AMS 高级应用程序部署选项

AMS 高级应用程序开发者指南

版本 September 13, 2024

AMS 高级应用程序开发者指南: AMS 高级应用程序部署选项

Table of Contents

应用程序入门

欢迎来到 AWS Managed Services (AMS) AMS 运营计划。本文档的目的是描述在设置了初始网络和访问管理后，在将应用程序加载到 AMS 时可以使用的各种方法，以及在选择这些方法时应考虑的问题。

本文档适用于系统集成商和应用程序开发人员，以帮助他们为AMS的新客户确定和制定申请流程。

什么是应用程序入门？

AMS 应用程序入门是指根据需要将资源和应用程序部署到您的 AMS 基础设施中。在 AMS 平台上构建应用程序和基础设施与在原生 AWS平台上构建应用程序和基础设施非常相似。遵循 AWS 应用程序和基础设施设计最佳实践，同时考虑 AMS 提供的功能，将生成托管在 AMS 环境中的强大且可操作的应用程序。

> **Note**
>
> - 美国东部（弗吉尼亚）
> - 美国西部（加利福尼亚北部）
> - 美国西部（俄勒冈州）
> - 美国东部（俄亥俄州）
> - 加拿大（中部）
> - 南美洲（圣保罗）
> - 欧洲（爱尔兰）
> - 欧洲（法兰克福）
> - 欧洲（伦敦）
> - 欧盟西部（巴黎）
> - 亚太地区（孟买）
> - 亚太地区（首尔）
> - 亚太地区（新加坡）
> - 亚太地区（悉尼）
> - 亚太地区（东京）
>
> 经常会添加新区域。要了解更多信息，请参阅AWS 区域 和可用区。

我们做什么，不做什么

AMS 为您提供部署 AWS 基础设施的标准化方法，并提供必要的持续运营管理。有关角色、职责和支持的服务的完整描述，请参阅服务描述。

> **Note**
>
> 要请求 AMS 提供其他 AWS 服务，请提交服务请求。有关更多信息，请参阅提出服务请求。

- 我们做什么：

  完成入职后，AMS 环境可以接收变更请求 (RFCs)、事件和服务请求。您与 AMS 服务的交互围绕着应用程序堆栈的生命周期。新堆栈是从预先配置的模板列表中订购的，启动到特定的虚拟私有云 (VPC) 子网，在运行期间通过更改请求 (RFCs) 对其进行修改，并全天候监控事件和事件。

  主动应用程序堆栈由 AMS 监控和维护，包括修补，除非需要更改或堆栈已停用，否则在堆栈的生命周期内无需采取任何进一步的操作。AMS 检测到的影响堆栈运行状况和功能的事件会生成通知，可能需要也可能不需要您采取措施来解决或验证。可以通过提交服务请求来提出操作方法问题和其他查询。

  此外，AMS 还允许您启用不由 AMS 管理的兼容 AWS 服务。有关兼容 AWS-AMS 的服务的信息，请参阅自助服务配置模式。

- 我们不做什么：

  虽然 AMS 通过提供许多手动和自动选项来简化应用程序部署，但您需要负责应用程序的开发、测试、更新和管理。AMS 为影响应用程序的基础设施问题提供故障排除帮助，但 AMS 无法访问或验证您的应用程序配置。

AMS Amazon 机器映像 (AMIs)

AMS 每月都会为支持 AMS 的操作系统生成更新的亚马逊系统映像 (AMIs)。此外，AMS 还基于 CIS 1 级基准测试为 AMS 支持的部分操作系统生成安全增强映像 (AMIs)。要了解哪些操作系统具有安全增强型映像，请参阅 AMS 安全用户指南，该指南可通过 AWS Artifact-> Reports 页面（在左侧导航窗格中找到“报告”选项）获得，该指南针对 AWS Managed Services 进行了筛选。要访问 AWS Artifact，可以联系您的 CSDM 获取说明或前往 AWS Artifact 入门。

要在新 AMS 发布时收到提醒，您可以订阅名 AMIs 为 “AMS AMI” 的亚马逊简单通知服务 (Amazon SNS) Simple Notification Service 通知主题。有关详细信息，请参阅通过 SNS 发送的 AMS AMI 通知。

AMS AMI 的命名惯例是:`customer-ams-<operating system>-<release date> - <version>`.（例如，`customer-ams-rhel6-2018.11-3`）

仅使用 AMIs 以开头的 AMS `customer`。

AMS 建议始终使用最新的 AMI。您可以通过以下任一方式找到最新 AMIs 的：

- 在 AMS 控制台中查看，在AMIs页面上。
- 查看最新的 AMS AMI CSV 文件，该文件可从您的 CSDM 中获得，也可以通过此 ZIP 文件获得：AMS 11.2024 AMI 内容和 ZIP 格式的 CSV 文件。

  有关过去的 AMI ZIP 文件，请参阅文档历史记录。
- 运行此 AMS `SKMS` 命令（需要 AMS SKMS 开发工具包）：

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

按操作系统 (OS) 向基础 AWS AMIs版添加的 AMS AMI 内容

- Linux AMIs：
  - AWS CLI 工具
  - NTP
  - 趋势科技端点保护服务代理
  - 代码部署
  - PBIS Enterprise /Beyond

    > **Note**
    >
    > 自2022年6月起， BeyondTrust 不再支持PBIS公开赛。2022 年 6 月之后，你不能在 AMS 支持 AMIs 的 PBIS Open 上使用。如果 AMS 在 2022 年 6 月之前支持您的 AMI，则您可以自行决定继续使用 PBIS Open。
  - SSM Agent

  - Yum 升级关键补丁
  - AMS 自定义脚本/管理软件（控制启动、AD 加入、监控、安全和日志记录）
- Windows 服务器 AMIs：
  - 微软.NET 框架 4.5
  - PowerShell 5.1
  - AWS 适用于 Windows 的工具 PowerShell
  - AMS PowerShell 模块控制启动、AD 加入、监控、安全和日志记录
  - 趋势科技端点保护服务代理
  - SSM Agent
  - CloudWatch Agent
  - EC2配置服务（通过 Windows Server 2012 R2）
  - EC2发布（2016 年 Windows Server 和 2019 年 Windows 服务器）
  - EC2LaunchV2（Windows Server 2022 及更高版本）

基于 Linux AMIs：

- 亚马逊 Linux 2023（最新次要版本）（不支持最低 AMI）
- 亚马逊 Linux 2（最新次要版本）
- 亚马逊 Linux (2ARM64)
- 红帽企业 8（最新次要版本）
- 红帽企业 9（最新次要版本）
- SUSE Linux 企业服务器 15 SP6
- Ubuntu Linux 20.04
- Ubuntu Linux 22.04
- Ubuntu Linux 24.04
- 亚马逊 Linux：有关产品概述、定价信息、使用信息和支持信息，请参阅亚马逊 Linux 2。

  有关更多信息，请参阅亚马逊 Linux 2 FAQs。
- 适用于 SAP 应用程序的 SUSE Linux 企业服务器 15 SP6：
  - 每个账户运行以下步骤一次：
    1. 导航到 AWS Marketplace。

2. 搜索 SUSE 15 SAP 产品。
3. 选择继续订阅。
4. 选择接受条款。

- 每次需要启动适用于 SAP Applications 的新 SUSE Linux 企业服务器 15 SP6 实例时，请完成以下步骤：
  1. 记下订阅的适用于 SAP Applications 的 SUSE Linux 企业服务器 15 AMI 的 AMI ID。
  2. 创建部署 | 高级堆栈组件 | EC2 堆栈 | 创建更改类型 ct-14027q0sjyt1h RFC。*InstanceAmiId*替换为您订阅的 AWS Marketplace AMI ID。

基于 Windows AMIs：

微软 Windows 服务器（2016、2019、2022 和 2025 年），基于最新的 Windows AMIs。

有关创建的示例 AMIs，请参阅创建 AMI。

离职 AMS AMIs：

AMS 不会在离职期间取消与您的任何 AMIs 共享信息，以免对您的任何依赖项造成影响。如果您想 AMIs 从您的账户中移除 AMS，可以使用 `cancel-image-launch-permission` API 隐藏特定内容 AMIs。例如，您可以使用以下脚本来隐藏之前与您的账户共享 AMIs 的所有 AMS：

```
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
```

您必须安装 AWS CLI v2，脚本才能毫无错误地执行。有关 AWS CLI 的安装步骤，请参阅安装或更新最新版本的 AWS CLI。有关该`cancel-image-launch-permission`命令的详细信息，请参阅`cancel-image-launch-permission`。

安全性得到增强 AMIs

AMS 根据 CIS 1 级基准为 AMS 支持的部分操作系统提供安全增强镜像 (AMIs)。要了解哪些操作系统具有可用的安全增强映像，请参阅 AWS Managed Services (AMS) 客户安全指南。要访问本指南，请打开 AWS Artifact，在左侧导航窗格中选择 “报告”，然后筛选 AWS Managed Services。有关如何访问的说明 AWS Artifact，请联系您的 CSDM 或参阅入门 AWS Artifact以了解更多信息。

AMS 关键术语

- AMS Advanced：AMS 高级文档的 “服务描述” 部分中描述的服务。参见服务说明。
- AMS 高级 AWS 账户：始终符合 AMS 高级入职要求中所有要求的账户。有关 AMS Advanced 权益、案例研究以及联系销售人员的信息，请参阅 AWS Managed Services。
- AMS 加速 AWS 账户：始终满足 AMS 加速入职要求中所有要求的账户。请参阅 AMS 加速入门。
- AWS Managed Services：AMS 和/或 AMS 加速。
- AWS Managed Services 账户：AMS 账户和/或 AMS Accelerate 账户。
- 关键建议： AWS 通过服务请求发布的建议，告知您必须采取行动来防范潜在的风险或资源中断或. AWS 服务如果您决定在指定日期之前不遵守关键建议，则您应对您的决定造成的任何损害承担全部责任。
- 客户请求的配置：以下文件中未标识的任何软件、服务或其他配置：
  - 加速：支持的配置或 AMS 加速；服务描述。
  - AMS 高级：支持的配置或 AMS 高级；服务描述。
- 事件沟通：AMS 通过在 AMS Accelerate 的 Support Center 和 AMS 控制台中创建的事件向您传达事件，或者您通过在 AMS Accelerate 的 AMS 控制台中创建的事件请求与 AMS 发生的事件。AMS Accelerate 控制台在控制面板上提供事件和服务请求摘要，并提供指向 Support Center 的链接以获取详细信息。
- 托管环境：由 AMS 运营的 AMS 高级账户和/或 AMS Accelerate 账户。

  对于 AMS Advanced，这些账户包括多账户着陆区 (MALZ) 和单账户着陆区 (SALZ) 账户。
- 账单开始日期： AWS 收到您在 AWS Managed Services 入职电子邮件中要求的信息后的下一个工作日。AWS Managed Services 入职电子邮件是指向您发送的电子邮件， AWS 用于收集在您的账户上激活 AWS Managed Services 所需的信息。

  对于您随后注册的账户，账单开始日期为 AWS Managed Services 为已注册账户发送 AWS Managed Services 激活通知后的第二天。AWS Managed Services 激活通知发生在以下情况下：

  1. 您授予对兼容 AWS 账户的访问权限并将其移交给 AWS Managed Services。
  2. AWS Managed Services 设计和建立 AWS 托管服务账户。
- 服务终止：您可以出于任何原因终止所有 AWS Managed Services 账户的 AWS Managed Services 账户的 AWS 托管服务，方法是通过服务请求提供至 AWS 少 30 天的通知。在服务终止日期，可以：

  1. AWS 将所有 AWS Managed Services 账户或指定的 AWS Managed Services 账户（如果适用）的控制权移交给您，或者

2. 双方删除 AWS 允许从所有 AWS Managed Services 账户或指定的 AWS Managed Services 账户进行访问的 AWS Identity and Access Management 角色（如果适用）。

- 服务终止日期：服务终止日期是必需的 30 天终止通知期结束后的日历月的最后一天。如果必要的终止通知期限在日历月的第20天之后，则服务终止日期为下一个日历月的最后一天。以下是终止日期的示例方案。
  - 如果终止通知是在4月12日提供的，则为期30天的通知将于5月12日结束。服务终止日期为5月31日。
  - 如果在4月29日发出终止通知，则为期30天的通知将于5月29日结束。服务终止日期为 6 月 30 日。
- 提供 AWS Manag AWS ed Services：从服务开始之日起，您可以访问和使用每个 AWS 托管服务账户的 AWS 托管服务。
- 终止指定的 AWS Managed Services 账户：您可以出于任何原因终止指定 AWS Managed Services 账户的 AWS 托管服务，方法是通过服务请求（“AMS 账户终止申请”） AWS 发出通知。

事件管理条款：

- 事件：您的 AMS 环境发生了变化。
- 警报：每当来自支持的事件 AWS 服务 超过阈值并触发警报时，系统就会创建警报并将通知发送到您的联系人列表。此外，还会在您的事件列表中创建事件。
- 事件：您的 AMS 环境或 AWS Managed Services 的计划外中断或性能降级，导致影响，如 AWS Managed Services 或您所报告的那样。
- 问题：一个或多个事件的共同根本原因。
- 事件解决或解决事件：
  - AMS 已将与该事件相关的所有不可用 AMS 服务或资源恢复到可用状态，或者
  - AMS 已确定不可用的堆栈或资源无法恢复到可用状态，或者
  - AMS 已启动经您授权的基础设施恢复。
- 事件响应时间：创建事件与 AMS 通过控制台、电子邮件、服务中心或电话提供初始响应之间的时间差。
- 事件解决时间：AMS 或您创建事件与事件解决时间之间的时间差。
- 事件优先级：AMS 或您如何将事件的优先级分为 “低”、“中” 或 “高”。
  - 低：您的 AMS 服务存在非严重问题。
  - 中：您的托管环境中的 AWS 服务可用，但未按预期运行（根据适用的服务描述）。

- 高：(1) AMS 控制台或托管环境 APIs 中的一个或多个 AMS 不可用；或 (2) 托管环境中的一个或多个 AMS 堆栈或资源不可用，且不可用会使您的应用程序无法执行其功能。

  AMS 可以根据上述指南对事件进行重新分类。
- 基础设施恢复：根据受影响堆栈的模板重新部署现有堆栈，并在无法解决事件时根据上次已知的还原点启动数据恢复，除非您另行指定。

基础设施条款：

- 托管生产环境：客户生产应用程序所在的客户帐户。
- 托管的非生产环境：仅包含非生产应用程序（例如用于开发和测试的应用程序）的客户帐户。
- AMS 堆栈：由 AMS 作为一个单元管理的一组或多个 AWS 资源。
- 不可变基础设施：Amazon A EC2 uto Scaling 组 (ASGs) 的典型基础设施维护模式，在这种模式中 AWS，每次部署都会替换更新的基础设施组件（在 AMI 中），而不是就地更新。不可变基础架构的优势在于，所有组件都保持同步状态，因为它们总是从同一个基础生成的。不可变性独立于任何用于构建 AMI 的工具或工作流程。
- 可变基础设施：一种典型的基础设施维护模型，适用于不是 Amazon A EC2 uto Scaling 组且包含单个实例或仅包含几个实例的堆栈。该模型最接近于传统的、基于硬件的系统部署，即在系统生命周期开始时部署系统，然后随着时间的推移将更新分层到该系统上。系统的任何更新都将单独应用于实例，并且可能由于应用程序或系统重启而导致系统停机（取决于堆栈配置）。
- 安全组：您的实例的虚拟防火墙，用于控制入站和出站流量。安全组在实例级别运行，而不是子网级别。因此，您的 VPC 子网中的每个实例都可以为其分配一组不同的安全组。
- 服务级别协议 (SLAs)：与您签订的 AMS 合同的一部分，其中定义了预期的服务级别。
- SLA 不可用和不可用：
  - 您提交的导致错误的 API 请求。
  - 您提交的控制台请求生成 5xx HTTP 响应（服务器无法执行请求）。
  - 如Service Health Dashboard所示，在 AMS 管理的基础设施中构成堆栈或资源的任何 AWS 服务产品都处于 “服务中断” 状态。
  - 在确定服务积分资格时，不考虑因AMS排除而直接或间接导致的不可用性。除非服务符合不可用标准，否则视为可用。
- 服务级别目标 (SLOs)：与您签订的 AMS 合同的一部分，其中定义了 AMS 服务的具体服务目标。

修补条款：

- 强制补丁：关键安全更新，用于解决可能危及您的环境或账户安全状态的问题。“关键安全更新” 是由 AMS 支持的操作系统的供应商评为 “严重” 的安全更新。
- 已发布补丁与已发布补丁：补丁通常按计划发布和发布。紧急补丁是在发现需要补丁时宣布的，通常不久之后，补丁就会发布。
- 补丁附加组件：针对 AMS 实例进行基于标签的修补，它利用 AWS Systems Manager (SSM) 功能，因此您可以使用基准和您配置的窗口标记实例并对这些实例进行修补。
- 补丁方法：
  - 就地修补：通过更改现有实例完成的修补。
  - AMI 替换补丁：通过更改现有 A EC2 uto Scaling 组启动配置的 AMI 参考参数来完成的修补。
- 补丁提供商（操作系统供应商、第三方）：补丁由应用程序的供应商或管理机构提供。
- 补丁类型：
  - 关键安全更新 (CSU)：被支持的操作系统的供应商评为 “严重” 的安全更新。
  - 重要更新 (IU)：被支持的操作系统的供应商评为 “重要” 的安全更新或评级为 “严重” 的非安全更新。
  - 其他更新 (OU)：供应商对不是 CSU 或 IU 的支持的操作系统的更新。
- 支持的补丁：AMS 支持操作系统级补丁。供应商发布升级是为了修复安全漏洞或其他错误或提高性能。有关当前支持的列表 OSs，请参阅 Supp ort 配置。

安全条款：

- Det@@ ective Contro ls：由 AMS 创建或启用的监控器组成的库，用于持续监督客户托管的环境和工作负载，以发现与安全、运营或客户控制不一致的配置，并通过通知所有者、主动修改或终止资源来采取行动。

服务请求条款：

- 服务请求：您请求采取行动，希望 AMS 代表您采取行动。
- 警报通知：AMS 在触发 AMS 警报时在您的服务请求列表页面上发布的通知。为您的账户配置的联系人也会通过配置的方法（例如电子邮件）收到通知。如果您的实例/资源上有联系人标签，并且已同意您的云服务交付经理 (CSDM) 接收基于标签的通知，则还会通知标签中的联系人信息（密钥值）以获取自动的 AMS 警报。
- 服务通知：AMS 发布到您的服务请求列表页面的通知。

其他术语：

- AWS Managed Services 接口：适用于 AMS：AWS Managed Services 高级控制台、AMS CM AP 支持 I 和 API。对于 AMS Accelerate： 支持 控制台和 支持 API。
- 客户满意度 (CSAT)：AMS CSAT 通过深入分析获得信息，包括每个案例或信件的案例信件评级（如果给出）、季度调查等。
- DevOps: DevOps 是一种开发方法，强烈倡导在所有步骤上实现自动化和监控。 DevOps 旨在通过在自动化基础上整合传统上独立的开发和运营功能，缩短开发周期，提高部署频率和更可靠的发布。当开发人员可以管理运营，运营为开发提供信息时，问题和问题就会更快地发现和解决，业务目标也更容易实现。
- ITIL：信息技术基础设施库（称为 ITIL）是一个 ITSM 框架，旨在标准化 IT 服务的生命周期。ITIL 分为五个阶段，涵盖了 IT 服务生命周期：服务策略、服务设计、服务过渡、服务运营和服务改进。
- IT 服务管理 (ITSM)：一套让 IT 服务与您的业务需求保持一致的实践。
- 托管监控服务 (MMS)：AMS 运营自己的监控系统，即托管监控服务 (MMS)，该系统使用 AWS 健康事件并汇总亚马逊数据 AWS 服务和其他数据，将通过 CloudWatch 亚马逊简单通知服务 (Amazon SNS) Simple Notification Service 主题创建的任何警报通知AMS操作员（全天候在线）。
- 命名空间：在创建 IAM 策略或使用 Amazon 资源名称 (ARNs) 时，您可以使用命名空间来识别。AWS 服务 您可以使用命名空间来标识操作和资源。

我的运营模式是什么？

作为 AMS 客户，您的组织已决定将应用程序和基础设施运营分开，并使用 AMS 进行基础设施运营。AMS 将与您的应用程序设计和开发团队以及您的基础设施设计团队合作，确保您的基础设施运营顺利运行。下图说明了这个概念：

AMS 负责您的 AWS 基础设施运营，而您的团队则负责您的应用程序运营。作为应用程序和基础设施设计团队，您必须了解在 AMS 基础设施中将应用程序部署到生产环境后由谁来操作。本指南涵盖了与应用程序部署和维护相关的基础架构设计的常用方法。

AWS Managed Services 中的服务管理

主题

AMS 服务如何为您服务。

AWS Managed Services 中的账户管理

本节介绍 AMS 账户管理。

您被指定为云服务交付经理 (CSDM)，负责在整个 AMS 中提供咨询帮助，并详细了解托管环境的用例和技术架构。 CSDMs 与客户经理、技术客户经理、AWS Managed Services 云架构师 (SAs) 和 AWS 解决方案架构师 () 合作（如适用），以帮助启动新项目，并在整个软件开发和运营过程中提供最佳实践建议。CAsCSDM 是 AMS 的主要联系点。您的 CSDM 的主要职责是：

- 组织并主持与客户的月度服务审查会议。
- 提供有关安全性、环境软件更新和优化机会的详细信息。
- 支持您的要求，包括 AMS 的功能请求。
- 回应并解决账单和服务报告请求。
- 为财务和容量优化建议提供见解。

在 AWS Managed Services 中开始提供服务

服务开始：AWS Managed S ervices 账户的服务开始日期是第一个日历月的第一天，在此之后，AWS 会通知您该 AWS Managed Services 账户的入职要求中规定的活动已经完成；前提是，如果 AWS 在某个日历月的第 20 天之后发出此类通知，则服务开始日期为该通知之日后第二个日历月的第一天。

服务开始

- R 代表负责任的一方，负责完成任务。
- 我代表知情；一个通报进展情况的当事方，通常只有在任务或可交付成果完成后才会被告知。

服务开始

步骤 #	步骤标题	说明	Custome	AMS
1.	移交客户 AWS 账户	客户创建一个新的 AWS 账户并将其移交给 AWS Managed Services	R	我
2.	AWS Managed Services 账户-设计	完成 AWS Managed Services 账户的设计	我	R
3.	AWS Managed Services 账户-构建	AWS Managed Services 账户是按照步骤 2 中的设计创建的	我	R

客户关系管理 (CRM)

AWS Managed Services (AMS) 提供了客户关系管理 (CRM) 流程，以确保与您建立和维持明确的关系。这种关系的基础是AMS对您的业务需求的洞察。CRM 流程有助于准确、全面地理解：

- 您的业务需求以及如何满足这些需求
- 你的能力和限制
- AMS 和您的不同责任和义务

CRM 流程允许 AMS 使用一致的方法向您提供服务，并管理您与 AMS 的关系。CRM 流程包括：

- 确定您的主要利益相关者
- 组建治理团队
- 与您举行和记录服务审查会议
- 提供带有上报程序的正式服务投诉程序
- 实施和监控您的满意度和反馈流程

- 管理您的合同

CRM 流程

CRM 流程包括以下活动：

- 识别和了解您的业务流程和需求。您与 AMS 的协议确定了您的利益相关者。
- 定义要提供的服务以满足您的需求和要求。
- 在服务审查会议上与您会面，讨论 AMS 服务范围、SLA、合同和您的业务需求的任何变化。可能会与您举行临时会议，讨论绩效、成就、问题和行动计划。
- 使用我们的客户满意度调查和会议反馈来监控您的满意度。
- 在内部衡量的月度绩效报告中报告绩效。
- 与您一起审查服务，以确定改进的机会。这包括就所提供的 AMS 服务的水平和质量与您进行频繁沟通。

CRM 会议

AMS 云服务交付经理 (CSDMs) 定期与您开会，讨论服务轨道（运营、安全和产品创新）和高管方向（SLA 报告、满意度衡量标准和业务需求的变化）。

会议	用途	Mode	参与者
每周状态回顾（可选）	未解决的问题或事件、补丁、安全事件、问题记录 12 周运营趋势 (+/-6) 应用程序操作员关注的问题 周末日程安排	现场客户 location/ Telecom/Chime	AMS：CSDM 和云架构师 (CA) 客户分配的团队成员（例如：云/基础架构、Application Support、架构团队等）
每月业务回顾	查看服务级别性能（报告、分析和趋势） 财务分析	现场客户 location/ Telecom/Chime	AMS：CSDM、云架构师 (CA)、AMS 客户

会议	用途	Mode	参与者
	产品路线图 CSAT		团队、AMS 技术产品经理 (TPM)（可选）、AMS 运营经理（可选） 您：应用程序运营商代表
季度业务回顾	记分卡和服务级别协议 (SLA) 绩效和趋势（6 个月） 即将到来的 12 年 3 月 6 日/9 月计划/迁移 风险和风险缓解 关键改进举措 产品路线图项目 未来方向一致的机会 金融 成本节约举措 业务优化	现场客户位置	AMS：CSDM、云架构师、AMS 客户团队、AMS 服务总监、AMS 运营经理 您：应用程序运营商代表、服务代表、服务主管

CRM 会议安排

AMS CSDM 负责记录会议，包括：

- 创建议程，包括行动项目、议题和与会者名单。
- 创建每次会议上审查的措施项目列表，以确保项目按计划完成和解决。
- 在会议结束后的一个工作日内通过电子邮件向与会者分发会议纪要和行动项目列表。
- 将会议记录存储在相应的文档存储库中。

在CSDM缺席的情况下，主持会议的AMS代表编写和分发会议记录。

> **Note**
>
> 您的 CSDM 会与您合作建立您的账户管理。

CRM 月度报告

您的 AMS CSDM 准备并发送每月服务绩效演示文稿。演讲包括以下方面的信息：

- 举报日期
- 摘要和见解：
  - Key Call Outs：堆栈总数和活跃堆栈数、堆栈补丁状态、账户入职状态（仅限入职期间）、客户特定问题摘要
  - 性能：事件解决、警报、修补、变更请求 (RFCs)、服务请求以及控制台和 API 可用性的统计信息
  - 问题、挑战、疑虑和风险：客户特定的问题状态
  - 即将推出的项目：客户特定的入职培训或事件解决计划
- 托管资源：堆栈的图表和饼图
- AMS 指标：监控和事件指标、事件指标、AMS SLA 遵守指标、服务请求指标、变更管理指标、存储指标、连续性指标、Trusted Advisor 指标和成本摘要（以多种方式呈现）。功能请求。联系信息。

> **Note**
>
> 除了上述信息外，您的 CSDM 还会告知您范围或条款的任何重大变化，包括AMS使用分包商进行运营活动。
>
> AMS 会生成有关修补和备份的报告，您的 CSDM 将这些报告包含在您的月度报告中。作为报告生成系统的一部分，AMS 会为您的账户添加一些您无法访问的基础设施：
>
> - 一个 S3 存储桶，报告了原始数据
> - 一个 Athena 实例，带有用于查询数据的查询定义
> - 用于从 S3 存储桶读取原始数据的 Glue 爬行器

AWS Managed Services 中的成本优化

AWS Managed Services 每月都会在您的每月业务评估中向您提供详细的成本利用率和节省报告（MBRs）。

AMS 遵循一套标准的流程和机制来确定您的托管账户中的成本节约途径，并帮助您规划和推出变更以优化 AWS 支出。

> **Note**
>
> AMS 正在开发一段有助于成本优化的视频。第一步是为您提供一份包含成本优化最佳实践的 PDF 和 Excel 电子表格。要访问这些资源，请打开成本优化快速指南 ZIP 文件。

成本优化框架

AMS 采用三阶段方法来优化您的 AWS 成本：

1. 确定托管环境中的成本优化途径
2. 向您提交成本优化计划
3. 以可衡量的方式协助实现成本优化

确定托管环境中的成本优化途径

AMS 利用成本资源管理器和 Trusted Advisor 等 AWS 原生工具，同时利用架构优化、 EC2 实例和以 AWS 客户为中心的优化中的 20 多种成本节省模式，为您制定量身定制的成本节约建议。

一些优化建议包括以下内容。

架构优化建议：

- S3 存储类的最佳使用：Amazon S3 提供了一系列存储类别，以满足基于数据访问权限、弹性和成本的各种工作负载要求。基于工作负载需求的 S3 智能分层和 S3 存储类分析使您能够高效地管理 S3 成本。
- 使用缓存架构：在适用的情况下，利用缓存实例可以帮助您替换某些数据库实例，同时满足 IOPS 要求。
- EBS 升级节省开支：将 EBS 卷从 gp2 迁移到 gp3 可节省高达 20% 的成本，无论卷大小如何，您都可以利用可预测的 3,000 IOPS 基准性能和 125 MiB/s。

- 使用弹性： AWS 提供的自动缩放功能允许有效的资源利用率和成本优化的途径。根据需要定期审查和更新实例扩展策略，可以进一步节省成本。

EC2 以实例为重点的建议

- 调整实例大小：建议侧重于根据使用情况调整实例大小和优化配置。建议还包括使用 Amazon A EC2 uto Scaling 功能，在适用的情况下将 EC2 实例替换为 Amazon S3 上的静态网页内容等。 AWS Lambda
- 实例调度：使用 AMS 资源调度器根据时间表自动启动和停止实例有助于控制成本，特别是对于非工作时间未使用的非生产实例。
- 订阅储蓄计划：储蓄计划是节省 AWS 使用量的最简单方法。与按需定价相比，Instance Savings Plans 可为您的亚马逊 EC2 实例使用量节省高达 72% 的费用。 EC2 Amazon SageMaker AI Savings Plans 可为您的亚马逊 A SageMaker I 服务使用量节省高达 64% 的费用。AMS 会根据您的 AWS 资源使用情况提供相应的储蓄计划建议。
- 预留实例 (RI) 使用和消费指南：与按需定价相比，Amazon EC2 Reserved Instances (RI) 可提供大幅折扣（高达 75%），并且在特定可用区域中使用时会提供容量预留。
- 竞价型实例使用率：容错工作负载可以利用竞价型实例并将价格降低多达 90%。
- 空闲实例终止：识别并报告处于空闲状态或利用率低且可以终止的实例。

以账户为中心的推荐

- 账户清理：在账户层面，AMS 还会识别未使用的 EBS 卷、重复的 CloudTrail 跟踪、带有未使用资源的空账户等，并提供清理建议。
- SLA 建议：此外，AMS 会定期审查您的高级版和高级版账户，并建议为这些账户选择正确的 SLA 级别。
- AMS 自动化优化：AMS 不断优化用于提供 AMS 服务的 AMS 自动化和基础设施。

向客户演示并协助规划

AMS每月与主要的客户利益相关者进行业务审查 (MBRs)，并介绍已确定的成本节约途径、机制和建议以及潜在的成本节约。我们将进一步与您合作，计划所需的更改。

协助实施建议并衡量成本影响

AMS 有助于实现和衡量成本影响和优化变更。

您可以评估所建议更改的应用程序影响、风险和成功标准，并通过 AMS 控制台提出相应的更改请求 (RFCs)。AMS 与您合作，在您的托管账户中实施与成本优化相关的变更。AMS衡量成本影响，并在月度业务评估中包括实现的节省（MBRs）。

成本优化责任矩阵

AMS 成本优化方面的职责。

成本优化 RACI

活动	Customer	AMS
编制节省成本的建议并准备报告	我	R
提交成本节约报告	C	R
与成本节约相关的计划变更	R	C
评估变更的影响和风险	R	C
RFCs 为实施变更筹款	R	C
审查 RFCs 并实施变更	C	R
测试应用程序并验证变更实施	R	C

活动	Customer	AMS
衡量变更后的成本影响并提交给客户	我	R

AWS Managed Services 中的服务时间

功能	AMS 高级版
	高级等级
服务请求	全天候
事件管理 (P2-P3)	全天候
备份和恢复	全天候
补丁管理	全天候
监控和提醒	全天候
自动申请变更 (RFC)	全天候
非自动变更请求 (RFC)	全天候
云服务交付管理器 (CSDM)	周一至周五：08:00 — 17:00，当地工作时间

在 AWS Managed Services 中获取帮助

AMS 全年 365 天、每周 7 天、每天 24 小时为您提供事件管理、服务请求管理和变更管理方面的支持（根据适用于该账户的 AMS 服务等级协议）。

要报告影响您的托管环境的 AWS 或 AMS 服务性能问题，请使用 AMS 控制台并提交事件报告。有关详细信息，请参阅报告事件。有关 AMS 事件管理的一般信息，请参阅事件响应。

要向 AMS 询问信息或建议，或请求其他服务，请使用 AMS 控制台并提交服务请求。有关详细信息，请参阅创建服务请求。有关 AMS 服务请求的一般信息，请参阅服务请求管理。

应用程序开发

应用程序开发流程和实践，可将应用程序有效设计和部署到 AWS Managed Services (AMS) 环境中。AMS 将指导您完成以下高级流程：

1. 设想并设计一个要开发或集成到您的 AMS 管理的环境中的应用程序。一些注意事项：
    a. 您将如何部署应用程序？ 使用诸如 Ansible 之类的部署工具实现自动化，还是通过直接上传所需文件来手动完成？
    b. 您将如何更新您的应用程序？ 使用可变方法分别更新每个实例，还是使用不可变方法在 Auto Scaling 组中使用单个已更新的 AMI 更新每个实例？
2. 使用 AWS 架构库、 AWS “Well-Architected” 指南、AMS 和其他云架构主题专家规划和架构将用于托管应用程序的基础架构。本指南的以下部分提供了可以帮助解决此问题的信息。
3. 选择基础架构部署方法：
    a. 完整堆栈：同时部署所有基础架构组件。
    b. Tier and Tie：基础设施部署是单独部署的，然后与安全组修改绑定在一起。这种类型的部署也可以通过串行配置相互构建的堆栈组件来实现；例如，指定您之前在创建 Auto Scaling 组时创建的负载均衡器。
    c. 您将采用哪些环境，例如开发、暂存和生产？
4. 选择 AMS 变更类型 (CTs)，以配置必要的堆栈或等级，并准备必要的变更请求 (RFCs)。
5. 提交 RFCs ，以触发向相应环境部署基础架构。
6. 使用所选的应用程序部署方法部署应用程序。
7. 根据需要重新设计基础架构和应用程序。
8. 假设您的首次部署是在非生产环境中，则将基础架构和应用程序部署到相应的后续环境。
9. 持续维护由运营底层基础设施的 AMS 和运行应用程序基础架构的运营团队负责。
10. 要停用应用程序，请终止该应用程序的 AMS 基础架构。

架构精良

AWS 我们认为，架构良好的系统会大大增加业务成功的可能性。AWS 建筑中心提供有关建筑设计的专家指导。 AWS 云

我们推荐以下文章和白皮书，以帮助您了解在构建系统时必须做出的决策的利弊。 AWS

你是 Well-Architected 吗？：介绍基于六大支柱 AWS 的 Well-Architected 框架：

- 卓越运营：卓越运营支柱侧重于运行和监控系统以实现业务价值，并不断改进流程和程序。关键主题包括管理和自动化变更、响应事件以及定义成功管理日常运营的标准。
- 安全：安全支柱侧重于保护信息和系统。关键主题包括数据的机密性和完整性、通过权限管理识别和管理谁可以做什么、保护系统以及建立检测安全事件的控制措施。
- 可靠性：可靠性支柱侧重于预防故障并从故障中快速恢复的能力，以满足业务和客户需求。关键主题包括围绕设置、跨项目要求、恢复计划以及我们如何处理变更的基本要素。
- 性能效率：性能效率支柱侧重于高效使用IT和计算资源。关键主题包括：根据工作负载要求选择合适的资源类型和大小、监控性能以及做出明智的决策以跟随业务需求的变化保持效率。
- 成本优化：成本优化支柱侧重于避免不必要的成本。关键主题包括了解和控制资金花在哪里，选择最合适、最正确的资源类型，分析一段时间内的支出，以及在不超支的情况下进行扩展以满足业务需求。
- 可持续性：可持续性支柱侧重于通过最大限度地提高已配置资源的收益和最大限度地减少所需总资源，从而通过降低能耗和提高工作量所有组成部分的效率来持续改善可持续发展影响的能力。

AWS W@@ ell-Architected Framework：描述 AWS 如何使客户能够评估和改进其基于云的架构，并更好地了解其设计决策对业务的影响。它阐述了六个概念领域的总体设计原则以及具体的最佳实践和指导，这些领域 AWS 定义了Well-Architected框架的支柱。

AMS 中应用层职责与基础架构层职责

通过使用 AMS，您的基础设施及其维护和增长所需的一切都由 AMS 维护。但是，无论您需要什么 line-of-business应用程序或产品应用程序，都由您开发、部署和维护。

借助应用程序部署工具（例如 CodeDeploy 和、Chef、Puppet CloudFormation、Ansible 或 Saltstack），您可以完全自动化地将应用程序部署到 AMS 托管的基础架构。

有关 AMS 做什么和不做什么的详细信息，请参阅我们做什么，不做什么。

AMS 中的 Amazon EC2 实例可变性

您和 AMS 可以通过以下两种方式之一在您的基础设施中维护亚马逊弹性计算云 (Amazon EC2) 实例：

- 不可变：此模型使用经过烘烤（创建AMIs）并具有必要功能的 Amazon 系统映像 ()。部署更新时，现有实例将被拆除，并完全替换为根据更新的 AMI 创建的新实例。为了最大限度地减少停机时间，此滚动过程使某些实例无法更新和访问，而其他实例则在最终完全部署新更改之前进行更新。

- Mutable：在此模型中，通过在云中的现有系统上部署新代码来更新基础架构。此模型混合了手动推送更新和使用 infrastructure-as-code来部署更新，并且不依赖新更新 AMIs。

本指南后面的章节将详细讨论这些维护模型。

将 S AWS ecrets Manager 与 AMS 资源配合使用

在许多情况下，您可能需要与 AMS 共享机密，例如：

- RDS 实例的主密码重置
- 负载均衡器证书
- 从 AMS 获取 IAM 用户的长期证书

与 AMS 共享机密信息的最安全方法是通过 S AWS ecrets Manager；请按照以下步骤操作：

1. 使用您的联合访问权限和单账户着陆区 (SALZ) 的 CustomerReadOnly 角色登录 AWS 控制台；使用其中任何一个角色、 AWSManagedServicesSecurityOpsRole AWSManagedServicesAdminRole、和 AWSManagedServicesChangeManagementRole 用于多账户着陆区 (MALZ)。
2. 导航到 AWS Secrets 管理器控制台，然后单击 “存储新密钥”。
3. 选择 “其他类型的机密”。
4. 以纯文本形式输入密钥值，然后单击 “下一步”。
5. 输入密钥名称和描述。名称应始终以 “客户共享/ *” 开头。例如 “客户共享/ 许可证 2018”。完成后，单击 “下一步” 继续。
6. 使用默认 KMS 加密。
7. 禁用自动轮换，然后单击 “下一步”。
8. 查看并单击 “存储” 以保存密钥。
9. 在 AMS 服务请求中回复我们，提供机密名称和 ARN，这样我们就可以识别和检索密钥。有关创建服务请求的信息，请参阅服务请求示例。

AMS 中的应用程序部署

在入职期间，AWS Managed Services (AMS) 会与您合作确定所需的基础设施。

基本基础设施包括 AWS 虚拟私有云 (VPC)、通过 ADFS 林信任实现的通信安全、跨两个可用区镜像并配置了托管 NAT、堡垒、公共负载均衡器 (DX) 的基本子网 Direct Connect （DMZ、共享服务和私有子网）以及所需的安全。您的应用程序资源将部署在您的私有子网或客户应用程序子网中。您可以在 AWS Managed Services 用户指南中了解有关典型 AMS 架构的更多信息。

基础知识完成后，您部署的基础架构应包括应用程序和应用程序开发的所有组件。

AMS 中的应用程序部署功能

在 AMS 中部署应用程序的一些方法。每种方法的详细信息如下。

应用程序部署功能示例

方法名称	基础设施部署	AMI 或关键要素	应用程序安装
可变应用程序，AMS AMI			
手动部署应用程序	全栈 CT 或等级和局 CTs	AMS 提供的 AMI	提交访问管理 CT，手动安装应用程序。
UserData 使用应用程序代理（即 Chef、Puppet 等）部署应用程序			将 Provisioning CT 与安装应用程序代理和 script/agent 安装应用程序的 UserData 脚本一起使用。
UserData 无代理应用程序部署（即 Ansible、Salt SSH 等）			提交访问管理 CT，安装应用程序代理。使用应用程序部署工具部署应用程序。
可变应用程序、自定义 AMI			
自定义 AMI 应用程序部署（非 ASG）	全栈 CT 或等级和局 CTs	自定义 AMI。AMS AMI-> 使用应用程序部署工具代理进行自	应用程序部署工具（即Chef），利用代理，部署应用程序。

方法名称	基础设施部署	AMI 或关键要素	应用程序安装
		定义-> 创建 EC2 实例 (CT)-> 创建 AMI (CT)。	
AWS 数据库迁移服务 (DMS) 应用程序部署	AWS DMS 同步到现有 AMS 关系数据库堆栈。	自定义 AMI	客户或合作伙伴使用 AWS Database Migration Service；AMS 会在启动时验证 AMS 组件
工作负载摄取应用程序部署	合作伙伴迁移 instance/AMI 和客户启动的工作负载摄取 CT。		合作伙伴迁移实例，在客户 AMS 管理的 VPC 中创建 AMI；客户使用 Workload Ingest CT 在 AMS 中启动堆栈。 有关更多信息，请参阅 AMS 工作负载摄取 (WIGS)。
不可变的应用程序			
自定义 AMI 应用程序部署 (ASG)	全栈 CT 或等级和局 CTs	AMS AMI-> 自定义-> 创建 EC2 实例 (CT)-> 创建 AMI (CT)-> 创建 Auto Scaling 组。	Auto Scaling 使用自定义 AMI 部署应用程序 有关更多信息，请参阅 在 AMS 中分层和绑定应用程序部署。
可变或不可变的应用程序			

