aws

Manual do usuário

Application Auto Scaling

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Application Auto Scaling : Manual do usuário

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre clientes ou que deprecie ou desprestigie a Amazon. Todas as outras marcas comerciais que não são propriedade da Amazon pertencem aos respectivos proprietários, os quais podem ou não ser afiliados, estar conectados ou ser patrocinados pela Amazon.

Table of Contents

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 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 Elastic Container Service (ECS)
- ElastiCache para clusters Redis (grupos de replicação)
- 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
- SageMaker componentes de inferência
- SageMaker Concorrência provisionada sem servidor
- Solicitações de frota spot
- 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 mais informações sobre como escalar sua frota de instâncias do Amazon EC2 usando grupos do Auto Scaling, consulte Manual do usuário do Amazon EC2 Auto 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.

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. Se você se inscreveu em uma AWS conta, acesse Application Auto Scaling fazendo login no. AWS Management Console Abra o console do serviço para um dos recursos listados na introdução. 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 ter mais informações, consulte AWS serviços 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 Configurar a AWS CLI. Para obter mais informações, consulte escalabilidade automática de aplicações na Referência de comando da AWS CLI .
- 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 AWS Tools for Windows PowerShell. Para obter mais informações, consulte Referência de Cmdlets do AWS Tools for PowerShell.

- AWS SDKs — Fornece operações de API específicas para cada 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 AWS SDKs.
- 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 ter mais informações, consulte Criar recursos do Application Auto Scaling com o AWS CloudFormation.

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

Comece a usar o 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 alterações fora de faixa diretamente no recurso subjacente que está fora desse intervalo, 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 monitorar uma métrica específica do 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 autoescalabilidade são publicadas pelo serviço de destino, mas você também pode publicar sua própria métrica no CloudWatch e, em seguida, 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

AWS serviços 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 escalabilidade de rastreamento de destino: 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%. O Application Auto Scaling executa ou encerra instâncias do EC2 conforme necessário para manter a utilização agregada da CPU em todos os servidores em 50%.

AWS serviços 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 os AWS SDKs.

CloudFormation access — Você pode configurar um AWS serviço compatível para iniciar o escalonamento automático usando um modelo de AWS CloudFormation pilha. Para ter mais informações, consulte Criar recursos do Application Auto Scaling com 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	Sim	Sim	Sim	Sim
Frota spot	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 do escalonamento agendado a partir do 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

ter 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 um dos SDKs, você pode usar AWS 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

Se você está apenas começando a usar o Application Auto Scaling, você pode encontrar mais informações úteis sobre como escalar seus recursos AppStream 2.0 na documentação a seguir:

Fleet Auto Scaling for AppStream 2.0 no Guia de administração do Amazon AppStream 2.0

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 ter 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 um dos SDKs, você pode usar AWS 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

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

Usar a autoescalabilidade do Amazon Aurora com réplicas do Aurora no Manual do usuário do Amazon RDS

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 ter 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 um dos SDKs, você pode usar AWS 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

Se você está apenas começando a usar o Application Auto Scaling, você pode encontrar mais informações úteis sobre a escalabilidade de seus recursos do Amazon Comprehend na seguinte documentação:

Escalabilidade automática com endpoints no Guia do desenvolvedor do Amazon Comprehend

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 ter 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 um dos SDKs, você pode usar AWS as seguintes opções:

- AWS CLI:

  Chame o register-scalable-targetcomando para obter a capacidade de gravação de uma tabela. O exemplo a seguir inscreve a capacidade de gravação provisionada de uma tabela chamada `my-table`, 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 dynamodb \
  --scalable-dimension dynamodb:table:WriteCapacityUnits \
  --resource-id table/my-table \
  --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"
}
```

  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 um mínimo cinco unidades de capacidade de leitura e um máximo de dez 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 obtiver êxito, 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 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 dynamodb \
  --scalable-dimension dynamodb:index:WriteCapacityUnits \
  --resource-id table/my-table/index/my-table-index \
  --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"
}
```

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 um mínimo de cinco unidades de capacidade de leitura e um máximo de dez 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 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 como escalar seus recursos do DynamoDB na seguinte documentação:

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

Você também pode encontrar um tutorial para escalonamento programado. Tutorial: comece a usar a escalabilidade programada usando a AWS CLI Nesse tutorial, você aprende as etapas básicas para configurar a escalabilidade para que sua tabela do DynamoDB seja escalada em horários programados.

Amazon ECS e Application Auto Scaling

É possível escalar os serviços do ECS usando políticas de dimensionamento com monitoramento do objetivo, políticas de escalabilidade em 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 ter 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 um dos SDKs, você pode usar AWS 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, você pode encontrar mais informações úteis sobre a escalabilidade de seus recursos do Amazon ECS na seguinte documentação:

- Escalabilidade automática de serviços no Guia do desenvolvedor do Amazon Elastic Container Service
- Configurando o escalonamento automático de serviços no Guia de melhores práticas do Amazon Elastic Container Service

> **Note**
>
> Para obter instruções sobre como suspender os processos de escalabilidade enquanto as implantações do Amazon ECS estão em andamento, consulte a seguinte documentação: Escalabilidade automática de serviços e implantações no Guia do desenvolvedor do Amazon Elastic Container Service

ElastiCache para Redis e Application Auto Scaling

Você pode escalar ElastiCache para grupos de replicação do Redis usando políticas de escalabilidade de rastreamento de destino e escalabilidade programada.

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

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 ter 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-se em grupos ElastiCache de replicação do Redis como destinos 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 replicação. ElastiCache 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 um dos SDKs, você pode usar AWS 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 `mycluster`, 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/mycluster \
   --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 inscreve o número desejado de réplicas por grupo de nós para um grupo de replicação chamado `mycluster`, 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:Replicas \
   --resource-id replication-group/mycluster \
   --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

Se você está apenas começando a usar o Application Auto Scaling, você pode encontrar mais informações úteis sobre como escalar seus ElastiCache recursos na documentação a seguir:

Auto Scaling ElastiCache para clusters Redis no Guia do usuário do Amazon ElastiCache for Redis

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 ter 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 um dos SDKs, você pode usar AWS 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

Se você está apenas começando a usar o Application Auto Scaling, você pode encontrar mais informações úteis sobre a escalabilidade de seus recursos do Amazon Keyspaces na seguinte documentação:

Gerenciando a capacidade de processamento com o escalonamento automático do Amazon Keyspaces no Guia do desenvolvedor do Amazon Keyspaces (para Apache Cassandra)

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 ter 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 um dos SDKs, você pode usar AWS 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, você pode encontrar mais informações úteis sobre como escalar suas funções Lambda na seguinte documentação:

- Configurando a simultaneidade provisionada no Guia do desenvolvedor 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 ter 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 um dos SDKs, você pode usar AWS 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

Se você está apenas começando a usar o Application Auto Scaling, você pode encontrar mais informações úteis sobre a escalabilidade de seus recursos do Amazon MSK na seguinte documentação:

Escalabilidade automática 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

ter 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 um dos SDKs, você pode usar AWS 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

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

Escalabilidade automática do número de réplicas em um cluster de banco de dados do Amazon Neptune no Guia do usuário do Neptune

Amazon SageMaker e Application Auto Scaling

Você pode escalar variantes de SageMaker endpoint, 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 se integrar SageMaker ao Application Auto Scaling.

Criação de uma função vinculada ao serviço para o SageMaker

A função vinculada ao serviço a seguir é criada automaticamente em você Conta da AWS ao registrar SageMaker 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 ter 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 SageMaker endpoint como destinos 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 SageMaker modelo (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 SageMaker console, registrará SageMaker automaticamente uma meta escalável para você.

Se quiser configurar o escalonamento automático usando a AWS CLI ou um dos SDKs, você pode usar AWS 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 SageMaker console, registrará SageMaker 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 SageMaker recursos no Amazon SageMaker Developer Guide:

- Dimensione automaticamente os SageMaker modelos da Amazon
- Dimensione automaticamente a simultaneidade provisionada para um endpoint sem servidor
- Defina políticas de escalonamento automático para implantações de endpoints de vários modelos
- Dimensione automaticamente um endpoint assíncrono

> **Note**
>
> Em 2023, SageMaker introduziu novos recursos de inferência baseados em endpoints de inferência em tempo real. Você cria um SageMaker endpoint 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 SageMaker hospedagem que você pode usar para implantar um modelo em um endpoint. Para obter informações sobre escalabilidade de componentes de inferência, consulte A Amazon SageMaker 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 no blog. AWS

Frota spot do Amazon EC2 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

ter 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 um dos SDKs, você pode usar AWS 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

Se você está apenas começando a usar o Application Auto Scaling, você pode encontrar mais informações úteis sobre como escalar sua frota spot na seguinte documentação:

Escalabilidade automática para frota spot no Manual do usuário do Amazon EC2

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 ter 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 um dos SDKs, você pode usar AWS 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, você pode encontrar mais informações úteis sobre como escalar seus recursos personalizados na documentação a seguir:

GitHubrepositório

Configurar o uso do Application Auto Scaling

Conclua as tarefas nesta seção para configurar o Application Auto Scaling pela primeira vez:

Tópicos

- Registre-se na AWS
- Configurar a AWS CLI
- Usar o AWS CloudShell para trabalhar com o Application Auto Scaling na linha de comando

Registre-se na AWS

Se você ainda não tem Conta da AWS, siga as etapas a seguir para criar um.

Para se cadastrar em uma Conta da AWS

1. Abra https://portal.aws.amazon.com/billing/signup.
2. Siga as instruções online.

   Parte do procedimento de inscrição envolve receber uma chamada telefônica e inserir um código de verificação no teclado do telefone.

   Quando você se cadastra em uma Conta da AWS, um Usuário raiz da conta da AWS é criado. O usuário raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua acesso administrativo a um usuário administrativo e use somente o usuário raiz para realizar as tarefas que exigem acesso do usuário raiz.

Usar o Application Auto Scaling em Regiões da AWS

O Application Auto Scaling está disponível em várias Regiões da AWS. Uma Conta da AWS global permite trabalhar com recursos na maioria das regiões. Ao usar o Application Auto Scaling com recursos nas regiões da China, lembre-se de que você deve ter uma conta separada da Amazon Web Services (China). Além disso, há algumas diferenças na forma como o Application Auto Scaling é implementado. Para obter mais informações sobre como usar o Application Auto Scaling nas regiões da China, consulte Application Auto Scaling na China.

Após configurar sua Conta da AWS, prossiga para o próximo tópico: Configurar a AWS CLI.

Configurar a AWS CLI

A AWS Command Line Interface (AWS CLI) é uma ferramenta de desenvolvedor unificada para gerenciar os serviços da AWS, incluindo o Application Auto Scaling. Siga as etapas para fazer download e configurar a AWS CLI.

Para configurar a AWS CLI

1. Baixe, instale e configure a versão 1 ou 2 da AWS CLI. A mesma funcionalidade do Application Auto Scaling está disponível nas versões 1 e 2. Para obter instruções, consulte os seguintes tópicos no Manual do usuário do AWS Command Line Interface:

   AWS CLI versão 1

   - Instalar, atualizar e desinstalar a AWS CLI
   - Configurar a AWS CLI

   AWS CLI versão 2

   - Instalar ou atualizar a versão mais recente da AWS CLI
   - Instalação rápida

   > **Note**
   >
   > Para acesso à CLI, você precisa de um ID de chave de acesso e de uma chave de acesso secreta. Use credenciais temporárias em vez de chaves de acesso de longo prazo quando possível. As credenciais temporárias incluem um ID de acesso, uma chave de acesso secreta e um token de segurança que indica quando as credenciais expiram. Para aumentar a segurança de sua Conta da AWS, recomendamos não usar as credenciais de acesso associadas ao usuário raiz da sua Conta da AWS. Para obter mais informações, consulte Acesso programático na Referência geral da AWS e nas Práticas recomendadas de segurança no IAM no Guia do usuário do IAM.

2. Para confirmar se o perfil da AWS CLI está configurado corretamente, execute o comando a seguir em uma janela de comando.

