aws

Guide du développeur

AWS Cloud Map

AWS Cloud Map: Guide du développeur

Table of Contents

Qu'est-ce que c'est AWS Cloud Map ?

AWS Cloud Map est une solution entièrement gérée que vous pouvez utiliser pour mapper des noms logiques aux services et ressources principaux dont dépendent vos applications. Il aide également vos applications à découvrir des ressources à l'aide de l' AWS SDKsun des appels d' RESTful API ou requêtes DNS. AWS Cloud Map ne fournit que des ressources saines, qui peuvent être des tables Amazon DynamoDB (DynamoDB), des files d'attente Amazon Simple Queue Service (Amazon SQS), des services applicatifs de niveau supérieur créés à l'aide d'instances Amazon Elastic Compute Cloud (Amazon) ou de tâches Amazon Elastic Container Service (EC2Amazon ECS), etc.

Composants de AWS Cloud Map

Espace de noms

Pour commencer, vous devez d'abord créer un espace de AWS Cloud Map noms qui fonctionne comme un moyen de regrouper les services d'une application. Un espace de noms identifie le nom que vous souhaitez utiliser pour localiser vos ressources et indique également comment vous souhaitez localiser les ressources : à l'aide d'appels d' AWS Cloud Map DiscoverInstancesAPI, de requêtes DNS dans un VPC ou de requêtes DNS publiques. Dans la plupart des cas, un espace de noms contient tous les services d'une application, telle qu'une application de facturation. Pour de plus amples informations, veuillez consulter AWS Cloud Map espaces de noms.

Service

Après avoir créé un espace de noms, vous créez un AWS Cloud Map service pour chaque type de ressource que vous souhaitez utiliser pour localiser les points AWS Cloud Map de terminaison. Par exemple, vous pouvez créer des services pour des serveurs web et des serveurs de base de données.

Un service est un modèle AWS Cloud Map utilisé lorsque votre application ajoute une autre ressource, telle qu'un autre serveur Web. Si vous avez choisi de localiser des ressources à l'aide de DNS lorsque vous avez créé l'espace de noms, un service contient des informations sur les types d'enregistrements que vous souhaitez utiliser pour rechercher le serveur web. Un service indique également si vous souhaitez vérifier l'état de santé de la ressource et si vous souhaitez utiliser les bilans de santé d'Amazon Route 53 ou un vérificateur de santé tiers. Pour de plus amples informations, veuillez consulter AWS Cloud Map services.

Instance de service

Lorsque votre application ajoute une ressource, vous pouvez appeler l'action AWS Cloud Map RegisterInstanceAPI dans le code, ce qui crée une instance de AWS Cloud Map service dans un service. L'instance de service contient des informations sur la manière dont votre application peut localiser la ressource, que ce soit à l'aide du DNS ou de l'action d' AWS Cloud Map DiscoverInstancesAPI.

Lorsque votre application doit se connecter à une ressource, elle appelle DiscoverInstancesou utilise des requêtes DNS publiques ou privées en spécifiant l'espace de noms et le service associés à la ressource. AWS Cloud Map renvoie des informations sur la manière de localiser une ou plusieurs ressources. Si vous avez spécifié la vérification de l'état lors de la création du service, AWS Cloud Map renvoie uniquement les instances saines. Pour de plus amples informations, veuillez consulter AWS Cloud Map instances de service.

Accès AWS Cloud Map

Vous pouvez y accéder AWS Cloud Map de différentes manières :

- AWS Management Console— Les procédures décrites dans ce guide expliquent comment utiliser le AWS Management Console pour effectuer des tâches.
- AWS SDKs— Si vous utilisez un langage de programmation qui AWS fournit un SDK pour, vous pouvez utiliser un SDK pour y accéder. AWS Cloud Map SDKs simplifiez l'authentification, intégrez facilement votre environnement de développement et donnez accès aux AWS Cloud Map commandes. Pour plus d’informations, consultez Outils pour Amazon Web Services.
- AWS Command Line Interface— Pour plus d'informations, voir Commencer avec le AWS CLI dans le guide de AWS Command Line Interface l'utilisateur.
- AWS Tools for Windows PowerShell— Pour plus d'informations, voir Commencer avec le AWS Tools for Windows PowerShell dans le guide de Outils AWS pour PowerShell l'utilisateur.
- AWS Cloud Map API — Si vous utilisez un langage de programmation pour lequel aucun SDK n'est disponible, consultez la référence des AWS Cloud Map API pour obtenir des informations sur les actions d'API et sur la manière de faire des demandes d'API.

> **Note**
>
> IPv6 Support client — Depuis le 22 juin 2023, dans toutes les nouvelles régions, toutes les commandes envoyées par les `IPv6` clients sont AWS Cloud Map routées vers un

nouveau point de terminaison dualstack (). `servicediscovery.<region>.api.aws` AWS Cloud Map `IPv6`-seuls les réseaux sont accessibles pour les terminaux legacy (**`servicediscovery.<region>.amazonaws.com`**) et dualstack dans les régions suivantes qui ont été publiées avant le 22 juin 2023 :

- USA Est (Ohio) – us-east-2
- USA Est (Virginie du Nord) – us-east-1
- USA Ouest (Californie du Nord) – us-west-1
- USA Ouest (Oregon) – us-west-2
- Afrique (Le Cap) – af-south-1
- Asie-Pacifique (Hong Kong) – ap-east-1
- Asie-Pacifique (Hyderabad) — ap-south-2
- Asie-Pacifique (Jakarta) : ap-southeast-3
- Asie-Pacifique (Melbourne) — ap-southeast-4
- Asie-Pacifique (Mumbai) – ap-south-1
- Asie-Pacifique (Osaka) – ap-northeast-3
- Asie-Pacifique (Séoul) – ap-northeast-2
- Asie-Pacifique (Singapour) – ap-southeast-1
- Asie-Pacifique (Sydney) – ap-southeast-2
- Asie-Pacifique (Tokyo) – ap-northeast-1
- Canada (Centre) – ca-central-1
- Europe (Francfort) – eu-central-1
- Europe (Irlande) – eu-west-1
- Europe (Londres) – eu-west-2
- Europe (Milan) – eu-south-1
- Europe (Paris) – eu-west-3
- Europe (Espagne) — eu-south-2
- Europe (Stockholm) – eu-north-1
- Europe (Zurich) — eu-central-2
- Moyen-Orient (Bahreïn) – me-south-1
- Moyen-Orient (Émirats arabes unis) — me-central-1
- Amérique du Sud (São Paulo) – sa-east-1

- AWS GovCloud (USA Est) — -1 us-gov-east
- AWS GovCloud (US-Ouest) — -1 us-gov-west

Gestion des identités et des accès AWS

AWS Cloud Map s'intègre à Gestion des identités et des accès AWS (IAM), un service que votre organisation peut utiliser pour effectuer les actions suivantes :

- Créez des utilisateurs et des groupes sous le AWS compte de votre organisation
- Partagez les ressources de votre AWS compte entre les utilisateurs du compte de manière efficace
- Attribuer des informations d'identification de sécurité uniques à chaque utilisateur
- Contrôler de façon détaillée l'accès utilisateur aux services et ressources

Par exemple, vous pouvez utiliser IAM AWS Cloud Map pour contrôler quels utilisateurs de votre AWS compte peuvent créer un nouvel espace de noms ou enregistrer des instances.

Pour obtenir des informations générales sur IAM, consultez les ressources suivantes :

- Identity and Access Management pour AWS Cloud Map
- Gestion des identités et des accès AWS
- Guide de l'utilisateur IAM

AWS Cloud Map Tarification

AWS Cloud Map la tarification est basée sur les ressources que vous enregistrez dans le registre des services et sur les appels d'API que vous effectuez pour les découvrir. Il n' AWS Cloud Map y a aucun paiement initial et vous ne payez que pour ce que vous utilisez.

Le cas échéant, vous pouvez activer la découverte basée sur DNS pour les ressources avec des adresses IP. Vous pouvez également activer la vérification de l'état de vos ressources à l'aide des vérifications d'état d'Amazon Route 53, que vous découvriez des instances à l'aide d'appels d'API ou de requêtes DNS. Vous devrez payer des frais supplémentaires liés au DNS Route 53 et à l'utilisation des bilans de santé.

Pour plus d'informations, consultez Tarification d'AWS Cloud Map.

AWS Cloud Map et conformité au AWS cloud

Pour plus d'informations sur AWS Cloud Map la conformité aux différentes réglementations de sécurité et normes d'audit, consultez les pages suivantes :

- AWS Conformité au cloud
- AWS Services visés par programme de conformité

Commencer avec AWS Cloud Map

Les guides suivants vous montrent comment configurer, utiliser AWS Cloud Map et exécuter des tâches courantes à l'aide d' AWS Cloud Map espaces de noms.

Vue d'ensemble du guide	En savoir plus
Inscription AWS et préparation à l'utilisation AWS Cloud Map	Configurer pour utiliser AWS Cloud Map
Utilisation de requêtes DNS et d'appels d'API pour découvrir les services principaux.	Découvrez comment utiliser la découverte AWS Cloud Map de services avec les requêtes DNS et les appels d'API
Utilisation de requêtes DNS et d'appels d'API pour découvrir les services principaux à l'aide du AWS CLI.	Découvrez comment utiliser la découverte de AWS Cloud Map services avec des requêtes DNS et des appels d'API à l'aide du AWS CLI
Création d'un exemple d'application et utilisation d'attributs personnalisés dans le code pour découvrir des ressources.	Découvrez comment utiliser la découverte AWS Cloud Map de services avec des attributs personnalisés
Création d'un exemple d'application et utilisation d'attributs personnalisés dans le code pour découvrir des ressources à l'aide du AWS CLI.	Apprenez à utiliser la découverte AWS Cloud Map de services avec des attributs personnalisés à l'aide du AWS CLI

Configurer pour utiliser AWS Cloud Map

La présentation et les procédures décrites dans les sections suivantes sont destinées à vous aider à démarrer AWS et à vous préparer à commencer à utiliser AWS Cloud Map.

Rubriques

- Inscrivez-vous pour AWS
- Accédez à l'API AWS CLI, AWS Tools for Windows PowerShell, ou au AWS SDKs
- Configurez le AWS Command Line Interface ou AWS Tools for Windows PowerShell
- Téléchargez un AWS SDK

Inscrivez-vous pour AWS

Inscrivez-vous pour un Compte AWS

Si vous n'en avez pas Compte AWS, procédez comme suit pour en créer un.

Pour vous inscrire à un Compte AWS

1. Ouvrez l'https://portal.aws.amazon.com/billing/inscription.
2. Suivez les instructions en ligne.

   Dans le cadre de la procédure d'inscription, vous recevrez un appel téléphonique ou un SMS et vous saisirez un code de vérification en utilisant le clavier numérique du téléphone.

   Lorsque vous vous inscrivez à un Compte AWS, un Utilisateur racine d'un compte AWSest créé. Par défaut, seul l'utilisateur racine a accès à l'ensemble des Services AWS et des ressources de ce compte. La meilleure pratique de sécurité consiste à attribuer un accès administratif à un utilisateur, et à utiliser uniquement l'utilisateur racine pour effectuer les tâches nécessitant un accès utilisateur racine.

AWS vous envoie un e-mail de confirmation une fois le processus d'inscription terminé. À tout moment, vous pouvez consulter l'activité actuelle de votre compte et gérer votre compte en accédant à https://aws.amazon.com/et en choisissant Mon compte.

Création d'un utilisateur doté d'un accès administratif

Après vous être inscrit à un Compte AWS, sécurisez Utilisateur racine d'un compte AWS AWS IAM Identity Center, activez et créez un utilisateur administratif afin de ne pas utiliser l'utilisateur root pour les tâches quotidiennes.

Sécurisez votre Utilisateur racine d'un compte AWS

1. Connectez-vous en AWS Management Consoletant que propriétaire du compte en choisissant Utilisateur root et en saisissant votre adresse Compte AWS e-mail. Sur la page suivante, saisissez votre mot de passe.

   Pour obtenir de l'aide pour vous connecter en utilisant l'utilisateur racine, consultez Connexion en tant qu'utilisateur racine dans le Guide de l'utilisateur Connexion à AWS .
2. Activez l'authentification multifactorielle (MFA) pour votre utilisateur racine.

Pour obtenir des instructions, voir Activer un périphérique MFA virtuel pour votre utilisateur Compte AWS root (console) dans le guide de l'utilisateur IAM.

Création d'un utilisateur doté d'un accès administratif

1. Activez IAM Identity Center.

   Pour obtenir des instructions, consultez Activation d' AWS IAM Identity Center dans le Guide de l'utilisateur AWS IAM Identity Center .
2. Dans IAM Identity Center, octroyez un accès administratif à un utilisateur.

   Pour un didacticiel sur l'utilisation du Répertoire IAM Identity Center comme source d'identité, voir Configurer l'accès utilisateur par défaut Répertoire IAM Identity Center dans le Guide de AWS IAM Identity Center l'utilisateur.

Connexion en tant qu'utilisateur doté d'un accès administratif

- Pour vous connecter avec votre utilisateur IAM Identity Center, utilisez l'URL de connexion qui a été envoyée à votre adresse e-mail lorsque vous avez créé l'utilisateur IAM Identity Center.

  Pour obtenir de l'aide pour vous connecter en utilisant un utilisateur d'IAM Identity Center, consultez la section Connexion au portail AWS d'accès dans le guide de l'Connexion à AWS utilisateur.

Attribution d'un accès à d'autres utilisateurs

1. Dans IAM Identity Center, créez un ensemble d'autorisations qui respecte la bonne pratique consistant à appliquer les autorisations de moindre privilège.

   Pour obtenir des instructions, consultez Création d'un ensemble d'autorisations dans le Guide de l'utilisateur AWS IAM Identity Center .
2. Attribuez des utilisateurs à un groupe, puis attribuez un accès par authentification unique au groupe.

   Pour obtenir des instructions, consultez Ajout de groupes dans le Guide de l'utilisateur AWS IAM Identity Center .

Accédez à l'API AWS CLI, AWS Tools for Windows PowerShell, ou au AWS SDKs

Pour utiliser l'API, le AWS CLI AWS Tools for Windows PowerShell, ou le AWS SDKs, vous devez créer des clés d'accès. Ces clés se composent d'un ID de clé d'accès et d'une clé d'accès secrète, qui permettent de signer les requêtes programmées auprès de AWS.

Les utilisateurs ont besoin d'un accès programmatique s'ils souhaitent interagir avec AWS l'extérieur du AWS Management Console. La manière d'accorder un accès programmatique dépend du type d'utilisateur qui y accède AWS.

Pour accorder aux utilisateurs un accès programmatique, choisissez l'une des options suivantes.

Quel utilisateur a besoin d'un accès programmatique ?	À	Méthode
IAM	(Recommandé) Utilisez les informations d'identification de la console comme informations d'identification temporaires pour signer les demandes programmatiques adressées au AWS CLI AWS SDKs, ou AWS APIs.	Suivez les instructions de l'interface que vous souhaitez utiliser. • Pour le AWS CLI, voir Connexion pour le développement AWS local dans le guide de AWS Command Line Interface l'utilisateur. • Pour AWS SDKs, voir Connexion pour le développement AWS local dans le guide de référence AWS SDKs and Tools.
Identité de la main-d'œuvre (Utilisateurs gérés dans IAM Identity Center)	Utilisez des informations d'identification temporaires pour signer les demandes programmatiques adressées au AWS CLI AWS SDKs, ou AWS APIs.	Suivez les instructions de l'interface que vous souhaitez utiliser. • Pour le AWS CLI, voir Configuration du AWS

Quel utilisateur a besoin d'un accès programmatique ?	À	Méthode
		CLI à utiliser AWS IAM Identity Center dans le guide de AWS Command Line Interface l'utilisateur. • Pour AWS SDKs, outils, et AWS APIs, voir Authentification IAM Identity Center dans le guide de référence AWS SDKs et Tools.
IAM	Utilisez des informations d'identification temporaires pour signer les demandes programmatiques adressées au AWS CLI AWS SDKs, ou AWS APIs.	Suivez les instructions de la section Utilisation d'informations d'identification temporaires avec AWS les ressources du Guide de l'utilisateur IAM.

Quel utilisateur a besoin d'un accès programmatique ?	À	Méthode
IAM	(Non recommandé) Utilisez des informations d'identification à long terme pour signer des demandes programmatiques adressées au AWS CLI AWS SDKs, ou AWS APIs.	Suivez les instructions de l'interface que vous souhaitez utiliser. • Pour le AWS CLI, voir Authentification à l'aide des informations d'identification utilisateur IAM dans le Guide de l'AWS Command Line Interface utilisateur. • Pour les outils AWS SDKs et, voir Authentifier à l'aide d'informations d'identif ication à long terme dans le guide de référence des outils AWS SDKs et. • Pour AWS APIs, voir Gestion des clés d'accès pour les utilisateurs IAM dans le Guide de l'utilisateur IAM.

