니다.

로컬 그룹 정책 편집기로 메모리 무결성을 비활성화하려면

1. RDP를 사용하여 관리자 권한이 있는 사용자 계정으로 인스턴스에 연결합니다. 자세한 내용은 the section called “RDP 클라이언트를 사용하여 연결” 섹션을 참조하세요.
2. 시작 메뉴를 열고 **cmd**를 검색하여 명령 프롬프트를 시작합니다.
3. `gpedit.msc` 명령을 실행하여 로컬 그룹 정책 편집기를 엽니다.
4. 로컬 그룹 정책 편집기에서 컴퓨터 구성, 관리 템플릿, 시스템 및 디바이스 가드를 선택합니다.
5. 가상화 기반 보안 켜기를 선택한 다음 정책 설정 편집을 선택합니다.
6. 가상화 기반 코드 무결성 보호에 대한 설정 드롭다운을 열고 비활성화를 선택한 후 적용을 선택합니다.
7. 인스턴스를 재부팅하여 변경 사항을 적용합니다.

Credential Guard 켜기

지원되는 인스턴스 유형과 호환되는 AMI로 Windows 인스턴스를 시작하고 메모리 무결성이 비활성화되었음을 확인한 후 Credential Guard를 켤 수 있습니다.

> **Important**
>
> 다음 단계를 수행하여 Credential Guard를 켜려면 관리자 권한이 필요합니다.

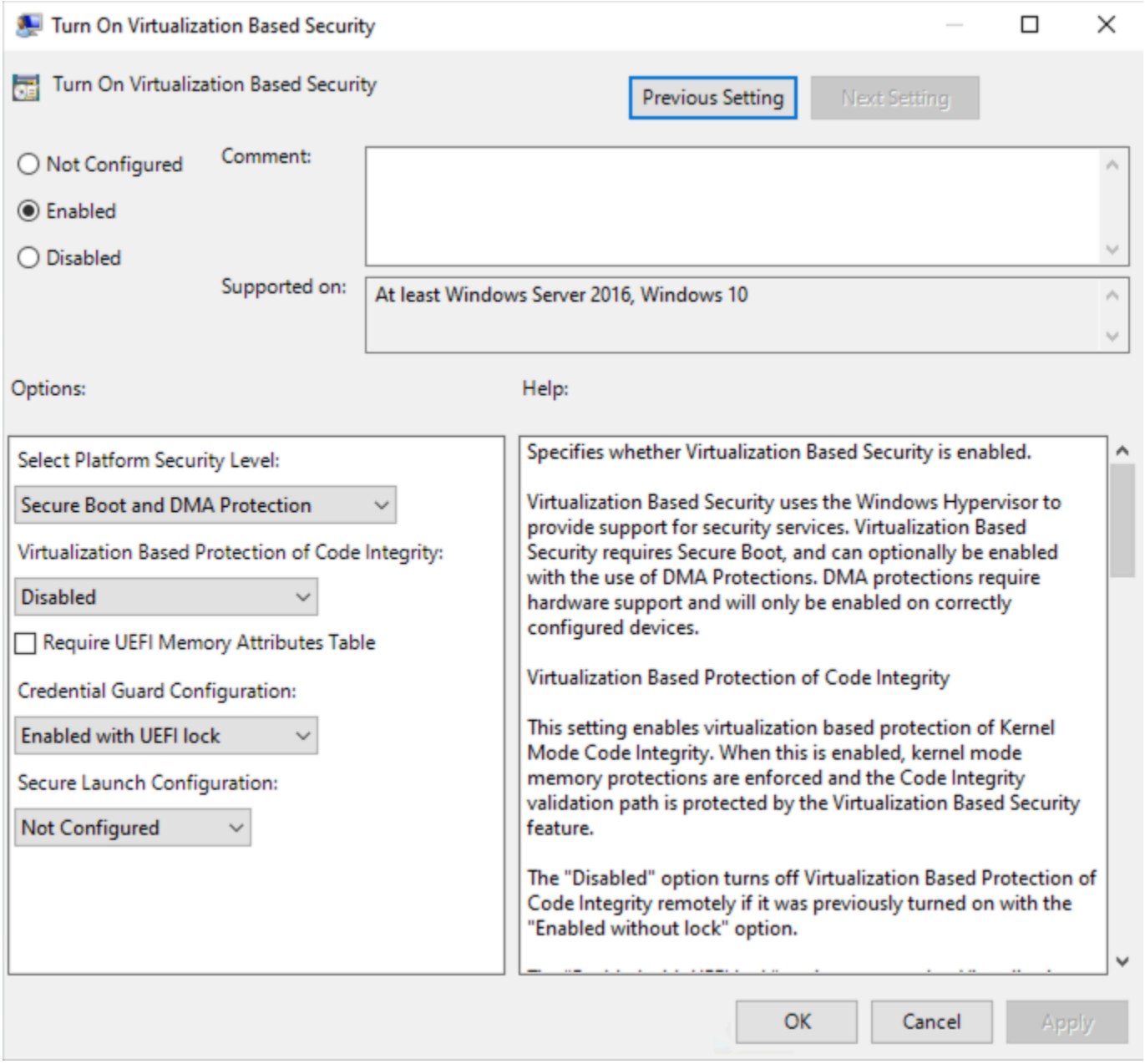
Credential Guard 켜기

1. RDP를 사용하여 관리자 권한이 있는 사용자 계정으로 인스턴스에 연결합니다. 자세한 내용은 the section called “RDP 클라이언트를 사용하여 연결” 섹션을 참조하세요.
2. 시작 메뉴를 열고 **cmd**를 검색하여 명령 프롬프트를 시작합니다.
3. `gpedit.msc` 명령을 실행하여 로컬 그룹 정책 편집기를 엽니다.
4. 로컬 그룹 정책 편집기에서 컴퓨터 구성, 관리 템플릿, 시스템 및 디바이스 가드를 선택합니다.
5. 가상화 기반 보안 켜기를 선택한 다음 정책 설정 편집을 선택합니다.

6. 가상화 기반 보안 켜기 메뉴에서 활성화를 선택합니다.
7. 플랫폼 보안 수준 선택에서 보안 부팅 및 DMA 보호를 선택합니다.
8. Credential Guard 구성에서 UEFI 잠금으로 활성화를 선택합니다.

> **Note**
>
> 나머지 정책 설정은 Credential Guard를 활성화하는 데 필요하지 않으며 구성되지 않음으로 둘 수 있습니다.

다음 이미지에는 앞서 설명한 대로 구성된 VBS 설정이 표시됩니다.



9. 인스턴스를 재부팅하여 설정을 적용합니다.

Credential Guard가 실행 중인지 확인

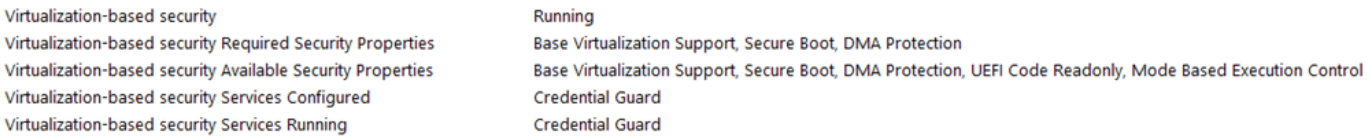
Microsoft 시스템 정보(`Msinfo32.exe`) 도구를 사용하여 Credential Guard가 실행 중인지 확인할 수 있습니다.

> **Important**
>
> 먼저 인스턴스를 재부팅하여 Credential Guard를 활성화하는 데 필요한 정책 설정 적용을 마쳐야 합니다.

Credential Guard가 실행 중인지 확인

1. 원격 데스크톱 프로토콜(RDP)을 사용하여 인스턴스에 연결합니다. 자세한 내용은 the section called “RDP 클라이언트를 사용하여 연결” 섹션을 참조하세요.
2. 인스턴스에 대한 RDP 세션 내에서 시작 메뉴를 열고 **cmd**를 검색하여 명령 프롬프트를 시작합니다.
3. `msinfo32.exe` 명령을 실행하여 시스템 정보를 엽니다.
4. Microsoft 시스템 정보 도구는 VBS 구성의 세부 정보를 나열합니다. 가상화 기반 보안 서비스 옆에서 Credential Guard가 실행 중으로 표시되는지 확인합니다.

   다음 이미지에는 앞서 설명한 대로 VBS가 실행 중임이 표시됩니다.



인터페이스 VPC 엔드포인트를 사용하여 Amazon EC2에 액세스

VPC와 Amazon EC2 API의 리소스 간에 프라이빗 연결을 생성하여 VPC의 보안 태세를 개선할 수 있습니다. 인터넷 게이트웨이, NAT 디바이스, VPN 연결 또는 AWS Direct Connect 연결을 사용하지 않고 VPC에 있는 것처럼 Amazon EC2 API에 액세스할 수 있습니다. VPC의 EC2 인스턴스는 Amazon EC2 API에 액세스하는 데 퍼블릭 IP 주소가 필요하지 않습니다.

자세한 정보는 AWS PrivateLink 가이드의 AWS PrivateLink를 통해 AWS 서비스에 액세스를 참조하세요.

내용

- 인터페이스 VPC 엔드포인트 생성
- 엔드포인트 정책 생성

인터페이스 VPC 엔드포인트 생성

다음 서비스 이름을 사용하여 Amazon EC2에 대한 인터페이스 엔드포인트를 생성합니다.

- com.amazonaws.***region***.ec2 - Amazon EC2 API 작업에 대한 엔드포인트를 생성합니다.

자세한 정보는 AWS PrivateLink 가이드의 인터페이스 VPC 엔드포인트를 사용하여 AWS 서비스에 액세스를 참조하세요.

엔드포인트 정책 생성

엔드포인트 정책은 인터페이스 엔드포인트에 연결할 수 있는 IAM 리소스입니다. 기본 엔드포인트 정책을 사용하면 인터페이스 엔드포인트를 통해 Amazon EC2 API에 대한 전체 액세스를 허용합니다. VPC에서 Amazon EC2 API에 허용되는 액세스를 제어하려면 사용자 지정 엔드포인트 정책을 인터페이스 엔드포인트에 연결합니다.

엔드포인트 정책은 다음 정보를 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 로깅되지 않을 수 있습니다.

자세한 정보는 AWS PrivateLink 가이드의 엔드포인트 정책을 사용하여 서비스에 대한 액세스 제어를 참조하세요.

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": "*",
        "Principal": "*"
    },
    {
        "Action": [
            "ec2:CreateVolume"
        ],
        "Effect": "Deny",
        "Resource": "*",
        "Principal": "*",
        "Condition": {
            "Bool": {
                "ec2:Encrypted": "false"
            }
        }
    },
    {
        "Action": [
            "ec2:RunInstances"
        ],
        "Effect": "Deny",
        "Resource": "*",
        "Principal": "*",
        "Condition": {
            "Bool": {
                "ec2:Encrypted": "false"
            }
        }
    }]
}
```

Amazon EC2 인스턴스의 스토리지 옵션

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.

블록 스토리지

- Amazon EBS - Amazon EBS는 인스턴스와 연결하고 분리할 수 있는 내구성이 뛰어난 블록 스토리지 볼륨을 제공합니다. 여러 EBS 볼륨을 하나의 인스턴스에 연결할 수 있습니다. EBS 볼륨은 연결된 인스턴스의 수명과 독립적으로 유지됩니다. EBS 볼륨을 암호화할 수 있습니다. 데이터의 백업 사본을 유지하기 위해 EBS 볼륨에서 스냅샷을 생성할 수 있습니다. 스냅샷은 Amazon S3에 저장됩니다. 스냅샷에서 EBS 볼륨을 생성할 수 있습니다.
- EC2 인스턴스용 인스턴스 스토어 임시 블록 스토리지 - 인스턴스 스토어는 인스턴스에 임시 블록 스토리지를 제공합니다. 인스턴스 스토어 볼륨의 수, 크기 및 유형은 인스턴스 유형과 인스턴스 크기에 따라 결정됩니다. 인스턴스 스토어에 저장된 데이터는 연관 인스턴스의 수명 기간 동안에0000 유지되고, 해당 인스턴스를 중단하거나 최대 절전 모드로 전환하거나 종료하면 인스턴스 스토어 볼륨의 데이터가 손실됩니다.

객체 스토리지

- 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 사용하여 데이터 및 애플리케이션의 백업 복사본을 저장할 수 있습니다. Amazon EC2는 Amazon S3를 사용하여 EBS 스냅샷과 인스턴스 스토어 지원 AMI를 저장합니다.

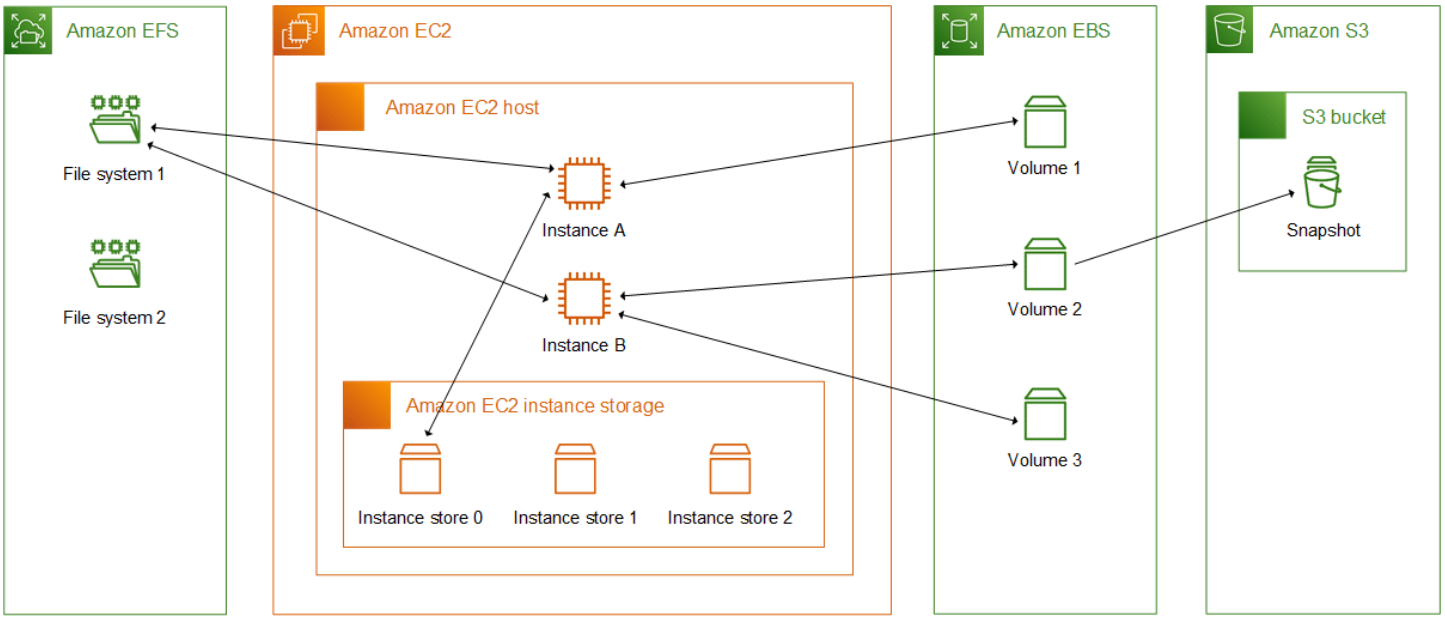
파일 스토리지

- Amazon EFS(Linux 인스턴스 전용) - 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.
- Amazon FSx - Amazon FSx를 사용하면 클라우드에서 기능이 풍부한 고성능 파일 시스템을 시작, 실행 및 확장할 수 있습니다. Amazon FSx는 광범위한 워크로드를 지원하는 완전관리형 서비스입니다. Lustre, NetApp ONTAP, OpenZFS 및 Windows File Server와 같이 널리 사용되는 파일 시스템 중에서 선택할 수 있습니다.

파일 캐싱

- Amazon EC2 인스턴스에서 Amazon File Cache 사용 - Amazon File Cache는 파일 데이터 처리를 위해 AWS에 고성능 임시 캐시를 제공합니다. 캐시는 1밀리초 미만의 지연 시간, 최대 수백 GB/s의 처리량, 최대 수백만 IOPS로 Amazon EC2 컴퓨팅 워크로드에 대한 읽기 및 쓰기 데이터 액세스를 제공합니다.

다음 그림은 이러한 스토리지 옵션과 인스턴스 간의 관계를 보여줍니다.



AWS 스토리지 요금

AWS 요금을 열고 AWS 제품 요금으로 스크롤한 후 스토리지를 선택합니다. 스토리지 제품을 선택하여 해당 요금 페이지를 엽니다.

Amazon EC2 인스턴스의 Amazon EBS 영구 블록 스토리지

Amazon Elastic Block Store(Amazon EBS)에서는 Amazon EC2 인스턴스와 함께 사용할 수 있는 확장 가능한 고성능 블록 스토리지 리소스를 제공합니다. Amazon EBS를 사용해 다음과 같은 블록 스토리지 리소스를 생성하고 관리할 수 있습니다.

- Amazon EBS 볼륨 - Amazon EC2 인스턴스에 연결하는 스토리지 볼륨입니다. 볼륨을 인스턴스에 연결하면 블록 스토리지를 사용할 때와 같은 방식으로 사용할 수 있습니다. 인스턴스는 로컬 드라이브에서와 마찬가지로 볼륨과 상호 작용할 수 있습니다.

- Amazon EBS 스냅샷 - 볼륨 자체와 관계없이 지속되는 Amazon EBS 볼륨의 특정 시점 백업입니다. Amazon EBS 볼륨의 데이터를 백업하는 스냅샷을 생성할 수 있습니다. 그러면 언제든지 해당 스냅샷에서 새 볼륨을 복원할 수 있습니다.

시작 중에 EBS 볼륨을 생성하여 인스턴스에 연결할 수 있으며, 시작 후 언제라도 EBS 볼륨을 생성하여 인스턴스에 연결할 수 있습니다. 또한 볼륨을 분리하거나 인스턴스를 다시 시작하지 않고도 EBS 볼륨의 크기나 성능을 늘릴 수 있습니다.

생성 후 언제라도 EBS 볼륨에서 EBS 스냅샷을 생성할 수 있습니다. EBS 스냅샷을 사용하여 볼륨에 저장된 데이터를 백업할 수 있습니다. 그런 다음에 해당 스냅샷을 사용하여 볼륨을 즉시 복원하거나 AWS 계정, AWS 리전 또는 가용 영역 간에 데이터를 마이그레이션할 수 있습니다. Amazon Data Lifecycle Manager나 AWS Backup을 사용하여 EBS 스냅샷의 생성, 보관 및 삭제를 자동화할 수 있습니다.

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

볼륨 및 스냅샷 작업에 대한 자세한 내용은 Amazon EBS 사용 설명서를 참조하세요.

Amazon EC2 인스턴스의 Amazon EBS 볼륨 한도

인스턴스에 연결할 수 있는 Amazon EBS 볼륨의 최대 수는 인스턴스 유형 및 인스턴스 크기에 따라 달라집니다. 인스턴스의 볼륨 연결 제한을 초과하면 `AttachmentLimitExceeded` 오류와 함께 연결 요청이 실패합니다.

인스턴스에 연결할 볼륨의 수를 고려할 때는 I/O 대역폭 증가 또는 스토리지 용량 증가의 필요성 여부를 고려해야 합니다.

대역폭 및 용량 비교

일관되고 예측 가능한 대역폭이 필요한 사용 사례에서는 Amazon EBS에 최적화된 인스턴스(범용 SSD 볼륨 또는 프로비저닝된 IOPS SSD 볼륨 포함)를 사용합니다. 성능을 극대화하려면 볼륨에 프로비저닝한 IOPS와 인스턴스 유형에 사용 가능한 대역폭을 일치시키세요.

RAID 구성의 경우 볼륨이 8개보다 큰 배열은 I/O 오버헤드가 증가하여 성능이 저하될 수 있습니다. 따라서 개별 애플리케이션의 성능을 테스트한 다음 필요에 따라 조정하세요.

목차

- Nitro 시스템에 구축된 인스턴스의 볼륨 제한

- 전용 EBS 볼륨 제한
- 공유 EBS 볼륨 제한
- Xen 기반 인스턴스의 볼륨 제한
  - Linux 인스턴스
  - Windows 인스턴스

Nitro 시스템에 구축된 인스턴스의 볼륨 제한

Nitro System 기반의 인스턴스에 대한 볼륨 제한은 인스턴스 유형에 따라 다릅니다. 일부 Nitro 인스턴스 유형에는 전용 EBS 볼륨 제한이 있는 반면, 대부분은 공유 볼륨 제한이 있습니다. 각 인스턴스 유형에 대한 볼륨 연결 제한을 보려면 다음을 참조하세요.

- Amazon EBS 사양 - 범용
- Amazon EBS 사양 - 컴퓨팅 최적화
- Amazon EBS 사양 - 메모리 최적화
- Amazon EBS 사양 - 스토리지 최적화
- Amazon EBS 사양 - 가속 컴퓨팅
- Amazon EBS 사양 - 고성능 컴퓨팅
- Amazon EBS 사양 - 이전 세대

전용 EBS 볼륨 제한

다음 Nitro 인스턴스 유형에는 인스턴스 크기에 따라 달라지는 전용 EBS 볼륨 제한이 있습니다. 제한은 다른 디바이스 연결과 공유되지 않습니다. 즉, NVMe 인스턴스 스토어 볼륨 및 네트워크 인터페이스와 같은 연결된 디바이스 수에 관계없이 볼륨 연결 제한까지 EBS 볼륨을 원하는 만큼 연결할 수 있습니다.

- 범용: M7a | M7i | M7i-flex | M8g | M8gd
- 컴퓨팅 최적화: C7a | C7i | C7i-flex | C8g | C8gd
- 메모리 최적화: R7a | R7i | R7iz | R8g | R8gd | U7i-6tb | U7i-8tb | U7i-12tb | U7in-16tb | U7in-24tb | U7in-32tb | U7inh-32tb | X8g
- 스토리지 최적화: I7i | I7ie | I8g
- 가속 컴퓨팅: F2 | G6 | G6e | Gr6 | P4d | P4de | P5 | P5e | P5en | P6-B200 | Trn2 | Trn2u
- 고성능 컴퓨팅: Hpc7a

공유 EBS 볼륨 제한

다른 모든 Nitro 인스턴스 유형(전용 EBS 볼륨 제한에 나열되지 않음)에는 Amazon EBS 볼륨, 네트워크 인터페이스, NVMe 인스턴스 스토어 볼륨 간에 공유되는 볼륨 연결 한도가 있습니다. 해당 제한에서 연결된 네트워크 인터페이스 및 NVMe 인스턴스 스토어 볼륨 수를 제외한 만큼 Amazon EBS 볼륨을 연결할 수 있습니다. 모든 인스턴스에는 네트워크 인터페이스가 하나 이상 있어야 하며 NVMe 인스턴스 스토어 볼륨은 시작 시 자동으로 연결된다는 점에 유의하세요.

Nitro 인스턴스는 대부분 최대 28개의 연결을 지원합니다. 다음 예제에서는 연결할 수 있는 EBS 볼륨 수를 계산하는 방법을 보여줍니다.

예시

- 기본 네트워크 인터페이스만 있는 `m5.xlarge` 인스턴스의 경우 27개의 EBS 볼륨을 연결할 수 있습니다.

  볼륨 28개 - 네트워크 인터페이스 1개 = 27개
- 두 개의 추가 네트워크 인터페이스가 있는 `m5.xlarge` 인스턴스의 경우 25개의 EBS 볼륨을 연결할 수 있습니다.

  볼륨 28개 - 네트워크 인터페이스 3개 = 25개
- 두 개의 추가 네트워크 인터페이스가 있는 `m5d.xlarge` 인스턴스의 경우 24개의 EBS 볼륨을 연결할 수 있습니다.

  볼륨 28개 - 네트워크 인터페이스 3개 - NVMe 인스턴스 스토어 볼륨 1개 = 24개

Xen 기반 인스턴스의 볼륨 제한

T2와 같은 Xen 기반 인스턴스의 볼륨 제한은 운영 체제에 따라 다릅니다.

자세한 내용은 Xen 기반 인스턴스를 참조하세요.

Linux 인스턴스

Xen 기반 Linux 인스턴스에 볼륨을 40개 이상 연결하면 부팅 오류가 발생할 수 있습니다. 이 개수에는 루트 볼륨과 함께 연결된 모든 인스턴스 스토어 볼륨 및 Amazon EBS 볼륨이 포함됩니다.

볼륨이 많은 인스턴스에서 부팅 문제가 발생하는 경우 인스턴스를 중지하고 부팅 프로세스에 필요하지 않은 볼륨을 분리하고 인스턴스를 실행한 다음 인스턴스가 실행된 후 해당 볼륨을 다시 연결하세요.

> **Important**
>
> Xen 기반 Linux 인스턴스에 볼륨을 40개 이상 연결하는 것은 최선의 노력을 기울인 경우에만 제공되며 보장되지 않습니다.

Windows 인스턴스

다음 표는 사용된 드라이버에 따른 Xen 기반 Windows 인스턴스의 볼륨 제한을 보여줍니다. 이러한 개수에는 루트 볼륨과 함께 연결된 모든 인스턴스 스토어 볼륨 및 Amazon EBS 볼륨이 포함됩니다.

드라이버	볼륨 제한
AWS PV	26
Citrix PV	26
Red Hat PV	17

성능 문제가 발생할 수 있으므로 AWS PV 또는 Citrix PV 드라이버를 사용하는 Xen 기반 Windows 인스턴스에 볼륨을 26개 이상 연결하지 않는 것이 좋습니다. 인스턴스에서 사용하는 PV 드라이버를 확인하거나 Red Hat에서 Citrix PV 드라이버로 Windows 인스턴스를 업그레이드하려면 the section called “PV 드라이버 업그레이드”을(를) 참조하세요.

> **Important**
>
> Xen 기반 Windows 인스턴스에 볼륨을 다음 개수 이상 연결하는 것은 최선의 노력을 기울인 경우에만 제공되며 보장되지 않습니다.

디바이스 이름이 볼륨과 어떻게 관련되는지에 대한 자세한 내용은 Amazon EC2 Windows 인스턴스용 볼륨 연결 및 매핑 방법을(를) 참조하세요.

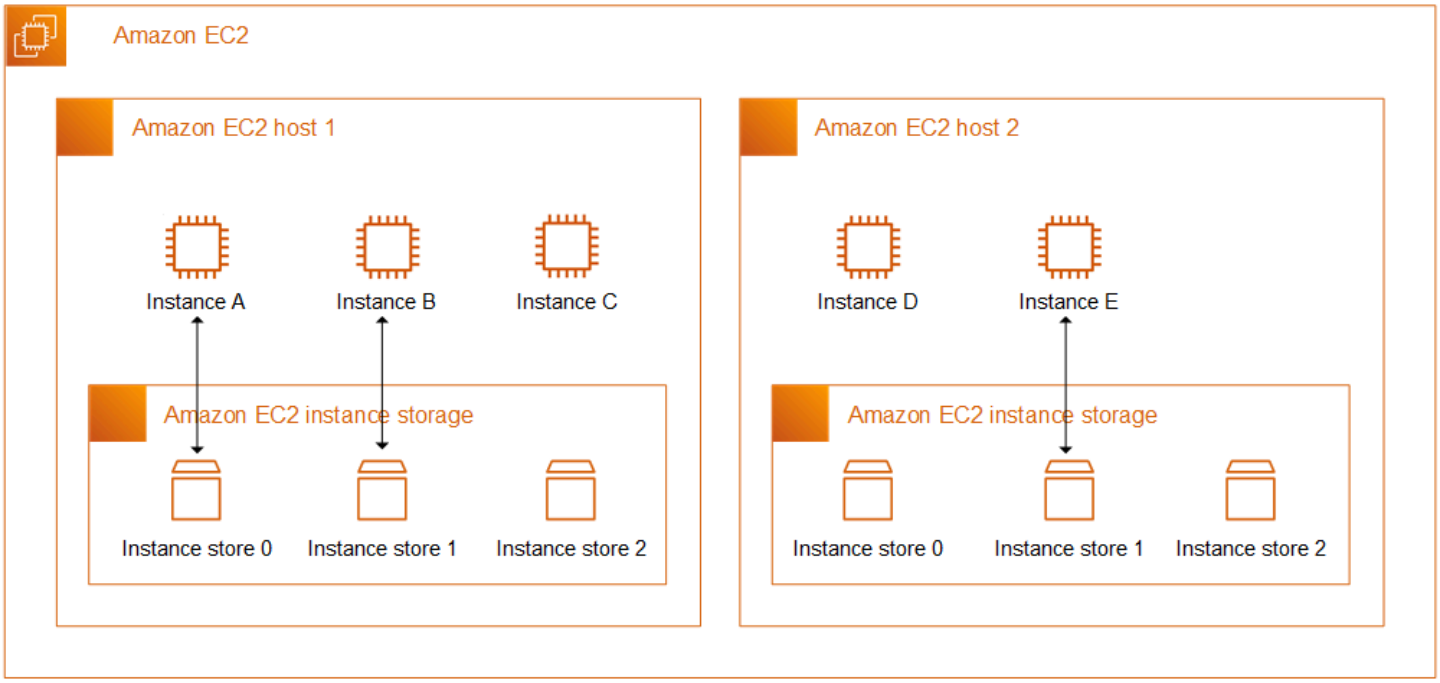
EC2 인스턴스용 인스턴스 스토어 임시 블록 스토리지

인스턴스 스토어는 EC2 인스턴스에 임시 블록 스토리지를 제공합니다. 이 스토리지는 호스트 컴퓨터에 물리적으로 연결된 디스크에서 제공합니다. 인스턴스 스토어는 버퍼, 캐시, 스크래치 데이터, 기타

임시 콘텐츠와 같이 자주 변경되는 정보의 임시 저장에 적합합니다. 또한 로드 밸런싱된 웹 서버 풀과 같은 인스턴스 플릿 전체에 복제하는 임시 데이터를 저장하는 데 사용할 수도 있습니다.

하나 이상의 인스턴스 스토어 볼륨으로 구성된 인스턴스 스토어는 블록 디바이스로 표시됩니다. 인스턴스 스토어의 크기는 물론 사용 가능한 디바이스의 수는 인스턴스 유형과 인스턴스 크기에 따라 다릅니다. 예를 들어 모든 인스턴스 유형이 인스턴스 스토어 볼륨을 제공하는 것은 아닙니다. 자세한 내용은 EC2 인스턴스의 인스턴스 스토어 볼륨 제한 섹션을 참조하세요.

인스턴스 스토어 볼륨의 가상 디바이스에는 `ephemeral0`부터 `ephemeral23`까지 순서대로 가상 디바이스 이름이 지정됩니다. 예를 들어, 인스턴스 스토어 볼륨 1개를 지원하는 인스턴스 유형의 경우 볼륨 1개의 가상 디바이스 이름은 `ephemeral0`입니다. 인스턴스 스토어 볼륨 4개를 지원하는 인스턴스 유형의 경우 볼륨 4개의 가상 디바이스 이름은 `ephemeral0`, `ephemeral1`, `ephemeral2` 및 `ephemeral3`입니다.



인스턴스 스토어 요금

인스턴스에 제공된 인스턴스 스토어 볼륨 사용에 따른 추가 요금은 없습니다. 인스턴스 스토어 볼륨은 인스턴스의 사용 요금의 일부로 포함됩니다.

내용

- Amazon EC2 인스턴스 스토어 볼륨의 데이터 지속성
- EC2 인스턴스의 인스턴스 스토어 볼륨 제한
- EC2 인스턴스용 SSD 인스턴스 스토어 볼륨

- EC2 인스턴스에 인스턴스 스토어 볼륨 추가
- M1 및 C1 EC2 인스턴스의 인스턴스 스토어 스왑 볼륨 활성화
- EC2 인스턴스의 인스턴스 스토어 볼륨 초기화

Amazon EC2 인스턴스 스토어 볼륨의 데이터 지속성

인스턴스 스토어 볼륨은 인스턴스 시작 시에만 연결됩니다. 시작 후에는 인스턴스 스토어 볼륨을 연결할 수 없습니다. 하나의 인스턴스에서 인스턴스 스토어 볼륨을 분리하고 다른 인스턴스에 연결할 수 없습니다.

인스턴스 스토어 볼륨은 연결된 인스턴스의 수명 기간 동안에만 존재합니다. 연결된 인스턴스의 수명 기간이 지난 후에도 유지되도록 인스턴스 스토어 볼륨을 구성할 수 없습니다.

인스턴스를 재부팅해도 인스턴스 스토어 볼륨의 데이터는 유지됩니다. 그러나 인스턴스가 중지 또는 종료되거나 최대 절전 모드로 전환된 경우에는 데이터가 유지되지 않습니다. 인스턴스가 중지 또는 종료되거나 최대 절전 모드로 전환되면 인스턴스 스토어 볼륨의 모든 블록이 암호화된 방식으로 지워집니다.

그러므로 중요한 장기 데이터의 경우 인스턴스 스토어 볼륨에 의존하지 마세요. 인스턴스 수명 기간이 지난 후에도 인스턴스 스토어 볼륨에 저장된 데이터를 유지해야 하는 경우 Amazon EBS 볼륨, Amazon S3 버킷 또는 Amazon EFS 파일 시스템과 같은 보다 영구적인 스토리지에 해당 데이터를 수동으로 복사해야 합니다.

인스턴스의 수명 기간 동안 데이터가 유지되지 않는 몇 가지 이벤트가 있습니다. 다음 표에서는 가상화 인스턴스와 베어 메탈 인스턴스 모두에 대해 특정 이벤트 동안 인스턴스 스토어 볼륨의 데이터가 유지되는지 여부를 보여줍니다.

Event	데이터는 어떻게 되나요?
사용자 시작 인스턴스 수명 주기 이벤트	
인스턴스가 재부팅됨	The data persists
인스턴스가 중지됨	The data does not persist
인스턴스가 최대 절전 모드로 전환됨	The data does not persist
인스턴스가 종료됨	The data does not persist

Event	데이터는 어떻게 되나요?
인스턴스 유형이 변경됨	The data does not persist *
인스턴스에서 EBS 지원 AMI가 생성됨	The data does not persist in the created AMI **
인스턴스에서 인스턴스 스토어 지원 AMI가 생성됨 (Linux instances)	The data persists in the AMI bundle uploaded to Amazon S3 ***
사용자가 시작한 OS 이벤트	
A shutdown is initiated	The data does not persist †
A restart is initiated	The data persists
AWS 예약된 이벤트	
인스턴스 중지	The data does not persist
인스턴스 재부팅	The data persists
시스템 재부팅	The data persists
인스턴스 만료	The data does not persist
계획되지 않은 이벤트	
간소화된 자동 복구	The data does not persist
CloudWatch 작업 기반 복구	The data does not persist
The underlying disk fails	The data on the failed disk does not persist
Power failure	The data persists upon reboot

* 새 인스턴스 유형이 인스턴스 스토어를 지원하는 경우 인스턴스는 새 인스턴스 유형에서 지원하는 인스턴스 스토어 볼륨 수를 가져오지만 데이터는 새 인스턴스로 전송되지 않습니다. 새 인스턴스 유형이 인스턴스 스토어를 지원하지 않는 경우 인스턴스는 인스턴스 스토어 볼륨을 가져오지 않습니다.

** 데이터는 EBS 지원 AMI에 포함되지 않으며 해당 AMI에서 시작된 인스턴스에 연결된 인스턴스 스토어 볼륨에도 포함되지 않습니다.

*** 데이터는 Amazon S3에 업로드되는 AMI 번들에 포함됩니다. 해당 AMI에서 인스턴스를 시작하면 인스턴스는 AMI가 생성될 때 포함된 데이터와 함께 AMI에 번들된 인스턴스 스토어 볼륨을 가져옵니다.

† 종료 방지 및 중지 방지는 인스턴스의 운영 체제를 통해 시작된 종료로 인한 인스턴스 중지 또는 종료로부터 인스턴스를 보호하지 않습니다. 인스턴스 스토어 볼륨에 저장된 데이터는 인스턴스 중지 및 종료 이벤트 모두에서 유지되지 않습니다.

EC2 인스턴스의 인스턴스 스토어 볼륨 제한

인스턴스 스토어 볼륨의 수, 크기 및 유형은 인스턴스 유형에 따라 결정됩니다. M6, C6, R6 등의 일부 인스턴스 유형은 인스턴스 스토어 볼륨을 지원하지 않지만, M5d, C6gd, R6gd 등의 다른 인스턴스 유형은 인스턴스 스토어 볼륨을 지원합니다. 인스턴스 유형에서 지원하는 것보다 많은 인스턴스 스토어 볼륨을 인스턴스에 연결할 수 없습니다. 인스턴스 스토어 볼륨을 지원하는 인스턴스 유형의 경우 인스턴스 스토어 볼륨의 수와 크기는 인스턴스 크기에 따라 다릅니다. 예를 들어 `m5d.large`는 75GB 인스턴스 스토어 볼륨 1개를 지원하고 `m5d.24xlarge`는 900GB 인스턴스 스토어 볼륨 4개를 지원합니다.

NVMe 인스턴스 스토어 볼륨이 있는 인스턴스 유형의 경우 지원되는 모든 인스턴스 스토어 볼륨이 시작 시 인스턴스에 자동으로 연결됩니다. C1, C3, M1, M2, M3, R3, D2, H1, I2, X1, X1e 등의 비NVMe 인스턴스 스토어 볼륨이 있는 인스턴스 유형의 경우 시작 시 연결할 인스턴스 스토어 볼륨에 대한 블록 디바이스 매핑을 수동으로 지정해야 합니다. 그런 다음 인스턴스가 시작된 후 연결된 인스턴스 스토어 볼륨을 사용하려면 먼저 볼륨을 포맷하고 탑재해야 합니다. 인스턴스를 시작한 후에는 인스턴스 스토어 볼륨을 연결할 수 없습니다.

일부 인스턴스 유형은 NVMe 또는 SATA 기반 솔리드 스테이트 드라이브(SSD)를 사용하고 다른 인스턴스 유형은 SATA 기반 하드 디스크 드라이브(HDD)를 사용합니다. SSD는 매우 짧은 지연 시간과 뛰어난 임의 I/O 성능을 제공하지만 인스턴스 종료 시 데이터를 유지할 필요가 없거나 내결함성 아키텍처를 활용할 수 있습니다. 자세한 내용은 EC2 인스턴스용 SSD 인스턴스 스토어 볼륨 섹션을 참조하세요.

NVMe 인스턴스 스토어 볼륨 및 일부 HDD 인스턴스 스토어 볼륨의 데이터는 저장 시 암호화됩니다. 자세한 내용은 Amazon EC2의 데이터 보호 섹션을 참조하세요.

NVMe 인스턴스 스토어 볼륨

Amazon EC2 인스턴스 유형 안내서에는 지원되는 각 인스턴스 유형에서 사용 가능한 인스턴스 스토어 볼륨의 수량, 크기, 유형 및 성능 최적화가 나와 있습니다. 자세한 내용은 다음 자료를 참조하세요.

- 인스턴스 스토어 사양 - 범용
- 인스턴스 스토어 사양 - 컴퓨팅 최적화

- 인스턴스 스토어 사양 - 메모리 최적화
- 인스턴스 스토어 사양 - 스토리지 최적화
- 인스턴스 스토어 사양 - 가속 컴퓨팅
- 인스턴스 스토어 사양 - 고성능 컴퓨팅
- 인스턴스 스토어 사양 - 이전 세대

Console

인스턴스 저장소 볼륨 정보를 검색하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스 유형을 선택합니다.
3. 로컬 인스턴스 스토리지 = true 필터를 추가합니다. 스토리지 열은 인스턴스 유형에 대한 인스턴스 스토리지의 총 크기를 나타냅니다.
4. (선택 사항) 기본 설정 아이콘을 클릭한 다음에 스토리지 디스크 수를 켭니다. 이 열은 인스턴스 저장소 볼륨 수를 나타냅니다.
5. (선택 사항) 필터를 추가하여 관심 있는 특정 인스턴스 유형에 범위를 추가로 지정합니다.

AWS CLI

인스턴스 저장소 볼륨 정보를 검색하는 방법

describe-instance-types 명령을 사용합니다. 다음 예제에서는 인스턴스 저장소 볼륨이 있는 모든 R6i 인스턴스 패밀리의 각 인스턴스 유형에 대한 인스턴스 스토리지의 총 크기가 표시됩니다.

```
aws ec2 describe-instance-types \
    --filters "Name=instance-type,Values=r6i*" "Name=instance-storage-
supported,Values=true" \
    --query "InstanceTypes[].[InstanceType, InstanceStorageInfo.TotalSizeInGB]" \
    --output table
```

출력의 예시는 다음과 같습니다.

```
----------------------------
|  DescribeInstanceTypes   |
+----------------+---------+
|  r6id.16xlarge  |  3800  |
|  r6idn.16xlarge |  3800  |
```

```
|  r6idn.8xlarge  |  1900  |
|  r6id.2xlarge   |  474   |
|  r6idn.xlarge   |  237   |
|  r6id.12xlarge  |  2850  |
|  r6idn.2xlarge  |  474   |
|  r6id.xlarge    |  237   |
|  r6idn.24xlarge |  5700  |
|  r6id.4xlarge   |  950   |
|  r6id.32xlarge  |  7600  |
|  r6id.24xlarge  |  5700  |
|  r6idn.large    |  118   |
|  r6idn.4xlarge  |  950   |
|  r6id.large     |  118   |
|  r6id.8xlarge   |  1900  |
|  r6idn.32xlarge |  7600  |
|  r6idn.metal    |  7600  |
|  r6id.metal     |  7600  |
|  r6idn.12xlarge |  2850  |
+-----------------+--------+
```

인스턴스 유형에 대한 전체 인스턴스 스토리지 세부 정보를 가져오는 방법

describe-instance-types 명령을 사용합니다.

```
aws ec2 describe-instance-types \
    --filters "Name=instance-type,Values=r6id.16xlarge" \
    --query "InstanceTypes[].InstanceStorageInfo"
```

예제 출력에서는 이 인스턴스 유형에 인스턴스 스토리지가 총 3,800GB인 2개의 1,900GB NVMe SSD 볼륨이 있다는 것을 보여줍니다.

```
[
    {
        "TotalSizeInGB": 3800,
        "Disks": [
            {
                "SizeInGB": 1900,
                "Count": 2,
                "Type": "ssd"
            }
        ],
        "NvmeSupport": "required",
        "EncryptionSupport": "required"
```

```
    }
]
```

PowerShell

인스턴스 저장소 볼륨 정보를 검색하는 방법

Get-EC2InstanceType cmdlet을 사용합니다. 다음 예제에서는 인스턴스 저장소 볼륨이 있는 모든 R6i 인스턴스 패밀리의 각 인스턴스 유형에 대한 인스턴스 스토리지의 총 크기가 표시됩니다.

```
(Get-EC2InstanceType -Filter @{Name="instance-type";
 Values="r6i*"},@{Name="instance-storage-supported"; Values="true"})
 | Format-Table @{Name="InstanceType";Expression={$_.InstanceType}},
 @{Name="TotalSize";Expression={$_.InstanceStorageInfo.TotalSizeInGB}}
```

출력의 예시는 다음과 같습니다.

```
InstanceType   TotalSize
------------   ---------
r6idn.16xlarge      3800
r6id.16xlarge       3800
r6id.xlarge          237
r6idn.8xlarge       1900
r6idn.2xlarge        474
r6id.12xlarge       2850
r6idn.xlarge         237
r6id.2xlarge         474
r6id.4xlarge         950
r6idn.24xlarge      5700
r6id.32xlarge       7600
r6id.24xlarge       5700
r6idn.large          118
r6id.large           118
r6idn.4xlarge        950
r6id.8xlarge        1900
r6id.metal          7600
r6idn.32xlarge      7600
r6idn.metal         7600
r6idn.12xlarge      2850
```

인스턴스 유형에 대한 전체 인스턴스 스토리지 세부 정보를 가져오는 방법

Get-EC2InstanceType cmdlet을 사용합니다. 출력은 JSON 형식으로 변환됩니다.

```
(Get-EC2InstanceType -Filter @{Name="instance-type";
 Values="r6id.16xlarge"}).InstanceStorageInfo | ConvertTo-Json
```

예제 출력에서는 이 인스턴스 유형에 인스턴스 스토리지가 총 3,800GB인 2개의 1,900GB NVMe SSD 볼륨이 있다는 것을 보여줍니다.

```
{
  "Disks": [
    {
      "Count": 2,
      "SizeInGB": 1900,
      "Type": "ssd"
    }
  ],
  "EncryptionSupport": {
    "Value": "required"
  },
  "NvmeSupport": {
    "Value": "required"
  },
  "TotalSizeInGB": 3800
}
```

EC2 인스턴스용 SSD 인스턴스 스토어 볼륨

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 볼륨 추가 섹션을 참조하세요.

NVMe SSD 볼륨

일부 인스턴스는 NVMe(Non-Volatile Memory Express) SSD(Solid State Drive) 인스턴스 스토어 볼륨을 제공합니다. 인스턴스 유형별로 지원되는 인스턴스 스토어 볼륨 유형에 대한 자세한 내용은 EC2 인스턴스의 인스턴스 스토어 볼륨 제한 섹션을 참조하세요.

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.

Linux 인스턴스

NVMe 볼륨에 액세스하려면 NVMe 드라이버가 설치되어 있어야 합니다. 다음 AMI가 이 요구 사항을 충족합니다.

- AL2023년
- Amazon Linux 2
- Amazon Linux AMI 2018.03 이상
- Ubuntu 14.04 이상(`linux-aws` 커널 포함)

> **Note**
>
> Ubuntu 18.04 이상(`linux-aws` 커널 포함)이 필요한 AWS Graviton 기반 인스턴스 유형

- 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

- Bottlerocket

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 확인하세요.

- 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 `nvme-cli` 명령을 사용하여 리포지토리에서 yum install 패키지를 설치할 수 있습니다. 지원되는 다른 Linux 버전에서는 `nvme-cli` 패키지가 이미지에 제공되지 않은 경우 다운로드할 수 있습니다.

Windows 인스턴스

다음 운영 체제용 최신 AWS Windows AMI에는 성능 향상을 위해 NVMe 블록 디바이스로 표시되는 SSD 인스턴스 스토어 볼륨과 상호 작용하는 데 사용되는 AWS NVMe 드라이버가 포함되어 있습니다.

- Windows Server 2025
- Windows Server 2022
- Windows Server 2019
- Windows Server 2016
- Windows Server 2012 R2

인스턴스에 연결한 후 디스크 관리자에서 NVMe 볼륨이 보이는지 확인할 수 있습니다. 작업 표시줄에서 마우스 오른쪽 버튼을 클릭하여 Windows 로고에 대한 컨텍스트 메뉴를 열고 디스크 관리(Disk Management)를 선택합니다.

Amazon에서 제공하는 AWS Windows AMI에는 AWS NVMe 드라이버가 포함되어 있습니다. 최신 AWS Windows AMI를 사용하고 있지 않은 경우 AWS현재 NVMe 드라이버를 설치할 수 있습니다.

비NVMe SSD 볼륨

C3, I2, M3, R3, X1 인스턴스는 비NVMe SSD를 사용하여 높은 랜덤 I/O 성능을 제공하는 인스턴스 스토어 볼륨을 지원합니다. 인스턴스 유형별로 지원되는 인스턴스 스토어 볼륨에 대한 자세한 내용은 EC2 인스턴스의 인스턴스 스토어 볼륨 제한 섹션을 참조하세요.

SSD 기반 인스턴스 스토어 볼륨 I/O 성능

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 오버프로비저닝을 위한 예약 공간을 가지고 있지 않습니다. 쓰기 작업 증폭의 영향 감쇄를 위해 최소한 볼륨의 10%를 파티션 처리되지 않은 상태로 두어서 SSD 컨트롤러가 이를 오버프로비저닝에 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 제공되므로 쓰기 작업 증폭을 줄이고 성능을 향상시킬 수 있습니다. 자세한 내용은 인스턴스 스토어 볼륨 TRIM 지원 섹션을 참조하세요.

인스턴스 스토어 볼륨 TRIM 지원

일부 인스턴스 유형은 TRIM이 포함된 SSD 볼륨을 지원합니다. 자세한 내용은 EC2 인스턴스의 인스턴스 스토어 볼륨 제한 섹션을 참조하세요.

> **Note**
>
> (Windows 인스턴스만 해당) Windows Server 2012 R2를 실행하는 인스턴스는 AWS PV 드라이버 버전 7.3.0부터 TRIM을 지원합니다. 이전 버전의 Windows Server를 실행하는 인스턴스는 TRIM을 지원하지 않습니다.

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.

(Windows 인스턴스) 초기 포맷 중에 TRIM 지원을 일시적으로 비활성화하려면 `fsutil behavior set DisableDeleteNotify 1` 명령을 사용합니다. 포맷이 완료되면 `fsutil behavior set DisableDeleteNotify 0`을 사용하여 TRIM 지원을 다시 활성화합니다.

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. Linux 인스턴스에서는 `fstrim` 명령을 사용하여 정기 TRIM을 사용하도록 설정합니다. Windows 인스턴스에서는 `fsutil behavior set DisableDeleteNotify 0` 명령을 사용하여 정상적인 작업 중에 TRIM 지원이 활성화되었는지 확인합니다.

EC2 인스턴스에 인스턴스 스토어 볼륨 추가

NVMe 인스턴스 스토어 볼륨이 있는 인스턴스 유형의 경우 지원되는 모든 인스턴스 스토어 볼륨이 시작 시 인스턴스에 자동으로 연결됩니다. 자동으로 열거되고 인스턴스 시작 시 디바이스 이름이 할당됩니다.

C1, C3, M1, M2, M3, R3, D2, H1, I2, X1, X1e 등의 비NVMe 인스턴스 스토어 볼륨이 있는 인스턴스 유형의 경우 시작 시 연결할 인스턴스 스토어 볼륨에 대한 블록 디바이스 매핑을 수동으로 지정해야 합니다. 블록 디바이스 매핑은 인스턴스 시작 요청 또는 인스턴스 시작에 사용되는 AMI에서 지정할 수 있습니다. 블록 디바이스 매핑에는 디바이스 이름과 매핑된 볼륨이 포함됩니다. 자세한 내용은 Amazon EC2 인스턴스의 볼륨에 대한 블록 디바이스 매핑 섹션을 참조하세요.

> **Important**
>
> 시작할 때만 인스턴스에 인스턴스 스토어 볼륨을 연결할 수 있습니다. 인스턴스를 실행한 이후에는 인스턴스 스토어 볼륨을 연결할 수 없습니다.

인스턴스를 실행한 후에는 인스턴스에 대한 인스턴스 스토어 볼륨이 사용하기에 앞서 포맷되고 마운트되었는지 확인해야 합니다. 인스턴스 스토어 지원 인스턴스의 루트 볼륨은 기본적으로 마운트됩니다.

루트 볼륨에 대한 고려 사항

블록 디바이스 매핑은 항상 인스턴스에 대한 루트 볼륨을 지정합니다. 루트 볼륨은 항상 자동으로 마운트됩니다.

Linux 인스턴스 – 루트 볼륨은 Amazon EBS 볼륨 또는 인스턴스 스토어 볼륨 중 하나입니다. 루트 볼륨에 대한 인스턴스 스토어 볼륨이 있는 인스턴스의 경우, 볼륨의 크기는 AMI에 따라 다르지만 최대 크기는 10GB입니다. 자세한 내용은 루트 디바이스 유형 섹션을 참조하세요.

Windows 인스턴스 - 루트 볼륨은 Amazon EBS 볼륨이어야 합니다. 루트 볼륨에는 인스턴스 스토어가 지원되지 않습니다.

내용

- Amazon EC2 AMI에 인스턴스 스토어 볼륨 추가
- 시작 중 EC2 인스턴스에 인스턴스 스토어 볼륨 추가
- EC2 인스턴스에서 인스턴스 스토어 볼륨을 사용할 수 있게 설정

Amazon EC2 AMI에 인스턴스 스토어 볼륨 추가

인스턴스 스토어 볼륨을 포함하는 블록 디바이스 매핑으로 AMI를 생성할 수 있습니다.

인스턴스 스토어 볼륨 블록 디바이스 매핑을 지정하는 AMI를 사용하여 비NVMe 인스턴스 스토어 볼륨을 지원하는 인스턴스를 시작하는 경우 인스턴스에 인스턴스 스토어 볼륨이 포함됩니다. AMI의 인스턴스 스토어 볼륨 블록 디바이스 매핑 수가 인스턴스에 사용 가능한 인스턴스 스토어 볼륨 수를 초과하면 추가 인스턴스 스토어 볼륨 블록 디바이스 매핑이 무시됩니다.

인스턴스 스토어 볼륨 블록 디바이스 매핑을 지정하는 AMI를 사용하여 NVMe 인스턴스 스토어 볼륨을 지원하는 인스턴스를 시작하면 인스턴스 스토어 볼륨 블록 디바이스 매핑이 무시됩니다. NVMe 인스

턴스 스토어 볼륨을 지원하는 인스턴스는 인스턴스 시작 요청 및 AMI에 지정된 블록 디바이스 매핑에 관계없이 지원되는 모든 인스턴스 스토어 볼륨을 가져옵니다. 이러한 볼륨의 디바이스 매핑은 운영 체제가 볼륨을 열거하는 순서에 따라 다릅니다.

고려 사항

- 사용할 수 있는 인스턴스 스토어 볼륨의 개수는 인스턴스 유형에 따라 다릅니다. 자세한 내용은 the section called “NVMe 인스턴스 스토어 볼륨” 섹션을 참조하세요.
- 각 블록 디바이스의 디바이스 이름을 지정해야 합니다. 자세한 내용은 Amazon EC2 인스턴스의 볼륨 디바이스 이름 섹션을 참조하세요.
- 인스턴스를 실행할 때 AMI 블록 디바이스 매핑에서 지정된 비NVMe 인스턴스 스토어 볼륨을 생략하거나 인스턴스 스토어 볼륨을 추가할 수 있습니다.
- M3 인스턴스의 경우 AMI가 아닌 인스턴스의 블록 디바이스 매핑에 인스턴스 스토어 볼륨을 지정합니다. Amazon EC2가 AMI의 인스턴스 스토어 볼륨 블록 디바이스 매핑을 무시할 수 있습니다.

Console

Amazon EBS 지원 AMI에 인스턴스 저장소 볼륨을 추가하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택하고 인스턴스를 선택합니다.
3. 작업(Actions), 이미지 및 템플릿(Image and templates), 이미지 생성(Create image)을 차례로 선택합니다.
4. [이미지 생성(Create image)] 페이지에서 이미지 이름 및 설명을 입력합니다.
5. 추가할 각 인스턴스 스토어 볼륨에서 [볼륨 추가(Add volume)]를 선택한 다음 [볼륨 유형(Volume type)]에서 인스턴스 스토어 볼륨을 선택하고 [디바이스(Device)]에서 디바이스 이름을 선택합니다.
6. 이미지 생성을 선택합니다.

AWS CLI

AMI에 인스턴스 저장소 볼륨을 추가하는 방법

create-image 명령을 `--block-device-mappings` 옵션과 함께 사용하여 EBS 지원 AMI에 대한 블록 디바이스 매핑을 지정합니다. register-image 명령을 `--block-device-mappings` 옵션과 함께 사용하여 인스턴스 저장소 지원 AMI에 대한 블록 디바이스 매핑을 지정합니다.

```
--block-device-mappings file://mapping.json
```

다음 블록 디바이스 매핑에서는 2개의 인스턴스 저장소 볼륨이 추가됩니다.

```
[
    {
        "DeviceName": "/dev/sdc",
        "VirtualName": "ephemeral0"
    },
    {
        "DeviceName": "/dev/sdd",
        "VirtualName": "ephemeral1"
    }
]
```

PowerShell

AMI에 인스턴스 저장소 볼륨을 추가하는 방법

New-EC2Image cmdlet을 `-BlockDeviceMapping` 파라미터와 함께 사용하여 EBS 지원 AMI에 대한 블록 디바이스 매핑을 지정합니다. Register-EC2Image을 `-BlockDeviceMapping` 파라미터와 함께 사용하여 인스턴스 저장소 지원 AMI에 대한 블록 디바이스 매핑을 지정합니다.

```
-BlockDeviceMapping $bdm
```

다음 블록 디바이스 매핑에서는 2개의 인스턴스 저장소 볼륨이 추가됩니다.

```
$bdm = @()

$sdc = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$sdc.DeviceName = "/dev/sdc"
$sdc.VirtualName = "ephemeral0"
$bdm += $sdc

$sdd = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$sdd.DeviceName = "/dev/sdd"
$sdd.VirtualName = "ephemeral1"
$bdm += $sdd
```

시작 중 EC2 인스턴스에 인스턴스 스토어 볼륨 추가

C1, C3, M1, M2, M3, R3, D2, H1, I2, X1, X1e 등의 비NVMe 인스턴스 스토어 볼륨이 있는 인스턴스 유형을 시작할 때 연결할 인스턴스 스토어 볼륨에 대한 블록 디바이스 매핑을 지정해야 합니다. 블록 디바이스 매핑은 인스턴스 시작 요청 또는 인스턴스 시작에 사용되는 AMI에서 지정해야 합니다.

AMI에 인스턴스 스토어 볼륨에 대한 블록 디바이스 매핑이 포함된 경우 AMI에 포함된 것보다 더 많은 인스턴스 스토어 볼륨이 필요한 경우가 아니면 인스턴스 시작 요청에서 블록 디바이스 매핑을 지정할 필요가 없습니다.

AMI에 인스턴스 스토어 볼륨에 대한 블록 디바이스 매핑이 포함되지 않은 경우 인스턴스 시작 요청에서 블록 디바이스 매핑을 지정해야 합니다.

NVMe 인스턴스 스토어 볼륨이 있는 인스턴스 유형의 경우 지원되는 모든 인스턴스 스토어 볼륨이 시작 시 인스턴스에 자동으로 연결됩니다.

고려 사항

- 사용할 수 있는 인스턴스 스토어 볼륨의 개수는 인스턴스 유형에 따라 다릅니다. 자세한 내용은 the section called “NVMe 인스턴스 스토어 볼륨” 섹션을 참조하세요.
- 각 블록 디바이스의 디바이스 이름을 지정해야 합니다. 자세한 내용은 Amazon EC2 인스턴스의 볼륨 디바이스 이름 섹션을 참조하세요.
- M3 인스턴스의 경우, 인스턴스의 블록 디바이스 매핑에서 지정하지 않더라도 인스턴스 스토어 볼륨을 받을 수 있습니다.

Console

인스턴스 시작 요청에 블록 디바이스 매핑을 지정하려면 다음을 수행하세요.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 대시보드에서 인스턴스 시작을 선택합니다.
3. 애플리케이션 및 OS 이미지(Application and OS Images) 섹션에서 사용할 AMI를 선택합니다.
4. 스토리지 구성 섹션의 인스턴스 스토어 볼륨 섹션에는 인스턴스에 연결할 수 있는 인스턴스 스토어 볼륨이 나열되어 있습니다.
5. 연결할 각 인스턴스 스토어 볼륨에 대해 디바이스 이름에서 사용할 디바이스 이름을 선택합니다.
6. 필요에 따라 나머지 인스턴스 설정을 구성한 다음 인스턴스 시작을 선택합니다.

AWS CLI

인스턴스 시작 요청에 블록 디바이스 매핑을 지정하려면 다음을 수행하세요.

run-instances 명령을 `--block-device-mappings` 옵션과 함께 사용합니다.

```
--block-device-mappings file://mapping.json
```

다음 블록 디바이스 매핑에서는 2개의 인스턴스 저장소 볼륨이 추가됩니다.

```
[
    {
        "DeviceName": "/dev/sdc",
        "VirtualName": "ephemeral0"
    },
    {
        "DeviceName": "/dev/sdd",
        "VirtualName": "ephemeral1"
    }
]
```

PowerShell

인스턴스 시작 요청에 블록 디바이스 매핑을 지정하려면 다음을 수행하세요.

New-EC2Instance cmdlet을 `-BlockDeviceMapping` 옵션과 함께 사용합니다.

```
-BlockDeviceMapping $bdm
```

다음 블록 디바이스 매핑에서는 2개의 인스턴스 저장소 볼륨이 추가됩니다.

```
$bdm = @()

$sdc = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$sdc.DeviceName = "/dev/sdc"
$sdc.VirtualName = "ephemeral0"
$bdm += $sdc

$sdd = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$sdd.DeviceName = "/dev/sdd"
$sdd.VirtualName = "ephemeral1"
$bdm += $sdd
```

EC2 인스턴스에서 인스턴스 스토어 볼륨을 사용할 수 있게 설정

연결된 인스턴스 스토어 볼륨으로 인스턴스를 시작한 후에 먼저 볼륨을 마운트해야 볼륨에 액세스할 수 있습니다.

Linux 인스턴스

인스턴스를 시작한 후 볼륨을 원하는 파일 시스템으로 포맷할 수 있습니다.

Linux에서 인스턴스 스토어 볼륨을 사용 가능하게 만들려면

1. SSH 클라이언트를 사용하여 인스턴스에 연결합니다. 자세한 내용은 SSH를 사용하여 Linux 인스턴스에 연결 섹션을 참조하세요.
2. `df -h` 명령을 사용하여 포맷되고 마운트된 볼륨을 봅니다.

```
$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        3.8G   72K  3.8G   1% /dev
tmpfs           3.8G     0  3.8G   0% /dev/shm
/dev/nvme0n1p1  7.9G  1.2G  6.6G  15% /
```

3. `lsblk`를 사용하여 시작 시에 매핑되지 않았지만 포맷되고 마운트된 볼륨을 봅니다.

```
$ lsblk
NAME          MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
nvme0n1       259:1    0    8G  0 disk
##nvme0n1p1   259:2    0    8G  0 part /
##nvme0n1p128 259:3    0    1M  0 part
nvme1n1       259:0    0 69.9G  0 disk
```

4. 매핑된 인스턴스 스토어 볼륨만 포맷하고 마운트하려면 다음을 수행합니다.
   a. `mkfs` 명령을 사용하여 디바이스에서 파일 시스템을 생성합니다.

   ```
   $ sudo mkfs -t xfs /dev/nvme1n1
   ```

   b. `mkdir` 명령을 사용하여 디바이스를 마운트할 디렉터리를 생성합니다.

   ```
   $ sudo mkdir /data
   ```

   c. `mount` 명령을 사용하여 새로 생성한 디렉터리에 디바이스를 마운트합니다.

```
$ sudo mount /dev/nvme1n1 /data
```

Windows 인스턴스

Windows 인스턴스의 경우 인스턴스 스토어 볼륨을 NTFS 파일 시스템으로 다시 포맷합니다.

Windows 디스크 관리를 사용하여 인스턴스 스토어 볼륨을 볼 수 있습니다. 자세한 내용은 비 NVMe 디스크 나열 섹션을 참조하세요.

인스턴스 스토어 볼륨을 수동으로 탑재하려면

1. 시작을 선택하고 컴퓨터 관리를 입력한 다음 Enter 키를 누릅니다.
2. 왼쪽 패널에서 디스크 관리를 선택합니다.
3. 볼륨을 초기화하라는 메시지가 나타나면 초기화할 볼륨을 선택하고, 사용 사례에 따라 필요한 파티션 유형을 선택한 다음 확인을 선택합니다.
4. 볼륨 목록에서 탑재할 볼륨을 마우스 오른쪽 버튼으로 클릭한 다음 새 단순 볼륨을 선택합니다.
5. 마법사에서 다음을 선택합니다.
6. 볼륨 크기 지정 화면에서 다음을 선택하여 최대 볼륨 크기를 사용합니다. 또는 최소 및 최대 디스크 공간 사이의 볼륨 크기를 선택합니다.
7. 드라이브 문자 또는 경로 할당 화면에서 다음 중 하나를 수행하고 다음을 선택합니다.
   - 드라이브 문자를 사용하여 볼륨을 탑재하려면 다음 드라이브 문자 할당을 선택한 다음, 사용할 드라이브 문자를 선택합니다.
   - 볼륨을 폴더로 탑재하려면 다음 빈 NTFS 폴더에서 탑재를 선택한 다음 찾아보기를 선택하여 사용할 폴더를 생성하거나 선택합니다.
   - 드라이브 문자나 경로 없이 볼륨을 탑재하려면 드라이브 문자 또는 드라이브 경로 할당 안 함을 선택합니다.
8. 파티션 포맷 화면에서 볼륨을 포맷할지 여부를 지정합니다. 볼륨을 포맷하도록 선택한 경우 필요한 파일 시스템 및 단위 크기를 선택하고 볼륨 레이블을 지정합니다.
9. 다음, 완료를 선택합니다.

M1 및 C1 EC2 인스턴스의 인스턴스 스토어 스왑 볼륨 활성화

> **Note**
>
> 이 주제는 `c1.medium` 및 `m1.small` Linux 인스턴스에만 적용됩니다.

`c1.medium` 및 `m1.small` 인스턴스 유형은 물리적 메모리 양이 제한되어 있습니다. 따라서 시작 시간에 Linux 시스템의 가상 메모리의 역할을 할 수 있는 900MiB의 스왑 볼륨, 즉 스왑 공간이 부여됩니다. 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 Amazon EC2 인스턴스 유형 변경 섹션을 참조하세요.
> - 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 xvda3를 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 EC2 인스턴스의 인스턴스 스토어 볼륨 제한 섹션을 참조하세요.

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/xvdb`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(`MOUNTPOINT` 명령 출력에 lsblk로 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 중단되거나 최대 절전 모드로 전환되면 인스턴스 스토어 볼륨 데이터가 손실됩니다. 여기에는 Step 3에서 생성한 인스턴스 스토어 스왑 공간 포맷도 포함됩니다. 따라서 인스턴스 스토어 스왑 공간을 사용하도록 구성한 인스턴스를 중단했다가 다시 시작할 경우에는 새로운 인스턴스 스토어 볼륨에서 Step 1부터 Step 5까지 반복해야 합니다.

EC2 인스턴스의 인스턴스 스토어 볼륨 초기화

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 인스턴스 유형은 초기화 없이 실행 시점에 최고 성능을 제공합니다. 각 인스턴스 유형의 인스턴스 스토어에 대한 자세한 내용은 EC2 인스턴스의 인스턴스 스토어 볼륨 제한 섹션을 참조하세요.

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
```

Amazon EC2 인스턴스용 루트 볼륨

인스턴스를 시작하면 해당 인스턴스에 대한 루트 볼륨이 생성됩니다. 루트 볼륨에는 인스턴스를 부팅하는 데 사용되는 이미지가 있습니다. 인스턴스마다 단일 루트 볼륨이 있습니다. 실행 중 또는 실행 후에 인스턴스에 스토리지 볼륨을 추가할 수 있습니다.

루트 볼륨의 유형은 인스턴스를 시작하는 데 사용하는 AMI에 따라 결정됩니다. Amazon EBS 지원 AMI(Linux 및 Windows 인스턴스) 또는 인스턴스 스토어 지원 AMI(Linux 인스턴스만 해당) 중 하나에서 인스턴스를 시작할 수 있습니다. 각 유형의 AMI로 수행할 수 있는 작업에는 상당한 차이가 있습니다. 해당 차이점에 대한 자세한 내용은 루트 디바이스 유형 섹션을 참조하세요.

이러한 인스턴스는 시작 속도가 더 빠르고 영구 스토리지를 사용하므로 Amazon EBS 지원 AMI를 사용하는 것이 좋습니다.

루트 볼륨용으로 특정 디바이스 이름을 예약합니다. 자세한 내용은 Amazon EC2 인스턴스의 볼륨 디바이스 이름 섹션을 참조하세요.

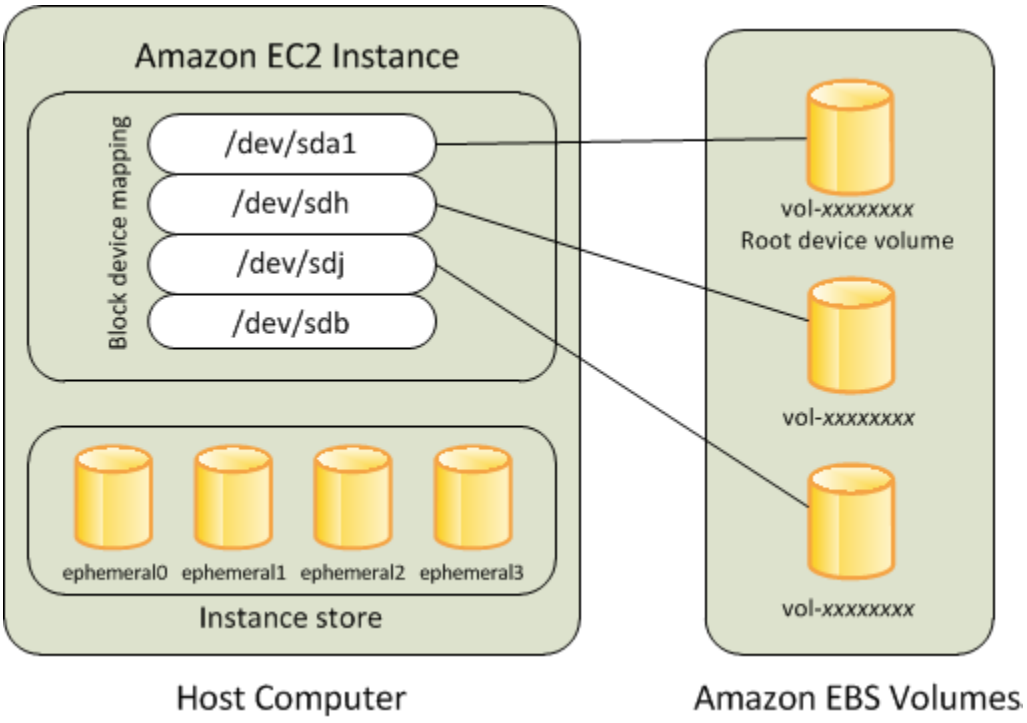
내용

- Amazon EBS 지원 인스턴스
- 인스턴스 스토어 지원 인스턴스(Linux 인스턴스만 해당)
- Amazon EC2 인스턴스가 종료된 후에도 Amazon EBS 루트 볼륨 유지
- Amazon EC2 인스턴스의 루트 볼륨을 중지하지 않고 교체

Amazon EBS 지원 인스턴스

루트 볼륨에 Amazon EBS를 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 태스크를 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. 자세한 내용은 Amazon EBS volumes를 참조하세요.



제한 사항

st1 또는 sc1 EBS 볼륨은 루트 볼륨으로 사용할 수 없습니다.

인스턴스 실패

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

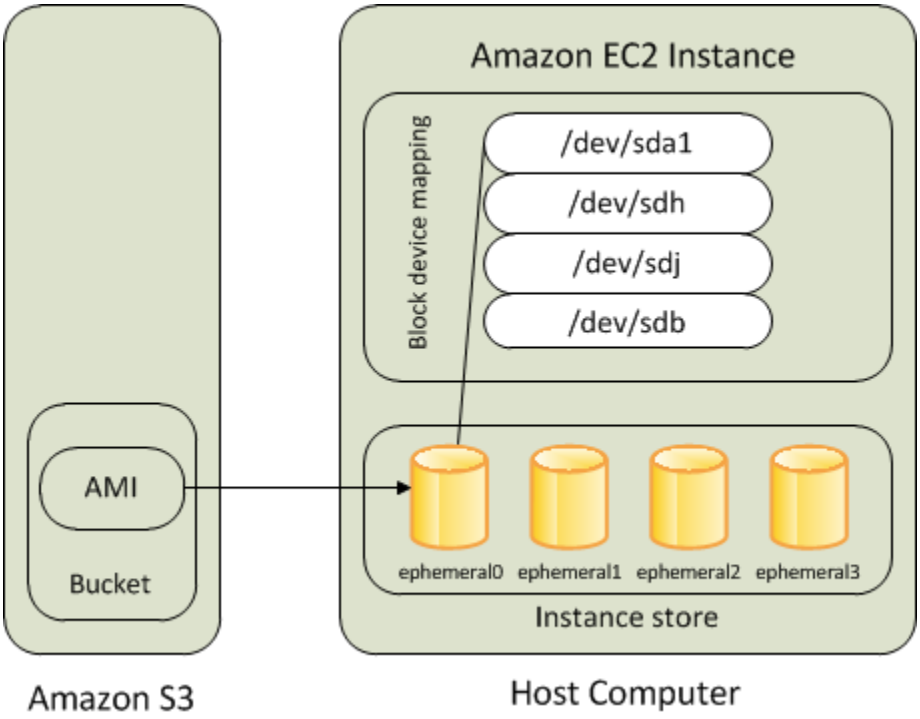
- 중지 후 다시 시작합니다(먼저 이 방법 시도).
- 모든 관련 볼륨의 스냅샷을 자동으로 생성하고 새 AMI를 생성합니다. 자세한 내용은 Amazon EBS 지원 AMI 생성 섹션을 참조하세요.
- 다음 단계에 따라 볼륨에 새 인스턴스를 연결합니다.
  1. 루트 볼륨의 스냅샷을 생성합니다.
  2. 스냅샷을 사용하여 새 AMI를 등록합니다.
  3. 새 AMI에서 새 인스턴스를 시작합니다.
  4. 나머지 Amazon EBS 볼륨을 이전 인스턴스에서 분리합니다.
  5. Amazon EBS 볼륨을 새 인스턴스에 다시 연결합니다.

인스턴스 스토어 지원 인스턴스(Linux 인스턴스만 해당)

> **Note**
>
> Windows 인스턴스는 인스턴스 저장소 백업 루트 볼륨을 지원하지 않습니다.

루트 볼륨에 인스턴스 스토어를 사용하는 인스턴스는 하나 이상의 인스턴스 스토어 볼륨을 자동으로 사용할 수 있으며, 이러한 볼륨 중 하나가 루트 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. 자세한 내용은 EC2 인스턴스용 인스턴스 스토어 임시 블록 스토리지 섹션을 참조하세요.



지원되는 인스턴스 유형

C1, C3, D2, I2, M1, M2, M3, R3, X1과 같은 인스턴스 유형만 인스턴스 스토어 볼륨을 루트 볼륨으로 지원합니다.

인스턴스 실패

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.

Amazon EC2 인스턴스가 종료된 후에도 Amazon EBS 루트 볼륨 유지

기본적으로 인스턴스의 Amazon EBS 루트 볼륨은 인스턴스 종료 시 삭제됩니다. 인스턴스가 종료된 후에도 Amazon EBS 루트 볼륨이 지속되도록 기본 동작을 변경할 수 있습니다. 기본 동작을 변경하려면 `DeleteOnTermination` 속성을 `false`로 설정합니다. 인스턴스 시작 이후에도 변경할 수 있습니다.

업무

- 인스턴스 시작 중 루트 볼륨이 지속되도록 구성
- 기존 인스턴스에서 루트 볼륨이 지속되도록 구성

- 루트 볼륨이 지속되도록 구성되었는지 확인

인스턴스 시작 중 루트 볼륨이 지속되도록 구성

인스턴스를 시작할 때 루트 볼륨이 지속되도록 구성할 수 있습니다.

Console

인스턴스를 시작할 때 루트 볼륨이 지속되도록 구성하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택한 후 인스턴스 시작을 선택합니다.
3. Amazon Machine Image(AMI)를 선택하고 인스턴스 유형을 선택한 후 키 페어를 선택하고 네트워크 설정을 구성합니다.
4. 스토리지 구성에서 고급을 선택합니다.
5. 루트 볼륨을 확장합니다.
6. 종료 시 삭제에서 예를 선택합니다.
7. 인스턴스 구성을 마치면 인스턴스 시작을 선택합니다.

AWS CLI

인스턴스를 시작할 때 루트 볼륨이 지속되도록 구성하는 방법

run-instances 명령을 사용하고 다음 옵션을 포함합니다.

```
--block-device-mappings file://mapping.json
```

`mapping.json`에서 `DeleteOnTermination` 속성을 `false`로 설정하는 블록 디바이스 매핑을 지정합니다.

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

PowerShell

인스턴스를 시작할 때 루트 볼륨이 지속되도록 구성하는 방법

New-EC2Instance cmdlet을 사용하고 다음 파라미터를 포함합니다.

```
-BlockDeviceMapping $bdm
```

`DeleteOnTermination` 속성을 `$false`로 설정하는 블록 디바이스 매핑을 생성합니다.

```
$ebs = New-Object Amazon.EC2.Model.EbsBlockDevice
$ebs.DeleteOnTermination = $false
$bdm = New-Object Amazon.EC2.Model.BlockDeviceMapping
$bdm.DeviceName = "dev/xvda"
$bdm.Ebs = $ebs
```

기존 인스턴스에서 루트 볼륨이 지속되도록 구성

실행 중인 인스턴스에 대해 루트 볼륨이 지속되도록 구성할 수 있습니다. 단, Amazon EC2 콘솔에서는 이 작업을 완료할 수 없습니다.

AWS CLI

기존 인스턴스에 대한 루트 볼륨이 지속되도록 구성하는 방법

`DeleteOnTermination` 속성을 `false`로 설정하는 블록 디바이스 매핑과 함께 modify-instance-attribute 명령을 사용합니다.

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

`mapping.json`에서 다음을 지정합니다.

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

```
]
```

PowerShell

기존 인스턴스에 대한 루트 볼륨이 지속되도록 구성하는 방법

`DeleteOnTermination` 속성을 `$false`로 설정하는 블록 디바이스 매핑과 함께 Edit-EC2InstanceAttribute cmdlet을 사용합니다.

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

루트 볼륨이 지속되도록 구성되었는지 확인

루트 볼륨이 지속되도록 구성되었는지 확인할 수 있습니다.

Console

루트 볼륨이 지속되도록 구성되었는지 확인하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택한 후 해당 인스턴스를 선택합니다.
3. 스토리지 탭의 블록 디바이스에서 루트 볼륨에 대한 항목을 찾습니다. 종료 시 삭제가 `No`인 경우 볼륨이 지속되도록 구성된 것입니다.

AWS CLI

루트 볼륨이 지속되도록 구성되었는지 확인하는 방법

describe-instances 명령을 사용하고 `DeleteOnTermination` 속성이 `false`로 설정되었는지 확인합니다.

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

```
    --query "Reservations[].Instances[].BlockDeviceMappings"
```

출력의 예시는 다음과 같습니다.

```
[
    [
        {
            "DeviceName": "/dev/xvda",
            "Ebs": {
                "AttachTime": "2024-07-12T04:05:33.000Z",
                "DeleteOnTermination": false,
                "Status": "attached",
                "VolumeId": "vol-1234567890abcdef0"

        }
    ]
]
```

PowerShell

루트 볼륨이 지속되도록 구성되었는지 확인하는 방법

Get-EC2Instance cmdlet을 사용하고 `DeleteOnTermination` 속성이 `False`로 설정되었는지 확인합니다.

```
(Get-EC2Instance -InstanceId i-
i-1234567890abcdef0).Instances.BlockDeviceMappings.Ebs
```

출력의 예시는 다음과 같습니다.

```
AssociatedResource  :
AttachTime          : 7/12/2024 4:05:33 AM
DeleteOnTermination : False
Operator            :
Status              : attached
VolumeId            : vol-1234567890abcdef0
```

Amazon EC2 인스턴스의 루트 볼륨을 중지하지 않고 교체

Amazon EC2를 사용하면 다음을 유지하면서 실행 중인 인스턴스의 루트 Amazon EBS 볼륨을 바꿀 수 있습니다.

- 인스턴스 스토어 볼륨에 저장된 데이터 - 인스턴스 스토어 볼륨은 루트 볼륨이 복원된 후에도 인스턴스에 연결된 상태로 유지됩니다.
- 데이터(루트가 아닌) Amazon EBS 볼륨에 저장된 데이터 - 루트가 아닌 Amazon EBS 볼륨은 루트 볼륨이 복원된 후에도 인스턴스에 연결된 상태로 유지됩니다.
- 네트워크 구성 — 모든 네트워크 인터페이스는 인스턴스에 연결된 상태로 유지되며 IP 주소, 식별자, 첨부 파일 ID를 유지합니다. 인스턴스를 사용할 수 있게 되면 보류 중인 모든 네트워크 트래픽이 플러시됩니다. 또한 인스턴스는 동일한 물리적 호스트에 유지되므로 퍼블릭 및 프라이빗 IP 주소와 DNS 이름을 유지합니다.
- IAM 정책 — IAM 프로필 및 인스턴스와 연결된 정책(예: 태그 기반 정책)은 유지되고 적용됩니다.

내용

- 루트 볼륨 대체 작동 방식
- 고려 사항
- 루트 볼륨 교체

루트 볼륨 대체 작동 방식

인스턴스의 루트 볼륨을 바꾸면 루트 볼륨 대체 작업을 생성하는 것입니다. 원래 루트 볼륨이 인스턴스에서 분리되고 새 루트 볼륨이 해당 위치에 있는 인스턴스에 연결됩니다. 인스턴스의 블록 디바이스 매핑은 대체 루트 볼륨의 ID를 반영하도록 업데이트됩니다.

인스턴스의 루트 볼륨을 바꾸면 새 볼륨의 스냅샷 소스를 지정해야 합니다. 가능한 옵션은 다음과 같습니다.

루트 볼륨을 원래 상태로 복원

이 옵션은 현재 루트 볼륨을 생성할 때 사용한 스냅샷을 기반으로 하는 볼륨으로 바꿉니다.

시작 상태 사용 시 고려 사항

대체 루트 볼륨은 원래 루트 볼륨과 동일한 유형, 크기 및 종료 시 삭제 속성을 갖습니다.

스냅샷을 사용하여 루트 볼륨 바꾸기

이 옵션은 현재 루트 볼륨을 지정한 스냅샷을 기반으로 하는 대체 볼륨으로 바꿉니다. 이 루트 볼륨에서 이전에 생성한 특정 스냅샷을 예로 들 수 있습니다. 이는 루트 볼륨 손상으로 인한 문제 또는 게스트 운영 체제의 네트워크 구성 오류로 인한 문제를 복구해야 하는 경우에 유용합니다.

대체 루트 볼륨은 원래 루트 볼륨과 동일한 유형, 크기 및 종료 시 삭제 속성을 갖습니다.

스냅샷 사용 시 고려 사항

- 현재 루트 볼륨과 동일한 계보에 속하는 스냅샷만 사용할 수 있습니다.
- 루트 볼륨의 스냅샷을 사용하여 생성된 스냅샷 복사본은 사용할 수 없습니다.
- 루트 볼륨을 성공적으로 바꾼 후에도 원래 루트 볼륨에서 가져온 스냅샷을 사용하여 새(대체) 루트 볼륨을 바꿀 수 있습니다.

AMI를 사용하여 루트 볼륨 바꾸기

이 옵션은 지정한 AMI를 사용하여 현재 루트 볼륨을 바꿉니다. 운영 체제 및 애플리케이션 패치 또는 업그레이드를 수행해야 하는 경우에 유용합니다. AMI는 인스턴스와 동일한 제품 코드, 결제 정보, 아키텍처 유형 및 가상화 유형을 가져야 합니다.

ENA 또는 sriov-net에 대해 인스턴스가 활성화된 경우 이러한 기능을 지원하는 AMI를 사용해야 합니다. ENA 또는 sriov-net에 대해 인스턴스가 활성화되지 않은 경우 해당 기능에 대한 지원이 포함되지 않은 AMI를 선택하거나 ENA 또는 sriov-net를 지원하는 AMI를 선택할 경우 지원을 자동으로 추가할 수 있습니다.

NitroTPM에 대해 인스턴스가 활성화된 경우 NitroTPM이 활성화된 AMI를 사용해야 합니다. 선택한 AMI와 상관없이 인스턴스가 이에 적절하게 구성되지 않은 경우 NitroTPM 지원이 활성화되지 않습니다.

인스턴스가 AMI의 부팅 모드를 지원하는 한, 인스턴스와 다른 부팅 모드의 AMI를 선택할 수 있습니다. 인스턴스에서 부팅 모드를 지원하지 않는 경우 요청이 실패합니다. 인스턴스에서 부팅 모드를 지원하는 경우 새 부팅 모드가 인스턴스에 전파되고 그에 따라 해당 UEFI 데이터가 업데이트됩니다. 부팅 순서를 수동으로 수정하거나 프라이빗 UEFI 보안 부팅 키를 추가하여 프라이빗 커널 모듈을 로드한 경우 루트 볼륨 대체 중 변경 사항이 손실됩니다.

대체 루트 볼륨은 원래 루트 볼륨과 동일한 볼륨 유형 및 종료 시 삭제 속성을 가지며, AMI 루트 볼륨 블록 디바이스 매핑의 크기를 가져옵니다.

> **Note**
>
> AMI의 루트 볼륨 블록 디바이스 매핑의 크기는 원래 루트 볼륨의 크기보다 크거나 같아야 합니다. AMI의 루트 볼륨 블록 디바이스 매핑 크기가 원래 루트 볼륨의 크기보다 작으면 요청이 실패합니다.

루트 볼륨 대체 작업이 완료된 후 콘솔, AWS CLI 또는 AWS SDK를 사용하여 인스턴스를 설명할 때 다음과 같은 새로운 정보와 업데이트된 정보가 반영됩니다.

- 새 AMI ID
- 루트 볼륨의 새 볼륨 ID
- 업데이트된 부팅 모드 구성(AMI에 의해 변경된 경우)
- 업데이트된 NitroTPM 구성(AMI에 의해 활성화된 경우)
- 업데이트된 ENA 구성(AMI에 의해 활성화된 경우)
- 업데이트된 sriov-net 구성(AMI에 의해 활성화된 경우)

새 AMI ID는 인스턴스 메타데이터에도 반영됩니다.

AMI 사용 시 고려 사항:

- 여러 블록 디바이스 매핑이 있는 AMI를 사용하는 경우 AMI의 루트 볼륨만 사용됩니다. 루트가 아닌 다른 볼륨은 무시됩니다.
- AMI 및 관련 루트 볼륨 스냅샷에 대한 권한이 있는 경우에만 이 기능을 사용할 수 있습니다. AWS Marketplace AMI에서는 이 기능을 사용할 수 없습니다.
- 인스턴스에 제품 코드가 없는 경우에만 제품 코드 없이 AMI를 사용할 수 있습니다.
- AMI의 루트 볼륨 블록 디바이스 매핑의 크기는 원래 루트 볼륨의 크기보다 크거나 같아야 합니다. AMI의 루트 볼륨 블록 디바이스 매핑 크기가 원래 루트 볼륨의 크기보다 작으면 요청이 실패합니다.
- 인스턴스에 대한 인스턴스 ID 문서는 자동으로 업데이트됩니다.
- 인스턴스가 NitroTPM을 지원하는 경우 인스턴스의 NitroTPM 데이터가 재설정되고 새 키가 생성됩니다.

루트 볼륨 대체 프로세스가 완료된 후 원래 루트 볼륨을 유지할지 여부를 선택할 수 있습니다. 대체 프로세스가 완료된 후 원래 루트 볼륨을 삭제하도록 선택하면 원래 루트 볼륨이 자동으로 삭제되어 복구할 수 없게 됩니다. 프로세스가 완료된 후 원래 루트 볼륨을 유지하도록 선택하면 볼륨이 계정에 프로비저닝된 상태로 유지됩니다. 더 이상 필요하지 않은 볼륨은 수동으로 삭제해야 합니다.

루트 볼륨 대체 작업은 다음 상태를 거칩니다.

- `pending` - 대체 볼륨이 생성되고 있습니다.
- `in-progress` - 원래 볼륨이 분리되고 대체 볼륨이 연결되고 있습니다.
- `succeeded` - 대체 볼륨이 인스턴스에 성공적으로 연결되었으며 인스턴스를 사용할 수 있습니다.

- `failing` - 대체 작업이 실패하는 중입니다.
- `failed` - 대체 작업이 실패했지만 루트 볼륨은 여전히 연결되어 있습니다.
- `failing-detached` - 대체 작업이 실패하는 중이며 인스턴스에 연결된 루트 볼륨이 없을 수 있습니다.
- `failed-detached` - 대체 작업이 실패했으며 인스턴스에 연결된 루트 볼륨이 없습니다.

루트 볼륨 대체 작업이 실패하면 인스턴스가 재부팅되고 원래 루트 볼륨은 인스턴스에 연결된 상태로 유지됩니다.

고려 사항

시작하기 전에 다음을 고려합니다.

요구 사항

- 인스턴스는 `running` 상태여야 합니다.
- 프로세스 중에 인스턴스가 자동으로 재부팅됩니다. 재부팅 중에는 메모리(RAM)의 내용이 지워집니다. 수동 재부팅은 필요하지 않습니다.
- 루트 볼륨이 인스턴스 스토어 볼륨이면 교체할 수 없습니다. Amazon EBS 루트 볼륨이 있는 인스턴스만 지원됩니다.
- 모든 가상화된 인스턴스 유형과 EC2 Mac 베어 메탈 인스턴스의 루트 볼륨을 바꿀 수 있습니다. 다른 모든 베어 메탈 인스턴스 유형은 지원되지 않습니다.
- 인스턴스의 이전 루트 볼륨과 동일한 계보에 속하는 모든 스냅샷을 사용할 수 있습니다.
- 현재 리전에서 계정에 기본적으로 Amazon EBS 암호화가 활성화되어 있는 경우 루트 볼륨 대체 작업으로 생성된 대체 루트 볼륨은 지정된 스냅샷이나 지정된 AMI의 루트 볼륨의 암호화 상태에 관계없이 항상 암호화됩니다.

암호화 결과

다음 표에는 가능한 암호화 결과가 요약되어 있습니다.

	원래 루트 볼륨	지정된 스냅샷 또는 AMI	암호화 기본 제공	대체 루트 볼륨	대체 루트 볼륨에 사용되는 암호화 키
초기 시작 상태로 대체 루트 볼륨 복원	Encrypted	해당 사항 없음	고려되지 않음	Encrypted	원래 루트 볼륨과 동일한 KMS 키
	암호화되지 않음	해당 사항 없음	비활성	암호화되지 않음	해당 사항 없음
	암호화되지 않음	해당 사항 없음	활성화됨	Encrypted	계정의 Amazon EBS 암호화를 위한 기본 KMS 키
스냅샷 또는 AMI에서 대체 루트 볼륨 복원	Encrypted	암호화되지 않음	고려되지 않음	Encrypted	원래 루트 볼륨과 동일한 KMS 키
	Encrypted	Encrypted	고려되지 않음	Encrypted	원래 루트 볼륨과 동일한 KMS 키
	암호화되지 않음	암호화되지 않음	비활성	암호화되지 않음	해당 사항 없음
	암호화되지 않음	암호화되지 않음	활성화됨	Encrypted	계정의 Amazon EBS 암호화를 위한 기본 KMS 키
	암호화되지 않음	Encrypted	고려되지 않음	Encrypted	AMI 또는 스냅샷이 계정 소유인 경우 대체 볼

	원래 루트 볼륨	지정된 스냅샷 또는 AMI	암호화 기본 제공	대체 루트 볼륨	대체 루트 볼륨에 사용되는 암호화 키
					륨은 AMI 또는 스냅샷의 KMS 키로 암호화됩니다. AMI 또는 스냅샷이 계정과 공유되는 경우 대체 볼륨은 계정의 Amazon EBS 암호화를 위한 기본 KMS 키로 암호화됩니다.

루트 볼륨 교체

인스턴스의 루트 볼륨을 바꾸면 루트 볼륨 대체 작업이 생성됩니다. 루트 볼륨 대체 작업을 사용하여 대체 프로세스의 진행 상황과 결과를 모니터링할 수 있습니다.

Console

루트 볼륨을 교체하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 루트 볼륨을 교체할 인스턴스를 선택하고 작업, 모니터링 및 문제 해결, 루트 볼륨 교체를 선택합니다.

> **Note**
>
> 선택한 인스턴스가 `running` 상태가 아니면 루트 볼륨 교체 작업이 비활성화됩니다.

4. 복원에서 다음 옵션 중 하나를 선택합니다.

   - 시작 상태 - 현재 루트 볼륨을 생성하는 데 사용된 스냅샷에서 대체 루트 볼륨을 복원합니다.
   - 스냅샷 - 대체 루트 볼륨을 지정한 스냅샷으로 복원합니다. 스냅샷에서 사용할 스냅샷을 선택합니다.
   - 이미지 - 지정한 AMI를 사용하여 대체 루트 볼륨을 복원합니다. 이미지에서 사용할 AMI를 선택합니다.

5. (선택 사항) 볼륨 초기화 속도의 경우 스냅샷 블록을 Amazon S3에서 볼륨으로 다운로드할 볼륨 초기화를 위한 Amazon EBS 프로비저닝 속도(볼륨 초기화 속도)를 MiB/s 단위로 지정할 수 있습니다. 자세한 내용은 볼륨 초기화에 Amazon EBS 프로비저닝 속도 사용을 참조하세요. 기본 초기화 속도 또는 빠른 스냅샷 복원(선택한 스냅샷에 대해 활성화된 경우)을 사용하려면 속도를 지정하지 마세요.
6. (선택 사항) 대체하려는 루트 볼륨을 삭제하려면 대체된 루트 볼륨 삭제를 선택합니다.
7. 대체 작업 생성을 선택합니다.
8. 대체 작업을 모니터링하려면 인스턴스의 스토리지 탭을 선택하고 최근 루트 볼륨 대체 작업을 확장합니다.

AWS CLI

시작 상태로 대체 루트 볼륨 복원

create-replace-root-volume-task 명령을 사용합니다. `--instance-id`에 대해 루트 볼륨을 대체할 인스턴스의 ID를 지정합니다. `--snapshot-id` 및 `--image-id` 파라미터를 생략합니다. 원래 루트 볼륨을 바꾼 후 삭제하려면 `--delete-replaced-root-volume`을 포함하고 `true`를 지정합니다. 스냅샷 블록이 Amazon S3에서 볼륨으로 다운로드되는 볼륨 초기화 속도를 지정하려면 `--volume-initialization-rate`에 `100`~`300`MiB/s 사이의 값을 지정합니다.

```
aws ec2 create-replace-root-volume-task \
--instance-id i-1234567890abcdef0 \
--delete-replaced-root-volume \
--volume-initialization-rate 150
```

대체 루트 볼륨을 특정 스냅샷으로 복원

create-replace-root-volume-task 명령을 사용합니다. `--instance-id`에 대해 루트 볼륨을 대체할 인스턴스의 ID를 지정합니다. `--snapshot-id`에 대해 사용할 스냅샷의 ID를 지정합니다. 원래 루트 볼륨을 바꾼 후 삭제하려면 `--delete-replaced-root-volume`을 포함하고 `true`를 지정합

니다. 스냅샷 블록이 Amazon S3에서 볼륨으로 다운로드되는 볼륨 초기화 속도를 지정하려면 `--volume-initialization-rate`에 `100~300`MiB/s 사이의 값을 지정합니다.

```
aws ec2 create-replace-root-volume-task \
--instance-id i-1234567890abcdef0 \
--snapshot-id snap-9876543210abcdef0 \
--delete-replaced-root-volume \
--volume-initialization-rate 150
```

AMI를 사용하여 대체 루트 볼륨 복원

create-replace-root-volume-task 명령을 사용합니다. `--instance-id`에 대해 루트 볼륨을 대체할 인스턴스의 ID를 지정합니다. `--image-id`에 대해 사용할 AMI의 ID를 지정합니다. 원래 루트 볼륨을 바꾼 후 삭제하려면 `--delete-replaced-root-volume`을 포함하고 `true`를 지정합니다. 스냅샷 블록이 Amazon S3에서 볼륨으로 다운로드되는 볼륨 초기화 속도를 지정하려면 `--volume-initialization-rate`에 `100~300`MiB/s 사이의 값을 지정합니다.

```
aws ec2 create-replace-root-volume-task \
--instance-id i-1234567890abcdef0 \
--image-id ami-09876543210abcdef \
--delete-replaced-root-volume \
--volume-initialization-rate 150
```

루트 볼륨 교체 작업의 상태를 보는 방법

describe-replace-root-volume-tasks 명령을 사용하고 보려는 루트 볼륨 교체 작업의 ID를 지정합니다.

```
aws ec2 describe-replace-root-volume-tasks \
    --replace-root-volume-task-ids replacevol-1234567890abcdef0 \
    --query ReplaceRootVolumeTasks[].TaskState
```

출력의 예시는 다음과 같습니다.

```
[
    "succeeded"
]
```

또는 인스턴스를 기준으로 결과를 필터링할 `instance-id` 필터를 지정합니다.

```
$ aws ec2 describe-replace-root-volume-tasks \
    --filters Name=instance-id,Values=i-1234567890abcdef0
```

PowerShell

시작 상태로 대체 루트 볼륨 복원

New-EC2ReplaceRootVolumeTask 명령을 사용합니다. -InstanceId에 대해 루트 볼륨을 대체할 인스턴스의 ID를 지정합니다. -SnapshotId 및 -ImageId 파라미터를 생략합니다. 원래 루트 볼륨을 바꾼 후 삭제하려면 -DeleteReplacedRootVolume을 포함하고 $true를 지정합니다. 스냅샷 블록이 Amazon S3에서 볼륨으로 다운로드되는 볼륨 초기화 속도를 지정하려면 -VolumeInitializationRate에 100~300MiB/s 사이의 값을 지정합니다.

```
New-EC2ReplaceRootVolumeTask `
    -InstanceId i-1234567890abcdef0 `
    -VolumeInitializationRate 150 `
    -DeleteReplacedRootVolume $true
```

대체 루트 볼륨을 특정 스냅샷으로 복원

New-EC2ReplaceRootVolumeTask 명령을 사용합니다. --InstanceId에 대해 루트 볼륨을 대체할 인스턴스의 ID를 지정합니다. -SnapshotId에 대해 사용할 스냅샷의 ID를 지정합니다. 원래 루트 볼륨을 바꾼 후 삭제하려면 -DeleteReplacedRootVolume을 포함하고 $true를 지정합니다. 스냅샷 블록이 Amazon S3에서 볼륨으로 다운로드되는 볼륨 초기화 속도를 지정하려면 -VolumeInitializationRate에 100~300MiB/s 사이의 값을 지정합니다.

```
New-EC2ReplaceRootVolumeTask `
    -InstanceId i-1234567890abcdef0 `
    -SnapshotId snap-9876543210abcdef0 `
    -VolumeInitializationRate 150 `
    -DeleteReplacedRootVolume $true
```

AMI를 사용하여 대체 루트 볼륨 복원

New-EC2ReplaceRootVolumeTask 명령을 사용합니다. -InstanceId에 대해 루트 볼륨을 대체할 인스턴스의 ID를 지정합니다. -ImageId에 대해 사용할 AMI의 ID를 지정합니다. 원래 루트 볼륨을 바꾼 후 삭제하려면 -DeleteReplacedRootVolume을 포함하고 $true를 지정합니다. 스냅샷 블록이 Amazon S3에서 볼륨으로 다운로드되는 볼륨 초기화 속도를 지정하려면 -VolumeInitializationRate에 100~300MiB/s 사이의 값을 지정합니다.

```
New-EC2ReplaceRootVolumeTask `
    -InstanceId i-1234567890abcdef0 `
    -ImageId ami-0abcdef1234567890 `
    -VolumeInitializationRate 150 `
    -DeleteReplacedRootVolume $true
```

루트 볼륨 교체 작업의 상태를 보는 방법

Get-EC2ReplaceRootVolumeTask 명령을 사용하고 보려는 루트 볼륨 대체 작업의 ID를 지정합니다.

```
(Get-EC2ReplaceRootVolumeTask `
    -ReplaceRootVolumeTaskIds replacevol-1234567890abcdef0).TaskState.Value
```

출력의 예시는 다음과 같습니다.

```
Succeeded
```

또는 인스턴스를 기준으로 결과를 필터링할 `instance-id` 필터를 지정합니다.

```
PS C:\> Get-EC2ReplaceRootVolumeTask -Filters @{Name = 'instance-id'; Values =
 'i-1234567890abcdef0'} | Format-Table
```

Amazon EC2 인스턴스의 볼륨 디바이스 이름

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 Amazon EC2 인스턴스의 Amazon EBS 볼륨 한도 섹션을 참조하세요.

내용

- 사용 가능한 디바이스 이름
- 디바이스 이름 고려 사항

사용 가능한 디바이스 이름

Linux 인스턴스

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 가상화 유형 단원을 참조하십시오.

다음 표는 블록 디바이스 매핑에서 또는 EBS 볼륨에 연결 시 지정할 수 있는 사용 가능한 디바이스 이름을 나열합니다.

가상화 유형	Available	루트 볼륨용으로 예약됨	EBS 데이터 볼륨 추천	인스턴스 스토어 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[a-c][a-z] /dev/xvdd[a-x]	AMI에 따라 다름 /dev/sda1 또는 /dev/xvda	/dev/sd[b-z] /dev/xvdb[b-z] *	/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.

Windows 인스턴스

AWS Windows AMI는 다음과 같은 드라이버 세트 중 하나를 이용해 가상화 하드웨어에 대한 액세스를 허용합니다.

- AWS PV: Windows 인스턴스의 반가상화 드라이버
- AWS NVMe: AWS NVMe 드라이버

Nitro 기반 인스턴스의 디바이스 이름

다음 표는 블록 디바이스 매핑에서 또는 EBS 볼륨을 Nitro 기반 인스턴스에 연결 시 지정할 수 있는 사용 가능한 디바이스 이름을 나열합니다.

드라이버 유형	Available	루트 볼륨용으로 예약됨	EBS 볼륨 추천	인스턴스 스토어 볼륨
AWS NVMe	xvd[a-z] xvd[a-c][a-z] xvdd[a-x] /dev/sda1	/dev/sda1	xvd[b-z] xvdb[b-z]	*

* NVMe 인스턴스 스토어 볼륨은 자동으로 열거되고 Windows 드라이브 문자가 할당됩니다.

Xen 기반 인스턴스의 디바이스 이름

다음 표는 블록 디바이스 매핑에서 또는 EBS 볼륨을 Xen 기반 인스턴스에 연결 시 지정할 수 있는 사용 가능한 디바이스 이름을 나열합니다.

드라이버 유형	Available	루트 볼륨용으로 예약됨	EBS 볼륨 추천	인스턴스 스토어 볼륨
AWS PV	xvd[b-z] xvd[b-c][a-z] /dev/sda1 /dev/sd[b-e]	/dev/sda1	xvd[f-z]	xvdc[a-x] xvd[a-e]
Citrix PV(더 이상 지원되지 않음)	xvd[b-z] xvd[b-c][a-z] /dev/sda1 /dev/sd[b-e]	/dev/sda1	xvd[f-z]	xvdc[a-x] xvd[a-e]
Red Hat PV(더 이상 지원되지 않음)	xvd[a-z] xvd[b-c][a-z] /dev/sda1 /dev/sd[b-e]	/dev/sda1	xvd[f-p]	xvdc[a-x] xvd[a-e]

인스턴스 스토어 볼륨에 대한 자세한 내용은 EC2 인스턴스용 인스턴스 스토어 임시 블록 스토리지 섹션을 참조하세요. EBS 디바이스를 식별하는 방법을 포함하여 NVMe EBS 볼륨(Nitro 기반 인스턴스)에 대한 자세한 내용은 Amazon EBS 사용 설명서의 Amazon EBS and NVMe를 참조하세요.

디바이스 이름 고려 사항

디바이스 이름을 선택할 때는 다음 사항에 주의하세요.

- 인스턴스 시작 시 문제가 발생할 수 있으므로 사용하는 디바이스 이름의 끝부분이 겹치지 않아야 합니다. 예를 들어 동일한 인스턴스에 연결된 볼륨에는 `/dev/xvdf`와 `xvdf` 같은 조합을 사용하지 마세요.
- 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 열거되고 NVMe 디바이스 이름(Linux 인스턴스) 또는 Windows 드라이브 문자(Windows 인스턴스)가 할당됩니다.
- (Windows 인스턴스) AWS Windows AMI는 처음 부팅할 때 인스턴스를 준비하는 추가 소프트웨어와 함께 제공됩니다. EC2Config 서비스(Windows Server 2016 이전의 Windows AMI) 또는 EC2Launch(Windows Server 2016 이상)입니다. 디바이스가 드라이브에 매핑된 후 이 서비스가 초기화되고 마운트됩니다. 루트 디바이스는 초기화된 다음 C:\C:\로 마운트됩니다. 기본적으로, EBS 볼륨이 Windows 인스턴스에 연결되면 인스턴스에서 드라이브 문자로 표시됩니다. 설정을 변경하여 사양에 따라 볼륨의 드라이브 문자를 설정할 수 있습니다. 인스턴스 스토어 볼륨의 경우, 기본값은 드라이버에 따라 다릅니다. AWS PV 드라이버 및 Citrix PV 드라이버는 인스턴스 스토어 볼륨에 Z:부터 A:까지의 드라이브 문자를 할당합니다. Red Hat 드라이버는 인스턴스 스토어 볼륨에 D:부터 Z:까지 드라이브 문자를 배정합니다. 자세한 정보는 Amazon EC2 Windows 인스턴스의 Windows 시작 에이전트 및 Amazon EC2 Windows 인스턴스용 볼륨 연결 및 매핑 방법 섹션을 참조하세요.
- (Linux 인스턴스) 커널의 블록 디바이스 드라이버에 따라 디바이스는 사용자가 지정한 것과는 다른 이름으로 디바이스가 연결될 수 있습니다. 예를 들어 /dev/sdh라는 디바이스 이름을 지정할 경우 디바이스 이름이 /dev/xvdh 또는 /dev/hdh로 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 수 있습니다(즉 /dev/sda가 /dev/xvde로 바뀔 수 있음). 이 경우 각 디바이스 이름에서 뒤에 오는 문자는 같은 수로 늘어납니다. 예를 들어 /dev/sdb 이름을 /dev/xvdf로 바꾼 경우 /dev/sdc 이름이 /dev/xvdg로 바뀝니다. Amazon Linux에서는 이름이 바뀐 디바이스에 사용자가 지정한 이름에 대한 심볼 링크가 생성됩니다. 운영 체제가 다를 경우 다르게 작동할 수 있습니다.
- (Linux 인스턴스) HVM AMI는 루트 디바이스용으로 예약된 /dev/sda1 및 /dev/sda2를 제외하고는 디바이스 이름에 후행 번호 사용을 지원하지 않습니다. /dev/sda2을 사용하는 것이 가능하지만 HVM 인스턴스와 함께 이 디바이스 매핑을 사용하지 않는 것이 좋습니다.
- (Linux 인스턴스) PV AMI를 사용할 때 후행 숫자가 있거나 없는 동일한 디바이스 문자를 공유하는 볼륨은 연결할 수 없습니다. 예를 들어, 볼륨을 /dev/sdc로 연결한 다음 다른 볼륨을 /dev/sdc1에 연결하면 인스턴스에서는 /dev/sdc만을 볼 수 있습니다. 디바이스 이름 끝에 숫자를 사용하려면 기본 문자가 동일한 모든 디바이스 이름의 끝에 숫자를 사용해야 합니다(/dev/sdc1, /dev/sdc2, /dev/sdc3 등).
- (Linux 인스턴스) 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.

선택한 디바이스 이름을 지정하기 전에 해당 이름을 사용할 수 있는지 확인하세요. 그렇지 않으면 디바이스 이름이 이미 사용 중이라는 오류가 발생합니다. 디스크 디바이스와 해당 마운트 지점을 보려면 lsblk 명령(Linux 인스턴스) 또는 디스크 관리 유틸리티 또는 diskpart 명령(Windows 인스턴스)을 사용합니다.

Amazon EC2 인스턴스의 볼륨에 대한 블록 디바이스 매핑

시작한 각 인스턴스에는 연결된 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 추가 EBS 볼륨을 더 연결할 수도 있습니다. 그러나 블록 디바이스 매핑을 사용하여 인스턴스가 시작되었을 때 볼륨을 연결하는 방식으로만 인스턴스에 인스턴스 스토어 볼륨을 연결할 수 있습니다.

내용

- 블록 디바이스 매핑의 개념
- AMI에 블록 디바이스 매핑 추가
- Amazon EC2 인스턴스에 블록 디바이스 매핑 추가

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. 예를 들어 하드 디스크, CD-ROM 드라이브 및 플래시 드라이브 등이 있습니다. 블록 디바이스는 컴퓨터에 물리적으로 장착될 수 있고 그렇지 않은 경우 컴퓨터에 물리적으로 장착된 것처럼 임의 액세스가 가능합니다.

Amazon EC2가 지원하는 두 가지 블록 디바이스 유형:

- 인스턴스 스토어 볼륨(기본 하드웨어가 인스턴스의 호스트 컴퓨터에 물리적으로 장착된 가상 디바이스)
- EBS 볼륨(원격 스토리지 디바이스)

블록 디바이스 매핑은 인스턴스에 연결할 블록 디바이스(인스턴스 볼륨 및 EBS 볼륨)를 정의합니다. AMI 생성 시 블록 디바이스 매핑을 지정하면 AMI에서 실행되는 모든 인스턴스가 해당 매핑을 사용할 수 있습니다. 아니면, 인스턴스 생성 시 블록 디바이스 매핑을 지정하여 이 매핑이 인스턴스가 실행된 AMI에서 지정된 매핑을 재정의하도록 할 수 있습니다. 인스턴스 유형에서 지원되는 모든 NVMe 인스

턴스 스토어 볼륨이 인스턴스 시작 시 자동으로 열거되고 디바이스 이름이 할당됩니다. 따라서 블록 디바이스 매핑에 이를 포함하는 것은 효과가 없습니다.

목차

- 블록 디바이스 매핑 항목
- 블록 디바이스 매핑 인스턴스 스토어 경고
- 블록 디바이스 매핑 예제
- 운영 체제에서 디바이스 사용 방법

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 Amazon EC2 인스턴스의 볼륨 디바이스 이름 섹션을 참조하세요.

인스턴스 스토어 볼륨의 경우 다음 정보도 지정합니다.

- 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 볼륨의 경우 다음 정보도 적용됩니다.

- 이러한 볼륨은 자동으로 열거되고 디바이스 이름이 할당되므로 블록 디바이스 매핑에 이를 포함하는 것은 효과가 없습니다.

EBS 볼륨의 경우 다음 정보도 지정합니다.

- 블록 디바이스를 생성하기 위해 사용하는 스냅샷 ID(snap-xxxxxxxx). 볼륨 크기를 지정하는 경우 이 값은 선택 사항입니다. 보관된 스냅샷의 ID를 지정할 수 없습니다.
- GiB 단위의 볼륨 크기입니다. 지정된 크기는 지정된 스냅샷 크기 이상이어야 합니다.
- 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.

- 볼륨 유형입니다. 범용 SSD의 경우 gp2 및 gp3, 프로비저닝된 IOPS SSD의 경우 io1 및 io2, 처리량 최적화 HDD의 경우 st1, 콜드 HDD의 경우 sc1, 마그네틱의 경우 standard일 수 있습니다.
- 볼륨이 지원하는 초당 입력/출력 작업 수(IOPS). (io1 및 io2 볼륨에만 사용됩니다.)

블록 디바이스 매핑 인스턴스 스토어 경고

블록 디바이스 매핑에 인스턴스 스토어 볼륨이 있는 AMIs에서 인스턴스를 시작하는 경우 고려해야 할 몇 가지 경고 사항이 있습니다.

