Opções avançadas de implantação de aplicativos AMS

Guia do desenvolvedor de aplicativos avançados do AMS

Versão August 18, 2026

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Guia do desenvolvedor de aplicativos avançados do AMS: Opções avançadas de implantação de aplicativos AMS

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

As marcas comerciais e imagens de marcas da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestigie a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

..... vii
Integração de aplicativos 1
O que é integração de aplicativos? 1
O que fazemos, o que não fazemos 2
Imagens de máquinas AMS Amazon (AMIs) 3
Segurança aprimorada AMIs 6
Principais termos 6
Qual é o meu modelo operacional? 13
Gerenciamento de serviços 14
Governança de contas 14
Início do serviço 15
Gestão do relacionamento com o cliente (CRM) 15
Processo de CRM 16
Reuniões de CRM 17
Organizações de reuniões de CRM 18
Relatórios mensais de CRM 19
Otimização de custo 20
Estrutura de otimização de custos 20
Matriz de responsabilidade de otimização de custos 23
Horário de atendimento 25
Como obter ajuda 26
Desenvolvimento de aplicações 27
Ser bem arquitetado 28
Responsabilidades da camada de aplicativo versus camada de infraestrutura 29
EC2 mutabilidade de instância 29
Usando o AWS Secrets Manager com recursos do AMS 30
Implantação de aplicativos no AMS 32
Capacidades de implantação de aplicativos 32
Planejando a implantação de seu aplicativo 36
Ingestão de workload do AMS (WIGS) 37
Migração de cargas de trabalho: pré-requisitos para Linux e Windows 37
Como a migração altera seu recurso 41
Migração de workloads: processo padrão 43
Migração de cargas de trabalho: CloudEndure landing zone (SALZ) 44

Conta de ferramentas (migrando cargas de trabalho) 47
Migração de cargas de trabalho: validação de pré-ingestão do Linux 52
Migração de workloads: validação de pré-ingestão do Windows 54
Pilha de ingestão de carga de trabalho: criação 58
CloudFormation Ingestão de AMS 63
CloudFormation Diretrizes, melhores práticas e limitações de ingestão 64
CloudFormation Ingerir: exemplos 84
Crie uma CloudFormation pilha de ingestão 91
Atualizar CloudFormation pilha de ingestão 96
Aprovar um conjunto de alterações da CloudFormation pilha de ingestão 100
Atualizar CloudFormation proteção de terminação de pilhas 103
Implantações automatizadas de IAM usando ingestão de CFN ou atualização de pilha CTs 107
CodeDeploy solicitações 112
CodeDeploy aplicação 112
CodeDeploy grupos de implantação 119
AWS Database Migration Service (AWS DMS) 126
Planejando para AWS DMS 126
Dados necessários para AWS DMS configuração 128
Tarefas para AWS DMS configuração 128
Gerenciando seu AWS DMS 159
Importação de banco de dados (DB) para o AMS RDS para SQL Server 166
Configuração 167
Importando o banco de dados 168
Limpeza 169
Implantações de aplicativos Tier and Tie 169
Implantações completas de aplicativos 170
Trabalhando com tipos de alteração de provisionamento () CTs 170
Veja se uma tomografia computadorizada existente atende aos seus requisitos 170
Solicite um novo CT 178
Teste o novo CT 179
Começos rápidos 180
Início rápido do AMS Resource Scheduler 180
Terminologia do AMS Resource Scheduler 180
Implementação do AGENDADOR DE RECURSOS 181
Configurando backups entre contas (dentro da região) 184

Tutoriais 187
Tutorial do console: pilha de dois níveis de alta disponibilidade (Linux/RHEL) 187
Antes de começar 188
Crie a infraestrutura 189
Crie, carregue e implante o aplicativo 193
Validar a implantação do aplicativo 198
Elimine a implantação de alta disponibilidade 198
Tutorial do console: implantando um site Tier and Tie WordPress 198
Criação de um RFC usando o console (noções básicas) 199
Criando a infraestrutura 201
Criar um WordPress CodeDeploy pacote 204
Implante o pacote de WordPress aplicativos com CodeDeploy 208
Validar a implantação do aplicativo 211
Elimine a implantação de aplicativos 211
Tutorial de CLI: pilha de duas camadas de alta disponibilidade (Linux/RHEL) 211
Antes de começar 212
Crie a infraestrutura 214
Crie, carregue e implante o aplicativo 219
Validar a implantação do aplicativo 225
Elimine a implantação de aplicativos 225
Tutorial de CLI: Implantação de um site Tier and Tie WordPress 227
Criando um RFC usando a CLI 228
Crie a infraestrutura 228
Crie um pacote WordPress de aplicativos para CodeDeploy 229
Implante o pacote de WordPress aplicativos com CodeDeploy 233
Validar a implantação do aplicativo 239
Acabe com a implantação de aplicativos 239
Manutenção de aplicativos 242
Estratégias de manutenção de aplicativos 242
Implantação mutável com uma AMI CodeDeploy habilitada 243
Implantação mutável, instâncias de aplicativos configuradas e atualizadas manualmente 245
Implantação mutável com uma AMI configurada por ferramenta de implantação baseada em pull 246
Implantação mutável com uma AMI configurada por ferramenta de implantação baseada em push 247
Implantação imutável com uma AMI dourada 249

Estratégias de atualização ... 250
Programador de recursos ... 251
Implantando o Agendador de Recursos ... 252
Personalizando o Agendador de Recursos ... 252
Usando o Agendador de Recursos ... 253
Estimador de custos do AMS Resource Scheduler ... 254
Melhores práticas do AMS Resource Scheduler ... 255
Considerações sobre segurança de aplicativos ... 258
Acesso para gerenciamento de configurações ... 258
Regras de firewall de acesso a aplicativos ... 258
Instâncias do Windows ... 258
Controlador de domínio principal, Windows ... 259
Controlador de domínio secundário, Windows ... 259
Instâncias Linux ... 260
Gerenciamento de tráfego de saída do AMS ... 262
Grupos de segurança ... 263
Security groups padrão ... 264
Criar, alterar ou excluir grupos de segurança ... 267
Encontre grupos de segurança ... 268
Apêndice: Questionário de integração de aplicativos ... 269
Resumo da implantação ... 269
Componentes de implantação de infraestrutura ... 270
Plataforma de hospedagem de aplicativos ... 271
Modelo de implantação de aplicativos ... 271
Dependências de aplicativos ... 271
Certificados SSL para aplicativos de produtos ... 272
Histórico do documento ... 273

Aviso de fim do suporte: em 30 de junho de 2027, o suporte para o AMS Advanced AWS será encerrado. Depois de 30 de junho de 2027, você não poderá mais acessar o console AMS Advanced ou os recursos do AMS Advanced. Para obter mais informações, consulte Fim do suporte avançado do AMS.

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.

Integração de aplicativos

Bem-vindo ao plano de operações AMS do AWS Managed Services (AMS). O objetivo deste documento é descrever os vários métodos que você pode usar ao integrar seus aplicativos ao AMS após a configuração inicial do gerenciamento de rede e acesso, e os problemas que você deve considerar ao escolher esses métodos.

Este documento é destinado a integradores de sistemas e desenvolvedores de aplicativos para auxiliar na determinação e elaboração de processos de aplicativos para novos clientes do AMS.

O que é integração de aplicativos?

A integração de aplicativos AMS se refere à implantação de recursos e aplicativos, conforme necessário, em sua infraestrutura AMS. Arquitetar aplicativos e infraestrutura na plataforma AMS é muito semelhante a fazer isso na plataforma nativa AWS. Seguir as melhores práticas de design de AWS aplicativos e infraestrutura, considerando os recursos fornecidos pelo AMS, produzirá aplicativos capazes e operacionais hospedados no ambiente AMS.

> **Note**
>
> - Leste dos EUA (Virgínia)
> - Oeste dos EUA (Norte da Califórnia)
> - Oeste dos EUA (Oregon)
> - Leste dos EUA (Ohio)
> - Canadá (Central)
> - América do Sul (São Paulo)
> - UE (Irlanda)
> - UE (Frankfurt)
> - UE (Londres)
> - Oeste da UE (Paris)
> - Ásia-Pacífico (Mumbai)
> - Ásia-Pacífico (Seul)
> - Ásia-Pacífico (Singapura)
> - Ásia-Pacífico (Sydney)

- Ásia-Pacífico (Tóquio)

Novas regiões são adicionadas com frequência. Para saber mais, consulte Regiões da AWS Zonas de disponibilidade.

O que fazemos, o que não fazemos

O AMS oferece uma abordagem padronizada para a implantação da infraestrutura da AWS e fornece o gerenciamento operacional contínuo necessário. Para obter uma descrição completa das funções, responsabilidades e serviços suportados, consulte Descrição do serviço.

> **Note**
>
> Para solicitar que o AMS forneça um serviço adicional da AWS, registre uma solicitação de serviço. Para obter mais informações, consulte Fazer solicitações de serviço.

- O que fazemos:

  Depois de concluir a integração, o ambiente AMS estará disponível para receber solicitações de mudança (RFCs), incidentes e solicitações de serviço. Sua interação com o serviço AMS gira em torno do ciclo de vida de uma pilha de aplicativos. As novas pilhas são ordenadas a partir de uma lista pré-configurada de modelos, lançadas em sub-redes específicas de nuvem privada virtual (VPC), modificadas durante sua vida operacional por meio de solicitações de mudança (RFCs) e monitoradas para eventos e incidentes 24 horas por dia, 7 dias por semana.

  As pilhas de aplicativos ativas são monitoradas e mantidas pelo AMS, incluindo a aplicação de patches, e não exigem nenhuma ação adicional durante a vida útil da pilha, a menos que seja necessária uma alteração ou que a pilha seja desativada. Os incidentes detectados pelo AMS que afetam a integridade e o funcionamento da pilha geram uma notificação e podem ou não precisar de sua ação para serem resolvidos ou verificados. Perguntas práticas e outras perguntas podem ser feitas enviando uma solicitação de serviço.

  Além disso, o AMS permite que você habilite serviços compatíveis da AWS que não são gerenciados pelo AMS. Para obter informações sobre serviços compatíveis com AWS-AMS, consulte Modo de provisionamento de autoatendimento.

- O que NÃO fazemos:

  Embora o AMS simplifique a implantação de aplicativos fornecendo várias opções manuais e automatizadas, você é responsável pelo desenvolvimento, teste, atualização e gerenciamento de seu aplicativo. O AMS fornece assistência para solução de problemas de infraestrutura que afetam os aplicativos, mas o AMS não pode acessar ou validar suas configurações de aplicativos.

Imagens de máquinas AMS Amazon (AMIs)

O AMS produz Amazon Machine Images (AMIs) atualizado todos os meses para sistemas operacionais compatíveis com o AMS. Além disso, o AMS também produz imagens de segurança aprimorada (AMIs) com base no benchmark CIS de nível 1 para um subconjunto dos sistemas operacionais suportados pelo AMS. Para descobrir quais sistemas operacionais têm uma imagem de segurança aprimorada disponível, consulte o Guia do usuário de segurança do AMS, que está disponível na página AWS Artifact -> Reports (encontre a opção Reports no painel de navegação esquerdo) filtrada para AWS Managed Services. Para acessar o AWS Artifact, entre em contato com seu CSDM para obter instruções ou acesse Getting Started with AWS Artifact.

Para receber alertas quando novos AMS AMIs forem lançados, você pode assinar um tópico de notificação do Amazon Simple Notification Service (Amazon SNS) chamado “AMS AMI”. Para obter detalhes, consulte Notificações da AMI do AMS com SNS.

A convenção de nomenclatura AMI do AMS é:`customer-ams-<operating system>-<release date> - <version>`. (por exemplo,`customer-ams-rhel6-2018.11-3`)

Use somente AMS AMIs que comecem com`customer`.

O AMS recomenda sempre usar a AMI mais recente. Você pode encontrar as mais recentes AMIs de qualquer uma das seguintes maneiras:

- Procurando no console do AMS, na AMIspágina.
- Visualizando o arquivo CSV da AMS AMI mais recente, disponível em seu CSDM ou por meio desse arquivo ZIP: conteúdo da AMS 11.2024 AMI e arquivo CSV em um ZIP.

  Para arquivos ZIP da AMI anteriores, consulte o Histórico do documento.
- Executando este `SKMS` comando AMS (é necessário o SDK AMS SKMS):

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

Conteúdo AMI do AMS adicionado à base AWS AMIs, por sistema operacional (SO)

- Linux AMIs:
  - AWS Ferramentas CLI
  - NTP
  - Agente de serviços do Trend Micro Endpoint Protection
  - Implantação de código
  - PBIS Enterprise//Beyond Trust AD Bridge

    > **Note**
    >
    > Em junho de 2022, BeyondTrust não oferece mais suporte ao PBIS Open. Você não pode usar o PBIS Open no AMIs suporte do AMS após junho de 2022. Se o AMS suportou sua AMI antes de junho de 2022, você pode continuar usando o PBIS Open a seu próprio critério.

  - SSM Agent
  - Atualização do Yum para patches críticos
  - Scripts personalizados e software de gerenciamento do AMS (controle de inicialização, junção do AD, monitoramento, segurança e registro)
- Servidor Windows AMIs:
  - Microsoft.NET Framework 4.5
  - PowerShell 5.1
  - AWS Ferramentas para Windows PowerShell
  - PowerShell Módulos AMS que controlam inicialização, junção de AD, monitoramento, segurança e registro
  - Agente de serviços do Trend Micro Endpoint Protection
  - SSM Agent
  - CloudWatch Agente
  - EC2Serviço de configuração (por meio do Windows Server 2012 R2)

- EC2Lançamento (Windows Server 2016 e Windows Server 2019)
- EC2LaunchV2 (Windows Server 2022 e versões posteriores)

Baseado em Linux: AMIs

- Amazon Linux 2023 (última versão secundária) (AMI mínima não suportada)
- Amazon Linux 2 (última versão secundária)
- Amazon Linux (2ARM64)
- Red Hat Enterprise 8 (última versão secundária)
- Red Hat Enterprise 9 (última versão secundária)
- Servidor SUSE Linux Enterprise 15 SP6
- Ubuntu Linux 20.04
- Ubuntu Linux 22.04
- Ubuntu Linux 24.04
- Amazon Linux: Para obter uma visão geral do produto, informações de preços, informações de uso e informações de suporte, consulte Amazon Linux 2.

  Para obter mais informações, consulte Amazon Linux 2 FAQs.
- SUSE Linux Enterprise Server para aplicativos SAP 15 SP6:
  - Execute as seguintes etapas uma vez por conta:
    1. Navegue até o AWS Marketplace.
    2. Pesquise o produto SUSE 15 SAP.
    3. Escolha Continue para assinar.
    4. Ecolha Aceitar termos.
  - Conclua as etapas a seguir sempre que precisar iniciar uma nova SP6 instância do SUSE Linux Enterprise Server for SAP Applications 15:
    1. Anote o ID da AMI da AMI assinada do SUSE Linux Enterprise Server for SAP Applications 15.
    2. Crie uma implantação | Componentes avançados da pilha | Pilha EC2 | Crie o tipo de alteração ct-14027q0sjyt1h RFC. *`InstanceAmiId`*Substitua pela ID da AWS Marketplace AMI na qual você se inscreveu.

Baseado em Windows: AMIs

Microsoft Windows Server (2016, 2019, 2022 e 2025), com base no Windows mais recente AMIs.

Para ver exemplos de criação AMIs, consulte Criar AMI.

Desembarcando o AMS AMIs:

O AMS não compartilha nada AMIs de você durante a desativação para evitar impacto em nenhuma de suas dependências. Se quiser remover o AMS AMIs da sua conta, você pode usar a `cancel-image-launch-permission` API para ocultar informações específicas AMIs. Por exemplo, você pode usar o script abaixo para ocultar todos os AMS AMIs que foram compartilhados com sua conta anteriormente:

```
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
```

Você deve ter a AWS CLI v2 instalada para que o script seja executado sem erros. Para ver as etapas de instalação da AWS CLI, consulte Instalando ou atualizando a versão mais recente da AWS CLI. Para obter detalhes sobre o `cancel-image-launch-permission` comando, consulte `cancel-image-launch-permission`.

Segurança aprimorada AMIs

O AMS fornece imagens de segurança aprimorada (AMIs) com base no benchmark CIS de nível 1 para um subconjunto dos sistemas operacionais suportados pelo AMS. Para descobrir quais sistemas operacionais têm uma imagem de segurança aprimorada disponível, consulte o Guia de segurança do cliente do AWS Managed Services (AMS). Para acessar esse guia, abra AWS Artifact, selecione Reports no painel de navegação esquerdo e, em seguida, filtre por AWS Managed Services. Para obter instruções sobre como acessar AWS Artifact, entre em contato com seu CSDM ou consulte Introdução AWS Artifact para obter mais informações.

Termos-chave do AMS

- AMS Advanced: Os serviços descritos na seção “Descrição do serviço” da documentação do AMS Advanced. Consulte a descrição do serviço.

- Contas avançadas do AMS: AWS contas que sempre atendem a todos os requisitos dos requisitos avançados de integração do AMS. Para obter informações sobre os benefícios do AMS Advanced, estudos de caso e para entrar em contato com um vendedor, consulte AWS Managed Services.
- Contas do AMS Accelerate: AWS contas que sempre atendem a todos os requisitos dos requisitos de integração do AMS Accelerate. Consulte Introdução ao AMS Accelerate.
- AWS Managed Services: AMS e/ou AMS Accelerate.
- Contas do AWS Managed Services: as contas do AMS e/ou contas do AMS Accelerate.
- Recomendação crítica: Uma recomendação emitida por AWS meio de uma solicitação de serviço informando que sua ação é necessária para se proteger contra possíveis riscos ou interrupções em seus recursos ou no. Serviços da AWS Se você decidir não seguir uma recomendação crítica até a data especificada, você é o único responsável por qualquer dano resultante de sua decisão.
- Configuração solicitada pelo cliente: qualquer software, serviço ou outras configurações que não estejam identificadas em:
  - Acelerar: configurações suportadas ou AMS Accelerate; Descrição do serviço.
  - AMS Advanced: configurações suportadas ou AMS Advanced; Descrição do serviço.
- Comunicação de incidentes: o AMS comunica um incidente a você ou você solicita um incidente ao AMS por meio de um incidente criado no Support Center for AMS Accelerate e no AMS Console for AMS. O console do AMS Accelerate fornece um resumo dos incidentes e solicitações de serviço no painel e links para o Support Center para obter detalhes.
- Ambiente gerenciado: as contas do AMS Advanced e/ou as contas do AMS Accelerate operadas pela AMS.

  Para o AMS Advanced, elas incluem contas de landing zone com várias contas (MALZ) e de landing zone com uma única conta (SALZ).
- Data de início do faturamento: No dia útil seguinte ao AWS recebimento das informações solicitadas no e-mail de integração do AWS Managed Services. O e-mail de integração do AWS Managed Services se refere ao e-mail enviado por AWS você para coletar as informações necessárias para ativar o AWS Managed Services em suas contas.

  Para contas inscritas posteriormente por você, a data de início do faturamento é o dia seguinte após o envio da Notificação de ativação do AWS Managed Services para a conta inscrita. Uma notificação de ativação do AWS Managed Services ocorre quando:

  1. Você concede acesso a uma AWS conta compatível e a entrega ao AWS Managed Services.
  2. O AWS Managed Services projeta e cria a conta do AWS Managed Services.

- Encerramento do serviço: você pode encerrar o AWS Managed Services para todas as contas do AWS Managed Services ou para uma conta específica do AWS Managed Services por qualquer motivo, fornecendo um aviso prévio de pelo AWS menos 30 dias por meio de uma solicitação de serviço. Na Data de Rescisão do Serviço, você pode:
  1. AWS entrega os controles de todas as contas do AWS Managed Services ou das contas especificadas do AWS Managed Services, conforme aplicável, para você, ou
  2. As partes removem as AWS Identity and Access Management funções que dão AWS acesso de todas as contas do AWS Managed Services ou das contas especificadas do AWS Managed Services, conforme aplicável.
- Data de rescisão do serviço: A data de rescisão do serviço é o último dia do mês civil após o final do período de notificação de rescisão obrigatório de 30 dias. Se o final do período de notificação de rescisão exigido cair após o vigésimo dia do mês civil, a data de rescisão do serviço será o último dia do mês civil seguinte. A seguir estão exemplos de cenários para datas de rescisão.
  - Se o aviso de rescisão for fornecido em 12 de abril, o aviso de 30 dias terminará em 12 de maio. A data de término do serviço é 31 de maio.
  - Se um aviso de rescisão for fornecido em 29 de abril, o aviso de 30 dias terminará em 29 de maio. A data de término do serviço é 30 de junho.
- Provisão do AWS Managed Services: AWS disponibiliza para você e você pode acessar e usar o AWS Managed Services para cada conta do AWS Managed Services a partir da data de início do serviço.
- Encerramento de contas específicas do AWS Managed Services: você pode encerrar o AWS Managed Services de uma conta específica do AWS Managed Services por qualquer motivo, enviando um AWS aviso por meio de uma solicitação de serviço (“Solicitação de encerramento da conta AMS”).

Termos de gerenciamento de incidentes:

- Evento: Uma mudança em seu ambiente AMS.
- Alerta: sempre que um evento de um suporte AWS service (Serviço da AWS) excede um limite e dispara um alarme, um alerta é criado e um aviso é enviado para sua lista de contatos. Além disso, um incidente é criado na sua lista de incidentes.
- Incidente: uma interrupção não planejada ou degradação do desempenho do seu ambiente de AMS ou do AWS Managed Services que resulta em um impacto conforme relatado pelo AWS Managed Services ou por você.

- Problema: Uma causa raiz subjacente compartilhada de um ou mais incidentes.
- Resolução de incidentes ou Resolver um incidente:
  - O AMS restaurou todos os serviços ou recursos indisponíveis do AMS relacionados a esse incidente para um estado disponível, ou
  - O AMS determinou que pilhas ou recursos indisponíveis não podem ser restaurados para um estado disponível, ou
  - O AMS iniciou uma restauração de infraestrutura autorizada por você.
- Tempo de resposta do incidente: a diferença de tempo entre o momento em que você cria um incidente e o momento em que o AMS fornece uma resposta inicial por meio do console, e-mail, centro de serviços ou telefone.
- Tempo de resolução do incidente: a diferença de tempo entre o momento em que você ou o AMS criam um incidente e o momento em que o incidente é resolvido.
- Prioridade do incidente: como os incidentes são priorizados pelo AMS ou por você como Baixa, Média ou Alta.
  - Baixo: um problema não crítico com seu serviço AMS.
  - Médio: um serviço da AWS em seu ambiente gerenciado está disponível, mas não está funcionando conforme o esperado (de acordo com a descrição do serviço aplicável).
  - Alto: (1) o console AMS ou um ou mais AMS APIs em seu ambiente gerenciado estão indisponíveis; ou (2) uma ou mais pilhas ou recursos do AMS em seu ambiente gerenciado estão indisponíveis e a indisponibilidade impede que seu aplicativo execute sua função.

  O AMS pode reclassificar os incidentes de acordo com as diretrizes acima.
- Restauração da infraestrutura: reimplantar pilhas existentes, com base em modelos de pilhas afetadas, e iniciar uma restauração de dados com base no último ponto de restauração conhecido, a menos que você especifique o contrário, quando a resolução de incidentes não for possível.

Termos de infraestrutura:

- Ambiente de produção gerenciado: uma conta de cliente na qual residem os aplicativos de produção do cliente.
- Ambiente gerenciado de não produção: uma conta de cliente que contém somente aplicativos que não são de produção, como aplicativos para desenvolvimento e teste.
- Pilha AMS: um grupo de um ou mais AWS recursos que são gerenciados pelo AMS como uma única unidade.

- Infraestrutura imutável: um modelo de manutenção de infraestrutura típico para grupos do Amazon EC2 Auto Scaling ASGs () em que componentes de infraestrutura atualizados (AWS na AMI) são substituídos em cada implantação, em vez de serem atualizados localmente. As vantagens da infraestrutura imutável são que todos os componentes permanecem em um estado síncrono, pois são sempre gerados a partir da mesma base. A imutabilidade é independente de qualquer ferramenta ou fluxo de trabalho para criar a AMI.
- Infraestrutura mutável: um modelo de manutenção de infraestrutura típico para pilhas que não são grupos do Amazon EC2 Auto Scaling e contêm uma única instância ou apenas algumas instâncias. Esse modelo representa mais de perto a implantação tradicional do sistema, baseada em hardware, em que um sistema é implantado no início de seu ciclo de vida e, em seguida, as atualizações são colocadas em camadas nesse sistema ao longo do tempo. Todas as atualizações do sistema são aplicadas às instâncias individualmente e podem causar tempo de inatividade do sistema (dependendo da configuração da pilha) devido à reinicialização do aplicativo ou do sistema.
- Grupos de segurança: firewalls virtuais para sua instância para controlar o tráfego de entrada e saída. Os grupos de segurança atuam no nível da instância e não no nível da sub-rede. Portanto, cada instância em uma sub-rede em sua VPC pode ter um conjunto diferente de grupos de segurança atribuídos a ela.
- Acordos de nível de serviço (SLAs): parte dos contratos do AMS com você que definem o nível de serviço esperado.
- SLA indisponível e indisponível:
  - Uma solicitação de API enviada por você que resulta em um erro.
  - Uma solicitação de console enviada por você que resulta em uma resposta HTTP 5xx (o servidor é incapaz de realizar a solicitação).
  - Qualquer uma das AWS service (Serviço da AWS) ofertas que constituem pilhas ou recursos em sua infraestrutura gerenciada pela AMS está em um estado de “interrupção do serviço”, conforme mostrado no Service Health Dashboard.
  - A indisponibilidade resultante direta ou indiretamente de uma exclusão do AMS não é considerada na determinação da elegibilidade para créditos de serviço. Os serviços são considerados disponíveis, a menos que atendam aos critérios de indisponibilidade.
- Objetivos de nível de serviço (SLOs): parte dos contratos do AMS com você que definem metas de serviço específicas para os serviços do AMS.

Termos de aplicação de patches:

- Patches obrigatórios: atualizações de segurança críticas para resolver problemas que podem comprometer o estado de segurança do seu ambiente ou conta. Uma “atualização crítica de segurança” é uma atualização de segurança classificada como “crítica” pelo fornecedor de um sistema operacional compatível com a AMS.
- Patches anunciados versus lançados: Os patches geralmente são anunciados e lançados de acordo com um cronograma. Os patches emergentes são anunciados quando a necessidade do patch é descoberta e, geralmente, logo depois, o patch é lançado.
- Complemento de patch: correção baseada em tags para instâncias do AMS que aproveita a funcionalidade AWS Systems Manager (SSM) para que você possa marcar instâncias e fazer com que essas instâncias sejam corrigidas usando uma linha de base e uma janela que você configura.
- Métodos de patch:
  - Aplicação de patches no local: correção que é feita alterando as instâncias existentes.
  - Patching de substituição de AMI: patch que é feito alterando o parâmetro de referência da AMI de uma configuração existente EC2 de lançamento de grupo do Auto Scaling.
- Provedor de patches (fornecedores de sistemas operacionais, terceiros): os patches são fornecidos pelo fornecedor ou pelo órgão regulador do aplicativo.
- Tipos de patch:
  - Atualização crítica de segurança (CSU): uma atualização de segurança classificada como “Crítica” pelo fornecedor de um sistema operacional compatível.
  - Atualização importante (IU): uma atualização de segurança classificada como “Importante” ou uma atualização não relacionada à segurança classificada como “Crítica” pelo fornecedor de um sistema operacional compatível.
  - Outra atualização (OU): uma atualização do fornecedor de um sistema operacional compatível que não é uma CSU ou uma IU.
- Patches suportados: O AMS oferece suporte a patches em nível de sistema operacional. As atualizações são lançadas pelo fornecedor para corrigir vulnerabilidades de segurança ou outros bugs ou para melhorar o desempenho. Para obter uma lista das opções atualmente suportadas OSs, consulte Support Configurations.

Termos de segurança:

- Detective Controls: uma biblioteca de monitores criados ou habilitados pela AMS que fornecem supervisão contínua dos ambientes e cargas de trabalho gerenciados pelo cliente para

configurações que não se alinham aos controles de segurança, operacionais ou do cliente e agem notificando os proprietários, modificando ou encerrando recursos de forma proativa.

Termos da solicitação de serviço:

- Solicitação de serviço: uma solicitação feita por você para uma ação que você deseja que o AMS execute em seu nome.
- Notificação de alerta: um aviso publicado pelo AMS em sua página de lista de solicitações de serviço quando um alerta do AMS é acionado. O contato configurado para sua conta também é notificado pelo método configurado (por exemplo, e-mail). Se você tiver tags de contato em suas instâncias/recursos e tiver fornecido consentimento ao seu gerente de prestação de serviços em nuvem (CSDM) para notificações baseadas em tags, as informações de contato (valor-chave) na tag também serão notificadas para alertas automatizados do AMS.
- Notificação de serviço: um aviso do AMS publicado na sua página de lista de solicitações de serviço.

Termos diversos:

- Interface de serviços gerenciados da AWS: Para AMS: o console avançado do AWS Managed Services, a API do AMS CM e a Suporte API. Para o AMS Accelerate: o Suporte console e a Suporte API.
- Satisfação do cliente (CSAT): O AMS CSAT é informado por meio de análises profundas, incluindo classificações de correspondência de casos em cada caso ou correspondência quando fornecida, pesquisas trimestrais e assim por diante.
- DevOps: DevOps é uma metodologia de desenvolvimento que defende fortemente a automação e o monitoramento em todas as etapas. DevOps visa ciclos de desenvolvimento mais curtos, maior frequência de implantação e lançamentos mais confiáveis, reunindo as funções tradicionalmente separadas de desenvolvimento e operações em uma base de automação. Quando os desenvolvedores podem gerenciar as operações e as operações informam o desenvolvimento, questões e problemas são descobertos e resolvidos mais rapidamente, e os objetivos de negócios são alcançados com mais facilidade.
- ITIL: A Biblioteca de Infraestrutura de Tecnologia da Informação (chamada ITIL) é uma estrutura de ITSM projetada para padronizar o ciclo de vida dos serviços de TI. O ITIL é organizado em cinco estágios que abrangem o ciclo de vida do serviço de TI: estratégia do serviço, design do serviço, transição do serviço, operação do serviço e melhoria do serviço.

- Gerenciamento de serviços de TI (ITSM): um conjunto de práticas que alinha os serviços de TI às necessidades da sua empresa.
- Serviços de monitoramento gerenciados (MMS): O AMS opera seu próprio sistema de monitoramento, o Managed Monitoring Service (MMS), que consome eventos de AWS saúde e agrega dados da CloudWatch Amazon e dados de Serviços da AWS outros, notificando operadores do AMS (on-line 24 horas por dia, 7 dias por semana) sobre quaisquer alarmes criados por meio de um tópico do Amazon Simple Notification Service (Amazon SNS).
- Namespace: ao criar políticas do IAM ou trabalhar com Amazon Resource Names (ARNs), você identifica uma AWS service (Serviço da AWS) usando um namespace. Os namespaces são usados ao identificar ações e recursos.

Qual é o meu modelo operacional?

Como cliente do AMS, sua organização decidiu separar as operações de aplicativos e infraestrutura e usar o AMS para operações de infraestrutura. O AMS trabalhará com sua equipe de design e desenvolvimento de aplicativos junto com sua equipe de design de infraestrutura para garantir que suas operações de infraestrutura funcionem sem problemas. O gráfico a seguir ilustra esse conceito:

O AMS assume a responsabilidade por suas operações de AWS infraestrutura, enquanto suas equipes são responsáveis pelas operações de seus aplicativos. Como equipes de design de aplicativos e infraestrutura, você deve entender quem operará o aplicativo depois que ele for implantado na produção na infraestrutura do AMS. Este guia aborda abordagens comuns para o projeto de infraestrutura no que se refere à implantação e manutenção de aplicativos.

Gerenciamento de serviços no AWS Managed Services

Tópicos

- Governança de contas no AWS Managed Services
- Início do serviço no AWS Managed Services
- Gestão do relacionamento com o cliente (CRM)
- Otimização de custos no AWS Managed Services
- Horas de serviço no AWS Managed Services
- Obter ajuda no AWS Managed Services

Como o serviço AMS funciona para você.

Governança de contas no AWS Managed Services

Esta seção aborda a governança de contas do AMS.

Você é designado como gerente de prestação de serviços em nuvem (CSDM) que fornece assistência consultiva em todo o AMS e tem uma compreensão detalhada do seu caso de uso e da arquitetura de tecnologia para o ambiente gerenciado. CSDMs trabalhe com gerentes de contas, gerentes técnicos de contas, arquitetos de nuvem do AWS Managed Services (CAs) e arquitetos de soluções da AWS (SAs), conforme aplicável, para ajudar a lançar novos projetos e oferecer recomendações de melhores práticas em todos os processos operacionais e de desenvolvimento de software. O CSDM é o principal ponto de contato do AMS. As principais responsabilidades do seu CSDM são:

- Organize e conduza reuniões mensais de análise de serviços com os clientes.
- Forneça detalhes sobre segurança, atualizações de software para o ambiente e oportunidades de otimização.
- Defenda seus requisitos, incluindo solicitações de recursos para o AMS.
- Responda e resolva solicitações de faturamento e relatórios de serviços.
- Forneça insights para recomendações financeiras e de otimização de capacidade.

Início do serviço no AWS Managed Services

Início do serviço: A data de início do serviço para uma conta do AWS Managed Services é o primeiro dia do primeiro mês civil após o qual a AWS notifica você de que as atividades definidas nos requisitos de integração dessa conta do AWS Managed Services foram concluídas; desde que, se a AWS fizer essa notificação após o vigésimo dia de um mês civil, a data de início do serviço será o primeiro dia do segundo mês civil após a data de tal notificação.

Início do serviço

- R significa parte responsável que faz o trabalho para realizar a tarefa.
- I significa informado; uma parte que é informada sobre o progresso, geralmente apenas após a conclusão da tarefa ou do resultado final.

Início do serviço

N.º da etapa	Título da etapa	Descrição	Cliente	AMS
1.	Entrega da conta AWS do cliente	O cliente cria uma nova conta da AWS e a entrega ao AWS Managed Services	R	eu
2.	Conta do AWS Managed Services — design	Finalize o design da conta do AWS Managed Services	eu	R
3.	Conta do AWS Managed Services — criar	Uma conta do AWS Managed Services é criada de acordo com o design na Etapa 2	eu	R

Gestão do relacionamento com o cliente (CRM)

O AWS Managed Services (AMS) fornece um processo de gerenciamento de relacionamento com o cliente (CRM) para garantir que um relacionamento bem definido seja estabelecido e mantido com você. A base desse relacionamento é baseada na visão da AMS sobre seus requisitos de negócios. O processo de CRM facilita a compreensão precisa e abrangente de:

- Suas necessidades de negócios e como atender a essas necessidades
- Suas capacidades e restrições
- AMS e suas diferentes responsabilidades e obrigações

O processo de CRM permite que a AMS use métodos consistentes para fornecer serviços a você e fornecer governança para seu relacionamento com a AMS. O processo de CRM inclui:

- Identificando suas principais partes interessadas
- Estabelecendo uma equipe de governança
- Conduzindo e documentando reuniões de revisão de serviços com você
- Fornecer um procedimento formal de reclamação de serviço com um procedimento de escalonamento
- Implementando e monitorando seu processo de satisfação e feedback
- Gerenciando seu contrato

Processo de CRM

O processo de CRM inclui as seguintes atividades:

- Identificar e entender seus processos e necessidades de negócios. Seu contrato com a AMS identifica suas partes interessadas.
- Definindo os serviços a serem fornecidos para atender às suas necessidades e exigências.
- Reunir-se com você nas reuniões de revisão de serviços para discutir quaisquer alterações no escopo do serviço AMS, no SLA, no contrato e nas necessidades de seus negócios. Reuniões provisórias podem ser realizadas com você para discutir desempenho, conquistas, problemas e planos de ação.
- Monitorando sua satisfação usando nossa pesquisa de satisfação do cliente e o feedback fornecido nas reuniões.
- Relatórios de desempenho em relatórios mensais de desempenho medidos internamente.
- Analisar o serviço com você para determinar oportunidades de melhorias. Isso inclui comunicação frequente com você sobre o nível e a qualidade do serviço de AMS fornecido.

Reuniões de CRM

Os gerentes de prestação de serviços em nuvem (CSDMs) da AMS realizam reuniões com você regularmente para discutir faixas de serviços (operações, segurança e inovações de produtos) e executivas (relatórios de SLA, medidas de satisfação e mudanças nas necessidades de seus negócios).

Reunião	Finalidade	Modo	Participantes
Revisão semanal do status (opcional)	Problemas ou incidentes pendentes , correções, eventos de segurança, registros de problemas Tendência operacional de 12 semanas (+/- 6) Preocupações do operador do aplicativo Programação de fim de	Cliente no local location/ Telecom/Chime	AMS: CSDM e arquiteto de nuvem (CA) Membros da equipe designados pelo cliente (ex: Cloud/Inf rastructure, Application Support, equipes de arquitetura etc.)
Análise mensal de negócios	Analise o desempenho do nível de serviço (relatórios, análises e tendências) Análise financeira Roteiro do produto GATO	Cliente no local location/ Telecom/Chime	AMS: CSDM, arquiteto de nuvem (CA), equipe de contas do AMS, gerente técnico de produto do AMS (TPM) (opcional), gerente do AMS OPS (opcional)

Reunião	Finalidade	Modo	Participantes
			Você: representante do operador do aplicativo
Análise trimestral de negócios	Desempenho e tendências do scorecard e do acordo de nível de serviço (SLA) (6 meses) Próximos planos/migrações de 3/6/9/12 meses Risco e mitigação de riscos Principais iniciativas de melhoria Itens do roteiro do produto Oportunidades alinhadas à direção futura Finanças Iniciativas de redução de custos Otimização de negócios	Localização do cliente no local	AMS: CSDM, arquiteto de nuvem, equipe de contas do AMS, diretor de serviços do AMS, gerente de operações do AMS Você: representante do operador do aplicativo, representante do serviço, diretor de serviços

Organizações de reuniões de CRM

O CSDM do AMS é responsável por documentar a reunião, incluindo:

- Criação da agenda, incluindo itens de ação, problemas e lista de participantes.
- Criar a lista de itens de ação revisados em cada reunião para garantir que os itens sejam concluídos e resolvidos dentro do cronograma.
- Distribuir as atas da reunião e a lista de itens de ação aos participantes da reunião por e-mail dentro de um dia útil após a reunião.
- Armazenar as atas da reunião no repositório de documentos apropriado.

Na ausência do CSDM, o representante da AMS que lidera a reunião cria e distribui atas.

> **Note**
>
> Seu CSDM trabalha com você para estabelecer a governança da sua conta.

Relatórios mensais de CRM

Seu CSDM do AMS prepara e envia apresentações mensais sobre o desempenho do serviço. As apresentações incluem informações sobre o seguinte:

- Data do relatório
- Resumo e insights:
  - Principais chamadas: contagem total e ativa de pilhas, status de patches de pilha, status de integração da conta (somente durante a integração), resumos de problemas específicos do cliente
  - Desempenho: estatísticas sobre resolução de incidentes, alertas, correções, solicitações de mudança (RFCs), solicitações de serviço e disponibilidade do console e da API
  - Problemas, desafios, preocupações e riscos: status de problemas específicos do cliente
  - Itens futuros: planos de integração ou resolução de incidentes específicos para o cliente
- Recursos gerenciados: gráficos e diagramas circulares de pilhas
- Métricas do AMS: métricas de monitoramento e eventos, métricas de incidentes, métricas de adesão ao AMS SLA, métricas de solicitação de serviço, métricas de gerenciamento de mudanças, métricas de armazenamento, métricas de continuidade, métricas do Trusted Advisor e resumos de custos (apresentados de várias maneiras). Solicitações de recursos. Informações de contato.

> **Note**
>
> Além das informações descritas, seu CSDM também o informa sobre qualquer alteração material no escopo ou nos termos, incluindo o uso de subcontratados pela AMS para atividades operacionais.
> O AMS gera relatórios sobre patches e backup que seu CSDM inclui em seu relatório mensal. Como parte do sistema de geração de relatórios, o AMS adiciona alguma infraestrutura à sua conta que não está acessível para você:

- Um bucket S3, com os dados brutos relatados
- Uma instância do Athena, com definições de consulta para consultar os dados
- Um Glue Crawler para ler os dados brutos do bucket S3

Otimização de custos no AWS Managed Services

O AWS Managed Services fornece relatórios detalhados de utilização de custos e economia todos os meses para você durante suas análises mensais de negócios (MBRs).

O AMS segue um conjunto padrão de processos e mecanismos para identificar formas de redução de custos em suas contas gerenciadas e ajudá-lo a planejar e implementar as mudanças para otimizar seus gastos com a AWS.

> **Note**
>
> A AMS está desenvolvendo um vídeo para ajudar na otimização de custos. A primeira etapa é fornecer a você um PDF e uma planilha do Excel com as melhores práticas de otimização de custos. Para acessar esses recursos, abra o arquivo ZIP do Guia rápido para otimização de custos.

Estrutura de otimização de custos

O AMS segue uma abordagem de três estágios com você para otimizar seus custos da AWS:

1. Identifique caminhos de otimização de custos em seu ambiente gerenciado
2. Apresente um plano de otimização de custos para você
3. Auxiliar na otimização de custos de forma mensurável

Identifique caminhos de otimização de custos no ambiente gerenciado

O AMS utiliza ferramentas AWS nativas, como o Cost Explorer e o Trusted Advisor, enquanto aproveita mais de 20 padrões de economia de custos em otimização de arquitetura, otimizações AWS focadas em EC2 instâncias e contas para criar recomendações personalizadas de economia de custos para você.

Algumas das recomendações de otimização incluem o seguinte.

Recomendações de otimização arquitetônica:

- Uso ideal da classe de armazenamento S3: O Amazon S3 oferece uma variedade de classes de armazenamento para atender a vários requisitos de carga de trabalho com base no acesso, resiliência e custo dos dados. O S3 Intelligent-Tiering e a análise da classe de armazenamento S3 com base nas necessidades da carga de trabalho permitem que você gerencie os custos do S3 com eficiência.
- Usando arquiteturas de cache: o uso de instâncias de cache, quando aplicável, pode ajudá-lo a substituir algumas instâncias de banco de dados e, ao mesmo tempo, atender aos requisitos de IOPS.
- Economia de upgrade do EBS: migrar seus volumes do EBS de gp2 para gp3 proporciona uma economia de até 20% e você pode aproveitar o desempenho previsível da linha de base de 3.000 IOPS e 125 MiB/s, independentemente do tamanho do volume.
- Usando elasticidade: os recursos de auto-scaling fornecidos permitem a utilização eficaz dos recursos AWS e caminhos para a otimização de custos. Revisar e atualizar as políticas de escalabilidade de instâncias regularmente com base na necessidade proporciona ainda mais economia de custos.

EC2 recomendações focadas na instância

- Dimensionamento correto da instância: recomendações focadas no dimensionamento das instâncias e nas configurações ideais com base no uso. As recomendações também incluem a utilização do recurso Amazon EC2 Auto Scaling e a EC2 substituição de instâncias, quando aplicável AWS Lambda , por conteúdo web estático no Amazon S3, etc.
- Agendamento de instâncias: usar o AMS Resource Scheduler para iniciar e interromper automaticamente instâncias com base em um cronograma ajuda a conter custos, especialmente para instâncias que não são de produção e que não são utilizadas fora do horário comercial.
- Assinando planos de poupança: O plano de economia é a maneira mais fácil de economizar no AWS uso. Os EC2 Instance Savings Plans oferecem até 72% de economia em comparação com os preços sob demanda no uso de suas EC2 instâncias da Amazon. Os Amazon SageMaker AI Savings Plans oferecem até 64% de economia no uso dos serviços de SageMaker IA da Amazon. O AMS fornece recomendações apropriadas sobre planos de economia com base no uso AWS de seus recursos.

- Orientação de uso e consumo de instâncias reservadas (RI): as Instâncias EC2 Reservadas (RI) da Amazon oferecem um desconto significativo (até 75%) em comparação com os preços sob demanda e fornecem uma reserva de capacidade quando usadas em uma zona de disponibilidade específica.
- Uso de instâncias spot: cargas de trabalho tolerantes a falhas podem utilizar instâncias spot e reduzir os preços em até 90%.
- Encerramento de instância ociosa: identificar e relatar instâncias que estão ociosas ou com baixa utilização que podem ser encerradas.

Recomendações focadas na conta

- Limpeza da conta: no nível da conta, o AMS também identifica volumes do EBS não utilizados, CloudTrail trilhas duplicadas, contas vazias com recursos não utilizados e assim por diante, além de fornecer recomendações para limpeza.
- Recomendações de SLA: Além disso, o AMS revisa regularmente suas contas Plus e Premium e recomenda escolher o nível de SLA certo para as contas.
- Otimização da automação do AMS: O AMS otimiza continuamente a automação e a infraestrutura do AMS usada para fornecer serviços do AMS.

Apresente aos clientes e auxilie no planejamento

A AMS conduz análises comerciais mensais (MBRs) com as principais partes interessadas do cliente e apresenta as vias, mecanismos e recomendações de redução de custos identificados, juntamente com possíveis economias de custos. Além disso, trabalhamos com você para planejar as mudanças necessárias.

Auxiliar na implementação de recomendações e medir o impacto nos custos

O AMS auxilia na obtenção e medição dos impactos nos custos e nas mudanças de otimização.

Você avalia o impacto do aplicativo, o risco e os critérios de sucesso das alterações recomendadas e gera as solicitações apropriadas de mudança (RFCs) por meio do console do AMS. O AMS colabora com você e implementa as mudanças relacionadas à otimização de custos em suas contas gerenciadas. O AMS mede o impacto nos custos e inclui as economias realizadas nas análises mensais de negócios (MBRs).

Matriz de responsabilidade de otimização de custos

Responsabilidades na otimização de custos do AMS.

Otimização de custos RACI

Atividades	Cliente	AMS
Compilando recomendações de redução de custos e preparando o relatório	eu	R
Apresentando relatório de economia de custos	C	R
Planejando mudanças associadas à economia de custos	R	C
Avaliando o impacto e o	R	C

Atividades	Cliente	AMS
risco da mudança		
Levantando RFCs para implementar as mudanças	R	C
Analisando RFCs e implementando as mudanças	C	R
Testando o aplicativo e validando a implementação da mudança	R	C

Atividades	Cliente	AMS
Medir o impacto nos custos após a mudança e apresentá-lo ao cliente	eu	R

Horas de serviço no AWS Managed Services

Recurso	AMS Avançado
	Nível Premium
Solicitação de serviço	24/7
Gerenciamento de incidentes (P2-P3)	24/7
Backup e recuperação	24/7
Gerenciamento de patches	24/7
Monitoramento e alertas	24/7
Solicitação automática de alteração (RFC)	24/7
Solicitação de alteração não automatizada (RFC)	24/7
Gerente de entrega de serviços em nuvem (CSDM)	Segunda a sexta-feira: das 08:00 às 17:00, horário comercial local

Obter ajuda no AWS Managed Services

O AMS oferece suporte com gerenciamento de incidentes, gerenciamento de solicitações de serviço e gerenciamento de alterações 24 horas por dia, 7 dias por semana, 365 dias por ano (de acordo com o Acordo de Nível de Serviço do AMS aplicado à conta).

Para relatar um problema de desempenho do serviço da AWS ou do AMS que afeta seu ambiente gerenciado, use o console do AMS e envie um relatório de incidentes. Para obter detalhes, consulte Relatar um incidente. Para obter informações gerais sobre o gerenciamento de incidentes do AMS, consulte Resposta a incidentes.

Para solicitar informações ou conselhos, ou para solicitar serviços adicionais do AMS, use o console do AMS e envie uma solicitação de serviço. Para obter detalhes, consulte Criando uma solicitação de serviço. Para obter informações gerais sobre solicitações de serviço do AMS, consulte Gerenciamento de solicitações de serviço.

Desenvolvimento de aplicações

Processos e práticas de desenvolvimento de aplicativos que permitem o design e a implantação eficazes de aplicativos em um ambiente AWS Managed Services (AMS). O AMS orienta você no seguinte processo de alto nível:

1. Imagine e arquitete um aplicativo a ser desenvolvido ou integrado ao seu ambiente gerenciado pelo AMS. Algumas considerações:

    a. Como você implantará seu aplicativo? Com a automação usando uma ferramenta de implantação, como o Ansible, ou manualmente, carregando diretamente os arquivos necessários?

    b. Como você atualizará seu aplicativo? Com uma abordagem mutável atualizando cada instância separadamente ou com uma abordagem imutável atualizando cada instância com uma única AMI atualizada em um grupo de Auto Scaling?

2. Planeje e arquitete a infraestrutura que será usada para hospedar o aplicativo usando bibliotecas de AWS arquitetura, orientação do AWS “Well-Architected” e AMS e outros especialistas no assunto de arquitetura de nuvem. As seções a seguir deste guia fornecem informações que podem ajudar com isso.
3. Selecione uma abordagem de implantação de infraestrutura:

    a. Full Stack: todos os componentes da infraestrutura são implantados ao mesmo tempo, juntos.

    b. Nível e tempo: as implantações de infraestrutura são implantadas separadamente e, depois, vinculadas às modificações do grupo de segurança. Esse tipo de implantação também é obtido por meio de uma configuração serial de componentes de pilha que se baseiam uns nos outros; por exemplo, especificando o balanceador de carga que você criou anteriormente ao criar um grupo de Auto Scaling.

    c. Quais ambientes, como Dev, Staging e Prod, você empregará?

4. Escolha os tipos de alteração (CTs) do AMS que provisionarão as pilhas ou camadas necessárias e prepararão as solicitações de alteração () RFCs necessárias.
5. Envie o RFCs para acionar a implantação da infraestrutura no ambiente apropriado.
6. Implante o aplicativo usando a abordagem de implantação de aplicativos selecionada.
7. Retrabalhe a infraestrutura e os aplicativos conforme necessário.

