aws

Guide de l’utilisateur

Application Autoscaling

Application Autoscaling: Guide de l’utilisateur

Table of Contents

En quoi consiste Application Auto Scaling ?

Application Auto Scaling est un service Web destiné aux développeurs et aux administrateurs système qui ont besoin d'une solution permettant de dimensionner automatiquement leurs ressources évolutives pour des AWS services individuels autres qu'Amazon EC2 Auto Scaling. Avec Application Auto Scaling, vous pouvez configurer le dimensionnement automatique pour les ressources suivantes :

- WorkSpaces Flottes d'applications
- Réplicas Aurora
- Classification de documents et points de terminaison de module de reconnaissance d'entité Amazon Comprehend
- Tables DynamoDB et index secondaires globaux
- Services Amazon ECS
- ElastiCache groupes de réplication (Redis OSS et Valkey) et clusters Memcached
- Clusters Amazon EMR
- Tables Amazon Keyspaces (for Apache Cassandra)
- Simultanéité allouée pour la fonction Lambda
- Stockage de l'agent Amazon Managed Streaming for Apache Kafka (MSK)
- Clusters Amazon Neptune
- SageMaker Variantes de terminaux AI
- SageMaker Composants d'inférence AI
- SageMaker Concurrence provisionnée sans serveur AI
- Demandes de parc d'instances Spot
- Pool d'Amazon WorkSpaces
- Ressources personnalisées fournies par vos propres applications ou services. Pour plus d'informations, consultez le GitHubréférentiel.

Pour connaître la disponibilité régionale de l'un des AWS services énumérés ci-dessus, consultez le tableau des .

Pour plus d'informations sur le dimensionnement de votre parc d' EC2 instances Amazon à l'aide de groupes Auto Scaling, consultez le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Caractéristiques d'Application Auto Scaling

Application Auto Scaling vous permet de mettre automatiquement à l'échelle vos ressources évolutives en fonction des conditions que vous définissez.

- Mise à l'échelle du suivi des cibles : redimensionnez une ressource en fonction de la valeur cible d'une CloudWatch métrique spécifique.
- Mise à l'échelle par étapes – Met à l'échelle une ressource en fonction d'un ensemble d'ajustements de mise à l'échelle qui varient en fonction de la valeur du niveau de l'alarme.
- Mise à l'échelle planifiée – Met à l'échelle une ressource une seule fois ou selon un calendrier récurrent.
- Mise à l'échelle prédictive : dimensionnez une ressource de manière proactive pour qu'elle corresponde à la charge prévue en fonction des données historiques.

Fonctionnent avec Application Auto Scaling

Vous pouvez configurer la mise à l'échelle à l'aide des interfaces suivantes en fonction de la ressource pour laquelle vous effectuez une mise à l'échelle :

- AWS Management Console - offre une interface Web que vous pouvez utiliser pour configurer la mise à l'échelle. Créez un AWS compte et connectez-vous au AWS Management Console. Ensuite, ouvrez la console du service pour l'une des ressources répertoriées dans l'introduction. Par exemple, pour redimensionner une fonction Lambda, ouvrez le. AWS Lambda console Assurez-vous d'ouvrir la console au même endroit Région AWS que la ressource avec laquelle vous souhaitez travailler.

  > **Note**
  >
  > L'accès par la console n'est pas disponible pour toutes les ressources. Pour de plus amples informations, veuillez consulter Services AWS que vous pouvez utiliser avec Application Auto Scaling.

- AWS Command Line Interface (AWS CLI) — Fournit des commandes pour un large éventail de Services AWS, et est compatible avec Windows, macOS et Linux. Consultez AWS Command Line Interface pour démarrer. Pour obtenir la liste des commandes, consultez la section application-autoscaling dans le AWS CLI manuel Command Reference.

- AWS Tools for Windows PowerShell— Fournit des commandes pour un large éventail de AWS produits pour ceux qui écrivent des scripts dans l' PowerShell environnement. Consultez le Guide de l'utilisateur Outils AWS pour PowerShell pour démarrer. Pour plus d'informations, consultez le Guide de référence des cmdlets Outils AWS pour PowerShell.
- AWS SDKs— Fournit des opérations d'API spécifiques au langage et prend en charge de nombreux détails de connexion, tels que le calcul des signatures, la gestion des nouvelles tentatives de demande et la gestion des erreurs. Pour plus d'informations, consultez la section Outils sur lesquels vous pouvez vous appuyer AWS.
- API HTTPS : Fournit des actions d'API de bas niveau appelées à l'aide de demandes HTTPS. Pour plus d'informations, consultez la Référence de l'API Application Auto Scaling.
- CloudFormation— Permet de configurer le dimensionnement à l'aide d'un CloudFormation modèle. Pour de plus amples informations, veuillez consulter Configurez les ressources Application Auto Scaling à l'aide de AWS CloudFormation.

Pour vous connecter par programmation à un Service AWS, vous utilisez un point de terminaison. Pour plus d'informations sur les points de terminaison pour les appels à Application Auto Scaling, consultez Application Auto Scaling endpoints and quotas in the Références générales AWS .

Concepts d'Application Auto Scaling

Cette rubrique explique les concepts clés qui vous permettront de découvrir Application Auto Scaling et de l'utiliser.

Cible évolutive

Une entité que vous créez pour spécifier la ressource que vous souhaitez mettre à l'échelle. Chaque cible évolutive est identifiée de manière unique par un espace de noms de service, un ID de ressource et une dimension évolutive, qui représente une certaine dimension de capacité du service sous-jacent. Par exemple, Amazon ECS service prend en charge la scalabilité automatique de son nombre de tâches, une table DynamoDB prend en charge la scalabilité automatique de la capacité de lecture et d'écriture de la table et de ses index secondaires globaux, et un cluster Aurora prend en charge la mise à l'échelle de son nombre de réplicas.

> **Tip**
>
> Chaque cible évolutive possède également une capacité minimale et maximale. Les politiques de mise à l'échelle ne seront jamais supérieures ou inférieures à la plage

minimum-maximum. Vous pouvez apporter out-of-band des modifications directement à la ressource sous-jacente qui se situent en dehors de cette plage, ce qu'Application Auto Scaling ne connaît pas. Cependant, chaque fois qu'une politique de mise à l'échelle est invoquée ou que l'API `RegisterScalableTarget` est appelée, Application Auto Scaling récupère la capacité actuelle et la compare à la capacité minimale et maximale. Si elle se situe en dehors de la plage minimum-maximum, la capacité est mise à jour pour se conformer aux minimums et maximums définis.

Mise à l'échelle horizontale

Lorsque l'Application Auto Scaling réduit automatiquement la capacité d'une cible évolutive, la cible évolutive est mise à l'échelle horizontale. Lorsque des politiques de mise à l'échelle sont définies, elles ne peuvent pas mettre à l'échelle horizontale la cible capable d'être mise à l'échelle en dessous de sa capacité minimale.

Monter en puissance

Lorsque l'Application Auto Scaling augmente automatiquement la capacité d'une cible évolutive, la cible évolutive subit une montée en puissance. Lorsque des politiques de mise à l'échelle sont définies, elles ne peuvent pas faire monter en puissance la cible capable d'être mise à l'échelle au dessus de sa capacité maximale.

Politique de mise à l'échelle

Une politique de dimensionnement indique à Application Auto Scaling de suivre une CloudWatch métrique spécifique. Ensuite, elle détermine l'action de mise à l'échelle à prendre lorsque la métrique est supérieure ou inférieure à une certaine valeur de seuil. Par exemple, vous pourriez vouloir augmenter la capacité de votre cluster si l'utilisation du CPU commence à augmenter, et la diminuer lorsqu'elle retombe.

Les métriques utilisées pour le dimensionnement automatique sont publiées par le service cible, mais vous pouvez également publier votre propre métrique sur, CloudWatch puis l'utiliser avec une politique de dimensionnement.

Un temps de stabilisation entre les activités de mise à l'échelle permet à la ressource de se stabiliser avant le début d'une autre activité de mise à l'échelle. Application Auto Scaling continue à évaluer les métriques pendant le temps de stabilisation. À la fin du temps de stabilisation, la politique de mise à l'échelle lance une autre activité de mise à l'échelle si nécessaire. Pendant le

temps de stabilisation, si une plus grande augmentation est nécessaire en fonction de la valeur métrique actuelle, la politique de mise à l'échelle effectue une augmentation immédiatement.

Action planifiée

Les actions planifiées mettent automatiquement à l'échelle les ressources à une date et une heure spécifiques. Elles fonctionnent en modifiant la capacité minimale et maximale d'une cible évolutive, et peuvent donc être utilisées pour effectuer une diminution ou une augmentation selon une planification en définissant une capacité minimale élevée ou une capacité maximale faible. Par exemple, vous pouvez utiliser des actions planifiées pour mettre à l'échelle une application qui ne consomme pas de ressources le week-end en diminuant la capacité le vendredi et en l'augmentant le lundi suivant.

Vous pouvez également utiliser des actions planifiées pour optimiser les valeurs minimales et maximales au fil du temps afin de vous adapter à des situations où l'on s'attend à un trafic supérieur à la normale, par exemple des campagnes de marketing ou des fluctuations saisonnières. Cela peut vous aider à améliorer les performances lorsque vous devez augmenter la capacité pour faire face à l'augmentation de l'utilisation, et à réduire les coûts lorsque vous utilisez moins de ressources.

En savoir plus

Services AWS que vous pouvez utiliser avec Application Auto Scaling : Cette section vous présente les services que vous pouvez mettre à l'échelle et vous aide à configurer la scalabilité automatique en enregistrant une cible évolutive. Elle décrit également chacun des rôles liés au service IAM que Application Auto Scaling crée pour accéder aux ressources du service cible.

Politique de suivi des cibles et d'échelonnement pour Application Auto Scaling : Les politiques de suivi des cibles et d'échelonnement sont l'une des principales caractéristiques d'Application Auto Scaling. Découvrez comment les politiques de suivi des cibles ajustent automatiquement la capacité souhaitée pour maintenir l'utilisation à un niveau constant en fonction des valeurs de métriques et d'objectifs que vous avez configurées. Par exemple, vous pouvez configurer le suivi de cible pour maintenir l'utilisation moyenne du CPU de votre parc d'instances Spot à 50 %. Application Auto Scaling lance ou arrête ensuite les EC2 instances selon les besoins afin de maintenir l'utilisation agrégée du processeur sur tous les serveurs à 50 %.

Services AWS que vous pouvez utiliser avec Application Auto Scaling

Application Auto Scaling s'intègre à d'autres AWS services afin que vous puissiez ajouter des fonctionnalités de dimensionnement pour répondre à la demande de votre application. La mise à l'échelle est une fonction facultative du service qui est désactivée par défaut dans presque tous les cas.

Le tableau suivant répertorie les AWS services que vous pouvez utiliser avec Application Auto Scaling, y compris des informations sur les méthodes prises en charge pour configurer le dimensionnement automatique. Vous pouvez également utiliser Application Auto Scaling avec des ressources personnalisées.

- Accès par la console : Vous pouvez configurer un service AWS compatible pour lancer la scalabilité automatique en configurant une politique de mise à l'échelle dans la console du service cible.
- Accès par la CLI : Vous pouvez configurer un service AWS compatible pour lancer la scalabilité automatique à l'aide de l' AWS CLI CLI.
- Accès au SDK : vous pouvez configurer un AWS service compatible pour démarrer le dimensionnement automatique à l'aide du AWS SDKs.
- CloudFormation accès — Vous pouvez configurer un AWS service compatible pour démarrer le dimensionnement automatique à l'aide d'un modèle de CloudFormation pile. Pour de plus amples informations, veuillez consulter Configurez les ressources Application Auto Scaling à l'aide de AWS CloudFormation.

AWS service	Accès à la console[1]	Accès par la CLI	Accès par le kit SDK	CloudFormation accès
WorkSpaces Applications	Oui	Oui	Oui	Oui

AWS service	Accès à la console[1]	Accès par la CLI	Accès par le kit SDK	CloudFormation accès
Aurora	 Oui	 Oui	 Oui	 Oui
Amazon Comprehend	 Nor	 Oui	 Oui	 Oui
Amazon DynamoDB	 Oui	 Oui	 Oui	 Oui
Amazon ECS	 Oui	 Oui	 Oui	 Oui
Amazon ElastiCache	 Oui	 Oui	 Oui	 Oui
Amazon EMR	 Oui	 Oui	 Oui	 Oui
Amazon Keyspaces	 Oui	 Oui	 Oui	 Oui
Lambda	 Nor	 Oui	 Oui	 Oui

AWS service	Accès à la console[1]	Accès par la CLI	Accès par le kit SDK	CloudFormation accès
Amazon MSK	 Oui	 Oui	 Oui	 Oui
Amazon Neptune	 Nor	 Oui	 Oui	 Oui
SageMaker AI	 Oui	 Oui	 Oui	 Oui
Parc d'instances Spot	 Oui	 Oui	 Oui	 Oui
WorkSpaces	 Oui	 Oui	 Oui	 Oui
Ressources personnalisées	 Nor	 Oui	 Oui	 Oui

[1] Accès à la console pour configurer les politiques de dimensionnement. La plupart des services ne prennent pas en charge la configuration du dimensionnement planifié depuis la console. Actuellement, seuls Amazon WorkSpaces Applications et Spot Fleet fournissent un accès à la console pour un dimensionnement planifié. ElastiCache

Amazon WorkSpaces Applications et Application Auto Scaling

Vous pouvez dimensionner WorkSpaces les flottes d'applications à l'aide des politiques de dimensionnement du suivi des cibles, des politiques de dimensionnement par étapes et du dimensionnement planifié.

Utilisez les informations suivantes pour vous aider à intégrer les WorkSpaces applications à Application Auto Scaling.

Rôle lié à un service créé pour les applications WorkSpaces

Le rôle lié au service suivant est automatiquement créé dans votre ordinateur Compte AWS lors de l'enregistrement des ressources d' WorkSpaces applications en tant que cibles évolutives avec Application Auto Scaling. Ce rôle permet à Application Auto Scaling d'effectuer des opérations prises en charge au sein de votre compte. Pour de plus amples informations, veuillez consulter Rôles liés à un service pour Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_AppStreamFleet`

Principal du service utilisé par le rôle lié à un service

Le rôle lié à un service dans la section précédente ne peut être assumé que par le principal du service autorisé par les relations d'approbation définies pour le rôle. Le rôle lié à un service utilisé par Application Auto Scaling donne l'accès au principal du service suivant :

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

Enregistrement de flottes WorkSpaces d'applications en tant que cibles évolutives avec Application Auto Scaling

Application Auto Scaling nécessite une cible évolutive avant de pouvoir créer des politiques de dimensionnement ou des actions planifiées pour un parc d' WorkSpaces applications. Une cible évolutive est une ressource qu'Application Auto Scaling peut augmenter et diminuer. Les cibles évolutives sont identifiées de manière unique par la combinaison de l'ID de ressource, de la dimension évolutive et de l'espace de noms.

Si vous configurez le dimensionnement automatique à l'aide de la console WorkSpaces WorkSpaces Applications, Applications enregistre automatiquement une cible évolutive pour vous.

Si vous souhaitez configurer le dimensionnement automatique à l'aide de la AWS CLI ou de l'une des options suivantes AWS SDKs, vous pouvez utiliser les options suivantes :

- AWS CLI:

  Appelez la register-scalable-targetcommande pour un parc d' WorkSpaces applications. L'exemple suivant enregistre la capacité souhaitée d'un parc appelé `sample-fleet`, avec une capacité minimale d'une instance de parc et une capacité maximale de cinq instances de parc.

  ```
  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
  ```

  En cas de réussite, cette commande renvoie l'ARN de la cible évolutive.

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

- AWS SDK :

  Appelez l'opération RegisterScalableTarget et fournissez `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` et `MaxCapacity` comme paramètres.

Ressources connexes

Pour plus d'informations, consultez Fleet Auto Scaling for Amazon WorkSpaces Applications dans le Amazon WorkSpaces Applications Administration Guide.

Amazon Aurora et Application Auto Scaling

Vous pouvez mettre à l'échelle les clusters de bases de données Aurora à l'aide de politiques de suivi des cibles et d'échelonnement et de mise à l'échelle planifiée.

Utilisez les informations suivantes pour vous aider à intégrer Aurora avec Application Auto Scaling.

Rôle lié un service créé pour Aurora

Le rôle lié au service suivant est automatiquement créé dans votre ordinateur Compte AWS lors de l'enregistrement des ressources Aurora en tant que cibles évolutives avec Application Auto Scaling. Ce rôle permet à Application Auto Scaling d'effectuer des opérations prises en charge au sein de votre compte. Pour de plus amples informations, veuillez consulter Rôles liés à un service pour Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_RDSCluster`

Principal du service utilisé par le rôle lié à un service

Le rôle lié à un service dans la section précédente ne peut être assumé que par le principal du service autorisé par les relations d'approbation définies pour le rôle. Le rôle lié à un service utilisé par Application Auto Scaling donne l'accès au principal du service suivant :

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

Enregistrementdes clusters de bases de données Aurora en tant que cibles évolutives avec Application Auto Scaling

Application Auto Scaling requiert une cible évolutive avant de pouvoir créer des politiques de mise à l'échelle ou des actions planifiées pour un cluster Aurora. Une cible évolutive est une ressource qu'Application Auto Scaling peut augmenter et diminuer. Les cibles évolutives sont identifiées de manière unique par la combinaison de l'ID de ressource, de la dimension évolutive et de l'espace de noms.

Si vous configurez la scalabilité automatique à l'aide de la console Aurora, Aurora enregistre automatiquement une cible évolutive pour vous.

Si vous souhaitez configurer le dimensionnement automatique à l'aide de la AWS CLI ou de l'une des options suivantes AWS SDKs, vous pouvez utiliser les options suivantes :

- AWS CLI:

  Appelez la commande register-scalable-target pour un cluster Aurora. L'exemple suivant enregistre le nombre de réplicas Aurora dans un cluster appelé `my-db-cluster`, avec une capacité minimale d'un réplica Aurora et une capacité maximale de huit réplicas 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
```

En cas de réussite, cette commande renvoie l'ARN de la cible évolutive.

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

- AWS SDK :

  Appelez l'opération RegisterScalableTarget et fournissez `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` et `MaxCapacity` comme paramètres.

Ressources connexes

Pour plus d'informations, consultez Amazon Aurora Auto Scaling with Aurora Replicas dans le guide de l'utilisateur Amazon RDS pour Aurora.

Amazon Comprehend et Application Auto Scaling

Vous pouvez mettre à l'échelle les points de terminaison de classification de documents et de reconnaissance d'entités Amazon Comprehend à l'aide de politiques de suivi des cibles et d'échelonnement et de mise à l'échelle planifiée.

Utilisez les informations suivantes pour vous aider à intégrer Amazon Comprehend avec Application Auto Scaling.

Rôle lié à un service créé pour Amazon Comprehend

Le rôle lié au service suivant est automatiquement créé dans votre compte Compte AWS lors de l'enregistrement des ressources Amazon Comprehend en tant que cibles évolutives avec Application Auto Scaling. Ce rôle permet à Application Auto Scaling d'effectuer des opérations prises en charge

au sein de votre compte. Pour de plus amples informations, veuillez consulter Rôles liés à un service pour Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_ComprehendEndpoint`

Principal du service utilisé par le rôle lié à un service

Le rôle lié à un service dans la section précédente ne peut être assumé que par le principal du service autorisé par les relations d'approbation définies pour le rôle. Le rôle lié à un service utilisé par Application Auto Scaling donne l'accès au principal du service suivant :

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

Enregistrement des ressources Amazon Comprehend en tant que cibles évolutives avec Application Auto Scaling

Application Auto Scaling nécessite une cible évolutive avant que vous puissiez créer des politiques de mise à l'échelle ou des actions planifiées pour un point de terminaison de classification de documents ou de reconnaissance d'entités Amazon Comprehend. Une cible évolutive est une ressource qu'Application Auto Scaling peut augmenter et diminuer. Les cibles évolutives sont identifiées de manière unique par la combinaison de l'ID de ressource, de la dimension évolutive et de l'espace de noms.

Pour configurer le dimensionnement automatique à l'aide de la AWS CLI ou de l'une des options suivantes AWS SDKs, vous pouvez utiliser les options suivantes :

- AWS CLI:

  Appelez la commande register-scalable-target pour un point de terminaison de classification de documents. L'exemple suivant enregistre le nombre souhaité d'unités d'inférence à utiliser par le modèle pour un point de terminaison de classification de documents en utilisant l'ARN du point de terminaison, avec une capacité minimale d'une unité d'inférence et une capacité maximale de trois unités d'inférence.

```
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
```

En cas de réussite, cette commande renvoie l'ARN de la cible évolutive.

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

Appelez la commande register-scalable-target pour un point de terminaison de reconnaissance d'entités. L'exemple suivant enregistre le nombre souhaité d'unités d'inférence à utiliser par le modèle pour une reconnaissance d'entités en utilisant l'ARN du point de terminaison, avec une capacité minimale d'une unité d'inférence et une capacité maximale de trois unités d'inférence.

```
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
```

En cas de réussite, cette commande renvoie l'ARN de la cible évolutive.

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

- AWS SDK :

  Appelez l'opération RegisterScalableTarget et fournissez `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` et `MaxCapacity` comme paramètres.

Ressources connexes

Pour plus d'informations, consultez la section Mise à l'échelle automatique avec les points de terminaison dans le manuel Amazon Comprehend Developer Guide.

Amazon DynamoDB et Application Auto Scaling

Vous pouvez mettre à l'échelle les tables et les index secondaires globaux de DynamoDB à l'aide de politiques de suivi des cibles et d'échelonnement et de mise à l'échelle planifiée.

Utilisez les informations suivantes pour vous aider à intégrer DynamoDB avec Application Auto Scaling.

Rôle lié un service créé pour DynamoDB

Le rôle lié au service suivant est automatiquement créé dans votre ordinateur Compte AWS lorsque vous enregistrez des ressources DynamoDB en tant que cibles évolutives avec Application Auto Scaling. Ce rôle permet à Application Auto Scaling d'effectuer des opérations prises en charge au sein de votre compte. Pour de plus amples informations, veuillez consulter Rôles liés à un service pour Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_DynamoDBTable`

Principal du service utilisé par le rôle lié à un service

Le rôle lié à un service dans la section précédente ne peut être assumé que par le principal du service autorisé par les relations d'approbation définies pour le rôle. Le rôle lié à un service utilisé par Application Auto Scaling donne l'accès au principal du service suivant :

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

Enregistrement des ressources DynamoDB en tant que cibles évolutives avec Application Auto Scaling

Application Auto Scaling requiert une cible évolutive avant de pouvoir créer des politiques de mise à l'échelle ou des actions planifiées pour une table ou un index secondaire global DynamoDB. Une cible évolutive est une ressource qu'Application Auto Scaling peut augmenter et diminuer. Les

cibles évolutives sont identifiées de manière unique par la combinaison de l'ID de ressource, de la dimension évolutive et de l'espace de noms.

Si vous configurez la scalabilité automatique à l'aide de la console DynamoDB, DynamoDB enregistre automatiquement une cible évolutive pour vous.

Si vous souhaitez configurer le dimensionnement automatique à l'aide de la AWS CLI ou de l'une des options suivantes AWS SDKs, vous pouvez utiliser les options suivantes :

- AWS CLI:

  Appelez la register-scalable-targetcommande concernant la capacité d'écriture d'une table. L'exemple suivant enregistre la capacité d'écriture allouée d'une table appeléemy-table, avec une capacité minimale de cinq unités de capacité d'écriture et une capacité maximale de 10 unités de capacité d'écriture :

  ```
  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
  ```

  En cas de succès, cette commande renvoie l'ARN de la cible évolutive :

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

  Appelez la register-scalable-targetcommande pour connaître la capacité de lecture d'une table. L'exemple suivant enregistre la capacité de lecture allouée d'une table appeléemy-table, avec une capacité minimale de cinq unités de capacité de lecture et une capacité maximale de 10 unités de lecture :

  ```
  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
```

En cas de succès, cette commande renvoie l'ARN de la cible évolutive :

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

Appelez la register-scalable-targetcommande pour connaître la capacité d'écriture d'un index secondaire global. L'exemple suivant enregistre la capacité d'écriture allouée d'un index secondaire global appelé`my-table-index`, avec une capacité minimale de cinq unités de capacité d'écriture et une capacité maximale de 10 unités de capacité d'écriture :

```
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
```

En cas de succès, cette commande renvoie l'ARN de la cible évolutive :

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

Appelez la register-scalable-targetcommande pour connaître la capacité de lecture d'un index secondaire global. L'exemple suivant enregistre la capacité de lecture allouée d'un index secondaire global appelé`my-table-index`, avec une capacité minimale de cinq unités de capacité de lecture et une capacité maximale de 10 unités de capacité de lecture :

```
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
```

En cas de succès, cette commande renvoie l'ARN de la cible évolutive :

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

- AWS SDK :

  Appelez l'opération RegisterScalableTarget et fournissez `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` et `MaxCapacity` comme paramètres.

Ressources connexes

Si vous débutez avec Application Auto Scaling, vous trouverez des informations supplémentaires utiles sur le dimensionnement de vos ressources DynamoDB dans la documentation suivante :

- Gestion de la capacité de débit avec DynamoDB Auto Scaling dans le Guide du développeur Amazon DynamoDB
- Évaluez les paramètres de mise à l'échelle automatique de votre table dans le manuel du développeur Amazon DynamoDB
- Comment l'utiliser CloudFormation pour configurer le dimensionnement automatique pour les tables et les index DynamoDB sur le blog AWS

Amazon ECS et Application Auto Scaling

Vous pouvez dimensionner les services ECS à l'aide des politiques de dimensionnement du suivi des cibles, des politiques de dimensionnement prédictif, des politiques de dimensionnement par étapes et du dimensionnement planifié.

Utilisez les informations suivantes pour vous aider à intégrer Amazon ECS avec Application Auto Scaling.

Rôle lié à un service créé pour Amazon ECS

Le rôle lié au service suivant est automatiquement créé dans votre compte Compte AWS lors de l'enregistrement des ressources Amazon ECS en tant que cibles évolutives avec Application Auto Scaling. Ce rôle permet à Application Auto Scaling d'effectuer des opérations prises en charge au

sein de votre compte. Pour de plus amples informations, veuillez consulter Rôles liés à un service pour Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_ECSService`

Principal du service utilisé par le rôle lié à un service

Le rôle lié à un service dans la section précédente ne peut être assumé que par le principal du service autorisé par les relations d'approbation définies pour le rôle. Le rôle lié à un service utilisé par Application Auto Scaling donne l'accès au principal du service suivant :

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

Enregistrement des services ECS en tant que cibles évolutives avec Application Auto Scaling

Application Auto Scaling requiert une cible évolutive avant de pouvoir créer des politiques de mise à l'échelle ou des actions planifiées pour un Amazon ECS service. Une cible évolutive est une ressource qu'Application Auto Scaling peut augmenter et diminuer. Les cibles évolutives sont identifiées de manière unique par la combinaison de l'ID de ressource, de la dimension évolutive et de l'espace de noms.

Si vous configurez la scalabilité automatique à l'aide de la console Amazon ECS, Amazon ECS enregistre automatiquement une cible évolutive pour vous.

Si vous souhaitez configurer le dimensionnement automatique à l'aide de la AWS CLI ou de l'une des options suivantes AWS SDKs, vous pouvez utiliser les options suivantes :

- AWS CLI:

  Appelez la commande register-scalable-target pour un Amazon ECS service. L'exemple suivant enregistre une cible évolutive pour un service appelé `sample-app-service`, exécuté sur le cluster `default`, avec un nombre de tâches minimum d'une tâche et un nombre de tâches maximum de 10 tâches.

```
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
```

En cas de réussite, cette commande renvoie l'ARN de la cible évolutive.

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

- AWS SDK :

  Appelez l'opération RegisterScalableTarget et fournissez `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` et `MaxCapacity` comme paramètres.

Ressources connexes

Si vous débutez avec Application Auto Scaling, vous trouverez des informations supplémentaires utiles sur le dimensionnement de vos ressources Amazon ECS dans la documentation suivante :

- Mise à l'échelle automatique des services dans le guide du développeur Amazon Elastic Container Service
- Optimisez le dimensionnement automatique du service Amazon ECS dans le manuel du développeur Amazon Elastic Container Service

> **Note**
>
> Pour obtenir des instructions sur la suspension des processus de scale-out pendant les déploiements d'Amazon ECS, consultez la documentation suivante : Mise à l'échelle et déploiements automatiques des services dans le guide du développeur Amazon Elastic Container Service

ElastiCache et Application Auto Scaling

Vous pouvez redimensionner horizontalement les groupes de ElastiCache réplication Amazon (Redis OSS et Valkey) et les clusters conçus par Memcached à l'aide des politiques de dimensionnement du suivi des cibles et du dimensionnement planifié.

Pour intégrer ElastiCache Application Auto Scaling, utilisez les informations suivantes.

Rôle lié à un service créé pour ElastiCache

Le rôle lié au service suivant est automatiquement créé dans votre ordinateur Compte AWS lorsque vous enregistrez des ElastiCache ressources en tant que cibles évolutives avec Application Auto Scaling. Ce rôle permet à Application Auto Scaling d'effectuer des opérations prises en charge au sein de votre compte. Pour de plus amples informations, veuillez consulter Rôles liés à un service pour Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG`

Principal du service utilisé par le rôle lié à un service

Le rôle lié à un service dans la section précédente ne peut être assumé que par le principal du service autorisé par les relations d'approbation définies pour le rôle. Le rôle lié à un service utilisé par Application Auto Scaling donne l'accès au principal du service suivant :

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

Enregistrement ElastiCache des ressources en tant que cibles évolutives avec Application Auto Scaling

Application Auto Scaling nécessite une cible évolutive avant de pouvoir créer des politiques de dimensionnement ou des actions planifiées pour un groupe, un cluster ou un nœud de ElastiCache réplication. Une cible évolutive est une ressource qu'Application Auto Scaling peut augmenter et diminuer. Les cibles évolutives sont identifiées de manière unique par la combinaison de l'ID de ressource, de la dimension évolutive et de l'espace de noms.

Si vous configurez le dimensionnement automatique à l'aide de la ElastiCache console, une cible évolutive est ElastiCache automatiquement enregistrée pour vous.

Si vous souhaitez configurer le dimensionnement automatique à l'aide de la AWS CLI ou de l'une des options suivantes AWS SDKs, vous pouvez utiliser les options suivantes :

- AWS CLI:

  Appelez la register-scalable-targetcommande correspondant à un groupe ElastiCache de réplication. L'exemple suivant enregistre le nombre souhaité de groupes de nœuds pour un groupe

de réplication appelé `mycluster1`, avec une capacité minimale de un et une capacité maximale de cinq.

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

En cas de réussite, cette commande renvoie l'ARN de la cible évolutive.

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

L'exemple suivant enregistre le nombre souhaité de répliques par groupe de nœuds pour un groupe de réplication appelé`mycluster2`, avec une capacité minimale de un et une capacité maximale de cinq.

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

En cas de réussite, cette commande renvoie l'ARN de la cible évolutive.

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

L'exemple suivant enregistre le nombre de nœuds souhaité pour un cluster appelé`mynode1`, avec une capacité minimale de 20 et une capacité maximale de 50.

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

```
    --scalable-dimension elasticache:cache-cluster:Nodes \
    --resource-id cache-cluster/mynode1 \
    --min-capacity 20 \
    --max-capacity 50
```

En cas de réussite, cette commande renvoie l'ARN de la cible évolutive.

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

- AWS SDK :

  Appelez l'opération RegisterScalableTarget et fournissez `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` et `MaxCapacity` comme paramètres.

Ressources connexes

Pour plus d'informations, consultez les sections Clusters Auto Scaling Valkey et Redis OSS et Scaling clusters for Memcached dans le guide de l'utilisateur Amazon. ElastiCache

Amazon Keyspaces (for Apache Cassandra) et Application Auto Scaling

Vous pouvez mettre à l'échelle les tables Amazon Keyspaces à l'aide de politiques de suivi des cibles et d'échelonnement et de mise à l'échelle planifiée.

Utilisez les informations suivantes pour vous aider à intégrer Amazon Keyspaces avec Application Auto Scaling.

Rôle lié à un service créé pour Amazon Keyspaces

Le rôle lié au service suivant est automatiquement créé dans votre compte Compte AWS lors de l'enregistrement des ressources Amazon Keyspaces en tant que cibles évolutives avec Application Auto Scaling. Ce rôle permet à Application Auto Scaling d'effectuer des opérations prises en charge au sein de votre compte. Pour de plus amples informations, veuillez consulter Rôles liés à un service pour Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_CassandraTable`

Principal du service utilisé par le rôle lié à un service

Le rôle lié à un service dans la section précédente ne peut être assumé que par le principal du service autorisé par les relations d'approbation définies pour le rôle. Le rôle lié à un service utilisé par Application Auto Scaling donne l'accès au principal du service suivant :

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

Enregistrement des tables Amazon Keyspaces en tant que cibles évolutives avec Application Auto Scaling

Application Auto Scaling requiert une cible évolutive avant de pouvoir créer des politiques de mise à l'échelle ou des actions planifiées pour une table Amazon Keyspaces. Une cible évolutive est une ressource qu'Application Auto Scaling peut augmenter et diminuer. Les cibles évolutives sont identifiées de manière unique par la combinaison de l'ID de ressource, de la dimension évolutive et de l'espace de noms.

Si vous configurez la scalabilité automatique à l'aide de la console Amazon Keyspaces, Amazon Keyspaces enregistre automatiquement une cible évolutive pour vous.

Si vous souhaitez configurer le dimensionnement automatique à l'aide de la AWS CLI ou de l'une des options suivantes AWS SDKs, vous pouvez utiliser les options suivantes :

- AWS CLI:

  Appelez la commande register-scalable-target pour une table Amazon Keyspaces. L'exemple suivant enregistre la capacité d'écriture allouée d'une table appelée `mytable`, avec une capacité minimale de cinq unités de capacité d'écriture et une capacité maximale de 10 unités de capacité d'écriture.

```
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
```

En cas de réussite, cette commande renvoie l'ARN de la cible évolutive.

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

L'exemple suivant enregistre la capacité de lecture allouée d'une table appelée `mytable`, avec une capacité minimale de cinq unités de capacité de lecture et une capacité maximale de 10 unités de capacité de lecture.

```
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
```

En cas de réussite, cette commande renvoie l'ARN de la cible évolutive.

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

- AWS SDK :

  Appelez l'opération RegisterScalableTarget et fournissez `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` et `MaxCapacity` comme paramètres.

Ressources connexes

Pour plus d'informations, consultez la section Gérer automatiquement la capacité de débit avec le dimensionnement automatique d'Amazon Keyspaces dans le manuel du développeur Amazon Keyspaces.

AWS Lambda et Application Auto Scaling

Vous pouvez dimensionner la AWS Lambda simultanéité provisionnée à l'aide des politiques de dimensionnement du suivi des cibles et du dimensionnement planifié.

Utilisez les informations suivantes pour vous aider à intégrer Lambda avec Application Auto Scaling.

Rôle lié un service créé pour Lambda

Le rôle lié au service suivant est automatiquement créé dans votre ordinateur Compte AWS lorsque vous enregistrez des ressources Lambda en tant que cibles évolutives avec Application Auto Scaling. Ce rôle permet à Application Auto Scaling d'effectuer des opérations prises en charge au sein de votre compte. Pour de plus amples informations, veuillez consulter Rôles liés à un service pour Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_LambdaConcurrency`

Principal du service utilisé par le rôle lié à un service

Le rôle lié à un service dans la section précédente ne peut être assumé que par le principal du service autorisé par les relations d'approbation définies pour le rôle. Le rôle lié à un service utilisé par Application Auto Scaling donne l'accès au principal du service suivant :

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

Enregistrement de fonctions Lambda en tant que cibles évolutives avec Application Auto Scaling

Application Auto Scaling requiert une cible évolutive avant de pouvoir créer des politiques de mise à l'échelle ou des actions planifiées pour une fonction Lambda. Une cible évolutive est une ressource qu'Application Auto Scaling peut augmenter et diminuer. Les cibles évolutives sont identifiées de manière unique par la combinaison de l'ID de ressource, de la dimension évolutive et de l'espace de noms.

Pour configurer le dimensionnement automatique à l'aide de la AWS CLI ou de l'une des options suivantes AWS SDKs, vous pouvez utiliser les options suivantes :

- AWS CLI:

Appelez la commande register-scalable-target pour une fonction Lambda. L'exemple suivant enregistre la simultanéité allouée pour un alias appelé `BLUE` pour une fonction appelée `my-function`, avec une capacité minimale de 0 et une capacité maximale 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
```

En cas de réussite, cette commande renvoie l'ARN de la cible évolutive.

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

- AWS SDK :

  Appelez l'opération RegisterScalableTarget et fournissez `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` et `MaxCapacity` comme paramètres.

Ressources connexes

Si vous débutez avec Application Auto Scaling, vous trouverez des informations supplémentaires utiles sur le dimensionnement de vos fonctions Lambda dans la documentation suivante :

- Configuration de la simultanéité provisionnée dans le guide du développeur AWS Lambda
- Planification de la simultanéité provisionnée par Lambda en cas de pic d'utilisation récurrent sur le blog AWS

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

Vous pouvez monter en puissance le stockage du cluster Amazon MSK à l'aide de politiques de suivi des objectifs et d'échelonnement. Mise à l'échelle horizontale par la politique de suivi de cible est désactivée.

Utilisez les informations suivantes pour vous aider à intégrer Amazon MSK avec Application Auto Scaling.

Rôle lié à un service créé pour Amazon MSK

Le rôle lié au service suivant est automatiquement créé dans votre compte Compte AWS lors de l'enregistrement des ressources Amazon MSK en tant que cibles évolutives avec Application Auto Scaling. Ce rôle permet à Application Auto Scaling d'effectuer des opérations prises en charge au sein de votre compte. Pour de plus amples informations, veuillez consulter Rôles liés à un service pour Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_KafkaCluster`

Principal du service utilisé par le rôle lié à un service

Le rôle lié à un service dans la section précédente ne peut être assumé que par le principal du service autorisé par les relations d'approbation définies pour le rôle. Le rôle lié à un service utilisé par Application Auto Scaling donne l'accès au principal du service suivant :

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

Enregistrement du stockage du cluster Amazon MSK en tant que cibles évolutives avec Application Auto Scaling

Application Auto Scaling nécessite une cible évolutive avant que vous puissiez créer une politique de mise à l'échelle pour la taille du volume de stockage par agent d'un cluster Amazon MSK. Une cible évolutive est une ressource qu'Application Auto Scaling peut mettre à l'échelle. Les cibles évolutives sont identifiées de manière unique par la combinaison de l'ID de ressource, de la dimension évolutive et de l'espace de noms.

Si vous configurez la scalabilité automatique à l'aide de la console Amazon MSK, Amazon MSK enregistre automatiquement une cible évolutive pour vous.

Si vous souhaitez configurer le dimensionnement automatique à l'aide de la AWS CLI ou de l'une des options suivantes AWS SDKs, vous pouvez utiliser les options suivantes :

- AWS CLI:

  Appelez la commande register-scalable-target pour un cluster Amazon MSK. L'exemple suivant enregistre la taille du volume de stockage par agent d'un cluster Amazon MSK, avec une capacité minimale de 100 GiB et une capacité maximale 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
  ```

  En cas de réussite, cette commande renvoie l'ARN de la cible évolutive.

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

- AWS SDK :

  Appelez l'opération RegisterScalableTarget et fournissez `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` et `MaxCapacity` comme paramètres.

> **Note**
>
> Lorsqu'un cluster Amazon MSK est la cible évolutive, la diminution de charge est désactivée et ne peut pas être activée.

Ressources connexes

Pour plus d'informations, consultez la section Mise à l'échelle automatique pour les clusters Amazon MSK dans le guide du développeur Amazon Managed Streaming for Apache Kafka.

Amazon Neptune et Application Auto Scaling

Vous pouvez mettre à l'échelle les clusters Neptune à l'aide de politiques de suivi des objectifs et d'échelonnement et de mise à l'échelle planifiée.

Utilisez les informations suivantes pour vous aider à intégrer Neptune avec Application Auto Scaling.

Rôle lié un service créé pour Neptune

Le rôle lié au service suivant est automatiquement créé dans votre ordinateur Compte AWS lors de l'enregistrement des ressources Neptune en tant que cibles évolutives avec Application Auto Scaling. Ce rôle permet à Application Auto Scaling d'effectuer des opérations prises en charge au sein de votre compte. Pour de plus amples informations, veuillez consulter Rôles liés à un service pour Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_NeptuneCluster`

Principal du service utilisé par le rôle lié à un service

Le rôle lié à un service dans la section précédente ne peut être assumé que par le principal du service autorisé par les relations d'approbation définies pour le rôle. Le rôle lié à un service utilisé par Application Auto Scaling donne l'accès au principal du service suivant :

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

Enregistrement des clusters de bases de données Neptune en tant que cibles évolutives avec Application Auto Scaling

Application Auto Scaling requiert une cible évolutive avant de pouvoir créer des politiques de mise à l'échelle ou des actions planifiées pour un cluster Neptune. Une cible évolutive est une ressource qu'Application Auto Scaling peut augmenter et diminuer. Les cibles évolutives sont identifiées de manière unique par la combinaison de l'ID de ressource, de la dimension évolutive et de l'espace de noms.

Pour configurer le dimensionnement automatique à l'aide de la AWS CLI ou de l'une des options suivantes AWS SDKs, vous pouvez utiliser les options suivantes :

- AWS CLI:

  Appelez la register-scalable-targetcommande correspondant à un cluster Neptune. L'exemple suivant enregistre la capacité souhaitée d'un cluster appelé `mycluster`, avec une capacité minimale de un et une capacité maximale de huit.

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

  En cas de réussite, cette commande renvoie l'ARN de la cible évolutive.

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

- AWS SDK :

  Appelez l'opération RegisterScalableTarget et fournissez `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` et `MaxCapacity` comme paramètres.

Ressources connexes

Pour plus d'informations, consultez la section Dimensionnement automatique du nombre de répliques dans un cluster de base de données Amazon Neptune dans le guide de l'utilisateur de Neptune.

Amazon SageMaker AI et Application Auto Scaling

Vous pouvez dimensionner les variantes des terminaux SageMaker AI, la simultanéité provisionnée pour les points de terminaison sans serveur et les composants d'inférence à l'aide de politiques de dimensionnement du suivi des cibles, de politiques de dimensionnement par étapes et de dimensionnement planifié.

Utilisez les informations suivantes pour vous aider à intégrer l' SageMaker IA à Application Auto Scaling.

Rôle lié à un service créé pour l'IA SageMaker

Le rôle lié au service suivant est automatiquement créé dans votre ordinateur Compte AWS lorsque vous enregistrez des ressources d' SageMaker IA en tant que cibles évolutives avec Application Auto Scaling. Ce rôle permet à Application Auto Scaling d'effectuer des opérations prises en charge au sein de votre compte. Pour de plus amples informations, veuillez consulter Rôles liés à un service pour Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_SageMakerEndpoint`

Principal du service utilisé par le rôle lié à un service

Le rôle lié à un service dans la section précédente ne peut être assumé que par le principal du service autorisé par les relations d'approbation définies pour le rôle. Le rôle lié à un service utilisé par Application Auto Scaling donne l'accès au principal du service suivant :

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

Enregistrement de variantes de terminaux SageMaker AI en tant que cibles évolutives avec Application Auto Scaling

Application Auto Scaling nécessite une cible évolutive avant de pouvoir créer des politiques de dimensionnement ou des actions planifiées pour un modèle d' SageMaker IA (variante). Une cible évolutive est une ressource qu'Application Auto Scaling peut augmenter et diminuer. Les cibles évolutives sont identifiées de manière unique par la combinaison de l'ID de ressource, de la dimension évolutive et de l'espace de noms.

Si vous configurez le dimensionnement automatique à l'aide de la console SageMaker AI, l' SageMaker IA enregistre automatiquement une cible évolutive pour vous.

Si vous souhaitez configurer le dimensionnement automatique à l'aide de la AWS CLI ou de l'une des options suivantes AWS SDKs, vous pouvez utiliser les options suivantes :

- AWS CLI:

Appelez la register-scalable-targetcommande correspondant à une variante de produit. L'exemple suivant enregistre le nombre d'instances souhaité pour une variante de produit appelée `my-variant`, exécutée sur le point de terminaison `my-endpoint`, avec une capacité minimale d'une instance et une capacité maximale de huit instances.

