AMS 고급 애플리케이션 배포 옵션

AMS 고급 애플리케이션 개발자 안내서

버전 September 13, 2024

AMS 고급 애플리케이션 개발자 안내서: AMS 고급 애플리케이션 배포 옵션

Table of Contents

애플리케이션 온보딩

AWS Managed Services(AMS) AMS 운영 계획에 오신 것을 환영합니다. 이 문서의 목적은 초기 네트워킹 및 액세스 관리가 설정된 후 애플리케이션을 AMS에 온보딩할 때 사용할 수 있는 다양한 방법과 이러한 방법을 선택할 때 고려해야 할 문제를 설명하는 것입니다.

이 문서는 시스템 통합자와 애플리케이션 개발자가 새로운 AMS 고객을 위한 애플리케이션 프로세스를 결정하고 만드는 데 도움을 주기 위한 것입니다.

애플리케이션 온보딩이란 무엇인가요?

AMS 애플리케이션 온보딩은 필요에 따라 AMS 인프라에 리소스와 애플리케이션을 배포하는 것을 말합니다. AMS 플랫폼에서 애플리케이션 및 인프라를 설계하는 것은 네이티브에서 설계하는 것과 매우 유사합니다 AWS. AMS에서 제공하는 기능을 고려하면서 애플리케이션 및 인프라 설계 모범 사례를 따르면 AWS AMS 환경에서 호스팅되는 기능 및 운영 가능한 애플리케이션이 생성됩니다.

> **Note**
>
> - 미국 동부(버지니아)
> - 미국 서부(캘리포니아 북부)
> - 미국 서부(오리건)
> - 미국 동부(오하이오)
> - 캐나다(중부)
> - 남아메리카(상파울루)
> - EU(아일랜드)
> - EU(프랑크푸르트)
> - EU(런던)
> - EU 서부(파리)
> - 아시아 태평양(뭄바이)
> - 아시아 태평양(서울)
> - 아시아 태평양(싱가포르)
> - 아시아 태평양(시드니)
> - 아시아 태평양(도쿄)

새 리전이 자주 추가됩니다. 자세한 내용은 AWS 리전 및 가용 영역을 참조하세요.

수행하는 작업, 수행하지 않는 작업

AMS는 AWS 인프라를 배포하는 표준화된 접근 방식을 제공하고 필요한 지속적인 운영 관리를 제공합니다. 역할, 책임 및 지원되는 서비스에 대한 전체 설명은 서비스 설명을 참조하세요.

> **Note**
>
> AMS가 추가 AWS 서비스를 제공하도록 요청하려면 서비스 요청을 제출합니다. 자세한 내용은 서비스 요청하기를 참조하세요.

- 수행하는 작업:

  온보딩을 완료하면 AMS 환경을 사용하여 변경(RFCs), 인시던트 및 서비스 요청에 대한 요청을 받을 수 있습니다. AMS 서비스와의 상호 작용은 애플리케이션 스택의 수명 주기를 중심으로 이루어집니다. 새 스택은 미리 구성된 템플릿 목록에서 정렬되고, 특정 Virtual Private Cloud(VPC) 서브넷으로 시작되며, 변경 요청(RFCs, 이벤트 및 인시던트가 연중무휴 모니터링됩니다.

  활성 애플리케이션 스택은 패치 적용을 포함하여 AMS에서 모니터링 및 유지 관리되며 변경이 필요하거나 스택이 폐기되지 않는 한 스택 수명 동안 추가 작업이 필요하지 않습니다. 스택의 상태 및 기능에 영향을 미치는 AMS에서 감지한 인시던트는 알림을 생성하며 해결하거나 확인하기 위해 조치가 필요할 수도 있고 필요하지 않을 수도 있습니다. 서비스 요청을 제출하여 방법 질문 및 기타 문의를 할 수 있습니다.

  또한 AMS를 사용하면 AMS에서 관리하지 않는 호환되는 AWS 서비스를 활성화할 수 있습니다. AWS-AMS 호환 서비스에 대한 자세한 내용은 셀프 서비스 프로비저닝 모드를 참조하세요.

- 하지 말아야 할 일:

  AMS는 다양한 수동 및 자동 옵션을 제공하여 애플리케이션 배포를 간소화하지만 애플리케이션의 개발, 테스트, 업데이트 및 관리는 사용자의 책임입니다. AMS는 애플리케이션에 영향을 미치는 인프라 문제에 대한 문제 해결 지원을 제공하지만 AMS는 애플리케이션 구성에 액세스하거나 검증할 수 없습니다.

AMS Amazon Machine Image(AMIs)

AMS는 AMS 지원 운영 체제에 대해 매달 업데이트된 Amazon Machine Image(AMIs 생성합니다. 또한 AMS는 AMIs)도 생성합니다. https://docs.aws.amazon.com/managedservices/latest/userguide/supported-configs.html 사용 가능한 보안 강화 이미지가 있는 운영 체제를 확인하려면 AWS Managed Services에 대해 필터링된 AWS Artifact -> 보고서 페이지(왼쪽 탐색 창에서 보고서 옵션 찾기)를 통해 사용할 수 있는 AMS 보안 사용 설명서를 참조하세요. AWS 아티팩트에 액세스하려면에서 CSDM에 문의하여 지침을 받거나 AWS 아티팩트 시작하기로 이동할 수 있습니다.

새 AMS AMIs 릴리스될 때 알림을 받으려면 "AMS AMI"라는 Amazon Simple Notification Service(Amazon SNS) 알림 주제를 구독하면 됩니다. 자세한 내용은 SNS를 사용한 AMS AMI 알림을 참조하세요.

AMS AMI 이름 지정 규칙은 입니다`customer-ams-<operating system>-<release date> -<version>`(예: `customer-ams-rhel6-2018.11-3`).

로 시작하는 AMS AMIs만 사용합니다`customer`.

AMS는 항상 최신 AMI를 사용할 것을 권장합니다. 다음 중 하나를 통해 최신 AMIs

- AMS 콘솔의 AMIs 페이지를 살펴봅니다.
- CSDM에서 또는이 ZIP 파일을 통해 사용할 수 있는 최신 AMS AMI CSV 파일 보기: ZIP의 AMS 11.2024 AMI 콘텐츠 및 CSV 파일.

  과거 AMI ZIP 파일은 문서 기록을 참조하세요.
- 이 AMS SKMS 명령 실행(AMS SKMS SDK 필요):

```
aws amsskms list-amis --vpc-id VPC_ID --query "Amis.sort_by(@,&Name)[?
 starts_with(Name,'customer')].[Name,AmiId,CreationTime]" --output table
```

운영 체제(OS)별로 AWS AMIs 콘텐츠

- Linux AMIs
  - AWS CLI 도구
  - NTP
  - Trend Micro 엔드포인트 보호 서비스 에이전트
  - 코드 배포

  - PBIS/신뢰할 수 없는 AD 브리지
  - SSM 에이전트
  - 중요 패치에 대한 Yum 업그레이드
  - AMS 사용자 지정 스크립트/관리 소프트웨어(부트, AD 조인, 모니터링, 보안 및 로깅 제어)
- Windows Server AMIs
  - Microsoft .NET Framework 4.5
  - PowerShell 5.1
  - AWS Windows PowerShell용 도구
  - 부팅, AD 조인, 모니터링, 보안 및 로깅을 제어하는 AMS PowerShell 모듈
  - Trend Micro 엔드포인트 보호 서비스 에이전트
  - SSM 에이전트
  - CloudWatch 에이전트
  - EC2Config 서비스(Windows Server 2012 R2를 통해 제공)
  - EC2Launch(Windows Server 2016 및 Windows Server 2019)
  - EC2LaunchV2(Windows Server 2022 이상)

Linux 기반 AMIs

- Amazon Linux 2023(최신 마이너 릴리스)(최소 AMI는 지원되지 않음)
- Amazon Linux 2(최신 마이너 릴리스)
- Amazon Linux 2(ARM64)
- Red Hat Enterprise 7(최신 마이너 릴리스)
- Red Hat Enterprise 8(최신 마이너 릴리스)
- Red Hat Enterprise 9(최신 마이너 릴리스)
- SUSE Linux Enterprise Server 15 SP6
- Ubuntu Linux 18.04
- Ubuntu Linux 20.04
- Ubuntu Linux 22.04
- Ubuntu Linux 24.04
- Amazon Linux: 제품 개요, 요금 정보, 사용 정보 및 지원 정보는 Amazon Linux AMI(HVM/64비트) 및 Amazon Linux 2를 참조하세요.

자세한 내용은 Amazon Linux 2 FAQs를 참조하세요.

- RedHat Enterprise Linux(RHEL): 제품 개요, 요금 정보, 사용 정보 및 지원 정보는 Red Hat Enterprise Linux(RHEL) 7(HVM)을 참조하세요.
- Ubuntu Linux 18.04: 제품 개요, 요금 정보, 사용 정보 및 지원 정보는 Ubuntu 18.04 LTS - Bionic을 참조하세요.
- SUSE Linux Enterprise Server for SAP 애플리케이션 15 SP6:
  - 계정당 다음 단계를 한 번 실행합니다.
    1. AWS Marketplace로 이동합니다.
    2. SUSE 15 SAP 제품을 검색합니다.
    3. 계속 구독하기(Continue to Subscribe)를 선택합니다.
    4. 약관에 동의를 선택합니다.
  - 새 SUSE Linux Enterprise Server for SAP Applications 15 SP6 인스턴스를 시작해야 할 때마다 다음 단계를 완료하세요.
    1. 구독한 SUSE Linux Enterprise Server for SAP Applications 15 AMI의 AMI ID를 기록해 둡니다.
    2. 배포 생성 | 고급 스택 구성 요소 | EC2 스택 | 변경 유형 생성 ct-14027q0sjyt1h RFC. *InstanceAmiId*를 구독한 AWS Marketplace AMI ID로 바꿉니다.

Windows 기반 AMIs

최신 Windows AMIs를 기반으로 하는 Microsoft Windows Server(2016, 2019 및 2022).

AMIs. https://docs.aws.amazon.com/managedservices/latest/ctref/ex-ami-create-col.html

AMS AMIs 오프보딩:

AMS는 종속성에 영향을 주지 않도록 오프보딩 중에 AMIs를 공유 해제하지 않습니다. 계정에서 AMS AMIs 제거하려면 `cancel-image-launch-permission` API를 사용하여 특정 AMIs를 숨길 수 있습니다. 예를 들어 아래 스크립트를 사용하여 이전에 계정과 공유된 모든 AMS AMIs를 숨길 수 있습니다.

```
for ami in $(aws ec2 describe-images --executable-users self --owners 027415890775 --
query 'Images[].ImageId' --output text) ;
    do
    aws ec2 cancel-image-launch-permission --image-id $ami ;
    done
```

스크립트가 오류 없이 실행되려면 AWS CLI v2가 설치되어 있어야 합니다. AWS CLI 설치 단계는 AWS CLI의 최신 버전 설치 또는 업데이트를 참조하세요. `cancel-image-launch-permission` 명령에 대한 자세한 내용은 섹션을 참조하세요`cancel-image-launch-permission`.

보안 강화 AMIs

AMS는 AMS가 지원하는 운영 체제의 하위 집합에 대해 CIS 레벨 1 벤치마크를 기반으로 보안 강화 이미지(AMIs)를 제공합니다. 사용 가능한 보안 강화 이미지가 있는 운영 체제를 찾으려면 AWS Managed Services(AMS) 고객 보안 가이드를 참조하세요. 이 가이드에 액세스하려면 왼쪽 탐색 창에서 보고서를 열고 AWS Artifact선택한 다음 AWS Managed Services를 필터링합니다. 에 액세스하는 방법에 대한 지침은 CSDM에 AWS Artifact문의하거나 AWS Artifact 시작하기에서 자세한 내용을 참조하세요.

AMS 키 용어

- AMS Advanced: AMS Advanced 설명서의 "서비스 설명" 섹션에 설명된 서비스입니다. 서비스 설명을 참조하세요.
- AMS 고급 계정: AWS AMS 고급 온보딩 요구 사항의 모든 요구 사항을 항상 충족하는 계정입니다. AMS Advanced 혜택, 사례 연구 및 영업 담당자에게 문의하는 방법에 대한 자세한 내용은 AWS Managed Services를 참조하세요.
- AMS Accelerate Accounts: AWS AMS Accelerate 온보딩 요구 사항의 모든 요구 사항을 항상 충족하는 계정입니다. AMS Accelerate 시작하기를 참조하세요.
- AWS Managed Services: AMS 및 또는 AMS Accelerate.
- AWS Managed Services 계정: AMS 계정 및 또는 AMS Accelerate 계정.
- 중요 권장 사항: 리소스 또는에 대한 잠재적 위험 또는 중단을 방지하기 위해 조치가 필요함을 알리는 서비스 요청을 AWS 통해에서 발급한 권장 사항입니다 AWS 서비스. 지정된 날짜까지 중요 권장 사항을 따르지 않기로 결정한 경우 결정으로 인한 모든 손해에 대한 책임은 전적으로 사용자에게 있습니다.
- 고객 요청 구성: 다음에서 식별되지 않은 모든 소프트웨어, 서비스 또는 기타 구성:
  - Accelerate: 지원되는 구성 또는 AMS Accelerate, 서비스 설명.
  - AMS Advanced: 지원되는 구성 또는 AMS Advanced, 서비스 설명.
- 인시던트 통신: AMS는 AMS Accelerate용 지원 센터와 AMS용 AMS 콘솔에서 생성된 인시던트를 통해 인시던트를 전달하거나 사용자가 AMS로 인시던트를 요청합니다. AMS Accelerate 콘솔은 대시보드의 인시던트 및 서비스 요청 요약과 자세한 내용은 지원 센터 링크를 제공합니다.
- 관리형 환경: AMS Advanced 계정 및 또는 AMS에서 운영하는 AMS Accelerate 계정.

AMS Advanced의 경우 여기에는 다중 계정 랜딩 존(MALZ) 및 단일 계정 랜딩 존(SALZ) 계정이 포함됩니다.

- 결제 시작 날짜:가 AWS Managed Services 온보딩 이메일에서 요청한 정보를 AWS 수신한 다음 영업일입니다. AWS Managed Services 온보딩 이메일은 계정에서 AWS Managed Services를 활성화하는 데 필요한 정보를 수집 AWS 하기 위해에서 사용자에게 보내는 이메일을 말합니다.

  이후에 등록한 계정의 경우 청구 시작일은 AWS Managed Services가 등록된 계정에 대한 AWS Managed Services 활성화 알림을 보낸 다음 날입니다. AWS Managed Services 활성화 알림은 다음과 같은 경우에 발생합니다.

  1. 호환되는 AWS 계정에 대한 액세스 권한을 부여하고 AWS Managed Services에 전달합니다.
  2. AWS Managed Services는 AWS Managed Services 계정을 설계하고 빌드합니다.
- 서비스 종료: 서비스 요청을 통해 AWS 최소 30일 전에 알림을 제공하여 어떤 이유로든 AWS Managed Services 모든 AWS Managed Services 계정 또는 지정된 AWS Managed Services 계정에 대해 AWS Managed Services를 종료할 수 있습니다. 서비스 종료 날짜에 다음 중 하나를 수행합니다.

  1. AWS 모든 AWS Managed Services 계정 또는 해당하는 경우 지정된 AWS Managed Services 계정의 제어를 사용자에게 인계하거나
  2. 양 당사자는 해당하는 경우 모든 AWS Managed Services 계정 또는 지정된 AWS Managed Services 계정에서 AWS 액세스 권한을 부여하는 AWS Identity and Access Management 역할을 제거합니다.
- 서비스 종료 날짜: 서비스 종료 날짜는 30일의 필수 종료 알림 기간이 끝나는 달의 마지막 날입니다. 필수 종료 알림 기간이 해당 월의 20일 이후인 경우 서비스 종료 날짜는 다음 달의 마지막 날입니다. 다음은 종료 날짜에 대한 예제 시나리오입니다.
  - 종료 알림이 4월 12일에 제공된 경우 30일 알림은 5월 12일에 종료됩니다. 서비스 종료 날짜는 5월 31일입니다.
  - 종료 알림이 4월 29일에 제공된 경우 30일 알림은 5월 29일에 종료됩니다. 서비스 종료 날짜는 6월 30일입니다.
- AWS Managed Services:makes를 제공하면 서비스 시작 날짜부터 각 AWS Managed Services 계정에 대한 AWS Managed Services에 액세스하고 사용할 수 있습니다. AWS
- 지정된 AWS Managed Services 계정에 대한 종료: 서비스 요청(“AMS 계정 종료 요청”)을 통해 AWS 알림을 제공하여 어떤 이유로든 지정된 AWS Managed Services 계정에 AWS Managed Services 대한 AWS Managed Services를 종료할 수 있습니다.

인시던트 관리 용어:

- 이벤트: AMS 환경의 변경 사항입니다.
- 알림: 지원되는의 이벤트가 임계값을 AWS 서비스 초과하고 경보를 트리거할 때마다 알림이 생성되고 연락처 목록으로 알림이 전송됩니다. 또한 인시던트 목록에 인시던트가 생성됩니다.
- 인시던트: AWS Managed Services 또는 사용자가 보고한 대로 영향을 미치는 AMS 환경 또는 AWS Managed Services의 예상치 못한 중단 또는 성능 저하입니다.
- 문제: 하나 이상의 인시던트에 대한 공유된 근본 원인입니다.
- 인시던트 해결 또는 인시던트 해결:
  - AMS가 해당 인시던트와 관련하여 사용할 수 없는 모든 AMS 서비스 또는 리소스를 사용 가능한 상태로 복원한 경우
  - AMS에서 사용할 수 없는 스택 또는 리소스를 사용 가능한 상태로 복원할 수 없다고 판단한 경우
  - AMS가 사용자가 승인한 인프라 복원을 시작했습니다.
- 인시던트 대응 시간: 인시던트를 생성하는 시점과 AMS가 콘솔, 이메일, 서비스 센터 또는 전화를 통해 초기 응답을 제공하는 시점 간의 시간 차이입니다.
- 인시던트 해결 시간: AMS 또는 사용자가 인시던트를 생성하는 시점과 인시던트가 해결되는 시점 간의 시간 차이입니다.
- 인시던트 우선 순위: AMS 또는 사용자가 인시던트의 우선 순위를 낮음, 중간 또는 높음으로 지정하는 방법입니다.
  - 낮음: AMS 서비스의 중요하지 않은 문제입니다.
  - 중간: 관리형 환경 내의 AWS 서비스를 사용할 수 있지만 의도한 대로 작동하지 않습니다(해당 서비스 설명에 따라).
  - 높음: (1) AMS 콘솔 또는 관리형 환경 내의 하나 이상의 AMS APIs를 사용할 수 없거나, (2) 관리형 환경 내의 하나 이상의 AMS 스택 또는 리소스를 사용할 수 없고 사용 불가능으로 인해 애플리케이션이 기능을 수행할 수 없습니다.

  AMS는 위 지침에 따라 인시던트를 다시 분류할 수 있습니다.
- 인프라 복원: 인시던트 해결이 불가능할 때 달리 지정하지 않는 한 영향을 받는 스택의 템플릿을 기반으로 기존 스택을 재배포하고 마지막으로 알려진 복원 지점을 기반으로 데이터 복원을 시작합니다.

인프라 용어:

- 관리형 프로덕션 환경: 고객의 프로덕션 애플리케이션이 있는 고객 계정입니다.
- 관리형 비프로덕션 환경: 개발 및 테스트용 애플리케이션과 같은 비프로덕션 애플리케이션만 포함하는 고객 계정입니다.

- AMS 스택: AMS에서 단일 단위로 관리하는 하나 이상의 AWS 리소스 그룹입니다.
- 변경 불가능한 인프라: Amazon EC2 Auto Scaling 그룹(ASGs)에 일반적으로 사용되는 인프라 유지 관리 모델로, 업데이트된 인프라 구성 요소(AMI)는 현재 위치에서 업데이트되지 않고 모든 배포 AWS에 대해 대체됩니다. 변경 불가능한 인프라의 장점은 모든 구성 요소가 항상 동일한 기반에서 생성되므로 동기 상태를 유지하는 것입니다. 변형성은 AMI 구축을 위한 도구 또는 워크플로와 무관합니다.
- 변경 가능한 인프라: Amazon EC2 Auto Scaling 그룹이 아니며 단일 인스턴스 또는 단 몇 개의 인스턴스를 포함하는 스택에 일반적으로 사용되는 인프라 유지 관리 모델입니다. 이 모델은 수명 주기가 시작될 때 시스템을 배포한 다음 시간이 지남에 따라 해당 시스템에 업데이트를 계층화하는 기존의 하드웨어 기반 시스템 배포를 가장 잘 나타냅니다. 시스템에 대한 모든 업데이트는 인스턴스에 개별적으로 적용되며 애플리케이션 또는 시스템 재시작으로 인해 시스템 가동 중지(스택 구성에 따라 다름)가 발생할 수 있습니다.
- 보안 그룹: 인바운드 및 아웃바운드 트래픽을 제어하는 인스턴스의 가상 방화벽입니다. 보안 그룹은 서브넷 수준이 아니라 인스턴스 수준에서 작동하므로 따라서 VPC의 서브넷에 있는 각 인스턴스에는 서로 다른 보안 그룹 세트가 할당될 수 있습니다.
- 서비스 수준 계약(SLAs): 예상 서비스 수준을 정의하는 AMS 계약의 일부입니다.
- SLA 사용 불가 및 사용 불가:
  - 오류가 발생하는 API 요청입니다.
  - 사용자가 제출한 콘솔 요청으로 5xx HTTP 응답이 발생합니다(서버가 요청을 수행할 수 없음).
  - AMS 관리형 인프라에서 스택 또는 리소스를 구성하는 모든 AWS 서비스 오퍼링은 서비스 상태 대시보드에 표시된 대로 “서비스 중단” 상태입니다.
  - AMS 제외로 인해 직접 또는 간접적으로 발생하는 사용 불가는 서비스 크레딧 자격을 결정하는 데 고려되지 않습니다. 사용 불가 기준을 충족하지 않는 한 서비스는 사용 가능한 것으로 간주됩니다.
- 서비스 수준 목표(SLOs): AMS 서비스에 대한 특정 서비스 목표를 정의하는 AMS 계약의 일부입니다.

패치 용어:

- 필수 패치: 환경 또는 계정의 보안 상태를 손상시킬 수 있는 문제를 해결하기 위한 중요한 보안 업데이트입니다. ‘중요 보안 업데이트’는 AMS 지원 운영 체제의 공급업체가 ‘중요’로 평가한 보안 업데이트입니다.
- 발표된 패치와 릴리스된 패치: 패치는 일반적으로 일정에 따라 발표되고 릴리스됩니다. 발생한 패치는 패치의 필요성이 발견되면 발표되며, 일반적으로 패치가 릴리스된 직후 발표됩니다.

- 패치 추가 기능: AWS Systems Manager (SSM) 기능을 활용하는 AMS 인스턴스에 대한 태그 기반 패치를 제공하므로 인스턴스에 태그를 지정하고 구성한 기준 및 창을 사용하여 해당 인스턴스에 패치를 적용할 수 있습니다.
- 패치 방법:
  - 현재 위치 패치: 기존 인스턴스를 변경하여 수행되는 패치입니다.
  - AMI 대체 패치: 기존 EC2 Auto Scaling 그룹 시작 구성의 AMI 참조 파라미터를 변경하여 수행되는 패치입니다.
- 패치 공급자(OS 공급업체, 타사): 패치는 애플리케이션의 공급업체 또는 관리 기관에서 제공합니다.
- 패치 유형:
  - CSU(Critical Security Update): 지원되는 운영 체제의 공급업체가 "Critical"로 평가한 보안 업데이트입니다.
  - 중요 업데이트(IU): 지원되는 운영 체제의 공급업체가 “중요”로 평가한 보안 업데이트 또는 “중요”로 평가한 비보안 업데이트입니다.
  - 기타 업데이트(OU): CSU 또는 IU가 아닌 지원되는 운영 체제의 공급업체에 의한 업데이트입니다.
- 지원되는 패치: AMS는 운영 체제 수준 패치를 지원합니다. 보안 취약성 또는 기타 버그를 수정하거나 성능을 개선하기 위해 공급업체에서 업그레이드를 릴리스합니다. 현재 지원되는 OSs 목록은 지원 구성을 참조하세요.

보안 용어:

- 탐지 제어: 보안, 운영 또는 고객 제어와 일치하지 않는 구성에 대해 고객 관리형 환경 및 워크로드를 지속적으로 감독하고 소유자에게 알리거나 리소스를 사전에 수정 또는 종료하여 조치를 취하는 AMS 생성 또는 활성화된 모니터 라이브러리입니다.

서비스 요청 조건:

- 서비스 요청: AMS가 사용자를 대신하여 수행할 작업에 대한 사용자의 요청입니다.
- 알림 알림: AMS 알림이 트리거될 때 AMS가 서비스 요청 목록 페이지에 게시한 알림입니다. 계정에 대해 구성된 연락처도 구성된 메서드(예: 이메일)를 통해 알림을 받습니다. 인스턴스/리소스에 연락처 태그가 있고 태그 기반 알림에 대해 클라우드 서비스 제공 관리자(CSDM)에 동의를 제공한 경우 태그의 연락처 정보(키 값)에도 자동 AMS 알림에 대한 알림이 전송됩니다.
- 서비스 알림: 서비스 요청 목록 페이지에 게시되는 AMS의 알림입니다.

기타 용어:

- AWS Managed Services 인터페이스: AMS: AWS Managed Services 고급 콘솔, AMS CM API 및 지원 API. AMS Accelerate의 경우: 지원 콘솔 및 지원 API.
- 고객 만족도(CSAT): AMS CSAT는 모든 사례 또는 서신에 대한 사례 대응 등급, 분기별 설문 조사 등을 포함한 심층 분석을 통해 정보를 제공합니다.
- DevOps: DevOps는 모든 단계에서 자동화 및 모니터링을 강력하게 지원하는 개발 방법론입니다. DevOps는 자동화의 토대를 통해 개발 및 운영의 기존 분리 기능을 결합하여 개발 주기를 단축하고, 배포 빈도를 늘리고, 보다 신뢰할 수 있는 릴리스를 목표로 합니다. 개발자가 운영을 관리할 수 있고 운영에서 개발을 알리면 문제와 문제가 더 빠르게 발견 및 해결되고 비즈니스 목표가 더 쉽게 달성됩니다.
- ITIL: Information Technology Infrastructure Library(ITIL이라고 함)는 IT 서비스의 수명 주기를 표준화하도록 설계된 ITSM 프레임워크입니다. ITIL은 서비스 전략, 서비스 설계, 서비스 전환, 서비스 운영, 서비스 개선 등 IT 서비스 수명 주기를 다루는 5단계로 구성됩니다.
- IT 서비스 관리(ITSM): IT 서비스를 비즈니스 요구 사항에 맞게 조정하는 일련의 사례입니다.
- 관리형 모니터링 서비스(MMS): AMS는 자체 모니터링 시스템인 관리형 모니터링 서비스(MMS)를 운영하여 AWS 상태 이벤트를 사용하고 Amazon CloudWatch 데이터 및 다른의 데이터를 집계 AWS 서비스하여 Amazon Simple Notification Service(Amazon SNS) 주제를 통해 생성된 모든 경보를 AMS 운영자(온라인 24x7)에게 알립니다.
- 네임스페이스: IAM 정책을 생성하거나 Amazon 리소스 이름(ARNs)으로 작업할 때 네임스페이스 AWS 서비스 를 사용하여를 식별합니다. 작업 및 리소스를 식별할 때 네임스페이스를 사용합니다.

운영 모델이란 무엇인가요?

AMS 고객으로서 조직은 애플리케이션 및 인프라 작업을 분리하고 인프라 작업에 AMS를 사용하기로 결정했습니다. AMS는 인프라 설계 팀과 함께 애플리케이션 설계 및 개발 팀과 협력하여 인프라 운영이 원활하게 실행되도록 합니다. 다음 그림은이 개념을 보여줍니다.

AMS는 AWS 인프라 운영에 대한 책임을 지고 팀은 애플리케이션 운영에 대한 책임을 집니다. 애플리케이션 및 인프라 설계 팀으로서 AMS 인프라의 프로덕션에 배포된 후 누가 애플리케이션을 운영할 것인지 이해해야 합니다. 이 가이드에서는 애플리케이션 배포 및 유지 관리와 관련된 인프라 설계에 대한 일반적인 접근 방식을 다룹니다.

AWS Managed Services의 서비스 관리

주제

AMS 서비스가 작동하는 방식입니다.

AWS Managed Services의 계정 거버넌스

이 섹션에서는 AMS 계정 거버넌스를 다룹니다.

AMS 전반에 걸쳐 자문 지원을 제공하고 관리형 환경의 사용 사례 및 기술 아키텍처를 자세히 이해하는 클라우드 서비스 제공 관리자(CSDM)로 지정됩니다. CSDMs 계정 관리자, 기술 계정 관리자, AWS Managed Services 클라우드 아키텍트(CAs) 및 해당하는 경우 AWS 솔루션 아키텍트(SAs)와 협력하여 소프트웨어 개발 및 운영 프로세스 전반에 걸쳐 새 프로젝트를 시작하고 모범 사례 권장 사항을 제공하는 데 도움이 됩니다. CSDM은 AMS의 기본 연락 지점입니다. CSDM의 주요 책임은 다음과 같습니다.

- 고객과 월별 서비스 검토 회의를 조직하고 주도합니다.
- 보안, 환경의 소프트웨어 업데이트 및 최적화 기회에 대한 세부 정보를 제공합니다.
- AMS에 대한 기능 요청을 포함하여 요구 사항을 충족합니다.
- 결제 및 서비스 보고 요청에 대응하고 해결합니다.
- 재무 및 용량 최적화 권장 사항에 대한 인사이트를 제공합니다.

AWS Managed Services에서 서비스 시작

서비스 시작: AWS Managed Services 계정의 서비스 시작 날짜는 AWS가 해당 AWS 관리형 서비스 계정의 온보딩 요구 사항에 명시된 활동이 완료 AWS Managed Services 되었음을 알리는 첫 번째 달의

첫 번째 날입니다. 단, AWS가 해당 월의 20일 이후에 해당 알림을 보내는 경우 서비스 시작 날짜는 해당 알림 날짜 이후 두 번째 달의 첫 번째 날입니다.

서비스 시작

- R은 작업을 달성하기 위해 작업을 수행하는 책임 당사자를 나타냅니다.
- 나는 정보, 즉 종종 작업 또는 결과물 완료 시에만 진행 상황에 대한 정보를 받는 당사자를 의미합니다.

서비스 시작

단계 #	단계 제목	설명	Custome	AMS
1.	고객 AWS 계정 인계	고객이 새 AWS 계정을 생성하고 이를 AWS Managed Services로 인계합니다.	R	정보
2.	AWS Managed Services 계정 - 디자인	AWS Managed Services 계정 설계 완료	정보	R
3.	AWS Managed Services 계정 - 빌드	AWS Managed Services 계정은 2단계의 설계에 따라 빌드됩니다.	정보	R

고객 관계 관리(CRM)

AWS Managed Services(AMS)는 고객 관계 관리(CRM) 프로세스를 제공하여 잘 정의된 관계가 설정되고 유지되도록 합니다. 이 관계의 기반은 비즈니스 요구 사항에 대한 AMS의 인사이트를 기반으로 합니다. CRM 프로세스를 통해 다음을 정확하고 포괄적으로 이해할 수 있습니다.

- 비즈니스 요구 사항 및 이러한 요구 사항을 충족하는 방법
- 기능 및 제약 조건
- AMS 및 다양한 책임과 의무

CRM 프로세스를 통해 AMS는 일관된 방법을 사용하여 서비스를 제공하고 AMS와의 관계에 대한 거버넌스를 제공할 수 있습니다. CRM 프로세스에는 다음이 포함됩니다.

- 주요 이해관계자 식별
- 거버넌스 팀 구성
- 나와 함께 서비스 검토 회의 수행 및 문서화
- 에스컬레이션 절차를 통해 공식 서비스 불만 제기 절차 제공
- 만족도 및 피드백 프로세스 구현 및 모니터링
- 계약 관리

CRM 프로세스

CRM 프로세스에는 다음과 같은 활동이 포함됩니다.

- 비즈니스 프로세스 및 요구 사항 식별 및 이해. AMS와의 계약은 이해관계자를 식별합니다.
- 요구 사항 및 요구 사항에 맞게 제공할 서비스를 정의합니다.
- 서비스 검토 회의에서 회의를 통해 AMS 서비스 범위, SLA, 계약 및 비즈니스 요구 사항의 변경 사항에 대해 논의합니다. 성과, 업적, 문제 및 행동 계획을 논의하기 위해 임시 회의가 열릴 수 있습니다.
- 고객 만족도 설문 조사와 회의에서 제공된 피드백을 사용하여 만족도를 모니터링합니다.
- 내부적으로 측정된 월별 성능 보고서에 대한 성능 보고.
- 서비스를 함께 검토하여 개선 기회를 결정합니다. 여기에는 제공된 AMS 서비스의 수준 및 품질과 관련하여 사용자와 자주 소통하는 것이 포함됩니다.

CRM 회의

AMS 클라우드 서비스 제공 관리자(CSDMs 정기적으로 회의를 개최하여 서비스 트랙(운영, 보안 및 제품 혁신) 및 경영진 트랙(SLA 보고서, 만족도 측정 및 비즈니스 요구 사항 변경)에 대해 논의합니다.

회의	용도	Mode	Participants
주간 상태 검토 (선택 사항)	미해결 문제 또는 인시던트, 패치, 보안 이벤트, 문제 레코드 12주 운영 추세 +/- 6) 애플리케이션 운영자 문제 주말 일정	현장 고객 location/ Telecom/Chime	AMS: CSDM 및 클라우드 아키텍트(CA) 고객이 할당한 팀원(예: 클라우드/인프라, 애플리케

회의	용도	Mode	Participants
			이션 지원, 아키텍처 팀 등)
월별 비즈니스 검토	서비스 수준 성능 검토(보고서, 분석 및 추세) 재무 분석 제품 로드맵 CSAT	현장 고객 location/ Telecom/Chime	AMS: CSDM, 클라우드 아키텍트(CA), AMS 계정 팀, AMS 기술 제품 관리자(TPM)(선택 사항), AMS OPS 관리자(선택 사항) 사용자: Application Operator 담당자
분기별 비즈니스 검토	점수표 및 서비스 수준 계약(SLA) 성능 및 추세(6개월) 향후 3/6/9/12개월 계획/마이그레이션 위험 및 위험 완화 주요 개선 이니셔티브 제품 로드맵 항목 미래 방향 조정 기회 재무 비용 절감 이니셔티브 비즈니스 최적화	현장 고객 위치	AMS: CSDM, 클라우드 아키텍트, AMS 계정 팀, AMS 서비스 디렉터, AMS 운영 관리자 사용자: 애플리케이션 운영자 담당자, 서비스 담당자, 서비스 디렉터

CRM 회의 계약

AMS CSDM은 다음을 포함하여 회의를 문서화할 책임이 있습니다.

- 작업 항목, 문제 및 참석자 목록을 포함한 의제 생성.
- 각 회의에서 검토한 작업 항목 목록을 생성하여 항목이 일정에 따라 완료되고 해결되었는지 확인합니다.
- 회의 후 영업일 기준 1일 이내에 이메일을 통해 회의록과 작업 항목 목록을 회의 참석자에게 배포합니다.
- 회의록을 적절한 문서 리포지토리에 저장합니다.

CSDM이 없는 경우 회의를 주도하는 AMS 담당자가 회의록을 생성하고 배포합니다.

> **Note**
>
> CSDM은 사용자와 협력하여 계정 거버넌스를 설정합니다.

CRM 월별 보고서

AMS CSDM은 월별 서비스 성능 프레젠테이션을 준비하고 전송합니다. 프레젠테이션에는 다음에 대한 정보가 포함되어 있습니다.

- 보고서 날짜
- 요약 및 인사이트:
  - 주요 콜 아웃: 총 및 활성 스택 수, 스택 패치 상태, 계정 온보딩 상태(온보딩만 해당), 고객별 문제 요약
  - 성능: 인시던트 해결, 알림, 패치 적용, 변경 요청(RFCs), 서비스 요청, 콘솔 및 API 가용성에 대한 통계
  - 문제, 과제, 우려 사항 및 위험: 고객별 문제 상태
  - 예정된 항목: 고객별 온보딩 또는 인시던트 해결 계획
- 관리형 리소스: 스택의 그래프 및 파이형 차트
- AMS 지표: 모니터링 및 이벤트 지표, 인시던트 지표, AMS SLA 준수 지표, 서비스 요청 지표, 변경 관리 지표, 스토리지 지표, 연속성 지표, Trusted Advisor 지표 및 비용 요약(여러 가지 방법 제시). 기능 요청. 연락처 정보.

> **Note**
>
> 설명된 정보 외에도 CSDM은 운영 활동에 대한 AMS의 하청업체 사용을 포함하여 범위 또는 조건의 중요한 변경 사항을 알립니다.
> AMS는 CSDM이 월별 보고서에 포함하는 패치 및 백업에 대한 보고서를 생성합니다. 보고서 생성 시스템의 일부로 AMS는 사용자가 액세스할 수 없는 일부 인프라를 계정에 추가합니다.
>
> - 원시 데이터가 보고된 S3 버킷
> - 데이터를 쿼리하기 위한 쿼리 정의가 있는 Athena 인스턴스
> - S3 버킷에서 원시 데이터를 읽기 위한 Glue 크롤러

AWS Managed Services의 비용 최적화

AWS Managed Services는 월별 비즈니스 검토(MBRs) 중에 매월 상세한 비용 사용률 및 절감 보고서를 제공합니다.

AMS는 표준 프로세스 및 메커니즘 세트를 따라 관리형 계정의 비용 절감 방법을 식별하고 AWS 지출을 최적화하기 위해 변경 사항을 계획하고 롤아웃하는 데 도움이 됩니다.

> **Note**
>
> AMS는 비용 최적화에 도움이 되는 비디오를 개발하고 있습니다. 첫 번째 단계는 비용 최적화 모범 사례의 PDF 및 Excel 스프레드시트를 제공하는 것입니다. 이러한 리소스에 액세스하려면 비용 최적화 ZIP 파일에 대한 빠른 안내서를 엽니다.

비용 최적화 프레임워크

AMS는 AWS 비용을 최적화하기 위해 3단계 접근 방식을 따릅니다.

1. 관리형 환경에서 비용 최적화 방법 식별
2. 비용 최적화 계획을 제시하세요.
3. 측정 가능한 방식으로 비용 최적화 달성 지원

관리형 환경에서 비용 최적화 방법 식별

AMS는 Cost Explorer 및 Trusted Advisor와 같은 AWS 네이티브 도구를 활용하는 동시에 아키텍처 최적화, EC2 인스턴스 및 AWS 계정 중심 최적화 전반에서 20개 이상의 비용 절감 패턴을 활용하여 맞춤형 비용 절감 권장 사항을 구축합니다.

최적화 권장 사항 중 일부는 다음과 같습니다.

아키텍처 최적화 권장 사항:

- 최적의 S3 스토리지 클래스 사용: Amazon S3는 데이터 액세스, 복원력 및 비용에 따라 다양한 워크로드 요구 사항을 충족하는 다양한 스토리지 클래스를 제공합니다. 워크로드 요구 사항에 따른 S3S3 Intelligent-Tiering 및 S3 스토리지 클래스 분석을 통해 S3 비용을 효율적으로 관리할 수 있습니다.
- 캐싱 아키텍처 사용: 해당하는 경우 캐시 인스턴스를 활용하면 IOPS 요구 사항을 충족하면서 일부 데이터베이스 인스턴스를 교체할 수 있습니다.
- EBS 업그레이드 비용 절감: EBS 볼륨을 gp2에서 gp3로 마이그레이션하면 최대 20%의 비용을 절감할 수 있으며 볼륨 크기에 관계없이 예측 가능한 3,000IOPS 기준 성능과 125MiB/s를 활용할 수 있습니다.
- 탄력성 사용:를 AWS 제공하는 Auto Scaling 기능을 사용하면 비용 최적화를 위한 효과적인 리소스 사용률과 방법을 얻을 수 있습니다. 필요에 따라 인스턴스 조정 정책을 정기적으로 검토하고 업데이트하면 비용을 절감할 수 있습니다.

EC2 인스턴스 중심 권장 사항

- 인스턴스 크기 조정: 인스턴스 크기 조정 및 사용량에 따른 최적의 구성에 중점을 둔 권장 사항입니다. 권장 사항에는 Amazon EC2 Auto Scaling 기능 활용 및 해당하는 경우 EC2 인스턴스를 Amazon S3의 AWS Lambda 또는 정적 웹 콘텐츠 등으로 대체하는 것도 포함됩니다.
- 인스턴스 예약: AMS Resource Scheduler를 사용하여 시간 일정에 따라 인스턴스를 자동으로 시작 및 중지하면 특히 업무 외 시간에 활용되지 않는 비프로덕션 인스턴스에 대한 비용을 절감하는 데 도움이 됩니다.
- 절감형 플랜 구독: 절감형 플랜은 AWS 사용량을 절약하는 가장 쉬운 방법입니다. EC2 Instance Savings Plans은 Amazon EC2 인스턴스 사용량에 대한 온디맨드 요금에 비해 최대 72%의 절감 효과를 제공합니다. Amazon SageMaker AI Savings Plans은 Amazon SageMaker AI 서비스 사용량을 최대 64% 절감합니다. AMS는 AWS 리소스 사용량에 따라 절감형 플랜에 대한 적절한 권장 사항을 제공합니다.

- 예약 인스턴스(RI) 사용 및 소비 지침: Amazon EC2 예약 인스턴스(RI)는 온디맨드 요금에 비해 상당한 할인(최대 75%)을 제공하고 특정 가용 영역에서 사용할 경우 용량 예약을 제공합니다.
- 스팟 인스턴스 사용: 내결함성 워크로드는 스팟 인스턴스를 활용하고 가격을 최대 90%까지 줄일 수 있습니다.
- 유휴 인스턴스 종료: 유휴 상태이거나 종료할 수 있는 사용률이 낮은 인스턴스를 식별하고 보고합니다.

계정 중심 권장 사항

- 계정 정리: 계정 수준에서 AMS는 사용되지 않는 EBS 볼륨, 중복 CloudTrail 추적, 미사용 리소스가 있는 빈 계정 등을 식별하고 정리를 위한 권장 사항을 제공합니다.
- SLA 권장 사항: 또한 AMS는 Plus 및 Premium 계정을 정기적으로 검토하고 계정에 적합한 SLA 수준을 선택할 것을 권장합니다.
- AMS 자동화 최적화: AMS는 AMS 서비스를 제공하는 데 사용되는 AMS 자동화 및 인프라를 지속적으로 최적화합니다.

고객에게 제시 및 계획 지원

AMS는 주요 고객 이해관계자와 함께 월별 비즈니스 검토(MBRs)를 수행하고 잠재적 비용 절감과 함께 식별된 비용 절감 방법, 메커니즘 및 권장 사항을 제시합니다. 또한 필요한 변경 사항을 계획하기 위해 고객과 협력합니다.

권장 사항 구현 지원 및 비용 영향 측정

AMS는 비용 영향 및 최적화 변경을 달성하고 측정하는 데 도움이 됩니다.

권장 변경 사항의 애플리케이션 영향, 위험 및 성공 기준을 평가하고 AMS 콘솔을 통해 적절한 변경 요청(RFCs)을 제기합니다. AMS는 사용자와 협업하고 관리형 계정에서 비용 최적화와 관련된 변경 사항을 구현합니다. AMS는 비용 영향을 측정하고 월별 비즈니스 검토(MBRs)에 실현된 절감액을 포함합니다.

비용 최적화 책임 매트릭스

AMS 비용 최적화에 대한 책임.

비용 최적화 RACI

활동	Customer	AMS
비용 절감 권장 사항 컴파일 및 보고서 준비	정보	R
비용 절감 보고서 제시	C	R
비용 절감과 관련된 변경 계획	R	C
변경 영향 및 위험 평가	R	C
변경 사항 구현 RFCs 증가	R	C
RFCs 검토 및 변경 사항 구현	C	R
애플리케이션 테스트 및 변경 구현 검증	R	C

활동	Customer	AMS
변경 후 비용에 미치는 영향 측정 및 고객에게 제시	정보	R

AWS Managed Services의 서비스 시간

Feature	AMS 고급
	프리미엄 티어
서비스 요청	24/7
인시던트 관리(P2-P3)	24/7
백업 및 복구	24/7
패치 관리	24/7
모니터링 및 알림	24/7
자동 변경 요청(RFC)	24/7
자동이 아닌 변경 요청(RFC)	24/7
클라우드 서비스 제공 관리자(CSDM)	월요일~금요일: 08:00~17:00, 현지 업무 시간

AWS Managed Services에 대한 도움말 보기

AMS는 하루 24시간, 주 7일, 1년 365일(계정에 적용된 AMS 서비스 수준 계약에 따라) Incident Management, Service Request Management 및 Change Management를 지원합니다.

관리형 환경에 영향을 미치는 AWS 또는 AMS 서비스 성능 문제를 보고하려면 AMS 콘솔을 사용하여 인시던트 보고서를 제출합니다. 자세한 내용은 인시던트 보고를 참조하세요. AMS 인시던트 관리에 대한 일반적인 정보는 인시던트 대응을 참조하세요.

정보나 조언을 요청하거나 AMS에서 추가 서비스를 요청하려면 AMS 콘솔을 사용하여 서비스 요청을 제출합니다. 자세한 내용은 서비스 요청 생성을 참조하세요. AMS 서비스 요청에 대한 일반 정보는 서비스 요청 관리를 참조하세요.

애플리케이션 개발

AWS Managed Services(AMS) 환경에 애플리케이션을 효과적으로 설계하고 배포할 수 있는 애플리케이션 개발 프로세스 및 사례입니다. AMS는 다음과 같은 상위 수준 프로세스를 안내합니다.

1. AMS 관리형 환경에 개발하거나 통합할 애플리케이션을 구상하고 설계합니다. 몇 가지 고려 사항은 다음과 같습니다.
   a. 애플리케이션을 어떻게 배포하나요? Ansible과 같은 배포 도구를 사용하여 자동화하거나 필요한 파일을 직접 업로드하여 수동으로 자동화할 수 있습니까?
   b. 애플리케이션을 업데이트하려면 어떻게 해야 하나요? 각 인스턴스를 개별적으로 업데이트하는 변경 가능한 접근 방식을 사용하거나 Auto Scaling 그룹에서 업데이트된 단일 AMI로 각 인스턴스를 업데이트하는 변경 불가능한 접근 방식을 사용하나요?
2. AWS 아키텍처 라이브러리, AWS "Well-Architected" 지침, AMS 및 기타 클라우드 아키텍처 주제 전문가를 사용하여 애플리케이션을 호스팅하는 데 사용할 인프라를 계획하고 설계합니다. 이 가이드의 다음 섹션에서는 이에 도움이 될 수 있는 정보를 제공합니다.
3. 인프라 배포 접근 방식을 선택합니다.
   a. 전체 스택: 모든 인프라 구성 요소가 한 번에 함께 배포됩니다.
   b. 티어 및 타이: 인프라 배포는 별도로 배포되며, 이후에 보안 그룹 수정 사항과 함께 연결됩니다. 이러한 유형의 배포는 Auto Scaling 그룹을 생성할 때 이전에 생성한 로드 밸런서를 지정하는 등 서로를 기반으로 빌드되는 스택 구성 요소의 직렬 구성으로도 수행됩니다.
   c. Dev, Staging, Prod와 같은 어떤 환경을 사용하시겠습니까?
4. 필요한 스택 또는 계층을 프로비저닝할 AMS 변경 유형(CTs)을 선택하고 필요한 변경 요청(RFCs)을 준비합니다.
5. RFCs를 제출하여 적절한 환경에 인프라 배포를 트리거합니다.
6. 선택한 애플리케이션 배포 접근 방식을 사용하여 애플리케이션을 배포합니다.
7. 필요에 따라 인프라와 애플리케이션을 재작업합니다.
8. 인프라와 애플리케이션을 적절한 후속 환경에 배포합니다. 단, 첫 번째 배포는 비프로덕션 환경에 배포하는 것으로 가정합니다.
9. 지속적인 유지 관리는 기본 인프라를 운영하는 AMS와 애플리케이션(들) 인프라를 운영하는 운영 팀에서 처리합니다.
10. 애플리케이션을 폐기하려면 애플리케이션을 위한 AMS 인프라를 종료합니다.

잘 설계됨

에서는 잘 설계된 시스템이 비즈니스 성공 가능성을 크게 높인다고 AWS 생각합니다. AWS 아키텍처 센터는의 아키텍처 설계에 대한 전문가 지침을 제공합니다 AWS 클라우드.

시스템을 구축할 때 내려야 하는 결정의 장단점을 이해하는 데 도움이 되는 다음 문서와 백서를 권장합니다 AWS.

Well-Architected입니까?: 다음 6가지 원칙을 기반으로 AWS Well-Architected 프레임워크를 소개합니다.

- 운영 우수성: 운영 우수성 원칙은 비즈니스 가치를 제공하고 프로세스와 절차를 지속적으로 개선하기 위해 시스템을 실행하고 모니터링하는 데 중점을 둡니다. 주요 주제에는 변경 사항 관리 및 자동화, 이벤트 대응, 일상적인 운영을 성공적으로 관리하기 위한 표준 정의가 포함됩니다.
- 보안: 보안 원칙은 정보와 시스템을 보호하는 데 중점을 둡니다. 주요 주제에는 데이터의 기밀성 및 무결성, 권한 관리를 통해 수행할 수 있는 작업 식별 및 관리, 시스템 보호, 보안 이벤트를 탐지하기 위한 제어 설정 등이 포함됩니다.
- 신뢰성: 신뢰성 원칙은 비즈니스 및 고객 수요를 충족하기 위해 장애를 방지하고 신속하게 복구하는 능력에 중점을 둡니다. 주요 주제에는 설정, 교차 프로젝트 요구 사항, 복구 계획 및 변경 처리 방법에 대한 기본 요소가 포함됩니다.
- 성능 효율성: 성능 효율성 원칙은 IT 및 컴퓨팅 리소스를 효율적으로 사용하는 데 중점을 둡니다. 주요 주제로는 워크로드 요구 사항을 기반으로 적절한 리소스 유형 및 크기 선택하기, 성능 모니터링하기, 비즈니스 요구 사항이 변화함에 따라 효율성을 유지하기 위해 정보에 입각하여 의사 결정 내리기가 있습니다.
- 비용 최적화: 비용 최적화 원칙은 불필요한 비용을 방지하는 데 중점을 둡니다. 주요 주제에는 비용이 지출되는 위치를 이해하고 제어, 가장 적절하고 적절한 리소스 유형 수 선택, 시간 경과에 따른 지출 분석, 과도한 지출 없이 비즈니스 요구 사항을 충족하도록 규모 조정 등이 포함됩니다.
- 지속 가능성: 지속 가능성 원칙은 프로비저닝된 리소스의 이점을 극대화하고 필요한 총 리소스를 최소화하여 워크로드의 모든 구성 요소에서 에너지 소비를 줄이고 효율성을 높여 지속 가능성 영향을 지속적으로 개선하는 능력에 중점을 둡니다.

AWS Well-Architected Framework:를 AWS 통해 고객이 클라우드 기반 아키텍처를 평가 및 개선하고 설계 결정의 비즈니스 영향을 더 잘 이해할 수 있는 방법을 설명합니다. 일반적인 설계 원칙과 Well-Architected Framework의 원칙으로를 AWS 정의하는 6가지 개념 영역의 특정 모범 사례 및 지침을 다룹니다.

AMS의 애플리케이션 계층 책임과 인프라 계층 책임 비교

AMS를 사용하면 AMS에서 인프라와 유지 관리 및 성장에 필요한 모든 것을 유지 관리합니다. 그러나 line-of-business 애플리케이션 또는 제품 애플리케이션에 필요한 모든 것은 사용자가 개발, 배포 및 유지 관리합니다.

CodeDeploy 및 Chef, Puppet AWS CloudFormation, Ansible 또는 Saltstack과 같은 애플리케이션 배포 도구를 사용하면 AMS 관리형 인프라에 대한 애플리케이션 배포를 완전히 자동화할 수 있습니다.

AMS가 수행하는 작업과 수행하지 않는 작업에 대한 자세한 내용은 섹션을 참조하세요수행하는 작업, 수행하지 않는 작업.

AMS의 Amazon EC2 인스턴스 변경 가능성

사용자와 AMS는 다음 두 가지 방법 중 하나로 인프라에서 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 유지할 수 있습니다.

- 변경 불가능:이 모델은 필요한 기능과 함께 베이크된(생성된) Amazon Machine Image(AMIs)를 사용합니다. 업데이트를 배포할 때 기존 인스턴스가 손상되고 업데이트된 AMI에서 생성된 새 인스턴스로 완전히 대체됩니다. 가동 중지 시간을 최소화하기 위해이 롤링 프로세스는 일부 인스턴스는 업데이트되지 않고 액세스할 수 없는 반면, 다른 인스턴스는 최종적으로 새 변경 사항이 완전히 배포될 때까지 업데이트되고 있습니다.
- 변경 가능:이 모델에서 인프라는 클라우드의 기존 시스템에 배포되는 새 코드로 업데이트됩니다. 이 모델은 수동으로 업데이트를 푸시하고 infrastructure-as-code 사용하여 업데이트를 배포하는 혼합이며 새 AMIs에 의존하지 않습니다.

이러한 유지 관리 모델은이 가이드의 후반부에서 자세히 설명합니다.

AMS 리소스와 함께 AWS Secrets Manager 사용

AMS와 암호를 공유해야 하는 경우가 많습니다. 예를 들면 다음과 같습니다.

- RDS 인스턴스의 마스터 암호 재설정
- 로드 밸런서 인증서
- AMS에서 IAM 사용자를 위한 수명이 긴 자격 증명 획득

기밀 정보를 AMS와 공유하는 가장 안전한 방법은 AWS Secrets Manager를 사용하는 것입니다. 다음 단계를 따르세요.

1. 연합 액세스 및 단일 계정 랜딩 존(SALZ)에 대한 CustomerReadOnly 역할을 사용하여 AWS 콘솔에 로그인합니다. 다중 계정 랜딩 존(MALZ)에 대해 AWSManagedServicesSecurityOpsRole, AWSManagedServicesAdminRole 및 AWSManagedServicesChangeManagementRole 중 하나를 사용합니다.
2. AWS Secrets 관리자 콘솔로 이동하여 새 보안 암호 저장을 클릭합니다.
3. "기타 유형의 보안 암호"를 선택합니다.
4. 보안 암호 값을 일반 텍스트로 입력하고 다음을 클릭합니다.
5. 보안 암호 이름과 설명을 입력합니다. 이름은 항상 "customer-shared/*"로 시작해야 합니다. 예: "customer-shared/license-2018". 완료되면 다음을 클릭하여 계속합니다.
6. 기본 KMS 암호화를 사용합니다.
7. 자동 교체를 비활성화한 상태로 두고 다음을 클릭합니다.
8. 검토 후 저장을 클릭하여 보안 암호를 저장합니다.
9. 보안 암호 이름과 ARN을 사용하여 AMS 서비스 요청에 회신하면 보안 암호를 식별하고 검색할 수 있습니다. 서비스 요청 생성에 대한 자세한 내용은 서비스 요청 예제를 참조하세요.

AMS에서의 애플리케이션 배포

온보딩 중에 AWS Managed Services(AMS)는 고객과 협력하여 필요한 인프라를 결정합니다.

기본 인프라에는 AWS 가상 프라이빗 클라우드(VPC), ADFS 포리스트 신뢰를 통한 통신 보안, 두 가용 영역에 걸쳐 미러링되고 관리형 NAT, 접속, 퍼블릭 로드 밸런서(DX) 및 필수 보안으로 구성된 기본 서브넷 AWS Direct Connect (DMZ, 공유 서비스 및 프라이빗)이 포함됩니다. 애플리케이션 리소스는 프라이빗 또는 고객 애플리케이션 서브넷에 배포됩니다. AWS Managed Services 사용 설명서에서 일반적인 AMS 아키텍처에 대해 자세히 알아볼 수 있습니다.

기본 사항이 완료되면 배포하는 인프라에는 애플리케이션 및 애플리케이션 개발을 위한 모든 구성 요소가 포함되어야 합니다.

AMS의 애플리케이션 배포 기능

AMS에서 애플리케이션을 배포할 수 있는 몇 가지 방법입니다. 각 메서드에 대한 세부 정보는 다음과 같습니다.

애플리케이션 배포 기능 예제

메서드 이름	인프라 배포	AMI 또는 키 요소(들)	애플리케이션 설치
변경 가능한 애플리케이션, AMS AMI			
수동 애플리케이션 배포	전체 스택 CT 또는 티어 및 타이 CTs	AMS 제공 AMI	액세스 관리 CT를 제출하고 애플리케이션을 수동으로 설치합니다.
애플리케이션 에이전트를 사용한 UserData 애플리케이션 배포(예: Chef, Puppet 등)			애플리케이션 에이전트를 설치하고 해당 스크립트/에이전트가 애플리케이션을 설치하는 UserData 스크립팅과 함께 프로비저닝 CT를 사용합니다.
UserData 에이전트 없는 애플리케이션 배포			액세스 관리 CT를 제출하고 애플리케이션

메서드 이름	인프라 배포	AMI 또는 키 요소(들)	애플리케이션 설치
(예: Ansible, Salt SSH 등)			에이전트를 설치합니다. 애플리케이션 배포 도구를 사용하여 애플리케이션을 배포합니다.
변경 가능한 애플리케이션, 사용자 지정 AMI			
사용자 지정 AMI 애플리케이션 배포(비 ASG)	전체 스택 CT 또는 티어 및 타이 CTs	사용자 지정 AMI. AMS AMI -> 애플리케이션 배포 도구 에이전트로 사용자 지정 -> EC2 인스턴스(CT) 생성 -> AMI(CT) 생성.	에이전트를 활용하는 애플리케이션 배포 도구(예: Chef)는 애플리케이션을 배포합니다.
AWS Database Migration Service(DMS) 애플리케이션 배포	AWS DMS를 기존 AMS 관계형 데이터베이스 스택에 동기화합니다.	사용자 지정 AMI	고객 또는 파트너가 AWS Database Migration Service를 사용합니다. AMS는 시작 시 AMS 구성 요소를 확인합니다.
워크로드 수집 애플리케이션 배포	파트너가 마이그레이션한 인스턴스/AMI 및 고객이 시작한 워크로드 수집 CT.		파트너가 인스턴스를 마이그레이션하고 고객 AMS 관리형 VPC에서 AMI를 생성합니다. 고객은 워크로드 수집 CT를 사용하여 AMS에서 스택을 시작합니다. 자세한 내용은 AMS 워크로드 인제스트(WIGS)을 참조하세요.
변경 불가능한 애플리케이션			

메서드 이름	인프라 배포	AMI 또는 키 요소(들)	애플리케이션 설치
사용자 지정 AMI 애플리케이션 배포(ASG)	전체 스택 CT 또는 티어 및 타이 CTs	AMS AMI -> 사용자 지정 -> EC2 인스턴스(CT) 생성 -> AMI(CT) 생성 -> Auto Scaling 그룹 생성.	Auto Scaling은 사용자 지정 AMI를 사용하여 애플리케이션을 배포합니다. 자세한 내용은 AMS의 티어 및 타이 앱 배포을 참조하세요.
변경 가능 또는 변경 불가능한 애플리케이션			
사용자 지정 CloudFormation 템플릿 애플리케이션 배포	CloudFormation 템플릿	AWS CloudFormation 템플릿 -> AMS 사용자 지정/준비 -> 배포 | 수집 | CloudFormation 템플릿의 스택 | 생성(ct-36cn2avfrrj9v).	AMS는 사용자 지정 CloudFormation 템플릿을 사용하여 계정에 애플리케이션을 배포하고 애플리케이션 배포를 검증합니다. 자세한 내용은 AMS CloudFormation 수집을 참조하세요.
SQL 데이터베이스 가져오기	AMS 작업(기타 | 기타 CT)	온프레미스 SQL 데이터베이스 -> .bak 파일 -> AMS RDS SQL 데이터베이스 -> 관리 | 기타 | 기타 | 가져오기를 위한 생성(ct-1e1xtak34nx76)	AMS는 온프레미스 데이터베이스를 AMS 관리형 RDS 데이터베이스로 가져옵니다. 자세한 내용은 AMS RDS for Microsoft SQL Server로 데이터베이스(DB) 가져오기을 참조하세요.

메서드 이름	인프라 배포	AMI 또는 키 요소(들)	애플리케이션 설치
Database Migration Service(DMS)	AMS 작업(다중 CTs)	온프레미스 데이터베이스 -> DMS 복제 인스턴스 -> DMS 복제 서브넷 그룹 -> DMS 대상 엔드포인트 -> DMS 소스 엔드포인트 -> DMS 복제 작업.	AMS는 온프레미스 데이터베이스를 AMS 관리형 S3 또는 대상 RDS 데이터베이스로 가져옵니다. 자세한 내용은 AWS Database Migration Service (AWS DMS)을 참조하세요.
CodeDeploy 애플리케이션 배포	CodeDeploy	애플리케이션 -> CodeDeploy 애플리케이션 -> CodeDeploy 배포 그룹 -> CodeDeploy 배포.	사용량에 따라 인플레이스 또는 블루/그린 애플리케이션 배포. 자세한 내용은 CodeDeploy 요청 섹션을 참조하십시오.

AMS에서 애플리케이션 배포 계획

애플리케이션 배포를 활성화하기 위해 답변해야 할 권장 질문 세트는 섹션을 참조하세요부록: 애플리케이션 온보딩 설문지. 질문에서는 다음을 설명합니다.

- 배포 요약
- 인프라 배포 구성 요소
- 애플리케이션 호스팅 플랫폼
- 애플리케이션 배포 모델
- 애플리케이션 종속성
- 제품 애플리케이션용 SSL 인증서

AMS 워크로드 인제스트(WIGS)

주제

- 워크로드 마이그레이션: Linux 및 Windows의 사전 조건

- 마이그레이션이 리소스를 변경하는 방법
- 워크로드 마이그레이션: 표준 프로세스
- 워크로드 마이그레이션: CloudEndure 랜딩 존(SALZ)
- AMS 도구 계정(워크로드 마이그레이션)
- 워크로드 마이그레이션: Linux 사전 수집 검증
- 워크로드 마이그레이션: Windows 사전 통합 검증
- 워크로드 수집 스택: 생성

AMS 클라우드 마이그레이션 파트너와 함께 AMS 워크로드 수집 변경 유형(CT)을 사용하여 기존 워크로드를 AMS 관리형 VPC로 이동합니다. AMS 워크로드 수집을 사용하면 마이그레이션된 인스턴스를 AMS로 이동한 후 사용자 지정 AMS AMI를 생성할 수 있습니다. 이 섹션에서는 마이그레이션 파트너와 사용자가 AMS 워크로드 수집을 위해 수행하는 프로세스, 사전 조건 및 단계에 대해 설명합니다.

Important

AMS 워크로드 수집에서 운영 체제를 지원해야 합니다. 지원되는 운영 체제는 섹션을 참조하세요워크로드 마이그레이션: Linux 및 Windows의 사전 조건.
워크로드와 계정마다 다릅니다. AMS는 성공적인 결과를 준비하기 위해 사용자와 협력합니다.

다음 다이어그램은 AMS 워크로드 수집 프로세스를 보여줍니다.

워크로드 마이그레이션: Linux 및 Windows의 사전 조건

온프레미스 인스턴스의 사본을 AWS Managed Services(AMS)에 수집하기 전에 특정 사전 조건을 충족해야 합니다. 이는 Windows 운영 체제와 Linux 운영 체제 간에 다른 사전 조건을 포함한 사전 조건입니다.