Configurez le AWS Command Line Interface ou AWS Tools for Windows PowerShell

Le AWS Command Line Interface (AWS CLI) est un outil unifié de gestion des AWS services. Pour plus d'informations sur l'installation et la configuration du AWS CLI, voir Installation ou mise à jour vers la dernière version du AWS CLI dans le guide de AWS Command Line Interface l'utilisateur.

Si vous avez de l'expérience avec Windows PowerShell, vous préférerez peut-être utiliser AWS Tools for Windows PowerShell. Pour plus d'informations, consultez Configuration de AWS Tools for Windows PowerShell dans le Guide de l'utilisateur Outils AWS pour PowerShell .

Téléchargez un AWS SDK

Si vous utilisez un langage de programmation qui AWS fournit un SDK pour, nous vous recommandons d'utiliser un SDK au lieu de l' AWS Cloud Map API. L'utilisation d'un SDK présente plusieurs avantages. SDKs simplifiez l'authentification, intégrez facilement votre environnement de développement et donnez accès aux AWS Cloud Map commandes. Pour plus d'informations, consultez Outils pour Amazon Web Services.

Découvrez comment utiliser la découverte AWS Cloud Map de services avec les requêtes DNS et les appels d'API

Le didacticiel suivant simule une architecture de microservices avec deux services principaux. Le premier service sera détectable à l'aide d'une requête DNS. Le deuxième service sera détectable uniquement à l'aide de l' AWS Cloud Map API.

> **Note**
>
> Les détails des ressources, tels que les noms de domaine et les adresses IP, ne sont fournis qu'à des fins de simulation. Ils ne peuvent pas être résolus sur Internet.

Pour une end-to-end AWS CLI version de ce didacticiel, voirDécouvrez comment utiliser la découverte de AWS Cloud Map services avec des requêtes DNS et des appels d'API à l'aide du AWS CLI.

Conditions préalables

Les conditions préalables suivantes doivent être remplies pour terminer le didacticiel avec succès.

- Avant de commencer, complétez les étapes détaillées dans Configurer pour utiliser AWS Cloud Map.
- Si vous ne l'avez pas encore installé AWS Command Line Interface, suivez les étapes décrites dans la section Installation ou mise à jour de la dernière version du AWS CLI pour l'installer.

  Ce tutoriel nécessite un terminal de ligne de commande ou un shell pour exécuter les commandes. Sous Linux et macOS, utilisez votre gestionnaire de shell et de package préféré.

> **Note**
>
> Sous Windows, certaines commandes CLI Bash que vous utilisez couramment avec Lambda (par exemple `zip`) ne sont pas prises en charge par les terminaux intégrés du système d'exploitation. Installez le sous-système Windows pour Linux afin d'obtenir une version intégrée à Windows d'Ubuntu et Bash.

- Le didacticiel nécessite un environnement local avec la commande `dig` DNS lookup utility.

Étape 1 : créer un espace de AWS Cloud Map noms

Au cours de cette étape, vous allez créer un espace de AWS Cloud Map noms public. AWS Cloud Map crée une zone hébergée Route 53 en votre nom avec le même nom. Cela vous permet de découvrir les instances de service créées dans cet espace de noms à l'aide d'enregistrements DNS publics ou à l'aide d'appels d' AWS Cloud Map API.

1. Connectez-vous à la AWS Cloud Map console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudmap/.
2. Choisissez Create namespace (Créer un espace de noms).
3. Pour le nom de l'espace de noms, spécifiez`cloudmap-tutorial.com`.

   > **Note**
   >
   > Si vous deviez l'utiliser en production, vous devez vous assurer d'avoir spécifié le nom d'un domaine que vous possédiez ou auquel vous aviez accès. Mais pour les besoins de ce didacticiel, il n'est pas nécessaire qu'il s'agisse d'un domaine réel utilisé.

4. (Facultatif) Pour la description de l'espace de noms, spécifiez la raison pour laquelle vous souhaitez utiliser l'espace de noms.
5. Pour la découverte d'instances, sélectionnez les appels d'API et les requêtes DNS publiques.
6. Conservez le reste des valeurs par défaut et choisissez Create namespace.

Étape 2 : Création des AWS Cloud Map services

Au cours de cette étape, vous allez créer deux services. Le premier service sera détectable à l'aide d'appels DNS et API publics. Le second service sera détectable uniquement à l'aide d'appels d'API.

1. Connectez-vous à la AWS Cloud Map console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudmap/.
2. Dans le volet de navigation de gauche, choisissez Namespaces pour répertorier les espaces de noms que vous avez créés.
3. Dans la liste des espaces de noms, sélectionnez l'espace de `cloudmap-tutorial.com` noms et choisissez Afficher les détails.
4. Dans la section Services, choisissez Créer un service et procédez comme suit pour créer le premier service.

   a. Pour Nom du service, entrez `public-service`. Le nom du service sera appliqué aux enregistrements DNS AWS Cloud Map créés. Le format utilisé est*`<service-name>`*`.`*`<namespace-name>`*.

   b. Pour la configuration de Service Discovery, sélectionnez API et DNS.

   c. Dans la section Configuration DNS, pour Politique de routage, sélectionnez Routage de réponses à valeurs multiples.

   > **Note**
   >
   > La console le traduira en MULTIVALUE une fois sélectionné. Pour plus d'informations sur les options de routage disponibles, voir Choisir une politique de routage dans le Guide du développeur de Route 53.

   d. Conservez le reste des valeurs par défaut et choisissez Create service pour revenir à la page de détails de l'espace de noms.
5. Dans la section Services, choisissez Créer un service et procédez comme suit pour créer le second service.

   a. Pour Nom du service, entrez `backend-service`.

   b. Pour la configuration de Service Discovery, sélectionnez API uniquement.

   c. Conservez le reste des valeurs par défaut et choisissez Create service.

Étape 3 : enregistrer les instances AWS Cloud Map de service

Au cours de cette étape, vous créez deux instances de service, une pour chaque service de notre espace de noms.

1. Connectez-vous à la AWS Cloud Map console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudmap/.
2. Dans la liste des espaces de noms, sélectionnez l'espace de noms que vous avez créé à l'étape 1 et choisissez Afficher les détails.
3. Sur la page des détails de l'espace de noms, dans la liste des services, sélectionnez le `public-service` service et choisissez Afficher les détails.
4. Dans la section Instances de service, choisissez Enregistrer une instance de service et procédez comme suit pour créer la première instance de service.

    a. Pour l'ID de l'instance de service, spécifiez`first`.

    b. Pour IPv4 l'adresse, spécifiez`192.168.2.1`.

    c. Conservez le reste des valeurs par défaut et choisissez Enregistrer une instance de service.
5. À l'aide du fil d'Ariane situé en haut de la page, sélectionnez cloudmap-tutorial.com pour revenir à la page détaillée de l'espace de noms.
6. Sur la page des détails de l'espace de noms, dans la liste des services, sélectionnez le service principal et choisissez Afficher les détails.
7. Dans la section Instances de service, choisissez Enregistrer une instance de service et procédez comme suit pour créer la deuxième instance de service.

    a. Pour l'ID de l'instance de service, indiquez `second` qu'il s'agit de la deuxième instance de service.

    b. Dans Type d'instance, sélectionnez Informations d'identification pour une autre ressource.

    c. Pour les attributs personnalisés, ajoutez une paire clé-valeur avec `service-name` comme clé et `backend` comme valeur.

    d. Choisissez Register service instance (Enregistrer une instance de service).

Étape 4 : Découvrez les instances AWS Cloud Map de service

Maintenant que l'espace de AWS Cloud Map noms, les services et les instances de service sont créés, vous pouvez vérifier que tout fonctionne en découvrant les instances. Utilisez la `dig` commande pour vérifier les paramètres DNS publics et l' AWS Cloud Map API pour vérifier le service principal. Pour plus d'informations sur la `dig` commande, voir dig - Utilitaire de recherche DNS.

1. Connectez-vous à la console Route 53 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/route53/.

2. Dans le volet gauche de navigation, choisissez Hosted zones (Zones hébergées).
3. Sélectionnez la zone hébergée sur cloudmap-tutorial.com. Cela affiche les détails de la zone hébergée dans un volet séparé. Prenez note des serveurs de noms associés à votre zone hébergée, car nous les utiliserons à l'étape suivante.
4. À l'aide de la commande dig et de l'un des serveurs de noms Route 53 de votre zone hébergée, interrogez les enregistrements DNS de votre instance de service.

```
dig @hosted-zone-nameserver public-service.cloudmap-tutorial.com
```

La `ANSWER SECTION` sortie doit afficher l' IPv4 adresse que vous avez associée à votre `public-service` service.

```
;; ANSWER SECTION:
public-service.cloudmap-tutorial.com. 300 IN A 192.168.2.1
```

5. À l'aide de AWS CLI, recherchez les attributs de vos secondes instances de service.

```
aws servicediscovery discover-instances --namespace-name cloudmap-tutorial.com --
service-name backend-service --region region
```

La sortie affiche les attributs que vous avez associés au service sous forme de paires clé-valeur.

```
{
    "Instances": [
        {
            "InstanceId": "second",
            "NamespaceName": "cloudmap-tutorial.com",
            "ServiceName": "backend-service",
            "HealthStatus": "UNKNOWN",
            "Attributes": {
                "service-name": "backend"
            }
        }
    ],
    "InstancesRevision": 71462688285136850
}
```

Étape 5 : Nettoyer les ressources

Une fois le didacticiel terminé, vous pouvez supprimer les ressources. AWS Cloud Map nécessite que vous les nettoyiez dans l'ordre inverse, les instances de service d'abord, puis les services et enfin l'espace de noms. AWS Cloud Map nettoiera les ressources de la Route 53 en votre nom lorsque vous suivrez ces étapes.

1. Connectez-vous à la AWS Cloud Map console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudmap/.
2. Dans la liste des espaces de noms, sélectionnez l'espace de `cloudmap-tutorial.com` noms et choisissez Afficher les détails.
3. Sur la page des détails de l'espace de noms, dans la liste des services, sélectionnez le `public-service` service et choisissez Afficher les détails.
4. Dans la section Instances de service, sélectionnez l'`first`instance et choisissez Désenregistrer.
5. À l'aide du fil d'Ariane situé en haut de la page, sélectionnez cloudmap-tutorial.com pour revenir à la page détaillée de l'espace de noms.
6. Sur la page des détails de l'espace de noms, dans la liste des services, sélectionnez le service public et choisissez Supprimer.
7. Répétez les étapes 3 à 6 pour. `backend-service`
8. Dans le volet de navigation de gauche, choisissez Namespaces.
9. Sélectionnez l'espace de `cloudmap-tutorial.com` noms, puis choisissez Supprimer.

> **Note**
>
> Bien qu'il AWS Cloud Map nettoie les ressources Route 53 en votre nom, vous pouvez accéder à la console Route 53 pour vérifier que la zone `cloudmap-tutorial.com` hébergée est supprimée.

Découvrez comment utiliser la découverte de AWS Cloud Map services avec des requêtes DNS et des appels d'API à l'aide du AWS CLI

Ce didacticiel explique comment utiliser la découverte AWS Cloud Map de services à l'aide de la AWS Command Line Interface (CLI). Vous allez créer une architecture de microservices avec deux

services principaux : l'un détectable à l'aide de requêtes DNS et l'autre à l'aide de l'API uniquement. AWS Cloud Map

Pour un didacticiel qui inclut les étapes AWS Cloud Map relatives à la console, voirDécouvrez comment utiliser la découverte AWS Cloud Map de services avec les requêtes DNS et les appels d'API.

Conditions préalables

Les conditions préalables suivantes doivent être remplies pour terminer le didacticiel avec succès.

- Avant de commencer, complétez les étapes détaillées dans Configurer pour utiliser AWS Cloud Map.
- Si vous ne l'avez pas encore installé AWS Command Line Interface, suivez les étapes décrites dans la section Installation ou mise à jour de la dernière version du AWS CLI pour l'installer.

  Ce tutoriel nécessite un terminal de ligne de commande ou un shell pour exécuter les commandes. Sous Linux et macOS, utilisez votre gestionnaire de shell et de package préféré.

  > **Note**
  >
  > Sous Windows, certaines commandes CLI Bash que vous utilisez couramment avec Lambda (par exemple `zip`) ne sont pas prises en charge par les terminaux intégrés du système d'exploitation. Installez le sous-système Windows pour Linux afin d'obtenir une version intégrée à Windows d'Ubuntu et Bash.

- Le didacticiel nécessite un environnement local avec la commande `dig` DNS lookup utility.

Création d'un espace AWS Cloud Map de noms

Vous allez d'abord créer un espace de AWS Cloud Map noms public. AWS Cloud Map créera une zone hébergée Route 53 portant le même nom, permettant la découverte de services par le biais d'enregistrements DNS et d'appels d'API.

1. Créez l'espace de noms DNS public :

```
aws servicediscovery create-public-dns-namespace \
    --name cloudmap-tutorial.com \
    --creator-request-id cloudmap-tutorial-request-1 \
```

```
    --region us-east-2
```

La commande renvoie un ID d'opération que vous pouvez utiliser pour vérifier l'état de création de l'espace de noms :

```
{
    "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9xmplyzd"
}
```

2. Vérifiez l'état de l'opération pour confirmer que l'espace de noms a bien été créé :

```
aws servicediscovery get-operation \
    --operation-id gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9xmplyzd \
    --region us-east-2
```

3. Une fois l'opération réussie, obtenez l'ID de l'espace de noms :

```
aws servicediscovery list-namespaces \
    --region us-east-2 \
    --query "Namespaces[?Name=='cloudmap-tutorial.com'].Id" \
    --output text
```

Cette commande renvoie l'ID de l'espace de noms, dont vous aurez besoin pour les étapes suivantes :

```
ns-abcd1234xmplefgh
```

Créez les AWS Cloud Map services

Créez maintenant deux services dans votre espace de noms. Le premier service sera détectable à l'aide d'appels DNS et d'API, tandis que le second sera détectable uniquement à l'aide d'appels d'API.

1. Créez le premier service avec la découverte DNS activée :

```
aws servicediscovery create-service \
    --name public-service \
    --namespace-id ns-abcd1234xmplefgh \
    --dns-config "RoutingPolicy=MULTIVALUE,DnsRecords=[{Type=A,TTL=300}]" \
```

```
    --region us-east-2
```

La commande renvoie des informations sur le service créé :

```
{
    "Service": {
        "Id": "srv-abcd1234xmplefgh",
        "Arn": "arn:aws:servicediscovery:us-east-2:123456789012:service/srv-
abcd1234xmplefgh",
        "Name": "public-service",
        "NamespaceId": "ns-abcd1234xmplefgh",
        "DnsConfig": {
            "NamespaceId": "ns-abcd1234xmplefgh",
            "RoutingPolicy": "MULTIVALUE",
            "DnsRecords": [
                {
                    "Type": "A",
                    "TTL": 300
                }
            ]
        },
        "CreateDate": 1673613600.000,
        "CreatorRequestId": "public-service-request"
    }
}
```

2. Créez le deuxième service avec une fonction de découverte basée uniquement sur l'API :

```
aws servicediscovery create-service \
    --name backend-service \
    --namespace-id ns-abcd1234xmplefgh \
    --type HTTP \
    --region us-east-2
```

La commande renvoie des informations sur le service créé :

```
{
    "Service": {
        "Id": "srv-ijkl5678xmplmnop",
        "Arn": "arn:aws:servicediscovery:us-east-2:123456789012:service/srv-
ijkl5678xmplmnop",
        "Name": "backend-service",
```

```
        "NamespaceId": "ns-abcd1234xmplefgh",
        "Type": "HTTP",
        "CreateDate": 1673613600.000,
        "CreatorRequestId": "backend-service-request"
    }
}
```

Enregistrez les instances AWS Cloud Map de service

Enregistrez ensuite des instances de service pour chacun de vos services. Ces instances représentent les ressources réelles qui seront découvertes.