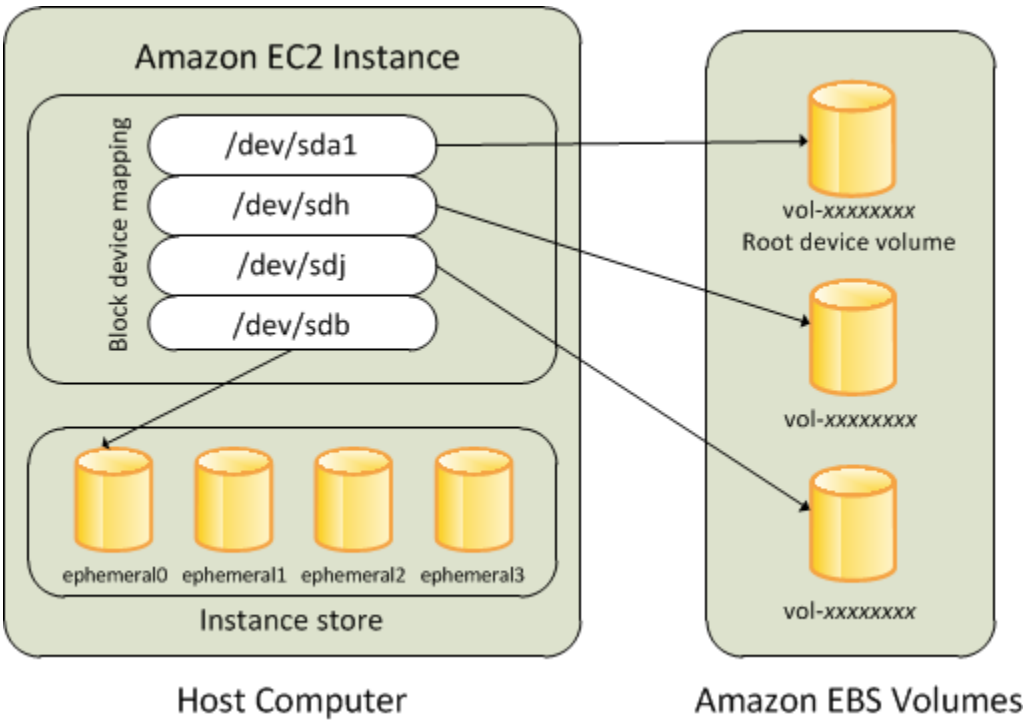
- 일부 인스턴스 유형은 다른 인스턴스보다 인스턴스 스토어 볼륨이 더 있거나 어떤 인스턴스 유형은 인스턴스 스토어 볼륨이 아예 없을 수도 있습니다. 인스턴스 볼륨이 1개의 인스턴스 스토어 볼륨을 지원하는 데 AMI에 2개의 인스턴스 스토어 볼륨이 있는 경우 인스턴스는 1개의 인스턴스 스토어 볼륨으로 실행됩니다.
- 인스턴스 스토어 볼륨은 실행 시에만 매핑될 수 있습니다. 인스턴스 스토어 볼륨이 없는 인스턴스(t2.micro 등)는 중지할 수 없으므로 해당 인스턴스를 인스턴스 스토어 볼륨을 지원하는 유형으로 변경한 다음 인스턴스 스토어 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 볼륨이 있는 경우 실행 시 인스턴스 스토어 블록 디바이스 매핑을 지정해야 인스턴스가 실행될 때 인스턴스 스토어 볼륨을 사용할 수 있습니다.

블록 디바이스 매핑 예제

이 그림은 EBS 기반 인스턴스의 블록 디바이스 매핑 예제를 보여줍니다. /dev/sdb를 ephemeral0으로 매핑하고 두 개의 EBS 볼륨을 각각 /dev/sdh 및 /dev/sdj로 매핑합니다. 또한 여기에서 루트 디바이스 볼륨인 EBS 볼륨은 /dev/sda1입니다.



이 예제 블록 디바이스 매핑에서는 이 주제와 관련된 예제 명령어 및 API가 사용되었습니다. AMI용 블록 디바이스 매핑 지정 및 인스턴스를 시작할 때 블록 디바이스 매핑 업데이트에서 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 인스턴스 – 블록 디바이스 매핑에 지정된 디바이스 이름은 인스턴스가 처음 부팅될 때 해당 블록 디바이스에 매핑됩니다. 인스턴스 유형에 따라 어느 인스턴스 스토어 볼륨이 포맷되고 기본 마운트될지가 결정됩니다. 사용자는 인스턴스 유형에 따라 사용 가능한 인스턴스 스토어 볼륨을 초과하지 않는 범위 내에서 실행 시 인스턴스 스토어 볼륨을 추가로 마운트할 수 있습니다. 자세한 내용은 EC2 인스턴스용 인스턴스 스토어 임시 블록 스토리지 섹션을 참조하세요. 인스턴스용 블록 디바이스 드라이버에 따라 볼륨 포맷 및 마운트 시 어느 디바이스가 사용될지가 결정됩니다.

Windows 인스턴스 - 인스턴스가 처음 부팅될 때 블록 디바이스 매핑에 지정된 디바이스 이름이 해당 블록 디바이스에 매핑된 다음 Ec2Config 서비스가 드라이브를 초기화하고 마운트합니다. 루트 디바이스 볼륨은 C:\로 마운트됩니다. 인스턴스 스토어 볼륨은 Z:\, Y:\ 등으로 마운트됩니다. EBS 볼륨

이 마운트될 때는 사용 가능한 드라이브 문자를 사용하여 마운트될 수 있습니다. 그러나 EBS 볼륨에 드라이브 문자를 할당하는 방법을 구성할 수 있습니다. 자세한 내용은 를 the section called “Windows 시작 에이전트” 섹션을 참조하세요.

AMI에 블록 디바이스 매핑 추가

각 AMI에는 AMI에서 시작될 때 인스턴스로 연결될 블록 디바이스를 지정하는 블록 디바이스 매핑이 있습니다. AMI에 추가 블록 디바이스를 추가하려면 고유 AMI를 생성해야 합니다.

목차

- AMI용 블록 디바이스 매핑 지정
- AMI 블록 디바이스 매핑에서 EBS 볼륨 보기

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.

Console

AMI에 볼륨을 추가하려면 다음을 수행하세요.

1. Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.

3. 인스턴스를 선택하고 [작업(Actions)], [이미지 및 템플릿(Image and templates)], [이미지 생성(Create image)]을 선택합니다.
4. 이미지의 이름과 설명을 입력합니다.
5. [인스턴스 볼륨(Instance volumes)] 아래에 인스턴스 볼륨이 나타납니다. 다른 볼륨을 추가하려면 [볼륨 추가(Add volume)]를 선택합니다.
6. [볼륨 유형(Volume type)]에서 볼륨 유형을 선택합니다. [디바이스(Device)]에서 디바이스 이름을 선택합니다. EBS 볼륨의 경우 스냅샷, 볼륨 크기, 볼륨 유형, IOPS 및 암호화 상태와 같은 추가 세부 정보를 지정할 수 있습니다.
7. 이미지 생성(Create image)을 선택합니다.

AWS CLI

AMI에 볼륨을 추가하려면 다음을 수행하세요.

create-image 명령을 사용하여 EBS 지원 AMI에 블록 디바이스 매핑을 지정합니다. register-image 명령을 사용하여 인스턴스 저장소 지원 AMI에 블록 디바이스 매핑을 지정합니다.

`--block-device-mappings` 파라미터를 사용하여 블록 디바이스 매핑을 지정합니다. 다음과 같이 JSON으로 인코딩된 인수를 명령줄에서 직접 지정하거나 JSON 파일을 참조하여 지정할 수 있습니다.

```
--block-device-mappings file://mapping.json
```

인스턴스 스토어 볼륨을 추가하려면 다음 매핑을 사용합니다. NVMe 인스턴스 저장소 볼륨은 자동으로 추가됩니다.

```
{
    "DeviceName": "device_name",
    "VirtualName": "ephemeral0"
}
```

비어 있는 100GiB 볼륨을 추가하려면 다음 매핑을 사용합니다.

```
{
    "DeviceName": "device_name",
    "Ebs": {
      "VolumeSize": 100
    }
```

```
}
```

스냅샷 기반 EBS 볼륨을 추가하려면 다음 매핑을 사용합니다.

```
{
    "DeviceName": "device_name",
    "Ebs": {
      "SnapshotId": "snap-1234567890abcdef0"
    }
}
```

디바이스에 대한 매핑을 생략하려면 다음 매핑을 사용합니다.

```
{
    "DeviceName": "device_name",
    "NoDevice": ""
}
```

PowerShell

New-EC2Image cmdlet을 사용하여 EBS 지원 AMI에 대한 블록 디바이스 매핑을 지정합니다. Register-EC2Image cmdlet을 사용하여 인스턴스 저장소 지원 AMI에 대한 블록 디바이스 매핑을 지정합니다.

bdm에 업데이트를 지정하여 `-BlockDeviceMapping` 옵션을 추가합니다.

```
-BlockDeviceMapping $bdm
```

다음 매핑은 스냅샷을 기반으로 볼륨을 추가합니다.

```
$ebd = New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice
$ebd.SnapshotId = "snap-1234567890abcdef0"
$bdm = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$bdm.DeviceName = "device_name"
$bdm.Ebs = $ebd
```

다음 매핑은 비어 있는 100GiB 볼륨을 추가합니다.

```
$ebd = New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice
$ebd.VolumeSize = 100
$bdm = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
```

```
$bdm.DeviceName = "device_name"
$bdm.Ebs = $ebd
```

AMI 블록 디바이스 매핑에서 EBS 볼륨 보기

AMI의 블록 디바이스 매핑에서 EBS 볼륨을 쉽게 확인할 수 있습니다.

Console

AMI용 EBS 볼륨을 보려면 다음을 수행하세요.

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)
   - 루트 디바이스 이름(예: /dev/sda1)
   - 블록 디바이스(예: /dev/sda1=snap-1234567890abcdef0:8:true)

   AMI가 블록 디바이스 매핑을 사용하여 추가 EBS 볼륨으로 생성된 경우 블록 디바이스 필드에 해당 추가 볼륨에 대한 매핑도 표시됩니다. (이 화면에는 인스턴스 스토어 볼륨이 표시되지 않습니다.)

AWS CLI

AMI용 EBS 볼륨을 보려면 다음을 수행하세요.

describe-images 명령을 사용합니다.

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query Image[0].BlockDeviceMappings
```

PowerShell

AMI용 EBS 볼륨을 보려면 다음을 수행하세요.

Get-EC2Image cmdlet을 사용합니다.

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).BlockDeviceMappings
```

Amazon EC2 인스턴스에 블록 디바이스 매핑 추가

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 사항

- 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.

업무

- 인스턴스를 시작할 때 블록 디바이스 매핑 업데이트
- 실행 중인 인스턴스의 블록 디바이스 매핑 업데이트
- 인스턴스 블록 디바이스 매핑에서 EBS 볼륨 보기
- 인스턴스 스토어 볼륨용 인스턴스 블록 디바이스 매핑 보기

인스턴스를 시작할 때 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 주의하세요.

Console

시작 시 인스턴스의 볼륨을 업데이트하는 방법

1. 절차에 따라 인스턴스를 시작하되 다음과 같은 단계를 완료하여 볼륨을 업데이트할 때까지는 인스턴스를 시작하지 마세요.

2. (선택 사항) 볼륨을 추가하려면 스토리지 구성, 새 볼륨 추가를 선택합니다. 볼륨 크기와 볼륨 유형을 선택합니다.
3. (선택 사항) AMI의 블록 디바이스 매핑에 따라 지정된 볼륨을 억제하려면 스토리지 구성, 제거를 선택합니다.
4. (선택 사항) EBS 볼륨의 구성을 수정하려면 스토리지 구성 창에서 고급을 선택합니다. 볼륨에 대한 정보를 확장하고 필요한 사항을 변경합니다.
5. Summary(요약) 패널에서 인스턴스 구성을 검토한 다음 Launch instance(인스턴스 시작)를 선택합니다. 자세한 내용은 콘솔에서 인스턴스 시작 마법사를 사용하여 EC2 인스턴스 시작 섹션을 참조하세요.

AWS CLI

시작 시 인스턴스의 볼륨을 업데이트하는 방법

run-instances 명령을 `--block-device-mappings` 옵션과 함께 사용합니다.

```
--block-device-mappings file://mapping.json
```

예를 들어, AMI 블록 디바이스 매핑에서 다음이 지정된다고 가정해 보겠습니다.

- `/dev/xvda` - EBS 루트 볼륨
- `/dev/sdh` - `snap-1234567890abcdef0`에서 생성된 EBS 볼륨
- `/dev/sdj` - 크기가 `100`인 비어 있는 EBS 볼륨
- `/dev/sdb` - 인스턴스 저장소 볼륨 `ephemeral0`

다음은 `mapping.json`의 인스턴스 블록 디바이스 매핑이라고 가정해 보겠습니다.

```
[
    {
        "DeviceName": "/dev/xvda",
        "Ebs": {
            "VolumeSize": 100
        }
    },
    {
        "DeviceName": "/dev/sdj",
        "NoDevice": ""
```

```
    },
    {
        "DeviceName": "/dev/sdh",
        "Ebs": {
            "VolumeSize": 300
        }
    },
    {
        "DeviceName": "/dev/sdc",
        "VirtualName": "ephemeral1"
    }
]
```

인스턴스 블록 디바이스 매핑에서 다음이 수행됩니다.

- 루트 볼륨(/dev/xvda)의 크기를 재정의하여 100GiB로 늘립니다.
- /dev/sdj가 인스턴스에 연결되지 않도록 합니다.
- /dev/sdh의 크기를 재정의하여 300GiB로 늘립니다. 스냅샷 ID는 다시 지정할 필요가 없습니다.
- 임시 볼륨(/dev/sdc)을 추가합니다. 다중 인스턴스 저장소 볼륨을 지원하지 않는 인스턴스 유형인 경우 임시 볼륨이 영향을 미치지 않습니다. 인스턴스 유형에서 NVMe 인스턴스 저장소 볼륨이 지원되는 경우 해당 볼륨은 자동으로 열거되고 인스턴스 블록 디바이스 매핑에 포함되며, 재정의할 수 없습니다.

PowerShell

시작 시 인스턴스의 볼륨을 업데이트하는 방법

-BlockDeviceMapping 파라미터가 있는 New-EC2Instance cmdlet과 함께 -BlockDeviceMapping 파라미터를 사용합니다.

```
-BlockDeviceMapping $bdm
```

다음은 $bdm의 인스턴스 블록 디바이스 매핑이라고 가정해 보겠습니다.

```
$bdm = @()

$root = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$root.DeviceName = "/dev/xvda"
$ebs1 = New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice
```

```
$ebs1.VolumeSize = 100
$root.Ebs = $ebs1
$bdm += $root

$sdj = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$sdj.DeviceName = "/dev/sdj"
$sdj.NoDevice = ""
$bdm += $sdj

$sdh = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$sdh.DeviceName = "/dev/sdh"
$ebs2 = New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice
$ebs2.VolumeSize = 300
$sdh.Ebs = $ebs2
$bdm += $sdh

$sdc = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$sdc.DeviceName = "/dev/sdc"
$sdc.VirtualName = "ephemeral1"
$bdm += $sdc
```

인스턴스 블록 디바이스 매핑에서 다음이 수행됩니다.

- 루트 볼륨(/dev/xvda)의 크기를 재정의하여 100GiB로 늘립니다.
- /dev/sdj가 인스턴스에 연결되지 않도록 합니다.
- /dev/sdh의 크기를 재정의하여 300GiB로 늘립니다. 스냅샷 ID는 다시 지정할 필요가 없습니다.
- 임시 볼륨(/dev/sdc)을 추가합니다. 다중 인스턴스 저장소 볼륨을 지원하지 않는 인스턴스 유형인 경우 임시 볼륨이 영향을 미치지 않습니다. 인스턴스 유형에서 NVMe 인스턴스 저장소 볼륨이 지원되는 경우 해당 볼륨은 자동으로 열거되고 인스턴스 블록 디바이스 매핑에 포함되며, 재정의할 수 없습니다.

실행 중인 인스턴스의 블록 디바이스 매핑 업데이트

이 속성을 변경하기 전에 인스턴스를 중지할 필요는 없습니다.

AWS CLI

실행 중인 인스턴스의 블록 디바이스 매핑을 업데이트하는 방법

modify-instance-attribute 명령을 사용합니다.

`--block-device-mappings` 옵션을 추가합니다.

```
--block-device-mappings file://mapping.json
```

`mapping.json`에서 업데이트를 지정합니다. 예를 들어, 다음 업데이트는 루트 디바이스 볼륨이 지속되도록 변경합니다.

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

PowerShell

실행 중인 인스턴스의 블록 디바이스 매핑을 업데이트하는 방법

Edit-EC2InstanceAttribute cmdlet을 사용합니다.

`-BlockDeviceMapping` 옵션을 추가합니다.

```
-BlockDeviceMapping $bdm
```

`bdm`에서 업데이트를 지정합니다. 예를 들어, 다음 업데이트는 루트 디바이스 볼륨이 지속되도록 변경합니다.

```
$ebd = New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice
$ebd.DeleteOnTermination = false
$bdm = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$bdm.DeviceName = "/dev/sda1"
$bdm.Ebs = $ebd
```

인스턴스 블록 디바이스 매핑에서 EBS 볼륨 보기

인스턴스에 매핑된 EBS 볼륨을 쉽게 확인할 수 있습니다.

Console

인스턴스용 EBS 볼륨을 보려면 다음을 수행하세요.

1. Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택한 후 스토리지 탭에 표시되는 세부 정보를 확인합니다. 루트 디바이스에서 최소한으로 사용 가능한 정보는 다음과 같습니다.

   - 루트 디바이스 유형(예: EBS)
   - 루트 디바이스 이름(예: /dev/xvda)
   - 블록 디바이스(예: /dev/xvda, /dev/sdf 및 /dev/sdj)

   인스턴스가 블록 디바이스 매핑을 사용하여 추가 EBS 볼륨으로 시작된 경우 [블록 디바이스(Block devices)] 아래에 나타납니다. 인스턴스 스토어 볼륨은 이 탭에 나타나지 않습니다.
4. EBS 볼륨에 대한 추가 정보를 표시하려면 볼륨 ID를 선택하여 볼륨 페이지로 이동합니다.

AWS CLI

인스턴스용 EBS 볼륨을 보려면 다음을 수행하세요.

아래와 같이 describe-instances 명령을 사용합니다.

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query Reservations[*].Instances[0].BlockDeviceMappings
```

PowerShell

인스턴스용 EBS 볼륨을 보려면 다음을 수행하세요.

Get-EC2Instance cmdlet을 사용합니다.

```
(Get-EC2Instance -InstanceId i-0bac57d7472c89bac).Instances.BlockDeviceMappings
```

인스턴스 스토어 볼륨용 인스턴스 블록 디바이스 매핑 보기

인스턴스 유형은 인스턴스에 사용 가능한 인스턴스 스토어 볼륨 수와 유형을 결정합니다. 블록 디바이스 매핑에 있는 인스턴스 스토어 볼륨 수가 인스턴스에 사용 가능한 인스턴스 스토어 볼륨 수를 초과한 경우 추가 볼륨이 무시됩니다. 인스턴스의 인스턴스 저장소 볼륨을 보려면 lsblk 명령을 실행하거나 (Linux 인스턴스) Windows 디스크 관리(Windows 인스턴스)를 엽니다. 각 인스턴스 유형에서 지원되는 인스턴스 스토어 볼륨의 수를 알아보려면 Amazon EC2 인스턴스 유형 사양을 참조하세요.

인스턴스에 대한 블록 디바이스 매핑을 볼 때 인스턴스 스토어 볼륨이 아닌 EBS 볼륨만 확인할 수 있습니다. 인스턴스의 인스턴스 스토어 볼륨을 보는 방법은 볼륨 유형에 따라 다릅니다.

NVMe 인스턴스 스토어 볼륨

Linux 인스턴스

NVMe 명령줄 패키지, nvme-cli를 사용하여 블록 디바이스 매핑의 NVMe 인스턴스 스토어 볼륨을 쿼리합니다. 인스턴스에 패키지를 다운로드하고 설치한 후 다음 명령을 실행합니다.

```
[ec2-user ~]$ sudo nvme list
```

다음은 인스턴스의 출력 예입니다. [모델(Model)] 열의 텍스트는 볼륨이 EBS 볼륨인지 아니면 인스턴스 스토어 볼륨인지를 나타냅니다. 이 예에서는 /dev/nvme1n1 및 /dev/nvme2n1이 모두 인스턴스 스토어 볼륨입니다.

```
Node             SN                   Model
 Namespace
---------------- -------------------- ----------------------------------------
 ---------
/dev/nvme0n1     vol06afc3f8715b7a597 Amazon Elastic Block Store               1

/dev/nvme1n1     AWS2C1436F5159EB6614 Amazon EC2 NVMe Instance Storage         1

/dev/nvme2n1     AWSB1F4FF0C0A6C281EA Amazon EC2 NVMe Instance Storage         1
  ...
```

Windows 인스턴스

디스크 관리 또는 PowerShell을 사용하여 EBS 및 인스턴스 스토어 NVMe 볼륨을 모두 나열할 수 있습니다. 자세한 내용은 the section called “볼륨에 NVME 디스크 매핑” 섹션을 참조하세요.

HDD 또는 SSD 인스턴스 스토어 볼륨

인스턴스 메타데이터를 사용하여 블록 디바이스 매핑에 있는 HDD 또는 SSD 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 메타데이터를 사용하여 EC2 인스턴스를 관리합니다. 섹션을 참조하세요.

Linux 인스턴스

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" 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 출력은 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" 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
```

Windows 인스턴스

우선, 실행 중인 인스턴스에 연결합니다. 인스턴스에서 이 쿼리를 사용하여 블록 디바이스 매핑을 가져옵니다.

```
PS C:\> Invoke-RestMethod -uri 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 출력은 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.

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/block-device-mapping/ephemeral0
```

Amazon EC2 Windows 인스턴스용 볼륨 연결 및 매핑 방법

> **Note**
>
> 이 주제는 Windows 인스턴스에만 적용됩니다.

Windows 인스턴스는 루트 볼륨의 기능을 하는 EBS 볼륨과 함께 제공됩니다. Windows 인스턴스가 AWS PV 또는 Citrix PV 드라이버를 사용하는 경우 최대 25개의 볼륨을 선택적으로 추가할 수 있으므로 총 볼륨은 26개가 됩니다. 자세한 내용은 Amazon EC2 인스턴스의 Amazon EBS 볼륨 한도 섹션을 참조하세요.

인스턴스의 인스턴스 유형에 따라 인스턴스는 0~24개의 인스턴스 스토어 볼륨을 가질 수 있습니다. 인스턴스에서 사용 가능한 인스턴스 스토어 볼륨을 사용하려면 AMI 생성 시 또는 인스턴스 시작 시에 인스턴스 스토어 볼륨을 지정해야 합니다. 또한 AMI를 생성하거나 인스턴스를 시작할 때 EBS 볼륨을 추가하거나, 인스턴스가 실행 중인 상태에서 해당 볼륨을 연결할 수 있습니다.

인스턴스에 볼륨을 추가할 때 Amazon EC2에서 사용하는 디바이스 이름을 지정합니다. 자세한 내용은 Amazon EC2 인스턴스의 볼륨 디바이스 이름 섹션을 참조하세요. AWS Windows Amazon Machine Image(AMI)에는 Amazon EC2가 인스턴스 스토어 및 EBS 볼륨을 Windows 디스크 및 드라이브 문자에 매핑하기 위해 사용하는 드라이버 세트가 포함됩니다.

디스크를 EBS 볼륨에 매핑하는 방법

- 볼륨에 Amazon EC2 Windows 인스턴스의 NVMe 디스크 매핑
- 볼륨에 Amazon EC2 Windows 인스턴스의 비 NVMe 디스크 매핑

볼륨에 Amazon EC2 Windows 인스턴스의 NVMe 디스크 매핑

Nitro 기반 인스턴스를 사용하면, EBS 볼륨은 NVMe 디바이스로 표시됩니다. 이 주제에서는 인스턴스의 Windows 운영 체제에서 사용할 수 있는 NVMe 디스크를 보는 방법을 설명합니다. 또한 이러한 NVMe 디스크를 기본 Amazon EBS 볼륨에 매핑하는 방법과 Amazon EC2에서 사용하는 블록 디바이스 매핑에 지정된 디바이스 이름을 보여줍니다.

주제

- NVMe 디스크 나열
- 볼륨에 NVMe 디스크 매핑

NVMe 디스크 나열

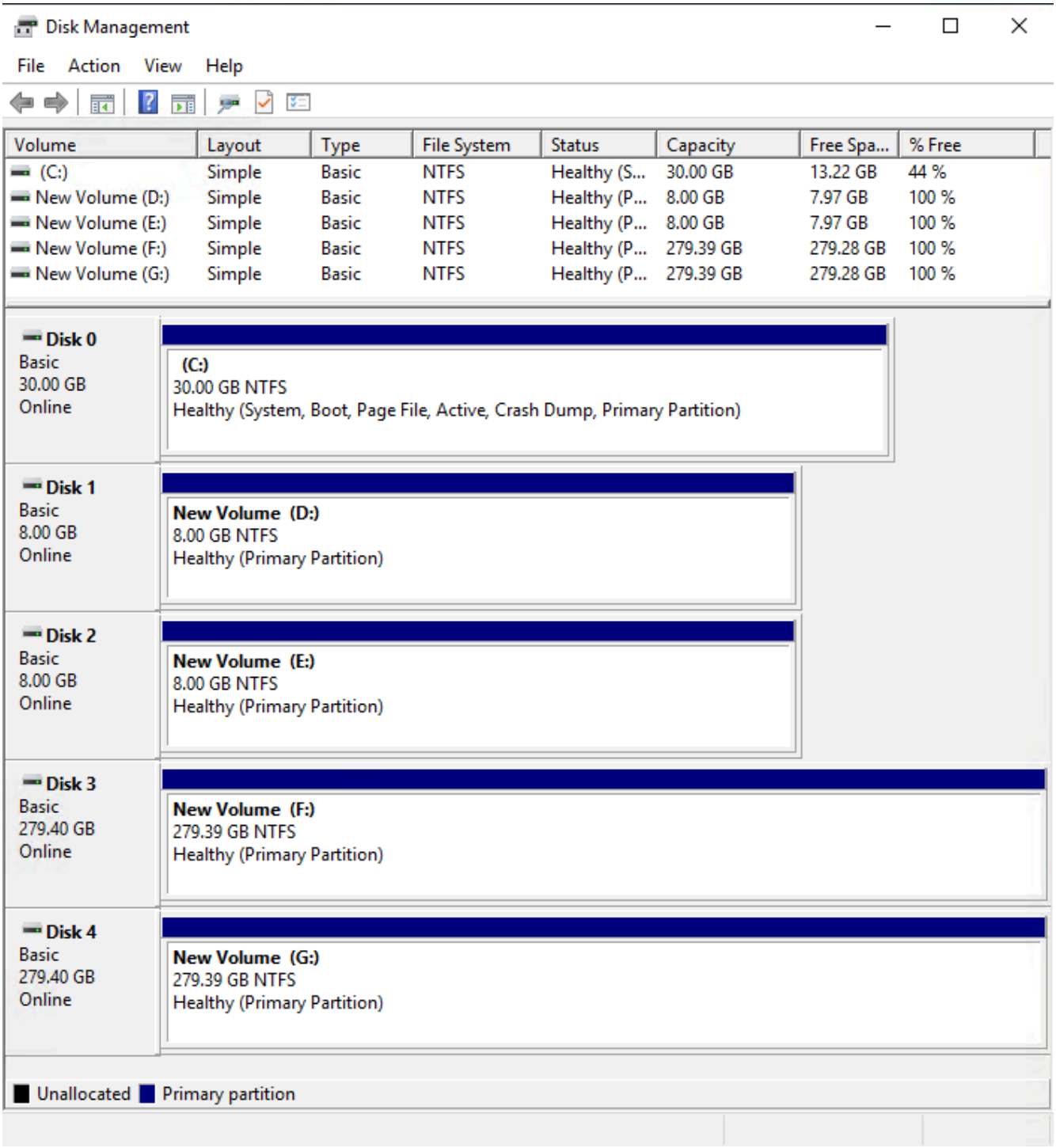
디스크 관리 또는 Powershell을 사용하여 Windows 인스턴스에 있는 디스크를 검색할 수 있습니다.

Disk Management

Windows 인스턴스에 있는 디스크를 검색하려면

1. 원격 데스크톱을 사용하여 Windows 인스턴스에 로그인합니다. 자세한 내용은 RDP를 사용하여 Windows 인스턴스에 연결 섹션을 참조하세요.
2. 디스크 관리 유틸리티를 시작합니다.
3. 디스크를 확인합니다. 루트 볼륨은 `C:\`로 마운트되는 EBS 볼륨입니다. 다른 디스크가 표시되지 않는 경우 AMI를 생성하거나 인스턴스를 시작할 때 추가 볼륨을 지정하지 않은 것입니다.

   다음 예는 추가 EBS 볼륨 2개가 포함된 `r5d.4xlarge` 인스턴스를 시작하는 경우 사용 가능한 디스크를 보여줍니다.



PowerShell

다음 PowerShell 스크립트는 각 디스크 및 해당 디바이스 이름과 볼륨을 목록으로 표시합니다. NVMe EBS 및 인스턴스 스토어 볼륨을 사용하는 Nitro 기반 인스턴스와 함께 사용하기 위한 것입니다.

Windows 인스턴스에 연결하고 다음 명령을 실행하여 PowerShell 스크립트 실행을 활성화합니다.

```
Set-ExecutionPolicy RemoteSigned
```

다음 스크립트를 복사하여 Windows 인스턴스에 `mapping.ps1`로 저장합니다.

```
# List the disks for NVMe volumes

function Get-EC2InstanceMetadata {
    param([string]$Path)
    (Invoke-WebRequest -Uri "http://169.254.169.254/latest/$Path").Content
}

function GetEBSVolumeId {
    param($Path)
    $SerialNumber = (Get-Disk -Path $Path).SerialNumber
    if($SerialNumber -clike 'vol*'){
        $EbsVolumeId = $SerialNumber.Substring(0,20).Replace("vol","vol-")
    }
    else {
       $EbsVolumeId = $SerialNumber.Substring(0,20).Replace("AWS","AWS-")
    }
    return $EbsVolumeId
}

function GetDeviceName{
    param($EbsVolumeId)
    if($EbsVolumeId -clike 'vol*'){

        $Device  = ((Get-EC2Volume -VolumeId $EbsVolumeId ).Attachment).Device
        $VolumeName = ""
    }
     else {
        $Device = "Ephemeral"
        $VolumeName = "Temporary Storage"
    }
    Return $Device,$VolumeName
}

function GetDriveLetter{
    param($Path)
    $DiskNumber =  (Get-Disk -Path $Path).Number
    if($DiskNumber -eq 0){
```

```
        $VirtualDevice = "root"
        $DriveLetter = "C"
        $PartitionNumber = (Get-Partition -DriveLetter C).PartitionNumber
    }
    else
    {
        $VirtualDevice = "N/A"
        $DriveLetter = (Get-Partition -DiskNumber $DiskNumber).DriveLetter
        if(!$DriveLetter)
        {
            $DriveLetter = ((Get-Partition -DiskId $Path).AccessPaths).Split(",")[0]
        }
        $PartitionNumber = (Get-Partition -DiskId $Path).PartitionNumber
    }

    return $DriveLetter,$VirtualDevice,$PartitionNumber

}

$Report = @()
foreach($Path in (Get-Disk).Path)
{
    $Disk_ID = ( Get-Partition -DiskId $Path).DiskId
    $Disk = ( Get-Disk -Path $Path).Number
    $EbsVolumeId  = GetEBSVolumeId($Path)
    $Size =(Get-Disk -Path $Path).Size
    $DriveLetter,$VirtualDevice, $Partition = (GetDriveLetter($Path))
    $Device,$VolumeName = GetDeviceName($EbsVolumeId)
    $Disk = New-Object PSObject -Property @{
      Disk          = $Disk
      Partitions    = $Partition
      DriveLetter   = $DriveLetter
      EbsVolumeId   = $EbsVolumeId
      Device        = $Device
      VirtualDevice = $VirtualDevice
      VolumeName= $VolumeName
    }
 $Report += $Disk
}

$Report | Sort-Object Disk | Format-Table -AutoSize -Property Disk, Partitions,
 DriveLetter, EbsVolumeId, Device, VirtualDevice, VolumeName
```

스크립트를 다음과 같이 실행합니다.

```
PS C:\> .\mapping.ps1
```

다음은 루트 볼륨 1개, EBS 볼륨 2개 및 인스턴스 스토어 볼륨 2개가 있는 인스턴스의 예제 출력입니다.

```
Disk Partitions DriveLetter EbsVolumeId           Device    VirtualDevice VolumeName
---- ---------- ----------- -----------           ------    ------------- ----------
   0          1 C           vol-03683f1d861744bc7 /dev/sda1 root
   1          1 D           vol-082b07051043174b9 xvdb      N/A
   2          1 E           vol-0a4064b39e5f534a2 xvdc      N/A
   3          1 F           AWS-6AAD8C2AEEE1193F0 Ephemeral N/A           Temporary
 Storage
   4          1 G           AWS-13E7299C2BD031A28 Ephemeral N/A           Temporary
 Storage
```

Windows 인스턴스에서 Windows PowerShell용 도구에 대한 자격 증명을 구성하지 않은 경우 스크립트는 EBS 볼륨 ID를 가져올 수 없으며 `EbsVolumeId` 열에 N/A를 사용합니다.

볼륨에 NVMe 디스크 매핑

Get-Disk 명령을 써서 Windows 디스크 번호를 EBS 볼륨 ID로 매핑할 수 있습니다.

```
PS C:\> Get-Disk
Number Friendly Name Serial Number                    HealthStatus
 OperationalStatus      Total Size Partition

                      Style
------ ------------- -------------                    ------------
 -----------------      ---------- ----------
3      NVMe Amazo... AWS6AAD8C2AEEE1193F0_00000001.   Healthy              Online
              279.4 GB MBR
4      NVMe Amazo... AWS13E7299C2BD031A28_00000001.   Healthy              Online
              279.4 GB MBR
2      NVMe Amazo... vol0a4064b39e5f534a2_00000001.   Healthy              Online
                  8 GB MBR
0      NVMe Amazo... vol03683f1d861744bc7_00000001.   Healthy              Online
                 30 GB MBR
1      NVMe Amazo... vol082b07051043174b9_00000001.   Healthy              Online
                  8 GB MBR
```

ebsnvme-id 명령을 실행하여 EBS 볼륨 ID 및 디바이스 이름에 NVMe 디스크 번호를 매핑할 수도 있습니다.

```
PS C:\> C:\PROGRAMDATA\Amazon\Tools\ebsnvme-id.exe
Disk Number: 0
Volume ID: vol-03683f1d861744bc7
Device Name: sda1

Disk Number: 1
Volume ID: vol-082b07051043174b9
Device Name: xvdb

Disk Number: 2
Volume ID: vol-0a4064b39e5f534a2
Device Name: xvdc
```

볼륨에 Amazon EC2 Windows 인스턴스의 비 NVMe 디스크 매핑

AWS PV 또는 Citrix PV 드라이버를 사용하는 Windows AMI에서 시작된 인스턴스의 경우, 이 페이지에서 설명된 관계를 활용하여 Windows 디스크를 인스턴스 스토어 및 EBS 볼륨에 매핑할 수 있습니다. 이 주제에서는 인스턴스의 Windows 운영 체제에서 사용할 수 있는 비 NVMe 디스크를 보는 방법을 설명합니다. 또한 이러한 비 NVMe 디스크를 기본 Amazon EBS 볼륨에 매핑하는 방법과 Amazon EC2에서 사용하는 블록 디바이스 매핑에 지정된 디바이스 이름을 보여줍니다.

> **Note**
>
> 인스턴스 시작 시 Windows AMI가 Red Hat PV를 사용하는 경우 인스턴스를 업데이트하면 Citrix 드라이버를 사용할 수 있습니다. 자세한 내용은 the section called “PV 드라이버 업그레이드” 단원을 참조하십시오.

주제

- 비 NVMe 디스크 나열
- 볼륨에 비 NVMe 디스크 매핑

비 NVMe 디스크 나열

디스크 관리 또는 PowerShell을 사용하여 Windows 인스턴스에 있는 디스크를 검색할 수 있습니다.

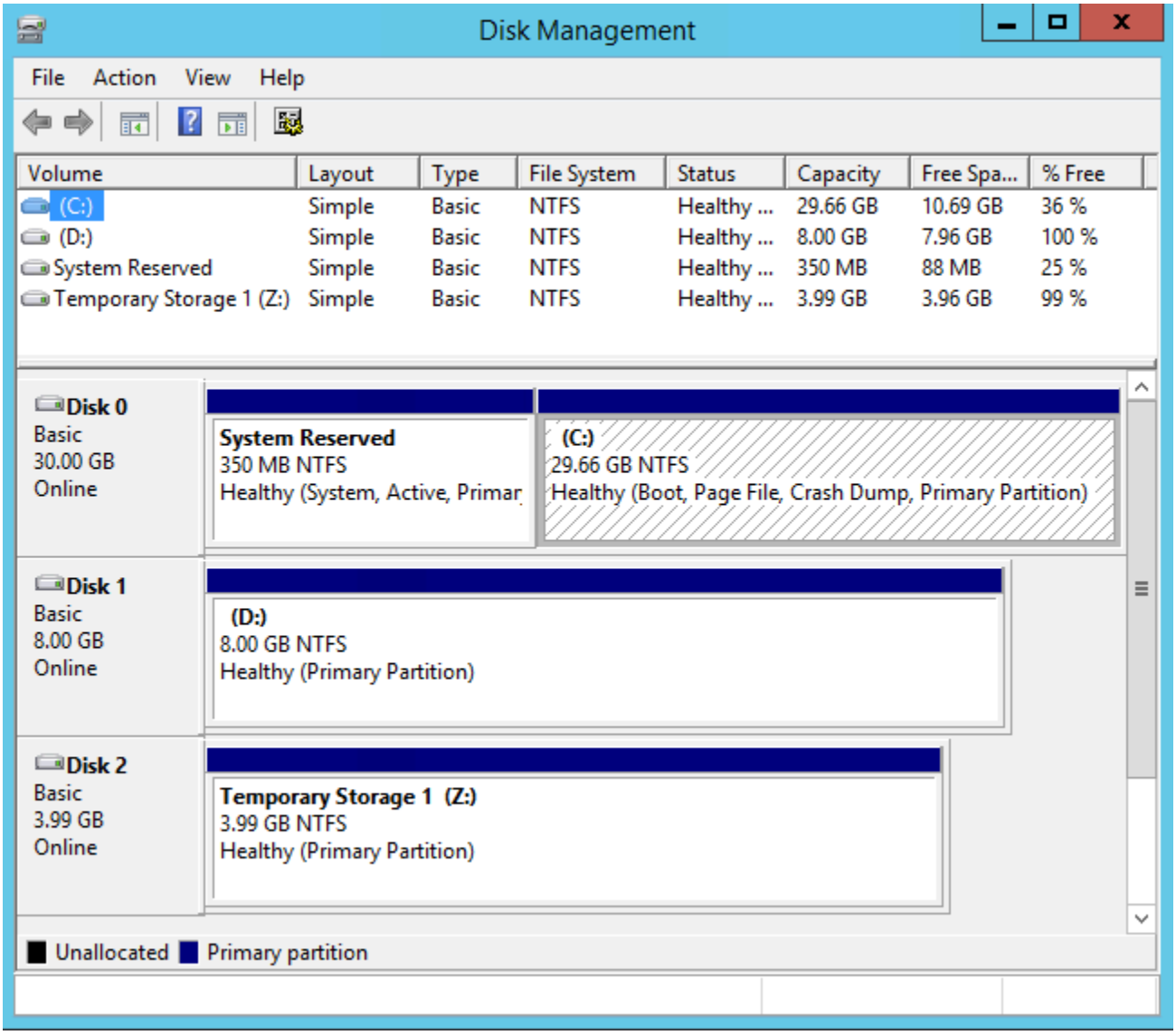
Disk Management

Windows 인스턴스에 있는 디스크를 검색하려면

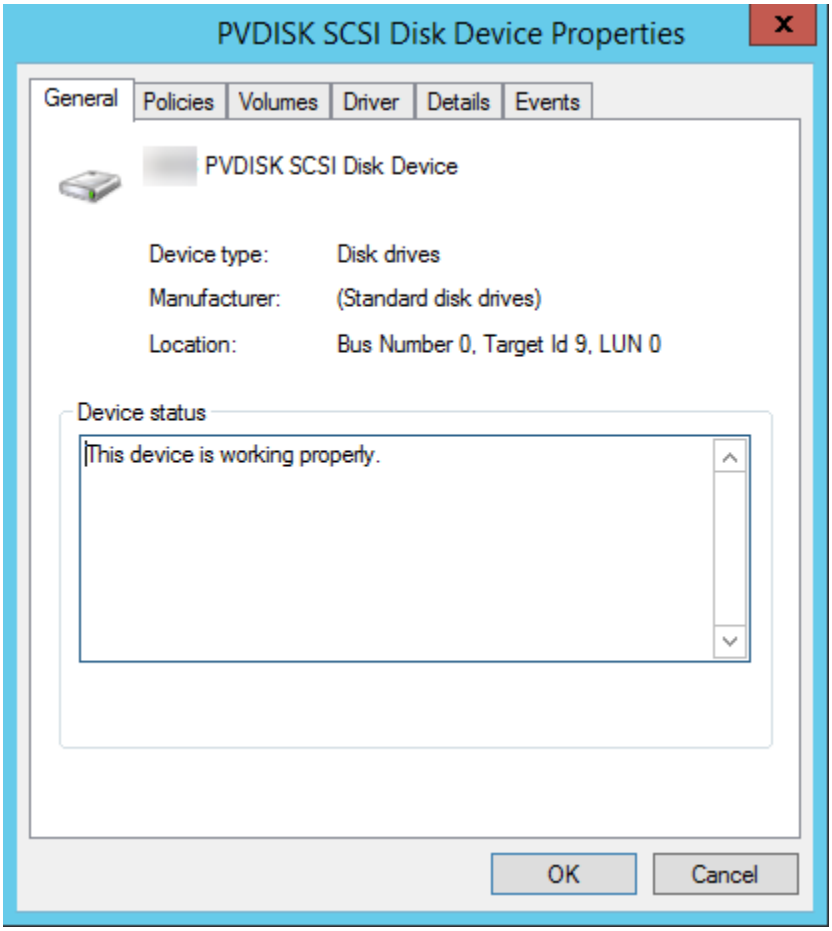
1. 원격 데스크톱을 사용하여 Windows 인스턴스에 로그인합니다. 자세한 내용은 RDP를 사용하여 Windows 인스턴스에 연결 섹션을 참조하세요.
2. 디스크 관리 유틸리티를 시작합니다.

   작업 표시줄에서 Windows 로고를 마우스 오른쪽 단추를 클릭한 다음 디스크 관리를 선택합니다.
3. 디스크를 확인합니다. 루트 볼륨은 `C:\`로 마운트되는 EBS 볼륨입니다. 다른 디스크가 표시되지 않는 경우 AMI를 생성하거나 인스턴스를 시작할 때 추가 볼륨을 지정하지 않은 것입니다.

   다음 예는 인스턴스 스토어 볼륨(디스크 2) 및 추가 EBS 볼륨(디스크 1)을 포함하여 `m3.medium` 인스턴스를 시작할 경우 사용 가능한 디스크를 보여줍니다.



4. 회색 창으로 표시된 디스크 1을 마우스 오른쪽 단추로 클릭한 후 속성을 선택합니다. 위치의 값을 볼륨에 비 NVMe 디스크 매핑의 표를 확인합니다. 예를 들어, 다음 디스크에는 버스 번호 0, 타겟 ID 9, LUN 0 위치가 있습니다. EBS 볼륨 테이블에 따르면 이 위치에 있는 디바이스 이름은 `xvdj`입니다.



PowerShell

다음 PowerShell 스크립트는 각 디스크 및 해당 디바이스 이름과 볼륨을 목록으로 표시합니다.

요구 사항 및 제한

- Windows Server 2012 이상이 필요합니다.
- EBS 볼륨 ID를 가져오려면 자격 증명이 필요합니다. Tools for PowerShell을 사용하여 프로필을 구성하거나 IAM 역할을 인스턴스에 연결할 수 있습니다.
- NVMe 볼륨을 지원하지 않습니다.
- 동적 디스크를 지원하지 않습니다.

Windows 인스턴스에 연결하고 다음 명령을 실행하여 PowerShell 스크립트 실행을 활성화합니다.

```
Set-ExecutionPolicy RemoteSigned
```

다음 스크립트를 복사하여 Windows 인스턴스에 mapping.ps1로 저장합니다.

```
# List the disks
function Convert-SCSITargetIdToDeviceName {
  param([int]$SCSITargetId)
  If ($SCSITargetId -eq 0) {
    return "sda1"
  }
  $deviceName = "xvd"
  If ($SCSITargetId -gt 25) {
    $deviceName += [char](0x60 + [int]($SCSITargetId / 26))
  }
  $deviceName += [char](0x61 + $SCSITargetId % 26)
  return $deviceName
}

[string[]]$array1 = @()
[string[]]$array2 = @()
[string[]]$array3 = @()
[string[]]$array4 = @()

Get-WmiObject Win32_Volume | Select-Object Name, DeviceID | ForEach-Object {
  $array1 += $_.Name
  $array2 += $_.DeviceID
}

$i = 0
While ($i -ne ($array2.Count)) {
  $array3 += ((Get-Volume -Path $array2[$i] | Get-Partition | Get-
Disk).SerialNumber) -replace "_[^ ]*$" -replace "vol", "vol-"
  $array4 += ((Get-Volume -Path $array2[$i] | Get-Partition | Get-
Disk).FriendlyName)
  $i ++
}

[array[]]$array = $array1, $array2, $array3, $array4

Try {
  $InstanceId = Get-EC2InstanceMetadata -Category "InstanceId"
  $Region = Get-EC2InstanceMetadata -Category "Region" | Select-Object -
ExpandProperty SystemName
}
Catch {
```

```
  Write-Host "Could not access the instance Metadata using AWS Get-
EC2InstanceMetadata CMDLet.
Verify you have AWSPowershell SDK version '3.1.73.0' or greater installed and
 Metadata is enabled for this instance." -ForegroundColor Yellow
}
Try {
  $BlockDeviceMappings = (Get-EC2Instance -Region $Region -Instance
 $InstanceId).Instances.BlockDeviceMappings
  $VirtualDeviceMap = (Get-EC2InstanceMetadata -Category
 "BlockDeviceMapping").GetEnumerator() | Where-Object { $_.Key -ne "ami" }
}
Catch {
  Write-Host "Could not access the AWS API, therefore, VolumeId is not available.
Verify that you provided your access keys or assigned an IAM role with adequate
 permissions." -ForegroundColor Yellow
}

Get-disk | ForEach-Object {
  $DriveLetter = $null
  $VolumeName = $null
  $VirtualDevice = $null
  $DeviceName = $_.FriendlyName

  $DiskDrive = $_
  $Disk = $_.Number
  $Partitions = $_.NumberOfPartitions
  $EbsVolumeID = $_.SerialNumber -replace "_[^ ]*$" -replace "vol", "vol-"
  if ($Partitions -ge 1) {
    $PartitionsData = Get-Partition -DiskId $_.Path
    $DriveLetter = $PartitionsData.DriveLetter | Where-object { $_ -notin @("",
 $null) }
    $VolumeName = (Get-PSDrive | Where-Object { $_.Name -in
 @($DriveLetter) }).Description | Where-object { $_ -notin @("", $null) }
  }
  If ($DiskDrive.path -like "*PROD_PVDISK*") {
    $BlockDeviceName = Convert-SCSITargetIdToDeviceName((Get-WmiObject -Class
 Win32_Diskdrive | Where-Object { $_.DeviceID -eq ("\\.\PHYSICALDRIVE" +
 $DiskDrive.Number) }).SCSITargetId)
    $BlockDeviceName = "/dev/" + $BlockDeviceName
    $BlockDevice = $BlockDeviceMappings | Where-Object { $BlockDeviceName -like "*"
 + $_.DeviceName + "*" }
    $EbsVolumeID = $BlockDevice.Ebs.VolumeId
    $VirtualDevice = ($VirtualDeviceMap.GetEnumerator() | Where-Object { $_.Value -
eq $BlockDeviceName }).Key | Select-Object -First 1
```

```
  }
  ElseIf ($DiskDrive.path -like "*PROD_AMAZON_EC2_NVME*") {
    $BlockDeviceName = (Get-EC2InstanceMetadata -Category
 "BlockDeviceMapping")."ephemeral$((Get-WmiObject -Class Win32_Diskdrive | Where-
Object { $_.DeviceID -eq ("\\.\PHYSICALDRIVE" + $DiskDrive.Number) }).SCSIPort - 2)"
    $BlockDevice = $null
    $VirtualDevice = ($VirtualDeviceMap.GetEnumerator() | Where-Object { $_.Value -
eq $BlockDeviceName }).Key | Select-Object -First 1
  }
  ElseIf ($DiskDrive.path -like "*PROD_AMAZON*") {
    if ($DriveLetter -match '[^a-zA-Z0-9]') {
      $i = 0
      While ($i -ne ($array3.Count)) {
        if ($array[2][$i] -eq $EbsVolumeID) {
          $DriveLetter = $array[0][$i]
          $DeviceName = $array[3][$i]
        }
        $i ++
      }
    }
    $BlockDevice = ""
    $BlockDeviceName = ($BlockDeviceMappings | Where-Object { $_.ebs.VolumeId -eq
 $EbsVolumeID }).DeviceName
  }
  ElseIf ($DiskDrive.path -like "*NETAPP*") {
    if ($DriveLetter -match '[^a-zA-Z0-9]') {
      $i = 0
      While ($i -ne ($array3.Count)) {
        if ($array[2][$i] -eq $EbsVolumeID) {
          $DriveLetter = $array[0][$i]
          $DeviceName = $array[3][$i]
        }
        $i ++
      }
    }
    $EbsVolumeID = "FSxN Volume"
    $BlockDevice = ""
    $BlockDeviceName = ($BlockDeviceMappings | Where-Object { $_.ebs.VolumeId -eq
 $EbsVolumeID }).DeviceName
  }
  Else {
    $BlockDeviceName = $null
    $BlockDevice = $null
  }
```

```
  New-Object PSObject -Property @{
    Disk          = $Disk;
    Partitions    = $Partitions;
    DriveLetter   = If ($DriveLetter -eq $null) { "N/A" } Else { $DriveLetter };
    EbsVolumeId   = If ($EbsVolumeID -eq $null) { "N/A" } Else { $EbsVolumeID };
    Device        = If ($BlockDeviceName -eq $null) { "N/A" } Else
 { $BlockDeviceName };
    VirtualDevice = If ($VirtualDevice -eq $null) { "N/A" } Else { $VirtualDevice };
    VolumeName    = If ($VolumeName -eq $null) { "N/A" } Else { $VolumeName };
    DeviceName    = If ($DeviceName -eq $null) { "N/A" } Else { $DeviceName };
  }
} | Sort-Object Disk | Format-Table -AutoSize -Property Disk, Partitions,
 DriveLetter, EbsVolumeId, Device, VirtualDevice, DeviceName, VolumeName
```

스크립트를 다음과 같이 실행합니다.

```
PS C:\> .\mapping.ps1
```

다음은 예제 출력입니다.

```
Disk  Partitions  DriveLetter   EbsVolumeId             Device      VirtualDevice
 DeviceName              VolumeName
----  ----------  -----------   -----------             ------      -------------
 ----------              ----------
   0           1            C   vol-0561f1783298efedd   /dev/sda1   N/A
 NVMe Amazon Elastic B   N/A
   1           1            D   vol-002a9488504c5e35a   xvdb        N/A
 NVMe Amazon Elastic B   N/A
   2           1            E   vol-0de9d46fcc907925d   xvdc        N/A
 NVMe Amazon Elastic B   N/A
```

Windows 인스턴스에서 자격 증명을 제공하지 않은 경우 스크립트는 EBS 볼륨 ID를 가져올 수 없으며 EbsVolumeId 열에 N/A가 표시됩니다.

볼륨에 비 NVMe 디스크 매핑

볼륨을 마운트할 때 인스턴스용 블록 디바이스 드라이버가 실제 볼륨 이름을 할당합니다.

매핑

- 인스턴스 스토어 볼륨

- EBS 볼륨

인스턴스 스토어 볼륨

다음 표는 Citrix PV 및 AWS PV 드라이버가 NVMe가 아닌 인스턴스 스토어 볼륨을 Windows 볼륨에 매핑하는 방법을 설명합니다. 사용할 수 있는 인스턴스 스토어 볼륨의 개수는 인스턴스 유형에 의해 결정됩니다. 자세한 내용은 EC2 인스턴스의 인스턴스 스토어 볼륨 제한 섹션을 참조하세요.

위치	디바이스 이름
버스 번호 0, 타겟 ID 78, LUN 0	xvdca
버스 번호 0, 타겟 ID 79, LUN 0	xvdcb
버스 번호 0, 타겟 ID 80, LUN 0	xvdcc
버스 번호 0, 타겟 ID 81, LUN 0	xvdcd
버스 번호 0, 타겟 ID 82, LUN 0	xvdce
버스 번호 0, 타겟 ID 83, LUN 0	xvdcf
버스 번호 0, 타겟 ID 84, LUN 0	xvdcg
버스 번호 0, 타겟 ID 85, LUN 0	xvdch
버스 번호 0, 타겟 ID 86, LUN 0	xvdci
버스 번호 0, 타겟 ID 87, LUN 0	xvdcj
버스 번호 0, 타겟 ID 88, LUN 0	xvdck
버스 번호 0, 타겟 ID 89, LUN 0	xvdcl

EBS 볼륨

다음 표는 Citrix PV 및 AWS PV 드라이버가 비 NVME EBS 볼륨을 Windows 볼륨에 매핑하는 방법을 설명합니다.

위치	디바이스 이름
버스 번호 0, 타겟 ID 0, LUN 0	/dev/sda1
버스 번호 0, 타겟 ID 1, LUN 0	xvdb
버스 번호 0, 타겟 ID 2, LUN 0	xvdc
버스 번호 0, 타겟 ID 3, LUN 0	xvdd
버스 번호 0, 타겟 ID 4, LUN 0	xvde
버스 번호 0, 타겟 ID 5, LUN 0	xvdf
버스 번호 0, 타겟 ID 6, LUN 0	xvdg
버스 번호 0, 타겟 ID 7, LUN 0	xvdh
버스 번호 0, 타겟 ID 8, LUN 0	xvdi
버스 번호 0, 타겟 ID 9, LUN 0	xvdj
버스 번호 0, 타겟 ID 10, LUN 0	xvdk
버스 번호 0, 타겟 ID 11, LUN 0	xvdl
버스 번호 0, 타겟 ID 12, LUN 0	xvdm
버스 번호 0, 타겟 ID 13, LUN 0	xvdn
버스 번호 0, 타겟 ID 14, LUN 0	xvdo
버스 번호 0, 타겟 ID 15, LUN 0	xvdp
버스 번호 0, 타겟 ID 16, LUN 0	xvdq
버스 번호 0, 타겟 ID 17, LUN 0	xvdr
버스 번호 0, 타겟 ID 18, LUN 0	xvds
버스 번호 0, 타겟 ID 19, LUN 0	xvdt

위치	디바이스 이름
버스 번호 0, 타겟 ID 20, LUN 0	xvdu
버스 번호 0, 타겟 ID 21, LUN 0	xvdv
버스 번호 0, 타겟 ID 22, LUN 0	xvdw
버스 번호 0, 타겟 ID 23, LUN 0	xvdx
버스 번호 0, 타겟 ID 24, LUN 0	xvdy
버스 번호 0, 타겟 ID 25, LUN 0	xvdz

Amazon EC2 Linux 인스턴스에서 손상된 쓰기 방지

> **Note**
>
> 찢어진 쓰기 방지는 Linux 인스턴스에서만 지원됩니다.

찢긴 쓰기 방지는 데이터 복원력에 부정적인 영향을 미치지 않으면서 I/O 집약적인 관계형 데이터베이스 워크로드의 성능을 개선하고 지연 시간을 줄이도록 설계된 AWS의 블록 스토리지 기능입니다. MySQL, MariaDB 등, InnoDB 또는 XtraDB를 데이터베이스 엔진으로 사용하는 관계형 데이터베이스는 찢긴 쓰기 방지 기능의 이점을 누릴 수 있습니다.

일반적으로 스토리지 디바이스의 전원 장애 원자성보다 큰 페이지를 사용하는 관계형 데이터베이스는 데이터 로깅 메커니즘을 사용하여 쓰기가 끊어지지 않도록 보호합니다. MariaDB와 MySQL은 데이터 테이블에 데이터를 쓰기 전에 이중 쓰기 버퍼 파일을 사용하여 데이터를 기록합니다. 쓰기 트랜잭션 도중 발생하는 운영 체제 충돌이나 전원 손실로 인해 쓰기가 불완전하거나 손상된 경우, 데이터베이스는 이중 쓰기 버퍼에서 데이터를 복구할 수 있습니다. 이중 쓰기 버퍼에 쓰는 것과 관련한 추가 I/O 오버헤드는 데이터베이스 성능과 애플리케이션 지연 시간에 영향을 미치며 초당 처리할 수 있는 트랜잭션 수를 줄입니다. 이중 쓰기 버퍼에 대한 자세한 내용은 MariaDB 및 MySQL 설명서를 참조하세요.

찢긴 쓰기 방지 기능을 사용하면 데이터가 전부 또는 전무 쓰기 트랜잭션을 통해 스토리지에 쓰여지므로 이중 쓰기 버퍼를 사용할 필요가 없습니다. 따라서 쓰기 트랜잭션 중에 운영 체제가 충돌하거나 전원이 유실될 경우 데이터의 일부 또는 손상된 데이터가 스토리지에 기록되지 않습니다. 워크로드의 복

원력을 저하시키지 않으면서 초당 처리되는 트랜잭션 수를 최대 30% 늘리고, 쓰기 지연 시간을 최대 50% 줄일 수 있습니다.

요금

찢긴 쓰기 방지 기능을 사용하는 데 따른 추가 비용은 없습니다.

내용

- Amazon EC2에서 손상된 쓰기 방지를 위한 블록 크기
- Amazon EC2에서 손상된 쓰기 방지를 사용하기 위한 요구 사항
- 손상된 쓰기 방지를 위한 Amazon EC2 인스턴스 지원 확인
- 손상된 쓰기 방지를 위해 Amazon EC2의 워크로드 구성

Amazon EC2에서 손상된 쓰기 방지를 위한 블록 크기

찢긴 쓰기 방지 기능은 4KiB, 8KiB 및 16KiB 데이터 블록에 대한 쓰기 작업을 지원합니다. 데이터 블록 시작 논리적 블록 주소(LBA)는 4KiB, 8KiB 또는 16KiB의 각 블록 경계 크기에 맞게 정렬되어야 합니다. 예를 들어 16KiB 쓰기 작업의 경우 데이터 블록 시작 LBA를 16KiB의 블록 경계 크기로 정렬해야 합니다.

다음 표에는 스토리지 및 인스턴스 유형별로 지원되는 블록 크기가 나와 있습니다.

	4KiB 블록	8KiB 블록	16KiB 블록
인스턴스 스토어 볼륨	모든 NVMe 인스턴스 스토어 볼륨은 현재 세대 I-패밀리 인스턴스에 연결됩니다.	AWS Nitro SSD에서 지원하는 I4i, Im4gn, Is4gen, I7i, I7ie 및 I8i 인스턴스	
Amazon EBS 볼륨	모든 Amazon EBS 볼륨은 Nitro 기반 인스턴스에 연결됩니다.		

인스턴스와 볼륨이 찢긴 쓰기 방지 기능을 지원하는지 확인하려면, 쿼리를 통해 해당 인스턴스가 찢긴 쓰기 방지 및 기타 세부 정보(예: 지원되는 블록 및 경계 크기)를 지원하는지 확인합니다. 자세한 내용은 손상된 쓰기 방지를 위한 Amazon EC2 인스턴스 지원 확인 섹션을 참조하세요.

Amazon EC2에서 손상된 쓰기 방지를 사용하기 위한 요구 사항

찢긴 쓰기 방지 기능이 제대로 작동하려면 I/O 작업이 NTWPU, NTWGU, NTWBU 필드에 지정된 크기, 정렬 및 경계 요구 사항을 충족해야 합니다. 디바이스에 제출되기 전에, 특정 스토리지 하위 시스템(파일 시스템, LVM, RAID 등)이 블록 병합, 분할 또는 블록 주소 재배치를 비롯한 스토리지 스택의 I/O 속성을 수정하지 않도록 운영 체제를 구성해야 합니다.

찢긴 쓰기 방지는 다음 구성을 사용하여 테스트되었습니다.

- 필요한 블록 크기를 지원하는 인스턴스 유형 및 스토리지 유형
- 커널 버전 5.10 이상을 사용하는 Amazon Linux 2
- `bigalloc`이 활성화되어 있고 클러스터 크기가 16KiB인 ext4와 최신 ext4 유틸리티(e2fsprogs 1.46.5 이상)
- Linux 커널 버퍼 캐시를 우회하는 `O_DIRECT` 파일 액세스 모드

> **Note**
>
> MySQL 및 MariaDB 워크로드에 대해 I/O 병합을 비활성화할 필요는 없습니다.

손상된 쓰기 방지를 위한 Amazon EC2 인스턴스 지원 확인

인스턴스와 볼륨이 찢긴 쓰기 방지를 지원하는지 확인하고 찢긴 쓰기 방지 정보가 포함된 NVMe 네임스페이스 공급업체별 데이터를 보려면 다음 명령을 사용합니다.

```
$ sudo nvme id-ns -v device_name
```

> **Note**
>
> 이 명령은 공급업체별 정보를 ASCII 해석과 함께 16진수로 반환합니다. 출력을 읽고 구문 분석할 수 있는 `ebsnvme-id`와 유사한 도구를 애플리케이션에 빌드해야 할 수도 있습니다.

예를 들어 다음 명령은 `/dev/nvme1n1`의 찢긴 쓰기 방지 정보가 포함된 NVMe 네임스페이스 공급업체별 데이터를 반환합니다.

```
$ sudo nvme id-ns -v /dev/nvme1n1
```

인스턴스와 볼륨이 찢긴 쓰기 방지를 지원하는 경우, NVMe 네임스페이스 공급업체별 데이터에 다음과 같은 AWS 찢긴 쓰기 방지 정보가 반환됩니다.

> **Note**
>
> 다음 표의 바이트는 NVMe 네임스페이스 공급업체별 데이터의 시작 부분으로부터의 오프셋(바이트)을 나타냅니다.

바이트	설명
0:31	디바이스 연결 마운트 지점의 이름(예: /dev/xvda) 볼륨 연결 요청 시에 이 정보를 제공하면 Amazon EC2 인스턴스에서 이 정보를 사용하여 NVMe 블록 디바이스(nvmeXn1)의 심볼릭 링크를 생성할 수 있습니다.
32:63	볼륨 ID 예를 들어 vol01234567890abcdef 입니다. 이 필드는 NVMe 디바이스를 연결된 볼륨에 매핑하는 데 사용할 수 있습니다.
64:255	추후 사용 예약.
256:257	네임스페이스 찢긴 쓰기 방지 단위 크기(NTWPU) 이 필드는 정전 또는 오류 조건에서 NVM에 원자적으로 기록되도록 보장되는 쓰기 작업의 네임스페이스별 크기를 나타냅니다. 이 필드는 0부터 시작하는 값으로 표시되는 논리적 블록으로 지정됩니다.
258:259	네임스페이스 찢긴 쓰기 방지 세분화 크기(NTWPG) 이 필드는 정전 또는 오류 조건에서 NVM에 원자적으로 기록되도록 보장되는 쓰기 작업의 네임스페이스별 크기 증분 단위(NTWPU 미만)를 나타냅니다. 즉, 크기는 NTWPG * n <= NTWPU이며 여기서 n은 양의 정수입니다. 쓰기 작업 LBA 오프셋도 이 필드에 맞추어 정렬되어야 합니다. 이 필드는 0부터 시작하는 값으로 표시되는 논리적 블록으로 지정됩니다.
260:263	네임스페이스 찢긴 쓰기 방지 경계 크기(NTWPB) 이 필드는 NTWPU 값에 대한 이 네임스페이스의 원자 경계 크기를 나타냅니다. 원자 경계를 벗어난 이 네임스페이스에 대한 쓰기는 정전 또는 오류 조건에서 NVM에 원자적으로 기록되도록 보장되지 않습니다. 0h 값은 정전 또는 오류 조건에 대한 원자 경계가 없음을 나타냅니다. 다른 모든 값은 NTWPU 필드와 동일한 인코딩을 사용하여 논리적 블록 단위로 크기를 지정합니다.

손상된 쓰기 방지를 위해 Amazon EC2의 워크로드 구성

찢긴 쓰기 방지 기능은 지원되는 볼륨을 사용한 지원되는 인스턴스 유형에서 기본적으로 활성화됩니다. 볼륨 또는 인스턴스의 찢긴 쓰기 방지를 활성화하기 위해 추가 설정을 활성화할 필요는 없습니다.

> **Note**
>
> 찢긴 쓰기 방지를 지원하지 않는 워크로드의 성능에는 영향을 미치지 않습니다. 이러한 워크로드는 변경할 필요가 없습니다.
> 찢긴 쓰기 방지를 지원하지만 이 기능을 사용하도록 구성되지 않은 워크로드는 이중 쓰기 버퍼를 계속 사용하며 성능상의 이점을 얻지 못합니다.

이중 쓰기 버퍼를 비활성화하고 찢긴 쓰기 방지 기능을 사용하도록 MySQL 또는 MariaDB 소프트웨어 스택을 구성하려면 다음 단계를 완료합니다.

1. BigAlloc 옵션을 사용하여 ext4 파일 시스템을 사용하도록 볼륨을 구성하고, 클러스터 크기를 4KiB, 8KiB 또는 16KiB로 설정합니다. 클러스터 크기가 4KiB, 8KiB 또는 16KiB인 BigAlloc을 사용하면 파일 시스템이 해당 경계에 맞춰 파일을 할당합니다.

   ```
   $  mkfs.ext4 -O bigalloc -C 4096|8192|16384 device_name
   ```

   > **Note**
   >
   > MySQL 및 MariaDB의 경우, `-C 16384`를 사용하여 데이터베이스 페이지 크기가 일치하도록 해야 합니다. 할당 세분화 수준을 페이지 크기의 배수가 아닌 값으로 설정하면, 할당되는 스토리지 디바이스의 찢긴 쓰기 방지 경계와 일치하지 않을 수 있습니다.

   예시:

   ```
   $  mkfs.ext4 -O bigalloc -C 16384 /dev/nvme1n1
   ```

2. `O_DIRECT` 플러싱 메서드를 사용하도록 InnoDB를 구성하고 InnoDB 이중 쓰기를 끕니다. 원하는 텍스트 편집기를 사용하여 `/etc/my.cnf`을 열고, 다음과 같이 `innodb_flush_method` 및 `innodb_doublewrite` 파라미터를 업데이트합니다.

   ```
   innodb_flush_method=O_DIRECT
   ```

```
innodb_doublewrite=0
```

Important

Logical Volume Manager(LVM) 또는 기타 스토리지 가상화 계층을 사용하는 경우, 볼륨의 시작 오프셋이 16KiB의 배수로 맞추어져 있는지 확인합니다. 이는 스토리지 가상화 계층에서 사용하는 메타데이터 헤더 및 슈퍼블록을 고려하여 기반 NVMe 스토리지를 기준으로 합니다. LVM 물리적 볼륨에 오프셋을 추가하면 파일 시스템 할당과 NVMe 디바이스의 오프셋 간에 정렬이 잘못되어 찢긴 쓰기 방지가 무효화될 수 있습니다. 자세한 내용은 Linux 매뉴얼 페이지에서 `--dataalignmentoffset` 섹션을 참조하세요.

Windows VSS 기반의 애플리케이션 일치 Amazon EBS 스냅샷

AWS Systems Manager Run Command를 사용하여 Amazon EC2 Windows 인스턴스에 연결된 모든 Amazon EBS 볼륨에 대해 애플리케이션 일치 스냅샷을 생성할 수 있습니다. 스냅샷 프로세스에서는 Windows VSS(Volume Shadow Copy Service)를 사용하여 VSS 인식 애플리케이션에 대해 EBS 볼륨 수준 백업을 받습니다. 스냅샷에는 이러한 애플리케이션과 디스크 간에 대기 중인 트랜잭션의 데이터가 포함됩니다. 연결된 볼륨을 모두 백업해야 하는 경우 인스턴스를 종료하거나 연결을 해제할 필요가 없습니다.

VSS 기반 EBS 스냅샷 사용에 따르는 추가 요금은 없습니다. 백업 프로세스에서 생성한 EBS 스냅샷에 대해서만 요금을 지불합니다. 자세한 내용은 제 Amazon EBS 스냅샷의 요금은 어떻게 청구되나요?를 참조하세요.

Note

애플리케이션에 대한 일관성이 있는 Windows VSS 기반 스냅샷은 Windows 인스턴스에서만 지원됩니다.

내용

- VSS란 무엇인가요?
- VSS 기반 Amazon EBS 스냅샷 솔루션의 작동 방식
- Windows VSS 기반 EBS 스냅샷을 생성하기 위한 사전 조건
- EC2 Windows 인스턴스를 위한 VSS 기반 EBS 스냅샷 생성

- Windows VSS 기반 EBS 스냅샷 문제 해결
- AWS VSS 솔루션을 사용하여 인스턴스의 데이터 복원
- AWS VSS 솔루션 버전 기록

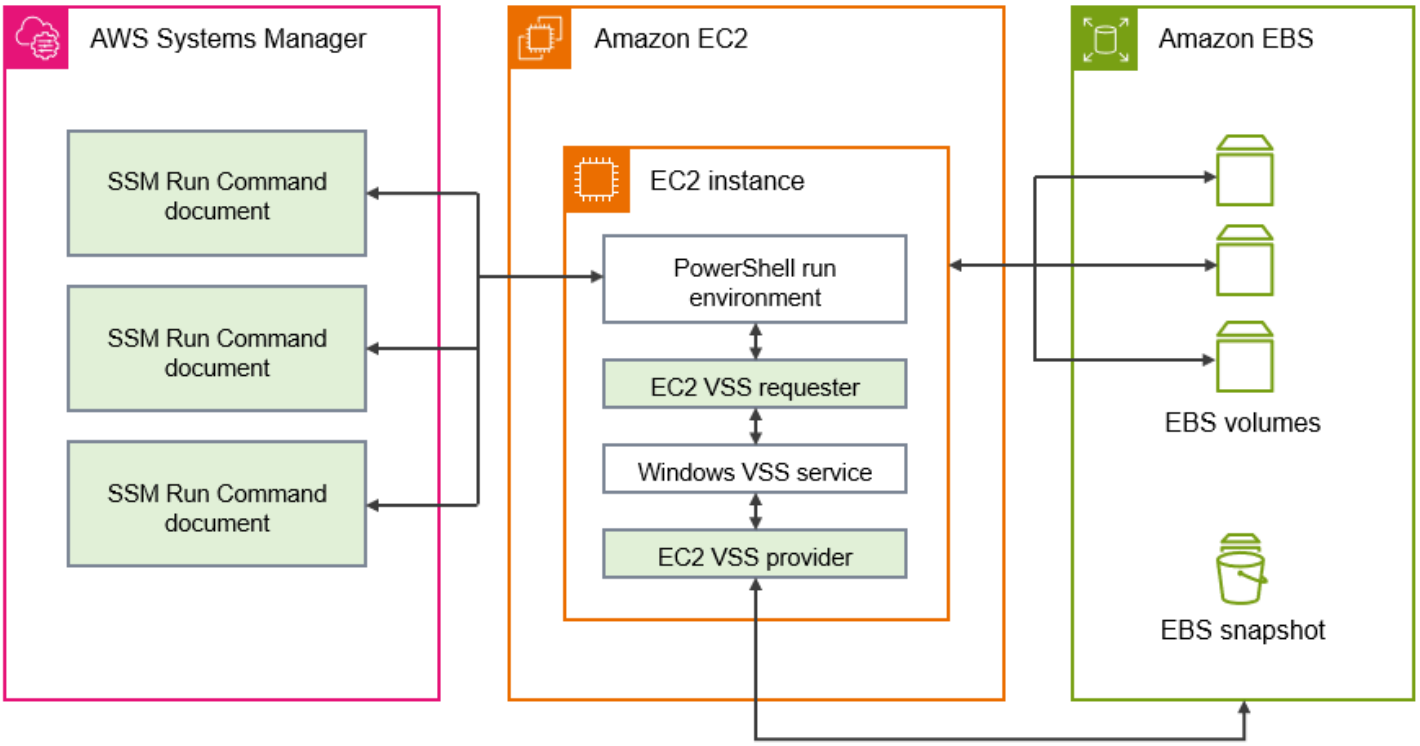
VSS란 무엇인가요?

VSS(Volume Shadow Copy Service)는 Microsoft Windows에 포함된 백업 및 복구 기술입니다. 사용 중인 컴퓨터 파일 또는 볼륨의 백업 복사본 또는 스냅샷을 생성할 수 있습니다. 자세한 내용은 Volume Shadow Copy Service를 참조하세요.

애플리케이션 일치 스냅샷을 생성하려면 다음 소프트웨어 구성 요소가 필요합니다.

- VSS 서비스 - Windows 운영 체제의 일부
- VSS 요청자 - 섀도우 복사본 생성을 요청하는 소프트웨어
- VSS 작성자 - 일관적인 데이터 세트의 백업을 보장하기 위해 대체로 애플리케이션의 일부로 제공 (예: SQL Server)
- VSS 공급자 - 기본 볼륨의 섀도우 복사본을 생성하는 구성 요소

Windows VSS 기반 Amazon EBS 스냅샷 솔루션은 백업 생성을 지원하는 여러 SSM(Systems Manager) Run Command 문서, 그리고 EC2 VSS 요청자 및 EC2 VSS 공급자를 포함하는 `AwsVssComponents`라고 하는 Systems Manager Distributor 패키지로 구성되어 있습니다. EBS 볼륨의 애플리케이션 일치 스냅샷을 가져하려면 `AwsVssComponents` 패키지를 EC2 Windows 인스턴스에 설치해야 합니다. 다음 다이어그램은 이러한 소프트웨어 구성 요소 간의 관계를 보여줍니다.



VSS 기반 Amazon EBS 스냅샷 솔루션의 작동 방식

다음 단계는 애플리케이션이 일치하는 VSS 기반 EBS 스냅샷 스크립트를 찍는 프로세스의 작동 방식입니다.

1. Windows VSS 기반 EBS 스냅샷을 생성하기 위한 사전 조건 단원을 완료합니다.
2. `AWSEC2-VssInstallAndSnapshot` SSM 문서에 대해 파라미터를 입력하고 Run Command를 사용하여 이 문서를 실행합니다. 자세한 내용은 AWSEC2-VssInstallAndSnapshot 명령 문서 실행(권장) 섹션을 참조하세요.
3. 해당 인스턴스에 있는 Windows VSS 서비스는 애플리케이션 실행을 위해 모든 지속적인 I/O 작업을 조정합니다.
4. 이 시스템은 모든 I/O 버퍼를 플러시하고 모든 I/O 작업을 일시적으로 중지합니다. 중지 시간은 최대 10초입니다.
5. 중지된 시간 동안 시스템은 인스턴스에 연결된 모든 볼륨의 스냅샷을 생성합니다.
6. 중지가 해제되면 I/O가 작업을 재개합니다.
7. 시스템은 새로 생성된 모든 스냅샷을 EBS 스냅샷 목록에 추가합니다. 시스템은 이 프로세스에 따라 성공적으로 생성된 모든 VSS 기반 EBS 스냅샷에 AppConsistent:true라는 태그를 붙입니다.

8. 스냅샷에서 복원해야 하는 경우 스냅샷에서 볼륨을 생성하는 표준 EBS 프로세스를 사용하거나 AWS VSS 솔루션을 사용하여 인스턴스의 데이터 복원에서 설명하는 대로 샘플 스크립트를 사용하여 인스턴스에 모든 볼륨을 복원할 수 있습니다.

Windows VSS 기반 EBS 스냅샷을 생성하기 위한 사전 조건

Systems Manager Run Command, AWS Backup 또는 Amazon Data Lifecycle Manager를 사용하여 VSS 기반 EBS 스냅샷을 생성할 수 있습니다. 다음 사전 조건은 모든 솔루션에 적용됩니다.

시스템 요구 사항

EC2 Windows 인스턴스가 지원되는 버전의 Windows 운영 체제, .NET Framework, PowerShell, AWS Tools for Windows PowerShell, AWS Systems Manager Agent를 포함하여 VSS 기반 스냅샷을 생성하기 위한 모든 시스템 요구 사항을 충족하는지 확인합니다.

IAM 권한

Amazon EC2 Windows 인스턴스에 연결된 IAM 역할에는 VSS에서 애플리케이션 일치 스냅샷을 생성할 수 있는 권한이 있어야 합니다. 필요한 권한을 부여하기 위해 `AWSEC2VssSnapshotPolicy` 관리형 정책을 인스턴스 프로파일에 연결할 수 있습니다.

VSS 구성 요소

Windows 운영 체제에서 애플리케이션에 일관되게 적용되는 스냅샷을 생성하려면 인스턴스에 `AwsVssComponents` 패키지를 설치해야 합니다. 패키지에는 VSS 요청자 역할을 하는 온인스턴스 EC2 VSS 에이전트와 EBS 볼륨용 EC2 VSS 공급자가 포함되어 있습니다.

시스템 요구 사항

Systems Manager 에이전트 설치

VSS는 PowerShell을 사용하는 Systems Manager Agent에 의해 오케스트레이션됩니다. EC2 인스턴스에 SSM Agent 버전 `3.0.502.0` 이상을 설치했는지 확인합니다. 이미 이전 버전의 SSM Agent를 사용하는 경우 Run Command를 사용하여 업데이트합니다. 자세한 내용은 AWS Systems Manager 사용 설명서의 Amazon EC2 인스턴스용 Systems Manager 설정 및 Windows Server용 Amazon EC2 인스턴스에서 SSM Agent 사용을 참조하세요.

Amazon EC2 Windows 인스턴스의 요구 사항

VSS 기반 EBS 스냅샷은 Windows Server 2016 이후 버전을 실행하는 인스턴스에 대해 지원됩니다.

.NET Framework 버전

`AwsVssComponents` 패키지에는 .NET Framework 버전 `4.6` 이상이 필요합니다. Windows Server 2016 이전의 Windows 운영 체제 버전은 기본적으로 이전 버전의 .NET Framework로 설정됩니다. 인스턴스에서 이전 버전의 .NET Framework를 사용하는 경우 Windows Update를 사용하여 버전 `4.6` 이상을 설치해야 합니다.

AWS Tools for Windows PowerShell 버전

인스턴스에서 AWS Tools for Windows PowerShell 버전 `3.3.48.0` 이상을 실행 중인지 확인합니다. 버전을 확인하려면 인스턴스의 PowerShell 터미널에서 다음 명령을 실행합니다.

```
C:\> Get-AWSPowerShellVersion
```

인스턴스에서 AWS Tools for Windows PowerShell를 업데이트해야 하는 AWS Tools for Windows PowerShell 사용 설명서의 AWS Tools for Windows PowerShell 설치를 참조하세요.

Windows PowerShell 버전

인스턴스에서 Windows PowerShell 주 버전 3, 4 또는 5를 실행 중인지 확인합니다. 버전을 확인하려면 인스턴스의 PowerShell 터미널에서 다음 명령을 실행합니다.

```
C:\> $PSVersionTable.PSVersion
```

PowerShell 언어 모드

인스턴스에 PowerShell 언어 모드가 `FullLanguage`로 설정되어 있는지 확인합니다. 자세한 내용은 Microsoft 설명서의 about_Language_Modes를 참조하세요.

IAM 관리형 정책을 사용하여 VSS 기반 스냅샷에 대한 권한 부여

AWSEC2VssSnapshotPolicy 관리형 정책을 사용하면 Systems Manager가 Windows 인스턴스에서 다음 작업을 수행할 수 있습니다.

- EBS 스냅샷 생성 및 태그 지정
- Amazon Machine Image(AMI) 생성 및 태그 지정
- 디바이스 ID와 같은 메타데이터를 VSS에서 생성하는 기본 스냅샷 태그에 연결합니다.

이 주제에서는 VSS 관리형 정책의 권한 세부 정보와 이를 EC2 인스턴스 프로파일 IAM 역할에 연결하는 방법을 다룹니다.

내용

- AWSEC2VssSnapshotPolicy 관리형 정책 세부 정보
- 인스턴스 프로파일 역할에 VSS 스냅샷 관리형 정책 연결

AWSEC2VssSnapshotPolicy 관리형 정책 세부 정보

AWS 관리형 정책은 AWS 고객을 위해 Amazon에서 제공하는 독립 실행형 정책입니다. AWS 관리형 정책은 일반 사용 사례에 대한 권한을 부여하도록 설계되었습니다. AWS 관리형 정책에 정의된 권한은 변경할 수 없습니다. 하지만 정책을 복사하여 사용 사례에 특정한 고객 관리형 정책의 기준으로 사용할 수 있습니다.

AWS 관리형 정책에 대한 자세한 정보는 IAM 사용 설명서에서 AWS 관리형 정책을 참조하십시오.

AWSEC2VssSnapshotPolicy 관리형 정책을 사용하려면 EC2 Windows 인스턴스에 연결된 IAM 역할에 정책을 연결하면 됩니다. 이 정책을 통해 EC2 VSS 솔루션에서 Amazon Machine Image(AMI) 및 EBS 스탭샷을 생성하고 이에 태그를 지정할 수 있습니다. 정책을 연결하려면 인스턴스 프로파일 역할에 VSS 스냅샷 관리형 정책 연결 섹션을 참조하세요.

AWSEC2VssSnapshotPolicy으로 부여된 권한

AWSEC2VssSnapshotPolicy 정책에는 Amazon EC2가 사용자를 대신하여 VSS 스냅샷을 생성하고 관리할 수 있도록 하는 다음과 같은 Amazon EC2 권한이 포함되어 있습니다. EC2 Windows 인스턴스에서 사용할 IAM 인스턴스 프로파일 역할에 이 관리형 정책을 연결할 수 있습니다.

- ec2:CreateTags – 리소스를 식별하고 분류하는 데 도움이 되도록 EBS 스냅샷 및 AMI에 태그를 추가합니다.
- ec2:DescribeInstanceAttribute – 대상 인스턴스에 연결된 EBS 볼륨 및 해당 블록 디바이스 매핑을 검색합니다.
- ec2:CreateSnapshots – EBS 볼륨의 스냅샷을 생성합니다.
- ec2:CreateImage – 실행 중인 EC2 인스턴스에서 AMI를 생성합니다.
- ec2:DescribeImages – EC2 AMI 및 스냅샷에 대한 정보를 검색합니다.
- ec2:DescribeSnapshots – 스냅샷의 생성 시간과 상태를 결정하여 애플리케이션 일관성을 확인합니다.

> **Note**
>
> 이 정책의 권한 세부 정보를 보려면 AWS 관리형 정책 참조의 AWSEC2VssSnapshotPolicy 섹션을 참조하세요.

특정 사용 사례에 대한 권한 간소화 - 고급

AWSEC2VssSnapshotPolicy 관리형 정책에는 VSS 기반 스냅샷을 생성할 수 있는 모든 방법에 대한 권한이 포함됩니다. 필요한 권한만 포함하는 사용자 지정 정책을 생성할 수 있습니다.

사용 사례: AMI 생성, 사용 사례: AWS Backup 서비스 사용

이 CreateAmi 옵션만 사용하거나 AWS Backup 서비스를 통해서만 VSS 기반 스냅샷을 생성하는 경우 다음과 같이 정책 설명을 간소화할 수 있습니다.

- 다음 명령문 ID(SID)로 식별되는 정책 명령문을 생략합니다.
  - CreateSnapshotsWithTag
  - CreateSnapshotsAccessInstance
  - CreateSnapshotsAccessVolume
- CreateTagsOnResourceCreation 문을 다음과 같이 조정합니다.
  - 리소스에서 arn:aws:ec2:*:*:snapshot/*을 제거합니다.
  - ec2:CreateAction 조건에서 CreateSnapshots를 제거합니다.
- CreateTagsAfterResourceCreation 문을 조정하여 리소스에서 arn:aws:ec2:*:*:snapshot/*을 제거합니다.
- DescribeImagesAndSnapshots 문을 조정하여 명령문 작업에서 ec2:DescribeSnapshots를 제거합니다.

사용 사례: 스냅샷만

CreateAmi 옵션을 사용하지 않는 경우 다음과 같이 정책 명령문을 간소화할 수 있습니다.

- 다음 명령문 ID(SID)로 식별되는 정책 명령문을 생략합니다.
  - CreateImageAccessInstance
  - CreateImageWithTag
- CreateTagsOnResourceCreation 문을 다음과 같이 조정합니다.

  - 리소스에서 `arn:aws:ec2:*:*:image/*`을 제거합니다.
  - `ec2:CreateAction` 조건에서 `CreateImage`를 제거합니다.
- `CreateTagsAfterResourceCreation` 문을 조정하여 리소스에서 `arn:aws:ec2:*:*:image/*`을 제거합니다.
- `DescribeImagesAndSnapshots` 문을 조정하여 명령문 작업에서 `ec2:DescribeImages`를 제거합니다.

> **Note**
>
> 사용자 지정된 정책이 예상대로 작동하려면 정기적으로 관리형 정책을 검토하고 업데이트를 통합하는 것이 좋습니다.

인스턴스 프로파일 역할에 VSS 스냅샷 관리형 정책 연결

EC2 Windows 인스턴스에 VSS 기반 스냅샷에 대한 권한을 부여하려면 다음과 같이 AWSEC2VssSnapshotPolicy 관리형 정책을 인스턴스 프로파일 역할에 연결합니다. 인스턴스가 모든 시스템 요구 사항을 충족하는지 확인해야 합니다.

> **Note**
>
> 관리형 정책을 사용하려면 인스턴스에 `AwsVssComponents` 패키지 버전 `2.3.1` 이상이 설치되어 있어야 합니다. 버전 기록은 AwsVssComponents 패키지 버전을(를) 참조하세요.

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 역할을 선택하여 액세스 권한이 있는 IAM 역할 목록을 표시합니다.
3. 인스턴스에 연결된 역할의 역할 이름 링크를 선택합니다. 그러면 역할 세부 정보 페이지가 열립니다.
4. 관리형 정책을 연결하려면 목록 패널의 오른쪽 상단에 있는 권한 추가를 선택합니다. 그리고 드롭다운 목록에서 정책 연결을 선택합니다.
5. 결과를 간소화하기 위해 검색 창(`AWSEC2VssSnapshotPolicy`)에 정책 이름을 입력합니다.
6. 연결할 정책 이름 옆의 확인란을 선택하고 권한 추가를 선택합니다.

Windows VSS 기반 EBS 스냅샷용 VSS 구성 요소 패키지 관리

VSS 기반 EBS 스냅샷을 생성하기 전에 Windows 인스턴스에 최신 버전의 VSS 구성 요소 패키지가 설치되어 있는지 확인합니다. 다음과 같은 여러 방법으로 기존 인스턴스에 `AwsVssComponents` 패키지를 설치할 수 있습니다.

- (권장) AWSEC2-VssInstallAndSnapshot 명령 문서 실행(권장). 그러면 실행할 때마다 필요한 경우 자동으로 설치 또는 업데이트됩니다.
- EC2 Windows 인스턴스에 수동으로 VSS 구성 요소 설치.
- EC2 Windows 인스턴스에서 VSS 구성 요소 패키지 업데이트.

`aws-vss-components-windows` 관리형 구성 요소를 사용하여 이미지에 대해 `AwsVssComponents` 패키지를 설치하는 EC2 Image Builder로 AMI를 생성할 수도 있습니다. 관리형 구성 요소는 AWS Systems Manager Distributor를 사용하여 패키지를 설치합니다. Image Builder에서 이미지를 생성한 후에는 연결된 AMI에서 시작하는 모든 인스턴스에 VSS 패키지가 설치됩니다. VSS 패키지가 설치된 AMI를 생성하는 방법에 대한 자세한 내용은 EC2 Image Builder 사용 설명서에서 Distributor package managed components for Windows를 참조하세요.

내용

- EC2 Windows 인스턴스에 수동으로 VSS 구성 요소 설치
- EC2 Windows 인스턴스에서 VSS 구성 요소 패키지 업데이트

EC2 Windows 인스턴스에 수동으로 VSS 구성 요소 설치

Systems Manager에서 애플리케이션에 일관되게 적용되는 스냅샷을 생성하기 전에 먼저 EC2 Windows 인스턴스에 VSS 구성 요소를 설치해야 합니다. 애플리케이션에 일관되게 적용되는 스냅샷을 생성할 때마다 패키지를 자동으로 설치 또는 업데이트하는 `AWSEC2-VssInstallAndSnapshot` 명령 문서를 실행하지 않는 경우 패키지를 수동으로 설치해야 합니다.

다음 방법 중 하나를 사용하여 EC2 인스턴스에서 애플리케이션에 일관되게 적용되는 스냅샷을 생성하려는 경우에도 수동으로 설치해야 합니다.

- AWS Backup을 사용하여 VSS 스냅샷 생성
- Amazon Data Lifecycle Manager를 사용하여 VSS 스냅샷 생성

수동 설치를 수행해야 하는 경우 최신 AWS VSS 구성 요소 패키지를 사용하여 EC2 Windows 인스턴스에서 애플리케이션에 일관되게 적용되는 스냅샷의 신뢰성과 성능을 개선하는 것이 좋습니다.

> **Note**
>
> 애플리케이션에 일관되게 적용되는 스냅샷을 생성할 때마다 `AwsVssComponents` 패키지를 자동으로 설치하거나 업데이트하려면 Systems Manager를 사용하여 `AWSEC2-VssInstallAndSnapshot` 문서를 실행하는 것이 좋습니다. 자세한 내용은 AWSEC2-VssInstallAndSnapshot 명령 문서 실행(권장) 섹션을 참조하세요.

Amazon EC2 Windows 인스턴스에 VSS 구성 요소를 설치하려면 원하는 환경에 해당하는 단계를 따르세요.

Console

SSM Distributor를 사용하여 VSS 구성 요소 설치

1. AWS Systems Manager 콘솔(https://console.aws.amazon.com/systems-manager/)을 엽니다.
2. 탐색 창에서 Run Command를 선택합니다.
3. Run 명령을 선택합니다.
4. 명령 문서에서 AWS-ConfigureAWSPackage 옆의 버튼을 선택합니다.
5. 명령 파라미터에서 다음을 수행합니다.
   a. 작업에서 설치로 설정되었는지 확인합니다.
   b. 이름(Name)에 `AwsVssComponents`을 입력합니다.
   c. 버전 필드에서 버전을 입력하거나 비워두어 Systems Manager가 최신 버전을 설치하도록 합니다.
6. 대상에서, 수동으로 태그를 지정하거나 인스턴스를 선택하여 이 작업을 실행할 인스턴스를 식별합니다.

   > **Note**
   >
   > 인스턴스를 수동으로 선택하려고 할 때 예상한 인스턴스가 목록에 없는 경우 AWS Systems Manager 사용 설명서의 내 인스턴스는 어디에 있나요?에서 문제 해결 팁을 확인하세요.

7. 다른 파라미터:
   - (선택 사항) 설명에 이 명령에 대한 정보를 입력합니다.

- 제한 시간(초)에서 전체 명령 실행이 실패할 때까지 시스템이 기다리는 시간을 초 단위로 지정합니다.

8. (선택 사항) 속도 제어:

   - 동시성에서 명령을 동시에 실행할 인스턴스의 백분율 또는 개수를 지정합니다.

     > **Note**
     >
     > Amazon EC2 태그를 선택하여 대상을 선택했지만 몇 개의 인스턴스가 선택된 태그를 사용할지 확실치 않다면 백분율을 지정하여 동시에 문서를 실행할 수 있는 인스턴스의 수를 제한하세요.

   - 오류 임계값에서, 명령이 인스턴스의 개수 또는 백분율에서 실패한 후 다른 인스턴스에서 해당 명령의 실행을 중지할 시간을 지정합니다. 예를 들어 세 오류를 지정하면 네 번째 오류를 받았을 때 Systems Manager가 명령 전송을 중지합니다. 여전히 명령을 처리 중인 인스턴스도 오류를 전송할 수 있습니다.

9. (선택 사항) 출력 옵션 섹션에서 명령 출력을 파일에 저장하려면 Enable writing to an S3 bucket(S3 버킷에 쓰기 활성화) 옆의 상자를 선택합니다. 버킷과 접두사 (폴더) 이름(선택 사항)을 지정합니다.

   > **Note**
   >
   > 데이터를 S3 버킷에 쓰는 기능을 부여하는 S3 권한은 이 작업을 수행하는 사용자의 권한이 아닌 인스턴스에 할당된 인스턴스 프로파일의 권한입니다. 자세한 내용은 AWS Systems Manager 사용 설명서의 EC2 인스턴스 권한 구성을 참조하세요.

10. (선택 사항) SNS 알림 옵션을 지정합니다.

    Run Command에 대한 Amazon SNS 알림 구성에 대한 자세한 내용은 AWS Systems Manager에 대한 Amazon SNS 알림 구성을 참조하세요.

11. 실행을 선택합니다.

AWS CLI

다음 절차를 통해 `AwsVssComponents`에서 Run Command를 사용하여 인스턴스에 AWS CLI 패키지를 다운로드하고 설치합니다. 패키지는 두 가지 구성 요소, 즉 VSS 요청자 및 VSS 공급자를 설

치합니다. 시스템은 이들 구성 요소들 인스턴스의 디렉터리에 복사한 후 공급자 DLL을 VSS 공급자로 등록합니다.

VSS 패키지를 설치하려면

다음 명령을 실행하여 Systems Manager용 필수 VSS 구성 요소를 다운로드하여 설치합니다.

```
aws ssm send-command \
    --document-name "AWS-ConfigureAWSPackage" \
    --instance-ids "i-1234567890abcdef0" \
    --parameters '{"action":["Install"],"name":["AwsVssComponents"]}'
```

PowerShell

다음 절차를 통해 Tools for Windows PowerShell에서 Run Command를 사용하여 인스턴스에 `AwsVssComponents` 패키지를 다운로드하고 설치합니다. 패키지는 두 가지 구성 요소, 즉 VSS 요청자 및 VSS 공급자를 설치합니다. 시스템은 이들 구성 요소들 인스턴스의 디렉터리에 복사한 후 공급자 DLL을 VSS 공급자로 등록합니다.

VSS 패키지를 설치하려면

다음 명령을 실행하여 Systems Manager용 필수 VSS 구성 요소를 다운로드하여 설치합니다.

```
Send-SSMCommand `
    -DocumentName "AWS-ConfigureAWSPackage" `
    -InstanceId "i-1234567890abcdef0" `
    -Parameter @{'action'='Install';'name'='AwsVssComponents'}
```

AWS VSS 구성 요소의 서명 확인

다음 절차를 사용하여 `AwsVssComponents` 패키지에 대한 서명을 확인합니다.

1. Windows 인스턴스에 연결합니다. 자세한 내용은 RDP를 사용하여 Windows 인스턴스에 연결 섹션을 참조하세요.
2. C:\Program Files\Amazon\AwsVssComponents로 이동합니다.
3. `ec2-vss-agent.exe`에 대한 컨텍스트 메뉴를 열고(마우스 오른쪽 버튼을 클릭) 속성을 선택합니다.
4. 디지털 서명 탭으로 이동하여 서명자의 이름이 Amazon Web Services Inc.인지 확인합니다.
5. 이전에 진행한 단계를 사용하여 `Ec2VssInstaller` 및 `Ec2VssProvider.dll`의 서명을 확인합니다.

EC2 Windows 인스턴스에서 VSS 구성 요소 패키지 업데이트

VSS 구성 요소를 최신 권장 버전으로 업데이트하는 것이 좋습니다. `AwsVssComponents` 패키지의 새 버전이 출시되면 구성 요소를 업데이트할 수 있는 여러 가지 방법이 있습니다.

업데이트 방법

- 새 버전의 AWS VSS 구성 요소가 릴리스되는 경우 EC2 Windows 인스턴스에 수동으로 VSS 구성 요소 설치에서 설명하는 단계를 반복할 수 있습니다.
- `AwsVssComponents` 패키지가 제공되면 새로운 VSS 구성 요소나 업데이트된 VSS 구성 요소를 자동으로 다운로드하고 설치하도록 Systems Manager State Manager 연결을 구성할 수 있습니다.
- Systems Manager를 사용하여 `AWSEC2-VssInstallAndSnapshot` 문서를 실행하는 경우 애플리케이션에 일관되게 적용되는 스냅샷을 생성할 때마다 `AwsVssComponents` 패키지를 자동으로 설치하거나 업데이트할 수 있습니다.

> **Note**
>
> 애플리케이션에 일관되게 적용되는 스냅샷이 생성되기 전에 Systems Manager를 사용하여 `AwsVssComponents` 패키지를 자동으로 설치하거나 업데이트하는 `AWSEC2-VssInstallAndSnapshot` 명령 문서를 실행하는 것이 좋습니다. 자세한 내용은 AWSEC2-VssInstallAndSnapshot 명령 문서 실행(권장) 섹션을 참조하세요.

Systems Manager State Manager 연결을 생성하려면 원하는 환경에 맞는 단계를 따르세요.

Console

Systems Manager State Manager 연결을 생성할 때 다음과 같이 `AwsVssComponents` 패키지를 업데이트하는 두 가지 옵션이 있습니다.

설치 제거 후 재설치

이 방법은 추가 사전 조건 없이 패키지를 다운로드하고 설치합니다.

현재 위치 업데이트

이 방법을 선택하면 패키지에 대한 현재 위치 업데이트가 수행되며 다음과 같은 사전 요구 사항이 있습니다.

- 인스턴스에 설치된 SSM 에이전트 버전이 버전 `3.3.808.0` 이상이어야 합니다. 자세한 내용은 AWS Systems Manager 사용 설명서의 Windows Server용 EC2 인스턴스에서 SSM Agent 사용을 참조하세요.
- 지정된 경우 `AwsVssComponents` 패키지 버전이 버전 `2.5.0` 이상이어야 합니다. 이전 버전은 현재 위치 업데이트를 지원하지 않습니다.

> **Note**
>
> 인스턴스가 이러한 사전 조건을 충족하지 않으면 현재 위치 업데이트가 실패합니다. 대신 설치 제거 후 재설치 옵션을 사용하세요.

State Manager 연결을 생성하려면

1. AWS Systems Manager 콘솔(https://console.aws.amazon.com/systems-manager/)을 엽니다.
2. 탐색 창에서 상태 관리자를 선택합니다.

   또는 Systems Manager 홈페이지가 먼저 열리면 탐색 창을 열고 State Manager를 선택합니다.
3. 연결 생성을 선택합니다.
4. [이름(Name)] 필드에 설명이 포함된 이름을 입력합니다.
5. 문서 목록에서 AWS-ConfigureAWSPackage를 선택합니다.
6. [파라미터(Parameters)] 섹션의 [작업(Action)] 목록에서 [설치(Install)]를 선택합니다.
7. 설치 유형에서 설치 제거 후 재설치 또는 현재 위치 업데이트를 선택합니다.
8. 이름 필드에 `AwsVssComponents`을 입력합니다. [버전(Version)] 및 [추가 인수(Additional Arguments)] 필드를 비워 둘 수 있습니다.
9. [대상(Targets)] 섹션에서 옵션을 선택합니다.

   > **Note**
   >
   > 태그를 사용하여 인스턴스 대상을 지정하고 Linux 인스턴스에 매핑되는 태그를 지정할 경우 Windows 인스턴스에서는 연결이 성공하지만 Linux 인스턴스에서는 실패합니다. 전체 연결 상태는 실패로 표시됩니다.
10. 일정 지정 섹션에서 옵션을 선택합니다.

11. 고급 옵션 섹션의 규정 준수 심각도에서 연결에 대한 심각도 수준을 선택합니다. 자세한 내용은 연결 규정 준수에 대해 알아보기를 참조하세요. 변경 일정의 경우 사전 구성된 변경 일정을 선택합니다. 자세한 내용은 AWS Systems Manager 변경 일정을 참조하세요.
12. 속도 제어의 경우 다음을 수행합니다.
    - Concurrency(동시성)에서 명령을 동시에 실행할 관리형 노드의 백분율 또는 개수를 지정합니다.
    - Error threshold(오류 임계값)에서, 명령이 노드의 개수 또는 백분율에서 실패한 후 다른 관리형 노드에서 해당 명령의 실행을 중지할 시간을 지정합니다.
13. (선택 사항) 출력 옵션에서 명령 출력을 파일에 저장하려면 S3 버킷에 쓰기 활성화 옆의 상자를 선택합니다. 상자에 버킷 및 접두사(폴더) 이름을 입력합니다.
14. [연결 생성(Create association)], [닫기(Close)]를 차례로 선택합니다. 시스템은 해당 인스턴스에 연결을 생성하고 그 상태를 즉시 적용하려고 합니다.

> **Note**
>
> Windows Server용 EC2 인스턴스의 상태가 실패로 표시될 경우 SSM Agent가 인스턴스에서 실행 중이고 Systems Manager에 대한 AWS Identity and Access Management(IAM) 역할로 인스턴스가 구성되었는지 확인합니다. 자세한 내용은 AWS Systems Manager 설정을 참조하세요.

AWS CLI

연결된 애플리케이션을 오프라인으로 전환하지 않고 일정에 따라 Distributor 패키지를 업데이트하려면 create-association 명령을 사용합니다. 패키지의 새 파일이나 업데이트된 파일만 바뀝니다.

State Manager 연결을 생성하려면

다음 명령을 실행하여 연결을 생성합니다. `--name` 값 즉, 문서 이름은 항상 `AWS-ConfigureAWSPackage`입니다. 다음 명령은 키 `InstanceIds`를 사용하여 대상 인스턴스를 지정합니다.

```
aws ssm create-association \
    --name "AWS-ConfigureAWSPackage" \
    --parameters action=Install,installationType="Uninstall and
 reinstall",name=AwsVssComponents \
    --targets Key=InstanceIds,Values=i-1234567890abcdef0,i-000011112222abcde
```

PowerShell

State Manager 연결을 생성하려면

New-SSMAssociation cmdlet을 사용합니다.

```
New-SSMAssociation `
    -Name "AWS-ConfigureAWSPackage" `
    -Parameter  @{
        "action" = "Install"
        "installationType" = "Uninstall and reinstall"
        "name" = "AwsVssComponents"
    } `
    -Target @{
        "Key" = "InstanceIds"
        "Values" = @("i-1234567890abcdef0", "i-000011112222abcde")
    }
```

EC2 Windows 인스턴스를 위한 VSS 기반 EBS 스냅샷 생성

Windows VSS 기반 EBS 스냅샷을 생성하기 위한 사전 조건을 모두 충족한 후에는 다음 방법 중 하나를 사용하여 EC2 인스턴스에서 VSS 기반 스냅샷을 생성할 수 있습니다.

AWS Systems Manager 명령 문서

Systems Manager 명령 문서 사용하여 VSS 기반 스냅샷을 생성합니다.

백업을 자동화하려면 `AWSEC2-VssInstallAndSnapshot` 명령 문서를 사용하는 AWS Systems Manager 유지 관리 기간 태스크를 생성하여 할 수 있습니다. 자세한 내용은 AWS Systems Manager 사용 설명서에서 유지 관리 기간 작업(콘솔)을 참조하세요.

AWS Backup

콘솔 또는 CLI에서 VSS를 활성화하여 AWS Backup 사용 시 VSS 백업을 생성할 수 있습니다. 이 기능에 대한 자세한 내용은 AWS Backup 개발자 안내서의 Windows VSS 백업 생성을 참조하세요.

> **Note**
>
> AWS Backup에서는 인스턴스에 `AwsVssComponents` 패키지를 자동으로 설치하지 않습니다. 인스턴스에서 수동 설치를 수행해야 합니다. 자세한 내용은 EC2 Windows 인스턴스에 수동으로 VSS 구성 요소 설치 단원을 참조하십시오.

Amazon Data Lifecycle Manager

스냅샷 수명 주기 정책에서 사전 및 사후 스크립트를 활성화하여 Amazon Data Lifecycle Manager로 VSS 스냅샷을 생성할 수 있습니다. 자세한 내용은 Amazon EBS 사용 설명서의 Automating application-consistent snapshots를 참조하세요.

> **Note**
>
> Amazon Data Lifecycle Manager에서는 인스턴스에 `AwsVssComponents` 패키지를 자동으로 설치하지 않습니다. 인스턴스에서 수동 설치를 수행해야 합니다. 자세한 내용은 EC2 Windows 인스턴스에 수동으로 VSS 구성 요소 설치 단원을 참조하십시오.

Systems Manager 명령 문서를 사용하여 VSS 기반 스냅샷 생성

AWS Systems Manager 명령 문서를 사용하여 VSS 기반 스냅샷을 생성할 수 있습니다. 다음 콘텐츠에서는 사용 가능한 명령 문서 및 해당 문서에서 스냅샷을 생성하는 데 사용하는 런타임 파라미터를 소개합니다.

Systems Manager 명령 문서를 사용하기 전에 모든 Windows VSS 기반 EBS 스냅샷을 생성하기 위한 사전 조건을 충족하는지 확인합니다.

주제

- Systems Manager VSS 스냅샷 문서의 파라미터
- Systems Manager VSS 스냅샷 명령 문서 실행

Systems Manager VSS 스냅샷 문서의 파라미터

VSS 스냅샷을 생성하는 Systems Manager 문서는 별도로 명시된 경우를 제외하고 다음 파라미터를 모두 사용합니다.

AmiName(문자열, 선택 사항)

CreateAmi 옵션이 `True`로 설정된 경우 백업에서 생성되는 AMI의 이름을 지정합니다.

description(문자열, 선택 사항)

이 프로세스에서 생성되는 스냅샷 또는 이미지에 대한 설명을 지정합니다.

CollectDiagnosticLogs(문자열, 선택 사항)

스냅샷 및 AMI 생성 단계에서 추가 정보를 수집하려면 이 파라미터를 “True“로 설정합니다. 이 파라미터의 기본값은 “False“입니다. 통합 진단 로그는 인스턴스의 다음 위치에 .zip 형식 아카이브로 저장됩니다.

C:\ProgramData\Amazon\AwsVss\Logs*timestamp*.zip

CopyOnly(문자열, 선택 사항)

AWS VSS 외에 기본 SQL Server 백업을 사용하는 경우 복사 전용 백업을 수행하면 AWS VSS가 기본 차등 백업 체인을 끊지 않습니다. 복사 전용 백업 작업을 수행하려면 이 파라미터를 True로 설정합니다.

이 파라미터의 기본값은 False이며, 이 값을 지정하면 AWS VSS에서 전체 백업 작업을 수행합니다.

CreateAmi(문자열, 선택 사항)

인스턴스를 백업하기 위해 VSS 기반 Amazon Machine Image(AMI)를 생성하려면 이 파라미터를 True로 설정합니다. 이 파라미터의 기본값은 False이며, 이 값을 지정하면 대신 EBS 스냅샷으로 인스턴스가 백업됩니다.

인스턴스에서 AMI 생성하기에 대한 자세한 내용은 Amazon EBS 지원 AMI 생성 섹션을 참조하세요.

executionTimeout(문자열, 선택 사항)

인스턴스에서 스냅샷 생성 프로세스를 실행하거나 인스턴스에서 AMI를 생성하는 최대 시간(초)을 지정합니다. 이 제한 시간을 늘리면 VSS가 동결을 시작하고 생성된 리소스의 태그 지정을 완료할 때까지 명령이 더 오래 기다릴 수 있습니다. 이 제한 시간은 스냅샷 또는 AMI 생성 단계에만 적용됩니다. AwsVssComponents 패키지를 설치하거나 업데이트하는 초기 단계는 제한 시간에 포함되지 않습니다.

ExcludeBootVolume(문자열, 선택 사항)

이 설정은 스냅샷을 생성하는 경우 백업 프로세스에서 부팅 볼륨을 제외합니다. 스냅샷에서 부팅 볼륨을 제외하려면 ExcludeBootVolume을 True로 설정하고 CreateAmi를 False로 설정합니다.

백업용 AMI를 생성하는 경우 이 파라미터를 False로 설정해야 합니다. 이 파라미터의 기본값은 False입니다.

NoWriters(문자열, 선택 사항)

스냅샷 프로세스에서 애플리케이션 VSS 작성자를 제외하려면 이 파라미터를 `True`로 설정합니다. 애플리케이션 VSS 작성자를 제외하면 타사 VSS 백업 구성 요소와의 갈등을 해결하는 데 도움이 될 수 있습니다. 이 파라미터의 기본값은 `False`입니다.

`SaveVssMetadata`가 `True`인 경우 파라미터를 `False`로 설정해야 합니다.

SaveVssMetadata(문자열, 선택 사항)

모든 스냅샷 중에 VSS 메타데이터 파일을 저장하려면 이 파라미터를 `True`로 설정하세요. 기본값은 `False`입니다. VSS 메타데이터 파일은 백업 작업에 포함된 구성 요소 또는 작성자와 각 구성 요소의 관련 파일에 대한 인사이트를 제공하는 데 도움이 됩니다.

메타데이터 파일 이름에는 연결된 스냅샷 세트 ID가 있으며 이는 인스턴스의 다음 위치에서 찾을 수 있습니다.

```
C:\ProgramData\Amazon\AwsVss\VssMetadata\
```

Warning

- VSS 메타데이터 파일을 저장하려면 `AwsVssComponents` 패키지 버전 2.4.0 이상이 필요합니다. 인스턴스에 이전 버전이 설치되어 있는 경우 `SaveVssMetadata`를 `True`로 설정하면 스냅샷 생성이 실패합니다.
- `NoWriters` 파라미터와 `SaveVssMetadata` 파라미터는 함께 사용할 수 없습니다. 둘 다 `True`로 설정하면 스냅샷 생성이 실패합니다.

tags(문자열, 선택 사항)

리소스를 찾고 관리하는 데 도움이 되도록 스냅샷과 이미지에 태그를 지정하는 것이 좋습니다(예: 스냅샷 목록에서 볼륨을 복원하는 경우). 시스템은 출력 스냅샷 또는 이미지에 적용할 이름을 지정할 수 있는 빈 값과 함께 `Name` 키를 추가합니다.

태그를 추가로 지정하려면 세미콜론을 사용해 태그를 구분합니다. 예: `Key=Environment,Value=Test;Key=User,Value=TestUser1`.

Note

태그 키 및 값에는 영숫자와 특수 문자 `() ./\-"'@_+:={}`만 포함되어야 합니다.

기본적으로 시스템은 VSS 기반 스냅샷 및 이미지에 대해 다음과 같은 예약된 태그를 추가합니다.

- 디바이스 - VSS 기반 스냅샷의 경우 스냅샷이 캡처하는 EBS 볼륨의 디바이스 이름입니다.
- AppConsistent - 이 태그는 VSS 기반 스냅샷 또는 AMI를 생성했음을 나타냅니다.
- AwsVssConfig - VSS가 활성화된 상태로 생성된 스냅샷 및 AMI를 식별합니다. 태그에는 `AwsVssComponents` 버전 및 스냅샷 세트 ID와 같은 메타 정보가 포함됩니다.

> **Warning**
>
> 파라미터 목록에 이러한 예약된 태그를 지정하면 오류가 발생합니다.

VssVersion(문자열, 선택 사항)

`AWSEC2-VssInstallAndSnapshot` 문서의 경우에만 `VssVersion` 파라미터를 지정하여 인스턴스에 특정 버전의 `AwsVssComponents` 패키지를 설치할 수 있습니다. 권장 기본 버전을 설치하려면 이 파라미터를 비워 둡니다.

지정된 버전의 `AwsVssComponents` 패키지가 이미 설치된 경우 스크립트는 설치 단계를 건너뛰고 백업 단계로 이동합니다. `AwsVssComponents` 패키지 버전 및 운영 지원 목록은 AWS VSS 솔루션 버전 기록 섹션을 참조하세요.

Systems Manager VSS 스냅샷 명령 문서 실행

다음과 같이 AWS Systems Manager 명령 문서를 사용하여 VSS 기반 EBS 스냅샷을 생성할 수 있습니다.

AWSEC2-VssInstallAndSnapshot 명령 문서 실행(권장)

AWS Systems Manager를 사용하여 `AWSEC2-VssInstallAndSnapshot` 문서를 실행하면 스크립트는 다음 단계를 실행합니다.