Note

인스턴스를 수집할 준비가 되었는지 확인하는 프로세스를 간소화하기 위해 Windows와 Linux 모두에 대한 검증 도구가 생성되었습니다. 이러한 도구는 온프레미스 서버와 AWS의 EC2 인스턴스에서 직접 다운로드하여 실행할 수 있습니다. Linux Pre-WIGS Validation.zip, Windows Pre-WIGS Validation.zip.

시작하기 전에 Linux 및 Windows의 경우:

- 전체 바이러스 스캔을 수행합니다.
- 인스턴스에는 `customer-mc-ec2-instance-profile` 인스턴스 프로파일이 있어야 합니다.
- Amazon EC2 Systems Manager(SSM) 에이전트를 설치하고 SSM 에이전트가 실행 중인지 확인합니다.
- AMS 워크로드 수집(WIGS)을 실행하려면 루트 볼륨에 최소 10GB의 여유 디스크 공간이 권장됩니다. 운영상 AMS는 75% 미만의 디스크 사용률을 권장하고 디스크 사용률이 85%에 도달하면 알림을 보냅니다.
- 마이그레이션 파트너와의 수집 기간을 결정합니다.
- 사용자 지정 AMI는 대상 프로덕션 AMS 계정에 EC2 인스턴스로 존재합니다(이것은 마이그레이션 파트너의 책임입니다).

Important

AMS 워크로드 수집에서 운영 체제를 지원해야 합니다.

- 지원되는 운영 체제는 다음과 같습니다.
  - Microsoft Windows Server: 2008 R2, 2012, 2012 R2, 2016, 2019 및 2022
  - Linux: Amazon Linux 2023, Amazon Linux 2 및 Amazon Linux, CentOS 7.x, CentOS 6.5-6.10, Oracle Linux 7: 마이너 버전 7.5 이상, Oracle Linux 8: 최대 8.3, RHEL 8.x, RHEL 7.x, RHEL 6.5-6.10, SUSE Linux Enterprise Server 15 SP3, SP4 및 SAP 특정 버전, SUSE Linux Enterprise Server 12 SP5, Ubuntu 18.04
- 다음 AMIs는 지원되지 않습니다.
  - Amazon Linux 2023 미니멀 AMI.

Note

AMS API/CLI(amscm 및 amsskms) 엔드포인트는 AWS 버지니아 북부 리전에 있습니다`us-east-1`. 인증 설정 방식과 계정 및 리소스가 있는 AWS 리전에 따라 명령을 실행할 `--region us-east-1` 때를 추가해야 할 수 있습니다. 인증 방법인 `--profile saml`경우를 추가해야 할 수도 있습니다.

LINUX 사전 조건

WIGS RFC를 제출하기 전에에 나열된 요구 사항을 준수하고 다음을 워크로드 마이그레이션: Linux 및 Windows의 사전 조건 확인합니다.

- 최신 향상된 네트워킹 드라이버가 설치되어 있습니다. Linux의 향상된 네트워킹을 참조하세요.
- AMS 구성 요소와 충돌하는 타사 소프트웨어 구성 요소가 제거되었습니다.
  - 바이러스 백신 클라이언트
  - 백업 클라이언트
  - 가상화 소프트웨어(예: VM 도구 또는 Hyper-V 통합 서비스)
  - 액세스 관리 소프트웨어(예: SSSD, Centrify 또는 PBIS)
- SSH가 올바르게 구성되었는지 확인 - 그러면 SSH에 대한 프라이빗 키 인증이 일시적으로 활성화됩니다. AMS는 구성 관리 도구와 함께 이를 사용합니다. 다음 명령을 사용합니다.

```
sudo grep -q "^PubkeyAuthentication" /etc/ssh/sshd_config && sudo sed "s/
^PubkeyAuthentication=.*/PubkeyAuthentication yes/" -i /etc/ssh/sshd_config || sudo
 sed "$ a\PubkeyAuthentication yes" -i /etc/ssh/sshd_config
```

```
sudo grep -q "^AuthorizedKeysFile" /etc/ssh/sshd_config && sudo sed "s/
^AuthorizedKeysFile=.*/AuthorizedKeysFile %h\/.ssh\/authorized_keys/" -i /etc/ssh/
sshd_config || sudo sed "$ a\AuthorizedKeysFile %h/.ssh/authorized_keys" -i /etc/ssh/
sshd_config
```

- Yum이 올바르게 구성되었는지 확인 - Yum 리포지토리를 사용하려면 RedHat에 라이선스가 필요합니다. 인스턴스는 Satellite Server 또는 RedHat Cloud Server를 통해 라이선스를 받아야 합니다. 라이선스가 필요한 경우 다음 링크 중 하나를 사용합니다.
  - Red Hat 위성
  - Red Hat 클라우드 액세스
- Red Hat 위성을 사용하는 경우 WIGS는 Red Hat 소프트웨어 컬렉션(RHSCL)을 추가해야 합니다. WIGS 시스템은 RHSCL을 사용하여 시스템에 구성된 것과 함께 Python3.6 인터프리터를 추가합니다. 이 솔루션을 지원하려면 다음 리포지토리를 사용할 수 있어야 합니다.
  - rhel-server-rhscl
  - rhel-server-releases-optional

Windows 사전 조건

WIGS RFC를 제출하기 전에에 나열된 요구 사항을 준수하고 다음을 워크로드 마이그레이션: Linux 및 Windows의 사전 조건 확인합니다.

- Powershell 버전 3 이상이 설치되어 있습니다.
- AWS EC2 Config는 마이그레이션할 워크로드와 함께 인스턴스에 설치됩니다.
- PV, ENA 및 NVMe와 같은 최신 세대 인스턴스 유형을 지원하는 AWS 드라이버를 설치합니다. 다음 링크의 정보를 사용할 수 있습니다.
  - Windows 인스턴스에서 PV 드라이버 업그레이드
  - Windows에서의 확장 네트워킹
  - Windows 인스턴스용 AWS NVMe 드라이버
  - 3부: AWS NVMe 드라이버 업그레이드
  - 5부: 베어 메탈 인스턴스용 직렬 포트 드라이버 설치
  - 6부: 전력 관리 설정 업데이트
- (선택 사항이지만 권장됨) 중요 서비스 비활성화 - 데이터베이스와 같은 중요한 애플리케이션 서비스를 비활성화로 설정하되 애플리케이션 확인 단계에서 원래 시작 모드로 되돌릴 수 있도록 변경 사항이 문서화되었는지 확인합니다.
- (선택 사항이지만 권장) 준비된 인스턴스에서 Failsafe AMI를 생성합니다.
  - 배포 사용 | 고급 스택 구성 요소 | AMI | 생성
  - 생성 중에 Key=Name, Value=APPLICATION-ID_IngestReady 태그를 추가합니다.
  - 계속하기 전에 AMI가 생성될 때까지 기다립니다.
- AMS 구성 요소와 충돌하는 타사 소프트웨어 구성 요소가 제거되었습니다.
  - 바이러스 백신 클라이언트
  - 백업 클라이언트
  - 가상화 소프트웨어(예: VM 도구 또는 Hyper-V 통합 서비스)

> **Note**
>
> Windows Server용 End-of-Support 마이그레이션 프로그램(EMP)에는 레거시 애플리케이션을 리팩터링 없이 Windows Server 2003, 2008 및 2008 R2에서 AWS에서 지원되는 최신 버전으로 마이그레이션하는 도구가 포함되어 있습니다.

마이그레이션이 리소스를 변경하는 방법

이 섹션에 설명된 수집 RFC는 AMS가 인스턴스를 관리할 수 있도록 AMS 계정으로 마이그레이션된 후 인스턴스에 구성을 추가하는 다음 단계를 수행합니다.

추가된 구성은 다음과 같이 AMS에 따라 다릅니다.

수집된 Linux 인스턴스에 대한 변경 사항:

- 설치된 소프트웨어:
  - Cloud Init: Jarvis Access에 대한 프라이빗 키를 구성하는 데 사용됩니다.
  - 지원되는 모든 운영 체제에 대한 Python 3(스크립팅 언어)(CentOS 6, RHEL 8, OracleLinux 7 제외).
  - AWS CloudFormation Python 헬퍼 스크립트: AWS CloudFormation은 Amazon EC2 인스턴스에서 소프트웨어를 설치하고 서비스를 시작하는 데 사용되는 스크립트를 제공합니다.
  - AWS CLI: AWS CLI는 AWS 서비스와 상호 작용하기 위한 명령을 제공하는 Python용 AWS SDK(Boto)를 기반으로 구축된 오픈 소스 도구입니다.
  - AWS SSM 에이전트: SSM 에이전트는 Systems Manager 서비스의 요청을 처리하여 요청에 지정된 대로 시스템을 구성합니다.
  - AWS CloudWatch Logs Agent: CloudWatch로 로그를 전송합니다.
  - AWS CodeDeploy: Amazon EC2 인스턴스, 온프레미스 인스턴스 또는 서버리스 Lambda 함수에 대한 애플리케이션 배포를 자동화하는 배포 서비스입니다.
  - Ruby: CodeDeploy에 필요
  - 시스템 성능 도구(sysstat): Sysstat에는 시스템 성능 및 사용 활동을 모니터링하는 다양한 유틸리티가 포함되어 있습니다.
  - AD 브리지(이전 PowerBroker Identity Services): Microsoft가 아닌 호스트를 Active Directory 도메인에 조인합니다.
  - Trend Micro Deep Security Agent: 바이러스 백신 소프트웨어.
- 변경된 소프트웨어:
  - 인스턴스는 UTC 시간대를 사용하도록 구성됩니다.

수집된 Windows 인스턴스에 대한 변경 사항:

- 설치된 소프트웨어:

  - Windows PowerShell용 AWS 도구: PowerShell용 AWS 도구를 사용하면 개발자와 관리자가 PowerShell 스크립팅 환경에서 AWS 서비스 및 리소스를 관리할 수 있습니다.
  - Trend Micro Deep Security Agent: 바이러스 백신 보호
  - 부팅, Active Directory 조인, 모니터링, 보안 및 로깅을 제어하기 위한 PowerShell 코드가 포함된 AMS PowerShell 모듈입니다.
- 변경된 소프트웨어:
  - 서버 메시지 블록(SMB) 버전 1이 비활성화되었습니다.
  - Windows 원격 관리(WinRM)는 포트 5986에서 수신 대기하도록 활성화 및 구성됩니다. 이 인바운드 포트를 허용하는 방화벽 규칙도 생성됩니다.
- 설치하거나 변경할 수 있는 소프트웨어:
  - Microsoft .Net Framework 4.5(개발자 플랫폼) 버전 이하가 감지되면 .Net Framework 4.5가 감지됩니다.
  - Windows 2012, Windows 2012R2의 경우 PowerShell 5.1로 업그레이드됩니다.

워크로드 마이그레이션: 표준 프로세스

> **Note**
>
> 이 프로세스에는 두 당사자가 필요하므로이 섹션에서는 AMS 클라우드 마이그레이션 파트너(마이그레이션 파트너)와 애플리케이션 소유자(사용자)의 각 작업에 대해 설명합니다.

1. 마이그레이션 파트너, 설정:
   a. 마이그레이션 파트너는 인스턴스를 마이그레이션할 목적으로 IAM 역할에 대한 서비스 요청을 AMS에 제출합니다. 서비스 요청 제출에 대한 자세한 내용은 서비스 요청 예제를 참조하세요.
   b. 마이그레이션 파트너가 관리자 액세스 요청을 제출합니다. AMS 운영 팀은 요청된 IAM 역할을 통해 마이그레이션 파트너에게 계정에 대한 액세스 권한을 제공합니다.
2. 마이그레이션 파트너, 개별 워크로드 마이그레이션:

a. 마이그레이션 파트너는 `customer-mc-ec2-instance-profile` IAM 인스턴스 프로파일(계정에 있어야 함)을 사용하여 기본 Amazon EC2 또는 기타 마이그레이션 도구를 통해 비 AWS 인스턴스를 AMS 계정의 서브넷으로 마이그레이션합니다.

b. 마이그레이션 파트너는 마이그레이션 파트너 마이그레이션 인스턴스의 배포 | 수집 | 스택 | CT 생성(ct-257p9zjk14ija)이 포함된 RFC를 제출합니다.이 RFC 생성 및 제출에 대한 자세한 내용은 섹션을 참조하세요워크로드 수집 스택: 생성.

   RFC의 실행 출력은 인스턴스 ID, IP 주소 및 AMI ID를 반환합니다.

   마이그레이션 파트너는 계정에 생성된 워크로드의 인스턴스 ID를 제공합니다.

3. 마이그레이션에 액세스하고 검증합니다.

a. 마이그레이션 파트너가 제공한 실행 출력(AMI ID, 인스턴스 ID 및 IP 주소)을 사용하여 액세스 RFC를 제출하고 새로 생성된 AMS 스택에 로그인하여 애플리케이션이 제대로 작동하는지 확인합니다. 자세한 내용은 인스턴스 액세스 요청을 참조하세요.

b. 만족하는 경우 시작된 인스턴스를 1티어 스택으로 계속 사용하거나 AMI를 사용하여 Auto Scaling 그룹을 포함한 추가 스택을 생성할 수 있습니다.

c. 마이그레이션이 만족스럽지 않은 경우 서비스 요청을 제출하고 스택 및 RFC IDs를 참조합니다. AMS는 사용자와 협력하여 문제를 해결합니다.

CloudEndure 랜딩 존 워크로드 수집 프로세스는 다음에 설명되어 있습니다.

워크로드 마이그레이션: CloudEndure 랜딩 존(SALZ)

이 섹션에서는 워크로드 수집(WIGS) RFC에 사용할 수 있도록 CloudEndure(CE) 전환 인스턴스에 대한 중간 마이그레이션 단일 계정 랜딩 존(SALZ)을 설정하는 방법에 대한 정보를 제공합니다.

CloudEndure에 대한 자세한 내용은 CloudEndure 마이그레이션을 참조하세요.

> **Note**
>
> 이는 사전 정의된 보안 강화 마이그레이션 LZ 및 패턴입니다.

사전 조건:

- 고객 AMS 계정
- AMS 계정과 고객 온프레미스 간의 네트워크 및 액세스 통합
- CloudEndure 계정
- CA 및/또는 CSDM으로 실행되는 AMS 보안 검토 및 승인을 위한 사전 승인 워크플로(예: IAM 사용자 영구 자격 증명을 오용하면 인스턴스 및 보안 그룹을 생성/삭제할 수 있음)

> **Note**
>
> 구체적인 준비 및 마이그레이션 프로세스는이 단원에서 설명합니다.

준비: 사용자 및 AMS 연산자:

1. 다음 리소스 및 업데이트를 위해 Management | Other | Other | Update change type to AMS를 사용하여 변경 요청(RFC)을 준비합니다. 별도의 기타 | 기타 업데이트 RFCs. 해당 RFC/CT에 대한 자세한 내용은 이러한 요청을 사용한 기타 | 기타 업데이트를 참조하세요.

   a. AMS VPC에 보조 CIDR 블록을 할당합니다. 마이그레이션이 완료된 후 제거될 임시 CIDR 블록입니다. 블록이 온프레미스 네트워크로 돌아가는 기존 경로와 충돌하지 않는지 확인합니다. 예를 들어 AMS VPC CIDR이 10.0.0.0/16이고 온프레미스 넷워드로 돌아가는 경로가 10.1.0.0/16인 경우 임시 보조 CIDR은 10.255.255.0/24일 수 있습니다. AWS CIDR 블록에 대한 자세한 내용은 VPC 및 서브넷 크기 조정을 참조하세요.

   b. 초기 가든 AMS VPC 내에 새 프라이빗 서브넷을 생성합니다. 예제 이름: `migration-temp-subnet`.

   c. 인스턴스 전환 및 가능한 중단 중에 소스 서버와의 충돌을 방지하려면 로컬 VPC 및 NAT(인터넷) 경로만 있는 서브넷에 대한 새 라우팅 테이블을 생성합니다. 패치 다운로드에 인터넷으로의 아웃바운드 트래픽이 허용되는지 확인하고 AMS WIGS 사전 조건을 다운로드하고 설치할 수 있도록 합니다.

   d. 와의 인바운드 및 아웃바운드 트래픽을 허용하도록 Managed AD 보안 그룹을 업데이트 합니다`migration-temp-subnet`. 또한 새로운 프라이빗 서브넷(예:)을 허용하도록 EPS 로드 밸런서(ELB) 보안 그룹(예: `mc-eps-McEpsElbPrivateSecurityGroup-M790XBZEEX74`)을 업데이트하도록 요청합니다. `migration-temp-subnet` 세 TCP 포트 모두에서 전용 CloudEndure(CE) 서브넷의 트래픽이 허용되지 않으면 WIGS 수집이 실패합니다.

e. 마지막으로 새 CloudEndure IAM 정책 및 IAM 사용자를 요청합니다. 정책에는 올바른 계정 번호가 필요하며, `RunInstances` 문의 서브넷 IDs는 <Customer Application Subnet(s) + Temp Migration Subnet>이어야 합니다.

AMS 사전 승인된 IAM CloudEndure 정책을 보려면: WIGS Cloud Endure Landing Zone 예제 파일의 압축을 풀고를 엽니다`customer_cloud_endure_policy.json`.

> **Note**
>
> 보다 허용적인 정책을 원하는 경우 CloudArchitect/CSDM과 필요한 사항에 대해 상의하고 필요한 경우 정책을 구현하는 RFC를 제출하기 전에 AMS 보안 검토 및 승인을 받습니다.

2. AMS 워크로드 수집에 CloudEndure를 사용하기 위한 준비 단계가 완료되고 마이그레이션 파트너가 준비 단계를 완료하면 마이그레이션을 수행할 준비가 된 것입니다. 마이그레이션 파트너가 WIGS RFC를 제출합니다.

> **Note**
>
> IAM 사용자 키는 직접 공유되지 않지만 화면 공유 세션에서 AMS 운영자가 CloudEndure 관리 콘솔에 입력해야 합니다.

준비: 마이그레이션 파트너 및 AMS 운영자:

1. CloudEndure 마이그레이션 프로젝트를 생성합니다.

   a. 프로젝트를 생성하는 동안 화면 공유 세션에 AMS 유형 입력 IAM 사용자 자격 증명이 있어야 합니다.

   b. 복제 설정 -> 복제 서버가 시작될 서브넷을 선택하고 customer-application-x 서브넷을 선택합니다.

   c. 복제 설정 -> 복제 서버에 적용할 보안 그룹을 선택하고 두 Sentinel 보안 그룹(프라이빗 전용 및 EgressAll)을 모두 선택합니다.

2. 시스템(인스턴스)에 대한 전환 옵션을 정의합니다.

   a. 서브넷: migration-temp-subnet.

   b. 보안 그룹: "Sentinel" 보안 그룹(프라이빗 전용 및 EgressAll).

전환 인스턴스는 AMS Managed AD 및 AWS 퍼블릭 엔드포인트와 통신할 수 있어야 합니다.

c. 탄력적 IP: 없음

d. 퍼블릭 IP: 아니요

e. IAM 역할: customer-mc-ec2-instance-profile

IAM 역할은 SSM 통신을 허용해야 합니다. AMS 기본값을 사용하는 것이 좋습니다.

f. 규칙에 따라 태그를 설정합니다.

마이그레이션: 마이그레이션 파트너:

1. AMS에서 더미 스택을 생성합니다. 스택 ID를 사용하여 Bastion에 액세스할 수 있습니다.
2. 소스 서버에 CloudEndure(CE) 에이전트를 설치합니다. 자세한 내용은 에이전트 설치를 참조하세요.
3. 소스 서버에서 로컬 관리자 자격 증명을 생성합니다.
4. 짧은 전환 기간을 예약하고 준비가 되면 전환을 클릭합니다. 이렇게 하면 마이그레이션이 완료되고 사용자를 대상 AWS 리전으로 리디렉션합니다.
5. 스택 더미 스택에 대한 관리자 액세스 권한을 요청하려면 관리자 액세스 요청을 참조하세요.
6. Bastion에 로그인한 다음 생성한 로컬 관리자 자격 증명을 사용하여 전환 인스턴스에 로그인합니다.
7. 페일세이프 AMI를 생성합니다. AMIs. https://docs.aws.amazon.com/managedservices/latest/ctref/ex-ami-create-col.html
8. 인스턴스 수집을 준비합니다. 단원을 참조하십시오워크로드 마이그레이션: Linux 및 Windows의 사전 조건.
9. 인스턴스에 대해 WIGS RFC를 실행합니다. 단원을 참조하십시오워크로드 수집 스택: 생성.

AMS 도구 계정(워크로드 마이그레이션)

다중 계정 랜딩 존 도구 계정(VPC 포함)은 마이그레이션 작업을 가속화하고, 보안 위치를 높이고, 비용과 복잡성을 줄이고, 사용 패턴을 표준화하는 데 도움이 됩니다.

도구 계정은 다음을 제공합니다.

- 프로덕션 워크로드 외부의 시스템 통합자를 위한 복제 인스턴스에 액세스하기 위한 잘 정의된 경계입니다.

- 격리된 체임버를 생성하여 다른 워크로드의 계정에 배치하기 전에 워크로드에 맬웨어 또는 알 수 없는 네트워크 경로가 있는지 확인할 수 있습니다.
- 정의된 계정 설정으로서 워크로드 마이그레이션을 위해 온보딩하고 설정하는 데 더 빠른 시간을 제공합니다.
- 격리된 네트워크는 온프레미스 -> CloudEndure -> 도구 계정 -> AMS 수집 이미지에서 트래픽을 보호하기 위해 라우팅됩니다. 이미지가 수집되면 AMS 관리 | 고급 스택 구성 요소 | AMI | 공유 (ct-1eiczxw8ihc18) RFC를 통해 대상 계정과 이미지를 공유할 수 있습니다.

상위 수준 아키텍처 다이어그램:

배포 | 관리형 랜딩 존 | 관리 계정 | 도구 계정 생성(VPC 사용) 변경 유형(ct-2j7q1hgf26x5c)을 사용하여 다중 계정 랜딩 존 환경 내에서 도구 계정을 빠르게 배포하고 워크로드 수집 프로세스를 인스턴스화합니다. 관리 계정, 도구 계정: 생성(VPC 사용)을 참조하세요.

> **Note**
>
> 마이그레이션 허브이므로 가용 영역(AZs)이 두 개 있는 것이 좋습니다.
> 기본적으로 AMS는 모든 계정에 다음과 같은 두 개의 보안 그룹(SGs)을 생성합니다. 이 두 SGs 있는지 확인합니다. 없는 경우 AMS 팀과 함께 새 서비스 요청을 열어 요청하세요.
>
> - SentinelDefaultSecurityGroupPrivateOnlyEgressAll
> - InitialGarden-SentinelDefaultSecurityGroupPrivateOnly
>
> 온프레미스로 돌아가는 경로가 있는 프라이빗 서브넷에서 CloudEndure 복제 인스턴스가 생성되었는지 확인합니다. 프라이빗 서브넷의 라우팅 테이블에 TGW로 돌아가는 기본 라우팅이 있는지 확인하여 확인할 수 있습니다. 그러나 CloudEndure 시스템 전환을 수행하면 온프레미스로 돌아가는 경로가 없는 "격리된" 프라이빗 서브넷으로 이동해야 하며 인터넷 아웃바운드 트래픽만 허용됩니다. 온프레미스 리소스에 대한 잠재적 문제를 방지하려면 격리된 서브넷에서 전환이 발생하도록 하는 것이 중요합니다.

사전 조건:

1. Plus 또는 Premium 지원 수준.
2. AMIs IDs입니다.
3. 앞서 설명한 대로 생성된 도구 계정입니다.

AWS Application Migration Service(AWS MGN)

AWS MGN(AWS Application Migration Service)은 도구 계정 프로비저닝 중에 자동으로 생성되는 `AWSManagedServicesMigrationRole` IAM 역할을 통해 MALZ 도구 계정에서 사용할 수 있습니다. AWS MGN을 사용하여 지원되는 버전의 Windows 및 Linux 운영 체제에서 실행되는 애플리케이션과 데이터베이스를 마이그레이션할 수 있습니다.

AWS 리전 지원에 대한 up-to-date 정보는 AWS 리전 서비스 목록을 참조하세요.

선호하는 AWS 리전 가 현재 AWS MGN에서 지원되지 않거나 애플리케이션이 실행되는 운영 체제가 현재 AWS MGN에서 지원되지 않는 경우 도구 계정에서 CloudEndure 마이그레이션을 대신 사용하는 것이 좋습니다.

AWS MGN 초기화 요청

AWS MGN은 처음 사용하기 전에 AMS로 초기화해야 합니다. 새 도구 계정에 대해 이를 요청하려면 도구 계정에서 다음 세부 정보와 함께 관리 | 기타 | 기타 RFC를 제출합니다.

```
RFC Subject=Please initialize AWS MGN in this account
RFC Comment=Please click 'Get started' on the MGN welcome page here:

    https://console.aws.amazon.com/mgn/home?region=MALZ_PRIMARY_REGION#/welcome using
 all default values
    to 'Create template' and complete the initialization process.
```

AMS가 RFC를 성공적으로 완료하고 도구 계정에서 AWS MGN을 초기화하면 `AWSManagedServicesMigrationRole`를 사용하여 요구 사항에 맞는 기본 템플릿을 편집할 수 있습니다.

새 AMS 도구 계정에 대한 액세스 활성화

도구 계정이 생성되면 AMS는 계정 ID를 제공합니다. 다음 단계는 새 계정에 대한 액세스를 구성하는 것입니다. 단계는 다음과 같습니다.

1. 적절한 Active Directory 그룹을 적절한 계정 IDs.

   새 AMS 생성 계정은 사용자가 RFC를 제출할 수 있도록 허용하는 역할뿐만 아니라 ReadOnly 역할 정책으로 프로비저닝됩니다. RFCs

   도구 계정에는 다음과 같은 추가 IAM 역할과 사용자가 있습니다.

- IAM 역할: `AWSManagedServicesMigrationRole`
- IAM 사용자: `customer_cloud_endure_user`

2. 서비스 통합 팀원이 다음 수준의 도구를 설정할 수 있도록 정책 및 역할을 요청합니다.

   AMS 콘솔로 이동하여 다음 RFCs를 제출합니다.

   a. KMS 키를 생성합니다. KMS 키 생성(자동) 또는 KMS 키 생성(검토 필요)을 사용합니다.

      KMS를 사용하여 수집된 리소스를 암호화할 때 다중 계정 랜딩 존 애플리케이션 계정의 나머지 부분과 공유되는 단일 KMS 키를 사용하면는 대상 계정에서 복호화할 수 있는 수집된 이미지에 대한 보안을 제공합니다.

   b. KMS 키를 공유합니다.

      관리 | 고급 스택 구성 요소 | KMS 키 | 공유(검토 필요) 변경 유형(ct-05yb337abq3x5)을 사용하여 수집된 AMIs가 상주할 애플리케이션 계정과 새 KMS 키를 공유하도록 요청합니다.

최종 계정 설정의 예제 그래픽:

AMS 사전 승인된 IAM CloudEndure 정책 예

AMS 사전 승인된 IAM CloudEndure 정책을 보려면: WIGS Cloud Endure Landing Zone 예제 파일의 압축을 풀고를 엽니다`customer_cloud_endure_policy.json`.

AMS 도구 계정 연결 및 end-to-end 설정 테스트

1. 먼저 CloudEndure를 구성하고 AMS에 복제할 서버에 CloudEndure 에이전트를 설치합니다.
2. CloudEndure에서 프로젝트를 생성합니다.
3. 암호 관리자를 통해 사전 조건을 수행할 때 공유된 AWS 자격 증명을 입력합니다.
4. 복제 설정에서:
   a. 복제 서버에 적용할 보안 그룹 선택 옵션에서 AMS "Sentinel" 보안 그룹(프라이빗 전용 및 EgressAll)을 모두 선택합니다.
   b. 머신(인스턴스)에 대한 전환 옵션을 정의합니다. 자세한 내용은 5단계를 참조하세요. 컷오버
   c. 서브넷: 프라이빗 서브넷입니다.
5. 보안 그룹:
   a. AMS "Sentinel" 보안 그룹(프라이빗 전용 및 EgressAll)을 모두 선택합니다.

b. 전환 인스턴스는 AMS 관리형 Active Directory(MAD) 및 AWS 퍼블릭 엔드포인트와 통신해야 합니다.

i. 탄력적 IP: 없음

ii. 퍼블릭 IP: 아니요

iii. IAM 역할: customer-mc-ec2-instance-profile

c. 내부 태그 지정 규칙에 따라 태그를 설정합니다.

6. 시스템에 CloudEndure 에이전트를 설치하고 EC2 콘솔의 AMS 계정에 표시될 복제 인스턴스를 찾습니다.

AMS 수집 프로세스:

AMS 도구 계정 위생

계정에서 AMI를 공유했으며 복제된 인스턴스가 더 이상 필요하지 않은 경우 정리해야 합니다.

- 사후 인스턴스 WIGs 수집:
  - 전환 인스턴스: AWS 콘솔을 통해 작업이 완료된 후 최소한이 인스턴스를 중지하거나 종료합니다.
  - 사전 수집 AMI 백업: 인스턴스가 수집되고 온프레미스 인스턴스가 종료되면 제거
  - AMS 수집 인스턴스: AMI가 공유되면 스택을 끄거나 종료합니다.
  - AMS 수집 AMIs: 대상 계정과의 공유가 완료되면 삭제
- 마이그레이션 정리 종료: 정기적으로 정리가 이루어지도록 개발자 모드를 통해 배포된 리소스를 문서화합니다. 예를 들면 다음과 같습니다.
  - 보안 그룹
  - 클라우드 형식을 통해 생성된 리소스
  - 네트워크 ACK
  - 서브넷
  - VPC
  - 라우팅 테이블
  - 역할
  - 사용자 및 계정

대규모 마이그레이션 - Migration Factory

AWS CloudEndure Migration Factory 솔루션 소개를 참조하세요.

워크로드 마이그레이션: Linux 사전 수집 검증

인스턴스가 AMS 계정으로 수집할 준비가 되었는지 확인할 수 있습니다. 워크로드 수집(WIGS) 사전 수집 검증은 운영 체제 유형, 사용 가능한 디스크 공간, 충돌하는 타사 소프트웨어의 존재 여부 등과 같은 검사를 수행합니다. 실행 시 WIGS 사전 수집 검증은 선택적 로그 파일이 있는 화면 테이블을 생성합니다. 결과는 실패 이유와 함께 각 검증 검사의 통과/실패 상태를 제공합니다. 또한 필요에 맞게 검증 테스트를 사용자 지정할 수 있습니다.

자주 묻는 질문:

- Linux WIGS 사전 수집 검증을 사용하려면 어떻게 해야 합니까?

  다음 단계에 따라 AMS Linux WIGS 사전 수집 검증 스크립트를 다운로드하고 사용합니다.

  1. 검증 스크립트를 사용하여 ZIP 파일 다운로드

     Linux WIGS 사전 수집 검증 zip 파일.
  2. 선택한 디렉터리에 연결된 규칙의 압축을 풉니다.
  3. readme.md 파일의 지침을 따릅니다.
- Linux WIGS 사전 수집 검증에서 수행하는 검증은 무엇입니까?

  AMS Linux WIGS 사전 수집 검증 솔루션은 다음을 검증합니다.

  1. 부팅 볼륨에는 최소 5기가바이트의 여유 공간이 있습니다.
  2. 운영 체제는 AMS에서 지원됩니다.
  3. 인스턴스에는 특정 인스턴스 프로파일이 있습니다.
  4. 인스턴스에는 바이러스 백신 소프트웨어 또는 가상화 소프트웨어가 포함되어 있지 않습니다.
  5. SSH가 올바르게 구성되었습니다.
  6. 인스턴스는 Yum 리포지토리에 액세스할 수 있습니다.
  7. 향상된 네트워킹 드라이버가 설치됩니다.
  8. 인스턴스에 SSM 에이전트가 있고 실행 중입니다.
- 사용자 지정 구성 파일을 지원하는 이유는 무엇입니까?

  스크립트는 온프레미스 물리적 서버와 AWS EC2 인스턴스 모두에서 실행되도록 설계되었습니다. 그러나 위 목록에 표시된 것처럼 일부 테스트는 온프레미스에서 실행될 때 실패합니다. 예를 들어 데

이터 센터의 물리적 서버에는 인스턴스 프로파일이 없습니다. 이러한 경우 구성 파일을 편집하여 인스턴스 프로파일 테스트를 건너뛰어 혼동을 방지할 수 있습니다.

- 스크립트의 최신 버전이 있는지 확인하려면 어떻게 해야 합니까?

  Linux WIGS 사전 수집 검증 솔루션의 up-to-date 버전은 기본 설명서 페이지의 AMS 헬퍼 파일 섹션에서 확인할 수 있습니다.

- 스크립트는 읽기 전용입니까?

  스크립트는 생성하는 로그 파일을 제외하고 읽기 전용으로 설계되었지만 비프로덕션 환경에서 스크립트를 실행하려면 모범 사례를 따라야 합니다.

- Windows에서 WIGS 사전 수집 검증을 사용할 수 있나요?

  예. 기본 설명서 페이지의 AMS 헬퍼 파일 섹션에서 사용할 수 있습니다.

워크로드 마이그레이션: Windows 사전 통합 검증

사전 WIGs기 스크립트를 사용하여 인스턴스를 AMS 계정으로 수집할 준비가 되었는지 확인할 수 있습니다. 워크로드 수집(WIGS) 사전 수집 검증은 운영 체제 유형, 사용 가능한 디스크 공간, 충돌하는 타사 소프트웨어의 존재 여부 등과 같은 검사를 수행합니다. 실행 시 WIGS 사전 수집 검증은 화면 테이블과 선택적 로그 파일을 생성합니다. 결과는 각 검증 검사의 통과/실패 상태와 실패 이유를 제공합니다. 또한 검증 테스트를 사용자 지정할 수 있습니다.

자주 묻는 질문:

- Windows WIGS 사전 수집 검증을 사용하려면 어떻게 해야 합니까?

  GUI 및 웹 브라우저에서 검증을 실행하거나 Windows PowerShell, SSM Run Command 또는 SSM Session Manager를 사용할 수 있습니다.

  옵션 1: GUI 및 웹 브라우저에서 실행

  GUI 및 웹 브라우저에서 Windows WIGs 검증을 실행하려면 다음을 수행합니다.

  1. 검증 스크립트가 포함된 ZIP 파일을 다운로드합니다.

     Windows WIGS 사전 수집 검증 ZIP 파일.
  2. 선택한 디렉터리에 연결된 규칙의 압축을 풉니다.
  3. README.md 파일의 지침을 따릅니다.

  옵션 2: Windows PowerShell, SSM Run Command 또는 SSM 세션 관리자에서 실행

Windows 2016 이상

1. 검증 스크립트와 함께 ZIP 파일을 다운로드합니다.

```
$DestinationFile = "$env:TEMP\WIGValidation.zip"

$Bucket = 'https://docs.aws.amazon.com/managedservices/latest/appguide/samples/
windows-prewigs-validation.zip'
$DestinationFile = "$env:TEMP\WIGValidation.zip"
$ScriptFolder = "$env:TEMP\AWSManagedServices.PreWigs.Validation"
```

2. 에서 기존 파일을 제거합니다`C:\Users\AppData\Local\Temp \AWSManagedServices.PreWigs.Validation`.

```
Remove-Item $scriptFolder -Recurse -Force -ErrorAction Ignore
```

3. 스크립트를 호출합니다.

```
Invoke-WebRequest -Uri $bucket -OutFile $DestinationFile
Add-Type -Assembly "system.io.compression.filesystem"
```

4. 선택한 디렉터리에 연결된 파일의 압축을 폽니다.

```
[io.compression.zipfile]::ExtractToDirectory($DestinationFile, $env:TEMP)
```

5. 대화형으로 검증 스크립트를 실행하고 결과를 확인합니다.

```
Import-Module .\AWSManagedServices.PreWigs.Validation.psm1 -force
Invoke-PreWIGsValidation -RunWithoutExitCodes
```

6. (선택 사항) 종료 코드 섹션에 나열된 오류 코드를 캡처하려면 `RunWithoutExitCodes` 옵션 없이 스크립트를 실행합니다. 이 명령은 활성 PowerShell 세션을 종료합니다.

```
Import-Module .\AWSManagedServices.PreWigs.Validation.psm1 -force
Invoke-PreWIGsValidation
```

Windows 2012 R2 이하

Windows Server 2012R2 이하를 실행하는 경우 zip 파일을 다운로드하기 전에 TLS를 설정해야 합니다. TLS를 설정하려면 다음 단계를 완료합니다.

1. 검증 스크립트와 함께 ZIP 파일을 다운로드합니다.

```
$DestinationFile = "$env:TEMP\WIGValidation.zip"

$Bucket = 'https://docs.aws.amazon.com/managedservices/latest/appguide/samples/
windows-prewigs-validation.zip'
$DestinationFile = "$env:TEMP\WIGValidation.zip"
$ScriptFolder = "$env:TEMP\AWSManagedServices.PreWigs.Validation"
```

2. 기존 검증 파일이 있는 경우 제거합니다.

```
Remove-Item $scriptFolder -Recurse -Force -ErrorAction Ignore
```

3. TLS 버전을 설정합니다.

```
[System.Net.ServicePointManager]::SecurityProtocol = 'TLS12'
```

4. WIG 검증을 다운로드합니다.

```
Invoke-WebRequest -Uri $bucket -OutFile $DestinationFile
Add-Type -Assembly "system.io.compression.filesystem"
```

5. 선택한 디렉터리에 연결된 규칙의 압축을 폽니다.

```
[io.compression.zipfile]::ExtractToDirectory($DestinationFile, $env:TEMP)
```

6. 대화형으로 검증 스크립트를 실행하고 결과를 확인합니다.

```
Import-Module .\AWSManagedServices.PreWigs.Validation.psm1 -force
Invoke-PreWIGsValidation -RunWithoutExitCodes
```

7. (선택 사항) 종료 코드 섹션에 나열된 오류 코드를 캡처하려면 RunWithoutExitCodes 옵션 없이 스크립트를 실행합니다. 이 명령은 활성 PowerShell 세션을 종료합니다.

```
Import-Module .\AWSManagedServices.PreWigs.Validation.psm1 -force
Invoke-PreWIGsValidation
```

> **Note**
>
> PowerShell 스크립트를 다운로드하여 실행할 수 있습니다. 이렇게 하려면 pre-wigs-validation-powershell-scripts.zip을 다운로드합니다.

- Windows WIGS 수집 전 검증에서 수행하는 검증은 무엇입니까?

  AMS Windows WIGS 사전 수집 검증 솔루션은 다음을 검증합니다.

  1. 부팅 볼륨에는 최소 10기가바이트의 여유 공간이 있습니다.
  2. 운영 체제는 AMS에서 지원됩니다.
  3. 인스턴스에는 특정 인스턴스 프로파일이 있습니다.
  4. 인스턴스에는 바이러스 백신 소프트웨어 또는 가상화 소프트웨어가 포함되어 있지 않습니다.
  5. DHCP는 하나 이상의 네트워크 어댑터에서 활성화됩니다.
  6. 인스턴스가 Sysprep 준비가 되었습니다.
     - 2008 R2 및 2012 Base 및 R2의 경우 Sysprep은 다음을 확인합니다.
       - unattend.xml 파일이 있습니다.
       - sppnp.dll 파일(있는 경우)이 손상되지 않았습니다.
       - 운영 체제가 업그레이드되지 않았습니다.
       - Sysprep이 Microsoft 지침에 따른 최대 횟수를 초과하여 실행되지 않음
     - 2016년 이상에서는 해당 OS에 문제가 발생하지 않으므로 위의 모든 검사를 건너뜁니다.
  7. Windows 관리 계측(WMI) 하위 시스템은 정상입니다.
  8. 필요한 드라이버가 설치됩니다.
  9. SSM 에이전트 및가 설치되고 실행 중입니다.
  10. RDS 라이선스 구성으로 인해 머신이 유예 기간 중인지 확인하는 경고가 표시됩니다.
  11. 필수 레지스트리 키가 올바르게 설정되었습니다. 자세한 내용은 사전 수집 검증 zip 파일의 README를 참조하세요.

- 사용자 지정 구성 파일을 지원하는 이유는 무엇입니까?

  스크립트는 온프레미스 물리적 서버와 AWS EC2 인스턴스 모두에서 실행되도록 설계되었습니다. 그러나 위 목록에 표시된 것처럼 일부 테스트는 온프레미스에서 실행될 때 실패합니다. 예를 들어 데이터 센터의 물리적 서버에는 인스턴스 프로파일이 없습니다. 이러한 경우 구성 파일을 편집하여 인스턴스 프로파일 테스트를 건너뛰어 혼동을 방지할 수 있습니다.

- 스크립트의 최신 버전이 있는지 확인하려면 어떻게 해야 합니까?

  Windows WIGS 사전 수집 검증 솔루션의 up-to-date 버전은 기본 설명서 페이지의 AMS 헬퍼 파일 섹션에서 확인할 수 있습니다.

- 스크립트는 읽기 전용입니까?

스크립트는 생성하는 로그 파일을 제외하고 읽기 전용으로 설계되었지만 비프로덕션 환경에서 스크립트를 실행하려면 모범 사례를 따라야 합니다.

- Linux에서 WIGS 사전 수집 검증을 사용할 수 있나요?

  예. Linux 버전은 2019년 10월 31일에 출시되었습니다. 기본 설명서 페이지의 AMS 헬퍼 파일 섹션에서 사용할 수 있습니다.

워크로드 수집 스택: 생성

콘솔을 사용하여 인스턴스를 AMS 스택으로 마이그레이션

AMS 콘솔에서이 변경 유형의 스크린샷:

작동 방식:

1. RFC 생성 페이지로 이동합니다. AMS 콘솔의 왼쪽 탐색 창에서 RFCs 클릭하여 RFCs 목록 페이지를 연 다음 RFC 생성을 클릭합니다.
2. 기본 변경 유형 찾아보기 보기에서 널리 사용되는 변경 유형(CT)을 선택하거나 범주별 선택 보기에서 CT를 선택합니다.
   - 변경 유형별 찾아보기: 빠른 생성 영역에서 인기 있는 CT를 클릭하여 RFC 실행 페이지를 즉시 열 수 있습니다. 빠른 생성으로 이전 CT 버전을 선택할 수 없습니다.

     CTs 정렬하려면 카드 또는 테이블 보기에서 모든 변경 유형 영역을 사용합니다. 어느 보기에서든 CT를 선택한 다음 RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다. 해당하는 경우 RFC 생성 버튼 옆에 이전 버전으로 생성 옵션이 나타납니다.
   - 범주별 선택: 범주, 하위 범주, 항목 및 작업을 선택하면 해당하는 경우 이전 버전으로 생성 옵션이 있는 CT 세부 정보 상자가 열립니다. RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다.
3. RFC 실행 페이지에서 CT 이름 영역을 열어 CT 세부 정보 상자를 확인합니다. 제목은 필수입니다(변경 유형 찾아보기 보기에서 CT를 선택하면 입력됨). 추가 구성 영역을 열어 RFC에 대한 정보를 추가합니다.

   실행 구성 영역에서 사용 가능한 드롭다운 목록을 사용하거나 필요한 파라미터의 값을 입력합니다. 선택적 실행 파라미터를 구성하려면 추가 구성 영역을 엽니다.
4. 완료되면 실행을 클릭합니다. 오류가 없는 경우 RFC가 성공적으로 생성된 페이지에 제출된 RFC 세부 정보와 초기 실행 출력이 표시됩니다.

5. 실행 파라미터 영역을 열어 제출한 구성을 확인합니다. 페이지를 새로 고쳐 RFC 실행 상태를 업데이트합니다. 선택적으로 RFC를 취소하거나 페이지 상단의 옵션을 사용하여 RFC 사본을 생성합니다.

> **Note**
>
> RFC가 거부되면 실행 출력에 Amazon CloudWatch logs에 대한 링크가 포함됩니다. AMS 워크로드 수집(WIGS) RFCs는 요구 사항이 충족되지 않을 때, 예를 들어 인스턴스에서 바이러스 백신 소프트웨어가 감지될 때 거부됩니다. CloudWatch 로그에는 실패한 요구 사항과 문제 해결을 위해 수행할 작업에 대한 정보가 포함됩니다.

CLI를 사용하여 인스턴스를 AMS 스택으로 마이그레이션

작동 방식:

1. 인라인 생성(모든 RFC 및 실행 파라미터가 포함된 `create-rfc` 명령을 실행) 또는 템플릿 생성(2개의 JSON 파일을 생성, 하나는 RFC 파라미터용이고 다른 하나는 실행 파라미터용)을 사용하고 두 파일을 입력으로 사용하여 `create-rfc` 명령을 실행합니다. 두 방법 모두 여기에 설명되어 있습니다.
2. 반환된 RFC ID로 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 명령을 제출합니다.

   RFC: `aws amscm get-rfc --rfc-id` *`ID`* 명령을 모니터링합니다.

변경 유형 버전을 확인하려면 다음 명령을 사용합니다.

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 변경 유형에 대한 스키마의 일부인지 여부에 관계없이 모든 `CreateRfc` 파라미터를 RFC와 함께 사용할 수 있습니다. 예를 들어 RFC 상태가 변경될 때 알림을 받으려면 요청의 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` RFC 파라미터 부분(실행 파라미터 아님)에이 줄을 추가합니다. 모든 CreateRfc 파라미터 목록은 AMS Change Management API 참조를 참조하세요.

AMS CLI를 사용하여 AMS 계정으로 마이그레이션된 비 AMS 인스턴스에서 AMS 인스턴스를 생성할 수 있습니다.

> **Note**
>
> 사전 조건을 따랐는지 확인하세요. 워크로드 마이그레이션: Linux 및 Windows의 사전 조건을 참조하세요.

변경 유형 버전을 확인하려면 다음 명령을 사용합니다.

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

인라인 생성:

인라인으로 제공된 실행 파라미터(실행 파라미터를 인라인으로 제공할 때 따옴표 이스케이프)로 RFC 생성 명령을 실행한 다음 반환된 RFC ID를 제출합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
aws amscm create-rfc --change-type-id "ct-257p9zjk14ija" --change-type-version "2.0" --
title "AMS-WIG-TEST-NO-ACTION" --execution-parameters "{\"InstanceId\":\"INSTANCE_ID\",
\"TargetVpcId\":\"VPC_ID\",\"TargetSubnetId\":\"SUBNET_ID\",\"TargetInstanceType\":
\"t2.large\",\"ApplyInstanceValidation\":true,\"Name\":\"WIG-TEST\",\"Description\":
\"WIG-TEST\",\"EnforceIMDSV2\":\"false\"}"
```

템플릿 생성:

1. 0이 변경 유형에 대한 실행 파라미터 JSON 스키마를 파일로 입력합니다. 예제 이름은 MigrateStackParams.json:입니다.

   ```
   aws amscm get-change-type-version --change-type-id "ct-257p9zjk14ija" --query
    "ChangeTypeVersion.ExecutionInputSchema" --output text > MigrateStackParams.json
   ```

2. 실행 파라미터 JSON 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

   ```
   {
   "InstanceId":          "MIGRATED_INSTANCE_ID",
   ```

```
"TargetVpcId":          "VPC_ID",
"TargetSubnetId":       "SUBNET_ID",
"Name":                 "Migrated-Stack",
"Description":          "Create-Migrated-Stack",
"EnforceIMDSV2":        "false"
}
```

3. RFC 템플릿 JSON 파일을 출력합니다. 예제 이름은 MigrateStackRfc.json:입니다.

```
aws amscm create-rfc --generate-cli-skeleton > MigrateStackRfc.json
```

4. MigrateStackRfc.json 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
{
"ChangeTypeId":          "ct-257p9zjk14ija",
"ChangeTypeVersion":     "2.0",
"Title":                 "Migrate-Stack-RFC"
}
```

5. MigrateStackRfc 파일과 MigrateStackParams 파일을 지정하여 RFC를 생성합니다.

```
aws amscm create-rfc --cli-input-json file://MigrateStackRfc.json  --execution-
parameters file://MigrateStackParams.json
```

응답에서 새 RFC의 ID를 수신하고 이를 사용하여 RFC를 제출하고 모니터링할 수 있습니다. 제출하기 전까지는 RFC가 편집 상태로 유지되고 시작되지 않습니다.

새 인스턴스가 관련 VPC의 애플리케이션 소유자 계정에 대한 인스턴스 목록에 나타납니다.

6. RFC가 성공적으로 완료되면 애플리케이션 소유자에게 알려 새 인스턴스에 로그인하고 워크로드가 작동하는지 확인할 수 있도록 합니다.

> **Note**
>
> RFC가 거부되면 실행 출력에 Amazon CloudWatch logs에 대한 링크가 포함됩니다. AMS 워크로드 수집(WIGS) RFCs는 요구 사항이 충족되지 않을 때, 예를 들어 인스턴스에서 바이러스 백신 소프트웨어가 감지될 때 거부됩니다. CloudWatch 로그에는 실패한 요구 사항과 문제 해결을 위해 수행할 작업에 대한 정보가 포함됩니다.

팁

> **Note**
>
> 사전 조건을 따랐는지 확인하세요. 워크로드 마이그레이션: Linux 및 Windows의 사전 조건을 참조하세요.

> **Note**
>
> 마이그레이션 중인 인스턴스의 태그에 RFC에 제공된 태그와 동일한 키가 있는 경우 RFC가 실패합니다.

> **Note**
>
> 최대 4개의 대상 IDs, 포트 및 가용 영역을 지정할 수 있습니다.

> **Note**
>
> RFC가 거부되면 실행 출력에 Amazon CloudWatch logs에 대한 링크가 포함됩니다. AMS 워크로드 수집(WIGS) RFCs는 요구 사항이 충족되지 않을 때, 예를 들어 인스턴스에서 바이러스 백신 소프트웨어가 감지될 때 거부됩니다. CloudWatch 로그에는 실패한 요구 사항과 문제 해결을 위해 수행할 작업에 대한 정보가 포함됩니다.

> **Note**
>
> RFC가 거부되면 실행 출력에 Amazon CloudWatch logs에 대한 링크가 포함됩니다. AMS 워크로드 수집(WIGS) RFCs는 요구 사항이 충족되지 않을 때, 예를 들어 인스턴스에서 바이러스 백신 소프트웨어가 감지될 때 거부됩니다. CloudWatch 로그에는 실패한 요구 사항과 문제 해결을 위해 수행할 작업에 대한 정보가 포함됩니다.

필요한 경우 워크로드 수집(WIGS) 실패를 참조하세요.

AMS CloudFormation 수집

AMS AWS CloudFormation 수집 변경 유형(CT)을 사용하면 일부 수정 사항과 함께 기존 CloudFormation 템플릿을 사용하여 AMS 관리형 VPC에 사용자 지정 스택을 배포할 수 있습니다.

주제

- AWS CloudFormation 수집 지침, 모범 사례 및 제한 사항
- AWS CloudFormation 수집: 예
- CloudFormation 수집 스택 생성
- AWS CloudFormation 수집 스택 업데이트
- CloudFormation 수집 스택 변경 세트 승인
- AWS CloudFormation 스택 종료 방지 업데이트
- AMS에서 CFN 수집 또는 스택 업데이트 CTs 사용한 자동화된 IAM 배포

AMS AWS CloudFormation 수집 프로세스에는 다음이 포함됩니다.

- 사용자 지정 CloudFormation 템플릿을 준비하여 S3 버킷에 업로드하거나 RFC를 생성할 때 템플릿을 인라인으로 제공합니다. 미리 서명된 URL이 있는 S3 버킷을 사용하는 경우 자세한 내용은 presign을 참조하세요.
- CloudFormation 수집 변경 유형을 RFC의 AMS에 제출합니다. CFN 수집 변경 유형 연습은 섹션을 참조하세요CloudFormation 수집 스택 생성. CFN 수집 예제는 섹션을 참조하세요AWS CloudFormation 수집: 예.
- 스택이 생성되면 스택을 업데이트하고 스택의 드리프트를 수정할 수 있습니다. 또한 업데이트가 실패하면 업데이트를 명시적으로 승인하고 구현할 수 있습니다. 이 단원에서는 이러한 모든 절차를 설명합니다.

  CFN 드리프트 감지에 대한 자세한 내용은 신규 - CloudFormation 드리프트 감지를 참조하세요.

> **Note**
>
> - 이제이 변경 유형에 버전 2.0이 있습니다. 버전 2.0은 자동화되며 수동으로 실행되지 않습니다. 이렇게 하면 CT 실행 속도가 더 빨라집니다. 이 버전에는 사용자 지정 CloudFormation 템플릿을 RFC에 붙여 넣을 수 있는 CloudFormationTemplate과 AMS 다중 계정 랜딩 존에

서 CloudFormation 수집을 사용할 수 있는 VpcId라는 두 가지 새로운 파라미터가 도입되었습니다. CloudFormation

- 버전 1.0은 수동 변경 유형입니다. 즉, AMS 연산자는 변경 유형을 성공적으로 종료하기 전에 몇 가지 조치를 취해야 합니다. 최소한 검토가 필요합니다. 또한이 버전에서는 CloudFormationTemplateS3Endpoint 파라미터 값이 미리 서명된 URL이어야 합니다.

AWS CloudFormation 수집 지침, 모범 사례 및 제한 사항

AMS가 CloudFormation 템플릿을 처리하려면 몇 가지 지침과 제한 사항이 있습니다.

지침

AWS CloudFormation 수집을 수행하는 동안 AWS CloudFormation 오류를 줄이려면 다음 지침을 따르세요.

- 템플릿에 자격 증명 또는 기타 민감한 정보를 임베드하지 마세요 - CloudFormation 템플릿은 AWS CloudFormation 콘솔에 표시되므로 템플릿에 자격 증명 또는 민감한 데이터를 임베드하지 않으려고 합니다. 템플릿에는 민감한 정보가 포함될 수 없습니다. 다음 리소스는 값에 AWS Secrets Manager를 사용하는 경우에만 허용됩니다.
  - `AWS::RDS::DBInstance` - [MasterUserPassword,TdeCredentialPassword]
  - `AWS::RDS::DBCluster` - [MasterUserPassword]
  - `AWS::ElastiCache::ReplicationGroup` - [AuthToken]

  > **Note**
  >
  > 리소스 속성에서 AWS Secrets Manager 보안 암호를 사용하는 방법에 대한 자세한 내용은 AWS CloudFormation 템플릿을 사용하여 AWS Secrets Manager에서 관리되는 보안 암호를 생성하고 검색하는 방법 및 동적 참조를 사용하여 템플릿 값 지정을 참조하세요.

- Amazon RDS 스냅샷을 사용하여 RDS DB 인스턴스 생성 - 이렇게 하면 MasterUserPassword를 제공할 필요가 없습니다.
- 제출하는 템플릿에 IAM 인스턴스 프로파일이 포함된 경우 'customer' 접두사가 붙어야 합니다. 예를 들어 이름이 'example-instance-profile'인 인스턴스 프로파일을 사용하면 오류가 발생합니다. 대신 이름이 'customer-example-instance-profile'인 인스턴스 프로파일을 사용합니다.

- - [UserData]에 민감한 데이터를 포함하지 마세요**`AWS::EC2::Instance`**. UserData에는 암호, API 키 또는 기타 민감한 데이터가 포함되어서는 안 됩니다. 이러한 유형의 데이터는 암호화하여 S3 버킷에 저장하고 UserData를 사용하여 인스턴스에 다운로드할 수 있습니다.
- CloudFormation 템플릿을 사용한 IAM 정책 생성은 제약 조건과 함께 지원됩니다. IAM 정책은 AMS SecOps에서 검토하고 승인해야 합니다. 현재는 사전 승인된 권한이 포함된 인라인 정책이 있는 IAM 역할 배포만 지원합니다. 다른 경우에는 AMS SecOps 프로세스를 재정의하기 때문에 CloudFormation 템플릿을 사용하여 IAM 정책을 생성할 수 없습니다.
- SSH KeyPairs는 지원되지 않습니다. Amazon EC2 인스턴스는 AMS 액세스 관리 시스템을 통해 액세스해야 합니다. AMS RFC 프로세스가 사용자를 인증합니다. SSH 키 페어를 생성하고 AMS 액세스 관리 모델을 재정의할 권한이 없으므로 CloudFormation 템플릿에 SSH 키 페어를 포함할 수 없습니다.
- 보안 그룹 수신 규칙이 제한됨 - 소스 CIDR 범위가 0.0.0.0/0이거나 TCP 포트가 80 또는 443이 아닌 공개적으로 라우팅 가능한 주소 공간을 가질 수 없습니다.
- CloudFormation 리소스 템플릿 작성 시 AWS CloudFormation 지침 준수 - 해당 리소스에 대한 AWS CloudFormation 사용 설명서를 참조하여 리소스에 올바른 데이터 유형/속성 이름을 사용해야 합니다. 예를 들어 AWS::EC2::Instance 리소스에서 SecurityGroupIds 속성의 데이터 형식은 '문자열 값 목록'이므로 ["sg-aaaaaaa"]는 괜찮지만 (괄호 포함) "sg-aaaaaaaaa"는 그렇지 않습니다(괄호 제외).

  자세한 내용은 AWS 리소스 및 속성 유형 참조를 참조하세요.
- AMS CloudFormation 수집 CT에 정의된 파라미터를 사용하도록 사용자 지정 CloudFormation 템플릿 구성 - AMS CloudFormation 수집 CT에 정의된 파라미터를 사용하도록 CloudFormation 템플릿을 구성할 때 관리 | 사용자 지정 스택 | CloudFormation 템플릿의 스택 | CT 업데이트 (ct-361tlo1k7339x)를 사용하여 CT 입력에 변경된 파라미터 값과 함께 CloudFormation 템플릿을 제출하여 유사한 스택을 생성할 수 있습니다. 예제는 AWS CloudFormation 수집 예제: 리소스 정의 섹션을 참조하세요.
- 미리 서명된 URL이 있는 Amazon S3 버킷 엔드포인트는 만료될 수 없음 - 미리 서명된 URL이 있는 Amazon S3 버킷 엔드포인트를 사용하는 경우 미리 서명된 Amazon S3 URL이 만료되지 않았는지 확인합니다. 만료된 미리 서명된 Amazon S3 버킷 URL로 제출된 CloudFormation 수집 RFC가 거부됩니다.
- 대기 조건에는 신호 로직 필요 - 대기 조건은 스택 리소스 생성을 스택 생성 외부의 구성 작업과 조정하는 데 사용됩니다. 템플릿에서 Wait Condition 리소스를 사용하는 경우는 성공 신호를 AWS CloudFormation 기다리고 성공 신호 수가 생성되지 않으면 스택 생성이 실패로 표시됩니다. Wait Condition 리소스를 사용하는 경우 신호에 대한 로직이 있어야 합니다. 자세한 내용은 템플릿에서 대기 조건 생성을 참조하세요.

모범 사례

다음은 AMS AWS CloudFormation 수집 프로세스를 사용하여 리소스를 마이그레이션하는 데 사용할 수 있는 몇 가지 모범 사례입니다.

- 하나의 CT에서 IAM 및 기타 정책 관련 리소스 제출 - CloudFormation Ingest와 같은 자동 CTs를 사용하여 IAM 역할을 배포할 수 있는 경우 그렇게 하는 것이 좋습니다. 다른 경우에는 모든 IAM 또는 기타 정책 관련 리소스를 수집하여 단일 관리 | 기타 | 기타 | 변경 유형 생성(ct-1e1xtak34nx76)에 제출하는 것이 좋습니다. 예를 들어 필요한 모든 IAM 역할, IAM Amazon EC2 인스턴스 프로파일, 기존 IAM 역할에 대한 IAM 정책 업데이트, Amazon S3 버킷 정책, Amazon SNS/Amazon SQS 정책 등을 결합하고 ct-1e1xtak34nx76 RFC를 제출하여 이러한 기존 리소스를 향후 CloudFormation 수집 템플릿 내에서 간단히 참조할 수 있도록 합니다.
- EC2 인스턴스는 부트스트래핑되어 도메인에 성공적으로 조인됩니다. 이는 모범 사례로 자동으로 수행됩니다. CloudFormation 수집 스택을 통해 시작된 Amazon EC2 인스턴스가 부트스트랩되고 도메인에 성공적으로 조인되도록 AMS에는 Auto Scaling 그룹 리소스에 대한 CreationPolicy 및 UpdatePolicy가 포함됩니다(즉, 이러한 정책이 아직 없는 경우).
- Amazon RDS DB 인스턴스 파라미터를 지정해야 함 AWS CloudFormation - 수집을 통해 Amazon RDS 데이터베이스를 생성할 때 이전 DB 스냅샷에서 복원하려면 `DBSnapshotIdentifier` 파라미터를 지정해야 합니다. 수집은 현재 민감한 데이터를 처리하지 AWS CloudFormation 않기 때문에 필요합니다.

AMS CloudFormation 템플릿 수집에 CloudFormation 템플릿을 사용하는 방법의 예는 섹션을 참조하세요AWS CloudFormation 수집: 예.

템플릿 검증

AMS에 제출하기 전에 CloudFormation 템플릿을 자체 검증할 수 있습니다.

AMS AWS CloudFormation 수집에 제출된 템플릿은 AMS 계정 내에서 안전하게 배포할 수 있도록 검증됩니다. 검증 프로세스는 다음을 확인합니다.

- 지원되는 리소스 AWS CloudFormation - AMS 수집 지원 리소스만 사용됩니다. 자세한 내용은 지원되는 리소스 단원을 참조하십시오.
- 지원되는 AMIs- 템플릿의 AMI는 AMS 지원 AMI입니다. AMS AMIsAMS Amazon Machine Image(AMIs).
- AMS 공유 서비스 서브넷 - 템플릿이 AMS 공유 서비스 서브넷으로 리소스를 시작하려고 시도하지 않습니다.

- 리소스 정책 - 공개적으로 읽을 수 있거나 쓸 수 있는 S3 버킷 정책과 같은 지나치게 허용적인 리소스 정책은 없습니다. AMS는 공개적으로 읽거나 쓸 수 있는 S3 버킷을 허용하지 않습니다 AWS 계정.

AWS CloudFormation Linter로 검증

AWS CloudFormation Linter 도구를 사용하여 AMS에 제출하기 전에 CloudFormation 템플릿을 자체 검증할 수 있습니다.

AWS CloudFormation Linter 도구는 리소스/속성 이름, 데이터 유형 및 함수에 대한 검증을 제공하므로 CloudFormation 템플릿을 검증하는 가장 좋은 방법입니다. 자세한 내용은 aws-cloudformation/cfn-python-lint를 참조하세요.

이전에 표시된 템플릿의 AWS CloudFormation Linter 출력은 다음과 같습니다.

```
$ cfn-lint -t ./testtmpl.json
E3002 Invalid Property Resources/SNSTopic/Properties/Name
./testtmpl.json:6:9
```

CloudFormation 템플릿의 오프라인 검증을 지원하기 위해 AMS는 AWS CloudFormation Linter 도구에 대한 플러그형 사용자 지정 검증 규칙 세트를 개발했습니다. AMS 콘솔의 개발자 리소스 페이지에 있습니다.

AWS CloudFormation 사전 수집 검증 스크립트를 사용하려면 다음 단계를 따르세요.

1. AWS CloudFormation Linter 도구를 설치합니다. 설치 지침은 aws-cloudformation/cfn-lint를 참조하세요.
2. 검증 스크립트가 포함된 .zip 파일을 다운로드합니다.

   CFN Lint 사용자 지정 규칙.
3. 선택한 디렉터리에 연결된 규칙의 압축을 풉니다.
4. 다음 명령을 실행하여 CloudFormation 템플릿을 검증합니다.

```
cfn-lint --template {TEMPLATE_FILE} --append-rules {DIRECTORY_WITH_CUSTOM_RULES}
```

CloudFormation 수집 스택: CFN 검사기 예제

이 예제는 성공적인 수집을 위해 템플릿을 준비하는 데 도움이 될 수 있습니다.