方法名称	基础设施部署	AMI 或关键要素	应用程序安装
自定义 CloudFormation 模板应用程序部署	CloudFormation 模板	AWS CloudFormation 模板-> customize/prepare 适用于 AMS-> 部署 | Ingestion | 来自 CloudFormation 模板的堆栈 | 创建 (ct-36cn2avfrrj9v)。	AMS 使用您的自定义 CloudFormation 模板将您的应用程序部署到您的账户，并验证应用程序部署。 有关更多信息，请参阅 AMS CloudFormation 摄取。
SQL 数据库导入	AMS 行动（其他 | 其他 CT）	本地 SQL 数据库-> .bak 文件-> AMS RDS SQL 数据库-> 管理 | 其他 | 其他 | 创建 (ct-1e1xtak34nx76) 进行导入。	AMS 将您的本地数据库导入到 AMS 管理的 RDS 数据库。有关更多信息，请参阅 将数据库 (DB) 导入到适用于微软 SQL Server 的 AMS RDS。
数据库迁移服务 (DMS)	AMS 行动（多个 CTs）	本地数据库-> DMS 复制实例-> DMS 复制子网组-> DMS 目标端点-> DMS 源端点-> DMS 复制任务。	AMS 将您的本地数据库导入到 AMS 托管的 S3 或目标 RDS 数据库。有关更多信息，请参阅 AWS Database Migration Service (AWS DMS)。
CodeDeploy 应用程序部署	CodeDeploy	应用程序-> CodeDeploy 应用程序-> CodeDeploy 部署组-> CodeDeploy 部署。	根据使用情况，就地部署或 Blue/Green 应用程序部署。有关详细信息，请参阅CodeDeploy 请求。

在 AMS 中规划您的应用程序部署

有关启用应用程序部署需要回答的一组推荐问题，请参阅附录：应用程序入职问卷。问题涵盖描述你的：

- 部署摘要
- 基础架构部署组件
- 应用程序托管平台
- 应用程序部署模型
- 应用程序依赖关系
- 适用于产品应用程序的 SSL 证书

AMS 工作负载摄取 (WIGS)

主题

- 迁移工作负载：Linux 和 Windows 的先决条件
- 迁移如何改变您的资源
- 迁移工作负载：标准流程
- 迁移工作负载：land CloudEndure ing zone (SALZ)
- AMS 工具账户（迁移工作负载）
- 迁移工作负载：Linux 摄取前验证
- 迁移工作负载：Windows 摄取前验证
- 工作负载摄取堆栈：创建

与 AMS 云迁移合作伙伴一起使用 AMS 工作负载摄取变更类型 (CT)，将您的现有工作负载迁移到 AMS 管理的 VPC 中。使用 AMS 工作负载摄取，您可以在将迁移的实例移至 AMS 后创建自定义 AMS AMI。本节介绍迁移合作伙伴和您自己为 AMS 工作负载摄取而采取的流程、先决条件和步骤。

> **Important**
>
> 操作系统必须受到 AMS 工作负载摄取的支持。有关支持的操作系统，请参见迁移工作负载：Linux 和 Windows 的先决条件。
> 每个工作负载和帐户都不一样。AMS 将与您合作，为取得成功的结果做好准备。

下图描述了 AMS 工作负载摄取过程。

迁移工作负载：Linux 和 Windows 的先决条件

在将本地实例的副本摄取到 AWS Managed Services (AMS) 之前，必须满足某些先决条件。这些是先决条件，包括那些在 Windows 和 Linux 操作系统之间存在差异的先决条件。

> **Note**
>
> 为了简化确定实例是否已准备好接收的过程，我们创建了适用于 Windows 和 Linux 的验证工具。这些工具可以下载并直接在您的本地服务器上运行，也可以在 AWS 中的 EC2 实例上运行。Linux pre-Wigs Validation.zip，Windows pre-Wigs Validation.zip。

在你开始之前，对于 Linux 和 Windows 来说：

- 执行全面的病毒扫描。
- 该实例必须具有`customer-mc-ec2-instance-profile`实例配置文件。
- 安装 Amazon S EC2 ystems Manager (SSM) 代理并确保 SSM 代理已启动并运行。
- 建议在根卷上至少有 10GB 的可用磁盘空间来运行 AMS 工作负载摄取 (WIGS)。从操作上讲，AMS 建议磁盘利用率低于 75%，并在磁盘利用率达到 85% 时发出警报。
- 与您的迁移合作伙伴一起确定摄取的时间范围。
- 自定义 AMI 作为 EC2 实例存在于目标生产 AMS 账户中（这是迁移合作伙伴的责任）。

> **Important**
>
> 操作系统必须受到 AMS 工作负载摄取的支持。
>
> - 支持以下操作系统：
>   - 微软 Windows 服务器：2008 R2、2012、2012 R2、2016、2019 和 2022
>   - Linux：亚马逊 Linux 2023、亚马逊 Linux 2 和亚马逊 Linux、CentOS 7.x、CentOS 6.5-6.10、Oracle Linux 7：次要版本 7.5 及以上、甲骨文 Linux 8：8.3 以下的次要版本、RHEL 8.x、RHEL 7.x、RHEL 6.5-6.10、SUSE Linux 企业服务器 15 和 SAP 特定版本 SUSE Linux 企业服务器 12、Ubuntu 18.04 SP3 SP4 SP5
> - 不支持 AMIs 以下内容：

- 亚马逊 Linux 2023 Minimal AMI。

> **Note**
>
> AMS API/CLI （ amscm 和 amsskms ）终端节点位于 AWS 弗吉尼亚北部区域。us-east-1根据您的身份验证设置方式以及您的账户和资源所在的 AWS 区域，您可能需要在发出命令--region us-east-1时进行添加。如果这是您的身份验证方法--profile saml，则可能还需要添加。

LINUX 必

在提交 WIGS RFC 之前，请遵守中列出的要求迁移工作负载：Linux 和 Windows 的先决条件并确保满足以下要求：

- 已安装最新的增强联网驱动程序；请参阅 Linux 上的增强联网。
- 与 AMS 组件冲突的第三方软件组件已被删除：
  - 防病毒客户端
  - Backup 客户端
  - 虚拟化软件（例如虚拟机工具或 Hyper-V 集成服务）
  - 访问管理软件（例如 SSSD、Centrify 或 PBIS）
- 确保 SSH 配置正确-这会暂时启用 SSH 的私钥身份验证。AMS 将其与我们的配置管理工具一起使用。使用以下命令：

```
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
```

- 确保 Yum 配置正确- RedHat 需要许可才能使用他们的 Yum 存储库。该实例需要通过卫星服务器或 RedHat 云服务器进行许可。如果需要许可，请使用以下链接之一：
  - 红帽卫星

  - 红帽云接入
- 如果您使用红帽卫星，WIGS 需要添加红帽软件集合 (RHSCL)。WIGS 系统使用 RHSCL 在系统上配置的任何内容旁边添加一个 Python3.6 解释器。要支持此解决方案，必须有以下存储库可用：
  - rhel-server-rhscl
  - rhel-server-releases-optional

Windows 先决条件

在提交 WIGS RFC 之前，请遵守中列出的要求迁移工作负载：Linux 和 Windows 的先决条件并确保满足以下要求：

- 已安装 Powershell 版本 3 或更高版本。
- AWS EC2 Config 安装在您要迁移的工作负载的实例上。
- 安装支持最新一代实例类型的 AWS 驱动程序：PV、ENA 和 NVMe。您可以使用以下链接中的信息：
  - 升级 Windows 实例上的半虚拟化驱动程序
  - Windows 上的增强联网功能
  - 适用于 Windows 实例的 AWS NVMe 驱动程序
  - 第 3 部分：升级 AWS NVMe 驱动程序
  - 第 5 部分：为裸机实例安装串行端口驱动程序
  - 第 6 部分：更新电源管理设置
- （可选但推荐）禁用关键服务-将关键应用程序服务（例如数据库）设置为禁用，但要确保记录所有更改，以便在应用程序验证阶段将其恢复到原始启动模式。
- （可选，但建议使用）使用准备好的实例创建故障安全 AMI：
  - 使用部署 | 高级堆栈组件 | AMI | 创建
  - 在创建过程中，添加标签 key=Name，value=application-id_ IngestReady
  - 等待 AMI 创建完成后再继续
- 与 AMS 组件冲突的第三方软件组件已被删除：
  - 防病毒客户端
  - Backup 客户端
  - 虚拟化软件（例如虚拟机工具或 Hyper-V 集成服务）

> **Note**
>
> 适用于 Windows 服务器的 End-of-Support迁移计划 (EMP) 包括无需任何重构即可将您的旧应用程序从 Windows Server 2003、2008 和 2008 R2 迁移到 AWS 上支持的新版本的工具。

迁移如何改变您的资源

本节中描述的接收 RFC 将在实例迁移到您的 AMS 账户后向其添加配置，这样 AMS 就可以对其进行管理。

添加的配置是特定于 AMS 的，如下所示。

对已@@ 摄取的 Linux 实例所做的更改：

- 已安装的软件：
  - Cloud Init：用于为 Jarvis Access 配置私钥。
  - 适用于所有支持的操作系统的 Python 3（脚本语言）（CentOS 6、RHEL 8、 OracleLinux 7 除外）。
  - AWS CloudFormation Python Helper 脚本：AWS CloudFormation 提供用于在亚马逊 EC2 实例上安装软件和启动服务的脚本。
  - AWS CLI：AWS CLI 是一款建立在适用于 Python 的 AWS 开发工具包 (Boto) 之上的开源工具，它提供用于与 AWS 服务交互的命令。
  - AWS SSM 代理：SSM 代理处理来自 Systems Manager 服务的请求，按照请求中的指定配置计算机。
  - AWS CloudWatch 日志代理：将日志发送到 CloudWatch。
  - AWS CodeDeploy：一种部署服务，可自动将应用程序部署到 Amazon EC2 实例、本地实例或无服务器 Lambda 函数。
  - 红宝石：必填项 CodeDeploy
  - 系统性能工具 (sysstat)：S ysstat 包含各种用于监控系统性能和使用活动的实用程序。
  - AD Bridge（前身为 PowerBroker 身份服务）：将非微软主机加入活动目录域。
  - 趋势科技趋势科技服务器深度安全防护系统客户端：防病毒软件。
- 已更改的软件：
  - 这些实例配置为使用 UTC 时区。

对已@@ 摄取的 Windows 实例所做的更改：

- 已安装的软件：
  - 适用于 Windows 的 AWS 工具 PowerShell：AWS 工具 PowerShell 允许开发人员和管理员在 PowerShell 脚本环境中管理他们的 AWS 服务和资源。
  - 趋势科技趋势科技服务器深度安全防护系统客户端：防病毒保护
  - AMS PowerShell 模块包含用于控制启动、Active Directory 加入、监控、安全和日志记录的 PowerShell 代码。
- 已更改的软件：
  - 服务器消息块 (SMB) 版本 1 已禁用。
  - Windows 远程管理 (WinRM) 已启用并配置为在端口 5986 上侦听。还创建了允许此入站端口的防火墙规则。
- 可能已安装或更改的软件：
  - 微软.Net Framework 4.5（开发者平台），如果检测到版本低于.Net Framework 4.5。
  - 对于 Windows 2012 和 Windows 2012R2，我们升级到 PowerShell 5.1。

迁移工作负载：标准流程

> **Note**
>
> 由于此过程需要两方，因此本节描述了双方的任务：AMS 云迁移合作伙伴（迁移合作伙伴）和应用程序所有者（您）。

1. 迁移伙伴，设置：
   a. 迁移合作伙伴向 AMS 提交 IAM 角色的服务请求，目的是迁移您的实例。有关提交服务请求的详细信息，请参阅服务请求示例。
   b. 迁移合作伙伴提交管理员访问请求。AMS 运营团队通过请求的 IAM 角色向迁移合作伙伴提供对您账户的访问权限。
2. 迁移合作伙伴，迁移单个工作负载：

a. 迁移合作伙伴使用 IAM AWS 实例配置文件（必须位于账户中）通过原生 Amazon EC2 或其他迁移工具，将您的非实例迁移到您 `customer-mc-ec2-instance-profile` AMS 账户中的子网。

b. 迁移合作伙伴提交 RFC，其中包含来自迁移伙伴迁移实例的部署 | Ingestion | 堆栈 | 创建 CT (ct-257p9zjk14ija)；有关创建和提交此 RFC 的详细信息，请参阅。工作负载摄取堆栈：创建

RFC 的执行输出返回实例 ID、IP 地址和 AMI ID。

迁移合作伙伴会为您提供在您的账户中创建的工作负载的实例 ID。

3. 您，访问并验证迁移：

a. 使用迁移合作伙伴为您提供的执行输出（AMI ID、实例 ID 和 IP 地址），提交访问 RFC 并登录新创建的 AMS 堆栈并验证您的应用程序是否正常运行。有关详细信息，请参阅请求实例访问权限。

b. 如果满意，您可以继续将启动的实例用作单层堆栈， and/or 使用 AMI 创建其他堆栈，包括 Auto Scaling 组。

c. 如果对迁移不满意，请提交服务请求并参考堆栈和 RFC IDs；AMS 将与您合作解决您的问题。

CloudEndure 接下来将介绍 landing zone 工作负载摄取过程。

迁移工作负载：land CloudEndure ing zone (SALZ)

本节提供有关为 (CE) 直接转换实例设置中间迁移单账户着陆区 (SALZ) 以供工作负载摄取 CloudEndure (WIGS) RFC 使用的信息。

要了解更多信息 CloudEndure，请参阅CloudEndure 迁移。

> **Note**
>
> 这是一种预定义的、经过安全保护的迁移 LZ 和模式。

先决条件：

- 客户的 AMS 账户

- AMS 账户与本地客户之间的网络和访问集成
- 一个 CloudEndure 账户
- 与 and/or CA CSDM 一起运行的 AMS 安全审查和签名的预批准工作流程（例如，滥用 IAM 用户永久证书为实例和安全组提供了权限） create/delete

> **Note**
>
> 本节描述了具体的准备和迁移过程。

准备：您和 AMS 操作员：

1. 使用管理 | 其他 | 其他 | 其他 | 将更改类型更新为 AMS 准备变更申请 (RFC)，以获取以下资源和更新。您可以单独提交 “其他 | 其他更新” RFCs，也可以提交一个。有关该 RFC/CT 的详细信息，请参阅 “其他 | 其他更新”，其中包含以下请求：

   a. 在您的 AMS VPC 中分配一个辅助 CIDR 块；一个将在迁移完成后移除的临时 CIDR 块。确保该区块不会与返回本地网络的任何现有路由发生冲突。例如，如果您的 AMS VPC CIDR 是 10.0.0.0/16，并且有一条返回您的本地网络 10.1.0.0/16 的路由，则临时辅助 CIDR 可能是 10.255.255.0/24。有关 AWS CIDR 区块的信息，请参阅 VPC 和子网大小调整。

   b. 在初始花园 AMS VPC 内创建一个新的私有子网。示例名称:`migration-temp-subnet`.

   c. 为仅包含本地 VPC 和 NAT (Internet) 路由的子网创建新的路由表，以避免在实例转换期间与源服务器发生冲突和可能的中断。确保允许互联网的出站流量用于补丁下载，以便可以下载和安装 AMS WIGS 先决条件。

   d. 更新您的托管 AD 安全组以允许入站和出站流量 to/from `migration-temp-subnet`。同时请求更新您的 EPS 负载均衡器 (ELB`mc-eps-McEpsElbPrivateSecurityGroup-M790XBZEEX74`) 安全组（例如：）以允许新的私有子网（即`migration-temp-subnet`）。如果所有三个 TCP 端口都不允许来自专用 CloudEndure (CE) 子网的流量，则 WIGS 摄取将失败。

   e. 最后，申请新 CloudEndure 的 IAM 策略和 IAM 用户。策略需要您正确的账号，`RunInstances`声明 IDs 中的子网应为：您的 <Customer Application Subnet (s) + Temp Migration Subnet>。

要查看 AMS 预先批准的 IAM CloudEndure 政策：解压 WIGS Cloud Endure 着陆区域示例文件并打开。`customer_cloud_endure_policy.json`

> **Note**
>
> 如果您想要更宽松的政策，请与您讨论您的需求， CloudArchitect/CSDM 并在必要时获得 AMS 安全审查和签署，然后再提交实施该政策的 RFC。

2. 您用 CloudEndure 于 AMS 工作负载摄取的准备步骤已完成，如果您的迁移合作伙伴已完成准备步骤，则可以执行迁移了。WIGS RFC 由您的迁移合作伙伴提交。

> **Note**
>
> IAM 用户密钥不会直接共享，但必须由 AMS 操作员在屏幕共享会话中键入 CloudEndure 管理控制台。

准备：迁移合作伙伴和 AMS 运营商：

1. 创建 CloudEndure 迁移项目。
   a. 在项目创建过程中，在屏幕共享会话中拥有 AMS 键入 IAM 用户证书。
   b. 在 “复制设置”-> “选择将在其中启动复制服务器的子customer-application-x网” 中，选择子网。
   c. 在 “复制设置”-> “选择要应用于复制服务器的安全组” 中，同时选择 Sentinel 安全组（仅限私有和 EgressAll）。
2. 为计算机（实例）定义转换选项。
   a. 子网：migration-temp-subnet。
   b. 安全组：两个 “Sentinel” 安全组（仅限私有和 EgressAll）。

      切换实例必须能够与 AMS 托管 AD 和 AWS 公共终端节点通信。
   c. 弹性 IP：无
   d. 公有 IP：否
   e. IAM 角色：customer-mc-ec双实例配置文件

IAM 角色必须允许 SSM 通信。最好使用 AMS 默认值。

f. 按照惯例设置标签。

迁移：迁移合作伙伴：

1. 在 AMS 上创建虚拟堆栈。您可以使用堆栈 ID 来访问堡垒。
2. 在源服务器上安装 CloudEndure (CE) 代理。有关详细信息，请参阅安装代理。
3. 在源服务器上创建本地管理员凭据。
4. 安排一个简短的直接转换窗口，准备就绪后单击 “直接转换”。这将完成迁移，并将用户重定向到目标 AWS 区域。
5. 请求堆栈管理员访问虚拟堆栈，请参阅管理员访问请求。
6. 使用您创建的本地管理员凭据登录堡垒，然后登录直接转换实例。
7. 创建故障安全 AMI。有关创建的详细信息 AMIs，请参阅 AMI 创建。
8. 为接收实例做好准备，请参阅。迁移工作负载：Linux 和 Windows 的先决条件
9. 对实例运行 WIGS RFC，请参阅。工作负载摄取堆栈：创建

AMS 工具账户（迁移工作负载）

您的多账户着陆区工具账户（使用 VPC）有助于加快迁移工作，提高您的安全地位，降低成本和复杂性，并标准化您的使用模式。

工具账户提供以下内容：

- 为系统集成商在生产工作负载之外访问复制实例提供了明确的边界。
- 允许您创建一个隔离的密室，以检查工作负载中是否存在恶意软件或未知的网络路由，然后再将其存入具有其他工作负载的帐户。
- 作为定义的帐户设置，它可以更快地为工作负载迁移做好准备和准备。
- 隔离的网络路由到来自本地-> 工具账户- CloudEndure > AMS 摄取的图像的安全流量。提取图像后，您可以通过 AMS 管理 | 高级堆栈组件 | AMI | 共享 (ct-1eiczxw8ihc18) RFC 将图像共享到目标账户。

高级架构图：

使用部署 | 托管着陆区 | 管理账户 | 创建工具账户（使用 VPC）更改类型 (ct-2j7q1hgf26x5c)，在多账户着陆区环境中快速部署工具账户并实例化工作负载摄取流程。参见管理账户，工具账户：创建（使用 VPC）。

> **Note**
>
> 我们建议有两个可用区 (AZs)，因为这是一个迁移中心。
> 默认情况下，AMS 在每个账户中创建以下两个安全组 (SGs)。确认这两个 SGs 都存在。如果他们不在场，请向 AMS 团队提交新的服务申请，请求他们。
>
> - SentinelDefaultSecurityGroupPrivateOnlyEgressAll
> - InitialGarden-SentinelDefaultSecurityGroupPrivateOnly
>
> 确保在私有子网中创建 CloudEndure 复制实例，那里有返回本地的路由。您可以通过确保私有子网的路由表中包含返回 TGW 的默认路由来确认这一点。但是，执行 CloudEndure 计算机切换操作应进入 “隔离” 私有子网，那里没有返回本地的路由，只允许 Internet 出站流量。确保在隔离子网中进行切换，以避免本地资源出现潜在问题，这一点至关重要。

先决条件：

1. 高级或高级支持级别。
2. 部署的 KMS 密钥 AMIs 的应用程序账户 IDs 。
3. 工具账户，如前所述。

AWS 应用程序迁移服务 (AWS MGN)

AWS 应用程序迁移服务 (AWS MGN) 可以通过在工具账户配置期间自动创建的 `AWSManagedServicesMigrationRole` IAM 角色在您的 MALZ Tools 账户中使用。您可以使用 AWS MGN 迁移在支持版本的 Windows 和 Linux 操作系统上运行的应用程序和数据库。

有关 AWS 区域 支持的大部分 up-to-date信息，请参阅AWS 区域服务列表。

如果 AWS MGN AWS 区域 目前不支持您的首选项，或者 AWS MGN 目前不支持运行应用程序的操作系统，请考虑改用工具账户中的CloudEndure 迁移。

正在请求 AWS MGN 初始化

AWS MGN 在首次使用前必须由 AMS 进行初始化。要申请新的 Tools 账户，请提交 Tools 账户中的管理 | 其他 | 其他 RFC，其中包含以下详细信息：

```
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.
```

AMS 成功完成 RFC 并在您的 Tools 账户中初始化 AWS MGN 后，您可以使用编辑默认模板`AWSManagedServicesMigrationRole`以满足您的要求。

启用对新 AMS 工具账户的访问权限

创建工具账户后，AMS 会为您提供账户 ID。下一步是配置对新账户的访问权限。执行以下步骤。

1. 将相应的 Active Directory 组更新为相应的帐户 IDs。

   AMS 创建的新账户将配置 ReadOnly 角色策略以及允许用户申报的角色。 RFCs

   Tools 账户还有一个额外的 IAM 角色和用户可用：

   - IAM 角色：`AWSManagedServicesMigrationRole`
   - IAM 用户：`customer_cloud_endure_user`
2. 请求策略和角色以允许服务集成团队成员设置更高级别的工具。

   导航到 AMS 控制台并归档以下内容 RFCs：

   a. 创建 KMS 密钥。使用创建 KMS 密钥 (auto) 或创建 KMS 密钥（托管自动化）。

      当您使用 KMS 加密提取的资源时，使用与其余多账户着陆区应用程序账户共享的单个 KMS 密钥可以为摄取的图像提供安全保护，这些图像可以在目标账户中解密。

   b. 共享 KMS 密钥。

      使用管理 | 高级堆栈组件 | KMS 密钥 | 共享（托管自动化）更改类型 (ct-05yb337abq3x5) 请求将新的 KMS 密钥共享给已提取的应用程序账户。 AMIs

最终账户设置的示例图：

AMS 预先批准的 IAM CloudEndure 政策示例

要查看 AMS 预先批准的 IAM CloudEndure 政策：解压 WIGS Cloud Endure 着陆区域示例文件并打开。`customer_cloud_endure_policy.json`

测试 AMS Tools 账户连接和 end-to-end设置

1. 首先在要复制到 AMS 的服务器上配置 CloudEndure 和安装 CloudEndure 代理。
2. 在中创建项目 CloudEndure。
3. 通过 secrets Manager 输入执行先决条件时共享的 AWS 凭据。
4. 在 “复制” 设置中：
   a. 选择 AMS “Sentinel” 安全组（仅限私有和 EgressAll），选择 “选择要应用于复制服务器的安全组” 选项。
   b. 为计算机（实例）定义转换选项。有关信息，请参阅步骤 5。切开
   c. 子网：私有子网。
5. 安全组：
   a. 同时选择 AMS “Sentinel” 安全组（仅限私有和 EgressAll）。
   b. 切换实例必须与 AMS 管理的 Active Directory (MAD) 和公共终端节点通信： AWS
      i. 弹性 IP：无
      ii. 公有 IP：没有
      iii. IAM 角色： customer-mc-ec双实例配置文件
   c. 按照您的内部标签惯例设置标签。
6. 在计算机上安装 CloudEndure 代理，然后在 EC2 控制台中查找要出现在您的 AMS 账户中的复制实例。

AMS 摄取过程：

AMS Tools 账户卫生

在账户中共享了 AMI 并且不再需要复制的实例之后，您需要进行清理：

- 实例 WIGs 摄取后：
  - 切换实例：至少在工作完成后，通过 AWS 控制台停止或终止此实例

- 摄取前 AMI 备份：在接入实例且本地实例终止后将其删除
- AMS 摄取的实例：在共享 AMI 后关闭堆栈或终止
- AMS-inge AMIs sted：与目标账户共享完成后删除

- 迁移结束清理：记录通过开发人员模式部署的资源，以确保定期进行清理，例如：
  - 安全组
  - 通过云形成创建的资源
  - 网络 ACK
  - 子网
  - VPC
  - 路由表
  - 角色
  - 用户和账户

大规模迁移-迁移工厂

请参阅 AWS CloudEndure 迁移工厂解决方案简介。

迁移工作负载：Linux 摄取前验证

您可以验证您的实例是否已准备好接入您的 AMS 账户。工作负载摄取 (WIGS) 摄取前验证会执行诸如操作系统类型、可用磁盘空间、是否存在冲突的第三方软件等检查。执行后，WIGS 摄取前验证会生成一个屏幕表，其中包含一个可选的日志文件。结果提供每项验证检查的 pass/fail 状态以及任何失败的原因。此外，您还可以根据需要自定义验证测试。

常见问题：

- 如何使用 Linux WIGS 摄取前验证？

  按照以下步骤下载和使用 AMS Linux WIGS 摄取前验证脚本：

  1. 下载包含验证脚本的 ZIP 文件

     Linux WIGS 摄取前验证 z ip 文件。
  2. 将附加的规则解压缩到您选择的目录中。
  3. 按照 readme.md 文件中的说明进行操作。
- Linux WIGS 摄取前验证会执行哪些验证？

AMS Linux WIGS 摄取前验证解决方案可验证以下内容：

1. 启动卷上至少有 5 GB 的可用空间。
2. AMS 支持该操作系统。
3. 该实例具有特定的实例配置文件。
4. 该实例不包含防病毒软件或虚拟化软件。
5. SSH 配置正确。
6. 该实例有权访问 Yum 存储库。
7. 已安装增强型网络驱动程序。
8. 该实例有 SSM 代理并且正在运行。

- 为什么支持自定义配置文件？

  这些脚本设计用于在本地物理服务器和 AWS EC2 实例上运行。但是，如上面的列表所示，有些测试在本地运行时会失败。例如，数据中心中的物理服务器将没有实例配置文件。在这种情况下，您可以编辑配置文件以跳过实例配置文件测试以避免混淆。

- 如何确保我拥有最新版本的脚本？

  Linux WIGS 预摄取验证解决方案的 up-to-date版本将在 “文档” 主页面的 “AMS Helper Files” 部分提供。

- 脚本是只读的吗？

  除了它生成的日志文件外，该脚本设计为只读模式，但应遵循最佳做法在非生产环境中运行该脚本。

- WIGS 摄取前验证是否可用于 Windows？

  是。它可在文档主页面的 “AMS 帮助文件” 部分中找到。

迁移工作负载：Windows 摄取前验证

您可以使用预验证器脚本来WIGs 验证您的实例是否已准备好接入您的 AMS 账户。工作负载摄取 (WIGS) 摄取前验证执行诸如操作系统类型、可用磁盘空间、是否存在冲突的第三方软件等检查。运行时，WIGS 摄取前验证会生成一个屏幕表和一个可选的日志文件。结果提供每项验证检查的 pass/fail 状态以及失败原因。此外，您还可以自定义验证测试。

常见问题：

- 如何使用 Windows WIGS 摄取前验证？

您可以通过 GUI 和 Web 浏览器运行验证，也可以使用 Windows PowerShell、SSM 运行命令或 SSM 会话管理器。

选项 1：从 GUI 和 Web 浏览器运行

要通过 GUI 和 Web 浏览器运行 Windows 预WIGs 验证，请执行以下操作：

1. 下载包含验证脚本的 ZIP 文件：

   Windows WIGS 摄取前验证 Z IP 文件。

2. 将附加的规则解压缩到您选择的目录中。
3. 按照 README.md 文件中的说明进行操作。

选项 2：从 Windows PowerShell、SSM 运行命令或 SSM 会话管理器运行

Windows 2016 及更高版本

1. 下载包含验证脚本的 ZIP 文件。

```
$DestinationFile = "$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. 从中删除现有文件`C:\Users\AppData\Local\Temp\AWSManagedServices.PreWigs.Validation`。

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

3. 调用脚本。

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

4. 将附件解压缩到您选择的目录中。

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

5. 以交互方式运行验证脚本并查看结果。

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

6. （可选）要捕获 “退出代码” 部分中列出的错误代码，请运行不带该RunWithoutExitCodes选项的脚本。请注意，此命令会终止活动会 PowerShell 话。

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

Windows 2012 R2 及更早版本

如果你运行的是 Windows Server 2012R2 或更低版本，则必须在下载 zip 文件之前设置 TLS。要设置 TLS，请完成以下步骤：

1. 下载包含验证脚本的 ZIP 文件。

```
$DestinationFile = "$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. 如果存在现有的验证文件，则将其删除。

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

3. 设置 TLS 版本。

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

4. 下载假发验证。

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

5. 将附加的规则解压缩到您选择的目录中。

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

6. 以交互方式运行验证脚本并查看结果。

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

7. （可选）要捕获 “退出代码” 部分中列出的错误代码，请运行不带该 RunWithoutExitCodes 选项的脚本。请注意，此命令会终止活动会 PowerShell 话。

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

> **Note**
>
> 您可以下载并运行 PowerShell 脚本。为此，请下载 pre-wigs-validation-powershell-scripts.zip。

- Windows WIGS 摄取前验证会执行哪些验证？

  AMS Windows WIGS 摄取前验证解决方案可验证以下内容：

  1. 启动卷上至少有 10 GB 的可用空间。
  2. AMS 支持该操作系统。
  3. 该实例具有特定的实例配置文件。
  4. 该实例不包含防病毒软件或虚拟化软件。
  5. 至少在一个网络适配器上启用了 DHCP。
  6. 该实例已为 Sysprep 做好了准备。
     - 对于 2008 年 R2 和 2012 Base 和 R2，Sysprep 会验证：
       - 有一个 unattend.xml 文件
       - sppnp.dll 文件（如果存在）未损坏
       - 操作系统尚未升级
       - 根据微软指南，Sysprep 的运行次数未超过最大值
     - 对于 2016 年及更高版本，将跳过上述所有检查，因为两者都不会给该操作系统造成问题
  7. Windows 管理工具 (WMI) 子系统运行正常。
  8. 已安装所需的驱动程序。
  9. SSM 代理已安装并正在运行。
  10. 系统会发出警告，以验证计算机是否因为 RDS 许可证配置而处于宽限期。

11所需的注册表项已正确设置。有关更多详细信息，请参阅摄取前验证 zip 文件中的自述文件。

- 为什么支持自定义配置文件？

  这些脚本设计用于在本地物理服务器和 AWS EC2 实例上运行。但是，如上面的列表所示，有些测试在本地运行时会失败。例如，数据中心中的物理服务器将没有实例配置文件。在这种情况下，您可以编辑配置文件以跳过实例配置文件测试以避免混淆。

- 如何确保我拥有最新版本的脚本？

  Windows WIGS 摄取前验证解决方案的 up-to-date版本将在文档主页面的 “AMS Helper Files” 部分下提供。

- 脚本是只读的吗？

  除了它生成的日志文件外，该脚本设计为只读模式，但应遵循最佳做法在非生产环境中运行该脚本。

- WIGS 摄取前验证是否可用于 Linux？

  是。Linux 版本于 2019 年 10 月 31 日推出。它可在文档主页面的 “AMS 帮助文件” 部分中找到。

工作负载摄取堆栈：创建

使用控制台将实例迁移到 AMS 堆栈

AMS 控制台中此更改类型的屏幕截图：

工作原理：

1. 导航到 “创建 RFC” 页面：在 AMS 控制台的左侧导航窗格中，单击RFCs打开 RFCs 列表页面，然后单击 “创建 R FC”。
2. 在默认的 “浏览更改类型” 视图中选择常用更改类型 (CT)，或者在 “按类别选择” 视图中选择 CT。
   - 按更改类型浏览：您可以单击 “快速创建” 区域中的常用 CT，立即打开 “运行 RFC” 页面。请注意，您不能使用快速创建来选择较旧的 CT 版本。

     要进行排序 CTs，请使用卡片视图或表格视图中的所有更改类型区域。在任一视图中，选择一个 CT，然后单击 “创建 RFC” 打开 “运行 RFC” 页面。如果适用，“创建 RFC” 按钮旁边会显示 “使用旧版本创建” 选项。
   - 按类别选择：选择类别、子类别、项目和操作，CT 详细信息框将打开，并显示 “使用旧版本创建” 选项（如果适用）。单击 “创建 RFC” 打开 “运行 RFC” 页面。

3. 在 “运行 RFC” 页面上，打开 CT 名称区域以查看 CT 详细信息框。必须填写主题（如果您在 “浏览更改类型” 视图中选择 CT，则会为您填写此主题）。打开其他配置区域以添加有关 RFC 的信息。

   在执行配置区域中，使用可用的下拉列表或输入所需参数的值。要配置可选的执行参数，请打开其他配置区域。
4. 完成后，单击 “运行”。如果没有错误，则会显示成功创建的 RFC 页面，其中包含已提交的 RFC 详细信息和初始运行输出。
5. 打开运行参数区域以查看您提交的配置。刷新页面以更新 RFC 的执行状态。（可选）取消 RFC 或使用页面顶部的选项创建一个 RFC 的副本。

> **Note**
>
> 如果 RFC 被拒绝，则执行输出将包含指向 Amazon CloudWatch 日志的链接。当不满足要求时，AMS Workload Ingest (WIGS) RFCs 将被拒绝；例如，如果在实例上检测到防病毒软件。CloudWatch 日志将包括有关未通过要求和补救措施的信息。

使用 CLI 将实例迁移到 AMS 堆栈

工作原理：

1. 使用 Inline Create（您发出包含所有 RFC 和执行参数的`create-rfc`命令）或模板创建（创建两个 JSON 文件，一个用于 RFC 参数，一个用于执行参数），然后以这两个文件作为输入发出`create-rfc`命令。这里描述了这两种方法。
2. 提交带有返回的 RFC ID 的 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 命令。

   监控 RFC: `aws amscm get-rfc --rfc-id` *`ID`* 命令。

要检查更改类型版本，请使用以下命令：

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

> **Note**
>
> 您可以将任何`CreateRfc`参数与任何 RFC 一起使用，无论它们是否属于变更类型的架构的一部分。例如，要在 RFC 状态更改时收到通知，请将此行添加到请求的 RFC 参数部分

（不是执行参数）。--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"有关所有 CreateRfc 参数的列表，请参阅《AMS 变更管理 API 参考》。

您可以使用 AMS CLI 从迁移到 AMS 账户的非 AMS 实例创建 AMS 实例。

> **Note**
>
> 请务必遵守先决条件；请参阅迁移工作负载：Linux 和 Windows 的先决条件。

要检查更改类型版本，请使用以下命令：

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

内联创建：

使用内联提供的执行参数（内联提供执行参数时使用转义引号）发出 create RFC 命令，然后提交返回的 RFC ID。例如，你可以用这样的东西替换内容：

```
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\"}"
```

模板创建：

1. 0将此更改类型的执行参数 JSON 架构输出到文件中；示例将其 MigrateStackParams命名为.json：

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

2. 修改并保存执行参数 JSON 文件。例如，你可以用这样的东西替换内容：

   ```
   {
   ```

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

3. 输出 RFC 模板 JSON 文件；示例将其命名为 MigrateStackRfc .json：

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

4. 修改并保存 MigrateStackRfc .json 文件。例如，你可以用这样的东西替换内容：

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

5. 创建 RFC，指定 MigrateStackRfc 文件和 MigrateStackParams 文件：

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

   您在响应中收到新 RFC 的 ID，并可以使用它来提交和监控 RFC。在您提交之前，RFC 仍处于编辑状态且无法启动。

   新实例显示在相关 VPC 的应用程序所有者账户的实例列表中。

6. 成功完成 RFC 后，通知应用程序所有者，以便他或她可以登录新实例并验证工作负载是否正常运行。

> **Note**
>
> 如果 RFC 被拒绝，则执行输出将包含指向 Amazon CloudWatch 日志的链接。当不满足要求时，AMS Workload Ingest (WIGS) RFCs 将被拒绝；例如，如果在实例上检测到防病毒软件。CloudWatch 日志将包括有关未通过要求和补救措施的信息。

提示

> **Note**
>
> 请务必遵守先决条件；请参阅迁移工作负载：Linux 和 Windows 的先决条件。

> **Note**
>
> 如果正在迁移的实例上的标签与 RFC 中提供的标签具有相同的密钥，则 RFC 将失败。

> **Note**
>
> 您最多可以指定四个目标 IDs、端口和可用区。

> **Note**
>
> 如果 RFC 被拒绝，则执行输出将包含指向 Amazon CloudWatch 日志的链接。当不满足要求时，AMS Workload Ingest (WIGS) RFCs 将被拒绝；例如，如果在实例上检测到防病毒软件。CloudWatch 日志将包括有关未通过要求和补救措施的信息。

> **Note**
>
> 如果 RFC 被拒绝，则执行输出将包含指向 Amazon CloudWatch 日志的链接。当不满足要求时，AMS Workload Ingest (WIGS) RFCs 将被拒绝；例如，如果在实例上检测到防病毒软件。CloudWatch 日志将包括有关未通过要求和补救措施的信息。

如果需要，请参阅工作负载接入 (WIGS) 失败。

AMS CloudFormation 摄取

AMS AWS CloudFormation 采集变更类型 (CT) 允许您使用现有 CloudFormation 模板（经过一些修改）在 AMS 托管的 VPC 中部署自定义堆栈。

主题

AMS CloudFormation 摄取过程涉及以下内容：

- 准备您的自定义 CloudFormation 模板并将其上传到 S3 存储桶，或者在创建 RFC 时提供内联模板。如果您使用的是带有预签名 URL 的 S3 存储桶；有关更多信息，请参阅 pr esign。
- 在 RF CloudFormation C 中向 AMS 提交采集更改类型。有关 CFN 采集更改类型演练，请参阅。创建 CloudFormation 采集堆栈有关 CFN 采集示例，请参阅。CloudFormation 收录：示例
- 创建堆栈后，您可以对其进行更新，并修复其中的偏差；此外，如果更新失败，您可以明确批准和实施更新。本节将介绍所有这些过程。

  有关 CFN 漂移检测的信息，请参阅新增- CloudFormation 偏移检测。

> **Note**
>
> - 此更改类型现在有 2.0 版。版本 2.0 是自动执行的；不是手动执行的。这使得 CT 的执行速度更快。此版本引入了两个新参数：CloudFormationTemplate，它允许您将自定义 CloudFormation 模板粘贴到 RFC 中；以及VpcId允许在 AMS 多账户 landing zone 中使用 CloudFormation 摄取。
> - 版本 1.0 是手动更改类型。这意味着 AMS 操作员必须采取一些措施才能成功完成变更类型。至少需要进行审查。此版本还要求 CloudFormationTemplateS3Endpoint 参数值为预签名 URL。

CloudFormation 摄取指南、最佳实践和限制

AMS 需要处理您的 CloudFormation 模板，需要遵守一些准则和限制。

指南

要减少执行载 CloudFormation 入时 CloudFormation 出现的错误，请遵循以下准则：

- 不要在模板中嵌入凭据或其他敏感信息- CloudFormation 模板在 CloudFormation 控制台中可见，因此您不想在模板中嵌入凭据或敏感数据。模板不能包含敏感信息。只有当您使用 AWS Secrets Manager 作为值时，才允许使用以下资源：
  - `AWS::RDS::DBInstance` - [MasterUserPassword,TdeCredentialPassword]
  - `AWS::RDS::DBCluster` - [MasterUserPassword]
  - `AWS::ElastiCache::ReplicationGroup` - [AuthToken]

  > **Note**
  >
  > 有关在资源属性中使用 S AWS ecrets Manager 密钥的信息，请参阅如何使用 AWS CloudFormation 模板创建和检索在 Secrets Manager 中管理的 AWS 密钥和使用动态引用指定模板值。

- 使用 Amazon RDS 快照创建 RDS 数据库实例 — 这样您就不必提供 MasterUserPassword。
- 如果您提交的模板包含 IAM 实例配置文件，则必须以 “客户” 为前缀。例如，使用名为 “example-instance-profile” 的实例配置文件会导致失败。相反，请使用名为 “customer-example-instance-profile” 的实例配置文件。
- 请勿在 **`AWS::EC2::Instance`**-[UserData] 中包含任何敏感数据。 UserData 不应包含密码、API 密钥或任何其他敏感数据。此类数据可以加密并存储在 S3 存储桶中，然后使用下载到实例上 UserData。
- 支持使用 CloudFormation 模板创建 IAM 策略有一些限制 — IAM 策略必须经过 AMS 的审核和批准 SecOps。目前，我们仅支持使用包含预先批准权限的内联策略部署 IAM 角色。在其他情况下，无法使用 CloudFormation 模板创建 IAM 策略，因为这会覆盖 AMS SecOps 流程。
- KeyPairs 不支持 SSH — Amazon EC2 实例必须通过 AMS 访问管理系统进行访问。AMS RFC 流程会对您进行身份验证。您不能在 CloudFormation 模板中包含 SSH 密钥对，因为您无权创建 SSH 密钥对和覆盖 AMS 访问管理模型。
- 安全组的入口规则受到限制 — 您不能将源 CIDR 范围设置为 0.0.0.0/0，也不能将可公开路由的地址空间设置为 80 或 443，TCP 端口不是 80 或 443。
- 编写 CloudFormation 资源模板时请遵循 CloudFormation 指导方针 — 参阅资源的《AWS CloudFormation 用户指南》，确保为该资源使用正确的数据 type/property 名称。例如，AWS::EC2::Instance 资源中 SecurityGroupIds 属性的数据类型是 “字符串值列表”，因此 ["sg-aaaaaaaa"] 没问题（带方括号），但是 “sg-aaaaaaaa” 不是（不带方括号）。