```
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
```

En cas de réussite, cette commande renvoie l'ARN de la cible évolutive.

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

- AWS SDK :

  Appelez l'opération RegisterScalableTarget et fournissez `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` et `MaxCapacity` comme paramètres.

Enregistrement de la concurrence provisionnée des points de terminaison sans serveur en tant que cibles évolutives avec Application Auto Scaling

Application Auto Scaling nécessite également une cible évolutive avant de pouvoir créer des politiques de mise à l'échelle ou des actions planifiées pour la concurrence provisionnée des points de terminaison sans serveur.

Si vous configurez le dimensionnement automatique à l'aide de la console SageMaker AI, l' SageMaker IA enregistre automatiquement une cible évolutive pour vous.

Sinon, utilisez l'une des méthodes suivantes pour enregistrer la cible évolutives :

- AWS CLI:

Appelez la register-scalable-targetcommande correspondant à une variante de produit. L'exemple suivant enregistre la concurrence provisionnée pour une variante de produit appelée `my-variant`, s'exécutant sur le point de terminaison `my-endpoint`, avec une capacité minimale de 1 et une capacité maximale de 10.

```
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
```

En cas de réussite, cette commande renvoie l'ARN de la cible évolutive.

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

- AWS SDK :

  Appelez l'opération RegisterScalableTarget et fournissez `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` et `MaxCapacity` comme paramètres.

Enregistrement des composants d'inférence en tant que cibles évolutives avec Application Auto Scaling

Application Auto Scaling requiert également une cible évolutive avant qu'il soit possible de créer des politiques de mise à l'échelle ou des actions planifiées pour les composants d'inférence.

- AWS CLI:

  Appelez la register-scalable-targetcommande d'un composant d'inférence. L'exemple suivant enregistre le nombre souhaité de copies pour un composant d'inférence appelé `my-inference-component`, avec une capacité minimale de zéro copie et une capacité maximale de trois.

```
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
```

En cas de réussite, cette commande renvoie l'ARN de la cible évolutive.

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

- AWS SDK :

  Appelez l'opération RegisterScalableTarget et fournissez `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` et `MaxCapacity` comme paramètres.

Ressources connexes

Si vous débutez avec Application Auto Scaling, vous trouverez des informations supplémentaires utiles sur le dimensionnement de vos ressources d' SageMaker IA dans le manuel Amazon SageMaker AI Developer Guide :

- Faites évoluer automatiquement les modèles Amazon SageMaker AI
- Adaptez automatiquement la simultanéité provisionnée pour un point de terminaison sans serveur
- Définissez des politiques de dimensionnement automatique pour les déploiements de terminaux multimodèles
- Mise à l'échelle automatique d'un point de terminaison asynchrone

> **Note**
>
> En 2023, l' SageMaker IA a introduit de nouvelles capacités d'inférence basées sur des points de terminaison d'inférence en temps réel. Vous créez un point de terminaison SageMaker AI avec une configuration de point de terminaison qui définit le type d'instance et le nombre initial d'instances pour le point de terminaison. Créez ensuite un composant d'inférence, qui est un objet d'hébergement d' SageMaker IA que vous pouvez utiliser pour déployer un modèle sur un point de terminaison. Pour plus d'informations sur le dimensionnement des composants d'inférence, consultez Amazon SageMaker AI ajoute de nouvelles fonctionnalités

d'inférence pour aider à réduire les coûts et la latence de déploiement des modèles de base et à réduire les coûts de déploiement des modèles de 50 % en moyenne en utilisant les dernières fonctionnalités d'Amazon SageMaker AI sur le AWS blog.

Application Auto Scaling pour le parc et les applications Amazon EC2 Spot

Vous pouvez mettre à l'échelle les parcs d'instances Spot à l'aide de politiques de suivi des cibles et d'échelonnement et de mise à l'échelle planifiée.

Utilisez les informations suivantes pour vous aider à intégrer un parc d'instances Spot avec Application Auto Scaling.

Rôle lié à un service créé pour un parc d'instances Spot

Le rôle lié au service suivant est automatiquement créé dans votre compte Compte AWS lorsque vous enregistrez les ressources de Spot Fleet en tant que cibles évolutives avec Application Auto Scaling. Ce rôle permet à Application Auto Scaling d'effectuer des opérations prises en charge au sein de votre compte. Pour de plus amples informations, veuillez consulter Rôles liés à un service pour Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_EC2SpotFleetRequest`

Principal du service utilisé par le rôle lié à un service

Le rôle lié à un service dans la section précédente ne peut être assumé que par le principal du service autorisé par les relations d'approbation définies pour le rôle. Le rôle lié à un service utilisé par Application Auto Scaling donne l'accès au principal du service suivant :

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

Enregistrement de parcs d'instances Spot en tant que cibles évolutives avec Application Auto Scaling

Application Auto Scaling requiert une cible évolutive avant de pouvoir créer des politiques de mise à l'échelle ou des actions planifiées pour un parc d'instances Spot. Une cible évolutive est une

ressource qu'Application Auto Scaling peut augmenter et diminuer. Les cibles évolutives sont identifiées de manière unique par la combinaison de l'ID de ressource, de la dimension évolutive et de l'espace de noms.

Si vous configurez la scalabilité automatique à l'aide de la console du parc d'instances Spot, le parc d'instances Spot enregistre automatiquement une cible évolutive pour vous.

Si vous souhaitez configurer le dimensionnement automatique à l'aide de la AWS CLI ou de l'une des options suivantes AWS SDKs, vous pouvez utiliser les options suivantes :

- AWS CLI:

  Appelez la commande register-scalable-target un parc d'instances Spot. L'exemple suivant enregistre la capacité cible d'un parc d'instances Spot à l'aide de son ID de demande, avec une capacité minimale de deux instances et une capacité maximale de 10 instances.

```
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
```

  En cas de réussite, cette commande renvoie l'ARN de la cible évolutive.

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

- AWS SDK :

  Appelez l'opération RegisterScalableTarget et fournissez `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` et `MaxCapacity` comme paramètres.

Ressources connexes

Pour plus d'informations, consultez Comprendre le dimensionnement automatique pour Spot Fleet dans le guide de EC2 l'utilisateur Amazon.

Amazon WorkSpaces et Application Auto Scaling

Vous pouvez redimensionner un pool WorkSpaces en utilisant des politiques de dimensionnement de suivi des cibles, des politiques de dimensionnement par étapes et un dimensionnement planifié.

Utilisez les informations suivantes pour vous aider à intégrer WorkSpaces Application Auto Scaling.

Rôle lié à un service créé pour WorkSpaces

Application Auto Scaling crée automatiquement le rôle lié au service nommé AWSServiceRoleForApplicationAutoScaling_WorkSpacesPool dans votre nom Compte AWS lorsque vous enregistrez des WorkSpaces ressources en tant que cibles évolutives avec Application Auto Scaling. Pour de plus amples informations, veuillez consulter Rôles liés à un service pour Application Auto Scaling.

Ce rôle lié à un service utilise la politique gérée. AWSApplication AutoscalingWorkSpacesPoolPolicy Cette politique accorde à Application Auto Scaling l'autorisation d'appeler Amazon WorkSpaces en votre nom. Pour plus d'informations, reportez-vous AWSApplicationAutoscalingWorkSpacesPoolPolicyà la section AWS Managed Policy Reference.

Principal du service utilisé par le rôle lié à un service

Le rôle lié au service fait confiance au principal de service suivant pour assumer le rôle :

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

Enregistrement WorkSpaces des pools en tant que cibles évolutives avec Application Auto Scaling

Application Auto Scaling nécessite une cible évolutive avant de pouvoir créer des politiques de dimensionnement ou des actions planifiées pour WorkSpaces. Une cible évolutive est une ressource qu'Application Auto Scaling peut augmenter et diminuer. Les cibles évolutives sont identifiées de manière unique par la combinaison de l'ID de ressource, de la dimension évolutive et de l'espace de noms.

Si vous configurez le dimensionnement automatique à l'aide de la WorkSpaces console, une cible évolutive est WorkSpaces automatiquement enregistrée pour vous.

Si vous souhaitez configurer le dimensionnement automatique à l'aide de la AWS CLI ou de l'une des options suivantes AWS SDKs, vous pouvez utiliser les options suivantes :

- AWS CLI:

  Appelez la register-scalable-targetcommande pour un pool de WorkSpaces. L'exemple suivant enregistre la capacité cible d'un pool WorkSpaces en utilisant son ID de demande, avec une capacité minimale de deux bureaux virtuels et une capacité maximale de dix bureaux virtuels.

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

  En cas de réussite, cette commande renvoie l'ARN de la cible évolutive.

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

- AWS SDK :

  Appelez l'opération RegisterScalableTarget et fournissez `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` et `MaxCapacity` comme paramètres.

Ressources connexes

Pour plus d'informations, consultez Auto Scaling for WorkSpaces Pools dans le guide d' WorkSpaces administration Amazon.

Ressources personnalisées et Application Auto Scaling

Vous pouvez mettre à l'échelle les ressources personnalisées à l'aide de politiques de suivi des cibles et d'échelonnement et de mise à l'échelle planifiée.

Utilisez les informations suivantes pour vous aider à intégrer les ressources personnalisées avec Application Auto Scaling.

Rôle lié à un service créé pour les ressources personnalisés

Le rôle lié au service suivant est automatiquement créé dans votre ordinateur Compte AWS lorsque vous enregistrez des ressources personnalisées en tant que cibles évolutives avec Application Auto Scaling. Ce rôle permet à Application Auto Scaling d'effectuer des opérations prises en charge au sein de votre compte. Pour de plus amples informations, veuillez consulter Rôles liés à un service pour Application Auto Scaling.

- `AWSServiceRoleForApplicationAutoScaling_CustomResource`

Principal du service utilisé par le rôle lié à un service

Le rôle lié à un service dans la section précédente ne peut être assumé que par le principal du service autorisé par les relations d'approbation définies pour le rôle. Le rôle lié à un service utilisé par Application Auto Scaling donne l'accès au principal du service suivant :

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

Enregistrement des ressources personnalisées en tant que cibles évolutives avec Application Auto Scaling

Application Auto Scaling requiert une cible évolutive avant de pouvoir créer des politiques de mise à l'échelle ou des actions planifiées pour une ressource personnalisée. Une cible évolutive est une ressource qu'Application Auto Scaling peut augmenter et diminuer. Les cibles évolutives sont identifiées de manière unique par la combinaison de l'ID de ressource, de la dimension évolutive et de l'espace de noms.

Pour configurer le dimensionnement automatique à l'aide de la AWS CLI ou de l'une des options suivantes AWS SDKs, vous pouvez utiliser les options suivantes :

- AWS CLI:

  Appelez la commande register-scalable-target pour une ressource personnalisée. L'exemple suivant enregistre une ressource personnalisée en tant que cible évolutive, avec un nombre minimum souhaité d'une unité de capacité et un nombre maximum souhaité de 10 unités de capacité. Le fichier `custom-resource-id.txt` contient une chaîne qui identifie l'ID de ressource, qui représente le chemin d'accès à la ressource personnalisée via votre point de terminaison 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
```

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

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

En cas de réussite, cette commande renvoie l'ARN de la cible évolutive.

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

- AWS SDK :

  Appelez l'opération RegisterScalableTarget et fournissez `ResourceId`, `ScalableDimension`, `ServiceNamespace`, `MinCapacity` et `MaxCapacity` comme paramètres.

Ressources connexes

Si vous débutez avec Application Auto Scaling, vous trouverez des informations supplémentaires utiles sur le dimensionnement de vos ressources personnalisées dans la documentation suivante :

GitHubréférentiel

Configurez les ressources Application Auto Scaling à l'aide de AWS CloudFormation

Application Auto Scaling est intégré à AWS CloudFormation un service qui vous aide à modéliser et à configurer vos AWS ressources afin que vous puissiez passer moins de temps à créer et à gérer vos ressources et votre infrastructure. Vous créez un modèle qui décrit toutes les AWS ressources que vous souhaitez, et qui CloudFormation fournit et configure ces ressources pour vous.

Lorsque vous l'utilisez CloudFormation, vous pouvez réutiliser votre modèle pour configurer vos ressources Application Auto Scaling de manière cohérente et répétée. Décrivez vos ressources une seule fois, puis fournissez les mêmes ressources encore et encore dans plusieurs Comptes AWS régions.

Application Auto Scaling et CloudFormation modèles

Pour allouer et configurer les ressources pour Application Auto Scaling et les services associés, vous devez maîtriser les modèles CloudFormation. Les modèles sont des fichiers texte formatés en JSON ou YAML. Ces modèles décrivent les ressources que vous souhaitez mettre à disposition dans vos CloudFormation piles. Si vous n'êtes pas familiarisé avec JSON ou YAML, vous pouvez utiliser CloudFormation Designer pour vous aider à démarrer avec les CloudFormation modèles. Pour plus d'informations, consultez Qu'est-ce que CloudFormation Designer ? dans le AWS CloudFormation Guide de l'utilisateur.

Lorsque vous créez un modèle de pile pour les ressources Application Auto Scaling, vous devez fournir les éléments suivants :

- Un espace de nom pour le service cible (par exemple, **`appstream`**). Consultez la AWS::ApplicationAutoScaling::ScalableTargetréférence pour obtenir les espaces de noms des services.
- Une dimension évolutive associée à la ressource cible (par exemple, **`appstream:fleet:DesiredCapacity`**). Consultez la AWS::ApplicationAutoScaling::ScalableTargetréférence pour obtenir des dimensions évolutives.
- Un ID de ressource pour la ressource cible (par exemple, **`fleet/sample-fleet`**). Consultez la AWS::ApplicationAutoScaling::ScalableTargetréférence pour obtenir des informations sur la syntaxe et des exemples de ressources spécifiques IDs.

- Un rôle lié au service pour la ressource cible (par exemple, **`arn:aws:iam::012345678910:role/aws-service-role/appstream.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_AppStreamFleet`**). Consultez le Référence ARN de rôle lié à un service tableau pour obtenir le rôle ARNs.

Pour en savoir plus sur les ressources Application Auto Scaling, consultez la référence Application Auto Scaling dans le AWS CloudFormation Guide de l'utilisateur.

Extraits de modèles d'exemple

Vous trouverez des exemples d'extraits à inclure dans les CloudFormation modèles dans les sections suivantes du Guide de l'AWS CloudFormation utilisateur :

- Pour des exemples de politiques de dimensionnement et d'actions planifiées, consultez la section Configurer les ressources Application Auto Scaling avec AWS CloudFormation.
- Pour plus d'exemples de politiques de dimensionnement, voir AWS::ApplicationAutoScaling::ScalingPolicy.

En savoir plus sur CloudFormation

Pour en savoir plus CloudFormation, consultez les ressources suivantes :

- AWS CloudFormation
- AWS CloudFormation Guide de l'utilisateur
- CloudFormation API Reference
- Guide de l'utilisateur de l'interface de ligne de commande AWS CloudFormation

Mise à l'échelle planifiée pour Application Auto Scaling

Avec la mise à l'échelle planifiée, vous pouvez configurer la mise à l'échelle automatique de votre application en fonction des changements de charge prévisibles en créant des actions planifiées qui augmentent ou diminuent la capacité à des moments précis. Cette procédure vous permet de mettre votre application à l'échelle de manière proactive afin qu'elle corresponde aux variations de charge prévisibles.

Supposons, par exemple, que vous ayez un schéma de trafic hebdomadaire régulier dans lequel la charge augmente en milieu de semaine et diminue vers la fin de la semaine. Vous pouvez configurer un calendrier de mise à l'échelle dans Application Auto Scaling qui s'aligne sur ce modèle :

- Mercredi matin, une action planifiée accroît la capacité en augmentant la capacité minimale précédemment définie de la cible évolutive.
- Vendredi soir, une autre action planifiée réduit la capacité en diminuant la capacité maximale précédemment définie de la cible évolutive.

Ces actions de mise à l'échelle planifiées vous permettent d'optimiser les coûts et les performances. Votre application dispose d'une capacité suffisante pour gérer le pic de trafic en milieu de semaine, mais elle ne surprovisionne pas de capacité inutile à d'autres moments.

Vous pouvez utiliser conjointement la mise à l'échelle planifiée et les stratégies de mise à l'échelle pour bénéficier des avantages des approches proactives et réactives de la mise à l'échelle. Après l'exécution d'une action de mise à l'échelle planifiée, la stratégie de mise à l'échelle peut continuer à prendre des décisions sur l'opportunité de poursuivre la mise à l'échelle de la capacité. Cela vous permet de vous assurer que vous avez une capacité suffisante pour gérer la charge de votre application. Bien que votre application soit mise à l'échelle pour répondre à la demande, la capacité actuelle doit se situer dans les limites de la capacité minimale et maximale qui a été fixée par votre action planifiée.

Table des matières

Comment fonctionne le dimensionnement planifié pour Application Auto Scaling

Cette rubrique décrit le fonctionnement du dimensionnement planifié et présente les principaux points à prendre en compte pour l'utiliser efficacement.

Table des matières

Comment ça marche

Pour utiliser la mise à l'échelle planifiée, créez des actions planifiées, qui indiquent à Application Auto Scaling d'effectuer des activités de mise à l'échelle à des heures spécifiques. Lorsque vous créez une action planifiée, vous spécifiez la cible évolutive, l'heure à laquelle l'activité de mise à l'échelle doit se produire, une capacité minimale et une capacité maximale. Vous pouvez créer des actions planifiées pour une mise à l'échelle unique ou selon une planification récurrente.

À l'heure spécifiée, Application Auto Scaling effectue une mise à l'échelle en fonction des nouvelles valeurs de capacité, en comparant la capacité actuelle à la capacité minimale et maximale spécifiée.

- Si la capacité actuelle est inférieure à la capacité minimale spécifiée, Application Auto Scaling effectue une augmentation puissance (augmente la capacité) jusqu'à la capacité minimale spécifiée.
- Si la capacité actuelle est supérieure à la capacité maximale spécifiée, Application Auto Scaling effectue une mise à l'échelle horizontale (diminue la capacité) jusqu'à la capacité maximale spécifiée.

Considérations

Lorsque vous créez une action planifiée, gardez les éléments suivants à l'esprit.

- Une action planifiée définit les `MinCapacity` et `MaxCapacity` sur ce qui est spécifié par l'action planifiée à la date et à l'heure spécifiées. La demande peut éventuellement inclure une seule de ces tailles. Par exemple, vous pouvez créer une action planifiée avec uniquement la capacité minimale spécifiée. Dans certains cas, cependant, vous devez inclure les deux tailles afin de garantir que la nouvelle capacité minimale ne dépasse pas la capacité maximale ou que la nouvelle capacité maximale n'est pas inférieure à la capacité minimale.
- Par défaut, les planifications récurrentes que vous définissez sont exprimées en heure UTC (temps universel coordonné). Vous pouvez modifier le fuseau horaire afin qu'elle corresponde à votre fuseau horaire local ou à celui d'une autre partie de votre réseau. Lorsque vous spécifiez un fuseau horaire qui observe l'heure d'été, l'action s'ajuste automatiquement pour l'heure d'été (DST). Pour de plus amples informations, veuillez consulter Planifiez des actions de dimensionnement récurrentes à l'aide d'Application Auto Scaling.
- Vous pouvez désactiver temporairement la mise à l'échelle planifiée pour une cible évolutive. Cela vous permet d'empêcher les actions planifiées d'être actives sans avoir à les supprimer. Vous pouvez ensuite reprendre la mise à l'échelle planifiée lorsque vous souhaitez l'utiliser à nouveau. Pour de plus amples informations, veuillez consulter Suspendez et reprenez le dimensionnement pour Application Auto Scaling.
- L'ordre dans lequel les actions planifiées s'exécutent est garanti pour la même cible évolutive, mais pas pour les actions planifiées entre les cibles évolutives.
- Pour une action planifiée complétée avec succès, la ressource spécifiée doit être dans un état évolutif dans le service cible. Si ce n'est pas le cas, la demande échoue et renvoie un message d'erreur, par exemple, `Resource Id [ActualResourceId] is not scalable. Reason: The status of all DB instances must be 'available' or 'incompatible-parameters'`.
- En raison de la nature distribuée d'Application Auto Scaling et des services cible, le délai entre le moment où l'action planifiée est déclenchée et celui où le service cible lance l'action de mise à l'échelle peut être de quelques secondes. Dans la mesure où les actions planifiées sont exécutées dans l'ordre dans lequel elles sont spécifiées, celles dont les heures de début planifiées sont trop proches les unes des autres peuvent prendre plus de temps à s'exécuter.

Commandes couramment utilisées pour la création, la gestion et la suppression d'actions planifiées

Les commandes couramment utilisées pour travailler avec la mise à l'échelle planifiée comprennent :

- register-scalable-targetpour enregistrer AWS ou personnaliser des ressources en tant que cibles évolutives (ressource qu'Application Auto Scaling peut redimensionner), et pour suspendre et reprendre le dimensionnement.
- put-scheduled-actionpour ajouter ou modifier des actions planifiées pour une cible évolutive existante.
- describe-scaling-activitiespour renvoyer des informations sur les activités de mise à l'échelle dans une AWS région.
- describe-scheduled-actionspour renvoyer des informations sur les actions planifiées dans une AWS région.
- delete-scheduled-actionpour supprimer une action planifiée.

Ressources connexes

Pour un exemple détaillé de l'utilisation du dimensionnement planifié, consultez le billet de blog Scheduling AWS Lambda Provisioned Concurrency for Recurrent Peak Usage sur le AWS Compute Blog.

Pour plus d'informations sur la création d'actions planifiées pour les groupes Auto Scaling, consultez la section Scheduled Scaling for Amazon EC2 Auto Scaling dans le guide de l'utilisateur Amazon EC2 Auto Scaling.

Limitations

Les limitations suivantes s'appliquent à l'utilisation de la mise à l'échelle planifiée :

- Les noms des actions planifiées doivent être uniques pour chaque cible évolutive.
- Application Auto Scaling ne fournit pas de précision de deuxième niveau dans les expressions de planification. Le niveau de résolution maximale lors de l'utilisation d'une expression Cron est d'une minute.
- La cible évolutive ne peut pas être un cluster Amazon MSK. La mise à l'échelle planifiée n'est pas prise en charge pour Amazon MSK.

- L'accès à la console pour consulter, ajouter, mettre à jour ou supprimer des actions planifiées sur des ressources évolutives dépend de la ressource que vous utilisez. Pour de plus amples informations, veuillez consulter Services AWS que vous pouvez utiliser avec Application Auto Scaling.

Créez des actions planifiées pour Application Auto Scaling à l'aide du AWS CLI

Les exemples suivants montrent comment créer des actions planifiées à l'aide de la AWS CLI put-scheduled-actioncommande. Lorsque vous indiquez la nouvelle capacité, vous pouvez indiquer une capacité minimum, une capacité maximum ou les deux.

Ces exemples utilisent des cibles évolutives pour certains des services intégrés à Application Auto Scaling. Pour utiliser une autre cible évolutive, spécifiez son espace de noms dans`--service-namespace`, sa dimension évolutive dans `--scalable-dimension` et son ID de ressource dans`--resource-id`.

Lorsque vous utilisez le AWS CLI, n'oubliez pas que vos commandes s'exécutent dans la Région AWS configuration adaptée à votre profil. Si vous souhaitez exécuter les commandes dans une autre région, modifiez la région par défaut pour votre profil, ou utilisez le paramètre `--region` avec la commande.

Exemples

- Créer une action planifiée qui ne se produit qu'une fois
- Créer une action planifiée qui s'exécute à un intervalle récurrent
- Créer une action planifiée qui s'exécute sur une planification récurrente
- Créer une action planifiée unique qui spécifie un fuseau horaire
- Créer une action planifiée récurrente qui spécifie un fuseau horaire

Créer une action planifiée qui ne se produit qu'une fois

Pour mettre à l'échelle automatiquement votre cible évolutive une seule fois, à une date et une heure spécifiées, utilisez l'option `--schedule "at(`*`yyyy-mm-ddThh:mm:ss`*`)"`.

Example Exemple : pour effectuer une montée en puissance unique

Voici un exemple de création d'une action planifiée pour augmenter la capacité à une date et une heure spécifiques.

À la date et à l'heure spécifiées pour `--schedule` (22 h 00 UTC le 31 mars 2021), si la valeur spécifiée pour `MinCapacity` est supérieure à la capacité actuelle, Application Auto Scaling augmente la capacité jusqu'à `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
```

Lorsque cette action planifiée est exécutée, si la capacité maximale est inférieure à la valeur spécifiée pour la capacité minimale, vous devez spécifier une nouvelle capacité minimale et maximale, et pas seulement la capacité minimale.

Example Exemple : pour effectuer une mise à l'échelle horizontale unique

Voici un exemple de création d'une action planifiée pour réduire la capacité à une date et une heure spécifiques.

À la date et à l'heure spécifiées pour `--schedule` (22 h 30 UTC le 31 mars 2021), si la valeur spécifiée pour `MaxCapacity` est inférieure à la capacité actuelle, Application Auto Scaling réduit la capacité jusqu'à `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
```

Créer une action planifiée qui s'exécute à un intervalle récurrent

Pour planifier la mise à l'échelle à un intervalle récurrent, utilisez l'option `--schedule "rate(value unit)"`. La valeur doit être un nombre entier positif. L'unité peut être `minute`, `minutes`, `hour`, `hours`, `day` ou `days`. Pour plus d'informations, consultez la section Expressions de taux dans le guide de EventBridge l'utilisateur Amazon.

Voici un exemple d'une action planifiée qui utilise une expression de fréquence.

Selon la planification spécifiée (toutes les 5 heures à partir du 30 janvier 2021 à 12 h 00 UTC et jusqu'au 31 janvier 2021 à 22 h 00 UTC), si la valeur spécifiée pour `MinCapacity` est supérieure à la capacité actuelle, Application Auto Scaling augmente la capacité jusqu'à `MinCapacity`. Si la valeur spécifiée pour `MaxCapacity` est inférieure à la capacité actuelle, Application Auto Scaling réduit la capacité jusqu'à `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
```

Créer une action planifiée qui s'exécute sur une planification récurrente

Pour planifier la mise à l'échelle selon un calendrier récurrent, utilisez l'option `--schedule "cron(fields)"` Pour de plus amples informations, veuillez consulter Planifiez des actions de dimensionnement récurrentes à l'aide d'Application Auto Scaling.

Voici un exemple d'une action planifiée qui utilise une expression cron.

Selon la planification spécifiée (toutes les jours à 9 h 00 UTC), si la valeur spécifiée pour `MinCapacity` est supérieure à la capacité actuelle, Application Auto Scaling augmente la capacité jusqu'à `MinCapacity`. Si la valeur spécifiée pour `MaxCapacity` est inférieure à la capacité actuelle, Application Auto Scaling réduit la capacité jusqu'à `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
```

Créer une action planifiée unique qui spécifie un fuseau horaire

Les actions planifiées sont définies par défaut sur le fuseau horaire UTC. Pour spécifier un fuseau horaire différent, incluez l'option `--timezone` et spécifiez le nom canonique du fuseau horaire (`America/New_York`, par exemple). Pour plus d'informations, consultez https://www.joda.org/joda-time/timezones.html, qui fournit des informations sur les fuseaux horaires IANA pris en charge lors des appels put-scheduled-action.

Voici un exemple qui utilise l'option `--timezone` lors de la création d'une action planifiée pour mettre à l'échelle la capacité à une date et une heure spécifiques.

À la date et à l'heure spécifiées pour `--schedule` (17 h, heure locale, le 31 janvier 2021), si la valeur spécifiée pour `MinCapacity` est supérieure à la capacité actuelle, Application Auto Scaling augmente la capacité jusqu'à `MinCapacity`. Si la valeur spécifiée pour `MaxCapacity` est inférieure à la capacité actuelle, Application Auto Scaling réduit la capacité jusqu'à `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
```

Créer une action planifiée récurrente qui spécifie un fuseau horaire

Voici un exemple qui utilise l'option `--timezone` lors de la création d'une action planifiée récurrente pour mettre à l'échelle la capacité. Pour de plus amples informations, veuillez consulter Planifiez des actions de dimensionnement récurrentes à l'aide d'Application Auto Scaling.

Selon la planification spécifiée (du lundi au vendredi à 18 h 00, heure locale), si la valeur spécifiée pour `MinCapacity` est supérieure à la capacité actuelle, Application Auto Scaling augmente la capacité jusqu'à `MinCapacity`. Si la valeur spécifiée pour `MaxCapacity` est inférieure à la capacité actuelle, Application Auto Scaling réduit la capacité jusqu'à `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
```

Décrivez le dimensionnement planifié pour Application Auto Scaling à l'aide du AWS CLI

Ces exemples de AWS CLI commandes décrivent les activités de dimensionnement et les actions planifiées à l'aide de ressources provenant de services intégrés à Application Auto Scaling. Pour une autre cible évolutive, spécifiez son espace de noms dans`--service-namespace`, sa dimension évolutive dans `--scalable-dimension` et son ID de ressource dans`--resource-id`.

Lorsque vous utilisez le AWS CLI, n'oubliez pas que vos commandes s'exécutent dans la Région AWS configuration adaptée à votre profil. Si vous souhaitez exécuter les commandes dans une autre région, modifiez la région par défaut pour votre profil, ou utilisez le paramètre `--region` avec la commande.

Exemples

- Décrire les activités de dimensionnement d'un service
- Décrire les actions planifiées pour un service

- Décrire les actions planifiées pour une cible évolutive

Décrire les activités de dimensionnement d'un service

Pour afficher les activités de dimensionnement pour toutes les cibles évolutives dans un espace de noms de service spécifié, utilisez la describe-scaling-activitiescommande.

L'exemple suivant récupère les activités de mise à l'échelle associées à l'espace de nom de service dynamodb.

Linux, macOS ou Unix

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

Windows

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

Output

Si la commande aboutit, elle renvoie un résultat similaire à ce qui suit.

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


Pour modifier cette commande afin qu'elle récupère les activités de mise à l'échelle pour une seule de vos cibles évolutives, ajoutez l'option `--resource-id`.

Décrire les actions planifiées pour un service

Pour décrire les actions planifiées pour toutes les cibles évolutives d'un espace de noms de service spécifié, utilisez la describe-scheduled-actionscommande.

L'exemple suivant récupère les actions planifiées associées à l'espace de nom du service `ec2`.

Linux, macOS ou Unix

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

Windows

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

Output

Si la commande aboutit, elle renvoie un résultat similaire à ce qui suit.

```
{
    "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
        }
    ]
}
```

Décrire les actions planifiées pour une cible évolutive

Pour récupérer des informations sur les actions planifiées pour une cible évolutive spécifiée, ajoutez l'`--resource-id`option lors de la description des actions planifiées à l'aide de la describe-scheduled-actionscommande.

Si vous incluez l'option `--scheduled-action-names` et spécifiez le nom d'une action planifiée comme valeur, la commande renvoie uniquement l'action planifiée dont le nom correspond, comme le montre l'exemple suivant.

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
```

Output

Si la commande aboutit, elle renvoie un résultat similaire à ce qui suit. Si vous avez fourni plusieurs valeurs pour`--scheduled-action-names`, la sortie inclut toutes les actions planifiées dont les noms correspondent.

```
{
    "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
        }
    ]
}
```

Planifiez des actions de dimensionnement récurrentes à l'aide d'Application Auto Scaling

> **Important**
>
> Pour obtenir de l'aide sur les expressions cron pour Amazon EC2 Auto Scaling, consultez la rubrique sur les plannings récurrents du guide de l'utilisateur d'Amazon EC2 Auto Scaling. Avec Amazon EC2 Auto Scaling, vous utilisez la syntaxe cron traditionnelle au lieu de la syntaxe cron personnalisée utilisée par Application Auto Scaling.

Vous pouvez créer des actions planifiées selon une planification récurrente à l'aide d'une expression cron.

Pour créer une planification récurrente, spécifiez une expression cron et un fuseau horaire à décrire quand cette action planifiée doit se répéter. Les valeurs de fuseau horaire prises en charge sont les noms canoniques des fuseaux horaires IANA pris en charge par Joda-Time (tels que `Etc/GMT+9` ou `Pacific/Tahiti`). Vous pouvez éventuellement spécifier une date et une heure pour l'heure de début, l'heure de fin, voire les deux. Pour un exemple de commande qui utilise le AWS CLI pour créer une action planifiée, consultezCréer une action planifiée récurrente qui spécifie un fuseau horaire.

L'expression cron prise en charge est constituée de six champs séparés par des espaces blancs : [Minutes] [Heures] [Jour_du_Mois] [Mois] [Jour_de_la_semaine] [Année]. Par exemple, l'expression cron `30 6 ? * MON *` configure une action planifiée qui se répète tous les lundis à 6h30. L'astérisque est utilisé comme caractère générique pour représenter toutes les valeurs d'un champ.

Pour plus d'informations sur la syntaxe cron pour les actions planifiées d'Application Auto Scaling, consultez la référence des expressions Cron dans le guide de EventBridge l'utilisateur Amazon.

Lorsque vous créez une planification récurrente, choisissez avec soin vos heures de début et de fin. Gardez à l'esprit les points suivants :

- Si vous spécifiez une heure de début, Application Auto Scaling exécute l'action à ce moment, puis exécute l'action basée sur la planification récurrente.
- Si vous spécifiez une heure de fin, l'action cesse de se répéter après cette heure. Application Auto Scaling ne garde pas la trace des valeurs précédentes et revient à ces valeurs précédentes une fois terminée.

- L'heure de début et l'heure de fin doivent être définies en UTC lorsque vous utilisez le AWS CLI ou AWS SDKs pour créer ou mettre à jour une action planifiée.

Exemples

Vous pouvez faire référence au tableau suivant lorsque vous créez une planification récurrente pour une cible scalable Application Auto Scaling. Les exemples suivants sont la syntaxe correcte pour utiliser Application Auto Scaling pour créer ou mettre à jour une action planifiée.

Minutes	Heures	Jour du mois	Mois	Jour de la semaine	Année	Signification
0 USD	10	*	*	?	*	Exécuter à 10 h 00 (UTC) chaque jour
15	12	*	*	?	*	Exécuter à 12 h 15 (UTC) chaque jour
0	18	?	*	MON-FRI	*	Exécuter à 18 h 00 (UTC) du lundi au vendredi
0	8	1	*	?	*	Exécuter à 8 h 00 (UTC) le 1er jour de chaque mois

Minutes	Heures	Jour du mois	Mois	Jour de la semaine	Année	Signification
0/15	*	*	*	?	*	Exécuter toutes les 15 minutes
0/10	*	?	*	MON-FRI	*	Exécuter toutes les 10 minutes du lundi au vendredi
0/5	8-17	?	*	MON-FRI	*	Exécuter toutes les 5 minutes du lundi au vendredi entre 8 h 00 et 17 h 55 (UTC)

Exception

Vous pouvez également créer une expression cron avec une valeur de chaîne contenant sept champs. Dans ce cas, vous pouvez utiliser les trois premiers champs pour spécifier l'heure à laquelle une action planifiée doit être exécutée, y compris les secondes. L'expression cron complète est constituée des champs séparés par des espaces suivants : [Secondes] [Minutes] [Heures] [Jour_du_Mois] [Mois] [Jour_du_la_semaine] [Année]. Toutefois, cette approche ne garantit pas que l'action planifiée s'exécutera à la seconde précise que vous spécifiez. De plus, certaines consoles de service peuvent ne pas prendre en charge le champ des secondes dans une expression cron.

Désactiver la mise à l'échelle planifiée pour une cible évolutive

Vous pouvez désactiver temporairement la mise à l'échelle planifiée sans supprimer vos actions planifiées. Pour de plus amples informations, veuillez consulter Suspendez et reprenez le dimensionnement pour Application Auto Scaling.

Pour suspendre le dimensionnement planifié

Suspendez le dimensionnement planifié sur une cible évolutive en utilisant la register-scalable-targetcommande associée à l'`--suspended-state`option et en spécifiant `true` comme valeur de l'`ScheduledScalingSuspended`attribut, comme indiqué dans l'exemple suivant.

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}"
```

Output

Si la commande aboutit, elle renvoie l'ARN de la cible évolutive. Voici un exemple de sortie.

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

Pour reprendre le dimensionnement planifié

Pour reprendre le dimensionnement planifié, réexécutez la register-scalable-target commande en spécifiant `false` comme valeur pour`ScheduledScalingSuspended`.

Supprimer une action planifiée pour Application Auto Scaling à l'aide du AWS CLI

Lorsque vous avez terminé avec une action planifiée, vous pouvez la supprimer.

Pour supprimer votre action planifiée

Utilisez la commande delete-scheduled-action. En cas de succès, cette commande ne renvoie aucune sortie.