8. Implante a infraestrutura e os aplicativos em ambientes subsequentes apropriados, supondo que sua primeira implantação seja em um ambiente que não seja de produção.
9. A manutenção contínua é feita pelo AMS que opera a infraestrutura subjacente e por suas equipes de operações que operam as infraestruturas do (s) aplicativo (s).
10. Para descomissionar um aplicativo, encerre a infraestrutura do AMS para ele.

Ser bem arquitetado

Na, AWS acreditamos que sistemas bem arquitetados aumentam muito a probabilidade de sucesso nos negócios. O AWS Architecture Center fornece orientação especializada em arquitetura no Nuvem AWS.

Recomendamos os seguintes artigos e whitepapers para ajudá-lo a entender os prós e os contras das decisões que você deve tomar ao criar sistemas. AWS

Você é Well-Architected? : Apresenta o AWS Well-Architected Framework, baseado em seis pilares:

- Excelência operacional: o pilar da excelência operacional se concentra na execução e monitoramento de sistemas para agregar valor aos negócios e melhorar continuamente os processos e procedimentos. Os principais tópicos incluem gerenciar e automatizar mudanças, responder a eventos e definir padrões para gerenciar com sucesso as operações diárias.
- Segurança: O pilar de segurança se concentra na proteção de informações e sistemas. Os principais tópicos incluem confidencialidade e integridade dos dados, identificação e gerenciamento de quem pode fazer o quê com o gerenciamento de permissões, proteção de sistemas e estabelecimento de controles para detectar eventos de segurança.
- Confiabilidade: o pilar da confiabilidade se concentra na capacidade de prevenir e se recuperar rapidamente de falhas para atender às demandas dos negócios e dos clientes. Os principais tópicos incluem elementos fundamentais sobre configuração, requisitos de vários projetos, planejamento de recuperação e como lidamos com as mudanças.
- Eficiência de desempenho: o pilar de eficiência de desempenho se concentra no uso eficiente dos recursos de TI e computação. Os principais tópicos incluem a seleção dos tipos e tamanhos certos de recursos com base nos requisitos da workload, o monitoramento da performance e a tomada de decisões informadas para manter a eficiência à medida que as necessidades dos negócios evoluem.
- Otimização de custos: o pilar de otimização de custos se concentra em evitar custos desnecessários. Os principais tópicos incluem entender e controlar onde o dinheiro está sendo

gasto, selecionar o número mais adequado e correto de tipos de recursos, analisar os gastos ao longo do tempo e escalar para atender às necessidades comerciais sem gastar demais.

- Sustentabilidade: o pilar de sustentabilidade se concentra na capacidade de melhorar continuamente os impactos da sustentabilidade, reduzindo o consumo de energia e aumentando a eficiência em todos os componentes de uma carga de trabalho, maximizando os benefícios dos recursos provisionados e minimizando o total de recursos necessários.

AWS Well-Architected Framework: descreve AWS como permite que os clientes avaliem e melhorem suas arquiteturas baseadas em nuvem e entendam melhor o impacto comercial de suas decisões de design. Ele aborda os princípios gerais de design, bem como as melhores práticas e orientações específicas em seis áreas conceituais que são AWS definidas como os pilares do Well-Architected Framework.

Responsabilidades da camada de aplicativo versus responsabilidades da camada de infraestrutura no AMS

Ao usar o AMS, sua infraestrutura e tudo o que ela precisa para manutenção e crescimento são mantidos pelo AMS. No entanto, tudo o que você precisa para line-of-business aplicativos ou aplicativos de produtos é desenvolvido, implantado e mantido por você.

Com a ajuda de ferramentas de implantação de aplicativos, como CodeDeploy e, ou Chef CloudFormation, Puppet, Ansible ou Saltstack, a implantação de seu aplicativo em sua infraestrutura gerenciada pelo AMS pode ser totalmente automatizada.

Para obter detalhes sobre o que o AMS faz e o que não faz, consulteO que fazemos, o que não fazemos.

Mutabilidade da EC2 instância Amazon no AMS

Você e o AMS podem manter as instâncias do Amazon Elastic Compute Cloud (Amazon EC2) em sua infraestrutura de duas maneiras:

- Imutável: esse modelo usa Amazon Machine Images (AMIs) criado (criado) com os recursos necessários. Ao implantar uma atualização, as instâncias existentes são desmontadas e completamente substituídas por novas criadas a partir de uma AMI atualizada. Para minimizar o tempo de inatividade, esse processo contínuo deixa algumas instâncias inatualizadas e

acessíveis, enquanto outras estão sendo atualizadas até que, eventualmente, a nova alteração seja completamente implantada.

- Mutável: nesse modelo, a infraestrutura é atualizada com o novo código sendo implantado nos sistemas existentes na nuvem. Esse modelo é uma mistura de envio manual de atualizações e uso infrastructure-as-code para implantar atualizações e não depende de atualizações novas AMIs.

Esses modelos de manutenção são discutidos com mais detalhes nas seções posteriores deste guia.

Usando o AWS Secrets Manager com recursos do AMS

Há muitos casos em que você pode precisar compartilhar segredos com o AMS, por exemplo:

- Redefinição de senha mestra para instância do RDS
- Certificados para balanceadores de carga
- Obtenção de credenciais duradouras do AMS para usuários do IAM

A maneira mais segura de compartilhar informações confidenciais com o AMS é por meio do AWS Secrets Manager; siga estas etapas:

1. Faça login no AWS console usando seu acesso federado e a CustomerReadOnly função de landing zone de conta única (SALZ); use qualquer uma dessas funções,, AWSManaged ServicesSecurityOpsRole AWSManagedServicesAdminRole, e AWSManaged ServicesChangeManagementRole para a landing zone de várias contas (MALZ).
2. Navegue até o console do gerenciador de segredos da AWS e clique em Armazenar um novo segredo.
3. Selecione “Outro tipo de segredos”.
4. Insira o valor secreto como texto sem formatação e clique em Avançar.
5. Insira o nome secreto e a descrição. O nome deve sempre começar com "customer-shared/*”. Por exemplo, "cliente-shared/license-2018”. Quando terminar, continue clicando em Avançar.
6. Use a criptografia KMS padrão.
7. Deixe a rotação automática desativada e clique em Avançar.
8. Revise e clique em Armazenar para salvar o segredo.

9. Responda a nós em uma solicitação de serviço do AMS com o nome secreto e o ARN, para que possamos identificar e recuperar o segredo. Para obter informações sobre a criação de solicitações de serviço, consulte Exemplos de solicitações de serviço.

Implantação de aplicativos no AMS

Durante a integração, o AWS Managed Services (AMS) trabalha com você para determinar a infraestrutura de que você precisa.

A infraestrutura básica inclui uma nuvem privada AWS virtual (VPC), segurança de comunicação por meio de uma confiança na floresta ADFS, sub-redes básicas (DMZ, serviços compartilhados e privadas) espelhadas em duas zonas de disponibilidade e configuradas com um NAT gerenciado, bastiões, balanceadores de carga públicos (DX) e a segurança necessária. Direct Connect Os recursos do seu aplicativo serão implantados em sua sub-rede privada ou de aplicativos de clientes. Você pode aprender mais sobre uma arquitetura típica de AMS no Guia do usuário do AWS Managed Services.

A infraestrutura que você implanta depois de concluir o básico deve incluir todos os componentes de seus aplicativos e do desenvolvimento de aplicativos.

Capacidades de implantação de aplicativos no AMS

Algumas das maneiras pelas quais você pode implantar aplicativos no AMS. Veja os detalhes de cada método a seguir.

Exemplos de recursos de implantação de aplicativos

Nome de método	Implantação de infraestrutura	AMI ou elemento (s) chave	Instalação do aplicativo
Aplicativos mutáveis, AMS AMI			
Implantação manual de aplicativos	Pilha completa CT ou Tier and Tie CTs	AMI fornecida pela AMS	Envie o CT de gerenciamento de acesso, instale o aplicativo manualmente.
UserData implantação de aplicativos com agente de aplicativo			Use o provisionamento de CT com UserData scripts que instalam um

Nome de método	Implantação de infraestrutura	AMI ou elemento (s) chave	Instalação do aplicativo
(ou seja, Chef, Puppet etc.)			agente de aplicativo e instalam script/agent o aplicativo.
UserData implantação de aplicativos sem agente (ou seja, Ansible, Salt SSH etc.)			Envie o CT de gerenciamento de acesso, instale o agente do aplicativo. Implante aplicativos com ferramentas de implantação de aplicativos.
Aplicativos mutáveis, AMI personalizada			
Implantação personalizada de aplicativos AMI (não ASG)	Pilha completa CT ou Tier and Tie CTs	AMI personalizada. AMI do AMS -> personalize com o agente de ferramentas de implantação do aplicativo -> criar EC2 instância (CT) -> criar AMI (CT).	Ferramentas de implantação de aplicativos (ou seja, Chef), aproveitando agentes, implanta o aplicativo.
Implantação do aplicativo AWS Database Migration Service (DMS)	Sincronização do AWS DMS com a pilha de banco de dados relacional AMS existente.	AMI personalizada	O cliente ou parceiro emprega o AWS Database Migration Service; o AMS verifica os componentes do AMS no lançamento

Nome de método	Implantação de infraestrutura	AMI ou elemento (s) chave	Instalação do aplicativo
Implantação do aplicativo Workload Ingest	Workload Ingest CT migrado pelo parceiro instance/AMI e iniciado pelo cliente.		O parceiro migra a instância, cria AMI na VPC gerenciada pelo AMS do cliente; o cliente usa o Workload Ingest CT para lançar a pilha no AMS. Para obter detalhes, consulte Ingestão de workload do AMS (WIGS).
Aplicativos imutáveis			
Implantação personalizada de aplicativos AMI (ASG)	Pilha completa CT ou Tier and Tie CTs	AMS AMI -> personalizar -> criar EC2 instância (CT) -> criar AMI (CT) -> criar grupo Auto Scaling.	O Auto Scaling implanta o aplicativo com a AMI personalizada Para obter detalhes, consulte Implantações de aplicativos Tier and Time no AMS.
Aplicativos mutáveis ou imutáveis			

Nome de método	Implantação de infraestrutura	AMI ou elemento (s) chave	Instalação do aplicativo
Implantação CloudFormation do aplicativo de modelo personalizado	CloudFormation modelo	CloudFormation Modelo da AWS -> customize/ prepare para AMS -> Implantação | Ingestão | Pilha a partir do CloudFormation modelo | Criar (ct-36cn2avfrrj9v).	O AMS implanta seu aplicativo em sua conta usando seu CloudFormation modelo personalizado e valida a implantação do aplicativo. Para obter detalhes, consulte CloudFormation Ingestão de AMS.
Importação de banco de dados SQL	Operações AMS (Outros | Outros CT)	Banco de dados SQL local -> arquivo.bak -> Banco de dados SQL AMS RDS -> Gerenciamento | Outros | Outros | Outros | Crie (ct-1e1xtak34nx76) para a importação.	O AMS importa seu banco de dados local para seu banco de dados RDS gerenciado pelo AMS. Para obter detalhes, consulte Importação de banco de dados (DB) para o AMS RDS para Microsoft SQL Server.

Nome de método	Implantação de infraestrutura	AMI ou elemento (s) chave	Instalação do aplicativo
Database Migration Service (DMS)	Operações AMS (múltiplas CTs)	Banco de dados local -> instância de replicação do DMS -> grupo de sub-rede de replicação do DMS -> endpoint de destino do DMS -> endpoint de origem do DMS -> tarefa de replicação do DMS.	O AMS importa seu banco de dados local para seu banco de dados S3 gerenciado pelo AMS ou RDS de destino. Para obter detalhes, consulte AWS Database Migration Service (AWS DMS).
CodeDeploy implantação de aplicativos	CodeDeploy	Aplicativo -> CodeDeploy aplicativo -> grupo CodeDeploy de implantação -> CodeDeploy implantação.	Dependendo do uso, da implantação local ou Blue/Green do aplicativo. Para obter mais detalhes, consulte CodeDeploy solicitações.

Planejando a implantação de seu aplicativo no AMS

Para obter um conjunto recomendado de perguntas a serem respondidas para permitir implantações de aplicativos, consulteApêndice: Questionário de integração de aplicativos. As perguntas abrangem a descrição de:

- Resumo da implantação
- Componentes de implantação de infraestrutura
- Plataforma de hospedagem de aplicativos
- Modelo de implantação de aplicativos
- Dependências de aplicativos
- Certificados SSL para aplicativos de produtos

Ingestão de workload do AMS (WIGS)

Tópicos

- Migração de cargas de trabalho: pré-requisitos para Linux e Windows
- Como a migração altera seu recurso
- Migração de workloads: processo padrão
- Migração de cargas de trabalho: CloudEndure landing zone (SALZ)
- Conta do AMS Tools (migrando cargas de trabalho)
- Migração de cargas de trabalho: validação de pré-ingestão do Linux
- Migração de workloads: validação de pré-ingestão do Windows
- Pilha de ingestão de carga de trabalho: criação

Use o tipo de alteração de ingestão de carga de trabalho (CT) do AMS com um parceiro de migração para a nuvem do AMS para mover suas cargas de trabalho existentes para uma VPC. AMS-managed Usando a ingestão de carga de trabalho do AMS, você pode criar uma AMI personalizada do AMS depois de mover as instâncias migradas para o AMS. Esta seção descreve o processo, os pré-requisitos e as etapas que você e seu parceiro de migração adotam para a ingestão da carga de trabalho do AMS.

> **Important**
>
> O sistema operacional deve ser suportado pela ingestão de carga de trabalho do AMS. Para obter os sistemas operacionais compatíveis, consulteMigração de cargas de trabalho: pré-requisitos para Linux e Windows.
> Cada carga de trabalho e cada conta são diferentes. A AMS trabalhará com você para se preparar para um resultado bem-sucedido.

O diagrama a seguir mostra o processo de ingestão da carga de trabalho do AMS.

Migração de cargas de trabalho: pré-requisitos para Linux e Windows

Antes de ingerir uma cópia de uma instância local no AWS Managed Services (AMS), alguns pré-requisitos devem ser atendidos. Esses são os pré-requisitos, incluindo aqueles que diferem entre os sistemas operacionais Windows e Linux.

> **Note**
>
> Para simplificar o processo de determinar se as instâncias estão prontas para ingestão, foram criadas ferramentas de validação para Windows e Linux. Essas ferramentas podem ser baixadas e executadas diretamente em seus servidores locais, bem como em EC2 instâncias na AWS. Linux pré-WIGS Validation.zip, Windows pré-WIGS Validation.zip.

ANTES DE COMEÇAR, para Linux e Windows:

- Execute uma verificação completa de vírus.
- A instância deve ter o perfil da `customer-mc-ec2-instance-profile` instância.
- Instale o agente do Amazon EC2 Systems Manager (SSM) e certifique-se de que o agente SSM esteja ativo e funcionando.
- Recomenda-se um mínimo de 10 GB de espaço livre em disco no volume raiz para executar o AMS Workload Ingest (WIGS). Operacionalmente, o AMS recomenda uma utilização de disco inferior a 75% e alerta quando a utilização do disco atinge 85%.
- Determine um prazo para a ingestão com seu parceiro de migração.
- A AMI personalizada existe como uma EC2 instância na conta AMS de produção de destino (essa é a responsabilidade do parceiro de migração).

> **Important**
>
> O sistema operacional deve ser suportado pela ingestão de carga de trabalho do AMS.
>
> - Há suporte para os seguintes sistemas operacionais:
>   - Microsoft Windows Server: 2008 R2, 2012, 2012 R2, 2016, 2019 e 2022
>   - Linux: Amazon Linux 2023, Amazon Linux 2 e Amazon Linux, CentOS 7.x, CentOS 6.5-6.10, Oracle Linux 7: versões secundárias 7.5 e superiores, Oracle Linux 8: versões secundárias até 8.3, RHEL 8.x, RHEL 7.x, RHEL 6.5-6.10, SUSE Linux Enterprise Server 15 e versões específicas do SAP, SUSE Linux Enterprise Server 12, Ubuntu 18.04 SP3 SP4 SP5
> - O seguinte não AMIs é suportado:
>   - AMI mínima do Amazon Linux 2023.

> **Note**
>
> Os endpoints AMS API/CLI (amscm e amsskms) estão na região da AWS Norte da Virgínia,. `us-east-1` Dependendo de como sua autenticação está configurada e em qual região da AWS sua conta e seus recursos estão, talvez seja necessário adicioná-la `--region us-east-1` ao emitir comandos. Talvez você também precise adicionar`--profile saml`, se esse for o seu método de autenticação.

Pré-requisitos do LINUX

Observe os requisitos listados Migração de cargas de trabalho: pré-requisitos para Linux e Windows e garanta o seguinte antes de enviar um WIGS RFC:

- Os drivers de rede aprimorados mais recentes estão instalados; consulte Rede aprimorada no Linux.
- Os componentes de software de terceiros que entrarão em conflito com os componentes do AMS foram removidos:
  - Clientes antivírus
  - Clientes de backup
  - Software de virtualização (como VM Tools ou serviços de integração Hyper-V)
  - Software de gerenciamento de acesso (como SSSD, Centrify ou PBIS)
- Verifique se o SSH está configurado corretamente - Isso ativa temporariamente a autenticação de chave privada para SSH. O AMS usa isso com nossa ferramenta de gerenciamento de configuração. Use estes comandos:

```
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
```

- Certifique-se de que o Yum esteja configurado corretamente - RedHat requer licenciamento para usar seus repositórios Yum. A instância precisa ser licenciada por meio de um servidor satélite ou servidor RedHat em nuvem. Use um desses links se o licenciamento for necessário:
  - Red Hat Satellite
  - Acesso à nuvem Red Hat
- Se você usa o Red Hat Satellite, o WIGS requer a adição do Red Hat Software Collections (RHSCL). O sistema WIGS usa o RHSCL para adicionar um interpretador Python3.6 junto com tudo o que está configurado no sistema. Para oferecer suporte a essa solução, os seguintes repositórios devem estar disponíveis:
  - rhel-server-rhscl
  - rhel-server-releases-optional

Pré-requisitos do para Windows

Observe os requisitos listados Migração de cargas de trabalho: pré-requisitos para Linux e Windows e garanta o seguinte antes de enviar um WIGS RFC:

- A versão 3 ou superior do Powershell está instalada.
- O AWS EC2 Config é instalado na instância com a carga de trabalho que você migrará.
- Instale os drivers da AWS que suportam os tipos de instância de última geração: PV, ENA e. NVMe Você pode usar as informações nos seguintes links:
  - Atualizando drivers fotovoltaicos em suas instâncias do Windows
  - Rede aprimorada no Windows
  - NVMe Drivers da AWS para instâncias do Windows
  - Parte 3: Atualizando os drivers da AWS NVMe
  - Parte 5: Instalando o driver da porta serial para instâncias bare metal
  - Parte 6: Atualizando as configurações de gerenciamento de energia
- (Opcional, mas recomendado) Desativar serviços críticos — Defina os serviços essenciais do aplicativo, como bancos de dados, como desativados, mas certifique-se de que todas as alterações sejam documentadas para que possam ser revertidas ao modo de inicialização original durante o estágio de verificação do aplicativo.
- (Opcional, mas recomendado) Crie uma AMI à prova de falhas a partir da instância preparada:
  - Use o Deployment | Advanced stack components | AMI | Create

  - Durante a criação, adicione uma tag Key=Name, value=application-id_ IngestReady
  - Espere até que a AMI seja criada antes de continuar
- Os componentes de software de terceiros que entrarão em conflito com os componentes do AMS foram removidos:
  - Clientes antivírus
  - Clientes de backup
  - Software de virtualização (como VM Tools ou serviços de integração Hyper-V)

> **Note**
>
> O Programa de End-of-Support Migração para Windows Server (EMP) inclui ferramentas para migrar seus aplicativos legados do Windows Server 2003, 2008 e 2008 R2 para versões mais novas e compatíveis na AWS, sem nenhuma refatoração.

Como a migração altera seu recurso

A RFC de ingestão descrita nesta seção dá a próxima etapa de adicionar configurações à instância, depois que ela é migrada para sua conta do AMS, para que o AMS possa gerenciá-la.

As configurações adicionadas são AMS-specific as seguintes.

Alterações feitas nas instâncias Linux ingeridas:

- Software que está instalado:
  - Cloud Init: usado para configurar chaves privadas para o Jarvis Access.
  - Python 3 (linguagem de script) para todos os sistemas operacionais compatíveis (exceto CentOS 6, RHEL 8, 7). OracleLinux
  - Scripts auxiliares do AWS CloudFormation Python: a AWS CloudFormation fornece scripts usados para instalar software e iniciar serviços em uma instância do Amazon EC2.
  - AWS CLI: A AWS CLI é uma ferramenta de código aberto criada com base no AWS SDK para Python (Boto) que fornece comandos para interagir com os serviços da AWS.
  - AWS SSM Agent: O agente SSM processa solicitações do serviço Systems Manager e configura a máquina conforme especificado na solicitação.
  - AWS CloudWatch Logs Agent: envia registros para CloudWatch.

  - AWS CodeDeploy: um serviço de implantação que automatiza implantações de aplicativos em instâncias do Amazon EC2, instâncias locais ou funções Lambda sem servidor.
  - Ruby: Obrigatório para CodeDeploy
  - Ferramentas de desempenho do sistema (sysstat): o Sysstat contém vários utilitários para monitorar o desempenho do sistema e a atividade de uso.
  - AD Bridge (anteriormente PowerBroker Identity Services): une hosts que não são da Microsoft aos domínios do Active Directory.
  - Agente de segurança Trend Micro Deep: Anti-Virus software.
- Software que foi alterado:
  - As instâncias são configuradas para usar o fuso horário UTC.

Alterações feitas nas instâncias ingeridas do Windows:

- Software que está instalado:
  - AWS Tools for Windows PowerShell: As ferramentas da AWS PowerShell permitem que desenvolvedores e administradores gerenciem seus serviços e recursos da AWS no ambiente de PowerShell scripting.
  - Agente de segurança Trend Micro Deep: Anti-Virus proteção
  - PowerShell Módulos AMS contendo PowerShell código para controle de inicialização, junção do Active Directory, monitoramento, segurança e registro.
- Software que foi alterado:
  - A versão 1 do Server Message Block (SMB) está desativada.
  - O Gerenciamento Remoto do Windows (WinRM) está habilitado e configurado para escutar na porta 5986. Uma regra de firewall que permite essa porta de entrada também é criada.
- Software que pode ser instalado ou alterado:
  - Microsoft.Net Framework 4.5 (plataforma de desenvolvedor), se for detectada uma versão inferior a.Net Framework 4.5.
  - Para o Windows 2012 e o Windows 2012R2, atualizamos para a versão 5.1. PowerShell

Migração de workloads: processo padrão

> **Note**
>
> Como são necessárias duas partes para esse processo, esta seção descreve as tarefas de cada uma: um parceiro de migração de nuvem do AMS (parceiro de migração) e um proprietário do aplicativo (você).

1. Parceiro de migração, configure:

   a. O parceiro de migração envia uma solicitação de serviço ao AMS para uma função do IAM com o objetivo de migrar sua instância. Para obter detalhes sobre o envio de solicitações de serviço, consulte Exemplos de solicitações de serviço.

   b. O parceiro de migração envia uma solicitação de acesso de administrador. A equipe de operações do AMS fornece ao parceiro de migração acesso à sua conta por meio da função do IAM solicitada.

2. Parceiro de migração, Migrate Individual Workloads:

   a. O parceiro de migração migra sua AWS não-instância para uma sub-rede em sua conta do AMS por meio do Amazon EC2 nativo ou de outras ferramentas de migração, com `customer-mc-ec2-instance-profile` o perfil de instância do IAM (deve estar na conta).

   b. O parceiro de migração envia uma RFC com a instância migrada Deployment | Inestion | Stack from migration partner migrated instance | Create CT (ct-257p9zjk14ija); para obter detalhes sobre como criar e enviar essa RFC, consulte. Pilha de ingestão de carga de trabalho: criação

      A saída de execução da RFC retorna um ID de instância, endereço IP e ID de AMI.

      O parceiro de migração fornece a você o ID da instância da carga de trabalho criada em sua conta.

3. Você, acessa e valida a migração:

   a. Usando a saída de execução fornecida a você (ID da AMI, ID da instância e endereço IP) pelo parceiro de migração, envie uma RFC de acesso, faça login na pilha AMS recém-

criada e verifique se o aplicativo está funcionando corretamente. Para obter detalhes, consulte Solicitação de acesso à instância.

b. Se estiver satisfeito, você pode continuar usando a instância executada como uma pilha de 1 camada and/or usando a AMI para criar pilhas adicionais, incluindo grupos de Auto Scaling.

c. Se não estiver satisfeito com a migração, registre uma solicitação de serviço e consulte a pilha e os IDs de RFC. O AMS trabalhará com você para resolver suas preocupações.

CloudEndure o processo de ingestão de carga de trabalho da landing zone é descrito a seguir.

Migração de cargas de trabalho: CloudEndure landing zone (SALZ)

Esta seção fornece informações sobre como configurar uma zona de pouso de conta única (SALZ) de migração intermediária para que instâncias de transição CloudEndure (CE) estejam disponíveis para uma RFC de ingestão de carga de trabalho (WIGS).

Para saber mais CloudEndure, consulte CloudEndure Migração.

> **Note**
>
> Esse é um LZ e um padrão de migração predefinidos e reforçados pela segurança.

Pré-requisitos:

- Uma conta AMS de cliente
- Integração de rede e acesso entre a conta AMS e o cliente no local
- Uma CloudEndure conta
- Um fluxo de trabalho de pré-aprovação para uma revisão e aprovação de segurança do AMS, executado com seu CA and/or CSDM (por exemplo, o uso indevido das credenciais permanentes do usuário do IAM possibilita instâncias e grupos de segurança) create/delete

> **Note**
>
> Os processos específicos de preparação e migração são descritos nesta seção.

Preparação: Você e o operador AMS:

1. Prepare uma Solicitação de Mudança (RFC) com o Gerenciamento | Outros | Outros | Atualize o tipo de alteração para AMS para os seguintes recursos e atualizações. Você pode enviar outros | Outros RFCs de atualização separados, ou um. Para obter detalhes sobre isso RFC/CT, consulte Outros | Outra atualização com essas solicitações:

   a. Atribua um bloco CIDR secundário em sua VPC do AMS; um bloco CIDR temporário que será removido após a conclusão da migração. Certifique-se de que o bloqueio não entre em conflito com nenhuma rota existente de volta à sua rede local. Por exemplo, se seu AMS VPC CIDR for 10.0.0. 0/16, e há uma rota de volta para sua rede local de 10.1.0. 0/16, então o CIDR secundário temporário poderia ser 10.255.255. 0/24. Para obter informações sobre blocos CIDR da AWS, consulte Dimensionamento de VPC e sub-rede.

   b. Crie uma nova sub-rede privada dentro do jardim inicial do AMS VPC. Nome de exemplo:`migration-temp-subnet`.

   c. Crie uma nova tabela de rotas para a sub-rede somente com rotas locais de VPC e NAT (Internet), para evitar conflitos com o servidor de origem durante a transição da instância e possíveis interrupções. Certifique-se de que o tráfego de saída para a Internet seja permitido para downloads de patches e para que os pré-requisitos do AMS WIGS possam ser baixados e instalados.

   d. Atualize seu grupo de segurança do AD gerenciado para permitir tráfego de entrada e saída. to/from `migration-temp-subnet` Solicite também que seu grupo de segurança (ex:) do balanceador de carga EPS (ELB`mc-eps-McEpsElbPrivateSecurityGroup-M790XBZEEX74`) seja atualizado para permitir a nova sub-rede privada (ou seja,). `migration-temp-subnet` Se o tráfego da sub-rede dedicada CloudEndure (CE) não for permitido nas três portas TCP, a ingestão do WIGS falhará.

   e. Por fim, solicite uma nova política CloudEndure do IAM e um novo usuário do IAM. <Customer Application Subnet (s) + Temp Migration Subnet>A política precisa do seu número de conta correto, e os IDs de sub-rede no `RunInstances` extrato devem ser: seu.

      Para ver uma CloudEndure política IAM pré-aprovada pelo AMS: Descompacte o arquivo de exemplo do WIGS Cloud Endure Landing Zone e abra o. `customer_cloud_endure_policy.json`

> **Note**
>
> Se você quiser uma política mais permissiva, discuta o que você precisa com você CloudArchitect/CSDM e obtenha, se necessário, uma revisão e aprovação de segurança do AMS antes de enviar uma RFC implementando a política.

2. Suas etapas de preparação CloudEndure para uso na ingestão da carga de trabalho do AMS estão concluídas e, se seu parceiro de migração tiver concluído as etapas de preparação, a migração estará pronta para ser executada. O WIGS RFC é enviado pelo seu parceiro de migração.

> **Note**
>
> As chaves de usuário do IAM não serão compartilhadas diretamente, mas devem ser digitadas no console CloudEndure de gerenciamento pelo operador do AMS em uma sessão de compartilhamento de tela.

Preparação: Parceiro de migração e operador de AMS:

1. Crie um projeto de CloudEndure migração.
    a. Durante a criação do projeto, faça com que o AMS digite as credenciais de usuário do IAM nas sessões de compartilhamento de tela.
    b. Em Configurações de replicação -> Escolha a sub-rede em que os servidores de replicação serão iniciados, selecione customer-application-x sub-net.
    c. Em Configurações de replicação -> Escolha os grupos de segurança a serem aplicados aos servidores de replicação, selecione os dois grupos de segurança do Sentinel (somente privado e). EgressAll
2. Defina as opções de transição para as máquinas (instâncias).
    a. Sub-rede: migration-temp-subnet.
    b. Grupo de segurança: Ambos os grupos de segurança “Sentinel” (somente privado e EgressAll).

As instâncias de transferência devem ser capazes de se comunicar com o AMS Managed AD e com os endpoints públicos da AWS.

c. IP elástico: nenhum

d. IP público: não

e. Função do IAM: customer-mc-ec2-instance-profile

A função do IAM deve permitir a comunicação SSM. Melhor usar o padrão AMS.

f. Defina as tags de acordo com a convenção.

Migração: Parceiro de migração:

1. Crie uma pilha fictícia no AMS. Você usa o ID da pilha para obter acesso aos bastiões.
2. Instale o agente CloudEndure (CE) no servidor de origem. Para obter detalhes, consulte Instalação dos agentes.
3. Crie credenciais de administrador local no servidor de origem.
4. Agende uma pequena janela de substituição e clique em Substituir, quando estiver pronto. Isso finaliza a migração e redireciona os usuários para a região da AWS de destino.
5. Solicite acesso de administrador à pilha fictícia, consulte Solicitação de acesso de administrador.
6. Faça login no bastion e, em seguida, na instância de transferência usando as credenciais de administrador local que você criou.
7. Crie uma AMI à prova de falhas. Para obter detalhes sobre a criação de AMIs, consulte Criação de AMI.
8. Prepare a instância para ingestão, consulteMigração de cargas de trabalho: pré-requisitos para Linux e Windows.
9. Execute o WIGS RFC na instância, consulte. Pilha de ingestão de carga de trabalho: criação

Conta do AMS Tools (migrando cargas de trabalho)

Sua conta de ferramentas da Multi-Account Landing Zone (com VPC) ajuda a acelerar os esforços de migração, aumenta sua posição de segurança, reduz custos e complexidade e padroniza seu padrão de uso.

Uma conta de ferramentas fornece o seguinte:

- Um limite bem definido para acesso às instâncias de replicação para integradores de sistemas fora de suas cargas de trabalho de produção.
- Permite criar uma câmara isolada para verificar se há malware em uma carga de trabalho ou rotas de rede desconhecidas antes de colocá-la em uma conta com outras cargas de trabalho.
- Como uma configuração de conta definida, ela fornece um tempo mais rápido de integração e configuração para migrar cargas de trabalho.
- Rotas de rede isoladas para proteger o tráfego do local -> -> Conta de ferramentas CloudEndure -> Imagem ingerida pelo AMS. Depois que uma imagem é ingerida, você pode compartilhá-la com a conta de destino por meio de um AMS Management | Advanced stack components | AMI | Share (ct-1eiczxw8ihc18) RFC.

Diagrama de arquitetura de alto nível:

Use o tipo de alteração Deployment | Managed landing zone | Management account | Create tools account (com VPC) (ct-2j7q1hgf26x5c) para implantar rapidamente uma conta de ferramentas e instanciar um processo de ingestão de carga de trabalho em um ambiente Landing Zone. Multi-Account Consulte Conta de gerenciamento, Conta de ferramentas: criação (com VPC).

> **Note**
>
> Recomendamos ter duas zonas de disponibilidade (AZs), já que esse é um hub de migração. Por padrão, o AMS cria os dois grupos de segurança (SGs) a seguir em cada conta. Confirme se esses dois SGs estão presentes. Se eles não estiverem presentes, abra uma nova solicitação de serviço com a equipe do AMS para solicitá-los.
>
> - SentinelDefaultSecurityGroupPrivateOnlyEgressAll
> - InitialGarden-SentinelDefaultSecurityGroupPrivateOnly
>
> Garanta que CloudEndure as instâncias de replicação sejam criadas na sub-rede privada, onde há rotas de volta para o local. Você pode confirmar isso garantindo que as tabelas de rotas da sub-rede privada tenham uma rota padrão de volta para o TGW. No entanto, realizar um corte CloudEndure de máquina deve ir para a sub-rede privada “isolada”, onde não há nenhuma rota de volta para o local, somente o tráfego de saída da Internet é permitido. É fundamental garantir que a transferência ocorra na sub-rede isolada para evitar possíveis problemas nos recursos locais.

Pré-requisitos:

1. Nível de suporte Plus ou Premium.
2. Os IDs da conta do aplicativo para a chave KMS em que as AMIs são implantadas.
3. A conta de ferramentas, criada conforme descrito anteriormente.

AWS Serviço de migração de aplicativos (AWS CANECA)

AWS O Application Migration Service (AWS MGN) pode ser usado em sua conta do MALZ Tools por meio da função `AWSManagedServicesMigrationRole` IAM que é criada automaticamente durante o provisionamento da conta do Tools. Você pode usar o AWS MGN para migrar aplicativos e bancos de dados que são executados em versões compatíveis dos sistemas operacionais Windows e Linux.

Para obter as informações mais atualizadas sobre Região da AWS suporte, consulte a Lista de serviços AWS regionais.

Se sua preferência não Região da AWS for suportada atualmente pela AWS MGN, ou se o sistema operacional no qual seus aplicativos são executados não for atualmente suportado pela AWS MGN, considere usar a CloudEndure Migração em sua conta de Ferramentas.

Solicitando a inicialização do AWS MGN

O AWS MGN deve ser inicializado pelo AMS antes do primeiro uso. Para solicitar isso para uma nova conta de Ferramentas, envie um RFC de Gerenciamento | Outro | Outro da conta de Ferramentas com estes detalhes:

```
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.
```

Depois que o AMS concluir com êxito o RFC e inicializar o AWS MGN em sua conta de Ferramentas, você poderá usá-lo `AWSManagedServicesMigrationRole` para editar o modelo padrão de acordo com seus requisitos.

Habilitar o acesso à nova conta do AMS Tools

Depois que a conta de ferramentas é criada, o AMS fornece um ID de conta. Sua próxima etapa é configurar o acesso à nova conta. Siga estas etapas.

1. Atualize os grupos apropriados do Active Directory com as IDs de conta apropriadas.

   As novas AMS-created contas são provisionadas com a política de funções, bem como com uma ReadOnly função para permitir que os usuários arquivem RFCs.

   A conta Tools também tem uma função e um usuário adicionais do IAM disponíveis:

   - Perfil do IAM: `AWSManagedServicesMigrationRole`
   - Usuário do IAM: `customer_cloud_endure_user`
2. Solicite políticas e funções para permitir que os membros da equipe de integração de serviços configurem o próximo nível de ferramentas.

   Navegue até o console do AMS e registre os seguintes RFCs:

   a. Crie uma chave KMS. Use Criar chave KMS (automático) ou Criar chave KMS (automação gerenciada).

      Ao usar o KMS para criptografar recursos ingeridos, o uso de uma única chave KMS que é compartilhada com o resto das contas do aplicativo Multi-Account Landing Zone fornece segurança para imagens ingeridas, onde elas podem ser descriptografadas na conta de destino.

   b. Compartilhe a chave KMS.

      Use o tipo de alteração Gerenciamento | Componentes avançados da pilha | Chave KMS | Compartilhar (automação gerenciada) (ct-05yb337abq3x5) para solicitar que a nova chave KMS seja compartilhada com suas contas de aplicativo onde residirão as AMIs ingeridas.

Exemplo de gráfico da configuração final de uma conta:

Exemplo de política de IAM CloudEndure pré-aprovada pelo AMS

Para ver uma CloudEndure política IAM pré-aprovada pelo AMS: Descompacte o arquivo de exemplo do WIGS Cloud Endure Landing Zone e abra o. `customer_cloud_endure_policy.json`

Testando a conectividade da conta do AMS Tools e a configuração de ponta a ponta

1. Comece configurando CloudEndure e instalando o CloudEndure agente em um servidor que será replicado para o AMS.
2. Crie um projeto em CloudEndure.
3. Insira as AWS credenciais compartilhadas quando você executou os pré-requisitos, por meio do gerenciador de segredos.
4. Nas configurações de replicação:
   a. Selecione os dois grupos de segurança “Sentinel” do AMS (somente privado e EgressAll) para a opção Escolher os grupos de segurança a serem aplicados aos servidores de replicação.
   b. Defina as opções de transição para as máquinas (instâncias). Para obter informações, consulte a Etapa 5. Cortar
   c. Sub-rede: sub-rede privada.
5. Grupo de segurança:
   a. Selecione os dois grupos de segurança “Sentinel” do AMS (somente privado e EgressAll).
   b. As instâncias de transferência precisam se comunicar com o AMS-managed Active Directory (MAD) e com endpoints AWS públicos:
      i. IP elástico: nenhum
      ii. IP público: não
      iii. Função do IAM: customer-mc-ec2-instance-profile
   c. Defina as tags de acordo com sua convenção interna de marcação.
6. Instale o CloudEndure agente na máquina e procure a instância de replicação que aparecerá na sua conta do AMS no console do EC2.

O processo de ingestão do AMS:

Higiene da conta AMS Tools

Você vai querer limpar depois de concluir que a conta compartilhou a AMI e não precisa mais das instâncias replicadas:

- WIGs Ingestão de pós-instância:

  - Instância de transição: no mínimo, interrompa ou encerre essa instância, após a conclusão do trabalho, por meio do console da AWS
  - Backups de AMI de pré-ingestão: remova quando a instância for ingerida e a instância local encerrada
  - Instâncias ingeridas pelo AMS: desative a pilha ou encerre depois que a AMI for compartilhada
  - Ingerido pelo AMS AMIs: exclua quando o compartilhamento com a conta de destino for concluído
- Limpeza do fim da migração: documente os recursos implantados por meio do modo Desenvolvedor para garantir que a limpeza ocorra regularmente, por exemplo:
  - Grupos de segurança
  - Recursos criados por meio do Cloud-formation
  - Rede ACK
  - Sub-rede
  - VPC
  - Tabela de rotas
  - Perfis
  - Usuários e contas

Migração em grande escala - Migration Factory

Consulte Apresentação da solução AWS CloudEndure Migration Factory.

Migração de cargas de trabalho: validação de pré-ingestão do Linux

Você pode validar se sua instância está pronta para ser ingerida em sua conta do AMS. A validação de pré-ingestão da carga de trabalho (WIGS) realiza verificações como tipo de sistema operacional, espaço em disco disponível, existência de software de terceiros conflitante etc. Quando executada, a validação de pré-ingestão do WIGS produz uma tabela na tela, com um arquivo de log opcional. Os resultados fornecem um pass/fail status para cada verificação de validação junto com o motivo de qualquer falha. Além disso, você pode personalizar os testes de validação para atender às suas necessidades.

Perguntas frequentes:

- Como faço para usar a validação de pré-ingestão do Linux WIGS?

Siga estas etapas para baixar e usar os scripts de validação de pré-ingestão do AMS Linux WIGS:

1. Baixe um arquivo ZIP com os scripts de validação

   Arquivo zip de validação de pré-ingestão do Linux WIGS.
2. Descompacte as regras anexadas em um diretório de sua escolha.
3. Siga as instruções no arquivo readme.md.

- Quais validações são realizadas pela validação de pré-ingestão do Linux WIGS?

  A solução de validação de pré-ingestão AMS Linux WIGS valida o seguinte:

  1. Há pelo menos 5 Gigabytes livres no volume de inicialização.
  2. O sistema operacional é suportado pelo AMS.
  3. A instância tem um perfil de instância específico.
  4. A instância não contém software antivírus ou software de virtualização.
  5. O SSH está configurado corretamente.
  6. A instância tem acesso aos repositórios Yum.
  7. Drivers de rede aprimorados estão instalados.
  8. A instância tem o agente SSM e está em execução.
- Por que há suporte para um arquivo de configuração personalizado?

  Os scripts foram projetados para serem executados em servidores físicos locais e em instâncias da AWS EC2 . No entanto, conforme mostrado na lista acima, alguns testes falharão quando executados localmente. Por exemplo, um servidor físico em um datacenter não teria um perfil de instância. Em casos como esses, você pode editar o arquivo de configuração para ignorar o teste do perfil da instância e evitar confusão.
- Como posso garantir que tenho a versão mais recente do script?

  Uma up-to-date versão da solução de validação de pré-ingestão Linux WIGS estará disponível na seção Arquivos auxiliares do AMS na página principal da documentação.
- O script é somente para leitura?

  O script foi projetado para ser somente para leitura, exceto pelos arquivos de log que ele produz, mas as melhores práticas devem ser seguidas para executar o script em um ambiente que não seja de produção.
- A validação de pré-ingestão do WIGS está disponível para Windows?

Sim. Ele está disponível na seção Arquivos auxiliares do AMS na página principal da documentação.

Migração de workloads: validação de pré-ingestão do Windows

Você pode usar o script WIGs pré-validador para validar se sua instância está pronta para ser ingerida em sua conta do AMS. A validação de pré-ingestão da carga de trabalho (WIGS) realiza verificações como tipo de sistema operacional, espaço em disco disponível, existência de software de terceiros conflitante e assim por diante. Quando executada, a validação de pré-ingestão do WIGS produz uma tabela na tela e um arquivo de log opcional. Os resultados fornecem um pass/fail status para cada verificação de validação junto com o motivo da falha. Além disso, você pode personalizar os testes de validação.

Perguntas frequentes:

- Como faço para usar a validação de pré-ingestão do Windows WIGS?

  Você pode executar a validação a partir de uma GUI e de um navegador da Web ou usar o Windows PowerShell, o SSM Run Command ou o SSM Session Manager.

  Opção 1: Executar a partir de uma GUI e de um navegador da web

  Para executar a WIGs pré-validação do Windows a partir de uma GUI e de um navegador da Web, faça o seguinte:

  1. Baixe um arquivo ZIP com os scripts de validação:

     Arquivo ZIP de validação de pré-ingestão do Windows WIGS.
  2. Descompacte as regras anexadas em um diretório de sua escolha.
  3. Siga as instruções no arquivo README.md.

  Opção 2: Executar a partir do Windows PowerShell, SSM Run Command ou SSM Session Manager

  Windows 2016 e versões posteriores

  1. Baixe o arquivo ZIP com os scripts de validação.

     ```
     $DestinationFile = "$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. Remova os arquivos existentes do`C:\Users\AppData\Local\Temp \AWSManagedServices.PreWigs.Validation`.

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

3. Invoque o script.

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

4. Descompacte os arquivos anexados em um diretório de sua escolha.

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

5. Execute o script de validação interativamente e veja os resultados.

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

6. (Opcional) Para capturar os códigos de erro listados na seção Códigos de saída, execute o script sem a `RunWithoutExitCodes` opção. Observe que esse comando encerra a PowerShell sessão ativa.