   ```
   aws configure
   ```

Se o seu perfil foi configurado corretamente, você deve ver uma saída semelhante à seguinte.

```
AWS Access Key ID [****************52FQ]:
AWS Secret Access Key [****************xgyZ]:
Default region name [us-east-1]:
Default output format [json]:
```

3. Execute o comando a seguir para verificar se os comandos do Application Auto Scaling para a AWS CLI estão instalados.

```
aws application-autoscaling help
```

Usar o AWS CloudShell para trabalhar com o Application Auto Scaling na linha de comando

O AWS CloudShell permite pular a instalação da AWS CLI em seu ambiente de desenvolvimento e usar o AWS Management Console em seu lugar. Além de evitar a instalação, não é necessário configurar credenciais nem especificar uma região. Sua sessão do AWS Management Console fornece esse contexto para a AWS CLI. O AWS CloudShell pode ser usado em Regiões da AWS compatíveis.

Você pode executar comandos AWS CLI em serviços usando seu shell preferido (Bash, PowerShell ou Z shell).

Você pode iniciar o AWS CloudShell pelo AWS Management Console usando um dos seguintes dois métodos:

- Clique no ícone AWS CloudShell na barra de navegação do console. Ele está à direita da caixa de pesquisa.
- Use a caixa de pesquisa na barra de navegação do console para pesquisar por CloudShell e escolha a opção CloudShell.

Quando o AWS CloudShell for iniciado em uma nova janela do navegador pela primeira vez, um painel de boas-vindas vai exibir e listar os principais recursos. Depois de fechar esse painel, as atualizações de status serão fornecidas enquanto o shell configura e encaminha suas credenciais do console. Quando o prompt de comando for exibido, o shell estará pronto para interação.

Para obter mais informações sobre esse serviço, consulte o Manual do usuário do AWS CloudShell.

Criar recursos do Application Auto Scaling com 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 Manual do usuário da 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 IDs de recursos 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`). Consulte a tabela Referência do ARN da função vinculada ao serviçopara obter ARNs de função.

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 ComandoAWS CloudFormation

Escalabilidade programada

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.

Tópicos

Como a escalabilidade programada funciona

Este tópico descreve como o escalonamento programado funciona e apresenta as principais considerações que você precisa entender para usá-lo com eficiência.

Conteúdo

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 ter mais informações, consulte Programar ações de escalabilidade recorrentes usando expressões cron.
- 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 ter 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 ver um tutorial que descreve como criar ações programadas usando exemplos de recursos da AWS , consulte Tutorial: comece a usar a escalabilidade programada usando a AWS CLI.

Para obter mais informações sobre a criação de ações programadas para grupos do Auto Scaling, consulte Escalabilidade programada para o Amazon EC2 Auto Scaling no Guia do usuário do Amazon EC2 Auto 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 ter mais informações, consulte AWS serviços que você pode usar com o Application Auto Scaling.

Programar ações de escalabilidade recorrentes usando expressões cron

> **Important**
>
> Para obter ajuda com expressões cron para o Amazon EC2 Auto Scaling, consulte o tópico Cronogramas recorrentes no Guia do usuário do Amazon EC2 Auto Scaling. Com o Amazon EC2 Auto Scaling, você usa a sintaxe cron tradicional em vez da sintaxe cron personalizada usada pelo Application Auto Scaling.

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 os 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

Minutos	Horas	Dia do mês	Mês	Dia da semana	Ano	Significado
0/15	*	*	*	?	*	Executada a cada 15 minutos
0/10	*	?	*	SEG-SEX	*	Executada a cada 10 minutos de 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.

Exemplo de ações programadas para o Application Auto Scaling

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

Para obter brevidade, os exemplos deste tópico ilustram comandos da CLI de alguns dos serviços que se integram ao Application Auto Scaling. 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 AWS serviços que você pode usar com o Application Auto Scaling.

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.

Conteúdo

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
```

> **Note**
>
> 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 do usuário do Amazon CloudWatch Events.

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 ter mais informações, consulte Programar ações de escalabilidade recorrentes usando expressões cron.

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 ter mais informações, consulte Programar ações de escalabilidade recorrentes usando expressões cron.

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
```

Gerenciar escalabilidade programada para o Application Auto Scaling

AWS CLI Isso inclui vários outros comandos que ajudam você a gerenciar suas ações agendadas.

Para obter brevidade, os exemplos deste tópico ilustram comandos da CLI de alguns dos serviços que se integram ao Application Auto Scaling. 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 AWS serviços que você pode usar com o Application Auto Scaling.

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.

Conteúdo

Visualizar atividades de escalabilidade para um serviço especificado

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
```

Se o comando for bem-sucedido, você verá um resultado semelhante a este.


```
{
    "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 todas as ações programadas para um serviço especificado

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
```

Se houver êxito, o comando gerará uma saída semelhante à seguinte.

```
{
```

```
    "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
```

A seguir, um exemplo de saída.

```
{
    "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
        }
    ]
}
```

Se houver mais de um valor fornecido para a opção `--scheduled-action-names`, todas as ações programadas cujos nomes são uma correspondência serão incluídas no resultado.

Desativar a escalabilidade programada para um destino escalável

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

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}"
```

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

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

Para retomar a escalabilidade programada, execute o comando novamente, especificando `false` como o valor do atributo `ScheduledScalingSuspended`.

Excluir uma ação programada

Ao concluir uma ação agendada, você pode excluí-la usando o delete-scheduled-actioncomando.

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
```

Se houver êxito, o comando retornará à solicitação.

Tutorial: comece a usar a escalabilidade programada usando a AWS CLI

O tutorial a seguir mostra como usar o para começar com AWS CLI a escalabilidade programada, ajudando você a criar ações programadas que escalam uma tabela de amostra do DynamoDB chamada. `TestTable` Se ainda não tiver uma tabela de `TestTable` do DynamoDB para testes,

você poderá criar uma agora ao executar o comando create-table mostrado em Etapa 1: criar uma tabela do DynamoDB no Guia do desenvolvedor do Amazon DynamoDB.

Ao usar o AWS CLI, lembre-se de que seus comandos são executados na AWS região configurada 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.

> **Note**
>
> Você pode incorrer em AWS cobranças como parte deste tutorial. Monitore seu uso do Nível gratuito e certifique-se de entender os custos associados ao número de unidades de capacidade de leitura e gravação que os bancos de dados do DynamoDB usam.

Conteúdo

Etapa 1: inscrever o destino escalável

Comece registrando a tabela do DynamoDB como um destino escalável com o Application Auto Scaling.

Para inscrever um destino escalável com o Application Auto Scaling

1. Primeiro, use o describe-scalable-targetscomando para verificar se algum recurso do DynamoDB já está registrado. Isso permite que você verifique se a tabela `TestTable` não está registrada, caso não seja uma nova tabela.

   Linux, macOS ou Unix

   ```
   aws application-autoscaling describe-scalable-targets \
     --service-namespace dynamodb
   ```

   Windows

```
aws application-autoscaling describe-scalable-targets --service-namespace dynamodb
```

Se não houver destinos dimensionáveis existentes, esta será a resposta.

```
{
    "ScalableTargets": []
}
```

2. Use o register-scalable-targetcomando a seguir para registrar a capacidade de gravação de sua tabela do DynamoDB chamada. `TestTable` Defina um mínimo de capacidade desejada de 5 unidades de capacidade de gravação e um máximo de capacidade desejada de 10 unidades de capacidade de gravação.

   Linux, macOS ou Unix

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

   Windows

```
aws application-autoscaling register-scalable-target --service-namespace dynamodb
 --scalable-dimension dynamodb:table:WriteCapacityUnits --resource-id table/
TestTable --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"
}
```

Etapa 2: criar duas ações programadas

O Application Auto Scaling permite programar a hora em que uma ação de escalabilidade deve ocorrer. Você especifica o destino dimensionável, a programação e a capacidade mínima e máxima.

Na hora especificada, o Application Auto Scaling atualiza os valores mínimo e máximo para o destino escalável. Se a capacidade atual estiver fora desse intervalo, isso resultará em uma ação de dimensionamento.

Programar atualizações para a capacidade mínima e máxima também será útil se você decidir criar uma política de dimensionamento. Uma política de dimensionamento permite que seus recursos sejam dimensionados dinamicamente com base na utilização atual de recursos. Uma proteção comum para uma política de dimensionamento é ter valores apropriados para capacidade máxima e mínima.

Para este exercício, criamos duas ações únicas para expandir e reduzir.

Para criar e visualizar as ações programadas

1. Para criar a primeira ação agendada, use o put-scheduled-actioncomando a seguir.

   O comando at (às) em `--schedule` programa a ação para ser executada uma vez em uma data e hora especificadas no futuro. Os horários estão no formato de 24 horas em UTC. Programe a ação para ocorrer cerca de 5 minutos a partir de agora.

   Na data e hora especificadas, o Application Auto Scaling atualiza os valores `MinCapacity` e `MaxCapacity`. Supondo que a tabela tenha atualmente cinco unidades de capacidade de gravação, o Application Auto Scaling aumenta a escala na horizontal para `MinCapacity`, para colocar a tabela dentro do novo intervalo desejado de 15 a 20 unidades de capacidade de gravação.

   Linux, macOS ou Unix

   ```
   aws application-autoscaling put-scheduled-action \
     --service-namespace dynamodb \
     --scalable-dimension dynamodb:table:WriteCapacityUnits \
     --resource-id table/TestTable \
     --scheduled-action-name my-first-scheduled-action \
     --schedule "at(2019-05-20T17:05:00)" \
     --scalable-target-action MinCapacity=15,MaxCapacity=20
   ```

   Windows

   ```
   aws application-autoscaling put-scheduled-action --service-namespace dynamodb
    --scalable-dimension dynamodb:table:WriteCapacityUnits --resource-id table/
   ```

```
TestTable --scheduled-action-name my-first-scheduled-action --schedule
 "at(2019-05-20T17:05:00)" --scalable-target-action MinCapacity=15,MaxCapacity=20
```

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

2. Para criar a segunda ação agendada que o Application Auto Scaling usa para escalar, use o comando a seguir put-scheduled-action.

   Programe a ação para ocorrer cerca de 10 minutos a partir de agora.

   Na data e hora especificadas, o Application Auto Scaling atualiza a `MinCapacity` e a `MaxCapacity` da tabela e reduz a escala na horizontal para `MaxCapacity`, para recolocar a tabela no intervalo original desejado de cinco a dez unidades de capacidade de gravação.

   Linux, macOS ou Unix

   ```
   aws application-autoscaling put-scheduled-action \
     --service-namespace dynamodb \
     --scalable-dimension dynamodb:table:WriteCapacityUnits \
     --resource-id table/TestTable \
     --scheduled-action-name my-second-scheduled-action \
     --schedule "at(2019-05-20T17:10:00)" \
     --scalable-target-action MinCapacity=5,MaxCapacity=10
   ```

   Windows

   ```
   aws application-autoscaling put-scheduled-action --service-namespace dynamodb
    --scalable-dimension dynamodb:table:WriteCapacityUnits --resource-id table/
   TestTable --scheduled-action-name my-second-scheduled-action --schedule
    "at(2019-05-20T17:10:00)" --scalable-target-action MinCapacity=5,MaxCapacity=10
   ```

3. (Opcional) Obtenha uma lista de ações agendadas para o namespace de serviço especificado usando o comando a seguir describe-scheduled-actions.

   Linux, macOS ou Unix

   ```
   aws application-autoscaling describe-scheduled-actions \
     --service-namespace dynamodb
   ```

   Windows

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

A seguir, um exemplo de saída.

```
{
    "ScheduledActions": [
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Schedule": "at(2019-05-20T18:35:00)",
            "ResourceId": "table/TestTable",
            "CreationTime": 1561571888.361,
            "ScheduledActionARN": "arn:aws:autoscaling:us-
east-1:123456789012:scheduledAction:2d36aa3b-cdf9-4565-b290-81db519b227d:resource/
dynamodb/table/TestTable: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/TestTable",
            "CreationTime": 1561571946.021,
            "ScheduledActionARN": "arn:aws:autoscaling:us-
east-1:123456789012:scheduledAction:2d36aa3b-cdf9-4565-b290-81db519b227d:resource/
dynamodb/table/TestTable:scheduledActionName/my-second-scheduled-action",
            "ScalableTargetAction": {
                "MinCapacity": 5,
                "MaxCapacity": 10
            },
            "ScheduledActionName": "my-second-scheduled-action",
            "ServiceNamespace": "dynamodb"
        }
    ]
}
```

Etapa 3: visualizar as atividades de escalabilidade

Nesta etapa, você visualiza as atividades de escalabilidade acionadas pelas ações programadas e verifica se o DynamoDB alterou a capacidade de gravação da tabela.

Para visualizar as atividades de dimensionamento

1. Aguarde o horário escolhido e verifique se as ações agendadas estão funcionando usando o describe-scaling-activitiescomando a seguir.

   Linux, macOS ou Unix

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

   Windows

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

   Veja a seguir um exemplo de saída da primeira ação programada enquanto ela está em andamento.

   As atividades de dimensionamento são ordenadas por data de criação, com as atividades de dimensionamento mais recentes retornadas primeiro.

   ```
   {
       "ScalingActivities": [
           {
               "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
               "Description": "Setting write capacity units to 15.",
               "ResourceId": "table/TestTable",
               "ActivityId": "d8ea4de6-9eaa-499f-b466-2cc5e681ba8b",
               "StartTime": 1561574108.904,
               "ServiceNamespace": "dynamodb",
               "Cause": "minimum capacity was set to 15",
               "StatusMessage": "Successfully set write capacity units to 15. Waiting
    for change to be fulfilled by dynamodb.",
               "StatusCode": "InProgress"
           },
           {
   ```

```
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Description": "Setting min capacity to 15 and max capacity to 20",
            "ResourceId": "table/TestTable",
            "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"
        }
    ]
}
```

Veja a seguir um exemplo de saída depois que ambas as ações programadas foram executadas.