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
```

Pour annuler l'inscription d'une cible évolutive

Si vous en avez également terminé avec la cible évolutive, vous pouvez la désenregistrer. Utilisez la commande deregister-scalable-target suivante. Si des politiques de dimensionnement ou des actions planifiées n'ont pas encore été supprimées, elles sont supprimées par cette commande. En cas de succès, cette commande ne renvoie aucune sortie.

Linux, macOS ou Unix

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

Windows

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

Politique de suivi des cibles et d'échelonnement pour Application Auto Scaling

Une stratégie de mise à l’échelle par suivi des cibles permet de mettre automatiquement à l’échelle votre application sur la base d’une valeur métrique cible. Cela permet à votre application de maintenir des performances et une rentabilité optimales sans intervention manuelle.

Avec le suivi des cibles, vous sélectionnez une métrique et une valeur cible pour représenter le niveau d’utilisation ou de débit moyen idéal pour votre application. Application Auto Scaling crée et gère les CloudWatch alarmes qui déclenchent des événements de dimensionnement lorsque la métrique s'écarte de la cible. Cela est similaire à la façon dont un thermostat maintient une température cible.

Supposons par exemple que vous avez une application Web qui s'exécute actuellement sur un parc d’instances Spot et vous souhaitez que l'utilisation de l'UC de la flotte reste à environ 50 % lorsque la charge sur l'application change. Vous disposez ainsi d'une plus grande capacité pour gérer les pics de trafic sans avoir à maintenir une quantité excessive des ressources inutilisées.

Vous pouvez répondre à ce besoin en créant une stratégie de suivi des objectifs et d'échelonnement qui cible une utilisation moyenne du CPU de 50 pour cent. Application Auto Scaling monte ensuite en puissance (augmente la capacité) lorsque le processeur dépasse 50 % pour faire face à une charge accrue. Il effectuera une mise à l’échelle horizontale (diminution de la capacité) lorsque la capacité du processeur sera inférieure à 50 %, afin d’optimiser les coûts pendant les périodes de faible utilisation.

Les politiques de suivi des cibles éliminent le besoin de définir manuellement les CloudWatch alarmes et les ajustements de dimensionnement. Application Auto Scaling gère cela automatiquement en fonction de la cible que vous avez définie.

Vous pouvez établir des stratégies de suivi des cibles en fonction d’indicateurs prédéfinis ou personnalisés :

- Métriques prédéfinies : métriques fournies par Application Auto Scaling, telles que l’utilisation moyenne du processeur ou le nombre moyen de requêtes par cible.
- Mesures personnalisées : vous pouvez utiliser les mathématiques des métriques pour combiner des métriques, tirer parti des métriques existantes ou utiliser vos propres métriques personnalisées publiées sur. CloudWatch

Choisissez une métrique qui change de manière inversement proportionnelle à un changement dans la capacité de votre cible évolutive. Donc, si vous doublez la capacité, la métrique diminue de 50 %. Cela permet aux données métriques de déclencher avec précision des événements de mise à l'échelle proportionnelle.

Table des matières

Comment fonctionne le dimensionnement du suivi des cibles pour Application Auto Scaling

Cette rubrique décrit le fonctionnement du dimensionnement du suivi des cibles et présente les éléments clés d'une politique de dimensionnement du suivi des cibles.

Table des matières

Comment ça marche

Pour utiliser le dimensionnement du suivi des cibles, vous devez créer une politique de dimensionnement du suivi des cibles et spécifier les éléments suivants :

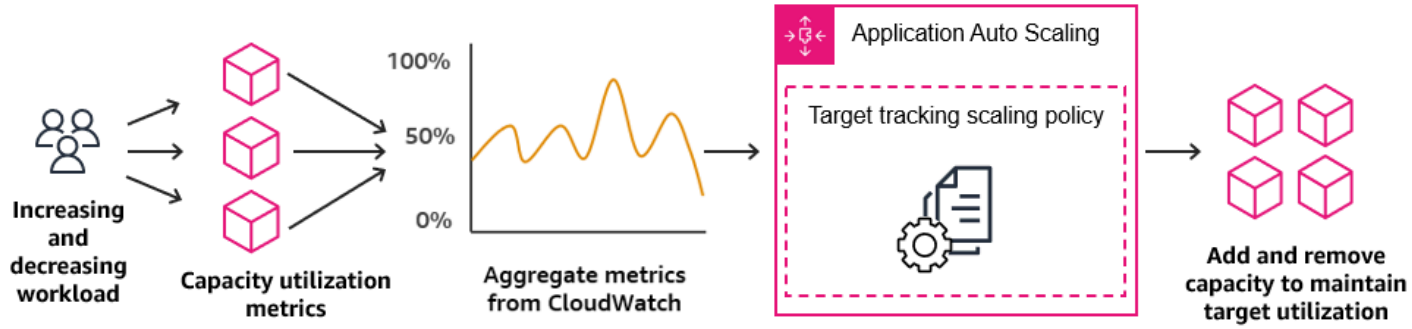
- Métrique : CloudWatch métrique à suivre, telle que l'utilisation moyenne du processeur ou le nombre moyen de demandes par cible.
- Valeur cible : la valeur cible de la métrique, telle que 50 % d'utilisation du processeur ou 1 000 demandes par cible et par minute.

Application Auto Scaling crée et gère les CloudWatch alarmes qui invoquent la politique de dimensionnement et calcule l'ajustement de dimensionnement en fonction de la métrique et de la valeur cible. Il ajoute et supprime de la capacité en fonction des besoins pour maintenir la métrique à la valeur cible spécifiée ou à une valeur proche de celle-ci.

Lorsque la métrique est supérieure à la valeur cible, Application Auto Scaling monte en puissance en ajoutant de la capacité afin de réduire la différence entre la valeur de la métrique et celle cible. Lorsque la métrique est inférieure à la valeur cible, Application Auto Scaling met à l'échelle de façon horizontale en supprimant de la capacité.

Les activités de mise à l'échelle sont effectuées avec des temps de stabilisation entre eux afin d'éviter des fluctuations rapides de capacité. Vous pouvez éventuellement configurer les temps de stabilisation de votre stratégie de mise à l'échelle.

Le diagramme suivant montre un aperçu du fonctionnement d'une politique de mise à l'échelle du suivi de cible lorsque la configuration est terminée.



Il convient de noter qu'une stratégie de mise à l'échelle de suivi des cibles est plus agressive pour ajouter de la capacité lorsque l'utilisation augmente que pour supprimer de la capacité lorsque l'utilisation diminue. Par exemple, si la métrique spécifiée de la politique atteint sa valeur cible, la

politique suppose que votre application est déjà massivement chargée. Elle répond en ajoutant une capacité proportionnelle à la valeur de la métrique aussi vite que possible. Plus la métrique est élevée, plus il y a de capacité ajoutée.

Lorsque la métrique tombe en dessous de la valeur cible, la politique ne s'adaptera pas si elle calcule que la suppression d'une unité de capacité minimale ramènerait probablement la métrique au-dessus de la valeur cible. Dans ce cas, elle ralentit donc la mise à l'échelle en supprimant la capacité uniquement lorsque l'utilisation dépasse un seuil suffisamment inférieur à la valeur cible (généralement plus de 10 % de moins) pour que l'utilisation soit considérée comme ayant ralenti. L'intention de ce comportement plus conservateur est de s'assurer que la suppression de la capacité ne se produit que lorsque l'application ne connaît plus une demande au même niveau élevé qu'auparavant.

Choisissez métriques

Vous pouvez créer des stratégies de suivi des objectifs de la mise à l'échelle avec des métriques prédéfinies ou des métriques personnalisées.

Lorsque vous créez une politique de mise à l'échelle de suivi des cibles avec une métrique prédéfinie, vous choisissez une métrique dans la liste de métriques prédéfinies dans Métrique prédéfinie pour la politique de mise à l'échelle de suivi des cibles.

Gardez les points suivants à l'esprit lorsque vous choisissez une métrique :

- Toutes les métriques personnalisées ne fonctionnent pas pour le suivi des cibles. La métrique doit être une métrique d'utilisation valide et décrire le degré d'occupation d'une cible évolutive. La valeur de la métrique doit augmenter et diminuer proportionnellement à la capacité de la cible évolutive pour que les données de la métrique puissent être utilisées afin d'augmenter ou réduire proportionnellement la cible évolutive.
- Pour utiliser la métrique `ALBRequestCountPerTarget`, vous devez spécifier le paramètre `ResourceLabel` permettant d'identifier le groupe cible associé à la métrique.
- Lorsqu'une métrique émet des valeurs réelles de 0 à CloudWatch (par exemple,`ALBRequestCountPerTarget`), Application Auto Scaling peut passer à 0 lorsqu'aucun trafic n'est acheminé vers votre application pendant une période prolongée. Pour que la capacité de votre cible évolutive diminue jusqu'à 0 lorsqu'aucune demande ne lui est acheminée, la capacité minimale de la cible évolutive doit être définie sur 0.
- Au lieu de publier de nouvelles métriques à utiliser dans votre politique de mise à l'échelle, vous pouvez utiliser les calculs de métriques pour combiner des métriques existantes. Pour de plus

amples informations, veuillez consulter Créer une stratégie de mise à l'échelle du suivi des cibles pour Application Auto Scaling à l'aide des mathématiques appliquées aux métriques..

- Pour savoir si le service que vous utilisez prend en charge la spécification d'une métrique personnalisée dans la console du service, consultez la documentation de ce service.
- Nous vous recommandons d'utiliser des mesures disponibles à des intervalles d'une minute pour vous aider à évoluer plus rapidement en fonction des changements d'utilisation. Le suivi des cibles évaluera les métriques agrégées avec une granularité d'une minute pour toutes les métriques prédéfinies et les métriques personnalisées, mais la métrique sous-jacente peut publier des données moins fréquemment. Par exemple, toutes les EC2 métriques Amazon sont envoyées à intervalles de cinq minutes par défaut, mais elles sont configurables à une minute (ce que l'on appelle la surveillance détaillée). Ce choix appartient à chaque service. La plupart essaient d'utiliser le plus petit intervalle possible.

Définition de la valeur cible

Lorsque vous créez une politique de suivi de la cible, vous devez spécifier une valeur cible. La valeur cible représente l'utilisation ou le débit moyen optimal pour votre application. Afin d'utiliser les ressources de manière efficiente, définissez une valeur cible aussi élevée que possible avec un tampon raisonnable en cas d'augmentation inattendue du trafic. Lorsque votre application est mise à l'échelle de manière optimale pour un flux de trafic normal, la valeur de métrique réelle doit être égale ou sensiblement inférieure à la valeur cible.

Lorsqu'une stratégie de dimensionnement est basée sur le débit, tel que le nombre de demandes par cible pour un Application Load Balancer, les I/O réseau ou d'autres métriques de nombre, la valeur cible représente le débit moyen optimal depuis une seule entité (comme une seule cible de votre groupe cible Application Load Balancer), pendant une période d'une minute.

Définir les temps de stabilisation

Vous pouvez éventuellement définir des temps de stabilisation dans votre politique de mise à l'échelle de suivi des cibles.

Le temps de stabilisation spécifie la durée pendant laquelle la politique de mise à l'échelle attend qu'une activité de mise à l'échelle précédente prenne effet.

Il existe deux types de temps de stabilisation :

- Avec le temps de stabilisation de montée en charge, l'intention est de monter continuellement en charge (mais sans excès). Une fois qu'Application Auto Scaling a réussi une montée en puissance à l'aide d'une politique de mise à l'échelle par étape, l'application commence à calculer le temps de stabilisation. La politique de mise à l'échelle n'augmente pas à nouveau la capacité souhaitée, sauf si une plus grande montée en puissance parallèle est déclenchée ou si le temps de stabilisation est écoulé. Tandis que le temps de stabilisation de la montée en puissance s'applique, la capacité ajoutée par l'activité de mise à l'échelle initiale est calculée dans le cadre de la capacité souhaitée pour la prochaine activité de montée en puissance.
- Avec le temps de stabilisation de mise à l'échelle horizontale, l'intention est de procéder à une mise à l'échelle horizontale de façon conservatrice pour protéger la disponibilité de votre application, de sorte que les activités de mise à l'échelle horizontale sont bloquées jusqu'à ce que le temps de stabilisation de la mise à l'échelle horizontale ait expiré. Toutefois, si une autre alarme déclenche une activité de montée en charge au cours du temps de stabilisation de la diminution de charge, Application Auto Scaling monte immédiatement en charge la cible. Dans ce cas, le temps de stabilisation de la mise à l'échelle horizontale s'arrête et ne se termine pas.

Chaque temps de stabilisation est mesuré en secondes et s'applique uniquement aux activités de mise à l'échelle liées à la stratégie de mise à l'échelle . Pendant un temps de stabilisation, lorsqu'une action planifiée démarre à l'heure prévue, elle peut déclencher une activité de mise à l'échelle immédiatement sans attendre l'expiration du temps de stabilisation.

Vous pouvez commencer par les valeurs par défaut, qui peuvent être affinées ultérieurement. Par exemple, vous devrez peut-être augmenter un temps de stabilisation pour éviter que votre stratégie de mise à l'échelle Suivi de la cible ne soit pas trop agressive face aux modifications survenant sur de courtes périodes.

Valeurs par défaut

Application Auto Scaling fournit une valeur par défaut de 600 ElastiCache et une valeur par défaut de 300 pour les cibles évolutives suivantes :

- WorkSpaces Flottes d'applications
- Clusters DB Aurora
- Services ECS
- Clusters Neptune
- SageMaker Variantes de terminaux AI

- SageMaker Composants d'inférence AI
- SageMaker Concurrence provisionnée sans serveur AI
- Spot Fleets
- Piscine de WorkSpaces
- Ressources personnalisées

Pour toutes les autres cibles évolutives, la valeur par défaut est 0 ou nulle :

- Classification de documents et points de terminaison de module de reconnaissance d'entité Amazon Comprehend
- Tables DynamoDB et index secondaires globaux
- Tables Amazon Keyspaces
- Simultanéité allouée Lambda
- Stockage Amazon MSK Broker

Les valeurs nulles sont traitées de la même manière que les valeurs 0 lorsqu'Application Auto Scaling évalue le temps de stabilisation.

Vous pouvez mettre à jour toutes les valeurs par défaut, y compris les valeurs nulles, pour définir vos propres temps de stabilisation.

Considérations

Les points suivants s'appliquent lors de l'utilisation des politiques de suivi des objectifs et d'échelonnement

- Ne créez pas, ne modifiez ni ne supprimez les CloudWatch alarmes utilisées avec une politique de dimensionnement du suivi des cibles. Application Auto Scaling crée et gère les CloudWatch alarmes associées à vos politiques de dimensionnement du suivi des cibles et les supprime lorsqu'elles ne sont plus nécessaires.
- S'il manque des points de données à la métrique, l'état de l' CloudWatch alarme passe à`INSUFFICIENT_DATA`. Dans ce cas, Application Auto Scaling ne peut pas mettre à l'échelle votre cible capable d'être mise à l'échelle tant que de nouveaux points de données ne sont pas trouvés. Pour plus d'informations, consultez la section Configuration de la façon dont les CloudWatch alarmes traitent les données manquantes dans le guide de CloudWatch l'utilisateur Amazon.

- Si la métrique est rarement rapportée, les calculs de métriques peuvent s'avérer utiles. Par exemple, pour utiliser les valeurs les plus récentes, utilisez la fonction `FILL(m1,REPEAT)` là où `m1` est la métrique.
- Vous pouvez constater des écarts entre la valeur cible et les points de données de métrique réels. Ceci est dû au fait qu'Application Auto Scaling agit toujours avec prudence en effectuant un arrondi vers le haut ou vers le bas quand il détermine la capacité à ajouter ou à enlever. Cela l'empêche d'ajouter une capacité insuffisante ou de retirer trop de capacité. Toutefois, pour une cible scalable avec une faible capacité, les points de données de métrique réels peuvent sembler éloignés de la valeur cible.

  Supposons, par exemple, que vous définissez une valeur cible de 50 % pour l'utilisation du processeur et que votre groupe Auto Scaling dépasse ensuite la cible. Nous pouvons déterminer que l'ajout de 1,5 instance diminuera l'utilisation de l'UC d'environ 50 %. Comme il n'est pas possible d'ajouter 1,5 instance, nous arrondissons à la valeur supérieure et ajoutons deux instances. Cela peut diminuer l'utilisation de la CPU à une valeur inférieure à 50 % mais cela garantit que votre application dispose de suffisamment de ressources pour le prendre en charge. De même, si nous déterminons que la suppression de 0,5 instance augmente l'utilisation de votre processeur à plus de 50 %, nous choisirons de ne pas procéder à une mise à l'échelle tant que la métrique ne sera pas suffisamment basse pour que nous puissions penser que la mise à l'échelle ne provoquera pas d'oscillation.

  Pour une cible scalable avec une plus grande capacité, l'ajout ou le retrait de capacité entraîne moins d'écarts entre la valeur cible et les points de données de métrique réels.
- Une politique de suivi des objectifs et d'échelonnement suppose qu'elle doit effectuer une montée en puissance ; lorsque la métrique spécifiée est au-dessus de la valeur cible. Vous ne pouvez pas utiliser une politique de suivi des objectifs et d'échelonnement pour effectuer une montée en puissance lorsque la métrique spécifiée est en dessous de la valeur cible.

Plusieurs stratégies de dimensionnement

Vous pouvez avoir plusieurs stratégies de dimensionnement Suivi de la cible pour une cible évolutive, dans la mesure où chacune d'elles utilise une métrique différente. L'objectif d'Application Auto Scaling est de toujours donner la priorité à la disponibilité, afin que son comportement diffère selon que les politiques de suivi des cibles et d'échelonnement sont prêtes pour une augmentation ou une diminution de la capacité. Il augmentera la taille de la cible scalable si une des stratégies Suivi de la cible est prête pour une augmentation de taille, mais la diminuera uniquement si toutes les stratégies

Suivi de la cible (avec la portion de diminution en charge activée) sont prêtes pour une diminution de taille.

Si plusieurs stratégies de mise à l'échelle indiquent simultanément à la cible capable d'être mise à l'échelle de procéder à une montée en puissance ou à une mise à l'échelle horizontale, Application Auto Scaling effectue la mise à l'échelle en fonction de la stratégie qui fournit la plus grande capacité à la fois pour la mise à l'échelle horizontale et la montée en puissance. Cela vous offre une plus grande flexibilité pour couvrir plusieurs scénarios et pouvoir toujours disposer d'une capacité suffisante pour traiter vos charges de travail.

Vous pouvez désactiver la portion de mise à l'échelle horizontale d'une politique de mise à l'échelle de suivi des cibles pour utiliser une méthode différente de mise à l'échelle horizontale par rapport à la montée en puissance. Vous pouvez, par exemple, utiliser une stratégie de mise à l'échelle par étapes pour effectuer une diminution en charge tout en utilisant une stratégie de dimensionnement Suivi de la cible pour effectuer une montée en charge.

Toutefois, nous vous recommandons d'être prudent lorsque vous utilisez des politiques de suivi des objectifs et d'échelonnement avec des politiques de mise à l'échelle par étapes, car les conflits entre ces politiques peuvent entraîner un comportement indésirable. Par exemple, si la politique de mise à l'échelle par étapes lance une activité de mise à l'échelle horizontale avant que la politique de suivi des objectifs et d'échelonnement ne soit prête pour la mise à l'échelle horizontale, l'activité de mise à l'échelle horizontale ne sera pas bloquée. Une fois l'activité de diminution en charge terminée, la stratégie de dimensionnement Suivi de la cible peut demander à la cible évolutive d'effectuer une montée en charge.

Pour les charges de travail de nature cyclique, vous avez également la possibilité d'automatiser les modifications de capacité sur une planification à l'aide d'une mise à l'échelle planifiée. Pour chaque action planifiée, une nouvelle valeur de capacité minimale et une nouvelle valeur de capacité maximale peuvent être définies. Ces valeurs constituent les limites de la stratégie de mise à l'échelle. La combinaison de la mise à l'échelle planifiée et de la mise à l'échelle Suivi des cibles peut aider à réduire l'impact d'une forte augmentation des niveaux d'utilisation, lorsque la capacité est nécessaire immédiatement.

Commandes couramment utilisées pour la création, la gestion et la suppression des politiques de mise à l'échelle

Les commandes couramment utilisées pour travailler avec les politiques de mise à l'échelle sont les suivantes :

- register-scalable-targetpour enregistrer AWS ou personnaliser des ressources en tant que cibles évolutives (ressource qu'Application Auto Scaling peut redimensionner), et pour suspendre et reprendre le dimensionnement.
- put-scaling-policypour ajouter ou modifier des politiques de dimensionnement pour une cible évolutive existante.
- describe-scaling-activitiespour renvoyer des informations sur les activités de mise à l'échelle dans une AWS région.
- describe-scaling-policiespour renvoyer des informations sur les politiques de dimensionnement dans une AWS région.
- delete-scaling-policypour supprimer une politique de dimensionnement.

Ressources connexes

Pour plus d'informations sur la création de politiques de dimensionnement du suivi des cibles pour les groupes Auto Scaling, consultez les politiques de dimensionnement du suivi des cibles pour Amazon EC2 Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Limitations

Les limitations sont les suivantes lors de l'utilisation des politiques de suivi des cibles et d'échelonnement :

- La cible évolutive ne peut pas être un cluster Amazon EMR. Les politiques de suivi des cibles et d'échelonnement ne sont pas prises en charge pour Amazon EMR.
- Lorsqu'un cluster Amazon MSK est la cible évolutive, la diminution de charge est désactivée et ne peut pas être activée.
- Vous ne pouvez pas utiliser les opérations de `PutScalingPolicy` l'API `RegisterScalableTarget` or pour mettre à jour un plan de AWS Auto Scaling dimensionnement.
- L'accès à la console pour consulter, ajouter, mettre à jour ou supprimer les politiques de suivi des cibles et de mise à l'échelle des ressources évolutives dépend de la ressource que vous utilisez. Pour de plus amples informations, veuillez consulter Services AWS que vous pouvez utiliser avec Application Auto Scaling.

Créez une politique de dimensionnement du suivi des cibles pour Application Auto Scaling à l'aide du AWS CLI

Cet exemple utilise des AWS CLI commandes pour créer une politique de rayonnage cible pour un Amazon EC2 Spot Fleet. Pour une autre cible évolutive, spécifiez son espace de noms dans`--service-namespace`, sa dimension évolutive dans `--scalable-dimension` et son ID de ressource dans`--resource-id`.

Lorsque vous utilisez le AWS CLI, n'oubliez pas que vos commandes s'exécutent dans la Région AWS configuration adaptée à votre profil. Si vous souhaitez exécuter les commandes dans une autre région, modifiez la région par défaut pour votre profil, ou utilisez le paramètre `--region` avec la commande.

Tâches

- Étape 1 : enregistrer une cible évolutive
- Étape 2 : créer une politique de suivi des objectifs et d'échelonnement
- Étape 3 : Décrire les politiques de dimensionnement du suivi des cibles

Étape 1 : enregistrer une cible évolutive

Si vous ne l'avez pas encore fait, enregistrez la cible évolutive. Utilisez la register-scalable-targetcommande pour enregistrer une ressource spécifique dans le service cible en tant que cible évolutive. L'exemple suivant enregistre une demande de parc d'instances Spot avec Application Auto Scaling. Application Auto Scaling peut mettre à l'échelle le nombre d'instances dans le parc d'instances Spot à un minimum de 2 instances et un maximum de 10. Remplacez chaque *`user input placeholder`* par vos propres informations.

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
```

Output

En cas de réussite, cette commande renvoie l'ARN de la cible évolutive. Voici un exemple de sortie.

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

Étape 2 : créer une politique de suivi des objectifs et d'échelonnement

Pour créer une politique de dimensionnement du suivi des cibles, vous pouvez utiliser les exemples suivants pour vous aider à démarrer.

Pour créer une politique de suivi des objectifs et d'échelonnement

1. Utilisez la `cat` commande suivante pour stocker une valeur cible pour votre politique de dimensionnement et une spécification de métrique prédéfinie dans un fichier JSON nommé `config.json` dans votre répertoire de base. Voici un exemple de configuration de suivi des cibles qui maintient l'utilisation moyenne du processeur à 50 %.

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

   Pour plus d'informations, reportez-vous PredefinedMetricSpecificationà la section Application Auto Scaling API Reference.

   Vous pouvez également utiliser une métrique personnalisée pour la mise à l'échelle en créant une spécification de métrique personnalisée et en ajoutant des valeurs pour chaque paramètre à partir de CloudWatch. Voici un exemple de configuration de suivi des cibles qui maintient l'utilisation moyenne de la métrique spécifiée à 100.

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

Pour plus d'informations, reportez-vous CustomizedMetricSpecificationà la section Application Auto Scaling API Reference.

2. Utilisez la commande put-scaling-policy suivante avec le fichier `config.json` que vous avez créé pour créer une stratégie de dimensionnement nommée `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
```

Output

En cas de succès, cette commande renvoie les noms ARNs et des deux CloudWatch alarmes créées en votre nom. Voici un exemple de sortie.

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

Étape 3 : Décrire les politiques de dimensionnement du suivi des cibles

Vous pouvez décrire toutes les stratégies de dimensionnement pour l'espace de noms de service à l'aide de la commande describe-scaling-policies suivante.

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

Vous pouvez filtrer les résultats pour découvrir uniquement les stratégies de dimensionnement Suivi de la cible à l'aide du paramètre `--query`. Pour plus d'informations sur la syntaxe de `query`, consultez Contrôle de la sortie de commande de AWS CLI dans le Guide de l'utilisateur de la 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`]"
```

Output

Voici un exemple de sortie.

```
[
    {
        "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
    }
]
```

Supprimez une politique de dimensionnement du suivi des cibles pour Application Auto Scaling à l'aide du AWS CLI

Lorsque vous avez terminé avec une stratégie de suivi de la cible, vous pouvez la supprimer à l'aide de la commande delete-scaling-policy.

La commande suivante supprime la stratégie de dimensionnement Suivi de la cible indiquée pour la demande de parc d'instances Spot. Il supprime également les CloudWatch alarmes créées en votre nom par Application Auto Scaling.

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
```

Créer une stratégie de mise à l'échelle du suivi des cibles pour Application Auto Scaling à l'aide des mathématiques appliquées aux métriques.

À l'aide des mathématiques métriques, vous pouvez interroger plusieurs CloudWatch métriques et utiliser des expressions mathématiques pour créer de nouvelles séries chronologiques basées sur ces métriques. Vous pouvez visualiser les séries chronologiques obtenues dans la console

CloudWatch et les ajouter à des tableaux de bord. Pour plus d'informations sur les mathématiques métriques, consultez la section Utilisation des mathématiques métriques dans le guide de CloudWatch l'utilisateur Amazon.

Les considérations suivantes s'appliquent aux expressions mathématiques appliquées aux métriques :

- Vous pouvez interroger n'importe quelle CloudWatch métrique disponible. Chaque métrique est une combinaison unique du nom de la métrique, de l'espace de noms et de zéro dimension ou plus.
- Vous pouvez utiliser n'importe quel opérateur arithmétique (+ - */^), fonction statistique (telle que AVG ou SUM) ou toute autre fonction compatible. CloudWatch
- Vous pouvez utiliser à la fois des métriques et les résultats d'autres expressions mathématiques dans les formules de l'expression mathématique.
- Toutes les expressions utilisées dans une spécification de métrique doivent finalement retourner une seule séries temporelles.
- Vous pouvez vérifier la validité d'une expression mathématique métrique à l'aide de la CloudWatch console ou de l' CloudWatch GetMetricDataAPI.

Rubriques

Exemple : file Amazon SQS des éléments en attente par tâche

Pour calculer la file Amazon SQS des éléments en attente par tâche, prenez le nombre approximatif de messages disponibles à la récupération dans la file d'attente et divisez ce nombre par le nombre de tâches Amazon ECS en cours d'exécution dans le service. Pour plus d'informations, consultez Amazon Elastic Container Service (ECS) Auto Scaling using custom metrics sur le AWS Compute Blog.

La logique de l'expression est la suivante :

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

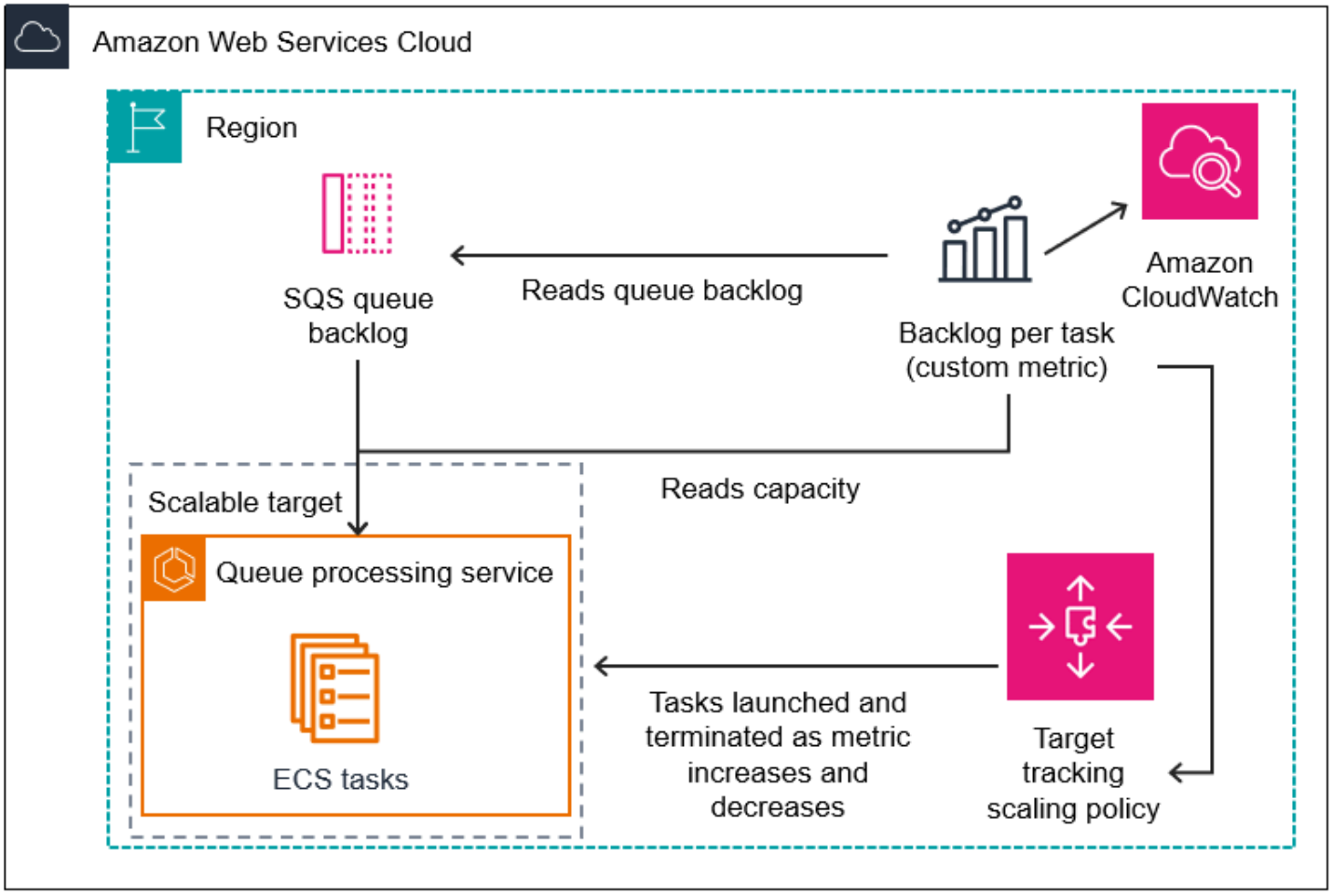
Vos informations CloudWatch métriques sont alors les suivantes.

ID	CloudWatch métrique	Statistique	Period
m1	ApproximateNumberOfMessagesVisible	Somme	1 minute
m2	RunningTaskCount	Moyenne	1 minute

Votre ID de mathématiques appliquées aux métriques et votre expression sont les suivantes :

ID	Expression
e1	`(m1)/(m2)`

Le schéma suivant illustre l'architecture de cette métrique :



Pour utiliser cette expression mathématique appliquée à une métrique pour créer une politique de suivi des cibles (AWS CLI)

1. Stockez l'expression mathématique appliquée aux métriques dans le cadre d'une spécification métrique personnalisée dans un fichier JSON nommé `config.json`.

   Utilisez l'exemple suivant pour vous aider à démarrer. Remplacez chaque *`user input placeholder`* par vos propres informations.

```
{
    "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
}
```

Pour plus d'informations, reportez-vous TargetTrackingScalingPolicyConfigurationà la section Application Auto Scaling API Reference.

> **Note**
>
> Voici quelques ressources supplémentaires qui peuvent vous aider à trouver des noms de métriques, des espaces de noms, des dimensions et des statistiques pour les CloudWatch métriques :
>
> - Pour plus d'informations sur les métriques disponibles pour les AWS services, consultez les AWS services qui publient CloudWatch des métriques dans le guide de CloudWatch l'utilisateur Amazon.
> - Pour obtenir le nom, l'espace de noms et les dimensions exacts (le cas échéant) d'une CloudWatch métrique avec le AWS CLI, consultez list-metrics.

2. Pour créer cette politique, exécutez la put-scaling-policycommande en utilisant le fichier JSON comme entrée, comme illustré dans l'exemple suivant.

```
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
```

En cas de succès, cette commande renvoie le nom de ressource Amazon (ARN) ARNs de la politique et les deux CloudWatch alarmes créées en votre nom.

```
{
    "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**
>
> Si cette commande génère une erreur, assurez-vous d'avoir mis à jour AWS CLI localement la dernière version.

Limitations

- La taille maximum d'une requête est 50 Ko. Il s'agit de la taille totale de la charge utile pour la demande PutScalingPolicyd'API lorsque vous utilisez des mathématiques métriques dans la définition de la politique. Si vous dépassez cette limite, Application Auto Scaling rejette la demande.
- Les services suivants ne sont pas pris en charge lors de l'utilisation des mathématiques appliquées aux métriques avec de politiques de suivi des objectifs de la mise à l'échelle :
  - Amazon Keyspaces (pour Apache Cassandra)
  - DynamoDB
  - Amazon EMR
  - Amazon MSK
  - Amazon Neptune

Politiques de mise à l'échelle par étapes pour Application Auto Scaling

Une politique de dimensionnement par étapes permet d'ajuster la capacité de votre application par incréments prédéfinis en fonction des CloudWatch alarmes. Vous pouvez définir des stratégies de mise à l'échelle distinctes pour gérer la montée en puissance (augmentation de la capacité) et la mise à l'échelle horizontale (diminution de la capacité) en cas de dépassement d'un seuil d'alarme.

Les politiques de dimensionnement par étapes vous permettent de créer et de gérer les CloudWatch alarmes qui déclenchent le processus de dimensionnement. Lorsqu'une alarme est déclenchée, Application Auto Scaling initie la stratégie de mise à l'échelle associée à cette alarme.

La stratégie de mise à l'échelle par étapes permet d'ajuster la capacité à l'aide d'un ensemble d'ajustements, appelés ajustements par étapes. La taille de l'ajustement varie en fonction de l'ampleur du déclenchement de l'alarme.

- Si le déclenchement dépasse le premier seuil, Application Auto Scaling appliquera le premier ajustement par étapes.
- Si le déclenchement dépasse le second seuil, Application Auto Scaling appliquera le deuxième ajustement par étapes, et ainsi de suite.

Cette procédure permet à la statégie de mise à l'échelle de répondre de manière appropriée aux modifications mineures et majeures de la métrique d'alarme.

La stratégie continuera à répondre à d'autres déclenchements d'alarme, même lorsqu'une activité de mise à l'échelle est en cours. Ce qui signifie qu'Application Auto Scaling évaluera tous les déclenchements d'alarme au fur et à mesure qu'ils se produiront. Un temps de stabilisation est utilisé pour éviter une mise à l'échelle démesurée dûe à de multiples déclenchements d'alarme se succèdant rapidement.

Tout comme le suivi des cibles, la mise à l'échelle par étapes peut aider à mettre automatiquement à l'échelle la capacité de votre application en fonction des modifications du trafic. Cependant, les stratégies de suivi des cibles tendent à être plus faciles à mettre en œuvre et à gérer pour des besoins de mise à l'échelle constante.

Cibles évolutives prises en charge

Vous pouvez utiliser des stratégies de mise à l'échelle par étapes avec les cibles évolutives suivantes :

- WorkSpaces Flottes d'applications
- Clusters DB Aurora
- Services ECS
- Clusters EMR
- SageMaker Variantes de terminaux AI
- SageMaker Composants d'inférence AI
- SageMaker Concurrence provisionnée sans serveur AI
- Spot Fleets
- Ressources personnalisées

Table des matières

Comment fonctionne le step scaling pour Application Auto Scaling

Cette rubrique décrit le fonctionnement de la mise à l'échelle des étapes et présente les éléments clés d'une politique de mise à l'échelle des étapes.

Table des matières

- Temps de stabilisation
- Commandes couramment utilisées pour la création, la gestion et la suppression des politiques de mise à l'échelle
- Considérations
- Ressources connexes
- Accès à la console

Comment ça marche

Pour utiliser le dimensionnement par étapes, vous devez créer une CloudWatch alarme qui surveille une métrique pour votre cible évolutive. Définissez la métrique, la valeur de seuil et le nombre de périodes d'évaluation qui déterminent le déclenchement d'une alarme. Vous créez également une stratégie de mise à l'échelle par étapes qui définit comment mettre à l'échelle la capacité lorsque le seuil de déclenchement de l'alarme est dépassé, et vous l'associez à votre cible évolutive.

Ajoutez les ajustements par étapes dans la stratégie. Vous pouvez définir différents ajustements par étapes en fonction de la taille du déclenchement de l'alarme. Par exemple :

- Monter en puissance par 10 unités de capacité si la métrique d'alarme atteint 60 %
- Monter en puissance par 30 unités de capacité si la métrique d'alarme atteint 75 %
- Monter en puissance par 40 unités de capacité si la métrique d'alarme atteint 85 %

Lorsque le seuil d'alarme est dépassé pendant le nombre de périodes d'évaluation spécifié, Application Auto Scaling applique les ajustements par étapes définis dans la stratégie. Les ajustements peuvent se poursuivre pour d'autres déclenchements d'alarme jusqu'à ce que l'état de l'alarme revienne à 0K.

Les activités de mise à l'échelle sont effectuées avec des temps de stabilisation entre eux afin d'éviter des fluctuations rapides de capacité. Vous pouvez éventuellement configurer les temps de stabilisation de votre stratégie de mise à l'échelle.

Ajustements d'étape

Lorsque vous créez une politique de mise à l'échelle par étapes, vous spécifiez un ou plusieurs ajustements par étapes qui redimensionnent automatiquement la capacité de la cible de manière

dynamique en fonction de la taille du seuil de l'alarme. Chaque ajustement par étapes précise ce qui suit :

- Une limite inférieure pour la valeur de la métrique
- Une limite supérieure pour la valeur de la métrique
- L'ampleur de la mise à l'échelle, en fonction du type d'ajustement de la mise à l'échelle

CloudWatch agrège les points de données métriques en fonction de la statistique de la métrique associée à votre CloudWatch alarme. En cas de violation de l'alarme, la stratégie de mise à l'échelle appropriée est appelée. Application Auto Scaling applique le type d'agrégation que vous avez spécifié aux points de données métriques les plus récents CloudWatch (par opposition aux données métriques brutes). Il compare cette valeur de métrique regroupée aux limites supérieures et inférieures définies par les ajustements d'étape afin de déterminer l'ajustement d'étape à réaliser.

Vous spécifiez les limites supérieure et inférieure par rapport au seuil d'une utilisation hors limites. Supposons, par exemple, que vous ayez défini une CloudWatch alarme et une politique de scale-out lorsque la métrique est supérieure à 50 %. Vous avez ensuite défini une deuxième alarme et une stratégie de mise à l'échelle horizontale pour les cas où la métrique est inférieure à 50 %. Vous avez effectué une série d'ajustements par étapes avec un type d'ajustement de `PercentChangeInCapacity` pour chaque stratégie :

Exemple : ajustements par étapes pour la politique d'évolutivité horizontale

Limite inférieure	Limite supérieure	Ajustement
0 USD	10	0 USD
10	20	10
20	null	30

Exemple : ajustements par étapes pour la politique de mise à l'échelle horizontale

Limite inférieure	Limite supérieure	Ajustement
-10	0	0
-20	-10	-10

Limite inférieure	Limite supérieure	Ajustement
null	-20	-30

La configuration de mise à l'échelle suivante est ainsi créée.

```
Metric value

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

Supposons maintenant que vous utilisiez cette configuration de mise à l'échelle sur une cible évolutive ayant une capacité de 10. Les points suivants résument le comportement de la configuration de mise à l'échelle par rapport à la capacité de la cible évolutive :

- La capacité d'origine est conservée, tandis que la valeur métrique cumulée est supérieure à 40 et inférieure à 60.
- Si la valeur métrique atteint 60, Application Auto Scaling augmente la capacité de la cible évolutive de 1, pour la porter à 11. Cette opération est effectuée en fonction du deuxième ajustement d'étape de la stratégie d'augmentation de la taille des instances (ajouter 10 % de 10). Une fois la nouvelle capacité ajoutée, Application Auto Scaling augment la capacité actuelle à 11. Si la valeur de la métrique atteint 70 même après cette augmentation de capacité, Application Auto Scaling augmente la capacité cible de 3, pour la porter à 14. Cette opération est effectuée en fonction du troisième ajustement d'étape de la stratégie d'augmentation de la taille des instances (ajouter 30 % de 11, 3,3, arrondis à 3).
- Si la valeur métrique atteint 40, Application Auto Scaling diminue la capacité de la cible évolutive de 1, pour la porter à 13, en fonction du deuxième ajustement d'étape de la politique de mise à l'échelle horizontale (enlevez 10 pour cent de 14, soit 1,4, arrondi à 1). Si la valeur métrique tombe à 30 même après sa baisse de capacité, Application Auto Scaling diminue la capacité cible de 3 pour atteindre 10, en fonction du troisième ajustement d'étape de la politique de mise à l'échelle horizontale des instances (supprimez 30 pour cent de 13, 3.9 arrondis à 3).

Lorsque vous spécifiez les ajustements d'étape pour votre politique de mise à l'échelle, notez les points suivants :

- Les plages d'ajustements d'étape peuvent se chevaucher ou avoir un écart.
- Un seul ajustement d'étape peut avoir une limite inférieure null (infini négatif). Si un seul ajustement d'étape possède une limite inférieure négative, un ajustement d'étape avec une limite inférieure null doit donc exister.
- Un seul ajustement d'étape peut avoir une limite supérieure null (infini positif). Si un seul ajustement d'étape possède une limite supérieure positive, un ajustement d'étape avec une limite supérieure null doit donc exister.
- Les limites supérieure et inférieure ne peuvent pas être null dans le même ajustement d'étape.
- Si la valeur métrique dépasse le seuil, la limite inférieure est inclusive et la limite supérieure est exclusive. Si la valeur métrique n'atteint pas le seuil, la limite inférieure est exclusive et la limite supérieure est inclusive.

Types d'ajustement de la mise à l'échelle

Vous pouvez définir une politique de mise à l'échelle qui exécute l'action de mise à l'échelle optimale, en fonction du type d'ajustement de mise à l'échelle que vous choisissez. Vous pouvez spécifier le type d'ajustement sous la forme d'un pourcentage de la capacité actuelle de votre cible évolutive ou en tant que nombres absolus.

Application Auto Scaling prend en charge les types d'ajustement suivants pour les politiques de mise à l'échelle par étapes :

- ChangeInCapacity: augmente ou diminue la capacité actuelle de la cible évolutive selon la valeur spécifiée. Une valeur positive augmente la capacité et une valeur négative la réduit. Par exemple : si la capacité actuelle est de 3 et l'ajustement de 5, Application Auto Scaling ajoute 5 à la capacité pour un total de 8.
- ExactCapacity—Remplacez la capacité actuelle de la cible évolutive par la valeur spécifiée. Spécifiez une valeur non négative avec ce type d'ajustement. Par exemple : si la capacité actuelle est de 3 et l'ajustement de 5, Application Auto Scaling fait passer la capacité à 5.
- PercentChangeInCapacity: augmente ou diminue la capacité actuelle de la cible évolutive du pourcentage spécifié. Une valeur positive augmente la capacité et une valeur négative la réduit. Par exemple : si la capacité actuelle est de 10 et l'ajustement de 10 pour cent, Application Auto Scaling ajoute 1 à la capacité pour un total de 11.

  Si la valeur générée n'est pas un nombre entier, Application Auto Scaling l'arrondit comme suit :

- Les valeurs supérieures à 1 sont arrondies à l'unité inférieure. Par exemple, `12.7` est arrondi à `12`.
- Les valeurs comprises entre 0 et 1 sont arrondies à 1. Par exemple, `.67` est arrondi à `1`.
- Les valeurs comprises entre 0 et -1 sont arrondies à -1. Par exemple, `-.58` est arrondi à `-1`.
- Les valeurs inférieures à -1 sont arrondies à l'unité supérieure. Par exemple, `-6.67` est arrondi à `-6`.

Avec PercentChangeInCapacity, vous pouvez également spécifier le montant minimum à redimensionner à l'aide du `MinAdjustmentMagnitude` paramètre. Par exemple, imaginons que vous ayez créé une stratégie qui ajoute 25 % et que vous spécifiiez montant minimum de 2. Si la cible évolutive a une capacité de 4 et que la stratégie de dimensionnement est réalisée, 25 pour cent de 4 est égal à 1. Cependant, comme vous avez spécifié un incrément minimal de 2, Application Auto Scaling ajoute 2.

Temps de stabilisation

Vous pouvez éventuellement définir un temps de stabilisation dans votre politique de mise à l'échelle par étape.

Le temps de stabilisation spécifie la durée pendant laquelle la politique de mise à l'échelle attend qu'une activité de mise à l'échelle précédente prenne effet.

Il existe deux manières de planifier l'utilisation de temps de stabilisation pour une configuration de mise à l'échelle par étape :

- Avec le temps de stabilisation pour les politiques de montée en puissance parallèle, l'intention est de monter continuellement en puissance (mais sans excès). Une fois qu'Application Auto Scaling a réussi une montée en puissance à l'aide d'une politique de mise à l'échelle par étape, l'application commence à calculer le temps de stabilisation. La politique de mise à l'échelle n'augmente pas à nouveau la capacité souhaitée, sauf si une plus grande montée en puissance parallèle est déclenchée ou si le temps de stabilisation est écoulé. Tandis que le temps de stabilisation de la montée en puissance s'applique, la capacité ajoutée par l'activité de mise à l'échelle initiale est calculée dans le cadre de la capacité souhaitée pour la prochaine activité de montée en puissance.
- Avec le temps de stabilisation pour les politiques de mise à l'échelle horizontale, l'intention est de procéder à une mise à l'échelle horizontale de façon conservatrice pour protéger la disponibilité de votre application, de sorte que les activités de mise à l'échelle horizontale sont bloquées jusqu'à ce que le temps de stabilisation de la mise à l'échelle horizontale ait expiré. Toutefois, si une

autre alarme déclenche une activité de montée en charge au cours du temps de stabilisation de la diminution de charge, Application Auto Scaling monte immédiatement en charge la cible. Dans ce cas, le temps de stabilisation de la mise à l'échelle horizontale s'arrête et ne se termine pas.

Par exemple, lorsqu'un pic de trafic se produit, une alarme est déclenchée et Application Auto Scaling ajoute automatiquement de la capacité pour aider à gérer la charge accrue. Si vous définissez un temps de stabilisation pour votre politique de montée en puissance parallèle, lorsque l'alarme déclenche la politique pour augmenter la capacité de 2, l'activité de mise à l'échelle s'achève avec succès et le temps de stabilisation de montée en puissance parallèle démarre. Si l'alarme se déclenche encore pendant le temps de stabilisation mais à un ajustement d'étape plus agressif de 3, l'augmentation précédente de 2 est considérée comme intégrée à la capacité actuelle. Ainsi, uniquement 1 est ajouté à la capacité. Cela permet une mise à l'échelle plus rapide par rapport à l'attente de l'expiration du temps de stabilisation, sans pour autant augmenter la capacité nécessaire.

Le temps de stabilisation est mesuré en secondes et s'applique uniquement aux activités de mise à l'échelle liées à la stratégie de mise à l'échelle. Pendant un temps de stabilisation, lorsqu'une action planifiée démarre à l'heure prévue, elle peut déclencher une activité de mise à l'échelle immédiatement sans attendre l'expiration du temps de stabilisation.

La valeur par défaut est 300 si aucune valeur n'est spécifiée.

Commandes couramment utilisées pour la création, la gestion et la suppression des politiques de mise à l'échelle

Les commandes couramment utilisées pour travailler avec les politiques de mise à l'échelle sont les suivantes :

- register-scalable-targetpour enregistrer AWS ou personnaliser des ressources en tant que cibles évolutives (ressource qu'Application Auto Scaling peut redimensionner), et pour suspendre et reprendre le dimensionnement.
- put-scaling-policypour ajouter ou modifier des politiques de dimensionnement pour une cible évolutive existante.
- describe-scaling-activitiespour renvoyer des informations sur les activités de mise à l'échelle dans une AWS région.
- describe-scaling-policiespour renvoyer des informations sur les politiques de dimensionnement dans une AWS région.
- delete-scaling-policypour supprimer une politique de dimensionnement.

Considérations

Les considérations suivantes s'appliquent lors de l'utilisation de politiques de mise à l'échelle par étapes :

- Déterminez si vous pouvez prédire les ajustements d'étape sur l'application avec suffisamment de précision pour utiliser la mise à l'échelle par étapes. Si votre métrique de mise à l'échelle augmente ou diminue proportionnellement à la capacité de la cible évolutive, nous vous recommandons d'utiliser plutôt une politique de suivi des cibles et de mise à l'échelle. Vous avez toujours la possibilité d'utiliser la mise à l'échelle par étapes comme politique supplémentaire pour une configuration plus avancée. Par exemple, si vous le souhaitez, vous pouvez configurer une réponse plus agressive lorsque l'utilisation atteint un certain niveau.
- Assurez-vous de choisir une marge adéquate entre les seuils de mise à l'échelle horizontale et de mise à l'échelle avec montée en puissance parallèle, afin d'éviter tout battement. Le battement est une boucle infinie de mise à l'échelle horizontale et de montage en puissance. En d'autres termes, si une action de mise à l'échelle est effectuée, la valeur de la métrique changera et déclenchera une autre action de mise à l'échelle dans le sens inverse.

Ressources connexes

Pour plus d'informations sur la création de politiques de dimensionnement par étapes pour les groupes Auto Scaling, consultez la section Step and Simple Scaling policies for Amazon EC2 Auto Scaling dans le guide de l'utilisateur Amazon EC2 Auto Scaling.

Accès à la console

L'accès à la console pour consulter, ajouter, mettre à jour ou supprimer les politiques de mise à l'échelle par étapes sur les ressources évolutives dépend de la ressource que vous utilisez. Pour de plus amples informations, veuillez consulter Services AWS que vous pouvez utiliser avec Application Auto Scaling.

Créez une politique de dimensionnement par étapes pour Application Auto Scaling à l'aide du AWS CLI

Cet exemple utilise des AWS CLI commandes pour créer une politique de dimensionnement par étapes pour un service Amazon ECS. Pour une autre cible évolutive, spécifiez son espace de noms

dans`--service-namespace`, sa dimension évolutive dans `--scalable-dimension` et son ID de ressource dans`--resource-id`.

Lorsque vous utilisez le AWS CLI, n'oubliez pas que vos commandes s'exécutent dans la Région AWS configuration adaptée à votre profil. Si vous souhaitez exécuter les commandes dans une autre région, modifiez la région par défaut pour votre profil, ou utilisez le paramètre `--region` avec la commande.

Tâches

- Étape 1 : enregistrer une cible évolutive
- Étape 2 : Création d'une politique de dimensionnement par étapes
- Étape 3 : créer une alarme qui invoque une politique de dimensionnement

Étape 1 : enregistrer une cible évolutive

Si vous ne l'avez pas encore fait, enregistrez la cible évolutive. Utilisez la register-scalable-targetcommande pour enregistrer une ressource spécifique dans le service cible en tant que cible évolutive. L'exemple suivant enregistre un service Amazon ECS avec Application Auto Scaling. Application Auto Scaling peut ajuster le nombre de tâches avec un minimum de 2 tâches et un maximum de 10 tâches. Remplacez chaque *`user input placeholder`* par vos propres informations.

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
```

Output

En cas de réussite, cette commande renvoie l'ARN de la cible évolutive. Voici un exemple de sortie.

