aws

Guia do usuário

Application Auto Scaling

Application Auto Scaling : Guia do usuário

Table of Contents

O que é Application Auto Scaling? 1
Recursos do Application Auto Scaling 2
Trabalho com o Application Auto Scaling 2
Conceitos 3
Saiba mais 5
Serviços que se integram 6
Amazon AppStream 2.0 8
Perfil vinculado a serviço 9
Entidade principal do serviço 9
Registrando frotas AppStream 2.0 como alvos escaláveis com o Application Auto Scaling 9
Recursos relacionados 10
Amazon Aurora 10
Perfil vinculado a serviço 10
Entidade principal do serviço 11
Registrar clusters de banco de dados do Aurora como destinos escaláveis com o Application Auto Scaling 11
Recursos relacionados 12
Amazon Comprehend 12
Perfil vinculado a serviço 12
Entidade principal do serviço 12
Registrar recursos do Amazon Comprehend como destinos escaláveis com o Application Auto Scaling 13
Recursos relacionados 14
Amazon DynamoDB 14
Perfil vinculado a serviço 15
Entidade principal do serviço 15
Registrar recursos do DynamoDB como destinos escaláveis com o Application Auto Scaling 15
Recursos relacionados 18
Amazon ECS 18
Perfil vinculado a serviço 18
Entidade principal do serviço 18
Registrar serviços do ECS como destinos escaláveis com o Application Auto Scaling 19
Recursos relacionados 20

Amazon ElastiCache 20
Perfil vinculado a serviço 20
Entidade principal do serviço 21
Registrando ElastiCache recursos como alvos escaláveis com o Application Auto Scaling 21
Recursos relacionados 23
Amazon Keyspaces (para Apache Cassandra) 23
Perfil vinculado a serviço 23
Entidade principal do serviço 23
Registrar as tabelas do Amazon Keyspaces como destinos escaláveis com o Application Auto Scaling 24
Recursos relacionados 25
AWS Lambda 25
Perfil vinculado a serviço 25
Entidade principal do serviço 26
Registrar funções do Lambda como destinos escaláveis com o Application Auto Scaling 26
Recursos relacionados 27
Amazon Managed Streaming for Apache Kafka (MSK) 27
Perfil vinculado a serviço 27
Entidade principal do serviço 28
Registrar o armazenamento de cluster do Amazon MSK como destinos escaláveis com o Application Auto Scaling 28
Recursos relacionados 29
Amazon Neptune 29
Perfil vinculado a serviço 29
Entidade principal do serviço 30
Registrar clusters de banco de dados do Neptune como destinos escaláveis com o Application Auto Scaling 30
Recursos relacionados 31
SageMaker IA da Amazon 31
Perfil vinculado a serviço 31
Entidade principal do serviço 31
Registrando variantes de endpoint de SageMaker IA como alvos escaláveis com o Application Auto Scaling 32
Registrar a simultaneidade provisionada de endpoints sem servidor como destinos escaláveis com o Application Auto Scaling 33

Registrar componentes de inferência como destinos escaláveis com o Application Auto Scaling ... 34
Recursos relacionados ... 34
Frota Spot (Amazon EC2) ... 35
Perfil vinculado a serviço ... 35
Entidade principal do serviço ... 36
Registrar frotas spot como destinos escaláveis com o Application Auto Scaling ... 36
Recursos relacionados ... 37
Amazon WorkSpaces ... 37
Perfil vinculado a serviço ... 37
Entidade principal do serviço ... 37
Registrando WorkSpaces pools como alvos escaláveis com o Application Auto Scaling ... 38
Recursos relacionados ... 39
Recursos personalizados ... 39
Perfil vinculado a serviço ... 39
Entidade principal do serviço ... 39
Registrar recursos personalizados como destinos escaláveis com o Application Auto Scaling ... 39
Recursos relacionados ... 41
Configure o escalonamento usando AWS CloudFormation ... 42
Application Auto Scaling e modelos AWS CloudFormation ... 42
Trechos de modelo de exemplo ... 43
Saiba mais sobre AWS CloudFormation ... 43
Escalabilidade programada ... 44
Como a escalabilidade programada funciona ... 45
Como funcionam ... 45
Considerações ... 45
Comandos normalmente usados ... 46
Recursos relacionados ... 47
Limitações ... 47
Criar ações programadas ... 48
Criar uma ação programada que ocorre apenas uma vez ... 48
Criar uma ação programada que é executada em um intervalo recorrente ... 50
Criar uma ação programada que é executada em uma programação recorrente ... 51
Criar uma única ação programada que especifica um fuso horário ... 51
Criar uma ação programada recorrente que especifica um fuso horário ... 52

Descrever a escalabilidade programada ... 53
Descrever atividades de escalabilidade para um serviço ... 53
Descrever as ações programadas para um serviço ... 55
Descrever uma ou mais ações programadas para um destino escalável ... 57
Programar ações de escalabilidade recorrentes ... 58
Desativar a escalabilidade programada ... 61
Excluir uma ação programada ... 62
Políticas de escalabilidade de rastreamento de destino ... 64
Como funciona o rastreamento de destino ... 65
Como funcionam ... 66
Escolher métricas ... 67
Definir valor de objetivo ... 68
Definir períodos de esfriamento ... 68
Considerações ... 70
Várias políticas de escalabilidade ... 71
Comandos normalmente usados ... 72
Recursos relacionados ... 72
Limitações ... 72
Criar uma política de dimensionamento com monitoramento do objetivo ... 73
Etapa 1: registrar um destino escalável ... 73
Etapa 2: Criar uma política de escalabilidade com monitoramento do objetivo ... 74
Etapa 3: descrever as políticas de escalabilidade com rastreamento de destino ... 77
Excluir uma política de dimensionamento com monitoramento do objetivo ... 78
Usar matemática de métricas ... 79
Exemplo: lista de pendências da fila do Amazon SQS por tarefa ... 79
Limitações ... 84
Políticas de escalabilidade em etapas ... 85
Como funciona a escalabilidade em etapas ... 86
Como funcionam ... 87
Ajustes em etapas ... 87
Tipos de ajuste da escalabilidade ... 90
Período de espera ... 91
Comandos normalmente usados ... 92
Considerações ... 92
Recursos relacionados ... 47
Acesso ao console ... 93

Criar uma política de escalabilidade em etapas 93
Etapa 1: registrar um destino escalável 94
Etapa 2: criar uma política de escalabilidade em etapas 94
Etapa 3: criar um alarme que invoca uma política de escalabilidade 98
Descrever políticas de escalabilidade em etapas 99
Excluir política de escalabilidade em etapas 101
Escalabilidade preditiva 103
Como funciona 103
Limites máximos de capacidade 104
Comandos normalmente usados para criação, exclusão e gerenciamento de política de escalabilidade 105
Considerações 105
Criar uma política de escalabilidade preditiva 106
Substituir a previsão 107
Etapa 1: (Opcional) Analisar dados de séries temporais 108
Etapa 2: Criar duas ações programadas 109
Usar métricas personalizadas 110
Práticas recomendadas 111
Pré-requisitos 111
Estruture o JSON para métricas personalizadas 112
Considerações sobre métricas personalizadas 120
Tutorial: configurar o ajuste de escala automático para processar uma workload pesada 122
Pré-requisitos 122
Etapa 1: inscrever o destino escalável 123
Etapa 2: configurar ações programadas de acordo com as suas necessidades 124
Etapa 3: adicionar uma política de dimensionamento com monitoramento do objetivo 128
Etapa 4: próximas etapas 130
Etapa 5: limpar 131
Suspender a escalabilidade 133
Atividades de escalabilidade 133
Suspender e retomar atividades de escalabilidade 135
Visualizar atividades de escalabilidade suspensas 137
Retomar atividades de escalabilidade 138
Atividades de escalabilidade 140
Pesquisar atividades de escalabilidade por destino escalável 140
Incluir atividades que não sofreram ajuste de escala 141

Códigos de motivo ... 143
Monitoramento ... 147
Monitore usando CloudWatch ... 148
CloudWatch métricas para monitorar o uso de recursos ... 149
Métricas predefinidas para políticas de escalação com rastreamento de destino ... 161
Registre chamadas de API usando CloudTrail ... 164
Eventos de gerenciamento do Application Auto Scaling em CloudTrail ... 165
Exemplos de eventos do Application Auto Scaling ... 165
O Application Auto Scaling ativa RemoveAction CloudWatch ... 167
Amazon EventBridge ... 167
Eventos do Application Auto Scaling ... 167
Trabalhando com AWS SDKs ... 172
Exemplos de código ... 174
Conceitos básicos ... 174
Ações ... 175
Suporte a marcação ... 214
Exemplo de marcação ... 214
Etiquetas para segurança ... 215
Controlar o acesso usando etiquetas ... 216
Segurança ... 218
Proteção de dados ... 219
Gerenciamento de Identidade e Acesso ... 220
Controle de acesso ... 220
Como o Application Auto Scaling funciona com o IAM ... 221
AWS políticas gerenciadas ... 227
Perfis vinculados a serviço ... 238
Exemplos de políticas baseadas em identidade ... 244
Solução de problemas ... 258
Validação de permissões ... 259
AWS PrivateLink ... 261
Criar um VPC endpoint de interface ... 261
Criar uma política de endpoint da VPC ... 262
Resiliência ... 262
Segurança da infraestrutura ... 263
Validação de conformidade ... 263
Cotas ... 265

Histórico de documentos 267
..... cclxxix

O que é Application Auto Scaling?

O Application Auto Scaling é um serviço web para desenvolvedores e administradores de sistemas que precisam de uma solução para escalar automaticamente seus recursos escaláveis para serviços individuais além do AWS Amazon Auto Scaling. EC2 Com o Application Auto Scaling, você pode configurar o escalonamento automático para os seguintes recursos: Com o recursos na Região Secreta: AWS

- AppStream 2.0 frotas
- Réplicas do Aurora
- Classificação de documentos e endpoints de reconhecimento de entidade do Amazon Comprehend
- Tabelas e índices secundários globais do DynamoDB
- Serviços do Amazon ECS
- ElastiCache grupos de replicação (Redis OSS e Valkey) e clusters Memcached
- Clusters do Amazon EMR
- Tabelas do Amazon Keyspaces (for Apache Cassandra)
- Simultaneidade provisionada pela função do Lambda
- Armazenamento de agente do Amazon Managed Streaming for Apache Kafka (MSK)
- Clusters do Amazon Neptune
- SageMaker Variantes de endpoint de IA
- SageMaker Componentes de inferência de IA
- SageMaker Concorrência provisionada por IA sem servidor
- Solicitações de frota spot
- Piscina da Amazon WorkSpaces
- Os recursos personalizados fornecidos por seus próprios aplicativos ou serviços. Para obter mais informações, consulte o GitHubrepositório.

Para ver a disponibilidade regional de qualquer um dos AWS serviços listados acima, consulte a tabela de regiões Tabela de .

Para obter informações sobre como escalar sua frota de EC2 instâncias da Amazon usando grupos do Auto Scaling, consulte o Guia do usuário do Amazon Auto EC2 Scaling.

Recursos do Application Auto Scaling

O Application Auto Scaling permite escalar automaticamente os recursos escaláveis de acordo com as condições definidas por você.

- Escala de rastreamento de metas — Dimensione um recurso com base em um valor alvo para uma CloudWatch métrica específica.
- Escalabilidade em etapas: escale um recurso com base em um conjunto de ajustes de escalabilidade que variam de acordo com o tamanho da ruptura do alarme.
- Escalabilidade programada: escale um recurso apenas uma vez ou em uma programação recorrente.
- Escalabilidade preditiva — Dimensione um recurso de forma proativa para corresponder à carga prevista com base em dados históricos.

Trabalho com o Application Auto Scaling

Você pode configurar a escalabilidade usando as seguintes interfaces, dependendo do recurso que você está escalando:

- AWS Management Console: fornece uma interface da Web que você pode usar para configurar a escalabilidade. Crie uma AWS conta e faça login no AWS Management Console. Abra o console do serviço para um dos recursos listados na introdução. Por exemplo, para escalar uma função Lambda, abra o. AWS Lambda console Certifique-se de abrir o console da Região da AWS mesma forma que o recurso com o qual você deseja trabalhar.

  > **Note**
  >
  > O acesso ao console não está disponível para todos os recursos. Para obter mais informações, consulte Serviços da AWS que você pode usar com o Application Auto Scaling.

- AWS Command Line Interface (AWS CLI) — Fornece comandos para um amplo conjunto de Serviços da AWS e é compatível com Windows, macOS e Linux. Para começar, consulte o AWS Command Line Interface. Para conferir uma lista de comandos, consulte application-autoscaling na AWS CLI Command Reference.

- AWS Tools for Windows PowerShell— Fornece comandos para um amplo conjunto de AWS produtos para quem cria scripts no PowerShell ambiente. Para começar a usar, consulte o Guia do usuário do Ferramentas da AWS para PowerShell. Para obter mais informações, consulte Referência de Cmdlets do Ferramentas da AWS para PowerShell.
- AWS SDKs— fornece operações de API específicas do idioma e cuida de muitos detalhes da conexão, como calcular assinaturas, lidar com novas tentativas de solicitação e lidar com erros. Para obter mais informações, consulte Ferramentas para desenvolver AWS.
- API HTTPS: fornece ações de API de nível inferior que você chama usando solicitações HTTPS. Para obter mais informações, consulte a Referência da API do Application Auto Scaling.
- AWS CloudFormation— Suporta a configuração do dimensionamento usando um CloudFormation modelo. Para obter mais informações, consulte Configurar recursos do Application Auto Scaling usando o AWS CloudFormation.

Para se conectar programaticamente a um AWS service (Serviço da AWS), você usa um endpoint. .

Conceitos do Application Auto Scaling

Este tópico explica conceitos-chave para ajudar a aprender sobre o Application Auto Scaling e começar a usá-lo.

Destinos escaláveis

Uma entidade que você cria para especificar o recurso que deseja dimensionar. Cada destino escalável é identificado exclusivamente por um namespace de serviço, ID de recurso e dimensão escalável, que representa uma dimensão de capacidade do serviço subjacente. Por exemplo, um serviço do Amazon ECS é compatível com escalabilidade automática de sua contagem de tarefas, uma tabela do DynamoDB é compatível com escalabilidade automática da capacidade de leitura e gravação da tabela e de seus índices secundários globais, e um cluster do Aurora é compatível com escalabilidade de sua contagem de réplicas.

> **Tip**
>
> Cada destino escalável também tem capacidades mínima e máxima. As políticas de escalabilidade nunca serão superiores ou inferiores ao intervalo mínimo máximo. Você pode fazer out-of-band alterações diretamente no recurso subjacente que está fora desse intervalo, o que o Application Auto Scaling não conhece. No entanto, sempre que

uma política de escalabilidade for invocada ou a API `RegisterScalableTarget` for chamada, Application Auto Scaling recuperará a capacidade atual e comparará com as capacidades mínima e máxima. Se sair do intervalo mínimo-máximo, então a capacidade será atualizada para cumprir com o mínimo e o máximo definidos.

Reduzir a escala

Quando o Application Auto Scaling diminui automaticamente a capacidade de um destino escalável, o destino escalável reduz a escala. Quando as políticas de escalabilidade estão definidas, elas não podem reduzir a escala horizontalmente no destino dimensionável abaixo de sua capacidade mínima.

Escalonamento horizontal

Quando o Application Auto Scaling diminui automaticamente a capacidade de um destino escalável, o destino escalável aumenta a escala. Quando as políticas de escalabilidade estão definidas, elas não podem aumentar a escala horizontalmente no destino dimensionável acima de sua capacidade máxima.

Política de escalabilidade

Uma política de escalabilidade instrui o Application Auto Scaling a rastrear uma métrica específica. CloudWatch Em seguida, determina a ação de escalabilidade a ser executada quando a métrica é maior ou menor do que um determinado valor limite. Por exemplo, convém aumentar a escala horizontalmente se o uso da CPU em todo o cluster começar a aumentar, e reduzir a escala horizontalmente quando ele cair novamente.

As métricas usadas para escalonamento automático são publicadas pelo serviço de destino, mas você também pode publicar sua própria métrica CloudWatch e usá-la com uma política de escalabilidade.

Um período de desaquecimento entre as atividades de escalabilidade permite que o recurso se estabilize antes que outra atividade de escalabilidade comece. O Application Auto Scaling continua a avaliar métricas durante o período de desaquecimento. Quando o período de desaquecimento termina, a política de escalabilidade inicia outra atividade de escalabilidade se necessário. Enquanto um período de desaquecimento estiver em vigor, se uma escala horizontal maior for necessária com base no valor da métrica atual, a política de escalabilidade aumentará a escala imediatamente.

Ação programada

As ações programadas escalam automaticamente os recursos em uma data e hora específicas. Eles funcionam modificando as capacidades mínima e máxima de um destino escalável e, portanto, podem ser usados para aumentar e reduzir a escala em uma programação, definindo a capacidade mínima alta ou a capacidade máxima baixa. Por exemplo, você pode usar ações programadas para escalar uma aplicação que não consome recursos nos fins de semana, diminuindo a capacidade na sexta-feira e aumentando a capacidade na segunda-feira seguinte.

Você também pode usar ações agendadas para otimizar os valores mínimo e máximo ao longo do tempo para se adaptar a situações em que é esperado um tráfego maior do que o normal, por exemplo, campanhas de marketing ou flutuações sazonais. Isso pode ajudar você a melhorar a performance em momentos em que você precisa aumentar a escala para o uso crescente e reduzir os custos quando você usa menos recursos.

Saiba mais

Serviços da AWS que você pode usar com o Application Auto Scaling: esta seção apresenta os serviços que você pode escalar e ajuda a configurar o Auto Scaling, registrando um destino escalável. Também descreve cada uma das funções vinculadas ao serviço do IAM que o Application Auto Scaling cria para acessar recursos no serviço de destino.

Políticas de dimensionamento com monitoramento do objetivo para o Application Auto Scaling: um dos principais recursos do Application Auto Scaling são as políticas de dimensionamento de monitoramento do objetivo. Saiba como as políticas de monitoramento do objetivo ajustam automaticamente a capacidade desejada para manter a utilização em um nível constante com base na métrica e nos valores de destino configurados. Por exemplo, é possível configurar o monitoramento do objetivo para manter a utilização de CPU da sua frota de servidores da Web em 50%. Em seguida, o Application Auto Scaling inicia ou encerra EC2 instâncias conforme necessário para manter a utilização agregada da CPU em todos os servidores em 50%.

Serviços da AWS que você pode usar com o Application Auto Scaling

O Application Auto Scaling se integra a outros AWS serviços para que você possa adicionar recursos de escalabilidade para atender à demanda do seu aplicativo. A escalabilidade automática é um recurso opcional do serviço que é desabilitado por padrão em quase todos os casos.

A tabela a seguir lista os AWS serviços que você pode usar com o Application Auto Scaling, incluindo informações sobre os métodos suportados para configurar o escalonamento automático. Você também pode usar o Application Auto Scaling com recursos personalizados.

- Acesso ao console: você pode configurar um serviço da AWS compatível para iniciar a escalabilidade automática configurando uma política de escalabilidade no console de serviço de destino.
- Acesso à CLI: você pode configurar um serviço da AWS compatível para iniciar a escalabilidade automática usando a AWS CLI.
- Acesso ao SDK — Você pode configurar um AWS serviço compatível para iniciar o escalonamento automático usando o. AWS SDKs
- CloudFormation access — Você pode configurar um AWS serviço compatível para iniciar o escalonamento automático usando um modelo de AWS CloudFormation pilha. Para obter mais informações, consulte Configurar recursos do Application Auto Scaling usando o AWS CloudFormation.

AWS serviço	Acesso ao console[1]	Acesso à CLI	Acesso ao SDK	CloudFormation acesso
AppStream 2.0	Sim	Sim	Sim	Sim
Aurora	Sim	Sim	Sim	Sim

AWS serviço	Acesso ao console[1]	Acesso à CLI	Acesso ao SDK	CloudFormation acesso
Amazon Comprehend	Não	Sim	Sim	Sim
Amazon DynamoDB	Sim	Sim	Sim	Sim
Amazon ECS	Sim	Sim	Sim	Sim
Amazon ElastiCache	Sim	Sim	Sim	Sim
Amazon EMR	Sim	Sim	Sim	Sim
Amazon Keyspaces	Sim	Sim	Sim	Sim
Lambda	Não	Sim	Sim	Sim
Amazon MSK	Sim	Sim	Sim	Sim

AWS serviço	Acesso ao console[1]	Acesso à CLI	Acesso ao SDK	CloudFormation acesso
Amazon Neptune	Não	Sim	Sim	Sim
SageMaker AI	Sim	Sim	Sim	Sim
Frota spot	Sim	Sim	Sim	Sim
WorkSpaces	Sim	Sim	Sim	Sim
Recursos personalizados	Não	Sim	Sim	Sim

[1] Acesso ao console para configurar políticas de escalabilidade. A maioria dos serviços não oferece suporte à configuração da escalabilidade programada pelo console. Atualmente, somente o Amazon AppStream 2.0 e o Spot Fleet fornecem acesso ao console para escalabilidade programada. ElastiCache

Amazon AppStream 2.0 e Application Auto Scaling

Você pode escalar frotas AppStream 2.0 usando políticas de escalabilidade de rastreamento de metas, políticas de escalabilidade por etapas e escalabilidade programada.

Use as informações a seguir para ajudá-lo a integrar o AppStream 2.0 com o Application Auto Scaling.

Função vinculada ao serviço criada para 2.0 AppStream

A função vinculada ao serviço a seguir é criada automaticamente em você Conta da AWS ao registrar recursos AppStream 2.0 como destinos escaláveis com o Application Auto Scaling. Essa função permite que o Application Auto Scaling realize as operações suportadas em sua conta. Para obter mais informações, consulte Funções vinculadas ao serviço necessárias para o Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_AppStreamFleet`

Principal de serviço primário usado pela função vinculada ao serviço

A função vinculada ao serviço na seção anterior pode ser assumida apenas pelo principal de serviço primário autorizado pelas relações de confiança definidas para a função. A função vinculada ao serviço usada pelo Application Auto Scaling concede acesso aos seguintes principais de serviço:

- `appstream.application-autoscaling.amazonaws.com`

Registrando frotas AppStream 2.0 como alvos escaláveis com o Application Auto Scaling

O Application Auto Scaling exige uma meta escalável antes que você possa criar políticas de escalabilidade ou ações programadas para uma frota 2.0. AppStream Um destino escalável é um recurso cuja escala pode ser aumentada ou reduzida horizontalmente pelo Application Auto Scaling. Os destinos escaláveis são identificados exclusivamente pela combinação de ID de recurso, dimensão escalável e namespace.

Se você configurar o escalonamento automático usando o console AppStream 2.0, o AppStream 2.0 registrará automaticamente uma meta escalável para você.

Se quiser configurar o escalonamento automático usando a AWS CLI ou uma das, você pode usar AWS SDKs as seguintes opções:

- AWS CLI:

  Chame o register-scalable-targetcomando de uma frota AppStream 2.0. O exemplo a seguir registra a capacidade desejada de uma frota chamada `sample-fleet`, com uma capacidade mínima de uma instância de frota e uma capacidade máxima de cinco instâncias de frota.

```
aws application-autoscaling register-scalable-target \
   --service-namespace appstream \
   --scalable-dimension appstream:fleet:DesiredCapacity \
   --resource-id fleet/sample-fleet \
   --min-capacity 1 \
   --max-capacity 5
```

Se obtiver êxito, esse comando retornará o ARN do destino escalável.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
}
```

- AWS SDK:

  Chame a operação RegisterScalableTarget e forneça `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` e `MaxCapacity` como parâmetros.

Recursos relacionados

Para obter mais informações, consulte Fleet Auto Scaling for Amazon AppStream 2.0 no Amazon AppStream 2.0 Administration Guide.

Amazon Aurora e Application Auto Scaling

É possível escalar clusters de banco de dados do Aurora usando políticas de dimensionamento com monitoramento do objetivo, políticas de escalabilidade de etapas e escalabilidade programada.

Use as informações a seguir para ajudar a integrar o Aurora com o Application Auto Scaling.

Criação de uma função vinculada ao serviço para o Aurora

A função vinculada ao serviço a seguir é criada automaticamente em você Conta da AWS ao registrar recursos do Aurora como destinos escaláveis com o Application Auto Scaling. Essa função permite que o Application Auto Scaling realize as operações suportadas em sua conta. Para obter mais informações, consulte Funções vinculadas ao serviço necessárias para o Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_RDSCluster`

Principal de serviço primário usado pela função vinculada ao serviço

A função vinculada ao serviço na seção anterior pode ser assumida apenas pelo principal de serviço primário autorizado pelas relações de confiança definidas para a função. A função vinculada ao serviço usada pelo Application Auto Scaling concede acesso aos seguintes principais de serviço primários:

- `rds.application-autoscaling.amazonaws.com`

Registrar clusters de banco de dados do Aurora como destinos escaláveis com o Application Auto Scaling

O Application Auto Scaling requer um destino escalável para que você possa criar políticas de escalabilidade ou ações programadas para um cluster do Aurora. Um destino escalável é um recurso que pode ser ampliado ou reduzido com o Application Auto Scaling. Os destinos escaláveis são identificados exclusivamente pela combinação de ID de recurso, dimensão escalável e namespace.

Se você configurar a escalabilidade automática usando o console do Aurora, o Aurora inscreverá automaticamente um destino escalável para você.

Se quiser configurar o escalonamento automático usando a AWS CLI ou uma das, você pode usar AWS SDKs as seguintes opções:

- AWS CLI:

  Chame o comando register-scalable-target para um cluster do Aurora. O exemplo a seguir registra a contagem de réplicas do Aurora em um cluster chamado `my-db-cluster`, com uma capacidade mínima de uma réplica do Aurora e capacidade máxima oito réplicas do Aurora.

```
aws application-autoscaling register-scalable-target \
  --service-namespace rds \
  --scalable-dimension rds:cluster:ReadReplicaCount \
  --resource-id cluster:my-db-cluster \
  --min-capacity 1 \
  --max-capacity 8
```

  Se obtiver êxito, esse comando retornará o ARN do destino escalável.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
}
```

- AWS SDK:

  Chame a operação RegisterScalableTarget e forneça `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` e `MaxCapacity` como parâmetros.

Recursos relacionados

Para obter mais informações, consulte Amazon Aurora Auto Scaling with Aurora Replicas no Guia do usuário do Amazon RDS para Aurora.

Amazon Comprehend e Application Auto Scaling

Você pode escalar classificação de documentos e endpoints de reconhecimento de entidade do Amazon Comprehend usando políticas de dimensionamento com monitoramento do objetivo e escalabilidade programada.

Use as informações a seguir para ajudar a integrar o Amazon Comprehend com o Application Auto Scaling.

Criação de uma função vinculada ao serviço para Amazon Comprehend

A seguinte função vinculada ao serviço é criada automaticamente em você Conta da AWS ao registrar os recursos do Amazon Comprehend como alvos escaláveis com o Application Auto Scaling. Essa função permite que o Application Auto Scaling realize as operações suportadas em sua conta. Para obter mais informações, consulte Funções vinculadas ao serviço necessárias para o Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_ComprehendEndpoint`

Principal de serviço primário usado pela função vinculada ao serviço

A função vinculada ao serviço na seção anterior pode ser assumida apenas pelo principal de serviço primário autorizado pelas relações de confiança definidas para a função. A função vinculada ao

serviço usada pelo Application Auto Scaling concede acesso aos seguintes principais de serviço primários:

- `comprehend.application-autoscaling.amazonaws.com`

Registrar recursos do Amazon Comprehend como destinos escaláveis com o Application Auto Scaling

O Application Auto Scaling requer um destino escalável para que você possa criar políticas de escalabilidade ou ações programadas para uma classificação de documento ou endpoint de reconhecimento de entidade do Amazon Comprehend. Um destino escalável é um recurso que pode ser ampliado ou reduzido com o Application Auto Scaling. Os destinos escaláveis são identificados exclusivamente pela combinação de ID de recurso, dimensão escalável e namespace.

Para configurar o escalonamento automático usando a AWS CLI ou uma das, você pode usar AWS SDKs as seguintes opções:

- AWS CLI:

  Chame o comando register-scalable-target para um ponto de extremidade de classificação de documento. O exemplo a seguir registra o número desejado de unidades de inferência a serem usadas pelo modelo para um ponto final de classificação de documentos usando o ARN do endpoint, com uma capacidade mínima de uma unidade de inferência e uma capacidade máxima de três unidades de inferência.

```
aws application-autoscaling register-scalable-target \
  --service-namespace comprehend \
  --scalable-dimension comprehend:document-classifier-endpoint:DesiredInferenceUnits
 \
  --resource-id arn:aws:comprehend:us-west-2:123456789012:document-classifier-
endpoint/EXAMPLE \
  --min-capacity 1 \
  --max-capacity 3
```

  Se obtiver êxito, esse comando retornará o ARN do destino escalável.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
```

```
}
```

Chame o comando register-scalable-target para um endpoint de reconhecimento de entidade. O exemplo a seguir registra o número desejado de unidades de inferência a serem usadas pelo modelo para um reconhecedor de entidade usando o ARN do ponto de extremidade, com uma capacidade mínima de uma unidade de inferência e uma capacidade máxima de três unidades de inferência.

```
aws application-autoscaling register-scalable-target \
  --service-namespace comprehend \
  --scalable-dimension comprehend:entity-recognizer-endpoint:DesiredInferenceUnits \
  --resource-id arn:aws:comprehend:us-west-2:123456789012:entity-recognizer-
endpoint/EXAMPLE \
  --min-capacity 1 \
  --max-capacity 3
```

Se obtiver êxito, esse comando retornará o ARN do destino escalável.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
}
```

- AWS SDK:

  Chame a operação RegisterScalableTarget e forneça `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` e `MaxCapacity` como parâmetros.

Recursos relacionados

Para obter mais informações, consulte Auto scaling with endpoints no Amazon Comprehend Developer Guide.

Amazon DynamoDB e Application Auto Scaling

Você pode escalar tabelas do DynamoDB e índices secundários globais usando políticas de dimensionamento com monitoramento do objetivo e escalabilidade programada.

Use as informações a seguir para ajudar a integrar o DynamoDB com o Application Auto Scaling.

Criação de uma função vinculada ao serviço para DynamoDB

A função vinculada ao serviço a seguir é criada automaticamente em você Conta da AWS ao registrar recursos do DynamoDB como destinos escaláveis com o Application Auto Scaling. Essa função permite que o Application Auto Scaling realize as operações suportadas em sua conta. Para obter mais informações, consulte Funções vinculadas ao serviço necessárias para o Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_DynamoDBTable`

Principal de serviço primário usado pela função vinculada ao serviço

A função vinculada ao serviço na seção anterior pode ser assumida apenas pelo principal de serviço primário autorizado pelas relações de confiança definidas para a função. A função vinculada ao serviço usada pelo Application Auto Scaling concede acesso aos seguintes principais de serviço primários:

- `dynamodb.application-autoscaling.amazonaws.com`

Registrar recursos do DynamoDB como destinos escaláveis com o Application Auto Scaling

O Application Auto Scaling requer um destino escalável para que você possa criar políticas de escalabilidade ou ações programadas para uma tabela do DynamoDB ou índices secundários globais. Um destino escalável é um recurso que pode ser ampliado ou reduzido com o Application Auto Scaling. Os destinos escaláveis são identificados exclusivamente pela combinação de ID de recurso, dimensão escalável e namespace.

Se você configurar a escalabilidade automática usando o console do DynamoDB, o DynamoDB inscreverá automaticamente um destino escalável para você.

Se quiser configurar o escalonamento automático usando a AWS CLI ou uma das, você pode usar AWS SDKs as seguintes opções:

- AWS CLI:

  Chame o register-scalable-targetcomando para obter a capacidade de gravação de uma tabela. O exemplo a seguir registra a capacidade de gravação provisionada de uma tabela chamadamy-

`table`, com uma capacidade mínima de cinco unidades de capacidade de gravação e uma capacidade máxima de 10 unidades de capacidade de gravação:

```
aws application-autoscaling register-scalable-target \
  --service-namespace dynamodb \
  --scalable-dimension dynamodb:table:WriteCapacityUnits \
  --resource-id table/my-table \
  --min-capacity 5 \
  --max-capacity 10
```

Se for bem-sucedido, esse comando retornará o ARN do destino escalável:

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Chame o register-scalable-targetcomando para saber a capacidade de leitura de uma tabela. O exemplo a seguir registra a capacidade de leitura provisionada de uma tabela chamada`my-table`, com uma capacidade mínima de cinco unidades de capacidade de leitura e uma capacidade máxima de 10 unidades de leitura:

```
aws application-autoscaling register-scalable-target \
  --service-namespace dynamodb \
  --scalable-dimension dynamodb:table:ReadCapacityUnits \
  --resource-id table/my-table \
  --min-capacity 5 \
  --max-capacity 10
```

Se for bem-sucedido, esse comando retornará o ARN do destino escalável:

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Chame o register-scalable-targetcomando para obter a capacidade de gravação de um índice secundário global. O exemplo a seguir registra a capacidade de gravação provisionada de um índice secundário global chamado`my-table-index`, com uma capacidade mínima de cinco

unidades de capacidade de gravação e uma capacidade máxima de 10 unidades de capacidade de gravação:

```
aws application-autoscaling register-scalable-target \
  --service-namespace dynamodb \
  --scalable-dimension dynamodb:index:WriteCapacityUnits \
  --resource-id table/my-table/index/my-table-index \
  --min-capacity 5 \
  --max-capacity 10
```

Se for bem-sucedido, esse comando retornará o ARN do destino escalável:

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Chame o register-scalable-targetcomando para obter a capacidade de leitura de um índice secundário global. O exemplo a seguir registra a capacidade de leitura provisionada de um índice secundário global chamado`my-table-index`, com uma capacidade mínima de cinco unidades de capacidade de leitura e uma capacidade máxima de 10 unidades de capacidade de leitura:

```
aws application-autoscaling register-scalable-target \
  --service-namespace dynamodb \
  --scalable-dimension dynamodb:index:ReadCapacityUnits \
  --resource-id table/my-table/index/my-table-index \
  --min-capacity 5 \
  --max-capacity 10
```

Se for bem-sucedido, esse comando retornará o ARN do destino escalável:

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

- AWS SDK:

  Chame a operação RegisterScalableTarget e forneça `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` e `MaxCapacity` como parâmetros.

Recursos relacionados

Se você está apenas começando a usar o Application Auto Scaling, pode encontrar informações adicionais úteis sobre como escalar os recursos do DynamoDB na seguinte documentação:

- Como gerenciar a capacidade de throughput com a autoescalabilidade do DynamoDB no Guia do desenvolvedor do Amazon DynamoDB
- Avaliar as configurações de Auto Scaling da sua tabela no Guia do desenvolvedor do Amazon DynamoDB
- Como usar AWS CloudFormation para configurar o auto scaling para tabelas e índices do DynamoDB no blog AWS

Amazon ECS e Application Auto Scaling

Você pode escalar os serviços do ECS usando políticas de escalabilidade de rastreamento de metas, políticas de escalabilidade preditiva, políticas de escalabilidade por etapas e escalabilidade programada.

Use as informações a seguir para ajudar a integrar o Amazon ECS com o Application Auto Scaling.

Criação de uma função vinculada ao serviço para Amazon ECS

A seguinte função vinculada ao serviço é criada automaticamente em você Conta da AWS ao registrar recursos do Amazon ECS como alvos escaláveis com o Application Auto Scaling. Essa função permite que o Application Auto Scaling realize as operações suportadas em sua conta. Para obter mais informações, consulte Funções vinculadas ao serviço necessárias para o Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_ECSService`

Principal de serviço primário usado pela função vinculada ao serviço

A função vinculada ao serviço na seção anterior pode ser assumida apenas pelo principal de serviço primário autorizado pelas relações de confiança definidas para a função. A função vinculada ao serviço usada pelo Application Auto Scaling concede acesso aos seguintes principais de serviço primários:

- `ecs.application-autoscaling.amazonaws.com`

Registrar serviços do ECS como destinos escaláveis com o Application Auto Scaling

O Application Auto Scaling requer um destino escalável para que você possa criar políticas de escalabilidade ou ações programadas para um serviço do Amazon ECS. Um destino escalável é um recurso que pode ser ampliado ou reduzido com o Application Auto Scaling. Os destinos escaláveis são identificados exclusivamente pela combinação de ID de recurso, dimensão escalável e namespace.

Se você configurar a a escalabilidade automática usando o console do Amazon ECS, o Amazon ECS inscreverá automaticamente um destino escalável para você.

Se quiser configurar o escalonamento automático usando a AWS CLI ou uma das, você pode usar AWS SDKs as seguintes opções:

- AWS CLI:

  Chame o comando register-scalable-target para um serviço do Amazon ECS. O exemplo a seguir inscreve um destino escalável para um serviço chamado `sample-app-service`, rodando no cluster do `default`, com uma contagem mínima de uma tarefa e uma contagem máxima de dez tarefas.

```
aws application-autoscaling register-scalable-target \
  --service-namespace ecs \
  --scalable-dimension ecs:service:DesiredCount \
  --resource-id service/default/sample-app-service \
  --min-capacity 1 \
  --max-capacity 10
```

  Se obtiver êxito, esse comando retornará o ARN do destino escalável.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

- AWS SDK:

Chame a operação RegisterScalableTarget e forneça `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` e `MaxCapacity` como parâmetros.

Recursos relacionados

Se você está apenas começando a usar o Application Auto Scaling, pode encontrar informações adicionais úteis sobre como escalar os recursos do Amazon ECS na seguinte documentação:

- Autoescalabilidade do serviço no Guia do desenvolvedor do Amazon Elastic Container Service
- Otimize o escalonamento automático do serviço Amazon ECS no Guia do desenvolvedor do Amazon Elastic Container Service

> **Note**
>
> Para conferir instruções sobre como suspender os processos de aumento horizontal da escala enquanto as implantações do Amazon ECS estiverem em andamento, consulte a documentação a seguir:
> Escalabilidade automática e implantações do serviço no Guia do desenvolvedor do Amazon Elastic Container Service

ElastiCache e Application Auto Scaling

Você pode escalar horizontalmente grupos de ElastiCache replicação da Amazon (Redis OSS e Valkey) e clusters autoprojetados pelo Memcached usando políticas de escalabilidade de rastreamento de metas e escalabilidade programada.

Para fazer a integração ElastiCache com o Application Auto Scaling, use as informações a seguir.

Criação de uma função vinculada ao serviço para o ElastiCache

A função vinculada ao serviço a seguir é criada automaticamente em você Conta da AWS ao registrar ElastiCache recursos como destinos escaláveis com o Application Auto Scaling. Essa função permite que o Application Auto Scaling realize as operações suportadas em sua conta. Para obter mais informações, consulte Funções vinculadas ao serviço necessárias para o Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG`

Principal de serviço primário usado pela função vinculada ao serviço

A função vinculada ao serviço na seção anterior pode ser assumida apenas pelo principal de serviço primário autorizado pelas relações de confiança definidas para a função. A função vinculada ao serviço usada pelo Application Auto Scaling concede acesso aos seguintes principais de serviço:

- `elasticache.application-autoscaling.amazonaws.com`

Registrando ElastiCache recursos como alvos escaláveis com o Application Auto Scaling

O Application Auto Scaling exige um destino escalável antes que você possa criar políticas de escalabilidade ou ações programadas para um grupo de ElastiCache replicação, cluster ou nó. Um destino escalável é um recurso cuja escala pode ser aumentada ou reduzida horizontalmente pelo Application Auto Scaling. Os destinos escaláveis são identificados exclusivamente pela combinação de ID de recurso, dimensão escalável e namespace.

Se você configurar o escalonamento automático usando o ElastiCache console, registrará ElastiCache automaticamente uma meta escalável para você.

Se quiser configurar o escalonamento automático usando a AWS CLI ou uma das, você pode usar AWS SDKs as seguintes opções:

- AWS CLI:

  Chame o register-scalable-targetcomando para um grupo de ElastiCache replicação. O exemplo a seguir inscreve o número desejado de grupos de nós para um grupo de replicação chamado `mycluster1`, com uma capacidade mínima de um e uma capacidade máxima de cinco.

```
aws application-autoscaling register-scalable-target \
   --service-namespace elasticache \
   --scalable-dimension elasticache:replication-group:NodeGroups \
   --resource-id replication-group/mycluster1 \
   --min-capacity 1 \
   --max-capacity 5
```

  Se obtiver êxito, esse comando retornará o ARN do destino escalável.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
}
```

O exemplo a seguir registra o número desejado de réplicas por grupo de nós para um grupo de replicação chamado`mycluster2`, com uma capacidade mínima de uma e uma capacidade máxima de cinco.

```
aws application-autoscaling register-scalable-target \
   --service-namespace elasticache \
   --scalable-dimension elasticache:replication-group:Replicas \
   --resource-id replication-group/mycluster2 \
   --min-capacity 1 \
   --max-capacity 5
```

Se obtiver êxito, esse comando retornará o ARN do destino escalável.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
target/234abcd56ab78cd901ef1234567890ab1234"
}
```

O exemplo a seguir registra o número desejado de nós para um cluster chamado`mynode1`, com uma capacidade mínima de 20 e uma capacidade máxima de 50.

```
aws application-autoscaling register-scalable-target \
   --service-namespace elasticache \
   --scalable-dimension elasticache:cache-cluster:Nodes \
   --resource-id cache-cluster/mynode1 \
   --min-capacity 20 \
   --max-capacity 50
```

Se obtiver êxito, esse comando retornará o ARN do destino escalável.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
target/01234abcd56ab78cd901ef1234567890ab12"
```

```
}
```

- AWS SDK:

  Chame a operação RegisterScalableTarget e forneça `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` e `MaxCapacity` como parâmetros.

Recursos relacionados

Para obter mais informações, consulte Auto Scaling Valkey e Redis OSS clusters e Scaling clusters for Memcached no Guia do usuário da Amazon. ElastiCache

Amazon Keyspaces (for Apache Cassandra) e Application Auto Scaling

Você pode escalar tabelas do Amazon Keyspaces usando políticas de dimensionamento com monitoramento do objetivo e escalabilidade programada.

Use as informações a seguir para ajudar a integrar o Amazon Keyspaces ao Application Auto Scaling.

Criação de uma função vinculada ao serviço para Amazon Keyspaces

A seguinte função vinculada ao serviço é criada automaticamente em você Conta da AWS ao registrar recursos do Amazon Keyspaces como alvos escaláveis com o Application Auto Scaling. Essa função permite que o Application Auto Scaling realize as operações suportadas em sua conta. Para obter mais informações, consulte Funções vinculadas ao serviço necessárias para o Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_CassandraTable`

Principal de serviço primário usado pela função vinculada ao serviço

A função vinculada ao serviço na seção anterior pode ser assumida apenas pelo principal de serviço primário autorizado pelas relações de confiança definidas para a função. A função vinculada ao serviço usada pelo Application Auto Scaling concede acesso aos seguintes principais de serviço:

- `cassandra.application-autoscaling.amazonaws.com`

Registrar as tabelas do Amazon Keyspaces como destinos escaláveis com o Application Auto Scaling

O Application Auto Scaling requer um destino escalável para que você possa criar políticas de escalabilidade ou ações programadas para uma tabela do Amazon Keyspaces. Um destino escalável é um recurso cuja escala pode ser aumentada ou reduzida na horizontal pelo Application Auto Scaling. Os destinos escaláveis são identificados exclusivamente pela combinação de ID de recurso, dimensão escalável e namespace.

Se você configurar a escalabilidade automática usando o console do Amazon Keyspaces, o Amazon Keyspaces inscreverá automaticamente um destino escalável para você.

Se quiser configurar o escalonamento automático usando a AWS CLI ou uma das, você pode usar AWS SDKs as seguintes opções:

- AWS CLI:

  Chame o register-scalable-target para uma tabela do Amazon Keyspaces. O exemplo a seguir inscreve a capacidade de gravação provisionada de uma tabela chamada `mytable`, com um mínimo cinco unidades de capacidade de gravação e um máximo de dez unidades de capacidade de gravação.

```
aws application-autoscaling register-scalable-target \
  --service-namespace cassandra \
  --scalable-dimension cassandra:table:WriteCapacityUnits \
  --resource-id keyspace/mykeyspace/table/mytable \
  --min-capacity 5 \
  --max-capacity 10
```

  Se obtiver êxito, esse comando retornará o ARN do destino escalável.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

  O exemplo a seguir registra a capacidade de leitura provisionada de uma tabela chamada `mytable`, com um mínimo cinco unidades de capacidade de leitura e um máximo de dez unidades de capacidade de leitura.

```
aws application-autoscaling register-scalable-target \
  --service-namespace cassandra \
  --scalable-dimension cassandra:table:ReadCapacityUnits \
  --resource-id keyspace/mykeyspace/table/mytable \
  --min-capacity 5 \
  --max-capacity 10
```

Se obtiver êxito, esse comando retornará o ARN do destino escalável.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

- AWS SDK:

  Chame a operação RegisterScalableTarget e forneça `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` e `MaxCapacity` como parâmetros.

Recursos relacionados

Para obter mais informações, consulte Gerenciar a capacidade de transferência automaticamente com o escalonamento automático do Amazon Keyspaces no Amazon Keyspaces Developer Guide.

AWS Lambda e Application Auto Scaling

Você pode escalar a simultaneidade AWS Lambda provisionada usando políticas de escalabilidade de rastreamento de metas e escalabilidade programada.

Use as informações a seguir para ajudar a integrar o Lambda com o Application Auto Scaling.

Criação de uma função vinculada ao serviço para o Lambda

A função vinculada ao serviço a seguir é criada automaticamente em você Conta da AWS ao registrar recursos do Lambda como alvos escaláveis com o Application Auto Scaling. Essa função permite que o Application Auto Scaling realize as operações suportadas em sua conta. Para obter mais informações, consulte Funções vinculadas ao serviço necessárias para o Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_LambdaConcurrency`

Principal de serviço primário usado pela função vinculada ao serviço

A função vinculada ao serviço na seção anterior pode ser assumida apenas pelo principal de serviço primário autorizado pelas relações de confiança definidas para a função. A função vinculada ao serviço usada pelo Application Auto Scaling concede acesso aos seguintes principais de serviço primários:

- `lambda.application-autoscaling.amazonaws.com`

Registrar funções do Lambda como destinos escaláveis com o Application Auto Scaling

O Application Auto Scaling requer um destino escalável para que você possa criar políticas de escalabilidade ou ações programadas para uma função do Lambda. Um destino escalável é um recurso que pode ser ampliado ou reduzido com o Application Auto Scaling. Os destinos escaláveis são identificados exclusivamente pela combinação de ID de recurso, dimensão escalável e namespace.

Para configurar o escalonamento automático usando a AWS CLI ou uma das, você pode usar AWS SDKs as seguintes opções:

- AWS CLI:

  Chamar o comando register-scalable-target para uma função do Lambda. O exemplo a seguir registra a simultaneidade provisionada para um alias chamado `BLUE` para uma função chamada `my-function`, com capacidade mínima de 0 e capacidade máxima de 100.

  ```
  aws application-autoscaling register-scalable-target \
    --service-namespace lambda \
    --scalable-dimension lambda:function:ProvisionedConcurrency \
    --resource-id function:my-function:BLUE \
    --min-capacity 0 \
    --max-capacity 100
  ```

  Se obtiver êxito, esse comando retornará o ARN do destino escalável.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
}
```

- AWS SDK:

  Chame a operação RegisterScalableTarget e forneça `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` e `MaxCapacity` como parâmetros.

Recursos relacionados

Se você está apenas começando a usar o Application Auto Scaling, pode encontrar informações adicionais úteis sobre como escalar as funções do Lambda na seguinte documentação:

- Configurar a simultaneidade provisionada no Guia do desenvolvedor do AWS Lambda
- Programando a simultaneidade provisionada do Lambda para pico de uso recorrente no blog AWS

Amazon Managed Streaming for Apache Kafka (MSK) e Application Auto Scaling

Você pode aumentar a escala do armazenamento de cluster do Amazon MSK na horizontal usando políticas de escalabilidade com monitoramento do objetivo. A redução da escala na horizontal pela política de monitoramento do objetivo está desabilitada.

Use as informações a seguir para ajudar a integrar o Amazon MSK com o Application Auto Scaling.

Criação de uma função vinculada ao serviço para Amazon MSK

A seguinte função vinculada ao serviço é criada automaticamente em você Conta da AWS ao registrar recursos do Amazon MSK como alvos escaláveis com o Application Auto Scaling. Essa função permite que o Application Auto Scaling realize as operações suportadas em sua conta. Para obter mais informações, consulte Funções vinculadas ao serviço necessárias para o Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_KafkaCluster`

Principal de serviço primário usado pela função vinculada ao serviço

A função vinculada ao serviço na seção anterior pode ser assumida apenas pelo principal de serviço primário autorizado pelas relações de confiança definidas para a função. A função vinculada ao serviço usada pelo Application Auto Scaling concede acesso aos seguintes principais de serviço primários:

- `kafka.application-autoscaling.amazonaws.com`

Registrar o armazenamento de cluster do Amazon MSK como destinos escaláveis com o Application Auto Scaling

O Application Auto Scaling requer um destino escalável antes de criar uma política de escalabilidade para o tamanho do volume de armazenamento por agente de um cluster do Amazon MSK. Um destino escalável é um recurso que pode ser escalado com o Application Auto Scaling. Os destinos escaláveis são identificados exclusivamente pela combinação de ID de recurso, dimensão escalável e namespace.

Se você configurar a escalabilidade automática usando o console do Amazon MSK, o Amazon MSK registrará automaticamente um destino escalável para você.