1. Enregistrez la première instance avec une IPv4 adresse pour la découverte du DNS :

   ```
   aws servicediscovery register-instance \
       --service-id srv-abcd1234xmplefgh \
       --instance-id first \
       --attributes AWS_INSTANCE_IPV4=192.168.2.1 \
       --region us-east-2
   ```

   La commande renvoie un identifiant d'opération :

   ```
   {
       "OperationId": "4yejorelbukcjzpnr6tlmrghsjwpngf4-k9xmplyzd"
   }
   ```

2. Vérifiez l'état du fonctionnement pour confirmer que l'instance a été correctement enregistrée :

   ```
   aws servicediscovery get-operation \
       --operation-id 4yejorelbukcjzpnr6tlmrghsjwpngf4-k9xmplyzd \
       --region us-east-2
   ```

3. Enregistrez la deuxième instance avec des attributs personnalisés pour la découverte des API :

   ```
   aws servicediscovery register-instance \
       --service-id srv-ijkl5678xmplmnop \
       --instance-id second \
       --attributes service-name=backend \
       --region us-east-2
   ```

La commande renvoie un identifiant d'opération :

```
{
    "OperationId": "7zxcvbnmasdfghjklqwertyuiop1234-k9xmplyzd"
}
```

4. Vérifiez l'état du fonctionnement pour confirmer que l'instance a été correctement enregistrée :

```
aws servicediscovery get-operation \
    --operation-id 7zxcvbnmasdfghjklqwertyuiop1234-k9xmplyzd \
    --region us-east-2
```

Découvrez les instances AWS Cloud Map de service

Maintenant que vous avez créé et enregistré vos instances de service, vous pouvez vérifier que tout fonctionne en les découvrant à l'aide de requêtes DNS et de l' AWS Cloud Map API.

1. Tout d'abord, obtenez l'ID de zone hébergée Route 53 :

```
aws route53 list-hosted-zones-by-name \
    --dns-name cloudmap-tutorial.com \
    --query "HostedZones[0].Id" \
    --output text
```

Cela renvoie l'ID de zone hébergée :

```
/hostedzone/Z1234ABCDXMPLEFGH
```

2. Obtenez les serveurs de noms de votre zone hébergée :

```
aws route53 get-hosted-zone \
    --id Z1234ABCDXMPLEFGH \
    --query "DelegationSet.NameServers[0]" \
    --output text
```

Cela renvoie l'un des serveurs de noms :

```
ns-1234.awsdns-12.org
```

3. Utilisez la `dig` commande pour interroger les enregistrements DNS de votre service public :

```
dig @ns-1234.awsdns-12.org public-service.cloudmap-tutorial.com
```

La sortie doit afficher l' IPv4 adresse que vous avez associée à votre service :

```
;; ANSWER SECTION:
public-service.cloudmap-tutorial.com. 300 IN A 192.168.2.1
```

4. Utilisez le AWS CLI pour découvrir l'instance du service principal :

```
aws servicediscovery discover-instances \
    --namespace-name cloudmap-tutorial.com \
    --service-name backend-service \
    --region us-east-2
```

La sortie affiche les attributs que vous avez associés au service :

```
{
    "Instances": [
        {
            "InstanceId": "second",
            "NamespaceName": "cloudmap-tutorial.com",
            "ServiceName": "backend-service",
            "HealthStatus": "UNKNOWN",
            "Attributes": {
                "service-name": "backend"
            }
        }
    ],
    "InstancesRevision": 71462688285136850
}
```

Nettoyez les ressources

Une fois le didacticiel terminé, nettoyez les ressources pour éviter d'encourir des frais. AWS Cloud Map nécessite que vous les nettoyiez dans l'ordre inverse : les instances de service d'abord, puis les services et enfin l'espace de noms.

1. Désenregistrez la première instance de service :

```
aws servicediscovery deregister-instance \
    --service-id srv-abcd1234xmplefgh \
    --instance-id first \
    --region us-east-2
```

2. Désenregistrez la deuxième instance de service :

```
aws servicediscovery deregister-instance \
    --service-id srv-ijkl5678xmplmnop \
    --instance-id second \
    --region us-east-2
```

3. Supprimer le service public :

```
aws servicediscovery delete-service \
    --id srv-abcd1234xmplefgh \
    --region us-east-2
```

4. Supprimez le service principal :

```
aws servicediscovery delete-service \
    --id srv-ijkl5678xmplmnop \
    --region us-east-2
```

5. Supprimez l'espace de noms de la fonction :

```
aws servicediscovery delete-namespace \
    --id ns-abcd1234xmplefgh \
    --region us-east-2
```

6. Vérifiez que la zone hébergée Route 53 est supprimée :

```
aws route53 list-hosted-zones-by-name \
    --dns-name cloudmap-tutorial.com
```

Découvrez comment utiliser la découverte AWS Cloud Map de services avec des attributs personnalisés

Le didacticiel suivant montre comment utiliser la découverte de AWS Cloud Map services avec des attributs personnalisés détectables à l'aide de l' AWS Cloud Map API. Le didacticiel vous explique comment créer et exécuter des applications clientes à l'aide de AWS CloudShell. Les applications utilisent deux fonctions Lambda pour écrire des données dans une table DynamoDB, puis les lire à partir de cette table. Les fonctions Lambda et la table DynamoDB sont enregistrées en tant qu'instances de service. AWS Cloud Map Le code des applications clientes et des fonctions Lambda utilise des attributs AWS Cloud Map personnalisés pour découvrir les ressources nécessaires à l'exécution du travail.

Pour une version AWS CLI basée de ce didacticiel, voirApprenez à utiliser la découverte AWS Cloud Map de services avec des attributs personnalisés à l'aide du AWS CLI.

> **Important**
>
> Vous créerez des AWS ressources au cours de l'atelier, ce qui entraînera des frais sur votre AWS compte. Il est recommandé de nettoyer les ressources dès la fin de l'atelier afin de minimiser les coûts.

Conditions préalables

Avant de commencer, complétez les étapes détaillées dans Configurer pour utiliser AWS Cloud Map.

Étape 1 : créer un espace de AWS Cloud Map noms

Au cours de cette étape, vous allez créer un espace de AWS Cloud Map noms. Un espace de noms est une construction utilisée pour regrouper les services d'une application. Lorsque vous créez l'espace de noms, vous spécifiez la manière dont les ressources seront détectables. Les ressources créées dans l'espace de noms créé au cours de cette étape seront détectables à l'aide d'appels d' AWS Cloud Map API utilisant des attributs personnalisés.

1. Connectez-vous à la AWS Cloud Map console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudmap/.
2. Choisissez Create namespace (Créer un espace de noms).

3. Pour le nom de l'espace de noms, spécifiez`cloudmap-tutorial`.
4. (Facultatif) Pour la description de l'espace de noms, spécifiez la raison pour laquelle vous souhaitez utiliser l'espace de noms.
5. Pour la découverte d'instances, sélectionnez Appels d'API.
6. Conservez le reste des valeurs par défaut et choisissez Create namespace.

Étape 2 : Création d'une table DynamoDB

Au cours de cette étape, vous allez créer une table DynamoDB. La table est utilisée pour stocker et récupérer les données de l'exemple d'application que vous allez créer dans les étapes suivantes.

Pour plus d'informations sur la création d'une DynamoDB, reportez-vous à Étape 1 : Création d'une table dans DynamoDB dans le Guide du développeur DynamoDB et utilisez le tableau suivant pour déterminer les options à spécifier.

Option	Value
Nom de la table	carte des nuages
Clé de partition	id

Conservez les valeurs par défaut pour le reste des paramètres et créez le tableau.

Étape 3 : Création d'un service de AWS Cloud Map données et enregistrement de la table DynamoDB en tant qu'instance

Au cours de cette étape, vous créez un AWS Cloud Map service, puis vous enregistrez la table DynamoDB créée lors de la dernière étape en tant qu'instance de service.

1. Ouvrez la AWS Cloud Map console à https://console.aws.amazon.com/cloudmap/
2. Dans la liste des espaces de noms, sélectionnez l'espace de `cloudmap-tutorial` noms et choisissez Afficher les détails.
3. Dans la section Services, choisissez Créer un service et procédez comme suit.
    a. Pour Nom du service, entrez `data-service`.

   b. Conservez le reste des valeurs par défaut et choisissez Create service.
4. Dans la section Services, sélectionnez le `data-service` service et choisissez Afficher les détails.
5. Dans la section Instances de service, choisissez Enregistrer une instance de service.
6. Sur la page Enregistrer une instance de service, procédez comme suit.

   a. Dans Type d'instance, sélectionnez Informations d'identification pour une autre ressource.

   b. Pour l'identifiant de l'instance de service, spécifiez`data-instance`.

   c. Dans la section Attributs personnalisés, spécifiez la paire clé-valeur suivante : clé =`tablename`, valeur =. `cloudmap`

Étape 4 : Création d'un rôle AWS Lambda d'exécution

Au cours de cette étape, vous créez un rôle IAM que la AWS Lambda fonction de l'étape suivante utilisera. Vous pouvez nommer le rôle IAM `cloudmap-tutorial-role` et omettre la limite d'autorisations, car le rôle n'est utilisé que pour ce didacticiel, et vous pouvez le supprimer par la suite.

Pour créer le rôle de service pour Lambda (console IAM)

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à https://console.aws.amazon.com/iam/l'adresse.
2. Dans le volet de navigation de la console IAM, sélectionnez Roles (Rôles), puis Create role (Créer un rôle).
3. Pour Trusted entity (Entité de confiance), choisissez Service AWS.
4. Pour Service ou cas d'utilisation, choisissez Lambda, puis choisissez le cas d'utilisation Lambda.
5. Choisissez Suivant.
6. Recherchez et cochez la case à côté de la `PowerUserAccess` politique, puis choisissez Suivant.
7. Choisissez Suivant.
8. Pour Nom du rôle, spécifiez`cloudmap-tutorial-role`.
9. Passez en revue les informations du rôle, puis choisissez Create role (Créer un rôle).

Étape 5 : Création de la fonction Lambda pour écrire des données

Au cours de cette étape, vous créez une fonction Lambda créée de toutes pièces qui écrit des données dans la table DynamoDB en utilisant l' AWS Cloud Map API pour interroger le service que vous avez créé. AWS Cloud Map

Pour plus d'informations sur la création d'une fonction Lambda, voir Création d'une fonction Lambda avec la console dans le Guide du AWS Lambda développeur et utilisez le tableau suivant pour déterminer les options à spécifier ou à choisir.

Option	Value
Nom de la fonction	fonction d'écriture
Environnement d’exécution	Python 3.12
Architecture	x86_64
Permissions	Utiliser un rôle existant
Rôle existant	cloudmap-tutorial-role

Après avoir créé la fonction, mettez à jour l'exemple de code pour qu'il reflète le code Python suivant, puis déployez la fonction. Notez que vous spécifiez l'attribut `datatable` personnalisé que vous avez associé à l'instance de AWS Cloud Map service que vous avez créée pour la table DynamoDB. La fonction génère une clé qui est un nombre aléatoire compris entre 1 et 100 et l'associe à une valeur qui est transmise à la fonction lors de son appel.

```
import json
import boto3
import random

def lambda_handler(event, context):

    serviceclient = boto3.client('servicediscovery')

    response = serviceclient.discover_instances(
        NamespaceName='cloudmap-tutorial',
        ServiceName='data-service')
```

```
    tablename = response["Instances"][0]["Attributes"]["tablename"]

    dynamodbclient = boto3.resource('dynamodb')

    table = dynamodbclient.Table(tablename)

    response = table.put_item(
        Item={ 'id': str(random.randint(1,100)), 'todo': event })

    return {
        'statusCode': 200,
        'body': json.dumps(response)
    }
```

Après avoir déployé la fonction, pour éviter les erreurs de temporisation, actualisez le délai d'expiration de la fonction à 5 secondes. Pour plus d'informations, voir Configurer le délai d'expiration de la fonction Lambda dans le Guide du AWS Lambda développeur.

Étape 6 : créer un service d' AWS Cloud Map application et enregistrer la fonction d'écriture Lambda en tant qu'instance

Au cours de cette étape, vous créez un AWS Cloud Map service, puis vous enregistrez la fonction d'écriture Lambda en tant qu'instance de service.

1. Ouvrez la AWS Cloud Map console à https://console.aws.amazon.com/cloudmap/
2. Dans le volet de navigation de gauche, choisissez Namespaces.
3. Dans la liste des espaces de noms, sélectionnez l'espace de `cloudmap-tutorial` noms et choisissez Afficher les détails.
4. Dans la section Services, choisissez Créer un service et procédez comme suit.
    a. Pour Nom du service, entrez `app-service`.
    b. Conservez le reste des valeurs par défaut et choisissez Create service.
5. Dans la section Services, sélectionnez le `app-service` service et choisissez Afficher les détails.
6. Dans la section Instances de service, choisissez Enregistrer une instance de service.
7. Sur la page Enregistrer une instance de service, procédez comme suit.
    a. Dans Type d'instance, sélectionnez Informations d'identification pour une autre ressource.

b. Pour l'identifiant de l'instance de service, spécifiez`write-instance`.

c. Dans la section Attributs personnalisés, spécifiez les paires clé-valeur suivantes.

- clé =`action`, valeur = `write`
- clé =`functionname`, valeur = `writefunction`

Étape 7 : Création de la fonction Lambda pour lire les données

Au cours de cette étape, vous allez créer une fonction Lambda créée de toutes pièces qui écrit des données dans la table DynamoDB que vous avez créée.

Pour plus d'informations sur la création d'une fonction Lambda, voir Création d'une fonction Lambda avec la console dans le Guide du AWS Lambda développeur et utilisez le tableau suivant pour déterminer les options à spécifier ou à choisir.

Option	Value
Nom de la fonction	fonction de lecture
Environnement d'exécution	Python 3.12
Architecture	x86_64
Permissions	Utiliser un rôle existant
Rôle existant	cloudmap-tutorial-role

Après avoir créé la fonction, mettez à jour l'exemple de code pour qu'il reflète le code Python suivant, puis déployez la fonction. La fonction scanne le tableau et renvoie tous les éléments.

```
import json
import boto3

def lambda_handler(event, context):
    serviceclient = boto3.client('servicediscovery')

    response = serviceclient.discover_instances(NamespaceName='cloudmap-tutorial',
 ServiceName='data-service')
```

```
    tablename = response["Instances"][0]["Attributes"]["tablename"]

    dynamodbclient = boto3.resource('dynamodb')

    table = dynamodbclient.Table(tablename)

    response = table.scan(Select='ALL_ATTRIBUTES')

    return {
        'statusCode': 200,
        'body': json.dumps(response)
    }
```

Après avoir déployé la fonction, pour éviter les erreurs de temporisation, actualisez le délai d'expiration de la fonction à 5 secondes. Pour plus d'informations, voir Configurer le délai d'expiration de la fonction Lambda dans le Guide du AWS Lambda développeur.

Étape 8 : enregistrer la fonction de lecture Lambda en tant qu'instance AWS Cloud Map de service

Au cours de cette étape, vous enregistrez la fonction de lecture Lambda en tant qu'instance de service dans le `app-service` service que vous avez créé précédemment.

1. Ouvrez la AWS Cloud Map console à https://console.aws.amazon.com/cloudmap/
2. Dans le volet de navigation de gauche, choisissez Namespaces.
3. Dans la liste des espaces de noms, sélectionnez l'espace de `cloudmap-tutorial` noms et choisissez Afficher les détails.
4. Dans la section Services, sélectionnez le `app-service` service et choisissez Afficher les détails.
5. Dans la section Instances de service, choisissez Enregistrer une instance de service.
6. Sur la page Enregistrer une instance de service, procédez comme suit.

    a. Dans Type d'instance, sélectionnez Informations d'identification pour une autre ressource.

    b. Pour l'identifiant de l'instance de service, spécifiez`read-instance`.

    c. Dans la section Attributs personnalisés, spécifiez les paires clé-valeur suivantes.

    - clé =`action`, valeur = `read`
    - clé =`functionname`, valeur = `readfunction`

Étape 9 : créer et exécuter des clients de lecture et d'écriture sur AWS CloudShell

Vous pouvez créer et exécuter des applications clientes AWS CloudShell qui utilisent du code pour découvrir les services que vous avez configurés AWS Cloud Map et appeler ces services.

1. Ouvrez la AWS CloudShell console à https://console.aws.amazon.com/cloudshell/
2. Utilisez la commande suivante pour créer un fichier appelé`writefunction.py`.

   ```
   vim writeclient.py
   ```

3. Dans le `writeclient.py` fichier, passez en mode insertion en appuyant sur le `i` bouton. Ensuite, copiez et collez le code suivant. Ce code découvre la fonction Lambda qui permet d'écrire des données en recherchant l'attribut personnalisé `name=writeservice` dans le `app-service` service. Le nom de la fonction Lambda chargée d'écrire les données dans la table DynamoDB est renvoyé. Ensuite, la fonction Lambda est invoquée, en transmettant un exemple de charge utile qui est écrit dans la table sous forme de valeur.