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

Windows 2012 R2 e versões anteriores

Se você estiver executando o Windows Server 2012R2 ou inferior, deverá definir o TLS antes de baixar o arquivo zip. Para definir o TLS, conclua as seguintes etapas:

1. Baixe o arquivo ZIP com os scripts de validação.

```
$DestinationFile = "$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. Se houver arquivos de validação existentes, remova-os.

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

3. Defina a versão do TLS.

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

4. Baixe a validação do WIG.

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

5. Descompacte as regras anexadas em um diretório de sua escolha.

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

6. Execute o script de validação interativamente e veja os resultados.

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

7. (Opcional) Para capturar os códigos de erro listados na seção Códigos de saída, execute o script sem a RunWithoutExitCodes opção. Observe que esse comando encerra a PowerShell sessão ativa.

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

> **Note**
>
> Você pode baixar e executar os PowerShell scripts. Para fazer isso, baixe o pre-wigs-validation-powershell-scripts.zip.

- Quais validações são realizadas pela Validação de Pré-Ingestão do Windows WIGS?

  A solução de validação de pré-ingestão AMS Windows WIGS valida o seguinte:

1. Há pelo menos 10 Gigabytes livres no volume de inicialização.
2. O sistema operacional é suportado pelo AMS.
3. A instância tem um perfil de instância específico.
4. A instância não contém software antivírus ou software de virtualização.
5. O DHCP está habilitado em pelo menos um adaptador de rede.
6. A instância está pronta para o Sysprep.
   - Para 2008 R2 e 2012 Base e R2, o Sysprep verifica se:
     - Existe um arquivo unattend.xml
     - O arquivo sppnp.dll (se presente) não está corrompido
     - O sistema operacional não foi atualizado
     - O Sysprep não foi executado mais do que o número máximo de vezes de acordo com as diretrizes da Microsoft
   - Para 2016 e versões posteriores, todas as verificações acima foram ignoradas, pois nenhuma delas causa problemas para esse sistema operacional
7. O subsistema de instrumentação de gerenciamento do Windows (WMI) está íntegro.
8. Os drivers necessários estão instalados.
9. O SSM Agent e está instalado e em execução.
10. É emitido um aviso para verificar se a máquina está em período de carência devido à configuração da licença do RDS.
11. As chaves de registro necessárias estão definidas corretamente. Para obter mais detalhes, consulte o README no arquivo zip de validação de pré-ingestão.

- Por que há suporte para um arquivo de configuração personalizado?

  Os scripts foram projetados para serem executados em servidores físicos locais e em instâncias da AWS EC2 . No entanto, conforme mostrado na lista acima, alguns testes falharão quando executados localmente. Por exemplo, um servidor físico em um datacenter não teria um perfil de instância. Em casos como esses, você pode editar o arquivo de configuração para ignorar o teste do perfil da instância e evitar confusão.

- Como posso garantir que tenho a versão mais recente do script?

  Uma up-to-date versão da solução de validação de pré-ingestão do Windows WIGS estará disponível na seção Arquivos auxiliares do AMS na página principal da documentação.

- O script é somente para leitura?

O script foi projetado para ser somente para leitura, exceto pelos arquivos de log que ele produz, mas as melhores práticas devem ser seguidas para executar o script em um ambiente que não seja de produção.

- A validação de pré-ingestão do WIGS está disponível para Linux?

  Sim. A versão Linux foi lançada em 31 de outubro de 2019. Ele está disponível na seção Arquivos auxiliares do AMS na página principal da documentação.

Pilha de ingestão de carga de trabalho: criação

Migração de uma instância para uma pilha AMS com o console

Captura de tela desse tipo de alteração no console AMS:

Como funciona:

1. Navegue até a página Criar RFC: No painel de navegação esquerdo do console AMS, clique em RFCs para abrir a página de lista de RFCs e, em seguida, clique em Criar RFC.
2. Escolha um tipo de alteração popular (CT) na visualização padrão Procurar tipos de alteração ou selecione uma CT na visualização Escolher por categoria.
   - Navegar por tipo de alteração: você pode clicar em um CT popular na área de criação rápida para abrir imediatamente a página Executar RFC. Observe que você não pode escolher uma versão mais antiga do CT com a criação rápida.

     Para classificar CTs, use a área Todos os tipos de alteração na visualização em Cartão ou Tabela. Em qualquer exibição, selecione uma CT e clique em Criar RFC para abrir a página Executar RFC. Se aplicável, a opção Criar com versão mais antiga aparece ao lado do botão Criar RFC.
   - Escolha por categoria: selecione uma categoria, subcategoria, item e operação e a caixa de detalhes do CT será aberta com a opção Criar com uma versão mais antiga, se aplicável. Clique em Criar RFC para abrir a página Executar RFC.
3. Na página Executar RFC, abra a área do nome do CT para ver a caixa de detalhes do CT. É necessário um Assunto (preenchido se você escolher seu CT na visualização Procurar tipos de alteração). Abra a área Configuração adicional para adicionar informações sobre o RFC.

Na área Configuração de execução, use as listas suspensas disponíveis ou insira valores para os parâmetros necessários. Para configurar parâmetros de execução opcionais, abra a área Configuração adicional.

4. Ao terminar, clique em Executar. Se não houver erros, a página RFC criada com sucesso será exibida com os detalhes da RFC enviada e a saída inicial de execução.
5. Abra a área Parâmetros de execução para ver as configurações que você enviou. Atualize a página para atualizar o status de execução do RFC. Opcionalmente, cancele a RFC ou crie uma cópia dela com as opções na parte superior da página.

> **Note**
>
> Se a RFC for rejeitada, a saída da execução incluirá um link para CloudWatch os registros da Amazon. Os RFCs do AMS Workload Ingest (WIGS) são rejeitados quando os requisitos não são atendidos; por exemplo, se um software antivírus for detectado na instância. Os CloudWatch registros incluirão informações sobre o requisito falhado e as ações a serem tomadas para remediação.

Migração de uma instância para uma pilha AMS com a CLI

Como funciona:

1. Use o Inline Create (você emite um `create-rfc` comando com todos os parâmetros de RFC e execução incluídos) ou o Template Create (você cria dois arquivos JSON, um para os parâmetros RFC e outro para os parâmetros de execução) e emita o `create-rfc` comando com os dois arquivos como entrada. Ambos os métodos são descritos aqui.
2. Envie o `aws amscm submit-rfc --rfc-id` *`ID`* comando RFC: com o ID RFC retornado.

   Monitore o `aws amscm get-rfc --rfc-id` *`ID`* comando RFC:.

Para verificar a versão do tipo de alteração, use este comando:

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

> **Note**
>
> Você pode usar qualquer `CreateRfc` parâmetro com qualquer RFC, independentemente de eles fazerem parte do esquema para o tipo de alteração. Por exemplo, para receber notificações quando o status da RFC mudar, adicione essa linha `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` à parte dos parâmetros da RFC da solicitação (não aos parâmetros de execução). Para obter uma lista de todos os CreateRfc parâmetros, consulte a Referência da API de gerenciamento de alterações do AMS.

Você pode usar a CLI do AMS para criar uma instância do AMS a partir de uma instância não AMS migrada para uma conta do AMS.

> **Note**
>
> Certifique-se de ter seguido os pré-requisitos; consulte Migração de cargas de trabalho: pré-requisitos para Linux e Windows.

Para verificar a versão do tipo de alteração, use este comando:

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

CRIAÇÃO EM LINHA:

Execute o comando create RFC com os parâmetros de execução fornecidos em linha (aspas de escape ao fornecer parâmetros de execução em linha) e, em seguida, envie a ID de RFC retornada. Por exemplo, você pode substituir o conteúdo por algo assim:

```
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\"}"
```

CRIAÇÃO DE MODELO:

1. 0emita os parâmetros de execução do esquema JSON para esse tipo de alteração em um arquivo; o exemplo o nomeia: MigrateStackParams.json

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

2. Modifique e salve o arquivo JSON dos parâmetros de execução. Por exemplo, você pode substituir o conteúdo por algo assim:

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

3. Exiba o arquivo JSON do modelo RFC; o exemplo o nomeia: MigrateStackRfc.json

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

4. Modifique e salve o MigrateStackRfc.json arquivo. Por exemplo, você pode substituir o conteúdo por algo assim:

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

5. Crie o RFC, especificando o MigrateStackRfc arquivo e o MigrateStackParams arquivo:

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

Você recebe o ID do novo RFC na resposta e pode usá-lo para enviar e monitorar o RFC. Até que você o envie, o RFC permanece no estado de edição e não inicia.

A nova instância aparece na lista de instâncias da conta do proprietário do aplicativo para a VPC relevante.

6. Depois que a RFC for concluída com êxito, notifique o proprietário do aplicativo para que ele possa fazer login na nova instância e verificar se a carga de trabalho está operacional.

> **Note**
>
> Se a RFC for rejeitada, a saída da execução incluirá um link para CloudWatch os registros da Amazon. Os RFCs do AMS Workload Ingest (WIGS) são rejeitados quando os requisitos não são atendidos; por exemplo, se um software antivírus for detectado na instância. Os CloudWatch registros incluirão informações sobre o requisito falhado e as ações a serem tomadas para remediação.

Dicas

> **Note**
>
> Certifique-se de ter seguido os pré-requisitos; consulte Migração de cargas de trabalho: pré-requisitos para Linux e Windows.

> **Note**
>
> Se uma tag na instância que está sendo migrada tiver a mesma chave de uma tag fornecida na RFC, a RFC falhará.

> **Note**
>
> Você pode especificar até quatro IDs de destino, portas e zonas de disponibilidade.

> **Note**
>
> Se a RFC for rejeitada, a saída da execução incluirá um link para CloudWatch os registros da Amazon. Os RFCs do AMS Workload Ingest (WIGS) são rejeitados quando os requisitos não são atendidos; por exemplo, se um software antivírus for detectado na instância. Os

CloudWatch registros incluirão informações sobre o requisito falhado e as ações a serem tomadas para remediação.

> **Note**
>
> Se a RFC for rejeitada, a saída da execução incluirá um link para CloudWatch os registros da Amazon. Os RFCs do AMS Workload Ingest (WIGS) são rejeitados quando os requisitos não são atendidos; por exemplo, se um software antivírus for detectado na instância. Os CloudWatch registros incluirão informações sobre o requisito falhado e as ações a serem tomadas para remediação.

Se necessário, consulte Falha na ingestão de carga de trabalho (WIGS).

CloudFormation Ingestão de AMS

O AMS AWS CloudFormation ingest change type (CT) permite que você use seus CloudFormation modelos existentes, com algumas modificações, para implantar pilhas personalizadas em uma AMS-managed VPC.

Tópicos

- CloudFormation Diretrizes, melhores práticas e limitações de ingestão
- CloudFormation Ingerir: exemplos
- Crie uma CloudFormation pilha de ingestão
- Atualizar CloudFormation pilha de ingestão
- Aprovar um conjunto de alterações da CloudFormation pilha de ingestão
- Atualizar CloudFormation proteção de terminação de pilhas
- Implantações automatizadas de IAM usando ingestão de CFN ou atualização de pilha no AMS CTs

O processo CloudFormation de ingestão do AMS envolve o seguinte:

- Prepare e carregue seu CloudFormation modelo personalizado em um bucket do S3 ou forneça o modelo em linha ao criar a RFC. Se você estiver usando um bucket do S3 com uma URL pré-assinada, para obter mais informações, consulte presign.

- Envie o tipo de alteração de CloudFormation ingestão para o AMS em uma RFC. Para ver o passo a passo do tipo de alteração de ingestão do CFN, consulte. Crie uma CloudFormation pilha de ingestão Para exemplos de ingestão de CFN, consulte. CloudFormation Ingerir: exemplos
- Depois que sua pilha for criada, você poderá atualizá-la; além disso, se a atualização falhar, você poderá aprovar e implementar explicitamente a atualização.

> **Note**
>
> A correção de desvios não é suportada pelo AMS para CloudFormation-ingested pilhas. Como proprietário de uma CloudFormation-ingested pilha, você é responsável por gerenciar diretamente a deriva da pilha.

Para obter informações sobre a detecção de desvio de CFN, consulte Novo — Detecção de CloudFormation deriva.

> **Note**
>
> - Esse tipo de alteração agora tem uma versão 2.0. A versão 2.0 é automatizada; não é executada manualmente. Isso permite que a execução da tomografia computadorizada seja mais rápida. Dois novos parâmetros são introduzidos nesta versão: CloudFormationTemplate, que permite colar um CloudFormation modelo personalizado na RFC e VpcId, que permite que a CloudFormation ingestão seja usada com a zona inicial de várias contas do AMS.
> - A versão 1.0 é um tipo de alteração manual. Isso significa que um operador do AMS deve realizar alguma ação antes que o tipo de alteração possa ser concluído com êxito. No mínimo, é necessária uma revisão. Essa versão também exige que o valor do CloudFormationTemplateS3Endpoint parâmetro seja um URL pré-assinado.

CloudFormation Diretrizes, melhores práticas e limitações de ingestão

Para que o AMS processe seu CloudFormation modelo, existem algumas diretrizes e restrições.

Diretrizes

Para reduzir CloudFormation erros ao realizar a CloudFormation ingestão, siga estas diretrizes:

- Não incorpore credenciais ou outras informações confidenciais no modelo — o CloudFormation modelo está visível no CloudFormation console, então você não quer incorporar credenciais ou dados confidenciais no modelo. O modelo não pode conter informações confidenciais. Os seguintes recursos são permitidos somente se você usar o AWS Secrets Manager para o valor:
  - `AWS::RDS::DBInstance` - [MasterUserPassword,TdeCredentialPassword]
  - `AWS::RDS::DBCluster` - [MasterUserPassword]
  - `AWS::ElastiCache::ReplicationGroup` - [AuthToken]

  > **Note**
  >
  > Para obter informações sobre o uso de um segredo do AWS Secrets Manager em uma propriedade de recurso, consulte Como criar e recuperar segredos gerenciados no AWS Secrets Manager usando CloudFormation modelos da AWS e Como usar referências dinâmicas para especificar valores de modelos.

- Use snapshots do Amazon RDS para criar instâncias de banco de dados do RDS — Ao fazer isso, você evita a necessidade de fornecer uma. MasterUserPassword
- Se o modelo enviado contiver um perfil de instância do IAM, ele deverá ser prefixado com “cliente”. Por exemplo, usar um perfil de instância com o nome example-instance-profile '' causa falha. Em vez disso, use um perfil de instância com o nome 'customer-example-instance-profile'.
- Não inclua dados confidenciais em **`AWS::EC2::Instance`** - [UserData]. UserData não deve conter senhas, chaves de API ou outros dados confidenciais. Esse tipo de dado pode ser criptografado e armazenado em um bucket do S3 e baixado na instância usando UserData.
- A criação de políticas do IAM usando CloudFormation modelos é suportada por restrições — as políticas do IAM precisam ser revisadas e aprovadas pelo AMS. SecOps Atualmente, só oferecemos suporte à implantação de funções do IAM com políticas em linha que contêm permissões pré-aprovadas. Em outros casos, as políticas do IAM não podem ser criadas usando CloudFormation modelos porque isso substituiria o SecOps processo do AMS.
- KeyPairs Não há suporte para SSH — as EC2 instâncias da Amazon devem ser acessadas por meio do sistema de gerenciamento de acesso AMS. O processo AMS RFC autentica você. Você não pode incluir pares de chaves SSH em CloudFormation modelos porque você não tem as

permissões para criar pares de chaves SSH e substituir o modelo de gerenciamento de acesso do AMS.

- As regras de entrada do grupo de segurança são restritas — você não pode ter um intervalo de CIDR de origem de 0.0.0.0/0 ou um espaço de endereço publicamente roteável com uma porta TCP diferente de 80 ou 443.
- Siga CloudFormation as diretrizes ao escrever modelos de CloudFormation recursos — Certifique-se de usar o type/property nome de dados correto para o recurso consultando o Guia AWS CloudFormation do usuário desse recurso. Por exemplo, o tipo de dados da SecurityGroupIds propriedade em um AWS::EC2::Instance recurso é “Lista de valores de string”, então ["sg-aaaaaaaa"] está ok (com colchetes), mas “sg-aaaaaaaa” não é (sem colchetes).

  Para obter mais informações, consulte a Referência de tipos de recursos e propriedades da AWS.
- Configure seus CloudFormation modelos personalizados para usar parâmetros definidos na CT de CloudFormation ingestão do AMS — Ao configurar seu CloudFormation modelo para usar parâmetros definidos na CT de CloudFormation ingestão do AMS, você pode reutilizar o CloudFormation modelo para criar pilhas semelhantes enviando-o com valores de parâmetros alterados na entrada do CT com o Gerenciamento | Pilha personalizada | Pilha a partir do CloudFormation modelo | Atualizar CT (ct-361tlo1k7339x). Para obter um exemplo, consulte CloudFormation Exemplos de ingestão: definição de recursos.
- Os endpoints de bucket do Amazon S3 com uma URL pré-assinada não podem expirar — Se você estiver usando um endpoint de bucket do Amazon S3 com uma URL pré-assinada, verifique se a URL pré-assinada do Amazon S3 não expirou. Uma RFC CloudFormation de ingestão enviada com uma URL de bucket Amazon S3 pré-assinada expirada é rejeitada.
- A condição de espera requer lógica de sinal — a condição de espera é usada para coordenar a criação de recursos da pilha com ações de configuração externas à criação da pilha. Se você usa o recurso Wait Condition no modelo, CloudFormation espera por um sinal de sucesso e marca a criação da pilha como uma falha se o número de sinais de sucesso não for gerado. Você precisa ter uma lógica para o sinal se usar o recurso Wait Condition. Para obter mais informações, consulte Criação de condições de espera em um modelo.

Práticas recomendadas

A seguir estão algumas das melhores práticas que você pode usar para migrar recursos usando o processo de CloudFormation ingestão do AMS:

- Envie o IAM e outros recursos relacionados à política em uma CT — Se você puder usar recursos automatizados, CTs como o CloudFormation Ingest, para implantar funções do IAM, recomendamos que você faça isso. Em outros casos, o AMS recomenda que você reúna todos os recursos do IAM ou outros recursos relacionados à política e os envie em um único Gerenciamento | Outro | Outro | Criar tipo de alteração (ct-1e1xtak34nx76). Por exemplo, combine todas as funções necessárias do IAM, perfis de EC2 instância do IAM Amazon, atualizações de políticas do IAM para funções existentes do IAM, políticas de bucket do Amazon S3, políticas do Amazon SNS/Amazon SQS e assim por diante, e envie uma RFC ct-1e1xtak34nx76 para que esses recursos preexistentes possam ser simplesmente referenciados nos futuros modelos de ingestão. CloudFormation
- EC2 as instâncias são inicializadas e unidas com sucesso ao domínio — isso é feito automaticamente como uma prática recomendada. Para garantir que as EC2 instâncias da Amazon lançadas por meio de uma pilha de CloudFormation ingestão sejam inicializadas e ingressem no domínio com sucesso, o AMS inclui um CreationPolicy e um para um recurso de grupo do UpdatePolicy Auto Scaling (ou seja, se essas políticas ainda não existirem).
- O parâmetro da instância de banco de dados do Amazon RDS deve ser especificado — Ao criar um banco de dados do Amazon RDS via CloudFormation ingestão, você deve especificar o `DBSnapshotIdentifier` parâmetro para restaurar a partir de um DB snapshot anterior. Isso é necessário porque o CloudFormation ingest atualmente não processa dados confidenciais.

Para obter um exemplo de como usar um CloudFormation modelo para ingestão de CloudFormation modelos do AMS, consulteCloudFormation Ingerir: exemplos.

Validação do modelo

Você pode autovalidar seu CloudFormation modelo antes de enviá-lo ao AMS.

Os modelos enviados ao AMS CloudFormation ingest são validados para garantir que sejam implantados com segurança em uma conta do AMS. O processo de validação verifica o seguinte:

- Recursos compatíveis — Somente os recursos CloudFormation suportados pela ingestão do AMS são usados. Para obter mais informações, consulte Recursos compatíveis.
- Compatível AMIs — A AMI no modelo é uma AMI compatível com a AMS. Para obter informações sobre o AMS AMIs, consulteImagens de máquinas AMS Amazon (AMIs).
- Sub-rede do AMS Shared Services — O modelo não tenta iniciar recursos na sub-rede do AMS Shared Services.

- Políticas de recursos — Não há políticas de recursos excessivamente permissivas, como uma política de bucket do S3 que possa ser lida ou gravada publicamente. O AMS não permite a entrada de buckets S3 legíveis ou graváveis publicamente. Contas da AWS

Valide com o CloudFormation Linter

Você pode autovalidar seu CloudFormation modelo antes de enviá-lo ao AMS usando a CloudFormation ferramenta Linter.

A ferramenta CloudFormation Linter é a melhor maneira de validar seu CloudFormation modelo, pois fornece validação para resource/property nomes, tipos de dados e funções. Para obter mais informações, consulte cfn-python-lintaws-cloudformation/.

A saída CloudFormation Linter do modelo mostrado anteriormente é a seguinte:

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

Para auxiliar na validação off-line de CloudFormation modelos, o AMS desenvolveu um conjunto de regras de validação personalizadas conectáveis para a ferramenta CloudFormation Linter. Eles estão localizados na página Recursos para desenvolvedores do console AMS.

Siga estas etapas para usar scripts de validação CloudFormation de pré-ingestão:

1. Instale a ferramenta CloudFormation Linter. Para obter instruções de instalação, consulte aws-cloudformation/cfn-lint.
2. Baixe um arquivo.zip com scripts de validação:

   Regras personalizadas do CFN Lint.
3. Descompacte as regras anexadas em um diretório de sua escolha.
4. Valide seu CloudFormation modelo executando o seguinte comando:

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

CloudFormation pilha de ingestão: exemplos de validadores CFN

Esses exemplos podem ajudar você a preparar seu modelo para uma ingestão bem-sucedida.

Validação de formato

Verifique se o modelo contém uma seção “Recursos” e se todos os recursos definidos nele têm um valor de “Tipo”.

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

Valide se as chaves raiz do modelo são permitidas. As chaves raiz permitidas são:

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

Validação manual da automação gerenciada

Se o modelo contiver os seguintes recursos, a validação automática falhará e você precisará de uma revisão manual.

As políticas mostradas são áreas de alto risco do ponto de vista da segurança. Por exemplo, uma política de bucket do S3 que permite que qualquer pessoa, exceto usuários ou grupos específicos, crie objetos ou escreva permissões é extremamente perigosa. Portanto, validamos as políticas e aprovamos ou negamos com base no conteúdo, e essas políticas não podem ser criadas automaticamente. Estamos investigando possíveis abordagens para resolver esse problema.

Atualmente, não temos validação automática dos seguintes recursos.

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

Validação de parâmetros

Valide isso se um parâmetro do modelo não tiver um valor fornecido, ele deverá ter um valor padrão.

Validação de atributos de recursos

Verificação de atributos obrigatória: certos atributos devem existir para determinados tipos de recursos.

- “VPCOptions" deve existir em `AWS::OpenSearch::Domain`
- “CludsterSubnetGroupName" deve existir em `AWS::Redshift::Cluster`

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

Verificação de atributos não permitidos: certos atributos *não* devem existir para determinados tipos de recursos.

- “SecretString" não deve existir em "AWS::SecretsManager::Secret”
- “MongoDbSettings" não deve existir em "AWS::DMS::Endpoint”

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

```
    "MongoDbSettings"
  ]
}
```

Verificação de parâmetros SSM: Para atributos na lista a seguir, os valores devem ser especificados por meio do Secrets Manager ou do Systems Manager Parameter Store (Secure String Parameter):

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

Alguns atributos devem estar em conformidade com determinados padrões; por exemplo, nomes de perfil de instância do IAM não devem começar com prefixos reservados do AMS, e o valor do atributo deve corresponder ao regex específico, conforme mostrado:

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

Validação de recursos

Somente os recursos da lista de permissões podem ser especificados no modelo; esses recursos estão descritos emRecursos compatíveis.

Pilhas EC2 e grupos de Auto Scaling ASGs () não são permitidos na mesma pilha devido a limitações de aplicação de patches.

Validação da regra de entrada do grupo de segurança

- Para solicitações provenientes dos tipos de alteração CFN Ingest Create ou Stack Update CT:
  - Se (`IpProtocol`é tcp ou 6) AND (a porta é 80 ou 443), não há restrições em relação ao valor `CidrIP`
  - Caso contrário, `CidrIP` não pode ser 0.0.0.0/0
- Para solicitações provenientes do Service Catalog (produtos do Service Catalog):

- Além da validação do tipo de alteração do CFN Ingest Create ou Stack Update CT, a porta `management_ports` com o protocolo de entrada só `ip_protocols` pode ser acessada via: `allowed_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"]
  }
```

Limitações

Atualmente, os seguintes recursos e funcionalidades não são compatíveis com o processo de CloudFormation ingestão do AMS.

- YAML — Não suportado. Somente CloudFormation modelos baseados em JSON são compatíveis.
- Pilhas aninhadas — Em vez disso, arquitete sua infraestrutura de aplicativos para usar um único modelo. Ou, como alternativa, você pode usar a referência entre pilhas para separar recursos em várias pilhas em que um recurso depende de outro. Para obter mais informações, consulte Passo a passo: consulte Saídas de recursos em outro AWS Stack. CloudFormation
- CloudFormation conjuntos de pilhas — Não há suporte, devido a implicações de segurança.
- Criação de recursos do IAM usando CloudFormation modelos — Somente funções do IAM são suportadas, devido a implicações de segurança.
- Dados confidenciais — Não há suporte. Não inclua dados confidenciais no modelo ou nos valores dos parâmetros. Se você precisar referenciar dados confidenciais, use o Secrets Manager para armazenar e recuperar esses valores. Para obter informações sobre o uso dos segredos do AWS Secrets Managers em uma propriedade de recurso, consulte Como criar e recuperar segredos gerenciados no AWS Secrets Manager usando CloudFormation modelos da AWS e Usando referências dinâmicas para especificar valores de modelos.

Recursos compatíveis

Os seguintes recursos da AWS são suportados no processo de CloudFormation ingestão do AMS.

CloudFormation Ingest Stack: recursos compatíveis

O sistema operacional da instância deve ser suportado pela ingestão da carga de trabalho do AMS. Somente os recursos da AWS listados aqui são compatíveis.

- Amazon API Gateway
  - AWS::ApiGateway::Account
  - AWS::ApiGateway::ApiKey
  - AWS::ApiGateway::Authorizer
  - AWS::ApiGateway::BasePathMapeamento
  - 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::UsagePlanChave
  - AWS::ApiGateway::VpcLink
- Amazon API Gateway V2
  - AWS::ApiGatewayV2::Api
  - AWS::ApiGatewayV2::ApiGatewayManagedOverrides
  - AWS::ApiGatewayV2::ApiMapping
  - AWS::ApiGatewayV2::Authorizer
  - AWS::ApiGatewayV2::Deployment
  - AWS::ApiGatewayV2::DomainName

- AWS::ApiGatewayV2::Integration
- AWS::ApiGatewayV2::IntegrationResponse
- AWS::ApiGatewayV2::Model
- AWS::ApiGatewayV2::Route
- AWS::ApiGatewayV2::RouteResponse
- AWS::ApiGatewayV2::Stage
- AWS::ApiGatewayV2::VpcLink

- 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

- CloudWatch Registros da Amazon
  - 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::UserPoolCliente
  - AWS::Cognito::UserPoolDomínio
  - AWS::Cognito::UserPoolGrupo
  - AWS::Cognito::UserPoolIdentityProvider
  - AWS::Cognito::UserPoolResourceServer
  - AWS::Cognito::UserPoolRiskConfigurationAttachment
  - AWS::Cognito::UserPoolUICustomizationAnexo
  - AWS::Cognito::UserPoolUsuário
  - AWS::Cognito::UserPoolUserToGroupAttachment
- Amazon DocumentDB
  - AWS::DocBanco de dados:: DBCluster
  - AWS::DocBanco de dados:: DBCluster ParameterGroup
  - AWS::DocBanco de dados:: DBInstance
  - AWS::DocDB:: DBSubnet Grupo
- 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::NetworkInterfaceAnexo
  - AWS::EC2::SecurityGroup
  - AWS::EC2::SecurityGroupEntrada
  - AWS::EC2::SecurityGroupSaída
  - AWS::EC2::LaunchTemplate
- AWS Batch
  - AWS::Batch::ComputeEnvironment
  - AWS::Batch::JobDefinition
  - AWS::Batch::JobQueue
- Amazon Elastic Container Registry (ECR)
  - AWS::ECR::Repository
- Amazon Elastic Container Service (ECS) (Fargate)
  - AWS::ECS::CapacityProvider
  - AWS::ECS::Cluster
  - AWS::ECS::PrimaryTaskConjunto
  - AWS::ECS::Service
  - AWS::ECS::TaskDefinition
  - AWS::ECS::TaskSet
- Amazon Elastic File System (EFS)
  - AWS::EFS::FileSystem
  - AWS::EFS::MountTarget
- Amazon ElastiCache
  - AWS::ElastiCache::CacheCluster
  - AWS::ElastiCache::ParameterGroup
  - AWS::ElastiCache::ReplicationGroup
  - AWS::ElastiCache::SecurityGroup
  - AWS::ElastiCache::SecurityGroupEntrada
  - AWS::ElastiCache::SubnetGroup

- Amazon EventBridge
  - AWS::Events::EventBus
  - AWS::Events::EventBusPolítica
  - 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:: Grupo DBParameter

  - AWS: :RDS:: Grupo DBSubnet
  - AWS::RDS::EventSubscription
  - AWS::RDS::OptionGroup
- Amazon Route 53
  - AWS::Route53::HealthCheck
  - AWS::Route53::HostedZone
  - AWS::Route53::RecordSet
  - AWS::Route53::RecordSetGrupo
  - AWS::Route53Resolver::ResolverRule
  - AWS::Route53Resolver::ResolverRuleAssociação
- Amazon S3
  - AWS::S3::Bucket
- Amazon Sagemaker
  - AWS::SageMaker::CodeRepository
  - AWS::SageMaker::Endpoint
  - AWS::SageMaker::EndpointConfig
  - AWS::SageMaker::Model
  - AWS::SageMaker::NotebookInstance
  - AWS::SageMaker::NotebookInstanceLifecycleConfig
  - AWS::SageMaker::Workteam
- Amazon Simple Email Service (SES)
  - AWS::SES::ConfigurationSet
  - AWS::SES::ConfigurationSetEventDestination
  - AWS::SES::ReceiptFilter
  - AWS::SES::ReceiptRule
  - AWS::SES::ReceiptRuleConjunto
  - 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
- Aplicativo AutoScaling
  - AWS::ApplicationAutoScaling::ScalableTarget
  - AWS::ApplicationAutoScaling::ScalingPolicy
- Amazon EC2 AutoScaling
  - AWS::AutoScaling::AutoScalingGrupo
  - 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::WaitConditionAlça
- 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::CustomActionTipo
  - AWS::CodePipeline::Pipeline
  - AWS::CodePipeline::Webhook
- AWS Database Migration Service (DMS)
  - AWS::DMS::Certificate
  - AWS::DMS::Endpoint
  - AWS::DMS::EventSubscription
  - AWS::DMS::ReplicationInstance
  - AWS::DMS::ReplicationSubnetGrupo
  - AWS::DMS::ReplicationTask

  A MongoDbSettings propriedade no AWS::DMS::Endpoint recurso não é permitida.

  As seguintes propriedades só são permitidas se forem resolvidas pelo AWS Secrets Manager: CertificatePem CertificateWallet propriedades no AWS::DMS::Certificate recurso e a propriedade Password no AWS::DMS::Endpoint recurso.
- AWS Elastic Load Balancing — Application Load Balancer/Network Load Balancer
  - AWS::ElasticLoadBalancingV2::Listener
  - AWS::ElasticLoadBalancingV2::ListenerCertificate
  - AWS::ElasticLoadBalancingV2::ListenerRule
  - AWS::ElasticLoadBalancingV2::LoadBalancer
  - AWS::ElasticLoadBalancingV2::TargetGroup
- AWS Elastic Load Balancing — Classic Load Balancer
  - AWS::ElasticLoadBalancing::LoadBalancer
- AWS Elemental MediaConvert
  - AWS::MediaConvert::JobTemplate
  - AWS::MediaConvert::Preset
  - AWS::MediaConvert::Queue
- AWS Elemental MediaStore

  - AWS::MediaStore::Container
- AWS Identity and Access Management (IAM)
  - AWS::IAM::Role
- AWS Managed Streaming para Apache Kafka (MSK)
  - AWS::MSK::Cluster
- AWS Glue
  - AWS::Glue::Classifier
  - AWS::Glue::Connection
  - AWS::Glue::Crawler
  - AWS::Glue::Database
  - AWS::Glue::DataCatalogEncryptionSettings
  - AWS::Glue::DevEndpoint
  - AWS::Glue::Job
  - AWS::Glue::MLTransform
  - AWS::Glue::Partition
  - AWS::Glue::SecurityConfiguration
  - AWS::Glue::Table
  - AWS::Glue::Trigger
  - AWS::Glue::Workflow
- Serviço de gerenciamento de chaves da AWS (KMS)
  - AWS::KMS::Key
  - AWS::KMS::Alias
- AWS Lake Formation
  - AWS::LakeFormation::DataLakeConfigurações
  - AWS::LakeFormation::Permissions
  - AWS::LakeFormation::Resource
- AWS Lambda
  - AWS::Lambda::Alias
  - AWS::Lambda::EventInvokeConfig
  - AWS::Lambda::EventSourceMapeamento

- AWS::Lambda::Function
- AWS::Lambda::LayerVersion
- AWS::Lambda::LayerVersionPermissão
- AWS::Lambda::Permission
- AWS::Lambda::Version

- Amazon Redshift
  - AWS::Redshift::Cluster
  - AWS::Redshift::ClusterParameterGrupo
  - AWS::Redshift::ClusterSubnetGrupo
- AWS Secrets Manager
  - AWS::SecretsManager::ResourcePolicy
  - AWS::SecretsManager::RotationSchedule
  - AWS::SecretsManager::Secret
  - AWS::SecretsManager::SecretTargetAnexo
- AWS Security Hub
  - AWS::SecurityHub::Hub
- AWS Step Functions
  - AWS::StepFunctions::Activity
  - AWS::StepFunctions::StateMachine
- AWS Systems Manager (SSM)
  - AWS::SSM::Parameter
- Amazon CloudWatch Synthetics
  - AWS::Synthetics::Canary
- Família AWS Transfer
  - AWS::Transfer::Server
  - AWS::Transfer::User
- AWS WAF
  - AWS::WAF::ByteMatchConjunto
  - AWS: :WAF:: IPSet
  - AWS::WAF::Rule

  - AWS::WAF::SizeConstraintConjunto
  - AWS::WAF::SqlInjectionMatchSet
  - AWS::WAF::WebACL
  - AWS::WAF::XssMatchConjunto
- AWS WAF Regional
  - AWS::WAFRegional::ByteMatchConjunto
  - AWS::WAFRegional::GeoMatchConjunto
  - AWS:WAFRegional:: IPSet
  - AWS::WAFRegional::RateBasedRegra
  - AWS::WAFRegional::RegexPatternConjunto
  - AWS::WAFRegional::Rule
  - AWS::WAFRegional::SizeConstraintConjunto
  - AWS::WAFRegional::SqlInjectionMatchSet
  - AWS::WAFRegional::WebACL
  - AWS::WAFRegional::WebACLAssociation
  - AWS::WAFRegional::XssMatchConjunto
- AWS WAFv2
  - AWS:WAFv2:: IPSet
  - AWS::WAFv2::RegexPatternConjunto
  - AWS::WAFv2::RuleGroup
  - AWS::WAFv2::WebACL
  - AWS::WAFv2::WebACLAssociation

CloudFormation Ingerir: exemplos

Veja aqui alguns exemplos detalhados de como usar a pilha Create com o tipo de alteração CloudFormation de modelo.

Para baixar um conjunto de CloudFormation modelos de amostra por Região da AWS, consulte Modelos de amostra.

Para obter informações de referência sobre CloudFormation recursos, consulte a Referência de tipos de recursos e propriedades da AWS. No entanto, o AMS oferece suporte a um conjunto menor de recursos, que estão descritos emCloudFormation Ingestão de AMS.

> **Note**
>
> O AMS recomenda que você reúna todos os recursos do IAM ou outros recursos relacionados à política e os envie em um único Gerenciamento | Outro | Outro | Criar tipo de alteração (ct-1e1xtak34nx76). Por exemplo, combine todas as funções do IAM, perfis de instância do IAM, atualizações de políticas do IAM para funções existentes do IAM, políticas de bucket do S3, políticas e assim por diante e, em seguida, envie uma RFC ct-1e1xtak34nx76 para que esses recursos preexistentes possam ser referenciados nos futuros modelos do CFN Ingest. SNS/SQS

Tópicos

- CloudFormation Exemplos de ingestão: definição de recursos
- CloudFormation Exemplos de ingestão: aplicativo Web de três camadas

CloudFormation Exemplos de ingestão: definição de recursos

Ao usar o AMS CloudFormation ingest, você personaliza um CloudFormation modelo e o envia ao AMS em um RFC com o tipo de alteração de CloudFormation ingestão (ct-36cn2avfrrj9v). Para criar um CloudFormation modelo que possa ser reutilizado várias vezes, você adiciona os parâmetros de configuração da pilha à entrada de execução do tipo de alteração de CloudFormation ingestão em vez de codificá-los no modelo. CloudFormation O maior benefício é que você pode reutilizar o modelo.

O esquema de entrada do tipo de alteração de CloudFormation ingestão do AMS permite que você escolha até sessenta parâmetros em um CloudFormation modelo e forneça valores personalizados.

Este exemplo mostra como definir uma propriedade de recurso, que pode ser usada em vários CloudFormation modelos, como um parâmetro na CT de CloudFormation ingestão do AMS. Os exemplos nesta seção mostram especificamente o uso de tópicos do SNS.

Tópicos

- Exemplo 1: codifique a TopicName propriedade do CloudFormation SNSTopic recurso

- Exemplo 2: Use um SNSTopic recurso para referenciar um parâmetro no tipo de alteração do AMS
- Exemplo 3: Crie um tópico do SNS enviando um arquivo de parâmetros de execução JSON com o tipo de alteração de ingestão do AMS
- Exemplo 4: Enviar um novo tipo de alteração que faça referência ao mesmo CloudFormation modelo
- Exemplo 5: Use os valores de parâmetros padrão no CloudFormation modelo

Exemplo 1: codifique a **`TopicName`** propriedade do CloudFormation SNSTopic recurso

Neste exemplo, você codifica a `TopicName` propriedade do CloudFormation SNSTopic recurso no CloudFormation modelo. Observe que a `Parameters` seção está vazia.

Para ter um CloudFormation modelo que permita alterar o valor do SNSTopic nome de uma nova pilha sem precisar criar um novo CloudFormation modelo, você pode usar a `Parameters` seção AMS do tipo de alteração de CloudFormation ingestão para fazer essa configuração. Ao fazer isso, você usa o mesmo CloudFormation modelo posteriormente para criar uma nova pilha com um `SNSTopic` nome diferente.

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

Exemplo 2: Use um SNSTopic recurso para referenciar um parâmetro no tipo de alteração do AMS

Neste exemplo, você usa uma `TopicName` propriedade de `SNSTopic` recurso definida no CloudFormation modelo para fazer referência a `Parameter` no tipo de alteração do AMS.

```
{
```

```
  "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"}
      }
    }
  }
}
```

Exemplo 3: Crie um tópico do SNS enviando um arquivo de parâmetros de execução JSON com o tipo de alteração de ingestão do AMS

Neste exemplo, você envia um arquivo de parâmetros de execução JSON com a CT de ingestão do AMS que cria o tópico SNS. `TopicName` O tópico SNS deve ser definido no CloudFormation modelo da forma modificável mostrada neste exemplo.

```
{
  "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
}
```

Exemplo 4: Enviar um novo tipo de alteração que faça referência ao mesmo CloudFormation modelo

Esse exemplo de JSON altera o `TopicName` valor do SNS sem fazer nenhuma alteração no CloudFormation modelo. Em vez disso, você envia um novo tipo de alteração Deployment | Inestion | Stack from CloudFormation Template | Create que faça referência ao mesmo modelo CFN.

```
{
  "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
}
```

Exemplo 5: Use os valores de parâmetros padrão no CloudFormation modelo

Neste exemplo, o SNS `TopicName` = 'MyTopicName' é criado porque nenhum `TopicName` valor foi fornecido no parâmetro de `Parameters` execução. Se você não fornecer `Parameters` definições, os valores de parâmetros padrão no CloudFormation modelo serão usados.

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

CloudFormation Exemplos de ingestão: aplicativo Web de três camadas

Ingira um CloudFormation modelo para um aplicativo Web padrão de três camadas.

Isso inclui um Application Load Balancer, um grupo-alvo do Application Load Balancer, um grupo de Auto Scaling, um modelo de lançamento do grupo Auto Scaling, o Amazon Relational Database Service (RDS for SQL Server) com um banco de dados MySQL, um repositório de parâmetros SSM e o Secrets Manager. AWS AWS Reserve de 30 a 60 minutos para analisar este exemplo.

Pré-requisitos

- Crie um segredo contendo um nome de usuário e senha com os valores correspondentes usando o AWS Secrets Manager. Você pode consultar esse exemplo de modelo JSON (arquivo zip) que contém o nome `ams-shared/myapp/dev/dbsecrets` secreto e substituí-lo pelo seu nome secreto. Para obter informações sobre como usar o AWS Secrets Manager com o AMS, consulteUsando o AWS Secrets Manager com recursos do AMS.
- Configure os parâmetros necessários no AWS SSM Parameter Store (PS). Neste exemplo, a `VPCId` e `Subnet-Id` das sub-redes pública e privada é armazenada no SSM PS em caminhos como`/app/DemoApp/PublicSubnet1a`,`PublicSubnet1c`, e. `PrivateSubnet1a` `PrivateSubnet1c` `VPCCidr` Atualize os caminhos e os nomes e valores dos parâmetros de acordo com suas necessidades.
- Crie uma função de instância IAM Amazon EC2 com permissões de leitura para os caminhos AWS Secrets Manager e SSM Parameter Store (a função do IAM criada e usada nesses exemplos é). `customer-ec2_secrets_manager_instance_profile` Se você criar IAM-standard políticas como função de perfil de instância, o nome da função deverá começar com`customer-`. Para criar uma nova função do IAM (você pode nomeá-la ou outra coisa)`customer-ec2_secrets_manager_instance_profile`, use o tipo de alteração do AMS Management | Applications | IAM instance profile | Create (ct-0ixp4ch2tiu04) CT e anexe as políticas necessárias. Você pode revisar as políticas padrão do AMS IAM `customer_secrets_manager_policy` `ecustomer_systemsmanager_parameterstore_policy`, no console do AWS IAM, para serem usadas como estão ou como referência.

Ingerir um CloudFormation modelo para um aplicativo Web padrão de três camadas

1. Faça upload do modelo CloudFormation JSON de amostra anexado como um arquivo zip, 3-tier-cfn-ingest.zip, em um bucket do S3 e gere uma URL assinada do S3 para usar no CFN Ingest RFC. Para obter mais informações, consulte presign. O modelo CFN também pode estar copy/pasted no CFN Ingest RFC quando você envia o RFC por meio do console AMS.
2. Crie um RFC CloudFormation de ingestão (Implantação | Ingestão | Pilha a partir do CloudFormation modelo | Criar (ct-36cn2avfrrj9v)), por meio do console do AMS ou da CLI do AMS. O processo de automação de CloudFormation ingestão valida o CloudFormation

modelo para garantir que ele tenha AMS-supported recursos válidos e cumpra os padrões de segurança.

- Usando o console - Para o tipo de alteração, selecione Implantação -> Ingestão -> Pilha do CloudFormation modelo -> Criar e, em seguida, adicione os seguintes parâmetros como exemplo (observe que o padrão para MultiAZDatabase é falso):

```
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"
```

- Usando o AWS CLI - Para obter detalhes sobre a criação de RFCs usando o AWS CLI, consulte Criação de RFCs. Por exemplo, execute o comando a seguir:

```
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
```

Encontre o URL do Application Load Balancer na saída de execução do CloudFormation RFC para acessar o site. Para obter informações sobre como acessar recursos, consulte Como acessar instâncias.

Crie uma CloudFormation pilha de ingestão

Criação de uma pilha de CloudFormation ingestão usando o console

Para criar uma pilha CloudFormation de ingestão usando o console

1. Navegue até a página Criar RFC: No painel de navegação esquerdo do console AMS, clique em RFCs para abrir a página da lista de RFCs e clique em Criar RFC.
2. Escolha um tipo de alteração (CT) popular na visualização padrão Procurar tipos de alteração ou selecione um CT na visualização Escolher por categoria.
   - Navegue por tipo de alteração: você pode clicar em um CT popular na área Criação rápida para abrir imediatamente a página Executar RFC. Observe que você não pode escolher uma versão mais antiga do CT com a criação rápida.

     Para classificar CTs, use a área Todos os tipos de alteração na visualização Cartão ou Tabela. Em qualquer visualização, selecione um CT e clique em Criar RFC para abrir a página Executar RFC. Se aplicável, a opção Criar com uma versão mais antiga aparece ao lado do botão Criar RFC.
   - Escolha por categoria: selecione uma categoria, subcategoria, item e operação e a caixa de detalhes do CT é aberta com a opção de Criar com a versão mais antiga, se aplicável. Clique em Criar RFC para abrir a página Executar RFC.
3. Na página Executar RFC, abra a área do nome do CT para ver a caixa de detalhes do CT. Um assunto é obrigatório (ele é preenchido para você se você escolher seu CT na visualização Procurar tipos de alteração). Abra a área Configuração adicional para adicionar informações sobre o RFC.

   Na área Configuração de execução, use as listas suspensas disponíveis ou insira valores para os parâmetros necessários. Para configurar parâmetros de execução opcionais, abra a área Configuração adicional.
4. Ao terminar, clique em Executar. Se não houver erros, a página RFC criada com êxito será exibida com os detalhes da RFC enviados e a saída inicial da Run.
5. Abra a área de parâmetros de execução para ver as configurações que você enviou. Atualize a página para atualizar o status de execução do RFC. Opcionalmente, cancele a RFC ou crie uma cópia dela com as opções na parte superior da página.

Criação de uma pilha de CloudFormation ingestão usando a CLI

Para criar uma pilha CloudFormation de ingestão usando a CLI

1. Use o Inline Create (você emite um `create-rfc` comando com todos os parâmetros RFC e de execução incluídos) ou o Template Create (você cria dois arquivos JSON, um para os parâmetros RFC e outro para os parâmetros de execução) e emita o `create-rfc` comando com os dois arquivos como entrada. Ambos os métodos são descritos aqui.
2. Envie o `aws amscm submit-rfc --rfc-id` *`ID`* comando RFC: com o ID RFC retornado.

   Monitore o `aws amscm get-rfc --rfc-id` *`ID`* comando RFC:.

Para verificar a versão do tipo de alteração, use este comando:

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

> **Note**
>
> Você pode usar qualquer `CreateRfc` parâmetro com qualquer RFC, independentemente de eles fazerem parte ou não do esquema do tipo de alteração. Por exemplo, para receber notificações quando o status da RFC mudar, adicione essa linha `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` à parte dos parâmetros RFC da solicitação (não aos parâmetros de execução). Para obter uma lista de todos os CreateRfc parâmetros, consulte a Referência da API de gerenciamento de alterações do AMS.

1. Prepare o CloudFormation modelo que você usará para criar a pilha e faça o upload para seu bucket do S3. Para obter detalhes importantes, consulte as diretrizes CloudFormation de ingestão, as melhores práticas e as limitações da AWS.
2. Crie e envie o RFC para o AMS:

   - Crie e salve o arquivo JSON dos parâmetros de execução, inclua os parâmetros do CloudFormation modelo que você deseja. O exemplo a seguir o nomeia CreateCfnParams.json.

     Exemplo de CreateCfnParams.json arquivo de pilha de aplicativos Web:

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

Exemplo de CreateCfnParams.json arquivo de tópico do SNS:

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

3. Crie e salve o arquivo JSON dos parâmetros RFC com o conteúdo a seguir. O exemplo a seguir nomeia esse CreateCfnRfc.json arquivo:

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

4. Crie o RFC, especificando o CreateCfnRfc arquivo e o CreateCfnParams arquivo:

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

Você recebe a ID da nova RFC na resposta e pode usá-la para enviar e monitorar a RFC. Até que você o envie, o RFC permanece no estado de edição e não inicia.

Dicas

> **Note**
>
> Esse tipo de alteração está na versão 2.0 e é automatizado (não executado manualmente). Isso permite que a execução do CT seja mais rápida e, um novo parâmetro CloudFormationTemplate, permite colar no RFC um CloudFormation modelo personalizado. Além disso, nesta versão, não anexamos os grupos de segurança padrão do AMS se você especificar seus próprios grupos de segurança. Se você não especificar seus próprios grupos de segurança na solicitação, o AMS anexará os grupos de segurança padrão do AMS. No CFN Ingest v1.0, sempre anexamos os grupos de segurança padrão do AMS, independentemente de você ter fornecido ou não seus próprios grupos de segurança.
> O AMS habilitou 17 Self-Provisioned serviços AMS para uso nesse tipo de alteração. Para obter informações sobre recursos compatíveis, consulte CloudFormation Ingest Stack: Recursos suportados.

> **Note**
>
> A versão 2.0 aceita um endpoint do S3 que não seja um URL pré-assinado.

Se você usar a versão anterior desse CT, o valor do CloudFormationTemplateS3Endpoint parâmetro deverá ser um URL pré-assinado.
Exemplo de comando para gerar um URL de bucket S3 pré-assinado ()Mac/Linux:

```
export S3_PRESIGNED_URL=$(aws s3 presign DASHDASHexpires-in 86400
 s3://BUCKET_NAME/CFN_TEMPLATE.json)
```

Exemplo de comando para gerar uma URL de bucket S3 pré-assinada (Windows):

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

Consulte também Criação de Pre-Signed URLs para buckets do Amazon S3.

Note

Se o bucket do S3 existir em uma conta do AMS, você deverá usar suas credenciais do AMS para esse comando. Por exemplo, talvez seja necessário acrescentar `--profile saml` depois de obter suas credenciais do AMS AWS Security Token Service (AWS STS).

Tipos de mudança relacionados:Aprovar um conjunto de alterações da CloudFormation pilha de ingestão, Atualizar CloudFormation pilha de ingestão

Para saber mais sobre a AWS CloudFormation, consulte AWS CloudFormation. Para ver os CloudFormation modelos, abra o AWS CloudFormation Template Reference.

Validando um CloudFormation ingerir

O modelo é validado para garantir que ele possa ser criado em uma conta do AMS. Se for aprovado na validação, ele será atualizado para incluir todos os recursos ou configurações necessários para que esteja em conformidade com o AMS. Isso inclui adicionar recursos, como CloudWatch alarmes da Amazon, para permitir que as operações do AMS monitorem a pilha.

A RFC será rejeitada se alguma das seguintes afirmações for verdadeira:

- A sintaxe RFC JSON está incorreta ou não segue o formato fornecido.
- O URL pré-assinado do bucket do S3 fornecido não é válido.

- O modelo não é uma CloudFormation sintaxe válida.
- O modelo não tem padrões definidos para todos os valores dos parâmetros.
- O modelo falha na validação do AMS. Para as etapas de validação do AMS, consulte as informações mais adiante neste tópico.

A RFC falhará se a CloudFormation pilha falhar na criação devido a um problema de criação de recursos.

Para saber mais sobre validação e validador de CFN, consulte Validação de modelos e pilha de CloudFormation ingestão: exemplos de validadores de CFN.

Atualizar CloudFormation pilha de ingestão

Atualização de uma pilha de CloudFormation ingestão usando o console

Para atualizar uma pilha CloudFormation de ingestão usando o console

1. Navegue até a página Criar RFC: No painel de navegação esquerdo do console AMS, clique em RFCs para abrir a página da lista de RFCs e clique em Criar RFC.
2. Escolha um tipo de alteração (CT) popular na visualização padrão Procurar tipos de alteração ou selecione um CT na visualização Escolher por categoria.
   - Navegue por tipo de alteração: você pode clicar em um CT popular na área Criação rápida para abrir imediatamente a página Executar RFC. Observe que você não pode escolher uma versão mais antiga do CT com a criação rápida.

     Para classificar CTs, use a área Todos os tipos de alteração na visualização Cartão ou Tabela. Em qualquer visualização, selecione um CT e clique em Criar RFC para abrir a página Executar RFC. Se aplicável, a opção Criar com uma versão mais antiga aparece ao lado do botão Criar RFC.
   - Escolha por categoria: selecione uma categoria, subcategoria, item e operação e a caixa de detalhes do CT é aberta com a opção de Criar com a versão mais antiga, se aplicável. Clique em Criar RFC para abrir a página Executar RFC.
3. Na página Executar RFC, abra a área do nome do CT para ver a caixa de detalhes do CT. Um assunto é obrigatório (ele é preenchido para você se você escolher seu CT na visualização Procurar tipos de alteração). Abra a área Configuração adicional para adicionar informações sobre o RFC.

Na área Configuração de execução, use as listas suspensas disponíveis ou insira valores para os parâmetros necessários. Para configurar parâmetros de execução opcionais, abra a área Configuração adicional.

4. Ao terminar, clique em Executar. Se não houver erros, a página RFC criada com êxito será exibida com os detalhes da RFC enviados e a saída inicial da Run.
5. Abra a área de parâmetros de execução para ver as configurações que você enviou. Atualize a página para atualizar o status de execução do RFC. Opcionalmente, cancele a RFC ou crie uma cópia dela com as opções na parte superior da página.

Atualização de uma pilha CloudFormation de ingestão usando a CLI

Para atualizar uma pilha CloudFormation de ingestão usando a CLI

1. Use o Inline Create (você emite um `create-rfc` comando com todos os parâmetros RFC e de execução incluídos) ou o Template Create (você cria dois arquivos JSON, um para os parâmetros RFC e outro para os parâmetros de execução) e emita o `create-rfc` comando com os dois arquivos como entrada. Ambos os métodos são descritos aqui.
2. Envie o `aws amscm submit-rfc --rfc-id` *`ID`* comando RFC: com o ID RFC retornado.

   Monitore o `aws amscm get-rfc --rfc-id` *`ID`* comando RFC:.

Para verificar a versão do tipo de alteração, use este comando:

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

> **Note**
>
> Você pode usar qualquer `CreateRfc` parâmetro com qualquer RFC, independentemente de eles fazerem parte ou não do esquema do tipo de alteração. Por exemplo, para receber notificações quando o status da RFC mudar, adicione essa linha `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` à parte dos parâmetros de RFC da solicitação (não aos parâmetros de execução). Para obter uma lista de todos os CreateRfc parâmetros, consulte a Referência da API de gerenciamento de alterações do AMS.

1. Prepare o CloudFormation modelo que você deseja usar para atualizar a pilha e faça o upload para seu bucket do S3. Para obter detalhes importantes, consulte as diretrizes CloudFormation de ingestão, as melhores práticas e as limitações da AWS.
2. Crie e envie o RFC para o AMS:

   - Crie e salve o arquivo JSON dos parâmetros de execução, inclua os parâmetros do CloudFormation modelo que você deseja. Este exemplo o nomeia UpdateCfnParams.json.

     UpdateCfnParams.json Arquivo de exemplo com atualizações de parâmetros em linha:

```
{
  "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
}
```

     UpdateCfnParams.json Arquivo de exemplo com o endpoint de bucket do S3 contendo um modelo atualizado CloudFormation :

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

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

3. Crie e salve o arquivo JSON dos parâmetros RFC com o conteúdo a seguir. Este exemplo o chama de UpdateCfnRfc.json arquivo.

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

4. Crie o RFC, especificando o UpdateCfnRfc arquivo e o UpdateCfnParams arquivo:

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

   Você recebe a ID da nova RFC na resposta e pode usá-la para enviar e monitorar a RFC. Até que você o envie, o RFC permanece no estado de edição e não inicia.