1. 스크립트는 먼저 `AwsVssComponents` 패키지가 이미 설치되어 있는지 여부에 따라 인스턴스에 이 패키지를 설치하거나 업데이트합니다.
2. 스크립트는 첫 번째 단계가 완료된 후 애플리케이션에 일관되게 적용되는 스냅샷을 생성합니다.

`AWSEC2-VssInstallAndSnapshot` 문서를 실행하려면 원하는 환경에 맞는 단계를 따르세요.

Console

콘솔에서 VSS 기반 EBS 스냅샷 생성

1. AWS Systems Manager 콘솔(https://console.aws.amazon.com/systems-manager/)을 엽니다.
2. 탐색 창에서 Run Command를 선택합니다. 해당하는 경우 계정에서 현재 실행 중인 명령 목록이 표시됩니다.
3. Run command(Run 명령)를 선택합니다. 액세스할 수 있는 명령 문서 목록이 열립니다.
4. 명령 문서 목록에서 `AWSEC2-VssInstallAndSnapshot`을 선택합니다. 결과를 간소화하기 위해 문서 이름 전체 또는 일부를 입력할 수 있습니다. 소유자, 플랫폼 유형 또는 태그로 필터링할 수도 있습니다.

   명령 문서를 선택하면 목록 아래에 세부 정보가 채워집니다.
5. 문서 버전 목록에서 `Default version at runtime`을 선택합니다.
6. 명령 파라미터를 구성하여 `AWSEC2-VssInstallAndSnapshot`이 `AwsVssComponents` 패키지를 설치하고 VSS 스냅샷 또는 AMI를 사용하여 백업하는 방법을 정의합니다. 파라미터 세부 정보는 Systems Manager VSS 스냅샷 문서의 파라미터 섹션을 참조하세요.
7. 대상 선택에서 태그를 지정하거나 인스턴스를 수동으로 선택하여 이 작업을 실행할 인스턴스를 식별합니다.

   > **Note**
   >
   > 인스턴스를 수동으로 선택할 때 예상한 인스턴스가 목록에 없는 경우 내 인스턴스는 어디에 있나요?에서 문제 해결 팁을 확인하세요.
8. 속도 제어와 같은 Systems Manager Run Command 동작을 정의하는 추가 파라미터에 대해 콘솔에서 명령 실행에 설명된 대로 값을 입력합니다.
9. 실행을 선택합니다.

   성공적으로 실행되면 명령은 EBS 스냅샷 목록에 새로운 스냅샷을 입력합니다. 지정한 태그를 검색하거나 `AppConsistent`를 검색하여 EBS 스냅샷 목록에서 이 스냅샷을 찾을 수 있습니다. 명령 실행에 실패하면 Systems Manager 명령 출력을 보고 명령이 실패한 이유에 대한 세부 정보를 확인합니다. 명령이 성공적으로 완료되었지만 특정 볼륨의 백업이 실패한 경우 EBS 볼륨 목록에서 문제를 해결할 수 있습니다.

AWS CLI

AWS CLI에서 다음 명령을 실행하여 VSS 기반 EBS 스냅샷을 생성하고 스냅샷 생성 상태를 가져올 수 있습니다.

VSS 기반 EBS 스냅샷 생성

다음 명령을 실행하여 VSS 기반 EBS 스냅샷을 생성합니다. 스냅샷을 생성하려면 `--instance-ids` 파라미터로 인스턴스를 식별해야 합니다. 추가할 수 있는 다른 파라미터에 대한 자세한 내용은 Systems Manager VSS 스냅샷 문서의 파라미터 섹션을 참조하세요.

```
aws ssm send-command \
 --document-name "AWSEC2-VssInstallAndSnapshot" \
 --instance-ids "i-01234567890abcdef" \
 --parameters '{"ExcludeBootVolume":["False"],"description":["Description"],"tags":
["Key=key_name,Value=tag_value"],"VssVersion":[""]}'
```

성공적으로 실행되면 명령 문서에서 EBS 스냅샷 목록을 새로운 스냅샷으로 채웁니다. 지정한 태그를 검색하거나 AppConsistent를 검색하여 EBS 스냅샷 목록에서 이 스냅샷을 찾을 수 있습니다. 명령 실행에 실패하면 명령 출력을 보고 명령이 실패한 이유에 대한 세부 정보를 확인합니다.

명령 상태 가져오기

스냅샷의 현재 상태를 확인하려면 send-command에서 반환된 명령 ID를 사용하여 다음 명령을 실행합니다.

```
aws ssm get-command-invocation
 --instance-ids "i-01234567890abcdef" \
 --command-id "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" \
 --plugin-name "CreateVssSnapshot"
```

PowerShell

AWS Tools for Windows PowerShell에서 다음 명령을 실행하여 VSS 기반 EBS 스냅샷을 생성하고 출력 생성에 대한 현재 런타임 상태를 확인합니다. 이전 목록에 설명된 파라미터를 지정하여 스냅샷 프로세스의 동작을 수정합니다.

Windows PowerShell용 도구로 VSS 기반 EBS 스냅샷 생성

다음 명령을 실행하여 VSS 기반 EBS 스냅샷 또는 AMI를 생성합니다.

```
Send-SSMCommand -DocumentName "AWSEC2-VssInstallAndSnapshot" -InstanceId
 "i-01234567890abcdef" -Parameter
 @{'ExcludeBootVolume'='False';'description'='a_description'
 ;'tags'='Key=key_name,Value=tag_value';'VssVersion'=''}
```

명령 상태 가져오기

스냅샷의 현재 상태를 확인하려면 Send-SSMCommand에서 반환된 명령 ID를 사용하여 다음 명령을 실행합니다.

```
Get-SSMCommandInvocationDetail -InstanceId "i-01234567890abcdef" -CommandId
 "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" -PluginName "CreateVssSnapshot"
```

성공적으로 실행되면 명령은 EBS 스냅샷 목록에 새로운 스냅샷을 입력합니다. 지정한 태그를 검색하거나 AppConsistent를 검색하여 EBS 스냅샷 목록에서 이 스냅샷을 찾을 수 있습니다. 명령 실행에 실패하면 명령 출력을 보고 명령이 실패한 이유에 대한 세부 정보를 확인합니다.

AWSEC2-CreateVssSnapshot 명령 문서 실행

AWSEC2-CreateVssSnapshot 문서를 실행하려면 원하는 환경에 맞는 단계를 따르세요.

Console

콘솔에서 VSS 기반 EBS 스냅샷 생성

1. AWS Systems Manager 콘솔(https://console.aws.amazon.com/systems-manager/)을 엽니다.
2. 탐색 창에서 Run Command를 선택합니다. 해당하는 경우 계정에서 현재 실행 중인 명령 목록이 표시됩니다.
3. Run command(Run 명령)를 선택합니다. 액세스할 수 있는 명령 문서 목록이 열립니다.
4. 명령 문서 목록에서 AWSEC2-CreateVssSnapshot을 선택합니다. 결과를 간소화하기 위해 문서 이름 전체 또는 일부를 입력할 수 있습니다. 소유자, 플랫폼 유형 또는 태그로 필터링할 수도 있습니다.

   명령 문서를 선택하면 목록 아래에 세부 정보가 채워집니다.
5. 문서 버전 목록에서 Default version at runtime을 선택합니다.
6. AWSEC2-CreateVssSnapshot을 통해 VSS 스냅샷 또는 AMI에서 백업하는 방법을 정의하도록 명령 파라미터를 구성합니다. 파라미터 세부 정보는 Systems Manager VSS 스냅샷 문서의 파라미터 섹션을 참조하세요.

7. 대상 선택에서 태그를 지정하거나 인스턴스를 수동으로 선택하여 이 작업을 실행할 인스턴스를 식별합니다.

   > **Note**
   >
   > 인스턴스를 수동으로 선택할 때 예상한 인스턴스가 목록에 없는 경우 내 인스턴스는 어디에 있나요?에서 문제 해결 팁을 확인하세요.

8. 속도 제어와 같은 Systems Manager Run Command 동작을 정의하는 추가 파라미터에 대해 콘솔에서 명령 실행에 설명된 대로 값을 입력합니다.
9. 실행을 선택합니다.

   성공적으로 실행되면 명령은 EBS 스냅샷 목록에 새로운 스냅샷을 입력합니다. 지정한 태그를 검색하거나 AppConsistent를 검색하여 EBS 스냅샷 목록에서 이 스냅샷을 찾을 수 있습니다. 명령 실행에 실패하면 Systems Manager 명령 출력을 보고 명령이 실패한 이유에 대한 세부 정보를 확인합니다. 명령이 성공적으로 완료되었지만 특정 볼륨의 백업이 실패한 경우 EBS 볼륨 목록에서 문제를 해결할 수 있습니다.

AWS CLI

AWS CLI에서 다음 명령을 실행하여 VSS 기반 EBS 스냅샷을 생성합니다.

VSS 기반 EBS 스냅샷 생성

다음 명령을 실행하여 VSS 기반 EBS 스냅샷을 생성합니다. 스냅샷을 생성하려면 `--instance-ids` 파라미터로 인스턴스를 식별해야 합니다. 추가할 수 있는 다른 파라미터에 대한 자세한 내용은 Systems Manager VSS 스냅샷 문서의 파라미터 섹션을 참조하세요.

```
aws ssm send-command \
 --document-name "AWSEC2-CreateVssSnapshot" \
 --instance-ids "i-01234567890abcdef" \
 --parameters '{"ExcludeBootVolume":["False"],"description":["Description"],"tags":
["Key=key_name,Value=tag_value"]}'
```

성공적으로 실행되면 명령 문서에서 EBS 스냅샷 목록을 새로운 스냅샷으로 채웁니다. 지정한 태그를 검색하거나 AppConsistent를 검색하여 EBS 스냅샷 목록에서 이 스냅샷을 찾을 수 있습니다. 명령 실행에 실패하면 명령 출력을 보고 명령이 실패한 이유에 대한 세부 정보를 확인합니다.

PowerShell

AWS Tools for Windows PowerShell에서 다음 명령을 실행하여 VSS 기반 EBS 스냅샷을 생성합니다.

Windows PowerShell용 도구로 VSS 기반 EBS 스냅샷 생성

다음 명령을 실행하여 VSS 기반 EBS 스냅샷을 생성합니다. 스냅샷을 생성하려면 `InstanceId` 파라미터로 인스턴스를 식별해야 합니다. 스냅샷을 생성할 인스턴스를 둘 이상 지정할 수 있습니다. 추가할 수 있는 다른 파라미터에 대한 자세한 내용은 Systems Manager VSS 스냅샷 문서의 파라미터 섹션을 참조하세요.

```
Send-SSMCommand -DocumentName AWSEC2-CreateVssSnapshot -InstanceId
 "i-01234567890abcdef" -Parameter
 @{'ExcludeBootVolume'='False';'description'='a_description'
 ;'tags'='Key=key_name,Value=tag_value'}
```

성공적으로 실행되면 명령은 EBS 스냅샷 목록에 새로운 스냅샷을 입력합니다. 지정한 태그를 검색하거나 `AppConsistent`를 검색하여 EBS 스냅샷 목록에서 이 스냅샷을 찾을 수 있습니다. 명령 실행에 실패하면 명령 출력을 보고 명령이 실패한 이유에 대한 세부 정보를 확인합니다. 명령이 성공적으로 완료되었지만 특정 볼륨의 백업이 실패한 경우 EBS 스냅샷 목록에서 문제를 해결할 수 있습니다.

공유 EBS 스토리지가 있는 Windows 장애 조치 클러스터의 명령 문서 실행

이전 섹션에서 설명한 모든 명령줄 절차를 사용하여 VSS 기반 스냅샷을 생성할 수 있습니다. 명령 문서(`AWSEC2-VssInstallAndSnapshot` 또는 `AWSEC2-CreateVssSnapshot`)는 클러스터의 프라이머리 노드에서 실행되어야 합니다. 보조 노드에서는 공유 디스크에 대한 액세스가 없으므로 문서가 실패합니다. 프라이머리 노드와 보조 노드가 동적으로 변경되는 경우 명령이 프라이머리 노드에서는 성공하고 보조 노드에서는 실패할 것으로 예상하면서 여러 노드에서 AWS Systems Manager Run Command 문서를 실행할 수 있습니다.

> **Note**
>
> 백업을 자동화하려면 `AWSEC2-VssInstallAndSnapshot` 문서를 사용하는 AWS Systems Manager 유지 관리 기간 작업을 생성하여 할 수 있습니다. 자세한 내용은 AWS Systems Manager 사용 설명서에서 유지 관리 기간 작업(콘솔)을 참조하세요.

Windows VSS 기반 EBS 스냅샷 문제 해결

다른 문제 해결 단계를 시도하기 전에 다음 세부 정보를 확인하는 것이 좋습니다.

- 모든 Windows VSS 기반 EBS 스냅샷을 생성하기 위한 사전 조건을 충족하는지 확인합니다.
- 운영 체제에 맞는 `AwsVssComponents` 패키지의 최신 Windows OS 버전 지원을 사용하고 있는지 확인합니다. 관찰된 문제가 최신 버전에서 해결되었을 수 있습니다.

주제

- 로그 파일 확인
- 추가 진단 로그 수집
- 프록시가 구성된 인스턴스에서 VSS 사용
- 오류: 파이프 연결 재개 시간 초과, 재개 시 오류, VSS 동결 대기 중 시간 초과 또는 기타 시간 초과 오류
- 오류: 메서드를 호출할 수 없습니다. 메서드 호출은 이 언어 모드의 핵심 유형에서만 지원됩니다.

로그 파일 확인

VSS 기반 EBS 스냅샷을 생성할 때 문제가 발생하거나 오류 메시지가 나타나면 Systems Manager 콘솔에서 명령 출력을 볼 수 있습니다.

VSS 스냅샷을 생성하는 Systems Manager 문서의 경우 런타임에 `CollectDiagnosticLogs` 파라미터를 “`True`“로 설정할 수 있습니다. `CollectDiagnosticLogs`파라미터를 "`True`"로 설정하면 VSS는 디버깅을 돕기 위해 추가 로그를 수집합니다. 자세한 내용은 추가 진단 로그 수집 섹션을 참조하세요.

진단 로그를 수집하는 경우 Systems Manager 문서는 해당 로그를 인스턴스의 `C:\ProgramData\Amazon\AwsVss\Logs\`*`timestamp`*`.zip`에 저장합니다 `CollectDiagnosticLogs` 파라미터의 기본 값은 "`False`"입니다.

> **Note**
>
> 디버깅에 대한 추가 지원이 필요한 경우 `.zip` 파일을 지원로 전송할 수 있습니다.

진단 로그 수집 여부에 관계없이 다음과 같은 추가 로그를 사용할 수 있습니다.

- `%ProgramData%\Amazon\SSM\InstanceData\InstanceID\document\orchestration\SSMCommandID\awsrunPowerShellScript\runPowerShellScript\stdout`
- `%ProgramData%\Amazon\SSM\InstanceData\InstanceID\document\orchestration\SSMCommandID\awsrunPowerShellScript\runPowerShellScript\stderr`

이벤트 뷰어 Windows 애플리케이션을 열고 [Windows 로그(Windows Logs)], [애플리케이션(Application)]을 선택하여 추가 로그를 볼 수도 있습니다. EC2 Windows VSS 공급자 및 볼륨 섀도 복사본 서비스의 특정 이벤트를 보려면 **`Ec2VssSoftwareProvider`** 및 **`VSS`** 조건에서 [소스(Source)]로 필터링합니다.

VPC 엔드포인트와 함께 Systems Manager를 사용하고 있고 Systems Manager send-command API 동작(콘솔에서 명령 실행(Run Command))이 실패한 경우 com.amazonaws.***region***.ec2 엔드포인트를 올바르게 구성했는지 확인하세요.

Amazon EC2 엔드포인트가 정의되어 있지 않으면 연결된 EBS 볼륨을 표시하는 호출이 실패하고 이에 따라 Systems Manager 명령에 실패합니다. Systems Manager를 통한 VPC 엔드포인트 설정에 대한 자세한 내용은 AWS Systems Manager 사용 설명서에서 Virtual Private Cloud 엔드포인트 생성을 참조하세요.

추가 진단 로그 수집

Systems Manager send 명령을 사용하여 VSS 스냅샷 문서를 실행할 때 추가 진단 로그를 수집하려면 런타임에 `CollectDiagnosticLogs` 입력 파라미터를 "`True`"로 설정합니다. 문제해결을 할 때에는 이 파라미터를 "`True`"으로 설정하길 권장합니다.

명령줄 예제를 알고 싶다면 다음 탭 중 하나를 선택하세요.

AWS CLI

다음 예제에서는 `AWSEC2-CreateVssSnapshot` Systems Manager 문서를 AWS CLI에서 실행합니다.

```
aws ssm send-command \
    --document-name "AWSEC2-CreateVssSnapshot" \
    --instance-ids "i-1234567890abcdef0" \
    --parameters '{"description":["Example - create diagnostic logs at
 runtime."],"tags":["Key=tag_name,Value=tag_value"],"CollectDiagnosticLogs":
["True"]}'
```

PowerShell

다음 예제에서는 PowerShell에서 AWSEC2-CreateVssSnapshot Systems Manager 문서를 실행합니다.

```
Send-SSMCommand `
    -DocumentName "AWSEC2-CreateVssSnapshot" `
    -InstanceId "i-1234567890abcdef0" `
    -Parameter @{'description'='Example - create diagnostic logs at
 runtime.';'tags'='Key=tag_name,Value=tag_value';'CollectDiagnosticLogs'='True'}
```

프록시가 구성된 인스턴스에서 VSS 사용

프록시를 사용하여 EC2 엔드포인트에 연결하는 인스턴스에서 VSS 기반 EBS 스냅샷을 생성할 때 문제가 발생하면 다음 인스턴스 설정을 확인하세요.

- AWS Tools for Windows PowerShell을 SYSTEM으로 실행하여 인스턴스의 리전과 IMDS에 있는 EC2 서비스 엔드포인트에 연결할 수 있도록 프록시가 구성되어 있는지 확인하세요.
- 시스템에서 구성된 WinHTTP 프록시 사용을 지원하려면 인스턴스에 최신 AwsVssComponents 버전이 설치되어 있어야 합니다. WinHTTP 프록시 구성에 대한 자세한 내용은 Microsoft 웹사이트의 Netsh Commands for Windows Hypertext Transfer Protocol (WINHTTP)을 참조하십시오.

오류: 파이프 연결 재개 시간 초과, 재개 시 오류, VSS 동결 대기 중 시간 초과 또는 기타 시간 초과 오류

인스턴스의 활동 또는 서비스로 인해 VSS 기반 스냅샷이 적시에 진행되지 않아 EC2 Windows VSS 공급자가 시간 초과될 수 있습니다. Windows VSS 프레임워크는 파일 시스템과의 통신이 일시 중지되는 동안 구성 불가능한 10초 기간을 제공합니다. 이 시간 동안 AWSEC2-CreateVssSnapshot은 볼륨의 스냅샷을 생성합니다.

다음 문제는 EC2 Windows VSS 공급자에서 스냅샷 생성 중에 시간 제한이 발생하여 발생할 수 있습니다.

- 볼륨에 대한 과도한 I/O
- 인스턴스에서 EC2 API의 느린 응답
- 조각난 볼륨
- 일부 바이러스 백신 소프트웨어와의 비호환성

- VSS 애플리케이션 기록기 관련 문제
- 많은 수의 PowerShell 모듈에 대해 모듈 로깅을 사용하도록 설정하면 PowerShell 스크립트가 느리게 실행될 수 있습니다.

`AWSEC2-CreateVssSnapshot` 명령 문서를 실행할 때 발생하는 대부분의 제한 시간 초과 문제는 백업할 때 인스턴스의 워크로드가 너무 높은 상황과 관련됩니다. 다음은 스냅샷을 성공적으로 생성하는 데 도움이 될 수 있는 작업입니다.

- `AWSEC2-CreateVssSnapshot` 명령을 다시 시도하여 스냅샷 시도가 성공했는지 확인합니다. 재시도가 성공하는 일부 경우 인스턴스 로드를 줄이면 스냅샷 생성 성공률이 증가할 수 있습니다.
- 인스턴스의 워크로드가 감소할 때까지 잠시 기다린 후 `AWSEC2-CreateVssSnapshot` 명령을 다시 시도합니다. 또는 인스턴스의 스트레스가 낮은 것으로 알려질 때 스냅샷을 시도할 수 있습니다.
- 시스템의 바이러스 백신 소프트웨어가 꺼져 있을 때 VSS 스냅샷을 시도합니다. 이 방법으로 문제가 해결되면 바이러스 백신 소프트웨어 지침을 참조하여 VSS 스냅샷을 허용하도록 구성합니다.
- 스냅샷을 실행하는 동일한 리전 내 계정에서 Amazon EC2 API 직접 호출이 많은 경우 API 제한으로 인해 스냅샷 작업이 지연될 수 있습니다. 제한 영향을 줄이려면 최신 `AwsVssComponents` 패키지를 사용하세요. 이 패키지는 EC2 `CreateSnapshots` API 작업을 활용하여 볼륨별 스냅샷 생성 및 태그 지정과 같은 변경 작업 수를 줄입니다.
- 여러 `AWSEC2-CreateVssSnapshot` 명령 스크립트를 동시에 실행하는 경우 다음 단계를 수행하여 동시 실행 문제를 줄일 수 있습니다.
  - API 활동이 적은 기간에 스냅샷을 예약하는 방법을 고려합니다.
  - Systems Manager 콘솔에서 Run Command 또는 API에서 SendCommand를 사용하여 명령 스크립트를 실행하는 경우 Systems Manager 속도 제어를 사용하여 동시 실행을 줄일 수 있습니다.

    Systems Manager 속도 제어를 사용하면 Systems Manager를 사용하여 명령 스크립트를 실행하는 AWS Backup 등의 서비스에서 동시 실행을 줄일 수도 있습니다.
- `vssadmin list writers` 셸에서 명령을 실행하고 시스템의 기록기에 대한 [마지막 오류(Last error)] 필드에 오류가 보고되는지 확인합니다. 기록기가 시간 초과 오류를 보고하는 경우 인스턴스의 로드가 적을 때 스냅샷을 다시 시도하는 것이 좋습니다.
- *`t2 | t3 | t3a`*.nano 또는 *`t2 | t3 | t3a`*.micro 등의 더 작은 인스턴스 유형을 사용하는 경우 메모리 및 CPU 제약 조건으로 인해 제한 시간이 초과될 수 있습니다. 다음 작업은 제한 시간 초과 문제를 줄이는 데 도움이 될 수 있습니다.
  - 스냅샷을 생성하기 전에 메모리 또는 CPU 사용량이 많은 애플리케이션을 닫아보세요.
  - 인스턴스 활동이 적은 시간대에 스냅샷을 생성하세요.

오류: 메서드를 호출할 수 없습니다. 메서드 호출은 이 언어 모드의 핵심 유형에서만 지원됩니다.

PowerShell 언어 모드가 `FullLanguage`로 설정되지 않은 경우 이 오류가 발생합니다. `AWSEC2-CreateVssSnapshot` SSM 문서를 사용하려면 PowerShell을 `FullLanguage` 모드로 구성해야 합니다.

언어 모드를 확인하려면 PowerShell 콘솔의 인스턴스에서 다음 명령을 실행합니다.

```
$ExecutionContext.SessionState.LanguageMode
```

언어 모드에 대한 자세한 내용은 Microsoft 설명서의 about_Language_Modes를 참조하세요.

AWS VSS 솔루션을 사용하여 인스턴스의 데이터 복원

AWS VSS 솔루션에서 생성한 VSS 기반 스냅샷에서 Windows 인스턴스의 EBS 볼륨을 복원할 수 있습니다. AWS VSS 솔루션 스냅샷에 Microsoft SQL Server 데이터베이스의 백업이 포함된 경우 `AWSEC2-RestoreSqlServerDatabaseWithVss` AWS Systems Manager 자동화 런북을 사용하여 데이터베이스를 복원할 수 있습니다.

데이터베이스 복원 런북은 스냅샷에서 볼륨을 생성하고 이를 인스턴스에 연결하는 등 전체 복원 프로세스를 자동화합니다. 자동화는 VSS 기술을 활용하여 데이터베이스를 복원하므로 SQL Server 애플리케이션을 중지하거나 활성 연결을 해제하지 않고도 복원할 수 있습니다.

Microsoft SQL Server 데이터베이스 런북을 사용하는 방법에 대한 자세한 지침은 Amazon EC2 사용 설명서의 Microsoft SQL Server에서 Restore from VSS based snapshots를 참조하세요.

AWS VSS 솔루션 스냅샷에서 EBS 볼륨을 복원하도록 스크립트 사용자 지정

`RestoreVssSnapshotSampleScript.ps1` 스크립트를 모델로 사용하여 AWS VSS 솔루션 스냅샷에서 EBS 볼륨을 복원하는 자체 사용자 지정 스크립트를 생성할 수 있습니다. 이 샘플 스크립트는 다음 작업을 수행합니다.

- 인스턴스 중지
- 인스턴스에서 기존 드라이브를 모두 제거(부팅 볼륨이 제외된 경우 제외된 부팅 볼륨은 제외)
- 스냅샷에서 새 볼륨 생성
- 스냅샷의 디바이스 ID 태그를 사용하여 볼륨을 인스턴스에 연결
- 인스턴스를 다시 시작

> **Important**
>
> 다음 스크립트는 인스턴스에 연결된 모든 볼륨을 분리한 후 스냅샷에서 새 볼륨을 만듭니다. 인스턴스를 적당히 백업했는지 확인하세요. 이전 볼륨은 삭제되지 않습니다. 삭제하고 싶다면 이전 볼륨을 삭제하도록 스크립트를 편집할 수 있습니다.

VSS 기반 EBS 스냅샷에서 볼륨을 복구하려면 다음을 수행하세요.

1. RestoreVssSnapshotSampleScript.zip 파일을 다운로드한 후에 파일 콘텐츠의 압축을 풉니다.
2. 텍스트 에디터에서 `RestoreVssSnapshotSampleScript.ps1`을 열고 유효한 EC2 인스턴스 ID, EBS 스냅샷 ID를 사용하여 스크립트 하단에서 샘플 호출을 편집한 다음 PowerShell에서 스크립트를 실행합니다.

AWS VSS 솔루션 버전 기록

이 페이지에는 AWS VSS 구성 요소 패키지의 버전별 릴리스 정보와 지원되는 각 Windows Server 버전의 구성 요소 및 스크립트 버전 요구 사항이 포함되어 있습니다.

주제

- AwsVssComponents 패키지 버전
- Windows OS 버전 지원

AwsVssComponents 패키지 버전

다음 표에서는 AWS VSS 구성 요소 패키지의 릴리스 버전에 대해 설명합니다.

버전	세부 정보	릴리스 날짜	다운로드 가능
2.5.1	대상 데이터베이스 파라미터를 지정할 때 SQL 데이터베이스 복원이 실패할 수 있는 문제를 해결했습니다.	2025년 3월 13일	예
2.5.0	• VSS 메타데이터 파일을 읽고 인스턴스에서 Microsoft SQL Server 데이터베이스를 복원하는 기능	2025년 1월 17일	예

버전	세부 정보	릴리스 날짜	다운로드 가능
	이 추가되었습니다. 자세한 내용은 Amazon EC2 사용 설명서의 Microsoft SQL Server에서 Restore from VSS based snapshots를 참조하세요. • `AwsVssComponents` 패키지를 설치하거나 업그레이드할 때 현재 위치 업데이트 옵션에 대한 지원이 추가되었습니다.		
2.4.0	스냅샷 생성 시 VSS 메타데이터 파일을 저장하는 기능이 추가되었습니다. 이 기능을 활성화하려면 Systems Manager VSS 스냅샷 문서의 파라미터에서 SaveVssMetadata를 참조하세요.	2024년 10월 7일	예
2.3.3	스냅샷을 생성하는 동안 `Ec2VssProvider` 가 사용되도록 VSS 에이전트를 업데이트했습니다.	2024년 6월 25일	예
2.3.2	제거 시 VSS 공급자 등록이 제거되지 않는 경우를 수정했습니다.	2024년 5월 9일	예
2.3.1	AWS VSS에서 생성한 스냅샷 및 AMI를 식별하기 위해 새 기본 태그(`AwsVssConfig`)가 추가되었습니다.	2024년 3월 7일	예

버전	세부 정보	릴리스 날짜	다운로드 가능
2.2.1	• `DescribeInstanceAttribute` API 사용에 대한 지원 추가됨. • 버그 수정 및 안정성 향상. • Windows Server 2012 및 2012 R2에 대한 지원 중단됨. AWS Windows Server 2012 및 2012 R2에 VSS 구성 요소 버전 2.2.1 설치가 실패합니다. AWS VSS 구성 요소 버전 2.1.0은 Windows Server 2012 및 2012 R2를 지원하는 마지막 버전입니다.	2024년 1월 18일	예
2.1.0	`CreateSnapshots` API 사용에 대한 지원 추가됨.	2023년 11월 6일	예
2.0.1	WinHTTP 프록시 설정 사용에 대한 지원이 추가되었습니다.	2023년 10월 26일	아니요
2.0.0	AWS VSS 구성 요소에 스냅샷 및 AMI를 생성하는 기능이 추가되어 PowerShell 모듈 로깅, 스크립트 블록 로깅 및 트랜스크립션 기능과의 호환성을 지원합니다.	2023년 4월 28일	아니요
1.3.2.0	설치 실패가 제대로 보고되지 않는 경우가 수정되었습니다.	2022년 5월 10일	아니요
1.3.1.0	• NTDS VSS 라이터 로깅 오류와 관련하여 도메인 컨트롤러에서 실패하는 스냅샷을 수정했습니다. • 버전 1.0 VSS 공급자를 제거할 때의 VSS 에이전트 오류를 해결했습니다.	2020년 2월 6일	예

버전	세부 정보	릴리스 날짜	다운로드 가능
1.3.00	• 원하지 않는 세부 사항을 줄여 로깅을 개선했습니다. • 설치 중 지역화 문제를 해결했습니다. • 일부 공급자 등록 오류 조건에 대한 반환 코드를 수정했습니다. • 다양한 설치 문제를 해결했습니다.	2019년 3월 19일	아니요
1.2.00	• 에이전트에 명령줄 파라미터 -nw(라이터 없음) 및 -copy(복사 전용)를 추가했습니다. • 부적절한 메모리 할당 호출로 인해 발생하는 EventLog 오류를 해결했습니다.	2018년 11월 15일	아니요
1.1	AWS VSS 구성 요소가 기본 Windows 백업 및 복원 공급자로 잘못 사용되는 문제가 해결되었습니다.	2017년 12월 12일	아니요
1.0	최초 릴리스.	2017년 11월 20일	아니요

Windows OS 버전 지원

다음 표에서는 Amazon EC2의 각 Windows Server 버전에서 어떤 AWS VSS 솔루션 버전을 실행해야 하는지를 보여줍니다.

Windows Server 버전	AwsVssComponents 버전	AWSEC2-VssInstallAndSnapshot 버전 이름	AWSEC2-CreateVssSnapshot 버전 이름
Windows Server 2025	기본값	기본값	기본값
Windows Server 2022	기본값	기본값	기본값
Windows Server 2019	기본값	기본값	기본값
Windows Server 2016	기본값	기본값	기본값
Windows Server 2012 R2	2.1.0	지원되지 않음	2012R2
Windows Server 2012	2.1.0	지원되지 않음	2012R2
Windows Server 2008 R2	1.3.1.0	지원되지 않음	2008R2

Amazon EC2의 객체 스토리지, 파일 스토리지 및 파일 캐싱

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 인스턴스에 있는 단일 노드 파일 서버에서 직접 클러스터링하는 솔루션 및 완전 관리형 솔루션에 이르는 다양한 파일 스토리지 솔루션이 존재합니다. 다음 콘텐츠에서는 Amazon EC2 인스턴스와 함께 사용할 수 있도록 AWS에서 제공하는 스토리지 서비스 중 일부를 소개합니다.

내용

- Amazon EC2 인스턴스에서 Amazon S3 사용
- Amazon EC2 Linux 인스턴스에서 Amazon EFS 사용
- Amazon EC2 인스턴스에서 Amazon FSx 사용
- Amazon EC2 인스턴스에서 Amazon File Cache 사용

Amazon EC2 인스턴스에서 Amazon S3 사용

Amazon Simple Storage Service(Amazon S3)는 업계 최고의 확장성, 데이터 가용성, 보안 및 성능을 제공하는 객체 스토리지 서비스입니다. Amazon S3를 사용하여 Amazon EC2 인스턴스 또는 인터넷을 통해 데이터 레이크, 웹 사이트, 백업 및 빅 데이터 분석과 같은 다양한 사용 사례에 대해 원하는 양의 데이터를 저장하고 검색할 수 있습니다. 자세한 내용은 Amazon S3란 무엇인가?를 참조하세요

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 `amzn-s3-demo-bucket1` 버킷에 저장되며, 다음 URL을 사용하여 주소를 지정할 수 있습니다. `https://amzn-s3-demo-bucket1.s3.amazonaws.com/photos/mygarden.jpg` 자세한 내용은 Amazon S3 작동 방식을 참조하세요.

사용 예제:

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 받

는 방법은 여러가지가 있습니다. 아래 설명한 예뿐만 아니라 여러 사람들이 작성한 다양한 도구가 있으며, 이를 사용해서 컴퓨터 또는 인스턴스에서 Amazon S3의 데이터에 액세스할 수 있습니다.

권한을 부여받은 경우, 다음 방법 중 하나를 사용해서 Amazon S3 및 인스턴스로 또는 인스턴스로부터 파일을 복사할 수 있습니다.

wget

> **Note**
>
> 이 메서드는 퍼블릭 객체에만 적용됩니다. 객체가 퍼블릭이 아닌 경우 `ERROR 403: Forbidden` 메시지가 표시됩니다. 이 오류가 발생하는 경우 Amazon S3 콘솔, AWS CLI, AWS API, AWS SDK 또는 AWS Tools for Windows PowerShell을 사용해야 하며, 필요한 권한이 있어야 합니다. 자세한 내용은 Amazon S3 사용 설명서에서 Amazon S3의 자격 증명 및 액세스 관리 및 객체 다운로드를 참조하세요.

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://amzn-s3-demo-bucket.s3.amazonaws.com/path-to-file
```

PowerShell

AWS Tools for Windows PowerShell을 사용하여 Amazon S3 사이에서 객체를 이동할 수 있습니다.

다음과 같이 Copy-S3Object cmdlet을 사용해서 Amazon S3 객체를 Windows 인스턴스로 복사합니다.

```
Copy-S3Object `
    -BucketName amzn-s3-demo-bucket `
    -Key path-to-file `
    -LocalFile my_copied_file.ext
```

또는 Windows 인스턴스에서 웹 브라우저를 사용하여 Amazon S3 콘솔을 열 수 있습니다.

AWS CLI

AWS Command Line Interface(AWS CLI)를 사용하여 Amazon S3에서 제한된 항목을 다운로드하고 항목을 업로드할 수도 있습니다. 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.

```
aws s3 cp s3://amzn-s3-demo-bucket/my_folder/my_file.ext my_copied_file.ext
```

다음 명령을 사용해서 인스턴스에서 Amazon S3로 객체를 복사합니다.

```
aws s3 cp my_copied_file.ext s3://amzn-s3-demo-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.

```
aws s3 sync s3://amzn-s3-demo-source-bucket local_directory
```

Amazon S3 API

개발자라면 API를 사용해서 Amazon S3의 데이터에 액세스할 수 있습니다. 이런 API를 사용해서 애플리케이션 개발을 지원하고 이를 다른 API 및 SDK와 통합할 수 있습니다. 자세한 내용은 Amazon Simple Storage Service API 참조의 AWS SDK를 사용하는 Amazon S3의 코드 예제를 참조하세요.

Amazon EC2 Linux 인스턴스에서 Amazon EFS 사용

> **Note**
>
> Amazon EFS는 Windows 인스턴스에서 지원되지 않습니다.

Amazon EFS는 Amazon EC2에서 사용할 수 있는 확장 가능한 파일 스토리지를 제공합니다. 하나의 EFS 파일 시스템을 여러 인스턴스에서 실행하는 워크로드 및 애플리케이션에 대한 공통 데이터 소스로 사용할 수 있습니다. 자세한 내용은 Amazon Elastic File System 제품 페이지를 참조하세요.

이 자습서에서는 인스턴스 시작 중 Amazon EFS 빠른 생성 마법사를 사용하여 Amazon EFS 파일 시스템을 생성하고 연결하는 방법을 보여줍니다. Amazon EFS 콘솔을 사용하여 파일 시스템을 생성하는 방법에 대한 자습서를 알아보려면 Amazon Elastic File System User Guide(Amazon Elastic File System 사용 설명서)의 Getting started with Amazon Elastic File System(Amazon Elastic File System 시작하기)을 참조하세요.

> **Note**
>
> EFS Quick Create를 사용하여 EFS 파일 시스템을 생성하는 경우 파일 시스템은 다음과 같은 서비스 권장 설정으로 생성됩니다.
>
> - 자동 백업 활성화
> - 선택한 VPC에서 탑재 대상을 관리합니다.
> - 범용 성능 모드
> - 버스팅 처리량 모드
> - Amazon EFS용 기본 키를 사용하여 저장 데이터 암호화 활성화(`aws/elasticfilesystem`)
> - 30일 정책으로 Amazon EFS 수명 주기 관리 활성화

업무

- Amazon EFS Quick Create를 사용하여 EFS 파일 시스템 생성
- EFS 파일 시스템 테스트
- EFS 파일 시스템 삭제

Amazon EFS Quick Create를 사용하여 EFS 파일 시스템 생성

Amazon EC2 인스턴스 시작 마법사의 Amazon EFS Quick Create 기능을 사용하여 인스턴스를 시작할 때 EFS 파일 시스템을 생성하고 인스턴스에 탑재할 수 있습니다.

Amazon EFS Quick Create를 사용하여 EFS 파일 시스템을 생성하려면

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

2. 인스턴스 시작을 선택합니다.
3. (선택 사항) Name and tags(이름 및 태그) 아래의 Name(이름)에 인스턴스를 식별하는 이름을 입력합니다.
4. Application and OS Images (Amazon Machine Image)(애플리케이션 및 OS 이미지(Amazon Machine Image))에서 Linux 운영 체제를 선택한 다음 Amazon Machine Image (AMI)(Amazon Machine Image(AMI))에 대해 Linux AMI를 선택합니다.
5. Instance type(인스턴스 유형)에서 Instance type(인스턴스 유형)에서 인스턴스 유형을 선택하거나 기본값을 유지합니다.
6. (선택 사항) 키 페어(로그인)(Key pair (login)) 아래의 키 페어 이름(Key pair name)에서 기존 키 페어를 선택하거나 새로 생성합니다.
7. Network settings(네트워크 설정)에서 Edit(편집)(오른쪽)을 선택한 다음 Subnet(서브넷)에서 서브넷을 선택합니다.

> **Note**
>
> EFS 파일 시스템을 추가하려면 먼저 서브넷을 선택해야 합니다.

8. Configure storage(스토리지 구성)에서 Edit(편집)(오른쪽 하단)을 선택한 후 다음을 수행합니다.
   a. 파일 시스템에서 EFS가 선택되었는지 확인한 다음, 새 공유 파일 시스템 생성을 선택합니다.
   b. 파일 시스템 이름에 Amazon EFS 파일 시스템의 이름을 입력하고 파일 시스템 생성을 선택합니다.
   c. 탑재 지점에서 사용자 지정 탑재 지점을 지정하거나 기본값을 유지합니다.
   d. 파일 시스템에 대한 액세스를 활성화하려면 Automatically create and attach security groups(보안 그룹 자동 생성 및 연결)를 선택합니다. 이 확인란을 선택하면 다음 보안 그룹이 자동으로 생성되어 파일 시스템의 인스턴스 및 탑재 대상에 연결됩니다.
      - 인스턴스 보안 그룹 - NFS 2049 포트를 통한 트래픽을 허용하지만 인바운드 규칙은 포함하지 않는 아웃바운드 규칙을 포함합니다.
      - 파일 시스템 탑재 대상 보안 그룹 - 위에서 설명한 인스턴스 보안 그룹에서 NFS 2049 포트를 통한 트래픽을 허용하는 인바운드 규칙과 NFS 2049 포트를 통한 트래픽을 허용하는 아웃바운드 규칙을 포함합니다.

> **Note**
>
> 또는 보안 그룹을 수동으로 생성하고 연결할 수 있습니다. 보안 그룹을 수동으로 만들고 연결하려면 Automatically create and attach the required security groups(자동으로 필요한 보안 그룹 생성 및 연결)을 선택 취소합니다.

e. 인스턴스가 시작될 때 공유 파일 시스템을 자동으로 탑재하려면 Automatically mount shared file system by attaching required user data script(필수 사용자 데이터 스크립트를 연결하여 공유 파일 시스템 자동 탑재)를 선택합니다. 자동으로 생성된 사용자 데이터를 보려면 Advanced details(고급 세부 정보)를 확장하고 User data(사용자 데이터)까지 아래로 스크롤합니다.

> **Note**
>
> 이 확인란을 선택하기 전에 사용자 데이터를 추가한 경우 자동으로 생성된 사용자 데이터가 원래 사용자 데이터를 덮어씁니다.

9. 필요에 따라 다른 인스턴스 구성 설정을 구성합니다.
10. Summary(요약) 패널에서 인스턴스 구성을 검토한 다음 Launch instance(인스턴스 시작)를 선택합니다. 자세한 내용은 콘솔에서 인스턴스 시작 마법사를 사용하여 EC2 인스턴스 시작 단원을 참조하십시오.

EFS 파일 시스템 테스트

인스턴스에 연결하여 지정한 디렉터리(예: /mnt/efs)에 해당 파일 시스템이 탑재되었는지 확인할 수 있습니다.

파일 시스템이 마운트되었는지 확인하려면

1. 인스턴스에 연결합니다. 자세한 내용은 SSH를 사용하여 Linux 인스턴스에 연결 단원을 참조하십시오.
2. 인스턴스의 터미널 창에서 df -T 명령을 실행하여 EFS 파일 시스템이 탑재되었는지 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) 인스턴스의 파일 시스템에서 파일을 하나 생성한 후 또 다른 인스턴스에서 해당 파일이 보이는지 확인합니다.

   a. 인스턴스에서 다음 명령을 실행하여 파일을 생성합니다.

   ```
   $ sudo touch /mnt/efs/test-file.txt
   ```

   b. 다른 인스턴스에서 다음 명령을 실행하여 파일을 봅니다.

   ```
   $ ls /mnt/efs
   test-file.txt
   ```

EFS 파일 시스템 삭제

파일 시스템이 더 이상 필요하지 않으면 삭제할 수 있습니다.

파일 시스템을 삭제하려면

1. Amazon Elastic File System 콘솔(https://console.aws.amazon.com/efs/)을 엽니다.
2. 삭제한 파일 시스템을 선택합니다.
3. 작업, 파일 시스템 삭제를 차례로 선택합니다.
4. 확인 메시지가 표시되면 파일 시스템 ID를 입력하고 파일 시스템 삭제(Delete file system)를 선택합니다.

Amazon EC2 인스턴스에서 Amazon FSx 사용

Amazon FSx 서비스 패밀리에서는 인기 있는 상용 및 오픈 소스 파일 시스템에서 제공되는 공유 스토리지를 쉽게 시작하고 실행하며 크기를 조정할 수 있습니다. 새로운 인스턴스 시작 마법사를 사용하여 시작 시 다음과 같은 Amazon FSx 파일 시스템 유형을 자동으로 Amazon EC2 인스턴스에 연결할 수 있습니다.

- Amazon FSx for NetApp ONTAP에서는 NetApp ONTAP의 인기 있는 데이터 액세스 및 관리 기능과 함께 AWS 클라우드의 완전관리형 공유 스토리지가 제공됩니다.
- Amazon FSx for OpenZFS에서는 인기 있는 OpenZFS 파일 시스템에서 제공되는 비용 효율적인 완전관리형 공유 스토리지가 제공됩니다.

> **Note**
>
> - 이 기능은 새로운 인스턴스 시작 마법사에서만 사용할 수 있습니다. 자세한 내용은 콘솔에서 인스턴스 시작 마법사를 사용하여 EC2 인스턴스 시작 섹션을 참조하세요.
> - Amazon FSx for Windows File Server 및 Amazon FSx for Lustre 파일 시스템은 시작 시 탑재할 수 없습니다. 이러한 파일 시스템은 시작 후 수동으로 탑재해야 합니다.

이전에 생성한 기존 파일 시스템이 탑재되도록 선택하거나 시작 시 인스턴스에 탑재되는 새 파일 시스템을 생성할 수 있습니다.

주제

- 보안 그룹 및 사용자 데이터 스크립트
- 시작 시 Amazon FSx 파일 시스템 탑재

보안 그룹 및 사용자 데이터 스크립트

인스턴스 시작 마법사를 사용하여 Amazon FSx 파일 시스템을 인스턴스에 탑재할 때 파일 시스템에 대한 액세스를 활성화하는 데 필요한 보안 그룹이 자동으로 생성되어 연결되는지 여부와 파일 시스템을 탑재하여 사용할 수 있도록 만드는 데 필요한 사용자 데이터 스크립트가 자동으로 포함되는지 여부를 선택할 수 있습니다.

주제

- 보안 그룹
- 사용자 데이터 스크립트

보안 그룹

파일 시스템에 대한 액세스를 활성화하는 데 필요한 보안 그룹이 자동으로 생성되도록 선택하면 하나는 인스턴스에 연결되고 다른 하나는 파일 시스템에 연결되는 2개의 보안 그룹이 인스턴스 시작 마법

사를 통해 생성되고 연결됩니다. 보안 그룹 요구 사항에 대한 자세한 내용은 Amazon VPC로 FSx for ONTAP 파일 시스템 액세스 제어 섹션과 Amazon VPC로 FSx for OpenZFS 파일 시스템 액세스 제어 섹션을 참조하세요.

생성되어 인스턴스에 연결된 보안 그룹에 `Name=instance-sg-1` 태그를 추가합니다. 태그의 값은 Amazon FSx 파일 시스템에 대한 보안 그룹이 인스턴스 시작 마법사를 통해 생성될 때마다 자동으로 증분됩니다.

보안 그룹에는 다음과 같은 출력 규칙이 포함되지만 인바운드 규칙은 포함되지 않습니다.

아웃바운드 규칙

프로토콜 유형	포트 번호	대상
UDP	111	## ### ## ##
UDP	20001~20003	## ### ## ##
UDP	4049	## ### ## ##
UDP	2049	## ### ## ##
UDP	635	## ### ## ##
UDP	4045~4046	## ### ## ##
TCP	4049	## ### ## ##
TCP	635	## ### ## ##
TCP	2049	## ### ## ##
TCP	111	## ### ## ##
TCP	4045~4046	## ### ## ##
TCP	20001~20003	## ### ## ##
모두	모두	## ### ## ##

생성되어 파일 시스템에 연결된 보안 그룹에는 `Name=fsx-sg-1` 태그가 지정됩니다. 태그의 값은 Amazon FSx 파일 시스템에 대한 보안 그룹이 인스턴스 시작 마법사를 통해 생성될 때마다 자동으로 증분됩니다.

보안 그룹에는 다음 규칙이 포함됩니다.

인바운드 규칙

프로토콜 유형	포트 번호	소스
UDP	2049	#### ## ##
UDP	20001~20003	#### ## ##
UDP	4049	#### ## ##
UDP	111	#### ## ##
UDP	635	#### ## ##
UDP	4045~4046	#### ## ##
TCP	4045~4046	#### ## ##
TCP	635	#### ## ##
TCP	2049	#### ## ##
TCP	4049	#### ## ##
TCP	20001~20003	#### ## ##
TCP	111	#### ## ##

아웃바운드 규칙

프로토콜 유형	포트 번호	대상
모두	모두	0.0.0.0/0

사용자 데이터 스크립트

사용자 데이터 스크립트가 자동으로 연결되도록 선택하면 인스턴스 시작 마법사를 통해 다음과 같은 사용자 데이터가 인스턴스에 추가됩니다. 이 스크립트에서는 인스턴스가 다시 시작될 때마다 파일 시스템이 자동으로 다시 탑재되도록 이 스크립트를 통해 필요한 패키지를 설치하고, 파일 시스템을 탑재하고, 인스턴스 설정을 업데이트할 수 있습니다.

```
#cloud-config
package_update: true
package_upgrade: true
runcmd:
- yum install -y nfs-utils
- apt-get -y install nfs-common
- svm_id_1=svm_id
- file_system_id_1=file_system_id
- vol_path_1=/vol1
- fsx_mount_point_1=/mnt/fsx/fs1
- mkdir -p "${fsx_mount_point_1}"
- if [ -z "$svm_id_1" ]; then printf "\n${file_system_id_1}.fsx.eu-
north-1.amazonaws.com:/${vol_path_1} ${fsx_mount_point_1} nfs4
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport,_netdev
 0 0\n" >> /etc/fstab; else printf "\n${svm_id_1}.${file_system_id_1}.fsx.eu-
north-1.amazonaws.com:/${vol_path_1} ${fsx_mount_point_1} nfs4
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport,_netdev 0
 0\n" >> /etc/fstab; fi
- retryCnt=15; waitTime=30; while true; do mount -a -t nfs4 defaults; if [ $? = 0 ] ||
 [ $retryCnt -lt 1 ]; then echo File system mounted successfully; break; fi; echo File
 system not available, retrying to mount.; ((retryCnt--)); sleep $waitTime; done;
```

시작 시 Amazon FSx 파일 시스템 탑재

시작 시 신규 또는 기존 Amazon FSx 파일 시스템을 탑재하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스(Instances)를 선택한 다음에 인스턴스 시작(Launch instance)을 선택하여 인스턴스 시작 마법사를 엽니다.
3. 애플리케이션 및 OS 이미지(Application and OS Images) 섹션에서 사용할 AMI를 선택합니다.
4. 인스턴스 유형(Instance type) 섹션에서 인스턴스 유형을 선택합니다.
5. 키 페어(Key pair) 섹션에서 기존 키 페어를 선택하거나 새로 하나를 생성합니다.
6. 네트워크 설정(Network settings) 섹션에서 다음을 수행합니다.

a. 편집을 선택합니다.

b. 기존 파일 시스템을 탑재하려면 서브넷(Subnet)의 경우 파일 시스템의 기본 설정 서브넷을 선택합니다. 성능을 최적화하려면 파일 시스템의 기본 설정 서브넷과 동일한 가용 영역에서 인스턴스를 시작하는 것이 좋습니다.

새 파일 시스템을 생성하여 인스턴스에 탑재하려면 서브넷(Subnet)의 경우 인스턴스를 시작할 서브넷을 선택합니다.

> **Important**
>
> 새로운 인스턴스 시작 마법사에서 Amazon FSx 기능을 사용하려면 서브넷을 선택해야 합니다. 서브넷을 선택하지 않으면 기존 파일 시스템을 탑재하거나 새 파일 시스템을 생성할 수 없습니다.

7. 스토리지(Storage) 섹션에서 다음을 수행합니다.

a. 필요에 따라 볼륨을 구성합니다.

b. 파일 시스템(File systems) 섹션을 확장하여 FSx를 선택합니다.

c. 공유 파일 시스템 추가(Add shared file system)를 선택합니다.

d. 파일 시스템(File system)의 경우 탑재할 파일 시스템을 선택합니다.

> **Note**
>
> 선택한 리전의 계정에 있는 모든 Amazon FSx for NetApp ONTAP 및 Amazon FSx for OpenZFS 파일 시스템이 목록에 표시됩니다.

e. 파일 시스템에 대한 액세스를 활성화하는 데 필요한 보안 그룹이 자동으로 생성되어 연결되도록 하려면 자동으로 보안 그룹 생성 및 연결(Automatically create and attach security groups)을 선택합니다. 보안 그룹을 수동으로 생성하려면 확인란의 선택을 취소합니다. 자세한 내용은 보안 그룹 단원을 참조하십시오.

f. 파일 시스템을 탑재하는 데 필요한 사용자 데이터 스크립트가 자동으로 첨부되도록 하려면 필요한 사용자 데이터 스크립트를 첨부하여 자동으로 공유 파일 시스템 탑재(Automatically mount shared file system by attaching required user data script)를 선택합니다. 사용자 데이터 스크립트를 수동으로 제공하려면 확인란의 선택을 취소합니다. 자세한 내용은 사용자 데이터 스크립트 단원을 참조하십시오.

8. 고급(Advanced) 섹션에서 필요에 따라 추가 인스턴스 설정을 구성합니다.

9. 시작을 선택합니다.

Amazon EC2 인스턴스에서 Amazon File Cache 사용

Amazon File Cache는 AWS에서 완전관리형 고속 캐시를 제공하여 데이터 저장 위치에 관계없이 파일 데이터를 보다 쉽게 처리할 수 있도록 지원합니다. Amazon File Cache는 온프레미스 파일 시스템, AWS 파일 시스템 및 Amazon Simple Storage Service(S3) 버킷에 저장된 데이터를 위한 임시 고성능 스토리지 위치 역할을 합니다. 이 기능을 사용하면 통합 보기를 통해 빠른 속도(1밀리초 미만의 지연 시간과 높은 처리량)로 AWS의 파일 기반 애플리케이션에 분산된 데이터 세트를 사용할 수 있습니다. 자세한 내용은 Amazon File Cache 사용 설명서를 참조하세요.

Amazon File Cache는 가장 널리 사용되는 Linux AMI에서 작동하며, x86 기반 인스턴스 유형 및 Graviton 인스턴스 유형과 호환됩니다. 오픈 소스 Lustre 클라이언트를 사용하여 Amazon EC2 인스턴스에서 캐시에 액세스할 수 있습니다. 캐시를 탑재한 다음 표준 Linux 명령을 사용하여 캐시에 있는 파일 및 디렉터리로 작업할 수 있습니다. 네트워크 구성이 VPC 내의 서브넷 전체에 대한 액세스를 허용하는 경우 Amazon EC2 인스턴스는 동일한 Virtual Private Cloud(VPC) 내의 다른 가용 영역에서 캐시에 액세스할 수 있습니다. 공유 VPC에 캐시를 생성할 수도 있습니다.

시작하려면 Amazon File Cache User Guide의 Getting started with Amazon File Cache를 참조하세요.

Amazon EC2 리소스 관리

리소스는 사용할 수 있는 엔터티입니다. Amazon EC2는 서비스의 기능을 사용할 때 리소스를 생성합니다. 예를 들어 Amazon EC2 리소스에는 이미지, 인스턴스, 플릿, 키 페어, 보안 그룹이 있습니다. 모든 Amazon EC2 리소스 유형에는 리소스를 설명하는 속성이 포함되어 있습니다. 예를 들어 이름, 설명, 리소스 식별자, Amazon 리소스 이름(ARN)이 있습니다.

Amazon EC2 리소스는 상주하는 AWS 리전 또는 영역에서만 사용됩니다. 예를 들어, Amazon Machine Image(AMI)는 AWS 리전에 한정되지만, AMI에서 시작하는 인스턴스는 이를 시작하는 영역에 한정됩니다. ARN을 사용하여 권한 정책에서 Amazon EC2 리소스를 지정할 수 있습니다.

AWS 계정에는 Amazon EC2에 대한 기본 할당량이 있습니다. 이러한 할당량은 생성할 수 있는 최대 리소스 수를 정의합니다. 예를 들어, 실행 중인 인스턴스 전반의 최대 vCPU 수에 대한 할당량이 있습니다. 인스턴스를 시작하거나 중지된 인스턴스를 시작할 때 할당량을 초과하게 되면 작업이 실패합니다.

리소스 ID나 태그를 사용하여 리전별로 AWS 계정의 특정 리소스를 검색할 수 있습니다. 여러 리전에서 특정 리소스 또는 리소스 유형을 검색하려면 Amazon EC2 Global View를 사용합니다.

내용

- Amazon EC2 리소스에 대한 리전 선택
- Amazon EC2 리소스 찾기
- Amazon EC2 Global View를 사용하여 리전 간 리소스 보기
- Amazon EC2 리소스 태깅
- Amazon EC2 서비스 할당량

Amazon EC2 리소스에 대한 리전 선택

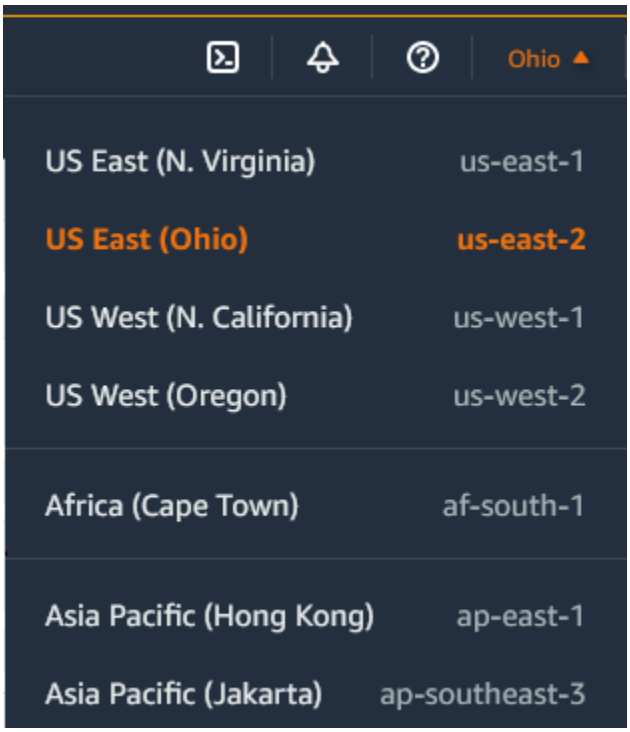
Amazon EC2 리소스는 상주하는 AWS 리전 또는 영역에서만 사용됩니다. Amazon EC2 리소스를 생성할 때 리소스에 대한 리전을 선택합니다.

고려 사항

- AWS 리소스는 일부 리전에서 사용할 수 없습니다. 리전에서 AWS 리소스 생성을 시작하기 전에 리전에서 필요한 리소스를 모두 생성할 수 있는지 확인합니다.
- 일부 리전은 기본적으로 비활성화되어 있습니다. 이 리전을 사용하려면 먼저 활성화해야 합니다. 자세한 내용은 AWS 리전을 참조하세요.

콘솔을 사용하여 리소스에 대한 리전을 선택하려면 다음을 수행하세요.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 모음에서 Regions(리전) 선택기를 선택한 다음 리전을 선택합니다.



3. 리전 선택기에는 해당 AWS 계정에서 사용할 수 있는 모든 리소스가 포함됩니다. 목록 하단에서 밑줄이 그어진 텍스트를 선택하면 계정에 사용할 수 없는 리전을 확인할 수 있습니다.

AWS CLI를 사용하여 리소스의 리전을 선택하려면

기본 리전을 사용하도록 AWS CLI를 구성할 수 있습니다. 명령에서 리전을 지정하지 않으면 기본 리전이 사용됩니다. 특정 명령에 다른 리전을 사용하려면 다음 옵션을 추가합니다.

```
--region us-east-1
```

Tools for PowerShell을 사용하여 리소스의 리전을 선택하려면

기본 리전을 사용하도록 Tools for Windows PowerShell을 구성할 수 있습니다. 명령에서 리전을 지정하지 않으면 기본 리전이 사용됩니다. 특정 명령에 다른 리전을 사용하려면 다음 파라미터를 추가합니다.

```
-Region us-east-1
```

Amazon EC2 리소스 찾기

Amazon EC2 콘솔을 사용하여 리소스의 유형 목록을 얻을 수 있습니다. 사용자는 해당 명령 또는 API 작업을 이용하여 리소스의 각 유형 목록을 획득할 수 있습니다. 리소스가 많은 경우에 결과를 필터링하여 특정 기준에 부합하는 리소스만 포함시키거나 제외할 수 있습니다.

목차

- 콘솔을 사용하여 리소스 나열 및 필터링
- 명령줄을 사용하여 나열 및 필터링
- Amazon EC2 Global View를 사용하여 리전 간 리소스 보기

콘솔을 사용하여 리소스 나열 및 필터링

목차

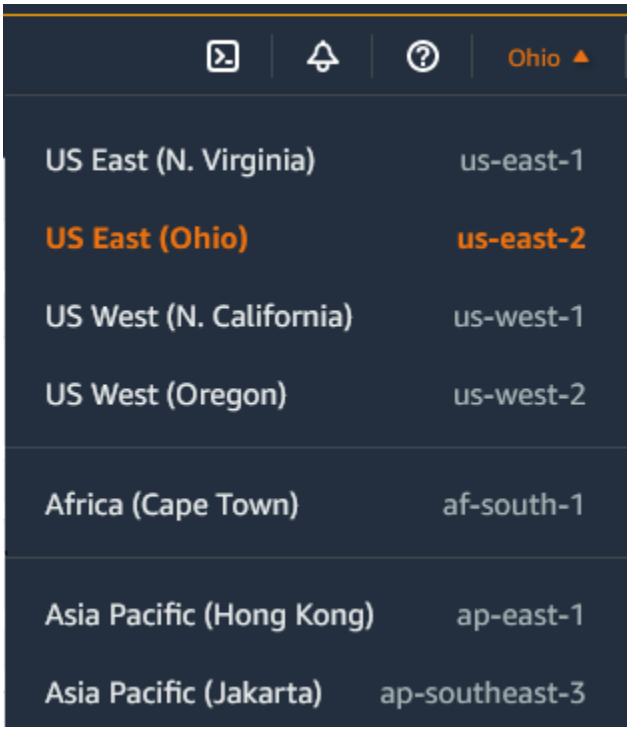
- 콘솔을 사용하여 리소스 나열
- 콘솔을 사용하여 리소스 필터링
  - 지원되는 필터
- 콘솔을 사용하여 필터 세트 저장
  - 주요 기능
  - 필터 세트 생성
  - 필터 세트 수정
  - 필터 세트 삭제

콘솔을 사용하여 리소스 나열

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. Amazon EC2 리소스는 AWS 리전에 한정됩니다. 탐색 모음의 리전 선택기에서 리전을 선택합니다.

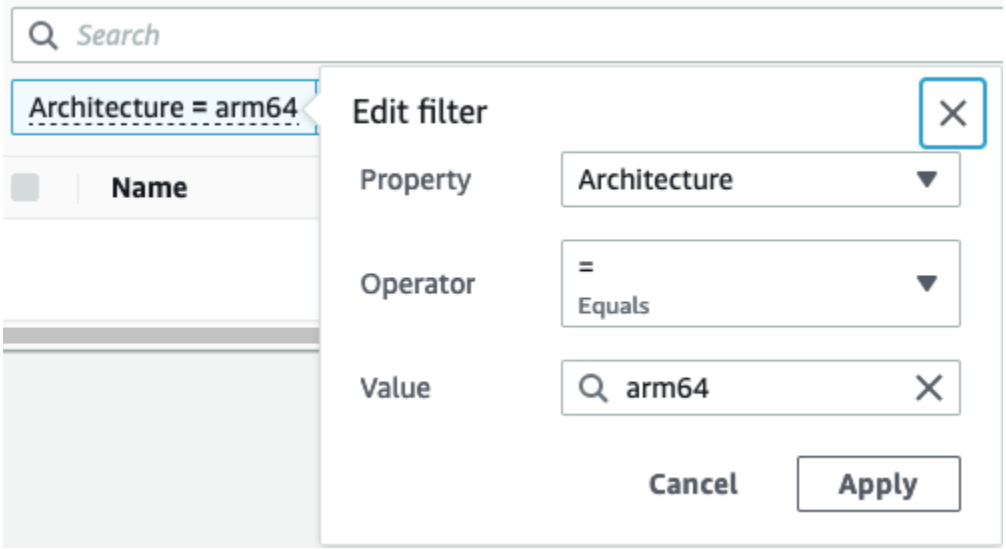


3. 탐색 창에서 리소스 유형에 해당하는 옵션을 선택합니다. 예를 들어, 모든 인스턴스를 나열하려면 인스턴스를 선택합니다.

콘솔을 사용하여 리소스 필터링

리소스 목록을 필터링하려면

1. 탐색 창에서 리소스 유형(예: [인스턴스])을 선택합니다.
2. 검색 필드를 선택합니다.
3. 목록에서 필터를 선택합니다.
4. 연산자를 선택합니다(예: =(같음)(= (Equals)). 일부 속성에는 선택할 수 있는 연산자가 더 많습니다. 모든 화면에서 운영자 선택을 지원하는 것은 아닙니다.
5. 필터 값을 선택합니다.
6. 선택한 필터를 편집하려면 필터 토큰(파란색 상자)을 선택하고 필요한 내용을 편집한 다음 적용(Apply)을 선택합니다. 모든 화면에서 선택한 필터 편집을 지원하는 것은 아닙니다.



7. 작업을 마쳤으면 필터를 제거합니다.

지원되는 필터

Amazon EC2 콘솔은 두 가지 유형의 필터링을 지원합니다.

- API 필터링은 서버 측에서 발생합니다. 필터링은 API 호출에 적용되어 서버에서 반환되는 리소스 수를 줄일 수 있습니다. 대규모 리소스 집합을 빠르게 필터링할 수 있으며 서버와 브라우저 간의 데이터 전송 시간과 비용을 줄일 수 있습니다. API 필터링은 =(같음) 및 :(포함) 연산자를 지원하며 항상 대소문자를 구분합니다.
- 클라이언트 필터링은 클라이언트 측에서 발생합니다. 브라우저에서 이미 사용 가능한 데이터(즉, API에서 이미 반환된 데이터)를 필터링할 수 있습니다. 클라이언트 필터링은 API 필터와 함께 작동하여 브라우저에서 더 작은 데이터 세트로 필터링합니다. 또한 =(같음) 및 :(포함) 연산자, 클라이언트 필터링은 >=(크거나 같음) 및 부정(역) 연산자(예:!=(같지 않음))와 같은 범위 연산자를 지원할 수도 있습니다.

Amazon EC2 콘솔에서는 다음과 같은 유형의 검색을 지원합니다.

키워드로 검색

키워드로 검색은 검색할 속성이나 태그 키를 지정하지 않고도 리소스의 모든 속성이나 태그에서 값을 검색할 수 있는 자유 텍스트 검색입니다.

> **Note**
>
> 모든 키워드 검색은 클라이언트 필터링을 사용합니다.

키워드로 검색하려면 검색하려는 키워드를 검색 필드에 입력 또는 붙여넣기한 다음 Enter를 누릅니다. 예를 들어 123을 검색하면 IP 주소, 인스턴스 ID, VPC ID 또는 AMI ID 또는 이름과 같은 태그에 123이 있는 모든 인스턴스가 일치합니다. 자유 텍스트 검색에서 예기치 않은 일치 항목이 반환되는 경우 추가 필터를 적용하세요.

속성으로 검색

속성으로 검색하면 모든 리소스에서 특정 속성을 검색할 수 있습니다.

> **Note**
>
> 속성 검색은 선택한 속성에 따라 API 필터링 또는 클라이언트 필터링을 사용합니다. 속성 검색을 수행할 때 그에 따라 속성이 그룹화됩니다.

예를 들어, 모든 인스턴스에 대한 인스턴스 상태 속성을 검색하여 `stopped` 상태에 있는 인스턴스만 반환할 수 있습니다. 방법:

1. 인스턴스 화면의 검색 필드에 `Instance state`를 입력합니다. 문자를 입력하면 인스턴스 상태에 대해 API 필터 및 클라이언트 필터의 두 가지 필터 유형이 표시됩니다.
2. 서버 측에서 검색하려면 API 필터 아래 인스턴스 상태를 선택합니다. 클라이언트 측에서 검색하려면 클라이언트 필터 아래 인스턴스 상태(클라이언트)를 선택합니다.

   선택한 속성에 사용할 수 있는 연산자 목록이 나타납니다.
3. =(같음) 연산자를 선택합니다.

   선택한 속성 및 연산자에 사용할 수 있는 값 목록이 나타납니다.
4. 목록에서 중지됨(Stopped)을 선택합니다.

태그로 검색

태그로 검색하면 현재 표시된 테이블의 리소스를 태그 키 또는 태그 값으로 필터링할 수 있습니다.

태그 검색은 기본 설정 창의 설정에 따라 API 필터링 또는 클라이언트 필터링 중 하나를 사용합니다.

태그에 대한 API 필터링 보장

1. 기본 설정(Preferences) 창을 엽니다.
2. 정규식 일치 사용 확인란을 선택 취소합니다. 이 확인란을 선택하면 클라이언트 필터링이 수행됩니다.
3. 대소문자 구분 일치 사용 확인란을 선택합니다. 이 확인란을 선택 취소하면 클라이언트 필터링이 수행됩니다.
4. 확인을 선택합니다.

태그별로 검색할 때 다음 값을 사용할 수 있습니다.

- (비어 있음)((empty)) - 지정된 태그 키는 있지만 태그 값은 없는 모든 리소스를 찾습니다.
- 모든 값(All values) - 지정된 태그 키와 임의의 태그 값을 가진 모든 리소스를 찾습니다.
- 태그 없음(Not tagged) - 지정된 태그 키가 없는 모든 리소스를 찾습니다.
- 표시된 값 - 지정된 태그 키와 지정된 태그 값을 가진 모든 리소스를 찾습니다.

다음 기법을 사용하여 검색을 향상시키거나 구체화할 수 있습니다.

역검색

역검색을 통해, 지정된 값과 일치하지 않는 리소스를 검색할 수 있습니다. 인스턴스(Instances) 및 AMI 화면에서, 역 검색은 !=(같지 않음) 또는 !:(포함하지 않음) 연산자를 선택한 다음 값을 선택하여 수행됩니다. 다른 화면에서 역검색은 검색 키워드 앞에 느낌표(!)를 붙여 수행됩니다.

> **Note**
>
> 역검색은 클라이언트 필터의 키워드 검색 및 속성 검색에서만 지원됩니다. API 필터에 대한 속성 검색에서는 지원되지 않습니다.

예를 들어, 모든 인스턴스의 인스턴스 상태 속성을 검색하여 `terminated` 상태의 모든 인스턴스를 제외할 수 있습니다. 방법:

1. 인스턴스 화면의 검색 필드에 `Instance state`를 입력합니다. 문자를 입력하면 인스턴스 상태에 대해 API 필터 및 클라이언트 필터의 두 가지 필터 유형이 표시됩니다.
2. 클라이언트 필터(Client filters) 아래 인스턴스 상태(클라이언트)(Instance state (client))를 선택합니다. 역검색은 클라이언트 필터에서만 지원됩니다.

선택한 속성에 사용할 수 있는 연산자 목록이 나타납니다.

3. !=(같지 않음)을 선택한 다음 종료(terminated)를 선택합니다.

인스턴스 상태 속성을 기준으로 인스턴스를 필터링하려면 인스턴스 상태 열의 검색 아이콘 ( ) 을 사용할 수도 있습니다. 더하기 기호(+)의 검색 아이콘은 해당 속성에 부합하는 모든 인스턴스를 표시합니다. 빼기 기호(-)의 검색 아이콘은 해당 속성에 부합하는 모든 인스턴스를 제외합니다.

다음은 역검색을 사용한 다른 예제입니다. `launch-wizard-1`라는 보안 그룹이 할당되지 않은 모든 인스턴스를 나열하려면 클라이언트 필터(Client filters)에서 보안 그룹 이름(Security group name) 속성으로 검색하여 !=을 선택한 다음 검색 창에서 `launch-wizard-1`을 입력합니다.

부분 검색

부분 검색을 사용하면 부분 문자열 값을 검색할 수 있습니다. 부분 검색을 수행하려면 검색할 키워드의 일부만 입력합니다. 인스턴스(Instances) 및 AMI 화면에서 부분 검색은 오직 :(포함) 연산자를 사용하여 수행할 수 있습니다. 다른 화면에서는 클라이언트 필터 속성을 선택하고 검색할 키워드의 일부만 즉시 입력할 수 있습니다. 예를 들어, 인스턴스 유형(Instance type) 화면에서 모든 `t2.micro`, `t2.small` 및 `t2.medium` 인스턴스를 검색하려면 인스턴스 유형(Instance Type) 속성으로 검색하고 키워드로 `t2`를 입력합니다.

정규식 검색

정규식 검색을 사용하려면 기본 설정 창에서 정규식 일치 사용을 선택해야 합니다.

필드의 값이 특정 패턴에 맞아야 하는 경우 정규식을 유용하게 활용할 수 있습니다. 예를 들어, `s`로 시작하는 값을 검색하려면 `^s`를 검색합니다. `xyz`로 끝나는 값을 검색하려면 `xyz$`를 검색합니다. 또는 뒤에 하나 이상의 문자가 오는 숫자로 시작하는 값을 검색하려면 `[0-9]+.*`를 검색합니다.

> **Note**
>
> 정규식 검색은 클라이언트 필터에 대한 키워드 검색 및 속성 검색에서만 지원됩니다. API 필터에 대한 속성 검색에서는 지원되지 않습니다.

대소문자 구분 검색

대/소문자 구분 검색을 사용하려면 기본 설정 창에서 대소문자 구분 일치 사용 확인란을 선택해야 합니다. 대소문자를 구분하는 기본 설정은 클라이언트 및 태그 필터에만 적용됩니다.

> **Note**
>
> API 필터는 항상 대소문자를 구분합니다.

와일드카드 검색

0개 이상의 문자와 일치시키려면 * 와일드카드를 사용합니다. 0개 또는 1개 이상의 문자와 일치시키려면 ? 와일드카드를 사용합니다. 예를 들어, prod, prods 및 production 값이 있는 데이터 세트가 있는 경우 prod* 검색은 모든 값과 일치하지만 prod?는 prod와 prods와만 일치합니다. 리터럴 값을 사용하려면 백슬래시(\)로 이스케이프합니다. 예를 들어, "prod*"는 prod*와 일치합니다.

> **Note**
>
> 와일드카드 검색은 API 필터에 대한 속성 및 태그 검색에서만 지원됩니다. 클라이언트 필터에 대한 키워드 검색 및 속성 및 태그 검색에서는 지원되지 않습니다.

검색 결합

일반적으로 동일한 속성을 가진 여러 필터는 OR을 사용하여 자동으로 조인됩니다. 예를 들어, Instance State : Running 및 Instance State : Stopped를 검색하면 실행 중이거나 중지된 모든 인스턴스가 반환됩니다. AND를 사용하여 검색을 조인하려면 여러 다른 속성을 검색합니다. 예를 들어 Instance State : Running 및 Instance Type : c4.large를 검색하면 c4.large 유형이면서 실행 중인 상태의 인스턴스만 반환됩니다.

콘솔을 사용하여 필터 세트 저장

저장된 필터 세트는 Amazon EC2 리소스를 효율적으로 보기 위해 생성하고 재사용할 수 있는 사용자 지정 필터 그룹입니다. 이 기능은 워크플로를 간소화하여 특정 리소스 보기에 빠르게 액세스할 수 있도록 합니다.

저장된 필터 세트는 Amazon EC2 콘솔에서만 지원되며, 현재 볼륨 페이지에서만 사용할 수 있습니다.

주요 기능

- 사용자 지정: 필요에 맞게 조정된 필터 세트를 생성합니다. 예를 들어 지정된 날짜 이후에 생성된 gp3 볼륨만 표시하도록 필터 세트를 생성할 수 있습니다.

- 기본 필터: 페이지의 기본 필터 세트를 설정하면 페이지로 이동할 때 기본 필터가 자동으로 적용됩니다. 기본값이 설정되지 않으면 필터가 적용되지 않습니다.
- 간편한 적용: 저장된 필터 세트를 선택하여 즉시 적용합니다. 그러면 Amazon EC2에 관련 리소스가 표시되고 활성 필터는 파란색 토큰으로 표시됩니다.
- 유연성: 필요에 따라 필터 세트를 생성, 수정, 삭제합니다.

필터 세트 생성

새 필터 세트를 생성하려면 다음과 같이 하세요.

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

> **Note**
> 저장된 필터 세트는 현재 볼륨에만 사용할 수 있습니다.

3. 검색 필드에서 필터 세트의 필터를 선택합니다.
4. 필터 지우기 버튼 옆의 화살표를 선택하고 새 필터 세트 저장을 선택합니다.
5. 필터 세트 저장 창에서 다음을 수행합니다.
   a. 필터 세트 이름에서 필터 세트 이름을 입력합니다.
   b. (선택 사항) 필터 세트 설명 필드에서 필터 세트 설명을 입력합니다.
   c. (선택 사항) 필터 세트를 기본 필터로 설정하려면 기본값으로 설정 확인란을 선택합니다.

> **Note**
> 콘솔 페이지를 열 때마다 기본 필터가 자동으로 적용됩니다.

   d. 저장을 선택합니다.

필터 세트 수정

필터 세트를 수정하려면 다음과 같이 하세요.

1. 저장된 필터 세트 목록에서 수정할 필터를 선택합니다.

2. 필터를 추가하려면 검색 필드에서 필터 세트에 추가할 필터를 선택합니다. 세트에서 필터를 삭제하려면 필터 토큰에서 X를 선택합니다.
3. 필터 지우기 버튼 옆의 화살표를 선택하고 필터 세트 수정을 선택합니다.
4. 필터 세트 수정 창에서 다음을 수행합니다.

   a. (선택 사항) 필터 세트를 기본 필터로 설정하려면 기본값으로 설정 확인란을 선택합니다.

   > **Note**
   >
   > 콘솔 페이지를 열 때마다 기본 필터가 자동으로 적용됩니다.

   b. 수정을 선택합니다.

필터 세트 삭제

필터 세트를 삭제하려면 다음과 같이 하세요.

1. 저장된 필터 세트 목록에서 삭제할 필터를 선택합니다.
2. 필터 지우기 버튼 옆의 화살표를 선택하고 필터 세트 삭제를 선택합니다.
3. 필터 세트 삭제 창에서 필터를 검토하여 삭제하려는 필터임을 확인한 다음 삭제를 선택합니다.

명령줄을 사용하여 나열 및 필터링

각 리소스 유형에는 본인이 해당 유형의 리소스를 설명하거나 나열하거나 가져오는 데 사용하는 해당 API 작업이 있습니다. 결과 리소스 목록은 길이가 길 수 있기 때문에 결과를 필터링하여 특정 기준에 부합하는 리소스만 포함시키는 것이 더 빠르고 유용할 수 있습니다.

필터링 고려 사항

- 단일 요청에서 최대 50개의 필터와 필터당 최대 200개의 값을 지정할 수 있습니다.
- 필터 문자열의 최대 길이는 255자입니다.
- 또한 필터 값과 함께 와일드카드를 사용할 수 있습니다. 별표(*)는 0개 이상의 문자에 해당하고 물음표(?)는 0개 또는 1개의 문자에 해당합니다.
- 필터 값은 대소문자를 구분합니다.
- 검색에는 와이드카드 문자의 리터럴 값이 포함될 수 있고 문자 앞에 백슬래시를 사용하면 벗어날 수 있습니다. 예를 들어, `\*amazon\?\\` 값은 리터럴 문자열 `*amazon?\`을 검색합니다.

- null의 필터 값은 지정할 수 없습니다. 대신 클라이언트 측 필터링을 사용합니다. 예를 들어 다음 명령은 --query 옵션을 사용하고 키 페어 없이 시작된 인스턴스의 ID를 반환합니다.

```
aws ec2 describe-instances \
    --query 'Reservations[*].Instances[?!not_null(KeyName)].InstanceId' \
    --output text
```

AWS CLI

Example 예: 단일 필터 지정

describe-instances를 사용하여 Amazon EC2 인스턴스를 나열할 수 있습니다. 필터가 없으면 응답에는 모든 리소스에 대한 정보가 포함됩니다. 다음 옵션을 사용하여 출력에 실행 중인 인스턴스만 포함할 수 있습니다.

```
--filters Name=instance-state-name,Values=running
```

실행 중인 인스턴스의 인스턴스 ID만 나열하려면 다음과 같이 `--query` 옵션을 추가합니다.

```
--query "Reservations[*].Instances[*].InstanceId"
```

Example 예: 여러 필터 또는 필터 값 지정

여러 필터 또는 여러 필터 값을 지정하는 경우, 리소스는 출력에 포함할 모든 필터와 일치해야 합니다.

다음 옵션을 사용하여 유형이 `m5.large` 또는 `m5d.large`인 모든 인스턴스를 나열할 수 있습니다.

```
--filters Name=instance-type,Values=m5.large,m5d.large
```

다음 옵션을 사용하여 유형이 `t2.micro`인 중지된 모든 인스턴스를 나열할 수 있습니다.

```
--filters Name=instance-state-name,Values=stopped Name=instance-type,Values=t2.micro
```

Example 예: 필터 값에 와일드카드 사용

describe-snapshots와 함께 다음 옵션을 사용하여 설명이 'database'인 스냅샷만 반환할 수 있습니다.

```
--filters Name=description,Values=database
```

* 와일드카드는 0개 이상의 문자와 일치합니다. 다음 옵션을 사용하여 설명에 database 단어가 포함된 스냅샷만 반환할 수 있습니다.

```
--filters Name=description,Values=*database*
```

? 와일드카드는 정확히 1개 문자와 일치합니다. 다음 옵션을 사용하여 설명이 문자 하나가 뒤에 있는 'database' 또는 'database'인 스냅샷만 반환할 수 있습니다.

```
--filters Name=description,Values=database?
```

다음 옵션을 사용하여 설명이 문자 4개까지 뒤에 있는 'database'인 스냅샷만 반환할 수 있습니다. 'database' 뒤에 5개 이상의 문자가 있는 설명은 제외됩니다.

```
--filters Name=description,Values=database????
```

Example 예: 날짜를 기준으로 필터링

AWS CLI를 사용하면 JMESPath를 통해 표현식을 사용하여 결과를 필터링할 수 있습니다. 예를 들어, 다음 describe-snapshots 명령에서는 지정된 날짜 이전에 지정된 AWS 계정에서 생성된 모든 스냅샷의 ID가 표시됩니다. 소유자를 지정하지 않으면 모든 퍼블릭 스냅샷이 결과에 포함됩니다.

```
aws ec2 describe-snapshots \
    --filters Name=owner-id,Values=123456789012 \
    --query "Snapshots[?(StartTime<='2024-03-31')].[SnapshotId]" \
    --output text
```

다음 예제에서는 지정된 날짜 범위에 생성된 모든 스냅샷의 ID가 표시됩니다.

```
aws ec2 describe-snapshots \
    --filters Name=owner-id,Values=123456789012 \
    --query "Snapshots[?(StartTime>='2024-01-01') && (StartTime<='2024-12-31')].
[SnapshotId]" \
    --output text
```

예: 태그를 기준으로 필터링

태그에 따라 리소스 목록을 필터링하는 방법에 대한 예는 태그로 Amazon EC2 리소스 필터링 단원을 참조하십시오.

PowerShell

Example 예: 단일 필터 지정

Get-EC2Instance를 사용하여 Amazon EC2 인스턴스를 나열할 수 있습니다. 필터가 없으면 응답에는 모든 리소스에 대한 정보가 포함됩니다. 다음과 같은 파라미터를 사용하여 출력에 실행 중인 인스턴스만 포함할 수 있습니다.

```
-Filter @{Name="instance-state-name"; Values="running"}
```

다음 예제에서는 실행 중인 인스턴스의 인스턴스 ID만 나열됩니다.

```
(Get-EC2Instance -Filter @{Name="instance-state-name"; Values="stopped"}).Instances
 | Select InstanceId
```

Example 예: 여러 필터 또는 필터 값 지정

여러 필터 또는 여러 필터 값을 지정하는 경우, 리소스는 결과에 포함할 모든 필터와 일치해야 합니다.

다음 옵션을 사용하여 유형이 `m5.large` 또는 `m5d.large`인 모든 인스턴스를 나열할 수 있습니다.

```
-Filter @{Name="instance-type"; Values="m5.large", "m5d.large"}
```

다음 옵션을 사용하여 유형이 `t2.micro`인 중지된 모든 인스턴스를 나열할 수 있습니다.

```
-Filter @{Name="instance-state-name"; Values="stopped"}, @{Name="instance-type";
 Values="t2.micro"}
```

Amazon EC2 Global View를 사용하여 리전 간 리소스 보기

Amazon EC2 Global View를 사용하면 Amazon EC2 및 Amazon VPC 리소스를 단일 AWS 리전에서, 또는 단일 콘솔에서 동시에 여러 리전에서 보고 검색할 수 있습니다. 자세한 내용은 Amazon EC2 Global View를 사용하여 리전 간 리소스 보기 섹션을 참조하세요.

Amazon EC2 Global View를 사용하여 리전 간 리소스 보기

Amazon EC2 Global View를 사용하면 일부 Amazon EC2 및 Amazon VPC 리소스를 단일 콘솔에서 단일 AWS 리전에 사용하거나 여러 리전에 걸쳐 사용할 수 있습니다. 또한 Amazon EC2 Global View는 여러 리전에서 특정 리소스 또는 특정 리소스 유형을 동시에 검색할 수 있는 전역 검색 기능을 제공합니다.

Amazon EC2 Global View에서는 어떤 식으로든 리소스를 수정할 수 없습니다.

지원되는 리소스

Amazon EC2 글로벌 뷰를 사용하면 AWS 계정이 활성화된 모든 리전에서 다음 리소스에 대한 글로벌 요약을 볼 수 있습니다.

- Auto Scaling 그룹
- 용량 예약 및 용량 블록
- DHCP 옵션 세트
- 외부 전용 인터넷 게이트웨이
- 탄력적 IP
- 엔드포인트 서비스
- 인스턴스
- 인터넷 게이트웨이
- 관리형 접두사 목록
- NAT 게이트웨이
- 네트워크 ACL
- 네트워크 인터페이스
- 라우팅 테이블
- 보안 그룹
- 서브넷
- 볼륨
- VPC
- VPC 엔드포인트
- VPC 피어링 연결

필수 권한

사용자에게 다음과 같은 Amazon EC2 Global View 사용 권한이 있어야 합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
  {
    "Effect": "Allow",
    "Action": [
    "autoscaling:DescribeAutoScalingGroups",
    "ec2:DescribeRegions",
    "ec2:DescribeCapacityReservations",
    "ec2:DescribeDhcpOptions",
    "ec2:DescribeEgressOnlyInternetGateways",
    "ec2:DescribeAddresses",
    "ec2:DescribeVpcEndpointServices",
    "ec2:DescribeInstances",
    "ec2:DescribeInternetGateways",
    "ec2:DescribePrefixLists",
    "ec2:DescribeNatGateways",
    "ec2:DescribeNetworkAcls",
    "ec2:DescribeNetworkInterfaces",
    "ec2:DescribeRouteTables",
    "ec2:DescribeSecurityGroups",
    "ec2:DescribeSubnets",
    "ec2:DescribeVolumes",
    "ec2:DescribeVpcs",
    "ec2:DescribeVpcEndpoints",
    "ec2:DescribeVpcPeeringConnections"
  ],
  "Resource": "*"
  }]
}
```

Amazon EC2 Global View를 사용하려면

https://console.aws.amazon.com/ec2globalview/home에서 Amazon EC2 Global View 콘솔을 엽니다.

> **Important**
>
> Firefox의 프라이빗 창을 사용하여 Amazon EC2 Global View에 액세스할 수 없습니다.

콘솔은 다음 구성 요소로 이루어져 있습니다.

- 리전 탐색기(Region explorer) - 이 탭에는 다음 섹션이 포함되어 있습니다.
  - 요약 - 모든 리전의 리소스에 대한 전반적인 개요를 제공합니다.

    활성화된 리전은 AWS 계정이 사용된 리전의 수를 나타냅니다. 나머지 필드는 현재 해당 리전에 있는 리소스의 수를 나타냅니다. 모든 리전에서 해당 유형의 리소스를 보려면 링크 중 하나를 선택합니다. 예를 들어 인스턴스(Instances) 레이블 아래의 링크가 10개 리전의 29개(29 in 10 Regions)인 경우 현재 10개 리전에 29개의 인스턴스가 있음을 나타냅니다. 29개 인스턴스의 전체 목록을 보려면 링크를 선택합니다.
  - 리소스 리전 수 - 모든 AWS 리전(계정이 사용되지 않은 리전 포함)을 나열하고 각 리전의 각 리소스 유형에 대한 합계를 제공합니다.

    해당 리전에 있는 모든 유형의 모든 리소스를 보려면 리전 이름을 선택합니다. 예를 들어 해당 리전의 모든 VPC, 서브넷, 인스턴스, 보안 그룹, 볼륨 및 Auto Scaling 그룹을 보려면 아프리카(케이프타운) af-south-1(Africa(Cape Town) af-south-1)을 선택합니다. 또는 리전을 선택하고 선택한 리전의 리소스 보기(View resources for selected Region)를 선택합니다.

    특정 리전에 있는 특정 유형의 리소스만 보려면 해당 리전의 해당 리소스 유형에 대한 값을 선택합니다. 예를 들어 해당 리전의 인스턴스만 보려면 아프리카(케이프타운) af-south-1(Africa (Cape Town) af-south-1)의 인스턴스에 대한 값을 선택합니다.
- 전역 검색(Global search) - 이 탭에서는 단일 리전 또는 여러 리전의 특정 리소스 또는 특정 리소스 유형을 검색할 수 있습니다. 또한 특정 리소스에 대한 세부 정보를 볼 수 있습니다.

  리소스를 검색하려면 그리드 앞의 필드에 검색 기준을 입력합니다. 리전, 리소스 유형 및 리소스에 할당된 태그를 기준으로 검색할 수 있습니다.

  특정 리소스에 대한 세부 정보를 보려면 그리드에서 해당 리소스를 선택합니다. 리소스의 리소스 ID를 선택하여 해당 콘솔에서 리소스를 열 수도 있습니다. 예를 들어 인스턴스 ID를 선택하여 Amazon EC2 콘솔에서 인스턴스를 열거나, 서브넷 ID를 선택하여 Amazon VPC 콘솔에서 서브넷을 엽니다.

> **Tip**
>
> 특정 리전 또는 리소스 유형만 사용하는 경우 해당 리전 및 리소스 유형만 표시하도록 Amazon EC2 Global View를 사용자 지정할 수 있습니다. 표시된 리전 및 리소스 유형을 사용자 지정하려면 탐색 패널에서 설정을 선택한 다음 리소스 및 리전 탭에서 Amazon EC2 Global View에 표시하지 않으려는 리전 및 리소스 유형을 선택합니다.

Amazon EC2 리소스 태깅

고유 메타데이터를 태그의 형태로 각 리소스에 배정하면 인스턴스, 이미지 및 기타 Amazon EC2 리소스를 쉽게 관리할 수 있습니다. 태그를 사용하면 용도, 소유자 또는 환경을 기준으로 하는 등 AWS 리소스를 다양한 방식으로 분류할 수 있습니다. 이 기능은 동일 유형의 리소스가 많을 때 유용합니다. 지정한 태그에 따라 특정 리소스를 빠르게 식별할 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

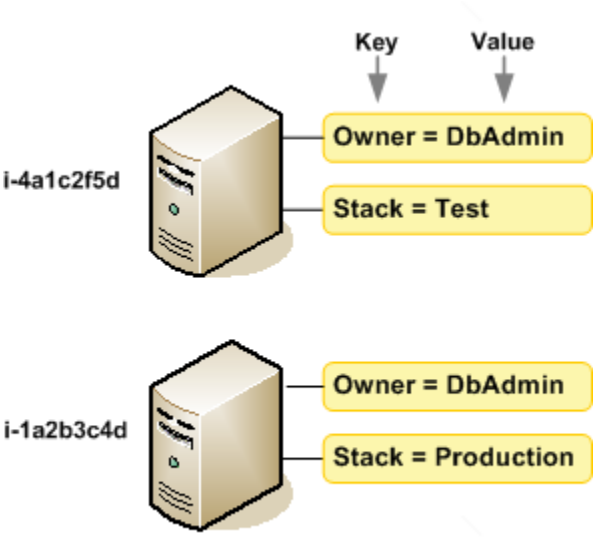
- 태그 기본 사항
- 리소스에 태그 지정
- 태그 제한
- 태그 및 액세스 관리
- 결제를 위한 리소스 태깅
- 생성 시 Amazon EC2 리소스 태그 지정에 대한 권한 부여
- Amazon EC2 리소스에 대한 태그 추가 및 제거
- 태그로 Amazon EC2 리소스 필터링
- 인스턴스 메타데이터를 사용하여 EC2 인스턴스의 태그 보기

태그 기본 사항

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

태그를 사용하면 용도, 소유자 또는 환경을 기준으로 하는 등 AWS 리소스를 다양한 방식으로 분류할 수 있습니다. 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 Owner라는 키가 있는 태그 하나와 Stack이라는 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 태깅 전략을 구현하는 방법에 관한 자세한 내용은 태깅 모범 사례 AWS 백서를 참조하세요.

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.

> **Note**
>
> 리소스를 삭제한 후에도 해당 태그가 콘솔, API 및 CLI 출력에 잠시 동안 표시될 수 있습니다. 이러한 태그는 리소스에서 서서히 연결이 해제되고 영구적으로 삭제됩니다.

리소스에 태그 지정

Amazon EC2 콘솔을 사용하는 경우, 관련 리소스 화면에서 태그 탭을 사용하여 리소스에 태그를 적용하거나 AWS Resource Groups 콘솔에서 태그 편집기를 사용할 수 있습니다. 일부 리소스 화면을 사용

하면 리소스를 생성할 때 리소스에 대해 태그를 지정할 수 있습니다. 예를 들어 Name의 키가 있는 태그와 지정하는 값이 있습니다. 대부분의 경우, 콘솔은 리소스 생성 직후(리소스 생성 중이 아니라) 태그를 적용합니다. 콘솔은 Name 태그에 따라 리소스를 조직할 수도 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. 사용자가 생성 시 리소스 태그를 지정할 수 있도록 하는 방법에 대한 자세한 내용은 생성 시 Amazon EC2 리소스 태그 지정에 대한 권한 부여 섹션을 참조하세요.

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 예: 태그 리소스 섹션을 참조하세요.

결제를 위한 리소스 태깅에 대한 자세한 내용은 AWS Billing 사용 설명서에서 비용 할당 태그 사용을 참조하세요.

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-8 형식의 유니코드 문자 128자
- 최대 값 길이 - UTF-8 형식의 유니코드 문자 256자
- 허용되는 문자

  - EC2는 태그에 모든 문자를 사용할 수 있지만, 다른 AWS 서비스에는 제한이 적용되기도 합니다. 모든 AWS 서비스에서 허용되는 문자는 UTF-8로 표현할 수 있는 문자(a-z, A-Z), 숫자(0-9) 및 공백과 특수 문자(+ - = . _ : / @)입니다.
  - 인스턴스 메타데이터에서 인스턴스 태그를 활성화하면 인스턴스 태그 키는 문자(a-z, A-Z), 숫자(0-9) 및 + - = . , _ : @ 문자만 사용할 수 있습니다. 인스턴스 태그 키에는 공백 또는 /가 포함될 수 없으며, .(마침표 1개), ..(마침표 2개) 또는 _index만으로 구성될 수 없습니다. 자세한 내용은 인스턴스 메타데이터를 사용하여 EC2 인스턴스의 태그 보기 섹션을 참조하세요.
- 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 DeleteMe라는 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 경우 할당하는 태그는 사용자의 AWS 계정에만 사용할 수 있으며 다른 AWS 계정은 해당 태그에 액세스할 수 없습니다. 공유 리소스에 대한 태그 기반 액세스 제어의 경우 각 AWS 계정은 리소스에 대한 액세스를 제어하기 위해 자체 태그 세트를 할당해야 합니다.

태그 및 액세스 관리

AWS Identity and Access Management(IAM)를 사용하는 경우 AWS 계정에서 태그를 생성, 편집 또는 삭제할 수 있는 권한이 있는 사용자를 제어할 수 있습니다. 자세한 내용은 생성 시 Amazon EC2 리소스 태그 지정에 대한 권한 부여 섹션을 참조하세요.

리소스 태그를 사용하여 속성 기반 제어(ABAC)를 구현할 수도 있습니다. 리소스에 대한 태그를 기반으로 작업을 허용하는 IAM 정책을 생성할 수 있습니다. 자세한 내용은 속성 기반 액세스를 사용하여 액세스 제어 섹션을 참조하세요.

결제를 위한 리소스 태깅

또한 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 사용 설명서에서 월간 비용 할당 보고서를 참조하세요. 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 사용 설명서에서 비용 할당 태그 사용을 참조하세요.

> **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 사용 설명서에서 Amazon Elastic Block Store 볼륨 및 스냅샷을 참조하세요.

> **Note**
>
> 태그된 탄력적 IP 주소는 비용 할당 보고서에 나타나지 않습니다.

생성 시 Amazon EC2 리소스 태그 지정에 대한 권한 부여

일부 리소스 생성 Amazon EC2 API 작업에서는 리소스를 생성할 때 태그를 지정할 수 있습니다. 리소스 태그를 사용하여 속성 기반 제어(ABAC)를 구현할 수 있습니다. 자세한 내용은 리소스에 태그 지정 및 속성 기반 액세스를 사용하여 액세스 제어 섹션을 참조하세요.

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 `Condition` 조건 키와 함께 `ec2:CreateAction` 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. 정책 예제는 시작 템플릿의 태그을 참조하세요.

특정 태그에 대한 액세스 제어

IAM 정책의 Condition 요소에 추가 조건을 사용하여 리소스에 적용할 수 있는 태그 키와 값을 제어할 수 있습니다.

앞 섹션의 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 Amazon EC2에 사용되는 작업, 리소스 및 조건 키를 참조하세요.

사용자가 리소스를 생성할 때 강제로 태그를 지정하도록 하려면 리소스 생성 작업에서 `aws:RequestTag` 조건 키 또는 `aws:TagKeys` 조건 키를 `ForAnyValue` 변경자와 함께 사용해야 합니다. 이때 사용자가 리소스 생성 시 태그를 지정하지 않으면 `ec2:CreateTags` 작업이 평가되지 않습니다.

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 Amazon EC2 API에 대한 액세스를 제어하는 정책 예제 섹션을 참조하세요. 자세한 내용은 IAM 사용자 가이드에서 다수의 컨텍스트 키 또는 값을 사용하는 조건를 참조하세요.

Amazon EC2 리소스에 대한 태그 추가 및 제거

Amazon EC2 리소스(예: Amazon EC2 인스턴스)를 생성할 때 리소스에 추가할 태그를 지정할 수 있습니다. 또한 Amazon EC2 콘솔을 사용하여 특정 Amazon EC2 리소스에 대한 태그를 표시할 수 있습니다. 기존 Amazon EC2 리소스에 태그를 추가하거나 리소스에서 태그를 제거할 수도 있습니다.

AWS Resource Groups 콘솔의 태그 편집기를 사용하여 모든 리전에 있는 모든 AWS 리소스의 태그를 보거나 추가 또는 제거할 수 있습니다. 동시에 여러 유형의 리소스에서 태그를 적용하거나 제거할 수 있습니다. 자세한 내용을 알아보려면 AWS 리소스 태깅 사용 설명서를 참조하세요.

업무

- 콘솔을 사용하여 태그 추가
- AWS CLI를 사용하여 태그 추가
- PowerShell을 사용하여 태그 추가
- CloudFormation을 사용하여 태그 추가

콘솔을 사용하여 태그 추가

리소스에 대한 페이지에서 바로 기존 리소스에 태그를 추가할 수 있습니다.

기존 리소스에 태그 추가

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 모음에서 리소스가 있는 리전을 선택합니다.
3. 탐색 창에서 리소스 유형(예: [인스턴스])을 선택합니다.
4. 목록에서 리소스를 선택합니다.
5. 태그 탭에서 태그 관리를 선택합니다.
6. 새 태그 추가를 선택하고 태그의 키와 값을 입력합니다.
7. 저장을 선택합니다.

AWS CLI를 사용하여 태그 추가

리소스를 생성할 때 또는 기존 리소스에 태그를 추가할 수 있습니다.

리소스 생성 시 태그를 추가하는 방법

생성 시 `-tag-specifications` 옵션을 사용하여 리소스에 태그를 지정합니다. 태그 사양에는 태그를 지정할 리소스 유형, 태그 키, 태그 값이 필요합니다. 다음 예제에서는 태그를 생성하고 태그 사양에 추가합니다.

```
--tag-specifications 'ResourceType=instance,Tags=[{Key=stack,Value=production}]'
```

기존 리소스에 태그를 추가하는 방법

다음 예제에서는 create-tags 명령을 사용하여 기존 리소스에 태그를 추가하는 방법을 보여줍니다.

Example 예제: 리소스에 태그 추가

이 예제에서는 지정된 이미지에 **`Stack=production`** 태그를 추가하거나 태그 키가 stack인 AMI의 기존 태그를 덮어씁니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

```
aws ec2 create-tags \
    --resources ami-0abcdef1234567890 \
    --tags Key=stack,Value=production
```

Example 예제: 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-0abcdef1234567890 i-1234567890abcdef0 \
    --tags Key=webserver,Value=  Key=stack,Value=Production
```

Example 예제: 특수 문자에 태그 추가

이 예제에서는 [Group]=test 태그를 인스턴스에 추가합니다. 대괄호([및])는 이스케이프해야 하는 특수 문자입니다.

Linux 또는 OS X를 사용하는 경우 특수 문자를 이스케이프하려면 특수 문자가 있는 요소를 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'
```

Windows를 사용하는 경우 특수 문자를 이스케이프하려면 다음과 같이 특수 문자가 있는 요소를 큰 따옴표(“)로 묶은 다음 각 큰 따옴표 문자 앞에 백슬래시(****)를 붙입니다.

```
aws ec2 create-tags ^
    --resources i-1234567890abcdef0 ^
    --tags Key=\"[Group]\",Value=test
```

Windows PowerShell을 사용하는 경우 특수 문자를 이스케이프하려면 다음과 같이 특수 문자가 있는 값을 큰 따옴표(**"**)로 묶고 각 큰 따옴표 문자 앞에 백슬래시(****)를 붙인 다음 전체 키 및 값 구조를 작은 따옴표(**'**)로 묶습니다.

```
aws ec2 create-tags `
    --resources i-1234567890abcdef0 `
    --tags 'Key=\"[Group]\",Value=test'
```

PowerShell을 사용하여 태그 추가

리소스를 생성할 때 또는 기존 리소스에 태그를 추가할 수 있습니다.

리소스 생성 시 태그를 추가하는 방법

생성 시 -TagSpecification 파라미터를 사용하여 리소스에 태그를 지정합니다. 태그 사양에는 태그를 지정할 리소스 유형, 태그 키, 태그 값이 필요합니다. 다음 예제에서는 태그를 생성하고 태그 사양에 추가합니다.

```
$tag = @{Key="stack"; Value="production"}
$tagspec = new-object Amazon.EC2.Model.TagSpecification
$tagspec.ResourceType = "instance"
$tagspec.Tags.Add($tag)
```

다음 예제에서는 -TagSpecification 파라미터에 이 태그를 지정합니다.

```
-TagSpecification $tagspec
```

기존 리소스에 태그를 추가하는 방법

New-EC2Tag cmdlet을 사용합니다. 리소스, 태그 키, 태그 값을 지정해야 합니다.

```
New-EC2Tag `
    -Resource i-1234567890abcdef0 `
    -Tag @{Key="purpose"; Value="production"}
```

CloudFormation을 사용하여 태그 추가

Amazon EC2 리소스 유형에서는 Tags 또는 TagSpecifications 속성을 사용하여 태그를 지정합니다.

다음 예에서는 Tags 속성을 사용하여 AWS::EC2::Instance에 **Stack=Production** 태그를 추가합니다.

Example 예: YAML 태그

```
Tags:
  - Key: "Stack"
    Value: "Production"
```

Example 예: JSON 태그

```
"Tags": [
```

```
    {
        "Key": "Stack",
        "Value": "Production"
    }
]
```

다음 예에서는 **Stack=Production** 속성을 사용하여 AWS::EC2::LaunchTemplate LaunchTemplateData에 TagSpecifications 태그를 추가합니다.

Example 예: YAML TagSpecifications

```
TagSpecifications:
  - ResourceType: "instance"
    Tags:
    - Key: "Stack"
      Value: "Production"
```

Example 예: JSON TagSpecifications

```
"TagSpecifications": [
    {
        "ResourceType": "instance",
        "Tags": [
            {
                "Key": "Stack",
                "Value": "Production"
            }
        ]
    }
]
```

태그로 Amazon EC2 리소스 필터링

태그를 추가한 후 태그 키와 태그 값을 기준으로 Amazon EC2 리소스를 필터링할 수 있습니다.

Console

태그로 리소스를 필터링하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 리소스 유형(예: [인스턴스])을 선택합니다.

3. 검색 필드를 선택합니다.
4. 목록의 태그에서 태그 키를 선택합니다.
5. 목록에서 해당 태그 값을 선택합니다.
6. 작업을 마쳤으면 필터를 제거합니다.

Amazon EC2 콘솔에서의 필터 사용에 관한 자세한 내용은 Amazon EC2 리소스 찾기 섹션을 참조하세요.

AWS CLI

지정된 태그 키가 있는 단일 유형의 리소스를 설명하는 방법

태그 값과 관계없이 다음 필터를 `describe` 명령에 추가하여 Stack 태그가 있는 해당 유형의 리소스를 설명합니다.

```
--filters Name=tag-key,Values=Stack
```

지정된 태그가 있는 단일 유형의 리소스를 설명하는 방법

다음 필터를 `describe` 명령에 추가하여 Stack=production 태그가 있는 해당 유형의 리소스를 설명합니다.

```
--filters Name=tag:Stack,Values=production
```

지정된 태그 값이 있는 단일 유형의 리소스를 설명하는 방법

태그 키와 관계없이 다음 필터를 `describe` 명령에 추가하여 값이 production인 태그가 있는 해당 유형의 리소스를 설명합니다.

```
--filters Name=tag-value,Values=production
```

지정된 태그가 있는 모든 EC2 리소스를 설명하는 방법

다음 필터를 describe-tags 명령에 추가하여 Stack=test 태그가 있는 모든 EC2 리소스를 설명합니다.

```
--filters Name=key,Values=Stack Name=value,Values=test
```

PowerShell

태그 키 기준으로 단일 유형의 리소스를 필터링하는 방법

태그 값과 관계없이 다음 필터를 Get cmdlet에 추가하여 Stack 태그가 있는 해당 유형의 리소스를 설명합니다.

```
-Filter @{Name="tag-key"; Values="Stack"}
```

태그 기준으로 단일 유형의 리소스를 필터링하는 방법

다음 필터를 Get cmdlet에 추가하여 Stack=production 태그가 있는 해당 유형의 리소스를 설명합니다.

```
-Filter @{Name="tag:Stack"; Values="production"}
```

태그 값을 기준으로 단일 유형의 리소스를 필터링하는 방법

태그 키 값과 관계없이 다음 필터를 Get cmdlet에 추가하여 값이 production인 태그가 있는 해당 유형의 리소스를 설명합니다.

```
-Filter @{Name="tag-value"; Values="production"}
```

태그로 모든 EC2 리소스를 필터링하는 방법

다음 필터를 Get-EC2Tag cmdlet에 추가하여 Stack=test 태그가 있는 모든 EC2 리소스를 설명합니다.

```
-Filter @{Name="tag:Stack"; Values="test"}
```

인스턴스 메타데이터를 사용하여 EC2 인스턴스의 태그 보기

인스턴스 메타데이터에서 인스턴스의 태그에 액세스할 수 있습니다. 인스턴스 메타데이터에서 태그에 액세스하면 더 이상 DescribeInstances 또는 DescribeTags API 호출을 사용하여 태그 정보를 검색할 필요가 없습니다. 그러면 초당 API 트랜잭션이 줄어들고 제어하는 인스턴스 수에 따라 태그 검색이 확장될 수 있습니다. 또한 인스턴스에서 실행 중인 로컬 프로세스는 인스턴스 메타데이터에서 직접 인스턴스의 태그 정보를 볼 수 있습니다.

기본적으로 태그는 인스턴스 메타데이터에서 사용할 수 없으므로 액세스를 명시적으로 허용해야 합니다. 인스턴스 시작 시 또는 실행 중이거나 중지된 인스턴스에서 시작 후 액세스를 허용할 수 있습니다. 시작 템플릿에서 이를 지정하여 태그에 대한 액세스를 허용할 수도 있습니다. 템플릿을 사용하여 시작된 인스턴스는 인스턴스 메타데이터의 태그에 대한 액세스를 허용합니다.

인스턴스 태그를 추가하거나 제거하면 인스턴스가 실행되는 동안 인스턴스를 중지했다가 시작할 필요 없이 인스턴스 메타데이터가 업데이트됩니다.

업무

- 인스턴스 메타데이터에서 태그에 대한 액세스 활성화
- 인스턴스 메타데이터에서 태그 검색
- 인스턴스 메타데이터에서 태그에 대한 액세스 비활성화

인스턴스 메타데이터에서 태그에 대한 액세스 활성화

기본적으로 인스턴스 메타데이터의 인스턴스 태그에 액세스할 수 없습니다. 각 인스턴스에 대해 명시적으로 액세스를 활성화해야 합니다.

> **Note**
>
> 인스턴스 메타데이터의 태그에 대한 액세스를 허용하는 경우 인스턴스 태그 키에 특정 제한이 적용됩니다. 규정을 준수하지 않으면 새 인스턴스의 시작이 실패하거나 기존 인스턴스에 대한 오류가 발생합니다. 제한 사항은 다음과 같습니다.
>
> - 문자(a-z, A-Z), 숫자(0-9), 다음 문자(+ - = . , _ : @)만 포함할 수 있습니다.
> - 공백 또는 /를 포함할 수 없습니다.
> - .(1기간), ..(2기간) 또는 _index로만 구성할 수 없습니다.
>
> 자세한 내용은 태그 제한 섹션을 참조하세요.

Console

인스턴스 시작 중 인스턴스 메타데이터의 태그에 대한 액세스를 활성화하는 방법

1. 인스턴스 시작 절차를 따릅니다.
2. 고급 세부 정보를 확장하고 메타데이터에서 태그 허용에 대해 활성화를 선택합니다.

3. Summary(요약) 패널에서 인스턴스 구성을 검토한 다음 Launch instance(인스턴스 시작)를 선택합니다. 자세한 내용은 콘솔에서 인스턴스 시작 마법사를 사용하여 EC2 인스턴스 시작 섹션을 참조하세요.

인스턴스 시작 후 인스턴스 메타데이터의 태그에 대한 액세스를 활성화하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택한 다음 작업, 인스턴스 설정, 인스턴스 메타데이터의 태그 허용을 선택합니다.
4. 인스턴스 메타데이터의 태그에 대한 액세스를 허용하려면 허용 확인란을 선택합니다.
5. 저장을 선택합니다.

AWS CLI

인스턴스 시작 중 인스턴스 메타데이터의 태그에 대한 액세스를 활성화하는 방법

run-instances 명령을 사용하고 다음 `--metadata-options` 옵션을 추가합니다.

```
--metadata-options "InstanceMetadataTags=enabled"
```

인스턴스 시작 후 인스턴스 메타데이터의 태그에 대한 액세스를 활성화하는 방법

다음 modify-instance-metadata-options 명령을 사용합니다.

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567890abcdef0 \
    --instance-metadata-tags enabled
```

인스턴스 메타데이터의 태그에 대한 액세스가 활성화되었는지 확인하는 방법

describe-instances 명령을 사용하고 `InstanceMetadataTags` 값을 확인합니다.

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query "Reservations[*].Instances[].MetadataOptions[].InstanceMetadataTags"
```

출력의 예시는 다음과 같습니다. 이때 값은 enabled 또는 disabled입니다.

```
[
    "enabled"
]
```

PowerShell

인스턴스 시작 중 인스턴스 메타데이터의 태그에 대한 액세스를 활성화하는 방법

New-EC2Instance cmdlet을 사용하고 다음 -MetadataOptions_InstanceMetadataTags 파라미터를 추가합니다.

```
-MetadataOptions_InstanceMetadataTags enabled
```

인스턴스 시작 후 인스턴스 메타데이터의 태그에 대한 액세스를 활성화하는 방법

Edit-EC2InstanceMetadataOption cmdlet을 사용합니다.