Se quiser configurar o escalonamento automático usando a AWS CLI ou uma das, você pode usar AWS SDKs as seguintes opções:

- AWS CLI:

  Chame o comando register-scalable-target para um cluster do Amazon MSK. O exemplo a seguir registra o tamanho do volume de armazenamento por agente de um cluster do Amazon MSK, com capacidade mínima de 100 GiB e capacidade máxima de 800 GiB.

```
aws application-autoscaling register-scalable-target \
   --service-namespace kafka \
   --scalable-dimension kafka:broker-storage:VolumeSize \
   --resource-id arn:aws:kafka:us-east-1:123456789012:cluster/demo-
cluster-1/6357e0b2-0e6a-4b86-a0b4-70df934c2e31-5 \
   --min-capacity 100 \
   --max-capacity 800
```

  Se obtiver êxito, esse comando retornará o ARN do destino escalável.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
}
```

- AWS SDK:

  Chame a operação RegisterScalableTarget e forneça `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` e `MaxCapacity` como parâmetros.

> **Note**
>
> Quando um cluster do Amazon MSK é o destino escalável, a redução é desabilitada e não pode ser ativada.

Recursos relacionados

Para obter mais informações, consulte Escalabilidade automática para clusters do Amazon MSK no Guia do desenvolvedor do Amazon Managed Streaming for Apache Kafka.

Amazon Neptune e Application Auto Scaling

Você pode escalar clusters do Neptune usando políticas de dimensionamento com monitoramento do objetivo e escalabilidade programada.

Use as informações a seguir para ajudar a integrar o Neptune com o Application Auto Scaling.

Criação de uma função vinculada ao serviço para o Neptune

A função vinculada ao serviço a seguir é criada automaticamente em você Conta da AWS ao registrar os recursos do Neptune como destinos escaláveis com o Application Auto Scaling. Essa função permite que o Application Auto Scaling realize as operações suportadas em sua conta. Para obter mais informações, consulte Funções vinculadas ao serviço necessárias para o Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_NeptuneCluster`

Principal de serviço primário usado pela função vinculada ao serviço

A função vinculada ao serviço na seção anterior pode ser assumida apenas pelo principal de serviço primário autorizado pelas relações de confiança definidas para a função. A função vinculada ao serviço usada pelo Application Auto Scaling concede acesso aos seguintes principais de serviço:

- `neptune.application-autoscaling.amazonaws.com`

Registrar clusters de banco de dados do Neptune como destinos escaláveis com o Application Auto Scaling

O Application Auto Scaling requer um destino escalável para que você possa criar políticas de escalabilidade ou ações programadas para um cluster do Neptune. Um destino escalável é um recurso cuja escala pode ser aumentada ou reduzida horizontalmente pelo Application Auto Scaling. Os destinos escaláveis são identificados exclusivamente pela combinação de ID de recurso, dimensão escalável e namespace.

Para configurar o escalonamento automático usando a AWS CLI ou uma das, você pode usar AWS SDKs as seguintes opções:

- AWS CLI:

  Chame o register-scalable-targetcomando para um cluster Neptune. O exemplo a seguir registra a capacidade desejada de um cluster chamado `mycluster`, com uma capacidade mínima de um e uma capacidade máxima de oito.

```
aws application-autoscaling register-scalable-target \
  --service-namespace neptune \
  --scalable-dimension neptune:cluster:ReadReplicaCount \
  --resource-id cluster:mycluster \
  --min-capacity 1 \
  --max-capacity 8
```

  Se obtiver êxito, esse comando retornará o ARN do destino escalável.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

- AWS SDK:

  Chame a operação RegisterScalableTarget e forneça `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` e `MaxCapacity` como parâmetros.

Recursos relacionados

Para obter mais informações, consulte Escalabilidade automática do número de réplicas em um cluster de banco de dados Amazon Neptune no Guia do usuário do Neptune.

Amazon SageMaker AI e Application Auto Scaling

Você pode escalar variantes de endpoint de SageMaker IA, simultaneidade provisionada para endpoints sem servidor e componentes de inferência usando políticas de escalabilidade de rastreamento de metas, políticas de escalonamento de etapas e escalabilidade programada.

Use as informações a seguir para ajudá-lo a integrar a SageMaker IA com o Application Auto Scaling.

Função vinculada ao serviço criada para IA SageMaker

A função vinculada ao serviço a seguir é criada automaticamente em você Conta da AWS ao registrar recursos de SageMaker IA como alvos escaláveis com o Application Auto Scaling. Essa função permite que o Application Auto Scaling realize as operações suportadas em sua conta. Para obter mais informações, consulte Funções vinculadas ao serviço necessárias para o Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_SageMakerEndpoint`

Principal de serviço primário usado pela função vinculada ao serviço

A função vinculada ao serviço na seção anterior pode ser assumida apenas pelo principal de serviço primário autorizado pelas relações de confiança definidas para a função. A função vinculada ao serviço usada pelo Application Auto Scaling concede acesso aos seguintes principais de serviço:

- `sagemaker.application-autoscaling.amazonaws.com`

Registrando variantes de endpoint de SageMaker IA como alvos escaláveis com o Application Auto Scaling

O Application Auto Scaling exige uma meta escalável antes que você possa criar políticas de escalabilidade ou ações programadas para um modelo de SageMaker IA (variante). Um destino escalável é um recurso cuja escala pode ser aumentada ou reduzida horizontalmente pelo Application Auto Scaling. Os destinos escaláveis são identificados exclusivamente pela combinação de ID de recurso, dimensão escalável e namespace.

Se você configurar o escalonamento automático usando o console de SageMaker IA, a SageMaker IA registrará automaticamente uma meta escalável para você.

Se quiser configurar o escalonamento automático usando a AWS CLI ou uma das, você pode usar AWS SDKs as seguintes opções:

- AWS CLI:

  Chame o register-scalable-targetcomando para uma variante do produto. O exemplo a seguir registra a contagem de instâncias desejada para uma variante de produto chamada `my-variant`, rodando em no endpoint `my-endpoint`, com capacidade mínima de uma instância e capacidade máxima de oito instâncias.

  ```
  aws application-autoscaling register-scalable-target \
    --service-namespace sagemaker \
    --scalable-dimension sagemaker:variant:DesiredInstanceCount \
    --resource-id endpoint/my-endpoint/variant/my-variant \
    --min-capacity 1 \
    --max-capacity 8
  ```

  Se obtiver êxito, esse comando retornará o ARN do destino escalável.

  ```
  {
      "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
  target/1234abcd56ab78cd901ef1234567890ab123"
  }
  ```

- AWS SDK:

  Chame a operação RegisterScalableTarget e forneça `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` e `MaxCapacity` como parâmetros.

Registrar a simultaneidade provisionada de endpoints sem servidor como destinos escaláveis com o Application Auto Scaling

O Application Auto Scaling também requer um destino escalável para você poder criar políticas de escalação ou ações programadas para a simultaneidade provisionada de endpoints sem servidor.

Se você configurar o escalonamento automático usando o console de SageMaker IA, a SageMaker IA registrará automaticamente uma meta escalável para você.

Caso contrário, use um dos seguintes métodos para registrar o destino escalável:

- AWS CLI:

  Chame o register-scalable-targetcomando para uma variante do produto. O exemplo a seguir registra a simultaneidade provisionada de uma variante de produto denominada `my-variant`, em execução no endpoint `my-endpoint`, com capacidade mínima de 1 instância e capacidade máxima de 10 instâncias.

  ```
  aws application-autoscaling register-scalable-target \
    --service-namespace sagemaker \
    --scalable-dimension sagemaker:variant:DesiredProvisionedConcurrency \
    --resource-id endpoint/my-endpoint/variant/my-variant \
    --min-capacity 1 \
    --max-capacity 10
  ```

  Se obtiver êxito, esse comando retornará o ARN do destino escalável.

  ```
  {
      "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
  target/1234abcd56ab78cd901ef1234567890ab123"
  }
  ```

- AWS SDK:

  Chame a operação RegisterScalableTarget e forneça `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` e `MaxCapacity` como parâmetros.

Registrar componentes de inferência como destinos escaláveis com o Application Auto Scaling

O Application Auto Scaling também requer um destino escalável para que você possa criar políticas de escalabilidade ou ações programadas para componentes de inferência.

- AWS CLI:

  Chame o register-scalable-targetcomando para um componente de inferência. O exemplo a seguir inscreve o número desejado de cópias para um componente de inferência chamado `my-inference-component`, com uma capacidade mínima de 0 cópia e uma capacidade máxima de 3 cópias.

  ```
  aws application-autoscaling register-scalable-target \
    --service-namespace sagemaker \
    --scalable-dimension sagemaker:inference-component:DesiredCopyCount \
    --resource-id inference-component/my-inference-component \
    --min-capacity 0 \
    --max-capacity 3
  ```

  Se obtiver êxito, esse comando retornará o ARN do destino escalável.

  ```
  {
      "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
  }
  ```

- AWS SDK:

  Chame a operação RegisterScalableTarget e forneça `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` e `MaxCapacity` como parâmetros.

Recursos relacionados

Se você está apenas começando a usar o Application Auto Scaling, você pode encontrar mais informações úteis sobre a escalabilidade de seus recursos de IA no Amazon SageMaker SageMaker AI Developer Guide:

- Dimensione automaticamente os modelos de SageMaker IA da Amazon

- Automatically scale Provisioned Concurrency for a serverless endpoint
- Set auto scaling policies for multi-model endpoint deployments
- Autoscale an asynchronous endpoint

> **Note**
>
> Em 2023, a SageMaker IA introduziu novos recursos de inferência baseados em endpoints de inferência em tempo real. Você cria um endpoint de SageMaker IA com uma configuração de endpoint que define o tipo de instância e a contagem inicial de instâncias para o endpoint. Em seguida, crie um componente de inferência, que é um objeto de hospedagem de SageMaker IA que você pode usar para implantar um modelo em um endpoint. Para obter informações sobre como escalar componentes de inferência, consulte A Amazon SageMaker AI adiciona novos recursos de inferência para ajudar a reduzir os custos e a latência de implantação do modelo básico e reduzir os custos de implantação do modelo em 50%, em média, usando os recursos mais recentes da Amazon SageMaker AI no blog. AWS

Amazon EC2 Spot Fleet e Application Auto Scaling

É possível escalar as frotas spot usando políticas de dimensionamento com monitoramento do objetivo, políticas de escalabilidade de etapas e escalabilidade programada.

Use as informações a seguir para ajudar a integrar frotas spot com o Application Auto Scaling.

Criação de função vinculada ao serviço para frota spot

A seguinte função vinculada ao serviço é criada automaticamente em você Conta da AWS ao registrar os recursos do Spot Fleet como alvos escaláveis com o Application Auto Scaling. Essa função permite que o Application Auto Scaling realize as operações suportadas em sua conta. Para obter mais informações, consulte Funções vinculadas ao serviço necessárias para o Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_EC2SpotFleetRequest`

Principal de serviço primário usado pela função vinculada ao serviço

A função vinculada ao serviço na seção anterior pode ser assumida apenas pelo principal de serviço primário autorizado pelas relações de confiança definidas para a função. A função vinculada ao serviço usada pelo Application Auto Scaling concede acesso aos seguintes principais de serviço primários:

- `ec2.application-autoscaling.amazonaws.com`

Registrar frotas spot como destinos escaláveis com o Application Auto Scaling

O Application Auto Scaling requer um destino escalável para que você possa criar políticas de escalabilidade ou ações programadas para um a frota spot. Um destino escalável é um recurso que pode ser ampliado ou reduzido com o Application Auto Scaling. Os destinos escaláveis são identificados exclusivamente pela combinação de ID de recurso, dimensão escalável e namespace.

Se você configurar a escalabilidade automática usando o console da frota spot, a frota spot inscreverá automaticamente um destino escalável para você.

Se quiser configurar o escalonamento automático usando a AWS CLI ou uma das, você pode usar AWS SDKs as seguintes opções:

- AWS CLI:

  Chame o comando register-scalable-target para uma frota spot. O exemplo a seguir registra a capacidade de destino de uma frota spot usando seu ID de solicitação, com uma capacidade mínima de duas instâncias e uma capacidade máxima de dez instâncias.

```
aws application-autoscaling register-scalable-target \
  --service-namespace ec2 \
  --scalable-dimension ec2:spot-fleet-request:TargetCapacity \
  --resource-id spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
  --min-capacity 2 \
  --max-capacity 10
```

  Se obtiver êxito, esse comando retornará o ARN do destino escalável.

```
{
```

```
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
}
```

- AWS SDK:

  Chame a operação RegisterScalableTarget e forneça `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` e `MaxCapacity` como parâmetros.

Recursos relacionados

Para obter mais informações, consulte Entenda a escalabilidade automática do Spot Fleet no Guia do EC2 usuário da Amazon.

Amazon WorkSpaces e Application Auto Scaling

Você pode escalar um pool WorkSpaces usando políticas de escalabilidade de rastreamento de metas, políticas de escalabilidade por etapas e escalabilidade programada.

Use as informações a seguir para ajudá-lo a se integrar WorkSpaces com o Application Auto Scaling.

Criação de uma função vinculada ao serviço para o WorkSpaces

O Application Auto Scaling cria automaticamente a função vinculada ao serviço chamada AWSServiceRoleForApplicationAutoScaling_WorkSpacesPool em seu Conta da AWS quando você registra WorkSpaces recursos como alvos escaláveis com o Application Auto Scaling. Para obter mais informações, consulte Funções vinculadas ao serviço necessárias para o Application Auto Scaling.

Essa função vinculada ao serviço usa a política gerenciada. AWSApplication AutoscalingWorkSpacesPoolPolicy Essa política concede ao Application Auto Scaling permissões para ligar para a Amazon WorkSpaces em seu nome. Para obter mais informações, consulte AWSApplicationAutoscalingWorkSpacesPoolPolicyna Referência de política AWS gerenciada.

Principal de serviço primário usado pela função vinculada ao serviço

O perfil vinculado ao serviço conta com a seguinte entidade principal de serviço para assumir o perfil:

- `workspaces.application-autoscaling.amazonaws.com`

Registrando WorkSpaces pools como alvos escaláveis com o Application Auto Scaling

O Application Auto Scaling exige uma meta escalável antes que você possa criar políticas de escalabilidade ou ações programadas para. WorkSpaces Um destino escalável é um recurso cuja escala pode ser aumentada ou reduzida horizontalmente pelo Application Auto Scaling. Os destinos escaláveis são identificados exclusivamente pela combinação de ID de recurso, dimensão escalável e namespace.

Se você configurar o escalonamento automático usando o WorkSpaces console, registrará WorkSpaces automaticamente uma meta escalável para você.

Se quiser configurar o escalonamento automático usando a AWS CLI ou uma das, você pode usar AWS SDKs as seguintes opções:

- AWS CLI:

  Chame o register-scalable-targetcomando para um pool de WorkSpaces. O exemplo a seguir registra a capacidade de destino de um pool WorkSpaces usando sua ID de solicitação, com uma capacidade mínima de dois desktops virtuais e uma capacidade máxima de dez desktops virtuais.

  ```
  aws application-autoscaling register-scalable-target \
    --service-namespace workspaces \
    --resource-id workspacespool/wspool-abcdef012 \
    --scalable-dimension workspaces:workspacespool:DesiredUserSessions \
    --min-capacity 2 \
    --max-capacity 10
  ```

  Se obtiver êxito, esse comando retornará o ARN do destino escalável.

  ```
  {
      "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
  target/1234abcd56ab78cd901ef1234567890ab123"
  }
  ```

- AWS SDK:

  Chame a operação RegisterScalableTarget e forneça `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` e `MaxCapacity` como parâmetros.

Recursos relacionados

Para obter mais informações, consulte Auto Scaling for WorkSpaces Pools no Amazon WorkSpaces Administration Guide.

Recursos personalizados e Application Auto Scaling

É possível escalar recursos personalizados usando políticas de dimensionamento com monitoramento do objetivo, políticas de escalabilidade de etapas e escalabilidade programada.

Use as informações a seguir para ajudar a integrar o recursos personalizados com o Application Auto Scaling.

Função vinculada ao serviço criada para recursos personalizados

A função vinculada ao serviço a seguir é criada automaticamente em você Conta da AWS ao registrar recursos personalizados como destinos escaláveis com o Application Auto Scaling. Essa função permite que o Application Auto Scaling realize as operações suportadas em sua conta. Para obter mais informações, consulte Funções vinculadas ao serviço necessárias para o Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_CustomResource`

Principal de serviço primário usado pela função vinculada ao serviço

A função vinculada ao serviço na seção anterior pode ser assumida apenas pelo principal de serviço primário autorizado pelas relações de confiança definidas para a função. A função vinculada ao serviço usada pelo Application Auto Scaling concede acesso aos seguintes principais de serviço primários:

- `custom-resource.application-autoscaling.amazonaws.com`

Registrar recursos personalizados como destinos escaláveis com o Application Auto Scaling

O Application Auto Scaling requer um destino escalável para que você possa criar políticas de escalabilidade ou ações programadas para um recurso personalizado. Um destino escalável

é um recurso que pode ser ampliado ou reduzido com o Application Auto Scaling. Os destinos escaláveis são identificados exclusivamente pela combinação de ID de recurso, dimensão escalável e namespace.

Para configurar o escalonamento automático usando a AWS CLI ou uma das, você pode usar AWS SDKs as seguintes opções:

- AWS CLI:

  Chame o comando register-scalable-target para um recurso personalizado. O exemplo a seguir registra um recurso personalizado como um destino escalável, com uma contagem mínima desejada de uma unidade de capacidade e uma contagem máxima desejada de dez unidades de capacidade. O arquivo `custom-resource-id.txt` contém uma string que identifica o ID do recurso, que representa o caminho para o recurso personalizado por meio do endpoint do Amazon API Gateway.

  ```
  aws application-autoscaling register-scalable-target \
    --service-namespace custom-resource \
    --scalable-dimension custom-resource:ResourceType:Property \
    --resource-id file://~/custom-resource-id.txt \
    --min-capacity 1 \
    --max-capacity 10
  ```

  Conteúdo de `custom-resource-id.txt`:

  ```
  https://example.execute-api.us-west-2.amazonaws.com/prod/
  scalableTargetDimensions/1-23456789
  ```

  Se obtiver êxito, esse comando retornará o ARN do destino escalável.

  ```
  {
      "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
  target/1234abcd56ab78cd901ef1234567890ab123"
  }
  ```

- AWS SDK:

  Chame a operação RegisterScalableTarget e forneça `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` e `MaxCapacity` como parâmetros.

Recursos relacionados

Se você está apenas começando a usar o Application Auto Scaling, pode encontrar informações adicionais úteis sobre como escalar recursos personalizados na seguinte documentação:

GitHubrepositório

Configurar recursos do Application Auto Scaling usando o AWS CloudFormation

O Application Auto Scaling é integrado com AWS CloudFormation, um serviço que ajuda você a modelar e configurar seus AWS recursos para que você possa gastar menos tempo criando e gerenciando seus recursos e infraestrutura. Você cria um modelo que descreve todos os AWS recursos que você deseja e AWS CloudFormation provisiona e configura esses recursos para você.

Ao usar AWS CloudFormation, você pode reutilizar seu modelo para configurar seus recursos do Application Auto Scaling de forma consistente e repetida. Descreva seus recursos uma vez e, em seguida, provisione os mesmos recursos repetidamente em várias Contas da AWS regiões.

Application Auto Scaling e modelos AWS CloudFormation

Para provisionar e configurar recursos para o Application Auto Scaling e serviços relacionados, você deve entender os modelos do AWS CloudFormation. Os modelos são arquivos de texto formatados em JSON ou YAML. Esses modelos descrevem os recursos que você deseja provisionar em suas AWS CloudFormation pilhas. Se você não estiver familiarizado com JSON ou YAML, você pode usar o AWS CloudFormation Designer para ajudá-lo a começar a usar modelos. AWS CloudFormation Para obter mais informações, consulte O que é o AWS CloudFormation Designer? no Guia do usuário do AWS CloudFormation .

Ao criar um modelo de pilha para recursos do Application Auto Scaling, você deve fornecer o seguinte:

- Um namespace para o serviço de destino (por exemplo, **`appstream`**). Consulte a AWS::ApplicationAutoScaling::ScalableTargetreferência para obter namespaces de serviço.
- Uma dimensão escalável associada ao recurso de destino (por exemplo, **`appstream:fleet:DesiredCapacity`**). Veja a AWS::ApplicationAutoScaling::ScalableTargetreferência para obter dimensões escaláveis.
- Um ID de recurso para o recurso de destino (por exemplo, **`fleet/sample-fleet`**). Consulte a AWS::ApplicationAutoScaling::ScalableTargetreferência para obter informações sobre a sintaxe e exemplos de recursos IDs específicos.
- Uma função vinculada ao serviço do recurso de destino (por exemplo, **`arn:aws:iam::012345678910:role/aws-service-role/appstream.application-autoscaling.amazonaws.com/`**

`AWSServiceRoleForApplicationAutoScaling_AppStreamFleet`). Veja a Referência do ARN da função vinculada ao serviço tabela para obter a função ARNs.

Para saber mais sobre os recursos do Application Auto Scaling, consulte a referência do Application Auto Scaling no Guia do usuário do AWS CloudFormation .

Trechos de modelo de exemplo

Você pode encontrar exemplos de trechos para incluir nos AWS CloudFormation modelos nas seguintes seções do Guia do AWS CloudFormation usuário:

- Para obter exemplos de políticas de escalabilidade e ações programadas, consulte Configurar recursos do Application Auto Scaling com. AWS CloudFormation
- Para obter mais exemplos de políticas de escalabilidade, consulte AWS::ApplicationAutoScaling::ScalingPolicy.

Saiba mais sobre AWS CloudFormation

Para saber mais sobre isso AWS CloudFormation, consulte os seguintes recursos:

- AWS CloudFormation
- AWS CloudFormation Guia do usuário
- AWS CloudFormation API Reference
- Guia do Usuário da Interface de Linha de Comando AWS CloudFormation

Escalabilidade programada para o Application Auto Scaling

Com a escalabilidade programada, é possível configurar a escalabilidade automática para a aplicação com base em alterações de carga previsíveis ao criar ações programadas que aumentam ou diminuem a capacidade em momentos específicos. Isso permite escalar a aplicação de forma proativa para corresponder às alterações de carga previsíveis.

Por exemplo, suponhamos que você experiencie um padrão de tráfego semanal regular, em que a carga aumenta no meio da semana e diminui no final da semana. É possível configurar uma escalabilidade programada no Application Auto Scaling que se alinhe a este padrão:

- Na manhã de quarta-feira, uma ação programada amplia a capacidade ao aumentar a capacidade mínima previamente definida do destino escalável.
- Na noite de sexta-feira, outra ação programada reduz a capacidade ao diminuir a capacidade máxima previamente definida do destino escalável.

Essas ações de escalabilidade programadas permitem otimizar os custos e a performance. A aplicação tem capacidade suficiente para lidar com o pico de tráfego no meio da semana, mas não faz provisionamento excessivo de capacidade desnecessária em outros momentos.

É possível usar a escalabilidade programada e as políticas de escalabilidade em conjunto para obter os benefícios de abordagens proativas e reativas para a escalabilidade. Após a execução de uma ação de escalabilidade programada, a política de escalabilidade pode continuar a tomar decisões sobre a necessidade de escalar ainda mais a capacidade. Isso ajuda a garantir que você tenha capacidade suficiente para lidar com a carga de sua aplicação. Embora sua aplicação seja escalada para atender à demanda, a capacidade atual deve estar dentro das capacidades mínima e máxima definidas pela ação agendada.

Conteúdo

- Como funciona a escalabilidade programada para o Application Auto Scaling
- Crie ações agendadas para o Application Auto Scaling usando o AWS CLI
- Descreva o escalonamento programado para o Application Auto Scaling usando o AWS CLI
- Programar ações de escalabilidade recorrentes usando o Application Auto Scaling
- Desativar a escalabilidade programada para um destino escalável
- Exclua uma ação agendada para o Application Auto Scaling usando o AWS CLI

Como funciona a escalabilidade programada para o Application Auto Scaling

Este tópico descreve como a escalabilidade programada funciona e apresenta as principais considerações que você precisa entender para usá-la de forma eficaz.

Conteúdo

- Como funcionam
- Considerações
- Comandos normalmente usados para criação, exclusão e gerenciamento de ações programadas
- Recursos relacionados
- Limitações

Como funcionam

Para usar a escalabilidade programada, crie ações programadas, que instruem o Application Auto Scaling a executar ações de escalabilidade em momentos específicos. Ao criar uma ação programada, você especifica o destino escalável, quando a ação de escalabilidade deve ocorrer, a capacidade mínima e a capacidade máxima. É possível criar ações programadas para escalar uma única vez ou de forma programada.

No momento especificado, o Application Auto Scaling reduzirá com base nos novos valores de capacidade, comparando a capacidade atual com a capacidade mínima e a capacidade máxima especificada.

- Se a capacidade atual for inferior à capacidade mínima especificada, o Application Auto Scaling aumentará (aumentará a capacidade) para a capacidade mínima especificada.
- Se a capacidade atual for inferior à capacidade mínima especificada, o Application Auto Scaling reduzirá (reduzirá a capacidade) para a capacidade máxima especificada.

Considerações

Ao criar uma ação programada, lembre-se do seguinte:

- Uma ação programada define `MinCapacity` e `MaxCapacity` como o que é especificado pela ação programada na data e hora especificadas. A solicitação pode, opcionalmente, incluir

apenas um desses tamanhos. Por exemplo, você pode criar uma ação programada apenas com a capacidade mínima especificada. Em alguns casos, no entanto, você deve incluir ambos os volumes para garantir que a nova capacidade mínima não seja maior do que a capacidade máxima, ou que a nova capacidade máxima não seja inferior à capacidade mínima.

- Por padrão, as programações recorrentes definidas por você estão no fuso horário UTC (Tempo Universal Coordenado). É possível alterar o fuso para corresponder a seu fuso horário local ou a um fuso horário de outra parte da rede. Quando você especificar um fuso horário que observa o horário de verão, a ação será ajustada automaticamente ao horário de verão (DST). Para obter mais informações, consulte Programar ações de escalabilidade recorrentes usando o Application Auto Scaling.
- Você pode desativar temporariamente a escalabilidade programada para um destino escalável. Isso ajuda você a impedir que ações programadas fiquem ativas sem precisar excluí-las. Em seguida, você pode retomar a escalabilidade programada quando quiser usá-la novamente. Para obter mais informações, consulte Suspender e retomar a escalabilidade do Application Auto Scaling.
- A ordem de execução das ações programadas é respeitada para o mesmo destino escalável, mas não para ações programadas em vários destinos escaláveis.
- Para concluir uma ação programada com êxito, o recurso especificado deve estar em um estado escalável no serviço de destino. Se não estiver, a solicitação falhará e retornará uma mensagem de erro, por exemplo, `Resource Id [ActualResourceId] is not scalable. Reason: The status of all DB instances must be 'available' or 'incompatible-parameters'`.
- Devido à natureza distribuída do Application Auto Scaling e aos serviços de destino, o atraso entre o momento em que a ação programada é acionada e o momento em que o serviço de destino honra a ação de escalabilidade pode ser de alguns segundos. Como as ações programadas são executadas na ordem em que são especificadas, as ações programadas com horas de início próximas umas das outras podem demorar mais para serem executadas.

Comandos normalmente usados para criação, exclusão e gerenciamento de ações programadas

Os comandos comumente usados para trabalhar com ações programadas incluem:

- register-scalable-targetregistrar AWS ou personalizar recursos como alvos escaláveis (um recurso que o Application Auto Scaling pode escalar) e suspender e retomar o escalonamento.

- put-scheduled-actionpara adicionar ou modificar ações agendadas para um alvo escalável existente.
- describe-scaling-activitiespara retornar informações sobre atividades de escalabilidade em uma AWS região.
- describe-scheduled-actionspara retornar informações sobre ações agendadas em uma AWS região.
- delete-scheduled-actionpara excluir uma ação agendada.

Recursos relacionados

Para ver um exemplo detalhado do uso da escalabilidade programada, consulte a postagem do blog Programando a simultaneidade AWS Lambda provisionada para picos de uso recorrente no blog de computação. AWS

Para obter informações sobre a criação de ações programadas para grupos de Auto Scaling, consulte Escalabilidade programada para Amazon EC2 Auto Scaling no Guia do usuário do Amazon Auto EC2 Scaling.

Limitações

As limitações de uso da escalabilidade programada são as seguintes:

- Os nomes das ações programadas devem ser exclusivos por grupo escalável.
- O Application Auto Scaling não fornece precisão no segundo nível em expressões de programação. A melhor resolução ao usar uma expressão cron é um minuto.
- O destino escalável não pode ser um cluster do Amazon MSK. A escalabilidade programada não é compatível com o Amazon MSK.
- O acesso ao console para visualizar, adicionar, atualizar ou remover ações programadas em recursos escaláveis depende do recurso utilizado. Para obter mais informações, consulte Serviços da AWS que você pode usar com o Application Auto Scaling.

Crie ações agendadas para o Application Auto Scaling usando o AWS CLI

Os exemplos a seguir mostram como criar ações agendadas usando o AWS CLI put-scheduled-actioncomando. Ao especificar a nova capacidade, você pode definir uma capacidade mínima, uma capacidade máxima ou as duas.

Esses exemplos usam destinos escaláveis para alguns dos serviços que se integram ao Application Auto Scaling. Para usar um destino escalável diferente, especifique o respectivo namespace em `--service-namespace`, a dimensão escalável em `--scalable-dimension` e o ID do recurso em `--resource-id`.

Ao usar o AWS CLI, lembre-se de que seus comandos são Região da AWS executados no configurado para o seu perfil. Se você deseja executar os comandos em uma região diferente, altere a região padrão para o seu perfil ou use o parâmetro `--region` com o comando.

Exemplos

- Criar uma ação programada que ocorre apenas uma vez
- Criar uma ação programada que é executada em um intervalo recorrente
- Criar uma ação programada que é executada em uma programação recorrente
- Criar uma única ação programada que especifica um fuso horário
- Criar uma ação programada recorrente que especifica um fuso horário

Criar uma ação programada que ocorre apenas uma vez

Para escalar automaticamente seu destino escalável apenas uma vez, em uma data e hora especificadas, use o opção `--schedule "at(`*`yyyy-mm-ddThh:mm:ss`*`)"`.

Example Exemplo: para escalar apenas uma vez

Veja a seguir um exemplo de criação de uma ação programada para aumentar a escala da capacidade em uma data e hora específicas.

Na data e hora especificadas para `--schedule` (22h UTC em 31 de março de 2021), se o valor especificado para `MinCapacity` estiver acima da capacidade atual, o Application Auto Scaling tera a escala ampliada horizontalmente para `MinCapacity`.

Linux, macOS ou Unix

```
aws application-autoscaling put-scheduled-action --service-namespace custom-resource \
  --scalable-dimension custom-resource:ResourceType:Property \
  --resource-id file://~/custom-resource-id.txt \
  --scheduled-action-name scale-out \
  --schedule "at(2021-03-31T22:00:00)" \
  --scalable-target-action MinCapacity=3
```

Windows

```
aws application-autoscaling put-scheduled-action --service-namespace custom-resource ^
  --scalable-dimension custom-resource:ResourceType:Property ^
  --resource-id file://~/custom-resource-id.txt ^
  --scheduled-action-name scale-out ^
  --schedule "at(2021-03-31T22:00:00)" ^
  --scalable-target-action MinCapacity=3
```

Quando essa ação agendada for executada, se a capacidade máxima for menor que o valor especificado para capacidade mínima, você deverá especificar novas capacidades mínima e máxima, e não apenas a capacidade mínima.

Example Exemplo: para escalar apenas uma vez

Veja a seguir um exemplo de criação de uma ação programada para reduzir a escala da capacidade em uma data e hora específicas.

Na data e hora especificadas para `--schedule` (22h30 UTC em 31 de março de 2021), se o valor especificado para `MaxCapacity` estiver abaixo da capacidade atual, o Application Auto Scaling terá a escala reduzida horizontalmente para `MaxCapacity`.

Linux, macOS ou Unix

```
aws application-autoscaling put-scheduled-action --service-namespace custom-resource \
  --scalable-dimension custom-resource:ResourceType:Property \
  --resource-id file://~/custom-resource-id.txt \
  --scheduled-action-name scale-in \
  --schedule "at(2021-03-31T22:30:00)" \
  --scalable-target-action MinCapacity=0,MaxCapacity=0
```

Windows

```
aws application-autoscaling put-scheduled-action --service-namespace custom-resource ^
  --scalable-dimension custom-resource:ResourceType:Property ^
```

```
  --resource-id file://~/custom-resource-id.txt ^
  --scheduled-action-name scale-in ^
  --schedule "at(2021-03-31T22:30:00)" ^
  --scalable-target-action MinCapacity=0,MaxCapacity=0
```

Criar uma ação programada que é executada em um intervalo recorrente

Para agendar a escalabilidade em um intervalo recorrente, use a opção `--schedule "rate(value unit)"`. O valor deve ser um inteiro positivo. A unidade pode ser `minute`, `minutes`, `hour`, `hours`, `day` ou `days`. Para obter mais informações, consulte Expressões de tarifas no Guia EventBridge do usuário da Amazon.

Veja a seguir um exemplo de uma ação programada que usa uma expressão de taxa.

Na programação especificada (a cada cinco horas, começando em 30 de janeiro de 2021 à 0h UTC e terminando em 31 de janeiro de 2021 às 22h UTC), se o valor especificado para `MinCapacity` estiver acima da capacidade atual, o Application Auto Scaling aumentará a escala na horizontal para `MinCapacity`. Se o valor especificado para `MaxCapacity` for inferior à capacidade atual, o Application Auto Scaling reduzirá a escala na horizontal para `MaxCapacity`.

Linux, macOS ou Unix

```
aws application-autoscaling put-scheduled-action --service-namespace ecs \
  --scalable-dimension ecs:service:DesiredCount \
  --resource-id service/my-cluster/my-service \
  --scheduled-action-name my-recurring-action \
  --schedule "rate(5 hours)" \
  --start-time 2021-01-30T12:00:00 \
  --end-time 2021-01-31T22:00:00 \
  --scalable-target-action MinCapacity=3,MaxCapacity=10
```

Windows

```
aws application-autoscaling put-scheduled-action --service-namespace ecs ^
  --scalable-dimension ecs:service:DesiredCount ^
  --resource-id service/my-cluster/my-service ^
  --scheduled-action-name my-recurring-action ^
  --schedule "rate(5 hours)" ^
  --start-time 2021-01-30T12:00:00 ^
  --end-time 2021-01-31T22:00:00 ^
  --scalable-target-action MinCapacity=3,MaxCapacity=10
```

Criar uma ação programada que é executada em uma programação recorrente

Para programar a escalabilidade em uma programação recorrente, use a opção `--schedule "cron(fields)"`. Para obter mais informações, consulte Programar ações de escalabilidade recorrentes usando o Application Auto Scaling.

Veja a seguir um exemplo de uma ação programada que usa uma expressão cron.

Na programação especificada (todo dia às 9h UTC), se o valor especificado para `MinCapacity` for superior à capacidade atual, o Application Auto Scaling reduzirá a escala horizontalmente para `MinCapacity`. Se o valor especificado para `MaxCapacity` for inferior à capacidade atual, o Application Auto Scaling reduzirá a escala na horizontal para `MaxCapacity`.

Linux, macOS ou Unix

```
aws application-autoscaling put-scheduled-action --service-namespace appstream \
  --scalable-dimension appstream:fleet:DesiredCapacity \
  --resource-id fleet/sample-fleet \
  --scheduled-action-name my-recurring-action \
  --schedule "cron(0 9 * * ? *)" \
  --scalable-target-action MinCapacity=10,MaxCapacity=50
```

Windows

```
aws application-autoscaling put-scheduled-action --service-namespace appstream ^
  --scalable-dimension appstream:fleet:DesiredCapacity ^
  --resource-id fleet/sample-fleet ^
  --scheduled-action-name my-recurring-action ^
  --schedule "cron(0 9 * * ? *)" ^
  --scalable-target-action MinCapacity=10,MaxCapacity=50
```

Criar uma única ação programada que especifica um fuso horário

As ações programadas são definidas para o fuso horário UTC por padrão. Para especificar um fuso horário diferente, inclua a opção `--timezone` e especifique o nome canônico do fuso horário (`America/New_York`, por exemplo). Para obter mais informações, consulte https://www.joda.org/joda-time/timezones.html, que fornece informações sobre os fusos horários da IANA que são suportados durante chamadas put-scheduled-action.

Veja a seguir um exemplo que usa uma opção `--timezone` ao criar uma ação programada para escalar capacidade em uma data e hora específicas.

Na data e hora especificadas para `--schedule` (17h horário local em 31 de janeiro de 2021), se o valor especificado para `MinCapacity` estiver acima da capacidade atual, o Application Auto Scaling terá a escala aumentada horizontalmente para `MinCapacity`. Se o valor especificado para `MaxCapacity` for inferior à capacidade atual, o Application Auto Scaling reduzirá a escala na horizontal para `MaxCapacity`.

Linux, macOS ou Unix

```
aws application-autoscaling put-scheduled-action --service-namespace comprehend \
  --scalable-dimension comprehend:document-classifier-endpoint:DesiredInferenceUnits \
  --resource-id arn:aws:comprehend:us-west-2:123456789012:document-classifier-endpoint/
EXAMPLE \
  --scheduled-action-name  my-one-time-action \
  --schedule "at(2021-01-31T17:00:00)" --timezone "America/New_York" \
  --scalable-target-action MinCapacity=1,MaxCapacity=3
```

Windows

```
aws application-autoscaling put-scheduled-action --service-namespace comprehend ^
  --scalable-dimension comprehend:document-classifier-endpoint:DesiredInferenceUnits ^
  --resource-id arn:aws:comprehend:us-west-2:123456789012:document-classifier-endpoint/
EXAMPLE ^
  --scheduled-action-name  my-one-time-action ^
  --schedule "at(2021-01-31T17:00:00)" --timezone "America/New_York" ^
  --scalable-target-action MinCapacity=1,MaxCapacity=3
```

Criar uma ação programada recorrente que especifica um fuso horário

Veja a seguir um exemplo que usa uma opção `--timezone` ao criar uma ação programada recorrente para escalar capacidade. Para obter mais informações, consulte Programar ações de escalabilidade recorrentes usando o Application Auto Scaling.

Na programação especificada (de segunda a sexta-feira às 18h horário local), se o valor especificado para `MinCapacity` for superior à capacidade atual, o Application Auto Scaling aumentará a escala horizontalmente para `MinCapacity`. Se o valor especificado para `MaxCapacity` for inferior à capacidade atual, o Application Auto Scaling reduzirá a escala na horizontal para `MaxCapacity`.

Linux, macOS ou Unix

```
aws application-autoscaling put-scheduled-action --service-namespace lambda \
  --scalable-dimension lambda:function:ProvisionedConcurrency \
  --resource-id function:my-function:BLUE \
  --scheduled-action-name my-recurring-action \
  --schedule "cron(0 18 ? * MON-FRI *)" --timezone "Etc/GMT+9" \
  --scalable-target-action MinCapacity=10,MaxCapacity=50
```

Windows

```
aws application-autoscaling put-scheduled-action --service-namespace lambda ^
  --scalable-dimension lambda:function:ProvisionedConcurrency ^
  --resource-id function:my-function:BLUE ^
  --scheduled-action-name my-recurring-action ^
  --schedule "cron(0 18 ? * MON-FRI *)" --timezone "Etc/GMT+9" ^
  --scalable-target-action MinCapacity=10,MaxCapacity=50
```

Descreva o escalonamento programado para o Application Auto Scaling usando o AWS CLI

Esses exemplos de AWS CLI comandos descrevem atividades de escalabilidade e ações programadas usando recursos de serviços que se integram ao Application Auto Scaling. Para usar um destino escalável diferente, especifique o respectivo namespace em `--service-namespace`, a dimensão escalável em `--scalable-dimension` e o ID do recurso em `--resource-id`.

Ao usar o AWS CLI, lembre-se de que seus comandos são Região da AWS executados no configurado para o seu perfil. Se você deseja executar os comandos em uma região diferente, altere a região padrão para o seu perfil ou use o parâmetro `--region` com o comando.

Exemplos

- Descrever atividades de escalabilidade para um serviço
- Descrever as ações programadas para um serviço
- Descrever uma ou mais ações programadas para um destino escalável

Descrever atividades de escalabilidade para um serviço

Para visualizar as atividades de escalabilidade de todos os destinos escaláveis em um namespace de serviço especificado, use o comando. describe-scaling-activities

O exemplo a seguir recupera as atividades de escalabilidade associadas à namespace de serviço dynamodb.

Linux, macOS ou Unix

```
aws application-autoscaling describe-scaling-activities --service-namespace dynamodb
```

Windows

```
aws application-autoscaling describe-scaling-activities --service-namespace dynamodb
```

Saída

Se o comando tiver êxito, ele gerará uma saída semelhante à mostrada a seguir.

```
{
    "ScalingActivities": [
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Description": "Setting write capacity units to 10.",
            "ResourceId": "table/my-table",
            "ActivityId": "4d1308c0-bbcf-4514-a673-b0220ae38547",
            "StartTime": 1561574415.086,
            "ServiceNamespace": "dynamodb",
            "EndTime": 1561574449.51,
            "Cause": "maximum capacity was set to 10",
            "StatusMessage": "Successfully set write capacity units to 10. Change
 successfully fulfilled by dynamodb.",
            "StatusCode": "Successful"
        },
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Description": "Setting min capacity to 5 and max capacity to 10",
            "ResourceId": "table/my-table",
            "ActivityId": "f2b7847b-721d-4e01-8ef0-0c8d3bacc1c7",
            "StartTime": 1561574414.644,
            "ServiceNamespace": "dynamodb",
            "Cause": "scheduled action name my-second-scheduled-action was triggered",
            "StatusMessage": "Successfully set min capacity to 5 and max capacity to
 10",
            "StatusCode": "Successful"
        },
```

```
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Description": "Setting write capacity units to 15.",
            "ResourceId": "table/my-table",
            "ActivityId": "d8ea4de6-9eaa-499f-b466-2cc5e681ba8b",
            "StartTime": 1561574108.904,
            "ServiceNamespace": "dynamodb",
            "EndTime": 1561574140.255,
            "Cause": "minimum capacity was set to 15",
            "StatusMessage": "Successfully set write capacity units to 15. Change
 successfully fulfilled by dynamodb.",
            "StatusCode": "Successful"
        },
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Description": "Setting min capacity to 15 and max capacity to 20",
            "ResourceId": "table/my-table",
            "ActivityId": "3250fd06-6940-4e8e-bb1f-d494db7554d2",
            "StartTime": 1561574108.512,
            "ServiceNamespace": "dynamodb",
            "Cause": "scheduled action name my-first-scheduled-action was triggered",
            "StatusMessage": "Successfully set min capacity to 15 and max capacity to
 20",
            "StatusCode": "Successful"
        }
    ]
}
```

Para alterar esse comando para que ele recupere as atividades de escalabilidade para apenas um de seus destinos escaláveis, adicione a opção `--resource-id`.

Descrever as ações programadas para um serviço

Para descrever as ações agendadas para todos os destinos escaláveis em um namespace de serviço especificado, use o comando. describe-scheduled-actions

O exemplo a seguir recupera as ações programadas associadas ao namespace de serviço `ec2`.

Linux, macOS ou Unix

```
aws application-autoscaling describe-scheduled-actions --service-namespace ec2
```

Windows

```
aws application-autoscaling describe-scheduled-actions --service-namespace ec2
```

Saída

Se o comando tiver êxito, ele gerará uma saída semelhante à mostrada a seguir.

```
{
    "ScheduledActions": [
        {
            "ScheduledActionName": "my-one-time-action",
            "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledAction:493a6261-fbb9-432d-855d-3c302c14bdb9:resource/ec2/
spot-fleet-request/sfr-107dc873-0802-4402-a901-37294EXAMPLE:scheduledActionName/my-one-
time-action",
            "ServiceNamespace": "ec2",
            "Schedule": "at(2021-01-31T17:00:00)",
            "Timezone": "America/New_York",
            "ResourceId": "spot-fleet-request/sfr-107dc873-0802-4402-
a901-37294EXAMPLE",
            "ScalableDimension": "ec2:spot-fleet-request:TargetCapacity",
            "ScalableTargetAction": {
                "MaxCapacity": 1
            },
            "CreationTime": 1607454792.331
        },
        {
            "ScheduledActionName": "my-recurring-action",
            "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledAction:493a6261-fbb9-432d-855d-3c302c14bdb9:resource/ec2/
spot-fleet-request/sfr-107dc873-0802-4402-a901-37294EXAMPLE:scheduledActionName/my-
recurring-action",
            "ServiceNamespace": "ec2",
            "Schedule": "rate(5 minutes)",
            "ResourceId": "spot-fleet-request/sfr-107dc873-0802-4402-
a901-37294EXAMPLE",
            "ScalableDimension": "ec2:spot-fleet-request:TargetCapacity",
            "StartTime": 1604059200.0,
            "EndTime": 1612130400.0,
            "ScalableTargetAction": {
                "MinCapacity": 3,
                "MaxCapacity": 10
            },
            "CreationTime": 1607454949.719
```

```
        },
        {
            "ScheduledActionName": "my-one-time-action",
            "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledAction:4bce34c7-bb81-4ecf-b776-5c726efb1567:resource/ec2/
spot-fleet-request/sfr-40edeb7b-9ae7-44be-bef2-5c4c8EXAMPLE:scheduledActionName/my-one-
time-action",
            "ServiceNamespace": "ec2",
            "Schedule": "at(2020-12-08T9:36:00)",
            "Timezone": "America/New_York",
            "ResourceId": "spot-fleet-request/sfr-40edeb7b-9ae7-44be-
bef2-5c4c8EXAMPLE",
            "ScalableDimension": "ec2:spot-fleet-request:TargetCapacity",
            "ScalableTargetAction": {
                "MinCapacity": 1,
                "MaxCapacity": 3
            },
            "CreationTime": 1607456031.391
        }
    ]
}
```

Descrever uma ou mais ações programadas para um destino escalável

Para recuperar informações sobre as ações agendadas para um alvo escalável especificado, adicione a `--resource-id` opção ao descrever as ações agendadas usando o describe-scheduled-actionscomando.

Se você incluir a opção `--scheduled-action-names` e especificar o nome de uma ação agendada como seu valor, o comando retornará somente a ação agendada cujo nome é uma correspondência, como mostrado no exemplo a seguir.

Linux, macOS ou Unix

```
aws application-autoscaling describe-scheduled-actions --service-namespace ec2 \
  --resource-id spot-fleet-request/sfr-40edeb7b-9ae7-44be-bef2-5c4c8EXAMPLE \
  --scheduled-action-names my-one-time-action
```

Windows

```
aws application-autoscaling describe-scheduled-actions --service-namespace ec2 ^
```

```
  --resource-id spot-fleet-request/sfr-40edeb7b-9ae7-44be-bef2-5c4c8EXAMPLE ^
  --scheduled-action-names my-one-time-action
```

Saída

Se o comando tiver êxito, ele gerará uma saída semelhante à mostrada a seguir. Se houver mais de um valor fornecido para `--scheduled-action-names`, a saída incluirá todas as ações programadas cujo nome corresponder.