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

Étape 2 : Création d'une politique de dimensionnement par étapes

Pour créer une politique d'échelonnement pour votre cible évolutive, vous pouvez utiliser les exemples suivants pour vous aider à démarrer.

Scale out

Pour créer une politique de mise à l'échelle progressive pour le scalage (augmentation de la capacité)

1. Utilisez la `cat` commande suivante pour enregistrer une configuration de politique de dimensionnement par étapes dans un fichier JSON nommé `config.json` dans votre répertoire de base. Voici un exemple de configuration avec un type de réglage `PercentChangeInCapacity` qui augmente la capacité de la cible évolutive en fonction des ajustements par étapes suivants (en supposant un seuil CloudWatch d'alarme de 70) :

   - Augmenter la capacité de 10 % lorsque la valeur de la métrique est supérieure ou égale à 70 mais inférieure à 85
   - Augmenter la capacité de 20 % lorsque la valeur de la métrique est supérieure ou égale à 85 mais inférieure à 95
   - Augmenter la capacité de 30 % lorsque la valeur de la métrique est supérieure ou égale à 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
    }
  ]
}
```

Pour plus d'informations, reportez-vous StepScalingPolicyConfigurationà la section Application Auto Scaling API Reference.

2. Utilisez la put-scaling-policycommande suivante, ainsi que le `config.json` fichier que vous avez créé, pour créer une politique de dimensionnement nomméemy-`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
   ```

   Output

   Le résultat comprend l'ARN qui sert de nom unique pour la stratégie. Vous en avez besoin pour créer une CloudWatch alarme pour votre police. Voici un exemple de sortie.

```
{
    "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

Pour créer une politique d'échelonnement à des fins d'évolutivité (diminution de la capacité)

1. Utilisez la `cat` commande suivante pour enregistrer une configuration de politique de dimensionnement par étapes dans un fichier JSON nommé `config.json` dans votre répertoire de base. Voici un exemple de configuration avec un type de réglage `ChangeInCapacity` qui réduit la capacité de la cible évolutive en fonction des ajustements par étapes suivants (en supposant un seuil CloudWatch d'alarme de 50) :

   - Diminuez la capacité de 1 lorsque la valeur de la métrique est inférieure ou égale à 50 mais supérieure à 40
   - Diminuez la capacité de 2 lorsque la valeur de la métrique est inférieure ou égale à 40 mais supérieure à 30
   - Diminuez la capacité de 3 lorsque la valeur de la métrique est inférieure ou égale à 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
    }
  ]
}
```

Pour plus d'informations, reportez-vous StepScalingPolicyConfigurationà la section Application Auto Scaling API Reference.

2. Utilisez la put-scaling-policycommande suivante, ainsi que le `config.json` fichier que vous avez créé, pour créer une politique de dimensionnement nomméemy-`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
   ```

   Output

   Le résultat comprend l'ARN qui sert de nom unique pour la stratégie. Vous avez besoin de cet ARN pour créer une CloudWatch alarme pour votre politique. Voici un exemple de sortie.

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

```
}
```

Étape 3 : créer une alarme qui invoque une politique de dimensionnement

Enfin, utilisez la CloudWatch put-metric-alarmcommande suivante pour créer une alarme à utiliser avec votre politique de dimensionnement des étapes. Dans cet exemple, nous utilisons une alarme basée sur l'utilisation moyenne de l'UC. L'alarme est configurée pour être dans un état ALARM si elle atteint un seuil de 70 % pendant au moins deux périodes d'évaluation consécutives de 60 secondes. Pour spécifier une autre CloudWatch métrique ou utiliser votre propre métrique personnalisée, spécifiez son nom dans `--metric-name` et son espace de noms dans`--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
```

Décrire les politiques de dimensionnement par étapes pour Application Auto Scaling à l'aide du AWS CLI

Vous pouvez décrire toutes les politiques de dimensionnement pour un espace de noms de service à l'aide de la describe-scaling-policiescommande. L'exemple suivant décrit toutes les politiques de

dimensionnement pour tous les services Amazon ECS. Pour les répertorier pour un service Amazon ECS spécifique, ajoutez uniquement l'`--resource-id`option.

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

Vous pouvez filtrer les résultats pour découvrir uniquement les stratégies de dimensionnement d'étape à l'aide du paramètre `--query`. Pour plus d'informations sur la syntaxe de `query`, consultez Contrôle de la sortie de commande de AWS CLI dans le Guide de l'utilisateur de la 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`]"
```

Output

Voici un exemple de sortie.

```
[
    {
        "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
    }
]
```

Supprimez une politique de dimensionnement par étapes pour Application Auto Scaling à l'aide du AWS CLI

Lorsque vous n'avez plus besoin d'une stratégie de dimensionnement d'étape, vous pouvez la supprimer. Pour supprimer à la fois la politique de dimensionnement et l' CloudWatch alarme associée, effectuez les tâches suivantes.

Pour supprimer une stratégie de dimensionnement

Utilisez la commande 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
```

Pour supprimer l' CloudWatch alarme

Utilisez la commande delete-alarms suivante. Vous pouvez supprimer une ou plusieurs alarmes en même temps. Par exemple, utilisez la commande suivante pour supprimer les alarmes `Step-Scaling-AlarmHigh-ECS:service/my-cluster/my-service` et `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
```

Scalage prédictif pour Application Auto Scaling

La mise à l'échelle prédictive fait évoluer votre application de manière proactive. La mise à l'échelle prédictive analyse les données de charge historiques pour détecter les tendances quotidiennes ou hebdomadaires des flux de trafic. Il utilise ces informations pour prévoir les futurs besoins de capacité afin d'augmenter de manière proactive la capacité de votre application pour qu'elle corresponde à la charge prévue.

La mise à l'échelle prédictive se prête particulièrement bien aux situations suivantes :

- Trafic cyclique, tel qu'une utilisation intensive des ressources pendant les heures de bureau et une faible utilisation le soir et le week-end
- Modèles on-and-off de charge de travail récurrents, tels que le traitement par lots, les tests ou l'analyse périodique des données.
- Applications dont l'initialisation prend beaucoup de temps, ce qui en termes de performances se traduit par une latence notable lors des événements de montée en puissance

Table des matières

Comment fonctionne la mise à l'échelle prédictive d'Application Auto Scaling

Pour utiliser la mise à l'échelle prédictive, créez une politique de mise à l'échelle prédictive qui spécifie la CloudWatch métrique à surveiller et à analyser. Vous pouvez utiliser une métrique prédéfinie ou une métrique personnalisée. Pour que la mise à l'échelle prédictive commence à prévoir les valeurs futures, cette métrique doit disposer d'au moins 24 heures de données.

Une fois que vous avez créé la stratégie, la mise à l'échelle prédictive commence à analyser les données métriques recueillies au cours des 14 derniers jours afin d'identifier des modèles. Il utilise cette analyse pour générer une prévision horaire des besoins de capacité pour les 48 prochaines

heures. Les prévisions sont mises à jour toutes les 6 heures en utilisant les dernières CloudWatch données. À mesure que de nouvelles données arrivent, la mise à l'échelle prédictive est en mesure d'améliorer en permanence la précision des prévisions futures.

Vous pouvez d'abord activer la mise à l'échelle prédictive en mode prévision uniquement. Dans ce mode, il génère des prévisions de capacité mais n'adapte pas réellement votre capacité en fonction de ces prévisions. Cela vous permet d'évaluer la précision et la pertinence des prévisions.

Après avoir examiné les données de prévision et décidé de commencer le dimensionnement en fonction de ces données, passez la politique de dimensionnement en mode prévision et échelle. Dans ce mode :

- Si les prévisions prévoient une augmentation de la charge, la mise à l'échelle prédictive augmentera la capacité.
- Si les prévisions prévoient une diminution de la charge, la mise à l'échelle prédictive ne sera pas adaptée pour réduire la capacité. Cela garantit que vous n'intervenez que lorsque la demande baisse réellement, et pas uniquement en fonction des prévisions. Pour supprimer la capacité qui n'est plus nécessaire, vous devez créer une politique de suivi des cibles ou de mise à l'échelle des étapes, car elles répondent aux données métriques en temps réel.

Par défaut, la mise à l'échelle prédictive redimensionne vos objectifs évolutifs au début de chaque heure en fonction des prévisions pour cette heure. Vous pouvez éventuellement spécifier une heure de début antérieure en utilisant la `SchedulingBufferTime` propriété dans l'opération `PutScalingPolicy` d'API. Cela vous permet de lancer la capacité prévue avant la demande prévue, ce qui donne à la nouvelle capacité le temps nécessaire pour être prête à gérer le trafic.

Limite de capacité maximale

Par défaut, lorsque des politiques de dimensionnement sont définies, elles ne peuvent pas augmenter la capacité au-delà de sa capacité maximale.

Vous pouvez également autoriser l'augmentation automatique de la capacité maximale de la cible évolutive si la capacité prévue approche ou dépasse la capacité maximale de la cible évolutive. Pour activer ce comportement, utilisez les `MaxCapacityBuffer` propriétés `MaxCapacityBreachBehavior` et dans le fonctionnement de l'`PutScalingPolicy`API ou le paramètre de comportement de capacité maximale dans le AWS Management Console.

> **Warning**
>
> Soyez prudent lorsque vous autorisez l'augmentation automatique de la capacité maximale. La capacité maximale ne diminue pas automatiquement pour revenir à la capacité maximale initiale.

Commandes couramment utilisées pour la création, la gestion et la suppression des politiques de mise à l'échelle

Les commandes couramment utilisées pour travailler avec les politiques de dimensionnement prédictif sont les suivantes :

- `register-scalable-target`pour enregistrer AWS ou personnaliser des ressources en tant que cibles évolutives, pour suspendre le dimensionnement et pour reprendre le dimensionnement.
- `put-scaling-policy`pour créer une politique de dimensionnement prédictive.
- `get-predictive-scaling-forecast`pour récupérer les données de prévision pour une politique de mise à l'échelle prédictive.
- `describe-scaling-activities`pour renvoyer des informations sur les activités de dimensionnement dans un Région AWS.
- `describe-scaling-policies`pour renvoyer des informations sur les politiques de dimensionnement dans un Région AWS.
- `delete-scaling-policy`pour supprimer une politique de dimensionnement.

Métriques personnalisées

Des métriques personnalisées peuvent être utilisées pour prévoir la capacité requise pour une application. Les métriques personnalisées sont utiles lorsque les métriques prédéfinies ne sont pas suffisantes pour capturer la charge de votre application.

Considérations

Les considérations suivantes s'appliquent lors de l'utilisation de la mise à l'échelle prédictive.

- Vérifiez si la mise à l'échelle prédictive est adaptée à votre application. Une application convient parfaitement à la mise à l'échelle prédictive si elle présente des modèles de charge récurrents

spécifiques au jour de la semaine ou à l'heure de la journée. Évaluez les prévisions avant de laisser le dimensionnement prédictif faire évoluer activement votre application.

- La mise à l'échelle prédictive requiert au moins 24 heures de données historiques pour commencer à élaborer des prévisions. Toutefois, les prévisions sont plus efficaces si les données historiques couvrent deux semaines complètes.
- Choisissez une métrique de charge qui représente avec précision la charge complète de votre application et qui constitue l'aspect le plus important de votre application à prendre en compte.

Création d'une politique de dimensionnement prédictive pour Application Auto Scaling

L'exemple de politique suivant utilise le AWS CLI pour configurer une politique de dimensionnement prédictif pour le service Amazon ECS. Remplacez chaque *user input placeholder* par vos propres informations.

Pour plus d'informations sur les CloudWatch métriques que vous pouvez spécifier, consultez PredictiveScalingMetricSpecificationle Amazon EC2 Auto Scaling API Reference.

Voici un exemple de stratégie avec une configuration mémoire prédéfinie.

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

L'exemple suivant illustre la création de la politique en exécutant la put-scaling-policycommande avec le fichier de configuration spécifié.

```
aws aas put-scaling-policy \
```

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

Si elle aboutit, cette commande renvoie l'ARN de la stratégie.

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

Remplacer des valeurs de prévision à l'aide d'actions planifiées

Parfois, vous pouvez disposer d'informations supplémentaires sur de futurs besoins de votre application que le calcul prédictif ne peut pas prendre en compte. Par exemple, les calculs prédictifs peuvent sous-estimer la capacité nécessaire pour un événement marketing à venir. Vous pouvez alors utiliser des actions planifiées pour remplacer temporairement la prévision au cours de périodes ultérieures. Les actions planifiées peuvent être exécutées de manière récurrente, ou à une date et une heure spécifiques en cas de fluctuations ponctuelles de la demande.

Par exemple, vous pouvez créer une action planifiée avec une capacité minimale plus élevée que ce qui est prédit. Au moment de l'exécution, Application Auto Scaling met à jour la capacité minimale de votre cible évolutive. Étant donné que la mise à l'échelle prédictive optimise la capacité, une action planifiée avec une capacité minimale supérieure aux valeurs prédites est honorée. Cela permet d'éviter que la capacité soit inférieure à celle prévue. Pour cesser de remplacer la prévision, utilisez une deuxième action planifiée afin de rétablir le paramètre d'origine de la capacité minimale.

La procédure suivante présente les étapes à suivre pour remplacer la prévision au cours de périodes ultérieures.

Rubriques

> **Important**
>
> Cette rubrique part du principe que vous essayez de remplacer les prévisions afin d'atteindre une capacité supérieure à celle prévue. Si vous devez réduire temporairement la capacité sans interférer avec une politique de dimensionnement prédictif, utilisez plutôt le mode prévision uniquement. En mode prévisions uniquement, la mise à l'échelle prédictive continuera de générer des prévisions, mais elle n'augmentera pas automatiquement la capacité. Vous pouvez ensuite surveiller l'utilisation des ressources et réduire manuellement la taille de votre groupe selon vos besoins.

Étape 1 : (facultatif) Analyser les données en séries chronologiques

Commencez par analyser les données en séries chronologiques de la prévision. Il s'agit d'une étape facultative, mais elle permet de comprendre les détails de la prévision.

1. Récupérer la prévision

   Une fois la prévision créée, vous pouvez interroger une période spécifique au sein de celle-ci. L'objectif de la requête est d'obtenir une vue complète des données en séries chronologiques d'une période spécifique.

   Votre requête peut inclure jusqu'à deux jours de données de prévision ultérieures. Si vous utilisez la mise à l'échelle prédictive depuis un certain temps, vous pouvez également accéder à vos données de prévision antérieures. Toutefois, la durée maximale entre le début et la fin est de 30 jours.

   Pour récupérer les prévisions, utilisez la get-predictive-scaling-forecastcommande. L'exemple suivant permet d'obtenir les prévisions de dimensionnement prédictif pour le service Amazon ECS.

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

La réponse inclut deux prévisions : `LoadForecast` et`CapacityForecast`. `LoadForecast`affiche les prévisions de charge horaire. `CapacityForecast`affiche les valeurs de prévision de la capacité nécessaire sur une base horaire pour gérer la charge prévue tout en maintenant une valeur spécifiée`TargetValue`.

2. Identifier la période cible

   Indiquez l'heure ou les heures où la fluctuation de la demande ponctuelle devrait avoir lieu. N'oubliez pas que les dates et les heures indiquées dans la prévision sont basées sur le fuseau horaire UTC.

Étape 2 : créer deux actions planifiées

Créez ensuite deux actions planifiées pour une période spécifique où votre application devra gérer une charge plus élevée que celle prédite. Par exemple, si vous organisez un événement marketing qui va générer du trafic sur votre site pendant une période limitée, vous pouvez planifier une action ponctuelle pour mettre à jour la capacité minimale au début de cet événement. Puis vous pouvez planifier une autre action pour rétablir le paramètre d'origine de la capacité minimale à la fin de l'événement.

Pour créer deux actions planifiées pour des événements ponctuels (AWS CLI)

Pour créer les actions planifiées, utilisez la put-scheduled-actioncommande.

L'exemple suivant définit un calendrier pour Amazon EC2 Auto Scaling qui maintient une capacité minimale de trois instances le 19 mai à 17 h 00 pendant huit heures. Les commandes suivantes montrent comment implémenter ce scénario.

La première commande put-scheduled-update-group-action demande à Amazon EC2 Auto Scaling de mettre à jour la capacité minimale du groupe Auto Scaling spécifié à 17 h 00 UTC le 19 mai 2021.

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

La deuxième commande demande à Amazon EC2 Auto Scaling de définir la capacité minimale du groupe à 1 h 00 UTC le 20 mai 2021.

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

Après avoir ajouté ces actions planifiées au groupe Auto Scaling, Amazon EC2 Auto Scaling effectue les opérations suivantes :

- À 17h00 UTC le 19 mai 2021, la première action planifiée s'exécute. Si le groupe compte actuellement moins de trois instances, il passe à trois instances. Pendant cette période et pendant les huit prochaines heures, Amazon EC2 Auto Scaling peut continuer à évoluer si la capacité prévue est supérieure à la capacité réelle ou si une politique de dimensionnement dynamique est en vigueur.
- À 1h00 UTC le 20 mai 2021, la seconde action planifiée s'exécute. Cette action rétablit le paramètre d'origine de la capacité minimale à la fin de l'événement.

Mise à l'échelle basée sur des planifications récurrentes

Pour remplacer la prévision applicable à la même période chaque semaine, créez deux actions planifiées et fournissez la logique d'heure et de date à l'aide d'une expression cron.

L'expression cron est constituée de cinq champs séparés par des espaces : [Minute] [Heure] [Jour_du_Mois] [Mois_de_Année] [Jour_de_Semaine]. Ces champs peuvent contenir toutes les valeurs autorisées, y compris des caractères spéciaux.

Par exemple, l'expression cron suivante exécute l'action tous les mardis à 6h30. L'astérisque est utilisé comme caractère générique pour représenter toutes les valeurs d'un champ.

```
30 6 * * 2
```

Politique de dimensionnement prédictive avancée utilisant des métriques personnalisées

Dans une politique de mise à l'échelle prédictive, vous pouvez utiliser des métriques prédéfinies ou personnalisées. Les métriques personnalisées sont utiles lorsque les métriques prédéfinies ne décrivent pas suffisamment la charge de votre application.

Lorsque vous créez une politique de dimensionnement prédictif avec des métriques personnalisées, vous pouvez spécifier d'autres CloudWatch métriques fournies par AWS, ou vous pouvez spécifier des métriques que vous définissez et publiez vous-même. Vous pouvez également utiliser les mathématiques des métriques pour agréger et transformer les métriques existantes en une nouvelle série chronologique qui AWS n'est pas automatiquement suivie. Lorsque vous combinez des valeurs dans vos données, par exemple, en calculant de nouvelles sommes ou moyennes, cela s'appelle l'agrégation. Les données résultantes sont appelées un agrégat.

La section suivante contient les bonnes pratiques et des exemples de construction de la structure JSON pour la politique.

Rubriques

Bonnes pratiques

Les bonnes pratiques suivantes peuvent vous aider à utiliser plus efficacement les métriques personnalisées :

- Pour la spécification de la métrique de charge, la métrique la plus utile est une métrique qui représente la charge sur votre application.
- La métrique de mise à l'échelle doit être inversement proportionnelle à la capacité. En d'autres termes, si la cible évolutive augmente, la métrique de mise à l'échelle devrait diminuer à peu près dans la même proportion. Pour que la mise à l'échelle prédictive se comporte comme prévu, la métrique de charge et la métrique de mise à l'échelle doivent également présenter une forte corrélation entre elles.
- L'utilisation cible doit correspondre au type de métrique de mise à l'échelle. Pour une configuration de politique qui utilise l'utilisation du CPU, il s'agit d'un pourcentage cible. Pour une configuration de politique qui utilise le débit, tel que le nombre de demandes ou de messages, il s'agit du nombre cible de demandes ou de messages par instance pendant tout intervalle d'une minute.
- Si ces recommandations ne sont pas suivies, les valeurs futures prédites des séries temporelles seront probablement incorrectes. Pour vérifier que les données sont correctes, vous pouvez

consulter les valeurs prévisionnelles. Sinon, après avoir créé votre politique de dimensionnement prédictif, inspectez les `CapacityForecast` objets `LoadForecast` et renvoyés par un appel à l'GetPredictiveScalingForecastAPI.

- Nous vous recommandons vivement de configurer la mise à l'échelle prédictive en mode prévision uniquement pour pouvoir évaluer la prévision avant que la mise à l'échelle prédictive ne commence à mettre activement à l'échelle la capacité.

Conditions préalables

Pour ajouter des métriques personnalisées à votre politique de mise à l'échelle, vous devez disposer des autorisations `cloudwatch:GetMetricData`.

Pour spécifier vos propres indicateurs au lieu des indicateurs AWS fournis, vous devez d'abord les publier sur CloudWatch. Pour plus d'informations, consultez la section Publication de métriques personnalisées dans le guide de CloudWatch l'utilisateur Amazon.

Si vous publiez vos propres métriques, veillez à publier les points de données à une fréquence minimale de cinq minutes. Application Auto Scaling extrait les points de données CloudWatch en fonction de la durée de la période dont elle a besoin. Par exemple, la spécification des métriques de charge utilise des métriques horaires pour mesurer la charge de votre application. CloudWatch utilise vos données métriques publiées pour fournir une valeur de données unique pour toute période d'une heure en agrégeant tous les points de données avec des horodatages correspondant à chaque période d'une heure.

Construction du fichier JSON pour les métriques personnalisées

La section suivante contient des exemples expliquant comment configurer le dimensionnement prédictif pour interroger des données CloudWatch destinées à Amazon EC2 Auto Scaling. Il existe deux méthodes pour configurer cette option, qui affecteront le format utilisé pour créer le fichier JSON de votre politique de mise à l'échelle prédictive. Lorsque vous utilisez des mathématiques de métriques, le format du fichier JSON varie davantage en fonction des mathématiques de métriques effectuées.

1. Pour créer une politique qui obtient des données directement à partir d'autres CloudWatch indicateurs fournis par AWS ou sur lesquels vous publiez CloudWatch, voirExemple de politique de mise à l'échelle prédictive avec des métriques de charge et de mise à l'échelle personnalisées (AWS CLI).

2. Pour créer une politique capable d'interroger plusieurs CloudWatch mesures et d'utiliser des expressions mathématiques pour créer de nouvelles séries chronologiques basées sur ces mesures, voirUtiliser des expressions mathématiques de métrique.

Exemple de politique de mise à l'échelle prédictive avec des métriques de charge et de mise à l'échelle personnalisées (AWS CLI)

Pour créer une politique de dimensionnement prédictive avec des métriques de charge et de dimensionnement personnalisées avec le AWS CLI, stockez les arguments pour `--predictive-scaling-configuration` dans un fichier JSON nomméconfig.json.

Vous commencez par ajouter des métriques personnalisées en remplaçant les valeurs remplaçables de l'exemple suivant par celles de vos métriques et de votre utilisation cible.

```
{
  "MetricSpecifications": [
    {
      "TargetValue": 50,
      "CustomizedScalingMetricSpecification": {
        "MetricDataQueries": [
          {
            "Id": "scaling_metric",
            "MetricStat": {
              "Metric": {
                "MetricName": "MyUtilizationMetric",
                "Namespace": "MyNameSpace",
                "Dimensions": [
                  {
                    "Name": "MyOptionalMetricDimensionName",
                    "Value": "MyOptionalMetricDimensionValue"
                  }
                ]
              },
              "Stat": "Average"
            }
          }
        ]
      },
      "CustomizedLoadMetricSpecification": {
        "MetricDataQueries": [
          {
```

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

Pour plus d'informations, consultez MetricDataQueryle manuel Amazon EC2 Auto Scaling API Reference.

> **Note**
>
> Voici quelques ressources supplémentaires qui peuvent vous aider à trouver des noms de métriques, des espaces de noms, des dimensions et des statistiques pour les CloudWatch métriques :
>
> - Pour plus d'informations sur les métriques disponibles pour les AWS services, consultez les AWS services qui publient CloudWatch des métriques dans le guide de CloudWatch l'utilisateur Amazon.
> - Pour obtenir le nom, l'espace de noms et les dimensions exacts (le cas échéant) d'une CloudWatch métrique avec le AWS CLI, consultez list-metrics.

Pour créer cette politique, exécutez la put-scaling-policycommande en utilisant le fichier JSON comme entrée, comme illustré dans l'exemple suivant.

```
aws autoscaling put-scaling-policy --policy-name my-predictive-scaling-policy \
```

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

Si elle aboutit, cette commande renvoie l'Amazon Resource Name (ARN) de la stratégie.

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

Utiliser des expressions mathématiques de métrique

La section suivante fournit des informations et des exemples de politiques de mise à l'échelle prédictive qui montrent comment vous pouvez utiliser les mathématiques de métriques dans votre politique.

Rubriques

- Comprendre les mathématiques de métrique
- Exemple de politique de dimensionnement prédictif pour Amazon EC2 Auto Scaling qui combine les métriques à l'aide des mathématiques métriques (AWS CLI)
- Exemple de politique de dimensionnement prédictif à utiliser dans un scénario de blue/green déploiement (AWS CLI)

Comprendre les mathématiques de métrique

Si vous souhaitez simplement agréger des données métriques existantes, les mathématiques CloudWatch métriques vous évitent les efforts et les coûts liés à la publication d'une autre métrique dans CloudWatch. Vous pouvez utiliser n'importe quelle métrique qui AWS fournit, et vous pouvez également utiliser des métriques que vous définissez dans le cadre de vos applications.

Pour plus d'informations, consultez la section Utilisation des mathématiques métriques dans le guide de CloudWatch l'utilisateur Amazon.

Si vous choisissez d'utiliser une expression mathématique de métrique dans votre politique de mise à l'échelle prédictive, tenez compte des points suivants :

- Les opérations mathématiques de métrique utilisent les points de données de la combinaison unique de nom de la métrique, d'espace de noms et de paires clé/valeur de dimension des métriques.
- Vous pouvez utiliser n'importe quel opérateur arithmétique (+ - */^), fonction statistique (telle que AVG ou SUM) ou toute autre fonction compatible. CloudWatch
- Vous pouvez utiliser à la fois des métriques et les résultats d'autres expressions mathématiques dans les formules de l'expression mathématique.
- Vos expressions mathématiques de métrique peuvent être composées de différentes agrégations. Cependant, une bonne pratique pour le résultat final de l'agrégation consiste à utiliser `Average` pour la métrique de mise à l'échelle et `Sum` pour la métrique de charge.
- Toutes les expressions utilisées dans une spécification de métrique doivent finalement retourner une seule séries temporelles.

Pour utiliser les mathématiques de métrique, procédez comme suit :

- Choisissez un ou plusieurs CloudWatch indicateurs. Créez ensuite l'expression. Pour plus d'informations, consultez la section Utilisation des mathématiques métriques dans le guide de CloudWatch l'utilisateur Amazon.
- Vérifiez que l'expression mathématique de la métrique est valide à l'aide de la CloudWatch console ou de l' CloudWatch GetMetricDataAPI.

Exemple de politique de dimensionnement prédictif pour Amazon EC2 Auto Scaling qui combine les métriques à l'aide des mathématiques métriques (AWS CLI)

Parfois, au lieu de spécifier la métrique directement, vous devrez d'abord traiter ses données d'une certaine manière. Par exemple, une application peut extraire le travail d'une file d'attente Amazon SQS et vous souhaitez utiliser le nombre d'éléments dans la file d'attente comme critère de mise à l'échelle prédictive. Le nombre de messages dans la file d'attente ne définit pas uniquement le nombre d'instances dont vous avez besoin. Par conséquent, un travail supplémentaire est nécessaire pour créer une métrique qui peut être utilisée pour calculer le backlog par instance.

Ce qui suit est un exemple de politique de mise à l'échelle prédictive pour ce scénario. Il spécifie les métriques de mise à l'échelle et de charge qui sont basées sur la métrique `ApproximateNumberOfMessagesVisible` d'Amazon SQS, qui est le nombre de messages disponibles pour la récupération de la file d'attente. Il utilise également la

`GroupInServiceInstances` métrique Amazon EC2 Auto Scaling et une expression mathématique pour calculer le backlog par instance pour la métrique de dimensionnement.

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

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

L'exemple renvoie l'ARN de la politique.

```
{
```

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

Exemple de politique de dimensionnement prédictif à utiliser dans un scénario de blue/green déploiement (AWS CLI)

Une expression de recherche fournit une option avancée dans laquelle vous pouvez demander une métrique à partir de plusieurs groupes Auto Scaling et effectuer des expressions mathématiques sur eux. Cela est particulièrement utile pour les blue/green déploiements.

> **Note**
>
> Un déploiement bleu/vert est une méthode de déploiement dans laquelle vous créez deux groupes Auto Scaling distincts mais identiques. Seul l'un des groupes reçoit le trafic de production. Le trafic utilisateur est initialement dirigé vers le groupe Auto Scaling précédent (« bleu »), tandis qu'un nouveau groupe (« vert ») est utilisé pour le test et l'évaluation d'une nouvelle version d'une application ou d'un service. Le trafic utilisateur est transféré vers le groupe Auto Scaling vert après qu'un nouveau déploiement ait été testé et accepté. Vous pouvez ensuite supprimer le groupe bleu après le succès du déploiement.

Lorsque de nouveaux groupes Auto Scaling sont créés dans le cadre d'un blue/green déploiement, l'historique des métriques de chaque groupe peut être automatiquement inclus dans la politique de dimensionnement prédictif sans que vous ayez à modifier ses spécifications métriques. Pour plus d'informations, consultez la section Utilisation des politiques de dimensionnement prédictif d' EC2 Auto Scaling avec des déploiements bleu/vert sur le AWS Compute Blog.

L'exemple de politique suivant montre comment cela peut être fait. Dans cet exemple, la politique utilise la `CPUUtilization` métrique émise par Amazon EC2. Il utilise la `GroupInServiceInstances` métrique Amazon EC2 Auto Scaling et une expression mathématique pour calculer la valeur de la métrique de dimensionnement par instance. Elle spécifie également une métrique de capacité pour obtenir la métrique `GroupInServiceInstances`.

L'expression de recherche trouve la `CPUUtilization` des instances dans plusieurs groupes Auto Scaling en fonction des critères de recherche spécifiés. Si vous créez ultérieurement un nouveau groupe Auto Scaling qui correspond aux mêmes critères de recherche, `CPUUtilization` des instances dans le nouveau groupe Auto Scaling est automatiquement incluse.

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

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

L'exemple renvoie l'ARN de la politique.

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

Considérations relatives aux métriques personnalisées dans le cadre d'une politique de dimensionnement prédictive

Si un problème survient lors de l'utilisation de métriques personnalisées, nous vous recommandons d'effectuer les opérations suivantes :

- Si un message d'erreur est fourni, lisez le message et résolvez le problème qu'il signale, si possible.
- Si vous n'avez pas validé une expression à l'avance, la put-scaling-policycommande la valide lorsque vous créez votre politique de dimensionnement. Cependant, il est possible que cette commande ne parvienne pas à identifier la cause exacte des erreurs détectées. Pour résoudre les problèmes, corrigez les erreurs que vous recevez en réponse à une demande de get-metric-datacommande. Vous pouvez également résoudre les problèmes liés à l'expression depuis la CloudWatch console.
- Vous devez spécifier `false` pour `ReturnData` si `MetricDataQueries` spécifie la fonction SEARCH() seule sans une fonction mathématique comme SUM(). Cela est dû au fait que les expressions de recherche peuvent renvoyer plusieurs séries temporelles et qu'une spécification métrique basée sur une expression ne peut renvoyer qu'une seule séries temporelles.
- Toutes les métriques impliquées dans une expression de recherche doivent avoir la même résolution.

Limitations

Les limites suivantes s'appliquent.

- Vous pouvez interroger des points de données de 10 métriques au maximum dans une spécification métrique.
- Dans le cadre de cette limite, une expression compte pour une métrique.

Tutoriel : Configurer la scalabilité automatique pour gérer une charge de travail importante

Dans ce tutoriel, vous apprendrez comment effectuer une mise à l'échelle horizontale en fonction des fenêtres de temps où votre application aura une charge de travail plus importante que la normale. Ceci est utile lorsque vous avez une application qui peut soudainement avoir un grand nombre de visiteurs sur une base régulière ou saisonnière.

Vous pouvez utiliser une politique de suivi des cibles et d'échelonnement avec la mise à l'échelle planifiée pour gérer la charge supplémentaire. La mise à l'échelle planifiée initie automatiquement les changements de votre `MinCapacity` et `MaxCapacity` en votre nom, en fonction d'une planification que vous spécifiez. Lorsqu'une politique de suivi des cibles et d'échelonnement est active sur la ressource, elle peut être mise à l'échelle de façon dynamique en fonction de l'utilisation actuelle de la ressource, dans la nouvelle plage de capacité minimale et maximale.

Après avoir terminé ce tutoriel, vous saurez comment :

- Utiliser la mise à l'échelle planifiée pour ajouter une capacité supplémentaire afin de répondre à une charge importante avant qu'elle n'arrive, puis retirer la capacité supplémentaire lorsqu'elle n'est plus nécessaire.
- Utilisez une politique de suivi des cibles et d'échelonnement pour mettre à l'échelle votre application en fonction de l'utilisation actuelle des ressources.

Table des matières

Prérequis

Le didacticiel présume que vous avez déjà effectué les étapes suivantes :

- A créé un Compte AWS.
- A installé et configuré le AWS CLI.
- Vous avez obtenu les autorisations nécessaires pour enregistrer et désenregistrer les ressources en tant que cibles évolutives avec Application Auto Scaling. En outre, a accordé les autorisations nécessaires pour créer des politiques de dimensionnement et des actions planifiées. Pour de plus amples informations, veuillez consulter Gestion des identités et des accès pour Application Auto Scaling.
- Création d'une ressource prise en charge dans un environnement hors production disponible pour ce didacticiel. Si vous n'en avez pas déjà, créez-en un dès maintenant. Pour obtenir des informations sur les services et ressources AWS qui fonctionnent avec Application Auto Scaling, consultez la section Services AWS que vous pouvez utiliser avec Application Auto Scaling.

> **Note**
>
> Au cours de ce tutoriel, il y a deux étapes dans lesquelles vous définissez les valeurs de capacité minimale et maximale de votre ressource sur 0 pour réinitialiser la capacité actuelle à 0. Selon la ressource que vous utilisez avec Application Auto Scaling, il se peut que vous ne puissiez pas réinitialiser la capacité actuelle à 0 pendant ces étapes. Pour vous aider à résoudre le problème, un message dans la sortie indiquera que la capacité minimale ne peut pas être inférieure à la valeur spécifiée et indiquera la valeur de capacité minimale que la AWS ressource peut accepter.

Étape 1 : Enregistrer votre cible évolutive

Commencez par enregistrer votre ressource en tant que cible évolutive avec Application Auto Scaling. Une cible évolutive est une ressource qu'Application Auto Scaling peut augmenter et diminuer.

Pour enregistrer votre cible évolutive avec Application Auto Scaling

- Utilisez la register-scalable-targetcommande suivante pour enregistrer une nouvelle cible évolutive. Définissez les valeurs `--min-capacity` et `--max-capacity` à 0 pour réinitialiser la capacité actuelle à 0.

  Remplacer l'exemple de texte pour `--service-namespace` avec l'espace de nom du service AWS que vous utilisez avec Application Auto Scaling, `--scalable-dimension` avec la

dimension évolutive associée à la ressource que vous enregistrez et `--resource-id` avec un identifiant pour la ressource. Ces valeurs varient en fonction de la ressource utilisée et de la manière dont l'ID de ressource est construit. Consultez les rubriques dans la section Services AWS que vous pouvez utiliser avec Application Auto Scaling pour plus d'informations. Ces rubriques incluent des exemples de commandes qui vous montrent comment enregistrer des cibles évolutives avec 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
```

En cas de réussite, cette commande renvoie l'ARN de la cible évolutive.

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

Étape 2 : Configurer les actions planifiées en fonction de vos besoins

Vous pouvez utiliser la put-scheduled-actioncommande pour créer des actions planifiées configurées pour répondre aux besoins de votre entreprise. Dans ce tutoriel, nous nous concentrons sur une configuration qui arrête la consommation des ressources en dehors des heures de travail en réduisant la capacité à 0.

Pour créer une action planifiée augmente la capacité le matin

1. Pour augmenter la taille de la cible évolutive, utilisez la put-scheduled-actioncommande suivante. Incluez le paramètre `--schedule` avec une planification récurrente, en UTC, en utilisant une expression cron.

   Selon la planification spécifiée (tous les jours à 9 h 00 UTC), Application Auto Scaling met à jour les valeurs `MinCapacity` et `MaxCapacity` dans la plage souhaitée de 1 à 5 unités de capacité.

   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
   ```

   Cette commande ne renvoie pas de sortie lorsqu'elle aboutit.

2. Pour vérifier que l'action planifiée existe, utilisez la describe-scheduled-actionscommande suivante.

   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`]"
```

Voici un exemple de sortie.

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

Pour créer une action planifiée qui diminue la capacité la nuit

1. Répétez la procédure précédente pour créer une autre action planifiée que Application Auto Scaling utilise pour effectuer une diminution de capacité en fin de journée.

   Selon le calendrier spécifié (tous les jours à 20 h 00 UTC), Application Auto Scaling met à jour le `MinCapacity` and de la cible `MaxCapacity` à 0, comme indiqué dans la put-scheduled-actioncommande suivante.

   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. Pour vérifier que l'action planifiée existe, utilisez la describe-scheduled-actionscommande suivante.

   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`]"
   ```

   Voici un exemple de sortie.

   ```
   [
       {
           "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
           },
           ...
   ```

```
    }
]
```

Étape 3 : Ajouter une politique de suivi des cibles et d'échelonnement

Maintenant que vous avez mis en place la planification de base, ajoutez une politique de suivi des cibles et d'échelonnement afin de mettre à l'échelle en fonction de l'utilisation actuelle des ressources.

Avec le suivi de cible, Application Auto Scaling compare la valeur cible de la politique à la valeur actuelle de la métrique spécifiée. Lorsqu'elles sont inégales pendant un certain temps, Application Auto Scaling ajoute ou retire de la capacité pour maintenir des performances stables. Lorsque la charge de votre application et la valeur métrique augmentent, Application Auto Scaling ajoute de la capacité aussi vite que possible sans dépasser `MaxCapacity`. Lorsque Application Auto Scaling supprime de la capacité parce que la charge est minimale, elle le fait sans dépasser `MinCapacity`. En ajustant la capacité en fonction de l'utilisation, vous ne payez que ce dont votre application a besoin.

Si la métrique ne contient pas suffisamment de données parce que votre application n'a pas de charge, Application Auto Scaling n'ajoute ni ne supprime de capacité. Autrement dit, Application Auto Scaling donne la priorité à la disponibilité dans les situations où il n'y a pas assez d'informations disponibles.

Vous pouvez ajouter plusieurs politiques de mise à l'échelle, mais veillez à ne pas ajouter des politiques de mise à l'échelle par étapes conflictuelles qui pourraient entraîner un comportement indésirable. Par exemple, si la politique de mise à l'échelle par étapes lance une activité de mise à l'échelle horizontale avant que la politique de suivi des objectifs et d'échelonnement ne soit prête pour la mise à l'échelle horizontale, l'activité de mise à l'échelle horizontale ne sera pas bloquée. Une fois l'activité de diminution de capacité terminée, la politique de suivi de cible peut demander à Application Auto Scaling de procéder de nouveau à une augmentation de capacité.

Pour créer une politique de suivi des objectifs et d'échelonnement

1. Utilisez la commande put-scaling-policy suivante pour créer la politique.

   Les métriques les plus fréquemment utilisées pour le suivi des cibles sont prédéfinies, et vous pouvez les utiliser sans fournir la spécification complète de la métrique depuis CloudWatch. Pour

plus d'informations sur les métriques prédéfinies disponibles, consultez Politique de suivi des cibles et d'échelonnement pour Application Auto Scaling.

Avant d'exécuter cette commande, assurez-vous que votre métrique prédéfinie attend la valeur cible. Par exemple, pour une montée en puissance lorsque le CPU atteint une utilisation de 50 %, spécifiez une valeur cible de 50,0. Ou, pour augmenter la concurrence allouée à Lambda lorsque l'utilisation atteint 70 %, spécifiez une valeur cible de 0,7. Pour plus d'informations sur les valeurs cibles pour une ressource particulière, reportez-vous à la documentation fournie par le service sur la façon de configurer le suivi de cible. Pour de plus amples informations, veuillez consulter Services AWS que vous pouvez utiliser avec 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\" }}"
```

En cas de succès, cette commande renvoie les noms ARNs et des deux CloudWatch alarmes créées en votre nom.