```
{
    "ScalingActivities": [
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Description": "Setting write capacity units to 10.",
            "ResourceId": "table/TestTable",
            "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/TestTable",
            "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/TestTable",
            "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/TestTable",
            "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"
        }
    ]
}
```

2. Depois de executar as ações programadas com êxito, acesse o console do DynamoDB e escolha a tabela com a qual você deseja trabalhar. Visualize as Write capacity units (Unidades de capacidade de gravação) na guia Capacity (Capacidade). Depois que a segunda ação de dimensionamento foi executada, as unidades de capacidade de gravação devem ter sido dimensionadas de 15 para 10.

   Você também pode verificar a capacidade de gravação atual da tabela usando comando describe-table a seguir. Para filtrar a saída, inclua a opção `--query`. Para obter mais

informações sobre os recursos de filtragem de saída do AWS CLI, consulte Controlando a saída do comando AWS CLI no Guia do AWS Command Line Interface Usuário.

Linux, macOS ou Unix

```
aws dynamodb describe-table --table-name TestTable \
  --query 'Table.[TableName,TableStatus,ProvisionedThroughput]'
```

Windows

```
aws dynamodb describe-table --table-name TestTable --query "Table.
[TableName,TableStatus,ProvisionedThroughput]"
```

A seguir, um exemplo de saída.

```
[
    "TestTable",
    "ACTIVE",
    {
        "NumberOfDecreasesToday": 1,
        "WriteCapacityUnits": 10,
        "LastIncreaseDateTime": 1561574133.264,
        "ReadCapacityUnits": 5,
        "LastDecreaseDateTime": 1561574435.607
    }
]
```

Etapa 4: próximas etapas

Se você quiser tentar dimensionar com escalabilidade programada e uma política de escalabilidade, siga as etapas em Tutorial: configurar o ajuste de escala automático para processar uma workload pesada.

Etapa 5: limpar

Quando terminar de trabalhar com os exercícios de conceitos básicos, você poderá limpar os recursos associados da maneira indicada a seguir.

Como excluir as ações programadas

O delete-scheduled-actioncomando a seguir exclui uma ação agendada especificada. Você poderá ignorar esta etapa se desejar manter a ação programada para uso futuro.

Linux, macOS ou Unix

```
aws application-autoscaling delete-scheduled-action \
  --service-namespace dynamodb \
  --scalable-dimension dynamodb:table:WriteCapacityUnits \
  --resource-id table/TestTable \
  --scheduled-action-name my-second-scheduled-action
```

Windows

```
aws application-autoscaling delete-scheduled-action --service-namespace dynamodb --
scalable-dimension dynamodb:table:WriteCapacityUnits --resource-id table/TestTable --
scheduled-action-name my-second-scheduled-action
```

Como cancelar o registro do destino dimensionável:

Use o deregister-scalable-targetcomando a seguir para cancelar o registro do alvo escalável. 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 dynamodb \
  --scalable-dimension dynamodb:table:WriteCapacityUnits \
  --resource-id table/TestTable
```

Windows

```
aws application-autoscaling deregister-scalable-target --service-namespace dynamodb --
scalable-dimension dynamodb:table:WriteCapacityUnits --resource-id table/TestTable
```

Para excluir uma tabela do DynamoDB

Use o comando delete-table a seguir para excluir a tabela usada neste tutorial. É possível ignorar esta etapa se quiser manter a tabela para uso futuro.

Linux, macOS ou Unix

```
aws dynamodb delete-table --table-name TestTable
```

Windows

```
aws dynamodb delete-table --table-name TestTable
```

Políticas de escalabilidade de rastreamento de destino

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.

Tópicos

Como funciona o escalonamento de rastreamento de metas

Este tópico descreve como o escalonamento de rastreamento de metas funciona e apresenta os principais elementos de uma política de escalabilidade de rastreamento de metas.

Conteúdo

Como funcionam

Para usar a escala de rastreamento de metas, você cria uma política de escalabilidade de rastreamento de metas e especifica o seguinte:

- Métrica — uma CloudWatch métrica a ser monitorada, como a utilização média da CPU ou a contagem média de solicitações por alvo.

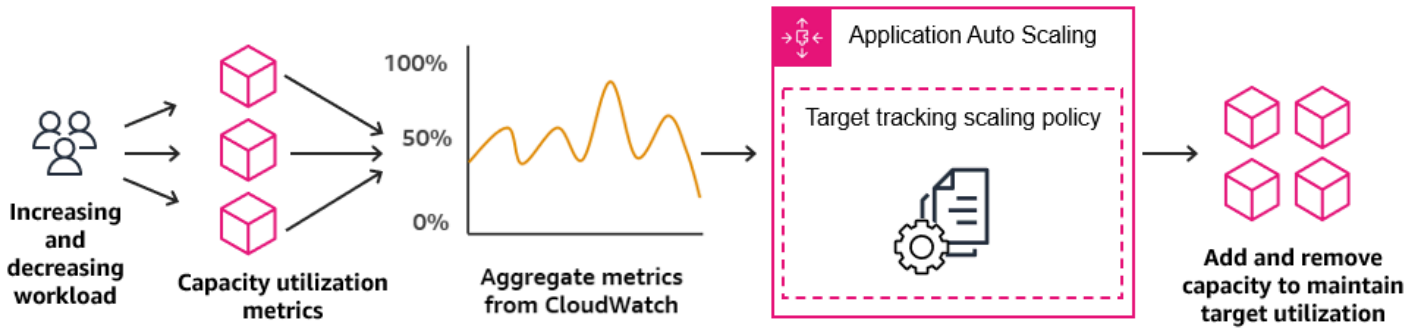
- Valor de destino: o valor de destino da métrica, como 50% de utilização da CPU ou mil solicitações por destino por minuto.

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 ter 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 métricas do Amazon EC2 são enviadas em intervalos de cinco minutos, por padrão, mas podem ser configuradas para um minuto (o que é 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 grupos de ElastiCache replicação 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
- SageMaker componentes de inferência
- SageMaker Concorrência provisionada sem servidor
- Spot Fleets
- 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 sobre como criar alarmes quando não houver dados suficientes, consulte Monitorar com alarmes do CloudWatch.
- 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 mais informações sobre a criação de políticas de escalabilidade de monitoramento do objetivo para grupos do Auto Scaling, consulte Políticas de escalabilidade de monitoramento do objetivo para o Amazon EC2 Auto Scaling no Guia do usuário do Amazon EC2 Auto 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. Para obter informações completas sobre o uso de planos de escalabilidade, consulte a documentação da AWS Auto Scaling.
- 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 ter mais informações, consulte AWS serviços que você pode usar com o Application Auto Scaling.

Crie uma política de escalabilidade de rastreamento de metas usando o AWS CLI

Você pode criar uma política de escalabilidade de rastreamento de destino para o Application Auto Scaling usando AWS CLI o para as seguintes tarefas de configuração.

1. Registrar um destino escalável.
2. Adicione uma política de escalabilidade de monitoramento do objetivo ao destino escalável.

Em resumo, os exemplos deste tópico ilustram comandos da CLI para uma frota spot do Amazon EC2. 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 AWS serviços que você pode usar com o Application Auto Scaling.

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.

Conteúdo

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 *espaço reservado para entrada do usuário* 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
```

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

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

Criar uma política de dimensionamento com monitoramento do objetivo

Para criar uma política de escalabilidade de rastreamento de metas, você pode usar os exemplos a seguir para ajudá-lo a começar.

Para criar uma política de escalabilidade com monitoramento do objetivo

1. Use o `cat` comando a seguir para armazenar um valor alvo para sua política de escalabilidade e uma especificação métrica predefinida em um arquivo JSON nomeado `config.json` em seu diretório inicial. Veja a seguir um exemplo de configuração de rastreamento de metas 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.

Como alternativa, você pode usar uma métrica personalizada para escalar criando uma especificação métrica personalizada e adicionando valores para cada parâmetro de CloudWatch. Veja a seguir um exemplo de configuração de rastreamento de metas 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 put-scaling-policycomando a seguir, junto com o `config.json` arquivo que você criou, para criar uma política de escalabilidade 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
```

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

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

Descrever as política de dimensionamento com monitoramento do objetivo

Você pode descrever todas as políticas de escalabilidade para o namespace de serviço especificado usando o comando a seguir. 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`]"
```

A seguir, 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
    }
]
```

Excluir uma política de dimensionamento com monitoramento do objetivo

Ao concluir uma política de escalabilidade de rastreamento de metas, você pode excluí-la usando o delete-scaling-policycomando.

O comando a seguir exclui a política de dimensionamento de rastreamento de destino que você especificou para a solicitação especificada da frota spot. 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 CloudWatch console 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

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)
```

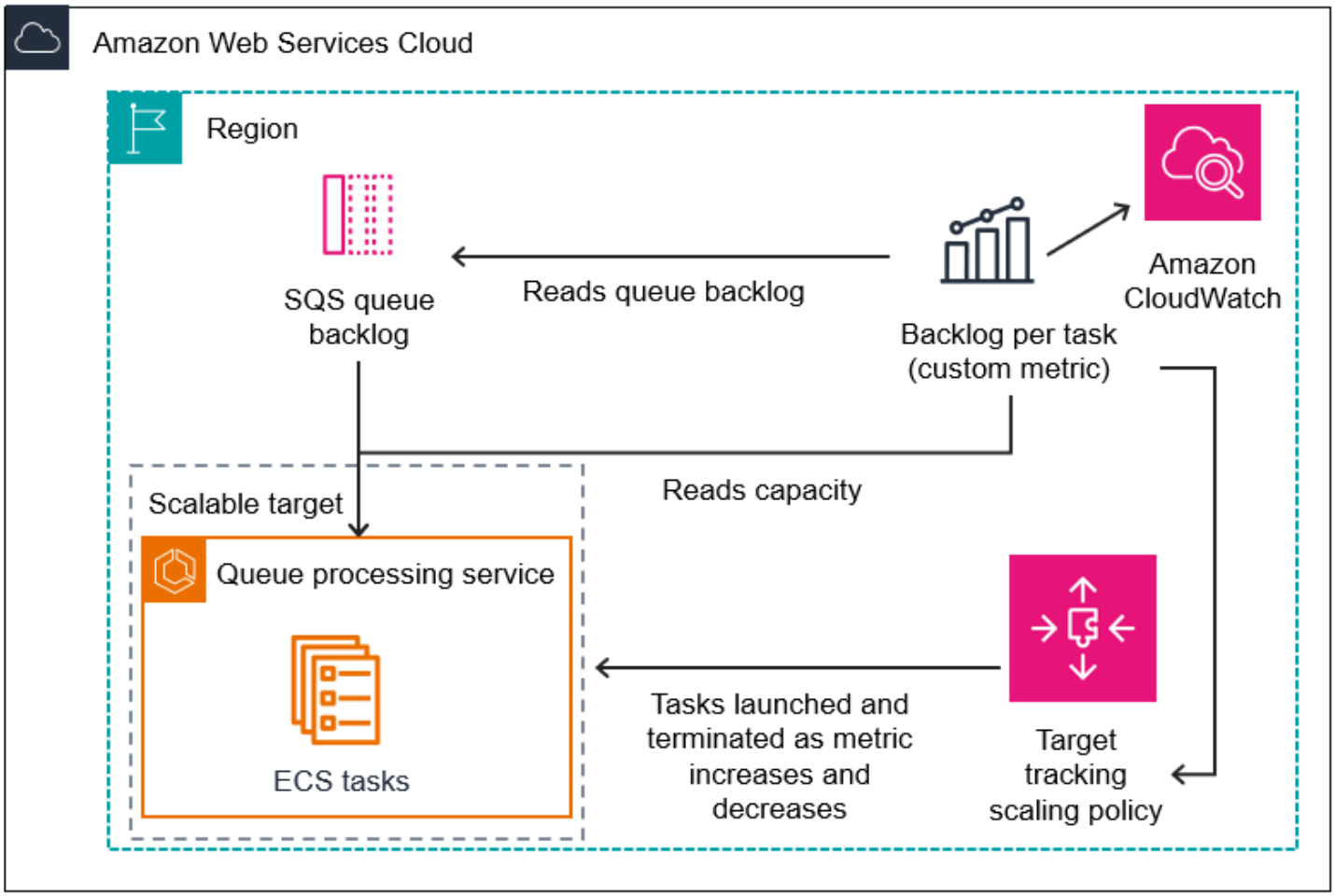
Então, suas informações CloudWatch métricas são as seguintes.

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 diagrama a seguir 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 *`espaço reservado para entrada do usuário`* 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 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.

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 os 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

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.

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
- SageMaker componentes de inferência
- SageMaker Concorrência provisionada sem servidor
- Spot Fleets
- Recursos personalizados

Tópicos

Como funciona o escalonamento por etapas

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

Conteúdo

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.

> **Note**
>
> 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.

Desaquecimento

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

Limitações

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

Criar uma política de escalabilidade em etapas usando a AWS CLI

Você pode criar uma política de escalabilidade de etapas para o Application Auto Scaling usando AWS CLI o para as seguintes tarefas de configuração.

1. Registrar um destino escalável.
2. Adicione uma política de escalabilidade em etapas ao destino escalável.
3. Crie um CloudWatch alarme para a política.

Em suma, os exemplos deste tópico ilustram comandos da CLI para um serviço do Amazon ECS. 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 AWS serviços que você pode usar com o Application Auto Scaling.

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.

Conteúdo

- Registrar um destino escalável
- Criar uma política de escalabilidade em etapas
- Criação de um alarme que invoca a política de escalabilidade
- Descrever políticas de escalabilidade em etapas
- Excluir política de escalabilidade em etapas

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 *espaço reservado para entrada do usuário* 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
```