```
{
    "ScheduledActions": [
        {
            "ScheduledActionName": "my-one-time-action",
            "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledAction:4bce34c7-bb81-4ecf-b776-5c726efb1567:resource/ec2/
spot-fleet-request/sfr-40edeb7b-9ae7-44be-bef2-5c4c8EXAMPLE:scheduledActionName/my-one-
time-action",
            "ServiceNamespace": "ec2",
            "Schedule": "at(2020-12-08T9:36:00)",
            "Timezone": "America/New_York",
            "ResourceId": "spot-fleet-request/sfr-40edeb7b-9ae7-44be-
bef2-5c4c8EXAMPLE",
            "ScalableDimension": "ec2:spot-fleet-request:TargetCapacity",
            "ScalableTargetAction": {
                "MinCapacity": 1,
                "MaxCapacity": 3
            },
            "CreationTime": 1607456031.391
        }
    ]
}
```

Programar ações de escalabilidade recorrentes usando o Application Auto Scaling

> **Important**
>
> Para obter ajuda com expressões cron para o Amazon EC2 Auto Scaling, consulte o tópico Programações recorrentes no Guia do usuário do Amazon Auto EC2 Scaling. Com o Amazon

EC2 Auto Scaling, você usa a sintaxe cron tradicional em vez da sintaxe cron personalizada que o Application Auto Scaling usa.

Você pode criar ações programadas para execução segundo uma programação recorrente usando uma expressão cron.

Para criar uma programação recorrente, especifique uma expressão cron e um fuso horário para descrever quando essa ação programada deverá ser repetida. Os valores de fuso horário compatíveis são os nomes canônicos dos fusos horários da IANA compatíveis com Joda Time (como `Etc/GMT+9` ou `Pacific/Tahiti`). Opcionalmente, você pode especificar uma data e hora para a hora de início, a hora de término ou ambas. Para obter um exemplo de comando que usa o AWS CLI para criar uma ação agendada, consulteCriar uma ação programada recorrente que especifica um fuso horário.

O formato da expressão cron compatível consiste em cinco campos separados por espaços: [Minutos] [Horas] [Dia_do_mês] [Mês] [Dia_da_semana] [Ano]. Por exemplo, a expressão cron `30 6 ? * MON *` configura uma ação programada que se repete todas as terças-feiras às 6h30. O asterisco é usado como um curinga para corresponder a todos os valores de um campo.

Para obter mais informações sobre a sintaxe cron para ações programadas do Application Auto Scaling, consulte a referência de expressões Cron no Guia do usuário da Amazon. EventBridge

Quando você criar uma programação recorrente, escolha os horários de início e fim cuidadosamente. Lembre-se do seguinte:

- Se você especificar uma hora de início, o Application Auto Scaling executará a ação nessa hora e depois executará a ação de acordo com a recorrência especificada.
- Se você especificar um horário de término, a ação não será mais repetida após esse horário. O Application Auto Scaling não monitora os valores anteriores e reverte para esses valores anteriores após o horário de término.
- A hora de início e a hora de término devem ser definidas em UTC quando você usa o AWS CLI ou o AWS SDKs para criar ou atualizar uma ação agendada.

Exemplos

Você pode consultar a tabela a seguir ao criar uma programação recorrente para um destino escalável do Application Auto Scaling. Os exemplos a seguir são a sintaxe correta para usar o Application Auto Scaling para criar ou atualizar uma ação programada.

Minutos	Horas	Dia do mês	Mês	Dia da semana	Ano	Significado
0	10	*	*	?	*	Executada às 10h (UTC) todos os dias
15	12	*	*	?	*	Executada às 12h15 (UTC) todos os dias
0	18	?	*	SEG-SEX	*	Executada às 18h (UTC) de segunda a sexta
0	8	1	*	?	*	Executada às 8h (UTC) todo primeiro dia do mês
0/15	*	*	*	?	*	Executada a cada 15 minutos
0/10	*	?	*	SEG-SEX	*	Executada a cada 10 minutos de

Minutos	Horas	Dia do mês	Mês	Dia da semana	Ano	Significado
						segunda a sexta
0/5	8-17	?	*	SEG-SEX	*	Executada a cada 5 minutos de segunda a sexta entre 8h e 17h55 (UTC)

Exceção

Você também pode criar uma expressão cron com um valor de string contendo sete campos. Nesse caso, você pode usar os três primeiros campos para especificar a hora na qual uma ação programada deverá ser executada, incluindo os segundos. A expressão cron completa tem os seguintes campos separados por espaços: [Segundos] [Minutos] [Horas] [Dia_do_mês] [Mês] [Dia_da_semana] [Ano]. Porém, essa abordagem não garante que a ação programada será executada no segundo preciso que você especificar. Além disso, alguns consoles de serviço podem não ser compatíveis com o campo de segundos em uma expressão cron.

Desativar a escalabilidade programada para um destino escalável

Você pode desativar temporariamente a escalabilidade programada sem excluir suas ações programadas. Para obter mais informações, consulte Suspender e retomar a escalabilidade do Application Auto Scaling.

Como suspender a escalabilidade programada

Suspenda o escalonamento programado em um destino escalável usando o register-scalable-targetcomando com a `--suspended-state` opção e especificando `true` o valor do `ScheduledScalingSuspended` atributo, conforme mostrado no exemplo a seguir.

Linux, macOS ou Unix

```
aws application-autoscaling register-scalable-target --service-namespace rds \
```

```
  --scalable-dimension rds:cluster:ReadReplicaCount --resource-id cluster:my-db-cluster
 \
  --suspended-state '{"ScheduledScalingSuspended": true}'
```

Windows

```
aws application-autoscaling register-scalable-target --service-namespace rds ^
  --scalable-dimension rds:cluster:ReadReplicaCount --resource-id cluster:my-db-cluster
 ^
  --suspended-state "{\"ScheduledScalingSuspended\": true}"
```

Saída

Se tiver êxito, esse comando retornará o ARN do destino escalável. O seguinte é um exemplo de saída.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Como retomar a escalabilidade programada

Para retomar a escalabilidade programada, execute o comando register-scalable-target novamente, especificando `false` como o valor para `ScheduledScalingSuspended`.

Exclua uma ação agendada para o Application Auto Scaling usando o AWS CLI

Quando não precisar mais de uma ação programada, você poderá excluí-la.

Como excluir uma ação programada

Use o comando delete-scheduled-action. Se tiver êxito, esse comando não retornará nenhuma saída.

Linux, macOS ou Unix

```
aws application-autoscaling delete-scheduled-action \
  --service-namespace ec2 \
  --scalable-dimension ec2:spot-fleet-request:TargetCapacity \
```

```
  --resource-id spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-37294EXAMPLE \
  --scheduled-action-name my-recurring-action
```

Windows

```
aws application-autoscaling delete-scheduled-action ^
  --service-namespace ec2 ^
  --scalable-dimension ec2:spot-fleet-request:TargetCapacity ^
  --resource-id spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-37294EXAMPLE ^
  --scheduled-action-name my-recurring-action
```

Como cancelar o registro do destino dimensionável:

Se você também tiver terminado de usar o destino escalável, poderá cancelar o registro dele. Use o seguinte comando deregister-scalable-target: Se houver alguma política de escalabilidade ou ação programada que ainda não tiver sido excluída, esse comando cuidará disso. Se tiver êxito, esse comando não retornará nenhuma saída.

Linux, macOS ou Unix

```
aws application-autoscaling deregister-scalable-target \
  --service-namespace ec2 \
  --scalable-dimension ec2:spot-fleet-request:TargetCapacity \
  --resource-id spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-37294EXAMPLE
```

Windows

```
aws application-autoscaling deregister-scalable-target ^
  --service-namespace ec2 ^
  --scalable-dimension ec2:spot-fleet-request:TargetCapacity ^
  --resource-id spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-37294EXAMPLE
```

Políticas de dimensionamento com monitoramento do objetivo para o Application Auto Scaling

Uma política de escalabilidade de rastreamento de destinos escala automaticamente a aplicação com base em um valor de métrica de destino. Isso permite que a aplicação mantenha uma performance ideal e uma eficiência de custos sem a necessidade de intervenção manual.

Com o rastreamento de destinos, você seleciona uma métrica e um valor de destino para representar a utilização média ideal ou o nível de throughput para a aplicação. O Application Auto Scaling cria e gerencia os CloudWatch alarmes que acionam eventos de escalabilidade quando a métrica se desvia do alvo. Isso é semelhante a como um termostato mantém a temperatura desejada.

Por exemplo, digamos que você tenha um aplicativo atualmente executado em uma frota spot e queira que a utilização de CPU da frota permaneça próximo de 50% quando a carga no aplicativo mudar. Isso fornece capacidade extra para lidar com picos de tráfego sem manter um número excessivo de recursos ociosos.

Você pode satisfazer essa necessidade criando uma política de escalabilidade com monitoramento de objetivo visando uma utilização média de 50% da CPU. Em seguida, o Application Auto Scaling aumentará a escala horizontalmente (aumento da capacidade) quando a CPU exceder 50% para lidar com o aumento de carga. Ele reduzirá a escala horizontalmente (diminuição da capacidade) quando a CPU estiver abaixo de 50% para otimizar os custos durante os períodos de baixa utilização.

As políticas de rastreamento de metas eliminam a necessidade de definir manualmente CloudWatch alarmes e ajustes de escala. O Application Auto Scaling lida com isso automaticamente com base no destino definido.

É possível basear as políticas de rastreamento de destinos em métricas definidas previamente ou personalizadas:

- Métricas definidas previamente: correspondem a métricas fornecidas pelo Application Auto Scaling, como a utilização média da CPU ou a contagem média de solicitações por destino.
- Métricas personalizadas — você pode usar a matemática métrica para combinar métricas, aproveitar métricas existentes ou usar suas próprias métricas personalizadas publicadas em. CloudWatch

Escolha uma métrica que realiza alterações inversamente proporcionais a uma alteração na capacidade do seu destino escalável. Portanto, se você dobrar a capacidade, a métrica diminuirá em 50%. Isso permite que os dados de métricas acionem com precisão eventos de escalabilidade proporcionais.

Conteúdo

- Como funciona a escalabilidade com rastreamento de destino para o Application Auto Scaling
- Crie uma política de escalabilidade de rastreamento de metas para o Application Auto Scaling usando o AWS CLI
- Exclua uma política de escalabilidade de rastreamento de destino para o Application Auto Scaling usando o AWS CLI
- Crie uma política de escalabilidade de rastreamento de destino para o Application Auto Scaling usando matemática em métricas

Como funciona a escalabilidade com rastreamento de destino para o Application Auto Scaling

Este tópico descreve como a escalabilidade com rastreamento de destino funciona e apresenta os principais elementos de uma política de escalabilidade com rastreamento de destino.

Conteúdo

- Como funcionam
- Escolher métricas
- Definir valor de objetivo
- Definir períodos de esfriamento
- Considerações
- Várias políticas de escalabilidade
- Comandos normalmente usados para criação, exclusão e gerenciamento de política de escalabilidade
- Recursos relacionados
- Limitações

Como funcionam

Para usar a escalabilidade com rastreamento de destino, crie uma política de escalabilidade com rastreamento de destino e especifique o seguinte:

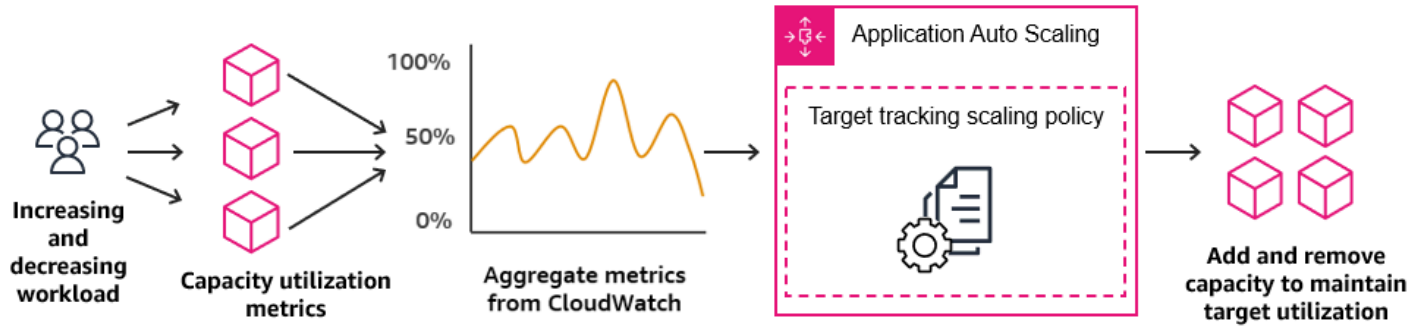
- Métrica — uma CloudWatch métrica a ser monitorada, como a utilização média da CPU ou a contagem média de solicitações por alvo.
- Valor de destino: o valor de destino da métrica, como 50% de utilização da CPU ou mil solicitações por destino por minuto.

O Application Auto Scaling cria e gerencia os CloudWatch alarmes que invocam a política de escalabilidade e calcula o ajuste de escalabilidade com base na métrica e no valor alvo. Ele adiciona e remove capacidade, conforme necessário, para manter a métrica no valor de destino especificado ou próxima a ele.

Quando a métrica está acima do valor de destino, o Application Auto Scaling aumenta a escala horizontalmente ao adicionar capacidade para reduzir a diferença entre o valor da métrica e o valor de destino. Quando a métrica está abaixo do valor de destino, o Application Auto Scaling reduz a escala horizontalmente ao remover a capacidade.

As atividades de escalabilidade são executadas com períodos de esfriamento entre elas para evitar flutuações rápidas na capacidade. Opcionalmente, é possível configurar os períodos de esfriamento para a política de escalabilidade.

O diagrama a seguir mostra uma visão geral de como uma política de escalonamento com monitoramento do destino funciona quando a configuração é concluída.



Observe que uma política de escalabilidade de rastreamento de destinos é mais agressiva na adição de capacidade quando a utilização aumenta do que na remoção de capacidade quando a utilização diminui. Por exemplo, se a métrica especificada da política atingir seu valor do objetivo, a política pressupõe que sua aplicação já esteja muito carregada. Assim, ela responde adicionando

capacidade proporcional ao valor da métrica o mais rápido possível. Quanto maior a métrica, mais capacidade é adicionada.

Quando a métrica fica abaixo do valor de destino, a política espera que a utilização aumente novamente. Nesse caso, ela vai desacelerar a escalabilidade removendo capacidade somente quando a utilização ultrapassar um limite suficientemente abaixo do valor do objetivo (geralmente mais de 10% menor) para que a utilização seja considerada reduzida. A intenção desse comportamento mais conservador é garantir que a remoção de capacidade aconteça somente quando o aplicativo não estiver mais tendo demanda no mesmo alto nível que estava anteriormente.

Escolher métricas

É possível criar políticas de escalabilidade de rastreamento de destino com métricas predefinidas ou personalizadas.

Ao criar uma política de escalação com rastreamento de destino com um tipo de métrica predefinida, você escolhe uma métrica na lista de métricas predefinidas em Métricas predefinidas para políticas de escalação com rastreamento de destino.

Lembre-se do seguinte ao escolher uma métrica:

- Nem todas as métricas personalizadas funcionam para rastreamento de destino. A métrica deve ser de utilização válida e descrever o quão ocupado um destino escalável está. O valor da métrica deve aumentar ou diminuir proporcionalmente à capacidade do destino escalável, de modo que os dados da métrica possam ser usados para escalá-lo proporcionalmente.
- Para usar a métrica `ALBRequestCountPerTarget`, é necessário especificar o parâmetro `ResourceLabel` a fim de identificar o grupo de destino que está associado à métrica.
- Quando uma métrica emite valores reais de 0 para CloudWatch (por exemplo,`ALBRequestCountPerTarget`), o Application Auto Scaling pode ser escalado para 0 quando não há tráfego para seu aplicativo por um longo período de tempo. Para que o seu destino escalável tenha a escala reduzida para 0 quando nenhuma solicitação é roteada, a capacidade mínima do destino escalável deve ser definida como 0.
- Em vez de publicar novas métricas para usar em sua política de escalabilidade, é possível usar a matemática métrica para combinar métricas existentes. Para obter mais informações, consulte Crie uma política de escalabilidade de rastreamento de destino para o Application Auto Scaling usando matemática em métricas.
- Para ver se o serviço que você está usando é compatível com a especificação de uma métrica personalizada no console, consulte a documentação do serviço.

- Recomendamos que você use as métricas que estão disponíveis em intervalos de um minuto para ajudar a escalar mais rapidamente em resposta a alterações na utilização. O rastreamento de destino avaliará as métricas agregadas com uma granularidade de um minuto para todas as métricas predefinidas e personalizadas, mas a métrica subjacente talvez publique os dados com menos frequência. Por exemplo, todas as EC2 métricas da Amazon são enviadas em intervalos de cinco minutos por padrão, mas são configuráveis para um minuto (conhecido como monitoramento detalhado). Essa escolha depende dos serviços individuais. A maioria tenta usar o menor intervalo possível.

Definir valor de objetivo

Ao criar uma política de escalabilidade com monitoramento de objetivo, você deve especificar um valor para o objetivo. O valor-alvo representa o uso ou o throughput médio ideal para o seu aplicativo. Para usar os recursos de maneira econômica, defina o valor do objetivo com o número mais alto possível considerando um buffer razoável para aumentos inesperados de tráfego. Quando seu aplicativo aumentar a escala horizontalmente para um fluxo de tráfego normal, o valor efetivo da métrica deve estar no valor desejado ou logo abaixo dele.

Quando uma política de dimensionamento é baseada no throughput, como o número de solicitações por destino para um Application Load Balancer, E/S de rede ou outras métricas de contabilização, o valor de destino representa o throughput médio ideal de uma única entidade (p. ex., um único destino do seu grupo de destinos do Application Load Balancer), por um período de um minuto.

Definir períodos de esfriamento

Opcionalmente, você pode definir períodos de esfriamento na política de escalação com rastreamento de destino.

O período de esfriamento especifica quanto tempo a política de escalação espera até uma atividade anterior de escalação ter efeito.

Há dois tipos de período de esfriamento:

- Com o período de desaquecimento após expansão, a intenção é expandir de forma contínua (mas não excessiva). Depois que o Application Auto Scaling aumenta a escala horizontalmente com êxito usando uma política de escalação em etapas, ele começa a calcular o tempo de esfriamento. A política de escalação não aumentará a capacidade desejada novamente a menos que um aumento maior da escala horizontal seja disparado ou que o período de esfriamento termine.

Enquanto o período de desaquecimento após expansão estiver em vigor, a capacidade adicionada pela ação de expansão de início será calculada como parte da capacidade desejada para a próxima ação de expansão.

- Com o período de esfriamento da redução da escala horizontal, a intenção é reduzir de maneira conservadora para proteger a disponibilidade da aplicação, de modo que as ações de redução de escala horizontal fiquem bloqueadas o período de esfriamento expirar. No entanto, se outro alarme acionar uma ação de ampliação durante o período de desaquecimento da redução da escala, o Application Auto Scaling expandirá o destino imediatamente. Nesse caso, o período de esfriamento da redução da escala horizontal é interrompido e não é concluído.

Cada período de desaquecimento é medido em segundos e se aplica somente a ações de escalabilidade relacionadas à política. Durante um período de desaquecimento, quando uma ação programada começa no horário programado, ela pode acionar uma ação de escalabilidade imediatamente, sem esperar que o período de desaquecimento expire.

É possível começar com os valores padrão, que podem ser ajustados posteriormente. Por exemplo, talvez seja necessário aumentar um período de desaquecimento para evitar que sua política de escalabilidade de rastreamento de destino seja muito agressiva em relação às alterações que ocorrem em curtos períodos.

Valores padrão

O Application Auto Scaling fornece um valor padrão de 600 para ElastiCache e um valor padrão de 300 para os seguintes destinos escaláveis:

- AppStream 2.0 frotas
- clusters de bancos de dados Aurora
- serviços da ECS
- Clusters do Neptune
- SageMaker Variantes de endpoint de IA
- SageMaker Componentes de inferência de IA
- SageMaker Concorrência provisionada por IA sem servidor
- Spot Fleets
- Piscina de WorkSpaces
- Recursos personalizados

Para todos os outros destinos escaláveis, o valor padrão é 0 ou nulo:

- Classificação de documentos e endpoints de reconhecimento de entidade do Amazon Comprehend
- Tabelas e índices secundários globais do DynamoDB
- Tabelas do Amazon Keyspaces
- Simultaneidade provisionada do Lambda
- Armazenamento de agentes do Amazon MSK

Os valores nulos são tratados da mesma forma que os valores zero quando o Application Auto Scaling avalia o período de esfriamento.

Você pode atualizar qualquer um dos valores padrão, inclusive os valores nulos, para definir seus próprios períodos de esfriamento.

Considerações

As considerações a seguir são aplicáveis ao trabalhar com políticas de escalabilidade com monitoramento de objetivo:

- Não crie, edite ou exclua os CloudWatch alarmes usados com uma política de escalabilidade de rastreamento de metas. O Application Auto Scaling cria e gerencia CloudWatch os alarmes associados às suas políticas de escalabilidade de rastreamento de destino e os exclui quando não são mais necessários.
- Se faltarem pontos de dados na métrica, isso fará com que o estado do CloudWatch alarme mude para`INSUFFICIENT_DATA`. Quando isso acontece, o Application Auto Scaling não poderá dimensionar seu destino dimensionável até que novos pontos de dados sejam encontrados. Para obter mais informações, consulte Configurando como CloudWatch os alarmes tratam dados perdidos no Guia CloudWatch do usuário da Amazon.
- A matemática métrica pode ser útil se a métrica for intencionalmente relatada de maneira esparsa. Por exemplo, para usar os valores mais recentes, use a função `FILL(m1,REPEAT)`, na qual `m1` é a métrica.
- É possível ver lacunas entre o valor de destino e os pontos de dados de métrica reais. Isso ocorre porque o Application Auto Scaling sempre funciona de maneira segura por arredondamento para cima ou para baixo, quando ele determina a capacidade a ser adicionada ou removida. Isso evita que ele adicione capacidade insuficiente ou remova muita capacidade. No entanto, para um

destino dimensionável com capacidade pequena, os pontos de dados de métricas reais podem parecer distantes do valor de destino.

Para um destino dimensionável com maior capacidade, a adição ou remoção de capacidade causa uma lacuna menor entre o valor de destino e os pontos de dados de métricas reais.

- Uma política de escalabilidade de rastreamento de destino pressupõe que ela deve aumentar a escalabilidade quando a métrica especificada estiver acima do valor de destino. Você não pode usar uma política de escalabilidade de rastreamento de destino para expandir quando a métrica especificada estiver abaixo do valor de destino.

Várias políticas de escalabilidade

Você pode ter várias políticas de escalabilidade de rastreamento de destino para um destino escalável, desde que cada uma delas use uma métrica diferente. A intenção do Application Auto Scaling é sempre priorizar a disponibilidade, portanto, seu comportamento será diferente dependendo se as políticas de monitoramento do objetivo estão prontas para aumentar ou reduzir a escala. Ele vai expandir o destino dimensionável se qualquer uma das políticas de rastreamento de destino estiverem prontas para expandir, mas vai reduzir somente se todas as políticas de rastreamento de destino (com a parte de redução habilitada) estiverem prontas para reduzir

Se várias políticas de dimensionamento instruírem o destino dimensionável a aumentar ou reduzir a escala na horizontal ao mesmo tempo, o Application Auto Scaling fará a escalabilidade com base na política que forneça a maior capacidade tanto para aumentar como para reduzir a escala horizontalmente. Isso proporciona maior flexibilidade para abordar vários cenários e garante que sempre haja capacidade suficiente para processar suas workloads.

Você pode desabilitar a parte de redução de escala horizontal de uma política de escalação com rastreamento de destino para usar um método de reduzir a escala horizontalmente diferente do que usa para aumentar a escala horizontalmente. Por exemplo, é possível usar uma política de escalabilidade em etapas pra reduzir ao mesmo tempo que usa uma política de escalabilidade de rastreamento de dentro para expandir,

No entanto, recomendamos cautela ao usar políticas de escalabilidade de rastreamento de destino com políticas de escalabilidade de etapas, pois conflitos entre essas políticas podem causar um comportamento indesejável. Por exemplo, se a política de escalabilidade de etapas iniciar uma atividade de redução antes que a política de rastreamento de destino esteja pronta para ser reduzida, a atividade de redução não será bloqueada. Após a conclusão da atividade de redução, a política de rastreamento de destino pode instruir o destino escalável a expandir novamente.

Para cargas de trabalho de natureza cíclica, você também tem a opção de automatizar alterações de capacidade em uma programação usando escalabilidade programada. Para cada ação programada, um novo valor de capacidade mínima e um novo valor de capacidade máxima podem ser definidos. Esses valores formam os limites da política de escalabilidade. A combinação da escalabilidade programada e da escalabilidade de rastreamento de destino pode ajudar a reduzir o impacto de um aumento acentuado nos níveis de utilização, quando a capacidade é necessária imediatamente.

Comandos normalmente usados para criação, exclusão e gerenciamento de política de escalabilidade

Os comandos comumente usados para trabalhar com políticas de escalabilidade incluem:

- register-scalable-targetregistrar AWS ou personalizar recursos como alvos escaláveis (um recurso que o Application Auto Scaling pode escalar) e suspender e retomar o escalonamento.
- put-scaling-policypara adicionar ou modificar políticas de escalabilidade para um alvo escalável existente.
- describe-scaling-activitiespara retornar informações sobre atividades de escalabilidade em uma AWS região.
- describe-scaling-policiespara retornar informações sobre políticas de escalabilidade em uma AWS região.
- delete-scaling-policypara excluir uma política de escalabilidade.

Recursos relacionados

Para obter informações sobre a criação de políticas de escalabilidade de rastreamento de metas para grupos de Auto Scaling, consulte Políticas de escalabilidade de rastreamento de metas para o Amazon Auto EC2 Scaling no Guia do usuário do Amazon Auto EC2 Scaling.

Limitações

Veja a seguir as limitações ao usar políticas de escalabilidade em etapas:

- O destino escalável não pode ser um cluster do Amazon EMR. As política de dimensionamento com monitoramento do objetivo não são compatíveis com o Amazon EMR.
- Quando um cluster do Amazon MSK é o destino escalável, a redução é desabilitada e não pode ser ativada.

- Você não pode usar as operações da `PutScalingPolicy` API `RegisterScalableTarget` ou da API para atualizar um plano AWS Auto Scaling de escalabilidade.
- O acesso ao console para visualizar, adicionar, atualizar ou remover políticas de escalabilidade de monitoramento do objetivo em recursos escaláveis depende do recurso utilizado. Para obter mais informações, consulte Serviços da AWS que você pode usar com o Application Auto Scaling.

Crie uma política de escalabilidade de rastreamento de metas para o Application Auto Scaling usando o AWS CLI

Este exemplo usa AWS CLI comandos para criar uma política de rastreamento alvo para uma frota EC2 spot da Amazon. Para usar um destino escalável diferente, especifique o respectivo namespace em `--service-namespace`, a dimensão escalável em `--scalable-dimension` e o ID do recurso em `--resource-id`.

Ao usar o AWS CLI, lembre-se de que seus comandos são Região da AWS executados no configurado para seu perfil. Se você deseja executar os comandos em uma região diferente, altere a região padrão para o seu perfil ou use o parâmetro `--region` com o comando.

Tarefas

- Etapa 1: registrar um destino escalável
- Etapa 2: Criar uma política de escalabilidade com monitoramento do objetivo
- Etapa 3: descrever as políticas de escalabilidade com rastreamento de destino

Etapa 1: registrar um destino escalável

Se você ainda não tiver feito isso, inscreva o destino escalável. Use o register-scalable-targetcomando para registrar um recurso específico no serviço de destino como um alvo escalável. O exemplo a seguir inscreve uma solicitação de frota spot com o Application Auto Scaling. O Application Auto Scaling pode escalar o número de instâncias da frota spot de no mínimo duas instâncias e no máximo dez. Substitua cada *`user input placeholder`* por suas próprias informações.

Linux, macOS ou Unix

```
aws application-autoscaling register-scalable-target --service-namespace ec2 \
  --scalable-dimension ec2:spot-fleet-request:TargetCapacity \
```

```
  --resource-id spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
  --min-capacity 2 --max-capacity 10
```

Windows

```
aws application-autoscaling register-scalable-target --service-namespace ec2 ^
  --scalable-dimension ec2:spot-fleet-request:TargetCapacity ^
  --resource-id spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE ^
  --min-capacity 2 --max-capacity 10
```

Saída

Se obtiver êxito, esse comando retornará o ARN do destino escalável. O seguinte é um exemplo de saída.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Etapa 2: Criar uma política de escalabilidade com monitoramento do objetivo

Para criar uma política de escalabilidade com rastreamento de destino, você pode usar os exemplos a seguir para começar.

Para criar uma política de escalabilidade com monitoramento do objetivo

1. Use o comando `cat` a seguir para especificar um valor de destino para a política de escalabilidade e uma especificação de métrica predefinida em um arquivo JSON chamado `config.json` em seu diretório inicial. Veja a seguir um exemplo de configuração de rastreamento de destino que mantém a utilização média da CPU em 50%.

   ```
   $ cat ~/config.json
   {
     "TargetValue": 50.0,
     "PredefinedMetricSpecification":
       {
         "PredefinedMetricType": "EC2SpotFleetRequestAverageCPUUtilization"
       }
   ```

```
}
```

Para obter mais informações, consulte a Referência PredefinedMetricSpecificationda API Application Auto Scaling.

Se preferir, você poderá personalizar a métrica usada para a escalabilidade criando uma especificação de métrica personalizada e adicionando valores para cada parâmetro do CloudWatch. Veja a seguir um exemplo de configuração de rastreamento de destino que mantém a utilização média da métrica especificada em 100.

```
$ cat ~/config.json
{
   "TargetValue": 100.0,
   "CustomizedMetricSpecification":{
      "MetricName": "MyUtilizationMetric",
      "Namespace": "MyNamespace",
      "Dimensions": [
         {
            "Name": "MyOptionalMetricDimensionName",
            "Value": "MyOptionalMetricDimensionValue"
         }
      ],
      "Statistic": "Average",
      "Unit": "Percent"
   }
}
```

Para obter mais informações, consulte a Referência CustomizedMetricSpecificationda API Application Auto Scaling.

2. Use o seguinte comando put-scaling-policy, com o arquivo `config.json` que você criou, para criar uma política de dimensionamento chamada `cpu50-target-tracking-scaling-policy`.

   Linux, macOS ou Unix

```
aws application-autoscaling put-scaling-policy --service-namespace ec2 \
  --scalable-dimension ec2:spot-fleet-request:TargetCapacity \
  --resource-id spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
  --policy-name cpu50-target-tracking-scaling-policy --policy-type
 TargetTrackingScaling \
```

```
  --target-tracking-scaling-policy-configuration file://config.json
```

Windows

```
aws application-autoscaling put-scaling-policy --service-namespace ec2 ^
  --scalable-dimension ec2:spot-fleet-request:TargetCapacity ^
  --resource-id spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE ^
  --policy-name cpu50-target-tracking-scaling-policy --policy-type
 TargetTrackingScaling ^
  --target-tracking-scaling-policy-configuration file://config.json
```

Saída

Se for bem-sucedido, esse comando retornará os nomes ARNs e os dois CloudWatch alarmes criados em seu nome. O seguinte é um exemplo de saída.

```
{
    "PolicyARN": "arn:aws:autoscaling:region:account-
id:scalingPolicy:policy-id:resource/ec2/spot-fleet-request/sfr-73fbd2ce-
aa30-494c-8788-1cee4EXAMPLE:policyName/cpu50-target-tracking-scaling-policy",
    "Alarms": [
        {
            "AlarmARN": "arn:aws:cloudwatch:region:account-id:alarm:TargetTracking-
spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE-AlarmHigh-d4f0770c-
b46e-434a-a60f-3b36d653feca",
            "AlarmName": "TargetTracking-spot-fleet-request/sfr-73fbd2ce-
aa30-494c-8788-1cee4EXAMPLE-AlarmHigh-d4f0770c-b46e-434a-a60f-3b36d653feca"
        },
        {
            "AlarmARN": "arn:aws:cloudwatch:region:account-id:alarm:TargetTracking-
spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE-AlarmLow-1b437334-
d19b-4a63-a812-6c67aaf2910d",
            "AlarmName": "TargetTracking-spot-fleet-request/sfr-73fbd2ce-
aa30-494c-8788-1cee4EXAMPLE-AlarmLow-1b437334-d19b-4a63-a812-6c67aaf2910d"
        }
    ]
}
```

Etapa 3: descrever as políticas de escalabilidade com rastreamento de destino

Você pode descrever todas as políticas de dimensionamento para o namespace de serviço especificado usando o seguinte comando describe-scaling-policies.

```
aws application-autoscaling describe-scaling-policies --service-namespace ec2
```

Você pode filtrar os resultados apenas para as políticas de escalabilidade de rastreamento de destino usando o parâmetro `--query`. Para mais informações sobre a sintaxe de `query`, consulte Controlar a saída do comando da AWS CLI no Manual do usuário da AWS Command Line Interface .

Linux, macOS ou Unix

```
aws application-autoscaling describe-scaling-policies --service-namespace ec2 \
  --query 'ScalingPolicies[?PolicyType==`TargetTrackingScaling`]'
```

Windows

```
aws application-autoscaling describe-scaling-policies --service-namespace ec2 ^
  --query "ScalingPolicies[?PolicyType==`TargetTrackingScaling`]"
```

Saída

O seguinte é um exemplo de saída.

```
[
    {
        "PolicyARN": "PolicyARN",
        "TargetTrackingScalingPolicyConfiguration": {
            "PredefinedMetricSpecification": {
                "PredefinedMetricType": "EC2SpotFleetRequestAverageCPUUtilization"
            },
            "TargetValue": 50.0
        },
        "PolicyName": "cpu50-target-tracking-scaling-policy",
        "ScalableDimension": "ec2:spot-fleet-request:TargetCapacity",
        "ServiceNamespace": "ec2",
        "PolicyType": "TargetTrackingScaling",
        "ResourceId": "spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
```

```
        "Alarms": [
            {
                "AlarmARN": "arn:aws:cloudwatch:region:account-id:alarm:TargetTracking-
spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE-AlarmHigh-d4f0770c-
b46e-434a-a60f-3b36d653feca",
                "AlarmName": "TargetTracking-spot-fleet-request/sfr-73fbd2ce-
aa30-494c-8788-1cee4EXAMPLE-AlarmHigh-d4f0770c-b46e-434a-a60f-3b36d653feca"
            },
            {
                "AlarmARN": "arn:aws:cloudwatch:region:account-id:alarm:TargetTracking-
spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE-AlarmLow-1b437334-
d19b-4a63-a812-6c67aaf2910d",
                "AlarmName": "TargetTracking-spot-fleet-request/sfr-73fbd2ce-
aa30-494c-8788-1cee4EXAMPLE-AlarmLow-1b437334-d19b-4a63-a812-6c67aaf2910d"
            }
        ],
        "CreationTime": 1515021724.807
    }
]
```

Exclua uma política de escalabilidade de rastreamento de destino para o Application Auto Scaling usando o AWS CLI

Ao finalizar uma política de dimensionamento de rastreamento de destino, você pode excluí-la usando o comando delete-scaling-policy.

O comando a seguir exclui a política de dimensionamento de rastreamento de destino que você especificou para a solicitação especificada da frota spot. Ele também exclui os CloudWatch alarmes que o Application Auto Scaling criou em seu nome.

Linux, macOS ou Unix

```
aws application-autoscaling delete-scaling-policy --service-namespace ec2 \
  --scalable-dimension ec2:spot-fleet-request:TargetCapacity \
  --resource-id spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
  --policy-name cpu50-target-tracking-scaling-policy
```

Windows

```
aws application-autoscaling delete-scaling-policy --service-namespace ec2 ^
```

```
  --scalable-dimension ec2:spot-fleet-request:TargetCapacity ^
  --resource-id spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE ^
  --policy-name cpu50-target-tracking-scaling-policy
```

Crie uma política de escalabilidade de rastreamento de destino para o Application Auto Scaling usando matemática em métricas

Usando a matemática métrica, você pode consultar várias CloudWatch métricas e usar expressões matemáticas para criar novas séries temporais com base nessas métricas. Você pode visualizar as séries temporais resultantes no console do CloudWatch e adicioná-las aos painéis. Para obter mais informações sobre matemática métrica, consulte Usando matemática métrica no Guia CloudWatch do usuário da Amazon.

As considerações a seguir se aplicam a expressões matemática em métricas:

- Você pode consultar qualquer CloudWatch métrica disponível. Cada métrica corresponde a uma combinação exclusiva de nome de métrica, espaço nominal e zero ou mais dimensões.
- Você pode usar qualquer operador aritmético (+ - */^), função estatística (como AVG ou SUM) ou outra função compatível. CloudWatch
- Você pode usar as métricas e os resultados de outras expressões matemáticas nas fórmulas da expressão matemática.
- Qualquer expressão usada em uma especificação de métrica deve eventualmente retornar uma única série temporal.
- Você pode verificar se uma expressão matemática métrica é válida usando o CloudWatch console ou a CloudWatch GetMetricDataAPI.

Tópicos

- Exemplo: lista de pendências da fila do Amazon SQS por tarefa
- Limitações

Exemplo: lista de pendências da fila do Amazon SQS por tarefa

Para calcular a lista de pendências da fila do Amazon SQS por tarefa, use o número aproximado de mensagens disponíveis para recuperação da fila e divida esse número pelo número de tarefas

do Amazon ECS em execução no serviço. Para obter mais informações, consulte Amazon Elastic Container Service (ECS) Auto Scaling usando métricas personalizadas AWS no blog de computação.

A lógica da expressão é a seguinte:

```
sum of (number of messages in the queue)/(number of tasks that are
currently in the RUNNING state)
```

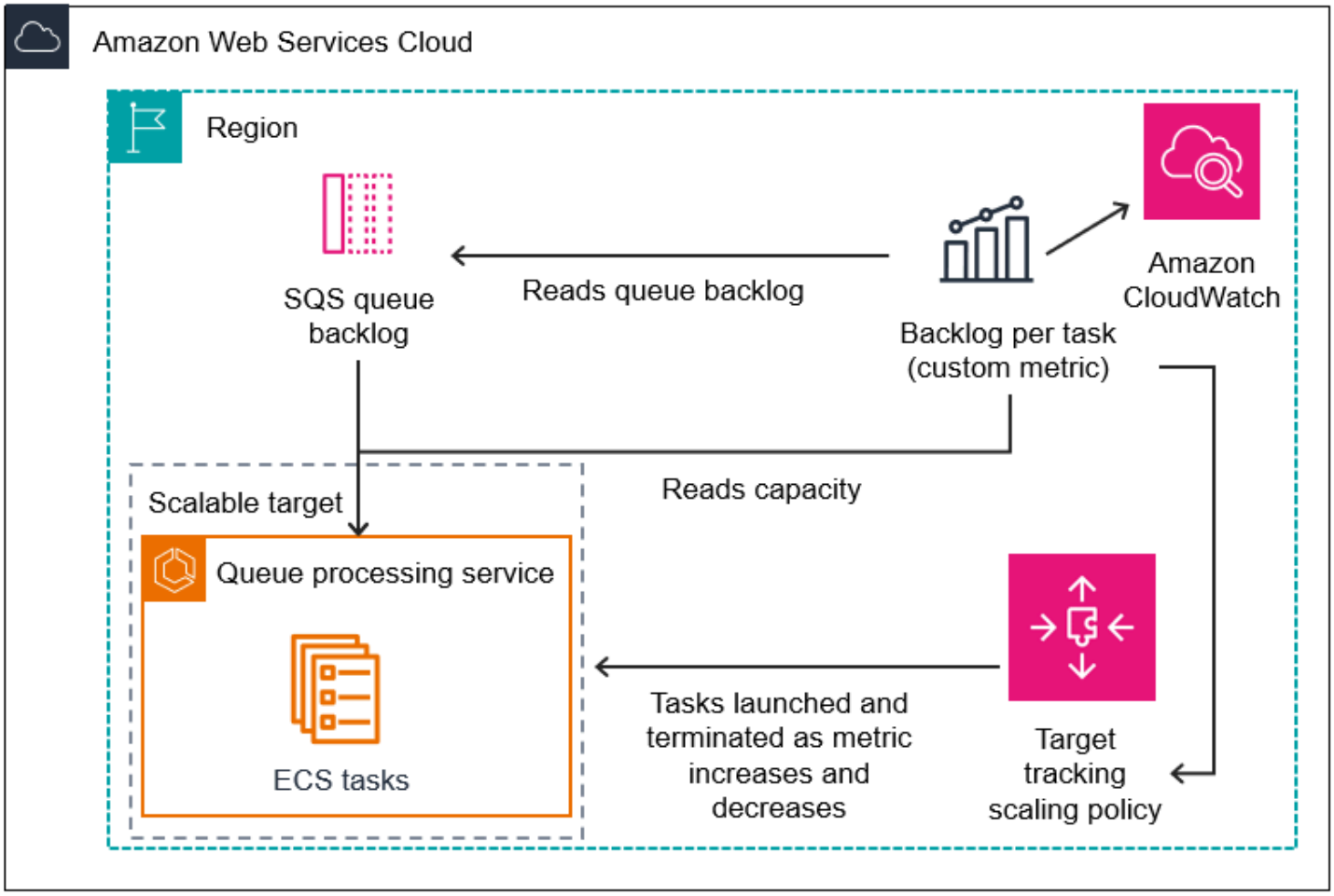
Então, suas informações CloudWatch métricas são as seguintes.

ID	CloudWatch métrica	Estatística	Período
m1	ApproximateNumberOfMessagesVisible	Soma	1 minuto
m2	RunningTaskCount	Média	1 minuto

O ID e a expressão matemáticos da métrica são os seguintes:

ID	Expressão
e1	`(m1)/(m2)`

O seguinte diagrama ilustra a arquitetura dessa métrica:



Para usar essa matemática em métricas na criação de uma política de escalabilidade com monitoramento de destino (AWS CLI)

1. Armazene a expressão matemática em métricas como parte de uma especificação de métrica personalizada em um arquivo JSON denominado `config.json`.

   Use o exemplo a seguir como auxílio para começar. Substitua cada *`user input placeholder`* por suas próprias informações.

```
{
    "CustomizedMetricSpecification": {
        "Metrics": [
            {
                "Label": "Get the queue size (the number of messages waiting to be
 processed)",
                "Id": "m1",
                "MetricStat": {
```

```
                    "Metric": {
                        "MetricName": "ApproximateNumberOfMessagesVisible",
                        "Namespace": "AWS/SQS",
                        "Dimensions": [
                            {
                                "Name": "QueueName",
                                "Value": "my-queue"
                            }
                        ]
                    },
                    "Stat": "Sum"
                },
                "ReturnData": false
            },
            {
                "Label": "Get the ECS running task count (the number of currently
running tasks)",
                "Id": "m2",
                "MetricStat": {
                    "Metric": {
                        "MetricName": "RunningTaskCount",
                        "Namespace": "ECS/ContainerInsights",
                        "Dimensions": [
                            {
                                "Name": "ClusterName",
                                "Value": "my-cluster"
                            },
                            {
                                "Name": "ServiceName",
                                "Value": "my-service"
                            }
                        ]
                    },
                    "Stat": "Average"
                },
                "ReturnData": false
            },
            {
                "Label": "Calculate the backlog per instance",
                "Id": "e1",
                "Expression": "m1 / m2",
                "ReturnData": true
            }
        ]
```

```
    },
    "TargetValue": 100
}
```

Para obter mais informações, consulte a Referência TargetTrackingScalingPolicyConfigurationda API Application Auto Scaling.

> **Note**
>
> Veja a seguir alguns recursos adicionais que podem ajudar você a encontrar nomes de métricas, namespaces, dimensões e estatísticas para CloudWatch métricas:
>
> - Para obter informações sobre as métricas disponíveis para AWS serviços, consulte AWS serviços que publicam CloudWatch métricas no Guia CloudWatch do usuário da Amazon.
> - Para obter o nome exato da métrica, o namespace e as dimensões (se aplicável) de uma CloudWatch métrica com o AWS CLI, consulte list-metrics.

2. Para criar essa política, execute o put-scaling-policycomando usando o arquivo JSON como entrada, conforme demonstrado no exemplo a seguir.

```
aws application-autoscaling put-scaling-policy --policy-name sqs-backlog-target-tracking-scaling-policy \
  --service-namespace ecs --scalable-dimension ecs:service:DesiredCount --resource-id service/my-cluster/my-service \
  --policy-type TargetTrackingScaling --target-tracking-scaling-policy-configuration file://config.json
```

Se for bem-sucedido, esse comando retornará o Amazon Resource Name (ARN) da política e o ARNs dos dois CloudWatch alarmes criados em seu nome.

```
{
    "PolicyARN": "arn:aws:autoscaling:us-west-2:012345678910:scalingPolicy:
 8784a896-b2ba-47a1-b08c-27301cc499a1:resource/ecs/service/my-cluster/my-service:policyName/sqs-backlog-target-tracking-scaling-policy",
    "Alarms": [
        {
            "AlarmARN": "arn:aws:cloudwatch:us-west-2:012345678910:alarm:TargetTracking-service/my-cluster/my-service-AlarmHigh-9bc77b56-0571-4276-ba0f-d4178882e0a0",
```

```
            "AlarmName": "TargetTracking-service/my-cluster/my-service-
AlarmHigh-9bc77b56-0571-4276-ba0f-d4178882e0a0"
        },
        {
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:TargetTracking-service/my-cluster/my-service-
AlarmLow-9b6ad934-6d37-438e-9e05-02836ddcbdc4",
            "AlarmName": "TargetTracking-service/my-cluster/my-service-
AlarmLow-9b6ad934-6d37-438e-9e05-02836ddcbdc4"
        }
    ]
}
```

> **Note**
>
> Se esse comando gerar um erro, verifique se você atualizou o AWS CLI localmente para a versão mais recente.

Limitações

- O tamanho máximo da solicitação é de 50 KB. Esse é o tamanho total da carga útil da solicitação de PutScalingPolicyAPI quando você usa matemática métrica na definição da política. Se você exceder esse limite, o Application Auto Scaling rejeitará a solicitação.
- Os seguintes serviços não têm suporte ao usar a matemática em métricas com políticas de escalabilidade de rastreamento de destino:
  - Amazon Keyspaces (para Apache Cassandra)
  - DynamoDB
  - Amazon EMR
  - Amazon MSK
  - Amazon Neptune

Políticas de escalabilidade em etapas para o Application Auto Scaling

Uma política de escalabilidade por etapas dimensiona a capacidade do seu aplicativo em incrementos predefinidos com base em alarmes. CloudWatch É possível definir políticas de escalabilidade separadas para lidar com o aumento horizontal da escala (aumento da capacidade) e com a redução horizontal da escala (diminuição da capacidade) quando um limite de alarme é violado.

Com as políticas de escalabilidade por etapas, você cria e gerencia os CloudWatch alarmes que invocam o processo de escalabilidade. Quando um alarme é violado, o Application Auto Scaling inicia a política de escalabilidade associada a esse alarme.

A política de escalabilidade em etapas escala a capacidade usando um conjunto de ajustes, conhecidos como ajustes de etapas. A dimensão dos ajustes varia de acordo com a magnitude da violação do alarme.

- Se a violação exceder o primeiro limite, o Application Auto Scaling aplicará o primeiro ajuste de etapa.
- Se a violação exceder o segundo limite, o Application Auto Scaling aplicará o segundo ajuste de etapa, e assim por diante.

Isso permite que a política de escalabilidade responda adequadamente a alterações menores e maiores na métrica de alarme.

A política continuará a responder a violações de alarmes adicionais, mesmo enquanto uma atividade de escalabilidade estiver em andamento. Isso significa que o Application Auto Scaling avaliará todas as violações de alarmes à medida que ocorrerem. Um período de esfriamento é usado para obter proteção contra a escalabilidade excessiva devido a múltiplas violações de alarmes que ocorrem em rápida sucessão.

De forma semelhante ao rastreamento de destinos, a escalabilidade em etapas pode ajudar a escalar automaticamente a capacidade da aplicação à medida que ocorrem alterações no tráfego. No entanto, as políticas de rastreamento de destinos tendem a ser mais fáceis de implementar e gerenciar para necessidades constantes de escalabilidade.

Destinos escaláveis compatíveis

É possível usar políticas de escalabilidade em etapas com os seguintes destinos escaláveis:

- AppStream 2.0 frotas
- clusters de bancos de dados Aurora
- serviços da ECS
- Clusters do EMR
- SageMaker Variantes de endpoint de IA
- SageMaker Componentes de inferência de IA
- SageMaker Concorrência provisionada por IA sem servidor
- Spot Fleets
- Recursos personalizados

Conteúdo

- Como funciona a escalabilidade em etapas para o Application Auto Scaling
- Crie uma política de escalabilidade de etapas para o Application Auto Scaling usando o AWS CLI
- Descreva as políticas de escalabilidade de etapas para Application Auto Scaling usando o AWS CLI
- Exclua uma política de escalabilidade de etapas para o Application Auto Scaling usando o AWS CLI

Como funciona a escalabilidade em etapas para o Application Auto Scaling

Este tópico descreve como a escalabilidade em etapas funciona e apresenta os principais elementos de uma política de escalabilidade em etapas.

Conteúdo

- Como funcionam
- Ajustes em etapas
- Tipos de ajuste da escalabilidade
- Período de espera

- Comandos normalmente usados para criação, exclusão e gerenciamento de política de escalabilidade
- Considerações
- Recursos relacionados
- Acesso ao console

Como funcionam

Para usar o escalonamento por etapas, você cria um CloudWatch alarme que monitora uma métrica para sua meta escalável. Defina a métrica, o valor limite e o número de períodos de avaliação que determinam uma violação de alarme. Além disso, você cria uma política de escalabilidade em etapas que define como escalar a capacidade quando o limite de alarme é violado e associá-la ao seu destino escalável.

Adicione os ajustes de etapas na política. É possível definir diferentes ajustes de etapas com base na dimensão da violação do alarme. Por exemplo:

- Aumentar a escala horizontalmente em 10 unidades de capacidade, se a métrica de alarme atingir 60%.
- Aumentar a escala horizontalmente em 30 unidades de capacidade, se a métrica de alarme atingir 75%.
- Aumentar a escala horizontalmente em 40 unidades de capacidade, se a métrica de alarme atingir 85%.

Quando o limite de alarme for violado durante o número especificado de períodos de avaliação, o Application Auto Scaling aplicará os ajustes de etapas definidos na política. Os ajustes podem continuar para violações de alarmes adicionais até que o estado do alarme retorne a OK.

As atividades de escalabilidade são executadas com períodos de esfriamento entre elas para evitar flutuações rápidas na capacidade. Opcionalmente, é possível configurar os períodos de esfriamento para a política de escalabilidade.

Ajustes em etapas

Ao criar uma política de escalabilidade em etapas, especifique um ou mais ajustes de etapa que ajustarão automaticamente a escala da capacidade do destino de maneira dinâmica com base no tamanho da violação do alarme. Cada ajuste em etapas especifica o seguinte:

- Um limite inferior para o valor da métrica
- Um limite superior para o valor da métrica
- O valor de acordo com o qual dimensionar com base no tipo de ajuste de dimensionamento

CloudWatch agrega pontos de dados métricos com base na estatística da métrica associada ao seu CloudWatch alarme. Quando o alarme é violado, a política de dimensionamento apropriada é invocada. O Application Auto Scaling aplica seu tipo de agregação especificado aos pontos de dados métricos mais recentes de CloudWatch (em oposição aos dados métricos brutos). Ele compara esse valor de métrica agregada com os limites superior e inferior definidos pelo ajustes em etapa para determinar qual deles deve ser executado.

Você especifica os limites superior e inferior em relação ao limite de ruptura. Por exemplo, digamos que você tenha criado um CloudWatch alarme e uma política de expansão para quando a métrica estiver acima de 50%. Em seguida, você criou um segundo alarme e uma política para reduzir a escala horizontalmente em momentos em que a métrica está abaixo de 50%. Você definiu um conjunto de ajustes de etapas com um tipo de ajuste `PercentChangeInCapacity` para cada política:

Exemplo: ajustes em etapas para política de expansão

Limite inferior	Limite superior	Ajuste
0	10	0
10	20	10
20	nulo	30

Exemplo: ajustes em etapas para política de redução

Limite inferior	Limite superior	Ajuste
-10	0	0
-20	-10	-10
nulo	-20	-30

Isso cria a seguinte configuração de escalabilidade.