有关更多信息，请参阅 AWS 资源和属性类型参考。

- 将@@ 您的自定义 CloudFormation 模板配置为使用 AMS CloudFormation 收录 CT 中定义的参数 — 当您将 CloudFormation 模板配置为使用 AMS 收录 C CloudFormation T 中定义的参数时，您可以使用 “管理 | 自定义堆栈 | 来自模板的堆栈 | 来自 CloudFormation 模板的堆栈 | 更新 CT” (ct-361tlo1k7339x) 在 CT 输入中使用更改的参数值来重复使用 CloudFormation 模板来创建类似的堆栈。有关示例，请参阅CloudFormation 摄取示例：定义资源。
- 带有预签名 URL 的 Amazon S3 存储桶终端节点不能过期 — 如果您使用带有预签名 URL 的 Amazon S3 存储桶终端节点，请确认预签名的 Amazon S3 URL 未过期。使用已过期的预签名 Amazon S3 存储桶 URL 提交的提 CloudFormation 取 RFC 被拒绝。
- 等待条件需要信号逻辑-等待条件用于协调堆栈资源的创建与堆栈创建外部的配置操作。如果您在模板中使用 CloudFormation Wait Condition 资源，则等待成功信号，如果未发出成功信号的数量，则会将堆栈创建标记为失败。如果您使用等待条件资源，则需要有信号的逻辑。有关更多信息，请参阅在模板中创建等待条件。

最佳实践

以下是您可以使用 AMS CloudFormation 采集流程迁移资源的一些最佳实践：

- 在一个 CT 中提交 IAM 和其他与策略相关的资源 — 如果您可以使用 CloudFormation Ingest CTs 等自动化方式部署 IAM 角色，我们建议您这样做。在其他情况下，AMS 建议您收集所有 IAM 或其他策略相关资源，并将其提交到单个管理 | 其他 | 其他 | 创建更改类型 (ct-1e1xtak34nx76) 中。例如，合并所需的所有 IAM 角色、IAM Amazon EC2 实例配置文件、现有 IAM 角色的 IAM 策略更新、Amazon S3 存储桶策略、Amazon S SNS/Amazon QS 策略等，然后提交 ct-1e1xtak34nx76 RFC，以便在未来的采集模板中可以简单地引用这些先前存在的资源。 CloudFormation
- EC2 实例已启动并成功加入域 — 这是一种最佳做法，这是自动完成的。为了确保通过 CloudFormation 采集堆栈启动的 Amazon EC2 实例被引导并成功加入域，AMS UpdatePolicy 为一个 Auto Scaling 组资源提供了 CreationPolicy 和一个（也就是说，如果这些策略尚不存在）。
- 必须指定 Amazon RDS 数据库实例参数 — 通过载 CloudFormation 入创建 Amazon RDS 数据库时，必须指定该`DBSnapshotIdentifier`参数才能从之前的数据库快照还原。这是必需的，因为 CloudFormation 载入当前不处理敏感数据。

有关如何使用模板进行 AMS CloudFormation CloudFormation 模板摄取的示例，请参阅CloudFormation 收录：示例。

模板验证

您可以先对 CloudFormation 模板进行自我验证，然后再将其提交给 AMS。

提交给 AMS CloudFormation ingest 的模板经过验证，以确保可以安全地部署在 AMS 账户中。验证过程会检查以下内容：

- 支持的资源-仅使用 AMS CloudFormation 摄取支持的资源。有关更多信息，请参阅 支持的资源。
- 支持 AMIs — 模板中的 AMI 是 AMS 支持的 AMI。有关 AMS 的信息 AMIs，请参阅AMS Amazon 机器映像 (AMIs)。
- AMS 共享服务子网 — 模板不尝试将资源启动到 AMS 共享服务子网。
- 资源策略 — 没有过于宽松的资源策略，例如公开可读或可写的 S3 存储桶策略。AMS 不允许公开读取或可写入的 S3 存储桶。 AWS 账户

使用 CloudFormation Linter 进行验证

在将 CloudFormation 模板提交给 AMS 之前，您可以使用 CloudFormation Linter 工具对其进行自我验证。

CloudFormation Linter 工具是验证 CloudFormation 模板的最佳方式，因为它可以验证 resource/property 名称、数据类型和函数。有关更多信息，请参阅 aws-cfn-python-lint cloudformation/。

前面显示的模板的 CloudFormation Linter 输出如下所示：

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

为了协助 CloudFormation 模板的离线验证，AMS 为 CloudFormation Linter 工具开发了一套可插拔的自定义验证规则。它们位于 AMS 控制台的开发者资源页面上。

请按照以下步骤使用 CloudFormation 摄取前验证脚本：

1. 安装 CloudFormation Linter 工具。有关安装说明，请参阅 aws-cloudformation/cfn-lint。
2. 下载包含验证脚本的.zip 文件：

   CFN Lint 自定义规则。
3. 将附加的规则解压缩到您选择的目录中。
4. 通过运行以下命令来验证您的 CloudFormation 模板：

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

CloudFormation 采集堆栈：CFN 验证器示例

这些示例可以帮助您准备模板以成功收录。

格式验证

验证模板是否包含 “资源” 部分，并且在其下定义的所有资源都具有 “类型” 值。

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

验证是否允许使用模板的根密钥。允许的根密钥是：

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

手动托管自动化验证

如果模板包含以下资源，则自动验证将失败，您需要手动审核。

从安全角度来看，所示的策略是高风险区域。例如，允许除特定用户或组之外的任何人创建对象或写入权限的 S3 存储桶策略极其危险。因此，我们会根据内容验证政策并批准或拒绝，这些政策无法自动创建。我们正在研究解决此问题的可能方法。

我们目前没有针对以下资源的自动验证。

```
[
    "S3::BucketPolicy",
    "SNS::TopicPolicy",
    "SQS::QueuePolicy"
]
```

参数验证

验证模板参数是否没有提供值；它必须具有默认值。

资源属性验证

必填属性检查：某些资源类型必须存在某些属性。

- “VPCOptions” 必须存在于 `AWS::OpenSearch::Domain`
- “CludsterSubnetGroupName” 必须存在于 `AWS::Redshift::Cluster`

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

不允许的属性检查：某些资源类型的某些属性必须*不*存在。

- “SecretString” 不能存在于 “AWS::SecretsManager::Secret”
- “MongoDbSettings” 不能存在于 “AWS::DMS::Endpoint”

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

```
}
```

SSM 参数检查：对于以下列表中的属性，必须通过 Secrets Manager 或 Systems Manager 参数存储（安全字符串参数）指定值：

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

某些属性必须符合某些模式；例如，IAM 实例配置文件名称不得以 AMS 保留前缀开头，并且属性值必须与特定的正则表达式匹配，如下所示：

```
{
    "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).+"
      ]
    }
}
```

资源验证

只能在模板中指定列入许可名单的资源；这些资源在中进行了介绍。支持的资源

由于修补限制，不允许将 EC2 堆栈和 Auto Scaling 组 (ASGs) 放在同一个堆栈中。

安全组入口规则验证

- 对于来自 CFN Ingest Create 或 Stack Update CT 更改类型的请求：
  - 如果 (`IpProtocol`是 tcp 或 6) AND（端口为 80 或 443），则该`CidrIP`值没有限制
  - 否则，`CidrIP`不能是 0.0.0.0/0
- 对于来自服务目录（服务目录产品）的请求：
  - 除了 CFN Ingest Create 或 Stack Update CT 更改类型验证外，协议所在的端口`ip_protocols`只能通过以下方式访问：`management_portsallowed_cidrs`

```
{
      "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"]
 }
```

限制

AMS CloudFormation 采集流程目前不支持以下特性和功能。

- YAML — 不支持。仅支持基于 JSON 的 CloudFormation模板。
- 嵌套堆栈 — 取而代之的是，使用单个模板来构建您的应用程序基础架构。或者，您可以利用跨堆栈引用将资源分隔到多个堆栈中，其中一个资源依赖于另一个资源。有关更多信息，请参阅演练：请参阅其他 AWS CloudFormation 堆栈中的资源输出。
- CloudFormation 堆栈集-由于存在安全隐患，不支持。
- 使用 CloudFormation 模板创建 IAM 资源-由于存在安全隐患，仅支持 IAM 角色。
- 敏感数据-不支持。请勿在模板或参数值中包含敏感数据。如果您需要引用敏感数据，请使用 Secrets Manager 来存储和检索这些值。有关在资源属性中使用 AWS Secrets Managers 密钥的信息，请参阅如何使用 AWS CloudFormation 模板创建和检索在 AWS Secrets Manager 中管理的密钥和使用动态引用指定模板值。

支持的资源

AMS CloudFormation 采集流程支持以下 AWS 资源。

CloudFormation 载入堆栈：支持的资源

AMS 工作负载摄取必须支持实例操作系统。仅支持此处列出的那些 AWS 资源。

- Amazon API Gateway
  - AWS::ApiGateway::Account
  - AWS::ApiGateway::ApiKey
  - AWS::ApiGateway::Authorizer
  - AWS::ApiGateway::BasePath映射
  - 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::UsagePlan钥匙
- AWS::ApiGateway::VpcLink

- 亚马逊 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
- AWS 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 日志
  - 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::UserPool客户
  - AWS::Cognito::UserPool域

- AWS::Cognito::UserPool群组
- AWS::Cognito::UserPoolIdentityProvider
- AWS::Cognito::UserPoolResourceServer
- AWS::Cognito::UserPoolRiskConfigurationAttachment
- AWS::Cognito::UserPoolUICustomization附件
- AWS::Cognito::UserPoolUser
- AWS::Cognito::UserPoolUserToGroupAttachment

- Amazon DocumentDB
  - AWS::Doc数据库:: DBCluster
  - AWS::Doc数据库:: DBCluster ParameterGroup
  - AWS::Doc数据库:: DBInstance
  - AWS::Doc数据库:: DBSubnet 群组
- 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::NetworkInterface附件
  - AWS::EC2::SecurityGroup
  - AWS::EC2::SecurityGroup入口
  - AWS::EC2::SecurityGroup出口
  - AWS::EC2::LaunchTemplate
- AWS Batch
  - AWS::Batch::ComputeEnvironment
  - AWS::Batch::JobDefinition
  - AWS::Batch::JobQueue

- 亚马逊弹性容器注册表 (ECR)
  - AWS::ECR::Repository
- 亚马逊弹性容器服务 (ECS) (Fargate)
  - AWS::ECS::CapacityProvider
  - AWS::ECS::Cluster
  - AWS::ECS::PrimaryTaskSet
  - AWS::ECS::Service
  - AWS::ECS::TaskDefinition
  - AWS::ECS::TaskSet
- 亚马逊 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::SecurityGroup入口
  - AWS::ElastiCache::SubnetGroup
- Amazon EventBridge
  - AWS::Events::EventBus
  - AWS::Events::EventBus政策
  - 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:: DBCluster ParameterGroup
  - AWS:: RDS:: DBInstance
  - AWS:: RDS:: 群组 DBParameter
  - AWS:: RDS:: 群组 DBSubnet
  - AWS::RDS::EventSubscription
  - AWS::RDS::OptionGroup
- Amazon Route 53
  - AWS::Route53::HealthCheck
  - AWS::Route53::HostedZone
  - AWS::Route53::RecordSet
  - AWS::Route53::RecordSet群组
  - AWS::Route53Resolver::ResolverRule
  - AWS::Route53Resolver::ResolverRule协会

- Amazon S3

  - AWS::S3::Bucket
- 亚马逊 Sagemaker
  - AWS::SageMaker::CodeRepository
  - AWS::SageMaker::Endpoint
  - AWS::SageMaker::EndpointConfig
  - AWS::SageMaker::Model
  - AWS::SageMaker::NotebookInstance
  - AWS::SageMaker::NotebookInstanceLifecycleConfig
  - AWS::SageMaker::Workteam
- 亚马逊简单电子邮件服务 (SES) Simple Service
  - 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
- 应用程序 AutoScaling
  - AWS::ApplicationAutoScaling::ScalableTarget
  - AWS::ApplicationAutoScaling::ScalingPolicy
- Amazon EC2 AutoScaling
  - AWS::AutoScaling::AutoScaling群组

  - AWS::AutoScaling::LaunchConfiguration
  - AWS::AutoScaling::LifecycleHook
  - AWS::AutoScaling::ScalingPolicy
  - AWS::AutoScaling::ScheduledAction
- AWS Certificate Manager
  - AWS::CertificateManager::Certificate
- AWS CloudFormation
  - AWS::CloudFormation::CustomResource
  - AWS::CloudFormation::Designer
  - AWS::CloudFormation::WaitCondition
  - AWS::CloudFormation::WaitCondition手柄
- 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::CustomAction类型
  - AWS::CodePipeline::Pipeline
  - AWS::CodePipeline::Webhook
- AWS 数据库迁移服务 (DMS)
  - AWS::DMS::Certificate
  - AWS::DMS::Endpoint
  - AWS::DMS::EventSubscription
  - AWS::DMS::ReplicationInstance

  - AWS::DMS::ReplicationSubnet群组
  - AWS::DMS::ReplicationTask

  不允许使用 AWS::DMS::Endpoint 资源中的 MongoDbSettings 属性。

  以下属性仅在由 AWS Secrets Manager 解析后才允许使用： CertificatePem 以及 AWS::DMS::Certificate 资源中的属性以及 AWS::DMS::Endpoint 资源中的密码 CertificateWallet 属性。
- AWS Elastic Load Balancing — 应用程序负载均衡器/网络负载均衡器 Network Load Balan
  - AWS::ElasticLoadBalancingV2::Listener
  - AWS::ElasticLoadBalancingV2::ListenerCertificate
  - AWS::ElasticLoadBalancingV2::ListenerRule
  - AWS::ElasticLoadBalancingV2::LoadBalancer
  - AWS::ElasticLoadBalancingV2::TargetGroup
- AWS 弹性负载平衡——经典负载均衡器
  - 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
- 适用于 Apache Kafka 的 AWS 托管流媒体 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 密钥管理服务 (KMS) Service
  - AWS::KMS::Key
  - AWS::KMS::Alias
- AWS Lake Formation
  - AWS::LakeFormation::DataLake设置
  - AWS::LakeFormation::Permissions
  - AWS::LakeFormation::Resource
- AWS Lambda
  - AWS::Lambda::Alias
  - AWS::Lambda::EventInvokeConfig
  - AWS::Lambda::EventSource映射
  - AWS::Lambda::Function
  - AWS::Lambda::LayerVersion
  - AWS::Lambda::LayerVersion权限
  - AWS::Lambda::Permission
  - AWS::Lambda::Version
- Amazon Redshift
  - AWS::Redshift::Cluster
  - AWS::Redshift::ClusterParameter群组
  - AWS::Redshift::ClusterSubnet群组
- AWS Secrets Manager
  - AWS::SecretsManager::ResourcePolicy

- AWS::SecretsManager::RotationSchedule
- AWS::SecretsManager::Secret
- AWS::SecretsManager::SecretTarget附件

- AWS Security Hub
  - AWS::SecurityHub::Hub
- AWS Step Functions
  - AWS::StepFunctions::Activity
  - AWS::StepFunctions::StateMachine
- AWS Systems Manager (SSM)
  - AWS::SSM::Parameter
- Amazon S CloudWatch ynthetics
  - 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 区域版
  - AWS::WAFRegional::ByteMatchSet
  - AWS::WAFRegional::GeoMatchSet
  - AWS:WAFRegional::: IPSet
  - AWS::WAFRegional::RateBased规则
  - 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

CloudFormation 收录：示例

在此处可以找到一些详细的示例，说明如何使用带有 CloudFormation 模板更改类型的创建堆栈。

要下载一组示例 CloudFormation 模板 AWS 区域，请参阅示例模板。

有关 CloudFormation 资源的参考信息，请参阅 AWS 资源和属性类型参考。但是，AMS 支持的资源集较少，如中所述AMS CloudFormation 摄取。

> **Note**
>
> AMS 建议您收集所有 IAM 或其他策略相关资源，并将其提交到单个管理 | 其他 | 其他 | 创建变更类型 (ct-1e1xtak34nx76) 中。例如，合并所有需要的 IAM 角色、IAM 实例配置文件、现有 IAM 角色的 IAM 策略更新、S3 存储桶 SNS/SQS 策略、策略等，然后提交 ct-1e1xtak34nx76 RFC，以便可以在未来的 CFN Ingest 模板中引用这些先前存在的资源。

主题

- CloudFormation 摄取示例：定义资源
- CloudFormation 采集示例：3 层 Web 应用程序

CloudFormation 摄取示例：定义资源

使用 AMS CloudFormation 采集时，您可以自定义 CloudFormation 模板并将其提交给采集更改类型 (ct-36cn2avfrrj9v) 的 CloudFormation RFC 中的 AMS。要创建可以多次重复使用的 CloudFormation 模板，您可以将堆栈配置参数添加到 CloudFormation 采集更改类型执行输入中，而不是在模板中对其进行硬编码。 CloudFormation 最大的好处是你可以重复使用该模板。

AMS CloudFormation 采集更改类型输入架构允许您在 CloudFormation 模板中选择多达 60 个参数并提供自定义值。

此示例说明如何将可用于各种 CloudFormation 模板的资源属性定义为 AMS CloudFormation 采集 CT 中的参数。本节中的示例专门展示了 SNS 主题的用法。

主题

示例 1：对 CloudFormation SNSTopic 资源**TopicName**属性进行硬编码

在此示例中，您在 CloudFormation 模板中对 CloudFormation SNSTopic 资源`TopicName`属性进行了硬编码。请注意，该`Parameters`部分为空。

要拥有一个允许您无需创建新 CloudFormation 模板即可更改新堆栈 SNSTopic 名称值的 CloudFormation 模板，您可以使用载入更改类型的 AMS `Parameters` 部分进行该配置。CloudFormation 通过执行此操作，您可以稍后使用相同的 CloudFormation 模板来创建具有不同`SNSTopic`名称的新堆栈。

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Description" : "My SNS Topic",
  "Parameters" : {
  },
  "Resources" : {
    "SNSTopic" : {
      "Type" : "AWS::SNS::Topic",
```

```
      "Properties" : {
        "TopicName" : "MyTopicName"
      }
    }
  }
}
```

示例 2：使用 SNSTopic 资源引用 AMS 更改类型中的参数

在此示例中，您使用 CloudFormation 模板中定义的`SNSTopic`资源`TopicName`属性来引用 AMS 更改类型`Parameter`中的。

```
{
  "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"}
      }
    }
  }
}
```

示例 3：通过提交具有 AMS 采集更改类型的 JSON 执行参数文件来创建 SNS 主题

在此示例中，您提交了一个 JSON 执行参数文件，其中包含创建 SNS 主题的 AMS 收录 CT。`TopicName`必须按照本示例中显示的可修改方式在 CloudFormation 模板中定义 SNS 主题。

```
{
  "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
}
```

示例 4：提交引用相同 CloudFormation 模板的新变更类型

此 JSON 示例在不更改 CloudFormation 模板的情况下更改 SNS TopicName 值。相反，您可以提交引用相同 CFN CloudFormation 模板的新部署 | Ingestion | Stack from Template | Create 更改类型。

```
{
  "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
}
```

示例 5：使用 CloudFormation 模板中的默认参数值

在此示例中，之所以创建 SNS TopicName = MyTopicName '，是因为Parameters执行参数中未提供任何TopicName值。如果您不提供Parameters定义，则使用 CloudFormation 模板中的默认参数值。

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

```
  ],
  "TimeoutInMinutes": 60
}
```

CloudFormation 采集示例：3 层 Web 应用程序

为标准 3 层 Web 应用程序采集 CloudFormation 模板。

这包括应用程序负载均衡器、应用程序负载均衡器目标组、Auto Scaling 组、Auto Scaling 组启动模板、带有 MySQL 数据库的亚马逊关系数据库服务（适用于 SQL Server 的 RDS）、 AWS SSM 参数存储和 S AWS ecrets Manager。请等待 30-60 分钟来完成此示例。

先决条件

- 使用 Secrets Manager 创建包含用户名和密码以及相应值的 AWS 密钥。您可以参考包含密钥名称的示例 JSON 模板（zip 文件）`ams-shared/myapp/dev/dbsecrets`，并将其替换为您的密钥名称。有关将 S AWS ecrets Manager 与 AMS 配合使用的信息，请参阅将 S AWS ecrets Manager 与 AMS 资源配合使用。
- 在 AWS SSM 参数存储 (PS) 中设置必需的参数。在此示例中，私有子网`VPCId`和`Subnet-Id`公有子网的和存储在 SSM PS 中，路径如`/app/DemoApp/PublicSubnet1a`、`PublicSubnet1c`、`PrivateSubnet1a`、`PrivateSubnet1c`和。`VPCCidr`根据需要更新路径、参数名称和值。
- 创建一个 IAM Amazon EC2 实例角色，该角色具有对 S AWS ecrets Manager 和 SSM 参数存储路径的读取权限（这些示例中创建和使用的 IAM 角色是`customer-ec2_secrets_manager_instance_profile`）。如果您创建 IAM 标准策略，例如实例配置文件角色，则角色名称必须以开头。`customer-`要创建新的 IAM 角色，（你可以给它`customer-ec2_secrets_manager_instance_profile`起名字或其他名字）使用 AMS 更改类型管理 | 应用程序 | IAM 实例配置文件 | 创建 (ct-0ixp4ch2tiu04) CT，然后附加所需的策略。您可以在 IAM 控制台中查看 AMS AWS IAM 标准策略`customer_secrets_manager_policy`和`customer_systemsmanager_parameterstore_polic`以便按原样使用或作为参考。

摄取标准 3 层 Web 应用程序的 CloudFormation 模板

1. 将随附的示例 CloudFormation JSON 模板作为 zip 文件（3-tier-cfn-ingest .zip）上传到 S3 存储桶，然后生成签名的 S3 网址以在 CFN Ingest RFC 中使用。有关更多信息，请参阅 presign。当你通过 AMS 控制台提交 RFC 时，CFN 模板也可以 copy/pasted 放在 CFN Ingest RFC 中。

2. 通过 AMS 控制台 CloudFormation 或 AMS CLI 创建 Ingest RFC（部署 | Ingestion | 来自 CloudFormation 模板的堆栈 | 创建 (ct-36cn2avfrrj9v)）。 CloudFormation 采集自动化流程会对 CloudFormation 模板进行验证，以确保该模板具有 AMS 支持的有效资源并符合安全标准。

   - 使用控制台-对于更改类型，从 “ CloudFormation 模板”-> “创建” 中选择 “部署”-> “接收”-> “堆栈”，然后添加以下参数作为示例（请注意，“多” 的默认值AZDatabase为 false）：

     ```
     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"
     ```

   - 使用 AWS CLI -有关 RFCs 使用创建的详细信息 AWS CLI，请参阅创建 RFCs。例如，运行以下命令：

     ```
     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
     ```

   在 CloudFormation RFC 执行输出中找到 Application Load Balancer 网址以访问该网站。有关访问资源的信息，请参阅访问实例。

创建 CloudFormation 采集堆栈

使用控制台创建 CloudFormation 采集堆栈

使用控制台创建 CloudFormation 采集堆栈

1. 导航到 “创建 RFC” 页面：在 AMS 控制台的左侧导航窗格中，单击RFCs打开 RFCs 列表页面，然后单击 “创建 R FC”。
2. 在默认的 “浏览更改类型” 视图中选择常用更改类型 (CT)，或者在 “按类别选择” 视图中选择 CT。
   - 按更改类型浏览：您可以单击 “快速创建” 区域中的常用 CT，立即打开 “运行 RFC” 页面。请注意，您不能使用快速创建来选择较旧的 CT 版本。

     要进行排序 CTs，请使用卡片视图或表格视图中的所有更改类型区域。在任一视图中，选择一个 CT，然后单击 “创建 RFC” 打开 “运行 RFC” 页面。如果适用，“创建 RFC” 按钮旁边会出现 “使用旧版本创建” 选项。
   - 按类别选择：选择类别、子类别、项目和操作，CT 详细信息框将打开，并显示 “使用旧版本创建” 选项（如果适用）。单击 “创建 RFC” 打开 “运行 RFC” 页面。
3. 在 “运行 RFC” 页面上，打开 CT 名称区域以查看 CT 详细信息框。必须填写主题（如果您在 “浏览更改类型” 视图中选择 CT，则会为您填写此主题）。打开其他配置区域以添加有关 RFC 的信息。

   在执行配置区域中，使用可用的下拉列表或输入所需参数的值。要配置可选的执行参数，请打开其他配置区域。
4. 完成后，单击 “运行”。如果没有错误，则会显示成功创建的 RFC 页面，其中包含已提交的 RFC 详细信息和初始运行输出。
5. 打开运行参数区域以查看您提交的配置。刷新页面以更新 RFC 的执行状态。（可选）取消 RFC 或使用页面顶部的选项创建一个 RFC 的副本。

使用 CL CloudFormation I 创建采集堆栈

使用 CL CloudFormation I 创建采集堆栈

1. 使用 Inline Create（您发出包含所有 RFC 和执行参数的`create-rfc`命令）或模板创建（创建两个 JSON 文件，一个用于 RFC 参数，一个用于执行参数），然后以这两个文件作为输入发出`create-rfc`命令。这里描述了这两种方法。
2. 提交带有返回的 RFC ID 的 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 命令。

   监控 RFC: `aws amscm get-rfc --rfc-id` *`ID`* 命令。

要检查更改类型版本，请使用以下命令：

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

> **Note**
>
> 您可以将任何CreateRfc参数与任何 RFC 一起使用，无论它们是否属于变更类型的架构的一部分。例如，要在 RFC 状态更改时收到通知，请将此行添加到请求的 RFC 参数部分（不是执行参数）。--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"有关所有 CreateRfc 参数的列表，请参阅《AMS 变更管理 API 参考》。

1. 准备用于创建堆栈的 CloudFormation 模板，然后将其上传到您的 S3 存储桶。有关重要详情，请参阅 AWS CloudFormation Ingest 指南、最佳实践和限制。
2. 创建 RFC 并将其提交给 AMS：

   - 创建并保存执行参数 JSON 文件，包括所需的 CloudFormation 模板参数。以下示例将其命名为 CreateCfnParams .json。

     Web 应用程序堆栈 CreateCfnParams .json 文件示例：

```
{
  "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"
   }
  ],
}
```

SNS 主题 CreateCfnParams .json 文件示例：

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

3. 创建并保存包含以下内容的 RFC 参数 JSON 文件。以下示例将其命名为 CreateCfnRfc .json 文件：

```
{
   "ChangeTypeId": "ct-36cn2avfrrj9v",
   "ChangeTypeVersion": "2.0",
   "Title": "cfn-ingest"
}
```

4. 创建 RFC，指定 CreateCfnRfc 文件和 CreateCfnParams 文件：

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

您在响应中收到新 RFC 的 ID，并可以使用它来提交和监控 RFC。在您提交之前，RFC 仍处于编辑状态且无法启动。

提示

> **Note**
>
> 此更改类型为版本 2.0，并且是自动的（不是手动执行的）。这样可以更快地执行 CT，而且，一个新参数允许您将自定义 CloudFormation 模板粘贴到 RFC 中。CloudFormationTemplate此外，在此版本中，如果您指定了自己的安全组，我们不会附加默认 AMS 安全组。如果您未在请求中指定自己的安全组，AMS 将附加 AMS 默认安全组。在 CFN Ingest v1.0 中，无论您是否提供了自己的安全组，我们都会附加 AMS 默认安全组。
> AMS 已启用 17 项 AMS 自行配置服务以用于此变更类型。有关支持的资源的信息，请参阅 CloudFormation Ingest Stack：支持的资源。

> **Note**
>
> 版本 2.0 接受不是预签名 URL 的 S3 终端节点。
> 如果您使用此 CT 的先前版本，则 CloudFormationTemplateS3Endpoint 参数值必须是预签名 URL。
> 生成预签名 S3 存储桶 URL 的命令示例 (Mac/Linux)：
>
> ```
> export S3_PRESIGNED_URL=$(aws s3 presign DASHDASHexpires-in 86400
>  s3://BUCKET_NAME/CFN_TEMPLATE.json)
> ```
>
> 生成预签名 S3 存储桶 URL 的命令示例 (Windows)：
>
> ```
> for /f %i in ('aws s3 presign DASHDASHexpires-in 86400
>  s3://BUCKET_NAME/CFN_TEMPLATE.json') do set S3_PRESIGNED_URL=%i
> ```
>
> 另请参阅 URLs 为 Amazon S3 存储桶创建预签名存储桶。

> **Note**
>
> 如果 S3 存储桶存在于 AMS 账户中，则必须使用您的 AMS 凭据执行此命令。例如，您可能需要`--profile saml`在获取 AMS AWS Security Token Service (AWS STS) 凭证后追加。

相关变更类型：批准 CloudFormation 采集堆栈变更集，更新 CloudFormation 采集堆栈

要了解有关 AWS 的更多信息 CloudFormation，请参阅 AW S CloudFormation。要查看 CloudFormation 模板，请打开 AWS CloudFormation 模板参考。

验证收录 CloudFormation

模板经过验证以确保可以在 AMS 账户中创建。如果通过验证，则会对其进行更新，使其包含符合 AMS 要求的所有资源或配置。这包括添加诸如 Amazon CloudWatch 警报之类的资源，以允许 AMS 运营部门监控堆栈。

如果满足以下任一条件，则 RFC 将被拒绝：

- RFC JSON 语法不正确或不符合给定格式。
- 提供的 S3 存储桶预签名 URL 无效。
- 模板的 CloudFormation 语法无效。
- 该模板没有为所有参数值设置默认值。
- 模板未通过 AMS 验证。有关 AMS 验证步骤，请参阅本主题后面的信息。

如果由于资源创建问题导致 CloudFormation 堆栈创建失败，则 RFC 将失败。

要了解有关 CFN 验证和验证器的更多信息，请参阅模板验证和CloudFormation 采集堆栈：CFN 验证器示例。

更新 CloudFormation 采集堆栈

使用控制台更新 CloudFormation 采集堆栈

使用控制台更新 CloudFormation 收录堆栈

1. 导航到 “创建 RFC” 页面：在 AMS 控制台的左侧导航窗格中，单击RFCs打开 RFCs 列表页面，然后单击 “创建 R FC”。
2. 在默认的 “浏览更改类型” 视图中选择常用更改类型 (CT)，或者在 “按类别选择” 视图中选择 CT。
   - 按更改类型浏览：您可以单击 “快速创建” 区域中的常用 CT，立即打开 “运行 RFC” 页面。请注意，您不能使用快速创建来选择较旧的 CT 版本。

     要进行排序 CTs，请使用卡片视图或表格视图中的所有更改类型区域。在任一视图中，选择一个 CT，然后单击 “创建 RFC” 打开 “运行 RFC” 页面。如果适用，“创建 RFC” 按钮旁边会出现 “使用旧版本创建” 选项。

- 按类别选择：选择类别、子类别、项目和操作，CT 详细信息框将打开，并显示 “使用旧版本创建” 选项（如果适用）。单击 “创建 RFC” 打开 “运行 RFC” 页面。

3. 在 “运行 RFC” 页面上，打开 CT 名称区域以查看 CT 详细信息框。必须填写主题（如果您在 “浏览更改类型” 视图中选择 CT，则会为您填写此主题）。打开其他配置区域以添加有关 RFC 的信息。

   在执行配置区域中，使用可用的下拉列表或输入所需参数的值。要配置可选的执行参数，请打开其他配置区域。

4. 完成后，单击 “运行”。如果没有错误，则会显示成功创建的 RFC 页面，其中包含已提交的 RFC 详细信息和初始运行输出。

5. 打开运行参数区域以查看您提交的配置。刷新页面以更新 RFC 的执行状态。（可选）取消 RFC 或使用页面顶部的选项创建一个 RFC 的副本。

使用 CL CloudFormation l 更新采集堆栈

使用 CL CloudFormation l 更新采集堆栈

1. 使用 Inline Create（您发出包含所有 RFC 和执行参数的`create-rfc`命令）或模板创建（创建两个 JSON 文件，一个用于 RFC 参数，一个用于执行参数），然后以这两个文件作为输入发出`create-rfc`命令。这里描述了这两种方法。
2. 提交带有返回的 RFC ID 的 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 命令。

   监控 RFC: `aws amscm get-rfc --rfc-id` *`ID`* 命令。

要检查更改类型版本，请使用以下命令：

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

> **Note**
>
> 您可以将任何`CreateRfc`参数与任何 RFC 一起使用，无论它们是否属于变更类型的架构的一部分。例如，要在 RFC 状态更改时收到通知，请将此行添加到请求的 RFC 参数部分（不是执行参数）。`--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"`有关所有 CreateRfc 参数的列表，请参阅《AMS 变更管理 API 参考》。

1. 准备要用于更新堆栈的 CloudFormation 模板，然后将其上传到您的 S3 存储桶。有关重要详情，请参阅 AWS CloudFormation Ingest 指南、最佳实践和限制。
2. 创建 RFC 并将其提交给 AMS：

   - 创建并保存执行参数 JSON 文件，包括所需的 CloudFormation 模板参数。此示例将其命名为 UpdateCfnParams .json。

     包含内联参数更新的 UpdateCfnParams .json 文件示例：

```
{
  "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
}
```

     带有 S3 存储桶端点的 UpdateCfnParams .json 文件示例，其中包含更新后的 CloudFormation 模板：

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

```
    {
      "Key": "DisplayName",
      "Value": "DisplayNameCLI"
    }
  ],
  "TimeoutInMinutes": 1080
}
```

3. 创建并保存包含以下内容的 RFC 参数 JSON 文件。此示例将其命名为 UpdateCfnRfc .json 文件。

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

4. 创建 RFC，指定 UpdateCfnRfc 文件和 UpdateCfnParams 文件：

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

您在响应中收到新 RFC 的 ID，并可以使用它来提交和监控 RFC。在您提交之前，RFC 仍处于编辑状态且无法启动。

提示

- 此更改类型现在是 2.0 版。更改包括删除此 CT 版本 1.0 中使用的AutoApproveUpdateForResources参数，以及添加两个新参数：AutoApproveRiskyUpdates和BypassDriftCheck。
- 如果 S3 存储桶存在于 AMS 账户中，则必须使用您的 AMS 凭据执行此命令。例如，您可能需要`--profile saml`在获取 AMS AWS Security Token Service (AWS STS) 凭证后追加。
- CloudFormation 模板中资源的所有`Parameter`值都必须有一个值，可以是默认值，也可以是通过 CT 参数部分的自定义值。您可以通过构造 CloudFormation 模板资源来引用 Parameters 键来覆盖参数值。有关演示操作方法的示例，请参阅CloudFormation 采集堆栈：CFN 验证器示例。

  重要：缺少表单中未明确提供的参数，默认为现有堆栈或模板上当前设置的值。
- 有关您可以使用 CloudFormation Ingest 添加哪些自行配置服务的列表，请参阅载入堆栈：CloudFormation 支持的资源。

要了解 CloudFormation更多信息，请参阅 AWS CloudFormation。

验证收录 CloudFormation

模板经过验证以确保可以在 AMS 账户中创建。如果通过验证，则会对其进行更新，使其包含符合 AMS 要求的所有资源或配置。这包括添加诸如 Amazon CloudWatch 警报之类的资源，以允许 AMS 运营部门监控堆栈。

如果满足以下任一条件，则 RFC 将被拒绝：

- RFC JSON 语法不正确或不符合给定格式。
- 提供的 S3 存储桶预签名 URL 无效。
- 模板的 CloudFormation 语法无效。
- 该模板没有为所有参数值设置默认值。
- 模板未通过 AMS 验证。有关 AMS 验证步骤，请参阅本主题后面的信息。

如果由于资源创建问题导致 CloudFormation 堆栈创建失败，则 RFC 将失败。

要了解有关 CFN 验证和验证器的更多信息，请参阅模板验证和CloudFormation 采集堆栈：CFN 验证器示例。

批准 CloudFormation 采集堆栈变更集

使用控制台批准和更新 CloudFormation 采集堆栈

使用控制台批准和更新 CloudFormation 采集堆栈

1. 导航到 “创建 RFC” 页面：在 AMS 控制台的左侧导航窗格中，单击RFCs打开 RFCs 列表页面，然后单击 “创建 R FC”。
2. 在默认的 “浏览更改类型” 视图中选择常用更改类型 (CT)，或者在 “按类别选择” 视图中选择 CT。
   - 按更改类型浏览：您可以单击 “快速创建” 区域中的常用 CT，立即打开 “运行 RFC” 页面。请注意，您不能使用快速创建来选择较旧的 CT 版本。

     要进行排序 CTs，请使用卡片视图或表格视图中的所有更改类型区域。在任一视图中，选择一个 CT，然后单击 “创建 RFC” 打开 “运行 RFC” 页面。如果适用，“创建 RFC” 按钮旁边会出现 “使用旧版本创建” 选项。

- 按类别选择：选择类别、子类别、项目和操作，CT 详细信息框将打开，并显示 “使用旧版本创建” 选项（如果适用）。单击 “创建 RFC” 打开 “运行 RFC” 页面。

3. 在 “运行 RFC” 页面上，打开 CT 名称区域以查看 CT 详细信息框。必须填写主题（如果您在 “浏览更改类型” 视图中选择 CT，则会为您填写此主题）。打开其他配置区域以添加有关 RFC 的信息。

   在执行配置区域中，使用可用的下拉列表或输入所需参数的值。要配置可选的执行参数，请打开其他配置区域。

4. 完成后，单击 “运行”。如果没有错误，则会显示成功创建的 RFC 页面，其中包含已提交的 RFC 详细信息和初始运行输出。

5. 打开运行参数区域以查看您提交的配置。刷新页面以更新 RFC 的执行状态。（可选）取消 RFC 或使用页面顶部的选项创建一个 RFC 的副本。

使用 CLI 批准和 CloudFormation 更新采集堆栈

使用 CLI 批准和更新 CloudFormation 采集堆栈

1. 使用 Inline Create（您发出包含所有 RFC 和执行参数的`create-rfc`命令）或模板创建（创建两个 JSON 文件，一个用于 RFC 参数，一个用于执行参数），然后以这两个文件作为输入发出`create-rfc`命令。这里描述了这两种方法。
2. 提交带有返回的 RFC ID 的 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 命令。

   监控 RFC: `aws amscm get-rfc --rfc-id` *`ID`* 命令。

要检查更改类型版本，请使用以下命令：

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

> **Note**
>
> 您可以将任何`CreateRfc`参数与任何 RFC 一起使用，无论它们是否属于变更类型的架构的一部分。例如，要在 RFC 状态更改时收到通知，请将此行添加到请求的 RFC 参数部分（不是执行参数）。`--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"`有关所有 CreateRfc 参数的列表，请参阅《AMS 变更管理 API 参考》。

1. 将此更改类型的执行参数 JSON 架构输出到当前文件夹中的文件。这个例子把它命名为 CreateAsgParams .json：

```
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. 按如下方式修改并保存架构：

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

提示

> **Note**
>
> 如果堆栈中有多个资源，并且您只想删除堆栈资源的子集，请使用 Update CT；请参阅 CloudFormation Ingest Stack： CloudFormation 更新。您也可以提交服务请求案例，如果需要，AMS 工程师可以帮助您制作变更集。

要了解更多信息 AWS CloudFormation，请参阅AWS CloudFormation。

更新 CloudFormation 堆栈终止保护

使用控制台更新 CloudFormation 终止保护堆栈

下面显示了 AMS 控制台中的此更改类型。

工作原理：

1. 导航到 “创建 RFC” 页面：在 AMS 控制台的左侧导航窗格中，单击RFCs打开 RFCs 列表页面，然后单击 “创建 R FC”。
2. 在默认的 “浏览更改类型” 视图中选择常用更改类型 (CT)，或者在 “按类别选择” 视图中选择 CT。

- 按更改类型浏览：您可以单击 “快速创建” 区域中的常用 CT，立即打开 “运行 RFC” 页面。请注意，您不能使用快速创建来选择较旧的 CT 版本。

  要进行排序 CTs，请使用卡片视图或表格视图中的所有更改类型区域。在任一视图中，选择一个 CT，然后单击 “创建 RFC” 打开 “运行 RFC” 页面。如果适用，“创建 RFC” 按钮旁边会出现 “使用旧版本创建” 选项。

- 按类别选择：选择类别、子类别、项目和操作，CT 详细信息框将打开，并显示 “使用旧版本创建” 选项（如果适用）。单击 “创建 RFC” 打开 “运行 RFC” 页面。

3. 在 “运行 RFC” 页面上，打开 CT 名称区域以查看 CT 详细信息框。必须填写主题（如果您在 “浏览更改类型” 视图中选择 CT，则会为您填写此主题）。打开其他配置区域以添加有关 RFC 的信息。

   在执行配置区域中，使用可用的下拉列表或输入所需参数的值。要配置可选的执行参数，请打开其他配置区域。

4. 完成后，单击 “运行”。如果没有错误，则会显示成功创建的 RFC 页面，其中包含已提交的 RFC 详细信息和初始运行输出。

5. 打开运行参数区域以查看您提交的配置。刷新页面以更新 RFC 的执行状态。（可选）取消 RFC 或使用页面顶部的选项创建一个 RFC 的副本。

使用 CLI 更新 CloudFormation 堆栈终止保护

工作原理：

1. 使用 Inline Create（您发出包含所有 RFC 和执行参数的`create-rfc`命令）或模板创建（创建两个 JSON 文件，一个用于 RFC 参数，一个用于执行参数），然后以这两个文件作为输入发出`create-rfc`命令。这里描述了这两种方法。
2. 提交带有返回的 RFC ID 的 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 命令。