형식 검증

템플릿에 "리소스" 섹션이 포함되어 있고 템플릿 아래에 정의된 모든 리소스에 "유형" 값이 있는지 확인합니다.

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description" : "Create a SNS topic",
  "Resources": {
    "SnsTopic": {
      "Type": "AWS::SNS::Topic"
    }
  }
}
```

템플릿의 루트 키가 허용되는지 확인합니다. 허용되는 루트 키는 다음과 같습니다.

```
[
  "AWSTemplateFormatVersion",
  "Description",
  "Mappings",
  "Parameters",
  "Conditions",
  "Resources",
  "Rules",
  "Outputs",
  "Metadata"
]
```

수동 검토 필수 검증

템플릿에 다음 리소스가 포함된 경우 자동 검증이 실패하고 수동 검토가 필요합니다.

표시된 정책은 보안 관점에서 고위험 영역입니다. 예를 들어 특정 사용자 또는 그룹을 제외한 모든 사용자가 객체를 생성하거나 권한을 쓸 수 있도록 허용하는 S3 버킷 정책은 매우 위험합니다. 따라서 정책을 검증하고 내용을 기반으로 승인 또는 거부하며 이러한 정책은 자동으로 생성할 수 없습니다. 이 문제를 해결하기 위해 가능한 접근 방식을 조사하고 있습니다.

현재 다음 리소스에 대한 자동 검증은 없습니다.

```
[
    "S3::BucketPolicy",
```

```
    "SNS::TopicPolicy",
    "SQS::QueuePolicy"
]
```

파라미터 확인

템플릿 파라미터에 값이 제공되지 않은 경우 기본값이 있어야 합니다.

리소스 속성 검증

필수 속성 검사: 특정 리소스 유형에 대해 특정 속성이 있어야 합니다.

- 에 "VPCOptions"가 있어야 합니다. `AWS::OpenSearch::Domain`
- "CludsterSubnetGroupName"이에 있어야 합니다. `AWS::Redshift::Cluster`

```
{
    "AWS::OpenSearch::Domain": [
      "VPCOptions"
    ],
    "AWS::Redshift::Cluster": [
      "ClusterSubnetGroupName"
    ]
}
```

허용되지 않는 속성 검사: 특정 리소스 유형에는 특정 속성이 *not* 존재해야 합니다.

- "AWS::SecretsManager::Secret"에 "SecretString"이 없어야 합니다.AWS::SecretsManager::Secret
- “AWS::DMS::EndpointMongoDbSettings”가 없어야 합니다.

```
{
  "AWS::SecretsManager::Secret": [
    "SecretString"
  ],
  "AWS::DMS::Endpoint": [
    "MongoDbSettings"
  ]
}
```

SSM 파라미터 검사: 다음 목록의 속성의 경우 Secrets Manager 또는 Systems Manager Parameter Store(Secure String Parameter)를 통해 값을 지정해야 합니다.

```
{
  "RDS::DBInstance": [
    "MasterUserPassword",
    "TdeCredentialPassword"
  ],
  "RDS::DBCluster": [
    "MasterUserPassword"
  ],
  "ElastiCache::ReplicationGroup": [
    "AuthToken"
  ],
  "DMS::Certificate": [
    "CertificatePem",
    "CertificateWallet"
  ],
  "DMS::Endpoint": [
    "Password"
  ],
  "CodePipeline::Webhook": {
    "AuthenticationConfiguration": [
        "SecretToken"
    ]
  },
  "DocDB::DBCluster": [
    "MasterUserPassword"
  ]
},
```

일부 속성은 특정 패턴을 준수해야 합니다. 예를 들어 IAM 인스턴스 프로파일 이름은 AMS 예약 접두사로 시작해서는 안 되며 속성 값은 다음과 같이 특정 정규식과 일치해야 합니다.

```
{
    "AWS::EC2::Instance": {
      "IamInstanceProfile": [
        "^(?!arn:aws:iam|ams|Ams|AMS|AWSManagedServices|Managed_Services|mc|Mc|MC|
sentinel|Sentinel).+",
        "arn:aws:iam::(\\$\\{AWS::AccountId\\}|[0-9]+):instance-profile/(?!ams|Ams|AMS|
AWSManagedServices|Managed_Services|mc|Mc|MC|sentinel|Sentinel).+"
      ]
    },
    "AWS::AutoScaling::LaunchConfiguration": {
      "IamInstanceProfile": [
```

```
        "^(?!arn:aws:iam|ams|Ams|AMS|AWSManagedServices|Managed_Services|mc|Mc|MC|
sentinel|Sentinel).+",
        "arn:aws:iam::(\\$\\{AWS::AccountId\\}|[0-9]+):instance-profile/(?!ams|Ams|AMS|
AWSManagedServices|Managed_Services|mc|Mc|MC|sentinel|Sentinel).+"
      ]
    },
    "AWS::EC2::LaunchTemplate": {
      "LaunchTemplateData.IamInstanceProfile.Name": [
        "^(?!ams|Ams|AMS|AWSManagedServices|Managed_Services|mc|Mc|MC|sentinel|
Sentinel).+"
      ],
      "LaunchTemplateData.IamInstanceProfile.Arn": [
        "arn:aws:iam::(\\$\\{AWS::AccountId\\}|[0-9]+):instance-profile\/(?!ams|Ams|
AMS|AWSManagedServices|Managed_Services|mc|Mc|MC|sentinel|Sentinel).+"
      ]
    }
}
```

리소스 검증

템플릿에는 허용 목록에 있는 리소스만 지정할 수 있습니다. 이러한 리소스는에 설명되어 있습니다지원되는 리소스.

패치 적용 제한으로 인해 동일한 스택에서 EC2 스택과 Auto Scaling 그룹(ASGs)이 허용되지 않습니다.

보안 그룹 수신 규칙 검증

- CFN Ingest Create 또는 Stack Update CT 변경 유형에서 오는 요청의 경우:
  - (`IpProtocol`는 tcp 또는 6) AND (포트는 80 또는 443) 인 경우 `CidrIP` 값에 대한 제한이 없습니다.
  - 그렇지 않으면는 0.0.0.0/0이 될 수 `CidrIP` 없습니다.
- Service Catalog(Service Catalog 제품)에서 오는 요청의 경우:
  - CFN Ingest Create 또는 Stack Update CT 변경 유형 검증 외에도의 프로토콜이 `management_ports` 있는의 포트는 `allowed_cidrs`를 통해서만 액세스할 `ip_protocols` 수 있습니다.

```
{
      "ip_protocols": ["tcp", "6", "udp", "17"],
      "management_ports": [22, 23, 389, 636, 1494, 1604, 2222, 3389, 5900, 5901,
 5985, 5986],
```

```
    "allowed_cidrs": ["10.0.0.0/8", "100.64.0.0/10", "172.16.0.0/12",
"192.168.0.0/16"]
 }
```

제한 사항

다음 기능 및 기능은 현재 AMS AWS CloudFormation 수집 프로세스에서 지원되지 않습니다.

- YAML - 지원되지 않습니다. JSON 기반 CloudFormation 템플릿만 지원됩니다.
- 중첩 스택 - 대신 단일 템플릿을 사용하도록 애플리케이션 인프라를 설계합니다. 또는 교차 스택 참조를 사용하여 한 리소스가 다른 리소스에 종속된 여러 스택에서 리소스를 분리할 수 있습니다. 자세한 내용은 연습: 다른 AWS CloudFormation 스택의 리소스 출력을 참조하세요.
- CloudFormation 스택 세트 - 보안 영향으로 인해 지원되지 않습니다.
- CloudFormation 템플릿을 사용한 IAM 리소스 생성 - 보안 영향으로 인해 IAM 역할만 지원됩니다.
- 민감한 데이터 - 지원되지 않습니다. 템플릿 또는 파라미터 값에 민감한 데이터를 포함하지 마십시오. 민감한 데이터를 참조해야 하는 경우 Secrets Manager를 사용하여 이러한 값을 저장하고 검색합니다. 리소스 속성에서 AWS Secrets Manager 보안 암호를 사용하는 방법에 대한 자세한 내용은 AWS AWS CloudFormation 템플릿을 사용하여 AWS Secrets Manager에서 관리되는 보안 암호를 생성하고 검색하는 방법 및 동적 참조를 사용하여 템플릿 값 지정을 참조하세요.

지원되는 리소스

다음 AWS 리소스는 AMS AWS CloudFormation 수집 프로세스에서 지원됩니다.

CloudFormation Ingest Stack: 지원되는 리소스

인스턴스 운영 체제는 AMS 워크로드 수집에서 지원되어야 합니다. 여기에 나열된 AWS 리소스만 지원됩니다.

- Amazon API Gateway
  - AWS::ApiGateway::Account
  - AWS::ApiGateway::ApiKey
  - AWS::ApiGateway::Authorizer
  - AWS::ApiGateway::BasePathMapping
  - AWS::ApiGateway::ClientCertificate

  - AWS::ApiGateway::Deployment
  - AWS::ApiGateway::DocumentationPart
  - AWS::ApiGateway::DocumentationVersion
  - AWS::ApiGateway::DomainName
  - AWS::ApiGateway::GatewayResponse
  - AWS::ApiGateway::Method
  - AWS::ApiGateway::Model
  - AWS::ApiGateway::RequestValidator
  - AWS::ApiGateway::Resource
  - AWS::ApiGateway::RestApi
  - AWS::ApiGateway::Stage
  - AWS::ApiGateway::UsagePlan
  - AWS::ApiGateway::UsagePlanKey
  - AWS::ApiGateway::VpcLink
- Amazon API Gateway V2
  - AWS::ApiGatewayV2::Api
  - AWS::ApiGatewayV2::ApiGatewayManagedOverrides
  - AWS::ApiGatewayV2::ApiMapping
  - AWS::ApiGatewayV2::Authorizer
  - AWS::ApiGatewayV2::Deployment
  - AWS::ApiGatewayV2::DomainName
  - AWS::ApiGatewayV2::Integration
  - AWS::ApiGatewayV2::IntegrationResponse
  - AWS::ApiGatewayV2::Model
  - AWS::ApiGatewayV2::Route
  - AWS::ApiGatewayV2::RouteResponse
  - AWS::ApiGatewayV2::Stage
  - AWS::ApiGatewayV2::VpcLink
- AppSync
  - AWS::AppSync::ApiCache

- AWS::AppSync::ApiKey
- AWS::AppSync::DataSource
- AWS::AppSync::FunctionConfiguration
- AWS::AppSync::GraphQLApi
- AWS::AppSync::GraphQLSchema
- AWS::AppSync::Resolver

- Amazon Athena
  - AWS::Athena::NamedQuery
  - AWS::Athena::WorkGroup
- AWS Backup
  - AWS::Backup::BackupVault
- Amazon CloudFront
  - AWS::CloudFront::Distribution
  - AWS::CloudFront::CloudFrontOriginAccessIdentity
  - AWS::CloudFront::StreamingDistribution
- Amazon CloudWatch
  - AWS::CloudWatch::Alarm
  - AWS::CloudWatch::AnomalyDetector
  - AWS::CloudWatch::CompositeAlarm
  - AWS::CloudWatch::Dashboard
  - AWS::CloudWatch::InsightRule
- Amazon CloudWatch Logs
  - AWS::Logs::LogGroup
  - AWS::Logs::LogStream
  - AWS::Logs::MetricFilter
  - AWS::Logs::SubscriptionFilter
- Amazon Cognito
  - AWS::Cognito::IdentityPool
  - AWS::Cognito::IdentityPoolRoleAttachment
  - AWS::Cognito::UserPool

- AWS::Cognito::UserPoolClient
- AWS::Cognito::UserPoolDomain
- AWS::Cognito::UserPoolGroup
- AWS::Cognito::UserPoolIdentityProvider
- AWS::Cognito::UserPoolResourceServer
- AWS::Cognito::UserPoolRiskConfigurationAttachment
- AWS::Cognito::UserPoolUICustomizationAttachment
- AWS::Cognito::UserPoolUser
- AWS::Cognito::UserPoolUserToGroupAttachment

- Amazon DocumentDB
  - AWS::DocDB::DBCluster
  - AWS::DocDB::DBClusterParameterGroup
  - AWS::DocDB::DBInstance
  - AWS::DocDB::DBSubnetGroup
- Amazon DynamoDB
  - AWS::DynamoDB::Table
- Amazon EC2
  - AWS::EC2::Volume
  - AWS::EC2::VolumeAttachment
  - AWS::EC2::Instance
  - AWS::EC2::EIP
  - AWS::EC2::EIPAssociation
  - AWS::EC2::NetworkInterface
  - AWS::EC2::NetworkInterfaceAttachment
  - AWS::EC2::SecurityGroup
  - AWS::EC2::SecurityGroupIngress
  - AWS::EC2::SecurityGroupEgress
  - AWS::EC2::LaunchTemplate
- AWS Batch
  - AWS::Batch::ComputeEnvironment

  - AWS::Batch::JobDefinition
  - AWS::Batch::JobQueue
- Amazon Elastic Container Registry(ECR)
  - AWS::ECR::Repository
- Amazon Elastic Container Service(ECS)(Fargate)
  - AWS::ECS::CapacityProvider
  - AWS::ECS::Cluster
  - AWS::ECS::PrimaryTaskSet
  - AWS::ECS::Service
  - AWS::ECS::TaskDefinition
  - AWS::ECS::TaskSet
- Amazon Elastic File System(EFS)
  - AWS::EFS::FileSystem
  - AWS::EFS::MountTarget
- Amazon ElastiCache
  - AWS::ElastiCache::CacheCluster
  - AWS::ElastiCache::ParameterGroup
  - AWS::ElastiCache::ReplicationGroup
  - AWS::ElastiCache::SecurityGroup
  - AWS::ElastiCache::SecurityGroupIngress
  - AWS::ElastiCache::SubnetGroup
- Amazon EventBridge
  - AWS::Events::EventBus
  - AWS::Events::EventBusPolicy
  - AWS::Events::Rule
- Amazon FSx
  - AWS::FSx::FileSystem
- Amazon Inspector
  - AWS::Inspector::AssessmentTarget
  - AWS::Inspector::AssessmentTemplate

  - AWS::Inspector::ResourceGroup
- Amazon Kinesis Data Analytics
  - AWS::KinesisAnalytics::Application
  - AWS::KinesisAnalytics::ApplicationOutput
  - AWS::KinesisAnalytics::ApplicationReferenceDataSource
- Amazon Kinesis Data Firehose
  - AWS::KinesisFirehose::DeliveryStream
- Amazon Kinesis Data Streams
  - AWS::Kinesis::Stream
  - AWS::Kinesis::StreamConsumer
- Amazon MQ
  - AWS::AmazonMQ::Broker
  - AWS::AmazonMQ::Configuration
  - AWS::AmazonMQ::ConfigurationAssociation
- Amazon OpenSearch
  - AWS::OpenSearchService::Domain
- Amazon Relational Database Service(RDS)
  - AWS::RDS::DBCluster
  - AWS::RDS::DBClusterParameterGroup
  - AWS::RDS::DBInstance
  - AWS::RDS::DBParameterGroup
  - AWS::RDS::DBSubnetGroup
  - AWS::RDS::EventSubscription
  - AWS::RDS::OptionGroup
- Amazon Route 53
  - AWS::Route53::HealthCheck
  - AWS::Route53::HostedZone
  - AWS::Route53::RecordSet
  - AWS::Route53::RecordSetGroup

  - AWS::Route53Resolver::ResolverRule

  - AWS::Route53Resolver::ResolverRuleAssociation
- Amazon S3
  - AWS::S3::Bucket
- Amazon Sagemaker
  - AWS::SageMaker::CodeRepository
  - AWS::SageMaker::Endpoint
  - AWS::SageMaker::EndpointConfig
  - AWS::SageMaker::Model
  - AWS::SageMaker::NotebookInstance
  - AWS::SageMaker::NotebookInstanceLifecycleConfig
  - AWS::SageMaker::Workteam
- Amazon Simple Email Service(SES)
  - AWS::SES::ConfigurationSet
  - AWS::SES::ConfigurationSetEventDestination
  - AWS::SES::ReceiptFilter
  - AWS::SES::ReceiptRule
  - AWS::SES::ReceiptRuleSet
  - AWS::SES::Template
- Amazon SimpleDB
  - AWS::SDB::Domain
- Amazon SNS
  - AWS::SNS::Subscription
  - AWS::SNS::Topic
- Amazon SQS
  - AWS::SQS::Queue
- Amazon WorkSpaces
  - AWS::WorkSpaces::Workspace
- 애플리케이션 AutoScaling
  - AWS::ApplicationAutoScaling::ScalableTarget
  - AWS::ApplicationAutoScaling::ScalingPolicy

- Amazon EC2 AutoScaling
  - AWS::AutoScaling::AutoScalingGroup
  - AWS::AutoScaling::LaunchConfiguration
  - AWS::AutoScaling::LifecycleHook
  - AWS::AutoScaling::ScalingPolicy
  - AWS::AutoScaling::ScheduledAction
- AWS Certificate Manager
  - AWS::CertificateManager::Certificate
- CloudFormation
  - AWS::CloudFormation::CustomResource
  - AWS::CloudFormation::Designer
  - AWS::CloudFormation::WaitCondition
  - AWS::CloudFormation::WaitConditionHandle
- AWS CodeBuild
  - AWS::CodeBuild::Project
  - AWS::CodeBuild::ReportGroup
  - AWS::CodeBuild::SourceCredential
- AWS CodeCommit
  - AWS::CodeCommit::Repository
- AWS CodeDeploy
  - AWS::CodeDeploy::Application
  - AWS::CodeDeploy::DeploymentConfig
  - AWS::CodeDeploy::DeploymentGroup
- AWS CodePipeline
  - AWS::CodePipeline::CustomActionType
  - AWS::CodePipeline::Pipeline
  - AWS::CodePipeline::Webhook
- AWS Database Migration Service(DMS)
  - AWS::DMS::Certificate
  - AWS::DMS::Endpoint

  - AWS::DMS::EventSubscription
  - AWS::DMS::ReplicationInstance
  - AWS::DMS::ReplicationSubnetGroup
  - AWS::DMS::ReplicationTask

  AWS::DMS::Endpoint 리소스의 MongoDbSettings 속성은 허용되지 않습니다.

  다음 속성은 AWS Secrets Manager에서 확인되는 경우에만 허용됩니다. AWS::DMS::Certificate 리소스의 CertificatePem 및 CertificateWallet 속성과 AWS::DMS::Endpoint 리소스의 Password 속성.
- AWS Elastic Load Balancing - Application Load Balancer/Network Load Balancer
  - AWS::ElasticLoadBalancingV2::Listener
  - AWS::ElasticLoadBalancingV2::ListenerCertificate
  - AWS::ElasticLoadBalancingV2::ListenerRule
  - AWS::ElasticLoadBalancingV2::LoadBalancer
  - AWS::ElasticLoadBalancingV2::TargetGroup
- AWS Elastic Load Balancing - Classic Load Balancer
  - AWS::ElasticLoadBalancing::LoadBalancer
- AWS Elemental MediaConvert
  - AWS::MediaConvert::JobTemplate
  - AWS::MediaConvert::Preset
  - AWS::MediaConvert::Queue
- AWS Elemental MediaStore
  - AWS::MediaStore::Container
- AWS Identity and Access Management (IAM)
  - AWS::IAM::Role
- AWS Managed Streaming for Apache Kafka(MSK)
  - AWS::MSK::Cluster
- AWS Glue
  - AWS::Glue::Classifier
  - AWS::Glue::Connection
  - AWS::Glue::Crawler
  - AWS::Glue::Database

  - AWS::Glue::DataCatalogEncryptionSettings
  - AWS::Glue::DevEndpoint
  - AWS::Glue::Job
  - AWS::Glue::MLTransform
  - AWS::Glue::Partition
  - AWS::Glue::SecurityConfiguration
  - AWS::Glue::Table
  - AWS::Glue::Trigger
  - AWS::Glue::Workflow
- AWS Key Management Service (KMS)
  - AWS::KMS::Key
  - AWS::KMS::Alias
- AWS Lake Formation
  - AWS::LakeFormation::DataLakeSettings
  - AWS::LakeFormation::Permissions
  - AWS::LakeFormation::Resource
- Lambda
  - AWS::Lambda::Alias
  - AWS::Lambda::EventInvokeConfig
  - AWS::Lambda::EventSourceMapping
  - AWS::Lambda::Function
  - AWS::Lambda::LayerVersion
  - AWS::Lambda::LayerVersionPermission
  - AWS::Lambda::Permission
  - AWS::Lambda::Version
- Amazon Redshift
  - AWS::Redshift::Cluster
  - AWS::Redshift::ClusterParameterGroup
  - AWS::Redshift::ClusterSubnetGroup

- AWS Secrets Manager

  - AWS::SecretsManager::ResourcePolicy
  - AWS::SecretsManager::RotationSchedule
  - AWS::SecretsManager::Secret
  - AWS::SecretsManager::SecretTargetAttachment
- AWS Security Hub
  - AWS::SecurityHub::Hub
- AWS Step Functions
  - AWS::StepFunctions::Activity
  - AWS::StepFunctions::StateMachine
- AWS Systems Manager(SSM)
  - AWS::SSM::Parameter
- Amazon CloudWatch Synthetics
  - AWS::Synthetics::Canary
- AWS Transfer Family
  - AWS::Transfer::Server
  - AWS::Transfer::User
- AWS WAF
  - AWS::WAF::ByteMatchSet
  - AWS::WAF::IPSet
  - AWS::WAF::Rule
  - AWS::WAF::SizeConstraintSet
  - AWS::WAF::SqlInjectionMatchSet
  - AWS::WAF::WebACL
  - AWS::WAF::XssMatchSet
- AWS WAF 리전
  - AWS::WAFRegional::ByteMatchSet
  - AWS::WAFRegional::GeoMatchSet
  - AWS::WAFRegional::IPSet
  - AWS::WAFRegional::RateBasedRule
  - AWS::WAFRegional::RegexPatternSet

- AWS::WAFRegional::Rule
- AWS::WAFRegional::SizeConstraintSet
- AWS::WAFRegional::SqlInjectionMatchSet
- AWS::WAFRegional::WebACL
- AWS::WAFRegional::WebACLAssociation
- AWS::WAFRegional::XssMatchSet

- AWS WAFv2
  - AWS::WAFv2::IPSet
  - AWS::WAFv2::RegexPatternSet
  - AWS::WAFv2::RuleGroup
  - AWS::WAFv2::WebACL
  - AWS::WAFv2::WebACLAssociation

AWS CloudFormation 수집: 예

여기에서 CloudFormation으로 스택 생성 템플릿 변경 유형을 사용하는 방법에 대한 몇 가지 세부 예제를 찾아보세요.

당 샘플 CloudFormation 템플릿 세트를 다운로드하려면 샘플 템플릿을 AWS 리전참조하세요. https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/cfn-sample-templates.html

AWS CloudFormation 리소스에 대한 참조 정보는 AWS 리소스 및 속성 유형 참조를 참조하세요. 그러나 AMS는에 설명된 더 작은 리소스 세트를 지원합니다AMS CloudFormation 수집.

> **Note**
>
> AMS는 모든 IAM 또는 기타 정책 관련 리소스를 수집하여 단일 관리 | 기타 | 기타 | 변경 유형 생성(ct-1e1xtak34nx76)에 제출하도록 권장합니다. 예를 들어 필요한 모든 IAM 역할, IAM 인스턴스 프로파일, 기존 IAM 역할에 대한 IAM 정책 업데이트, S3 버킷 정책, SNS/SQS 정책 등을 결합한 다음 ct-1e1xtak34nx76 RFC를 제출하여 이러한 기존 리소스를 향후 CFN Ingest 템플릿 내에서 참조할 수 있도록 합니다.

주제

- AWS CloudFormation 수집 예제: 리소스 정의

- CloudFormation Ingest 예제: 3티어 웹 애플리케이션

AWS CloudFormation 수집 예제: 리소스 정의

AMS AWS CloudFormation 수집을 사용하는 경우 CloudFormation 템플릿을 사용자 지정하고 CloudFormation 수집 변경 유형(ct-36cn2avfrrj9v)을 사용하여 RFC의 AMS에 제출합니다. 여러 번 재사용할 수 있는 CloudFormation 템플릿을 생성하려면 CloudFormation 템플릿에서 하드 코딩하는 대신 CloudFormation 수집 변경 유형 실행 입력에 스택 구성 파라미터를 추가합니다. 가장 큰 이점은 템플릿을 재사용할 수 있다는 것입니다.

AMS CloudFormation 수집 변경 유형 입력 스키마를 사용하면 CloudFormation 템플릿에서 최대 60개의 파라미터를 선택하고 사용자 지정 값을 제공할 수 있습니다.

이 예제에서는 다양한 CloudFormation 템플릿에서 AMS CloudFormation 수집 CT의 파라미터로 사용할 수 있는 리소스 속성을 정의하는 방법을 보여줍니다. 이 섹션의 예제는 특히 SNS 주제 사용을 보여줍니다.

주제

- 예제 1: AWS CloudFormation SNSTopic 리소스 TopicName 속성 하드 코드
- 예제 2: SNSTopic 리소스를 사용하여 AMS 변경 유형의 파라미터 참조
- 예제 3: AMS 수집 변경 유형이 있는 JSON 실행 파라미터 파일을 제출하여 SNS 주제 생성
- 예제 4: 동일한 CloudFormation 템플릿을 참조하는 새 변경 유형 제출
- 예제 5: CloudFormation 템플릿의 기본 파라미터 값 사용

예제 1: AWS CloudFormation SNSTopic 리소스 **`TopicName`** 속성 하드 코드

이 예제에서는 CloudFormation 템플릿에서 AWS CloudFormation SNSTopic 리소스 `TopicName` 속성을 하드 코딩합니다. `Parameters` 섹션은 비어 있습니다.

새 CloudFormation 템플릿을 생성할 필요 없이 새 스택의 SNSTopic 이름 값을 변경할 수 있는 CloudFormation 템플릿을 사용하려면 CloudFormation 수집 변경 유형의 AMS `Parameters` 섹션을 사용하여 해당 구성을 만들 수 있습니다. 이렇게 하면 나중에 동일한 CloudFormation 템플릿을 사용하여 다른 `SNSTopic` 이름으로 새 스택을 생성할 수 있습니다.

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Description" : "My SNS Topic",
```

```
  "Parameters" : {
  },
  "Resources" : {
    "SNSTopic" : {
      "Type" : "AWS::SNS::Topic",
      "Properties" : {
        "TopicName" : "MyTopicName"
      }
    }
  }
}
```

예제 2: SNSTopic 리소스를 사용하여 AMS 변경 유형의 파라미터 참조

이 예제에서는 CloudFormation 템플릿에 정의된 SNSTopic 리소스 TopicName 속성을 사용하여 AMS 변경 유형의를 참조Parameter합니다.

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Description" : "My SNS Topic",
  "Parameters" : {
    "TopicName" : {
      "Type" : "String",
      "Description" : "Topic ID",
      "Default" : "MyTopicName"
    }
  },
  "Resources" : {
    "SNSTopic" : {
      "Type" : "AWS::SNS::Topic",
      "Properties" : {
        "TopicName" : { "Ref" : "TopicName"}
      }
    }
  }
}
```

예제 3: AMS 수집 변경 유형이 있는 JSON 실행 파라미터 파일을 제출하여 SNS 주제 생성

이 예제에서는 SNS 주제를 생성하는 AMS 수집 CT와 함께 JSON 실행 파라미터 파일을 제출합니다TopicName. SNS 주제는이 예제에 표시된 수정 가능한 방식으로 CloudFormation 템플릿에 정의되어야 합니다.

```
{
  "Name": "cfn-ingest",
  "Description": "CFNIngest Web Application Stack",
  "CloudFormationTemplateS3Endpoint": "$S3_PRESIGNED_URL",
  "VpcId": "VPC_ID",
  "Tags": [
    {"Key": "Enviroment Type", "Value": "Dev"}
  ],
  "Parameters": [
    {"Name": "TopicName", "Value": "MyTopic1"}
  ],
  "TimeoutInMinutes": 60
}
```

예제 4: 동일한 CloudFormation 템플릿을 참조하는 새 변경 유형 제출

이 JSON 예제에서는 CloudFormation 템플릿을 변경하지 않고 SNS TopicName 값을 변경합니다. 대신 동일한 CFN 템플릿을 참조하는 새 배포 | 수집 | CloudFormation 템플릿의 스택 | 변경 유형 생성을 제출합니다.

```
{
  "Name": "cfn-ingest",
  "Description": "CFNIngest Web Application Stack",
  "CloudFormationTemplateS3Endpoint": "$S3_PRESIGNED_URL",
  "VpcId": "VPC_ID",
  "Tags": [
    {"Key": "Enviroment Type", "Value": "Dev"}
  ],
  "Parameters": [
    {"Name": "TopicName", "Value": "MyTopic2"}
  ],
  "TimeoutInMinutes": 60
}
```

예제 5: CloudFormation 템플릿의 기본 파라미터 값 사용

이 예제에서는 Parameters 실행 파라미터에 TopicName 값이 제공되지 않았기 때문에 SNS TopicName = 'MyTopicName'이 생성됩니다. Parameters 정의를 제공하지 않으면 CloudFormation 템플릿의 기본 파라미터 값이 사용됩니다.

```
{
  "Name": "cfn-ingest",
```

```
  "Description": "CFNIngest Web Application Stack",
  "CloudFormationTemplateS3Endpoint": "$S3_PRESIGNED_URL",
  "VpcId": "VPC_ID",
  "Tags": [
    {"Key": "Enviroment Type", "Value": "Dev"}
  ],
  "TimeoutInMinutes": 60
}
```

CloudFormation Ingest 예제: 3티어 웹 애플리케이션

표준 3-Tier 웹 애플리케이션에 대한 CloudFormation 템플릿을 수집합니다.

여기에는 Application Load Balancer, Application Load Balancer 대상 그룹, Auto Scaling 그룹, Auto Scaling 그룹 시작 템플릿, MySQL 데이터베이스가 있는 Amazon Relational Database Service(SQL Server용 RDS), AWS SSM 파라미터 스토어 및 AWS Secrets Manager가 포함됩니다. 이 예제를 살펴보려면 30~60분을 기다립니다.

사전 조건

- Secrets Manager를 사용하여 해당 값을 가진 사용자 이름과 암호가 포함된 AWS 암호를 생성합니다. 보안 암호 이름가 포함된이 샘플 JSON 템플릿(zip 파일)을 참조하여 보안 암호 이름으로 `ams-shared/myapp/dev/dbsecrets`바꿀 수 있습니다. AMS에서 AWS Secrets Manager를 사용하는 방법에 대한 자세한 내용은 섹션을 참조하세요AMS 리소스와 함께 AWS Secrets Manager 사용.
- AWS SSM Parameter Store(PS)에서 필요한 파라미터를 설정합니다. 이 예제에서는 프라이빗 및 퍼블릭 서브넷`Subnet-Id`의 `VPCId` 및가 , `/app/DemoApp/PublicSubnet1a`, 및 `PublicSubnet1cPrivateSubnet1aPrivateSubnet1c`와 같은 경로의 SSM PS에 저장됩니다`VPCCidr`. 필요에 따라 경로와 파라미터 이름 및 값을 업데이트합니다.
- AWS Secrets Manager 및 SSM Parameter Store 경로에 대한 읽기 권한이 있는 IAM Amazon EC2 인스턴스 역할을 생성합니다(이 예제에서 생성 및 사용되는 IAM 역할은 임`customer-ec2_secrets_manager_instance_profile`). 인스턴스 프로파일 역할과 같은 IAM 표준 정책을 생성하는 경우 역할 이름은 로 시작해야 합니다`customer-`. 새 IAM 역할을 생성하려면 (이름 또는 다른 `customer-ec2_secrets_manager_instance_profile`이름을 지정할 수 있음) AMS 변경 유형 관리 | 애플리케이션 | IAM 인스턴스 프로파일 | 생성(ct-0ixp4ch2tiu04) CT를 사용하고 필요한 정책을 연결합니다. AWS IAM 콘솔`customer_systemsmanager_parameterstore_policy`에서 AMS IAM 표준 정책 `customer_secrets_manager_policy` 및를 검토하여 있는 그대로 또는 참조로 사용할 수 있습니다.

표준 3-Tier 웹 애플리케이션을 위한 CloudFormation 템플릿 수집

1. 첨부된 샘플 CloudFormation JSON 템플릿을 zip 파일인 3-tier-cfn-ingest.zip으로 S3 버킷에 업로드하고 CFN Ingest RFC에 사용할 서명된 S3 URL을 생성합니다. 자세한 내용은 presign을 참조하세요. AMS 콘솔을 통해 RFC를 제출할 때 CFN 템플릿을 CFN Ingest RFC에 복사/붙여넣을 수도 있습니다.
2. AMS 콘솔 또는 AMS CLI를 통해 CloudFormation Ingest RFC(배포 | 수집 | CloudFormation 템플릿의 스택 | 생성(ct-36cn2avfrrj9v))를 생성합니다. CloudFormation 수집 자동화 프로세스는 CloudFormation 템플릿을 검증하여 템플릿에 유효한 AMS 지원 리소스가 있는지 확인하고 보안 표준을 준수합니다.

   - 콘솔 사용 - 변경 유형에서 CloudFormation 템플릿에서 배포 -> 수집 -> 스택 -> 생성을 선택한 다음, 다음 파라미터를 예로 추가합니다(MultiAZDatabase의 기본값은 false임). CloudFormation

     ```
     CloudFormationTemplateS3Endpoint: "https://s3-ap-
     southeast-2.amazonaws.com/amzn-s3-demo-bucket/3-tier-cfn-ingest.json?
     AWSAccessKeyId=#{S3_ACCESS_KEY_ID}&Expires=#{EXPIRE_DATE}&Signature=#{SIGNATURE}"
     VpcId: "VPC_ID"
     TimeoutInMinutes: 120
     IAMEC2InstanceProfile: "customer_ec2_secrets_manager_instance_profile"
     MultiAZDatabase: "true"
     WebServerCapacity: "2"
     ```

   - 사용 AWS CLI -를 사용하여 RFCs를 생성하는 방법에 대한 자세한 내용은 RFC 생성을 AWS CLI참조하세요. RFCs 예를 들어 다음 명령을 실행합니다.

     ```
     aws --profile=saml amscm create-rfc  --change-type-id ct-36cn2avfrrj9v
      --change-type-version "2.0" --title "TEST_CFN_INGEST" --execution-
     parameters "{\"CloudFormationTemplateS3Endpoint\":\"https://s3-
     ap-southeast-2.amazonaws.com/my-bucket/3-tier-cfn-ingest.json?
     AWSAccessKeyId=#{S3_ACCESS_KEY_ID}&Expires=#{EXPIRE_DATE}&Signature=#{SIGNATURE}\",
     \"TimeoutInMinutes\":120,\"Description\":\"TEST\",\"VpcId\":\"VPC_ID\",
     \"Name\":\"MY_TEST\",\"Tags\":[{\"Key\":\"env\",\"Value\":\"test\"}],
     \"Parameters\":[{\"Name\":\"IAMEC2InstanceProfile\",\"Value\":
     \"customer_ec2_secrets_manager_instance_profile\"},{\"Name\":\"MultiAZDatabase\",
     \"Value\":\"true\"},{\"Name\":\"VpcId\",\"Value\":\"VPC_ID\"},{\"Name\":
     \"WebServerCapacity\",\"Value\":\"2\"}]}" --endpoint-url https://amscm.us-
     east-1.amazonaws.com/operational/ --no-verify-ssl
     ```

AWS CloudFormation RFC 실행 출력에서 Application Load Balancer URL을 찾아 웹 사이트에 액세스합니다. 리소스 액세스에 대한 자세한 내용은 인스턴스 액세스를 참조하세요.

CloudFormation 수집 스택 생성

콘솔을 사용하여 CloudFormation 수집 스택 생성

콘솔을 사용하여 CloudFormation 수집 스택을 생성하려면

1. RFC 생성 페이지로 이동합니다. AMS 콘솔의 왼쪽 탐색 창에서 RFCs 클릭하여 RFCs 목록 페이지를 연 다음 RFC 생성을 클릭합니다.
2. 기본 변경 유형 찾아보기 보기에서 인기 있는 변경 유형(CT)을 선택하거나 범주별 선택 보기에서 CT를 선택합니다.
   - 변경 유형별 찾아보기: 빠른 생성 영역에서 인기 있는 CT를 클릭하여 RFC 실행 페이지를 즉시 열 수 있습니다. 빠른 생성으로 이전 CT 버전을 선택할 수 없습니다.

     CTs 정렬하려면 카드 또는 테이블 보기에서 모든 변경 유형 영역을 사용합니다. 어느 보기에서든 CT를 선택한 다음 RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다. 해당하는 경우 RFC 생성 버튼 옆에 이전 버전으로 생성 옵션이 나타납니다.
   - 범주별 선택: 범주, 하위 범주, 항목 및 작업을 선택하면 해당하는 경우 이전 버전으로 생성 옵션이 있는 CT 세부 정보 상자가 열립니다. RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다.
3. RFC 실행 페이지에서 CT 이름 영역을 열어 CT 세부 정보 상자를 확인합니다. 제목은 필수입니다(변경 유형 찾아보기 보기에서 CT를 선택하면 입력됨). 추가 구성 영역을 열어 RFC에 대한 정보를 추가합니다.

   실행 구성 영역에서 사용 가능한 드롭다운 목록을 사용하거나 필요한 파라미터의 값을 입력합니다. 선택적 실행 파라미터를 구성하려면 추가 구성 영역을 엽니다.
4. 완료되면 실행을 클릭합니다. 오류가 없는 경우 성공적으로 생성된 RFC 페이지에 제출된 RFC 세부 정보와 초기 실행 출력이 표시됩니다.
5. 실행 파라미터 영역을 열어 제출한 구성을 확인합니다. 페이지를 새로 고쳐 RFC 실행 상태를 업데이트합니다. 선택적으로 RFC를 취소하거나 페이지 상단의 옵션을 사용하여 RFC 사본을 생성합니다.

CLI를 사용하여 CloudFormation 수집 스택 생성

CLI를 사용하여 CloudFormation 수집 스택을 생성하려면

1. 인라인 생성(모든 RFC 및 실행 파라미터가 포함된 `create-rfc` 명령을 실행) 또는 템플릿 생성(2개의 JSON 파일을 생성, 하나는 RFC 파라미터용이고 다른 하나는 실행 파라미터용)을 사용하고 두 파일을 입력으로 사용하여 `create-rfc` 명령을 실행합니다. 두 방법 모두 여기에 설명되어 있습니다.
2. 반환된 RFC ID로 RFC: `aws amscm submit-rfc --rfc-id ID` 명령을 제출합니다.

   RFC: `aws amscm get-rfc --rfc-id ID` 명령을 모니터링합니다.

변경 유형 버전을 확인하려면 다음 명령을 사용합니다.

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 변경 유형에 대한 스키마의 일부인지 여부에 관계없이 모든 RFC에서 `CreateRfc` 파라미터를 사용할 수 있습니다. 예를 들어 RFC 상태가 변경될 때 알림을 받으려면 요청의 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` RFC 파라미터 부분(실행 파라미터 아님)에이 줄을 추가합니다. 모든 CreateRfc 파라미터 목록은 AMS Change Management API 참조를 참조하세요.

1. 스택을 생성하는 데 사용할 CloudFormation 템플릿을 준비하고 S3 버킷에 업로드합니다. 중요한 세부 정보는 AWS CloudFormation 수집 지침, 모범 사례 및 제한을 참조하세요.
2. RFC를 생성하여 AMS에 제출합니다.

   - 실행 파라미터 JSON 파일을 생성 및 저장하고 원하는 CloudFormation 템플릿 파라미터를 포함합니다. 다음 예제에서는 CreateCfnParams.json.

     웹 애플리케이션 스택 CreateCfnParams.json 파일 예제:

     ```
     {
       "Name": "cfn-ingest",
       "Description": "CFNIngest Web Application Stack",
       "VpcId": "VPC_ID",
     ```

```
  "CloudFormationTemplateS3Endpoint": "$S3_URL",
  "TimeoutInMinutes": 120,
  "Tags": [
   {
    "Key":   "Enviroment Type"
    "Value": "Dev",
   },
   {
    "Key":   "Application"
    "Value": "PCS",
   }
  ],
  "Parameters": [
   {
    "Name": "Parameter-for-S3Bucket-Name",
    "Value":  "BUCKET-NAME"
   },
   {
    "Name": "Parameter-for-Image-Id",
    "Value":  "AMI-ID"
   }
  ],
}
```

예제 SNS 주제 CreateCfnParams.json 파일:

```
{
  "Name": "cfn-ingest",
  "Description": "CFNIngest Web Application Stack",
  "CloudFormationTemplateS3Endpoint": "$S3_URL",
  "Tags": [
    {"Key": "Enviroment Type", "Value": "Dev"}
  ],
  "Parameters": [
    {"Name": "TopicName", "Value": "MyTopic1"}
  ]
}
```

3. 다음 콘텐츠와 함께 RFC 파라미터 JSON 파일을 생성하고 저장합니다. 다음 예제에서는 CreateCfnRfc.json 파일의 이름을 지정합니다.

```
{
   "ChangeTypeId": "ct-36cn2avfrrj9v",
```

```
   "ChangeTypeVersion": "2.0",
   "Title": "cfn-ingest"
}
```

4. CreateCfnRfc 파일과 CreateCfnParams 파일을 지정하여 RFC를 생성합니다.

```
aws amscm create-rfc --cli-input-json file://CreateCfnRfc.json  --execution-
parameters file://CreateCfnParams.json
```

응답에서 새 RFC의 ID를 수신하고 이를 사용하여 RFC를 제출하고 모니터링할 수 있습니다. 제출하기 전까지는 RFC가 편집 상태로 유지되고 시작되지 않습니다.

팁

> **Note**
>
> 이 변경 유형은 버전 2.0이며 자동화됩니다(수동 실행되지 않음). 이렇게 하면 CT 실행이 더 빠르게 진행되고, 새로운 파라미터인 CloudFormationTemplate을 사용하면 RFC에 사용자 지정 CloudFormation 템플릿을 붙여넣을 수 있습니다. 또한이 버전에서는 사용자가 자체 보안 그룹을 지정하는 경우 기본 AMS 보안 그룹을 연결하지 않습니다. 요청에 자체 보안 그룹을 지정하지 않으면 AMS가 AMS 기본 보안 그룹을 연결합니다. CFN Ingest v1.0에서는 사용자가 자체 보안 그룹을 제공했는지 여부에 관계없이 항상 AMS 기본 보안 그룹을 추가했습니다.
> AMS는이 변경 유형에 사용할 수 있도록 17개의 AMS 자체 프로비저닝 서비스를 활성화했습니다. 지원되는 리소스에 대한 자세한 내용은 CloudFormation Ingest Stack: 지원되는 리소스를 참조하세요.

> **Note**
>
> 버전 2.0은 미리 서명된 URL이 아닌 S3 엔드포인트를 허용합니다.
> 이 CT의 이전 버전을 사용하는 경우 CloudFormationTemplateS3Endpoint 파라미터 값은 미리 서명된 URL이어야 합니다.
> 미리 서명된 S3 버킷 URL(Mac/Linux)을 생성하기 위한 명령 예제:
>
> ```
> export S3_PRESIGNED_URL=$(aws s3 presign DASHDASHexpires-in 86400
>  s3://BUCKET_NAME/CFN_TEMPLATE.json)
> ```
>
> 미리 서명된 S3 버킷 URL을 생성하기 위한 명령 예제(Windows):

```
for /f %i in ('aws s3 presign DASHDASHexpires-in 86400
 s3://BUCKET_NAME/CFN_TEMPLATE.json') do set S3_PRESIGNED_URL=%i
```

Amazon S3 버킷에 대해 미리 서명된 URLs 생성을 참조하세요.

> **Note**
>
> S3 버킷이 AMS 계정에 있는 경우이 명령에 AMS 자격 증명을 사용해야 합니다. 예를 들어 AMS AWS Security Token Service (AWS STS) 자격 증명을 얻은 `--profile saml` 후를 추가해야 할 수 있습니다.

관련 변경 유형: CloudFormation 수집 스택 변경 세트 승인, AWS CloudFormation 수집 스택 업데이트

AWS CloudFormation에 대한 자세한 내용은 AWS Cloud Formation을 참조하세요. CloudFormation 템플릿을 보려면 AWS CloudFormation 템플릿 참조를 엽니다.

AWS CloudFormation 수집 검증

템플릿은 AMS 계정에서 생성할 수 있도록 검증되었습니다. 검증을 통과하면 AMS를 준수하는 데 필요한 리소스 또는 구성을 포함하도록 업데이트됩니다. 여기에는 AMS Operations가 스택을 모니터링할 수 있도록 Amazon CloudWatch 경보와 같은 리소스 추가가 포함됩니다.

다음 중 하나에 해당하는 경우 RFC가 거부됩니다.

- RFC JSON 구문이 잘못되었거나 지정된 형식을 따르지 않습니다.
- 제공된 S3 버킷 미리 서명된 URL이 유효하지 않습니다.
- 템플릿이 유효한 AWS CloudFormation 구문이 아닙니다.
- 템플릿에는 모든 파라미터 값에 대해 기본값이 설정되어 있지 않습니다.
- 템플릿이 AMS 검증에 실패합니다. AMS 검증 단계는이 주제 뒷부분의 정보를 참조하세요.

리소스 생성 문제로 인해 CloudFormation 스택이 생성되지 않으면 RFC가 실패합니다.

CFN 검증 및 검사기에 대한 자세한 내용은 템플릿 검증 및 CloudFormation 수집 스택: CFN 검사기 예제를 참조하세요.

AWS CloudFormation 수집 스택 업데이트

콘솔을 사용하여 CloudFormation 수집 스택 업데이트

콘솔을 사용하여 CloudFormation Ingest Stack을 업데이트하려면

1. RFC 생성 페이지로 이동합니다. AMS 콘솔의 왼쪽 탐색 창에서 RFCs 클릭하여 RFCs 목록 페이지를 연 다음 RFC 생성을 클릭합니다.
2. 기본 변경 유형 찾아보기 보기에서 인기 있는 변경 유형(CT)을 선택하거나 범주별 선택 보기에서 CT를 선택합니다.
   - 변경 유형별 찾아보기: 빠른 생성 영역에서 인기 있는 CT를 클릭하여 RFC 실행 페이지를 즉시 열 수 있습니다. 빠른 생성으로 이전 CT 버전을 선택할 수 없습니다.

     CTs 정렬하려면 카드 또는 테이블 보기에서 모든 변경 유형 영역을 사용합니다. 어느 보기에서든 CT를 선택한 다음 RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다. 해당하는 경우 RFC 생성 버튼 옆에 이전 버전으로 생성 옵션이 나타납니다.
   - 범주별 선택: 범주, 하위 범주, 항목 및 작업을 선택하면 해당하는 경우 이전 버전으로 생성 옵션이 있는 CT 세부 정보 상자가 열립니다. RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다.
3. RFC 실행 페이지에서 CT 이름 영역을 열어 CT 세부 정보 상자를 확인합니다. 제목은 필수입니다(변경 유형 찾아보기 보기에서 CT를 선택하면 입력됨). 추가 구성 영역을 열어 RFC에 대한 정보를 추가합니다.

   실행 구성 영역에서 사용 가능한 드롭다운 목록을 사용하거나 필요한 파라미터의 값을 입력합니다. 선택적 실행 파라미터를 구성하려면 추가 구성 영역을 엽니다.
4. 완료되면 실행을 클릭합니다. 오류가 없는 경우 RFC가 성공적으로 생성된 페이지에 제출된 RFC 세부 정보와 초기 실행 출력이 표시됩니다.
5. 실행 파라미터 영역을 열어 제출한 구성을 확인합니다. 페이지를 새로 고쳐 RFC 실행 상태를 업데이트합니다. 선택적으로 RFC를 취소하거나 페이지 상단의 옵션을 사용하여 RFC 사본을 생성합니다.

CLI를 사용하여 CloudFormation 수집 스택 업데이트

CLI를 사용하여 CloudFormation 수집 스택을 업데이트하려면

1. 인라인 생성(모든 RFC 및 실행 파라미터가 포함된 `create-rfc` 명령을 실행) 또는 템플릿 생성(2개의 JSON 파일을 생성, 하나는 RFC 파라미터용이고 다른 하나는 실행 파라미터용)을 사용하고 두

파일을 입력으로 사용하여 `create-rfc` 명령을 실행합니다. 두 방법 모두 여기에 설명되어 있습니다.

2. 반환된 RFC ID로 RFC: `aws amscm submit-rfc --rfc-id ID` 명령을 제출합니다.

   RFC: `aws amscm get-rfc --rfc-id ID` 명령을 모니터링합니다.

변경 유형 버전을 확인하려면 다음 명령을 사용합니다.

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 변경 유형에 대한 스키마의 일부인지 여부에 관계없이 모든 RFC에서 `CreateRfc` 파라미터를 사용할 수 있습니다. 예를 들어 RFC 상태가 변경될 때 알림을 받으려면 요청의 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` RFC 파라미터 부분(실행 파라미터 아님)에이 줄을 추가합니다. 모든 CreateRfc 파라미터 목록은 AMS Change Management API 참조를 참조하세요.

1. 스택을 업데이트하는 데 사용할 AWS CloudFormation 템플릿을 준비하고 S3 버킷에 업로드합니다. 중요한 세부 정보는 AWS CloudFormation 수집 지침, 모범 사례 및 제한을 참조하세요.
2. RFC를 생성하여 AMS에 제출합니다.

   - 실행 파라미터 JSON 파일을 생성 및 저장하고 원하는 CloudFormation 템플릿 파라미터를 포함합니다. 이 예제에서는 UpdateCfnParams.json.

     인라인 파라미터 업데이트가 포함된 UpdateCfnParams.json 파일의 예:

```
{
  "StackId": "stack-yjjoo9aicjyqw4ro2",
  "VpcId": "VPC_ID",
  "CloudFormationTemplate": "{\"AWSTemplateFormatVersion\":\"2010-09-09\",
\"Description\":\"Create a SNS topic\",\"Parameters\":{\"TopicName\":{\"Type
\":\"String\"},\"DisplayName\":{\"Type\":\"String\"}},\"Resources\":{\"SnsTopic
\":{\"Type\":\"AWS::SNS::Topic\",\"Properties\":{\"TopicName\":{\"Ref\":
\"TopicName\"},\"DisplayName\":{\"Ref\":\"DisplayName\"}}}}}",
  "TemplateParameters": [
    {
```

```
      "Key": "TopicName",
      "Value": "TopicNameCLI"
    },
    {
      "Key": "DisplayName",
      "Value": "DisplayNameCLI"
    }
  ],
  "TimeoutInMinutes": 1440
}
```

업데이트된 CloudFormation 템플릿이 포함된 S3 버킷 엔드포인트가 있는 UpdateCfnParams.json 파일의 예:

```
{
  "StackId": "stack-yjjoo9aicjyqw4ro2",
  "VpcId": "VPC_ID",
  "CloudFormationTemplateS3Endpoint": "s3_url",
  "TemplateParameters": [
    {
      "Key": "TopicName",
      "Value": "TopicNameCLI"
    },
    {
      "Key": "DisplayName",
      "Value": "DisplayNameCLI"
    }
  ],
  "TimeoutInMinutes": 1080
}
```

3. 다음 콘텐츠와 함께 RFC 파라미터 JSON 파일을 생성하고 저장합니다. 이 예제에서는 UpdateCfnRfc.json 파일의 이름을 지정합니다.

```
{
   "ChangeTypeId": "ct-361tlo1k7339x",
   "ChangeTypeVersion": "1.0",
   "Title": "cfn-ingest-template-update"
}
```

4. UpdateCfnRfc 파일과 UpdateCfnParams 파일을 지정하여 RFC를 생성합니다.

```
aws amscm create-rfc --cli-input-json file://UpdateCfnRfc.json  --execution-
parameters file://UpdateCfnParams.json
```

응답에서 새 RFC의 ID를 수신하고 이를 사용하여 RFC를 제출하고 모니터링할 수 있습니다. 제출하기 전까지는 RFC가 편집 상태로 유지되고 시작되지 않습니다.

팁

- 이 변경 유형은 이제 버전 2.0입니다. 변경 사항에는이 CT 버전 1.0에서 사용된 AutoApproveUpdateForResources 파라미터 제거와 AutoApproveRiskyUpdates 및 BypassDriftCheck라는 두 가지 새 파라미터 추가가 포함됩니다.
- S3 버킷이 AMS 계정에 있는 경우이 명령에 AMS 자격 증명을 사용해야 합니다. 예를 들어 AMS AWS Security Token Service (AWS STS) 자격 증명을 얻은 `--profile saml` 후를 추가해야 할 수 있습니다.
- CloudFormation 템플릿의 리소스에 대한 모든 `Parameter` 값은 기본값을 통해 또는 CT의 파라미터 섹션을 통해 사용자 지정 값을 가져야 합니다. 파라미터 키를 참조하도록 CloudFormation 템플릿 리소스를 구성하여 파라미터 값을 재정의할 수 있습니다. 방법을 보여주는 예제는 CloudFormation 수집 스택: CFN 검사기 예제를 참조하세요.

  중요: 양식에 명시적으로 제공되지 않은 파라미터가 누락되었습니다. 기본값은 기존 스택 또는 템플릿에서 현재 설정된 값입니다.
- AWS CloudFormation Ingest를 사용하여 추가할 수 있는 자체 프로비저닝된 서비스의 목록은 CloudFormation Ingest Stack: Supported Resources를 참조하세요.

  에 대한 자세한 내용은 AWS Cloud Formation을 AWS CloudFormation참조하세요.

AWS CloudFormation 수집 검증

템플릿은 AMS 계정에서 생성할 수 있도록 검증되었습니다. 검증을 통과하면 AMS를 준수하는 데 필요한 리소스 또는 구성을 포함하도록 업데이트됩니다. 여기에는 AMS Operations가 스택을 모니터링할 수 있도록 Amazon CloudWatch 경보와 같은 리소스 추가가 포함됩니다.

다음 중 하나에 해당하는 경우 RFC가 거부됩니다.

- RFC JSON 구문이 잘못되었거나 지정된 형식을 따르지 않습니다.
- 제공된 S3 버킷 미리 서명된 URL이 유효하지 않습니다.

- 템플릿이 유효한 AWS CloudFormation 구문이 아닙니다.
- 템플릿에는 모든 파라미터 값에 대해 기본값이 설정되어 있지 않습니다.
- 템플릿이 AMS 검증에 실패합니다. AMS 검증 단계는이 주제 뒷부분의 정보를 참조하세요.

리소스 생성 문제로 인해 CloudFormation 스택이 생성되지 않으면 RFC가 실패합니다.

CFN 검증 및 검사기에 대한 자세한 내용은 템플릿 검증 및 CloudFormation 수집 스택: CFN 검사기 예제를 참조하세요.

CloudFormation 수집 스택 변경 세트 승인

콘솔을 사용하여 CloudFormation 수집 스택 승인 및 업데이트

콘솔을 사용하여 CloudFormation 수집 스택을 승인하고 업데이트하려면

1. RFC 생성 페이지로 이동합니다. AMS 콘솔의 왼쪽 탐색 창에서 RFCs 클릭하여 RFCs 목록 페이지를 연 다음 RFC 생성을 클릭합니다.
2. 기본 변경 유형 찾아보기 보기에서 널리 사용되는 변경 유형(CT)을 선택하거나 범주별 선택 보기에서 CT를 선택합니다.
   - 변경 유형별 찾아보기: 빠른 생성 영역에서 인기 있는 CT를 클릭하여 RFC 실행 페이지를 즉시 열 수 있습니다. 빠른 생성으로 이전 CT 버전을 선택할 수 없습니다.

     CTs 정렬하려면 카드 또는 테이블 보기에서 모든 변경 유형 영역을 사용합니다. 어느 보기에서든 CT를 선택한 다음 RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다. 해당하는 경우 RFC 생성 버튼 옆에 이전 버전으로 생성 옵션이 나타납니다.
   - 범주별 선택: 범주, 하위 범주, 항목 및 작업을 선택하면 해당하는 경우 이전 버전으로 생성 옵션이 있는 CT 세부 정보 상자가 열립니다. RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다.
3. RFC 실행 페이지에서 CT 이름 영역을 열어 CT 세부 정보 상자를 확인합니다. 제목은 필수입니다 (변경 유형 찾아보기 보기에서 CT를 선택하면 입력됨). 추가 구성 영역을 열어 RFC에 대한 정보를 추가합니다.

   실행 구성 영역에서 사용 가능한 드롭다운 목록을 사용하거나 필요한 파라미터의 값을 입력합니다. 선택적 실행 파라미터를 구성하려면 추가 구성 영역을 엽니다.
4. 완료되면 실행을 클릭합니다. 오류가 없는 경우 성공적으로 생성된 RFC 페이지에 제출된 RFC 세부 정보와 초기 실행 출력이 표시됩니다.

5. 실행 파라미터 영역을 열어 제출한 구성을 확인합니다. 페이지를 새로 고쳐 RFC 실행 상태를 업데이트합니다. 선택적으로 RFC를 취소하거나 페이지 상단의 옵션을 사용하여 RFC 사본을 생성합니다.

CLI를 사용하여 CloudFormation 수집 스택 승인 및 업데이트

CLI를 사용하여 CloudFormation 수집 스택을 승인하고 업데이트하려면

1. 인라인 생성(모든 RFC 및 실행 파라미터가 포함된 `create-rfc` 명령을 실행) 또는 템플릿 생성(2개의 JSON 파일을 생성, 하나는 RFC 파라미터용이고 다른 하나는 실행 파라미터용)을 사용하고 두 파일을 입력으로 사용하여 `create-rfc` 명령을 실행합니다. 두 방법 모두 여기에 설명되어 있습니다.
2. 반환된 RFC ID로 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 명령을 제출합니다.

   RFC: `aws amscm get-rfc --rfc-id` *`ID`* 명령을 모니터링합니다.

변경 유형 버전을 확인하려면 다음 명령을 사용합니다.

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 변경 유형에 대한 스키마의 일부인지 여부에 관계없이 모든 `CreateRfc` 파라미터를 RFC와 함께 사용할 수 있습니다. 예를 들어 RFC 상태가 변경될 때 알림을 받으려면 요청의 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` RFC 파라미터 부분(실행 파라미터 아님)에이 줄을 추가합니다. 모든 CreateRfc 파라미터 목록은 AMS Change Management API 참조를 참조하세요.

1. 이 변경 유형에 대한 실행 파라미터 JSON 스키마를 현재 폴더의 파일로 출력합니다. 이 예제에서는 CreateAsgParams.json:의 이름을 지정합니다.

   ```
   aws amscm create-rfc --change-type-id "ct-1404e21baa2ox" --change-
   type-version "1.0" --title "Approve Update" --execution-parameters
    file://PATH_TO_EXECUTION_PARAMETERS --profile saml
   ```

2. 다음과 같이 스키마를 수정하고 저장합니다.

```
{
  "StackId": "STACK_ID",
  "VpcId": "VPC_ID",
  "ChangeSetName": "UPDATE-ef81e2bc-03f6-4b17-a3c7-feb700e78faa",
  "TimeoutInMinutes": 1080
}
```

팁

> **Note**
>
> 스택에 여러 리소스가 있고 스택 리소스의 하위 집합만 삭제하려는 경우 CloudFormation Update CT를 사용합니다. CloudFormation Ingest Stack: Updating을 참조하세요. 또한 서비스 요청 사례를 제출하면 AMS 엔지니어가 필요한 경우 변경 세트를 만드는 데 도움을 줄 수 있습니다.

에 대한 자세한 내용은 단원을 AWS CloudFormation참조하십시오AWS CloudFormation.

AWS CloudFormation 스택 종료 방지 업데이트

콘솔을 사용하여 AWS CloudFormation 종료 방지 스택 업데이트

다음은 AMS 콘솔에서이 변경 유형을 보여줍니다.

작동 방식:

1. RFC 생성 페이지로 이동합니다. AMS 콘솔의 왼쪽 탐색 창에서 RFCs 클릭하여 RFCs 목록 페이지를 연 다음 RFC 생성을 클릭합니다.
2. 기본 변경 유형 찾아보기 보기에서 인기 있는 변경 유형(CT)을 선택하거나 범주별 선택 보기에서 CT를 선택합니다.
   - 변경 유형별 찾아보기: 빠른 생성 영역에서 인기 있는 CT를 클릭하여 RFC 실행 페이지를 즉시 열 수 있습니다. 빠른 생성으로 이전 CT 버전을 선택할 수 없습니다.

     CTs 정렬하려면 카드 또는 테이블 보기에서 모든 변경 유형 영역을 사용합니다. 어느 보기에서든 CT를 선택한 다음 RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다. 해당하는 경우 RFC 생성 버튼 옆에 이전 버전으로 생성 옵션이 나타납니다.

- 범주별 선택: 범주, 하위 범주, 항목 및 작업을 선택하면 해당하는 경우 이전 버전으로 생성 옵션이 있는 CT 세부 정보 상자가 열립니다. RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다.

3. RFC 실행 페이지에서 CT 이름 영역을 열어 CT 세부 정보 상자를 확인합니다. 제목은 필수입니다(변경 유형 찾아보기 보기에서 CT를 선택하면 입력됨). 추가 구성 영역을 열어 RFC에 대한 정보를 추가합니다.

   실행 구성 영역에서 사용 가능한 드롭다운 목록을 사용하거나 필요한 파라미터의 값을 입력합니다. 선택적 실행 파라미터를 구성하려면 추가 구성 영역을 엽니다.

4. 완료되면 실행을 클릭합니다. 오류가 없는 경우 성공적으로 생성된 RFC 페이지에 제출된 RFC 세부 정보와 초기 실행 출력이 표시됩니다.

5. 실행 파라미터 영역을 열어 제출한 구성을 확인합니다. 페이지를 새로 고쳐 RFC 실행 상태를 업데이트합니다. 선택적으로 RFC를 취소하거나 페이지 상단의 옵션을 사용하여 RFC 사본을 생성합니다.

CLI를 사용하여 AWS CloudFormation 스택 종료 보호 업데이트

작동 방식:

1. 인라인 생성(모든 RFC 및 실행 파라미터가 포함된 `create-rfc` 명령을 실행) 또는 템플릿 생성(2개의 JSON 파일을 생성, 하나는 RFC 파라미터용이고 다른 하나는 실행 파라미터용)을 사용하고 두 파일을 입력으로 사용하여 `create-rfc` 명령을 실행합니다. 두 방법 모두 여기에 설명되어 있습니다.
2. 반환된 RFC ID로 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 명령을 제출합니다.

   RFC: `aws amscm get-rfc --rfc-id` *`ID`* 명령을 모니터링합니다.

변경 유형 버전을 확인하려면 다음 명령을 사용합니다.

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 변경 유형에 대한 스키마의 일부인지 여부에 관계없이 모든 `CreateRfc` 파라미터를 RFC와 함께 사용할 수 있습니다. 예를 들어 RFC 상태가 변경될 때 알림을 받으려면 요청의 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com`

\"]}}" RFC 파라미터 부분(실행 파라미터 아님)에이 줄을 추가합니다. 모든 CreateRfc 파라미터 목록은 AMS Change Management API 참조를 참조하세요.

변경하려는 파라미터만 지정합니다. 없는 파라미터는 기존 값을 유지합니다.

인라인 생성:

인라인으로 제공된 실행 파라미터(실행 파라미터를 인라인으로 제공할 때 이스케이프 따옴표)로 RFC 생성 명령을 실행한 다음 반환된 RFC ID를 제출합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
aws amscm create-rfc \
--change-type-id "ct-2uzbqr7x7mekd" \
--change-type-version "1.0" \
--title "Enable termination protection on CFN stack" \
--execution-parameters "{\"DocumentName\":\"AWSManagedServices-
ManageResourceTerminationProtection\",\"Region\":\"us-east-1\",\"Parameters\":
{\"ResourceId\":[\"stack-psvnq6cupymio3enl\"],\"TerminationProtectionDesiredState\":
[\"enabled\"]}}"
```

템플릿 생성:

1. 이 변경 유형의 실행 파라미터를 JSON 파일로 출력합니다.이 예제에서는 EnableTermProCFNParams.json:이라는 이름을 지정합니다.