```
Metric value

-infinity          30%    40%          60%     70%             infinity
-------------------------------------------------------------------------
          -30%      | -10% | Unchanged  | +10%  |       +30%
-------------------------------------------------------------------------
```

Agora, suponhamos que você use essa configuração de escalabilidade em um destino escalável com uma capacidade de 10. Os pontos a seguir resumem o comportamento da configuração de escalabilidade em relação à capacidade do destino escalável:

- A capacidade original será mantida enquanto o valor agregado da métrica for maior que 40 e menor que 60.
- Se o valor da métrica chegar a 60, o Application Auto Scaling aumentará a capacidade do destino escalável em 1, totalizando 11. Isso é com base no segundo ajuste em etapas da política de expansão (adicionar 10% de 10). Depois de adicionar a nova capacidade, o Application Auto Scaling aumentará a capacidade atual para 11. Se o valor da métrica aumentar para 70 mesmo depois desse aumento da capacidade, o Application Auto Scaling aumentará a capacidade de destino em 3, totalizando 14. Isso é com base no terceiro ajuste em etapas da política de expansão (adicionar 30% de 11, 3,3, arredondado para 3).
- Se o valor da métrica chegar a 40, o Application Auto Scaling diminuirá a capacidade do destino escalável em 1, para 13, com base na segunda etapa de ajuste da política de redução da escala na horizontal (remoção 10% de 14; ou seja 1,4 arredondado para 1). Se o valor da métrica cair para 30 mesmo após essa redução de capacidade, o Application Auto Scaling diminuirá a capacidade do destino em 3, para 10, com base no ajuste da terceira etapa da política de redução da escala na horizontal (remover 30% de 13, 3,9, arredondado para baixo, ou seja, para 3).

Ao especificar os ajustes em etapas para sua política de escalabilidade, observe o seguinte:

- Os intervalos de seus ajustes em etapas não podem se sobrepor ou ter uma lacuna.
- Somente um ajuste em etapas pode ter um limite inferior nulo (infinito negativo). Se um ajuste em etapas tiver um limite inferior negativo, não deverá haver um ajuste em etapas com um limite inferior nulo.

- Somente um ajuste em etapas pode ter um limite superior nulo (infinito positivo). Se um ajuste em etapas tiver um limite superior positivo, deverá haver um ajuste em etapas com um limite superior nulo.
- Os limites inferior e superior não podem ser nulos no mesmo ajuste em etapas.
- Se o valor da métrica estiver acima do limite de violação, o limite inferior será inclusivo e o limite superior será exclusivo. Se o valor da métrica estiver abaixo do limite de violação, o limite inferior será exclusivo e o limite superior será inclusivo.

Tipos de ajuste da escalabilidade

É possível definir uma política de escalabilidade que execute a ação de escalabilidade ideal, com base no tipo de ajuste de escalabilidade escolhido. É possível especificar o tipo de ajuste como uma porcentagem da capacidade atual do seu alvo escalável ou em números absolutos.

O Application Auto Scaling oferece suporte aos seguintes tipos de políticas de escalabilidade em etapas:

- ChangeInCapacity—Aumente ou diminua a capacidade atual da meta escalável de acordo com o valor especificado. Um valor positivo aumenta a capacidade e um valor negativo diminui a capacidade. Por exemplo: se a capacidade atual for de 3 e o ajuste for 5, o Application Auto Scaling adicionará 5 à capacidade, totalizando 8.
- ExactCapacity—Altere a capacidade atual do alvo escalável para o valor especificado. Especifique um valor não negativo com esse tipo de ajuste. Por exemplo: se a capacidade atual for de 3 e o ajuste for 5, o Application Auto Scaling alterará a capacidade para 5.
- PercentChangeInCapacity—Aumente ou diminua a capacidade atual da meta escalável na porcentagem especificada. Um valor positivo aumenta a capacidade e um valor negativo diminui a capacidade. Por exemplo: se a capacidade atual for de 10 e o ajuste for 10%, o Application Auto Scaling adicionará 1 à capacidade, totalizando 11.

  Se o valor resultante não for um inteiro, o Application Auto Scaling arredondará da seguinte forma:

  - Valores maiores que 1 serão arredondados para baixo. Por exemplo, `12.7` será arredondado para `12`.
  - Os valores entre 0 e 1 serão arredondados para 1. Por exemplo, `.67` será arredondado para `1`.
  - Os valores entre 0 e -1 serão arredondados para -1. Por exemplo, `-.58` será arredondado para `-1`.

- Os valores menores que -1 serão arredondado para cima. Por exemplo, `-6.67` será arredondado para `-6`.

Com PercentChangeInCapacity, você também pode especificar o valor mínimo a ser escalado usando o `MinAdjustmentMagnitude` parâmetro. Por exemplo, suponha que você crie uma política que adiciona 25% e especifique, no mínimo, 2. Se o destino escalável tiver uma capacidade de 4 e a política de escalabilidade for realizada, 25% de 4 é 1. No entanto, como você especificou um incremento mínimo de 2, o Application Auto Scaling adicionará 2.

Período de espera

Opcionalmente, você pode definir um período de esfriamento na política de escalação em etapas.

O período de esfriamento especifica quanto tempo a política de escalação espera até uma atividade anterior de escalação ter efeito.

Há duas maneiras de planejar o uso de períodos de esfriamento para uma configuração de escalação em etapas:

- Com o período de esfriamento para políticas de aumento de escala horizontal, a intenção é aumentar a escala horizontalmente de modo contínuo (mas não excessivo). Depois que o Application Auto Scaling aumenta a escala horizontalmente com êxito usando uma política de escalação em etapas, ele começa a calcular o tempo de esfriamento. A política de escalação não aumentará a capacidade desejada novamente a menos que um aumento maior da escala horizontal seja disparado ou que o período de esfriamento termine. Enquanto o período de desaquecimento após expansão estiver em vigor, a capacidade adicionada pela ação de expansão de início será calculada como parte da capacidade desejada para a próxima ação de expansão.
- Com o período de esfriamento para políticas de redução de escala horizontal, a intenção é reduzir de maneira conservadora para proteger a disponibilidade da aplicação, de modo que as ações de redução de escala horizontal fiquem bloqueadas até o período de esfriamento expirar. No entanto, se outro alarme acionar uma ação de ampliação durante o período de desaquecimento da redução da escala, o Application Auto Scaling expandirá o destino imediatamente. Nesse caso, o período de esfriamento da redução da escala horizontal é interrompido e não é concluído.

Por exemplo, quando ocorre um pico de tráfego, um alarme é disparado e o Application Auto Scaling automaticamente adiciona capacidade para ajudar a lidar com o aumento da carga. Se você definir um período de esfriamento para a política de aumento de escala horizontal, quando o alarme acionar

a política para aumentar a capacidade em 2, a ação de escalação será concluída com sucesso e o período de esfriamento do aumento da escala horizontal será iniciado. Se o alarme disparar novamente durante período de esfriamento, mas com um ajuste em etapas mais agressivo de 3, o aumento de 2 anterior será considerado parte da capacidade atual. Portanto, apenas 1 será adicionado à capacidade. Isso permite uma escalação mais rápida do que esperar a expiração do esfriamento, mas sem adicionar mais capacidade do que o necessário.

O período de desaquecimento é medido em segundos e se aplica somente a ações de escalabilidade relacionadas à política. Durante um período de desaquecimento, quando uma ação programada começa no horário programado, ela pode acionar uma ação de escalabilidade imediatamente, sem esperar que o período de desaquecimento expire.

O valor padrão é 300 se nenhum valor for especificado.

Comandos normalmente usados para criação, exclusão e gerenciamento de política de escalabilidade

Os comandos comumente usados para trabalhar com políticas de escalabilidade incluem:

- register-scalable-targetregistrar AWS ou personalizar recursos como alvos escaláveis (um recurso que o Application Auto Scaling pode escalar) e suspender e retomar o escalonamento.
- put-scaling-policypara adicionar ou modificar políticas de escalabilidade para um alvo escalável existente.
- describe-scaling-activitiespara retornar informações sobre atividades de escalabilidade em uma AWS região.
- describe-scaling-policiespara retornar informações sobre políticas de escalabilidade em uma AWS região.
- delete-scaling-policypara excluir uma política de escalabilidade.

Considerações

As considerações a seguir são aplicáveis ao trabalhar com políticas de escalabilidade em etapas:

- Avalie se é possível prever os ajustes em etapas na aplicação com precisão suficiente para usar a escalabilidade em etapas. Se a métrica de escalabilidade aumentar ou diminuir proporcionalmente à capacidade do destino dimensionável, recomendamos que você use uma política de escalabilidade de rastreamento do objetivo. Você ainda tem a opção de usar a escalabilidade em

etapas como política adicional para uma configuração mais avançada. Por exemplo, é possível configurar uma resposta mais agressiva quando a utilização atinge determinado nível.

- Para evitar oscilações, certifique-se de escolher uma margem adequada entre os limites de redução e aumento da escala. Oscilação é um ciclo infinito de aumento e redução de escala horizontal. Ou seja, se o sistema adotar alguma ação de escalabilidade, o valor da métrica mudaria e iniciaria outra ação de escalabilidade na direção inversa.

Recursos relacionados

Para obter informações sobre a criação de políticas de escalabilidade por etapas para grupos de Auto Scaling, consulte Políticas de escalabilidade por etapas e simples para o Amazon Auto EC2 Scaling no Guia do usuário do Amazon Auto EC2 Scaling.

Acesso ao console

O acesso ao console para visualizar, adicionar, atualizar ou remover políticas de escalabilidade em etapas nos recursos escaláveis depende do recurso utilizado. Para obter mais informações, consulte Serviços da AWS que você pode usar com o Application Auto Scaling.

Crie uma política de escalabilidade de etapas para o Application Auto Scaling usando o AWS CLI

Este exemplo usa AWS CLI comandos para criar uma política de escalabilidade por etapas para um serviço do Amazon ECS. Para usar um destino escalável diferente, especifique o respectivo namespace em `--service-namespace`, a dimensão escalável em `--scalable-dimension` e o ID do recurso em `--resource-id`.

Ao usar o AWS CLI, lembre-se de que seus comandos são Região da AWS executados no configurado para o seu perfil. Se você deseja executar os comandos em uma região diferente, altere a região padrão para o seu perfil ou use o parâmetro `--region` com o comando.

Tarefas

- Etapa 1: registrar um destino escalável
- Etapa 2: criar uma política de escalabilidade em etapas
- Etapa 3: criar um alarme que invoca uma política de escalabilidade

Etapa 1: registrar um destino escalável

Se você ainda não tiver feito isso, inscreva o destino escalável. Use o register-scalable-targetcomando para registrar um recurso específico no serviço de destino como um alvo escalável. O exemplo a seguir inscreve um serviço do Amazon ECS com o Application Auto Scaling. O Application Auto Scaling pode escalar o número de tarefas em um mínimo de duas tarefas e um máximo de dez. Substitua cada *user input placeholder* por suas próprias informações.

Linux, macOS ou Unix

```
aws application-autoscaling register-scalable-target --service-namespace ecs \
  --scalable-dimension ecs:service:DesiredCount \
  --resource-id service/my-cluster/my-service \
  --min-capacity 2 --max-capacity 10
```

Windows

```
aws application-autoscaling register-scalable-target --service-namespace ecs ^
  --scalable-dimension ecs:service:DesiredCount ^
  --resource-id service/my-cluster/my-service ^
  --min-capacity 2 --max-capacity 10
```

Saída

Se obtiver êxito, esse comando retornará o ARN do destino escalável. O seguinte é um exemplo de saída.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Etapa 2: criar uma política de escalabilidade em etapas

Para criar uma política de escalabilidade em etapas para um destino escalável, você pode usar os exemplos a seguir para começar.

Scale out

Como criar uma política de escalabilidade em etapas para aumentar a escala horizontalmente (aumentar a capacidade)

1. Use o comando `cat` a seguir para especificar uma configuração de política de escalabilidade em etapas em um arquivo JSON chamado `config.json` em seu diretório inicial. Veja a seguir um exemplo de configuração com um tipo de ajuste `PercentChangeInCapacity` que aumenta a capacidade do alvo escalável com base nos seguintes ajustes de etapa (assumindo um limite de CloudWatch alarme de 70):

   - Aumentar a capacidade em 10% quando o valor da métrica for maior ou igual a 70, mas menor que 85.
   - Aumentar a capacidade em 20% quando o valor da métrica for maior ou igual a 85, mas menor que 95.
   - Aumentar a capacidade em 30% quando o valor da métrica for maior ou igual a 95.

```
$ cat ~/config.json
{
  "AdjustmentType": "PercentChangeInCapacity",
  "MetricAggregationType": "Average",
  "Cooldown": 60,
  "MinAdjustmentMagnitude": 1,
  "StepAdjustments": [
    {
      "MetricIntervalLowerBound": 0.0,
      "MetricIntervalUpperBound": 15.0,
      "ScalingAdjustment": 10
    },
    {
      "MetricIntervalLowerBound": 15.0,
      "MetricIntervalUpperBound": 25.0,
      "ScalingAdjustment": 20
    },
    {
      "MetricIntervalLowerBound": 25.0,
      "ScalingAdjustment": 30
    }
  ]
}
```

Para obter mais informações, consulte a Referência StepScalingPolicyConfigurationda API Application Auto Scaling.

2. Use o put-scaling-policycomando a seguir, junto com o `config.json` arquivo que você criou, para criar uma política de escalabilidade chamada`my-step-scaling-policy`.

   Linux, macOS ou Unix

```
aws application-autoscaling put-scaling-policy --service-namespace ecs \
  --scalable-dimension ecs:service:DesiredCount \
  --resource-id service/my-cluster/my-service \
  --policy-name my-step-scaling-policy --policy-type StepScaling \
  --step-scaling-policy-configuration file://config.json
```

   Windows

```
aws application-autoscaling put-scaling-policy --service-namespace ecs ^
  --scalable-dimension ecs:service:DesiredCount ^
  --resource-id service/my-cluster/my-service ^
  --policy-name my-step-scaling-policy --policy-type StepScaling ^
  --step-scaling-policy-configuration file://config.json
```

   Saída

   O resultado inclui o ARN que serve como um nome exclusivo para a política. Você precisa dele para criar um CloudWatch alarme para sua política. O seguinte é um exemplo de saída.

```
{
    "PolicyARN":
 "arn:aws:autoscaling:region:123456789012:scalingPolicy:ac542982-cbeb-4294-891c-a5a941dfa787:resource/ecs/service/my-cluster/my-service:policyName/my-step-scaling-policy"
}
```

Scale in

Como criar uma política de escalabilidade em etapas para reduzir a escala horizontalmente (diminuir a capacidade)

1. Use o comando `cat` a seguir para especificar uma configuração de política de escalabilidade em etapas em um arquivo JSON chamado `config.json` em seu diretório inicial. Veja a seguir um exemplo de configuração com um tipo de ajuste `ChangeInCapacity` que diminui a capacidade do alvo escalável com base nos seguintes ajustes de etapa (assumindo um limite de CloudWatch alarme de 50):

   - Diminuir a capacidade em 1 quando o valor da métrica é menor ou igual a 50, mas maior que 40.
   - Diminuir a capacidade em 2 quando o valor da métrica é menor ou igual a 40, mas maior que 30.
   - Diminuir a capacidade em 3 quando o valor da métrica é menor ou igual a 30.

```
$ cat ~/config.json
{
  "AdjustmentType": "ChangeInCapacity",
  "MetricAggregationType": "Average",
  "Cooldown": 60,
  "StepAdjustments": [
    {
      "MetricIntervalUpperBound": 0.0,
      "MetricIntervalLowerBound": -10.0,
      "ScalingAdjustment": -1
    },
    {
      "MetricIntervalUpperBound": -10.0,
      "MetricIntervalLowerBound": -20.0,
      "ScalingAdjustment": -2
    },
    {
      "MetricIntervalUpperBound": -20.0,
      "ScalingAdjustment": -3
    }
  ]
}
```

Para obter mais informações, consulte a Referência StepScalingPolicyConfigurationda API Application Auto Scaling.

2. Use o put-scaling-policycomando a seguir, junto com o `config.json` arquivo que você criou, para criar uma política de escalabilidade chamada`my-step-scaling-policy`.

   Linux, macOS ou Unix

```
aws application-autoscaling put-scaling-policy --service-namespace ecs \
  --scalable-dimension ecs:service:DesiredCount \
  --resource-id service/my-cluster/my-service \
  --policy-name my-step-scaling-policy --policy-type StepScaling \
  --step-scaling-policy-configuration file://config.json
```

   Windows

```
aws application-autoscaling put-scaling-policy --service-namespace ecs ^
  --scalable-dimension ecs:service:DesiredCount ^
  --resource-id service/my-cluster/my-service ^
  --policy-name my-step-scaling-policy --policy-type StepScaling ^
  --step-scaling-policy-configuration file://config.json
```

   Saída

   O resultado inclui o ARN que serve como um nome exclusivo para a política. Você precisa desse ARN para criar um CloudWatch alarme para sua política. O seguinte é um exemplo de saída.

```
{
    "PolicyARN":
 "arn:aws:autoscaling:region:123456789012:scalingPolicy:ac542982-cbeb-4294-891c-a5a941dfa787:resource/ecs/service/my-cluster/my-service:policyName/my-step-scaling-policy"
}
```

Etapa 3: criar um alarme que invoca uma política de escalabilidade

Por fim, use o CloudWatch put-metric-alarmcomando a seguir para criar um alarme para usar com sua política de escalabilidade de etapas. Neste exemplo, você tem um alarme com base na utilização

média da CPU. O alarme é configurado para entrar em um estado de ALARME se atingir o limite de 70% por, no mínimo, dois períodos de avaliação consecutivos de 60 segundos. Para especificar uma CloudWatch métrica diferente ou usar sua própria métrica personalizada, especifique seu nome em `--metric-name` e seu namespace em. `--namespace`

Linux, macOS ou Unix

```
aws cloudwatch put-metric-alarm --alarm-name Step-Scaling-AlarmHigh-ECS:service/my-cluster/my-service \
  --metric-name CPUUtilization --namespace AWS/ECS --statistic Average \
  --period 60 --evaluation-periods 2 --threshold 70 \
  --comparison-operator GreaterThanOrEqualToThreshold \
  --dimensions Name=ClusterName,Value=default Name=ServiceName,Value=sample-app-service \
  --alarm-actions PolicyARN
```

Windows

```
aws cloudwatch put-metric-alarm --alarm-name Step-Scaling-AlarmHigh-ECS:service/my-cluster/my-service ^
  --metric-name CPUUtilization --namespace AWS/ECS --statistic Average ^
  --period 60 --evaluation-periods 2 --threshold 70 ^
  --comparison-operator GreaterThanOrEqualToThreshold ^
  --dimensions Name=ClusterName,Value=default Name=ServiceName,Value=sample-app-service ^
  --alarm-actions PolicyARN
```

Descreva as políticas de escalabilidade de etapas para Application Auto Scaling usando o AWS CLI

Você pode descrever todas as políticas de escalabilidade para um namespace de serviço usando o comando. describe-scaling-policies O exemplo a seguir descreve todas as políticas de escalabilidade para todos os serviços do Amazon ECS. Para listá-las para um serviço específico do Amazon ECS, adicione apenas a opção `--resource-id`.

```
aws application-autoscaling describe-scaling-policies --service-namespace ecs
```

Você pode filtrar os resultados apenas para as políticas de escalabilidade em etapas usando o parâmetro `--query`. Para mais informações sobre a sintaxe de `query`, consulte Controlar a saída do comando da AWS CLI no Manual do usuário da AWS Command Line Interface .

Linux, macOS ou Unix

```
aws application-autoscaling describe-scaling-policies --service-namespace ecs \
  --query 'ScalingPolicies[?PolicyType==`StepScaling`]'
```

Windows

```
aws application-autoscaling describe-scaling-policies --service-namespace ecs ^
  --query "ScalingPolicies[?PolicyType==`StepScaling`]"
```

Saída

O seguinte é um exemplo de saída.

```
[
    {
        "PolicyARN": "PolicyARN",
        "StepScalingPolicyConfiguration": {
            "MetricAggregationType": "Average",
            "Cooldown": 60,
            "StepAdjustments": [
                {
                    "MetricIntervalLowerBound": 0.0,
                    "MetricIntervalUpperBound": 15.0,
                    "ScalingAdjustment": 1
                },
                {
                    "MetricIntervalLowerBound": 15.0,
                    "MetricIntervalUpperBound": 25.0,
                    "ScalingAdjustment": 2
                },
                {
                    "MetricIntervalLowerBound": 25.0,
                    "ScalingAdjustment": 3
                }
            ],
            "AdjustmentType": "ChangeInCapacity"
        },
```

```
        "PolicyType": "StepScaling",
        "ResourceId": "service/my-cluster/my-service",
        "ServiceNamespace": "ecs",
        "Alarms": [
            {
                "AlarmName": "Step-Scaling-AlarmHigh-ECS:service/my-cluster/my-
service",
                "AlarmARN": "arn:aws:cloudwatch:region:012345678910:alarm:Step-Scaling-
AlarmHigh-ECS:service/my-cluster/my-service"
            }
        ],
        "PolicyName": "my-step-scaling-policy",
        "ScalableDimension": "ecs:service:DesiredCount",
        "CreationTime": 1515024099.901
    }
]
```

Exclua uma política de escalabilidade de etapas para o Application Auto Scaling usando o AWS CLI

Quando você não precisar mais de uma política de dimensionamento em etapas, poderá excluí-la. Para excluir a política de escalabilidade e o CloudWatch alarme associado, conclua as tarefas a seguir.

Para excluir a política de dimensionamento

Use o comando delete-scaling-policy.

Linux, macOS ou Unix

```
aws application-autoscaling delete-scaling-policy --service-namespace ecs \
  --scalable-dimension ecs:service:DesiredCount \
  --resource-id service/my-cluster/my-service \
  --policy-name my-step-scaling-policy
```

Windows

```
aws application-autoscaling delete-scaling-policy --service-namespace ecs ^
  --scalable-dimension ecs:service:DesiredCount ^
  --resource-id service/my-cluster/my-service ^
```

```
  --policy-name my-step-scaling-policy
```

Para excluir o CloudWatch alarme

Use o comando delete-alarms. É possível excluir um ou mais alarmes por vez. Por exemplo, use o comando a seguir para excluir os alarmes `Step-Scaling-AlarmHigh-ECS:service/my-cluster/my-service` e `Step-Scaling-AlarmLow-ECS:service/my-cluster/my-service`.

```
aws cloudwatch delete-alarms --alarm-name Step-Scaling-AlarmHigh-ECS:service/my-cluster/my-service Step-Scaling-AlarmLow-ECS:service/my-cluster/my-service
```

Escalabilidade preditiva para Application Auto Scaling

O escalonamento preditivo escala proativamente seu aplicativo. O escalonamento preditivo analisa os dados históricos de carga para detectar padrões diários ou semanais nos fluxos de tráfego. Ele usa essas informações para prever as necessidades futuras de capacidade para aumentar proativamente a capacidade do seu aplicativo para corresponder à carga prevista.

A escalabilidade preditiva é adequada para situações em que há:

- Tráfego cíclico, como alta utilização de recursos durante o horário comercial e baixa utilização de recursos durante a noite e nos fins de semana
- Padrões on-and-off de carga de trabalho recorrentes, como processamento em lote, testes ou análise periódica de dados.
- Aplicações que demoram muito para inicializar, causando um impacto de latência considerável na performance da aplicação durante eventos de aumento da escala na horizontal

Conteúdo

- Como funciona a escalabilidade preditiva do Application Auto Scaling
- Crie uma política de escalabilidade preditiva para Application Auto Scaling
- Substituir valores de previsão usando ações programadas
- Configurações avançadas de política de escalabilidade preditiva usando métricas personalizadas

Como funciona a escalabilidade preditiva do Application Auto Scaling

Para usar a escala preditiva, crie uma política de escalabilidade preditiva que especifique a CloudWatch métrica a ser monitorada e analisada. Você pode usar uma métrica predefinida ou personalizada. Para que a escala preditiva comece a prever valores futuros, essa métrica deve ter pelo menos 24 horas de dados.

Depois que você cria a política, a escala preditiva começa a analisar os dados de métricas dos últimos 14 dias para identificar padrões. Ele usa essa análise para gerar uma previsão horária dos requisitos de capacidade para as próximas 48 horas. A previsão é atualizada a cada 6 horas usando os CloudWatch dados mais recentes. À medida que novos dados chegam, a escala preditiva é capaz de melhorar continuamente a precisão das previsões futuras.

Primeiro, você pode ativar a escala preditiva no modo somente de previsão. Nesse modo, ele gera previsões de capacidade, mas na verdade não escala sua capacidade com base nessas previsões. Isso permite que você avalie a exatidão e a adequação da previsão.

Depois de revisar os dados de previsão e decidir iniciar a escala com base nesses dados, alterne a política de escala para o modo de previsão e escala. Neste modo:

- Se a previsão esperar um aumento na carga, o escalonamento preditivo aumentará a capacidade.
- Se a previsão esperar uma diminuição na carga, a escalabilidade preditiva não será ampliada para remover a capacidade. Isso garante que você aumente a escala somente quando a demanda realmente cair, e não apenas com base nas previsões. Para remover a capacidade que não é mais necessária, você deve criar uma política de rastreamento de metas ou escalabilidade de etapas, pois elas respondem aos dados métricos em tempo real.

Por padrão, a escala preditiva dimensiona suas metas escaláveis no início de cada hora com base na previsão para aquela hora. Opcionalmente, você pode especificar um horário de início anterior usando a `SchedulingBufferTime` propriedade na operação da `PutScalingPolicy` API. Isso permite que você lance a capacidade prevista antes da demanda prevista, o que dá à nova capacidade tempo suficiente para se preparar para lidar com o tráfego.

Limites máximos de capacidade

Quando as políticas de escalabilidade estão definidas, um grupo não pode aumentar sua capacidade desejada acima do limite da capacidade máxima.

Como alternativa, você pode permitir que a capacidade máxima da meta escalável seja aumentada automaticamente se a capacidade prevista se aproximar ou exceder a capacidade máxima da meta escalável. Para ativar esse comportamento, use as propriedades `MaxCapacityBreachBehavior` e `MaxCapacityBuffer` na operação da API `PutScalingPolicy` ou a configuração de comportamento de capacidade máxima no AWS Management Console.

> **Warning**
>
> Tenha cuidado ao permitir que a capacidade máxima seja aumentada automaticamente. A capacidade máxima não diminui automaticamente de volta ao máximo original.

Comandos normalmente usados para criação, exclusão e gerenciamento de política de escalabilidade

Os comandos comumente usados para trabalhar com políticas de escalabilidade preditiva incluem:

- `register-scalable-target`para registrar AWS ou personalizar recursos como alvos escaláveis, suspender o escalonamento e retomar o escalonamento.
- `put-scaling-policy`para criar uma política de escalabilidade preditiva.
- `get-predictive-scaling-forecast`para recuperar os dados de previsão para uma política de escalabilidade preditiva.
- `describe-scaling-activities`para retornar informações sobre atividades de escalabilidade em um Região da AWS.
- `describe-scaling-policies`para retornar informações sobre políticas de escalabilidade em um Região da AWS.
- `delete-scaling-policy`para excluir uma política de escalabilidade.

Métricas personalizadas

Métricas personalizadas podem ser usadas para prever a capacidade necessária para um aplicativo. Métricas personalizadas são úteis quando métricas predefinidas não são suficientes para capturar a carga em seu aplicativo.

Considerações

As considerações a seguir se aplicam ao trabalhar com escalabilidade preditiva.

- Confirme se a escalabilidade preditiva é adequada para seu aplicativo. Um aplicativo é uma boa opção para escalabilidade preditiva se apresentar padrões de carga recorrentes específicos do dia da semana ou da hora do dia. Avalie a previsão antes de permitir que a escalabilidade preditiva escale ativamente seu aplicativo.
- A escalabilidade preditiva precisa de pelo menos 24 horas de dados históricos para começar a previsão. No entanto, as previsões serão mais eficazes se os dados históricos abrangerem duas semanas completas.
- Escolha uma métrica de carga que represente com precisão a carga total da sua aplicação e que seja o aspecto mais importante a ser escalado.

Crie uma política de escalabilidade preditiva para Application Auto Scaling

O exemplo de política a seguir usa o AWS CLI para configurar uma política de escalabilidade preditiva para o serviço Amazon ECS. Substitua cada *user input placeholder* por suas próprias informações.

Para obter mais informações sobre as CloudWatch métricas que você pode especificar, consulte PredictiveScalingMetricSpecificationa Amazon EC2 Auto Scaling API Reference.

Veja a seguir um exemplo de política com uma configuração de memória predefinida.

```
cat policy.json
{
    "MetricSpecifications": [
        {
            "TargetValue": 40,
            "PredefinedMetricPairSpecification": {
                "PredefinedMetricType": "ECSServiceMemoryUtilization"
            }
        }
    ],
    "SchedulingBufferTime": 3600,
    "MaxCapacityBreachBehavior": "HonorMaxCapacity",
    "Mode": "ForecastOnly"
}
```

O exemplo a seguir ilustra a criação da política executando o put-scaling-policycomando com o arquivo de configuração especificado.

```
aws aas put-scaling-policy \
--service-namespace ecs \
--region us-east-1 \
--policy-name predictive-scaling-policy-example \
--resource-id service/MyCluster/test \
--policy-type PredictiveScaling \
--scalable-dimension ecs:service:DesiredCount \
--predictive-scaling-policy-configuration file://policy.json
```

Em caso de êxito, esse comando retornará o ARN da política.

```
{
"PolicyARN": "arn:aws:autoscaling:us-
east-1:012345678912:scalingPolicy:d1d72dfe-5fd3-464f-83cf-824f16cb88b7:resource/ecs/
service/MyCluster/test:policyName/predictive-scaling-policy-example",
"Alarms": []
}
```

Substituir valores de previsão usando ações programadas

Às vezes, você pode ter informações adicionais sobre seus futuros requisitos de aplicações que o cálculo de previsão não pode levar em conta. Por exemplo, os cálculos de previsão podem subestimar a capacidade necessária para um evento de marketing futuro. Você pode usar ações programadas para substituir temporariamente a previsão durante períodos futuros. As ações programadas podem ser executadas de forma recorrente ou em uma data e hora específicas quando houver flutuações de demanda únicas.

Por exemplo, você pode criar uma ação programada com uma capacidade mínima maior do que a prevista. Em tempo de execução, o Application Auto Scaling atualiza a capacidade mínima de sua meta escalável. Como a escalabilidade preditiva otimiza a capacidade, uma ação agendada com uma capacidade mínima maior que os valores de previsão é honrada. Isso impede que a capacidade seja menor do que o esperado. Para interromper a substituição da previsão, use uma segunda ação programada para retornar a capacidade mínima à configuração original.

O procedimento a seguir descreve as etapas necessárias para substituir a previsão durante períodos futuros.

Tópicos

- Etapa 1: (Opcional) Analisar dados de séries temporais
- Etapa 2: Criar duas ações programadas

> **Important**
>
> Este tópico pressupõe que você esteja tentando substituir a previsão para escalar para uma capacidade maior do que a prevista. Se você precisar diminuir temporariamente a capacidade sem interferência de uma política de escala preditiva, use o modo somente de previsão. Enquanto estiver no modo somente de previsão, a escala preditiva continuará a gerar previsões, mas não aumentará automaticamente a capacidade. Em seguida, você

pode monitorar a utilização dos recursos e diminuir manualmente o tamanho do grupo, conforme necessário.

Etapa 1: (Opcional) Analisar dados de séries temporais

Comece analisando os dados de séries temporais de previsão. Essa é uma etapa opcional, mas é útil quando você deseja entender os detalhes da previsão.

1. Recuperar a previsão

   Após a criação da previsão, é possível consultar um período específico na previsão. O objetivo da consulta é obter uma visão completa dos dados de séries temporais para um período específico.

   Sua consulta pode incluir até dois dias de dados de previsão futura. Se você usa a escalabilidade preditiva há algum tempo, também pode acessar seus dados de previsão anteriores. No entanto, a duração máxima de tempo entre as horas inicial e final é de 30 dias.

   Para recuperar a previsão, use o get-predictive-scaling-forecastcomando. O exemplo a seguir obtém a previsão de escalabilidade preditiva para o serviço Amazon ECS.

   ```
   aws application-autoscaling get-predictive-scaling-forecast --service-namespace ecs
    \
       --scalable-dimension ecs:service:DesiredCount \
       --resource-id 1234567890abcdef0
       --policy-name predictive-scaling-policy \
       --start-time "2021-05-19T17:00:00Z" \
       --end-time "2021-05-19T23:00:00Z"
   ```

   A resposta inclui duas previsões: `LoadForecast` e. `CapacityForecast` `LoadForecast`mostra a previsão de carga horária. `CapacityForecast`mostra os valores previstos para a capacidade necessária por hora para lidar com a carga prevista e, ao mesmo tempo, manter uma determinada`TargetValue`.

2. Identificar o período-alvo

   Identifique a hora ou horas em que a flutuação de demanda única deverá ocorrer. Lembre-se de que as datas e os horários mostrados na previsão estão em UTC.

Etapa 2: Criar duas ações programadas

Em seguida, crie duas ações programadas para um período específico em que sua aplicação terá uma carga maior do que a prevista. Por exemplo, se você tiver um evento de marketing que irá direcionar o tráfego para seu site por um período limitado, poderá programar uma ação única para atualizar a capacidade mínima quando ele começar. Em seguida, agende outra ação para retornar a capacidade mínima para a configuração original quando o evento terminar.

Para criar duas ações programadas para eventos únicos (AWS CLI)

Para criar as ações agendadas, use o put-scheduled-actioncomando.

O exemplo a seguir define uma programação para o Amazon EC2 Auto Scaling que mantém uma capacidade mínima de três instâncias em 19 de maio às 17h por oito horas. Os comandos a seguir mostram como implementar esse cenário.

O primeiro comando put-scheduled-update-group-action instrui o Amazon Auto EC2 Scaling a atualizar a capacidade mínima do grupo de Auto Scaling especificado às 17h UTC de 19 de maio de 2021.

```
aws autoscaling put-scheduled-update-group-action --scheduled-action-name my-event-
start \
  --auto-scaling-group-name my-asg --start-time "2021-05-19T17:00:00Z" --minimum-
capacity 3
```

O segundo comando instrui o Amazon EC2 Auto Scaling a definir a capacidade mínima do grupo como uma à 1h UTC de 20 de maio de 2021.

```
aws autoscaling put-scheduled-update-group-action --scheduled-action-name my-event-end
 \
  --auto-scaling-group-name my-asg --start-time "2021-05-20T01:00:00Z" --minimum-
capacity 1
```

Depois de adicionar essas ações programadas ao grupo Auto Scaling, o Amazon Auto EC2 Scaling faz o seguinte:

- Às 17h UTC em 19 de maio de 2021, a primeira ação programada é executada. Se o grupo tiver menos de três instâncias, ele será expandido para três instâncias. Durante esse período e pelas próximas oito horas, o Amazon EC2 Auto Scaling pode continuar a escalar se a capacidade

prevista for maior do que a capacidade real ou se houver uma política de escalabilidade dinâmica em vigor.

- À 1h da manhã UTC em 20 de maio de 2021, a segunda ação programada é executada. Isso retorna a capacidade mínima para sua configuração original no final do evento.

Escalabilidade com base em programações recorrentes

Para substituir a previsão para o mesmo período de tempo todas as semanas, crie duas ações programadas e forneça a lógica de hora e data usando uma expressão cron.

A expressão cron consiste em cinco campos separados por espaços: [Minute] [Hour] [Day_of_Month] [Month_of_Year] [Day_of_Week]. Os campos podem conter quaisquer valores permitidos, incluindo caracteres especiais.

Por exemplo, esta expressão cron executa a ação todas as terças-feiras às 6h30. O asterisco é usado como um curinga para corresponder a todos os valores de um campo.

```
30 6 * * 2
```

Configurações avançadas de política de escalabilidade preditiva usando métricas personalizadas

Em uma política de escalabilidade preditiva é possível usar métricas predefinidas ou personalizadas. As métricas personalizadas são úteis quando as métricas predefinidas não descrevem suficientemente a carga do seu aplicativo.

Ao criar uma política de escalabilidade preditiva com métricas personalizadas, você pode especificar outras CloudWatch métricas fornecidas por AWS, ou você pode especificar métricas que você mesmo define e publica. Você também pode usar a matemática métrica para agregar e transformar métricas existentes em uma nova série temporal que AWS não é rastreada automaticamente.
A combinação de valores em seus dados, por exemplo, calculando novas somas ou médias, é chamada de agregação. Os dados resultantes são chamados de um agregado.

A seção a seguir contém as práticas recomendados e exemplos de como sstruturar o JSON para a política.

Tópicos

- Práticas recomendadas
- Pré-requisitos
- Estruture o JSON para métricas personalizadas
- Criar uma política de escalação preditiva no console (métricas personalizadas)

Práticas recomendadas

As seguintes práticas recomendadas podem ajudar no uso mais eficaz de métricas personalizadas:

- Para a especificação da métrica de carga, a métrica mais útil é aquela que representa a carga em seu aplicativo.
- A métrica de escalabilidade deve ser inversamente proporcional à capacidade. Ou seja, se a meta escalável aumentar, a métrica de escala deverá diminuir aproximadamente na mesma proporção. Para garantir que a escalabilidade preditiva se comporte conforme o esperado, a métrica de carga e a métrica de escalabilidade também devem se correlacionar fortemente entre si.
- A utilização visada deve corresponder ao tipo de métrica de escalabilidade. Para uma configuração de política que use a utilização da CPU, essa é uma porcentagem visada. Para uma configuração de política que use throughput, como o número de solicitações ou mensagens, esse é o número visado de solicitações ou mensagens por instância durante qualquer intervalo de um minuto.
- Se essas recomendações não forem seguidas, provavelmente os valores futuros previstos da série temporal estarão incorretos. Para validar se os dados estão corretos, você pode visualizar os valores previstos. Como alternativa, depois de criar sua política de escalabilidade preditiva, inspecione os `CapacityForecast` objetos `LoadForecast` retornados por uma chamada para a API. GetPredictiveScalingForecast
- Recomendamos a configuração da escalabilidade preditiva no modo apenas previsão para avaliar a previsão antes que a escalabilidade preditiva comece a modificar ativamente a capacidade.

Pré-requisitos

Para adicionar métricas personalizadas à política de escalação preditiva, você deve ter as permissões `cloudwatch:GetMetricData`.

Para especificar suas próprias métricas em vez das métricas AWS fornecidas, você deve primeiro publicar suas métricas em CloudWatch. Para obter mais informações, consulte Publicação de métricas personalizadas no Guia CloudWatch do usuário da Amazon.

Se publicar suas próprias métricas, certifique-se de publicar os pontos de dados com uma frequência mínima de cinco minutos. O Application Auto Scaling recupera os pontos de dados CloudWatch com base na duração do período necessário. Por exemplo, a especificação da métrica de carga usa métricas horárias para medir a carga em seu aplicativo. CloudWatch usa seus dados métricos publicados para fornecer um único valor de dados para qualquer período de uma hora, agregando todos os pontos de dados com registros de data e hora que se enquadram em cada período de uma hora.

Estruture o JSON para métricas personalizadas

A seção a seguir contém exemplos de como configurar a escalabilidade preditiva para consultar dados do CloudWatch Amazon Auto EC2 Scaling. Há dois métodos diferentes de configurar essa opção, e o método escolhido afeta qual será o formato usado para estruturar JSON para a política de escalação preditiva. Quando você usa matemática de métricas, o formato do JSON varia ainda mais com base na matemática de métrica que está sendo aplicada.

1. Para criar uma política que obtenha dados diretamente de outras CloudWatch métricas fornecidas AWS ou nas quais você publica CloudWatch, consulteExemplo de política de escalação preditiva com métricas personalizadas de carga e de dimensionamento (AWS CLI).
2. Para criar uma política que possa consultar várias CloudWatch métricas e usar expressões matemáticas para criar novas séries temporais com base nessas métricas, consulteUsar expressões de matemática métrica.

Exemplo de política de escalação preditiva com métricas personalizadas de carga e de dimensionamento (AWS CLI)

Para criar uma política de escalabilidade preditiva com métricas personalizadas de carga e escalabilidade com o. AWS CLI, armazene os argumentos para `--predictive-scaling-configuration` em um arquivo JSON chamado. `config.json`

Você começa a adicionar métricas personalizadas substituindo os valores substituíveis no exemplo a seguir por suas métricas e sua meta de utilização.

```
{
  "MetricSpecifications": [
    {
      "TargetValue": 50,
      "CustomizedScalingMetricSpecification": {
        "MetricDataQueries": [
```

```
          {
            "Id": "scaling_metric",
            "MetricStat": {
              "Metric": {
                "MetricName": "MyUtilizationMetric",
                "Namespace": "MyNameSpace",
                "Dimensions": [
                  {
                    "Name": "MyOptionalMetricDimensionName",
                    "Value": "MyOptionalMetricDimensionValue"
                  }
                ]
              },
              "Stat": "Average"
            }
          }
        ]
      },
      "CustomizedLoadMetricSpecification": {
        "MetricDataQueries": [
          {
            "Id": "load_metric",
            "MetricStat": {
              "Metric": {
                "MetricName": "MyLoadMetric",
                "Namespace": "MyNameSpace",
                "Dimensions": [
                  {
                    "Name": "MyOptionalMetricDimensionName",
                    "Value": "MyOptionalMetricDimensionValue"
                  }
                ]
              },
              "Stat": "Sum"
            }
          }
        ]
      }
    }
  ]
}
```

Para obter mais informações, consulte MetricDataQuerya Amazon EC2 Auto Scaling API Reference.

> **Note**
>
> Veja a seguir alguns recursos adicionais que podem ajudá-lo a encontrar nomes de métricas, namespaces, dimensões e estatísticas para CloudWatch métricas:
>
> - Para obter informações sobre as métricas disponíveis para AWS serviços, consulte AWS serviços que publicam CloudWatch métricas no Guia CloudWatch do usuário da Amazon.
> - Para obter o nome exato da métrica, o namespace e as dimensões (se aplicável) de uma CloudWatch métrica com o AWS CLI, consulte list-metrics.

Para criar essa política, execute o put-scaling-policycomando usando o arquivo JSON como entrada, conforme demonstrado no exemplo a seguir.

```
aws autoscaling put-scaling-policy --policy-name my-predictive-scaling-policy \
  --auto-scaling-group-name my-asg --policy-type PredictiveScaling \
  --predictive-scaling-configuration file://config.json
```

Se bem-sucedido, esse comando gerará o nome do recurso da Amazon (ARN) da política.

```
{
  "PolicyARN": "arn:aws:autoscaling:region:account-id:scalingPolicy:2f4f5048-d8a8-4d14-
b13a-d1905620f345:autoScalingGroupName/my-asg:policyName/my-predictive-scaling-policy",
  "Alarms": []
}
```

Usar expressões de matemática métrica

A seção a seguir fornece informações e exemplos de políticas de escalação preditiva que mostram como você pode usar a matemática de métricas em sua política.

Tópicos

- Noções básicas de matemática métrica
- Exemplo de política de escalabilidade preditiva para o Amazon Auto EC2 Scaling que combina métricas usando matemática métrica ()AWS CLI
- Exemplo de política de escalação preditiva para usar em um cenário de implantação azul/verde (AWS CLI)

Noções básicas de matemática métrica

Se tudo o que você quer fazer é agregar dados métricos existentes, a matemática CloudWatch métrica poupa o esforço e o custo de publicar outra métrica no. CloudWatch Você pode usar qualquer métrica que AWS forneça e também pode usar métricas que você define como parte de seus aplicativos.

Para obter mais informações, consulte Usando matemática métrica no Guia CloudWatch do usuário da Amazon.

Se você optar por usar uma expressão matemática métrica em sua política de escalabilidade preditiva, considere os seguintes pontos:

- As operações matemáticas métricas usam os pontos de dados da combinação exclusiva de nome da métrica, namespace e pares de métricas de chaves-valor da dimensão.
- Você pode usar qualquer operador aritmético (+ - */^), função estatística (como AVG ou SUM) ou outra função compatível. CloudWatch
- Você pode usar as métricas e os resultados de outras expressões matemáticas nas fórmulas da expressão matemática.
- Suas expressões matemáticas métricas podem ser compostas de agregações diferentes. No entanto, uma prática recomendada para o resultado final da agregação é usar `Average` para a métrica de escalabilidade e `Sum` para a métrica de carga.
- Qualquer expressão usada em uma especificação de métrica deve eventualmente retornar uma única série temporal.

Para usar matemática métrica, faça o seguinte:

- Escolha uma ou mais CloudWatch métricas. Em seguida, crie a expressão. Para obter mais informações, consulte Usando matemática métrica no Guia CloudWatch do usuário da Amazon.
- Verifique se a expressão matemática métrica é válida usando o CloudWatch console ou a CloudWatch GetMetricDataAPI.

Exemplo de política de escalabilidade preditiva para o Amazon Auto EC2 Scaling que combina métricas usando matemática métrica ()AWS CLI

Às vezes, ao invés de especificar a métrica diretamente, talvez seja necessário processar seus dados de alguma forma, primeiramente. Por exemplo, você pode ter uma aplicação que extrai o

trabalho de uma fila do Amazon SQS e talvez queira usar o número de itens na fila como critério para escalabilidade preditiva. O número de mensagens na fila não define unicamente o número necessário de instâncias. Portanto, é necessário mais trabalho para criar uma métrica que possa ser usada para calcular a lista de pendências por instância.

Veja a seguir um exemplo de política de escalabilidade preditiva para esse cenário. Ele especifica métricas de escalabilidade e carga baseadas na métrica `ApproximateNumberOfMessagesVisible` do Amazon SQS, que é o número de mensagens disponíveis para recuperação da fila. Ele também usa a `GroupInServiceInstances` métrica Amazon EC2 Auto Scaling e uma expressão matemática para calcular o backlog por instância da métrica de escalabilidade.

```
aws autoscaling put-scaling-policy --policy-name my-sqs-custom-metrics-policy \
  --auto-scaling-group-name my-asg --policy-type PredictiveScaling \
  --predictive-scaling-configuration file://config.json
{
  "MetricSpecifications": [
    {
      "TargetValue": 100,
      "CustomizedScalingMetricSpecification": {
        "MetricDataQueries": [
          {
            "Label": "Get the queue size (the number of messages waiting to be
 processed)",
            "Id": "queue_size",
            "MetricStat": {
              "Metric": {
                "MetricName": "ApproximateNumberOfMessagesVisible",
                "Namespace": "AWS/SQS",
                "Dimensions": [
                  {
                    "Name": "QueueName",
                    "Value": "my-queue"
                  }
                ]
              },
              "Stat": "Sum"
            },
            "ReturnData": false
          },
          {
            "Label": "Get the group size (the number of running instances)",
```

```
            "Id": "running_capacity",
            "MetricStat": {
              "Metric": {
                "MetricName": "GroupInServiceInstances",
                "Namespace": "AWS/AutoScaling",
                "Dimensions": [
                  {
                    "Name": "AutoScalingGroupName",
                    "Value": "my-asg"
                  }
                ]
              },
              "Stat": "Sum"
            },
            "ReturnData": false
          },
          {
            "Label": "Calculate the backlog per instance",
            "Id": "scaling_metric",
            "Expression": "queue_size / running_capacity",
            "ReturnData": true
          }
        ]
      },
      "CustomizedLoadMetricSpecification": {
        "MetricDataQueries": [
          {
            "Id": "load_metric",
            "MetricStat": {
              "Metric": {
                "MetricName": "ApproximateNumberOfMessagesVisible",
                "Namespace": "AWS/SQS",
                "Dimensions": [
                  {
                    "Name": "QueueName",
                    "Value": "my-queue"
                  }
                ],
              },
              "Stat": "Sum"
            },
            "ReturnData": true
          }
        ]
```

```
      }
    }
  ]
}
```

O exemplo retorna o ARN da política.

```
{
  "PolicyARN": "arn:aws:autoscaling:region:account-id:scalingPolicy:2f4f5048-d8a8-4d14-
b13a-d1905620f345:autoScalingGroupName/my-asg:policyName/my-sqs-custom-metrics-policy",
  "Alarms": []
}
```

Exemplo de política de escalação preditiva para usar em um cenário de implantação azul/verde (AWS CLI)

Uma expressão de pesquisa fornece uma opção avançada na qual você pode consultar para obter uma métrica de vários grupos do Auto Scaling e realizar expressões matemáticas neles. Isso é útil especialmente para implantações azul/verde.

> **Note**
>
> Uma implantação azul/verde é um método de implantação no qual você cria dois grupos do Auto Scaling separados, mas idênticos. Apenas um dos grupos recebe tráfego de produção. O tráfego do usuário é inicialmente direcionado para o grupo do Auto Scaling anterior (“azul”), enquanto um novo grupo (“verde”) é usado para testar e avaliar uma nova versão de uma aplicação ou serviço. O tráfego do usuário é deslocado para o grupo do Auto Scaling verde depois que uma nova implantação é testada e aceita. Em seguida, é possível excluir o grupo azul depois que a implantação for bem-sucedida.

Quando novos grupos do Auto Scaling são criados como parte de uma implantação azul/verde, o histórico de métricas de cada grupo pode ser incluído automaticamente na política de escalabilidade preditiva sem que você precise alterar suas especificações métricas. Para obter mais informações, consulte Como usar políticas de escalabilidade preditiva do EC2 Auto Scaling com implantações azul/verde no blog de computação. AWS

O exemplo de política a seguir mostra como isso pode ser feito. Neste exemplo, a política usa a `CPUUtilization` métrica emitida pela Amazon EC2. Ele usa a `GroupInServiceInstances` métrica Amazon EC2 Auto Scaling e uma expressão matemática para calcular o valor da métrica de

escalabilidade por instância. Ela também especifica uma especificação de métrica de capacidade para obter a métrica `GroupInServiceInstances`.

A expressão de pesquisa encontra o `CPUUtilization` de instâncias em vários grupos do Auto Scaling com base nos critérios de pesquisa especificados. Se, posteriormente, você criar um novo grupo do Auto Scaling que corresponda aos mesmos critérios de pesquisa, o `CPUUtilization` das instâncias no novo grupo do Auto Scaling são incluídas automaticamente.