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

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

Criar uma política de escalabilidade em etapas

Para criar uma política de escalabilidade por etapas para sua meta escalável, você pode usar os exemplos a seguir para ajudá-lo a começar.

Scale out

Para criar uma política de escalabilidade por etapas para expansão horizontal (aumentar a capacidade)

1. Use o `cat` comando a seguir para armazenar uma configuração de política de escalabilidade de etapas em um arquivo JSON nomeado `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):

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

   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.

   ```
   {
       "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

Para criar uma política de escalabilidade por etapas para escalar (diminuir a capacidade)

1. Use o `cat` comando a seguir para armazenar uma configuração de política de escalabilidade de etapas em um arquivo JSON nomeado `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):

- Diminua a capacidade em 1 quando o valor da métrica for menor ou igual a 50, mas maior que 40
- Diminua a capacidade em 2 quando o valor da métrica for menor ou igual a 40, mas maior que 30
- Diminua a capacidade em 3 quando o valor da métrica for 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
```

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.

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

Criação de um alarme que invoca a política de escalabilidade

Por fim, use o CloudWatch [put-metric-alarm](#)comando 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
```

Descrever políticas de escalabilidade em etapas

Você pode descrever todas as políticas de escalabilidade para o namespace de serviço especificado usando o comando a seguir. describe-scaling-policies

```
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`]"
```

A seguir, 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
    }
]
```

Excluir política de escalabilidade em etapas

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, conclua as tarefas a seguir.

Para excluir a política de dimensionamento

Use o seguinte 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
```

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

> **Important**
>
> Antes de explorar este tutorial, recomendamos que você primeiramente examine o tutorial introdutório: Tutorial: comece a usar a escalabilidade programada usando a AWS CLI.

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.

Índice

Pré-requisitos

Este tutorial pressupõe que você já tenha feito o seguinte:

- Você criou uma conta da AWS. Para obter mais informações, consulte Configurar o uso do Application Auto Scaling.
- Você instalou e configurou a AWS CLI. Para obter mais informações, consulte Configurar a AWS CLI.
- Sua conta tem todas as permissões necessárias para inscrever e cancelar o registro de recursos como destinos escaláveis com o Application Auto Scaling. Também tem todas 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.
- Você tem um recurso compatível em um ambiente de não 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 AWS serviços 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 ajudar a resolver o problema, uma mensagem na saída indicará que a capacidade mínima não pode ser menor do que o valor especificado e fornecerá o valor mínimo de capacidade que o recurso da AWS 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 comando register-scalable-target para inscrever um novo destino 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 AWS serviços 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 comando put-scheduled-action para criar ações programadas que são configuradas para atender às suas necessidades empresariais. 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 aumentar a escala na horizontal do destino escalável, use o comando put-scheduled-action. 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 se a ação programada existe, use o comando describe-scheduled-actions.

   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`]"
```

A seguir está 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 `MinCapacity` e `MaxCapacity` do destino como 0, conforme instruído pelo comando 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 se a ação programada existe, use o comando describe-scheduled-actions.

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`]"
```

A seguir está 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 put-scaling-policy para criar a política.

   As métricas usadas com mais frequência para o monitoramento do objetivo 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 escalabilidade de rastreamento de destino.

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 AWS serviços 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 tiver êxito, esse comando retornará os ARNs e os nomes dos dois alarmes do CloudWatch criados em seu nome.

2. Para confirmar se a ação programada existe, use o comando describe-scheduled-actions.

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`]"
```

A seguir está 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 describe-scaling-activities.

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 de uma configuração de escalabilidade não exclui seus recursos da AWS subjacente. 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 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 comando delete-scaling-policy exclui uma política de dimensionamento com monitoramento do objetivo 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 comando deregister-scalable-target para cancelar a inscrição de um destino escalável. 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

> **Note**
>
> Para obter instruções sobre como suspender os processos de escalabilidade enquanto as implantações do Amazon ECS estão em andamento, consulte a seguinte documentação: Escalabilidade automática de serviços e implantações 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 as atividades de escalonamento

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 AWS serviços 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
```

A seguir, 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 escalação por alvo 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 não escadas

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 não escalonada será registrada no histórico de atividades de escalonamento. Quaisquer atividades subsequentes não escalonadas não gerarão novas entradas, a menos que o motivo da não escalabilidade mude.

Entender os códigos dos motivos de não escalação

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-alvo colocou temporari amente o recurso em um estado não escalável. O Application Auto Scaling tentará novamente se as condições de escalação automática configuradas na política de escalação 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.			

Monitoramento do Application Auto Scaling

Monitorar é uma parte importante da manutenção da confiabilidade, da disponibilidade e da performance do Application Auto Scaling, além das outras soluções da AWS. É necessário coletar dados de monitoramento de todas as partes de sua solução da AWS para que seja possível depurar mais facilmente uma falha de vários pontos caso ocorra. A AWS fornece ferramentas de monitoramento para observar o Application Auto Scaling, relatar quando algo está errado e executar ações automáticas quando apropriado.

Você pode usar os seguintes recursos para ajudar a gerenciar seus recursos da AWS:

AWS CloudTrail

Com o AWS CloudTrail, você pode rastrear as chamadas feitas para a API do Application Auto Scaling por ou em nome de sua Conta da AWS. O 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 Log de chamadas de API do Application Auto Scaling com o AWS CloudTrail.

> **Note**
>
> Para obter informações sobre outros serviços da AWS que podem ajudar você a registrar em log e coletar dados sobre suas workloads, consulte o Guia de registro em log e monitoramento para proprietários de aplicações na Orientação prescritiva da AWS.

Amazon CloudWatch

O Amazon CloudWatch ajuda você a analisar logs, além de monitorar em tempo real as métricas dos seus recursos e aplicações hospedadas na AWS. É possível 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 instruir o CloudWatch a rastrear a utilização de recursos e notificar você quando a utilização for muito alta ou quando o alarme da métrica entrar no estado `INSUFFICIENT_DATA`. Para obter mais informações, consulte Monitorar seus recursos usando o CloudWatch.

O CloudWatch também rastreia métricas de uso da API da AWS para o 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 Métricas de uso da AWS no Guia do usuário do Amazon CloudWatch.

Amazon EventBridge

O Amazon EventBridge é um serviço de barramento de eventos sem servidor que facilita a conexão de aplicações a dados de diversas origens. O EventBridge fornece um fluxo de dados em tempo real de suas próprias aplicações, de aplicações de software como serviço (SaaS) e de serviços da AWS e roteia 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 Monitorar eventos do Application Auto Scaling com o Amazon EventBridge.

AWS Health Dashboard

O AWS Health Dashboard (PHD) exibe informações e também fornece notificações que são invocadas por alterações na integridade dos recursos da AWS. 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 obter mais informações, consulte Notificações de AWS Health Dashboard do Application Auto Scaling.

Log de chamadas de API do Application Auto Scaling com o AWS CloudTrail

O Application Auto Scaling é integrado ao AWS CloudTrail, um serviço que fornece um registro das ações executadas por um usuário, função ou serviço da AWS usando a API do Application Auto Scaling. O CloudTrail captura todas as chamadas de API para o Application Auto Scaling na forma de eventos. As chamadas capturadas incluem as chamadas do AWS Management Console e as chamadas de código à API do Application Auto Scaling. Se criar uma trilha, você poderá habilitar a entrega contínua de eventos do CloudTrail para um bucket do Amazon S3, incluindo eventos do Application Auto Scaling. Se você não configurar uma trilha, ainda poderá visualizar os eventos mais recentes no console do CloudTrail em Event history (Histórico de eventos). Usando as informações coletadas pelo CloudTrail, você pode determinar a solicitação que foi feita para o Application Auto Scaling, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para saber mais sobre o CloudTrail, consulte o Guia do usuário do AWS CloudTrail.

Informações do Application Auto Scaling no CloudTrail

O CloudTrail é habilitado em sua Conta da AWS quando ela é criada. Quando há uma atividade no Application Auto Scaling, ela é registrada em um evento do CloudTrail junto com outros eventos de serviços da AWS em Histórico de eventos. Você pode visualizar, pesquisar e baixar eventos recentes em sua Conta da AWS. Para obter mais informações, consulte Visualizar eventos com o histórico de eventos do CloudTrail.

Para obter um registro contínuo de eventos em sua Conta da AWS, incluindo eventos para o Application Auto Scaling, crie uma trilha. Uma trilha permite que o CloudTrail entregue arquivos de log a um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas as Regiões da AWS. A trilha registra em log eventos de todas as regiões na partição da AWS e entrega os arquivos de log para o bucket do Amazon S3 especificado por você. Além disso, você pode configurar outros serviços da Amazon Web Services para analisar mais profundamente e agir sobre os dados de eventos coletados nos logs do CloudTrail. Para obter mais informações, consulte as informações a seguir.

- Visão geral da criação de uma trilha
- Serviços e integrações compatíveis com o CloudTrail
- Configurar notificações do Amazon SNS para o CloudTrail
- Receber arquivos de log do CloudTrail de várias regiões e Receber arquivos de log do CloudTrail de várias contas

Todas as ações do Application Auto Scaling são registradas em log pelo CloudTrail e documentadas na Referência da API do Application Auto Scaling. Por exemplo, as chamadas para as APIs `PutScalingPolicy`, `DeleteScalingPolicy` e `DescribeScalingPolicies` geram entradas nos arquivos de log do CloudTrail.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar:

- Se a solicitação foi feita com credenciais de usuário raiz ou do AWS Identity and Access Management (IAM).
- Se a solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado.
- Se a solicitação foi feita por outro serviço da AWS.

Para obter mais informações, consulte Elemento userIdentity do CloudTrail.

Noções básicas sobre entradas do arquivo de log do Application Auto Scaling

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log a um bucket do Amazon S3 especificado. Os arquivos de log do CloudTrail contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros de solicitação e assim por diante. Os arquivos de log do CloudTrail não são um rastreamento de pilha ordenada de chamadas de API pública. Dessa forma, eles não são exibidos em uma ordem específica.

O exemplo a seguir mostra uma entrada de log do CloudTrail que demonstra a ação `DescribeScalableTargets`.