2. Pour vérifier que l'action planifiée existe, utilisez la describe-scaling-policiescommande suivante.

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`]"
```

Voici un exemple de sortie.

```
[
    {
        "PolicyARN": "arn",
        "TargetTrackingScalingPolicyConfiguration": {
            "PredefinedMetricSpecification": {
                "PredefinedMetricType": "predefinedmetric"
            },
            "TargetValue": 50.0
        },
        "PolicyName": "my-scaling-policy",
        "PolicyType": "TargetTrackingScaling",
        "Alarms": [],
        ...
    }
]
```

Étape 4 : étapes suivantes

Lorsqu'une activité de dimensionnement se produit, vous voyez un enregistrement de celle-ci dans la sortie des activités de dimensionnement de la cible évolutive, par exemple :

```
Successfully set desired count to 1. Change successfully fulfilled by ecs.
```

Pour surveiller vos activités de dimensionnement avec Application Auto Scaling, vous pouvez utiliser la describe-scaling-activitiescommande suivante.

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
```

Étape 5 : nettoyer

Pour empêcher votre compte d'accumuler des frais pour les ressources créées lors de la mise à l'échelle active, vous pouvez nettoyer la configuration de mise à l'échelle associée comme suit.

La suppression de la configuration de dimensionnement ne supprime pas la AWS ressource sous-jacente. Elle ne la ramène pas non plus à sa capacité d'origine. Vous pouvez utiliser la console du service où vous avez créé la ressource pour la supprimer ou ajuster sa capacité.

Pour supprimer les actions planifiées

La commande delete-scheduled-action suivante supprime une action planifiée spécifique. Vous pouvez ignorer cette étape si vous souhaitez conserver les actions planifiées que vous avez créées.

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
```

Pour supprimer la politique de mise à l'échelle

La delete-scaling-policycommande suivante supprime une politique de dimensionnement du suivi des cibles spécifiée. Vous pouvez ignorer cette étape si vous souhaitez conserver la politique de mise à l'échelle que vous avez créée.

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
```

Pour annuler l'inscription d'une cible évolutive

Utilisez la commande deregister-scalable-target suivante pour annuler l'enregistrement de la cible scalable. Si vous avez créé des stratégies de dimensionnement ou si vous avez des actions planifiées qui n'ont pas encore été supprimées, cette commande les supprime. Vous pouvez omettre cette étape si vous souhaitez conserver la cible scalable enregistrée pour une utilisation ultérieure.

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
```

Suspendez et reprenez le dimensionnement pour Application Auto Scaling

Cette rubrique explique comment suspendre puis reprendre une ou plusieurs activités de mise à l'échelle pour les cibles scalables dans votre application. La fonctionnalité de suspension-reprise est utilisée pour suspendre temporairement les activités de mise à l'échelle déclenchées par vos stratégies de dimensionnement et les actions planifiées. Cette fonction est utile, par exemple lorsque vous ne souhaitez pas que la mise à l'échelle automatique risque d'interférer pendant une modification ou l'examen d'un problème de configuration. Vos stratégies de dimensionnement et les actions planifiées peuvent être mises en suspens, puis, lorsque vous êtes prêt, les activités peuvent reprendre.

Dans les exemples de commandes de la CLI qui suivent, vous passez les paramètres au format JSON dans un fichier config.json. Vous pouvez également transmettre ces paramètres sur la ligne de commande en utilisant des guillemets pour entourer la structure de données JSON. Pour plus d'informations, consultez Utilisation des guillemets avec les chaînes de caractères dans AWS CLI du Guide d'utilisateur de la AWS Command Line Interface .

Table des matières

> **Note**
>
> Pour obtenir des instructions sur la suspension des processus de scale-out pendant les déploiements d'Amazon ECS, consultez la documentation suivante : Mise à l'échelle et déploiements automatiques des services dans le guide du développeur Amazon Elastic Container Service

Activités de mise à l'échelle

Application Auto Scaling prend en charge la suspension des activités de mise à l'échelle suivantes :

- Toutes les activités de diminution en charge qui sont déclenchées par une stratégie de mise à l'échelle.

- Toutes les activités de montée en charge qui sont déclenchées par une stratégie de mise à l'échelle.
- Toutes les activités de mise à l'échelle qui impliquent des actions planifiées.

Les descriptions suivantes expliquent ce qui se produit lorsque des activités de mise à l'échelle individuelles sont suspendues. Chaque activité peut être suspendue et reprise de manière indépendante. En fonction de la raison pour laquelle une activité de mise à l'échelle est suspendue, vous pourrez avoir besoin de suspendre plusieurs activités de mise à l'échelle.

`DynamicScalingInSuspended`

- Application Auto Scaling ne supprime pas la capacité lorsqu'une politique de suivi des cibles et d'échelonnement ou une politique de mise à l'échelle par étapes est déclenchée. Cela vous permet de désactiver temporairement les activités de diminution en charge associées aux stratégies de mise à l'échelle sans supprimer les stratégies de mise à l'échelle ou les alarmes CloudWatch qui y sont associées. Lorsque vous reprenez le processus de diminution en charge, Application Auto Scaling évalue les politiques avec des seuils d'alarme qui sont actuellement hors limites.

`DynamicScalingOutSuspended`

- Application Auto Scaling n'ajoute pas la capacité lorsqu'une politique de suivi des cibles et d'échelonnement ou une politique de mise à l'échelle par étapes est déclenchée. Cela vous permet de désactiver temporairement les activités de montée en charge associées aux stratégies de mise à l'échelle sans supprimer les stratégies de mise à l'échelle ou les alarmes CloudWatch qui y sont associées. Lorsque vous reprenez le processus d'augmentation de capacité, Application Auto Scaling évalue les politiques avec des seuils d'alarme qui sont actuellement hors limites.

`ScheduledScalingSuspended`

- Application Auto Scaling n'initie pas les actions de mise à l'échelle qui sont planifiées pour être exécutées pendant la période de suspension. Lorsque vous reprenez la mise à l'échelle planifiée, Application Auto Scaling évalue uniquement les actions planifiées dont l'heure d'exécution n'est pas encore passée.

Suspendre et reprendre les activités de dimensionnement

Vous pouvez suspendre et reprendre des activités de mise à l'échelle individuelles ou toutes les activités de mise à l'échelle pour votre cible évolutive Application Auto Scaling.

> **Note**
>
> Par souci de concision, ces exemples illustrent la suspension et la reprise de la mise à l'échelle pour une table DynamoDB. Pour spécifier une autre cible évolutive, indiquez son espace de noms dans `--service-namespace`, sa dimension évolutive dans `--scalable-dimension`, et son ID de ressource dans `--resource-id`. Pour plus d'informations et des exemples pour chaque service, consultez les rubriques d'Services AWS que vous pouvez utiliser avec Application Auto Scaling.

Pour suspendre une activité de mise à l'échelle

Ouvrez une fenêtre de ligne de commande et utilisez la commande register-scalable-target avec l'option `--suspended-state` comme suit.

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
```

En cas de réussite, cette commande renvoie l'ARN de la cible évolutive.

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

Pour suspendre uniquement les activités de diminution en charge qui sont déclenchées par une stratégie de mise à l'échelle, spécifiez les informations suivantes dans le fichier config.json.

```
{
    "DynamicScalingInSuspended":true
}
```

Pour suspendre uniquement les activités de montée en charge qui sont déclenchées par une stratégie de mise à l'échelle, spécifiez les informations suivantes dans le fichier config.json.

```
{
    "DynamicScalingOutSuspended":true
}
```

Pour suspendre uniquement les activités de mise à l'échelle qui impliquent des actions planifiées, spécifiez les informations suivantes dans le fichier config.json.

```
{
    "ScheduledScalingSuspended":true
}
```

Pour suspendre tous les activités de mise à l'échelle

Utilisez la commande register-scalable-target avec l'option `--suspended-state` comme suit.

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
```

Cet exemple suppose que le fichier config.json contient les paramètres au format JSON suivants.

```
{
```

```
    "DynamicScalingInSuspended":true,
    "DynamicScalingOutSuspended":true,
    "ScheduledScalingSuspended":true
}
```

En cas de réussite, cette commande renvoie l'ARN de la cible évolutive.

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

Affichage des activités de mise à l'échelle suspendues

Utilisez la commande describe-scalable-targets pour déterminer les activités de mise à l'échelle dont l'état est Suspendu pour une cible scalable.

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
```

Voici un exemple de sortie.

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

Reprise des activités de mise à l'échelle

Lorsque vous êtes prêt à reprendre l'activité de mise à l'échelle, vous pouvez utiliser la commande register-scalable-target.

L'exemple de commande suivant reprend toutes les activités de mise à l'échelle pour la cible scalable spécifiée.

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
```

Cet exemple suppose que le fichier config.json contient les paramètres au format JSON suivants.

```
{
    "DynamicScalingInSuspended":false,
    "DynamicScalingOutSuspended":false,
    "ScheduledScalingSuspended":false
}
```

En cas de réussite, cette commande renvoie l'ARN de la cible évolutive.

```
{
```

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

Activités de mise à l'échelle pour Application Auto Scaling

Application Auto Scaling surveille les CloudWatch métriques de votre politique de dimensionnement et lance une activité de dimensionnement lorsque les seuils sont dépassés. Il lance également des activités de mise à l'échelle lorsque vous modifiez la taille maximale ou minimale de la cible évolutive, manuellement ou selon un calendrier.

Lorsqu'une activité de mise à l'échelle se produit, Application Auto Scaling effectue l'une des opérations suivantes :

- Augmente la capacité de la cible évolutive (ce qui est appelé monter en puissance)
- Diminue la capacité de la cible évolutive (ce qui est appelé mise à l'échelle horizontale)

Vous pouvez consulter les activités de mise à l'échelle ayant eu lieu au cours des six dernières semaines.

Recherchez les activités de mise à l'échelle par cible évolutive

Pour voir les activités de dimensionnement pour une cible évolutive spécifique, utilisez la describe-scaling-activitiescommande suivante.

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
```

Voici un exemple de réponse, dans lequel `StatusCode` contient le statut actuel de l'activité et `StatusMessage` contient des informations sur le statut de l'activité de mise à l'échelle.

```
{
```

```
    "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"
        }
    ]
}
```

Pour une description des champs de la réponse, reportez-vous ScalingActivityà la section Application Auto Scaling API Reference.

Les codes d'état suivants indiquent quand l'événement de mise à l'échelle à l'origine de l'activité de mise à l'échette atteint un état terminé :

- `Successful`— La mise à l'échelle s'est terminée avec succès.
- `Overridden`— La capacité souhaitée a été mise à jour par un nouvel événement de mise à l'échelle.
- `Unfulfilled`— Le délai de mise à l'échelle a expiré ou le service cible ne peut pas répondre à la demande.
- `Failed`— La mise à l'échelle a échoué avec une exception.

> **Note**
>
> L'activité de mise à l'échelle peut également avoir le statut `Pending` ou `InProgress`. Toutes les activités de mise à l'échelle ont un statut `Pending` avant que le service cible ne réponde. Une fois que la cible a répondu, le statut de l'activité de mise à l'échelle passe à `InProgress`.

Inclure les activités non dimensionnées

Par défaut, les activités de mise à l'échelle ne reflètent pas les moments où Application Auto Scaling décide de ne pas procéder à une mise à l'échelle.

Supposons, par exemple, qu'un service Amazon ECS dépasse le seuil maximum d'un indicateur donné, mais que le nombre de tâches atteint déjà le nombre maximum de tâches autorisées. Dans ce cas, Application Auto Scaling ne monte pas en puissance le nombre de tâches souhaité.

Pour inclure des activités qui ne sont pas dimensionnées (pas des activités dimensionnées) dans la réponse, ajoutez l'`--include-not-scaled-activities`option à la describe-scaling-activitiescommande.

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**
>
> Si cette commande génère une erreur, assurez-vous d'avoir mis à jour AWS CLI localement la dernière version.

Pour confirmer que la réponse inclut les activités non échelonnées, l'élément `NotScaledReasons` est affiché dans la sortie pour certaines, voire toutes, les activités de mise à l'échelle ayant échoué.

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

Pour une description des champs de la réponse, reportez-vous ScalingActivityà la section Application Auto Scaling API Reference.

Si une activité non échelonnée est renvoyée, selon la raison, le code répertorié dans `Code`, des attributs tels que `CurrentCapacity`, `MaxCapacity`, et `MinCapacity` peuvent être présents dans la réponse.

Pour éviter un grand nombre de doublons, seule la première activité non dimensionnée sera enregistrée dans l'historique des activités de dimensionnement. Les activités ultérieures non dimensionnées ne généreront pas de nouvelles entrées, sauf si la raison de la non-mise à l'échelle change.

Codes de raison

Les codes de motif d'une activité non mise à l'échelle sont les suivants.

Code de motif	Définition			
AutoScalingAnticipatedFlapping	L'algorithme de mise à l'échelle automatique a décidé de ne pas			

Code de motif	Définition			
	effectuer d'action de mise à l'échelle car cela entraîner ait des battement s. Le battement est une boucle infinie de mise à l'échelle horizonta le et de montage en puissance. En d'autres termes, si une action de mise à l'échelle est effectuée, la valeur de la métrique changera pour lancer une autre action de mise à l'échelle dans l’autre sens.			

Code de motif	Définition			
TargetServicePutResourceAsNscalable	Le service cible a temporairement mis la ressource dans un état non évolutif. Application Auto Scaling tentera à nouveau de procéder à une mise à l'échelle lorsque les conditions de dimensionnement automatique spécifiées dans la politique de dimensionnement seront satisfaites.			
AlreadyAtMaxCapacity	La mise à l'échelle est bloquée par la capacité maximale que vous avez indiquée. Si vous souhaitez qu'Application Auto Scaling monte en puissance, vous devez changer la capacité maximale.			

Code de motif	Définition			
AlreadyAtMinCapacity	La mise à l'échelle est bloquée par la capacité minimale que vous avez indiquée. Si vous souhaitez qu'Application Auto Scaling effectue une mise à l'échelle horizontale, vous devez réduire la capacité minimale.			
AlreadyAtDesiredCapacity	L'algorithme de mise à l'échelle automatique a calculé que la capacité révisée était déjà égale à la capacité actuelle.			

Surveillance de l'application Auto Scaling

La surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilité et des performances d'Application Auto Scaling et de vos autres AWS solutions. Vous devez collecter des données de surveillance provenant de toutes les parties de votre AWS solution afin de pouvoir corriger plus facilement une défaillance multipoint, le cas échéant. AWS fournit des outils de surveillance permettant de surveiller Application Auto Scaling, de signaler tout problème et de prendre des mesures automatiques le cas échéant.

Vous pouvez utiliser les fonctionnalités suivantes pour vous aider à gérer vos AWS ressources :

AWS CloudTrail

Avec AWS CloudTrail, vous pouvez suivre les appels passés à l'API Application Auto Scaling par ou au nom de votre Compte AWS. CloudTrail stocke les informations dans des fichiers journaux du compartiment Amazon S3 que vous spécifiez. Vous pouvez identifier les utilisateurs et les comptes qui ont appelé l'Application Auto Scaling, l'adresse IP source à partir de laquelle les appels ont été émis, ainsi que le moment où les appels ont eu lieu. Pour de plus amples informations, veuillez consulter Enregistrez les appels d'API Application Auto Scaling à l'aide de AWS CloudTrail.

> **Note**
>
> Pour plus d'informations sur les autres AWS services qui peuvent vous aider à enregistrer et à collecter des données relatives à vos charges de travail, consultez le guide de journalisation et de surveillance destiné aux propriétaires d'applications dans le guide AWS prescriptif.

Amazon CloudWatch

Amazon vous CloudWatch aide à analyser les journaux et, en temps réel, à surveiller les indicateurs de vos AWS ressources et de vos applications hébergées. Vous pouvez collecter et suivre les métriques, créer des tableaux de bord personnalisés, et définir des alarmes qui vous informent ou prennent des mesures lorsqu'une métrique spécifique atteint un seuil que vous spécifiez. Par exemple, vous pouvez CloudWatch suivre l'utilisation des ressources et vous avertir lorsque l'utilisation est très élevée ou lorsque l'alarme de la métrique est passée

à l'`INSUFFICIENT_DATA`état. Pour de plus amples informations, veuillez consulter Surveillez l'utilisation de ressources évolutives à l'aide de CloudWatch.

CloudWatch suit également les métriques d'utilisation des AWS API pour Application Auto Scaling. Vous pouvez utiliser ces métriques pour configurer des alarmes qui vous alertent lorsque le volume d'appels de votre API dépasse un seuil que vous avez défini. Pour plus d'informations, consultez les statistiques AWS d'utilisation dans le guide de CloudWatch l'utilisateur Amazon.

Amazon EventBridge

Amazon EventBridge est un service de bus d'événements sans serveur qui permet de connecter facilement vos applications à des données provenant de diverses sources. EventBridge fournit un flux de données en temps réel à partir de vos propres applications, applications Software-as-a-Service (SaaS) et AWS services et achemine ces données vers des cibles telles que Lambda. Cela vous permet de surveiller les événements qui se produisent dans les services et de créer des architectures basées sur les événements. Pour de plus amples informations, veuillez consulter Surveillez les événements d'Application Auto Scaling à l'aide d'Amazon EventBridge.

AWS Health Dashboard

Le Health Dashboard (PHD) affiche des informations et fournit également des notifications qui sont invoquées en cas de modification de l'état des AWS ressources. Les informations sont présentées de deux manières : sur un tableau de bord qui montre les événements récents et à venir organisés par catégorie, et dans un journal des événements complet qui contient tous les événements des 90 derniers jours. Pour plus d'informations, voir Commencer avec votre Health Dashboard.

Surveillez l'utilisation de ressources évolutives à l'aide de CloudWatch

Avec Amazon CloudWatch, vous bénéficiez d'une visibilité quasi continue de vos applications sur l'ensemble de ressources évolutives. CloudWatch est un service de surveillance des AWS ressources. Vous pouvez l'utiliser CloudWatch pour collecter et suivre les métriques, définir des alarmes et réagir automatiquement aux modifications de vos AWS ressources. Vous pouvez également créer des tableaux de bord pour surveiller les mesures ou les ensembles de mesures spécifiques dont vous avez besoin.

Lorsque vous interagissez avec les services intégrés à Application Auto Scaling, ils envoient les métriques indiquées dans le tableau suivant à CloudWatch. Dans CloudWatch, les métriques sont

regroupées d'abord par l'espace de noms du service, puis par les différentes combinaisons de dimensions au sein de chaque espace de noms. Ces mesures peuvent vous aider à surveiller l'utilisation des ressources et à planifier la capacité de vos applications. Si la charge de travail de votre application n'est pas constante, cela indique que vous devez envisager d'utiliser la mise à l'échelle automatique. Pour obtenir des descriptions détaillées de ces métriques, consultez la documentation relative à la métrique concernée.

Table des matières

CloudWatch métriques pour surveiller l'utilisation des ressources

Le tableau suivant répertorie les CloudWatch mesures disponibles pour prendre en charge le suivi de l'utilisation des ressources. La liste n'est pas exhaustive mais vous donnera un bon point de départ. Si ces mesures ne s'affichent pas dans la CloudWatch console, assurez-vous d'avoir terminé la configuration de la ressource. Pour plus d'informations, consultez le guide de CloudWatch l'utilisateur Amazon.

Ressources évolutives	Namespace	CloudWatch métrique	Lien vers la documentation
WorkSpaces Applications			
Parcs	AWS/AppStream	Nom : AvailableCapacity Dimension flotte	WorkSpaces Métriques relatives aux applications
Parcs	AWS/AppStream	Nom : CapacityUtilization	WorkSpaces Métriques relatives aux applications

Ressources évolutives	Namespace	CloudWatch métrique	Lien vers la documentation
		Dimension flotte	
Aurora			
Réplicas	AWS/RDS	Nom : CPUUtilization Dimensions : DBClusteridentifiant, rôle (LECTEUR	Métriques Aurora de niveau cluster
Réplicas	AWS/RDS	Nom : DatabaseConnections Dimensions : DBClusteridentifiant, rôle (LECTEUR	Métriques Aurora de niveau cluster
Amazon Comprehend			

Ressources évolutives	Namespace	CloudWatch métrique	Lien vers la documentation
Points de terminaison de classification des documents	AWS/Comprehend	Nom : InferenceUtilization Dimension EndpointArn	Métriques de point de terminaison Amazon Comprehend
Points de terminaison du système de reconnaissance d'entités	AWS/Comprehend	Nom : InferenceUtilization Dimension EndpointArn	Métriques de point de terminaison Amazon Comprehend
DynamoDB			
Tables et index secondaires globaux	AWS/DynamoDB	Nom : ProvisionedReadCapacityUnits Dimensions : TableName, GlobalSecondaryIndexName	Métriques DynamoDB

Ressources évolutives	Namespace	CloudWatch métrique	Lien vers la documentation
Tables et index secondaires globaux	AWS/DynamoDB	Nom : ProvisionedWriteCapacityUnits Dimensions : TableName, GlobalSecondaryIndexName	Métriques DynamoDB
Tables et index secondaires globaux	AWS/DynamoDB	Nom : ConsumedReadCapacityUnits Dimensions : TableName, GlobalSecondaryIndexName	Métriques DynamoDB

Ressources évolutives	Namespace	CloudWatch métrique	Lien vers la documentation
Tables et index secondaires globaux	AWS/DynamoDB	Nom : ConsumedWriteCapacityUnits Dimensions : TableName, GlobalSecondaryIndexName	Métriques DynamoDB
Amazon ECS			
Services	AWS/ECS	Nom : CPUUtilization Dimensions : ClusterName, ServiceName	Métriques Amazon ECS

Ressources évolutives	Namespace	CloudWatch métrique	Lien vers la documentation
Services	AWS/ECS	Nom : MemoryUtilization Dimensions : ClusterName, ServiceName	Métriques Amazon ECS
Services	AWS/ApplicationELB	Nom : RequestCountPerTarget Dimension TargetGroup	Métriques Application Load Balancer
ElastiCache			
Clusters (groupes de réplication)	AW/ElastiCache	Nom : DatabaseMemoryUsageCountedForEvictPercentage Dimension ReplicationGroupId	ElastiCache Métriques Valkey et Redis OSS

Ressources évolutives	Namespace	CloudWatch métrique	Lien vers la documentation
Clusters (groupes de réplication)	AWS/ElastiCache	Nom : DatabaseCapacityUsageCountedForEvictPercentage Dimension ReplicationGroupId	ElastiCache Métriques Valkey et Redis OSS
Clusters (groupes de réplication)	AWS/ElastiCache	Nom : EngineCPUUtilization Dimensions : ReplicationGroupId, Rôle (principal)	ElastiCache Métriques Valkey et Redis OSS

Ressources évolutives	Namespace	CloudWatch métrique	Lien vers la documentation
Clusters (groupes de réplication)	AWS/ElastiCache	Nom : EngineCPUUtilization Dimensions : ReplicationGroupId, rôle (réplique)	ElastiCache Métriques Valkey et Redis OSS
Clusters (cache)	AWS/ElastiCache	Nom : EngineCPUUtilization Dimensions : CacheClusterId, Nœud	ElastiCache Métriques Memcached
Clusters (cache)	AWS/ElastiCache	Nom : DatabaseCapacityMemoryUsagePercentage Dimensions : CacheClusterId	ElastiCache Métriques Memcached

Ressources évolutives	Namespace	CloudWatch métrique	Lien vers la documentation
Amazon EMR			
Clusters	AW/ ElasticMapReduce	Nom : YARNMemory AvailablePercentage Dimension ClusterId	Métriques Amazon EMR
Amazon Keyspaces			
Tables	AWS/ Cassandra	Nom : ProvisionedReadCapacityUnits Dimensions : Keyspace, TableName	Métriques Amazon Keyspaces
Tables	AWS/ Cassandra	Nom : ProvisionedWriteCapacityUnits Dimensions : Keyspace, TableName	Métriques Amazon Keyspaces

Ressources évolutives	Namespace	CloudWatch métrique	Lien vers la documentation
Tables	AWS/Cassandra	Nom : ConsumedReadCapacityUnits Dimensions : Keyspace, TableName	Métriques Amazon Keyspaces
Tables	AWS/Cassandra	Nom : ConsumedWriteCapacityUnits Dimensions : Keyspace, TableName	Métriques Amazon Keyspaces
Lambda			
Simultanéité allouée	AWS/Lambda	Nom : ProvisionedConcurrencyUtilization Dimensions : FunctionName, Ressource	Métriques de la fonction Lambda

Ressources évolutives	Namespace	CloudWatch métrique	Lien vers la documentation
Amazon MSK			
Stockage du courtier	AWS/Kafka	Nom : KafkaDataLogsDiskUsed Dimensions : nom du cluster	Métriques Amazon MSK
Stockage du courtier	AWS/Kafka	Nom : KafkaDataLogsDiskUsed Dimensions : nom du cluster, identifiant du courtier	Métriques Amazon MSK
Neptune			

Ressources évolutives	Namespace	CloudWatch métrique	Lien vers la documentation
Clusters	AWS/Neptune	Nom : CPUUtilization Dimensions : DBClusteridentifiant, rôle (LECTEUR	Métriques Neptune
SageMaker AI			
Variantes de point de terminaison	AW/SageMaker	Nom : InvocationsPerInstance Dimensions : EndpointName, VariantName	Métriques d'invocation
Composants Inférence	AW/SageMaker	Nom : InvocationsPerCopy Dimensions : InferenceComponentName	Métriques d'invocation

Ressources évolutives	Namespace	CloudWatch métrique	Lien vers la documentation
Concurrence provisionnée pour un point de terminaison sans serveur	AWS/SageMaker	Nom : ServerlessProvisionedConcurrencyUtilization Dimensions : EndpointName, VariantName	Métriques de point de terminaison sans serveur
Spot Fleet (Amazon EC2)			
Spot Fleets	AWS/Spot EC2	Nom : CPUUtilization Dimension FleetRequestId	Métriques du parc d'instances Spot
Spot Fleets	AWS/Spot EC2	Nom : NetworkIn Dimension FleetRequestId	Métriques du parc d'instances Spot

Ressources évolutives	Namespace	CloudWatch métrique	Lien vers la documentation
Spot Fleets	AWS/Spot EC2	Nom : NetworkOut Dimension FleetRequestId	Métriques du parc d'instances Spot
Spot Fleets	AWS/ApplicationELB	Nom : RequestCountPerTarget Dimension TargetGroup	Métriques Application Load Balancer

Métrique prédéfinie pour la politique de mise à l'échelle de suivi des cibles

Le tableau suivant répertorie les types de mesures prédéfinis issus de l'API Application Auto Scaling API Reference avec le nom de CloudWatch métrique correspondant. Chaque métrique prédéfinie représente une agrégation des valeurs de la CloudWatch métrique sous-jacente. Le résultat est l'utilisation moyenne des ressources sur une période d'une minute, sur la base d'un pourcentage, sauf indication contraire. Les mesures prédéfinies ne sont utilisées que dans le cadre de la mise en place de politiques de mise à l'échelle de suivi des cibles.

Vous trouverez plus d'informations sur ces mesures dans la documentation du service qui est disponible à partir du tableau dans CloudWatch métriques pour surveiller l'utilisation des ressources.

Type de métrique prédéfinie	CloudWatch nom de la métrique
WorkSpaces Applications	

Type de métrique prédéfinie	CloudWatch nom de la métrique
`AppStreamAverageCapacityUtilization`	CapacityUtilization
Aurora	
`RDSReaderAverageCPUUtilization`	CPUUtilization
`RDSReaderAverageDatabaseConnections`	DatabaseConnections[1]
Amazon Comprehend	
`ComprehendInferenceUtilization`	InferenceUtilization
DynamoDB	
`DynamoDBReadCapacityUtilization`	ProvisionedReadCapacityUnits, ConsumedReadCapacityUnits m[2]
`DynamoDBWriteCapacityUtilization`	ProvisionedWriteCapacityUnits, ConsumedWriteCapacityUnits m[2]
Amazon ECS	
`ECSServiceAverageCPUUtilization`	CPUUtilization
`ECSServiceAverageMemoryUtilization`	MemoryUtilization
`ALBRequestCountPerTarget`	RequestCountPerTarget[1]
ElastiCache	
`ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage`	DatabaseMemoryUsageCountedForEvictPercentage
`ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage`	DatabaseCapacityUsageCountedForEvictPercentage

Type de métrique prédéfinie	CloudWatch nom de la métrique
`ElastiCachePrimaryEngineCPUUtilization`	Moteur CPUUtilization
`ElastiCacheReplicaEngineCPUUtilization`	Moteur CPUUtilization
`ElastiCacheEngineCPUUtilization`	Moteur CPUUtilization
`ElastiCacheDatabaseMemoryUsagePercentage`	DatabaseMemoryUsagePercentage
Amazon Keyspaces	
`CassandraReadCapacityUtilization`	ProvisionedReadCapacityUnits, ConsumedReadCapacityUnits m²
`CassandraWriteCapacityUtilization`	ProvisionedWriteCapacityUnits, ConsumedWriteCapacityUnits m²
Lambda	
`LambdaProvisionedConcurrencyUtilization`	ProvisionedConcurrencyUtilization
Amazon MSK	
`KafkaBrokerStorageUtilization`	KafkaDataLogsDiskUsed
Neptune	
`NeptuneReaderAverageCPUUtilization`	CPUUtilization
SageMaker AI	
`SageMakerVariantInvocationsPerInstance`	InvocationsPerInstance[1]

Type de métrique prédéfinie	CloudWatch nom de la métrique
`SageMakerInferenceComponentInvocationsPerCopy`	InvocationsPerCopy[1]
`SageMakerVariantProvisionedConcurrencyUtilization`	ServerlessProvisionedConcurrencyUtilization
`SageMakerInferenceComponentConcurrentRequestsPerCopyHighResolution`	ConcurrentRequestsPerCopy
`SageMakerVariantConcurrentRequestsPerModelHighResolution`	ConcurrentRequestsPerModel
Parc d'instances Spot	
`EC2SpotFleetRequestAverageCPUUtilization`	CPUUtilization[3]
`EC2SpotFleetRequestAverageNetworkIn`[3]	NetworkIn[1] [3]
`EC2SpotFleetRequestAverageNetworkOut`[3]	NetworkOut[1] [3]
`ALBRequestCountPerTarget`	RequestCountPerTarget[1]

[1] La métrique est basée sur un nombre et non sur un pourcentage.

[2] Pour DynamoDB et Amazon Keyspaces, les mesures prédéfinies sont une agrégation de deux mesures destinées à faciliter le dimensionnement en fonction CloudWatch de la consommation de débit provisionnée.

[3] Pour de meilleures performances de mise à l'échelle, la surveillance EC2 détaillée d'Amazon doit être utilisée.

Métriques et dimensions de mise à l'échelle

L'espace de `AWS/ApplicationAutoScaling` noms inclut les métriques suivantes pour les politiques de dimensionnement prédictif. Ces mesures sont disponibles avec une résolution d'une heure et peuvent vous aider à évaluer la précision des prévisions en comparant les valeurs prévisionnelles aux valeurs réelles.

Métrique	Description	Dimensions
`PredictiveScalingLoadForecast`	La quantité de charge qui devrait être générée par votre application. Le`Average`,`Minimum`, et`Maximum`les statistiques sont utiles, mais`Sum`la statistique ne l'est pas. Critères de déclaration : Reporté après la création de la prévision initiale.	`ResourceId` , `ServiceNamespace` , `PolicyName` , `ScalableDimension` , `PairIndex`
`PredictiveScalingCapacityForecast`	La quantité prévue de capacité nécessaire pour répondre à la demande des applications. Ceci est basé sur la prévision de charge et le niveau d'utilisation cible auxquels vous souhaitez maintenir vos ressources Application Auto Scaling. Les statistiques `Average`, `Minimum` et `Maximum` statistiques sont utiles, mais la statistique `Sum` ne l'est pas. Critères de déclaration : Reporté après la création de la prévision initiale.	`ResourceId` , `ServiceNamespace` , `PolicyName` , `ScalableDimension`
`PredictiveScalingMetricPairCorrelation`	Corrélation entre la métrique de mise à l'échelle et la moyenne par instance de la métrique de charge. La mise à l'échelle prédictive suppose une corrélation élevée. Par conséquent, si vous observez une valeur faible pour cette métrique,	`ResourceId` , `ServiceNamespace` , `PolicyName` , `ScalableD`

Métrique	Description	Dimensions
	il est préférable de ne pas utiliser de paire de métriques. Les statistiques `Average`, `Minimum` et `Maximum` statistiques sont utiles, mais la statistique `Sum` ne l'est pas. Critères de déclaration : Reporté après la création de la prévision initiale.	`imension` , `PairIndex`

Enregistrez les appels d'API Application Auto Scaling à l'aide de AWS CloudTrail

Application Auto Scaling est intégré à AWS CloudTrailun service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un Service AWS. CloudTrail capture les appels d'API pour Application Auto Scaling sous forme d'événements. Les appels capturés incluent les appels provenant des appels de code AWS Management Console et aux opérations de l'API Application Auto Scaling. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande envoyée à Application Auto Scaling, l'adresse IP à partir de laquelle la demande a été faite, la date à laquelle elle a été faite et des informations supplémentaires.

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer :

- Si la demande a été effectuée avec des informations d'identification d'utilisateur root ou d'utilisateur root.
- Si la demande a été faite au nom d'un utilisateur du centre d'identité IAM.
- Si la demande a été effectuée avec les informations d'identification de sécurité temporaires d'un rôle ou d'un utilisateur fédéré.
- Si la requête a été effectuée par un autre Service AWS.

CloudTrail est actif dans votre compte Compte AWS lorsque vous créez le compte et vous avez automatiquement accès à l'historique des CloudTrail événements. L'historique des CloudTrail événements fournit un enregistrement consultable, consultable, téléchargeable et immuable

des 90 derniers jours des événements de gestion enregistrés dans un. Région AWS Pour plus d'informations, consultez la section Utilisation de l'historique des CloudTrail événements dans le guide de AWS CloudTrail l'utilisateur. La consultation de CloudTrail l'historique des événements est gratuite.

Pour un enregistrement continu des événements survenus au Compte AWS cours des 90 derniers jours, créez un parcours.

CloudTrail sentiers

Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Tous les sentiers créés à l'aide du AWS Management Console sont multirégionaux. Vous ne pouvez créer un journal de suivi en une ou plusieurs régions à l'aide de l' AWS CLI. Il est recommandé de créer un parcours multirégional, car vous capturez l'activité dans l'ensemble Régions AWS de votre compte. Si vous créez un journal de suivi pour une seule région, il convient de n'afficher que les événements enregistrés dans le journal de suivi pour une seule région Région AWS. Pour plus d'informations sur les journaux de suivi, consultez Créez un journal de suivi dans vos Compte AWS et Création d'un journal de suivi pour une organisation dans le AWS CloudTrail Guide de l'utilisateur.

Vous pouvez envoyer une copie de vos événements de gestion en cours dans votre compartiment Amazon S3 gratuitement CloudTrail en créant un journal. Toutefois, des frais de stockage Amazon S3 sont facturés. Pour plus d'informations sur la CloudTrail tarification, consultez la section AWS CloudTrail Tarification. Pour obtenir des informations sur la tarification Amazon S3, consultez Tarification Amazon S3.

Événements de gestion d'Application Auto Scaling dans CloudTrail

Les événements de gestion fournissent des informations sur les opérations de gestion effectuées sur les ressources de votre Compte AWS. Ils sont également connus sous le nom opérations de plan de contrôle. Par défaut, CloudTrail enregistre les événements de gestion.

Application Auto Scaling enregistre toutes les opérations du plan de contrôle Application Auto Scaling en tant qu'événements de gestion. Pour obtenir la liste des opérations du plan de contrôle Application Auto Scaling auxquelles Application Auto Scaling se connecte CloudTrail, consultez le manuel Application Auto Scaling API Reference.

Exemples d'événements Application Auto Scaling

Un événement représente une demande unique provenant de n'importe quelle source et inclut des informations sur l'opération d'API demandée, la date et l'heure de l'opération, les paramètres de la demande, etc. CloudTrail les fichiers journaux ne constituent pas une trace ordonnée des appels d'API publics. Les événements n'apparaissent donc pas dans un ordre spécifique.

L'exemple suivant montre un CloudTrail événement illustrant l'`DescribeScalableTargets`opération.


```
{
    "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"
}
```

Pour plus d'informations sur le contenu des CloudTrail enregistrements, voir le contenu des CloudTrail enregistrements dans le Guide de AWS CloudTrail l'utilisateur.

Application Auto Scaling RemoveAction fait appel à CloudWatch

Votre AWS CloudTrail journal peut indiquer qu'Application Auto Scaling appelle l' CloudWatch `RemoveAction`API lorsqu'Application Auto Scaling demande CloudWatch de supprimer l'action de dimensionnement automatique d'une alarme. Cela peut se produire si vous annulez l'enregistrement d'une cible évolutive, supprimez une politique de dimensionnement ou si une alarme invoque une politique de dimensionnement inexistante.

Surveillez les événements d'Application Auto Scaling à l'aide d'Amazon EventBridge

Amazon EventBridge, anciennement appelé CloudWatch Events, vous aide à surveiller les événements spécifiques à Application Auto Scaling et à lancer des actions cibles qui en utilisent d'autres Services AWS. Les événements de Services AWS sont transmis EventBridge en temps quasi réel.

À l'aide de EventBridge, vous pouvez créer des règles qui correspondent aux événements entrants et les acheminer vers des cibles à des fins de traitement.

Pour plus d'informations, consultez Getting started with Amazon EventBridge dans le guide de EventBridge l'utilisateur Amazon.

Événements Application Auto Scaling

Voici des exemples qui montrent des évènements d'Application Auto Scaling. Les événements sont générés sur la base du meilleur effort.

Seuls les événements spécifiques à scaled to max et aux appels d'API via CloudTrail sont actuellement disponibles pour Application Auto Scaling.

Types d'événements

- Événement de modification d'état : mise à l'échelle à la capacité maximale
- Événements pour les appels d'API via CloudTrail

Événement de modification d'état : mise à l'échelle à la capacité maximale

L'exemple d'événement suivant montre qu'Application Auto Scaling a augmenté (fait monter en puissance) la capacité de la cible capable d'être mise à l'échelle à sa limite de taille maximale. Si la demande augmente à nouveau, la mise à l'échelle automatique des applications ne pourra pas mettre à l'échelle la cible à une taille plus grande, car elle est déjà mise à l'échelle à sa taille maximale.

Dans l'objet `detail`, les valeurs des attributs `resourceId`, `serviceNamespace`, et `scalableDimension` identifient la cible capable d'être mise en l'échelle. Les valeurs des attributs `newDesiredCapacity` et `oldDesiredCapacity` font référence à la nouvelle capacité après la montée en puissance de l'événement et à la capacité d'origine avant la montée en puissance de l'événement. La `maxCapacity` est la limite de taille maximale de la cible capable d'être mise à l'échelle.

```
{
  "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"
```

```
}
```

Pour créer une règle qui capture tous les événements de changement d'état `scaledToMax` pour toutes les cibles capables d'être mises à l'échelle, utilisez l'exemple de modèle d'événement suivant.

```
{
  "source": [
    "aws.application-autoscaling"
  ],
  "detail-type": [
    "Application Auto Scaling Scaling Activity State Change"
  ],
  "detail": {
    "scaledToMax": [
      true
    ]
  }
}
```

Événements pour les appels d'API via CloudTrail

Un trail est une configuration AWS CloudTrail utilisée pour transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3. CloudTrail les fichiers journaux contiennent des entrées de journal. Un événement représente une entrée de journal, et il comprend des informations sur l'action demandée, la date et l'heure de l'action et les paramètres de la demande. Pour savoir comment démarrer CloudTrail, consultez la section Création d'un parcours dans le guide de AWS CloudTrail l'utilisateur.

Les événements diffusés via CloudTrail ont `AWS API Call via CloudTrail` pour valeur de`detail-type`.