```
aws autoscaling put-scaling-policy --policy-name my-blue-green-predictive-scaling-
policy \
  --auto-scaling-group-name my-asg --policy-type PredictiveScaling \
  --predictive-scaling-configuration file://config.json
{
  "MetricSpecifications": [
    {
      "TargetValue": 25,
      "CustomizedScalingMetricSpecification": {
        "MetricDataQueries": [
          {
            "Id": "load_sum",
            "Expression": "SUM(SEARCH('{AWS/EC2,AutoScalingGroupName} MetricName=
\"CPUUtilization\" ASG-myapp', 'Sum', 300))",
            "ReturnData": false
          },
          {
            "Id": "capacity_sum",
            "Expression": "SUM(SEARCH('{AWS/AutoScaling,AutoScalingGroupName}
 MetricName=\"GroupInServiceInstances\" ASG-myapp', 'Average', 300))",
            "ReturnData": false
          },
          {
            "Id": "weighted_average",
            "Expression": "load_sum / capacity_sum",
            "ReturnData": true
          }
        ]
      },
      "CustomizedLoadMetricSpecification": {
        "MetricDataQueries": [
          {
            "Id": "load_sum",
            "Expression": "SUM(SEARCH('{AWS/EC2,AutoScalingGroupName} MetricName=
\"CPUUtilization\" ASG-myapp', 'Sum', 3600))"
```

```
          }
        ]
      },
      "CustomizedCapacityMetricSpecification": {
        "MetricDataQueries": [
          {
            "Id": "capacity_sum",
            "Expression": "SUM(SEARCH('{AWS/AutoScaling,AutoScalingGroupName}
 MetricName=\"GroupInServiceInstances\" ASG-myapp', 'Average', 300))"
          }
        ]
      }
    }
  ]
}
```

O exemplo retorna o ARN da política.

```
{
  "PolicyARN": "arn:aws:autoscaling:region:account-id:scalingPolicy:2f4f5048-d8a8-4d14-
b13a-d1905620f345:autoScalingGroupName/my-asg:policyName/my-blue-green-predictive-
scaling-policy",
  "Alarms": []
}
```

Criar uma política de escalação preditiva no console (métricas personalizadas)

Se ocorrer um problema ao usar métricas personalizadas, recomendamos fazer o seguinte:

- Se uma mensagem de erro for fornecida, leia a mensagem e resolva o problema que ela relata, se possível.
- Se você não validou uma expressão com antecedência, o put-scaling-policycomando a valida ao criar sua política de escalabilidade. No entanto, existe a possibilidade de que esse comando não identifique a causa exata dos erros detectados. Para corrigir os problemas, solucione os erros que você recebe em uma resposta de uma solicitação ao get-metric-datacomando. Você também pode solucionar o problema da expressão no CloudWatch console.
- Você deve especificar `false` para `ReturnData` se `MetricDataQueries` especificar a função SEARCH() (BUSCAR) por conta própria sem uma função matemática como SUM() (SOMA).

Isso ocorre porque as expressões de pesquisa podem retornar várias séries temporais, e uma especificação métrica baseado em uma expressão pode retornar apenas uma série temporal.

- Todas as métricas envolvidas em uma expressão de pesquisa devem ter a mesma resolução.

Limitações

As limitações a seguir são aplicáveis.

- Você pode consultar pontos de dados de até 10 métricas em uma especificação métrica.
- Para os propósitos desse limite, uma expressão conta como uma métrica.

Tutorial: configurar o ajuste de escala automático para processar uma workload pesada

Neste tutorial, você verá como aumentar a escala horizontalmente e com base em janelas de tempo quando a aplicação tiver uma workload mais pesada do que o normal. Isso é útil quando você tem uma aplicação que pode de repente ter um grande número de visitantes em um horário regular ou em uma base sazonal.

Você pode usar uma política de dimensionamento com monitoramento do objetivo com a escalabilidade agendada para lidar com a carga extra. A escalabilidade agendada inicia automaticamente as alterações nas suas `MinCapacity` e `MaxCapacity` em seu nome com base em uma programação especificada por você. Quando uma política de dimensionamento com monitoramento do objetivo está ativa no recurso, ela pode ser escalada dinamicamente com base na utilização atual de recursos dentro do novo intervalo de capacidade mínima e máxima.

Após concluir este tutorial, você saberá como:

- Usar a escalabilidade programada para adicionar capacidade extra para atender a uma carga pesada antes que ela chegue e remover a capacidade extra quando ela não for mais necessária.
- Usar uma política de dimensionamento com monitoramento do objetivo para escalar a aplicação com base na utilização atual de recursos.

Conteúdo

- Pré-requisitos
- Etapa 1: inscrever o destino escalável
- Etapa 2: configurar ações programadas de acordo com as suas necessidades
- Etapa 3: adicionar uma política de dimensionamento com monitoramento do objetivo
- Etapa 4: próximas etapas
- Etapa 5: limpar

Pré-requisitos

Este tutorial pressupõe que você já tenha feito o seguinte:

- Criou um Conta da AWS.

- Instalou e configurou AWS CLI o.
- Concedeu as permissões necessárias para registrar e cancelar o registro de recursos como destinos escaláveis no Application Auto Scaling. Além disso, concedeu as permissões necessárias para criar políticas de escalabilidade e ações programadas. Para obter mais informações, consulte Gerenciamento de Identidade e Acesso para o Application Auto Scaling.
- Criou um recurso compatível em um ambiente que não é de produção disponível para uso neste tutorial. Se não tiver, crie uma conta agora. Para obter mais informações sobre os serviços e os recursos da AWS que você pode usar com o Application Auto Scaling, consulte a seção Serviços da AWS que você pode usar com o Application Auto Scaling.

> **Note**
>
> Ao concluir este tutorial, há duas etapas nas quais você define os valores de capacidade mínimo e máximo do seu recurso como 0 para redefinir a capacidade atual como 0. Dependendo do recurso que escolheu usar com o Application Auto Scaling, talvez você não consiga redefinir a capacidade atual para 0 durante essas etapas. Para ajudá-lo a resolver o problema, uma mensagem na saída indicará que a capacidade mínima não pode ser menor que o valor especificado e fornecerá o valor mínimo de capacidade que o AWS recurso pode aceitar.

Etapa 1: inscrever o destino escalável

Comece inscrevendo o recurso como um destino escalável com o Application Auto Scaling. Um destino escalável é um recurso cuja escala pode ser aumentada ou reduzida horizontalmente pelo Application Auto Scaling.

Para inscrever um destino escalável com o Application Auto Scaling

- Use o register-scalable-targetcomando a seguir para registrar um novo alvo escalável. Defina os valores `--min-capacity` e `--max-capacity` como 0 para redefinir a capacidade atual como 0.

  Substitua o texto de amostra por `--service-namespace` com o namespace do serviço da AWS que você está usando com o Application Auto Scaling, `--scalable-dimension` com a dimensão escalável associada ao recurso que você está registrando e `--resource-id` com um identificador para o recurso. Esses valores variam com base em qual recurso é usado

e como o ID do recurso é construído. Veja os tópicos na seção Serviços da AWS que você pode usar com o Application Auto Scaling para obter mais informações. Esses tópicos incluem exemplos de comandos que mostram como registrar destinos escaláveis com o Application Auto Scaling.

Linux, macOS ou Unix

```
aws application-autoscaling register-scalable-target \
  --service-namespace namespace \
  --scalable-dimension dimension \
  --resource-id identifier \
  --min-capacity 0 --max-capacity 0
```

Windows

```
aws application-autoscaling register-scalable-target --service-namespace namespace
 --scalable-dimension dimension --resource-id identifier --min-capacity 0 --max-
capacity 0
```

Se obtiver êxito, esse comando retornará o ARN do destino escalável.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-
id:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Etapa 2: configurar ações programadas de acordo com as suas necessidades

Você pode usar o put-scheduled-actioncomando para criar ações agendadas que são configuradas para atender às suas necessidades comerciais. Neste tutorial, focaremos em uma configuração que para de consumir recursos fora do horário de trabalho, reduzindo a capacidade para 0.

Criar uma ação programada que seja ampliada pela manhã

1. Para escalar a meta escalável, use o put-scheduled-actioncomando a seguir. Incluia o parâmetro `--schedule` com uma programação recorrente, em UTC, usando uma expressão cron.

Na programação especificada (todos os dias às 9:00 UTC), o Application Auto Scaling atualiza os valores `MinCapacity` e `MaxCapacity` para a faixa desejada de uma a cinco unidades de capacidade.

Linux, macOS ou Unix

```
aws application-autoscaling put-scheduled-action \
  --service-namespace namespace \
  --scalable-dimension dimension \
  --resource-id identifier \
  --scheduled-action-name my-first-scheduled-action \
  --schedule "cron(0 9 * * ? *)" \
  --scalable-target-action MinCapacity=1,MaxCapacity=5
```

Windows

```
aws application-autoscaling put-scheduled-action --service-namespace namespace --
scalable-dimension dimension --resource-id identifier --scheduled-action-name my-
first-scheduled-action --schedule "cron(0 9 * * ? *)" --scalable-target-action
 MinCapacity=1,MaxCapacity=5
```

Esse comando não retornará nenhuma saída se for bem-sucedido.

2. Para confirmar que sua ação agendada existe, use o describe-scheduled-actionscomando a seguir.

Linux, macOS ou Unix

```
aws application-autoscaling describe-scheduled-actions \
  --service-namespace namespace \
  --query 'ScheduledActions[?ResourceId==`identifier`]'
```

Windows

```
aws application-autoscaling describe-scheduled-actions --service-
namespace namespace --query "ScheduledActions[?ResourceId==`identifier`]"
```

O seguinte é um exemplo de saída.

```
[
```

```
    {
        "ScheduledActionName": "my-first-scheduled-action",
        "ScheduledActionARN": "arn",
        "Schedule": "cron(0 9 * * ? *)",
        "ScalableTargetAction": {
            "MinCapacity": 1,
            "MaxCapacity": 5
        },
        ...
    }
]
```

Criar uma ação programada que seja reduzida à noite

1. Repita o procedimento anterior para criar outra ação programada que o Application Auto Scaling use para reduzir a escala ao final do dia.

   Na programação especificada (todos os dias às 20h UTC), o Application Auto Scaling atualiza o `MaxCapacity` e `MinCapacity` do alvo para 0, conforme as instruções do comando a seguir. put-scheduled-action

   Linux, macOS ou Unix

   ```
   aws application-autoscaling put-scheduled-action \
     --service-namespace namespace \
     --scalable-dimension dimension \
     --resource-id identifier \
     --scheduled-action-name my-second-scheduled-action \
     --schedule "cron(0 20 * * ? *)" \
     --scalable-target-action MinCapacity=0,MaxCapacity=0
   ```

   Windows

   ```
   aws application-autoscaling put-scheduled-action --service-namespace namespace --
   scalable-dimension dimension --resource-id identifier --scheduled-action-name my-
   second-scheduled-action --schedule "cron(0 20 * * ? *)" --scalable-target-action
    MinCapacity=0,MaxCapacity=0
   ```

2. Para confirmar que sua ação agendada existe, use o describe-scheduled-actionscomando a seguir.

Linux, macOS ou Unix

```
aws application-autoscaling describe-scheduled-actions \
  --service-namespace namespace \
  --query 'ScheduledActions[?ResourceId==`identifier`]'
```

Windows

```
aws application-autoscaling describe-scheduled-actions --service-
namespace namespace --query "ScheduledActions[?ResourceId==`identifier`]"
```

O seguinte é um exemplo de saída.

```
[
    {
        "ScheduledActionName": "my-first-scheduled-action",
        "ScheduledActionARN": "arn",
        "Schedule": "cron(0 9 * * ? *)",
        "ScalableTargetAction": {
            "MinCapacity": 1,
            "MaxCapacity": 5
        },
        ...
    },
    {
        "ScheduledActionName": "my-second-scheduled-action",
        "ScheduledActionARN": "arn",
        "Schedule": "cron(0 20 * * ? *)",
        "ScalableTargetAction": {
            "MinCapacity": 0,
            "MaxCapacity": 0
        },
        ...
    }
]
```

Etapa 3: adicionar uma política de dimensionamento com monitoramento do objetivo

Agora que você tem a programação básica em vigor, adicione uma política de dimensionamento com monitoramento do objetivo para escalar com base na utilização atual de recursos.

Com o monitoramento do objetivo, o Application Auto Scaling compara o valor do objetivo na política com o valor atual da métrica especificada. Quando eles são desiguais por um período de tempo, o Application Auto Scaling adiciona ou remove capacidade para manter uma performance estável. À medida que a carga na aplicação e o valor métrico aumentam, o Application Auto Scaling adiciona capacidade o mais rápido possível sem ultrapassar `MaxCapacity`. Quando o Application Auto Scaling remove a capacidade porque a carga é mínima, ele faz isso sem ultrapassar `MinCapacity`. Ao ajustar a capacidade com base no uso, você paga apenas pelas necessidades da aplicação.

Se a métrica tiver dados insuficientes porque a aplicação não tem nenhuma carga, o Application Auto Scaling não adicionará ou removerá capacidade. Em outras palavras, o Application Auto Scaling prioriza a disponibilidade em situações em que não haja informação suficiente disponível.

Você pode adicionar várias políticas de escalabilidade, mas certifique-se de não adicionar políticas de escalabilidade em etapa conflitantes, o que pode causar comportamento indesejável. Por exemplo, se a política de escalabilidade de etapas iniciar uma atividade de redução antes que a política de rastreamento de destino esteja pronta para ser reduzida, a atividade de redução não será bloqueada. Após a conclusão da atividade de redução, a política de monitoramento do objetivo pode instruir o Application Auto Scaling a aumentar a escala novamente.

Para criar uma política de escalabilidade com monitoramento do objetivo

1. Use o comando da put-scaling-policy a seguir para criar a política.

   As métricas usadas com mais frequência para o monitoramento de destinos são predefinidas e você pode usá-las sem fornecer a especificação de métrica completa do CloudWatch. Para mais informações sobre as métricas predefinidas disponíveis, consulte Políticas de dimensionamento com monitoramento do objetivo para o Application Auto Scaling.

   Antes de executar esse comando, certifique-se de que a métrica predefinida espere o valor do objetivo. Por exemplo, para aumentar a escala horizontalmente quando a CPU atinge 50% de utilização, especifique um valor alvo de 50,0. Ou, para aumentar a escala horizontalmente da simultaneidade provisionada do Lambda quando o uso atingir 70% de utilização, especifique um valor do objetivo de 0,7. Para obter informações sobre valores de destino para um

recurso específico, consulte a documentação fornecida pelo serviço sobre como configurar o monitoramento do objetivo. Para obter mais informações, consulte Serviços da AWS que você pode usar com o Application Auto Scaling.

Linux, macOS ou Unix

```
aws application-autoscaling put-scaling-policy \
  --service-namespace namespace \
  --scalable-dimension dimension \
  --resource-id identifier \
  --policy-name my-scaling-policy --policy-type TargetTrackingScaling \
  --target-tracking-scaling-policy-configuration '{ "TargetValue": 50.0,
 "PredefinedMetricSpecification": { "PredefinedMetricType": "predefinedmetric" }}'
```

Windows

```
aws application-autoscaling put-scaling-policy --service-namespace namespace --
scalable-dimension dimension --resource-id identifier --policy-name my-scaling-
policy --policy-type TargetTrackingScaling --target-tracking-scaling-policy-
configuration "{ \"TargetValue\": 50.0, \"PredefinedMetricSpecification\":
 { \"PredefinedMetricType\": \"predefinedmetric\" }}"
```

Se for bem-sucedido, esse comando retornará os nomes ARNs e dos dois CloudWatch alarmes que foram criados em seu nome.

2. Para confirmar que sua ação agendada existe, use o describe-scaling-policiescomando a seguir.

Linux, macOS ou Unix

```
aws application-autoscaling describe-scaling-policies --service-namespace namespace
 \
  --query 'ScalingPolicies[?ResourceId==`identifier`]'
```

Windows

```
aws application-autoscaling describe-scaling-policies --service-namespace namespace
 --query "ScalingPolicies[?ResourceId==`identifier`]"
```

O seguinte é um exemplo de saída.

```
[
    {
        "PolicyARN": "arn",
        "TargetTrackingScalingPolicyConfiguration": {
            "PredefinedMetricSpecification": {
                "PredefinedMetricType": "predefinedmetric"
            },
            "TargetValue": 50.0
        },
        "PolicyName": "my-scaling-policy",
        "PolicyType": "TargetTrackingScaling",
        "Alarms": [],
        ...
    }
]
```

Etapa 4: próximas etapas

Quando ocorre uma ação de escalabilidade, você verá um registro dela na saída das ações de escalabilidade para o destino escalável, por exemplo:

```
Successfully set desired count to 1. Change successfully fulfilled by ecs.
```

Para monitorar suas atividades de escalabilidade com o Application Auto Scaling, você pode usar o comando a describe-scaling-activitiesseguir.

Linux, macOS ou Unix

```
aws application-autoscaling describe-scaling-activities
  --service-namespace namespace \
  --scalable-dimension dimension \
  --resource-id identifier
```

Windows

```
aws application-autoscaling describe-scaling-activities --service-namespace namespace
 --scalable-dimension dimension --resource-id identifier
```

Etapa 5: limpar

Para evitar que sua conta acumule cobranças de recursos criados durante a escalabilidade ativa, você pode limpar a configuração de escalabilidade associada da seguinte maneira.

A exclusão da configuração de escalabilidade não exclui o recurso subjacente AWS . Também não os devolve à sua capacidade original. Você pode usar o console do serviço em que criou o recurso para excluí-lo ou ajustar sua capacidade.

Como excluir as ações programadas

O seguinte comando delete-scheduled-action exclui uma ação programada especificada. Você pode ignorar esta etapa se deseja manter as ações programadas criadas.

Linux, macOS ou Unix

```
aws application-autoscaling delete-scheduled-action \
  --service-namespace namespace \
  --scalable-dimension dimension \
  --resource-id identifier \
  --scheduled-action-name my-second-scheduled-action
```

Windows

```
aws application-autoscaling delete-scheduled-action --service-namespace namespace
 --scalable-dimension dimension --resource-id identifier --scheduled-action-name my-
second-scheduled-action
```

Excluir a política de escalabilidade

O delete-scaling-policycomando a seguir exclui uma política de escalabilidade de rastreamento de destino especificada. Você pode ignorar esta etapa se deseja manter as políticas de escalabilidade criadas.

Linux, macOS ou Unix

```
aws application-autoscaling delete-scaling-policy \
  --service-namespace namespace \
  --scalable-dimension dimension \
  --resource-id identifier \
```

```
  --policy-name my-scaling-policy
```

Windows

```
aws application-autoscaling delete-scaling-policy --service-namespace namespace --
scalable-dimension dimension --resource-id identifier --policy-name my-scaling-policy
```

Como cancelar o registro do destino dimensionável:

Use o seguinte comando deregister-scalable-target para cancelar o registro do destino dimensionável. Se tiver qualquer política de dimensionamento que você criou ou qualquer ação programada que ainda não foi excluída, elas serão excluídas por esse comando. Você poderá ignorar esta etapa se desejar manter o destino dimensionável registrado para uso futuro.

Linux, macOS ou Unix

```
aws application-autoscaling deregister-scalable-target \
  --service-namespace namespace \
  --scalable-dimension dimension \
  --resource-id identifier
```

Windows

```
aws application-autoscaling deregister-scalable-target --service-namespace namespace --
scalable-dimension dimension --resource-id identifier
```

Suspender e retomar a escalabilidade do Application Auto Scaling

Este tópico explica como suspender e retomar uma ou mais das ações de dimensionamento para os destinos dimensionáveis no aplicativo. O recurso de suspender e retomar é usado para pausar temporariamente as atividades de dimensionamento acionadas pelas políticas de dimensionamento e pelas ações programadas. Isso pode ser útil, por exemplo, quando você quer eliminar a possibilidade de o dimensionamento automático interferir enquanto você está fazendo uma alteração ou investigando um problema de configuração. As políticas de dimensionamento e as ações programadas podem ser mantidas e, quando você estiver pronto, as ações de dimensionamento poderão ser reiniciadas.

O exemplo de comandos da CLI a seguir, é necessário passar os parâmetros formatados em JSON em um arquivo config.json. Você também pode passar esses parâmetros na linha de comando usando aspas para incluir a estrutura de dados JSON. Para obter mais informações, consulte Usar aspas com strings na AWS CLI no Manual do usuário da AWS Command Line Interface .

Conteúdo

- Atividades de escalabilidade
- Suspender e retomar atividades de escalabilidade

> **Note**
>
> Para conferir instruções sobre como suspender os processos de aumento horizontal da escala enquanto as implantações do Amazon ECS estiverem em andamento, consulte a documentação a seguir:
> Escalabilidade automática e implantações do serviço no Guia do desenvolvedor do Amazon Elastic Container Service

Atividades de escalabilidade

O Application Auto Scaling oferece suporte para que as atividades de escalabilidade a seguir sejam colocadas em um estado suspenso:

- Todas as atividades de redução que são acionadas por uma política de dimensionamento.

- Todas as atividades de expansão que são acionados por uma política de dimensionamento.
- Todas as ações de dimensionamento que envolvem ações programadas.

As descrições a seguir explicam o que acontece quando as ações de dimensionamento individuais são suspensas. Cada uma pode ser suspensa e retomada de forma independente. Dependendo do motivo da suspensão de uma ação de dimensionamento, pode ser necessário suspender várias ações de dimensionamento em conjunto.

`DynamicScalingInSuspended`

- O Application Auto Scaling não remove a capacidade quando uma política de dimensionamento do monitoramento do objetivo ou uma política de escalabilidade de etapa é acionada. Isso permite que você desabilite temporariamente atividades de redução associadas a políticas de dimensionamento sem excluir as políticas de dimensionamento ou seus alarmes do CloudWatch associados. Quando você retomar a redução, o Application Auto Scaling avalia políticas com limites de alarme que estão atualmente em falha.

`DynamicScalingOutSuspended`

- O Application Auto Scaling não remove a capacidade quando uma política de dimensionamento com monitoramento do objetivo ou uma política de escalabilidade de etapa é acionada. Isso permite que você desabilite temporariamente atividades de ampliação associadas a políticas de dimensionamento sem excluir as políticas de dimensionamento ou seus alarmes do CloudWatch associados. Quando você retomar o aumento da escala, o Application Auto Scaling avalia políticas com limites de alarme que estão atualmente em falha.

`ScheduledScalingSuspended`

- O Application Auto Scaling não inicia as ações de escalabilidade programadas para execução durante o período de suspensão. Quando você retomar a escalabilidade programada, o Application Auto Scaling somente avaliará as ações programadas cujo tempo de execução ainda não tenha passado.

Suspender e retomar atividades de escalabilidade

Você pode suspender e retomar atividades de escalabilidade individuais ou todas as atividades de escalabilidade para o destino de escalabilidade do Application Auto Scaling.

> **Note**
>
> Em resumo, esses exemplos ilustram como suspender e retomar a escalabilidade para uma tabela do DynamoDB. Para especificar um destino escalável diferente, especifique o namespace em `--service-namespace`, sua dimensão escalável em `--scalable-dimension`, e o ID do recurso em `--resource-id`. Para obter mais informações e exemplos de cada serviço, consulte os tópicos na Serviços da AWS que você pode usar com o Application Auto Scaling.

Para suspender uma atividade de dimensionamento

Abra uma janela da linha de comando e use o comando register-scalable-target com a opção `--suspended-state` da maneira indicada a seguir.

Linux, macOS ou Unix

```
aws application-autoscaling register-scalable-target --service-namespace dynamodb \
  --scalable-dimension dynamodb:table:ReadCapacityUnits --resource-id table/my-table \
  --suspended-state file://config.json
```

Windows

```
aws application-autoscaling register-scalable-target --service-namespace dynamodb --
scalable-dimension dynamodb:table:ReadCapacityUnits --resource-id table/my-table --
suspended-state file://config.json
```

Se obtiver êxito, esse comando retornará o ARN do destino escalável.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Para suspender somente atividades de redução que são acionadas por uma política de dimensionamento, especifique o seguinte no config.json.

```
{
    "DynamicScalingInSuspended":true
}
```

Para suspender somente atividades de ampliação que são acionadas por uma política de dimensionamento, especifique o seguinte no config.json.

```
{
    "DynamicScalingOutSuspended":true
}
```

Para suspender somente atividades de dimensionamento que envolvem ações programadas, especifique o seguinte no config.json.

```
{
    "ScheduledScalingSuspended":true
}
```

Para suspender todas as atividades de dimensionamento

Use o comando register-scalable-target com a opção `--suspended-state` da seguinte forma.

Linux, macOS ou Unix

```
aws application-autoscaling register-scalable-target --service-namespace dynamodb \
  --scalable-dimension dynamodb:table:ReadCapacityUnits --resource-id table/my-table \
  --suspended-state file://config.json
```

Windows

```
aws application-autoscaling register-scalable-target --service-namespace dynamodb --
scalable-dimension dynamodb:table:ReadCapacityUnits --resource-id table/my-table --
suspended-state file://config.json
```

Este exemplo pressupõe que o arquivo config.json contém os parâmetros formatados em JSON a seguir.

```
{
    "DynamicScalingInSuspended":true,
    "DynamicScalingOutSuspended":true,
    "ScheduledScalingSuspended":true
}
```

Se obtiver êxito, esse comando retornará o ARN do destino escalável.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Visualizar atividades de escalabilidade suspensas

Use o comando describe-scalable-targets para determinar quais ações de escalabilidade estão em um estado suspenso para um destino dimensionável.

Linux, macOS ou Unix

```
aws application-autoscaling describe-scalable-targets --service-namespace dynamodb \
  --scalable-dimension dynamodb:table:ReadCapacityUnits --resource-id table/my-table
```

Windows

```
aws application-autoscaling describe-scalable-targets --service-namespace dynamodb --
scalable-dimension dynamodb:table:ReadCapacityUnits --resource-id table/my-table
```

O seguinte é um exemplo de saída.

```
{
    "ScalableTargets": [
        {
            "ServiceNamespace": "dynamodb",
            "ScalableDimension": "dynamodb:table:ReadCapacityUnits",
            "ResourceId": "table/my-table",
            "MinCapacity": 1,
            "MaxCapacity": 20,
            "SuspendedState": {
                "DynamicScalingOutSuspended": true,
```

```
                "DynamicScalingInSuspended": true,
                "ScheduledScalingSuspended": true
            },
            "CreationTime": 1558125758.957,
            "RoleARN": "arn:aws:iam::123456789012:role/aws-
service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable"
        }
    ]
}
```

Retomar atividades de escalabilidade

Quando estiver pronto para retomar a atividade de dimensionamento, você poderá retomá-la usando o comando register-scalable-target.

O exemplo de comando a seguir retoma todas as atividades de dimensionamento para o destino dimensionável especificado.

Linux, macOS ou Unix

```
aws application-autoscaling register-scalable-target --service-namespace dynamodb \
  --scalable-dimension dynamodb:table:ReadCapacityUnits --resource-id table/my-table \
  --suspended-state file://config.json
```

Windows

```
aws application-autoscaling register-scalable-target --service-namespace dynamodb --
scalable-dimension dynamodb:table:ReadCapacityUnits --resource-id table/my-table --
suspended-state file://config.json
```

Este exemplo pressupõe que o arquivo config.json contém os parâmetros formatados em JSON a seguir.

```
{
    "DynamicScalingInSuspended":false,
    "DynamicScalingOutSuspended":false,
    "ScheduledScalingSuspended":false
}
```

Se obtiver êxito, esse comando retornará o ARN do destino escalável.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Atividades de escalação para o Application Auto Scaling

O Application Auto Scaling monitora as CloudWatch métricas da sua política de escalabilidade e inicia uma atividade de escalabilidade quando os limites são excedidos. Ele também inicia atividades de escalação quando você modifica o tamanho máximo ou mínimo do alvo escalável, seja manualmente ou seguindo um cronograma.

Quando ocorre uma atividade de escalação, o Application Auto Scaling faz uma das seguintes ações:

- Aumenta a capacidade do alvo escalável (chamado de aumento de escala horizontal)
- Diminui a capacidade do alvo escalável (chamado de aumento de escala horizontal)

Você pode pesquisar as atividades de escalação das últimas seis semanas.

Pesquisar atividades de escalabilidade por destino escalável

Para ver as atividades de escalabilidade de um alvo escalável específico, use o comando a seguir describe-scaling-activities.

Linux, macOS ou Unix

```
aws application-autoscaling describe-scaling-activities --service-namespace ecs \
  --scalable-dimension ecs:service:DesiredCount --resource-id service/my-cluster/my-service
```

Windows

```
aws application-autoscaling describe-scaling-activities --service-namespace ecs --
scalable-dimension ecs:service:DesiredCount --resource-id service/my-cluster/my-service
```

Veja a seguir um exemplo de resposta no qual `StatusCode` contém o status atual da atividade e `StatusMessage` contém a mensagem sobre o status da atividade de escalação.

```
{
    "ScalingActivities": [
        {
            "ScalableDimension": "ecs:service:DesiredCount",
            "Description": "Setting desired count to 1.",
            "ResourceId": "service/my-cluster/my-service",
```

```
            "ActivityId": "e6c5f7d1-dbbb-4a3f-89b2-51f33e766399",
            "StartTime": 1462575838.171,
            "ServiceNamespace": "ecs",
            "EndTime": 1462575872.111,
            "Cause": "monitor alarm web-app-cpu-lt-25 in state ALARM triggered policy
 web-app-cpu-lt-25",
            "StatusMessage": "Successfully set desired count to 1. Change successfully
 fulfilled by ecs.",
            "StatusCode": "Successful"
        }
    ]
}
```

Para obter uma descrição dos campos na resposta, consulte ScalingActivitya Referência da API Application Auto Scaling.

Os códigos de status a seguir indicam quando o evento de escalação que leva à atividade de escalação atinge um estado concluído:

- `Successful`: a escalação foi concluída com êxito
- `Overridden`: a capacidade desejada foi atualizada por um evento de escalação mais recente
- `Unfulfilled`: a escalação atingiu o tempo limite ou o serviço alvo não pode atender à solicitação
- `Failed`: a escalação falhou com uma exceção

> **Note**
>
> A atividade de escalação também pode ter um status `Pending` ou `InProgress`. Todas as atividades de escalação têm um status `Pending` até que o serviço-alvo responda. Depois que o alvo responde, o status da atividade de escalação passa a ser `InProgress`.

Incluir atividades que não sofreram ajuste de escala

Por padrão, as atividades de escalação não refletem as ocasiões em que o Application Auto Scaling toma uma decisão sobre se a escalação não deve ser feita.

Por exemplo, suponha que um serviço do Amazon ECS exceda o limite máximo de uma determinada métrica, mas o número de tarefas já tenha atingido o máximo permitido. Nesse caso, o Application Auto Scaling não aumenta horizontalmente a escala do número desejado de tarefas.

Para incluir atividades que não são escalonadas (não atividades escalonadas) na resposta, adicione a `--include-not-scaled-activities` opção ao describe-scaling-activitiescomando.

Linux, macOS ou Unix

```
aws application-autoscaling describe-scaling-activities --include-not-scaled-activities
 \
  --service-namespace ecs --scalable-dimension ecs:service:DesiredCount \
  --resource-id service/my-cluster/my-service
```

Windows

```
aws application-autoscaling describe-scaling-activities --include-not-scaled-activities
 --service-namespace ecs --scalable-dimension ecs:service:DesiredCount --resource-
id service/my-cluster/my-service
```

> **Note**
>
> Se esse comando gerar um erro, verifique se você atualizou o AWS CLI localmente para a versão mais recente.

Para confirmar que a resposta inclui as atividades não escaladas, o elemento `NotScaledReasons` é mostrado na saída para algumas ou para todas as atividades de escalação que falharam.

```
{
    "ScalingActivities": [
        {
            "ScalableDimension": "ecs:service:DesiredCount",
            "Description": "Attempting to scale due to alarm triggered",
            "ResourceId": "service/my-cluster/my-service",
            "ActivityId": "4d759079-a31f-4d0c-8468-504c56e2eecf",
            "StartTime": 1664928867.915,
            "ServiceNamespace": "ecs",
            "Cause": "monitor alarm web-app-cpu-gt-75 in state ALARM triggered policy
 web-app-cpu-gt-75",
            "StatusCode": "Failed",
            "NotScaledReasons": [
                {
                    "Code": "AlreadyAtMaxCapacity",
                    "MaxCapacity": 4
```

```
                }
            ]
        }
    ]
}
```

Para obter uma descrição dos campos na resposta, consulte ScalingActivitya Referência da API Application Auto Scaling.

Se uma atividade não escalada for retornada, dependendo do código de motivo listado em `Code`, atributos como `CurrentCapacity`, `MaxCapacity` e `MinCapacity` podem estar presentes na resposta.

Para evitar grandes quantidades de entradas duplicadas, somente a primeira atividade que não sofreu ajuste de escala será registrada no histórico de atividades de escalabilidade. As atividades subsequentes que não sofreram ajuste de escala não gerarão novas entradas, a menos que o motivo da não terem sofrido ajuste mude.

Códigos de motivo

A seguir estão os códigos de motivos para uma atividade não escalada.

Código do motivo	Definição			
AutoScalingAnticipatedFlapping	O algoritmo de escalação automático decidiu não realizar uma ação de escalação porque isso causaria oscilações. Oscilação é um ciclo infinito de aumento e redução de escala horizontal. Ou seja, se uma			

Código do motivo	Definição			
	ação de escalação fosse feita, o valor da métrica seria alterado para iniciar outra ação de escalação na direção inversa.			
TargetSer vicePutRe sourceAsl nscalable	O serviço de destino colocou temporariamente o recurso em um estado não escalável. O Application Auto Scaling tentará escalar novamente quando as condições de escalabil idade automátic a especificadas na política de escalabilidade forem atendidas.			

Código do motivo	Definição			
AlreadyAtMaxCapacity	A escalação é impedida pela capacidade máxima que você especificou. Se você quiser que o Application Auto Scaling aumente a escala horizontalmente, será necessário aumentar a capacidade máxima.			
AlreadyAtMinCapacity	A escalação é impedida pela capacidade mínima que você especificou. Se você quiser que o Application Auto Scaling reduza a escala horizontalmente, será necessário diminuir a capacidade máxima.			

Código do motivo	Definição			
AlreadyAtDesiredCapacity	O algoritmo de escalação automática calculou que a capacidade revisada é igual à capacidade atual.			

Monitorar o Application Auto Scaling

O monitoramento é uma parte importante da manutenção da confiabilidade, disponibilidade e desempenho do Application Auto Scaling e de suas outras AWS soluções. Você deve coletar dados de monitoramento de todas as partes da sua AWS solução para poder depurar com mais facilidade uma falha multiponto, caso ocorra. AWS fornece ferramentas de monitoramento para monitorar o Application Auto Scaling, relatar quando algo está errado e realizar ações automáticas quando apropriado.

Você pode usar os seguintes recursos para ajudá-lo a gerenciar seus AWS recursos:

AWS CloudTrail

Com AWS CloudTrail, você pode acompanhar as chamadas feitas para a API Application Auto Scaling por ou em nome de você. Conta da AWS CloudTrail armazena as informações em arquivos de log no bucket do Amazon S3 que você especificar. É possível identificar quais usuários e contas chamaram o Application Auto Scaling, o endereço IP de origem das chamadas e quando elas ocorreram. Para obter mais informações, consulte Registre as chamadas da API Application Auto Scaling usando AWS CloudTrail.

> **Note**
>
> Para obter informações sobre outros AWS serviços que podem ajudá-lo a registrar e coletar dados sobre suas cargas de trabalho, consulte o guia de registro e monitoramento para proprietários de aplicativos na Orientação AWS prescritiva.

Amazon CloudWatch

CloudWatch A Amazon ajuda você a analisar registros e, em tempo real, monitorar as métricas de seus AWS recursos e aplicativos hospedados. Você pode coletar e rastrear métricas, criar painéis personalizados e definir alarmes que o notificam ou que realizam ações quando uma métrica especificada atinge um limite definido. Por exemplo, você pode CloudWatch rastrear a utilização de recursos e notificá-lo quando a utilização estiver muito alta ou quando o alarme da métrica estiver no `INSUFFICIENT_DATA` estado. Para obter mais informações, consulte Monitore o uso de recursos escaláveis usando CloudWatch.

CloudWatch também monitora as métricas de uso AWS da API para Application Auto Scaling. Você pode usar essas métricas para configurar alarmes que alertem quando o volume de

chamadas da API violar um limite definido por você. Para obter mais informações, consulte as métricas de AWS uso no Guia CloudWatch do usuário da Amazon.

Amazon EventBridge

EventBridge A Amazon é um serviço de ônibus de eventos sem servidor que facilita a conexão de seus aplicativos com dados de várias fontes. EventBridge fornece um fluxo de dados em tempo real de seus próprios aplicativos, aplicativos Software-as-a-Service (SaaS) e AWS serviços e encaminha esses dados para destinos como o Lambda. Isso permite monitorar eventos que ocorrem em serviços e criar arquiteturas orientadas a eventos. Para obter mais informações, consulte Monitore eventos do Application Auto Scaling usando a Amazon EventBridge.

AWS Health Dashboard

O AWS Health Dashboard (PHD) exibe informações e também fornece notificações que são invocadas por mudanças na integridade dos AWS recursos. As informações são apresentadas de duas formas: em um painel que mostra eventos recentes e futuros organizados por categoria e em um log de eventos completo que mostra todos os eventos dos últimos 90 dias. Para ter mais informações, consulte Conceitos básicos do AWS Health Dashboard.

Monitore o uso de recursos escaláveis usando CloudWatch

Com a Amazon CloudWatch, você obtém visibilidade quase contínua de seus aplicativos em recursos escaláveis. CloudWatch é um serviço de monitoramento de AWS recursos. Você pode usar CloudWatch para coletar e monitorar métricas, definir alarmes e reagir automaticamente às mudanças em seus AWS recursos. Você também pode criar painéis para monitorar as métricas específicas ou os conjuntos de métricas de que você precisa.

Quando você interage com os serviços que se integram ao Application Auto Scaling, eles enviam as métricas mostradas na tabela a seguir para. CloudWatch Em CloudWatch, as métricas são agrupadas primeiro pelo namespace do serviço e depois pelas várias combinações de dimensões em cada namespace. Essas métricas podem ajudar você a monitorar o uso de recursos e a planejar capacidade para as aplicações. Se a workload da sua aplicação não for constante, você deverá considerar o uso do Auto Scaling. Para obter descrições detalhadas dessas métricas, consulte a documentação referente à métrica de interesse.

Conteúdo

- CloudWatch métricas para monitorar o uso de recursos
- Métricas predefinidas para políticas de escalação com rastreamento de destino

CloudWatch métricas para monitorar o uso de recursos

A tabela a seguir lista as CloudWatch métricas que estão disponíveis para apoiar o monitoramento do uso de recursos. A lista não é exaustiva, mas é um bom ponto de partida. Se você não vê essas métricas no CloudWatch console, certifique-se de ter concluído a configuração do recurso. Para obter mais informações, consulte o Guia CloudWatch do usuário da Amazon.

Recursos escaláveis	Namespace	CloudWatch métrica	Link para a documentação
AppStream 2.0			
Frotas	AWS/AppStream	Nome: AvailableCapacity Dimensão: frota	AppStream Métricas 2.0
Frotas	AWS/AppStream	Nome: CapacityUtilization Dimensão: frota	AppStream Métricas 2.0
Aurora			
Réplicas	AWS/RDS	Nome: CPUUtilization Dimensões: DBClusteridentificador, função (LEITOR)	Métricas no nível do cluster do Aurora

Recursos escaláveis	Namespace	CloudWatch métrica	Link para a documentação
Réplicas	AWS/RDS	Nome: DatabaseConnections Dimensões: DBClusteridentificador, função (LEITOR)	Métricas no nível do cluster do Aurora
Amazon Comprehend			
Endpoints de classificação de documento	AWS/Comprehend	Nome: InferenceUtilization Dimensão: EndpointArn	Métricas de endpoint do Amazon Comprehend
Endpoints do reconhecedor de entidades	AWS/Comprehend	Nome: InferenceUtilization Dimensão: EndpointArn	Métricas de endpoint do Amazon Comprehend
DynamoDB			

Recursos escaláveis	Namespace	CloudWatch métrica	Link para a documentação
Tabelas e índices secundários globais	AWS/DynamoDB	Nome: ProvisionedReadCapacityUnits Dimensões: TableName, GlobalSecondaryIndexName	Métricas do DynamoDB
Tabelas e índices secundários globais	AWS/DynamoDB	Nome: ProvisionedWriteCapacityUnits Dimensões: TableName, GlobalSecondaryIndexName	Métricas do DynamoDB

Recursos escaláveis	Namespace	CloudWatch métrica	Link para a documentação
Tabelas e índices secundários globais	AWS/DynamoDB	Nome: ConsumedReadCapacityUnits Dimensões: TableName, GlobalSecondaryIndexName	Métricas do DynamoDB
Tabelas e índices secundários globais	AWS/DynamoDB	Nome: ConsumedWriteCapacityUnits Dimensões: TableName, GlobalSecondaryIndexName	Métricas do DynamoDB
Amazon ECS			

Recursos escaláveis	Namespace	CloudWatch métrica	Link para a documentação
Serviços	AWS/ECS	Nome: CPUUtilization Dimensões: ClusterName, ServiceName	Métricas do Amazon ECS
Serviços	AWS/ECS	Nome: MemoryUtilization Dimensões: ClusterName, ServiceName	Métricas do Amazon ECS
Serviços	AWS/ApplicationELB	Nome: RequestCountPerTarget Dimensão: TargetGroup	Métricas do Application Load Balancer
ElastiCache			

Recursos escaláveis	Namespace	CloudWatch métrica	Link para a documentação
Clusters (grupos de replicação)	AWS/ElastiCache	Nome: DatabaseMemoryUsageCountedForEvictPercentage Dimensão: ReplicationGroupId	ElastiCache Métricas do Valkey e do Redis OSS
Clusters (grupos de replicação)	AWS/ElastiCache	Nome: DatabaseCapacityUsageCountedForEvictPercentage Dimensão: ReplicationGroupId	ElastiCache Métricas do Valkey e do Redis OSS
Clusters (grupos de replicação)	AWS/ElastiCache	Nome: MotorCPUUtilization Dimensões: ReplicationGroupId, Função (primária)	ElastiCache Métricas do Valkey e do Redis OSS

Recursos escaláveis	Namespace	CloudWatch métrica	Link para a documentação
Clusters (grupos de replicação)	AWS/ElastiCache	Nome: MotorCPUUtilization Dimensões: ReplicationGroupId, Função (réplica)	ElastiCache Métricas do Valkey e do Redis OSS
Clusters (cache)	AWS/ElastiCache	Nome: MotorCPUUtilization Dimensões: CacheClusterId, Node	ElastiCache Métricas do Memcached
Clusters (cache)	AWS/ElastiCache	Nome: DatabaseCapacityMemoryUsagePercentage Dimensões: CacheClusterId	ElastiCache Métricas do Memcached

Recursos escaláveis	Namespace	CloudWatch métrica	Link para a documentação
Amazon EMR			
Clusters	AWS/ElasticMapReduce	Nome: YARNMemoryAvailablePercentage Dimensão: ClusterId	Métricas do Amazon EMR
Amazon Keyspaces			
Tabelas	AWS/Cassandra	Nome: ProvisionedReadCapacityUnits Dimensões: Keyspace, TableName	Métricas do Amazon Keyspaces
Tabelas	AWS/Cassandra	Nome: ProvisionedWriteCapacityUnits Dimensões: Keyspace, TableName	Métricas do Amazon Keyspaces

Recursos escaláveis	Namespace	CloudWatch métrica	Link para a documentação
Tabelas	AWS/Cassandra	Nome: ConsumedReadCapacityUnits Dimensões: Keyspace, TableName	Métricas do Amazon Keyspaces
Tabelas	AWS/Cassandra	Nome: ConsumedWriteCapacityUnits Dimensões: Keyspace, TableName	Métricas do Amazon Keyspaces
Lambda			
Simultaneidade provisionada	AWS/Lambda	Nome: ProvisionedConcurrencyUtilization Dimensões: FunctionName, Recurso	Métricas de função do Lambda
Amazon MSK			

Recursos escaláveis	Namespace	CloudWatch métrica	Link para a documentação
Amazenamento de agente	AWS/Kafka	Nome: KafkaDataLogsDiskUsed Dimensões: nome do cluster	Métricas do Amazon MSK
Amazenamento de agente	AWS/Kafka	Nome: KafkaDataLogsDiskUsed Dimensions: Cluster Name, Broker ID	Métricas do Amazon MSK
Neptune			
Clusters	AWS/Neptune	Nome: CPUUtilization Dimensões: DBCluster identificador, função (LEITOR)	Métricas do Neptune
SageMaker AI			

Recursos escaláveis	Namespace	CloudWatch métrica	Link para a documentação
Variantes de endpoint	AWS/SageMaker	Nome: InvocationsPerInstance Dimensões: EndpointName, VariantName	Métricas de invocação
Componentes de inferência	AWS/SageMaker	Nome: InvocationsPerCopy Dimensões: InferenceComponentName	Métricas de invocação
Simultaneidade provisionada para um endpoint sem servidor	AWS/SageMaker	Nome: ServerlessProvisionedConcurrencyUtilization Dimensões: EndpointName, VariantName	Métricas de endpoint de tecnologia sem servidor

Recursos escaláveis	Namespace	CloudWatch métrica	Link para a documentação
Frota Spot (Amazon EC2)			
Spot Fleets	AWS/Spot EC2	Nome: CPUUtilization Dimensão: FleetRequestId	Métricas de frota spot
Spot Fleets	AWS/Spot EC2	Nome: NetworkIn Dimensão: FleetRequestId	Métricas de frota spot
Spot Fleets	AWS/Spot EC2	Nome: NetworkOut Dimensão: FleetRequestId	Métricas de frota spot
Spot Fleets	AWS/ApplicationELB	Nome: RequestCountPerTarget Dimensão: TargetGroup	Métricas do Application Load Balancer

Métricas predefinidas para políticas de escalação com rastreamento de destino

A tabela a seguir lista os tipos de métricas predefinidos da Application Auto Scaling API Reference com o nome da métrica CloudWatch correspondente. Cada métrica predefinida representa uma agregação dos valores da métrica CloudWatch subjacente. O resultado é o uso médio dos recursos durante um período de um minuto, baseado em uma porcentagem, salvo indicação em contrário. As métricas predefinidas só são usadas no contexto de configuração de políticas de escalação com rastreamento de destino.

Mais informações sobre essas métricas podem ser encontradas na documentação do serviço que está disponível na tabela em CloudWatch métricas para monitorar o uso de recursos.

Tipo de métrica predefinida	CloudWatch nome da métrica
AppStream 2.0	
`AppStreamAverageCapacityUtilization`	CapacityUtilization
Aurora	
`RDSReaderAverageCPUUtilization`	CPUUtilization
`RDSReaderAverageDatabaseConnections`	DatabaseConnections[1]
Amazon Comprehend	
`ComprehendInferenceUtilization`	InferenceUtilization
DynamoDB	
`DynamoDBReadCapacityUtilization`	ProvisionedReadCapacityUnits, ConsumedReadCapacityUnits [2]
`DynamoDBWriteCapacityUtilization`	ProvisionedWriteCapacityUnits, ConsumedWriteCapacityUnits [2]
Amazon ECS	

Tipo de métrica predefinida	CloudWatch nome da métrica
`ECSServiceAverageCPUUtilization`	CPUUtilization
`ECSServiceAverageMemoryUtilization`	MemoryUtilization
`ALBRequestCountPerTarget`	RequestCountPerTarget[1]
ElastiCache	
`ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage`	DatabaseMemoryUsageCountedForEvictPercentage
`ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage`	DatabaseCapacityUsageCountedForEvictPercentage
`ElastiCachePrimaryEngineCPUUtilization`	Motor CPUUtilization
`ElastiCacheReplicaEngineCPUUtilization`	Motor CPUUtilization
`ElastiCacheEngineCPUUtilization`	Motor CPUUtilization
`ElastiCacheDatabaseMemoryUsagePercentage`	DatabaseMemoryUsagePercentage
Amazon Keyspaces	
`CassandraReadCapacityUtilization`	ProvisionedReadCapacityUnits, ConsumedReadCapacityUnits [2]
`CassandraWriteCapacityUtilization`	ProvisionedWriteCapacityUnits, ConsumedWriteCapacityUnits [2]
Lambda	
`LambdaProvisionedConcurrencyUtilization`	ProvisionedConcurrencyUtilization

Tipo de métrica predefinida	CloudWatch nome da métrica
Amazon MSK	
`KafkaBrokerStorageUtilization`	KafkaDataLogsDiskUsed
Neptune	
`NeptuneReaderAverageCPUUtilization`	CPUUtilization
SageMaker AI	
`SageMakerVariantInvocationsPerInstance`	InvocationsPerInstance[1]
`SageMakerInferenceComponentInvocationsPerCopy`	InvocationsPerCopy[1]
`SageMakerVariantProvisionedConcurrencyUtilization`	ServerlessProvisionedConcurrencyUtilization
`SageMakerInferenceComponentConcurrentRequestsPerCopyHighResolution`	ConcurrentRequestsPerCopy
`SageMakerVariantConcurrentRequestsPerModelHighResolution`	ConcurrentRequestsPerModel
Frota spot	
`EC2SpotFleetRequestAverageCPUUtilization`	CPUUtilization[3]
`EC2SpotFleetRequestAverageNetworkIn`[3]	NetworkIn[1] [3]
`EC2SpotFleetRequestAverageNetworkOut`[3]	NetworkOut[1] [3]

Tipo de métrica predefinida	CloudWatch nome da métrica
`ALBRequestCountPerTarget`	RequestCountPerTarget[1]

[1] A métrica é baseada em uma contagem em vez de uma porcentagem.

[2] Para o DynamoDB e o Amazon Keyspaces, as métricas predefinidas são uma agregação de duas métricas para oferecer suporte à escalabilidade com base no consumo CloudWatch de taxa de transferência provisionada.

[3] Para obter o melhor desempenho de escalabilidade, o monitoramento EC2 detalhado da Amazon deve ser usado.

Registre as chamadas da API Application Auto Scaling usando AWS CloudTrail

O Application Auto Scaling é integrado com AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou um. AWS service (Serviço da AWS) CloudTrail captura chamadas de API para Application Auto Scaling como eventos. As chamadas capturadas incluem as chamadas do AWS Management Console e as chamadas de código às operações de API do Application Auto Scaling. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita ao Application Auto Scaling, o endereço IP a partir do qual a solicitação foi feita, quando foi feita e detalhes adicionais.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar o seguinte:

- Se a solicitação foi feita com credenciais de usuário raiz ou credenciais de usuário.
- Se a solicitação foi feita em nome de um usuário do Centro de Identidade do IAM.
- Se a solicitação foi feita com credenciais de segurança temporárias de um perfil ou de um usuário federado.
- Se a solicitação foi feita por outro AWS service (Serviço da AWS).

CloudTrail está ativo Conta da AWS quando você cria a conta e você tem acesso automático ao histórico de CloudTrail eventos. O histórico de CloudTrail eventos fornece um registro visível, pesquisável, baixável e imutável dos últimos 90 dias de eventos de gerenciamento registrados

em um. Região da AWS Para obter mais informações, consulte Trabalhando com o histórico de CloudTrail eventos no Guia AWS CloudTrail do usuário. Não há CloudTrail cobrança pela visualização do histórico de eventos.

Para um registro contínuo dos eventos dos Conta da AWS últimos 90 dias, crie uma trilha.

CloudTrail trilhas

Uma trilha permite CloudTrail entregar arquivos de log para um bucket do Amazon S3. Todas as trilhas criadas usando o AWS Management Console são multirregionais. Só é possível criar uma trilha de região única ou de várias regiões usando a AWS CLI. É recomendável criar uma trilha multirregional porque você captura todas as atividades Regiões da AWS em sua conta. Ao criar uma trilha de região única, é possível visualizar somente os eventos registrados na Região da AWS da trilha. Para obter mais informações sobre trilhas, consulte Criar uma trilha para a Conta da AWS e Criar uma trilha para uma organização no Guia do usuário do AWS CloudTrail .

Você pode entregar uma cópia dos seus eventos de gerenciamento contínuos para o bucket do Amazon S3 sem nenhum custo CloudTrail criando uma trilha. No entanto, há cobranças de armazenamento do Amazon S3. Para obter mais informações sobre CloudTrail preços, consulte AWS CloudTrail Preços. Para receber informações sobre a definição de preços do Amazon S3, consulte Definição de preços do Amazon S3.

Eventos de gerenciamento do Application Auto Scaling em CloudTrail

Os eventos de gerenciamento fornecem informações sobre as operações de gerenciamento que são realizadas nos recursos do seu Conta da AWS. Também são conhecidas como operações de ambiente de gerenciamento. Por padrão, CloudTrail registra eventos de gerenciamento.

O Application Auto Scaling registra em log todas as operações do ambiente de gerenciamento do Application Auto Scaling na forma de eventos de gerenciamento. Para obter uma lista das operações do plano de controle do Application Auto Scaling nas quais o Application Auto Scaling CloudTrail registra, consulte a Referência da API do Application Auto Scaling.

Exemplos de eventos do Application Auto Scaling

Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a operação de API solicitada, a data e a hora da operação, os parâmetros da solicitação e assim por diante. CloudTrail os arquivos de log não são um rastreamento de pilha ordenado das chamadas públicas de API, portanto, os eventos não aparecem em nenhuma ordem específica.

O exemplo a seguir mostra um CloudTrail evento que demonstra a `DescribeScalableTargets` operação.


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "Root",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:root",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-08-21T17:05:42Z"
            }
        }
    },
    "eventTime": "2018-08-16T23:20:32Z",
    "eventSource": "autoscaling.amazonaws.com",
    "eventName": "DescribeScalableTargets",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "72.21.196.68",
    "userAgent": "EC2 Spot Console",
    "requestParameters": {
        "serviceNamespace": "ec2",
        "scalableDimension": "ec2:spot-fleet-request:TargetCapacity",
        "resourceIds": [
            "spot-fleet-request/sfr-05ceaf79-3ba2-405d-e87b-612857f1357a"
        ]
    },
    "responseElements": null,
    "additionalEventData": {
        "service": "application-autoscaling"
    },
    "requestID": "0737e2ea-fb2d-11e3-bfd8-99133058e7bb",
    "eventID": "3fcfb182-98f8-4744-bd45-b38835ab61cb",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```


Para obter informações sobre o conteúdo do CloudTrail registro, consulte o conteúdo do CloudTrail registro no Guia AWS CloudTrail do usuário.

O Application Auto Scaling ativa RemoveAction CloudWatch

Seu AWS CloudTrail registro pode mostrar que o Application Auto Scaling chama a CloudWatch `RemoveAction` API quando o Application Auto Scaling CloudWatch instrui a remover a ação de escalonamento automático de um alarme. Isso pode acontecer se você cancelar o registro de uma meta escalável, excluir uma política de escalabilidade ou se um alarme invocar uma política de escalabilidade inexistente.

Monitore eventos do Application Auto Scaling usando a Amazon EventBridge

A Amazon EventBridge, anteriormente chamada de CloudWatch Events, ajuda você a monitorar eventos específicos do Application Auto Scaling e a iniciar ações-alvo que usam outros. Serviços da AWS Os eventos de Serviços da AWS são entregues quase EventBridge em tempo real.

Usando EventBridge, você pode criar regras que correspondam aos eventos recebidos e encaminhá-los aos alvos para processamento.

Para obter mais informações, consulte Introdução à Amazon EventBridge no Guia do EventBridge usuário da Amazon.

Eventos do Application Auto Scaling

Os seguintes exemplos mostram eventos do Application Auto Scaling. Os eventos são emitidos com base no melhor esforço.

Atualmente, somente eventos específicos para escalado até o máximo e para chamadas de API via CloudTrail estão disponíveis para o Application Auto Scaling.

Tipos de eventos

- Evento para alteração de estado: dimensionado ao máximo
- Eventos para chamadas de API via CloudTrail

Evento para alteração de estado: dimensionado ao máximo

O seguinte evento de exemplo mostra que o Application Auto Scaling elevou (aumentou a escala horizontalmente) a capacidade do destino dimensionável até seu limite de tamanho máximo. Se a

demanda aumentar novamente, o Application Auto Scaling será impedido de dimensionar o destino para um tamanho maior, pois ele já está dimensionado com seu tamanho máximo.

No objeto `detail`, os valores para os atributos `resourceId`, `serviceNamespace` e `scalableDimension` identificam o destino dimensionável. Os valores dos atributos `newDesiredCapacity` e `oldDesiredCapacity` referem-se à nova capacidade após o evento de aumento da escala na horizontal e à capacidade original antes do evento de aumento da escala. O `maxCapacity` é o limite máximo de tamanho do destino dimensionável.