   ```
   import boto3

   serviceclient = boto3.client('servicediscovery')

   response = serviceclient.discover_instances(NamespaceName='cloudmap-tutorial',
    ServiceName='app-service', QueryParameters={ 'action': 'write' })

   functionname = response["Instances"][0]["Attributes"]["functionname"]

   lambdaclient = boto3.client('lambda')

   resp = lambdaclient.invoke(FunctionName=functionname, Payload='"This is a test
    data"')

   print(resp["Payload"].read())
   ```

4. Appuyez sur la touche d'échappement`:wq`, tapez et appuyez sur la touche Entrée pour enregistrer le fichier et quitter.
5. Utilisez la commande suivante pour exécuter le code Python.

   ```
   python3 writeclient.py
   ```

Le résultat doit être une `200` réponse, similaire à ce qui suit.

```
b'{"statusCode": 200, "body": "{\\"ResponseMetadata\\": {\\"RequestId\\": \
\"Q0M038IT0BPBVBJK8OCKK6I6M7VV4KQNSO5AEMVJF66Q9ASUAAJG\\", \\"HTTPStatusCode\
\": 200, \\"HTTPHeaders\\": {\\"server\\": \\"Server\\", \\"date\\": \\"Wed, 06
 Mar 2024 22:46:09 GMT\\", \\"content-type\\": \\"application/x-amz-json-1.0\\",
 \\"content-length\\": \\"2\\", \\"connection\\": \\"keep-alive\\", \\"x-amzn-
requestid\\": \\"Q0M038IT0BPBVBJK8OCKK6I6M7VV4KQNSO5AEMVJF66Q9ASUAAJG\\", \\"x-amz-
crc32\\": \\"2745614147\\"}, \\"RetryAttempts\\": 0}}"}'
```

6. Pour vérifier que l'écriture a réussi à l'étape précédente, créez un client de lecture.

   a. Utilisez la commande suivante pour créer un fichier appelé`readfunction.py`.

   ```
   vim readclient.py
   ```

   b. Dans le `readclient.py` fichier, appuyez sur le `i` bouton pour passer en mode insertion. Ensuite, copiez et collez le code suivant. Ce code scanne le tableau et renvoie la valeur que vous y avez écrite à l'étape précédente.

   ```
   import boto3

   serviceclient = boto3.client('servicediscovery')

   response = serviceclient.discover_instances(NamespaceName='cloudmap-tutorial',
    ServiceName='app-service', QueryParameters={ 'action': 'read' })

   functionname = response["Instances"][0]["Attributes"]["functionname"]

   lambdaclient = boto3.client('lambda')

   resp = lambdaclient.invoke(FunctionName=functionname,
    InvocationType='RequestResponse')

   print(resp["Payload"].read())
   ```

   c. Appuyez sur la touche d'échappement`:wq`, tapez et appuyez sur la touche Entrée pour enregistrer le fichier et quitter.

   d. Utilisez la commande suivante pour exécuter le code Python.

```
python3 readclient.py
```

La sortie doit ressembler à ce qui suit, répertoriant la valeur écrite dans la table en exécutant `writefunction.py` et la clé aléatoire générée dans la fonction d'écriture Lambda.

```
  b'{"statusCode": 200, "body": "{\\"Items\\": [{\\"id\\": \\"45\
\", \\"todo\\": \\"This is a test data\\"}], \\"Count\\": 1, \
\"ScannedCount\\": 1, \\"ResponseMetadata\\": {\\"RequestId\\": \
\"9JF8J6SFQCKR6IDT5JG5NOM3CNVV4KQNSO5AEMVJF66Q9ASUAAJG\\", \\"HTTPStatusCode\
\": 200, \\"HTTPHeaders\\": {\\"server\\": \\"Server\\", \\"date\\": \\"Thu, 25
 Jul 2024 20:43:33 GMT\\", \\"content-type\\": \\"application/x-amz-json-1.0\
\", \\"content-length\\": \\"91\\", \\"connection\\": \\"keep-alive\\", \\"x-
amzn-requestid\\": \\"9JF8J6SFQCKR6IDT5JG5NOM3CNVV4KQNSO5AEMVJF66Q9ASUAAJG\\",
 \\"x-amz-crc32\\": \\"1163081893\\"}, \\"RetryAttempts\\": 0}}"}'
```

Étape 10 : Nettoyer les ressources

Une fois le didacticiel terminé, supprimez les ressources pour éviter d'encourir des frais supplémentaires. AWS Cloud Map nécessite que vous les nettoyiez dans l'ordre inverse, les instances de service d'abord, puis les services et enfin l'espace de noms. Les étapes suivantes vous guident dans le nettoyage des AWS Cloud Map ressources utilisées dans le didacticiel.

Pour supprimer les AWS Cloud Map ressources

1. Connectez-vous à la AWS Cloud Map console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudmap/.
2. Dans la liste des espaces de noms, sélectionnez l'espace de `cloudmap-tutorial` noms et choisissez Afficher les détails.
3. Sur la page des détails de l'espace de noms, dans la liste des services, sélectionnez le `data-service` service et choisissez Afficher les détails.
4. Dans la section Instances de service, sélectionnez l'`data-instance`instance et choisissez Désenregistrer.
5. À l'aide du fil d'Ariane situé en haut de la page, sélectionnez cloudmap-tutorial.com pour revenir à la page détaillée de l'espace de noms.
6. Sur la page des détails de l'espace de noms, dans la liste des services, sélectionnez le service de données et choisissez Supprimer.

7. Répétez les étapes 3 à 6 pour le `app-service` service `write-instance` et les instances `read-instance` de service.
8. Dans le volet de navigation de gauche, choisissez Namespaces.
9. Sélectionnez l'espace de `cloudmap-tutorial` noms, puis choisissez Supprimer.

Le tableau suivant répertorie les procédures que vous pouvez suivre pour supprimer les autres ressources utilisées dans le didacticiel.

Ressource	Étapes	
Tableau DynamoDB	Étape 6 : (Facultatif) Supprimez votre table DynamoDB pour nettoyer les ressources dans le manuel du développeur Amazon DynamoDB	
Fonctions Lambda et rôle d'exécution IAM associé	Faites le ménage dans le guide du AWS Lambda développeur	

Apprenez à utiliser la découverte AWS Cloud Map de services avec des attributs personnalisés à l'aide du AWS CLI

Ce didacticiel explique comment utiliser la découverte AWS Cloud Map de services avec des attributs personnalisés. Vous allez créer une application de microservices qui permet de découvrir les ressources de manière dynamique AWS Cloud Map à l'aide d'attributs personnalisés. L'application se compose de deux fonctions Lambda qui écrivent et lisent des données dans une table DynamoDB, toutes les ressources y étant enregistrées. AWS Cloud Map

Pour une AWS Management Console version du didacticiel, voirDécouvrez comment utiliser la découverte AWS Cloud Map de services avec des attributs personnalisés.

Conditions préalables

Avant de commencer ce didacticiel, suivez les étapes décrites dansConfigurer pour utiliser AWS Cloud Map.

Création d'un espace AWS Cloud Map de noms

Un espace de noms est une construction utilisée pour regrouper les services d'une application. Au cours de cette étape, vous allez créer un espace de noms qui permet de découvrir les ressources par le biais d'appels d' AWS Cloud Map API.

1. Exécutez la commande suivante pour créer un espace de noms HTTP :

   ```
   aws servicediscovery create-http-namespace \
     --name cloudmap-tutorial \
     --creator-request-id cloudmap-tutorial-request
   ```

   La commande renvoie un identifiant d'opération. Vous pouvez vérifier le statut de l'opération à l'aide de la commande suivante :

   ```
   aws servicediscovery get-operation \
     --operation-id operation-id
   ```

2. Une fois l'espace de noms créé, vous pouvez récupérer son identifiant pour l'utiliser dans les commandes suivantes :

   ```
   aws servicediscovery list-namespaces \
     --query "Namespaces[?Name=='cloudmap-tutorial'].Id" \
     --output text
   ```

3. Stockez l'ID de l'espace de noms dans une variable pour une utilisation ultérieure :

   ```
   NAMESPACE_ID=$(aws servicediscovery list-namespaces \
     --query "Namespaces[?Name=='cloudmap-tutorial'].Id" \
     --output text)
   ```

Créez une table DynamoDB

Créez ensuite une table DynamoDB qui stockera les données de votre application :

1. Exécutez la commande suivante pour créer la table :

```
aws dynamodb create-table \
  --table-name cloudmap \
  --attribute-definitions AttributeName=id,AttributeType=S \
  --key-schema AttributeName=id,KeyType=HASH \
  --billing-mode PAY_PER_REQUEST
```

2. Attendez que la table soit active avant de continuer :

```
aws dynamodb wait table-exists --table-name cloudmap
```

   Cette commande attend que la table soit complètement créée et prête à être utilisée.

Création d'un service de AWS Cloud Map données et enregistrement de la table DynamoDB

Créez maintenant un service dans votre espace de noms pour représenter les ressources de stockage de données :

1. Exécutez la commande suivante pour créer un AWS Cloud Map service pour les ressources de stockage de données :

```
aws servicediscovery create-service \
  --name data-service \
  --namespace-id $NAMESPACE_ID \
  --creator-request-id data-service-request
```

2. Obtenez l'identifiant du service de données :

```
DATA_SERVICE_ID=$(aws servicediscovery list-services \
  --query "Services[?Name=='data-service'].Id" \
  --output text)
```

3. Enregistrez la table DynamoDB en tant qu'instance de service avec un attribut personnalisé qui spécifie le nom de la table :

```
aws servicediscovery register-instance \
  --service-id $DATA_SERVICE_ID \
  --instance-id data-instance \
```

```
  --attributes tablename=cloudmap
```

L'attribut personnalisé `tablename=cloudmap` permet aux autres services de découvrir le nom de la table DynamoDB de manière dynamique.

Création d'un rôle IAM pour les fonctions Lambda

Créez un rôle IAM que les fonctions Lambda utiliseront pour accéder aux AWS ressources :

1. Créez le document de politique de confiance pour le rôle IAM à l'aide du JSON suivant :

   JSON

   ```
   {
     "Version":"2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
           "Service": "lambda.amazonaws.com"
         },
         "Action": "sts:AssumeRole"
       }
     ]
   }
   ```

2. Exécutez la commande suivante pour créer le rôle IAM à l'aide de la politique de confiance :

   ```
   aws iam create-role \
     --role-name cloudmap-tutorial-role \
     --assume-role-policy-document file://lambda-trust-policy.json
   ```

3. Créez un fichier pour une politique IAM personnalisée avec les autorisations de moindre privilège à l'aide du code JSON suivant :

   JSON

   ```
   {
     "Version":"2012-10-17",
     "Statement": [
   ```

```
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "servicediscovery:DiscoverInstances"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:PutItem",
        "dynamodb:Scan"
      ],
      "Resource": "arn:aws:dynamodb:*:*:table/cloudmap"
    }
  ]
}
```

4. Créez et associez la politique au rôle IAM :

```
aws iam create-policy \
  --policy-name CloudMapTutorialPolicy \
  --policy-document file://cloudmap-policy.json

POLICY_ARN=$(aws iam list-policies \
  --query "Policies[?PolicyName=='CloudMapTutorialPolicy'].Arn" \
  --output text)

aws iam attach-role-policy \
  --role-name cloudmap-tutorial-role \
  --policy-arn $POLICY_ARN

aws iam attach-role-policy \
  --role-name cloudmap-tutorial-role \
```

```
  --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole
```

Créez la fonction Lambda pour écrire des données

Pour créer une fonction Lambda qui écrit des données dans la table DynamoDB, procédez comme suit :

1. Créez le fichier Python pour la fonction d'écriture :

```
cat > writefunction.py << EOF
import json
import boto3
import random

def lambda_handler(event, context):
    try:
        serviceclient = boto3.client('servicediscovery')

        response = serviceclient.discover_instances(
            NamespaceName='cloudmap-tutorial',
            ServiceName='data-service')

        if not response.get("Instances"):
            return {
                'statusCode': 500,
                'body': json.dumps({"error": "No instances found"})
            }

        tablename = response["Instances"][0]["Attributes"].get("tablename")
        if not tablename:
            return {
                'statusCode': 500,
                'body': json.dumps({"error": "Table name attribute not found"})
            }

        dynamodbclient = boto3.resource('dynamodb')

        table = dynamodbclient.Table(tablename)

        # Validate input
        if not isinstance(event, str):
            return {
```

```
                'statusCode': 400,
                'body': json.dumps({"error": "Input must be a string"})
            }

        response = table.put_item(
            Item={ 'id': str(random.randint(1,100)), 'todo': event })

        return {
            'statusCode': 200,
            'body': json.dumps(response)
        }
    except Exception as e:
        return {
            'statusCode': 500,
            'body': json.dumps({"error": str(e)})
        }
EOF
```

Cette fonction permet AWS Cloud Map de découvrir le nom de la table DynamoDB à partir de l'attribut personnalisé, puis d'écrire des données dans la table.

2. Package et déploiement de la fonction Lambda :

```
zip writefunction.zip writefunction.py

ROLE_ARN=$(aws iam get-role --role-name cloudmap-tutorial-role \
  --query 'Role.Arn' --output text)

aws lambda create-function \
  --function-name writefunction \
  --runtime python3.12 \
  --role $ROLE_ARN \
  --handler writefunction.lambda_handler \
  --zip-file fileb://writefunction.zip \
  --architectures x86_64
```

3. Mettez à jour le délai d'expiration de la fonction pour éviter les erreurs de temporisation :

```
aws lambda update-function-configuration \
  --function-name writefunction \
  --timeout 5
```

Créez un service d' AWS Cloud Map application et enregistrez la fonction d'écriture Lambda

Pour créer un autre service dans votre espace de noms afin de représenter les fonctions de l'application, procédez comme suit :

1. Créez un service pour les fonctions de l'application :

```
aws servicediscovery create-service \
  --name app-service \
  --namespace-id $NAMESPACE_ID \
  --creator-request-id app-service-request
```

2. Obtenez l'ID de service pour le service d'application :

```
APP_SERVICE_ID=$(aws servicediscovery list-services \
  --query "Services[?Name=='app-service'].Id" \
  --output text)
```

3. Enregistrez la fonction d'écriture Lambda en tant qu'instance de service avec des attributs personnalisés :

```
aws servicediscovery register-instance \
  --service-id $APP_SERVICE_ID \
  --instance-id write-instance \
  --attributes action=write,functionname=writefunction
```

   Les attributs personnalisés `functionname=writefunction` permettent `action=write` aux clients de découvrir cette fonction en fonction de son objectif.

Créez la fonction Lambda pour lire les données

Pour créer une fonction Lambda qui lit les données de la table DynamoDB, procédez comme suit :

1. Créez le fichier Python pour la fonction de lecture :

```
cat > readfunction.py << EOF
import json
import boto3
```

```
def lambda_handler(event, context):
    try:
        serviceclient = boto3.client('servicediscovery')

        response = serviceclient.discover_instances(
            NamespaceName='cloudmap-tutorial',
            ServiceName='data-service')

        if not response.get("Instances"):
            return {
                'statusCode': 500,
                'body': json.dumps({"error": "No instances found"})
            }

        tablename = response["Instances"][0]["Attributes"].get("tablename")
        if not tablename:
            return {
                'statusCode': 500,
                'body': json.dumps({"error": "Table name attribute not found"})
            }

        dynamodbclient = boto3.resource('dynamodb')

        table = dynamodbclient.Table(tablename)

        # Use pagination for larger tables
        response = table.scan(
            Select='ALL_ATTRIBUTES',
            Limit=50  # Limit results for demonstration purposes
        )

        # For production, you would implement pagination like this:
        # items = []
        # while 'LastEvaluatedKey' in response:
        #     items.extend(response['Items'])
        #     response = table.scan(
        #         Select='ALL_ATTRIBUTES',
        #         ExclusiveStartKey=response['LastEvaluatedKey']
        #     )
        # items.extend(response['Items'])

        return {
            'statusCode': 200,
            'body': json.dumps(response)
```

```
        }
    except Exception as e:
        return {
            'statusCode': 500,
            'body': json.dumps({"error": str(e)})
        }
EOF
```

Cette fonction permet également AWS Cloud Map de découvrir le nom de la table DynamoDB, puis de lire les données de la table. Il inclut la gestion des erreurs et les commentaires de pagination.

2. Package et déploiement de la fonction Lambda :

```
zip readfunction.zip readfunction.py

aws lambda create-function \
  --function-name readfunction \
  --runtime python3.12 \
  --role $ROLE_ARN \
  --handler readfunction.lambda_handler \
  --zip-file fileb://readfunction.zip \
  --architectures x86_64
```

3. Mettez à jour le délai d'expiration de la fonction :

```
aws lambda update-function-configuration \
  --function-name readfunction \
  --timeout 5
```

Enregistrez la fonction de lecture Lambda en tant qu'instance de service

Pour enregistrer la fonction de lecture Lambda en tant qu'autre instance de service dans le service d'application, procédez comme suit :

```
aws servicediscovery register-instance \
  --service-id $APP_SERVICE_ID \
  --instance-id read-instance \
  --attributes action=read,functionname=readfunction
```

Les attributs personnalisés `functionname=readfunction` permettent `action=read` aux clients de découvrir cette fonction en fonction de son objectif.