L'exemple d'événement suivant représente une entrée de fichier CloudTrail journal qui indique qu'un utilisateur de console a appelé l'RegisterScalableTargetaction 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"
  }
}
```

Pour créer une règle basée sur tous les appels d'DeregisterScalableTargetAPI DeleteScalingPolicyet pour toutes les cibles évolutives, utilisez l'exemple de modèle d'événements suivant :

```
{
  "source": [
    "aws.autoscaling"
  ],
  "detail-type": [
    "AWS API Call via CloudTrail"
  ],
  "detail": {
    "eventSource": [
      "autoscaling.amazonaws.com"
    ],
    "eventName": [
       "DeleteScalingPolicy",
       "DeregisterScalableTarget"
    ],
    "additionalEventData": {
      "service": [
        "application-autoscaling"
      ]
    }
  }
}
```

Pour plus d'informations sur l'utilisation CloudTrail, consultezEnregistrez les appels d'API Application Auto Scaling à l'aide de AWS CloudTrail.

Utilisation de ce service avec un AWS SDK

AWS des kits de développement logiciel (SDKs) sont disponibles pour de nombreux langages de programmation courants. Chaque kit SDK fournit une API, des exemples de code et de la documentation qui facilitent la création d'applications par les développeurs dans leur langage préféré.

Documentation SDK	Exemples de code
AWS SDK pour C++	AWS SDK pour C++ exemples de code
AWS CLI	AWS CLI exemples de code
AWS SDK pour Go	AWS SDK pour Go exemples de code
AWS SDK pour Java	AWS SDK pour Java exemples de code
AWS SDK pour JavaScript	AWS SDK pour JavaScript exemples de code
AWS SDK pour Kotlin	AWS SDK pour Kotlin exemples de code
AWS SDK pour .NET	AWS SDK pour .NET exemples de code
AWS SDK pour PHP	AWS SDK pour PHP exemples de code
Outils AWS pour PowerShell	Outils AWS pour PowerShell exemples de code
AWS SDK pour Python (Boto3)	AWS SDK pour Python (Boto3) exemples de code
AWS SDK pour Ruby	AWS SDK pour Ruby exemples de code
AWS SDK pour Rust	AWS SDK pour Rust exemples de code
AWS SDK pour SAP ABAP	AWS SDK pour SAP ABAP exemples de code
AWS SDK pour Swift	AWS SDK pour Swift exemples de code

Exemple de disponibilité

Vous n'avez pas trouvé ce dont vous avez besoin ? Demandez un exemple de code en utilisant le lien Faire un commentaire en bas de cette page.

Exemples de code pour Application Auto Scaling à l'aide de AWS SDKs

Les exemples de code suivants montrent comment utiliser Application Auto Scaling avec un kit de développement AWS logiciel (SDK).

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit de développement logiciel (SDK).

Exemples de code

- Exemples de base pour l'utilisation d'Application Auto Scaling AWS SDKs
  - Actions pour Application Auto Scaling à l'aide de AWS SDKs
    - Utilisation DeleteScalingPolicy avec un AWS SDK ou une CLI
    - Utilisation de DeleteScheduledAction avec une CLI
    - Utilisation de DeregisterScalableTarget avec une CLI
    - Utilisation de DescribeScalableTargets avec une CLI
    - Utilisation de DescribeScalingActivities avec une CLI
    - Utilisation DescribeScalingPolicies avec un AWS SDK ou une CLI
    - Utilisation de DescribeScheduledActions avec une CLI
    - Utilisation de PutScalingPolicy avec une CLI
    - Utilisation de PutScheduledAction avec une CLI
    - Utilisation RegisterScalableTarget avec un AWS SDK ou une CLI

Exemples de base pour l'utilisation d'Application Auto Scaling AWS SDKs

Les exemples de code suivants montrent comment utiliser les bases d'Application Auto Scaling avec AWS SDKs.

Exemples

Actions pour Application Auto Scaling à l'aide de AWS SDKs

Les exemples de code suivants montrent comment effectuer des actions Application Auto Scaling individuelles avec AWS SDKs. Chaque exemple inclut un lien vers GitHub, où vous pouvez trouver des instructions pour configurer et exécuter le code.

Les exemples suivants incluent uniquement les actions les plus couramment utilisées. Pour obtenir une liste complète, consultez la Référence des API Application Auto Scaling.

Exemples

- Utilisation DescribeScalingPolicies avec un AWS SDK ou une CLI
- Utilisation de DescribeScheduledActions avec une CLI
- Utilisation de PutScalingPolicy avec une CLI
- Utilisation de PutScheduledAction avec une CLI
- Utilisation RegisterScalableTarget avec un AWS SDK ou une CLI

Utilisation **`DeleteScalingPolicy`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `DeleteScalingPolicy`.

CLI

AWS CLI

Pour supprimer une stratégie de mise à l'échelle

Cet exemple supprime une stratégie de mise à l'échelle pour l'application Web du service Amazon ECS exécutée dans le cluster par défaut.

Commande :

```
aws application-autoscaling delete-scaling-policy --policy-name web-app-cpu-lt-25
 --scalable-dimension ecs:service:DesiredCount --resource-id service/default/web-
app --service-namespace ecs
```

- Pour plus de détails sur l'API, reportez-vous DeleteScalingPolicyà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.regions.Region;
```

```
import
 software.amazon.awssdk.services.applicationautoscaling.ApplicationAutoScalingClient;
import
 software.amazon.awssdk.services.applicationautoscaling.model.ApplicationAutoScalingExcep
import
 software.amazon.awssdk.services.applicationautoscaling.model.DeleteScalingPolicyRequest;
import
 software.amazon.awssdk.services.applicationautoscaling.model.DeregisterScalableTargetReq
import
 software.amazon.awssdk.services.applicationautoscaling.model.DescribeScalableTargetsRequ
import
 software.amazon.awssdk.services.applicationautoscaling.model.DescribeScalableTargetsResp
import
 software.amazon.awssdk.services.applicationautoscaling.model.DescribeScalingPoliciesRequ
import
 software.amazon.awssdk.services.applicationautoscaling.model.DescribeScalingPoliciesResp
import
 software.amazon.awssdk.services.applicationautoscaling.model.ScalableDimension;
import
 software.amazon.awssdk.services.applicationautoscaling.model.ServiceNamespace;

/**
 * Before running this Java V2 code example, set up your development environment,
 including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class DisableDynamoDBAutoscaling {
    public static void main(String[] args) {
        final String usage = """

            Usage:
               <tableId> <policyName>\s

            Where:
               tableId - The table Id value (for example, table/Music).\s
               policyName - The name of the policy (for example, $Music5-scaling-
policy).

            """;
```

```
        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        ApplicationAutoScalingClient appAutoScalingClient =
ApplicationAutoScalingClient.builder()
            .region(Region.US_EAST_1)
            .build();

        ServiceNamespace ns = ServiceNamespace.DYNAMODB;
        ScalableDimension tableWCUs =
ScalableDimension.DYNAMODB_TABLE_WRITE_CAPACITY_UNITS;
        String tableId = args[0];
        String policyName = args[1];

        deletePolicy(appAutoScalingClient, policyName, tableWCUs, ns, tableId);
        verifyScalingPolicies(appAutoScalingClient, tableId, ns, tableWCUs);
        deregisterScalableTarget(appAutoScalingClient, tableId, ns, tableWCUs);
        verifyTarget(appAutoScalingClient, tableId, ns, tableWCUs);
    }

    public static void deletePolicy(ApplicationAutoScalingClient
appAutoScalingClient, String policyName, ScalableDimension tableWCUs,
ServiceNamespace ns, String tableId) {
        try {
            DeleteScalingPolicyRequest delSPRequest =
DeleteScalingPolicyRequest.builder()
                .policyName(policyName)
                .scalableDimension(tableWCUs)
                .serviceNamespace(ns)
                .resourceId(tableId)
                .build();

            appAutoScalingClient.deleteScalingPolicy(delSPRequest);
            System.out.println(policyName +" was deleted successfully.");

        } catch (ApplicationAutoScalingException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
        }
    }

    // Verify that the scaling policy was deleted
```

```
   public static void verifyScalingPolicies(ApplicationAutoScalingClient
appAutoScalingClient, String tableId, ServiceNamespace ns, ScalableDimension
tableWCUs) {
       DescribeScalingPoliciesRequest dscRequest =
DescribeScalingPoliciesRequest.builder()
           .scalableDimension(tableWCUs)
           .serviceNamespace(ns)
           .resourceId(tableId)
           .build();

       DescribeScalingPoliciesResponse response =
appAutoScalingClient.describeScalingPolicies(dscRequest);
       System.out.println("DescribeScalableTargets result: ");
       System.out.println(response);
   }

   public static void deregisterScalableTarget(ApplicationAutoScalingClient
appAutoScalingClient, String tableId, ServiceNamespace ns, ScalableDimension
tableWCUs) {
       try {
           DeregisterScalableTargetRequest targetRequest =
DeregisterScalableTargetRequest.builder()
               .scalableDimension(tableWCUs)
               .serviceNamespace(ns)
               .resourceId(tableId)
               .build();

           appAutoScalingClient.deregisterScalableTarget(targetRequest);
           System.out.println("The scalable target was deregistered.");

       } catch (ApplicationAutoScalingException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
       }
   }

   public static void verifyTarget(ApplicationAutoScalingClient
appAutoScalingClient, String tableId, ServiceNamespace ns, ScalableDimension
tableWCUs) {
       DescribeScalableTargetsRequest dscRequest =
DescribeScalableTargetsRequest.builder()
           .scalableDimension(tableWCUs)
           .serviceNamespace(ns)
           .resourceIds(tableId)
           .build();
```

```
        DescribeScalableTargetsResponse response =
 appAutoScalingClient.describeScalableTargets(dscRequest);
        System.out.println("DescribeScalableTargets result: ");
        System.out.println(response);
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteScalingPolicyà la section Référence des AWS SDK for Java 2.x API.

PowerShell

Outils pour PowerShell V4

Exemple 1 : cette applet de commande supprime la stratégie de mise à l'échelle spécifiée pour une cible évolutive Application Auto Scaling.

```
Remove-AASScalingPolicy -ServiceNamespace AppStream -PolicyName "default-scale-
out" -ResourceId fleet/Test -ScalableDimension appstream:fleet:DesiredCapacity
```

- Pour plus de détails sur l'API, reportez-vous DeleteScalingPolicyà la section Référence des Outils AWS pour PowerShell applets de commande (V4).

Outils pour PowerShell V5

Exemple 1 : cette applet de commande supprime la stratégie de mise à l'échelle spécifiée pour une cible évolutive Application Auto Scaling.

```
Remove-AASScalingPolicy -ServiceNamespace AppStream -PolicyName "default-scale-
out" -ResourceId fleet/Test -ScalableDimension appstream:fleet:DesiredCapacity
```

- Pour plus de détails sur l'API, reportez-vous DeleteScalingPolicyà la section Référence des Outils AWS pour PowerShell applets de commande (V5).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation de **`DeleteScheduledAction`** avec une CLI

Les exemples de code suivants illustrent comment utiliser `DeleteScheduledAction`.

CLI

AWS CLI

Pour supprimer une action planifiée

L'`delete-scheduled-action`exemple suivant supprime l'action planifiée spécifiée de la flotte Amazon AppStream 2.0 spécifiée :

```
aws application-autoscaling delete-scheduled-action \
    --service-namespace appstream \
    --scalable-dimension appstream:fleet:DesiredCapacity \
    --resource-id fleet/sample-fleet \
    --scheduled-action-name my-recurring-action
```

Cette commande ne produit aucune sortie.

Pour plus d'informations, consultez Mise à l'échelle planifiée dans le Guide de l'utilisateur Application Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous DeleteScheduledActionà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell V4

Exemple 1 : cette applet de commande supprime l'action planifiée spécifiée pour une cible évolutive Application Auto Scaling.

```
Remove-AASScheduledAction -ServiceNamespace AppStream -ScheduledActionName
 WeekDaysFleetScaling -ResourceId fleet/MyFleet -ScalableDimension
 appstream:fleet:DesiredCapacity
```

Sortie :

```
Confirm
Are you sure you want to perform this action?
```

```
Performing the operation "Remove-AASScheduledAction (DeleteScheduledAction)" on
 target "WeekDaysFleetScaling".
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is
 "Y"): Y
```

- Pour plus de détails sur l'API, reportez-vous DeleteScheduledActionà la section Référence des Outils AWS pour PowerShell applets de commande (V4).

Outils pour PowerShell V5

Exemple 1 : cette applet de commande supprime l'action planifiée spécifiée pour une cible évolutive Application Auto Scaling.

```
Remove-AASScheduledAction -ServiceNamespace AppStream -ScheduledActionName
 WeekDaysFleetScaling -ResourceId fleet/MyFleet -ScalableDimension
 appstream:fleet:DesiredCapacity
```

Sortie :

```
Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-AASScheduledAction (DeleteScheduledAction)" on
 target "WeekDaysFleetScaling".
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is
 "Y"): Y
```

- Pour plus de détails sur l'API, reportez-vous DeleteScheduledActionà la section Référence des Outils AWS pour PowerShell applets de commande (V5).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation de **`DeregisterScalableTarget`** avec une CLI

Les exemples de code suivants illustrent comment utiliser `DeregisterScalableTarget`.

CLI

AWS CLI

Pour annuler l'enregistrement d'une cible évolutive

Cet exemple annule l'enregistrement d'une cible évolutive pour un service Amazon ECS appelé web-app qui s'exécute dans le cluster par défaut.

Commande :

```
aws application-autoscaling deregister-scalable-target --service-namespace ecs --
scalable-dimension ecs:service:DesiredCount --resource-id service/default/web-app
```

Cet exemple annule l'enregistrement d'une cible évolutive pour une ressource personnalisée. Le fichier custom-resource-id .txt contient une chaîne qui identifie l'ID de ressource, qui, pour une ressource personnalisée, est le chemin d'accès à la ressource personnalisée via votre point de terminaison Amazon API Gateway.

Commande :

```
aws application-autoscaling deregister-scalable-target --
service-namespace custom-resource --scalable-dimension custom-
resource:ResourceType:Property --resource-id file://~/custom-resource-id.txt
```

Contenu du fichier custom-resource-id .txt :

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

- Pour plus de détails sur l'API, reportez-vous DeregisterScalableTargetà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell V4

Exemple 1 : cette applet de commande annule l'enregistrement d'une cible évolutive Application Auto Scaling. Le désenregistrement d'une cible évolutive supprime les politiques de mise à l'échelle qui lui sont associées.

```
Remove-AASScalableTarget -ResourceId fleet/MyFleet -ScalableDimension
 appstream:fleet:DesiredCapacity -ServiceNamespace AppStream
```

Sortie :

```
Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-AASScalableTarget (DeregisterScalableTarget)" on
 target "fleet/MyFleet".
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is
 "Y"): Y
```

- Pour plus de détails sur l'API, reportez-vous DeregisterScalableTargetà la section Référence des Outils AWS pour PowerShell applets de commande (V4).

Outils pour PowerShell V5

Exemple 1 : cette applet de commande annule l'enregistrement d'une cible évolutive Application Auto Scaling. Le désenregistrement d'une cible évolutive supprime les politiques de mise à l'échelle qui lui sont associées.

```
Remove-AASScalableTarget -ResourceId fleet/MyFleet -ScalableDimension
 appstream:fleet:DesiredCapacity -ServiceNamespace AppStream
```

Sortie :

```
Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-AASScalableTarget (DeregisterScalableTarget)" on
 target "fleet/MyFleet".
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is
 "Y"): Y
```

- Pour plus de détails sur l'API, reportez-vous DeregisterScalableTargetà la section Référence des Outils AWS pour PowerShell applets de commande (V5).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation de **`DescribeScalableTargets`** avec une CLI

Les exemples de code suivants illustrent comment utiliser `DescribeScalableTargets`.

CLI

AWS CLI

Pour décrire des cibles évolutives

L'exemple `describe-scalable-targets` suivant décrit les cibles évolutives pour l'espace de noms de service `ecs`.

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

Sortie :

```
{
    "ScalableTargets": [
        {
            "ServiceNamespace": "ecs",
            "ScalableDimension": "ecs:service:DesiredCount",
            "ResourceId": "service/default/web-app",
            "MinCapacity": 1,
            "MaxCapacity": 10,
            "RoleARN": "arn:aws:iam::123456789012:role/
aws-service-role/ecs.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ECSService",
            "CreationTime": 1462558906.199,
            "SuspendedState": {
                "DynamicScalingOutSuspended": false,
                "ScheduledScalingSuspended": false,
                "DynamicScalingInSuspended": false
            },
            "ScalableTargetARN": "arn:aws:application-autoscaling:us-
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
        }
    ]
}
```

Pour plus d'informations, consultez Services AWS que vous pouvez utiliser avec Application Auto Scaling dans le Guide de l'utilisateur Application Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous DescribeScalableTargetsà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell V4

Exemple 1 : cet exemple fournit des informations sur les cibles évolutives de l'application Autoscaling dans l'espace de noms spécifié.

```
Get-AASScalableTarget -ServiceNamespace "AppStream"
```

Sortie :

```
CreationTime      : 11/7/2019 2:30:03 AM
MaxCapacity       : 5
MinCapacity       : 1
ResourceId        : fleet/Test
RoleARN           : arn:aws:iam::012345678912:role/aws-
service-role/appstream.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_AppStreamFleet
ScalableDimension : appstream:fleet:DesiredCapacity
ServiceNamespace  : appstream
SuspendedState    : Amazon.ApplicationAutoScaling.Model.SuspendedState
```

- Pour plus de détails sur l'API, reportez-vous DescribeScalableTargetsà la section Référence des Outils AWS pour PowerShell applets de commande (V4).

Outils pour PowerShell V5

Exemple 1 : cet exemple fournit des informations sur les cibles évolutives de l'application Autoscaling dans l'espace de noms spécifié.

```
Get-AASScalableTarget -ServiceNamespace "AppStream"
```

Sortie :

```
CreationTime      : 11/7/2019 2:30:03 AM
MaxCapacity       : 5
MinCapacity       : 1
ResourceId        : fleet/Test
RoleARN           : arn:aws:iam::012345678912:role/aws-
service-role/appstream.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_AppStreamFleet
```

```
ScalableDimension : appstream:fleet:DesiredCapacity
ServiceNamespace  : appstream
SuspendedState    : Amazon.ApplicationAutoScaling.Model.SuspendedState
```

- Pour plus de détails sur l'API, reportez-vous DescribeScalableTargetsà la section Référence des Outils AWS pour PowerShell applets de commande (V5).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation de **`DescribeScalingActivities`** avec une CLI

Les exemples de code suivants illustrent comment utiliser `DescribeScalingActivities`.

CLI

AWS CLI

Exemple 1 : pour décrire les activités de mise à l'échelle pour le service Amazon ECS spécifié

L'exemple `describe-scaling-activities` suivant décrit les activités de mise à l'échelle d'un service Amazon ECS appelé `web-app` qui s'exécute dans le cluster `default`. La sortie montre une activité de mise à l'échelle initiée par une politique de mise à l'échelle.

```
aws application-autoscaling describe-scaling-activities \
    --service-namespace ecs \
    --resource-id service/default/web-app
```

Sortie :

```
{
    "ScalingActivities": [
        {
            "ScalableDimension": "ecs:service:DesiredCount",
            "Description": "Setting desired count to 1.",
            "ResourceId": "service/default/web-app",
            "ActivityId": "e6c5f7d1-dbbb-4a3f-89b2-51f33e766399",
            "StartTime": 1462575838.171,
            "ServiceNamespace": "ecs",
            "EndTime": 1462575872.111,
```

```
            "Cause": "monitor alarm web-app-cpu-lt-25 in state ALARM triggered
 policy web-app-cpu-lt-25",
            "StatusMessage": "Successfully set desired count to 1. Change
 successfully fulfilled by ecs.",
            "StatusCode": "Successful"
        }
    ]
}
```

Pour plus d'informations, consultez Activités de mise à l'échelle pour Application Auto Scaling dans le Guide de l'utilisateur Application Auto Scaling.

Exemple 2 : pour décrire les activités de mise à l'échelle pour la table DynamoDB spécifiée

L'exemple `describe-scaling-activities` suivant décrit les activités de mise à l'échelle pour une table DynamoDB appelée `TestTable`. La sortie montre les activités de mise à l'échelle initiées par deux actions planifiées différentes.

```
aws application-autoscaling describe-scaling-activities \
    --service-namespace dynamodb \
    --resource-id table/TestTable
```

Sortie :

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


Pour plus d'informations, consultez Activités de mise à l'échelle pour Application Auto Scaling dans le Guide de l'utilisateur Application Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous DescribeScalingActivitiesà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell V4

Exemple 1 : fournit des informations descriptives pour les activités de mise à l'échelle avec un espace de noms de service spécifié pour les six dernières semaines.

```
Get-AASScalingActivity -ServiceNamespace AppStream
```

Sortie :

```
ActivityId        : 2827409f-b639-4cdb-a957-8055d5d07434
Cause             : monitor alarm Appstream2-MyFleet-default-scale-in-Alarm in
 state ALARM triggered policy default-scale-in
Description       : Setting desired capacity to 2.
Details           :
EndTime           : 12/14/2019 11:32:49 AM
ResourceId        : fleet/MyFleet
ScalableDimension : appstream:fleet:DesiredCapacity
ServiceNamespace  : appstream
StartTime         : 12/14/2019 11:32:14 AM
StatusCode        : Successful
StatusMessage     : Successfully set desired capacity to 2. Change successfully
 fulfilled by appstream.
```

- Pour plus de détails sur l'API, reportez-vous DescribeScalingActivitiesà la section Référence des Outils AWS pour PowerShell applets de commande (V4).

Outils pour PowerShell V5

Exemple 1 : fournit des informations descriptives pour les activités de mise à l'échelle avec un espace de noms de service spécifié pour les six dernières semaines.

```
Get-AASScalingActivity -ServiceNamespace AppStream
```

Sortie :

```
ActivityId        : 2827409f-b639-4cdb-a957-8055d5d07434
Cause             : monitor alarm Appstream2-MyFleet-default-scale-in-Alarm in
 state ALARM triggered policy default-scale-in
Description       : Setting desired capacity to 2.
Details           :
EndTime           : 12/14/2019 11:32:49 AM
```

```
ResourceId        : fleet/MyFleet
ScalableDimension : appstream:fleet:DesiredCapacity
ServiceNamespace  : appstream
StartTime         : 12/14/2019 11:32:14 AM
StatusCode        : Successful
StatusMessage     : Successfully set desired capacity to 2. Change successfully
 fulfilled by appstream.
```

- Pour plus de détails sur l'API, reportez-vous DescribeScalingActivitiesà la section Référence des Outils AWS pour PowerShell applets de commande (V5).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`DescribeScalingPolicies`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `DescribeScalingPolicies`.

CLI

AWS CLI

Pour décrire les politiques de mise à l'échelle

Cet exemple de commande décrit les politiques de mise à l'échelle pour l'espace de noms du service ecs.

Commande :

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

Sortie :

```
{
    "ScalingPolicies": [
        {
            "PolicyName": "web-app-cpu-gt-75",
            "ScalableDimension": "ecs:service:DesiredCount",
            "ResourceId": "service/default/web-app",
            "CreationTime": 1462561899.23,
            "StepScalingPolicyConfiguration": {
```

```
                "Cooldown": 60,
                "StepAdjustments": [
                    {
                        "ScalingAdjustment": 200,
                        "MetricIntervalLowerBound": 0.0
                    }
                ],
                "AdjustmentType": "PercentChangeInCapacity"
            },
            "PolicyARN": "arn:aws:autoscaling:us-
west-2:012345678910:scalingPolicy:6d8972f3-efc8-437c-92d1-6270f29a66e7:resource/
ecs/service/default/web-app:policyName/web-app-cpu-gt-75",
            "PolicyType": "StepScaling",
            "Alarms": [
                {
                    "AlarmName": "web-app-cpu-gt-75",
                    "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:web-app-cpu-gt-75"
                }
            ],
            "ServiceNamespace": "ecs"
        },
        {
            "PolicyName": "web-app-cpu-lt-25",
            "ScalableDimension": "ecs:service:DesiredCount",
            "ResourceId": "service/default/web-app",
            "CreationTime": 1462562575.099,
            "StepScalingPolicyConfiguration": {
                "Cooldown": 1,
                "StepAdjustments": [
                    {
                        "ScalingAdjustment": -50,
                        "MetricIntervalUpperBound": 0.0
                    }
                ],
                "AdjustmentType": "PercentChangeInCapacity"
            },
            "PolicyARN": "arn:aws:autoscaling:us-
west-2:012345678910:scalingPolicy:6d8972f3-efc8-437c-92d1-6270f29a66e7:resource/
ecs/service/default/web-app:policyName/web-app-cpu-lt-25",
            "PolicyType": "StepScaling",
            "Alarms": [
                {
                    "AlarmName": "web-app-cpu-lt-25",
```

```
                    "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:web-app-cpu-lt-25"
                }
            ],
            "ServiceNamespace": "ecs"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeScalingPoliciesà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell V4

Exemple 1 : cette applet de commande décrit les politiques de mise à l'échelle d'Application Auto Scaling pour l'espace de noms de service spécifié.

```
Get-AASScalingPolicy -ServiceNamespace AppStream
```

Sortie :

```
Alarms                                  : {Appstream2-LabFleet-default-scale-
out-Alarm}
CreationTime                            : 9/3/2019 2:48:15 AM
PolicyARN                               : arn:aws:autoscaling:us-
west-2:012345678912:scalingPolicy:5659b069-b5cd-4af1-9f7f-3e956d36233e:resource/
appstream/fleet/LabFleet:
                                            policyName/default-scale-out
PolicyName                              : default-scale-out
PolicyType                              : StepScaling
ResourceId                              : fleet/LabFleet
ScalableDimension                       : appstream:fleet:DesiredCapacity
ServiceNamespace                        : appstream
StepScalingPolicyConfiguration          :
 Amazon.ApplicationAutoScaling.Model.StepScalingPolicyConfiguration
TargetTrackingScalingPolicyConfiguration :

Alarms                                  : {Appstream2-LabFleet-default-scale-in-
Alarm}
CreationTime                            : 9/3/2019 2:48:15 AM
```

```
PolicyARN                                   : arn:aws:autoscaling:us-
west-2:012345678912:scalingPolicy:5659b069-b5cd-4af1-9f7f-3e956d36233e:resource/
appstream/fleet/LabFleet:
                                              policyName/default-scale-in
PolicyName                                  : default-scale-in
PolicyType                                  : StepScaling
ResourceId                                  : fleet/LabFleet
ScalableDimension                           : appstream:fleet:DesiredCapacity
ServiceNamespace                            : appstream
StepScalingPolicyConfiguration              :
 Amazon.ApplicationAutoScaling.Model.StepScalingPolicyConfiguration
TargetTrackingScalingPolicyConfiguration :
```

- Pour plus de détails sur l'API, reportez-vous DescribeScalingPoliciesà la section Référence des Outils AWS pour PowerShell applets de commande (V4).

Outils pour PowerShell V5

Exemple 1 : cette applet de commande décrit les politiques de mise à l'échelle d'Application Auto Scaling pour l'espace de noms de service spécifié.

```
Get-AASScalingPolicy -ServiceNamespace AppStream
```

Sortie :

```
Alarms                                      : {Appstream2-LabFleet-default-scale-
out-Alarm}
CreationTime                                : 9/3/2019 2:48:15 AM
PolicyARN                                   : arn:aws:autoscaling:us-
west-2:012345678912:scalingPolicy:5659b069-b5cd-4af1-9f7f-3e956d36233e:resource/
appstream/fleet/LabFleet:
                                              policyName/default-scale-out
PolicyName                                  : default-scale-out
PolicyType                                  : StepScaling
ResourceId                                  : fleet/LabFleet
ScalableDimension                           : appstream:fleet:DesiredCapacity
ServiceNamespace                            : appstream
StepScalingPolicyConfiguration              :
 Amazon.ApplicationAutoScaling.Model.StepScalingPolicyConfiguration
TargetTrackingScalingPolicyConfiguration :

Alarms                                      : {Appstream2-LabFleet-default-scale-in-
Alarm}
```

```
CreationTime                                : 9/3/2019 2:48:15 AM
PolicyARN                                   : arn:aws:autoscaling:us-
west-2:012345678912:scalingPolicy:5659b069-b5cd-4af1-9f7f-3e956d36233e:resource/
appstream/fleet/LabFleet:
                                              policyName/default-scale-in
PolicyName                                  : default-scale-in
PolicyType                                  : StepScaling
ResourceId                                  : fleet/LabFleet
ScalableDimension                           : appstream:fleet:DesiredCapacity
ServiceNamespace                            : appstream
StepScalingPolicyConfiguration              :
 Amazon.ApplicationAutoScaling.Model.StepScalingPolicyConfiguration
TargetTrackingScalingPolicyConfiguration :
```

- Pour plus de détails sur l'API, reportez-vous DescribeScalingPoliciesà la section Référence des Outils AWS pour PowerShell applets de commande (V5).

Rust

SDK pour Rust

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
async fn show_policies(client: &Client) -> Result<(), Error> {
    let response = client
        .describe_scaling_policies()
        .service_namespace(ServiceNamespace::Ec2)
        .send()
        .await?;
    println!("Auto Scaling Policies:");
    for policy in response.scaling_policies() {
        println!("{:?}\n", policy);
    }
    println!("Next token: {:?}", response.next_token());

    Ok(())
}
```

- Pour plus de détails sur l'API, voir DescribeScalingPolicieslа section de référence de l'API AWS SDK for Rust.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation de **`DescribeScheduledActions`** avec une CLI

Les exemples de code suivants illustrent comment utiliser `DescribeScheduledActions`.

CLI

AWS CLI

Pour décrire des actions planifiées

L'exemple `describe-scheduled-actions` suivant affiche les informations des actions planifiées pour l'espace de noms de service spécifié :

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

Sortie :

```
{
    "ScheduledActions": [
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Schedule": "at(2019-05-20T18:35:00)",
            "ResourceId": "table/my-table",
            "CreationTime": 1561571888.361,
            "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledAction:2d36aa3b-cdf9-4565-
b290-81db519b227d:resource/dynamodb/table/my-table:scheduledActionName/my-first-
scheduled-action",
            "ScalableTargetAction": {
                "MinCapacity": 15,
```

```
                "MaxCapacity": 20
            },
            "ScheduledActionName": "my-first-scheduled-action",
            "ServiceNamespace": "dynamodb"
        },
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Schedule": "at(2019-05-20T18:40:00)",
            "ResourceId": "table/my-table",
            "CreationTime": 1561571946.021,
            "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledAction:2d36aa3b-cdf9-4565-
b290-81db519b227d:resource/dynamodb/table/my-table:scheduledActionName/my-second-
scheduled-action",
            "ScalableTargetAction": {
                "MinCapacity": 5,
                "MaxCapacity": 10
            },
            "ScheduledActionName": "my-second-scheduled-action",
            "ServiceNamespace": "dynamodb"
        }
    ]
}
```

Pour plus d'informations, consultez Mise à l'échelle planifiée dans le Guide de l'utilisateur Application Auto Scaling.

- Pour plus de détails sur l'API, voir DescribeScheduledActionsla section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell V4

Exemple 1 : cette applet de commande répertorie les actions planifiées pour votre groupe Auto Scaling qui n'ont pas été exécutées ou qui n'ont pas atteint leur heure de fin.

```
Get-AASScheduledAction -ServiceNamespace AppStream
```

Sortie :

```
CreationTime         : 12/22/2019 9:25:52 AM
```

```
EndTime              : 1/1/0001 12:00:00 AM
ResourceId           : fleet/MyFleet
ScalableDimension    : appstream:fleet:DesiredCapacity
ScalableTargetAction : Amazon.ApplicationAutoScaling.Model.ScalableTargetAction
Schedule             : cron(0 0 8 ? * MON-FRI *)
ScheduledActionARN   : arn:aws:autoscaling:us-
west-2:012345678912:scheduledAction:4897ca24-3caa-4bf1-8484-851a089b243c:resource/
appstream/fleet/MyFleet:scheduledActionName
                       /WeekDaysFleetScaling
ScheduledActionName  : WeekDaysFleetScaling
ServiceNamespace     : appstream
StartTime            : 1/1/0001 12:00:00 AM
```

- Pour plus de détails sur l'API, reportez-vous DescribeScheduledActionsà la section Référence des Outils AWS pour PowerShell applets de commande (V4).

Outils pour PowerShell V5

Exemple 1 : cette applet de commande répertorie les actions planifiées pour votre groupe Auto Scaling qui n'ont pas été exécutées ou qui n'ont pas atteint leur heure de fin.

```
Get-AASScheduledAction -ServiceNamespace AppStream
```

Sortie :

```
CreationTime         : 12/22/2019 9:25:52 AM
EndTime              : 1/1/0001 12:00:00 AM
ResourceId           : fleet/MyFleet
ScalableDimension    : appstream:fleet:DesiredCapacity
ScalableTargetAction : Amazon.ApplicationAutoScaling.Model.ScalableTargetAction
Schedule             : cron(0 0 8 ? * MON-FRI *)
ScheduledActionARN   : arn:aws:autoscaling:us-
west-2:012345678912:scheduledAction:4897ca24-3caa-4bf1-8484-851a089b243c:resource/
appstream/fleet/MyFleet:scheduledActionName
                       /WeekDaysFleetScaling
ScheduledActionName  : WeekDaysFleetScaling
ServiceNamespace     : appstream
StartTime            : 1/1/0001 12:00:00 AM
```

- Pour plus de détails sur l'API, reportez-vous DescribeScheduledActionsà la section Référence des Outils AWS pour PowerShell applets de commande (V5).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation de **`PutScalingPolicy`** avec une CLI

Les exemples de code suivants illustrent comment utiliser `PutScalingPolicy`.

CLI

AWS CLI

Exemple 1 : pour appliquer une politique de suivi des objectifs et d'échelonnement avec une spécification de métrique prédéfinie

L'exemple `put-scaling-policy` suivant applique une politique de suivi des cibles et d'échelonnement avec une spécification de métrique prédéfinie sur un service Amazon ECS appelé web-app dans le cluster par défaut. La politique maintient l'utilisation moyenne de l'UC du service à 75 %, avec des temps de stabilisation de montée en puissance et de mise à l'échelle horizontale de 60 secondes. La sortie contient les noms ARNs et les noms des deux CloudWatch alarmes créées en votre nom.

```
aws application-autoscaling put-scaling-policy --service-namespace ecs \
--scalable-dimension ecs:service:DesiredCount \
--resource-id service/default/web-app \
--policy-name cpu75-target-tracking-scaling-policy --policy-
type TargetTrackingScaling \
--target-tracking-scaling-policy-configuration file://config.json
```

Cet exemple suppose que le répertoire en cours contient un fichier config.json dont le contenu est le suivant :

```
{
     "TargetValue": 75.0,
     "PredefinedMetricSpecification": {
         "PredefinedMetricType": "ECSServiceAverageCPUUtilization"
     },
     "ScaleOutCooldown": 60,
    "ScaleInCooldown": 60
}
```

Sortie :

```
{
    "PolicyARN": "arn:aws:autoscaling:us-
west-2:012345678910:scalingPolicy:6d8972f3-efc8-437c-92d1-6270f29a66e7:resource/
ecs/service/default/web-app:policyName/cpu75-target-tracking-scaling-policy",
    "Alarms": [
        {
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:TargetTracking-service/default/web-app-AlarmHigh-
d4f0770c-b46e-434a-a60f-3b36d653feca",
            "AlarmName": "TargetTracking-service/default/web-app-AlarmHigh-
d4f0770c-b46e-434a-a60f-3b36d653feca"
        },
        {
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:TargetTracking-service/default/web-app-
AlarmLow-1b437334-d19b-4a63-a812-6c67aaf2910d",
            "AlarmName": "TargetTracking-service/default/web-app-
AlarmLow-1b437334-d19b-4a63-a812-6c67aaf2910d"
        }
    ]
}
```

Exemple 2 : pour appliquer une politique de suivi des objectifs et d'échelonnement avec une spécification de métrique personnalisée

L'exemple `put-scaling-policy` suivant applique une politique de suivi des cibles et d'échelonnement avec un type de métrique personnalisé à un service ECS appelé web-app dans le cluster par défaut. La politique maintient l'utilisation moyenne du service à 75 %, avec des temps de stabilisation de montée en puissance et de mise à l'échelle horizontale de 60 secondes. La sortie contient les noms ARNs et les noms des deux CloudWatch alarmes créées en votre nom.

```
aws application-autoscaling put-scaling-policy --service-namespace ecs \
--scalable-dimension ecs:service:DesiredCount \
--resource-id service/default/web-app \
--policy-name cms75-target-tracking-scaling-policy
--policy-type TargetTrackingScaling \
--target-tracking-scaling-policy-configuration file://config.json
```

Cet exemple suppose que le répertoire en cours contient un fichier config.json dont le contenu est le suivant :

```
{
    "TargetValue":75.0,
    "CustomizedMetricSpecification":{
        "MetricName":"MyUtilizationMetric",
        "Namespace":"MyNamespace",
        "Dimensions": [
            {
                "Name":"MyOptionalMetricDimensionName",
                "Value":"MyOptionalMetricDimensionValue"
            }
        ],
        "Statistic":"Average",
        "Unit":"Percent"
    },
    "ScaleOutCooldown": 60,
    "ScaleInCooldown": 60
}
```

Sortie :

```
{
    "PolicyARN": "arn:aws:autoscaling:us-west-2:012345678910:scalingPolicy:
 8784a896-b2ba-47a1-b08c-27301cc499a1:resource/ecs/service/default/web-
app:policyName/cms75-target-tracking-scaling-policy",
    "Alarms": [
        {
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:TargetTracking-service/default/web-app-
AlarmHigh-9bc77b56-0571-4276-ba0f-d4178882e0a0",
            "AlarmName": "TargetTracking-service/default/web-app-
AlarmHigh-9bc77b56-0571-4276-ba0f-d4178882e0a0"
        },
        {
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:TargetTracking-service/default/web-app-
AlarmLow-9b6ad934-6d37-438e-9e05-02836ddcbdc4",
            "AlarmName": "TargetTracking-service/default/web-app-
AlarmLow-9b6ad934-6d37-438e-9e05-02836ddcbdc4"
        }
    ]
```

```
}
```

Exemple 3 : pour appliquer une politique de suivi des objectifs et d'échelonnement uniquement en vue d'une évolutivité horizontale

L'exemple `put-scaling-policy` suivant applique une politique de suivi des cibles et d'échelonnement sur un service Amazon ECS appelé `web-app` dans le cluster par défaut. La politique est utilisée pour augmenter horizontalement le service ECS lorsque la métrique `RequestCountPerTarget` de l'Application Load Balancer dépasse le seuil. La sortie contient l'ARN et le nom de l' CloudWatch alarme créée en votre nom.

```
aws application-autoscaling put-scaling-policy \
    --service-namespace ecs \
    --scalable-dimension ecs:service:DesiredCount \
    --resource-id service/default/web-app \
    --policy-name alb-scale-out-target-tracking-scaling-policy \
    --policy-type TargetTrackingScaling \
    --target-tracking-scaling-policy-configuration file://config.json
```

Contenu de `config.json` :

```
{
     "TargetValue": 1000.0,
     "PredefinedMetricSpecification": {
         "PredefinedMetricType": "ALBRequestCountPerTarget",
         "ResourceLabel": "app/EC2Co-EcsEl-1TKLTMITMM0EO/f37c06a68c1748aa/
targetgroup/EC2Co-Defau-LDNM7Q3ZH1ZN/6d4ea56ca2d6a18d"
     },
     "ScaleOutCooldown": 60,
    "ScaleInCooldown": 60,
    "DisableScaleIn": true
}
```

Sortie :

```
{
    "PolicyARN": "arn:aws:autoscaling:us-
west-2:123456789012:scalingPolicy:6d8972f3-efc8-437c-92d1-6270f29a66e7:resource/
ecs/service/default/web-app:policyName/alb-scale-out-target-tracking-scaling-
policy",
    "Alarms": [
        {
```

```
            "AlarmName": "TargetTracking-service/default/web-app-AlarmHigh-
d4f0770c-b46e-434a-a60f-3b36d653feca",
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-service/default/web-app-AlarmHigh-
d4f0770c-b46e-434a-a60f-3b36d653feca"
        }
    ]
}
```

Pour plus d'informations, consultez Politique de suivi des cibles et d'échelonnement pour Application Auto Scaling dans le AWS Guide de l'utilisateur Application Auto Scaling.

- Pour plus de détails sur l'API, voir PutScalingPolicyla section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell V4

Exemple 1 : cette applet de commande crée ou met à jour une politique pour une cible évolutive Application Auto Scaling. Chaque cible évolutive est identifiée par un espace de noms de service, un ID de ressource et une dimension évolutive.

```
Set-AASScalingPolicy -ServiceNamespace AppStream -PolicyName ASFleetScaleInPolicy
 -PolicyType StepScaling  -ResourceId fleet/MyFleet -ScalableDimension
 appstream:fleet:DesiredCapacity -StepScalingPolicyConfiguration_AdjustmentType
 ChangeInCapacity -StepScalingPolicyConfiguration_Cooldown 360
 -StepScalingPolicyConfiguration_MetricAggregationType Average -
StepScalingPolicyConfiguration_StepAdjustments @{ScalingAdjustment = -1;
 MetricIntervalUpperBound = 0}
```

Sortie :

```
Alarms    PolicyARN
------    ---------
{}        arn:aws:autoscaling:us-
west-2:012345678912:scalingPolicy:4897ca24-3caa-4bf1-8484-851a089b243c:resource/
appstream/fleet/MyFleet:policyName/ASFleetScaleInPolicy
```

- Pour plus de détails sur l'API, reportez-vous PutScalingPolicyà la section Référence des Outils AWS pour PowerShell applets de commande (V4).

Outils pour PowerShell V5

Exemple 1 : cette applet de commande crée ou met à jour une politique pour une cible évolutive Application Auto Scaling. Chaque cible évolutive est identifiée par un espace de noms de service, un ID de ressource et une dimension évolutive.

```
Set-AASScalingPolicy -ServiceNamespace AppStream -PolicyName ASFleetScaleInPolicy
 -PolicyType StepScaling  -ResourceId fleet/MyFleet -ScalableDimension
 appstream:fleet:DesiredCapacity -StepScalingPolicyConfiguration_AdjustmentType
 ChangeInCapacity -StepScalingPolicyConfiguration_Cooldown 360
 -StepScalingPolicyConfiguration_MetricAggregationType Average -
StepScalingPolicyConfiguration_StepAdjustments @{ScalingAdjustment = -1;
 MetricIntervalUpperBound = 0}
```

Sortie :

```
Alarms    PolicyARN
------    ---------
{}        arn:aws:autoscaling:us-
west-2:012345678912:scalingPolicy:4897ca24-3caa-4bf1-8484-851a089b243c:resource/
appstream/fleet/MyFleet:policyName/ASFleetScaleInPolicy
```

- Pour plus de détails sur l'API, reportez-vous PutScalingPolicyà la section Référence des Outils AWS pour PowerShell applets de commande (V5).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation de **`PutScheduledAction`** avec une CLI

Les exemples de code suivants illustrent comment utiliser `PutScheduledAction`.

CLI

AWS CLI

Pour ajouter une action planifiée à une table DynamoDB

Cet exemple ajoute une action planifiée à une table DynamoDB TestTable appelée pour effectuer une mise à l'échelle selon un calendrier récurrent. Selon le calendrier spécifié

(tous les jours à 12 h 15 UTC), si la capacité actuelle est inférieure à la valeur spécifiée pour MinCapacity, Application Auto Scaling s'adapte à la valeur spécifiée par MinCapacity.

Commande :

```
aws application-autoscaling put-scheduled-action --service-
namespace dynamodb --scheduled-action-name my-recurring-action --
schedule "cron(15 12 * * ? *)" --resource-id table/TestTable --
scalable-dimension dynamodb:table:WriteCapacityUnits --scalable-target-
action MinCapacity=6
```

Pour plus d'informations, consultez Dimensionnement planifié dans le Guide de l'utilisateur Application Auto Scaling.

- Pour plus de détails sur l'API, voir PutScheduledActionla section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell V4

Exemple 1 : cette applet de commande crée ou met à jour une action planifiée pour une cible évolutive Application Auto Scaling. Chaque cible évolutive est identifiée par un espace de noms de service, un ID de ressource et une dimension évolutive.

```
Set-AASScheduledAction -ServiceNamespace AppStream -ResourceId fleet/
MyFleet -Schedule "cron(0 0 8 ? * MON-FRI *)" -ScalableDimension
 appstream:fleet:DesiredCapacity -ScheduledActionName WeekDaysFleetScaling -
ScalableTargetAction_MinCapacity 5 -ScalableTargetAction_MaxCapacity 10
```

- Pour plus de détails sur l'API, reportez-vous PutScheduledActionà la section Référence des Outils AWS pour PowerShell applets de commande (V4).

Outils pour PowerShell V5

Exemple 1 : cette applet de commande crée ou met à jour une action planifiée pour une cible évolutive Application Auto Scaling. Chaque cible évolutive est identifiée par un espace de noms de service, un ID de ressource et une dimension évolutive.

```
Set-AASScheduledAction -ServiceNamespace AppStream -ResourceId fleet/
MyFleet -Schedule "cron(0 0 8 ? * MON-FRI *)" -ScalableDimension
```

```
 appstream:fleet:DesiredCapacity -ScheduledActionName WeekDaysFleetScaling -
ScalableTargetAction_MinCapacity 5 -ScalableTargetAction_MaxCapacity 10
```

- Pour plus de détails sur l'API, reportez-vous PutScheduledActionà la section Référence des Outils AWS pour PowerShell applets de commande (V5).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`RegisterScalableTarget`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `RegisterScalableTarget`.

CLI

AWS CLI

Exemple 1 : pour enregistrer un service ECS en tant que cible évolutive

L'exemple `register-scalable-target` suivant enregistre un service Amazon ECS avec Application Auto Scaling. Il ajoute également une balise avec le nom de la clé `environment` et la valeur `production` à la cible évolutive.

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

Sortie :

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:us-
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Pour des exemples d'autres AWS services et ressources personnalisées, consultez les rubriques relatives AWS aux services que vous pouvez utiliser avec Application Auto Scaling dans le Guide de l'utilisateur d'Application Auto Scaling.

Exemple 2 : pour suspendre les activités de mise à l'échelle pour une cible évolutive

L'exemple `register-scalable-target` suivant suspend les activités de mise à l'échelle pour une cible évolutive existante.

```
aws application-autoscaling register-scalable-target \
    --service-namespace dynamodb \
    --scalable-dimension dynamodb:table:ReadCapacityUnits \
    --resource-id table/my-table \
    --suspended-
state DynamicScalingInSuspended=true,DynamicScalingOutSuspended=true,ScheduledScalingSusp
```

Sortie :

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:us-
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Pour plus d'informations, consultez Suspension et reprise de la mise à l'échelle pour Application Auto Scaling dans le Guide de l'utilisateur Application Auto Scaling.

Exemple 3 : pour reprendre les activités de mise à l'échelle pour une cible évolutive

L'exemple `register-scalable-target` suivant reprend les activités de mise à l'échelle pour une cible évolutive existante.