Dicas

- Esse tipo de alteração está agora na versão 2.0. As mudanças incluem a remoção do AutoApproveUpdateForResources parâmetro, que foi usado na versão 1.0 deste CT, e a adição de dois novos parâmetros: AutoApproveRiskyUpdates BypassDriftCheck e.
- Se o bucket do S3 existir em uma conta do AMS, você deverá usar suas credenciais do AMS para esse comando. Por exemplo, talvez seja necessário acrescentar `--profile saml` depois de obter suas credenciais do AMS AWS Security Token Service (AWS STS).
- Todos os `Parameter` valores dos recursos no CloudFormation modelo devem ter um valor, seja por meio de um valor padrão ou personalizado por meio da seção de parâmetros do CT. Você pode substituir o valor do parâmetro estruturando os recursos do CloudFormation modelo para

referenciar uma chave de parâmetros. Para ver exemplos que mostram como fazer, consulte CloudFormation ingest stack: exemplos do validador CFN.

IMPORTANTE: Os parâmetros ausentes não fornecidos explicitamente no formulário são padronizados para os valores atualmente definidos na pilha ou no modelo existente.

- Para obter uma lista dos serviços autoprovisionados que você pode adicionar usando o Ingest, consulte CloudFormation CloudFormation Ingest Stack: Recursos suportados.

  Para saber mais sobre isso CloudFormation, consulte AWS CloudFormation.

Validando um CloudFormation ingerir

O modelo é validado para garantir que ele possa ser criado em uma conta do AMS. Se for aprovado na validação, ele será atualizado para incluir todos os recursos ou configurações necessários para que esteja em conformidade com o AMS. Isso inclui adicionar recursos, como CloudWatch alarmes da Amazon, para permitir que as operações do AMS monitorem a pilha.

A RFC será rejeitada se alguma das seguintes afirmações for verdadeira:

- A sintaxe RFC JSON está incorreta ou não segue o formato fornecido.
- O URL pré-assinado do bucket do S3 fornecido não é válido.
- O modelo não é uma CloudFormation sintaxe válida.
- O modelo não tem padrões definidos para todos os valores dos parâmetros.
- O modelo falha na validação do AMS. Para as etapas de validação do AMS, consulte as informações mais adiante neste tópico.

A RFC falhará se a CloudFormation pilha falhar na criação devido a um problema de criação de recursos.

Para saber mais sobre validação e validador de CFN, consulte Validação de modelos e pilha de CloudFormation ingestão: exemplos de validadores de CFN.

Aprovar um conjunto de alterações da CloudFormation pilha de ingestão

Aprovação e atualização de uma pilha CloudFormation de ingestão usando o console

Para aprovar e atualizar uma pilha CloudFormation de ingestão usando o console

1. Navegue até a página Criar RFC: No painel de navegação esquerdo do console AMS, clique em RFCs para abrir a página da lista de RFCs e clique em Criar RFC.
2. Escolha um tipo de alteração (CT) popular na visualização padrão Procurar tipos de alteração ou selecione um CT na visualização Escolher por categoria.
   - Navegue por tipo de alteração: você pode clicar em um CT popular na área Criação rápida para abrir imediatamente a página Executar RFC. Observe que você não pode escolher uma versão mais antiga do CT com a criação rápida.

     Para classificar CTs, use a área Todos os tipos de alteração na visualização Cartão ou Tabela. Em qualquer visualização, selecione um CT e clique em Criar RFC para abrir a página Executar RFC. Se aplicável, a opção Criar com uma versão mais antiga aparece ao lado do botão Criar RFC.
   - Escolha por categoria: selecione uma categoria, subcategoria, item e operação e a caixa de detalhes do CT é aberta com a opção de Criar com a versão mais antiga, se aplicável. Clique em Criar RFC para abrir a página Executar RFC.
3. Na página Executar RFC, abra a área do nome do CT para ver a caixa de detalhes do CT. Um assunto é obrigatório (ele é preenchido para você se você escolher seu CT na visualização Procurar tipos de alteração). Abra a área Configuração adicional para adicionar informações sobre o RFC.

   Na área Configuração de execução, use as listas suspensas disponíveis ou insira valores para os parâmetros necessários. Para configurar parâmetros de execução opcionais, abra a área Configuração adicional.
4. Ao terminar, clique em Executar. Se não houver erros, a página RFC criada com êxito será exibida com os detalhes da RFC enviados e a saída inicial da Run.
5. Abra a área de parâmetros de execução para ver as configurações que você enviou. Atualize a página para atualizar o status de execução do RFC. Opcionalmente, cancele a RFC ou crie uma cópia dela com as opções na parte superior da página.

Aprovação e atualização de uma pilha CloudFormation de ingestão usando a CLI

Para aprovar e atualizar uma pilha CloudFormation de ingestão usando a CLI

1. Use o Inline Create (você emite um `create-rfc` comando com todos os parâmetros RFC e de execução incluídos) ou o Template Create (você cria dois arquivos JSON, um para os parâmetros

RFC e outro para os parâmetros de execução) e emita o `create-rfc` comando com os dois arquivos como entrada. Ambos os métodos são descritos aqui.

2. Envie o `aws amscm submit-rfc --rfc-id` *ID* comando RFC: com o ID RFC retornado.

   Monitore o `aws amscm get-rfc --rfc-id` *ID* comando RFC:.

Para verificar a versão do tipo de alteração, use este comando:

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

> **Note**
>
> Você pode usar qualquer `CreateRfc` parâmetro com qualquer RFC, independentemente de eles fazerem parte ou não do esquema do tipo de alteração. Por exemplo, para receber notificações quando o status da RFC mudar, adicione essa linha `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` à parte dos parâmetros RFC da solicitação (não aos parâmetros de execução). Para obter uma lista de todos os CreateRfc parâmetros, consulte a Referência da API de gerenciamento de alterações do AMS.

1. Envie o esquema JSON dos parâmetros de execução para esse tipo de alteração em um arquivo na sua pasta atual. Este exemplo o nomeia 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. Modifique e salve o esquema da seguinte forma:

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

Dicas

> **Note**
>
> Se houver vários recursos em uma pilha e você quiser excluir somente um subconjunto dos recursos da pilha, use o CloudFormation Update CT; consulte CloudFormation Ingest Stack: Updating. Você também pode enviar um caso de solicitação de serviço e os engenheiros da AMS podem ajudá-lo a criar o conjunto de alterações, se necessário.

Para saber mais sobre AWS CloudFormation, consulte AWS CloudFormation.

Atualizar CloudFormation proteção de terminação de pilhas

Atualizando um CloudFormation pilha de proteção de terminação com o console

O seguinte mostra esse tipo de alteração no console AMS.

Como funciona:

1. Navegue até a página Criar RFC: No painel de navegação esquerdo do console AMS, clique em RFCs para abrir a página da lista de RFCs e clique em Criar RFC.
2. Escolha um tipo de alteração (CT) popular na visualização padrão Procurar tipos de alteração ou selecione um CT na visualização Escolher por categoria.
   - Navegue por tipo de alteração: você pode clicar em um CT popular na área Criação rápida para abrir imediatamente a página Executar RFC. Observe que você não pode escolher uma versão mais antiga do CT com a criação rápida.

     Para classificar CTs, use a área Todos os tipos de alteração na visualização Cartão ou Tabela. Em qualquer visualização, selecione um CT e clique em Criar RFC para abrir a página Executar RFC. Se aplicável, a opção Criar com uma versão mais antiga aparece ao lado do botão Criar RFC.
   - Escolha por categoria: selecione uma categoria, subcategoria, item e operação e a caixa de detalhes do CT é aberta com a opção de Criar com a versão mais antiga, se aplicável. Clique em Criar RFC para abrir a página Executar RFC.
3. Na página Executar RFC, abra a área do nome do CT para ver a caixa de detalhes do CT. Um assunto é obrigatório (ele é preenchido para você se você escolher seu CT na visualização

Procurar tipos de alteração). Abra a área Configuração adicional para adicionar informações sobre o RFC.

Na área Configuração de execução, use as listas suspensas disponíveis ou insira valores para os parâmetros necessários. Para configurar parâmetros de execução opcionais, abra a área Configuração adicional.

4. Ao terminar, clique em Executar. Se não houver erros, a página RFC criada com êxito será exibida com os detalhes da RFC enviados e a saída inicial da Run.
5. Abra a área de parâmetros de execução para ver as configurações que você enviou. Atualize a página para atualizar o status de execução do RFC. Opcionalmente, cancele a RFC ou crie uma cópia dela com as opções na parte superior da página.

Atualizando um CloudFormation proteção de terminação de pilha com a CLI

Como funciona:

1. Use o Inline Create (você emite um `create-rfc` comando com todos os parâmetros RFC e de execução incluídos) ou o Template Create (você cria dois arquivos JSON, um para os parâmetros RFC e outro para os parâmetros de execução) e emita o `create-rfc` comando com os dois arquivos como entrada. Ambos os métodos são descritos aqui.
2. Envie o `aws amscm submit-rfc --rfc-id` *`ID`* comando RFC: com o ID RFC retornado.

   Monitore o `aws amscm get-rfc --rfc-id` *`ID`* comando RFC:.

Para verificar a versão do tipo de alteração, use este comando:

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

> **Note**
>
> Você pode usar qualquer `CreateRfc` parâmetro com qualquer RFC, independentemente de eles fazerem parte ou não do esquema do tipo de alteração. Por exemplo, para receber notificações quando o status da RFC mudar, adicione essa linha `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` à parte dos parâmetros RFC da solicitação (não aos parâmetros de execução). Para obter uma

lista de todos os CreateRfc parâmetros, consulte a Referência da API de gerenciamento de alterações do AMS.

Especifique somente os parâmetros que você deseja alterar. Os parâmetros ausentes mantêm os valores existentes.

CRIAÇÃO EM LINHA:

Emita o comando create RFC com os parâmetros de execução fornecidos em linha (aspas de escape ao fornecer parâmetros de execução em linha) e, em seguida, envie a ID de RFC retornada. Por exemplo, você pode substituir o conteúdo por algo assim:

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

CRIAÇÃO DO MODELO:

1. Exiba os parâmetros de execução desse tipo de alteração em um arquivo JSON; este exemplo o nomeia EnableTermProCFNParams.json:

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

2. Modifique e salve o EnableTermProCFNParams arquivo, mantendo somente os parâmetros que você deseja alterar. Por exemplo, você pode substituir o conteúdo por algo assim:

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

```
}
```

3. Envie o modelo RFC para um arquivo em sua pasta atual; este exemplo o nomeia EnableTermProCFNRfc.json:

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

4. Modifique e salve o EnableTermProCFNRfc.json arquivo. Por exemplo, você pode substituir o conteúdo por algo assim:

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

5. Crie o RFC, especificando o EnableTermProCFNRfc arquivo e o EnableTermProCFNParams arquivo:

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

   Você recebe a ID da nova RFC na resposta e pode usá-la para enviar e monitorar a RFC. Até que você o envie, o RFC permanece no estado de edição e não inicia.

Dicas

> **Note**
>
> Há um CT relacionado ao Amazon EC2, EC2 stack: Atualizar a proteção contra encerramento.

Para saber mais sobre a proteção contra encerramento, consulte Protegendo uma pilha contra a exclusão.

Implantações automatizadas de IAM usando ingestão de CFN ou atualização de pilha no AMS CTs

Você pode usar esses tipos de alteração do AMS para implantar funções do IAM (o `AWS::IAM::Role` recurso) na zona de pouso com várias contas (MALZ) e na zona de destino de conta única (SALZ):

- Implantação | Ingestão | Pilha a partir do CloudFormation modelo | Criar (ct-36cn2avfrrj9v)
- Gerenciamento | Pilha personalizada | Pilha a partir do CloudFormation modelo | Atualização (ct-361tlo1k7339x)
- Gerenciamento | Pilha personalizada | Empilhar a partir do CloudFormation modelo | Aprovar e atualizar (ct-1404e21baa2ox)

Validações realizadas nas funções do IAM em seu modelo CFN:

- ManagedPolicyArns: O atributo não ManagedPolicyArnsdeve existir em`AWS::IAM::Role`. A validação não permite anexar políticas gerenciadas à função que está sendo provisionada. Em vez disso, as permissões para a função podem ser gerenciadas usando a política embutida por meio das Políticas da propriedade.
- PermissionsBoundary: A política usada para definir o limite de permissões para a função só pode ser a política gerenciada por vendedores do AMS:. `AWSManagedServices_IAM_PermissionsBoundary` Essa política atua como uma barreira que protege os recursos da infraestrutura do AMS de serem modificados usando a função que está sendo provisionada. Com esse limite de permissões padrão, os benefícios de segurança que o AMS oferece são preservados.

  O `AWSManagedServices_IAM_PermissionsBoundary` (o padrão) é obrigatório, sem ele, a solicitação é rejeitada.
- MaxSessionDuration: a duração máxima da sessão que pode ser definida para a função do IAM é de 1 a 4 horas. O padrão técnico do AMS exige que o cliente aceite o risco para uma duração de sessão superior a 4 horas.
- RoleName: os namespaces a seguir são preservados pelo AMS e não podem ser usados como prefixos de nome de função do IAM:

```
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
```

- Políticas: a política embutida na função do IAM só pode incluir um conjunto de ações do IAM que são pré-aprovadas pelo AMS. Esse é o limite superior de todas as ações do IAM permitidas para criar uma função do IAM com (política de controle). A política de controle consiste em:
  - Todas as ações na política AWS gerenciada ReadOnlyAccess que fornece acesso somente de leitura a todos os recursos Serviços da AWS
  - As ações a seguir, com a restrição de ações do S3 entre contas, ou seja, ações permitidas do S3, só podem ser executadas em recursos presentes na mesma conta da função que está sendo criada:

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

```
sns:Publish
```

Qualquer função do IAM criada ou atualizada por meio do CFN ingest pode permitir ações listadas nessa política de controle ou ações com escopo inferior (menos permissivo) às ações listadas na política de controle. Atualmente, permitimos essas ações seguras do IAM que podem ser categorizadas como ações somente para leitura, além das ações não somente para leitura mencionadas acima, que não podem ser realizadas por meio do padrão técnico do AMS CTs e são pré-aprovadas de acordo com o padrão técnico do AMS.

- AssumeRolePolicyDocument: as seguintes entidades são pré-aprovadas e podem ser incluídas na política de confiança para assumir a função que está sendo criada:
  - Qualquer entidade do IAM (função, usuário, usuário raiz, sessão de função assumida pelo STS) na mesma conta pode assumir a função.
  - O seguinte Serviços da AWS pode assumir a função:

```
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
```

- O provedor de SAML na mesma conta pode assumir a função. Atualmente, o único nome de provedor SAML compatível é `customer-saml`.

Se uma ou mais validações falharem, a RFC será rejeitada. Um exemplo de motivo de rejeição de RFC tem a seguinte aparência:

```
{"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"}
```

Se você precisar de ajuda com uma falha na validação ou execução da RFC, use a correspondência da RFC para entrar em contato com o AMS. Para obter instruções, consulte Correspondência e anexo RFC (console). Para qualquer outra dúvida, envie uma solicitação de serviço. Para obter instruções, consulte Como criar uma solicitação de serviço.

> **Note**
>
> Atualmente, não aplicamos nenhuma das melhores práticas do IAM como parte de nossas validações do IAM. Para ver as melhores práticas do IAM, consulte Melhores práticas de segurança no IAM.

Criação de funções do IAM com ações mais permissivas ou aplicação das melhores práticas do IAM

Crie suas entidades do IAM com os seguintes tipos de alteração manual:

- Implantação | Componentes avançados da pilha | Identity and Access Management (IAM) | Criar entidade ou política (ct-3dpd8mdd9jn1r)
- Gerenciamento | Componentes avançados da pilha | Identity and Access Management (IAM) | Atualizar entidade ou política (ct-27tuth19k52b4)

Recomendamos que você leia e compreenda nossos padrões técnicos antes de preencher este manual RFCs. Para obter acesso, consulte Como acessar os padrões técnicos.

> **Note**
>
> Cada função do IAM criada diretamente com esses tipos de alteração manual pertence à sua própria pilha individual e não reside na mesma pilha em que os outros recursos de infraestrutura são criados por meio do CFN Ingest CT.

Atualização de funções do IAM criadas com a ingestão de CFN por meio de tipos de alteração manual quando as atualizações não podem ser feitas por meio de tipos de alteração automatizados

Use os componentes de pilha Management | Advanced Stack | Identity and Access Management (IAM) | Atualizar tipo de alteração de entidade ou política (ct-27tuth19k52b4).

> **Important**
>
> As atualizações nas funções do IAM por meio da CT manual não são refletidas nos modelos de pilha do CFN e causam o desvio da pilha. Depois que a função é atualizada por meio de uma solicitação manual para um estado que não passa em nossas validações, ela não pode ser atualizada novamente usando o Stack Update CT (ct-361tlo1k7339x), desde que continue não em conformidade com nossas validações. A atualização CT só pode ser usada se o modelo de pilha CFN estiver em conformidade com nossas validações. No entanto, a pilha ainda pode ser atualizada por meio do Stack Update CT (ct-361tlo1k7339x), desde que o recurso do IAM que não esteja em conformidade com nossas validações não esteja sendo atualizado e o modelo CFN seja aprovado em nossas validações.

Excluindo suas funções do IAM criadas por meio AWS CloudFormation do ingest

Se você quiser excluir toda a pilha, use o seguinte tipo de alteração automática de Excluir pilha. Para obter instruções, consulte Delete Stack:

- ID do tipo de alteração: ct-0q0bic0ywqk6c
- Classificação: Gerenciamento | Pilhas padrão | Pilha | Excluir e gerenciar | Componentes avançados da pilha | Pilha | Excluir

Se quiser excluir uma função do IAM sem excluir toda a pilha, você pode remover a função do IAM do CloudFormation modelo e usar o modelo atualizado como entrada para o tipo de alteração automatizada da atualização da pilha:

- ID do tipo de alteração: ct-361tlo1k7339x
- Classificação: Gerenciamento | Pilha personalizada | Pilha a partir do CloudFormation modelo | Atualização

Para obter instruções, consulte Atualizar pilha AWS CloudFormation de ingestão.

CodeDeploy solicitações

Você pode usar CodeDeploy a AWS para criar contêineres de aplicativos que podem ser implantados por meio de um grupo de CodeDeploy aplicativos. Para obter mais informações sobre isso CodeDeploy, consulte a CodeDeploy documentação da AWS.

Trabalhar com a AWS CodeDeploy envolve o seguinte processo:

1. Crie um CodeDeploy aplicativo. O CodeDeploy aplicativo é um nome ou contêiner usado CodeDeploy para garantir que a revisão, a configuração de implantação e o grupo de implantação corretos sejam referenciados durante uma implantação.
2. Crie um grupo CodeDeploy de implantação. Um grupo de CodeDeploy implantação define um conjunto de instâncias individuais destinadas a uma implantação. O AMS tem um tipo de alteração separado para grupos de CodeDeploy implantação EC2.
3. Implante o CodeDeploy aplicativo por meio do grupo CodeDeploy de implantação.

CodeDeploy aplicação

Crie ou implante CodeDeploy aplicativos.

Criar CodeDeploy aplicativo

Criando um CodeDeploy aplicativo com o console

Como funciona:

1. Navegue até a página Criar RFC: No painel de navegação esquerdo do console AMS, clique em RFCs para abrir a página de lista de RFCs e, em seguida, clique em Criar RFC.
2. Escolha um tipo de alteração popular (CT) na visualização padrão Procurar tipos de alteração ou selecione uma CT na visualização Escolher por categoria.
   - Navegar por tipo de alteração: você pode clicar em um CT popular na área de criação rápida para abrir imediatamente a página Executar RFC. Observe que você não pode escolher uma versão mais antiga do CT com a criação rápida.

     Para classificar CTs, use a área Todos os tipos de alteração na visualização em Cartão ou Tabela. Em qualquer exibição, selecione uma CT e clique em Criar RFC para abrir a página Executar RFC. Se aplicável, a opção Criar com uma versão mais antiga aparece ao lado do botão Criar RFC.
   - Escolha por categoria: selecione uma categoria, subcategoria, item e operação e a caixa de detalhes do CT será aberta com a opção Criar com uma versão mais antiga, se aplicável. Clique em Criar RFC para abrir a página Executar RFC.
3. Na página Executar RFC, abra a área do nome do CT para ver a caixa de detalhes do CT. É necessário um Assunto (preenchido se você escolher seu CT na visualização Procurar tipos de alteração). Abra a área Configuração adicional para adicionar informações sobre o RFC.

   Na área Configuração de execução, use as listas suspensas disponíveis ou insira valores para os parâmetros necessários. Para configurar parâmetros de execução opcionais, abra a área Configuração adicional.
4. Ao terminar, clique em Executar. Se não houver erros, a página RFC criada com sucesso será exibida com os detalhes da RFC enviada e a saída inicial de execução.
5. Abra a área Parâmetros de execução para ver as configurações que você enviou. Atualize a página para atualizar o status de execução do RFC. Opcionalmente, cancele a RFC ou crie uma cópia dela com as opções na parte superior da página.

Criando um CodeDeploy aplicativo com a CLI

Como funciona:

1. Use o Inline Create (você emite um `create-rfc` comando com todos os parâmetros de RFC e execução incluídos) ou o Template Create (você cria dois arquivos JSON, um para os parâmetros RFC e outro para os parâmetros de execução) e emita o `create-rfc` comando com os dois arquivos como entrada. Ambos os métodos são descritos aqui.

2. Envie o `aws amscm submit-rfc --rfc-id` *`ID`* comando RFC: com o ID RFC retornado.

   Monitore o `aws amscm get-rfc --rfc-id` *`ID`* comando RFC:.

Para verificar a versão do tipo de alteração, use este comando:

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

> **Note**
>
> Você pode usar qualquer `CreateRfc` parâmetro com qualquer RFC, independentemente de eles fazerem parte do esquema para o tipo de alteração. Por exemplo, para receber notificações quando o status da RFC mudar, adicione essa linha `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` à parte dos parâmetros da RFC da solicitação (não aos parâmetros de execução). Para obter uma lista de todos os CreateRfc parâmetros, consulte a Referência da API de gerenciamento de alterações do AMS.

CRIAÇÃO EM LINHA:

Execute o comando create RFC com os parâmetros de execução fornecidos em linha (aspas de escape ao fornecer parâmetros de execução em linha) e, em seguida, envie a ID de RFC retornada. Por exemplo, você pode substituir o conteúdo por algo assim:

```
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\"}}"
```

CRIAÇÃO DE MODELO:

1. Exiba os parâmetros de execução do esquema JSON do CodeDeploy aplicativo CT para um arquivo na sua pasta atual; este exemplo o nomeia: CreateCDAppParams.json

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

2. Modifique e salve o arquivo JSON da seguinte maneira. Por exemplo, você pode substituir o conteúdo por algo assim:

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

3. Envie o modelo JSON CreateRfc para um arquivo em sua pasta atual; este exemplo o nomeia CreateCDAppRfc.json:

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

4. Modifique e salve o arquivo JSON da seguinte maneira. Por exemplo, você pode substituir o conteúdo por algo assim:

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

5. Crie o RFC, especificando o CreateCDAppRfc arquivo e o arquivo de parâmetros de execução:

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

   Você recebe a ID da nova RFC na resposta e pode usá-la para enviar e monitorar a RFC. Até que você o envie, o RFC permanece no estado de edição e não inicia.

Dicas

Para obter mais informações sobre a AWS CodeDeploy, consulte Criar um aplicativo com a AWS CodeDeploy.

Implantar CodeDeploy aplicativo

Implantando um CodeDeploy aplicativo com o console

Como funciona:

1. Navegue até a página Criar RFC: No painel de navegação esquerdo do console AMS, clique em RFCs para abrir a página de lista de RFCs e, em seguida, clique em Criar RFC.
2. Escolha um tipo de alteração popular (CT) na visualização padrão Procurar tipos de alteração ou selecione uma CT na visualização Escolher por categoria.
   - Navegar por tipo de alteração: você pode clicar em um CT popular na área de criação rápida para abrir imediatamente a página Executar RFC. Observe que você não pode escolher uma versão mais antiga do CT com a criação rápida.

     Para classificar CTs, use a área Todos os tipos de alteração na visualização em Cartão ou Tabela. Em qualquer exibição, selecione uma CT e clique em Criar RFC para abrir a página Executar RFC. Se aplicável, a opção Criar com uma versão mais antiga aparece ao lado do botão Criar RFC.
   - Escolha por categoria: selecione uma categoria, subcategoria, item e operação e a caixa de detalhes do CT será aberta com a opção Criar com uma versão mais antiga, se aplicável. Clique em Criar RFC para abrir a página Executar RFC.
3. Na página Executar RFC, abra a área do nome do CT para ver a caixa de detalhes do CT. É necessário um Assunto (preenchido se você escolher seu CT na visualização Procurar tipos de alteração). Abra a área Configuração adicional para adicionar informações sobre o RFC.

   Na área Configuração de execução, use as listas suspensas disponíveis ou insira valores para os parâmetros necessários. Para configurar parâmetros de execução opcionais, abra a área Configuração adicional.
4. Ao terminar, clique em Executar. Se não houver erros, a página RFC criada com sucesso será exibida com os detalhes da RFC enviada e a saída inicial de execução.
5. Abra a área Parâmetros de execução para ver as configurações que você enviou. Atualize a página para atualizar o status de execução do RFC. Opcionalmente, cancele a RFC ou crie uma cópia dela com as opções na parte superior da página.

Implantação de um CodeDeploy aplicativo com a CLI

Como funciona:

1. Use o Inline Create (você emite um `create-rfc` comando com todos os parâmetros de RFC e execução incluídos) ou o Template Create (você cria dois arquivos JSON, um para os parâmetros RFC e outro para os parâmetros de execução) e emita o `create-rfc` comando com os dois arquivos como entrada. Ambos os métodos são descritos aqui.
2. Envie o `aws amscm submit-rfc --rfc-id` *`ID`* comando RFC: com o ID RFC retornado.

   Monitore o `aws amscm get-rfc --rfc-id` *`ID`* comando RFC:.

Para verificar a versão do tipo de alteração, use este comando:

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

> **Note**
>
> Você pode usar qualquer `CreateRfc` parâmetro com qualquer RFC, independentemente de eles fazerem parte do esquema para o tipo de alteração. Por exemplo, para receber notificações quando o status da RFC mudar, adicione essa linha `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` à parte dos parâmetros da RFC da solicitação (não aos parâmetros de execução). Para obter uma lista de todos os CreateRfc parâmetros, consulte a Referência da API de gerenciamento de alterações do AMS.

CRIAÇÃO EM LINHA:

Emita o comando create RFC com os parâmetros de execução fornecidos em linha (aspas de escape ao fornecer parâmetros de execução em linha) e, em seguida, envie a ID de RFC retornada. Por exemplo, você pode substituir o conteúdo por algo assim:

```
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\"}}"
```

CRIAÇÃO DE MODELO:

1. Exiba o esquema JSON dos parâmetros de execução para o CT de implantação do CodeDeploy aplicativo; este exemplo o nomeia: DeployCDAppParams.json

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

2. Modifique o arquivo JSON da seguinte maneira. Por exemplo, você pode substituir o conteúdo por algo assim:

   ```
   {
   "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. Envie o modelo JSON CreateRfc para um arquivo em sua pasta atual; este exemplo o nomeia DeployCDAppRfc.json:

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

4. Modifique e salve o DeployCDAppRfc.json arquivo. Por exemplo, você pode substituir o conteúdo por algo assim:

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

5. Crie o RFC, especificando o arquivo de parâmetros de execução e o DeployCDAppRfc arquivo:

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

   Você recebe a ID da nova RFC na resposta e pode usá-la para enviar e monitorar a RFC. Até que você o envie, o RFC permanece no estado de edição e não inicia.

Dicas

Para obter mais informações, consulte Criar uma implantação com CodeDeploy.

CodeDeploy grupos de implantação

Crie grupos CodeDeploy de aplicativos.

Criar grupo CodeDeploy de implantação

Criação de um grupo de CodeDeploy implantação com o console

Como funciona:

1. Navegue até a página Criar RFC: No painel de navegação esquerdo do console AMS, clique em RFCs para abrir a página de lista de RFCs e, em seguida, clique em Criar RFC.
2. Escolha um tipo de alteração popular (CT) na visualização padrão Procurar tipos de alteração ou selecione uma CT na visualização Escolher por categoria.
   - Navegar por tipo de alteração: você pode clicar em um CT popular na área de criação rápida para abrir imediatamente a página Executar RFC. Observe que você não pode escolher uma versão mais antiga do CT com a criação rápida.

     Para classificar CTs, use a área Todos os tipos de alteração na visualização em Cartão ou Tabela. Em qualquer exibição, selecione uma CT e clique em Criar RFC para abrir a página

Executar RFC. Se aplicável, a opção Criar com uma versão mais antiga aparece ao lado do botão Criar RFC.

- Escolha por categoria: selecione uma categoria, subcategoria, item e operação e a caixa de detalhes do CT será aberta com a opção Criar com uma versão mais antiga, se aplicável. Clique em Criar RFC para abrir a página Executar RFC.

3. Na página Executar RFC, abra a área do nome do CT para ver a caixa de detalhes do CT. É necessário um Assunto (preenchido se você escolher seu CT na visualização Procurar tipos de alteração). Abra a área Configuração adicional para adicionar informações sobre o RFC.

   Na área Configuração de execução, use as listas suspensas disponíveis ou insira valores para os parâmetros necessários. Para configurar parâmetros de execução opcionais, abra a área Configuração adicional.

4. Ao terminar, clique em Executar. Se não houver erros, a página RFC criada com sucesso será exibida com os detalhes da RFC enviada e a saída inicial de execução.

5. Abra a área Parâmetros de execução para ver as configurações que você enviou. Atualize a página para atualizar o status de execução do RFC. Opcionalmente, cancele a RFC ou crie uma cópia dela com as opções na parte superior da página.

Criação de um grupo de CodeDeploy implantação com a CLI

Como funciona:

1. Use o Inline Create (você emite um `create-rfc` comando com todos os parâmetros de RFC e execução incluídos) ou o Template Create (você cria dois arquivos JSON, um para os parâmetros RFC e outro para os parâmetros de execução) e emita o `create-rfc` comando com os dois arquivos como entrada. Ambos os métodos são descritos aqui.
2. Envie o `aws amscm submit-rfc --rfc-id` *`ID`* comando RFC: com o ID RFC retornado.

   Monitore o `aws amscm get-rfc --rfc-id` *`ID`* comando RFC:.

Para verificar a versão do tipo de alteração, use este comando:

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

> **Note**
>
> Você pode usar qualquer `CreateRfc` parâmetro com qualquer RFC, independentemente de eles fazerem parte do esquema para o tipo de alteração. Por exemplo, para receber notificações quando o status da RFC mudar, adicione essa linha `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` à parte dos parâmetros da RFC da solicitação (não aos parâmetros de execução). Para obter uma lista de todos os CreateRfc parâmetros, consulte a Referência da API de gerenciamento de alterações do AMS.

CRIAÇÃO EM LINHA:

Emita o comando create RFC com os parâmetros de execução fornecidos em linha (aspas de escape ao fornecer parâmetros de execução em linha) e, em seguida, envie a ID de RFC retornada. Por exemplo, você pode substituir o conteúdo por algo assim:

```
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\"}}"
```

CRIAÇÃO DE MODELO:

1. Envie os parâmetros de execução do esquema JSON para um arquivo na sua pasta atual; este exemplo o nomeia: CreateCDDepGroupParams.json

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

2. Modifique e salve o arquivo JSON. Por exemplo, você pode substituir o conteúdo por algo assim:

   ```
   {
   "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. Envie o modelo JSON CreateRfc para um arquivo em sua pasta atual; este exemplo o nomeia CreateCDDepGroupRfc.json:

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

4. Modifique e salve o arquivo JSON. Por exemplo, você pode substituir o conteúdo por algo assim:

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

5. Crie o RFC, especificando o CreateCDDepGroupRfc arquivo e o arquivo de parâmetros de execução:

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

   Você recebe a ID da nova RFC na resposta e pode usá-la para enviar e monitorar a RFC. Até que você o envie, o RFC permanece no estado de edição e não inicia.

Dicas

Para obter mais informações sobre grupos de CodeDeploy implantação da AWS, consulte Criar um grupo de implantação com a AWS CodeDeploy.

Crie um grupo CodeDeploy de implantação para o EC2

Criação de um grupo de CodeDeploy implantação para o EC2 com o console

Como funciona:

1. Navegue até a página Criar RFC: No painel de navegação esquerdo do console AMS, clique em RFCs para abrir a página de lista de RFCs e, em seguida, clique em Criar RFC.
2. Escolha um tipo de alteração popular (CT) na visualização padrão Procurar tipos de alteração ou selecione uma CT na visualização Escolher por categoria.
   - Navegar por tipo de alteração: você pode clicar em um CT popular na área de criação rápida para abrir imediatamente a página Executar RFC. Observe que você não pode escolher uma versão mais antiga do CT com a criação rápida.

     Para classificar CTs, use a área Todos os tipos de alteração na visualização em Cartão ou Tabela. Em qualquer exibição, selecione uma CT e clique em Criar RFC para abrir a página Executar RFC. Se aplicável, a opção Criar com uma versão mais antiga aparece ao lado do botão Criar RFC.
   - Escolha por categoria: selecione uma categoria, subcategoria, item e operação e a caixa de detalhes do CT será aberta com a opção Criar com uma versão mais antiga, se aplicável. Clique em Criar RFC para abrir a página Executar RFC.
3. Na página Executar RFC, abra a área do nome do CT para ver a caixa de detalhes do CT. É necessário um Assunto (preenchido se você escolher seu CT na visualização Procurar tipos de alteração). Abra a área Configuração adicional para adicionar informações sobre o RFC.

   Na área Configuração de execução, use as listas suspensas disponíveis ou insira valores para os parâmetros necessários. Para configurar parâmetros de execução opcionais, abra a área Configuração adicional.
4. Ao terminar, clique em Executar. Se não houver erros, a página RFC criada com sucesso será exibida com os detalhes da RFC enviada e a saída inicial de execução.
5. Abra a área Parâmetros de execução para ver as configurações que você enviou. Atualize a página para atualizar o status de execução do RFC. Opcionalmente, cancele a RFC ou crie uma cópia dela com as opções na parte superior da página.

Criação de um grupo CodeDeploy de implantação para o EC2 com a CLI

Como funciona:

1. Use o Inline Create (você emite um `create-rfc` comando com todos os parâmetros de RFC e execução incluídos) ou o Template Create (você cria dois arquivos JSON, um para os parâmetros RFC e outro para os parâmetros de execução) e emita o `create-rfc` comando com os dois arquivos como entrada. Ambos os métodos são descritos aqui.
2. Envie o `aws amscm submit-rfc --rfc-id` *`ID`* comando RFC: com o ID RFC retornado.

   Monitore o `aws amscm get-rfc --rfc-id` *`ID`* comando RFC:.

Para verificar a versão do tipo de alteração, use este comando:

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

> **Note**
>
> Você pode usar qualquer `CreateRfc` parâmetro com qualquer RFC, independentemente de eles fazerem parte do esquema para o tipo de alteração. Por exemplo, para receber notificações quando o status da RFC mudar, adicione essa linha `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` à parte dos parâmetros da RFC da solicitação (não aos parâmetros de execução). Para obter uma lista de todos os CreateRfc parâmetros, consulte a Referência da API de gerenciamento de alterações do AMS.

CRIAÇÃO EM LINHA:

Execute o comando create RFC com os parâmetros de execução fornecidos em linha (aspas de escape ao fornecer parâmetros de execução em linha) e, em seguida, envie a ID de RFC retornada. Por exemplo, você pode substituir o conteúdo por algo assim:

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

CRIAÇÃO DE MODELO:

1. Exiba os parâmetros de execução do esquema JSON para um arquivo; este exemplo o nomeia: CreateCDDepGroupEc2Params.json

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

2. Modifique e salve o arquivo JSON. Por exemplo, você pode substituir o conteúdo por algo assim:

```
{
"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. Envie o modelo JSON CreateRfc para um arquivo em sua pasta atual; este exemplo o nomeia CreateCDDepGroupEc2Rfc.json:

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

4. Modifique e salve o arquivo JSON. Por exemplo, você pode substituir o conteúdo por algo assim:

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

5. Crie o RFC, especificando o CreateCDDepGroupEc2Rfc arquivo e o arquivo de parâmetros de execução:

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

   Você recebe a ID da nova RFC na resposta e pode usá-la para enviar e monitorar a RFC. Até que você o envie, o RFC permanece no estado de edição e não inicia.

Dicas

Para obter mais informações sobre grupos de CodeDeploy implantação da AWS, consulte Criar um grupo de implantação com a AWS CodeDeploy.

AWS Database Migration Service (AWS DMS)

AWS Database Migration Service (AWS DMS) ajuda você a migrar bancos de dados para o AMS com facilidade e segurança. É possível migrar dados de e para os bancos de dados comerciais de código aberto mais usados, como Oracle, MySQL e PostgreSQL. O serviço suporta migrações homogêneas, como Oracle para Oracle, e também migrações heterogêneas entre diferentes plataformas de banco de dados, como Oracle para PostgreSQL ou MySQL para Oracle. AWS DMS é um AWS serviço; os CTs do AMS ajudam você a criar AWS DMS recursos em sua conta AMS-managed

O gráfico a seguir mostra o fluxo de trabalho de uma migração de banco de dados.

Tópicos

- AWS Database Migration Service (AWS DMS), antes de começar
- AWS DMS, dados necessários para configuração
- AWS DMS tarefas de configuração
- AWS DMS gerenciamento

AWS Database Migration Service (AWS DMS), antes de começar

Ao planejar uma migração de banco de dados usando o AMS AWS DMS, considere o seguinte:

- Pontos finais de origem e destino: você precisa saber quais informações e tabelas no banco de dados de origem precisam ser migradas para o banco de dados de destino. AWS DMS O AMS oferece suporte à migração básica do esquema, incluindo a criação de tabelas e chaves primárias. No entanto, o AMS AWS DMS não cria automaticamente índices secundários, chaves estrangeiras, contas etc. no banco de dados de destino. Consulte Fontes para migração de dados e Destinos para migração de dados para obter mais informações.
- Migração de esquema/código: o AMS AWS DMS não realiza conversão de esquema ou código. Você pode usar ferramentas como o Oracle SQL Developer, MySQL Workbench, ou pgAdmin III para converter seu schema. Se quiser converter um esquema existente em um mecanismo de banco de dados diferente, você pode usar a AWS Schema Conversion Tool. Ela pode criar um schema de destino, além de poder gerar e criar um schema inteiro: tabelas, índices, exibições e assim por diante. Você também pode usar a ferramenta para converter PL/SQL ou TSQL em PgSQL e outros formatos.
- Tipos de dados não suportados: alguns tipos de dados de origem precisam ser convertidos em tipos de dados equivalentes para o banco de dados de destino.

AWS DMS cenários a serem considerados

Os cenários a seguir, documentados, podem ajudá-lo a criar seu próprio caminho de migração de banco de dados.

- Migre dados de um servidor MySQL local para o Amazon RDS MySQL: veja a postagem no blog da AWS Migrar dados MySQL locais para o Amazon RDS (e vice-versa)
- Migre dados de um banco de dados Oracle para o Amazon RDS Aurora PostgreSQL: veja a publicação no blog da AWS Uma rápida introdução à migração de um banco de dados Oracle para um banco de dados Amazon Aurora PostgreSQL
- Migre dados do RDS MySQL para o S3: veja a postagem no blog da AWS Como arquivar dados de bancos de dados relacionais no Amazon Glacier usando o AWS DMS

Para uma migração de banco de dados, faça o seguinte:

- Planeje a migração do banco de dados, isso inclui a configuração de um grupo de sub-redes de replicação.
- Aloque uma instância de replicação que execute todos os processos da migração.
- Especifique um endpoint de banco de dados de origem e de destino.

- Crie uma tarefa ou um conjunto de tarefas para definir as tabelas e os processos de replicação a serem utilizados.
- Crie o AWS DMS IAM `dms-cloudwatch-logs-role` e as `dms-vpc-role` funções. Se você usa o Amazon Redshift como um banco de dados de destino, você também deve criar e adicionar a função do IAM `dms-access-for-endpoint` à sua conta da AWS. Para obter mais informações, consulte Criação de funções do IAM para usar com a AWS CLI e a API do AWS DMS.

Essas orientações fornecem um exemplo do uso do console do AMS ou da CLI do AMS para criar um (). AWS Database Migration Service AWS DMS São fornecidos comandos CLI para criar a instância de AWS DMS replicação, o grupo de sub-rede e a tarefa, bem como um endpoint de AWS DMS origem e um endpoint de destino.

Para saber mais sobre o AMS AWS DMS, consulte AWS Database Migration Servicepara obter informações AWS Database Migration Service FAQsgerais e respostas a perguntas comuns.

AWS DMS, dados necessários para configuração

Para cada uma das AWS DMS orientações a seguir, alguns dados em comum são necessários.

- `Description`: informações significativas sobre o recurso, separadas de outras `Description` opções de parâmetros.
- `VpcId`: A VPC a ser usada. Você pode descobrir isso executando a ListVpcSummaries operação da API do SKMS (`list-vpc-summaries`na CLI) ou consultando a página no console VPCsdo AMS. Para a referência da API AMS SKMS, consulte a guia Relatórios no console do AWS Artifact.
- `Name`: um nome para a pilha ou componente da pilha; isso se torna o nome da pilha.
- `TimeoutInMinutes`: quantos minutos são permitidos para a criação da pilha antes que o RFC falhe. Essa configuração não atrasará a execução do RFC, mas você deve dar tempo suficiente (por exemplo, não especificar`"5"`).
- `ChangeTypeId`,`ChangeTypeVersion`, `eStackTemplateId`: Eles são obrigatórios, mas variam de acordo com a TC e seus valores são fornecidos em cada seção relevante, a seguir.

AWS DMS tarefas de configuração

Configure AWS DMS com as instruções a seguir.

1: AWS DMS grupo de sub-rede de replicação: Criar

Você pode usar o console do AMS ou API/CLI criar um grupo de sub-redes de AWS DMS replicação do AMS.

Criar AWS DMS grupo de sub-rede de replicação

Criando um AWS DMS grupo de sub-rede de replicação com o console

> **Note**
>
> Essa CT falhará se a função `dms-vpc-role` do IAM não existir na conta.

Como funciona:

1. Navegue até a página Criar RFC: No painel de navegação esquerdo do console AMS, clique em RFCs para abrir a página de lista de RFCs e, em seguida, clique em Criar RFC.
2. Escolha um tipo de alteração popular (CT) na visualização padrão Procurar tipos de alteração ou selecione uma CT na visualização Escolher por categoria.
   - Navegar por tipo de alteração: você pode clicar em um CT popular na área de criação rápida para abrir imediatamente a página Executar RFC. Observe que você não pode escolher uma versão mais antiga do CT com a criação rápida.

     Para classificar CTs, use a área Todos os tipos de alteração na visualização em Cartão ou Tabela. Em qualquer exibição, selecione uma CT e clique em Criar RFC para abrir a página Executar RFC. Se aplicável, a opção Criar com uma versão mais antiga aparece ao lado do botão Criar RFC.
   - Escolha por categoria: selecione uma categoria, subcategoria, item e operação e a caixa de detalhes do CT será aberta com a opção Criar com uma versão mais antiga, se aplicável. Clique em Criar RFC para abrir a página Executar RFC.
3. Na página Executar RFC, abra a área do nome do CT para ver a caixa de detalhes do CT. É necessário um Assunto (preenchido se você escolher seu CT na visualização Procurar tipos de alteração). Abra a área Configuração adicional para adicionar informações sobre o RFC.

Na área Configuração de execução, use as listas suspensas disponíveis ou insira valores para os parâmetros necessários. Para configurar parâmetros de execução opcionais, abra a área Configuração adicional.

4. Ao terminar, clique em Executar. Se não houver erros, a página RFC criada com sucesso será exibida com os detalhes da RFC enviada e a saída inicial de execução.
5. Abra a área Parâmetros de execução para ver as configurações que você enviou. Atualize a página para atualizar o status de execução do RFC. Opcionalmente, cancele a RFC ou crie uma cópia dela com as opções na parte superior da página.

Criando um AWS DMS grupo de sub-rede de replicação com a CLI

> **Note**
>
> Essa CT falhará se a função `dms-vpc-role` do IAM não existir na conta.

Como funciona:

1. Use o Inline Create (você emite um `create-rfc` comando com todos os parâmetros de RFC e execução incluídos) ou o Template Create (você cria dois arquivos JSON, um para os parâmetros RFC e outro para os parâmetros de execução) e emita o `create-rfc` comando com os dois arquivos como entrada. Ambos os métodos são descritos aqui.
2. Envie o `aws amscm submit-rfc --rfc-id` *`ID`* comando RFC: com o ID RFC retornado.

   Monitore o `aws amscm get-rfc --rfc-id` *`ID`* comando RFC:.

Para verificar a versão do tipo de alteração, use este comando:

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

> **Note**
>
> Você pode usar qualquer `CreateRfc` parâmetro com qualquer RFC, independentemente de eles fazerem parte do esquema para o tipo de alteração. Por exemplo, para receber notificações quando o status da RFC mudar, adicione essa linha `--notification`

`"{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` à parte dos parâmetros da RFC da solicitação (não aos parâmetros de execução). Para obter uma lista de todos os CreateRfc parâmetros, consulte a Referência da API de gerenciamento de alterações do AMS.

CRIAÇÃO EM LINHA:

Emita o comando create RFC com os parâmetros de execução fornecidos em linha (aspas de escape ao fornecer parâmetros de execução em linha) e, em seguida, envie o ID de RFC retornado. Por exemplo, você pode substituir o conteúdo por algo assim:

```
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
\"}"
```

CRIAÇÃO DE MODELO:

1. Exiba os parâmetros de execução desse tipo de alteração em um arquivo JSON; este exemplo o nomeia CreateDmsRsgParams.json:

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

2. Modifique e salve o CreateDmsRsgParams.json arquivo de parâmetros de execução. Por exemplo, você pode substituir o conteúdo por algo assim:

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

```
}
```

3. Envie o modelo JSON para um arquivo na sua pasta atual; este exemplo o nomeia CreateDmsRsgRfc.json:

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

4. Modifique e salve o CreateDmsRsgRfc.json arquivo. Por exemplo, você pode substituir o conteúdo por algo assim:

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

5. Crie o RFC, especificando o arquivo de parâmetros de execução e o CreateDmsRsgRfc arquivo:

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

   Você recebe a ID da nova RFC na resposta e pode usá-la para enviar e monitorar a RFC. Até que você o envie, o RFC permanece no estado de edição e não inicia.

Dicas

- Essa CT falhará se a função `dms-vpc-role` do IAM não existir na conta.
- Você pode adicionar até 50 tags, mas para isso você deve ativar a visualização de configuração adicional.

  Para obter mais informações sobre instâncias de replicação do DMS e grupos de sub-redes, consulte Configurando uma rede para uma instância de replicação.

2: AWS DMS instância de replicação: criar

Você pode usar o console do AMS ou API/CLI criar uma instância de AWS DMS replicação do AMS.

Criar AWS DMS instância de replicação

Criando um AWS DMS instância de replicação com o console

Captura de tela desse tipo de alteração no console AMS:

Como funciona:

1. Navegue até a página Criar RFC: No painel de navegação esquerdo do console AMS, clique em RFCs para abrir a página de lista de RFCs e, em seguida, clique em Criar RFC.
2. Escolha um tipo de alteração popular (CT) na visualização padrão Procurar tipos de alteração ou selecione uma CT na visualização Escolher por categoria.
   - Navegar por tipo de alteração: você pode clicar em um CT popular na área de criação rápida para abrir imediatamente a página Executar RFC. Observe que você não pode escolher uma versão mais antiga do CT com a criação rápida.

     Para classificar CTs, use a área Todos os tipos de alteração na visualização em Cartão ou Tabela. Em qualquer exibição, selecione uma CT e clique em Criar RFC para abrir a página Executar RFC. Se aplicável, a opção Criar com uma versão mais antiga aparece ao lado do botão Criar RFC.
   - Escolha por categoria: selecione uma categoria, subcategoria, item e operação e a caixa de detalhes do CT será aberta com a opção Criar com uma versão mais antiga, se aplicável. Clique em Criar RFC para abrir a página Executar RFC.
3. Na página Executar RFC, abra a área do nome do CT para ver a caixa de detalhes do CT. É necessário um Assunto (preenchido se você escolher seu CT na visualização Procurar tipos de alteração). Abra a área Configuração adicional para adicionar informações sobre o RFC.

   Na área Configuração de execução, use as listas suspensas disponíveis ou insira valores para os parâmetros necessários. Para configurar parâmetros de execução opcionais, abra a área Configuração adicional.
4. Ao terminar, clique em Executar. Se não houver erros, a página RFC criada com sucesso será exibida com os detalhes da RFC enviada e a saída inicial de execução.
5. Abra a área Parâmetros de execução para ver as configurações que você enviou. Atualize a página para atualizar o status de execução do RFC. Opcionalmente, cancele a RFC ou crie uma cópia dela com as opções na parte superior da página.

Criando um AWS DMS instância de replicação com a CLI

Como funciona:

1. Use o Inline Create (você emite um `create-rfc` comando com todos os parâmetros de RFC e execução incluídos) ou o Template Create (você cria dois arquivos JSON, um para os parâmetros RFC e outro para os parâmetros de execução) e emita o `create-rfc` comando com os dois arquivos como entrada. Ambos os métodos são descritos aqui.
2. Envie o `aws amscm submit-rfc --rfc-id` *`ID`* comando RFC: com o ID RFC retornado.

   Monitore o `aws amscm get-rfc --rfc-id` *`ID`* comando RFC:.

Para verificar a versão do tipo de alteração, use este comando:

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

> **Note**
>
> Você pode usar qualquer `CreateRfc` parâmetro com qualquer RFC, independentemente de eles fazerem parte do esquema para o tipo de alteração. Por exemplo, para receber notificações quando o status da RFC mudar, adicione essa linha `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` à parte dos parâmetros da RFC da solicitação (não aos parâmetros de execução). Para obter uma lista de todos os CreateRfc parâmetros, consulte a Referência da API de gerenciamento de alterações do AMS.

CRIAÇÃO EM LINHA:

Execute o comando create RFC com os parâmetros de execução fornecidos em linha (aspas de escape ao fornecer parâmetros de execução em linha) e, em seguida, envie a ID de RFC retornada. Por exemplo, você pode substituir o conteúdo por algo assim:

```
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\"}"
```

Enquanto a instância de replicação é criada, você pode especificar os datastores de origem e de destino. Os armazenamentos de dados de origem e destino podem estar em uma instância do Amazon Elastic Compute Cloud (Amazon EC2), em um AWS S3 Bucket, em uma instância de banco de dados do Amazon Relational Database Service (Amazon RDS) ou em um banco de dados local.

CRIAÇÃO DE MODELO:

1. Exiba os parâmetros de execução desse tipo de alteração em um arquivo JSON; este exemplo o nomeia CreateDmsRiParams.json:

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

2. Modifique e salve o CreateDmsRiParams.json arquivo de parâmetros de execução. Por exemplo, você pode substituir o conteúdo por algo assim:

   ```
   {
   "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. Envie o modelo JSON para um arquivo na sua pasta atual; este exemplo o nomeia CreateDmsRiRfc.json:

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

4. Modifique e salve o CreateDmsRiRfc.json arquivo. Por exemplo, você pode substituir o conteúdo por algo assim:

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

5. Crie o RFC, especificando o arquivo de parâmetros de execução e o CreateDmsRiRfc arquivo:

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

   Você recebe a ID da nova RFC na resposta e pode usá-la para enviar e monitorar a RFC. Até que você o envie, o RFC permanece no estado de edição e não inicia.