   ```
   aws amscm get-change-type-version --change-type-id "ct-2uzbqr7x7mekd"
    --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
    EnableTermProCFNParams.json
   ```

2. 변경하려는 파라미터만 유지하면서 EnableTermProCFNParams 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

   ```
   {
     "DocumentName": "AWSManagedServices-ManageResourceTerminationProtection",
     "Region": "us-east-1",
     "Parameters": {
       "ResourceId": ["stack-psvnq6cupymio3enl"],
       "TerminationProtectionDesiredState": ["enabled"]
     }
   }
   ```

3. RFC 템플릿을 현재 폴더의 파일로 출력합니다.이 예제에서는 이름을 EnableTermProCFNRfc.json:으로 지정합니다.

```
aws amscm create-rfc --generate-cli-skeleton > EnableTermProCFNRfc.json
```

4. EnableTermProCFNRfc.json 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
{
    "ChangeTypeId": "ct-2uzbqr7x7mekd",
    "ChangeTypeVersion": "1.0",
    "Title": "Enable termination protection on CFN instance"
}
```

5. EnableTermProCFNRfc 파일과 EnableTermProCFNParams 파일을 지정하여 RFC를 생성합니다.

```
aws amscm create-rfc --cli-input-json file://EnableTermProCFNRfc.json  --execution-
parameters file://EnableTermProCFNParams.json
```

응답에서 새 RFC의 ID를 수신하고 이를 사용하여 RFC를 제출하고 모니터링할 수 있습니다. 제출하기 전까지는 RFC가 편집 상태로 유지되고 시작되지 않습니다.

팁

> **Note**
>
> Amazon EC2, EC2 스택: 종료 방지 업데이트와 관련된 CT가 있습니다.

종료 방지에 대한 자세한 내용은 스택이 삭제되지 않도록 보호를 참조하세요.

AMS에서 CFN 수집 또는 스택 업데이트 CTs 사용한 자동화된 IAM 배포

이러한 AMS 변경 유형을 사용하여 다중 계정 랜딩 존(MALZ)과 단일 계정 랜딩 존(SALZ) 모두에서 IAM 역할(`AWS::IAM::Role`리소스)을 배포할 수 있습니다.

- 배포 | 수집 | CloudFormation 템플릿의 스택 | 생성(ct-36cn2avfrrj9v)
- 관리 | 사용자 지정 스택 | CloudFormation 템플릿의 스택 | 업데이트(ct-361tlo1k7339x)

- 관리 | 사용자 지정 스택 | CloudFormation 템플릿의 스택 | 승인 및 업데이트(ct-1404e21baa2ox)

CFN 템플릿의 IAM 역할에 대해 수행된 검증:

- ManagedPolicyArns: 속성 ManagedPolicyArns가에 없어야 합니다`AWS::IAM::Role`. 검증을 통해 프로비저닝되는 역할에 관리형 정책을 연결할 수 없습니다. 대신 속성 정책을 통해 인라인 정책을 사용하여 역할에 대한 권한을 관리할 수 있습니다.
- PermissionsBoundary: 역할에 대한 권한 경계를 설정하는 데 사용되는 정책은 AMS 판매 관리형 정책인 만 가능합니다`AWSManagedServices_IAM_PermissionsBoundary`. 이 정책은 프로비저닝되는 역할을 사용하여 AMS 인프라 리소스가 수정되지 않도록 보호하는 가드레일 역할을 합니다. 이 기본 권한 경계를 사용하면 AMS가 제공하는 보안 이점이 유지됩니다.

  `AWSManagedServices_IAM_PermissionsBoundary` (기본값)이 필요하지 않으면 요청이 거부됩니다.
- MaxSessionDuration: IAM 역할에 대해 설정할 수 있는 최대 세션 기간은 1~4시간입니다. AMS 기술 표준에서는 4시간을 초과하는 세션 기간 동안 고객 위험 승인을 요구합니다.
- RoleName: 다음 네임스페이스는 AMS에서 보존되며 IAM 역할 이름 접두사로 사용할 수 없습니다.

```
AmazonSSMRole,
AMS,
Ams,
ams,
AWSManagedServices,
customer_developer_role,
customer-mc-,
Managed_Services,
MC,
Mc,
mc,
SENTINEL,
Sentinel,
sentinel,
StackSet-AMS,
StackSet-Ams,
StackSet-ams,
StackSet-AWS,
StackSet-MC,
StackSet-Mc,
StackSet-mc
```

- 정책: IAM 역할에 포함된 인라인 정책에는 AMS에서 사전 승인한 IAM 작업 세트만 포함될 수 있습니다. (제어 정책)을 사용하여 IAM 역할을 생성할 수 있는 모든 IAM 작업의 상한입니다. 제어 정책은 다음으로 구성됩니다.
  - 모든 AWS 서비스 및 리소스에 대한 읽기 전용 액세스를 제공하는 AWS 관리형 정책 ReadOnlyAccess의 모든 작업
  - 계정 간 S3 작업, 즉 허용된 S3 작업에 대한 제한이 있는 다음 작업은 생성 중인 역할과 동일한 계정에 있는 리소스에서만 수행할 수 있습니다.

```
amscm:*,
amsskms:*,
lambda:InvokeFunction,
logs:CreateLogStream,
logs:PutLogEvents,
s3:AbortMultipartUpload,
s3:DeleteObject,
s3:DeleteObjectVersion,
s3:ObjectOwnerOverrideToBucketOwner,
s3:PutObject,
s3:ReplicateTags,
secretsmanager:GetRandomPassword,
sns:Publish
```

  CFN 수집을 통해 생성되거나 업데이트된 모든 IAM 역할은이 제어 정책에 나열된 작업 또는 제어 정책에 나열된 작업에서 범위가 축소된 작업(미만 허용)을 허용할 수 있습니다. 현재 읽기 전용 작업으로 분류할 수 있는 이러한 안전한 IAM 작업과 CTs를 통해 수행할 수 없고 AMS 기술 표준에 따라 사전 승인된 위에서 언급한 비읽기 전용 작업을 허용합니다.

- AssumeRolePolicyDocument: 다음 엔터티는 사전 승인되었으며 생성 중인 역할을 수임하기 위해 신뢰 정책에 포함될 수 있습니다.
  - 동일한 계정의 모든 IAM 엔터티(역할, 사용자, 루트 사용자, STS 수임된 역할 세션)가 역할을 수임할 수 있습니다.
  - 다음은 역할을 수임할 AWS 서비스 수 있습니다.

```
apigateway.amazonaws.com,
autoscaling.amazonaws.com,
cloudformation.amazonaws.com,
codebuild.amazonaws.com,
codedeploy.amazonaws.com,
codepipeline.amazonaws.com,
datapipeline.amazonaws.com,
```

```
datasync.amazonaws.com,
dax.amazonaws.com,
dms.amazonaws.com,
ec2.amazonaws.com,
ecs-tasks.amazonaws.com,
ecs.application-autoscaling.amazonaws.com,
elasticmapreduce.amazonaws.com,
es.amazonaws.com,
events.amazonaws.com,
firehose.amazonaws.com,
glue.amazonaws.com,
lambda.amazonaws.com,
monitoring.rds.amazonaws.com,
pinpoint.amazonaws.com,
rds.amazonaws.com,
redshift.amazonaws.com,
s3.amazonaws.com,
sagemaker.amazonaws.com,
servicecatalog.amazonaws.com,
sns.amazonaws.com,
ssm.amazonaws.com,
states.amazonaws.com,
storagegateway.amazonaws.com,
transfer.amazonaws.com,
vmie.amazonaws.com
```

- 동일한 계정의 SAML 공급자가 역할을 수임할 수 있습니다. 현재 지원되는 유일한 SAML 공급자 이름은 입니다`customer-saml`.

하나 이상의 검증이 실패하면 RFC가 거부됩니다. 샘플 RFC 거부 이유는 다음과 같습니다.

```
{"errorMessage":"[ 'LambdaRole: The maximum session duration (in seconds) should be a
 numeric value in the range 3600 to 14400 (i.e. 1 to 4 hours).', 'lambda-policy: Policy
 document is too permissive.']","errorType":"ClientError"}
```

실패한 RFC 검증 또는 실행에 도움이 필요한 경우 RFC 서신을 사용하여 AMS에 문의하세요. 지침은 RFC 서신 및 첨부 파일(콘솔)을 참조하세요. 다른 질문이 있는 경우 서비스 요청을 제출하세요. 사용 방법은 서비스 요청 생성을 참조하세요.

> **Note**
>
> 현재 IAM 검증의 일환으로 IAM 모범 사례를 적용하지 않습니다. IAM 모범 사례는 IAM의 보안 모범 사례를 참조하세요.

더 허용적인 작업을 사용하여 IAM 역할 생성 또는 IAM 모범 사례 적용

다음과 같은 수동 변경 유형을 사용하여 IAM 엔터티를 생성합니다.

- 배포 | 고급 스택 구성 요소 | Identity and Access Management(IAM) | 개체 또는 정책 생성 (ct-3dpd8mdd9jn1r)
- 관리 | 고급 스택 구성 요소 | Identity and Access Management(IAM) | 엔터티 또는 정책 업데이트 (ct-27tuth19k52b4)

이러한 수동 RFCs를 제출하기 전에 기술 표준을 읽고 이해하는 것이 좋습니다. 액세스는 기술 표준에 액세스하는 방법을 참조하세요.

> **Note**
>
> 이러한 수동 변경 유형으로 직접 생성된 각 IAM 역할은 자체 개별 스택에 속하며 다른 인프라 리소스가 CFN Ingest CT를 통해 생성되는 동일한 스택에 상주하지 않습니다.

자동 변경 유형을 통해 업데이트를 수행할 수 없는 경우 수동 변경 유형을 통해 CFN 수집으로 생성된 IAM 역할 업데이트

관리 | 고급 스택 구성 요소 | Identity and Access Management(IAM) | 엔터티 또는 정책 업데이트 (ct-27tuth19k52b4) 변경 유형을 사용합니다.

> **Important**
>
> 수동 CT를 통한 IAM 역할 업데이트는 CFN 스택 템플릿에 반영되지 않으므로 스택 드리프트가 발생합니다. 검증을 통과하지 못한 상태로 수동 요청을 통해 역할이 업데이트되면 검증을 계속 준수하지 않는 한 스택 업데이트 CT(ct-361tlo1k7339x)를 다시 사용하여 역할을 더 이상 업데이트할 수 없습니다. 업데이트 CT는 CFN 스택 템플릿이 검증을 준수하는 경우에만 사용할 수 있습니다. 그러나 검증을 준수하지 않는 IAM 리소스가 업데이트되지 않고 CFN 템플릿이 검증을 통과하면 스택 업데이트 CT(ct-361tlo1k7339x)를 통해 스택을 업데이트할 수 있습니다.

AWS CloudFormation 수집을 통해 생성된 IAM 역할 삭제

전체 스택을 삭제하려면 다음과 같은 자동 스택 삭제 변경 유형을 사용합니다. 지침은 스택 삭제:

- 유형 ID 변경: ct-0q0bic0ywqk6c
- 분류: 관리 | 표준 스택 | 스택 | 삭제 및 관리 | 고급 스택 구성 요소 | 스택 | 삭제

전체 스택을 삭제하지 않고 IAM 역할을 삭제하려는 경우 CloudFormation 템플릿에서 IAM 역할을 제거하고 업데이트된 템플릿을 자동 스택 업데이트 변경 유형에 대한 입력으로 사용할 수 있습니다.

- 유형 ID 변경: ct-361tlo1k7339x
- 분류: 관리 | 사용자 지정 스택 | CloudFormation 템플릿의 스택 | 업데이트

지침은 AWS CloudFormation 수집 스택 업데이트를 참조하세요.

CodeDeploy 요청

AWS CodeDeploy를 사용하여 애플리케이션 컨테이너를 생성한 다음 CodeDeploy 애플리케이션 그룹을 통해 배포할 수 있습니다. CodeDeploy에 대한 자세한 내용은 AWS CodeDeploy 설명서를 참조하세요.

AWS CodeDeploy 작업에는 다음 프로세스가 포함됩니다.

1. CodeDeploy 애플리케이션을 만듭니다. CodeDeploy 애플리케이션은 배포 중에 올바른 개정, 배포 구성 및 배포 그룹이 참조되도록 CodeDeploy에서 사용하는 이름 또는 컨테이너입니다.
2. CodeDeploy 배포 그룹을 생성합니다. CodeDeploy 배포 그룹은 배포를 대상으로 하는 개별 인스턴스 세트를 정의합니다. AMS에는 EC2용 CodeDeploy 배포 그룹에 대한 별도의 변경 유형이 있습니다.
3. CodeDeploy 배포 그룹을 통해 CodeDeploy 애플리케이션을 배포합니다.

CodeDeploy 애플리케이션

CodeDeploy 애플리케이션을 생성하거나 배포합니다.

CodeDeploy 애플리케이션 생성

콘솔을 사용하여 CodeDeploy 애플리케이션 생성

작동 방식:

1. RFC 생성 페이지로 이동합니다. AMS 콘솔의 왼쪽 탐색 창에서 RFCs 클릭하여 RFCs 목록 페이지를 연 다음 RFC 생성을 클릭합니다.
2. 기본 변경 유형 찾아보기 보기에서 널리 사용되는 변경 유형(CT)을 선택하거나 범주별 선택 보기에서 CT를 선택합니다.
   - 변경 유형별 찾아보기: 빠른 생성 영역에서 인기 있는 CT를 클릭하여 RFC 실행 페이지를 즉시 열 수 있습니다. 빠른 생성으로 이전 CT 버전을 선택할 수 없습니다.

     CTs 정렬하려면 카드 또는 테이블 보기에서 모든 변경 유형 영역을 사용합니다. 어느 보기에서든 CT를 선택한 다음 RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다. 해당하는 경우 RFC 생성 버튼 옆에 이전 버전으로 생성 옵션이 나타납니다.
   - 범주별 선택: 범주, 하위 범주, 항목 및 작업을 선택하면 해당하는 경우 이전 버전으로 생성 옵션이 있는 CT 세부 정보 상자가 열립니다. RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다.
3. RFC 실행 페이지에서 CT 이름 영역을 열어 CT 세부 정보 상자를 확인합니다. 제목은 필수입니다(변경 유형 찾아보기 보기에서 CT를 선택하면 입력됨). 추가 구성 영역을 열어 RFC에 대한 정보를 추가합니다.

   실행 구성 영역에서 사용 가능한 드롭다운 목록을 사용하거나 필요한 파라미터의 값을 입력합니다. 선택적 실행 파라미터를 구성하려면 추가 구성 영역을 엽니다.
4. 완료되면 실행을 클릭합니다. 오류가 없는 경우 RFC가 성공적으로 생성된 페이지에 제출된 RFC 세부 정보와 초기 실행 출력이 표시됩니다.
5. 실행 파라미터 영역을 열어 제출한 구성을 확인합니다. 페이지를 새로 고쳐 RFC 실행 상태를 업데이트합니다. 선택적으로 RFC를 취소하거나 페이지 상단의 옵션을 사용하여 RFC 사본을 생성합니다.

CLI를 사용하여 CodeDeploy 애플리케이션 생성

작동 방식:

1. 인라인 생성(모든 RFC 및 실행 파라미터가 포함된 `create-rfc` 명령을 실행) 또는 템플릿 생성(2개의 JSON 파일을 생성, 하나는 RFC 파라미터용이고 다른 하나는 실행 파라미터용)을 사용하고 두 파일을 입력으로 사용하여 `create-rfc` 명령을 실행합니다. 두 방법 모두 여기에 설명되어 있습니다.
2. 반환된 RFC ID로 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 명령을 제출합니다.

   RFC: `aws amscm get-rfc --rfc-id` *`ID`* 명령을 모니터링합니다.

변경 유형 버전을 확인하려면 다음 명령을 사용합니다.

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> CreateRfc 변경 유형에 대한 스키마의 일부인지 여부에 관계없이 모든 파라미터를 RFC와 함께 사용할 수 있습니다. 예를 들어 RFC 상태가 변경될 때 알림을 받으려면 요청의 --notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}" RFC 파라미터 부분(실행 파라미터 아님)에이 줄을 추가합니다. 모든 CreateRfc 파라미터 목록은 AMS Change Management API 참조를 참조하세요.

인라인 생성:

인라인으로 제공된 실행 파라미터(실행 파라미터를 인라인으로 제공할 때 이스케이프 따옴표)로 RFC 생성 명령을 실행한 다음 반환된 RFC ID를 제출합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
aws amscm create-rfc --change-type-id "ct-0ah3gwb9seqk2" --change-type-version "1.0"
 --title "Stack-Create-CD-App" --execution-parameters "{\"Description\":\"TestCdApp\",
\"VpcId\":\"VPC_ID\",\"StackTemplateId\":\"stm-sft6rv00000000000\",\"Name\":\"Test\",
\"TimeoutInMinutes\":60,\"Parameters\":{\"CodeDeployApplicationName\":\"Test\"}}"
```

템플릿 생성:

1. CodeDeploy 애플리케이션 CT의 실행 파라미터 JSON 스키마를 현재 폴더의 파일로 출력합니다. 이 예제에서는 CreateCDAppParams.json:이라는 이름을 지정합니다.

   ```
   aws amscm get-change-type-version --change-type-id "ct-0ah3gwb9seqk2" --query
    "ChangeTypeVersion.ExecutionInputSchema" --output text > CreateCDAppParams.json
   ```

2. 다음과 같이 JSON 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

   ```
   {
   "Description":                    "Create WP CodeDeploy App",
   "VpcId":                          "VPC_ID",
   ```

```
"StackTemplateId":                  "stm-sft6rv00000000000",
"Name":                             "WpCDApp",
"TimeoutInMinutes":                 60,
"Parameters":   {
    "CodeDeployApplicationName":    "WordPressCDApp"
    }
}
```

3. CreateRfc용 JSON 템플릿을 현재 폴더의 파일로 출력합니다.이 예제에서는 CreateCDAppRfc.json:이라는 이름을 지정합니다.

```
aws amscm create-rfc --generate-cli-skeleton > CreateCDAppRfc.json
```

4. 다음과 같이 JSON 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-0ah3gwb9seqk2",
"Title":                "CD-App-Stack-RFC"
}
```

5. CreateCDAppRfc 파일과 실행 파라미터 파일을 지정하여 RFC를 생성합니다.

```
aws amscm create-rfc --cli-input-json file://CreateCDAppRfc.json --execution-
parameters file://CreateCDAppParams.json
```

응답에서 새 RFC의 ID를 수신하고 이를 사용하여 RFC를 제출하고 모니터링할 수 있습니다. 제출하기 전까지는 RFC가 편집 상태로 유지되고 시작되지 않습니다.

팁

AWS CodeDeploy에 대한 자세한 내용은 AWS CodeDeploy를 사용하여 애플리케이션 생성을 참조하세요.

CodeDeploy 애플리케이션 배포

콘솔을 사용하여 CodeDeploy 애플리케이션 배포

작동 방식:

1. RFC 생성 페이지로 이동합니다. AMS 콘솔의 왼쪽 탐색 창에서 RFCs 클릭하여 RFCs 목록 페이지를 연 다음 RFC 생성을 클릭합니다.
2. 기본 변경 유형 찾아보기 보기에서 널리 사용되는 변경 유형(CT)을 선택하거나 범주별 선택 보기에서 CT를 선택합니다.
   - 변경 유형별 찾아보기: 빠른 생성 영역에서 인기 있는 CT를 클릭하여 RFC 실행 페이지를 즉시 열 수 있습니다. 빠른 생성으로 이전 CT 버전을 선택할 수 없습니다.

     CTs 정렬하려면 카드 또는 테이블 보기에서 모든 변경 유형 영역을 사용합니다. 어느 보기에서든 CT를 선택한 다음 RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다. 해당하는 경우 RFC 생성 버튼 옆에 이전 버전으로 생성 옵션이 나타납니다.
   - 범주별 선택: 범주, 하위 범주, 항목 및 작업을 선택하면 해당하는 경우 이전 버전으로 생성 옵션이 있는 CT 세부 정보 상자가 열립니다. RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다.
3. RFC 실행 페이지에서 CT 이름 영역을 열어 CT 세부 정보 상자를 확인합니다. 제목은 필수입니다(변경 유형 찾아보기 보기에서 CT를 선택하면 입력됨). 추가 구성 영역을 열어 RFC에 대한 정보를 추가합니다.

   실행 구성 영역에서 사용 가능한 드롭다운 목록을 사용하거나 필요한 파라미터의 값을 입력합니다. 선택적 실행 파라미터를 구성하려면 추가 구성 영역을 엽니다.
4. 완료되면 실행을 클릭합니다. 오류가 없는 경우 성공적으로 생성된 RFC 페이지에 제출된 RFC 세부 정보와 초기 실행 출력이 표시됩니다.
5. 실행 파라미터 영역을 열어 제출한 구성을 확인합니다. 페이지를 새로 고쳐 RFC 실행 상태를 업데이트합니다. 선택적으로 RFC를 취소하거나 페이지 상단의 옵션을 사용하여 RFC 사본을 생성합니다.

CLI를 사용하여 CodeDeploy 애플리케이션 배포

작동 방식:

1. 인라인 생성(모든 RFC 및 실행 파라미터가 포함된 `create-rfc` 명령을 실행) 또는 템플릿 생성(2개의 JSON 파일을 생성, 하나는 RFC 파라미터용이고 다른 하나는 실행 파라미터용)을 사용하고 두 파일을 입력으로 사용하여 `create-rfc` 명령을 실행합니다. 두 방법 모두 여기에 설명되어 있습니다.
2. 반환된 RFC ID로 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 명령을 제출합니다.

   RFC: `aws amscm get-rfc --rfc-id` *`ID`* 명령을 모니터링합니다.

변경 유형 버전을 확인하려면 다음 명령을 사용합니다.

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> CreateRfc 변경 유형에 대한 스키마의 일부인지 여부에 관계없이 모든 파라미터를 RFC와 함께 사용할 수 있습니다. 예를 들어 RFC 상태가 변경될 때 알림을 받으려면 요청의 --notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}" RFC 파라미터 부분(실행 파라미터 아님)에이 줄을 추가합니다. 모든 CreateRfc 파라미터 목록은 AMS Change Management API 참조를 참조하세요.

인라인 생성:

인라인으로 제공된 실행 파라미터(실행 파라미터를 인라인으로 제공할 때 이스케이프 따옴표)로 RFC 생성 명령을 실행한 다음 반환된 RFC ID를 제출합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
aws amscm create-rfc --change-type-id "ct-2edc3sd1sqmrb" --change-
type-version "2.0" --title "Stack-Deploy-CD-App" --execution-
parameters "{\"Description\":\"MyCDAppDeployTest\",\"VpcId\":
\"VPC_ID\",\"Name\":\"Test\",\"TimeoutInMinutes\":60,\"Parameters\":
{\"CodeDeployApplicationName\":\"TestCDApp\",\"CodeDeployDeploymentConfigName\":
\"CodeDeployDefault.OneAtATime\",\"CodeDeployDeploymentGroupName\":\"TestCDDepGroup\",
\"CodeDeployIgnoreApplicationStopFailures\":false,\"CodeDeployRevision\":
{\"RevisionType\":\"S3\",\"S3Location\":{\"S3Bucket\":\"amzn-s3-demo-bucket\",
\"S3BundleType\":\"tar\",\"S3Key\":\"TestKey\"}}}}"Test\"}}"
```

템플릿 생성:

1. CodeDeploy 애플리케이션 배포 CT의 실행 파라미터 JSON 스키마를 출력합니다.이 예제에서는 DeployCDAppParams.json:이라는 이름을 지정합니다.

   ```
   aws amscm get-change-type-version --change-type-id "ct-2edc3sd1sqmrb" --query
    "ChangeTypeVersion.ExecutionInputSchema" --output text > DeployCDAppParams.json
   ```

2. 다음과 같이 JSON 파일을 수정합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
{
"Description":                        "Deploy WordPress CodeDeploy Application",
"VpcId":                              "VPC_ID",
"Name":                               "WP CodeDeploy Deployment Group",
"TimeoutInMinutes":                   360,
"Parameters":   {
    "CodeDeployApplicationName":         "WordPressCDApp",
    "CodeDeployDeploymentGroupName":     "WordPressCDDepGroup",
    "CodeDeployIgnoreApplicationStopFailures": false,
    "CodeDeployRevision": {
      "RevisionType": "S3",
      "S3Location": {
        "S3Bucket": "amzn-s3-demo-bucket",
        "S3BundleType": "zip",
        "S3Key": "wordpress.zip" }
        }
    }
}
```

3. CreateRfc용 JSON 템플릿을 현재 폴더의 파일로 출력합니다.이 예제에서는 DeployCDAppRfc.json:이라는 이름을 지정합니다.

```
aws amscm create-rfc --generate-cli-skeleton > DeployCDAppRfc.json
```

4. DeployCDAppRfc.json 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
{
"ChangeTypeVersion":    "2.0",
"ChangeTypeId":         "ct-2edc3sd1sqmrb",
"Title":                "CD-Deploy-For-CD-APP-Stack-RFC"
}
```

5. 실행 파라미터 파일과 DeployCDAppRfc 파일을 지정하여 RFC를 생성합니다.

```
aws amscm create-rfc --cli-input-json file://DeployCDAppRfc.json  --execution-
parameters file://DeployCDAppParams.json
```

응답에서 새 RFC의 ID를 수신하고 이를 사용하여 RFC를 제출하고 모니터링할 수 있습니다. 제출하기 전까지는 RFC가 편집 상태로 유지되고 시작되지 않습니다.

팁

자세한 내용은 CodeDeploy를 사용하여 배포 생성을 참조하세요.

CodeDeploy 배포 그룹

CodeDeploy 애플리케이션 그룹을 생성합니다.

CodeDeploy 배포 그룹 생성

콘솔을 사용하여 CodeDeploy 배포 그룹 생성

작동 방식:

1. RFC 생성 페이지로 이동합니다. AMS 콘솔의 왼쪽 탐색 창에서 RFCs 클릭하여 RFCs 목록 페이지를 연 다음 RFC 생성을 클릭합니다.
2. 기본 변경 유형 찾아보기 보기에서 널리 사용되는 변경 유형(CT)을 선택하거나 범주별 선택 보기에서 CT를 선택합니다.
   - 변경 유형별 찾아보기: 빠른 생성 영역에서 인기 있는 CT를 클릭하여 RFC 실행 페이지를 즉시 열 수 있습니다. 빠른 생성으로 이전 CT 버전을 선택할 수 없습니다.

     CTs 정렬하려면 카드 또는 테이블 보기에서 모든 변경 유형 영역을 사용합니다. 어느 보기에서든 CT를 선택한 다음 RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다. 해당하는 경우 RFC 생성 버튼 옆에 이전 버전으로 생성 옵션이 나타납니다.
   - 범주별 선택: 범주, 하위 범주, 항목 및 작업을 선택하면 해당하는 경우 이전 버전으로 생성 옵션이 있는 CT 세부 정보 상자가 열립니다. RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다.
3. RFC 실행 페이지에서 CT 이름 영역을 열어 CT 세부 정보 상자를 확인합니다. 제목은 필수입니다(변경 유형 찾아보기 보기에서 CT를 선택하면 입력됨). 추가 구성 영역을 열어 RFC에 대한 정보를 추가합니다.

   실행 구성 영역에서 사용 가능한 드롭다운 목록을 사용하거나 필요한 파라미터의 값을 입력합니다. 선택적 실행 파라미터를 구성하려면 추가 구성 영역을 엽니다.
4. 완료되면 실행을 클릭합니다. 오류가 없는 경우 RFC가 성공적으로 생성된 페이지에 제출된 RFC 세부 정보와 초기 실행 출력이 표시됩니다.
5. 실행 파라미터 영역을 열어 제출한 구성을 확인합니다. 페이지를 새로 고쳐 RFC 실행 상태를 업데이트합니다. 선택적으로 RFC를 취소하거나 페이지 상단의 옵션을 사용하여 RFC 사본을 생성합니다.

CLI를 사용하여 CodeDeploy 배포 그룹 생성

작동 방식:

1. 인라인 생성(모든 RFC 및 실행 파라미터가 포함된 `create-rfc` 명령을 실행) 또는 템플릿 생성(2개의 JSON 파일을 생성, 하나는 RFC 파라미터용이고 다른 하나는 실행 파라미터용)을 사용하고 두 파일을 입력으로 사용하여 `create-rfc` 명령을 실행합니다. 두 방법 모두 여기에 설명되어 있습니다.
2. 반환된 RFC ID로 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 명령을 제출합니다.

   RFC: `aws amscm get-rfc --rfc-id` *`ID`* 명령을 모니터링합니다.

변경 유형 버전을 확인하려면 다음 명령을 사용합니다.

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 변경 유형에 대한 스키마의 일부인지 여부에 관계없이 모든 `CreateRfc` 파라미터를 RFC와 함께 사용할 수 있습니다. 예를 들어 RFC 상태가 변경될 때 알림을 받으려면 요청의 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` RFC 파라미터 부분(실행 파라미터 아님)에이 줄을 추가합니다. 모든 CreateRfc 파라미터 목록은 AMS Change Management API 참조를 참조하세요.

인라인 생성:

인라인으로 제공된 실행 파라미터(실행 파라미터를 인라인으로 제공할 때 이스케이프 따옴표)로 RFC 생성 명령을 실행한 다음 반환된 RFC ID를 제출합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
aws amscm create-rfc --change-type-id "ct-2gd0u847qd9d2" --change-type-version
 "1.0" --title "Stack-Create-CD-Dep-Group" --execution-parameters "{\"Description
\":\"TestCdDepGroupRfc\",\"VpcId\":\"VPC_ID\",\"StackTemplateId\":\"stm-
sp9lrk00000000000\",\"Name\":\"MyTestCDDepGroup\",\"TimeoutInMinutes\":60,\"Parameters
\":{\"CodeDeployApplicationName\":\"TestCDApp\",\"CodeDeployAutoScalingGroups\":
[\"TestASG\"],\"CodeDeployDeploymentConfigName\":\"CodeDeployDefault.OneAtATime\",
```

```
\"CodeDeployDeploymentGroupName\":\"Test\",\"CodeDeployServiceRoleArn\":
\"arn:aws:iam::000000000:role/aws-codedeploy-role\"}}"
```

템플릿 생성:

1. 실행 파라미터 JSON 스키마를 현재 폴더의 파일로 출력합니다.이 예제에서는 CreateCDDepGroupParams.json:이라는 이름을 지정합니다.

```
aws amscm get-change-type-version --change-type-id "ct-2gd0u847qd9d2"
 --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
 CreateCDDepGroupParams.json
```

2. JSON 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
{
"Description":                      "CreateCDDeploymentGroup",
"VpcId":                            "VPC_ID",
"StackTemplateId":                  "stm-sp9lrk00000000000",
"Name":                             "WordPressCDAppGroup",
"TimeoutInMinutes":                 60,
"Parameters":   {
    "CodeDeployApplicationName":        "WordPressCDApp",
    "CodeDeployAutoScalingGroups":      ["ASG_NAME"],
    "CodeDeployDeploymentConfigName":   "CodeDeployDefault.HalfAtATime",
    "CodeDeployDeploymentGroupName":    "UNIQUE_CDDepGroupNAME",
    "CodeDeployServiceRoleArn":         "arn:aws:iam::ACCOUNT_ID:role/aws-
codedeploy-role"
    }
}
```

3. CreateRfc용 JSON 템플릿을 현재 폴더의 파일로 출력합니다.이 예제에서는 CreateCDDepGroupRfc.json:이라는 이름을 지정합니다.

```
aws amscm create-rfc --generate-cli-skeleton > CreateCDDepGroupRfc.json
```

4. JSON 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-2gd0u847qd9d2",
"Title":                "CD-Dep-Group-RFC"
}
```

5. CreateCDDepGroupRfc 파일과 실행 파라미터 파일을 지정하여 RFC를 생성합니다.

```
aws amscm create-rfc --cli-input-json file://CreateCDDepGroupRfc.json --execution-
parameters file://CreateCDDepGroupParams.json
```

응답에서 새 RFC의 ID를 수신하고 이를 사용하여 RFC를 제출하고 모니터링할 수 있습니다. 제출하기 전까지는 RFC가 편집 상태로 유지되고 시작되지 않습니다.

팁

AWS CodeDeploy 배포 그룹에 대한 자세한 내용은 AWS CodeDeploy를 사용하여 배포 그룹 생성을 참조하세요.

EC2용 CodeDeploy 배포 그룹 생성

콘솔을 사용하여 EC2용 CodeDeploy 배포 그룹 생성

작동 방식:

1. RFC 생성 페이지로 이동합니다. AMS 콘솔의 왼쪽 탐색 창에서 RFCs 클릭하여 RFCs 목록 페이지를 연 다음 RFC 생성을 클릭합니다.
2. 기본 변경 유형 찾아보기 보기에서 인기 있는 변경 유형(CT)을 선택하거나 범주별 선택 보기에서 CT를 선택합니다.
   - 변경 유형별 찾아보기: 빠른 생성 영역에서 인기 있는 CT를 클릭하여 RFC 실행 페이지를 즉시 열 수 있습니다. 빠른 생성으로 이전 CT 버전을 선택할 수 없습니다.

     CTs 정렬하려면 카드 또는 테이블 보기에서 모든 변경 유형 영역을 사용합니다. 어느 보기에서든 CT를 선택한 다음 RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다. 해당하는 경우 RFC 생성 버튼 옆에 이전 버전으로 생성 옵션이 나타납니다.
   - 범주별 선택: 범주, 하위 범주, 항목 및 작업을 선택하면 해당하는 경우 이전 버전으로 생성 옵션이 있는 CT 세부 정보 상자가 열립니다. RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다.
3. RFC 실행 페이지에서 CT 이름 영역을 열어 CT 세부 정보 상자를 확인합니다. 제목은 필수입니다(변경 유형 찾아보기 보기에서 CT를 선택하면 입력됨). 추가 구성 영역을 열어 RFC에 대한 정보를 추가합니다.

   실행 구성 영역에서 사용 가능한 드롭다운 목록을 사용하거나 필요한 파라미터의 값을 입력합니다. 선택적 실행 파라미터를 구성하려면 추가 구성 영역을 엽니다.

4. 완료되면 실행을 클릭합니다. 오류가 없는 경우 RFC가 성공적으로 생성된 페이지에 제출된 RFC 세부 정보와 초기 실행 출력이 표시됩니다.
5. 실행 파라미터 영역을 열어 제출한 구성을 확인합니다. 페이지를 새로 고쳐 RFC 실행 상태를 업데이트합니다. 선택적으로 RFC를 취소하거나 페이지 상단의 옵션을 사용하여 RFC 사본을 생성합니다.

CLI를 사용하여 EC2용 CodeDeploy 배포 그룹 생성

작동 방식:

1. 인라인 생성(모든 RFC 및 실행 파라미터가 포함된 `create-rfc` 명령을 실행) 또는 템플릿 생성(2개의 JSON 파일을 생성, 하나는 RFC 파라미터용이고 다른 하나는 실행 파라미터용)을 사용하고 두 파일을 입력으로 사용하여 `create-rfc` 명령을 실행합니다. 두 방법 모두 여기에 설명되어 있습니다.
2. 반환된 RFC ID로 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 명령을 제출합니다.

   RFC: `aws amscm get-rfc --rfc-id` *`ID`* 명령을 모니터링합니다.

변경 유형 버전을 확인하려면 다음 명령을 사용합니다.

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 변경 유형에 대한 스키마의 일부인지 여부에 관계없이 모든 `CreateRfc` 파라미터를 RFC와 함께 사용할 수 있습니다. 예를 들어 RFC 상태가 변경될 때 알림을 받으려면 요청의 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` RFC 파라미터 부분(실행 파라미터 아님)에이 줄을 추가합니다. 모든 CreateRfc 파라미터 목록은 AMS Change Management API 참조를 참조하세요.

인라인 생성:

인라인으로 제공된 실행 파라미터(실행 파라미터를 인라인으로 제공할 때 이스케이프 따옴표)로 RFC 생성 명령을 실행한 다음 반환된 RFC ID를 제출합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
aws amscm create-rfc --change-type-id "ct-00tlkda4242x7" --change-type-
version "1.0" --title "Stack-Create-CD-Ec2-Dep-Group" --execution-parameters
 "{\"Description\":\"MyTestCdDepEc2DepGroup\",\"VpcId\":\"VPC_ID\",\"Name\":
\"TestCDDepEc2Group\",\"StackTemplateId\":\"stm-n3hsoirgqeqqdbpk2\",\"TimeoutInMinutes
\":60,\"Parameters\":{\"ApplicationName\":\"TestCDApp\",\"DeploymentConfigName\":
\"CodeDeployDefault.OneAtATime\",\"AutoRollbackEnabled\":\"False\",\"EC2FilterTag\":
\"Name=Test\",\"EC2FilterTag2\":\"\",\"EC2FilterTag3\":\"\",\"ServiceRoleArn\":\"\"}}"
```

템플릿 생성:

1. 실행 파라미터 JSON 스키마를 파일로 출력합니다.이 예제에서는 CreateCDDepGroupEc2Params.json:이라는 이름을 지정합니다.

   ```
   aws amscm get-change-type-version --change-type-id "ct-00tlkda4242x7"
    --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
    CreateCDDepGroupEc2Params.json
   ```

2. JSON 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

   ```
   {
   "Description":                      "CreateCDDepGroupEc2",
   "VpcId":                            "VPC_ID",
   "StackTemplateId":                  "stm-n3hsoirgqeqqdbpk2",
   "Name":                             "CDAppGroupEc2",
   "TimeoutInMinutes":                 60,
   "Parameters":   {
       "ApplicationName":        "CDAppEc2",
       "DeploymentConfigName":   "CodeDeployDefault.OneAtATime",
       "CodeDeployDeploymentGroupName":    "UNIQUE_CDDepGroupNAME",
       "CodeDeployServiceRoleArn":         "arn:aws:iam::ACCOUNT_ID:role/aws-
   codedeploy-role"
       }
   }
   ```

3. CreateRfc용 JSON 템플릿을 현재 폴더의 파일로 출력합니다.이 예제에서는 CreateCDDepGroupEc2Rfc.json:이라는 이름을 지정합니다.

   ```
   aws amscm create-rfc --generate-cli-skeleton > CreateCDDepGroupEc2Rfc.json
   ```

4. JSON 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

   ```
   {
   ```

```
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-00tlkda4242x7",
"Title":                "CD-Dep-Group-For-Ec2-Stack-RFC"
}
```

5. CreateCDDepGroupEc2Rfc 파일과 실행 파라미터 파일을 지정하여 RFC를 생성합니다.

```
aws amscm create-rfc --cli-input-json file://CreateCDDepGroupEc2Rfc.json --
execution-parameters file://CreateCDDepGroupEc2Params.json
```

응답에서 새 RFC의 ID를 수신하고 이를 사용하여 RFC를 제출하고 모니터링할 수 있습니다. 제출하기 전까지는 RFC가 편집 상태로 유지되고 시작되지 않습니다.

팁

AWS CodeDeploy 배포 그룹에 대한 자세한 내용은 AWS CodeDeploy를 사용하여 배포 그룹 생성을 참조하세요.

AWS Database Migration Service (AWS DMS)

AWS Database Migration Service (AWS DMS)를 사용하면 데이터베이스를 AMS로 쉽고 안전하게 마이그레이션할 수 있습니다. Oracle, MySQL, PostgreSQL와 같이 가장 널리 사용되는 상용 및 오픈 소스 데이터베이스의 데이터를 마이그레이션할 수 있습니다. 이 서비스는 Oracle에서 Oracle로의 동종 마이그레이션과 Oracle에서 PostgreSQL로의 마이그레이션 또는 MySQL에서 Oracle로의 마이그레이션을 지원합니다. AWS DMS 는 AWS 서비스입니다. AMS CTs AMS 관리형 계정에서 AWS DMS 리소스를 생성하는 데 도움이 됩니다.

다음 그림은 데이터베이스 마이그레이션의 워크플로를 보여줍니다.

주제

- AWS Database Migration Service 시작하기 전에 (AWS DMS)
- AWS DMS, 설정에 필요한 데이터
- AWS DMS 설정 작업
- AWS DMS 관리

AWS Database Migration Service 시작하기 전에 (AWS DMS)

AMS를 사용하여 데이터베이스 마이그레이션을 계획할 때는 다음 사항을 AWS DMS고려하세요.

- 소스 및 대상 엔드포인트: 소스 데이터베이스의 어떤 정보와 테이블을 대상 데이터베이스로 마이그레이션해야 하는지 알아야 합니다. AMS는 테이블 및 기본 키 생성을 포함한 기본 스키마 마이그레이션을 AWS DMS 지원합니다. 그러나 AMS AWS DMS 는 대상 데이터베이스에서 보조 인덱스, 외래 키, 계정 등을 자동으로 생성하지 않습니다. 자세한 내용은 데이터 마이그레이션 소스 및 데이터 마이그레이션 대상을 참조하세요.
- 스키마/코드 마이그레이션: AMS AWS DMS 는 스키마 또는 코드 변환을 수행하지 않습니다. Oracle SQL Developer, MySQL Workbench 또는 pgAdmin III와 같은 도구를 사용하여 스키마를 변환할 수 있습니다. 기존 스키마를 다른 데이터베이스 엔진으로 변환하려면 AWS Schema Conversion Tool을 사용할 수 있습니다. 대상 스키마를 생성하고 전체 스키마(테이블, 인덱스, 보기 등)을 생성할 수도 있습니다. 또한 PL/SQL 또는 TSQL을 PgSQL 및 기타 형식으로 변환하는 도구를 사용할 수도 있습니다.
- 지원되지 않는 데이터 형식: 일부 소스 데이터 형식을 대상 데이터베이스와 동일한 데이터 형식으로 변환해야 합니다.

AWS DMS 고려해야 할 시나리오

다음과 같은 문서화된 시나리오는 자체 데이터베이스 마이그레이션 경로를 만드는 데 도움이 될 수 있습니다.

- 온프레미스 MySQL 서버에서 Amazon RDS MySQL로 데이터 마이그레이션: AWS 블로그 게시물 온프레미스 MySQL 데이터를 Amazon RDS로 마이그레이션(및 뒤로)을 참조하세요.
- Oracle 데이터베이스에서 Amazon RDS Aurora PostgreSQL 데이터베이스로 데이터 마이그레이션: AWS 블로그 게시물 참조 Oracle 데이터베이스에서 Amazon Aurora PostgreSQL 데이터베이스로 마이그레이션하는 방법에 대한 간략한 소개
- RDS MySQL에서 S3로 데이터 마이그레이션: AWS 블로그 게시물 AWS DMS를 사용하여 관계형 데이터베이스의 데이터를 Amazon Glacier로 아카이브하는 방법 참조

데이터베이스 마이그레이션을 위해서는 다음 작업을 수행해야 합니다.

- 데이터베이스 마이그레이션을 계획합니다. 여기에는 복제 서브넷 그룹 설정이 포함됩니다.
- 마이그레이션을 위한 모든 프로세스를 수행하는 복제 인스턴스를 할당합니다.
- 소스 및 대상 데이터베이스 엔드포인트를 지정합니다.

- 하나의 작업이나 작업 집합을 생성하여 사용할 테이블과 복제 프로세스를 정의합니다.
- AWS DMS IAM `dms-cloudwatch-logs-role` 및 `dms-vpc-role` 역할을 생성합니다. Amazon Redshift를 대상 데이터베이스로 사용하는 경우 IAM 역할도 생성하고 `dms-access-for-endpoint` AWS 계정에 추가해야 합니다. 자세한 내용은 AWS CLI 및 AWS DMS API와 함께 사용할 IAM 역할 생성을 참조하세요.

이 연습에서는 AMS 콘솔 또는 AMS CLI를 사용하여 AWS Database Migration Service ()를 생성하는 예를 제공합니다AWS DMS. AWS DMS 복제 인스턴스, 서브넷 그룹 및 작업과 AWS DMS 소스 엔드포인트 및 대상 엔드포인트를 생성하기 위한 CLI 명령이 제공됩니다.

AMS에 대해 자세히 알아보려면 일반 정보는 AWS Database Migration Service 섹션을 AWS DMS참조하고 일반적인 질문에 대한 답변은 AWS Database Migration Service FAQs를 참조하세요.

AWS DMS, 설정에 필요한 데이터

다음 각 AWS DMS 연습에 대해 몇 가지 일반적인 데이터가 필요합니다.

- `Description`: 리소스에 대한 의미 있는 정보로, 다른 파라미터 `Description` 옵션과는 별개입니다.
- `VpcId`: 사용할 VPC입니다. SKMS API의 ListVpcSummaries 작업(`list-vpc-summaries`CLI에서)을 실행하거나 AMS 콘솔의 VPCs 페이지에서 확인할 수 있습니다. AMS SKMS API 참조는 AWS 아티팩트 콘솔의 보고서 탭을 참조하세요.
- `Name`: 스택 또는 스택 구성 요소의 이름입니다. 스택 이름이 됩니다.
- `TimeoutInMinutes`: RFC가 실패하기 전에 스택을 생성하는 데 허용되는 분 수입니다. 이 설정은 RFC 실행을 지연시키지 않지만 충분한 시간을 주어야 합니다(예:를 지정하지 않음`"5"`).
- `ChangeTypeId`, `ChangeTypeVersion`및 `StackTemplateId`: 필수 사항이지만 CT에 따라 다르며 해당 값은 다음 각 관련 섹션에 나와 있습니다.

AWS DMS 설정 작업

다음 연습 AWS DMS 을 통해를 설정합니다.

1: AWS DMS 복제 서브넷 그룹: 생성

AMS 콘솔 또는 API/CLI를 사용하여 AMS AWS DMS 복제 서브넷 그룹을 생성할 수 있습니다.

AWS DMS 복제 서브넷 그룹 생성

콘솔을 사용하여 AWS DMS 복제 서브넷 그룹 생성

> **Note**
>
> 계정에 `dms-vpc-role` IAM 역할이 없는 경우이 CT가 실패합니다.

작동 방식:

1. RFC 생성 페이지로 이동합니다. AMS 콘솔의 왼쪽 탐색 창에서 RFCs 클릭하여 RFCs 목록 페이지를 연 다음 RFC 생성을 클릭합니다.
2. 기본 변경 유형 찾아보기 보기에서 널리 사용되는 변경 유형(CT)을 선택하거나 범주별 선택 보기에서 CT를 선택합니다.
   - 변경 유형별 찾아보기: 빠른 생성 영역에서 인기 있는 CT를 클릭하여 RFC 실행 페이지를 즉시 열 수 있습니다. 빠른 생성으로 이전 CT 버전을 선택할 수 없습니다.

     CTs 정렬하려면 카드 또는 테이블 보기에서 모든 변경 유형 영역을 사용합니다. 어느 보기에서든 CT를 선택한 다음 RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다. 해당하는 경우 RFC 생성 버튼 옆에 이전 버전으로 생성 옵션이 나타납니다.
   - 범주별 선택: 범주, 하위 범주, 항목 및 작업을 선택하면 해당하는 경우 이전 버전으로 생성 옵션이 있는 CT 세부 정보 상자가 열립니다. RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다.
3. RFC 실행 페이지에서 CT 이름 영역을 열어 CT 세부 정보 상자를 확인합니다. 제목은 필수입니다(변경 유형 찾아보기 보기에서 CT를 선택하면 입력됨). 추가 구성 영역을 열어 RFC에 대한 정보를 추가합니다.

   실행 구성 영역에서 사용 가능한 드롭다운 목록을 사용하거나 필요한 파라미터의 값을 입력합니다. 선택적 실행 파라미터를 구성하려면 추가 구성 영역을 엽니다.
4. 완료되면 실행을 클릭합니다. 오류가 없는 경우 RFC가 성공적으로 생성된 페이지에 제출된 RFC 세부 정보와 초기 실행 출력이 표시됩니다.
5. 실행 파라미터 영역을 열어 제출한 구성을 확인합니다. 페이지를 새로 고쳐 RFC 실행 상태를 업데이트합니다. 선택적으로 RFC를 취소하거나 페이지 상단의 옵션을 사용하여 RFC 사본을 생성합니다.

CLI를 사용하여 AWS DMS 복제 서브넷 그룹 생성

> **Note**
>
> 계정에 dms-vpc-role IAM 역할이 없는 경우이 CT가 실패합니다.

작동 방식:

1. 인라인 생성(모든 RFC 및 실행 파라미터가 포함된 create-rfc 명령을 실행) 또는 템플릿 생성(2개의 JSON 파일을 생성, 하나는 RFC 파라미터용이고 다른 하나는 실행 파라미터용)을 사용하고 두 파일을 입력으로 사용하여 create-rfc 명령을 실행합니다. 두 방법 모두 여기에 설명되어 있습니다.
2. 반환된 RFC ID로 RFC: `aws amscm submit-rfc --rfc-id ID` 명령을 제출합니다.

   RFC: `aws amscm get-rfc --rfc-id ID` 명령을 모니터링합니다.

변경 유형 버전을 확인하려면 다음 명령을 사용합니다.

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> CreateRfc 변경 유형에 대한 스키마의 일부인지 여부에 관계없이 모든 파라미터를 RFC와 함께 사용할 수 있습니다. 예를 들어 RFC 상태가 변경될 때 알림을 받으려면 요청의 --notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}" RFC 파라미터 부분(실행 파라미터 아님)에이 줄을 추가합니다. 모든 CreateRfc 파라미터 목록은 AMS Change Management API 참조를 참조하세요.

인라인 생성:

인라인으로 제공된 실행 파라미터(실행 파라미터를 인라인으로 제공할 때 따옴표 이스케이프)로 RFC 생성 명령을 실행한 다음 반환된 RFC ID를 제출합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
aws --profile saml --region us-east-1 amscm create-rfc --change-type-id
 "ct-2q5azjd8p1ag5" --change-type-version "1.0" --title "TestDMSRepSG" --execution-
```

```
parameters "{\"Description\":\"DMSTestRepSG\",\"VpcId\":\"VPC-ID\",\"Name\":\"Test
 Stack\",\"Parameters\":{\"Description\":\"DESCRIPTION\",\"SubnetIds\":[\"SUBNET-ID\",
 \"SUBNET-ID\"]},\"TimeoutInMinutes\":60,\"StackTemplateId\":\"stm-j637f96ls1h4oy5fj
\"}"
```

템플릿 생성:

1. 이 변경 유형의 실행 파라미터를 JSON 파일로 출력합니다.이 예제에서는 CreateDmsRsgParams.json:이라는 이름을 지정합니다.

   ```
   aws amscm get-change-type-version --change-type-id "ct-2q5azjd8p1ag5" --query
    "ChangeTypeVersion.ExecutionInputSchema" --output text > CreateDmsRsgParams.json
   ```

2. 실행 파라미터 CreateDmsRsgParams.json 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

   ```
   {
   "Description":          "DMSTestRepSG",
   "VpcId":                "VPC_ID",
   "TimeoutInMinutes":     60,
   "StackTemplateId":      "stm-j637f96ls1h4oy5fj",
   "Name":                 "Test RSG",
   "Parameters":   {
       "Description":            "DESCRIPTION",
       "SubnetIds":              ["SUBNET_ID", "SUBNET_ID"]
       }
   }
   ```

3. JSON 템플릿을 현재 폴더의 파일로 출력합니다.이 예제에서는 CreateDmsRsgRfc.json:이라는 이름을 지정합니다.

   ```
   aws amscm create-rfc --generate-cli-skeleton > CreateDmsRsgRfc.json
   ```

4. CreateDmsRsgRfc.json 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

   ```
   {
   "ChangeTypeVersion":    "1.0",
   "ChangeTypeId":         "ct-2q5azjd8p1ag5",
   "Title":                "DMS-RSG-Create-RFC"
   }
   ```

5. 실행 파라미터 파일과 CreateDmsRsgRfc 파일을 지정하여 RFC를 생성합니다.

```
aws amscm create-rfc --cli-input-json file://CreateDmsRsgRfc.json --execution-
parameters file://CreateDmsRsgParams.json
```

응답에서 새 RFC의 ID를 수신하고 이를 사용하여 RFC를 제출하고 모니터링할 수 있습니다. 제출하기 전까지는 RFC가 편집 상태로 유지되고 시작되지 않습니다.

팁

- 계정에 `dms-vpc-role` IAM 역할이 없는 경우이 CT가 실패합니다.
- 최대 50개의 태그를 추가할 수 있지만 추가하려면 추가 구성 보기를 활성화해야 합니다.

DMS 복제 인스턴스 및 서브넷 그룹에 대한 자세한 내용은 복제 인스턴스에 대한 네트워크 설정을 참조하세요.

2: AWS DMS 복제 인스턴스: 생성

AMS 콘솔 또는 API/CLI를 사용하여 AMS AWS DMS 복제 인스턴스를 생성할 수 있습니다.

AWS DMS 복제 인스턴스 생성

콘솔을 사용하여 AWS DMS 복제 인스턴스 생성

AMS 콘솔에서이 변경 유형의 스크린샷:

작동 방식:

1. RFC 생성 페이지로 이동합니다. AMS 콘솔의 왼쪽 탐색 창에서 RFCs 클릭하여 RFCs 목록 페이지를 연 다음 RFC 생성을 클릭합니다.
2. 기본 변경 유형 찾아보기 보기에서 널리 사용되는 변경 유형(CT)을 선택하거나 범주별 선택 보기에서 CT를 선택합니다.
   - 변경 유형별 찾아보기: 빠른 생성 영역에서 인기 있는 CT를 클릭하여 RFC 실행 페이지를 즉시 열 수 있습니다. 빠른 생성으로 이전 CT 버전을 선택할 수 없습니다.

     CTs 정렬하려면 카드 또는 테이블 보기에서 모든 변경 유형 영역을 사용합니다. 어느 보기에서든 CT를 선택한 다음 RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다. 해당하는 경우 RFC 생성 버튼 옆에 이전 버전으로 생성 옵션이 나타납니다.

- 범주별 선택: 범주, 하위 범주, 항목 및 작업을 선택하면 해당하는 경우 이전 버전으로 생성 옵션이 있는 CT 세부 정보 상자가 열립니다. RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다.

3. RFC 실행 페이지에서 CT 이름 영역을 열어 CT 세부 정보 상자를 확인합니다. 제목은 필수입니다(변경 유형 찾아보기 보기에서 CT를 선택하면 입력됨). 추가 구성 영역을 열어 RFC에 대한 정보를 추가합니다.

   실행 구성 영역에서 사용 가능한 드롭다운 목록을 사용하거나 필요한 파라미터의 값을 입력합니다. 선택적 실행 파라미터를 구성하려면 추가 구성 영역을 엽니다.

4. 완료되면 실행을 클릭합니다. 오류가 없는 경우 RFC가 성공적으로 생성된 페이지에 제출된 RFC 세부 정보와 초기 실행 출력이 표시됩니다.

5. 실행 파라미터 영역을 열어 제출한 구성을 확인합니다. 페이지를 새로 고쳐 RFC 실행 상태를 업데이트합니다. 선택적으로 RFC를 취소하거나 페이지 상단의 옵션을 사용하여 RFC 사본을 생성합니다.

CLI를 사용하여 AWS DMS 복제 인스턴스 생성

작동 방식:

1. 인라인 생성(모든 RFC 및 실행 파라미터가 포함된 `create-rfc` 명령을 실행) 또는 템플릿 생성(2개의 JSON 파일을 생성, 하나는 RFC 파라미터용이고 다른 하나는 실행 파라미터용)을 사용하고 두 파일을 입력으로 사용하여 `create-rfc` 명령을 실행합니다. 두 방법 모두 여기에 설명되어 있습니다.
2. 반환된 RFC ID로 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 명령을 제출합니다.

   RFC: `aws amscm get-rfc --rfc-id` *`ID`* 명령을 모니터링합니다.

변경 유형 버전을 확인하려면 다음 명령을 사용합니다.

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 변경 유형에 대한 스키마의 일부인지 여부에 관계없이 모든 `CreateRfc` 파라미터를 RFC와 함께 사용할 수 있습니다. 예를 들어 RFC 상태가 변경될 때 알림을 받으려면 요청의 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com`

\"]}}" RFC 파라미터 부분(실행 파라미터 아님)에이 줄을 추가합니다. 모든 CreateRfc 파라미터 목록은 AMS Change Management API 참조를 참조하세요.

인라인 생성:

인라인으로 제공된 실행 파라미터(실행 파라미터를 인라인으로 제공할 때 따옴표 이스케이프)로 RFC 생성 명령을 실행한 다음 반환된 RFC ID를 제출합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
aws --profile saml --region us-east-1 amscm create-rfc --change-type-id
 "ct-27apldkhqr0ol" --change-type-version "1.0" --title "TestDMSRepInstance" --
execution-parameters "{\"Description\":\"DMSTestRepInstance\",\"VpcId\":\"VPC-ID\",
\"Name\":\"REP-INSTANCE-NAME\",\"Parameters\":{\"InstanceClass\":\"dms.t2.micro\",
\"ReplicationSubnetGroupIdentifier\":\"TEST-REP-SG\",\"SecurityGroupIds\":\"SG-ID, SG-
ID\"},\"TimeoutInMinutes\":60,\"StackTemplateId\":\"stm-3n1j5hdrmiiiuqk6v\"}"
```

복제 인스턴스가 생성되는 동안 원본과 대상 데이터 스토어를 지정할 수 있습니다. 소스 및 대상 데이터 스토어는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스, AWS S3 버킷, Amazon Relational Database Service(Amazon RDS) DB 인스턴스 또는 온프레미스 데이터베이스에 있을 수 있습니다.

템플릿 생성:

1. 이 변경 유형의 실행 파라미터를 JSON 파일로 출력합니다.이 예제에서는 CreateDmsRiParams.json:이라는 이름을 지정합니다.

   ```
   aws amscm get-change-type-version --change-type-id "ct-27apldkhqr0ol" --query
    "ChangeTypeVersion.ExecutionInputSchema" --output text > CreateDmsRiParams.json
   ```

2. 실행 파라미터 CreateDmsRiParams.json 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

   ```
   {
   "Description":          "DMSTestRepInstance",
   "VpcId":                "VPC_ID",
   "Name":                 "Test RI",
   "StackTemplateId":      "stm-3n1j5hdrmiiiuqk6v",
   "TimeoutInMinutes":     60,
   "Parameters":   {
   ```

```
    "Description":                          "DESCRIPTION",
    "InstanceClass":                        "dms.t2.micro",
    "ReplicationSubnetGroupIdentifier": "TEST-REP-SG",
    "SecurityGroupIds":                     ["SG-ID, SG-ID"}
    }
}
```

3. JSON 템플릿을 현재 폴더의 파일로 출력합니다.이 예제에서는 CreateDmsRiRfc.json:이라는 이름을 지정합니다.

```
aws amscm create-rfc --generate-cli-skeleton > CreateDmsRiRfc.json
```

4. CreateDmsRiRfc.json 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-27apldkhqr0ol",
"Title":                "DMS-RI-Create-RFC"
}
```

5. 실행 파라미터 파일과 CreateDmsRiRfc 파일을 지정하여 RFC를 생성합니다.

```
aws amscm create-rfc --cli-input-json file://CreateDmsRiRfc.json --execution-
parameters file://CreateDmsRiParams.json
```

응답에서 새 RFC의 ID를 수신하고 이를 사용하여 RFC를 제출하고 모니터링할 수 있습니다. 제출하기 전까지는 RFC가 편집 상태로 유지되고 시작되지 않습니다.

팁

- 최대 50개의 태그를 추가할 수 있지만, 추가하려면 추가 구성 보기를 활성화해야 합니다.
- 소스 데이터베이스에서 대상 데이터베이스로 데이터를 할당하고 마이그레이션하는 작업을 수행하기에 충분한 스토리지 및 처리 능력이 있는 AMS VPC의 EC2 인스턴스에 복제 인스턴스를 생성해야 합니다. 이 인스턴스의 필요한 크기는 마이그레이션해야 하는 데이터 양과 인스턴스 실행에 필요한 작업에 따라 달라집니다. 복제 인스턴스는 `MultiAZ` 옵션을 선택할 때 다중 AZ 배포를 사용하여 고가용성 및 장애 조치 지원을 제공합니다. 복제 인스턴스에 대한 자세한 내용은 AWS DMS 복제 인스턴스 작업을 참조하세요.

3: AWS DMS 소스 엔드포인트: 생성, Mongo DB용 생성, S3용 생성

AMS 콘솔 또는 API/CLI를 사용하여 다양한 데이터베이스에 대한 AMS DMS 소스 엔드포인트를 생성할 수 있습니다. 세 가지 예를 제공합니다.

DMS 소스 엔드포인트: 생성

콘솔을 사용하여 DMS 소스 엔드포인트 생성

AMS 콘솔에서이 변경 유형의 스크린샷:

작동 방식:

1. RFC 생성 페이지로 이동합니다. AMS 콘솔의 왼쪽 탐색 창에서 RFCs 클릭하여 RFCs 목록 페이지를 연 다음 RFC 생성을 클릭합니다.
2. 기본 변경 유형 찾아보기 보기에서 널리 사용되는 변경 유형(CT)을 선택하거나 범주별 선택 보기에서 CT를 선택합니다.
   - 변경 유형별 찾아보기: 빠른 생성 영역에서 인기 있는 CT를 클릭하여 RFC 실행 페이지를 즉시 열 수 있습니다. 빠른 생성으로 이전 CT 버전을 선택할 수 없습니다.

     CTs 정렬하려면 카드 또는 테이블 보기에서 모든 변경 유형 영역을 사용합니다. 어느 보기에서든 CT를 선택한 다음 RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다. 해당하는 경우 RFC 생성 버튼 옆에 이전 버전으로 생성 옵션이 나타납니다.
   - 범주별 선택: 범주, 하위 범주, 항목 및 작업을 선택하면 해당하는 경우 이전 버전으로 생성 옵션이 있는 CT 세부 정보 상자가 열립니다. RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다.
3. RFC 실행 페이지에서 CT 이름 영역을 열어 CT 세부 정보 상자를 확인합니다. 제목은 필수입니다(변경 유형 찾아보기 보기에서 CT를 선택하면 입력됨). 추가 구성 영역을 열어 RFC에 대한 정보를 추가합니다.

   실행 구성 영역에서 사용 가능한 드롭다운 목록을 사용하거나 필요한 파라미터의 값을 입력합니다. 선택적 실행 파라미터를 구성하려면 추가 구성 영역을 엽니다.
4. 완료되면 실행을 클릭합니다. 오류가 없는 경우 RFC가 성공적으로 생성된 페이지에 제출된 RFC 세부 정보와 초기 실행 출력이 표시됩니다.
5. 실행 파라미터 영역을 열어 제출한 구성을 확인합니다. 페이지를 새로 고쳐 RFC 실행 상태를 업데이트합니다. 선택적으로 RFC를 취소하거나 페이지 상단의 옵션을 사용하여 RFC 사본을 생성합니다.

CLI를 사용하여 DMS 소스 엔드포인트 생성

작동 방식:

1. 인라인 생성(모든 RFC 및 실행 파라미터가 포함된 `create-rfc` 명령을 실행) 또는 템플릿 생성(2개의 JSON 파일을 생성, 하나는 RFC 파라미터용이고 다른 하나는 실행 파라미터용)을 사용하고 두 파일을 입력으로 사용하여 `create-rfc` 명령을 실행합니다. 두 방법 모두 여기에 설명되어 있습니다.
2. 반환된 RFC ID로 RFC: `aws amscm submit-rfc --rfc-id ID` 명령을 제출합니다.

   RFC: `aws amscm get-rfc --rfc-id ID` 명령을 모니터링합니다.

변경 유형 버전을 확인하려면 다음 명령을 사용합니다.

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 변경 유형에 대한 스키마의 일부인지 여부에 관계없이 모든 `CreateRfc` 파라미터를 RFC와 함께 사용할 수 있습니다. 예를 들어 RFC 상태가 변경될 때 알림을 받으려면 요청의 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` RFC 파라미터 부분(실행 파라미터 아님)에이 줄을 추가합니다. 모든 CreateRfc 파라미터 목록은 AMS Change Management API 참조를 참조하세요.

인라인 생성:

인라인으로 제공된 실행 파라미터(실행 파라미터를 인라인으로 제공할 때 따옴표 이스케이프)로 RFC 생성 명령을 실행한 다음 반환된 RFC ID를 제출합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
aws --profile saml --region us-east-1 amscm create-rfc --title "MariaDB-DMS-
Source-Endpoint" --aws-account-id ACCOUNT-ID --change-type-id ct-0attesnjqy2cx --
change-type-version 1.0 --execution-parameters "{\"Description\":\"DESCRIPTION.\",
\"VpcId\":\"VPC-ID\",\"Name\":\"MariaDB-DMS-SE\",\"Parameters\":{\"EngineName\":
\"mariadb\",\"ServerName\":\"mariadb.db.example.com\",\"Port\":3306,\"Username\":
\"DB-USER\",\"Password\":\"DB-PW\"},\"TimeoutInMinutes\":60,\"StackTemplateId\":\"stm-
pud4ghhkp7395n9bc\"}"
```

템플릿 생성:

1. 이 변경 유형의 실행 파라미터를 CreateDmsSeParams.json.

```
aws amscm get-change-type-version --change-type-id "ct-0attesnjqy2cx" --query
 "ChangeTypeVersion.ExecutionInputSchema" --output text > CreateDmsSeParams.json
```

2. 실행 파라미터 JSON 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
{
"Description":          "MariaDB-DMS-SE",
"VpcId":                "VPC_ID",
"Name":                 "Test SE",
"StackTemplateId":      "stm-pud4ghhkp7395n9bc",
"TimeoutInMinutes":     60,
"Parameters":   {
    "Description":      "DESCRIPTION",
    "EngineName":       "mariadb",
    "ServerName":       "mariadb.db.example.com",
    "Port":             "3306",
    "Username":         "DB-USER",
    "Password":         "DB-PW",}
    }
}
```

3. JSON 템플릿을 현재 폴더의 파일로 출력합니다.이 예제에서는 CreateDmsSeRfc.json:이라는 이름을 지정합니다.