```
{
  "version": "0",
  "id": "11112222-3333-4444-5555-666677778888",
  "detail-type": "Application Auto Scaling Scaling Activity State Change",
  "source": "aws.application-autoscaling",
  "account": "123456789012",
  "time": "2019-06-12T10:23:40Z",
  "region": "us-west-2",
  "resources": [],
  "detail": {
    "startTime": "2022-06-12T10:20:43Z",
    "endTime": "2022-06-12T10:23:40Z",
    "newDesiredCapacity": 8,
    "oldDesiredCapacity": 5,
    "minCapacity": 2,
    "maxCapacity": 8,
    "resourceId": "table/my-table",
    "scalableDimension": "dynamodb:table:WriteCapacityUnits",
    "serviceNamespace": "dynamodb",
    "statusCode": "Successful",
    "scaledToMax": true,
    "direction": "scale-out"
}
```

Para criar uma regra que capture todos os eventos de alteração de estado `scaledToMax` para todos os destinos dimensionáveis, use a seguinte amostra de padrão de evento.

```
{
  "source": [
    "aws.application-autoscaling"
  ],
  "detail-type": [
    "Application Auto Scaling Scaling Activity State Change"
```

```
  ],
  "detail": {
    "scaledToMax": [
      true
    ]
  }
}
```

Eventos para chamadas de API via CloudTrail

Uma trilha é uma configuração AWS CloudTrail usada para entregar eventos como arquivos de log em um bucket do Amazon S3. CloudTrail arquivos de log contêm entradas de log. Um evento representa uma entrada de log e inclui informações sobre a ação solicitada, a data e hora da ação e os parâmetros da solicitação. Para saber como começar CloudTrail, consulte Criação de uma trilha no Guia do AWS CloudTrail usuário.

Os eventos que são entregues por meio de CloudTrail têm `AWS API Call via CloudTrail` como valor`detail-type`.

O evento de exemplo a seguir representa uma entrada de arquivo de CloudTrail log que mostra que um usuário do console chamou a ação Application Auto Scaling. RegisterScalableTarget

```
{
  "version": "0",
  "id": "99998888-7777-6666-5555-444433332222",
  "detail-type": "AWS API Call via CloudTrail",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "2022-07-13T16:50:15Z",
  "region": "us-west-2",
  "resources": [],
  "detail": {
    "eventVersion": "1.08",
    "userIdentity": {
      "type": "IAMUser",
      "principalId": "123456789012",
      "arn": "arn:aws:iam::123456789012:user/Bob",
      "accountId": "123456789012",
      "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
      "sessionContext": {
        "sessionIssuer": {
          "type": "Role",
```

```
          "principalId": "123456789012",
          "arn": "arn:aws:iam::123456789012:role/Admin",
          "accountId": "123456789012",
          "userName": "Admin"
        },
        "webIdFederationData": {},
        "attributes": {
          "creationDate": "2022-07-13T15:17:08Z",
          "mfaAuthenticated": "false"
        }
      }
    },
    "eventTime": "2022-07-13T16:50:15Z",
    "eventSource": "autoscaling.amazonaws.com",
    "eventName": "RegisterScalableTarget",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "EC2 Spot Console",
    "requestParameters": {
      "resourceId": "spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
      "serviceNamespace": "ec2",
      "scalableDimension": "ec2:spot-fleet-request:TargetCapacity",
      "minCapacity": 2,
      "maxCapacity": 10
    },
    "responseElements": null,
    "additionalEventData": {
      "service": "application-autoscaling"
    },
    "requestID": "e9caf887-8d88-11e5-a331-3332aa445952",
    "eventID": "49d14f36-6450-44a5-a501-b0fdcdfaeb98",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management",
    "sessionCredentialFromConsole": "true"
  }
}
```


Para criar uma regra com base em todas DeleteScalingPolicyas chamadas de DeregisterScalableTargetAPI para todos os destinos escaláveis, use o seguinte exemplo de padrão de evento:

```
{
  "source": [
    "aws.autoscaling"
  ],
  "detail-type": [
    "AWS API Call via CloudTrail"
  ],
  "detail": {
    "eventSource": [
      "autoscaling.amazonaws.com"
    ],
    "eventName": [
       "DeleteScalingPolicy",
       "DeregisterScalableTarget"
    ],
    "additionalEventData": {
      "service": [
        "application-autoscaling"
      ]
    }
  }
}
```

Para obter mais informações sobre o uso CloudTrail, consulteRegistre as chamadas da API Application Auto Scaling usando AWS CloudTrail.

Usando esse serviço com um AWS SDK

AWS kits de desenvolvimento de software (SDKs) estão disponíveis para muitas linguagens de programação populares. Cada SDK fornece uma API, exemplos de código e documentação que permitem que os desenvolvedores criem facilmente aplicações em seu idioma de preferência.

Documentação do SDK	Exemplos de código
AWS SDK para C++	AWS SDK para C++ exemplos de código
AWS CLI	AWS CLI exemplos de código
AWS SDK para Go	AWS SDK para Go exemplos de código
AWS SDK para Java	AWS SDK para Java exemplos de código
AWS SDK para JavaScript	AWS SDK para JavaScript exemplos de código
AWS SDK para Kotlin	AWS SDK para Kotlin exemplos de código
AWS SDK para .NET	AWS SDK para .NET exemplos de código
AWS SDK para PHP	AWS SDK para PHP exemplos de código
Ferramentas da AWS para PowerShell	Ferramentas da AWS para PowerShell exemplos de código
AWS SDK para Python (Boto3)	AWS SDK para Python (Boto3) exemplos de código
AWS SDK para Ruby	AWS SDK para Ruby exemplos de código
AWS SDK para Rust	AWS SDK para Rust exemplos de código
SDK da AWS para SAP ABAP	SDK da AWS para SAP ABAP exemplos de código
AWS SDK for Swift	AWS SDK for Swift exemplos de código

Exemplo de disponibilidade

Não consegue encontrar o que precisa? Solicite um exemplo de código usando o link Fornecer feedback na parte inferior desta página.

Exemplos de código para Application Auto Scaling usando AWS SDKs

Os exemplos de código a seguir mostram como usar o Application Auto Scaling com um kit de desenvolvimento AWS de software (SDK).

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar perfis de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando esse serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Exemplos de código

- Exemplos básicos de Application Auto Scaling usando AWS SDKs
  - Ações para o Application Auto Scaling usando AWS SDKs
    - Use DeleteScalingPolicy com um AWS SDK ou CLI
    - Usar DeleteScheduledAction com uma CLI
    - Usar DeregisterScalableTarget com uma CLI
    - Usar DescribeScalableTargets com uma CLI
    - Usar DescribeScalingActivities com uma CLI
    - Use DescribeScalingPolicies com um AWS SDK ou CLI
    - Usar DescribeScheduledActions com uma CLI
    - Usar PutScalingPolicy com uma CLI
    - Usar PutScheduledAction com uma CLI
    - Use RegisterScalableTarget com um AWS SDK ou CLI

Exemplos básicos de Application Auto Scaling usando AWS SDKs

Os exemplos de código a seguir mostram como usar os conceitos básicos do Application Auto AWS SDKs Scaling com.

Exemplos

- Ações para o Application Auto Scaling usando AWS SDKs
  - Use DeleteScalingPolicy com um AWS SDK ou CLI
  - Usar DeleteScheduledAction com uma CLI
  - Usar DeregisterScalableTarget com uma CLI
  - Usar DescribeScalableTargets com uma CLI
  - Usar DescribeScalingActivities com uma CLI
  - Use DescribeScalingPolicies com um AWS SDK ou CLI
  - Usar DescribeScheduledActions com uma CLI
  - Usar PutScalingPolicy com uma CLI
  - Usar PutScheduledAction com uma CLI
  - Use RegisterScalableTarget com um AWS SDK ou CLI

Ações para o Application Auto Scaling usando AWS SDKs

Os exemplos de código a seguir demonstram como realizar ações individuais do Application Auto Scaling com. AWS SDKs Cada exemplo inclui um link para GitHub, onde você pode encontrar instruções para configurar e executar o código.

Os exemplos a seguir incluem apenas as ações mais utilizadas. Para conferir uma lista completa, consulte a Referência de API do Application Auto Scaling.

Exemplos

- Use DeleteScalingPolicy com um AWS SDK ou CLI
- Usar DeleteScheduledAction com uma CLI
- Usar DeregisterScalableTarget com uma CLI
- Usar DescribeScalableTargets com uma CLI
- Usar DescribeScalingActivities com uma CLI
- Use DescribeScalingPolicies com um AWS SDK ou CLI
- Usar DescribeScheduledActions com uma CLI
- Usar PutScalingPolicy com uma CLI
- Usar PutScheduledAction com uma CLI
- Use RegisterScalableTarget com um AWS SDK ou CLI

Use **`DeleteScalingPolicy`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `DeleteScalingPolicy`.

CLI

AWS CLI

Como excluir uma política de escalabilidade

Este exemplo exclui uma política de escalabilidade para a aplicação web do serviço Amazon ECS em execução no cluster padrão.

Comando:

```
aws application-autoscaling delete-scaling-policy --policy-name web-app-cpu-lt-25
 --scalable-dimension ecs:service:DesiredCount --resource-id service/default/web-
app --service-namespace ecs
```

- Para obter detalhes da API, consulte DeleteScalingPolicyem Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.applicationautoscaling.ApplicationAutoScalingClient;
import
 software.amazon.awssdk.services.applicationautoscaling.model.ApplicationAutoScalingExcep
import
 software.amazon.awssdk.services.applicationautoscaling.model.DeleteScalingPolicyRequest;
import
 software.amazon.awssdk.services.applicationautoscaling.model.DeregisterScalableTargetReq
```

```
import
 software.amazon.awssdk.services.applicationautoscaling.model.DescribeScalableTargetsRequ
import
 software.amazon.awssdk.services.applicationautoscaling.model.DescribeScalableTargetsResp
import
 software.amazon.awssdk.services.applicationautoscaling.model.DescribeScalingPoliciesRequ
import
 software.amazon.awssdk.services.applicationautoscaling.model.DescribeScalingPoliciesResp
import
 software.amazon.awssdk.services.applicationautoscaling.model.ScalableDimension;
import
 software.amazon.awssdk.services.applicationautoscaling.model.ServiceNamespace;

/**
 * Before running this Java V2 code example, set up your development environment,
 including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class DisableDynamoDBAutoscaling {
    public static void main(String[] args) {
        final String usage = """

            Usage:
               <tableId> <policyName>\s

            Where:
               tableId - The table Id value (for example, table/Music).\s
               policyName - The name of the policy (for example, $Music5-scaling-
policy).

            """;
        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        ApplicationAutoScalingClient appAutoScalingClient =
 ApplicationAutoScalingClient.builder()
            .region(Region.US_EAST_1)
```

```
            .build();

        ServiceNamespace ns = ServiceNamespace.DYNAMODB;
        ScalableDimension tableWCUs =
ScalableDimension.DYNAMODB_TABLE_WRITE_CAPACITY_UNITS;
        String tableId = args[0];
        String policyName = args[1];

        deletePolicy(appAutoScalingClient, policyName, tableWCUs, ns, tableId);
        verifyScalingPolicies(appAutoScalingClient, tableId, ns, tableWCUs);
        deregisterScalableTarget(appAutoScalingClient, tableId, ns, tableWCUs);
        verifyTarget(appAutoScalingClient, tableId, ns, tableWCUs);
    }

    public static void deletePolicy(ApplicationAutoScalingClient
appAutoScalingClient, String policyName, ScalableDimension tableWCUs,
ServiceNamespace ns, String tableId) {
        try {
            DeleteScalingPolicyRequest delSPRequest =
DeleteScalingPolicyRequest.builder()
                .policyName(policyName)
                .scalableDimension(tableWCUs)
                .serviceNamespace(ns)
                .resourceId(tableId)
                .build();

            appAutoScalingClient.deleteScalingPolicy(delSPRequest);
            System.out.println(policyName +" was deleted successfully.");

        } catch (ApplicationAutoScalingException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
        }
    }

    // Verify that the scaling policy was deleted
    public static void verifyScalingPolicies(ApplicationAutoScalingClient
appAutoScalingClient, String tableId, ServiceNamespace ns, ScalableDimension
tableWCUs) {
        DescribeScalingPoliciesRequest dscRequest =
DescribeScalingPoliciesRequest.builder()
            .scalableDimension(tableWCUs)
            .serviceNamespace(ns)
            .resourceId(tableId)
            .build();
```

```
        DescribeScalingPoliciesResponse response =
 appAutoScalingClient.describeScalingPolicies(dscRequest);
        System.out.println("DescribeScalableTargets result: ");
        System.out.println(response);
    }

    public static void deregisterScalableTarget(ApplicationAutoScalingClient
 appAutoScalingClient, String tableId, ServiceNamespace ns, ScalableDimension
 tableWCUs) {
        try {
            DeregisterScalableTargetRequest targetRequest =
 DeregisterScalableTargetRequest.builder()
                .scalableDimension(tableWCUs)
                .serviceNamespace(ns)
                .resourceId(tableId)
                .build();

            appAutoScalingClient.deregisterScalableTarget(targetRequest);
            System.out.println("The scalable target was deregistered.");

        } catch (ApplicationAutoScalingException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
        }
    }

    public static void verifyTarget(ApplicationAutoScalingClient
 appAutoScalingClient, String tableId, ServiceNamespace ns, ScalableDimension
 tableWCUs) {
        DescribeScalableTargetsRequest dscRequest =
 DescribeScalableTargetsRequest.builder()
            .scalableDimension(tableWCUs)
            .serviceNamespace(ns)
            .resourceIds(tableId)
            .build();

        DescribeScalableTargetsResponse response =
 appAutoScalingClient.describeScalableTargets(dscRequest);
        System.out.println("DescribeScalableTargets result: ");
        System.out.println(response);
    }
}
```

- Para obter detalhes da API, consulte DeleteScalingPolicya Referência AWS SDK for Java 2.x da API.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse cmdlet exclui a política de escalabilidade especificada para um destino escalável do Application Auto Scaling.

```
Remove-AASScalingPolicy -ServiceNamespace AppStream -PolicyName "default-scale-
out" -ResourceId fleet/Test -ScalableDimension appstream:fleet:DesiredCapacity
```

- Para obter detalhes da API, consulte DeleteScalingPolicyem Referência de Ferramentas da AWS para PowerShell cmdlet (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse cmdlet exclui a política de escalabilidade especificada para um destino escalável do Application Auto Scaling.

```
Remove-AASScalingPolicy -ServiceNamespace AppStream -PolicyName "default-scale-
out" -ResourceId fleet/Test -ScalableDimension appstream:fleet:DesiredCapacity
```

- Para obter detalhes da API, consulte DeleteScalingPolicyem Referência de Ferramentas da AWS para PowerShell cmdlet (V5).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando esse serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DeleteScheduledAction`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DeleteScheduledAction`.

CLI

AWS CLI

Para excluir uma ação programada

O `delete-scheduled-action` exemplo a seguir exclui a ação programada especificada da frota Amazon AppStream 2.0 especificada:

```
aws application-autoscaling delete-scheduled-action \
    --service-namespace appstream \
    --scalable-dimension appstream:fleet:DesiredCapacity \
    --resource-id fleet/sample-fleet \
    --scheduled-action-name my-recurring-action
```

Este comando não produz saída.

Para obter mais informações, consulte Escalabilidade programada no Guia do usuário do Application Auto Scaling.

- Para obter detalhes da API, consulte DeleteScheduledActionem Referência de AWS CLI Comandos.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse cmdlet exclui a ação programada especificada para um destino escalável do Application Auto Scaling.

```
Remove-AASScheduledAction -ServiceNamespace AppStream -ScheduledActionName
 WeekDaysFleetScaling -ResourceId fleet/MyFleet -ScalableDimension
 appstream:fleet:DesiredCapacity
```

Saída:

```
Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-AASScheduledAction (DeleteScheduledAction)" on
 target "WeekDaysFleetScaling".
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is
 "Y"): Y
```

- Para obter detalhes da API, consulte DeleteScheduledActionem Referência de Ferramentas da AWS para PowerShell cmdlet (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse cmdlet exclui a ação programada especificada para um destino escalável do Application Auto Scaling.

```
Remove-AASScheduledAction -ServiceNamespace AppStream -ScheduledActionName
 WeekDaysFleetScaling -ResourceId fleet/MyFleet -ScalableDimension
 appstream:fleet:DesiredCapacity
```

Saída:

```
Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-AASScheduledAction (DeleteScheduledAction)" on
 target "WeekDaysFleetScaling".
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is
 "Y"): Y
```

- Para obter detalhes da API, consulte DeleteScheduledActionem Referência de Ferramentas da AWS para PowerShell cmdlet (V5).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando esse serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DeregisterScalableTarget`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DeregisterScalableTarget`.

CLI

AWS CLI

Como cancelar o registro de um destino escalável

Este exemplo cancela o registro de um destino escalável para um serviço do Amazon ECS chamado web-app que está sendo executado no cluster padrão.

Comando:

```
aws application-autoscaling deregister-scalable-target --service-namespace ecs --
scalable-dimension ecs:service:DesiredCount --resource-id service/default/web-app
```

Este exemplo cancela o registro de uma meta escalável para um recurso personalizado. O custom-resource-id arquivo.txt contém uma string que identifica o ID do recurso, que, para um recurso personalizado, é o caminho para o recurso personalizado por meio do endpoint do Amazon API Gateway.

Comando:

```
aws application-autoscaling deregister-scalable-target --
service-namespace custom-resource --scalable-dimension custom-
resource:ResourceType:Property --resource-id file://~/custom-resource-id.txt
```

Conteúdo do custom-resource-id arquivo.txt:

```
https://example.execute-api.us-west-2.amazonaws.com/prod/
scalableTargetDimensions/1-23456789
```

- Para obter detalhes da API, consulte DeregisterScalableTargetem Referência de AWS CLI Comandos.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse cmdlet cancela o registro de um destino escalável do Application Auto Scaling. O cancelamento do registro de um destino escalável exclui as políticas de escalabilidade associadas a ele.

```
Remove-AASScalableTarget -ResourceId fleet/MyFleet -ScalableDimension
 appstream:fleet:DesiredCapacity -ServiceNamespace AppStream
```

Saída:

```
Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-AASScalableTarget (DeregisterScalableTarget)" on
 target "fleet/MyFleet".
```

```
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is
 "Y"): Y
```

- Para obter detalhes da API, consulte DeregisterScalableTargetem Referência de Ferramentas da AWS para PowerShell cmdlet (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse cmdlet cancela o registro de um destino escalável do Application Auto Scaling. O cancelamento do registro de um destino escalável exclui as políticas de escalabilidade associadas a ele.

```
Remove-AASScalableTarget -ResourceId fleet/MyFleet -ScalableDimension
 appstream:fleet:DesiredCapacity -ServiceNamespace AppStream
```

Saída:

```
Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-AASScalableTarget (DeregisterScalableTarget)" on
 target "fleet/MyFleet".
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is
 "Y"): Y
```

- Para obter detalhes da API, consulte DeregisterScalableTargetem Referência de Ferramentas da AWS para PowerShell cmdlet (V5).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando esse serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeScalableTargets`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DescribeScalableTargets`.

CLI

AWS CLI

Como descrever destinos escaláveis

O exemplo `describe-scalable-targets` a seguir descreve os destinos escaláveis para o namespace do serviço `ecs`.

```
aws application-autoscaling describe-scalable-targets \
    --service-namespace ecs
```

Saída:

```
{
    "ScalableTargets": [
        {
            "ServiceNamespace": "ecs",
            "ScalableDimension": "ecs:service:DesiredCount",
            "ResourceId": "service/default/web-app",
            "MinCapacity": 1,
            "MaxCapacity": 10,
            "RoleARN": "arn:aws:iam::123456789012:role/
aws-service-role/ecs.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ECSService",
            "CreationTime": 1462558906.199,
            "SuspendedState": {
                "DynamicScalingOutSuspended": false,
                "ScheduledScalingSuspended": false,
                "DynamicScalingInSuspended": false
            },
            "ScalableTargetARN": "arn:aws:application-autoscaling:us-
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
        }
    ]
}
```

Para obter mais informações, consulte Serviços da AWS que podem ser usados com o Application Auto Scaling no Guia do usuário do Application Auto Scaling.

- Para obter detalhes da API, consulte DescribeScalableTargetsem Referência de AWS CLI Comandos.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: este exemplo fornecerá informações sobre os destinos escaláveis do Application Auto Scaling no namespace especificado.

```
Get-AASScalableTarget -ServiceNamespace "AppStream"
```

Saída:

```
CreationTime      : 11/7/2019 2:30:03 AM
MaxCapacity       : 5
MinCapacity       : 1
ResourceId        : fleet/Test
RoleARN           : arn:aws:iam::012345678912:role/aws-
service-role/appstream.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_AppStreamFleet
ScalableDimension : appstream:fleet:DesiredCapacity
ServiceNamespace  : appstream
SuspendedState    : Amazon.ApplicationAutoScaling.Model.SuspendedState
```

- Para obter detalhes da API, consulte DescribeScalableTargetsem Referência de Ferramentas da AWS para PowerShell cmdlet (V4).

Ferramentas para PowerShell V5

Exemplo 1: este exemplo fornecerá informações sobre os destinos escaláveis do Application Auto Scaling no namespace especificado.

```
Get-AASScalableTarget -ServiceNamespace "AppStream"
```

Saída:

```
CreationTime      : 11/7/2019 2:30:03 AM
MaxCapacity       : 5
MinCapacity       : 1
ResourceId        : fleet/Test
RoleARN           : arn:aws:iam::012345678912:role/aws-
service-role/appstream.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_AppStreamFleet
ScalableDimension : appstream:fleet:DesiredCapacity
```

```
ServiceNamespace  : appstream
SuspendedState    : Amazon.ApplicationAutoScaling.Model.SuspendedState
```

- Para obter detalhes da API, consulte DescribeScalableTargetsem Referência de Ferramentas da AWS para PowerShell cmdlet (V5).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando esse serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeScalingActivities`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DescribeScalingActivities`.

CLI

AWS CLI

Exemplo 1: como descrever as atividades de escalabilidade para o serviço do Amazon ECS especificado

O exemplo `describe-scaling-activities` a seguir descreve as atividades de escalabilidade de um serviço do Amazon ECS chamado `web-app` que está sendo executado no cluster `default`. A saída mostra uma atividade de ajuste de escala iniciada por uma política de ajuste de escala.

```
aws application-autoscaling describe-scaling-activities \
    --service-namespace ecs \
    --resource-id service/default/web-app
```

Saída:

```
{
    "ScalingActivities": [
        {
            "ScalableDimension": "ecs:service:DesiredCount",
            "Description": "Setting desired count to 1.",
            "ResourceId": "service/default/web-app",
            "ActivityId": "e6c5f7d1-dbbb-4a3f-89b2-51f33e766399",
            "StartTime": 1462575838.171,
            "ServiceNamespace": "ecs",
```

```
            "EndTime": 1462575872.111,
            "Cause": "monitor alarm web-app-cpu-lt-25 in state ALARM triggered
 policy web-app-cpu-lt-25",
            "StatusMessage": "Successfully set desired count to 1. Change
 successfully fulfilled by ecs.",
            "StatusCode": "Successful"
        }
    ]
}
```

Para obter mais informações, consulte Atividades de escalabilidade para o Application Auto Scaling no Guia do usuário do Application Auto Scaling.

Exemplo 2: descrever as ações de ajuste de escala da tabela do DynamoDB especificada

O exemplo `describe-scaling-activities` a seguir descreve as atividades de ajuste de escala de uma tabela do DynamoDB chamada `TestTable`. O resultado mostra as atividades de ajuste de escala iniciadas por duas ações agendadas diferentes.

```
aws application-autoscaling describe-scaling-activities \
    --service-namespace dynamodb \
    --resource-id table/TestTable
```

Saída:

```
{
    "ScalingActivities": [
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Description": "Setting write capacity units to 10.",
            "ResourceId": "table/my-table",
            "ActivityId": "4d1308c0-bbcf-4514-a673-b0220ae38547",
            "StartTime": 1561574415.086,
            "ServiceNamespace": "dynamodb",
            "EndTime": 1561574449.51,
            "Cause": "maximum capacity was set to 10",
            "StatusMessage": "Successfully set write capacity units to 10. Change
 successfully fulfilled by dynamodb.",
            "StatusCode": "Successful"
        },
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Description": "Setting min capacity to 5 and max capacity to 10",
```

```
            "ResourceId": "table/my-table",
            "ActivityId": "f2b7847b-721d-4e01-8ef0-0c8d3bacc1c7",
            "StartTime": 1561574414.644,
            "ServiceNamespace": "dynamodb",
            "Cause": "scheduled action name my-second-scheduled-action was
 triggered",
            "StatusMessage": "Successfully set min capacity to 5 and max capacity
 to 10",
            "StatusCode": "Successful"
        },
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Description": "Setting write capacity units to 15.",
            "ResourceId": "table/my-table",
            "ActivityId": "d8ea4de6-9eaa-499f-b466-2cc5e681ba8b",
            "StartTime": 1561574108.904,
            "ServiceNamespace": "dynamodb",
            "EndTime": 1561574140.255,
            "Cause": "minimum capacity was set to 15",
            "StatusMessage": "Successfully set write capacity units to 15. Change
 successfully fulfilled by dynamodb.",
            "StatusCode": "Successful"
        },
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Description": "Setting min capacity to 15 and max capacity to 20",
            "ResourceId": "table/my-table",
            "ActivityId": "3250fd06-6940-4e8e-bb1f-d494db7554d2",
            "StartTime": 1561574108.512,
            "ServiceNamespace": "dynamodb",
            "Cause": "scheduled action name my-first-scheduled-action was
 triggered",
            "StatusMessage": "Successfully set min capacity to 15 and max
 capacity to 20",
            "StatusCode": "Successful"
        }
    ]
}
```

Para obter mais informações, consulte Atividades de escalabilidade para o Application Auto Scaling no Guia do usuário do Application Auto Scaling.

- Para obter detalhes da API, consulte DescribeScalingActivitiesem Referência de AWS CLI Comandos.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: fornece informações descritivas sobre as atividades de escalabilidade no namespace especificado das últimas seis semanas.

```
Get-AASScalingActivity -ServiceNamespace AppStream
```

Saída:

```
ActivityId        : 2827409f-b639-4cdb-a957-8055d5d07434
Cause             : monitor alarm Appstream2-MyFleet-default-scale-in-Alarm in
 state ALARM triggered policy default-scale-in
Description       : Setting desired capacity to 2.
Details           :
EndTime           : 12/14/2019 11:32:49 AM
ResourceId        : fleet/MyFleet
ScalableDimension : appstream:fleet:DesiredCapacity
ServiceNamespace  : appstream
StartTime         : 12/14/2019 11:32:14 AM
StatusCode        : Successful
StatusMessage     : Successfully set desired capacity to 2. Change successfully
 fulfilled by appstream.
```

- Para obter detalhes da API, consulte DescribeScalingActivitiesem Referência de Ferramentas da AWS para PowerShell cmdlet (V4).

Ferramentas para PowerShell V5

Exemplo 1: fornece informações descritivas sobre as atividades de escalabilidade no namespace especificado das últimas seis semanas.

```
Get-AASScalingActivity -ServiceNamespace AppStream
```

Saída:

```
ActivityId        : 2827409f-b639-4cdb-a957-8055d5d07434
Cause             : monitor alarm Appstream2-MyFleet-default-scale-in-Alarm in
 state ALARM triggered policy default-scale-in
Description       : Setting desired capacity to 2.
Details           :
EndTime           : 12/14/2019 11:32:49 AM
```

```
ResourceId        : fleet/MyFleet
ScalableDimension : appstream:fleet:DesiredCapacity
ServiceNamespace  : appstream
StartTime         : 12/14/2019 11:32:14 AM
StatusCode        : Successful
StatusMessage     : Successfully set desired capacity to 2. Change successfully
 fulfilled by appstream.
```

- Para obter detalhes da API, consulte DescribeScalingActivitiesem Referência de Ferramentas da AWS para PowerShell cmdlet (V5).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando esse serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`DescribeScalingPolicies`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `DescribeScalingPolicies`.

CLI

AWS CLI

Como descrever políticas de escalabilidade

Este exemplo de comando descreve as políticas de ajuste de escala para o namespace do serviço ecs.

Comando:

```
aws application-autoscaling describe-scaling-policies --service-namespace ecs
```

Saída:

```
{
    "ScalingPolicies": [
        {
            "PolicyName": "web-app-cpu-gt-75",
            "ScalableDimension": "ecs:service:DesiredCount",
            "ResourceId": "service/default/web-app",
            "CreationTime": 1462561899.23,
            "StepScalingPolicyConfiguration": {
```

```
                "Cooldown": 60,
                "StepAdjustments": [
                    {
                        "ScalingAdjustment": 200,
                        "MetricIntervalLowerBound": 0.0
                    }
                ],
                "AdjustmentType": "PercentChangeInCapacity"
            },
            "PolicyARN": "arn:aws:autoscaling:us-
west-2:012345678910:scalingPolicy:6d8972f3-efc8-437c-92d1-6270f29a66e7:resource/
ecs/service/default/web-app:policyName/web-app-cpu-gt-75",
            "PolicyType": "StepScaling",
            "Alarms": [
                {
                    "AlarmName": "web-app-cpu-gt-75",
                    "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:web-app-cpu-gt-75"
                }
            ],
            "ServiceNamespace": "ecs"
        },
        {
            "PolicyName": "web-app-cpu-lt-25",
            "ScalableDimension": "ecs:service:DesiredCount",
            "ResourceId": "service/default/web-app",
            "CreationTime": 1462562575.099,
            "StepScalingPolicyConfiguration": {
                "Cooldown": 1,
                "StepAdjustments": [
                    {
                        "ScalingAdjustment": -50,
                        "MetricIntervalUpperBound": 0.0
                    }
                ],
                "AdjustmentType": "PercentChangeInCapacity"
            },
            "PolicyARN": "arn:aws:autoscaling:us-
west-2:012345678910:scalingPolicy:6d8972f3-efc8-437c-92d1-6270f29a66e7:resource/
ecs/service/default/web-app:policyName/web-app-cpu-lt-25",
            "PolicyType": "StepScaling",
            "Alarms": [
                {
                    "AlarmName": "web-app-cpu-lt-25",
```

```
                    "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:web-app-cpu-lt-25"
                }
            ],
            "ServiceNamespace": "ecs"
        }
    ]
}
```

- Para obter detalhes da API, consulte DescribeScalingPoliciesem Referência de AWS CLI Comandos.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse cmdlet descreve as políticas de escalabilidade do Application Auto Scaling para o namespace de serviço especificado.

```
Get-AASScalingPolicy -ServiceNamespace AppStream
```

Saída:

```
Alarms                                  : {Appstream2-LabFleet-default-scale-
out-Alarm}
CreationTime                            : 9/3/2019 2:48:15 AM
PolicyARN                               : arn:aws:autoscaling:us-
west-2:012345678912:scalingPolicy:5659b069-b5cd-4af1-9f7f-3e956d36233e:resource/
appstream/fleet/LabFleet:
                                            policyName/default-scale-out
PolicyName                              : default-scale-out
PolicyType                              : StepScaling
ResourceId                              : fleet/LabFleet
ScalableDimension                       : appstream:fleet:DesiredCapacity
ServiceNamespace                        : appstream
StepScalingPolicyConfiguration          :
 Amazon.ApplicationAutoScaling.Model.StepScalingPolicyConfiguration
TargetTrackingScalingPolicyConfiguration :

Alarms                                  : {Appstream2-LabFleet-default-scale-in-
Alarm}
CreationTime                            : 9/3/2019 2:48:15 AM
```

```
PolicyARN                                   : arn:aws:autoscaling:us-
west-2:012345678912:scalingPolicy:5659b069-b5cd-4af1-9f7f-3e956d36233e:resource/
appstream/fleet/LabFleet:
                                              policyName/default-scale-in
PolicyName                                  : default-scale-in
PolicyType                                  : StepScaling
ResourceId                                  : fleet/LabFleet
ScalableDimension                           : appstream:fleet:DesiredCapacity
ServiceNamespace                            : appstream
StepScalingPolicyConfiguration              :
 Amazon.ApplicationAutoScaling.Model.StepScalingPolicyConfiguration
TargetTrackingScalingPolicyConfiguration :
```

- Para obter detalhes da API, consulte DescribeScalingPoliciesem Referência de Ferramentas da AWS para PowerShell cmdlet (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse cmdlet descreve as políticas de escalabilidade do Application Auto Scaling para o namespace de serviço especificado.

```
Get-AASScalingPolicy -ServiceNamespace AppStream
```

Saída:

```
Alarms                                      : {Appstream2-LabFleet-default-scale-
out-Alarm}
CreationTime                                : 9/3/2019 2:48:15 AM
PolicyARN                                   : arn:aws:autoscaling:us-
west-2:012345678912:scalingPolicy:5659b069-b5cd-4af1-9f7f-3e956d36233e:resource/
appstream/fleet/LabFleet:
                                              policyName/default-scale-out
PolicyName                                  : default-scale-out
PolicyType                                  : StepScaling
ResourceId                                  : fleet/LabFleet
ScalableDimension                           : appstream:fleet:DesiredCapacity
ServiceNamespace                            : appstream
StepScalingPolicyConfiguration              :
 Amazon.ApplicationAutoScaling.Model.StepScalingPolicyConfiguration
TargetTrackingScalingPolicyConfiguration :

Alarms                                      : {Appstream2-LabFleet-default-scale-in-
Alarm}
```

```
CreationTime                                : 9/3/2019 2:48:15 AM
PolicyARN                                   : arn:aws:autoscaling:us-
west-2:012345678912:scalingPolicy:5659b069-b5cd-4af1-9f7f-3e956d36233e:resource/
appstream/fleet/LabFleet:
                                              policyName/default-scale-in
PolicyName                                  : default-scale-in
PolicyType                                  : StepScaling
ResourceId                                  : fleet/LabFleet
ScalableDimension                           : appstream:fleet:DesiredCapacity
ServiceNamespace                            : appstream
StepScalingPolicyConfiguration              :
 Amazon.ApplicationAutoScaling.Model.StepScalingPolicyConfiguration
TargetTrackingScalingPolicyConfiguration :
```

- Para obter detalhes da API, consulte DescribeScalingPoliciesem Referência de Ferramentas da AWS para PowerShell cmdlet (V5).

Rust

SDK para Rust

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
async fn show_policies(client: &Client) -> Result<(), Error> {
    let response = client
        .describe_scaling_policies()
        .service_namespace(ServiceNamespace::Ec2)
        .send()
        .await?;
    println!("Auto Scaling Policies:");
    for policy in response.scaling_policies() {
        println!("{:?}\n", policy);
    }
    println!("Next token: {:?}", response.next_token());

    Ok(())
}
```

- Para obter detalhes da API, consulte a DescribeScalingPoliciesreferência da API AWS SDK for Rust.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando esse serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeScheduledActions`** com uma CLI

Os exemplos de código a seguir mostram como usar o `DescribeScheduledActions`.

CLI

AWS CLI

Como descrever ações programadas

O exemplo `describe-scheduled-actions` a seguir exibe detalhes das ações agendadas para o namespace de serviço especificado:

```
aws application-autoscaling describe-scheduled-actions \
    --service-namespace dynamodb
```

Saída:

```
{
    "ScheduledActions": [
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Schedule": "at(2019-05-20T18:35:00)",
            "ResourceId": "table/my-table",
            "CreationTime": 1561571888.361,
            "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledAction:2d36aa3b-cdf9-4565-
b290-81db519b227d:resource/dynamodb/table/my-table:scheduledActionName/my-first-
scheduled-action",
            "ScalableTargetAction": {
                "MinCapacity": 15,
```

```
                "MaxCapacity": 20
            },
            "ScheduledActionName": "my-first-scheduled-action",
            "ServiceNamespace": "dynamodb"
        },
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Schedule": "at(2019-05-20T18:40:00)",
            "ResourceId": "table/my-table",
            "CreationTime": 1561571946.021,
            "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledAction:2d36aa3b-cdf9-4565-
b290-81db519b227d:resource/dynamodb/table/my-table:scheduledActionName/my-second-
scheduled-action",
            "ScalableTargetAction": {
                "MinCapacity": 5,
                "MaxCapacity": 10
            },
            "ScheduledActionName": "my-second-scheduled-action",
            "ServiceNamespace": "dynamodb"
        }
    ]
}
```

Para obter mais informações, consulte Escalabilidade programada no Guia do usuário do Application Auto Scaling.

- Para obter detalhes da API, consulte DescribeScheduledActionsem Referência de AWS CLI Comandos.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse cmdlet lista as ações programadas para o grupo do Auto Scaling que ainda não foram executadas ou que ainda não atingiram o horário de término.

```
Get-AASScheduledAction -ServiceNamespace AppStream
```

Saída:

```
CreationTime         : 12/22/2019 9:25:52 AM
```

```
EndTime              : 1/1/0001 12:00:00 AM
ResourceId           : fleet/MyFleet
ScalableDimension    : appstream:fleet:DesiredCapacity
ScalableTargetAction : Amazon.ApplicationAutoScaling.Model.ScalableTargetAction
Schedule             : cron(0 0 8 ? * MON-FRI *)
ScheduledActionARN   : arn:aws:autoscaling:us-
west-2:012345678912:scheduledAction:4897ca24-3caa-4bf1-8484-851a089b243c:resource/
appstream/fleet/MyFleet:scheduledActionName
                       /WeekDaysFleetScaling
ScheduledActionName  : WeekDaysFleetScaling
ServiceNamespace     : appstream
StartTime            : 1/1/0001 12:00:00 AM
```

- Para obter detalhes da API, consulte DescribeScheduledActionsem Referência de Ferramentas da AWS para PowerShell cmdlet (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse cmdlet lista as ações programadas para o grupo do Auto Scaling que ainda não foram executadas ou que ainda não atingiram o horário de término.

```
Get-AASScheduledAction -ServiceNamespace AppStream
```

Saída:

```
CreationTime         : 12/22/2019 9:25:52 AM
EndTime              : 1/1/0001 12:00:00 AM
ResourceId           : fleet/MyFleet
ScalableDimension    : appstream:fleet:DesiredCapacity
ScalableTargetAction : Amazon.ApplicationAutoScaling.Model.ScalableTargetAction
Schedule             : cron(0 0 8 ? * MON-FRI *)
ScheduledActionARN   : arn:aws:autoscaling:us-
west-2:012345678912:scheduledAction:4897ca24-3caa-4bf1-8484-851a089b243c:resource/
appstream/fleet/MyFleet:scheduledActionName
                       /WeekDaysFleetScaling
ScheduledActionName  : WeekDaysFleetScaling
ServiceNamespace     : appstream
StartTime            : 1/1/0001 12:00:00 AM
```

- Para obter detalhes da API, consulte DescribeScheduledActionsem Referência de Ferramentas da AWS para PowerShell cmdlet (V5).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando esse serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`PutScalingPolicy`** com uma CLI

Os exemplos de código a seguir mostram como usar o `PutScalingPolicy`.

CLI

AWS CLI

Exemplo 1: como aplicar uma política de escalabilidade com monitoramento do objetivo com uma especificação de métrica predefinida

O exemplo `put-scaling-policy` a seguir aplica uma política de ajuste de escala de rastreamento de destino com uma especificação de métrica predefinida a um serviço do ECS chamado web-app no cluster padrão. A política mantém a utilização média da CPU do serviço em 75%, com períodos de espera de aumento e redução de escala horizontalmente de 60 segundos. A saída contém os nomes ARNs e dos dois CloudWatch alarmes criados em seu nome.

```
aws application-autoscaling put-scaling-policy --service-namespace ecs \
--scalable-dimension ecs:service:DesiredCount \
--resource-id service/default/web-app \
--policy-name cpu75-target-tracking-scaling-policy --policy-
type TargetTrackingScaling \
--target-tracking-scaling-policy-configuration file://config.json
```

Este exemplo pressupõe que você tenha um arquivo config.json no diretório atual com o seguinte conteúdo:

```
{
     "TargetValue": 75.0,
     "PredefinedMetricSpecification": {
         "PredefinedMetricType": "ECSServiceAverageCPUUtilization"
     },
     "ScaleOutCooldown": 60,
    "ScaleInCooldown": 60
}
```

Saída:

```
{
    "PolicyARN": "arn:aws:autoscaling:us-
west-2:012345678910:scalingPolicy:6d8972f3-efc8-437c-92d1-6270f29a66e7:resource/
ecs/service/default/web-app:policyName/cpu75-target-tracking-scaling-policy",
    "Alarms": [
        {
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:TargetTracking-service/default/web-app-AlarmHigh-
d4f0770c-b46e-434a-a60f-3b36d653feca",
            "AlarmName": "TargetTracking-service/default/web-app-AlarmHigh-
d4f0770c-b46e-434a-a60f-3b36d653feca"
        },
        {
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:TargetTracking-service/default/web-app-
AlarmLow-1b437334-d19b-4a63-a812-6c67aaf2910d",
            "AlarmName": "TargetTracking-service/default/web-app-
AlarmLow-1b437334-d19b-4a63-a812-6c67aaf2910d"
        }
    ]
}
```

Exemplo 2: como aplicar uma política de escalabilidade com monitoramento do objetivo com uma especificação de métrica personalizada

O exemplo `put-scaling-policy` a seguir aplica uma política de ajuste de escala de rastreamento de destino com uma especificação de métrica personalizada a um serviço do ECS chamado web-app no cluster padrão. A política mantém a utilização média do serviço em 75%, com períodos de espera de aumento e redução de escala horizontalmente de 60 segundos. A saída contém os nomes ARNs e dos dois CloudWatch alarmes criados em seu nome.