```
Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -InstanceMetadataTags enabled
```

인스턴스 메타데이터의 태그에 대한 액세스가 활성화되었는지 확인하는 방법

Get-EC2Instance cmdlet을 사용하고 InstanceMetadataTags 값을 확인합니다.

```
(Get-EC2Instance `
    -
InstanceId i-1234567890abcdef0).Instances.MetadataOptions.InstanceMetadataTags.Value
```

출력의 예시는 다음과 같습니다. 이때 값은 enabled 또는 disabled입니다.

```
enabled
```

인스턴스 메타데이터에서 태그 검색

인스턴스 메타데이터에서 인스턴스 태그 액세스를 허용한 이후 인스턴스 메타데이터에서 tags/instance 카테고리에 액세스할 수 있습니다. 자세한 내용은 EC2 인스턴스에 대한 인스턴스 메타데이터에 액세스 섹션을 참조하세요.

IMDSv2

Linux

Linux 인스턴스에서 다음 명령을 실행하여 인스턴스에 대한 모든 태그 키를 나열합니다.

```
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" http://169.254.169.254/latest/meta-data/tags/instance
```

이 예제에서는 이전 예제에서 얻은 키 값을 가져옵니다. IDMSv2 요청은 이전 예제의 명령을 사용하여 생성한 저장된 토큰을 사용합니다. 토큰이 만료되지 않아야 합니다.

```
curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/tags/instance/tag-key
```

Windows

Windows 인스턴스에서 다음 cmdlet을 실행하여 인스턴스에 대한 모든 태그 키를 나열합니다.

```
$token = Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/tags/instance
```

이 예제에서는 이전 예제에서 얻은 키 값을 가져옵니다. IDMSv2 요청은 이전 예제의 명령을 사용하여 생성한 저장된 토큰을 사용합니다. 토큰이 만료되지 않아야 합니다.

```
Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/tags/instance/tag-key
```

IMDSv1

Linux

Linux 인스턴스에서 다음 명령을 실행하여 인스턴스에 대한 모든 태그 키를 나열합니다.

```
curl http://169.254.169.254/latest/meta-data/tags/instance
```

이 예제에서는 이전 예제에서 얻은 키 값을 가져옵니다.

```
curl http://169.254.169.254/latest/meta-data/tags/instance/tag-key
```

Windows

Windows 인스턴스에서 다음 cmdlet을 실행하여 인스턴스에 대한 모든 태그 키를 나열합니다.

```
Invoke-RestMethod -Uri http://169.254.169.254/latest/meta-data/tags/instance
```

이 예제에서는 이전 예제에서 얻은 키 값을 가져옵니다.

```
Invoke-RestMethod -Uri http://169.254.169.254/latest/meta-data/tags/instance/tag-key
```

인스턴스 메타데이터에서 태그에 대한 액세스 비활성화

인스턴스 메타데이터에서 인스턴스 태그에 대한 액세스를 비활성화할 수 있습니다. 인스턴스 메타데이터의 인스턴스 태그에 대한 액세스는 기본적으로 해제되어 있으므로 시작 시 비활성화할 필요가 없습니다.

Console

인스턴스 메타데이터에서 태그에 대한 액세스를 비활성화하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택한 다음 작업(Actions), 인스턴스 설정(Instance settings), 인스턴스 메타데이터의 태그 허용(Allow tags in instance metadata)을 선택합니다.
4. 인스턴스 메타데이터의 태그에 대한 액세스를 해제하려면 허용 확인란 선택을 취소합니다.
5. 저장을 선택합니다.

AWS CLI

인스턴스 메타데이터에서 태그에 대한 액세스를 비활성화하는 방법

다음 modify-instance-metadata-options 명령을 사용합니다.

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567890abcdef0 \
    --instance-metadata-tags disabled
```

PowerShell

인스턴스 메타데이터에서 태그에 대한 액세스를 비활성화하는 방법

Edit-EC2InstanceMetadataOption cmdlet을 사용합니다.

```
Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -InstanceMetadataTag disabled
```

Amazon EC2 서비스 할당량

AWS 계정을 생성하면 AWS 리소스에 대한 기본 할당량(한도라고도 함)이 리전별로 설정됩니다. 리소스 할당량을 초과하려고 하면 요청이 실패합니다. 예를 들어 한 리전에서 온디맨드 인스턴스용으로 프로비저닝할 수 있는 최대 Amazon EC2 vCPU 수가 있습니다. 한 리전에서 인스턴스를 시작하려고 하는데 이 요청으로 인해 사용량이 할당량을 초과하면 요청은 실패합니다. 이 경우 리소스 사용량을 줄이거나 할당량 증가를 요청할 수 있습니다.

Service Quotas 콘솔은 AWS 서비스에 대한 할당량을 보고 관리할 수 있으며 사용하는 많은 리소스에 대한 할당량 증가를 요청할 수 있는 중앙 위치입니다. 제공되는 할당량 정보를 사용하여 AWS 인프라를 관리하세요. 실제로 필요할 시점보다 미리 할당량 증가를 요청하도록 계획하세요.

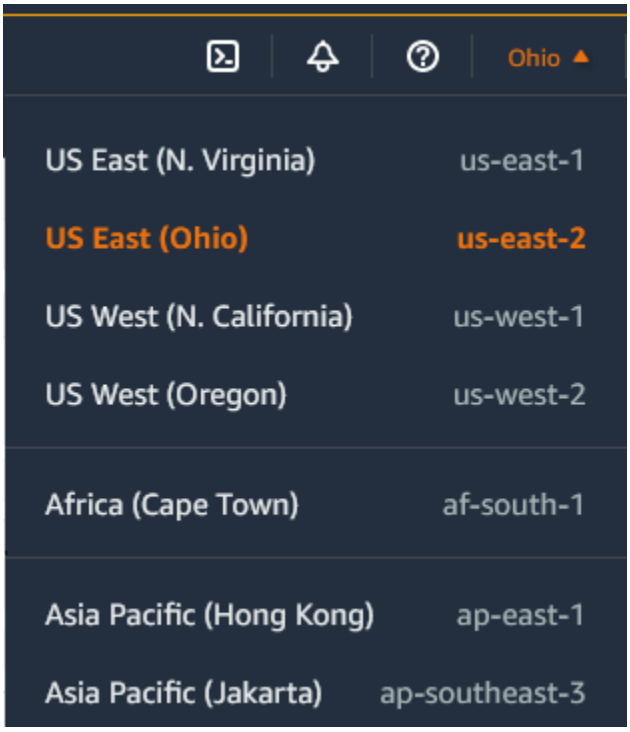
자세한 내용은 Amazon Web Services 일반 참조의 Amazon EC2 endpoints and quotas 및 Amazon EBS endpoints and quotas를 참조하세요.

현재 할당량 보기

Service Quotas 콘솔을 사용하여 각 리전에 대한 할당량을 볼 수 있습니다.

Service Quotas 콘솔을 사용하여 현재 할당량 보기

1. https://console.aws.amazon.com/servicequotas/home/services/ec2/quotas/에서 Service Quotas 콘솔을 엽니다.
2. 탐색 모음(화면 상단)에서 리전을 선택합니다.



3. 필터 필드를 사용하여 리소스 이름별로 목록을 필터링합니다. 예를 들어, 온디맨드 인스턴스의 할당량을 찾으려면 **On-Demand**을(를) 입력합니다.
4. 자세한 정보를 보려면 할당량 이름을 선택하여 할당량에 대한 세부 정보 페이지를 엽니다.

증가 요청

각 리전에 대해 할당량 증가 요청을 할 수 있습니다.

Service Quotas 콘솔을 사용하여 증가 요청

1. https://console.aws.amazon.com/servicequotas/home/services/ec2/quotas/에서 Service Quotas 콘솔을 엽니다.
2. 탐색 모음(화면 상단)에서 리전을 선택합니다.
3. 필터 필드를 사용하여 리소스 이름별로 목록을 필터링합니다. 예를 들어, 온디맨드 인스턴스의 할당량을 찾으려면 **On-Demand**을(를) 입력합니다.
4. 할당량을 조정할 수 있는 경우 할당량을 선택한 다음 할당량 증가 요청을 선택합니다.
5. 할당량 값 변경에 새 할당량 값을 입력합니다.
6. 요청을 선택합니다.
7. 콘솔에서 보류 중이거나 최근에 해결된 요청을 보려면 탐색 창에서 대시보드를 선택합니다. 보류 중인 요청의 경우 요청 상태를 선택하여 요청 접수증을 엽니다. 요청의 초기 상태는 Pending(보류

중)입니다. 상태가 할당량 요청됨으로 변경되면 지원의 케이스 번호가 표시됩니다. 이 케이스 번호를 선택하여 요청의 티켓을 엽니다.

AWS CLI 또는 SDK를 사용하여 할당량 증가를 요청하는 방법을 비롯한 자세한 내용은 Service Quotas 사용 설명서의 할당량 증가 요청을 참조하세요.

포트 25를 사용하여 전송되는 이메일 관련 제한

기본적으로, Amazon EC2에서는 프라이빗 IPv4 주소에 대한 포트 25 경유 아웃바운드 트래픽이 허용됩니다. 퍼블릭 IPv4 주소 및 IPv6 주소에 대한 포트 25 경유 트래픽은 차단됩니다.

이 제한을 제거하도록 요청할 수 있습니다. 자세한 내용은 에서 Amazon EC2 인스턴스 또는 Lamda 함수에서 포트 25와 관련한 제한을 없애려면 어떻게 해야 하나요?를 참조하세요.

대상이 다음과 같은 포트 25 경유 아웃바운드 트래픽에는 이 제한이 적용되지 않습니다.

- 원래 네트워크 인터페이스가 있는 VPC의 기본 CIDR 블록에 있는 IP 주소입니다.
- RFC 1918, RFC 6598 및 RFC 4193에 정의된 CIDR의 IP 주소입니다.

Amazon EC2 리소스 모니터링

모니터링은 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.

AWS는 Amazon EC2를 모니터링하는 데 사용할 수 있는 다양한 도구를 제공합니다. Amazon EC2 및 CloudWatch 콘솔 대시보드에서는 Amazon EC2 환경을 한 눈에 파악할 수 있습니다. 추가로 다음을 제공합니다.

- 시스템 상태 확인 - 인스턴스를 사용하는 데 필요한 AWS 시스템을 모니터링하여 올바르게 작동 중인지 확인합니다. 이러한 확인에서는 복구 시 AWS 개입이 필요한 인스턴스 관련 문제를 찾아냅니다. 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 Amazon EC2 인스턴스 상태 확인 섹션을 참조하세요.
- 인스턴스 상태 확인 – 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 Amazon EC2 인스턴스 상태 확인 섹션을 참조하세요.
- 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) 주제 또는 Amazon EC2 Auto Scaling 정책에 전송되는 알림입니

다. 경보는 지속적인 상태 변경에 대해서만 작업을 호출합니다. CloudWatch 경보는 특정 상태에 있다는 이유만으로는 작업을 호출하지 않습니다. 상태가 변경되고 지정한 기간 동안 유지되어야 합니다. 자세한 내용은 CloudWatch를 사용하여 인스턴스 모니터링 단원을 참조하십시오.

- Amazon EventBridge 이벤트 – AWS 서비스를 자동화하여 시스템 이벤트에 자동으로 응답합니다. AWS 서비스 이벤트는 거의 실시간으로 EventBridge에 전송되며, 전송된 이벤트가 사용자가 정의한 규칙과 일치할 경우 실행할 자동 작업을 지정할 수 있습니다. 자세한 내용은 the section called "EventBridge를 사용하여 자동화" 단원을 참조하십시오.
- AWS CloudTrail 로그 - Amazon EC2 API에 보낸 호출에 대한 자세한 정보를 캡처하고 Amazon S3에 로그 파일로 저장합니다. 이러한 CloudTrail 로그를 사용하여 어떤 요청이 이루어졌는지, 호출의 IP 주소, 누가 언제 호출을 요청했는지 등을 확인할 수 있습니다. 자세한 내용은 the section called "CloudTrail을 사용하여 API 호출 로깅" 단원을 참조하십시오.
- CloudWatch 에이전트 – EC2 인스턴스와 온프레미스 서버의 호스트 및 게스트 모두에서 로그와 시스템 수준 지표를 수집합니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 에이전트를 사용하여 Amazon EC2 인스턴스 및 온프레미스 서버로부터 지표 및 로그 수집을 참조하세요.

Amazon EC2 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 상태도 확인할 수 있습니다. 이벤트 상태는 재부팅이나 만료 등 인스턴스에 대해 설정된 예정 활동에 대한 정보를 제공합니다. 또한 각 이벤트의 예약된 시작 시간과 종료 시간도 제공합니다.

내용

- Amazon EC2 인스턴스 상태 확인
- Amazon EC2 인스턴스의 상태 변경 이벤트
- Amazon EC2 인스턴스의 예약된 이벤트

Amazon EC2 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 EC2는 실행 중인 모든 EC2 인스턴스에서 자동 확인을 수행하여 하드웨어 및 소프트웨어 문제를 식별합니다. 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 섹션을 참조하세요. 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 확인에 실패한 Amazon EC2 인스턴스에 대한 CloudWatch 경보 생성 섹션을 참조하세요.

Amazon EC2 인스턴스를 모니터링하고 기본 문제로 인해 인스턴스가 손상된 경우 인스턴스를 자동으로 복구하는 Amazon CloudWatch 경보를 생성할 수도 있습니다. 자세한 내용은 자동 인스턴스 복구 섹션을 참조하세요.

목차

- 상태 확인 유형
- Amazon EC2 인스턴스 상태 확인 보기
- 상태 확인에 실패한 Amazon EC2 인스턴스에 대한 CloudWatch 경보 생성

상태 확인 유형

상태 확인에는 세 가지 유형이 있습니다.

- 시스템 상태 확인
- 인스턴스 상태 확인
- 연결된 EBS 상태 확인

시스템 상태 확인

시스템 상태 확인은 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 인스턴스를 중지한 후 시작할 수 있으며 대부분의 경우 이 인스턴스를 새 호스트로 마이그레이션합니다. 인스턴스 저장소 기반 인스턴스(Linux 인스턴스에서만 지원됨)의 경우 인스턴스를 종료하고 교체할 수 있습니다. 인스턴스 스토어 볼륨은 일시적이며 인스턴스가 중지되면 모든 데이터가 손실됩니다.

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

시스템 상태 검사에 실패하면 StatusCheckFailed_System 지표가 증가합니다.

베어 메탈 인스턴스

베어 메탈 인스턴스의 운영 체제에서 다시 시작하는 경우 시스템 상태 확인에서 일시적으로 실패 상태를 반환할 수 있습니다. 인스턴스를 사용할 수 있게 되면 시스템 상태 확인에서 통과 상태를 반환해야 합니다.

인스턴스 상태 확인

인스턴스 상태 검사에서는 개별 인스턴스에 대한 소프트웨어 및 네트워크 연결을 모니터링합니다. Amazon EC2는 네트워크 인터페이스(NIC)로 주소 확인 프로토콜(ARP)을 전송하여 인스턴스의 상태를 확인합니다. 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.

> **Note**
>
> 네트워크 구성에 `systemd-networkd`를 사용하는 최신 Linux 배포판은 이전 배포판과 다르게 상태 확인을 보고할 수 있습니다. 이러한 유형의 네트워크는 부팅 프로세스 중에 더 일찍 시작되어 인스턴스 상태에 영향을 미칠 수 있는 다른 시작 작업보다 먼저 완료될 수 있습니다. 네

트워크 가용성에 따라 달라지는 상태 확인은 다른 태스크가 완료되기 전에 정상 상태를 보고할 수 있습니다.

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
- 재부팅 중에 인스턴스 상태 확인은 인스턴스를 다시 사용할 수 있을 때까지 실패를 보고합니다.

인스턴스 상태 검사에 실패하면 StatusCheckFailed_Instance 지표가 증가합니다.

베어 메탈 인스턴스

베어 메탈 인스턴스의 운영 체제에서 다시 시작하는 경우 인스턴스 상태 확인에서 일시적으로 실패 상태를 반환할 수 있습니다. 인스턴스를 사용할 수 있게 되면 인스턴스 상태 확인에서 통과 상태를 반환해야 합니다.

연결된 EBS 상태 확인

연결된 EBS 상태 확인은 인스턴스에 연결된 Amazon EBS 볼륨이 연결 가능하고 I/O 작업을 완료할 수 있는지 모니터링합니다. `StatusCheckFailed_AttachedEBS` 지표는 인스턴스에 연결된 하나 이상의 EBS 볼륨이 I/O 작업을 완료할 수 없는 경우 손상을 나타내는 이진 값입니다. 이러한 상태 확인은 컴퓨팅 또는 Amazon EBS 인프라의 근본적인 문제를 감지합니다. 연결된 EBS 상태 확인 지표가 실패하면 AWS에서 문제가 해결될 때까지 기다리거나 영향을 받는 볼륨의 교체 또는 인스턴스 중지 후 재시작 등의 조치를 취할 수 있습니다.

다음은 연결된 EBS 상태 확인의 실패 원인이 되는 몇 가지 문제의 예입니다.

- EBS 볼륨의 기반이 되는 스토리지 하위 시스템의 하드웨어 또는 소프트웨어 문제
- EBS 볼륨의 연결성에 영향을 주는 물리적 호스트의 하드웨어 문제
- 인스턴스와 EBS 볼륨 간의 연결 문제

StatusCheckFailed_AttachedEBS 지표를 사용하여 워크로드의 복원성을 개선할 수 있습니다. 이 지표를 사용하여 상태 확인 결과를 기준으로 트리거되는 Amazon CloudWatch 경보를 생성할 수 있습니다. 예를 들어, 장기간의 영향이 감지되면 보조 인스턴스 또는 가용 영역으로 장애 조치할 수 있습니다. 아니면 EBS CloudWatch 지표를 사용하여 연결된 각 볼륨의 I/O 성능을 모니터링하여 손상된 볼륨을 감지하고 교체할 수 있습니다. 워크로드가 인스턴스에 연결된 EBS 볼륨으로 I/O를 제공하지 않고, 연결된 EBS 상태 확인에서 장애가 표시되는 경우 인스턴스를 중지하고 시작하여 새 호스트로 이동할 수 있습니다. 이렇게 하면 EBS 볼륨의 연결성에 영향을 미치는 기본적인 호스트 문제를 해결할 수 있습니다. 자세한 내용은 Amazon CloudWatch metrics for Amazon EBS를 참조하세요.

또한 연결된 EBS 상태 확인 실패를 탐지하도록 Amazon EC2 Auto Scaling 그룹을 구성한 다음, 영향을 받는 인스턴스를 새 인스턴스로 대체할 수 있습니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 손상된 Amazon EBS 볼륨이 있는 Auto Scaling 인스턴스 모니터링 및 교체를 참조하세요.

> **Note**
>
> 연결된 EBS 상태 확인 지표는 Nitro 인스턴스에만 사용할 수 있습니다.

Amazon EC2 인스턴스 상태 확인 보기

인스턴스 상태 확인이 실패할 경우에는 일반적으로 사용자가 인스턴스를 재부팅하거나 인스턴스 구성을 변경하는 등의 방법으로 문제를 직접 해결해야 합니다. 시스템 또는 인스턴스 상태 확인 실패 문제를 직접 해결하려면 상태 확인에 실패한 Amazon EC2 Linux 인스턴스 문제 해결 섹션을 참조하세요.

Console

상태 확인 보기

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스(Instances) 페이지의 상태 검사(Status check) 열에는 각 인스턴스의 운영 상태가 목록으로 표시됩니다.
4. 특정 인스턴스의 상태를 보려면 인스턴스를 선택하고 상태 및 경보 탭을 선택합니다.
5. 상태 확인에 대한 CloudWatch 지표를 검토하려면 상태 및 경보 탭에서 지표를 확장하여 다음 지표에 대한 그래프를 확인합니다.
   - 시스템에서 상태 확인 실패
   - 인스턴스에서 상태 확인 실패

- 연결된 EBS의 상태 확인 실패

자세한 내용은 the section called “상태 확인 지표” 섹션을 참조하세요.

AWS CLI

상태 확인 보기

describe-instance-status 명령을 사용합니다.

예: 실행 중인 모든 인스턴스의 상태 가져오기

```
aws ec2 describe-instance-status
```

예: 모든 인스턴스의 상태 가져오기

```
aws ec2 describe-instance-status --include-all-instances
```

예: 실행 중인 단일 인스턴스의 상태 가져오기

```
aws ec2 describe-instance-status --instance-ids i-1234567890abcdef0
```

예: 상태가 `impaired`인 모든 인스턴스 가져오기

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

PowerShell

상태 확인 보기

Get-EC2InstanceStatus 명령을 사용합니다.

예: 실행 중인 모든 인스턴스의 상태 가져오기

```
Get-EC2InstanceStatus
```

예: 모든 인스턴스의 상태 가져오기

```
Get-EC2InstanceStatus -IncludeAllInstance $true
```

예: 실행 중인 단일 인스턴스의 상태 가져오기

```
Get-EC2InstanceStatus -InstanceId i-1234567890abcdef0
```

예: 상태가 `impaired`인 모든 인스턴스 가져오기

```
Get-EC2InstanceStatus \
-Filter @{Name="instance-status.status"; Values="impaired"}
```

상태 확인에 실패한 Amazon EC2 인스턴스에 대한 CloudWatch 경보 생성

상태 확인 지표를 사용하여 인스턴스에 실패한 상태 확인이 있을 때 알리는 CloudWatch 경보를 생성할 수 있습니다.

누락된 지표 데이터 요소가 있는 경우 상태 검사 및 상태 검사 경보가 일시적으로 데이터 부족 상태로 전환될 수 있습니다. 드물기는 하지만, 지표 보고 시스템이 중단되면 인스턴스가 정상인 경우에도 이 문제가 발생할 수 있습니다. 상태 확인 실패 또는 경보 위반 대신 이 상태를 누락된 데이터로 처리하는 것이 좋습니다. 응답으로 인스턴스에서 중지, 종료, 재부팅 또는 복구 작업을 수행할 때 특히 중요합니다.

Console

이 예제는 인스턴스가 상태 확인에 실패할 때 알림을 보내는 경보를 구성합니다. 선택적으로 인스턴스를 중지, 종료 또는 복구할 수 있습니다.

상태 확인 경보를 생성하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택하고, 상태 검사(Status Checks) 탭을 선택한 후 작업(Actions), 상태 검사 경보 생성(Create status check alarm)을 선택합니다.
4. [CloudWatch 경보 관리(Manage CloudWatch alarms)] 페이지의 [경보 추가 또는 편집(Add or edit alarm)]에서 [경보 생성(Create an alarm)]을 선택합니다.
5. 경보 알림(Alarm notification)에서 토글을 켜서 Amazon Simple Notification Service(Amazon SNS) 알림을 구성합니다. 기존 Amazon SNS 주제를 선택하거나 이름을 입력하여 새 주제를 생성합니다.

수신자 목록에 이메일 주소를 추가하거나 새 주제를 생성하는 경우 Amazon SNS에서는 각 새 주소로 확인 이메일을 보냅니다. 각 수신자는 이메일에서 확인 링크를 선택해야 합니다. 확인된 주소만 경보 알림을 수신합니다.

6. 경보 작업(Alarm action)에서 토글을 켜서 경보가 트리거될 때 수행할 작업을 지정합니다. 작업을 선택합니다.
7. [경보 임곗값(Alarm thresholds)]에서 경보에 대한 지표와 기준을 선택합니다.

   샘플 그룹화 기준(Group samples by)(평균(Average)) 및 샘플링할 데이터 유형(Type of data to sample)(상태 확인 실패: 모두(Status check failed:either))을 기본 설정으로 두거나 요구 사항에 적합하게 변경할 수 있습니다.

   [연속 기간(Consecutive Period)]에서 평가 주기의 개수를 설정하고 [기간(Period)]에서 경보가 실행되고 이메일 전송이 이루어지기 전에 적용할 평가 주기의 시간 단위를 설정합니다.
8. (선택 사항) 샘플 지표 데이터의 경우 대시보드에 추가를 선택합니다.
9. Create를 선택합니다.

인스턴스 상태 경보를 변경해야 하는 경우 편집할 수 있습니다.

상태 확인 경보를 편집하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 작업, 모니터링, CloudWatch 경보 관리를 차례로 선택합니다.
4. [CloudWatch 경보 관리(Manage CloudWatch alarms)] 페이지의 [경보 추가 또는 편집(Add or edit alarm)]에서 [경보 편집(Edit an alarm)]을 선택합니다.
5. [경보 검색(Search for alarm)]에서 경보를 선택합니다.
6. 변경을 마치면 [업데이트(Update)]를 선택합니다.

AWS CLI

다음 예제에서는 인스턴스가 2번 이상 연속으로 인스턴스 검사 또는 시스템 상태 확인에 실패하면 경보가 SNS 주제에 알림을 게시합니다. 사용된 CloudWatch 지표는 `StatusCheckFailed`입니다.

상태 확인 경보를 생성하는 방법

1. 기존의 SNS 주제를 선택하거나 새로운 주제를 생성합니다. 자세한 내용은 AWS Command Line Interface 사용 설명서에서 AWS CLI에서 Amazon SNS 액세스를 참조하세요.
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 임곗값과 비교해야 하는 연속 기간의 수입니다. 이 예제에서는 2를 사용합니다. 경보 작업은 경보가 트리거될 때 수행할 작업입니다.

PowerShell

상태 확인 경보를 생성하는 방법

다음과 같이 Write-CWMetricAlarm cmdlet을 사용하여 인스턴스가 2번 이상 연속으로 상태 확인에 실패할 때 SNS 주제에 알림을 게시합니다.

```
Write-CWMetricAlarm `
    -AlarmName "StatusCheckFailed-Alarm-for-i-1234567890abcdef0" `
    -MetricName "StatusCheckFailed" `
    -Namespace "AWS/EC2" `
```

```
    -Statistic "Maximum" `
    -Dimension @{Name="InstanceId"; Values="i-1234567890abcdef0"} `
    -Unit "Count" `
    -Period 300 `
    -EvaluationPeriod 2 `
    -Threshold 1 `
    -ComparisonOperator "GreaterThanOrEqualToThreshold" `
    -AlarmAction "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 임곗값과 비교해야 하는 연속 기간의 수입니다. 이 예제에서는 2를 사용합니다. 경보 작업은 경보가 트리거될 때 수행할 작업입니다.

Amazon EC2 인스턴스의 상태 변경 이벤트

Amazon EC2는 인스턴스 상태가 변경되면 `EC2 Instance State-change Notification` 이벤트를 Amazon EventBridge로 보냅니다.

다음은 이 이벤트의 예제 데이터입니다. 이 예제에서는 인스턴스가 `pending` 상태가 되었습니다.

```
{
   "id":"7bf73129-1428-4cd3-a780-95db273d1602",
   "detail-type":"EC2 Instance State-change Notification",
   "source":"aws.ec2",
   "account":"123456789012",
   "time":"2021-11-11T21:29:54Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0"
   ],
   "detail":{
      "instance-id":"i-1234567890abcdef0",
      "state":"pending"
   }
}
```

`state`에 대해 가능한 값은 다음과 같습니다.

- `pending`
- `running`

- stopping
- stopped
- shutting-down
- terminated

인스턴스를 시작하면 pending 상태로 전환되고 나서 running 상태로 전환됩니다. 인스턴스를 중지하면 stopping 상태로 전환되고 나서 stopped 상태로 전환됩니다. 인스턴스를 종료하면 shutting-down 상태로 전환되고 나서 terminated 상태로 전환됩니다. 자세한 내용은 Amazon EC2 인스턴스 상태 변경 섹션을 참조하세요.

Amazon EC2 인스턴스의 상태가 변경될 때 이메일을 보내는 경보 생성

인스턴스 상태가 변경될 때 이메일 알림을 받으려면 Amazon SNS 주제를 생성한 다음 EC2 Instance State-change Notification 이벤트에 대한 EventBridge 규칙을 생성합니다.

SNS 주제를 생성하려면

1. https://console.aws.amazon.com/sns/v3/home에서 Amazon SNS 콘솔을 엽니다.
2. 탐색 창에서 주제를 선택합니다.
3. 주제 생성을 선택합니다.
4. 유형에서 표준을 선택합니다.
5. Name(이름)에 주제의 이름을 입력합니다.
6. 주제 생성을 선택합니다.
7. 구독 생성을 선택합니다.
8. 프로토콜에서 이메일을 선택합니다.
9. Endpoint(엔드포인트)에 알림을 받는 데 사용할 이메일 주소를 입력합니다.
10. 구독 생성을 선택합니다.
11. AWS Notification - Subscription Confirmation이라는 제목의 이메일 메시지를 받게 됩니다. 지시에 따라 구독을 확인합니다.

EventBridge 규칙을 생성하려면

1. https://console.aws.amazon.com/events/에서 Amazon EventBridge 콘솔을 엽니다.
2. Create rule을 선택합니다.

3. Name(이름)에 규칙의 이름을 입력합니다.
4. 규칙 유형(Rule type)에서 이벤트 패턴이 있는 규칙(Rule with an event pattern)을 생성합니다.
5. 다음을 선택합니다.
6. 이벤트 패턴(Event pattern)에서 다음을 수행하세요:
   a. 이벤트 소스(Event source)에서 AWS 서비스를 선택합니다.
   b. AWS 서비스로 EC2를 선택합니다.
   c. 이벤트 유형에서 EC2 인스턴스 상태 변경 알림을 선택합니다.
   d. 기본적으로 모든 인스턴스의 상태 변경에 대한 알림을 보냅니다. 원하는 경우 특정 상태 또는 특정 인스턴스를 선택할 수 있습니다.
7. 다음을 선택합니다.
8. 다음과 같이 대상을 지정합니다.
   a. 대상 유형(Target types)에서 AWS 서비스를 선택합니다.
   b. 대상 선택에서 SNS 주제를 선택합니다.
   c. Topic(주제)에서 이전 절차에서 생성한 SNS 주제를 선택합니다.
9. 다음을 선택합니다.
10. (선택 사항) 규칙에 태그를 추가합니다.
11. 다음을 선택합니다.
12. 규칙 생성을 선택합니다.
13. 규칙을 테스트하려면 상태 변경을 시작합니다. 예를 들어 중지된 인스턴스를 시작하거나 실행 중인 인스턴스를 중지하거나 인스턴스를 시작합니다. AWS Notification Message라는 제목의 이메일 메시지를 받게 됩니다. 이메일 본문에는 이벤트 데이터가 포함되어 있습니다.

Amazon EC2 인스턴스의 예약된 이벤트

인프라 신뢰성 및 성능을 보장하기 위해 AWS는 인스턴스를 재부팅, 중지 및 사용 중지하도록 이벤트를 예약할 수 있습니다. 이러한 이벤트들은 자주 발생하지 않습니다.

인스턴스 중 하나가 예약된 이벤트의 영향을 받는 경우 AWS는 AWS 계정과 연결된 이메일 주소를 사용하여 이메일로 미리 알립니다. 이메일은 시작 날짜 및 종료 날짜 등 이벤트에 대한 세부 정보를 제공합니다. 이벤트 유형에 따라 이벤트 타이밍을 제어하는 작업을 수행할 수 있습니다. AWS는 Amazon EventBridge를 사용하여 모니터링 및 관리할 수 있는 AWS Health 이벤트도 보냅니다. 자세한 내용은 Monitoring events in AWS Health with Amazon EventBridge를 참조하세요.

예약된 이벤트는 AWS에서 관리합니다. 인스턴스에 대한 이벤트를 예약할 수 없습니다. 그러나 다음을 수행할 수 있습니다.

- 인스턴스에 대해 예약된 이벤트를 확인합니다.
- 이메일 알림에서 태그를 포함하거나 제거하도록 예약된 이벤트 알림을 사용자 지정합니다.
- 예약된 특정 이벤트를 다시 예약합니다.
- 예약된 이벤트에 대한 사용자 지정 이벤트 기간을 생성합니다.
- 인스턴스가 재부팅, 중지 또는 사용 중지되도록 예약된 경우 관련 작업을 수행합니다.

예약된 이벤트 알림을 수신하는지 확인하려면 계정 페이지에서 연락처 정보를 확인합니다.

> **Note**
>
> 인스턴스가 예약된 이벤트의 영향을 받고 Auto Scaling 그룹의 일부인 경우 Amazon EC2 Auto Scaling은 결국 상태 확인의 일부로 인스턴스를 대체하며, 추가 작업이 필요하지 않습니다. Amazon EC2 Auto Scaling이 수행하는 상태 확인에 대한 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Auto Scaling 그룹의 인스턴스에 대한 상태 확인을 참조하세요.

예약된 이벤트 유형

Amazon EC2는 예약된 시간에 이벤트가 발생하는 인스턴스에 대해 다음과 같은 유형의 예약된 이벤트를 생성할 수 있습니다.

이벤트 유형	이벤트 코드	이벤트 작업
인스턴스 중지	`instance-stop`	예약된 시간에 인스턴스가 중지됩니다. 인스턴스를 다시 시작하면 새 호스트로 마이그레이션됩니다. Amazon EBS 루트 볼륨이 있는 인스턴스에만 적용됩니다.
인스턴스 만료	`instance-retirement`	예약된 시간에 Amazon EBS 루트 볼륨이 있는 경우 인스턴스가 중지되고, 인스턴스 저장

이벤트 유형	이벤트 코드	이벤트 작업
		소 루트 볼륨이 있는 경우 인스턴스가 종료됩니다.
인스턴스 재부팅	`instance-reboot`	예약된 시간에 인스턴스가 재부팅됩니다. 인스턴스는 호스트에 남아 있으며 재부팅 중에 호스트는 유지 관리 상태입니다. 이를 인플레이스 재부팅이라고 합니다.
시스템 재부팅	`system-reboot`	예약된 시간에 인스턴스는 재부팅되고 새 호스트로 마이그레이션됩니다. 이를 재부팅 마이그레이션이라고 합니다.
시스템 유지 관리	`system-maintenance`	예약된 시간에 네트워크 또는 전력 유지 관리로 인스턴스가 일시적인 영향을 받을 수 있습니다.

이벤트 유형 결정

인스턴스에 대해 예약된 이벤트 유형을 확인할 수 있습니다.

Console

이벤트 유형을 확인하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [이벤트(Events)]를 선택합니다.
3. 표에서 이벤트 유형 열에 이벤트 코드가 나타납니다.
4. 인스턴스의 이벤트만 표시하도록 표를 필터링하려면 검색 필드의 필터 목록에서 리소스 유형: 인스턴스를 선택하세요.

AWS CLI

인스턴스의 이벤트 유형을 확인하려면

describe-instance-status 명령을 사용합니다. 인스턴스에 연결된 예약된 이벤트가 있는 경우 출력은 예약된 이벤트에 대한 정보를 제공합니다.

```
aws ec2 describe-instance-status \
    --instance-id i-1234567890abcdef0 \
    --query InstanceStatuses[].Events
```

출력의 예시는 다음과 같습니다. 예약된 이벤트 코드는 `system-reboot`입니다.

```
[
    "Events": [
        {
            "InstanceEventId": "instance-event-0d59937288b749b32",
            "Code": "system-reboot",
            "Description": "The instance is scheduled for a reboot",
            "NotAfter": "2020-03-14T22:00:00.000Z",
            "NotBefore": "2020-03-14T20:00:00.000Z",
            "NotBeforeDeadline": "2020-04-05T11:00:00.000Z"
        }
    ]
]
```

PowerShell

인스턴스의 이벤트 유형을 확인하려면

Get-EC2InstanceStatus cmdlet을 사용합니다. 인스턴스에 연결된 예약된 이벤트가 있는 경우 출력은 예약된 이벤트에 대한 정보를 제공합니다.

```
(Get-EC2InstanceStatus `
    -InstanceId i-1234567890abcdef0).Events
```

출력의 예시는 다음과 같습니다. 예약된 이벤트 코드는 `system-reboot`입니다.

```
Code              : system-reboot
Description       : The instance is scheduled for a reboot
InstanceEventId   : instance-event-0d59937288b749b32
NotAfter          : 2020-03-14T22:00:00.000Z
```

```
NotBefore          : 2020-03-14T20:00:00.000Z
NotBeforeDeadline : 2020-04-05T11:00:00.000Z
```

내용

- 중지 또는 사용 중지하도록 Amazon EC2 인스턴스 관리
- 재부팅하도록 예약된 Amazon EC2 인스턴스 관리
- 유지 관리를 위해 예약된 Amazon EC2 인스턴스 관리
- Amazon EC2 인스턴스에 영향을 미치는 예약된 이벤트 보기
- EC2 인스턴스에 대한 예약된 이벤트 알림 사용자 지정
- EC2 인스턴스에 대해 예약된 이벤트 다시 예약
- Amazon EC2 인스턴스에 영향을 미치는 예약된 이벤트에 대한 사용자 지정 이벤트 기간 생성

중지 또는 사용 중지하도록 Amazon EC2 인스턴스 관리

AWS가 인스턴스의 기본 호스트에서 복구 불가능한 결함을 감지하면 인스턴스의 루트 볼륨 유형에 따라 인스턴스의 중지 또는 종료를 예약합니다.

- 인스턴스에 Amazon EBS 루트 볼륨이 있는 경우 인스턴스 중지가 예약됩니다.
- 인스턴스에 인스턴스 저장소 루트 볼륨이 있는 경우 인스턴스 종료가 예약됩니다.

자세한 내용은 인스턴스 만료 섹션을 참조하세요.

> **Important**
>
> 인스턴스가 중단되거나 최대 절전 모드로 전환되거나 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 중단되거나 최대 절전 모드로 전환되거나 종료되기 전에 반드시 저장해야 합니다.

수행할 수 있는 작업

EBS 루트 볼륨이 있는 인스턴스에서 수행할 수 있는 작업

예약된 `instance-stop` 이벤트 알림을 수신하면 다음 작업 중 하나를 수행할 수 있습니다.

- 예약된 중지 대기: 예약된 유지 관리 기간에 인스턴스가 중지될 때까지 기다릴 수 있습니다.
- 수동 중지 및 시작 수행: 사용자에게 편한 시간에 직접 인스턴스를 중지하고 시작할 수 있으며, 이때 인스턴스는 새 호스트로 마이그레이션됩니다. 이는 인스턴스 재부팅과 동일하지 않습니다. 자세한 내용은 Amazon EC2 인스턴스 중지 및 시작 섹션을 참조하세요.
- 중지 및 시작 자동화: 예약된 `instance-stop` 이벤트에 대한 응답으로 즉시 중지 및 시작을 자동화할 수 있습니다. 자세한 내용은 AWS Health 사용 설명서의 AWS Health에서 이벤트에 대한 응답으로 EC2 인스턴스에서 자동으로 작업 실행을 참조하세요.

인스턴스 저장소 루트 볼륨이 있는 인스턴스에서 수행할 수 있는 작업

예약된 `system-retirement` 이벤트 알림을 수신한 경우 데이터를 유지하려면 다음 작업을 수행할 수 있습니다.

1. 최신 AMI에서 대체 인스턴스를 시작하세요.
2. 인스턴스 종료를 예약하기 전에 필요한 모든 데이터를 대체 인스턴스로 마이그레이션하세요.
3. 원본 인스턴스를 종료하거나 예약한 대로 종료될 때까지 기다리세요.

수행할 수 있는 작업에 대한 자세한 내용은 인스턴스 만료 섹션을 참조하세요.

재부팅하도록 예약된 Amazon EC2 인스턴스 관리

AWS가 업데이트 설치나 기본 호스트 유지 관리 등의 작업을 수행해야 하는 경우 인스턴스 재부팅을 예약할 수 있습니다. 예약된 재부팅 중에 인스턴스는 다음과 같이 이벤트에 따라 동일한 호스트에서 유지되거나 다른 호스트로 마이그레이션됩니다.

- `instance-reboot` 이벤트를 트리거합니다
  - 재부팅하는 동안 인스턴스는 호스트에 남아 있습니다. 이를 인플레이스 재부팅이라고 합니다.
  - 현재 호스트가 유지 관리를 수행합니다.
  - 일반적으로 몇 초 만에 완료됩니다.
- `system-reboot` 이벤트를 트리거합니다
  - 재부팅 중에 인스턴스는 새 호스트로 마이그레이션됩니다. 이를 재부팅 마이그레이션이라고 합니다.
  - 일반적으로 몇 분 안에 완료됩니다.

인스턴스에 대해 예약된 이벤트 유형을 확인하려면 이벤트 유형 결정 섹션을 참조하세요.

수행할 수 있는 작업

예약된 `instance-reboot` 또는 `system-reboot` 이벤트 알림을 수신하면 다음 작업 중 하나를 수행할 수 있습니다.

- 예약된 재부팅 대기: 예약된 유지 관리 기간에 인스턴스 재부팅이 수행될 때까지 기다릴 수 있습니다.
- 재부팅 다시 예약: 사용자에게 편한 날짜 및 시간으로 인스턴스 재부팅을 다시 예약할 수 있습니다.
- 사용자 시작 재부팅 수행: 사용자에게 편한 시간에 인스턴스를 직접 수동으로 재부팅할 수 있습니다. 그러나 결과는 이벤트에 따라 달라집니다.
  - `instance-reboot` 이벤트 - 인스턴스가 현재 하드웨어에 남아 있고(인플레이스 재부팅), 호스트 유지 관리가 수행되지 않으며, 이벤트가 열린 상태로 유지됩니다.
  - `system-reboot` 이벤트를 트리거합니다
    - 인스턴스에서 재부팅 마이그레이션이 활성화된 경우 사용자 시작 재부팅은 인스턴스를 새 하드웨어로 마이그레이션하려고 시도합니다. 성공하면 이벤트가 지워집니다. 실패하면 인플레이스 재부팅이 발생하고 이벤트는 예약된 상태로 유지됩니다.
    - 인스턴스에서 재부팅 마이그레이션이 비활성화된 경우 사용자 시작 재부팅은 인스턴스를 동일한 하드웨어에 유지하고(인플레이스 재부팅), 호스트 유지 관리가 수행되지 않으며, 이벤트는 예약된 상태로 유지됩니다. 예약된 이벤트가 최종적으로 발생하면 AWS는 인스턴스를 새 하드웨어로 이동합니다(재부팅 마이그레이션).

AWS에서 인스턴스를 재부팅한 후

AWS에서 인스턴스를 재부팅한 후 다음이 적용됩니다.

- 예약된 이벤트가 지워집니다.
- 이벤트 설명이 업데이트됩니다.
- `instance-reboot` 이벤트의 경우:
  - 기본 호스트의 유지 관리가 완료됩니다.
- `system-reboot` 이벤트의 경우:
  - 인스턴스가 새 호스트로 이동합니다.
  - 인스턴스는 IP 주소 및 DNS 이름을 유지합니다.
  - 로컬 인스턴스 저장소 볼륨의 모든 데이터는 보존됩니다.
- 인스턴스가 완전히 부팅된 후에 인스턴스를 사용할 수 있습니다.

대체 옵션

사용자 시작 재부팅의 경우 재부팅 이벤트를 다시 예약하거나 재부팅 마이그레이션을 활성화할 수 없지만 예약된 유지 관리 기간에 정상 작업을 유지하려면 다음을 수행할 수 있습니다.

- EBS 루트 볼륨이 있는 인스턴스의 경우
  - 수동으로 인스턴스를 중지한 후 시작해 새 호스트로 마이그레이션합니다. 이는 인스턴스 수동 재부팅(인스턴스가 동일한 호스트에서 유지됨)과 동일하지 않습니다.
  - 선택적으로 예약된 재부팅 이벤트에 대한 응답으로 즉시 인스턴스 중지 및 시작을 자동화합니다. 자세한 내용은 AWS Health 사용 설명서의 AWS Health에서 이벤트에 대한 응답으로 EC2 인스턴스에서 자동으로 작업 실행을 참조하세요.

> **Important**
>
> 인스턴스가 중지되면 인스턴스 저장소 볼륨의 데이터가 유실됩니다. 자세한 내용은 Amazon EC2 인스턴스 중지 및 시작 섹션을 참조하세요.

- 인스턴스 저장소 루트 볼륨이 있는 인스턴스의 경우
  1. 최신 AMI에서 대체 인스턴스를 시작하세요.
  2. 예약된 유지 관리 기간 전에 필요한 모든 데이터를 대체 인스턴스로 마이그레이션하세요.
  3. 원래 인스턴스를 종료하세요.

재부팅 마이그레이션 활성화 또는 비활성화

인스턴스가 `system-reboot` 이벤트에 예약되면 이벤트 전에 인스턴스를 재부팅할 수 있습니다. 사용자 시작 재부팅의 결과는 인스턴스의 재부팅 마이그레이션 설정에 따라 달라집니다.

- 활성화됨 - 사용자 시작 재부팅이 인스턴스를 새 하드웨어로 마이그레이션하려고 시도합니다(재부팅 마이그레이션). 성공하면 이벤트가 지워집니다. 실패하면 인플레이스 재부팅이 발생하고 이벤트는 예약된 상태로 유지됩니다. 활성화한 경우에도 인스턴스가 재부팅 마이그레이션 요구 사항을 충족하는 경우에만 재부팅 마이그레이션이 발생할 수 있습니다.
- 비활성화됨 - 사용자 시작 재부팅은 인스턴스를 동일한 하드웨어에 유지하고(인플레이스 재부팅), 호스트 유지 관리가 수행되지 않으며, 이벤트는 예약된 상태로 유지됩니다. 예약된 이벤트가 최종적으로 발생하면 AWS는 인스턴스를 새 하드웨어로 이동합니다(재부팅 마이그레이션).

마이그레이션을 포함한 재부팅은 인플레이스 재부팅보다 오래 걸립니다.

- 인플레이스 재부팅: 약 30초
- 마이그레이션을 포함한 재부팅: 몇 분

> **Note**
>
> 사용자 시작 재부팅 마이그레이션의 경우 `system-reboot` 이벤트 알림을 수신하는 인스턴스가 기본적으로 활성화됩니다.

재부팅 마이그레이션 활성화 요구 사항

다음 기준을 충족하는 인스턴스에서 재부팅 마이그레이션을 활성화할 수 있습니다.

인스턴스 타입

모든 인스턴스 유형이 재부팅 마이그레이션 활성화를 지원하는 것은 아닙니다. 재부팅 마이그레이션 활성화를 지원하는 인스턴스 유형을 볼 수 있습니다.

Console

재부팅 마이그레이션 활성화를 지원하는 인스턴스 유형을 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 인스턴스 유형(Instance Types)을 선택합니다.
3. 필터 표시줄에 Reboot Migration support: supported를 입력합니다. 문자를 입력할 때 필터 이름이 나타나면 해당 필터 이름을 선택할 수 있습니다.

   인스턴스 유형 테이블에 재부팅 마이그레이션 활성화를 지원하는 모든 인스턴스 유형이 표시됩니다.

AWS CLI

재부팅 마이그레이션 활성화를 지원하는 인스턴스 유형을 보려면

describe-instance-types 명령을 `reboot-migration-support` 필터와 함께 사용합니다.

```
aws ec2 describe-instance-types \
    --filters Name=reboot-migration-support,Values=supported \
    --query "InstanceTypes[*].[InstanceType]" \
```

```
    --output text | sort
```

PowerShell

재부팅 마이그레이션 활성화를 지원하는 인스턴스 유형을 보려면

Get-EC2InstanceType cmdlet을 `reboot-migration-support` 필터와 함께 사용합니다.

```
Get-EC2InstanceType `
    -Filter @{Name="reboot-migration-support";Values="true"} | `
    Select InstanceType | Sort-Object InstanceType
```

Tenancy

- 공유
- Dedicated Instance

자세한 내용은 Amazon EC2 전용 인스턴스 섹션을 참조하세요.

제한 사항

다음과 같은 특성을 가진 인스턴스에는 재부팅 마이그레이션이 지원되지 않습니다.

- 플랫폼: Xen 하이퍼바이저에서 기본적으로 실행되는 인스턴스
- 인스턴스 크기: `metal` 인스턴스
- 테넌시: 전용 호스트. 전용 호스트의 경우 전용 호스트 자동 복구를 대신 사용합니다.
- 스토리지: 인스턴스 저장소 볼륨이 있는 인스턴스
- 네트워킹: Elastic Fabric Adapter를 사용하는 인스턴스
- Auto Scaling: Auto Scaling 그룹에 포함된 인스턴스

재부팅 마이그레이션 활성화 또는 비활성화 단계

인스턴스는 `system-reboot` 이벤트를 수신하면 기본적으로 재부팅 마이그레이션이 활성화됩니다. 사용자 시작 재부팅 중에 인스턴스가 동일한 하드웨어에 유지되도록 재부팅 마이그레이션을 비활성화할 수 있습니다(인플레이스 재부팅).

`default` 구성은 지원되지 않는 인스턴스의 재부팅 마이그레이션을 활성화하지 않습니다. 자세한 내용은 재부팅 마이그레이션 활성화 요구 사항 섹션을 참조하세요.

실행 중이거나 중지된 인스턴스에서 재부팅 마이그레이션을 비활성화하거나 활성화할 수 있습니다.

AWS CLI

재부팅 마이그레이션을 비활성화하려면

modify-instance-maintenance-options 명령을 사용하고 `--reboot-migration` 파라미터를 `disabled`로 설정합니다.

```
aws ec2 modify-instance-maintenance-options \

    --instance-id i-0abcdef1234567890 \

    --reboot-migration disabled
```

재부팅 마이그레이션을 활성화하려면

modify-instance-maintenance-options 명령을 사용하고 `--reboot-migration` 파라미터를 `default`로 설정합니다.

```
aws ec2 modify-instance-maintenance-options \

    --instance-id i-0abcdef1234567890 \

    --reboot-migration default
```

PowerShell

재부팅 마이그레이션을 비활성화하려면

Edit-EC2InstanceMaintenanceOption cmdlet을 사용합니다.

```
Edit-EC2InstanceMaintenanceOption `
    -InstanceId  `
    -RebootMigration Disabled
```

재부팅 마이그레이션을 활성화하려면

Edit-EC2InstanceMaintenanceOption cmdlet을 사용합니다.

```
Edit-EC2InstanceMaintenanceOption `
    -InstanceId i-1234567890abcdef0 `
```

```
    -RebootMigration Enabled
```

유지 관리를 위해 예약된 Amazon EC2 인스턴스 관리

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.

수행할 수 있는 작업

EBS 루트 볼륨이 있는 인스턴스에서 수행할 수 있는 작업

`system-maintenance` 이벤트 알림을 수신하면 다음 작업 중 하나를 수행할 수 있습니다.

- 예약된 유지 관리 대기: 유지 관리가 예약된 대로 수행될 때까지 기다릴 수 있습니다.
- 수동 중지 및 시작 수행: 사용자가 인스턴스를 중지하고 시작할 수 있으며, 이때 인스턴스는 새 호스트로 마이그레이션됩니다. 이는 인스턴스 재부팅과 동일하지 않습니다. 자세한 내용은 Amazon EC2 인스턴스 중지 및 시작 섹션을 참조하세요.
- 중지 및 시작 자동화: 예약된 유지 관리 이벤트에 대한 응답으로 즉시 중지 및 시작을 자동화할 수 있습니다. 자세한 내용은 AWS Health 사용 설명서의 AWS Health에서 이벤트에 대한 응답으로 EC2 인스턴스에서 자동으로 작업 실행을 참조하세요.

인스턴스 저장소 루트 볼륨이 있는 인스턴스에서 수행할 수 있는 작업

`system-maintenance` 이벤트 알림을 수신하면 다음 작업 중 하나를 수행할 수 있습니다.

- 예약된 유지 관리 대기: 유지 관리가 예약된 대로 수행될 때까지 기다릴 수 있습니다.
- 대체 인스턴스 시작: 예약된 유지 관리 기간에 정상 작업을 유지하는 방법:

1. 최신 AMI에서 대체 인스턴스를 시작하세요.
2. 예약된 유지 관리 기간 전에 필요한 모든 데이터를 대체 인스턴스로 마이그레이션하세요.
3. 원래 인스턴스를 종료하세요.

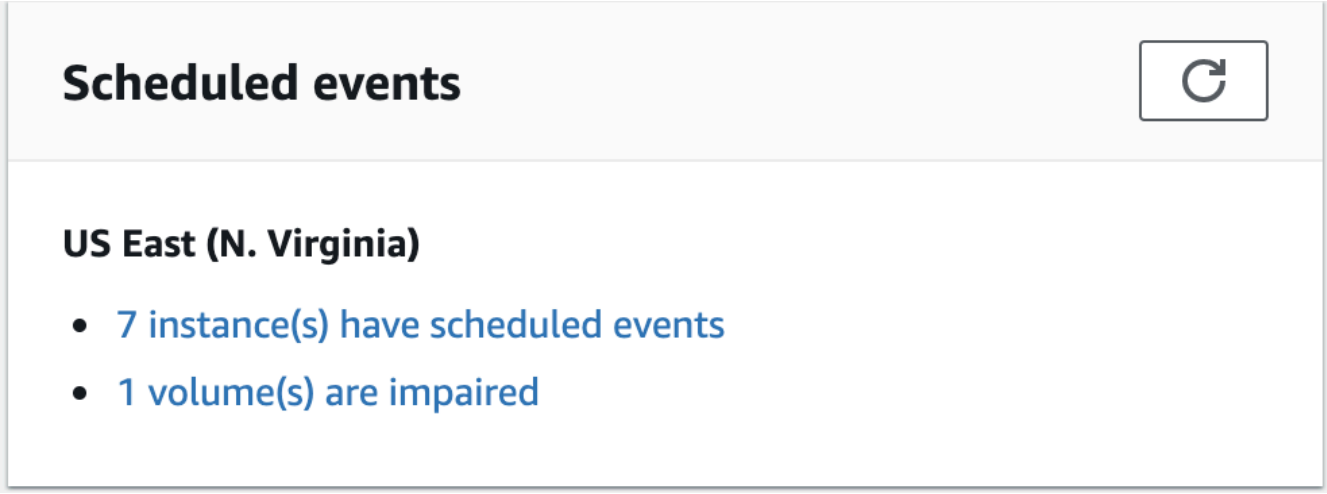
Amazon EC2 인스턴스에 영향을 미치는 예약된 이벤트 보기

예약된 이벤트에 대한 알림 메시지를 이메일로 받는 것 외에도, 예약된 이벤트를 확인할 수 있습니다.

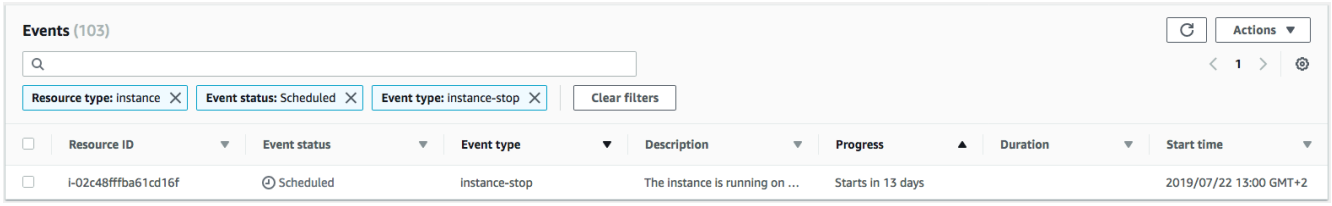
Console

인스턴스에 예약된 이벤트를 확인하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 대시보드에 예약된 이벤트 아래에 연결된 이벤트가 있는 모든 리소스가 표시됩니다.



3. 자세한 내용은 탐색 창에서 이벤트를 선택하세요. 연결된 이벤트가 있는 모든 리소스가 표시됩니다. 이벤트 유형, 리소스 유형 및 가용 영역 등의 특징을 기준으로 필터링할 수 있습니다.



AWS CLI

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

인스턴스에 예약된 이벤트를 확인하는 방법

다음 Get-EC2InstanceStatus 명령을 사용합니다.

```
(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 메타데이터에서 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" 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 메타데이터에서 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" 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"
  }
]
```

AWS Health

AWS Health Dashboard을(를) 사용하여 인스턴스에 영향을 줄 수 있는 이벤트에 대해 알아볼 수 있습니다. AWS Health Dashboard는 미해결 문제, 예약된 변경, 기타 알림이라는 세 그룹으로 문제를 정리합니다. 예약된 변경 그룹에는 진행 중이거나 예정된 항목이 포함됩니다.

자세한 내용은 AWS Health 사용 설명서의 AWS Health Dashboard 시작하기를 참조하세요.

EC2 인스턴스에 대한 예약된 이벤트 알림 사용자 지정

이메일 알림에 태그를 포함하도록 예약된 이벤트 알림을 사용자 지정할 수 있습니다. 이렇게 하면 영향을 받는 리소스(인스턴스 또는 전용 호스트)를 더 쉽게 식별하고 다가오는 이벤트에 대한 작업의 우선 순위를 지정할 수 있습니다.

태그를 포함하도록 이벤트 알림을 사용자 지정할 때 다음을 포함하도록 선택할 수 있습니다.

- 영향을 받는 리소스와 연결된 모든 태그
- 영향을 받는 리소스와 연결된 특정 태그만

예를 들어 모든 인스턴스에 `application`, `costcenter`, `project` 및 `owner` 태그를 할당하는 경우, 이벤트 알림에 이러한 모든 태그를 포함하도록 선택하거나 이벤트 알림에 `owner` 및 `project` 태그만 표시하기 위해 이러한 태그만 포함하도록 선택할 수 있습니다.

포함할 태그를 선택하면 이벤트 알림에 영향을 받는 리소스와 연결된 리소스 ID(인스턴스 전용 호스트 ID 또는 ID)와 태그 키 및 값 페어가 포함됩니다.

업무

- 이벤트 알림에 태그 포함
- 이벤트 알림에서 태그 제거
- 이벤트 알림에 포함할 태그 보기

이벤트 알림에 태그 포함

포함하도록 선택한 태그는 선택한 리전의 모든 리소스(인스턴스 및 전용 호스트)에 적용됩니다. 다른 지역에서 이벤트 알림을 사용자 지정하려면 먼저 필요한 리전을 선택한 후 다음 단계를 수행합니다.

Console

이벤트 알림에 태그를 포함하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Events]를 선택합니다.
3. 작업, Manage event notifications(이벤트 알림 관리)를 선택합니다.

4. 이벤트 알림에 태그 포함을 켭니다.
5. 이벤트 알림에 포함할 태그에 따라 다음 중 하나를 수행합니다.
   - 영향을 받는 인스턴스 또는 전용 호스트와 연결된 모든 태그를 포함하려면 모든 태그 포함을 선택합니다.
   - 포함할 태그를 선택하려면 포함할 태그 선택을 선택한 다음 태그 키를 선택하거나 입력합니다.
6. 저장(Save)을 선택합니다.

AWS CLI

이벤트 알림에 모든 태그를 포함하려면

register-instance-event-notification-attributes 명령을 사용하고 `IncludeAllTagsOfInstance` 파라미터를 `true`로 설정합니다.

```
aws ec2 register-instance-event-notification-attributes \
    --instance-tag-attribute "IncludeAllTagsOfInstance=true"
```

이벤트 알림에 특정 태그를 포함하려면

register-instance-event-notification-attributes 명령을 사용하고 `InstanceTagKeys` 파라미터를 사용하여 포함할 태그를 지정합니다.

```
aws ec2 register-instance-event-notification-attributes \
    --instance-tag-attribute 'InstanceTagKeys=["tag_key_1", "tag_key_2",
 "tag_key_3"]'
```

PowerShell

이벤트 알림에 모든 태그를 포함하려면

Register-EC2InstanceEventNotificationAttribute cmdlet을 사용합니다.

```
Register-EC2InstanceEventNotificationAttribute `
    -InstanceTagAttribute_IncludeAllTagsOfInstance $true
```

이벤트 알림에 특정 태그를 포함하려면

Register-EC2InstanceEventNotificationAttribute cmdlet을 사용합니다.

```
Register-EC2InstanceEventNotificationAttribute `
    -InstanceTagAttribute_InstanceTagKey tag_key_1, tag_key_2, tag_key_3
```

이벤트 알림에서 태그 제거

이벤트 알림에서 태그를 제거할 수 있습니다.

Console

이벤트 알림에서 태그를 제거하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Events]를 선택합니다.
3. 작업, Manage event notifications(이벤트 알림 관리)를 선택합니다.
4. 이벤트 알림에서 모든 태그를 제거하려면 이벤트 알림에 태그 포함을 끕니다.
5. 이벤트 알림에서 특정 태그를 제거하려면 해당 태그 키에 대해 X를 선택합니다.
6. 저장(Save)을 선택합니다.

AWS CLI

이벤트 알림에서 모든 태그를 제거하려면

deregister-instance-event-notification-attributes 명령을 사용하고 `IncludeAllTagsOfInstance` 파라미터를 `false`로 설정합니다.

```
aws ec2 deregister-instance-event-notification-attributes \
    --instance-tag-attribute "IncludeAllTagsOfInstance=false"
```

이벤트 알림에서 태그를 제거하려면

deregister-instance-event-notification-attributes 명령을 사용하고 `InstanceTagKeys` 파라미터를 사용하여 제거할 태그를 지정합니다.

```
aws ec2 deregister-instance-event-notification-attributes \
    --instance-tag-attribute 'InstanceTagKeys=["tag_key_3"]'
```

PowerShell

이벤트 알림에서 모든 태그를 제거하려면

Unregister-EC2InstanceEventNotificationAttribute cmdlet을 사용합니다.

```
Unregister-EC2InstanceEventNotificationAttribute `
    -InstanceTagAttribute_IncludeAllTagsOfInstance $false
```

이벤트 알림에서 태그를 제거하려면

Unregister-EC2InstanceEventNotificationAttribute cmdlet을 사용합니다.

```
Unregister-EC2InstanceEventNotificationAttribute `
    -InstanceTagAttribute_InstanceTagKey tag_key_3
```

이벤트 알림에 포함할 태그 보기

이벤트 알림에 포함할 태그를 볼 수 있습니다.

Console

이벤트 알림에 포함할 태그를 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Events]를 선택합니다.
3. 작업, Manage event notifications(이벤트 알림 관리)를 선택합니다.

AWS CLI

이벤트 알림에 포함할 태그를 보는 방법

describe-instance-event-notification-attributes 명령을 사용합니다.

```
aws ec2 describe-instance-event-notification-attributes
```

PowerShell

이벤트 알림에 포함할 태그를 보는 방법

Get-EC2InstanceEventNotificationAttribute cmdlet을 사용합니다.

```
Get-EC2InstanceEventNotificationAttribute
```

EC2 인스턴스에 대해 예약된 이벤트 다시 예약

이벤트가 적합한 특정 날짜와 시간에 발생하도록 다시 예약할 수 있습니다. 이벤트를 다시 예약한 후 업데이트된 날짜가 표시되기까지 1~2분이 걸릴 수 있습니다.

제한 사항

- 이벤트 기한이 있는 이벤트만 다시 예약할 수 있습니다. 이 이벤트는 이벤트 기한까지 다시 예약될 수 있습니다. 이벤트 기한 날짜는 기한 열(콘솔) 및 `NotBeforeDeadline` 필드(AWS CLI)에 표시됩니다.
- 아직 시작하지 않은 이벤트만 다시 예약할 수 있습니다. 시작 시간은 시작 시간 열(콘솔) 및 `NotBefore` 필드(AWS CLI)에 표시됩니다. 다음 5분 안에 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.

Console

이벤트를 다시 예약하는 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. 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. 저장(Save)을 선택합니다.

업데이트된 이벤트 시작 시간이 콘솔에 반영되려면 1~2분이 걸릴 수 있습니다.

AWS CLI

이벤트를 다시 예약하는 방법

modify-instance-event-start-time 명령을 사용합니다.

```
aws ec2 modify-instance-event-start-time \
    --instance-id i-1234567890abcdef0 \
    --instance-event-id instance-event-0d59937288b749b32 \
    --not-before 2020-03-25T10:00:00.000
```

PowerShell

이벤트를 다시 예약하는 방법

Edit-EC2InstanceEventStartTime cmdlet을 사용합니다.

```
Edit-EC2InstanceEventStartTime `
    -InstanceId i-1234567890abcdef0 `
    -InstanceEventId instance-event-0d59937288b749b32 `
    -NotBefore 2020-03-25T10:00:00.000
```

Amazon EC2 인스턴스에 영향을 미치는 예약된 이벤트에 대한 사용자 지정 이벤트 기간 생성

Amazon EC2 인스턴스를 재부팅, 중지 또는 종료하는 예약된 이벤트에 대해 매주 반복되는 사용자 지정 이벤트 기간을 정의할 수 있습니다. 하나 이상의 인스턴스를 이벤트 기간에 연결할 수 있습니다. 해당 인스턴스에 대해 예약된 이벤트가 계획되면 AWS는 관련 이벤트 기간 내에 이벤트를 예약합니다.

이벤트 기간을 사용하여 워크로드가 적은 기간 동안 발생하는 이벤트 기간을 지정하여 워크로드 가용성을 극대화할 수 있습니다. 또한 이벤트 기간을 내부 유지 관리 스케줄에 맞출 수도 있습니다.

시간 범위 세트를 지정하여 이벤트 기간을 정의합니다. 최소 시간 범위는 2시간입니다. 결합된 시간 범위는 최소 4시간 이상이어야 합니다.

인스턴스 ID 또는 인스턴스 태그를 사용하여 하나 이상의 인스턴스를 이벤트 기간에 연결할 수 있습니다. 호스트 ID를 사용하여 전용 호스트를 이벤트 기간에 연결할 수도 있습니다.

> **Warning**
>
> 이벤트 기간은 인스턴스를 중지, 재부팅 또는 종료하는 예약된 이벤트에만 적용할 수 있습니다.
> 다음에는 이벤트 기간을 적용할 수 없습니다.
>
> - 예약된 신속 이벤트 및 네트워크 유지 관리 이벤트.
> - 자동 인스턴스 복구 및 계획되지 않은 재부팅과 같은 예약되지 않은 유지 관리.

이벤트 기간 작업

- 고려 사항
- 이벤트 기간 생성
- 이벤트 기간에 대상 연결
- 이벤트 기간에서 대상 연결 해제
- 이벤트 기간 수정
- 이벤트 기간 삭제

고려 사항

- 모든 이벤트 기간 시간은 UTC입니다.
- 최소 주간 이벤트 기간은 4시간입니다.
- 이벤트 기간 내의 시간 범위는 각각 2시간 이상이어야 합니다.
- 하나의 대상 유형(인스턴스 ID, 전용 호스트 ID 또는 인스턴스 태그)만 이벤트 기간에 연결할 수 있습니다.
- 대상(인스턴스 ID, 전용 호스트 ID 또는 인스턴스 태그)은 하나의 이벤트 기간에만 연결할 수 있습니다.
- 이벤트 기간에는 최대 100개의 인스턴스 ID 또는 50개의 전용 호스트 ID 또는 50개의 인스턴스 태그를 연결할 수 있습니다. 인스턴스 태그는 제한 없이 원하는 수의 인스턴스에 연결할 수 있습니다.
- AWS 리전당 최대 200개의 이벤트 기간을 생성할 수 있습니다.
- 이벤트 기간에 연결된 여러 인스턴스에서 예약된 이벤트가 동시에 발생할 수 있습니다.
- AWS가 이미 이벤트를 예약한 경우 이벤트 기간을 수정해도 예약된 이벤트의 시간은 변경되지 않습니다. 이벤트에 기한 날짜가 있는 경우 이벤트 스케줄을 조정할 수 있습니다.

- 예약된 이벤트 전에 인스턴스를 중지하고 시작할 수 있습니다. 그러면 인스턴스가 새 호스트로 마이그레이션되고 이벤트가 지워집니다.

이벤트 기간 생성

하나 이상의 이벤트 기간을 생성할 수 있습니다. 각 이벤트 기간에 대해 시간 블록을 하나 이상 지정합니다. 예를 들어 매일 오전 4시에 2시간 동안 발생하는 시간 블록으로 이벤트 기간을 생성할 수 있습니다. 또는 일요일 오전 2시~오전 4시 및 수요일 오전 3시~오전 5시에 발생하는 시간 블록으로 이벤트 기간을 생성할 수도 있습니다.

이벤트 기간은 사용자가 삭제할 때까지 매주 반복됩니다.

Console

이벤트 기간을 생성하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Events]를 선택합니다.
3. [작업(Actions)], [Windows 이벤트 관리(Manage event windows)]를 차례로 선택합니다.
4. [인스턴스 이벤트 기간 생성(Create instance event window)]을 선택합니다.
5. [이벤트 기간 이름(Event window name)]에 이벤트 기간을 설명하는 이름을 입력합니다.
6. [이벤트 기간 일정(Event window schedule)]에서 cron 일정 작성기를 사용하거나 시간 범위를 지정하여 이벤트 기간에서 시간 블록을 지정하도록 선택합니다.
   - [Cron 일정 작성기(Cron schedule builder)]를 선택한 경우 다음을 지정합니다.
     1. [요일(UTC)(Days (UTC))]에서 이벤트 기간이 발생하는 요일을 지정합니다.
     2. [시작 시간(UTC)(Start time (UTC))]에서 이벤트 기간이 시작되는 시간을 지정합니다.
     3. [지속 시간(Duration)]에서 이벤트 기간에 있는 시간 블록의 지속 시간을 지정합니다. 시간 블록당 최소 지속 시간은 2시간입니다. 이벤트 기간의 최소 지속 시간은 총 4시간 이상이어야 합니다. 모든 시간은 협정 세계시(UTC)입니다.
   - [시간 범위(Time ranges)]를 선택한 경우 [새 시간 범위 추가(Add new time range)]를 선택하고 시작 날짜 및 시간, 종료 날짜 및 시간을 지정합니다. 각 시간 범위에 대해 반복합니다. 시간 범위당 최소 지속 시간은 2시간입니다. 결합된 모든 시간 범위의 최소 지속 시간은 총 4시간 이상이어야 합니다.
7. (선택 사항) 대상 세부 정보에서 하나 이상의 인스턴스를 이벤트 창에 연결하세요. 인스턴스 ID 또는 인스턴스 태그를 사용하여 인스턴스를 연결하세요. 호스트 ID를 사용하여 전용 호스트를

연결하세요. 유지 관리를 위해 이러한 대상이 예약되면 이 이벤트 기간에 이벤트가 발생합니다.

대상을 기간에 연결하지 않고 이벤트 기간을 생성할 수도 있습니다. 나중에 기간을 수정하여 하나 이상의 대상을 연결할 수 있습니다.

8. (선택 사항) [이벤트 기간 태그(Event window tags)]에서 [태그 추가(Add tag)]를 선택하고 해당 태그의 키와 값을 입력합니다. 각 태그에 대해 반복합니다.
9. [이벤트 기간 생성(Create event window)]을 선택합니다.

AWS CLI

시간 범위로 이벤트 기간을 생성하는 방법

create-instance-event-window 명령을 사용합니다.

```
aws ec2 create-instance-event-window \
    --time-range StartWeekDay=monday,StartHour=2,EndWeekDay=wednesday,EndHour=8 \
    --tag-specifications "ResourceType=instance-event-
window,Tags=[{Key=K1,Value=V1}]" \
    --name myEventWindowName
```

출력의 예시는 다음과 같습니다.

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "TimeRanges": [
            {
                "StartWeekDay": "monday",
                "StartHour": 2,
                "EndWeekDay": "wednesday",
                "EndHour": 8
            }
        ],
        "Name": "myEventWindowName",
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
```

```
        ]
    }
}
```

cron 표현식으로 이벤트 기간을 생성하는 방법

create-instance-event-window 명령을 사용합니다.

```
aws ec2 create-instance-event-window \
    --cron-expression "* 21-23 * * 2,3" \
    --tag-specifications "ResourceType=instance-event-
window,Tags=[{Key=K1,Value=V1}]" \
    --name myEventWindowName
```

출력의 예시는 다음과 같습니다.

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
        ]
    }
}
```

PowerShell

시간 범위로 이벤트 기간을 생성하는 방법

New-EC2InstanceEventWindow cmdlet을 사용합니다.

```
$timeRange = New-Object Amazon.EC2.Model.InstanceEventWindowTimeRangeRequest
$timeRange.StartWeekDay = "monday"
$timeRange.EndWeekDay = "wednesday"
$timeRange.StartHour = 2
$timeRange.EndHour = 8
```

```
$tag = @{Key="key1"; Value="value1"}
$tagspec = New-Object Amazon.EC2.Model.TagSpecification
$tagspec.ResourceType = "instance-event-window"
$tagspec.Tags.Add($tag)
New-EC2InstanceEventWindow `
    -Name my-event-window `
    -TagSpecification $tagspec `
    -TimeRange @($timeRange)
```

출력의 예시는 다음과 같습니다.

```
AssociationTarget     :
CronExpression        :
InstanceEventWindowId : iew-0abcdef1234567890
Name                  : my-event-window
State                 : creating
Tags                  : {key1}
TimeRanges            : {Amazon.EC2.Model.InstanceEventWindowTimeRange}
```

cron 표현식으로 이벤트 기간을 생성하는 방법

New-EC2InstanceEventWindow cmdlet을 사용합니다.

```
$tag = @{Key="key1"; Value="value1"}
$tagspec = New-Object Amazon.EC2.Model.TagSpecification
$tagspec.ResourceType = "instance-event-window"
$tagspec.Tags.Add($tag)
New-EC2InstanceEventWindow `
    -Name my-event-window `
    -TagSpecification $tagspec`
    -CronExpression "* 21-23 * * 2,3"
```

출력의 예시는 다음과 같습니다.

```
AssociationTarget     :
CronExpression        : * 21-23 * * 2,3
InstanceEventWindowId : iew-0abcdef1234567890
Name                  : my-event-window
State                 : creating
Tags                  : {key1}
TimeRanges            : {}
```

이벤트 기간에 대상 연결

이벤트 기간을 생성한 후 대상을 이벤트 기간과 연결할 수 있습니다. 이벤트 기간에는 한 가지 유형의 대상만 연결할 수 있습니다. 인스턴스 ID, 전용 호스트 ID 또는 인스턴스 태그를 지정할 수 있습니다.

Console

대상을 이벤트 기간에 연결하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [이벤트(Events)]를 선택합니다.
3. 수정할 이벤트 기간을 선택합니다.
4. 작업, 인스턴스 이벤트 기간 수정을 선택합니다.

AWS CLI

인스턴스 태그를 이벤트 기간에 연결하는 방법

associate-instance-event-window 명령을 사용합니다.

```
aws ec2 associate-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "InstanceTags=[{Key=k2,Value=v2},{Key=k1,Value=v1}]"
```

출력의 예시는 다음과 같습니다.

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [
                {
                    "Key": "k2",
                    "Value": "v2"
                },
                {
                    "Key": "k1",
                    "Value": "v1"
```

```
                }
            ],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

인스턴스 ID를 이벤트 기간에 연결하려면

associate-instance-event-window 명령을 사용합니다.

```
aws ec2 associate-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "InstanceIds=i-1234567890abcdef0,i-0598c7d356eba48d7"
```

출력의 예시는 다음과 같습니다.

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [
                "i-1234567890abcdef0",
                "i-0598c7d356eba48d7"
            ],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

전용 호스트를 이벤트 기간에 연결하는 방법

associate-instance-event-window 명령을 사용합니다.

```
aws ec2 associate-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "DedicatedHostIds=h-029fa35a02b99801d"
```

출력의 예시는 다음과 같습니다.

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [],
            "DedicatedHostIds": [
                "h-029fa35a02b99801d"
            ]
        },
        "State": "creating"
    }
}
```

PowerShell

인스턴스 태그를 이벤트 기간에 연결하는 방법

Register-EC2InstanceEventWindow cmdlet을 사용합니다.

```
$tag1 = @{Key="key1"; Value="value1"}
$tag2 = @{Key="key2"; Value="value2"}
Register-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -AssociationTarget_InstanceTag @($tag1,$tag2)
```

인스턴스 ID를 이벤트 기간에 연결하려면

Register-EC2InstanceEventWindow cmdlet을 사용합니다.

```
Register-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -AssociationTarget_InstanceId i-1234567890abcdef0, i-0598c7d356eba48d7
```

전용 호스트를 이벤트 기간에 연결하는 방법

Register-EC2InstanceEventWindow cmdlet을 사용합니다.

```
Register-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -AssociationTarget_DedicatedHostId h-029fa35a02b99801d
```

이벤트 기간에서 대상 연결 해제

Console

이벤트 기간에서 대상을 연결 해제하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [이벤트(Events)]를 선택합니다.
3. 수정할 이벤트 기간을 선택합니다.
4. 작업, 인스턴스 이벤트 기간 수정을 선택합니다.

AWS CLI

이벤트 기간에서 인스턴스 태그를 연결 해제하는 방법

disassociate-instance-event-window 명령을 사용합니다.

```
aws ec2 disassociate-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "InstanceTags=[{Key=k2,Value=v2},{Key=k1,Value=v1}]"
```

출력의 예시는 다음과 같습니다.

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
```

```
}
```

이벤트 기간에서 인스턴스 ID를 연결 해제하려면

disassociate-instance-event-window 명령을 사용합니다.

```
aws ec2 disassociate-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "InstanceIds=i-1234567890abcdef0,i-0598c7d356eba48d7"
```

출력의 예시는 다음과 같습니다.

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

이벤트 기간에서 전용 호스트를 연결 해제하는 방법

disassociate-instance-event-window 명령을 사용합니다.

```
aws ec2 disassociate-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target DedicatedHostIds=h-029fa35a02b99801d
```

출력의 예시는 다음과 같습니다.

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
```

```
            "InstanceIds": [],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

PowerShell

이벤트 기간에서 인스턴스 태그를 연결 해제하는 방법

Unregister-EC2InstanceEventWindow cmdlet을 사용합니다.

```
$tag1 = @{Key="key1"; Value="value1"}
$tag2 = @{Key="key2"; Value="value2"}
Unregister-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -AssociationTarget_InstanceTag @($tag1, $tag2)
```

이벤트 기간에서 인스턴스 ID를 연결 해제하려면

Unregister-EC2InstanceEventWindow cmdlet을 사용합니다.

```
Unregister-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -AssociationTarget_InstanceId i-1234567890abcdef0, i-0598c7d356eba48d7
```

이벤트 기간에서 전용 호스트를 연결 해제하는 방법

Unregister-EC2InstanceEventWindow cmdlet을 사용합니다.

```
Unregister-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -AssociationTarget_DedicatedHostId h-029fa35a02b99801d
```

이벤트 기간 수정

ID를 제외한 이벤트 기간의 모든 필드를 수정할 수 있습니다. 예를 들어 일광 절약 시간이 시작될 때 이벤트 기간 일정을 수정할 수 있습니다. 기존 이벤트 기간의 경우 대상을 추가하거나 제거할 수 있습니다.

이벤트 기간을 수정할 때 시간 범위 또는 cron 표현식을 수정할 수 있지만 둘 다 수정할 수는 없습니다.

Console

이벤트 기간을 수정하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Events]를 선택합니다.
3. [작업(Actions)], [Windows 이벤트 관리(Manage event windows)]를 차례로 선택합니다.
4. 수정할 이벤트 기간을 선택한 후 [작업(Actions)], [인스턴스 이벤트 기간 수정(Modify instance event window)]을 차례로 선택합니다.
5. 이벤트 기가에서 필드를 수정한 후 [이벤트 기간 수정(Modify event window)]을 선택합니다.

AWS CLI

이벤트 기간의 시간 범위를 수정하는 방법

modify-instance-event-window 명령을 사용합니다.

```
aws ec2 modify-instance-event-window
    --instance-event-window-id iew-0abcdef1234567890 \
    --time-range StartWeekDay=monday,StartHour=2,EndWeekDay=wednesday,EndHour=8
```

출력의 예시는 다음과 같습니다.

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "TimeRanges": [
            {
                "StartWeekDay": "monday",
                "StartHour": 2,
                "EndWeekDay": "wednesday",
                "EndHour": 8
            }
        ],
        "Name": "myEventWindowName",
        "AssociationTarget": {
            "InstanceIds": [
                "i-0abcdef1234567890",
```

```
                "i-0be35f9acb8ba01f0"
            ],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
        ]
    }
}
```

이벤트 기간의 시간 범위 세트를 수정하는 방법

modify-instance-event-window 명령을 사용합니다.

```
aws ec2 modify-instance-event-window
    --instance-event-window-id iew-0abcdef1234567890 \
    --time-range '[{"StartWeekDay": "monday", "StartHour": 2, "EndWeekDay":
 "wednesday", "EndHour": 8},
   {"StartWeekDay": "thursday", "StartHour": 2, "EndWeekDay": "friday",
 "EndHour": 8}]'
```

출력의 예시는 다음과 같습니다.

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "TimeRanges": [
            {
                "StartWeekDay": "monday",
                "StartHour": 2,
                "EndWeekDay": "wednesday",
                "EndHour": 8
            },
            {
                "StartWeekDay": "thursday",
                "StartHour": 2,
                "EndWeekDay": "friday",
                "EndHour": 8
```

```
            }
        ],
        "Name": "myEventWindowName",
        "AssociationTarget": {
            "InstanceIds": [
                "i-0abcdef1234567890",
                "i-0be35f9acb8ba01f0"
            ],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
        ]
    }
}
```

이벤트 기간의 cron 표현식을 수정하는 방법

modify-instance-event-window 명령을 사용합니다.

```
aws ec2 modify-instance-event-window
    --instance-event-window-id iew-0abcdef1234567890 \
    --cron-expression "* 21-23 * * 2,3"
```

출력의 예시는 다음과 같습니다.

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [
                "i-0abcdef1234567890",
                "i-0be35f9acb8ba01f0"
            ],
            "Tags": [],
            "DedicatedHostIds": []
```

```
        },
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
        ]
    }
}
```

PowerShell

이벤트 기간의 시간 범위를 수정하는 방법

Edit-EC2InstanceEventWindow cmdlet을 사용합니다.

```
$timeRange1 = New-Object Amazon.EC2.Model.InstanceEventWindowTimeRangeRequest
$timeRange1.StartWeekDay = "monday"
$timeRange1.EndWeekDay = "wednesday"
$timeRange1.StartHour = 2
$timeRange1.EndHour = 8
$timeRange2 = New-Object Amazon.EC2.Model.InstanceEventWindowTimeRangeRequest
$timeRange2.StartWeekDay = "thursday"
$timeRange2.EndWeekDay = "friday"
$timeRange2.StartHour = 1
$timeRange2.EndHour = 6
Edit-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -TimeRange @($timeRange1, $timeRange2)
```

이벤트 기간의 cron 표현식을 수정하는 방법

Edit-EC2InstanceEventWindow cmdlet을 사용합니다.

```
Edit-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -CronExpression "* 21-23 * * 2,3"
```

이벤트 기간 삭제

이벤트 기간을 한 번에 하나씩 삭제할 수 있습니다.

Console

이벤트 기간을 삭제하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Events]를 선택합니다.
3. [작업(Actions)], [Windows 이벤트 관리(Manage event windows)]를 차례로 선택합니다.
4. 삭제할 이벤트 기간을 선택한 후 [작업(Actions)], [인스턴스 이벤트 기간 삭제(Delete instance event window)]를 차례로 선택합니다.
5. 메시지가 표시되면 **delete**를 입력한 후 삭제를 선택합니다.

AWS CLI

이벤트 기간을 삭제하는 방법

delete-instance-event-window 명령을 사용하고 삭제할 이벤트 기간을 지정합니다.

```
aws ec2 delete-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890
```

이벤트 기간을 강제 삭제하는 방법

현재 이벤트 기간이 대상과 연결되어 있는 경우 `--force-delete` 파라라미터를 사용합니다.

```
aws ec2 delete-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890 \
    --force-delete
```

PowerShell

이벤트 기간을 삭제하는 방법

Remove-EC2InstanceEventWindow cmdlet을 사용합니다.

```
Remove-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890
```

이벤트 기간을 강제 삭제하는 방법

Remove-EC2InstanceEventWindow cmdlet을 사용합니다.

```
Remove-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -ForceDelete $true
```

CloudWatch를 사용하여 인스턴스 모니터링

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 EC2 인스턴스에 대한 세부 모니터링 관리 섹션을 참조하세요.

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 CloudWatch 결제 및 비용 정보는 Amazon CloudWatch 사용 설명서의 CloudWatch 결제 및 비용을 참조하세요.

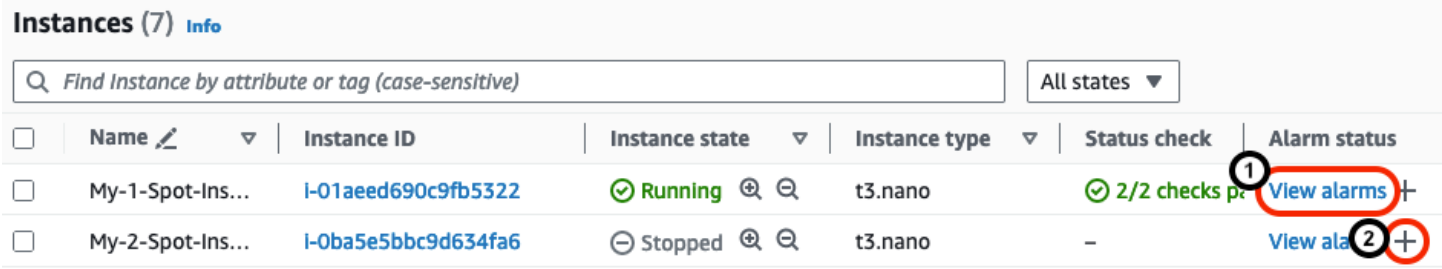
내용

- Amazon EC2 콘솔에서 EC2 인스턴스에 대한 CloudWatch 경보 관리
- EC2 인스턴스에 대한 세부 모니터링 관리
- 인스턴스에 사용 가능한 CloudWatch 지표
- Amazon EC2 콘솔을 사용하여 CloudWatch 에이전트를 설치 및 구성하고 지표 추가
- 인스턴스에 대한 CloudWatch 지표 통계
- 인스턴스에 대한 모니터링 그래프 보기
- 인스턴스에 대해 CloudWatch 경보 만들기
- 인스턴스를 중지, 종료, 재부팅 또는 복구하는 경보 만들기

Amazon EC2 콘솔에서 EC2 인스턴스에 대한 CloudWatch 경보 관리

Amazon EC2 콘솔의 인스턴스 화면에서 인스턴스에 대한 Amazon CloudWatch 경보를 관리할 수 있습니다. 인스턴스 테이블의 경보 상태 열에는 두 개의 콘솔 컨트롤이 있습니다. 하나는 경보를 보기 위

한 컨트롤이고 다른 하나는 경보 생성 또는 편집을 위한 컨트롤입니다. 다음 스크린샷은 이러한 콘솔 컨트롤을 나타냅니다. 1번(경보 보기) 및 2번(경보 생성 또는 편집을 위한 + 기호).



인스턴스 화면에서 경보 보기

인스턴스 화면에서 각 인스턴스의 경보를 볼 수 있습니다.

인스턴스 화면에서 인스턴스의 경보를 보는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스 테이블에서 선택한 인스턴스에 대해 경보 보기(위 스크린샷에서 번호가 1)를 선택합니다.
4. ***i-1234567890abcdef0***의 경보 세부 정보 창에서 경보 이름을 선택하여 CloudWatch 콘솔에서 경보를 봅니다.

인스턴스 화면에서 경보 생성

인스턴스 화면에서 각 인스턴스에 대한 경보를 생성할 수 있습니다.

인스턴스 화면에서 인스턴스에 대한 경보를 생성하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스 테이블에서 선택한 인스턴스에 대해 더하기 기호(위 스크린샷에서 번호가 2)를 선택합니다.
4. CloudWatch 경보 관리 화면에서 경보를 생성합니다. 자세한 내용은 인스턴스에 대해 CloudWatch 경보 만들기 섹션을 참조하세요.

인스턴스 화면에서 경보 편집

인스턴스 화면에서 각 인스턴스에 대한 경보를 편집할 수 있습니다.

인스턴스 화면에서 인스턴스에 대한 경보를 편집하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스 테이블에서 선택한 인스턴스에 대해 더하기 기호(위 스크린샷에서 번호가 2)를 선택합니다.
4. CloudWatch 경보 관리 화면에서 경보를 편집합니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 경보 편집 또는 삭제를 참조하세요.

EC2 인스턴스에 대한 세부 모니터링 관리

Amazon CloudWatch는 기본 모니터링과 세부 모니터링이라는 두 가지 모니터링 범주를 제공합니다. 기본적으로 인스턴스는 기본 모니터링 기능이 구성되어 있습니다. 필요에 따라 세부 모니터링을 활성화하여 운영 문제를 보다 신속하게 식별하고 조치를 취할 수 있습니다. 인스턴스가 실행 중이거나 중지되었을 때 또는 시작 시 세부 모니터링을 활성화하거나 비활성화할 수 있습니다.

인스턴스에 대한 세부 모니터링 기능을 활성화해도 연결된 EBS 볼륨 모니터링에는 영향이 없습니다. 자세한 내용은 Amazon CloudWatch metrics for Amazon EBS를 참조하세요.

다음 표는 인스턴스에 대한 기본 모니터링과 세부 모니터링의 차이를 설명합니다.

모니터링 유형	설명	요금
기본 모니터링	상태 확인 지표는 1분 기간으로 제공됩니다. 다른 모든 지표는 5분 기간으로 제공됩니다.	무료입니다.
세부 모니터링	상태 확인 지표를 포함한 모든 지표는 1분 기간으로 제공됩니다. 이러한 데이터 수준을 얻으려면 인스턴스에 대해 해당 수준을 사용하도록 설정해야 합니다. 세부 모니터링을 활성화한 인스턴스의 경우 유사한 인스턴스 그룹	Amazon EC2에서 CloudWatch로 전송하는 지표에 따라 요금이 청구됩니다. 데이터 스토리지에는 요금이 부과되지 않습니다. 자세한 내용은 Amazon CloudWatch 요금 페이지의 유료 티어 및 예제 1 - EC2 세부 모니터링을 참조하세요.

모니터링 유형	설명	요금
	간에 집계된 데이터를 얻을 수도 있습니다.	

내용

- 필수 권한
- 시작 시 세부 모니터링 활성화
- 세부 모니터링 관리

필수 권한

인스턴스에 대한 세부 모니터링을 활성화하려면 사용자에게 MonitorInstances API 작업을 사용할 권한이 있어야 합니다. 인스턴스에 대한 세부 모니터링을 비활성화하려면 사용자에게 UnmonitorInstances API 작업을 사용할 권한이 있어야 합니다.

시작 시 세부 모니터링 활성화

다음 절차에 따라 시작 시 세부 모니터링을 사용합니다. 기본적으로 인스턴스는 기본 모니터링을 사용합니다.

Console

인스턴스 시작 시 세부 모니터링을 활성화하려면

Amazon EC2 콘솔을 사용하여 인스턴스를 시작할 때 고급 세부 정보 아래에서 세부 CloudWatch 모니터링 확인란을 선택합니다.

AWS CLI

인스턴스 시작 시 세부 모니터링을 활성화하려면

run-instances 명령을 `--monitoring` 옵션과 함께 사용합니다.

```
--monitoring Enabled=true
```

PowerShell

인스턴스 시작 시 세부 모니터링을 활성화하려면

`-Monitoring` 파라미터와 함께 New-EC2Instance cmdlet을 사용합니다.

```
-Monitoring $true
```

세부 모니터링 관리

다음 절차에 따라 실행 중이거나 중지된 인스턴스에 대한 세부 모니터링을 관리합니다.

Console

세부 모니터링을 관리하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다.
4. 작업, 모니터링 및 문제 해결, 세부 모니터링 관리를 선택합니다.
5. 세부 모니터링 페이지에서 세부 모니터링에 대해 다음 중 하나를 수행합니다.
   - 세부 모니터링 - 활성화를 선택합니다.
   - 기본 모니터링 - 활성화 선택을 취소합니다.
6. 확인을 선택합니다.

AWS CLI

세부 모니터링을 활성화하는 방법

다음 monitor-instances 명령을 사용합니다.

```
aws ec2 monitor-instances --instance-ids i-1234567890abcdef0
```

세부 모니터링을 비활성화하는 방법

unmonitor-instances 명령을 사용합니다.

```
aws ec2 unmonitor-instances --instance-ids i-1234567890abcdef0
```

PowerShell

세부 모니터링을 활성화하는 방법

Start-EC2InstanceMonitoring cmdlet을 사용합니다.

```
Start-EC2InstanceMonitoring -InstanceId i-1234567890abcdef0
```

세부 모니터링을 비활성화하는 방법

Stop-EC2InstanceMonitoring cmdlet을 사용합니다.

```
Stop-EC2InstanceMonitoring -InstanceId i-1234567890abcdef0
```

인스턴스에 사용 가능한 CloudWatch 지표

Amazon EC2는 측정치를 Amazon CloudWatch로 전송합니다. AWS Management Console, AWS CLI 또는 API를 사용하여 Amazon EC2가 CloudWatch로 전송하는 지표를 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. Minimum, Maximum, Average 통계의 경우 EC2에서 제공하는 지표의 최소 세분화 수준은 1분입니다.

AWS Management Console 또는 AWS CLI를 사용하여 사용 가능한 지표를 보는 방법에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서의 사용 가능한 지표 보기를 참조하세요.

이 측정치에 대한 통계를 얻는 방법에 대한 자세한 내용은 인스턴스에 대한 CloudWatch 지표 통계 단원을 참조하세요.

목차

- 인스턴스 지표
- CPU 크레딧 지표
- 전용 호스트 지표
- Nitro 기반 인스턴스용 Amazon EBS 지표
- 상태 확인 지표
- 트래픽 미러링 지표
- Auto Scaling 그룹 지표
- Amazon EC2 지표 차원
- Amazon EC2 사용량 지표

인스턴스 지표

AWS/EC2 네임스페이스에는 다음 인스턴스 지표가 포함되어 있습니다.

측정치	설명	단위	의미 있는 통계
CPUUtilization	Amazon EC2에서 EC2 인스턴스 실행에 사용하는 물리적 CPU 시간의 비율로, 사용자 코드와 Amazon EC2 코드를 실행하는 데 소요되는 시간이 모두 포함됩니다. 대략적으로 CPUUtilization 은 게스트 CPUUtilization 및 하이퍼바이저 CPUUtilization 의 합입니다. 레거시 디바이스 시뮬레이션, 비레거시 디바이스 구성, 인터럽트가 많은 워크로드, 실시간 마이그레이션, 실시간 업데이트 등의 요인으로 인해 운영 체제의 도구에 CloudWatch와 다른 비율이 표시될 수 있습니다.	%	• 평균 • 최소 • Maximum
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 나누세요. 인스턴스 스토어 볼륨이 없으면 값이 0이거나 측정치가 보고되지 않습니다.	개수	• 합계 • 평균 • 최소 • Maximum
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 나누세요.	개수	• 합계 • 평균 • 최소 • Maximum

측정치	설명	단위	의미 있는 통계
	인스턴스 스토어 볼륨이 없으면 값이 0이거나 측정치가 보고되지 않습니다.		
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. CloudWatch 지표 수학 함수 DIFF_TIME 을 사용하여 초당 바이트 수를 찾을 수도 있습니다. 예를 들어 CloudWatch에서 DiskReadBytes 을 m1로 그래프로 표시한 경우 지표 수학 공식 m1/(DIFF_TIME(m1)) 은 지표(바이트/초)를 반환합니다. DIFF_TIME 및 기타 지표 수학 함수에 대한 자세한 내용은 Amazon CloudWatch User Guide의 Use metric math를 참조하세요. 인스턴스 스토어 볼륨이 없으면 값이 0이거나 측정치가 보고되지 않습니다.	바이트	• Sum • 평균 • 최소 • Maximum

측정치	설명	단위	의미 있는 통계
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. CloudWatch 지표 수학 함수 DIFF_TIME 을 사용하여 초당 바이트 수를 찾을 수도 있습니다. 예를 들어 CloudWatch에서 DiskWriteBytes 을 m1로 그래프로 표시한 경우 지표 수학 공식 m1/(DIFF_TIME(m1)) 은 지표(바이트/초)를 반환합니다. DIFF_TIME 및 기타 지표 수학 함수에 대한 자세한 내용은 Amazon CloudWatch User Guide의 Use metric math를 참조하세요. 인스턴스 스토어 볼륨이 없으면 값이 0이거나 측정치가 보고되지 않습니다.	바이트	• Sum • 평균 • 최소 • Maximum

측정치	설명	단위	의미 있는 통계
MetadataNoToken	토큰을 사용하지 않는 방법으로 인스턴스 메타데이터 서비스(IMDS)에 성공적으로 액세스한 횟수입니다. 이 지표는 토큰을 사용하지 않는 인스턴스 메타데이터 서비스 버전 1(IMDSv1)을 사용 중인 인스턴스 메타데이터에 액세스하는 프로세스가 있는지 확인하는 데 사용됩니다. 모든 요청이 토큰 지원 세션을 사용하는 경우(즉, 인스턴스 메타데이터 서비스 버전 2(IMDSv2)) 값은 0입니다. 자세한 내용은 인스턴스 메타데이터 서비스 버전 2 사용으로 전환 섹션을 참조하세요.	Nitro 인스턴스: 없음 Xen 인스턴스: 개수	• Sum • 백분위수
MetadataNoTokenRejected	ImDSv1이 비활성화된 후 ImDSv1 직접 호출을 시도한 횟수입니다. 이 지표가 나타나면 ImDSv1 직접 호출이 시도되었지만 거부되었음을 나타냅니다. IMDSv1을 다시 활성화하거나 모든 직접 호출에서 IMDSv2를 사용하도록 할 수 있습니다. 자세한 내용은 인스턴스 메타데이터 서비스 버전 2 사용으로 전환 섹션을 참조하세요.	Nitro 인스턴스: 없음 Xen 인스턴스: 개수	• Sum • 백분위수

측정치	설명	단위	의미 있는 통계
NetworkIn	모든 네트워크 인터페이스에서 인스턴스가 받은 바이트 수입니다. 이 측정치는 단일 인스턴스로 들어오는 네트워크 트래픽의 볼륨을 식별합니다. 보고된 숫자는 해당 기간에 수신된 바이트 수입니다. 기본(5분) 모니터링을 사용하고 통계가 합계인 경우, 이 숫자를 300으로 나누어 바이트/초를 찾을 수 있습니다. 세부(1분) 모니터링으로 설정되어 있고 통계가 합계인 경우 60으로 나눕니다. CloudWatch 지표 수학 함수 DIFF_TIME 을 사용하여 초당 바이트 수를 찾을 수도 있습니다. 예를 들어 CloudWatch에서 NetworkIn 을 m1로 그래프로 표시한 경우 지표 수학 공식 m1/(DIFF_TIME(m1)) 은 지표(바이트/초)를 반환합니다. DIFF_TIME 및 기타 지표 수학 함수에 대한 자세한 내용은 Amazon CloudWatch User Guide의 Use metric math를 참조하세요.	바이트	• Sum • 평균 • 최소 • Maximum

측정치	설명	단위	의미 있는 통계
NetworkOut	모든 네트워크 인터페이스에서 인스턴스가 보낸 바이트 수입니다. 이 측정치는 단일 인스턴스에서 나가는 네트워크 트래픽의 볼륨을 식별합니다. 보고된 숫자는 해당 기간에 전송된 바이트 수입니다. 기본(5분) 모니터링을 사용하고 통계가 합계인 경우, 이 숫자를 300으로 나누어 바이트/초를 찾을 수 있습니다. 세부(1분) 모니터링으로 설정되어 있고 통계가 합계인 경우 60으로 나눕니다. CloudWatch 지표 수학 함수 DIFF_TIME 을 사용하여 초당 바이트 수를 찾을 수도 있습니다. 예를 들어 CloudWatch에서 NetworkOut 을 m1로 그래프로 표시한 경우 지표 수학 공식 m1/(DIFF_TIME(m1)) 은 지표(바이트/초)를 반환합니다. DIFF_TIME 및 기타 지표 수학 함수에 대한 자세한 내용은 Amazon CloudWatch User Guide의 Use metric math를 참조하세요.	바이트	• Sum • 평균 • 최소 • Maximum

측정치	설명	단위	의미 있는 통계
NetworkPacketsIn	모든 네트워크 인터페이스에서 인스턴스가 받은 패킷 수입니다. 이 지표는 단일 인스턴스에서 수신 트래픽의 볼륨을 패킷 수 기준으로 식별합니다. 이 지표는 기본 모니터링에만 사용할 수 있습니다(5분간). 인스턴스가 수신한 PPS(패킷/초) 수를 계산하려면 이 수를 300으로 나눕니다. CloudWatch 지표 수학 함수 DIFF_TIME 을 사용하여 초당 패킷 수를 찾을 수도 있습니다. 예를 들어 CloudWatch에서 NetworkPacketsIn 을 m1로 그래프로 표시한 경우 지표 수학 공식 m1/(DIFF_TIME(m1)) 은 지표(패킷/초)를 반환합니다. DIFF_TIME 및 기타 지표 수학 함수에 대한 자세한 내용은 Amazon CloudWatch User Guide의 Use metric math를 참조하세요.	개수	• 합계 • 평균 • 최소 • Maximum

측정치	설명	단위	의미 있는 통계
NetworkPacketsOut	모든 네트워크 인터페이스에서 인스턴스가 보낸 패킷 수입니다. 이 지표는 단일 인스턴스에서 발신 트래픽의 볼륨을 패킷 수 기준으로 식별합니다. 이 지표는 기본 모니터링에만 사용할 수 있습니다(5분간). 인스턴스가 5분 동안 전송한 PPS(패킷/초) 수를 계산하려면 통계 값 합계를 300으로 나눕니다. CloudWatch 지표 수학 함수 DIFF_TIME 을 사용하여 초당 패킷 수를 찾을 수도 있습니다. 예를 들어 CloudWatch에서 NetworkPacketsOut 을 m1로 그래프로 표시한 경우 지표 수학 공식 m1/(DIFF_TIME(m1)) 은 지표(패킷/초)를 반환합니다. DIFF_TIME 및 기타 지표 수학 함수에 대한 자세한 내용은 Amazon CloudWatch User Guide의 Use metric math를 참조하세요.	개수	• 합계 • 평균 • 최소 • Maximum

CPU 크레딧 지표

AWS/EC2 네임스페이스에는 성능 순간 확장 가능 인스턴스에 대한 다음 CPU 크레딧 지표가 포함되어 있습니다.

측정치	설명	단위	의미 있는 통계
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).	크레딧(vCPU-분)	• Sum • 평균 • 최소 • Maximum

측정치	설명	단위	의미 있는 통계
	CPU 크레딧 지표는 5분 간격으로만 제공됩니다. 5분 이상의 시간을 지정할 경우 Sum 통계 대신 Average 통계를 사용하세요.		
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 지표는 5분 간격으로만 제공됩니다.	크레딧 (vCPU-분)	• Sum • 평균 • 최소 • Maximum

측정치	설명	단위	의미 있는 통계
CPUSurplusCreditBalance	unlimited 값이 0일 때 CPUCreditBalance 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 지표는 5분 간격으로만 제공됩니다.	크레딧 (vCPU-분)	• Sum • 평균 • 최소 • Maximum
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 standard로 전환됩니다. CPU 크레딧 지표는 5분 간격으로만 제공됩니다.	크레딧 (vCPU-분)	• Sum • 평균 • 최소 • Maximum

전용 호스트 지표

AWS/EC2 네임스페이스에는 다음과 같이 전용 호스트의 지표가 포함되어 있습니다.

지표	설명	단위	의미 있는 통계
DedicatedHostCPUUtilization	전용 호스트에서 실행 중인 인스턴스에서 현재 사용 중인 할당된 컴퓨팅 용량의 백분율입니다.	%	• Sum • 평균 • 최소 • Maximum

Nitro 기반 인스턴스용 Amazon EBS 지표

AWS/EC2 네임스페이스에는 베어 메탈 인스턴스가 아닌 Nitro 기반 인스턴스에 연결된 볼륨에 대한 추가 Amazon EBS 지표가 포함됩니다.

지표	설명	단위	의미 있는 통계
EBSReadOps	지정된 기간 내에 인스턴스에 연결된 모든 Amazon EBS 볼륨에서 완료된 읽기 작업입니다. 해당 기간의 초당 평균 읽기 I/O 작업 수(읽기 IOPS)를 계산하려면 해당 기간의 총 작업 수를 해당 기간의 초 수로 나누세요. 기본(5분) 모니터링을 사용하는 경우, 이 숫자를 300으로 나누어 읽기 IOPS를 계산할 수 있습니다. 세부(1분) 모니터링의 경우 60으로 나눕니다. CloudWatch 지표 수학 함수 DIFF_TIME 을 사용하여 초당 작업 수를 찾을 수도 있습니다. 예를 들어 CloudWatch에서 EBSReadOps 을 m1로 그래프로 표시한 경우 지표 수학 공식 m1/(DIFF_TIME(m1)) 은 지표(작업/초)를 반환합니다. DIFF_TIME 및 기타 지표 수학 함수에 대한 자세한 내용은 Amazon CloudWatch User Guide의 Use metric math를 참조하세요.	개수	• 합계 • 평균 • 최소 • Maximum

지표	설명	단위	의미 있는 통계
EBSWriteOps	지정된 기간 내에 인스턴스에 연결된 모든 EBS 볼륨으로의 완료된 쓰기 작업입니다. 해당 기간의 초당 평균 쓰기 I/O 작업 수(쓰기 IOPS)를 계산하려면 해당 기간의 총 작업 수를 해당 기간의 초 수로 나누세요. 기본(5분) 모니터링을 사용하는 경우, 이 숫자를 300으로 나누어 쓰기 IOPS를 계산할 수 있습니다. 세부(1분) 모니터링의 경우 60으로 나눕니다. CloudWatch 지표 수학 함수 DIFF_TIME 을 사용하여 초당 작업 수를 찾을 수도 있습니다. 예를 들어 CloudWatch에서 EBSWriteOps 을 m1로 그래프로 표시한 경우 지표 수학 공식 m1/(DIFF_TIME(m1)) 은 지표(작업/초)를 반환합니다. DIFF_TIME 및 기타 지표 수학 함수에 대한 자세한 내용은 Amazon CloudWatch User Guide의 Use metric math를 참조하세요.	개수	• 합계 • 평균 • 최소 • Maximum
EBSReadBytes	지정된 기간 내에 인스턴스에 연결된 모든 EBS 볼륨에서의 바이트 읽기 작업입니다. 보고된 숫자는 해당 기간에 읽은 바이트 수입니다. 기본(5분) 모니터링을 사용하는 경우, 이 숫자를 300으로 나누어 읽기 바이트/초를 찾을 수 있습니다. 세부(1분) 모니터링의 경우 60으로 나눕니다. CloudWatch 지표 수학 함수 DIFF_TIME 을 사용하여 초당 바이트 수를 찾을 수도 있습니다. 예를 들어 CloudWatch에서 EBSReadBytes 을 m1로 그래프로 표시한 경우 지표 수학 공식 m1/(DIFF_TIME(m1)) 은 지표(바이트/초)를 반환합니다. DIFF_TIME 및 기타 지표 수학 함수에 대한 자세한 내용은 Amazon CloudWatch User Guide의 Use metric math를 참조하세요.	바이트	• Sum • 평균 • 최소 • Maximum

지표	설명	단위	의미 있는 통계
EBSWriteBytes	지정된 기간 내에 인스턴스에 연결된 모든 EBS 볼륨으로의 바이트 쓰기 작업입니다. 보고된 숫자는 해당 기간에 써진 바이트 수입니다. 기본(5분) 모니터링을 사용하는 경우, 이 숫자를 300으로 나누어 쓰기 바이트/초를 찾을 수 있습니다. 세부(1분) 모니터링의 경우 60으로 나눕니다. CloudWatch 지표 수학 함수 DIFF_TIME 을 사용하여 초당 바이트 수를 찾을 수도 있습니다. 예를 들어 CloudWatch에서 EBSWriteBytes 을 m1로 그래프로 표시한 경우 지표 수학 공식 m1/(DIFF_TIME(m1)) 은 지표(바이트/초)를 반환합니다. DIFF_TIME 및 기타 지표 수학 함수에 대한 자세한 내용은 Amazon CloudWatch User Guide의 Use metric math를 참조하세요.	바이트	• Sum • 평균 • 최소 • Maximum
EBSIOBalance%	버스트 버킷에 남아 있는 I/O 크레딧의 비율에 대한 정보를 제공합니다. 기본 모니터링에서만 이 지표를 사용할 수 있습니다. 이 지표는 24시간에 한 번 이상 30분 동안만 최대 성능을 발휘하는 일부 *.4xlarge 인스턴스 크기 이하에서만 사용할 수 있습니다. Sum 통계는 이 지표에 적용할 수 없습니다.	%	• 최소 • Maximum

지표	설명	단위	의미 있는 통계
EBSByteBalance%	버스트 버킷에 남아 있는 처리량 크레딧의 비율에 대한 정보를 제공합니다. 기본 모니터링에서만 이 지표를 사용할 수 있습니다. 이 지표는 24시간에 한 번 이상 30분 동안만 최대 성능을 발휘하는 일부 *.4xlarge 인스턴스 크기 이하에서만 사용할 수 있습니다. Sum 통계는 이 지표에 적용할 수 없습니다.	%	• 최소 • Maximum

EBS 볼륨에 대해 제공되는 지표에 대한 자세한 내용은 Amazon EBS 사용 설명서의 Metrics for Amazon EBS volumes를 참조하세요. EC2 플릿 및 스팟 플릿에 제공되는 지표에 대한 자세한 내용은 CloudWatch를 사용하여 EC2 플릿 또는 스팟 플릿 모니터링의 내용을 참조하세요.

상태 확인 지표

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 running 상태로 시작되는 몇 분 내). EC2 상태 확인에 대한 자세한 내용은 Amazon EC2 인스턴스 상태 확인 단원을 참조하세요.

AWS/EC2 네임스페이스에는 다음과 같은 상태 확인 지표가 포함되어 있습니다.

지표	설명	단위	의미 있는 통계
StatusCheckFailed	인스턴스가 마지막으로 모든 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.	개수	• 평균 • 최소 • Maximum

지표	설명	단위	의미 있는 통계
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.	개수	• 평균 • 최소 • Maximum
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.	개수	• 평균 • 최소 • Maximum
StatusCheckFailed_AttachedEBS	인스턴스가 마지막으로 연결된 EBS 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.	개수	• 평균 • 최소 • Maximum

AWS/EBS 네임스페이스에는 다음과 같은 상태 확인 지표가 포함되어 있습니다.

지표	설명	단위	의미 있는 통계
VolumeStalledIOCheck	참고: Nitro 인스턴스에만 해당됩니다. Amazon ECS 및 AWS Fargate 작업에 연결된 볼륨에 대해서는 게시되지 않았습니다.	없음	• 평균 • 최소 • Maximum

지표	설명	단위	의미 있는 통계
	볼륨이 마지막 1분 동안 멈춘 IO 검사를 통과했는지 아니면 실패했는지 보고합니다. 이 지표는 0(통과) 또는 1(실패)이 될 수 있습니다.		

트래픽 미러링 지표

AWS/EC2 네임스페이스에는 미러링된 트래픽에 대한 지표가 포함됩니다. 자세한 내용은 Amazon VPC Traffic Mirroring 가이드의 Amazon CloudWatch를 사용한 미러링된 트래픽 모니터링을 참조하세요.

Auto Scaling 그룹 지표

AWS/AutoScaling 네임스페이스에는 Auto Scaling 그룹에 대한 지표가 포함됩니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서에서 Auto Scaling 그룹 및 인스턴스에 대한 CloudWatch 지표 모니터링을 참조하세요.

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.
ImageId	이 차원은 사용자가 이 Amazon EC2 Amazon Machine Image(AMI)를 실행하는 모든 인스턴스에 대해 요청하는 데이터를 필터링합니다. 세부 모니터링이 설정된 인스턴스에 사용할 수 있습니다.

차원	설명
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 Service Quotas에 해당합니다. 사용량이 서비스 할당량에 가까워지면 경고하는 경보를 구성할 수 있습니다. CloudWatch Service Quotas 통합에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서의 AWS 사용량 지표를 참조하세요.