```
{
    "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"
}
```

Recursos relacionados

Com o CloudWatch Logs, você pode monitorar e receber alertas para eventos específicos capturados pelo CloudTrail. Esses eventos enviados para o CloudWatch Logs são aqueles configurados para serem registrados em log por sua trilha, portanto, certifique-se de ter configurado a trilha ou as trilhas para registrar em log os tipos de evento que deseja monitorar. O CloudWatch Logs pode monitorar informações nos arquivos de log e notificar você quando determinados limites forem atingidos. Você também pode arquivar seus dados de log em armazenamento resiliente. Para obter mais informações, consulte o Guia do usuário do Amazon CloudWatch Logs e o tópico Monitorar arquivos de log do CloudTrail com o Amazon CloudWatch Logs no Guia do usuário do AWS CloudTrail.

Monitorar seus recursos usando o CloudWatch

Esta seção fornece informações sobre métricas de monitoramento para seus recursos escaláveis usando o CloudWatch.

Tópicos

- Criar painéis com o CloudWatch
- Monitorar com alarmes do CloudWatch
- Monitorar o uso de recursos com o CloudWatch

Criar painéis com o CloudWatch

Você pode monitorar como a aplicação usa recursos com o Amazon CloudWatch, que gera métricas sobre uso e performance. O CloudWatch coleta dados brutos de seus recursos da AWS e das aplicações executadas na AWS e os processa em métricas legíveis quase em tempo real. As

métricas são mantidas por 15 meses de maneira que você possa acessar informações históricas para ter uma perspectiva melhor sobre a performance do aplicativo. Para obter mais informações, consulte o Guia do usuário do Amazon CloudWatch.

Os painéis do CloudWatch são páginas iniciais personalizáveis no console do CloudWatch que você pode usar para monitorar seus recursos em uma única visualização, mesmo os recursos distribuídos em regiões diferentes. Você pode usar os painéis do CloudWatch para criar visualizações personalizadas de métricas selecionadas para seus recursos da AWS. Você pode selecionar a cor usada para cada métrica em cada gráfico, para que possa monitorar com mais facilidade a mesma métrica em vários gráficos.

Criar um painel do CloudWatch

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboard (Painel) e selecione Create new dashboard (Criar novo painel).
3. Insira um nome para o painel, como o nome do serviço para o qual você deseja visualizar os dados do CloudWatch.
4. Escolha Create dashboard (Criar painel).
5. Escolha um tipo de widget para adicionar ao seu painel, como um gráfico de linhas. Depois, escolha Configure (Configurar) e a métrica que deseja adicionar ao painel. Para obter mais informações, consulte Adicionar ou remover um gráfico de um painel do CloudWatch no Manual do usuário do Amazon CloudWatch

Por padrão, as métricas criadas nos painéis do CloudWatch são médias. Embora o CloudWatch permita escolher qualquer estatística para cada métrica, nem todas as combinações são úteis. Por exemplo, as estatísticas para utilização da CPU média, mínima e máxima são úteis, mas a estatística de soma não é.

Uma medida de performance do aplicativo normalmente usada é a utilização média da CPU. Se houver um aumento na utilização da CPU e sua capacidade for insuficiente para lidar com isso, o aplicativo poderá parar de responder. Por outro lado, se você tiver muita capacidade e os recursos estiverem sendo executados quando a utilização for baixa, haverá aumento dos custos de uso desse serviço.

Dependendo do serviço, você também tem métricas que rastreiam a quantidade de throughput provisionada disponível. Por exemplo, para o número de invocações que estão sendo processadas

em um alias de função ou versão com simultaneidade provisionada, o Lambda emite a métrica `ProvisionedConcurrencyUtilization`. Se você estiver iniciando um trabalho grande e invocar a mesma função várias vezes simultaneamente, o trabalho poderá enfrentar latência quando exceder a quantidade de simultaneidade provisionada disponível. Por outro lado, se você tiver mais simultaneidade provisionada do que o necessário, seus custos poderão ser maiores do que deveriam.

As métricas não serão exibidas até que o recurso tenha sido totalmente configurado. Além disso, se não houve uma métrica publicada nos últimos 14 dias, você não poderá encontrá-la quando estiver procurando métricas para adicionar a um gráfico em um painel do CloudWatch. Para obter informações sobre como adicionar uma métrica manualmente, consulte Criar gráficos de métricas manualmente em um painel do CloudWatch no Manual do usuário do Amazon CloudWatch.

Para obter mais informações, consulte a documentação do serviço que está disponível na tabela em Monitorar o uso de recursos com o CloudWatch.

Monitorar com alarmes do CloudWatch

Você pode criar alarmes para notificar quando o Amazon CloudWatch detectar problemas que possam exigir sua atenção.

Um alarme do CloudWatch observa uma única métrica. Eles invocam uma ou mais ações somente quando o estado mudar e persistir pelo período especificado por você. Por exemplo, defina um alarme que notifique quando o valor da métrica fica abaixo ou excede um determinado nível, garantindo que você seja notificado antes que ocorra um problema em potencial.

O CloudWatch também permite que você defina um alarme que notifique quando a métrica estiver no estado `INSUFFICIENT_DATA`. Qualquer métrica, para qualquer serviço da AWS, pode ativar o alarme `INSUFFICIENT_DATA`. Esse é o estado inicial de um novo alarme, mas o estado do alarme também mudará para `INSUFFICIENT_DATA` se as métricas do CloudWatch se tornarem indisponíveis ou se não houver dados suficientes para a métrica determinar o estado do alarme. Por exemplo, AWS Lambda emite a métrica `ProvisionedConcurrencyUtilization` para o CloudWatch a cada minuto somente quando a função do Lambda está ativa. Se a função estiver inativa, isso fará com que o alarme vá para `INSUFFICIENT_DATA` enquanto aguarda as métricas. Isso é normal e pode não significar necessariamente que há um problema, mas pode ser indicativo de problema se você esperava atividade dentro de um período de tempo, mas não houve nenhuma.

Este tópico explica como criar um alarme que envie uma notificação quando a métrica estiver dentro ou fora de um limite definido por você ou quando não houver dados suficientes. Para

obter informações detalhadas sobre configuração de alarmes, consulte Usar alarmes do Amazon CloudWatch no Manual do usuário do Amazon CloudWatch.

Criar um alarmeque envie um e-mail

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Alarms, Create Alarm.
3. Escolha Select Metric (Selecionar métrica).

   Você é direcionado para uma página na qual pode encontrar todas as métricas. Os tipos de métrica disponíveis dependem dos serviços e recursos que você usa. As métricas são agrupadas primeiro pelo namespace do serviço e, em seguida, por várias combinações de dimensão dentro de cada namespace.
4. Selecione um namespace de métrica (por exemplo, Lambda) e uma dimensão de métrica (por exemplo, By Function Name (Nome por função)).

   A guia All metrics (Todas as métricas) exibe todas as métricas da dimensão e do namespace selecionados.
5. Marque a caixa de seleção ao lado da métrica para a qual você deseja criar um alarme e escolha Select Metric (Selecionar métrica).
6. Siga as instruções a seguir para configurar o alarme e, em seguida, escolha Next (Avançar):

   - Em Metric (Métrica), selecione um período de agregação de`1 minute` ou `5 minutes`. Se você escolher um minuto como o período de agregação para uma métrica, haverá um ponto de dados a cada minuto. O período mais curto cria um alarme mais sensível.
   - Em Conditions (Condições), configure seu limite, por exemplo, o valor que a métrica deve exceder antes que uma notificação seja gerada.
   - Em Additional configuration (Configuração adicional), para Datapoints to alarm (Pontos de dados para alarme), insira o número de pontos de dados (períodos de avaliação) durante os quais o valor da métrica deve atender às condições de limite para acionar o alarme. Por exemplo, dois períodos consecutivos de 5 minutos precisariam de 10 minutos para acionar o alarme.
   - Em Missing Data Treatment (Tratamento de dados ausentes), mantenha o padrão e trate pontos de dados ausentes como ausentes.

     Algumas métricas são relatadas somente quando há atividade ocorrendo. Isso pode resultar em uma métrica pouco relatada. Se uma métrica tiver frequentemente pontos de dados

ausentes por projeto, o estado do alarme será `INSUFFICIENT_DATA` durante esses períodos. Para forçar o alarme a manter o estado `ALARM` ou `OK` anterior para evitar que os alertas soem, você pode optar por ignorar os dados ausentes.

7. Em Notification (Notificação), escolha um tópico do SNS para notificar quando o alarme estiver no estado `ALARM`, `OK` ou `INSUFFICIENT_DATA`. Para que o alarme envie várias notificações para o mesmo estado de alarme ou para diferentes estados de alarme, escolha Add notification (Adicionar notificação).
8. Quando terminar, escolha Next (Próximo).
9. Insira um nome e, opcionalmente, uma descrição para o alarme e escolha Next (Próximo).
10. Selecione Criar alarme.

Para verificar o estado dos alarmes

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Alarms (Alarmes) para ver uma lista de alarmes.
3. Para filtrar alarmes, use os filtros suspensos ao lado do campo de pesquisa e escolha a opção de filtro que deseja aplicar.
4. Para editar ou excluir um alarme, selecione o alarme e depois escolha Actions (Ações), Edit (Editar) ou Actions (Ações), Delete (Excluir).

Monitorar o uso de recursos com o CloudWatch

Com o Amazon CloudWatch, você tem visibilidade quase contínua das aplicações nos recursos escaláveis. O CloudWatch é um serviço de monitoramento para recursos da AWS. O CloudWatch pode ser usado para coletar e rastrear métricas, definir alarmes e reagir automaticamente a alterações nos recursos da AWS. 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 integrados ao Application Auto Scaling, eles enviam as métricas exibidas na tabela a seguir para o CloudWatch. No CloudWatch, as métricas são agrupadas primeiro pelo namespace do serviço e, em seguida, por várias combinações de dimensão dentro de 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.

Índice

Métricas do CloudWatch para monitorar o uso de recursos

A tabela a seguir lista as métricas do CloudWatch que estão disponíveis para auxiliar no monitoramento do uso de recursos. A lista não é exaustiva, mas é um bom ponto de partida. Se você não vir essas métricas no console do CloudWatch, verifique se você concluiu a configuração do recurso. Para obter mais informações, consulte o Guia do usuário do Amazon CloudWatch.

Recursos escaláveis	Namespace	métrica do cloudwatch	Link para a documentação
AppStream 2.0			
Frotas	AWS/AppStream	Nome: AvailableCapacity Dimensão: frota	Métricas do AppStream 2.0
Frotas	AWS/AppStream	Nome: CapacityUtilization Dimensão: frota	Métricas do AppStream 2.0
Aurora			
Réplicas	AWS/RDS	Nome: CPUUtilization	Métricas no nível do cluster do Aurora

Recursos escaláveis	Namespace	métrica do cloudwatch	Link para a documentação
		Dimensões: DBClusterIdentifier, perfil (LEITOR)	
Réplicas	AWS/RDS	Nome: DatabaseConnections Dimensões: DBClusterIdentifier, perfil (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

Recursos escaláveis	Namespace	métrica do cloudwatch	Link para a documentação
Endpoints do reconhecedor de entidades	AWS/Comprehend	Nome: InferenceUtilization Dimensão: EndpointArn	Métricas de endpoint do Amazon Comprehend
DynamoDB			
Tabelas e índices secundários globais	AWS/DynamoDB	Nome: ProvisionedReadCapacityUnits Dimensões: TableName, GlobalSecondaryIndexName	Métricas do DynamoDB

Recursos escaláveis	Namespace	métrica do cloudwatch	Link para a documentação
Tabelas e índices secundários globais	AWS/DynamoDB	Nome: ProvisionedWriteCapacityUnits Dimensões: TableName, GlobalSecondaryIndexName	Métricas do DynamoDB
Tabelas e índices secundários globais	AWS/DynamoDB	Nome: ConsumedReadCapacityUnits Dimensões: TableName, GlobalSecondaryIndexName	Métricas do DynamoDB

Recursos escaláveis	Namespace	métrica do cloudwatch	Link para a documentação
Tabelas e índices secundários globais	AWS/DynamoDB	Nome: ConsumedWriteCapacityUnits Dimensões: TableName, GlobalSecondaryIndexName	Métricas do DynamoDB
Amazon ECS			
Serviços	AWS/ECS	Nome: CPUUtilization Dimensões: ClusterName, ServiceName	Métricas do Amazon ECS

Recursos escaláveis	Namespace	métrica do cloudwatch	Link para a documentação
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			
Clusters (grupos de replicação)	AWS/ElastiCache	Nome: DatabaseMemoryUsageCountedForEvictPercentage Dimensão: ReplicationGroupId	Métricas do ElastiCache para Redis

Recursos escaláveis	Namespace	métrica do cloudwatch	Link para a documentação
Clusters (grupos de replicação)	AWS/ElastiCache	Nome: DatabaseCapacityUsageCountedForEvictPercentage Dimensão: ReplicationGroupId	Métricas do ElastiCache para Redis
Clusters (grupos de replicação)	AWS/ElastiCache	Nome: EngineCPUUtilization Dimensões: ReplicationGroupId, perfil (primário)	Métricas do ElastiCache para Redis
Clusters (grupos de replicação)	AWS/ElastiCache	Nome: EngineCPUUtilization Dimensões: ReplicationGroupId, perfil (réplica)	Métricas do ElastiCache para Redis

Recursos escaláveis	Namespace	métrica do cloudwatch	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	métrica do cloudwatch	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

Recursos escaláveis	Namespace	métrica do cloudwatch	Link para a documentação
Amazon MSK			
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: DBClusterIdentifier, perfil (LEITOR)	Métricas do Neptune

Recursos escaláveis	Namespace	métrica do cloudwatch	Link para a documentação
SageMaker			
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

Recursos escaláveis	Namespace	métrica do cloudwatch	Link para a documentação
Simultaneidade provisionada para um endpoint sem servidor	AWS/SageMaker	Nome: ServerlessProvisionedConcurrencyUtilization Dimensões: EndpointName, VariantName	Métricas de endpoint de tecnologia sem servidor
Frota spot (Amazon EC2)			
Spot Fleets	AWS/EC2spot	Nome: CPUUtilization Dimensão: FleetRequestId	Métricas de frota spot
Spot Fleets	AWS/EC2spot	Nome: NetworkIn Dimensão: FleetRequestId	Métricas de frota spot

Recursos escaláveis	Namespace	métrica do cloudwatch	Link para a documentação
Spot Fleets	AWS/EC2Spot	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 predefinidas da Referência de API do Application Auto Scaling com o nome da métrica correspondente do CloudWatch. Cada métrica predefinida representa uma agregação dos valores da métrica subjacente do CloudWatch. 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 Métricas do CloudWatch para monitorar o uso de recursos.

Tipo de métrica predefinida	Nome da métrica do CloudWatch
AppStream 2.0	

Tipo de métrica predefinida	Nome da métrica do CloudWatch
`AppStreamAverageCapacityUtilization`	CapacityUtilization
Aurora	
`RDSReaderAverageCPUUtilization`	CPUUtilization
`RDSReaderAverageDatabaseConnections`	DatabaseConnections[1]
Amazon Comprehend	
`ComprehendInferenceUtilization`	InferenceUtilization
DynamoDB	
`DynamoDBReadCapacityUtilization`	ProvisionedReadCapacityUnits, ConsumedReadCapacityUnits[2]
`DynamoDBWriteCapacityUtilization`	ProvisionedWriteCapacityUnits, ConsumedWriteCapacityUnits[2]
Amazon ECS	
`ECSServiceAverageCPUUtilization`	CPUUtilization
`ECSServiceAverageMemoryUtilization`	MemoryUtilization
`ALBRequestCountPerTarget`	RequestCountPerTarget[1]
ElastiCache	
`ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage`	DatabaseMemoryUsageCountedForEvictPercentage
`ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage`	DatabaseCapacityUsageCountedForEvictPercentage

Tipo de métrica predefinida	Nome da métrica do CloudWatch
`ElastiCachePrimaryEngineCPUUtilization`	EngineCPUUtilization
`ElastiCacheReplicaEngineCPUUtilization`	EngineCPUUtilization
Amazon Keyspaces	
`CassandraReadCapacityUtilization`	ProvisionedReadCapacityUnits, ConsumedReadCapacityUnits[2]
`CassandraWriteCapacityUtilization`	ProvisionedWriteCapacityUnits, ConsumedWriteCapacityUnits[2]
Lambda	
`LambdaProvisionedConcurrencyUtilization`	ProvisionedConcurrencyUtilization
Amazon MSK	
`KafkaBrokerStorageUtilization`	KafkaDataLogsDiskUsed
Neptune	
`NeptuneReaderAverageCPUUtilization`	CPUUtilization
SageMaker	
`SageMakerVariantInvocationsPerInstance`	InvocationsPerInstance[1]
`SageMakerInferenceComponentInvocationsPerCopy`	InvocationsPerCopy[1]
`SageMakerVariantProvisionedConcurrencyUtilization`	ServerlessProvisionedConcurrencyUtilization

Tipo de métrica predefinida	Nome da métrica do CloudWatch
Frota spot	
`EC2SpotFleetRequestAverageCPUUtilization`	CPUUtilization[3]
`EC2SpotFleetRequestAverageNetworkIn`[3]	NetworkIn[1] [3]
`EC2SpotFleetRequestAverageNetworkOut`[3]	NetworkOut[1] [3]
`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 do CloudWatch para permitir a escalação com base no consumo do throughput provisionado.

[3] Para obter o melhor desempenho de escalação, o monitoramento detalhado do Amazon EC2 deve ser usado.

Monitorar eventos do Application Auto Scaling com o Amazon EventBridge

Anteriormente chamado de CloudWatch Events, o Amazon EventBridge ajuda a monitorar eventos que sejam específicos do Application Auto Scaling e a iniciar ações direcionados que utilizem outros Serviços da AWS. Os eventos dos Serviços da AWS são entregues ao EventBridge praticamente em tempo real.

Usando o Amazon EventBridge, você pode criar regras que façam a correspondência com eventos recebidos e os encaminhem aos destinos para processamento.

Para obter mais informações, consulte Começar a usar o Amazon EventBridge no Manual do usuário do Amazon EventBridge.

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 escalonamento máximo e 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 por meio do 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 por meio do CloudTrail

Uma trilha é uma configuração que o AWS CloudTrail usa para entregar eventos como arquivos de log a um bucket do Amazon S3. Os arquivos de log do CloudTrail 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 a usar o CloudTrail, consulte Criar uma trilha no Guia do usuário do AWS CloudTrail.

Os eventos que são entregues por meio do CloudTrail têm `AWS API Call via CloudTrail` como o valor para `detail-type`.

O evento de exemplo a seguir representa uma entrada de arquivo de log do CloudTrail que mostra que um usuário do console chamou a ação RegisterScalableTarget do Application Auto Scaling.