   监控 RFC: `aws amscm get-rfc --rfc-id` *`ID`* 命令。

要检查更改类型版本，请使用以下命令：

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

> **Note**
>
> 您可以将任何CreateRfc参数与任何 RFC 一起使用，无论它们是否属于变更类型的架构的一部分。例如，要在 RFC 状态更改时收到通知，请将此行添加到请求的 RFC 参数部分（不是执行参数）。--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"有关所有 CreateRfc 参数的列表，请参阅《AMS 变更管理 API 参考》。

仅指定要更改的参数。缺少的参数会保留现有值。

内联创建：

使用内联提供的执行参数发出 create RFC 命令（内联提供执行参数时请转义引号），然后提交返回的 RFC ID。例如，你可以用这样的东西替换内容：

```
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\"]}}"
```

模板创建：

1. 将此更改类型的执行参数输出到 JSON 文件；此示例将其命名为 EnableTermPro CFNParams .json：

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

2. 修改并保存 EnableTermProCFNParams 文件，仅保留要更改的参数。例如，你可以用这样的东西替换内容：

   ```
   {
     "DocumentName": "AWSManagedServices-ManageResourceTerminationProtection",
     "Region": "us-east-1",
     "Parameters": {
   ```

```
    "ResourceId": ["stack-psvnq6cupymio3enl"],
    "TerminationProtectionDesiredState": ["enabled"]
  }
}
```

3. 将 RFC 模板输出到当前文件夹中的一个文件中；此示例将其命名为 EnableTermPro CFNRfc .json：

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

4. 修改并保存 EnableTermPro CFNRfc .json 文件。例如，你可以用这样的东西替换内容：

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

5. 创建 RFC，指定 EnableTermProCFNRfc 文件和 EnableTermProCFNParams 文件：

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

   您在响应中收到新 RFC 的 ID，并可以使用它来提交和监控 RFC。在您提交之前，RFC 仍处于编辑状态且无法启动。

提示

> **Note**
>
> Amazon EC2 有一个相关的 CT，即 EC2 堆栈：更新终止保护。

要了解有关终止保护的更多信息，请参阅保护堆栈不被删除。

在 AMS 中使用 CFN 采集或堆栈更新 CTs 自动部署 IAM

您可以使用这些 AMS 变更类型在多账户着陆区 (MALZ) 和单账户着陆区 (SALZ) 中部署 IAM 角色（`AWS::IAM::Role`资源）：

- 部署 | 摄取 | 从 CloudFormation 模板堆栈 | 创建 (ct-36cn2avfrrj9v)

- 管理 | 自定义堆栈 | 来自 CloudFormation 模板的堆栈 | 更新 (ct-361tlo1k7339x)
- 管理 | 自定义堆栈 | 来自 CloudFormation 模板的堆栈 | 批准和更新 (ct-1404e21baa2ox)

对您的 CFN 模板中的 IAM 角色执行的验证：

- ManagedPolicyArns: 该属性ManagedPolicyArns不得存在于中`AWS::IAM::Role`。该验证不允许将托管策略附加到正在置备的角色。相反，可以通过属性 Policies 使用内联策略来管理角色的权限。
- PermissionsBoundary: 用于为角色设置权限边界的策略只能是 AMS 提供的托管策略:`AWSManagedServices_IAM_PermissionsBoundary`。此策略起到防护栏的作用，可保护 AMS 基础设施资源不被使用所配置的角色进行修改。使用此默认权限边界，AMS 提供的安全优势得以保留。

  `AWSManagedServices_IAM_PermissionsBoundary`（默认）为必填项，否则，请求将被拒绝。
- MaxSessionDuration：可以为 IAM 角色设置的最大会话持续时间为 1 到 4 小时。AMS 技术标准要求客户接受会话持续时间超过 4 小时的风险。
- RoleName：以下命名空间由 AMS 保留，不能用作 IAM 角色名称前缀：

```
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
```

- 策略：IAM 角色中嵌入的内联策略只能包含一组 AMS 预先批准的 IAM 操作。这是允许使用（控制策略）创建 IAM 角色的所有 IAM 操作的上限。控制策略包括：
  - AWS 托管策略中的所有操作 ReadOnlyAccess ，提供对所有资源 AWS 服务 和资源的只读访问权限
  - 以下操作仅限于跨账户 S3 操作，即允许的 S3 操作，只能对与正在创建的角色相同的账户中存在的资源执行：

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

    通过 CFN ingest 创建或更新的任何 IAM 角色都可以允许此控制策略中列出的操作，或者允许范围从控制策略中列出的操作范围缩小（不那么宽松）的操作。目前，我们允许这些可归类为只读操作的安全 IAM 操作，以及上述无法完成 CTs 且按照 AMS 技术标准预先批准的非只读操作。

- AssumeRolePolicyDocument：以下实体已获得预先批准，可以包含在信任策略中以承担正在创建的角色：
  - 同一账户中的任何 IAM 实体（角色、用户、根用户、STS 假设角色会话）都可以担任该角色。
  - 以下人员 AWS 服务 可以担任该角色：

```
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
```

- 同一账户中的 SAML 提供商可以担任该角色。目前，唯一支持的 SAML 提供商名称是`customer-saml`。

如果一项或多项验证失败，则 RFC 将被拒绝。RFC 拒绝原因示例如下：

```
{"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"}
```

如果您在 RFC 验证或执行失败时需要帮助，请使用 RFC 信函与 AMS 联系。有关说明，请参阅 RFC 通信和附件（控制台）。如有任何其他问题，请提交服务请求。有关操作方法，请参阅创建服务请求。

> **Note**
>
> 作为我们的 IAM 验证的一部分，我们目前不强制执行任何 IAM 最佳实践。有关 IAM 最佳实践，请参阅 IAM 中的安全最佳实践。

创建具有更宽松操作的 IAM 角色或执行 IAM 最佳实践

使用以下手动更改类型创建您的 IAM 实体：

- 部署 | 高级堆栈组件 | 身份和访问管理 (IAM) Management | 创建实体或策略 (ct-3dpd8mdd9jn1r)
- 管理 | 高级堆栈组件 | 身份和访问管理 (IAM) | 更新实体或策略 (ct-27tuth19k52b4)

我们建议您在提交这些手册之前阅读并理解我们的技术标准 RFCs。有关访问权限，请参阅如何访问技术标准。

> **Note**
>
> 使用这些手动更改类型直接创建的每个 IAM 角色都属于自己的单独堆栈，并且不位于通过 CFN Ingest CT 创建其他基础设施资源的同一个堆栈中。

当无法通过自动更改类型进行更新时，通过手动更改类型更新使用 CFN 提取创建的 IAM 角色

使用管理 | 高级堆栈组件 | 身份和访问管理 (IAM) | 更新实体或策略 (ct-27tuth19k52b4) 更改类型。

> **Important**
>
> 通过手动 CT 对 IAM 角色的更新不会反映在 CFN 堆栈模板中，这会导致堆栈偏移。通过手动请求将角色更新到未通过我们验证的状态后，只要角色仍然不符合我们的验证，就无法再次使用堆栈更新 CT (ct-361tlo1k7339x) 对其进行进一步更新。只有当 CFN 堆栈模板符合我们的验证标准时，才能使用更新 CT。但是，只要不更新不符合我们验证的 IAM 资源并且 CFN 模板通过我们的验证，仍然可以通过堆栈更新 CT (ct-361tlo1k7339x) 更新堆栈。

删除通过 AWS CloudFormation 采集创建的 IAM 角色

如果要删除整个堆栈，请使用以下自动删除堆栈更改类型。有关说明，请参阅删除堆栈：

- 更改类型 ID：ct-0q0bic0ywqk6c
- 分类：管理 | 标准堆栈 | 堆栈 | 删除和管理 | 高级堆栈组件 | 堆栈 | 删除

如果您想在不删除整个堆栈的情况下删除 IAM 角色，则可以从 CloudFormation 模板中移除 IAM 角色，然后使用更新的模板作为自动堆栈更新更改类型的输入：

- 更改类型 ID：ct-361tlo1k7339x
- 分类：管理 | 自定义堆栈 | 来自 CloudFormation 模板的堆栈 | 更新

有关说明，请参阅更新 AWS CloudFormation 采集堆栈。

CodeDeploy 请求

您可以使用 AWS CodeDeploy 创建应用程序容器，然后通过 CodeDeploy 应用程序组部署这些容器。有关的更多信息 CodeDeploy，请参阅 AWS CodeDeploy 文档。

使用 AWS CodeDeploy 涉及以下过程：

1. 创建 CodeDeploy 应用程序。 CodeDeploy 应用程序是用于确保在部署期间引 CodeDeploy 用正确的修订版、部署配置和部署组的名称或容器。
2. 创建 CodeDeploy 部署组。 CodeDeploy 部署组定义了一组针对部署的单个实例。AMS 为 CodeDeploy 部署组提供了单独的更改类型 EC2。
3. 通过部署组 CodeDeploy 部署 CodeDeploy 应用程序。

CodeDeploy 应用程序

创建或部署 CodeDeploy 应用程序。

创建 CodeDeploy 应用程序

使用控制台创建 CodeDeploy 应用程序

工作原理：

1. 导航到 “创建 RFC” 页面：在 AMS 控制台的左侧导航窗格中，单击RFCs打开 RFCs 列表页面，然后单击 “创建 R FC”。
2. 在默认的 “浏览更改类型” 视图中选择常用更改类型 (CT)，或者在 “按类别选择” 视图中选择 CT。
   - 按更改类型浏览：您可以单击 “快速创建” 区域中的常用 CT，立即打开 “运行 RFC” 页面。请注意，您不能使用快速创建来选择较旧的 CT 版本。

     要进行排序 CTs，请使用卡片视图或表格视图中的所有更改类型区域。在任一视图中，选择一个 CT，然后单击 “创建 RFC” 打开 “运行 RFC” 页面。如果适用，“创建 RFC” 按钮旁边会出现 “使用旧版本创建” 选项。

- 按类别选择：选择类别、子类别、项目和操作，CT 详细信息框将打开，并显示 “使用旧版本创建” 选项（如果适用）。单击 “创建 RFC” 打开 “运行 RFC” 页面。

3. 在 “运行 RFC” 页面上，打开 CT 名称区域以查看 CT 详细信息框。必须填写主题（如果您在 “浏览更改类型” 视图中选择 CT，则会为您填写此主题）。打开其他配置区域以添加有关 RFC 的信息。

   在执行配置区域中，使用可用的下拉列表或输入所需参数的值。要配置可选的执行参数，请打开其他配置区域。

4. 完成后，单击 “运行”。如果没有错误，则会显示成功创建的 RFC 页面，其中包含已提交的 RFC 详细信息和初始运行输出。

5. 打开运行参数区域以查看您提交的配置。刷新页面以更新 RFC 的执行状态。（可选）取消 RFC 或使用页面顶部的选项创建一个 RFC 的副本。

使用 CLI 创建 CodeDeploy 应用程序

工作原理：

1. 使用 Inline Create（您发出包含所有 RFC 和执行参数的`create-rfc`命令）或模板创建（创建两个 JSON 文件，一个用于 RFC 参数，一个用于执行参数），然后以这两个文件作为输入发出`create-rfc`命令。这里描述了这两种方法。
2. 提交带有返回的 RFC ID 的 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 命令。

   监控 RFC: `aws amscm get-rfc --rfc-id` *`ID`* 命令。

要检查更改类型版本，请使用以下命令：

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

> **Note**
>
> 您可以将任何`CreateRfc`参数与任何 RFC 一起使用，无论它们是否属于变更类型的架构的一部分。例如，要在 RFC 状态更改时收到通知，请将此行添加到请求的 RFC 参数部分（不是执行参数）。`--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"`有关所有 CreateRfc 参数的列表，请参阅《AMS 变更管理 API 参考》。

内联创建：

使用内联提供的执行参数发出 create RFC 命令（内联提供执行参数时请转义引号），然后提交返回的 RFC ID。例如，你可以用这样的东西替换内容：

```
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\"}}"
```

模板创建：

1. 将 CodeDeploy 应用程序 CT 的执行参数 JSON 架构输出到当前文件夹中的一个文件中；此示例将其命名为 Create CDApp params.json：

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

2. 按如下方式修改并保存 JSON 文件。例如，你可以用这样的东西替换内容：

   ```
   {
   "Description":                      "Create WP CodeDeploy App",
   "VpcId":                            "VPC_ID",
   "StackTemplateId":                  "stm-sft6rv00000000000",
   "Name":                             "WpCDApp",
   "TimeoutInMinutes":                 60,
   "Parameters":   {
       "CodeDeployApplicationName":    "WordPressCDApp"
       }
   }
   ```

3. 将的 JSON 模板输出 CreateRfc 到当前文件夹中的一个文件中；此示例将其命名为 Create CDApp rfc.json：

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

4. 按如下方式修改并保存 JSON 文件。例如，你可以用这样的东西替换内容：

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

```
}
```

5. 创建 RFC，指定创建 CDApp Rfc 文件和执行参数文件：

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

您在响应中收到新 RFC 的 ID，并可以使用它来提交和监控 RFC。在您提交之前，RFC 仍处于编辑状态且无法启动。

提示

有关 AWS 的更多信息 CodeDeploy，请参阅使用 AWS 创建应用程序 CodeDeploy。

部署 CodeDeploy 应用程序

使用控制台部署 CodeDeploy 应用程序

工作原理：

1. 导航到 “创建 RFC” 页面：在 AMS 控制台的左侧导航窗格中，单击RFCs打开 RFCs 列表页面，然后单击 “创建 R FC”。
2. 在默认的 “浏览更改类型” 视图中选择常用更改类型 (CT)，或者在 “按类别选择” 视图中选择 CT。
   - 按更改类型浏览：您可以单击 “快速创建” 区域中的常用 CT，立即打开 “运行 RFC” 页面。请注意，您不能使用快速创建来选择较旧的 CT 版本。

     要进行排序 CTs，请使用卡片视图或表格视图中的所有更改类型区域。在任一视图中，选择一个 CT，然后单击 “创建 RFC” 打开 “运行 RFC” 页面。如果适用，“创建 RFC” 按钮旁边会出现 “使用旧版本创建” 选项。
   - 按类别选择：选择类别、子类别、项目和操作，CT 详细信息框将打开，并显示 “使用旧版本创建” 选项（如果适用）。单击 “创建 RFC” 打开 “运行 RFC” 页面。
3. 在 “运行 RFC” 页面上，打开 CT 名称区域以查看 CT 详细信息框。必须填写主题（如果您在 “浏览更改类型” 视图中选择 CT，则会为您填写此主题）。打开其他配置区域以添加有关 RFC 的信息。

   在执行配置区域中，使用可用的下拉列表或输入所需参数的值。要配置可选的执行参数，请打开其他配置区域。
4. 完成后，单击 “运行”。如果没有错误，则会显示成功创建的 RFC 页面，其中包含已提交的 RFC 详细信息和初始运行输出。

5. 打开运行参数区域以查看您提交的配置。刷新页面以更新 RFC 的执行状态。（可选）取消 RFC 或使用页面顶部的选项创建一个 RFC 的副本。

使用 CLI 部署 CodeDeploy 应用程序

工作原理：

1. 使用 Inline Create（您发出包含所有 RFC 和执行参数的`create-rfc`命令）或模板创建（创建两个 JSON 文件，一个用于 RFC 参数，一个用于执行参数），然后以这两个文件作为输入发出`create-rfc`命令。这里描述了这两种方法。
2. 提交带有返回的 RFC ID 的 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 命令。

   监控 RFC: `aws amscm get-rfc --rfc-id` *`ID`* 命令。

要检查更改类型版本，请使用以下命令：

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

> **Note**
>
> 您可以将任何`CreateRfc`参数与任何 RFC 一起使用，无论它们是否属于变更类型的架构的一部分。例如，要在 RFC 状态更改时收到通知，请将此行添加到请求的 RFC 参数部分（不是执行参数）。`--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"`有关所有 CreateRfc 参数的列表，请参阅《AMS 变更管理 API 参考》。

内联创建：

使用内联提供的执行参数发出 create RFC 命令（内联提供执行参数时请转义引号），然后提交返回的 RFC ID。例如，你可以用这样的东西替换内容：

```
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\"}}"
```

模板创建：

1. 输出 CodeDeploy 应用程序部署 CT 的执行参数 JSON 架构；此示例将其命名为 Deploy CDApp params.json：

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

2. 按如下方式修改 JSON 文件。例如，你可以用这样的东西替换内容：

```
{
"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. 将的 JSON 模板输出 CreateRfc 到当前文件夹中的一个文件中；此示例将其命名为 Deploy CDApp rfc.json：

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

4. 修改并保存 Deploy CDApp rfc.json 文件。例如，你可以用这样的东西替换内容：

```
{
```

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

5. 创建 RFC，指定执行参数文件和 Deploy CDApp Rfc 文件：

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

您在响应中收到新 RFC 的 ID，并可以使用它来提交和监控 RFC。在您提交之前，RFC 仍处于编辑状态且无法启动。

提示

有关更多信息，请参阅使用创建部署 CodeDeploy。

CodeDeploy 部署组

创建 CodeDeploy 应用程序组。

创建 CodeDeploy 部署组

使用控制台创建 CodeDeploy 部署组

工作原理：

1. 导航到 “创建 RFC” 页面：在 AMS 控制台的左侧导航窗格中，单击RFCs打开 RFCs 列表页面，然后单击 “创建 R FC”。
2. 在默认的 “浏览更改类型” 视图中选择常用更改类型 (CT)，或者在 “按类别选择” 视图中选择 CT。
   - 按更改类型浏览：您可以单击 “快速创建” 区域中的常用 CT，立即打开 “运行 RFC” 页面。请注意，您不能使用快速创建来选择较旧的 CT 版本。

     要进行排序 CTs，请使用卡片视图或表格视图中的所有更改类型区域。在任一视图中，选择一个 CT，然后单击 “创建 RFC” 打开 “运行 RFC” 页面。如果适用，“创建 RFC” 按钮旁边会出现 “使用旧版本创建” 选项。
   - 按类别选择：选择类别、子类别、项目和操作，CT 详细信息框将打开，并显示 “使用旧版本创建” 选项（如果适用）。单击 “创建 RFC” 打开 “运行 RFC” 页面。

3. 在 “运行 RFC” 页面上，打开 CT 名称区域以查看 CT 详细信息框。必须填写主题（如果您在 “浏览更改类型” 视图中选择 CT，则会为您填写此主题）。打开其他配置区域以添加有关 RFC 的信息。

   在执行配置区域中，使用可用的下拉列表或输入所需参数的值。要配置可选的执行参数，请打开其他配置区域。

4. 完成后，单击 “运行”。如果没有错误，则会显示成功创建的 RFC 页面，其中包含已提交的 RFC 详细信息和初始运行输出。

5. 打开运行参数区域以查看您提交的配置。刷新页面以更新 RFC 的执行状态。（可选）取消 RFC 或使用页面顶部的选项创建一个 RFC 的副本。

使用 CLI 创建 CodeDeploy 部署组

工作原理：

1. 使用 Inline Create（您发出包含所有 RFC 和执行参数的`create-rfc`命令）或模板创建（创建两个 JSON 文件，一个用于 RFC 参数，一个用于执行参数），然后以这两个文件作为输入发出`create-rfc`命令。这里描述了这两种方法。
2. 提交带有返回的 RFC ID 的 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 命令。

   监控 RFC: `aws amscm get-rfc --rfc-id` *`ID`* 命令。

要检查更改类型版本，请使用以下命令：

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

> **Note**
>
> 您可以将任何`CreateRfc`参数与任何 RFC 一起使用，无论它们是否属于变更类型的架构的一部分。例如，要在 RFC 状态更改时收到通知，请将此行添加到请求的 RFC 参数部分（不是执行参数）。`--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"`有关所有 CreateRfc 参数的列表，请参阅《AMS 变更管理 API 参考》。

内联创建：

使用内联提供的执行参数发出 create RFC 命令（内联提供执行参数时请转义引号），然后提交返回的 RFC ID。例如，你可以用这样的东西替换内容：

```
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\"}}"
```

模板创建：

1. 将执行参数 JSON 架构输出到当前文件夹中的一个文件中；此示例将其命名为 Cre CDDep GroupParams ate.json：

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

2. 修改并保存 JSON 文件。例如，你可以用这样的东西替换内容：

   ```
   {
   "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. 将的 JSON 模板输出 CreateRfc 到当前文件夹中的一个文件中；此示例将其命名为 Cre CDDep GroupRfc ate.json：

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

4. 修改并保存 JSON 文件。例如，你可以用这样的东西替换内容：

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

5. 创建 RFC，指定创建CDDepGroupRfc 文件和执行参数文件：

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

   您在响应中收到新 RFC 的 ID，并可以使用它来提交和监控 RFC。在您提交之前，RFC 仍处于编辑状态且无法启动。

提示

有关 AWS CodeDeploy 部署组的更多信息，请参阅使用 AWS 创建部署组 CodeDeploy。

为 EC2 创建 CodeDeploy 部署组

使用控制台为 EC2 创建 CodeDeploy 部署组

工作原理：

1. 导航到 “创建 RFC” 页面：在 AMS 控制台的左侧导航窗格中，单击RFCs打开 RFCs 列表页面，然后单击 “创建 R FC”。
2. 在默认的 “浏览更改类型” 视图中选择常用更改类型 (CT)，或者在 “按类别选择” 视图中选择 CT。
   - 按更改类型浏览：您可以单击 “快速创建” 区域中的常用 CT，立即打开 “运行 RFC” 页面。请注意，您不能使用快速创建来选择较旧的 CT 版本。

     要进行排序 CTs，请使用卡片视图或表格视图中的所有更改类型区域。在任一视图中，选择一个 CT，然后单击 “创建 RFC” 打开 “运行 RFC” 页面。如果适用，“创建 RFC” 按钮旁边会出现 “使用旧版本创建” 选项。

- 按类别选择：选择类别、子类别、项目和操作，CT 详细信息框将打开，并显示 “使用旧版本创建” 选项（如果适用）。单击 “创建 RFC” 打开 “运行 RFC” 页面。

3. 在 “运行 RFC” 页面上，打开 CT 名称区域以查看 CT 详细信息框。必须填写主题（如果您在 “浏览更改类型” 视图中选择 CT，则会为您填写此主题）。打开其他配置区域以添加有关 RFC 的信息。

   在执行配置区域中，使用可用的下拉列表或输入所需参数的值。要配置可选的执行参数，请打开其他配置区域。

4. 完成后，单击 “运行”。如果没有错误，则会显示成功创建的 RFC 页面，其中包含已提交的 RFC 详细信息和初始运行输出。

5. 打开运行参数区域以查看您提交的配置。刷新页面以更新 RFC 的执行状态。（可选）取消 RFC 或使用页面顶部的选项创建一个 RFC 的副本。

使用 CLI 为 EC2 创建 CodeDeploy 部署组

工作原理：

1. 使用 Inline Create（您发出包含所有 RFC 和执行参数的`create-rfc`命令）或模板创建（创建两个 JSON 文件，一个用于 RFC 参数，一个用于执行参数），然后以这两个文件作为输入发出`create-rfc`命令。这里描述了这两种方法。
2. 提交带有返回的 RFC ID 的 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 命令。

   监控 RFC: `aws amscm get-rfc --rfc-id` *`ID`* 命令。

要检查更改类型版本，请使用以下命令：

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

> **Note**
>
> 您可以将任何`CreateRfc`参数与任何 RFC 一起使用，无论它们是否属于变更类型的架构的一部分。例如，要在 RFC 状态更改时收到通知，请将此行添加到请求的 RFC 参数部分（不是执行参数）。`--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"`有关所有 CreateRfc 参数的列表，请参阅《AMS 变更管理 API 参考》。

内联创建：

使用内联提供的执行参数发出 create RFC 命令（内联提供执行参数时请转义引号），然后提交返回的 RFC ID。例如，你可以用这样的东西替换内容：

```
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\":\"\"}}"
```

模板创建：

1. 将执行参数 JSON 架构输出到文件中；此示例将其命名为 Create CDDep GroupEc 2params.json：

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

2. 修改并保存 JSON 文件。例如，你可以用这样的东西替换内容：

   ```
   {
   "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. 将的 JSON 模板输出 CreateRfc 到当前文件夹中的一个文件中；此示例将其命名为 Create CDDep GroupEc 2rfc.json：

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

4. 修改并保存 JSON 文件。例如，你可以用这样的东西替换内容：

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

5. 创建 RFC，指定创建 CDDep GroupEc 2Rfc 文件和执行参数文件：

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

您在响应中收到新 RFC 的 ID，并可以使用它来提交和监控 RFC。在您提交之前，RFC 仍处于编辑状态且无法启动。

提示

有关 AWS CodeDeploy 部署组的更多信息，请参阅使用 AWS 创建部署组 CodeDeploy。

AWS Database Migration Service (AWS DMS)

AWS Database Migration Service (AWS DMS) 可帮助您轻松安全地将数据库迁移到 AMS。您可以在与最广泛使用的商用和开源数据库 (例如 Oracle、MySQL 和 PostgreSQL) 之间迁移数据。该服务支持同构迁移，例如 Oracle 到 Oracle，也支持不同数据库平台之间的异构迁移，例如 Oracle 到 PostgreSQL 或 MySQL 到 Oracle。 AWS DMS 是一项 AWS 服务；AMS CTs 可帮助您在 AMS AWS DMS 管理的账户中创建资源

下图描绘了数据库迁移的工作流程。

主题

- AWS Database Migration Service (AWS DMS)，在你开始之前
- AWS DMS，设置所需的数据
- AWS DMS 设置任务
- AWS DMS 管理

AWS Database Migration Service (AWS DMS)，在你开始之前

使用 AMS 计划数据库迁移时 AWS DMS，请考虑以下几点：

- 源端点和目标端点：您需要知道源数据库中的哪些信息和表需要迁移到目标数据库。AMS AWS DMS 支持基本架构迁移，包括创建表和主键。但是，AMS AWS DMS 不会自动在目标数据库中创建二级索引、外键、帐户等。有关更多信息，请参阅数据迁移源和数据迁移目标。
- 架构/代码迁移：AMS AWS DMS 不执行架构或代码转换。您可以使用 Oracle SQL Developer、MySQL Workbench 或 pgAdmin III 等工具来转换架构。如果您想将现有架构转换为其他数据库引擎，可以使用 AWS Schema Conversion Tool。它可以创建目标架构，也可以生成和创建整个架构：表、索引、视图等。您还可以使用该工具将 TSQL 转换为 PL/SQL pgSQL 和其他格式。
- 不支持的数据类型：某些源数据类型需要转换为目标数据库的等效数据类型。

AWS DMS 需要考虑的场景

以下记录在案的场景可能会帮助您制定自己的数据库迁移路径。

- 将数据从本地 MySQL 服务器迁移到 Amazon RDS MySQL：请参阅 AWS 博客文章将本地 MySQL 数据迁移到 Amazon RDS（然后返回）
- 将数据从 Oracle 数据库迁移到 Amazon RDS Aurora PostgreSQL 数据库：参见 AWS 博客文章从 Oracle 数据库迁移到 Amazon Aurora PostgreSQL 数据库的简要介绍
- 将数据从 RDS MySQL 迁移到 S3：参见 AWS 博客文章如何使用 AWS DMS 将关系数据库中的数据存档到 Amazon Glacier

对于数据库迁移，您必须执行以下操作：

- 规划数据库迁移，包括设置复制子网组。
- 分配一个执行所有迁移过程的复制实例。
- 指定源和目标数据库端点。
- 创建一个任务或一组任务来定义要使用的表和复制过程。
- 创建 l AWS DMS AM `dms-cloudwatch-logs-role` 和`dms-vpc-role`角色。如果您使用 Amazon Redshift 作为目标数据库，则还必须创建 IAM 角色`dms-access-for-endpoint`并将其添加到您的 AWS 账户。有关更多信息，请参阅创建要与 AWS CLI 和 AWS DMS API 配合使用的 IAM 角色。

这些演练提供了使用 AMS 控制台或 AMS CLI 创建 AWS Database Migration Service (AWS DMS) 的示例。提供了用于创建 AWS DMS 复制实例、子网组和任务以及 AWS DMS 源终端节点和目标终端节点的 CLI 命令。

要了解有关 AMS 的更多信息 AWS DMS，AWS Database Migration Service请参阅，了解一般信息和AWS Database Migration Service FAQs常见问题的答案。

AWS DMS，设置所需的数据

以下每项 AWS DMS 演练都需要一些共同的数据。

- `Description`：有关资源的有意义的信息，与其他参数`Description`选项是分开的。
- `VpcId`：要使用的 VPC。您可以通过运行 SKMS API（`list-vpc-summaries`在 CLI 中）的 ListVpcSummaries 操作或查看 AMS 控制台中的VPCs页面来找出答案。有关 AMS SKMS API 参考，请参阅 AWS Artifact 控制台中的 “报告” 选项卡。
- `Name`: 堆栈或堆栈组件的名称；这将成为堆栈名称。
- `TimeoutInMinutes`：在 RFC 失败之前，允许多少分钟来创建堆栈。此设置不会延迟 RFC 的执行，但必须留出足够的时间（例如，不要指定`"5"`）。
- `ChangeTypeId`、`ChangeTypeVersion`、和`StackTemplateId`：这些是必填项，但因 CT 而异，其值将在以下每个相关部分中提供。

AWS DMS 设置任务

使用以下演练 AWS DMS 进行设置。

1： AWS DMS 复制子网组：创建

您可以使用 AMS 控制台或 API/CLI 创建 AMS AWS DMS 复制子网组。

创建 AWS DMS 复制子网组

使用控制台创建 AWS DMS 复制子网组

> **Note**
>
> 如果账户中不存在 `dms-vpc-role` IAM 角色，则此 CT 将失败。

工作原理：

1. 导航到 “创建 RFC” 页面：在 AMS 控制台的左侧导航窗格中，单击RFCs打开 RFCs 列表页面，然后单击 “创建 R FC”。
2. 在默认的 “浏览更改类型” 视图中选择常用更改类型 (CT)，或者在 “按类别选择” 视图中选择 CT。
   - 按更改类型浏览：您可以单击 “快速创建” 区域中的常用 CT，立即打开 “运行 RFC” 页面。请注意，您不能使用快速创建来选择较旧的 CT 版本。

     要进行排序 CTs，请使用卡片视图或表格视图中的所有更改类型区域。在任一视图中，选择一个 CT，然后单击 “创建 RFC” 打开 “运行 RFC” 页面。如果适用，“创建 RFC” 按钮旁边会显示 “使用旧版本创建” 选项。
   - 按类别选择：选择类别、子类别、项目和操作，CT 详细信息框将打开，并显示 “使用旧版本创建” 选项（如果适用）。单击 “创建 RFC” 打开 “运行 RFC” 页面。
3. 在 “运行 RFC” 页面上，打开 CT 名称区域以查看 CT 详细信息框。必须填写主题（如果您在 “浏览更改类型” 视图中选择 CT，则会为您填写此主题）。打开其他配置区域以添加有关 RFC 的信息。

   在执行配置区域中，使用可用的下拉列表或输入所需参数的值。要配置可选的执行参数，请打开其他配置区域。
4. 完成后，单击 “运行”。如果没有错误，则会显示成功创建的 RFC 页面，其中包含已提交的 RFC 详细信息和初始运行输出。
5. 打开运行参数区域以查看您提交的配置。刷新页面以更新 RFC 的执行状态。（可选）取消 RFC 或使用页面顶部的选项创建一个 RFC 的副本。

使用 CLI 创建 AWS DMS 复制子网组

> **Note**
>
> 如果账户中不存在 `dms-vpc-role` IAM 角色，则此 CT 将失败。

工作原理：

1. 使用 Inline Create（您发出包含所有 RFC 和执行参数的`create-rfc`命令）或模板创建（创建两个 JSON 文件，一个用于 RFC 参数，一个用于执行参数），然后以这两个文件作为输入发出`create-rfc`命令。这里描述了这两种方法。
2. 提交带有返回的 RFC ID 的 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 命令。

监控 RFC: `aws amscm get-rfc --rfc-id ID` 命令。

要检查更改类型版本，请使用以下命令：

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

> **Note**
>
> 您可以将任何CreateRfc参数与任何 RFC 一起使用，无论它们是否属于变更类型的架构的一部分。例如，要在 RFC 状态更改时收到通知，请将此行添加到请求的 RFC 参数部分（不是执行参数）。--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"有关所有 CreateRfc 参数的列表，请参阅《AMS 变更管理 API 参考》。

内联创建：

使用内联提供的执行参数（内联提供执行参数时使用转义引号）发出 create RFC 命令，然后提交返回的 RFC ID。例如，你可以用这样的东西替换内容：

```
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
\"}"
```

模板创建：

1. 将此更改类型的执行参数输出到 JSON 文件；此示例将其命名为 CreateDmsRsgParams .json：

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

2. 修改并保存执行参数 CreateDmsRsgParams .json 文件。例如，你可以用这样的东西替换内容：

   ```
   {
   "Description":          "DMSTestRepSG",
   ```

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

3. 将 JSON 模板输出到当前文件夹中的一个文件中；此示例将其命名为 CreateDmsRsgRfc .json：

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

4. 修改并保存 CreateDmsRsgRfc .json 文件。例如，你可以用这样的东西替换内容：

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

5. 创建 RFC，指定执行参数文件和 CreateDmsRsgRfc 文件：

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

   您在响应中收到新 RFC 的 ID，并可以使用它来提交和监控 RFC。在您提交之前，RFC 仍处于编辑状态且无法启动。

提示

- 如果账户中不存在 `dms-vpc-role` IAM 角色，则此 CT 将失败。
- 您最多可以添加 50 个标签，但要这样做，您必须启用其他配置视图。

  有关 DMS 复制实例和子网组的更多信息，请参阅为复制实例设置网络。

2： AWS DMS 复制实例：创建

您可以使用 AMS 控制台或 API/CLI 创建 AMS AWS DMS 复制实例。

创建 AWS DMS 复制实例

使用控制台创建 AWS DMS 复制实例

AMS 控制台中此更改类型的屏幕截图：

工作原理：

1. 导航到创建 RFC 页面：在 AMS 控制台的左侧导航窗格中，单击RFCs打开 RFCs 列表页面，然后单击创建 R FC。
2. 在默认的 “浏览更改类型” 视图中选择常用更改类型 (CT)，或者在 “按类别选择” 视图中选择 CT。
   - 按更改类型浏览：您可以单击 “快速创建” 区域中的常用 CT，立即打开 “运行 RFC” 页面。请注意，您不能使用快速创建来选择较旧的 CT 版本。

     要进行排序 CTs，请使用卡片视图或表格视图中的所有更改类型区域。在任一视图中，选择一个 CT，然后单击 “创建 RFC” 打开 “运行 RFC” 页面。如果适用，“创建 RFC” 按钮旁边会显示 “使用旧版本创建” 选项。
   - 按类别选择：选择类别、子类别、项目和操作，CT 详细信息框将打开，并显示 “使用旧版本创建” 选项（如果适用）。单击 “创建 RFC” 打开 “运行 RFC” 页面。
3. 在 “运行 RFC” 页面上，打开 CT 名称区域以查看 CT 详细信息框。必须填写主题（如果您在 “浏览更改类型” 视图中选择 CT，则会为您填写此主题）。打开其他配置区域以添加有关 RFC 的信息。

   在执行配置区域中，使用可用的下拉列表或输入所需参数的值。要配置可选的执行参数，请打开其他配置区域。
4. 完成后，单击 “运行”。如果没有错误，则会显示成功创建的 RFC 页面，其中包含已提交的 RFC 详细信息和初始运行输出。
5. 打开运行参数区域以查看您提交的配置。刷新页面以更新 RFC 的执行状态。（可选）取消 RFC 或使用页面顶部的选项创建一个 RFC 的副本。

使用 CLI 创建 AWS DMS 复制实例

工作原理：

1. 使用 Inline Create（您发出包含所有 RFC 和执行参数的`create-rfc`命令）或模板创建（创建两个 JSON 文件，一个用于 RFC 参数，一个用于执行参数），然后以这两个文件作为输入发出`create-rfc`命令。这里描述了这两种方法。
2. 提交带有返回的 RFC ID 的 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 命令。

监控 RFC: `aws amscm get-rfc --rfc-id ID` 命令。

要检查更改类型版本，请使用以下命令：

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

> **Note**
>
> 您可以将任何CreateRfc参数与任何 RFC 一起使用，无论它们是否属于变更类型的架构的一部分。例如，要在 RFC 状态更改时收到通知，请将此行添加到请求的 RFC 参数部分（不是执行参数）。--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"有关所有 CreateRfc 参数的列表，请参阅《AMS 变更管理 API 参考》。

内联创建：

使用内联提供的执行参数（内联提供执行参数时使用转义引号）发出 create RFC 命令，然后提交返回的 RFC ID。例如，你可以用这样的东西替换内容：

```
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\"}"
```

在创建复制实例时，您可以指定源和目标数据存储。源和目标数据存储可以存储在亚马逊弹性计算云 (Amazon EC2) 实例、AWS S3 存储桶、亚马逊关系数据库服务 (Amazon RDS) 数据库实例或本地数据库上。

模板创建：

1. 将此更改类型的执行参数输出到 JSON 文件；此示例将其命名为 CreateDmsRiParams .json：

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

2. 修改并保存执行参数 CreateDmsRiParams .json 文件。例如，你可以用这样的东西替换内容：

```
{
"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. 将 JSON 模板输出到当前文件夹中的一个文件中；此示例将其命名为 CreateDmsRiRfc .json：

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

4. 修改并保存 CreateDmsRiRfc .json 文件。例如，你可以用这样的东西替换内容：

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

5. 创建 RFC，指定执行参数文件和 CreateDmsRiRfc 文件：

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

   您在响应中收到新 RFC 的 ID，并可以使用它来提交和监控 RFC。在您提交之前，RFC 仍处于编辑状态且无法启动。

提示

- 您最多可以添加 50 个标签，但要这样做，您必须启用其他配置视图。
- 您必须在 AMS VPC 中的 EC2 实例上创建具有足够存储空间和处理能力的复制实例，以执行您分配的任务，并将数据从源数据库迁移到目标数据库。此实例的所需大小是变化的，具体取决于需迁移的

数据量和需要实例执行的任务数。当您选择该MultiAZ选项时，复制实例使用多可用区部署提供高可用性和故障转移支持。有关复制实例的更多信息，请参阅使用 AWS DMS 复制实例。

3： AWS DMS 源端点：创建、为 Mongo 数据库创建、为 S3 创建

您可以使用 AMS 控制台或 API/CLI 为各种数据库创建 AMS DMS 源端点，我们提供了三个示例。

DMS 源端点：创建

使用控制台创建 DMS 源端点

AMS 控制台中此更改类型的屏幕截图：

工作原理：

1. 导航到 “创建 RFC” 页面：在 AMS 控制台的左侧导航窗格中，单击RFCs打开 RFCs 列表页面，然后单击 “创建 R FC”。
2. 在默认的 “浏览更改类型” 视图中选择常用更改类型 (CT)，或者在 “按类别选择” 视图中选择 CT。
   - 按更改类型浏览：您可以单击 “快速创建” 区域中的常用 CT，立即打开 “运行 RFC” 页面。请注意，您不能使用快速创建来选择较旧的 CT 版本。

     要进行排序 CTs，请使用卡片视图或表格视图中的所有更改类型区域。在任一视图中，选择一个 CT，然后单击 “创建 RFC” 打开 “运行 RFC” 页面。如果适用，“创建 RFC” 按钮旁边会出现 “使用旧版本创建” 选项。
   - 按类别选择：选择类别、子类别、项目和操作，CT 详细信息框将打开，并显示 “使用旧版本创建” 选项（如果适用）。单击 “创建 RFC” 打开 “运行 RFC” 页面。
3. 在 “运行 RFC” 页面上，打开 CT 名称区域以查看 CT 详细信息框。必须填写主题（如果您在 “浏览更改类型” 视图中选择 CT，则会为您填写此主题）。打开其他配置区域以添加有关 RFC 的信息。

   在执行配置区域中，使用可用的下拉列表或输入所需参数的值。要配置可选的执行参数，请打开其他配置区域。
4. 完成后，单击 “运行”。如果没有错误，则会显示成功创建的 RFC 页面，其中包含已提交的 RFC 详细信息和初始运行输出。
5. 打开运行参数区域以查看您提交的配置。刷新页面以更新 RFC 的执行状态。（可选）取消 RFC 或使用页面顶部的选项创建一个 RFC 的副本。

使用 CLI 创建 DMS 源端点

工作原理：

1. 使用 Inline Create（您发出包含所有 RFC 和执行参数的`create-rfc`命令）或模板创建（创建两个 JSON 文件，一个用于 RFC 参数，一个用于执行参数），然后以这两个文件作为输入发出`create-rfc`命令。这里描述了这两种方法。
2. 提交带有返回的 RFC ID 的 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 命令。

   监控 RFC: `aws amscm get-rfc --rfc-id` *`ID`* 命令。

要检查更改类型版本，请使用以下命令：

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

> **Note**
>
> 您可以将任何`CreateRfc`参数与任何 RFC 一起使用，无论它们是否属于变更类型的架构的一部分。例如，要在 RFC 状态更改时收到通知，请将此行添加到请求的 RFC 参数部分（不是执行参数）。`--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"`有关所有 CreateRfc 参数的列表，请参阅《AMS 变更管理 API 参考》。

内联创建：

使用内联提供的执行参数（内联提供执行参数时使用转义引号）发出 create RFC 命令，然后提交返回的 RFC ID。例如，你可以用这样的东西替换内容：

```
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\"}"
```