Dicas

- Você pode adicionar até 50 tags, mas para isso você deve habilitar a visualização de configuração adicional.
- Você deve criar uma instância de replicação em uma instância do EC2 em sua VPC do AMS que tenha armazenamento e poder de processamento suficientes para realizar as tarefas que você atribui e migrar dados do seu banco de dados de origem para o banco de dados de destino. O tamanho necessário dessa instância varia de acordo com a quantidade de dados necessários para migrar e as tarefas que ela deve realizar. A instância de replicação fornece alta disponibilidade e suporte de failover usando uma Multi-AZ implantação quando você seleciona a `MultiAZ` opção. Para obter mais informações sobre instâncias de replicação, consulte Como trabalhar com uma instância de replicação do AWS DMS.

3: AWS DMS endpoint de origem: criar, criar para Mongo DB, criar para S3

Você pode usar o console do AMS ou, API/CLI para criar um endpoint de origem do AMS DMS para vários bancos de dados, fornecemos três exemplos.

Endpoint de origem do DMS: criação

Criando um endpoint de origem do DMS com o console

Captura de tela desse tipo de alteração no console AMS:

Como funciona:

1. Navegue até a página Criar RFC: No painel de navegação esquerdo do console AMS, clique em RFCs para abrir a página de lista de RFCs e, em seguida, clique em Criar RFC.
2. Escolha um tipo de alteração popular (CT) na visualização padrão Procurar tipos de alteração ou selecione uma CT na visualização Escolher por categoria.
   - Navegar por tipo de alteração: você pode clicar em um CT popular na área de criação rápida para abrir imediatamente a página Executar RFC. Observe que você não pode escolher uma versão mais antiga do CT com a criação rápida.

     Para classificar CTs, use a área Todos os tipos de alteração na visualização em Cartão ou Tabela. Em qualquer exibição, selecione uma CT e clique em Criar RFC para abrir a página Executar RFC. Se aplicável, a opção Criar com versão mais antiga aparece ao lado do botão Criar RFC.
   - Escolha por categoria: selecione uma categoria, subcategoria, item e operação e a caixa de detalhes do CT será aberta com a opção Criar com uma versão mais antiga, se aplicável. Clique em Criar RFC para abrir a página Executar RFC.
3. Na página Executar RFC, abra a área do nome do CT para ver a caixa de detalhes do CT. É necessário um Assunto (preenchido se você escolher seu CT na visualização Procurar tipos de alteração). Abra a área Configuração adicional para adicionar informações sobre o RFC.

   Na área Configuração de execução, use as listas suspensas disponíveis ou insira valores para os parâmetros necessários. Para configurar parâmetros de execução opcionais, abra a área Configuração adicional.
4. Ao terminar, clique em Executar. Se não houver erros, a página RFC criada com sucesso será exibida com os detalhes da RFC enviada e a saída inicial de execução.
5. Abra a área Parâmetros de execução para ver as configurações que você enviou. Atualize a página para atualizar o status de execução do RFC. Opcionalmente, cancele a RFC ou crie uma cópia dela com as opções na parte superior da página.

Criando um endpoint de origem do DMS com a CLI

Como funciona:

1. Use o Inline Create (você emite um `create-rfc` comando com todos os parâmetros de RFC e execução incluídos) ou o Template Create (você cria dois arquivos JSON, um para os parâmetros

RFC e outro para os parâmetros de execução) e emita o `create-rfc` comando com os dois arquivos como entrada. Ambos os métodos são descritos aqui.

2. Envie o `aws amscm submit-rfc --rfc-id` *`ID`* comando RFC: com o ID RFC retornado.

   Monitore o `aws amscm get-rfc --rfc-id` *`ID`* comando RFC:.

Para verificar a versão do tipo de alteração, use este comando:

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

> **Note**
>
> Você pode usar qualquer `CreateRfc` parâmetro com qualquer RFC, independentemente de eles fazerem parte do esquema para o tipo de alteração. Por exemplo, para receber notificações quando o status da RFC mudar, adicione essa linha `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` à parte dos parâmetros da RFC da solicitação (não aos parâmetros de execução). Para obter uma lista de todos os CreateRfc parâmetros, consulte a Referência da API de gerenciamento de alterações do AMS.

CRIAÇÃO EM LINHA:

Execute o comando create RFC com os parâmetros de execução fornecidos em linha (aspas de escape ao fornecer parâmetros de execução em linha) e, em seguida, envie a ID de RFC retornada. Por exemplo, você pode substituir o conteúdo por algo assim:

```
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\"}"
```

CRIAÇÃO DE MODELO:

1. Envie os parâmetros de execução desse tipo de alteração para um arquivo JSON chamado CreateDmsSeParams.json.

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

2. Modifique e salve o arquivo JSON dos parâmetros de execução. Por exemplo, você pode substituir o conteúdo por algo assim:

```
{
"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. Envie o modelo JSON para um arquivo na sua pasta atual; este exemplo o nomeia CreateDmsSeRfc.json:

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

4. Modifique e salve o CreateDmsSeRfc.json arquivo. Por exemplo, você pode substituir o conteúdo por algo assim:

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

5. Crie o RFC, especificando o arquivo de parâmetros de execução e o CreateDmsSeRfc arquivo:

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

Você recebe a ID da nova RFC na resposta e pode usá-la para enviar e monitorar a RFC. Até que você o envie, o RFC permanece no estado de edição e não inicia.

Dicas

Antes de criar o endpoint do DMS, certifique-se de que sua senha não contenha caracteres incompatíveis. Para obter mais informações, consulte Criação de endpoints de origem e destino no Guia do AWS Database Migration Service usuário.

Para saber mais, consulte Fontes para migração de dados.

Para um endpoint de origem S3, consulte. Endpoint de origem DMS para S3: criação

Para um endpoint de origem do Mongo DB, consulte. Endpoint de origem DMS para MongoDB: Criação

Endpoint de origem DMS para MongoDB: Criação

Criando um endpoint de origem de banco de dados DMS Mongo com o console

Captura de tela desse tipo de alteração no console AMS:

Como funciona:

1. Navegue até a página Criar RFC: No painel de navegação esquerdo do console AMS, clique em RFCs para abrir a página de lista de RFCs e, em seguida, clique em Criar RFC.
2. Escolha um tipo de alteração popular (CT) na visualização padrão Procurar tipos de alteração ou selecione uma CT na visualização Escolher por categoria.
   - Navegar por tipo de alteração: você pode clicar em um CT popular na área de criação rápida para abrir imediatamente a página Executar RFC. Observe que você não pode escolher uma versão mais antiga do CT com a criação rápida.

     Para classificar CTs, use a área Todos os tipos de alteração na visualização em Cartão ou Tabela. Em qualquer exibição, selecione uma CT e clique em Criar RFC para abrir a página

Executar RFC. Se aplicável, a opção Criar com versão mais antiga aparece ao lado do botão Criar RFC.

- Escolha por categoria: selecione uma categoria, subcategoria, item e operação e a caixa de detalhes do CT será aberta com a opção Criar com uma versão mais antiga, se aplicável. Clique em Criar RFC para abrir a página Executar RFC.

3. Na página Executar RFC, abra a área do nome do CT para ver a caixa de detalhes do CT. É necessário um Assunto (preenchido se você escolher seu CT na visualização Procurar tipos de alteração). Abra a área Configuração adicional para adicionar informações sobre o RFC.

   Na área Configuração de execução, use as listas suspensas disponíveis ou insira valores para os parâmetros necessários. Para configurar parâmetros de execução opcionais, abra a área Configuração adicional.

4. Ao terminar, clique em Executar. Se não houver erros, a página RFC criada com sucesso será exibida com os detalhes da RFC enviada e a saída inicial de execução.

5. Abra a área Parâmetros de execução para ver as configurações que você enviou. Atualize a página para atualizar o status de execução do RFC. Opcionalmente, cancele a RFC ou crie uma cópia dela com as opções na parte superior da página.

Criando um endpoint de origem de banco de dados DMS Mongo com a CLI

Como funciona:

1. Use o Inline Create (você emite um `create-rfc` comando com todos os parâmetros de RFC e execução incluídos) ou o Template Create (você cria dois arquivos JSON, um para os parâmetros RFC e outro para os parâmetros de execução) e emita o `create-rfc` comando com os dois arquivos como entrada. Ambos os métodos são descritos aqui.
2. Envie o `aws amscm submit-rfc --rfc-id` *`ID`* comando RFC: com o ID RFC retornado.

   Monitore o `aws amscm get-rfc --rfc-id` *`ID`* comando RFC:.

Para verificar a versão do tipo de alteração, use este comando:

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

> **Note**
>
> Você pode usar qualquer `CreateRfc` parâmetro com qualquer RFC, independentemente de eles fazerem parte do esquema para o tipo de alteração. Por exemplo, para receber notificações quando o status da RFC mudar, adicione essa linha `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` à parte dos parâmetros da RFC da solicitação (não aos parâmetros de execução). Para obter uma lista de todos os CreateRfc parâmetros, consulte a Referência da API de gerenciamento de alterações do AMS.

CRIAÇÃO EM LINHA:

Execute o comando create RFC com os parâmetros de execução fornecidos em linha (aspas de escape ao fornecer parâmetros de execução em linha) e, em seguida, envie a ID de RFC retornada. Por exemplo, você pode substituir o conteúdo por algo assim:

```
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\"}}"
```

CRIAÇÃO DE MODELO:

1. Envie os parâmetros de execução desse tipo de alteração para um arquivo JSON chamado CreateDmsSeMongoParams.json.

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

2. Modifique e salve o arquivo JSON dos parâmetros de execução. Por exemplo, você pode substituir o conteúdo por algo assim:

   ```
   {
   "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. Envie o modelo JSON para um arquivo na sua pasta atual; este exemplo o nomeia CreateDmsSeMongoRfc.json:

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

4. Modifique e salve o CreateDmsSeMongoRfc.json arquivo. Por exemplo, você pode substituir o conteúdo por algo assim:

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

5. Crie o RFC, especificando o arquivo de parâmetros de execução e o CreateDmsSeMongoRfc arquivo:

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

Você recebe a ID da nova RFC na resposta e pode usá-la para enviar e monitorar a RFC. Até que você o envie, o RFC permanece no estado de edição e não inicia.

Dicas

> **Note**
>
> Você pode adicionar até 50 tags, mas para isso você deve ativar a visualização de configuração adicional.

O AMS DMS pode usar o Mongo ou qualquer Relational Database Service (RDS) como um endpoint de origem. Para um endpoint de origem S3, consulte. Endpoint de origem DMS para S3: criação

Endpoint de origem DMS para S3: criação

Criando um endpoint de origem do DMS S3 com o console

Captura de tela desse tipo de alteração no console AMS:

Como funciona:

1. Navegue até a página Criar RFC: No painel de navegação esquerdo do console AMS, clique em RFCs para abrir a página de lista de RFCs e, em seguida, clique em Criar RFC.
2. Escolha um tipo de alteração popular (CT) na visualização padrão Procurar tipos de alteração ou selecione uma CT na visualização Escolher por categoria.
   - Navegar por tipo de alteração: você pode clicar em um CT popular na área de criação rápida para abrir imediatamente a página Executar RFC. Observe que você não pode escolher uma versão mais antiga do CT com a criação rápida.

     Para classificar CTs, use a área Todos os tipos de alteração na visualização em Cartão ou Tabela. Em qualquer exibição, selecione uma CT e clique em Criar RFC para abrir a página Executar RFC. Se aplicável, a opção Criar com versão mais antiga aparece ao lado do botão Criar RFC.
   - Escolha por categoria: selecione uma categoria, subcategoria, item e operação e a caixa de detalhes do CT será aberta com a opção Criar com uma versão mais antiga, se aplicável. Clique em Criar RFC para abrir a página Executar RFC.
3. Na página Executar RFC, abra a área do nome do CT para ver a caixa de detalhes do CT. É necessário um Assunto (preenchido se você escolher seu CT na visualização Procurar tipos de alteração). Abra a área Configuração adicional para adicionar informações sobre o RFC.

Na área Configuração de execução, use as listas suspensas disponíveis ou insira valores para os parâmetros necessários. Para configurar parâmetros de execução opcionais, abra a área Configuração adicional.

4. Ao terminar, clique em Executar. Se não houver erros, a página RFC criada com sucesso será exibida com os detalhes da RFC enviada e a saída inicial de execução.
5. Abra a área Parâmetros de execução para ver as configurações que você enviou. Atualize a página para atualizar o status de execução do RFC. Opcionalmente, cancele a RFC ou crie uma cópia dela com as opções na parte superior da página.

Criando um endpoint de origem do DMS S3 com a CLI

Como funciona:

1. Use o Inline Create (você emite um `create-rfc` comando com todos os parâmetros de RFC e execução incluídos) ou o Template Create (você cria dois arquivos JSON, um para os parâmetros RFC e outro para os parâmetros de execução) e emita o `create-rfc` comando com os dois arquivos como entrada. Ambos os métodos são descritos aqui.
2. Envie o `aws amscm submit-rfc --rfc-id` *`ID`* comando RFC: com o ID RFC retornado.

   Monitore o `aws amscm get-rfc --rfc-id` *`ID`* comando RFC:.

Para verificar a versão do tipo de alteração, use este comando:

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

> **Note**
>
> Você pode usar qualquer `CreateRfc` parâmetro com qualquer RFC, independentemente de eles fazerem parte do esquema para o tipo de alteração. Por exemplo, para receber notificações quando o status da RFC mudar, adicione essa linha `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` à parte dos parâmetros da RFC da solicitação (não aos parâmetros de execução). Para obter uma lista de todos os CreateRfc parâmetros, consulte a Referência da API de gerenciamento de alterações do AMS.

CRIAÇÃO EM LINHA:

Execute o comando create RFC com os parâmetros de execução fornecidos em linha (aspas de escape ao fornecer parâmetros de execução em linha) e, em seguida, envie a ID de RFC retornada. Por exemplo, você pode substituir o conteúdo por algo assim:

```
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\"}"
```

CRIAÇÃO DE MODELO:

1. Envie os parâmetros de execução desse tipo de alteração para um arquivo JSON chamado CreateDmsSeS3Params.json.

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

2. Modifique e salve o arquivo JSON dos parâmetros de execução. Por exemplo, você pode substituir o conteúdo por algo assim:

   ```
   {
   "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. Envie o modelo JSON para um arquivo na sua pasta atual; este exemplo o nomeia CreateDmsSeS3Rfc.json:

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

4. Modifique e salve o CreateDmsSeS3Rfc.json arquivo. Por exemplo, você pode substituir o conteúdo por algo assim:

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

5. Crie o RFC, especificando o arquivo de parâmetros de execução e o CreateDmsSeS3Rfc arquivo:

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

Você recebe a ID da nova RFC na resposta e pode usá-la para enviar e monitorar a RFC. Até que você o envie, o RFC permanece no estado de edição e não inicia.

Dicas

> **Note**
>
> Você pode adicionar até 50 tags, mas para isso você deve ativar a visualização de configuração adicional.

O AMS DMS pode usar o S3 ou qualquer endpoint de origem do Relational Database Service (RDS) Relational Database Service (RDS). Para um endpoint de origem do Mongo DB, consulte. Endpoint de origem DMS para MongoDB: Criação

4: AWS DMS endpoint de destino: criar, criar para o S3

Você pode usar o console do AMS ou, API/CLI para criar um endpoint de destino do AMS DMS para vários bancos de dados, fornecemos dois exemplos.

Ponto final de destino do DMS: criação

O AMS DMS pode usar o S3 ou qualquer Relational Database Service (RDS) com MySQL, MariaDB, Oracle, Postgresql ou Microsoft SQL como um endpoint de destino.

Criando um endpoint de destino do DMS com o console

Captura de tela desse tipo de alteração no console AMS:

Como funciona:

1. Navegue até a página Criar RFC: No painel de navegação esquerdo do console AMS, clique em RFCs para abrir a página de lista de RFCs e, em seguida, clique em Criar RFC.
2. Escolha um tipo de alteração popular (CT) na visualização padrão Procurar tipos de alteração ou selecione uma CT na visualização Escolher por categoria.
   - Navegar por tipo de alteração: você pode clicar em um CT popular na área de criação rápida para abrir imediatamente a página Executar RFC. Observe que você não pode escolher uma versão mais antiga do CT com a criação rápida.

     Para classificar CTs, use a área Todos os tipos de alteração na visualização em Cartão ou Tabela. Em qualquer exibição, selecione uma CT e clique em Criar RFC para abrir a página

Executar RFC. Se aplicável, a opção Criar com uma versão mais antiga aparece ao lado do botão Criar RFC.

- Escolha por categoria: selecione uma categoria, subcategoria, item e operação e a caixa de detalhes do CT será aberta com a opção Criar com uma versão mais antiga, se aplicável. Clique em Criar RFC para abrir a página Executar RFC.

3. Na página Executar RFC, abra a área do nome do CT para ver a caixa de detalhes do CT. É necessário um Assunto (preenchido se você escolher seu CT na visualização Procurar tipos de alteração). Abra a área Configuração adicional para adicionar informações sobre o RFC.

   Na área Configuração de execução, use as listas suspensas disponíveis ou insira valores para os parâmetros necessários. Para configurar parâmetros de execução opcionais, abra a área Configuração adicional.

4. Ao terminar, clique em Executar. Se não houver erros, a página RFC criada com sucesso será exibida com os detalhes da RFC enviada e a saída inicial de execução.

5. Abra a área Parâmetros de execução para ver as configurações que você enviou. Atualize a página para atualizar o status de execução do RFC. Opcionalmente, cancele a RFC ou crie uma cópia dela com as opções na parte superior da página.

Criando um endpoint de destino do DMS com a CLI

Como funciona:

1. Use o Inline Create (você emite um `create-rfc` comando com todos os parâmetros de RFC e execução incluídos) ou o Template Create (você cria dois arquivos JSON, um para os parâmetros RFC e outro para os parâmetros de execução) e emita o `create-rfc` comando com os dois arquivos como entrada. Ambos os métodos são descritos aqui.
2. Envie o `aws amscm submit-rfc --rfc-id` *`ID`* comando RFC: com o ID RFC retornado.

   Monitore o `aws amscm get-rfc --rfc-id` *`ID`* comando RFC:.

Para verificar a versão do tipo de alteração, use este comando:

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

> **Note**
>
> Você pode usar qualquer `CreateRfc` parâmetro com qualquer RFC, independentemente de eles fazerem parte do esquema para o tipo de alteração. Por exemplo, para receber notificações quando o status da RFC mudar, adicione essa linha `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` à parte dos parâmetros da RFC da solicitação (não aos parâmetros de execução). Para obter uma lista de todos os CreateRfc parâmetros, consulte a Referência da API de gerenciamento de alterações do AMS.

CRIAÇÃO EM LINHA:

Execute o comando create RFC com os parâmetros de execução fornecidos em linha (aspas de escape ao fornecer parâmetros de execução em linha) e, em seguida, envie a ID de RFC retornada. Por exemplo, você pode substituir o conteúdo por algo assim:

```
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\"}}"
```

CRIAÇÃO DE MODELO:

1. Exiba os parâmetros de execução desse tipo de alteração em um arquivo JSON chamado CreateDmsTeParams.json.

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

2. Modifique e salve o arquivo JSON dos parâmetros de execução. Por exemplo, você pode substituir o conteúdo por algo assim:

   ```
   {
   "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. Envie o modelo JSON para um arquivo na sua pasta atual; este exemplo o nomeia CreateDmsTeRfc.json:

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

4. Modifique e salve o CreateDmsTeRfc.json arquivo. Por exemplo, você pode substituir o conteúdo por algo assim:

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

5. Crie o RFC, especificando o arquivo de parâmetros de execução e o CreateDmsTeRfc arquivo:

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

   Você recebe a ID da nova RFC na resposta e pode usá-la para enviar e monitorar a RFC. Até que você o envie, o RFC permanece no estado de edição e não inicia.

Dicas

- Esse tipo de alteração está agora na versão 2.0.
- O AMS DMS pode usar o S3 ou qualquer Relational Database Service (RDS) com MySQL, MariaDB, Oracle, Postgresql ou Microsoft SQL como um endpoint de destino. Para um endpoint de destino do S3, consulte. Ponto final de destino do DMS para S3: criação
- Para obter mais informações, consulte Metas para migração de dados.

- Você pode adicionar até 50 tags, mas para isso você deve habilitar a visualização de configuração adicional.

Ponto final de destino do DMS para S3: criação

Criando um endpoint de destino do DMS S3 com o console

Captura de tela desse tipo de alteração no console AMS:

Como funciona:

1. Navegue até a página Criar RFC: No painel de navegação esquerdo do console AMS, clique em RFCs para abrir a página de lista de RFCs e, em seguida, clique em Criar RFC.
2. Escolha um tipo de alteração popular (CT) na visualização padrão Procurar tipos de alteração ou selecione uma CT na visualização Escolher por categoria.
   - Navegar por tipo de alteração: você pode clicar em um CT popular na área de criação rápida para abrir imediatamente a página Executar RFC. Observe que você não pode escolher uma versão mais antiga do CT com a criação rápida.

     Para classificar CTs, use a área Todos os tipos de alteração na visualização em Cartão ou Tabela. Em qualquer exibição, selecione uma CT e clique em Criar RFC para abrir a página Executar RFC. Se aplicável, a opção Criar com uma versão mais antiga aparece ao lado do botão Criar RFC.
   - Escolha por categoria: selecione uma categoria, subcategoria, item e operação e a caixa de detalhes do CT será aberta com a opção Criar com uma versão mais antiga, se aplicável. Clique em Criar RFC para abrir a página Executar RFC.
3. Na página Executar RFC, abra a área do nome do CT para ver a caixa de detalhes do CT. É necessário um Assunto (preenchido se você escolher seu CT na visualização Procurar tipos de alteração). Abra a área Configuração adicional para adicionar informações sobre o RFC.

   Na área Configuração de execução, use as listas suspensas disponíveis ou insira valores para os parâmetros necessários. Para configurar parâmetros de execução opcionais, abra a área Configuração adicional.
4. Ao terminar, clique em Executar. Se não houver erros, a página RFC criada com sucesso será exibida com os detalhes da RFC enviada e a saída inicial de execução.

5. Abra a área Parâmetros de execução para ver as configurações que você enviou. Atualize a página para atualizar o status de execução do RFC. Opcionalmente, cancele a RFC ou crie uma cópia dela com as opções na parte superior da página.

Criando um endpoint de destino do DMS S3 com a CLI

Como funciona:

1. Use o Inline Create (você emite um `create-rfc` comando com todos os parâmetros de RFC e execução incluídos) ou o Template Create (você cria dois arquivos JSON, um para os parâmetros RFC e outro para os parâmetros de execução) e emita o `create-rfc` comando com os dois arquivos como entrada. Ambos os métodos são descritos aqui.
2. Envie o `aws amscm submit-rfc --rfc-id` *`ID`* comando RFC: com o ID RFC retornado.

   Monitore o `aws amscm get-rfc --rfc-id` *`ID`* comando RFC:.

Para verificar a versão do tipo de alteração, use este comando:

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

> **Note**
>
> Você pode usar qualquer `CreateRfc` parâmetro com qualquer RFC, independentemente de eles fazerem parte do esquema para o tipo de alteração. Por exemplo, para receber notificações quando o status da RFC mudar, adicione essa linha `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` à parte dos parâmetros da RFC da solicitação (não aos parâmetros de execução). Para obter uma lista de todos os CreateRfc parâmetros, consulte a Referência da API de gerenciamento de alterações do AMS.

CRIAÇÃO EM LINHA:

Execute o comando create RFC com os parâmetros de execução fornecidos em linha (aspas de escape ao fornecer parâmetros de execução em linha) e, em seguida, envie a ID de RFC retornada. Por exemplo, você pode substituir o conteúdo por algo assim:

```
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\"}}"
```

CRIAÇÃO DE MODELO:

1. Exiba os parâmetros de execução desse tipo de alteração em um arquivo JSON; este exemplo o nomeia CreateDmsTeS3Params.json:

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

2. Modifique e salve o CreateDmsTeS3Params.json arquivo de parâmetros de execução. Por exemplo, você pode substituir o conteúdo por algo assim:

   ```
   {
   "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. Envie o modelo JSON para um arquivo na sua pasta atual; este exemplo o nomeia CreateDmsTeS3Rfc.json:

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

4. Modifique e salve o CreateDmsTeS3Rfc.json arquivo. Por exemplo, você pode substituir o conteúdo por algo assim:

   ```
   {
   ```

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

5. Crie o RFC, especificando o arquivo de parâmetros de execução e o CreateDmsTeS3Rfc arquivo:

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

   Você recebe a ID da nova RFC na resposta e pode usá-la para enviar e monitorar a RFC. Até que você o envie, o RFC permanece no estado de edição e não inicia.

Dicas

> **Note**
>
> Você pode adicionar até 50 tags, mas para isso você deve habilitar a visualização de configuração adicional.

O AMS fornece um tipo de alteração separado para criar um endpoint de destino para o S3. Para obter mais informações, consulte Usando o Amazon S3 como destino para o AWS Database Migration Service e Atributos extras de conexão ao usar o Amazon S3 como destino para o AWS DMS.

5: AWS DMS tarefa de replicação: Criar

Você pode usar o console do AMS ou API/CLI criar uma tarefa de AWS DMS replicação do AMS.

Criar AWS DMS tarefa de replicação

Criando um AWS DMS Tarefa de replicação com o console

Captura de tela desse tipo de alteração no console AMS:

Como funciona:

1. Navegue até a página Criar RFC: No painel de navegação esquerdo do console AMS, clique em RFCs para abrir a página de lista de RFCs e, em seguida, clique em Criar RFC.
2. Escolha um tipo de alteração popular (CT) na visualização padrão Procurar tipos de alteração ou selecione uma CT na visualização Escolher por categoria.
   - Navegar por tipo de alteração: você pode clicar em um CT popular na área de criação rápida para abrir imediatamente a página Executar RFC. Observe que você não pode escolher uma versão mais antiga do CT com a criação rápida.

     Para classificar CTs, use a área Todos os tipos de alteração na visualização em Cartão ou Tabela. Em qualquer exibição, selecione uma CT e clique em Criar RFC para abrir a página Executar RFC. Se aplicável, a opção Criar com uma versão mais antiga aparece ao lado do botão Criar RFC.
   - Escolha por categoria: selecione uma categoria, subcategoria, item e operação e a caixa de detalhes do CT será aberta com a opção Criar com uma versão mais antiga, se aplicável. Clique em Criar RFC para abrir a página Executar RFC.
3. Na página Executar RFC, abra a área do nome do CT para ver a caixa de detalhes do CT. É necessário um Assunto (preenchido se você escolher seu CT na visualização Procurar tipos de alteração). Abra a área Configuração adicional para adicionar informações sobre o RFC.

   Na área Configuração de execução, use as listas suspensas disponíveis ou insira valores para os parâmetros necessários. Para configurar parâmetros de execução opcionais, abra a área Configuração adicional.
4. Ao terminar, clique em Executar. Se não houver erros, a página RFC criada com sucesso será exibida com os detalhes da RFC enviada e a saída inicial de execução.
5. Abra a área Parâmetros de execução para ver as configurações que você enviou. Atualize a página para atualizar o status de execução do RFC. Opcionalmente, cancele a RFC ou crie uma cópia dela com as opções na parte superior da página.

Criando um AWS DMS Tarefa de replicação com a CLI

Como funciona:

1. Use o Inline Create (você emite um `create-rfc` comando com todos os parâmetros de RFC e execução incluídos) ou o Template Create (você cria dois arquivos JSON, um para os parâmetros RFC e outro para os parâmetros de execução) e emita o `create-rfc` comando com os dois arquivos como entrada. Ambos os métodos são descritos aqui.

2. Envie o `aws amscm submit-rfc --rfc-id` *ID* comando RFC: com o ID RFC retornado.

   Monitore o `aws amscm get-rfc --rfc-id` *ID* comando RFC:.

Para verificar a versão do tipo de alteração, use este comando:

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

> **Note**
>
> Você pode usar qualquer `CreateRfc` parâmetro com qualquer RFC, independentemente de eles fazerem parte do esquema para o tipo de alteração. Por exemplo, para receber notificações quando o status da RFC mudar, adicione essa linha `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` à parte dos parâmetros da RFC da solicitação (não aos parâmetros de execução). Para obter uma lista de todos os CreateRfc parâmetros, consulte a Referência da API de gerenciamento de alterações do AMS.

CRIAÇÃO EM LINHA:

Execute o comando create RFC com os parâmetros de execução fornecidos em linha (aspas de escape ao fornecer parâmetros de execução em linha) e, em seguida, envie a ID de RFC retornada. Por exemplo, você pode substituir o conteúdo por algo assim:

```
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}"
```

CRIAÇÃO DE MODELO:

1. Exiba os parâmetros de execução desse tipo de alteração em um arquivo JSON; este exemplo o nomeia CreateDmsRtParams.json:

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

2. Modifique e salve o arquivo JSON dos parâmetros de execução. Por exemplo, você pode substituir o conteúdo por algo assim:

```
{
"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. Envie o modelo JSON para um arquivo na sua pasta atual; este exemplo o nomeia CreateDmsRtRfc.json:

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

4. Modifique e salve o CreateDmsRtRfc.json arquivo. Por exemplo, você pode substituir o conteúdo por algo assim:

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

5. Crie o RFC, especificando o arquivo de parâmetros de execução e o CreateDmsRtRfc arquivo:

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

Você recebe a ID da nova RFC na resposta e pode usá-la para enviar e monitorar a RFC. Até que você o envie, o RFC permanece no estado de edição e não inicia.

Dicas

Você pode criar uma AWS DMS tarefa que capture três tipos diferentes de alterações ou dados. Para obter mais informações, consulte Como trabalhar com tarefas do AWS DMS, Criar uma tarefa e Criar tarefas para replicação contínua usando o AWS DMS.

AWS DMS gerenciamento

AWS DMS exemplos de gerenciamento.

Início AWS DMS tarefa de replicação

Iniciando um AWS DMS tarefa de replicação com o console

Captura de tela desse tipo de alteração no console AMS:

Como funciona:

1. Navegue até a página Criar RFC: No painel de navegação esquerdo do console AMS, clique em RFCs para abrir a página de lista de RFCs e, em seguida, clique em Criar RFC.
2. Escolha um tipo de alteração popular (CT) na visualização padrão Procurar tipos de alteração ou selecione uma CT na visualização Escolher por categoria.
   - Navegar por tipo de alteração: você pode clicar em um CT popular na área de criação rápida para abrir imediatamente a página Executar RFC. Observe que você não pode escolher uma versão mais antiga do CT com a criação rápida.

     Para classificar CTs, use a área Todos os tipos de alteração na visualização em Cartão ou Tabela. Em qualquer exibição, selecione uma CT e clique em Criar RFC para abrir a página Executar RFC. Se aplicável, a opção Criar com uma versão mais antiga aparece ao lado do botão Criar RFC.

- Escolha por categoria: selecione uma categoria, subcategoria, item e operação e a caixa de detalhes do CT será aberta com a opção Criar com uma versão mais antiga, se aplicável. Clique em Criar RFC para abrir a página Executar RFC.

3. Na página Executar RFC, abra a área do nome do CT para ver a caixa de detalhes do CT. É necessário um Assunto (preenchido se você escolher seu CT na visualização Procurar tipos de alteração). Abra a área Configuração adicional para adicionar informações sobre o RFC.

   Na área Configuração de execução, use as listas suspensas disponíveis ou insira valores para os parâmetros necessários. Para configurar parâmetros de execução opcionais, abra a área Configuração adicional.
4. Ao terminar, clique em Executar. Se não houver erros, a página RFC criada com sucesso será exibida com os detalhes da RFC enviada e a saída inicial de execução.
5. Abra a área Parâmetros de execução para ver as configurações que você enviou. Atualize a página para atualizar o status de execução do RFC. Opcionalmente, cancele a RFC ou crie uma cópia dela com as opções na parte superior da página.

Iniciando um AWS DMS tarefa de replicação com a CLI

Como funciona:

1. Use o Inline Create (você emite um `create-rfc` comando com todos os parâmetros de RFC e execução incluídos) ou o Template Create (você cria dois arquivos JSON, um para os parâmetros RFC e outro para os parâmetros de execução) e emita o `create-rfc` comando com os dois arquivos como entrada. Ambos os métodos são descritos aqui.
2. Envie o `aws amscm submit-rfc --rfc-id` *`ID`* comando RFC: com o ID RFC retornado.

   Monitore o `aws amscm get-rfc --rfc-id` *`ID`* comando RFC:.

Para verificar a versão do tipo de alteração, use este comando:

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

> **Note**
>
> Você pode usar qualquer `CreateRfc` parâmetro com qualquer RFC, independentemente de eles fazerem parte do esquema para o tipo de alteração. Por exemplo, para receber

notificações quando o status da RFC mudar, adicione essa linha `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` à parte dos parâmetros da RFC da solicitação (não aos parâmetros de execução). Para obter uma lista de todos os CreateRfc parâmetros, consulte a Referência da API de gerenciamento de alterações do AMS.

CRIAÇÃO EM LINHA:

Execute o comando create RFC com os parâmetros de execução fornecidos em linha (aspas de escape ao fornecer parâmetros de execução em linha) e, em seguida, envie a ID de RFC retornada. Por exemplo, você pode substituir o conteúdo por algo assim:

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

CRIAÇÃO DE MODELO:

1. Exiba os parâmetros de execução desse tipo de alteração em um arquivo JSON; este exemplo o nomeia StartDmsRtParams.json:

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

2. Modifique e salve o arquivo JSON dos parâmetros de execução. Por exemplo, você pode substituir o conteúdo por algo assim:

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

```
        ""
      ],
      "CdcStopPosition": [
        ""
      ]
  }
}
```

3. Envie o modelo JSON para um arquivo na sua pasta atual; este exemplo o nomeia StartDmsRtRfc.json:

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

4. Modifique e salve o StartDmsRtRfc.json arquivo. Por exemplo, você pode substituir o conteúdo por algo assim:

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

5. Crie o RFC, especificando o arquivo de parâmetros de execução e o StartDmsRtRfc arquivo:

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

   Você recebe a ID da nova RFC na resposta e pode usá-la para enviar e monitorar a RFC. Até que você o envie, o RFC permanece no estado de edição e não inicia.

Dicas

Você pode iniciar uma tarefa de AWS DMS replicação usando o console do AMS ou o AMS API/CLI. Para obter mais informações, consulte Como trabalhar com tarefas do AWS DMS.

Interromper AWS DMS tarefa de replicação

Parando um AWS DMS tarefa de replicação com o console

Captura de tela desse tipo de alteração no console AMS:

Como funciona:

1. Navegue até a página Criar RFC: No painel de navegação esquerdo do console AMS, clique em RFCs para abrir a página de lista de RFCs e, em seguida, clique em Criar RFC.
2. Escolha um tipo de alteração popular (CT) na visualização padrão Procurar tipos de alteração ou selecione uma CT na visualização Escolher por categoria.
   - Navegar por tipo de alteração: você pode clicar em um CT popular na área de criação rápida para abrir imediatamente a página Executar RFC. Observe que você não pode escolher uma versão mais antiga do CT com a criação rápida.

     Para classificar CTs, use a área Todos os tipos de alteração na visualização em Cartão ou Tabela. Em qualquer exibição, selecione uma CT e clique em Criar RFC para abrir a página Executar RFC. Se aplicável, a opção Criar com uma versão mais antiga aparece ao lado do botão Criar RFC.
   - Escolha por categoria: selecione uma categoria, subcategoria, item e operação e a caixa de detalhes do CT será aberta com a opção Criar com uma versão mais antiga, se aplicável. Clique em Criar RFC para abrir a página Executar RFC.
3. Na página Executar RFC, abra a área do nome do CT para ver a caixa de detalhes do CT. É necessário um Assunto (preenchido se você escolher seu CT na visualização Procurar tipos de alteração). Abra a área Configuração adicional para adicionar informações sobre o RFC.

   Na área Configuração de execução, use as listas suspensas disponíveis ou insira valores para os parâmetros necessários. Para configurar parâmetros de execução opcionais, abra a área Configuração adicional.
4. Ao terminar, clique em Executar. Se não houver erros, a página RFC criada com sucesso será exibida com os detalhes da RFC enviada e a saída inicial de execução.
5. Abra a área Parâmetros de execução para ver as configurações que você enviou. Atualize a página para atualizar o status de execução do RFC. Opcionalmente, cancele a RFC ou crie uma cópia dela com as opções na parte superior da página.

Parando um AWS DMS tarefa de replicação com a CLI

Como funciona:

1. Use o Inline Create (você emite um `create-rfc` comando com todos os parâmetros de RFC e execução incluídos) ou o Template Create (você cria dois arquivos JSON, um para os parâmetros

RFC e outro para os parâmetros de execução) e emita o `create-rfc` comando com os dois arquivos como entrada. Ambos os métodos são descritos aqui.

2. Envie o `aws amscm submit-rfc --rfc-id` *`ID`* comando RFC: com o ID RFC retornado.

   Monitore o `aws amscm get-rfc --rfc-id` *`ID`* comando RFC:.

Para verificar a versão do tipo de alteração, use este comando:

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

> **Note**
>
> Você pode usar qualquer `CreateRfc` parâmetro com qualquer RFC, independentemente de eles fazerem parte do esquema para o tipo de alteração. Por exemplo, para receber notificações quando o status da RFC mudar, adicione essa linha `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` à parte dos parâmetros da RFC da solicitação (não aos parâmetros de execução). Para obter uma lista de todos os CreateRfc parâmetros, consulte a Referência da API de gerenciamento de alterações do AMS.

CRIAÇÃO EM LINHA:

Execute o comando create RFC com os parâmetros de execução fornecidos em linha (aspas de escape ao fornecer parâmetros de execução em linha) e, em seguida, envie a ID de RFC retornada. Por exemplo, você pode substituir o conteúdo por algo assim:

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

CRIAÇÃO DE MODELO:

1. Exiba os parâmetros de execução desse tipo de alteração em um arquivo JSON; este exemplo o nomeia StopDmsRtParams.json:

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

2. Modifique e salve o arquivo JSON dos parâmetros de execução. Por exemplo, você pode substituir o conteúdo por algo assim:

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

3. Envie o modelo JSON para um arquivo na sua pasta atual; este exemplo o nomeia StopDmsRtRfc.json:

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

4. Modifique e salve o StopDmsRtRfc.json arquivo. Por exemplo, você pode substituir o conteúdo por algo assim:

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

5. Crie o RFC, especificando o arquivo de parâmetros de execução e o StopDmsRtRfc arquivo:

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

Você recebe a ID da nova RFC na resposta e pode usá-la para enviar e monitorar a RFC. Até que você o envie, o RFC permanece no estado de edição e não inicia.

Dicas

Você pode interromper uma tarefa de replicação do DMS usando o console do AMS ou o AMS. API/CLI Para obter mais informações, consulte Como trabalhar com tarefas do AWS DMS.

Importação de banco de dados (DB) para o AMS RDS para Microsoft SQL Server

> **Note**
>
> Os endpoints AMS API/CLI (amscm e amsskms) estão na região da AWS Norte da Virgínia,. `us-east-1` Dependendo de como sua autenticação está configurada e em qual região da AWS sua conta e seus recursos estão, talvez seja necessário adicioná-la `--region us-east-1` ao emitir comandos. Talvez você também precise adicionar`--profile saml`, se esse for o seu método de autenticação.

O processo de importação do banco de dados para o AMS RDS for SQL Server depende dos tipos de alteração do AMS (CTs) enviados como solicitações de alteração (RFCs) e usa os parâmetros da API do Amazon RDS como entrada. MicroSoft O SQL Server é um sistema de gerenciamento de banco de dados relacional (RDBMS). Para saber mais, consulte também: Amazon Relational Database Service (Amazon RDS) e rds ou referência de API do Amazon RDS.

> **Note**
>
> Certifique-se de que cada RFC seja concluída com êxito antes de passar para a próxima etapa.

Etapas de importação de alto nível:

1. Faça backup de seu banco de dados MS SQL local de origem em um arquivo.bak (backup)
2. Copie o arquivo.bak no bucket de trânsito (criptografado) do Amazon Simple Storage Service (S3)
3. Importe o arquivo.bak em um novo banco de dados em sua instância de destino do Amazon RDS MS SQL

Requisitos:

- Pilha MS SQL RDS no AMS
- Pilha RDS com opção de restauração () `SQLSERVER_BACKUP_RESTORE`
- Caçamba Transit S3
- Função do IAM com acesso ao bucket, permitindo que o Amazon RDS assuma a função
- Uma EC2 instância com o MS SQL Management Studio instalado para gerenciar o RDS (pode ser uma estação de trabalho local)

Configuração

Conclua essas tarefas para iniciar o processo de importação.

1. Envie um RFC para criar uma pilha RDS usando Deployment | Advanced stack components | RDS database stack | Create (ct-2z60dyvto9g6c). Não use o nome do banco de dados de destino (`RDSDBName`parâmetro) na solicitação de criação, o banco de dados de destino será criado durante a importação. Certifique-se de deixar espaço suficiente (`RDSAllocatedStorage`parâmetro). Para obter detalhes sobre como fazer isso, consulte o Guia de gerenciamento de alterações do AMS RDS DB Stack | Create.
2. Envie uma RFC para criar o bucket S3 de trânsito (se ainda não existir) usando Deployment | Advanced stack components | S3 storage | Create (ct-1a68ck03fn98r). Para obter detalhes sobre como fazer isso, consulte o Guia de Gerenciamento de Alterações do AMS S3 Storage | Create.
3. Envie um RFC de gerenciamento | Outro | Outro | Atualize (ct-1e1xtak34nx76) para implementar o com estes detalhes: `customer_rds_s3_role`

   No console do:

   - Assunto: “Para oferecer suporte à importação de banco de dados do MS SQL Server, implemente `customer_rds_s3_role` nesta conta.
   - Nome do bucket Transit S3:*`BUCKET_NAME`*.
   - Informações de contato:*`EMAIL`*.

   Com um ImportDbParams arquivo.json para a CLI:

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

4. Envie um RFC de gerenciamento | Outro | Outro | Atualize o RFC solicitando que o AMS defina a `SQLSERVER_BACKUP_RESTORE` opção para o RDS criado na etapa 1 (use o ID da pilha da saída da etapa 1 e a função `customer_rds_s3_role` do IAM nessa solicitação, nessa solicitação).
5. Envie uma RFC para criar uma EC2 instância (você pode usar qualquer estação/instância de trabalho existente EC2 ou local) e instale o Microsoft SQL Management Studio na instância.

Importando o banco de dados

Para importar o banco de dados (DB), siga estas etapas.

1. Faça backup de seu banco de dados local de origem usando o backup e restauração nativos do MS SQL (consulte Support for native backup and restore in SQL Server). Como resultado da execução dessa operação, você deve ter um arquivo.bak (backup).
2. Faça upload do arquivo.bak em um bucket S3 de trânsito existente usando a CLI do AWS S3 ou o console do AWS S3. Para obter informações sobre buckets S3 em trânsito, consulte Proteção de dados usando criptografia.
3. Importe o arquivo.bak em um novo banco de dados na sua instância MS SQL do RDS for SQL Server de destino (para obter detalhes sobre os tipos, consulte os tipos de instância do Amazon RDS for MySQL):

    a. Faça login na EC2 instância (estação de trabalho local) e abra o MS SQL Management Studio

    b. Conecte-se à instância RDS de destino criada como pré-requisito na etapa #1. Siga este procedimento para se conectar: Conectando-se a uma instância de banco de dados executando o Microsoft SQL Server Database Engine

    c. Inicie o trabalho de importação (restauração) com uma nova consulta SQL (Structured Query Language) (para obter detalhes sobre consultas SQL, consulte Introdução ao SQL). O nome do banco de dados de destino deve ser novo (não use o mesmo nome do banco de dados que você criou anteriormente). Exemplo sem criptografia:

    ```
    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. Verifique periodicamente o status do trabalho de importação executando essa consulta em uma janela separada:

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

Se o status mudar para Falha, procure os detalhes da falha na mensagem.

Limpeza

Depois de importar o banco de dados, talvez você queira remover recursos desnecessários, siga estas etapas.

1. Exclua o arquivo de backup (.bak) do bucket do S3. Você pode usar o console S3 para fazer isso. Para o comando da CLI para excluir um objeto de um bucket do S3, consulte rm na Referência de Comandos da CLI da AWS.
2. Exclua o bucket do S3 se você não planeja usá-lo. Para ver as etapas para fazer isso, consulte Excluir pilha.
3. Se você não planeja fazer importações do MS SQL, envie uma RFC Management | Other | Other | Update (ct-0xdawir96cy7k) e solicite que o AMS exclua a função do IAM. `customer_rds_s3_role`

Implantações de aplicativos Tier and Time no AMS

Uma implantação Tier and Tie é onde você cria, configura e implanta os recursos de uma pilha de forma independente usando componentes separados RFCs e usa os componentes IDs da pilha à medida que avança para associá-los uns aos outros.

Por exemplo, para implantar um site de alta disponibilidade (redundante) por trás de um balanceador de carga e um banco de dados, usando uma abordagem Tier and Tie, envie RFCs um banco de dados, um balanceador de carga e duas EC2 instâncias ou um grupo de Auto Scaling e configure as instâncias ou o grupo EC2 Auto Scaling com o ID do ELB que você criou.

Após a implantação dos recursos, você pode enviar uma alteração de criação de grupo de segurança para permitir que os recursos se comuniquem com o banco de dados. Para obter detalhes sobre a criação de grupos de segurança, consulte Criar grupo de segurança.

Implantações completas de aplicativos no AMS

Uma implantação do Full Stack é onde você envia uma RFC com uma CT que cria e configura tudo o que você precisa de uma só vez. Por exemplo, para implantar o site de alta disponibilidade que acabamos de descrever (EC2 instâncias, balanceador de carga e banco de dados), você usaria uma CT que, em conjunto, criasse e configurasse um grupo de Auto Scaling, um balanceador de carga, um banco de dados e as configurações do grupo de segurança necessárias para que todas as instâncias funcionassem como uma pilha. Exemplos de dois AMS CTs que fazem isso são descritos a seguir.

- Pilha de duas camadas de alta disponibilidade (ct-06mjngx5flwto): esse tipo de alteração permite criar uma pilha e configurar um grupo de Auto Scaling, banco de dados baseado em RDS, Load Balancer e aplicativo e configuração. CodeDeploy Observe que o balanceador de carga não é considerado um nível, pois é compartilhado entre vários aplicativos como um dispositivo de rede e as CodeDeploy funções também são consideradas um dispositivo. Além disso, ele cria um grupo de CodeDeploy implantação (com o nome que você dá ao CodeDeploy aplicativo) que pode ser usado para implantar seus aplicativos. As configurações do grupo de segurança para permitir que os recursos funcionem juntos são criadas automaticamente.
- Pilha de camada única de alta disponibilidade (ct-09t6q7j9v5hrn): esse tipo de alteração permite criar uma pilha e configurar um grupo de Auto Scaling e um Application Load Balancer. As configurações do grupo de segurança que permitem que os recursos funcionem juntos são criadas automaticamente.

Trabalhando com tipos de alteração de provisionamento () CTs

O AMS é responsável por sua infraestrutura gerenciada. Para fazer alterações, você deve enviar uma RFC com a classificação CT correta (categoria, subcategoria, item e operação). Esta seção descreve como encontrar CTs, determinar se alguma é adequada às suas necessidades e solicitar uma nova tomografia computadorizada, se nenhuma for.

Veja se uma tomografia computadorizada existente atende aos seus requisitos

Depois de determinar o que você deseja implantar com o AMS, a próxima etapa é estudar o existente CTs e os CloudFormation modelos para ver se já existe uma solução.

Ao criar um RFC, você deve especificar o CT. Você pode usar a Console de gerenciamento da AWS API/CLI do AMS. Exemplos de uso de ambos são descritos a seguir.

Você pode usar o console ou o API/CLI para encontrar uma ID de tipo de alteração (CT) ou versão. Existem dois métodos: pesquisar ou escolher a classificação. Para ambos os tipos de seleção, você pode classificar a pesquisa escolhendo Usado com mais frequência, Usado mais recentemente ou Alfabético.

YouTube Vídeo: Como faço para criar uma RFC usando a CLI do AWS Managed Services e onde posso encontrar o esquema de CT?

No console do AMS, na página RFCs-> Criar RFC:

- Com a opção Procurar por tipo de alteração selecionada (o padrão), você pode:
  - Use a área de criação rápida para selecionar entre as mais populares do AMS CTs. Clique em um rótulo e a página Executar RFC será aberta com a opção Assunto preenchida automaticamente para você. Preencha as opções restantes conforme necessário e clique em Executar para enviar a RFC.
  - Ou desça até a área Todos os tipos de alteração e comece a digitar um nome de CT na caixa de opções, você não precisa ter o nome exato ou completo do tipo de alteração. Você também pode pesquisar uma tomografia computadorizada por ID do tipo de alteração, classificação ou modo de execução (automático ou manual) inserindo as palavras relevantes.