```
{
  "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 as chamadas de API DeleteScalingPolicy e DeregisterScalableTarget para todos os destinos escaláveis, use a seguinte amostra 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 ter mais informações sobre o uso de CloudTrail, consulte Log de chamadas de API do Application Auto Scaling com o AWS CloudTrail.

Notificações de AWS Health Dashboard do Application Auto Scaling

Para ajudar a gerenciar eventos de escalabilidade com falha, o AWS Health Dashboard é compatível com notificações emitidas pelo Application Auto Scaling. Somente eventos de expansão que são específicos para seus recursos do DynamoDB estão disponíveis no momento.

O exemplo de AWS Health Dashboard é parte do serviço AWS Health. Ele não requer nenhuma configuração e pode ser visualizado por qualquer usuário autenticado em sua conta. Para obter mais informações, consulte Conceitos básicos sobre o AWS Health Dashboard.

Se os recursos do DynamoDB não estiverem expandindo devido aos limites de cota de serviço do DynamoDB, você receberá uma mensagem semelhante à seguinte. Se você receber esta mensagem, ela deverá ser tratada como um alarme para executar uma ação.

```
  Hello,

  A scaling action has attempted to scale out your DynamoDB resources in the
  eu-west-1 region. This operation has been prevented because it would have exceeded a
  table-level write throughput limit (Provisioned mode). This limit restricts the
  provisioned write capacity of the table and all of its associated global secondary
  indexes. To address the issue, refer to the Amazon DynamoDB Developer Guide for
  current limits and how to request higher limits [1].

  To identify your DynamoDB resources that are impacted, use the
  describe-scaling-activities command or the DescribeScalingActivities operation [2]
[3].
  Look for a scaling activity with StatusCode "Failed" and a StatusMessage similar to
  "Failed to set write capacity units to 45000. Reason: The requested
 WriteCapacityUnits,
  45000, is above the per table maximum for the account in eu-west-1. Per table
 maximum:
  40000." You can also view these scaling activities from the Capacity tab of your
 tables
  in the AWS Management Console for DynamoDB.

  We strongly recommend that you address this issue to ensure that your tables
  are prepared to handle increases in traffic. This notification is sent only once in
```

```
  each 12 hour period, even if another failed scaling action occurs.

  [1] https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/
Limits.html#default-limits-throughput-capacity-modes
  [2] https://docs.aws.amazon.com/cli/latest/reference/application-autoscaling/
describe-scaling-activities.html
  [3] https://docs.aws.amazon.com/autoscaling/application/APIReference/
API_DescribeScalingActivities.html

  Sincerely,
  Amazon Web Services
```

Suporte de marcação para o Application Auto Scaling

É possível usar a AWS CLI ou um SDK para adicionar etiquetas a destinos escaláveis do Application Auto Scaling. Os destinos escaláveis correspondem às entidades que representam a AWS ou os recursos personalizados que podem ser escalados com o Application Auto Scaling.

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-target para marcar novos destinos escaláveis ao registrá-los.
- tag-resource para adicionar etiquetas a um destino escalável existente.
- list-tags-for-resource para retornar as etiquetas em um destino escalável.
- untag-resource para excluir uma etiqueta.

Exemplo de marcação

Use o comando register-scalable-target, a seguir, com a opção `--tags`. 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 exemplo por `--min-capacity` e `--max-capacity` e o texto de exemplo por `--service-namespace` com o namespace do serviço da AWS que você está usando com o Application Auto Scaling, `--scalable-dimension` pela dimensão escalável associada ao recurso que você está registrando e `--resource-id` por um identificador para o recurso. Para obter mais informações e exemplos de cada serviço, consulte os tópicos na AWS serviços 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 lançar um erro, verifique se você atualizou a 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.

```
{
    "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.

Tópicos

Application Auto Scaling e endpoints da VPC de interface

É possível melhorar o procedimento de segurança da sua VPC configurando o Application Auto Scaling para usar um endpoint da VPC de interface. Os endpoints de interface são alimentados por AWS PrivateLink uma tecnologia que permite acessar de forma privada as APIs do Application Auto Scaling, restringindo todo o tráfego de rede entre sua VPC e o Application Auto Scaling à rede. AWS Com endpoints de interface, também não são necessários um gateway da Internet, um dispositivo NAT nem um gateway privado virtual.

Não é necessário configurar AWS PrivateLink, mas é recomendado. Para obter mais informações sobre AWS PrivateLink endpoints de VPC, consulte O que é? AWS PrivateLink no AWS PrivateLink Guia.

Tópicos

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).

Application Auto Scaling e proteção de dados

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 seu conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para ter mais informações sobre a privacidade de dados, consulte as Perguntas frequentes sobre privacidade de dados. Para ter mais informações sobre a proteção de dados na Europa, consulte a postagem do blog AWS Shared Responsibility Model and GDPR 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 os recursos. AWS Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure a API e o registro de atividades do usuário com AWS CloudTrail.
- 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 140-2 ao acessar AWS por meio de uma interface de linha de comando ou de uma API, use um endpoint FIPS. Para ter mais informações sobre endpoints do FIPS, consulte Federal Information Processing Standard (FIPS) 140-2.

É altamente recomendável que nunca sejam colocadas informações de identificação confidenciais, como endereços de e-mail dos seus clientes, em marcações 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 os AWS CLI SDKs. Quaisquer dados inseridos em tags ou 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, recomendamos fortemente que não sejam incluídas informações de credenciais no URL para validar a solicitação a esse 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 usar o Application Auto Scaling, você precisa de um Conta da AWS e de suas credenciais de segurança para entrar na sua conta. Para ter mais informações, consulte Configurar o uso do Application Auto Scaling.

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.

Tópicos

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

Atributo do IAM	Compatibilidade com o aplicativo 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 (rótulos em políticas)	Parcial
Credenciais temporárias	Sim
Perfis de serviço	Sim
Perfis vinculados ao 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

É 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, grupo de usuários ou perfil do IAM. Essas políticas controlam quais ações os usuários e funções podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Criar políticas do IAM no Guia do usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações ou recursos permitidos ou negados, bem 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 função à qual ela está anexada. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte Referência de elementos da 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

Oferece suporte a 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 de ações do Application Auto Scaling, consulte Ações definidas pelo AWS Application Auto Scaling na Referência de Autorização de Serviço.

Recursos

Oferece suporte a recursos de políticas	Sim

Em uma instrução de política do IAM, o elemento `Resource` especifica o objeto ou os objetos abrangidos pela instrução. Para o Application Auto Scaling, cada instrução de política do IAM se aplica aos destinos escaláveis especificados usando os nomes dos recursos da Amazon (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 caso uma ação de API específica não ofereça suporte a ARNs, use um curinga (*) como o elemento `Resource`, conforme descrito a seguir.