模板创建：

1. 将此更改类型的执行参数输出到名为 CreateDmsSeParams .json 的 JSON 文件中。

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

2. 修改并保存执行参数 JSON 文件。例如，你可以用这样的东西替换内容：

```
{
"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. 将 JSON 模板输出到当前文件夹中的一个文件中；此示例将其命名为 CreateDmsSeRfc .json：

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

4. 修改并保存 CreateDmsSeRfc .json 文件。例如，你可以用这样的东西替换内容：

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

5. 创建 RFC，指定执行参数文件和 CreateDmsSeRfc 文件：

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

您在响应中收到新 RFC 的 ID，并可以使用它来提交和监控 RFC。在您提交之前，RFC 仍处于编辑状态且无法启动。

提示

在创建 DMS 端点之前，请确保您的密码不包含不支持的字符。有关更多信息，请参阅AWS Database Migration Service 用户指南中的创建源端点和目标端点。

要了解更多信息，请参阅数据迁移来源。

有关 S3 源终端节点，请参阅适用于 S3 的 DMS 源端点：创建。

有关 Mongo 数据库源端点的信息，请参阅适用于 MongoDB 的 DMS 源端点：正在创建。

适用于 MongoDB 的 DMS 源端点：正在创建

使用控制台创建 DMS Mongo 数据库源端点

AMS 控制台中此更改类型的屏幕截图：

工作原理：

1. 导航到 “创建 RFC” 页面：在 AMS 控制台的左侧导航窗格中，单击RFCs打开 RFCs 列表页面，然后单击 “创建 R FC”。
2. 在默认的 “浏览更改类型” 视图中选择常用更改类型 (CT)，或者在 “按类别选择” 视图中选择 CT。
   - 按更改类型浏览：您可以单击 “快速创建” 区域中的常用 CT，立即打开 “运行 RFC” 页面。请注意，您不能使用快速创建来选择较旧的 CT 版本。

     要进行排序 CTs，请使用卡片视图或表格视图中的所有更改类型区域。在任一视图中，选择一个 CT，然后单击 “创建 RFC” 打开 “运行 RFC” 页面。如果适用，“创建 RFC” 按钮旁边会出现 “使用旧版本创建” 选项。
   - 按类别选择：选择类别、子类别、项目和操作，CT 详细信息框将打开，并显示 “使用旧版本创建” 选项（如果适用）。单击 “创建 RFC” 打开 “运行 RFC” 页面。
3. 在 “运行 RFC” 页面上，打开 CT 名称区域以查看 CT 详细信息框。必须填写主题（如果您在 “浏览更改类型” 视图中选择 CT，则会为您填写此主题）。打开其他配置区域以添加有关 RFC 的信息。

   在执行配置区域中，使用可用的下拉列表或输入所需参数的值。要配置可选的执行参数，请打开其他配置区域。
4. 完成后，单击 “运行”。如果没有错误，则会显示成功创建的 RFC 页面，其中包含已提交的 RFC 详细信息和初始运行输出。
5. 打开运行参数区域以查看您提交的配置。刷新页面以更新 RFC 的执行状态。（可选）取消 RFC 或使用页面顶部的选项创建一个 RFC 的副本。

使用 CLI 创建 DMS Mongo 数据库源端点

工作原理：

1. 使用 Inline Create（您发出包含所有 RFC 和执行参数的`create-rfc`命令）或模板创建（创建两个 JSON 文件，一个用于 RFC 参数，一个用于执行参数），然后以这两个文件作为输入发出`create-rfc`命令。这里描述了这两种方法。
2. 提交带有返回的 RFC ID 的 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 命令。

   监控 RFC: `aws amscm get-rfc --rfc-id` *`ID`* 命令。

要检查更改类型版本，请使用以下命令：

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

> **Note**
>
> 您可以将任何`CreateRfc`参数与任何 RFC 一起使用，无论它们是否属于变更类型的架构的一部分。例如，要在 RFC 状态更改时收到通知，请将此行添加到请求的 RFC 参数部分（不是执行参数）。`--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"`有关所有 CreateRfc 参数的列表，请参阅《AMS 变更管理 API 参考》。

内联创建：

使用内联提供的执行参数（内联提供执行参数时使用转义引号）发出 create RFC 命令，然后提交返回的 RFC ID。例如，你可以用这样的东西替换内容：

```
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\"}}"
```

模板创建：

1. 将此更改类型的执行参数输出到名为 CreateDmsSeMongoParams .json 的 JSON 文件中。

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

2. 修改并保存执行参数 JSON 文件。例如，你可以用这样的东西替换内容：

```
{
"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. 将 JSON 模板输出到当前文件夹中的一个文件中；此示例将其命名为 CreateDmsSeMongoRfc .json：

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

4. 修改并保存 CreateDmsSeMongoRfc .json 文件。例如，你可以用这样的东西替换内容：

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

5. 创建 RFC，指定执行参数文件和 CreateDmsSeMongoRfc 文件：

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

您在响应中收到新 RFC 的 ID，并可以使用它来提交和监控 RFC。在您提交之前，RFC 仍处于编辑状态且无法启动。

提示

> **Note**
>
> 您最多可以添加 50 个标签，但要这样做，您必须启用其他配置视图。

AMS DMS 可以使用 Mongo 或任何关系数据库服务 (RDS) 作为源端点。有关 S3 源终端节点，请参阅适用于 S3 的 DMS 源端点：创建。

适用于 S3 的 DMS 源端点：创建

使用控制台创建 DMS S3 源端点

AMS 控制台中此更改类型的屏幕截图：

工作原理：

1. 导航到 “创建 RFC” 页面：在 AMS 控制台的左侧导航窗格中，单击RFCs打开 RFCs 列表页面，然后单击 “创建 R FC”。
2. 在默认的 “浏览更改类型” 视图中选择常用更改类型 (CT)，或者在 “按类别选择” 视图中选择 CT。
   - 按更改类型浏览：您可以单击 “快速创建” 区域中的常用 CT，立即打开 “运行 RFC” 页面。请注意，您不能使用快速创建来选择较旧的 CT 版本。

     要进行排序 CTs，请使用卡片视图或表格视图中的所有更改类型区域。在任一视图中，选择一个 CT，然后单击 “创建 RFC” 打开 “运行 RFC” 页面。如果适用，“创建 RFC” 按钮旁边会出现 “使用旧版本创建” 选项。
   - 按类别选择：选择类别、子类别、项目和操作，CT 详细信息框将打开，并显示 “使用旧版本创建” 选项（如果适用）。单击 “创建 RFC” 打开 “运行 RFC” 页面。
3. 在 “运行 RFC” 页面上，打开 CT 名称区域以查看 CT 详细信息框。必须填写主题（如果您在 “浏览更改类型” 视图中选择 CT，则会为您填写此主题）。打开其他配置区域以添加有关 RFC 的信息。

   在执行配置区域中，使用可用的下拉列表或输入所需参数的值。要配置可选的执行参数，请打开其他配置区域。

4. 完成后，单击 “运行”。如果没有错误，则会显示成功创建的 RFC 页面，其中包含已提交的 RFC 详细信息和初始运行输出。
5. 打开运行参数区域以查看您提交的配置。刷新页面以更新 RFC 的执行状态。（可选）取消 RFC 或使用页面顶部的选项创建一个 RFC 的副本。

使用 CLI 创建 DMS S3 源端点

工作原理：

1. 使用 Inline Create（您发出包含所有 RFC 和执行参数的`create-rfc`命令）或模板创建（创建两个 JSON 文件，一个用于 RFC 参数，一个用于执行参数），然后以这两个文件作为输入发出`create-rfc`命令。这里描述了这两种方法。
2. 提交带有返回的 RFC ID 的 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 命令。

   监控 RFC: `aws amscm get-rfc --rfc-id` *`ID`* 命令。

要检查更改类型版本，请使用以下命令：

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

> **Note**
>
> 您可以将任何`CreateRfc`参数与任何 RFC 一起使用，无论它们是否属于变更类型的架构的一部分。例如，要在 RFC 状态更改时收到通知，请将此行添加到请求的 RFC 参数部分（不是执行参数）。`--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"`有关所有 CreateRfc 参数的列表，请参阅《AMS 变更管理 API 参考》。

内联创建：

使用内联提供的执行参数（内联提供执行参数时使用转义引号）发出 create RFC 命令，然后提交返回的 RFC ID。例如，你可以用这样的东西替换内容：

```
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\"}"
```

模板创建：

1. 将此更改类型的执行参数输出到名为 s CreateDmsSe 3Params.json 的 JSON 文件中。

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

2. 修改并保存执行参数 JSON 文件。例如，你可以用这样的东西替换内容：

```
{
"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. 将 JSON 模板输出到当前文件夹中的一个文件中；此示例将其命名 CreateDmsSe为 s3rfc.json：

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

4. 修改并保存 CreateDmsSe s3rfc.json 文件。例如，你可以用这样的东西替换内容：

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

5. 创建 RFC，指定执行参数文件和 CreateDmsSe S3Rfc 文件：

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

   您在响应中收到新 RFC 的 ID，并可以使用它来提交和监控 RFC。在您提交之前，RFC 仍处于编辑状态且无法启动。

提示

> **Note**
>
> 您最多可以添加 50 个标签，但要这样做，您必须启用其他配置视图。

AMS DMS 可以使用 S3 或任何关系数据库服务 (RDS) Service 源端点。有关 Mongo 数据库源端点的信息，请参阅适用于 MongoDB 的 DMS 源端点：正在创建。

4： AWS DMS 目标端点：创建，为 S3 创建

您可以使用 AMS 控制台或 API/CLI 为各种数据库创建 AMS DMS 目标终端节点，我们提供了两个示例。

DMS 目标终端节点：正在创建

AMS DMS 可以使用 S3 或任何带有 MySQL、MariaDB、Oracle、Postgresql 或微软 SQL 的关系数据库服务 (RDS) 作为目标端点。

使用控制台创建 DMS 目标端点

AMS 控制台中此更改类型的屏幕截图：

工作原理：

1. 导航到 “创建 RFC” 页面：在 AMS 控制台的左侧导航窗格中，单击RFCs打开 RFCs 列表页面，然后单击 “创建 R FC”。
2. 在默认的 “浏览更改类型” 视图中选择常用更改类型 (CT)，或者在 “按类别选择” 视图中选择 CT。
   - 按更改类型浏览：您可以单击 “快速创建” 区域中的常用 CT，立即打开 “运行 RFC” 页面。请注意，您不能使用快速创建来选择较旧的 CT 版本。

     要进行排序 CTs，请使用卡片视图或表格视图中的所有更改类型区域。在任一视图中，选择一个 CT，然后单击 “创建 RFC” 打开 “运行 RFC” 页面。如果适用，“创建 RFC” 按钮旁边会显示 “使用旧版本创建” 选项。
   - 按类别选择：选择类别、子类别、项目和操作，CT 详细信息框将打开，并显示 “使用旧版本创建” 选项（如果适用）。单击 “创建 RFC” 打开 “运行 RFC” 页面。
3. 在 “运行 RFC” 页面上，打开 CT 名称区域以查看 CT 详细信息框。必须填写主题（如果您在 “浏览更改类型” 视图中选择 CT，则会为您填写此主题）。打开其他配置区域以添加有关 RFC 的信息。

   在执行配置区域中，使用可用的下拉列表或输入所需参数的值。要配置可选的执行参数，请打开其他配置区域。
4. 完成后，单击 “运行”。如果没有错误，则会显示成功创建的 RFC 页面，其中包含已提交的 RFC 详细信息和初始运行输出。
5. 打开运行参数区域以查看您提交的配置。刷新页面以更新 RFC 的执行状态。（可选）取消 RFC 或使用页面顶部的选项创建一个 RFC 的副本。

使用 CLI 创建 DMS 目标端点

工作原理：

1. 使用 Inline Create（您发出包含所有 RFC 和执行参数的`create-rfc`命令）或模板创建（创建两个 JSON 文件，一个用于 RFC 参数，一个用于执行参数），然后以这两个文件作为输入发出`create-rfc`命令。这里描述了这两种方法。
2. 提交带有返回的 RFC ID 的 RFC: `aws amscm submit-rfc --rfc-id` *ID* 命令。

   监控 RFC: `aws amscm get-rfc --rfc-id` *ID* 命令。

要检查更改类型版本，请使用以下命令：

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

> **Note**
>
> 您可以将任何`CreateRfc`参数与任何 RFC 一起使用，无论它们是否属于变更类型的架构的一部分。例如，要在 RFC 状态更改时收到通知，请将此行添加到请求的 RFC 参数部分（不是执行参数）。`--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"`有关所有 CreateRfc 参数的列表，请参阅《AMS 变更管理 API 参考》。

内联创建：

使用内联提供的执行参数（内联提供执行参数时使用转义引号）发出 create RFC 命令，然后提交返回的 RFC ID。例如，你可以用这样的东西替换内容：

```
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\"}}"
```

模板创建：

1. 将此更改类型的执行参数输出到名为 CreateDmsTeParams .json 的 JSON 文件中。

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

2. 修改并保存执行参数 JSON 文件。例如，你可以用这样的东西替换内容：

```
{
"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. 将 JSON 模板输出到当前文件夹中的一个文件中；此示例将其命名为 CreateDmsTeRfc .json：

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

4. 修改并保存 CreateDmsTeRfc .json 文件。例如，你可以用这样的东西替换内容：

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

5. 创建 RFC，指定执行参数文件和 CreateDmsTeRfc 文件：

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

您在响应中收到新 RFC 的 ID，并可以使用它来提交和监控 RFC。在您提交之前，RFC 仍处于编辑状态且无法启动。

提示

- 此更改类型现在是 2.0 版。
- AMS DMS 可以使用 S3 或任何带有 MySQL、MariaDB、Oracle、Postgresql 或微软 SQL 的关系数据库服务 (RDS) 作为目标端点。有关 S3 目标终端节点，请参阅S3 的 DMS 目标终端节点：创建。
- 有关更多信息，请参阅数据迁移目标。
- 您最多可以添加 50 个标签，但要这样做，您必须启用其他配置视图。

S3 的 DMS 目标终端节点：创建

使用控制台创建 DMS S3 目标端点

AMS 控制台中此更改类型的屏幕截图：

工作原理：

1. 导航到 “创建 RFC” 页面：在 AMS 控制台的左侧导航窗格中，单击RFCs打开 RFCs 列表页面，然后单击 “创建 R FC”。
2. 在默认的 “浏览更改类型” 视图中选择常用更改类型 (CT)，或者在 “按类别选择” 视图中选择 CT。
   - 按更改类型浏览：您可以单击 “快速创建” 区域中的常用 CT，立即打开 “运行 RFC” 页面。请注意，您不能使用快速创建来选择较旧的 CT 版本。

     要进行排序 CTs，请使用卡片视图或表格视图中的所有更改类型区域。在任一视图中，选择一个 CT，然后单击 “创建 RFC” 打开 “运行 RFC” 页面。如果适用，“创建 RFC” 按钮旁边会显示 “使用旧版本创建” 选项。
   - 按类别选择：选择类别、子类别、项目和操作，CT 详细信息框将打开，并显示 “使用旧版本创建” 选项（如果适用）。单击 “创建 RFC” 打开 “运行 RFC” 页面。
3. 在 “运行 RFC” 页面上，打开 CT 名称区域以查看 CT 详细信息框。必须填写主题（如果您在 “浏览更改类型” 视图中选择 CT，则会为您填写此主题）。打开其他配置区域以添加有关 RFC 的信息。

   在执行配置区域中，使用可用的下拉列表或输入所需参数的值。要配置可选的执行参数，请打开其他配置区域。
4. 完成后，单击 “运行”。如果没有错误，则会显示成功创建的 RFC 页面，其中包含已提交的 RFC 详细信息和初始运行输出。
5. 打开运行参数区域以查看您提交的配置。刷新页面以更新 RFC 的执行状态。（可选）取消 RFC 或使用页面顶部的选项创建一个 RFC 的副本。

使用 CLI 创建 DMS S3 目标端点

工作原理：

1. 使用 Inline Create（您发出包含所有 RFC 和执行参数的`create-rfc`命令）或模板创建（创建两个 JSON 文件，一个用于 RFC 参数，一个用于执行参数），然后以这两个文件作为输入发出`create-rfc`命令。这里描述了这两种方法。
2. 提交带有返回的 RFC ID 的 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 命令。

   监控 RFC: `aws amscm get-rfc --rfc-id` *`ID`* 命令。

要检查更改类型版本，请使用以下命令：

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

> **Note**
>
> 您可以将任何`CreateRfc`参数与任何 RFC 一起使用，无论它们是否属于变更类型的架构的一部分。例如，要在 RFC 状态更改时收到通知，请将此行添加到请求的 RFC 参数部分（不是执行参数）。`--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"`有关所有 CreateRfc 参数的列表，请参阅《AMS 变更管理 API 参考》。

内联创建：

使用内联提供的执行参数（内联提供执行参数时使用转义引号）发出 create RFC 命令，然后提交返回的 RFC ID。例如，你可以用这样的东西替换内容：

```
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\"}}"
```

模板创建：

1. 将此更改类型的执行参数输出到 JSON 文件；此示例将其命名为 s CreateDmsTe 3params.json：

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

2. 修改并保存执行参数 CreateDmsTe s3Params.json 文件。例如，你可以用这样的东西替换内容：

```
{
"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. 将 JSON 模板输出到当前文件夹中的一个文件中；此示例将其命名 CreateDmsTe为 s3rfc.json：

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

4. 修改并保存 CreateDmsTe s3rfc.json 文件。例如，你可以用这样的东西替换内容：

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

5. 创建 RFC，指定执行参数文件和 CreateDmsTe S3Rfc 文件：

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

   您在响应中收到新 RFC 的 ID，并可以使用它来提交和监控 RFC。在您提交之前，RFC 仍处于编辑状态且无法启动。

提示

> **Note**
>
> 您最多可以添加 50 个标签，但要这样做，您必须启用其他配置视图。

AMS 提供了一种单独的更改类型，用于为 S3 创建目标终端节点。有关更多信息，请参阅使用 Amazon S3 作为 AWS 数据库迁移服务的目标以及使用 Amazon S3 作为 AWS DMS 目标时的额外连接属性。

5： AWS DMS 复制任务：创建

您可以使用 AMS 控制台或 API/CLI 创建 AMS AWS DMS 复制任务。

创建 AWS DMS 复制任务

使用控制台创建 AWS DMS 复制任务

AMS 控制台中此更改类型的屏幕截图：

工作原理：

1. 导航到 “创建 RFC” 页面：在 AMS 控制台的左侧导航窗格中，单击RFCs打开 RFCs 列表页面，然后单击 “创建 R FC”。
2. 在默认的 “浏览更改类型” 视图中选择常用更改类型 (CT)，或者在 “按类别选择” 视图中选择 CT。
   - 按更改类型浏览：您可以单击 “快速创建” 区域中的常用 CT，立即打开 “运行 RFC” 页面。请注意，您不能使用快速创建来选择较旧的 CT 版本。

     要进行排序 CTs，请使用卡片视图或表格视图中的所有更改类型区域。在任一视图中，选择一个 CT，然后单击 “创建 RFC” 打开 “运行 RFC” 页面。如果适用，“创建 RFC” 按钮旁边会出现 “使用旧版本创建” 选项。
   - 按类别选择：选择类别、子类别、项目和操作，CT 详细信息框将打开，并显示 “使用旧版本创建” 选项（如果适用）。单击 “创建 RFC” 打开 “运行 RFC” 页面。
3. 在 “运行 RFC” 页面上，打开 CT 名称区域以查看 CT 详细信息框。必须填写主题（如果您在 “浏览更改类型” 视图中选择 CT，则会为您填写此主题）。打开 “其他配置” 区域以添加有关 RFC 的信息。

在执行配置区域中，使用可用的下拉列表或输入所需参数的值。要配置可选的执行参数，请打开其他配置区域。

4. 完成后，单击 “运行”。如果没有错误，则会显示成功创建的 RFC 页面，其中包含已提交的 RFC 详细信息和初始运行输出。
5. 打开运行参数区域以查看您提交的配置。刷新页面以更新 RFC 的执行状态。（可选）取消 RFC 或使用页面顶部的选项创建一个 RFC 的副本。

使用 CLI 创建 AWS DMS 复制任务

工作原理：

1. 使用 Inline Create（您发出包含所有 RFC 和执行参数的`create-rfc`命令）或模板创建（创建两个 JSON 文件，一个用于 RFC 参数，一个用于执行参数），然后以这两个文件作为输入发出`create-rfc`命令。这里描述了这两种方法。
2. 提交带有返回的 RFC ID 的 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 命令。

   监控 RFC: `aws amscm get-rfc --rfc-id` *`ID`* 命令。

要检查更改类型版本，请使用以下命令：

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

> **Note**
>
> 您可以将任何`CreateRfc`参数与任何 RFC 一起使用，无论它们是否属于变更类型的架构的一部分。例如，要在 RFC 状态更改时收到通知，请将此行添加到请求的 RFC 参数部分（不是执行参数）。`--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"`有关所有 CreateRfc 参数的列表，请参阅《AMS 变更管理 API 参考》。

内联创建：

使用内联提供的执行参数（内联提供执行参数时使用转义引号）发出 create RFC 命令，然后提交返回的 RFC ID。例如，你可以用这样的东西替换内容：

```
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}"
```

模板创建：

1. 将此更改类型的执行参数输出到 JSON 文件；此示例将其命名为 CreateDmsRtParams .json：

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

2. 修改并保存执行参数 JSON 文件。例如，你可以用这样的东西替换内容：

   ```
   {
   "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. 将 JSON 模板输出到当前文件夹中的一个文件中；此示例将其命名为 CreateDmsRtRfc .json：

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

4. 修改并保存 CreateDmsRtRfc .json 文件。例如，你可以用这样的东西替换内容：

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

5. 创建 RFC，指定执行参数文件和 CreateDmsRtRfc 文件：

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

   您在响应中收到新 RFC 的 ID，并可以使用它来提交和监控 RFC。在您提交之前，RFC 仍处于编辑状态且无法启动。

提示

您可以创建捕获三种不同类型的更改或数据的 AWS DMS 任务。有关更多信息，请参阅使用 AWS DMS 任务、创建任务和使用 A WS DMS 创建用于持续复制的任务。

AWS DMS 管理

AWS DMS 管理示例。

启动 AWS DMS 复制任务

使用控制台启动 AWS DMS 复制任务

AMS 控制台中此更改类型的屏幕截图：

工作原理：

1. 导航到创建 RFC 页面：在 AMS 控制台的左侧导航窗格中，单击RFCs打开 RFCs 列表页面，然后单击创建 R FC。
2. 在默认的 “浏览更改类型” 视图中选择常用更改类型 (CT)，或者在 “按类别选择” 视图中选择 CT。
   - 按更改类型浏览：您可以单击 “快速创建” 区域中的常用 CT，立即打开 “运行 RFC” 页面。请注意，您不能使用快速创建来选择较旧的 CT 版本。

要进行排序 CTs，请使用卡片视图或表格视图中的所有更改类型区域。在任一视图中，选择一个 CT，然后单击 “创建 RFC” 打开 “运行 RFC” 页面。如果适用，“创建 RFC” 按钮旁边会显示 “使用旧版本创建” 选项。

- 按类别选择：选择类别、子类别、项目和操作，CT 详细信息框将打开，并显示 “使用旧版本创建” 选项（如果适用）。单击 “创建 RFC” 打开 “运行 RFC” 页面。

3. 在 “运行 RFC” 页面上，打开 CT 名称区域以查看 CT 详细信息框。必须填写主题（如果您在 “浏览更改类型” 视图中选择 CT，则会为您填写此主题）。打开其他配置区域以添加有关 RFC 的信息。

   在执行配置区域中，使用可用的下拉列表或输入所需参数的值。要配置可选的执行参数，请打开其他配置区域。
4. 完成后，单击 “运行”。如果没有错误，则会显示成功创建的 RFC 页面，其中包含已提交的 RFC 详细信息和初始运行输出。
5. 打开运行参数区域以查看您提交的配置。刷新页面以更新 RFC 的执行状态。（可选）取消 RFC 或使用页面顶部的选项创建一个 RFC 的副本。

使用 CLI 启动 AWS DMS 复制任务

工作原理：

1. 使用 Inline Create（您发出包含所有 RFC 和执行参数的`create-rfc`命令）或模板创建（创建两个 JSON 文件，一个用于 RFC 参数，一个用于执行参数），然后以这两个文件作为输入发出`create-rfc`命令。这里描述了这两种方法。
2. 提交带有返回的 RFC ID 的 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 命令。

   监控 RFC: `aws amscm get-rfc --rfc-id` *`ID`* 命令。

要检查更改类型版本，请使用以下命令：

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

> **Note**
>
> 您可以将任何`CreateRfc`参数与任何 RFC 一起使用，无论它们是否属于变更类型的架构的一部分。例如，要在 RFC 状态更改时收到通知，请将此行添加到请求的 RFC 参数部分（不是执行参数）。`--notification "{\"Email\": {\"EmailRecipients\" :`

[\"email@example.com\"]}}"有关所有 CreateRfc 参数的列表，请参阅《AMS 变更管理 API 参考》。

内联创建：

使用内联提供的执行参数（内联提供执行参数时使用转义引号）发出 create RFC 命令，然后提交返回的 RFC ID。例如，你可以用这样的东西替换内容：

```
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\":[\"\"]}}"
```

模板创建：

1. 将此更改类型的执行参数输出到 JSON 文件；此示例将其命名为 StartDmsRtParams .json：

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

2. 修改并保存执行参数 JSON 文件。例如，你可以用这样的东西替换内容：

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

```
}
```

3. 将 JSON 模板输出到当前文件夹中的一个文件中；此示例将其命名为 StartDmsRtRfc .json：

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

4. 修改并保存 StartDmsRtRfc .json 文件。例如，你可以用这样的东西替换内容：

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

5. 创建 RFC，指定执行参数文件和 StartDmsRtRfc 文件：

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

   您在响应中收到新 RFC 的 ID，并可以使用它来提交和监控 RFC。在您提交之前，RFC 仍处于编辑状态且无法启动。

提示

您可以使用 AMS 控制台或 AMS API/CLI 启动 AWS DMS 复制任务。有关更多信息，请参阅处理 AWS DMS 任务。

停止 AWS DMS 复制任务

使用控制台停止 AWS DMS 复制任务

AMS 控制台中此更改类型的屏幕截图：

工作原理：

1. 导航到创建 RFC 页面：在 AMS 控制台的左侧导航窗格中，单击RFCs打开 RFCs 列表页面，然后单击创建 R FC。
2. 在默认的 “浏览更改类型” 视图中选择常用更改类型 (CT)，或者在 “按类别选择” 视图中选择 CT。
   - 按更改类型浏览：您可以单击 “快速创建” 区域中的常用 CT，立即打开 “运行 RFC” 页面。请注意，您不能使用快速创建来选择较旧的 CT 版本。

要进行排序 CTs，请使用卡片视图或表格视图中的所有更改类型区域。在任一视图中，选择一个 CT，然后单击 “创建 RFC” 打开 “运行 RFC” 页面。如果适用，“创建 RFC” 按钮旁边会显示 “使用旧版本创建” 选项。

- 按类别选择：选择类别、子类别、项目和操作，CT 详细信息框将打开，并显示 “使用旧版本创建” 选项（如果适用）。单击 “创建 RFC” 打开 “运行 RFC” 页面。

3. 在 “运行 RFC” 页面上，打开 CT 名称区域以查看 CT 详细信息框。必须填写主题（如果您在 “浏览更改类型” 视图中选择 CT，则会为您填写此主题）。打开其他配置区域以添加有关 RFC 的信息。

   在执行配置区域中，使用可用的下拉列表或输入所需参数的值。要配置可选的执行参数，请打开其他配置区域。
4. 完成后，单击 “运行”。如果没有错误，则会显示成功创建的 RFC 页面，其中包含已提交的 RFC 详细信息和初始运行输出。
5. 打开运行参数区域以查看您提交的配置。刷新页面以更新 RFC 的执行状态。（可选）取消 RFC 或使用页面顶部的选项创建一个 RFC 的副本。

使用 CLI 停止 AWS DMS 复制任务

工作原理：

1. 使用 Inline Create（您发出包含所有 RFC 和执行参数的`create-rfc`命令）或模板创建（创建两个 JSON 文件，一个用于 RFC 参数，一个用于执行参数），然后以这两个文件作为输入发出`create-rfc`命令。这里描述了这两种方法。
2. 提交带有返回的 RFC ID 的 RFC: `aws amscm submit-rfc --rfc-id` *`ID`* 命令。

   监控 RFC: `aws amscm get-rfc --rfc-id` *`ID`* 命令。

要检查更改类型版本，请使用以下命令：

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

> **Note**
>
> 您可以将任何`CreateRfc`参数与任何 RFC 一起使用，无论它们是否属于变更类型的架构的一部分。例如，要在 RFC 状态更改时收到通知，请将此行添加到请求的 RFC 参数部分（不是执行参数）。`--notification "{\"Email\": {\"EmailRecipients\" :`

[\"email@example.com\"]}}"有关所有 CreateRfc 参数的列表，请参阅《AMS 变更管理 API 参考》。

内联创建：

使用内联提供的执行参数（内联提供执行参数时使用转义引号）发出 create RFC 命令，然后提交返回的 RFC ID。例如，你可以用这样的东西替换内容：

```
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\"]}}"
```

模板创建：

1. 将此更改类型的执行参数输出到 JSON 文件；此示例将其命名为 StopDmsRtParams .json：

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

2. 修改并保存执行参数 JSON 文件。例如，你可以用这样的东西替换内容：

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

3. 将 JSON 模板输出到当前文件夹中的一个文件中；此示例将其命名为 StopDmsRtRfc .json：

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

4. 修改并保存 StopDmsRtRfc .json 文件。例如，你可以用这样的东西替换内容：

   ```
   {
     "ChangeTypeId": "ct-1vd3y4ygbqmfk",
   ```

```
  "ChangeTypeVersion": "1.0",
  "Title": "Stop DMS Replication Task"
}
```

5. 创建 RFC，指定执行参数文件和 StopDmsRtRfc 文件：

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

您在响应中收到新 RFC 的 ID，并可以使用它来提交和监控 RFC。在您提交之前，RFC 仍处于编辑状态且无法启动。

提示

您可以使用 AMS 控制台或 AMS API/CLI 停止 DMS 复制任务。有关更多信息，请参阅处理 AWS DMS 任务。

将数据库 (DB) 导入到适用于微软 SQL Server 的 AMS RDS

> **Note**
>
> AMS API/CLI （ amscm 和 amsskms ）终端节点位于 AWS 弗吉尼亚北部区域。`us-east-1`根据您的身份验证设置方式以及您的账户和资源所在的 AWS 区域，您可能需要在发出命令`--region us-east-1`时进行添加。如果这是您的身份验证方法`--profile saml`，则可能还需要添加。

将数据库导入到 AMS RDS for SQL Server 的过程依赖于作为更改请求 (CTs) 提交的 AMS 更改类型 (RFCs)，并使用 Amazon RDS API 参数作为输入。 MicroSoft SQL Server 是一个关系数据库管理系统 (RDBMS)。要了解更多信息，另请参阅：亚马逊关系数据库服务 (Amazon RDS) 和 rds 或 Amazon RDS API 参考。

> **Note**
>
> 在继续下一步之前，请确保每个 RFC 都成功完成。

高级导入步骤：

1. 将本地源 MS SQL 数据库备份到.bak（备份）文件中
2. 将.bak 文件复制到传输（加密）亚马逊简单存储服务 (S3) 存储桶中
3. 将.bak 导入目标 Amazon RDS MS SQL 实例上的新数据库

要求：

- AMS 中的 MS SQL RDS 堆栈
- 带还原选项的 RDS 堆栈 (SQLSERVER_BACKUP_RESTORE)
- 公交 S3 存储桶
- 具有存储桶访问权限的 IAM 角色允许 Amazon RDS 担任该角色
- 安装了 MS SQL Management Studio 以管理 RDS 的 EC2 实例（可以是本地工作站）

设置

完成这些任务以开始导入过程。

1. 使用部署 | 高级堆栈组件 | RDS 数据库堆栈 | 创建 (ct-2z60dyvto9g6c) 提交 RFC 以创建 RDS 堆栈。请勿在创建请求中使用目标数据库名称（`RDSDBName`参数），目标数据库将在导入过程中创建。确保留出足够的空间（`RDSAllocatedStorage`参数）。有关执行此操作的详细信息，请参阅 AMS 变更管理指南 RDS DB Stack | Creat e。
2. 使用部署 | 高级堆栈组件 | S3 存储 | 创建 (ct-1a68ck03fn98r) 提交 RFC 以创建 Transit S3 存储桶（如果尚不存在）。有关执行此操作的详细信息，请参阅《AMS 变更管理指南 S3 存储 | 创建》。
3. 提交管理 | 其他 | 其他 | 更新 (ct-1e1xtak34nx76) RFC 以实施包含以下详细信息的内容：`customer_rds_s3_role`

   在控制台中，执行以下操作：

   - 主题：“要支持 MS SQL Server 数据库导入，请使用此帐户实现`customer_rds_s3_role`。
   - Transit S3 存储桶名称：*BUCKET_NAME*。
   - 联系信息:*EMAIL*.

   使用用于 CLI 的 ImportDbParams .json 文件：

```
{
       "Comment": "{"Transit S3 bucket name":"BUCKET_NAME"}",
       "Priority": "High"
```

```
    }
}
```

4. 提交管理 | 其他 | 其他 | 更新 RFC，请求 AMS 为步骤 1 中创建的 RDS 设置SQLSERVER_BACKUP_RESTORE选项（在此请求中使用步骤 1 输出中的堆栈 ID 以及此请求customer_rds_s3_role中的 IAM 角色）。
5. 提交 RFC 以创建 EC2 实例（您可以使用任何现有 EC2 或本地工作站/实例），然后在该实例上安装 Microsoft SQL Management Studio。

导入数据库

要导入数据库 (DB)，请执行以下步骤。

1. 使用 MS SQL 原生备份和恢复备份源本地数据库（参见 SQL S erver 中对本机备份和还原的支持）。运行该操作后，您应该有一个.bak（备份）文件。
2. 使用 AWS S3 CLI 或 AWS S3 控制台将.bak 文件上传到现有传输 S3 存储桶。有关传输 S3 存储桶的信息，请参阅使用加密保护数据。
3. 将.bak 文件导入您的目标 RDS for SQL Server MS SQL 实例上的新数据库（有关类型的详细信息，请参阅适用于 M ySQL 的 Amazon RDS 实例类型）：

   a. 登录 EC2 实例（本地工作站）并打开 MS SQL Management Studio

   b. 连接到在步骤 #1 中作为先决条件创建的目标 RDS 实例。按照以下步骤进行连接：连接到运行 Microsoft SQL Server 数据库引擎的数据库实例

   c. 使用新的结构化查询语言 (SQL) 查询启动导入（恢复）作业（有关 SQL 查询的详细信息，请参阅 SQL 简介）。目标数据库名称必须是新的（不要使用与之前创建的数据库相同的名称）。未加密的示例：

   ```
   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. 通过在单独的窗口中运行以下查询，定期检查导入任务的状态：

   ```
   exec msdb.dbo.rds_task_status;
   ```

   如果状态更改为 Failed，请在消息中查找失败详情。

清理

导入数据库后，可能需要删除不必要的资源，请按照以下步骤操作。

1. 从 S3 存储桶中删除备份文件 (.bak)。您可以使用 S3 控制台来执行此操作。有关从 S3 存储桶中删除对象的 CLI 命令，请参阅 AWS CLI 命令参考中的 rm。
2. 如果您不打算使用 S3 存储桶，请将其删除。有关执行此操作的步骤，请参阅删除堆栈。
3. 如果你不打算导入 MS SQL，请提交管理 | 其他 | 其他 | 更新 (ct-0xdawir96cy7k) RFC 并请求 AMS 删除 IAM 角色。`customer_rds_s3_role`

在 AMS 中分层和绑定应用程序部署

Tier and Tie 部署是指使用单独的方法独立创建 RFCs、配置和部署堆栈资源，并在逐步将堆栈组件相互关联时使用堆栈组件的资源。 IDs

例如，要在负载均衡器后面部署高可用性（冗余）网站和数据库，请使用 Tier and Tie 方法提交 RFCs 数据库、负载均衡器、两个 EC2实例或 Auto Scaling 组，然后使用您创建的 ELB 的 ID 配置 EC2 实例或 Auto Scaling 组。

资源部署后，您可以提交安全组创建更改，以允许资源与数据库通信。有关创建安全组的详细信息，请参阅创建安全组。

AMS 中的全栈应用程序部署

全栈部署是指您提交带有 CT 的 RFC，该CT可以立即创建和配置所需的一切。例如，要部署刚才描述的高可用性网站（EC2 实例、负载均衡器和数据库），您可以使用 CT 来创建和配置 Auto Scaling 组、负载均衡器、数据库以及所有实例作为堆栈运行所需的安全组设置。接下来将介绍两 CTs 个 AMS 执行此操作的示例。

- 高可用性双层堆栈 (ct-06mjngx5flwto)：此更改类型允许您创建堆栈并配置 Auto Scaling 组、RDS 支持的数据库、Load Balancer 以及应用程序和配置。 CodeDeploy 请注意，负载均衡器不被视为一个层，因为它作为网络设备在多个应用程序之间共享，其 CodeDeploy 功能也被视为设备。此外，它还会创建一个可用于 CodeDeploy部署 CodeDeploy 应用程序的部署组（使用您为应用程序指定的名称）。系统会自动创建允许资源协同运行的安全组设置。
- 高可用性单层堆栈（ct-09t6q7j9v5hrn）：此更改类型允许您创建堆栈并配置 Auto Scaling 组和应用程序负载均衡器。系统会自动创建允许资源协同工作的安全组设置。

处理配置变更类型 (CTs)

AMS 对您的托管基础设施负责，要进行更改，您必须提交包含正确的 CT 分类（类别、子类别、项目和操作）的 RFC。本节介绍如何查找 CTs、确定是否有任何符合您需求的 CT，以及如何申请新的 CT（如果没有 CT）。

查看现有的 CT 是否符合您的要求

确定要使用 AMS 部署的内容后，下一步就是研究现有解决方案 CTs 和 CloudFormation 模板，看看是否已经存在解决方案。

创建 RFC 时，必须指定 CT。您可以使用 AWS 管理控制台 或 AMS API/CLI。接下来将介绍使用两者的示例。

您可以使用控制台或 API/CLI 来查找更改类型 ID (CT) 或版本。有两种方法，要么是搜索，要么是选择分类。对于这两种选择类型，您可以通过选择 “最常用”、“最近使用” 或 “按字母顺序” 对搜索进行排序。

YouTube 视频：如何使用 AWS Managed Services CLI 创建 RFC，在哪里可以找到 CT 架构？

在 AMS 控制台中，在 RFCs-> 创建 RFC 页面上：

- 选择 “按更改类型浏览”（默认）后，可以：
  - 使用 “快速创建” 区域从 AMS 最受欢迎的 AMS 中进行选择 CTs。点击标签，将打开 “运行 RFC” 页面，并自动为您填写 “主题” 选项。根据需要完成其余选项，然后单击 “运行” 提交 RFC。
  - 或者，向下滚动到 “所有变更类型” 区域并开始在选项框中键入 CT 名称，您不必输入确切或完整的更改类型名称。您还可以通过输入相关词语按更改类型 ID、分类或执行模式（自动或手动）搜索 CT。

    选择默认卡片视图后，匹配的 CT 卡片会在您键入时出现，选择一张卡片并单击 “创建 RFC”。选择表格视图后，选择相关的 CT，然后单击 “创建 RFC”。两种方法都会打开 “运行 RFC” 页面。
- 或者，要浏览更改类型选择，请单击页面顶部的按类别选择以打开一系列下拉选项框。
- 选择 “类别”、“子类别”、“物料” 和 “工序”。该更改类型的信息框显示在页面底部显示一个面板。
- 准备就绪后，按 Enter，将显示匹配的更改类型列表。
- 从列表中选择更改类型。该更改类型的信息框显示在页面底部。
- 选择正确的更改类型后，选择 “创建 RFC”。

> **Note**
>
> 必须安装 AMS CLI 才能使这些命令生效。要安装 AMS API 或 CLI，请前往 AMS 控制台开发者资源页面。有关 AMS CM API 或 AMS SKMS API 的参考资料，请参阅《用户指南》中的 “AMS 信息资源” 部分。您可能需要添加身份验证`--profile`选项；例如，`aws amsskms ams-cli-command --profile SAML`。您可能还需要添加该`--region`选项，因为所有 AMS 命令都将使用 us-east-1；例如。`aws amscm ams-cli-command --region=us-east-1`

> **Note**
>
> AMS API/CLI（amscm 和 amsskms）终端节点位于 AWS 弗吉尼亚北部区域。`us-east-1`根据您的身份验证设置方式以及您的账户和资源所在的 AWS 区域，您可能需要在发出命令`--region us-east-1`时进行添加。如果这是您的身份验证方法`--profile saml`，则可能还需要添加。

要使用 AMS CM API（参见 ListChangeTypeClassificationSummaries）或 CLI 搜索更改类型，请执行以下操作：

您可以使用筛选器或查询进行搜索。该 ListChangeTypeClassificationSummaries 操作具有`Category`、`SubcategoryItem`、和的 “筛选器” 选项`Operation`，但这些值必须与现有值完全匹配。要在使用 CLI 时获得更灵活的结果，可以使用`--query`选项。

使用 AMS CM API/CLI 更改类型筛选

属性	有效值	有效/默认条件	备注
ChangeTypeId	任何表示 a 的字符串 ChangeTypeId（例如：ct-abc123xyz7890）	Equals	有关更改类型 IDs，请参阅更改类型参考。 有关变更类型 IDs，请参阅查找变更类型或 CSIO。
类别	任何自由格式的文本	包含	不支持每个字段中的正则表达式。不区分大小写的搜索
子类别			

属性	有效值	有效/默认条件	备注
Item			
操作			

1. 以下是一些商品变更类型分类的示例：

   以下命令列出了所有更改类型类别。

   ```
   aws amscm list-change-type-categories
   ```

   以下命令列出了属于指定类别的子类别。

   ```
   aws amscm list-change-type-subcategories --category CATEGORY
   ```

   以下命令列出了属于指定类别和子类别的项目。

   ```
   aws amscm list-change-type-items --category CATEGORY --subcategory SUBCATEGORY
   ```

2. 以下是一些使用 CLI 查询搜索变更类型的示例：

   以下命令在 CT 分类摘要中搜索项目名称中包含 “S3” 的摘要，并以表格形式创建类别、子类别、项目、操作和更改类型 ID 的输出。

   ```
   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. 然后，您可以使用更改类型 ID 获取 CT 架构并检查参数。以下命令将架构输出到名为 creates3Params.schema.json 的 JSON 文件中。

```
aws amscm get-change-type-version --change-type-id "ct-1a68ck03fn98r"
 --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
 CreateS3Params.schema.json
```