Amazon EC2는 AWS/Usage 네임스페이스에 다음 지표를 게시합니다.

측정치	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 MAXIMUM입니다.

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 EC2입니다.
Type	보고되는 엔터티의 유형입니다. 현재 Amazon EC2 사용량 지표에 대한 유일한 유효 값은 Resource입니다.
Resource	실행 중인 리소스의 유형입니다. 현대 Amazon EC2 사용량 지표에 대한 유일한 유효 값은 실행 중인 인스턴스에 대한 정보를 반환하는 vCPU입니다.
Class	추적 중인 리소스의 클래스. vCPU 차원의 값이 Resource인 Amazon EC2 사용량 지표의 경우 유효한 값은 Standard/OnDemand , F/OnDemand , G/OnDemand , Inf/OnDemand , P/OnDemand 및 X/OnDemand 입니다. 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.

Amazon EC2 콘솔을 사용하여 CloudWatch 에이전트를 설치 및 구성하고 지표 추가

Amazon EC2 콘솔을 사용하여 CloudWatch 에이전트를 설치 및 구성하는 기능은 Amazon EC2의 베타 버전에서 제공되며, 변경 가능합니다.

기본적으로 Amazon CloudWatch는 Amazon EC2 인스턴스를 모니터링하기 위한 CPUUtilization 및 NetworkIn 등의 기본 지표를 제공합니다. 추가 지표를 수집하려면 EC2 인스턴스에 CloudWatch 에이전트를 설치한 다음 선택한 지표를 내보내도록 에이전트를 구성합니다. 모든 EC2 인스턴스에 CloudWatch 에이전트를 수동으로 설치하고 구성하는 대신 Amazon EC2 콘솔을 사용하여 이 작업을 대신 수행할 수 있습니다.

이 주제에서는 Amazon EC2 콘솔을 사용하여 인스턴스에 CloudWatch 에이전트를 설치하고 선택한 지표를 내보내도록 에이전트를 구성하는 방법을 설명합니다.

이 프로세스의 수동 단계는 Amazon CloudWatch 사용 설명서의 AWS Systems Manager를 사용하여 CloudWatch 에이전트 설치를 참조하세요. CloudWatch 에이전트에 대한 자세한 내용은 CloudWatch 에이전트를 사용하여 지표, 로그 및 추적 수집을 참조하세요.

주제

- 사전 조건
- 작동 방법
- 비용
- CloudWatch 에이전트 설치 및 구성

사전 조건

Amazon EC2를 사용하여 CloudWatch 에이전트를 설치 및 구성하려면 이 섹션에 설명된 사용자 및 인스턴스 사전 조건을 충족해야 합니다.

사용자 사전 조건

이 기능을 사용하려면 IAM 콘솔 사용자 또는 역할에 Amazon EC2를 사용하는 데 필요한 권한 및 다음과 같은 IAM 권한이 있어야 합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameter",
                "ssm:PutParameter"
            ],
            "Resource": "arn:aws:ssm:*:*:parameter/EC2-Custom-Metrics-*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand",
                "ssm:ListCommandInvocations",
                "ssm:DescribeInstanceInformation"
```

```
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetInstanceProfile",
                "iam:SimulatePrincipalPolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

인스턴스 사전 조건

- 인스턴스 상태: `running`
- 지원되는 운영 체제: Linux
- AWS Systems Manager Agent(SSM Agent): 설치되었습니다. SSM Agent에 대한 두 가지 참고 사항:
  - SSM Agent는 AWS 및 신뢰할 수 있는 타사에서 제공하는 일부 Amazon Machine Image(AMI)에 사전 설치되어 있습니다. 지원되는 AMI와 SSM Agent 설치 지침에 대한 자세한 내용은 AWS Systems Manager 사용 설명서의 SSM Agent가 사전 설치된 Amazon Machine Image(AMI)를 참조하세요.
  - SSM Agent에서 문제가 발생하는 경우 AWS Systems Manager 사용 설명서의 SSM Agent 문제 해결을 참조하세요.
- 인스턴스에 대한 IAM 권한: 인스턴스에 연결된 IAM 역할에 다음과 같은 AWS 관리형 정책을 추가해야 합니다.
  - AmazonSSMManagedInstanceCore – 인스턴스를 통해 Systems Manager를 사용하여 CloudWatch 에이전트를 설치 및 구성합니다.
  - CloudWatchAgentServerPolicy – 인스턴스를 통해 CloudWatch 에이전트를 사용하여 CloudWatch에 데이터를 씁니다.

  인스턴스에 IAM 권한을 추가하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 인스턴스 프로파일 사용을 참조하세요.

작동 방법

Amazon EC2 콘솔을 사용하여 CloudWatch 에이전트를 설치하고 구성하려면 먼저 IAM 사용자 또는 역할, 그리고 지표를 추가하려는 인스턴스가 특정 사전 조건을 충족하는지 확인해야 합니다. 그런 다음 Amazon EC2 콘솔을 사용하여 선택한 인스턴스에 CloudWatch 에이전트를 설치하고 구성할 수 있습니다.

먼저 사전 조건을 충족해야 합니다.

- 필수 IAM 권한 필요 - 시작하기 전에 콘솔 사용자 또는 역할에 이 기능을 사용하는 데 필요한 IAM 권한이 있는지 확인합니다.
- 인스턴스 - 이 기능을 사용하려면 EC2 인스턴스가 Linux 인스턴스이고, SSM Agent가 설치되어 있으며, 필요한 IAM 권한이 있고, 실행 중이어야 합니다.

그러면 이 기능을 사용할 수 있습니다.

1. 인스턴스 선택 - Amazon EC2 콘솔에서 CloudWatch 에이전트를 설치하고 구성할 인스턴스를 선택합니다. 그런 다음 CloudWatch 에이전트 구성을 선택하여 프로세스를 시작합니다.
2. SSM Agent 검증 - Amazon EC2는 SSM Agent가 각 인스턴스에 설치되고 시작되었는지 확인합니다. 이 검사에 실패한 모든 인스턴스는 프로세스에서 제외됩니다. SSM Agent는 이 프로세스 중에 인스턴스에서 작업을 수행하는 데 사용됩니다.
3. IAM 권한 검증 - Amazon EC2는 각 인스턴스에 이 프로세스에 필요한 IAM 권한이 있는지 확인합니다. 이 검사에 실패한 모든 인스턴스는 프로세스에서 제외됩니다. IAM 권한을 통해 CloudWatch 에이전트는 인스턴스에서 지표를 수집하고 AWS Systems Manager와 통합하여 SSM Agent를 사용할 수 있습니다.
4. CloudWatch 에이전트 검증 - Amazon EC2는 CloudWatch 에이전트가 각 인스턴스에 설치되어 실행 중인지 확인합니다. 이 검사에 실패한 인스턴스가 있는 경우 Amazon EC2에서 CloudWatch 에이전트를 설치하고 시작하도록 제안합니다. CloudWatch 에이전트는 이 프로세스가 완료되면 각 인스턴스에서 선택한 지표를 수집합니다.
5. 지표 구성 선택 - CloudWatch 에이전트가 인스턴스에서 내보낼 지표를 선택합니다. 선택한 후에는 Amazon EC2가 Parameter Store에 구성 파일을 저장하며, 이 파일은 프로세스가 완료될 때까지 여기에 보관됩니다. Amazon EC2는 프로세스가 중단되지 않는 한 Parameter Store에서 구성 파일을 삭제합니다. 지표를 선택하지 않았지만 이전에 인스턴스에 지표를 추가한 경우 이 프로세스가 완료되면 인스턴스에서 해당 지표가 제거됩니다.

6. CloudWatch 에이전트 구성 업데이트 - Amazon EC2는 지표 구성을 CloudWatch 에이전트에 보냅니다. 이 프로세스의 마지막 단계입니다. 성공하면 인스턴스가 선택한 지표에 대한 데이터를 내보낼 수 있으며 Amazon EC2는 Parameter Store에서 구성 파일을 삭제합니다.

비용

이 프로세스 중에 추가하는 지표는 사용자 지정 지표로 청구됩니다. CloudWatch 지표 요금에 대한 자세한 내용은 Amazon CloudWatch 요금을 참조하세요.

CloudWatch 에이전트 설치 및 구성

Amazon EC2 콘솔을 사용하여 CloudWatch 에이전트를 설치 및 구성하고 지표를 추가할 수 있습니다.

> **Note**
>
> 이 절차를 수행할 때마다 기존 CloudWatch 에이전트 구성을 덮어씁니다. 이전에 선택한 지표를 선택하지 않으면 해당 지표가 인스턴스에서 제거됩니다.

Amazon EC2 콘솔을 사용하여 CloudWatch 에이전트 설치 및 구성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. CloudWatch 에이전트를 설치하고 구성할 인스턴스를 선택합니다.
4. 는 업데이트되고 완전히 패치된 Windows AMI를 Microsoft의 패치 화요일(매달 둘째 주 화요일)을 기준으로 5영업일 내에 제공합니다.

   > **Tip**
   >
   > 이 기능은 일부 AWS 리전에서는 아직 사용할 수 없습니다. CloudWatch 에이전트 구성을 사용할 수 없는 경우 다른 리전을 사용해 보세요.

5. 프로세스의 각 단계에서 콘솔 텍스트를 읽은 후 다음을 선택합니다.
6. 프로세스를 완료하려면 마지막 단계에서 완료를 선택합니다.

인스턴스에 대한 CloudWatch 지표 통계

인스턴스에 대한 CloudWatch 측정치 통계를 볼 수 있습니다. 통계는 지정한 기간에 걸친 지표 데이터 집계입니다. CloudWatch는 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.
`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).

내용

- 특정 인스턴스에 대한 통계 가져오기
- 여러 인스턴스의 통계 집계
- Auto Scaling 그룹별 통계 집계
- AMI별 집계 통계

특정 인스턴스에 대한 통계 가져오기

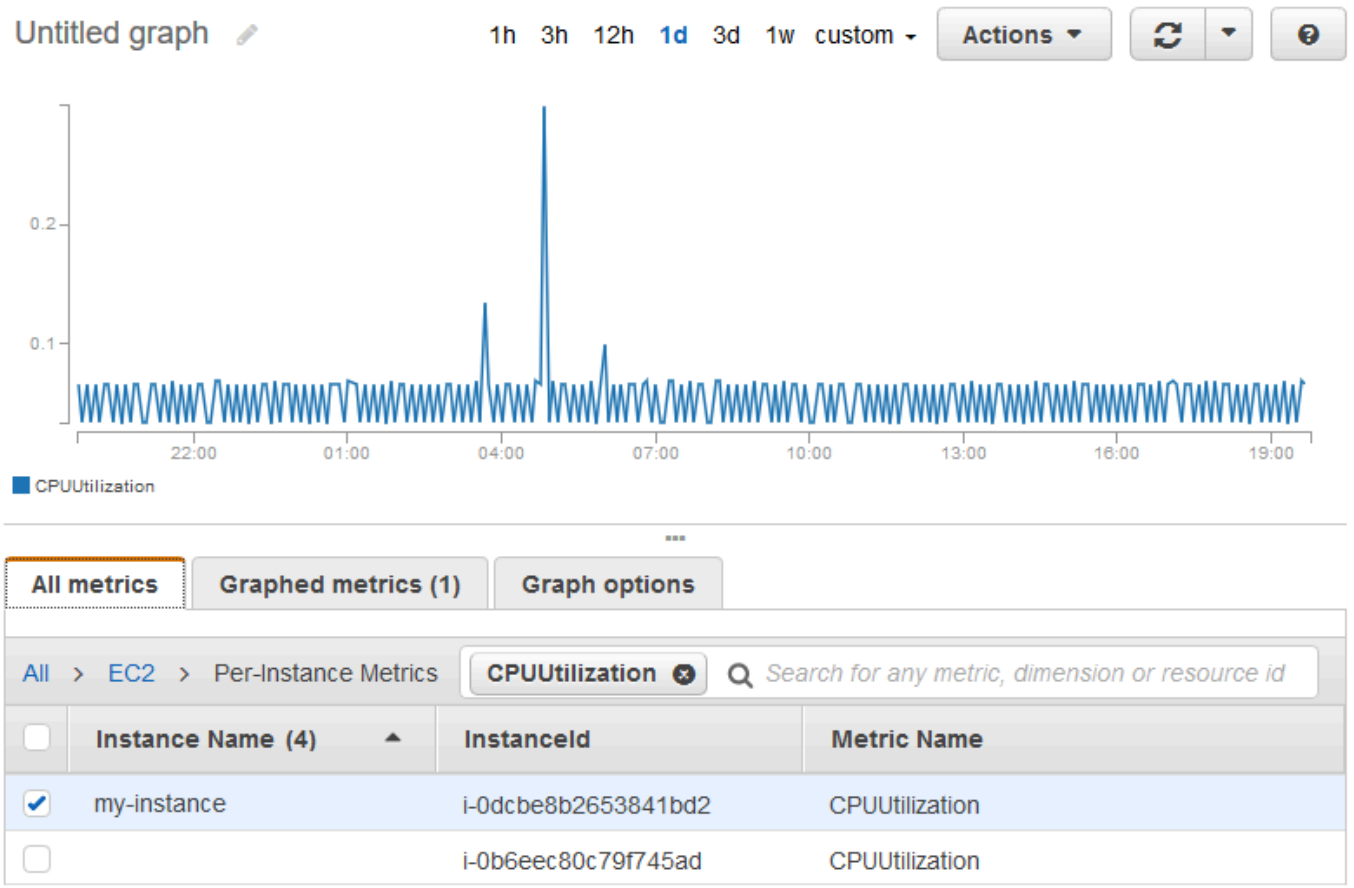
AWS Management Console 또는 AWS CLI를 사용하여 특정 인스턴스에 대한 통계를 구할 수 있습니다. 다음 예제는 AWS Management Console 또는 AWS CLI 명령을 사용하여 특정 EC2 인스턴스의 최대 CPU 사용률을 확인하는 방법을 보여 줍니다.

요구 사항

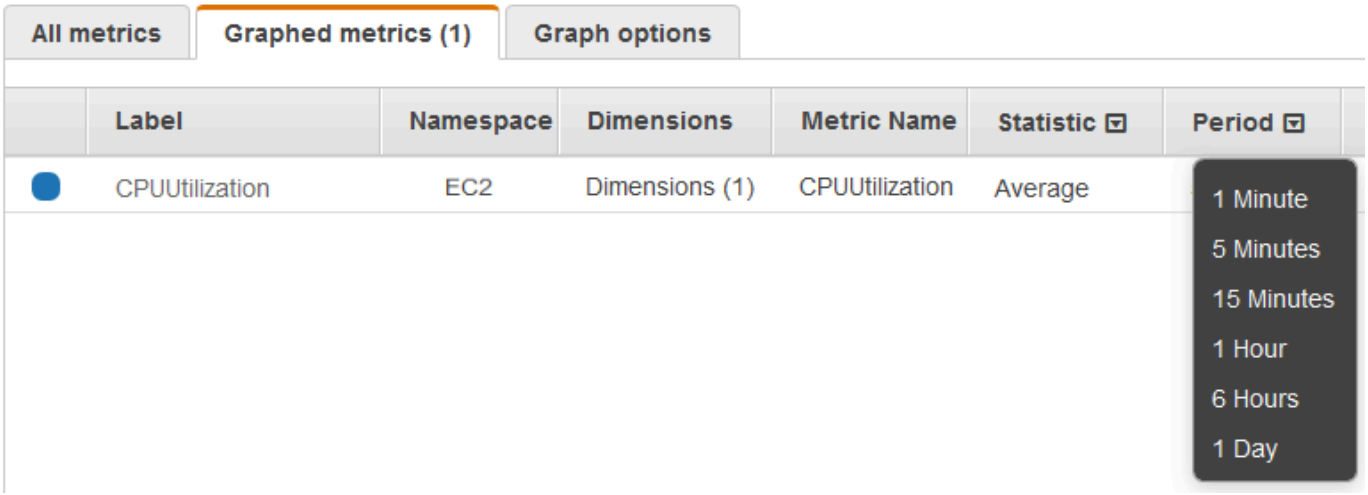
- 인스턴스의 ID가 필요합니다. 인스턴스 ID는 AWS Management Console이나 describe-instances 명령을 사용하여 확인할 수 있습니다.
- 기본적으로 기본 모니터링이 사용되지만 세부 모니터링을 사용하도록 설정할 수 있습니다. 자세한 내용은 EC2 인스턴스에 대한 세부 모니터링 관리 섹션을 참조하세요.

특정 인스턴스에 대한 CPU 사용률을 표시하려면(콘솔)

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [지표(Metrics)]를 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 2022-10-18T23:18:00 --end-time 2022-10-19T23:18:00
```

다음은 예제 출력입니다. 각 값은 단일 EC2 인스턴스에 대한 최대 CPU 사용률을 나타냅니다.

```
{
    "Datapoints": [
        {
            "Timestamp": "2022-10-19T00:18:00Z",
            "Maximum": 0.33000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2022-10-19T03:18:00Z",
            "Maximum": 99.670000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2022-10-19T07:18:00Z",
            "Maximum": 0.34000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2022-10-19T12:18:00Z",
            "Maximum": 0.34000000000000002,
            "Unit": "Percent"
        }
    ],
    "Label": "CPUUtilization"
}
```

여러 인스턴스의 통계 집계

세부 모니터링이 활성화된 인스턴스에 대해서만 통계를 집계할 수 있습니다. 기본 모니터링을 사용하는 인스턴스는 집계에 포함되지 않습니다. 인스턴스 간에 집계된 통계를 얻으려면 1분 기간의 데이터를 제공하는 세부 모니터링(추가 비용 발생)을 활성화해야 합니다.

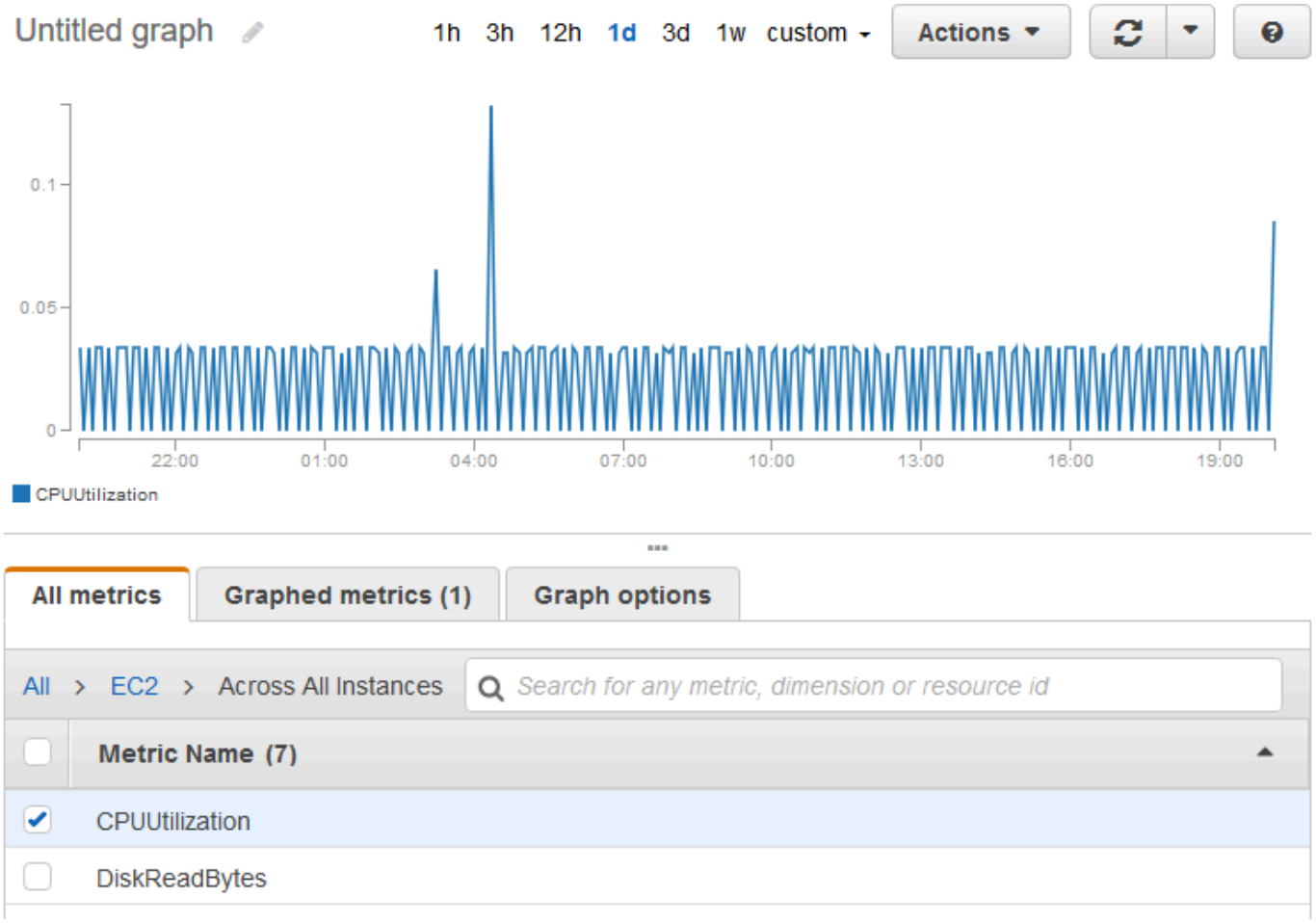
Amazon CloudWatch는 AWS 리전 전체의 데이터는 집계할 수 없습니다. 리전마다 지표가 완전히 분리되어 있습니다.

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 [지표(Metrics)]를 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 2022-10-11T23:18:00 \
    --end-time 2022-10-12T23:18:00
```

다음은 예제 출력입니다.

```
{
    "Datapoints": [
        {
            "SampleCount": 238.0,
            "Timestamp": "2022-10-12T07:18:00Z",
            "Average": 0.038235294117647062,
            "Unit": "Percent"
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2022-10-12T09:18:00Z",
            "Average": 0.16670833333333332,
            "Unit": "Percent"
        },
        {
            "SampleCount": 238.0,
            "Timestamp": "2022-10-11T23:18:00Z",
            "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 AWS 리전 전체의 데이터는 집계할 수 없습니다. 리전마다 지표가 완전히 분리되어 있습니다.

이 예제는 하나의 Auto Scaling 그룹에 대해 디스크에 기록되는 총 바이트 수를 확인하는 방법을 보여줍니다. 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 [지표(Metrics)]를 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 표시하려면(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 2022-10-16T23:18:00 --end-time 2022-10-18T23:18:00
```

다음은 예제 출력입니다.

```
{
    "Datapoints": [
        {
            "SampleCount": 18.0,
            "Timestamp": "2022-10-19T21:36:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        },
        {
            "SampleCount": 5.0,
            "Timestamp": "2022-10-19T21:42:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        }
    ],
    "Label": "DiskWriteBytes"
}
```

AMI별 집계 통계

세부 모니터링이 활성화된 인스턴스에 대해 AMI별로 통계를 집계할 수 있습니다. 기본 모니터링을 사용하는 인스턴스는 집계에 포함되지 않습니다. 인스턴스 간에 집계된 통계를 얻으려면 1분 기간의 데이터를 제공하는 세부 모니터링(추가 비용 발생)을 활성화해야 합니다.

Amazon CloudWatch는 AWS 리전 전체의 데이터는 집계할 수 없습니다. 리전마다 지표가 완전히 분리되어 있습니다.

이 예제는 특정 Amazon Machine Image(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 [지표(Metrics)]를 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 2022-10-10T00:00:00 --end-time 2022-10-11T00:00:00
```

다음은 예제 출력입니다. 각 값은 지정한 AMI를 실행 중인 EC2 인스턴스의 평균 CPU 사용률을 나타냅니다.

```
{
    "Datapoints": [
        {
            "Timestamp": "2022-10-10T07:00:00Z",
            "Average": 0.041000000000000009,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2022-10-10T14:00:00Z",
            "Average": 0.079579831932773085,
```

```
            "Unit": "Percent"
        },
        {
            "Timestamp": "2022-10-10T06:00:00Z",
            "Average": 0.036000000000000011,
            "Unit": "Percent"
        }
    ],
    "Label": "CPUUtilization"
}
```


인스턴스에 대한 모니터링 그래프 보기

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 사용 가능한 CloudWatch 지표 단원을 참조하세요.

CloudWatch 콘솔을 사용한 측정치 그래프

CloudWatch 콘솔을 사용하여 Amazon EC2 및 기타 AWS 서비스에서 생성한 지표 데이터의 그래프를 생성할 수도 있습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서에서 지표 그래프를 참조하세요.

인스턴스에 대해 CloudWatch 경보 만들기

인스턴스 중 하나에 대한 CloudWatch 지표를 모니터링하는 CloudWatch 경보를 생성할 수 있습니다. 지표가 지정된 임곗값에 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 참조하세요.

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 선택하고 [작업(Actions)], [모니터링 및 문제 해결(Monitor and troubleshoot)], [CloudWatch 경보 관리(Manage CloudWatch alarms)]를 선택합니다.
4. [CloudWatch 경보 관리(Manage CloudWatch alarms)] 세부 정보 페이지의 [경보 추가 또는 편집(Add or edit alarm)]에서 [경보 생성(Create an alarm)]을 선택합니다.
5. 경보 알림의 경우 Amazon Simple Notification Service(SNS) 알림을 구성할지 여부를 선택합니다. 기존 Amazon SNS 주제를 입력하거나 이름을 입력하여 새 주제를 생성합니다.
6. 경보 작업의 경우 경보가 트리거될 때 수행할 작업을 지정할지 여부를 선택합니다. 목록에서 작업을 선택합니다.
7. 경보 임곗값에 대해 경보에 대한 지표와 기준을 선택합니다. 예를 들어, 5분 동안 CPU 사용률이 80%에 도달할 때 트리거되는 경보를 생성하려면 다음을 수행합니다.
   a. 그룹 샘플링 기준(평균) 및 샘플링할 데이터 유형(CPU 사용률)에 대한 기본 설정을 그대로 유지합니다.
   b. 경보 시기에 대해 **>=**를 선택하고 백분율에 대해 **0.80**을 입력합니다.
   c. 연속 기간에 **1**을 입력하고 기간에 5분을 선택합니다.
8. (선택 사항) 샘플 지표 데이터의 경우 대시보드에 추가를 선택합니다.
9. Create를 선택합니다.

Amazon EC2 콘솔 또는 CloudWatch 콘솔에서 CloudWatch 경보 설정을 편집할 수 있습니다. 경보를 삭제하려는 경우 CloudWatch 콘솔에서 삭제할 수 있습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 경보 편집 또는 삭제를 참조하세요.

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.

> **Note**
>
> Amazon CloudWatch 경보 결제 및 요금 정보는 Amazon CloudWatch 사용 설명서의 CloudWatch 결제 및 비용을 참조하세요.

`AWSServiceRoleForCloudWatchEvents`는 AWS 서비스 연결 역할을 통해 사용자를 대신하여 경보 작업을 수행할 수 있습니다. AWS Management Console, AWS CLI 또는 IAM API에서 처음으로 경보를 생성하면 CloudWatch가 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 시작할 수 있고 동일한 인스턴스 ID 및 루트 볼륨을 유지할 수 있다는 점입니다. 그러나 종료된 인스턴스를 시작할 수는 없습니다. 대신, 새 인스턴스를 시작해야 합니다. 인스턴스가 중지되거나 종료되면 인스턴스 스토어 볼륨의 데이터가 손실됩니다.

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.

> **Important**
>
> 누락된 지표 데이터 요소가 있는 경우 상태 검사 경보가 일시적으로 `INSUFFICIENT_DATA` 상태로 전환될 수 있습니다. 드물기는 하지만, 지표 보고 시스템이 중단되면 인스턴스가 정상인 경우에도 이 문제가 발생할 수 있습니다. 특히 인스턴스를 중지, 종료, 재부팅 또는 복구하도록 경보를 구성할 때는 `INSUFFICIENT_DATA` 상태를 경보 위반 대신 데이터 누락으로 취급하는 것이 좋습니다.

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 참조하세요.

권한

EC2 경보 작업을 수행하는 경보를 생성하거나 수정하려면 `iam:CreateServiceLinkedRole`이 있어야 합니다. 서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하는 것으로 가정하는 IAM 역할입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 정보는 IAM 사용 설명서의 AWS 서비스에 대한 권한을 위임할 역할 생성을 참조하세요.

내용

- Amazon CloudWatch 경보에 중지 작업 추가
- Amazon CloudWatch 경보에 종료 작업 추가
- Amazon CloudWatch 경보에 재부팅 작업 추가
- Amazon CloudWatch 경보에 복구 작업 추가
- Amazon CloudWatch 경보 작업 시나리오

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. 인스턴스가 종료되거나 중지되면 인스턴스 스토어 볼륨의 데이터가 손실됩니다.

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 선택하고 [작업(Actions)], [모니터링 및 문제 해결(Monitor and troubleshoot)], [CloudWatch 경보 관리(Manage CloudWatch alarms)]를 선택합니다.

또는 [경보 상태(Alarm status)] 열에서 더하기 기호
(+)
를 선택해도 됩니다.

4. [CloudWatch 경보 관리(Manage CloudWatch alarms)] 페이지에서 다음을 수행합니다.

   a. [경보 생성(Create an alarm)]을 선택합니다.

   b. 경보가 트리거될 때 이메일을 받으려면 [경보 알림(Alarm notification)]에 대해 기존 Amazon SNS 주제를 선택합니다. 먼저 Amazon SNS 콘솔을 사용하여 Amazon SNS 주제를 생성해야 합니다. 자세한 내용은 Amazon Simple Notification Service 개발자 안내서에서 A2P(Application-to-Person) 메시징에 Amazon SNS 사용을 참조하세요.

   c. [경보 작업(Alarm action)]을 켜고 [중지(Stop)]를 선택합니다.

   d. [샘플 그룹화 기준(Group samples by)]과 [샘플링할 데이터 유형(Type of data to sample)]에 대해 통계 및 지표를 선택합니다. 이 예에서는 [평균(Average)] 및 [CPU 사용률(CPU utilization)]을 선택합니다.

   e. [경보 시기(Alarm When)] 및 [백분율(Percent)]에서 지표 임계값을 지정합니다. 이 예에서는 >= 및 10%를 지정합니다.

   f. [연속 기간(Consecutive period)]과 [기간(Period)]에 대해 경보의 평가 기간을 지정합니다. 이 예에서는 [1] 연속 기간([5분(5 Minutes)])을 지정합니다.

   g. Amazon CloudWatch에서 자동으로 경보 이름이 생성됩니다. 이름을 변경하려면 [경보 이름(Alarm name)]에 새 이름을 입력합니다. 경보 이름은 ASCII 문자만 포함해야 합니다.

      > **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.

   h. 생성(Create)을 선택합니다.

Amazon CloudWatch 경보에 종료 작업 추가

인스턴스에 대해 종료 보호가 비활성화되어 있는 경우에 한해서 특정 임계값에 도달한 경우 EC2 인스턴스를 자동으로 종료하는 경보를 만들 수 있습니다. 예를 들어 인스턴스의 작업 완료 후 해당 인스턴스가 다시 필요 없는 경우 인스턴스를 종료하려고 할 수 있습니다. 나중에 인스턴스를 사용하려는 경우에는 종료하지 말고 중지해야 합니다. 인스턴스가 종료될 때 인스턴스 스토어 볼륨의 데이터가 손실됩

니다. 인스턴스 종료 방지 기능의 활성화/비활성화에 대한 자세한 내용은 인스턴스 종료 방지 변경 섹션을 참조하세요.

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 선택하고 [작업(Actions)], [모니터링 및 문제 해결(Monitor and troubleshoot)], [CloudWatch 경보 관리(Manage CloudWatch alarms)]를 선택합니다.

   또는 [경보 상태(Alarm status)] 열에서 더하기 기호
   (+)
   를 선택해도 됩니다.
4. [CloudWatch 경보 관리(Manage CloudWatch alarms)] 페이지에서 다음을 수행합니다.

   a. [경보 생성(Create an alarm)]을 선택합니다.

   b. 경보가 트리거될 때 이메일을 받으려면 [경보 알림(Alarm notification)]에 대해 기존 Amazon SNS 주제를 선택합니다. 먼저 Amazon SNS 콘솔을 사용하여 Amazon SNS 주제를 생성해야 합니다. 자세한 내용은 Amazon Simple Notification Service 개발자 안내서에서 A2P(Application-to-Person) 메시징에 Amazon SNS 사용을 참조하세요.

   c. [경보 작업(Alarm action)]을 켜고 [종료(Terminate)]를 선택합니다.

   d. [샘플 그룹화 기준(Group samples by)]과 [샘플링할 데이터 유형(Type of data to sample)]에 대해 통계 및 지표를 선택합니다. 이 예에서는 [평균(Average)] 및 [CPU 사용률(CPU utilization)]을 선택합니다.

   e. [경보 시기(Alarm When)] 및 [백분율(Percent)]에서 지표 임계값을 지정합니다. 이 예에서는 => 및 10%를 지정합니다.

   f. [연속 기간(Consecutive period)]과 [기간(Period)]에 대해 경보의 평가 기간을 지정합니다. 이 예에서는 [24] 연속 기간([1시간(1 Hour)])을 지정합니다.

   g. Amazon CloudWatch에서 자동으로 경보 이름이 생성됩니다. 이름을 변경하려면 [경보 이름(Alarm name)]에 새 이름을 입력합니다. 경보 이름은 ASCII 문자만 포함해야 합니다.

> **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.

h. 생성(Create)을 선택합니다.

Amazon CloudWatch 경보에 재부팅 작업 추가

Amazon EC2 인스턴스를 모니터링하고 인스턴스를 자동으로 재부팅하는 Amazon CloudWatch 경보를 만들 수 있습니다. 재부팅 경보 작업은 인스턴스 상태 확인 오류(복구 경보 작업은 시스템 상태 확인 오류에 적합)에 권장됩니다. 인스턴스 재부팅은 운영 체제 재부팅과 같습니다. 대부분의 경우 인스턴스를 재부팅하는 데는 몇 분 밖에 걸리지 않습니다. 인스턴스를 재부팅하는 경우 동일한 물리적 호스트에 남아 있으므로 퍼블릭 DNS 이름, 프라이빗 IP 주소 및 인스턴스 스토어 볼륨의 모든 데이터가 유지됩니다.

인스턴스를 재부팅해도 인스턴스를 중지했다가 다시 시작할 때와는 달리 인스턴스 청구 기간(최소 1분 요금 포함)이 새로 시작되지 않습니다. 인스턴스가 재부팅될 때 인스턴스 스토어 볼륨의 데이터가 유지됩니다. 재부팅 후 인스턴스 스토어 볼륨을 파일 시스템에 다시 탑재해야 합니다. 자세한 내용은 Amazon EC2 인스턴스 재부팅 단원을 참조하십시오.

> **Important**
>
> 재부팅과 복원 작업 간에 경합 상태가 발생하지 않도록 하려면 재부팅 경보와 복원 경보에 동일한 평가 기간 값을 설정하지 마세요. 재부팅 경보를 각각 1분의 평가 기간 3회로 설정하는 것이 좋습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 경보 평가를 참조하세요.

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 선택하고 [작업(Actions)], [모니터링 및 문제 해결(Monitor and troubleshoot)], [CloudWatch 경보 관리(Manage CloudWatch alarms)]를 선택합니다.

또는 [경보 상태(Alarm status)] 열에서 더하기 기호
(+)
를 선택해도 됩니다.

4. [CloudWatch 경보 관리(Manage CloudWatch alarms)] 페이지에서 다음을 수행합니다.

   a. [경보 생성(Create an alarm)]을 선택합니다.

   b. 경보가 트리거될 때 이메일을 받으려면 [경보 알림(Alarm notification)]에 대해 기존 Amazon SNS 주제를 선택합니다. 먼저 Amazon SNS 콘솔을 사용하여 Amazon SNS 주제를 생성해야 합니다. 자세한 내용은 Amazon Simple Notification Service 개발자 안내서에서 A2P(Application-to-Person) 메시징에 Amazon SNS 사용을 참조하세요.

   c. [경보 작업(Alarm action)]을 켜고 [재부팅(Reboot)]을 선택합니다.

   d. [샘플 그룹화 기준(Group samples by)]과 [샘플링할 데이터 유형(Type of data to sample)]에 대해 통계 및 지표를 선택합니다. 이 예에서는 [평균(Average)] 및 [상태 확인 실패: 인스턴스(Status check failed: instance)]를 선택합니다.

   e. [연속 기간(Consecutive period)]과 [기간(Period)]에 대해 경보의 평가 기간을 지정합니다. 이 예에서는 1분의 3 연속 기간을 입력합니다. 1분이 비활성화된 경우 세부 모니터링을 활성화해야 합니다. 그렇지 않으면 5분을 대신 선택할 수 있습니다.

   f. Amazon CloudWatch에서 자동으로 경보 이름이 생성됩니다. 이름을 변경하려면 [경보 이름(Alarm name)]에 새 이름을 입력합니다. 경보 이름은 ASCII 문자만 포함해야 합니다.

   g. 생성(Create)을 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 있습니다. 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.

> **Note**
>
> 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 특정 특성을 충족하는 인스턴스에서만 지원됩니다. 자세한 내용은 자동 인스턴스 복구 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 마세요. 복구 경보는 각각 1분의 평가 기간 2회로 설정하는 것이 좋습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 경보 평가를 참조하세요.

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 선택하고 [작업(Actions)], [모니터링 및 문제 해결(Monitor and troubleshoot)], [CloudWatch 경보 관리(Manage CloudWatch alarms)]를 선택합니다.

또는 [경보 상태(Alarm status)] 열에서 더하기 기호
(+)
를 선택해도 됩니다.

4. [CloudWatch 경보 관리(Manage CloudWatch alarms)] 페이지에서 다음을 수행합니다.

   a. [경보 생성(Create an alarm)]을 선택합니다.

   b. 경보가 트리거될 때 이메일을 받으려면 [경보 알림(Alarm notification)]에 대해 기존 Amazon SNS 주제를 선택합니다. 먼저 Amazon SNS 콘솔을 사용하여 Amazon SNS 주제를 생성해야 합니다. 자세한 내용은 Amazon Simple Notification Service 개발자 안내서에서 A2P(Application-to-Person) 메시징에 Amazon SNS 사용을 참조하세요.

   > **Note**
   >
   > 경보가 트리거될 때 이메일 알림을 수신하려면 사용자는 지정된 SNS 주제를 구독해야 합니다. AWS 계정 루트 사용자는 SNS 주제가 지정되지 않았거나 루트 사용자가 지정된 SNS 주제를 구독하지 않더라도 자동 인스턴스 복구 작업이 발생하면 항상 이메일 알림을 받습니다.

   c. [경보 작업(Alarm action)]을 켜고 [복구(Recover)]를 선택합니다.

   d. [샘플 그룹화 기준(Group samples by)]과 [샘플링할 데이터 유형(Type of data to sample)]에 대해 통계 및 지표를 선택합니다. 이 예에서는 [평균(Average)] 및 [상태 확인 실패: 시스템(Status check failed: system)]을 선택합니다.

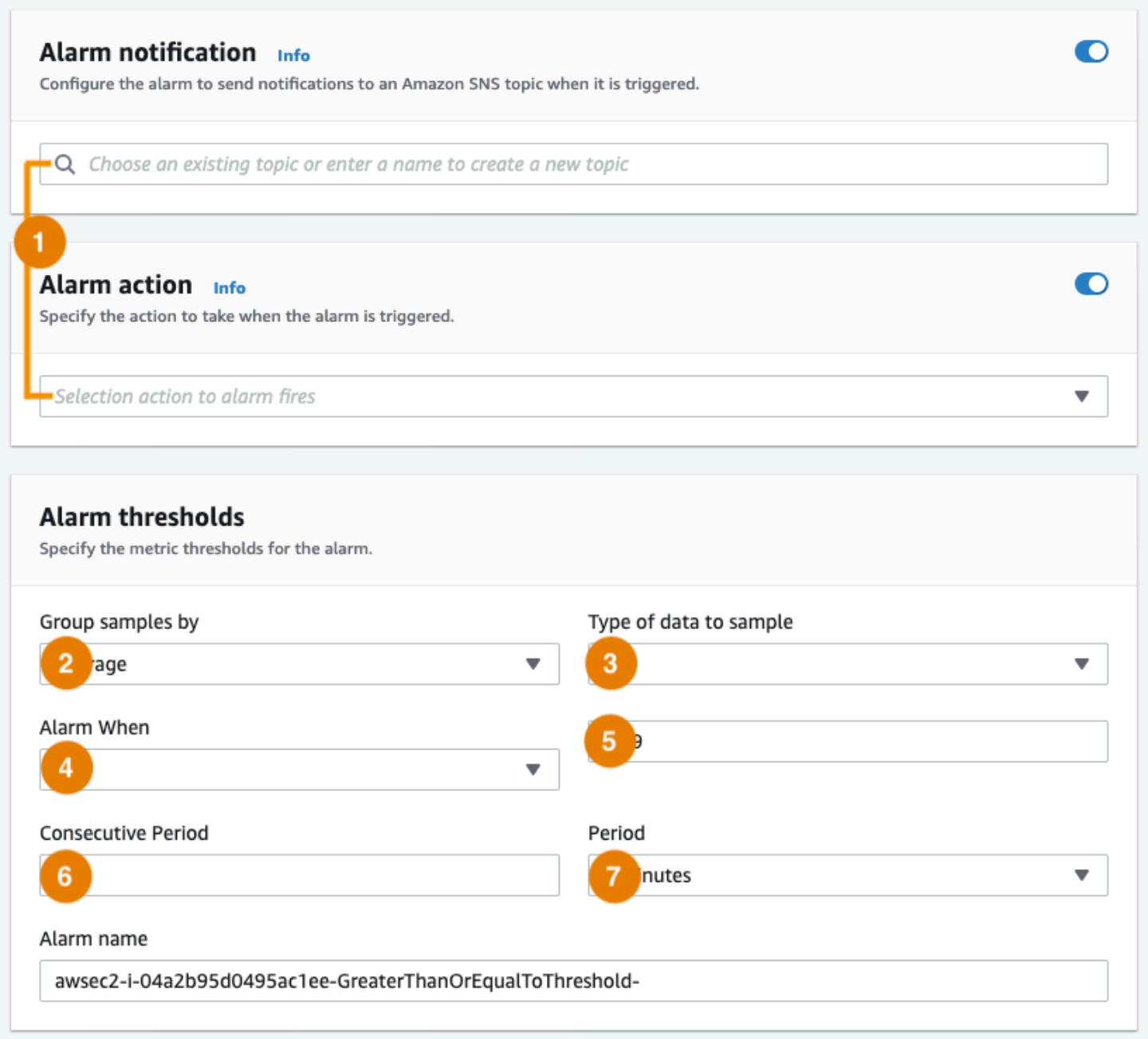
   e. [연속 기간(Consecutive period)]과 [기간(Period)]에 대해 경보의 평가 기간을 지정합니다. 이 예에서는 1분의 2 연속 기간을 입력합니다. 1분이 비활성화된 경우 세부 모니터링을 활성화해야 합니다. 그렇지 않으면 5분을 대신 선택할 수 있습니다.

   f. Amazon CloudWatch에서 자동으로 경보 이름이 생성됩니다. 이름을 변경하려면 [경보 이름(Alarm name)]에 새 이름을 입력합니다. 경보 이름은 ASCII 문자만 포함해야 합니다.

   g. 생성(Create)을 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

설정	값
2	Maximum
3	CPU 사용률
4	<=
5	10%
6	1
7	1시간

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	CPU 사용률
4	<=
5	5%
6	24
7	1시간

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	네트워크 출력
4	>
5	10GB
6	24
7	1시간

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	네트워크 출력
4	>
5	1GB
6	1
7	1시간

시나리오 5: 손상된 인스턴스 중지

5분 간격으로 수행된 연속 3회의 상태 확인에 실패한 인스턴스를 중지하는 경보를 만듭니다.

설정	값
1	Stop
2	Average
3	상태 확인 실패: 시스템
4	-
5	-
6	1
7	15분

시나리오 6: 배치 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	네트워크 출력
4	<=
5	100,000 bytes
6	1
7	5분

EventBridge를 사용하여 Amazon EC2 자동화

Amazon EventBridge를 사용하여 AWS 서비스를 자동화하고 애플리케이션 가용성 문제나 리소스 변경 등의 시스템 이벤트에 자동으로 대응할 수 있습니다. AWS 서비스의 이벤트는 거의 실시간으로 EventBridge로 전송됩니다. 관심 있는 이벤트와 이벤트가 규칙과 일치할 때 수행할 작업을 표시하는 규칙을 생성할 수 있습니다. 자동으로 트리거할 수 있는 태스크는 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 알림
- Amazon SQS 대기열 생성

다음은 Amazon EC2에서 EventBridge를 사용하는 방법의 예입니다.

- 인스턴스가 실행 상태가 될 때마다 Lambda 함수를 활성화합니다.
- Amazon EBS 볼륨이 생성되거나 수정되면 Amazon SNS 주제에 알립니다.
- 다른 AWS 서비스에서 특정 이벤트 발생 시 Amazon EC2 Run Command를 사용하여 명령을 하나 이상의 Amazon EC2 인스턴스에 전송합니다.

자세한 내용은 Amazon EventBridge 사용 설명서를 참조하세요.

Amazon EC2 이벤트 유형

Amazon EC2는 다음 이벤트 유형을 지원합니다.

- EC2 AMI 상태 변경
- EC2 빠른 시작 상태 변경 알림
- EC2 플릿 오류
- EC2 플릿 정보
- EC2 플릿 인스턴스 변경
- EC2 플릿 스팟 인스턴스 요청 변경
- EC2 플릿 상태 변경
- EC2 인스턴스 재분배 권장 사항

- EC2 인스턴스 상태 변경 알림
- EC2 스팟 플릿 오류
- EC2 스팟 플릿 정보
- EC2 스팟 플릿 인스턴스 변경
- EC2 스팟 플릿 스팟 인스턴스 요청 변경
- EC2 스팟 플릿 상태 변경
- EC2 스팟 인스턴스 중단 경고
- EC2 스팟 인스턴스 요청 이행
- EC2 ODCR 사용률 부족 알림

Amazon EBS에서 지원하는 이벤트 유형에 대한 자세한 내용은 Amazon EBS용 Amazon EventBridge를 참조하세요.

AWS CloudTrail을 사용하여 Amazon EC2 API 호출 로깅

Amazon EC2 API는 사용자, 역할 또는 AWS 서비스가 수행한 작업 기록을 제공하는 서비스인 AWS CloudTrail과 통합되어 있습니다. CloudTrail은 모든 Amazon EC2 API 호출을 이벤트로 캡처합니다. 캡처된 호출에는 콘솔에서의 호출이 포함됩니다. CloudTrail에서 수집한 정보를 사용하여 Amazon EC2 API에 수행된 요청, 요청이 수행된 IP 주소, 요청이 수행된 시간, 추가 세부 정보 등을 확인할 수 있습니다.

모든 이벤트 및 로그 항목에는 요청을 생성한 사용자에 대한 정보가 들어 있습니다. 보안 인증 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청을 루트 사용자로 했는지 사용자 보안 인증으로 했는지 여부.
- IAM Identity Center 사용자를 대신하여 요청이 이루어졌는지 여부입니다.
- 역할 또는 페더레이션 사용자에 대한 임시 보안 인증 정보를 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청했는지 여부.

계정을 생성할 때 AWS 계정에서 CloudTrail이 활성 상태이며, CloudTrail 이벤트 기록에 자동으로 액세스할 수 있습니다. CloudTrail 이벤트 기록은 지난 90일 간 AWS 리전의 관리 이벤트에 대해 보기, 검색 및 다운로드가 가능하고, 수정이 불가능한 레코드를 제공합니다. 자세한 설명은 AWS CloudTrail 사용 설명서의 CloudTrail 이벤트 기록 작업을 참조하세요. Event history(이벤트 기록) 보기는 CloudTrail 요금이 부과되지 않습니다.

지난 90일 동안 AWS 계정에서 진행 중인 이벤트 기록을 보려면 추적 또는 CloudTrail Lake 이벤트 데이터 스토어를 생성합니다.

CloudTrail 추적

CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. AWS Management Console을 사용하여 만든 추적은 모두 다중 리전입니다. AWS CLI를 사용하여 단일 리전 또는 다중 리전 추적을 생성할 수 있습니다. 계정의 모든 AWS 리전에서 활동을 캡처하므로, 다중 리전 추적 생성이 권장됩니다. 단일 리전 추적을 생성하는 경우 추적의 AWS 리전에 로깅된 이벤트만 볼 수 있습니다. 추적에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 Creating a trail for your AWS 계정 및 Creating a trail for an organization을 참조하세요.

CloudTrail에서 추적을 생성하여 진행 중인 관리 이벤트의 사본 하나를 Amazon S3 버킷으로 무료로 전송할 수는 있지만, Amazon S3 스토리지 요금이 부과됩니다. CloudTrail 요금에 대한 자세한 내용은 AWS CloudTrail 요금을 참조하세요. Amazon S3 요금에 대한 자세한 내용은 Amazon S3 요금을 참조하세요.

CloudTrail Lake 이벤트 데이터 스토어

CloudTrail Lake를 사용하면 이벤트에 대해 SQL 기반 쿼리를 실행할 수 있습니다. CloudTrail Lake는 행 기반 JSON 형식의 기존 이벤트를 Apache ORC 형식으로 변환합니다. ORC는 빠른 데이터 검색에 최적화된 열 기반 스토리지 형식입니다. 이벤트는 이벤트 데이터 스토어로 집계되며, 이벤트 데이터 스토어는 고급 이벤트 선택기를 적용하여 선택한 기준을 기반으로 하는 변경 불가능한 이벤트 컬렉션입니다. 이벤트 데이터 스토어에 적용하는 선택기는 어떤 이벤트가 지속되고 쿼리할 수 있는지 제어합니다. CloudTrail Lake에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 Working with AWS CloudTrail Lake를 참조하세요.

CloudTrail Lake 이벤트 데이터 스토어 및 쿼리에는 비용이 발생합니다. 이벤트 데이터 스토어를 생성할 때 이벤트 데이터 스토어에 사용할 요금 옵션을 선택합니다. 요금 옵션에 따라 이벤트 모으기 및 저장 비용과 이벤트 데이터 스토어의 기본 및 최대 보존 기간이 결정됩니다. CloudTrail 요금에 대한 자세한 내용은 AWS CloudTrail 요금을 참조하세요.

CloudTrail의 Amazon EC2 API 관리 이벤트

관리 이벤트는 AWS 계정의 리소스에 대해 수행되는 관리 작업에 대한 정보를 제공합니다. 이를 제어 영역 작업이라고도 합니다. 기본적으로 CloudTrail은 관리 이벤트를 로깅합니다.

모든 Amazon EC2 API 작업은 관리 이벤트로 로깅됩니다. CloudTrail로 로깅된 API 작업 목록은 Amazon EC2 API 참조를 확인하세요. 예를 들어, RunInstances, DescribeInstances 및 StopInstances 작업 호출은 관리 이벤트로 로깅됩니다.

Amazon EC2 API 이벤트 예제

이벤트는 모든 소스로부터의 단일 요청을 나타내며 요청된 API 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보가 들어 있습니다. CloudTrail 로그 파일은 퍼블릭 API 직접 호출의 주문 스택 추적이 아니므로 이벤트가 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"
  }
 ]
}
```

CloudTrail 레코드 콘텐츠에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 CloudTrail record contents를 참조하세요.

EC2 Instance Connect를 사용한 연결 감사

AWS CloudTrail을 사용하여 EC2 Instance Connect를 통해 인스턴스에 연결하는 사용자를 감사할 수 있습니다.

AWS CloudTrail 콘솔을 사용하여 EC2 Instance Connect를 통한 SSH 활동을 감사하려면

1. https://console.aws.amazon.com/cloudtrail/에서 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 사용 설명서에서 CloudTrail 로그 파일 가져오기 및 보기를 참조하세요.

CloudWatch Application Insights를 사용하여 .NET 및 SQL Server 애플리케이션 모니터링

CloudWatch Application Insights를 사용하면 다른 AWS 애플리케이션 리소스와 함께 Amazon EC2 인스턴스를 사용하는 .NET 및 SQL Server 애플리케이션을 모니터링할 수 있습니다. 이 기능은 애플리케이션 리소스 및 기술 스택(예: Microsoft SQL Server 데이터베이스, 웹(IIS) 및 애플리케이션 서버, OS, 로드 밸런서, 대기열 등) 전반에서 주요 지표 로그 및 경보를 파악하고 설정합니다. 지표 및 로그를 지속적으로 모니터링하여 이상 및 오류를 감지하고 연결합니다. 오류 및 이상이 감지되면 Application Insights에서 알림을 설정하고 작업을 수행할 수 있는 이벤트를 생성합니다. 문제 해결을 돕기 위해 감지한 문제에 대한 자동 대시보드를 생성함으로써 상관관계가 있는 지표 이상 항목 및 로그 오류와 함께 잠재적인 근본 원인을 알려주는 추가 통찰력을 제공합니다. 자동화된 대시보드를 사용하면 애플리케이션의 상태를 정상으로 유지하고 애플리케이션의 최종 사용자에게 미치는 영향을 방지하기 위해 신속한 조치를 취할 수 있습니다.

감지된 문제에 대해 다음과 같은 정보가 제공됩니다.

- 문제의 간략한 요약
- 문제의 시작 시간 및 날짜
- 문제의 심각도: 높음/중간/낮음
- 감지된 문제의 상태: 진행 중/해결됨
- 통찰력: 감지된 문제 및 가능한 근본 원인에 대해 자동으로 생성된 통찰력
- 통찰력에 대한 피드백: .NET 및 SQL Server용 CloudWatch Application Insights에서 생성한 통찰력의 유용성에 대한 피드백
- 관련 관찰: 다양한 애플리케이션 구성 요소에서 발생한 문제와 관련된 로그의 오류 조각 및 지표 이상의 상세 보기

Feedback

감지된 문제에 대해 자동으로 생성된 통찰력에 대한 유용성을 평가하여 피드백을 제공할 수 있습니다. 통찰력에 대한 피드백은 애플리케이션 진단(지표 이상 및 로그 예외)과 함께 향후 유사한 문제의 탐지를 개선하는 데 사용됩니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch Application Insights 설명서를 참조하세요.

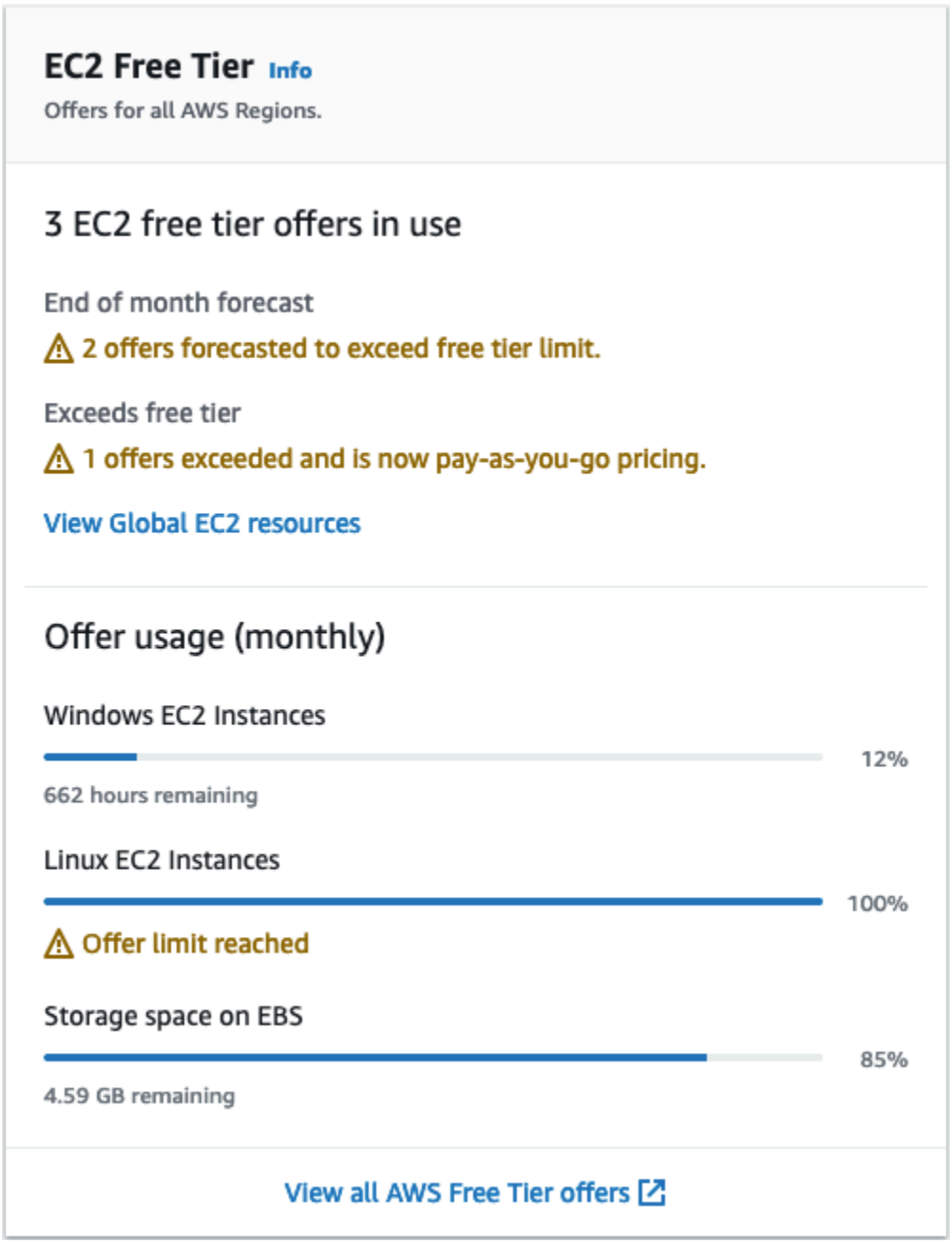
Amazon EC2의 프리 티어 사용량 추적

AWS 고객이 된 지 12개월 미만이고 AWS 프리 티어 사용량 한도를 초과하지 않는 경우 요금 발생 없이 Amazon EC2를 사용할 수 있습니다. 과금 문제를 방지하려면 프리 티어 사용량을 추적하는 것이 중요합니다. 프리 티어 한도를 초과하면 표준 종량 과금제에 따라 요금이 부과됩니다. 자세한 내용은 AWS 프리 티어 단원을 참조하십시오.

> **Note**
>
> 12개월 이상 AWS 고객인 경우 더 이상 프리 티어 사용 자격이 없으며 다음 절차에 설명된 EC2 프리 티어 상자가 표시되지 않습니다.

프리 티어 사용량 추적

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 EC2 대시보드를 선택합니다.
3. EC2 프리 티어 상자(오른쪽 상단)를 찾습니다.



4. EC2 프리 티어 상자에서 다음과 같이 프리 티어 사용량을 확인합니다.

   - 사용 중인 EC2 프리 티어 혜택에서 다음 경고에 유의합니다.
     - 월말 예측 - 현재 사용 패턴을 계속하면 이번 달에 요금이 발생한다는 경고입니다.
     - 프리 티어 초과 - 프리 티어 한도를 초과했으며 이미 요금이 발생하고 있다는 경고입니다.
   - 혜택 사용량(월별)에서 Linux 인스턴스, Windows 인스턴스 및 EBS 스토리지의 사용량에 유의합니다. 백분율은 이번 달에 사용한 프리 티어 한도를 나타냅니다. 100%이면 추가 사용에 대한 요금이 발생합니다.

> **Note**
>
> 이 정보는 인스턴스를 생성한 후에만 표시됩니다. 그러나 사용 정보는 실시간으로 업데이트되지 않고 하루에 세 번 업데이트됩니다.

5. 추가 요금이 발생하지 않도록 하려면 현재 요금이 발생하고 있거나 프리 티어 한도 사용량을 초과할 경우 요금이 발생하는 리소스를 모두 삭제합니다.

   - 인스턴스 삭제 지침을 보려면 Amazon EC2 인스턴스 종료 섹션을 참조하세요.
   - 다른 리전에 요금이 발생할 수 있는 리소스가 있는지 확인하려면 EC2 프리 티어 상자에서 글로벌 EC2 리소스 보기를 선택하여 EC2 글로벌 뷰를 엽니다. 자세한 내용은 Amazon EC2 Global View를 사용하여 리전 간 리소스 보기 단원을 참조하십시오.

6. AWS 프리 티어의 모든 AWS 서비스에 대한 리소스 사용량을 보려면 EC2 프리 티어 상자 하단에서 모든 AWS 프리 티어 혜택 보기를 선택합니다. 자세한 내용은 AWS 결제 사용 설명서의 AWS 프리 티어를 사용한 서비스 사용을 참조하세요.

Amazon EC2 인스턴스 문제 해결

다음 절차와 팁은 Amazon EC2 인스턴스의 문제를 해결하는 데 도움이 될 수 있습니다.

문제

- Amazon EC2 인스턴스 시작 문제 해결
- Amazon EC2 인스턴스 중지 문제 해결
- Amazon EC2 인스턴스 종료 문제 해결
- 연결할 수 없는 Amazon EC2 인스턴스 문제 해결
- Amazon EC2 Linux 인스턴스 연결 문제 해결
- 상태 확인에 실패한 Amazon EC2 Linux 인스턴스 문제 해결
- 잘못된 볼륨에서 부팅되는 Amazon EC2 Linux 인스턴스 문제 해결
- Amazon EC2 Windows 인스턴스 연결 문제 해결
- Amazon EC2 Windows 인스턴스 시작 문제 해결
- Amazon EC2 Windows 인스턴스 문제 해결
- Amazon EC2 Windows 인스턴스의 Windows 관리자 암호 재설정
- Amazon EC2 Windows 인스턴스 Sysprep 문제 해결
- EC2Rescue로 손상된 Amazon EC2 Linux 인스턴스 문제 해결
- EC2Rescue를 사용하여 손상된 Amazon EC2 Windows 인스턴스 문제 해결
- 인스턴스용 EC2 직렬 콘솔
- 연결할 수 없는 Amazon EC2 인스턴스를 디버깅하기 위한 진단 인터럽트 보내기

Amazon EC2 인스턴스 시작 문제 해결

다음은 Amazon EC2 인스턴스를 시작할 때 발생하는 문제를 해결하는 데 도움이 되는 문제 해결 팁입니다.

시작 문제

- 잘못된 디바이스 이름
- 인스턴스 제한 초과됨

- 부족한 인스턴스 용량
- 요청된 구성이 현재 지원되지 않습니다. 지원되는 구성은 설명서를 참조하세요.
- 인스턴스 즉시 종료
- 권한 부족
- Windows 시작 직후 높은 CPU 사용량(Windows 인스턴스만 해당)

잘못된 디바이스 이름

설명

새 인스턴스를 시작하려고 할 때 `Invalid device name` *`device_name`* 오류가 발생합니다.

원인

인스턴스를 시작하려고 할 때 이 오류가 발생하면 요청에서 하나 이상의 볼륨에 대해 지정된 디바이스 이름에 잘못된 디바이스 이름이 있는 것입니다. 가능한 원인은 다음과 같습니다.

- 선택한 AMI에서 디바이스 이름을 사용 중일 수 있습니다.
- 디바이스 이름이 루트 볼륨용으로 예약되어 있을 수 있습니다.
- 요청에서 다른 볼륨에 디바이스 이름을 사용할 수 있습니다.
- 디바이스 이름이 운영 체제에 유효하지 않을 수 있습니다.

Solution

문제 해결

- 선택한 AMI에서 디바이스 이름이 사용되지 않았는지 확인합니다. 다음 명령을 실행하여 AMI에서 사용하는 디바이스 이름을 확인합니다.

```
aws ec2 describe-images --image-id ami-0abcdef1234567890 --query
 'Images[*].BlockDeviceMappings[].DeviceName'
```

- 루트 볼륨용으로 예약된 디바이스 이름을 사용하고 있지 않은지 확인합니다. 자세한 내용은 사용 가능한 디바이스 이름 섹션을 참조하세요.
- 요청에 지정된 각 볼륨에 고유한 디바이스 이름이 있는지 확인합니다.

- 지정한 디바이스 이름이 올바른 형식인지 확인하세요. 자세한 내용은 사용 가능한 디바이스 이름 섹션을 참조하세요.

인스턴스 제한 초과됨

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 제한에 도달한 것입니다. AWS 계정을 생성할 때 지역별로 실행할 수 있는 인스턴스의 기본 제한이 설정됩니다.

Solution

리전을 기준으로 인스턴스 한도 증가를 요청할 수 있습니다. 자세한 내용은 Amazon EC2 서비스 할당량 섹션을 참조하세요.

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 이 오류가 발생하면 현재 AWS에 요청을 이행할 만큼 충분한 가용 온디맨드 용량이 없는 것입니다.

Solution

다음에 따라 문제를 해결하세요.

- 몇 분 정도 기다린 후 다시 요청을 제출합니다. 용량은 자주 변할 수 있습니다.

- 인스턴스 수가 줄어든 새 요청을 제출하세요. 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 제출하세요.
- 인스턴스를 시작하고 있는 경우 이후의 단계에서 크기를 조정할 수 있는 다른 인스턴스 유형을 사용하여 새 요청을 제출하세요. 자세한 내용은 Amazon EC2 인스턴스 유형 변경 섹션을 참조하세요.
- 클러스터 배치 그룹으로 인스턴스를 시작한 경우 용량 부족 오류가 발생할 수 있습니다.

요청된 구성이 현재 지원되지 않습니다. 지원되는 구성은 설명서를 참조하세요.

설명

지원되지 않는 인스턴스 구성으로 새 인스턴스를 시작하려고 하면 `Unsupported` 오류가 발생합니다.

원인

오류 메시지에서 추가 세부 정보를 확인할 수 있습니다. 예를 들어 지정된 리전 또는 가용 영역에서 인스턴스 유형 또는 인스턴스 구매 옵션이 지원되지 않을 수 있습니다.

Solution

다른 인스턴스 구성을 시도합니다. 요구 사항에 맞는 인스턴스 유형을 검색하려면 Amazon EC2 인스턴스 유형 찾기 섹션을 참조하세요.

인스턴스 즉시 종료

설명

인스턴스가 `pending` 상태에서 `terminated` 상태로 전환됩니다.

원인

인스턴스가 즉시 종료되는 이유에는 다음과 같이 몇 가지가 있습니다.

- EBS 볼륨 제한을 초과했습니다. 자세한 내용은 Amazon EC2 인스턴스의 Amazon EBS 볼륨 한도 섹션을 참조하세요.
- EBS 스냅샷이 손상되었습니다.

- 루트 EBS 볼륨이 암호화되었는데 사용자는 암호 해독을 위하여 KMS 키에 액세스할 권한이 없습니다.
- AMI에 대한 블록 디바이스 매핑에 지정된 스냅샷이 암호화되었는데 사용자가 암호 해독을 위해 KMS 키에 액세스할 권한이 없거나 복원된 볼륨을 암호화하기 위해 KMS 키에 액세스할 수 없습니다.
- 인스턴스를 시작하는 데 사용한 인스턴스 스토어 지원 AMI에 필수 부분(image.part.xx 파일).

자세한 내용을 알아보려면 다음 방법 중 하나를 사용하여 종료 이유를 확인하세요.

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. 첫 번째 탭에서 상태 전환 이유(State transition reason) 옆에 나온 이유를 확인합니다.

AWS CLI을 사용해서 종료 이유를 파악하는 방법

1. describe-instances 명령을 사용하여 인스턴스 ID를 지정합니다.

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

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"
},
```

AWS CloudTrail을 사용해서 종료 이유를 파악하는 방법

자세한 내용은 AWS CloudTrail 사용 설명서에서 CloudTrail 이벤트 기록을 사용하여 이벤트 보기를 참조하세요.

Solution

종료 이유에 따라 다음 작업 중 하나를 수행합니다.

- **`Client.VolumeLimitExceeded: Volume limit exceeded`** — 사용되지 않는 볼륨을 삭제합니다. 볼륨 제한을 늘리도록 요청을 제출할 수 있습니다.
- **`Client.InternalError: Client error on launch`** - 볼륨 복호화 및 암호화에 사용되는 AWS KMS keys에 액세스하는 데 필요한 권한이 있는지 확인합니다. 자세한 내용은 AWS Key Management Service 개발자 안내서에서 AWS KMS의 키 정책 사용을 참조하세요.

권한 부족

설명

새 인스턴스를 시작하려고 할 때 `"errorMessage": "You are not authorized to perform this operation."` 오류가 발생하고 시작이 실패합니다.

원인

인스턴스를 시작하려고 할 때 이 오류가 발생하는 경우 인스턴스를 시작하는 데 필요한 IAM 권한이 없는 것입니다.

다음과 같은 권한이 누락된 것일 수 있습니다.

- `ec2:RunInstances`
- `iam:PassRole`

다른 권한이 필요할 수도 있습니다. 인스턴스를 시작하는 데 필요한 권한 목록은 예: EC2 시작 인스턴스 마법사 사용 및 인스턴스 시작(RunInstances) 아래에서 예제 IAM 정책을 참조하세요.

Solution

문제 해결

- IAM 사용자로 요청을 하는 경우, 다음과 같은 권한이 있는지 확인합니다.
  - 와일드카드 리소스("*")가 포함된 `ec2:RunInstances`
  - 역할 ARN과 일치하는 리소스가 포함된 `iam:PassRole`(예: `arn:aws:iam::999999999999:role/ExampleRoleName`)
- 이와 같은 권한이 없는 경우 IAM 역할 또는 사용자와 연결된 IAM 정책을 편집하여 누락된 필수 권한을 추가하세요.

문제가 해결되지 않고 시작 실패 오류가 계속 발생하는 경우 오류에 포함된 권한 부여 실패 메시지를 디코딩할 수 있습니다. 디코딩된 메시지에는 IAM 정책에서 누락된 권한이 포함되어 있습니다. 자세한 내용은 EC2 인스턴스 시작 중에 "UnauthorizedOperation" 오류가 발생한 후 권한 부여 실패 메시지를 디코딩하는 방법을 참조하세요.

Windows 시작 직후 높은 CPU 사용량(Windows 인스턴스만 해당)

> **Note**
>
> 이 문제 해결 팁은 Windows 인스턴스에만 해당됩니다.

Windows Update가 업데이트를 확인하지만 다운로드 및 설치 여부는 직접 선택(기본 인스턴스 설정)으로 설정되어 있는 경우 이 검사를 실행하면 인스턴스 CPU가 50~99%까지 소비될 수 있습니다. 이로 인해 애플리케이션에 문제가 발생할 경우에는 제어판에서 Windows Update 설정을 수동으로 변경하거나, 혹은 Amazon EC2 사용자 데이터 필드에서 다음 스크립트를 사용하세요.

```
reg add "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Auto Update" /v
 AUOptions /t REG_DWORD /d 3 /f net stop wuauserv net start wuauserv
```

이 스크립트를 실행할 때는 /d 값을 지정합니다. 기본값은 3입니다. 가능한 값은 다음을 포함합니다.

1. 업데이트를 확인하지 않음
2. 업데이트를 확인하지만 다운로드 및 설치 여부는 직접 선택
3. 업데이트를 다운로드하지만 설치 여부는 직접 선택
4. 업데이트 자동 설치

인스턴스의 사용자 데이터를 변경한 후에는 인스턴스를 실행할 수 있습니다. 자세한 내용은 시작 시 Windows 인스턴스에서 명령 실행을 참조하세요.

Amazon EC2 인스턴스 중지 문제 해결

Amazon EBS 지원 인스턴스가 `stopping` 상태로 멈춰 있는 것처럼 보일 경우 기본 호스트 컴퓨터에 문제가 있을 수 있습니다.

이 문제를 해결하려면 다음 단계에 따릅니다.

1. 인스턴스 강제 중지

   Amazon EC2 콘솔 또는 AWS CLI를 사용하여 인스턴스를 강제로 중지합니다. 이러한 단계는 인스턴스 강제 중지을 참조하세요.

   인스턴스는 먼저 파일 시스템 캐시 및 메타데이터 비우기를 포함한 정상 종료를 시도합니다. 제한 시간 내에 정상 종료가 완료되지 않으면 파일 시스템 캐시 및 메타데이터를 비우지 않고 인스턴스가 강제로 종료됩니다.

2. 강제 중지 후

   파일 시스템 검사 및 복구 절차를 수행합니다.

   > **Important**
   >
   > 강제 중지 시 파일 시스템 캐시 및 메타데이터가 비워지지 않으므로 이러한 절차를 수행하는 것이 중요합니다.

3. 강제 중지가 실패하는 경우

   10분 후에도 인스턴스가 중지되지 않은 경우 다음을 수행합니다.

   a. AWS re:Post에 도움 요청을 게시합니다. 해결 방법을 신속히 찾아내려면 인스턴스 ID를 포함하고 자신이 이미 수행했던 단계에 대해 설명하세요.
   b. 지원 플랜이 있는 경우에는 지원 센터에서 기술 지원 사례를 요청할 수 있습니다.
   c. 지원을 기다리는 동안 필요한 경우 대체 인스턴스를 생성할 수 있습니다. 이러한 단계는 (선택 사항) 대체 인스턴스 생성을 참조하세요.

인스턴스가 `stopping` 상태 또는 `running`를 제외한 다른 상태에 있는 동안에는 인스턴스 사용 요금이 부과되지 않습니다. 인스턴스가 `running` 상태인 경우 인스턴스 사용량에 대해서만 요금이 부과됩니다.

내용

- 인스턴스 강제 중지
- (선택 사항) 대체 인스턴스 생성

인스턴스 강제 중지

인스턴스를 강제로 중지할 수 있습니다. 10분 후에도 인스턴스가 중지되지 않는 경우 AWS re:Post에 도움을 요청하는 글을 게시하세요. 해결 방법을 신속히 찾아내려면 인스턴스 ID를 포함하고 자신이 이미 수행했던 단계에 대해 설명하세요. 지원 플랜이 있는 경우에는 지원 센터에서 기술 지원 사례를 요청할 수 있습니다.

> **Note**
>
> 콘솔을 사용하여 인스턴스가 `stopping` 상태에 있는 동안에만 인스턴스를 통해 콘솔 사용을 강제로 중지할 수 있습니다. AWS CLI를 사용하여 인스턴스가 `pending`, `running` 또는 `stopping` 상태에 있는 동안에만 인스턴스를 통해 콘솔 사용을 강제로 중지할 수 있습니다.

Console

인스턴스를 강제로 중지하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택하고 멈춘 인스턴스를 선택합니다.
3. 인스턴스 상태(Instance state), 인스턴스 강제 중지(Force stop instance), 중지(Stop)를 선택합니다.

   인스턴스 강제 중지(Force stop instance)는 인스턴스가 `stopping` 상태일 때만 콘솔에서 사용할 수 있습니다. 인스턴스가 다른 상태인 경우(`shutting-down` 및 `terminated` 제외) AWS CLI를 사용하여 인스턴스를 강제로 중지할 수 있습니다.

AWS CLI

인스턴스를 강제로 중지하려면

stop-instances 명령을 `--force` 옵션과 함께 사용합니다.

```
aws ec2 stop-instances \
    --instance-ids i-1234567890abcdef0 \
    --force
```

PowerShell

인스턴스를 강제로 중지하려면

Stop-EC2Instance cmdlet을 사용하고 `-Enforce`를 `true`로 설정합니다.

```
Stop-EC2Instance `
    -InstanceId i-1234567890abcdef0 `
    -Enforce $true
```

(선택 사항) 대체 인스턴스 생성

AWS re:Post 또는 지원 센터의 도움을 기다리는 동안 필요한 경우 대체 인스턴스를 생성할 수 있습니다. 멈춘 인스턴스에서 AMI를 생성하고 새로운 AMI를 사용하여 새 인스턴스를 시작합니다.

> **Important**
>
> 인스턴스 상태 확인 시 AMI가 손상된 운영 체제의 정확한 복제본을 복사하므로 멈춘 인스턴스에서 시스템 상태 확인만 발생하는 경우 대체 인스턴스를 생성할 수 있습니다. 상태 메시지를 확인한 후 AMI를 생성하고, 새 AMI를 사용하여 새 인스턴스를 시작합니다.

Console

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(Actions), 이미지 및 템플릿(Image and templates), 이미지 생성(Create image)을 차례로 선택합니다.
4. 이미지 생성(Create image) 페이지에서 다음을 수행합니다.
   a. AMI 이름 및 설명을 입력합니다.
   b. 인스턴스 재부팅 선택을 취소합니다.
   c. 이미지 생성을 선택합니다.

   자세한 내용은 the section called “인스턴스에서 AMI 생성” 섹션을 참조하세요.
5. AMI에서 새로운 인스턴스를 시작하고 새로운 인스턴스가 작동하는지 확인합니다.

6. 멈춰 있는 인스턴스를 선택하고 작업, 인스턴스 상태, 인스턴스 종료(삭제)를 차례로 선택합니다. 또한 인스턴스가 종료 중 상태로 멈추는 경우 Amazon EC2에서 몇 시간 내에 해당 인스턴스를 자동으로 종료합니다.

이전 절차에 설명된 대로 AMI를 만들 수 없으면 다음과 같이 대체 인스턴스를 설정할 수 있습니다.

(대안) 콘솔을 사용하여 대체 인스턴스를 생성하려면

1. 인스턴스를 선택하고 설명, 블록 디바이스를 선택합니다. 각 볼륨을 선택하고 볼륨 ID를 기록합니다. 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 인스턴스(Instances)를 선택하고 방금 시작한 인스턴스를 선택한 다음 인스턴스 상태(Instance state), 인스턴스 중지(Stop instance)를 차례로 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(Actions), 볼륨 연결(Attach Volume)을 선택한 후, 이 볼륨을 새 인스턴스에 루트 볼륨으로 연결합니다(기록해 놓은 디바이스 이름 사용). 루트 이외의 다른 추가 볼륨을 인스턴스에 연결합니다.
8. 탐색 창에서 인스턴스를 선택하고 대체 인스턴스를 선택합니다. 인스턴스 상태, 인스턴스 시작을 차례로 선택합니다. 인스턴스가 작동 중인지 확인합니다.
9. 멈춘 인스턴스를 선택하고 인스턴스 상태, 인스턴스 종료(삭제)를 차례로 선택합니다. 또한 인스턴스가 종료 중 상태로 멈추는 경우 Amazon EC2에서 몇 시간 내에 해당 인스턴스를 자동으로 종료합니다.

AWS CLI

대체 인스턴스를 생성하려면

1. create-image 명령과 `--no-reboot` 옵션을 함께 사용하여 멈춰 있는 인스턴스에서 AMI를 생성합니다.

```
aws ec2 create-image \
    --instance-id i-1234567890abcdef0 \
    --name "my-replacement-ami" \
    --description "AMI for replacement instance" \
    --no-reboot
```

2. run-instances 명령을 사용하여 방금 생성한 AMI에서 새 인스턴스를 시작합니다.
3. 새로운 인스턴스가 작동 중인지 확인합니다.
4. (선택 사항) terminate-instances 명령을 사용하여 멈춰 있는 인스턴스를 종료합니다.

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0
```

PowerShell

대체 인스턴스를 생성하려면

1. New-EC2Image cmdlet을 사용하여 멈춰 있는 인스턴스에서 AMI를 생성하고 `-NoReboot`를 `true`로 설정합니다.

```
New-EC2Image `
    -InstanceId i-1234567890abcdef0 `
    -Name "my-replacement-ami" `
    -Description "AMI for replacement instance" `
    -NoReboot $true
```

2. New-EC2Instance cmdlet을 사용하여 방금 생성한 AMI에서 새 인스턴스를 시작합니다.
3. 새로운 인스턴스가 작동 중인지 확인합니다.
4. (선택 사항) Remove-EC2Instance cmdlet을 사용하여 멈춰 있는 인스턴스를 종료합니다.

```
Remove-EC2Instance -InstanceId i-1234567890abcdef0
```

Amazon EC2 인스턴스 종료 문제 해결

인스턴스를 종료하거나 삭제하는 것을 인스턴스 종료라고 합니다. 다음 정보는 인스턴스 종료 시 문제를 해결하는 데 도움이 될 수 있습니다.

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.

인스턴스 즉시 종료

몇 가지 문제로 인해 인스턴스가 시작 시 즉시 종료될 수 있습니다. 자세한 정보는 인스턴스 즉시 종료을 참조하세요.

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 AWS re:Post에 도움을 요청하는 글을 게시하십시오. 해결 방법을 신속히 찾아내려면 인스턴스 ID를 포함하고 자신이 이미 수행했던 단계에 대해 설명하세요. 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 문의하세요.

오류: 인스턴스가 종료되지 않을 수 있습니다. 'disableApiTermination' 인스턴스 특성 수정

인스턴스를 종료하려고 할 때 `The instance i-1234567890abcdef0 may not be terminated. Modify its 'disableApiTermination' instance attribute` 오류 메시지가 표시되면 해당 인스턴스에 종료 방지 기능이 활성화되었음을 나타냅니다. 종료 방지 기능은 인스턴스가 실수로 종료되는 것을 방지합니다.

인스턴스를 종료하려면 먼저 종료 방지 기능을 비활성화해야 합니다.

자세한 내용은 인스턴스 종료 방지 변경 섹션을 참조하세요.

인스턴스가 자동으로 시작되거나 종료됨

일반적으로 다음과 같은 동작은 사용자가 Amazon EC2 Auto Scaling, EC2 플릿 또는 스팟 플릿을 사용하여 정의한 기준에 따라 컴퓨팅 리소스의 크기를 자동으로 조정했음을 의미합니다.

- 인스턴스를 종료하면 새 인스턴스가 자동으로 시작됩니다.
- 인스턴스를 시작하면 인스턴스 중 하나가 자동으로 종료됩니다.
- 인스턴스를 중지하면 인스턴스가 종료되고 새 인스턴스가 자동으로 시작됩니다.

자동 규모 조정을 중지하려면 인스턴스를 시작하는 Auto Scaling 그룹 또는 플릿을 찾아 용량을 0으로 설정하거나 삭제합니다.

연결할 수 없는 Amazon EC2 인스턴스 문제 해결

다음 정보는 연결할 수 없는 Amazon EC2 인스턴스의 문제를 해결하는 데 도움이 될 수 있습니다. 문제를 진단하고 인스턴스를 재부팅해야 하는지 여부를 판단하는 데 도움이 되도록 스크린샷을 캡처하거나 콘솔 출력에 액세스할 수 있습니다. 연결할 수 없는 Windows 인스턴스의 경우 서비스에서 반환된 스크린샷을 검토하여 문제를 해결하세요.

내용

- 인스턴스 재부팅
- 인스턴스 콘솔 출력
- 연결할 수 없는 인스턴스의 스크린샷 캡처
- 연결할 수 없는 Windows 인스턴스 문제 해결을 위한 일반 스크린샷
- 호스트 컴퓨터 실패 시 인스턴스 복구
- 인스턴스가 오프라인으로 표시되고 예기치 않게 재부팅됨

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 Amazon EC2 인스턴스 재부팅 섹션을 참조하세요.

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 인스턴스 – 인스턴스 콘솔 출력은 컴퓨터에 연결된 실제 모니터에 일반적으로 표시되는 정확한 콘솔 출력을 표시합니다. 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 인스턴스 - 인스턴스 콘솔 출력에는 최근 3개의 시스템 이벤트 로그 오류가 포함됩니다.

인스턴스 소유자만 콘솔 출력에 액세스할 수 있습니다.

인스턴스 수명 주기 동안 최신 직렬 콘솔 출력을 검색할 수 있습니다. 이 옵션은 Nitro 기반 인스턴스에서만 지원됩니다.

Console

콘솔 출력을 가져오려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다.
4. [작업(Actions)], [모니터링 및 문제 해결(Monitor and troubleshoot)], [시스템 로그 가져오기(Get system log)]를 선택합니다.

AWS CLI

콘솔 출력을 가져오려면

get-console-output 명령을 사용합니다.

```
aws ec2 get-console-output --instance-id i-1234567890abcdef0
```

PowerShell

콘솔 출력을 가져오려면

Get-EC2ConsoleOutput cmdlet을 사용합니다.

```
Get-EC2ConsoleOutput -InstanceId i-1234567890abcdef0
```

연결할 수 없는 인스턴스의 스크린샷 캡처

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. 이미지는 JPG 형식으로 생성되며, 100KB보다 크지 않습니다. 스크린샷의 데이터 전송 비용은 따로 들지 않습니다.

제한 사항

이 기능은 다음에서 지원되지 않습니다.

- 베어 메탈 인스턴스(`*.metal` 유형의 인스턴스)
- 인스턴스가 NVIDIA GRID 드라이버를 사용하고 있음
- ARM 기반 Graviton 프로세서로 구동되는 인스턴스
- AWS Outposts의 Windows 인스턴스
- AWS 로컬 영역의 Windows 인스턴스

리전 지원

다음 리전에서는 이 기능을 사용할 수 없습니다.

- 아시아 태평양(태국)
- 멕시코(중부)

Console

인스턴스의 스크린샷을 가져오는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 인스턴스를 선택합니다.
3. 캡처할 인스턴스를 선택합니다.
4. [작업(Actions)], [모니터링 및 문제 해결(Monitor and troubleshoot)], [인스턴스 스크린샷 가져오기(Get instance screenshot)]를 선택합니다.

5. [다운로드(Download)]를 선택하거나 이미지를 마우스 오른쪽 버튼으로 클릭하여 다운로드하고 저장합니다.

AWS CLI

인스턴스 스크린샷을 캡처하는 방법

get-console-screenshot 명령을 사용합니다. 출력은 base64 인코딩됩니다.

```
aws ec2 get-console-screenshot --instance-id i-1234567890abcdef0
```

PowerShell

인스턴스 스크린샷을 캡처하는 방법

Get-EC2ConsoleScreenshot cmdlet을 사용합니다. 출력은 base64 인코딩됩니다.

```
Get-EC2ConsoleScreenshot -InstanceId i-1234567890abcdef0
```

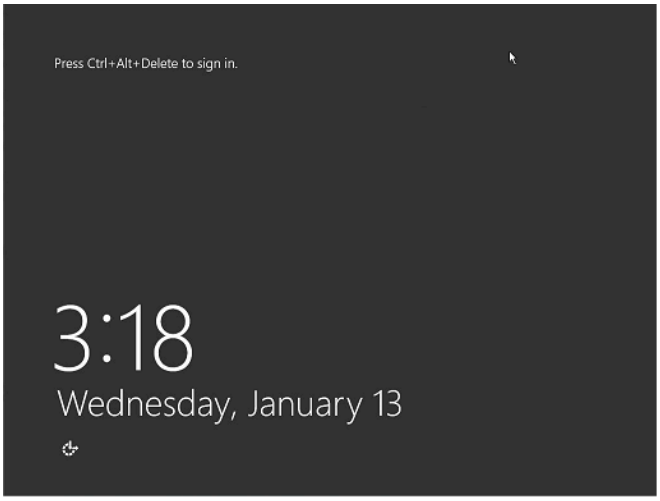
연결할 수 없는 Windows 인스턴스 문제 해결을 위한 일반 스크린샷

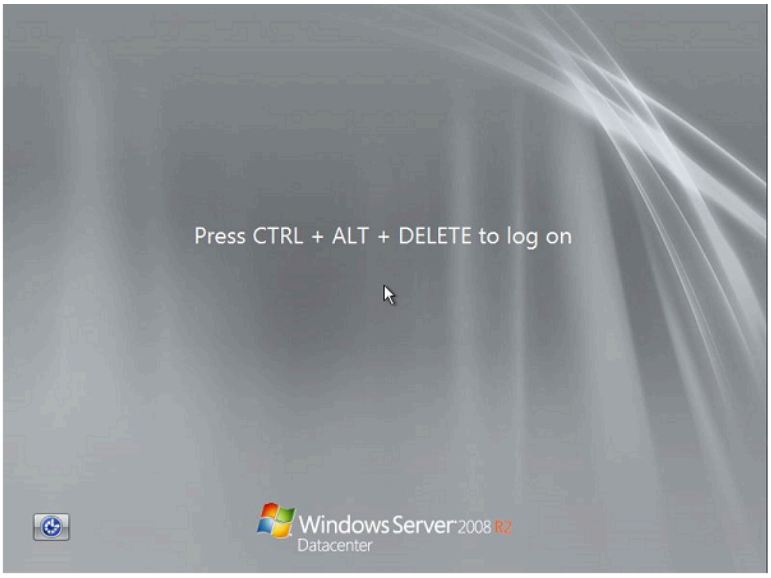
다음 정보를 사용하면 서비스에서 반환되는 스크린샷을 바탕으로 연결할 수 없는 Windows 인스턴스의 문제 해결에 도움을 받을 수 있습니다.

- 로그온 화면(Ctrl+Alt+Delete)
- 복구 콘솔 화면
- Windows 부팅 관리자 화면
- Sysprep 화면
- 준비 화면
- Windows 업데이트 화면
- Chkdsk

로그온 화면(Ctrl+Alt+Delete)

콘솔 스크린샷 서비스에서 다음을 반환했습니다.





인스턴스가 로그온 중에 연결할 수 없게 되면 네트워크 구성 또는 Windows 원격 데스크톱 서비스에 문제가 있을 수 있습니다. 프로세스에서 대량의 CPU를 사용 중인 경우 인스턴스가 응답하지 않을 수도 있습니다.

네트워크 구성

다음 정보를 사용하여 AWS, Microsoft Windows와 로컬(또는 온프레미스) 네트워크 구성이 인스턴스에 대한 액세스를 차단하고 있지 않은지 확인합니다.

AWS 네트워크 구성

구성	확인
보안 그룹 구성	포트 3389가 보안 그룹용으로 열려 있는지 확인합니다. 올바른 퍼블릭 IP 주소에 연결 중인지 확인합니다. 인스턴스가 탄력적 IP와 연결되지 않은 경우, 퍼블릭 IP는 인스턴스가 중지/시작된 후 바뀝니다. 자세한 내용은 원격 데스크톱으로 원격 컴퓨터에 연결할 수 없음 섹션을 참조하세요.
VPC 구성(네트워크 ACL)	Amazon VPC에 대한 ACL(액세스 제어 목록)이 액세스를 차단하고 있지 않은지 확인합니다. 자세한 내용은 Amazon VPC 사용 설명서의 네트워크 ACL을 참조하세요.

구성	확인
VPN 구성	가상 프라이빗 네트워크(VPN)를 사용하여 VPC에 연결 중인 경우 VPN 터널 연결을 확인하세요. 자세한 내용은 Amazon VPC에 대한 VPN 터널 연결의 문제 해결 방법 섹션을 참조하세요.

Windows 네트워크 구성

구성	확인
Windows 방화벽	Windows 방화벽이 인스턴스에 대한 연결을 차단하고 있지 않은지 확인하세요. 원격 데스크톱 문제 해결 섹션 원격 데스크톱으로 원격 컴퓨터에 연결할 수 없음의 7번째 글머리 기호 항목에 설명되어 있는 것처럼 Windows 방화벽을 비활성화합니다.
고급 TCP/IP 구성(정적 IP 사용)	정적 IP 주소를 구성했기 때문에 인스턴스가 응답하지 않을 수도 있습니다. VPC의 경우 네트워크 인터페이스를 생성하고 인스턴스에 연결합니다.

로컬 또는 온프레미스 네트워크 구성

로컬 네트워크 구성이 액세스를 차단하고 있지 않은지 확인합니다. 연결할 수 없는 인스턴스와 똑같은 VPC에서 다른 인스턴스에 연결해 보세요. 다른 인스턴스에 액세스할 수 없는 경우 로컬 네트워크 관리자와 협력하여 로컬 정책이 액세스를 제한하고 있는지 확인합니다.

원격 데스크톱 서비스 문제

인스턴스가 로그온 중에 연결할 수 없게 되면 해당 인스턴스에 원격 데스크톱 서비스(RDS) 문제가 있을 수 있습니다.

> **Tip**
>
> `AWSSupport-TroubleshootRDP` 런북을 사용하여 원격 데스크톱 프로토콜(RDP) 연결에 영향을 줄 수 있는 다양한 설정을 확인하고 수정할 수 있습니다. 자세한 내용은 AWS Systems Manager Automation 실행서 참조에서 `AWSSupport-TroubleshootRDP`를 참조하세요.

원격 데스크톱 서비스 구성

구성	확인
RDS가 실행 중임	인스턴스에서 RDS가 실행 중인지 확인합니다. Microsoft Management Console(MMC) 서비스 스냅인(`services.msc`)을 사용하여 인스턴스에 연결합니다. 서비스 목록에서 원격 데스크톱 서비스가 실행인지 확인합니다. 그렇지 않은 경우 이 서비스를 시작한 다음 시작 유형을 자동으로 설정합니다. 서비스 스냅인을 사용하여 인스턴스에 연결할 수 없으면 인스턴스에서 루트 볼륨을 분리하고, 볼륨의 스냅샷을 생성하거나 볼륨에서 AMI를 만들고, 원본 볼륨을 보조 볼륨과 동일한 가용 영역에 있는 다른 인스턴스에 연결하고, 시작 레지스트리 키를 수정합니다. 위 작업을 마쳤으면 루트 볼륨을 원본 인스턴스에 다시 연결합니다.
RDS가 활성화됨	서비스가 시작되더라도 비활성화될 수 있습니다. 원격 레지스트리를 사용하여 EC2 인스턴스에서 원격 데스크톱 활성화에 설명되어 있는 것처럼 인스턴스에서 루트 볼륨을 분리하고, 볼륨의 스냅샷을 생성하거나 볼륨에서 AMI를 만들고, 원본 볼륨을 보조 볼륨과 동일한 가용 영역에 있는 다른 인스턴스에 연결하고, Terminal Server(터미널 서버) 레지스트리 키를 수정하여 서비스를 활성화합니다. 위 작업을 마쳤으면 루트 볼륨을 원본 인스턴스에 다시 연결합니다.

높은 CPU 사용률

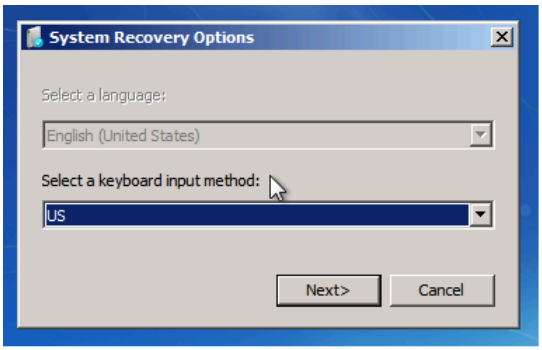
Amazon CloudWatch를 사용하여 인스턴스에서 CPUUtilization (Maximum) 측정치를 확인합니다. CPUUtilization (Maximum) 수치가 높으면 CPU 사용률이 낮아지기를 기다렸다가 다시 연결해 보세요. 다음과 같은 원인으로 CPU 사용률이 높아질 수 있습니다.

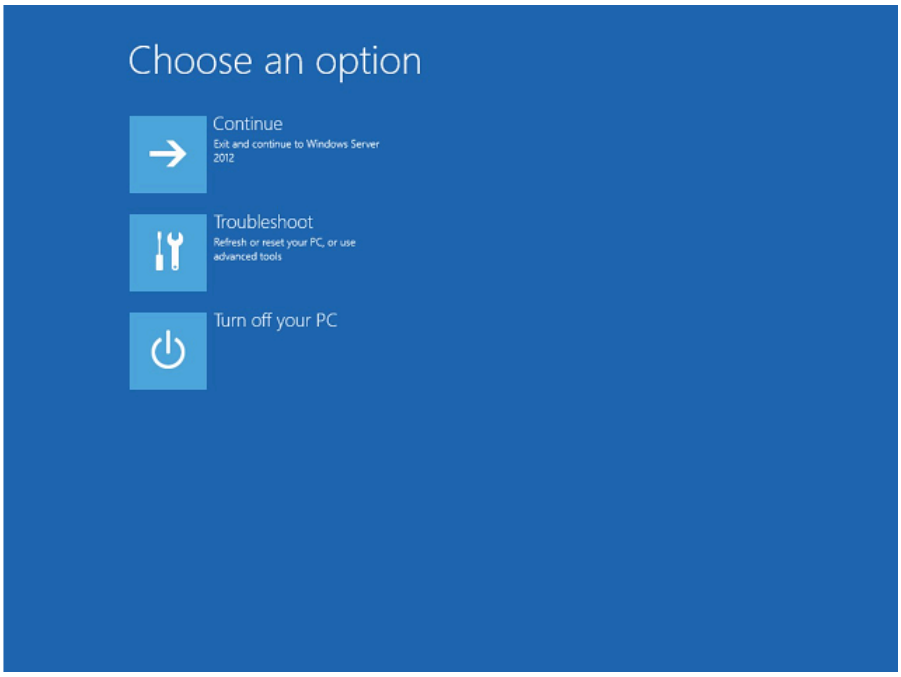
- Windows 업데이트
- 보안 소프트웨어 검사
- 사용자 지정 시작 스크립트
- 작업 스케줄러

자세한 내용은 Amazon CloudWatch 사용 설명서의 특정 리소스에 대한 통계 얻기를 참조하세요. 추가적인 문제 해결 팁은 Windows 시작 직후 높은 CPU 사용량(Windows 인스턴스만 해당) 섹션을 참조하세요.

복구 콘솔 화면

콘솔 스크린샷 서비스에서 다음을 반환했습니다.





`bootstatuspolicy`가 `ignoreallfailures`로 설정되어 있지 않은 경우 운영 체제가 복구 콘솔로 부팅되고 이 상태로 고착될 수 있습니다. 다음 절차에 따라 `bootstatuspolicy` 구성을 `ignoreallfailures`로 변경합니다.

기본적으로 AWS에서 제공하는 퍼블릭 Windows AMI에 대한 정책 구성은 `ignoreallfailures`로 설정됩니다.

1. 연결할 수 없는 인스턴스를 중지합니다.
2. 루트 볼륨의 스냅샷을 생성합니다. 루트 볼륨은 `/dev/sda1`로서 인스턴스에 연결됩니다.

연결할 수 없는 인스턴스에서 루트 볼륨을 분리하고, 볼륨의 스냅샷을 생성하거나 볼륨에서 AMI를 만들고, 이를 보조 볼륨과 동일한 가용 영역에 있는 다른 인스턴스에 연결합니다.

> **Warning**
>
> 임시 인스턴스와 원본 인스턴스가 동일한 AMI를 사용하여 시작되는 경우 추가 단계를 수행해야 합니다. 그렇지 않으면 디스크 서명 충돌로 인해 루트 볼륨 복원 후 원본 인스턴스를 부팅할 수 없습니다. 동일한 AMI를 사용하여 임시 인스턴스를 생성해야 하는 경우 디스크 서명 충돌을 피하기 위해 디스크 서명 충돌의 단계를 완료하세요.
> 또는 임시 인스턴스에 대한 다른 AMI를 선택합니다. 예를 들어 원본 인스턴스에서 Windows Server 2016용 AMI를 사용할 경우 Windows Server 2019용 AMI를 사용하여 임시 인스턴스를 시작합니다.

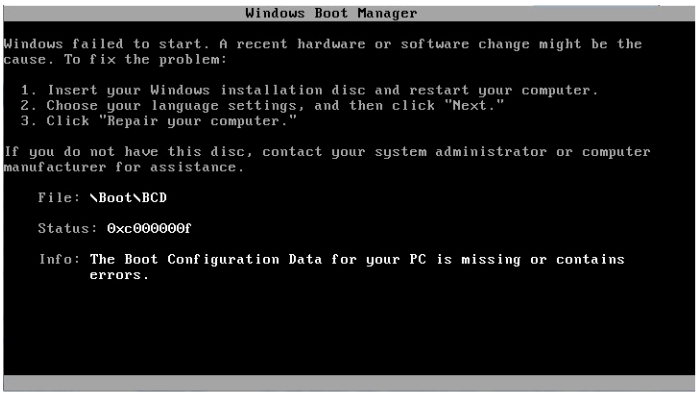
3. 인스턴스에 로그인하고 명령 프롬프트에서 다음 명령을 실행하여 `bootstatuspolicy` 구성을 `ignoreallfailures`로 변경합니다.

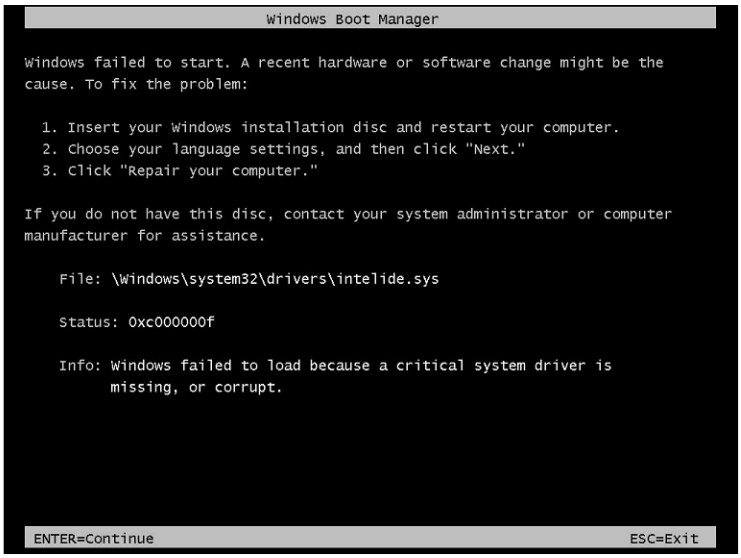
```
bcdedit /store Drive Letter:\boot\bcd /set {default} bootstatuspolicy
 ignoreallfailures
```

4. 볼륨을 연결하지 못했던 인스턴스에 다시 연결하고 인스턴스를 다시 시작합니다.

Windows 부팅 관리자 화면

콘솔 스크린샷 서비스에서 다음을 반환했습니다.

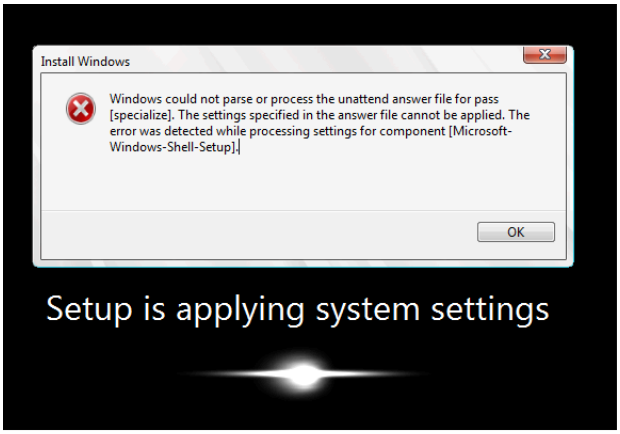


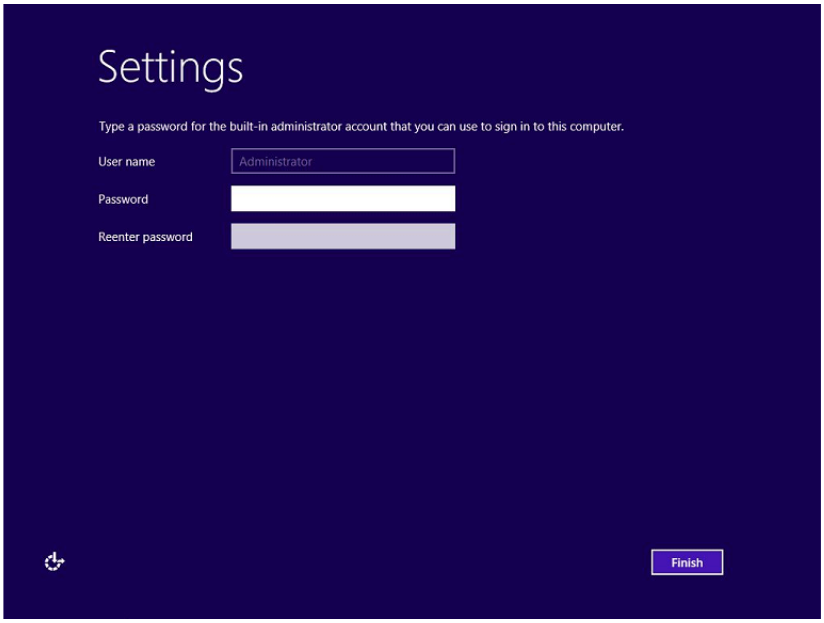


운영 체제의 시스템 파일 및/또는 레지스트리에서 치명적 손상이 발생했습니다. 인스턴스가 이 상태로 고착되면 최근 백업 AMI에서 인스턴스를 복구하거나 대체 인스턴스를 시작해야 합니다. 인스턴스 상의 데이터에 액세스할 필요가 있는 경우에는 연결할 수 없는 인스턴스에서 모든 루트 볼륨을 분리하고, 이들 볼륨의 스냅샷을 생성하거나 볼륨에서 AMI를 만들고, 이들을 보조 볼륨과 동일한 가용 영역에 있는 다른 인스턴스에 연결합니다.

Sysprep 화면

콘솔 스크린샷 서비스에서 다음을 반환했습니다.

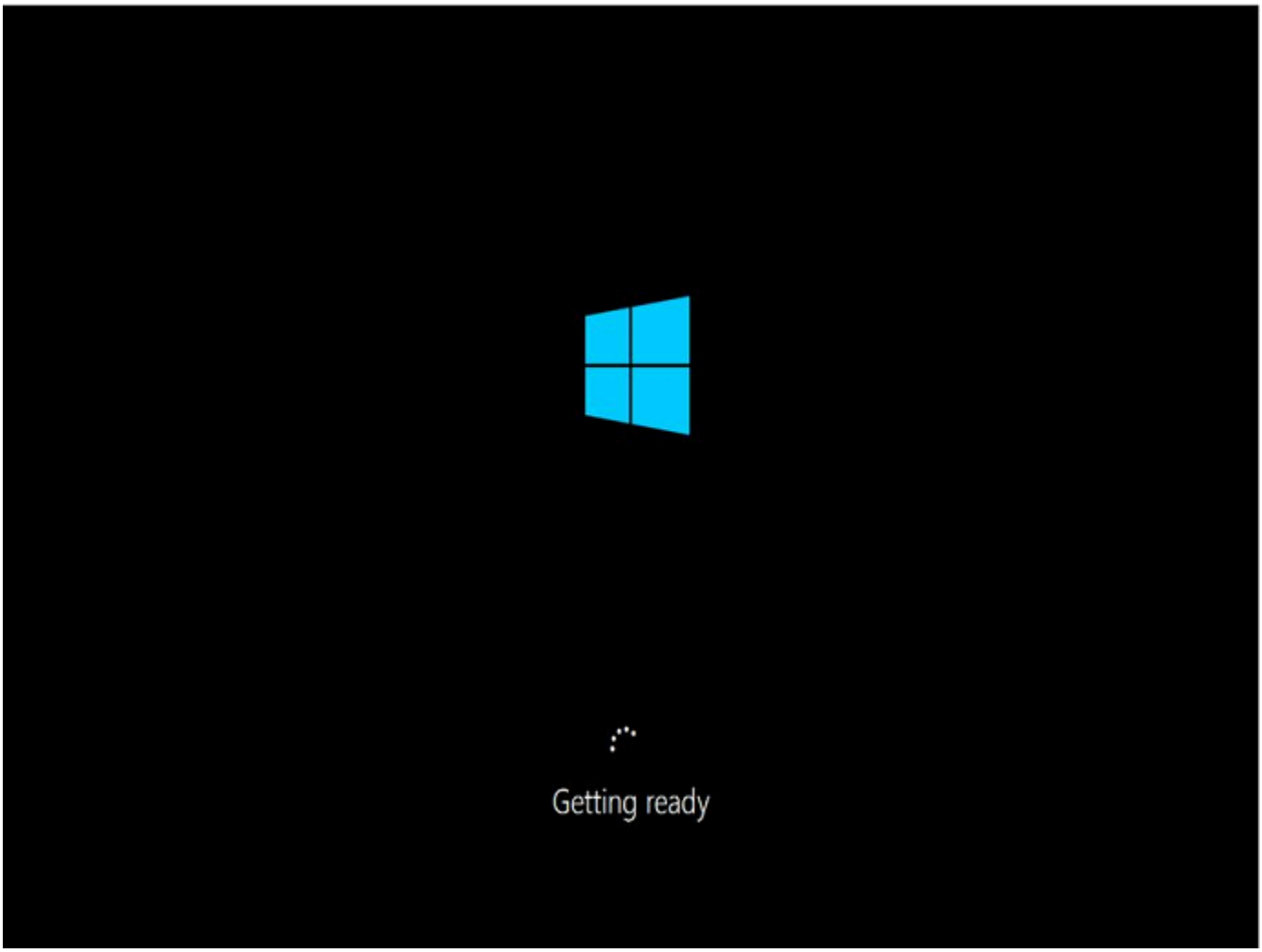




Sysprep 호출에 EC2Config Service를 사용하지 않았거나 Sysprep 실행 중에 운영 체제에 장애가 발생한 경우 이 화면이 나타날 수 있습니다. EC2Rescue를 사용하여 암호를 재설정할 수 있습니다. 그렇지 않으면 Windows Sysprep을 사용하여 Amazon EC2 AMI 생성 단원을 참조하세요.

준비 화면

콘솔 스크린샷 서비스에서 다음을 반환했습니다.



인스턴스 콘솔 스크린샷 서비스를 반복적으로 새로 고쳐 진행률 링이 돌고 있는지 확인합니다. 링이 돌고 있으면 운영 체제가 시작할 때까지 기다립니다. Amazon CloudWatch를 사용하여 인스턴스에서 CPUUtilization (Maximum) 측정치를 확인하여 운영 체제가 활성 상태인지 알아볼 수도 있습니다. 진행률 링이 돌고 있지 않으면 인스턴스가 부팅 프로세스에서 더 진행되지 못하고 중단될 수 있습니다. 인스턴스를 재부팅합니다. 다시 부팅했는데도 이 문제가 해결되지 않으면 최근 백업 AMI에서 인스턴스를 복구하거나 대체 인스턴스를 시작하세요. 인스턴스 상의 데이터에 액세스할 필요가 있는 경우에는 연결할 수 없는 인스턴스에서 루트 볼륨을 분리하고, 볼륨의 스냅샷을 생성하거나 볼륨에서 AMI를 만듭니다. 그런 다음, 보조 볼륨과 동일한 가용 영역에 있는 다른 인스턴스에 연결합니다.

Windows 업데이트 화면

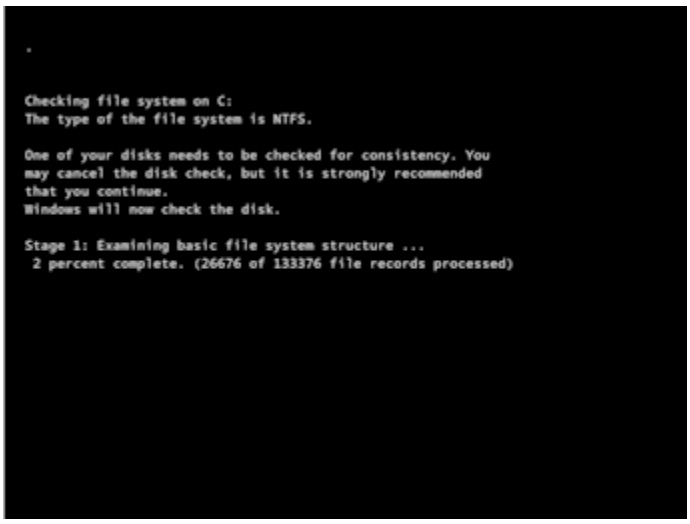
콘솔 스크린샷 서비스에서 다음을 반환했습니다.

Windows 업데이트 프로세스에서는 레지스트리가 업데이트됩니다. 업데이트를 마칠 때까지 기다리세요. 업데이트 중에 다시 부팅하거나 인스턴스를 중지하면 데이터가 손상될 수 있으므로 그렇게 하지 마세요.

> **Note**
>
> Windows 업데이트 프로세스는 업데이트 중에 서버의 리소스를 소비할 수 있습니다. 이 문제가 자주 발생하면 더 빠른 인스턴스 유형과 더 빠른 EBS 볼륨의 사용을 고려해 보세요.