```
aws amscm create-rfc --generate-cli-skeleton > CreateDmsSeRfc.json
```

4. CreateDmsSeRfc.json 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-0attesnjqy2cx",
"Title":                "MariaDB-DMS-Source-Endpoint"
}
```

5. 실행 파라미터 파일과 CreateDmsSeRfc 파일을 지정하여 RFC를 생성합니다.

```
aws amscm create-rfc --cli-input-json file://CreateDmsSeRfc.json --execution-
parameters file://CreateDmsSeParams.json
```

응답에서 새 RFC의 ID를 수신하고 이를 사용하여 RFC를 제출하고 모니터링할 수 있습니다. 제출하기 전까지는 RFC가 편집 상태로 유지되고 시작되지 않습니다.

팁

DMS 엔드포인트를 생성하기 전에 암호에 지원되지 않는 문자가 포함되어 있지 않은지 확인합니다. 자세한 내용은 AWS Database Migration Service 사용 설명서의 소스 및 대상 엔드포인트 생성을 참조하세요.

자세한 내용은 데이터 마이그레이션을 위한 소스를 참조하세요.

S3 소스 엔드포인트는 섹션을 참조하세요S3용 DMS 소스 엔드포인트: 생성.

Mongo DB 소스 엔드포인트는 섹션을 참조하세요MongoDB용 DMS 소스 엔드포인트: 생성.

MongoDB용 DMS 소스 엔드포인트: 생성

콘솔을 사용하여 DMS Mongo DB 소스 엔드포인트 생성

AMS 콘솔에서이 변경 유형의 스크린샷:

작동 방식:

1. RFC 생성 페이지로 이동합니다. AMS 콘솔의 왼쪽 탐색 창에서 RFCs 클릭하여 RFCs 목록 페이지를 연 다음 RFC 생성을 클릭합니다.
2. 기본 변경 유형 찾아보기 보기에서 인기 있는 변경 유형(CT)을 선택하거나 범주별 선택 보기에서 CT를 선택합니다.
   - 변경 유형별 찾아보기: 빠른 생성 영역에서 인기 있는 CT를 클릭하여 RFC 실행 페이지를 즉시 열 수 있습니다. 빠른 생성으로 이전 CT 버전을 선택할 수 없습니다.

     CTs 정렬하려면 카드 또는 테이블 보기에서 모든 변경 유형 영역을 사용합니다. 어느 보기에서든 CT를 선택한 다음 RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다. 해당하는 경우 RFC 생성 버튼 옆에 이전 버전으로 생성 옵션이 나타납니다.
   - 범주별 선택: 범주, 하위 범주, 항목 및 작업을 선택하면 해당하는 경우 이전 버전으로 생성 옵션이 있는 CT 세부 정보 상자가 열립니다. RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다.

3. RFC 실행 페이지에서 CT 이름 영역을 열어 CT 세부 정보 상자를 확인합니다. 제목은 필수입니다(변경 유형 찾아보기 보기에서 CT를 선택하면 입력됨). 추가 구성 영역을 열어 RFC에 대한 정보를 추가합니다.

   실행 구성 영역에서 사용 가능한 드롭다운 목록을 사용하거나 필요한 파라미터의 값을 입력합니다. 선택적 실행 파라미터를 구성하려면 추가 구성 영역을 엽니다.

4. 완료되면 실행을 클릭합니다. 오류가 없는 경우 RFC가 성공적으로 생성된 페이지에 제출된 RFC 세부 정보와 초기 실행 출력이 표시됩니다.
5. 실행 파라미터 영역을 열어 제출한 구성을 확인합니다. 페이지를 새로 고쳐 RFC 실행 상태를 업데이트합니다. 선택적으로 RFC를 취소하거나 페이지 상단의 옵션을 사용하여 RFC 사본을 생성합니다.

CLI를 사용하여 DMS Mongo DB 소스 엔드포인트 생성

작동 방식:

1. 인라인 생성(모든 RFC 및 실행 파라미터가 포함된 `create-rfc` 명령을 실행) 또는 템플릿 생성(2개의 JSON 파일을 생성, 하나는 RFC 파라미터용이고 다른 하나는 실행 파라미터용)을 사용하고 두 파일을 입력으로 사용하여 `create-rfc` 명령을 실행합니다. 두 방법 모두 여기에 설명되어 있습니다.
2. 반환된 RFC ID로 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 명령을 제출합니다.

   RFC: `aws amscm get-rfc --rfc-id` *`ID`* 명령을 모니터링합니다.

변경 유형 버전을 확인하려면 다음 명령을 사용합니다.

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 변경 유형에 대한 스키마의 일부인지 여부에 관계없이 모든 `CreateRfc` 파라미터를 RFC와 함께 사용할 수 있습니다. 예를 들어 RFC 상태가 변경될 때 알림을 받으려면 요청의 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` RFC 파라미터 부분(실행 파라미터 아님)에이 줄을 추가합니다. 모든 CreateRfc 파라미터 목록은 AMS Change Management API 참조를 참조하세요.

인라인 생성:

인라인으로 제공된 실행 파라미터(실행 파라미터를 인라인으로 제공할 때 따옴표 이스케이프)로 RFC 생성 명령을 실행한 다음 반환된 RFC ID를 제출합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
aws amscm  --profile saml --region us-east-1 create-rfc --change-type-id
 "ct-2hxcllf1b4ey0" --change-type-version "1.0" --title 'DMS_Source_MongoDB'
 --description "DESCRIPTION"  --execution-parameters "{\"Description\":
\"DMS_MongoDB_Source_Endpoint\",\"VpcId\":\"VPC_ID\",\"Name\":\"DMS-Mongo-SE\",
\"StackTemplateId\":\"stm-pud4ghhkp7395n9bc\",\"TimeoutInMinutes\":60,\"Parameters\":
{\"DatabaseName\":\"mytestdb\",\"EngineName\":\"mongodb\",\"Port\":27017,\"ServerName
\":\"test.example.com\"}}"
```

템플릿 생성:

1. 이 변경 유형의 실행 파라미터를 CreateDmsSeMongoParams.json.

   ```
   aws amscm get-change-type-version --change-type-id "ct-2hxcllf1b4ey0"
    --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
    CreateDmsSeMongoParams.json
   ```

2. 실행 파라미터 JSON 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

   ```
   {
   "Description":          "MongoDB-DMS-SE",
   "VpcId":                "VPC_ID",
   "StackTemplateId":      "stm-pud4ghhkp7395n9bc",
   "Name":                 "Test Mongo SE",
   "TimeoutInMinutes":     60,
   "Parameters":   {
       "Description":      "DESCRIPTION",
       "DatabaseName":       "mytestdb",
       "EngineName":       "mongodb",
       "ServerName":       "test.example.com",
       "Port":             "27017"
       }
   }
   ```

3. JSON 템플릿을 현재 폴더의 파일로 출력합니다.이 예제에서는 CreateDmsSeMongoRfc.json:이라는 이름을 지정합니다.

```
aws amscm create-rfc --generate-cli-skeleton > CreateDmsSeMongoRfc.json
```

4. CreateDmsSeMongoRfc.json 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-2hxcllf1b4ey0",
"Title":                "DMS_Source_MongoDB"
}
```

5. 실행 파라미터 파일과 CreateDmsSeMongoRfc 파일을 지정하여 RFC를 생성합니다.

```
aws amscm create-rfc --cli-input-json file://CreateDmsSeMongoRfc.json --execution-
parameters file://CreateDmsSeMongoParams.json
```

응답에서 새 RFC의 ID를 수신하고 이를 사용하여 RFC를 제출하고 모니터링할 수 있습니다. 제출하기 전까지는 RFC가 편집 상태로 유지되고 시작되지 않습니다.

팁

> **Note**
>
> 최대 50개의 태그를 추가할 수 있지만 추가하려면 추가 구성 보기를 활성화해야 합니다.

AMS DMS는 Mongo 또는 관계형 데이터베이스 서비스(RDS)를 소스 엔드포인트로 사용할 수 있습니다. S3 소스 엔드포인트는 섹션을 참조하세요S3용 DMS 소스 엔드포인트: 생성.

S3용 DMS 소스 엔드포인트: 생성

콘솔을 사용하여 DMS S3 소스 엔드포인트 생성

AMS 콘솔에서이 변경 유형의 스크린샷:

작동 방식:

1. RFC 생성 페이지로 이동합니다. AMS 콘솔의 왼쪽 탐색 창에서 RFCs 클릭하여 RFCs 목록 페이지를 연 다음 RFC 생성을 클릭합니다.

2. 기본 변경 유형 찾아보기 보기에서 널리 사용되는 변경 유형(CT)을 선택하거나 범주별 선택 보기에서 CT를 선택합니다.
   - 변경 유형별 찾아보기: 빠른 생성 영역에서 인기 있는 CT를 클릭하여 RFC 실행 페이지를 즉시 열 수 있습니다. 빠른 생성으로 이전 CT 버전을 선택할 수 없습니다.

     CTs 정렬하려면 카드 또는 테이블 보기에서 모든 변경 유형 영역을 사용합니다. 어느 보기에서든 CT를 선택한 다음 RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다. 해당하는 경우 RFC 생성 버튼 옆에 이전 버전으로 생성 옵션이 나타납니다.
   - 범주별 선택: 범주, 하위 범주, 항목 및 작업을 선택하면 해당하는 경우 이전 버전으로 생성 옵션이 있는 CT 세부 정보 상자가 열립니다. RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다.
3. RFC 실행 페이지에서 CT 이름 영역을 열어 CT 세부 정보 상자를 확인합니다. 제목은 필수입니다(변경 유형 찾아보기 보기에서 CT를 선택하면 입력됨). 추가 구성 영역을 열어 RFC에 대한 정보를 추가합니다.

   실행 구성 영역에서 사용 가능한 드롭다운 목록을 사용하거나 필요한 파라미터의 값을 입력합니다. 선택적 실행 파라미터를 구성하려면 추가 구성 영역을 엽니다.
4. 완료되면 실행을 클릭합니다. 오류가 없는 경우 RFC가 성공적으로 생성된 페이지에 제출된 RFC 세부 정보와 초기 실행 출력이 표시됩니다.
5. 실행 파라미터 영역을 열어 제출한 구성을 확인합니다. 페이지를 새로 고쳐 RFC 실행 상태를 업데이트합니다. 선택적으로 RFC를 취소하거나 페이지 상단의 옵션을 사용하여 RFC 사본을 생성합니다.

CLI를 사용하여 DMS S3 소스 엔드포인트 생성

작동 방식:

1. 인라인 생성(모든 RFC 및 실행 파라미터가 포함된 `create-rfc` 명령을 실행) 또는 템플릿 생성(2개의 JSON 파일을 생성, 하나는 RFC 파라미터용이고 다른 하나는 실행 파라미터용)을 사용하고 두 파일을 입력으로 사용하여 `create-rfc` 명령을 실행합니다. 두 방법 모두 여기에 설명되어 있습니다.
2. 반환된 RFC ID로 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 명령을 제출합니다.

   RFC: `aws amscm get-rfc --rfc-id` *`ID`* 명령을 모니터링합니다.

변경 유형 버전을 확인하려면 다음 명령을 사용합니다.

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 변경 유형에 대한 스키마의 일부인지 여부에 관계없이 모든 RFC에서 CreateRfc 파라미터를 사용할 수 있습니다. 예를 들어 RFC 상태가 변경될 때 알림을 받으려면 요청의 --notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}" RFC 파라미터 부분(실행 파라미터 아님)에이 줄을 추가합니다. 모든 CreateRfc 파라미터 목록은 AMS Change Management API 참조를 참조하세요.

인라인 생성:

인라인으로 제공된 실행 파라미터(실행 파라미터를 인라인으로 제공할 때 따옴표 이스케이프)로 RFC 생성 명령을 실행한 다음 반환된 RFC ID를 제출합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
aws --profile saml --region us-east-1 amscm create-rfc --title "S3DMSSourceEndpoint" --
aws-account-id ACCOUNT-ID --change-type-id ct-2oxl37nphsrjz --change-type-version 1.0
 --execution-parameters "{\"Description\":\"TestS3DMS-SE\",\"VpcId\":\"VPC-ID\",\"Name
\":\"S3-DMS-SE\",\"Parameters\":{\"EngineName\":\"s3\",\"S3BucketName\":\"amzn-s3-
demo-bucket\",\"S3ExternalTableDefinition\":\"{\\\"TableCount\\\":\\\"1\\\",\\\"Tables
\\\":[{\\\"TableName\\\":\\\"employee\\\",\\\"TablePath\\\":\\\"hr/employee/\\\",\\
\"TableOwner\\\":\\\"hr\\\",\\\"TableColumns\\\":[{\\\"ColumnName\\\":\\\"Id\\\",\\
\"ColumnType\\\":\\\"INT8\\\",\\\"ColumnNullable\\\":\\\"false\\\",\\\"ColumnIsPk\\\":
\\\"true\\\"},{\\\"ColumnName\\\":\\\"LastName\\\",\\\"ColumnType\\\":\\\"STRING\\\",
\\\"ColumnLength\\\":\\\"20\\\"},{\\\"ColumnName\\\":\\\"FirstName\\\",\\\"ColumnType
\\\":\\\"STRING\\\",\\\"ColumnLength\\\":\\\"30\\\"},{\\\"ColumnName\\\":\\\"HireDate\
\\",\\\"ColumnType\\\":\\\"DATETIME\\\"},{\\\"ColumnName\\\":\\\"OfficeLocation\\\",\\
\"ColumnType\\\":\\\"STRING\\\",\\\"ColumnLength\\\":\\\"20\\\"}],\\\"TableColumnsTotal
\\\":\\\"5\\\"}]}\",\"S3ServiceAccessRoleArn\":\"arn:aws:iam::123456789101:role/ams-
ops-ct-authors-dms-s3-test-role\"},\"TimeoutInMinutes\":60,\"StackTemplateId\":\"stm-
pud4ghhkp7395n9bc\"}"
```

템플릿 생성:

1. 이 변경 유형의 실행 파라미터를 CreateDmsSeS3Params.json.

```
aws amscm get-change-type-version --change-type-id "ct-2oxl37nphsrjz" --query
 "ChangeTypeVersion.ExecutionInputSchema" --output text > CreateDmsSeS3Params.json
```

2. 실행 파라미터 JSON 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
{
"Description":          "TestS3DMS-SE",
"VpcId":                "VPC_ID",
"Name":                 "S3-DMS-SE",
"StackTemplateId":      "stm-pud4ghhkp7395n9bc",
"TimeoutInMinutes":     60,
"Parameters":   {
    "EngineName":               "s3",
    "S3BucketName":              "amzn-s3-demo-bucket",
    "S3ExternalTableDefinition":  "BUCKET-NAME",
    {"TableCount":                "1",
      "Tables":[{"TableName":"employee","TablePath":"hr/
employee/","TableOwner":"hr","TableColumns":
[{"ColumnName":"Id","ColumnType":"INT8","ColumnNullable":"false","ColumnIsPk":"true"},
{"ColumnName":"LastName","ColumnType":"STRING","ColumnLength":"20"},
{"ColumnName":"FirstName","ColumnType":"STRING","ColumnLength":"30"},
{"ColumnName":"HireDate","ColumnType":"DATETIME"},
{"ColumnName":"OfficeLocation","ColumnType":"STRING","ColumnLength":"20"}],"TableColumnsTot
    "S3ServiceAccessRoleArn":      "arn:aws:iam::123456789101:role/ams-ops-ct-
authors-dms-s3-test-role",
      }
}
```

3. JSON 템플릿을 현재 폴더의 파일로 출력합니다.이 예제에서는 CreateDmsSeS3Rfc.json:이라는 이름을 지정합니다.

```
aws amscm create-rfc --generate-cli-skeleton > CreateDmsSeS3Rfc.json
```

4. CreateDmsSeS3Rfc.json 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-2oxl37nphsrjz",
"Title":                "DMS_Source_S3"
```

```
}
```

5. 실행 파라미터 파일과 CreateDmsSeS3Rfc 파일을 지정하여 RFC를 생성합니다.

```
aws amscm create-rfc --cli-input-json file://CreateDmsSeS3Rfc.json --execution-
parameters file://CreateDmsSeS3Params.json
```

응답에서 새 RFC의 ID를 수신하고 이를 사용하여 RFC를 제출하고 모니터링할 수 있습니다. 제출하기 전까지는 RFC가 편집 상태로 유지되고 시작되지 않습니다.

팁

> **Note**
>
> 최대 50개의 태그를 추가할 수 있지만 추가하려면 추가 구성 보기를 활성화해야 합니다.

AMS DMS는 S3 또는 관계형 데이터베이스 서비스(RDS) 소스 엔드포인트를 사용할 수 있습니다. Mongo DB 소스 엔드포인트는 섹션을 참조하세요MongoDB용 DMS 소스 엔드포인트: 생성.

4: AWS DMS 대상 엔드포인트: S3에 대한 생성

AMS 콘솔 또는 API/CLI를 사용하여 다양한 데이터베이스에 대한 AMS DMS 대상 엔드포인트를 생성할 수 있습니다. 두 가지 예를 제공합니다.

DMS 대상 엔드포인트: 생성

AMS DMS는 S3 또는 RDS(관계형 데이터베이스 서비스)를 MySQL, MariaDB, Oracle, Postgresql 또는 Microsoft SQL과 함께 대상 엔드포인트로 사용할 수 있습니다.

콘솔을 사용하여 DMS 대상 엔드포인트 생성

AMS 콘솔에서이 변경 유형의 스크린샷:

작동 방식:

1. RFC 생성 페이지로 이동합니다. AMS 콘솔의 왼쪽 탐색 창에서 RFCs 클릭하여 RFCs 목록 페이지를 연 다음 RFC 생성을 클릭합니다.
2. 기본 변경 유형 찾아보기 보기에서 널리 사용되는 변경 유형(CT)을 선택하거나 범주별 선택 보기에서 CT를 선택합니다.

- 변경 유형별 찾아보기: 빠른 생성 영역에서 인기 있는 CT를 클릭하여 RFC 실행 페이지를 즉시 열 수 있습니다. 빠른 생성으로 이전 CT 버전을 선택할 수 없습니다.

  CTs 정렬하려면 카드 또는 테이블 보기에서 모든 변경 유형 영역을 사용합니다. 어느 보기에서든 CT를 선택한 다음 RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다. 해당하는 경우 RFC 생성 버튼 옆에 이전 버전으로 생성 옵션이 나타납니다.
- 범주별 선택: 범주, 하위 범주, 항목 및 작업을 선택하면 해당하는 경우 이전 버전으로 생성 옵션이 있는 CT 세부 정보 상자가 열립니다. RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다.

3. RFC 실행 페이지에서 CT 이름 영역을 열어 CT 세부 정보 상자를 확인합니다. 제목은 필수입니다(변경 유형 찾아보기 보기에서 CT를 선택하면 입력됨). 추가 구성 영역을 열어 RFC에 대한 정보를 추가합니다.

   실행 구성 영역에서 사용 가능한 드롭다운 목록을 사용하거나 필요한 파라미터의 값을 입력합니다. 선택적 실행 파라미터를 구성하려면 추가 구성 영역을 엽니다.
4. 완료되면 실행을 클릭합니다. 오류가 없는 경우 RFC가 성공적으로 생성된 페이지에 제출된 RFC 세부 정보와 초기 실행 출력이 표시됩니다.
5. 실행 파라미터 영역을 열어 제출한 구성을 확인합니다. 페이지를 새로 고쳐 RFC 실행 상태를 업데이트합니다. 선택적으로 RFC를 취소하거나 페이지 상단의 옵션을 사용하여 RFC 사본을 생성합니다.

CLI를 사용하여 DMS 대상 엔드포인트 생성

작동 방식:

1. 인라인 생성(모든 RFC 및 실행 파라미터가 포함된 `create-rfc` 명령을 실행) 또는 템플릿 생성(2개의 JSON 파일을 생성, 하나는 RFC 파라미터용이고 다른 하나는 실행 파라미터용)을 사용하고 두 파일을 입력으로 사용하여 `create-rfc` 명령을 실행합니다. 두 방법 모두 여기에 설명되어 있습니다.
2. 반환된 RFC ID로 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 명령을 제출합니다.

   RFC: `aws amscm get-rfc --rfc-id` *`ID`* 명령을 모니터링합니다.

변경 유형 버전을 확인하려면 다음 명령을 사용합니다.

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> CreateRfc 변경 유형에 대한 스키마의 일부인지 여부에 관계없이 모든 파라미터를 RFC와 함께 사용할 수 있습니다. 예를 들어 RFC 상태가 변경될 때 알림을 받으려면 요청의 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` RFC 파라미터 부분(실행 파라미터 아님)에이 줄을 추가합니다. 모든 CreateRfc 파라미터 목록은 AMS Change Management API 참조를 참조하세요.

인라인 생성:

인라인으로 제공된 실행 파라미터(실행 파라미터를 인라인으로 제공할 때 따옴표 이스케이프)로 RFC 생성 명령을 실행한 다음 반환된 RFC ID를 제출합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
aws --profile saml --region us-east-1 amscm create-rfc --change-type-id
 "ct-3gf8dolbo8x9p" --change-type-version "1.0" --title "TestDMSTargetEndpoint" --
execution-parameters "{\"Description\":\"TestTE\",\"VpcId\":\"VPC-ID\",\"Name\":
\"TE-NAME\",\"StackTemplateId\":\"stm-knghtmmgefafdq89u\",\"TimeoutInMinutes\":60,
\"Parameters\":{\"EngineName\":\"mysql\",\"Password\":\"testpw123\",\"Port\":\"3306\",
\"ServerName\":\"mytestdb.d5fga0rf2wpi.ap-southeast-2.rds.amazonaws.com\",\"Username\":
\"USERNAME\"}}"
```

템플릿 생성:

1. 이 변경 유형의 실행 파라미터를 CreateDmsTeParams.json.

   ```
   aws amscm get-change-type-version --change-type-id "ct-3gf8dolbo8x9p" --query
    "ChangeTypeVersion.ExecutionInputSchema" --output text > CreateDmsTeParams.json
   ```

2. 실행 파라미터 JSON 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

   ```
   {
   "Description":          "TestTE",
   "VpcId":                "VPC_ID",
   "StackTemplateId":      "stm-knghtmmgefafdq89u",
   "Name":                 "TE-NAME",
   "TimeoutInMinutes":     60,
   "Parameters":   {
   ```

```
    "EngineName":       "mysql",
    "ServerName":       "sql.db.example.com",
    "Port":             "3306",
    "Username":         "DB-USER",
    "Password":         "DB-PW",}
    }
}
```

3. JSON 템플릿을 현재 폴더의 파일로 출력합니다.이 예제에서는 CreateDmsTeRfc.json:이라는 이름을 지정합니다.

```
aws amscm create-rfc --generate-cli-skeleton > CreateDmsTeRfc.json
```

4. CreateDmsTeRfc.json 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-3gf8dolbo8x9p",
"Title":                "DB-DMS-Target-Endpoint"
}
```

5. 실행 파라미터 파일과 CreateDmsTeRfc 파일을 지정하여 RFC를 생성합니다.

```
aws amscm create-rfc --cli-input-json file://CreateDmsTeRfc.json --execution-
parameters file://CreateDmsTeParams.json
```

응답에서 새 RFC의 ID를 수신하고 이를 사용하여 RFC를 제출하고 모니터링할 수 있습니다. 제출하기 전까지는 RFC가 편집 상태로 유지되고 시작되지 않습니다.

팁

- 이 변경 유형은 이제 버전 2.0입니다.
- AMS DMS는 S3 또는 RDS(관계형 데이터베이스 서비스)를 MySQL, MariaDB, Oracle, Postgresql 또는 Microsoft SQL과 함께 대상 엔드포인트로 사용할 수 있습니다. S3 대상 엔드포인트는 섹션을 참조하세요S3용 DMS 대상 엔드포인트: 생성.
- 자세한 내용은 데이터 마이그레이션 대상을 참조하세요.
- 최대 50개의 태그를 추가할 수 있지만 추가하려면 추가 구성 보기를 활성화해야 합니다.

S3용 DMS 대상 엔드포인트: 생성

콘솔을 사용하여 DMS S3 대상 엔드포인트 생성

AMS 콘솔에서이 변경 유형의 스크린샷:

작동 방식:

1. RFC 생성 페이지로 이동합니다. AMS 콘솔의 왼쪽 탐색 창에서 RFCs 클릭하여 RFCs 목록 페이지를 연 다음 RFC 생성을 클릭합니다.
2. 기본 변경 유형 찾아보기 보기에서 인기 있는 변경 유형(CT)을 선택하거나 범주별 선택 보기에서 CT를 선택합니다.
   - 변경 유형별 찾아보기: 빠른 생성 영역에서 인기 있는 CT를 클릭하여 RFC 실행 페이지를 즉시 열 수 있습니다. 빠른 생성으로 이전 CT 버전을 선택할 수 없습니다.

     CTs 정렬하려면 카드 또는 테이블 보기에서 모든 변경 유형 영역을 사용합니다. 어느 보기에서든 CT를 선택한 다음 RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다. 해당하는 경우 RFC 생성 버튼 옆에 이전 버전으로 생성 옵션이 나타납니다.
   - 범주별 선택: 범주, 하위 범주, 항목 및 작업을 선택하면 해당하는 경우 이전 버전으로 생성 옵션이 있는 CT 세부 정보 상자가 열립니다. RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다.
3. RFC 실행 페이지에서 CT 이름 영역을 열어 CT 세부 정보 상자를 확인합니다. 제목은 필수입니다 (변경 유형 찾아보기 보기에서 CT를 선택하면 입력됨). 추가 구성 영역을 열어 RFC에 대한 정보를 추가합니다.

   실행 구성 영역에서 사용 가능한 드롭다운 목록을 사용하거나 필요한 파라미터의 값을 입력합니다. 선택적 실행 파라미터를 구성하려면 추가 구성 영역을 엽니다.
4. 완료되면 실행을 클릭합니다. 오류가 없는 경우 성공적으로 생성된 RFC 페이지에 제출된 RFC 세부 정보와 초기 실행 출력이 표시됩니다.
5. 실행 파라미터 영역을 열어 제출한 구성을 확인합니다. 페이지를 새로 고쳐 RFC 실행 상태를 업데이트합니다. 선택적으로 RFC를 취소하거나 페이지 상단의 옵션을 사용하여 RFC 사본을 생성합니다.

CLI를 사용하여 DMS S3 대상 엔드포인트 생성

작동 방식:

1. 인라인 생성(모든 RFC 및 실행 파라미터가 포함된 `create-rfc` 명령을 실행) 또는 템플릿 생성(2개의 JSON 파일을 생성, 하나는 RFC 파라미터용이고 다른 하나는 실행 파라미터용)을 사용하고 두 파일을 입력으로 사용하여 `create-rfc` 명령을 실행합니다. 두 방법 모두 여기에 설명되어 있습니다.
2. 반환된 RFC ID로 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 명령을 제출합니다.

   RFC: `aws amscm get-rfc --rfc-id` *`ID`* 명령을 모니터링합니다.

변경 유형 버전을 확인하려면 다음 명령을 사용합니다.

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> `CreateRfc` 변경 유형에 대한 스키마의 일부인지 여부에 관계없이 모든 파라미터를 RFC와 함께 사용할 수 있습니다. 예를 들어 RFC 상태가 변경될 때 알림을 받으려면 요청의 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` RFC 파라미터 부분(실행 파라미터 아님)에이 줄을 추가합니다. 모든 CreateRfc 파라미터 목록은 AMS Change Management API 참조를 참조하세요.

인라인 생성:

인라인으로 제공된 실행 파라미터(실행 파라미터를 인라인으로 제공할 때 따옴표 이스케이프)로 RFC 생성 명령을 실행한 다음 반환된 RFC ID를 제출합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
aws --profile saml --region us-east-1 amscm create-rfc --change-type-id
 "ct-05muqzievnxk5" --change-type-version "1.0" --title "TestDMSTargetEndpointS3"
 --execution-parameters "{\"Description\":\"TestS3TE\",\"VpcId\":\"VPC-ID\",\"Name
\":\"S3TE-NAME\",\"StackTemplateId\":\"stm-knghtmmgefafdq89u\",\"TimeoutInMinutes
\":60,\"Parameters\":{\"EngineName\":\"s3\",\"S3BucketName\":\"amzn-s3-demo-bucket\",
\"S3ServiceAccessRoleArn\":\"arn:aws:iam::123456789123:role/my-s3-role\"}}"
```

템플릿 생성:

1. 이 변경 유형의 실행 파라미터를 JSON 파일로 출력합니다.이 예제에서는 CreateDmsTeS3Params.json:이라는 이름을 지정합니다.

```
aws amscm get-change-type-version --change-type-id "ct-05muqzievnxk5" --query
 "ChangeTypeVersion.ExecutionInputSchema" --output text > CreateDmsTeS3Params.json
```

2. 실행 파라미터 CreateDmsTeS3Params.json 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
{
"Description":          "TestS3DMS-TE",
"VpcId":                "VPC_ID",
"StackTemplateId":      "stm-knghtmmgefafdq89u",
"Name":                 "DMS-S3-TE",
"TimeoutInMinutes":     60,
"Parameters":   {
    "EngineName":       "s3",
    "S3BucketName":      "amzn-s3-demo-bucket",
    "S3ServiceAccessRoleArn":       "arn:aws:iam::123456789101:role/ams-ops-ct-
authors-dms-s3-test-role"
     }
}
```

3. JSON 템플릿을 현재 폴더의 파일로 출력합니다.이 예제에서는 CreateDmsTeS3Rfc.json:이라는 이름을 지정합니다.

```
aws amscm create-rfc --generate-cli-skeleton > CreateDmsTeS3Rfc.json
```

4. CreateDmsTeS3Rfc.json 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-05muqzievnxk5",
"Title":                "DMS_Target_S3"
}
```

5. 실행 파라미터 파일과 CreateDmsTeS3Rfc 파일을 지정하여 RFC를 생성합니다.

```
aws amscm create-rfc --cli-input-json file://CreateDmsTeS3Rfc.json --execution-
parameters file://CreateDmsTeS3Params.json
```

응답에서 새 RFC의 ID를 수신하고 이를 사용하여 RFC를 제출하고 모니터링할 수 있습니다. 제출하기 전까지는 RFC가 편집 상태로 유지되고 시작되지 않습니다.

팁

> **Note**
>
> 최대 50개의 태그를 추가할 수 있지만 추가하려면 추가 구성 보기를 활성화해야 합니다.

AMS는 S3용 대상 엔드포인트를 생성하기 위한 별도의 변경 유형을 제공합니다. 자세한 내용은 AWS Database Migration Service의 대상으로 Amazon S3 사용 및 AWS AWS Database Migration Service DMS의 대상으로 Amazon S3 사용 시 추가 연결 속성을 참조하세요.

5: AWS DMS 복제 작업: 생성

AMS 콘솔 또는 API/CLI를 사용하여 AMS AWS DMS 복제 작업을 생성할 수 있습니다.

AWS DMS 복제 작업 생성

콘솔을 사용하여 AWS DMS 복제 작업 생성

AMS 콘솔에서이 변경 유형의 스크린샷:

작동 방식:

1. RFC 생성 페이지로 이동합니다. AMS 콘솔의 왼쪽 탐색 창에서 RFCs 클릭하여 RFCs 목록 페이지를 연 다음 RFC 생성을 클릭합니다.
2. 기본 변경 유형 찾아보기 보기에서 인기 있는 변경 유형(CT)을 선택하거나 범주별 선택 보기에서 CT를 선택합니다.
   - 변경 유형별 찾아보기: 빠른 생성 영역에서 인기 있는 CT를 클릭하여 RFC 실행 페이지를 즉시 열 수 있습니다. 빠른 생성으로 이전 CT 버전을 선택할 수 없습니다.

     CTs 정렬하려면 카드 또는 테이블 보기에서 모든 변경 유형 영역을 사용합니다. 어느 보기에서든 CT를 선택한 다음 RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다. 해당하는 경우 RFC 생성 버튼 옆에 이전 버전으로 생성 옵션이 나타납니다.
   - 범주별 선택: 범주, 하위 범주, 항목 및 작업을 선택하면 해당하는 경우 이전 버전으로 생성 옵션이 있는 CT 세부 정보 상자가 열립니다. RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다.

3. RFC 실행 페이지에서 CT 이름 영역을 열어 CT 세부 정보 상자를 확인합니다. 제목은 필수입니다(변경 유형 찾아보기 보기에서 CT를 선택하면 입력됨). 추가 구성 영역을 열어 RFC에 대한 정보를 추가합니다.

   실행 구성 영역에서 사용 가능한 드롭다운 목록을 사용하거나 필요한 파라미터의 값을 입력합니다. 선택적 실행 파라미터를 구성하려면 추가 구성 영역을 엽니다.
4. 완료되면 실행을 클릭합니다. 오류가 없으면 제출된 RFC 세부 정보와 초기 실행 출력과 함께 RFC가 성공적으로 생성된 페이지가 표시됩니다.
5. 실행 파라미터 영역을 열어 제출한 구성을 확인합니다. 페이지를 새로 고쳐 RFC 실행 상태를 업데이트합니다. 선택적으로 RFC를 취소하거나 페이지 상단의 옵션을 사용하여 RFC 사본을 생성합니다.

CLI를 사용하여 AWS DMS 복제 작업 생성

작동 방식:

1. 인라인 생성(모든 RFC 및 실행 파라미터가 포함된 `create-rfc` 명령을 실행) 또는 템플릿 생성(2개의 JSON 파일을 생성, 하나는 RFC 파라미터용이고 다른 하나는 실행 파라미터용)을 사용하고 두 파일을 입력으로 사용하여 `create-rfc` 명령을 실행합니다. 두 방법 모두 여기에 설명되어 있습니다.
2. 반환된 RFC ID로 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 명령을 제출합니다.

   RFC: `aws amscm get-rfc --rfc-id` *`ID`* 명령을 모니터링합니다.

변경 유형 버전을 확인하려면 다음 명령을 사용합니다.

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 변경 유형에 대한 스키마의 일부인지 여부에 관계없이 모든 RFC에서 `CreateRfc` 파라미터를 사용할 수 있습니다. 예를 들어 RFC 상태가 변경될 때 알림을 받으려면 요청의 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` RFC 파라미터 부분(실행 파라미터 아님)에이 줄을 추가합니다. 모든 CreateRfc 파라미터 목록은 AMS Change Management API 참조를 참조하세요.

인라인 생성:

인라인으로 제공된 실행 파라미터(실행 파라미터를 인라인으로 제공할 때 따옴표 이스케이프)로 RFC 생성 명령을 실행한 다음 반환된 RFC ID를 제출합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
aws --profile saml --region us-east-1 amscm create-rfc --change-type-id
 "ct-1d2fml15b9eth" --change-type-version "1.0" --title "TestDMSRepTask" --
execution-parameters "{\"Description\":\"TestRepTask\",\"VpcId\":\"VPC-ID\",\"Name
\":\"DMSRepTask\",\"Parameters\":{\"CdcStartTime\":\1533776569\"MigrationType\":
\"full-load\",\"ReplicationInstanceArn\":\"REP_INSTANCE_ARN\",\"SourceEndpointArn
\":\"SOURCE_ENDPOINT_ARN\",\"TableMappings\":\"{\\\"rules\\\": [{\\\"rule-type
\\\": \\\"selection\\\",\\\"rule-id\\\": \\\"1\\\",\\\"rule-name\\\": \\\"1\\\
\",\\\"object-locator\\\": {\\\"schema-name\\\": \\\"Test\\\",\\\"table-name\\\
\": \\\"%\\\"}, \\\"rule-action\\\": \\\"include\\\"}] }\",\"TargetEndpointArn
\":\"TARGET_ENDPOINT_ARN\"},\"StackTemplateId\":\"stm-eos7uq0usnmeggdet\",
\"TimeoutInMinutes\":60}"
```

템플릿 생성:

1. 이 변경 유형의 실행 파라미터를 JSON 파일로 출력합니다.이 예제에서는 CreateDmsRtParams.json:이라는 이름을 지정합니다.

   ```
   aws amscm get-change-type-version --change-type-id "ct-1d2fml15b9eth" --query
    "ChangeTypeVersion.ExecutionInputSchema" --output text > CreateDmsRtParams.json
   ```

2. 실행 파라미터 JSON 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

   ```
   {
   "Description":          "DMSTestRepTask",
   "VpcId":                "VPC_ID",
   "StackTemplateId":      "stm-eos7uq0usnmeggdet",
   "Name":                 "Test DMS RT",
   "TimeoutInMinutes":     60,
   "Parameters":   {
       "CdcStartTime":           "1533776569",
       "MigrationType":          "full-load",
       "ReplicationInstanceArn": "REP_INSTANCE_ARN",
       "SourceEndpointArn":      "SOURCE_ENDPOINT_ARN",
       "TargetEndpointArn":      "TARGET_ENDPOINT_ARN"
   ```

```
    "TableMappings":          {"rules": [{"rule-type": "selection","rule-id":
 "1","rule-name": "1","object-locator": {"schema-name": "Test","table-name": "%"},
 "rule-action": "include"}] }",
    }
}
```

3. JSON 템플릿을 현재 폴더의 파일로 출력합니다.이 예제에서는 CreateDmsRtRfc.json:이라는 이름을 지정합니다.

```
aws amscm create-rfc --generate-cli-skeleton > CreateDmsRtRfc.json
```

4. CreateDmsRtRfc.json 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-1d2fml15b9eth",
"Title":                "DMS-RI-Create-RFC"
}
```

5. 실행 파라미터 파일과 CreateDmsRtRfc 파일을 지정하여 RFC를 생성합니다.

```
aws amscm create-rfc --cli-input-json file://CreateDmsRtRfc.json --execution-
parameters file://CreateDmsRtParams.json
```

응답에서 새 RFC의 ID를 수신하고 이를 사용하여 RFC를 제출하고 모니터링할 수 있습니다. 제출하기 전까지는 RFC가 편집 상태로 유지되고 시작되지 않습니다.

팁

세 가지 유형의 변경 사항 또는 데이터를 캡처하는 AWS DMS 작업을 생성할 수 있습니다. 자세한 내용은 AWS DMS 작업 작업, 작업 생성 및 AWS DMS를 사용하여 지속적 복제를 위한 작업 생성을 참조하세요.

AWS DMS 관리

AWS DMS 관리 예제.

AWS DMS 복제 작업 시작

콘솔을 사용하여 AWS DMS 복제 작업 시작

AMS 콘솔에서이 변경 유형의 스크린샷:

작동 방식:

1. RFC 생성 페이지로 이동합니다. AMS 콘솔의 왼쪽 탐색 창에서 RFCs 클릭하여 RFCs 목록 페이지를 연 다음 RFC 생성을 클릭합니다.
2. 기본 변경 유형 찾아보기 보기에서 널리 사용되는 변경 유형(CT)을 선택하거나 범주별 선택 보기에서 CT를 선택합니다.
   - 변경 유형별 찾아보기: 빠른 생성 영역에서 인기 있는 CT를 클릭하여 RFC 실행 페이지를 즉시 열 수 있습니다. 빠른 생성으로 이전 CT 버전을 선택할 수 없습니다.

     CTs 정렬하려면 카드 또는 테이블 보기에서 모든 변경 유형 영역을 사용합니다. 어느 보기에서든 CT를 선택한 다음 RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다. 해당하는 경우 RFC 생성 버튼 옆에 이전 버전으로 생성 옵션이 나타납니다.
   - 범주별 선택: 범주, 하위 범주, 항목 및 작업을 선택하면 해당하는 경우 이전 버전으로 생성 옵션이 있는 CT 세부 정보 상자가 열립니다. RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다.
3. RFC 실행 페이지에서 CT 이름 영역을 열어 CT 세부 정보 상자를 확인합니다. 제목은 필수입니다 (변경 유형 찾아보기 보기에서 CT를 선택하면 입력됨). 추가 구성 영역을 열어 RFC에 대한 정보를 추가합니다.

   실행 구성 영역에서 사용 가능한 드롭다운 목록을 사용하거나 필요한 파라미터의 값을 입력합니다. 선택적 실행 파라미터를 구성하려면 추가 구성 영역을 엽니다.
4. 완료되면 실행을 클릭합니다. 오류가 없는 경우 RFC가 성공적으로 생성된 페이지에 제출된 RFC 세부 정보와 초기 실행 출력이 표시됩니다.
5. 실행 파라미터 영역을 열어 제출한 구성을 확인합니다. 페이지를 새로 고쳐 RFC 실행 상태를 업데이트합니다. 선택적으로 RFC를 취소하거나 페이지 상단의 옵션을 사용하여 RFC 사본을 생성합니다.

CLI를 사용하여 AWS DMS 복제 작업 시작

작동 방식:

1. 인라인 생성(모든 RFC 및 실행 파라미터가 포함된 `create-rfc` 명령을 실행) 또는 템플릿 생성(2개의 JSON 파일을 생성, 하나는 RFC 파라미터용이고 다른 하나는 실행 파라미터용)을 사용하고 두 파일을 입력으로 사용하여 `create-rfc` 명령을 실행합니다. 두 방법 모두 여기에 설명되어 있습니다.
2. 반환된 RFC ID로 RFC: `aws amscm submit-rfc --rfc-id ID` 명령을 제출합니다.

   RFC: `aws amscm get-rfc --rfc-id ID` 명령을 모니터링합니다.

변경 유형 버전을 확인하려면 다음 명령을 사용합니다.

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 변경 유형에 대한 스키마의 일부인지 여부에 관계없이 모든 `CreateRfc` 파라미터를 RFC와 함께 사용할 수 있습니다. 예를 들어 RFC 상태가 변경될 때 알림을 받으려면 요청의 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` RFC 파라미터 부분(실행 파라미터 아님)에이 줄을 추가합니다. 모든 CreateRfc 파라미터 목록은 AMS Change Management API 참조를 참조하세요.

인라인 생성:

인라인으로 제공된 실행 파라미터(실행 파라미터를 인라인으로 제공할 때 따옴표 이스케이프)로 RFC 생성 명령을 실행한 다음 반환된 RFC ID를 제출합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
aws amscm create-rfc --change-type-id "ct-1yq7hhqse71yg" --change-type-version
 "1.0" --title "Start DMS Replication Task" --execution-parameters "{\"DocumentName
\":\"AWSManagedServices-StartDmsTask\",\"Region\":\"us-east-1\",\"Parameters\":
{\"ReplicationTaskArn\":[\"TASK_ARN\"],\"StartReplicationTaskType\":[\"start-
replication\"],\"CdcStartPosition\":[\"\"],\"CdcStopPosition\":[\"\"]}}"
```

템플릿 생성:

1. 이 변경 유형의 실행 파라미터를 JSON 파일로 출력합니다.이 예제에서는 StartDmsRtParams.json:이라는 이름을 지정합니다.

```
aws amscm get-change-type-version --change-type-id "ct-1yq7hhqse71yg" --query
 "ChangeTypeVersion.ExecutionInputSchema" --output text > StartDmsRtParams.json
```

2. 실행 파라미터 JSON 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
{
  "DocumentName": "AWSManagedServices-StartDmsTask",
  "Region": "us-east-1",
  "Parameters": {
      "ReplicationTaskArn": [
        "TASK_ARN"
      ],
      "StartReplicationTaskType": [
        "start-replication"
      ],
      "CdcStartPosition": [
        ""
      ],
      "CdcStopPosition": [
        ""
      ]
  }
}
```

3. JSON 템플릿을 현재 폴더의 파일로 출력합니다.이 예제에서는 StartDmsRtRfc.json:이라는 이름을 지정합니다.

```
aws amscm create-rfc --generate-cli-skeleton > StartDmsRtRfc.json
```

4. StartDmsRtRfc.json 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
{
  "ChangeTypeId": "ct-1yq7hhqse71yg",
  "ChangeTypeVersion": "1.0",
  "Title": "Start DMS Replication Task"
}
```

5. 실행 파라미터 파일과 StartDmsRtRfc 파일을 지정하여 RFC를 생성합니다.

```
aws amscm create-rfc --cli-input-json file://StartDmsRtRfc.json --execution-
parameters file://StartDmsRtParams.json
```

응답에서 새 RFC의 ID를 수신하고 이를 사용하여 RFC를 제출하고 모니터링할 수 있습니다. 제출하기 전까지는 RFC가 편집 상태로 유지되고 시작되지 않습니다.

팁

AMS 콘솔 또는 AMS API/CLI를 사용하여 AWS DMS 복제 작업을 시작할 수 있습니다. 자세한 내용은 AWS DMS 작업 작업을 참조하세요.

AWS DMS 복제 작업 중지

콘솔을 사용하여 AWS DMS 복제 작업 중지

AMS 콘솔에서이 변경 유형의 스크린샷:

작동 방식:

1. RFC 생성 페이지로 이동합니다. AMS 콘솔의 왼쪽 탐색 창에서 RFCs 클릭하여 RFCs 목록 페이지를 연 다음 RFC 생성을 클릭합니다.
2. 기본 변경 유형 찾아보기 보기에서 인기 있는 변경 유형(CT)을 선택하거나 범주별 선택 보기에서 CT를 선택합니다.
   - 변경 유형별 찾아보기: 빠른 생성 영역에서 인기 있는 CT를 클릭하여 RFC 실행 페이지를 즉시 열 수 있습니다. 빠른 생성으로 이전 CT 버전을 선택할 수 없습니다.

     CTs 정렬하려면 카드 또는 테이블 보기에서 모든 변경 유형 영역을 사용합니다. 어느 보기에서든 CT를 선택한 다음 RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다. 해당하는 경우 RFC 생성 버튼 옆에 이전 버전으로 생성 옵션이 나타납니다.
   - 범주별 선택: 범주, 하위 범주, 항목 및 작업을 선택하면 해당하는 경우 이전 버전으로 생성 옵션이 있는 CT 세부 정보 상자가 열립니다. RFC 생성을 클릭하여 RFC 실행 페이지를 엽니다.
3. RFC 실행 페이지에서 CT 이름 영역을 열어 CT 세부 정보 상자를 확인합니다. 제목은 필수입니다(변경 유형 찾아보기 보기에서 CT를 선택하면 입력됨). 추가 구성 영역을 열어 RFC에 대한 정보를 추가합니다.

   실행 구성 영역에서 사용 가능한 드롭다운 목록을 사용하거나 필요한 파라미터의 값을 입력합니다. 선택적 실행 파라미터를 구성하려면 추가 구성 영역을 엽니다.

4. 완료되면 실행을 클릭합니다. 오류가 없는 경우 RFC가 성공적으로 생성된 페이지에 제출된 RFC 세부 정보와 초기 실행 출력이 표시됩니다.
5. 실행 파라미터 영역을 열어 제출한 구성을 확인합니다. 페이지를 새로 고쳐 RFC 실행 상태를 업데이트합니다. 선택적으로 RFC를 취소하거나 페이지 상단의 옵션을 사용하여 RFC 사본을 생성합니다.

CLI를 사용하여 AWS DMS 복제 작업 중지

작동 방식:

1. 인라인 생성(모든 RFC 및 실행 파라미터가 포함된 `create-rfc` 명령을 실행) 또는 템플릿 생성(2개의 JSON 파일을 생성, 하나는 RFC 파라미터용이고 다른 하나는 실행 파라미터용)을 사용하고 두 파일을 입력으로 사용하여 `create-rfc` 명령을 실행합니다. 두 방법 모두 여기에 설명되어 있습니다.
2. 반환된 RFC ID로 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 명령을 제출합니다.

   RFC: `aws amscm get-rfc --rfc-id` *`ID`* 명령을 모니터링합니다.

변경 유형 버전을 확인하려면 다음 명령을 사용합니다.

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> 변경 유형에 대한 스키마의 일부인지 여부에 관계없이 모든 `CreateRfc` 파라미터를 RFC와 함께 사용할 수 있습니다. 예를 들어 RFC 상태가 변경될 때 알림을 받으려면 요청의 `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` RFC 파라미터 부분(실행 파라미터 아님)에이 줄을 추가합니다. 모든 CreateRfc 파라미터 목록은 AMS Change Management API 참조를 참조하세요.

인라인 생성:

인라인으로 제공된 실행 파라미터(실행 파라미터를 인라인으로 제공할 때 따옴표 이스케이프)로 RFC 생성 명령을 실행한 다음 반환된 RFC ID를 제출합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

```
aws amscm create-rfc --change-type-id "ct-1vd3y4ygbqmfk" --change-type-version
 "1.0" --title "Stop DMS Replication Task" --execution-parameters "{\"DocumentName
\":\"AWSManagedServices-StopDmsTask\",\"Region\":\"us-east-1\",\"Parameters\":
{\"ReplicationTaskArn\":[\"TASK_ARN\"]}}"
```

템플릿 생성:

1. 이 변경 유형의 실행 파라미터를 JSON 파일로 출력합니다.이 예제에서는 StopDmsRtParams.json:이라는 이름을 지정합니다.

   ```
   aws amscm get-change-type-version --change-type-id "ct-1vd3y4ygbqmfk" --query
    "ChangeTypeVersion.ExecutionInputSchema" --output text > StopDmsRtParams.json
   ```

2. 실행 파라미터 JSON 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

   ```
   {
     "DocumentName": "AWSManagedServices-StopDmsTask",
     "Region": "us-east-1",
     "Parameters": {
         "ReplicationTaskArn": [
           "TASK_ARN"
         ]
     }
   }
   ```

3. JSON 템플릿을 현재 폴더의 파일로 출력합니다.이 예제에서는 StopDmsRtRfc.json:이라는 이름을 지정합니다.

   ```
   aws amscm create-rfc --generate-cli-skeleton > StopDmsRtRfc.json
   ```

4. StopDmsRtRfc.json 파일을 수정하고 저장합니다. 예를 들어 콘텐츠를 다음과 같은 내용으로 바꿀 수 있습니다.

   ```
   {
     "ChangeTypeId": "ct-1vd3y4ygbqmfk",
     "ChangeTypeVersion": "1.0",
     "Title": "Stop DMS Replication Task"
   }
   ```

5. 실행 파라미터 파일과 StopDmsRtRfc 파일을 지정하여 RFC를 생성합니다.

```
aws amscm create-rfc --cli-input-json file://StopDmsRtRfc.json --execution-
parameters file://StopDmsRtParams.json
```

응답에서 새 RFC의 ID를 수신하고 이를 사용하여 RFC를 제출하고 모니터링할 수 있습니다. 제출하기 전까지는 RFC가 편집 상태로 유지되고 시작되지 않습니다.

팁

AMS 콘솔 또는 AMS API/CLI를 사용하여 DMS 복제 작업을 중지할 수 있습니다. 자세한 내용은 AWS DMS 작업 작업을 참조하세요.

AMS RDS for Microsoft SQL Server로 데이터베이스(DB) 가져오기

> **Note**
>
> AMS API/CLI(amscm 및 amsskms) 엔드포인트는 AWS 버지니아 북부 리전에 있습니다`us-east-1`. 인증 설정 방식과 계정 및 리소스가 있는 AWS 리전에 따라 명령을 실행할 `--region us-east-1` 때를 추가해야 할 수 있습니다. 인증 방법인 `--profile saml`경우를 추가해야 할 수도 있습니다.

DB를 AMS RDS for SQL Server로 가져오고, 프로세스는 변경 요청(RFCs)으로 제출된 AMS 변경 유형(CTs)에 의존하며, Amazon RDS API 파라미터를 입력으로 사용합니다. MicroSoft SQL Server는 관계형 데이터베이스 관리 시스템(RDBMS)입니다. 자세한 내용은 Amazon Relational Database Service(RDS) 및 rds 또는 Amazon RDS API 참조도 참조하세요.

> **Note**
>
> 다음 단계로 넘어가기 전에 각 RFC가 성공적으로 완료되었는지 확인합니다.

상위 수준 가져오기 단계:

1. 소스 온프레미스 MS SQL 데이터베이스를 .bak(백업) 파일로 백업
2. .bak 파일을 전송(암호화) Amazon Simple Storage Service(S3) 버킷에 복사합니다.
3. 대상 Amazon RDS MS SQL 인스턴스의 새 DB로 .bak 가져오기

요구 사항:

- AMS의 MS SQL RDS 스택
- 복원 옵션이 있는 RDS 스택(SQLSERVER_BACKUP_RESTORE)
- 전송 S3 버킷
- Amazon RDS가 역할을 수임할 수 있도록 버킷 액세스 권한이 있는 IAM 역할
- RDS를 관리하기 위해 MS SQL Management Studio가 설치된 EC2 인스턴스(온프레미스 워크스테이션일 수 있음)

설정

다음 작업을 완료하여 가져오기 프로세스를 시작합니다.

1. RFC를 제출하여 배포 | 고급 스택 구성 요소 | RDS 데이터베이스 스택 | 생성(ct-2z60dyvto9g6c)을 사용하여 RDS 스택을 생성합니다. 생성 요청에 대상 DB 이름(파라미터)을 사용하지 마십시오. 가져오기 중에 대상 DB가 생성됩니다.RDSDBName 충분한 공간(RDSAllocatedStorage 파라미터)을 확보해야 합니다. 이에 대한 자세한 내용은 AMS 변경 관리 가이드 RDS DB 스택 | 생성을 참조하세요.
2. RFC를 제출하여 배포 | 고급 스택 구성 요소 | S3 스토리지 | 생성(ct-1a68ck03fn98r)을 사용하여 전송 S3 버킷(이미 없는 경우)을 생성합니다. 이에 대한 자세한 내용은 AMS 변경 관리 가이드 S3 스토리지 | 생성을 참조하세요.
3. 관리 제출 | 기타 | 기타 | 업데이트(ct-1e1xtak34nx76) RFC를 제출하여 다음 세부 정보로 customer_rds_s3_role를 구현합니다.

   콘솔에서:

   - 제목: "MS SQL Server 데이터베이스 가져오기를 지원하려면이 계정customer_rds_s3_role에서를 구현합니다.
   - Transit S3 버킷 이름: *BUCKET_NAME*.
   - 연락처 정보: *EMAIL*.

   CLI용 ImportDbParams.json 파일 사용:

```
{
       "Comment": "{"Transit S3 bucket name":"BUCKET_NAME"}",
       "Priority": "High"
     }
```

```
}
```

4. 관리 제출 | 기타 | 기타 | AMS를 요청하여 SQLSERVER_BACKUP_RESTORE 옵션을 1단계에서 생성된 RDS로 설정하도록 RFC를 업데이트합니다(이 요청에서는 1단계 출력의 스택 ID와이 요청의 `customer_rds_s3_role` IAM 역할 사용).
5. RFC를 제출하여 EC2 인스턴스를 생성하고(기존 EC2 또는 온프레미스 워크스테이션/인스턴스 사용 가능) 인스턴스에 Microsoft SQL Management Studio를 설치합니다.

데이터베이스 가져오기

데이터베이스(DB)를 가져오려면 다음 단계를 따릅니다.

1. MS SQL 네이티브 백업 및 복원을 사용하여 소스 온프레미스 데이터베이스를 백업합니다(SQL Server에서 네이티브 백업 및 복원 지원 참조). 해당 작업을 실행한 결과 .bak(백업) 파일이 있어야 합니다.
2. AWS S3 CLI 또는 AWS S3 S3 콘솔을 사용하여 및 기존 전송 S3 버킷에 .bak 파일을 업로드합니다. 전송 S3 버킷에 대한 자세한 내용은 암호화를 사용하여 데이터 보호를 참조하세요.
3. .bak 파일을 대상 RDS for SQL Server MS SQL 인스턴스의 새 DB로 가져옵니다(유형에 대한 자세한 내용은 Amazon RDS for MySQL 인스턴스 유형 참조).

   a. EC2 인스턴스(온프레미스 워크스테이션)에 로그인하고 MS SQL Management Studio를 엽니다.

   b. 1단계에서 사전 조건으로 생성된 대상 RDS 인스턴스에 연결합니다. 다음 절차에 따라 연결합니다. Microsoft SQL Server 데이터베이스 엔진을 실행하는 DB 인스턴스에 연결

   c. 새 구조화 쿼리 언어(SQL) 쿼리로 가져오기(복원) 작업을 시작합니다(SQL 쿼리에 대한 자세한 내용은 SQL 소개 참조). 대상 데이터베이스 이름은 새 이름이어야 합니다(이전에 생성한 데이터베이스와 동일한 이름을 사용하지 마십시오). 암호화가 없는 예제:

   ```
   exec msdb.dbo.rds_restore_database
           @restore_db_name=TARGET_DB_NAME,

           @s3_arn_to_restore_from='arn:aws:s3:::BUCKET_NAME/FILENAME.bak';
   ```

   d. 이 쿼리를 별도의 창에서 실행하여 가져오기 작업의 상태를 정기적으로 확인합니다.

   ```
   exec msdb.dbo.rds_task_status;
   ```

상태가 실패로 변경되면 메시지에서 실패 세부 정보를 찾습니다.

정리

데이터베이스를 가져온 후에는 불필요한 리소스를 제거하고 싶을 수 있습니다. 다음 단계를 따르세요.

1. S3 버킷에서 백업 파일(.bak)을 삭제합니다. S3 콘솔을 사용하여이 작업을 수행할 수 있습니다. S3 버킷에서 객체를 삭제하는 CLI 명령은 AWS CLI 명령 참조의 rm을 참조하세요.
2. S3 버킷을 사용할 계획이 없는 경우 삭제합니다. 이 작업을 수행하는 단계는 스택 삭제를 참조하세요.
3. MS SQL 가져오기를 계획하고 있지 않은 경우 관리 | 기타 | 기타 | 업데이트(ct-0xdawir96cy7k) RFC를 제출하고 AMS에 IAM 역할 삭제를 요청합니다`customer_rds_s3_role`.

AMS의 티어 및 타이 앱 배포

티어 및 타이 배포는 별도의 RFCs를 사용하여 스택의 리소스를 독립적으로 생성, 구성 및 배포하고 진행하면서 스택 구성 요소의 IDs를 사용하여 서로 연결하는 곳입니다.

예를 들어 티어 및 타이 접근 방식을 사용하여 로드 밸런서와 데이터베이스 뒤에 고가용성(중복) 웹 사이트를 배포하려면 데이터베이스, 로드 밸런서 및 두 개의 EC2 인스턴스 또는 Auto Scaling 그룹에 대한 RFCs를 제출하고 생성한 ELB의 ID로 EC2 인스턴스 또는 Auto Scaling 그룹을 구성합니다.

리소스가 배포된 후 보안 그룹 생성 변경 사항을 제출하여 리소스가 데이터베이스와 통신할 수 있도록 할 수 있습니다. 보안 그룹 생성에 대한 자세한 내용은 보안 그룹 생성을 참조하세요.

AMS의 전체 스택 앱 배포

Full Stack 배포는 필요한 모든 것을 한 번에 생성하고 구성하는 CT와 함께 RFC를 제출하는 곳입니다. 예를 들어, 방금 설명한 고가용성 웹 사이트(EC2 인스턴스, 로드 밸런서 및 데이터베이스)를 배포하려면 Auto Scaling 그룹, 로드 밸런서, 데이터베이스 및 모든 인스턴스가 스택으로 작동하는 데 필요한 보안 그룹 설정을 생성하고 구성한 CT를 사용합니다. 이를 수행하는 두 AMS CTs의 예는 다음에 설명되어 있습니다.

- 고가용성 2계층 스택(ct-06mjngx5flwto):이 변경 유형을 사용하면 스택을 생성하고 Auto Scaling 그룹, RDS 지원 데이터베이스, Load Balancer, CodeDeploy 애플리케이션 및 구성을 구성할 수 있습니다. 로드 밸런서는 여러 애플리케이션에서 네트워크 어플라이언스로 공유되므로 계층으로 간주되지

않으며 CodeDeploy 함수도 어플라이언스로 간주됩니다. 또한 애플리케이션을 배포하는 데 사용할 수 있는 CodeDeploy 배포 그룹(CodeDeploy 애플리케이션에 지정한 이름 포함)을 생성합니다. 리소스가 함께 작동하도록 허용하는 보안 그룹 설정이 자동으로 생성됩니다.

- 고가용성 1계층 스택(ct-09t6q7j9v5hrn):이 변경 유형을 사용하면 스택을 생성하고 Auto Scaling 그룹 및 Application Load Balancer를 구성할 수 있습니다. 리소스가 함께 작동하도록 허용하는 보안 그룹 설정이 자동으로 생성됩니다.

프로비저닝 변경 유형(CTs) 작업

AMS는 관리형 인프라에 대한 책임이 있습니다. 변경하려면 올바른 CT 분류(범주, 하위 범주, 항목 및 작업)를 사용하여 RFC를 제출해야 합니다. 이 섹션에서는 CTs를 찾고, 필요에 맞는 CT가 있는지 확인하고, 없는 경우 새 CT를 요청하는 방법을 설명합니다.

기존 CT가 요구 사항을 충족하는지 확인

AMS로 배포할 항목을 결정한 후 다음 단계는 기존 CTs 및 CloudFormation 템플릿을 조사하여 솔루션이 이미 존재하는지 확인하는 것입니다.

RFC를 생성할 때 CT를 지정해야 합니다. AWS Management Console 또는 AMS API/CLI를 사용할 수 있습니다. 둘 다 사용하는 예는 다음에 설명되어 있습니다.

콘솔 또는 API/CLI를 사용하여 변경 유형 ID(CT) 또는 버전을 찾을 수 있습니다. 검색 또는 분류 선택이라는 두 가지 방법이 있습니다. 두 선택 유형 모두 가장 자주 사용, 가장 최근에 사용 또는 알파벳순을 선택하여 검색을 정렬할 수 있습니다.

YouTube 동영상: AWS Managed Services CLI를 사용하여 RFC를 생성하려면 어떻게 해야 하고 CT 스키마는 어디에서 찾을 수 있습니까?

AMS 콘솔의 RFCs

- 변경 유형별 찾아보기(기본값)를 선택한 상태에서 다음 중 하나를 수행합니다.
  - 빠른 생성 영역을 사용하여 AMS의 가장 인기 CTs 중에서 선택합니다. 레이블을 클릭하면 제목 옵션이 자동으로 채워진 RFC 실행 페이지가 열립니다. 필요에 따라 나머지 옵션을 완료하고 실행을 클릭하여 RFC를 제출합니다.
  - 또는 모든 변경 유형 영역까지 아래로 스크롤하여 옵션 상자에 CT 이름을 입력하기 시작합니다. 정확한 변경 유형 이름이나 전체 변경 유형 이름을 가질 필요는 없습니다. 관련 단어를 입력하여 변경 유형 ID, 분류 또는 실행 모드(자동 또는 수동)별로 CT를 검색할 수도 있습니다.

기본 카드 보기를 선택하면 입력 시 일치하는 CT 카드가 나타나고 카드를 선택한 다음 RFC 생성을 클릭합니다. 테이블 보기를 선택한 상태에서 관련 CT를 선택하고 RFC 생성을 클릭합니다. 두 방법 모두 RFC 실행 페이지를 엽니다.

- 또는 및 변경 유형 선택을 탐색하려면 페이지 상단의 범주별 선택을 클릭하여 일련의 드롭다운 옵션 상자를 엽니다.
- 범주, 하위 범주, 항목 및 작업을 선택합니다. 해당 변경 유형에 대한 정보 상자가 페이지 하단에 패널이 나타납니다.
- 준비가 되면 Enter 키를 누르면 일치하는 변경 유형 목록이 나타납니다.
- 목록에서 변경 유형을 선택합니다. 해당 변경 유형에 대한 정보 상자가 페이지 하단에 나타납니다.
- 올바른 변경 유형을 지정한 후 RFC 생성을 선택합니다.

> **Note**
>
> 이러한 명령이 작동하려면 AMS CLI가 설치되어 있어야 합니다. AMS API 또는 CLI를 설치하려면 AMS 콘솔 개발자 리소스 페이지로 이동합니다. AMS CM API 또는 AMS SKMS API에 대한 참조 자료는 사용 설명서의 AMS 정보 리소스 섹션을 참조하세요. 인증 `--profile` 옵션을 추가해야 할 수 있습니다. 예: `aws amsskms ams-cli-command --profile SAML`. 와 같이 모든 AMS 명령이 us-east-1에서 실행되므로 `--region` 옵션을 추가해야 할 수도 있습니다`aws amscm ams-cli-command --region=us-east-1`.

> **Note**
>
> AMS API/CLI(amscm 및 amsskms) 엔드포인트는 AWS 버지니아 북부 리전에 있습니다`us-east-1`. 인증 설정 방식과 계정 및 리소스가 있는 AWS 리전에 따라 명령을 실행할 `--region us-east-1` 때를 추가해야 할 수 있습니다. 인증 방법인 `--profile saml`경우를 추가해야 할 수도 있습니다.

AMS CM API(ListChangeTypeClassificationSummaries 참조) 또는 CLI를 사용하여 변경 유형을 검색하려면

필터 또는 쿼리를 사용하여 검색할 수 있습니다. ListChangeTypeClassificationSummaries 작업에는 `Category`, `SubcategoryItem`, 및에 대한 필터 옵션이 `Operation`있지만 값은 기존 값과 정확히 일치해야 합니다. CLI를 사용할 때 보다 유연한 결과를 얻으려면 `--query` 옵션을 사용할 수 있습니다.

AMS CM API/CLI를 사용한 유형 필터링 변경

속성	유효값	유효/기본 조건	참고
ChangeTypeId	ChangeTypeId를 나타내는 모든 문자열(예: ct-abc123xyz7890)	같음	변경 유형 IDs는 변경 유형 참조를 참조하세요. 변경 유형 IDs는 변경 유형 찾기 또는 CSIO를 참조하세요.
범주	모든 자유 형식 텍스트	포함	각 개별 필드의 정규식은 지원되지 않습니다. 대/소문자를 구분하지 않는 검색
Subcategory			
Item			
연산			

1. 다음은 변경 유형 분류를 나열하는 몇 가지 예입니다.

   다음 명령은 모든 변경 유형 범주를 나열합니다.

   ```
   aws amscm list-change-type-categories
   ```

   다음 명령은 지정된 범주에 속하는 하위 범주를 나열합니다.

   ```
   aws amscm list-change-type-subcategories --category CATEGORY
   ```

   다음 명령은 지정된 범주 및 하위 범주에 속하는 항목을 나열합니다.

   ```
   aws amscm list-change-type-items --category CATEGORY --subcategory SUBCATEGORY
   ```

2. 다음은 CLI 쿼리를 사용하여 변경 유형을 검색하는 몇 가지 예입니다.

   다음 명령은 항목 이름에 "S3"가 포함된 CT 분류 요약을 검색하고 범주, 하위 범주, 항목, 작업 및 변경 유형 ID의 출력을 테이블 형식으로 생성합니다.