有关使用 CLI 查询的信息，请参阅如何使用--query 选项过滤输出和查询语言参考JMESPath 规范。

4. 获得变更类型 ID 后，我们建议您验证变更类型的版本，以确保它是最新版本。使用以下命令查找指定更改类型的版本：

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

要查找AutomationStatus特定更改类型的，请运行以下命令：

```
aws amscm --profile saml get-change-type-version --change-type-id CHANGE_TYPE_ID --
query "ChangeTypeVersion.{AutomationStatus:AutomationStatus.Name}"
```

要查找ExpectedExecutionDurationInMinutes特定更改类型的，请运行以下命令：

```
aws amscm --profile saml get-change-type-version --change-type-id ct-14027q0sjyt1h
 --query "ChangeTypeVersion.{ExpectedDuration:ExpectedExecutionDurationInMinutes}"
```

找到您认为合适的 CT 后，请查看与其关联的执行参数 JSON 架构，以了解它是否可以解决您的用例。

使用此命令将 CT 架构输出到以 CT 命名的 JSON 文件中；此示例输出创建 S3 存储架构：

```
aws amscm get-change-type-version --change-type-id "ct-1a68ck03fn98r"
--query "ChangeTypeVersion.ExecutionInputSchema" --output text >
CreateBucketParams.json
```

让我们仔细看看这个架构提供了什么。

S3 存储桶创建架构

```
{
  "$schema": "http://json-schema.org/draft-04/sch
ema#",
```

架构以 CT（“描述”）开头，它告诉你架构的用途。在本例中，用于创建 S3 存储堆栈。

```
"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":{
      "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": {
```

接下来，您可以指定必填属性和可选属性。给出了默认的属性值。所需的属性列在架构的末尾。

在该 StackTemplateId 区域中，您可以看到此 CT 和架构有一个特定的堆栈模板，其 ID 是必填的属性值。

该架构允许您标记正在创建的堆栈，以用于内部记账。此外，某些选项（例如备份）需要标记 key: Backup 和 value: True。如需深入了解，请阅读标记您的 Amazon EC2 资源。

```
            "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",
      "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]$",
```

您可以在 CT JSON 架构的 “参数” 部分中提供执行参数。

对于此架构，只有 ACL 和 BucketName 是必需的执行参数。

```
          "minLength": 3,
          "maxLength": 63
        }
      },
      "additionalProperties": false,
      "required": [
        "AccessControl",
        "BucketName"
      ]
    }
  },
  "additionalProperties": false,
  "required": [
    "Description",
    "VpcId",
    "StackTemplateId",
    "Name",
    "TimeoutInMinutes",
    "Parameters"
  ]
}
```

申请新的 CT

检查架构后，您可能会认为它没有提供足够的参数来创建所需的部署。如果是这样的话，请检查现有的 CloudFormation 模板，找到一个更接近你想要的模板。知道需要哪些其他参数后，提交管理 | 其他 | 其他 | 创建 CT。

> **Note**
>
> All Other | Other Create and Update 会 CTs 收到 AMS 操作员的注意，他们将与您联系以讨论新的 CT。

要提交新 CT 申请，请通过常规访问AMS控制台，AWS 管理控制台然后按照以下步骤操作。

1. 在左侧导航栏中，单击RFCs。

   RFCs 仪表板页面打开。
2. 单击创建。

将打开 “创建更改请求” 页面。

3. 在 “类别” 下拉列表中选择 “管理”，在 “子类别和项目” 中选择 “其他”。对于操作，选择创建。RFC 需要获得批准才能实施。
4. 输入您为何需要 CT 的信息，例如：请求修改后的 Create S3 存储 CT，允许根据现有的 Create S3 存储 CT 进行自定义 ACLs。这应该会生成一个新的 CT：部署 | 高级堆栈组件 | S3 存储 | 创建 S3 自定义 ACL。这个新的 CT 可能会公开。
5. 单击 Submit (提交)。

   您的 RFC 会显示在 RFC 控制面板上。

测试新 CT

AWS Managed Services 创建了新的 CT 后，您可以通过提交 RFC 对其进行测试。如果您与 AMS 合作预先批准了新的 CT，那么您只需遵循标准的 RFC 提交并观察结果即可（有关提交的详细信息 RFCs，请参阅创建和提交 R FC）。如果新 CT 未获得预先批准（您要确保它永远不会在未经明确批准的情况下运行），那么每次要运行时，都需要与 AMS 讨论其实施情况。

快速入门

主题

使用 AMS 变更类型的组合，您可以完成复杂的任务。

您可以使用 AMS 变更管理系统为多账户着陆区 (MALZ) 或单账户着陆区 (SALZ) 账户设置 AMS 资源调度器。过程各不相同。此外，还可以进行文件传输和跨账户快照。

AMS 资源调度器快速入门

使用此快速入门指南实现 AMS 资源调度器，这是一种基于标签的实例调度程序，可在 AMS Advanced 中节省成本。

AMS 资源计划程序基于 AWS 实例计划程序。

AMS 资源调度器术语

在开始之前，最好先熟悉 AMS 资源调度器术语：

- 周期：每个计划必须至少包含一个时间段，用于定义实例应运行的时间。一个时间表可以包含多个时段。当计划中使用多个时段时，当至少有一个周期规则为真时，资源调度器会应用相应的开始操作。
- timezon e：有关将在稍后引用的DefaultTimezone参数中使用的可接受时区值列表，请参阅 TZ 数据库时区列表的 TZ 列。
- hibernat e：设置为 true 时，启用休眠功能并满足休眠要求的 EC2 实例将被休眠 ()。suspend-to-disk检查 EC2 控制台以了解您的实例是否已启用休眠功能。对于运行亚马逊 Linux 的已停止亚马逊 EC2 实例，请使用休眠模式。
- 强制：如果设置为 true，则根据定义的计划，如果资源在运行期之外手动启动，则资源调度器会停止该资源；如果资源在运行期间手动停止，则资源调度器会启动该资源。
- r@@ etain_run ning：设置为 true 时，如果实例是在周期开始之前手动启动的，则防止资源调度器在运行周期结束时停止该实例。例如，如果配置的时间段为上午 9 点至下午 5 点的实例在上午 9 点之前手动启动，则资源调度器不会在下午 5 点停止该实例。
- ssm-maintenance-window：将 AWS Systems Manager 维护时段作为运行周期添加到计划中。当您指定与您部署的堆栈位于同一账户和 AWS 区域中的维护时段名称来调度 Amazon EC2 实例时，如

果没有其他运行周期指定该实例应运行，并且维护事件已完成，则资源调度器将在维护时段开始之前启动实例，并在维护时段结束时停止实例。

资源调度器使用您在初始配置期间指定的 AWS Lambda 频率来确定在维护时段之前多久可以启动您的实例。如果将频率 AWS CloudFormation 参数设置为 10 分钟或更短，则资源计划程序会在维护时段前 10 分钟启动实例。如果您将频率设置为大于 10 分钟，则资源计划程序启动实例的时间与您指定的频率相同。例如，如果您将 Systems Manager 的维护时段频率设置为 30 分钟，则资源调度器会在维护时段前 30 分钟启动实例。

有关更多信息，请参阅AWS Systems Manager 维护窗口。

- o@@ verride-stat us：暂时覆盖资源调度器配置的计划启动和停止操作。如果将该字段设置为 r unning，则资源调度器将启动但不会停止适用的实例。实例会一直运行，直到您手动将其停止。如果您将覆盖状态设置为已停止，则资源调度器会停止但不会启动适用的实例。除非您手动启动实例，否则该实例才会运行。

AMS 资源调度器实现

要部署 AMS 资源调度程序解决方案，请按照以下步骤操作。

1. 提交部署 | AMS 资源调度器 | 解决方案 | 部署 (ct-0ywnhc8e 5k9z5) RFC 并提供以下参数：
   - SchedulingActive：“是” 表示启用资源调度，“否” 表示禁用。默认值为是。
   - ScheduledServices：输入以逗号分隔的服务列表，以便为其安排资源。有效值包括自动缩放、ec2 和 rds 的组合。默认为自动缩放、ec2、rds。
   - TagName：将资源计划架构与服务资源关联的标签密钥的名称。默认为 “计划”。

     > **Note**
     >
     > 您的资源调度器部署将仅对带有此标签的资源进行操作。

   - DefaultTimezone：将用作默认时区的时区名称，格式为 US/Pacific。默认为 UTC。
2. 在收到第一步中的 RFC 成功执行的确认后，您可以提交 Perio d | Add 变更类型。
3. 最后，提交 RFC，为在第二步中创建的时间段添加时间表。使用 “时间表” | “添加更改类型”。

AMS 资源调度程序的实现和使用 FAQs

有关 AMS 资源调度程序的常见问题。

问：如果我启用了休眠但 EC2 实例不支持休眠会怎样？

答：休眠会将实例内存 (RAM) 中的内容保存到您的亚马逊 Elastic Block Store (Amazon EBS) 根卷中。如果将此字段设置为 true，则当资源调度器停止实例时，实例将处于休眠状态。

如果您将 Resource Scheduler 设置为使用休眠，但您的实例未启用休眠功能，或者它们不符合休眠先决条件，则资源调度器会记录警告，并且实例将在不进入休眠状态的情况下停止。有关更多信息，请参阅休眠您的实例。

问：如果我同时设置了 o verride_status 和强制执行会发生什么？

答：如果您将 o verride_stat us 设置为正在运行并将强制设置为 true（防止在运行期之外手动启动实例），则资源调度器会停止该实例。

如果您将 o verride_stat us 设置为已停止，并将强制设置为 true（防止实例在运行期间被手动停止），则资源调度器会重新启动该实例。

> **Note**
>
> 如果强制执行为 false，则应用配置的覆盖行为。

问：部署 AMS 资源调度器后，如何在我的账户中禁用或启用资源调度器？

答：要禁用或启用 AMS 资源调度器，请执行以下操作：

- 禁用：使用 “状态” | “禁用” 创建 RFC。请务必将设置为 “禁SchedulerState用”
- 启用：使用 “状态” | “启用” 创建 RFC。请务必将设置为 “启SchedulerState用”

问：如果 AMS 资源调度程序期限在我的修补维护窗口内，会发生什么？

答：资源调度器根据其配置的计划运行。如果将其配置为在进行修补时停止实例，则它将停止该实例，除非在修补开始之前将修补窗口添加为计划中的一段时间。换句话说，除非配置了指定的时段，否则资源调度器不会自动启动任何已停止的实例进行修补。为避免与修补维护时段发生冲突，请将分配给修补的时间窗口作为周期添加到资源调度器计划中。要向现有计划添加周期，请使用周期 | 添加创建 RFC。

问：如果我需要为不同的 EC2 实例设置不同的计划，我能否在我的账户中设置多个计划？

答：是的，您可以创建多个计划。根据要求，每个时间表可以有多个时段。在账户中启用 AMS 资源调度器后，将配置标签密钥。例如，如果标签键为 “计划”，则标签值可能会因与 AMS 资源调度器的计划

名称相对应的不同计划而有所不同。要添加新计划，您可以使用管理 | AMS 资源调度器 | 计划 | 添加 (ct-2bxelbn765ive) 更改类型创建 RFC，请参阅计划 | 添加。

问：在哪里可以找到 AMS 资源调度器支持的所有不同更改类型？

答：AMS 有资源调度器更改类型，用于将 AMS 资源调度器部署到您的账户；启用或禁用它；定义、添加、更新和删除要与之配合使用的计划和时段；以及描述（获取详细描述）计划和时段。

设置跨账户备份（区域内）

AWS Backup 支持将快照从一个账户复制到同一 AWS 区域内的另一个账户，前提是这两个账户位于同一 AWS 组织内。例如，在 AMS Advanced 多账户着陆区 (MALZ) 中，您可以使用此快速入门功能在同一 AWS 区域内设置跨账户快照副本。

有关更多信息，请参阅 AWS Backup 和 AWS Organizations 推出跨账户备份功能

您可以跨账户复制快照以进行灾难恢复 (DR)。为了保护数据，您可能需要将快照保存在同一 AWS 区域内，但要跨越账户边界。

概述：

简而言之，以下是在 AMS 中进行跨账户备份的步骤：

- 创建目标账户，在托管 AMS landing zone 的 AWS 区域托管备份（步骤 1）
- 创建 KMS 密钥以加密目标账户中的备份（步骤 3）
- 在与 AMS Advanced 着陆区域相同的目标账户中创建备份保管库（步骤 4）
- 在您的管理账户中启用跨账户设置（步骤 5）
- 创建或修改源账户备份计划和规则（步骤 6）

> **Note**
>
> 确保源账户和目标账户位于同一个区域。如果要跨区域复制备份，请联系您的 CA 或 CSDM。

要启用和设置跨账户备份，请执行以下操作：

1. 创建用于托管备份的目标帐户；如果您已经有这样的帐户，则可以跳过此步骤。要创建账户，请使用部署 | 托管着陆区 | 管理账户 | 创建应用程序账户（使用 VPC）更改类型 (ct-1zdasmc2ewzrs) 从您的管理付款人账户提交 RFC。
2. [可选] 如果源账户（例如 Prod）中的资源或快照已加密，请与目标账户共享用于加密的 KMS 密钥。为此，请使用管理 | 高级堆栈组件 | KMS 密钥 | 更新更改类型 (ct-3ovo7px2vsa6n) 提交 RFC。
3. 在目标账户中，创建用于 Backup Vault 加密的 KMS 密钥。为此，请使用部署 | 高级堆栈组件 | KMS 密钥 | 创建（自动）更改类型 (ct-1d84keiri1jhg) 提交 RFC。
4. 在目标账户中，使用之前创建的密钥创建 Backup Vault。AWS Backup Vaults 可以通过使用 CFN 提取自动更改类型 “部署 | 摄取 | CloudFormation 模板中的堆栈 | 创建” (ct-36cn2avfrrj9v) 创建。在同一个请求中，需要修改文件库访问策略以允许源账户访问文件库。以下是策略示例：

   Backup Vault 的示例 CloudFormation 模板：

```
{
  "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. 在您的管理付款人账户中，启用跨账户备份。为此，请使用管理 | AWS Backup | Backup 计划 | 启用跨账户复制（管理账户）更改类型 (ct-2yja7ihh30ply) 提交 RFC。
6. 最后，从备份来源的源帐户中，创建管理备份的备份计划规则或规则，以跨账户复制快照。为此，请使用部署 | AWS Backup | Backup 计划 | 创建更改类型 (ct-2 hyozbpa0sx0m) 提交 RFC。如果您需要更新现有的备份计划，请使用管理 | 其他 | 其他 | 更新更改类型 (ct-0xdawir96cy7k) 提交 RFC，其中包含以下信息：

   1. 备份计划名称以及要更新的规则名称。
   2. destination/ICE 账户备份保管库 ARN。
   3. days/months 您希望将快照保留在目标 ICE 保管库中的保留期。

教程

主题

以下教程详细介绍了创建具有高可用性 (ct-06mjngx5flwto) 的两层堆栈、使用 CLI、使用控制台以及部署 Linux 或 RHEL Amazon Auto Scaling 组 (ASG) 的步骤。 EC2 每个 tier-and-tie教程（一个用于控制台，一个用于 CLI）紧随其后，它使用单独 CTs的教程，其创建顺序允许您在创建资源时将其绑定在一起。

所有 CT 选项（包括）的描述都 ChangeTypeId 可以在managedservices/latest/ctref/更改类型参考中找到。

控制台教程：高可用性双层堆栈 (Linux/RHEL)

本节介绍如何使用 AMS 控制台将高可用性 (HA) WordPress 站点部署到 AMS 环境中。

> **Note**
>
> 本部署演练已在 AMZN Linux 和 RHEL 环境中进行了测试。

任务和所需任务摘要 RFCs：

1. 创建基础架构（HA 双层堆栈）
2. 为 CodeDeploy 应用程序创建 S3 存储桶
3. 创建 WordPress 应用程序包并将其上传到 S3 存储桶
4. 使用部署应用程序 CodeDeploy
5. 访问该 WordPress 网站并登录以验证部署
6. 拆除部署

所有 CT 选项（包括 ChangeTypeId）的描述均可在 AMS 更改类型参考中找到。

开始前的准备工作

部署 | 高级堆栈组件 | 高可用性双层堆栈 | 创建 CT 可创建 Auto Scaling 组、负载均衡器、数据库以及 CodeDeploy 应用程序名称和部署组（与您为应用程序指定的名称相同）。有关信息， CodeDeploy 请参阅什么是 CodeDeploy？

本演练使用高可用性双层堆栈 RFC，其中包括`UserData`并描述了如何创建可部署的 WordPress CodeDeploy 捆绑包。

示例中`UserData`显示的通过查询 http://169.254.169.254/latest/meta-data/ 上提供的实例元数据服务，从正在运行的实例中获取 EC2 实例元数据，例如实例 ID、区域等。用户数据脚本中的这一行：`REGION=$(curl 169.254.169.254/latest/meta-data/placement/availability-zone/ | sed 's/[a-z]$//')`，将可用区名称从元数据服务检索到我们支持区域的 $REGION 变量中，并使用它来填写下载代理的 S3 存储桶的 URL。 CodeDeploy 169.254.169.254 IP 只能在 VPC 内路由（所有人都可以查询该服务）。 VPCs 有关该服务的信息，请参阅实例元数据和用户数据。另请注意，以身份输入 UserData 的脚本以 “root” 用户身份执行，不需要使用 “sudo” 命令。

本演练将以下参数保留为默认值（如图所示）：

- Auto Scaling 群组：`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`。
- 数据库：`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`。
- 应用程序：`DeploymentConfigName=CodeDeployDefault.OneAtATime`。

变量参数：

控制台为开始时间提供了 “尽快” 选项，本演练建议使用该选项。ASAP 会在批准通过后立即执行 RFC。

> **Note**
>
> 您可以选择设置许多与所示不同的参数。示例中显示的这些参数的值已经过测试，但可能不适合您。示例中仅显示必填值。应更改*replaceable*字体值，因为它们是您的帐户所特有的。

创建基础架构

此过程使用高可用性双层堆栈 CT，然后使用 Create S3 存储 CT。

在开始之前收集以下数据可以加快部署速度。

必需的数据有堆栈：

- AutoScalingGroup:
  - UserData：本教程中提供了此值。它包括为其设置资源 CodeDeploy 和启动 CodeDeploy 代理的命令。
  - AMI-ID：此值决定了您的 Auto Scaling 组 (ASG) 将启动的 EC2 实例的操作系统。在您的账户中选择一个以 “customer-” 开头且具有所需操作系统的 AMI。 IDs 在 AMS 控制台 VPCs -> VPCs 详情页面中查找 AMI。本演练适用于 ASGs 配置为使用亚马逊 Linux 或 RHEL AMI。
- 数据库：
  - 尽管示例中显示的值已经过测试，但这些参数EngineVersion、、和LicenseModel应根据您的情况进行设置。DBEngine本教程分别使用以下值：*MySQL*、*8.0.16*、*general-public-license*。
  - 部署应用程序包时需要MasterUsername这些参数DBNameMasterUserPassword、、和。本教程分别使用以下值：*wordpressDB*、*p4ssw0rd*、*admin*。请注意， DBName 只能包含字母数字字符。
  - 当您输入 RDS 数据库时，它将以明文形式显示，因此请尽快登录数据库并更改密码以确保您的安全。MasterUsername
  - 对于 RDSSubnetID，请使用两个私有子网。每次输入一个，然后按 “Enter”。 IDs 使用 AMS SKMS API 参考查找子网，请参阅 AWS Artifact 控制台中的报告选项卡。操作 (CLI: list-subnet-summaries) 或 AMS 控制台-> VPCs VPC 详细信息页面。

- LoadBalancer:
  - 将此参数 Public 设置为 true，因为本教程使用公有 ELB 子网。
  - ELBSubnetID：使用两个公有子网。每次输入一个，然后按 “Enter”。 IDs 使用 AMS SKMS API 参考查找子网，请参阅 AWS Artifact 控制台中的报告选项卡。操作 (CLI: list-subnet-summaries) 或 AMS 控制台-> VPCs VPC 详细信息页面。
- 应用程序：该ApplicationName值设置 CodeDeploy 应用程序名称和 CodeDeploy 部署组名称。你可以用它来部署你的应用程序。它在账户中必须是唯一的。要查看您的账户中的 CodeDeploy 姓名，请访问 CodeDeploy 控制台。该示例使用了，*WordPress*但是，如果您要使用该值，请确保该值尚未被使用。

1. 启动高可用性堆栈。

   a. 在 “创建 RFC” 页面上，从列表中选择类别 “部署”、“标准堆栈” 子类别、“高可用性双层堆栈” 和 “创建” 操作。

   b. 重要：选择 “高级”，然后如图所示设置值。

      您只需要为已加星标 (*) 的选项输入值，示例中显示了测试值；您可以将非必填的空选项留空。

   c. 对于 RFC 描述部分：

      ```
      Subject: WP-HA-2-Tier-RFC
      ```

   d. 在 “资源信息” 部分，为 “数据库” AutoScalingGroup、“应用程序” 和 “标签” 设置参数。LoadBalancer

      此外，“AppName” 标签密钥的目的是让您可以轻松地在 EC2 控制台中搜索 ASG 实例；您可以将此标签密钥称为 “名称” 或任何其他您想要的密钥名称。请注意，您最多可以添加 50 个标签。

      ```
      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. 完成后单击 “提交”。

2. 登录到您创建的数据库并更改密码。
3. 启动 S3 存储桶堆栈。

   在开始之前收集以下数据可以加快部署速度。

   必需的数据 S3 存储桶：

   - VPC-ID：此值决定您的 S3 存储桶将位于何处。 IDs 使用有关 AMS SKMS API 参考查找 VPC，请参阅 AWS Artifact 控制台中的报告选项卡。操作 (CLI: list-vpc-summaries) 或 AMS 控制台 VPCs 页面中。
   - BucketName：此值设置 S3 存储桶名称，您可以使用它来上传您的应用程序包。它在账户所在区域内必须是唯一的，并且不能包含大写字母。不要求将您的账户 ID 作为其中的 BucketName

一部分，但可以更轻松地在以后识别存储桶。要查看账户中存在哪些 S3 存储桶名称，请访问您的账户的 Amazon S3 控制台。

a. 在 “创建 RFC” 页面上，从 RFC CT 选择列表中选择部署类别、子类别 “高级堆栈组件”、“项目 S3 存储” 和 “创建” 操作。

b. 保留默认的 “基本” 选项，并如图所示设置值。

```
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. 完成后单击 “提交”。使用此更改类型部署的存储桶允许对整个账户进行完全 read/write 访问权限。

创建、上传和部署应用程序

首先，创建 WordPress 应用程序包，然后 CodeDeploy CTs 使用创建和部署应用程序。

1. 下载 WordPress、解压缩文件并创建. /scripts 目录。

   Linux 命令：

   ```
   wget https://github.com/WordPress/WordPress/archive/master.zip
   ```

   Windows：粘贴`https://github.com/WordPress/WordPress/archive/master.zip`到浏览器窗口并下载 zip 文件。

   创建用于组装软件包的临时目录。

   Linux：

   ```
   mkdir /tmp/WordPress
   ```

   Windows：创建一个 “WordPress” 目录，稍后将使用该目录路径。

2. 将 WordPress 源代码解压缩到 “WordPress” 目录并创建一个。 /scripts 目录。

   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：转到你创建的 “WordPress” 目录并在那里创建一个 “脚本” 目录。

   如果您在 Windows 环境中，请务必将脚本文件的中断类型设置为 Unix (LF)。在 Notepad ++ 中，这是窗口右下角的一个选项。

3. 在 WordPress 目录中创建 CodeDeploy appsec.yml 文件（如果复制示例，请检查缩进，每个空格都很重要）。重要：确保将 WordPress 文件（在本例中为 WordPress 目录中）复制到预期目标 (/var/www/html/WordPress) 的 “源” 路径是正确的。在示例中，appsec.yml 文件位于文件所在的目录中，因此只需要使用 “/”。 WordPress 另外，即使你在 Auto Scaling 组中使用了 RHEL AMI，也要保留 “操作系统：linux” 一行不变。appspec.yml 文件示例：

   ```
   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. 在中创建 bash 文件脚本。 WordPress /scripts 目录。

   首先，config_wordpress.sh使用以下内容创建（如果您愿意，可以直接编辑 wp-config.php 文件）。

> **Note**
>
> *DBName*替换为 HA 堆栈 RFC 中给出的值（例如，wordpress）。
> *DB_MasterUsername*替换为 HA 堆栈 RFC 中给出的MasterUsername值（例如，admin）。
> *DB_MasterUserPassword*替换为 HA 堆栈 RFC 中给出的MasterUserPassword值（例如，p4ssw0rd）。
> 在 HA 堆栈 RFC 的执行输出中替换*DB_ENDPOINT*为终端节点 DNS 名称（例如srt1cz23n45sfg.clgvd67uvydk.us-east-1.rds.amazonaws.com）。你可以通过GetRfc操作（CLI：get-rfc--rfc-id RFC_ID）或者在之前提交的 HA Stack RFC 的 AMS 控制台 RFC 详情页面中找到它。

```
#!/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. 在同一个目录中创建install_dependencies.sh包含以下内容的内容：

```
#!/bin/bash
yum install -y php
yum install -y php-mysql
yum install -y mysql
service httpd restart
```

> **Note**
>
> HTTPS 是在启动时作为用户数据的一部分安装的，以便运行状况检查从一开始就起作用。

6. 在同一个目录中创建`start_server.sh`包含以下内容的内容：

   - 对于亚马逊 Linux 实例，请使用以下命令：

   ```
   #!/bin/bash
   service httpd start
   ```

   - 对于 RHEL 实例，请使用以下命令（额外的命令是允许 SELINUX 接受的策略）： WordPress

   ```
   #!/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. 在同一个目录中创建`stop_server.sh`包含以下内容的内容：

   ```
   #!/bin/bash
   service httpd stop
   ```

8. 创建 zip 捆绑包。

   Linux：

   ```
   $ cd /tmp/WordPress
   $ zip -r wordpress.zip .
   ```

   Windows：前往 “WordPress” 目录选择所有文件并创建一个 zip 文件，一定要将其命名为 wordpress.zip。

1. 将应用程序包上传到 S3 存储桶

   要继续部署堆栈，软件包需要准备就绪。

您可以自动访问自己创建的任何 S3 存储桶实例。您可以通过 Bastions（请参阅访问实例）或 S3 控制台对其进行访问，然后使用 drag-and-drop文件或浏览并选择文件来上传 CodeDeploy 软件包。

您也可以在 shell 窗口中使用以下命令；请确保您的 zip 文件路径正确：

```
aws s3 cp wordpress/wordpress.zip s3://BUCKET_NAME/
```

2. 部署 WordPress CodeDeploy 应用程序包

必需的数据代码部署应用程序部署：

- CodeDeployApplicationName: 你给 CodeDeploy 应用程序起的名字。
- CodeDeployGroupName：由于 CodeDeploy 应用程序和组都是根据您在 HA 堆栈 RFC 中为 CodeDeploy 应用程序指定的名称创建的，因此该名称与。CodeDeployApplicationName
- S3Bucket：你给 S3 存储桶起的名字。
- S3 BundleType 和 S3Key：它们是您部署的 WordPress 应用程序包的一部分。
- VpcId：相关的 VPC。

a. 在 “创建 RFC” 页面上，从 RFC CT 选择列表中选择 “部署”、“CodeDeploy 应用程序” 子类别、“应用程序” 和 “操作” 部署类别。

b. 保留默认的 “基本” 选项，并如图所示设置值。

> **Note**
>
> 引用之前创建的 CodeDeploy 应用程序、 CodeDeploy 部署组、S3 存储桶和捆绑包。

```
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. 完成后单击 “提交”。

验证应用程序部署

导航到先前创建的负载均衡器的终端节点 (LoadBalancerCName)， WordPress 部署的路径为：/。WordPress例如：

```
http://stack-ID-FOR-ELB.us-east-1.elb.amazonaws.com/WordPress
```

你应该会看到这样的页面：

拆除高可用性部署

要取消部署，您可以提交针对 HA 双层堆栈和 S3 存储桶的 Delete Stack CT，然后可以请求删除 RDS 快照（它们将在十天后自动删除，但在此期间确实会花费少量费用）。收集 HA 堆栈 IDs 和 S3 存储桶的堆栈，然后按照以下步骤操作。参见堆栈 | 删除。

控制台教程：部署 Tier and Tie WordPress 网站

本节介绍如何使用 AMS 控制台将高可用性 (HA) WordPress 站点部署到 AMS 环境中。这组说明包括创建必要的 WordPress CodeDeploy兼容软件包（例如 zip）文件的示例。资源的配置遵循的顺序允许您将它们绑定在一起形成 “等级”。

> **Note**
>
> 本部署演练专为与 AMZN Linux 操作系统配合使用而设计。
> 基本变量参数表示为*replaceable*；但是，您可能需要修改其他参数以适应您的情况。

任务和所需任务摘要 RFCs：

1. 创建基础架构：
   a. 创建 MySQL RDS 数据库集群
   b. 创建负载均衡器
   c. 创建 Auto Scaling 组并将其绑定到负载均衡器
   d. 为 CodeDeploy 应用程序创建 S3 存储桶
2. 创建 WordPress 应用程序包（不需要 RFC）
3. 使用以下命令部署 WordPress 应用程序包 CodeDeploy：
   a. 创建 CodeDeploy 应用程序
   b. 创建 CodeDeploy 部署组
   c. 将您的 WordPress 应用程序包上传到 S3 存储桶（不需要 RFC）
   d. 部署 CodeDeploy 应用程序
4. 验证部署
5. 拆除部署

所有 CT 选项（包括）的描述均 ChangeTypeId 可在 AMS 更改类型参考中找到。

使用控制台创建 RFC（基础知识）

这些是每次使用控制台创建 RFC 时都必须遵循的一些步骤。

1. RFCs在左侧导航窗格中选择打开 RFCs 列表页面，然后选择创建 RFC。

   将打开 “创建 RFC” 页面。

2. 选择 “浏览更改类型”（默认）或 “按类别选择”。
3. 浏览变更类型：
   a. 选择快速创建选项，使用最常用的更改类型之一开始 RFC。

      将打开该更改类型的常规配置区域，填写主题行。要查看更改类型的详细信息，请打开页面顶部的区域。

   b. 使用所有更改类型区域。

      筛选、在卡片或表格视图之间切换，或者对更改类型进行排序。找到所需的 RFC 后，将其选中，然后选择页面顶部的 “创建 RFC”。

将打开该更改类型的常规配置区域，填写主题行。要查看更改类型的详细信息，请打开页面顶部的区域。

4. 按类别选择：

   a. 选择相应的 “类别”、“子类别”、“物料” 和 “操作”。

      更改类型详细信息框出现在页面底部。

   b. 选择页面底部的创建 RFC。

   c. 将打开该更改类型的常规配置区域，填写主题行。要查看更改类型的详细信息，请打开页面顶部的区域。

5. 为确保某些人收到 RFC 进度的通知，请填写电子邮件地址。要添加有关变更类型的详细信息，请填写描述。打开 “其他配置” 区域以添加有关 RFC 的更多详细信息。

6. 在 “计划” 中，选择 “尽快执行此更改” 或 “安排此更改”。如果您选择 “尽快执行此更改”，则您的 RFC 将在批准通过后立即执行。如果您选择 “安排此更改类型”，则会显示选择日历、时间和时区，并且您的 RFC 将在提交后按计划启动。

7. 在 “执行配置” 区域中，配置更改类型参数。要查看可选参数，请打开其他配置区域。

8. 准备就绪后，选择 “运行”。

创建基础架构

登录目标 AMS 账户的 AWS 控制台，然后登录该账户的 AMS 控制台。

以下过程介绍如何创建 RDS 数据库、负载均衡器和 Auto Scaling 组，以便您使用资源 IDs 来构建基础架构。

创建 RDS 堆栈

请参阅 RDS 堆栈 | 创建。

创建 ELB 堆栈

启动公共弹性负载均衡。

必填数据：

- `VpcId`：您正在使用的 VPC，应与之前使用的 VPC 相同。

- ELBSubnetIds：负载均衡器将在其中分配流量的子网数组。选择公有子网或私有子网。 IDs 使用 AMS SKMS API 参考查找子网，请参阅 AWS Artifact 控制台中的报告选项卡。操作 (CLI: list-subnet-summaries) 或 AMS 控制台-> VPCs VPC 详细信息页面。
- VpcId：您正在使用的 VPC，应与之前使用的 VPC 相同。

1. 在创建 RFC 页面上，选择类别 “部署”、“高级堆栈组件” 子类别、“负载均衡器 (ELB) 堆栈” 项，然后单击 “创建”。选择 “高级”，接受除下方显示的默认值之外的所有默认值（包括没有值的默认值）。

```
Subject:                          WP-ELB-RFC
ELBSubnetIds:                     PUBLIC_AZ1
                                PUBLIC_AZ2
ELBScheme                         true
ELBCookieExpirationPeriod         600
VpcId:                            VPC_ID
Name:                             WP-Public-ELB
```

2. 完成后单击 “提交”。

创建 Auto Scaling 组堆栈

启动 Auto Scaling 组。

必填数据：

- VpcId：您正在使用的 VPC，应与之前使用的 VPC 相同。
- AMI-ID：此值决定了您的 Auto Scaling 组 (ASG) 将启动哪种 EC2 实例。请务必在您的账户中选择以 “customer-” 开头且具有所需操作系统的 AMI。 IDs 使用 AMS SKMS API 参考查找 AMI，请参阅 AWS Artifact 控制台的 “报告” 选项卡。操作 (CLI: list-amis) 或 AMS 控制台-> 详情页面。 VPCs VPCs 本演练适用于 ASGs 配置为使用 Linux AMI。
- ASGLoadBalancerNames: 您之前创建的负载均衡器——通过查看 EC2 控制台->负载均衡器（在左侧导航栏中）找到名称。请注意，这不是您之前创建 ELB 时指定的 “名称”。

1. 在创建 RFC 页面上，选择类别 “部署”、“高级堆栈组件” 子类别、“自动伸缩组” 项目，然后单击 “创建”。选择 “高级”，接受除下方显示的默认值之外的所有默认值（包括没有值的默认值）。

> **Note**
>
> 指定最新的 AMS AMI。指定先前创建的 ELB。

```
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. 完成后单击 “提交”。

创建 S3 堆栈

启动 S3 存储桶。S3 存储桶是您上传创建的应用程序包的地方。

所需数据：

- VPC-ID：此值决定您的 S3 存储桶将位于何处，该值应与之前使用的 VPC 相同。
- AccessControl: 预设 AccessControl 列表 (ACL) 选项为Private、和。PublicRead有关更多信息，请参阅 Amazon 简单存储服务预装 ACL。

- BucketName：此值设置 S3 存储桶名称，您可以使用它来上传您的应用程序包。它在账户所在区域必须是唯一的，并且不能包含大写字母。不要求将您的账户 ID 作为其中的 BucketName 一部分，但可以更轻松地在以后识别存储桶。要查看账户中存在哪些 S3 存储桶名称，请访问您的账户的 Amazon S3 控制台。

1. 在“创建 RFC”页面上，选择部署类别、子类别“高级堆栈组件”、“S3 存储”项目，然后单击“创建”。

   如上所述，您可以将默认参数选项保留在“基本”以接受默认值。要设置不同的值，请选择“高级”。

   > **Note**
   >
   > 使用此更改类型部署的存储桶允许对整个账户进行完全 read/write 访问权限，可能需要新的更改类型才能允许更受限制的访问权限。

   ```
   Subject:            S3-Bucket-RFC
   BucketName:         ACCOUNT_ID-codedeploy-bundles
   AccessControl:      Private

   VpcId:              VPC_ID
   Name:               S3BucketForWP
   ```

2. 完成后单击“提交”。

创建 WordPress CodeDeploy 捆绑包

本节提供了创建应用程序部署包的示例。

1. 下载 WordPress、解压缩文件并创建. /scripts 目录。

   Linux 命令：

   ```
   wget https://github.com/WordPress/WordPress/archive/master.zip
   ```

   Windows：粘贴https://github.com/WordPress/WordPress/archive/master.zip到浏览器窗口并下载 zip 文件。

创建用于组装软件包的临时目录。

Linux：

```
mkdir /tmp/WordPress
```

Windows：创建一个 “WordPress” 目录，稍后将使用该目录路径。

2. 将 WordPress 源代码解压缩到 “WordPress” 目录并创建一个。 /scripts 目录。

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：转到你创建的 “WordPress” 目录并在那里创建一个 “脚本” 目录。

如果您在 Windows 环境中，请务必将脚本文件的中断类型设置为 Unix (LF)。在 Notepad ++ 中，这是窗口右下角的一个选项。

3. 在 WordPress 目录中创建 CodeDeploy appsec.yml 文件（如果复制示例，请检查缩进，每个空格都很重要）。重要：确保将 WordPress 文件（在本例中为 WordPress 目录中）复制到预期目标 (/var/www/html/WordPress) 的 “源” 路径是正确的。在示例中，appsec.yml 文件位于文件所在的目录中，因此只需要使用 “/”。 WordPress 另外，即使你在 Auto Scaling 组中使用了 RHEL AMI，也要保留 “操作系统：linux” 一行不变。appspec.yml 文件示例：

```
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. 在中创建 bash 文件脚本。 WordPress /scripts 目录。

   首先，config_wordpress.sh使用以下内容创建（如果您愿意，可以直接编辑 wp-config.php 文件）。

   > **Note**
   >
   > *DBName*替换为 HA 堆栈 RFC 中给出的值（例如，wordpress）。
   > *DB_MasterUsername*替换为 HA 堆栈 RFC 中给出的MasterUsername值（例如，admin）。
   > *DB_MasterUserPassword*替换为 HA 堆栈 RFC 中给出的MasterUserPassword值（例如，p4ssw0rd）。
   > 在 HA 堆栈 RFC 的执行输出中替换*DB_ENDPOINT*为终端节点 DNS 名称（例如srt1cz23n45sfg.clgvd67uvydk.us-east-1.rds.amazonaws.com）。你可以通过GetRfc操作（CLI：get-rfc--rfc-id RFC_ID）或者在你之前提交的 HA Stack RFC 的 AMS 控制台 RFC 详情页面中找到它。

```
#!/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. 在同一个目录中创建install_dependencies.sh包含以下内容的内容：

```
#!/bin/bash
yum install -y php
yum install -y php-mysql
yum install -y mysql
service httpd restart
```

> **Note**
>
> HTTPS 是在启动时作为用户数据的一部分安装的，以便运行状况检查从一开始就起作用。

6. 在同一个目录中创建`start_server.sh`包含以下内容的内容：

   - 对于亚马逊 Linux 实例，请使用以下命令：

   ```
   #!/bin/bash
   service httpd start
   ```

   - 对于 RHEL 实例，请使用以下命令（额外的命令是允许 SELINUX 接受的策略）： WordPress

   ```
   #!/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. 在同一个目录中创建`stop_server.sh`包含以下内容的内容：

   ```
   #!/bin/bash
   service httpd stop
   ```

8. 创建 zip 捆绑包。

   Linux：

   ```
   $ cd /tmp/WordPress
   $ zip -r wordpress.zip .
   ```

Windows：前往 “WordPress” 目录选择所有文件并创建一个 zip 文件，一定要将其命名为 wordpress.zip。

使用部署 WordPress 应用程序包 CodeDeploy

CodeDeploy 是一项 AWS 部署服务，可自动将应用程序部署到 Amazon EC2 实例。该过程的这一部分包括创建 CodeDeploy 应用程序、创建 CodeDeploy 部署组，然后使用部署应用程序 CodeDeploy。

创建 CodeDeploy 应用程序

CodeDeploy 应用程序只是 AWS CodeDeploy 使用的名称或容器，用于确保在部署期间引用正确的修订版、部署配置和部署组。在本例中，部署配置就是您之前创建的 WordPress 捆绑包。

必填数据：

- `VpcId`：您正在使用的 VPC，应与之前使用的 VPC 相同。
- `CodeDeployApplicationName`: 账户中必须是唯一的。查看 CodeDeploy 控制台以检查现有的应用程序名称。

1. 为以下对象创建 CodeDeploy 应用程序 WordPress

   在 “创建 RFC” 页面上，从 RFC CT 选择列表中选择 “部署”、“应用程序” 子类别、“CodeDeploy 应用程序” 和 “操作” “创建”。选择 “基本”，然后如图所示设置值。完成后单击 “提交”。

   ```
   Subject:                    CD-WP-App-RFC
   CodeDeployApplicationName:  WordPress
   VpcId:                      VPC_ID
   Name:                       WP-CD-App
   ```

2. 完成后单击 “提交”。

创建 CodeDeploy 部署组

创建 CodeDeploy 部署组。

CodeDeploy 部署组定义了一组针对部署的单个实例。

所需数据：

- VpcId：您正在使用的 VPC，应与之前使用的 VPC 相同。
- CodeDeployApplicationName：使用您之前创建的值。
- CodeDeployAutoScalingGroups：使用您之前创建的 Auto Scaling 组的名称。
- CodeDeployDeploymentGroupName：部署组的名称。此名称对于与部署组关联的每个应用程序来说必须是唯一的。
- CodeDeployServiceRoleArn：使用示例中给出的公式。