```
aws application-autoscaling register-scalable-target \
    --service-namespace dynamodb \
    --scalable-dimension dynamodb:table:ReadCapacityUnits \
    --resource-id table/my-table \
    --suspended-
state DynamicScalingInSuspended=false,DynamicScalingOutSuspended=false,ScheduledScalingSu
```

Sortie :

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:us-
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Pour plus d'informations, consultez Suspension et reprise de la mise à l'échelle pour Application Auto Scaling dans le Guide de l'utilisateur Application Auto Scaling.

- Pour plus de détails sur l'API, voir RegisterScalableTargetla section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.applicationautoscaling.ApplicationAutoScalingClient;
import
 software.amazon.awssdk.services.applicationautoscaling.model.ApplicationAutoScalingExcep
import
 software.amazon.awssdk.services.applicationautoscaling.model.DescribeScalableTargetsRequ
import
 software.amazon.awssdk.services.applicationautoscaling.model.DescribeScalableTargetsResp
import
 software.amazon.awssdk.services.applicationautoscaling.model.DescribeScalingPoliciesRequ
import
 software.amazon.awssdk.services.applicationautoscaling.model.DescribeScalingPoliciesResp
import software.amazon.awssdk.services.applicationautoscaling.model.PolicyType;
import
 software.amazon.awssdk.services.applicationautoscaling.model.PredefinedMetricSpecificati
import
 software.amazon.awssdk.services.applicationautoscaling.model.PutScalingPolicyRequest;
import
 software.amazon.awssdk.services.applicationautoscaling.model.RegisterScalableTargetReque
import
 software.amazon.awssdk.services.applicationautoscaling.model.ScalingPolicy;
import
 software.amazon.awssdk.services.applicationautoscaling.model.ServiceNamespace;
import
 software.amazon.awssdk.services.applicationautoscaling.model.ScalableDimension;
```

```
import software.amazon.awssdk.services.applicationautoscaling.model.MetricType;
import
 software.amazon.awssdk.services.applicationautoscaling.model.TargetTrackingScalingPolicy
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development environment,
 including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class EnableDynamoDBAutoscaling {
    public static void main(String[] args) {
        final String usage = """

            Usage:
               <tableId> <roleARN> <policyName>\s

            Where:
               tableId - The table Id value (for example, table/Music).
               roleARN - The ARN of the role that has ApplicationAutoScaling
 permissions.
               policyName - The name of the policy to create.

            """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        System.out.println("This example registers an Amazon DynamoDB table,
 which is the resource to scale.");
        String tableId = args[0];
        String roleARN = args[1];
        String policyName = args[2];
        ServiceNamespace ns = ServiceNamespace.DYNAMODB;
        ScalableDimension tableWCUs =
 ScalableDimension.DYNAMODB_TABLE_WRITE_CAPACITY_UNITS;
        ApplicationAutoScalingClient appAutoScalingClient =
 ApplicationAutoScalingClient.builder()
```

```
            .region(Region.US_EAST_1)
            .build();

       registerScalableTarget(appAutoScalingClient, tableId, roleARN, ns,
tableWCUs);
       verifyTarget(appAutoScalingClient, tableId, ns, tableWCUs);
       configureScalingPolicy(appAutoScalingClient, tableId, ns, tableWCUs,
policyName);
   }

   public static void registerScalableTarget(ApplicationAutoScalingClient
appAutoScalingClient, String tableId, String roleARN, ServiceNamespace ns,
ScalableDimension tableWCUs) {
       try {
           RegisterScalableTargetRequest targetRequest =
RegisterScalableTargetRequest.builder()
               .serviceNamespace(ns)
               .scalableDimension(tableWCUs)
               .resourceId(tableId)
               .roleARN(roleARN)
               .minCapacity(5)
               .maxCapacity(10)
               .build();

           appAutoScalingClient.registerScalableTarget(targetRequest);
           System.out.println("You have registered " + tableId);

       } catch (ApplicationAutoScalingException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
       }
   }

   // Verify that the target was created.
   public static void verifyTarget(ApplicationAutoScalingClient
appAutoScalingClient, String tableId, ServiceNamespace ns, ScalableDimension
tableWCUs) {
       DescribeScalableTargetsRequest dscRequest =
DescribeScalableTargetsRequest.builder()
           .scalableDimension(tableWCUs)
           .serviceNamespace(ns)
           .resourceIds(tableId)
           .build();
```

```
        DescribeScalableTargetsResponse response =
appAutoScalingClient.describeScalableTargets(dscRequest);
        System.out.println("DescribeScalableTargets result: ");
        System.out.println(response);
    }

    // Configure a scaling policy.
    public static void configureScalingPolicy(ApplicationAutoScalingClient
appAutoScalingClient, String tableId, ServiceNamespace ns, ScalableDimension
tableWCUs, String policyName) {
        // Check if the policy exists before creating a new one.
        DescribeScalingPoliciesResponse describeScalingPoliciesResponse =
appAutoScalingClient.describeScalingPolicies(DescribeScalingPoliciesRequest.builder()
            .serviceNamespace(ns)
            .resourceId(tableId)
            .scalableDimension(tableWCUs)
            .build());

        if (!describeScalingPoliciesResponse.scalingPolicies().isEmpty()) {
            // If policies exist, consider updating an existing policy instead of
creating a new one.
            System.out.println("Policy already exists. Consider updating it
instead.");
            List<ScalingPolicy> polList =
describeScalingPoliciesResponse.scalingPolicies();
            for (ScalingPolicy pol : polList) {
                System.out.println("Policy name:" +pol.policyName());
            }
        } else {
            // If no policies exist, proceed with creating a new policy.
            PredefinedMetricSpecification specification =
PredefinedMetricSpecification.builder()

.predefinedMetricType(MetricType.DYNAMO_DB_WRITE_CAPACITY_UTILIZATION)
                .build();

            TargetTrackingScalingPolicyConfiguration policyConfiguration =
TargetTrackingScalingPolicyConfiguration.builder()
                .predefinedMetricSpecification(specification)
                .targetValue(50.0)
                .scaleInCooldown(60)
                .scaleOutCooldown(60)
                .build();
```

```
            PutScalingPolicyRequest putScalingPolicyRequest =
 PutScalingPolicyRequest.builder()
                .targetTrackingScalingPolicyConfiguration(policyConfiguration)
                .serviceNamespace(ns)
                .scalableDimension(tableWCUs)
                .resourceId(tableId)
                .policyName(policyName)
                .policyType(PolicyType.TARGET_TRACKING_SCALING)
                .build();

            try {
                appAutoScalingClient.putScalingPolicy(putScalingPolicyRequest);
                System.out.println("You have successfully created a scaling
 policy for an Application Auto Scaling scalable target");
            } catch (ApplicationAutoScalingException e) {
                System.err.println("Error: " +
 e.awsErrorDetails().errorMessage());
            }
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous RegisterScalableTargetà la section Référence des AWS SDK for Java 2.x API.

PowerShell

Outils pour PowerShell V4

Exemple 1 : cette applet de commande enregistre ou met à jour une cible évolutive. Une cible évolutive est une ressource qu'Application Auto Scaling peut augmenter et diminuer.

```
Add-AASScalableTarget -ServiceNamespace AppStream -ResourceId fleet/MyFleet -
ScalableDimension appstream:fleet:DesiredCapacity -MinCapacity 2 -MaxCapacity 10
```

- Pour plus de détails sur l'API, reportez-vous RegisterScalableTargetà la section Référence des Outils AWS pour PowerShell applets de commande (V4).

Outils pour PowerShell V5

Exemple 1 : cette applet de commande enregistre ou met à jour une cible évolutive. Une cible évolutive est une ressource qu'Application Auto Scaling peut augmenter et diminuer.

```
Add-AASScalableTarget -ServiceNamespace AppStream -ResourceId fleet/MyFleet -
ScalableDimension appstream:fleet:DesiredCapacity -MinCapacity 2 -MaxCapacity 10
```

- Pour plus de détails sur l'API, reportez-vous RegisterScalableTargetà la section Référence des Outils AWS pour PowerShell applets de commande (V5).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Prise en charge du balisage pour Application Auto Scaling

Vous pouvez utiliser le AWS CLI ou un SDK pour baliser les cibles évolutives d'Application Auto Scaling. Les cibles évolutives sont les entités qui représentent les ressources AWS ou les ressources personnalisées qu'Application Auto Scaling peut redimensionner.

Chaque balise est une étiquette composée d'une clé et d'une valeur définies par l'utilisateur à l'aide de l'API Application Auto Scaling. Les balises peuvent vous aider à configurer un accès détaillé à des cibles évolutives spécifiques en fonction des besoins de votre organisation. Pour de plus amples informations, veuillez consulter ABAC avec Application Auto Scaling.

Vous pouvez ajouter des balises à de nouvelles cibles évolutives lorsque vous les enregistrez, ou les ajouter à des cibles évolutives existantes.

Les commandes couramment utilisées pour gérer les balises sont les suivantes :

- register-scalable-targetpour étiqueter de nouvelles cibles évolutives lorsque vous les enregistrez.
- tag-resource pour ajouter des balises à une cible évolutive existante.
- list-tags-for-resourcepour renvoyer les balises sur une cible évolutive.
- untag-resource pour supprimer une balise.

Exemple de balisage

Utilisez la register-scalable-targetcommande suivante avec l'`--tags`option. Cet exemple balise une cible évolutive avec deux balises : une clé de balise nommée **`environment`** avec la valeur de balise de **`production`** et une clé de balise nommée **`iscontainerbased`** avec la valeur de balise de **`true`**.

Remplacez les exemples de valeurs pour `--min-capacity` `--max-capacity` et et de texte pour `--service-namespace` par l'espace de noms du AWS service que vous utilisez avec Application Auto Scaling, `--scalable-dimension` par la dimension évolutive associée à la ressource que vous enregistrez et `--resource-id` par un identifiant pour la ressource. Pour plus d'informations et des exemples pour chaque service, consultez les rubriques d'Services AWS que vous pouvez utiliser avec 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
```

En cas de réussite, cette commande renvoie l'ARN de la cible évolutive.

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

> **Note**
>
> Si cette commande génère une erreur, assurez-vous d'avoir mis à jour AWS CLI localement la dernière version.

Balises pour la sécurité

Utilisez des balises pour vérifier que le demandeur (tel qu'un utilisateur ou un rôle IAM) est autorisé à effectuer certaines actions. Fournissez des informations de balise dans l'élément de condition d'une politique IAM à l'aide des clés de condition suivantes :

- Utilisez `aws:ResourceTag/tag-key: tag-value` pour accorder (ou refuser) aux utilisateurs des actions sur des cibles évolutives avec des balises spécifiques.
- Utilisez `aws:RequestTag/tag-key: tag-value` pour exiger qu'une balise spécifique soit présente (ou non) dans une demande.
- Utilisez `aws:TagKeys [tag-key, ...]` pour exiger que des clés de balise spécifiques soient présentes (ou non) dans une demande.

Par exemple, la politique IAM suivante accorde des autorisations à l'utilisateur pour les actions `DeregisterScalableTarget`, `DeleteScalingPolicy` et `DeleteScheduledAction`. Cependant, elle refuse également les actions si le cible évolutive sur lequel porte l'action dispose de la balise **`environment=production`**.

JSON

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

Contrôler l'accès aux balises

Utilisez des balises pour vérifier que le demandeur (tel qu'un utilisateur ou un rôle IAM) dispose des autorisations d'ajouter, modifier ou supprimer des balises pour des cibles évolutives.

Par exemple, vous pouvez créer une politique IAM qui permet de supprimer uniquement la balise avec la clé **`temporary`** dans les cibles évolutives.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "application-autoscaling:UntagResource",
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": { "aws:TagKeys": ["temporary"] }
            }
        }
    ]
}
```

La sécurité dans Application Auto Scaling

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez d'un centre de données et d'une architecture réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le modèle de responsabilité partagée décrit la sécurité du cloud et la sécurité dans le cloud comme suit :

- Sécurité du cloud : AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le AWS cadre des programmes de de). Pour en savoir plus sur les programmes de conformité qui s'appliquent à Application Auto Scaling, consultez la section AWS services concernés par programme de conformité et .
- Sécurité dans le cloud — Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris la sensibilité de vos données, les exigences de votre entreprise, ainsi que la législation et la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation d'Application Auto Scaling. Les rubriques suivantes vous montrent comment configurer Application Auto Scaling pour répondre à vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres AWS services qui vous aident à surveiller et à sécuriser vos ressources Application Auto Scaling.

Table des matières

Protection des données dans Application Auto Scaling

Le modèle de responsabilité AWS partagée s'applique à la protection des données dans Application Auto Scaling. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour plus d'informations sur la confidentialité des données, consultez Questions fréquentes (FAQ) sur la confidentialité des données. Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog Modèle de responsabilité partagée d'AWS et RGPD (Règlement général sur la protection des données) sur le Blog de sécuritéAWS .

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou Gestion des identités et des accès AWS (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- SSL/TLS À utiliser pour communiquer avec AWS les ressources. Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail. Pour plus d'informations sur l'utilisation des CloudTrail sentiers pour capturer AWS des activités, consultez la section Utilisation des CloudTrail sentiers dans le guide de AWS CloudTrail l'utilisateur.
- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut qu'ils contiennent Services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de modules cryptographiques validés par la norme FIPS 140-3 pour accéder AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour plus d'informations sur les points de terminaison FIPS disponibles, consultez Norme FIPS (Federal Information Processing Standard) 140-3.

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Nom. Cela inclut lorsque vous travaillez avec Application Auto Scaling ou autre

Services AWS à l'aide de la console AWS CLI, de l'API ou AWS SDKs. Toutes les données que vous entrez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez une adresse URL à un serveur externe, nous vous recommandons fortement de ne pas inclure d'informations d'identification dans l'adresse URL permettant de valider votre demande adressée à ce serveur.

Gestion des identités et des accès pour Application Auto Scaling

Gestion des identités et des accès AWS (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Des administrateurs IAM contrôlent les personnes qui peuvent être authentifiées(connectées) et autorisées (disposant d'autorisations) pour utiliser des ressources Application Auto Scaling. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

Pour une documentation IAM complète, consultez le Guide de l'utilisateur IAM.

Contrôle d'accès

Vous pouvez avoir des informations d'identification valides pour authentifier vos demandes, mais à moins d'avoir les autorisations requises, vous ne pouvez pas créer de ressources Application Auto Scaling ni accéder à de telles ressources. Par exemple, vous devez disposer des autorisations nécessaires pour créer des stratégies de dimensionnement, configurer un dimensionnement programmé, etc.

Les sections suivantes fournissent des détails sur la manière dont un administrateur IAM peut utiliser IAM pour sécuriser vos AWS ressources, en contrôlant qui peut effectuer les actions de l'API Application Auto Scaling.

Table des matières

Fonctionnement d'Application Auto Scaling avec IAM

> **Note**
>
> En décembre 2017, une mise à jour a été publiée pour Application Auto Scaling, permettant d'utiliser plusieurs rôles liés à un service pour les services intégrés à Application Auto Scaling. Des autorisations IAM spécifiques et un rôle Application Auto Scaling lié à un service (ou une fonction du service pour la mise à l'échelle automatique d'Amazon EMR) sont nécessaires pour que les utilisateurs puissent configurer la mise à l'échelle.

Avant d'utiliser IAM pour gérer l'accès à Application Auto Scaling, découvrez quelles sont les fonctions IAM qui peuvent être utilisées avec Application Auto Scaling.

Fonctionnalités IAM que vous pouvez utiliser avec Application Auto Scaling

Fonctionnalité IAM	Prise en charge d'Application Auto Scaling
Politiques basées sur l'identité	Oui
Actions de politique	Oui
Ressources de politique	Oui
Clés de condition de politique (spécifiques au service)	Oui
Politiques basées sur les ressources	Non
ACLs	Non
ABAC (identifications dans les politiques)	Partielle
Informations d'identification temporaires	Oui
Rôles de service	Oui
Rôles liés à un service	Oui

Pour obtenir une vue d'ensemble du fonctionnement d'Application Auto Scaling et d'autres Services AWS fonctionnalités avec la plupart des fonctionnalités IAM, consultez Services AWS le guide de l'utilisateur IAM consacré à leur fonctionnement avec IAM.

Stratégies Application Auto Scaling basées sur une identité

Prend en charge les politiques basées sur l'identité : oui

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez Définition d'autorisations IAM personnalisées avec des politiques gérées par le client dans le Guide de l'utilisateur IAM.

Avec les politiques IAM basées sur l'identité, vous pouvez spécifier des actions et ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. Pour découvrir tous les éléments que vous utilisez dans une politique JSON, consultez Références des éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.

Exemples de stratégies basées sur l'identité pour Application Auto Scaling

Pour voir des exemples de stratégies Application Auto Scaling basées sur l'identité, consultez Exemples de stratégies Application Auto Scaling basées sur une identité.

Actions

Prend en charge les actions de politique : oui

Dans une déclaration de politique IAM, vous pouvez spécifier une action d'API à partir de n'importe quel service prenant en charge IAM. Pour Application Auto Scaling, utilisez le préfixe suivant avec le nom de l'action d'API : `application-autoscaling:`. Par exemple : `application-autoscaling:RegisterScalableTarget`, `application-autoscaling:PutScalingPolicy` et `application-autoscaling:DeregisterScalableTarget`.

Pour préciser plusieurs actions dans une seule déclaration, séparez-les par des virgules comme l'indique l'exemple suivant.

```
"Action": [
      "application-autoscaling:DescribeScalingPolicies",
```

```
      "application-autoscaling:DescribeScalingActivities"
```

Vous pouvez aussi préciser plusieurs actions à l'aide de caractères génériques (*). Par exemple, pour spécifier toutes les actions qui commencent par le mot `Describe`, incluez l'action suivante.

```
"Action": "application-autoscaling:Describe*"
```

Pour obtenir la liste des actions d'Application Auto Scaling, consultez la section Actions définies par AWS Application Auto Scaling dans le Service Authorization Reference.

Ressources

Prend en charge les ressources de politique : oui

Dans une instruction de politiqe IAM, l'élément `Resource` spécifie l'objet ou les objets couverts par l'instruction. Pour Application Auto Scaling, chaque déclaration de politique IAM s'applique aux cibles évolutives que vous spécifiez à l'aide de leur Amazon Resource Names (ARNs).

Format de ressource ARN pour les cibles évolutives :

```
arn:aws:application-autoscaling:region:account-id:scalable-target/unique-identifier
```

Par exemple, vous pouvez indiquer une cible évolutive spécifique dans votre instruction à l'aide de son ARN, comme suit : L'ID unique (1234abcd56ab78cd901ef1234567890ab123) est une valeur attribuée par Application Auto Scaling à la cible évolutive.

```
"Resource": "arn:aws:application-autoscaling:us-east-1:123456789012:scalable-
target/1234abcd56ab78cd901ef1234567890ab123"
```

Vous pouvez spécifier toutes les instances qui appartiennent à un compte spécifique en remplaçant l'identifiant unique par un caractère générique (*) comme suit :

```
"Resource": "arn:aws:application-autoscaling:us-east-1:123456789012:scalable-target/*"
```

Pour spécifier toutes les ressources, ou si une action d'API spécifique n'est pas prise en charge ARNs, utilisez un caractère générique (*) comme `Resource` élément comme suit.

```
"Resource": "*"
```

Pour plus d'informations, consultez la section Types de ressources définis par AWS Application Auto Scaling dans le Service Authorization Reference.

Clés de condition

Prend en charge les clés de condition de politique spécifiques au service : oui

Vous pouvez spécifier des conditions dans les stratégies IAM qui contrôlent l'accès aux ressources Application Auto Scaling. L'Instruction de politique est en vigueur uniquement lorsque les conditions sont vérifiées.

Application Auto Scaling prend en charge les clés de condition définies par les services suivantes que vous pouvez utiliser dans les stratégies basées sur l'identité pour déterminer qui peut exécuter des actions d'API dans Application Auto Scaling.

- `application-autoscaling:scalable-dimension`
- `application-autoscaling:service-namespace`

Pour savoir avec quelles actions de l'API Application Auto Scaling vous pouvez utiliser une clé de condition, consultez la section Actions définies par AWS Application Auto Scaling dans le Service Authorization Reference. Pour plus d'informations sur l'utilisation des clés de condition Application Auto Scaling, consultez la section Clés de condition pour AWS Application Auto Scaling.

Pour afficher les clés de condition globales disponibles pour tous les services, consultez Clés de contexte de condition globales AWS dans le Guide de l'utilisateur IAM.

Politiques basées sur les ressources

Prend en charge les politiques basées sur les ressources : non

D'autres AWS services, tels qu'Amazon Simple Storage Service, prennent en charge les politiques d'autorisation basées sur les ressources. Par exemple, vous pouvez attacher une stratégie d'autorisation à un compartiment S3 pour gérer les autorisations d'accès à ce compartiment.

Application Auto Scaling ne prend pas en charge les stratégies basées sur une ressource.

Listes de contrôle d'accès (ACLs)

Supports ACLs : Non

Application Auto Scaling ne prend pas en charge les listes de contrôle d'accès (ACLs).

ABAC avec Application Auto Scaling

Prend en charge ABAC (identifications dans les politiques) : partiellement

Le contrôle d'accès par attributs (ABAC) est une stratégie d'autorisation qui définit des autorisations en fonction des attributs. Dans AWS, ces attributs sont appelés balises. Vous pouvez associer des balises aux entités IAM (utilisateurs ou rôles) et à de nombreuses AWS ressources. L'étiquetage des entités et des ressources est la première étape d'ABAC. Vous concevez ensuite des politiques ABAC pour autoriser des opérations quand l'identification du principal correspond à celle de la ressource à laquelle il tente d'accéder.

L'ABAC est utile dans les environnements qui connaissent une croissance rapide et pour les cas où la gestion des politiques devient fastidieuse.

Pour contrôler l'accès basé sur des étiquettes, vous devez fournir les informations d'étiquette dans l'élément de condition d'une politique utilisant les clés de condition `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`.

ABAC est possible pour les ressources qui prennent en charge les balises. Toutefois, toutes les ressources ne prennent pas en charge les balises. Les actions planifiées et les stratégies de mise à l'échelle ne prennent pas en charge les balises, mais les cibles évolutives le font. Pour de plus amples informations, veuillez consulter Prise en charge du balisage pour Application Auto Scaling.

Pour plus d'informations sur l'ABAC, consultez Qu'est-ce que le contrôle d'accès basé sur les attributs (ABAC) ? dans le Guide de l'utilisateur IAM. Pour accéder à un didacticiel décrivant les étapes de configuration de l'ABAC, consultez Utilisation du contrôle d'accès par attributs (ABAC) dans le Guide de l'utilisateur IAM.

Utilisation d'informations d'identification temporaires avec Application Auto Scaling

Prend en charge les informations d'identification temporaires : oui

Les informations d'identification temporaires fournissent un accès à court terme aux AWS ressources et sont automatiquement créées lorsque vous utilisez la fédération ou que vous changez de rôle. AWS recommande de générer dynamiquement des informations d'identification temporaires au lieu d'utiliser des clés d'accès à long terme. Pour plus d'informations, consultez Informations d'identification de sécurité temporaires dans IAM et Services AWS compatibles avec IAM dans le Guide de l'utilisateur IAM.

Rôles de service

Prend en charge les rôles de service : oui

Si votre cluster Amazon EMR utilise la mise en échelle automatique, cette fonction autorise Application Auto Scaling à endosser une fonction du service en votre nom. Comme pour un rôle lié à un service, une fonction du service permet au service d'accéder à des ressources dans d'autres services pour effectuer une action en votre nom. Les rôles de service s'affichent sur votre compte IAM et sont la propriété du compte. Cela signifie qu'un administrateur IAM peut modifier les autorisations associées à ce rôle. Toutefois, une telle action peut perturber le bon fonctionnement du service.

Application Auto Scaling prend en charge les rôles de service uniquement pour Amazon EMR. Pour la documentation sur la rôle de service EMR, consultez la section Utilisation de la mise à l'échelle automatique avec une stratégie personnalisée pour les groupes d'instances dans le Guide de gestion Amazon EMR.

> **Note**
>
> Avec l'introduction des rôles liés à un service, plusieurs rôles de service hérités ne sont plus requis, par exemple pour Amazon ECS et Spot Fleet.

Rôles liés à un service

Prend en charge les rôles liés à un service : oui

Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

Pour plus d'informations sur les rôles liés à un service pour Application Auto Scaling, veuillez consulter Rôles liés à un service pour Application Auto Scaling.

AWS politiques gérées pour Application Auto Scaling

Une politique AWS gérée est une politique autonome créée et administrée par AWS. AWS les politiques gérées sont conçues pour fournir des autorisations pour de nombreux cas d'utilisation

courants afin que vous puissiez commencer à attribuer des autorisations aux utilisateurs, aux groupes et aux rôles.

N'oubliez pas que les politiques AWS gérées peuvent ne pas accorder d'autorisations de moindre privilège pour vos cas d'utilisation spécifiques, car elles sont accessibles à tous les AWS clients. Nous vous recommandons de réduire encore les autorisations en définissant des politiques gérées par le client qui sont propres à vos cas d'utilisation.

Vous ne pouvez pas modifier les autorisations définies dans les politiques AWS gérées. Si les autorisations définies dans une politique AWS gérée sont AWS mises à jour, la mise à jour affecte toutes les identités principales (utilisateurs, groupes et rôles) auxquelles la politique est attachée. AWS est le plus susceptible de mettre à jour une politique AWS gérée lorsqu'une nouvelle Service AWS est lancée ou lorsque de nouvelles opérations d'API sont disponibles pour les services existants.

Pour plus d'informations, consultez Politiques gérées par AWS dans le Guide de l'utilisateur IAM.

AWS politique gérée : WorkSpaces applications et CloudWatch

Nom de la stratégie : AWSApplicationAutoscalingAppStreamFleetPolicy

Cette politique est attachée au rôle lié au service nommé AWSServiceRoleForApplicationAutoScaling_AppStreamFleetpour permettre à Application Auto Scaling d'appeler Amazon AppStream CloudWatch et d'effectuer le dimensionnement en votre nom.

Détails de l'autorisation

La politique d'autorisations permet à Application Auto Scaling d'effectuer les actions suivantes sur toutes les ressources associées (« Resource » : « * ») :

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

AWS politique gérée : Aurora et CloudWatch

Nom de la politique : AWSApplicationAutoscaling RDSCluster Policy

Cette politique est attachée au rôle lié au service nommé AWSServiceRoleForApplicationAutoScaling_RDSClusterpour permettre à Application Auto Scaling d'appeler Aurora CloudWatch et d'effectuer le dimensionnement en votre nom.

Détails de l'autorisation

La politique d'autorisations permet à Application Auto Scaling d'effectuer les actions suivantes sur toutes les ressources associées (« Resource » : « * ») :

- Action : `rds:AddTagsToResource`
- Action : `rds:CreateDBInstance`
- Action : `rds:DeleteDBInstance`
- Action : `rds:DescribeDBClusters`
- Action : `rds:DescribeDBInstance`
- Action : `cloudwatch:DescribeAlarms`
- Action : `cloudwatch:PutMetricAlarm`
- Action : `cloudwatch:DeleteAlarms`

AWS politique gérée : Amazon Comprehend et CloudWatch

Nom de la stratégie : AWSApplicationAutoscalingComprehendEndpointPolicy

Cette politique est attachée au rôle lié au service nommé pour permettre AWSServiceRoleForApplicationAutoScaling_ComprehendEndpointà Application Auto Scaling d'appeler Amazon Comprehend CloudWatch et d'effectuer le dimensionnement en votre nom.

Détails de l'autorisation

La politique d'autorisations permet à Application Auto Scaling d'effectuer les actions suivantes sur toutes les ressources associées (« Resource » : « * ») :

- Action : `comprehend:UpdateEndpoint`
- Action : `comprehend:DescribeEndpoint`
- Action : `cloudwatch:DescribeAlarms`
- Action : `cloudwatch:PutMetricAlarm`

- Action : `cloudwatch:DeleteAlarms`

AWS stratégie gérée : DynamoDB et CloudWatch

Nom de la politique : AWSApplicationAutoscalingDynamoDBTablePolicy

Cette politique est attachée au rôle lié au service nommé AWSServiceRoleForApplicationAutoScaling_DynamoDBTablepour permettre à Application Auto Scaling d'appeler Dynamo DBand CloudWatch et d'effectuer le dimensionnement en votre nom.

Détails de l'autorisation

La politique d'autorisations permet à Application Auto Scaling d'effectuer les actions suivantes sur toutes les ressources associées (« Resource » : « * ») :

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

AWS politique gérée : Amazon ECS et CloudWatch

Nom de la politique : AWSApplicationAutoscaling ECSService Policy

Cette politique est attachée au rôle lié au service nommé AWSServiceRoleForApplicationAutoScaling_ECSServicepour permettre à Application Auto Scaling d'appeler Amazon ECS CloudWatch et d'effectuer le dimensionnement en votre nom.

Détails de l'autorisation

La politique d'autorisations permet à Application Auto Scaling d'effectuer les actions suivantes sur toutes les ressources associées (« Resource » : « * ») :

- Action : `ecs:DescribeServices`
- Action : `ecs:UpdateService`
- Action : `cloudwatch:PutMetricAlarm`
- Action : `cloudwatch:DescribeAlarms`

- Action : `cloudwatch:GetMetricData`
- Action : `cloudwatch:DeleteAlarms`

AWS politique gérée : ElastiCache et CloudWatch

Nom de la stratégie : AWSApplicationAutoscalingElastiCacheRGPolicy

Cette politique est attachée au rôle lié au service nommé AWSServiceRoleForApplicationAutoScaling_ElastiCacheRGpour permettre à Application Auto Scaling d'appeler ElastiCache CloudWatch et d'effectuer le dimensionnement en votre nom. Ce rôle lié à un service peut être utilisé pour ElastiCache Memcached, Redis OSS et Valkey.

Détails de l'autorisation

La politique d'autorisations permet à Application Auto Scaling d'effectuer les actions suivantes sur les ressources spécifiées :

- Action : `elasticache:DescribeReplicationGroups` sur toutes les ressources
- Action : `elasticache:ModifyReplicationGroupShardConfiguration` sur toutes les ressources
- Action : `elasticache:IncreaseReplicaCount` sur toutes les ressources
- Action : `elasticache:DecreaseReplicaCount` sur toutes les ressources
- Action : `elasticache:DescribeCacheClusters` sur toutes les ressources
- Action : `elasticache:DescribeCacheParameters` sur toutes les ressources
- Action : `elasticache:ModifyCacheCluster` sur toutes les ressources
- Action : `cloudwatch:DescribeAlarms` sur la ressource `arn:aws:cloudwatch:*:*:alarm:*`
- Action : `cloudwatch:PutMetricAlarm` sur la ressource `arn:aws:cloudwatch:*:*:alarm:TargetTracking*`
- Action : `cloudwatch:DeleteAlarms` sur la ressource `arn:aws:cloudwatch:*:*:alarm:TargetTracking*`

AWS politique gérée : Amazon Keyspaces et CloudWatch

Nom de la stratégie : AWSApplicationAutoscalingCassandraTablePolicy

Cette politique est attachée au rôle lié au service nommé AWSServiceRoleForApplicationAutoScaling_CassandraTablepour permettre à Application Auto Scaling d'appeler Amazon Keyspaces CloudWatch et d'effectuer le dimensionnement en votre nom.

Détails de l'autorisation

La politique d'autorisations permet à Application Auto Scaling d'effectuer les actions suivantes sur les ressources spécifiées :

- Action : `cassandra:Select` sur les ressources suivantes :
  - `arn:*:cassandra:*:*:/keyspace/system/table/*`
  - `arn:*:cassandra:*:*:/keyspace/system_schema/table/*`
  - `arn:*:cassandra:*:*:/keyspace/system_schema_mcs/table/*`
- Action : `cassandra:Alter` sur toutes les ressources
- Action : `cloudwatch:DescribeAlarms` sur toutes les ressources
- Action : `cloudwatch:PutMetricAlarm` sur toutes les ressources
- Action : `cloudwatch:DeleteAlarms` sur toutes les ressources

AWS politique gérée : Lambda et CloudWatch

Nom de la stratégie : AWSApplicationAutoscalingLambdaConcurrencyPolicy

Cette politique est attachée au rôle lié au service nommé AWSServiceRoleForApplicationAutoScaling_LambdaConcurrencypour permettre à Application Auto Scaling d'appeler Lambda CloudWatch et d'effectuer le dimensionnement en votre nom.

Détails de l'autorisation

La politique d'autorisations permet à Application Auto Scaling d'effectuer les actions suivantes sur toutes les ressources associées (« Resource » : « * ») :

- Action : `lambda:PutProvisionedConcurrencyConfig`
- Action : `lambda:GetProvisionedConcurrencyConfig`
- Action : `lambda:DeleteProvisionedConcurrencyConfig`
- Action : `cloudwatch:DescribeAlarms`
- Action : `cloudwatch:PutMetricAlarm`

- Action : `cloudwatch:DeleteAlarms`

AWS politique gérée : Amazon MSK et CloudWatch

Nom de la stratégie : AWSApplicationAutoscalingKafkaClusterPolicy

Cette politique est attachée au rôle lié au service nommé AWSServiceRoleForApplicationAutoScaling_KafkaClusterpour permettre à Application Auto Scaling d'appeler Amazon MSK CloudWatch et d'effectuer le dimensionnement en votre nom.

Détails de l'autorisation

La politique d'autorisations permet à Application Auto Scaling d'effectuer les actions suivantes sur toutes les ressources associées (« Resource » : « * ») :

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

AWS politique gérée : Neptune et CloudWatch

Nom de la stratégie : AWSApplicationAutoscalingNeptuneClusterPolicy

Cette politique est attachée au rôle lié au service nommé AWSServiceRoleForApplicationAutoScaling_NeptuneClusterpour permettre à Application Auto Scaling d'appeler Neptune CloudWatch et d'effectuer le dimensionnement en votre nom.

Détails de l'autorisation

La politique d'autorisations permet à Application Auto Scaling d'effectuer les actions suivantes sur les ressources spécifiées :

- Action : `rds:ListTagsForResource` sur toutes les ressources
- Action : `rds:DescribeDBInstances` sur toutes les ressources
- Action : `rds:DescribeDBClusters` sur toutes les ressources

- Action : `rds:DescribeDBClusterParameters` sur toutes les ressources
- Action : `cloudwatch:DescribeAlarms` sur toutes les ressources
- Action :`rds:AddTagsToResource`sur les ressources avec le préfixe`autoscaled-reader`dans le moteur de base de données Amazon Neptune (`"Condition":{"StringEquals":{"rds:DatabaseEngine":"neptune"}}`)
- Action : `rds:CreateDBInstance`sur les ressources avec le préfixe`autoscaled-reader`dans tous les clusters DB (`"Resource":"arn:*:rds:*:*:db:autoscaled-reader*"`, `"arn:aws:rds:*:*:cluster:*"`) dans le moteur de base de données Amazon Neptune (`"Condition":{"StringEquals":{"rds:DatabaseEngine":"neptune"}}`)
- Action : `rds:DeleteDBInstance` sur la ressource `arn:aws:rds:*:*:db:autoscaled-reader*`
- Action : `cloudwatch:PutMetricAlarm` sur la ressource `arn:aws:cloudwatch:*:*:alarm:TargetTracking*`
- Action : `cloudwatch:DeleteAlarms` sur la ressource `arn:aws:cloudwatch:*:*:alarm:TargetTracking*`

AWS politique gérée : SageMaker IA et CloudWatch

Nom de la stratégie : AWSApplicationAutoscalingSageMakerEndpointPolicy

Cette politique est attachée au rôle lié au service nommé AWSServiceRoleForApplicationAutoScaling_SageMakerEndpointpour permettre à Application Auto Scaling d'appeler SageMaker AI CloudWatch et d'effectuer le dimensionnement en votre nom.

Détails de l'autorisation

La politique d'autorisations permet à Application Auto Scaling d'effectuer les actions suivantes sur les ressources spécifiées :

- Action : `sagemaker:DescribeEndpoint` sur toutes les ressources
- Action : `sagemaker:DescribeEndpointConfig` sur toutes les ressources
- Action : `sagemaker:DescribeInferenceComponent` sur toutes les ressources
- Action : `sagemaker:UpdateEndpointWeightsAndCapacities` sur toutes les ressources
- Action : `sagemaker:UpdateInferenceComponentRuntimeConfig` sur toutes les ressources
- Action : `cloudwatch:DescribeAlarms` sur toutes les ressources

- Action : `cloudwatch:GetMetricData` sur toutes les ressources
- Action : `cloudwatch:PutMetricAlarm` sur la ressource `arn:aws:cloudwatch:*:*:alarm:TargetTracking*`
- Action : `cloudwatch:DeleteAlarms` sur la ressource `arn:aws:cloudwatch:*:*:alarm:TargetTracking*`

AWS politique gérée : EC2 Spot Fleet et CloudWatch

Nom de la politique : AWSApplicationAutoscaling EC2 SpotFleetRequestPolicy

Cette politique est attachée au rôle lié au service nommé AWSServiceRoleForApplicationAutoScaling_EC2 SpotFleetRequest pour permettre à Application Auto Scaling d'appeler Amazon EC2 CloudWatch et d'effectuer le dimensionnement en votre nom.

Détails de l'autorisation

La politique d'autorisations permet à Application Auto Scaling d'effectuer les actions suivantes sur toutes les ressources associées (« Resource » : « * ») :

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

AWS politique gérée : WorkSpaces et CloudWatch

Nom de la stratégie : AWSApplicationAutoscalingWorkSpacesPoolPolicy

Cette politique est attachée au rôle lié au service nommé AWSServiceRoleForApplicationAutoScaling_WorkSpacesPoolpour permettre à Application Auto Scaling d'appeler WorkSpaces CloudWatch et d'effectuer le dimensionnement en votre nom.

Détails de l'autorisation

La politique d'autorisations permet à Application Auto Scaling d'effectuer les actions suivantes sur les ressources spécifiées :

- Action : `workspaces:DescribeWorkspacesPools` sur toutes les ressources du même compte que le SLR
- Action : `workspaces:UpdateWorkspacesPool` sur toutes les ressources du même compte que le SLR
- Action : `cloudwatch:DescribeAlarms` sur toutes les alarmes du même compte que le reflex
- Action : `cloudwatch:PutMetricAlarm` sur toutes les alarmes provenant du même compte que le reflex, dont le nom commence par TargetTracking
- Action : `cloudwatch:DeleteAlarms` sur toutes les alarmes provenant du même compte que le reflex, dont le nom commence par TargetTracking

AWS politique gérée : ressources personnalisées et CloudWatch

Nom de la stratégie : AWSApplicationAutoScalingCustomResourcePolicy

Cette politique est attachée au rôle lié au service nommé AWSServiceRoleForApplicationAutoScaling_CustomResourcepour permettre à Application Auto Scaling d'appeler vos ressources personnalisées disponibles via API Gateway CloudWatch et d'effectuer le dimensionnement en votre nom.

Détails de l'autorisation

La politique d'autorisations permet à Application Auto Scaling d'effectuer les actions suivantes sur toutes les ressources associées (« Resource » : « * ») :

- Action : `execute-api:Invoke`
- Action : `cloudwatch:DescribeAlarms`
- Action : `cloudwatch:PutMetricAlarm`
- Action : `cloudwatch:DeleteAlarms`

Application Auto Scaling met à jour les politiques AWS gérées

Consultez les détails des mises à jour apportées aux politiques AWS gérées pour Application Auto Scaling depuis que ce service a commencé à suivre ces modifications. Pour obtenir des alertes automatiques concernant les modifications apportées à cette page, abonnez-vous au flux RSS de la page Historique du document d'Application Auto Scaling.

Modifier	Description	Date
AWSApplicationAutoscalingElastiCacheRGPolicy— Mettre à jour une politique existante	Ajout de l'autorisation d'appeler l'action d' ElastiCache `ModifyCacheCluster` API pour prendre en charge le dimensionnement automatique de Memcached.	10 avril 2025
AWSApplicationECSServicePolitique de mise à l'échelle automatique — Mettre à jour une politique existante	Ajout de l'autorisation d'appeler l'action de l' CloudWatch `GetMetricData` API afin de prendre en charge le dimensionnement prédictif.	21 novembre 2024
AWSApplicationAutoscalingWorkSpacesPoolPolicy : nouvelle politique	Ajout d'une politique gérée pour Amazon WorkSpaces. Cette politique est associée à un rôle lié à un service qui permet à Application Auto Scaling d'appeler WorkSpaces CloudWatch et d'effectuer le dimensionnement en votre nom.	24 juin 2024
AWSApplicationAutoscalingSageMakerEndpointPolicy : mise à jour d'une politique existante	Des autorisations ont été ajoutées pour appeler l' SageMaker IA `DescribeInferenceComponent` et les actions d'`UpdateInferenceComponentRuntimeConfig` API afin de garantir la compatibilité et la mise à l'échelle automatique des ressources d' SageMaker IA en vue d'une intégration	13 novembre 2023

Modifier	Description	Date
	à venir. La politique limite également désormais les actions de l'`DeleteAlarms` API CloudWatch `PutMetricAlarm` et de l'API aux CloudWatch alarmes utilisées avec les politiques de dimensionnement du suivi des cibles.	
AWSApplicationAutoscalingNeptuneClusterPolicy : nouvelle politique	Ajout d'une politique gérée pour Neptune. Cette politique est associée à un rôle lié à un service qui permet à Application Auto Scaling d'appeler Neptune CloudWatch et d'effectuer le dimensionnement en votre nom.	6 octobre 2021
AWSApplicationPolitique de mise à l'échelle automatique — Nouvelle RDSCluster politique	Ajout d'une politique gérée pour ElastiCache. Cette politique est associée à un rôle lié à un service qui permet à Application Auto Scaling d'appeler ElastiCache CloudWatch et d'effectuer le dimensionnement en votre nom.	19 août 2021
Application Auto Scaling a commencé à assurer le suivi des modifications	Application Auto Scaling a commencé à suivre les modifications apportées AWS à ses politiques gérées.	19 août 2021

Rôles liés à un service pour Application Auto Scaling

Application Auto Scaling utilise des rôles liés aux services pour obtenir les autorisations dont elle a besoin pour appeler d'autres AWS services en votre nom. Un rôle lié à un service est un type unique de rôle Gestion des identités et des accès AWS (IAM) directement lié à un service. AWS Les rôles liés à un service constituent un moyen sécurisé de déléguer des autorisations aux AWS services, car seul le service lié peut assumer un rôle lié au service.

Pour les services qui s'intègrent à Application Auto Scaling, Application Auto Scaling crée des rôles liés à un service pour vous. Il y a un rôle lié à un service pour chaque service. Chaque rôle lié à un service approuve le principal du service précisé afin d'endosser ce rôle. Pour de plus amples informations, veuillez consulter Référence ARN de rôle lié à un service.

Application Auto Scaling inclut toutes les autorisations nécessaires pour chaque rôle lié à un service. Ces autorisations gérées sont créées et gérées par Application Auto Scaling, et elles définissent les actions autorisées pour chaque type de ressource. Pour plus d'informations sur les autorisations accordées par chaque rôle, consultez AWS politiques gérées pour Application Auto Scaling.

Table des matières

Autorisations requises pour créer un rôle lié à un service

Application Auto Scaling a besoin d'autorisations pour créer un rôle lié à un service la première fois qu'un de vos utilisateurs Compte AWS appelle `RegisterScalableTarget` pour un service donné. Application Auto Scaling crée un rôle lié à un service pour le service cible dans votre compte, si ce rôle n'existe pas déjà. Le rôle lié à un service accorde des autorisations à Application Auto Scaling, afin qu'il puisse appeler le service cible en votre nom.

Pour que la création automatique de rôle réussisse, les utilisateurs doivent avoir l'autorisation pour l'action `iam:CreateServiceLinkedRole`.