Chkdsk

콘솔 스크린샷 서비스에서 다음을 반환했습니다.



Windows는 드라이브에서 chkdsk 시스템 도구를 실행하여 파일 시스템 레지스트리를 확인하고 논리적 파일 시스템 오류를 수정합니다. 프로세스가 완료될 때까지 기다립니다.

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 Amazon EC2 인스턴스 중지 및 시작 섹션을 참조하세요.

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.

인스턴스가 오프라인으로 표시되고 예기치 않게 재부팅됨

인스턴스가 오프라인으로 표시되고 예기치 않게 재부팅된 경우 자동 인스턴스 복구가 수행되었을 수 있습니다. 이는 AWS가 기본 하드웨어 또는 소프트웨어 문제로 인해 인스턴스를 사용할 수 없음을 감지하고 인스턴스에서 간소화된 자동 복구 또는 CloudWatch 작업 기반 복구가 활성화된 경우에 발생합니다.

복구 프로세스 중에는 AWS가 인스턴스를 다른 하드웨어로 마이그레이션하여 인스턴스의 가용성을 복원하려고 시도합니다. 인스턴스에 자동 인스턴스 복구가 발생했는지 확인하려면 자동 인스턴스 복구가 발생했는지 확인 섹션을 참조하세요.

> **Note**
>
> 워크로드 또는 애플리케이션이 응답하지 않는 경우 인스턴스에서 실행 중인지 확인합니다. 그렇지 않은 경우 수동으로 시작하세요. 향후 이 문제를 방지하려면 인스턴스 복구 후 워크로드 또는 애플리케이션이 제대로 작동하는지 확인하기 위한 복구 계획을 구현합니다.

Amazon EC2 Linux 인스턴스 연결 문제 해결

다음 정보와 일반적인 오류는 Linux 인스턴스 연결 문제를 해결하는 데 도움이 될 수 있습니다.

연결 문제

- 연결 문제의 일반적인 원인
- 인스턴스 연결 중 오류 발생: 연결 시간 초과
- 오류: 키를 로드할 수 없습니다... 예상: 모든 개인 키
- 오류: 서버에서 사용자 키를 인식하지 못함
- 오류: 사용 권한이 거부되었거나 [인스턴스] 포트 22에 의해 연결이 닫힘
- 오류: 보호되지 않는 프라이빗 키 파일
- 오류: 프라이빗 키는 '-----BEGIN RSA PRIVATE KEY-----'로 시작하고 '-----END RSA PRIVATE KEY-----'로 끝나야 합니다.
- 오류: 호스트 키 확인 실패
- 오류: 서버에서 키 거부 또는 지원되는 인증 방법이 없음
- 인스턴스를 ping할 수 없음
- 오류: 서버에서 예기치 않게 네트워크 연결을 차단함
- 오류: EC2 Instance Connect에 대한 호스트 키 유효성 검사 실패
- EC2 Instance Connect를 사용하여 Ubuntu 인스턴스에 연결할 수 없음
- 프라이빗 키를 분실했습니다. 인스턴스에 연결하려면 어떻게 해야 하나요?

연결 문제의 일반적인 원인

다음 태스크를 정확하게 수행했는지 확인하여 인스턴스 연결 문제 해결을 시작하는 것이 좋습니다.

인스턴스의 사용자 이름 확인

사용자 계정의 사용자 이름 또는 인스턴스를 시작하는 데 사용한 AMI의 기본 사용자 이름을 사용하여 인스턴스에 연결할 수 있습니다.

- 사용자 계정의 사용자 이름을 가져옵니다.

  사용자 계정을 생성하는 방법에 대한 자세한 내용은 Amazon EC2 Linux 인스턴스에서 시스템 사용자 관리 섹션을 참조하세요.
- 인스턴스를 시작하는 데 사용한 AMI의 기본 사용자 이름을 가져옵니다.

인스턴스를 시작하는 데 사용된 AMI	기본 사용자 이름
Amazon Linux	`ec2-user`
CentOS	`centos` 또는 `ec2-user`
Debian	`admin`
Fedora	`fedora` 또는 `ec2-user`
RHEL	`ec2-user` 또는 `root`
SUSE	`ec2-user` 또는 `root`
Ubuntu	`ubuntu`
Oracle	`ec2-user`
Bitnami	`bitnami`
Rocky Linux	`rocky`
기타	AMI 제공업체에 문의

보안 그룹 규칙이 트래픽을 허용하는지 확인

인스턴스와 연관된 보안 그룹이 IP 주소로부터 들어오는 SSH 트래픽을 허용하는지 확인하세요. VPC의 기본 보안 그룹은 기본적으로 수신 SSH 트래픽을 허용하지 않습니다. 인스턴스 시작 마법사가 생성한 보안 그룹은 기본적으로 SSH 트래픽을 활성화합니다. Linux 인스턴스에 인바운드 SSH 트래픽에 대한 규칙을 추가하는 단계는 컴퓨터에서 인스턴스 연결에 대한 규칙 섹션을 참조하세요. 확인 단계는 인스턴스 연결 중 오류 발생: 연결 시간 초과 섹션을 참조하세요.

인스턴스가 준비되었는지 확인

인스턴스를 시작한 후, 인스턴스가 연결 요청을 수락할 준비가 될 때까지 몇 분 정도 걸릴 수 있습니다. 인스턴스가 실행 중이고 상태 검사를 통과했는지 확인합니다.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택한 다음 인스턴스를 선택합니다.
3. 다음을 확인합니다.
   a. [인스턴스 상태(Instance state)] 열에서 인스턴스가 `running` 상태인지 확인합니다.
   b. [상태 확인(Status Check)] 열에서 인스턴스가 두 가지 상태 확인을 통과했는지 확인합니다.

연결에 필요한 모든 사전 조건을 충족했는지 확인

연결에 필요한 모든 정보가 있는지 확인합니다. 자세한 내용은 SSH를 사용하여 Linux 인스턴스에 연결 단원을 참조하십시오.

Linux 또는 macOS X에서 연결

로컬 컴퓨터 운영 체제가 Linux 또는 macOS X인 경우 Linux 인스턴스에 연결하기 위한 특정 사전 조건을 확인하세요.

- SSH 클라이언트
- EC2 Instance Connect
- AWS Systems Manager Session Manager

Windows에서 연결

로컬 컴퓨터 운영 체제가 Windows인 경우 Linux 인스턴스에 연결하기 위한 특정 사전 조건을 확인하세요.

- OpenSSH
- PuTTY
- AWS Systems Manager Session Manager
- Windows Subsystem for Linux

인스턴스가 관리형 인스턴스인지 확인

관리형 인스턴스에 대한 사용자 시작 연결은 허용되지 않습니다. 인스턴스가 관리되는지 확인하려면 인스턴스의 관리형 필드를 찾습니다. 값이 true이면 관리형 인스턴스입니다. 자세한 내용은 Amazon EC2 관리형 인스턴스 단원을 참조하십시오.

인스턴스 연결 중 오류 발생: 연결 시간 초과

인스턴스에 연결하려고 할 때 `Network error: Connection timed out` 또는 `Error connecting to [instance], reason: -> Connection timed out: connect`라는 오류 메시지가 표시되면 다음과 같이 합니다.

보안 그룹 규칙을 확인합니다.

로컬 컴퓨터의 퍼블릭 IPv4 주소에서 들어오는 인바운드 트래픽을 적절한 포트로 허용하는 보안 그룹 규칙이 필요합니다.

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스(Instances)를 선택한 다음 인스턴스를 선택합니다.
3. 콘솔 페이지 아래쪽에 있는 보안(Security) 탭의 인바운드 규칙(Inbound rules)에서 선택한 인스턴스에 적용되는 규칙 목록을 확인합니다. 로컬 컴퓨터에서 포트 22(SSH)로의 트래픽을 허용하는 규칙이 있는지 확인합니다.

   보안 그룹에 로컬 컴퓨터에서 들어오는 인바운드 트래픽을 허용하는 규칙이 없을 경우 보안 그룹에 규칙을 추가합니다. 자세한 내용은 컴퓨터에서 인스턴스 연결에 대한 규칙 단원을 참조하십시오.
4. 인바운드 트래픽을 허용하는 규칙은 소스(Source) 필드를 확인합니다. 값이 단일 IP 주소이고 IP 주소가 고정적이지 않으면 컴퓨터를 다시 시작할 때마다 새 IP 주소가 할당됩니다. 그러면 컴퓨터의 IP 주소 트래픽이 규칙에 포함되지 않습니다. 컴퓨터가 회사 네트워크에 있거나 인터넷 서비스 제공업체(ISP)를 통해 연결하는 경우 IP 주소가 고정되지 않거나 컴퓨터를 다시 시작할 때마다 컴퓨터 IP 주소가 동적으로 변경될 수 있습니다. 보안 그룹 규칙에서 소스(Source)에 단일 IP 주소를 지정하는 대신 로컬 컴퓨터에서 들어오는 인바운드 트래픽을 허용하려면 클라이언트 컴퓨터에서 사용하는 IP 주소의 범위를 지정합니다.

   보안 그룹 규칙에 대한 자세한 내용은 Amazon VPC 사용 설명서의 보안 그룹 규칙을 참조하세요.

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. [네트워킹(Networking)] 탭에서 [VPC ID] 및 [서브넷 ID(Subnet ID)]의 값을 기록합니다.
4. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
5. 탐색 창에서 [Internet Gateways]를 선택합니다. VPC에 인터넷 게이트웨이가 연결되어 있는지 확인합니다. 그렇지 않은 경우 [인터넷 게이트웨이 생성(Create internet gateway)]을 선택하고 인터넷 게이트웨이의 이름을 입력한 다음 [인터넷 게이트웨이 생성(Create internet gateway)]을 선택합니다. 그런 다음 생성한 인터넷 게이트웨이에 대해 [작업(Actions)], [VPC에 연결(Attach to VPC)]을 선택하고, VPC를 선택한 다음 [인터넷 게이트웨이 연결(Attach internet gateway)]을 선택하여 VPC에 연결합니다.
6. 탐색 창에서 서브넷을 선택한 후 해당 서브넷을 선택합니다.
7. [라우팅 테이블(Route table)] 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 포트 22의 로컬 IP 주소에서 수신되는 인바운드 SSH 트래픽을 허용해야 합니다. 또한 임시 포트(1024~65535)로의 아웃바운드 트래픽도 허용해야 합니다.

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Subnets를 선택합니다.
3. 서브넷을 선택합니다.
4. [네트워크 ACL(Network ACL)] 탭의 [인바운드 규칙(Inbound rules)]에서 규칙이 컴퓨터의 필수 포트에서 트래픽을 허용하는지 확인합니다. 허용하지 않을 경우 트래픽을 차단하는 규칙을 삭제하거나 수정합니다.

5. [아웃바운드 규칙(Outbound rules)] 탭에서 규칙이 임시 포트에서 컴퓨터로의 트래픽을 허용하는지 확인합니다. 허용하지 않을 경우 트래픽을 차단하는 규칙을 삭제하거나 수정합니다.

컴퓨터가 회사 네트워크에 있는 경우

네트워크 관리자에게 내부 방화벽이 포트 22의 로컬 컴퓨터에서 수신되는 인바운드 및 아웃바운드 트래픽을 허용하는지 여부를 문의합니다.

컴퓨터에 방화벽이 있을 경우 방화벽이 포트 22의 컴퓨터에서 수신되는 인바운드 및 아웃바운드 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 섹션을 참조하세요.

인스턴스에서 CPU 부하를 확인합니다. 서버 과부하가 발생했을 수 있습니다.

AWS에서는 Amazon CloudWatch 지표 및 인스턴스 상태 등과 같은 데이터를 자동으로 제공하므로, 이러한 정보를 사용하여 인스턴스에 대한 CPU 로드를 확인하고 필요할 경우 로드 처리 방법을 조정할 수 있습니다. 자세한 내용은 CloudWatch를 사용하여 인스턴스 모니터링 섹션을 참조하세요.

- 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 Amazon EC2 인스턴스 유형 변경 섹션을 참조하세요.

IPv6 주소를 사용해 인스턴스에 연결하려면 다음을 확인합니다.

- 서브넷은 IPv6 트래픽(`::/0`)을 인터넷 게이트웨이로 이어주는 경로가 있는 라우팅 테이블과 연결되어야 합니다.
- 보안 그룹 규칙이 포트 22의 로컬 IPv6 주소에서 수신되는 인바운드 트래픽을 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 자동으로 인식되지 않음). 자세한 내용을 알아보려면 Amazon VPC 사용 설명서의 인스턴스에서 IPv6 구성하기를 참조하세요.
- 로컬 컴퓨터에 IPv6 주소가 있고 IPv6를 사용하도록 컴퓨터를 구성해야 합니다.

오류: 키를 로드할 수 없습니다... 예상: 모든 개인 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 사용하여 키 페어 생성 단원을 참조하십시오.

   > **Note**
   >
   > 또는 타사 도구를 사용해 새 키 페어를 만들 수 있습니다. 자세한 내용은 서드 파티 도구를 사용하여 키 페어를 생성하고 Amazon EC2로 퍼블릭 키 가져오기 단원을 참조하십시오.

2. 새 키 페어를 인스턴스에 추가합니다. 자세한 내용은 프라이빗 키를 분실했습니다. 인스턴스에 연결하려면 어떻게 해야 하나요? 섹션을 참조하세요.
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 path/key-pair-name.pem instance-user-
name@ec2-203-0-113-25.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 Linux 인스턴스에 연결 섹션을 참조하세요.

> **Note**
>
> PuTTYgen에서 프라이빗 키 파일을 불러온 후 생성이 아니라 Save Private Key(프라이빗 키 저장)를 선택합니다.

- AMI에 적합한 사용자 이름을 사용하여 연결하고 있는지 확인합니다. PuTTY 구성 창에서 호스트 이름 상자에 사용자 이름을 입력합니다.

인스턴스를 시작하는 데 사용된 AMI	기본 사용자 이름
Amazon Linux	`ec2-user`
CentOS	`centos` 또는 `ec2-user`
Debian	`admin`
Fedora	`fedora` 또는 `ec2-user`
RHEL	`ec2-user` 또는 `root`
SUSE	`ec2-user` 또는 `root`
Ubuntu	`ubuntu`
Oracle	`ec2-user`
Bitnami	`bitnami`
Rocky Linux	`rocky`
기타	AMI 제공업체에 문의

- 해당 포트로의 인바운드 트래픽을 허용할 인바운드 보안 그룹 규칙이 있는지 확인합니다. 자세한 내용은 컴퓨터에서 인스턴스 연결에 대한 규칙 단원을 참조하십시오.

오류: 사용 권한이 거부되었거나 [인스턴스] 포트 22에 의해 연결이 닫힘

SSH를 사용하여 인스턴스에 연결할 때 `Host key not found in [directory]`, `Permission denied (publickey)`, `Authentication failed, permission denied`, 또는 `Connection closed by [instance] port 22` 오류 중 하나가 발생하는 경우 AMI에 적합한 사용자 이름을 사용하여 연결하고 있는지, 그리고 인스턴스에 대한 올바른 프라이빗 키(`.pem`)) 파일을 지정했는지 확인합니다.

적합한 사용자 이름은 다음과 같습니다.

인스턴스를 시작하는 데 사용된 AMI	기본 사용자 이름
Amazon Linux	`ec2-user`
CentOS	`centos` 또는 `ec2-user`
Debian	`admin`
Fedora	`fedora` 또는 `ec2-user`
RHEL	`ec2-user` 또는 `root`
SUSE	`ec2-user` 또는 `root`
Ubuntu	`ubuntu`
Oracle	`ec2-user`
Bitnami	`bitnami`
Rocky Linux	`rocky`
기타	AMI 제공업체에 문의

예를 들어 SSH 클라이언트를 사용하여 Amazon Linux 인스턴스에 연결하려면 다음 명령을 사용합니다.

```
ssh -i /path/key-pair-name.pem instance-user-
name@ec2-203-0-113-25.compute-1.amazonaws.com
```

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. 탐색 창에서 [인스턴스(Instances)]를 선택한 다음 인스턴스를 선택합니다.
3. [세부 정보(Details)] 탭의 [인스턴스 세부 정보(Instance details)]에서 [키 페어 이름(Key pair name)]의 값을 확인합니다.
4. 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 분실했습니다. 인스턴스에 연결하려면 어떻게 해야 하나요? 섹션을 참조하세요.

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/instance-user-name/.ssh/authorized_keys`에 대한 권한은 소유자로만 제한되어야 합니다.

인스턴스에서 권한을 확인하려면

1. 인스턴스를 중지하고 루트 볼륨을 분리합니다. 자세한 내용은 Amazon EC2 인스턴스 중지 및 시작 단원을 참조하십시오.
2. 현재의 인스턴스와 동일한 가용 영역에서 임시 인스턴스를 시작하고(현재의 인스턴스에 사용한 것과 비슷하거나 동일한 AMI 사용) 루트 볼륨을 임시 인스턴스에 연결합니다.
3. 임시 인스턴스에 연결하고 마운트 지점을 생성한 후 연결한 볼륨을 마운트합니다.
4. 임시 인스턴스에서 연결된 볼륨의 /home/*instance-user-name*/ 디렉토리의 권한을 확인합니다. 필요하다면 다음과 같이 권한을 조정합니다.

```
[ec2-user ~]$ chmod 600 mount_point/home/instance-user-name/.ssh/authorized_keys
```

```
[ec2-user ~]$ chmod 700 mount_point/home/instance-user-name/.ssh
```

```
[ec2-user ~]$ chmod 700 mount_point/home/instance-user-name
```

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.

macOS 또는 Linux에서 연결하는 경우 프라이빗 키 파일 경로를 대체하는 다음 명령을 실행하여 이 오류를 해결합니다.

```
[ec2-user ~]$ chmod 0400 .ssh/my_private_key.pem
```

Windows에서 Linux 인스턴스에 연결하는 경우 로컬 컴퓨터에서 다음 단계를 수행합니다.

1. .pem 파일로 이동합니다.
2. .pem 파일을 마우스 오른쪽 버튼으로 클릭하고 [속성(Properties)]을 선택합니다.
3. 보안 탭을 선택합니다.
4. [고급(Advanced)]을 선택합니다.
5. 파일의 소유자인지 확인합니다. 그렇지 않은 경우 소유자를 사용자 이름으로 변경합니다.
6. [상속 비활성화(Disable inheritance)] 및 [이 객체에서 상속된 모든 권한 제거(Remove all inherited permissions from this object)]를 선택합니다.
7. [추가(Add)], [보안 주체 선택(Select a principal)]을 차례로 선택하고 사용자 이름을 입력한 다음 [확인(OK)]을 선택합니다.
8. [권한 항목(Permission Entry)] 창에서 [읽기(Read)] 권한을 부여하고 [확인(OK)]을 선택합니다.
9. 적용(Apply)을 클릭하여 모든 설정이 저장되었는지 확인합니다.
10. [확인(OK)]을 선택하여 [고급 보안 설정(Advanced Security Settings)] 창을 닫습니다.
11. [확인(OK)]을 선택하여 [속성(Properties)] 창을 닫습니다.
12. Windows에서 SSH를 사용하여 Linux 인스턴스에 연결할 수 있어야 합니다.

Windows 명령 프롬프트 창에서 다음 명령을 실행합니다.

1. 명령 프롬프트에서 .pem 파일의 파일 경로 위치로 이동합니다.
2. 다음 명령을 실행하여 명시적 사용 권한을 재설정하고 제거합니다.

```
icacls.exe $path /reset
```

3. 다음 명령을 실행하여 현재 사용자에게 읽기 권한을 부여합니다.

```
icacls.exe $path /GRANT:R "$($env:USERNAME):(R)"
```

4. 다음 명령을 실행하여 상속을 비활성화하고 상속된 사용 권한을 제거합니다.

```
icacls.exe $path /inheritance:r
```

5. Windows에서 SSH를 사용하여 Linux 인스턴스에 연결할 수 있어야 합니다.

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: 호스트 키 확인 실패

이 오류는 known_hosts 파일의 인스턴스 및 클라이언트에 저장된 호스트 키가 불일치하는 경우에 발생합니다. 예를 들어 한 퍼블릭 IP 주소를 사용하여 인스턴스에 연결한 다음 다른 퍼블릭 IP 주소를 사용하여 다시 연결하려고 하면 불일치가 발생할 수 있습니다. 탄력적 IP 주소를 추가하거나 제거한 후 이 작업으로 인해 인스턴스의 퍼블릭 IP 주소가 변경되므로 이 문제가 발생할 수 있습니다.

이 오류를 해결하려면 먼저 호스트 키 또는 인스턴스의 네트워크 구성에 대해 예상되는 변경 사항이 있는지 확인합니다. 인스턴스에 연결하기 전에 호스트 지문을 확인해야 할 수도 있습니다. 인스턴스에 연

결한 후 known_hosts 파일에서 이전 호스트 키를 제거할 수 있습니다. 관련 지침은 인스턴스에서 사용 중인 Linux 배포 설명서를 참조하세요.

오류: 서버에서 키 거부 또는 지원되는 인증 방법이 없음

PuTTY를 사용하여 인스턴스에 연결하고 Error: Server refused our key 또는 Error: No supported authentication methods available 중 하나가 발생하면 AMI에 적합한 사용자 이름으로 연결했는지 확인하십시오. PuTTY 구성 창에서 사용자 이름에 사용자 이름을 입력합니다.

적합한 사용자 이름은 다음과 같습니다.

인스턴스를 시작하는 데 사용된 AMI	기본 사용자 이름
Amazon Linux	ec2-user
CentOS	centos 또는 ec2-user
Debian	admin
Fedora	fedora 또는 ec2-user
RHEL	ec2-user 또는 root
SUSE	ec2-user 또는 root
Ubuntu	ubuntu
Oracle	ec2-user
Bitnami	bitnami
Rocky Linux	rocky
기타	AMI 제공업체에 문의

다음 사항도 확인해야 합니다.

- PuTTY의 최신 버전을 사용하고 있습니다. 자세한 내용은 PuTTY 웹 페이지를 참조하세요.
- 프라이빗 키(.pem) 파일이 PuTTY(.ppk)에서 인식하는 형식으로 변환되었습니다. 프라이빗 키 변환에 대한 자세한 내용은 PuTTY를 사용하여 Linux 인스턴스에 연결 섹션을 참조하세요.

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.

`Ping` 명령은 방화벽에 의해 차단되거나 네트워크 지연 또는 하드웨어 문제로 인해 시간 초과될 수도 있습니다. 추가 문제 해결에 대한 도움말은 로컬 네트워크 또는 시스템 관리자에게 문의해야 합니다.

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.

오류: EC2 Instance Connect에 대한 호스트 키 유효성 검사 실패

인스턴스 호스트 키를 교체할 경우 새 호스트 키가 AWS의 신뢰할 수 있는 호스트 키 데이터베이스에 자동으로 업로드되지 않습니다. 이로 인해 EC2 Instance Connect 브라우저 기반 클라이언트를 사용하여 인스턴스에 연결하려고 하면 호스트 키 유효성 검사가 실패하고 인스턴스에 연결할 수 없게 됩니다.

이 오류를 해결하려면 새 호스트 키를 EC2 Instance Connect에 업로드하는 인스턴스에서 `eic_harvest_hostkeys` 스크립트를 실행해야 합니다. 스크립트는 `/opt/aws/bin/`(Amazon Linux 2 인스턴스) 및 `/usr/share/ec2-instance-connect/`(Ubuntu 인스턴스)에 있습니다.

Amazon Linux 2

Amazon Linux 2 인스턴스에서 호스트 키 유효성 검사 실패 오류를 해결하려면

1. SSH로 인스턴스에 연결합니다.

   EC2 Instance Connect CLI를 사용하여 연결하거나, 인스턴스를 시작할 때 인스턴스에 할당된 SSH 키 페어와 인스턴스를 시작할 때 사용한 AMI의 기본 사용자 이름을 사용하여 연결할 수 있습니다. Amazon Linux 2의 경우, 기본 사용자 이름은 `ec2-user`입니다.

예를 들어 인스턴스가 Amazon Linux 2를 사용하여 시작되었고 인스턴스의 퍼블릭 DNS 이름이 ec2-a-b-c-d.us-west-2.compute.amazonaws.com이고 키 페어가 my_ec2_private_key.pem인 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 SSH 클라이언트를 사용하여 Linux 인스턴스에 연결 주제를 참조하십시오.

2. 다음 폴더로 이동합니다.

```
[ec2-user ~]$ cd /opt/aws/bin/
```

3. 인스턴스에서 다음 명령을 실행합니다.

```
[ec2-user ~]$ ./eic_harvest_hostkeys
```

호출이 성공하면 출력이 표시되지 않습니다.

이제 EC2 Instance Connect 브라우저 기반 클라이언트를 사용하여 인스턴스에 연결할 수 있습니다.

Ubuntu

Ubuntu 인스턴스에서 호스트 키 유효성 검사 실패 오류를 해결하려면

1. SSH로 인스턴스에 연결합니다.

EC2 Instance Connect CLI를 사용하여 연결하거나, 인스턴스를 시작할 때 인스턴스에 할당된 SSH 키 페어와 인스턴스를 시작할 때 사용한 AMI의 기본 사용자 이름을 사용하여 연결할 수 있습니다. Ubuntu의 기본 사용자 이름은 ubuntu입니다.

예를 들어 인스턴스가 Ubuntu를 사용하여 시작되었고 인스턴스의 퍼블릭 DNS 이름이 ec2-a-b-c-d.us-west-2.compute.amazonaws.com이고 키 페어가 my_ec2_private_key.pem인 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 SSH 클라이언트를 사용하여 Linux 인스턴스에 연결 주제를 참조하십시오.

2. 다음 폴더로 이동합니다.

```
[ec2-user ~]$ cd /usr/share/ec2-instance-connect/
```

3. 인스턴스에서 다음 명령을 실행합니다.

```
[ec2-user ~]$ ./eic_harvest_hostkeys
```

호출이 성공하면 출력이 표시되지 않습니다.

이제 EC2 Instance Connect 브라우저 기반 클라이언트를 사용하여 인스턴스에 연결할 수 있습니다.

EC2 Instance Connect를 사용하여 Ubuntu 인스턴스에 연결할 수 없음

EC2 Instance Connect를 사용하여 Ubuntu 인스턴스에 연결하고 연결을 시도할 때 오류가 발생하는 경우, 다음 정보를 사용하여 문제를 해결할 수 있습니다.

가능한 원인

인스턴스의 `ec2-instance-connect` 패키지가 최신 버전이 아닙니다.

솔루션

인스턴스의 `ec2-instance-connect` 패키지를 다음과 같이 최신 버전으로 업데이트합니다.

1. EC2 Instance Connect 이외의 방법을 사용하여 인스턴스에 연결합니다.
2. 인스턴스에 다음 명령을 사용하여 `ec2-instance-connect` 패키지를 최신 버전으로 업데이트합니다.

```
apt update && apt upgrade
```

프라이빗 키를 분실했습니다. 인스턴스에 연결하려면 어떻게 해야 하나요?

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 Amazon EC2 인스턴스 상태 변경에서 확인하십시오.

이 절차는 EBS 루트 볼륨이 있는 인스턴스에만 지원됩니다. 루트 디바이스가 인스턴스 스토어 볼륨인 경우 이 절차를 사용하여 인스턴스에 대한 액세스 권한을 다시 얻을 수 없습니다. 인스턴스에 연결하려면 프라이빗 키가 있어야 합니다. 인스턴스의 루트 디바이스 유형을 확인하려면 Amazon EC2 콘솔을 열고 인스턴스를 선택한 다음 인스턴스를 선택하고 스토리지 탭을 선택한 다음 루트 디바이스 세부 정보 섹션에서 루트 디바이스 유형 값을 확인합니다.

이때 값은 `EBS` 또는 `INSTANCE-STORE`입니다.

다음 단계 외에도 프라이빗 키를 분실한 경우 Linux 인스턴스에 연결하는 다른 방법도 있습니다. 자세한 내용은 처음 시작한 후 SSH 키 페어를 분실한 경우 Amazon EC2 인스턴스에 연결하려면 어떻게 해야 합니까?를 참조하세요.

키 페어가 다른 EBS 지원 인스턴스에 연결하는 단계

- 1단계: 새 키 페어 생성
- 2단계: 원본 인스턴스와 루트 볼륨에 대한 정보 가져오기
- 3단계: 원본 인스턴스 중지
- 4단계: 임시 인스턴스 시작
- 5단계: 원본 인스턴스에서 루트 볼륨을 분리하고 임시 인스턴스에 연결
- 6단계: 임시 인스턴스에 마운트된 원본 볼륨의 authorized_keys에 새 퍼블릭 키 추가
- 7단계: 임시 인스턴스에서 원본 볼륨을 마운트 해제하고 분리한 다음 원본 인스턴스에 다시 연결
- 8단계: 새 키 페어를 사용하여 원본 인스턴스에 연결
- 9단계: 정리

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 대

한 자세한 내용은 Amazon EC2를 사용하여 키 페어 생성 또는 서드 파티 도구를 사용하여 키 페어를 생성하고 Amazon EC2로 퍼블릭 키 가져오기 단원을 참조하십시오.

2단계: 원본 인스턴스와 루트 볼륨에 대한 정보 가져오기

이 절차를 완료하는 데 필요한 다음 정보를 기록해 둡니다.

원래 인스턴스에 대한 정보를 가져오려면

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. [세부 정보(Details)] 탭에서 인스턴스 ID와 AMI ID를 기록합니다.
4. [네트워킹(Networking)] 탭에서 가용 영역을 기록합니다.
5. [스토리지(Storage)] 탭의 [루트 디바이스 이름(Root device name)] 아래에서 루트 볼륨의 디바이스 이름(예: `/dev/xvda`)을 기록합니다. 그런 다음 [블록 디바이스(Block devices)]에서 이 디바이스 이름을 찾아 볼륨 ID(예: vol-0a1234b5678c910de)를 기록합니다.

3단계: 원본 인스턴스 중지

인스턴스 상태, 인스턴스 중지를 차례로 선택합니다. 이 옵션이 비활성화되어 있으면 해당 인스턴스가 이미 중지되었거나 해당 루트 디바이스가 인스턴스 스토어 볼륨인 것입니다.

> **Warning**
>
> 인스턴스를 중지하면 인스턴스 스토어 볼륨의 데이터가 삭제됩니다. 인스턴스 스토어 볼륨의 데이터를 유지하려면 영구 스토리지에 백업하세요.

4단계: 임시 인스턴스 시작

임시 인스턴스를 실행합니다.

1. 탐색 창에서 Instances(인스턴스)를 선택한 후 Launch instances(인스턴스 시작)를 선택합니다.
2. 이름 및 태그(Name and tags) 섹션에서 이름(Name)에 임시(Temporary)를 입력합니다.
3. 애플리케이션 및 OS 이미지(Application and OS Images) 섹션에서 원본 인스턴스를 시작하는 데 사용한 것과 동일한 AMI를 선택합니다. 이 AMI가 표시되지 않는 경우 중지된 인스턴스에서 사용

가능한 AMI를 만들 수 있습니다. 자세한 내용은 Amazon EBS 지원 AMI 생성 단원을 참조하십시오.

4. 인스턴스 유형(Instance type) 섹션에서 기본 인스턴스 유형을 유지합니다.
5. 키 페어(Key pair) 섹션의 키 페어 이름(Key pair name)에서 사용할 기존 키 페어를 선택하거나 새로 하나 생성합니다.
6. 네트워크 설정(Network settings) 섹션에서 편집(Edit)을 선택한 다음 서브넷(Subnet)에 대해 원본 인스턴스와 동일한 가용 영역에 있는 서브넷을 선택합니다.
7. 요약(Summary) 패널에서 시작(Launch)을 선택합니다.

5단계: 원본 인스턴스에서 루트 볼륨을 분리하고 임시 인스턴스에 연결

1. 탐색 창에서 [볼륨(Volumes)]을 선택하고 원본 인스턴스에 대한 루트 디바이스 볼륨을 선택합니다(전 단계에서 기록한 볼륨 ID). Actions(작업), Detach volume(볼륨 분리)를 선택하고 Detach(분리)를 선택합니다. 볼륨이 `available` 상태가 될 때까지 기다리십시오. (새로 고침 아이콘을 클릭해야 할 수도 있습니다.)
2. 해당 볼륨을 선택한 상태에서 Actions(작업)을 선택한 후 Attach volume(볼륨 연결)을 선택합니다. 임시 인스턴스의 인스턴스 ID를 선택하고 Device name(디바이스 이름)에서 지정된 디바이스(예: `/dev/sdf`)를 기록한 후 Attach volume(볼륨 연결)을 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/sdf로 탑재된 디바이스가 인스턴스에서는 /dev/xvdf로 표시되기도 합니다. Red Hat 중 일부 버전(CentOS 등 변형 버전 포함)은 후행 문자가 4자씩 늘어나기도 하며, 이 경우 /dev/sd*f*가 /dev/xvd*k*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/xvdf는 파티션 볼륨이고, /dev/xvdg는 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. 디바이스 이름은 인스턴스에서 다르게 표시될 수 있습니다. 자세한 내용은 6단계의 note 섹션을 참조하십시오.

- Amazon Linux, Ubuntu 및 Debian

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

- Amazon Linux 2, CentOS, SUSE Linux 12, 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_keys를 임시 인스턴스에 대한authorized_keys의 새로운 퍼블릭 키로 업데이트합니다.

> **Important**
>
> 다음 예는 Amazon Linux 사용자 이름 ec2-user를 사용합니다. Ubuntu 인스턴스의 경우 ubuntu처럼 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/tempvol에서 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 사용하세요.

```
[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 탑재 해제한 볼륨): Amazon EC2 콘솔의 탐색 창에서 Volumes(볼륨)를 선택하고 원본 인스턴스의 루트 디바이스 볼륨을 선택하고(이전 단계에서 기록한 볼륨 ID 참조) Actions(작업), Detach volume(볼륨 분리)를 선택한 다음 Detach(분리)를 선택합니다. 볼륨이 `available` 상태가 될 때까지 기다리십시오. (새로 고침 아이콘을 클릭해야 할 수도 있습니다.)
3. 볼륨을 원본 인스턴스에 다시 연결: 볼륨을 선택한 상태에서 Actions(작업), Attach volume(볼륨 연결)을 선택합니다. 원본 인스턴스의 인스턴스 ID를 선택하고, 앞서 2단계에서 원래 루트 디바이스 연결을 위해 메모한 디바이스 이름(`/dev/sda1` 또는 `/dev/xvda`)을 지정한 뒤 Attach volume(볼륨 연결)을 선택합니다.

   > **Important**
   >
   > 원래 연결과 동일한 디바이스 이름을 지정하지 않으면 원본 인스턴스를 시작할 수 없습니다. Amazon EC2는 `sda1` 또는 `/dev/xvda`에서 루트 디바이스 볼륨을 찾습니다.

8단계: 새 키 페어를 사용하여 원본 인스턴스에 연결

원래 인스턴스를 선택하고 인스턴스 상태, 인스턴스 시작을 차례로 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 인스턴스 상태, 인스턴스 종료(삭제)를 차례로 선택합니다.

상태 확인에 실패한 Amazon EC2 Linux 인스턴스 문제 해결

다음 정보는 Linux 인스턴스에서 상태 확인에 실패하는 경우 문제를 해결하는 데 도움이 될 수 있습니다. 먼저 애플리케이션에 문제가 있는지 여부를 결정합니다. 인스턴스에서 애플리케이션이 예상대로 실행되고 있지 않은지 확인하면 상태 점검 정보와 시스템 로그를 검토합니다.

상태 확인이 실패하게 될 수 있는 문제의 예시는 Amazon EC2 인스턴스 상태 확인 섹션을 참조하세요.

목차

- 상태 점검 정보 검토
- 시스템 로그 검색
- Linux 인스턴스의 시스템 로그 오류 문제 해결
- 메모리 부족: 프로세스 중지
- ERROR: mmu_update failed(메모리 관리 업데이트 실패)
- I/O 오류(블록 디바이스 장애)
- I/O ERROR: neither local nor remote disk(분산된 블록 디바이스 손상)
- request_module: runaway loop modprobe(이전 Linux 버전에서 레거시 커널 modprobe 반복)
- "FATAL: kernel too old" 및 "fsck: No such file or directory while trying to open /dev"(커널과 AMI 불일치)
- "FATAL: Could not load /lib/modules" 또는 "BusyBox"(커널 모듈 누락)
- ERROR Invalid kernel(EC2 커널이 호환되지 않음)
- fsck: No such file or directory while trying to open... (파일 시스템을 찾을 수 없음)
- 파일 시스템 마운트 관련 일반 오류(마운트 실패)
- VFS: Unable to mount root fs on unknown-block(루트 파일 시스템 불일치)

- Error: Unable to determine major/minor number of root device... (루트 파일 시스템/디바이스 불일치)
- XENBUS: Device with no driver...
- ... days without being checked, check forced(파일 시스템 검사 필요)
- fsck died with exit status... (디바이스 누락)
- GRUB 프롬프트(grubdom>)
- 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)
- 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)
- XENBUS: Timeout connecting to devices(Xenbus 시간 초과)

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 인스턴스(Instances)를 선택한 다음 인스턴스를 선택합니다.
3. 상태 및 경보 탭을 선택하여 모든 시스템 상태 확인, 인스턴스 상태 확인, 연결된 EBS 상태 확인에 대한 개별 결과를 봅니다.

상태 확인이 실패한 경우 다음 옵션 중 하나를 시도할 수 있습니다.

- 상태 확인 실패에 대한 응답으로 인스턴스를 복구하기 위한 경보를 생성합니다. 자세한 내용은 인스턴스를 중지, 종료, 재부팅 또는 복구하는 경보 만들기 섹션을 참조하세요.
- (인스턴스 상태 확인) 인스턴스 유형을 Nitro 기반 인스턴스로 변경한 경우에는 필요한 ENA 및 NVMe 드라이버가 없는 인스턴스에서 마이그레이션하면 상태 확인이 실패합니다. 자세한 내용은 인스턴스 유형 변경을 위한 호환성 섹션을 참조하세요.
- EBS 지원 인스턴스의 경우, 인스턴스를 중지했다가 다시 시작합니다. 자세한 내용은 Amazon EC2 인스턴스 중지 및 시작 섹션을 참조하세요.
- 인스턴스 스토어 지원 인스턴스의 경우 해당 인스턴스를 종료한 후 대체 인스턴스를 시작합니다. 자세한 내용은 Amazon EC2 인스턴스 종료 섹션을 참조하세요.
- Amazon EC2에서 문제를 해결할 때까지 기다립니다.

- 지원에 문의하거나 AWS re:Post에 문제를 게시합니다.
- 인스턴스가 Auto Scaling 그룹에 있는 경우:
  - (시스템 상태 확인 및 인스턴스 상태 확인) 기본적으로 Amazon EC2 Auto Scaling은 자동으로 대체 인스턴스를 시작합니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Auto Scaling 그룹의 인스턴스 상태 확인 섹션을 참조하세요.
  - (연결된 EBS 상태 확인) 대체 인스턴스를 자동으로 시작하도록 Amazon EC2 Auto Scaling을 구성해야 합니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 손상된 Amazon EBS 볼륨이 있는 Auto Scaling 인스턴스 모니터링 및 교체를 참조하세요.
- 시스템 로그를 검색하여 오류가 있는지 검토합니다. 자세한 내용은 시스템 로그 검색 섹션을 참조하세요.

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. [인스턴스 상태(Instance state)], [인스턴스 재부팅(Reboot instance)]을 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` 상태가 되면 [작업(Actions)], [모니터링 및 문제 해결(Monitor and troubleshoot)], [시스템 로그 가져오기(Get system log)]를 차례로 선택합니다.
6. 화면에 표시되는 로그를 검토한 후 아래에 나와 있는 시스템 오류 구문 목록을 참조하여 문제를 해결합니다.
7. 문제가 해결되지 않으면 해당 문제를 AWS re:Post에 게시할 수 있습니다.

Linux 인스턴스의 시스템 로그 오류 문제 해결

Linux 인스턴스가 인스턴스 액세스 검사와 같은 인스턴스 상태 확인에 실패한 경우에는 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
- ERROR: mmu_update failed(메모리 관리 업데이트 실패)

디바이스 오류

- I/O 오류(블록 디바이스 장애)
- I/O ERROR: neither local nor remote disk(분산된 블록 디바이스 손상)

커널 오류

- request_module: runaway loop modprobe(이전 Linux 버전에서 레거시 커널 modprobe 반복)
- "FATAL: kernel too old" 및 "fsck: No such file or directory while trying to open /dev"(커널과 AMI 불일치)
- "FATAL: Could not load /lib/modules" 또는 "BusyBox"(커널 모듈 누락)
- ERROR Invalid kernel(EC2 커널이 호환되지 않음)

파일 시스템 오류

- fsck: No such file or directory while trying to open... (파일 시스템을 찾을 수 없음)
- 파일 시스템 마운트 관련 일반 오류(마운트 실패)
- VFS: Unable to mount root fs on unknown-block(루트 파일 시스템 불일치)
- Error: Unable to determine major/minor number of root device... (루트 파일 시스템/디바이스 불일치)
- XENBUS: Device with no driver...
- ... days without being checked, check forced(파일 시스템 검사 필요)
- fsck died with exit status... (디바이스 누락)

운영 체제 오류

- GRUB 프롬프트(grubdom>)

- 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)
- 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)
- XENBUS: Timeout connecting to devices(Xenbus 시간 초과)

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 수행하세요. • 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 수행하세요.

이 인스턴스 유형의 경우	조치
	• 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

AWS re:Post에 문제를 게시하거나 지원에 문의합니다.

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 수행하세요. 1. 인스턴스를 중지합니다.

이 인스턴스 유형의 경우	조치
	2. 볼륨을 분리합니다. 3. 볼륨 복구를 시도합니다. **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. 인스턴스를 시작합니다.
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 수행하세요. 1. 인스턴스를 중지합니다. 2. 최신 커널이 사용되도록 구성을 수정합니다. 3. 인스턴스를 시작합니다.
인스턴스 스토어 지원	다음 절차를 수행하세요. 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 수행하세요. 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 수행하세요. 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 수행하세요. 1. 인스턴스를 중지합니다. 2. 작동 중인 커널로 대체합니다. 3. 대체 커널을 설치합니다. 4. 올바른 커널로 변경하여 AMI를 수정합니다.
인스턴스 스토어 지원	다음 절차를 수행하세요. 1. 인스턴스를 종료하고 올바른 커널을 사용하여 새 인스턴스를 시작합니다. 2. 올바른 커널을 사용하여 AMI를 만듭니다. 3. (선택 사항) 지원에 데이터 복구를 위한 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 수행하세요. 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 참조하세요.
인스턴스 스토어 지원	다음 절차를 수행하세요. 1. 인스턴스를 종료하고 새 인스턴스를 시작합니다.

이 인스턴스 유형의 경우	조치
	2. 잘못된 Amazon EBS 볼륨을 모두 분리하고 인스턴스를 재부팅합니다. 3. (선택 사항) 지원에 데이터 복구를 위한 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. • ramdisk와 조합의 불일치(예: Debian ramdisk와 SUSE AMI).

권장 조치

이 인스턴스 유형의 경우	조치
Amazon EBS 지원	다음 절차를 수행하세요. 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.

이 인스턴스 유형의 경우	조치
	7. 중지된 인스턴스에 볼륨을 연결합니다. 8. 인스턴스를 시작합니다. 9. 인스턴스 상태를 다시 확인합니다.
인스턴스 스토어 지원	다음 중 하나를 시도하세요. • 새 인스턴스를 시작합니다. • (선택 사항) 지원에 데이터 복구를 위한 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. • 필요한 형식의 파일 시스템이 아닙니다. • 레거시 커널(예: 2.6.16-XenU)이 사용되었습니다. • 인스턴스의 최신 커널 업데이트(잘못된 업데이트 또는 업데이트 버그)

인스턴스 유형	예상 원인
인스턴스 스토어 지원	하드웨어 디바이스 실패.

권장 조치

이 인스턴스 유형의 경우	조치
Amazon EBS 지원	다음 중 하나를 수행하세요. • 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 수행하세요. 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 수행하세요. 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)
Initializing network drop monitor service
Freeing unused kernel memory: 508k freed
:: Starting udevd...
done.
:: Running Hook [udev]
:: Triggering uevents...<30>udevd[65]: starting version 173
done.
Waiting 10 seconds for device /dev/xvda1 ...
Root device '/dev/xvda1' doesn't exist. Attempting to create it.
ERROR: Unable to determine major/minor number of root device '/dev/xvda1'.
You are being dropped to a recovery shell
    Type 'exit' to try and continue booting
sh: can't access tty; job control turned off
[ramfs /]#
```

예상 원인

- 가상 블록 디바이스 드라이버가 없거나 잘못 구성되었습니다.
- 디바이스 열거형이 충돌합니다(sda와 xvda).
- 잘못된 인스턴스 커널을 선택했습니다.

권장 조치

이 인스턴스 유형의 경우	조치
Amazon EBS 지원	다음 절차를 수행하세요. 1. 인스턴스를 중지합니다. 2. 볼륨을 분리합니다. 3. 디바이스 매핑 문제를 해결합니다. 4. 인스턴스를 시작합니다.

이 인스턴스 유형의 경우	조치
	5. 디바이스 매핑 문제를 해결하도록 AMI를 수정합니다.
인스턴스 스토어 지원	다음 절차를 수행하세요. 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 해결하세요. • 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 해결하세요. • 올바른 도구로 ramdisk 번들을 다시 구성합니다. • 일관성 요구 사항을 제거하도록 파일 시스템 튜닝 매개 변수를 수정합니다(권장되지 않음). • 인스턴스를 종료하고 새 인스턴스를 시작합니다.

이 인스턴스 유형의 경우	조치
	• (선택 사항) 지원에 데이터 복구를 위한 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.

인스턴스 유형	예상 원인
	• 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.

이 인스턴스 유형의 경우	조치
	6. GRUB 버전에서 기본 파일 시스템 유형을 지원하는지 확인하고 필요할 경우 GRUB을 업그레이드합니다. 7. 파일 시스템을 분리합니다. 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 지원에 문의하세요.

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 수행하세요. • 하드 코딩이 제거되도록 AMI를 수정하고 인스턴스를 다시 시작합니다. • 하드 코딩된 MAC 주소가 제거되도록 인스턴스를 수정합니다. 또는 다음 절차를 수행하세요. 1. 인스턴스를 중지합니다. 2. 루트 볼륨을 분리합니다.

이 인스턴스 유형의 경우	조치
	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 수행하세요. • 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 수행하세요. 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 참조하세요. **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 수행하세요. 1. 인스턴스를 종료하고 새 인스턴스를 시작합니다.

이 인스턴스 유형의 경우	조치
	2. (선택 사항) 지원에 데이터 복구를 위한 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 수행하세요. • 현대식 커널을 사용하도록 AMI 및 인스턴스를 수정하고 인스턴스를 다시 시작합니다. • 인스턴스를 재부팅합니다.
인스턴스 스토어 지원	다음 중 하나를 수행하세요. • 인스턴스를 종료합니다. • 현대식 커널을 사용하도록 AMI를 수정하고 이 AMI를 사용하여 새 인스턴스를 시작합니다.

잘못된 볼륨에서 부팅되는 Amazon EC2 Linux 인스턴스 문제 해결

/dev/xvda 또는 /dev/sda에 연결된 볼륨이 아닌 다른 볼륨이 Linux 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/fstab의 내용이 다음과 같을 수 있습니다.

```
LABEL=/ / ext4 defaults,noatime 1 1
tmpfs /dev/shm tmpfs defaults 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0
```

이때 두 볼륨의 레이블을 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/xvda1이 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/fstab에서 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.

Amazon EC2 Windows 인스턴스 연결 문제 해결

다음 정보와 일반적인 오류는 Windows 인스턴스 연결에 연결할 때 발생하는 문제를 해결하는 데 도움이 될 수 있습니다.

연결 문제

- 원격 데스크톱으로 원격 컴퓨터에 연결할 수 없음
- macOS RDP 클라이언트 사용 중 오류 발생
- RDP에 바탕 화면이 아닌 빈 화면 표시
- 관리자가 아닌 사용자로 인스턴스에 원격 로그온할 수 없음

- AWS Systems Manager를 사용하여 원격 데스크톱 연결 문제 해결
- 원격 레지스트리를 사용하여 EC2 인스턴스에서 원격 데스크톱 활성화
- 프라이빗 키를 분실했습니다. 내 Windows 인스턴스에 연결하려면 어떻게 해야 하나요?

원격 데스크톱으로 원격 컴퓨터에 연결할 수 없음

인스턴스 연결과 관련된 문제를 해결하려면 다음을 수행합니다.

- 올바른 퍼블릭 DNS 호스트 이름을 사용하고 있는지 확인합니다. Amazon EC2 콘솔에서 인스턴스를 선택하고 세부 정보 창에서 퍼블릭 DNS(IPv4)를 확인합니다. 인스턴스가 VPC에 있는데 퍼블릭 DNS 이름이 표시되지 않는 경우 DNS 호스트 이름을 활성화해야 합니다. 자세한 내용을 알아보려면 Amazon VPC 사용 설명서의 VPC에 대한 DNS 속성을 참조하세요.
- 인스턴스에 퍼블릭 IPv4 주소가 있는지 확인합니다. 퍼블릭 IP 주소가 없을 경우 인스턴스와 탄력적 IP 주소를 연결할 수 있습니다. 자세한 내용은 탄력적 IP 주소 섹션을 참조하세요.
- IPv6 주소를 사용해 인스턴스에 연결하려면 로컬 컴퓨터에 IPv6 주소가 있고 IPv6를 사용하도록 구성되어 있어야 합니다. 자세한 내용을 알아보려면 Amazon VPC 사용 설명서의 인스턴스에서 IPv6 구성하기를 참조하세요.
- 보안 그룹에 포트 3389의 RDP 액세스를 허용하는 규칙이 있는지 확인합니다.
- 암호를 복사했는데 `Your credentials did not work` 오류가 표시되는 경우 메시지가 나타날 때 수동으로 입력해 봅니다. 암호를 복사할 때 빠진 문자가 있거나 공백 문자가 잘못 들어갔을 수 있습니다.
- 인스턴스가 상태 확인을 통과했는지 확인합니다. 자세한 내용은 Amazon EC2 인스턴스 상태 확인 및 the section called “Linux 인스턴스 상태 확인 실패”(을)를 참조하세요.
- 서브넷의 라우팅 테이블에 VPC 외부로 향하는 모든 트래픽을 VPC의 인터넷 게이트웨이로 전송하는 경로가 있는지 확인합니다. 자세한 내용은 Amazon VPC 사용 설명서의 사용자 지정 라우팅 테이블 만들기(인터넷 게이트웨이)를 참조하세요.
- Windows 방화벽이나 기타 방화벽 소프트웨어가 인스턴스로 전송되는 RDP 트래픽을 차단하지 않는지 확인합니다. Windows 방화벽을 비활성화하고 보안 그룹 규칙을 사용하여 인스턴스의 액세스를 제어하는 것이 좋습니다. 다음 옵션 중 하나를 사용해 보세요.
  - AWSSupport-TroubleshootRDP를 사용하여 disable the Windows Firewall profiles using SSM Agent. 이 옵션을 사용하려면 인스턴스가 AWS Systems Manager에 대해 구성되어 있어야 합니다.
  - AWSSupport-ExecuteEC2Rescue를 사용합니다.

  - 인스턴스를 중지하고 루트 볼륨을 분리한 다음 볼륨을 임시 인스턴스에 데이터 볼륨으로 연결합니다. 임시 인스턴스에 연결하고 볼륨을 온라인 상태로 전환합니다. 데이터 볼륨에서 레지스트리 하이브를 로드합니다. `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicyStandardProfile`에서 `EnableFirewall`을 0으로 설정합니다. 레지스트리 하이브를 언로드한 다음 볼륨을 오프라인으로 전환합니다. 볼륨을 임시 인스턴스에서 분리하여 원본 인스턴스에 루트 볼륨으로 연결합니다. 원본 인스턴스를 시작합니다.
- Active Directory 도메인의 일부가 아닌 인스턴스에서 네트워크 수준 인증이 비활성화되어 있는지 확인합니다(AWSSupport-TroubleshootRDP를 사용하여 disable NLA).
- 원격 데스크톱 서비스(TermService) 시작 유형이 자동으로 설정되어 있으며 해당 서비스가 시작된 상태인지 확인합니다(AWSSupport-TroubleshootRDP를 사용하여 enable and start the RDP service).
- 기본값이 3389인 올바른 원격 데스크톱 프로토콜 포트에 연결 중인지 확인합니다(AWSSupport-TroubleshootRDP를 사용하여 read the current RDP port 및 change it back to 3389).
- 원격 데스크톱 연결이 사용자의 인스턴스에 허용되어 있는지 확인합니다(AWSSupport-TroubleshootRDP를 사용하여 enable Remote Desktop connections).
- 암호가 만료되지 않았는지 확인합니다. 암호가 만료된 경우 재설정할 수 있습니다. 자세한 내용은 Amazon EC2 Windows 인스턴스의 Windows 관리자 암호 재설정 섹션을 참조하세요.
- 인스턴스에 생성한 사용자로 연결하려고 하면 `The user cannot connect to the server due to insufficient access privileges` 오류가 발생하는 경우 사용자에게 로컬 로그인 권한을 부여했는지 확인합니다. 자세한 내용은 멤버에게 로컬 로그인 권한 부여를 참조하세요.
- 동시에 허용되는 최대 RDP 세션 수를 초과한 경우 `Your Remote Desktop Services session has ended. Another user connected to the remote computer, so your connection was lost.` 메시지와 함께 세션이 종료됩니다. 기본적으로 허용되는 인스턴스의 동시 RDP 세션 수는 2개입니다.

macOS RDP 클라이언트 사용 중 오류 발생

Microsoft 웹 사이트의 원격 데스크톱 연결을 사용하여 Windows Server 인스턴스에 연결하려는 경우 다음 오류가 발생할 수 있습니다.

```
Remote Desktop Connection cannot verify the identity of the computer that you want to
 connect to.
```

Mac App Store에서 Microsoft Remote Desktop 앱을 다운로드한 후 이 앱을 사용하여 인스턴스에 연결합니다.

RDP에 바탕 화면이 아닌 빈 화면 표시

다음에 따라 문제를 해결하세요.

- 콘솔 출력에서 상세 정보를 확인합니다. Amazon EC2 콘솔을 사용하여 인스턴스에 대한 콘솔 출력을 가져오려면 인스턴스를 선택한 다음 [작업(Actions)], [모니터링 및 문제 해결(Monitor and troubleshoot)], [시스템 로그 가져오기(Get system log)]를 차례로 선택합니다.
- 최신 버전의 RDP 클라이언트를 실행하는지 확인합니다.
- RDP 클라이언트에 기본 설정을 적용해 봅니다.
- 원격 데스크톱 연결을 사용하는 경우 다음과 같이 `/admin` 옵션과 함께 시작해 봅니다.

```
mstsc /v:instance /admin
```

- 서버에서 전체 화면 애플리케이션을 실행하는 경우 응답이 중지되었을 수 있습니다. Ctrl+Shift+Esc를 눌러 Windows 작업 관리자를 시작하고 애플리케이션을 닫습니다.
- 서버 사용률이 과다한 경우 응답이 중지되었을 수 있습니다. Amazon EC2 콘솔을 사용하여 인스턴스를 모니터링하려면 인스턴스를 선택하고 모니터링 탭을 선택합니다. 인스턴스 유형을 더욱 큰 유형으로 변경해야 하는 경우 Amazon EC2 인스턴스 유형 변경 섹션을 참조하세요.

관리자가 아닌 사용자로 인스턴스에 원격 로그온할 수 없음

관리자 계정이 아닌 사용자로 Windows 인스턴스에 원격으로 로그온할 수 없는 경우 로컬에서 로그온할 수 있는 사용자 권한을 받았는지 확인합니다. 사용자 또는 그룹에 도메인의 도메인 컨트롤러에 로컬로 로그온할 수 있는 권한 부여를 참조하세요.

AWS Systems Manager를 사용하여 원격 데스크톱 연결 문제 해결

AWS Systems Manager를 통해 RDP를 사용하는 Windows 인스턴스에 연결하는 문제를 해결할 수 있습니다.

AWSSupport-TroubleshootRDP

사용자는 AWSSupport-TroubleshootRDP 자동화 문서를 사용하여 RDP 포트, 네트워크 계층 인증(NLA) 및 Windows 방화벽 프로파일 등 원격 데스크톱 프로토콜(RDP) 연결에 영향을 미치는 대상 인

스턴스의 일반 설정을 확인하거나 수정할 수 있습니다. 기본적으로 이 문서는 이러한 설정 값을 읽고 출력합니다.

AWSSupport-TroubleshootRDP 자동화 문서는 EC2 인스턴스, 온프레미스 인스턴스 및 AWS Systems Manager(관리형 인스턴스)에서 사용할 수 있도록 설정된 VM(가상 머신)에 사용할 수 있습니다. 또한 Systems Manager와 함께 사용하도록 설정되지 않은 Windows Server용 EC2 인스턴스에도 사용할 수 있습니다. AWS Systems Manager에서 사용할 인스턴스 활성화에 대한 자세한 내용을 알아보려면 AWS Systems Manager 사용 설명서의 관리형 노드를 참조하세요.

AWSSupport-TroubleshootRDP 문서 사용 문제를 해결하려면

1. Systems Manager 콘솔에 로그인합니다.
2. 손상된 인스턴스와 동일한 리전에 있는지 확인합니다.
3. 왼쪽 탐색 창에서 Documents(문서)를 선택합니다.
4. Owned by Amazon(Amazon 소유) 탭에서 검색 필드에 `AWSSupport-TroubleshootRDP`를 입력합니다. `AWSSupport-TroubleshootRDP` 문서가 표시되면 선택합니다.
5. 자동화 실행(Execute automation)을 선택합니다.
6. Execution Mode(실행 모드)에서 Simple execution(단순 실행)을 선택합니다.
7. Input parameters(입력 파라미터)의 InstanceId에서 Show interactive instance picker(대화형 인스턴스 선택기 표시)를 활성화합니다.
8. Amazon EC2 인스턴스를 선택합니다.
9. 예제를 검토한 후 실행을 선택합니다.
10. 실행 상황을 모니터링하려면 실행 상태에서 상태가 대기 중에서 성공으로 바뀔 때까지 기다립니다. 출력을 확장하여 결과를 봅니다. 실행된 단계에서 개별 단계의 출력을 보려면 스텝 ID를 선택합니다.

AWSSupport-TroubleshootRDP 예

다음 예제는 AWSSupport-TroubleshootRD를 통해 일반적인 문제 해결 작업을 수행하는 방법을 보여줍니다. 예제 AWS CLI start-automation-execution 명령이나 AWS Management Console에 대해 제공된 링크를 사용할 수 있습니다.

Example 예: 현재의 RDP 상태 확인

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --
parameters "InstanceId=i-1234567890abcdef0, Action=Custom" --region region_code
```

AWS Systems Manager 콘솔:

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region#documentVersion=$LATEST
```

Example 예: Windows 방화벽 비활성화

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP"
 --parameters "InstanceId=i-1234567890abcdef0, Action=Custom, Firewall=Disable" --
region region_code
```

AWS Systems Manager 콘솔:

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region_code#documentVersion=$LATEST&Firewall=Disable
```

Example 예: 네트워크 수준 인증 비활성화

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --
parameters "InstanceId=i-1234567890abcdef0, Action=Custom, NLASettingAction=Disable" --
region region_code
```

AWS Systems Manager 콘솔:

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region_code#documentVersion
```

Example 예: RDP 서비스 시작 유형을 '자동'으로 설정하고 RDP 서비스를 시작

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --
parameters "InstanceId=i-1234567890abcdef0, Action=Custom, RDPServiceStartupType=Auto,
 RDPServiceAction=Start" --region region_code
```

AWS Systems Manager 콘솔:

```
https://console.aws.amazon.com/systems-manager/automation/execute/
AWSSupport-TroubleshootRDP?region=region_code#documentVersion=
$LATEST&RDPServiceStartupType=Auto&RDPServiceAction=Start
```

Example 예: 기본 RDP 포트(3389) 복원

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --
parameters "InstanceId=i-1234567890abcdef0, Action=Custom, RDPPortAction=Modify" --
region region_code
```

AWS Systems Manager 콘솔:

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region_code#documentVersion=$LATEST&RDPPortAction=Modify
```

Example 예: 원격 연결 허용

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --
parameters "InstanceId=i-1234567890abcdef0, Action=Custom, RemoteConnections=Enable" --
region region_code
```

AWS Systems Manager 콘솔:

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region_code#documentVersion=$LATEST&RemoteConnections=Enable
```

AWSSupport-ExecuteEC2Rescue

AWSSupport-ExecuteEC2Rescue 자동화 문서에서는 Windows Server용 EC2Rescue를 사용하여 EC2 인스턴스 연결 및 RDP 문제를 자동으로 해결하고 복원합니다. 자세한 내용을 알아보려면 연결할 수 없는 인스턴스에서 EC2Rescue 도구 실행을 참조하세요.

AWSSupport-ExecuteEC2Rescue 자동화 문서는 해당 인스턴스를 중지하고 재시작해야 합니다. Systems Manager 자동화는 인스턴스를 중지하고 Amazon 머신 이미지(AMI)를 생성합니다. 인스턴스 스토어 볼륨에 저장되어 있는 데이터는 사라집니다. 탄력적 IP 주소를 사용하지 않는 경우 퍼블릭 IP 주소가 변경됩니다. 자세한 내용을 알아보려면 AWS Systems Manager 사용 설명서의 연결할 수 없는 인스턴스에서 EC2Rescue 도구 실행을 참조하세요.

AWSSupport-ExecuteEC2Rescue 문서 사용 문제를 해결하려면

1. Systems Manager 콘솔을 엽니다.
2. 손상된 Amazon EC2 인스턴스와 동일한 리전에 있는지 확인합니다.
3. 탐색 패널에서 문서를 선택합니다.
4. `AWSSupport-ExecuteEC2Rescue` 문서를 검색하고 선택한 다음 자동화 실행을 선택합니다.
5. Execution Mode(실행 모드)에서 Simple execution(단순 실행)을 선택합니다.
6. 입력 파라미터 섹션에서 연결할 수 없는 인스턴스의 Amazon EC2 인스턴스 ID를 UnreachableInstanceId에 입력합니다.
7. (선택 사항) Amazon EC2 인스턴스의 문제 해결을 위해 운영 체제 로그를 수집하려면 LogDestination에 Amazon Simple Storage Service(Amazon S3) 버킷 이름을 입력합니다. 지정된 버킷으로 로그가 자동으로 업로드됩니다.
8. 실행을 선택합니다.
9. 실행 상황을 모니터링하려면 실행 상태에서 상태가 대기 중에서 성공으로 바뀔 때까지 기다립니다. 출력을 확장하여 결과를 봅니다. 실행된 단계에서 개별 단계의 출력을 보려면 스텝 ID를 선택합니다.

원격 레지스트리를 사용하여 EC2 인스턴스에서 원격 데스크톱 활성화

연결할 수 없는 인스턴스가 AWS Systems Manager Session Manager에 의해 관리되지 않는 경우 원격 레지스트리를 통해 원격 데스크톱을 사용하도록 설정할 수 있습니다.

1. EC2 콘솔에서 연결할 수 없는 인스턴스를 중지합니다.

2. 연결할 수 없는 인스턴스의 루트 볼륨을 분리하고 스토리지 볼륨과 동일한 가용 영역에 있는 연결 가능한 인스턴스에 연결합니다. 동일한 가용 영역에 연결 가능한 인스턴스가 없는 경우 하나 시작합니다. 연결할 수 없는 인스턴스에 있는 루트 볼륨의 디바이스 이름을 기록해 둡니다.
3. 연결 가능한 인스턴스에서 디스크 관리를 엽니다. 명령 프롬프트 창에서 다음 명령을 실행하면 됩니다.

```
diskmgmt.msc
```

4. 연결할 수 없는 인스턴스에서 온 새로 연결된 볼륨을 마우스 오른쪽 버튼으로 클릭한 다음 온라인을 선택합니다.
5. Windows 레지스트리 편집기를 엽니다. 명령 프롬프트 창에서 다음 명령을 실행하면 됩니다.

```
regedit
```

6. 레지스트리 편집기에서 HKEY_LOCAL_MACHINE을 선택한 다음 파일, Hive 로드를 차례로 선택합니다.
7. 연결된 볼륨의 드라이브를 선택하고 `\Windows\System32\config\`로 이동하여 `SYSTEM`을 선택한 다음 열기를 선택합니다.
8. 키 이름에 Hive의 고유한 이름을 입력하고 확인을 선택합니다.
9. 레지스트리를 변경하기 전에 레지스트리 Hive를 백업합니다.
   a. 레지스트리 편집기 콘솔 트리에서 로드한 하이브(HKEY_LOCAL_MACHINE*your-key-name*)를 선택합니다.
   b. 파일, 내보내기를 선택합니다.
   c. Export Registry File(레지스트리 파일 내보내기) 대화 상자에서 백업 복사본을 저장할 위치를 선택한 다음 파일 이름 필드에 백업 파일의 이름을 입력합니다.
   d. 저장을 선택합니다.
10. 레지스트리 편집기에서 `HKEY_LOCAL_MACHINE\`*`your key name`*`\ControlSet001\Control\Terminal Server`로 이동한 다음 세부 정보 창에서 fDenyTSConnections를 두 번 클릭합니다.
11. Edit DWORD(DWORD 편집) 값 상자의 Value data(값 데이터) 필드에 `0`을 입력합니다.
12. 확인을 선택합니다.

> **Note**
>
> Value data(값 데이터) 필드의 값이 1인 경우 인스턴스에서 원격 데스크톱 연결을 거부합니다. `0` 값은 원격 데스크톱 연결을 허용합니다.

13. 레지스트리 편집기에서 HKEY_LOCAL_MACHINE*your-key-name*을 선택한 다음 파일, 하이브 언로드를 선택합니다.
14. 레지스트리 편집기와 디스크 관리를 닫습니다.
15. EC2 콘솔에서 연결할 수 있는 인스턴스에서 볼륨을 분리하고 연결할 수 없는 인스턴스에 볼륨을 다시 연결합니다. 연결할 수 없는 인스턴스에 볼륨을 연결할 때 디바이스 필드에 이전에 저장한 디바이스 이름을 입력합니다.
16. 연결할 수 없는 인스턴스를 다시 시작합니다.

프라이빗 키를 분실했습니다. 내 Windows 인스턴스에 연결하려면 어떻게 해야 하나요?

새로 시작된 Windows 인스턴스에 연결하는 경우 인스턴스를 시작할 때 지정한 키 페어의 프라이빗 키를 사용하여 관리자 계정에 대한 암호를 해독합니다.

관리자 암호를 분실하여 더 이상 프라이빗 키를 가지고 있지 않은 경우 암호를 재설정하거나 새 인스턴스를 생성해야 합니다. 자세한 내용은 Amazon EC2 Windows 인스턴스의 Windows 관리자 암호 재설정 섹션을 참조하세요. Systems Manager 문서를 사용하여 암호를 재설정하는 단계는 AWS Systems Manager 사용 설명서의 EC2 인스턴스의 암호 및 SSH 키 재설정을 참조하세요.

Amazon EC2 Windows 인스턴스 시작 문제 해결

다음은 Amazon EC2 Windows 인스턴스와 관련된 암호 및 활성화 문제를 해결하는 데 도움이 되는 문제 해결 팁입니다.

문제

- "Password is not available"
- "Password not available yet"
- "Cannot retrieve Windows password"
- "Waiting for the metadata service"

- "Unable to activate Windows"
- "Windows is not genuine (0x80070005)"
- "No Terminal Server License Servers available to provide a license"
- "일부 설정이 사용자의 조직에 의해 관리됩니다.(Some settings are managed by your organization.)"

"Password is not available"

원격 데스크톱을 사용하여 Windows 인스턴스에 연결하려면 계정과 암호를 지정해야 합니다. 제공되는 계정과 암호는 인스턴스를 시작하는 데 사용한 AMI를 기준으로 합니다. 관리자 계정의 자동 생성 암호를 검색할 수도 있고, AMI가 생성된 원래 인스턴스에서 사용하는 계정과 암호를 사용할 수도 있습니다.

사용자 지정 Windows AMI를 사용하여 시작한 인스턴스의 관리자 계정 암호를 생성할 수 있습니다. 암호를 생성하려면 AMI 생성 전에 운영 체제에서 일부 설정을 구성해야 합니다. 자세한 내용은 Amazon EBS 지원 AMI 생성 단원을 참조하십시오.

Windows 인스턴스가 무작위 암호를 생성하도록 구성되지 않은 경우 콘솔을 사용하여 자동 생성 암호를 검색할 때 다음 메시지가 표시됩니다.

```
Password is not available.
The instance was launched from a custom AMI, or the default password has changed. A
password cannot be retrieved for this instance. If you have forgotten your password,
 you can
reset it using the Amazon EC2 configuration service. For more information, see
 Passwords for a
Windows Server instance.
```

인스턴스의 콘솔 출력을 확인하여 인스턴스를 시작하는 데 사용한 AMI가 암호 생성이 해제된 상태로 생성되었는지 확인합니다. 암호 생성이 해제된 경우 콘솔 출력에 다음 내용이 포함됩니다.

```
Ec2SetPassword: Disabled
```

암호 생성이 해제되어 있으며 원래 인스턴스의 암호를 모르는 경우 이 인스턴스의 암호를 재설정할 수 있습니다. 자세한 내용은 Amazon EC2 Windows 인스턴스의 Windows 관리자 암호 재설정 섹션을 참조하세요.

"Password not available yet"

원격 데스크톱을 사용하여 Windows 인스턴스에 연결하려면 계정과 암호를 지정해야 합니다. 제공되는 계정과 암호는 인스턴스를 시작하는 데 사용한 AMI를 기준으로 합니다. 관라자 계정의 자동 생성 암호를 검색할 수도 있고, AMI가 생성된 원래 인스턴스에서 사용하는 계정과 암호를 사용할 수도 있습니다.

몇 분 이내에 암호가 제공됩니다. 암호가 제공되지 않는 경우 콘솔을 사용하여 자동 생성 암호를 검색할 때 다음 메시지가 표시됩니다.

```
Password not available yet.
Please wait at least 4 minutes after launching an instance before trying to retrieve
 the
auto-generated password.
```

4분이 지났지만 암호를 받지 못한 경우 인스턴스의 시작 에이전트가 암호를 생성하도록 구성되지 않았을 수 있습니다. 콘솔 출력이 비어 있는지 확인해 봅니다. 자세한 내용은 콘솔 출력을 가져올 수 없음 섹션을 참조하세요.

또한 Management Portal에 액세스하는 데 사용되는 AWS Identity and Access Management(IAM) 계정에 `ec2:GetPasswordData` 작업이 허용되었는지 확인합니다. IAM 권한에 대한 자세한 내용은 IAM 이란?을 참조하세요.

"Cannot retrieve Windows password"

관리자 계정의 자동 생성 암호를 검색하려면 인스턴스 시작 시에 지정한 키 페어의 프라이빗 키를 사용해야 합니다. 인스턴스 시작 시에 키 페어를 지정하지 않은 경우 다음 메시지가 표시됩니다.

```
Cannot retrieve Windows password
```

이 인스턴스를 종료하고 같은 AMI를 사용하여 키 페어를 지정하면서 새 인스턴스를 시작할 수 있습니다.

"Waiting for the metadata service"

Windows 인스턴스는 인스턴스 메타데이터 정보를 가져와야 활성화될 수 있습니다. 기본적으로 `WaitForMetaDataAvailable` 설정은 EC2Config 서비스가 인스턴스 메타데이터에 액세스되는 것을 기다린 후에 부팅 프로세스를 진행하도록 합니다. 자세한 내용은 인스턴스 메타데이터를 사용하여 EC2 인스턴스를 관리합니다. 섹션을 참조하세요.

인스턴스가 인스턴스 연결성 테스트에 실패한 경우 다음 방법으로 이 문제를 해결하세요.

- VPC에 대한 CIDR 블록을 검사합니다. IP 주소의 범위가 `224.0.0.0~255.255.255.255`인 VPC로 시작한 경우(Class D 및 Class E IP 주소 범위) Windows 인스턴스가 올바르게 부팅되지 않습니다. 이러한 IP 주소 범위는 예약되어 있으므로 호스트 디바이스에 할당해서는 안 됩니다. RFC 1918 규격에 따라 사설(비공개적으로 라우팅 가능) IP 주소 범위에 속하는 CIDR 블록으로 VPC를 생성하는 것이 좋습니다.
- 시스템이 고정 IP 주소로 구성되었을 수 있습니다. 네트워크 인터페이스를 생성하고 인스턴스에 연결합니다.
- 연결할 수 없는 Windows 인스턴스의 DHCP를 활성화하려면 다음을 수행합니다.

  1. 해당 인스턴스를 중지하고 루트 볼륨을 분리합니다.
  2. 해당 인스턴스와 동일한 가용 영역에서 임시 인스턴스를 시작합니다.

     > **Warning**
     >
     > 임시 인스턴스가 원본 인스턴스와 동일한 AMI를 기반으로 하는 경우 추가 단계를 수행해야 합니다. 그렇지 않으면 디스크 서명 충돌로 인해 루트 볼륨 복원 후 원본 인스턴스를 부팅할 수 없습니다. 또는 임시 인스턴스에 대한 다른 AMI를 선택합니다. 예를 들어 원본 인스턴스에서 Windows Server 2016용 AWS Windows AMI를 사용할 경우 Windows Server 2019용 AWS Windows AMI를 사용하여 임시 인스턴스를 시작합니다.

  3. 해당 인스턴스의 루트 볼륨을 이 임시 인스턴스에 연결합니다. 임시 인스턴스에 연결하고 디스크 관리 유틸리티를 열어서 드라이브를 온라인 상태로 만듭니다.
  4. 임시 인스턴스에서 Regedit를 열고 HKEY_LOCAL_MACHINE을 선택합니다. 파일 메뉴에서 Hive 로드를 선택합니다. 드라이브를 선택하고, `Windows\System32\config\SYSTEM` 파일을 열고, 메시지가 나타나면 키 이름을 임의로 지정합니다.
  5. 방금 로드한 키를 선택하고 `ControlSet001\Services\Tcpip\Parameters\Interfaces` 경로로 이동합니다. 각 네트워크 인터페이스가 GUID에 따라 나열됩니다. 올바른 네트워크 인터페이스를 선택합니다. DHCP를 비활성화하고 고정 IP 주소를 할당한 경우 `EnableDHCP`는 0으로 설정됩니다. DHCP를 활성화하려면 `EnableDHCP`를 1로 설정하고 `NameServer`, `SubnetMask`, `IPAddress` 및 `DefaultGateway` 키가 있는 경우 해당 키를 삭제합니다. 키를 한 번 더 선택하고 파일 메뉴에서 Hive 로드 취소를 선택합니다.

> **Note**
>
> 네트워크 인터페이스가 여러 개일 경우 DHCP를 사용할 인터페이스를 식별해야 합니다. 해당 네트워크 인터페이스를 확인하려면 `NameServer`, `SubnetMask`, `IPAddress`, `DefaultGateway` 키 값을 확인합니다. 이들 값은 이전 인스턴스의 정적 구성을 표시합니다.

6. (선택 사항) DHCP가 이미 활성화되어 있다면 메타데이터 서비스로 연결되는 경로가 없는 경우일 수 있습니다. EC2Config를 업데이트하면 문제가 해결됩니다.

   a. 최신 버전의 EC2Config 서비스를 다운로드하여 설치합니다. 이 서비스 설치에 대한 자세한 내용은 the section called “EC2Config 설치” 섹션을 참조하세요.

   b. 연결한 드라이브의 `.zip` 디렉터리에 `Temp` 파일의 압축을 풉니다.

   c. Regedit를 열고 HKEY_LOCAL_MACHINE을 선택합니다. 파일 메뉴에서 Hive 로드를 선택합니다. 드라이브를 선택하고, `Windows\System32\config\SOFTWARE` 파일을 열고, 메시지가 나타나면 키 이름을 임의로 지정합니다.

   d. 방금 로드한 키를 선택하고 `Microsoft\Windows\CurrentVersion` 경로로 이동합니다. `RunOnce` 키를 선택합니다. 이 키가 없는 경우 `CurrentVersion`을 마우스 오른쪽 버튼으로 클릭하고 새로 생성을 가리킨 후 키를 선택하고 키의 이름을 `RunOnce`로 지정합니다. 마우스 오른쪽 버튼을 클릭하고 키를 가리킨 후 문자열 값을 선택합니다. `Ec2Install`을 이름으로, `C:\Temp\Ec2Install.exe -q`를 데이터로 입력합니다.

   e. 키를 한 번 더 선택하고 파일 메뉴에서 Hive 로드 취소를 선택합니다.

7. (선택 사항) 임시 인스턴스가 원본 인스턴스와 동일한 AMI를 기반으로 하는 경우 다음 단계를 수행해야 합니다. 그렇지 않으면 디스크 서명 충돌로 인해 루트 볼륨 복원 후 원본 인스턴스를 부팅할 수 없습니다.

> **Warning**
>
> 다음 절차에서는 레지스트리 편집기를 사용하여 Windows 레지스트리를 편집하는 방법을 설명합니다. Windows 레지스트리에 대해 또는 레지스트리 편집기를 사용하여 안전하게 수정하는 방법을 잘 알지 못하는 경우 레지스트리 구성을 참조하세요.

   a. 명령 프롬프트를 열고 regedit.exe를 입력한 후 Enter 키를 누릅니다.

b. 레지스트리 편집기의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)에서 HKEY_LOCAL_MACHINE을 선택한 후 찾기를 선택합니다.

c. Windows Boot Manager를 입력한 후 다음 찾기를 선택합니다.

d. 키 `11000001`을 선택합니다. 이 키는 이전 단계에서 찾은 키 바로 위의 키입니다.

e. 오른쪽 창에서 `Element`를 선택한 후 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)에서 수정을 선택합니다.

f. 데이터에서 오프셋 0x38의 4바이트 디스크 서명을 찾습니다. 바이트를 거꾸로 하여 디스크 서명을 만들고 기록해 둡니다. 예를 들어, 다음 데이터가 나타내는 디스크 서명은 `E9EB3AA5`입니다.

```
...
0030  00 00 00 00 01 00 00 00
0038  A5 3A EB E9 00 00 00 00
0040  00 00 00 00 00 00 00 00
...
```

g. 명령 프롬프트 창에서 다음 명령을 실행하여 Microsoft DiskPart를 시작합니다.

```
diskpart
```

h. 다음 DiskPart 명령을 실행하여 볼륨을 선택합니다. 디스크 관리 유틸리티를 사용하여 디스크 번호가 1인지 확인할 수 있습니다.

```
DISKPART> select disk 1

Disk 1 is now the selected disk.
```

i. 다음 DiskPart 명령을 실행하여 디스크 서명을 봅니다.

```
DISKPART>  uniqueid disk

Disk ID: 0C764FA8
```

j. 이전 단계에서 표시된 디스크 서명이 앞에서 기록한 BCD의 디스크 서명과 일치하지 않을 경우 다음 DiskPart 명령을 사용하여 일치하도록 디스크 서명을 변경합니다.

```
DISKPART> uniqueid disk id=E9EB3AA5
```

8. 디스크 관리 유틸리티를 사용하여 드라이브를 오프라인으로 설정합니다.

> **Note**
>
> 임시 인스턴스가 영향을 받는 인스턴스와 동일한 운영 체제를 실행하고 있는 경우 드라이브가 자동으로 오프라인 상태가 되므로 사용자가 수동으로 드라이브를 오프라인으로 설정할 필요가 없습니다.

9. 임시 인스턴스에서 볼륨을 분리합니다. 임시 인스턴스를 더 이상 사용하지 않는 경우 해당 인스턴스는 종료해도 됩니다.
10. 볼륨을 /dev/sda1로 연결하여 해당 인스턴스의 루트 볼륨을 복원합니다.
11. 영향을 받는 인스턴스를 시작합니다.

인스턴스에 연결된 경우 인스턴스에서 인터넷 브라우저를 열고 다음과 같이 메타데이터 서버 URL을 입력합니다.

```
http://169.254.169.254/latest/meta-data/
```

메타데이터 서버에 접속할 수 없는 경우 다음 방법으로 문제를 해결하세요.

- 최신 버전의 EC2Config 서비스를 다운로드하여 설치합니다. 이 서비스 설치에 대한 자세한 내용은 the section called “EC2Config 설치” 단원을 참조하세요.
- Windows 인스턴스에서 Red Hat PV 드라이버를 실행하는지 여부를 확인합니다. 실행하는 경우 Citrix PV 드라이버를 업데이트합니다. 자세한 내용은 the section called “PV 드라이버 업그레이드” 단원을 참조하십시오.
- 방화벽, IPSec 및 프록시 설정으로 인해 메타데이터 서비스(169.254.169.254) 또는 AWS KMS 서버(TargetKMSServer의 C:\Program Files\Amazon\Ec2ConfigService\Settings\ActivationSettings.xml 요소에 지정된 주소)로 송신되는 트래픽이 차단되지 않는지 확인합니다.
- 다음 명령을 사용하여 메타데이터 서비스(169.254.169.254)로 연결되는 경로가 있는지 확인합니다.

  ```
  route print
  ```

- 인스턴스의 가용 영역에 영향을 줄 수 있는 네트워크 문제가 있는지 확인합니다. http://status.aws.amazon.com/으로 이동합니다.

"Unable to activate Windows"

Windows 인스턴스는 Windows AWS KMS 정품 인증을 사용합니다. 인스턴스가 `A problem occurred when Windows tried to activate. Error Code 0xC004F074` 서버에 접속할 수 없는 경우 AWS KMS 메시지가 표시됩니다. 180일마다 Windows 정품을 인증해야 합니다. EC2Config는 정품 인증 기간이 만료되기 전에 AWS KMS 서버 접속을 시도하여 Windows의 정품 인증 상태를 유지합니다.

Windows 정품 인증에 문제가 발생하는 경우, 다음 절차에 따라 문제를 해결하세요.

EC2Config의 경우(Windows Server 2012 R2 AMI 이전)

1. 최신 버전의 EC2Config 서비스를 다운로드하여 설치합니다. 이 서비스 설치에 대한 자세한 내용은 the section called “EC2Config 설치” 섹션을 참조하세요.
2. 인스턴스에 로그인하여 `C:\Program Files\Amazon\Ec2ConfigService\Settings\config.xml` 파일을 엽니다.
3. `config.xml` 파일에서 Ec2WindowsActivate 플러그인을 찾습니다. 상태를 활성화로 변경한 다음 변경 사항을 저장합니다.
4. Windows Services 스냅인에서 EC2Config 서비스를 재시작하거나 인스턴스를 재부팅합니다.

이렇게 해도 정품 인증 문제가 해결되지 않는 경우, 다음과 같은 추가 단계를 수행하세요.

1. AWS KMS 대상 설정: C:\> slmgr.vbs /skms 169.254.169.250:1688
2. Windows 정품 인증: C:\> slmgr.vbs /ato

EC2Launch의 경우(Windows Server 2016 AMI 이상)

1. 관리 권한이 있는 PowerShell 프롬프트에서 EC2Launch 모듈을 가져옵니다.

```
PS C:\> Import-Module "C:\ProgramData\Amazon\EC2-Windows\Launch\Module
\Ec2Launch.psd1"
```

2. Add-Routes 함수를 호출하여 새 라우팅 목록을 표시합니다.

```
PS C:\> Add-Routes
```

3. Set-ActivationSettings 함수를 호출합니다.

```
PS C:\> Set-Activationsettings
```

4. 그런 다음 아래 스크립트를 실행하여 Windows를 활성화합니다.

```
PS C:\> cscript "${env:SYSTEMROOT}\system32\slmgr.vbs" /ato
```

EC2Config와 EC2Launch의 경우, 그래도 정품 인증 오류가 발생한다면 다음 정보를 확인해 보세요.

- AWS KMS 서버로 연결되는 경로가 있는지 확인합니다. `C:\Program Files\Amazon\Ec2ConfigService\Settings\ActivationSettings.xml`을 열고 `TargetKMSServer` 요소를 찾습니다. 다음 명령을 실행하여 이러한 AWS KMS 서버 주소가 나열되어 있는지 확인합니다.

```
route print
```

- AWS KMS 클라이언트 키가 설정되어 있는지 확인합니다. 다음 명령을 실행하고 출력을 확인합니다.

```
C:\Windows\System32\slmgr.vbs /dlv
```

  출력에 Error: product key not found 메시지가 있는 경우, AWS KMS 클라이언트 키가 설정되지 않은 것입니다. AWS KMS 클라이언트 키가 설정되지 않은 경우 이 Microsoft 문서(AWS KMS 클라이언트 설정 및 제품 키)의 설명에 따라 클라이언트 키를 조회하고 다음 명령을 실행하여 AWS KMS 클라이언트 키를 설정합니다.

```
C:\Windows\System32\slmgr.vbs /ipk client_key
```

- 시스템의 시간과 표준 시간대가 올바른지 확인합니다. UTC 이외의 표준 시간대를 사용하는 경우 `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TimeZoneInformation\RealTimeIsUniversal` 레지스트리 키를 추가하고 값을 `1`로 설정하여 시간을 정확하게 맞춥니다.
- Windows 방화벽이 활성화된 경우 다음 명령을 사용하여 임시로 비활성화합니다.

```
netsh advfirewall set allprofiles state off
```

"Windows is not genuine (0x80070005)"

Windows 인스턴스는 Windows AWS KMS 정품 인증을 사용합니다. 인스턴스에서 정품 인증 프로세스를 완료할 수 없는 경우 Windows 사본이 정품이 아니라는 메시지가 표시됩니다.

"Unable to activate Windows"의 권장 사항을 따르세요.

"No Terminal Server License Servers available to provide a license"

Windows Server의 기본 라이선스는 원격 데스크톱 사용자 2명 동시 연결입니다. 3명 이상의 사용자가 원격 데스크톱으로 Windows 인스턴스에 액세스해야 하는 경우 원격 데스크톱 서비스 CAL(Client Access License)을 구입하고 원격 데스크톱 세션 호스트 및 원격 데스크톱 라이선싱 서버 역할을 설치해야 합니다.

다음과 같은 문제가 있는지 확인합니다.

- 최대 동시 RDP 세션 수를 초과했습니다.
- Windows 원격 데스크톱 서비스 역할을 설치했습니다.
- 라이선스가 만료되었습니다. 라이선스가 만료된 경우 Windows 인스턴스에 사용자로 연결할 수 없습니다. 다음과 같은 방법을 시도할 수 있습니다.
  - 다음과 같이 명령줄에서 `/admin` 파라미터를 사용하여 인스턴스에 연결합니다.

    ```
    mstsc /v:instance /admin
    ```

    자세한 내용은 Microsoft 문서(Access Remote Desktop Via Command Line)를 참조하세요.
  - 인스턴스를 중지하고 Amazon EBS 볼륨을 분리한 후 같은 가용 영역의 다른 인스턴스에 연결하여 데이터를 복구합니다.

"일부 설정이 사용자의 조직에 의해 관리됩니다.(Some settings are managed by your organization.)"

최신 Windows Server AMI에서 시작한 인스턴스에서 "일부 설정이 사용자의 조직에 의해 관리됩니다.(Some settings are managed by your organization.)"라는 Windows 업데이트 대화 상자 메시지를 표시할 수 있습니다. 이 메시지는 Windows Server의 변경 사항으로 인해 나타나며 업데이트 설정 관리 기능 또는 Windows 업데이트 동작에는 영향을 주지 않습니다.

경고를 제거하려면

1. `gpedit.msc`를 열고 컴퓨터 구성(Computer Configuration), 관리 템플릿(Administrative Templates), Windows 구성 요소(Windows Components), Windows 업데이트(Windows updates)로 이동합니다. 자동 업데이트 구성(Configure Automatic Update)을 편집하고 활성(enabled)으로 설정합니다.
2. 명령 프롬프트에서 gpupdate /force을(를) 사용하여 그룹 정책을 업데이트합니다.
3. Windows 업데이트 설정을 닫은 다음 다시 엽니다. 사용자에게는 위와 같이 사용자의 설정이 사용자의 조직에 의해 관리된다는 메시지가 표시되며, 이어서 “We’ll automatically download updates, except on metered connections (where charges may apply). In that case, we’ll automatically download those updates required to keep Windows running smoothly.(측정된 연결(요금 발생 가능)을 제외하고 업데이트를 자동으로 다운로드합니다. 이 경우에는 원활한 Windows 실행 유지에 필요한 업데이트를 자동으로 다운로드합니다.)”라는 메시지도 표시됩니다.
4. `gpedit.msc`로 돌아가 그룹 정책을 다시 구성되지 않음(not configured)으로 설정합니다. gpupdate /force를 다시 실행합니다.
5. 명령 프롬프트를 닫고 몇 분간 기다립니다.
6. Windows 업데이트 설정 다시 엽니다. "일부 설정이 사용자의 조직에 의해 관리됩니다.(Some settings are managed by your organization)"라는 메시지가 표시되지 않아야 합니다.

Amazon EC2 Windows 인스턴스 문제 해결

다음은 Amazon EC2 Windows 인스턴스와 관련된 문제를 해결하는 데 도움이 되는 문제 해결 팁입니다.

문제

- AWS Systems Manager Sessions Manager를 Windows Server 2025 인스턴스에 연결할 수 없음
- EBS 볼륨이 Windows Server 2016 및 2019에서 초기화를 수행하지 않음
- DSRM(Directory Services Restore Mode)로 EC2 Windows 인스턴스 부팅
- 인스턴스의 네트워크 연결이 끊어지거나 예약된 작업이 예정 시간에 실행되지 않음
- 콘솔 출력을 가져올 수 없음
- 네트워크에서 Windows Server 2012 R2를 이용할 수 없는 경우
- 디스크 서명 충돌

AWS Systems Manager Sessions Manager를 Windows Server 2025 인스턴스에 연결할 수 없음

AWS Systems Manager Sessions Manager를 Windows Server 2025 인스턴스에 연결하는 데 문제가 발생할 수 있습니다. 이 문제를 해결하려면 인스턴스에 로그인한 다음 `Settings > Apps > Optional Features`으로 이동하여 `WMIC`를 추가합니다. SSM Agent 서비스를 다시 시작하거나 인스턴스를 재부팅하면 Sessions Manager가 연결됩니다.

다음 PowerShell 명령을 사용하여 동일한 작업을 수행할 수도 있습니다.

```
Start-Process -FilePath "$env:SystemRoot\system32\Dism.exe" -ArgumentList @('/
Online', '/Add-Capability', '/CapabilityName:WMIC~~~~') -Wait; Restart-Service -Name
 AmazonSSMAgent
```

EBS 볼륨이 Windows Server 2016 및 2019에서 초기화를 수행하지 않음

Windows Server 2016 및 2019용 Amazon Machine Image(AMI)에서 생성한 인스턴스는 EBS 볼륨 초기화를 비롯한 다양한 시작 작업에 EC2Launch v1 에이전트를 사용합니다. 기본적으로 EC2Launch v1은 두 번째 볼륨을 초기화하지 않습니다. 하지만 다음과 같이 자동으로 이러한 디스크를 초기화하도록 EC2Launch v1을 구성할 수 있습니다.

드라이브 문자를 볼륨에 매핑

1. 구성할 인스턴스에 연결하고 `C:\ProgramData\Amazon\EC2-Windows\Launch\Config\DriveLetterMappingConfig.json` 텍스트 편집기에 파일을 엽니다.
2. 다음과 같이 볼륨 설정을 지정합니다.

   ```
   {
   "driveLetterMapping": [
    {
      "volumeName": "sample volume",
      "driveLetter": "H"
    }]
   }
   ```

3. 변경 내용을 저장하고 파일을 닫습니다.
4. 디스크를 초기화할 EC2Launch v1 스크립트를 수행하기 위해서 Windows PowerShell을 열고 다음 명령을 실행합니다.

```
PS C:\>  C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1
```

인스턴스를 부팅할 때마다 디스크를 초기화하려면 다음과 같이 -Schedule 플래그를 추가하세요.

```
PS C:\>  C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1 -
Schedule
```

EC2Launch v1 에이전트는 InitializeInstance.ps1 스크립트와 동시에 실행되는 initializeDisks.ps1 같은 인스턴스 초기화 스크립트를 실행할 수 있습니다. InitializeInstance.ps1 스크립트가 인스턴스를 재부팅하는 경우 인스턴스 시작 시 실행되는 다른 예약된 작업이 중단될 수 있습니다. 잠재적인 충돌을 방지하려면 먼저 인스턴스 초기화가 완료되었는지 확인하는 로직을 initializeDisks.ps1 스크립트에 추가하는 것이 좋습니다.

> **Note**
>
> EC2Launch 스크립트가 볼륨을 초기화하지 않는 경우 볼륨이 온라인 상태인지 확인하세요. 볼륨이 오프라인인 경우 다음 명령을 실행하여 모든 디스크를 온라인으로 전환합니다.
>
> ```
> PS C:\> Get-Disk | Where-Object IsOffline -Eq $True | Set-Disk -IsOffline
>  $False
> ```

DSRM(Directory Services Restore Mode)로 EC2 Windows 인스턴스 부팅

Microsoft Active Directory를 실행 중인 인스턴스에서 시스템 오류 또는 그 밖에 심각한 문제가 발생할 경우 DSRM(Directory Services Restore)이라는 특수한 버전의 안전 모드로 부팅해 인스턴스 문제를 해결할 수 있습니다. DSRM에서 Active Directory를 복구할 수 있습니다.

DSRM에 대한 드라이버 지원

DSRM을 활성화하고 인스턴스로 부팅하는 방법은 인스턴스가 실행 중인 드라이버에 따라 다릅니다. EC2 콘솔에서는 시스템 로그에서 인스턴스에 대한 드라이버 버전 세부 정보를 볼 수 있습니다. 다음 표에서는 DSRM이 지원되는 드라이버를 보여 줍니다.

드라이버 버전	DSRM 지원 여부	다음 단계
Citrix PV 5.9	아니요	백업에서 인스턴스를 복원합니다. DSRM을 활성화할 수 없습니다.
AWS PV 7.2.0	아니요	이 드라이버에 대한 DSRM은 지원되지 않지만, 그래도 인스턴스에서 루트 볼륨을 분리하고, 볼륨의 스냅샷을 생성하거나 볼륨에서 AMI를 만들고, 이를 보조 볼륨과 동일한 가용 영역에 있는 다른 인스턴스에 연결할 수 있습니다. (이 섹션의 설명처럼) DSRM을 활성화할 수 있습니다.
AWS PV 7.2.2 이상	예	루트 볼륨을 분리하고 다른 인스턴스에 연결하고 DSRM(이 섹션에 설명된 대로)을 활성화합니다.
향상된 네트워킹	예	루트 볼륨을 분리하고 다른 인스턴스에 연결하고 DSRM(이 섹션에 설명된 대로)을 활성화합니다.

향상된 네트워킹을 활성화하는 방법에 대한 자세한 내용은 the section called “ENA(Elastic Network Adapter)” 섹션을 참조하세요. AWS PV 드라이버 업그레이드에 대한 자세한 내용은 Upgrade PV drivers on Windows instances를 참조하세요.

DSRM으로 부팅하도록 인스턴스 구성

EC2 Windows 인스턴스는 운영 체제가 실행되기 전에 네트워크에 연결되지 않습니다. 이러한 이유로 키보드에서 F8 버튼을 눌러 부팅 옵션을 선택할 수 없습니다. 다음 절차 중 하나를 사용하여 DSRM으로 EC2 Windows Server 인스턴스를 부팅할 수 있습니다.

Active Directory가 손상되었는데도 인스턴스가 여전히 실행 중이라고 의심된다면 System Configuration 대화 상자나 명령 프롬프트를 사용하여 DSRM으로 부팅하도록 인스턴스를 구성할 수 있습니다.

System Configuration 대화 상자를 사용하여 DSRM으로 온라인 인스턴스를 부팅하려면

1. 실행 대화 상자에 `msconfig`를 입력하고 Enter 키를 누릅니다.
2. 부팅 탭을 선택합니다.
3. 부팅 옵션 아래에서 안전 부팅을 선택합니다.
4. Active Directory 복구를 선택한 다음 확인을 선택합니다. 시스템에서 서버를 재부팅하라는 메시지를 표시합니다.

명령줄을 사용하여 DSRM으로 온라인 인스턴스를 부팅하려면

명령 프롬프트 창에서 다음 명령을 실행합니다.

```
bcdedit /set safeboot dsrepair
```

인스턴스가 오프라인이고 연결할 수 없으면 루트 볼륨을 분리하고 다른 인스턴스에 연결하여 DSRM 모드를 활성화해야 합니다.

DSRM으로 오프라인 인스턴스를 부팅하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances(인스턴스)를 선택합니다.
3. 영향을 받는 인스턴스를 찾아 선택합니다. 인스턴스 상태, 인스턴스 중지를 차례로 선택합니다.
4. 인스턴스 시작을 선택하고 문제가 발생한 인스턴스와 동일한 가용 영역에 임시 인스턴스를 생성합니다. 다른 버전의 Windows를 사용하는 인스턴스 유형을 선택합니다. 예를 들어, 인스턴스가 Windows Server 2016이면 Windows Server 2019 인스턴스를 선택합니다.

   > **Important**
   >
   > 문제가 발생한 인스턴스와 동일한 가용 영역에서 인스턴스를 생성하지 않는 경우에는 문제가 발생한 인스턴스의 루트 볼륨을 새 인스턴스에 연결할 수 없습니다.

5. 탐색 창에서 볼륨을 선택합니다.
6. 문제가 발생한 인스턴스의 루트 볼륨을 찾습니다. 볼륨을 분리하고 이전에 생성한 임시 인스턴스에 연결합니다. 기본 디바이스 이름(xvdf)으로 연결합니다.
7. 원격 데스크톱을 사용하여 임시 인스턴스에 연결한 후 디스크 관리 유틸리티를 사용하여 볼륨을 사용할 수 있도록 지정합니다.
8. 명령 프롬프트를 열고 다음 명령을 실행합니다. 방금 연결한 보조 볼륨의 실제 드라이브 문자로 D를 바꿉니다.

   ```
   bcdedit /store D:\Boot\BCD /set {default} safeboot dsrepair
   ```

9. 디스크 관리(Disk Management) 유틸리티에서 이전에 연결한 드라이브를 선택하고 컨텍스트(오른쪽 클릭) 메뉴를 열고 오프라인(Offline)을 선택합니다.

10. EC2 콘솔에서 임시 인스턴스로부터 문제가 발생한 볼륨을 분리하고 디바이스 이름 /dev/sda1을 사용하여 원래 인스턴스에 다시 연결합니다. 볼륨을 루트 볼륨으로 지정하려면 이 디바이스 이름을 지정해야 합니다.
11. 인스턴스를 시작합니다.
12. 인스턴스가 EC2 콘솔에서 상태 확인을 통과하면 원격 데스크톱을 사용하여 인스턴스에 연결하고 DSRM 모드로 부팅되는지 확인합니다.
13. (선택 사항) 앞에서 임시로 생성한 인스턴스는 이 절차에서 삭제하거나 중단합니다.

인스턴스의 네트워크 연결이 끊어지거나 예약된 작업이 예정 시간에 실행되지 않음

인스턴스를 다시 시작하면 네트워크 연결이 끊어지는 경우 인스턴스의 시간이 잘못 설정되었을 수 있습니다.

기본적으로 Windows 인스턴스는 협정 세계시(UTC)를 사용합니다. 인스턴스의 시간을 다른 표준 시간대로 설정한 후 인스턴스를 다시 시작하면 시간이 어긋나면서 인스턴스가 일시적으로 IP 주소를 손실합니다. 인스턴스의 네트워크 연결은 이후에 복구되지만 여기에 몇 시간이 걸릴 수 있습니다. 인스턴스의 네트워크 연결이 복구되는 데 걸리는 시간은 UTC와 다른 표준 시간대의 차이에 따라 달라집니다.

이와 같은 시간 문제로 인해 예약된 작업이 예정 시간에 실행되지 않을 수도 있습니다. 이러한 경우 인스턴스의 시간이 정확하지 않기 때문에 예약된 작업이 예정 시간에 실행되지 않는 것입니다.

UTC 이외의 표준 시간대를 지속적으로 사용하려면 RealTimeIsUniversal 레지스트리 키를 설정해야 합니다. 이 키가 없으면 인스턴스가 다시 시작된 후 UTC가 사용됩니다.

네트워크 연결이 끊어지게 만든 시간 문제를 해결하려면 다음을 수행합니다.

1. 권장 PV 드라이버를 사용하고 있는지 확인합니다. 자세한 내용은 the section called “PV 드라이버 업그레이드” 단원을 참조하십시오.
2. HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TimeZoneInformation\RealTimeIsUniversal 레지스트리 키가 있으며 값이 1로 설정되었는지 확인합니다.

콘솔 출력을 가져올 수 없음

Windows 인스턴스의 경우 인스턴스 콘솔은 Windows 부팅 프로세스 중에 수행된 작업의 출력을 표시합니다. Windows가 성공적으로 부팅된 경우 마지막으로 기록되는 메시지는 `Windows is Ready to`

use입니다. 콘솔에 이벤트 로그 메시지를 표시할 수도 있지만 이 기능은 Windows 버전에 따라 기본적으로 활성화되어 있지 않을 수 있습니다. 자세한 내용은 the section called “Windows 시작 에이전트” 단원을 참조하십시오.

Amazon EC2 콘솔을 사용하여 인스턴스에 대한 콘솔 출력을 가져오려면 인스턴스를 선택한 다음 [작업(Actions)], [모니터링 및 문제 해결(Monitor and troubleshoot)], [시스템 로그 가져오기(Get system log)]를 차례로 선택합니다. 명령줄을 사용하여 콘솔 출력을 확인하려면 get-console-output(AWS CLI) 또는 Get-EC2ConsoleOutput(AWS Tools for Windows PowerShell) 명령 중 하나를 사용합니다.

Windows Server 2012 R2 및 이전 버전을 실행하는 인스턴스에서 콘솔 출력이 비어 있는 경우 EC2Config 서비스에 구성 파일 오류 또는 Windows 부팅 실패 등의 문제가 발생한 것일 수 있습니다. 문제를 해결하려면 최신 버전의 EC2Config를 다운로드하여 설치합니다. 자세한 내용은 the section called “EC2Config 설치” 단원을 참조하십시오.

네트워크에서 Windows Server 2012 R2를 이용할 수 없는 경우

네트워크에서 사용할 수 없는 Windows Server 2012 R2 인스턴스의 문제 해결에 대한 자세한 내용은 Windows Server 2012 R2 loses network and storage connectivity after an instance reboot을 참조하세요.

디스크 서명 충돌

EC2Rescue for Windows Server를 사용하여 디스크 서명 충돌을 확인하고 해결할 수 있습니다. 또는 다음 단계를 수행하여 디스크 서명 문제를 수동으로 해결할 수 있습니다.

> **Warning**
>
> 다음 절차에서는 레지스트리 편집기를 사용하여 Windows 레지스트리를 편집하는 방법을 설명합니다. Windows 레지스트리에 대해 또는 레지스트리 편집기를 사용하여 안전하게 수정하는 방법을 잘 알지 못하는 경우 레지스트리 구성을 참조하세요.

1. 명령 프롬프트를 열고 regedit.exe를 입력한 후 Enter 키를 누릅니다.
2. 레지스트리 편집기의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)에서 HKEY_LOCAL_MACHINE을 선택한 후 찾기를 선택합니다.
3. Windows Boot Manager를 입력한 후 다음 찾기를 선택합니다.
4. 키 `11000001`을 선택합니다. 이 키는 이전 단계에서 찾은 키 바로 위의 키입니다.

5. 오른쪽 창에서 Element를 선택한 후 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)에서 수정을 선택합니다.
6. 데이터에서 오프셋 0x38의 4바이트 디스크 서명을 찾습니다. 부팅 구성 데이터베이스 서명(BCD) 입니다. 바이트를 거꾸로 하여 디스크 서명을 만들고 기록해 둡니다. 예를 들어, 다음 데이터가 나타내는 디스크 서명은 E9EB3AA5입니다.

```
...
0030  00 00 00 00 01 00 00 00
0038  A5 3A EB E9 00 00 00 00
0040  00 00 00 00 00 00 00 00
...
```

7. 명령 프롬프트 창에서 다음 명령을 실행하여 Microsoft DiskPart를 시작합니다.

```
diskpart
```

8. `select disk` DiskPart 명령을 실행하고 디스크 서명 충돌이 있는 볼륨의 디스크 번호를 지정합니다.

> **Tip**
>
> 디스크 서명 충돌이 있는 볼륨의 디스크 번호를 확인하려면 디스크 관리 유틸리티를 사용합니다. 명령 프롬프트를 열고 `compmgmt.msc`를 입력한 후 Enter 키를 누릅니다. 왼쪽 탐색 패널에서 디스크 관리를 두 번 클릭합니다. 디스크 관리 유틸리티에서 디스크 서명 충돌이 있는 오프라인 볼륨의 디스크 번호를 확인합니다.

```
DISKPART> select disk 1
Disk 1 is now the selected disk.
```

9. 다음 DiskPart 명령을 실행하여 디스크 서명을 봅니다.

```
DISKPART>  uniqueid disk
Disk ID: 0C764FA8
```

10. 이전 단계에서 표시된 디스크 서명이 앞에서 기록한 디스크 서명과 일치하지 않을 경우 다음 DiskPart 명령을 사용하여 일치하도록 디스크 서명을 변경합니다.