    Com a visualização padrão dos cartões selecionada, os cartões CT correspondentes aparecem conforme você digita, seleciona um cartão e clica em Criar RFC. Com a exibição da tabela selecionada, escolha o CT relevante e clique em Criar RFC. Ambos os métodos abrem a página Executar RFC.
- Como alternativa, e para explorar as opções de tipo de alteração, clique em Escolher por categoria na parte superior da página para abrir uma série de caixas de opções suspensas.
- Escolha uma categoria, uma subcategoria, um item e uma operação. A caixa de informações desse tipo de alteração aparece e um painel aparece na parte inferior da página.
- Quando estiver pronto, pressione Enter e uma lista dos tipos de alteração correspondentes será exibida.
- Escolha um tipo de alteração na lista. A caixa de informações desse tipo de alteração aparece na parte inferior da página.
- Depois de ter o tipo de alteração correto, escolha Criar RFC.

> **Note**
>
> A CLI do AMS deve estar instalada para que esses comandos funcionem. Para instalar a API ou a CLI do AMS, acesse a página Recursos para desenvolvedores do console AMS. Para obter material de referência sobre a API AMS CM ou a API AMS SKMS, consulte a seção Recursos de informação do AMS no Guia do usuário. Talvez seja necessário adicionar uma `--profile` opção para autenticação; por exemplo,`aws amsskms` *`ams-cli-command`* `--profile SAML`. Talvez você também precise adicionar a `--region` opção, pois todos os comandos do AMS saem de us-east-1; por exemplo. `aws amscm` *`ams-cli-command`* `--region=us-east-1`

> **Note**
>
> Os endpoints AMS API/CLI (amscm e amsskms) estão na região da AWS Norte da Virgínia,. `us-east-1` Dependendo de como sua autenticação está configurada e em qual região da AWS sua conta e seus recursos estão, talvez seja necessário adicioná-la `--region us-east-1` ao emitir comandos. Talvez você também precise adicionar`--profile saml`, se esse for o seu método de autenticação.

Para pesquisar um tipo de alteração usando a API AMS CM (consulte ListChangeTypeClassificationSummaries) ou a CLI:

Você pode usar um filtro ou uma consulta para pesquisar. A ListChangeTypeClassificationSummaries operação tem opções de filtros para `CategorySubcategory`,`Item`, e`Operation`, mas os valores devem corresponder exatamente aos valores existentes. Para obter resultados mais flexíveis ao usar a CLI, você pode usar a `--query` opção.

Altere a filtragem de tipo com a API/CLI do AMS CM

Atributo	Valores válidos	Condição válida/padrão	Observações
ChangeTypeId	Qualquer string representando um ChangeTypeId (por	Igual	Para o tipo de alteração IDs,

Atributo	Valores válidos	Condição válida/padrão	Observações
	exemplo: ct-abc123 xyz7890)		consulte a Referência do tipo de alteração. Para o tipo de alteração IDs, consulte Encontrando um tipo de alteração ou CSIO.
Categoria	Qualquer texto de formato livre	Contém	Expressões regulares em cada campo individual não são suportadas. Pesquisa sem distinção entre maiúsculas e min
Subcategory			
Item			
Operação			

1. Aqui estão alguns exemplos de classificações de tipo de alteração de listagem:

   O comando a seguir lista todas as categorias de tipo de alteração.

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

   O comando a seguir lista as subcategorias pertencentes a uma categoria especificada.

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

   O comando a seguir lista os itens pertencentes a uma categoria e subcategoria especificadas.

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

2. Aqui estão alguns exemplos de pesquisa de tipos de alteração com consultas CLI:

   O comando a seguir pesquisa resumos de classificação de CT para aqueles que contêm “S3" no nome do item e cria a saída da categoria, subcategoria, item, operação e ID do tipo de alteração em forma de tabela.

```
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. Em seguida, você pode usar o ID do tipo de alteração para obter o esquema de CT e examinar os parâmetros. O comando a seguir gera o esquema em um arquivo JSON chamado Creates3Params.schema.json.

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

Para obter informações sobre o uso de consultas CLI, consulte Como filtrar a saída com a opção --query e a referência da linguagem de consulta, Especificação. JMESPath

4. Depois de ter o ID do tipo de alteração, recomendamos verificar a versão do tipo de alteração para garantir que seja a versão mais recente. Use esse comando para encontrar a versão de um tipo de alteração especificado:

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

Para encontrar o `AutomationStatus` para um tipo de alteração específico, execute este comando:

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

Para encontrar o `ExpectedExecutionDurationInMinutes` para um tipo de alteração específico, execute este comando:

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

Depois de encontrar uma CT que considere apropriada, examine os parâmetros de execução do esquema JSON associados a ela para saber se ela aborda seu caso de uso.

Use esse comando para gerar um esquema CT em um arquivo JSON com o nome do CT; este exemplo gera o esquema de armazenamento Create S3:

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

Vamos dar uma olhada mais de perto no que esse esquema oferece.

Esquema de criação do S3 Bucket

```
{
  "$schema": "http://json-schema.org/draft-04/sch
ema#",
"name": "Create S3 Storage
"description": "Use to create an Amazon Simple
 Storage Service stack.",
  "type": "object",
  "properties": {
    "Description": {
      "description": "The description of the
 stack.",
      "type": "string",
      "minLength": 1,
      "maxLength": 500
    },
    "VpcId": {
      "description": "ID of the VPC to create the S3
 Bucket in, in the form vpc-a1b2c3d4e5f67890e.",
      "type": "string",
      "pattern": "^vpc-[a-z0-9]{17}$"
    },
    "StackTemplateId": {
      "description": "Required value: stm-s2b72
beb000000000.",
```

O esquema começa com a CT (“descrição”), que informa para que serve o esquema. Nesse caso, para criar uma pilha de armazenamento S3.

Em seguida, você tem propriedades obrigatórias e opcionais que você pode especificar. Os valores padrão das propriedades são fornecidos. As propriedades obrigatórias estão listadas no final do esquema.

Na StackTemplateId área, você vê que há um modelo de pilha específico para esse CT e esquema, e seu ID é um valor de propriedade obrigatór io.

```
      "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": {
            "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
    },
```

O esquema permite marcar a pilha que você está criando, para fins de contabilidade interna. Além disso, algumas opções, como backup, exigem uma tag de key:BACKUP e value:TRUE. Para obter informações detalhadas, leia Como marcar seus recursos da Amazon EC2 .

```
    "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]$",
          "minLength": 3,
          "maxLength": 63
        }
      },
      "additionalProperties": false,
      "required": [
        "AccessControl",
        "BucketName"
      ]
    }
  },
  "additionalProperties": false,
  "required": [
    "Description",
    "VpcId",
    "StackTemplateId",
    "Name",
    "TimeoutInMinutes",
    "Parameters"
  ]
```

A seção Parâmetros do esquema CT JSON é onde você fornece os parâmetros de execução.

Para esse esquema, somente a ACL e os parâmetros de execução BucketName são obrigatórios.

```
}
```

Solicite um novo CT

Depois de examinar o esquema, você pode decidir que ele não fornece parâmetros suficientes para criar a implantação desejada. Se for esse o caso, examine os CloudFormation modelos existentes para encontrar um que seja mais próximo do que você deseja. Depois de saber quais parâmetros adicionais você precisa, envie um Management | Other | Other | Create CT.

> **Note**
>
> Todos os outros | Outros Create and Update CTs recebem a atenção de um operador de AMS, que entrará em contato com você para discutir o novo CT.

Para enviar uma solicitação para um novo CT, acesse o console do AMS por meio do console normal Console de gerenciamento da AWSe siga estas etapas.

1. Na navegação à esquerda, clique em RFCs.

   A página RFCs do painel é aberta.
2. Clique em Criar.

   A página Criar uma solicitação de alteração é aberta.
3. Selecione Gerenciamento na lista suspensa Categoria e Outro para Subcategoria e Item. Para a Operação, escolha Criar. O RFC precisará de aprovação antes de ser implementado.
4. Insira as informações do motivo pelo qual você deseja a CT, por exemplo: Solicitando uma CT de armazenamento Create S3 modificada que permita a personalização ACLs, com base na CT de armazenamento Create S3 existente. Isso deve resultar em uma nova CT: Implantação | Componentes avançados de pilha | Armazenamento S3 | Criar ACL personalizada do S3. Essa nova tomografia computadorizada pode ser pública.
5. Clique em Enviar.

   Seu RFC é exibido no painel do RFC.

Teste o novo CT

Depois que o AWS Managed Services criar essa nova CT, você a testa enviando uma RFC com ela. Se você trabalhou com o AMS para pré-aprovar o novo CT, basta seguir um envio padrão de RFC e observar o resultado (para obter detalhes sobre o envio RFCs, consulte Criação e envio de um RFC). Se a nova CT não for pré-aprovada (você quer ter certeza de que ela nunca será executada sem aprovação explícita), você precisará discutir sua implementação com o AMS sempre que quiser executá-la.

Começos rápidos

Tópicos

- Início rápido do AMS Resource Scheduler
- Configurando backups entre contas (dentro da região)

Usando uma combinação de tipos de alteração do AMS, você pode realizar tarefas complexas.

Você pode usar o sistema de gerenciamento de alterações do AMS para configurar o AMS Resource Scheduler, para uma zona de pouso com várias contas (MALZ) ou para uma conta de zona de pouso com uma única conta (SALZ). O processo varia. Além disso, para fazer transferências de arquivos e instantâneos entre contas.

Início rápido do AMS Resource Scheduler

Use este guia de início rápido para implementar o AMS Resource Scheduler, um agendador de instâncias baseado em tags para economizar custos no AMS Advanced.

O AMS Resource Scheduler é baseado no AWS Instance Scheduler.

Terminologia do AMS Resource Scheduler

Antes de começar, é bom se familiarizar com a terminologia do AMS Resource Scheduler:

- período: cada agendamento deve conter pelo menos um período que defina o (s) horário (s) em que a instância deve ser executada. Um cronograma pode conter mais de um período. Quando mais de um período é usado em um cronograma, o Agendador de Recursos aplica a ação inicial apropriada quando pelo menos uma das regras do período é verdadeira.
- fuso horário: para obter uma lista de valores de fuso horário aceitáveis a serem usados no DefaultTimezoneparâmetro referenciado posteriormente, consulte a coluna TZ da Lista de fusos horários do banco de dados TZ.
- hibernar: quando configuradas como verdadeiras, as EC2 instâncias habilitadas para hibernação e que atendem aos requisitos de hibernação são hibernadas (). suspend-to-disk Verifique o EC2 console para descobrir se suas instâncias estão habilitadas para hibernação. Use a hibernação para EC2 instâncias paradas da Amazon que executam o Amazon Linux.

- obrigatório: quando definido como verdadeiro, com base no cronograma definido, o Agendador de Recursos interrompe um recurso em execução se ele for iniciado manualmente fora do período de execução e inicia um recurso se ele for interrompido manualmente durante o período de execução.
- retain_running: quando definido como true, impede que o Resource Scheduler interrompa uma instância no final de um período de execução se a instância tiver sido iniciada manualmente antes do início do período. Por exemplo, se uma instância com um período configurado que vai das 9h às 17h for iniciada manualmente antes das 9h, o Agendador de Recursos não interromperá a instância às 17h.
- ssm-maintenance-window: adicione uma janela AWS Systems Manager de manutenção como um período de execução a uma programação. Quando você especifica o nome de uma janela de manutenção que existe na mesma conta e região da AWS da sua pilha implantada para programar suas EC2 instâncias da Amazon, o Resource Scheduler iniciará a instância antes do início da janela de manutenção e interromperá a instância no final da janela de manutenção, se nenhum outro período de execução especificar que a instância deve ser executada e se o evento de manutenção for concluído.

  O Resource Scheduler usa a AWS Lambda frequência especificada durante a configuração inicial para determinar quanto tempo antes da janela de manutenção sua instância deve ser iniciada. Se você definir o AWS CloudFormation parâmetro Frequência para 10 minutos ou menos, o Agendador de Recursos iniciará a instância 10 minutos antes da janela de manutenção. Se você definir a frequência para mais de 10 minutos, o Agendador de Recursos iniciará a instância com o mesmo número de minutos que a frequência especificada. Por exemplo, se você definir a frequência da janela de manutenção do Systems Manager como 30 minutos, os Resource Schedulers iniciarão a instância 30 minutos antes da janela de manutenção.

  Para obter mais informações, consulte Janelas AWS Systems Manager de manutenção.
- override-status: substitui temporariamente as ações de início e término do agendamento configuradas pelo Agendador de Recursos. Se você definir o campo como sendo executado, o Agendador de Recursos iniciará, mas não interromperá, a instância aplicável. A instância é executada até que você a interrompa manualmente. Se você definir o status de substituição como interrompido, o Agendador de Recursos interrompe, mas não inicia a instância aplicável. A instância não é executada até que você a inicie manualmente.

Implementação do AGENDADOR DE RECURSOS

Para implantar uma solução de agendador de recursos do AMS, siga estas etapas.

1. Envie um RFC Deployment | AMS Resource Scheduler | Solution | Deploy (ct-0ywnhc8e5k9z5) e forneça os seguintes parâmetros:

   - SchedulingActive: Sim para habilitar o agendamento de recursos, Não para desabilitar. O padrão é Sim.
   - ScheduledServices: insira uma lista de serviços separados por vírgulas para os quais agendar recursos. Os valores válidos incluem uma combinação de escalonamento automático, ec2 e rds. O padrão é escalonamento automático, ec2, rds.
   - TagName: o nome da chave de tag que associa esquemas de agendamento de recursos a recursos de serviço. O padrão é Programação.

     > **Note**
     >
     > Sua implantação do Resource Scheduler só funcionará em recursos que tenham essa tag.

   - DefaultTimezone: o nome do fuso horário, no formato EUA/Pacífico, a ser usado como fuso horário padrão. O padrão é UTC.
2. Depois de receber uma confirmação de que a RFC na primeira etapa foi executada com êxito, você pode enviar o tipo de alteração Período | Adicionar.
3. Por fim, envie uma RFC para adicionar uma programação ao período que foi criado na etapa dois. Use o Cronograma | Adicionar tipo de alteração.

Implantação e uso do AMS Resource Scheduler FAQs

Perguntas frequentes sobre o AMS Resource Scheduler.

P: O que acontece se eu ativar a hibernação, mas a EC2 instância não oferecer suporte a ela?

R: A hibernação salva o conteúdo da memória da instância (RAM) no volume raiz do Amazon Elastic Block Store (Amazon EBS). Se esse campo for definido como verdadeiro, as instâncias serão hibernadas quando o Resource Scheduler as interromper.

Se você definir o Resource Scheduler para usar a hibernação, mas suas instâncias não estiverem habilitadas para hibernação ou não atenderem aos pré-requisitos de hibernação, o Resource Scheduler registrará um aviso e as instâncias serão interrompidas sem hibernação. Para obter mais informações, consulte Hibernate Your Instance.

P: O que acontece se eu definir override_status e forced?

R: Se você definir override_status como running e definir enforced como true (evita que uma instância seja iniciada manualmente fora de um período de execução), o Resource Scheduler interrompe a instância.

Se você definir override_status como stop e definir enforced como true (evita que uma instância seja interrompida manualmente durante um período de execução), o Resource Scheduler reinicia a instância.

> **Note**
>
> Se obrigatório for falso, o comportamento de substituição configurado será aplicado.

P: Depois que o Agendador de Recursos do AMS é implantado, como faço para desativar ou ativar o agendador de recursos na minha conta?

R: Para desativar ou ativar o Agendador de Recursos do AMS:

- Para desativar: Crie um RFC usando State | Disable. Certifique-se de definir o como SchedulerStateDESATIVAR
- Para habilitar: Crie um RFC usando State | Enable. Certifique-se de definir o como SchedulerStateHABILITAR

P O que acontece se o período do AMS Resource Scheduler estiver dentro da minha janela de manutenção de patches?

R: O Resource Scheduler funciona com base em seus agendamentos configurados. Se estiver configurado para interromper uma instância enquanto a aplicação de patches estiver em andamento, ela interromperá a instância, a menos que a janela de correção seja adicionada como um período ao cronograma antes do início da correção. Em outras palavras, o Resource Scheduler não inicia automaticamente nenhuma instância parada para aplicação de patches, a menos que um período designado seja configurado. Para evitar conflitos com sua janela de manutenção de patches, adicione a janela de tempo alocada para a aplicação de patches ao cronograma do Resource Scheduler como um período. Para adicionar um período ao cronograma existente, crie um RFC usando Período | Adicionar.

P Se eu precisar ter uma programação diferente para EC2 instâncias diferentes, posso ter mais de uma configuração de agendamento dentro da minha conta?

R: Sim, você pode criar várias agendas. Cada cronograma pode ter vários períodos com base no requisito. Quando o Agendador de Recursos do AMS está ativado na conta, uma chave de tag é configurada. Por exemplo, se a chave da etiqueta for “Programação”, o valor da etiqueta pode diferir com base em diferentes agendas, o que corresponde ao nome da agenda do Agendador de Recursos do AMS. Para adicionar uma nova agenda, você pode criar uma RFC usando o tipo de alteração Management | AMS Resource Scheduler | Schedule | Add (ct-2bxelbn765ive), consulte Programação | Adicionar.

P: Onde posso encontrar todos os diferentes tipos de alteração compatíveis com o AMS Resource Scheduler?

R: O AMS tem tipos de alteração do Agendador de Recursos para implantar o Agendador de Recursos do AMS em sua conta; ativá-lo ou desativá-lo; definir, adicionar, atualizar e excluir horários e períodos para usar com ele; e descrever (obter uma descrição detalhada) dos horários e períodos.

Configurando backups entre contas (dentro da região)

AWS Backup suporta a capacidade de copiar snapshots de uma conta para outra dentro da mesma região da AWS, desde que as duas contas estejam dentro da mesma organização da AWS. Por exemplo, no AMS Advanced multi-account landing zone (MALZ), você pode configurar uma cópia instantânea de várias contas dentro da mesma região da AWS usando esse início rápido.

Para obter mais informações, consulte AWS Backup e AWS Organizations trazem recurso de backup entre contas

Você copia instantâneos entre contas para recuperação de desastres (DR). Você pode ter requisitos para manter snapshots dentro da mesma região da AWS, mas fora dos limites da conta, para proteção de dados.

Visão geral:

Em um alto nível, estas são as etapas para backups entre contas no AMS:

- Crie uma conta de destino para hospedar backups na região da AWS em que sua landing zone do AMS está hospedada (etapa 1)
- Crie uma chave KMS para criptografar backups na conta de destino (etapa 3)

- Crie um cofre de backup na conta de destino da mesma região da sua landing zone do AMS Advanced (etapa 4)
- Ative a configuração de várias contas em sua conta de gerenciamento (etapa 5)
- Criar ou modificar o plano e as regras de backup da conta de origem (etapa 6)

> **Note**
>
> Certifique-se de que as contas de origem e de destino estejam na mesma região. Se você quiser copiar seus backups entre regiões, entre em contato com sua CA ou CSDM.

Para ativar e configurar backups entre contas:

1. Crie uma conta de destino para hospedar backups; se você já tiver essa conta, pode pular esta etapa. Para criar a conta, envie uma RFC da sua conta Management Payer usando o tipo de alteração Deployment | Managed landing zone | Management account | Create application account (com VPC) (ct-1zdasmc2ewzrs).
2. [Opcional] Se recursos ou instantâneos estiverem criptografados na conta de origem (por exemplo, Prod), compartilhe a chave KMS usada para criptografia com a conta de destino. Para fazer isso, envie uma RFC usando o Gerenciamento | Componentes avançados da pilha | Chave KMS | Tipo de alteração de atualização (ct-3ovo7px2vsa6n).
3. Na conta de destino, crie uma chave KMS para ser usada na criptografia do Backup Vault. Para fazer isso, envie um RFC usando Deployment | Advanced stack components | KMS key | Create (auto) change type (ct-1d84keiri1jhg).
4. Na conta de destino, crie um Cofre de Backup usando a chave criada anteriormente. Os AWS Backup Vaults podem ser criados usando o tipo de alteração automática CFN ingest, Deployment | Inestion | Stack from CloudFormation Template | Create (ct-36cn2avfrrj9v). Na mesma solicitação, a política de acesso ao cofre precisa ser modificada para permitir que as contas de origem acessem o cofre. Aqui está um exemplo de política:

   CloudFormation Modelo de exemplo para um Backup Vault:

```
{
  "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. Na sua conta do Management Payer, ative o Cross-Account backup. Para fazer isso, envie uma RFC usando o plano Gerenciamento | AWS Backup | Backup | Habilitar cópia entre contas (Conta de gerenciamento) do tipo de alteração (ct-2yja7ihh30ply).
6. Por fim, na conta de origem em que os backups são originados, crie a regra ou as regras do plano de backup que regem os backups para copiar instantâneos entre contas. Para fazer isso, envie um RFC usando o plano Deployment | AWS Backup | Backup Backup | Create change type (ct-2hyozbpa0sx0m). Se precisar atualizar um plano de backup existente, envie uma RFC usando o tipo de alteração Management | Other | Other | Update (ct-0xdawir96cy7k) com as seguintes informações:

    1. O nome do plano de backup, bem como o nome da regra a ser atualizada.
    2. O ARN do cofre de backup da destination/ICE conta.
    3. A retenção para a days/months qual você gostaria de manter os instantâneos no cofre ICE de destino.

Tutoriais

Tópicos

- Tutorial do console: pilha de dois níveis de alta disponibilidade (Linux/RHEL)
- Tutorial do console: implantando um site Tier and Tie WordPress
- Tutorial de CLI: pilha de duas camadas de alta disponibilidade (Linux/RHEL)
- Tutorial de CLI: Implantação de um site Tier and Tie WordPress

Os tutoriais a seguir detalham as etapas para criar uma pilha de dois níveis com alta disponibilidade (ct-06mjngx5flwto), usar a CLI e usar o console e implantar um grupo Amazon Auto Scaling (ASG) Linux ou RHEL. EC2 Um tier-and-tie tutorial semelhante segue cada um (um para o console e outro para a CLI), que usa um tutorial separado CTs, criado em uma ordem que permite unir recursos à medida que são criados.

As descrições de todas as opções de tomografia computadorizada, inclusive, ChangeTypeId podem ser encontradas em managedservices/latest/ctref /Change Type Reference.

Tutorial do console: pilha de dois níveis de alta disponibilidade (Linux/RHEL)

Esta seção descreve como implantar um WordPress site de alta disponibilidade (HA) em um ambiente AMS usando o console AMS.

> **Note**
>
> Esse passo a passo de implantação foi testado em ambientes AMZN Linux e RHEL.

Resumo das tarefas e tarefas necessárias RFCs:

1. Crie infraestrutura (pilha HA de dois níveis)
2. Crie um bucket S3 para aplicativos CodeDeploy
3. Crie o pacote de WordPress aplicativos e faça o upload para o bucket do S3
4. Implante o aplicativo com CodeDeploy
5. Acesse o WordPress site e faça login para validar a implantação

6. Destrua a implantação

As descrições de todas as opções de tomografia computadorizada ChangeTypeId, inclusive, podem ser encontradas na Referência de tipo de alteração do AMS.

Antes de começar

O Deployment | Advanced Stack Components | High Availability Two Tier Stack | Create CT cria um grupo de Auto Scaling, um balanceador de carga, um banco de dados e CodeDeploy um nome de aplicativo e grupo de implantação (com o mesmo nome que você dá ao aplicativo). Para obter informações sobre, CodeDeploy consulte O que é CodeDeploy?

Este passo a passo usa um RFC de pilha de duas camadas de alta disponibilidade que inclui `UserData` e também descreve como criar um WordPress pacote que pode ser implantado. CodeDeploy

O exemplo `UserData` mostrado no exemplo obtém metadados da instância, como ID da instância, região etc., de dentro de uma instância em execução, consultando o serviço de metadados da EC2 instância disponível em http://169.254.169.254/latest/meta-data/. Essa linha no script de dados do usuário:`REGION=$(curl 169.254.169.254/latest/meta-data/placement/availability-zone/ | sed 's/[a-z]$//')`, recupera o nome da zona de disponibilidade do serviço de metadados na variável $REGION de nossas regiões suportadas e a usa para preencher a URL do bucket do S3 em que o agente é baixado. CodeDeploy O IP 169.254.169.254 é roteável somente dentro da VPC (todos podem consultar o serviço). VPCs Para obter informações sobre o serviço, consulte Metadados da instância e dados do usuário. Observe também que os scripts inseridos como UserData são executados como usuário “root” e não precisam usar o comando “sudo”.

Essa explicação passo a passo deixa os seguintes parâmetros no valor padrão (mostrado):

- Grupo de Auto Scaling:. `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`

- Balanceador de carga:. `HealthCheckInterval=30, HealthCheckTimeout=5`
- Banco de dados:`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`.
- Aplicação:`DeploymentConfigName=CodeDeployDefault.OneAtATime`.

Parâmetros variáveis:

O console fornece uma opção ASAP para a hora de início e este passo a passo recomenda usá-la. O ASAP faz com que o RFC seja executado assim que as aprovações forem aprovadas.

> **Note**
>
> Há muitos parâmetros que você pode escolher definir de forma diferente dos mostrados. Os valores desses parâmetros mostrados no exemplo foram testados, mas podem não ser adequados para você. Somente os valores obrigatórios são mostrados nos exemplos. Os valores na *`replaceable`* fonte devem ser alterados, pois são específicos da sua conta.

Crie a infraestrutura

Esse procedimento utiliza a CT de pilha de duas camadas de alta disponibilidade seguida pela CT de armazenamento Create S3.

Coletar os dados a seguir antes de começar fará com que a implantação seja mais rápida.

OS DADOS NECESSÁRIOS TÊM UMA PILHA:

- AutoScalingGroup:
  - UserData: esse valor é fornecido neste tutorial. Ele inclui comandos para configurar o recurso CodeDeploy e iniciar o CodeDeploy agente.
  - AMI-ID: esse valor determina o sistema operacional das EC2 instâncias que seu grupo de Auto Scaling (ASG) criará. Selecione uma AMI em sua conta que comece com “cliente-” e seja do

sistema operacional que você deseja. Encontre a AMI IDs no console do AMS VPCs -> página de VPCs detalhes. Este passo a passo é para ASGs configurar o uso de uma AMI Amazon Linux ou RHEL.

- Banco de dados:
  - Esses parâmetros, DBEngine, EngineVersion, e LicenseModeldevem ser definidos de acordo com sua situação, embora os valores mostrados no exemplo tenham sido testados. O tutorial usa esses valores, respectivamente:*MySQL*,*8.0.16*,*general-public-license*.
  - Esses parâmetros, DBName, MasterUserPassword, e MasterUsernamesão necessários ao implantar o pacote de aplicativos. O tutorial usa esses valores, respectivamente:*wordpressDB*,*p4ssw0rd*,*admin*. Observe que só DBName pode conter caracteres alfanuméricos.
  - Quando você insere o MasterUsernamepara o banco de dados do RDS, ele aparecerá em texto não criptografado, então faça login no banco de dados o mais rápido possível e altere a senha para garantir sua segurança.
  - Para RDSSubnetIDs, use duas sub-redes privadas. Digite-os um de cada vez, pressionando “Enter” após cada um. Encontre Subnet IDs com a referência da API For the AMS SKMS, consulte a guia Relatórios na operação do AWS Artifact Console. (CLI list-subnet-summaries:) ou no console AMS -> página de detalhes da VPC. VPCs
- LoadBalancer:
  - Defina esse parâmetro, Public como true, pois o tutorial usa sub-redes públicas do ELB.
  - ELBSubnetIDs: Use duas sub-redes públicas. Digite-os um de cada vez, pressionando “Enter” após cada um. Encontre Subnet IDs com a referência da API For the AMS SKMS, consulte a guia Relatórios na operação do AWS Artifact Console. (CLI list-subnet-summaries:) ou no console AMS -> página de detalhes da VPC. VPCs
- Aplicativo: o ApplicationNamevalor define o nome do CodeDeploy aplicativo e o nome do grupo de CodeDeploy implantação. Você o usa para implantar seu aplicativo. Ele deve ser exclusivo na conta. Para verificar os CodeDeploy nomes da sua conta, consulte o CodeDeploy console. O exemplo usa, *WordPress* mas, se você usar esse valor, certifique-se de que ele ainda não esteja em uso.

1. Inicie a pilha de alta disponibilidade.

   a. Na página Criar RFC, selecione a categoria Implantação, a subcategoria Pilhas padrão, o item Pilha de duas camadas de alta disponibilidade e a operação Criar, na lista.

b. IMPORTANTE: Escolha Avançado e defina os valores conforme mostrado.

Você só precisa inserir valores para as opções marcadas com estrela (*); os valores testados são mostrados no exemplo; você pode deixar as opções vazias não obrigatórias em branco.

c. Para a seção Descrição do RFC:

```
Subject: WP-HA-2-Tier-RFC
```

d. Para a seção Informações sobre recursos, defina parâmetros para AutoScalingGroupBanco de Dados LoadBalancer, Aplicativo e Tags.

Além disso, o objetivo da chave de tag AppName "" é que você possa pesquisar facilmente as instâncias do ASG no EC2 console; você pode chamar essa chave de tag de “Nome” ou qualquer outro nome de chave que desejar. Observe que você pode adicionar até 50 tags.

```
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. Clique em Enviar quando terminar.

2. Faça login no banco de dados que você criou e altere a senha.
3. Inicie um bucket Stack do S3.

   Coletar os dados a seguir antes de começar fará com que a implantação seja mais rápida.

   BUCKET S3 DE DADOS NECESSÁRIO:

   - VPC-ID: esse valor determina onde seu S3 Bucket estará. Encontre uma VPC IDs com a referência da API For the AMS SKMS, consulte a guia Relatórios na operação do AWS Artifact Console. (CLI:) ou na página do console AMS. list-vpc-summaries VPCs
   - BucketName: esse valor define o nome do S3 Bucket, você o usa para carregar seu pacote de aplicativos. Ele deve ser exclusivo em toda a região da conta e não pode incluir letras maiúsculas. Incluir o ID da sua conta como parte do não BucketName é obrigatório, mas facilita a identificação posterior do bucket. Para ver quais nomes de bucket do S3 existem na conta, acesse o console do Amazon S3 da sua conta.

   a. Na página Criar RFC, selecione a categoria Implantação, a subcategoria Advanced Stack Components, o item Armazenamento S3 e a operação Criar na lista de opções RFC CT.

   b. Mantenha a opção Básica padrão e defina os valores conforme mostrado.

```
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. Clique em Enviar quando terminar. O bucket implantado com esse tipo de alteração permite read/write acesso total a toda a conta.

Crie, carregue e implante o aplicativo

Primeiro, crie um pacote de WordPress aplicativos e, em seguida, use o CodeDeploy CTs para criar e implantar o aplicativo.

1. Baixe WordPress, extraia os arquivos e crie um. diretório /scripts.

   Comando Linux:

   ```
   wget https://github.com/WordPress/WordPress/archive/master.zip
   ```

   Windows: cole `https://github.com/WordPress/WordPress/archive/master.zip` em uma janela do navegador e baixe o arquivo zip.

   Crie um diretório temporário no qual montar o pacote.

   Linux

   ```
   mkdir /tmp/WordPress
   ```

   Windows: Crie um diretório WordPress "", você usará o caminho do diretório posteriormente.

2. Extraia a WordPress fonte para o diretório WordPress "" e crie um. diretório /scripts.

   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: vá para o diretório WordPress "" que você criou e crie um diretório de “scripts” lá.

Se você estiver em um ambiente Windows, certifique-se de definir o tipo de interrupção dos arquivos de script como Unix (LF). No Notepad ++, essa é uma opção na parte inferior direita da janela.

3. Crie o arquivo CodeDeploy appspec.yml, no WordPress diretório (se estiver copiando o exemplo, verifique o recuo, cada espaço conta). IMPORTANTE: Certifique-se de que o caminho de “origem” esteja correto para copiar os WordPress arquivos (nesse caso, em seu WordPress diretório) para o destino esperado (/var/www/html/WordPress). No exemplo, o arquivo appspec.yml está no diretório com os WordPress arquivos, portanto, somente “/” é necessário. Além disso, mesmo que você tenha usado uma AMI RHEL para seu grupo de Auto Scaling, deixe a linha “os: linux” como está. Exemplo de arquivo appspec.yml:

```
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. Crie scripts de arquivo bash no WordPress . diretório /scripts.

   Primeiro, crie `config_wordpress.sh` com o conteúdo a seguir (se preferir, você pode editar o arquivo wp-config.php diretamente).

> **Note**
>
> *DBName* Substitua pelo valor fornecido no HA Stack RFC (por exemplo, `wordpress`).
> *DB_MasterUsername* Substitua pelo `MasterUsername` valor fornecido no HA Stack RFC (por exemplo, `admin`).
> *DB_MasterUserPassword* Substitua pelo `MasterUserPassword` valor fornecido no HA Stack RFC (por exemplo, `p4ssw0rd`).
> *DB_ENDPOINT* Substitua pelo nome DNS do endpoint nas saídas de execução do HA Stack RFC (por exemplo,). `srt1cz23n45sfg.clgvd67uvydk.us-east-1.rds.amazonaws.com` Você pode encontrar isso na GetRfcoperação (CLI: get-rfc --rfc-id RFC_ID) ou na página de detalhes do RFC do console AMS para o HA Stack RFC que você enviou anteriormente.

```
#!/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. No mesmo diretório, crie `install_dependencies.sh` com o seguinte conteúdo:

```
#!/bin/bash
yum install -y php
yum install -y php-mysql
yum install -y mysql
service httpd restart
```

> **Note**
>
> O HTTPS é instalado como parte dos dados do usuário no lançamento para permitir que as verificações de saúde funcionem desde o início.

6. No mesmo diretório, crie `start_server.sh` com o seguinte conteúdo:

- Para instâncias do Amazon Linux, use isso:

```
#!/bin/bash
service httpd start
```

- Para instâncias do RHEL, use isso (os comandos extras são políticas que permitem que o SELINUX aceite): 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. No mesmo diretório, crie `stop_server.sh` com o seguinte conteúdo:

```
#!/bin/bash
service httpd stop
```

8. Crie o pacote zip.

   Linux

```
$ cd /tmp/WordPress
$ zip -r wordpress.zip .
```

   Windows: Vá para o diretório WordPress "", selecione todos os arquivos e crie um arquivo zip, não se esqueça de chamá-lo de wordpress.zip.

1. Faça o upload do pacote de aplicativos para o bucket do S3

   O pacote precisa estar pronto para continuar implantando a pilha.

   Você tem acesso automático a qualquer instância de bucket do S3 que você criar. Você pode acessá-lo por meio de seu Bastions (consulte Instâncias de acesso) ou por meio do console S3 e fazer o upload do CodeDeploy pacote com drag-and-drop, ou navegando até o arquivo e selecionando-o.

Você também pode usar o seguinte comando em uma janela de shell; verifique se você tem o caminho correto para o arquivo zip:

```
aws s3 cp wordpress/wordpress.zip s3://BUCKET_NAME/
```

2. Implantar o pacote WordPress CodeDeploy de aplicativos

   CÓDIGO DE DADOS NECESSÁRIO/IMPLANTAÇÃO DO APLICATIVO:

   - CodeDeployApplicationName: O nome que você deu ao CodeDeploy aplicativo.
   - CodeDeployGroupName: Como o CodeDeploy aplicativo e o grupo foram criados a partir do nome que você deu ao CodeDeploy aplicativo na RFC da pilha HA, esse é o mesmo nome do. CodeDeployApplicationName
   - S3Bucket: O nome que você deu ao bucket S3.
   - S3 BundleType e S3Key: fazem parte do pacote de WordPress aplicativos que você implantou.
   - VpcId: A VPC relevante.

   a. Na página Criar RFC, selecione a categoria Implantação, subcategoria Aplicativos, item CodeDeploy aplicativo e operação Implantar na lista de opções RFC CT.

   b. Mantenha a opção Básica padrão e defina os valores conforme mostrado.

   > **Note**
   >
   > Faça referência ao CodeDeploy aplicativo, ao grupo de CodeDeploy implantação, ao bucket S3 e ao pacote criados anteriormente.

```
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. Clique em Enviar quando terminar.

Validar a implantação do aplicativo

Navegue até o endpoint (LoadBalancerCName) do balanceador de carga criado anteriormente, com o WordPress caminho implantado:/. WordPress Por exemplo:

```
http://stack-ID-FOR-ELB.us-east-1.elb.amazonaws.com/WordPress
```

Você deve ver uma página como esta:

Elimine a implantação de alta disponibilidade

Para reduzir a implantação, você envia o Delete Stack CT para a pilha HA de dois níveis e para o bucket S3, e pode solicitar que os snapshots do RDS sejam excluídos (eles são excluídos automaticamente após dez dias, mas custam uma pequena quantia enquanto estão lá). Reúna a pilha IDs para a pilha HA e o bucket S3 e siga estas etapas. Consulte Stack | Delete.

Tutorial do console: implantando um site Tier and Tie WordPress

Esta seção descreve como implantar um WordPress site de alta disponibilidade (HA) em um ambiente AMS usando o console AMS. Esse conjunto de instruções inclui um exemplo de criação do arquivo de pacote WordPress CodeDeploy compatível necessário (por exemplo, zip). O provisionamento dos recursos segue uma ordem que permite uni-los para formar “camadas”.

> **Note**
>
> Este passo a passo de implantação foi projetado para uso com um sistema operacional AMZN Linux.

Os parâmetros essenciais da variável são anotados como`replaceable`; no entanto, talvez você queira modificar outros parâmetros para se adequar à sua situação.

Resumo das tarefas e tarefas necessárias RFCs:

1. Crie a infraestrutura:
   a. Crie um cluster de banco de dados MySQL RDS
   b. Criar um balanceador de carga
   c. Crie um grupo de Auto Scaling e vincule-o ao balanceador de carga
   d. Crie um bucket S3 para aplicativos CodeDeploy
2. Crie um pacote de WordPress aplicativos (não requer um RFC)
3. Implante o pacote de WordPress aplicativos com CodeDeploy:
   a. Crie um CodeDeploy aplicativo
   b. Crie um grupo CodeDeploy de implantação
   c. Carregue seu pacote de WordPress aplicativos no bucket do S3 (não requer um RFC)
   d. Implemente o CodeDeploy aplicativo
4. Valide a implantação
5. Destrua a implantação

As descrições de todas as opções de tomografia computadorizada, inclusive, ChangeTypeId podem ser encontradas na Referência de tipo de alteração do AMS.

Criação de um RFC usando o console (noções básicas)

Essas são algumas etapas que você deve seguir sempre que criar uma RFC usando o console.

1. Escolha RFCsno painel de navegação esquerdo para abrir a página da RFCs lista e, em seguida, escolha Criar RFC.

   A página Criar RFC é aberta.
2. Escolha Procurar tipos de alteração (o padrão) ou Escolher por categoria.
3. Procure os tipos de alteração:

a. Escolha uma opção de criação rápida para iniciar uma RFC com um dos tipos de alteração mais usados.

   A área de configuração geral desse tipo de alteração é aberta e a linha de assunto é preenchida. Para ver os detalhes do tipo de alteração, abra a área na parte superior da página.

b. Use a área Todos os tipos de alteração.

   Filtre, alterne entre uma exibição de cartas ou tabela ou classifique os tipos de alteração. Quando você encontrar o que deseja, selecione-o e escolha Criar RFC na parte superior da página.

   A área de configuração geral desse tipo de alteração é aberta e a linha de assunto é preenchida. Para ver os detalhes do tipo de alteração, abra a área na parte superior da página.

4. Escolha por categoria:

   a. Selecione a categoria, a subcategoria, o item e a operação apropriados.

      A caixa de detalhes do tipo de alteração aparece na parte inferior da página.

   b. Escolha Criar RFC na parte inferior da página.

   c. A área de configuração geral desse tipo de alteração é aberta e a linha de assunto é preenchida. Para ver os detalhes do tipo de alteração, abra a área na parte superior da página.

5. Para garantir que certas pessoas recebam notificações sobre o progresso da RFC, preencha os endereços de e-mail. Para adicionar detalhes sobre o tipo de alteração, preencha a Descrição. Abra a área Configuração adicional para adicionar mais detalhes sobre o RFC.

6. Em Programação, selecione Executar esta alteração o mais rápido possível ou Agendar esta alteração. Se você selecionar Executar esta alteração o mais rápido possível, sua RFC será executada assim que as aprovações forem aprovadas. Se você selecionar Agendar este tipo de alteração, uma escolha de calendário, hora e fuso horário será exibida e sua RFC começará, após o envio, conforme programado.

7. Na área Configuração de execução, configure os parâmetros do tipo de alteração. Para ver os parâmetros opcionais, abra a área Configuração adicional.

8. Quando estiver pronto, escolha Executar.

Criando a infraestrutura

Faça login no Console AWS para a conta AMS de destino e, em seguida, no Console AMS para a conta.

Os procedimentos a seguir descrevem a criação de um banco de dados do RDS, um balanceador de carga e um grupo de Auto Scaling de forma que você use o IDs recurso para criar a infraestrutura.

Crie uma pilha RDS

Consulte RDS stack | Create.

Crie uma pilha ELB

Lance um ELB público.

DADOS NECESSÁRIOS:

- `VpcId`: a VPC que você está usando deve ser a mesma que a VPC usada anteriormente.
- `ELBSubnetIds`: uma matriz de sub-redes pelas quais o balanceador de carga distribuirá o tráfego. Escolha sub-redes públicas ou privadas. Encontre Subnet IDs com a referência da API For the AMS SKMS, consulte a guia Relatórios na operação do AWS Artifact Console. (CLI list-subnet-summaries:) ou no console AMS -> página de detalhes da VPC. VPCs
- `VpcId`: a VPC que você está usando deve ser a mesma que a VPC usada anteriormente.

1. Na página Criar RFC, selecione a categoria Implantação, a subcategoria Componentes avançados da pilha, o item pilha do balanceador de carga (ELB) e clique em Criar. Escolha Avançado e aceite todos os padrões (incluindo aqueles sem valor), exceto os mostrados a seguir.

```
Subject:                        WP-ELB-RFC
ELBSubnetIds:                   PUBLIC_AZ1
                              PUBLIC_AZ2
ELBScheme                       true
ELBCookieExpirationPeriod       600
VpcId:                          VPC_ID
Name:                           WP-Public-ELB
```

2. Clique em Enviar quando terminar.

Crie uma pilha de grupos do Auto Scaling

Inicie um grupo de Auto Scaling.

DADOS NECESSÁRIOS:

- `VpcId`: a VPC que você está usando deve ser a mesma que a VPC usada anteriormente.
- `AMI-ID`: esse valor determina que tipo de EC2 instâncias seu grupo de Auto Scaling (ASG) criará. Certifique-se de selecionar uma AMI em sua conta que comece com “cliente-” e seja do sistema operacional que você deseja. Encontre a AMI IDs com a referência da API For the AMS SKMS, consulte a guia Relatórios no AWS Artifact Console. operação (CLI: list-amis) ou no console do AMS -> página de detalhes. VPCs VPCs Este passo a passo é para ASGs configurar o uso de uma AMI Linux.
- `ASGLoadBalancerNames`: O balanceador de carga que você criou anteriormente - encontre o nome examinando o EC2 Console -> Balanceadores de carga (no painel de navegação à esquerda). Observe que esse não é o “Nome” que você especificou quando criou o ELB anteriormente.

1. Na página Criar RFC, selecione a categoria Implantação, a subcategoria Advanced Stack Components, o item Grupo de escalonamento automático e clique em Criar. Escolha Avançado e aceite todos os padrões (incluindo aqueles sem valor), exceto os mostrados a seguir.

   > **Note**
   >
   > Especifique a AMI AMI mais recente. Especifique o ELB criado anteriormente.

```
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. Clique em Enviar quando terminar.

Crie uma pilha S3

Inicie um bucket S3. O bucket do S3 é onde você carrega o pacote de aplicativos que você criou.

DADOS NECESSÁRIOS:

- `VPC-ID`: esse valor determina onde seu S3 Bucket estará. Ele deve ser o mesmo da VPC usada anteriormente.
- `AccessControl`: As opções da AccessControl lista predefinida (ACL) são `Private` e. `PublicRead` Para obter mais informações, consulte Amazon Simple Storage Service Canned ACL.
- `BucketName`: esse valor define o nome do S3 Bucket, você o usa para carregar seu pacote de aplicativos. Ele deve ser exclusivo em toda a região da conta e não pode incluir letras maiúsculas. Incluir o ID da sua conta como parte do não BucketName é obrigatório, mas facilita a identificação posterior do bucket. Para ver quais nomes de bucket do S3 existem na conta, acesse o console do Amazon S3 da sua conta.

1. Na página Criar RFC, selecione a categoria Implantação, a subcategoria Advanced Stack Components, o item Armazenamento S3 e clique em Criar.

   Você pode deixar a opção de parâmetro padrão em Básico para aceitar os padrões conforme descrito. Para definir valores diferentes, escolha Avançado.

> **Note**
>
> O bucket implantado com esse tipo de alteração permite read/write acesso total a toda a conta. Novos tipos de alteração podem ser necessários para permitir permissões de acesso mais restritas.

```
Subject:            S3-Bucket-RFC
BucketName:         ACCOUNT_ID-codedeploy-bundles
AccessControl:      Private

VpcId:              VPC_ID
Name:               S3BucketForWP
```

2. Clique em Enviar quando terminar.

Criar um WordPress CodeDeploy pacote

A seção fornece um exemplo de criação de um pacote de implantação de aplicativos.

1. Baixe WordPress, extraia os arquivos e crie um. diretório /scripts.

   Comando Linux:

   ```
   wget https://github.com/WordPress/WordPress/archive/master.zip
   ```

   Windows: cole `https://github.com/WordPress/WordPress/archive/master.zip` em uma janela do navegador e baixe o arquivo zip.

   Crie um diretório temporário no qual montar o pacote.

   Linux

   ```
   mkdir /tmp/WordPress
   ```

   Windows: Crie um diretório WordPress "", você usará o caminho do diretório posteriormente.

2. Extraia a WordPress fonte para o diretório WordPress "" e crie um. diretório /scripts.

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: vá para o diretório WordPress "" que você criou e crie um diretório de “scripts” lá.

Se você estiver em um ambiente Windows, certifique-se de definir o tipo de interrupção dos arquivos de script como Unix (LF). No Notepad ++, essa é uma opção na parte inferior direita da janela.

3. Crie o arquivo CodeDeploy appspec.yml, no WordPress diretório (se estiver copiando o exemplo, verifique o recuo, cada espaço conta). IMPORTANTE: Certifique-se de que o caminho de “origem” esteja correto para copiar os WordPress arquivos (nesse caso, em seu WordPress diretório) para o destino esperado (/var/www/html/WordPress). No exemplo, o arquivo appspec.yml está no diretório com os WordPress arquivos, portanto, somente “/” é necessário. Além disso, mesmo que você tenha usado uma AMI RHEL para seu grupo de Auto Scaling, deixe a linha “os: linux” como está. Exemplo de arquivo appspec.yml:

```
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. Crie scripts de arquivo bash no WordPress . diretório /scripts.

   Primeiro, crie `config_wordpress.sh` com o conteúdo a seguir (se preferir, você pode editar o arquivo wp-config.php diretamente).

   > **Note**
   >
   > *DBName*Substitua pelo valor fornecido no HA Stack RFC (por exemplo,`wordpress`).
   > *DB_MasterUsername*Substitua pelo `MasterUsername` valor fornecido no HA Stack RFC (por exemplo,`admin`).
   > *DB_MasterUserPassword*Substitua pelo `MasterUserPassword` valor fornecido no HA Stack RFC (por exemplo,`p4ssw0rd`).
   > *DB_ENDPOINT*Substitua pelo nome DNS do endpoint nas saídas de execução do HA Stack RFC (por exemplo,). `srt1cz23n45sfg.clgvd67uvydk.us-east-1.rds.amazonaws.com` Você pode encontrar isso na GetRfcoperação (CLI: get-rfc --rfc-id RFC_ID) ou na página de detalhes do RFC do console AMS para o HA Stack RFC que você enviou anteriormente.

```
#!/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. No mesmo diretório, crie `install_dependencies.sh` com o seguinte conteúdo:

```
#!/bin/bash
yum install -y php
yum install -y php-mysql
yum install -y mysql
```

```
service httpd restart
```

> **Note**
>
> O HTTPS é instalado como parte dos dados do usuário no lançamento para permitir que as verificações de saúde funcionem desde o início.

6. No mesmo diretório, crie `start_server.sh` com o seguinte conteúdo:

   - Para instâncias do Amazon Linux, use isso:

   ```
   #!/bin/bash
   service httpd start
   ```

   - Para instâncias do RHEL, use isso (os comandos extras são políticas que permitem que o SELINUX aceite): 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. No mesmo diretório, crie `stop_server.sh` com o seguinte conteúdo:

   ```
   #!/bin/bash
   service httpd stop
   ```

8. Crie o pacote zip.

   Linux

   ```
   $ cd /tmp/WordPress
   $ zip -r wordpress.zip .
   ```

   Windows: Vá para o diretório WordPress "", selecione todos os arquivos e crie um arquivo zip, não se esqueça de chamá-lo de wordpress.zip.

Implante o pacote de WordPress aplicativos com CodeDeploy

CodeDeploy É um serviço de implantação da AWS que automatiza implantações de aplicativos em instâncias da Amazon EC2 . Essa parte do processo envolve a criação de um CodeDeploy aplicativo, a criação de um grupo de CodeDeploy implantação e, em seguida, a implantação do aplicativo usando CodeDeploy.

Criar um CodeDeploy aplicativo

O CodeDeploy aplicativo é simplesmente um nome ou contêiner usado pela AWS CodeDeploy para garantir que a revisão, a configuração de implantação e o grupo de implantação corretos sejam referenciados durante uma implantação. A configuração de implantação, nesse caso, é o WordPress pacote que você criou anteriormente.