```
"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 do 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

Oferece suporte a políticas baseadas em recurso	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)

Oferece suporte a ACLs	Não

O Application Auto Scaling não é compatível com listas de controla de acesso (ACLs).

ABAC com o Application Auto Scaling

Oferece suporte a ABAC (tags em políticas)	Parcial

O controle de acesso baseado em recurso (ABAC) é uma estratégia de autorização que define permissões com base em recursos. Em AWS, esses atributos são chamados de tags. Você pode anexar tags a entidades do IAM (usuários ou funções) e a vários AWS recursos. A marcação de entidades e recursos 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 está 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 `aws:TagKeys` chaves de condição.

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 ter mais informações, consulte Suporte de marcação para o Application Auto Scaling.

Para 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 Usar controle de acesso por atributo (ABAC) no Guia do usuário do IAM.

Usar credenciais temporárias com o Application Auto Scaling

Oferece suporte a 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 mais informações sobre como alternar funções, consulte Alternar para uma função (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 mais informações, consulte Credenciais de segurança temporárias no IAM.

Perfis de serviço

Oferece suporte a 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 indica que um administrador do IAM pode alterar as permissões para essa função. 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.

Perfis vinculados ao serviço

Oferece suporte a funções vinculadas ao serviço	Sim

Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode assumir 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 pode editar as permissões para perfis vinculados ao 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 políticas gerenciadas pelo cliente 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) for lançada ou novas operações de API forem disponibilizadas para serviços existentes.

Para mais informações, consulte Políticas gerenciadas pela AWS no Manual do usuário do IAM.

Conteúdo

- AWS política gerenciada que concede acesso ao Amazon MSK e CloudWatch
- AWS política gerenciada que concede acesso a Neptune e CloudWatch
- AWS política gerenciada que concede acesso a e SageMaker CloudWatch
- AWS política gerenciada que concede acesso ao EC2 Spot Fleet e CloudWatch
- AWS política gerenciada que concede acesso aos seus recursos personalizados e CloudWatch
- Atualizações do Application Auto Scaling para políticas AWS gerenciadas

AWS política gerenciada que concede acesso ao AppStream 2.0 e CloudWatch

Nome da política: AWSApplicationAutoscalingAppStreamFleetPolicy

Não é possível associar `AWSApplicationAutoscalingAppStreamFleetPolicy` às suas identidades do IAM (usuários ou perfis). Essa política está vinculada a uma função vinculada ao serviço que permite que o Application Auto Scaling ligue para a AppStream Amazon CloudWatch e realize escalabilidade em seu nome.

Detalhes de permissões

A política de permissões da função vinculada ao serviço da `AWSServiceRoleForApplicationAutoScaling_AppStreamFleet` permite que Application Auto Scaling conclua as seguintes ações em todos os recursos relacionados ("Recurso": "*"):

- Ação: `appstream:DescribeFleets`
- Ação: `appstream:UpdateFleet`
- Ação: `cloudwatch:DescribeAlarms`
- Ação: `cloudwatch:PutMetricAlarm`
- Ação: `cloudwatch:DeleteAlarms`

AWS política gerenciada que concede acesso ao Aurora e CloudWatch

Nome da política: AWSApplicationAutoscalingRDSClusterPolicy

Não é possível associar `AWSApplicationAutoscalingRDSClusterPolicy` às suas identidades do IAM (usuários ou perfis). Essa política está vinculada a uma função vinculada ao serviço que permite que o Application Auto Scaling chame a Aurora CloudWatch e realize o escalonamento em seu nome.

Detalhes de permissões

A política de permissões da função vinculada ao serviço da `AWSServiceRoleForApplicationAutoScaling_RDSCluster` permite que Application Auto Scaling conclua as seguintes ações em todos os recursos relacionados ("Recurso": "*"):

- 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 que concede acesso ao Amazon Comprehend e CloudWatch

Nome da política: AWSApplicationAutoscalingComprehendEndpointPolicy

Não é possível associar `AWSApplicationAutoscalingComprehendEndpointPolicy` às suas identidades do IAM (usuários ou perfis). Essa política está vinculada a uma função vinculada ao serviço que permite que o Application Auto Scaling chame o Amazon Comprehend e realize escalabilidade em seu CloudWatch nome.

Detalhes de permissões

A política de permissões da função vinculada ao serviço da `AWSServiceRoleForApplicationAutoScaling_ComprehendEndpoint` permite que Application Auto Scaling conclua as seguintes ações em todos os recursos relacionados ("Recurso": "*"):

- Ação: `comprehend:UpdateEndpoint`
- Ação: `comprehend:DescribeEndpoint`
- Ação: `cloudwatch:DescribeAlarms`
- Ação: `cloudwatch:PutMetricAlarm`
- Ação: `cloudwatch:DeleteAlarms`

AWS política gerenciada que concede acesso ao DynamoDB e CloudWatch

Nome da política: AWSApplicationAutoscalingDynamoDBTablePolicy

Não é possível associar `AWSApplicationAutoscalingDynamoDBTablePolicy` às suas identidades do IAM (usuários ou perfis). Essa política está vinculada a uma função vinculada ao serviço que permite que o Application Auto Scaling chame o DynamoDB e realize o escalonamento em seu nome. CloudWatch

Detalhes de permissões

A política de permissões da função vinculada ao serviço da `AWSServiceRoleForApplicationAutoScaling_DynamoDBTable` permite que Application Auto Scaling conclua as seguintes ações em todos os recursos relacionados ("Recurso": "*"):

- Ação: `dynamodb:DescribeTable`
- Ação: `dynamodb:UpdateTable`
- Ação: `cloudwatch:DescribeAlarms`
- Ação: `cloudwatch:PutMetricAlarm`
- Ação: `cloudwatch:DeleteAlarms`

AWS política gerenciada que concede acesso ao Amazon ECS e CloudWatch

Nome da política: AWSApplicationAutoscalingECSServicePolicy

Não é possível associar `AWSApplicationAutoscalingECSServicePolicy` às suas identidades do IAM (usuários ou perfis). Essa política está vinculada a uma função vinculada ao serviço que permite que o Application Auto Scaling chame o Amazon ECS CloudWatch e realize escalabilidade em seu nome.

Detalhes de permissões

A política de permissões da função vinculada ao serviço da `AWSServiceRoleForApplicationAutoScaling_ECSService` permite que Application Auto Scaling conclua as seguintes ações em todos os recursos relacionados ("Recurso": "*"):

- Ação: `ecs:DescribeServices`
- Ação: `ecs:UpdateService`

- Ação: `cloudwatch:DescribeAlarms`
- Ação: `cloudwatch:PutMetricAlarm`
- Ação: `cloudwatch:DeleteAlarms`

AWS política gerenciada que concede acesso a e ElastiCache CloudWatch

Nome da política: AWSApplicationAutoscalingElastiCacheRGPolicy

Não é possível associar `AWSApplicationAutoscalingElastiCacheRGPolicy` às suas identidades do IAM (usuários ou perfis). 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.

Detalhes de permissões

A política de permissões da função vinculada a serviço `AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG` permite 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: `cloudwatch:DescribeAlarms` em todos os recursos
- Ação: `cloudwatch:PutMetricAlarm` no recurso `arn:*:cloudwatch:*:*:alarm:TargetTracking*`
- Ação: `cloudwatch:DeleteAlarms` no recurso `arn:*:cloudwatch:*:*:alarm:TargetTracking*`
- Ação: `cloudwatch:DeleteAlarms`

AWS política gerenciada que concede acesso ao Amazon Keyspaces e CloudWatch

Nome da política: AWSApplicationAutoscalingCassandraTablePolicy

Não é possível associar `AWSApplicationAutoscalingCassandraTablePolicy` às suas identidades do IAM (usuários ou perfis). Essa política está vinculada a uma função vinculada ao serviço que permite que o Application Auto Scaling chame o Amazon Keyspaces CloudWatch e realize escalabilidade em seu nome.

Detalhes de permissões

A política de permissões da função vinculada a serviço `AWSServiceRoleForApplicationAutoScaling_CassandraTable` permite que o Application Auto Scaling conclua as seguintes ações nos recursos especificados:

- Ação: `cassandra:Select` no recurso `arn:*:cassandra:*:*:/keyspace/system/table/*`
- Ação: `cassandra:Select` no recurso `arn:*:cassandra:*:*:/keyspace/system_schema/table/*`
- Ação: `cassandra:Select` no recurso `arn:*:cassandra:*:*:/keyspace/system_schema_mcs/table/*`
- Ação: `cassandra:Alter` no recurso `arn:*:cassandra:*:*:"*"`
- Ação: `cloudwatch:DescribeAlarms`
- Ação: `cloudwatch:PutMetricAlarm`
- Ação: `cloudwatch:DeleteAlarms`

AWS política gerenciada que concede acesso ao Lambda e CloudWatch

Nome da política: AWSApplicationAutoscalingLambdaConcurrencyPolicy

Não é possível associar `AWSApplicationAutoscalingLambdaConcurrencyPolicy` às suas identidades do IAM (usuários ou perfis). Essa política é anexada a uma função vinculada ao serviço que permite que o Application Auto Scaling chame o Lambda CloudWatch e realize o escalonamento em seu nome.

Detalhes de permissões

A política de permissões da função vinculada ao serviço da `AWSServiceRoleForApplicationAutoScaling_LambdaConcurrency` permite que Application Auto Scaling conclua as seguintes ações em todos os recursos relacionados ("Recurso": "*"):

- Ação: `lambda:PutProvisionedConcurrencyConfig`

- Ação: `lambda:GetProvisionedConcurrencyConfig`
- Ação: `lambda:DeleteProvisionedConcurrencyConfig`
- Ação: `cloudwatch:DescribeAlarms`
- Ação: `cloudwatch:PutMetricAlarm`
- Ação: `cloudwatch:DeleteAlarms`

AWS política gerenciada que concede acesso ao Amazon MSK e CloudWatch

Nome da política: AWSApplicationAutoscalingKafkaClusterPolicy

Não é possível associar `AWSApplicationAutoscalingKafkaClusterPolicy` às suas identidades do IAM (usuários ou perfis). Essa política está vinculada a uma função vinculada ao serviço que permite que o Application Auto Scaling chame o Amazon MSK CloudWatch e realize a escalabilidade em seu nome.

Detalhes de permissões

A política de permissões da função vinculada ao serviço da `AWSServiceRoleForApplicationAutoScaling_KafkaCluster` permite que Application Auto Scaling conclua as seguintes ações em todos os recursos relacionados ("Recurso": "*"):

- Ação: `kafka:DescribeCluster`
- Ação: `kafka:DescribeClusterOperation`
- Ação: `kafka:UpdateBrokerStorage`
- Ação: `cloudwatch:DescribeAlarms`
- Ação: `cloudwatch:PutMetricAlarm`
- Ação: `cloudwatch:DeleteAlarms`

AWS política gerenciada que concede acesso a Neptune e CloudWatch

Nome da política: AWSApplicationAutoscalingNeptuneClusterPolicy

Não é possível associar `AWSApplicationAutoscalingNeptuneClusterPolicy` às suas identidades do IAM (usuários ou perfis). 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.