```
DISKPART> uniqueid disk id=E9EB3AA5
```

Amazon EC2 Windows 인스턴스의 Windows 관리자 암호 재설정

Windows 관리자 암호가 분실 또는 만료되어 Windows Amazon EC2 인스턴스에 연결할 수 없는 경우 암호를 재설정할 수 있습니다.

> **Note**
>
> 로컬 관리자 암호를 재설정하는 데 필요한 수동 단계를 자동으로 적용하는 AWS Systems Manager 자동화 문서가 있습니다. 자세한 내용은 AWS Systems Manager 사용 설명서의 EC2 인스턴스의 암호 및 SSH 키 재설정을 참조하세요.

관리자 암호를 수동으로 재설정하는 방식에서는 EC2Launch v2, EC2Config 또는 EC2Launch를 사용합니다.

- EC2Launch v2 에이전트가 포함된 지원되는 모든 Windows AMI에는 EC2Launch v2를 사용합니다.
- Windows Server 2016 이전의 Windows AMI에 대해서는 EC2Config 서비스를 사용합니다.
- Windows Server 2016 이후 AMI에 대해서는 EC2Launch 서비스를 사용합니다.

이러한 절차는 인스턴스 생성에 사용한 키 페어를 분실한 경우 인스턴스에 연결할 수 있는 방법에 대해서도 설명합니다. Amazon EC2는 퍼블릭 키를 사용하여 데이터 조각(예: 암호)을 암호화하고, 프라이빗 키를 사용하여 해당 데이터를 복호화합니다. 퍼블릭 키와 프라이빗 키를 키 페어라고 합니다. Windows 인스턴스에서는 키 페어를 사용하여 관리자 암호를 가져오고 RDP를 사용하여 로그인합니다.

> **Note**
>
> 인스턴스에서 로컬 관리자 계정을 사용 중지했거나 인스턴스가 Systems Manager에 대해 구성된 경우 EC2Rescue 및 Run Command를 사용하여 로컬 관리자 암호를 다시 사용하도록 설정하고 재설정할 수도 있습니다. 자세한 내용은 Use EC2Rescue for Windows Server with Systems Manager Run Command를 참조하세요.

내용

- EC2Launch v2를 사용하여 EC2 인스턴스의 Windows 관리자 암호 재설정
- EC2Launch를 사용하여 EC2 인스턴스의 Windows 관리자 암호 재설정

- EC2Config를 사용하여 EC2 인스턴스의 Windows 관리자 암호 재설정

EC2Launch v2를 사용하여 EC2 인스턴스의 Windows 관리자 암호 재설정

EC2Launch v2 에이전트가 포함된 지원되는 Windows AMI를 사용하고 있는 동안 Windows 관리자 암호를 잊은 경우 EC2Launch v2를 사용하여 새 암호를 생성할 수 있습니다.

EC2Launch v2 에이전트가 포함되지 않은 Windows Server 2016 이상의 AMI를 사용하는 경우 EC2Launch를 사용하여 EC2 인스턴스의 Windows 관리자 암호 재설정 섹션을 참조하세요.

EC2Launch v2 에이전트가 포함되지 않은 Windows Server 2016 이전 버전의 Windows Server AMI를 사용하는 경우 EC2Config를 사용하여 EC2 인스턴스의 Windows 관리자 암호 재설정 섹션을 참조하세요.

> **Note**
>
> 인스턴스에서 로컬 관리자 계정을 사용 중지했거나 인스턴스가 Systems Manager에 대해 구성된 경우 EC2Rescue 및 Run Command를 사용하여 로컬 관리자 암호를 다시 사용하도록 설정하고 재설정할 수도 있습니다. 자세한 내용은 Use EC2Rescue for Windows Server with Systems Manager Run Command를 참조하세요.

> **Note**
>
> 로컬 관리자 암호를 재설정하는 데 필요한 수동 단계를 자동으로 적용하는 AWS Systems Manager 자동화 문서가 있습니다. 자세한 내용은 AWS Systems Manager 사용 설명서의 EC2 인스턴스의 암호 및 SSH 키 재설정을 참조하세요.

EC2Launch v2를 사용하여 Windows 관리자 암호를 재설정하려면 다음 작업을 수행해야 합니다.

- 1단계: EC2Launch v2 에이전트가 실행 중인지 확인
- 2단계: 인스턴스에서 루트 볼륨 분리
- 3단계: 임시 인스턴스에 볼륨 연결
- 4단계: .run-once 파일 삭제
- 5단계: 원본 인스턴스에서 다시 시작

1단계: EC2Launch v2 에이전트가 실행 중인지 확인

관리자 암호를 재설정하기 전에 EC2Launch v2 에이전트가 설치되어 실행 중인지 확인합니다. 이 단원의 뒷부분에서 EC2Launch v2 에이전트를 사용하여 관리자 암호를 재설정하게 됩니다.

EC2Launch v2 에이전트가 실행 중인지 확인하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [인스턴스(Instances)]를 선택한 후 암호 재설정이 필요한 인스턴스를 선택합니다. 이 절차에서는 이 인스턴스를 원본 인스턴스라고 합니다.
3. [작업(Actions)], [모니터링 및 문제 해결(Monitor and troubleshoot)], [시스템 로그 가져오기(Get system log)]를 선택합니다.
4. EC2 시작 항목(예 시작: EC2Launch v2 서비스 v2.0.124)을 찾습니다. 이 항목이 보이면 EC2Launch v2 서비스가 실행 중인 것입니다.

   시스템 로그 출력이 비어 있거나 EC2Launch v2 에이전트가 실행 중이지 않을 경우 인스턴스 콘솔 스크린샷 서비스를 사용하여 인스턴스 문제를 해결합니다. 자세한 내용은 연결할 수 없는 인스턴스의 스크린샷 캡처 단원을 참조하십시오.

2단계: 인스턴스에서 루트 볼륨 분리

암호가 저장된 볼륨이 인스턴스에 루트 볼륨으로 연결되어 있는 경우 EC2Launch v2를 사용하여 관리자 암호를 재설정할 수 없습니다. 원본 인스턴스에서 볼륨을 분리해야 이 볼륨을 임시 인스턴스에 부 볼륨으로 연결할 수 있습니다.

인스턴스에서 루트 볼륨을 분리하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances(인스턴스)를 선택합니다.
3. 암호 재설정이 필요한 인스턴스를 선택하고 인스턴스 상태, 인스턴스 중지를 선택합니다. 인스턴스 상태가 [중지됨(Stopped)]으로 변경되면 다음 단계를 계속합니다.
4. (선택 사항) 이 인스턴스를 시작할 때 지정한 프라이빗 키가 있는 경우 다음 단계로 계속합니다. 그렇게 하지 않으면 다음 단계를 사용하여 새 키 페어를 사용하여 시작한 새 인스턴스로 인스턴스를 바꿉니다.

   a. Amazon EC2 콘솔을 사용하여 새 키 페어를 생성합니다. 새 키 페어에 분실한 프라이빗 키와 동일한 이름을 지정하려면 먼저 기존 키 페어를 삭제해야 합니다.

b. 바꿀 인스턴스를 선택합니다. 인스턴스의 인스턴스 유형, VPC, 서브넷, 보안 그룹 및 IAM 역할을 기록해 둡니다.

c. 인스턴스를 선택하고 작업, 이미지 및 템플릿, 이미지 생성을 선택합니다. 이미지 이름과 설명을 입력하고 [이미지 생성(Create image)]을 선택합니다.

d. 탐색 창에서 AMI를 선택합니다. 이미지 상태가 사용 가능으로 변경될 때까지 기다립니다. 그런 다음 이미지를 선택하고 AMI로 인스턴스 시작을 선택합니다.

e. 필드를 완성하여 인스턴스를 시작하고 바꿀 인스턴스와 동일한 인스턴스 유형, VPC, 서브넷, 보안 그룹 및 IAM 역할을 선택한 후 인스턴스 시작을 선택합니다.

f. 메시지가 나타나면 새 인스턴스를 위해 생성한 키 페어를 선택한 다음 인스턴스 시작을 선택합니다.

g. (선택 사항) 원본 인스턴스에 연결된 탄력적 IP 주소가 있는 경우 이 주소를 새 인스턴스와 연결합니다. 원본 인스턴스에 루트 볼륨 외에도 EBS 볼륨이 있는 경우 해당 볼륨을 새 인스턴스로 전송합니다.

5. 다음과 같은 방법으로 루트 볼륨을 원본 인스턴스에서 분리합니다.

a. 원본 인스턴스를 선택하고 스토리지 탭을 선택합니다. 루트 디바이스 이름 아래에 루트 디바이스 이름을 기록합니다. 블록 디바이스에서 이 디바이스 이름을 가진 볼륨을 찾아 볼륨 ID를 기록합니다.

b. 탐색 창에서 볼륨을 선택합니다.

c. 볼륨 목록에서 루트 디바이스로 표시한 볼륨을 선택하고 작업, 볼륨 분리를 선택합니다. 볼륨 상태가 사용 가능로 변경된 후 이어서 다음 단계를 수행합니다.

6. 원본 인스턴스를 대체하기 위해 새 인스턴스를 생성한 경우 지금 원본 인스턴스를 종료할 수 있습니다. 더 이상 필요하지 않습니다. 이 절차의 나머지 단계에서 원본 인스턴스에 대한 모든 참조는 생성한 새 인스턴스에 적용됩니다.

3단계: 임시 인스턴스에 볼륨 연결

그다음에는 임시 인스턴스를 시작하여 이 인스턴스에 볼륨을 부 볼륨으로 연결합니다. 이것은 구성 파일을 수정하는 데 사용하는 인스턴스입니다.

임시 인스턴스를 시작하고 볼륨을 연결하려면

1. 다음과 같이 임시 인스턴스를 시작합니다.

a. 탐색 창에서 [인스턴스(Instances)], [인스턴스 시작(Launch instances)]을 차례로 선택한 후 AMI를 선택합니다.

> **Important**
>
> 디스크 서명 충돌을 방지하려면 다른 Windows 버전에 대한 AMI를 선택해야 합니다. 예를 들어 원본 인스턴스에서 Windows Server 2019를 실행하는 경우 Windows Server 2016용 기본 AMI를 사용하여 임시 인스턴스를 시작합니다.

b. 기본 인스턴스 유형을 그대로 두고 다음: 인스턴스 세부 정보 구성을 선택합니다.

c. 인스턴스 세부 정보 구성 페이지에서 서브넷에 대해 원래 인스턴스와 동일한 가용 영역을 선택한 다음 검토 및 시작을 선택합니다.

> **Important**
>
> 임시 인스턴스는 원래 인스턴스와 동일한 가용 영역에 있어야 합니다. 임시 인스턴스가 다른 가용 영역에 있으면 원래 인스턴스의 루트 볼륨을 인스턴스에 연결할 수 없습니다.

d. 인스턴스 시작 검토 페이지에서 시작을 선택합니다.

e. 메시지가 나타나면 새 키 페어를 생성하고, 컴퓨터에서 안전한 위치에 다운로드한 후 인스턴스 시작을 선택합니다.

2. 다음과 같은 방법으로 볼륨을 임시 인스턴스에 부 볼륨으로 연결합니다.

a. 탐색 창에서 볼륨을 선택하고 원본 인스턴스에서 분리한 볼륨을 선택한 후 작업, 볼륨 연결을 선택합니다.

b. 볼륨 연결 대화 상자에서 인스턴스에 임시 인스턴스의 이름이나 ID를 입력한 후 목록에서 해당 인스턴스를 선택합니다.

c. 디바이스에 대해 **xvdf**를 입력하고(아직 없는 경우) 연결을 선택합니다.

4단계: .run-once 파일 삭제

이제 인스턴스에 연결된 오프라인 볼륨에서 `.run-once` 파일을 삭제해야 합니다. 이렇게 하면 EC2Launch v2에서 관리자 암호 설정을 포함하여 빈도가 once인 모든 태스크를 실행합니다. 연결한 보조 볼륨의 파일 경로는 `D:\ProgramData\Amazon\EC2Launch\state\.run-once`와(과) 비슷합니다.

.run-once 파일을 삭제하는 방법

1. 디스크 관리 유틸리티를 열고 Make an Amazon EBS volume available for use 지침에 따라 드라이브를 온라인 상태로 만듭니다.
2. 온라인 상태의 디스크에서 `.run-once` 파일을 찾습니다.
3. .run-once 파일을 삭제합니다.

> **Important**
> 한 번 실행하도록 설정된 모든 스크립트가 이 작업에 의해 트리거됩니다.

5단계: 원본 인스턴스에서 다시 시작

`.run-once` 파일을 삭제한 후 볼륨을 원본 인스턴스에 루트 볼륨으로 다시 연결하고 키 페어를 사용해 인스턴스에 연결하여 관리자 암호를 검색합니다.

1. 다음과 같은 방법으로 볼륨을 원본 인스턴스에 다시 연결합니다.
   a. 탐색 창에서 볼륨을 선택하고 임시 인스턴스에서 분리한 볼륨을 선택한 후 작업, 볼륨 연결을 선택합니다.
   b. 볼륨 연결 대화 상자에서 인스턴스에 원본 인스턴스의 이름이나 ID를 입력한 다음 해당 인스턴스를 선택합니다.
   c. 디바이스에 **`/dev/sda1`**을 입력합니다.
   d. 연결을 선택합니다. 볼륨 상태가 `in-use`로 변경된 후 이어서 다음 단계를 수행합니다.
2. 탐색 창에서 Instances(인스턴스)를 선택합니다. 원본 인스턴스를 선택하고 [인스턴스 상태(Instance state)], [인스턴스 시작(Start instance)]을 차례로 선택합니다. 인스턴스 상태가 `Running`로 변경된 후 이어서 다음 단계를 수행합니다.
3. 새 키 페어의 프라이빗 키를 사용하여 새 Windows 관리자 암호를 가져온 다음 인스턴스에 연결합니다. 자세한 내용은 RDP를 사용하여 Windows 인스턴스에 연결 단원을 참조하십시오.

> **Important**
> 인스턴스를 중지했다가 시작하면 인스턴스가 새 퍼블릭 IP 주소를 가져옵니다. 현재 퍼블릭 DNS 이름을 사용하여 인스턴스에 연결해야 합니다. 자세한 내용은 Amazon EC2 인스턴스 상태 변경 단원을 참조하십시오.

4. (선택 사항) 임시 인스턴스를 더 이상 사용하지 않는 경우 해당 인스턴스는 종료해도 됩니다. 임시 인스턴스를 선택하고 [인스턴스 상태(Instance State)], [인스턴스 종료(Terminate instance)]를 차례로 선택합니다.

EC2Launch를 사용하여 EC2 인스턴스의 Windows 관리자 암호 재설정

Windows 관리자 암호를 잊어버렸고 Windows Server 2016 이상의 AMI를 사용하는 경우 EC2Launch 서비스로 새 암호를 생성하는 EC2Rescue 도구를 사용할 수 있습니다.

EC2Launch v2 에이전트가 포함되지 않은 Windows Server 2016 이상의 AMI를 사용하는 경우, EC2Launch v2를 사용하여 새 암호를 생성할 수 있습니다.

Windows Server 2016 이전의 Windows Server AMI를 사용하는 경우 EC2Config를 사용하여 EC2 인스턴스의 Windows 관리자 암호 재설정 단원을 참조하십시오.

Warning

인스턴스를 중지하면 인스턴스 스토어 볼륨의 데이터가 삭제됩니다. 인스턴스 스토어 볼륨의 데이터를 유지하려면 영구 스토리지에 백업하세요.

Note

인스턴스에서 로컬 관리자 계정을 사용 중지했거나 인스턴스가 Systems Manager에 대해 구성된 경우 EC2Rescue 및 Run Command를 사용하여 로컬 관리자 암호를 다시 사용하도록 설정하고 재설정할 수도 있습니다. 자세한 내용은 Use EC2Rescue for Windows Server with Systems Manager Run Command를 참조하세요.

Note

로컬 관리자 암호를 재설정하는 데 필요한 수동 단계를 자동으로 적용하는 AWS Systems Manager 자동화 문서가 있습니다. 자세한 내용은 AWS Systems Manager 사용 설명서의 EC2 인스턴스의 암호 및 SSH 키 재설정을 참조하세요.

EC2Launch를 사용하여 Windows 관리자 암호를 재설정하려면 다음 작업을 수행해야 합니다.

- 1단계: 인스턴스에서 루트 볼륨 분리
- 2단계: 임시 인스턴스에 볼륨 연결
- 3단계: 관리자 암호 재설정
- 4단계: 원본 인스턴스 다시 시작

1단계: 인스턴스에서 루트 볼륨 분리

암호가 저장된 볼륨이 인스턴스에 루트 볼륨으로 연결되어 있는 경우 EC2Launch 서비스를 사용하여 관리자 암호를 재설정할 수 없습니다. 원본 인스턴스에서 볼륨을 분리해야 이 볼륨을 임시 인스턴스에 부 볼륨으로 연결할 수 있습니다.

인스턴스에서 루트 볼륨을 분리하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances(인스턴스)를 선택합니다.
3. 암호 재설정이 필요한 인스턴스를 선택하고 인스턴스 상태, 인스턴스 중지를 선택합니다. 인스턴스 상태가 [중지됨(Stopped)]으로 변경되면 다음 단계를 계속합니다.
4. (선택 사항) 이 인스턴스를 시작할 때 지정한 프라이빗 키가 있는 경우 다음 단계로 계속합니다. 그렇게 하지 않으면 다음 단계를 사용하여 새 키 페어를 사용하여 시작한 새 인스턴스로 인스턴스를 바꿉니다.

    a. Amazon EC2 콘솔을 사용하여 새 키 페어를 생성합니다. 새 키 페어에 분실한 프라이빗 키와 동일한 이름을 지정하려면 먼저 기존 키 페어를 삭제해야 합니다.

    b. 바꿀 인스턴스를 선택합니다. 인스턴스의 인스턴스 유형, VPC, 서브넷, 보안 그룹 및 IAM 역할을 기록해 둡니다.

    c. 인스턴스를 선택하고 작업, 이미지 및 템플릿, 이미지 생성을 선택합니다. 이미지 이름과 설명을 입력하고 [이미지 생성(Create image)]을 선택합니다.

    d. 탐색 창에서 AMI를 선택합니다. 이미지 상태가 사용 가능으로 변경될 때까지 기다립니다. 그런 다음 이미지를 선택하고 AMI로 인스턴스 시작을 선택합니다.

    e. 필드를 완성하여 인스턴스를 시작하고 바꿀 인스턴스와 동일한 인스턴스 유형, VPC, 서브넷, 보안 그룹 및 IAM 역할을 선택한 후 인스턴스 시작을 선택합니다.

    f. 메시지가 나타나면 새 인스턴스를 위해 생성한 키 페어를 선택한 다음 인스턴스 시작을 선택합니다.

g. (선택 사항) 원본 인스턴스에 연결된 탄력적 IP 주소가 있는 경우 이 주소를 새 인스턴스와 연결합니다. 원본 인스턴스에 루트 볼륨 외에도 EBS 볼륨이 있는 경우 해당 볼륨을 새 인스턴스로 전송합니다.

5. 다음과 같은 방법으로 루트 볼륨을 원본 인스턴스에서 분리합니다.

   a. 원본 인스턴스를 선택하고 스토리지 탭을 선택합니다. 루트 디바이스 이름 아래에 루트 디바이스 이름을 기록합니다. 블록 디바이스에서 이 디바이스 이름을 가진 볼륨을 찾아 볼륨 ID를 기록합니다.

   b. 탐색 창에서 볼륨을 선택합니다.

   c. 볼륨 목록에서 루트 디바이스로 표시한 볼륨을 선택하고 작업, 볼륨 분리를 선택합니다. 볼륨 상태가 사용 가능로 변경된 후 이어서 다음 단계를 수행합니다.

6. 원본 인스턴스를 대체하기 위해 새 인스턴스를 생성한 경우 지금 원본 인스턴스를 종료할 수 있습니다. 더 이상 필요하지 않습니다. 이 절차의 나머지 단계에서 원본 인스턴스에 대한 모든 참조는 생성한 새 인스턴스에 적용됩니다.

2단계: 임시 인스턴스에 볼륨 연결

그다음에는 임시 인스턴스를 시작하여 이 인스턴스에 볼륨을 부 볼륨으로 연결합니다. 이것은 EC2Launch를 실행하는 데 사용하는 인스턴스입니다.

임시 인스턴스를 시작하고 볼륨을 연결하려면

1. 다음과 같이 임시 인스턴스를 시작합니다.

   a. 탐색 창에서 [인스턴스(Instances)], [인스턴스 시작(Launch instances)]을 차례로 선택한 후 AMI를 선택합니다.

   > **Important**
   >
   > 디스크 서명 충돌을 방지하려면 다른 Windows 버전에 대한 AMI를 선택해야 합니다. 예를 들어 원본 인스턴스에서 Windows Server 2019를 실행하는 경우 Windows Server 2016용 기본 AMI를 사용하여 임시 인스턴스를 시작합니다.

   b. 기본 인스턴스 유형을 그대로 두고 다음: 인스턴스 세부 정보 구성을 선택합니다.

   c. 인스턴스 세부 정보 구성 페이지에서 서브넷에 대해 원래 인스턴스와 동일한 가용 영역을 선택한 다음 검토 및 시작을 선택합니다.

> **Important**
>
> 임시 인스턴스는 원래 인스턴스와 동일한 가용 영역에 있어야 합니다. 임시 인스턴스가 다른 가용 영역에 있으면 원래 인스턴스의 루트 볼륨을 인스턴스에 연결할 수 없습니다.

d. 인스턴스 시작 검토 페이지에서 시작을 선택합니다.

e. 메시지가 나타나면 새 키 페어를 생성하고, 컴퓨터에서 안전한 위치에 다운로드한 후 인스턴스 시작을 선택합니다.

2. 다음과 같은 방법으로 볼륨을 임시 인스턴스에 부 볼륨으로 연결합니다.

a. 탐색 창에서 볼륨을 선택하고 원본 인스턴스에서 분리한 볼륨을 선택한 후 작업, 볼륨 연결을 선택합니다.

b. 볼륨 연결 대화 상자에서 인스턴스에 임시 인스턴스의 이름이나 ID를 입력한 후 목록에서 해당 인스턴스를 선택합니다.

c. 디바이스에 대해 **xvdf**를 입력하고(아직 없는 경우) 연결을 선택합니다.

3단계: 관리자 암호 재설정

그다음에는 임시 인스턴스에 연결한 후 EC2Launch를 사용하여 관리자 암호를 재설정합니다.

관리자 암호를 재설정하려면

1. 임시 인스턴스에 연결하고 인스턴스에서 EC2Rescue for Windows Server 도구를 사용하여 관리자 암호를 다음과 같이 재설정합니다.

a. EC2Rescue for Windows Server zip 파일을 다운로드하여 압축을 푼 후 EC2Rescue.exe를 실행합니다.

b. 라이선스 계약에서 라이선스 계약을 읽고, 약관에 동의하는 경우 I Agree(동의함)를 선택합니다.

c. EC2Rescue for Windows Server 시작 화면에서 다음을 선택합니다.

d. Select mode(모드 선택)에서 Offline instance(오프라인 인스턴스)를 선택합니다.

e. Select a disk(디스크 선택) 화면에서 xvdf 디바이스를 선택하고 다음을 선택합니다.

f. 디스크 선택을 확인한 후 예를 선택합니다.

g. 볼륨 로드가 완료되면 확인을 선택합니다.

h. Select Offline Instance Option(오프라인 인스턴스 옵션 선택)에서 Diagnose and Rescue(진단 및 복구)를 선택합니다.

i. 요약 화면에서 정보를 검토한 다음 다음을 선택합니다.

j. Detected possible issues(감지된 잠재적 문제) 화면에서 Reset Administrator Password(관리자 암호 재설정)를 선택하고 다음을 선택합니다.

k. 확인 화면에서 Rescue(복구), 확인을 선택합니다.

l. 완료 화면에서 완료를 선택합니다.

m. EC2Rescue for Windows Server 도구를 종료하고 임시 인스턴스에서 연결을 해제한 후 Amazon EC2 콘솔로 돌아갑니다.

2. 다음과 같은 방법으로 부(`xvdf`) 볼륨을 임시 인스턴스에서 분리합니다.

a. 탐색 창에서 인스턴스를 선택하고 임시 인스턴스를 선택합니다.

b. 임시 인스턴스의 [스토리지(Storage)] 탭에서 xvdf로 나열된 EBS 볼륨의 ID를 확인합니다.

c. 탐색 창에서 볼륨을 선택합니다.

d. 앞 단계에서 적어둔 볼륨을 볼륨 목록에서 선택한 다음 작업, Detach Volume(볼륨 분리)을 선택합니다. 볼륨 상태가 사용 가능로 변경된 후 이어서 다음 단계를 수행합니다.

4단계: 원본 인스턴스 다시 시작

EC2Launch를 사용해 관리자 암호를 재설정한 후 볼륨을 원본 인스턴스에 루트 볼륨으로 다시 연결하고 키 페어를 사용해 인스턴스에 연결하여 관리자 암호를 검색합니다.

원본 인스턴스를 다시 시작하려면

1. 다음과 같은 방법으로 볼륨을 원본 인스턴스에 다시 연결합니다.

a. 탐색 창에서 볼륨을 선택하고 임시 인스턴스에서 분리한 볼륨을 선택한 후 작업, 볼륨 연결을 선택합니다.

b. 볼륨 연결 대화 상자에서 인스턴스에 원본 인스턴스의 이름이나 ID를 입력한 다음 해당 인스턴스를 선택합니다.

c. 디바이스에 **`/dev/sda1`**을 입력합니다.

d. 연결을 선택합니다. 볼륨 상태가 `in-use`로 변경된 후 이어서 다음 단계를 수행합니다.

2. 탐색 창에서 Instances(인스턴스)를 선택합니다. 원본 인스턴스를 선택하고 [인스턴스 상태(Instance state)], [인스턴스 시작(Start instance)]을 차례로 선택합니다. 인스턴스 상태가 `Running`로 변경된 후 이어서 다음 단계를 수행합니다.

3. 새 키 페어의 프라이빗 키를 사용하여 새 Windows 관리자 암호를 가져온 다음 인스턴스에 연결합니다. 자세한 내용은 RDP를 사용하여 Windows 인스턴스에 연결 단원을 참조하십시오.
4. (선택 사항) 임시 인스턴스를 더 이상 사용하지 않는 경우 해당 인스턴스는 종료해도 됩니다. 임시 인스턴스를 선택하고 [인스턴스 상태(Instance State)], [인스턴스 종료(Terminate instance)]를 차례로 선택합니다.

EC2Config를 사용하여 EC2 인스턴스의 Windows 관리자 암호 재설정

Windows Server 2016 이전의 Windows AMI를 사용하고 있는 동안 Windows 관리자 암호를 잊은 경우 EC2Config 에이전트를 사용하여 새 암호를 생성할 수 있습니다.

Windows Server 2016 이상의 AMI를 사용하는 경우 EC2Launch를 사용하여 EC2 인스턴스의 Windows 관리자 암호 재설정 섹션을 참조하세요. 또는 EC2Launch 서비스로 새 암호를 생성하는 EC2Rescue 도구를 사용할 수 있습니다.

> **Note**
>
> 인스턴스에서 로컬 관리자 계정을 사용 중지했거나 인스턴스가 Systems Manager에 대해 구성된 경우 EC2Rescue 및 Run Command를 사용하여 로컬 관리자 암호를 다시 사용하도록 설정하고 재설정할 수도 있습니다. 자세한 내용은 Use EC2Rescue for Windows Server with Systems Manager Run Command를 참조하세요.

> **Note**
>
> 로컬 관리자 암호를 재설정하는 데 필요한 수동 단계를 자동으로 적용하는 AWS Systems Manager 자동화 문서가 있습니다. 자세한 내용은 AWS Systems Manager 사용 설명서의 EC2 인스턴스의 암호 및 SSH 키 재설정을 참조하세요.

EC2Config를 사용하여 Windows 관리자 암호를 재설정하려면 다음 작업을 수행해야 합니다.

- 1단계: EC2Config 서비스가 실행 중인지 확인
- 2단계: 인스턴스에서 루트 볼륨 분리
- 3단계: 임시 인스턴스에 볼륨 연결
- 4단계: 구성 파일 수정
- 5단계: 원본 인스턴스에서 다시 시작

1단계: EC2Config 서비스가 실행 중인지 확인

관리자 암호를 재설정하기 전에 EC2Config 서비스가 설치되어 실행 중인지 확인합니다. 이 단원의 뒷부분에서 EC2Config 서비스를 사용하여 관리자 암호를 재설정하게 됩니다.

EC2Config 서비스가 실행 중인지 확인하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [인스턴스(Instances)]를 선택한 후 암호 재설정이 필요한 인스턴스를 선택합니다. 이 절차에서는 이 인스턴스를 원본 인스턴스라고 합니다.
3. [작업(Actions)], [모니터링 및 문제 해결(Monitor and troubleshoot)], [시스템 로그 가져오기(Get system log)]를 선택합니다.
4. EC2 에이전트 항목을 찾습니다(예: EC2 에이전트: Ec2Config 서비스 v3.18.1118). 이 항목이 보이면 EC2Config 서비스가 실행 중인 것입니다.

   시스템 로그 출력이 비어 있거나 EC2Config 서비스가 실행 중이지 않을 경우 인스턴스 콘솔 스크린샷 서비스를 사용하여 인스턴스 문제를 해결합니다. 자세한 내용은 연결할 수 없는 인스턴스의 스크린샷 캡처 단원을 참조하십시오.

2단계: 인스턴스에서 루트 볼륨 분리

암호가 저장된 볼륨이 인스턴스에 루트 볼륨으로 연결되어 있는 경우 EC2Config 서비스를 사용하여 관리자 암호를 재설정할 수 없습니다. 원본 인스턴스에서 볼륨을 분리해야 이 볼륨을 임시 인스턴스에 부 볼륨으로 연결할 수 있습니다.

인스턴스에서 루트 볼륨을 분리하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances(인스턴스)를 선택합니다.
3. 암호 재설정이 필요한 인스턴스를 선택하고 인스턴스 상태, 인스턴스 중지를 선택합니다. 인스턴스 상태가 [중지됨(Stopped)]으로 변경되면 다음 단계를 계속합니다.
4. (선택 사항) 이 인스턴스를 시작할 때 지정한 프라이빗 키가 있는 경우 다음 단계로 계속합니다. 그렇게 하지 않으면 다음 단계를 사용하여 새 키 페어를 사용하여 시작한 새 인스턴스로 인스턴스를 바꿉니다.

   a. Amazon EC2 콘솔을 사용하여 새 키 페어를 생성합니다. 새 키 페어에 분실한 프라이빗 키와 동일한 이름을 지정하려면 먼저 기존 키 페어를 삭제해야 합니다.

b. 바꿀 인스턴스를 선택합니다. 인스턴스의 인스턴스 유형, VPC, 서브넷, 보안 그룹 및 IAM 역할을 기록해 둡니다.

c. 인스턴스를 선택하고 작업, 이미지 및 템플릿, 이미지 생성을 선택합니다. 이미지 이름과 설명을 입력하고 [이미지 생성(Create image)]을 선택합니다.

d. 탐색 창에서 AMI를 선택합니다. 이미지 상태가 사용 가능으로 변경될 때까지 기다립니다. 그런 다음 이미지를 선택하고 AMI로 인스턴스 시작을 선택합니다.

e. 필드를 완성하여 인스턴스를 시작하고 바꿀 인스턴스와 동일한 인스턴스 유형, VPC, 서브넷, 보안 그룹 및 IAM 역할을 선택한 후 인스턴스 시작을 선택합니다.

f. 메시지가 나타나면 새 인스턴스를 위해 생성한 키 페어를 선택한 다음 인스턴스 시작을 선택합니다.

g. (선택 사항) 원본 인스턴스에 연결된 탄력적 IP 주소가 있는 경우 이 주소를 새 인스턴스와 연결합니다. 원본 인스턴스에 루트 볼륨 외에도 EBS 볼륨이 있는 경우 해당 볼륨을 새 인스턴스로 전송합니다.

5. 다음과 같은 방법으로 루트 볼륨을 원본 인스턴스에서 분리합니다.

a. 원본 인스턴스를 선택하고 스토리지 탭을 선택합니다. 루트 디바이스 이름 아래에 루트 디바이스 이름을 기록합니다. 블록 디바이스에서 이 디바이스 이름을 가진 볼륨을 찾아 볼륨 ID를 기록합니다.

b. 탐색 창에서 볼륨을 선택합니다.

c. 볼륨 목록에서 루트 디바이스로 표시한 볼륨을 선택하고 작업, 볼륨 분리를 선택합니다. 볼륨 상태가 사용 가능로 변경된 후 이어서 다음 단계를 수행합니다.

6. 원본 인스턴스를 대체하기 위해 새 인스턴스를 생성한 경우 지금 원본 인스턴스를 종료할 수 있습니다. 더 이상 필요하지 않습니다. 이 절차의 나머지 단계에서 원본 인스턴스에 대한 모든 참조는 생성한 새 인스턴스에 적용됩니다.

3단계: 임시 인스턴스에 볼륨 연결

그다음에는 임시 인스턴스를 시작하여 이 인스턴스에 볼륨을 부 볼륨으로 연결합니다. 이것은 구성 파일을 수정하는 데 사용하는 인스턴스입니다.

임시 인스턴스를 시작하고 볼륨을 연결하려면

1. 다음과 같이 임시 인스턴스를 시작합니다.

a. 탐색 창에서 [인스턴스(Instances)], [인스턴스 시작(Launch instances)]을 차례로 선택한 후 AMI를 선택합니다.

> **Important**
>
> 디스크 서명 충돌을 방지하려면 다른 Windows 버전에 대한 AMI를 선택해야 합니다. 예를 들어 원본 인스턴스에서 Windows Server 2019를 실행하는 경우 Windows Server 2016용 기본 AMI를 사용하여 임시 인스턴스를 시작합니다.

b. 기본 인스턴스 유형을 그대로 두고 다음: 인스턴스 세부 정보 구성을 선택합니다.

c. 인스턴스 세부 정보 구성 페이지에서 서브넷에 대해 원래 인스턴스와 동일한 가용 영역을 선택한 다음 검토 및 시작을 선택합니다.

> **Important**
>
> 임시 인스턴스는 원래 인스턴스와 동일한 가용 영역에 있어야 합니다. 임시 인스턴스가 다른 가용 영역에 있으면 원래 인스턴스의 루트 볼륨을 인스턴스에 연결할 수 없습니다.

d. 인스턴스 시작 검토 페이지에서 시작을 선택합니다.

e. 메시지가 나타나면 새 키 페어를 생성하고, 컴퓨터에서 안전한 위치에 다운로드한 후 인스턴스 시작을 선택합니다.

2. 다음과 같은 방법으로 볼륨을 임시 인스턴스에 부 볼륨으로 연결합니다.

a. 탐색 창에서 볼륨을 선택하고 원본 인스턴스에서 분리한 볼륨을 선택한 후 작업, 볼륨 연결을 선택합니다.

b. 볼륨 연결 대화 상자에서 인스턴스에 임시 인스턴스의 이름이나 ID를 입력한 후 목록에서 해당 인스턴스를 선택합니다.

c. 디바이스에 대해 **xvdf**를 입력하고(아직 없는 경우) 연결을 선택합니다.

4단계: 구성 파일 수정

볼륨을 임시 인스턴스에 부 볼륨으로 연결한 후 구성 파일에서 `Ec2SetPassword` 플러그인을 수정합니다.

구성 파일을 수정하려면

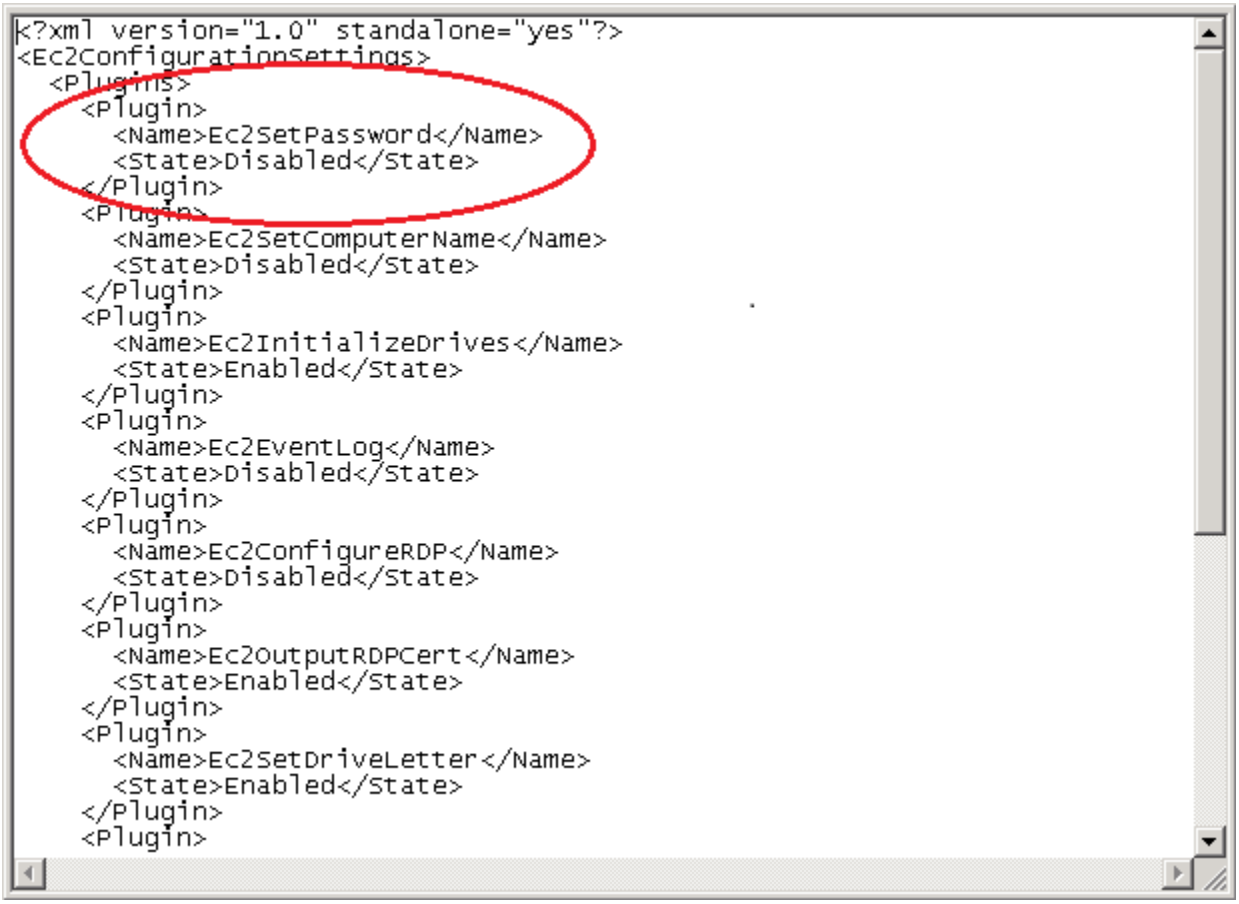
1. 임시 인스턴스에서 부 볼륨에 있는 구성 파일을 다음과 같이 수정합니다.

a. 임시 인스턴스를 시작하여 이 인스턴스에 연결합니다.

b. 드라이브를 온라인 상태로 만들려면 Amazon EBS 볼륨을 사용할 수 있도록 만들기의 지침을 따르세요.

c. 두 번째 볼륨으로 이동한 다음, 메모장과 같은 텍스트 편집기를 사용하여 `\Program Files\Amazon\Ec2ConfigService\Settings\config.xml`을 엽니다.

d. 스크린샷에 표시된 것처럼 파일 맨 위에서 이름이 `Ec2SetPassword`인 플러그인을 찾습니다. 상태를 `Disabled`에서 `Enabled`로 변경하고 파일을 저장합니다.



2. 구성 파일을 수정한 후에 다음과 같이 임시 인스턴스에서 부 볼륨을 분리합니다.

a. Disk Management(디스크 관리) 유틸리티를 사용하여 볼륨을 오프라인으로 설정합니다.

b. 임시 인스턴스에서 연결을 해제하고 Amazon EC2 콘솔로 돌아갑니다.

c. 탐색 창에서 볼륨을 선택하고 해당 볼륨을 선택한 후 작업, Detach Volume(볼륨 분리)을 선택합니다. 볼륨의 상태가 사용 가능으로 변경되면 다음 단계로 계속합니다.

5단계: 원본 인스턴스에서 다시 시작

구성 파일을 수정한 후 볼륨을 원본 인스턴스에 루트 볼륨으로 다시 연결하고 키 페어를 사용해 인스턴스에 연결하여 관리자 암호를 검색합니다.

1. 다음과 같은 방법으로 볼륨을 원본 인스턴스에 다시 연결합니다.
   a. 탐색 창에서 볼륨을 선택하고 임시 인스턴스에서 분리한 볼륨을 선택한 후 작업, 볼륨 연결을 선택합니다.
   b. 볼륨 연결 대화 상자에서 인스턴스에 원본 인스턴스의 이름이나 ID를 입력한 다음 해당 인스턴스를 선택합니다.
   c. 디바이스에 **`/dev/sda1`**을 입력합니다.
   d. 연결을 선택합니다. 볼륨 상태가 `in-use`로 변경된 후 이어서 다음 단계를 수행합니다.
2. 탐색 창에서 Instances(인스턴스)를 선택합니다. 원본 인스턴스를 선택하고 [인스턴스 상태(Instance state)], [인스턴스 시작(Start instance)]을 차례로 선택합니다. 인스턴스 상태가 `Running`로 변경된 후 이어서 다음 단계를 수행합니다.
3. 새 키 페어의 프라이빗 키를 사용하여 새 Windows 관리자 암호를 가져온 다음 인스턴스에 연결합니다. 자세한 내용은 RDP를 사용하여 Windows 인스턴스에 연결 단원을 참조하십시오.

   > **Important**
   > 인스턴스를 중지했다가 시작하면 인스턴스가 새 퍼블릭 IP 주소를 가져옵니다. 현재 퍼블릭 DNS 이름을 사용하여 인스턴스에 연결해야 합니다. 자세한 내용은 Amazon EC2 인스턴스 상태 변경 단원을 참조하십시오.

4. (선택 사항) 임시 인스턴스를 더 이상 사용하지 않는 경우 해당 인스턴스는 종료해도 됩니다. 임시 인스턴스를 선택하고 [인스턴스 상태(Instance State)], [인스턴스 종료(Terminate instance)]를 차례로 선택합니다.

Amazon EC2 Windows 인스턴스 Sysprep 문제 해결

이미지 준비 중에 문제가 발생하거나 오류 메시지를 받는 경우 다음 로그를 검토하세요. 로그 위치는 Sysprep에서 EC2Config, EC2Launch v1 또는 EC2Launch v2 중 무엇을 실행하는지에 따라 다릅니다.

- `%WINDIR%\Panther\Unattendgc`(EC2Config, EC2Launch v1 및 EC2Launch v2)
- `%WINDIR%\System32\Sysprep\Panther`(EC2Config, EC2Launch v1 및 EC2Launch v2)

- `C:\Program Files\Amazon\Ec2ConfigService\Logs\Ec2ConfigLog.txt`(EC2Config만 해당)
- `C:\ProgramData\Amazon\Ec2Config\Logs`(EC2Config만 해당)
- `C:\ProgramData\Amazon\EC2-Windows\Launch\Log\EC2Launch.log`(EC2Launch v1만)
- `%ProgramData%\Amazon\EC2Launch\log\agent.log`(EC2Launch v2만)

Sysprep으로 이미지를 준비하는 과정 중에 오류 메시지를 받는 경우에는 OS에 접근하지 못할 수도 있습니다. 로그 파일을 검토하려면 인스턴스를 중단하고 그 인스턴스의 루트 볼륨을 부 볼륨인 다른 정상적인 인스턴스에 연결한 다음, 부 볼륨에서 앞서 언급한 로그를 검토해야 합니다. 이름별 로그 파일의 용도에 대한 자세한 내용은 Microsoft 설명서의 Windows 설정 관련 로그 파일을 참조하세요.

Unattendgc 로그 파일에서 오류의 위치를 찾는 경우에는 Microsoft Error Lookup Tool을 이용해 오류에 대한 세부 정보를 얻으세요. Unattendgc 로그 파일에 보고되는 다음 문제는 인스턴스에서 1개 이상의 사용자 프로파일이 오염됨으로써 나타나는 전형적인 결과입니다.

```
Error [Shell Unattend] _FindLatestProfile failed (0x80070003) [gle=0x00000003]
Error [Shell Unattend] CopyProfile failed (0x80070003) [gle=0x00000003]
```

이 문제를 해결하기 위한 두 가지 옵션이 있습니다.

옵션 1

인스턴스에서 Regedit을 이용해 다음 키를 검색합니다. 삭제된 사용자에 대해 프로파일 레지스트리 키가 없는지 확인하십시오.

```
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion
\ProfileList\
```

옵션 2

1. 관련 파일을 다음과 같이 편집합니다.
   - Windows Server 2012 R2 및 이전 버전 - EC2Config 응답 파일(`C:\Program Files\Amazon\Ec2ConfigService\sysprep2008.xml`)을 편집합니다.
   - Windows Server 2016 및 2019 – unattend.xml 응답 파일(`C:\ProgramData\Amazon\EC2-Windows\Launch\Sysprep\Unattend.xml`)을 편집합니다.
   - Windows Server 2022 – unattend.xml 응답 파일(`C:\ProgramData\Amazon\EC2Launch\sysprep\unattend.xml`)을 편집합니다.

2. <CopyProfile>true</CopyProfile>를 <CopyProfile>false</CopyProfile>로 변경합니다.
3. Sysprep을 다시 실행합니다. Sysprep이 완료된 후에는 이러한 구성 변경이 내장된 관리자 사용자 프로파일을 삭제한다는 것에 유의하세요.

EC2Rescue로 손상된 Amazon EC2 Linux 인스턴스 문제 해결

Linux용 EC2Rescue는 Amazon EC2 Linux 인스턴스에서 실행하여 100개 이상의 모듈 라이브러리로 일반적인 문제를 진단, 문제 해결 및 수정할 수 있는 사용하기 쉬운 오픈 소스 도구입니다. 모듈은 BASH 또는 Python 스크립트와 필요한 메타데이터를 포함하는 YAML 파일입니다.

Linux용 EC2Rescue 인스턴스의 몇 가지 일반적인 사용 사례는 다음과 같습니다.

- syslog 및 패키지 관리자 로그 수집
- 리소스 사용률 데이터 수집
- 알려진 문제가 있는 커널 파라미터 및 일반적인 OpenSSH 문제 진단 및 해결

> **Note**
>
> AWSSupport-TroubleshootSSH AWS Systems Manager Automation 런북은 Linux용 EC2Rescue를 설치한 다음 이 도구를 사용하여 Linux 인스턴스에 대한 SSH 연결을 방해하는 일반적인 문제를 확인하거나 수정하려고 합니다. 자세한 내용은 AWSSupport-TroubleshootSSH를 참조하세요.

Windows 인스턴스를 사용하는 경우 the section called “Windows 인스턴스용 EC2Rescue” 섹션을 참조하세요.

주제

- Amazon EC2 Linux 인스턴스에 EC2Rescue 설치
- Amazon EC2 Linux 인스턴스에서 EC2Rescue 명령 실행
- Amazon EC2 Linux 인스턴스용 EC2Rescue 모듈 개발

Amazon EC2 Linux 인스턴스에 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+
  - SUSE Linux Enterprise Server 12+
  - RHEL 7+
  - Ubuntu 16.04+
- 소프트웨어 요구 사항:
  - Python 2.7.9+ 또는 3.2+

EC2Rescue 설치

`AWSSupport-TroubleshootSSH` Runbook은 Linux용 EC2Rescue를 설치한 다음 이 도구를 사용하여 SSH를 통한 Linux 시스템에 대한 원격 연결을 방해하는 일반적인 문제를 확인하거나 수정하려고 합니다. 자세한 내용을 보고 이 자동화를 실행하려면 지원-TroubleshootSSH를 참조하세요.

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) Linux용 EC2Rescue 설치 파일의 서명을 확인합니다. 자세한 내용은 (선택 사항) Linux용 EC2Rescue의 서명 확인 섹션을 참조하세요.
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 -xzvf 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 참조하세요.

(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 섹션의 절차를 수행한 후에 Linux용 EC2Rescue용 소프트웨어가 변경되거나 손상된 것을 발견한 경우 설치 파일을 실행하지 마세요. 대신 Amazon Web Services에 문의하세요.

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 https://www.gnupg.org/를 참조하세요.

첫 번째 단계는 소프트웨어 게시자와 신뢰를 구축하는 것입니다. 소프트웨어 게시자의 퍼블릭 키를 다운로드하고, 퍼블릭 키의 소유자가 정당한 소유자인지 확인한 다음, 퍼블릭 키를 키링에 추가합니다. 키링은 알려진 퍼블릭 키의 모음입니다. 퍼블릭 키의 신뢰성을 설정한 후, 이를 사용하여 애플리케이션의 서명을 확인할 수 있습니다.

업무

- 퍼블릭 키 인증 및 가져오기

- 패키지의 서명 확인

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)
   ```

   > **Tip**
   >
   > 명령을 찾을 수 없다는 오류가 표시되면 `apt-get install gnupg2`(Debian 기반 Linux) 또는 `yum install gnupg2`(Red Hat 기반 Linux)를 사용하여 GnuPG 유틸리티를 설치합니다.

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 마세요.

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 https://www.gnupg.org/를 참조하세요.

Amazon EC2 Linux 인스턴스에서 EC2Rescue 명령 실행

EC2Rescue는 명령줄 도구입니다. Linux 인스턴스에 EC2Rescue를 설치한 후 `./ec2rl help`를 실행하여 도구 사용 방법에 대한 일반적인 도움말을 볼 수 있습니다. `./ec2rl list`를 실행하여 사용 가능한 모듈을 보고, `./ec2rl help` *module_name*을 실행하여 특정 모듈에 대한 도움말을 볼 수 있습니다.

다음은 이 도구를 사용하여 시작할 때 수행할 수 있는 일반적인 작업입니다.

업무

- EC2Rescue 모듈 실행
- EC2Rescue 모듈 결과 업로드
- Amazon EC2 Linux 인스턴스의 백업 생성

EC2Rescue 모듈 실행

모든 EC2Rescue 모듈을 실행하려면 다음을 수행하세요.

추가 파라미터를 지정하지 않고 ./ec2rl run 명령을 사용합니다. 일부 모듈에는 루트 액세스가 필요합니다. 루트 사용자가 아니라면 명령을 실행할 때 sudo를 사용합니다.

```
./ec2rl run
```

특정 EC2Rescue 모듈을 실행하려면 다음을 수행하세요.

./ec2rl run 명령을 사용하고 `--only-modules`에 대해 실행할 모듈의 이름을 지정합니다. 일부 모듈은 사용하려면 인수가 필요합니다.

```
./ec2rl run --only-modules=module_name --arguments
```

예를 들어, dig 모듈을 실행하여 `amazon.com` 도메인을 쿼리하려면 다음 명령을 사용합니다.

```
./ec2rl run --only-modules=dig --domain=amazon.com
```

EC2Rescue 모듈의 결과를 보려면 다음을 수행하세요.

모듈을 실행하고 `cat /var/tmp/ec2rl/logfile_location`에서 로그 파일을 봅니다. 예를 들어, dig 모듈의 로그 파일은 다음 위치에서 찾을 수 있습니다.

```
cat /var/tmp/ec2rl/timestamp/mod_out/run/dig.log
```

EC2Rescue 모듈 결과 업로드

지원에서 EC2Rescue 모듈에 대한 결과를 요청한 경우 EC2Rescue 도구를 사용하여 로그 파일을 업로드할 수 있습니다. 지원에서 제공하는 위치나 사용자가 소유한 Amazon S3 버킷에 결과를 업로드할 수 있습니다.

지원에서 제공하는 위치에 결과를 업로드하려면 다음을 수행하세요.

./ec2rl upload 명령을 사용합니다. `--upload-directory`에 로그 파일의 위치를 지정합니다. `--support-url`에 지원에서 제공하는 URL을 지정합니다.

```
./ec2rl upload --upload-directory=/var/tmp/ec2rl/logfile_location --support-
url="url_provided_by_aws_support"
```

Amazon S3 버킷에 결과를 업로드하려면 다음을 수행하세요.

./ec2rl upload 명령을 사용합니다. `--upload-directory`에 로그 파일의 위치를 지정합니다. `--presigned-url`에 S3 버킷의 미리 서명된 URL을 지정합니다. Amazon S3에 대해 미리 서명된 URL 생성에 대한 자세한 내용은 미리 서명된 URL을 사용하여 객체 업로드를 참조하세요.

```
./ec2rl upload --upload-directory=/var/tmp/ec2rl/logfile_location --presigned-
url="presigned_s3_url"
```

Amazon EC2 Linux 인스턴스의 백업 생성

EC2Rescue를 사용하여 AMI를 생성하거나 연결된 볼륨의 스냅샷을 생성하여 Linux 인스턴스를 백업할 수 있습니다.

API 생성

`./ec2rl run` 명령을 사용하고 `--backup`에 `ami`를 지정합니다.

```
./ec2rl run --backup=ami
```

연결된 모든 볼륨의 다중 볼륨 스냅샷을 생성하려면 다음을 수행하세요.

`./ec2rl run` 명령을 사용하고 `--backup`에 `allvolumes`를 지정합니다.

```
./ec2rl run --backup=allvolumes
```

연결된 특정 볼륨의 스냅샷을 생성하려면 다음을 수행하세요.

`./ec2rl run` 명령을 사용하고 `--backup`에 백업할 볼륨의 ID를 지정합니다.

```
./ec2rl run --backup=vol-01234567890abcdef
```

Amazon EC2 Linux 인스턴스용 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.
제목	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 포함하세요.
	
	예:
	
	`helptext: !!str |`
	`  Collect output from ps for system analysis`

속성	설명
	Consumes --times= for number of times to repeat Consumes --period= for time period between repetition
배치	모듈을 실행해야 하는 단계입니다. 지원되는 값: • prediagnostic • run • postdiagnostic
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.

속성	설명
도메인	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.

속성	설명
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.
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.py의 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. 예시: • xen_netfront • ixgbevf • ena
EC2RL_SUDO	Linux용 EC2Rescue가 루트로 실행되면 true이고, 그렇지 않으면 false입니다.
EC2RL_VIRT_TYPE	인스턴스 메타데이터에 의해 제공된 가상화 유형입니다. 예시: • default-hvm

환경 변수	설명
	• 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
```

EC2Rescue를 사용하여 손상된 Amazon EC2 Windows 인스턴스 문제 해결

EC2Rescue for Windows Server는 가능한 문제를 진단하고 해결하기 위해 Amazon EC2 Windows Server 인스턴스에서 실행하는 편리한 도구입니다. 이 도구는 로그 파일을 수집하고 문제를 해결하며, 문제가 발생할 가능성이 있는 영역을 사전에 찾아내는 데 매우 유용하게 사용할 수 있습니다. 또한 다

른 인스턴스의 Amazon EBS 루트 볼륨을 검사하고 해당 볼륨을 사용하여 Windows Server 인스턴스 문제를 해결하기 위한 관련 로그를 수집할 수 있습니다. 다음은 EC2Rescue에서 해결할 수 있는 몇 가지 일반적인 문제입니다.

- 방화벽, RDP(원격 데스크톱 프로토콜) 또는 네트워크 인터페이스 구성으로 인한 인스턴스 연결 문제
- 중단 오류, 부팅 루프 또는 손상된 레지스트리로 인한 운영 체제 부팅 문제
- 고급 로그 분석 및 문제 해결이 필요할 수 있는 문제

EC2Rescue for Windows Server에는 두 가지 모듈이 있습니다.

- 모든 다양한 소스에서 데이터를 수집하는 데이터 수집기 모듈
- 사전 정의된 여러 규칙에 대해 수집된 데이터의 구문을 분석하여 문제를 식별하고 제안 사항을 제공하는 분석기 모듈

EC2Rescue for Windows Server 도구는 Windows Server 2012 이상 버전을 실행하는 Amazon EC2 인스턴스에서만 실행됩니다. 이 도구는 시작되면 Amazon EC2 인스턴스에서 실행 중인지 여부를 확인합니다.

> **Note**
>
> `AWSSupport-ExecuteEC2Rescue` AWS Systems Manager Automation 런북은 EC2Rescue 도구를 사용하여 문제를 해결하고 가능한 경우 지정된 EC2 인스턴스와 관련된 일반적인 연결 문제를 해결합니다. 자세한 내용을 보고 이 자동화를 실행하려면 >AWSSupport-ExecuteEC2Rescue를 참조하세요.

Linux 인스턴스를 사용하는 경우 the section called “Linux용 EC2Rescue 인스턴스” 섹션을 참조하세요.

주제

- EC2Rescue GUI로 손상된 Windows 인스턴스 문제 해결
- EC2Rescue CLI로 손상된 Windows 인스턴스 문제 해결
- EC2Rescue 및 Systems Manager로 손상된 Windows 인스턴스 문제 해결

EC2Rescue GUI로 손상된 Windows 인스턴스 문제 해결

EC2Rescue for Windows Server는 오프라인 인스턴스에서 다음 분석을 수행할 수 있습니다.

옵션	설명

Diagnose and Rescue	EC2Rescue for Windows Server는 다음 서비스 설정을 확인하고 해당 문제를 해결할 수 있습니다.
	• 시스템 시간
	  • RealTimeisUniversal - `RealTimeisUniversal` 레지스트리의 사용 여부를 감지합니다. 비활성화되었을 경우 시간대가 UTC 외의 값으로 설정되면 Windows 시스템 시간에 오차가 발생합니다.
	• Windows 방화벽
	  • 도메인 네트워크 - 이 Windows 방화벽 프로파일의 사용 여부를 감지합니다.
	  • 프라이빗 네트워크 - 이 Windows 방화벽 프로파일의 사용 여부를 감지합니다.
	  • 게스트 또는 퍼블릭 네트워크 - 이 Windows 방화벽 프로파일의 사용 여부를 감지합니다.
	• 원격 데스크톱
	  • 서비스 시작 - 원격 데스크톱 서비스의 사용 여부를 감지합니다.
	  • 원격 데스크톱 연결 - 사용 여부를 감지합니다.
	  • TCP 포트 - 원격 데스크톱 서비스에서 데이터를 수신하는 포트를 확인합니다.

옵션	설명

	• EC2Config(Windows Server 2012 R2 및 이전 버전)
	  • 설치 - 설치되어 있는 EC2Config 버전을 확인합니다.
	  • 서비스 시작 - EC2Config 서비스의 사용 여부를 감지합니다.
	  • Ec2SetPassword - 새로운 관리자 암호를 생성합니다.
	  • Ec2HandleUserData - 인스턴스의 다음 부팅 시 사용자 데이터 스크립트를 실행할 수 있습니다.
	• EC2Launch(Windows Server 2016 이상)
	  • 설치 - 설치되어 있는 EC2Launch 버전을 확인합니다.
	  • Ec2SetPassword - 새로운 관리자 암호를 생성합니다.
	• 네트워크 인터페이스
	  • DHCP 서비스 시작 - DHCP 서비스의 사용 여부를 감지합니다.
	  • 이더넷 세부 정보 - 네트워크 드라이버 버전에 대한 정보를 표시합니다(확인될 경우).
	  • 이너넷 기반 DHCP - DHCP의 사용 여부를 감지합니다.
	• 디스크 서명 상태
	  • 디스크 서명(Signature on disk) 및 부팅 구성 데이터베이스 서명(BCD)) - 디스크 서명과 BCD 서명이 동일한지 여부를 감지합니다. 값이 다른 경우 EC2Rescue는 디스크 서명을 BCD의 서명으로 덮어쓰려고 시도합니다.

옵션	설명
복원	다음 조치 중 하나를 취하세요. • 마지막으로 성공한 구성 - 인스턴스를 마지막으로 확인된 부팅 가능 상태로 부팅하려고 시도합니다. • 백업에서 레지스트리 복원 - `\Windows\System32\config\RegBack` 의 레지스트리를 복원합니다.
Capture Logs	분석을 위해 인스턴스에 대한 로그를 캡처할 수 있습니다.

EC2Rescue for Windows Server는 활성 인스턴스 및 오프라인 인스턴스에서 다음 데이터를 수집할 수 있습니다.

Item	설명
Event Log(이벤트 로그)	애플리케이션, 시스템, EC2Config 이벤트 로그를 수집합니다.
레지스트리	`SYSTEM` 및 `SOFTWARE` Hive를 수집합니다.
Windows Update Log	Windows 업데이트에서 생성된 로그 파일을 수집합니다. **Note** Windows Server 2016 이상 버전의 경우 로그는 Windows용 이벤트 추적(ETW) 형식으로 수집됩니다.
Sysprep Log	Windows System Preparation 도구가 생성하는 로그 파일을 수집합니다.

Item	설명
드라이버 설정 로그	Windows SetupAPI 로그(`setupapi.dev.log` 및 `setupapi.setup.log`)를 수집합니다.
Boot Configuration	`HKEY_LOCAL_MACHINE\BCD00000000` Hive를 수집합니다.
Memory Dump	인스턴스에 있는 메모리 덤프 파일을 수집합니다.
EC2Config File	EC2Config 서비스에서 생성되는 로그 파일을 수집합니다.
EC2Launch File	EC2Launch 스크립트가 생성하는 로그 파일을 수집합니다.
SSM Agent File	SSM Agent 및 패치 관리자 로그에 의해 생성된 로그 파일을 수집합니다.
EC2 ElasticGPU 파일	엘라스틱 GPU와 관련된 이벤트 로그를 수집합니다.
ECS	Amazon ECS와 관련된 로그를 수집합니다.
CloudEndure	CloudEndure 에이전트와 관련된 로그 파일을 수집합니다.
MGN 또는 DRS 로그 파일용 AWS Replication Agent	AWS Application Migration Service 또는 AWS Elastic Disaster Recovery와 관련된 로그 파일을 수집합니다.

EC2Rescue for Windows Server는 활성 인스턴스에서 다음과 같은 추가 데이터를 수집할 수 있습니다.

Item	설명
System Information	MSInfo32를 수집합니다.
그룹 정책 결과	그룹 정책 보고서를 수집합니다.

오프라인 인스턴스 분석

오프라인 인스턴스 옵션은 Windows 인스턴스와 관련된 부팅 문제를 해결할 때 유용합니다.

오프라인 인스턴스에서 작업을 수행하려면

1. 작동하는 Windows Server 인스턴스에서 EC2Rescue for Windows Server 도구를 다운로드하여 파일의 압축을 풉니다.

   다음의 PowerShell 명령을 실행하여 Internet Explorer ESC(보안 강화 구성)의 변경 없이 EC2Rescue를 다운로드할 수 있습니다.

   ```
   Invoke-WebRequest https://s3.amazonaws.com/ec2rescue/windows/EC2Rescue_latest.zip -
   OutFile $env:USERPROFILE\Desktop\EC2Rescue_latest.zip
   ```

   이 명령을 사용하면 현재 로그인한 사용자의 데스크톱으로 .zip 형식의 EC2Rescue 파일이 다운로드됩니다.

   > **Note**
   >
   > Windows Server 2016 또는 이전 버전을 사용 중이고 파일을 다운로드할 때 오류가 발생하는 경우 PowerShell 터미널에서 TLS 1.2를 활성화해야 할 수 있습니다. 다음 명령을 사용하여 현재 PowerShell 세션에 대해 TLS 1.2를 활성화한 다음 다시 시도해보세요.
   >
   > ```
   > [Net.ServicePointManager]::SecurityProtocol =
   >  [Net.SecurityProtocolType]::Tls12
   > ```

2. 문제가 있는 인스턴스가 아직 중지되지 않은 경우 해당 인스턴스를 중지합니다.
3. 오류가 발생한 인스턴스에서 EBS 루트 볼륨을 분리한 후 이 볼륨을 EC2Rescue for Windows Server가 설치되고 작동하는 Windows 인스턴스에 연결합니다.

4. 작동하는 인스턴스에서 EC2Rescue for Windows Server 도구를 실행하고 오프라인 인스턴스를 선택합니다.
5. 새로 탑재한 디스크를 선택한 후 다음을 선택합니다.
6. 디스크 선택을 확인한 후 예를 선택합니다.
7. 수행할 오프라인 인스턴스 옵션을 선택한 후 다음을 선택합니다.

EC2Rescue for Windows Server 도구가 볼륨을 스캔한 후, 선택된 로그 파일을 바탕으로 문제 해결 정보를 수집합니다.

활성 인스턴스에서 데이터 수집

활성 인스턴스에서 로그와 기타 데이터를 수집할 수 있습니다.

활성 인스턴스에서 데이터를 수집하려면

1. Windows 인스턴스에 연결합니다.
2. EC2Rescue for Windows Server 도구를 Windows 인스턴스에 다운로드한 후 파일의 압축을 풉니다.

   다음의 PowerShell 명령을 실행하여 Internet Explorer ESC(보안 강화 구성)의 변경 없이 EC2Rescue를 다운로드할 수 있습니다.

   ```
   Invoke-WebRequest https://s3.amazonaws.com/ec2rescue/windows/EC2Rescue_latest.zip -
   OutFile $env:USERPROFILE\Desktop\EC2Rescue_latest.zip
   ```

   이 명령을 사용하면 현재 로그인한 사용자의 데스크톱으로 .zip 형식의 EC2Rescue 파일이 다운로드됩니다.

   > **Note**
   >
   > Windows Server 2016 또는 이전 버전을 사용 중이고 파일을 다운로드할 때 오류가 발생하는 경우 PowerShell 터미널에서 TLS 1.2를 활성화해야 할 수 있습니다. 다음 명령을 사용하여 현재 PowerShell 세션에 대해 TLS 1.2를 활성화한 다음 다시 시도해보세요.
   >
   > ```
   > [Net.ServicePointManager]::SecurityProtocol =
   >  [Net.SecurityProtocolType]::Tls12
   > ```

3. EC2Rescue for Windows Server 애플리케이션을 열고 라이선스 계약에 동의합니다.
4. 다음, 현재 인스턴스, 로그 캡처(Capture logs)를 차례로 선택합니다.
5. 수집할 데이터 항목을 선택하고 수집...(Collect...)을 선택합니다. 경고를 읽고 예를 선택하여 계속 진행합니다.
6. ZIP 파일의 파일 이름과 위치를 선택한 다음 저장을 선택합니다.
7. EC2Rescue for Windows Server에서 작업을 완료하면 포함된 폴더 열기(Open Containing Folder)를 선택하여 ZIP 파일을 봅니다.
8. 마침을 클릭합니다.

EC2Rescue CLI로 손상된 Windows 인스턴스 문제 해결

EC2Rescue for Windows Server 명령줄 인터페이스(CLI)를 사용해 프로그래밍 방식으로 EC2Rescue for Windows Server 플러그인("작업"이라고 부름)을 실행할 수 있습니다.

EC2Rescue for Windows Server 도구에는 두 가지 실행 모드가 있습니다.

- /online — 이 모드에서는 EC2Rescue for Windows Server가 설치된 인스턴스에 대해 로그 파일 수집과 같은 작업을 수행할 수 있습니다.
- /offline:<device_id> — 이 모드에서는 EC2Rescue for Windows Server가 설치된 인스턴스에서 별도의 Amazon EC2 Windows 인스턴스에 연결된 오프라인 루트 볼륨에 대해 작업을 수행할 수 있습니다.

EC2Rescue for Windows Server 도구를 Windows 인스턴스에 다운로드한 후 파일의 압축을 풉니다. 다음 명령을 사용하여 도움말 파일을 볼 수 있습니다.

```
EC2RescueCmd.exe /help
```

EC2Rescue for Windows Server는 Amazon EC2 Windows 인스턴스에 대해 다음 작업을 수행할 수 있습니다.

- 수집 작업
- 복구 작업
- 복원 작업

수집 작업

> **Note**
>
> 모든 로그, 전체 로그 그룹 또는 특정 그룹 내 개별 로그를 수집할 수 있습니다.

EC2Rescue for Windows Server은(는) 활성 인스턴스 및 오프라인 인스턴스에서 다음 데이터를 수집할 수 있습니다.

로그 그룹	사용 가능한 로그	설명
`all`		모든 사용 가능한 로그를 수집합니다.
`eventlog`	• `'Application'` • `'System'` • `'EC2ConfigService'`	애플리케이션, 시스템, EC2Config 이벤트 로그를 수집합니다.
`memory-dump`	• `'Memory Dump File'` • `'Mini Dump Files'`	인스턴스에 있는 메모리 덤프 파일을 수집합니다.
`ec2config`	• `'Log Files'` • `'Configuration Files'`	EC2Config 서비스에서 생성되는 로그 파일을 수집합니다.
`ec2launch`	• `'Logs'` • `'Config'`	EC2Launch 스크립트가 생성하는 로그 파일을 수집합니다.
`ssm-agent`	• `'Log Files'` • `'Patch Baseline Logs'` • `'InstanceData'`	SSM Agent 및 패치 관리자 로그에 의해 생성된 로그 파일을 수집합니다.
`sysprep`	`'Log Files'`	Windows System Preparation 도구가 생성하는 로그 파일을 수집합니다.

로그 그룹	사용 가능한 로그	설명
driver-setup	• 'SetupAPI Log Files' • 'DPInst Log File' • 'AWS PV Setup Log File'	Windows SetupAPI 로그(setupapi.dev.log 및 setupapi.setup.log)를 수집합니다.
registry	• 'SYSTEM' • 'SOFTWARE' • 'BCD'	SYSTEM 및 SOFTWARE Hive를 수집합니다.
egpu	• 'Event Log' • 'System Files'	엘라스틱 GPU와 관련된 이벤트 로그를 수집합니다.
boot-config	'BCDEDIT Output'	HKEY_LOCAL_MACHINE\BCD00000000 Hive를 수집합니다.
windows-update	'Log Files'	Windows 업데이트에서 생성된 로그 파일을 수집합니다. **Note** Windows Server 2016 이상 버전의 경우 로그는 Windows용 이벤트 추적(ETW) 형식으로 수집됩니다.
cloudendure	• 'Migrate Script Logs' • 'Driver Logs' • 'CloudEndure File List'	CloudEndure 에이전트와 관련된 로그 파일을 수집합니다.

EC2Rescue for Windows Server은(는) 활성 인스턴스에서 다음과 같은 추가 데이터를 수집할 수 있습니다.

로그 그룹	사용 가능한 로그	설명
system-info	'MSInfo32 Output'	MSInfo32를 수집합니다.
gpresult	'GPResult Output'	그룹 정책 보고서를 수집합니다.

다음과 같은 옵션을 사용할 수 있습니다.

- /output:<outputFilePath> - 수집된 로그 파일을 zip 형식으로 저장하는 데 필요한 대상 파일 경로 위치입니다.
- /no-offline - 오프라인 모드에서 사용되는 선택적 속성입니다. 작업 완료 후 볼륨을 오프라인으로 설정하지 않습니다.
- /no-fix-signature - 오프라인 모드에서 사용되는 선택적 속성입니다. 작업 완료 후 발생할 수 있는 디스크 서명 충돌을 해결하지 않습니다.

예시

다음은 EC2Rescue for Windows Server CLI를 사용하는 예제입니다.

온라인 모드 예제

모든 사용 가능한 로그를 수집합니다.

```
EC2RescueCmd /accepteula /online /collect:all /output:<outputFilePath>
```

특정 로그 그룹만 수집합니다.

```
EC2RescueCmd /accepteula /online /collect:ec2config /output:<outputFilePath>
```

로그 그룹 내 개별 로그를 수집합니다.

```
EC2RescueCmd /accepteula /online /collect:'ec2config.Log Files,driver-setup.SetupAPI
 Log Files' /output:<outputFilePath>
```

오프라인 모드 예제

EBS 볼륨에서 모든 사용 가능한 로그를 수집합니다. 볼륨은 device_id value로 지정합니다.

```
EC2RescueCmd /accepteula /offline:xvdf /collect:all /output:<outputFilePath>
```

특정 로그 그룹만 수집합니다.

```
EC2RescueCmd /accepteula /offline:xvdf /collect:ec2config /output:<outputFilePath>
```

복구 작업

EC2Rescue for Windows Server는 다음 서비스 설정을 확인하고 해당 문제를 해결할 수 있습니다.

서비스 그룹	사용 가능한 작업	설명
`all`		
`system-time`	`'RealTimeIsUniversal'`	시스템 시간 • RealTimeisUniversal - `RealTimeisUniversal` 레지스트리의 사용 여부를 감지합니다. 비활성화되었을 경우 시간대가 UTC 외의 값으로 설정되면 Windows 시스템 시간에 오차가 발생합니다.
`firewall`	• `'Domain networks'` • `'Private networks'` • `'Guest or public networks'`	Windows 방화벽 • 도메인 네트워크 - 이 Windows 방화벽 프로파일의 사용 여부를 감지합니다. • 프라이빗 네트워크 - 이 Windows 방화벽 프로파일의 사용 여부를 감지합니다. • 게스트 또는 퍼블릭 네트워크 - 이 Windows 방화벽 프

서비스 그룹	사용 가능한 작업	설명
		로파일의 사용 여부를 감지합니다.
rdp	• 'Service Start' • 'Remote Desktop Connections' • 'TCP Port'	원격 데스크톱 • 서비스 시작 - 원격 데스크톱 서비스의 사용 여부를 감지합니다. • 원격 데스크톱 연결 - 사용 여부를 감지합니다. • TCP 포트 - 원격 데스크톱 서비스에서 데이터를 수신하는 포트를 확인합니다.
ec2config	• 'Service Start' • 'Ec2SetPassword' • 'Ec2HandleUserData'	EC2Config • 서비스 시작 - EC2Config 서비스의 사용 여부를 감지합니다. • Ec2SetPassword - 새로운 관리자 암호를 생성합니다. • Ec2HandleUserData - 인스턴스의 다음 부팅 시 사용자 데이터 스크립트를 실행할 수 있습니다.
ec2launch	'Reset Administrator Password'	새 Windows 관리자 암호를 생성합니다.
network	'DHCP Service Startup'	네트워크 인터페이스 • DHCP 서비스 시작 - DHCP 서비스의 사용 여부를 감지합니다.

다음과 같은 옵션을 사용할 수 있습니다.

- /level:<level> - 작업이 트리거해야 하는 확인 수준에 대한 선택적 속성입니다. 허용되는 값은 `information`, `warning`, `error`, `all`입니다. 기본적으로 `error`로 설정됩니다.
- /check-only - 보고서를 생성하지만 오프라인 볼륨은 수정하지 않는 선택적 속성입니다.

> **Note**
>
> EC2Rescue for Windows Server에서 잠재적 디스크 서명 충돌을 감지하면 `/check-only` 옵션을 사용하는 경우에도 기본적으로 오프라인 프로세스가 완료된 후 서명을 수정합니다. 수정을 방지하려면 `/no-fix-signature` 옵션을 사용해야 합니다.

- /no-offline - 작업 완료 후 볼륨을 오프라인으로 설정하는 것을 금지하는 선택적 속성입니다.
- /no-fix-signature - 작업 완료 후 발생 가능한 디스크 서명 충돌을 해결하지 않는 선택적 옵션입니다.

복구 예제

다음은 EC2Rescue for Windows Server CLI를 사용하는 예제입니다. 볼륨은 device_id value를 사용하여 지정합니다.

볼륨에서 식별된 문제를 모두 해결하려고 시도합니다.

```
EC2RescueCmd /accepteula /offline:xvdf /rescue:all
```

볼륨에서 특정 서비스 그룹의 모든 문제를 해결하려고 시도합니다.

```
EC2RescueCmd /accepteula /offline:xvdf /rescue:firewall
```

볼륨에서 특정 서비스 그룹의 특정 항목을 해결하려고 시도합니다.

```
EC2RescueCmd /accepteula /offline:xvdf /rescue:rdp.'Service Start'
```

볼륨에서 해결을 시도할 여러 문제를 지정합니다.

```
EC2RescueCmd /accepteula /offline:xvdf /rescue:'system-
time.RealTimeIsUniversal,ec2config.Service Start'
```

복원 작업

EC2Rescue for Windows Server는 다음 서비스 설정을 확인하고 해당 문제를 해결할 수 있습니다.

서비스 그룹	사용 가능한 작업	설명
마지막으로 확인된 양호한 구성을 복원	`lkgc`	마지막으로 성공한 구성 - 인스턴스를 마지막으로 확인된 부팅 가능 상태로 부팅하려고 시도합니다.
최신 백업으로부터 Windows 레지스트리를 복원	`regback`	백업에서 레지스트리 복원 - `\Windows\System32\config\RegBack` 의 레지스트리를 복원합니다.

다음과 같은 옵션을 사용할 수 있습니다.

- /no-offline — 작업 완료 후 볼륨을 오프라인으로 설정하는 것을 금지하는 선택적 속성입니다.
- /no-fix-signature — 작업 완료 후 발생 가능한 디스크 서명 충돌을 해결하지 않는 선택적 옵션입니다.

복원 예제

다음은 EC2Rescue for Windows Server CLI를 사용하는 예제입니다. 볼륨은 device_id value를 사용하여 지정합니다.

볼륨에서 마지막으로 확인된 양호한 구성을 복원합니다.

```
EC2RescueCmd /accepteula /offline:xvdf /restore:lkgc
```

볼륨에서 마지막 Windows 레지스트리 백업을 복원합니다.

```
EC2RescueCmd /accepteula /offline:xvdf /restore:regback
```

EC2Rescue 및 Systems Manager로 손상된 Windows 인스턴스 문제 해결

지원는 Systems Manager 사용 인스턴스에서 EC2Rescue for Windows Server를 실행할 때 인터페이스로 사용할 수 있는 Systems Manager Run Command 문서를 제공합니다. 이 Run Command 문서를 `AWSSupport-RunEC2RescueForWindowsTool`이라고 합니다.

이 Systems Manager Run Command 문서는 다음의 작업을 수행합니다.

- EC2Rescue for Windows Server를 다운로드하고 확인합니다.
- 간편하게 도구와 상호 작용하도록 PowerShell 모듈을 가져옵니다.
- 제공된 명령 및 파라미터를 사용하여 EC2RescueCmd를 실행합니다.

Systems Manager Run Command 문서는 세 가지 파라미터를 수락합니다.

- 명령—EC2Rescue for Windows Server 작업. 현재 허용되는 값은 다음과 같습니다.
  - ResetAccess — 로컬 관리자 암호를 재설정합니다. 현재 인스턴스의 로컬 관리자 암호가 재설정되고 임의로 생성된 암호가 Parameter Store에 `/EC2Rescue/Password/<INSTANCE_ID>`로 안전하게 저장됩니다. 이 작업을 선택하고 파라미터를 제공하지 않으면 기본 KMS 키을(를) 사용하여 암호가 자동으로 암호화됩니다. 선택 사항으로, 파라미터에서 KMS 키 ID를 지정하여 고유의 키로 암호를 암호화할 수 있습니다.
  - CollectLogs — `/collect:all` 작업으로 EC2Rescue for Windows Server를 실행합니다. 이 작업을 선택할 경우 로그를 업로드할 Amazon S3 버킷 이름이 `Parameters`에 포함되어 있어야 합니다.
  - FixAll — `/rescue:all` 작업으로 EC2Rescue for Windows Server를 실행합니다. 이 작업을 선택할 경우 `Parameters`에 복구할 블록 디바이스 이름이 포함되어야 합니다.
- 파라미터 — 지정된 명령에 대해 전달할 PowerShell 파라미터입니다.

요구 사항

ResetAccess 작업을 실행하려면 Amazon EC2 인스턴스에 암호화된 암호를 Parameter Store에 쓸 수 있는 권한을 부여하는 정책이 연결되어 있어야 합니다. 이 정책을 관련 IAM 역할에 연결하고 나서 몇 분 기다렸다가 인스턴스의 암호를 재설정하세요.

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
    {
      "Effect": "Allow",
      "Action": [
        "ssm:PutParameter"
      ],
      "Resource": [
        "arn:aws:ssm:region:account_id:parameter/EC2Rescue/Passwords/<instanceid>"
      ]
    }
  ]
}
```

기본 KMS 키가 아닌 사용자 지정 KMS 키를 사용하는 경우 이 정책을 대신 사용합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:PutParameter"
      ],
      "Resource": [
        "arn:aws:ssm:region:account_id:parameter/EC2Rescue/Passwords/<instanceid>"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:Encrypt"
      ],
      "Resource": [
        "arn:aws:kms:region:account_id:key/<kmskeyid>"
      ]
    }
  ]
}
```

문서의 JSON 보기

다음 절차는 이 문서의 JSON을 보는 방법을 설명합니다.

Systems Manager Run Command 문서의 JSON을 보려면

1. AWS Systems Manager 콘솔(https://console.aws.amazon.com/systems-manager/)을 엽니다.
2. 탐색 창에서 변경 관리 도구를 확장하고 문서를 선택합니다.
3. 검색 창에 `AWSSupport-RunEC2RescueForWindowsTool`을 입력한 다음 `AWSSupport-RunEC2RescueForWindowsTool` 문서를 선택합니다.
4. 콘텐츠 탭을 선택합니다.

예시

다음은 Systems Manager Run Command 문서를 사용하여 AWS CLI에서 EC2Rescue for Windows Server를 실행하는 몇 가지 예제입니다. AWS CLI를 사용한 명령 전송에 대한 자세한 내용은 send-command를 참조하세요.

예시

- 오프라인 루트 볼륨에서 식별된 모든 문제 해결 시도
- 현재 Amazon EC2 Windows 인스턴스에서 로그 수집
- 로컬 관리자 암호 재설정

오프라인 루트 볼륨에서 식별된 모든 문제 해결 시도

다음과 같이 Amazon EC2 Windows 인스턴스에 연결된 오프라인 루트 볼륨에서 식별된 문제를 모두 해결하려고 시도합니다.

```
aws ssm send-command --instance-ids "i-0cb2b964d3e14fd9f" --document-name "AWSSupport-
RunEC2RescueForWindowsTool" --parameters "Command=FixAll, Parameters='xvdf'" --output
 text
```

현재 Amazon EC2 Windows 인스턴스에서 로그 수집

다음과 같이 현재 온라인 Amazon EC2 Windows 인스턴스에서 모든 로그를 수집하여 Amazon S3 버킷으로 업로드합니다.

```
aws ssm send-command --instance-ids "i-0cb2b964d3e14fd9f" --document-name "AWSSupport-
RunEC2RescueForWindowsTool" --parameters "Command=CollectLogs, Parameters='amzn-s3-
demo-bucket'" --output text
```

로컬 관리자 암호 재설정

다음 예제에서는 로컬 관리자 암호를 재설정하는 데 사용할 수 있는 방법을 보여 줍니다. 출력에는 Parameter Store에 대한 링크가 제공됩니다.여기에서 임의로 생성된 보안 암호를 찾은 다음 이 암호를 사용하여 RDP를 통해 Amazon EC2 Windows 인스턴스에 로컬 관리자로 연결할 수 있습니다.

기본 AWS KMS key alias/aws/ssm을 사용하여 온라인 인스턴스의 로컬 관리자 암호를 재설정합니다.

```
aws ssm send-command --instance-ids "i-0cb2b964d3e14fd9f" --document-name "AWSSupport-
RunEC2RescueForWindowsTool" --parameters "Command=ResetAccess" --output text
```

KMS 키을(를) 사용하여 온라인 인스턴스의 로컬 관리자 암호를 재설정합니다.

```
aws ssm send-command --instance-ids "i-0cb2b964d3e14fd9f" --document-name "AWSSupport-
RunEC2RescueForWindowsTool" --parameters "Command=ResetAccess, Parameters=a133dc3c-
a2g4-4fc6-a873-6c0720104bf0" --output text
```

> **Note**
>
> 이 예시에서 KMS 키는 a133dc3c-a2g4-4fc6-a873-6c0720104bf0입니다.

인스턴스용 EC2 직렬 콘솔

EC2 직렬 콘솔을 사용하면 Amazon EC2 인스턴스의 직렬 포트에 액세스하고 이 포트를 사용하여 부팅, 네트워크 구성 및 기타 문제를 해결할 수 있습니다. 직렬 콘솔은 인스턴스에서 네트워킹 기능 없이 사용할 수 있습니다. 직렬 콘솔을 사용하면 키보드와 모니터가 인스턴스의 직렬 포트에 직접 연결된 것처럼 인스턴스에 명령을 입력할 수 있습니다. 직렬 콘솔 세션은 인스턴스를 재부팅하고 중지하는 동안 지속됩니다. 재부팅할 때는 처음부터 모든 부팅 메시지를 볼 수 있습니다.

직렬 콘솔에 대한 액세스는 기본적으로 제공되지 않습니다. 조직에서 계정에 직렬 콘솔에 대한 액세스 권한을 부여하고, 사용자에게 직렬 콘솔 액세스 권한을 부여하는 IAM 정책을 구성해야 합니다. 인스턴스 ID, 리소스 태그 및 기타 IAM 레버를 사용하여 세분화된 수준에서 직렬 콘솔 액세스를 제어할 수 있습니다. 자세한 내용은 EC2 직렬 콘솔에 대한 액세스 구성 섹션을 참조하세요.

직렬 콘솔에는 EC2 콘솔 또는 AWS CLI를 사용하여 액세스할 수 있습니다.

직렬 콘솔은 추가 비용 없이 제공됩니다.

주제

- EC2 직렬 콘솔 사전 조건
- EC2 직렬 콘솔에 대한 액세스 구성
- EC2 직렬 콘솔에 연결
- EC2 직렬 콘솔 연결 해제
- EC2 직렬 콘솔을 사용하여 Amazon EC2 인스턴스 문제 해결

EC2 직렬 콘솔 사전 조건

EC2 직렬 콘솔에 연결하고 문제 해결을 위해 선택한 도구를 사용하려면 다음 사전 요구 사항을 충족해야 합니다.

- AWS 리전
- Wavelength 영역 및 AWS Outposts
- 로컬 영역
- 인스턴스 타입
- 액세스 권한 부여
- 브라우저 기반 클라이언트 지원
- 인스턴스 상태
- Amazon EC2 Systems Manager
- 선택한 문제 해결 도구 구성

AWS 리전

아시아 태평양(태국) 및 멕시코(중부)를 제외한 모든 AWS 리전에서 지원됩니다.

Wavelength 영역 및 AWS Outposts

지원하지 않음.

로컬 영역

모든 로컬 영역에서 지원됩니다.

인스턴스 타입

지원되는 인스턴스 유형

- Linux
  - Nitro 시스템에 구축된 모든 가상화된 인스턴스에 대해 지원됩니다.
  - 다음을 제외한 모든 베어 메탈 인스턴스에 대한 인자:
    - 범용: `a1.metal`, `mac1.metal`, `mac2.metal`
    - 액셀러레이티드 컴퓨팅: `g5g.metal`
    - 메모리 최적화: `u-6tb1.metal`, `u-9tb1.metal`, `u-12tb1.metal`, `u-18tb1.metal`, `u-24tb1.metal`
- Windows

  Nitro 시스템에 구축된 모든 가상화된 인스턴스에 대해 지원됩니다. 베어 메탈 인스턴스에서는 지원되지 않습니다.

액세스 권한 부여

EC2 직렬 콘솔에 대한 액세스 권한을 부여하는 구성 작업을 완료해야 합니다. 자세한 내용은 EC2 직렬 콘솔에 대한 액세스 구성 섹션을 참조하세요.

브라우저 기반 클라이언트 지원

브라우저 기반 클라이언트를 사용하여 직렬 콘솔에 연결하려면 브라우저가 WebSocket을 지원해야 합니다. 브라우저가 WebSocket을 지원하지 않는 경우 자체 키 및 SSH 클라이언트를 사용하여 직렬 콘솔에 연결합니다.

인스턴스 상태

`running`여야 합니다.

인스턴스가 `pending`, `stopping`, `stopped`, `shutting-down` 또는 `terminated` 상태인 경우 직렬 콘솔에 연결할 수 없습니다.

인스턴스 상태에 대한 자세한 내용은 Amazon EC2 인스턴스 상태 변경 섹션을 참조하세요.

Amazon EC2 Systems Manager

인스턴스에서 Amazon EC2 Systems Manager를 사용하는 경우 SSM Agent 버전 3.0.854.0 이상을 인스턴스에 설치해야 합니다. SSM Agent에 대한 자세한 내용은 AWS Systems Manager 사용 설명서에서 SSM Agent 작업을 참조하세요.

선택한 문제 해결 도구 구성

직렬 콘솔을 통해 인스턴스 문제를 해결하기 위해 Linux 인스턴스에서는 GRUB 또는 SysRq를, Windows 인스턴스에서는 Special Admin Console(SAC)을 사용할 수 있습니다. 이러한 도구를 사용하려면 먼저 도구를 사용할 모든 인스턴스에서 구성 단계를 수행해야 합니다.

인스턴스의 운영 체제에 대한 지침을 사용하여 선택한 문제 해결 도구를 구성합니다.

(Linux 인스턴스) GRUB 구성

GRUB을 구성하려면 인스턴스를 시작할 때 사용된 AMI에 따라 다음 절차 중 하나를 선택합니다.

Amazon Linux 2

Amazon Linux 2 인스턴스에서 GRUB을 구성하려면

1. SSH를 사용하여 Linux 인스턴스에 연결
2. `/etc/default/grub`에서 다음 옵션을 추가하거나 변경합니다.
   - `GRUB_TIMEOUT=1`을 설정합니다.
   - 추가 `GRUB_TERMINAL="console serial"`.
   - 추가 `GRUB_SERIAL_COMMAND="serial --speed=115200"`.

   다음은 `/etc/default/grub`의 예제입니다. 시스템 설정에 따라 구성을 변경해야 할 수 있습니다.

   ```
   GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,115200n8 net.ifnames=0
    biosdevname=0 nvme_core.io_timeout=4294967295 rd.emergency=poweroff rd.shell=0"
   GRUB_TIMEOUT=1
   GRUB_DISABLE_RECOVERY="true"
   GRUB_TERMINAL="console serial"
   GRUB_SERIAL_COMMAND="serial --speed=115200"
   ```
3. 다음 명령을 실행하여 업데이트된 구성을 적용합니다.

   ```
   [ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
   ```

Ubuntu

Ubuntu 인스턴스에서 GRUB을 구성하려면

1. 인스턴스에 연결합니다.
2. /etc/default/grub.d/50-cloudimg-settings.cfg에서 다음 옵션을 추가하거나 변경합니다.

   - GRUB_TIMEOUT=1을 설정합니다.
   - 추가 GRUB_TIMEOUT_STYLE=menu.
   - 추가 GRUB_TERMINAL="console serial".
   - GRUB_HIDDEN_TIMEOUT를 제거합니다.
   - 추가 GRUB_SERIAL_COMMAND="serial --speed=115200".

   다음은 /etc/default/grub.d/50-cloudimg-settings.cfg의 예제입니다. 시스템 설정에 따라 구성을 변경해야 할 수 있습니다.

```
# Cloud Image specific Grub settings for Generic Cloud Images
# CLOUD_IMG: This file was created/modified by the Cloud Image build process

# Set the recordfail timeout
GRUB_RECORDFAIL_TIMEOUT=0

# Do not wait on grub prompt
GRUB_TIMEOUT=1
GRUB_TIMEOUT_STYLE=menu

# Set the default commandline
GRUB_CMDLINE_LINUX_DEFAULT="console=tty1 console=ttyS0
 nvme_core.io_timeout=4294967295"

# Set the grub console type
GRUB_TERMINAL="console serial"
GRUB_SERIAL_COMMAND="serial --speed 115200"
```

3. 다음 명령을 실행하여 업데이트된 구성을 적용합니다.

```
[ec2-user ~]$ sudo update-grub
```

RHEL

RHEL 인스턴스에서 GRUB을 구성하려면

1. 인스턴스에 연결합니다.
2. /etc/default/grub에서 다음 옵션을 추가하거나 변경합니다.

   - GRUB_TERMINAL_OUTPUT를 제거합니다.
   - 추가 GRUB_TERMINAL="console serial".
   - 추가 GRUB_SERIAL_COMMAND="serial --speed=115200".

   다음은 /etc/default/grub의 예제입니다. 시스템 설정에 따라 구성을 변경해야 할 수 있습니다.

   ```
   GRUB_TIMEOUT=1
   GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
   GRUB_DEFAULT=saved
   GRUB_DISABLE_SUBMENU=true
   GRUB_CMDLINE_LINUX="console=tty0 console=ttyS0,115200n8 net.ifnames=0
    rd.blacklist=nouveau nvme_core.io_timeout=4294967295 crashkernel=auto"
   GRUB_DISABLE_RECOVERY="true"
   GRUB_ENABLE_BLSCFG=true
   GRUB_TERMINAL="console serial"
   GRUB_SERIAL_COMMAND="serial --speed=115200"
   ```
3. 다음 명령을 실행하여 업데이트된 구성을 적용합니다.

   ```
   [ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg --update-bls-cmdline
   ```

   RHEL 9.2 이하의 경우 다음 명령을 사용합니다.

   ```
   [ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
   ```

CentOS

CentOS AMI를 사용하여 시작되는 인스턴스의 경우 GRUB은 기본적으로 직렬 콘솔에 대해 구성됩니다.

다음은 /etc/default/grub의 예제입니다. 시스템 설정에 따라 구성이 다를 수 있습니다.

```
GRUB_TIMEOUT=1
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL="serial console"
GRUB_SERIAL_COMMAND="serial --speed=115200"
GRUB_CMDLINE_LINUX="console=tty0 crashkernel=auto console=ttyS0,115200"
GRUB_DISABLE_RECOVERY="true"
```

(Linux 인스턴스) SysRq 구성

SysRq를 구성하려면 현재 부팅 주기에 대해 SysRq 명령을 활성화합니다. 구성을 영구적으로 만들려면 후속 부팅에 대해서도 SysRq 명령을 활성화할 수 있습니다.

현재 부팅 주기에 대해 모든 SysRq 명령을 활성화하려면

1. 인스턴스에 연결합니다.
2. 다음 명령을 실행합니다.

```
[ec2-user ~]$ sudo sysctl -w kernel.sysrq=1
```

> **Note**
>
> 이 설정은 다음 재부팅 시 지워집니다.

후속 부팅에 대해 모든 SysRq 명령을 활성화하려면

1. `/etc/sysctl.d/99-sysrq.conf` 파일을 생성하고 자주 사용하는 편집기에서 엽니다.

```
[ec2-user ~]$ sudo vi /etc/sysctl.d/99-sysrq.conf
```

2. 다음 행을 추가합니다.

```
kernel.sysrq=1
```

3. 인스턴스를 재부팅하여 변경 사항을 적용합니다.

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

4. `login` 프롬프트에서 이전에 설정한 암호 기반 사용자의 사용자 이름을 입력한 다음 Enter 키를 누릅니다.
5. `Password` 프롬프트에서 암호를 입력한 다음 Enter 키를 누릅니다.

(Windows 인스턴스) SAC 및 부팅 메뉴 활성화

> **Note**
>
> 인스턴스에서 SAC를 활성화하면 암호 검색에 의존하는 EC2 서비스가 Amazon EC2 콘솔에서 작동하지 않습니다. Amazon EC2 Windows 시작 에이전트(EC2Config, EC2Launch v1 및 EC2Launch v2)는 직렬 콘솔을 사용하여 다양한 작업을 실행합니다. 인스턴스에서 SAC를 사용 설정하면 이러한 작업이 성공적으로 수행되지 않습니다. Amazon EC2 Windows 시작 에이전트에 대한 자세한 내용은 the section called “Windows 인스턴스 구성” 섹션을 참조하세요. SAC를 활성화하면 나중에 비활성화할 수 있습니다. 자세한 내용은 SAC 및 부팅 메뉴 비활성화 섹션을 참조하세요.

다음 방법 중 하나를 사용하여 인스턴스에서 SAC 및 부팅 메뉴를 활성화합니다.

PowerShell

Windows 인스턴스에서 SAC 및 부팅 메뉴를 활성화하려면

1. 인스턴스에 연결하고 상승된 PowerShell 명령줄에서 다음 단계를 수행합니다.
2. SAC를 활성화합니다.

```
bcdedit /ems '{current}' on
bcdedit /emssettings EMSPORT:1 EMSBAUDRATE:115200
```

3. 부팅 메뉴를 활성화합니다.

```
bcdedit /set '{bootmgr}' displaybootmenu yes
bcdedit /set '{bootmgr}' timeout 15
bcdedit /set '{bootmgr}' bootems yes
```

4. 인스턴스를 재부팅하여 업데이트된 구성을 적용합니다.

```
shutdown -r -t 0
```

Command prompt

Windows 인스턴스에서 SAC 및 부팅 메뉴를 활성화하려면

1. 인스턴스에 연결하고 명령 프롬프트에서 다음 단계를 수행합니다.
2. SAC를 활성화합니다.

```
bcdedit /ems {current} on
bcdedit /emssettings EMSPORT:1 EMSBAUDRATE:115200
```

3. 부팅 메뉴를 활성화합니다.

```
bcdedit /set {bootmgr} displaybootmenu yes
bcdedit /set {bootmgr} timeout 15
bcdedit /set {bootmgr} bootems yes
```

4. 인스턴스를 재부팅하여 업데이트된 구성을 적용합니다.

```
shutdown -r -t 0
```

EC2 직렬 콘솔에 대한 액세스 구성

직렬 콘솔에 대한 액세스를 구성하려면 계정 수준에서 직렬 콘솔 액세스 권한을 부여한 다음 사용자에게 액세스 권한을 부여하는 IAM 정책을 구성해야 합니다. Linux 인스턴스의 경우 사용자가 직렬 콘솔을 사용하여 문제를 해결할 수 있도록 모든 인스턴스에서 암호 기반 사용자도 구성해야 합니다.

시작하기 전에 필수 조건을 확인하세요.

내용

- EC2 직렬 콘솔에 대한 액세스 수준
- EC2 직렬 콘솔에 대한 계정 액세스 관리
- EC2 직렬 콘솔 액세스에 대한 IAM 정책 구성
- Linux 인스턴스에서 OS 사용자 암호 설정

EC2 직렬 콘솔에 대한 액세스 수준

기본적으로 계정 수준에서는 Serial 콘솔에 액세스할 수 없습니다. 계정 수준에서 Serial 콘솔에 대한 액세스 권한을 명시적으로 부여해야 합니다. 자세한 내용은 EC2 직렬 콘솔에 대한 계정 액세스 관리 섹션을 참조하세요.

서비스 제어 정책(SCP)을 사용하여 조직 내의 직렬 콘솔에 대한 액세스를 허용할 수 있습니다. 그런 다음 IAM 정책을 사용하여 액세스를 제어함으로써 사용자 수준에서 세분화된 액세스를 제어할 수 있습니다. SCP와 IAM 정책을 조합하여 사용하면 직렬 콘솔에 대한 액세스를 다양한 수준에서 제어할 수 있습니다.

조직 수준

서비스 제어 정책(SCP)을 사용하여 조직 내 멤버 계정에 대한 직렬 콘솔 액세스를 허용할 수 있습니다. SCP에 대한 자세한 내용은 AWS Organizations 사용 설명서에서 서비스 제어 정책을 참조하세요.

인스턴스 수준

IAM PrincipalTag 및 ResourceTag 구성을 사용하고 인스턴스 ID로 인스턴스를 지정하여 직렬 콘솔 액세스 정책을 구성할 수 있습니다. 자세한 내용은 EC2 직렬 콘솔 액세스에 대한 IAM 정책 구성 섹션을 참조하세요.

사용자 수준

지정된 사용자가 SSH 퍼블릭 키를 특정 인스턴스의 직렬 콘솔 서비스로 푸시할 수 있는 권한을 허용하거나 거부하는 IAM 정책을 구성하여 사용자 수준에서 액세스를 구성할 수 있습니다. 자세한 내용은 EC2 직렬 콘솔 액세스에 대한 IAM 정책 구성 섹션을 참조하세요.

OS 수준(Linux 인스턴스만 해당)

게스트 OS 수준에서 사용자 암호를 설정할 수 있습니다. 이렇게 하면 일부 사용 사례에서 직렬 콘솔에 대한 액세스를 제공할 수 있습니다. 그러나 로그를 모니터링하는 경우에는 암호 기반 사용자가 필요하지 않습니다. 자세한 내용은 Linux 인스턴스에서 OS 사용자 암호 설정 섹션을 참조하세요.

EC2 직렬 콘솔에 대한 계정 액세스 관리

기본적으로 계정 수준에서는 Serial 콘솔에 액세스할 수 없습니다. 계정 수준에서 Serial 콘솔에 대한 액세스 권한을 명시적으로 부여해야 합니다.

> **Note**
>
> 이 설정은 계정 수준에서 직접 구성되거나 선언적 정책을 사용하여 구성됩니다. 이는 직렬 콘솔에 대한 액세스 권한을 부여하려는 각 AWS 리전에서 구성해야 합니다. 선언적 정책을 사용하면 여러 리전과 여러 계정에 동시에 설정을 적용할 수 있습니다. 선언적 정책을 사용 중인 경우 계정 내에서 직접 설정을 수정할 수 없습니다. 이 주제에서는 계정 내에서 직접 설정을 구성하는 방법을 설명합니다. 선언적 정책 사용에 대한 자세한 내용은 AWS Organizations 사용 설명서의 선언적 정책을 참조하세요.

목차

- 사용자에게 계정 액세스 관리 권한 부여
- 직렬 콘솔에 대한 계정 액세스 상태 보기
- 직렬 콘솔에 대한 계정 액세스 권한 부여
- 직렬 콘솔에 대한 계정 액세스 거부

사용자에게 계정 액세스 관리 권한 부여

사용자가 EC2 직렬 콘솔에 대한 계정 액세스를 관리할 수 있도록 허용하려면 필요한 IAM 권한을 부여해야 합니다.

다음 정책은 계정 상태를 보고 EC2 직렬 콘솔에 대한 계정 액세스를 허용 및 차단할 수 있는 권한을 부여합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:GetSerialConsoleAccessStatus",
                "ec2:EnableSerialConsoleAccess",
                "ec2:DisableSerialConsoleAccess"
            ],
            "Resource": "*"
        }
    ]
}
```

자세한 내용은 IAM 사용 설명서에서 IAM 정책 생성을 참조하세요.

직렬 콘솔에 대한 계정 액세스 상태 보기

Console

직렬 콘솔에 대한 계정 액세스 권한을 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 [EC2 대시보드(EC2 Dashboard)]를 선택합니다.
3. 계정 속성(Account attributes)에서 EC2 직렬 콘솔(EC2 Serial Console)을 선택합니다.
4. EC2 직렬 콘솔 탭에서 EC2 직렬 콘솔 액세스 값은 허용됨됨 또는 금지됨입니다.

AWS CLI

직렬 콘솔에 대한 계정 액세스 권한을 보려면

get-serial-console-access-status 명령을 사용합니다.

```
aws ec2 get-serial-console-access-status
```

출력의 예시는 다음과 같습니다. 값이 `true`이면 계정이 직렬 콘솔에 액세스할 수 있음을 나타냅니다.

```
{
    "SerialConsoleAccessEnabled": true,
    "ManagedBy": "account"
}
```

`ManagedBy` 필드는 설정을 구성한 엔터티를 나타냅니다. 가능한 값은 `account`(직접 구성함) 또는 `declarative-policy`입니다. 자세한 내용은 AWS Organizations 사용 설명서의 선언적 정책을 참조하세요.

PowerShell

직렬 콘솔에 대한 계정 액세스 권한을 보려면

Get-EC2SerialConsoleAccessStatus cmdlet을 사용합니다.

```
Get-EC2SerialConsoleAccessStatus -Select *
```

출력의 예시는 다음과 같습니다. 값이 True이면 계정이 직렬 콘솔에 액세스할 수 있음을 나타냅니다.

```
ManagedBy SerialConsoleAccessEnabled
--------- --------------------------
account   True
```

ManagedBy 필드는 설정을 구성한 엔터티를 나타냅니다. 가능한 값은 account(직접 구성함) 또는 declarative-policy입니다. 자세한 내용은 AWS Organizations 사용 설명서의 선언적 정책을 참조하세요.

직렬 콘솔에 대한 계정 액세스 권한 부여

Console

계정에 직렬 콘솔 액세스 권한을 부여하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 [EC2 대시보드(EC2 Dashboard)]를 선택합니다.
3. 계정 속성(Account attributes)에서 EC2 직렬 콘솔(EC2 Serial Console)을 선택합니다.
4. 관리를 선택합니다.
5. 계정에 있는 모든 인스턴스의 EC2 직렬 콘솔에 대한 액세스를 허용하려면 허용 확인란을 선택합니다.
6. 업데이트를 선택합니다.

AWS CLI

계정에 직렬 콘솔 액세스 권한을 부여하려면

enable-serial-console-access 명령을 사용합니다.

```
aws ec2 enable-serial-console-access
```

출력의 예시는 다음과 같습니다.

```
{
    "SerialConsoleAccessEnabled": true
}
```

PowerShell

계정에 직렬 콘솔 액세스 권한을 부여하려면

Enable-EC2SerialConsoleAccess cmdlet을 사용합니다.

```
Enable-EC2SerialConsoleAccess
```

출력의 예시는 다음과 같습니다.

```
True
```

직렬 콘솔에 대한 계정 액세스 거부

Console

직렬 콘솔에 대한 계정 액세스를 거부하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 [EC2 대시보드(EC2 Dashboard)]를 선택합니다.
3. 계정 속성(Account attributes)에서 EC2 직렬 콘솔(EC2 Serial Console)을 선택합니다.
4. 관리를 선택합니다.
5. 계정에 있는 모든 인스턴스의 EC2 직렬 콘솔에 대한 액세스를 방지하려면 허용 확인란을 선택 취소합니다.
6. 업데이트를 선택합니다.

AWS CLI

직렬 콘솔에 대한 계정 액세스를 거부하려면

disable-serial-console-access 명령을 사용합니다.

```
aws ec2 disable-serial-console-access
```

출력의 예시는 다음과 같습니다.

```
{
    "SerialConsoleAccessEnabled": false
```

```
}
```

PowerShell

직렬 콘솔에 대한 계정 액세스를 거부하려면

Disable-EC2SerialConsoleAccess cmdlet을 사용합니다.

```
Disable-EC2SerialConsoleAccess
```

출력의 예시는 다음과 같습니다.

```
False
```

EC2 직렬 콘솔 액세스에 대한 IAM 정책 구성

기본적으로 사용자는 직렬 콘솔에 액세스할 수 없습니다. 조직은 사용자에게 필요한 액세스 권한을 부여하는 IAM 정책을 구성해야 합니다. 자세한 내용은 IAM 사용 설명서에서 IAM 정책 생성을 참조하세요.

직렬 콘솔 액세스의 경우 `ec2-instance-connect:SendSerialConsoleSSHPublicKey` 작업이 포함된 JSON 정책 문서를 생성합니다. 이 작업은 사용자에게 직렬 콘솔 서비스에 퍼블릭 키를 푸시하여 직렬 콘솔 세션을 시작할 수 있는 권한을 부여합니다. 특정 EC2 인스턴스로 액세스를 제한하는 것이 좋습니다. 그렇게 하지 않으면 이 권한이 있는 모든 사용자가 모든 EC2 인스턴스의 직렬 콘솔에 연결할 수 있습니다.

예제 IAM 정책

- 직렬 콘솔에 대한 액세스를 명시적으로 허용
- 직렬 콘솔에 대한 액세스를 명시적으로 거부
- 리소스 태그를 사용하여 직렬 콘솔에 대한 액세스 제어

직렬 콘솔에 대한 액세스를 명시적으로 허용

기본적으로는 누구도 직렬 콘솔에 액세스할 수 없습니다. 직렬 콘솔에 대한 액세스 권한을 부여하려면 액세스를 명시적으로 허용하는 정책을 구성해야 합니다. 특정 인스턴스로 액세스를 제한하는 정책을 구성하는 것이 좋습니다.

다음 정책은 인스턴스 ID로 식별되는 특정 인스턴스의 직렬 콘솔에 대한 액세스를 허용합니다.

DescribeInstances, DescribeInstanceTypes 및 GetSerialConsoleAccessStatus 작업은 리소스 수준 권한을 지원하지 않으므로 이러한 작업에 대해 *(별표)로 표시된 모든 리소스를 지정해야 합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowDescribeInstances",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceTypes",
                "ec2:GetSerialConsoleAccessStatus"
            ],
             "Resource": "*"
        },
        {
            "Sid": "AllowinstanceBasedSerialConsoleAccess",
            "Effect": "Allow",
            "Action": [
                "ec2-instance-connect:SendSerialConsoleSSHPublicKey"
            ],
            "Resource": "arn:aws:ec2:region:account-id:instance/i-0598c7d356eba48d7"
        }
    ]
}
```

직렬 콘솔에 대한 액세스를 명시적으로 거부

다음 IAM 정책은 *(별표)로 표시된 모든 인스턴스의 직렬 콘솔에 대한 액세스를 허용하고 인스턴스 ID로 식별되는 특정 인스턴스의 직렬 콘솔에 대한 액세스를 명시적으로 거부합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSerialConsoleAccess",
            "Effect": "Allow",
            "Action": [
                "ec2-instance-connect:SendSerialConsoleSSHPublicKey",
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceTypes",
```

```
                "ec2:GetSerialConsoleAccessStatus"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DenySerialConsoleAccess",
            "Effect": "Deny",
            "Action": [
                "ec2-instance-connect:SendSerialConsoleSSHPublicKey"
            ],
            "Resource": "arn:aws:ec2:region:account-id:instance/i-0598c7d356eba48d7"
        }
    ]
}
```

리소스 태그를 사용하여 직렬 콘솔에 대한 액세스 제어

리소스 태그를 사용하여 인스턴스의 직렬 콘솔에 대한 액세스를 제어할 수 있습니다.

속성 기반 액세스 제어는 사용자 및 AWS 리소스에 연결할 수 있는 태그를 기반으로 권한을 정의하는 권한 부여 전략입니다. 예를 들어, 다음 정책은 인스턴스의 리소스 태그와 보안 주체의 태그에 동일한 `SerialConsole` 태그 키 값이 포함되는 경우에만 사용자가 인스턴스에 대한 직렬 콘솔 연결을 시작하는 것을 허용합니다.

태그를 사용하여 AWS 리소스에 대한 액세스를 제어하는 방법에 대한 자세한 내용은 IAM 사용 설명서에서 AWS 리소스에 대한 액세스 제어를 참조하세요.

`DescribeInstances`, `DescribeInstanceTypes` 및 `GetSerialConsoleAccessStatus` 작업은 리소스 수준 권한을 지원하지 않으므로 이러한 작업에 대해 *(별표)로 표시된 모든 리소스를 지정해야 합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowDescribeInstances",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceTypes",
                "ec2:GetSerialConsoleAccessStatus"
            ],
            "Resource": "*"
```

```
        },
        {
            "Sid": "AllowTagBasedSerialConsoleAccess",
            "Effect": "Allow",
            "Action": [
                "ec2-instance-connect:SendSerialConsoleSSHPublicKey"
            ],
            "Resource": "arn:aws:ec2:region:account-id:instance/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/SerialConsole":
 "${aws:PrincipalTag/SerialConsole}"
                }
            }
        }
    ]
}
```

Linux 인스턴스에서 OS 사용자 암호 설정

> **Note**
>
> 이 섹션은 Linux 인스턴스에만 적용됩니다.

직렬 콘솔에는 암호 없이 연결할 수 있습니다. 그러나 Linux 인스턴스 문제 해결에 직렬 콘솔을 사용하려면 인스턴스에 암호 기반 OS 사용자가 있어야 합니다.

루트 사용자를 포함하여 모든 OS 사용자의 암호를 설정할 수 있습니다. 루트 사용자는 모든 파일을 수정할 수 있지만 각 OS 사용자는 제한된 권한을 가질 수 있습니다.

직렬 콘솔을 사용할 모든 인스턴스에 대해 사용자 암호를 설정해야 합니다. 각 인스턴스에 대해 한 번만 수행하면 됩니다.

> **Note**
>
> 다음 지침은 AWS가 제공한 Linux AMI를 사용하여 인스턴스를 시작한 경우에만 적용됩니다. 기본적으로 AWS가 제공한 AMI는 암호 기반 사용자로 구성되지 않기 때문입니다. 이미 루트 사용자 암호가 구성되어 있는 AMI를 사용하여 인스턴스를 시작한 경우 이 지침을 건너뛸 수 있습니다.

Linux 인스턴스에서 OS 사용자 암호를 설정하려면 다음을 수행합니다.

1. 인스턴스에 연결합니다. EC2 직렬 콘솔 연결 방법을 제외한 모든 방법으로 인스턴스에 연결할 수 있습니다.
2. 사용자의 암호를 설정하려면 passwd 명령을 사용합니다. 다음 예제에서 사용자는 `root`입니다.

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

   다음은 예제 출력입니다.

   ```
   Changing password for user root.
   New password:
   ```

3. `New password` 프롬프트에서 새 암호를 입력합니다.
4. 프롬프트에서 암호를 다시 입력합니다.

EC2 직렬 콘솔에 연결

Amazon EC2 콘솔 또는 SSH를 통해 EC2 인스턴스의 직렬 콘솔에 연결할 수 있습니다. 직렬 콘솔에 연결한 후에는 직렬 콘솔을 사용하여 부팅, 네트워크 구성 및 기타 문제를 해결할 수 있습니다. 문제 해결에 대한 자세한 내용은 EC2 직렬 콘솔을 사용하여 Amazon EC2 인스턴스 문제 해결 섹션을 참조하세요.

고려 사항

- 인스턴스당 하나의 활성 직렬 콘솔 연결만 지원됩니다.
- 일반적으로 직렬 콘솔 연결은 연결을 해제하지 않는 한 1시간 동안 지속됩니다. 그러나 시스템 유지 관리 중에는 Amazon EC2에서 직렬 콘솔 세션 연결을 해제합니다.

  연결 기간은 IAM 자격 증명 기간에 따라 결정되지 않습니다. IAM 자격 증명이 만료된 경우 직렬 콘솔 연결의 최대 기간에 도달할 때까지 연결이 계속 유지됩니다. EC2 직렬 콘솔의 콘솔 환경을 사용할 때 IAM 자격 증명이 만료되면 브라우저 페이지를 닫아 연결을 종료합니다.
- 직렬 콘솔에서 연결을 끊은 후 새 세션을 허용하려면 세션을 삭제하는 데 30초가 걸립니다.
- 지원되는 직렬 콘솔 포트: `ttyS0`(Linux 인스턴스) 및 `COM1`(Windows 인스턴스)
- 직렬 콘솔에 연결하는 경우 인스턴스 처리량이 약간 저하될 수 있습니다.

주제

- 브라우저 기반 클라이언트를 사용하여 연결
- 자체 키 및 SSH 클라이언트를 사용하여 연결
- EC2 직렬 콘솔 엔드포인트 및 지문

브라우저 기반 클라이언트를 사용하여 연결

브라우저 기반 클라이언트를 사용하여 EC2 인스턴스의 직렬 콘솔에 연결할 수 있습니다. Amazon EC2 콘솔에서 인스턴스를 선택하고 직렬 콘솔에 연결하도록 선택하면 됩니다. 브라우저 기반 클라이언트에서 사용 권한이 처리되고 성공적으로 연결됩니다.

EC2 직렬 콘솔은 대부분의 브라우저에서 작동하며 키보드 및 마우스 입력을 지원합니다.

연결하기 전에 사전 요구 사항을 충족하는지 확인합니다.

브라우저 기반 클라이언트를 사용하여 인스턴스의 직렬 포트에 연결하려면(Amazon EC2 콘솔)

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 작업(Actions), 모니터링 및 문제 해결(Monitor and troubleshoot), EC2 직렬 콘솔(EC2 Serial Console), 연결(Connect)을 선택합니다.

   또는 인스턴스를 선택하고 연결(Connect), EC2 직렬 콘솔(EC2 Serial Console), 연결(Connect)을 선택합니다.

   브라우저 내 터미널 창이 열립니다.
4. Enter(입력) 키를 누릅니다. 로그인 프롬프트가 반환되면 직렬 콘솔에 연결된 것입니다.

   화면이 검은색으로 유지되면 다음 정보를 사용하여 직렬 콘솔 연결 문제를 해결할 수 있습니다.

   - 직렬 콘솔에 대한 액세스를 구성했는지 확인합니다. 자세한 내용은 EC2 직렬 콘솔에 대한 액세스 구성 섹션을 참조하세요.
   - (Linux 인스턴스만 해당) SysRq를 사용하여 직렬 콘솔에 연결합니다. SysRq에서는 브라우저 기반 클라이언트를 통해 연결할 필요가 없습니다. 자세한 내용은 (Linux 인스턴스) SysRq를 사용하여 인스턴스 문제 해결 섹션을 참조하세요.
   - (Linux 인스턴스만 해당) getty를 다시 시작합니다. 인스턴스에 대한 SSH 액세스 권한이 있는 경우 SSH를 사용하여 인스턴스에 연결하고 다음 명령을 사용하여 getty를 다시 시작합니다.

     ```
     [ec2-user ~]$ sudo systemctl restart serial-getty@ttyS0
     ```

- 인스턴스를 재부팅합니다. SysRq(Linux 인스턴스), EC2 콘솔 또는 AWS CLI를 사용하여 인스턴스를 재부팅할 수 있습니다. 자세한 내용은 (Linux 인스턴스) SysRq를 사용하여 인스턴스 문제 해결(Linux 인스턴스) 또는 Amazon EC2 인스턴스 재부팅 섹션을 참조하세요.

5. (Linux 인스턴스만 해당) `login` 프롬프트에서 이전에 설정한 암호 기반 사용자의 사용자 이름을 입력한 다음 Enter 키를 누릅니다.
6. (Linux 인스턴스만 해당) `Password` 프롬프트에서 암호를 입력한 다음 Enter 키를 누릅니다.

이제 인스턴스에 로그인되었고 직렬 콘솔을 사용하여 문제 해결을 수행할 수 있습니다.

자체 키 및 SSH 클라이언트를 사용하여 연결

자체 SSH 키를 사용하면서 직렬 콘솔 API를 사용하여 선택한 SSH 클라이언트의 인스턴스에 연결할 수 있습니다. 이렇게 하면 직렬 콘솔 기능을 활용하여 퍼블릭 키를 인스턴스로 푸시할 수 있습니다.

연결하기 전에 사전 요구 사항을 충족하는지 확인합니다.

SSH를 사용하여 인스턴스의 직렬 콘솔에 연결하려면

1. SSH 퍼블릭 키를 인스턴스로 푸시하여 직렬 콘솔 세션 시작

   send-serial-console-ssh-public-key 명령을 사용하여 SSH 퍼블릭 키를 인스턴스에 푸시합니다. 이렇게 하면 직렬 콘솔 세션이 시작됩니다.

   이 인스턴스에 대해 직렬 콘솔 세션이 이미 시작된 경우 한 번에 하나의 세션만 열 수 있기 때문에 명령이 실패합니다. 직렬 콘솔에서 연결을 끊은 후 새 세션을 허용하려면 세션을 삭제하는 데 30초가 걸립니다.

   ```
   aws ec2-instance-connect send-serial-console-ssh-public-key \
       --instance-id i-001234a4bf70dec41EXAMPLE \
       --serial-port 0 \
       --ssh-public-key file://my_key.pub \
       --region us-east-1
   ```

2. 프라이빗 키를 사용하여 직렬 콘솔에 연결

   직렬 콘솔 서비스에서 퍼블릭 키를 제거하기 전에 ssh 명령을 사용하여 직렬 콘솔에 연결합니다. 키는 60초 후에 제거됩니다.

   퍼블릭 키에 해당하는 프라이빗 키를 사용합니다.

사용자 이름 형식은 인스턴스 ID와 포트 0으로 구성되는 `instance-id.port0`입니다. 다음 예에서 사용자 이름은 `i-001234a4bf70dec41EXAMPLE.port0`입니다.

직렬 콘솔 서비스의 엔드포인트는 각 리전마다 다릅니다. 각 리전의 엔드포인트는 EC2 직렬 콘솔 엔드포인트 및 지문 표를 참조하세요. 다음 예에서 직렬 콘솔 서비스는 us-east-1 리전에 있습니다.