DADOS NECESSÁRIOS:

- `VpcId`: a VPC que você está usando, deve ser a mesma que a VPC usada anteriormente.
- `CodeDeployApplicationName`: deve ser exclusivo na conta. Consulte o CodeDeploy console para verificar os nomes dos aplicativos existentes.

1. Crie o CodeDeploy aplicativo para WordPress

   Na página Criar RFC, selecione a categoria Implantação, subcategoria Aplicativos, item CodeDeploy aplicativo e operação Criar na lista de opções RFC CT. Escolha Básico e defina os valores conforme mostrado. Clique em Enviar quando terminar.

   ```
   Subject:                    CD-WP-App-RFC
   CodeDeployApplicationName:  WordPress
   VpcId:                      VPC_ID
   Name:                       WP-CD-App
   ```

2. Clique em Enviar quando terminar.

Criar um grupo CodeDeploy de implantação

Crie o grupo CodeDeploy de implantação.

Um grupo de CodeDeploy implantação define um conjunto de instâncias individuais destinadas a uma implantação.

DADOS NECESSÁRIOS:

- `VpcId`: a VPC que você está usando deve ser a mesma que a VPC usada anteriormente.
- `CodeDeployApplicationName`: use o valor que você criou anteriormente.
- `CodeDeployAutoScalingGroups`: use o nome do grupo Auto Scaling que você criou anteriormente.
- `CodeDeployDeploymentGroupName`: um nome para o grupo de implantação. Esse nome deve ser exclusivo para cada aplicativo associado ao grupo de implantação.
- `CodeDeployServiceRoleArn`: Use a fórmula fornecida no exemplo.

1. Na página Criar RFC, selecione a categoria Implantação, a subcategoria Aplicativos, o grupo de CodeDeploy implantação do item e a operação Criar na lista de opções RFC CT. Escolha Avançado e defina os valores conforme mostrado (somente um Assunto é necessário para o RFC). Clique em Enviar quando terminar.

   > **Note**
   >
   > Faça referência ao ARN da função de CodeDeploy serviço nesse formato `"arn:aws:iam::085398962942:role/aws-codedeploy-role"` e use o nome do grupo de Auto Scaling criado anteriormente para “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. Clique em Enviar quando terminar.

Faça o upload do WordPress aplicativo

Você tem acesso automático a qualquer instância de bucket do S3 que você criar. Você pode acessá-lo por meio de seu Bastions (consulte Acesso a instâncias) ou por meio do console S3 e fazer o upload do CodeDeploy pacote. O pacote precisa estar pronto para continuar implantando a pilha. O exemplo usa o nome do bucket criado anteriormente.

Você pode usar esse comando da AWS para compactar o pacote:

```
aws s3 cp wordpress/wordpress.zip s3://ACCOUNT_ID-codedeploy-bundles/
```

Implante o WordPress aplicativo com CodeDeploy

Implante o CodeDeploy aplicativo.

DADOS NECESSÁRIOS:

- `VPC-ID`: a VPC que você está usando deve ser a mesma que a VPC usada anteriormente.
- `CodeDeployApplicationName`: use o nome do CodeDeploy aplicativo que você criou anteriormente.
- `CodeDeployDeploymentGroupName`: use o nome do grupo de CodeDeploy implantação que você criou anteriormente.
- `S3Location`(onde você fez o upload do pacote de aplicativos)`S3Bucket`:: O BucketName que você criou anteriormente `S3BundleType` e`S3Key`: O tipo e o nome do pacote que você colocou na sua loja do S3.

1. Implantar o pacote WordPress CodeDeploy de aplicativos

   Na página Criar RFC, selecione a categoria Implantação, subcategoria Aplicativos, item CodeDeploy aplicativo e operação Implantar na lista de opções RFC CT. Escolha Básico e defina os valores conforme mostrado. Clique em Enviar quando terminar.

   > **Note**
   >
   > Faça referência ao CodeDeploy aplicativo, ao grupo de CodeDeploy implantação, ao bucket S3 e ao pacote criados anteriormente.

```
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. Clique em Enviar quando terminar.

Validar a implantação do aplicativo

Navegue até o endpoint (ELB CName) do balanceador de carga criado anteriormente, com o caminho implantado:/. WordPress WordPress Por exemplo:

```
http://stack-ID-FOR-ELB.us-east-1.elb.amazonaws.com/WordPress
```

Elimine a implantação de aplicativos

Para reduzir a implantação, você envia o Delete Stack CT contra a pilha de banco de dados do RDS, o balanceador de carga do aplicativo, o grupo Auto Scaling, o bucket S3 e o aplicativo e grupo Code Deploy — seis no total. RFCs Além disso, você pode enviar uma solicitação de serviço para que os instantâneos do RDS sejam excluídos (eles são excluídos automaticamente após dez dias, mas custam uma pequena quantia enquanto estão lá). Reúna a pilha IDs para todos e siga estas etapas. Consulte Stack | Delete.

Tutorial de CLI: pilha de duas camadas de alta disponibilidade (Linux/RHEL)

Esta seção descreve como implantar uma pilha de duas camadas de alta disponibilidade (HA) em um ambiente AMS usando a CLI do AMS.

> **Note**
>
> Esse passo a passo de implantação foi testado em ambientes AMZN Linux e RHEL.

Resumo das tarefas e tarefas necessárias RFCs:

1. Crie infraestrutura (pilha HA de dois níveis)
2. Crie um bucket S3 para aplicativos CodeDeploy
3. Crie o pacote de WordPress aplicativos e faça o upload para o bucket do S3
4. Implante o aplicativo com CodeDeploy
5. Acesse o WordPress site e faça login para validar a implantação

Antes de começar

O Deployment | Advanced Stack Components | High Availability Two Tier Stack Advanced | Create CT cria um grupo de Auto Scaling, um balanceador de carga, um banco de dados e CodeDeploy um nome de aplicativo e grupo de implantação (com o mesmo nome que você dá ao aplicativo). Para obter informações sobre, CodeDeploy consulte O que é CodeDeploy?

Este passo a passo usa uma RFC de pilha de duas camadas de alta disponibilidade (avançada) que inclui UserData e também descreve como criar um WordPress pacote que pode ser implantado. CodeDeploy

O exemplo `UserData` mostrado no exemplo obtém metadados da instância, como ID da instância, região etc., de dentro de uma instância em execução, consultando o serviço de metadados da EC2 instância disponível em http://169.254.169.254/latest/meta-data/. Essa linha no script de dados do usuário:`REGION=$(curl 169.254.169.254/latest/meta-data/placement/availability-zone/ | sed 's/[a-z]$//')`, recupera o nome da zona de disponibilidade do serviço de metadados na variável $REGION de nossas regiões suportadas e a usa para preencher a URL do bucket do S3 em que o agente é baixado. CodeDeploy O IP 169.254.169.254 é roteável somente dentro da VPC (todos podem consultar o serviço). VPCs Para obter informações sobre o serviço, consulte Metadados da instância e dados do usuário. Observe também que os scripts inseridos como UserData são executados como usuário “root” e não precisam usar o comando “sudo”.

Essa explicação passo a passo deixa os seguintes parâmetros no valor padrão (mostrado):

- Grupo de Auto Scaling:. `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`
- Balanceador de carga:. `HealthCheckInterval=30, HealthCheckTimeout=5`
- Banco de dados:`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`.
- Aplicação:`DeploymentConfigName=CodeDeployDefault.OneAtATime`.
- Balde S3:`AccessControl=Private`.

CONFIGURAÇÕES ADICIONAIS:

`RequestedStartTime`e `RequestedEndTime` se você quiser agendar sua RFC: você pode usar Time.is para determinar a hora UTC correta. Os exemplos fornecidos devem ser ajustados adequadamente. Uma RFC não pode continuar se a hora de início tiver passado. Como alternativa, você pode deixar esses valores desativados para criar um ASAP RFC que seja executado assim que as aprovações forem aprovadas.

> **Note**
>
> Há muitos parâmetros que você pode escolher definir de forma diferente dos mostrados. Os valores desses parâmetros mostrados no exemplo foram testados, mas podem não ser adequados para você.

Crie a infraestrutura

Coletar os dados a seguir antes de começar fará com que a implantação seja mais rápida.

OS DADOS NECESSÁRIOS TÊM UMA PILHA:

- AutoScalingGroup:
  - `UserData`: esse valor é fornecido neste tutorial. Ele inclui comandos para configurar o recurso CodeDeploy e iniciar o CodeDeploy agente.
  - `AMI-ID`: esse valor determina que tipo de EC2 instâncias seu grupo de Auto Scaling (ASG) criará. Certifique-se de selecionar uma AMI em sua conta que comece com “cliente-” e seja do sistema operacional que você deseja. Encontre a AMI IDs com a referência da API For the AMS SKMS, consulte a guia Relatórios no AWS Artifact Console. operação (CLI: list-amis) ou no console do AMS -> página de detalhes. VPCs VPCs Este passo a passo é para ASGs configurar o uso de uma AMI Linux.
- Banco de dados:
  - Esses parâmetros,`DBEngine`,`EngineVersion`, e `LicenseModel` devem ser definidos de acordo com sua situação, embora os valores mostrados no exemplo tenham sido testados.
  - Esses parâmetros,`RDSSubnetIds`,, `DBNameMasterUsername`, e `MasterUserPassword` são necessários ao implantar o pacote de aplicativos. Para RDSSubnet IDs, use duas sub-redes privadas.
- LoadBalancer:
  - Esses parâmetros,`DBEngine`,`EngineVersion`, e `LicenseModel` devem ser definidos de acordo com sua situação, embora os valores mostrados no exemplo tenham sido testados.
  - `ELBSubnetIds`: Use duas sub-redes públicas.
- Aplicativo: o `ApplicationName` valor define o nome do CodeDeploy aplicativo e o nome do grupo de CodeDeploy implantação. Você o usa para implantar seu aplicativo. Ele deve ser exclusivo na conta. Para verificar os CodeDeploy nomes da sua conta, consulte o CodeDeploy console. O exemplo usa "WordPress", mas, se você usar esse valor, certifique-se de que ele ainda não esteja em uso.

Esse procedimento utiliza a CT de duas camadas (avançada) de alta disponibilidade (ct-06mjngx5flwto) e a CT de armazenamento Create S3 (ct-1a68ck03fn98r). Na sua conta autenticada, siga estas etapas na linha de comando.

1. Inicie a pilha de infraestrutura.

a. Envie os parâmetros de execução do esquema JSON para a pilha HA de duas camadas CT em um arquivo em sua pasta atual chamado .json. CreateStackParams

```
aws amscm get-change-type-version --change-type-id "ct-06mjngx5flwto"
 --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
 CreateStackParams.json
```

b. Modifique o esquema. Substitua o *variables* conforme apropriado. Por exemplo, use o sistema operacional que você deseja para as EC2 instâncias que o ASG criará. Grave o `ApplicationName` como você o usará posteriormente para implantar o aplicativo. Observe que você pode adicionar até 50 tags.

```
{
"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. Envie o modelo CreateRfc JSON para um arquivo na sua pasta atual chamado CreateStackRfc .json:

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

d. Modifique o modelo RFC da seguinte forma e salve-o, você pode excluir e substituir o conteúdo. Observe que `RequestedStartTime` agora `RequestedEndTime` são opcionais; excluí-los cria um ASAP RFC que é executado assim que aprovado (o que geralmente acontece automaticamente). Para enviar um RFC agendado, adicione esses valores.

```
{
"ChangeTypeVersion":    "3.0",
"ChangeTypeId":         "ct-06mjngx5flwto",
"Title":                "HA-Stack-For-WP-RFC"
}
```

e. Crie o RFC, especificando o CreateStackRfc arquivo.json e o arquivo de parâmetros de execução CreateStackParams .json:

```
aws amscm create-rfc --cli-input-json file://CreateStackRfc.json --execution-
parameters file://CreateStackParams.json
```

Você recebe o ID do RFC na resposta. Salve o ID para as etapas subsequentes.

f. Envie o RFC:

```
aws amscm submit-rfc --rfc-id  RFC_ID
```

Se o RFC for bem-sucedido, você não receberá nenhuma saída.

g. Para verificar o status do RFC, execute

```
aws amscm get-rfc --rfc-id RFC_ID
```

Anote o ID do RFC.

2. Inicie um bucket S3

Coletar os dados a seguir antes de começar fará com que a implantação seja mais rápida.

BUCKET S3 DE DADOS NECESSÁRIO:

- `VPC-ID`: esse valor determina onde seu S3 Bucket estará. Use a mesma VPC ID que você usou anteriormente.
- `BucketName`: esse valor define o nome do S3 Bucket, você o usa para carregar seu pacote de aplicativos. Ele deve ser exclusivo em toda a região da conta e não pode incluir letras maiúsculas. Incluir o ID da sua conta como parte do não BucketName é obrigatório, mas facilita a identificação posterior do bucket. Para ver quais nomes de bucket do S3 existem na conta, acesse o console do Amazon S3 da sua conta.

a. Envie os parâmetros de execução do esquema JSON para o armazenamento S3 create CT em um arquivo JSON chamado createS3 .json. StoreParams

```
aws amscm get-change-type-version --change-type-id "ct-1a68ck03fn98r"
 --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
 CreateS3StoreParams.json
```

b. Modifique o esquema da seguinte forma, você pode excluir e substituir o conteúdo. Substitua *VPC_ID* adequadamente. Os valores no exemplo foram testados, mas podem não ser adequados para você.

> **Tip**
>
> O `BucketName` deve ser exclusivo em toda a região da conta e não pode incluir letras maiúsculas. Incluir o ID da sua conta como parte do não BucketName é

obrigatório, mas facilita a identificação posterior do bucket. Para ver quais nomes de bucket do S3 existem na conta, acesse o console do Amazon S3 da sua conta.

```
{
"Description":      "S3BucketForWordPressBundle",
"VpcId":            "VPC_ID",
"StackTemplateId":  "stm-s2b72beb000000000",
"Name":             "S3BucketForWP",
"TimeoutInMinutes":  60,
"Parameters":   {
    "AccessControl":    "Private",
    "BucketName":       "ACCOUNT_ID-BUCKET_NAME"
    }
}
```

c. Envie o modelo JSON CreateRfc para um arquivo, na sua pasta atual, chamado StoreRfc createS3 .json:

```
aws amscm create-rfc --generate-cli-skeleton > CreateS3StoreRfc.json
```

d. Modifique e salve o arquivo CreateS3 StoreRfc .json, você pode excluir e substituir o conteúdo. Observe que `RequestedStartTime` agora `RequestedEndTime` são opcionais; excluí-los cria um ASAP RFC que é executado assim que aprovado (o que geralmente acontece automaticamente). Para enviar um RFC agendado, adicione esses valores.

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-1a68ck03fn98r",
"Title":                "S3-Stack-For-WP-RFC"
}
```

e. Crie o RFC, especificando o arquivo CreateS3 .json e o arquivo de parâmetros de execução CreateS3 StoreRfc .json: StoreParams

```
aws amscm create-rfc --cli-input-json file://CreateS3StoreRfc.json  --
execution-parameters file://CreateS3StoreParams.json
```

Você recebe o RfcId do novo RFC na resposta. Salve o ID para as etapas subsequentes.

f. Envie o RFC:

```
aws amscm submit-rfc --rfc-id RFC_ID
```

Se o RFC for bem-sucedido, você não receberá nenhuma saída.

g. Para verificar o status do RFC, execute

```
aws amscm get-rfc --rfc-id RFC_ID
```

Crie, carregue e implante o aplicativo

Primeiro, crie um pacote de WordPress aplicativos e, em seguida, use o CodeDeploy CTs para criar e implantar o aplicativo.

1. Baixe WordPress, extraia os arquivos e crie um. diretório /scripts.

   Comando Linux:

   ```
   wget https://github.com/WordPress/WordPress/archive/master.zip
   ```

   Windows: cole `https://github.com/WordPress/WordPress/archive/master.zip` em uma janela do navegador e baixe o arquivo zip.

   Crie um diretório temporário no qual montar o pacote.

   Linux

   ```
   mkdir /tmp/WordPress
   ```

   Windows: Crie um diretório WordPress "", você usará o caminho do diretório posteriormente.

2. Extraia a WordPress fonte para o diretório WordPress "" e crie um. diretório /scripts.

   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: vá para o diretório WordPress "" que você criou e crie um diretório de “scripts” lá.

Se você estiver em um ambiente Windows, certifique-se de definir o tipo de interrupção dos arquivos de script como Unix (LF). No Notepad ++, essa é uma opção na parte inferior direita da janela.

3. Crie o arquivo CodeDeploy appspec.yml, no WordPress diretório (se estiver copiando o exemplo, verifique o recuo, cada espaço conta). IMPORTANTE: Certifique-se de que o caminho de “origem” esteja correto para copiar os WordPress arquivos (nesse caso, em seu WordPress diretório) para o destino esperado (/var/www/html/WordPress). No exemplo, o arquivo appspec.yml está no diretório com os WordPress arquivos, portanto, somente “/” é necessário. Além disso, mesmo que você tenha usado uma AMI RHEL para seu grupo de Auto Scaling, deixe a linha “os: linux” como está. Exemplo de arquivo appspec.yml:

```
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. Crie scripts de arquivo bash no WordPress . diretório /scripts.

Primeiro, crie `config_wordpress.sh` com o conteúdo a seguir (se preferir, você pode editar o arquivo wp-config.php diretamente).

> **Note**
>
> *DBName* Substitua pelo valor fornecido no HA Stack RFC (por exemplo, `wordpress`).
> *DB_MasterUsername* Substitua pelo `MasterUsername` valor fornecido no HA Stack RFC (por exemplo, `admin`).
> *DB_MasterUserPassword* Substitua pelo `MasterUserPassword` valor fornecido no HA Stack RFC (por exemplo, `p4ssw0rd`).
> *DB_ENDPOINT* Substitua pelo nome DNS do endpoint nas saídas de execução do HA Stack RFC (por exemplo,). `srt1cz23n45sfg.clgvd67uvydk.us-east-1.rds.amazonaws.com` Você pode encontrar isso na GetRfcoperação (CLI: get-rfc --rfc-id RFC_ID) ou na página de detalhes do RFC do console AMS para o HA Stack RFC que você enviou anteriormente.

```
#!/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. No mesmo diretório, crie `install_dependencies.sh` com o seguinte conteúdo:

```
#!/bin/bash
yum install -y php
yum install -y php-mysql
yum install -y mysql
service httpd restart
```

> **Note**
>
> O HTTPS é instalado como parte dos dados do usuário no lançamento para permitir que as verificações de saúde funcionem desde o início.

6. No mesmo diretório, crie `start_server.sh` com o seguinte conteúdo:

   - Para instâncias do Amazon Linux, use isso:

   ```
   #!/bin/bash
   service httpd start
   ```

   - Para instâncias do RHEL, use isso (os comandos extras são políticas que permitem que o SELINUX aceite): 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. No mesmo diretório, crie `stop_server.sh` com o seguinte conteúdo:

   ```
   #!/bin/bash
   service httpd stop
   ```

8. Crie o pacote zip.

   Linux

   ```
   $ cd /tmp/WordPress
   $ zip -r wordpress.zip .
   ```

   Windows: Vá para o diretório WordPress "", selecione todos os arquivos e crie um arquivo zip, não se esqueça de chamá-lo de wordpress.zip.

1. Faça o upload do pacote de aplicativos no bucket do S3.

O pacote precisa estar pronto para continuar implantando a pilha.

Você tem acesso automático a qualquer instância de bucket do S3 que você criar. Você pode acessá-lo por meio de seus bastiões ou do console S3 e fazer o upload do WordPress pacote drag-and-drop ou navegar até o arquivo zip e selecioná-lo.

Você também pode usar o seguinte comando em uma janela de shell; verifique se você tem o caminho correto para o arquivo zip:

```
aws s3 cp wordpress.zip s3://BUCKET_NAME/
```

2. Implante o pacote de WordPress aplicativos.

   Coletar os dados a seguir antes de começar fará com que a implantação seja mais rápida.

   DADOS NECESSÁRIOS:

   - `VPC-ID`: esse valor determina onde seu S3 Bucket estará. Use a mesma VPC ID que você usou anteriormente.
   - `CodeDeployApplicationNameeCodeDeployApplicationName`: O `ApplicationName` valor que você usou no HA 2-Tier Stack RFC definiu o CodeDeployApplicationName e o. CodeDeployDeploymentGroupName O exemplo usa "WordPress", mas você pode ter usado um valor diferente.
   - `S3Location`: Para`S3Bucket`, use o `BucketName` que você criou anteriormente. Os `S3BundleType` e `S3Key` são do pacote que você colocou na sua loja S3.

   a. Exiba os parâmetros de execução do esquema JSON para o CodeDeploy aplicativo deploy CT em um arquivo JSON chamado Deploy Params.json. CDApp

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

   b. Modifique o esquema da seguinte forma e salve-o como, você pode excluir e substituir o conteúdo.

      ```
      {
      "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. Envie o modelo JSON CreateRfc para um arquivo, na sua pasta atual, chamado Deploy CDApp RFC.json:

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

d. Modifique e salve o arquivo Deploy CDApp RFC.json, você pode excluir e substituir o conteúdo. Observe que `RequestedStartTime` agora `RequestedEndTime` são opcionais; excluí-los cria um ASAP RFC que é executado assim que aprovado (o que geralmente acontece automaticamente). Para enviar um RFC agendado, adicione esses valores.

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-2edc3sd1sqmrb",
"Title":                "CD-Deploy-For-WP-RFC"
}
```

e. Crie o RFC, especificando o arquivo Deploy CDApp Rfc e o arquivo de parâmetros de execução do Deploy CDApp Params:

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

Você recebe o RfcId do novo RFC na resposta. Salve o ID para as etapas subsequentes.

f. Envie o RFC:

```
aws amscm submit-rfc --rfc-id RFC_ID
```

Se o RFC for bem-sucedido, você não receberá nenhuma saída.

g. Para verificar o status do RFC, execute

```
aws amscm get-rfc --rfc-id RFC_ID
```

Validar a implantação do aplicativo

Navegue até o endpoint (ELB CName) do balanceador de carga criado anteriormente, com o caminho implantado:/. WordPress WordPress Por exemplo:

```
http://stack-ID-FOR-ELB.us-east-1.elb.amazonaws.com/WordPress
```

Elimine a implantação de aplicativos

Depois de concluir o tutorial, você deverá desmontar a implantação para não ser cobrado pelos recursos.

A seguir está uma operação genérica de exclusão de pilha. Você vai querer enviá-lo duas vezes, uma para a pilha HA de 2 níveis e outra para a pilha de buckets S3. Como acompanhamento final, envie uma solicitação de serviço para que todos os snapshots do bucket do S3 (incluindo o ID da pilha do bucket do S3 na solicitação de serviço) sejam excluídos. Eles são excluídos automaticamente após 10 dias, mas excluí-los antecipadamente economiza um pouco de custo.

Este passo a passo fornece um exemplo do uso do console AMS para excluir uma pilha do S3; esse procedimento se aplica à exclusão de qualquer pilha usando o console do AMS.

> **Note**
>
> Se você excluir um bucket do S3, primeiro ele deverá ser esvaziado de objetos.

DADOS NECESSÁRIOS:

- `StackId`: A pilha a ser usada. Você pode encontrar isso acessando a página AMS Console Stacks, disponível por meio de um link no painel de navegação esquerdo. Usando a API/CLI do

AMS SKMS, execute a referência da API Para o AMS SKMS, consulte a guia Relatórios no AWS Artifact Console. operação (na CLI). `list-stack-summaries`

- O ID do tipo de alteração para este passo a passo `éct-0q0bic0ywqk6c`: a versão é “1.0". Para descobrir a versão mais recente, execute este comando:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=ct-0q0bic0ywqk6c
```

CRIAÇÃO EM LINHA:

- Execute o comando create RFC com os parâmetros de execução fornecidos em linha (aspas de escape ao fornecer parâmetros de execução em linha). E

```
aws amscm create-rfc --change-type-id "ct-0q0bic0ywqk6c" --change-type-version "1.0"
 --title "Delete My Stack" --execution-parameters "{\"StackId\":\"STACK_ID\"}"
```

- Envie a RFC usando a ID da RFC retornada na operação de criação da RFC. Até ser enviada, a RFC permanece no `Editing` estado e não é aplicada.

```
aws amscm submit-rfc --rfc-id RFC_ID
```

- Monitore o status da RFC e visualize a saída da execução:

```
aws amscm get-rfc --rfc-id RFC_ID
```

CRIAÇÃO DE MODELO:

1. Envie o modelo RFC para um arquivo em sua pasta atual; o exemplo o chama de DeleteStackRfc .json:

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

2. Modifique e salve o DeleteStackRfc arquivo.json. Como a exclusão de uma pilha tem apenas um parâmetro de execução, os parâmetros de execução podem estar no próprio DeleteStackRfc arquivo.json (não há necessidade de criar um arquivo JSON separado com parâmetros de execução).

As aspas internas na extensão ExecutionParameters JSON devem ser excluídas com uma barra invertida (\). Exemplo sem horário de início e término:

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-0q0bic0ywqk6c",
"Title":                "Delete-My-Stack-RFC"
"ExecutionParameters":  "{
        \"StackId\":\"STACK_ID\"}"
}
```

3. Crie o RFC:

```
aws amscm create-rfc --cli-input-json file://DeleteStackRfc.json
```

Você recebe o RfcId do novo RFC na resposta. Por exemplo:

```
{
"RfcId": "daaa1867-ffc5-1473-192a-842f6b326102"
}
```

Salve o ID para as etapas subsequentes.

4. Envie o RFC:

```
aws amscm submit-rfc --rfc-id RFC_ID
```

Se o RFC for bem-sucedido, você não receberá nenhuma confirmação na linha de comando.

5. Para monitorar o status da solicitação e visualizar a Saída de Execução:

```
aws amscm get-rfc --rfc-id RFC_ID --query "Rfc.
{Status:Status.Name,Exec:ExecutionOutput}" --output table
```

Tutorial de CLI: Implantação de um site Tier and Tie WordPress

Esta seção descreve como implantar um WordPress site de alta disponibilidade (HA) em um ambiente AMS usando a CLI do AMS. Esse conjunto de instruções inclui um exemplo de criação do arquivo de pacote WordPress CodeDeploy compatível necessário (por exemplo, zip).

> **Note**
>
> Este passo a passo de implantação foi projetado para uso com um ambiente Linux AMZN. Os parâmetros essenciais da variável são anotados como*replaceable*; no entanto, talvez você queira modificar outros parâmetros para se adequar à sua situação.

Resumo das tarefas e tarefas necessárias RFCs:

1. Crie a infraestrutura:
   a. Crie uma pilha RDS (CLI)
   b. Criar um balanceador de carga
   c. Crie um grupo de Auto Scaling e vincule-o ao balanceador de carga
   d. Crie um bucket S3 para aplicativos CodeDeploy
2. Crie um pacote de WordPress aplicativos (não requer um RFC)
3. Implante o pacote de WordPress aplicativos com CodeDeploy:
   a. Crie um CodeDeploy aplicativo
   b. Crie um grupo CodeDeploy de implantação
   c. Carregue seu pacote de WordPress aplicativos no bucket do S3 (não requer um RFC)
   d. Implemente o CodeDeploy aplicativo
4. Valide a implantação
5. Destrua a implantação

Siga todas as etapas na linha de comando da sua conta autenticada.

Criando um RFC usando a CLI

Para obter informações detalhadas sobre a criação RFCs, consulte Criação RFCs; para obter uma explicação dos parâmetros comuns de RFC, consulte Parâmetros comuns de RFC.

Crie a infraestrutura

Os procedimentos a seguir descrevem a criação de um banco de dados do RDS, um balanceador de carga e um grupo de Auto Scaling de forma que você use o IDs recurso para criar a infraestrutura.

Crie uma pilha RDS (CLI)

Consulte RDS stack | Create.

Crie uma pilha ELB

Inicie um balanceador de carga público (ELB). Consulte Load Balancer (ELB) Stack | Create.

Crie uma pilha de grupos do Auto Scaling

Inicie um grupo de Auto Scaling.

Consulte Grupo de Auto Scaling | Criar.

Crie uma loja S3

Inicie um bucket S3. O bucket do S3 é onde você carrega o pacote de aplicativos que você criou. Consulte Armazenamento S3 | Criar.

Crie um pacote WordPress de aplicativos para CodeDeploy

Esta seção fornece um exemplo de criação de um pacote de implantação de aplicativos.

1. Baixe WordPress, extraia os arquivos e crie um. diretório /scripts.

   Comando Linux:

   ```
   wget https://github.com/WordPress/WordPress/archive/master.zip
   ```

   Windows: cole `https://github.com/WordPress/WordPress/archive/master.zip` em uma janela do navegador e baixe o arquivo zip.

   Crie um diretório temporário no qual montar o pacote.

   Linux

   ```
   mkdir /tmp/WordPress
   ```

   Windows: Crie um diretório WordPress "", você usará o caminho do diretório posteriormente.
2. Extraia a WordPress fonte para o diretório WordPress "" e crie um. diretório /scripts.

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: vá para o diretório WordPress "" que você criou e crie um diretório de “scripts” lá.

Se você estiver em um ambiente Windows, certifique-se de definir o tipo de interrupção dos arquivos de script como Unix (LF). No Notepad ++, essa é uma opção na parte inferior direita da janela.

3. Crie o arquivo CodeDeploy appspec.yml no WordPress diretório (se estiver copiando o exemplo, verifique o recuo, cada espaço conta). IMPORTANTE: Certifique-se de que o caminho de “origem” esteja correto para copiar os WordPress arquivos (nesse caso, em seu WordPress diretório) para o destino esperado (/var/www/html/WordPress). No exemplo, o arquivo appspec.yml está no diretório com os WordPress arquivos, portanto, somente “/” é necessário. Além disso, mesmo que você tenha usado uma AMI RHEL para seu grupo de Auto Scaling, deixe a linha “os: linux” como está. Exemplo de arquivo appspec.yml:

```
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. Crie scripts de arquivo bash no WordPress . diretório /scripts.

   Primeiro, crie `config_wordpress.sh` com o conteúdo a seguir (se preferir, você pode editar o arquivo wp-config.php diretamente).

   > **Note**
   >
   > *DBName*Substitua pelo valor fornecido no HA Stack RFC (por exemplo,`wordpress`).
   > *DB_MasterUsername*Substitua pelo `MasterUsername` valor fornecido no HA Stack RFC (por exemplo,`admin`).
   > *DB_MasterUserPassword*Substitua pelo `MasterUserPassword` valor fornecido no HA Stack RFC (por exemplo,`p4ssw0rd`).
   > *DB_ENDPOINT*Substitua pelo nome DNS do endpoint nas saídas de execução do HA Stack RFC (por exemplo,). `srt1cz23n45sfg.clgvd67uvydk.us-east-1.rds.amazonaws.com` Você pode encontrar isso na GetRfcoperação (CLI: get-rfc --rfc-id RFC_ID) ou na página de detalhes do RFC do console AMS para o HA Stack RFC que você enviou anteriormente.

```
#!/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. No mesmo diretório, crie `install_dependencies.sh` com o seguinte conteúdo:

```
#!/bin/bash
yum install -y php
yum install -y php-mysql
yum install -y mysql
```

```
service httpd restart
```

> **Note**
>
> O HTTPS é instalado como parte dos dados do usuário no lançamento para permitir que as verificações de saúde funcionem desde o início.

6. No mesmo diretório, crie `start_server.sh` com o seguinte conteúdo:

   - Para instâncias do Amazon Linux, use isso:

   ```
   #!/bin/bash
   service httpd start
   ```

   - Para instâncias do RHEL, use isso (os comandos extras são políticas que permitem que o SELINUX aceite): 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. No mesmo diretório, crie `stop_server.sh` com o seguinte conteúdo:

   ```
   #!/bin/bash
   service httpd stop
   ```

8. Crie o pacote zip.

   Linux

   ```
   $ cd /tmp/WordPress
   $ zip -r wordpress.zip .
   ```

   Windows: Vá para o diretório WordPress "", selecione todos os arquivos e crie um arquivo zip, não se esqueça de chamá-lo de wordpress.zip.

Implante o pacote de WordPress aplicativos com CodeDeploy

CodeDeploy É um serviço de implantação da AWS que automatiza implantações de aplicativos em instâncias da Amazon EC2 . Essa parte do processo envolve a criação de um CodeDeploy aplicativo, a criação de um grupo de CodeDeploy implantação e, em seguida, a implantação do aplicativo usando CodeDeploy.

Criar um CodeDeploy aplicativo

O CodeDeploy aplicativo é simplesmente um nome ou contêiner usado pela AWS CodeDeploy para garantir que a revisão, a configuração de implantação e o grupo de implantação corretos sejam referenciados durante uma implantação. A configuração de implantação, nesse caso, é o WordPress pacote que você criou anteriormente.

DADOS NECESSÁRIOS:

- `VpcId`: a VPC que você está usando, deve ser a mesma que a VPC usada anteriormente.
- `CodeDeployApplicationName`: deve ser exclusivo na conta. Consulte o CodeDeploy console para verificar os nomes dos aplicativos existentes.
- `ChangeTypeIdeChangeTypeVersion`: O ID do tipo de alteração para este passo a passo `éct-0ah3gwb9seqk2`, para descobrir a versão mais recente, execute este comando:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=ct-0ah3gwb9seqk2
```

1. Exiba os parâmetros de execução do esquema JSON do CodeDeploy aplicativo CT para um arquivo na sua pasta atual; o exemplo o chama de Create CDApp Params.json.

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

2. Modifique e salve o arquivo JSON da seguinte forma; você pode excluir e substituir o conteúdo.

```
{
"Description":                    "Create WordPress CodeDeploy App",
"VpcId":                          "VPC_ID",
"StackTemplateId":                "stm-sft6rv00000000000",
"Name":                           "WordPressCDApp",
"TimeoutInMinutes":               60,
```

```
"Parameters":   {
    "CodeDeployApplicationName":    "WordPressCDApp"
    }
}
```

3. Envie o modelo JSON CreateRfc para um arquivo em sua pasta atual; o exemplo o chama de Create CDApp RFC.json.

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

4. Modifique e salve o arquivo JSON da seguinte forma; você pode excluir e substituir o conteúdo. Observe que `RequestedStartTime` agora `RequestedEndTime` são opcionais; excluí-los faz com que o RFC seja executado assim que for aprovado (o que geralmente acontece automaticamente). Para enviar um RFC “agendado”, adicione esses valores.

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

5. Crie o RFC, especificando o arquivo Create CDApp Rfc e o arquivo de parâmetros de execução:

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

   Você recebe o ID de RFC do novo RFC na resposta. Salve o ID para as etapas subsequentes.

6. Envie o RFC:

```
aws amscm submit-rfc --rfc-id RFC_ID
```

   Se o RFC for bem-sucedido, você não receberá nenhuma saída.

7. Envie o RFC:

```
aws amscm get-rfc --rfc-id RFC_ID
```

Criar um grupo CodeDeploy de implantação

Crie o grupo CodeDeploy de implantação.

Um grupo de CodeDeploy implantação define um conjunto de instâncias individuais destinadas a uma implantação.

DADOS NECESSÁRIOS:

- `VpcId`: a VPC que você está usando, deve ser a mesma que a VPC usada anteriormente.
- `CodeDeployApplicationName`: use o valor que você criou anteriormente.
- `CodeDeployAutoScalingGroups`: use o nome do grupo Auto Scaling que você criou anteriormente.
- `CodeDeployDeploymentGroupName`: um nome para o grupo de implantação. Esse nome deve ser exclusivo para cada aplicativo associado ao grupo de implantação.
- `CodeDeployServiceRoleArn`: Use a fórmula fornecida no exemplo.
- `ChangeTypeIdeChangeTypeVersion`: O ID do tipo de alteração para este passo a passo `éct-2gd0u847qd9d2`, para descobrir a versão mais recente, execute este comando:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=ct-2gd0u847qd9d2
```

1. Envie os parâmetros de execução do esquema JSON para um arquivo em sua pasta atual; o exemplo o chama de Create CDDep GroupParams .json.

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

2. Modifique e salve o arquivo JSON da seguinte forma; você pode excluir e substituir o conteúdo.

```
{
"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. Envie o modelo JSON CreateRfc para um arquivo em sua pasta atual; o exemplo o chama de Create CDDep GroupRfc .json.

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

4. Modifique e salve o arquivo JSON da seguinte forma; você pode excluir e substituir o conteúdo. Observe que `RequestedStartTime` agora `RequestedEndTime` são opcionais; excluí-los faz com que o RFC seja executado assim que for aprovado (o que geralmente acontece automaticamente). Para enviar um RFC “agendado”, adicione esses valores.

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

5. Crie o RFC, especificando o CDDep GroupRfc arquivo Create e o arquivo de parâmetros de execução:

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

   Você recebe o ID de RFC do novo RFC na resposta. Salve o ID para as etapas subsequentes.

6. Envie o RFC:

```
aws amscm submit-rfc --rfc-id RFC_ID
```

   Se o RFC for bem-sucedido, você não receberá nenhuma saída.

7. Verifique o status do RFC:

```
aws amscm get-rfc --rfc-id RFC_ID
```

Faça o upload do WordPress aplicativo

Você tem acesso automático a qualquer instância de bucket do S3 que você criar. Você pode acessá-lo por meio de seu Bastions (consulte Acesso a instâncias) ou por meio do console S3 e fazer o upload do CodeDeploy pacote. O pacote precisa estar pronto para continuar implantando a pilha. O exemplo usa o nome do bucket criado anteriormente.

```
aws s3 cp wordpress/wordpress.zip s3://ACCOUNT_ID-codedeploy-bundles/
```

Implante o WordPress aplicativo com CodeDeploy

Implante o CodeDeploy aplicativo.

Depois de ter seu pacote de CodeDeploy aplicativos e grupo de implantação, use esse RFC para implantar o aplicativo.

DADOS NECESSÁRIOS:

- `VPC-ID`: a VPC que você está usando deve ser a mesma que a VPC usada anteriormente.
- `CodeDeployApplicationName`: use o nome do CodeDeploy aplicativo que você criou anteriormente.
- `CodeDeployDeploymentGroupName`: use o nome do grupo de CodeDeploy implantação que você criou anteriormente.
- `S3Location`(onde você fez o upload do pacote de aplicativos)`S3Bucket`:: O BucketName que você criou anteriormente `S3BundleType` e`S3Key`: O tipo e o nome do pacote que você colocou na sua loja do S3.
- `ChangeTypeIdeChangeTypeVersion`: O ID do tipo de alteração para este passo a passo `éct-2edc3sd1sqmrb`, para descobrir a versão mais recente, execute este comando:

  ```
  aws amscm list-change-type-version-summaries --filter
   Attribute=ChangeTypeId,Value=ct-2edc3sd1sqmrb
  ```

1. Envie o esquema JSON dos parâmetros de execução para a CT de implantação do CodeDeploy aplicativo em um arquivo na sua pasta atual; o exemplo o chama de Deploy CDApp Params.json.

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

2. Modifique o arquivo JSON da seguinte forma; você pode excluir e substituir o conteúdo. Para`S3Bucket`, use o `BucketName` que você criou anteriormente.

```
{
"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. Envie o modelo JSON CreateRfc para um arquivo em sua pasta atual; o exemplo o chama de Deploy CDApp RFC.json:

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

4. Modifique e salve o arquivo Deploy CDApp RFC.json; você pode excluir e substituir o conteúdo.

```
{
"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. Crie o RFC, especificando o arquivo de parâmetros de execução e o arquivo Deploy CDApp Rfc:

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

Você recebe o RfcId do novo RFC na resposta. Salve o ID para as etapas subsequentes.

6. Envie o RFC:

```
aws amscm submit-rfc --rfc-id RFC_ID
```

Se o RFC for bem-sucedido, você não receberá nenhuma saída.

Validar a implantação do aplicativo

Navegue até o endpoint (ELB CName) do balanceador de carga criado anteriormente, com o caminho WordPress implantado:/. WordPress Por exemplo:

```
http://stack-ID-FOR-ELB.us-east-1.elb.amazonaws.com/WordPress
```

Acabe com a implantação de aplicativos

Para reduzir a implantação, você envia o Delete Stack CT contra a pilha de banco de dados do RDS, o balanceador de carga do aplicativo, o grupo Auto Scaling, o bucket S3 e o aplicativo e grupo Code Deploy — seis no total. RFCs Além disso, você pode enviar uma solicitação de serviço para que os instantâneos do RDS sejam excluídos (eles são excluídos automaticamente após dez dias, mas custam uma pequena quantia enquanto estão lá). Reúna a pilha IDs para todos e siga estas etapas.

Este passo a passo fornece um exemplo do uso do console AMS para excluir uma pilha do S3; esse procedimento se aplica à exclusão de qualquer pilha usando o console do AMS.

> **Note**
>
> Se você excluir um bucket do S3, primeiro ele deverá ser esvaziado de objetos.

DADOS NECESSÁRIOS:

- `StackId`: A pilha a ser usada. Você pode encontrar isso acessando a página AMS Console Stacks, disponível por meio de um link no painel de navegação esquerdo. Usando a API/CLI do

AMS SKMS, execute a referência da API Para o AMS SKMS, consulte a guia Relatórios no AWS Artifact Console. operação (na CLI). `list-stack-summaries`

- O ID do tipo de alteração para este passo a passo `éct-0q0bic0ywqk6c`: a versão é “1.0". Para descobrir a versão mais recente, execute este comando:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=ct-0q0bic0ywqk6c
```

CRIAÇÃO EM LINHA:

- Execute o comando create RFC com os parâmetros de execução fornecidos em linha (aspas de escape ao fornecer parâmetros de execução em linha). E

```
aws amscm create-rfc --change-type-id "ct-0q0bic0ywqk6c" --change-type-version "1.0"
 --title "Delete My Stack" --execution-parameters "{\"StackId\":\"STACK_ID\"}"
```

- Envie a RFC usando a ID da RFC retornada na operação de criação da RFC. Até ser enviada, a RFC permanece no `Editing` estado e não é aplicada.

```
aws amscm submit-rfc --rfc-id RFC_ID
```

- Monitore o status da RFC e visualize a saída da execução:

```
aws amscm get-rfc --rfc-id RFC_ID
```

CRIAÇÃO DE MODELO:

1. Envie o modelo RFC para um arquivo em sua pasta atual; o exemplo o chama de DeleteStackRfc .json:

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

2. Modifique e salve o DeleteStackRfc arquivo.json. Como a exclusão de uma pilha tem apenas um parâmetro de execução, os parâmetros de execução podem estar no próprio DeleteStackRfc arquivo.json (não há necessidade de criar um arquivo JSON separado com parâmetros de execução).