```
aws amscm list-change-type-classification-summaries --query
 "ChangeTypeClassificationSummaries [?contains(Item, 'S3')].
[Category,Subcategory,Item,Operation,ChangeTypeId]" --output table
```

```
+-------------------------------------------------------------+
|              ListChangeTypeClassificationSummaries          |
+----------+-------------------------+--+------+---------------+
|Deployment|Advanced Stack Components|S3|Create|ct-1a68ck03fn98r|
+----------+-------------------------+--+------+---------------+
```

3. 그런 다음 변경 유형 ID를 사용하여 CT 스키마를 가져오고 파라미터를 검사할 수 있습니다. 다음 명령은 스키마를 CreateS3Params.schema.json.

```
aws amscm get-change-type-version --change-type-id "ct-1a68ck03fn98r"
 --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
 CreateS3Params.schema.json
```

CLI 쿼리 사용에 대한 자세한 내용은 --query 옵션을 사용하여 출력을 필터링하는 방법 및 쿼리 언어 참조인 JMESPath 사양을 참조하세요.

4. 변경 유형 ID가 있으면 변경 유형의 버전을 확인하여 최신 버전인지 확인하는 것이 좋습니다. 이 명령을 사용하여 지정된 변경 유형의 버전을 찾습니다.

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CHANGE_TYPE_ID
```

특정 변경 유형에 AutomationStatus 대한를 찾으려면 다음 명령을 실행합니다.

```
aws amscm --profile saml get-change-type-version --change-type-id CHANGE_TYPE_ID --
query "ChangeTypeVersion.{AutomationStatus:AutomationStatus.Name}"
```

특정 변경 유형에 ExpectedExecutionDurationInMinutes 대한를 찾으려면 다음 명령을 실행합니다.

```
aws amscm --profile saml get-change-type-version --change-type-id ct-14027q0sjyt1h
 --query "ChangeTypeVersion.{ExpectedDuration:ExpectedExecutionDurationInMinutes}"
```

적합하다고 생각되는 CT를 찾았으면 관련 실행 파라미터 JSON 스키마를 살펴보고 사용 사례를 해결하는지 알아봅니다.

이 명령을 사용하여 CT 스키마를 CT 이름의 JSON 파일로 출력합니다.이 예제에서는 S3 스토리지 스키마 생성을 출력합니다.

```
aws amscm get-change-type-version --change-type-id "ct-1a68ck03fn98r"
--query "ChangeTypeVersion.ExecutionInputSchema" --output text >
CreateBucketParams.json
```

이 스키마가 제공하는 내용을 자세히 살펴보겠습니다.

S3 버킷 스키마 생성

```
{
  "$schema": "http://json-schema.org/draft-04/sch
ema#",
"name": "Create S3 Storage
"description": "Use to create an Amazon Simple
 Storage Service stack.",
  "type": "object",
  "properties": {
    "Description": {
      "description": "The description of the
 stack.",
      "type": "string",
      "minLength": 1,
      "maxLength": 500
    },
    "VpcId": {
      "description": "ID of the VPC to create the S3
 Bucket in, in the form vpc-a1b2c3d4e5f67890e.",
      "type": "string",
      "pattern": "^vpc-[a-z0-9]{17}$"
    },
    "StackTemplateId": {
      "description": "Required value: stm-s2b72
beb000000000.",
      "type": "string",
      "enum": ["stm-s2b72beb000000000"]
    },
    "Name":{
```

스키마는 스키마의 용도를 알려주는 CT("설명")로 시작합니다. 이 경우 S3 스토리지 스택을 생성합니다.

다음으로 지정할 수 있는 필수 및 선택적 속성이 있습니다. 기본 속성 값이 지정됩니다. 필요한 속성은 스키마 끝에 나열됩니다.

StackTemplateId 영역에는이 CT 및 스키마에 대한 특정 스택 템플릿이 하나 있으며 해당 ID는 필수 속성 값입니다.

스키마를 사용하면 내부 북키핑 목적으로 생성 중인 스택에 태그를 지정할 수 있습니다. 또한 백업과 같은 일부 옵션에는 Key:backup 및 Value:true 태그가 필요합니다. 자세한 내용은 Amazon EC2 리소스 태그 지정을 참조하세요.

```
      "description": "The name of the stack to
create.",
      "type": "string",
      "minLength": 1,
      "maxLength": 255
    },
    "Tags": {
      "description": "Up to seven tags (key/value
pairs) for the stack.",
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "Key": {
            "type": "string",
            "minLength": 1,
            "maxLength": 127
          },
          "Value": {
            "type": "string",
            "minLength": 1,
            "maxLength": 255
          }
        },
        "additionalProperties": false,
        "required": [
          "Key",
          "Value"
        ]
      },
      "minItems": 1,
      "maxItems": 7
    },
    "TimeoutInMinutes": {
      "description": "The amount of time, in minutes,
to allow for creation of the stack.",
      "type": "number",
      "minimum": 0,
      "maximum": 60
    },
    "Parameters": {
      "description": "Specifications for the
stack.",
      "type": "object",
```

CT JSON 스키마의 파라미터 섹션에서는 실행 파라미터를 제공합니다.

```
      "properties": {
        "AccessControl": {
          "description": "The canned (predefined)
 access control list (ACL) to assign to the bucket.",
          "type": "string",
          "enum": [
            "Private",
            "PublicRead",
            "AuthenticatedRead",
            "BucketOwnerRead"
          ]
        },
        "BucketName": {
          "description": "A name for the bucket.
 The bucket name must contain only lowercase letters,
 numbers, periods (.), and hyphens (-).",
          "type": "string",
          "pattern": "^[a-z0-9]([-.a-z0-9]+)[a-z
0-9]$",
          "minLength": 3,
          "maxLength": 63
        }
      },
      "additionalProperties": false,
      "required": [
        "AccessControl",
        "BucketName"
      ]
    }
  },
  "additionalProperties": false,
  "required": [
    "Description",
    "VpcId",
    "StackTemplateId",
    "Name",
    "TimeoutInMinutes",
    "Parameters"
  ]
}
```

이 스키마의 경우 ACL 및 BucketName만 필수 실행 파라미터입니다.

새 CT 요청

스키마를 검사한 후 원하는 배포를 생성하기에 충분한 파라미터를 제공하지 않는다고 결정할 수 있습니다. 이 경우 기존 CloudFormation 템플릿을 검사하여 원하는 것과 더 가까운 템플릿을 찾습니다. 필요한 추가 파라미터를 알고 나면 관리 | 기타 | 기타 | CT 생성을 제출합니다.

> **Note**
>
> 기타 모든 | 기타 CT 생성 및 업데이트는 새 CTs에 대해 논의하기 위해 연락을 드릴 AMS 운영자의 주의를 받습니다.

새 CT에 대한 요청을 제출하려면 일반를 통해 AMS 콘솔에 액세스AWS Management Console한 다음 다음 단계를 따릅니다.

1. 왼쪽 탐색 창에서 RFCs 클릭합니다.

   RFCs 대시보드 페이지가 열립니다.
2. 생성을 클릭합니다.

   변경 요청 생성 페이지가 열립니다.
3. 범주 드롭다운 목록에서 관리를 선택하고 하위 범주 및 항목에 대해 기타를 선택합니다. 작업에서 생성을 선택합니다. RFC를 구현하려면 먼저 승인이 필요합니다.
4. CT를 원하는 이유에 대한 정보를 입력합니다. 예를 들어 기존 S3 스토리지 CT 생성을 기반으로 사용자 지정 ACLs을 허용하는 수정된 S3 스토리지 CT 생성을 요청합니다. 그러면 새 CT: 배포 | 고급 스택 구성 요소 | S3 스토리지 | S3 사용자 지정 ACL 생성이 발생합니다. 이 새 CT는 퍼블릭일 수 있습니다.
5. Submit을 클릭합니다.

   RFC 대시보드에 RFC가 표시됩니다.

새 CT 테스트

AWS Managed Services에서 새 CT를 생성한 후에는 해당 CT와 함께 RFC를 제출하여 테스트합니다. AMS와 협력하여 새 CT를 사전 승인한 경우 표준 RFC 제출을 따르고 결과를 확인할 수 있습니다(RFC 제출RFCs 생성 및 제출 참조). 새 CT가 사전 승인되지 않은 경우(명시적 승인 없이 실행되지 않도록 하려면) 실행할 때마다 AMS와 구현에 대해 논의해야 합니다.

빠른 시작

주제

- AMS Resource Scheduler 빠른 시작
- 교차 계정 백업 설정(리전 내)

AMS 변경 유형의 조합을 사용하여 복잡한 작업을 수행할 수 있습니다.

AMS 변경 관리 시스템을 사용하여 AMS Resource Scheduler, 다중 계정 랜딩 존(MALZ) 또는 단일 계정 랜딩 존(SALZ) 계정을 설정할 수 있습니다. 프로세스는 다양합니다. 또한 파일 전송 및 교차 계정 스냅샷을 수행합니다.

AMS Resource Scheduler 빠른 시작

이 빠른 시작 안내서를 사용하여 AMS Advanced에서 비용을 절감하기 위한 태그 기반 인스턴스 스케줄러인 AMS Resource Scheduler를 구현합니다.

AMS Resource Scheduler는 AWS 인스턴스 스케줄러를 기반으로 합니다.

AMS Resource Scheduler 용어

시작하기 전에 AMS Resource Scheduler 용어를 숙지하는 것이 좋습니다.

- 기간: 각 일정에는 인스턴스가 실행되어야 하는 시간(들)을 정의하는 기간이 하나 이상 포함되어야 합니다. 일정은 기간을 두 개 이상 포함할 수 있습니다. 일정에 둘 이상의 기간이 사용되는 경우 Resource Scheduler는 기간 규칙 중 하나 이상이 true일 때 적절한 시작 작업을 적용합니다.
- 시간대: 나중에 참조되는 DefaultTimezone 파라미터에 사용할 수 있는 시간대 값 목록은 TZ 데이터베이스 시간대 목록의 TZ 열을 참조하세요.
- 최대 절전 모드: 최대 절전 모드로 활성화되고 최대 절전 모드 요구 사항을 충족하는 참 EC2 인스턴스로 설정하면 최대 절전 모드로 전환됩니다(suspend-to-disk). EC2 콘솔을 확인하여 인스턴스가 최대 절전 모드로 활성화되어 있는지 확인합니다. Amazon Linux를 실행하는 중지된 Amazon EC2 인스턴스에 최대 절전 모드를 사용합니다.
- 적용: true로 설정하면 정의된 일정에 따라 Resource Scheduler는 실행 기간 외에 수동으로 시작된 경우 실행 리소스를 중지하고 실행 기간 중에 수동으로 중지된 경우 리소스를 시작합니다.
- retain_running: true로 설정하면 기간이 시작되기 전에 인스턴스가 수동으로 시작된 경우 Resource Scheduler가 실행 기간이 끝날 때 인스턴스를 중지하지 못하도록 합니다. 예를 들어, 오전 9시부

터 오후 5시까지 기간이 구성된 인스턴스가 오전 9시 이전에 수동으로 시작된 경우 Resource Scheduler는 오후 5시에 인스턴스를 중지하지 않습니다.

- ssm-maintenance-window: 일정에 유지 AWS Systems Manager 관리 기간을 실행 기간으로 추가합니다. Amazon EC2 인스턴스를 예약하기 위해 배포된 스택과 동일한 계정 및 AWS 리전에 있는 유지 관리 기간의 이름을 지정하면 다른 실행 기간에서 인스턴스를 실행하도록 지정하지 않고 유지 관리 이벤트가 완료되는 경우 Resource Scheduler는 유지 관리 기간이 시작되기 전에 인스턴스를 시작하고 유지 관리 기간이 끝날 때 인스턴스를 중지합니다.

  Resource Scheduler는 초기 구성 중에 지정한 AWS Lambda 빈도를 사용하여 유지 관리 기간이 인스턴스를 시작하기까지 걸리는 시간을 결정합니다. 빈도 AWS CloudFormation 파라미터를 10분 이하로 설정하면 Resource Scheduler는 유지 관리 기간 10분 전에 인스턴스를 시작합니다. 빈도를 10분 이상으로 설정하면 Resource Scheduler는 지정한 빈도와 동일한 분 동안 인스턴스를 시작합니다. 예를 들어 Systems Manager 유지 관리 기간 빈도를 30분으로 설정하면 Resource Scheduler는 유지 관리 기간 30분 전에 인스턴스를 시작합니다.

  자세한 내용은 AWS Systems Manager 유지 관리 기간을 참조하세요.
- override-status: Resource Scheduler의 구성된 일정 시작 및 중지 작업을 일시적으로 재정의합니다. 필드를 실행 중으로 설정하면 Resource Scheduler가 해당 인스턴스를 시작하지만 중지하지는 않습니다. 인스턴스는 사용자가 수동으로 중지할 때까지 실행됩니다. override-status를 중지됨으로 설정하면 Resource Scheduler는 해당 인스턴스를 중지하지만 시작하지는 않습니다. 인스턴스는 수동으로 시작할 때까지 실행되지 않습니다.

AMS Resource Scheduler 구현

AMS 리소스 스케줄러 솔루션을 배포하려면 다음 단계를 따릅니다.

1. 배포 | AMS 리소스 스케줄러 | 솔루션 | 배포(ct-0ywnhc8e5k9z5) RFC를 제출하고 다음 파라미터를 제공합니다.

   - SchedulingActive: 리소스 예약을 활성화하려면 예, 비활성화하려면 아니요입니다. 기본값은 Yes입니다.
   - ScheduledServices: 리소스를 예약할 서비스의 쉼표로 구분된 목록을 입력합니다. 유효한 값에는 Autoscaling, ec2 및 rds의 조합이 포함됩니다. 기본값은 autoscaling,ec2,rds입니다.
   - TagName: 리소스 일정 스키마를 서비스 리소스와 연결하는 태그 키의 이름입니다. 기본값은 일정입니다.

> **Note**
>
> Resource Scheduler 배포는이 태그가 있는 리소스에서만 작동합니다.

- DefaultTimezone: 기본 시간대로 사용할 미국/태평양 형식의 시간대 이름입니다. 기본값은 UTC입니다.

2. 1단계의 RFC가 성공적으로 실행되었다는 확인을 받으면 기간 | 변경 유형 추가를 제출할 수 있습니다.
3. 마지막으로 RFC를 제출하여 2단계에서 생성된 기간에 일정을 추가합니다. 일정 | 변경 유형 추가를 사용합니다.

AMS Resource Scheduler 구현 및 사용 FAQs

AMS Resource Scheduler에 대해 자주 묻는 질문입니다.

Q: 최대 절전 모드를 활성화했지만 EC2 인스턴스가 이를 지원하지 않는 경우 어떻게 되나요?

A: 최대 절전 모드는 인스턴스 메모리(RAM)의 콘텐츠를 Amazon Elastic Block Store(Amazon EBS) 루트 볼륨에 저장합니다. 이 필드를 true로 설정하면 Resource Scheduler가 인스턴스를 중지할 때 인스턴스가 최대 절전 모드로 전환됩니다.

Resource Scheduler가 최대 절전 모드를 사용하도록 설정했지만 인스턴스가 최대 절전 모드에 대해 활성화되지 않았거나 최대 절전 모드 사전 조건을 충족하지 않는 경우 Resource Scheduler는 경고를 로깅하고 인스턴스는 최대 절전 모드 없이 중지됩니다. 자세한 내용은 인스턴스 최대 절전 모드를 참조하세요.

Q: override_status와 enforced를 모두 설정하면 어떻게 되나요?

A: override_status를 running으로 설정하고 enforced를 true로 설정하면(실행 기간 외에 인스턴스가 수동으로 시작되지 않도록 방지) Resource Scheduler가 인스턴스를 중지합니다.

override_status를 중지됨으로 설정하고 enforced를 true로 설정하면(실행 기간 동안 인스턴스가 수동으로 중지되는 것을 방지) Resource Scheduler가 인스턴스를 다시 시작합니다.

> **Note**
>
> 적용이 false인 경우 구성된 재정의 동작이 적용됩니다.

Q: AMS Resource Scheduler가 배포된 후 내 계정에서 리소스 스케줄러를 비활성화하거나 활성화하려면 어떻게 해야 하나요?

A: AMS Resource Scheduler를 비활성화하거나 활성화하려면:

- 비활성화하려면: 상태 | 비활성화를 사용하여 RFC를 생성합니다. SchedulerState를 비활성화로 설정해야 합니다.
- 활성화하려면: 상태 | 활성화를 사용하여 RFC를 생성합니다. SchedulerState를 ENABLE로 설정해야 합니다.

Q AMS Resource Scheduler 기간이 패치 유지 관리 기간에 속하는 경우 어떻게 되나요?

A: Resource Scheduler는 구성된 일정에 따라 작동합니다. 패치 적용이 진행되는 동안 인스턴스를 중지하도록 구성된 경우 패치 적용 기간이 패치 적용이 시작되기 전에 일정에 기간으로 추가되지 않는 한 인스턴스를 중지합니다. 즉, Resource Scheduler는 지정된 기간이 구성되지 않는 한 패치 적용을 위해 중지된 인스턴스를 자동으로 시작하지 않습니다. 패치 유지 관리 기간과의 충돌을 방지하려면 패치에 할당된 기간을 Resource Scheduler 일정에 기간으로 추가합니다. 기존 일정에 기간을 추가하려면 기간 | 추가를 사용하여 RFC를 생성합니다.

Q 다른 EC2 인스턴스에 대해 다른 일정을 설정해야 하는 경우 내 계정 내에서 둘 이상의 일정 설정을 할 수 있습니까?

A: 예, 여러 일정을 생성할 수 있습니다. 각 일정에는 요구 사항에 따라 여러 기간이 있을 수 있습니다. 계정에서 AMS Resource Scheduler가 활성화되면 태그 키가 구성됩니다. 예를 들어 태그 키가 "일정"인 경우 AMS Resource Scheduler의 일정 이름에 해당하는 다른 일정에 따라 태그 값이 다를 수 있습니다. 새 일정을 추가하려면 관리 | AMS 리소스 스케줄러 | 일정 | 추가(ct-2bxelbn765ive) 변경 유형을 사용하여 RFC를 생성할 수 있습니다. 일정 | 추가를 참조하세요.

Q: AMS Resource Scheduler에 지원되는 다양한 변경 유형은 어디에서 찾을 수 있나요?

A: AMS에는 계정에 AMS Resource Scheduler를 배포하고, 활성화 또는 비활성화하고, AMS와 함께 사용할 일정 및 기간을 정의, 추가, 업데이트 및 삭제하고, 일정 및 기간을 설명(자세한 설명 얻기)하는 Resource Scheduler 변경 유형이 있습니다.

교차 계정 백업 설정(리전 내)

AWS Backup 는 두 계정이 동일한 AWS Organization 내에 있는 한 동일한 AWS 리전 내의 한 계정에서 다른 계정으로 스냅샷을 복사할 수 있는 기능을 지원합니다. 예를 들어 AMS Advanced 다중 계정

랜딩 존(MALZ)에서이 빠른 시작을 사용하여 동일한 AWS 리전 내에서 교차 계정 스냅샷 복사본을 설정할 수 있습니다.

자세한 내용은 AWS Backup 및 AWS Organizations 교차 계정 백업 가져오기 기능을 참조하세요.

재해 복구(DR)를 위해 계정 간 스냅샷을 복사합니다. 데이터 보호를 위해 스냅샷을 동일한 AWS 리전 내에서 계정 경계를 넘어 유지해야 하는 요구 사항이 있을 수 있습니다.

개요:

상위 수준에서는 AMS 내에서 교차 계정 백업을 위한 단계입니다.

- AMS 랜딩 존이 호스팅되는 AWS 리전에서 백업을 호스팅할 대상 계정을 생성합니다(1단계).
- 대상 계정에서 백업을 암호화하기 위한 KMS 키 생성(3단계)
- AMS Advanced 랜딩 존과 동일한 리전의 대상 계정에 백업 볼트 생성(4단계)
- 관리 계정에서 교차 계정 설정 활성화(5단계)
- 소스 계정 백업 계획 및 규칙(들) 생성 또는 수정(6단계)

> **Note**
>
> 소스 계정과 대상 계정이 모두 동일한 리전에 있는지 확인합니다. 리전 간 백업을 복사하려면 CA 또는 CSDM에 문의하십시오.

교차 계정 백업을 활성화하고 설정하려면:

1. 백업을 호스팅할 대상 계정을 생성합니다. 이러한 계정이 이미 있는 경우이 단계를 건너뛸 수 있습니다. 계정을 생성하려면 배포 | 관리형 랜딩 존 | 관리 계정 | 애플리케이션 계정 생성(VPC 사용) 변경 유형(ct-1zdasmc2ewzrs)을 사용하여 Management Payer 계정에서 RFC를 제출합니다.
2. [선택 사항] 리소스 또는 스냅샷이 소스 계정(예: Prod)에서 암호화된 경우 암호화에 사용되는 KMS 키를 대상 계정과 공유합니다. 이렇게 하려면 관리 | 고급 스택 구성 요소 | KMS 키 | 변경 유형 업데이트(ct-3ovo7px2vsa6n)를 사용하여 RFC를 제출합니다.
3. 대상 계정에서 Backup Vault 암호화에 사용할 KMS 키를 생성합니다. 이렇게 하려면 배포 | 고급 스택 구성 요소 | KMS 키 | 생성(자동) 변경 유형(ct-1d84keiri1jhg)을 사용하여 RFC를 제출합니다.
4. 대상 계정에서 이전에 생성한 키를 사용하여 백업 볼트를 생성합니다. AWS Backup 볼트는 CFN 수집 자동 변경 유형인 배포 | 수집 | CloudFormation 템플릿의 스택 | 생성(ct-36cn2avfrrj9v)을 사

용하여 생성할 수 있습니다. 동일한 요청에서 소스 계정(들)이 볼트에 액세스할 수 있도록 볼트 액세스 정책을 수정해야 합니다. 다음은 정책 예제입니다.

백업 볼트에 대한 CloudFormation 템플릿 예제:

```
{
  "Description": "Test infrastructure",
  "Resources": {
  "BackupVaultForTesting": {
    "Type": "AWS::Backup::BackupVault",
    "Properties": {
      "BackupVaultName": "backup-vault-for-test",
      "EncryptionKeyArn" : "arn:aws:kms:us-east-2:123456789012:key/227d8xxx-
aefx-44ex-a09x-b90c487b4xxx",
        "AccessPolicy" : {
          "Version": "2012-10-17",
          "Statement": [
            {
              "Sid": "AllowSrcAccountPermissionsToCopy",
              "Effect": "Allow",
              "Action": "backup:CopyIntoBackupVault",
              "Resource": "*",
              "Principal": {
                "AWS": ["arn:aws:iam::987654321098:root"]
              }
            }
          ]
        }
      }
    }
  }
}
```

5. Management Payer 계정에서 교차 계정 백업을 활성화합니다. 이렇게 하려면 관리 | AWS Backup | 백업 계획 | 교차 계정 복사 활성화(관리 계정) 변경 유형(ct-2yja7ihh30ply)을 사용하여 RFC를 제출합니다.
6. 마지막으로 백업이 소싱되는 소스 계정에서 백업을 관리하는 백업 계획의 규칙 또는 규칙을 생성하여 계정 간에 스냅샷을 복사합니다. 이렇게 하려면 배포 | AWS Backup | 백업 계획 | 변경 유형 생성(ct-2hyozbpa0sx0m)을 사용하여 RFC를 제출합니다. 기존 백업 계획을 업데이트해야 하는 경우 다음 정보와 함께 관리 | 기타 | 기타 | 변경 유형 업데이트(ct-0xdawir96cy7k)를 사용하여 RFC를 제출합니다.

1. 백업 계획 이름과 업데이트할 규칙 이름입니다.
2. 대상/ICE 계정 백업 볼트 ARN입니다.
3. 스냅샷을 대상 ICE 볼트에 보관하려는 보존 일/월입니다.

자습서

주제

다음 자습서에서는 CLI를 사용하고 콘솔을 사용하고 Linux 또는 RHEL Amazon EC2 Auto Scaling 그룹(ASG)을 배포하여 고가용성(ct-06mjngx5flwto)으로 2계층 스택을 생성하는 단계를 자세히 설명합니다. Auto Scaling 유사한 tier-and-tie 자습서가 각 자습서(콘솔용 자습서와 CLI용 자습서)를 따르며, 별도의 CTs를 사용하여 리소스를 생성할 때 리소스를 연결할 수 있는 순서로 생성됩니다.

ChangeTypeId를 포함한 모든 CT 옵션에 대한 설명은 managedservices/latest/ctref/ Change Type Reference에서 확인할 수 있습니다.

콘솔 자습서: 고가용성 2계층 스택(Linux/RHEL)

이 섹션에서는 AMS 콘솔을 사용하여 AMS 환경에 고가용성(HA) WordPress 사이트를 배포하는 방법을 설명합니다.

> **Note**
>
> 이 배포 연습은 AMZN Linux 및 RHEL 환경에서 테스트되었습니다.

작업 및 필수 RFCs 요약:

1. 인프라 생성(HA 2계층 스택)
2. CodeDeploy 애플리케이션용 S3 버킷 생성
3. WordPress 애플리케이션 번들을 생성하여 S3 버킷에 업로드합니다.
4. CodeDeploy를 사용하여 애플리케이션 배포
5. WordPress 사이트에 액세스하고 로그인하여 배포를 검증합니다.
6. 배포를 중단합니다.

ChangeTypeId를 포함한 모든 CT 옵션에 대한 설명은 AMS 변경 유형 참조에서 확인할 수 있습니다.

시작하기 전

배포 | 고급 스택 구성 요소 | 고가용성 2계층 스택 | CT 생성은 Auto Scaling 그룹, 로드 밸런서, 데이터베이스, CodeDeploy 애플리케이션 이름 및 배포 그룹(애플리케이션과 동일한 이름)을 생성합니다. CodeDeploy에 대한 자세한 내용은 CodeDeploy란 무엇입니까?를 참조하세요.

이 연습UserData에서는 CodeDeploy가 배포할 수 있는 WordPress 번들을 생성하고 생성하는 방법을 포함하는 고가용성 2계층 스택 RFC를 사용합니다.

예제에 UserData 표시된는 http://169.254.169.254/latest/meta-data/ 사용할 수 있는 EC2 인스턴스 메타데이터 서비스를 쿼리하여 실행 중인 인스턴스 내에서 인스턴스 ID, 리전 등과 같은 인스턴스 메타데이터를 가져옵니다. 사용자 데이터 스크립트의이 줄: `REGION=$(curl 169.254.169.254/latest/meta-data/placement/availability-zone/ | sed 's/[a-z]$//')`는 메타데이터 서비스에서 지원되는 리전의 $REGION 변수로 가용 영역 이름을 검색하고 이를 사용하여 CodeDeploy 에이전트가 다운로드되는 S3 버킷의 URL을 완료합니다. 169.254.169.254 IP는 VPC 내에서만 라우팅할 수 있습니다(모든 VPCs 서비스를 쿼리할 수 있음). 서비스에 대한 자세한 내용은 인스턴스 메타데이터 및 사용자 데이터를 참조하세요. UserData로 입력된 스크립트는 "루트" 사용자로 실행되므로 "sudo" 명령을 사용할 필요가 없습니다.

이 연습에서는 다음 파라미터를 기본값(그림 참조)으로 둡니다.

- Auto Scaling 그룹: `Cooldown=300, DesiredCapacity=2, EBSOptimized=false, HealthCheckGracePeriod=600, IAMInstanceProfile=customer-mc-ec2-instance-profile, InstanceDetailedMonitoring=true, InstanceRootVolumeIops=0, InstanceRootVolumeType=standard, InstanceType=m3.medium, MaxInstances=2, MinInstances=2, ScaleDownPolicyCooldown=300, ScaleDownPolicyEvaluationPeriods=4, ScaleDownPolicyPeriod=60, ScaleDownPolicyScalingAdjustment=-1, ScaleDownPolicyStatistic=Average, ScaleDownPolicyThreshold=35, ScaleMetricName=CPUUtilization, ScaleUpPolicyCooldown=60, ScaleUpPolicyEvaluationPeriods=2, ScaleUpPolicyPeriod=60, ScaleUpPolicyScalingAdjustment=2, ScaleUpPolicyStatistic=Average, ScaleUpPolicyThreshold=75.`
- Load Balancer: `HealthCheckInterval=30, HealthCheckTimeout=5.`
- 데이터베이스: `BackupRetentionPeriod=7, Backups=true, InstanceType=db.m3.medium, IOPS=0, MultiAZ=true, PreferredBackupWindow=22:00-23:00, PreferredMaintenanceWindow=wed:03:32-`

`wed:04:02, StorageEncrypted=false, StorageEncryptionKey="", StorageType=gp2`.

- 애플리케이션: `DeploymentConfigName=CodeDeployDefault.OneAtATime`.

변수 파라미터:

콘솔은 시작 시간에 대한 ASAP 옵션을 제공하며이 연습에서는 이를 사용할 것을 권장합니다. ASAP를 사용하면 승인이 전달되는 즉시 RFC가 실행됩니다.

> **Note**
>
> 표시된 것과 다르게 설정하도록 선택할 수 있는 파라미터가 많습니다. 예제에 표시된 파라미터의 값은 테스트되었지만 적합하지 않을 수 있습니다. 예제에는 필요한 값만 표시됩니다. *## # ##* 글꼴의 값은 계정에 특정하므로 변경해야 합니다.

인프라 생성

이 절차에서는 고가용성 2계층 스택 CT와 S3 스토리지 CT 생성을 차례로 활용합니다.

시작하기 전에 다음 데이터를 수집하면 배포 속도가 빨라집니다.

필수 데이터 HA 스택:

- AutoScalingGroup:
  - UserData:이 값은이 자습서에서 제공됩니다. 여기에는 CodeDeploy에 대한 리소스를 설정하고 CodeDeploy 에이전트를 시작하는 명령이 포함되어 있습니다.
  - AMI-ID:이 값은 Auto Scaling 그룹(ASG)이 실행할 EC2 인스턴스의 운영 체제를 결정합니다. 계정에서 "customer-"로 시작하고 원하는 운영 체제의 AMI를 선택합니다. AMS 콘솔 VPCs -> VPC 세부 정보 페이지에서 AMI IDs를 찾습니다. VPCs 이 연습은 Amazon Linux 또는 RHEL AMI를 사용하도록 구성된 ASGs를 위한 것입니다.
- 데이터베이스:
  - 예제에 표시된 값이 테스트되었지만 이러한 파라미터, DBEngine, EngineVersion 및 LicenseModel은 상황에 따라 설정해야 합니다. 이 자습서에서는 각각 *`MySQL`*, *`8.0.16`*, *`general-public-license`* 값을 사용합니다.

- 이러한 파라미터, DBName, MasterUserPassword 및 MasterUsername은 애플리케이션 번들을 배포할 때 필요합니다. 이 자습서에서는 각각 *wordpressDB*, *p4ssw0rd*, *admin* 등의 값을 사용합니다. DBName은 영숫자만 포함할 수 있습니다.
- RDS DB의 MasterUsername을 입력하면 일반 텍스트로 표시되므로 최대한 빨리 데이터베이스에 로그인하고 암호를 변경하여 보안을 보장합니다.
- RDSSubnetIds의 경우 두 개의 프라이빗 서브넷을 사용합니다. 다음에 "Enter"를 눌러 한 번에 하나씩 입력합니다. 를 사용하여 서브넷 IDs 찾기 AMS SKMS API 참조의 경우 AWS 아티팩트 콘솔의 보고서 탭을 참조하세요. 작업(CLI: list-subnet-summaries) 또는 AMS 콘솔 VPCs.

- LoadBalancer:
  - 자습서에서는 퍼블릭 ELB 서브넷을 사용하기 때문에이 파라미터인 퍼블릭을 true로 설정합니다.
  - ELBSubnetIds: 두 개의 퍼블릭 서브넷을 사용합니다. 다음에 "Enter"를 눌러 한 번에 하나씩 입력합니다. 를 사용하여 서브넷 IDs 찾기 AMS SKMS API 참조의 경우 AWS Artifact 콘솔의 보고서 탭을 참조하세요. 작업(CLI: list-subnet-summaries) 또는 AMS 콘솔 VPCs.
- 애플리케이션: ApplicationName 값은 CodeDeploy 애플리케이션 이름과 CodeDeploy 배포 그룹 이름을 설정합니다. 이를 사용하여 애플리케이션을 배포합니다. 계정에서 고유해야 합니다. 계정에 CodeDeploy 이름이 있는지 확인하려면 CodeDeploy 콘솔을 참조하세요. 이 예제에서는 *WordPress*를 사용하지만 해당 값을 사용할 경우 아직 사용되지 않았는지 확인합니다.

1. 고가용성 스택을 시작합니다.

   a. RFC 생성 페이지의 목록에서 범주 배포, 하위 범주 표준 스택, 고가용성 2계층 스택 및 생성 작업을 선택합니다.

   b. 중요: 고급을 선택하고 표시된 대로 값을 설정합니다.

      별표(*) 옵션에 대한 값만 입력하면 되며, 테스트된 값은 예제에 표시됩니다. 필수가 아닌 빈 옵션은 비워 둘 수 있습니다.

   c. RFC 설명 섹션의 경우:

      ```
      Subject: WP-HA-2-Tier-RFC
      ```

   d. 리소스 정보 섹션에서 AutoScalingGroup, 데이터베이스, LoadBalancer, 애플리케이션 및 태그에 대한 파라미터를 설정합니다.

      또한 "AppName" 태그 키의 목적은 EC2 콘솔에서 ASG 인스턴스를 쉽게 검색할 수 있도록 하기 위한 것입니다.이 태그 키 "Name" 또는 원하는 다른 키 이름을 호출할 수 있습니다. 최대 50개의 태그를 추가할 수 있습니다.

```
UserData:
    #!/bin/bash
    REGION=$(curl 169.254.169.254/latest/meta-data/placement/availability-zone/
 | sed 's/[a-z]$//')
    yum -y install ruby httpd
    chkconfig httpd on
    service httpd start
    touch /var/www/html/status
    cd /tmp
    curl -O https://aws-codedeploy-$REGION.s3.amazonaws.com/latest/install
    chmod +x ./install
    ./install auto
    chkconfig codedeploy-agent on
    service codedeploy-agent start
AmiId:               AMI-ID
Description:         WP-HA-2-Tier-Stack

Database:
    LicenseModel:    general-public-license (USE RADIO BUTTON)
    EngineVersion:   8.0.16
    DBEngine:        MySQL
    RDSSubnetIds:    PRIVATE_AZ1 PRIVATE_AZ2 (ENTER ONE AT A TIME PRESSING
 "ENTER" AFTER EACH)
    MasterUserPassword:  p4ssw0rd
    MasterUsername:      admin
    DBName:              wordpressDB

LoadBalancer:
    Public:              true (USE RADIO BUTTON)
    ELBSubnetIds:        PUBLIC_AZ1 PUBLIC_AZ2

Application:
    ApplicationName:     WordPress

Tags:
    Name:                WP-Rhel-Stack
```

e. 완료되면 제출을 클릭합니다.

2. 생성한 데이터베이스에 로그인하고 암호를 변경합니다.
3. S3 버킷 스택을 시작합니다.

   시작하기 전에 다음 데이터를 수집하면 배포 속도가 빨라집니다.

필수 데이터 S3 버킷:

- VPC-ID:이 값은 S3 버킷의 위치를 결정합니다. 를 사용하여 VPC IDs 찾기 AMS SKMS API 참조의 경우 AWS Artifact 콘솔의 보고서 탭을 참조하세요. 작업(CLI: list-vpc-summaries) 또는 AMS 콘솔 VPCs.
- BucketName:이 값은 S3 버킷 이름을 설정하고 이를 사용하여 애플리케이션 번들을 업로드합니다. 계정의 리전 전체에서 고유해야 하며 대문자를 포함할 수 없습니다. 계정 ID를 BucketName의 일부로 포함하는 것은 필수 사항이 아니지만 나중에 버킷을 더 쉽게 식별할 수 있습니다. 계정에 있는 S3 버킷 이름을 확인하려면 계정의 Amazon S3 콘솔로 이동합니다.

a. RFC 생성 페이지에서 범주 배포, 하위 범주 고급 스택 구성 요소, 항목 S3 스토리지 및 RFC CT 선택 목록에서 생성 작업을 선택합니다.

b. 기본 기본 옵션을 유지하고 표시된 대로 값을 설정합니다.

```
Subject:              S3-Bucket-WP-HA-RFC
Description:          S3BucketForWordPressBundles
BucketName:           ACCOUNT_ID-BUCKET_NAME
AccessControl:        Private
VpcId:                VPC_ID
Name:                 S3-Bucket-WP-HA-Stack
TimeoutInMinutes:     60
```

c. 완료되면 제출을 클릭합니다. 이 변경 유형으로 배포된 버킷은 전체 계정에 대한 전체 읽기/쓰기 액세스를 허용합니다.

애플리케이션 생성, 업로드 및 배포

먼저 WordPress 애플리케이션 번들을 생성한 다음 CodeDeploy CTs를 사용하여 애플리케이션을 생성하고 배포합니다.

1. WordPress를 다운로드하고 파일을 추출한 다음 ./scripts 디렉터리를 생성합니다.

   Linux 명령:

```
wget https://github.com/WordPress/WordPress/archive/master.zip
```

Windows: 브라우저 창에 붙여`https://github.com/WordPress/WordPress/archive/master.zip`넣고 zip 파일을 다운로드합니다.

패키지를 어셈블할 임시 디렉터리를 생성합니다.

Linux:

```
mkdir /tmp/WordPress
```

Windows: "WordPress" 디렉터리를 생성합니다. 나중에 디렉터리 경로를 사용합니다.

2. WordPress 소스를 "WordPress" 디렉터리로 추출하고 ./scripts 디렉터리를 생성합니다.

   Linux:

   ```
   unzip master.zip -d /tmp/WordPress_Temp
   cp -paf /tmp/WordPress_Temp/WordPress-master/* /tmp/WordPress
   rm -rf /tmp/WordPress_Temp
   rm -f master
   cd /tmp/WordPress
   mkdir scripts
   ```

   Windows: 생성한 "WordPress" 디렉터리로 이동하여 여기에 "scripts" 디렉터리를 생성합니다.

   Windows 환경에 있는 경우 스크립트 파일의 브레이크 유형을 Unix(LF)로 설정해야 합니다. 메모장 ++에서 창 오른쪽 하단에 있는 옵션입니다.

3. WordPress 디렉터리에서 CodeDeploy appspec.yml 파일을 생성합니다(예제를 복사하는 경우 들여쓰기를 확인하고 각 공간을 계산합니다). 중요: WordPress 파일(이 경우 WordPress 디렉터리)을 예상 대상(/var/www/html/WordPress)으로 복사하기 위해 "소WordPress" 경로가 올바른지 확인합니다. 예제에서 appspec.yml 파일은 WordPress 파일이 있는 디렉터리에 있으므로 "/"만 있으면 됩니다. 또한 Auto Scaling 그룹에 RHEL AMI를 사용했더라도 "os: linux" 줄을 그대로 둡니다. appspec.yml 파일의 예:

   ```
   version: 0.0
   os: linux
   files:
     - source: /
       destination: /var/www/html/WordPress
   hooks:
     BeforeInstall:
   ```

```
    - location: scripts/install_dependencies.sh
      timeout: 300
      runas: root
  AfterInstall:
    - location: scripts/config_wordpress.sh
      timeout: 300
      runas: root
  ApplicationStart:
    - location: scripts/start_server.sh
      timeout: 300
      runas: root
  ApplicationStop:
    - location: scripts/stop_server.sh
      timeout: 300
      runas: root
```

4. WordPress ./scripts 디렉터리에서 bash 파일 스크립트를 생성합니다.

   먼저 다음 콘텐츠config_wordpress.sh로를 생성합니다(원하는 경우 wp-config.php 파일을 직접 편집할 수 있음).

> **Note**
>
> *DBName*을 HA 스택 RFC에 지정된 값으로 바꿉니다(예: wordpress).
> *DB_MasterUsername*을 HA 스택 RFC에 지정된 MasterUsername 값으로 바꿉니다(예: admin).
> *DB_MasterUserPassword*를 HA 스택 RFC에 지정된 MasterUserPassword 값으로 바꿉니다(예: p4ssw0rd).
> *DB_ENDPOINT*를 HA 스택 RFC의 실행 출력에서 엔드포인트 DNS 이름으로 바꿉니다(예: srt1cz23n45sfg.clgvd67uvydk.us-east-1.rds.amazonaws.com). GetRfc 작업(CLI: get-rfc --rfc-id RFC_ID) 또는 이전에 제출한 HA 스택 RFC의 AMS 콘솔 RFC 세부 정보 페이지에서 이를 찾을 수 있습니다.

```
#!/bin/bash
chmod -R 755 /var/www/html/WordPress
cp /var/www/html/WordPress/wp-config-sample.php /var/www/html/WordPress/wp-
config.php
cd /var/www/html/WordPress
sed -i "s/database_name_here/DBName/g" wp-config.php
sed -i "s/username_here/DB_MasterUsername/g" wp-config.php
```

```
sed -i "s/password_here/DB_MasterUserPassword/g" wp-config.php
sed -i "s/localhost/DB_ENDPOINT/g" wp-config.php
```

5. 동일한 디렉터리에서 다음 콘텐츠`install_dependencies.sh`로를 생성합니다.

```
#!/bin/bash
yum install -y php
yum install -y php-mysql
yum install -y mysql
service httpd restart
```

> **Note**
>
> HTTPS는 상태 확인이 처음부터 작동하도록 시작 시 사용자 데이터의 일부로 설치됩니다.

6. 동일한 디렉터리에서 다음 콘텐츠`start_server.sh`로를 생성합니다.

   - Amazon Linux 인스턴스의 경우 다음을 사용합니다.

```
#!/bin/bash
service httpd start
```

   - RHEL 인스턴스의 경우 다음을 사용합니다(추가 명령은 SELINUX가 WordPress를 수락하도록 허용하는 정책임).

```
#!/bin/bash
setsebool -P  httpd_can_network_connect_db 1
setsebool -P  httpd_can_network_connect 1
chcon -t httpd_sys_rw_content_t /var/www/html/WordPress/wp-content -R
restorecon -Rv /var/www/html
service httpd start
```

7. 동일한 디렉터리에서 다음 콘텐츠`stop_server.sh`로를 생성합니다.

```
#!/bin/bash
service httpd stop
```

8. zip 번들을 생성합니다.

   Linux:

```
$ cd /tmp/WordPress
$ zip -r wordpress.zip .
```

Windows: "WordPress" 디렉터리로 이동하여 모든 파일을 선택하고 zip 파일을 생성합니다. 이름을 wordpress.zip으로 지정해야 합니다.

1. 애플리케이션 번들을 S3 버킷에 업로드

   스택을 계속 배포하려면 패키지가 있어야 합니다.

   생성한 모든 S3 버킷 인스턴스에 자동으로 액세스할 수 있습니다. Bastions(인스턴스 액세스 참조) 또는 S3 콘솔을 통해 액세스하고 drag-and-drop 또는 파일을 찾아 선택하여 CodeDeploy 패키지를 업로드할 수 있습니다.

   쉘 창에서 다음 명령을 사용할 수도 있습니다. zip 파일의 경로가 올바른지 확인하세요.

   ```
   aws s3 cp wordpress/wordpress.zip s3://BUCKET_NAME/
   ```

2. WordPress CodeDeploy 애플리케이션 번들 배포

   필수 데이터 코드 배포 애플리케이션 배포:

   - CodeDeployApplicationName: CodeDeploy 애플리케이션에 지정한 이름입니다.
   - CodeDeployGroupName: CodeDeploy 애플리케이션과 그룹은 모두 HA 스택 RFC에서 CodeDeploy 애플리케이션에 부여한 이름으로 생성되었으므로 CodeDeployApplicationName과 동일한 이름입니다.
   - S3Bucket: S3 버킷에 지정한 이름입니다.
   - S3BundleType 및 S3Key: 배포한 WordPress 애플리케이션 번들의 일부입니다.
   - VpcId: 관련 VPC입니다.

   a. RFC 생성 페이지의 RFC CT 선택 목록에서 범주 배포, 하위 범주 애플리케이션, 항목 CodeDeploy 애플리케이션 및 배포를 선택합니다.

   b. 기본 옵션을 유지하고 표시된 대로 값을 설정합니다.

> **Note**
>
> 이전에 생성한 CodeDeploy 애플리케이션, CodeDeploy 배포 그룹, S3 버킷 및 번들을 참조하세요.

```
Subject:                                WP-CD-Deploy-RFC
Description:                            DeployWordPress
S3Bucket:                               BUCKET_NAME
S3Key:                                  wordpress.zip
S3BundleType:                           zip
CodeDeployApplicationName:              WordPress
CodeDeployDeploymentGroupName:          WordPress
CodeDeployIgnoreApplicationStopFailures: false
RevisionType:                           S3



VpcId:                                  VPC_ID
Name:                                   WP-CD-Deploy-Op
TimeoutInMinutes:                       60
```

c. 완료되면 제출을 클릭합니다.

애플리케이션 배포 검증

WordPress 배포 경로 /WordPress를 사용하여 이전에 생성한 로드 밸런서의 엔드포인트(LoadBalancerCName)로 이동합니다. 예:

```
http://stack-ID-FOR-ELB.us-east-1.elb.amazonaws.com/WordPress
```

다음과 같은 페이지가 표시됩니다.

고가용성 배포 줄이기

배포를 해제하려면 HA 2계층 스택 및 S3 버킷에 대해 스택 삭제 CT를 제출하고 RDS 스냅샷을 삭제하도록 요청할 수 있습니다(10일 후에 자동으로 삭제되지만 그 동안 약간의 비용이 발생함). HA 스택과 S3 버킷의 스택 IDs를 수집한 다음 다음 다음 단계를 따릅니다. 스택 | 삭제를 참조하세요.

콘솔 자습서: 티어 및 타이 WordPress 웹 사이트 배포

이 섹션에서는 AMS 콘솔을 사용하여 AMS 환경에 고가용성(HA) WordPress 사이트를 배포하는 방법을 설명합니다. 이 지침 세트에는 필요한 WordPress CodeDeploy 호환 패키지(예: zip) 파일을 생성하는 예제가 포함되어 있습니다. 리소스 프로비저닝은 "계층"을 구성하기 위해 리소스를 함께 연결할 수 있는 순서를 따릅니다.

> **Note**
>
> 이 배포 연습은 AMZN Linux OS와 함께 사용하도록 설계되었습니다.
> 필수 변수 파라미터는 ## ### 것으로 표기되어 있지만 상황에 맞게 다른 파라미터를 수정하는 것이 좋습니다.

작업 및 필수 RFCs 요약:

1. 인프라를 생성합니다.
   a. MySQL RDS 데이터베이스 클러스터 생성
   b. 로드 밸런서 생성
   c. Auto Scaling 그룹을 생성하여 로드 밸런서에 연결합니다.
   d. CodeDeploy 애플리케이션용 S3 버킷 생성
2. WordPress 애플리케이션 번들 생성(RFC 필요 없음)
3. CodeDeploy를 사용하여 WordPress 애플리케이션 번들을 배포합니다.
   a. CodeDeploy 애플리케이션 생성
   b. CodeDeploy 배포 그룹 생성
   c. WordPress 애플리케이션 번들을 S3 버킷에 업로드(RFC 필요 없음)
   d. CodeDeploy 애플리케이션 배포
4. 배포 검증
5. 배포를 중단합니다.

ChangeTypeId를 포함한 모든 CT 옵션에 대한 설명은 AMS 변경 유형 참조에서 확인할 수 있습니다.

콘솔을 사용하여 RFC 생성(기본)

다음은 콘솔을 사용하여 RFC를 생성할 때마다 따라야 하는 몇 가지 단계입니다.

1. 왼쪽 탐색 창에서 RFCs를 클릭하여 RFCs 목록 페이지를 연 다음 RFC 생성을 클릭합니다.

   RFC 생성 페이지가 열립니다.
2. 변경 유형 찾아보기(기본값) 또는 범주별 선택을 선택합니다.
3. 변경 유형 찾아보기:

   a. 빠른 생성 옵션을 클릭하여 가장 많이 사용되는 변경 유형 중 하나로 RFC를 시작합니다.

      해당 변경 유형에 대한 일반 구성 영역이 열리고 제목 줄이 채워집니다. 변경 유형 세부 정보를 보려면 페이지 상단의 영역을 엽니다.

   b. 모든 변경 유형 영역을 사용합니다.

      카드 또는 테이블 보기를 필터링하거나 전환하거나 변경 유형을 정렬합니다. 원하는 항목을 찾으면 선택하고 페이지 상단에서 RFC 생성을 클릭합니다.

      해당 변경 유형에 대한 일반 구성 영역이 열리고 제목 줄이 채워집니다. 변경 유형 세부 정보를 보려면 페이지 상단의 영역을 엽니다.
4. 범주별로 선택:

   a. 적절한 범주, 하위 범주, 항목 및 작업을 선택합니다.

      페이지 하단에 변경 유형 세부 정보 상자가 나타납니다.

   b. 페이지 하단에서 RFC 생성을 클릭합니다.

   c. 해당 변경 유형에 대한 일반 구성 영역이 열리고 제목 줄이 채워집니다. 변경 유형 세부 정보를 보려면 페이지 상단의 영역을 엽니다.
5. 특정 사용자가 RFC 진행 상황에 대한 알림을 받도록 하려면 이메일 주소를 입력합니다. 변경 유형에 대한 세부 정보를 추가하려면 설명을 입력합니다. 추가 구성 영역을 열어 RFC에 대한 세부 정보를 추가합니다.
6. 예약에서 이 변경 사항을 최대한 빨리 실행 또는 이 변경 사항 예약을 선택합니다. 이 변경 사항을 최대한 빨리 실행을 선택하면 승인이 통과되는 즉시 RFC가 실행됩니다. 이 변경 유형 예약을 선택하면 선택 일정, 시간 및 시간대가 나타나고 RFC가 제출 후 일정에 따라 시작됩니다.
7. 실행 구성 영역에서 변경 유형 파라미터를 구성합니다. 선택적 파라미터를 보려면 추가 구성 영역을 엽니다.
8. 준비가 되면 실행을 클릭합니다.

인프라 생성

대상 AMS 계정의 AWS 콘솔에 로그인한 다음 계정의 AMS 콘솔에 로그인합니다.

다음 절차에서는 리소스 IDs를 사용하여 인프라를 빌드하는 방식으로 RDS 데이터베이스, 로드 밸런서 및 Auto Scaling 그룹을 생성하는 방법을 설명합니다.

RDS 스택 생성

RDS 스택 | 생성을 참조하세요.

ELB 스택 생성

퍼블릭 ELB를 시작합니다.

필수 데이터:

- `VpcId`: 사용 중인 VPC로, 이전에 사용한 VPC와 동일해야 합니다.
- `ELBSubnetIds`: 로드 밸런서가 트래픽을 분산할 서브넷의 배열입니다. 퍼블릭 또는 프라이빗 서브넷을 선택합니다. 를 사용하여 서브넷 IDs 찾기 AMS SKMS API 참조의 경우 AWS Artifact 콘솔의 보고서 탭을 참조하세요. 작업(CLI: list-subnet-summaries) 또는 AMS 콘솔 VPCs.
- `VpcId`: 사용 중인 VPC로, 이전에 사용한 VPC와 동일해야 합니다.

1. RFC 생성 페이지에서 범주 배포, 하위 범주 고급 스택 구성 요소, 항목 로드 밸런서(ELB) 스택을 선택하고 생성을 클릭합니다. 고급을 선택하고 다음에 표시된 기본값(값이 없는 기본값 포함)을 제외한 모든 기본값을 수락합니다.

```
Subject:                        WP-ELB-RFC
ELBSubnetIds:                   PUBLIC_AZ1
                              PUBLIC_AZ2
ELBScheme                       true
ELBCookieExpirationPeriod       600
VpcId:                          VPC_ID
Name:                           WP-Public-ELB
```

2. 완료되면 제출을 클릭합니다.

Auto Scaling 그룹 스택 생성

Auto Scaling 그룹을 시작합니다.

필수 데이터:

- `VpcId`: 사용 중인 VPC로, 이전에 사용한 VPC와 동일해야 합니다.
- `AMI-ID`:이 값은 Auto Scaling 그룹(ASG)이 실행할 EC2 인스턴스의 종류를 결정합니다. 계정에서 "customer-"로 시작하고 원하는 운영 체제의 AMI를 선택해야 합니다. 를 사용하여 AMI IDs 찾기 AMS SKMS API 참조의 경우 AWS 아티팩트 콘솔의 보고서 탭(CLI: list-amis) 또는 AMS 콘솔 VPCs VPCs 세부 정보 페이지를 참조하세요. 이 연습은 Linux ASGs를 위한 것입니다.
- `ASGLoadBalancerNames`: 이전에 생성한 로드 밸런서 - EC2 콘솔 -> 로드 밸런서(왼쪽 탐색창)를 보고 이름을 찾았습니다. 이는 이전에 ELB를 생성할 때 지정한 "이름"이 아닙니다.

1. RFC 생성 페이지에서 범주 배포, 하위 범주 고급 스택 구성 요소, 항목 Auto Scaling 그룹을 선택하고 생성을 클릭합니다. 고급을 선택하고 다음에 표시된 기본값(값이 없는 기본값 포함)을 제외한 모든 기본값을 수락합니다.

   > **Note**
   >
   > 최신 AMS AMI를 지정합니다. 이전에 생성한 ELB를 지정합니다.

```
Subject:                        WP-ASG-RFC
ASGSubnetIds:                   PRIVATE_AZ1
                                                PRIVATE_AZ2
ASGAmiId:                       AMI_ID
VpcId:                          VPC_ID
Name:                           WP_ASG
ASGLoadBalancerNames:           ELB_NAME
ASGUserData:
#!/bin/bash
REGION=$(curl 169.254.169.254/latest/meta-data/placement/availability-zone/ | sed
 's/[a-z]$//')
yum -y install ruby httpd
chkconfig httpd on
service httpd start
touch /var/www/html/status
cd /tmp
curl -O https://aws-codedeploy-$REGION.s3.amazonaws.com/latest/install
chmod +x ./install
./install auto
```

```
chkconfig codedeploy-agent on
service codedeploy-agent start
```

2. 완료되면 제출을 클릭합니다.

S3 스택 생성

S3 버킷을 시작합니다. S3 버킷은 생성한 애플리케이션 번들을 업로드하는 곳입니다.

필수 데이터:

- `VPC-ID`:이 값은 S3 버킷의 위치를 결정하며, 이전에 사용한 VPC와 동일해야 합니다.
- `AccessControl`: 사전 설정된 AccessControl 목록(ACL) 옵션은 `Private`, 및 입니다`PublicRead`. 자세한 내용은 Amazon Simple Storage Service 미리 준비된 ACL을 참조하세요.
- `BucketName`:이 값은 S3 버킷 이름을 설정하며, 이를 사용하여 애플리케이션 번들을 업로드합니다. 계정의 리전 전체에서 고유해야 하며 대문자를 포함할 수 없습니다. 계정 ID를 BucketName의 일부로 포함하는 것은 요구 사항이 아니지만 나중에 버킷을 더 쉽게 식별할 수 있습니다. 계정에 존재하는 S3 버킷 이름을 확인하려면 계정의 Amazon S3 콘솔로 이동합니다.

1. RFC 생성 페이지에서 범주 배포, 하위 범주 고급 스택 구성 요소, 항목 S3 스토리지를 선택하고 생성을 클릭합니다.

   기본 파라미터 옵션을 기본으로 두어 설명된 기본값을 수락할 수 있습니다. 다른 값을 설정하려면 고급을 선택합니다.

   > **Note**
   >
   > 이 변경 유형으로 배포된 버킷은 전체 계정에 대한 전체 읽기/쓰기 액세스를 허용하며, 더 제한된 액세스 권한을 허용하려면 새 변경 유형이 필요할 수 있습니다.

   ```
   Subject:            S3-Bucket-RFC
   BucketName:         ACCOUNT_ID-codedeploy-bundles
   AccessControl:      Private

   VpcId:              VPC_ID
   Name:               S3BucketForWP
   ```

2. 완료되면 제출을 클릭합니다.

WordPress CodeDeploy 번들 생성

이 섹션에서는 애플리케이션 배포 번들을 생성하는 예를 제공합니다.

1. WordPress를 다운로드하고 파일을 추출한 다음 ./scripts 디렉터리를 생성합니다.

   Linux 명령:

   ```
   wget https://github.com/WordPress/WordPress/archive/master.zip
   ```

   Windows: 브라우저 창에 붙여`https://github.com/WordPress/WordPress/archive/master.zip`넣고 zip 파일을 다운로드합니다.

   패키지를 어셈블할 임시 디렉터리를 생성합니다.

   Linux:

   ```
   mkdir /tmp/WordPress
   ```

   Windows: "WordPress" 디렉터리를 생성합니다. 나중에 디렉터리 경로를 사용합니다.
2. WordPress 소스를 "WordPress" 디렉터리로 추출하고 ./scripts 디렉터리를 생성합니다.

   Linux:

   ```
   unzip master.zip -d /tmp/WordPress_Temp
   cp -paf /tmp/WordPress_Temp/WordPress-master/* /tmp/WordPress
   rm -rf /tmp/WordPress_Temp
   rm -f master
   cd /tmp/WordPress
   mkdir scripts
   ```

   Windows: 생성한 "WordPress" 디렉터리로 이동하여 여기에 "scripts" 디렉터리를 생성합니다.