Création et exécution d'applications clientes

Pour créer une application cliente Python qui AWS Cloud Map découvre et invoque la fonction d'écriture, procédez comme suit :

1. Créez un fichier Python pour l'application cliente d'écriture :

```
cat > writeclient.py << EOF
import boto3
import json

try:
    serviceclient = boto3.client('servicediscovery')

    print("Discovering write function...")
    response = serviceclient.discover_instances(
        NamespaceName='cloudmap-tutorial',
        ServiceName='app-service',
        QueryParameters={ 'action': 'write' }
    )

    if not response.get("Instances"):
        print("Error: No instances found")
        exit(1)

    functionname = response["Instances"][0]["Attributes"].get("functionname")
    if not functionname:
        print("Error: Function name attribute not found")
        exit(1)

    print(f"Found function: {functionname}")

    lambdaclient = boto3.client('lambda')

    print("Invoking Lambda function...")
    resp = lambdaclient.invoke(
        FunctionName=functionname,
        Payload='"This is a test data"'
    )
```

```
    payload = resp["Payload"].read()
    print(f"Response: {payload.decode('utf-8')}")

except Exception as e:
    print(f"Error: {str(e)}")
EOF
```

Ce client utilise l'`QueryParameters`option pour rechercher des instances de service avec l'`action=write`attribut.

2. Créez un fichier Python pour l'application cliente de lecture :

```
cat > readclient.py << EOF
import boto3
import json

try:
    serviceclient = boto3.client('servicediscovery')

    print("Discovering read function...")
    response = serviceclient.discover_instances(
        NamespaceName='cloudmap-tutorial',
        ServiceName='app-service',
        QueryParameters={ 'action': 'read' }
    )

    if not response.get("Instances"):
        print("Error: No instances found")
        exit(1)

    functionname = response["Instances"][0]["Attributes"].get("functionname")
    if not functionname:
        print("Error: Function name attribute not found")
        exit(1)

    print(f"Found function: {functionname}")

    lambdaclient = boto3.client('lambda')

    print("Invoking Lambda function...")
    resp = lambdaclient.invoke(
        FunctionName=functionname,
        InvocationType='RequestResponse'
```

```
    )

    payload = resp["Payload"].read()
    print(f"Response: {payload.decode('utf-8')}")

except Exception as e:
    print(f"Error: {str(e)}")
EOF
```

3. Exécutez le client d'écriture pour ajouter des données à la table DynamoDB :

```
python3 writeclient.py
```

La sortie doit afficher une réponse réussie avec le code d'état HTTP 200.

4. Exécutez le client de lecture pour récupérer les données de la table DynamoDB :

```
python3 readclient.py
```

La sortie doit afficher les données écrites dans la table, y compris l'identifiant généré de manière aléatoire et la valeur « Ceci est une donnée de test ».

nettoyer des ressources ;

Lorsque vous avez terminé le didacticiel, nettoyez les ressources pour éviter d'encourir des frais supplémentaires.

1. Exécutez d'abord la commande suivante pour désenregistrer les instances de service :

```
aws servicediscovery deregister-instance \
  --service-id $APP_SERVICE_ID \
  --instance-id read-instance

aws servicediscovery deregister-instance \
  --service-id $APP_SERVICE_ID \
  --instance-id write-instance

aws servicediscovery deregister-instance \
  --service-id $DATA_SERVICE_ID \
  --instance-id data-instance
```

2. Exécutez la commande suivante pour supprimer les services :

```
aws servicediscovery delete-service \
  --id $APP_SERVICE_ID

aws servicediscovery delete-service \
  --id $DATA_SERVICE_ID
```

3. Exécutez la commande suivante pour supprimer l'espace de noms :

```
aws servicediscovery delete-namespace \
  --id $NAMESPACE_ID
```

4. Exécutez la commande suivante pour supprimer les fonctions Lambda :

```
aws lambda delete-function --function-name writefunction
aws lambda delete-function --function-name readfunction
```

5. Exécutez la commande suivante pour supprimer le rôle et la politique IAM :

```
aws iam detach-role-policy \
  --role-name cloudmap-tutorial-role \
  --policy-arn $POLICY_ARN

aws iam detach-role-policy \
  --role-name cloudmap-tutorial-role \
  --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole

aws iam delete-policy \
  --policy-arn $POLICY_ARN

aws iam delete-role --role-name cloudmap-tutorial-role
```

6. Exécutez la commande suivante pour supprimer la table DynamoDB :

```
aws dynamodb delete-table --table-name cloudmap
```

7. Exécutez la commande suivante pour nettoyer les fichiers temporaires :

```
rm -f lambda-trust-policy.json cloudmap-policy.json writefunction.py
 readfunction.py writefunction.zip readfunction.zip writeclient.py readclient.py
```

AWS Cloud Map espaces de noms

Un espace de noms est une entité logique utilisée pour regrouper AWS Cloud Map les services d'une application sous un nom et un niveau de découvrabilité communs. Lorsque vous créez un espace de noms, vous spécifiez les éléments suivants :

- Nom que vous souhaitez que votre application utilise pour découvrir des instances.
- Méthode par laquelle les instances de service auprès desquelles vous vous inscrivez AWS Cloud Map peuvent être découvertes. Vous pouvez décider si vos ressources doivent être découvertes publiquement sur Internet, en privé dans un cloud privé virtuel (VPC) spécifique ou uniquement par des appels d'API.

Les concepts généraux relatifs aux espaces de noms sont présentés ci-dessous.

- Les espaces de noms sont spécifiques à l'endroit dans Région AWS lequel ils ont été créés. Pour l'utiliser AWS Cloud Map dans plusieurs régions, vous devez créer des espaces de noms dans chaque région.
- Si vous créez un espace de noms pour permettre, par exemple, la découverte d'instances par des requêtes DNS dans un VPC AWS Cloud Map , une zone hébergée Route 53 privée est automatiquement créée. Cette zone hébergée peut être associée à plusieurs VPCs. Pour plus d'informations, consultez Associate VPCWith HostedZone dans le manuel Amazon Route 53 API Reference.

Rubriques

Création d'un AWS Cloud Map espace de noms pour regrouper les services d'application

Vous pouvez créer un espace de noms pour regrouper les services de votre application sous un nom convivial qui permet de découvrir les ressources de l'application par le biais d'appels d'API ou de requêtes DNS.

Options de découverte d'instances

Le tableau suivant récapitule les différentes options de découverte d'instance AWS Cloud Map et le type d'espace de noms correspondant que vous pouvez créer, en fonction des services et de la configuration de votre application.

Type d'espace de noms	Méthode de découverte des instances	Comment ça marche	Informations supplémentaires
HTTP	Appels d’API	Les ressources de votre application peuvent découvrir d'autres ressources en appelant uniquement l'`DiscoverInstances` API.	• DiscoverInstances • CreateHttpNamespace
DNS privé	Appels d'API et requêtes DNS dans un VPC	Lorsque vous créez un espace de noms DNS privé, vous AWS Cloud Map créez une zone hébergée privée Amazon Route 53 correspondante. Les ressources de votre application peuvent découvrir d'autres ressources en appelant l'`DiscoverI`	• DiscoverInstances • CreatePrivateDnsNamespace

Type d'espace de noms	Méthode de découverte des instances	Comment ça marche	Informations supplémentaires
		`nstances` API et en interrogeant les serveurs de noms de la zone hébergée privée Route 53 qui AWS Cloud Map se crée automatiquement. La zone hébergée créée par AWS Cloud Map porte le même nom que l'espace de noms et contient des enregistrements DNS dont les noms sont au format *`service-name`* . *`namespace-name`*. **Note** Route 53 Resolver résout les requêtes DNS provenant du VPC à l'aide des enregistrements de la zone hébergée privée. Si la zone	

Type d'espace de noms	Méthode de découverte des instances	Comment ça marche	Informations supplémentaires
		hébergée privée n'inclut aucun enregistrement correspondant au nom de domaine d'une requête DNS, Route 53 répond à la requête par `NXDOMAIN` (domaine inexistant).	

Type d'espace de noms	Méthode de découverte des instances	Comment ça marche	Informations supplémentaires
DNS public	API calls and public DNS queries (Appels d'API et requêtes DNS publiques)	Lorsque vous créez un espace de noms DNS public, vous AWS Cloud Map créez une zone hébergée publique Amazon Route 53 correspondante. Les ressources de votre application peuvent découvrir d'autres ressources en appelant l'`DiscoverInstances` API et en interrogeant les serveurs de noms de la zone hébergée publique Route 53 qui AWS Cloud Map se crée automatiquement. La zone hébergée publique porte le même nom que l'espace de noms et contient des enregistrements DNS dont les noms sont au format*`service-name`*.*`namespace-name`*.	• DiscoverInstances • CreatePublicDnsNamespace

Type d'espace de noms	Méthode de découverte des instances	Comment ça marche	Informations supplémentaires
		Note Dans ce cas, le nom de l'espace de noms doit être un nom de domaine que vous avez enregistré.	

Procédure

Vous pouvez suivre ces étapes pour créer un espace de noms à l'aide du AWS CLI AWS Management Console, ou du SDK pour Python.

AWS Management Console

1. Connectez-vous à la AWS Cloud Map console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudmap/.
2. Choisissez Create namespace (Créer un espace de noms).
3. Dans Nom de l'espace de noms, entrez un nom qui sera utilisé pour découvrir les instances.

   Note

   - Les espaces de noms configurés pour les requêtes DNS publiques doivent se terminer par un domaine de premier niveau. Par exemple, `.com`.
   - Vous pouvez spécifier un nom de domaine international (IDN) si vous convertissez d'abord le nom en Punycode. Pour plus d'informations sur les convertisseurs en ligne, recherchez « convertisseur punycode » sur Internet.

     Vous pouvez également convertir un nom de domaine international (IDN) en Punycode quand vous créez un espace de noms par programmation. Par exemple,

si vous utilisez Java, vous pouvez convertir une valeur Unicode en Punycode à l'aide de la méthode `toASCII` de la bibliothèque java.net.IDN.

4. (Facultatif) Pour la description de l'espace de noms, entrez les informations relatives à l'espace de noms qui seront visibles sur la page Espaces de noms et sous Informations sur l'espace de noms. Vous pouvez utiliser ces informations pour identifier facilement un espace de noms.
5. Pour la découverte d'instances, vous pouvez choisir entre des appels d'API, des appels d'API et des requêtes DNS dans VPCs, et des appels d'API et des requêtes DNS publiques pour créer respectivement un espace de noms HTTP, DNS privé ou DNS public. Pour de plus amples informations, veuillez consulter Options de découverte d'instances.

   En fonction de votre sélection, procédez comme suit.

   - Si vous choisissez des appels d'API et des requêtes DNS dans VPCs, pour VPC, choisissez un cloud privé virtuel (VPC) auquel vous souhaitez associer l'espace de noms.
   - Si vous choisissez des appels d'API et des requêtes DNS dans VPCs ou des appels d'API et des requêtes DNS publiques, pour le TTL, spécifiez une valeur numérique en secondes. La valeur du temps de vie (TTL) détermine la durée pendant laquelle les résolveurs DNS mettent en cache les informations relatives à l'enregistrement DNS de début d'autorité (SOA) de la zone hébergée Route 53 créée avec votre espace de noms. Pour plus d'informations sur le TTL, consultez TTL (secondes) dans le manuel Amazon Route 53 Developer Guide.
6. (Facultatif) Sous Balises, choisissez Ajouter des balises, puis spécifiez une clé et une valeur pour étiqueter votre espace de noms. Vous pouvez spécifier une ou plusieurs balises à ajouter à votre espace de noms. Les balises vous permettent de classer vos AWS ressources afin de les gérer plus facilement. Pour de plus amples informations, veuillez consulter Marquer vos ressources AWS Cloud Map.
7. Choisissez Create namespace (Créer un espace de noms). Vous pouvez consulter le statut de l'opération en utilisant ListOperations. Pour plus d'informations, consultez le ListOperationsGuide de référence de l'AWS Cloud Map API

AWS CLI

- Créez un espace de noms avec la commande correspondant au type de découverte d'instance que vous préférez (remplacez les *red* valeurs par les vôtres).

- Créez un espace de noms HTTP à l'aide `create-http-namespace` de. Les instances de service enregistrées à l'aide d'un espace de noms HTTP peuvent être découvertes à l'aide d'une `DiscoverInstances` requête, mais elles ne peuvent pas être découvertes à l'aide du DNS.

```
aws servicediscovery create-http-namespace --name name-of-namespace
```

- Créez un espace de noms privé basé sur le DNS et visible uniquement dans un Amazon `create-private-dns-namespace` VPC spécifié à l'aide de. Vous pouvez découvrir des instances enregistrées avec un espace de noms DNS privé en utilisant une `DiscoverInstances` requête ou en utilisant le DNS

```
aws servicediscovery create-private-dns-namespace --name name-of-namespace --
vpc vpc-xxxxxxxxx
```

- Créez un espace de noms public basé sur le DNS qui est visible sur Internet à l'aide `create-public-dns-namespace` de. Vous pouvez détecter les instances qui ont été enregistrées dans un espace de noms DNS public en utilisant une demande `DiscoverInstances` ou le DNS.

```
aws servicediscovery create-public-dns-namespace --name name-of-namespace
```

AWS SDK for Python (Boto3)

1. Si ce n'est pas déjà `Boto3` fait, vous trouverez les instructions d'installation, de configuration et d'utilisation `Boto3` ici.
2. Importez `Boto3` et utilisez `servicediscovery` en tant que service.

```
import boto3
client = boto3.client('servicediscovery')
```

3. Créez un espace de noms avec la commande correspondant au type de découverte d'instance que vous préférez (remplacez les *red* valeurs par les vôtres) :

   - Créez un espace de noms HTTP à l'aide `create_http_namespace()` de. Les instances de service enregistrées à l'aide d'un espace de noms HTTP peuvent être découvertes à

l'aide du DNS`discover_instances()`, mais elles ne peuvent pas être découvertes à l'aide du DNS.

```
response = client.create_http_namespace(
    Name='name-of-namespace',
)
# If you want to see the response
print(response)
```

- Créez un espace de noms privé basé sur le DNS et visible uniquement dans un Amazon `create_private_dns_namespace()` VPC spécifié à l'aide de. Vous pouvez découvrir les instances enregistrées avec un espace de noms DNS privé en utilisant soit le DNS, `discover_instances()` soit en utilisant le DNS

```
response = client.create_private_dns_namespace(
    Name='name-of-namespace',
    Vpc='vpc-1c56417b',
)
# If you want to see the response
print(response)
```

- Créez un espace de noms public basé sur le DNS qui est visible sur Internet à l'aide `create_public_dns_namespace()` de. Vous pouvez découvrir les instances enregistrées auprès d'un espace de noms DNS public en utilisant l'un `discover_instances()` ou l'autre des systèmes DNS.

```
response = client.create_public_dns_namespace(
    Name='name-of-namespace',
)
# If you want to see the response
print(response)
```

- Exemple de sortie de réponse

```
{
    'OperationId': 'gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd',
    'ResponseMetadata': {
        '...': '...',
    },
}
```

Étapes suivantes

Après avoir créé un espace de noms, vous pouvez créer des services dans l'espace de noms pour regrouper les ressources d'application qui répondent collectivement à un objectif particulier dans votre application. Un service agit comme un modèle pour enregistrer les ressources de l'application en tant qu'instances. Pour plus d'informations sur la création de AWS Cloud Map services, consultezCréation d'un AWS Cloud Map service pour un composant d'application.

Lister les espaces AWS Cloud Map de noms

Après avoir créé des espaces de noms, vous pouvez consulter la liste des espaces de noms que vous avez créés en suivant ces étapes.

AWS Management Console

1. Connectez-vous à la AWS Cloud Map console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudmap/.
2. Dans le volet de navigation, choisissez Namespaces pour afficher la liste des espaces de noms. Vous pouvez classer les espaces de noms par nom, description, mode de découverte d'instance, propriétaire ou ID d'espace de noms. Vous pouvez également saisir un nom ou un ID d'espace de noms dans le champ de recherche pour localiser et afficher un espace de noms spécifique.