Detalhes de permissões

A política de permissões da função vinculada a serviço `AWSServiceRoleForApplicationAutoScaling_NeptuneCluster` permite que o Application Auto Scaling conclua as seguintes ações nos recursos especificados:

- 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:ListTagsForResource` em todos os recursos
- 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:DescribeDBInstances` em todos os recursos
- Ação: `rds:DescribeDBClusters` em todos os recursos
- Ação: `rds:DescribeDBClusterParameters` em todos os recursos
- Ação: `rds:DeleteDBInstance` no recurso `arn:*:rds:*:*:db:autoscaled-reader*`
- Ação: `cloudwatch:DescribeAlarms` em todos os recursos
- Ação: `cloudwatch:PutMetricAlarm` no recurso `arn:*:cloudwatch:*:*:alarm:TargetTracking*`
- Ação: `cloudwatch:DeleteAlarms` no recurso `arn:*:cloudwatch:*:*:alarm:TargetTracking*`
- Ação: `cloudwatch:DeleteAlarms`

AWS política gerenciada que concede acesso a e SageMaker CloudWatch

Nome da política: AWSApplicationAutoscalingSageMakerEndpointPolicy

Não é possível associar `AWSApplicationAutoscalingSageMakerEndpointPolicy` às suas identidades do IAM (usuários ou perfis). Essa política está vinculada a uma função vinculada ao serviço que permite que o Application Auto Scaling SageMaker chame CloudWatch e realize o escalonamento em seu nome.

Detalhes de permissões

A política de permissões da função vinculada a serviço `AWSServiceRoleForApplicationAutoScaling_SageMakerEndpoint` permite 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:PutMetricAlarm` no recurso `arn:*:cloudwatch:*:*:alarm:TargetTracking*`
- Ação: `cloudwatch:DeleteAlarms` no recurso `arn:*:cloudwatch:*:*:alarm:TargetTracking*`

AWS política gerenciada que concede acesso ao EC2 Spot Fleet e CloudWatch

Nome da política: AWSApplicationAutoscalingEC2SpotFleetRequestPolicy

Não é possível associar `AWSApplicationAutoscalingEC2SpotFleetRequestPolicy` às suas identidades do IAM (usuários ou perfis). Essa política está vinculada a uma função vinculada ao serviço que permite que o Application Auto Scaling chame o Amazon EC2 CloudWatch e realize escalabilidade em seu nome.

Detalhes de permissões

A política de permissões da função vinculada ao serviço da `AWSServiceRoleForApplicationAutoScaling_EC2SpotFleetRequest` permite que Application Auto Scaling conclua as seguintes ações em todos os recursos relacionados ("Recurso": "*"):

- Ação: `ec2:DescribeSpotFleetRequests`
- Ação: `ec2:ModifySpotFleetRequest`
- Ação: `cloudwatch:DescribeAlarms`
- Ação: `cloudwatch:PutMetricAlarm`
- Ação: `cloudwatch:DeleteAlarms`

AWS política gerenciada que concede acesso aos seus recursos personalizados e CloudWatch

Nome da política: AWSApplicationAutoScalingCustomResourcePolicy

Não é possível associar `AWSApplicationAutoScalingCustomResourcePolicy` às suas identidades do IAM (usuários ou perfis). Essa política está vinculada a uma função vinculada ao serviço que permite 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 de permissões da função vinculada ao serviço da `AWSServiceRoleForApplicationAutoScaling_CustomResource` permite que Application Auto Scaling conclua as seguintes ações em todos os recursos relacionados ("Recurso": "*"):

- 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
O Application Auto Scaling adiciona permissões à sua função vinculada ao serviço SageMaker	Agora, essa política concede permissões ao serviço para chamar as ações de `UpdateInferenceComponentRuntimeConfig` API SageMaker `DescribeInferenceComponent` e	13 de novembro de 2023

Alteração	Descrição	Data
	dar suporte à compatibilidade do escalonamento automático de SageMaker recursos 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.	
O Application Auto Scaling adiciona a política do Neptune	O Application Auto Scaling adicionou uma nova 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
Application Auto Scaling adiciona à política do ElastiCache Redis	O Application Auto Scaling adicionou uma nova 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

Alteração	Descrição	Data
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.

Conteúdo

Visão geral

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 ter mais informações, consulte AWS serviços que você pode usar com o Application Auto Scaling.

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.

As seções a seguir descrevem como criar e gerenciar funções vinculadas ao serviço do Application Auto Scaling. Comece configurando permissões para que uma entidade do IAM (por exemplo, um usuário, um grupo ou uma função) crie, edite ou exclua uma 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.

```
{
    "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 AWS serviços que você pode usar com o Application Auto Scaling.

Criar funções vinculadas a serviços (automático)

Não é necessário criar manualmente uma função vinculada a 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 obter mais informações, consulte Criar uma função vinculada ao serviço no Manual do usuário do IAM.

Para criar uma função vinculada a serviço (AWS CLI)

Use o comando create-service-linked-roleCLI 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 AWS serviços 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 ter mais informações, consulte Editar um 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 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 um perfil vinculado 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

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`

Serviço	ARN
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/AWSServiceRoleForApplicationAutoScaling_LambdaConcurrency
MSK	arn:aws:iam:: *012345678910* :role/aws-service-role/kafka.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_KafkaCluster

Serviço	ARN
Neptune	`arn:aws:iam:: 012345678910 :role/aws-service-role/neptune.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_NeptuneCluster`
SageMaker	`arn:aws:iam:: 012345678910 :role/aws-service-role/sagemaker.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_SageMakerEndpoint`
Spot Fleets	`arn:aws:iam:: 012345678910 :role/aws-service-role/ec2.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_EC2SpotFleetRequest`
Recursos personalizados	`arn:aws:iam:: 012345678910 :role/aws-service-role/custom-resource.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_CustomResource`

> **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

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.

```
{
    "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.

```
{
    "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.

```
{
    "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

A política baseada em identidade a seguir concede permissões para todas as ações AppStream 2.0 e de CloudWatch API necessárias.

```
{
    "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.

```
{
    "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 necessárias.

```
{
    "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.

```
{
    "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.

```
{
    "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.

```
{
    "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 Amazon EMR necessárias.

```
{
```

```
    "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.

```
{
    "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.

```
{
    "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.

```
{
    "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.

```
{
    "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

A política baseada em identidade a seguir concede permissões para todas SageMaker as ações de CloudWatch API necessárias.

```
{
    "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": "*"
        }
    ]
}
```

Frotas spot (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.

```
{
    "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.

```
{
    "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 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 escalação de frota no console do Amazon EC2 deve ter as permissões adequadas para os serviços compatíveis com escalação dinâmica.

```
{
    "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 console do Amazon EC2 e criem e CloudWatch gerenciem alarmes no console. CloudWatch

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 usar 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 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 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 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:CreateDBInstance: para determinar se o usuário tem essa permissão, enviamos uma solicitação para a operação da API `CreateDBInstance`, tentando criar uma instância de banco

de dados com parâmetros inválidos (ID de instância vazio) no cluster de banco de dados do Aurora especificado pelo usuário. 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:DeleteDBInstance: enviamos um ID de instância vazio para a operação da API `DeleteDBInstance`. 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:DescribeDBCluster: descrevemos o nome do cluster para o recurso que está sendo registrado para autoescalabilidade. 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:DescribeDBInstance: chamamos a API `DescribeDBInstance` com um filtro `db-cluster-id` 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 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.

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:

- Guias de início rápido sobre segurança e conformidade — Esses guias de implantação discutem considerações arquitetônicas e fornecem etapas para a implantação de ambientes básicos AWS focados em segurança e conformidade.
- Arquitetura para segurança e conformidade com a HIPAA na Amazon Web Services — Este whitepaper descreve como as empresas podem usar AWS para criar aplicativos qualificados para a HIPAA.

> **Note**
>
> Nem todos Serviços da AWS são elegíveis para a HIPAA. Para obter mais informações, consulte a Referência dos serviços qualificados pela 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.
- 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.

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 mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de datacenter 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 suporte para:

- Transport Layer Security (TLS). Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Conjuntos de criptografia com sigilo de encaminhamento perfeito (perfect forward secrecy, ou PFS) como DHE (Ephemeral Diffie-Hellman, ou Efêmero Diffie-Hellman) ou ECDHE (Ephemeral Elliptic Curve Diffie-Hellman, ou Curva elíptica efêmera Diffie-Hellman). A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas utilizando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou você pode usar o AWS Security Token Service (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Cotas do Application Auto Scaling

A Conta da AWS tem cotas padrão, anteriormente chamadas de limites, para cada 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. Se a cota ainda não estiver disponível em Service Quotas, use o Application Auto Scaling limits form (Formulário de aumento de limite do Application Auto Scaling). Certifique-se de especificar o tipo de recurso na sua solicitação de aumento, por exemplo, Amazon ECS ou DynamoDB.

A Conta da AWS tem as seguintes cotas relacionadas ao Application Auto Scaling.

Cotas-padrão por região por conta

Item	Padrão	Ajustável
Número máximo de destinos dimensionáveis por tipo de recurso	As cotas padrão variam dependendo do tipo de recurso. Até 5.000 destinos escaláveis do Amazon DynamoDB, 3.000 destinos escaláveis do ECS, 1.500 destinos escaláveis do Amazon Keyspaces e 500 alvos escalávei s, cada um para todos os outros tipos de recursos.	Sim

Item	Padrão	Ajustável
O número máximo de políticas de escalabilidade por destino escalável	50 Inclui políticas de escalabilidade em etapas e políticas de rastreamento de destino.	Não
O número máximo de ações programadas por destino dimensionável	200	Não
O número máximo de ajustes em etapas por política de escalabilidade em etapas	20	Não

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 outros serviços da AWS, consulte Endpoints e cotas de serviços no Referência geral da Amazon Web Services.

Histórico do documento

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
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 SageMaker inferência	Use o Application Auto Scaling para escalar cópias de um componente de inferência.	29 de novembro de 2023
Atualizar permissões de função vinculada a serviços do IAM	O Application Auto Scaling atualiza a política da `AWSApplicationAutoscalingSageMakerEndpointPolicy` . Para obter mais informações, consulte Atualizações do Application Auto Scaling para políticas gerenciadas da AWS.	13 de novembro de 2023
Support para simultaneidade SageMaker provisionada sem servidor	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	20 de março de 2023

	Auto Scaling na forma de etiquetas. Consulte Suporte de marcação para o Application Auto Scaling.	
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
Alterações do guia	Um novo tópico no Guia do usuário do Application Auto Scaling ajuda você a começar a usar o AWS CloudShell com o Application Auto Scaling. Consulte Usar o AWS CloudShell para trabalhar com o Application Auto Scaling na linha de comando.	17 de fevereiro 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 as alterações nas políticas gerenciadas pela AWS	Desde 19 de agosto de 2021, as alterações nas políticas gerenciadas são relatadas no tópico Atualizações do Application Auto Scaling nas políticas gerenciadas pela AWS. A primeira alteração listada é a adição das permissões necessárias ElastiCache para o Redis.	19 de agosto de 2021
Adicione suporte ElastiCache para grupos de replicação do Redis	Use o Application Auto Scaling para escalar o número de grupos de nós e o número de réplicas por grupo de nós ElastiCache para um grupo de replicação do Redis (cluster). Para obter mais informações, consulte ElastiCache Redis e Application Auto Scaling.	19 de agosto de 2021

Alterações do guia	Novos tópicos do IAM no Manual do usuário do Application Auto Scaling ajudam você a solucionar problemas de acesso ao Application Auto Scaling. Para obter mais informações, consulte Gerenciamento de Identidade e Acesso para o Application 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 Examplo de políticas para trabalhar com AWS CLI ou 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. Além disso, um novo tópico explica como acionar uma notificação quando for CloudWatch detectado algum problema que possa exigir sua atenção.	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 compatibilidade com funções do AWS Lambda	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
Nova seção	A seção Setting up (Configuração) foi adicionada à documentação do Application Auto Scaling. Melhorias e correções menores foram feitas em todo o guia do usuário.	28 de junho de 2019

Alterações do guia	A documentação do Application Auto Scaling nas seções Scheduled scaling (Escalabilidade programada), Step scaling policies (Políticas de escalabilidade em etapas) e Target tracking scaling policies (Políticas de dimensionamento 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 SageMaker de endpoint	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	17 de novembro de 2017

Alteração	Descrição	Data
	no Manual do usuário do Amazon RDS.	
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

Alteração	Descrição	Data
Adicionar compatibilidade com capacidade de leitura e gravação provisionada para tabelas e índices secundários globais do DynamoDB	Use o Application Auto Scaling para escalar o throughput provisionado (capacidade de leitura e gravação). Para obter mais informações, consulte Como gerenciar a capacidade de throughput com a autoescalabilidade do DynamoDB no Guia do desenvolvedor do Amazon DynamoDB.	14 de junho de 2017
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 automática no Amazon EMR no Guia de gerenciamento do Amazon EMR.	18 de novembro de 2016

Alteração	Descrição	Data
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 Manual do usuário do Amazon EC2 para instâncias do Linux.	1 de setembro de 2016
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.