   Windows 환경에 있는 경우 스크립트 파일의 브레이크 유형을 Unix(LF)로 설정해야 합니다. 메모장 ++에서 창 오른쪽 하단에 있는 옵션입니다.
3. WordPress 디렉터리에서 CodeDeploy appspec.yml 파일을 생성합니다(예제를 복사하는 경우 들여쓰기를 확인하고 각 공간을 계산합니다). 중요: WordPress 파일(이 경우 WordPress 디렉터리)을 예상 대상(/var/www/html/WordPress)으로 복사하기 위해 "소WordPress" 경로가 올바른지 확인합니다. 예제에서 appspec.yml 파일은 WordPress 파일이 있는 디렉터리에 있으므로 "/"만 있으

면 됩니다. 또한 Auto Scaling 그룹에 RHEL AMI를 사용했더라도 "os: linux" 줄을 그대로 둡니다. appspec.yml 파일의 예:

```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/html/WordPress
hooks:
  BeforeInstall:
    - location: scripts/install_dependencies.sh
      timeout: 300
      runas: root
  AfterInstall:
    - location: scripts/config_wordpress.sh
      timeout: 300
      runas: root
  ApplicationStart:
    - location: scripts/start_server.sh
      timeout: 300
      runas: root
  ApplicationStop:
    - location: scripts/stop_server.sh
      timeout: 300
      runas: root
```

4. WordPress ./scripts 디렉터리에서 bash 파일 스크립트를 생성합니다.

   먼저 다음 콘텐츠`config_wordpress.sh`로를 생성합니다(원하는 경우 wp-config.php 파일을 직접 편집할 수 있음).

   > **Note**
   >
   > *DBName*을 HA 스택 RFC에 지정된 값으로 바꿉니다(예: `wordpress`).
   > *DB_MasterUsername*을 HA 스택 RFC에 지정된 `MasterUsername` 값으로 바꿉니다(예: `admin`).
   > *DB_MasterUserPassword*를 HA 스택 RFC에 지정된 `MasterUserPassword` 값으로 바꿉니다(예: `p4ssw0rd`).
   > *DB_ENDPOINT*를 HA 스택 RFC의 실행 출력에서 엔드포인트 DNS 이름으로 바꿉니다(예: `srt1cz23n45sfg.clgvd67uvydk.us-east-1.rds.amazonaws.com`). GetRfc 작업

(CLI: get-rfc --rfc-id RFC_ID) 또는 이전에 제출한 HA 스택 RFC의 AMS 콘솔 RFC 세부 정보 페이지에서 이를 찾을 수 있습니다.

```
#!/bin/bash
chmod -R 755 /var/www/html/WordPress
cp /var/www/html/WordPress/wp-config-sample.php /var/www/html/WordPress/wp-
config.php
cd /var/www/html/WordPress
sed -i "s/database_name_here/DBName/g" wp-config.php
sed -i "s/username_here/DB_MasterUsername/g" wp-config.php
sed -i "s/password_here/DB_MasterUserPassword/g" wp-config.php
sed -i "s/localhost/DB_ENDPOINT/g" wp-config.php
```

5. 동일한 디렉터리에서 다음 콘텐츠`install_dependencies.sh`로를 생성합니다.

```
#!/bin/bash
yum install -y php
yum install -y php-mysql
yum install -y mysql
service httpd restart
```

> **Note**
>
> HTTPS는 상태 확인이 처음부터 작동하도록 시작 시 사용자 데이터의 일부로 설치됩니다.

6. 동일한 디렉터리에서 다음 콘텐츠`start_server.sh`로를 생성합니다.

   - Amazon Linux 인스턴스의 경우 다음을 사용합니다.

   ```
   #!/bin/bash
   service httpd start
   ```

   - RHEL 인스턴스의 경우 다음을 사용합니다(추가 명령은 SELINUX가 WordPress를 수락하도록 허용하는 정책임).

   ```
   #!/bin/bash
   setsebool -P  httpd_can_network_connect_db 1
   setsebool -P  httpd_can_network_connect 1
   chcon -t httpd_sys_rw_content_t /var/www/html/WordPress/wp-content -R
   ```

```
restorecon -Rv /var/www/html
service httpd start
```

7. 동일한 디렉터리에서 다음 콘텐츠stop_server.sh로를 생성합니다.

```
#!/bin/bash
service httpd stop
```

8. zip 번들을 생성합니다.

   Linux:

```
$ cd /tmp/WordPress
$ zip -r wordpress.zip .
```

   Windows: "WordPress" 디렉터리로 이동하여 모든 파일을 선택하고 zip 파일을 생성합니다. 이름을 wordpress.zip으로 지정해야 합니다.

CodeDeploy를 사용하여 WordPress 애플리케이션 번들 배포

CodeDeploy는 Amazon EC2 인스턴스에 대한 애플리케이션 배포를 자동화하는 AWS 배포 서비스입니다. 프로세스의이 부분에는 CodeDeploy 애플리케이션 생성, CodeDeploy 배포 그룹 생성, CodeDeploy를 사용하여 애플리케이션 배포가 포함됩니다.

CodeDeploy 애플리케이션 생성

CodeDeploy 애플리케이션은 AWS CodeDeploy에서 배포 중에 올바른 개정, 배포 구성 및 배포 그룹을 참조하는 데 사용하는 이름 또는 컨테이너입니다. 이 경우 배포 구성은 이전에 생성한 WordPress 번들입니다.

필수 데이터:

- `VpcId`: 사용 중인 VPC로, 이전에 사용한 VPC와 동일해야 합니다.
- `CodeDeployApplicationName`: 계정에서 고유해야 합니다. CodeDeploy 콘솔에서 기존 애플리케이션 이름을 확인합니다.

1. WordPress용 CodeDeploy 애플리케이션 생성

RFC 생성 페이지에서 범주 배포, 하위 범주 애플리케이션, CodeDeploy 애플리케이션 및 RFC CT 선택 목록에서 생성 작업을 선택합니다. 기본을 선택하고 표시된 대로 값을 설정합니다. 완료되면 제출을 클릭합니다.

```
Subject:                        CD-WP-App-RFC
CodeDeployApplicationName:      WordPress
VpcId:                          VPC_ID
Name:                           WP-CD-App
```

2. 완료되면 제출을 클릭합니다.

CodeDeploy 배포 그룹 생성

CodeDeploy 배포 그룹을 생성합니다.

CodeDeploy 배포 그룹은 배포를 대상으로 하는 개별 인스턴스 세트를 정의합니다.

필수 데이터:

- `VpcId`: 사용 중인 VPC로, 이전에 사용한 VPC와 동일해야 합니다.
- `CodeDeployApplicationName`: 이전에 생성한 값을 사용합니다.
- `CodeDeployAutoScalingGroups`: 이전에 생성한 Auto Scaling 그룹의 이름을 사용합니다.
- `CodeDeployDeploymentGroupName`: 배포 그룹의 이름입니다. 이 이름은 배포 그룹과 연결된 각 애플리케이션에 대해 고유해야 합니다.
- `CodeDeployServiceRoleArn`: 예제에 제공된 공식을 사용합니다.

1. RFC 생성 페이지에서 범주 배포, 하위 범주 애플리케이션, CodeDeploy 배포 그룹 항목 및 RFC CT 선택 목록에서 생성 작업을 선택합니다. 고급을 선택하고 표시된 대로 값을 설정합니다(RFC에는 주체만 필요). 완료되면 제출을 클릭합니다.

> **Note**
>
> CodeDeploy 서비스 역할 ARN을이 형식으로 참조`"arn:aws:iam::085398962942:role/aws-codedeploy-role"`하고 "ASG_NAME"에 대해 이전에 생성된 Auto Scaling 그룹 이름을 사용합니다.

```
Description:                    Create CodeDeploy Deployment Group for WP
CodeDeployApplicationName:      WordPress
CodeDeployAutoScalingGroups:    ASG_NAME
CodeDeployDeploymentConfigName: CodeDeployDefault.HalfAtATime
CodeDeployDeploymentGroupName:  WP CD Group
CodeDeployServiceRoleArn:       arn:aws:iam::ACCOUNT_ID:role/aws-codedeploy-role

VpcId:                          VPC_ID
Name:                           WP Deployment Group
```

2. 완료되면 제출을 클릭합니다.

WordPress 애플리케이션 업로드

생성한 모든 S3 버킷 인스턴스에 자동으로 액세스할 수 있습니다. Bastions(인스턴스 액세스 참조) 또는 S3 콘솔을 통해 액세스하고 CodeDeploy 번들을 업로드할 수 있습니다. 스택을 계속 배포하려면 번들이 있어야 합니다. 이 예제에서는 이전에 생성한 버킷 이름을 사용합니다.

이 AWS 명령을 사용하여 번들을 압축할 수 있습니다.

```
aws s3 cp wordpress/wordpress.zip s3://ACCOUNT_ID-codedeploy-bundles/
```

CodeDeploy를 사용하여 WordPress 애플리케이션 배포

CodeDeploy 애플리케이션을 배포합니다.

필수 데이터:

- VPC-ID: 사용 중인 VPC로, 이전에 사용한 VPC와 동일해야 합니다.
- CodeDeployApplicationName: 이전에 생성한 CodeDeploy 애플리케이션의 이름을 사용합니다.
- CodeDeployDeploymentGroupName: 이전에 생성한 CodeDeploy 배포 그룹의 이름을 사용합니다.
- S3Location (애플리케이션 번들을 업로드한 위치): S3Bucket: 이전에 생성한 BucketName S3BundleType 및 S3Key: S3 스토어에 배치한 번들의 유형 및 이름입니다.

1. WordPress CodeDeploy 애플리케이션 번들 배포

RFC 생성 페이지의 RFC CT 선택 목록에서 범주 배포, 하위 범주 애플리케이션, 항목 CodeDeploy 애플리케이션 및 배포를 선택합니다. 기본을 선택하고 표시된 대로 값을 설정합니다. 완료되면 제출을 클릭합니다.

> **Note**
>
> 이전에 생성한 CodeDeploy 애플리케이션, CodeDeploy 배포 그룹, S3 버킷 및 번들을 참조하세요.

```
Subject:                          WP-CD-Deploy-RFC
CodeDeployApplicationName:        WordPress
CodeDeployDeploymentGroupName:    WPCDGroup
RevisionType:                     S3
S3Bucket:                         ACCOUNT_ID-codedeploy-bundles
S3BundleType:                     zip
S3Key:                            wordpress.zip
VpcId:                            VPC_ID
Name:                             WordPress
```

2. 완료되면 제출을 클릭합니다.

애플리케이션 배포 검증

WordPress 배포 경로 /WordPress를 사용하여 이전에 생성한 로드 밸런서의 엔드포인트(ELB CName)로 이동합니다. 예:

```
http://stack-ID-FOR-ELB.us-east-1.elb.amazonaws.com/WordPress
```

애플리케이션 배포 해체

배포를 해제하려면 RDS 데이터베이스 스택, 애플리케이션 로드 밸런서, Auto Scaling 그룹, S3 버킷, 모든의 코드 배포 애플리케이션 및 그룹 --6 RFCs에 대해 스택 삭제 CT를 제출합니다. 또한 삭제할 RDS 스냅샷에 대한 서비스 요청을 제출할 수 있습니다(10일 후에 자동으로 삭제되지만 그 동안 약간의 비용이 발생함). 모든의 스택 IDs 수집한 다음 다음 단계를 따릅니다. 스택 | 삭제를 참조하세요.

CLI 자습서: 고가용성 2계층 스택(Linux/RHEL)

이 섹션에서는 AMS CLI를 사용하여 AMS 환경에 고가용성(HA) 2계층 스택을 배포하는 방법을 설명합니다.

> **Note**
>
> 이 배포 연습은 AMZN Linux 및 RHEL 환경에서 테스트되었습니다.

작업 및 필수 RFCs 요약:

1. 인프라 생성(HA 2계층 스택)
2. CodeDeploy 애플리케이션용 S3 버킷 생성
3. WordPress 애플리케이션 번들을 생성하여 S3 버킷에 업로드합니다.
4. CodeDeploy를 사용하여 애플리케이션 배포
5. WordPress 사이트에 액세스하고 로그인하여 배포를 검증합니다.

시작하기 전

배포 | 고급 스택 구성 요소 | 고가용성 2계층 스택 고급 | CT 생성은 Auto Scaling 그룹, 로드 밸런서, 데이터베이스, CodeDeploy 애플리케이션 이름 및 배포 그룹(애플리케이션과 동일한 이름)을 생성합니다. CodeDeploy에 대한 자세한 내용은 CodeDeploy란 무엇입니까?를 참조하세요.

이 연습에서는 UserData가 포함된 고가용성 2계층 스택(고급) RFC를 사용하고 CodeDeploy가 배포할 수 있는 WordPress 번들을 생성하는 방법을 설명합니다.

예제에 `UserData` 표시된는 http://169.254.169.254/latest/meta-data/ 사용할 수 있는 EC2 인스턴스 메타데이터 서비스를 쿼리하여 실행 중인 인스턴스 내에서 인스턴스 ID, 리전 등과 같은 인스턴스 메타데이터를 가져옵니다. 사용자 데이터 스크립트의이 줄: `REGION=$(curl 169.254.169.254/latest/meta-data/placement/availability-zone/ | sed 's/[a-z]$//')`는 메타데이터 서비스에서 지원되는 리전의 $REGION 변수로 가용 영역 이름을 검색하고 이를 사용하여 CodeDeploy 에이전트가 다운로드되는 S3 버킷의 URL을 완료합니다. 169.254.169.254 IP는 VPC 내에서만 라우팅할 수 있습니다(모든 VPCs 서비스를 쿼리할 수 있음). 서비스에 대한 자세한 내용은 인스턴스 메타데이터 및 사용자 데이터를 참조하세요. UserData로 입력된 스크립트는 "루트" 사용자로 실행되므로 "sudo" 명령을 사용할 필요가 없습니다.

이 연습에서는 다음 파라미터를 기본값(그림 참조)으로 둡니다.

- Auto Scaling 그룹: `Cooldown=300, DesiredCapacity=2, EBSOptimized=false, HealthCheckGracePeriod=600, IAMInstanceProfile=customer-mc-ec2-instance-profile, InstanceDetailedMonitoring=true, InstanceRootVolumeIops=0, InstanceRootVolumeType=standard, InstanceType=m3.medium, MaxInstances=2, MinInstances=2, ScaleDownPolicyCooldown=300, ScaleDownPolicyEvaluationPeriods=4, ScaleDownPolicyPeriod=60, ScaleDownPolicyScalingAdjustment=-1, ScaleDownPolicyStatistic=Average, ScaleDownPolicyThreshold=35, ScaleMetricName=CPUUtilization, ScaleUpPolicyCooldown=60, ScaleUpPolicyEvaluationPeriods=2, ScaleUpPolicyPeriod=60, ScaleUpPolicyScalingAdjustment=2, ScaleUpPolicyStatistic=Average, ScaleUpPolicyThreshold=75`.
- Load Balancer: `HealthCheckInterval=30, HealthCheckTimeout=5`.
- 데이터베이스: `BackupRetentionPeriod=7, Backups=true, InstanceType=db.m3.medium, IOPS=0, MultiAZ=true, PreferredBackupWindow=22:00-23:00, PreferredMaintenanceWindow=wed:03:32-wed:04:02, StorageEncrypted=false, StorageEncryptionKey="", StorageType=gp2`.
- 애플리케이션: `DeploymentConfigName=CodeDeployDefault.OneAtATime`.
- S3 버킷: `AccessControl=Private`.

추가 설정:

`RequestedStartTime` RFC를 예약하려는 `RequestedEndTime` 경우 및 : Time.is 사용하여 올바른 UTC 시간을 확인할 수 있습니다. 제공된 예제는 적절하게 조정해야 합니다. 시작 시간이 경과하면 RFC를 진행할 수 없습니다. 또는 이러한 값을 끄면 승인이 전달되는 즉시 실행되는 ASAP RFC를 생성할 수 있습니다.

> **Note**
>
> 표시된 것과 다르게 설정하도록 선택할 수 있는 파라미터가 많습니다. 예제에 표시된 파라미터의 값은 테스트되었지만 적합하지 않을 수 있습니다.

인프라 생성

시작하기 전에 다음 데이터를 수집하면 배포 속도가 빨라집니다.

필수 데이터 HA 스택:

- AutoScalingGroup:
  - `UserData`:이 값은이 자습서에서 제공됩니다. 여기에는 CodeDeploy에 대한 리소스를 설정하고 CodeDeploy 에이전트를 시작하는 명령이 포함되어 있습니다.
  - `AMI-ID`:이 값은 Auto Scaling 그룹(ASG)이 실행할 EC2 인스턴스의 종류를 결정합니다. 계정에서 "customer-"로 시작하고 원하는 운영 체제의 AMI를 선택해야 합니다. 를 사용하여 AMI IDs를 찾습니다. AMS SKMS API 참조의 경우 AWS 아티팩트 콘솔의 보고서 탭(CLI: list-amis) 또는 AMS 콘솔 VPCs VPCs 세부 정보 페이지를 참조하세요. 이 연습은 Linux ASGs를 위한 것입니다.
- 데이터베이스:
  - 예제에 표시된 값이 테스트되었지만 상황에 따라 이러한 파라미터 `EngineVersion`, 및 `DBEngine`를 설정해야 `LicenseModel` 합니다.
  - 이러한 파라미터 `RDSSubnetIds`, `MasterUsername`, 및 `DBNameMasterUserPassword`는 애플리케이션 번들을 배포할 때 필요합니다. RDSSubnetIds의 경우 두 개의 프라이빗 서브넷을 사용합니다.
- LoadBalancer:
  - 예제에 표시된 값이 테스트되었지만 상황에 따라 이러한 파라미터 `EngineVersion`, 및 `DBEngine`를 설정해야 `LicenseModel` 합니다.
  - `ELBSubnetIds`: 두 개의 퍼블릭 서브넷을 사용합니다.
- 애플리케이션:이 `ApplicationName` 값은 CodeDeploy 애플리케이션 이름과 CodeDeploy 배포 그룹 이름을 설정합니다. 이를 사용하여 애플리케이션을 배포합니다. 계정에서 고유해야 합니다. 계정에 CodeDeploy 이름이 있는지 확인하려면 CodeDeploy 콘솔을 참조하세요. 이 예제에서는 "WordPress"를 사용하지만 해당 값을 사용할 경우 아직 사용되지 않았는지 확인합니다.

이 절차는 고가용성 2계층 스택(고급) CT(ct-06mjngx5flwto) 및 S3 스토리지 CT 생성 (ct-1a68ck03fn98r)을 활용합니다. 인증된 계정에서 명령줄의 다음 단계를 따릅니다.

1. 인프라 스택을 시작합니다.

   a. HA 2계층 스택 CT의 실행 파라미터 JSON 스키마를 CreateStackParams.json.

```
aws amscm get-change-type-version --change-type-id "ct-06mjngx5flwto"
 --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
 CreateStackParams.json
```

b. 스키마를 수정합니다. 변수를 적절하게 바꿉니다. 예를 들어 ASG가 생성할 EC2 인스턴스에 대해 원하는 OS를 사용합니다. 나중에 애플리케이션을 배포하는 데 `ApplicationName` 사용할를 기록합니다. 최대 50개의 태그를 추가할 수 있습니다.

```
{
"Description":      "HA two tier stack for WordPress",
"Name":             "WordPressStack",
"TimeoutInMinutes":  360,
"Tags": [
        {
            "Key": "ApplicationName",
            "Value": "WordPress"
        }
    ],
"AutoScalingGroup": {
            "AmiId":    "AMI-ID",
            "UserData": "#!/bin/bash \n
            REGION=$(curl 169.254.169.254/latest/meta-data/placement/
availability-zone/ | sed 's/[a-z]$//') \n
            yum -y install ruby httpd \n
            chkconfig httpd on \n
            service httpd start \n
            touch /var/www/html/status \n
            cd /tmp \n
            curl -O https://aws-codedeploy-$REGION.s3.amazonaws.com/latest/
install \n
            chmod +x ./install \n
            ./install auto \n
            chkconfig codedeploy-agent on \n
            service codedeploy-agent start"
    },
    "LoadBalancer": {
        "Public":               true,
        "HealthCheckTarget":    "HTTP:80/status"
    },
    "Database":     {
        "DBEngine":             "MySQL",
        "DBName":               "wordpress",
        "EngineVersion":        "8.0.16 ",
        "LicenseModel":         "general-public-license",
        "MasterUsername":       "admin",
        "MasterUserPassword":   "p4ssw0rd"
    },
```

```
    "Application":  {
    "ApplicationName":  "WordPress"
        }
}
```

c. CreateRfc JSON 템플릿을 CreateStackRfc.json:이라는 현재 폴더의 파일로 출력합니다.

```
aws amscm create-rfc --generate-cli-skeleton > CreateStackRfc.json
```

d. 다음과 같이 RFC 템플릿을 수정하고 저장하면 콘텐츠를 삭제하고 바꿀 수 있습니다. 이제 RequestedStartTime 및 RequestedEndTime는 선택 사항입니다. 제외하면 승인되는 즉시 실행되는 ASAP RFC가 생성됩니다(일반적으로 자동으로 발생함). 예약된 RFC를 제출하려면 해당 값을 추가합니다.

```
{
"ChangeTypeVersion":    "3.0",
"ChangeTypeId":         "ct-06mjngx5flwto",
"Title":                "HA-Stack-For-WP-RFC"
}
```

e. CreateStackRfc.json 파일과 CreateStackParams.json 실행 파라미터 파일을 지정하여 RFC를 생성합니다.

```
aws amscm create-rfc --cli-input-json file://CreateStackRfc.json --execution-
parameters file://CreateStackParams.json
```

응답에서 RFC ID를 수신합니다. 후속 단계를 위해 ID를 저장합니다.

f. RFC를 제출합니다.

```
aws amscm submit-rfc --rfc-id  RFC_ID
```

RFC가 성공하면 출력이 수신되지 않습니다.

g. RFC 상태를 확인하려면를 실행합니다.

```
aws amscm get-rfc --rfc-id RFC_ID
```

RFC ID를 기록해 둡니다.

2. S3 버킷 시작

시작하기 전에 다음 데이터를 수집하면 배포 속도가 빨라집니다.

필수 데이터 S3 버킷:

- VPC-ID:이 값은 S3 버킷의 위치를 결정합니다. 이전에 사용한 것과 동일한 VPC ID를 사용합니다.
- BucketName:이 값은 S3 버킷 이름을 설정하며, 이를 사용하여 애플리케이션 번들을 업로드합니다. 계정의 리전 전체에서 고유해야 하며 대문자를 포함할 수 없습니다. 계정 ID를 BucketName의 일부로 포함하는 것은 요구 사항이 아니지만 나중에 버킷을 더 쉽게 식별할 수 있습니다. 계정에 있는 S3 버킷 이름을 확인하려면 계정의 Amazon S3 콘솔로 이동합니다.

a. S3 스토리지 생성 CT에 대한 실행 파라미터 JSON 스키마를 CreateS3StoreParams.json.

```
aws amscm get-change-type-version --change-type-id "ct-1a68ck03fn98r"
 --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
 CreateS3StoreParams.json
```

b. 다음과 같이 스키마를 수정하면 콘텐츠를 삭제하고 바꿀 수 있습니다. *VPC_ID*를 적절하게 바꿉니다. 예제의 값은 테스트되었지만 적합하지 않을 수 있습니다.

> **Tip**
>
> 는 계정 리전 전체에서 고유BucketName해야 하며 대문자를 포함할 수 없습니다. 계정 ID를 BucketName의 일부로 포함하는 것은 요구 사항이 아니지만 나중에 버킷을 더 쉽게 식별할 수 있습니다. 계정에 있는 S3 버킷 이름을 확인하려면 계정의 Amazon S3 콘솔로 이동합니다.

```
{
"Description":      "S3BucketForWordPressBundle",
"VpcId":            "VPC_ID",
"StackTemplateId":  "stm-s2b72beb000000000",
"Name":             "S3BucketForWP",
"TimeoutInMinutes":  60,
"Parameters":   {
    "AccessControl":    "Private",
    "BucketName":       "ACCOUNT_ID-BUCKET_NAME"
    }
```

```
}
```

c. CreateRfc용 JSON 템플릿을 CreateS3StoreRfc.json:

```
aws amscm create-rfc --generate-cli-skeleton > CreateS3StoreRfc.json
```

d. CreateS3StoreRfc.json 파일을 수정하고 저장하면 콘텐츠를 삭제하고 바꿀 수 있습니다. 이제 `RequestedStartTime` 및 `RequestedEndTime`는 선택 사항입니다. 제외하면 승인되는 즉시 실행되는 ASAP RFC가 생성됩니다(일반적으로 자동으로 발생함). 예약된 RFC를 제출하려면 해당 값을 추가합니다.

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-1a68ck03fn98r",
"Title":                "S3-Stack-For-WP-RFC"
}
```

e. CreateS3StoreRfc.json 파일과 CreateS3StoreParams.json 실행 파라미터 파일을 지정하여 RFC를 생성합니다.

```
aws amscm create-rfc --cli-input-json file://CreateS3StoreRfc.json  --
execution-parameters file://CreateS3StoreParams.json
```

응답에서 새 RFC의 RfcId를 수신합니다. 후속 단계를 위해 ID를 저장합니다.

f. RFC를 제출합니다.

```
aws amscm submit-rfc --rfc-id RFC_ID
```

RFC가 성공하면 출력이 수신되지 않습니다.

g. RFC 상태를 확인하려면를 실행합니다.

```
aws amscm get-rfc --rfc-id RFC_ID
```

애플리케이션 생성, 업로드 및 배포

먼저 WordPress 애플리케이션 번들을 생성한 다음 CodeDeploy CTs를 사용하여 애플리케이션을 생성하고 배포합니다.

1. WordPress를 다운로드하고 파일을 추출한 다음 ./scripts 디렉터리를 생성합니다.

   Linux 명령:

   ```
   wget https://github.com/WordPress/WordPress/archive/master.zip
   ```

   Windows: 브라우저 창에 붙여`https://github.com/WordPress/WordPress/archive/master.zip`넣고 zip 파일을 다운로드합니다.

   패키지를 어셈블할 임시 디렉터리를 생성합니다.

   Linux:

   ```
   mkdir /tmp/WordPress
   ```

   Windows: "WordPress" 디렉터리를 생성합니다. 나중에 디렉터리 경로를 사용합니다.

2. WordPress 소스를 "WordPress" 디렉터리로 추출하고 ./scripts 디렉터리를 생성합니다.

   Linux:

   ```
   unzip master.zip -d /tmp/WordPress_Temp
   cp -paf /tmp/WordPress_Temp/WordPress-master/* /tmp/WordPress
   rm -rf /tmp/WordPress_Temp
   rm -f master
   cd /tmp/WordPress
   mkdir scripts
   ```

   Windows: 생성한 "WordPress" 디렉터리로 이동하여 여기에 "scripts" 디렉터리를 생성합니다.

   Windows 환경에 있는 경우 스크립트 파일의 브레이크 유형을 Unix(LF)로 설정해야 합니다. 메모장 ++에서 창 오른쪽 하단에 있는 옵션입니다.

3. WordPress 디렉터리에서 CodeDeploy appspec.yml 파일을 생성합니다(예제를 복사하는 경우 들여쓰기를 확인하고 각 공간을 계산합니다). 중요: WordPress 파일(이 경우 WordPress 디렉터리)을 예상 대상(/var/www/html/WordPress)으로 복사하기 위해 "소WordPress" 경로가 올바른지 확인합니다. 예제에서 appspec.yml 파일은 WordPress 파일이 있는 디렉터리에 있으므로 "/"만 있으면 됩니다. 또한 Auto Scaling 그룹에 RHEL AMI를 사용했더라도 "os: linux" 줄을 그대로 둡니다. appspec.yml 파일의 예:

   ```
   version: 0.0
   ```

```
os: linux
files:
  - source: /
    destination: /var/www/html/WordPress
hooks:
  BeforeInstall:
    - location: scripts/install_dependencies.sh
      timeout: 300
      runas: root
  AfterInstall:
    - location: scripts/config_wordpress.sh
      timeout: 300
      runas: root
  ApplicationStart:
    - location: scripts/start_server.sh
      timeout: 300
      runas: root
  ApplicationStop:
    - location: scripts/stop_server.sh
      timeout: 300
      runas: root
```

4. WordPress ./scripts 디렉터리에서 bash 파일 스크립트를 생성합니다.

   먼저 다음 콘텐츠`config_wordpress.sh`로를 생성합니다(원하는 경우 wp-config.php 파일을 직접 편집할 수 있음).

   > **Note**
   >
   > *`DBName`*을 HA 스택 RFC에 지정된 값으로 바꿉니다(예: `wordpress`).
   > *`DB_MasterUsername`*을 HA 스택 RFC에 지정된 `MasterUsername` 값으로 바꿉니다(예: `admin`).
   > *`DB_MasterUserPassword`*를 HA 스택 RFC에 지정된 `MasterUserPassword` 값으로 바꿉니다(예: `p4ssw0rd`).
   > *`DB_ENDPOINT`*를 HA 스택 RFC의 실행 출력에서 엔드포인트 DNS 이름으로 바꿉니다(예: `srt1cz23n45sfg.clgvd67uvydk.us-east-1.rds.amazonaws.com`). GetRfc 작업(CLI: get-rfc --rfc-id RFC_ID) 또는 이전에 제출한 HA 스택 RFC의 AMS 콘솔 RFC 세부 정보 페이지에서 이를 찾을 수 있습니다.

```
#!/bin/bash
```

```
chmod -R 755 /var/www/html/WordPress
cp /var/www/html/WordPress/wp-config-sample.php /var/www/html/WordPress/wp-
config.php
cd /var/www/html/WordPress
sed -i "s/database_name_here/DBName/g" wp-config.php
sed -i "s/username_here/DB_MasterUsername/g" wp-config.php
sed -i "s/password_here/DB_MasterUserPassword/g" wp-config.php
sed -i "s/localhost/DB_ENDPOINT/g" wp-config.php
```

5. 동일한 디렉터리에서 다음 콘텐츠`install_dependencies.sh`로를 생성합니다.

```
#!/bin/bash
yum install -y php
yum install -y php-mysql
yum install -y mysql
service httpd restart
```

> **Note**
>
> HTTPS는 상태 확인이 처음부터 작동하도록 시작 시 사용자 데이터의 일부로 설치됩니다.

6. 동일한 디렉터리에서 다음 콘텐츠`start_server.sh`로를 생성합니다.

   - Amazon Linux 인스턴스의 경우 다음을 사용합니다.

   ```
   #!/bin/bash
   service httpd start
   ```

   - RHEL 인스턴스의 경우 다음을 사용합니다(추가 명령은 SELINUX가 WordPress를 수락하도록 허용하는 정책임).

   ```
   #!/bin/bash
   setsebool -P  httpd_can_network_connect_db 1
   setsebool -P  httpd_can_network_connect 1
   chcon -t httpd_sys_rw_content_t /var/www/html/WordPress/wp-content -R
   restorecon -Rv /var/www/html
   service httpd start
   ```

7. 동일한 디렉터리에서 다음 콘텐츠`stop_server.sh`로를 생성합니다.

```
#!/bin/bash
```

```
service httpd stop
```

8. zip 번들을 생성합니다.

   Linux:

   ```
   $ cd /tmp/WordPress
   $ zip -r wordpress.zip .
   ```

   Windows: "WordPress" 디렉터리로 이동하여 모든 파일을 선택하고 zip 파일을 생성합니다. 이름을 wordpress.zip으로 지정해야 합니다.

1. 애플리케이션 번들을 S3 버킷에 업로드합니다.

   스택을 계속 배포하려면 번들이 있어야 합니다.

   생성한 모든 S3 버킷 인스턴스에 자동으로 액세스할 수 있습니다. Bastion 또는 S3 콘솔을 통해 액세스하고 zip 파일을 drag-and-drop거나 탐색하여 선택하여 WordPress 번들을 업로드할 수 있습니다.

   쉘 창에서 다음 명령을 사용할 수도 있습니다. zip 파일의 경로가 올바른지 확인하세요.

   ```
   aws s3 cp wordpress.zip s3://BUCKET_NAME/
   ```

2. WordPress 애플리케이션 번들을 배포합니다.

   시작하기 전에 다음 데이터를 수집하면 배포 속도가 빨라집니다.

   필수 데이터:

   - `VPC-ID`:이 값은 S3 버킷의 위치를 결정합니다. 이전에 사용한 것과 동일한 VPC ID를 사용합니다.
   - `CodeDeployApplicationName` 및 `CodeDeployApplicationName`: HA 2-Tier 스택 RFC에서 사용한 `ApplicationName` 값은 CodeDeployApplicationName 및 CodeDeployDeploymentGroupName을 설정합니다. 이 예제에서는 "WordPress"를 사용하지만 다른 값을 사용했을 수 있습니다.
   - `S3Location`:의 경우 이전에 생성한 `BucketName`를 `S3Bucket`사용합니다. `S3BundleType` 및는 S3 스토어에 배치한 번들`S3Key`에서 가져온 것입니다.

a. CodeDeploy 애플리케이션의 실행 파라미터 JSON 스키마를 출력하여 DeployCDAppParams.json.

```
aws amscm get-change-type-version --change-type-id "ct-2edc3sd1sqmrb"
 --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
 DeployCDAppParams.json
```

b. 다음과 같이 스키마를 수정하고 다른 이름으로 저장하면 콘텐츠를 삭제하고 바꿀 수 있습니다.

```
{
"Description":                          "DeployWPCDApp",
"VpcId":                                "VPC_ID",
"Name":                                 "WordPressCDAppDeploy",
"TimeoutInMinutes":                     60,
"Parameters":   {
    "CodeDeployApplicationName":                "WordPress",
    "CodeDeployDeploymentGroupName":            "WordPress",
    "CodeDeployIgnoreApplicationStopFailures":   false,
    "CodeDeployRevision": {
      "RevisionType": "S3",
      "S3Location": {
        "S3Bucket":     "BUCKET_NAME",
        "S3BundleType": "zip",
        "S3Key":        "wordpress.zip" }
        }
    }
}
```

c. CreateRfc용 JSON 템플릿을 DeployCDAppRfc.json:

```
aws amscm create-rfc --generate-cli-skeleton > DeployCDAppRfc.json
```

d. DeployCDAppRfc.json 파일을 수정하고 저장하면 콘텐츠를 삭제하고 바꿀 수 있습니다. 이제 RequestedStartTime 및 RequestedEndTime는 선택 사항입니다. 제외하면 승인되는 즉시 실행되는 ASAP RFC가 생성됩니다(일반적으로 자동으로 발생함). 예약된 RFC를 제출하려면 해당 값을 추가합니다.

```
{
"ChangeTypeVersion":    "1.0",
```

```
"ChangeTypeId":         "ct-2edc3sd1sqmrb",
"Title":                "CD-Deploy-For-WP-RFC"
}
```

e. DeployCDAppRfc 파일과 DeployCDAppParams 실행 파라미터 파일을 지정하여 RFC를 생성합니다.

```
aws amscm create-rfc --cli-input-json file://DeployCDAppRfc.json  --execution-
parameters file://DeployCDAppParams.json
```

응답에서 새 RFC의 RfcId를 수신합니다. 후속 단계를 위해 ID를 저장합니다.

f. RFC 제출:

```
aws amscm submit-rfc --rfc-id RFC_ID
```

RFC가 성공하면 출력이 수신되지 않습니다.

g. RFC 상태를 확인하려면를 실행합니다.

```
aws amscm get-rfc --rfc-id RFC_ID
```

애플리케이션 배포 검증

WordPress 배포 경로 /WordPress를 사용하여 이전에 생성한 로드 밸런서의 엔드포인트(ELB CName)로 이동합니다. 예:

```
http://stack-ID-FOR-ELB.us-east-1.elb.amazonaws.com/WordPress
```

애플리케이션 배포 해체

자습서를 마치면 리소스에 대한 요금이 부과되지 않도록 배포를 해제해야 합니다.

다음은 일반 스택 삭제 작업입니다. HA 2-Tier 스택의 경우 한 번, S3 버킷 스택의 경우 한 번 두 번 제출하는 것이 좋습니다. 최종 후속 조치로 S3 버킷의 모든 스냅샷(서비스 요청에 S3 버킷 스택 ID 포함)을 삭제하라는 서비스 요청을 제출합니다. 10일 후에 자동으로 삭제되지만 조기에 삭제하면 약간의 비용이 절감됩니다.

이 연습에서는 AMS 콘솔을 사용하여 S3 스택을 삭제하는 예제를 제공합니다.이 절차는 AMS 콘솔을 사용하여 스택을 삭제하는 데 적용됩니다.

> **Note**
>
> S3 버킷을 삭제하는 경우 먼저 객체를 비워야 합니다.

필수 데이터:

- `StackId`: 사용할 스택입니다. 왼쪽 탐색 창의 링크를 통해 제공되는 AMS 콘솔 스택 페이지를 보면 이를 찾을 수 있습니다. AMS SKMS API/CLI를 사용하여 AMS SKMS API 참조의 경우 AWS 아티팩트 콘솔의 보고서 탭을 참조하세요(CLI의 `list-stack-summaries`).
- 이 연습의 변경 유형 ID는 이고`ct-0q0bic0ywqk6c`, 버전은 "1.0"이며, 최신 버전을 확인하려면 다음 명령을 실행합니다.

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=ct-0q0bic0ywqk6c
```

인라인 생성:

- 인라인으로 제공된 실행 파라미터를 사용하여 RFC 생성 명령을 실행합니다(실행 파라미터를 인라인으로 제공할 때 따옴표 이스케이프). E

```
aws amscm create-rfc --change-type-id "ct-0q0bic0ywqk6c" --change-type-version "1.0"
 --title "Delete My Stack" --execution-parameters "{\"StackId\":\"STACK_ID\"}"
```

- RFC 생성 작업에서 반환된 RFC ID를 사용하여 RFC를 제출합니다. 제출될 때까지 RFC는 `Editing` 상태를 유지하고 조치를 취하지 않습니다.

```
aws amscm submit-rfc --rfc-id RFC_ID
```

- RFC 상태를 모니터링하고 실행 출력을 봅니다.

```
aws amscm get-rfc --rfc-id RFC_ID
```

템플릿 생성:

1. RFC 템플릿을 현재 폴더의 파일로 출력합니다. 예제 이름은 DeleteStackRfc.json:입니다.

```
aws amscm create-rfc --generate-cli-skeleton > DeleteStackRfc.json
```

2. DeleteStackRfc.json 파일을 수정하고 저장합니다. 스택 삭제에는 실행 파라미터가 하나뿐이므로 실행 파라미터는 DeleteStackRfc.json 파일 자체에 있을 수 있습니다(실행 파라미터가 있는 별도의 JSON 파일을 생성할 필요가 없음).

   ExecutionParameters JSON 확장의 내부 따옴표는 백슬래시(\)로 이스케이프 처리해야 합니다. 시작 및 종료 시간이 없는 예:

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-0q0bic0ywqk6c",
"Title":                "Delete-My-Stack-RFC"
"ExecutionParameters":  "{
        \"StackId\":\"STACK_ID\"}"
}
```

3. RFC를 생성합니다.

```
aws amscm create-rfc --cli-input-json file://DeleteStackRfc.json
```

   응답에서 새 RFC의 RfcId를 수신합니다. 예:

```
{
"RfcId": "daaa1867-ffc5-1473-192a-842f6b326102"
}
```

   후속 단계를 위해 ID를 저장합니다.

4. RFC를 제출합니다.

```
aws amscm submit-rfc --rfc-id RFC_ID
```

   RFC가 성공하면 명령줄에서 확인 메시지가 표시되지 않습니다.

5. 요청 상태를 모니터링하고 실행 출력을 보려면:

```
aws amscm get-rfc --rfc-id RFC_ID --query "Rfc.
{Status:Status.Name,Exec:ExecutionOutput}" --output table
```

CLI 자습서: 티어 및 타이 WordPress 웹 사이트 배포

이 섹션에서는 AMS CLI를 사용하여 AMS 환경에 고가용성(HA) WordPress 사이트를 배포하는 방법을 설명합니다. 이 지침 세트에는 필요한 WordPress CodeDeploy 호환 패키지(예: zip) 파일을 생성하는 예제가 포함되어 있습니다.

> **Note**
>
> 이 배포 연습은 AMZN Linux 환경에서 사용하도록 설계되었습니다.
> 필수 변수 파라미터는 ## ### 것으로 표기되어 있지만 상황에 맞게 다른 파라미터를 수정하는 것이 좋습니다.

작업 및 필수 RFCs 요약:

1. 인프라를 생성합니다.
   a. RDS 스택 생성(CLI)
   b. 로드 밸런서 생성
   c. Auto Scaling 그룹을 생성하여 로드 밸런서에 연결합니다.
   d. CodeDeploy 애플리케이션용 S3 버킷 생성
2. WordPress 애플리케이션 번들 생성(RFC 필요 없음)
3. CodeDeploy를 사용하여 WordPress 애플리케이션 번들을 배포합니다.
   a. CodeDeploy 애플리케이션 생성
   b. CodeDeploy 배포 그룹 생성
   c. WordPress 애플리케이션 번들을 S3 버킷에 업로드(RFC 필요 없음)
   d. CodeDeploy 애플리케이션 배포
4. 배포 검증
5. 배포를 중단합니다.

인증된 계정에서 명령줄의 모든 단계를 따릅니다.

CLI를 사용하여 RFC 생성

RFC 생성에 대한 자세한 내용은 RFCs 생성을 참조하고, 일반적인 RFC 파라미터에 대한 설명은 RFC 일반 파라미터 단원을 참조하십시오.

인프라 생성

다음 절차에서는 리소스 IDs를 사용하여 인프라를 빌드하는 방식으로 RDS 데이터베이스, 로드 밸런서 및 Auto Scaling 그룹을 생성하는 방법을 설명합니다.

RDS 스택 생성(CLI)

RDS 스택 | 생성을 참조하세요.

ELB 스택 생성

퍼블릭 로드 밸런서(ELB)를 시작합니다. Load Balancer(ELB) 스택 | 생성을 참조하세요.

Auto Scaling 그룹 스택 생성

Auto Scaling 그룹을 시작합니다.

Auto Scaling 그룹 | 생성을 참조하세요.

S3 스토어 생성

S3 버킷을 시작합니다. S3 버킷은 생성한 애플리케이션 번들을 업로드하는 곳입니다. S3 스토리지 | 생성을 참조하세요.

CodeDeploy용 WordPress 애플리케이션 번들 생성

이 섹션에서는 애플리케이션 배포 번들을 생성하는 예제를 제공합니다.

1. WordPress를 다운로드하고 파일을 추출한 다음 ./scripts 디렉터리를 생성합니다.

   Linux 명령:

   ```
   wget https://github.com/WordPress/WordPress/archive/master.zip
   ```

   Windows: 브라우저 창에 붙여`https://github.com/WordPress/WordPress/archive/master.zip`넣고 zip 파일을 다운로드합니다.

   패키지를 어셈블할 임시 디렉터리를 생성합니다.

   Linux:

   ```
   mkdir /tmp/WordPress
   ```

Windows: "WordPress" 디렉터리를 생성합니다. 나중에 디렉터리 경로를 사용합니다.

2. WordPress 소스를 "WordPress" 디렉터리로 추출하고 ./scripts 디렉터리를 생성합니다.

   Linux:

```
unzip master.zip -d /tmp/WordPress_Temp
cp -paf /tmp/WordPress_Temp/WordPress-master/* /tmp/WordPress
rm -rf /tmp/WordPress_Temp
rm -f master
cd /tmp/WordPress
mkdir scripts
```

   Windows: 생성한 "WordPress" 디렉터리로 이동하여 여기에 "scripts" 디렉터리를 생성합니다.

   Windows 환경에 있는 경우 스크립트 파일의 브레이크 유형을 Unix(LF)로 설정해야 합니다. 메모장 ++에서 창 오른쪽 하단에 있는 옵션입니다.

3. WordPress 디렉터리에서 CodeDeploy appspec.yml 파일을 생성합니다(예제를 복사하는 경우 들여쓰기를 확인하고 각 공간을 계산합니다). 중요: WordPress 파일(이 경우 WordPress 디렉터리)을 예상 대상(/var/www/html/WordPress)으로 복사하기 위해 "소WordPress" 경로가 올바른지 확인합니다. 예제에서 appspec.yml 파일은 WordPress 파일이 있는 디렉터리에 있으므로 "/"만 있으면 됩니다. 또한 Auto Scaling 그룹에 RHEL AMI를 사용했더라도 "os: linux" 줄을 그대로 둡니다. appspec.yml 파일의 예:

```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/html/WordPress
hooks:
  BeforeInstall:
    - location: scripts/install_dependencies.sh
      timeout: 300
      runas: root
  AfterInstall:
    - location: scripts/config_wordpress.sh
      timeout: 300
      runas: root
  ApplicationStart:
    - location: scripts/start_server.sh
      timeout: 300
```

```
      runas: root
  ApplicationStop:
    - location: scripts/stop_server.sh
      timeout: 300
      runas: root
```

4. WordPress ./scripts 디렉터리에서 bash 파일 스크립트를 생성합니다.

   먼저 다음 콘텐츠`config_wordpress.sh`로를 생성합니다(원하는 경우 wp-config.php 파일을 직접 편집할 수 있음).

   > **Note**
   >
   > *DBName*을 HA 스택 RFC에 지정된 값으로 바꿉니다(예: `wordpress`).
   > *DB_MasterUsername*을 HA 스택 RFC에 지정된 `MasterUsername` 값으로 바꿉니다(예: `admin`).
   > *DB_MasterUserPassword*를 HA 스택 RFC에 지정된 `MasterUserPassword` 값으로 바꿉니다(예: `p4ssw0rd`).
   > HA 스택 RFC의 실행 출력에서 *DB_ENDPOINT*를 엔드포인트 DNS 이름으로 바꿉니다(예: `srt1cz23n45sfg.clgvd67uvydk.us-east-1.rds.amazonaws.com`). GetRfc 작업(CLI: get-rfc --rfc-id RFC_ID) 또는 이전에 제출한 HA 스택 RFC의 AMS 콘솔 RFC 세부 정보 페이지에서 이를 찾을 수 있습니다.

```
#!/bin/bash
chmod -R 755 /var/www/html/WordPress
cp /var/www/html/WordPress/wp-config-sample.php /var/www/html/WordPress/wp-
config.php
cd /var/www/html/WordPress
sed -i "s/database_name_here/DBName/g" wp-config.php
sed -i "s/username_here/DB_MasterUsername/g" wp-config.php
sed -i "s/password_here/DB_MasterUserPassword/g" wp-config.php
sed -i "s/localhost/DB_ENDPOINT/g" wp-config.php
```

5. 동일한 디렉터리에서 다음 콘텐츠`install_dependencies.sh`로를 생성합니다.

```
#!/bin/bash
yum install -y php
yum install -y php-mysql
yum install -y mysql
```

```
service httpd restart
```

> **Note**
>
> HTTPS는 상태 확인이 처음부터 작동하도록 시작 시 사용자 데이터의 일부로 설치됩니다.

6. 동일한 디렉터리에서 다음 콘텐츠`start_server.sh`로를 생성합니다.

   - Amazon Linux 인스턴스의 경우 다음을 사용합니다.

   ```
   #!/bin/bash
   service httpd start
   ```

   - RHEL 인스턴스의 경우 다음을 사용합니다(추가 명령은 SELINUX가 WordPress를 수락하도록 허용하는 정책임).

   ```
   #!/bin/bash
   setsebool -P  httpd_can_network_connect_db 1
   setsebool -P  httpd_can_network_connect 1
   chcon -t httpd_sys_rw_content_t /var/www/html/WordPress/wp-content -R
   restorecon -Rv /var/www/html
   service httpd start
   ```

7. 동일한 디렉터리에서 다음 콘텐츠`stop_server.sh`로를 생성합니다.

   ```
   #!/bin/bash
   service httpd stop
   ```

8. zip 번들을 생성합니다.

   Linux:

   ```
   $ cd /tmp/WordPress
   $ zip -r wordpress.zip .
   ```

   Windows: "WordPress" 디렉터리로 이동하여 모든 파일을 선택하고 zip 파일을 생성합니다. 이름을 wordpress.zip으로 지정해야 합니다.

CodeDeploy를 사용하여 WordPress 애플리케이션 번들 배포

CodeDeploy는 Amazon EC2 인스턴스에 대한 애플리케이션 배포를 자동화하는 AWS 배포 서비스입니다. 프로세스의이 부분에는 CodeDeploy 애플리케이션 생성, CodeDeploy 배포 그룹 생성, CodeDeploy를 사용하여 애플리케이션 배포가 포함됩니다.

CodeDeploy 애플리케이션 생성

CodeDeploy 애플리케이션은 AWS CodeDeploy에서 배포 중에 올바른 개정, 배포 구성 및 배포 그룹을 참조하는 데 사용하는 이름 또는 컨테이너입니다. 이 경우 배포 구성은 이전에 생성한 WordPress 번들입니다.

필수 데이터:

- `VpcId`: 사용 중인 VPC로, 이전에 사용한 VPC와 동일해야 합니다.
- `CodeDeployApplicationName`: 계정에서 고유해야 합니다. CodeDeploy 콘솔에서 기존 애플리케이션 이름을 확인합니다.
- `ChangeTypeId` 및 `ChangeTypeVersion`:이 연습의 변경 유형 ID는 입니다. 최신 버전을 `ct-0ah3gwb9seqk2`확인하려면 다음 명령을 실행합니다.

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=ct-0ah3gwb9seqk2
```

1. CodeDeploy 애플리케이션 CT의 실행 파라미터 JSON 스키마를 현재 폴더의 파일로 출력합니다. 예제 이름은 CreateCDAppParams.json.

```
aws amscm get-change-type-version --change-type-id "ct-0ah3gwb9seqk2" --query
 "ChangeTypeVersion.ExecutionInputSchema" --output text > CreateCDAppParams.json
```

2. 다음과 같이 JSON 파일을 수정하고 저장합니다. 콘텐츠를 삭제하고 바꿀 수 있습니다.

```
{
"Description":                    "Create WordPress CodeDeploy App",
"VpcId":                          "VPC_ID",
"StackTemplateId":                "stm-sft6rv00000000000",
"Name":                           "WordPressCDApp",
"TimeoutInMinutes":               60,
"Parameters":   {
    "CodeDeployApplicationName":    "WordPressCDApp"
```

```
    }
}
```

3. CreateRfc용 JSON 템플릿을 현재 폴더의 파일로 출력합니다. 예제 이름은 CreateCDAppRfc.json.

```
aws amscm create-rfc --generate-cli-skeleton > CreateCDAppRfc.json
```

4. 다음과 같이 JSON 파일을 수정하고 저장합니다. 콘텐츠를 삭제하고 바꿀 수 있습니다. 이제 `RequestedStartTime` 및 `RequestedEndTime`는 선택 사항입니다. 이를 제외하면 RFC가 승인되는 즉시(일반적으로 자동으로 발생함) RFC가 실행됩니다. “예약된” RFC를 제출하려면 해당 값을 추가합니다.

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-0ah3gwb9seqk2",
"Title":                "CD-App-For-WP-Stack-RFC"
}
```

5. CreateCDAppRfc 파일과 실행 파라미터 파일을 지정하여 RFC를 생성합니다.

```
aws amscm create-rfc --cli-input-json file://CreateCDAppRfc.json --execution-
parameters file://CreateCDAppParams.json
```

응답에서 새 RFC의 RFC ID를 수신합니다. 후속 단계를 위해 ID를 저장합니다.

6. RFC 제출:

```
aws amscm submit-rfc --rfc-id RFC_ID
```

RFC가 성공하면 출력이 수신되지 않습니다.

7. RFC 제출:

```
aws amscm get-rfc --rfc-id RFC_ID
```

CodeDeploy 배포 그룹 생성

CodeDeploy 배포 그룹을 생성합니다.

CodeDeploy 배포 그룹은 배포를 대상으로 하는 개별 인스턴스 세트를 정의합니다.

필수 데이터:

- `VpcId`: 사용 중인 VPC로, 이전에 사용한 VPC와 동일해야 합니다.
- `CodeDeployApplicationName`: 이전에 생성한 값을 사용합니다.
- `CodeDeployAutoScalingGroups`: 이전에 생성한 Auto Scaling 그룹의 이름을 사용합니다.
- `CodeDeployDeploymentGroupName`: 배포 그룹의 이름입니다. 이 이름은 배포 그룹과 연결된 각 애플리케이션에 대해 고유해야 합니다.
- `CodeDeployServiceRoleArn`: 예제에 제공된 공식을 사용합니다.
- `ChangeTypeId` 및 `ChangeTypeVersion`:이 연습의 변경 유형 ID는 입니다. 최신 버전을 `ct-2gd0u847qd9d2`확인하려면 다음 명령을 실행합니다.

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=ct-2gd0u847qd9d2
```

1. 실행 파라미터 JSON 스키마를 현재 폴더의 파일로 출력합니다. 예제 이름은 CreateCDDepGroupParams.json.

```
aws amscm get-change-type-version --change-type-id "ct-2gd0u847qd9d2"
 --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
 CreateCDDepGroupParams.json
```

2. 다음과 같이 JSON 파일을 수정하고 저장합니다. 콘텐츠를 삭제하고 바꿀 수 있습니다.

```
{
"Description":                      "CreateWPCDDeploymentGroup",
"VpcId":                            "VPC_ID",
"StackTemplateId":                  "stm-sp9lrk00000000000",
"Name":                             "WordPressCDAppGroup",
"TimeoutInMinutes":                 60,
"Parameters":   {
    "CodeDeployApplicationName":        "WordPressCDApp",
    "CodeDeployAutoScalingGroups":      ["ASG_NAME"],
    "CodeDeployDeploymentConfigName":   "CodeDeployDefault.HalfAtATime",
    "CodeDeployDeploymentGroupName":    "UNIQUE_CDDepGroupNAME",
    "CodeDeployServiceRoleArn":         "arn:aws:iam::ACCOUNT_ID:role/aws-
codedeploy-role"
    }
}
```

3. CreateRfc용 JSON 템플릿을 현재 폴더의 파일로 출력합니다. 예제 이름은 CreateCDDepGroupRfc.json.

```
aws amscm create-rfc --generate-cli-skeleton > CreateCDDepGroupRfc.json
```

4. 다음과 같이 JSON 파일을 수정하고 저장합니다. 콘텐츠를 삭제하고 바꿀 수 있습니다. 이제 RequestedStartTime 및 RequestedEndTime는 선택 사항입니다. 이를 제외하면 RFC가 승인되는 즉시(일반적으로 자동으로 발생함) RFC가 실행됩니다. "예약" RFC를 제출하려면 해당 값을 추가합니다.

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-2gd0u847qd9d2",
"Title":                "CD-Dep-Group-For-WP-Stack-RFC"
}
```

5. CreateCDDepGroupRfc 파일과 실행 파라미터 파일을 지정하여 RFC를 생성합니다.

```
aws amscm create-rfc --cli-input-json file://CreateCDDepGroupRfc.json --execution-
parameters file://CreateCDDepGroupParams.json
```

응답에서 새 RFC의 RFC ID를 수신합니다. 후속 단계를 위해 ID를 저장합니다.

6. RFC 제출:

```
aws amscm submit-rfc --rfc-id RFC_ID
```

RFC가 성공하면 출력이 수신되지 않습니다.

7. RFC 상태를 확인합니다.

```
aws amscm get-rfc --rfc-id RFC_ID
```

WordPress 애플리케이션 업로드

생성한 모든 S3 버킷 인스턴스에 자동으로 액세스할 수 있습니다. Bastions(인스턴스 액세스 참조) 또는 S3 콘솔을 통해 액세스하고 CodeDeploy 번들을 업로드할 수 있습니다. 스택을 계속 배포하려면 번들이 있어야 합니다. 이 예제에서는 이전에 생성한 버킷 이름을 사용합니다.

```
aws s3 cp wordpress/wordpress.zip s3://ACCOUNT_ID-codedeploy-bundles/
```

CodeDeploy를 사용하여 WordPress 애플리케이션 배포

CodeDeploy 애플리케이션을 배포합니다.

CodeDeploy 애플리케이션 번들 및 배포 그룹이 있으면이 RFC를 사용하여 애플리케이션을 배포합니다.

필수 데이터:

- `VPC-ID`: 사용 중인 VPC로, 이전에 사용한 VPC와 동일해야 합니다.
- `CodeDeployApplicationName`: 이전에 생성한 CodeDeploy 애플리케이션의 이름을 사용합니다.
- `CodeDeployDeploymentGroupName`: 이전에 생성한 CodeDeploy 배포 그룹의 이름을 사용합니다.
- `S3Location` (애플리케이션 번들을 업로드한 위치): `S3Bucket`: 이전에 생성한 BucketName `S3BundleType` 및 `S3Key`: S3 스토어에 배치한 번들의 유형 및 이름입니다.
- `ChangeTypeId` 및 `ChangeTypeVersion`:이 연습의 변경 유형 ID는 입니다. 최신 버전을 `ct-2edc3sd1sqmrb`확인하려면 다음 명령을 실행합니다.

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=ct-2edc3sd1sqmrb
```

1. CodeDeploy 애플리케이션 배포 CT의 실행 파라미터 JSON 스키마를 현재 폴더의 파일로 출력합니다. 예제 이름은 DeployCDAppParams.json.

```
aws amscm get-change-type-version --change-type-id "ct-2edc3sd1sqmrb" --query
 "ChangeTypeVersion.ExecutionInputSchema" --output text > DeployCDAppParams.json
```

2. 다음과 같이 JSON 파일을 수정합니다. 콘텐츠를 삭제하고 바꿀 수 있습니다. 의 경우 이전에 생성한 `BucketName`를 `S3Bucket`사용합니다.

```
{
"Description":                    "Deploy WordPress CodeDeploy Application",
"VpcId":                          "VPC_ID",
"Name":                           "WP CodeDeploy Deployment Group",
"TimeoutInMinutes":               60,
"Parameters":   {
```

```
    "CodeDeployApplicationName":        "WordPressCDApp",
    "CodeDeployDeploymentGroupName":    "WordPressCDDepGroup",
    "CodeDeployIgnoreApplicationStopFailures": false,
    "CodeDeployRevision": {
      "RevisionType": "S3",
      "S3Location": {
        "S3Bucket": "ACCOUNT_ID.BUCKET_NAME",
        "S3BundleType": "zip",
        "S3Key": "wordpress.zip" }
        }
    }
}
```

3. CreateRfc용 JSON 템플릿을 현재 폴더의 파일로 출력합니다. 예제 이름은 DeployCDAppRfc.json:.

```
aws amscm create-rfc --generate-cli-skeleton > DeployCDAppRfc.json
```

4. DeployCDAppRfc.json 파일을 수정하고 저장합니다. 콘텐츠를 삭제하고 바꿀 수 있습니다.

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-2edc3sd1sqmrb",
"Title":                "CD-Deploy-For-WP-Stack-RFC",
"RequestedStartTime":   "2017-04-28T22:45:00Z",
"RequestedEndTime":     "2017-04-28T22:45:00Z"
}
```

5. 실행 파라미터 파일과 DeployCDAppRfc 파일을 지정하여 RFC를 생성합니다.

```
aws amscm create-rfc --cli-input-json file://DeployCDAppRfc.json  --execution-
parameters file://DeployCDAppParams.json
```

응답에서 새 RFC의 RfcId를 수신합니다. 후속 단계를 위해 ID를 저장합니다.

6. RFC 제출:

```
aws amscm submit-rfc --rfc-id RFC_ID
```

RFC가 성공하면 출력이 수신되지 않습니다.

애플리케이션 배포 검증

WordPress 배포 경로 /WordPress를 사용하여 이전에 생성한 로드 밸런서의 엔드포인트(ELB CName)로 이동합니다. 예:

```
http://stack-ID-FOR-ELB.us-east-1.elb.amazonaws.com/WordPress
```

애플리케이션 배포 해체

배포를 해제하려면 RDS 데이터베이스 스택, 애플리케이션 로드 밸런서, Auto Scaling 그룹, S3 버킷, 모든의 코드 배포 애플리케이션 및 그룹 --6 RFCs에 대해 스택 삭제 CT를 제출합니다. 또한 삭제할 RDS 스냅샷에 대한 서비스 요청을 제출할 수 있습니다(10일 후에 자동으로 삭제되지만 그 동안 약간의 비용이 발생함). 모든에 대한 스택 IDs 수집한 다음 다음 단계를 따릅니다.

이 연습에서는 AMS 콘솔을 사용하여 S3 스택을 삭제하는 예제를 제공합니다.이 절차는 AMS 콘솔을 사용하여 스택을 삭제하는 데 적용됩니다.

> **Note**
>
> S3 버킷을 삭제하는 경우 먼저 객체를 비워야 합니다.

필수 데이터:

- `StackId`: 사용할 스택입니다. 왼쪽 탐색 창의 링크를 통해 제공되는 AMS 콘솔 스택 페이지를 보면 이를 찾을 수 있습니다. AMS SKMS API/CLI를 사용하여 AMS SKMS API 참조의 경우 AWS 아티팩트 콘솔의 보고서 탭을 참조하세요(CLI의 `list-stack-summaries`).
- 이 연습의 변경 유형 ID는 이고`ct-0q0bic0ywqk6c`, 버전은 "1.0"이며, 최신 버전을 확인하려면 다음 명령을 실행합니다.

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=ct-0q0bic0ywqk6c
```

인라인 생성:

- 인라인으로 제공된 실행 파라미터를 사용하여 RFC 생성 명령을 실행합니다(실행 파라미터를 인라인으로 제공할 때 따옴표 이스케이프). E

```
aws amscm create-rfc --change-type-id "ct-0q0bic0ywqk6c" --change-type-version "1.0"
 --title "Delete My Stack" --execution-parameters "{\"StackId\":\"STACK_ID\"}"
```

- RFC 생성 작업에서 반환된 RFC ID를 사용하여 RFC를 제출합니다. 제출될 때까지 RFC는 `Editing` 상태를 유지하고 조치를 취하지 않습니다.

```
aws amscm submit-rfc --rfc-id RFC_ID
```

- RFC 상태를 모니터링하고 실행 출력을 봅니다.

```
aws amscm get-rfc --rfc-id RFC_ID
```

템플릿 생성:

1. RFC 템플릿을 현재 폴더의 파일로 출력합니다. 예제 이름은 DeleteStackRfc.json:입니다.

   ```
   aws amscm create-rfc --generate-cli-skeleton > DeleteStackRfc.json
   ```

2. DeleteStackRfc.json 파일을 수정하고 저장합니다. 스택 삭제에는 실행 파라미터가 하나뿐이므로 실행 파라미터는 DeleteStackRfc.json 파일 자체에 있을 수 있습니다(실행 파라미터가 있는 별도의 JSON 파일을 생성할 필요가 없음).

   ExecutionParameters JSON 확장의 내부 따옴표는 백슬래시(\)로 이스케이프 처리해야 합니다. 시작 및 종료 시간이 없는 예:

   ```
   {
   "ChangeTypeVersion":    "1.0",
   "ChangeTypeId":         "ct-0q0bic0ywqk6c",
   "Title":                "Delete-My-Stack-RFC"
   "ExecutionParameters":  "{
           \"StackId\":\"STACK_ID\"}"
   }
   ```

3. RFC를 생성합니다.

   ```
   aws amscm create-rfc --cli-input-json file://DeleteStackRfc.json
   ```

   응답에서 새 RFC의 RfcId를 수신합니다. 예:

```
{
"RfcId": "daaa1867-ffc5-1473-192a-842f6b326102"
}
```

후속 단계를 위해 ID를 저장합니다.

4. RFC 제출:

```
aws amscm submit-rfc --rfc-id RFC_ID
```

RFC가 성공하면 명령줄에서 확인 메시지가 표시되지 않습니다.

5. 요청 상태를 모니터링하고 실행 출력을 보려면:

```
aws amscm get-rfc --rfc-id RFC_ID --query "Rfc.
{Status:Status.Name,Exec:ExecutionOutput}" --output table
```

애플리케이션 유지 관리

인프라가 배포되면 QA에서 스테이징, 프로덕션에 이르기까지 모든 AMS 환경에서 일관된 방식으로 인프라를 업데이트하는 것이 어렵습니다.