1. 在 “创建 RFC” 页面上，从 RFC CT 选择列表中选择 “类别”、“部署”、“应用程序” 子类别、“项目CodeDeploy 部署组” 和 “创建” 操作。选择 “高级” 并按所示设置值（RFC 只需要主题即可）。完成后单击 “提交”。

> **Note**
>
> 以这种格式引用 CodeDeploy 服务角色 ARN，`"arn:aws:iam::085398962942:role/aws-codedeploy-role"`并使用之前创建的 Auto Scaling 组名称作为 “ASG_NAME”。

```
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. 完成后单击 “提交”。

上传 WordPress 应用程序

您可以自动访问自己创建的任何 S3 存储桶实例。您可以通过 Bastions（请参阅访问实例）或 S3 控制台访问它，然后上传 CodeDeploy 捆绑包。要继续部署堆栈，捆绑包需要准备就绪。该示例使用先前创建的存储桶名称。

您可以使用以下 AWS 命令来压缩捆绑包：

```
aws s3 cp wordpress/wordpress.zip s3://ACCOUNT_ID-codedeploy-bundles/
```

使用部署 WordPress 应用程序 CodeDeploy

部署 CodeDeploy 应用程序。

所需数据：

- VPC-ID：您正在使用的 VPC，应与之前使用的 VPC 相同。
- CodeDeployApplicationName：使用您之前创建的 CodeDeploy 应用程序的名称。
- CodeDeployDeploymentGroupName：使用您之前创建的 CodeDeploy 部署组的名称。
- S3Location（您上传应用程序包的位置）:S3Bucket: 您之前创建的，S3BundleType以及S3Key：您放在 S3 商店中的捆绑包的类型和名称。 BucketName

1. 部署 WordPress CodeDeploy 应用程序包

   在创建 RFC 页面上，从 RFC CT 选择列表中选择部署类别、子类别CodeDeploy 应用程序、项目应用程序和操作部署。选择 “基本”，然后如图所示设置值。完成后单击 “提交”。

   > **Note**
   >
   > 引用之前创建的 CodeDeploy 应用程序、 CodeDeploy 部署组、S3 存储桶和捆绑包。

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. 完成后单击 “提交”。

验证应用程序部署

导航到先前创建的负载均衡器的终端节点 (ELB CName)，WordPress 部署的路径为：/。WordPress 例如：

```
http://stack-ID-FOR-ELB.us-east-1.elb.amazonaws.com/WordPress
```

拆除应用程序部署

要取消部署，您需要提交针对 RDS 数据库堆栈、应用程序负载均衡器、Auto Scaling 组、S3 存储桶以及 Code Deploy 应用程序和组（总共六个）的删除堆栈 C RFCs T。此外，您可以提交服务请求以删除 RDS 快照（它们将在十天后自动删除，但在此期间确实会花费少量费用）。收集所有 IDs 人的堆栈，然后按照以下步骤操作。参见堆栈 | 删除。

CLI 教程：高可用性双层堆栈 (Linux/RHEL)

本节介绍如何使用 AMS CLI 将高可用性 (HA) 双层堆栈部署到 AMS 环境中。

> **Note**
>
> 本部署演练已在 AMZN Linux 和 RHEL 环境中进行了测试。

任务和所需任务摘要 RFCs：

1. 创建基础架构（HA 双层堆栈）
2. 为 CodeDeploy 应用程序创建 S3 存储桶
3. 创建 WordPress 应用程序包并将其上传到 S3 存储桶
4. 使用部署应用程序 CodeDeploy
5. 访问该 WordPress 网站并登录以验证部署

开始前的准备工作

部署 | 高级堆栈组件 | 高可用性双层堆栈高级 | 创建 CT 可创建 Auto Scaling 组、负载均衡器、数据库以及 CodeDeploy 应用程序名称和部署组（与您为应用程序指定的名称相同）。有关信息，CodeDeploy 请参阅什么是 CodeDeploy？

本演练使用高可用性双层堆栈（高级）RFC，其中包括 UserData 并描述了如何创建可部署的 WordPress CodeDeploy 捆绑包。

示例中UserData显示的通过查询 http://169.254.169.254/latest/meta-data/ 上提供的实例元数据服务，从正在运行的实例中获取 EC2 实例元数据，例如实例 ID、区域等。用户数据脚本中的这一行：REGION=$(curl 169.254.169.254/latest/meta-data/placement/availability-zone/ | sed 's/[a-z]$//')，将可用区名称从元数据服务检索到我们支持区域的 $REGION 变量中，并使用它来填写下载代理的 S3 存储桶的 URL。 CodeDeploy 169.254.169.254 IP 只能在 VPC 内路由（所有人都可以查询该服务）。 VPCs 有关该服务的信息，请参阅实例元数据和用户数据。另请注意，以身份输入 UserData 的脚本以 “root” 用户身份执行，不需要使用 “sudo” 命令。

本演练将以下参数保留为默认值（如图所示）：

- Auto Scaling 群组：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。
- 数据库：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。
- 应用程序：DeploymentConfigName=CodeDeployDefault.OneAtATime。
- S3 存储桶：AccessControl=Private。

其他设置：

RequestedStartTimeRequestedEndTime如果你想安排 RFC：你可以使用 Time.is 来确定正确的 UT C 时间。必须对所提供的示例进行适当调整。如果开始时间已过，RFC 将无法继续。或者，您可以省略这些值以创建一个 ASAP RFC，该RFC将在批准通过后立即执行。

> **Note**
>
> 您可以选择设置许多与所示不同的参数。示例中显示的这些参数的值已经过测试，但可能不适合您。

创建基础架构

在开始之前收集以下数据可以加快部署速度。

必需的数据有堆栈：

- AutoScalingGroup:
  - `UserData`：本教程中提供了此值。它包括为其设置资源 CodeDeploy 和启动 CodeDeploy 代理的命令。
  - `AMI-ID`：此值决定了您的 Auto Scaling 组 (ASG) 将启动哪种 EC2 实例。请务必在您的账户中选择以 “customer-” 开头且具有所需操作系统的 AMI。 IDs 使用 AMS SKMS API 参考查找 AMI，请参阅 AWS Artifact 控制台的 “报告” 选项卡。操作 (CLI: list-amis) 或 AMS 控制台-> 详情页面。VPCs VPCs 本演练适用于 ASGs 配置为使用 Linux AMI。
- 数据库：
  - 尽管示例中显示的值已经过测试，但这些参数`EngineVersion`、、和`LicenseModel`应根据您的情况进行设置。`DBEngine`
  - 部署应用程序包时需要`MasterUserPassword`这些参数`RDSSubnetIds``DBName``MasterUsername`、、、和。对于 RDSSubnet ID，请使用两个私有子网。
- LoadBalancer:
  - 尽管示例中显示的值已经过测试，但这些参数`EngineVersion`、、和`LicenseModel`应根据您的情况进行设置。`DBEngine`
  - `ELBSubnetIds`：使用两个公有子网。
- 应用程序：该`ApplicationName`值设置 CodeDeploy 应用程序名称和 CodeDeploy 部署组名称。你可以用它来部署你的应用程序。它在账户中必须是唯一的。要查看您的账户中的 CodeDeploy 姓名，请访问 CodeDeploy 控制台。该示例使用 “WordPress”，但是，如果您要使用该值，请确保该值尚未被使用。

此过程使用高可用性双层堆栈（高级）CT（ct-06mjngx5flwto）和创建 S3 存储 CT（ct-1a68ck03fn98r）。在经过身份验证的账户中，在命令行中执行以下步骤。

1. 启动基础架构堆栈。

   a. 将 HA 双层堆栈 CT 的执行参数 JSON 架构输出到当前文件夹中名为 CreateStackParams .json 的文件中。

```
aws amscm get-change-type-version --change-type-id "ct-06mjngx5flwto"
 --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
 CreateStackParams.json
```

   b. 修改架构。根据需要*variables*更换。例如，对于 ASG 将要创建的 EC2 实例，使用所需的操作系统。记录下来ApplicationName，因为您稍后将使用它来部署应用程序。请注意，您最多可以添加 50 个标签。

```
{
"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. 将 CreateRfc JSON 模板输出到当前文件夹中名为 CreateStackRfc .json 的文件中：

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

d. 按如下方式修改 RFC 模板并保存，即可删除和替换内容。请注意，RequestedStartTime和现在RequestedEndTime是可选的；排除它们会创建一个 ASAP RFC，该RFC在获得批准后立即执行（通常会自动发生）。要提交计划的 RFC，请添加这些值。

```
{
"ChangeTypeVersion":    "3.0",
"ChangeTypeId":         "ct-06mjngx5flwto",
"Title":                "HA-Stack-For-WP-RFC"
}
```

e. 创建 RFC，指定 CreateStackRfc .json 文件和.js CreateStackParams on 执行参数文件：

```
aws amscm create-rfc --cli-input-json file://CreateStackRfc.json --execution-
parameters file://CreateStackParams.json
```

您在响应中收到 RFC ID。保存 ID 以供后续步骤使用。

f. 提交 RFC：

```
aws amscm submit-rfc --rfc-id  RFC_ID
```

如果 RFC 成功，则不会收到任何输出。

g. 要检查 RFC 状态，请运行

```
aws amscm get-rfc --rfc-id RFC_ID
```

记下 RFC ID。

2. 启动 S3 存储桶

在开始之前收集以下数据可以加快部署速度。

必需的数据 S3 存储桶：

- VPC-ID：此值决定您的 S3 存储桶将位于何处。使用您之前使用的 VPC ID。
- BucketName：此值设置 S3 存储桶名称，您可以使用它来上传您的应用程序包。它在账户所在区域内必须是唯一的，并且不能包含大写字母。不要求将您的账户 ID 作为其中的 BucketName 一部分，但可以更轻松地在以后识别存储桶。要查看账户中存在哪些 S3 存储桶名称，请访问您的账户的 Amazon S3 控制台。

a. 将 S3 存储创建 CT 的执行参数 JSON 架构输出到名为 createS3 StoreParams .json 的 JSON 文件中。

```
aws amscm get-change-type-version --change-type-id "ct-1a68ck03fn98r"
 --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
 CreateS3StoreParams.json
```

b. 按如下方式修改架构，可以删除和替换其中的内容。*VPC_ID*适当地更换。示例中的值已经过测试，但可能不适合您。

> **Tip**
>
> 在账户所在区域内BucketName必须是唯一的，并且不能包含大写字母。不要求将您的账户 ID 作为其中的 BucketName 一部分，但可以更轻松地在以后识别存储桶。要查看账户中存在哪些 S3 存储桶名称，请访问您的账户的 Amazon S3 控制台。

```
{
"Description":      "S3BucketForWordPressBundle",
"VpcId":            "VPC_ID",
"StackTemplateId":  "stm-s2b72beb000000000",
"Name":             "S3BucketForWP",
"TimeoutInMinutes":  60,
"Parameters":   {
    "AccessControl":    "Private",
    "BucketName":       "ACCOUNT_ID-BUCKET_NAME"
    }
}
```

c. 将的 JSON 模板输出 CreateRfc 到当前文件夹中名为 createS3 StoreRfc .json 的文件中：

```
aws amscm create-rfc --generate-cli-skeleton > CreateS3StoreRfc.json
```

d. 修改并保存 createS3 StoreRfc .json 文件，你可以删除和替换其中的内容。请注意，RequestedStartTime和现在RequestedEndTime是可选的；排除它们会创建一个 ASAP RFC，该RFC在获得批准后立即执行（通常会自动发生）。要提交计划的 RFC，请添加这些值。

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-1a68ck03fn98r",
"Title":                "S3-Stack-For-WP-RFC"
}
```

e. 创建 RFC，指定 createS3 StoreRfc .json 文件和 createS3 .json StoreParams 执行参数文件：

```
aws amscm create-rfc --cli-input-json file://CreateS3StoreRfc.json  --
execution-parameters file://CreateS3StoreParams.json
```

您会在回复 RfcId 中收到新 RFC 的信息。保存 ID 以供后续步骤使用。

f. 提交 RFC：

```
aws amscm submit-rfc --rfc-id RFC_ID
```

如果 RFC 成功，则不会收到任何输出。

g. 要检查 RFC 状态，请运行

```
aws amscm get-rfc --rfc-id RFC_ID
```

创建、上传和部署应用程序

首先，创建 WordPress 应用程序包，然后 CodeDeploy CTs 使用创建和部署应用程序。

1. 下载 WordPress、解压缩文件并创建. /scripts 目录。

   Linux 命令：

   ```
   wget https://github.com/WordPress/WordPress/archive/master.zip
   ```

   Windows：粘贴`https://github.com/WordPress/WordPress/archive/master.zip`到浏览器窗口并下载 zip 文件。

   创建用于组装软件包的临时目录。

   Linux：

   ```
   mkdir /tmp/WordPress
   ```

   Windows：创建一个 “WordPress” 目录，稍后将使用该目录路径。

2. 将 WordPress 源代码解压缩到 “WordPress” 目录并创建一个。 /scripts 目录。

   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：转到你创建的 “WordPress” 目录并在那里创建一个 “脚本” 目录。

如果您在 Windows 环境中，请务必将脚本文件的中断类型设置为 Unix (LF)。在 Notepad ++ 中，这是窗口右下角的一个选项。

3. 在 WordPress 目录中创建 CodeDeploy appsec.yml 文件（如果复制示例，请检查缩进，每个空格都很重要）。重要：确保将 WordPress 文件（在本例中为 WordPress 目录中）复制到预期目标 (/var/www/html/WordPress) 的 “源” 路径是正确的。在示例中，appsec.yml 文件位于文件所在的目录中，因此只需要 “/”。 WordPress 另外，即使你在 Auto Scaling 组中使用了 RHEL AMI，也要保留 “操作系统：linux” 一行不变。appspec.yml 文件示例：

```
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. 在中创建 bash 文件脚本。 WordPress /scripts 目录。

首先，`config_wordpress.sh`使用以下内容创建（如果您愿意，可以直接编辑 wp-config.php 文件）。

> **Note**
>
> *DBName*替换为 HA 堆栈 RFC 中给出的值（例如，`wordpress`）。

*DB_MasterUsername*替换为 HA 堆栈 RFC 中给出的MasterUsername值（例如，admin）。
*DB_MasterUserPassword*替换为 HA 堆栈 RFC 中给出的MasterUserPassword值（例如，p4ssw0rd）。
在 HA 堆栈 RFC 的执行输出中替换*DB_ENDPOINT*为终端节点 DNS 名称（例如srt1cz23n45sfg.clgvd67uvydk.us-east-1.rds.amazonaws.com）。你可以通过GetRfc操作（CLI：get-rfc--rfc-id RFC_ID）或之前提交的 HA Stack RFC 的 AMS 控制台 RFC 详情页面中找到这一点。

```
#!/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. 在同一个目录中创建install_dependencies.sh包含以下内容的内容：

```
#!/bin/bash
yum install -y php
yum install -y php-mysql
yum install -y mysql
service httpd restart
```

> **Note**
>
> HTTPS 是在启动时作为用户数据的一部分安装的，以便运行状况检查从一开始就起作用。

6. 在同一个目录中创建start_server.sh包含以下内容的内容：

   - 对于亚马逊 Linux 实例，请使用以下命令：

```
#!/bin/bash
service httpd start
```

- 对于 RHEL 实例，请使用以下命令（额外的命令是允许 SELINUX 接受的策略）： WordPress

```
#!/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. 在同一个目录中创建`stop_server.sh`包含以下内容的内容：

```
#!/bin/bash
service httpd stop
```

8. 创建 zip 捆绑包。

   Linux：

```
$ cd /tmp/WordPress
$ zip -r wordpress.zip .
```

   Windows：前往 “WordPress” 目录选择所有文件并创建一个 zip 文件，一定要将其命名为 wordpress.zip。

1. 将应用程序包上传到 S3 存储桶。

   要继续部署堆栈，捆绑包需要准备就绪。

   您可以自动访问自己创建的任何 S3 存储桶实例。您可以通过堡垒或 S3 控制台访问它，然后上传 WordPress 捆绑包 drag-and-drop或浏览并选择 zip 文件。

   您也可以在 shell 窗口中使用以下命令；请确保您的 zip 文件路径正确：

```
aws s3 cp wordpress.zip s3://BUCKET_NAME/
```

2. 部署 WordPress 应用程序包。

   在开始之前收集以下数据可以加快部署速度。

   必填数据：

- VPC-ID：此值决定您的 S3 存储桶将位于何处。使用您之前使用的 VPC ID。
- CodeDeployApplicationName和CodeDeployApplicationName：你在 HA 2 层堆栈 RFC 中使用的ApplicationName值设置 CodeDeployApplicationName 和。CodeDeployDeploymentGroupName该示例使用 “WordPress”，但您可能使用了不同的值。
- S3Location: 对于 S3BucketBucketName，请使用您之前创建的。S3BundleType和来自S3Key您放在 S3 商店中的捆绑包。

a. 将 CodeDeploy 应用程序部署 CT 的执行参数 JSON 架构输出到名为 Deploy p CDApp arams.json 的 JSON 文件中。

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

b. 按如下方式修改架构并将其另存为，您可以删除和替换内容。

```
{
"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. 将的 JSON 模板输出 CreateRfc 到当前文件夹中名为 Deploy CDApp rfc.json 的文件中：

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

d. 修改并保存 Deplo CDApp y rfc.json 文件，你可以删除和替换其中的内容。请注意，RequestedStartTime和现在RequestedEndTime是可选的；排除它们会创建一个 ASAP RFC，该RFC在获得批准后立即执行（通常会自动发生）。要提交计划的 RFC，请添加这些值。

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-2edc3sd1sqmrb",
"Title":                "CD-Deploy-For-WP-RFC"
}
```

e. 创建 RFC，指定 Deploy CDApp Rfc 文件和 De CDApp ploy Params 执行参数文件：

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

您会在回复 RfcId 中收到新 RFC 的信息。保存 ID 以供后续步骤使用。

f. 提交 RFC：

```
aws amscm submit-rfc --rfc-id RFC_ID
```

如果 RFC 成功，则不会收到任何输出。

g. 要检查 RFC 状态，请运行

```
aws amscm get-rfc --rfc-id RFC_ID
```

验证应用程序部署

导航到先前创建的负载均衡器的终端节点 (ELB CName)， WordPress 部署的路径为：/。WordPress 例如：

```
http://stack-ID-FOR-ELB.us-east-1.elb.amazonaws.com/WordPress
```

拆除应用程序部署

完成本教程后，您将需要取消部署，这样您就无需为资源付费。

以下是一个通用的堆栈删除操作。您需要提交两次，一次用于 HA 2 层堆栈，一次用于 S3 存储桶堆栈。最后，提交服务请求，要求删除 S3 存储桶的所有快照（在服务请求中包括 S3 存储桶堆栈 ID）。它们会在 10 天后自动删除，但提早删除它们可以节省一点成本。

本演练提供了使用 AMS 控制台删除 S3 堆栈的示例；此过程适用于使用 AMS 控制台删除任何堆栈。

> **Note**
>
> 如果删除 S3 存储桶，则必须先将其中的对象清空。

必填数据：

- `StackId`: 要使用的堆栈。你可以通过查看 AMS Console Stack s 页面来找到它，该页面可通过左侧导航栏中的链接获得。使用 AMS SKMS API/CLI 运行有关 AMS SKMS API 参考，请参阅 AWS Artifact 控制台中的 “报告” 选项卡。操作（在 CLI 中`list-stack-summaries`）。
- 本演练的更改类型 ID 为`ct-0q0bic0ywqk6c`，版本为 “1.0”，要查找最新版本，请运行以下命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=ct-0q0bic0ywqk6c
```

内联创建：

- 使用内联提供的执行参数发出 create RFC 命令（内联提供执行参数时使用转义引号）。E

```
aws amscm create-rfc --change-type-id "ct-0q0bic0ywqk6c" --change-type-version "1.0"
 --title "Delete My Stack" --execution-parameters "{\"StackId\":\"STACK_ID\"}"
```

- 使用创建 RFC 操作中返回的 RFC 编号提交 RFC。在提交之前，RFC 仍处于该`Editing`状态，不会被付诸行动。

```
aws amscm submit-rfc --rfc-id RFC_ID
```

- 监控 RFC 状态并查看执行输出：

```
aws amscm get-rfc --rfc-id RFC_ID
```

模板创建：

1. 将 RFC 模板输出到当前文件夹中的一个文件中；示例将其命名为 DeleteStackRfc .json：

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

2. 修改并保存 DeleteStackRfc .json 文件。由于删除堆栈只有一个执行参数，因此执行参数可以在 DeleteStackRfc .json 文件本身中（无需创建带有执行参数的单独的 JSON 文件）。

   ExecutionParameters JSON 扩展中的内部引号必须使用反斜杠 (\) 进行转义。没有开始和结束时间的示例：

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-0q0bic0ywqk6c",
"Title":                "Delete-My-Stack-RFC"
"ExecutionParameters":  "{
        \"StackId\":\"STACK_ID\"}"
}
```

3. 创建 RFC：

```
aws amscm create-rfc --cli-input-json file://DeleteStackRfc.json
```

   您会在回复 RfcId 中收到新 RFC 的信息。例如：

```
{
"RfcId": "daaa1867-ffc5-1473-192a-842f6b326102"
}
```

   保存 ID 以供后续步骤使用。

4. 提交 RFC：

```
aws amscm submit-rfc --rfc-id RFC_ID
```

   如果 RFC 成功，则不会在命令行收到任何确认。

5. 要监控请求的状态并查看执行输出，请执行以下操作：

```
aws amscm get-rfc --rfc-id RFC_ID --query "Rfc.
{Status:Status.Name,Exec:ExecutionOutput}" --output table
```

CLI 教程：部署 Tier and Tie WordPress 网站

本节介绍如何使用 AMS CLI 将高可用性 (HA) WordPress 站点部署到 AMS 环境中。这组说明包括创建必要的 WordPress CodeDeploy兼容软件包（例如 zip）文件的示例。

> **Note**
>
> 本部署演练专为在 AMZN Linux 环境中使用而设计。
> 基本变量参数标注为*replaceable*；但是，您可能需要修改其他参数以适应您的情况。

任务和所需任务摘要 RFCs：

1. 创建基础架构：
   a. 创建 RDS 堆栈 (CLI)
   b. 创建负载均衡器
   c. 创建 Auto Scaling 组并将其绑定到负载均衡器
   d. 为 CodeDeploy 应用程序创建 S3 存储桶
2. 创建 WordPress 应用程序包（不需要 RFC）
3. 使用以下命令部署 WordPress 应用程序包 CodeDeploy：
   a. 创建 CodeDeploy 应用程序
   b. 创建 CodeDeploy 部署组
   c. 将您的 WordPress 应用程序包上传到 S3 存储桶（不需要 RFC）
   d. 部署 CodeDeploy 应用程序
4. 验证部署
5. 拆除部署

在经过身份验证的账户中，按照命令行中的所有步骤进行操作。

使用 CLI 创建 RFC

有关创建的详细信息 RFCs，请参阅创建 RFCs；有关常见 RFC 参数的说明，请参阅 RFC 常用参数。

创建基础架构

以下过程介绍如何创建 RDS 数据库、负载均衡器和 Auto Scaling 组，IDs 以便您使用资源构建基础架构。

创建 RDS 堆栈 (CLI)

请参阅 RDS 堆栈 | 创建。

创建 ELB 堆栈

启动公共负载均衡器 (ELB)。参见 Load Balancer (ELB) 堆栈 | 创建。

创建 Auto Scaling 组堆栈

启动 Auto Scaling 组。

请参阅 Auto Scaling 群组 | 创建。

创建 S3 存储

启动 S3 存储桶。S3 存储桶是您上传创建的应用程序包的地方。请参阅 S3 存储 | 创建。

为创建 WordPress 应用程序包 CodeDeploy

本节提供了创建应用程序部署包的示例。

1. 下载 WordPress、解压缩文件并创建. /scripts 目录。

   Linux 命令：

   ```
   wget https://github.com/WordPress/WordPress/archive/master.zip
   ```

   Windows：粘贴`https://github.com/WordPress/WordPress/archive/master.zip`到浏览器窗口并下载 zip 文件。

   创建用于组装软件包的临时目录。

   Linux：

   ```
   mkdir /tmp/WordPress
   ```

Windows：创建一个 “WordPress” 目录，稍后将使用该目录路径。

2. 将 WordPress 源代码解压缩到 “WordPress” 目录并创建一个。 /scripts 目录。

   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：转到你创建的 “WordPress” 目录并在那里创建一个 “脚本” 目录。

   如果您在 Windows 环境中，请务必将脚本文件的中断类型设置为 Unix (LF)。在 Notepad ++ 中，这是窗口右下角的一个选项。

3. 在 WordPress 目录中创建 CodeDeploy appsec.yml 文件（如果复制示例，请检查缩进，每个空格都很重要）。重要：确保将 WordPress 文件（在本例中为 WordPress 目录中）复制到预期目标 (/var/www/html/WordPress) 的 “源” 路径是正确的。在示例中，appsec.yml 文件位于文件所在的目录中，因此只需要使用 “/”。 WordPress 另外，即使你在 Auto Scaling 组中使用了 RHEL AMI，也要保留 “操作系统：linux” 一行不变。appspec.yml 文件示例：

   ```
   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. 在中创建 bash 文件脚本。 WordPress /scripts 目录。

   首先，config_wordpress.sh使用以下内容创建（如果您愿意，可以直接编辑 wp-config.php 文件）。

> **Note**
>
> *DBName*替换为 HA 堆栈 RFC 中给出的值（例如，wordpress）。
> *DB_MasterUsername*替换为 HA 堆栈 RFC 中给出的MasterUsername值（例如，admin）。
> *DB_MasterUserPassword*替换为 HA 堆栈 RFC 中给出的MasterUserPassword值（例如，p4ssw0rd）。
> 在 HA 堆栈 RFC 的执行输出中替换*DB_ENDPOINT*为终端节点 DNS 名称（例如srt1cz23n45sfg.clgvd67uvydk.us-east-1.rds.amazonaws.com）。你可以通过GetRfc操作（CLI：get-rfc--rfc-id RFC_ID）或者在你之前提交的 HA Stack RFC 的 AMS 控制台 RFC 详情页面中找到它。

```
#!/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. 在同一个目录中创建install_dependencies.sh包含以下内容的内容：

```
#!/bin/bash
yum install -y php
yum install -y php-mysql
yum install -y mysql
service httpd restart
```

> **Note**
>
> HTTPS 是在启动时作为用户数据的一部分安装的，以便运行状况检查从一开始就起作用。

6. 在同一个目录中创建`start_server.sh`包含以下内容的内容：

   - 对于亚马逊 Linux 实例，请使用以下命令：

   ```
   #!/bin/bash
   service httpd start
   ```

   - 对于 RHEL 实例，请使用以下命令（额外的命令是允许 SELINUX 接受的策略）： WordPress

   ```
   #!/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. 在同一个目录中创建`stop_server.sh`包含以下内容的内容：

   ```
   #!/bin/bash
   service httpd stop
   ```

8. 创建 zip 捆绑包。

   Linux：

   ```
   $ cd /tmp/WordPress
   $ zip -r wordpress.zip .
   ```

   Windows：前往 “WordPress” 目录选择所有文件并创建一个 zip 文件，一定要将其命名为 wordpress.zip。

使用部署 WordPress 应用程序捆绑包 CodeDeploy

CodeDeploy 是一项 AWS 部署服务，可自动将应用程序部署到 Amazon EC2 实例。该过程的这一部分包括创建 CodeDeploy 应用程序、创建 CodeDeploy 部署组，然后使用部署应用程序 CodeDeploy。

创建 CodeDeploy 应用程序

CodeDeploy 应用程序只是 AWS CodeDeploy 使用的名称或容器，用于确保在部署期间引用正确的修订版、部署配置和部署组。在本例中，部署配置就是您之前创建的 WordPress 捆绑包。

所需数据：

- `VpcId`：您正在使用的 VPC，应与之前使用的 VPC 相同。
- `CodeDeployApplicationName`: 账户中必须是唯一的。查看 CodeDeploy 控制台以检查现有的应用程序名称。
- `ChangeTypeIdandChangeTypeVersion`：本演练的更改类型 ID 是`ct-0ah3gwb9seqk2`，要找出最新版本，请运行以下命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=ct-0ah3gwb9seqk2
```

1. 将 CodeDeploy 应用程序 CT 的执行参数 JSON 架构输出到当前文件夹中的一个文件中；示例将其命名为 Create CDApp params.json。

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

2. 按如下方式修改和保存 JSON 文件；您可以删除和替换其中的内容。

```
{
"Description":                          "Create WordPress CodeDeploy App",
"VpcId":                                "VPC_ID",
"StackTemplateId":                      "stm-sft6rv00000000000",
"Name":                                 "WordPressCDApp",
"TimeoutInMinutes":                     60,
"Parameters":   {
    "CodeDeployApplicationName":    "WordPressCDApp"
    }
}
```

3. 将的 JSON 模板输出 CreateRfc 到当前文件夹中的一个文件中；示例将其命名为 Create CDApp rfc.json。

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

4. 按如下方式修改和保存 JSON 文件；您可以删除和替换其中的内容。请注意，RequestedStartTime和现在RequestedEndTime是可选的；排除它们会导致 RFC 在获得批准后立即执行（这通常是自动发生的）。要提交 “计划的” RFC，请添加这些值。

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

5. 创建 RFC，指定创建 CDApp Rfc 文件和执行参数文件：

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

   您将在响应中收到新 RFC 的 RFC ID。保存 ID 以供后续步骤使用。

6. 提交 RFC：

```
aws amscm submit-rfc --rfc-id RFC_ID
```

   如果 RFC 成功，则不会收到任何输出。

7. 提交 RFC：

```
aws amscm get-rfc --rfc-id RFC_ID
```

创建 CodeDeploy 部署组

创建 CodeDeploy 部署组。

CodeDeploy 部署组定义了一组以部署为目标的单个实例。

所需数据：

- VpcId：您正在使用的 VPC，应与之前使用的 VPC 相同。

- CodeDeployApplicationName：使用您之前创建的值。
- CodeDeployAutoScalingGroups：使用您之前创建的 Auto Scaling 组的名称。
- CodeDeployDeploymentGroupName：部署组的名称。此名称对于与部署组关联的每个应用程序来说必须是唯一的。
- CodeDeployServiceRoleArn：使用示例中给出的公式。
- ChangeTypeIdandChangeTypeVersion：本演练的更改类型 ID 是ct-2gd0u847qd9d2，要找出最新版本，请运行以下命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=ct-2gd0u847qd9d2
```

1. 将执行参数 JSON 架构输出到当前文件夹中的一个文件中；示例将其命名为 Cre CDDep GroupParams ate.json。

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

2. 按如下方式修改和保存 JSON 文件；您可以删除和替换其中的内容。

```
{
"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. 将的 JSON 模板输出 CreateRfc 到当前文件夹中的一个文件中；示例将其命名为 Cre CDDep GroupRfc ate.json。

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

4. 按如下方式修改和保存 JSON 文件；您可以删除和替换其中的内容。请注意，RequestedStartTime和现在RequestedEndTime是可选的；排除它们会导致 RFC 在获得批准后立即执行（这通常是自动发生的）。要提交 “计划的” RFC，请添加这些值。

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

5. 创建 RFC，指定创建CDDepGroupRfc 文件和执行参数文件：

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

   您将在响应中收到新 RFC 的 RFC ID。保存 ID 以供后续步骤使用。

6. 提交 RFC：

```
aws amscm submit-rfc --rfc-id RFC_ID
```

   如果 RFC 成功，则不会收到任何输出。

7. 检查 RFC 状态：

```
aws amscm get-rfc --rfc-id RFC_ID
```

上传 WordPress 应用程序

您可以自动访问自己创建的任何 S3 存储桶实例。您可以通过 Bastions（请参阅访问实例）或 S3 控制台访问它，然后上传 CodeDeploy 捆绑包。要继续部署堆栈，捆绑包需要准备就绪。该示例使用先前创建的存储桶名称。

```
aws s3 cp wordpress/wordpress.zip s3://ACCOUNT_ID-codedeploy-bundles/
```

使用部署 WordPress 应用程序 CodeDeploy

部署 CodeDeploy 应用程序。

拥有 CodeDeploy 应用程序包和部署组后，使用此 RFC 部署应用程序。

所需数据：

- VPC-ID：您正在使用的 VPC，应与之前使用的 VPC 相同。
- CodeDeployApplicationName：使用您之前创建的 CodeDeploy 应用程序的名称。
- CodeDeployDeploymentGroupName：使用您之前创建的 CodeDeploy 部署组的名称。
- S3Location（您上传应用程序包的位置）:S3Bucket: 您之前创建的，S3BundleType以及S3Key：您放在 S3 商店中的捆绑包的类型和名称。 BucketName
- ChangeTypeIdandChangeTypeVersion：本演练的更改类型 ID 是ct-2edc3sd1sqmrb，要找出最新版本，请运行以下命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=ct-2edc3sd1sqmrb
```

1. 将 CodeDeploy 应用程序部署 CT 的执行参数 JSON 架构输出到当前文件夹中的一个文件中；示例将其命名为 Deploy P CDApp arams.json。

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

2. 按如下方式修改 JSON 文件；您可以删除和替换其中的内容。对于 S3BucketBucketName，请使用您之前创建的。

```
{
"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. 将的 JSON 模板输出 CreateRfc 到当前文件夹中的一个文件中；示例将其命名为 Deploy CDApp rfc.json：

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

4. 修改并保存 Deplo CDApp y rfc.json 文件；您可以删除和替换其中的内容。

```
{
"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. 创建 RFC，指定执行参数文件和 Deploy CDApp Rfc 文件：

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

您会在回复 RfcId 中收到新 RFC 的信息。保存 ID 以供后续步骤使用。

6. 提交 RFC：

```
aws amscm submit-rfc --rfc-id RFC_ID
```

如果 RFC 成功，则不会收到任何输出。

验证应用程序部署

导航到先前创建的负载均衡器的终端节点 (ELB CName)， WordPress 部署的路径为：/WordPress。例如：

```
http://stack-ID-FOR-ELB.us-east-1.elb.amazonaws.com/WordPress
```

拆除应用程序部署

要取消部署，您需要提交针对 RDS 数据库堆栈、应用程序负载均衡器、Auto Scaling 组、S3 存储桶以及 Code Deploy 应用程序和组（总共六个）的删除堆栈 C RFCs T。此外，您可以提交服务请求以删除 RDS 快照（它们将在十天后自动删除，但在此期间确实会花费少量费用）。收集所有 IDs 人的堆栈，然后按照以下步骤操作。

本演练提供了使用 AMS 控制台删除 S3 堆栈的示例；此过程适用于使用 AMS 控制台删除任何堆栈。

> **Note**
>
> 如果删除 S3 存储桶，则必须先将其中的对象清空。

所需数据：

- `StackId`: 要使用的堆栈。你可以通过查看 AMS Console Stack s 页面来找到它，该页面可通过左侧导航栏中的链接获得。使用 AMS SKMS API/CLI 运行有关 AMS SKMS API 参考，请参阅 AWS Artifact 控制台中的 “报告” 选项卡。操作（在 CLI 中`list-stack-summaries`）。
- 本演练的更改类型 ID 为`ct-0q0bic0ywqk6c`，版本为 “1.0”，要查找最新版本，请运行以下命令：

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=ct-0q0bic0ywqk6c
```

内联创建：

- 使用内联提供的执行参数发出 create RFC 命令（内联提供执行参数时使用转义引号）。E

```
aws amscm create-rfc --change-type-id "ct-0q0bic0ywqk6c" --change-type-version "1.0"
 --title "Delete My Stack" --execution-parameters "{\"StackId\":\"STACK_ID\"}"
```

- 使用创建 RFC 操作中返回的 RFC 编号提交 RFC。在提交之前，RFC 仍处于该`Editing`状态，不会被付诸行动。

```
aws amscm submit-rfc --rfc-id RFC_ID
```

- 监控 RFC 状态并查看执行输出：

```
aws amscm get-rfc --rfc-id RFC_ID
```

模板创建：

1. 将 RFC 模板输出到当前文件夹中的一个文件中；示例将其命名为 DeleteStackRfc .json：

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

2. 修改并保存 DeleteStackRfc .json 文件。由于删除堆栈只有一个执行参数，因此执行参数可以在 DeleteStackRfc .json 文件本身中（无需创建带有执行参数的单独的 JSON 文件）。

   ExecutionParameters JSON 扩展中的内部引号必须使用反斜杠 (\) 进行转义。没有开始和结束时间的示例：

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-0q0bic0ywqk6c",
"Title":                "Delete-My-Stack-RFC"
"ExecutionParameters":  "{
        \"StackId\":\"STACK_ID\"}"
}
```

3. 创建 RFC：

```
aws amscm create-rfc --cli-input-json file://DeleteStackRfc.json
```

   您会在回复 RfcId 中收到新 RFC 的信息。例如：

```
{
"RfcId": "daaa1867-ffc5-1473-192a-842f6b326102"
}
```

   保存 ID 以供后续步骤使用。

4. 提交 RFC：

```
aws amscm submit-rfc --rfc-id RFC_ID
```

如果 RFC 成功，则不会在命令行收到任何确认。

5. 要监控请求的状态并查看执行输出，请执行以下操作：

```
aws amscm get-rfc --rfc-id RFC_ID --query "Rfc.
{Status:Status.Name,Exec:ExecutionOutput}" --output table
```

应用程序维护

部署基础设施后，在所有 AMS 环境（从 QA 到暂存再到生产）中以一致的方式对其进行更新就是一项挑战。

本节概述了 AMS 工作负载摄取流程，并举例说明了您可以用来保持云基础设施层处于最新状态的不同方法。

应用程序维护策略

部署应用程序的方式会影响应用程序的维护方式。本节提供了一些应用程序维护策略。

环境更新可能涉及以下任何更改：

- 安全更新
- 应用程序的新版本
- 应用程序配置更改
- 依赖项的更新

> **Note**
>
> 对于任何应用程序部署，无论采用何种方法，请务必事先提交服务请求，让 AMS 知道您将要部署应用程序。

不可变与可变应用程序安装示例

计算实例可变性	应用程序安装方法	AMI
Mutable	与 CodeDeploy	AMS 提供
	手动方式	
	使用厨师或木偶，Pull-Based	
	使用 Ansible 或 Salt，基于 Push	

计算实例可变性	应用程序安装方法	AMI
不可变的	有了金色 AMI	自定义（基于 AMS 提供的信息）

使用启用了 AMI 的可变 CodeDeploy部署

AWS CodeDeploy 是一项可自动将代码部署到任何实例（包括 Amazon EC2 实例和本地运行的实例）的服务。您可以 CodeDeploy 与 AMS 配合使用来创建和部署 CodeDeploy 应用程序。请注意，AMS 为 CodeDeploy 应用程序提供了默认的实例配置文件。

- 亚马逊 Linux（第 1 版）
- Amazon Linux 2
- RedHat 7
- CentOS 7

在 CodeDeploy 首次使用之前，必须完成一些设置步骤：

1. 安装或升级 AWS CLI
2. 为 AWS 创建服务角色 CodeDeploy，在部署中使用服务角色 ARN

IDs 所有 CT 选项都可以在更改类型参考中找到。

> **Note**
>
> 目前，您必须在此解决方案中使用 Amazon S3 存储。

此处概述了基本步骤，《AMS 用户指南》中详细介绍了操作步骤。

1. 创建 Amazon S3 存储桶。CT：ct-1a68ck03fn98r。S3 存储桶必须启用版本控制（有关执行此操作的信息，请参阅启用存储桶版本控制）。
2. 把你捆绑的 CodeDeploy 神器放在上面。您可以通过 Amazon S3 控制台执行此操作，而无需通过 AMS 请求访问权限。或者使用这个命令的变体：

```
aws s3 cp ZIP_FILEPATH_AND_NAME s3://S3BUCKET_NAME/
```

3. 查找 AMS `customer-` AMI；使用以下任一方法：
   - AMS 控制台：相关 VPC 的 VPC 详细信息页面
   - AMS API 有关 AMS SKMS API 参考，请参阅 AWS Artifact 控制台中的 “报告” 选项卡。或 CLI：`aws amsskms list-amis`
4. 创建自动扩缩组 (ASG)。CT：ct-2tylseo8rxfsc。指定 AMS AMI，将负载均衡器设置为开放端口，`customer-mc-ec2-instance-profile`为指定`ASGIAMInstanceProfile`。
5. 创建您的 CodeDeploy 应用程序。CT：ct-0ah3gwb9seqk2。参数包括应用程序名称；例如`WordpressProd`。
6. 创建您的 CodeDeploy 部署组。CT：ct-2gd0u847qd9d2。参数包括您的 CodeDeploy 应用程序名称、ASG 名称、配置类型名称和服务角色 ARN。
7. 部署 CodeDeploy 应用程序。CT：ct-2edc3sd1sd1sqmrb。参数包括您的 CodeDeploy 应用程序名称、配置类型名称、部署组名称、修订类型以及项目所在的 S3 存储 CodeDeploy 桶位置。

可变部署、手动配置和更新的应用程序实例

此应用程序部署策略是对应用程序实例的简单手动更新。这些是基本步骤。

IDs 所有 CT 选项都可以在更改类型参考中找到。

> **Note**
>
> 目前，您必须在此解决方案中使用 Amazon S3 存储。

此处概述了基本步骤；AMS 用户指南中详细介绍了各种步骤。

1. 创建 Amazon S3 存储桶。CT：ct-1a68ck03fn98r。S3 存储桶必须启用版本控制（有关执行此操作的信息，请参阅启用存储桶版本控制）。
2. 将捆绑的应用程序工件放在上面（应用程序启动并运行所需的一切）。您可以通过 Amazon S3 控制台执行此操作，而无需通过 AMS 请求访问权限。或者使用这个命令的变体：