```
aws application-autoscaling put-scaling-policy --service-namespace ecs \
--scalable-dimension ecs:service:DesiredCount \
--resource-id service/default/web-app \
--policy-name cms75-target-tracking-scaling-policy
--policy-type TargetTrackingScaling \
--target-tracking-scaling-policy-configuration file://config.json
```

Este exemplo pressupõe que você tenha um arquivo config.json no diretório atual com o seguinte conteúdo:

```
{
    "TargetValue":75.0,
    "CustomizedMetricSpecification":{
        "MetricName":"MyUtilizationMetric",
        "Namespace":"MyNamespace",
        "Dimensions": [
            {
                "Name":"MyOptionalMetricDimensionName",
                "Value":"MyOptionalMetricDimensionValue"
            }
        ],
        "Statistic":"Average",
        "Unit":"Percent"
    },
    "ScaleOutCooldown": 60,
    "ScaleInCooldown": 60
}
```

Saída:

```
{
    "PolicyARN": "arn:aws:autoscaling:us-west-2:012345678910:scalingPolicy:
 8784a896-b2ba-47a1-b08c-27301cc499a1:resource/ecs/service/default/web-
app:policyName/cms75-target-tracking-scaling-policy",
    "Alarms": [
        {
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:TargetTracking-service/default/web-app-
AlarmHigh-9bc77b56-0571-4276-ba0f-d4178882e0a0",
            "AlarmName": "TargetTracking-service/default/web-app-
AlarmHigh-9bc77b56-0571-4276-ba0f-d4178882e0a0"
        },
        {
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:TargetTracking-service/default/web-app-
AlarmLow-9b6ad934-6d37-438e-9e05-02836ddcbdc4",
            "AlarmName": "TargetTracking-service/default/web-app-
AlarmLow-9b6ad934-6d37-438e-9e05-02836ddcbdc4"
        }
    ]
```

```
}
```

Exemplo 3: como aplicar uma política de escalabilidade com monitoramento do objetivo somente para expansão

O exemplo `put-scaling-policy` a seguir aplica uma política de ajuste de escala de rastreamento de destino a um serviço do ECS chamado `web-app` no cluster padrão. A política é usada para aumentar a escala horizontalmente o serviço ECS quando a métrica `RequestCountPerTarget` do Application Load Balancer excede o limite. A saída contém o ARN e o nome do CloudWatch alarme criado em seu nome.

```
aws application-autoscaling put-scaling-policy \
    --service-namespace ecs \
    --scalable-dimension ecs:service:DesiredCount \
    --resource-id service/default/web-app \
    --policy-name alb-scale-out-target-tracking-scaling-policy \
    --policy-type TargetTrackingScaling \
    --target-tracking-scaling-policy-configuration file://config.json
```

Conteúdo de `config.json`:

```
{
     "TargetValue": 1000.0,
     "PredefinedMetricSpecification": {
         "PredefinedMetricType": "ALBRequestCountPerTarget",
         "ResourceLabel": "app/EC2Co-EcsEl-1TKLTMITMM0EO/f37c06a68c1748aa/
targetgroup/EC2Co-Defau-LDNM7Q3ZH1ZN/6d4ea56ca2d6a18d"
     },
     "ScaleOutCooldown": 60,
    "ScaleInCooldown": 60,
    "DisableScaleIn": true
}
```

Saída:

```
{
    "PolicyARN": "arn:aws:autoscaling:us-
west-2:123456789012:scalingPolicy:6d8972f3-efc8-437c-92d1-6270f29a66e7:resource/
ecs/service/default/web-app:policyName/alb-scale-out-target-tracking-scaling-
policy",
    "Alarms": [
        {
```

```
            "AlarmName": "TargetTracking-service/default/web-app-AlarmHigh-
d4f0770c-b46e-434a-a60f-3b36d653feca",
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-service/default/web-app-AlarmHigh-
d4f0770c-b46e-434a-a60f-3b36d653feca"
        }
    ]
}
```

Para obter mais informações, consulte Políticas de ajuste de escala de rastreamento de destino para o Application Auto Scaling no Guia do usuário do AWS Application Auto Scaling.

- Para obter detalhes da API, consulte PutScalingPolicyem Referência de AWS CLI Comandos.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse cmdlet cria ou atualiza uma política para um destino escalável do Application Auto Scaling. Cada destino escalável é identificado por um namespace de serviço, ID de recurso e dimensão escalável.

```
Set-AASScalingPolicy -ServiceNamespace AppStream -PolicyName ASFleetScaleInPolicy
 -PolicyType StepScaling  -ResourceId fleet/MyFleet -ScalableDimension
 appstream:fleet:DesiredCapacity -StepScalingPolicyConfiguration_AdjustmentType
 ChangeInCapacity -StepScalingPolicyConfiguration_Cooldown 360
 -StepScalingPolicyConfiguration_MetricAggregationType Average -
StepScalingPolicyConfiguration_StepAdjustments @{ScalingAdjustment = -1;
 MetricIntervalUpperBound = 0}
```

Saída:

```
Alarms    PolicyARN
------    ---------
{}        arn:aws:autoscaling:us-
west-2:012345678912:scalingPolicy:4897ca24-3caa-4bf1-8484-851a089b243c:resource/
appstream/fleet/MyFleet:policyName/ASFleetScaleInPolicy
```

- Para obter detalhes da API, consulte PutScalingPolicyem Referência de Ferramentas da AWS para PowerShell cmdlet (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse cmdlet cria ou atualiza uma política para um destino escalável do Application Auto Scaling. Cada destino escalável é identificado por um namespace de serviço, ID de recurso e dimensão escalável.

```
Set-AASScalingPolicy -ServiceNamespace AppStream -PolicyName ASFleetScaleInPolicy
 -PolicyType StepScaling  -ResourceId fleet/MyFleet -ScalableDimension
 appstream:fleet:DesiredCapacity -StepScalingPolicyConfiguration_AdjustmentType
 ChangeInCapacity -StepScalingPolicyConfiguration_Cooldown 360
 -StepScalingPolicyConfiguration_MetricAggregationType Average -
StepScalingPolicyConfiguration_StepAdjustments @{ScalingAdjustment = -1;
 MetricIntervalUpperBound = 0}
```

Saída:

```
Alarms    PolicyARN
------    ---------
{}        arn:aws:autoscaling:us-
west-2:012345678912:scalingPolicy:4897ca24-3caa-4bf1-8484-851a089b243c:resource/
appstream/fleet/MyFleet:policyName/ASFleetScaleInPolicy
```

- Para obter detalhes da API, consulte PutScalingPolicyem Referência de Ferramentas da AWS para PowerShell cmdlet (V5).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando esse serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`PutScheduledAction`** com uma CLI

Os exemplos de código a seguir mostram como usar o `PutScheduledAction`.

CLI

AWS CLI

Como adicionar uma ação programada a uma tabela do DynamoDB

Este exemplo adiciona uma ação programada a uma tabela do DynamoDB TestTable chamada para escalar em uma programação recorrente. Na programação especificada (todos

os dias às 12h15 UTC), se a capacidade atual estiver abaixo do valor especificado para MinCapacity, o Application Auto Scaling se expande até o valor especificado por. MinCapacity

Comando:

```
aws application-autoscaling put-scheduled-action --service-
namespace dynamodb --scheduled-action-name my-recurring-action --
schedule "cron(15 12 * * ? *)" --resource-id table/TestTable --
scalable-dimension dynamodb:table:WriteCapacityUnits --scalable-target-
action MinCapacity=6
```

Para obter mais informações, consulte Escalabilidade programada no Guia do usuário do Application Auto Scaling.

- Para obter detalhes da API, consulte PutScheduledActionem Referência de AWS CLI Comandos.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse cmdlet cria ou atualiza uma ação programada para um destino escalável do Application Auto Scaling. Cada destino escalável é identificado por um namespace de serviço, ID de recurso e dimensão escalável.

```
Set-AASScheduledAction -ServiceNamespace AppStream -ResourceId fleet/
MyFleet -Schedule "cron(0 0 8 ? * MON-FRI *)" -ScalableDimension
 appstream:fleet:DesiredCapacity -ScheduledActionName WeekDaysFleetScaling -
ScalableTargetAction_MinCapacity 5 -ScalableTargetAction_MaxCapacity 10
```

- Para obter detalhes da API, consulte PutScheduledActionem Referência de Ferramentas da AWS para PowerShell cmdlet (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse cmdlet cria ou atualiza uma ação programada para um destino escalável do Application Auto Scaling. Cada destino escalável é identificado por um namespace de serviço, ID de recurso e dimensão escalável.

```
Set-AASScheduledAction -ServiceNamespace AppStream -ResourceId fleet/
MyFleet -Schedule "cron(0 0 8 ? * MON-FRI *)" -ScalableDimension
```

```
 appstream:fleet:DesiredCapacity -ScheduledActionName WeekDaysFleetScaling -
ScalableTargetAction_MinCapacity 5 -ScalableTargetAction_MaxCapacity 10
```

- Para obter detalhes da API, consulte PutScheduledActionem Referência de Ferramentas da AWS para PowerShell cmdlet (V5).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando esse serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`RegisterScalableTarget`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `RegisterScalableTarget`.

CLI

AWS CLI

Exemplo 1: como registrar um serviço do ECS como um destino escalável

O exemplo `register-scalable-target` a seguir inscreve um serviço do Amazon ECS com o Application Auto Scaling. Ele também adiciona uma tag com o nome `environment` e o valor `production` da chave ao destino escalável.

```
aws application-autoscaling register-scalable-target \
    --service-namespace ecs \
    --scalable-dimension ecs:service:DesiredCount \
    --resource-id service/default/web-app \
    --min-capacity 1 --max-capacity 10 \
    --tags environment=production
```

Saída:

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:us-
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Para obter exemplos de outros AWS serviços e recursos personalizados, consulte os tópicos em AWS serviços que você pode usar com o Application Auto Scaling no Guia do Usuário do Application Auto Scaling.

Exemplo 2: suspender as atividades de escalabilidade de um destino escalável

O exemplo `register-scalable-target` a seguir suspende as atividades de ajuste de escala de um destino escalável existente.

```
aws application-autoscaling register-scalable-target \
    --service-namespace dynamodb \
    --scalable-dimension dynamodb:table:ReadCapacityUnits \
    --resource-id table/my-table \
    --suspended-
state DynamicScalingInSuspended=true,DynamicScalingOutSuspended=true,ScheduledScalingSusp
```

Saída:

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:us-
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Para obter mais informações, consulte Como suspender e retomar o ajuste de escala do Application Auto Scaling no Guia do usuário do Application Auto Scaling.

Exemplo 3: retomar as atividades de escalabilidade para um destino escalável

O exemplo `register-scalable-target` a seguir retoma as atividades de ajuste de escala de um destino escalável existente.

```
aws application-autoscaling register-scalable-target \
    --service-namespace dynamodb \
    --scalable-dimension dynamodb:table:ReadCapacityUnits \
    --resource-id table/my-table \
    --suspended-
state DynamicScalingInSuspended=false,DynamicScalingOutSuspended=false,ScheduledScalingSu
```

Saída:

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:us-
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Para obter mais informações, consulte Como suspender e retomar o ajuste de escala do Application Auto Scaling no Guia do usuário do Application Auto Scaling.

- Para obter detalhes da API, consulte RegisterScalableTargetem Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.applicationautoscaling.ApplicationAutoScalingClient;
import
 software.amazon.awssdk.services.applicationautoscaling.model.ApplicationAutoScalingExcep
import
 software.amazon.awssdk.services.applicationautoscaling.model.DescribeScalableTargetsRequ
import
 software.amazon.awssdk.services.applicationautoscaling.model.DescribeScalableTargetsResp
import
 software.amazon.awssdk.services.applicationautoscaling.model.DescribeScalingPoliciesRequ
import
 software.amazon.awssdk.services.applicationautoscaling.model.DescribeScalingPoliciesResp
import software.amazon.awssdk.services.applicationautoscaling.model.PolicyType;
import
 software.amazon.awssdk.services.applicationautoscaling.model.PredefinedMetricSpecificati
import
 software.amazon.awssdk.services.applicationautoscaling.model.PutScalingPolicyRequest;
import
 software.amazon.awssdk.services.applicationautoscaling.model.RegisterScalableTargetReque
import
 software.amazon.awssdk.services.applicationautoscaling.model.ScalingPolicy;
import
 software.amazon.awssdk.services.applicationautoscaling.model.ServiceNamespace;
import
 software.amazon.awssdk.services.applicationautoscaling.model.ScalableDimension;
```

```
import software.amazon.awssdk.services.applicationautoscaling.model.MetricType;
import
 software.amazon.awssdk.services.applicationautoscaling.model.TargetTrackingScalingPolicy
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development environment,
 including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class EnableDynamoDBAutoscaling {
    public static void main(String[] args) {
        final String usage = """

            Usage:
               <tableId> <roleARN> <policyName>\s

            Where:
               tableId - The table Id value (for example, table/Music).
               roleARN - The ARN of the role that has ApplicationAutoScaling
 permissions.
               policyName - The name of the policy to create.

            """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        System.out.println("This example registers an Amazon DynamoDB table,
 which is the resource to scale.");
        String tableId = args[0];
        String roleARN = args[1];
        String policyName = args[2];
        ServiceNamespace ns = ServiceNamespace.DYNAMODB;
        ScalableDimension tableWCUs =
 ScalableDimension.DYNAMODB_TABLE_WRITE_CAPACITY_UNITS;
        ApplicationAutoScalingClient appAutoScalingClient =
 ApplicationAutoScalingClient.builder()
```

```
            .region(Region.US_EAST_1)
            .build();

        registerScalableTarget(appAutoScalingClient, tableId, roleARN, ns,
tableWCUs);
        verifyTarget(appAutoScalingClient, tableId, ns, tableWCUs);
        configureScalingPolicy(appAutoScalingClient, tableId, ns, tableWCUs,
policyName);
    }

    public static void registerScalableTarget(ApplicationAutoScalingClient
appAutoScalingClient, String tableId, String roleARN, ServiceNamespace ns,
ScalableDimension tableWCUs) {
        try {
            RegisterScalableTargetRequest targetRequest =
RegisterScalableTargetRequest.builder()
                .serviceNamespace(ns)
                .scalableDimension(tableWCUs)
                .resourceId(tableId)
                .roleARN(roleARN)
                .minCapacity(5)
                .maxCapacity(10)
                .build();

            appAutoScalingClient.registerScalableTarget(targetRequest);
            System.out.println("You have registered " + tableId);

        } catch (ApplicationAutoScalingException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
        }
    }

    // Verify that the target was created.
    public static void verifyTarget(ApplicationAutoScalingClient
appAutoScalingClient, String tableId, ServiceNamespace ns, ScalableDimension
tableWCUs) {
        DescribeScalableTargetsRequest dscRequest =
DescribeScalableTargetsRequest.builder()
            .scalableDimension(tableWCUs)
            .serviceNamespace(ns)
            .resourceIds(tableId)
            .build();
```

```
        DescribeScalableTargetsResponse response =
appAutoScalingClient.describeScalableTargets(dscRequest);
        System.out.println("DescribeScalableTargets result: ");
        System.out.println(response);
    }

    // Configure a scaling policy.
    public static void configureScalingPolicy(ApplicationAutoScalingClient
appAutoScalingClient, String tableId, ServiceNamespace ns, ScalableDimension
tableWCUs, String policyName) {
        // Check if the policy exists before creating a new one.
        DescribeScalingPoliciesResponse describeScalingPoliciesResponse =
appAutoScalingClient.describeScalingPolicies(DescribeScalingPoliciesRequest.builder()
            .serviceNamespace(ns)
            .resourceId(tableId)
            .scalableDimension(tableWCUs)
            .build());

        if (!describeScalingPoliciesResponse.scalingPolicies().isEmpty()) {
            // If policies exist, consider updating an existing policy instead of
creating a new one.
            System.out.println("Policy already exists. Consider updating it
instead.");
            List<ScalingPolicy> polList =
describeScalingPoliciesResponse.scalingPolicies();
            for (ScalingPolicy pol : polList) {
                System.out.println("Policy name:" +pol.policyName());
            }
        } else {
            // If no policies exist, proceed with creating a new policy.
            PredefinedMetricSpecification specification =
PredefinedMetricSpecification.builder()

.predefinedMetricType(MetricType.DYNAMO_DB_WRITE_CAPACITY_UTILIZATION)
                .build();

            TargetTrackingScalingPolicyConfiguration policyConfiguration =
TargetTrackingScalingPolicyConfiguration.builder()
                .predefinedMetricSpecification(specification)
                .targetValue(50.0)
                .scaleInCooldown(60)
                .scaleOutCooldown(60)
                .build();
```

```
            PutScalingPolicyRequest putScalingPolicyRequest =
 PutScalingPolicyRequest.builder()
                .targetTrackingScalingPolicyConfiguration(policyConfiguration)
                .serviceNamespace(ns)
                .scalableDimension(tableWCUs)
                .resourceId(tableId)
                .policyName(policyName)
                .policyType(PolicyType.TARGET_TRACKING_SCALING)
                .build();

            try {
                appAutoScalingClient.putScalingPolicy(putScalingPolicyRequest);
                System.out.println("You have successfully created a scaling
 policy for an Application Auto Scaling scalable target");
            } catch (ApplicationAutoScalingException e) {
                System.err.println("Error: " +
 e.awsErrorDetails().errorMessage());
            }
        }
    }
}
```

- Para obter detalhes da API, consulte RegisterScalableTargeta Referência AWS SDK for Java 2.x da API.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: esse cmdlet registra ou atualiza um destino escalável. Um destino escalável é um recurso cuja escala pode ser aumentada ou reduzida horizontalmente pelo Application Auto Scaling.

```
Add-AASScalableTarget -ServiceNamespace AppStream -ResourceId fleet/MyFleet -
ScalableDimension appstream:fleet:DesiredCapacity -MinCapacity 2 -MaxCapacity 10
```

- Para obter detalhes da API, consulte RegisterScalableTargetem Referência de Ferramentas da AWS para PowerShell cmdlet (V4).

Ferramentas para PowerShell V5

Exemplo 1: esse cmdlet registra ou atualiza um destino escalável. Um destino escalável é um recurso cuja escala pode ser aumentada ou reduzida horizontalmente pelo Application Auto Scaling.

```
Add-AASScalableTarget -ServiceNamespace AppStream -ResourceId fleet/MyFleet -
ScalableDimension appstream:fleet:DesiredCapacity -MinCapacity 2 -MaxCapacity 10
```

- Para obter detalhes da API, consulte RegisterScalableTargetem Referência de Ferramentas da AWS para PowerShell cmdlet (V5).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando esse serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Suporte de marcação para o Application Auto Scaling

Você pode usar o AWS CLI ou um SDK para marcar destinos escaláveis do Application Auto Scaling. Os alvos escaláveis são as entidades que representam os recursos AWS ou recursos personalizados que o Application Auto Scaling pode escalar.

Cada etiqueta é um rótulo que consiste em uma chave e um valor definidos pelo usuário usando a API do Application Auto Scaling. As etiquetas podem ajudar você a configurar o acesso granular a destinos escaláveis específicos de acordo com as necessidades da sua organização. Para obter mais informações, consulte ABAC com o Application Auto Scaling.

É possível adicionar etiquetas a novos destinos escaláveis ao registrá-los ou adicioná-las a destinos escaláveis existentes.

Os comandos comumente usados para gerenciar etiquetas incluem:

- register-scalable-targetpara marcar novos alvos escaláveis ao registrá-los.
- tag-resource para adicionar etiquetas a um destino escalável existente.
- list-tags-for-resourcepara retornar as tags em um destino escalável.
- untag-resource para excluir uma etiqueta.

Exemplo de marcação

Use o register-scalable-targetcomando a seguir com a `--tags` opção. Este exemplo adiciona uma etiqueta a um destino escalável com duas etiquetas: uma chave de etiqueta nomeada **`environment`** com o valor de etiqueta **`production`**, e uma chave de etiqueta nomeada **`iscontainerbased`** com o valor de etiqueta **`true`**.

Substitua os valores de amostra de `--min-capacity` `--max-capacity` e e o texto de amostra de pelo `--service-namespace` namespace do AWS serviço que você está usando com o Application Auto Scaling`--scalable-dimension`, pela dimensão escalável associada ao recurso que você está registrando `--resource-id` e por um identificador para o recurso. Para obter mais informações e exemplos de cada serviço, consulte os tópicos na Serviços da AWS que você pode usar com o Application Auto Scaling.

```
aws application-autoscaling register-scalable-target \
  --service-namespace namespace \
```

```
  --scalable-dimension dimension \
  --resource-id identifier \
  --min-capacity 1 --max-capacity 10 \
  --tags environment=production,iscontainerbased=true
```

Se obtiver êxito, esse comando retornará o ARN do destino escalável.

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:region:account-id:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

> **Note**
>
> Se esse comando gerar um erro, verifique se você atualizou o AWS CLI localmente para a versão mais recente.

Etiquetas para segurança

Use etiquetas para verificar se o solicitante (como um perfil ou usuário do IAM) tem permissões para executar determinadas ações. Forneça informações de tags no elemento de condição de uma política do IAM usando uma ou mais das seguintes chaves de condição:

- Use `aws:ResourceTag/tag-key: tag-value` para permitir (ou negar) ações do usuário em destinos escaláveis com etiquetas específicas.
- Use `aws:RequestTag/tag-key: tag-value` para exigir que uma tag específica esteja presente (ou ausente) em uma solicitação.
- Use `aws:TagKeys [tag-key, ...]` para exigir que chaves de tag específicas estejam presentes (ou ausentes) em uma solicitação.

Por exemplo, a seguinte política do IAM concede permissões para usar as ações `DeregisterScalableTarget`, `DeleteScalingPolicy` e `DeleteScheduledAction`. No entanto, ela também negará as ações se o destino escalável que está recebendo a ação tiver a etiqueta **`environment=production`**.

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:DeregisterScalableTarget",
                "application-autoscaling:DeleteScalingPolicy",
                "application-autoscaling:DeleteScheduledAction"
            ],
            "Resource": "*"
            }
        },
        {
            "Effect": "Deny",
            "Action": [
                "application-autoscaling:DeregisterScalableTarget",
                "application-autoscaling:DeleteScalingPolicy",
                "application-autoscaling:DeleteScheduledAction"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {"aws:ResourceTag/environment": "production"}
            }
        }
    ]
}
```

Controlar o acesso usando etiquetas

Use etiquetas para verificar se o solicitante (como um perfil ou usuário do IAM) tem permissões para adicionar, modificar ou excluir etiquetas para destinos escaláveis.

Por exemplo, é possível criar uma política do IAM que permita remover apenas a etiqueta com a chave **`temporary`** dos destinos escaláveis.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
```

```
            "Action": "application-autoscaling:UntagResource",
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": { "aws:TagKeys": ["temporary"] }
            }
        }
    ]
}
```

Segurança no Application Auto Scaling

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de uma arquitetura de data center e rede criada para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O modelo de responsabilidade compartilhada descreve isso como segurança da nuvem e segurança na nuvem:

- Segurança da nuvem — AWS é responsável por proteger a infraestrutura que executa AWS os serviços na AWS nuvem. AWS também fornece serviços que você pode usar com segurança. Auditores terceirizados testam e verificam regularmente a eficácia de nossa segurança como parte dos AWS programas de de . Para saber mais sobre os programas de conformidade que se aplicam ao Application Auto Scaling, consulte AWS serviços em escopo por programa de conformidade AWS .
- Segurança na nuvem — Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da sua empresa e as leis e normas aplicáveis.

Esta documentação ajuda a entender como aplicar o modelo de responsabilidade compartilhada ao usar o Application Auto Scaling. Os tópicos a seguir mostram como configurar o Application Auto Scaling para atender aos seus objetivos de segurança e compatibilidade. Você também aprenderá a usar outros AWS serviços que ajudam a monitorar e proteger seus recursos do Application Auto Scaling.

Conteúdo

- Proteção de dados no Application Auto Scaling
- Gerenciamento de Identidade e Acesso para o Application Auto Scaling
- Acessar o Application Auto Scaling usando endpoints de interface da VPC
- Resiliência no Application Auto Scaling
- Segurança da infraestrutura no Application Auto Scaling
- Validação da compatibilidade para o Application Auto Scaling

Proteção de dados no Application Auto Scaling

O modelo de responsabilidade AWS compartilhada O modelo se aplica à proteção de dados no Application Auto Scaling. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre o conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para obter mais informações sobre a privacidade de dados, consulte as Data Privacy FAQ. Para obter mais informações sobre a proteção de dados na Europa, consulte a postagem do blog AWS Shared Responsibility Model and RGPD no Blog de segurança da AWS .

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com AWS os recursos. Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure a API e o registro de atividades do usuário com AWS CloudTrail. Para obter informações sobre o uso de CloudTrail trilhas para capturar AWS atividades, consulte Como trabalhar com CloudTrail trilhas no Guia AWS CloudTrail do usuário.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-3 ao acessar AWS por meio de uma interface de linha de comando ou de uma API, use um endpoint FIPS. Para obter mais informações sobre os endpoints FIPS disponíveis, consulte Federal Information Processing Standard (FIPS) 140-3.

É altamente recomendável que nunca sejam colocadas informações confidenciais ou sigilosas, como endereços de e-mail de clientes, em tags ou campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com o Application Auto Scaling ou outro Serviços da AWS usando o console, a API ou. AWS CLI AWS SDKs Quaisquer dados inseridos em tags ou em campos de texto

de formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, é fortemente recomendável que não sejam incluídas informações de credenciais no URL para validar a solicitação nesse servidor.

Gerenciamento de Identidade e Acesso para o Application Auto Scaling

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (ter permissões) para usar os recursos do Application Auto Scaling. O IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Para concluir a documentação do IAM, consulte o Guia do usuário do IAM.

Controle de acesso

É possível ter credenciais válidas para autenticar suas solicitações. No entanto, a menos que tenha permissões, não é possível criar nem acessar os recursos do Application Auto Scaling. Por exemplo, você deve ter permissões para criar políticas de escalabilidade, configurar escalabilidade agendada e assim por diante.

As seções a seguir fornecem detalhes sobre como um administrador do IAM pode usar o IAM para ajudar a proteger seus AWS recursos, controlando quem pode realizar ações da API Application Auto Scaling.

Conteúdo

- Como o Application Auto Scaling funciona com o IAM
- AWS políticas gerenciadas para Application Auto Scaling
- Funções vinculadas ao serviço necessárias para o Application Auto Scaling
- Políticas baseadas em identidade do Application Auto Scaling
- Solução de problemas de acesso ao Application Auto Scaling
- Validação de permissões para chamadas de API do Application Auto Scaling em recursos de destino

Como o Application Auto Scaling funciona com o IAM

> **Note**
>
> Em dezembro de 2017, houve uma atualização do Application Auto Scaling, habilitando várias funções vinculadas a serviços para os serviços integrados do Application Auto Scaling. Permissões específicas do IAM e uma função vinculada ao serviço do Application Auto Scaling (ou uma função de serviço para a escalabilidade automática do Amazon EMR) são necessárias para que os usuários possam configurar a escalabilidade.

Antes de usar o IAM para gerenciar o acesso ao Application Auto Scaling, aprenda quais recursos do IAM estão disponíveis para uso com o Application Auto Scaling.

Recursos do IAM que você pode usar com o Application Auto Scaling

Recurso do IAM	Compatibilidade com a aplicação Auto Scaling
Políticas baseadas em identidade	Sim
Ações de políticas	Sim
Recursos de políticas	Sim
Chaves de condição de política (específicas do serviço)	Sim
Políticas baseadas em recurso	Não
ACLs	Não
ABAC (tags em políticas)	Parcial
Credenciais temporárias	Sim
Perfis de serviço	Sim
Perfis vinculados a serviço	Sim

Para ter uma visão de alto nível de como o Application Auto Scaling e Serviços da AWS outros funcionam com a maioria dos recursos do IAM, Serviços da AWS consulte esse trabalho com o IAM no Guia do usuário do IAM.

Políticas baseadas em identidade do Application Auto Scaling

Compatível com políticas baseadas em identidade: sim

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário do IAM, grupo de usuários ou perfil. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente no Guia do Usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações e recursos permitidos ou negados, assim como as condições sob as quais as ações são permitidas ou negadas. Você não pode especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou perfil ao qual ela está anexada. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte Referência de elemento de política JSON do IAM no Guia do usuário do IAM.

Exemplos de políticas baseadas em identidade do Application Auto Scaling

Para visualizar exemplos de políticas baseadas em identidade do , Application Auto Scaling consulte Políticas baseadas em identidade do Application Auto Scaling.

Ações

Compatível com ações de políticas: sim

Em uma declaração de política do IAM, é possível especificar qualquer ação de API de qualquer serviço que dê suporte ao IAM. Para o Application Auto Scaling, use o seguinte prefixo com o nome da ação da API: `application-autoscaling:`. Por exemplo: `application-autoscaling:RegisterScalableTarget`, `application-autoscaling:PutScalingPolicy` e `application-autoscaling:DeregisterScalableTarget`.

Para especificar várias ações em uma única declaração, separe-as com vírgulas, conforme exibido no exemplo a seguir.

```
"Action": [
```

```
      "application-autoscaling:DescribeScalingPolicies",
      "application-autoscaling:DescribeScalingActivities"
```

Você também pode especificar várias ações usando caracteres curinga (*). Por exemplo, para especificar todas as ações que começam com a palavra `Describe`, inclua a ação a seguir:

```
"Action": "application-autoscaling:Describe*"
```

Para obter uma lista das ações do Application Auto Scaling, consulte Ações definidas pelo AWS Application Auto Scaling na Referência de Autorização de Serviço.

Recursos

Compatível com recursos de políticas: sim

Em uma instrução de política do IAM, o elemento `Resource` especifica o objeto ou os objetos abrangidos pela instrução. Para Application Auto Scaling, cada declaração de política do IAM se aplica às metas escaláveis que você especifica usando seus Amazon Resource Names (). ARNs

O formato de recurso do ARN para destinos escaláveis:

```
arn:aws:application-autoscaling:region:account-id:scalable-target/unique-identifier
```

Por exemplo, é possível indicar um destino escalável específico em sua instrução usando o ARN da maneira descrita a seguir. O ID exclusivo (1234abcd56ab78cd901ef1234567890ab123) é um valor atribuído pelo Application Auto Scaling ao destino escalável.

```
"Resource": "arn:aws:application-autoscaling:us-east-1:123456789012:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
```

É possível especificar todas as instâncias pertencentes a uma conta específica ao substituir o identificador exclusivo por um curinga (*), conforme descrito a seguir.

```
"Resource": "arn:aws:application-autoscaling:us-east-1:123456789012:scalable-target/*"
```

Para especificar todos os recursos, ou se uma ação de API específica não for compatível ARNs, use um caractere curinga (*) como `Resource` elemento da seguinte forma.

```
"Resource": "*"
```

Para obter mais informações, consulte Tipos de recursos definidos pelo AWS Application Auto Scaling na Referência de Autorização de Serviço.

Chaves de condição

Compatível com chaves de condição de política específicas de serviço: sim

É possível especificar condições nas políticas do IAM que controlam o acesso aos recursos do Application Auto Scaling. A declaração de política é efetiva apenas quando as condições forem verdadeiras.

O Application Auto Scaling oferece suporte às chaves de condição a seguir definidas pelo serviço que você pode usar em políticas baseadas em identidade para determinar quem pode executar ações de API do Application Auto Scaling.

- `application-autoscaling:scalable-dimension`
- `application-autoscaling:service-namespace`

Para saber com quais ações da API Application Auto Scaling você pode usar uma chave de condição, consulte Ações definidas pelo AWS Application Auto Scaling na Referência de Autorização de Serviço. Para obter mais informações sobre o uso das chaves de condição do Application Auto Scaling, consulte Chaves de condição do AWS Application Auto Scaling.

Para visualizar as chaves de condição globais disponíveis para todos os serviços, consulte Chaves de contexto de condição globais da AWS no Guia do usuário do IAM.

Políticas baseadas em recursos

Compatibilidade com políticas baseadas em recursos: não

Outros AWS serviços, como o Amazon Simple Storage Service, oferecem suporte a políticas de permissões baseadas em recursos. Por exemplo: você pode anexar uma política de permissões a um bucket do S3 para gerenciar permissões de acesso a esse bucket.

O Application Auto Scaling não é compatível com políticas baseadas em recurso.

Listas de controle de acesso (ACLs)

Suportes ACLs: Não

O Application Auto Scaling não suporta listas de controle de acesso ()ACLs.

ABAC com o Application Auto Scaling

Compatível com ABAC (tags em políticas): parcial

O controle de acesso por atributo (ABAC) é uma estratégia de autorização que define as permissões com base em atributos. Em AWS, esses atributos são chamados de tags. Você pode anexar tags a entidades do IAM (usuários ou funções) e a vários AWS recursos. Marcar de entidades e atributos é a primeira etapa do ABAC. Em seguida, você cria políticas de ABAC para permitir operações quando a tag da entidade principal corresponder à tag do recurso que ela estiver tentando acessar.

O ABAC é útil em ambientes que estão crescendo rapidamente e ajuda em situações em que o gerenciamento de políticas se torna um problema.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no elemento de condição de uma política usando as `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou chaves de condição `aws:TagKeys`.

É possível usar o ABAC em recursos compatíveis com tags, mas nem tudo é compatível com tags. As ações programadas e as políticas de escalabilidade não oferecem suporte para etiquetas, mas os destinos escaláveis oferecem suporte para etiquetas. Para obter mais informações, consulte Suporte de marcação para o Application Auto Scaling.

Para obter mais informações sobre o ABAC, consulte O que é ABAC? no Guia do Usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte Utilizar controle de acesso baseado em atributos (ABAC) no Guia do usuário do IAM.

Usar credenciais temporárias com o Application Auto Scaling

Compatível com credenciais temporárias: sim

Alguns Serviços da AWS não funcionam quando você faz login usando credenciais temporárias. Para obter informações adicionais, incluindo quais Serviços da AWS funcionam com credenciais temporárias, consulte Serviços da AWS “Trabalhe com o IAM” no Guia do usuário do IAM.

Você está usando credenciais temporárias se fizer login AWS Management Console usando qualquer método, exceto um nome de usuário e senha. Por exemplo, quando você acessa AWS usando o link de login único (SSO) da sua empresa, esse processo cria automaticamente credenciais temporárias. Você também cria automaticamente credenciais temporárias quando faz login no

console como usuário e, em seguida, alterna perfis. Para obter mais informações sobre como alternar funções, consulte Alternar para um perfil do IAM (console) no Guia do usuário do IAM.

Você pode criar manualmente credenciais temporárias usando a AWS API AWS CLI ou. Em seguida, você pode usar essas credenciais temporárias para acessar AWS. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para obter mais informações, consulte Credenciais de segurança temporárias no IAM.

Perfis de serviço

Compatível com perfis de serviço: sim

Se o cluster do Amazon EMR usa escalabilidade automática. Esse recurso permite que o Application Auto Scaling assuma uma função de serviço em seu nome. Semelhante a uma função vinculada ao serviço, uma função de serviço permite que o serviço acesse recursos em outros serviços para concluir uma ação em seu nome. Os perfis de serviço aparecem em sua conta do IAM e são de propriedade da conta. Isso significa que um administrador do IAM pode alterar as permissões para esse perfil. Porém, fazer isso pode alterar a funcionalidade do serviço.

O Application Auto Scaling é compatível com funções de serviço apenas para o Amazon EMR. Para obter a documentação sobre a função de serviço do EMR, consulte Usar escalabilidade automática com uma política personalizada para grupos de instâncias no Guia de gerenciamento do Amazon EMR.

> **Note**
>
> Com a introdução de funções vinculadas ao serviço, várias funções de serviço herdadas não são mais necessárias, por exemplo, para Amazon ECS e Frota spot.

Funções vinculadas ao serviço

Compatibilidade com perfis vinculados a serviços: sim

Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode presumir o perfil para executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não editar as permissões para perfis vinculados a serviço.

Para obter mais informações sobre funções vinculadas ao serviço para o Application Auto Scaling, consulte Funções vinculadas ao serviço necessárias para o Application Auto Scaling.

AWS políticas gerenciadas para Application Auto Scaling

Uma política AWS gerenciada é uma política autônoma criada e administrada por AWS. AWS as políticas gerenciadas são projetadas para fornecer permissões para muitos casos de uso comuns, para que você possa começar a atribuir permissões a usuários, grupos e funções.

Lembre-se de que as políticas AWS gerenciadas podem não conceder permissões de privilégio mínimo para seus casos de uso específicos porque estão disponíveis para uso de todos os AWS clientes. Recomendamos que você reduza ainda mais as permissões definindo as políticas gerenciadas pelo cliente que são específicas para seus casos de uso.

Você não pode alterar as permissões definidas nas políticas AWS gerenciadas. Se AWS atualizar as permissões definidas em uma política AWS gerenciada, a atualização afetará todas as identidades principais (usuários, grupos e funções) às quais a política está anexada. AWS é mais provável que atualize uma política AWS gerenciada quando uma nova AWS service (Serviço da AWS) é lançada ou novas operações de API são disponibilizadas para serviços existentes.

Para mais informações, consulte Políticas gerenciadas pela AWS no Manual do usuário do IAM.

AWS política gerenciada: AppStream 2.0 e CloudWatch

Nome da política: AWSApplicationAutoscalingAppStreamFleetPolicy

Essa política é anexada à função vinculada ao serviço nomeada AWSServiceRoleForApplicationAutoScaling_AppStreamFleetpara permitir que o Application Auto Scaling ligue para a AppStream Amazon CloudWatch e realize escalabilidade em seu nome.

Detalhes de permissões

A política concede permissões para que o Application Auto Scaling conclua as seguintes ações em todos os recursos relacionados ("Resource": "*"):

- Ação: `appstream:DescribeFleets`
- Ação: `appstream:UpdateFleet`
- Ação: `cloudwatch:DescribeAlarms`
- Ação: `cloudwatch:PutMetricAlarm`
- Ação: `cloudwatch:DeleteAlarms`

AWS política gerenciada: Aurora e CloudWatch

Nome da política: Política de AWSApplicationescalonamento automático RDSCluster

Essa política está anexada à função vinculada ao serviço nomeada AWSServiceRoleForApplicationAutoScaling_RDSClusterpara permitir que o Application Auto Scaling chame a Aurora CloudWatch e realize o escalonamento em seu nome.

Detalhes de permissões

A política concede permissões para que o Application Auto Scaling conclua as seguintes ações em todos os recursos relacionados ("Resource": "*"):

- Ação: `rds:AddTagsToResource`
- Ação: `rds:CreateDBInstance`
- Ação: `rds:DeleteDBInstance`
- Ação: `rds:DescribeDBClusters`
- Ação: `rds:DescribeDBInstance`
- Ação: `cloudwatch:DescribeAlarms`
- Ação: `cloudwatch:PutMetricAlarm`
- Ação: `cloudwatch:DeleteAlarms`

AWS política gerenciada: Amazon Comprehend e CloudWatch

Nome da política: AWSApplicationAutoscalingComprehendEndpointPolicy

Essa política está anexada à função vinculada ao serviço nomeada AWSServiceRoleForApplicationAutoScaling_ComprehendEndpointpara permitir que o Application Auto Scaling chame o Amazon Comprehend e realize escalabilidade em seu CloudWatch nome.

Detalhes de permissões

A política concede permissões para que o Application Auto Scaling conclua as seguintes ações em todos os recursos relacionados ("Resource": "*"):

- Ação: `comprehend:UpdateEndpoint`
- Ação: `comprehend:DescribeEndpoint`
- Ação: `cloudwatch:DescribeAlarms`

- Ação: `cloudwatch:PutMetricAlarm`
- Ação: `cloudwatch:DeleteAlarms`

AWS política gerenciada: DynamoDB e CloudWatch

Nome da política: AWSApplicationAutoscalingDynamoDBTablePolítica

Essa política está anexada à função vinculada ao serviço AWSServiceRoleForApplicationAutoScaling_DynamoDBTablenomeada para permitir que o Application Auto Scaling chame DBand CloudWatch o Dynamo e realize o escalonamento em seu nome.

Detalhes de permissões

A política concede permissões para que o Application Auto Scaling conclua as seguintes ações em todos os recursos relacionados ("Resource": "*"):

- Ação: `dynamodb:DescribeTable`
- Ação: `dynamodb:UpdateTable`
- Ação: `cloudwatch:DescribeAlarms`
- Ação: `cloudwatch:PutMetricAlarm`
- Ação: `cloudwatch:DeleteAlarms`

AWS política gerenciada: Amazon ECS e CloudWatch

Nome da política: Política de AWSApplicationescalonamento automático ECSService

Essa política está anexada à função vinculada ao serviço nomeada AWSServiceRoleForApplicationAutoScaling_ECSServicepara permitir que o Application Auto Scaling chame o Amazon ECS CloudWatch e realize escalabilidade em seu nome.

Detalhes de permissões

A política concede permissões para que o Application Auto Scaling conclua as seguintes ações em todos os recursos relacionados ("Resource": "*"):

- Ação: `ecs:DescribeServices`
- Ação: `ecs:UpdateService`

- Ação: `cloudwatch:PutMetricAlarm`
- Ação: `cloudwatch:DescribeAlarms`
- Ação: `cloudwatch:GetMetricData`
- Ação: `cloudwatch:DeleteAlarms`

AWS política gerenciada: ElastiCache e CloudWatch

Nome da política: AWSApplicationAutoscalingElastiCacheRGPolicy

Essa política é anexada à função vinculada ao serviço nomeada AWSServiceRoleForApplicationAutoScaling_ElastiCacheRGpara permitir que o Application Auto Scaling ElastiCache chame CloudWatch e realize o escalonamento em seu nome. Essa função vinculada ao serviço pode ser usada para ElastiCache Memcached, Redis OSS e Valkey.

Detalhes de permissões

A política concede permissões para que o Application Auto Scaling conclua as seguintes ações nos recursos especificados:

- Ação: `elasticache:DescribeReplicationGroups` em todos os recursos
- Ação: `elasticache:ModifyReplicationGroupShardConfiguration` em todos os recursos
- Ação: `elasticache:IncreaseReplicaCount` em todos os recursos
- Ação: `elasticache:DecreaseReplicaCount` em todos os recursos
- Ação: `elasticache:DescribeCacheClusters` em todos os recursos
- Ação: `elasticache:DescribeCacheParameters` em todos os recursos
- Ação: `elasticache:ModifyCacheCluster` em todos os recursos
- Ação: `cloudwatch:DescribeAlarms` no recurso `arn:aws:cloudwatch:*:*:alarm:*`
- Ação: `cloudwatch:PutMetricAlarm` no recurso `arn:aws:cloudwatch:*:*:alarm:TargetTracking*`
- Ação: `cloudwatch:DeleteAlarms` no recurso `arn:aws:cloudwatch:*:*:alarm:TargetTracking*`

AWS política gerenciada: Amazon Keyspaces e CloudWatch

Nome da política: AWSApplicationAutoscalingCassandraTablePolicy

Essa política está anexada à função vinculada ao serviço nomeada AWSServiceRoleForApplicationAutoScaling_CassandraTablepara permitir que o Application Auto Scaling chame o Amazon Keyspaces CloudWatch e realize escalabilidade em seu nome.

Detalhes de permissões

A política concede permissões para que o Application Auto Scaling conclua as seguintes ações nos recursos especificados:

- Ação: `cassandra:Select` nos seguintes recursos:
  - `arn:*:cassandra:*:*:/keyspace/system/table/*`
  - `arn:*:cassandra:*:*:/keyspace/system_schema/table/*`
  - `arn:*:cassandra:*:*:/keyspace/system_schema_mcs/table/*`
- Ação: `cassandra:Alter` em todos os recursos
- Ação: `cloudwatch:DescribeAlarms` em todos os recursos
- Ação: `cloudwatch:PutMetricAlarm` em todos os recursos
- Ação: `cloudwatch:DeleteAlarms` em todos os recursos

AWS política gerenciada: Lambda e CloudWatch

Nome da política: AWSApplicationAutoscalingLambdaConcurrencyPolicy

Essa política é anexada à função vinculada ao serviço nomeada AWSServiceRoleForApplicationAutoScaling_LambdaConcurrencypara permitir que o Application Auto Scaling chame o Lambda CloudWatch e realize o escalonamento em seu nome.

Detalhes de permissões

A política concede permissões para que o Application Auto Scaling conclua as seguintes ações em todos os recursos relacionados ("Resource": "*"):

- Ação: `lambda:PutProvisionedConcurrencyConfig`
- Ação: `lambda:GetProvisionedConcurrencyConfig`
- Ação: `lambda:DeleteProvisionedConcurrencyConfig`
- Ação: `cloudwatch:DescribeAlarms`
- Ação: `cloudwatch:PutMetricAlarm`
- Ação: `cloudwatch:DeleteAlarms`

AWS política gerenciada: Amazon MSK e CloudWatch

Nome da política: AWSApplicationAutoscalingKafkaClusterPolicy

Essa política é anexada à função vinculada ao serviço nomeada AWSServiceRoleForApplicationAutoScaling_KafkaClusterpara permitir que o Application Auto Scaling chame o Amazon MSK CloudWatch e realize escalabilidade em seu nome.

Detalhes de permissões

A política concede permissões para que o Application Auto Scaling conclua as seguintes ações em todos os recursos relacionados ("Resource": "*"):

- Ação: `kafka:DescribeCluster`
- Ação: `kafka:DescribeClusterOperation`
- Ação: `kafka:UpdateBrokerStorage`
- Ação: `cloudwatch:DescribeAlarms`
- Ação: `cloudwatch:PutMetricAlarm`
- Ação: `cloudwatch:DeleteAlarms`

AWS política gerenciada: Neptune e CloudWatch

Nome da política: AWSApplicationAutoscalingNeptuneClusterPolicy

Essa política está anexada à função vinculada ao serviço nomeada AWSServiceRoleForApplicationAutoScaling_NeptuneClusterpara permitir que o Application Auto Scaling chame o Neptune CloudWatch e realize o escalonamento em seu nome.

Detalhes de permissões

A política concede permissões para que o Application Auto Scaling conclua as seguintes ações nos recursos especificados:

- Ação: `rds:ListTagsForResource` em todos os recursos
- Ação: `rds:DescribeDBInstances` em todos os recursos
- Ação: `rds:DescribeDBClusters` em todos os recursos
- Ação: `rds:DescribeDBClusterParameters` em todos os recursos
- Ação: `cloudwatch:DescribeAlarms` em todos os recursos

- Ação: `rds:AddTagsToResource` em recursos com o prefixo autoscaled-reader no mecanismo de banco de dados do Amazon Neptune (`"Condition":{"StringEquals":{"rds:DatabaseEngine":"neptune"}}`)
- Ação: `rds:CreateDBInstance` em recursos com o prefixo autoscaled-reader em todos os clusters de banco de dados (`"Resource":"arn:*:rds:*:*:db:autoscaled-reader*"`, `"arn:aws:rds:*:*:cluster:*"`) no mecanismo de banco de dados do Amazon Neptune (`"Condition":{"StringEquals":{"rds:DatabaseEngine":"neptune"}}`)
- Ação: `rds:DeleteDBInstance` no recurso `arn:aws:rds:*:*:db:autoscaled-reader*`
- Ação: `cloudwatch:PutMetricAlarm` no recurso `arn:aws:cloudwatch:*:*:alarm:TargetTracking*`
- Ação: `cloudwatch:DeleteAlarms` no recurso `arn:aws:cloudwatch:*:*:alarm:TargetTracking*`

AWS política gerenciada: SageMaker IA e CloudWatch

Nome da política: AWSApplicationAutoscalingSageMakerEndpointPolicy

Essa política é anexada à função vinculada ao serviço nomeada AWSServiceRoleForApplicationAutoScaling_SageMakerEndpointpara permitir que o Application Auto Scaling SageMaker chame a IA CloudWatch e realize o escalonamento em seu nome.

Detalhes de permissões

A política concede permissões para que o Application Auto Scaling conclua as seguintes ações nos recursos especificados:

- Ação: `sagemaker:DescribeEndpoint` em todos os recursos
- Ação: `sagemaker:DescribeEndpointConfig` em todos os recursos
- Ação: `sagemaker:DescribeInferenceComponent` em todos os recursos
- Ação: `sagemaker:UpdateEndpointWeightsAndCapacities` em todos os recursos
- Ação: `sagemaker:UpdateInferenceComponentRuntimeConfig` em todos os recursos
- Ação: `cloudwatch:DescribeAlarms` em todos os recursos
- Ação: `cloudwatch:GetMetricData` em todos os recursos
- Ação: `cloudwatch:PutMetricAlarm` no recurso `arn:aws:cloudwatch:*:*:alarm:TargetTracking*`

- Ação: `cloudwatch:DeleteAlarms` no recurso `arn:aws:cloudwatch:*:*:alarm:TargetTracking*`

AWS política gerenciada: EC2 Spot Fleet e CloudWatch

Nome da política: AWSApplicationEscalonamento automático EC2 SpotFleetRequestPolicy

Essa política é anexada à função vinculada ao serviço chamada AWSServiceRoleForApplicationAutoScaling_EC2 SpotFleetRequest para permitir que o Application Auto Scaling ligue para a EC2 Amazon CloudWatch e realize escalabilidade em seu nome.

Detalhes de permissões

A política concede permissões para que o Application Auto Scaling conclua as seguintes ações em todos os recursos relacionados ("Resource": "*"):

- Ação: `ec2:DescribeSpotFleetRequests`
- Ação: `ec2:ModifySpotFleetRequest`
- Ação: `cloudwatch:DescribeAlarms`
- Ação: `cloudwatch:PutMetricAlarm`
- Ação: `cloudwatch:DeleteAlarms`

AWS política gerenciada: WorkSpaces e CloudWatch

Nome da política: AWSApplicationAutoscalingWorkSpacesPoolPolicy

Essa política é anexada à função vinculada ao serviço nomeada AWSServiceRoleForApplicationAutoScaling_WorkSpacesPoolpara permitir que o Application Auto Scaling WorkSpaces chame CloudWatch e realize o escalonamento em seu nome.