AWS CLI

- Répertoriez les espaces de noms à l'aide de la `list-namespaces` commande.

```
aws servicediscovery list-namespaces
```

AWS SDK for Python (Boto3)

1. Si ce n'est pas déjà `Boto3` fait, vous trouverez les instructions d'installation, de configuration et d'utilisation `Boto3` ici.
2. Importez `Boto3` et utilisez `servicediscovery` en tant que service.

```
import boto3
client = boto3.client('servicediscovery')
```

3. Répertoriez les espaces de noms avec`list_namespaces()`.

```
response = client.list_namespaces()
# If you want to see the response
print(response)
```

Exemple de sortie de réponse

```
{
    'Namespaces': [
        {
            'Arn': 'arn:aws::servicediscovery:us-west-2:123456789012:namespace/
ns-xxxxxxxxxxxxxx',
            'CreateDate': 1585354387.357,
            'Id': 'ns-xxxxxxxxxxxxxx',
            'Name': 'myFirstNamespace',
            'Properties': {
                'DnsProperties': {
                    'HostedZoneId': 'Z06752353VBUDTC32S84S',
                },
                'HttpProperties': {
                    'HttpName': 'myFirstNamespace',
                },
            },
            'Type': 'DNS_PRIVATE',
        },
        {
            'Arn': 'arn:aws::servicediscovery:us-west-2:123456789012:namespace/
ns-xxxxxxxxxxxxxx',
            'CreateDate': 1586468974.698,
            'Description': 'My second namespace',
            'Id': 'ns-xxxxxxxxxxxxxx',
            'Name': 'mySecondNamespace.com',
            'Properties': {
                'DnsProperties': {
                },
                'HttpProperties': {
                    'HttpName': 'mySecondNamespace.com',
                },
            },
            'Type': 'HTTP',
        },
        {
```

```
            'Arn': 'arn:aws::servicediscovery:us-west-2:123456789012:namespace/
ns-xxxxxxxxxxxxxx',
            'CreateDate': 1587055896.798,
            'Id': 'ns-xxxxxxxxxxxxxx',
            'Name': 'myThirdNamespace.com',
            'Properties': {
                'DnsProperties': {
                    'HostedZoneId': 'Z09983722P0QME1B3KC8I',
                },
                'HttpProperties': {
                    'HttpName': 'myThirdNamespace.com',
                },
            },
            'Type': 'DNS_PRIVATE',
        },
    ],
    'ResponseMetadata': {
        '...': '...',
    },
}
```

Supprimer un espace de AWS Cloud Map noms

Une fois que vous avez fini d'utiliser un espace de noms, vous pouvez le supprimer. Lorsque vous supprimez un espace de noms, vous ne pouvez plus l'utiliser pour enregistrer ou découvrir des instances de service.

> **Note**
>
> Lorsque vous supprimez un espace de noms DNS, AWS Cloud Map la zone hébergée Amazon Route 53 correspondante créée lors de la création de l'espace de noms est supprimée.

Avant de supprimer un espace de noms, vous devez désenregistrer toutes les instances de service, puis supprimer tous les services créés dans l'espace de noms. Pour plus d'informations, consultez Annulation de l'enregistrement d'une instance de service AWS Cloud Map et Supprimer un AWS Cloud Map service.

Après avoir désenregistré les instances et supprimé les services créés dans un espace de noms, procédez comme suit pour supprimer l'espace de noms.

AWS Management Console

1. Connectez-vous à la AWS Cloud Map console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudmap/.
2. Dans le panneau de navigation, choisissez Namespaces (Espaces de noms).
3. Sélectionnez l'espace de noms que vous souhaitez supprimer, puis choisissez Supprimer.
4. Confirmez que vous souhaitez supprimer le service en sélectionnant à nouveau Supprimer.

AWS CLI

- Supprimez un espace de noms à l'aide de la `delete-namespace` commande (remplacez la *red* valeur par la vôtre). Si l'espace de noms contient toujours un ou plusieurs services, la demande échoue.

```
aws servicediscovery delete-namespace --id ns-xxxxxxxxxxx
```

AWS SDK for Python (Boto3)

1. Si ce n'est pas déjà `Boto3` fait, vous trouverez les instructions d'installation, de configuration et d'utilisation `Boto3` ici.
2. Importez `Boto3` et utilisez `servicediscovery` en tant que service.

```
import boto3
client = boto3.client('servicediscovery')
```

3. Supprimez un espace de noms par `delete_namespace()` (remplacez la *red* valeur par la vôtre). Si l'espace de noms contient toujours un ou plusieurs services, la demande échoue.

```
response = client.delete_namespace(
    Id='ns-xxxxxxxxxxx',
)
# If you want to see the response
print(response)
```

Exemple de sortie de réponse

```
{
    'OperationId': 'gv4g5meo7ndmeh4fqskygvk23d2fijwa-k98y6drk',
    'ResponseMetadata': {
        '...': '...',
    },
}
```

Espaces de AWS Cloud Map noms partagés

AWS Cloud Map permet aux propriétaires d'espaces de noms de partager leurs espaces de noms avec d'autres personnes Comptes AWS ou au sein d'une organisation afin de simplifier la découverte AWS Organizations des services entre comptes et le registre des services. Cela permet d'utiliser plus facilement les espaces de noms gérés par d'autres personnes Comptes AWS ou par des équipes au sein d'une AWS organisation.

AWS Cloud Map s'intègre à AWS Resource Access Manager (AWS RAM) pour permettre le partage des ressources. AWS RAM est un service qui vous permet de partager certaines AWS Cloud Map ressources avec d'autres personnes Comptes AWS ou par le biais de AWS Organizations. Avec AWS RAM, vous partagez les ressources que vous possédez en créant un partage de ressources. Un partage de ressources spécifie les ressources à partager, ainsi que les consommateurs avec qui elles seront partagées. Les consommateurs peuvent être :

- Spécifique au Comptes AWS sein de son organisation en AWS Organizations
- Une unité organisationnelle au sein de son organisation dans AWS Organizations
- Toute son organisation en AWS Organizations

Pour plus d'informations AWS RAM, consultez le guide de AWS RAM l'utilisateur.

Cette rubrique explique comment partager les ressources que vous possédez et comment utiliser les ressources qui sont partagées avec vous.

Table des matières

- Arrêter de partager un espace AWS Cloud Map de noms
- Identification d'un espace de AWS Cloud Map noms partagé
- Octroi d'autorisations pour partager un espace de noms
- Responsabilités et autorisations pour les espaces de noms partagés
- Facturation et mesures
- Quotas

Considérations relatives au partage d'espaces de noms

- Pour partager un espace de noms, vous devez en être le propriétaire dans votre Compte AWS. Cela signifie que la ressource doit être allouée ou provisionnée dans votre compte. Vous ne pouvez pas partager un espace de noms qui a été partagé avec vous.
- Pour partager un espace de noms avec votre organisation ou une unité organisationnelle dans AWS Organizations, vous devez activer le partage avec AWS Organizations. Pour plus d'informations, consultez Activation du partage avec AWS Organizations dans le Guide de l'utilisateur AWS RAM .
- Pour la découverte de services à l'aide de requêtes DNS dans un espace de noms DNS privé partagé, le propriétaire de l'espace de noms devra appeler `create-vpc-association-authorization` avec l'ID de la zone hébergée privée associée à l'espace de noms et le VPC du consommateur.

  ```
  aws route53 create-vpc-association-authorization --hosted-zone-id Z1234567890ABC --
  vpc VPCRegion=us-east-1,VPCId=vpc-12345678
  ```

  Le consommateur de l'espace de noms devra appeler `associate-vpc-with-hosted-zone` avec l'ID de la zone hébergée privée.

  ```
  aws route53 associate-vpc-with-hosted-zone --hosted-zone-id Z1234567890ABC --vpc
   VPCRegion=us-east-1,VPCId=vpc-12345678
  ```

  Pour plus d'informations, consultez la section Associer un Amazon VPC à une zone hébergée privée que vous avez créée avec une autre méthode Comptes AWS dans le manuel Amazon Route 53 Developer Guide.
- Après avoir découvert les emplacements up-to-date réseau des services associés à un espace de noms DNS partagé, il peut être nécessaire de configurer la connectivité inter-VPC pour

communiquer avec les services s'ils se trouvent dans des emplacements différents. VPCs Cela peut être réalisé à l'aide d'une connexion d'appairage de VPC. Pour plus d'informations, consultez la section Création ou suppression d'une connexion d'appairage de VPC dans le Guide d'appairage de VPC Amazon Virtual Private Cloud.

- Vous ne pouvez pas l'utiliser `ListOperations` pour répertorier les opérations effectuées par d'autres comptes sur des espaces de noms partagés.
- Le balisage n'est pas pris en charge pour les espaces de noms partagés.

Partage d'un espace AWS Cloud Map de noms

Lorsque vous partagez un AWS Cloud Map espace de noms dont vous êtes propriétaire avec d'autres Comptes AWS (consommateurs), vous permettez à ces comptes de découvrir l'emplacement up-to-date réseau des services dans cet espace de noms sans avoir besoin d'informations d'identification temporaires.

Pour partager un espace de noms, vous devez l'ajouter à un partage de ressources. Un partage de ressources est une ressource AWS RAM qui vous permet de partager vos ressources entre des Comptes AWS. Un partage de ressources spécifie les ressources à partager, ainsi que les consommateurs avec qui elles seront partagées. Pour ajouter l'espace de noms à un nouveau partage de ressources, vous devez d'abord créer le partage de ressources à l'aide de la AWS RAM console.

Si vous faites partie d'une organisation AWS Organizations et que le partage au sein de votre organisation est activé, les consommateurs de votre organisation ont automatiquement accès à l'espace de noms partagé. Dans le cas contraire, les consommateurs reçoivent une invitation à rejoindre le partage de ressources et ont accès à l'espace de noms partagé après avoir accepté l'invitation.

Vous pouvez partager un espace de noms dont vous êtes propriétaire à l'aide de la AWS RAM console ou du AWS CLI.

AWS RAM console

Pour partager un espace de noms dont vous êtes propriétaire à l'aide de la console AWS RAM

Consultez Création d'un partage de ressources dans AWS RAM dans le Guide de l'utilisateur AWS RAM .

AWS CLI

Pour partager un espace de noms dont vous êtes propriétaire à l'aide du AWS CLI

Utilisez la commande AWS RAM create-resource-share.

Arrêter de partager un espace AWS Cloud Map de noms

Lorsqu'un espace de noms n'est plus partagé, le consommateur ne peut plus accéder à l'espace de noms ainsi qu'aux services et instances qui lui sont associés. Comptes AWS Cela inclut les ressources créées dans l'espace de noms par les consommateurs lorsqu'ils avaient accès à l'espace de noms.

Pour arrêter de partager un espace de noms dont vous êtes le propriétaire, vous devez le supprimer du partage de ressources. Vous pouvez le faire à l'aide de la AWS RAM console ou du AWS CLI.

AWS RAM console

Pour arrêter de partager un espace de noms dont vous êtes propriétaire à l'aide de la console AWS RAM

Consultez Mise à jour d'un partage de ressources dans le Guide de l'utilisateur AWS RAM .

AWS CLI

Pour arrêter de partager un espace de noms dont vous êtes le propriétaire à l'aide du AWS CLI

Utilisez la commande disassociate-resource-share.

Identification d'un espace de AWS Cloud Map noms partagé

Les propriétaires et les consommateurs peuvent identifier les espaces de noms partagés à l'aide de la AWS Cloud Map console et AWS CLI. Le propriétaire de l'espace de noms peut être identifié à l'aide de la `ResourceOwner` propriété. Celui Compte AWS qui crée un service ou enregistre une instance dans l'espace de noms partagé peut être identifié à l'aide de la `CreatedByAccount` propriété.

AWS Cloud Map console

Pour identifier un espace de noms partagé à l'aide de la console AWS Cloud Map

1. Connectez-vous à la AWS Cloud Map console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudmap/.
2. Sur la page Espaces de noms, sous Propriétaire de la ressource, vous trouverez l'ID du propriétaire de Compte AWS l'espace de noms.
3. Choisissez le nom de domaine de l'espace de noms que vous souhaitez identifier.
4. Sur la *namespace-name* page Namespace :, dans la section Informations sur l'espace de noms, sous Propriétaire de la ressource, vous pouvez trouver l'ID du propriétaire de l' Compte AWS espace de noms.

AWS CLI

Pour identifier un espace de noms partagé à l'aide de AWS CLI, utilisez la commande list-namespaces. La commande renvoie les espaces de noms que vous possédez et les espaces de noms partagés avec vous. Le `ResourceOwner` champ indique l'ID de AWS compte du propriétaire de l'espace de noms.

L'`list-namespaces`appel suivant est effectué par compte`111122223333`.

```
aws servicediscovery list-namespaces
```

Sortie :

```
{
    "Namespaces": [
        {
            "Arn": "arn:aws:servicediscovery:us-west-2:111122223333:namespace/ns-
abcdef01234567890",
            "CreateDate": 1585354387.357,
            "Id": "ns-abcdef01234567890",
            "Name": "local",
            "Properties": {
                "DnsProperties": {
                    "HostedZoneId": "Z06752353VBUDTC32S84S"
                },
                "HttpProperties": {
                    "HttpName": "local"
```

```
                }
            },
            "Type": "DNS_PRIVATE",
            "ServiceCount": 2,
           "ResourceOwner": "111122223333"
        },
        {
            "Arn": "arn:aws:servicediscovery:us-west-2:444455556666:namespace/
ns-021345abcdef6789",
            "CreateDate": 1586468974.698,
            "Description": "Shared second namespace",
            "Id": "ns-021345abcdef6789",
            "Name": "My-second-namespace",
            "Properties": {
                "DnsProperties": {},
                "HttpProperties": {
                    "HttpName": "Shared-second-namespace"
                }
            },
            "Type": "HTTP",
            "ServiceCount": 0,
            "ResourceOwner": "444455556666"
        }
    ]
}
```

Dans ce scénario, l'espace de noms `ns-abcdef01234567890` est créé et détenu par `111122223333` et l'espace de noms `ns-021345abcdef6789` est créé et détenu par. `444455556666` L'espace de noms `ns-021345abcdef6789` est partagé compte `111122223333` par compte. `444455556666`

Octroi d'autorisations pour partager un espace de noms

Un ensemble minimal d'autorisations est requis pour qu'un principal IAM puisse partager un espace de noms. Nous vous recommandons d'utiliser les politiques `AWSResourceAccessManagerFullAccess` gérées `AWSCloudMapFullAccess` et pour vous assurer que vos principaux IAM disposent des autorisations requises pour partager et utiliser des espaces de noms partagés.

Si vous utilisez une politique IAM personnalisée, les `servicediscovery:DeleteResourcePolicy` actions

`servicediscovery:PutResourcePolicyservicediscovery:GetResourcePolicy`, et sont requises pour partager des espaces de noms. Il s'agit d'actions IAM avec autorisation uniquement. Si ces autorisations ne sont pas accordées à un principal IAM, une erreur se produira lors de la tentative de partage de l'espace de noms en utilisant. AWS RAM

Pour plus d'informations sur l' AWS RAM utilisation d'IAM, consultez la section Comment AWS RAM utilise IAM dans le guide de l'AWS RAM utilisateur.

Responsabilités et autorisations pour les espaces de noms partagés

Le propriétaire et le consommateur de l'espace de noms peuvent effectuer différentes actions sur un espace de noms partagé.

Autorisations accordées aux propriétaires

Le propriétaire d'un espace de noms peut effectuer les actions suivantes sur un espace de noms partagé :

- Accédez aux services associés à l'espace de noms, y compris les services créés par les comptes clients et les instances enregistrées auprès de ces services.
- Révoquez l'accès à l'espace de noms, y compris l'accès aux services créés par les comptes de consommateurs et les instances enregistrées auprès de ces services.
- Configurez les autorisations pour les autres comptes afin d'enregistrer et de désenregistrer des instances dans les services créés dans l'espace de noms partagé par les consommateurs ou le propriétaire de l'espace de noms.
- Supprimez des services et annulez l'enregistrement des instances, y compris les services créés et les instances enregistrées par les comptes clients.
- Mettez à jour ou supprimez un espace de noms partagé.

Autorisations accordées aux consommateurs

Un consommateur d'espace de noms peut effectuer les actions suivantes sur un espace de noms partagé :

- Créez et supprimez des services dans l'espace de noms.
- Enregistrez et désenregistrez des instances dans les services créés dans l'espace de noms.
- Découvrez les instances enregistrées auprès des services créés dans l'espace de noms.

Un consommateur ne peut pas mettre à jour ou supprimer un espace de noms partagé. Après avoir perdu l'accès à l'espace de noms partagé, les comptes clients perdront également l'accès aux services qu'ils ont créés dans l'espace de noms.