```
"Action": "iam:CreateServiceLinkedRole"
```

La stratégie basée sur l'identité suivante accorde l'autorisation de créer un rôle lié à un service pour le parc d'instances Spot. Vous pouvez spécifier le rôle lié à un service dans le champ `Resource` de la stratégie en tant qu'ARN et le principal du service pour votre rôle lié à un service en tant que condition, comme illustré. Pour connaître l'ARN pour chaque service, consultez Référence ARN de rôle lié à un service.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-
service-role/ec2.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_EC2SpotFleetRequest",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName":"ec2.application-
autoscaling.amazonaws.com"
                }
            }
        }
    ]
}
```

> **Note**
>
> La clé de condition IAM `iam:AWSServiceName` précise le principal du service auquel le rôle est attaché, qui est indiqué dans cet exemple de stratégie comme *`ec2.application-autoscaling`*`.amazonaws.com`. N'essayez pas de deviner le principal du service. Pour afficher le principal d'un service, consultez Services AWS que vous pouvez utiliser avec Application Auto Scaling.

Création de rôles liés à un service (Automatique)

Vous n'avez pas besoin de créer manuellement un rôle lié à un service. Application Auto Scaling crée pour vous le rôle lié à un service approprié lorsque vous appelez `RegisterScalableTarget`. Par exemple, si vous définissez la fonction de mise à l'échelle automatique pour un Amazon ECS service, Application Auto Scaling crée le rôle `AWSServiceRoleForApplicationAutoScaling_ECSService`.

Création de rôles liés à un service (Manuel)

Pour créer le rôle lié à un service, vous pouvez utiliser la console IAM ou l'API AWS CLI IAM. Pour plus d'informations, consultez la section Créer un rôle lié à un service dans le guide de l'utilisateur IAM.

Pour créer un rôle lié à un service (AWS CLI)

Utilisez la create-service-linked-rolecommande suivante pour créer le rôle lié au service Application Auto Scaling. Dans la demande, précisez le nom de service « prefix ».

Pour trouver le préfixe du nom de service, reportez-vous aux informations sur le principal du service pour le rôle lié au service pour chaque service dans la section Services AWS que vous pouvez utiliser avec Application Auto Scaling. Le nom du service et le principal du service partagent le même préfixe. Par exemple, pour créer le rôle AWS Lambda lié à un service, utilisez. `lambda.application-autoscaling.amazonaws.com`

```
aws iam create-service-linked-role --aws-service-name prefix.application-
autoscaling.amazonaws.com
```

Modification des rôles liés à un service

Avec les rôles liés à un service créés par Application Auto Scaling, vous ne pouvez modifier que leurs descriptions. Pour plus d'informations, consultez la section Modification d'un rôle lié à un service dans le Guide de l'utilisateur IAM.

Suppression des rôles liés à un service

Si vous n'utilisez plus Application Auto Scaling avec un service pris en charge, nous vous recommandons de supprimer le rôle lié à un service correspondant.

Vous pouvez supprimer un rôle lié à un service uniquement après la suppression préalable des ressources AWS connexes. Cela vous évite de révoquer involontairement les autorisations

Application Auto Scaling sur vos ressources. Pour plus d'informations, consultez la documentation de la ressource scalable. Par exemple, pour supprimer un service Amazon ECS, consultez Supprimer un service Amazon ECS dans le manuel Amazon Elastic Container Service Developer Guide.

Vous pouvez utiliser IAM pour supprimer le rôle lié à un service. Pour plus d'informations, consultez la section Suppression d'un rôle lié à un service dans le Guide de l'utilisateur IAM.

Une fois que vous avez supprimé un rôle lié à un service, Application Auto Scaling recrée le rôle lorsque vous appelez `RegisterScalableTarget`.

Régions prises en charge pour les rôles liés à un service pour Application Auto Scaling

Application Auto Scaling prend en charge l'utilisation de rôles liés à un service dans toutes les AWS régions où le service est disponible.

Référence ARN de rôle lié à un service

Le tableau suivant répertorie l'Amazon Resource Name (ARN) du rôle lié à un service pour chaque Service AWS rôle compatible avec Application Auto Scaling.

Service	ARN
AppStream 2,0	`arn:aws:iam:: 012345678910 :role/aws-service-role/appstream.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_AppStreamFleet`
Aurora	`arn:aws:iam:: 012345678910 :role/aws-service-role/rds.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_RDSCluster`
Comprehend	`arn:aws:iam:: 012345678910 :role/aws-service-role/comprehend.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_ComprehendEndpoint`
DynamoDB	`arn:aws:iam:: 012345678910 :role/aws-service-role/dynamodb.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_DynamoDBTable`

Service	ARN
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
Neptune	arn:aws:iam:: *012345678910* :role/aws-service-role/neptune.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_NeptuneCluster
SageMaker IA	arn:aws:iam:: *012345678910* :role/aws-service-role/sagemaker.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_SageMakerEndpoint

Service	ARN
Spot Fleets	`arn:aws:iam:: 012345678910 :role/aws-service-role/ec2.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_EC2SpotFleetRequest`
WorkSpaces	`arn:aws:iam:: 012345678910 :role/aws-service-role/workspaces.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_WorkSpacesPool`
Ressources personnalisées	`arn:aws:iam:: 012345678910 :role/aws-service-role/custom-resource.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_CustomResource`

> **Note**
>
> Vous pouvez spécifier l'ARN d'un rôle lié à un service pour la `RoleARN` propriété d'une AWS::ApplicationAutoScaling::ScalableTargetressource dans vos modèles de CloudFormation pile, même si le rôle lié à un service spécifié n'existe pas encore. Application Auto Scaling crée automatiquement le rôle pour vous.

Exemples de stratégies Application Auto Scaling basées sur une identité

Par défaut, un nouvel utilisateur n' Compte AWS est pas autorisé à faire quoi que ce soit. Un administrateur IAM doit créer et assigner des politiques IAM qui accordent à une identité IAM (utilisateur ou rôle, par exemple) l'autorisation d'effectuer des actions d'API dans Application Auto Scaling.

Pour savoir comment créer une stratégie IAM à l'aide des exemples de documents de stratégie JSON suivants, consultez Création de stratégies dans l'onglet JSON dans le Guide de l'utilisateur IAM.

Table des matières

- Autorisations requises pour les actions d'API sur les services cibles et CloudWatch
- Autorisations pour travailler dans AWS Management Console

Autorisations requises pour les actions de l'API Application Auto Scaling

Les stratégies suivantes accordent des autorisations pour les cas d'utilisation courants lors de l'appel de l'API Application Auto Scaling. Reportez-vous à cette section lorsque vous rédigez des stratégies basées sur l'identité. Chaque stratégie accorde l'autorisation d'effectuer tout ou partiellement les actions de l'API Application Auto Scaling. Vous devez également vous assurer que les utilisateurs finaux disposent des autorisations pour le service cible CloudWatch (voir la section suivante pour plus de détails).

La stratégie basée sur l'identité suivante accorde l'autorisation d'effectuer les actions de l'API Application Auto Scaling.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "application-autoscaling:*"
            ],
            "Resource": "*"
        }
    ]
}
```

La stratégie basée sur l'identité suivante accorde l'autorisation d'effectuer toutes les actions de l'API Application Auto Scaling requises pour configurer les stratégies de mise à l'échelle et non pas les actions programmées.

JSON

```
{
    "Version":"2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "application-autoscaling:RegisterScalableTarget",
              "application-autoscaling:DescribeScalableTargets",
              "application-autoscaling:DeregisterScalableTarget",
              "application-autoscaling:PutScalingPolicy",
              "application-autoscaling:DescribeScalingPolicies",
              "application-autoscaling:DescribeScalingActivities",
              "application-autoscaling:DeleteScalingPolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

La stratégie basée sur l'identité suivante accorde l'autorisation d'effectuer toutes les actions de l'API Application Auto Scaling requises pour configurer les actions programmées et non les stratégies de mise à l'échelle.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "application-autoscaling:RegisterScalableTarget",
              "application-autoscaling:DescribeScalableTargets",
              "application-autoscaling:DeregisterScalableTarget",
              "application-autoscaling:PutScheduledAction",
              "application-autoscaling:DescribeScheduledActions",
              "application-autoscaling:DescribeScalingActivities",
              "application-autoscaling:DeleteScheduledAction"
            ],
            "Resource": "*"
        }
    ]
}
```

Autorisations requises pour les actions d'API sur les services cibles et CloudWatch

Pour configurer et utiliser correctement Application Auto Scaling avec le service cible, les utilisateurs finaux doivent disposer d'autorisations pour Amazon CloudWatch et pour chaque service cible pour lequel ils configureront le dimensionnement. Utilisez les politiques suivantes pour accorder les autorisations minimales requises pour travailler avec les services cibles et CloudWatch.

Table des matières

AppStream Flottes 2.0

La politique basée sur l'identité suivante accorde des autorisations à toutes les actions AppStream 2.0 et CloudWatch API requises.

JSON

```
{
    "Version":"2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "appstream:DescribeFleets",
              "appstream:UpdateFleet",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Réplicas Aurora

La politique basée sur l'identité suivante accorde des autorisations à toutes les actions Aurora et CloudWatch API requises.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "rds:AddTagsToResource",
              "rds:CreateDBInstance",
              "rds:DeleteDBInstance",
              "rds:DescribeDBClusters",
              "rds:DescribeDBInstances",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Classification de documents et points de terminaison de module de reconnaissance d'entité Amazon Comprehend

La politique basée sur l'identité suivante accorde des autorisations à toutes les actions Amazon Comprehend CloudWatch et API requises.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "comprehend:UpdateEndpoint",
              "comprehend:DescribeEndpoint",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Tables DynamoDB et index secondaires globaux

La politique basée sur l'identité suivante accorde des autorisations à toutes les actions DynamoDB et CloudWatch API requises.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "dynamodb:DescribeTable",
              "dynamodb:UpdateTable",
```

```
                "cloudwatch:DescribeAlarms",
                "cloudwatch:PutMetricAlarm",
                "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Services ECS

La politique basée sur l'identité suivante accorde des autorisations à toutes les actions ECS et CloudWatch API requises.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecs:DescribeServices",
                "ecs:UpdateService",
                "cloudwatch:DescribeAlarms",
                "cloudwatch:PutMetricAlarm",
                "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

ElastiCache groupes de réplication

La politique basée sur l'identité suivante accorde des autorisations à toutes ElastiCache les actions CloudWatch d'API requises.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "elasticache:ModifyReplicationGroupShardConfiguration",
              "elasticache:IncreaseReplicaCount",
              "elasticache:DecreaseReplicaCount",
              "elasticache:DescribeReplicationGroups",
              "elasticache:DescribeCacheClusters",
              "elasticache:DescribeCacheParameters",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Clusters Amazon EMR

La politique basée sur l'identité suivante accorde des autorisations à toutes les actions Amazon EMR et CloudWatch API requises.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "elasticmapreduce:ModifyInstanceGroups",
              "elasticmapreduce:ListInstanceGroups",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
```

```
            ],
            "Resource": "*"
        }
    ]
}
```

Tables Amazon Keyspaces

La politique basée sur l'identité suivante accorde des autorisations à toutes les actions Amazon Keyspaces et CloudWatch API requises.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "cassandra:Select",
              "cassandra:Alter",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Fonctions Lambda

La politique basée sur l'identité suivante accorde des autorisations à toutes les actions Lambda et CloudWatch API requises.

JSON

```
{
    "Version":"2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "lambda:PutProvisionedConcurrencyConfig",
              "lambda:GetProvisionedConcurrencyConfig",
              "lambda:DeleteProvisionedConcurrencyConfig",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Stockage de l'agent Amazon Managed Streaming for Apache Kafka (MSK)

La politique basée sur l'identité suivante accorde des autorisations à toutes les actions Amazon MSK et CloudWatch API requises.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "kafka:DescribeCluster",
              "kafka:DescribeClusterOperation",
              "kafka:UpdateBrokerStorage",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Clusters Neptune

La politique basée sur l'identité suivante accorde des autorisations pour toutes les actions Neptune et CloudWatch API requises.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "rds:AddTagsToResource",
              "rds:CreateDBInstance",
              "rds:DescribeDBInstances",
              "rds:DescribeDBClusters",
              "rds:DescribeDBClusterParameters",
              "rds:DeleteDBInstance",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

SageMaker Points de terminaison IA

La politique basée sur l'identité suivante accorde des autorisations à toutes les actions d' SageMaker IA et CloudWatch d'API requises.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
```

```
            "Action": [
              "sagemaker:DescribeEndpoint",
              "sagemaker:DescribeEndpointConfig",
              "sagemaker:DescribeInferenceComponent",
              "sagemaker:UpdateEndpointWeightsAndCapacities",
              "sagemaker:UpdateInferenceComponentRuntimeConfig",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Flottes de véhicules Spot (Amazon EC2)

La politique basée sur l'identité suivante accorde des autorisations à toutes les actions de Spot Fleet et CloudWatch d'API requises.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "ec2:DescribeSpotFleetRequests",
              "ec2:ModifySpotFleetRequest",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Ressources personnalisées

La stratégie basée sur l'identité suivante accorde l'autorisation d'exécuter l'API Gateway API. Cette politique accorde également des autorisations pour toutes les CloudWatch actions requises.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "execute-api:Invoke",
              "cloudwatch:DescribeAlarms",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```

Autorisations pour travailler dans AWS Management Console

Il n'existe pas de console autonome d'Application Auto Scaling. La plupart des services qui s'intègrent avec Application Auto Scaling ont des fonctions dédiées pour vous aider à configurer la mise à l'échelle avec leur console.

Dans la plupart des cas, chaque service fournit des politiques IAM AWS gérées (prédéfinies) qui définissent l'accès à sa console, y compris les autorisations relatives aux actions de l'API Application Auto Scaling. Pour de plus amples informations, reportez-vous à la documentation du service dont vous souhaitez utiliser la console.

Vous pouvez également créer vos propres stratégies IAM personnalisées afin de donner aux utilisateurs des autorisations précises pour afficher et utiliser des actions spécifiques de l'API Application Auto Scaling sur la AWS Management Console. Vous pouvez utiliser les exemples de politiques présentés dans les sections précédentes ; toutefois, ils sont conçus pour les demandes effectuées avec le AWS CLI ou un SDK. Puisque la console utilise des actions d'API supplémentaires pour ses fonctions, ces politiques peuvent ne pas fonctionner comme escompté. Par exemple, pour

configurer le step scaling, les utilisateurs peuvent avoir besoin d'autorisations supplémentaires pour créer et gérer des CloudWatch alarmes.

> **Tip**
>
> Pour vous aider à découvrir les actions d'API requises pour exécuter des tâches sur la console, vous pouvez utiliser un service tel que AWS CloudTrail. Pour plus d'informations, consultez le Guide de l'utilisateur AWS CloudTrail.

La stratégie basée sur l'identité suivante accorde l'autorisation de configurer des politiques de mise à l'échelle pour le parc d'instances Spot. Outre les autorisations IAM pour Spot Fleet, l'utilisateur de la console qui accède aux paramètres de dimensionnement du parc depuis la EC2 console Amazon doit disposer des autorisations appropriées pour les services qui prennent en charge le dimensionnement dynamique.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:*",
                "ec2:DescribeSpotFleetRequests",
                "ec2:ModifySpotFleetRequest",
                "cloudwatch:DeleteAlarms",
                "cloudwatch:DescribeAlarmHistory",
                "cloudwatch:DescribeAlarms",
                "cloudwatch:DescribeAlarmsForMetric",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics",
                "cloudwatch:PutMetricAlarm",
                "cloudwatch:DisableAlarmActions",
                "cloudwatch:EnableAlarmActions",
                "sns:CreateTopic",
                "sns:Subscribe",
                "sns:Get*",
                "sns:List*"
            ],
```

```
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-
service-role/ec2.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_EC2SpotFleetRequest",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName":"ec2.application-
autoscaling.amazonaws.com"
                }
            }
        }
    ]
}
```

Cette politique permet aux utilisateurs de la console de consulter et de modifier les politiques de dimensionnement dans la EC2 console Amazon, ainsi que de créer et de gérer des CloudWatch alarmes dans la CloudWatch console.

Vous pouvez ajuster les actions de l'API pour limiter l'accès des utilisateurs. Par exemple, le remplacement de `application-autoscaling:*` par `application-autoscaling:Describe*` signifie que l'utilisateur dispose d'un accès en lecture seule.

Vous pouvez également ajuster les CloudWatch autorisations selon les besoins pour limiter l'accès des utilisateurs aux CloudWatch fonctionnalités. Pour plus d'informations, consultez la section Autorisations nécessaires pour la CloudWatch console dans le guide de CloudWatch l'utilisateur Amazon.

Résolution des problèmes liés à l'accès à Application Auto Scaling

Si vous rencontrez une exception `AccessDeniedException` ou des difficultés similaires lors de l'utilisation d'Application Auto Scaling, consultez les informations de cette section.

Je ne suis pas autorisé à effectuer une action dans Application Auto Scaling

Si vous recevez un message `AccessDeniedException` lors de l'appel d'une opération d' AWS API, cela signifie que les informations d'identification Gestion des identités et des accès AWS (IAM) que vous utilisez ne disposent pas des autorisations requises pour effectuer cet appel.

L'exemple d'erreur suivant se produit lorsque l'utilisateur `mateojackson` tente d'afficher les détails d'une cible évolutive, mais ne dispose pas de l'autorisation `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
```

Si vous recevez cette erreur ou des erreurs similaires, vous devez contacter votre administrateur pour obtenir de l'aide.

L'administrateur de votre compte devra s'assurer que vous êtes autorisé à accéder à toutes les actions d'API utilisées par Application Auto Scaling pour accéder aux ressources du service cible et CloudWatch. Les autorisations requises sont différentes selon les ressources avec lesquelles vous travaillez. Application Auto Scaling nécessite également l'autorisation pour créer un rôle lié à un service la première fois qu'un utilisateur configure la mise à l'échelle pour une ressource donnée.

Je suis un administrateur et ma stratégie IAM a renvoyé une erreur ou ne fonctionne pas comme prévu

Outre les actions Application Auto Scaling, vos politiques IAM doivent accorder des autorisations pour appeler le service cible et CloudWatch. Si un utilisateur ou une application ne dispose pas de ces autorisations supplémentaires, son accès peut être refusé de façon inattendue. Pour écrire des politiques IAM pour les utilisateurs et les applications dans vos comptes, consultez les informations dans Exemples de stratégies Application Auto Scaling basées sur une identité.

Pour plus d'informations sur la façon dont la validation est effectuée, consultez Validation des autorisations pour les appels d'API Application Auto Scaling sur les ressources cibles.

Notez que certains problèmes d'autorisation peuvent également être dus à un problème de création des rôles liés à un service utilisés par Application Auto Scaling. Pour plus d'informations sur la création de ces rôles liés à un service, consultez Rôles liés à un service pour Application Auto Scaling.

Validation des autorisations pour les appels d'API Application Auto Scaling sur les ressources cibles

Pour envoyer des demandes autorisées aux actions de l'API Application Auto Scaling, l'appelant de l'API doit être autorisé à accéder aux AWS ressources dans le service cible et dans CloudWatch.

Application Auto Scaling valide les autorisations pour les demandes associées à la fois au service cible et CloudWatch avant de traiter la demande. Pour ce faire, nous émettons une série d'appels pour valider les autorisations IAM sur les ressources cibles. Lorsqu'une réponse est renvoyée, elle est lue par Application Auto Scaling. Si les autorisations IAM n'autorisent pas une action donnée, Application Auto Scaling échoue la demande et renvoie une erreur à l'utilisateur contenant des informations sur l'autorisation manquante. Cela garantit que la configuration de mise à l'échelle que l'utilisateur souhaite déployer fonctionne comme prévu et qu'une erreur utile est renvoyée en cas d'échec de la demande.

À titre d'exemple de la manière dont cela fonctionne, les informations suivantes fournissent des détails sur la manière dont Application Auto Scaling effectue les validations d'autorisations avec Aurora et CloudWatch.

Lorsqu'un utilisateur appelle l'API `RegisterScalableTarget` contre un cluster de base de données Aurora, Application Auto Scaling effectue tous les contrôles suivants pour vérifier que l'utilisateur dispose des autorisations requises (en gras).

- RDS:create DBInstance : pour déterminer si l'utilisateur dispose de cette autorisation, nous envoyons une demande à l'opération `CreateDBInstance` API pour tenter de créer une instance de base de données avec des paramètres non valides (ID d'instance vide) dans le cluster de base de données Aurora spécifié par l'utilisateur. Pour un utilisateur autorisé, l'API renvoie une réponse avec un code d'erreur `InvalidParameterValue` après avoir vérifié la demande. Cependant, pour un utilisateur non autorisé, nous obtenons une erreur `AccessDenied` et nous faisons échouer la demande d'Application Auto Scaling avec une erreur `ValidationException` pour l'utilisateur qui énumère les autorisations manquantes.
- RDS:delete DBInstance : nous envoyons un ID d'instance vide à l'`DeleteDBInstance`opération API. Pour un utilisateur autorisé, cette demande aboutit à une erreur `InvalidParameterValue`. Pour un utilisateur non autorisé, le résultat est `AccessDenied` et une exception de validation est envoyée à l'utilisateur (même traitement que celui décrit dans le premier point).
- rds : AddTagsToResource : Étant donné que l'opération d'`AddTagsToResource`API nécessite un nom de ressource Amazon (ARN), il est nécessaire de spécifier une ressource « fictive » en utilisant un identifiant de compte (12345) et un identifiant d'instance factice () non valides pour créer l'ARN (non-existing-db). `arn:aws:rds:us-east-1:12345:db:non-existing-db` Pour un utilisateur autorisé, cette demande aboutit à une erreur `InvalidParameterValue`. Pour un utilisateur non autorisé, le résultat est `AccessDenied` et une exception de validation est envoyée à l'utilisateur.

- RDS:describe DBClusters : nous décrivons le nom du cluster pour la ressource en cours d'enregistrement pour le dimensionnement automatique. Pour un utilisateur autorisé, nous obtenons un résultat de description valide. Pour un utilisateur non autorisé, le résultat est `AccessDenied` et une exception de validation est envoyée à l'utilisateur.
- RDS:describe DBInstances : nous appelons l'`DescribeDBInstances`API avec un `db-cluster-id` filtre qui filtre en fonction du nom de cluster fourni par l'utilisateur pour enregistrer la cible évolutive. Pour un utilisateur autorisé, nous sommes autorisés à décrire toutes les instances de la base de données dans le cluster de bases de données. Pour un utilisateur non autorisé, cet appel aboutit à `AccessDenied` et envoie une exception de validation à l'utilisateur.
- cloudwatch : PutMetricAlarm : Nous appelons l'`PutMetricAlarm`API sans aucun paramètre. Parce que le nom de l'alarme est manquant, la demande aboutit à `ValidationError` pour un utilisateur autorisé. Pour un utilisateur non autorisé, le résultat est `AccessDenied` et une exception de validation est envoyée à l'utilisateur.
- cloudwatch : DescribeAlarms : Nous appelons l'`DescribeAlarms`API avec la valeur maximale du nombre d'enregistrements fixée à 1. Pour un utilisateur autorisé, nous attendons des informations sur une alarme dans la réponse. Pour un utilisateur non autorisé, cet appel aboutit à `AccessDenied` et envoie une exception de validation à l'utilisateur.
- cloudwatch : DeleteAlarms : Comme `PutMetricAlarm` ci-dessus, nous ne fournissons aucun paramètre à `DeleteAlarms` demander. Comme il manque un nom d'alarme dans la demande, cet appel échoue avec `ValidationError` pour un utilisateur autorisé. Pour un utilisateur non autorisé, le résultat est `AccessDenied` et une exception de validation est envoyée à l'utilisateur.

Chaque fois que l'une de ces exceptions de validation se produit, elle est enregistrée. Vous pouvez prendre des mesures pour identifier manuellement les appels qui n'ont pas été validés en utilisant AWS CloudTrail. Pour plus d'informations, consultez le Guide de l'utilisateur AWS CloudTrail.

> **Note**
>
> Si vous recevez des alertes concernant des événements Application Auto Scaling en utilisant CloudTrail, ces alertes incluront les appels d'Application Auto Scaling pour valider les autorisations des utilisateurs par défaut. Pour filtrer ces alertes, utilisez le champ `invokedBy`, qui contiendra `application-autoscaling.amazonaws.com` pour ces vérifications de validation.

Accédez à Application Auto Scaling à l'aide des points de terminaison VPC de l'interface

Vous pouvez l'utiliser AWS PrivateLink pour créer une connexion privée entre votre VPC et Application Auto Scaling. Vous pouvez accéder à Application Auto Scaling comme si elle se trouvait dans votre VPC, sans utiliser de passerelle Internet, de périphérique NAT, de connexion VPN ou Direct Connect de connexion. Les instances de votre VPC n'ont pas besoin d'adresses IP publiques pour accéder à Application Auto Scaling.

Vous établissez cette connexion privée en créant un point de terminaison d'interface optimisé par AWS PrivateLink. Nous créons une interface réseau de point de terminaison dans chaque sous-réseau que vous activez pour le point de terminaison d'interface. Il s'agit d'interfaces réseau gérées par les demandeurs qui servent de point d'entrée pour le trafic destiné à Application Auto Scaling.

Pour plus d'informations, consultez la section Accès Services AWS par AWS PrivateLink le biais du AWS PrivateLink guide.

Table des matières

Création d'un point de terminaison d'un VPC d'interface

Création d'un point de terminaison pour Application Auto Scaling à l'aide du nom de service suivant :

```
com.amazonaws.region.application-autoscaling
```

Pour plus d'informations, consultez la section Accès à un AWS service à l'aide d'un point de terminaison VPC d'interface dans le AWS PrivateLink Guide.

Vous n'avez pas besoin de modifier les autres paramètres. Application Auto Scaling appelle d'autres AWS services en utilisant soit des points de terminaison de service, soit des points de terminaison VPC d'interface privée, selon ceux utilisés.

Création d'une stratégie de point de terminaison de VPC

Vous pouvez attacher une stratégie à votre point de terminaison de VPC pour contrôler l'accès à l'API Application Auto Scaling. La stratégie précise :

- Le principal qui peut exécuter des actions.
- Les actions qui peuvent être effectuées.
- La ressource sur laquelle les actions peuvent être effectuées.

L'exemple suivant montre une politique de point de terminaison de VPC qui refuse à tout le monde l'autorisation de supprimer une politique de dimensionnement via le point de terminaison. L'exemple de politique accorde également à tout le monde l'autorisation d'effectuer toutes les autres actions.

```
{
   "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        },
        {
            "Action": "application-autoscaling:DeleteScalingPolicy",
            "Effect": "Deny",
            "Resource": "*",
            "Principal": "*"
        }
    ]
}
```

Pour de plus amples informations, veuillez consulter la rubrique Politiques de point de terminaison d'un VPC dans le Guide AWS PrivateLink .

Résilience dans Application Auto Scaling

L'infrastructure AWS mondiale est construite autour des AWS régions et des zones de disponibilité.

AWS Les régions fournissent plusieurs zones de disponibilité physiquement séparées et isolées, connectées par un réseau à faible latence, à haut débit et hautement redondant.

Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone à l'autre sans interruption. Les zones de disponibilité sont davantage disponibles, tolérantes aux pannes et ont une plus grande capacité de mise à l'échelle que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour plus d'informations sur AWS les régions et les zones de disponibilité, consultez la section Infrastructure AWS globale.

Sécurité de l'infrastructure dans Application Auto Scaling

En tant que service géré, Application Auto Scaling est protégé par la sécurité du réseau AWS mondial. Pour plus d'informations sur les services AWS de sécurité et sur la manière dont AWS l'infrastructure est protégée, consultez la section Sécurité du AWS cloud. Pour concevoir votre AWS environnement en utilisant les meilleures pratiques en matière de sécurité de l'infrastructure, consultez la section Protection de l'infrastructure dans le cadre AWS bien architecturé du pilier de sécurité.

Vous utilisez des appels d'API AWS publiés pour accéder à Application Auto Scaling via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Protocole TLS (Transport Layer Security). Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Ses suites de chiffrement PFS (Perfect Forward Secrecy) comme DHE (Ephemeral Diffie-Hellman) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

Validation de la conformité pour Application Auto Scaling

Pour savoir si un programme Services AWS de conformité Service AWS s'inscrit dans le champ d'application de programmes de conformité spécifiques, consultez Services AWS la section de conformité et sélectionnez le programme de conformité qui vous intéresse. Pour des informations générales, voir Programmes de AWS conformité Programmes AWS de .

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, voir Téléchargement de rapports dans AWS Artifact .

Votre responsabilité en matière de conformité lors de l'utilisation Services AWS est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise et les lois et réglementations applicables. Pour plus d'informations sur votre responsabilité en matière de conformité lors de l'utilisation Services AWS, consultez AWS la documentation de sécurité.

Quotas pour Application Auto Scaling

Vous Compte AWS disposez de quotas par défaut, anciennement appelés limites, pour chacun d'entre eux Service AWS. Sauf indication contraire, chaque quota est spécifique à la région. Vous pouvez demander des augmentations pour certains quotas, et d'autres quotas ne peuvent pas être augmentés.

Pour afficher les quotas pour Application Auto Scaling, ouvrez la console Service Quotas. Dans le panneau de navigation, choisissez AWS services (services) et sélectionnez Application Auto Scaling.

Pour demander une augmentation de quota, consultez Demander une augmentation de quota dans le Guide de l'utilisateur de Service Quotas.

Vous Compte AWS disposez des quotas suivants liés à Application Auto Scaling.

Nom	Par défaut	Ajustable
Objectifs évolutifs par type de ressource	Amazon DynamoDB : 5 000 | Amazon ECS : 3 000 | Amazon Keyspaces : 1 500 | Autres types de ressources : 500	Oui
Politiques de dimensionnement par cible évolutive (politiques de dimensionnement par étapes et de suivi des cibles)	50	Non
Actions planifiées par cible évolutive	200	Non
Ajustements par étape par politique de mise à l'échelle d'étape	20	Oui

Gardez à l'esprit les quotas de service lorsque vous mettez à l'échelle vos charges de travail. Par exemple, lorsque vous atteignez le nombre maximal d'unités de capacité autorisé par un service, la montée en puissance cesse. Si la demande chute et que la capacité actuelle diminue, Application Auto Scaling peut réaliser à nouveau une montée en puissance. Pour éviter d'atteindre à nouveau cette limite de capacité, vous pouvez demander une augmentation. Chaque service a ses propres

quotas par défaut pour la capacité maximale de la ressource. Pour plus d'informations sur les quotas par défaut pour les autres services Amazon Web Services, consultez Service endpoints and quotas (Points de terminaison et quotas de service) dans le Référence générale d'Amazon Web Services.

Historique du document pour Application Auto Scaling

Le tableau ci-dessous décrit les ajouts majeurs à la documentation Application Auto Scaling depuis janvier 2018. Pour recevoir les notifications sur les mises à jour de cette documentation, vous pouvez vous abonner au Flux RSS.

Modification	Description	Date
Ajout de la prise en charge des ElastiCache clusters Memcached	Utilisez Application Auto Scaling pour redimensionner horizontalement le nombre de nœuds d'un cluster Memcached. Pour plus d'informations, consultez ElastiCache Application Auto Scaling.	10 avril 2025
AWS mises à jour des politiques gérées	Application Auto Scaling a mis à jour la `AWSApplicationAutoscalingElastiCacheRGPolicy` politique.	10 avril 2025
Modifications du guide	Une nouvelle rubrique du Guide de l'utilisateur d'Application Auto Scaling vous aide à commencer à utiliser le dimensionnement prédictif avec Application Auto Scaling. Voir Application Auto Scaling Predictive Scaling.	21 novembre 2024
AWS mises à jour des politiques gérées	Application Auto Scaling a mis à jour la `AWSApplicationAutoscalingECSServicePolicy` politique.	21 novembre 2024

Ajoutez la prise en charge d'un pool de WorkSpaces	Utilisez Application Auto Scaling pour redimensionner un pool de WorkSpaces. Pour plus d'informations, consultez Amazon WorkSpaces et Application Auto Scaling. La rubrique Application Auto Scaling met à jour les politiques AWS gérées a été mise à jour pour répertorier une nouvelle politique gérée avec laquelle l'intégration doit être effectuée WorkSpaces.	27 juin 2024
Modifications du guide	Mise à jour d'entrée de Nombre maximal de cibles évolutives par type de ressource dans la documentation sur les quotas. Consultez Quotas pour Application Auto Scaling.	16 janvier 2024
Support pour les composants d'inférence de l' SageMaker IA	Utilisez Application Auto Scaling pour mettre à l'échelle les copies d'un composant d'inférence.	29 novembre 2023
AWS mises à jour des politiques gérées	Application Auto Scaling a mis à jour la `AWSApplicationAutoscalingSageMakerEndpointPolicy` politique.	13 novembre 2023

Support de la SageMaker simultanéité provisionnée sans serveur AI	Utilisez Application Auto Scaling pour mettre à l'échelle la concurrence provisionnée d'un point de terminaison sans serveur.	9 mai 2023
Classez vos cibles évolutives à l'aide de balises	Vous pouvez maintenant attribuer des métadonnées à vos cibles évolutives d'Application Auto Scaling sous forme de balises. Consultez Prise en charge du balisage pour Application Auto Scaling.	20 mars 2023
Support pour les mathématiques CloudWatch métriques	Vous pouvez désormais utiliser une expression mathématique appliquée à une métrique lorsque vous créez des politiques de dimensionnement de suivi des cibles. Avec les mathématiques métriques, vous pouvez interroger plusieurs CloudWatch métriques et utiliser des expressions mathématiques pour créer de nouvelles séries chronologiques basées sur ces métriques. Consultez Créer une stratégie de mise à l'échelle du suivi des cibles pour Application Auto Scaling à l'aide d'une expression mathématique appliquée à une métrique.	14 mars 2023

Raisons de ne pas dimensionner	Il est désormais possible de récupérer les raisons lisibles par machine pour lesquelles Application Auto Scaling ne dimensionne pas vos ressources à l'aide de l'API Application Auto Scaling. Reportez-vous à la section Activités de dimensionnement pour Application Auto Scaling.	4 janvier 2023
Modifications du guide	Mise à jour d'entrée de Nombre maximal de cibles évolutives par type de ressource dans la documentation sur les quotas. Consultez Quotas pour Application Auto Scaling.	6 mai 2022
Ajout de prise en charge des clusters Amazon Neptune	Utilisez Application Auto Scaling pour dimensionner le nombre de réplicas dans un cluster de bases de données Amazon Neptune DB. Pour de plus amples informations, veuillez consulterAmazon Neptune et Application Auto Scaling. La rubrique Mises à jour 'Application Auto Scaling au profit des stratégies gérées par AWS a été mis à jour pour répertorier une nouvelle stratégie gérée pour l'intégration à Neptune.	6 octobre 2021

Application Auto Scaling signale désormais les modifications apportées à ses politiques AWS gérées	À compter du 19 août 2021, les modifications apportées aux politiques gérées sont signalées dans la rubrique Application Auto Scaling : mises à jour des politiques AWS gérées. La première modification répertoriée est l'ajout des autorisations nécessaires pour ElastiCache (Redis OSS).	19 août 2021
Ajout de la prise en charge des ElastiCache groupes de réplication (Redis OSS)	Utilisez Application Auto Scaling pour dimensionner le nombre de groupes de nœuds et le nombre de répliques par groupe de nœuds pour un groupe de réplication ElastiCache (cluster) (Redis OSS). Pour plus d'informations, consultez ElastiCache (Redis OSS) et Application Auto Scaling.	19 août 2021

Modifications du guide	De nouvelles rubriques IAM dans le Guide de l'utilisateur Application Auto Scaling vous aident à résoudre les problèmes d'accès à Application Auto Scaling. Pour plus d'informations, consultez Identity and Access Management pour Application Auto Scaling. De nouveaux exemples de politiques d'autorisation IAM ont également été ajoutés pour les actions sur les services cibles et Amazon CloudWatch. Pour plus d'informations, voir Exemples de politiques pour l'utilisation du SDK AWS CLI ou d'un SDK.	23 février 2021
Ajout de la prise en charge pour les fuseaux horaires locaux	Vous pouvez désormais créer des actions programmées dans le fuseau horaire local. Si votre fuseau horaire observe l'heure d'été, il s'adapte automatiquement à l'heure d'été (DST). Pour plus d'informations, consultez Mise à l'échelle programmée.	2 février 2021

Modifications du guide	Un nouveau tutoriel dans le Guide de l'utilisateur Application Auto Scaling vous aide à comprendre comment utiliser les politiques de suivi des cibles et d'échelonnement et la mise à l'échelle programmée pour augmenter la disponibilité de votre application lors de l'utilisation d'Application Auto Scaling.	15 octobre 2020
Ajout de la prise en charge du stockage de cluster Amazon Managed Streaming pour Apache Kafka	Utilisez une politique de suivi des cibles et d'échelonnement pour augmenter la quantité de stockage d'agent associée à un cluster Amazon MSK.	30 septembre 2020
Ajout de la prise en charge des points de terminaison de reconnaissance d'entités Amazon Comprehend	Utilisez Application Auto Scaling pour mettre à l'échelle le nombre d'unités d'inférence allouées à vos points de terminaison de reconnaissance d'entités Amazon Comprehend.	28 septembre 2020
Ajout de la prise en charge des tables Amazon Keyspaces (for Apache Cassandra)	Utilisez l'Application Auto Scaling pour mettre à l'échelle le débit alloué (capacité de lecture et d'écriture) d'une table Amazon Keyspaces.	23 avril 2020

Nouveau chapitre « Sécurité »	Un nouveau chapitre Sécurité dans le Guide de l'utilisateur Application Auto Scaling vous aide à comprendre comment appliquer le modèle de responsabilité partagée lorsque vous utilisez Application Auto Scaling. Dans le cadre de cette mise à jour, le chapitre « Authentification et contrôle d'accès » du guide de l'utilisateur a été remplacé par une nouvelle section plus utile, Gestion des identités et des accès pour Application Auto Scaling.	16 janvier 2020
Mises à jour mineures	Diverses améliorations et corrections.	15 janvier 2020
Ajout d'une fonctionnalité de notification	Application Auto Scaling envoie désormais des événements à Amazon EventBridge et vous envoie des notifications AWS Health Dashboard lorsque certaines actions se produisent. Pour plus d'informations, consultez Surveillance d'Application Auto Scaling.	20 décembre 2019
Ajouter le support pour les AWS Lambda fonctions	Utilisez Application Auto Scaling pour mettre à l'échelle la concurrence allouée d'une fonction Lambda.	3 décembre 2019

Ajout de la prise en charge des points de terminaison de classification des documents Amazon Comprehend	Utilisez Application Auto Scaling pour mettre à l'échelle la capacité de débit d'un point de terminaison de classific ation de document Amazon Comprehend.	25 novembre 2019
Ajout de la prise en charge des WorkSpaces applicati ons pour les politiques de dimensionnement du suivi des cibles	Utilisez les politiques de dimensionnement du suivi des cibles pour adapter la taille d'un parc d' WorkSpaces applications.	25 novembre 2019
Prise en charge des points de terminaison Amazon VPC	Vous pouvez maintenant établir une connexion privée entre votre VPC et Applicati on Auto Scaling. Pour des considérations et des instructi ons sur la migration, consultez Application Auto Scaling et points de terminaison d'un VPC d'interface.	22 novembre 2019
Suspension et reprise de mise à l'échelle	Ajout de la prise en charge de la suspension et de la reprise du dimensionnement. Pour plus d'informations, consultez Suspension et reprise de la mise à l'échelle pour Applicati on Auto Scaling.	29 août 2019

Modifications du guide	Amélioration de la documentation sur Application Auto Scaling dans les sections Mise à l'échelle planifiée, Politiques de mise à l'échelle par étapes et Politiques de suivi des cibles et d'échelonnement.	11 mars 2019
Ajout de la prise en charge de ressources personnalisées	Utilisez Application Auto Scaling pour mettre à l'échelle les ressources personnalisées fournies par vos propres applications ou services. Pour plus d'informations, consultez notre GitHubréférentiel.	9 juillet 2018
Ajouter la prise en charge des variantes de terminaux SageMaker AI	Utilisez Application Auto Scaling pour mettre à l'échelle le nombre d'instances de points de terminaison allouées pour une variante.	28 février 2018

Le tableau suivant décrit les modifications importantes apportées à la documentation Application Auto Scaling avant janvier 2018.

Modifier	Description	Date
Ajout de prise en charge des réplicas Aurora	Utilisez Application Auto Scaling pour ajuster au nombre souhaité. Pour plus d'informations, consultez Utilisation d'Amazon Aurora Auto Scaling avec des réplicas Aurora dans le Guide de l'utilisateur Amazon RDS.	17 novembre 2017

Modifier	Description	Date
Ajout de prise en charge du dimensionnement planifié	Utilisez le dimensionnement planifié pour faire évoluer les ressources à des heures ou intervalles prédéfinis spécifiques. Pour plus d'informations, consultez Mise à l'échelle planifiée dans le Guide de l'utilisateur Application Auto Scaling.	8 novembre 2017
Ajout de prise en charge des stratégies de dimensionnement Suivi de la cible	Utilisez des stratégies de dimensionnement Suivi de la cible pour configurer le dimensionnement dynamique pour votre application en quelques étapes simples. Pour plus d'informations, consultez Politiques de suivi des cibles et d'échelonnement pour Application Auto Scaling.	12 juillet 2017
Ajout de la prise en charge de la capacité de lecture et d'écriture allouée aux tables et aux index secondaires globaux de DynamoDB	Utilisez Application Auto Scaling pour mettre à l'échelle le débit alloué (capacité de lecture et d'écriture). Pour plus d'informations, consultez la section Gestion de la capacité de débit avec DynamoDB Auto Scaling dans le Guide du développeur Amazon DynamoDB.	14 juin 2017

Modifier	Description	Date
Ajouter le support pour les flottes WorkSpaces d'applications	Utilisez Application Auto Scaling pour mettre à l'échelle la taille de la flotte. Pour plus d'informations, consultez Fleet Auto Scaling for WorkSpaces Applications dans le guide d'administration d'Amazon WorkSpaces Applications.	23 mars 2017
Ajout de prise en charge des clusters Amazon EMR	Utilisez Application Auto Scaling pour mettre à l'échelle les nœuds principaux et de tâches. Pour plus d'informations, consultez Utilisation de la scalabilité automatique dans Amazon EMR dans le Guide de gestion Amazon EMR.	le 18 novembre 2016
Ajout de la prise en charge des parcs Spot	Utilisez Application Auto Scaling pour mettre à l'échelle la capacité cible. Pour plus d'informations, consultez la section Dimensionnement automatique pour le parc Spot dans le guide de EC2 l'utilisateur Amazon.	1er septembre 2016
Ajout de prise en charge des services Amazon ECS	Utilisez Application Auto Scaling pour ajuster au nombre souhaité. Pour plus d'informations, consultez Scalabilité automatique de service dans le Guide du développeur Amazon Elastic Container Service.	9 août 2016

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.