```
ssh -i my_key i-001234a4bf70dec41EXAMPLE.port0@serial-console.ec2-instance-connect.us-east-1.aws
```

다음 예제에서는 `timeout 3600`을 사용하여 SSH 세션이 1시간 후에 종료되도록 설정합니다. 세션 중에 시작된 프로세스는 세션이 종료된 후에도 인스턴스에서 계속 실행될 수 있습니다.

```
timeout 3600 ssh -i my_key i-001234a4bf70dec41EXAMPLE.port0@serial-console.ec2-instance-connect.us-east-1.aws
```

3. (선택 사항) 지문 확인

   직렬 콘솔에 처음 연결하면 지문을 확인하라는 메시지가 표시됩니다. 직렬 콘솔 지문을 표시된 지문과 비교하여 확인할 수 있습니다. 이들 지문이 일치하지 않으면 누군가가 "메시지 가로채기(man-in-the-middle)" 공격을 시도하고 있는 것일 수 있습니다. 일치하는 경우 직렬 콘솔에 안심하고 연결할 수 있습니다.

   다음 지문은 us-east-1 리전의 직렬 콘솔 서비스에 대한 것입니다. 각 리전의 지문은 EC2 직렬 콘솔 엔드포인트 및 지문 섹션을 참조하세요.

   ```
   SHA256:dXwn5ma/xadVMeBZGEru5l2gx+yI5LDiJaLUcz0FMmw
   ```

   > **Note**
   >
   > 지문은 직렬 콘솔에 처음 연결할 때만 나타납니다.

4. Enter(입력) 키를 누릅니다. 프롬프트가 반환되면 직렬 콘솔에 연결된 것입니다.

   화면이 검은색으로 유지되면 다음 정보를 사용하여 직렬 콘솔 연결 문제를 해결할 수 있습니다.

   - 직렬 콘솔에 대한 액세스를 구성했는지 확인합니다. 자세한 내용은 EC2 직렬 콘솔에 대한 액세스 구성 섹션을 참조하세요.

- (Linux 인스턴스만 해당) SysRq를 사용하여 직렬 콘솔에 연결합니다. SysRq에서는 SSH를 통해 연결할 필요가 없습니다. 자세한 내용은 (Linux 인스턴스) SysRq를 사용하여 인스턴스 문제 해결 섹션을 참조하세요.
- (Linux 인스턴스만 해당) getty를 다시 시작합니다. 인스턴스에 대한 SSH 액세스 권한이 있는 경우 SSH를 사용하여 인스턴스에 연결하고 다음 명령을 사용하여 getty를 다시 시작합니다.

```
[ec2-user ~]$ sudo systemctl restart serial-getty@ttyS0
```

- 인스턴스를 재부팅합니다. SysRq(Linux 인스턴스만 해당), EC2 콘솔 또는 AWS CLI를 사용하여 인스턴스를 재부팅할 수 있습니다. 자세한 내용은 (Linux 인스턴스) SysRq를 사용하여 인스턴스 문제 해결(Linux 인스턴스만 해당) 또는 Amazon EC2 인스턴스 재부팅 섹션을 참조하세요.

5. (Linux 인스턴스만 해당) `login` 프롬프트에서 이전에 설정한 암호 기반 사용자의 사용자 이름을 입력한 다음 Enter 키를 누릅니다.
6. (Linux 인스턴스만 해당) `Password` 프롬프트에서 암호를 입력한 다음 Enter 키를 누릅니다.

이제 인스턴스에 로그인되었고 직렬 콘솔을 사용하여 문제 해결을 수행할 수 있습니다.

EC2 직렬 콘솔 엔드포인트 및 지문

다음은 EC2 직렬 콘솔의 서비스 엔드포인트 및 지문입니다. 프로그래밍 방식으로 인스턴스의 직렬 콘솔에 연결하려면 EC2 직렬 콘솔 엔드포인트를 사용하세요. EC2 직렬 콘솔 엔드포인트 및 지문은 각 AWS 리전에 대해 고유합니다.

리전 이름	지역	엔드포인트	지문
미국 동부(오하이오)	us-east-2	serial-console.ec2-instance-connect.us-east-2.aws	SHA256:EhwPkTzRtTY7TRSzz26XbB0/HvV9jRM7mCZN0xw/d/0
미국 동부(버지니아 북부)	us-east-1	serial-console.ec2-instance-connect.us-east-1.aws	SHA256:dXwn5ma/xadVMeBZGEru5l2gx+yI5LDiJaLUcz0FMmw

리전 이름	지역	엔드포인트	지문
미국 서부(캘리포니아 북부)	us-west-1	serial-console.ec2-instance-connect.us-west-1.aws	SHA256:OH ldlcMET8u 7QLSX3jmR TRAPFHVtq byoLZBMUCqiH3Y
미국 서부(오리건)	us-west-2	serial-console.ec2-instance-connect.us-west-2.aws	SHA256:EM CIe23TqKaBI6yGHain qZcMwqNkD hhAVHa1O2JxVUc
아프리카(케이프타운)	af-south-1	ec2-serial-console.af-south-1.api.aws	SHA256:RM WWZ2fVePe JUqzjO5jL2KIgXsczo Hlz21Ed00biiWI
아시아 태평양(홍콩)	ap-east-1	ec2-serial-console.ap-east-1.api.aws	SHA256:T0Q1IpiXxCh oZHplnAkjbP7tkm2xX ViC9bJFsjYnifk
아시아 태평양(하이데라바드)	ap-south-2	ec2-serial-console.ap-south-2.api.aws	SHA256:WJ gPBSwV4/shN +OPITValoewAuYj1 5DVW845JEhDKRs
아시아 태평양(자카르타)	ap-southeast-3	ec2-serial-console.ap-southeast-3.api.aws	SHA256:5ZwgrCh+lfn s32XITqL/4O0zIfbx4 bZgsYFqy3o8mIk
아시아 태평양(말레이시아)	ap-southeast-5	ec2-serial-console.ap-southeast-5.api.aws	SHA256:cQ XTHQMRcqR dIjmAGoAM BSExeoRob YyRwT67yTjnEiA

리전 이름	지역	엔드포인트	지문
아시아 태평양(멜버른)	ap-southeast-4	ec2-serial-console.ap-southeast-4.api.aws	SHA256:Avaq27hFgLvjn5gTSShZ0oV7h90p0GG46wfOeT6ZJvM
아시아 태평양(뭄바이)	ap-south-1	serial-console.ec2-instance-connect.ap-south-1.aws	SHA256:oBLXcYmklqHHEbliARxEgH8IsO51rezTPiSM35BsU40
아시아 태평양(오사카)	ap-northeast-3	ec2-serial-console.ap-northeast-3.api.aws	SHA256:Am0/jiBKBnBuFnHr9aXsgEV3G8Tu/vVHFXE/3UcyjsQ
아시아 태평양(서울)	ap-northeast-2	serial-console.ec2-instance-connect.ap-northeast-2.aws	SHA256:FoqWXNX+DZ++GuNTztg9PK49WYMqBX+FrcZM2dSrqrI
아시아 태평양(싱가포르)	ap-southeast-1	serial-console.ec2-instance-connect.ap-southeast-1.aws	SHA256:PLFNn7WnCQDHx3qmwLu1Gy/O8TUX7LQgZuaC6L45CoY
아시아 태평양(시드니)	ap-southeast-2	serial-console.ec2-instance-connect.ap-southeast-2.aws	SHA256:yFvMwUK9lEUQjQTRoXXzuN+cW9/VSe9W984Cf5Tgzo4

리전 이름	지역	엔드포인트	지문
아시아 태평양(도쿄)	ap-northeast-1	serial-console.ec2-instance-connect.ap-northeast-1.aws	SHA256:RQfsDCZTOfQawewTRDV1t9Em/HMrFQe+CRlIOT5um4k
캐나다(중부)	ca-central-1	serial-console.ec2-instance-connect.ca-central-1.aws	SHA256:P2O2jOZwmpMwkpO6YW738FIOTHdUTyEv2gczYMMO7s4
캐나다 서부(캘거리)	ca-west-1	ec2-serial-console.ca-west-1.api.aws	SHA256:s3rc8Il2xhbhr3iedjJNxGAFLPGOLjx7IxxXrGckk6Q
중국(베이징)	cn-north-1	ec2-serial-console.cn-north-1.api.amazonwebservices.com.cn	SHA256:2gHVFy4H7uU3+WaFUxD28v/ggMeqjvSlgngpgLgGT+Y
중국(닝샤)	cn-northwest-1	ec2-serial-console.cn-northwest-1.api.amazonwebservices.com.cn	SHA256:TdgrNZkiQOdVfYEBUhO4SzUA09VWI5rYOZGTogpwmiM
유럽(프랑크푸르트)	eu-central-1	serial-console.ec2-instance-connect.eu-central-1.aws	SHA256:aCMFS/yIcOdOlkXvOl8AmZ1Toe+bBnrJJ3Fy0k0De2c

리전 이름	지역	엔드포인트	지문
유럽(아일랜드)	eu-west-1	serial-console.ec2-instance-connect.eu-west-1.aws	SHA256:h2AaGAWO4Hathhtm6ezs3Bj7udgUxi2qTrHjZAwCW6E
유럽(런던)	eu-west-2	serial-console.ec2-instance-connect.eu-west-2.aws	SHA256:a69rd5CE/AEG4Amm53I6lkD1ZPvS/BCV3tTPW2RnJg8
유럽(밀라노)	eu-south-1	ec2-serial-console.eu-south-1.api.aws	SHA256:IC0kOVJnpgFyBVrxn0A7n99ecLbXSX95cuuS7X7QK30
유럽(파리)	eu-west-3	serial-console.ec2-instance-connect.eu-west-3.aws	SHA256:q8ldnAf9pymeNe8BnFVngY3RPAr/kxswJUzfrlxeEWs
유럽(스페인)	eu-south-2	ec2-serial-console.eu-south-2.api.aws	SHA256:GoCW2DFRlu669QNxqFxEcsR6fZUz/4F4n7T45ZcwoEc
유럽(스톡홀름)	eu-north-1	serial-console.ec2-instance-connect.eu-north-1.aws	SHA256:tkGFFUVUDvocDiGSS3Cu8Gdl6w2ul32EPNpKFKLwX84
유럽(취리히)	eu-central-2	ec2-serial-console.eu-central-2.api.aws	SHA256:8Ppx2mBMf6WdCw0NUlzKfwM4/IfRz4OaXFutQXWp6mk

리전 이름	지역	엔드포인트	지문
이스라엘(텔아비브)	il-central-1	ec2-serial-console.il-central-1.api.aws	SHA256:JR 6q8v6kNNP i8+QSFQ4d j5dimNmZP TgwgsM1SNvtYyU
중동(바레인)	me-south-1	ec2-serial-console .me-south-1.api.aws	SHA256:nP jLLKHu2Qn LdUq2kVAr soK5xvPJO MRJKCBzCDqC3k8
중동(UAE)	me-central-1	ec2-serial-console .me-central-1.api.aws	SHA256:zpb5duKiBZ +l0dFwPeyy kB4MPBYhI/ XzXNeFSDKBvLE
남아메리카(상파울루)	sa-east-1	serial-console.ec2-instance-connect.sa-east-1.aws	SHA256:rd2+/32Ognj ew1yVIemENaQzC +Botbih62OqAPDq1dI
AWS GovCloud(미국 동부)	us-gov-east-1	serial-console.ec2 -instance-connect. us-gov-east-1.amaz onaws.com	SHA256:tI we19GWsoy LClrtvu38YEEh+DHIk qnDcZnmtebvF28
AWS GovCloud(미국 서부)	us-gov-west-1	serial-console.ec2 -instance-connect. us-gov-west-1.amaz onaws.com	SHA256:kf OFRWLaOZfB +utbd3bRf8OlPf8nG O2YZLqXZiIw5DQ

EC2 직렬 콘솔 연결 해제

인스턴스의 EC2 직렬 콘솔에 더 이상 연결할 필요가 없으면 연결을 해제할 수 있습니다. 직렬 콘솔과의 연결을 해제해도 인스턴스에서 실행 중인 모든 쉘 세션은 계속 실행됩니다. 쉘 세션을 종료하려면 직렬 콘솔과의 연결을 해제하기 전에 쉘 세션을 종료해야 합니다.

고려 사항

- 일반적으로 직렬 콘솔 연결은 연결을 해제하지 않는 한 1시간 동안 지속됩니다. 그러나 시스템 유지 관리 중에는 Amazon EC2에서 직렬 콘솔 세션 연결을 해제합니다.
- 직렬 콘솔에서 연결을 끊은 후 새 세션을 허용하려면 세션을 삭제하는 데 30초가 걸립니다.

직렬 콘솔과의 연결을 해제하는 방법은 클라이언트에 따라 다릅니다.

브라우저 기반 클라이언트

직렬 콘솔 연결을 해제하려면 직렬 콘솔 브라우저 내 터미널 창을 닫습니다.

표준 OpenSSH 클라이언트

직렬 콘솔 연결을 해제하려면 다음 명령을 사용하여 SSH 연결을 종료합니다. 새 줄 바로 다음에 이 명령을 실행해야 합니다.

```
~.
```

SSH 연결을 닫는 데 사용하는 명령은 사용 중인 SSH 클라이언트에 따라 다를 수 있습니다.

EC2 직렬 콘솔을 사용하여 Amazon EC2 인스턴스 문제 해결

EC2 직렬 콘솔을 사용하면 인스턴스의 직렬 포트에 연결하여 부팅, 네트워크 구성 및 기타 문제를 해결할 수 있습니다.

인스턴스의 운영 체제 지침과 인스턴스에서 구성한 도구의 지침을 사용하세요.

도구

- (Linux 인스턴스) GRUB을 사용하여 인스턴스 문제 해결
- (Linux 인스턴스) SysRq를 사용하여 인스턴스 문제 해결
- (Windows 인스턴스) SAC를 사용하여 인스턴스 문제 해결

사전 조건

시작하기 전에 선택한 문제 해결 도구 구성을 포함한 사전 조건을 완료했는지 확인합니다.

(Linux 인스턴스) GRUB을 사용하여 인스턴스 문제 해결

GNU GRUB(GNU GRand Unified Bootloader의 약어, 일반적으로 GRUB이라고 함)은 대부분의 Linux 운영 체제에 대한 기본 부트 로더입니다. GRUB 메뉴에서 부팅할 커널을 선택하거나 메뉴 항목을 수정하여 커널 부팅 방법을 변경할 수 있습니다. 이는 장애가 발생한 인스턴스의 문제를 해결할 때 유용할 수 있습니다.

부팅 프로세스 중에 GRUB 메뉴가 표시됩니다. 이 메뉴는 일반 SSH를 통해 액세스할 수 없으며 EC2 직렬 콘솔을 통해 액세스할 수 있습니다.

단일 사용자 모드 또는 응급 모드로 부팅할 수 있습니다. 단일 사용자 모드는 낮은 실행 수준에서 커널을 부팅합니다. 예를 들어 파일 시스템을 탑재하지만 네트워크를 활성화하지 않음으로써 인스턴스를 수정하는 데 필요한 유지 관리를 수행할 수 있는 기회를 제공합니다. 응급 모드는 커널이 가능한 가장 낮은 실행 수준에서 실행된다는 점을 제외하면 단일 사용자 모드와 유사합니다.

단일 사용자 모드로 부팅하려면

1. 인스턴스의 직렬 콘솔에 연결합니다.
2. 다음 명령을 사용하여 인스턴스를 재부팅하세요.

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

3. 재부팅하는 동안 GRUB 메뉴가 나타나면 아무 키나 눌러 부팅 프로세스를 중지합니다.
4. GRUB 메뉴에서 화살표 키를 사용하여 부팅할 커널을 선택하고 키보드에서 e 키를 누릅니다.
5. 화살표 키를 사용하여 커널이 있는 줄에 커서를 놓습니다. 이 줄은 인스턴스를 시작할 때 사용된 AMI에 따라 `linux` 또는 `linux16`으로 시작됩니다. Ubuntu의 경우 두 줄이 `linux`로 시작되며 다음 단계에서 수정해야 합니다.
6. 줄 끝에 `single` 단어를 추가합니다.

   다음은 Amazon Linux 2의 예제입니다.

   ```
   linux /boot/vmlinuz-4.14.193-149.317.amzn2.aarch64 root=UUID=d33f9c9a-\
   dadd-4499-938d-ebbf42c3e499 ro  console=tty0 console=ttyS0,115200n8 net.ifname\
   s=0 biosdevname=0 nvme_core.io_timeout=4294967295 rd.emergency=poweroff rd.she\
   ll=0 single
   ```

7. Ctrl+X 키를 눌러 단일 사용자 모드로 부팅합니다.
8. `login` 프롬프트에서 이전에 설정한 암호 기반 사용자의 사용자 이름을 입력한 다음 Enter 키를 누릅니다.
9. `Password` 프롬프트에서 암호를 입력한 다음 Enter 키를 누릅니다.

응급 모드로 부팅하려면

단일 사용자 모드와 동일한 단계를 따르되 6단계에서 `single` 대신 `emergency`라는 단어를 추가합니다.

(Linux 인스턴스) SysRq를 사용하여 인스턴스 문제 해결

"매직 SysRq"라고도 하는 시스템 요청(SysRq) 키를 사용하면 셸 외부에서 커널에 직접 명령을 전송할 수 있습니다. 커널은 수행하는 작업과 관계없이 응답합니다. 예를 들어 인스턴스가 응답을 중지한 경우 SysRq 키를 사용하여 커널에 충돌 또는 재부팅을 지시할 수 있습니다. 자세한 내용은 Wikipedia에서 Magic SysRq key를 참조하세요.

EC2 직렬 콘솔 브라우저 기반 클라이언트 또는 SSH 클라이언트에서 SysRq 명령을 사용할 수 있습니다. 중단 요청을 보내는 명령은 클라이언트마다 다릅니다.

SysRq를 사용하려면 사용 중인 클라이언트에 따라 다음 절차 중 하나를 선택합니다.

Browser-based client

직렬 콘솔 브라우저 기반 클라이언트에서 SysRq를 사용하려면

1. 인스턴스의 직렬 콘솔에 연결합니다.
2. 중단 요청을 보내려면 `CTRL+0`(영)을 누릅니다. 키보드가 지원하는 경우 Pause 또는 Break 키를 사용하여 중단 요청을 보낼 수도 있습니다.

```
[ec2-user ~]$ CTRL+0
```

3. SysRq 명령을 실행하려면 키보드에서 필요한 명령에 해당하는 키를 누릅니다. 예를 들어 SysRq 명령 목록을 표시하려면 h 키를 누릅니다.

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

h 명령은 다음과 비슷한 결과를 출력합니다.

```
[ 1169.389495] sysrq: HELP : loglevel(0-9) reboot(b) crash(c) terminate-all-
tasks(e) memory-full-oom-kill(f) kill-all-tasks(i) thaw-filesystems
(j) sak(k) show-backtrace-all-active-cpus(l) show-memory-usage(m) nice-all-RT-
tasks(n) poweroff(o) show-registers(p) show-all-timers(q) unraw(r
) sync(s) show-task-states(t) unmount(u) show-blocked-tasks(w) dump-ftrace-
buffer(z)
```

SSH client

SSH 클라이언트에서 SysRq를 사용하려면

1. 인스턴스의 직렬 콘솔에 연결합니다.
2. 중단 요청을 보내려면 ~B(물결표 다음에 대문자 B)를 누릅니다.

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

3. SysRq 명령을 실행하려면 키보드에서 필요한 명령에 해당하는 키를 누릅니다. 예를 들어 SysRq 명령 목록을 표시하려면 h 키를 누릅니다.

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

h 명령은 다음과 비슷한 결과를 출력합니다.

```
[ 1169.389495] sysrq: HELP : loglevel(0-9) reboot(b) crash(c) terminate-all-
tasks(e) memory-full-oom-kill(f) kill-all-tasks(i) thaw-filesystems
(j) sak(k) show-backtrace-all-active-cpus(l) show-memory-usage(m) nice-all-RT-
tasks(n) poweroff(o) show-registers(p) show-all-timers(q) unraw(r
) sync(s) show-task-states(t) unmount(u) show-blocked-tasks(w) dump-ftrace-
buffer(z)
```

> **Note**
>
> 중단 요청을 보내는 데 사용하는 명령은 사용 중인 SSH 클라이언트에 따라 다를 수 있습니다.

(Windows 인스턴스) SAC를 사용하여 인스턴스 문제 해결

Windows의 Special Admin Console(SAC) 기능은 Windows 인스턴스 문제를 해결할 수 있는 방법을 제공합니다. 인스턴스의 직렬 콘솔에 연결하고 SAC를 사용하여 부팅 프로세스를 중단하고 Windows를 안전 모드로 부팅할 수 있습니다.

> **Note**
>
> 인스턴스에서 SAC를 활성화하면 암호 검색에 의존하는 EC2 서비스가 Amazon EC2 콘솔에서 작동하지 않습니다. Amazon EC2 Windows 시작 에이전트(EC2Config, EC2Launch v1 및 EC2Launch v2)는 직렬 콘솔을 사용하여 다양한 작업을 실행합니다. 인스턴스에서 SAC를 사용 설정하면 이러한 작업이 성공적으로 수행되지 않습니다. Amazon EC2 Windows 시작 에이전트에 대한 자세한 내용은 the section called “Windows 인스턴스 구성” 섹션을 참조하세요. SAC를 활성화하면 나중에 비활성화할 수 있습니다. 자세한 내용은 SAC 및 부팅 메뉴 비활성화 섹션을 참조하세요.

업무

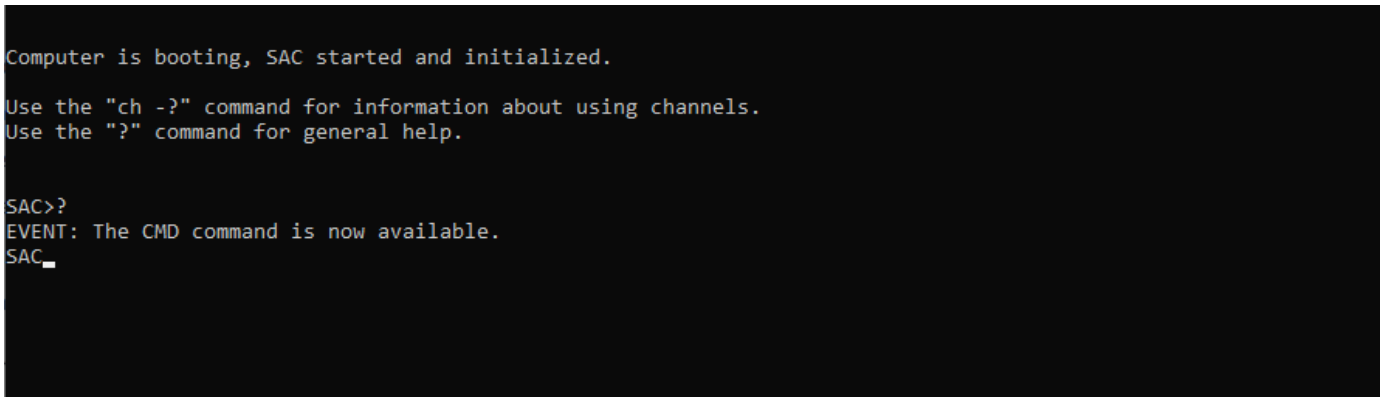
- SAC 사용
- 부팅 메뉴 사용
- SAC 및 부팅 메뉴 비활성화

SAC 사용

SAC를 사용하려면

1. 직렬 콘솔에 연결합니다.

   SAC가 인스턴스에서 활성화된 경우 직렬 콘솔에 SAC> 프롬프트가 표시됩니다.



2. SAC 명령을 표시하려면 ?를 입력하고 Enter를 누릅니다.

   예상 결과

```
SAC>?
ch                    Channel management commands.  Use ch -? for more help.
cmd                   Create a Command Prompt channel.
d                     Dump the current kernel log.
f                     Toggle detailed or abbreviated tlist info.
? or help             Display this list.
i                     List all IP network numbers and their IP addresses.
i <#> <ip> <subnet> <gateway> Set IPv4 addr., subnet and gateway.
id                    Display the computer identification information.
k <pid>               Kill the given process.
l <pid>               Lower the priority of a process to the lowest possible.
lock                  Lock access to Command Prompt channels.
m <pid> <MB-allow>    Limit the memory usage of a process to <MB-allow>.
p                     Toggle paging the display.
r <pid>               Raise the priority of a process by one.
s                     Display the current time and date (24 hour clock used).
s mm/dd/yyyy hh:mm    Set the current time and date (24 hour clock used).
t                     Tlist.
restart               Restart the system immediately.
shutdown              Shutdown the system immediately.
crashdump             Crash the system. You must have crash dump enabled.
```

3. 명령 프롬프트 채널(예: `cmd0001` 또는 `cmd0002`)을 생성하려면 cmd를 입력한 다음 Enter를 누릅니다.

4. 명령 프롬프트 채널을 보려면 ESC를 누른 다음 TAB을 누릅니다.

   예상 결과

```
Name:                 Cmd0001
Description:          Command
Type:                 VT-UTF8
Channel GUID:         ef9f20a0-1287-11eb-82b0-0e4ba51872e5
Application Type GUID: 63d02271-8aa4-11d5-bccf-00b0d014a2d0

Press <esc><tab> for next channel.
Press <esc><tab>0 to return to the SAC channel.
Use any other key to view this channel.
```

5. 채널을 전환하려면 ESC+TAB+채널 번호를 함께 누릅니다. 예를 들어, `cmd0002` 채널(생성된 경우)로 전환하려면 ESC+탭+2를 누릅니다.

6. 명령 프롬프트 채널에 필요한 자격 증명을 입력합니다.

```
Please enter login credentials.
Username: Administrator
Domain  : .
Password: *******
```

명령 프롬프트는 이미 출력된 문자를 읽을 수 없다는 점을 제외하고 데스크톱에서 사용할 수 있는 모든 기능을 갖춘 명령 셸입니다.

```
Microsoft Windows [Version 10.0.17763.1457]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>diskpart

Microsoft DiskPart version 10.0.17763.1

Copyright (C) Microsoft Corporation.
On computer: EC2AMAZ-ASR4SAI

DISKPART> list disk

  Disk ###  Status         Size     Free     Dyn  Gpt
  --------  -------------  -------  -------  ---  ---
  Disk 0    Online           30 GB      0 B
  Disk 1    Online           46 GB    46 GB

DISKPART>
```

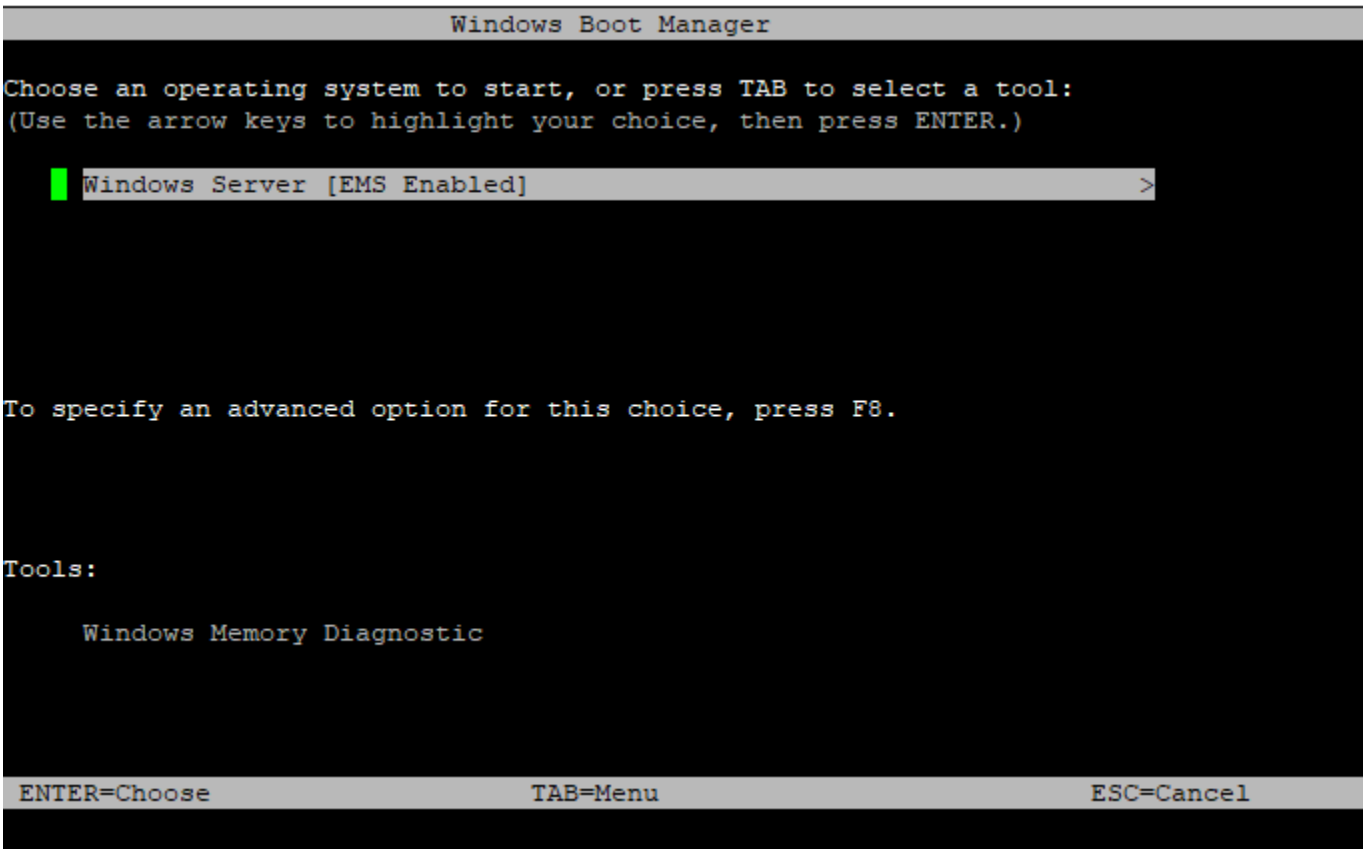
명령 프롬프트에서 PowerShell을 사용할 수도 있습니다.

참고로 진행률 기본 설정을 자동 모드로 설정해야 할 수 있습니다.

```
PS C:\Windows\system32> $ProgressPreference="SilentlyContinue"
PS C:\Windows\system32> $computerInfo = Get-ComputerInfo
PS C:\Windows\system32> $computerInfo.Csprocessors[0].Name
Intel(R) Xeon(R) Platinum 8124M CPU @ 3.00GHz
PS C:\Windows\system32> $computerInfo.Csprocessors[0].Description
Intel64 Family 6 Model 85 Stepping 4
PS C:\Windows\system32>
```

부팅 메뉴 사용

인스턴스에 부팅 메뉴가 활성화되어 있는 경우 SSH를 통해 연결한 후 다시 시작하면 다음과 같이 부팅 메뉴가 표시됩니다.



부팅 메뉴 명령

ENTER

선택한 운영 체제 항목을 시작합니다.

Tab

도구 메뉴로 전환합니다.

ESC

인스턴스를 취소하고 다시 시작합니다.

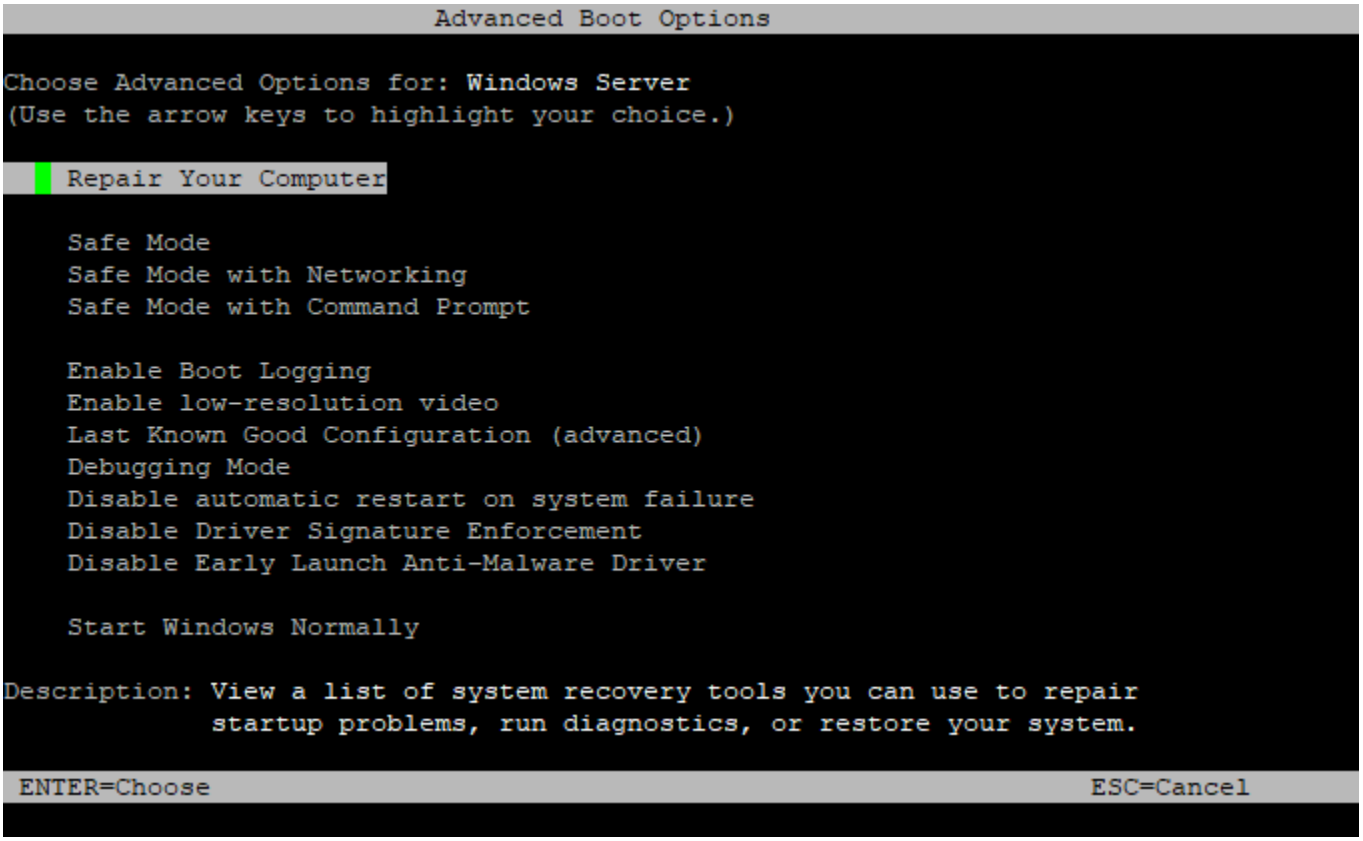
ESC 다음에 8

F8키를 누르는 것과 같습니다. 선택한 항목에 대한 고급 옵션을 표시합니다.

ESC 키+왼쪽 화살표

초기 부팅 메뉴로 돌아갑니다.

> **Note**
>
> ESC 키만 누르면 Windows가 이스케이프 시퀀스가 진행 중인지 확인하기 위해 기다리기 때문에 주 메뉴로 돌아가지 않습니다.



SAC 및 부팅 메뉴 비활성화

SAC 및 부팅 메뉴를 활성화하면 나중에 이러한 기능을 비활성화할 수 있습니다.

다음 방법 중 하나를 사용하여 인스턴스에서 SAC 및 부팅 메뉴를 비활성화합니다.

PowerShell

Windows 인스턴스에서 SAC 및 부팅 메뉴 활성화

1. 인스턴스에 연결하고 상승된 PowerShell 명령줄에서 다음 단계를 수행합니다.
2. 먼저 값을 no로 변경하여 부팅 메뉴를 비활성화합니다.

```
bcdedit /set '{bootmgr}' displaybootmenu no
```

3. 그런 다음 값을 off로 변경하여 SAC를 비활성화합니다.

```
bcdedit /ems '{current}' off
```

4. 인스턴스를 재부팅하여 업데이트된 구성을 적용합니다.

```
shutdown -r -t 0
```

Command prompt

Windows 인스턴스에서 SAC 및 부팅 메뉴 활성화

1. 인스턴스에 연결하고 명령 프롬프트에서 다음 단계를 수행합니다.
2. 먼저 값을 no로 변경하여 부팅 메뉴를 비활성화합니다.

```
bcdedit /set {bootmgr} displaybootmenu no
```

3. 그런 다음 값을 off로 변경하여 SAC를 비활성화합니다.

```
bcdedit /ems {current} off
```

4. 인스턴스를 재부팅하여 업데이트된 구성을 적용합니다.

```
shutdown -r -t 0
```

연결할 수 없는 Amazon EC2 인스턴스를 디버깅하기 위한 진단 인터럽트 보내기

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 응답하지 않는 인스턴스에 진단 인터럽트를 전송하여 Linux 인스턴스의 경우 커널 패닉을 수동으로 트리거하거나 Windows 인스턴스의 경우 중지 오류(일반적으로 블루 스크린 오류라고 함)를 트리거할 수 있습니다.

Linux 인스턴스

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.

Windows 인스턴스

일반적으로 중단 오류가 발생하면 Windows 운영 체제가 충돌하고 다시 부팅되지만 특정 동작은 해당 구성에 따라 다릅니다. 중단 오류가 발생하면 운영 체제는 커널 메모리 덤프와 같은 디버깅 정보를 파일에 쓸 수도 있습니다. 이 정보를 사용하여 근본 원인 분석을 수행하여 인스턴스를 디버그할 수 있습니다. 메모리 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
- 사전 조건
- 진단 인터럽트 보내기

지원되는 인스턴스 유형

진단 인터럽트는 AWS Graviton 프로세서에서 제공되는 경우를 제외한 모든 Nitro 기반 인스턴스 유형에서 지원됩니다. 자세한 내용은 AWS Nitro 시스템에 구축된 인스턴스 및 AWS Graviton을 참조하세요.

사전 조건

진단 인터럽트를 사용하기 전에 인스턴스의 운영 체제를 구성해야 합니다. 이렇게 하면 커널 패닉(Linux 인스턴스) 또는 중지 오류(Windows 인스턴스)가 발생할 때 필요한 작업을 수행할 수 있습니다.

Linux 인스턴스

커널 패닉이 발생할 때 크래시 덤프를 생성하도록 Amazon Linux 2 또는 Amazon Linux 2023을 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 구성하세요. 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 추가하세요: `crashkernel=`*`memory_to_reserve`* 예를 들어 256MB를 예약하려면 `grub` 파일을 다음과 같이 수정합니다.

```
GRUB_CMDLINE_LINUX_DEFAULT="crashkernel=256M 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 연결하세요.
8. 커널이 올바른 `crashkernel` 매개 변수로 부팅되었는지 확인하세요.

```
$ 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=256M 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 수정하세요.

SUSE Linux Enterprise, Ubuntu 또는 Red Hat Enterprise Linux를 구성하려면

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 운영 체제의 구성 파일을 조정하여 커널이 충돌하도록 구성해야 합니다. 커널이 충돌하도록 구성하는 방법에 대한 자세한 내용은 운영 체제에 대한 설명서를 참조하세요.

- SUSE Linux Enterprise
- Ubuntu
- Red Hat Enterprise Linux (RHEL)

Windows 인스턴스

중단 오류가 발생할 때 메모리 덤프를 생성하도록 Windows를 구성하려면

1. 인스턴스에 연결합니다.
2. 제어판을 열고 시스템, 고급 시스템 설정을 선택합니다.
3. 시스템 속성 대화 상자에서 권한 탭을 선택합니다.
4. 시작 및 복구 섹션에서 설정...을 선택합니다.
5. 시스템 오류 섹션에서 필요에 따라 설정을 구성한 다음 확인을 선택하세요.

Windows 중단 오류 구성에 대한 자세한 내용은 Windows의 메모리 덤프 파일 옵션 개요를 참조하세요.

진단 인터럽트 보내기

필요한 구성 변경을 완료하면 AWS CLI 또는 Amazon EC2 API를 사용하여 인스턴스에 진단 인터럽트를 보낼 수 있습니다.

AWS CLI

인스턴스에 진단 인터럽트를 보내려면

send-diagnostic-interrupt 명령을 사용합니다.

```
aws ec2 send-diagnostic-interrupt --instance-id i-1234567890abcdef0
```

PowerShell

인스턴스에 진단 인터럽트를 보내려면

Send-EC2DiagnosticInterrupt cmdlet을 사용합니다.

```
Send-EC2DiagnosticInterrupt -InstanceId i-1234567890abcdef0
```

Amazon EC2 사용 설명서에 대한 문서 기록

다음 표에서는 2019년부터 Amazon EC2 사용 설명서에 추가된 중요 사항에 대해 설명합니다. 사용자가 보낸 의견을 수렴하기 위해 설명서를 자주 업데이트합니다.

변경 사항	설명	날짜
하드웨어 패킷 타임스탬핑	EC2 인스턴스에서 하드웨어 패킷 타임스탬핑을 활성화하여 모든 수신 네트워크 패킷에 64비트 나노초 정밀도 타임스탬프를 추가할 수 있습니다.	2025년 6월 23일
AMI 등록 취소 중 스냅샷 삭제	AMI 등록을 취소할 때 연결된 스냅샷을 자동으로 동시에 삭제할 수 있습니다.	2025년 5월 28일
사용자 시작 재부팅 마이그레이션	인스턴스가 EC2 예약된 재부팅 이벤트 알림을 수신하는 경우 이벤트 전에 인스턴스를 재부팅하면 인스턴스 마이그레이션이 기본적으로 활성화됩니다.	2025년 5월 20일
Amazon EC2 Mac 인스턴스의 SIP 구성	x86 Mac 인스턴스와 Apple 실리콘 Mac 인스턴스의 시스템 무결성 보호(SIP) 설정을 구성할 수 있습니다.	2025년 5월 20일
루트 볼륨 소유권 위임	이제 Apple 실리콘 Mac 인스턴스에서 운영 체제를 업데이트할 때 Amazon EBS 루트 볼륨의 루트 볼륨 소유권 위임을 자동화할 수 있습니다.	2025년 5월 20일

P6-B200 인스턴스의 인스턴스 토폴로지	`p6-b200.48xlarge` 인스턴스에 대한 인스턴스 토폴로지 정보를 가져올 수 있습니다.	2025년 5월 15일
EC2 Fast Launch 서비스 연결 역할의 관리형 정책 업데이트	Amazon EventBridge 이벤트 규칙을 생성하고 유지하기 위한 액세스 권한과 EC2 Fast Launch 경험을 개선하기 위한 기타 작업으로 EC2 Fast Launch 서비스 연결 역할 정책을 업데이트했습니다. 서비스 연결 역할이 부여하는 권한에 대한 자세한 정보는 `EC2FastLaunchServiceRolePolicy` 에서 부여한 권한을 참조하세요.	2025년 5월 14일
EC2 Fast Launch 전체 액세스 정책의 관리형 정책 업데이트	시작 템플릿 작업을 생성 및 수정하는 작업 권한과 네트워크 및 보안 그룹 리소스를 생성하고, 수신 규칙을 승인하고, EC2 Fast Launch가 생성한 리소스를 삭제할 수 있는 권한이 있는 EC2 Fast Launch 전체 액세스 정책을 업데이트했습니다. 이 정책은 CloudFormation 스택을 추가로 설명 및 생성하고 EC2 Fast Launch가 생성한 스택을 삭제할 수 있습니다.	2025년 5월 14일
EC2 Fast Launch를 위한 간소화된 온보딩	온보딩 프로세스를 간소화하기 위해 EC2 Fast Launch를 업데이트했습니다.	2025년 5월 14일

허용된 AMI와 AWS Config 통합	ec2-instance-launched-with-allowed-ami AWS Config 규칙을 사용하여 실행 중이거나 중지된 인스턴스가 허용된 AMI 기준을 충족하는 AMI로 시작되었는지 확인합니다.	2025년 3월 11일
AWSEC2CapacityReservationFleetRolePolicy 관리형 정책이 업데이트됨	Amazon EC2는 `StringLike` 조건 연산자 대신 `ArnLike` 조건 연산자를 사용하도록 `AWSEC2CapacityReservationFleetRolePolicy` 관리형 정책을 업데이트했습니다.	2025년 3월 3일
시간 기반 AMI 복사	이제 EBS 지원 AMI 복사 작업에 대한 완료 기간을 요청하여 AMI 복사본이 특정 기간에 완료되도록 할 수 있습니다.	2025년 2월 25일
AmazonEC2ReadOnlyAccess 정책 업데이트	Amazon EC2는 기존 AmazonEC2ReadOnlyAccess 정책에 `GetSecurityGroupsForVpc` 작업을 추가했습니다.	2024년 12월 27일
관리형 인스턴스	이제 EC2 콘솔에서 Amazon EC2 관리형 인스턴스를 볼 수 있습니다.	2024년 12월 1일

선언적 정책	이제 선언적 정책을 사용하여 여러 리전과 계정에 계정 수준 설정을 동시에 적용할 수 있습니다. 선언적 정책은 EC2 직렬 콘솔 액세스, 허용된 AMI, IMDS 기본값을 구성하고 VPC, AMI 및 스냅샷에 대한 퍼블릭 액세스 설정을 차단하기 위해 지원됩니다. 자세한 내용은 AWS Organizations 사용 설명서의 선언적 정책 및 지원되는 각 기능에 대한 관련 설명서를 참조하세요.	2024년 12월 1일
허용된 AMI	이제 AMI가 충족해야 하는 기준을 지정하여 Amazon EC2에서 AMI 검색 및 사용을 제어할 수 있습니다.	2024년 12월 1일
인스턴트 용량 블록	이제 30분 안에 빠르게 시작할 수 있는 ML용 용량 블록을 예약할 수 있습니다.	2024년 11월 21일
장래 날짜 용량 예약	이제 장래 날짜를 위한 용량 예약을 요청할 수 있습니다.	2024년 11월 21일
용량 블록 연장	이제 기존 용량 블록의 기간을 연장할 수 있습니다.	2024년 11월 21일
6개월 용량 블록	이제 최대 6개월(182일) 동안 ML용 용량 블록을 예약할 수 있습니다.	2024년 11월 21일
저장된 필터 세트	이제 사용자 지정 필터 그룹을 생성하고 재사용하여 EC2 리소스를 효율적으로 볼 수 있습니다.	2024년 11월 20일

성능 보호	EC2 플릿 또는 스팟 플릿에 속성 기반 인스턴스 유형 선택을 사용하는 경우 이제 성능 보호를 활성화하여 선택한 인스턴스 유형이 지정된 성능 기준과 유사하거나 이를 초과하도록 할 수 있습니다.	2024년 11월 20일
용량 예약만	이제 인스턴스가 용량 예약에서만 실행되거나 용량 예약 리소스 그룹에서만 실행되도록 지정할 수 있습니다.	2024년 11월 20일
소스 AMI 식별	이제 AMI를 생성하는 데 사용된 소스 AMI를 식별할 수 있습니다.	2024년 11월 13일
용량 분할	기존 용량 예약에서 용량을 분할하여 새 용량 예약을 생성할 수 있습니다.	2024년 10월 30일
용량 이동	이제 한 용량 예약에서 다른 용량 예약으로 용량을 이동할 수 있습니다.	2024년 10월 30일
초보자 자습서	초보자를 위한 두 가지 새로운 자습서인 첫 번째 EC2 인스턴스 시작하기 및 테스트 EC2 인스턴스 시작 및 연결하기가 있습니다.	2024년 10월 21일
Windows Server 2025 지원	Windows Server 2025에 대한 지원 추가	2024년 10월 16일
EC2 Global View	이제 EC2 Global View를 통해 모든 리전의 계정에서 용량 예약 및 용량 블록을 확인할 수 있습니다.	2024년 10월 16일

라이브 마이그레이션 호스트 유지 관리	이제 Amazon EC2 전용 호스트는 지원되는 인스턴스를 중지하거나 다시 시작하지 않고도 손상된 전용 호스트에서 대체 전용 호스트로 자동으로 마이그레이션하는 라이브 마이그레이션 호스트 유지 관리를 지원합니다.	2024년 10월 15일
공유 용량 예약에 대한 결제 할당	이제 공유 용량 예약의 가용 용량에 대한 결제를 동일한 AWS 조직에 속해 있는 소비자 계정에 할당할 수 있습니다.	2024년 10월 14일
PTP 하드웨어 클럭 - 추가 리전 지원	PTP 하드웨어 클럭은 현재 미국 동부(오하이오) 및 아시아 태평양(말레이시아)에서도 사용할 수 있습니다.	2024년 9월 23일
EC2 Instance Connect는 IPv6를 지원합니다	이제 EC2 Instance Connect를 사용하여 인스턴스의 퍼블릭 IPv6 주소에 연결할 수 있습니다.	2024년 9월 23일
EC2 Instance Connect 접두사 목록	이제 보안 그룹에서 EC2 Instance Connect 서비스의 SSH 트래픽을 허용하는 규칙을 생성할 때 IPv4 또는 IPv6 주소의 관리형 접두사 목록을 선택할 수 있습니다.	2024년 9월 23일
온디맨드 용량 예약을 관리하는 새로운 기능	이제 용량 예약을 분할하고, 용량 예약 간에 용량을 이동하고, 용량 예약의 인스턴스 자격 속성을 수정할 수 있습니다.	2024년 8월 14일

C6g, C6gn, C6gd, C7g, C7gd, M6g, M6gd, M7g, M7gd, R6g, R6gd의 최대 절전 모드 지원	C6g, C6gn, C6gd, C7g, C7gd, M6g, M6gd, M7g, M7gd, R6g, R6gd 인스턴스 유형에서 실행되는 새로 시작된 인스턴스를 최대 절전 모드로 전환합니다.	2024년 7월 30일
Graviton 인스턴스 유형을 지원하는 AMI에 대한 최대 절전 모드 지원	Graviton 인스턴스 유형을 지원하는 Amazon Linux 또는 Ubuntu AMI에서 새로 시작된 인스턴스를 최대 절전 모드로 전환합니다.	2024년 7월 30일
Credential Guard에 지원되는 추가 인스턴스 유형	이제 C7i, C7-flex, M7i, M7i-flex, R7i, R7i-flex, T3 인스턴스에서 Credential Guard를 활성화할 수 있습니다.	2024년 6월 26일
EC2 M1 Ultra Mac 인스턴스	Apple M1 Ultra 프로세서가 특징인 새로운 범용 인스턴스 유형입니다.	2024년 6월 17일
EC2 인스턴스 유형 찾기 - 추가 파라미터	이제 EC2 인스턴스 유형 찾기에서 워크로드에 대한 보다 자세한 요구 사항을 지정할 수 있는 추가 파라미터를 제공합니다.	2024년 6월 5일
U7i-12tb, U7in-16tb, U7in-24tb 및 U7in-32tb 인스턴스	4세대 인텔 제온 스케일러블 프로세서를 탑재한 새로운 고용량 메모리 인스턴스 유형입니다.	2024년 5월 28일
EC2 Fast Launch를 위한 새로운 관리형 정책	인스턴스에서 EC2 Fast Launch 기능과 관련된 API 작업을 수행하는 EC2FastLaunchFullAccess 정책을 추가했습니다.	2024년 5월 14일

AMI 등록 취소 보호	AMI에서 등록 취소 보호 기능을 켜서 우발적이거나 악의적인 삭제를 방지할 수 있습니다.	2024년 4월 23일
PTP 하드웨어 클럭 - 인스턴스 유형 지원	PTP 하드웨어 클럭은 이제 C7a, C7i, M7a, M7g, M7i, R7a, R7i 인스턴스 유형에서 사용할 수 있습니다.	2024년 4월 22일
향상된 네트워킹을 위해 추가된 Nitro 성능 고려 사항	이 페이지에서는 Nitro 기반 Amazon EC2 인스턴스의 성능 튜닝에 도움이 되는 네트워크 고려 사항을 중점적으로 다룹니다.	2024년 4월 4일
VSS 기반 EBS 스냅샷용 새로운 관리형 정책	Amazon EC2 VSS에는 권한을 최신 상태로 유지하고 모범 사례를 따르도록 인스턴스 프로파일 역할에 추가할 수 있는 새로운 IAM 관리형 정책이 있습니다.	2024년 3월 28일
PTP 하드웨어 클럭- 미국 동부(버지니아 북부)	PTP 하드웨어 클럭은 현재 미국 동부(버지니아 북부) 리전에서 사용할 수 있습니다.	2024년 3월 26일
IMDSv2를 계정 기본값으로 설정	기본적으로 인스턴스 메타데이터 서비스 버전 2(IMDSv2)를 사용하도록 계정에서 모든 새로운 EC2 인스턴스 시작을 설정할 수 있습니다.	2024년 3월 25일
스냅샷에서 생성된 새 Linux AMI에 태그 지정	스냅샷에서 Linux AMI를 생성하면 새 AMI에 태그를 지정할 수 있습니다.	2024년 3월 7일

복사할 때 새 AMI 및 스냅샷에 태그 지정	AMI를 복사할 때 동일한 태그 또는 다른 태그를 사용하여 새 AMI와 새 스냅샷에 태그를 지정할 수 있습니다.	2024년 3월 7일
AWS 관리 팩 제거 페이지	AWS 관리 팩은 주로 Windows Server 2012 이전 버전에서 사용되었습니다. 이러한 레거시 OS 플랫폼 버전은 더 이상 지원되지 않습니다. AWS 및 온프레미스에서 실행되는 서버 플릿을 관리하고 문제를 해결하려면 AWS Systems Manager Fleet Manager를 참조하세요.	2024년 2월 12일
macOS AMI에 사전 설치된 EC2 Instance Connect	EC2 Instance Connect는 이제 macOS Sonoma 14.2.1 이상, macOS Ventura 13.6.3 이상 및 macOS Monterey 12.7.2 이상 AMI에 사전 설치되어 제공됩니다.	2024년 1월 26일
CentOS, macOS 및 RHEL에 대한 EC2 Instance Connect 지원	이제 EC2 Instance Connect를 지원되는 CentOS, macOS 및 RHEL AMI에 설치할 수 있습니다.	2023년 12월 6일
C7a, C7i, R7a, R7i 및 R7iz에 대한 최대 절전 모드 지원	C7a, C7i, R7a, R7i 및 R7iz 인스턴스 유형에서 실행되는 새로 시작된 인스턴스를 최대 절전 모드로 전환합니다.	2023년 12월 1일

Amazon Q EC2 인스턴스 유형 선택기	Amazon Q EC2 인스턴스 유형 선택기는 사용 사례, 워크로드 유형, CPU 제조업체 선호도는 물론 가격과 성능의 우선 순위 지정 방법을 고려합니다. 그런 다음 이 데이터를 사용하여 새 워크로드에 가장 적합한 Amazon EC2 인스턴스 유형에 대한 지침과 제안 사항을 제공합니다.	2023년 11월 28일
EC2 프리 티어	EC2 대시보드에서 EC2 프리 티어 사용량을 추적할 수 있습니다.	2023년 11월 26일
콘솔 투 코드	콘솔 투 코드는 자동화 코드를 시작하는 데 도움이 됩니다. 콘솔 투 코드는 콘솔 작업을 기록한 다음 생성형 AI를 사용하여 선호하는 코드형 인프라 형식으로 코드를 제안합니다. 코드를 시작점으로 사용하여 특정 사용 사례에 맞게 프로덕션에 바로 사용할 수 있도록 사용자 지정할 수 있습니다.	2023년 11월 26일
구성 가능한 유휴 연결 추적 제한 시간	유휴 상태로 유지되는 보안 그룹 연결은 연결 추적 소진으로 이어져 연결이 추적되지 않고 패킷이 삭제될 수 있습니다. 이제 탄력적 네트워크 인터페이스에서 보안 그룹 연결 추적에 대한 제한 시간(초)을 설정할 수 있습니다.	2023년 11월 17일

PTP 하드웨어 클럭	이제 지원되는 인스턴스에 PTP(Precision Time Protocol) 하드웨어 클럭이 있습니다. PTP 하드웨어 클럭은 NTP 또는 직접 PTP 연결을 지원합니다.	2023년 11월 16일
최대 절전 모드가 활성화된 인스턴스의 인스턴스 유형 변경	이제 `stopped` 상태에 있을 때 최대 절전 모드가 활성화된 인스턴스의 인스턴스 유형을 변경할 수 있습니다.	2023년 11월 16일
인스턴스 토폴로지	DescribeInstanceTopology API를 사용하여 인스턴스의 위치를 감지한 다음 이 정보로 물리적으로 서로 더 가까운 인스턴스에서 HPC 및 ML 작업을 실행하여 HPC 및 ML 작업을 최적화할 수 있습니다.	2023년 11월 13일
EC2 Fast Launch 공유 AMI 지원	이제 공유된 AMI에서 EC2 Fast Launch를 활성화할 수 있습니다. 공유 AMI에서 EC2 Fast Launch를 활성화하면 더 빠른 실행을 위해 사전 프로비저닝된 스냅샷이 계정에 생성됩니다.	2023년 11월 6일
ML용 용량 블록	이제는 미래 날짜의 GPU 인스턴스를 예약하여 단기간의 기계 학습(ML) 워크로드를 지원할 수 있습니다.	2023년 10월 31일

스팟 인스턴스 최대 절전 모드	이제는 현재 온디맨드 인스턴스에 사용할 수 있는 최대 절전 모드 환경과 인스턴스 패밀리를 동일하게 사용하여 스팟 인스턴스를 최대 절전 모드로 전환할 수 있습니다.	2023년 10월 24일
AMI에 대한 퍼블릭 액세스 차단 기본 설정	이제는 모든 신규 계정과 퍼블릭 AMI가 없는 기존 계정의 AMI에 대한 퍼블릭 액세스 차단이 기본적으로 활성화됩니다.	2023년 10월 20일
Amazon EC2 Global View	Amazon EC2 Global View는 추가 리소스 유형과 사용자 지정 가능한 디스플레이 옵션을 지원합니다.	2023년 10월 18일
Ubuntu 22.04.2 LTS(Jammy Jellyfish)에 대한 최대 절전 모드 지원	Ubuntu 22.04.2 LTS(Jammy Jellyfish) AMI에서 시작된 새로 시작된 인스턴스를 최대 절전 모드로 전환합니다.	2023년 10월 16일
AMI 비활성화	인스턴시 시작에 사용되지 않도록 AMI를 비활성화할 수 있습니다.	2023년 10월 12일
연결된 EBS 상태 확인	연결된 EBS 상태 확인을 사용하여 인스턴스에 연결된 Amazon EBS 볼륨에 연결할 수 있는지 모니터링할 수 있습니다.	2023년 10월 11일
Red Hat Enterprise Linux 9에 대한 최대 절전 모드 지원	Red Hat Enterprise Linux 9 AMI에서 시작된 새로 시작된 인스턴스를 최대 절전 모드로 전환합니다.	2023년 10월 2일

Microsoft Windows Server 2022에 대한 최대 절전 모드 지원	Microsoft Windows Server 2022 AMI에서 시작된 새로 시작된 인스턴스를 최대 절전 모드로 전환합니다.	2023년 10월 2일
AL2023에 대한 최대 절전 모드 지원	AL2023 AMI에서 새로 시작된 인스턴스를 최대 절전 모드로 전환합니다.	2023년 10월 2일
스팟 플릿에서 스팟 인스턴스 중단 시작	Amazon EC2 콘솔에서 스팟 플릿을 선택하고 플릿에서 스팟 인스턴스 중단을 시작하여 스팟 인스턴스의 애플리케이션이 중단을 어떻게 처리하는지 테스트할 수 있습니다.	2023년 9월 21일
AMI에 대한 퍼블릭 액세스 차단	계정 수준에서 AMI에 대한 공개 액세스 차단을 활성화하여 AMI를 공개하려는 모든 시도를 차단할 수 있습니다.	2023년 9월 12일
M7i 및 M7i-flex 최대 절전 모드 지원	M7i 및 M7i-flex 인스턴스 유형에서 실행되는 새로 시작된 인스턴스를 최대 절전 모드로 전환합니다.	2023년 8월 22일
EC2-Classic 더 이상 사용되지 않음	EC2-Classic을 사용하여 EC2 인스턴스는 다른 고객과 공유되는 단일 플랫 네트워크에서 실행되었습니다. Amazon VPC가 EC2-Classic을 대체합니다. Amazon VPC 사용을 통해 AWS 계정에 속하도록 논리적으로 독립된 Virtual Private Cloud(VPC)에서 인스턴스가 실행됩니다.	2023년 8월 8일

전용 호스트	Outpost의 특정 하드웨어 자산에 전용 호스트를 할당할 수 있습니다.	2023년 6월 20일
EC2 Instance Connect 엔드포인트	이제 인스턴스에 퍼블릭 IPv4 주소가 없어도 SSH 또는 RDP를 통해 인스턴스에 연결할 수 있습니다.	2023년 6월 13일
IMDS 패키지 분석기	이제 IMDS 패킷 분석기를 사용하여 EC2 인스턴스에서 IMDSv1 호출 소스를 식별할 수 있습니다.	2023년 6월 1일
EC2 직렬 콘솔 베어 메탈 인스턴스	이제 EC2 직렬 콘솔은 선택한 베어메탈 인스턴스의 직렬 포트 연결을 지원합니다.	2023년 4월 11일
시작 템플릿 할당량	이제 Service Quotas 콘솔과 Service Quotas CLI를 통해 시작 템플릿 및 시작 템플릿 버전의 할당량을 볼 수 있습니다.	2023년 4월 3일
용량 예약 사용률 알림	이제 계정에서 용량 예약의 용량 사용률이 20% 미만으로 떨어지면 AWS Health가 다음 알림을 보냅니다.	2023년 4월 3일
용량 예약 그룹	이제 내게 공유된 용량 예약을 내가 소유하고 있는 용량 예약 그룹에 추가할 수 있습니다.	2023년 3월 30일
인스턴스 메타데이터 옵션 수정	이제 Amazon EC2 콘솔을 사용하여 인스턴스 메타데이터 옵션을 수정할 수 있습니다.	2023년 3월 20일

현재 위치 macOS 운영 체제 업데이트	이제 M1 Mac 인스턴스에서 현재 위치 Apple macOS 운영 체제 업데이트를 수행할 수 있습니다.	2023년 3월 14일
UEFI 기본	이제 통합 확장 가능 펌웨어 인터페이스(UEFI)와 레거시 BIOS 부팅 모드를 모두 지원하는 단일 AMI를 생성할 수 있습니다.	2023년 3월 3일
IMDSv2를 위해 AMI 수정	AMI에서 실행되는 인스턴스에 기본적으로 IMDSv2가 필요하도록 기존 AMI를 수정합니다.	2023년 2월 28일
Windows 가상화 기반 보안 - Credential Guard	지원되는 Amazon EC2 인스턴스에서 가상화 기반 보안(VBS) 기능인 Credential Guard를 활성화할 수 있습니다.	2023년 1월 31일
시작 템플릿의 AMI 별칭	시작 템플릿에서 AMI ID 대신 AWS Systems Manager 파라미터를 지정하면 AMI ID가 변경될 때마다 템플릿을 업데이트하지 않아도 됩니다.	2023년 1월 19일
C6i, I3en 및 M6i 최대 절전 모드 지원	C6i, I3en 및 M6i 인스턴스 유형에서 실행되는 새로 시작된 인스턴스를 최대 절전 모드로 전환합니다.	2022년 12월 19일
찢긴 쓰기 방지	블록 스토리지 기능인 찢긴 쓰기 방지를 사용하면 데이터 복원력에 부정적인 영향을 미치지 않으면서 I/O 집약적인 관계형 데이터베이스 워크로드의 성능을 개선하고 지연 시간을 줄일 수 있습니다.	2022년 11월 29일

ENA Express	ENA Express를 사용하면 EC2 인스턴스 간 네트워크 트래픽의 스루풋을 늘리고 테일 지연 시간을 최소화할 수 있습니다.	2022년 11월 28일
AMI 태그 복사	AMI를 복사할 때 사용자 정의 AMI 태그도 같이 복사할 수 있습니다.	2022년 11월 18일
저장 및 복원을 위한 AMI 크기	Amazon S3 버킷에 저장하고 Amazon S3 버킷에서 복원할 수 있는 AMI의 크기(압축 전)가 이제 최대 5,000GB로 늘어났습니다.	2022년 11월 16일
스팟 인스턴스의 priceCapacityOptimized 할당 전략	`priceCapacityOptimized` 할당 전략을 사용하는 스팟 플릿은 가격과 용량을 모두 고려하여 중단될 가능성이 가장 낮으면서 가격이 가장 낮은 스팟 인스턴스 풀을 선택합니다.	2022년 11월 10일
스팟 인스턴스의 price-capacity-optimized 할당 전략	`price-capacity-optimized` 할당 전략을 사용하는 EC2 플릿은 가격과 용량을 모두 고려하여 중단될 가능성이 가장 낮으면서 가격이 가장 낮은 스팟 인스턴스 풀을 선택합니다.	2022년 11월 10일
계정과 AMI 공유 취소	AMI가 AWS 계정과 공유된 경우 더 이상 계정과 이를 공유하지 않으려면 AMI의 시작 권한에서 해당 계정을 제거합니다.	2022년 11월 4일

탄력적 IP 주소 전송	이제 하나의 AWS 계정에서 다른 계정으로 탄력적 IP 주소를 전송할 수 있습니다.	2022년 10월 31일
루트 볼륨 바꾸기	AMI를 사용하여 실행 중인 인스턴스의 루트 Amazon EBS 볼륨을 바꿀 수 있습니다.	2022년 10월 27일
인스턴스를 데이터베이스에 자동으로 연결	자동 연결 기능을 사용해 하나 이상의 EC2 인스턴스를 RDS 데이터베이스에 빠르게 연결하여 이들 간의 트래픽을 허용합니다.	2022년 10월 10일
AMI 할당량	이제 할당량이 AMI 생성 및 공유에 적용됩니다.	2022년 10월 10일
IMDSv2를 위해 AMI 구성	AMI에서 시작된 인스턴스에서 기본적으로 IMDSv2를 사용하도록 AMI를 구성합니다.	2022년 10월 3일
스팟 인스턴스 중단 시작	Amazon EC2 콘솔에서 스팟 인스턴스를 선택하고 중단을 시작하여 스팟 인스턴스의 애플리케이션이 중단되는 것을 어떻게 처리하는지 테스트할 수 있습니다.	2022년 9월 26일
검증된 AMI 공급자	Amazon EC2 콘솔에서는 Amazon이 소유한 퍼블릭 AMI 또는 검증된 Amazon 파트너가 확인된 공급 업체(Verified provider)로 표시됩니다.	2022년 7월 22일
AWS Outposts의 배치 그룹	Outpost에 배치 그룹을 위한 호스트 분산 전략을 추가했습니다.	2022년 6월 30일

AWS Outposts의 전용 호스트	AWS Outposts에 전용 호스트를 할당할 수 있습니다.	2022년 5월 31일
인스턴스 중지 방지	인스턴스의 우발적 중지를 방지하기 위해 해당 인스턴스에 대한 중지 방지를 사용 설정할 수 있습니다.	2022년 5월 24일
UEFI 보안 부팅	UEFI 보안 부팅은 Amazon EC2의 장기 보안 부팅 프로세스를 기반으로 구축되며, 재부팅 시 지속적인 위협으로부터 소프트웨어를 보호할 수 있는 추가적인 심층 방어 기능을 제공합니다.	2022년 5월 10일
NitroTPM	NitroTPM(Nitro Trusted Platform Module)은 AWS Nitro System에서 제공하는 가상 디바이스로서 TPM 2.0 사양을 준수합니다.	2022년 5월 10일
AMI 상태 변경 이벤트	이제 AMI가 상태를 변경하면 Amazon EC2가 이벤트를 생성합니다. Amazon EventBridge를 사용하여 이러한 이벤트를 감지하고 대응할 수 있습니다.	2022년 5월 9일
퍼블릭 키 설명	Amazon EC2 키 페어의 퍼블릭 키 및 생성 날짜를 쿼리할 수 있습니다.	2022년 4월 28일
키 페어 생성	새 키 페어를 생성하는 경우 키 형식(PEM 또는 PPK)을 지정할 수 있습니다.	2022년 4월 28일

시작 시 Amazon FSx 파일 시스템 탑재	새 인스턴스 시작 마법사를 사용하여 시작 시 신규 또는 기존 Amazon FSx for NetApp ONTAP 또는 Amazon FSx for OpenZFS 파일 시스템을 탑재할 수 있습니다.	2022년 4월 12일
새 인스턴스 시작 마법사	EC2 인스턴스를 더 빠르고 더 쉽게 시작할 수 있는 방법이 제공되는 Amazon EC2 콘솔의 새롭게 개선된 시작 환경입니다.	2022년 4월 5일
자동으로 퍼블릭 AMI 사용 중단	기본적으로 모든 퍼블릭 AMI의 사용 중단 날짜가 AMI 생성 날짜로부터 2년으로 설정됩니다.	2022년 3월 31일
인스턴스 메타데이터 카테고리: 자동 크기 조정/대상 수명 주기 상태	Auto Scaling 그룹을 사용할 때, 인스턴스 메타데이터에서 인스턴스의 대상 수명 주기 상태에 액세스할 수 있습니다.	2022년 3월 24일
AMI 마지막 시작 시간	`lastLaunchedTime` 은 AMI가 인스턴스를 시작하는 데 마지막으로 사용된 시기를 나타냅니다.	2022년 2월 28일
ED25519 키	이제 EC2 Instance Connect 및 EC2 직렬 콘솔에 대해 ED25519 키가 지원됩니다.	2022년 1월 20일
용량 예약을 위한 추가 RHEL 플랫폼	온디맨드 용량 예약을 위한 추가 Red Hat Enterprise Linux 플랫폼입니다.	2022년 1월 11일
더 빠른 시작을 위해 Windows AMI 구성	사전 프로비저닝된 스냅샷을 사용하여 인스턴스를 최대 65% 더 빠르게 시작하도록 Windows AMI를 구성합니다.	2022년 1월 10일

인스턴스 메타데이터의 인스턴스 태그	인스턴스 메타데이터에서 인스턴스의 태그에 액세스할 수 있습니다.	2022년 1월 6일
클러스터 배치 그룹의 용량 예약	클러스터 배치 그룹에 용량 예약을 생성할 수 있습니다.	2022년 1월 6일
스팟 플릿 출시 후 종료	스팟 플릿은 새로운 대체 스팟 인스턴스가 시작된 후 재조정 알림을 받는 스팟 인스턴스를 종료할 수 있습니다.	2021년 11월 4일
EC2 플릿 출시 후 종료	EC2 플릿은 새로운 대체 스팟 인스턴스가 시작된 후 재조정 알림을 받는 스팟 인스턴스를 종료할 수 있습니다.	2021년 11월 4일
타임스탬프 비교	Amazon EC2 Linux 인스턴스의 타임스탬프를 ClockBound와 비교하여 이벤트의 실제 시간을 확인할 수 있습니다.	2021년 11월 2일
조직 또는 OU와 AMI 공유	이제 AWS 리소스(조직 및 조직 단위(OU))와 AMI를 공유할 수 있습니다.	2021년 10월 29일
스팟 배치 점수	스팟 용량 요구 사항에 따라 AWS 리전 또는 가용 영역을 추천 받습니다.	2021년 10월 27일
스팟 플릿에 대한 속성 기반 인스턴스 유형 선택	인스턴스에 있어야 하는 속성을 지정하면 Amazon EC2는 해당 속성으로 모든 인스턴스 유형을 식별합니다.	2021년 10월 27일

EC2 플릿에 대한 속성 기반 인스턴스 유형 선택	인스턴스에 있어야 하는 속성을 지정하면 Amazon EC2는 해당 속성으로 모든 인스턴스 유형을 식별합니다.	2021년 10월 27일
온디맨드 용량 예약 플릿	용량 예약 플릿을 사용하여 용량 예약의 그룹 또는 플릿을 시작할 수 있습니다.	2021년 10월 5일
Ubuntu 20.04 LTS - Focal의 최대 절전 모드 지원	Ubuntu 20.04 LTS - Focal AMI에서 시작된 새로 시작된 인스턴스를 최대 절전 모드로 전환합니다.	2021년 10월 4일
EC2 플릿 및 대상 온디맨드 용량 예약	EC2 플릿은 온디맨드 인스턴스를 `targeted` 용량 예약으로 시작할 수 있습니다.	2021년 9월 22일
전용 호스트의 T3 인스턴스	Amazon EC2 전용 호스트에서 T3 인스턴스 지원	2021년 9월 14일
RHEL, Fedora, CentOS에 대한 최대 절전 모드 지원	RHEL, Fedora, CentOS AMI에서 새로 시작된 인스턴스를 최대 절전 모드로 전환합니다.	2021년 9월 2일
Amazon EC2 Global View	Amazon EC2 Global View를 사용하면 단일 콘솔의 여러 AWS 리전에 걸쳐 VPC, 서브넷, 인스턴스, 보안 그룹 및 볼륨을 볼 수 있습니다.	2021년 9월 1일
C5d, M5d 및 R5d에 대한 최대 절전	M5a 및 R5a 인스턴스 유형에서 실행되는 새로 시작된 인스턴스를 최대 절전 모드로 전환합니다.	2021년 8월 19일

Amazon EC2 키 페어	Amazon EC2 이제 Linux 및 Mac 인스턴스에서 ED25519 키를 지원합니다.	2021년 8월 17일
네트워크 인터페이스 접두어	프라이빗 IPv4 또는 IPv6 CIDR 범위를 자동 또는 수동으로 네트워크 인터페이스에 할당할 수 있습니다.	2021년 7월 22일
이벤트 기간	Amazon EC2 인스턴스를 재부팅, 중지 또는 종료하는 예약된 이벤트에 대해 매주 반복되는 사용자 지정 이벤트 기간을 정의할 수 있습니다.	2021년 7월 15일
보안 그룹 규칙에 대한 리소스 ID 및 태깅 지원	리소스 ID별로 보안 그룹 규칙을 참조할 수 있습니다. 보안 그룹 규칙에 태그를 추가할 수도 있습니다.	2021년 7월 7일
AMI 사용 중지	이제 AMI가 사용 중지되는 시점을 지정할 수 있습니다.	2021년 6월 11일
Windows 초당 결제	Amazon EC2는 Windows 및 SQL Server 기반 사용량에 대해 초 단위로 요금을 청구하며 최소 1분 요금이 부과됩니다.	2021년 6월 10일
AWS Outposts의 용량 예약	이제 AWS Outposts에서 용량 예약을 사용할 수 있습니다.	2021년 5월 24일
용량 예약 공유	이제 Local Zones 및 Wavelength Zones에 생성된 용량 예약을 공유할 수 있습니다.	2021년 5월 24일

루트 볼륨 교체	이제 루트 볼륨 교체 작업을 사용하여, 실행 중인 인스턴스의 루트 EBS 볼륨을 교체할 수 있습니다.	2021년 4월 22일
S3를 사용하여 AMI 저장 및 복원	EBS-backed AMI를 S3에 저장하고 S3에서 복원하여 파티션 간 AMI 복사를 지원할 수 있습니다.	2021년 4월 6일
EC2 직렬 콘솔	인스턴스의 직렬 포트에 대한 연결을 설정하여 부팅 및 네트워크 연결 문제를 해결할 수 있습니다.	2021년 3월 30일
부팅 모드	Amazon EC2는 이제 선택한 AMD 및 인텔 기반 EC2 인스턴스에서 UEFI 부팅을 지원합니다.	2021년 3월 22일
역방향 DNS 레코드 생성	이제 탄력적 IP 주소에 대해 역방향 DNS 조회를 설정할 수 있습니다.	2021년 2월 3일
AMI 생성 시 AMI 및 스냅샷 태그 지정	AMI를 생성할 때 동일한 태그 또는 다른 태그를 사용하여 AMI와 스냅샷에 태그를 지정할 수 있습니다.	2020년 12월 4일
Amazon EventBridge를 사용하여 스팟 플릿 이벤트 모니터링	스팟 플릿 상태 변경 및 오류에 대한 응답으로 프로그래밍 작업을 트리거하는 EventBridge 규칙을 생성합니다.	2020년 11월 20일
EC2 플릿 이벤트 모니터링에 Amazon EventBridge 사용	EC2 집합 상태 변경 및 오류에 대한 응답으로 프로그래밍 작업을 트리거하는 EventBridge 규칙을 생성합니다.	2020년 11월 20일

instant 플릿 삭제	단일 API 호출로 `instant` 유형의 EC2 집합을 삭제하고 플릿의 모든 인스턴스를 종료합니다.	2020년 11월 18일
T3 및 T3a에 대한 최대 절전 모드 지원	T3 및 T3a 인스턴스 유형에서 실행되는 새로 시작된 인스턴스를 최대 절전 모드로 전환합니다.	2020년 11월 17일
Amazon EFS 빠른 생성	Amazon EFS Quick Create를 사용하여 시작 시 Amazon EFS 파일 시스템을 생성하고 인스턴스에 마운트할 수 있습니다.	2020년 11월 9일
인스턴스 메타데이터 카테고리: events/recommendations/rebalance	인스턴스에 대해 EC2 인스턴스 리밸런싱 권고 알림이 생성되는 대략적인 시간(UTC)입니다.	2020년 11월 4일
EC2 인스턴스 리밸런싱 권장 사항	스팟 인스턴스의 중단 위험이 높은 경우 이를 알려주는 신호입니다.	2020년 11월 4일
Wavelength 영역의 용량 예약	이제 Wavelength Zone에서 용량 예약을 생성하고 사용할 수 있습니다.	2020년 11월 4일
용량 재조정	Amazon EC2에서 리밸런싱 권고가 생성될 때 대체 스팟 인스턴스를 시작하도록 스팟 플릿 또는 EC2 플릿을 구성합니다.	2020년 11월 4일
I3, M5ad 및 R5ad에 대한 최대 절전 모드 지원	I3, M5ad 및 R5ad 인스턴스 유형에서 실행되는 새로 시작된 인스턴스를 최대 절전 모드로 전환합니다.	2020년 10월 21일

스팟 인스턴스 vCPU 제한	스팟 인스턴스 제한은 실행 중인 스팟 인스턴스가 사용하거나 미결 요청의 이행 보류 중에 사용할 vCPU 수를 기준으로 관리됩니다.	2020년 10월 1일
Local Zones의 용량 예약	이제 Local Zones에서 용량 예약을 생성하고 사용할 수 있습니다.	2020년 9월 30일
M5a 및 R5a에 대한 최대 절전 모드 지원	M5a 및 R5a 인스턴스 유형에서 실행되는 새로 시작된 인스턴스를 최대 절전 모드로 전환합니다.	2020년 8월 28일
인스턴스 메타데이터는 인스턴스 위치 및 배치 정보를 제공합니다.	`placement` 범주 아래의 새 인스턴스 메타데이터 필드: 리전, 배치 그룹 이름, 파티션 번호, 호스트 ID 및 가용 영역 ID.	2020년 8월 24일
용량 예약 그룹	AWS Resource Groups를 사용하여 용량 예약의 논리적 모음을 생성한 다음 해당 그룹으로 대상 인스턴스를 시작할 수 있습니다.	2020년 7월 29일
EC2Launch v2	인스턴스 시작 중일 경우, 인스턴스가 중지된 후 나중에 시작된 경우, 인스턴스가 재시작된 경우 및 온디맨드 방식으로 EC2Launch v2를 사용하여 태스크를 수행할 수 있습니다. EC2Launch v2는 모든 버전의 Windows Server를 지원하며 EC2Launch와 EC2Config를 대체합니다.	2020년 6월 30일

고유 IPv6 주소 가져오기	온프레미스 네트워크에서 AWS 계정으로 IPv6 주소 범위의 부분 또는 전체를 가져올 수 있습니다.	2020년 5월 21일
Systems Manager 파라미터를 사용하여 인스턴스 시작	인스턴스를 시작할 때 AMI 대신 AWS Systems Manager 파라미터를 지정할 수 있습니다.	2020년 5월 5일
예약된 이벤트 알림 사용자 지정	이메일 알림에 태그를 포함하도록 예약된 이벤트 알림을 사용자 지정할 수 있습니다.	2020년 5월 4일
전용 호스트의 Windows Server	Amazon에서 제공하는 Windows Server AMI를 사용하여 전용 호스트에서 최신 Windows Server 버전을 실행할 수 있습니다.	2020년 4월 7일
스팟 인스턴스 중지 및 시작	중지 중단 동작을 사용하는 대신 Amazon EBS 기반 스팟 인스턴스를 중지했다가 원할 때 시작합니다.	2020년 1월 13일
리소스에 태깅	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.	2020년 1월 10일
세션 관리자를 사용하여 인스턴스에 연결	Amazon EC2 콘솔에서 인스턴스로 세션 관리자 세션을 시작할 수 있습니다.	2019년 12월 18일

전용 호스트 및 호스트 리소스 그룹	이제 전용 호스트를 호스트 리소스 그룹과 함께 사용할 수 있습니다.	2019년 12월 2일
전용 호스트 공유	이제 AWS 계정 전체에서 전용 호스트를 공유할 수 있습니다.	2019년 12월 2일
계정 수준의 기본 크레딧 사양	AWS 리전별로 계정 수준에서 성능 순간 확장 가능 인스턴스 패밀리마다 기본 크레딧 사양을 설정할 수 있습니다.	2019년 11월 25일
인스턴스 유형 검색	필요에 맞는 인스턴스 유형을 찾을 수 있습니다.	2019년 11월 22일
전용 호스트	이제 한 인스턴스 패밀리의 여러 인스턴스 유형을 지원하도록 전용 호스트를 구성할 수 있습니다.	2019년 11월 21일
인스턴스 메타데이터 서비스 버전 2	인스턴스 메타데이터를 요청하는 세션 지향 방법인 인스턴스 메타데이터 서비스 버전 2를 사용할 수 있습니다	2019년 11월 19일
Elastic Fabric Adapter(EFA)	이제 Elastic Fabric Adapter를 Intel MPI 2019 Update 6에 사용할 수 있습니다.	2019년 11월 15일
온디맨드 Windows 인스턴스에 대한 최대 절전 모드 지원	온디맨드 Windows 인스턴스를 최대 절전 모드로 실행할 수 있습니다.	2019년 10월 14일
예약 인스턴스 대기열 구매	최대 3년 전에 예약 인스턴스 구매를 대기할 수 있습니다.	2019년 10월 4일

진단 인터럽트	연결할 수 없거나 응답이 없는 인스턴스에 진단 인터럽트를 보내 커널 패닉을 트리거할 수 있습니다.	2019년 8월 14일
용량 최적화 할당 전략	EC2 플릿 또는 스팟 플릿을 사용하여 시작 중인 인스턴스 수에 대한 용량이 최적화된 스팟 풀에서 스팟 인스턴스를 시작할 수 있습니다.	2019년 8월 12일
온디맨드 용량 예약 공유	이제 AWS 계정 전체에서 용량 예약을 공유할 수 있습니다.	2019년 7월 29일
Elastic Fabric Adapter(EFA)	EFA는 이제 Open MPI 3.1.4 및 Intel MPI 2019 Update 4를 지원합니다.	2019년 7월 26일
EC2 Instance Connect	EC2 Instance Connect는 Secure Shell(SSH)을 사용하여 인스턴스에 연결하는 간단하고 안전한 방법입니다.	2019년 6월 27일
호스트 복구	전용 호스트에서 예상치 못한 하드웨어 오류가 발생하는 경우 새 호스트에서 인스턴스를 자동으로 다시 시작합니다.	2019년 6월 5일
VSS 애플리케이션이 일치하는 스냅샷	AWS Systems Manager Run Command를 사용하여 Windows 인스턴스에 연결된 모든 Amazon EBS 볼륨의 애플리케이션 일치 스냅샷을 생성합니다.	2019년 5월 13일

Windows에서 Linux로 Microsoft SQL Server 데이터베이스를 위한 리플랫포밍 어시스턴트	기존 Microsoft SQL Server 워크로드를 Windows에서 Linux 운영 체제로 이전합니다. Amazon EC2 사용 설명서의 Microsoft SQL Server에 대한 링크 포인트가 업데이트되었습니다.	2019년 5월 8일
Windows 자동 업그레이드	AWS Systems Manager를 사용하여 EC2 Windows 인스턴스의 자동 업그레이드를 수행합니다.	2019년 5월 6일
Elastic Fabric Adapter(EFA)	인스턴스에 Elastic Fabric Adapter를 연결하여 HPC(고성능 컴퓨팅) 애플리케이션 속도를 높일 수 있습니다.	2019년 4월 29일

Amazon EC2의 인스턴스 유형 릴리스에 대한 자세한 내용은 Amazon EC2 인스턴스 유형 가이드의 문서 기록을 참조하세요.

2018년 및 그 이전 기록

다음 표에서는 2018년 및 그 이전 몇 년의 Amazon EC2 사용 설명서에 대한 중요 추가 사항을 설명합니다.

Feature	API 버전	설명	릴리스 날짜
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 섹션을 참조하세요.	2018년 12월 20일

Feature	API 버전	설명	릴리스 날짜
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 Amazon EC2 인스턴스를 최대 절전 모드로 전환 섹션을 참조하세요.	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.	2018년 11월 28일
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 플릿 생성 섹션을 참조하세요.	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 플릿 및 스팟 플릿 요청 유형 섹션을 참조하세요.	2018년 11월 14일
스팟 절감 정보	2016-11-15	단일 스팟 플릿 또는 모든 스팟 인스턴스의 스팟 인스턴스를 사용하여 얻은 절감액을 볼 수 있습니다. 자세한 내용은 스팟 인스턴스 구입으로 절감되는 비용 섹션을 참조하세요.	2018년 11월 5일

Feature	API 버전	설명	릴리스 날짜
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 Amazon EC2 인스턴스의 CPU 옵션 섹션을 참조하세요.	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 Amazon EC2 시작 템플릿 생성 섹션을 참조하세요.	2018년 10월 30일
On-Demand Capacity Reservations	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 EC2 온디맨드 용량 예약으로 컴퓨팅 용량 예약 섹션을 참조하세요.	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 AWS(으)로 가져오고 나면 이러한 주소가 계정에 주소 풀로 나타납니다. 주소 풀에서 탄력적 IP 주소를 생성하여 AWS 리소스에 사용할 수 있습니다. 자세한 내용은 Amazon EC2로 고유 IP 주소 가져오기(BYOIP) 섹션을 참조하세요.	2018년 10월 23일
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 계정에서 사용할 Amazon EC2 전용 호스트 할당 섹션을 참조하세요.	2018년 10월 08일

Feature	API 버전	설명	릴리스 날짜
스팟 플릿의 예약 크기 조정 작업에 대한 콘솔 지원	2016-11-15	날짜 및 시간을 기준으로 플릿의 현재 용량을 늘리거나 줄입니다. 자세한 내용은 예약된 규모 조정: 일정에 따라 스팟 플릿 규모 조정 섹션을 참조하세요.	2018년 9월 20일
EC2 집합를 위한 할당 전략	2016-11-15	온디맨드 용량을 가격(최저 가격 우선)을 기준으로 채울지 또는 우선순위(최우선순위 우선)를 기준으로 채울지 지정할 수 있습니다. 대상 스팟 용량을 할당할 스팟 풀 수를 지정할 수 있습니다. 자세한 내용은 할당 전략을 사용하여 EC2 플릿 또는 스팟 플릿이 스팟 및 온디맨드 용량을 충족하는 방법 결정 섹션을 참조하세요.	2018년 7월 26일
스팟 집합를 위한 할당 전략	2016-11-15	온디맨드 용량을 가격(최저 가격 우선)을 기준으로 채울지 또는 우선순위(최우선순위 우선)를 기준으로 채울지 지정할 수 있습니다. 대상 스팟 용량을 할당할 스팟 풀 수를 지정할 수 있습니다. 자세한 내용은 할당 전략을 사용하여 EC2 플릿 또는 스팟 플릿이 스팟 및 온디맨드 용량을 충족하는 방법 결정 섹션을 참조하세요.	2018년 7월 26일
스냅샷 수명 주기 자동화	2016-11-15	Amazon Data Lifecycle Manager를 사용하여 EBS 볼륨의 스냅샷 생성 및 삭제를 자동화할 수 있습니다. 자세한 내용은 Amazon Data Lifecycle Manager를 참조하세요.	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 Amazon EC2 시작 템플릿 생성 섹션을 참조하세요.	2018년 7월 11일
전용 호스트 태그 지정	2016-11-15	전용 호스트에 태그를 지정할 수 있습니다.	2018년 7월 3일

Feature	API 버전	설명	릴리스 날짜
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 Amazon EC2 인스턴스의 CPU 옵션 섹션을 참조하세요.	2018년 5월 8일
EC2 Fleet	2016-11-15	EC2 플릿을 사용하여 다양한 EC2 인스턴스 유형과 가용 영역, 온디맨드 인스턴스, 예약 인스턴스, 스팟 인스턴스 구매 모델에서 인스턴스 그룹을 시작할 수 있습니다. 자세한 내용은 EC2 플릿 및 스팟 플릿 섹션을 참조하세요.	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 EC2 플릿 및 스팟 플릿 섹션을 참조하세요.	2018년 5월 2일
생성 시 EBS 스냅샷 태그 지정	2016-11-15	생성 중 스냅샷에 태그를 적용할 수 있습니다.	2018년 4월 2일
배치 그룹 변경	2016-11-15	배치 그룹 내로 또는 밖으로 인스턴스를 이동하거나, 인스턴스의 배치 그룹을 변경할 수 있습니다. 자세한 내용은 EC2 인스턴스의 배치 변경 섹션을 참조하세요.	2018년 3월 1일
더 긴 리소스 ID	2016-11-15	더 많은 리소스 유형에 대해 더 긴 ID 형식을 활성화할 수 있습니다.	2018년 2월 9일
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.	2018년 1월 24일

Feature	API 버전	설명	릴리스 날짜
탄력적 IP 주소 태그 지정	2016-11-15	탄력적 IP 주소에 태그를 지정할 수 있습니다.	2017년 12월 21일
Amazon Time Sync Service	2016-11-15	Amazon Time Sync Service를 사용하여 인스턴스에서 정확한 시간을 유지할 수 있습니다. 자세한 내용은 EC2 인스턴스의 정밀 클럭 및 시간 동기화 섹션을 참조하세요.	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 섹션을 참조하세요.	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 Amazon EC2 시작 템플릿에 인스턴스 시작 파라미터 저장 섹션을 참조하세요.	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 섹션을 참조하세요.	2017년 11월 29일
스팟 인스턴스 최대 절전 모드	2016-11-15	스팟 서비스는 중단 시 스팟 인스턴스를 최대 절전 모드로 전환할 수 있습니다.	2017년 11월 28일
스팟 집합 대상 추적	2016-11-15	스팟 집합에 대해 대상 추적 조정 정책을 설정할 수 있습니다. 자세한 내용은 대상 추적 조정: 특정 지표에 대한 값을 대상으로 지정하여 스팟 플릿 규모 조정 섹션을 참조하세요.	2017년 11월 17일

Feature	API 버전	설명	릴리스 날짜
스팟 플릿과 Elastic Load Balancing 통합	2016-11-15	스팟 집합에 하나 이상의 로드 밸런서를 연결할 수 있습니다.	2017년 11월 10일
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 섹션을 참조하세요.	2017년 11월 6일
VPC 테넌시 수정	2016-11-15	VPC의 인스턴스 테넌시 속성은 `dedicated` 에서 `default`로 변경할 수 있습니다. 자세한 내용은 VPC의 인스턴스 테넌시 변경 섹션을 참조하세요.	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 스팟 인스턴스 중단 동작 섹션을 참조하세요.	2017년 9월 18일
NAT 게이트웨이에 태그 지정	2016-11-15	NAT 게이트웨이에 태그를 지정할 수 있습니다. 자세한 내용은 리소스에 태그 지정 섹션을 참조하세요.	2017년 9월 7일
보안 그룹 규칙 설명	2016-11-15	보안 그룹 규칙에 설명을 추가할 수 있습니다.	2017년 8월 31일
Elastic Graphics	2016-11-15	Elastic Graphics 액셀러레이터를 인스턴스에 연결하여 애플리케이션의 그래픽 성능을 높이세요.	2017년 8월 29일

Feature	API 버전	설명	릴리스 날짜
탄력적 IP 주소 복구	2016-11-15	VPC에서 사용하기 위해 탄력적 IP 주소를 해제한 경우 복구할 수 있습니다.	2017년 8월 11일
스팟 플릿 인스턴스 태깅	2016-11-15	스팟 플릿에서 시작되는 인스턴스를 자동으로 태깅하도록 스팟 플릿을 구성할 수 있습니다.	2017년 7월 24일
생성 시 태그 리소스	2016-11-15	생성 단계에서 인스턴스와 볼륨에 태그를 적용할 수 있습니다. 자세한 내용은 리소스에 태그 지정 섹션을 참조하세요. 또한 태그 기반 리소스 권한을 사용하여 적용되는 태그를 제어할 수도 있습니다. 자세한 내용은 생성 시 Amazon EC2 리소스 태그 지정에 대한 권한 부여 섹션을 참조하세요.	2017년 3월 28일
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.	2017년 2월 13일
IAM 역할 연결	2016-11-15	기존 인스턴스에 대한 IAM 역할을 연결, 분리하거나 교체할 수 있습니다. 자세한 내용은 Amazon EC2의 IAM 역할 섹션을 참조하세요.	2017년 2월 9일
전용 스팟 인스턴스	2016-11-15	Virtual Private Cloud(VPC)의 단일 테넌트 하드웨어에서 스팟 인스턴스를 실행할 수 있습니다. 자세한 내용은 단일 테넌트 하드웨어에서 시작 섹션을 참조하세요.	2017년 1월 19일
IPv6 지원	2016-11-15	IPv6 CIDR을 VPC와 서브넷에 연결하고 IPv6 주소를 VPC의 인스턴스에 할당할 수 있습니다. 자세한 내용은 Amazon EC2 인스턴스 IP 주소 지정 섹션을 참조하세요.	2016년 12월 1일

Feature	API 버전	설명	릴리스 날짜
스팟 플릿의 자동 크기 조정		이제 스팟 플릿에 대한 크기 조정 정책을 설정할 수 있습니다. 자세한 내용은 스팟 플릿의 자동 규모 조정 이해 섹션을 참조하세요.	2016년 9월 1일
ENA(Elastic Network Adapter)	2016-04-01	이제 ENA를 사용하여 네트워킹 수준을 높일 수 있습니다. 자세한 내용은 Amazon EC2 인스턴스의 향상된 네트워킹 섹션을 참조하세요.	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.	2016년 6월 23일
AWS 계정 간 암호화된 Amazon EBS 스냅샷 복사	2016-04-01	이제 AWS 계정 간에 암호화된 EBS 스냅샷을 복사할 수 있습니다.	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 섹션을 참조하세요.	2016년 5월 24일
두 가지 새로운 EBS 볼륨 유형	2015-10-01	이제 처리량에 최적화된 HDD(st1) 및 콜드 HDD(sc1) 볼륨을 생성할 수 있습니다.	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 섹션을 참조하세요.	2016년 3월 23일
스팟 플릿에 대한 CloudWatch 지표		이제 스팟 플릿에 대한 CloudWatch 지표를 확인할 수 있습니다. 자세한 내용은 CloudWatch를 사용하여 EC2 플릿 또는 스팟 플릿 모니터링 섹션을 참조하세요.	2016년 3월 21일

Feature	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.	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.	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.	2016년 1월 11일
전용 호스트	2015-10-01	Amazon EC2 전용 호스트는 고객 전용의 인스턴스 용량을 갖춘 물리적 서버입니다. 자세한 내용은 Amazon EC2 전용 호스트 섹션을 참조하세요.	2015년 11월 23일
스팟 인스턴스 지속 시간	2015-10-01	이제 스팟 인스턴스의 지속 시간을 지정할 수 있습니다. 스팟 블록은 지원되지 않습니다(2023년 1월).	2015년 10월 6일
스팟 플릿 수정 요청	2015-10-01	이제 스팟 플릿 요청의 목표 용량을 수정할 수 있습니다. 자세한 내용은 스팟 플릿 요청 수정 섹션을 참조하세요.	2015년 9월 29일
스팟 플릿 다각화 할당 전략	2015-04-15	단일 스팟 플릿 요청을 사용하여 여러 스팟 풀의 스팟 인스턴스를 할당할 수 있습니다. 자세한 내용은 할당 전략을 사용하여 EC2 플릿 또는 스팟 플릿이 스팟 및 온디맨드 용량을 충족하는 방법 결정 섹션을 참조하세요.	2015년 9월 15일

Feature	API 버전	설명	릴리스 날짜
스팟 플릿 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 내용은 인스턴스 가중치를 사용하여 EC2 플릿 또는 스팟 플릿의 비용 및 성능 관리 섹션을 참조하세요.	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 섹션을 참조하세요.	2015년 7월 23일
Spot Fleets	2015-04-15	별도의 스팟 인스턴스 요청을 관리하는 대신 스팟 인스턴스의 모음 또는 플릿을 관리할 수 있습니다. 자세한 내용은 EC2 플릿 및 스팟 플릿 섹션을 참조하세요.	2015년 5월 18일
탄력적 IP 주소의 EC2-Classic 마이그레이션	2015-04-15	EC2-Classic에서 사용하도록 할당한 탄력적 IP 주소를 VPC에서 사용하도록 마이그레이션할 수 있습니다.	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 참조하세요.	2015년 4월 23일
Systems Manager		Systems Manager를 통해 EC2 인스턴스를 구성하고 관리할 수 있습니다.	2015년 2월 17일
Microsoft SCVMM 1.5용 Systems Manager		이제 Microsoft SCVMM용 Systems Manager를 사용하여 인스턴스를 시작하고 SCVMM에서 Amazon EC2(으)로 VM을 가져올 수 있습니다.	2015년 1월 21일

Feature	API 버전	설명	릴리스 날짜
EC2 인스턴스용 자동 복구		Amazon EC2 인스턴스를 모니터링하고 기본 하드웨어 장애나 복구에 AWS 개입이 필요한 문제로 인해 인스턴스가 손상된 경우 인스턴스를 자동으로 복구하는 Amazon CloudWatch 경보를 생성할 수 있습니다. 복구된 인스턴스는 인스턴스 ID, IP 주소 및 모든 인스턴스 메타데이터를 포함하여 원본 인스턴스와 동일합니다. 자세한 내용은 자동 인스턴스 복구 섹션을 참조하세요.	2015년 1월 12일
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.	2015년 1월 7일
스팟 인스턴스 종료 공지		스팟 인스턴스 중단으로부터 보호하는 가장 좋은 방법은 내결함성을 갖추도록 애플리케이션을 설계하는 것입니다. 또한 Amazon EC2가 스팟 인스턴스를 종료하기 2분 전에 경고하는 스팟 인스턴스 종료 공지를 활용할 수 있습니다. 자세한 내용은 스팟 인스턴스 중단 공지 섹션을 참조하세요.	2015년 1월 5일
Systems Manager for Microsoft SCVMM		Systems Manager for Microsoft SCVMM은 EC2 인스턴스와 같은 AWS 리소스를 관리할 수 있는 간단하고 사용하기 쉬운 인터페이스를 제공합니다. 이 도구는 Microsoft SCVMM에서 제공합니다.	2014년 10월 29일

Feature	API 버전	설명	릴리스 날짜
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 참조하세요.	2014년 10월 23일
Amazon CloudWatch Logs에 대한 지원 추가		Amazon CloudWatch Logs을 사용하여 인스턴스나 다른 원본에서 시스템, 애플리케이션 및 사용자 지정 로그 파일을 모니터링, 저장 및 액세스할 수 있습니다. Amazon CloudWatch 콘솔, AWS CLI의 CloudWatch Logs 명령 또는 CloudWatch Logs SDK를 사용하여 CloudWatch Logs에서 관련 로그 데이터를 가져올 수 있습니다.	2014년 7월 10일
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 SSD 볼륨은 광범위한 작업에서 이상적으로 사용될 수 있는 비용 효과적인 스토리지를 제공합니다. 이러한 볼륨은 10밀리초 미만의 지연 시간, 연장된 기간 동안 3,000 IOPS로 버스트할 수 있는 기능 및 3 IOPS/GiB의 기본 성능을 제공합니다. 범용 SSD 볼륨 크기는 1GiB~1TiB입니다.	2014년 6월 16일
AWS 관리 팩		AWS이제 System Center Operations Manager 2012 R2에 대해 관리 팩이 지원됩니다.	2014년 5월 22일

Feature	API 버전	설명	릴리스 날짜
Amazon EBS 암호화	2014-05-01	Amazon EBS 암호화에서는 EBS 데이터 볼륨 및 스냅샷에 완벽한 암호를 제공하므로 보안 키 관리 인프라를 구축하고 유지 관리할 필요가 없습니다. EBS 암호화는 AWS 관리형 키를 사용하여 데이터를 암호화하여 상주 데이터에 대한 보안을 활성화합니다. EC2 인스턴스를 호스트하는 서버에서 암호화가 이루어지기 때문에 EC2 인스턴스와 EBS 스토리지 사이를 이동하는 데이터도 암호화됩니다.	2014년 5월 21일
Amazon EC2 사용 보고서		Amazon EC2 사용 보고서는 EC2 사용에 대한 비용 및 사용량 데이터를 보여 주는 보고서 세트입니다.	2014년 1월 28일
Linux 가상 머신 가져오기	2013-10-15	VM Import 프로세스에서 이제 Linux 인스턴스 가져오기를 지원합니다. 자세한 내용은 VM Import/Export 사용 설명서를 참조하세요.	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 EC2의 자격 증명 및 액세스 관리 단원을 참조하세요.	2013년 11월 20일
AWS Marketplace에서 인스턴스 시작		이제 Amazon EC2 Launch Wizard를 사용하여 AWS Marketplace에서 인스턴스를 시작할 수 있습니다. 자세한 내용은 AWS Marketplace AMI에서 Amazon EC2 인스턴스 시작 섹션을 참조하세요.	2013년 11월 11일
새로운 Launch Wizard		재설계된 새로운 EC2 Launch Wizard가 제공됩니다. 자세한 내용은 콘솔에서 인스턴스 시작 마법사를 사용하여 EC2 인스턴스 시작 섹션을 참조하세요.	2013년 10월 10일

Feature	API 버전	설명	릴리스 날짜
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 섹션을 참조하세요.	2013년 10월 9일
Amazon EC2 예약 인스턴스 수정	2013-08-15	이제 리전에서 예약 인스턴스를 수정할 수 있습니다. 자세한 내용은 예약 인스턴스 수정 섹션을 참조하세요.	2013년 9월 11일
퍼블릭 IP 주소 배정	2013-07-15	이제는 VPC에서 인스턴스를 시작할 때 퍼블릭 IP 주소를 배정할 수 있습니다. 자세한 내용은 시작 시 퍼블릭 IPv4 주소 할당 섹션을 참조하세요.	2013년 8월 20일
리소스 수준의 권한 부여	2013-06-15	Amazon EC2에서는 새로운 Amazon 리소스 이름(ARN)과 조건 키를 지원합니다. 자세한 내용은 Amazon EC2의 ID 기반 정책 섹션을 참조하세요.	2013년 7월 8일
증분형 스냅샷 사본	2013-02-01	이제 증분형 스냅샷 사본을 사용할 수 있습니다.	2013년 6월 11일
AWS 관리 팩		AWS Management Pack은 Amazon EC2 인스턴스와 해당 인스턴스에서 실행되는 Windows 또는 Linux 운영 체제를 연결합니다. AWS 관리 팩은 Microsoft System Center Operations Manager의 확장 기능입니다.	2013년 5월 8일
새 태그 페이지		Amazon EC2 콘솔에 새 태그 페이지가 있습니다. 자세한 내용은 Amazon EC2 리소스 태깅 섹션을 참조하세요.	2013년 4월 4일

Feature	API 버전	설명	릴리스 날짜
리전 간 AMI 복사	2013-02-01	리전 간에 AMI를 복사하여 둘 이상의 AWS 리전에서 일관된 인스턴스를 빠르고 쉽게 시작할 수 있습니다. 자세한 내용은 Amazon EC2 AMI 복사 섹션을 참조하세요.	2013년 3월 11일
인스턴스를 기본 VPC로 시작	2013-02-01	AWS 계정에서 리전별로 EC2-Classic 또는 VPC로, 또는 VPC로만 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
EBS 스냅샷 복사	2012-12-01	스냅샷 복사본으로 데이터 백업, 새 Amazon EBS 볼륨 또는 Amazon Machine Image(AMI)를 생성할 수 있습니다.	2012년 12월 17일
Provisioned IOPS SSD 볼륨에 대한 EBS 지표 및 상태 확인 업데이트	2012-10-01	Provisioned IOPS SSD 볼륨에 대한 새 지표 두 개를 포함하도록 EBS 지표가 업데이트되었습니다. 또한 Provisioned IOPS SSD 볼륨에 대한 새로운 상태 확인도 추가되었습니다.	2012년 11월 20일
스팟 인스턴스 요청 상태	2012-10-01	스팟 인스턴스 요청 상태를 사용하면 스팟 요청의 상태를 손쉽게 확인할 수 있습니다.	2012년 10월 14일

Feature	API 버전	설명	릴리스 날짜
Amazon EC2 예약 인스턴스 Marketplace	2012-08-15	예약 인스턴스 Marketplace는 더 이상 필요하지 않은 Amazon EC2 예약 인스턴스를 보유한 판매자와 추가 용량을 원하는 구매자를 연결해줍니다. 예약 인스턴스 Marketplace를 통해 구매 및 판매되는 예약 인스턴스는 다른 예약 인스턴스와 동일하게 작동합니다. 단, 이러한 인스턴스는 표준 기간보다 남은 기간이 짧을 수 있으며 다른 가격으로 판매될 수 있습니다.	2012년 9월 11일
Amazon EBS용 Provisioned IOPS SSD	2012-07-20	Provisioned IOPS SSD 볼륨은 일관되고 빠른 응답 시간을 이용하는 데이터베이스 애플리케이션처럼 I/O 집약적 작업을 위한 예측 가능하고 우수한 성능을 제공합니다.	2012년 7월 31일
Amazon EC2 인스턴스의 IAM 역할	2012-06-01	Amazon EC2의 IAM 역할은 다음을 제공합니다. • AWSAmazon EC2 인스턴스에서 실행 중인 애플리케이션의 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

Feature	API 버전	설명	릴리스 날짜
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 스팟 인스턴스 요청을 참조하세요. • 인스턴스가 시작되거나 종료될 때 알림을 받습니다. • AWS CloudFormation 템플릿을 사용하여 AWS 리소스가 포함된 스택에서 스팟 인스턴스를 시작합니다.	2012년 6월 7일
EC2 인스턴스 내보내기와 Amazon EC2 상태 확인을 위한 타임스탬프	2012-05-01	원래 EC2로 가져왔던 Windows Server 인스턴스에 대한 내보내기 지원을 추가했습니다. 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
AWS Marketplace AMI	2012-04-01	AWS Marketplace AMI에 대한 지원을 추가했습니다.	2012년 4월 19일

Feature	API 버전	설명	릴리스 날짜
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 섹션을 참조하세요.	2011년 12월 21일
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 Amazon EC2 인스턴스 상태 모니터링 섹션을 참조하세요.	2011년 11월 16일
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

Feature	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 설명서를 참조하세요.	2011년 8월 24일
VMware vCenter용 Amazon EC2 VM Import Connector 업데이트		VMware vCenter 가상 어플라이언스용 Amazon EC2 VM Import Connector 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
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 일부로 새 가용 영역 요금 옵션이 추가되었습니다. 이러한 추가를 통해 스팟 인스턴스를 특정 가용 영역으로 시작하는 데 필요한 가격을 보다 쉽게 확인할 수 있습니다.	2011년 5월 26일

Feature	API 버전	설명	릴리스 날짜
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 EC2의 자격 증명 및 액세스 관리 섹션을 참조하세요.	2011년 4월 26일
전용 인스턴스		Amazon Virtual Private Cloud(Amazon VPC) 내에서 시작되는 전용 인스턴스는 호스트 하드웨어 수준에서 물리적으로 구분되어 있는 인스턴스입니다. 전용 인스턴스에서는 탄력적인 온디맨드 프로비저닝을 포함한 다양한 혜택과 함께 Amazon VPC와 AWS 클라우드를 활용하고 사용하는 서비스에 대해서만 요금을 지불할 수 있으며, 하드웨어 수준에서 Amazon EC2 컴퓨팅 인스턴스를 구분할 수 있습니다. 자세한 내용은 Amazon EC2 전용 인스턴스 섹션을 참조하세요.	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.	2011년 3월 27일
메타데이터 정보	2011-01-01	2011년 1월 1일 릴리스의 변경 내용을 반영하여 메타데이터에 대한 정보를 추가했습니다. 자세한 내용은 인스턴스 메타데이터를 사용하여 EC2 인스턴스를 관리합니다. 및 인스턴스 메타데이터 카테고리 섹션을 참조하세요.	2011년 3월 11일

Feature	API 버전	설명	릴리스 날짜
VMware vCenter용 Amazon EC2 VM Import Connector		VMware vCenter 가상 어플라이언스용 Amazon EC2 VM Import Connector에 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 인스턴스에서 Amazon EBS 볼륨을 강제로 분리할 수 있습니다.	2011년 2월 23일
인스턴스 종료 방지		이제 AWS Management Console을 사용하여 인스턴스가 종료되는 것을 방지할 수 있습니다. 자세한 내용은 인스턴스 종료 방지 변경 섹션을 참조하세요.	2011년 2월 23일
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 설명서를 참조하세요.	2010년 12월 15일
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 Amazon EC2 리소스 찾기 및 Amazon EC2 리소스 태깅 섹션을 참조하세요.	2010년 9월 19일
멱등성 인스턴스 시작	2010-08-31	인스턴스 실행 시 멱등성 유지에 대한 정보를 추가했습니다.	2010년 9월 19일

Feature	API 버전	설명	릴리스 날짜
Amazon EC2용 AWS Identity and Access Management		Amazon EC2는 이제 AWS Identity and Access Management(IAM)와 통합됩니다. 자세한 내용은 Amazon EC2의 자격 증명 및 액세스 관리 섹션을 참조하세요.	2010년 9월 2일
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 대한 Amazon CloudWatch 모니터링이 자동으로 제공됩니다.	2010년 6월 14일
Windows용 예약 인스턴스		이제 Amazon EC2에서 Windows용 예약 인스턴스를 지원합니다.	2010년 2월 22일