Facturation et mesures

Les propriétaires sont facturés pour toutes les instances qu'ils enregistrent dans l'espace de noms partagé et pour tous les bilans de santé de Route 53 créés lors de l'enregistrement de ces instances. Les consommateurs sont facturés pour toutes les instances qu'ils enregistrent dans l'espace de noms et pour tous les bilans de santé Route 53 créés lors de l'enregistrement de ces instances. Si l'espace de noms partagé est un espace de noms DNS, le propriétaire de l'espace de noms est facturé pour les enregistrements DNS Route 53 créés lors de la création de services dans l'espace de noms. Les propriétaires sont facturés pour tous `DiscoverInstances` les `DiscoverInstancesRevision` appels qu'ils passent. Les consommateurs sont facturés pour tous `DiscoverInstances` les `DiscoverInstancesRevision` appels qu'ils passent.

Quotas

Les espaces de noms partagés ne sont pris en compte que pour les espaces de noms du propriétaire de l'espace de noms par quota de région. Les instances enregistrées par un consommateur dans l'espace de noms partagé sont prises en compte dans le quota d'instances du propriétaire par espace de noms. Si un client crée un service dans un espace de noms partagé, toutes les instances enregistrées dans le service sont prises en compte dans le quota d'instances du consommateur par service. Si un propriétaire crée un service dans un espace de noms partagé, toutes les instances enregistrées dans le service sont prises en compte dans le calcul des instances du propriétaire par quota de service.

AWS Cloud Map services

Un AWS Cloud Map service est un modèle d'enregistrement des instances de service qui comprend le nom du service et la configuration DNS, le cas échéant, du service. Vous pouvez également configurer un bilan de santé pour déterminer l'état de santé des instances du service et filtrer les ressources non saines. Un service peut représenter un composant de votre application. Par exemple, vous pouvez créer un service pour les ressources qui gèrent les paiements sur votre application et un autre pour les ressources qui gèrent les utilisateurs.

Un service vous permet de localiser les ressources d'une application en récupérant un ou plusieurs points de terminaison qui peuvent être utilisés pour se connecter à la ressource. L'emplacement des ressources est effectué à l'aide de requêtes DNS ou de l'action de l' AWS Cloud Map `DiscoverInstances`API, selon la façon dont vous avez configuré l'espace de noms. Vous pouvez utiliser la AWS Cloud Map console pour définir le périmètre de découverte des instances au niveau du service.

Vous pouvez également spécifier des métadonnées personnalisées sous forme d'attributs au niveau du service à l'aide de l'`UpdateServiceAttributes`API. Vous pouvez définir des attributs de service pour éviter de dupliquer les attributs entre les instances et modifier ces attributs sans avoir à apporter de modifications aux attributs de l'instance. Les informations que vous pouvez spécifier sous forme d'attributs au niveau du service incluent, sans toutefois s'y limiter, les suivantes :

- Pondération des terminaux pour l'évolution du trafic lors de déploiements progressifs.
- Les préférences de service, telles que les délais d'expiration des API et les politiques de nouvelle tentative suggérées.

Pour plus d'informations, consultez UpdateServiceAttributesla référence de AWS Cloud Map l'API.

Les rubriques suivantes décrivent le contrôle de santé et les configurations DNS des services et incluent des instructions pour créer, répertorier, mettre à jour et supprimer un service.

Rubriques

- Répertorier AWS Cloud Map les services dans un espace de noms
- Supprimer un AWS Cloud Map service

AWS Cloud Map configuration du contrôle de santé du service

Les bilans de santé permettent de déterminer si les instances de service sont saines ou non. Si vous ne configurez pas de contrôle de santé lors de la création du service, le trafic sera acheminé vers les instances de service quel que soit l'état de santé des instances. Lorsque vous configurez un bilan de santé, AWS Cloud Map renvoie des ressources saines par défaut. Vous pouvez utiliser le `HealthStatus` paramètre de l'`DiscoverInstances`API pour filtrer les ressources par état de santé et obtenir une liste des ressources non fonctionnelles. Vous pouvez également utiliser l'`GetInstancesHealthStatus`API pour récupérer l'état de santé d'une instance de service particulière.

Vous pouvez configurer une vérification de l'état de Route 53 ou une vérification de santé personnalisée par un tiers lorsque vous créez un AWS Cloud Map service.

Surveillances d'états Route 53

Si vous définissez les paramètres d'un bilan de santé d'Amazon Route 53, AWS Cloud Map crée un bilan de santé Route 53 chaque fois que vous enregistrez une instance et supprimez le bilan de santé lorsque vous annulez l'enregistrement de l'instance.

Pour les espaces de noms DNS publics, AWS Cloud Map associe le contrôle de santé à l'enregistrement Route 53 AWS Cloud Map créé lorsque vous enregistrez une instance. Si vous spécifiez les deux `A` types d'`AAAA`enregistrement dans la configuration DNS d'un service, AWS Cloud Map crée un contrôle de santé qui utilise l' IPv4 adresse pour vérifier l'état de la ressource. Si le point de terminaison spécifié par l' IPv4 adresse est défectueux, Route 53 considère que les `AAAA` enregistrements `A` et ne sont pas sains. Si vous spécifiez un type d'`CNAME`enregistrement dans la configuration DNS d'un service, vous ne pouvez pas configurer une vérification de l'état de Route 53.

Pour les espaces de noms pour lesquels vous utilisez des appels d'API pour découvrir des instances, AWS Cloud Map crée une vérification de l'état de Route 53. Cependant, il n'existe aucun enregistrement DNS AWS Cloud Map auquel associer le bilan de santé. Pour déterminer si un bilan de santé est sain, vous pouvez configurer la surveillance à l'aide de la console Route 53 ou d'Amazon CloudWatch. Pour plus d'informations sur l'utilisation de la console Route 53, consultez Get Notified When a Health Check Fails dans le manuel Amazon Route 53 Developer Guide. Pour

plus d'informations sur l'utilisation CloudWatch, consultez PutMetricAlarmle Amazon CloudWatch API Reference.

> **Note**
>
> - Vous ne pouvez pas configurer un contrôle de santé Amazon Route 53 pour un service créé dans un espace de noms DNS privé.
> - Lors de chaque contrôle d'état, un contrôleur de santé Route 53 Région AWS envoie une demande de bilan de santé à un point de terminaison toutes les 30 secondes. En moyenne, votre point de terminaison reçoit une demande de vérification de l'état toutes les deux secondes. Cependant, les contrôleurs de l'état ne se coordonnent pas les uns avec les autres. C'est la raison pour laquelle vous verrez parfois plusieurs demandes en une seconde, suivies par quelques secondes sans surveillance de l'état. Pour une liste des régions où l'état de santé est vérifié, voir Régions.

Pour plus d'informations sur les frais liés aux bilans de santé de la Route 53, consultez la section Tarification de la Route 53.

Surveillances d'état personnalisées

Si vous configurez AWS Cloud Map pour utiliser un contrôle de santé personnalisé lorsque vous enregistrez une instance, vous devez utiliser un vérificateur de santé tiers pour évaluer l'état de vos ressources. Les vérifications de l'état personnalisées s'avèrent utiles dans les situations suivantes :

- Vous ne pouvez pas utiliser le bilan de santé de Route 53 car la ressource n'est pas disponible sur Internet. Supposons, par exemple, que vous disposiez d'une instance située dans un Amazon VPC. Vous pouvez utiliser un bilan de santé personnalisé pour cette instance. Toutefois, pour que le bilan de santé fonctionne, votre vérificateur de santé doit également se trouver dans le même VPC que votre instance.
- Vous souhaitez utiliser un outil de vérification de l'état tiers quel que soit l'emplacement de vos ressources.

Lorsque vous utilisez un bilan de santé personnalisé, AWS Cloud Map il ne vérifie pas directement l'état d'une ressource donnée. Au lieu de cela, le vérificateur d'état tiers vérifie l'état de santé de la ressource et renvoie un statut à votre application. Votre candidature devra ensuite soumettre une `UpdateInstanceCustomHealthStatus` demande qui transmettra ce statut

à AWS Cloud Map. Si le statut initial transmis l'estUNHEALTHY, et s'il n'y en a pas un autre `UpdateInstanceCustomHealthStatus` dans les 30 secondes qui indique le statut deHEALTHY, il est confirmé que la ressource n'est pas saine. AWS Cloud Map arrête d'acheminer le trafic vers cette ressource.

AWS Cloud Map configuration DNS du service

Lorsque vous créez un service dans un espace de noms qui prend en charge la découverte d'instances par des requêtes DNS, il AWS Cloud Map crée des enregistrements DNS Route 53. Vous devez spécifier une politique de routage Route 53 et un type d'enregistrement DNS qui s'appliqueront à tous les enregistrements DNS Route 53 AWS Cloud Map créés.

Stratégie de routage

Une politique de routage détermine la manière dont Route 53 répond aux requêtes DNS utilisées pour la découverte des instances de service. Les politiques de routage prises en charge et leur relation avec AWS Cloud Map celles-ci sont les suivantes.

Weighted routing (Routage pondéré)

Route 53 renvoie la valeur applicable à partir d'une instance de AWS Cloud Map service sélectionnée au hasard parmi les instances que vous avez enregistrées en utilisant le même AWS Cloud Map service. Tous les enregistrements ont la même pondération. Vous ne pouvez donc pas acheminer plus ou moins de trafic vers des instances.

Supposons, par exemple, que le service inclut des configurations pour un enregistrement A et un bilan de santé, et que vous utilisiez le service pour enregistrer 10 instances. Route 53 répond aux requêtes DNS avec l'adresse IP pour une instance sélectionnée de façon aléatoire parmi les instances saines. Si aucune instance n'est saine, Route 53 répond aux requêtes DNS comme si toutes les instances étaient saines.

Si vous ne définissez pas une vérification de l'état pour le service, Route 53 suppose que toutes les instances sont saines et renvoie la valeur applicable pour une instance sélectionnée de façon aléatoire.

Pour plus d'informations, consultez la section Weighted Routing dans le guide du développeur Amazon Route 53.

Multivalue answer routing (Routage de réponse multivaleur)

Si vous définissez un bilan de santé pour le service et que le résultat du bilan de santé est sain, Route 53 renvoie la valeur applicable pour un maximum de huit instances.

Supposons, par exemple, que le service inclut des configurations pour un enregistrement A et un bilan de santé. et que vous utilisez le service pour enregistrer 10 instances. Route 53 répond aux requêtes DNS avec des adresses IP pour un maximum de huit instances saines. Si moins de huit instances sont saines, Route 53 répond à chaque requête DNS avec les adresses IP de toutes les instances saines.

Si vous ne définissez pas une vérification de l'état pour le service, Route 53 suppose que toutes les instances sont saines et renvoie les valeurs pour huit instances maximum.

Pour plus d'informations, consultez la section Routage des réponses à valeurs multiples dans le manuel Amazon Route 53 Developer Guide.

Type de registre

Un type d'enregistrement DNS Route 53 détermine le type de valeur renvoyée par Route 53 en réponse aux requêtes DNS utilisées pour la découverte d'instances de service. Les différents types d'enregistrement DNS que vous pouvez spécifier et les valeurs associées renvoyées par Route 53 en réponse aux requêtes sont les suivants.

A

Si vous spécifiez ce type, Route 53 renvoie l'adresse IP de la ressource au IPv4 format 192.0.2.44.

AAAA

Si vous spécifiez ce type, Route 53 renvoie l'adresse IP de la ressource dans un IPv6 format tel que 2001:0 db 8:85 a 3:0000:0000:abcd : 0001:2345.

CNAME

Si vous spécifiez ce type, Route 53 renvoie le nom de domaine de la ressource (tel que www.example.com).

> **Note**
>
> - Pour configurer un enregistrement DNS CNAME, vous devez spécifier la politique de routage pondérée.
> - Lorsque vous configurez un enregistrement DNS CNAME, vous ne pouvez pas configurer un contrôle de santé Route 53.

SRV

Si vous spécifiez ce type, Route 53 renvoie la valeur d'un SRV enregistrement. La valeur pour un enregistrement SRV utilise les valeurs suivantes :

```
priority weight port service-hostname
```

Éléments à prendre en compte :

- Les valeurs de `priority` et `weight` sont toutes les deux définies sur 1 et ne peuvent pas être modifiées.
- Pour`port`, AWS Cloud Map utilise la valeur que vous spécifiez pour Port (AWS_INSTANCE_PORT) lorsque vous enregistrez une instance.
- La valeur de `service-hostname` est une concaténation des valeurs suivantes :
  - La valeur que vous spécifiez pour l'ID d'instance de service (InstanceID) lorsque vous enregistrez une instance
  - Le nom du service
  - Le nom de l'espace de noms

  Supposons, par exemple, que vous spécifiez test comme ID d'instance lorsque vous enregistrez une instance. Le nom du service est backend et le nom de l'espace de noms est example.com. AWS Cloud Map attribue la valeur suivante à l'`service-hostname`attribut dans l'enregistrement SRV :

  ```
  test.backend.example.com
  ```

> **Note**
>
> Si vous spécifiez des valeurs IPv4 telles qu'une IPv6 adresse, une adresse ou les deux lorsque vous enregistrez une instance, des enregistrements A and/or AAAA portant le

même nom que la valeur de `service-hostname` l'enregistrement SRV sont AWS Cloud Map automatiquement créés.

Vous pouvez spécifier des types d'enregistrement dans les combinaisons suivantes :

- A
- AAAA
- A et AAAA
- CNAME
- SRV

Si vous spécifiez les types d'enregistrement A et AAAA, vous pouvez spécifier une adresse IPv4 IP, une adresse IPv6 IP ou les deux lorsque vous enregistrez une instance.

Création d'un AWS Cloud Map service pour un composant d'application

Après avoir créé un espace de noms, vous pouvez créer des services pour représenter les différents composants de votre application qui répondent à des objectifs particuliers. Par exemple, vous pouvez créer un service pour les ressources de votre application qui traitent les paiements.

> **Note**
>
> Vous ne pouvez pas créer plusieurs services accessibles par des requêtes DNS dont les noms ne diffèrent que par cas (par exemple et exemple). Si vous essayez de le faire, ces services porteront le même nom DNS. Si vous utilisez un espace de noms accessible uniquement par des appels d'API, vous pouvez créer des services dont les noms ne diffèrent qu'au cas par cas.

Procédez comme suit pour créer un service à l'aide du AWS Management Console AWS CLI, et du SDK pour Python.

AWS Management Console

1. Connectez-vous à la AWS Cloud Map console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudmap/.
2. Dans le panneau de navigation, choisissez Namespaces (Espaces de noms).
3. Sur la page Namespaces (Espaces de noms), choisissez l'espace de noms auquel vous souhaitez ajouter le service.
4. Sur la *namespace-name* page Namespace :, choisissez Create service.
5. Dans Nom du service, entrez un nom qui décrit les instances que vous enregistrez lorsque vous utilisez ce service. La valeur est utilisée pour découvrir les instances AWS Cloud Map de service dans les appels d'API ou dans les requêtes DNS.

   > **Note**
   >
   > Si vous AWS Cloud Map souhaitez créer un enregistrement SRV lorsque vous enregistrez une instance et que vous utilisez un système qui nécessite un format SRV spécifique (tel que HAProxy), spécifiez ce qui suit pour le nom du service :
   >
   > - Commencez le nom par un trait de soulignement (_), par exemple _exampleservice.
   > - Terminez le nom par *._protocol*, par exemple. _tcp.
   >
   > Lorsque vous enregistrez une instance, AWS Cloud Map créez un enregistrement SRV et attribuez un nom en concaténant le nom du service et le nom de l'espace de noms, par exemple : _exampleservice._tcp.example.com

6. (Facultatif) Dans Description du service, entrez une description du service. La description que vous entrez ici apparaît sur la page Services et sur la page détaillée de chaque service.
7. Si l'espace de noms prend en charge les requêtes DNS, sous Configuration de la découverte des services, vous pouvez configurer la découvrabilité au niveau du service. Choisissez entre autoriser à la fois les appels d'API et les requêtes DNS ou uniquement les appels d'API pour la découverte d'instances dans ce service.

> **Note**
>
> Si vous choisissez les appels d'API, aucun enregistrement SRV ne AWS Cloud Map sera créé lorsque vous enregistrez une instance.

Si vous choisissez API et DNS, procédez comme suit pour configurer les enregistrements DNS. Vous pouvez ajouter ou supprimer des enregistrements DNS.

1. Pour la politique de routage, sélectionnez la politique de routage Amazon Route 53 pour les enregistrements DNS AWS Cloud Map créés lorsque vous enregistrez des instances. Vous pouvez choisir entre le routage pondéré et le routage des réponses à valeurs multiples. Pour de plus amples informations, veuillez consulter Stratégie de routage.

   > **Note**
   >
   > Vous ne pouvez pas utiliser la console AWS Cloud Map pour configurer la création d'un enregistrement d'alias Route 53 lorsque vous enregistrez une instance. Si vous AWS Cloud Map souhaitez créer des enregistrements d'alias pour un équilibreur de charge Elastic Load Balancing lorsque vous enregistrez des instances par programmation, choisissez Weighted routing for Routing policy.