```
aws s3 cp ZIP_FILEPATH_AND_NAME s3://S3BUCKET_NAME/
```

3. 找一个 AMS AMI，一切都会出现在他们身 CodeDeploy 上。要查找 “客户” AMI，请使用以下任一方法：

   - AMS 控制台：相关 VPC 的 VPC 详细信息页面
   - AMS API 有关 AMS SKMS API 参考，请参阅 AWS Artifact 控制台中的 “报告” 选项卡。或 CLI：`aws amsskms list-amis`

4. 使用该 AMI 创建 EC2 实例。CT：ct-14027q0sjyt1h。指定 AMS AMI，设置标签`Key=backup,Value=truecustomer-mc-ec2-instance-profile`并为`InstanceProfile`参数指定。记下返回的实例 ID。
5. 请求管理员访问该实例。CT：ct-1dmlg9g1l91h6。你的账户需要有 FQDN。如果您不确定自己的 FQDN 是什么，可以通过以下方式找到它：

   - 使用目录服务的 AWS 管理控制台（在 “安全和身份” 下）“目录名称” 选项卡。
   - 运行以下命令之一（返回目录类；DC+DC+DC=FQDN）：Windows：或 Linux：。`whoami /fqdn` `hostname --fqdn`

6. 登录实例，请参阅 AMS 用户指南中的通过堡垒访问实例。
7. 将捆绑的应用程序文件从 S3 存储桶下载到实例。
8. 向 AMS 请求服务请求立即进行备份，您需要知道实例 ID。
9. 当您需要更新应用程序时，请将新文件加载到 S3 存储桶，然后按照步骤 3 到 8 进行操作。

使用基于拉取的部署工具配置的 AMI 进行可变部署

此策略依赖于 Managed Services EC2 Create CT 中的`InstanceUserData`参数。有关使用此参数的更多信息，请参阅使用用户数据配置实例。此示例假设使用基于拉取的应用程序部署工具，例如 Chef 或 Puppet。

所有 AMS 都支持该 CodeDeploy 代理 AMIs。以下是支持的列表 AMIs：

- 亚马逊 Linux（第 1 版）
- Amazon Linux 2
- RedHat 7
- CentOS 7

IDs 所有 CT 选项都可以在变更类型参考中找到。

> **Note**
>
> 目前，您必须在此解决方案中使用 Amazon S3 存储。

此处概述了基本步骤，《AMS 用户指南》中详细介绍了操作步骤。

1. 创建 Amazon S3 存储桶。CT：ct-1a68ck03fn98r。S3 存储桶必须启用版本控制（有关执行此操作的信息，请参阅启用存储桶版本控制）。
2. 把你捆绑的 CodeDeploy 神器放在上面。您可以通过 Amazon S3 控制台执行此操作，而无需通过 AMS 请求访问权限。或者使用这个命令的变体：

   ```
   aws s3 cp ZIP_FILEPATH_AND_NAME s3://S3BUCKET_NAME/
   ```

3. 查找 AMS `customer-` AMI；使用以下任一方法：
   - AMS 控制台：相关 VPC 的 VPC 详细信息页面
   - AMS API 有关 AMS SKMS API 参考，请参阅 AWS Artifact 控制台中的 “报告” 选项卡。或 CLI：`aws amsskms list-amis`
4. 创建实 EC2 例。CT：ct-14027q0sjyt1h；设置一个标签`Key=backup, Value=true`，然后使用`InstanceUserData`参数指定引导程序和其他脚本（下载 Chef/Puppet 代理等），并包括必要的授权密钥。您可以在《AMS 用户指南》的 “变更管理” 部分的创建 HA 双层部署示例中找到执行此操作的示例。或者，请求访问并登录该实例，然后使用必要的部署工件对其进行配置。请记住，基于拉取的部署命令会从您的实例上的代理发送到您的公司主服务器，并且可能需要授权才能通过堡垒。您可能需要向 AMS 提出服务请求，才能在没有堡垒的情况下请求安全 group/AD 组访问权限。
5. 重复步骤 4，创建另一个 EC2 实例，并使用部署工具主服务器对其进行配置。
6. 当您需要更新应用程序时，请使用部署工具将更新部署到您的实例。

使用基于推送的部署工具配置的 AMI 进行可变部署

此策略依赖于 Managed Services EC2 Create CT 中的`InstanceUserData`参数。有关使用此参数的更多信息，请参阅使用用户数据配置实例。此示例假设使用基于拉取的应用程序部署工具，例如 Chef 或 Puppet。

IDs 所有 CT 选项都可以在更改类型参考中找到。

> **Note**
>
> 目前，您必须在此解决方案中使用 Amazon S3 存储。

此处概述了基本步骤，《AMS 用户指南》中详细介绍了操作步骤。

1. 创建 Amazon S3 存储桶。CT：ct-1a68ck03fn98r。S3 存储桶必须启用版本控制（有关执行此操作的信息，请参阅启用存储桶版本控制）。
2. 把你捆绑的 CodeDeploy 神器放在上面。您可以通过 Amazon S3 控制台执行此操作，而无需通过 AMS 请求访问权限。或者使用这个命令的变体：

   ```
   aws s3 cp ZIP_FILEPATH_AND_NAME s3://S3BUCKET_NAME/
   ```

3. 找一个 AMS AMI，一切都会出现在他们身 CodeDeploy 上。要查找 “客户” AMI，请使用以下任一方法：

   - AMS 控制台：相关 VPC 的 VPC 详细信息页面
   - AMS API 有关 AMS SKMS API 参考，请参阅 AWS Artifact 控制台中的 “报告” 选项卡。或 CLI：`aws amsskms list-amis`

4. 创建实 EC2 例。CT：ct-14027q0sjyt1h；设置标签，然后使用`InstanceUserData`参数运行引导程序和其他脚本，包括授权密钥`Key=backup, Value=true`、SALT 堆栈（引导小兵——有关更多信息，请参阅使用 Cl oud-Init 的 Linux EC2 上的 Bootstrapping Salt）或 Ansible（安装密钥对），了解更多信息，请参阅 Ansible 入门和动态亚马逊库存管理）。 EC2 或者，请求访问并登录实例，然后使用必要的部署工件对其进行配置。请记住，基于推送的命令从您的公司子网发送到您的实例，您可能需要为它们配置授权才能通过堡垒。您可能需要向 AMS 提出服务请求，才能在没有堡垒的情况下请求安全 group/AD 组访问权限。
5. 重复步骤 4，创建另一个 EC2 实例，并使用部署工具主服务器对其进行配置。
6. 当您需要更新应用程序时，请使用部署工具将更新部署到您的实例。

使用金色 AMI 进行不可变部署

此策略采用 “黄金” AMI，您已将其配置为按照您希望的所有应用程序实例运行。例如，使用此金色 AMI 创建的实例将自行加入正确的域和 DNS、自行配置、重启并启动所有必要的系统。当您想要更新应用程序实例时，可以重新创建金色 AMI 并使用它推出全新的应用程序实例。

所有 AMS 都支持该 CodeDeploy 代理 AMIs。以下是支持的列表 AMIs：

- 亚马逊 Linux（第 1 版）
- Amazon Linux 2
- RedHat 7
- CentOS 7

IDs 所有 CT 选项都可以在更改类型参考中找到。

> **Note**
>
> 目前，您必须在此解决方案中使用 Amazon S3 存储。

1. 创建 Amazon S3 存储桶。CT：ct-1a68ck03fn98r。S3 存储桶必须启用版本控制（有关执行此操作的信息，请参阅启用存储桶版本控制）。
2. 将捆绑的应用程序工件放在上面（应用程序启动并运行所需的一切）。您可以通过 Amazon S3 控制台执行此操作，而无需通过 AMS 请求访问权限。或者使用这个命令的变体：

   ```
   aws s3 cp ZIP_FILEPATH_AND_NAME s3://S3BUCKET_NAME/
   ```

3. 查找 AMS `customer-` AMI；使用以下任一方法：

   - AMS 控制台：相关 VPC 的 VPC 详细信息页面
   - AMS API 有关 AMS SKMS API 参考，请参阅 AWS Artifact 控制台中的 “报告” 选项卡。或 CLI：`aws amsskms list-amis`

4. 使用该 AMI 创建 EC2 实例。CT：ct-14027q0sjyt1h。指定 AMS AMI，设置标签`Key=backup,Value=true`并指定`customer-mc-ec2-instance-profile`InstanceProfile。记下返回的实例 ID。
5. 请求管理员访问该实例。CT：ct-1dmlg9g1l91h6。你的账户需要有 FQDN。如果您不确定自己的 FQDN 是什么，可以通过以下方式找到它：

   - 使用目录服务的 AWS 管理控制台（在 “安全和身份” 下）“目录名称” 选项卡。
   - 运行以下命令之一（返回目录类；DC+DC+DC=FQDN）：Windows：或 Linux：。`whoami /fqdn hostname --fqdn`

6. 登录实例，请参阅 AMS 用户指南中的访问实例。
7. 从 S3 存储桶中将捆绑的应用程序文件下载到实例。配置实例，使其在启动时自行部署功能齐全的应用程序。

8. 在实例上创建黄金 AMI。CT：ct-3rqqu43krekby。有关详细信息，请参阅 AMI | 创建。
9. 配置 Auto Scaling 组以使用该 AMI 创建新实例。CT：ct-2tylseo8rxfsc。当您需要更新应用程序时，请按照以下步骤操作，并请求 AMS 更新 ASG 以使用新的金色 AMI；为此使用管理 | 其他 | 其他 | 更新 CT。

更新策略

您可以采用几种不同的策略来更新 AMS 托管环境中的应用程序或实例。

- 计划停机时间：这种简单的策略包括安排应用程序离线和手动更新的时间。为此，请提交管理 | 其他 | 其他 | 更新 CT (ct-0xdawir96cy7k) 请求以停止所需实例。进行必要的更新，然后提交另一个管理 | 其他 | 其他 | 更新 CT (ct-0xdawir96cy7k) 请求以启动实例。

- 蓝/绿：此策略要求您有一个冗余环境（两个完全正常运行的环境），并使用域名系统 (DNS) 或 Web 防火墙 (WAF) 更新使一个环境脱机以重定向流量。更新一个环境，然后再次重定向以更新另一个环境。

  要了解更多信息，请参阅 AWS CodeDeploy 推出 Blue/Green 部署。

- 使用新 AMI 进行滚动更新：在这里，您可以自定义一个新的 AMI（请参阅创建 AMI），然后请求 AM S 将其部署到您的 Auto Scaling 组。使用管理 | 其他 | 其他 | 更新 CT (ct-0xdawir96cy7k) 来执行此操作。

AWS Managed Services 资源调度器

使用 AWS Managed Services (AMS) 资源计划程序安排账户中 AutoScaling 群组、Amazon EC2 实例和 RDS 实例的自动启动和停止。在资源本来不打算全天候运行的情况下，这有助于降低基础设施成本。该解决方案建立在实例调度器之上 AWS，但包含特定于 AMS 需求的其他功能和自定义设置。

> **Note**
>
> 默认情况下，AMS 资源调度器不与不属于 AWS CloudFormation 堆栈的资源进行交互。资源必须是以 “stack-”、“sc-” 或 “SC-” 开头的堆栈的一部分。要计划不属于 CloudFormation 堆栈的资源，可以将资源调度器堆栈参数更新`ScheduleNonStackResources`为`Yes`。

AMS 资源调度器使用周期和计划：

- 周期定义资源调度器运行的时间，例如开始时间、结束时间和当月中的几天。
- 计划包含您定义的时间段以及其他配置，例如 SSM 维护窗口、时区、休眠设置等；并根据配置的时间段规则指定资源何时运行。

您可以使用 AMS 资源计划程序的自动更改类型（CTs）来配置这些时间段和计划。

有关 AMS 资源调度器可用设置的完整详细信息，请参阅解决方案组件中的相应 AWS 实例计划程序文档。有关解决方案的架构视图，请参阅 Architecture overview.html 上的相应 AWS 实例调度器文档。

部署 AMS 资源调度器

要部署 AMS 资源调度器，请使用自动更改类型 (CT)：部署 | AMS 资源调度器 | 解决方案 | 部署 (ct-0ywnhc8e5k9z5) 提出 RFC，然后在您的账户中部署解决方案。执行 RFC 后，包含默认配置的 AMS 资源调度器资源的 CloudFormation 堆栈将自动配置到您的账户中。有关资源调度器更改类型的更多信息，请参阅 AMS 资源调度器。

> **Note**
>
> 要了解您的账户中是否已部署 AMS 资源调度器，请查看该账户的 Lamb AWS da 控制台并查找计划程序函数AMSResource。

在您的账户中配置 AMS 资源调度器后，我们建议您查看默认配置，并在需要时根据自己的偏好自定义配置，例如标签密钥、时区、计划服务等。有关推荐的自定义项的详细信息自定义 AMS 资源调度器，请参阅 “下一步”。

要进行自定义配置，或者只是确认资源调度器配置，

自定义 AMS 资源调度器

我们建议您使用更新的 AMS 资源计划程序更改类型自定义 AMS 资源调度器的以下属性，请参阅 AMS 资源调度器。

- 标签名称：资源调度器将用于将实例计划与资源关联的标签的名称。默认值为 “计划”。
- 定时服务：以逗号分隔的资源调度器可以管理的服务列表。默认值为 “ec2、rds、自动缩放”。有效值为 “ec2”、“rds” 和 “自动缩放”
- 默认时区：指定资源调度器要使用的默认时区。默认值为 UTC。

- 使用 CMK：以逗号分隔的 Amazon KMS 客户托管密钥 (CMK) 列表 ARNs ，可以向资源计划程序授予权限。
- 使用 LicenseManager：以逗号分隔的 AWS 许可证管理器列表，可以 ARNs 向该资源调度器授予权限。

> **Note**
>
> AMS 可能会不时发布功能和修复程序，以使您的账户中的 AMS 资源调度器保持最新状态。发生这种情况时，您对 AMS 资源调度器所做的任何自定义都将保留。

使用 AMS 资源调度器

要在解决方案部署后配置 AMS 资源调度器，请使用自动资源调度器 CTs 创建、删除、更新和描述（获取有关）AMS 资源调度器周期（资源调度器运行的时间）和计划（配置的时间段和其他选项）。有关使用 AMS 资源计划程序更改类型的示例，请参阅 AMS 资源计划程序。

要选择由 AMS 资源调度器管理的资源，在创建部署和计划后，您可以使用 AMS 标签创建 CTs 来标记 Auto Scaling 组、Amazon RDS 堆栈和 Amazon EC2 资源，使用您在部署期间提供的标签密钥和定义的时间表作为标签值。标记资源后，将根据您定义的资源计划程序计划安排资源启动或停止。

使用 AMS 资源调度器不会产生额外费用。但是，该解决方案会使用多个资源， AWS 服务 并且您需要为这些资源的使用付费。有关更多详细信息，请参阅架构概述。

要选择退出 AMS 资源调度器，请执行以下操作：

- 对于临时选择退出或禁用：使用自动管理 | AMS 资源调度器 | 状态 | 禁用更改类型 (ct-14v49adibs4db) 提交 RFC
- 要永久移除：提交管理层 | 其他 | 其他 | 更新（需要审核）(ct-0xdawir96cy7k) RFC 请求从资源调度器发布自动化系统中删除

AMS 资源调度器成本估算器

为了跟踪成本节省情况，AMS 资源调度器具有一个组件，该组件每小时计算由计划程序管理的 Amazon EC2 和 RDS 资源的估计成本节约。然后，这些成本节省数据将作为 CloudWatch 指标 (`AMS/ResourceScheduler`) 发布，以帮助您对其进行跟踪。成本节省估算器仅估算实例运行时间节省的时间。它不考虑任何其他成本，例如与资源相关的数据传输成本。

使用资源调度器启用了成本节约估算器。它每小时运行一次，并从中 AWS Cost Explorer检索成本和使用数据。它根据该数据计算出每种实例类型的平均每小时成本，然后预测在未计划的情况下运行一整天的成本。节省的成本是给定日期的预计成本与Cost Explorer的实际报告成本之间的差额。

例如，如果实例 A 的资源调度器配置为从上午 9 点到下午 5 点运行，则在给定一天中运行八个小时。Cost Explorer 将成本报告为 1 美元，使用量报告为 8。因此，每小时的平均成本为0.125美元。如果实例未使用资源调度器进行计划，则该实例将在当天运行 24 小时。在这种情况下，成本将为24x0.125 = 3美元。资源调度器帮助您节省了2美元的成本。

为了使成本节省估算器仅检索由 Cost Explorer 中的资源调度器管理的资源的成本和使用量，需要在账单控制面板中激活资源调度器用于定位资源的标签密钥作为成本分配标签。如果该账户属于某个组织，则需要在该组织的管理账户中激活标签密钥。有关执行此操作的信息，请参阅激活用户定义的成本分配标签和用户定义的成本分配标签

将标签密钥激活为成本分配标签后， AWS 计费开始跟踪资源调度器管理的资源的成本和使用情况，在这些数据可用之后，成本节省估算器开始计算节省的成本并将数据发布在中的`AMS/ResourceScheduler`指标命名空间下。 CloudWatch

成本估算器提示

Cost Savings Estimator 在计算时不考虑预留实例、储蓄计划等折扣。估算器从 Cost Explorer 中提取使用成本，然后计算资源每小时的平均成本。有关更多详细信息，请参阅 “了解您的 AWS 成本数据集：备忘单”

为了使成本节省估算器仅检索由 Cost Explorer 中的资源调度器管理的资源的成本和使用量，需要在账单控制面板中激活资源调度器用于定位资源的标签密钥作为成本分配标签。如果该账户属于某个组织，则需要在该组织的管理账户中激活标签密钥。有关执行此操作的信息，请参阅用户定义的成本分配标签。如果未激活成本分配标签，则即使启用了该指标，估算器也无法计算节省的费用并发布该指标。

AMS 资源调度器最佳实践

计划 Amazon EC2 实例

- 实例关闭行为必须设置为`stop`，而不是设置为`terminate`。`stop`对于使用 AMS Amazon Create 自动更改类型 (ct-14027q0sjyt1h) EC2 创建的实例，通过将属性设置为，可以为通过摄取创建的 EC2 亚马逊实例将其设置为。 AWS CloudFormation `InstanceInitiatedShutdownBehavior` `stop`如果实例的关闭行为设置为`terminate`，则当资源调度器停止实例并且调度器无法启动它们时，实例就会结束。
- 作为 Auto Scaling 组一部分的 Amazon EC2 实例不会由 AMS 资源计划程序单独处理，即使它们已被标记。

- 如果目标实例根卷使用 KMS 客户主密钥 (CMK) 加密，则需要向您的资源调度器 IAM 角色添加额外的`kms:CreateGrant`权限，以便计划程序能够启动此类实例。为了提高安全性，默认情况下不会将此权限添加到角色中。如果您需要此权限，请使用管理 | AMS 资源计划程序 | 解决方案 | 更新更改类型提交 RFC，并指定以逗号分隔的 KMS 列表。 ARNs CMKs

安排 Auto Scaling 群组

- AMS 资源调度器启动或停止 Auto Scaling 组的自动扩展，而不是组中的单个实例。也就是说，调度器恢复 Auto Scaling 组的大小（开始）或将大小设置为 0（停止）。
- 使用指定标签标记 AutoScaling 组，而不是该组中的实例。
- 在停止期间，AMS 资源调度器会存储 Auto Scaling 组的最小、所需和最大容量值，并将最小和所需容量设置为 0。在启动期间，调度器会将 Auto Scaling 组的大小恢复为停止时的状态。因此，Auto Scaling 组实例必须使用适当的容量配置，这样实例的终止和重新启动就不会影响在 Auto Scaling 组中运行的任何应用程序。
- 如果在运行期间修改 Auto Scaling 组（最小或最大容量），则计划程序会存储新的 Auto Scaling 组大小，并在停止计划结束时恢复该组时使用该大小。

安排 Amazon RDS 实例

- 调度器可以在停止 RDS 实例之前拍摄快照（不适用于 Aurora 数据库集群）。默认情况下，此功能处于开启状态，创建 RDS 实例快照 CloudFormation 模板参数设置为 true。快照将一直保留到下次停止 Amazon RDS 实例并创建新快照为止。

  调度器可以是属于集群或 start/stop Amazon RDS Aurora 数据库或多可用区（多可用区）配置的 Amazon RDS 实例。但是，当计划程序无法停止 Amazon RDS 实例（尤其是多可用区实例）时，请检查 Amazon RDS 限制。要安排 Aurora 集群的启动或停止，请使用安排 Aurora 集群模板参数（默认为 t rue）。Aurora 集群（不是集群中的单个实例）必须使用初始配置期间定义的标签键进行标记，并将计划名称作为标签值来调度该集群。

  每个 Amazon RDS 实例都有一个每周维护时段，在此期间会应用任何系统更改。在维护时段内，Amazon RDS 将自动启动已停止超过七天的实例以进行维护。请注意，维护事件完成后，Amazon RDS 不会停止实例。

  计划程序允许指定是否将 Amazon RDS 实例的首选维护时段作为运行时段添加到其计划中。如果没有其他运行周期指定实例应运行，并且维护事件已完成，则解决方案将在维护时段开始时启动实例，并在维护时段结束时停止该实例。

如果维护事件在维护时段结束时仍未完成，则该实例将一直运行到维护事件完成后的计划间隔。

> **Note**
>
> 调度程序不验证资源是否已启动或停止。它调用 API 并继续前进。如果 API 调用失败，它会记录错误以供调查。

应用程序安全注意事项

应用程序安全性包括考虑应用程序需要哪些权限才能运行、哪些防火墙规则、应启用哪些 IAM 角色才能访问应用程序。

要更好地了解一般 AWS 安全性，请参阅安全、身份和合规性最佳实践。

配置管理访问权限

AWS Managed Services (AMS) 力求为您提供无忧的基础设施，因此您不必担心安全问题、修补问题、备份问题等。为此，AMS 建议使用最少的 IAM 角色，仅允许特定组或主服务器（如果使用应用程序部署工具）访问运行您的应用程序的实例。

应用程序访问防火墙规则

就像操作系统 (OS) 一样，所有应用程序访问都应使用活动目录 (AD) 组进行管理。以 Amazon Relational Database Service（Amazon RDS）为例，您必须打破镜像（复制）才能添加新用户。最好的方法是在 AD 中创建一个群组，并在创建数据库时将其添加。在 AMS AD 中拥有群组意味着您可以 CTs 为应用程序访问创建群组。有关 AD 的官方分组策略的信息，请参阅使用群组嵌套策略-群组策略的 AD 最佳实践。

要了解有关域树和域名的更多信息，请参阅 parent/child 域和域名林的工作原理。

以下规则说明了适用于用户位于子域中的多域林信任的解决方案。

Windows 实例

这些是要为您的 Windows 父域和子域控制器配置的规则。

父域控制器，Windows

从：父域控制器到：Windows 堆栈和共享服务子网

源端口	目的地端口	协议
88	49152 - 65535	TCP

源端口	目的地端口	协议
389	49152 - 65535	UDP

发件人：将子网（包括共享服务）堆叠到：Windows 林根域控制器

源端口	目的地端口	协议
49152 - 65535	88	TCP
49152 - 65535	389	UDP

子域控制器，Windows

从：子域控制器到：Windows AWS 域控制器

源端口	目的地端口	协议
49152 - 65535	53	TCP
49152 - 65535	88	TCP
49152 - 65535	389	UDP

从：子域控制器到：Windows 堆栈和共享服务子网

源端口	目的地端口	协议
88	49152 - 65535	TCP
135	49152 - 65535	TCP
389	49152 - 65535	TCP
389	49152 - 65535	UDP
445	49152 - 65535	TCP

源端口	目的地端口	协议
49152 - 65535	49152 - 65535	TCP

发件人：堆叠子网（包括共享服务）到：Windows 子域控制器

源端口	目的地端口	协议
49152 - 65535	88	TCP
49152 - 65535	135	TCP
49152 - 65535	389	TCP
49152 - 65535	389	UDP
49152 - 65535	445	TCP
49152 - 65535	49152 - 65535	TCP

Linux 实

这些是要为您的 Linux 父域和子域控制器配置的规则。

所有测试均使用亚马逊 Linux 进行。虽然 Windows 的动态端口范围为 49152 到 65535，但许多 Linux 内核使用的端口范围是 32768 到 61000。运行以下命令查看 IP 端口范围。

```
cat /proc/sys/net/ipv4/ip_local_port_range
```

父域控制器，Linux

从：父域控制器到：Linux 堆栈和共享服务子网

源端口	目的地端口	协议
389	32768-61000	UDP
88	32768-61000	TCP

发件人：堆叠子网（包括共享服务）到：Linux 林根域控制器

源端口	目的地端口	协议
32768-61000	88	TCP
32768-61000	389	UDP

子域控制器，Linux

从：子域控制器到：Linux AWS 域控制器

源端口	目的地端口	协议
49152 - 65535	53	TCP
49152 - 65535	88	TCP
389	49152 - 65535	UDP
49152 - 65535	389	UDP

从：子域控制器到：Linux 堆栈和共享服务子网

源端口	目的地端口	协议
88	32768-61000	TCP
389	32768-61000	UDP

发件人：堆叠子网（包括共享服务）到：Linux 子域控制器

源端口	目的地端口	协议
32768-61000	88	TCP
32768-61000	389	UDP

AMS 出口流量管理

默认情况下，AMS 私有子网和客户应用程序子网的目标 CIDR 为 0.0.0.0/0 的路由将网络地址转换 (NAT) 网关作为目标。AMS TrendMicro 服务和补丁是必须具有互联网出口访问权限的组件，这样 AMS 才能提供其服务， TrendMicro 并且操作系统可以获取更新。

AMS 支持通过客户管理的出口设备将出口流量转移到互联网，前提是：

- 它充当隐式（例如，透明）代理。

  以及

- 它允许 AMS HTTP 和 HTTPS 依赖关系（在本节中列出），以便能够持续修补和维护 AMS 托管基础架构。

部分示例包括：

- 公交网关 (TGW) 的默认路由，通过多账户着陆区网络账户中的 AWS Direct Connect 连接指向客户管理的本地防火墙。
- TGW 有一个默认路由，指向利用 AWS 的多账户着陆区出口 VPC 中的 AWS 终端节点，指向另一个 PrivateLink AWS 账户中的客户管理的代理。
- TGW 的默认路由指向另一个 AWS 账户中的客户管理的防火墙， site-to-siteVPN 连接作为多账户着陆区 TGW 的附件。

AMS 已经确定了相应的 AMS HTTP 和 HTTPS 依赖关系，并不断开发和完善这些依赖关系。请参阅 egressMgmt.zip。除了 JSON 文件外，ZIP 还包含一个自述文件。

> **Note**
>
> - 此信息并不全面，此处未列出一些必需的外部站点。
> - 请勿在拒绝列表或屏蔽策略下使用此列表。
> - 此列表旨在作为出口过滤规则集的起点，期望使用报告工具来精确确定实际流量与列表的偏离位置。

要询问有关筛选出口流量的信息，请发送电子邮件至 CSDM：ams-csdm@amazon.com。

安全组

在 AWS 中 VPCs，AWS 安全组充当虚拟防火墙，控制一个或多个堆栈（一个或一组实例）的流量。堆栈启动时，它会与一个或多个安全组相关联，这些安全组决定了允许哪些流量到达堆栈：

- 对于公有子网中的堆栈，默认安全组接受来自所有位置（互联网）的 HTTP (80) 和 HTTPS (443) 流量。这些堆栈还接受来自您的公司网络和 AWS 堡垒的内部 SSH 和 RDP 流量。然后，这些堆栈可以通过任何端口导出到互联网。它们还可以输出到您的私有子网和公有子网中的其他堆栈。
- 私有子网中的堆栈可以输出到私有子网中的任何其他堆栈，并且堆栈中的实例可以通过任何协议相互完全通信。

> **Important**
>
> 私有子网上堆栈的默认安全组允许私有子网中的所有堆栈与该私有子网中的其他堆栈通信。如果要限制私有子网内堆栈之间的通信，则必须创建描述限制的新安全组。例如，如果您想限制与数据库服务器的通信，使该私有子网中的堆栈只能通过特定端口从特定的应用程序服务器进行通信，请请求特殊的安全组。本节将介绍如何执行此操作。

默认安全组

MALZ

下表描述了堆栈的默认入站安全组 (SG) 设置。SG 名为“SentinelDefaultSecurityGroupPrivateOnly-vpc-id”，其中是 *ID* AMS 多账户着陆区账户中的 VPC ID。允许所有流量通过此安全组出站到 SentinelDefaultSecurityGroupPrivateOnly “mc-initial-garden-”（允许堆栈子网内的所有本地流量）。

第二个安全组 “” 允许所有流量出站到 0.0.0.0/0。SentinelDefaultSecurityGroupPrivateOnly

> **Tip**
>
> 如果您为 AMS 更改类型选择安全组，例如 EC2 创建或 OpenSearch 创建域，则应使用此处描述的默认安全组之一，或者使用您创建的安全组。您可以在 AWS EC2 控制台或 VPC 控制台中找到每个 VPC 的安全组列表。

还有其他用于内部 AMS 目的的默认安全组。

AMS 默认安全组（入站流量）

Type	协议	端口范围	来源
所有流量	All	全部	SentinelDefaultSecurityGroupPrivateOnly（限制同一安全组成员的出站流量）
所有流量	All	全部	SentinelDefaultSecurityGroupPrivateOnlyEgressAll（不限制出站流量）
HTTP、HS、SSH、	TCP	80/443（来源 0.0.0.0/0） 允许从堡垒访问 SSH 和 RDP	SentinelDefaultSecurityGroupPublic（不限制出站流量）
MALZ 堡垒：			
SSH	TCP	22	SharedServices VPC CIDR 和 DMZ VPC CIDR，以及客户提供的本地部署 CIDRs
SSH	TCP	22	
RDP	TCP	3389	
RDP	TCP	3389	
SALZ 堡垒：			
SSH	TCP	22	mc-initial-garden-LinuxBastion SG
SSH	TCP	22	mc-initial-garden-LinuxBastion DMZSG
RDP	TCP	3389	mc-initial-garden-WindowsBastion SG
RDP	TCP	3389	mc-initial-garden-WindowsBastion DMZSG

SALZ

下表描述了堆栈的默认入站安全组 (SG) 设置。SG 名为 “mc-initial-garden-SentinelDefaultSecurityGroupPrivateOnly-*ID*”，其中*ID*是唯一标识符。允许所有流量通过此安全组

出站到 SentinelDefaultSecurityGroupPrivateOnly “mc-initial-garden-”（允许堆栈子网内的所有本地流量）。

第二个安全组 “mc-initial-garden--” 允许所有流量出站到 0.0.0.0/0。SentinelDefaultSecurityGroupPrivateOnlyEgressAll *ID*

> **Tip**
>
> 如果您为 AMS 更改类型选择安全组，例如 EC2 创建或 OpenSearch 创建域，则应使用此处描述的默认安全组之一，或者使用您创建的安全组。您可以在 AWS EC2 控制台或 VPC 控制台中找到每个 VPC 的安全组列表。

还有其他用于内部 AMS 目的的默认安全组。

AMS 默认安全组（入站流量）

Type	协议	端口范围	来源
所有流量	All	全部	SentinelDefaultSecurityGroupPrivateOnly（限制同一安全组成员的出站流量）
所有流量	All	全部	SentinelDefaultSecurityGroupPrivateOnlyEgressAll（不限制出站流量）
HTTP、HS、SSH、	TCP	80/443（来源 0.0.0.0/0） 允许从堡垒访问 SSH 和 RDP	SentinelDefaultSecurityGroupPublic（不限制出站流量）
MALZ 堡垒：			
SSH	TCP	22	SharedServices VPC CIDR 和 DMZ VPC CIDR，以及客户提供的本地部署 CIDRs
SSH	TCP	22	
RDP	TCP	3389	
RDP	TCP	3389	

Type	协议	端口范围	来源
SALZ 堡垒：			
SSH	TCP	22	mc-initial-garden-LinuxBastion SG
SSH	TCP	22	mc-initial-garden-LinuxBastion DMZSG
RDP	TCP	3389	mc-initial-garden-WindowsBastion SG
RDP	TCP	3389	mc-initial-garden-WindowsBastion DMZSG

创建、更改或删除安全组

您可以请求自定义安全组。如果默认安全组不能满足您的应用程序或组织的需求，则可以修改或创建新的安全组。此类申请将被视为需要批准，并将由AMS运营团队进行审查。

要在堆栈之外创建安全组 VPCs，请使用`Deployment | Advanced stack components | Security group | Create (managed automation)`更改类型 (ct-1oxx2g2d7hc90) 提交 RFC。

要修改活动目录 (AD) 安全组，请使用以下更改类型：

- 添加用户：使用管理 | Directory Service | 用户和群组 | 将用户添加到群组 [ct-24pi85mjtza8k] 提交 RFC
- 要移除用户：使用管理 | Directory Service | 用户和群组 | 从群组中移除用户 [ct-2019s9y3nfml4] 提交 RFC

> **Note**
>
> 使用手动操作时 CTs，AMS 建议您使用 “尽快安排” 选项（在控制台中选择 “尽快”，在 AP I/CLI 中将开始和结束时间留空），因为这 CTs 需要 AMS 操作员检查 RFC，并可能在批准和运行之前与您沟通。如果您安排这些活动 RFCs，请务必留出至少 24 小时的时间。如果在预定开始时间之前未获得批准，RFC 将被自动拒绝。

查找安全组

要查找附加到堆栈或实例的安全组，请使用 EC2 控制台。找到堆栈或实例后，您可以看到与其关联的所有安全组。

有关在命令行中查找安全组并筛选输出的方法，请参阅describe-security-groups。

附录：应用程序入职问卷

使用此问卷描述您的部署元素和结构，以便 AMS 可以确定需要哪些基础设施组件。 Line-of-Business(LoB) 应用程序的入职要求与产品应用程序有很大不同，因此本问卷旨在解决这两个问题。

主题

部署摘要

部署的描述。例如：

- 此帐户用于 Line-of-Business (LoB) 应用程序部署（而不是产品应用程序部署）。
- 部署涉及账户子网内的自动扩展 ARP（经过身份验证的 public/DMZ 反向代理）。
- Web 和应用程序服务器将部署在账户的私有子网中。
- Amazon RDS（亚马逊关系数据库服务）实例也将部署在账户的私有子网中。
- 服务器（ARP、Web、应用程序、数据库、负载均衡器等）分为不同的安全组。
- 该账户需要跨可用区 () (即多可用区）进行高可用性（高可用性AZs）设计。

基础架构部署组件

为了支持您的应用程序，需要配置哪些不同的组件？

- 区域：需要什么 AWS 区域 或区域？
- 高可用性 (HA)：将使用哪些可用区？
- 虚拟私有云 (VPC) Private Cloud：VPC 的 CIDR 块是什么？
- 需要哪些服务器实例？

  - 经过身份验证的反向代理 (ARP)：操作系统、AMI、实例类型、子网 ID、安全组、入口端口？
  - 应用程序部署工具服务器：操作系统、AMI、实例类型、子网 ID、安全组、入口端口（Chef、Puppet）或出口端口（Ansible、Saltstack）端口？
  - 带有 MySQL 的 Amazon RDS：数据库版本、使用类型、实例类别、子网 ID、安全组、数据库实例 ID、存储大小、多可用区、身份验证类型、加密？
  - 存储：您的应用程序是无状态的吗？ 你需要 S3 存储桶吗？ 您是否需要永久存储？ 您是否需要对 EBS 卷进行静态数据加密？ 你需要数据库加密吗？
  - 外部（Managed Services VPC）服务器终端节点：SMTP？ LDAP？
  - 网络要求：网络过滤（基于安全组？）？ 网络流量检查（入站？ 出站？）？
- 标记：应该使用什么标签将资源分组为逻辑集合？ 例如，应用程序堆栈的所有资源。为您的用例选择标签；例如`backup=true`，启用备份。此外，您必须使用标签才能使您创建的任何 EC2 实例`name=value`在控制台中显示名称。
- 安全组：
  - 需要哪些安全组？
  - 安全组入口规则？
  - 安全组出站规则？

应用程序托管平台

对于您的应用程序托管平台，请考虑以下可能的要求：

- 数据库已加密？
- 加密密钥由谁管理？
- 所有传输中的数据和静态数据都已加密？
- 所有用户都通过 HTTPS 访问系统？
- 您的安全运营团队是否批准了所有 system-to-system互动？

应用程序部署模型

关于如何规划应用程序部署的注意事项。请参阅 我的运营模式是什么？。

- 自动还是手动？ 没有部署自动化意味着没有自动扩展。如果您请求访问权限并登录并手动更新应用程序，则更新失败。AMS 希望您回滚更新或通过服务请求提醒我们，以便我们为您提供帮助。

- 如果是自动化，框架是什么？ 脚本？ 基于代理 (puppet/chef)? Agentless (SALT/Ansible)？ CodeDeploy？ 基于代理和无代理的部署工具需要创建单独的实例并将其部署为该工具的主服务器。AMS 希望您了解成功部署应用程序工具所必需的所有要素；但是，我们很乐意帮助解决相关的基础设施问题。
- 您的 Line-of-Business应用程序（用于创建和管理应用程序的应用程序）是否需要修补？

应用程序依赖关系

您是否需要 Line-of-Business (LoB) 应用程序的实例？ 用于产品应用？

您的产品应用程序需要什么才能正常运行？

- 网络级别的依赖关系：例如， Direct Connect
- Package 依赖关系：例如，`pip`
- 此应用程序所依赖的应用程序：例如， MySql
- 防火墙依赖关系？

你的 LoB 应用程序需要什么才能正常运行？

- 网络级别的依赖关系：例如， Direct Connect
- Package 依赖关系：例如 Firefox Saucy
- 此应用程序所依赖的应用程序：例如， MySql
- 防火墙依赖关系？

适用于产品应用程序的 SSL 证书

您的服务器需要哪些 SSL 证书，这样您的应用程序（LoB 和产品）才能访问其运行所需的一切并可供访问？

- Auto Scaling Group？
- 数据库（亚马逊 RDS）？
- Load Balancer？
- 部署工具服务器？
- Web 应用程序防火墙 (AWS WAF)？

- 其他实例？

例如，对于上面列出的每个实例，您可能需要以下证书：

WAF（证书 1）-> elb-ext（证书 2）-> ARP（证书 3）-> elb-int（证书 4）-> 网站（证书 5）-> elb-int（证书 6）-> Web 服务（证书 7）。

文档历史记录

下表描述了此版本的 AMS 的文档。

- API 版本：2019-05- 21
- 最新文档更新：2023 年 2 月 16 日

更改	描述	链接
已移除目录链接	已移除 TOC AWS 词汇表链接。	2025年8月8日
更新内容：迁移工作负载：Windows 摄取前验证	更新了部分以包括使用验证前脚本验证前脚本来验证您的 Windows 实例是否已准备好接入您的 AMS 账户的详细步骤；。WIGs	迁移工作负载：Windows 摄取前验证
更新了内容，DMS 配置	关于所需角色的重要说明， dms-vpc-role。	1： AWS DMS 复制子网组：创建
更新了内容，CFN Ingest 支持的资源	已添加 OpenSearch。	支持的资源
更新了内容，迁移工作负载	更新了摄取前验证说明。	迁移工作负载：Windows 摄取前验证
更新了内容，CFN Ingest。	从 CFN 收录内容中移除了受限的 “支持的资源”。	CloudFormation 载入堆栈：支持的资源
更新了支持的 Windows 版本	增加了对 Windows 服务器 2022 的支持。	AMS Amazon 机器映像 (AMIs)、迁移工作负载：Linux 和

更改	描述	链接
		Windows 的先决条件 和 迁移工作负载：Windows 摄取前验证
更新了内容，资源调度器。	更新了使用专用部署 CT 的说明 ct-0ywnhc 8e5k9z5，适用于 SALZ 和 MALZ。	AMS 资源调度器快速入门
更新了内容，工作负载摄取。	更新了支持的 SUSE Linux 版本。	迁移工作负载：Linux 和 Windows 的先决条件
更新了内容，Database Migration Service。	已添加到先决条件中，并针对实用性和可用性进行了一些更改。	AWS Database Migration Service (AWS DMS)
更新了内容，工作负载摄取。	Linux 假发前验证压缩包已更新。	迁移工作负载：Linux 和 Windows 的先决条件
更新了内容。	更新了 Linux 版的假发前验证压缩包。此外，还将 Windows Server 2008 R2 添加为支持的操作系统。	迁移工作负载：Linux 和 Windows 的先决条件
新增内容	快速入门和教程已从已停用的《AMS 高级变更管理指南》移至此处。	快速入门, 教程.

更改	描述	链接
更新了内容	部署 | 高级堆栈组件 | 数据库迁移服务 (DMS) | 启动复制任务 (ct-1yq7hhqse71yg) 已更新以指明DocumentName和区域是必填参数；以前，它们被错误地列为可选参数。	Database Migration Service (DMS) | 启动复制任务
更新了内容	CloudFormation 摄取 已更新，指出了两个新的支持资源，AWS::Route53Resolver::ResolverRuleAssociation 和 AWS::Route53Resolver::ResolverRule。	支持的资源
更新了内容	迁移工作负载：Windows 摄取前验证	Sysprep 信息已更新，其中包含更多细节。 迁移工作负载：Windows 摄取前验证
更新了内容	管理 | 自定义堆栈 | 来自 CloudFormation 模板的堆栈 | 批准变更集和更新 (ct-1404e21baa2ox) 该ChangeSetName参数的 CT 演练描述已更新，其中包含其他信息。	来自 CloudFormation 模板的堆栈 | 批准变更集并更新
	Ubuntu 18.04 和 Oracle Linux 8.3 上	迁移工作负载：Linux 和 Windows 的先决条件
新内容：	通过 CFN 载入和堆栈更新进行 IAM 部署。 CTs	2022 年 2 月 10 日

更改	描述	链接
Database Migration Service (DMS) 复制任务	更改类型已更新，因此正则表达式允许 ARNs 包含连字符的任务。 启动 AWS DMS 复制任务和 D atabase Migration Service (DMS) | 停止复制任务。	2022 年 1 月 13 日
Linux WIGS 摄取前验证	zip 文件已更新。 迁移工作负载：Linux 摄取前验证。	2022 年 1 月 13 日
固定链接	数据库 (DB) 导入 AMS SQL RDS-> 设置 部分有一些错误的链接。	2022 年 1 月 13 日

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。