Detalhes de permissões

A política concede permissões para que o Application Auto Scaling conclua as seguintes ações nos recursos especificados:

- Ação: `workspaces:DescribeWorkspacesPools` em todos os recursos da mesma conta do SLR
- Ação: `workspaces:UpdateWorkspacesPool` em todos os recursos da mesma conta do SLR
- Ação: `cloudwatch:DescribeAlarms` em todos os alarmes da mesma conta do SLR

- Ação: `cloudwatch:PutMetricAlarm` em todos os alarmes da mesma conta do SLR, em que o nome do alarme começa com TargetTracking
- Ação: `cloudwatch:DeleteAlarms` em todos os alarmes da mesma conta do SLR, em que o nome do alarme começa com TargetTracking

AWS política gerenciada: recursos personalizados e CloudWatch

Nome da política: AWSApplicationAutoScalingCustomResourcePolicy

Essa política é anexada à função vinculada ao serviço nomeada AWSServiceRoleForApplicationAutoScaling_CustomResourcepara permitir que o Application Auto Scaling chame seus recursos personalizados que estão disponíveis por meio do API Gateway CloudWatch e realize o escalonamento em seu nome.

Detalhes de permissões

A política concede permissões para que o Application Auto Scaling conclua as seguintes ações em todos os recursos relacionados ("Resource": "*"):

- Ação: `execute-api:Invoke`
- Ação: `cloudwatch:DescribeAlarms`
- Ação: `cloudwatch:PutMetricAlarm`
- Ação: `cloudwatch:DeleteAlarms`

Atualizações do Application Auto Scaling para políticas AWS gerenciadas

Veja detalhes sobre as atualizações das políticas AWS gerenciadas do Application Auto Scaling desde que esse serviço começou a rastrear essas alterações. Para receber alertas automáticos sobre alterações feitas nesta página, inscreva-se no feed de RSS na págin Document History (Histórico de documentos) do Application Auto Scaling.

Alteração	Descrição	Data
AWSApplicationAutoscalingElastiCacheRGPolicy— Atualizar uma política existente	Foi adicionada permissão para chamar a ação da ElastiCache `ModifyCacheCluster` API para oferecer suporte ao	10 de abril de 2025

Alteração	Descrição	Data
	escalonamento automático do Memcached.	
AWSApplicationPolítica de escalonamento automático — atualize uma ECSService política existente	Foi adicionada permissão para chamar a ação da CloudWatch `GetMetricData` API para oferecer suporte à escalabilidade preditiva.	21 de novembro de 2024
AWSApplicationAutoscalingWorkSpacesPoolPolicy – Nova política	Foi adicionada uma política gerenciada para a Amazon WorkSpaces. Essa política está vinculada a uma função vinculada ao serviço que permite que o Application Auto Scaling WorkSpaces chame CloudWatch e realize o escalonamento em seu nome.	24 de junho de 2024

Alteração	Descrição	Data
AWSApplicationAutoscalingSageMakerEndpointPolicy: atualização para uma política existente	Foram adicionadas permissões para chamar as ações de SageMaker IA `DescribeInferenceComponent` e `UpdateInferenceComponentRuntimeConfig` API para oferecer suporte à compatibilidade do escalonamento automático de recursos de SageMaker IA para uma integração futura. Agora, a política também restringe as CloudWatch `PutMetricAlarm` ações da `DeleteAlarms` API aos CloudWatch alarmes usados com políticas de escalabilidade de rastreamento de metas.	13 de novembro de 2023
AWSApplicationAutoscalingNeptuneClusterPolicy – Nova política	Adição de uma política gerenciada para o Neptune. Essa política está vinculada a uma função vinculada ao serviço que permite que o Application Auto Scaling chame o Neptune CloudWatch e realize o escalonamento em seu nome.	6 de outubro de 2021

Alteração	Descrição	Data
AWSApplicationPolítica de escalonamento automático — RDSCluster Nova política	Foi adicionada uma política gerenciada para ElastiCache. Essa política está vinculada a uma função vinculada ao serviço que permite que o Application Auto Scaling ElastiCache chame CloudWatch e realize o escalonamento em seu nome.	19 de agosto de 2021
O Application Auto Scaling começou a monitorar alterações	O Application Auto Scaling começou a monitorar as mudanças em suas políticas AWS gerenciadas.	19 de agosto de 2021

Funções vinculadas ao serviço necessárias para o Application Auto Scaling

O Application Auto Scaling usa funções vinculadas a serviços para obter as permissões necessárias para chamar outros AWS serviços em seu nome. Uma função vinculada ao serviço é um tipo exclusivo de função AWS Identity and Access Management (IAM) vinculada diretamente a um AWS serviço. As funções vinculadas ao serviço fornecem uma maneira segura de delegar permissões aos AWS serviços porque somente o serviço vinculado pode assumir uma função vinculada ao serviço.

Para serviços que se integram ao Application Auto Scaling, o Application Auto Scaling cria funções vinculadas ao serviço para você. Há uma função vinculada ao serviço para cada serviço. Cada função vinculada ao serviço confia que o serviço principal especificado a assumirá. Para obter mais informações, consulte Referência do ARN da função vinculada ao serviço.

O Application Auto Scaling inclui todas as permissões necessárias para cada função vinculada ao serviço. Essas permissões gerenciadas são criadas e gerenciadas pelo Application Auto Scaling e definem as ações permitidas para cada tipo de recurso. Para obter detalhes sobre as permissões concedidas por cada função, consulte AWS políticas gerenciadas para Application Auto Scaling.

Conteúdo

- Permissões necessárias para criar uma função vinculada ao serviço

- Criar funções vinculadas a serviços (automático)
- Criar funções vinculadas a serviços (manual)
- Editar funções vinculadas ao serviço
- Excluir funções vinculadas ao serviço
- Regiões compatíveis com funções vinculadas ao serviço do Application Auto Scaling
- Referência do ARN da função vinculada ao serviço

Permissões necessárias para criar uma função vinculada ao serviço

O Application Auto Scaling exige permissões para criar uma função vinculada ao serviço na primeira vez que qualquer usuário em suas Conta da AWS chamadas `RegisterScalableTarget` para um determinado serviço. O Application Auto Scaling criará uma função vinculada ao serviço para o serviço de destino na sua conta se a função ainda não existir. A função vinculada ao serviço concede permissões ao Application Auto Scaling para que ele possa chamar o serviço de destino em seu nome.

Para que a criação automática da função seja bem-sucedida, os usuários devem ter permissão para a ação `iam:CreateServiceLinkedRole`.

```
"Action": "iam:CreateServiceLinkedRole"
```

Veja a seguir uma política baseada em identidade que concede permissão para criar um perfil vinculado ao serviço para o Spot Fleet. É possível especificar a função vinculada ao serviço n campo `Resource` da política como ARN, o serviço principal para sua função vinculada ao serviço como condição, conforme mostrado. Para obter o ARN para cada serviço, consulte Referência do ARN da função vinculada ao serviço.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
```

```
            "Resource": "arn:aws:iam::*:role/aws-
service-role/ec2.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_EC2SpotFleetRequest",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName":"ec2.application-
autoscaling.amazonaws.com"
                }
            }
        }
    ]
}
```

> **Note**
>
> A chave de condição do IAM `iam:AWSServiceName` especifica o principal de serviço ao qual a função está anexada, o que é indicado neste exemplo de política como *`ec2.application-autoscaling`*`.amazonaws.com`. oNão tente adivinhar a entidade principal do serviço. Para visualizar a entidade principal do serviço, consulte Serviços da AWS que você pode usar com o Application Auto Scaling.

Criar funções vinculadas a serviços (automático)

Não é necessário criar manualmente um perfil vinculado ao serviço. O Application Auto Scaling criará a função vinculada ao serviço adequada para você quando você chamar `RegisterScalableTarget`. Por exemplo, se você configurar a escalabilidade automática para um serviço do Amazon ECS, o Application Auto Scaling criará a função `AWSServiceRoleForApplicationAutoScaling_ECSService`.

Criar funções vinculadas a serviços (manual)

Para criar a função vinculada ao serviço, você pode usar o console do IAM ou a AWS CLI API do IAM. Para ter mais informações, consulte Criar um perfil vinculado ao serviço no Guia do usuário do IAM.

Para criar uma função vinculada a serviço (AWS CLI)

Use o create-service-linked-rolecomando a seguir para criar a função vinculada ao serviço Application Auto Scaling. Na solicitação, especifique o nome do serviço “prefix”.

Para localizar o prefixo de nome de serviço, consulte as informações sobre o principal de serviço para a função vinculada ao serviço para cada serviço na seção Serviços da AWS que você pode usar com o Application Auto Scaling. O nome do serviço e o principal de serviço compartilham o mesmo prefixo. Por exemplo, para criar a função AWS Lambda vinculada ao serviço, use. `lambda.application-autoscaling.amazonaws.com`

```
aws iam create-service-linked-role --aws-service-name prefix.application-
autoscaling.amazonaws.com
```

Editar funções vinculadas ao serviço

Com as funções vinculadas ao serviço criadas pelo Application Auto Scaling, é possível editar somente suas descrições. Para obter mais informações, consulte Editar uma descrição de perfil vinculado ao serviço no Guia do usuário do IAM.

Excluir funções vinculadas ao serviço

Se você não precisar mais usar o com um serviço compatível com o Application Auto Scaling, recomendamos que exclua a função vinculada ao serviço correspondente.

Você pode excluir uma função vinculada ao serviço somente depois de excluir os recursos relacionados da AWS . Isso evita que você revogue acidentalmente as permissões do Application Auto Scaling para seus recursos. Para obter mais informações, consulte a documentação do recurso dimensionável. Por exemplo, para excluir um serviço do Amazon ECS, consulte Excluir um serviço do Amazon ECS no Guia do desenvolvedor do Amazon Elastic Container Service.

É possível usar o IAM para excluir uma função vinculada ao serviço. Para obter mais informações, consulte Excluir uma função vinculada ao serviço no Guia do usuário do IAM.

Depois que você excluir uma função vinculada ao serviço, o Application Auto Scaling criará novamente quando você chamar `RegisterScalableTarget`.

Regiões compatíveis com funções vinculadas ao serviço do Application Auto Scaling

O Application Auto Scaling suporta o uso de funções vinculadas ao serviço em todas as AWS regiões em que o serviço está disponível.

Referência do ARN da função vinculada ao serviço

A tabela a seguir lista o Amazon Resource Name (ARN) da função vinculada ao serviço para cada uma AWS service (Serviço da AWS) que funciona com o Application Auto Scaling.

Serviço	ARN
AppStream 2.0	arn:aws:iam:: *012345678910* :role/aws-service-role/appstream.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_AppStreamFleet
Aurora	arn:aws:iam:: *012345678910* :role/aws-service-role/rds.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_RDSCluster
Comprehend	arn:aws:iam:: *012345678910* :role/aws-service-role/comprehend.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_ComprehendEndpoint
DynamoDB	arn:aws:iam:: *012345678910* :role/aws-service-role/dynamodb.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_DynamoDBTable
ECS	arn:aws:iam:: *012345678910* :role/aws-service-role/ecs.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_ECSService
ElastiCache	arn:aws:iam:: *012345678910* :role/aws-service-role/elasticache.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG
Keyspaces	arn:aws:iam:: *012345678910* :role/aws-service-role/cassandra.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_CassandraTable
Lambda	arn:aws:iam:: *012345678910* :role/aws-service-role/lambda.application-autoscaling.amazonaws.com/AWSS

Serviço	ARN
	erviceRoleForApplicationAutoScaling_LambdaCon currency
MSK	arn:aws:iam:: *012345678910* :role/aws-service-role/ kafka.application-autoscaling.amazonaws.com/AWSSe rviceRoleForApplicationAutoScaling_KafkaCluster
Neptune	arn:aws:iam:: *012345678910* :role/aws-service-role/ neptune.application-autoscaling.amazonaws.com/ AWSServiceRoleForApplicationAutoScaling_NeptuneC luster
SageMaker IA	arn:aws:iam:: *012345678910* :role/aws-service-role/ sagemaker.application-autoscaling.amazonaws.com/ AWSServiceRoleForApplicationAutoScaling_SageMa kerEndpoint
Spot Fleets	arn:aws:iam:: *012345678910* :role/aws-service-role/ ec2.application-autoscaling.amazonaws.com/AWSServ iceRoleForApplicationAutoScaling_EC2SpotFleet Request
WorkSpaces	arn:aws:iam:: *012345678910* :role/aws-service-role/ workspaces.application-autoscaling.amazonaws.com/ AWSServiceRoleForApplicationAutoScaling_WorkS pacesPool
Recursos personali zados	arn:aws:iam:: *012345678910* :role/aws-service-role/cust om-resource.application-autoscaling.amazonaws.com/ AWSServiceRoleForApplicationAutoScaling_CustomRes ource

Note

Você pode especificar o ARN de uma função vinculada ao serviço para a `RoleARN` propriedade de um AWS::ApplicationAutoScaling::ScalableTargetrecurso em seus modelos

de AWS CloudFormation pilha, mesmo que a função vinculada ao serviço especificada ainda não exista. O Application Auto Scaling cria automaticamente a função para você.

Políticas baseadas em identidade do Application Auto Scaling

Por padrão, um novo usuário não Conta da AWS tem permissão para fazer nada. Um administrador do IAM deve criar e atribuir políticas do IAM que concedam a uma identidade do IAM (como um usuário ou perfil) permissão para executar ações de API do Application Auto Scaling.

Para saber como criar uma política do IAM usando os exemplos de documentos de política JSON a seguir, consulte Criar políticas na aba JSON no Manual do usuário do IAM.

Conteúdo

- Permissões necessárias para ações da API do Application Auto Scaling
- Permissões necessárias para ações de API nos serviços de destino e CloudWatch
- Permissões para trabalhar no AWS Management Console

Permissões necessárias para ações da API do Application Auto Scaling

As políticas a seguir concedem permissões para casos de uso comuns ao chamar a API do Application Auto Scaling. Consulte esta seção ao escrever políticas baseadas em identidade. Cada política concede permissões para todas ou para algumas ações de API do Application Auto Scaling. Você também precisa garantir que os usuários finais tenham permissões para o serviço de destino e CloudWatch (consulte a próxima seção para obter detalhes).

A política baseada em identidade a seguir concede permissões para todas as ações de API do Application Auto Scaling.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
```

```
              "application-autoscaling:*"
            ],
            "Resource": "*"
        }
    ]
}
```

A política baseada em identidade a seguir concede permissões para todas as ações de API do Application Auto Scaling que são necessárias para configurar políticas de escalação e ações não agendadas.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "application-autoscaling:RegisterScalableTarget",
              "application-autoscaling:DescribeScalableTargets",
              "application-autoscaling:DeregisterScalableTarget",
              "application-autoscaling:PutScalingPolicy",
              "application-autoscaling:DescribeScalingPolicies",
              "application-autoscaling:DescribeScalingActivities",
              "application-autoscaling:DeleteScalingPolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

A política baseada em identidade a seguir concede permissões para todas as ações de API do Application Auto Scaling que são necessárias para configurar ações programadas e políticas de não escalação.

JSON

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "application-autoscaling:RegisterScalableTarget",
              "application-autoscaling:DescribeScalableTargets",
              "application-autoscaling:DeregisterScalableTarget",
              "application-autoscaling:PutScheduledAction",
              "application-autoscaling:DescribeScheduledActions",
              "application-autoscaling:DescribeScalingActivities",
              "application-autoscaling:DeleteScheduledAction"
            ],
            "Resource": "*"
        }
    ]
}
```

Permissões necessárias para ações de API nos serviços de destino e CloudWatch

Para configurar e usar com sucesso o Application Auto Scaling com o serviço de destino, os usuários finais devem receber permissões para a Amazon CloudWatch e para cada serviço de destino para o qual eles configurarão a escalabilidade. Use as políticas a seguir para conceder as permissões mínimas necessárias para trabalhar com os serviços de destino CloudWatch e.

Conteúdo

- AppStream 2.0 frotas
- Réplicas do Aurora
- Classificação de documentos e endpoints de reconhecimento de entidade do Amazon Comprehend
- Tabelas e índices secundários globais do DynamoDB
- serviços da ECS
- ElastiCache grupos de replicação
- Clusters do Amazon EMR
- Tabelas do Amazon Keyspaces
- Funções do Lambda
- Armazenamento de agente do Amazon Managed Streaming for Apache Kafka (MSK)

- Clusters do Neptune
- SageMaker Endpoints de IA
- Spot Fleets (Amazon EC2)
- Recursos personalizados

AppStream 2.0 frotas

A política baseada em identidade a seguir concede permissões para todas as ações AppStream 2.0 e de CloudWatch API necessárias.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "appstream:DescribeFleets",
              "appstream:UpdateFleet",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Réplicas do Aurora

A política baseada em identidade a seguir concede permissões para todas as ações do Aurora e CloudWatch da API que são necessárias.

JSON

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "rds:AddTagsToResource",
              "rds:CreateDBInstance",
              "rds:DeleteDBInstance",
              "rds:DescribeDBClusters",
              "rds:DescribeDBInstances",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Classificação de documentos e endpoints de reconhecimento de entidade do Amazon Comprehend

A política baseada em identidade a seguir concede permissões para todas as ações de API CloudWatch e Amazon Comprehend que são necessárias.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "comprehend:UpdateEndpoint",
              "comprehend:DescribeEndpoint",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Tabelas e índices secundários globais do DynamoDB

A política baseada em identidade a seguir concede permissões para todas as ações necessárias do DynamoDB e CloudWatch da API.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "dynamodb:DescribeTable",
              "dynamodb:UpdateTable",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

serviços da ECS

A política baseada em identidade a seguir concede permissões para todas as ações do ECS e CloudWatch da API que são necessárias.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "ecs:DescribeServices",
              "ecs:UpdateService",
              "cloudwatch:DescribeAlarms",
```

```
                "cloudwatch:PutMetricAlarm",
                "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

ElastiCache grupos de replicação

A política baseada em identidade a seguir concede permissões para todas ElastiCache as ações de CloudWatch API necessárias.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "elasticache:ModifyReplicationGroupShardConfiguration",
              "elasticache:IncreaseReplicaCount",
              "elasticache:DecreaseReplicaCount",
              "elasticache:DescribeReplicationGroups",
              "elasticache:DescribeCacheClusters",
              "elasticache:DescribeCacheParameters",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Clusters do Amazon EMR

A política baseada em identidade a seguir concede permissões para todas as ações de CloudWatch API e do Amazon EMR que são necessárias.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "elasticmapreduce:ModifyInstanceGroups",
              "elasticmapreduce:ListInstanceGroups",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Tabelas do Amazon Keyspaces

A política baseada em identidade a seguir concede permissões para todas as ações de CloudWatch API e Amazon Keyspaces necessárias.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "cassandra:Select",
              "cassandra:Alter",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
```

```
}
```

Funções do Lambda

A política baseada em identidade a seguir concede permissões para todas as ações do Lambda e da CloudWatch API que são necessárias.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "lambda:PutProvisionedConcurrencyConfig",
              "lambda:GetProvisionedConcurrencyConfig",
              "lambda:DeleteProvisionedConcurrencyConfig",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Armazenamento de agente do Amazon Managed Streaming for Apache Kafka (MSK)

A política baseada em identidade a seguir concede permissões para todas as ações de CloudWatch API e MSK da Amazon que são necessárias.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
```

```
            "Action": [
              "kafka:DescribeCluster",
              "kafka:DescribeClusterOperation",
              "kafka:UpdateBrokerStorage",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Clusters do Neptune

A política baseada em identidade a seguir concede permissões para todas as ações do Neptune e CloudWatch da API que são necessárias.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "rds:AddTagsToResource",
              "rds:CreateDBInstance",
              "rds:DescribeDBInstances",
              "rds:DescribeDBClusters",
              "rds:DescribeDBClusterParameters",
              "rds:DeleteDBInstance",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

SageMaker Endpoints de IA

A política baseada em identidade a seguir concede permissões para todas as ações de SageMaker IA e CloudWatch API necessárias.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "sagemaker:DescribeEndpoint",
              "sagemaker:DescribeEndpointConfig",
              "sagemaker:DescribeInferenceComponent",
              "sagemaker:UpdateEndpointWeightsAndCapacities",
              "sagemaker:UpdateInferenceComponentRuntimeConfig",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Spot Fleets (Amazon EC2)

A política baseada em identidade a seguir concede permissões para todas as ações da Spot Fleet e CloudWatch da API que são necessárias.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
```

```
                "ec2:DescribeSpotFleetRequests",
                "ec2:ModifySpotFleetRequest",
                "cloudwatch:DescribeAlarms",
                "cloudwatch:PutMetricAlarm",
                "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Recursos personalizados

A política baseada em identidade a seguir concede permissão para a ação de execução de API do serviço API Gateway. Essa política também concede permissões para todas CloudWatch as ações necessárias.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "execute-api:Invoke",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Permissões para trabalhar no AWS Management Console

Não há console autônomo do Application Auto Scaling. A maioria dos serviços que se integram ao Application Auto Scaling tem recursos dedicados para ajudar você a configurar a escalabilidade com seu console.

Na maioria dos casos, cada serviço fornece políticas AWS gerenciadas (predefinidas) do IAM que definem o acesso ao console, o que inclui permissões para as ações da API Application Auto Scaling. Para obter mais informações, consulte a documentação do serviço do qual você deseja usar o console.

Também é possível criar suas próprias políticas personalizadas do IAM para conceder aos usuários permissões refinadas para visualizar e trabalhar com ações da API do Application Auto Scaling específicas no AWS Management Console. Você pode usar as políticas de exemplo nas seções anteriores; no entanto, elas foram projetadas para solicitações feitas com o AWS CLI ou com um SDK. O console usa ações de API adicionais para seus recursos, portanto, essas políticas talvez não funcionem como esperado. Por exemplo, para configurar o escalonamento de etapas, os usuários podem precisar de permissões adicionais para criar e gerenciar CloudWatch alarmes.

> **Tip**
>
> Para ajudar a descobrir quais ações de API são necessárias para realizar tarefas no console, é possível usar um serviço como o AWS CloudTrail. Para obter mais informações, consulte o Guia do usuário do AWS CloudTrail.

A política baseada em identidade a seguir concede permissões para configurar políticas de escalação para o Spot Fleet. Além das permissões do IAM para o Spot Fleet, o usuário do console que acessa as configurações de escalabilidade da frota a partir do EC2 console da Amazon deve ter as permissões apropriadas para os serviços que oferecem suporte à escalabilidade dinâmica.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:*",
                "ec2:DescribeSpotFleetRequests",
                "ec2:ModifySpotFleetRequest",
                "cloudwatch:DeleteAlarms",
                "cloudwatch:DescribeAlarmHistory",
                "cloudwatch:DescribeAlarms",
```

```
                "cloudwatch:DescribeAlarmsForMetric",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics",
                "cloudwatch:PutMetricAlarm",
                "cloudwatch:DisableAlarmActions",
                "cloudwatch:EnableAlarmActions",
                "sns:CreateTopic",
                "sns:Subscribe",
                "sns:Get*",
                "sns:List*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-
service-role/ec2.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_EC2SpotFleetRequest",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName":"ec2.application-
autoscaling.amazonaws.com"
                }
            }
        }
    ]
}
```

Essa política permite que os usuários do console visualizem e modifiquem políticas de escalabilidade no EC2 console da Amazon e criem e gerenciem CloudWatch alarmes no CloudWatch console.

É possível ajustar as ações da API para limitar o acesso do usuário. Por exemplo, substituir `application-autoscaling:*` por `application-autoscaling:Describe*` significa que o usuário terá acesso somente leitura.

Você também pode ajustar as CloudWatch permissões conforme necessário para limitar o acesso do usuário aos CloudWatch recursos. Para obter mais informações, consulte Permissões necessárias para o CloudWatch console no Guia CloudWatch do usuário da Amazon.

Solução de problemas de acesso ao Application Auto Scaling

Se você encontrar `AccessDeniedException` ou dificuldades semelhantes ao trabalhar com o Application Auto Scaling, consulte as informações nesta seção.

Não tenho autorização para executar uma ação no Application Auto Scaling

Se você receber um `AccessDeniedException` ao chamar uma operação de AWS API, isso significa que as credenciais AWS Identity and Access Management (IAM) que você está usando não têm as permissões necessárias para fazer essa chamada.

O exemplo de erro a seguir ocorre quando o usuário `mateojackson` tenta visualizar detalhes sobre um alvo escalável, mas não tem permissão para `application-autoscaling:DescribeScalableTargets`.

```
An error occurred (AccessDeniedException) when calling the DescribeScalableTargets
 operation: User: arn:aws:iam::123456789012:user/mateojackson is not authorized to
 perform: application-autoscaling:DescribeScalableTargets
```

Se você receber esse erro ou erros semelhantes, entre em contato com o administrador para obter assistência.

Um administrador da sua conta precisará garantir que você tenha permissões para acessar todas as ações de API que o Application Auto Scaling usa para acessar recursos no serviço de destino e. CloudWatch Existem diferentes permissões necessárias, dependendo dos recursos com os quais você está trabalhando. O Application Auto Scaling requer permissões para criar uma função vinculada ao serviço na primeira vez um usuário configura escalabilidade para um determinado recurso.

Eu sou um administrador e minha política do IAM retornou um erro ou não está funcionando conforme esperado

Além das ações do Application Auto Scaling, suas políticas do IAM devem conceder permissões para chamar o serviço de destino e. CloudWatch Se um usuário ou uma aplicação não tiver essas permissões adicionais, seu acesso poderá ser negado inesperadamente. Para escrever políticas do IAM para usuários e aplicações em suas contas, consulte as informações em Políticas baseadas em identidade do Application Auto Scaling.

Para obter informações sobre como a validação é executada, consulte Validação de permissões para chamadas de API do Application Auto Scaling em recursos de destino.

Observe que alguns problemas de permissão também podem ser causados por um problema com a criação das funções vinculadas ao serviço usadas pelo Application Auto Scaling. Para obter mais informações sobre a criação dessas funções vinculadas a serviços, consulte Funções vinculadas ao serviço necessárias para o Application Auto Scaling.

Validação de permissões para chamadas de API do Application Auto Scaling em recursos de destino

Fazer solicitações autorizadas às ações da API Application Auto Scaling exige que o chamador da API tenha permissões para acessar AWS recursos no serviço de destino e no. CloudWatch O Application Auto Scaling valida as permissões para solicitações associadas ao serviço de destino e CloudWatch antes de prosseguir com a solicitação. Para fazer isso, emitimos uma série de chamadas para validar as permissões do IAM nos recursos de destino. Quando uma resposta é retornada, ela é lida pelo Application Auto Scaling. Se as permissões do IAM não permitirem uma determinada ação, haverá fallha na solicitação do Application Auto Scaling, que retornará um erro ao usuário contendo informações sobre a permissão ausente. Isso garante que a configuração de escalabilidade que o usuário deseja implantar funcione conforme pretendido e que um erro útil seja retornado se a solicitação falhar.

Como exemplo de como isso funciona, as informações a seguir fornecem detalhes sobre como o Application Auto Scaling realiza validações de permissões com o Aurora e. CloudWatch

Quando um usuário chama a API `RegisterScalableTarget` em um cluster de bancos de dados do Aurora, o Application Auto Scaling realiza todas as verificações a seguir para confirmar que o usuário tem as permissões necessárias (em negrito).

- rds:create DBInstance: para determinar se o usuário tem essa permissão, enviamos uma solicitação para a operação da `CreateDBInstance` API, tentando criar uma instância de banco de dados com parâmetros inválidos (ID de instância vazia) no cluster de banco de dados Aurora que o usuário especificou. Para um usuário autorizado, a API retorna uma resposta de código de erro `InvalidParameterValue` depois de auditar a solicitação. No entanto, para um usuário não autorizado, obtemos um erro `AccessDenied` e a solicitação do Application Auto Scaling falha, com um erro `ValidationException`para o usuário que lista as permissões ausentes.
- rds:delete DBInstance: enviamos um ID de instância vazio para a operação da `DeleteDBInstance` API. Para um usuário autorizado, essa solicitação resulta em um erro `InvalidParameterValue`. Para um usuário não autorizado, isso resulta em `AccessDenied` e envia uma exceção de validação para o usuário (mesmo tratamento descrito no primeiro marcador).

- rds:AddTagsToResource: Como a operação da `AddTagsToResource` API exige um nome de recurso da Amazon (ARN), é necessário especificar um recurso "fictício" usando um ID de conta inválido (12345) e um ID de instância fictício () para criar o ARN (non-existing-db). `arn:aws:rds:us-east-1:12345:db:non-existing-db` Para um usuário autorizado, essa solicitação resulta em um erro `InvalidParameterValue`. Para um usuário não autorizado, isso resulta em `AccessDenied` e envia uma exceção de validação para o usuário.
- rds:describeDBClusters: descrevemos o nome do cluster do recurso que está sendo registrado para escalonamento automático. Para um usuário autorizado, obtemos um resultado de descrição válido. Para um usuário não autorizado, isso resulta em `AccessDenied` e envia uma exceção de validação para o usuário.
- rds:describe DBInstances: chamamos a `DescribeDBInstances` API com um `db-cluster-id` filtro que filtra o nome do cluster fornecido pelo usuário para registrar o destino escalável. Para um usuário autorizado, temos permissão para descrever todas as instâncias de banco de dados no cluster do banco de dados. Para um usuário não autorizado, essa chamada resulta em `AccessDenied` e envia uma exceção de validação para o usuário.
- cloudwatch:PutMetricAlarm: Chamamos a `PutMetricAlarm` API sem nenhum parâmetro. Como o nome do alarme está ausente, a solicitação resulta em `ValidationError` para um usuário autorizado. Para um usuário não autorizado, isso resulta em `AccessDenied` e envia uma exceção de validação para o usuário.
- cloudwatch:DescribeAlarms: Chamamos a `DescribeAlarms` API com o valor do número máximo de registros definido como 1. Para um usuário autorizado, esperamos informações sobre um alarme na resposta. Para um usuário não autorizado, essa chamada resulta em `AccessDenied` e envia uma exceção de validação para o usuário.
- cloudwatch:DeleteAlarms: Semelhante ao `PutMetricAlarm` descrito acima, não fornecemos parâmetros para `DeleteAlarms` solicitar. Como o nome do alarme está ausente da solicitação, essa chamada falhará com um `ValidationError` para um usuário autorizado. Para um usuário não autorizado, isso resulta em `AccessDenied` e envia uma exceção de validação para o usuário.

Sempre que qualquer um desses erros de validação ocorrer, ele será registrado. Você pode tomar medidas para identificar manualmente quais chamadas falharam na validação usando AWS CloudTrail. Para obter mais informações, consulte o Guia do usuário do AWS CloudTrail.

> **Note**
>
> Se você receber alertas sobre o uso de eventos do Application Auto Scaling CloudTrail, esses alertas incluirão as chamadas do Application Auto Scaling para validar as permissões

do usuário por padrão. Para filtrar esses alertas, use o campo `invokedBy`, que conterá `application-autoscaling.amazonaws.com` para essas verificações de validação.

Acessar o Application Auto Scaling usando endpoints de interface da VPC

Você pode usar AWS PrivateLink para criar uma conexão privada entre sua VPC e o Application Auto Scaling. Você pode acessar o Application Auto Scaling como se estivesse em sua VPC, sem o uso de um gateway de internet, dispositivo NAT, conexão VPN ou conexão. AWS Direct Connect As instâncias na VPC não precisam de endereços IP públicos para acessar o Application Auto Scaling.

Estabeleça essa conectividade privada criando um endpoint de interface, habilitado pelo AWS PrivateLink. Criaremos um endpoint de interface de rede em cada sub-rede que você habilitar para o endpoint de interface. Tratam-se de interfaces de rede gerenciadas pelo solicitante que servem como ponto de entrada para o tráfego destinado ao Application Auto Scaling.

Para obter mais informações, consulte Acesso Serviços da AWS por meio AWS PrivateLink do AWS PrivateLink Guia.

Conteúdo

- Criar um VPC endpoint de interface
- Criar uma política de endpoint da VPC

Criar um VPC endpoint de interface

Crie um endpoint para o Application Auto Scaling usando o seguinte nome de serviço:

```
com.amazonaws.region.application-autoscaling
```

Para obter mais informações, consulte Acessar um AWS serviço usando uma interface VPC endpoint no Guia.AWS PrivateLink

Não é necessário alterar nenhuma outra configuração. O Application Auto Scaling chama outros AWS serviços usando endpoints de serviço ou endpoints VPC de interface privada, os que estiverem em uso.

Criar uma política de endpoint da VPC

Você pode anexar uma política ao endpoint da VPC para controlar o aceso à API do Application Auto Scaling. A política especifica:

- O principal que pode executar ações.
- As ações que podem ser executadas.
- O recurso no qual as ações podem ser executadas.

O exemplo a seguir mostra uma política de VPC endpoint que nega a todos permissão para excluir uma política de escalabilidade por meio do endpoint. O exemplo de política também concede a todos permissão para executar todas as outras ações.

```
{
   "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        },
        {
            "Action": "application-autoscaling:DeleteScalingPolicy",
            "Effect": "Deny",
            "Resource": "*",
            "Principal": "*"
        }
    ]
}
```

Para obter mais informações, consulte VPC endpoint policies (Políticas de endpoint da VPC) no AWS PrivateLink Guide (Guia do).

Resiliência no Application Auto Scaling

A infraestrutura AWS global é construída em torno de AWS regiões e zonas de disponibilidade.

AWS As regiões fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, conectadas a redes de baixa latência, alta taxa de transferência e alta redundância.

Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre AWS regiões e zonas de disponibilidade, consulte infraestrutura AWS global.

Segurança da infraestrutura no Application Auto Scaling

Como um serviço gerenciado, o Application Auto Scaling é protegido pela segurança de rede AWS global. Para obter informações sobre serviços AWS de segurança e como AWS proteger a infraestrutura, consulte AWS Cloud Security. Para projetar seu AWS ambiente usando as melhores práticas de segurança de infraestrutura, consulte Proteção de infraestrutura no Security Pillar AWS Well-Architected Framework.

Você usa chamadas de API AWS publicadas para acessar o Application Auto Scaling pela rede. Os clientes devem oferecer compatibilidade com:

- Transport Layer Security (TLS). Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Conjuntos de criptografia com perfect forward secrecy (PFS) como DHE (Ephemeral Diffie-Hellman) ou ECDHE (Ephemeral Elliptic Curve Diffie-Hellman). A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou é possível usar o AWS Security Token Service (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Validação da compatibilidade para o Application Auto Scaling

Para saber se um AWS service (Serviço da AWS) está dentro do escopo de programas de conformidade específicos, consulte Serviços da AWS Escopo por Programa de Conformidade Serviços da AWS e escolha o programa de conformidade em que você está interessado. Para obter informações gerais, consulte Programas de AWS conformidade Programas AWS de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte Baixar relatórios em AWS Artifact .

Sua responsabilidade de conformidade ao usar Serviços da AWS é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis e regulamentações aplicáveis. AWS fornece os seguintes recursos para ajudar na conformidade:

- Governança e conformidade de segurança: esses guias de implementação de solução abordam considerações sobre a arquitetura e fornecem etapas para implantar recursos de segurança e conformidade.
- Referência de serviços qualificados para HIPAA: lista os serviços qualificados para HIPAA. Nem todos Serviços da AWS são elegíveis para a HIPAA.
- AWS Recursos de https://aws.amazon.com/compliance/resources/ de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- AWS Guias de conformidade do cliente — Entenda o modelo de responsabilidade compartilhada sob a ótica da conformidade. Os guias resumem as melhores práticas de proteção Serviços da AWS e mapeiam as diretrizes para controles de segurança em várias estruturas (incluindo o Instituto Nacional de Padrões e Tecnologia (NIST), o Conselho de Padrões de Segurança do Setor de Cartões de Pagamento (PCI) e a Organização Internacional de Padronização (ISO)).
- Avaliação de recursos com regras no Guia do AWS Config desenvolvedor — O AWS Config serviço avalia o quão bem suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.
- AWS Security Hub— Isso AWS service (Serviço da AWS) fornece uma visão abrangente do seu estado de segurança interno AWS. O Security Hub usa controles de segurança para avaliar os recursos da AWS e verificar a conformidade com os padrões e as práticas recomendadas do setor de segurança. Para obter uma lista dos serviços e controles aceitos, consulte a Referência de controles do Security Hub.
- Amazon GuardDuty — Isso AWS service (Serviço da AWS) detecta possíveis ameaças às suas cargas de trabalho Contas da AWS, contêineres e dados monitorando seu ambiente em busca de atividades suspeitas e maliciosas. GuardDuty pode ajudá-lo a atender a vários requisitos de conformidade, como o PCI DSS, atendendo aos requisitos de detecção de intrusões exigidos por determinadas estruturas de conformidade.
- AWS Audit Manager— Isso AWS service (Serviço da AWS) ajuda você a auditar continuamente seu AWS uso para simplificar a forma como você gerencia o risco e a conformidade com as regulamentações e os padrões do setor.

Cotas do Application Auto Scaling

Você Conta da AWS tem cotas padrão, anteriormente chamadas de limites, para cada um. AWS service (Serviço da AWS) A menos que especificado de outra forma, cada cota é específica da região . Você pode solicitar o aumento de algumas cotas, porém, algumas delas não podem ser aumentadas.

Para visualizar as cotas do Application Auto Scaling, abra o console do Service Quotas. No painel de navegação, escolha AWS services (serviços da) e selecione Application Auto Scaling.

Para solicitar o aumento da cota, consulte Solicitar um aumento de cota no Guia do usuário do Service Quotas.

Você Conta da AWS tem as seguintes cotas relacionadas ao Application Auto Scaling.

Name	Padrão	Ajustável
Destinos escaláveis por tipo de recurso	Amazon DynamoDB: 5.000 | Amazon ECS: 3.000 | Amazon Keyspaces: 1.500 | Outros tipos de recurso: 500	Sim
Políticas de escalabilidade por destino escalável (políticas de escalabilidade em etapas e rastreamento de destino)	50	Não
Ações programadas por destino escalável	200	Não
Ajustes de etapa por política de escalabilidade de etapa	20	Sim

Tenha em mente as cotas de serviço ao aumentar suas cargas de trabalho. Por exemplo, quando você atingir o número máximo de unidades de capacidade permitidas por um serviço, a expansão será interrompida. Se a demanda cair e a capacidade atual diminuir, o Application Auto Scaling poderá aumentar novamente. Para evitar atingir o limite da capacidade novamente, é possível solicitar um aumento. Cada serviço tem suas próprias cotas padrão para a capacidade máxima

do recurso. Para obter informações sobre as cotas padrão para outras ofertas da Amazon Web Services, consulte Endpoints e cotas de serviços no Referência geral da Amazon Web Services.

Histórico de documentos do Application Auto Scaling

A tabela a seguir descreve adições importantes feitas na documentação do Application Auto Scaling a partir de janeiro de 2018. Para receber notificações sobre atualizações dessa documentação, você pode se inscrever em o feed RSS.

Alteração	Descrição	Data
Adicionar suporte para clusters ElastiCache Memcached	Use o Application Auto Scaling para escalar horizontalmente o número de nós de um cluster Memcached. Para obter mais informações, consulte ElastiCache Application Auto Scaling.	10 de abril de 2025
AWS atualizações de políticas gerenciadas	O Application Auto Scaling atualizou a políticaAWSApplicationAutoscalingElastiCacheRGPolicy .	10 de abril de 2025
Alterações do guia	O novo tópico no Guia do Usuário do Application Auto Scaling ajuda você a começar a usar a escalabilidade preditiva com o Application Auto Scaling. Consulte Escalabilidade preditiva do Application Auto Scaling.	21 de novembro de 2024
AWS atualizações de políticas gerenciadas	O Application Auto Scaling atualizou a políticaAWSApplicationAutoscalingECSServicePolicy .	21 de novembro de 2024
Adicione suporte para um pool de WorkSpaces	Use o Application Auto Scaling para escalar um pool de.	27 de junho de 2024

	WorkSpaces Para obter mais informações, consulte Amazon WorkSpaces e Application Auto Scaling. O tópico Atualizações do Application Auto Scaling para políticas AWS gerenciadas foi atualizado para listar uma nova política gerenciada para a integração com. WorkSpaces	
Alterações do guia	Atualização da entrada Número máximo de destinos escaláveis por tipo de recurso na documentação sobre cotas. Consulte Cotas do Application Auto Scaling.	16 de janeiro de 2024
Support para componentes de inferência de SageMaker IA	Use o Application Auto Scaling para escalar cópias de um componente de inferência.	29 de novembro de 2023
AWS atualizações de políticas gerenciadas	O Application Auto Scaling atualizou a política `AWSApplicationAutoscalingSageMakerEndpointPolicy` .	13 de novembro de 2023
Support para simultaneidade SageMaker provisionada com IA Serverless	Use o Application Auto Scaling para escalar a simultaneidade provisionada de um endpoint sem servidor.	9 de maio de 2023

Categorização de destinos escaláveis usando etiquetas	Atualmente, é possível atribuir metadados aos destinos escaláveis do Application Auto Scaling na forma de etiquetas. Consulte Suporte de marcação para o Application Auto Scaling.	20 de março de 2023
Support para matemática CloudWatch métrica	Agora você pode usar a matemática métrica ao criar políticas de dimensionamento de rastreamento de destino. Com a matemática métrica, você pode consultar várias CloudWatch métricas e usar expressões matemáticas para criar novas séries temporais com base nessas métricas. Consulte Crie uma política de escalabilidade de rastreamento de destino para o Application Auto Scaling usando matemática em métricas.	14 de março de 2023
Motivos para não escalar	Agora você pode recuperar os motivos legíveis por máquina para o Application Auto Scaling não escalar seus recursos usando a API do Application Auto Scaling. Consulte Atividades de escalação para o Application Auto Scaling.	4 de janeiro de 2023

Alterações do guia	Atualização da entrada Número máximo de destinos escaláveis por tipo de recurso na documentação sobre cotas. Consulte Cotas do Application Auto Scaling.	6 de maio de 2022
Adicionar suporte a clusters do Amazon Neptune	Use o Application Auto Scaling para escalar o número de réplicas em um cluster de banco de dados do Amazon Neptune. Para obter mais informações, consulte Amazon Neptune e Application Auto Scaling. O tópico Atualizações do Application Auto Scaling para políticas gerenciadas pela AWS foi atualizado para listar uma nova política gerenciada para a integração com o Neptune.	6 de outubro de 2021
O Application Auto Scaling agora relata alterações em suas AWS políticas gerenciadas	A partir de 19 de agosto de 2021, as alterações nas políticas gerenciadas são relatadas no tópico Atualizações das políticas AWS gerenciadas do Application Auto Scaling. A primeira alteração listada é a adição das permissões necessárias para ElastiCache (Redis OSS).	19 de agosto de 2021

Adicionar suporte para grupos de ElastiCache replicação (Redis OSS)	Use o Application Auto Scaling para escalar o número de grupos de nós e o número de réplicas por grupo de nós para um grupo de replicaçã o ElastiCache (cluster) (Redis OSS). Para obter mais informações, consulte ElastiCache (Redis OSS) e Application Auto Scaling.	19 de agosto de 2021
Alterações do guia	Novos tópicos do IAM no Manual do usuário do Applicati on Auto Scaling ajudam você a solucionar problemas de acesso ao Application Auto Scaling. Para obter mais informações, consulte Gerenciamento de Identidad e e Acesso para o Applicati on Auto Scaling. Também foram adicionados novos exemplos de políticas de permissões do IAM para ações nos serviços de destino e na Amazon CloudWatch. Para obter mais informações, consulte Exemplos de políticas para trabalhar com o AWS CLI ou com um SDK.	23 de fevereiro de 2021

Adicionar compatibilidade com fusos horários locais	Agora você pode criar ações programadas no fuso horário local da zona. Se o fuso horário seguir o horário de verão, ele se ajustará automaticamente ao horário de verão (DST). Para obter mais informações, consulte Escalabilidade programada.	2 de fevereiro de 2021
Alterações do guia	Um novo tutorial no Manual do usuário do Application Auto Scaling ajuda você a entender como usar políticas de dimensionamento de monitoramento do objetivo e escalabilidade programad a para aumentar a disponibi lidade de sua aplicação ao usar o Application Auto Scaling.	15 de outubro de 2020
Adicionar compatibilidade com o cluster de armazenam ento do Amazon Managed Streaming for Apache Kafka	Use uma política de escalabil idade de monitoramento do objetivo para expandir a quantidade de armazenam ento do agente associada a um cluster do Amazon MSK.	30 de setembro de 2020
Adicionar suporte para endpoints de identificação de entidade do Amazon Comprehend	Use o Application Auto Scaling para escalar o número de unidades de inferênci a provisionadas para seus endpoints de reconhecimento de entidade do Amazon Comprehend.	28 de setembro de 2020

Adicionar suporte para tabelas do Amazon Keyspaces (for Apache Cassandra	Use o Application Auto Scaling para escalar o throughput provisionado (capacidade de leitura e gravação) de uma tabela do Amazon Keyspaces.	23 de abril de 2020
Novo capítulo “Segurança”	Um novo capítulo Security (Segurança) no Manual do usuário do Application Auto Scaling ajuda a entender como aplicar o modelo de responsabilidade compartilhada ao usar o Application Auto Scaling. Como parte dessa atualização, o capítulo do manual do usuário "Authentication and Access Control" (Autenticação e controle de acesso) foi substituído por uma seção nova e mais simples, Identity and Access Management para o Application Auto Scaling (Gerenciamento de Identidade e Acesso para o Application Auto Scaling).	16 de janeiro de 2020
Atualizações menores	Várias melhorias e correções.	15 de janeiro de 2020

Adicionar funcionalidade de notificação	O Application Auto Scaling agora envia eventos para a Amazon EventBridge e notificações para você AWS Health Dashboard quando determinadas ações ocorrem. Para obter mais informações, consulte Monitoramento do Application Auto Scaling.	20 de dezembro de 2019
Adicionar suporte para AWS Lambda funções	Use o Application Auto Scaling para escalar a simultaneidade provisionada de uma função do Lambda.	3 de dezembro de 2019
Adicionr compatibilidade com endpoints de classificação de documentos do Amazon Comprehend	Use o Application Auto Scaling para escalar a capacidade de throughput de um endpoint de classificação de documentos do Amazon Comprehend.	25 de novembro de 2019
Adicione suporte AppStream 2.0 para políticas de escalabilidade de rastreamento de metas	Use políticas de escalabilidade de rastreamento de metas para escalar o tamanho de uma frota AppStream 2.0.	25 de novembro de 2019
Suporte para endpoints da VPC da Amazon	Agora você pode estabelecer uma conexão privada entre sua VPC e o Application Auto Scaling. Para ver as considerações e instruções de migração, consulte Application Auto Scaling e endpoints da VPC de interface.	22 de novembro de 2019

Suspender e retomar a escalabilidade	Adicionado suporte para suspender e retomar a escalabilidade. Para obter mais informações, consulte Suspender e retomar a escalabilidade do Application Auto Scaling.	29 de agosto de 2019
Alterações do guia	A documentação do Applicati on Auto Scaling nas seções Scheduled scaling (Escalabi lidade programada), Step scaling policies (Políticas de escalabilidade em etapas) e Target tracking scaling policies (Políticas de dimension amento com monitoramento do objetivo) foi aprimorada.	11 de março de 2019
Adicionar compatibilidade com recursos personalizados	Use o Application Auto Scaling para escalar recursos personalizados fornecidos por suas próprias aplicações ou serviços. Para obter mais informações, consulte nosso GitHubrepositório.	9 de julho de 2018
Adicione suporte para variantes de endpoint de SageMaker IA	Use o Application Auto Scaling para escalar o número de instâncias de endpoint provisionadas para uma variante.	28 de fevereiro de 2018

A tabela a seguir descreve alterações importantes feitas na documentação do Application Auto Scaling antes de janeiro de 2018.

Alteração	Descrição	Data
Adicionar suporte para as réplicas do Aurora	Use o Application Auto Scaling para escalar a quantidade desejada. Para obter mais informações, consulte Usar o Auto Scaling do Amazon Aurora com réplicas do Aurora no Manual do usuário do Amazon RDS.	17 de novembro de 2017
Adicionar suporte para a escalabilidade programadas	Use a escalabilidade programada para escalar recursos em horários ou intervalos predefinidos específicos. Para obter mais informações, consulte Escalabilidade programda do Application Auto Scaling.	8 de novembro de 2017
Adicionar suporte para as políticas de escalabilidade de rastreamento de destino	Use políticas de escalabil idade de rastreamento de destino para configurar escalabilidade dinâmica para o seu aplicativo em apenas algumas etapas simples. Para obter mais informaçõ es, consulte Políticas de dimensionamento com monitoramento do objetivo para o Application Auto Scaling.	12 de julho de 2017
Adicionar compatibilidade com capacidade de leitura e gravação provisionada para	Use o Application Auto Scaling para escalar o throughpu t provisionado (capacida de de leitura e gravação).	14 de junho de 2017

Alteração	Descrição	Data
tabelas e índices secundários globais do DynamoDB	Para obter mais informaçõ es, consulte Como gerenciar a capacidade de throughpu t com a autoescalabilidade do DynamoDB no Guia do desenvolvedor do Amazon DynamoDB.	
Adicione suporte para frotas AppStream 2.0	Use o Application Auto Scaling para escalar o tamanho da frota. Para obter mais informações, consulte Fleet Auto Scaling for AppStream 2.0 no Amazon AppStream 2.0 Administration Guide.	23 de março de 2017
Adicionar compatibilidade com clusters do Amazon EMR	Use o Application Auto Scaling para escalar os nós principais e os nós de tarefa. Para obter mais informações, consulte Usar escalabilidade automátic a no Amazon EMR no Guia de gerenciamento do Amazon EMR.	18 de novembro de 2016
Adicionar suporte às frotas spot	Use o Application Auto Scaling para escalar a capacidad e de destino. Para obter mais informações, consulte Escalabilidade automática para frota spot no Guia do EC2 usuário da Amazon.	1 de setembro de 2016

Alteração	Descrição	Data
Adicionar compatibilidade com serviços da Amazon ECS	Use o Application Auto Scaling para escalar a quantidade desejada. Para obter mais informações, consulte Usar escalabilidade automática no Guia do desenvolvedor do Amazon Elastic Container Service.	9 de agosto de 2016

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.