As aspas internas na extensão ExecutionParameters JSON devem ser excluídas com uma barra invertida (\). Exemplo sem horário de início e término:

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-0q0bic0ywqk6c",
"Title":                "Delete-My-Stack-RFC"
"ExecutionParameters":  "{
        \"StackId\":\"STACK_ID\"}"
}
```

3. Crie o RFC:

```
aws amscm create-rfc --cli-input-json file://DeleteStackRfc.json
```

Você recebe o RfcId do novo RFC na resposta. Por exemplo:

```
{
"RfcId": "daaa1867-ffc5-1473-192a-842f6b326102"
}
```

Salve o ID para as etapas subsequentes.

4. Envie o RFC:

```
aws amscm submit-rfc --rfc-id RFC_ID
```

Se o RFC for bem-sucedido, você não receberá nenhuma confirmação na linha de comando.

5. Para monitorar o status da solicitação e visualizar a Saída de Execução:

```
aws amscm get-rfc --rfc-id RFC_ID --query "Rfc.
{Status:Status.Name,Exec:ExecutionOutput}" --output table
```

Manutenção de aplicativos

Depois que a infraestrutura é implantada, atualizá-la de forma consistente em todos os seus ambientes de AMS, do controle de qualidade à preparação e à produção, é o desafio.

Esta seção fornece uma visão geral do processo de ingestão de carga de trabalho do AMS e alguns exemplos de métodos diferentes que você pode usar para manter sua camada de infraestrutura de nuvem atualizada.

Estratégias de manutenção de aplicativos

A forma como você implanta seus aplicativos afeta a forma como você os mantém. Esta seção fornece algumas estratégias para manutenção de aplicativos.

As atualizações do ambiente podem envolver qualquer uma dessas mudanças:

- Atualizações de segurança
- Novas versões de seus aplicativos
- Alterações na configuração do aplicativo
- Atualizações nas dependências

> **Note**
>
> Para qualquer implantação de aplicativo, não importa o método, sempre registre uma solicitação de serviço com antecedência para que o AMS saiba que você vai implantar um aplicativo.

Exemplos de instalação de aplicativos imutáveis versus mutáveis

Mutabilidade da instância de computação	Método de instalação do aplicativo	AMI
Mutable	Com CodeDeploy	Fornecido pela AMS
	Manualmente	

Mutabilidade da instância de computação	Método de instalação do aplicativo	AMI
	Com um chef ou fantoche, à base de puxar	
	Com Ansible ou Salt, baseado em push	
Imutável	Com uma AMI dourada	Personalizado (com base no fornecido pela AMS)

Implantação mutável com uma AMI CodeDeploy habilitada

CodeDeployA AWS é um serviço que automatiza implantações de código em qualquer instância, incluindo instâncias da Amazon e EC2 instâncias executadas localmente. Você pode usar CodeDeploy com o AMS para criar e implantar um CodeDeploy aplicativo. Observe que o AMS fornece um perfil de instância padrão para CodeDeploy aplicativos.

- Amazon Linux (versão 1)
- Amazon Linux 2
- RedHat 7
- CentOS 7

Antes de usar CodeDeploy pela primeira vez, você deve concluir várias etapas de configuração:

1. Instale ou atualize a AWS CLI
2. Crie uma função de serviço para a AWS CodeDeploy, você usa o ARN da função de serviço na implantação

IDs todas as opções de tomografia computadorizada podem ser encontradas na Referência de tipo de alteração.

> **Note**
>
> Atualmente, você deve usar o armazenamento Amazon S3 com essa solução.

As etapas básicas estão descritas aqui e o procedimento está detalhado no Guia do usuário do AMS.

1. Crie um bucket de armazenamento Amazon S3. CT: ct-1a68ck03fn98r. O bucket do S3 deve ter o versionamento ativado (para obter informações sobre como fazer isso, consulte Habilitando o versionamento do bucket).
2. Coloque seus CodeDeploy artefatos agrupados nele. Você pode fazer isso com o console do Amazon S3 sem solicitar acesso por meio do AMS. Ou usando uma variação desse comando:

   ```
   aws s3 cp ZIP_FILEPATH_AND_NAME s3://S3BUCKET_NAME/
   ```

3. Encontre uma `customer-` AMI do AMS; use uma das seguintes opções:
   - Console AMS: a página de detalhes da VPC para a VPC relevante
   - API AMS Para a referência da API AMS SKMS, consulte a guia Relatórios no AWS Artifact Console. ou CLI: `aws amsskms list-amis`
4. Crie um grupo de escalonamento automático (ASG). CT: ct-2tylseo8rxfsc. Especifique a AMI do AMS, defina o balanceador de carga para ter portas abertas e especifique `customer-mc-ec2-instance-profile` para o. `ASGIAMInstanceProfile`
5. Crie seu CodeDeploy aplicativo. CT: ct-0ah3gwb9seqk2. Os parâmetros incluem um nome de aplicativo, por exemplo`WordpressProd`.
6. Crie seu grupo CodeDeploy de implantação. CT: ct-2gd0u847qd9d2. Os parâmetros incluem o nome do CodeDeploy aplicativo, o nome do ASG, o nome do tipo de configuração e o ARN da função de serviço.
7. Implante o CodeDeploy aplicativo. CT: ct-2edc3sd1sqmrb. Os parâmetros incluem o nome do CodeDeploy aplicativo, o nome do tipo de configuração, o nome do grupo de implantação, o tipo de revisão e a localização do bucket do S3 onde estão os CodeDeploy artefatos.

Implantação mutável, instâncias de aplicativos configuradas e atualizadas manualmente

Essa estratégia de implantação de aplicativos é uma atualização simples e manual das instâncias do aplicativo. Essas são as etapas básicas.

IDs todas as opções de tomografia computadorizada podem ser encontradas na Referência de tipo de alteração.

> **Note**
>
> Atualmente, você deve usar o armazenamento Amazon S3 com essa solução.

As etapas básicas estão descritas aqui; os vários procedimentos estão detalhados no Guia do Usuário do AMS.

1. Crie um bucket de armazenamento Amazon S3. CT: ct-1a68ck03fn98r. O bucket do S3 deve ter o versionamento ativado (para obter informações sobre como fazer isso, consulte Habilitando o versionamento do bucket).
2. Coloque os artefatos do aplicativo agrupados nele (tudo o que seu aplicativo precisa para iniciar na inicialização e funcionar). Você pode fazer isso com o console do Amazon S3 sem solicitar acesso por meio do AMS. Ou usando uma variação desse comando:

   ```
   aws s3 cp ZIP_FILEPATH_AND_NAME s3://S3BUCKET_NAME/
   ```

3. Encontre uma AMI AMI, tudo estará CodeDeploy nele. Para encontrar uma AMI “cliente”, use:

   - Console AMS: a página de detalhes da VPC para a VPC relevante
   - API AMS Para a referência da API AMS SKMS, consulte a guia Relatórios no AWS Artifact Console. ou CLI: `aws amsskms list-amis`

4. Crie uma EC2 instância com essa AMI. CT: ct-14027q0sjyt1h. Especifique a AMI do AMS, defina uma tag `Key=backup, Value=true` e especifique a `customer-mc-ec2-instance-profile` para o `InstanceProfile` parâmetro. Observe o ID da instância que é retornado.
5. Solicite acesso de administrador à instância. CT: ct-1dmlg9g1l91h6. Você precisará do FQDN para sua conta. Se você não tiver certeza do que é seu FQDN, você pode encontrá-lo em:

- Usando a guia Nome do diretório do AWS Management Console para serviços de diretório (em Segurança e identidade).
- Executando um desses comandos (classes de diretório de retorno; DC+DC+DC=FQDN): Windows: ou Linux:. `whoami /fqdn` `hostname --fqdn`

6. Faça login na instância, consulte Como acessar instâncias via Bastions no Guia do usuário do AMS.
7. Faça o download dos arquivos de aplicativos agrupados do bucket do S3 para a instância.
8. Solicite um backup imediato com uma solicitação de serviço para o AMS, você precisará saber o ID da instância.
9. Quando precisar atualizar seu aplicativo, carregue novos arquivos no bucket do S3 e siga as etapas de 3 a 8.

Implantação mutável com uma AMI configurada por ferramenta de implantação baseada em pull

Essa estratégia se baseia no `InstanceUserData` parâmetro do Managed Services Create EC2 CT. Para obter mais informações sobre como usar esse parâmetro, consulte Como configurar instâncias com dados do usuário. Este exemplo pressupõe uma ferramenta de implantação de aplicativos baseada em pull, como Chef ou Puppet.

O CodeDeploy agente é suportado em todos os AMS AMIs. Aqui está a lista dos compatíveis AMIs:

- Amazon Linux (versão 1)
- Amazon Linux 2
- RedHat 7
- CentOS 7

IDs todas as opções de CT podem ser encontradas na Referência de Tipos de Alteração.

> **Note**
>
> Atualmente, você deve usar o armazenamento Amazon S3 com essa solução.

As etapas básicas estão descritas aqui e o procedimento está detalhado no Guia do usuário do AMS.

1. Crie um bucket de armazenamento Amazon S3. CT: ct-1a68ck03fn98r. O bucket do S3 deve ter o versionamento ativado (para obter informações sobre como fazer isso, consulte Habilitando o versionamento do bucket).
2. Coloque seus CodeDeploy artefatos agrupados nele. Você pode fazer isso com o console do Amazon S3 sem solicitar acesso por meio do AMS. Ou usando uma variação desse comando:

   ```
   aws s3 cp ZIP_FILEPATH_AND_NAME s3://S3BUCKET_NAME/
   ```

3. Encontre uma `customer-` AMI do AMS; use uma das seguintes opções:
   - Console AMS: a página de detalhes da VPC para a VPC relevante
   - API AMS Para a referência da API AMS SKMS, consulte a guia Relatórios no AWS Artifact Console. ou CLI: `aws amsskms list-amis`
4. Crie uma EC2 instância. CT: ct-14027q0sjyt1h; defina uma tag `Key=backup, Value=true` e use o `InstanceUserData` parâmetro para especificar um bootstrap e outros scripts (Chef/Puppet agente de download etc.) e inclua as chaves de autorização necessárias. Você pode encontrar um exemplo de como fazer isso no Guia do Usuário do AMS, na seção Gerenciamento de Alterações, exemplos de como criar uma implantação de HA em dois níveis. Como alternativa, solicite acesso e faça login na instância e configure-a com os artefatos de implantação necessários. Lembre-se de que os comandos de implantação baseados em pull vão dos agentes em suas instâncias para o servidor principal corporativo e podem precisar de autorização para passar pelos bastiões. Você pode precisar de uma solicitação de serviço ao AMS para solicitar acesso ao group/AD grupo de segurança sem bastiões.
5. Repita a etapa 4 para criar outra EC2 instância e configurá-la com o servidor mestre da ferramenta de implantação.
6. Quando precisar atualizar seu aplicativo, use a ferramenta de implantação para implantar as atualizações em suas instâncias.

Implantação mutável com uma AMI configurada por ferramenta de implantação baseada em push

Essa estratégia se baseia no `InstanceUserData` parâmetro do Managed Services Create EC2 CT. Para obter mais informações sobre como usar esse parâmetro, consulte Como configurar instâncias

com dados do usuário. Este exemplo pressupõe uma ferramenta de implantação de aplicativos baseada em pull, como Chef ou Puppet.

IDs todas as opções de tomografia computadorizada podem ser encontradas na Referência de tipo de alteração.

> **Note**
>
> Atualmente, você deve usar o armazenamento Amazon S3 com essa solução.

As etapas básicas estão descritas aqui e o procedimento está detalhado no Guia do usuário do AMS.

1. Crie um bucket de armazenamento Amazon S3. CT: ct-1a68ck03fn98r. O bucket do S3 deve ter o versionamento ativado (para obter informações sobre como fazer isso, consulte Habilitando o versionamento do bucket).
2. Coloque seus CodeDeploy artefatos agrupados nele. Você pode fazer isso com o console Amazon S3 sem solicitar acesso por meio do AMS. Ou usando uma variação desse comando:

   ```
   aws s3 cp ZIP_FILEPATH_AND_NAME s3://S3BUCKET_NAME/
   ```

3. Encontre uma AMI AMI, tudo estará CodeDeploy nele. Para encontrar uma AMI “cliente”, use:

   - Console AMS: a página de detalhes da VPC para a VPC relevante
   - API AMS Para a referência da API AMS SKMS, consulte a guia Relatórios no AWS Artifact Console. ou CLI: `aws amsskms list-amis`

4. Crie uma EC2 instância. CT: ct-14027q0sjyt1h; defina uma tag e use o `InstanceUserData` parâmetro para executar um bootstrap e outros scripts`Key=backup, Value=true`, incluindo chaves de autorização, pilha SALT (inicialize um minion — para obter mais informações, consulte Bootstrapping Salt no Linux EC2 com Cloud-Init) ou Ansible (instale um par de chaves — para obter mais informações, consulte Introdução ao Ansible e ao Dynamic Amazon Inventory Management). EC2 Como alternativa, solicite acesso e faça login na instância e configure-a com os artefatos de implantação necessários. Lembre-se de que os comandos baseados em push vêm da sua sub-rede corporativa para suas instâncias e talvez você precise configurar a autorização para que eles passem pelos bastiões. Você pode precisar de uma solicitação de serviço ao AMS para solicitar acesso ao group/AD grupo de segurança sem bastiões.
5. Repita a etapa 4 para criar outra EC2 instância e configurá-la com o servidor mestre da ferramenta de implantação.

6. Quando precisar atualizar seu aplicativo, use a ferramenta de implantação para implantar as atualizações em suas instâncias.

Implantação imutável com uma AMI dourada

Essa estratégia emprega uma AMI "dourada" que você configurou para se comportar da maneira desejada em todas as instâncias do seu aplicativo. Por exemplo, as instâncias criadas com essa AMI dourada uniriam automaticamente o domínio e o DNS corretos, se autoconfigurariam, reinicializariam e iniciariam todos os sistemas necessários. Quando quiser atualizar as instâncias do seu aplicativo, você recria a AMI dourada e implementa instâncias de aplicativos totalmente novas com ela.

O CodeDeploy agente é suportado em todos os AMS AMIs. Aqui está a lista dos compatíveis AMIs:

- Amazon Linux (versão 1)
- Amazon Linux 2
- RedHat 7
- CentOS 7

IDs todas as opções de tomografia computadorizada podem ser encontradas na Referência de tipo de alteração.

> **Note**
>
> Atualmente, você deve usar o armazenamento Amazon S3 com essa solução.

1. Crie um bucket de armazenamento Amazon S3. CT: ct-1a68ck03fn98r. O bucket do S3 deve ter o versionamento ativado (para obter informações sobre como fazer isso, consulte Habilitando o versionamento do bucket).
2. Coloque os artefatos do aplicativo agrupados nele (tudo o que seu aplicativo precisa para iniciar na inicialização e funcionar). Você pode fazer isso com o console Amazon S3 sem solicitar acesso por meio do AMS. Ou usando uma variação desse comando:

   ```
   aws s3 cp ZIP_FILEPATH_AND_NAME s3://S3BUCKET_NAME/
   ```

3. Encontre uma `customer-` AMI do AMS; use uma das seguintes opções:

- Console AMS: a página de detalhes da VPC da VPC relevante
- API AMS Para a referência da API AMS SKMS, consulte a guia Relatórios no AWS Artifact Console. ou CLI: `aws amsskms list-amis`

4. Crie uma EC2 instância com essa AMI. CT: ct-14027q0sjyt1h. Especifique a AMI do AMS, defina uma tag `Key=backup, Value=true` e especifique `customer-mc-ec2-instance-profile` para `InstanceProfile` o. Observe o ID da instância que é retornado.
5. Solicite acesso de administrador à instância. CT: ct-1dmlg9g1l91h6. Você precisará do FQDN para sua conta. Se você não tiver certeza do que é seu FQDN, você pode encontrá-lo em:

   - Usando a guia Nome do diretório do AWS Management Console para serviços de diretório (em Segurança e identidade).
   - Executando um desses comandos (classes de diretório de retorno; DC+DC+DC=FQDN): Windows: ou Linux:. `whoami /fqdn` `hostname --fqdn`
6. Faça login na instância, consulte Como acessar instâncias no Guia do usuário do AMS.
7. Faça o download para a instância dos arquivos de aplicativos agrupados do bucket do S3. Configure a instância para que ela implante automaticamente o aplicativo totalmente funcional na inicialização.
8. Crie a AMI dourada na instância. CT: ct-3rqqu43krekby. Para obter detalhes, consulte AMI | Create.
9. Configure um grupo de Auto Scaling para criar novas instâncias usando essa AMI. CT: ct-2tylseo8rxfsc. Quando precisar atualizar seu aplicativo, siga este procedimento e solicite que o AMS atualize o ASG para usar a nova AMI dourada; use um Management | Other | Other | Update CT para isso.

Estratégias de atualização

Há algumas estratégias diferentes que você pode empregar para atualizar seus aplicativos ou instâncias em seu ambiente gerenciado pelo AMS.

- Tempo de inatividade programado: essa estratégia simples envolve programar o tempo para que seu aplicativo fique off-line e seja atualizado manualmente. Para fazer isso, envie uma solicitação Management | Other | Other | Update CT (ct-0xdawir96cy7k) para interromper as instâncias necessárias. Faça as atualizações necessárias e, em seguida, envie outra solicitação Management | Other | Other | Update CT (ct-0xdawir96cy7k) para iniciar as instâncias.

- Azul/verde: essa estratégia exige que você tenha um ambiente redundante (dois ambientes completamente funcionais) e coloque um ambiente off-line usando atualizações do sistema de nomes de domínio (DNS) ou do firewall da web (WAF) para redirecionar o tráfego. Atualize um ambiente e redirecione novamente para atualizar o outro ambiente.

  Para saber mais, consulte AWS CodeDeploy introduz Blue/Green implantações.

- Atualização contínua com a nova AMI: é aqui que você personaliza uma nova AMI (consulte Criar AMI) e depois solicita que o AMS a implante em seu grupo de Auto Scaling. Use um Gerenciamento | Outro | Outro | Atualize CT (ct-0xdawir96cy7k) para fazer isso.

Agendador de recursos do AWS Managed Services

Use o AWS Managed Services (AMS) Resource Scheduler para programar o início e a parada automáticos de AutoScaling grupos, EC2 instâncias da Amazon e instâncias do RDS em sua conta. Isso ajuda a reduzir os custos de infraestrutura quando os recursos não devem funcionar 24 horas por dia, 7 dias por semana. A solução foi criada com base no Instance Scheduler on AWS, mas contém recursos adicionais e personalizações específicas para as necessidades do AMS.

> **Note**
>
> Por padrão, o AMS Resource Scheduler não interage com recursos que não fazem parte de uma AWS CloudFormation pilha. O recurso deve fazer parte de uma pilha que começa com “stack-”, “sc-” ou “SC-”. Para agendar os recursos que não fazem parte de uma CloudFormation pilha, você pode atualizar o parâmetro da pilha do Resource Scheduler para. `ScheduleNonStackResources` `Yes`

O Agendador de Recursos do AMS usa períodos e programações:

- Os períodos definem os horários em que o Resource Scheduler é executado, como horário de início, horário de término e dias do mês.
- As programações contêm seus períodos definidos, juntamente com configurações adicionais, como janela de manutenção do SSM, fuso horário, configuração de hibernação e assim por diante; e especificam quando os recursos devem ser executados, de acordo com as regras de período configuradas.

Você pode configurar esses períodos e cronogramas usando os tipos de alteração automatizada () CTs do AMS Resource Scheduler.

Para obter detalhes completos sobre as configurações disponíveis para o AMS Resource Scheduler, consulte a documentação correspondente do AWS Instance Scheduler em Componentes da solução. Para uma visão arquitetônica da solução, consulte a documentação correspondente do AWS Instance Scheduler em Architecture overview.html.

Implantando o Agendador de Recursos do AMS

Para implantar o AMS Resource Scheduler, use o tipo de alteração automática (CT): Deployment | AMS Resource Scheduler | Solution | Deploy (ct-0ywnhc8e5k9z5) para gerar uma RFC que, em seguida, implanta a solução em sua conta. Depois que a RFC é executada, uma CloudFormation pilha contendo recursos do AMS Resource Scheduler com configuração padrão é automaticamente provisionada em sua conta. Para obter mais informações sobre os tipos de alteração do Agendador de Recursos, consulte Agendador de Recursos do AMS.

> **Note**
>
> Para descobrir se o AMS Resource Scheduler já está implantado em sua conta, verifique o console AWS Lambda dessa conta e procure a função Scheduler. AMSResource

Depois que o Agendador de Recursos do AMS for provisionado em sua conta, recomendamos que você revise a configuração padrão e, se necessário, personalize configurações como chave de tag, fuso horário, serviços agendados e assim por diante, com base em suas preferências. Para obter detalhes sobre as personalizações recomendadasPersonalizando o Agendador de Recursos do AMS, consulte a seguir.

Para fazer as configurações personalizadas ou apenas confirmar a configuração do Resource Scheduler,

Personalizando o Agendador de Recursos do AMS

Recomendamos que você personalize as seguintes propriedades do Agendador de Recursos do AMS usando a atualização dos tipos de alteração do Agendador de Recursos do AMS, consulte Agendador de Recursos do AMS.

- Nome da tag: o nome da tag que o Resource Scheduler usará para associar agendamentos de instâncias a recursos. O valor padrão é Programação.

- Serviços agendados: uma lista de serviços separados por vírgulas que o Resource Scheduler pode gerenciar. O valor padrão é “ec2, rds, autoscaling”. Os valores válidos são “ec2", “rds” e “autoscaling”
- Fuso horário padrão: especifique o fuso horário padrão para o Agendador de Recursos usar. O valor padrão é UTC.
- Use a CMK: uma lista separada por vírgulas da Chave Gerenciada pelo Cliente (CMK) do Amazon KMS para a ARNs qual o Resource Scheduler pode receber permissões.
- Uso LicenseManager: Uma lista separada por vírgulas do Gerenciador de AWS Licenças ARNs para esse Agendador de Recursos pode receber permissões.

> **Note**
>
> O AMS pode, de tempos em tempos, lançar recursos e correções para manter o AMS Resource Scheduler atualizado em sua conta. Quando isso acontece, qualquer personalização feita no Agendador de Recursos do AMS é preservada.

Usando o Agendador de Recursos do AMS

Para configurar o Agendador de Recursos do AMS após a implantação da solução, use o Agendador de Recursos automatizado CTs para criar, excluir, atualizar e descrever (obter detalhes sobre) os períodos do Agendador de Recursos do AMS (os horários em que o Agendador de Recursos é executado) e programações (os períodos configurados e outras opções). Para obter um exemplo do uso dos tipos de alteração do Agendador de Recursos do AMS, consulte Agendador de Recursos do AMS.

Para selecionar recursos a serem gerenciados pelo AMS Resource Scheduler, após a implantação e a criação do cronograma, você usa o AMS Tag Create CTs para marcar grupos de Auto Scaling, pilhas do Amazon RDS e recursos EC2 da Amazon com a chave de tag que você forneceu durante a implantação, e o cronograma definido como o valor da tag. Depois que os recursos são marcados, os recursos são programados para início ou término de acordo com sua programação definida do Agendador de Recursos.

Não há custo adicional para usar o AMS Resource Scheduler. No entanto, a solução usa vários Serviços da AWS e você é cobrado por esses recursos à medida que eles são usados. Para obter mais detalhes, consulte Visão geral da arquitetura.

Para optar por não participar do AMS Resource Scheduler:

- Para cancelamento ou desativação temporária: envie um RFC usando o Gerenciamento automatizado | Agendador de Recursos do AMS | Estado | Desativar o tipo de alteração (ct-14v49adibs4db)
- Para remoção permanente: Envie um RFC de gerenciamento | Outro | Outro | Atualização (revisão necessária) (ct-0xdawir96cy7k) solicitando a remoção do sistema de automação de lançamentos do Resource Scheduler

Estimador de custos do AMS Resource Scheduler

Para monitorar a economia de custos, o AMS Resource Scheduler apresenta um componente que calcula de hora em hora a economia estimada de custos dos recursos da Amazon EC2 e do RDS que são gerenciados pelo programador. Esses dados de economia de custos são então publicados como uma CloudWatch métrica (`AMS/ResourceScheduler`) para ajudá-lo a rastreá-los. O estimador de economia de custos estima apenas as economias nas horas de operação da instância. Ele não contabiliza nenhum outro custo, como os custos de transferência de dados associados a um recurso.

O estimador de economia de custos é ativado com o Resource Scheduler. Ele é executado de hora em hora e recupera dados de custo e uso de AWS Cost Explorer. A partir desses dados, ele calcula o custo médio por hora para cada tipo de instância e, em seguida, projeta o custo por um dia inteiro se estivesse sendo executado sem ser programado. A economia de custos é a diferença entre o custo projetado e o custo real reportado pelo Cost Explorer para um determinado dia.

Por exemplo, se a instância A estiver configurada com o Resource Scheduler para ser executada das 9h às 17h, são oito horas em um determinado dia. O Cost Explorer relata o custo como $1 e o uso como 8. O custo médio por hora é, portanto, de $0,125. Se a instância não fosse agendada com o Resource Scheduler, a instância seria executada 24 horas nesse dia. Nesse caso, o custo teria sido 24x0,125 = $3. O Resource Scheduler ajudou você a obter uma economia de custos de $2.

Para que o estimador de economia de custos recupere o custo e o uso somente dos recursos gerenciados pelo Resource Scheduler a partir do Cost Explorer, a chave de tag que o Resource Scheduler usa para direcionar recursos precisa ser ativada como a tag de alocação de custos no Painel de cobrança. Se a conta pertencer a uma organização, a chave da tag precisará ser ativada na conta de gerenciamento da organização. Para obter informações sobre como fazer isso, consulte Ativando tags de alocação de custos definidas pelo usuário e tags de alocação de custos definidas pelo usuário

Depois que a chave de tag é ativada como etiqueta de alocação de custos, o AWS faturamento começa a rastrear o custo e o uso dos recursos gerenciados pelo Resource Scheduler e, depois que os dados são disponibilizados, o estimador de economia de custos começa a calcular a economia e publicar os dados no namespace métrico em. `AMS/ResourceScheduler` CloudWatch

Dicas para estimadores de custos

O Cost Savings Estimator não aceita descontos como instâncias reservadas, planos de economia etc. em consideração em seu cálculo. O Estimator pega os custos de uso do Cost Explorer e calcula o custo médio por hora dos recursos. Para obter mais detalhes, consulte Entendendo seus conjuntos de dados de AWS custo: uma folha de dicas

Para que o estimador de economia de custos recupere o custo e o uso somente dos recursos gerenciados pelo Resource Scheduler a partir do Cost Explorer, a chave de tag que o Resource Scheduler usa para direcionar recursos precisa ser ativada como a tag de alocação de custos no painel de faturamento. Se a conta pertencer a uma organização, a chave da tag precisará ser ativada na conta de gerenciamento da organização. Para obter informações sobre como fazer isso, consulte Tags de alocação de custos definidas pelo usuário. Se a etiqueta de alocação de custos não for ativada, o estimador não poderá calcular a economia e publicar a métrica, mesmo que ela esteja ativada.

Melhores práticas do AMS Resource Scheduler

Programação de instâncias da Amazon EC2

- O comportamento de desligamento da instância deve ser definido como `stop` e não como`terminate`. Isso é predefinido `stop` para instâncias criadas com o tipo de alteração automática AMS Amazon EC2 Create (ct-14027q0sjyt1h) e pode ser definido para instâncias da EC2 Amazon criadas com ingestão, definindo a propriedade como. AWS CloudFormation `InstanceInitiatedShutdownBehavior` `stop` Se as instâncias tiverem o comportamento de encerramento definido como`terminate`, elas terminarão quando o Agendador de Recursos as interromper e o agendador não conseguirá reiniciá-las.
- EC2 As instâncias da Amazon que fazem parte de um grupo de Auto Scaling não são processadas individualmente pelo AMS Resource Scheduler, mesmo que estejam marcadas.
- Se o volume raiz da instância de destino for criptografado com uma chave mestra de cliente (CMK) do KMS, uma `kms:CreateGrant` permissão adicional precisará ser adicionada à sua função do IAM do Resource Scheduler para que o agendador possa iniciar essas instâncias. Essa permissão não é adicionada à função por padrão para melhorar a segurança. Se você

precisar dessa permissão, envie uma RFC com o tipo de alteração Management | AMS Resource Scheduler | Solution | Update e especifique uma lista separada por vírgulas ARNs do KMS. CMKs

Agendamento de grupos de Auto Scaling

- O AMS Resource Scheduler inicia ou interrompe o escalonamento automático de grupos de Auto Scaling, não de instâncias individuais no grupo. Ou seja, o programador restaura o tamanho do grupo Auto Scaling (início) ou define o tamanho como 0 (parada).
- AutoScaling Grupo de tags com a tag especificada e não com as instâncias dentro do grupo.
- Durante a parada, o AMS Resource Scheduler armazena os valores de capacidade mínima, desejada e máxima do grupo Auto Scaling e define a capacidade mínima e desejada como 0. Durante o início, o programador restaura o tamanho do grupo do Auto Scaling como estava durante a parada. Portanto, as instâncias do grupo Auto Scaling devem usar uma configuração de capacidade apropriada para que o encerramento e a reinicialização das instâncias não afetem nenhum aplicativo executado no grupo do Auto Scaling.
- Se o grupo do Auto Scaling for modificado (a capacidade mínima ou máxima) durante um período de execução, o programador armazenará o novo tamanho do grupo do Auto Scaling e o usará ao restaurar o grupo no final de um cronograma de parada.

Programação de instâncias do Amazon RDS

- O agendador pode tirar um snapshot antes de interromper as instâncias do RDS (não se aplica ao cluster de banco de dados Aurora). Esse recurso é ativado por padrão com o parâmetro de CloudFormation modelo Create RDS Instance Snapshot definido como verdadeiro. O snapshot é mantido até a próxima vez em que a instância do Amazon RDS for interrompida e um novo snapshot for criado.

  O Scheduler pode usar instâncias do start/stop Amazon RDS que fazem parte de um cluster ou banco de dados Aurora do Amazon RDS ou em uma configuração de várias zonas de disponibilidade (Multi-AZ). No entanto, verifique a limitação do Amazon RDS quando o programador não conseguir interromper a instância do Amazon RDS, especialmente as instâncias Multi-AZ. Para programar o Aurora Cluster para iniciar ou parar, use o parâmetro do modelo Schedule Aurora Clusters (o padrão é true). O cluster Aurora (não as instâncias individuais dentro do cluster) deve ser marcado com a chave de tag definida durante a configuração inicial e o nome da programação como o valor da tag para programar esse cluster.

Cada instância do Amazon RDS tem uma janela de manutenção semanal durante a qual todas as alterações do sistema são aplicadas. Durante a janela de manutenção, o Amazon RDS iniciará automaticamente instâncias que foram interrompidas por mais de sete dias para aplicar a manutenção. Observe que o Amazon RDS não interromperá a instância após a conclusão do evento de manutenção.

O programador permite especificar se deseja adicionar a janela de manutenção preferencial de uma instância do Amazon RDS como um período de execução à sua programação. A solução iniciará a instância no início da janela de manutenção e interromperá a instância no final da janela de manutenção se nenhum outro período de execução especificar que a instância deve ser executada e se o evento de manutenção for concluído.

Se o evento de manutenção não for concluído até o final da janela de manutenção, a instância será executada até o intervalo de agendamento após a conclusão do evento de manutenção.

> **Note**
>
> O Agendador não valida se um recurso foi iniciado ou interrompido. Ele faz a chamada da API e segue em frente. Se a chamada da API falhar, ela registrará o erro para investigação.

Considerações sobre segurança de aplicativos

A segurança do aplicativo inclui considerar quais permissões o aplicativo precisará executar, quais regras de firewall e quais funções do IAM devem ser habilitadas para acessar o aplicativo.

Para entender melhor a AWS segurança geral, consulte Melhores práticas de segurança, identidade e conformidade.

Acesso para gerenciamento de configurações

O AWS Managed Services (AMS) busca fornecer a você uma infraestrutura livre de dores de cabeça para que você não precise se preocupar com problemas de segurança, problemas de patches, problemas de backup etc. Para fazer isso, o AMS recomenda funções mínimas do IAM, permitindo que somente um grupo específico ou um servidor mestre, se estiver usando uma ferramenta de implantação de aplicativos, acesse as instâncias que executam seu aplicativo.

Regras de firewall de acesso a aplicativos

Assim como o sistema operacional (OS), todo o acesso ao aplicativo deve ser controlado usando grupos do Active Directory (AD). Usando o Amazon Relational Database Service (Amazon RDS) como exemplo, você deve quebrar o espelho (replicação) para adicionar um novo usuário. A melhor abordagem é criar um grupo no AD e adicioná-lo no momento da criação do banco de dados. Ter os grupos em seu AMS AD significa que você pode criar CTs para acesso ao aplicativo. Para obter informações sobre a estratégia de agrupamento oficial do AD, consulte Usando a estratégia de agrupamento de grupos — Melhores práticas do AD para a estratégia de grupo.

Para saber mais sobre árvores de domínio e parent/child domínios, consulte Como funcionam os domínios e as florestas.

As regras a seguir ilustram uma solução apropriada para uma relação de confiança florestal de vários domínios com usuários localizados em domínios secundários.

Instâncias do Windows

Essas são as regras a serem configuradas para seus controladores de domínio pai e filho do Windows.

Controlador de domínio principal, Windows

DE: Controladores de domínio pai ATÉ: pilha do Windows e sub-redes de serviços compartilhados

Porta de origem	Porta de destino	Protocolo
88	49152 – 65535	TCP
389	49152 – 65535	UDP

DE: Sub-redes de pilha, incluindo serviços compartilhados, ATÉ: controladores de domínio raiz de floresta do Windows

Porta de origem	Porta de destino	Protocolo
49152 – 65535	88	TCP
49152 – 65535	389	UDP

Controlador de domínio secundário, Windows

DE: Controladores de domínio secundários ATÉ: controladores de domínio Windows AWS

Porta de origem	Porta de destino	Protocolo
49152 – 65535	53	TCP
49152 – 65535	88	TCP
49152 – 65535	389	UDP

DE: Controladores de domínio secundários ATÉ: pilha do Windows e sub-redes de serviços compartilhados

Porta de origem	Porta de destino	Protocolo
88	49152 – 65535	TCP

Porta de origem	Porta de destino	Protocolo
135	49152 – 65535	TCP
389	49152 – 65535	TCP
389	49152 – 65535	UDP
445	49152 – 65535	TCP
49152 – 65535	49152 – 65535	TCP

DE: Stack sub-redes, incluindo serviços compartilhados, ATÉ: controladores de domínio secundários do Windows

Porta de origem	Porta de destino	Protocolo
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

Instâncias Linux

Essas são as regras a serem configuradas para seus controladores de domínio pai e filho do Linux.

Todos os testes foram realizados usando o Amazon Linux. Enquanto o intervalo de portas dinâmicas para Windows é de 49152 a 65535, muitos kernels Linux usam o intervalo de portas de 32768 a 61000. Execute o comando abaixo para ver o intervalo de portas IP.

```
cat /proc/sys/net/ipv4/ip_local_port_range
```

Controlador de domínio principal, Linux

DE: controladores de domínio pai ATÉ: pilha Linux e sub-redes de serviços compartilhados

Porta de origem	Porta de destino	Protocolo
389	32768 - 61000	UDP
88	32768 - 61000	TCP

DE: Stack sub-redes, incluindo serviços compartilhados, ATÉ: controladores de domínio raiz de floresta Linux

Porta de origem	Porta de destino	Protocolo
32768 - 61000	88	TCP
32768 - 61000	389	UDP

Controlador de domínio secundário, Linux

DE: Controladores de domínio secundários A: controladores de domínio Linux AWS

Porta de origem	Porta de destino	Protocolo
49152 – 65535	53	TCP
49152 – 65535	88	TCP
389	49152 – 65535	UDP
49152 – 65535	389	UDP

DE: Controladores de domínio secundários ATÉ: pilha Linux e sub-redes de serviços compartilhados

Porta de origem	Porta de destino	Protocolo
88	32768 - 61000	TCP

Porta de origem	Porta de destino	Protocolo
389	32768 - 61000	UDP

DE: Stack sub-redes, incluindo serviços compartilhados, ATÉ: controlador de domínio secundário Linux

Porta de origem	Porta de destino	Protocolo
32768 - 61000	88	TCP
32768 - 61000	389	UDP

Gerenciamento de tráfego de saída do AMS

Por padrão, a rota com um CIDR de destino de 0.0.0.0/0 para sub-redes privadas e de aplicativos de clientes do AMS tem um gateway de tradução de endereços de rede (NAT) como destino. Os serviços TrendMicro e patches do AMS são componentes que devem ter acesso de saída à Internet para que o AMS possa fornecer seus serviços TrendMicro e que os sistemas operacionais possam obter atualizações.

O AMS suporta o desvio do tráfego de saída para a Internet por meio de um dispositivo de saída gerenciado pelo cliente, desde que:

- Ele atua como um proxy implícito (por exemplo, transparente).

  and

- Ele permite dependências HTTP e HTTPS do AMS (listadas nesta seção) para permitir a aplicação contínua de patches e a manutenção da infraestrutura gerenciada pelo AMS.

Alguns exemplos são:

- O gateway de trânsito (TGW) tem uma rota padrão apontando para o firewall local gerenciado pelo cliente pela conexão do AWS Direct Connect na conta de rede de zona de destino de várias contas.

- O TGW tem uma rota padrão apontando para um endpoint da AWS na VPC de saída da zona de destino de várias contas utilizando a AWS, apontando para um proxy gerenciado pelo cliente em outra conta PrivateLink da AWS.
- O TGW tem uma rota padrão apontando para um firewall gerenciado pelo cliente em outra conta da AWS, com conexão site-to-site VPN como anexo ao Multi-Account Landing Zone TGW.

O AMS identificou as dependências HTTP e HTTPS correspondentes do AMS e desenvolve e refina essas dependências continuamente. Consulte egressMgmt.zip. Junto com o arquivo JSON, o ZIP contém um README.

> **Note**
>
> - Essas informações não são abrangentes. Alguns sites externos obrigatórios não estão listados aqui.
> - Não use essa lista em uma lista de negação ou estratégia de bloqueio.
> - Essa lista serve como ponto de partida para um conjunto de regras de filtragem de saída, com a expectativa de que as ferramentas de geração de relatórios sejam usadas para determinar com precisão onde o tráfego real diverge da lista.

Para solicitar informações sobre como filtrar o tráfego de saída, envie um e-mail para seu CSDM: ams-csdm@amazon.com.

Grupos de segurança

Na AWS VPCs, os grupos de segurança da AWS atuam como firewalls virtuais, controlando o tráfego de uma ou mais pilhas (uma instância ou um conjunto de instâncias). Quando uma pilha é iniciada, ela é associada a um ou mais grupos de segurança, que determinam qual tráfego pode chegar até ela:

- Para pilhas em suas sub-redes públicas, os grupos de segurança padrão aceitam tráfego de HTTP (80) e HTTPS (443) de todos os locais (a Internet). As pilhas também aceitam tráfego SSH e RDP interno da sua rede corporativa e dos bastiões da AWS. Essas pilhas podem então sair por qualquer porta para a Internet. Eles também podem sair para suas sub-redes privadas e outras pilhas em sua sub-rede pública.

- As pilhas em suas sub-redes privadas podem sair para qualquer outra pilha em sua sub-rede privada, e as instâncias dentro de uma pilha podem se comunicar totalmente entre si por meio de qualquer protocolo.

> **Important**
>
> O grupo de segurança padrão para pilhas em sub-redes privadas permite que todas as pilhas em sua sub-rede privada se comuniquem com outras pilhas nessa sub-rede privada. Se quiser restringir as comunicações entre pilhas em uma sub-rede privada, você deve criar novos grupos de segurança que descrevam a restrição. Por exemplo, se você quiser restringir as comunicações com um servidor de banco de dados para que as pilhas nessa sub-rede privada só possam se comunicar de um servidor de aplicativos específico por meio de uma porta específica, solicite um grupo de segurança especial. Como fazer isso está descrito nesta seção.

Security groups padrão

MALZ

A tabela a seguir descreve as configurações padrão do grupo de segurança de entrada (SG) para suas pilhas. O SG é chamado de "SentinelDefaultSecurityGroupPrivateOnly-VPC-ID”, onde está *ID* um ID de VPC em sua conta de landing zone com várias contas do AMS. Todo o tráfego pode ser enviado para "mc-initial-garden-SentinelDefaultSecurityGroupPrivateOnly" por meio desse grupo de segurança (todo o tráfego local nas sub-redes da pilha é permitido).

Todo o tráfego pode ser enviado para 0.0.0.0/0 por um segundo grupo de segurança "". SentinelDefaultSecurityGroupPrivateOnly

> **Tip**
>
> Se você estiver escolhendo um grupo de segurança para um tipo de alteração do AMS, como criar ou OpenSearch criar domínio do EC2, você usaria um dos grupos de segurança padrão descritos aqui ou um grupo de segurança criado por você. Você pode encontrar a lista de grupos de segurança, por VPC, no console AWS EC2 ou no console VPC.

Há grupos de segurança padrão adicionais que são usados para fins internos do AMS.

Grupos de segurança padrão do AMS (tráfego de entrada)

Tipo	Protocolo	Intervalo de portas	Fonte
Todo o tráfego	Tudo	Todos	SentinelDefaultSecurityGroupPrivateOnly (restringe o tráfego de saída a membros do mesmo grupo de segurança)
Todo o tráfego	Tudo	Todos	SentinelDefaultSecurityGroupPrivateOnlyEgressAll (não restringe o tráfego de saída)
HTTP, HTTPS, SSH, RDP	TCP	80/443 (Fonte 0.0.0.0/0) O acesso SSH e RDP é permitido a partir dos bastiões	SentinelDefaultSecurityGroupPublic (não restringe o tráfego de saída)
Bastiões MALZ:			
SSH	TCP	22	SharedServices VPC CIDR e DMZ VPC CIDR, além de VPC CIDR fornecido pelo cliente no local CIDRs
SSH	TCP	22	
RDP	TCP	3389	
RDP	TCP	3389	
Bastiões de SALZ:			
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

A tabela a seguir descreve as configurações padrão do grupo de segurança de entrada (SG) para suas pilhas. O SG é denominado "mc-initial-garden- SentinelDefaultSecurityGroupPrivateOnly -*ID*", onde *ID* é um identificador exclusivo. Todo o tráfego pode ser enviado para "mc-initial-garden-SentinelDefaultSecurityGroupPrivateOnly" por meio desse grupo de segurança (todo o tráfego local nas sub-redes da pilha é permitido).

Todo o tráfego pode ser enviado para 0.0.0.0/0 por um segundo grupo de segurança "- -". mc-initial-garden SentinelDefaultSecurityGroupPrivateOnlyEgressAll *ID*

> **Tip**
>
> Se você estiver escolhendo um grupo de segurança para um tipo de alteração do AMS, como criar ou OpenSearch criar domínio do EC2, você usaria um dos grupos de segurança padrão descritos aqui ou um grupo de segurança criado por você. Você pode encontrar a lista de grupos de segurança, por VPC, no console AWS EC2 ou no console VPC.

Há grupos de segurança padrão adicionais que são usados para fins internos do AMS.

Grupos de segurança padrão do AMS (tráfego de entrada)

Tipo	Protocolo	Intervalo de portas	Fonte
Todo o tráfego	Tudo	Todos	SentinelDefaultSecurityGroupPrivateOnly (restringe o tráfego de saída a membros do mesmo grupo de segurança)
Todo o tráfego	Tudo	Todos	SentinelDefaultSecurityGroupPrivateOnlyEgressAll (não restringe o tráfego de saída)
HTTP, HTTPS, SSH, RDP	TCP	80/443 (Fonte 0.0.0.0/0) O acesso SSH e RDP é permitido a partir dos bastiões	SentinelDefaultSecurityGroupPublic (não restringe o tráfego de saída)

Tipo	Protocolo	Intervalo de portas	Fonte
Bastiões MALZ:			
SSH	TCP	22	SharedServices VPC CIDR e DMZ VPC CIDR, além de VPC CIDR fornecido pelo cliente no local CIDRs
SSH	TCP	22	
RDP	TCP	3389	
RDP	TCP	3389	
Bastiões de SALZ:			
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

Criar, alterar ou excluir grupos de segurança

Você pode solicitar grupos de segurança personalizados. Nos casos em que os grupos de segurança padrão não atendam às necessidades dos seus aplicativos ou da sua organização, você pode modificar ou criar novos grupos de segurança. Essa solicitação seria considerada necessária para aprovação e seria analisada pela equipe de operações da AMS.

Para criar um grupo de segurança fora das pilhas e VPCs enviar uma RFC usando o tipo de `Deployment | Advanced stack components | Security group | Create (managed automation)` alteração (ct-1oxx2g2d7hc90).

Para modificações no grupo de segurança do Active Directory (AD), use os seguintes tipos de alteração:

- Para adicionar um usuário: Envie um RFC usando Management | Directory Service | Usuários e grupos | Adicionar usuário ao grupo [ct-24pi85mjtza8k]
- Para remover um usuário: envie um RFC usando Management | Directory Service | Usuários e grupos | Remover usuário do grupo [ct-2019s9y3nfml4]

> **Note**
>
> Ao usar o manual CTs, o AMS recomenda que você use a opção ASAP Scheduling (escolha ASAP no console, deixe as horas de início e término em branco na API/CLI), pois elas CTs exigem que um operador do AMS examine a RFC e, possivelmente, se comunique com você antes que ela possa ser aprovada e executada. Se você agendá-los RFCs, aguarde pelo menos 24 horas. Se a aprovação não ocorrer antes do horário de início programado, a RFC será rejeitada automaticamente.

Encontre grupos de segurança

Para encontrar os grupos de segurança anexados a uma pilha ou instância, use o console do EC2. Depois de encontrar a pilha ou instância, você pode ver todos os grupos de segurança anexados a ela.

Para saber como encontrar grupos de segurança na linha de comando e filtrar a saída, consulte `describe-security-groups`.

Apêndice: Questionário de integração de aplicativos

Use esse questionário para descrever seus elementos e estrutura de implantação para que o AMS possa determinar quais componentes de infraestrutura são necessários. Os requisitos de integração para aplicativos Line-of-Business (LoB) são significativamente diferentes dos aplicativos de produtos, portanto, este questionário foi elaborado para abordar ambos.

Tópicos

- Resumo da implantação
- Componentes de implantação de infraestrutura
- Plataforma de hospedagem de aplicativos
- Modelo de implantação de aplicativos
- Dependências de aplicativos
- Certificados SSL para aplicativos de produtos

Resumo da implantação

Uma descrição da implantação. Por exemplo:

- Essa conta é para uma implantação de aplicativo Line-of-Business (LoB) (em oposição à implantação de um aplicativo de produto).
- A implantação envolve um ARP (proxy reverso autenticado) escalonado automaticamente na sub-rede da conta. public/DMZ
- Os servidores web e de aplicativos serão implantados na sub-rede privada da conta.
- Uma instância do Amazon RDS (Amazon Relational Database Service) também será implantada na sub-rede privada da conta.
- Os servidores (ARP, web, aplicativo, banco de dados, balanceador de carga etc.) são separados em grupos de segurança distintos.
- A conta exige um design HA (alta disponibilidade) espalhado pelas zonas de disponibilidade (AZs), ou seja, Multi-AZ.

Componentes de implantação de infraestrutura

Quais são os diferentes componentes que precisarão ser configurados para dar suporte ao seu aplicativo?

- Região: Quais Região da AWS ou regiões são necessárias?
- Alta disponibilidade (HA): quais zonas de disponibilidade serão usadas?
- Virtual Private Cloud (VPC): o que é o bloco CIDR para a VPC?
- Quais instâncias de servidor são necessárias?
  - Proxy reverso autenticado (ARP): sistema operacional, AMI, tipo de instância, ID de sub-rede, grupo de segurança, porta de entrada?
  - Servidor da ferramenta de implantação de aplicativos: sistema operacional, AMI, tipo de instância, ID de sub-rede, grupo de segurança, porta de entrada (Chef, Puppet) ou porta de saída (Ansible, Saltstack)?
  - Amazon RDS com MySQL: versão do banco de dados, tipo de uso, classe da instância, ID da sub-rede, grupo de segurança, ID da instância de banco de dados, tamanho do armazenamento, Multi-AZ, tipo de autenticação, criptografia?
  - Armazenamento: seu aplicativo não tem estado? Você precisa de buckets S3? Você precisa de armazenamento persistente? Você precisa de criptografia de dados em repouso em seus volumes do EBS? Você precisa de criptografia de banco de dados?
  - Endpoints de servidor externos (para o Managed Services VPC): SMTP? SALTO?
  - Requisitos de rede: filtragem de rede (com base em grupos de segurança?)? Inspeção de tráfego na web (entrada? de saída?)?
- Marcação: quais tags devem ser usadas para agrupar recursos em coleções lógicas? Por exemplo, todos os recursos de uma pilha de aplicativos. Selecione tags para seu caso de uso; por exemplo, `backup=true` para ativar backups. Além disso, você deve usar a tag `name=value` para que todas EC2 as instâncias criadas exibam um nome no console.
- Grupos de segurança:
  - Quais grupos de segurança são necessários?
  - Regras de entrada de grupos de segurança?
  - Regras de saída do grupo de segurança?

Plataforma de hospedagem de aplicativos

Para sua plataforma de hospedagem de aplicativos, considere os seguintes requisitos possíveis:

- Bancos de dados criptografados?
- Chaves de criptografia gerenciadas por quem?
- Todos os dados em trânsito e em repouso são criptografados?
- Todos os usuários acessam o sistema via HTTPS?
- Todas as system-to-system interações foram aprovadas pela sua equipe de operações de segurança?

Modelo de implantação de aplicativos

Considerações sobre como você planeja suas implantações de aplicativos. Consulte Qual é o meu modelo operacional?

- Automatizado ou manual? Sem automação de implantação, não há Auto Scale. Se você solicitar acesso, faça login e atualize manualmente seu aplicativo, a atualização falhará. O AMS espera que você reverta sua atualização ou nos alerte por meio de uma solicitação de serviço para que possamos ajudá-lo.
- Se automatizado, qual é a estrutura? Roteiros? Baseado em agente ()puppet/chef)? Agentless (SALT/Ansible? CodeDeploy? As ferramentas de implantação baseadas e sem agente exigem que uma instância separada seja criada e implantada como o servidor mestre das ferramentas. O AMS espera que você conheça todos os elementos necessários para o sucesso das ferramentas de implantação de aplicativos; no entanto, teremos prazer em ajudar com questões relacionadas à infraestrutura.
- Seus Line-of-Business aplicativos (aqueles que você usa para criar e gerenciar seus aplicativos) precisam de patches?

Dependências de aplicativos

Você precisa de instâncias para aplicativos Line-of-Business (LoB)? Para aplicações de produtos?

O que seus aplicativos de produtos precisam para funcionar corretamente?

- Dependências no nível da rede: por exemplo, Direct Connect

- Dependências de pacotes: por exemplo, `pip`
- Aplicativos dos quais esse aplicativo depende: Por exemplo, MySql
- Dependências de firewall?

O que seus aplicativos LoB precisam para funcionar corretamente?

- Dependências no nível da rede: por exemplo, Direct Connect
- Dependências de pacotes: por exemplo, Firefox Saucy
- Aplicativos dos quais esse aplicativo depende: Por exemplo, MySql
- Dependências de firewall?

Certificados SSL para aplicativos de produtos

Quais certificados SSL seus servidores precisarão para que seus aplicativos (LoB e produto) possam alcançar tudo o que precisam para serem executados e acessíveis?

- Grupo de Auto Scaling?
- Banco de dados (Amazon RDS)?
- Balanceador de carga?
- Servidor de ferramentas de implantação?
- Firewall de aplicativos da Web (AWS WAF)?
- Outras instâncias?

Por exemplo, para cada uma das instâncias listadas acima, talvez você precise dos seguintes certificados:

WAF (certificado 1) - > ELB-ext (certificado 2) - > ARP (certificado 3) - > ELB-int (certificado 4) -> Site (certificado 5) - > ELB-int (certificado 6) -> serviço Web (certificado 7).

Histórico do documento

A tabela a seguir descreve a documentação desta versão do AMS.

- Versão da API: 21-05-2019
- Última atualização da documentação: 16 de fevereiro de 2023

Alteração	Description	Link
A remediação de deriva não é suportada para CloudFormation pilhas ingeridas	Foi adicionada uma observação de que a correção de deriva não é suportada pelo AMS para pilhas CloudFormation ingeridas e removeu a menção à correção de deriva da visão geral do processo de ingestão.	CloudFormation Ingestão de AMS
Link do TOC removido	O link do AWS glossário do TOC foi removido.	08 de agosto de 2025
Conteúdo atualizado: Migração de cargas de trabalho: validação pré-ingestão do Windows	Seção atualizada para incluir etapas detalhadas para usar o script de validação pré-WIGS para validar se sua instância do Windows está pronta para ser ingerida em sua conta AMS;.	Migração de workloads: validação de pré-ingestão do Windows
Conteúdo atualizado, configuração do DMS	uma observação importante sobre a função exigida, dms-vpc-role.	1: AWS DMS grupo de sub-rede de replicação: Criar
Conteúdo atualizado, recursos compatíveis com o CFN Ingest	Adicionado OpenSearch.	Recursos compatíveis
Conteúdo atualizado, Migração de cargas de trabalho	Instruções atualizadas para validação antes da ingestão.	Migração de workloads: validação de

Alteração	Description	Link
		pré-ingestão do Windows
Conteúdo atualizado, CFN Ingest.	Os “recursos suportados” restritos foram removidos do conteúdo de ingestão do CFN.	CloudFormation Ingest Stack: recursos compatíveis
Versões atualizadas do Windows suportadas	Suporte adicionado para o Windows Server 2022.	Imagens de máquinas AMS Amazon (AMIs), Migração de cargas de trabalho: pré-requisitos para Linux e Windows, e Migração de workloads: validação de pré-ingestão do Windows
Conteúdo atualizado, Resource Scheduler.	Instruções atualizadas para usar o CT de implantação dedicado, ct-0ywnhc8e5k9z5, aplicável tanto ao SALZ quanto ao MALZ.	Início rápido do AMS Resource Scheduler
Conteúdo atualizado, Workload Ingest.	Versões atualizadas do SUSE Linux suportadas.	Migração de cargas de trabalho: pré-requisitos para Linux e Windows

Alteração	Description	Link
Conteúdo atualizado, Database Migration Service.	Foi adicionado aos pré-requisitos e fez várias alterações de utilidade e usabilidade.	AWS Database Migration Service (AWS DMS)
Conteúdo atualizado, Workload Ingest.	O Zip Pre-WIGS de validação do Linux foi atualizado.	Migração de cargas de trabalho: pré-requisitos para Linux e Windows
Conteúdo atualizado.	Atualizou o zip de validação pré-WIGS para Linux. Além disso, adicionou o Windows Server 2008 R2 como sistema operacional compatível.	Migração de cargas de trabalho: pré-requisitos para Linux e Windows
Novo conteúdo	Os Inícios Rápidos e os Tutoriais foram transferidos para cá do Guia Avançado de Gerenciamento de Alterações do AMS, que foi descontinuado.	Começos rápidos, Tutoriais.
Conteúdo atualizado	Implantação | Componentes avançados de pilha | Serviço de migração de banco de dados (DMS) | Iniciar tarefa de replicação (ct-1yq7h hqse71yg) Atualizado para indicar que os parâmetros DocumentName e Região são obrigatórios; anteriormente, eles eram listados erroneame nte como opcionais.	Serviço de migração de banco de dados (DMS) | Iniciar tarefa de replicação

Alteração	Description	Link
Conteúdo atualizado	CloudFormation Ingerir Atualizado para indicar dois novos recursos suportados, AWS::Route 53Resolver:: ResolverRuleAssociation e AWS::Route 53Resolver::. ResolverRule	Recursos compatíveis
Conteúdo atualizado	Migração de workloads: validação de pré-inges tão do Windows	Informações do Sysprep atualizadas com mais detalhes. Migração de workloads: validação de pré-ingestão do Windows
Conteúdo atualizado	Gerenciamento | Pilha personalizada | Empilhar a partir do CloudFormation modelo | Aprovar conjunto de alterações e atualização (ct-1404e 21baa2ox) A descrição passo a passo do CT para o ChangeSetName parâmetro foi atualizada com informações adicionais.	Empilhar a partir do CloudFormation modelo | Aprovar o conjunto de alterações e atualizar
	Ubuntu 18.04 e Oracle Linux 8.3 disponíveis	Migração de cargas de trabalho: pré-requisitos para Linux e Windows

Alteração	Description	Link
Novo conteúdo:	Implantações de IAM por meio de CTs CFN Ingest e Stack Update.	10 de fevereiro de 2022
Tarefas de replicação do Database Migration Service (DMS)	Os tipos de alteração foram atualizados para que expressões regulares permitam ARNs de tarefas que contenham hífens. Início AWS DMS tarefa de replicaçãoe Serviço de migração de banco de dados (DMS) | Interromper a tarefa de replicação.	13 de janeiro de 2022
Validação de pré-ingestão do Linux WIGS	O arquivo zip foi atualizado. Migração de cargas de trabalho: validação de pré-ingestão do Linux.	13 de janeiro de 2022
Links fixos	A Configuração seção Importação de banco de dados (DB) para AMS SQL RDS -> tinha alguns links incorretos.	13 de janeiro de 2022