이 섹션에서는 AMS 워크로드 수집 프로세스에 대한 개요와 클라우드 인프라 계층을 최신 상태로 유지하는 데 사용할 수 있는 다양한 방법의 몇 가지 예를 제공합니다.

애플리케이션 유지 관리 전략

애플리케이션을 배포하는 방법은 애플리케이션을 유지 관리하는 방법에 영향을 미칩니다. 이 섹션에서는 애플리케이션 유지 관리를 위한 몇 가지 전략을 제공합니다.

환경 업데이트에는 다음과 같은 변경 사항이 포함될 수 있습니다.

- 보안 업데이트
- 애플리케이션의 새 버전
- 애플리케이션 구성 변경 사항
- 종속성 업데이트

> **Note**
>
> 모든 애플리케이션 배포의 경우 메서드에 관계없이 항상 서비스 요청을 미리 제출하여 AMS에 애플리케이션을 배포할 것임을 알립니다.

변경 불가능 및 변경 가능한 애플리케이션 설치 예제

컴퓨팅 인스턴스 이동성	앱 설치 방법	AMI
Mutable	CodeDeploy 사용	AMS 제공
	직접	
	Chef 또는 Puppet 사용, 풀 기반	
	Ansible 또는 Salt 사용, 푸시 기반	

컴퓨팅 인스턴스 이동성	앱 설치 방법	AMI
변경 불가능	Golden AMI 사용	사용자 지정 (AMS 제공 기준)

CodeDeploy 지원 AMI를 사용한 변경 가능한 배포

AWS CodeDeploy는 Amazon EC2 인스턴스 및 온프레미스에서 실행되는 인스턴스를 포함하여 모든 인스턴스에 대한 코드 배포를 자동화하는 서비스입니다. CodeDeploy를 AMS와 함께 사용하여 CodeDeploy 애플리케이션을 생성하고 배포할 수 있습니다. AMS는 CodeDeploy 애플리케이션에 대한 기본 인스턴스 프로파일을 제공합니다.

- Amazon Linux(버전 1)
- Amazon Linux 2
- RedHat 7
- CentOS 7

CodeDeploy를 처음 사용하기 전에 여러 설정 단계를 완료해야 합니다.

1. AWS CLI 설치 또는 업그레이드
2. AWS CodeDeploy에 대한 서비스 역할 생성, 배포에서 서비스 역할 ARN 사용

모든 CT 옵션IDs는 변경 유형 참조에서 찾을 수 있습니다.

> **Note**
>
> 현재이 솔루션과 함께 Amazon S3 스토리지를 사용해야 합니다.

기본 단계는 여기에 요약되어 있으며 절차는 AMS 사용 설명서에 자세히 설명되어 있습니다.

1. Amazon S3 스토리지 버킷을 생성합니다. CT: ct-1a68ck03fn98r. S3 버킷에는 버전 관리가 활성화되어 있어야 합니다(이 작업에 대한 자세한 내용은 버킷 버전 관리 활성화 참조).
2. 번들 CodeDeploy 아티팩트를 여기에 넣습니다. AMS를 통한 액세스 요청 없이 Amazon S3 콘솔에서이 작업을 수행할 수 있습니다. 또는이 명령의 변형을 사용합니다.

```
aws s3 cp ZIP_FILEPATH_AND_NAME s3://S3BUCKET_NAME/
```

3. AMS `customer-` AMI를 찾아 다음 중 하나를 사용합니다.

   - AMS 콘솔: 관련 VPC의 VPC 세부 정보 페이지
   - AMS API AMS SKMS API 참조는 AWS Artifact 콘솔의 보고서 탭을 참조하세요. 또는 CLI: `aws amsskms list-amis`

4. Autoscaling 그룹(ASG)을 생성합니다. CT: ct-2tylseo8rxfsc. AMS AMI를 지정하고, 로드 밸런서가 열린 포트를 갖도록 설정하고, `customer-mc-ec2-instance-profile`에를 지정합니다`ASGIAMInstanceProfile`.
5. CodeDeploy 애플리케이션을 생성합니다. CT: ct-0ah3gwb9seqk2. 파라미터에는 애플리케이션 이름이 포함됩니다. 예: `WordpressProd`.
6. CodeDeploy 배포 그룹을 생성합니다. CT: ct-2gd0u847qd9d2. 파라미터에는 CodeDeploy 애플리케이션 이름, ASG 이름, 구성 유형 이름 및 서비스 역할 ARN이 포함됩니다.
7. CodeDeploy 애플리케이션을 배포합니다. CT: ct-2edc3sd1™rb. 파라미터에는 CodeDeploy 애플리케이션 이름, 구성 유형 이름, 배포 그룹 이름, 개정 유형 및 CodeDeploy 아티팩트가 있는 S3 버킷 위치가 포함됩니다.

변경 가능한 배포, 수동으로 구성 및 업데이트된 애플리케이션 인스턴스

이 애플리케이션 배포 전략은 애플리케이션 인스턴스를 간단하고 수동으로 업데이트하는 것입니다. 다음은 기본 단계입니다.

모든 CT 옵션IDs는 변경 유형 참조에서 찾을 수 있습니다.

> **Note**
>
> 현재이 솔루션과 함께 Amazon S3 스토리지를 사용해야 합니다.

기본 단계는 여기에 요약되어 있습니다. 다양한 절차는 AMS 사용 설명서에 자세히 설명되어 있습니다.

1. Amazon S3 스토리지 버킷을 생성합니다. CT: ct-1a68ck03fn98r. S3 버킷에는 버전 관리가 활성화되어 있어야 합니다(이 작업에 대한 자세한 내용은 버킷 버전 관리 활성화 참조).

2. 번들링된 애플리케이션 아티팩트를 여기에 넣습니다(애플리케이션이 부팅 및 작동 시 시작해야 하는 모든 것). AMS를 통한 액세스 요청 없이 Amazon S3 콘솔에서이 작업을 수행할 수 있습니다. 또는이 명령의 변형을 사용합니다.

```
aws s3 cp ZIP_FILEPATH_AND_NAME s3://S3BUCKET_NAME/
```

3. AMS AMI를 찾으면 모든 AMI에 CodeDeploy가 추가됩니다. "customer-" AMI를 찾으려면 다음 중 하나를 사용합니다.

   - AMS 콘솔: 관련 VPC의 VPC 세부 정보 페이지
   - AMS API AMS SKMS API 참조는 AWS Artifact 콘솔의 보고서 탭을 참조하세요. 또는 CLI: `aws amsskms list-amis`

4. 해당 AMI를 사용하여 EC2 인스턴스를 생성합니다. CT: ct-14027q0sjyt1h. AMS AMI를 지정`Key=backup, Value=true`하고 태그를 설정한 다음 `InstanceProfile` 파라미터에 `customer-mc-ec2-instance-profile`를 지정합니다. 반환되는 인스턴스 ID를 기록해 둡니다.

5. 인스턴스에 대한 관리자 액세스를 요청합니다. CT: ct-1dmlg9g1l91h6. 계정에 대한 FQDN이 필요합니다. FQDN이 무엇인지 잘 모르는 경우 다음을 통해 찾을 수 있습니다.

   - AWS Management Console for Directory Services(보안 및 자격 증명 아래) 디렉터리 이름 탭을 사용합니다.
   - 다음 명령 중 하나 실행(디렉터리 클래스 반환, DC+DC+DC=FQDN): Windows: `whoami / fqdn` 또는 Linux: `hostname --fqdn`.

6. 인스턴스에 로그인하려면 AMS 사용 설명서의 Bastions를 통한 인스턴스 액세스를 참조하세요.
7. 번들링된 애플리케이션 파일을 S3 버킷에서 인스턴스로 다운로드합니다.
8. AMS에 대한 서비스 요청과 함께 즉시 백업을 요청하려면 인스턴스 ID를 알아야 합니다.
9. 애플리케이션을 업데이트해야 하는 경우 새 파일을 S3 버킷에 로드한 다음 3~8단계를 따릅니다.

풀 기반 배포 도구 구성 AMI를 사용한 변경 가능한 배포

이 전략은 관리형 서비스 EC2 CT 생성의 `InstanceUserData` 파라미터에 의존합니다. 이 파라미터 사용에 대한 자세한 내용은 사용자 데이터로 인스턴스 구성을 참조하세요. 이 예제에서는 Chef 또는 Puppet과 같은 풀 기반 애플리케이션 배포 도구를 가정합니다.

CodeDeploy 에이전트는 모든 AMS AMIs. 지원되는 AMIs

- Amazon Linux(버전 1)
- Amazon Linux 2
- RedHat 7
- CentOS 7

모든 CT 옵션IDs는 변경 유형 참조에서 찾을 수 있습니다.

> **Note**
>
> 현재이 솔루션과 함께 Amazon S3 스토리지를 사용해야 합니다.

기본 단계는 여기에 요약되어 있으며 절차는 AMS 사용 설명서에 자세히 설명되어 있습니다.

1. Amazon S3 스토리지 버킷을 생성합니다. CT: ct-1a68ck03fn98r. S3 버킷에는 버전 관리가 활성화되어 있어야 합니다(이 작업에 대한 자세한 내용은 버킷 버전 관리 활성화 참조).
2. 번들 CodeDeploy 아티팩트를 여기에 넣습니다. AMS를 통한 액세스 요청 없이 Amazon S3 콘솔에서이 작업을 수행할 수 있습니다. 또는이 명령의 변형을 사용합니다.

   ```
   aws s3 cp ZIP_FILEPATH_AND_NAME s3://S3BUCKET_NAME/
   ```

3. AMS `customer-` AMI를 찾아 다음 중 하나를 사용합니다.
   - AMS 콘솔: 관련 VPC의 VPC 세부 정보 페이지
   - AMS API AMS SKMS API 참조는 AWS Artifact 콘솔의 보고서 탭을 참조하세요. 또는 CLI: `aws amsskms list-amis`
4. EC2 인스턴스 생성합니다. CT: ct-14027q0sjyt1h, 태그를 설정하고 `InstanceUserData` 파라미터를 `Key=backup, Value=true`사용하여 부트스트랩 및 기타 스크립트(Chef/Puppet 에이전트 다운로드 등)를 지정하고 필요한 권한 부여 키를 포함합니다. AMS 사용 설명서의 HA 2계층 배포 생성에 대한 관리 변경 섹션 예제에서이 작업을 수행하는 예제를 찾을 수 있습니다. 또는 인스턴스에 대한 액세스를 요청하고 로그인한 다음 필요한 배포 아티팩트로 구성합니다. 풀 기반 배포 명령은 인스턴스의 에이전트에서 회사 마스터 서버로 이동하며 접속을 통과하려면 권한이 필요할 수 있습니다. 접속 없이 보안 그룹/AD 그룹 액세스를 요청하려면 AMS에 대한 서비스 요청이 필요할 수 있습니다.
5. 4단계를 반복하여 다른 EC2 인스턴스를 생성하고 배포 도구 마스터 서버로 구성합니다.

6. 애플리케이션을 업데이트해야 하는 경우 배포 도구를 사용하여 인스턴스에 업데이트를 롤아웃합니다.

푸시 기반 배포 도구 구성 AMI를 사용한 변경 가능한 배포

이 전략은 관리형 서비스 EC2 CT 생성의 `InstanceUserData` 파라미터에 의존합니다. 이 파라미터 사용에 대한 자세한 내용은 사용자 데이터로 인스턴스 구성을 참조하세요. 이 예제에서는 Chef 또는 Puppet과 같은 풀 기반 애플리케이션 배포 도구를 가정합니다.

모든 CT 옵션IDs는 변경 유형 참조에서 찾을 수 있습니다.

> **Note**
>
> 현재이 솔루션과 함께 Amazon S3 스토리지를 사용해야 합니다.

기본 단계는 여기에 요약되어 있으며 절차는 AMS 사용 설명서에 자세히 설명되어 있습니다.

1. Amazon S3 스토리지 버킷을 생성합니다. CT: ct-1a68ck03fn98r. S3 버킷에는 버전 관리가 활성화되어 있어야 합니다(이 작업에 대한 자세한 내용은 버킷 버전 관리 활성화 참조).
2. 번들 CodeDeploy 아티팩트를 여기에 넣습니다. AMS를 통한 액세스 요청 없이 Amazon S3 콘솔에서이 작업을 수행할 수 있습니다. 또는이 명령의 변형을 사용합니다.

   ```
   aws s3 cp ZIP_FILEPATH_AND_NAME s3://S3BUCKET_NAME/
   ```

3. AMS AMI를 찾으면 모든 AMI에 CodeDeploy가 추가됩니다. "customer-" AMI를 찾으려면 다음 중 하나를 사용합니다.

   - AMS 콘솔: 관련 VPC의 VPC 세부 정보 페이지
   - AMS API AMS SKMS API 참조는 AWS Artifact 콘솔의 보고서 탭을 참조하세요. 또는 CLI: `aws amsskms list-amis`

4. EC2 인스턴스 생성합니다. CT: ct-14027q0sjyt1h, 태그를 설정하고 `InstanceUserData` 파라미터를 사용하여 권한 부여 키`Key=backup, Value=true`, SALT 스택(미니온 부트스트랩 - 자세한 내용은 Cloud-Init을 사용하여 Linux EC2에서 솔트 부트스트래핑 참조) 또는 Ansible(키 페어 설치 - 자세한 내용은 Ansible 및 Dynamic Amazon EC2 인벤토리 관리 시작하기 참조)을 포함한 부트스트랩 및 기타 스크립트를 실행합니다. 또는 인스턴스에 대한 액세스를 요청하고 인스턴스에 로그인한 다음 필요한 배포 아티팩트로 구성합니다. 푸시 기반 명령은 회사 서브넷에서 인스턴스

로 전송되며 Bastion을 통과하도록 권한 부여를 구성해야 할 수 있습니다. 접속 없이 보안 그룹/AD 그룹 액세스를 요청하려면 AMS에 대한 서비스 요청이 필요할 수 있습니다.

5. 4단계를 반복하여 다른 EC2 인스턴스를 생성하고 배포 도구 마스터 서버로 구성합니다.
6. 애플리케이션을 업데이트해야 하는 경우 배포 도구를 사용하여 인스턴스에 업데이트를 롤아웃합니다.

골든 AMI를 사용한 변경 불가능한 배포

이 전략은 모든 애플리케이션 인스턴스가 원하는 대로 작동하도록 구성한 "골드" AMI를 사용합니다. 예를 들어이 골든 AMI로 생성된 인스턴스는 올바른 도메인 및 DNS에 자체 조인하고, 자체 구성하고, 재부팅하고, 필요한 모든 시스템을 시작합니다. 애플리케이션 인스턴스를 업데이트하려는 경우 골든 AMI를 다시 생성하고 이를 사용하여 완전히 새로운 애플리케이션 인스턴스를 롤아웃합니다.

CodeDeploy 에이전트는 모든 AMS AMIs. 지원되는 AMIs

- Amazon Linux(버전 1)
- Amazon Linux 2
- RedHat 7
- CentOS 7

모든 CT 옵션IDs는 변경 유형 참조에서 찾을 수 있습니다.

> **Note**
>
> 현재이 솔루션과 함께 Amazon S3 스토리지를 사용해야 합니다.

1. Amazon S3 스토리지 버킷을 생성합니다. CT: ct-1a68ck03fn98r. S3 버킷에는 버전 관리가 활성화되어 있어야 합니다(이 작업에 대한 자세한 내용은 버킷 버전 관리 활성화 참조).
2. 번들링된 애플리케이션 아티팩트를 여기에 넣습니다(애플리케이션이 부팅 및 작동 시 시작해야 하는 모든 것). AMS를 통한 액세스 요청 없이 Amazon S3 콘솔에서이 작업을 수행할 수 있습니다. 또는이 명령의 변형을 사용합니다.

   ```
   aws s3 cp ZIP_FILEPATH_AND_NAME s3://S3BUCKET_NAME/
   ```

3. AMS `customer-` AMI를 찾아 다음 중 하나를 사용합니다.

- AMS 콘솔: 관련 VPC의 VPC 세부 정보 페이지
- AMS API AMS SKMS API 참조는 AWS Artifact 콘솔의 보고서 탭을 참조하세요. 또는 CLI: `aws amsskms list-amis`

4. 해당 AMI를 사용하여 EC2 인스턴스를 생성합니다. CT: ct-14027q0sjyt1h. AMS AMI를 지정`Key=backup, Value=true`하고 태그를 설정한 다음 `customer-mc-ec2-instance-profile`에를 지정합니다`InstanceProfile`. 반환되는 인스턴스 ID를 기록해 둡니다.
5. 인스턴스에 대한 관리자 액세스를 요청합니다. CT: ct-1dmlg9g1l91h6. 계정에 대한 FQDN이 필요합니다. FQDN이 무엇인지 잘 모르는 경우 다음을 통해 찾을 수 있습니다.
   - AWS Management Console for Directory Services(보안 및 자격 증명 아래) 디렉터리 이름 탭 사용.
   - 다음 명령 중 하나 실행(Return 디렉터리 클래스, DC+DC+DC=FQDN): Windows: `whoami /fqdn` 또는 Linux: `hostname --fqdn`.
6. 인스턴스에 로그인하려면 AMS 사용 설명서의 인스턴스 액세스를 참조하세요.
7. S3 버킷에서 번들링된 애플리케이션 파일을 인스턴스로 다운로드합니다. 부팅 시 완전히 작동하는 애플리케이션을 자체 배포하도록 인스턴스를 구성합니다.
8. 인스턴스에서 골든 AMI를 생성합니다. CT: ct-3rqqu43krekby. 자세한 내용은 AMI | 생성을 참조하세요.
9. 해당 AMI를 사용하여 새 인스턴스를 생성하도록 Auto Scaling 그룹을 구성합니다. CT: ct-2tylseo8rxfsc. 애플리케이션을 업데이트해야 하는 경우이 절차에 따라 AMS에 새 골든 AMI를 사용하도록 ASG를 업데이트하도록 요청하고 이를 위해 Management | Other | Other | Update CT를 사용합니다.

전략 업데이트

AMS 관리형 환경에서 애플리케이션 또는 인스턴스를 업데이트하는 데 사용할 수 있는 몇 가지 전략이 있습니다.

- 예약된 가동 중지:이 간단한 전략에는 애플리케이션이 오프라인 상태가 되고 수동으로 업데이트되는 시간을 예약하는 작업이 포함됩니다. 이렇게 하려면 Management | Other | Other | Update CT (ct-0xdawir96cy7k) 요청을 제출하여 필요한 인스턴스를 중지합니다. 필요한 업데이트를 수행한 다음 다른 관리 | 기타 | 기타 | CT 업데이트(ct-0xdawir96cy7k) 요청을 제출하여 인스턴스를 시작합니다.

- 블루/그린:이 전략을 사용하려면 중복 환경(완전히 작동하는 두 환경)이 있어야 하며 도메인 이름 시스템(DNS) 또는 웹 방화벽(WAF) 업데이트를 사용하여 하나의 환경을 오프라인으로 전환하여 트래픽을 리디렉션해야 합니다. 한 환경을 업데이트한 다음 다시 리디렉션하여 다른 환경을 업데이트합니다.

  자세한 내용은 AWS CodeDeploy의 블루/그린 배포 소개를 참조하세요.
- 새 AMI로 롤링 업데이트: 새 AMI를 사용자 지정한 다음(AMI 생성 참조) AMS가 Auto Scaling 그룹에 배포하도록 요청하는 위치입니다. 관리 | 기타 | 기타 | CT 업데이트(ct-0xdawir96cy7k)를 사용하여이 작업을 수행합니다.

AWS Managed Services 리소스 스케줄러

AWS Managed Services(AMS) Resource Scheduler를 사용하여 계정에서 AutoScaling 그룹, Amazon EC2 인스턴스 및 RDS 인스턴스의 자동 시작 및 중지를 예약합니다. 이렇게 하면 리소스가 연중무휴로 실행되지 않는 인프라 비용을 줄일 수 있습니다. 이 솔루션은 의 인스턴스 스케줄러 AWS를 기반으로 구축되지만 AMS 요구 사항에 맞는 추가 기능 및 사용자 지정이 포함되어 있습니다.

> **Note**
>
> 기본적으로 AMS Resource Scheduler는 AWS CloudFormation 스택의 일부가 아닌 리소스와 상호 작용하지 않습니다. 리소스는 "stack-", "sc-" 또는 "SC-"로 시작하는 스택의 일부여야 합니다. CloudFormation 스택에 포함되지 않은 리소스를 예약하려면 Resource Scheduler 스택 파라미터를 `ScheduleNonStackResources`로 업데이트하면 됩니다`Yes`.

AMS Resource Scheduler는 기간과 일정을 사용합니다.

- 기간은 시작 시간, 종료 시간, 월의 일 등 Resource Scheduler가 실행되는 시간을 정의합니다.
- 일정에는 SSM 유지 관리 기간, 시간대, 최대 절전 모드 설정 등과 같은 추가 구성과 함께 정의된 기간이 포함되며, 구성된 기간 규칙을 고려하여 리소스를 실행해야 하는 시기를 지정합니다.

AMS Resource Scheduler의 자동 변경 유형(CTs.

AMS Resource Scheduler에 사용할 수 있는 설정에 대한 자세한 내용은 솔루션 구성 요소의 해당 AWS 인스턴스 스케줄러 설명서를 참조하세요. 솔루션의 아키텍처 보기는 Architecture overview.html의 해당 AWS 인스턴스 스케줄러 설명서를 참조하세요.

AMS Resource Scheduler 배포

AMS Resource Scheduler를 배포하려면 자동 변경 유형(CT) : Deployment | AMS Resource Scheduler | Solution | Deploy(ct-0ywnhc8e5k9z5)를 사용하여 계정에 솔루션을 배포하는 RFC를 생성합니다. RFC가 실행되면 기본 구성의 AMS Resource Scheduler 리소스가 포함된 CloudFormation 스택이 계정에 자동으로 프로비저닝됩니다. Resource Scheduler 변경 유형에 대한 자세한 내용은 AMS Resource Scheduler를 참조하세요.

> **Note**
>
> AMS Resource Scheduler가 계정에 이미 배포되어 있는지 확인하려면 AWS Lambda 콘솔에서 해당 계정을 확인하고 AMSResourceScheduler 함수를 찾습니다.

계정에 AMS Resource Scheduler를 프로비저닝한 후에는 기본 구성을 검토하고 필요한 경우 기본 설정에 따라 태그 키, 시간대, 예약된 서비스 등과 같은 구성을 사용자 지정하는 것이 좋습니다. 권장 사용자 지정에 대한 자세한 내용은 AMS Resource Scheduler 사용자 지정다음 단원을 참조하십시오.

사용자 지정 구성을 만들거나 Resource Scheduler 구성을 확인하려면

AMS Resource Scheduler 사용자 지정

업데이트 AMS Resource Scheduler 변경 유형을 사용하여 AMS Resource Scheduler의 다음 속성을 사용자 지정하는 것이 좋습니다. AMS Resource Scheduler를 참조하세요.

- 태그 이름: Resource Scheduler가 인스턴스 일정을 리소스와 연결하는 데 사용할 태그의 이름입니다. 기본값은 일정입니다.
- 예약된 서비스: Resource Scheduler가 관리할 수 있는 쉼표로 구분된 서비스 목록입니다. 기본값은 "ec2,rds,autoscaling"입니다. 유효한 값은 "ec2", "rds" 및 "autoscaling"입니다.
- 기본 시간대: Resource Scheduler에서 사용할 기본 시간대를 지정합니다. 기본값은 UTC입니다.
- CMK 사용: Resource Scheduler에 권한을 부여할 수 있는 Amazon KMS 고객 관리형 키(CMK) ARNs의 쉼표로 구분된 목록입니다.
- LicenseManager 사용: Resource Scheduler에 대한 AWS Licence Manager ARNs 목록을 쉼표로 구분하여 권한을 부여할 수 있습니다.

> **Note**
>
> AMS는 AMS Resource Scheduler를 계정의 최신 상태로 유지하기 위해 기능 및 수정 사항을 릴리스할 수 있습니다. 이 경우 AMS Resource Scheduler에 대한 사용자 지정은 유지됩니다.

AMS Resource Scheduler 사용

솔루션을 배포한 후 AMS Resource Scheduler를 구성하려면 자동화된 Resource Scheduler CTs를 사용하여 AMS Resource Scheduler 기간(리소스 스케줄러가 실행되는 시간) 및 일정(구성된 기간 및 기타 옵션)을 생성, 삭제, 업데이트 및 설명(상세 정보 가져오기)합니다. AMS Resource Scheduler 변경 유형을 사용하는 예는 AMS Resource Scheduler를 참조하세요.

AMS Resource Scheduler에서 관리할 리소스를 선택하려면 배포 및 일정 생성 후 AMS 태그 CTs 생성을 사용하여 배포 중에 제공한 태그 키로 Auto Scaling 그룹, Amazon RDS 스택 및 Amazon EC2 리소스에 태그를 지정하고 정의된 일정을 태그 값으로 지정합니다. 리소스에 태그를 지정하면 정의된 Resource Scheduler 일정에 따라 리소스가 시작 또는 중지되도록 예약됩니다.

AMS Resource Scheduler를 사용하는 데 드는 추가 비용은 없습니다. 그러나 솔루션은 여러를 사용 AWS 서비스 하며 이러한 리소스가 사용될 때 요금이 부과됩니다. 자세한 내용은 아키텍처 개요를 참조하세요.

AMS Resource Scheduler를 옵트아웃하려면

- 임시 옵트아웃 또는 비활성화의 경우: 자동 관리 | AMS 리소스 스케줄러 | 상태 | 변경 유형 비활성화(ct-14v49adibs4db)를 사용하여 RFC 제출
- 영구 제거의 경우: 관리 제출 | 기타 | 기타 | 업데이트(검토 필요)(ct-0xdawir96cy7k) Resource Scheduler 릴리스 자동화 시스템에서 제거를 요청하는 RFC

AMS Resource Scheduler 비용 예측기

비용 절감을 추적하기 위해 AMS Resource Scheduler는 스케줄러에서 관리하는 Amazon EC2 및 RDS 리소스의 예상 비용 절감을 시간당 계산하는 구성 요소를 제공합니다. 그러면이 비용 절감 데이터가 CloudWatch 지표(`AMS/ResourceScheduler`)로 게시되어 이를 추적하는 데 도움이 됩니다. 비용 절감 예측기는 인스턴스 실행 시간에 대한 절감액만 추정합니다. 리소스와 관련된 데이터 전송 비용과 같은 다른 비용은 고려하지 않습니다.

비용 절감 예측기는 Resource Scheduler에서 활성화됩니다. 시간당 실행되며에서 비용 및 사용량 데이터를 검색합니다 AWS Cost Explorer. 해당 데이터에서 각 인스턴스 유형에 대한 시간당 평균 비용을

계산한 다음 예약 없이 실행되는 경우 하루 종일 비용을 프로젝션합니다. 비용 절감은 예상 비용과 지정된 날짜 동안 Cost Explorer에서 실제로 보고된 비용 간의 차이입니다.

예를 들어 인스턴스 A가 오전 9시부터 오후 5시까지 실행되도록 Resource Scheduler로 구성된 경우 해당 날짜의 8시간입니다. Cost Explorer는 비용을 1 USD로, 사용량을 8로 보고합니다. 따라서 시간당 평균 비용은 0.125 USD입니다. 인스턴스가 Resource Scheduler로 예약되지 않은 경우 인스턴스는 해당 날짜에 24시간 실행됩니다. 이 경우 비용은 24x0.125 = 3 USD였습니다. Resource Scheduler를 사용하면 2 USD의 비용 절감을 달성할 수 있습니다.

비용 절감 예측기가 Cost Explorer에서 Resource Scheduler에서 관리하는 리소스에 대해서만 비용 및 사용량을 검색하려면 Resource Scheduler가 리소스를 대상으로 지정하는 데 사용하는 태그 키를 결제 대시보드에서 비용 할당 태그로 활성화해야 합니다. 계정이 조직에 속한 경우 조직의 관리 계정에서 태그 키를 활성화해야 합니다. 이에 대한 자세한 내용은 사용자 정의 비용 할당 태그 및 사용자 정의 비용 할당 태그 활성화를 참조하세요.

태그 키가 비용 할당 태그로 활성화되면 AWS 청구는 Resource Scheduler에서 관리하는 리소스의 비용 및 사용량을 추적하기 시작하고 해당 데이터를 사용할 수 있게 되면 비용 절감 예측기는 비용 절감을 계산하고 CloudWatch의 `AMS/ResourceScheduler` 지표 네임스페이스에 데이터를 게시하기 시작합니다.

비용 예측기 팁

Cost Savings Estimator는 계산 시 예약 인스턴스, 절감형 플랜 등과 같은 할인을 허용하지 않습니다. 예측기는 Cost Explorer에서 사용 비용을 받아 리소스의 시간당 평균 비용을 계산합니다. 자세한 내용은 AWS 비용 데이터세트 이해: 치트 시트를 참조하세요.

비용 절감 예측기가 Cost Explorer에서 Resource Scheduler에서 관리하는 리소스에 대해서만 비용 및 사용량을 검색하려면 Resource Scheduler가 리소스를 대상으로 지정하는 데 사용하는 태그 키를 결제 대시보드에서 비용 할당 태그로 활성화해야 합니다. 계정이 조직에 속한 경우 조직의 관리 계정에서 태그 키를 활성화해야 합니다. 이에 대한 자세한 내용은 사용자 정의 비용 할당 태그를 참조하세요. 비용 할당 태그가 활성화되지 않은 경우 예측기는 절감액을 계산하고 활성화된 경우에도 지표를 게시할 수 없습니다.

AMS Resource Scheduler 모범 사례

Amazon EC2 인스턴스 예약

- 인스턴스 종료 동작은가 `stop` 아닌 로 설정해야 합니다`terminate`. 이는 AMS Amazon EC2 자동 변경 유형 생성(ct-14027q0sjyt1h)으로 생성된 인스턴스의 `stop` 경우 로 사전 설정되며

속성을 AWS CloudFormation 로 설정하여 수집으로 생성된 Amazon EC2 인스턴스의 경우 `InstanceInitiatedShutdownBehavior`로 설정할 수 있습니다`stop`. 인스턴스 종료 동작이로 설정된 경우 Resource Scheduler가 인스턴스를 중지하고 스케줄러가 다시 시작할 수 없을 때 `terminate`인스턴스가 종료됩니다.

- Auto Scaling 그룹의 일부인 Amazon EC2 인스턴스는 태그가 지정된 경우에도 AMS Resource Scheduler에서 개별적으로 처리되지 않습니다.
- 대상 인스턴스 루트 볼륨이 KMS 고객 마스터 키(CMK)로 암호화된 경우 스케줄러가 이러한 인스턴스를 시작할 수 있도록 Resource Scheduler IAM 역할에 추가 `kms:CreateGrant` 권한을 추가해야 합니다. 보안 향상을 위해이 권한은 기본적으로 역할에 추가되지 않습니다. 이 권한이 필요한 경우 Management | AMS Resource Scheduler | Solution | Update change type을 사용하여 RFC를 제출하고 KMS CMKs의 쉼표로 구분된 ARNs 목록을 지정합니다.

Auto Scaling 그룹 예약

- AMS Resource Scheduler는 오토 스케일링의 개별 인스턴스가 아닌 Auto Scaling의 오토 스케일링을 시작하거나 중지합니다. 즉, 스케줄러는 Auto Scaling 그룹의 크기를 복원하거나(시작) 크기를 0(중지)으로 설정합니다.
- AutoScaling 그룹에 그룹 내 인스턴스가 아닌 지정된 태그로 태그를 지정합니다.
- 중지하는 동안 AMS Resource Scheduler는 Auto Scaling 그룹의 최소, 원하는 및 최대 용량 값을 저장하고 최소 및 원하는 용량을 0으로 설정합니다. 시작하는 동안 스케줄러는 Auto Scaling 그룹 크기를 중지하는 동안과 동일하게 복원합니다. 따라서 Auto Scaling 그룹 인스턴스는 인스턴스의 종료 및 다시 시작이 Auto Scaling 그룹에서 실행되는 애플리케이션에 영향을 주지 않도록 적절한 용량 구성을 사용해야 합니다.
- 실행 기간 동안 Auto Scaling 그룹이 수정된 경우(최소 또는 최대 용량) 스케줄러는 새 Auto Scaling 그룹 크기를 저장하고 중지 일정이 끝날 때 그룹을 복원할 때 사용합니다.

Amazon RDS 인스턴스 예약

- 스케줄러는 RDS 인스턴스를 중지하기 전에 스냅샷을 생성할 수 있습니다(Aurora DB 클러스터에는 적용되지 않음). 이 기능은 기본적으로 RDS 인스턴스 스냅샷 생성 AWS CloudFormation 템플릿 파라미터를 true로 설정하여 활성화됩니다. 스냅샷은 다음에 Amazon RDS 인스턴스가 중지되고 새 스냅샷이 생성될 때까지 유지됩니다.

  스케줄러는 클러스터 또는 Amazon RDS Aurora 데이터베이스의 일부이거나 다중 가용 영역(다중 AZ) 구성에 있는 Amazon RDS 인스턴스를 시작/중지할 수 있습니다. 그러나 스케줄러가 Amazon RDS 인스턴스, 특히 다중 AZ 인스턴스를 중지할 수 없는 경우 Amazon RDS 제한을 확인합니다.

Aurora 클러스터의 시작 또는 중지를 예약하려면 Aurora 클러스터 예약 템플릿 파라미터(기본값은 true)를 사용합니다. Aurora 클러스터(클러스터 내의 개별 인스턴스 아님)에는 초기 구성 중에 정의된 태그 키와 해당 클러스터를 예약하기 위한 태그 값으로 일정 이름을 지정해야 합니다.

모든 Amazon RDS 인스턴스에는 시스템 변경 사항이 적용되는 주간 유지 관리 기간이 있습니다. 유지 관리 기간 동안 Amazon RDS는 유지 관리를 적용하기 위해 7일 이상 중지된 인스턴스를 자동으로 시작합니다. 유지 관리 이벤트가 완료되면 Amazon RDS가 인스턴스를 중지하지 않습니다.

스케줄러를 사용하면 Amazon RDS 인스턴스의 기본 유지 관리 기간을 일정에 실행 기간으로 추가할지 여부를 지정할 수 있습니다. 다른 실행 기간에서 인스턴스를 실행하도록 지정하지 않고 유지 관리 이벤트가 완료된 경우 솔루션은 유지 관리 기간 시작 시 인스턴스를 시작하고 유지 관리 기간 종료 시 인스턴스를 중지합니다.

유지 관리 기간이 끝날 때까지 유지 관리 이벤트가 완료되지 않으면 유지 관리 이벤트가 완료된 후 예약 간격까지 인스턴스가 실행됩니다.

> **Note**
>
> 스케줄러는 리소스가 시작되거나 중지되었는지 확인하지 않습니다. API를 호출하고 계속 진행합니다. API 호출이 실패하면 조사를 위해 오류를 기록합니다.

애플리케이션 보안 고려 사항

애플리케이션 보안에는 애플리케이션을 실행해야 하는 권한, 방화벽 규칙, 애플리케이션 액세스를 위해 활성화해야 하는 IAM 역할을 고려하는 것이 포함됩니다.

일반적인 AWS 보안을 더 잘 이해하려면 보안, 자격 증명 및 규정 준수 모범 사례를 참조하세요.

구성 관리를 위한 액세스

AWS Managed Services(AMS)는 보안 문제, 패치 문제, 백업 문제 등에 대해 걱정할 필요가 없도록 문제 없는 인프라를 제공하려고 합니다. 이를 위해 AMS는 특정 그룹 또는 마스터 서버만 허용하는 최소 IAM 역할을 권장합니다. 애플리케이션 배포 도구를 사용하는 경우 애플리케이션을 실행하는 인스턴스에 액세스할 수 있습니다.

애플리케이션 액세스 방화벽 규칙

운영 체제(OS)와 마찬가지로 모든 애플리케이션 액세스는 Active Directory(AD) 그룹을 사용하여 관리해야 합니다. 예를 들어 Amazon Relational Database Service(RDS)를 사용하여 미러(복제)를 해제하여 새 사용자를 추가해야 합니다. 가장 좋은 방법은 AD에서 그룹을 생성하고 데이터베이스 생성 시 추가하는 것입니다. AMS AD에 그룹이 있으면 애플리케이션 액세스를 위한 CTs 생성할 수 있습니다. AD에 대한 공식 그룹화 전략에 대한 자세한 내용은 그룹 중첩 전략 사용 – 그룹 전략에 대한 AD 모범 사례를 참조하세요.

도메인 트리 및 상위/하위 도메인에 대한 자세한 내용은 도메인 및 포리스트 작동 방식을 참조하세요.

다음 규칙은 하위 도메인에 있는 사용자와의 다중 도메인 포리스트 신뢰에 적합한 솔루션을 보여줍니다.

Windows 인스턴스

다음은 Windows 상위 및 하위 도메인 컨트롤러에 대해를 구성하는 규칙입니다.

상위 도메인 컨트롤러, Windows

FROM: 상위 도메인 컨트롤러 TO: Windows 스택 및 공유 서비스 서브넷

소스 포트	대상 포트	프로토콜
88	49,152~65,535	TCP

소스 포트	대상 포트	프로토콜
389	49,152~65,535	UDP

FROM: 공유 서비스를 포함한 스택 서브넷 TO: Windows 포리스트 루트 도메인 컨트롤러

소스 포트	대상 포트	프로토콜
49,152~65,535	88	TCP
49,152~65,535	389	UDP

하위 도메인 컨트롤러, Windows

FROM: 하위 도메인 컨트롤러 TO: Windows AWS 도메인 컨트롤러

소스 포트	대상 포트	프로토콜
49,152~65,535	53	TCP
49,152~65,535	88	TCP
49,152~65,535	389	UDP

FROM: 하위 도메인 컨트롤러 TO: Windows 스택 및 공유 서비스 서브넷

소스 포트	대상 포트	프로토콜
88	49,152~65,535	TCP
135	49,152~65,535	TCP
389	49,152~65,535	TCP
389	49,152~65,535	UDP
445	49,152~65,535	TCP

소스 포트	대상 포트	프로토콜
49,152~65,535	49,152~65,535	TCP

FROM: 공유 서비스를 포함한 스택 서브넷 TO: Windows 하위 도메인 컨트롤러

소스 포트	대상 포트	프로토콜
49,152~65,535	88	TCP
49,152~65,535	135	TCP
49,152~65,535	389	TCP
49,152~65,535	389	UDP
49,152~65,535	445	TCP
49,152~65,535	49,152~65,535	TCP

Linux 인스턴스

다음은 Linux 상위 및 하위 도메인 컨트롤러에 대해를 구성하는 규칙입니다.

모든 테스트는 Amazon Linux를 사용하여 수행되었습니다. Windows의 동적 포트 범위는 49152~65535이지만 많은 Linux 커널은 포트 범위 32768~61000을 사용합니다. 아래 명령을 실행하여 IP 포트 범위를 확인합니다.

```
cat /proc/sys/net/ipv4/ip_local_port_range
```

상위 도메인 컨트롤러, Linux

FROM: 상위 도메인 컨트롤러 TO: Linux 스택 및 공유 서비스 서브넷

소스 포트	대상 포트	프로토콜
389	32768~61000	UDP
88	32768~61000	TCP

FROM: 공유 서비스를 포함한 스택 서브넷 TO: Linux 포리스트 루트 도메인 컨트롤러

소스 포트	대상 포트	프로토콜
32768~61000	88	TCP
32768~61000	389	UDP

하위 도메인 컨트롤러, Linux

FROM: 하위 도메인 컨트롤러 TO: Linux AWS 도메인 컨트롤러

소스 포트	대상 포트	프로토콜
49,152~65,535	53	TCP
49,152~65,535	88	TCP
389	49,152~65,535	UDP
49,152~65,535	389	UDP

FROM: 하위 도메인 컨트롤러 TO: Linux 스택 및 공유 서비스 서브넷

소스 포트	대상 포트	프로토콜
88	32768~61000	TCP
389	32768~61000	UDP

FROM: 공유 서비스를 포함한 스택 서브넷 TO: Linux 하위 도메인 컨트롤러

소스 포트	대상 포트	프로토콜
32768~61000	88	TCP
32768~61000	389	UDP

AMS 송신 트래픽 관리

기본적으로 AMS 프라이빗 및 고객 애플리케이션 서브넷의 대상 CIDR이 0.0.0.0/0인 라우팅에는 NAT(네트워크 주소 변환) 게이트웨이가 대상으로 사용됩니다. AMS 서비스인 TrendMicro 및 패치는 AMS가 서비스를 제공할 수 있고 TrendMicro 및 운영 체제가 업데이트를 받을 수 있도록 인터넷에 대한 외부 액세스 권한이 있어야 하는 구성 요소입니다.

AMS는 다음과 같은 경우에 한해 고객 관리형 송신 디바이스를 통해 송신 트래픽을 인터넷으로 전환하도록 지원합니다.

- 암시적(예: 투명) 프록시 역할을 합니다.

  and

- AMS HTTP 및 HTTPS 종속성(이 섹션에 나열됨)을 허용하여 AMS 관리형 인프라를 지속적으로 패치하고 유지 관리할 수 있습니다.

다음은 몇 가지 예시입니다.

- 전송 게이트웨이(TGW)에는 다중 계정 랜딩 존 네트워킹 계정의 AWS Direct Connect 연결을 통해 고객 관리형 온프레미스 방화벽을 가리키는 기본 경로가 있습니다.
- TGW에는 AWS PrivateLink를 활용하는 다중 계정 랜딩 존 송신 VPC의 AWS 엔드포인트를 가리키는 기본 경로가 있으며, 다른 AWS 계정의 고객 관리형 프록시를 가리킵니다.
- TGW에는 다른 AWS 계정의 고객 관리형 방화벽을 가리키는 기본 경로가 있으며, site-to-site VPN 연결은 다중 계정 랜딩 존 TGW에 대한 연결로 사용됩니다.

AMS는 해당 AMS HTTP 및 HTTPS 종속성을 식별했으며 이러한 종속성을 지속적으로 개발하고 구체화합니다. egressMgmt.zip을 참조하세요. ZIP에는 JSON 파일과 함께 README가 포함되어 있습니다.

> **Note**
>
> - 이 정보는 포괄적이지 않습니다. 일부 필수 외부 사이트는 여기에 나열되지 않습니다.
> - 거부 목록 또는 차단 전략에서이 목록을 사용하지 마십시오.
> - 이 목록은 송신 필터링 규칙 세트의 시작점으로 사용되며, 보고 도구를 사용하여 실제 트래픽이 목록에서 분기하는 위치를 정확하게 결정할 것으로 예상됩니다.

송신 트래픽 필터링에 대한 정보를 요청하려면 CSDM에 ams-csdm@amazon.com으로 이메일을 보내세요.

보안 그룹

AWS VPCs에서 AWS 보안 그룹은 하나 이상의 스택(인스턴스 또는 인스턴스 세트)에 대한 트래픽을 제어하는 가상 방화벽 역할을 합니다. 스택이 시작되면 스택이 하나 이상의 보안 그룹과 연결되어 스택에 도달할 수 있는 트래픽이 결정됩니다.

- 퍼블릭 서브넷의 스택의 경우 기본 보안 그룹은 모든 위치(인터넷)의 HTTP(80) 및 HTTPS(443) 트래픽을 수락합니다. 또한 스택은 회사 네트워크의 내부 SSH 및 RDP 트래픽과 AWS 접속을 허용합니다. 그러면 이러한 스택이 모든 포트를 통해 인터넷으로 나갈 수 있습니다. 또한 프라이빗 서브넷과 퍼블릭 서브넷의 다른 스택으로 나갈 수도 있습니다.
- 프라이빗 서브넷의 스택은 프라이빗 서브넷의 다른 스택으로 송신할 수 있으며, 스택 내의 인스턴스는 모든 프로토콜을 통해 서로 완전히 통신할 수 있습니다.

> **Important**
>
> 프라이빗 서브넷의 스택에 대한 기본 보안 그룹을 사용하면 프라이빗 서브넷의 모든 스택이 해당 프라이빗 서브넷의 다른 스택과 통신할 수 있습니다. 프라이빗 서브넷 내의 스택 간 통신을 제한하려면 제한을 설명하는 새 보안 그룹을 생성해야 합니다. 예를 들어 프라이빗 서브넷의 스택이 특정 포트를 통해서만 특정 애플리케이션 서버에서 통신할 수 있도록 데이터베이스 서버와의 통신을 제한하려면 특수 보안 그룹을 요청합니다. 이렇게 하는 방법은이 단원에 설명되어 있습니다.

기본 보안 그룹

MALZ

다음 표에서는 스택의 기본 인바운드 보안 그룹(SG) 설정을 설명합니다. SG의 이름은 "SentinelDefaultSecurityGroupPrivateOnly-vpc-ID"이며, 여기서 *ID*는 AMS 다중 계정 랜딩 존 계정의 VPC ID입니다. 모든 트래픽은이 보안 그룹을 통해 "mc-initial-garden-SentinelDefaultSecurityGroupPrivateOnly"로 아웃바운드할 수 있습니다(스택 서브넷 내의 모든 로컬 트래픽 허용).

모든 트래픽은 두 번째 보안 그룹 "SentinelDefaultSecurityGroupPrivateOnly"에 의해 0.0.0.0/0으로 아웃바운드할 수 있습니다.

> **Tip**
>
> EC2 생성 또는 OpenSearch 생성 도메인과 같은 AMS 변경 유형에 대한 보안 그룹을 선택하는 경우 여기에 설명된 기본 보안 그룹 중 하나 또는 생성한 보안 그룹을 사용합니다. AWS EC2 콘솔 또는 VPC 콘솔에서 VPC당 보안 그룹 목록을 찾을 수 있습니다.

내부 AMS용으로 사용되는 추가 기본 보안 그룹이 있습니다.

AMS 기본 보안 그룹(인바운드 트래픽)

Type	프로토콜	포트 범위	소스
모든 트래픽	모두	모두	SentinelDefaultSecurityGroupPrivateOnly(동일한 보안 그룹의 구성원으로의 아웃바운드 트래픽 제한)
모든 트래픽	모두	모두	SentinelDefaultSecurityGroupPrivateOnlyEgressAll(아웃바운드 트래픽을 제한하지 않음)
HTTP, HTTPS, SSH, RDP	TCP	80/443(소스 0.0.0.0/0) Bastion에서 SSH 및 RDP 액세스 허용	SentinelDefaultSecurityGroupPublic(아웃바운드 트래픽을 제한하지 않음)
MALZ 접속:			
SSH	TCP	22	SharedServices VPC CIDR 및 DMZ VPC CIDR과 고객이 제공한 온프레미스 CIDRs
SSH	TCP	22	
RDP	TCP	3389	
RDP	TCP	3389	
SALZ 접속:			

Type	프로토콜	포트 범위	소스
SSH	TCP	22	mc-initial-garden-LinuxBastionSG
SSH	TCP	22	mc-initial-garden-LinuxBastionDMZSG
RDP	TCP	3389	mc-initial-garden-WindowsBastionSG
RDP	TCP	3389	mc-initial-garden-WindowsBastionDMZSG

SALZ

다음 표에서는 스택의 기본 인바운드 보안 그룹(SG) 설정을 설명합니다. SG의 이름은 "mc-initial-garden-SentinelDefaultSecurityGroupPrivateOnly-*ID*"이며, 여기서 *ID*는 고유 식별자입니다. 모든 트래픽은이 보안 그룹을 통해 "mc-initial-garden-SentinelDefaultSecurityGroupPrivateOnly"로 아웃바운드할 수 있습니다(스택 서브넷 내의 모든 로컬 트래픽 허용).

모든 트래픽은 두 번째 보안 그룹 "mc-initial-garden-SentinelDefaultSecurityGroupPrivateOnlyEgressAll-*ID*"에 의해 0.0.0.0/0으로 아웃바운드할 수 있습니다.

> **Tip**
>
> EC2 생성 또는 OpenSearch 생성 도메인과 같은 AMS 변경 유형에 대한 보안 그룹을 선택하는 경우 여기에 설명된 기본 보안 그룹 중 하나 또는 생성한 보안 그룹을 사용합니다. AWS EC2 콘솔 또는 VPC 콘솔에서 VPC당 보안 그룹 목록을 찾을 수 있습니다.

내부 AMS용으로 사용되는 추가 기본 보안 그룹이 있습니다.

AMS 기본 보안 그룹(인바운드 트래픽)

Type	프로토콜	포트 범위	소스
모든 트래픽	모두	모두	SentinelDefaultSecurityGroupPrivateOnly(동일한 보안 그룹의 구성원으로의 아웃바운드 트래픽 제한)

Type	프로토콜	포트 범위	소스
모든 트래픽	모두	모두	SentinelDefaultSecurityGroupPrivateOnlyEgressAll(아웃바운드 트래픽을 제한하지 않음)
HTTP, HTTPS, SSH, RDP	TCP	80/443(소스 0.0.0.0/0) Bastion에서 SSH 및 RDP 액세스 허용	SentinelDefaultSecurityGroupPublic(아웃바운드 트래픽을 제한하지 않음)
MALZ 접속:			
SSH	TCP	22	SharedServices VPC CIDR 및 DMZ VPC CIDR과 고객이 제공한 온프레미스 CIDRs
SSH	TCP	22	
RDP	TCP	3389	
RDP	TCP	3389	
SALZ 접속:			
SSH	TCP	22	mc-initial-garden-LinuxBastionSG
SSH	TCP	22	mc-initial-garden-LinuxBastionDMZSG
RDP	TCP	3389	mc-initial-garden-WindowsBastionSG
RDP	TCP	3389	mc-initial-garden-WindowsBastionDMZSG

보안 그룹 생성, 변경 또는 삭제

사용자 지정 보안 그룹을 요청할 수 있습니다. 기본 보안 그룹이 애플리케이션 또는 조직의 요구 사항을 충족하지 않는 경우 새 보안 그룹을 수정하거나 생성할 수 있습니다. 이러한 요청은 승인 필수로 간주되며 AMS 운영 팀에서 검토합니다.

스택 및 VPCs 외부에서 보안 그룹을 생성하려면 `Deployment | Advanced stack components | Security group | Create (review required)` 변경 유형(ct-1oxx2g2d7hc90)을 사용하여 RFC를 제출합니다.

Active Directory(AD) 보안 그룹 수정의 경우 다음 변경 유형을 사용합니다.

- 사용자를 추가하려면: 관리 | 디렉터리 서비스 | 사용자 및 그룹 | 그룹에 사용자 추가 [ct-24pi85mjtza8k]를 사용하여 RFC 제출
- 사용자를 제거하려면: 관리 | 디렉터리 서비스 | 사용자 및 그룹 | 그룹에서 사용자 제거 [ct-2019s9y3nfml4]를 사용하여 RFC 제출

> **Note**
>
> '검토 필요' CTs를 사용하는 경우 ASAP 예약 옵션(콘솔에서 ASAP 선택, API/CLI에서 시작 및 종료 시간 비워 두기)을 사용하는 것이 좋습니다. 이러한 CTs는 AMS 운영자가 RFC를 검사하고 승인 및 실행 전에 사용자와 통신해야 하기 때문입니다. 이러한 RFCs 예약하는 경우 최소 24시간을 허용해야 합니다. 예약된 시작 시간 전에 승인이 이루어지지 않으면 RFC가 자동으로 거부됩니다.

보안 그룹 찾기

스택 또는 인스턴스에 연결된 보안 그룹을 찾으려면 EC2 콘솔을 사용합니다. 스택 또는 인스턴스를 찾은 후 스택 또는 인스턴스에 연결된 모든 보안 그룹을 볼 수 있습니다.

명령줄에서 보안 그룹을 찾고 출력을 필터링하는 방법은 섹션을 참조하세요describe-security-groups.

부록: 애플리케이션 온보딩 설문지

AMS가 필요한 인프라 구성 요소를 결정할 수 있도록이 설문지를 사용하여 배포 요소와 구조를 설명합니다. LoB(LineLine-of-Business) 애플리케이션의 온보딩 요구 사항은 제품 애플리케이션과 크게 다르므로이 설문은 두 가지 모두를 해결하도록 설계되었습니다.

주제

배포 요약

배포에 대한 설명입니다. 예:

- 이 계정은 (제품 애플리케이션 배포가 아닌) LoB(LineLine-of-Business) 애플리케이션 배포용입니다.
- 배포에는 계정의 퍼블릭/DMZ 서브넷 내에서 자동 조정된 ARP(인증된 역방향 프록시)가 포함됩니다.
- 웹 및 애플리케이션 서버는 계정의 프라이빗 서브넷 내에 배포됩니다.
- Amazon RDS(Amazon Relational Database Service) 인스턴스도 계정의 프라이빗 서브넷 내에 배포됩니다.
- 서버(ARP, 웹, 애플리케이션, 데이터베이스, 로드 밸런서 등)는 고유한 보안 그룹으로 구분됩니다.
- 계정에는 가용 영역(AZs), 즉 다중 AZ에 분산된 HA(고가용성) 설계가 필요합니다.

인프라 배포 구성 요소

애플리케이션을 지원하도록 구성해야 하는 구성 요소는 모두 무엇입니까?

- 리전: 필요한 AWS 리전 또는 리전은 무엇입니까?
- 고가용성(HA): 어떤 가용 영역이 사용됩니까?

- Virtual Private Cloud(VPC): VPC의 CIDR 블록은 무엇입니까?
- 필요한 서버 인스턴스는 무엇입니까?
  - 인증된 역방향 프록시(ARP): OS, AMI, 인스턴스 유형, 서브넷 ID, 보안 그룹, 수신 포트
  - Application Deployment Tool 서버: OS, AMI, 인스턴스 유형, 서브넷 ID, 보안 그룹, 수신 포트(Chef, Puppet) 또는 송신 포트(Ansible, Saltstack) 포트?
  - MySQL을 사용하는 Amazon RDS: DB 버전, 사용 유형, 인스턴스 클래스, 서브넷 ID, 보안 그룹, DB 인스턴스 ID, 스토리지 크기, 다중 AZ, 인증 유형, 암호화?
  - 스토리지: 앱이 상태 비저장입니까? S3 버킷이 필요합니까? 영구 스토리지가 필요합니까? EBS 볼륨에 저장 데이터 암호화가 필요합니까? DB 암호화가 필요합니까?
  - 외부(관리형 서비스 VPC) 서버 엔드포인트: SMTP? LDAP?
  - 네트워크 요구 사항: 네트워크 필터링(보안 그룹 기반?) 웹 트래픽 검사(인바운드?아웃바운드?)
- 태그 지정: 리소스를 논리적 컬렉션으로 그룹화하는 데 사용해야 하는 태그는 무엇입니까? 예를 들어 애플리케이션 스택의 모든 리소스입니다. 예를 들어 백업을 활성화`backup=true`하려면 사용 사례에 맞는 태그를 선택합니다. 또한 생성한 `name=value` EC2 인스턴스가 콘솔에 이름을 표시하려면 태그를 사용해야 합니다.
- 보안 그룹:
  - 필요한 보안 그룹은 무엇입니까?
  - 보안 그룹 수신 규칙
  - 보안 그룹 송신 규칙

애플리케이션 호스팅 플랫폼

애플리케이션 호스팅 플랫폼의 경우 다음과 같은 가능한 요구 사항을 고려하세요.

- 데이터베이스 암호화 여부
- 암호화 키는 누가 관리하나요?
- 전송 중 및 유휴 상태의 모든 데이터는 암호화되어 있습니까?
- HTTPS를 통해 시스템에 액세스하는 모든 사용자
- 보안 운영 팀에서 승인한 모든 system-to-system 상호 작용

애플리케이션 배포 모델

애플리케이션 배포를 계획하는 방법에 대한 고려 사항입니다. 운영 모델이란 무엇인가요? 섹션을 참조하세요

- 자동 또는 수동? 배포 자동화가 없다는 것은 Auto Scale이 없다는 의미입니다. 액세스를 요청하고 로그인한 후 애플리케이션을 수동으로 업데이트하면 업데이트가 실패합니다. AMS는 사용자가 서비스 요청을 통해 업데이트를 롤백하거나 알림을 보내 도움을 줄 것으로 예상합니다.
- 자동화된 경우 프레임워크란 무엇입니까? 스크립트? 에이전트 기반(puppet/chef)? 에이전트 없음(SALT/Ansible)? CodeDeploy? 에이전트 기반 및 에이전트 없는 배포 도구를 사용하려면 별도의 인스턴스를 생성하고 도구의 마스터 서버로 배포해야 합니다. AMS는 성공적인 애플리케이션 배포 도구에 필요한 모든 요소를 알고 있기를 기대하지만 관련 인프라 질문에 도움을 드릴 수 있습니다.
- Line-of-Business 애플리케이션(애플리케이션을 생성하고 관리하는 데 사용하는 애플리케이션)에 패치가 필요합니까?

애플리케이션 종속성

LoB(LineLine-of-Business) 애플리케이션용 인스턴스가 필요합니까? 제품 애플리케이션의 경우

제품 애플리케이션이 제대로 작동하려면 무엇이 필요한가요?

- 네트워크 수준 종속성: 예: AWS Direct Connect
- 패키지 종속성: 예: `pip`
- 이 애플리케이션이 의존하는 애플리케이션: 예: MySql
- 방화벽 종속성

LoB 애플리케이션이 제대로 작동하려면 무엇이 필요한가요?

- 네트워크 수준 종속성: 예: AWS Direct Connect
- 패키지 종속성: 예: Firefox Saucy
- 이 애플리케이션이 의존하는 애플리케이션: 예: MySql
- 방화벽 종속성

제품 애플리케이션용 SSL 인증서

애플리케이션(LoB 및 제품)이 실행하고 액세스하는 데 필요한 모든 것에 도달할 수 있도록 서버에 필요한 SSL 인증서는 무엇입니까?

- Auto Scaling 그룹?
- 데이터베이스(Amazon RDS)?
- Load Balancer서?
- 배포 도구 서버?
- 웹 애플리케이션 방화벽(AWS WAF)?
- 다른 인스턴스?

예를 들어 위에 나열된 각 인스턴스에 대해 다음 인증서가 필요할 수 있습니다.

WAF(인증서 1) - > ELB-Ext(인증서 2) - > ARP(인증서 3) - > ELB-Int(인증서 4) -> 웹 사이트(인증서 5) - > ELB-Int(인증서 6) -> 웹 서비스(인증서 7).

문서 이력

다음 표에서는이 AMS 릴리스에 대한 설명서를 설명합니다.

- API 버전: 2019-05-21
- 최종 설명서 업데이트: 2023년 2월 16일

변경 사항	설명	링크
TOC 링크 제거됨	TOC AWS 용어집 링크가 제거되었습니다.	2025년 8월 8일
업데이트된 콘텐츠: 워크로드 마이그레이션: Windows 사전 수집 검증	사전 WIGs 검증기 스크립트를 사용하여 Windows 인스턴스를 AMS 계정으로 수집할 준비가 되었는지 확인하는 자세한 단계를 포함하도록 섹션을 업데이트했습니다.	워크로드 마이그레이션: Windows 사전 통합 검증
업데이트된 콘텐츠, DMS 구성	필수 역할인 dms-vpc-role에 대한 중요한 참고 사항입니다.	1: AWS DMS 복제 서브넷 그룹: 생성
업데이트된 콘텐츠, CFN Ingest 지원 리소스	OpenSearch를 추가했습니다.	지원되는 리소스
업데이트된 콘텐츠, 워크로드 마이그레이션	사전 수집 검증에 대한 지침이 업데이트되었습니다.	워크로드 마이그레이션: Windows 사전 통합 검증
업데이트된 콘텐츠, CFN Ingest.	CFN 수집 콘텐츠에서 제한된 "지원되는 리소스"를 제거했습니다.	CloudFormation Ingest Stack: 지원되는 리소스
지원되는 Windows 버전 업데이트	Windows Server 2022에 대한 지원이 추가되었습니다.	AMS Amazon Machine Image(AMI

변경 사항	설명	링크
		s), 워크로드 마이그레이션: Linux 및 Windows의 사전 조건 및 워크로드 마이그레이션: Windows 사전 통합 검증
업데이트된 콘텐츠, Resource Scheduler.	SALZ와 MALZ 모두에 적용되는 전용 배포 CT인 ct-0ywnhc8e5k9z5를 사용하도록 지침을 업데이트했습니다.	AMS Resource Scheduler 빠른 시작
업데이트된 콘텐츠, 워크로드 수집.	지원되는 SUSE Linux 버전을 업데이트했습니다.	워크로드 마이그레이션: Linux 및 Windows의 사전 조건
업데이트된 콘텐츠, Database Migration Service.	사전 조건에를 추가하고 유용성과 유용성을 위해 몇 가지를 변경했습니다.	AWS Database Migration Service (AWS DMS)
업데이트된 콘텐츠, 워크로드 수집.	Linux Pre-WIGS Validation Zip이 업데이트되었습니다.	워크로드 마이그레이션: Linux 및 Windows의 사전 조건

변경 사항	설명	링크
콘텐츠가 업데이트되었습니다.	Linux용 WIGS 사전 검증 zip을 업데이트했습니다. 또한는 Windows Server 2008 R2를 지원되는 운영 체제로 추가했습니다.	워크로드 마이그레이션: Linux 및 Windows의 사전 조건
새로운 내용	퀵 스타트 및 자습서가 사용 중지된 AMS 고급 변경 관리 안내서에서 이동되었습니다.	빠른 시작, 자습서.
업데이트된 콘텐츠	배포 | 고급 스택 구성 요소 | Database Migration Service(DMS) | 복제 작업 시작(ct-1yq7hhqse71yg) DocumentName 및 리전이 필수 파라미터임을 나타내도록 업데이트되었습니다. 이전에는 선택 사항으로 잘못 나열되었습니다.	Database Migration Service(DMS) | 복제 작업 시작
업데이트된 콘텐츠	CloudFormation 수집 지원되는 두 가지 새로운 리소스인 AWS::Route53Resolver::ResolverRuleAssociation 및 AWS::Route53Resolver::ResolverRule을 나타내도록 업데이트되었습니다.	지원되는 리소스
업데이트된 콘텐츠	워크로드 마이그레이션: Windows 사전 통합 검증	Sysprep 정보가 더 구체적인 내용으로 업데이트되었습니다. 워크로드 마이그레이션: Windows 사전 통합 검증

변경 사항	설명	링크
업데이트된 콘텐츠	관리 | 사용자 지정 스택 | CloudFormation 템플릿의 스택 | 변경 세트 승인 및 업데이트(ct-1404e21baa2ox) ChangeSetName 파라미터에 대한 CT 연습 설명이 추가 정보로 업데이트되었습니다.	CloudFormation 템플릿의 스택 | 변경 세트 승인 및 업데이트
	Ubuntu 18.04 및 Oracle Linux 8.3 사용 가능	워크로드 마이그레이션: Linux 및 Windows의 사전 조건
새 콘텐츠:	CFN Ingest 및 Stack Update CTs 통한 IAM 배포.	2022년 2월 10일
Database Migration Service(DMS) 복제 작업	정규 표현식이 하이픈이 포함된 작업 ARNs 허용하도록 변경 유형이 업데이트되었습니다. AWS DMS 복제 작업 시작 및 Database Migration Service(DMS) | 복제 작업 중지.	2022년 1월 13일
Linux WIGS 사전 수집 검증	zip 파일이 업데이트되었습니다. 워크로드 마이그레이션: Linux 사전 수집 검증	2022년 1월 13일
링크 수정	데이터베이스(DB) AMS SQL RDS로 가져오기 -> 설정 섹션에 몇 가지 잘못된 링크가 있습니다.	2022년 1월 13일

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.