2. Pour Type d'enregistrement, choisissez le type d'enregistrement DNS qui détermine ce que Route 53 renvoie en réponse aux requêtes DNS par AWS Cloud Map. Pour de plus amples informations, veuillez consulter Type de registre.
3. Pour le TTL, spécifiez une valeur numérique pour définir la valeur du temps de vie (TTL), en secondes, au niveau du service. La valeur de TTL détermine la durée pendant laquelle les résolveurs DNS mettent en cache les informations pour cet enregistrement avant de transférer une autre requête DNS vers Amazon Route 53 pour obtenir des paramètres mis à jour.

8. Sous Configuration du bilan de santé, pour les options de contrôle de santé, choisissez le type de contrôle de santé applicable aux instances de service. Vous pouvez choisir de ne configurer aucun contrôle de santé, ou vous pouvez choisir entre un contrôle de santé Route 53 ou un contrôle de santé externe pour vos instances. Pour de plus amples informations, veuillez consulter AWS Cloud Map configuration du contrôle de santé du service.

> **Note**
>
> Les contrôles de santé de Route 53 sont configurables uniquement pour les services dans des espaces de noms DNS publics.

Si vous choisissez les bilans de santé Route 53, fournissez les informations suivantes.

1. Pour le seuil de défaillance, fournissez un nombre compris entre 1 et 10 qui définit le nombre de contrôles de santé consécutifs de la Route 53 qu'une instance de service doit réussir ou échouer pour que son état de santé change.
2. Pour le protocole de vérification de l'état, sélectionnez la méthode que Route 53 utilisera pour vérifier l'état des instances de service.
3. Si vous choisissez le protocole de contrôle de santé HTTP ou HTTPS, pour Health check path, indiquez le chemin que vous souhaitez qu'Amazon Route 53 demande lors de l'exécution des contrôles de santé. Le chemin peut être n'importe quelle valeur, telle que le fichier`/docs/route53-health-check.html`. Lorsque la ressource est saine, la valeur renvoyée est un code d'état HTTP au format 2xx ou 3xx. Vous pouvez également inclure des paramètres de chaîne de requête, par exemple, `/welcome.html?language=jp&login=y`. La console AWS Cloud Map ajoute automatiquement une barre oblique (/) au début.

Pour plus d'informations sur les bilans de santé de Route 53, consultez la section Comment Amazon Route 53 détermine si un bilan de santé est sain dans le manuel du développeur Amazon Route 53.

9. (Facultatif) Sous Balises, choisissez Ajouter des balises, puis spécifiez une clé et une valeur pour étiqueter votre espace de noms. Vous pouvez spécifier une ou plusieurs balises à ajouter à votre espace de noms. Les balises vous permettent de classer vos AWS ressources afin de les gérer plus facilement. Pour de plus amples informations, veuillez consulter Marquer vos ressources AWS Cloud Map.
10. Choisissez Créer un service.

AWS CLI

- Créez un service à l'aide de la `create-service` commande. Remplacez les *red* valeurs par les vôtres.

```
aws servicediscovery create-service \
    --name service-name \
    --namespace-id  ns-xxxxxxxxxxx \
    --dns-config "NamespaceId=ns-
xxxxxxxxxxx,RoutingPolicy=MULTIVALUE,DnsRecords=[{Type=A,TTL=60}]"
```

Sortie :

```
{
        "Service": {
        "Id": "srv-xxxxxxxxxxx",
        "Arn": "arn:aws:servicediscovery:us-west-2:123456789012:service/srv-
xxxxxxxxxxx",
        "Name": "service-name",
        "NamespaceId": "ns-xxxxxxxxxxx",
        "DnsConfig": {
            "NamespaceId": "ns-xxxxxxxxxxx",
            "RoutingPolicy": "MULTIVALUE",
            "DnsRecords": [
                {
                    "Type": "A",
                    "TTL": 60
                }
            ]
        },
        "CreateDate": 1587081768.334,
        "CreatorRequestId": "567c1193-6b00-4308-bd57-ad38a8822d25"
    }
}
```

AWS SDK for Python (Boto3)

Si ce n'est pas déjà `Boto3` fait, vous trouverez les instructions d'installation, de configuration et d'utilisation `Boto3` ici.

1. Importez `Boto3` et utilisez `servicediscovery` comme service.

```
import boto3
client = boto3.client('servicediscovery')
```

2. Créez un service avec`create_service()`. Remplacez les *red* valeurs par les vôtres. Pour plus d'informations, consultez create_service.

```
response = client.create_service(
    DnsConfig={
        'DnsRecords': [
            {
                'TTL': 60,
                'Type': 'A',
            },
        ],
        'NamespaceId': 'ns-xxxxxxxxxxx',
        'RoutingPolicy': 'MULTIVALUE',
    },
    Name='service-name',
    NamespaceId='ns-xxxxxxxxxxx',
)
```

Exemple de sortie de réponse

```
{
    'Service': {
        'Arn': 'arn:aws:servicediscovery:us-west-2:123456789012:service/srv-
xxxxxxxxxxx',
        'CreateDate': 1587081768.334,
        'DnsConfig': {
            'DnsRecords': [
                {
                    'TTL': 60,
                    'Type': 'A',
                },
            ],
            'NamespaceId': 'ns-xxxxxxxxxxx',
            'RoutingPolicy': 'MULTIVALUE',
        },
        'Id': 'srv-xxxxxxxxxxx',
        'Name': 'service-name',
        'NamespaceId': 'ns-xxxxxxxxxxx',
```

```
    },
    'ResponseMetadata': {
        '...': '...',
    },
}
```

Étapes suivantes

Après avoir créé un service, vous pouvez enregistrer les ressources de votre application en tant qu'instances de service contenant des informations sur la manière dont votre application peut localiser la ressource. Pour plus d'informations sur l'enregistrement des instances de AWS Cloud Map service, consultezEnregistrement d'une ressource en tant qu'instance AWS Cloud Map de service.

Vous pouvez également spécifier des métadonnées personnalisées telles que le poids des points de terminaison, les délais d'expiration des API et les politiques de nouvelle tentative en tant qu'attributs de service après avoir créé un service. Pour plus d'informations, veuillez consulter les sections ServiceAttributes et `UpdateServiceAttributes` (français non garanti) de la Référence d'API AWS Cloud Map .

Mise à jour d'un AWS Cloud Map service

En fonction de la configuration d'un service, vous pouvez mettre à jour ses balises, le seuil d'échec du contrôle de santé Route 53 et le temps de vie (TTL) pour les résolveurs DNS. Pour mettre à jour un service, effectuez la procédure suivante.

> **Note**
>
> Vous ne pouvez pas mettre à jour les paramètres des services associés aux espaces de noms HTTP.

AWS Management Console

1. Connectez-vous à la AWS Cloud Map console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudmap/.
2. Dans le panneau de navigation, choisissez Namespaces (Espaces de noms).
3. Sur la page Espaces de noms, choisissez l'espace de noms dans lequel le service est créé.

4. Sur la *namespace-name* page Namespace :, sélectionnez le service que vous souhaitez modifier et choisissez Afficher les détails.
5. Sur la *service-name* page Service :, choisissez Modifier.

   > **Note**
   >
   > Vous ne pouvez pas utiliser le flux de travail du bouton Modifier pour modifier les valeurs des services qui autorisent uniquement les appels d'API pour la découverte d'instances. Vous pouvez toutefois ajouter ou supprimer des balises sur la *service-name* page Service :.

6. Sur la page Modifier le service, sous Description du service, vous pouvez mettre à jour toute description précédemment définie pour le service ou ajouter une nouvelle description. Vous pouvez également ajouter des balises et mettre à jour le TTL pour les résolveurs DNS.
7. Dans le cadre de la configuration DNS, pour le TTL, vous pouvez spécifier une période de mise à jour, en secondes, qui détermine la durée pendant laquelle les résolveurs DNS mettent en cache les informations relatives à cet enregistrement avant qu'ils ne transmettent une autre requête DNS à Amazon Route 53 pour obtenir les paramètres mis à jour.
8. Si vous avez configuré les contrôles de santé Route 53, pour le seuil de défaillance, vous pouvez spécifier un nouveau nombre compris entre 1 et 10 qui définit le nombre de contrôles de santé consécutifs qu'une instance de service doit réussir ou échouer pour que son état de santé change.
9. Choisissez le service de mise à jour.

AWS CLI

- Mettez à jour un service à l'aide de la `update-service` commande (remplacez la *red* valeur par la vôtre).

```
aws servicediscovery update-service \
    --id  srv-xxxxxxxxxxx \
    --service "Description=new
 description,DnsConfig={DnsRecords=[{Type=A,TTL=60}]}"
```

Sortie :

```
{
```

```
    "OperationId": "l3pfx7f4ynndrbj3cfq5fm2qy2z37bms-5m6iaoty"
}
```

AWS SDK for Python (Boto3)

1. Si ce n'est pas déjà `Boto3` fait, vous trouverez les instructions d'installation, de configuration et d'utilisation `Boto3` ici.
2. Importez `Boto3` et utilisez `servicediscovery` en tant que service.

```
import boto3
client = boto3.client('servicediscovery')
```

3. Mettez à jour un service avec `update_service()` (remplacez la *red* valeur par la vôtre).

```
response = client.update_service(
    Id='srv-xxxxxxxxxxx',
    Service={
        'DnsConfig': {
            'DnsRecords': [
                {
                    'TTL': 300,
                    'Type': 'A',
                },
            ],
        },
        'Description': "new description",
    }
)
```

Exemple de sortie de réponse

```
{
    "OperationId": "l3pfx7f4ynndrbj3cfq5fm2qy2z37bms-5m6iaoty"
}
```

Répertorier AWS Cloud Map les services dans un espace de noms

Pour afficher une liste de services que vous avez créés dans un espace de noms, utilisez la procédure suivante.

AWS Management Console

1. Connectez-vous à la AWS Cloud Map console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudmap/.
2. Dans le panneau de navigation, choisissez Namespaces (Espaces de noms).
3. Choisissez le nom de domaine de l'espace de noms qui contient les services que vous souhaitez répertorier. Vous pouvez consulter la liste de tous les services sous Services et saisir le nom ou l'ID du service dans le champ de recherche pour trouver un service spécifique. Vous pouvez identifier Compte AWS celui qui a créé le service en utilisant le champ Créé par et le compte propriétaire du service en utilisant le champ Propriétaire de la ressource.

> **Note**
>
> Si l'espace de noms est un espace de noms partagé, l' Compte AWS ID sous Propriétaire de la ressource est le compte qui a créé et partagé l'espace de noms. L'ID de compte sous Created by peut être différent de l'ID sous Resource owner si un utilisateur d'espace de noms a créé le service. Le compte n'est IDs peut-être pas le même que votre identifiant de compte. Pour plus d'informations sur les espaces de noms partagés, consultezEspaces de AWS Cloud Map noms partagés.

AWS CLI

- Répertoriez les services à l'aide de la `list-services` commande. La commande suivante répertorie tous les services d'un espace de noms en utilisant l'ID d'espace de noms comme filtre. Remplacez la valeur *red* par votre propre valeur.

```
aws servicediscovery list-services --filters
 Name=NAMESPACE_ID,Values=ns-1234567890abcdef,Condition=EQ
```

AWS SDK for Python (Boto3)

1. Si ce n'est pas déjà `Boto3` fait, vous trouverez les instructions d'installation, de configuration et d'utilisation `Boto3` ici.
2. Importez `Boto3` et utilisez `servicediscovery` en tant que service.

```
import boto3
client = boto3.client('servicediscovery')
```

3. Répertoriez les services avec`list_services()`.

```
response = client.list_services()
# If you want to see the response
print(response)
```

Exemple de sortie de réponse

```
{
    'Services': [
        {
            'Arn': 'arn:aws:servicediscovery:us-west-2:123456789012:service/srv-
xxxxxxxxxxxxxxx',
            'CreateDate': 1587081768.334,
            'DnsConfig': {
                'DnsRecords': [
                    {
                        'TTL': 60,
                        'Type': 'A',
                    },
                ],
                'RoutingPolicy': 'MULTIVALUE',
            },
            'Id': 'srv-xxxxxxxxxxxxxxx',
            'Name': 'myservice',
        },
    ],
    'ResponseMetadata': {
        '...': '...',
    },
}
```

Supprimer un AWS Cloud Map service

Avant de pouvoir supprimer un service, vous devez annuler l'enregistrement de toutes les instances de service qui ont été enregistrées à l'aide de ce service. Pour de plus amples informations, veuillez consulter Annulation de l'enregistrement d'une instance de service AWS Cloud Map.

Après avoir désenregistré toutes les instances enregistrées à l'aide du service, effectuez la procédure suivante pour supprimer le service.

AWS Management Console

1. Connectez-vous à la AWS Cloud Map console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudmap/.
2. Dans le panneau de navigation, choisissez Namespaces (Espaces de noms).
3. Choisissez l'option pour l'espace de noms qui contient le service que vous souhaitez supprimer.
4. Sur la *namespace-name* page Namespace :, choisissez l'option correspondant au service que vous souhaitez supprimer.
5. Sélectionnez Delete (Supprimer).
6. Confirmez que vous voulez supprimer le service.

AWS CLI

- Supprimez un service à l'aide de la `delete-service` commande (remplacez la *red* valeur par la vôtre).

```
aws servicediscovery delete-service --id srv-xxxxxx
```

AWS SDK for Python (Boto3)

1. Si ce n'est pas déjà `Boto3` fait, vous trouverez les instructions d'installation, de configuration et d'utilisation `Boto3` ici.
2. Importez `Boto3` et utilisez `servicediscovery` en tant que service.

```
import boto3
client = boto3.client('servicediscovery')
```

3. Supprimez un service par `delete_service()` (remplacez la *red* valeur par la vôtre).

```
response = client.delete_service(
    Id='srv-xxxxxx',
)
# If you want to see the response
print(response)
```

Exemple de sortie de réponse

```
{
    'ResponseMetadata': {
        '...': '...',
    },
}
```

AWS Cloud Map instances de service

Une instance de service contient des informations sur comment rechercher une ressource, comme un serveur web ou une application. Après avoir enregistré des instances, vous les localisez à l'aide de requêtes DNS ou de l'action AWS Cloud Map DiscoverInstancesAPI. Les ressources que vous pouvez enregistrer incluent, sans toutefois s'y limiter, les suivantes :

- Instances Amazon EC2
- Tables Amazon DynamoDB
- Compartiments Amazon S3
- Files d’attente Amazon Simple Queue Service (Amazon SQS)
- APIs déployé au-dessus d'Amazon API Gateway

Vous pouvez spécifier des valeurs d'attribut pour les instances de services, et les clients peuvent utiliser ces attributs pour filtrer les ressources AWS Cloud Map renvoyées. Par exemple, une application peut demander des ressources dans une étape de déploiement particulière, comme BETA ou PROD. Vous pouvez également utiliser des attributs pour le versionnement.

Les procédures suivantes décrivent comment enregistrer des ressources dans votre application en tant qu'instances de service, afficher la liste des instances enregistrées dans un service, modifier certains paramètres d'instance et annuler l'enregistrement d'une instance.

Rubriques

Enregistrement d'une ressource en tant qu'instance AWS Cloud Map de service

Vous pouvez enregistrer les ressources de votre application en tant qu'instances dans un AWS Cloud Map service. Supposons, par exemple, que vous ayez créé un service appelé `users` pour toutes les

ressources de l'application qui gèrent les données utilisateur. Vous pouvez ensuite enregistrer une table DynamoDB utilisée pour stocker les données utilisateur en tant qu'instance dans ce service.

> **Note**
>
> Les fonctionnalités suivantes ne sont pas disponibles sur la AWS Cloud Map console :
>
> - Lorsque vous enregistrez une instance de service à l'aide de la console, vous ne pouvez pas créer d'enregistrement d'alias qui achemine le trafic vers un équilibreur de charge Elastic Load Balancing (ELB). Lorsque vous enregistrez une instance, vous devez inclure l'attribut `AWS_ALIAS_DNS_NAME`. Pour plus d'informations, consultez RegisterInstance dans la Référence d'API AWS Cloud Map .
> - Si vous enregistrez une instance à l'aide d'un service qui comprend une vérification de l'état personnalisée, vous ne pouvez pas spécifier le statut initial de la vérification de l'état personnalisée. Par défaut, le statut initial de la vérification de l'état personnalisée est Healthy (Sain). Si vous souhaitez que le statut d'état de santé initial soit Unhealthy (Non sain), enregistrez l'instance par programmation et incluez l'attribut `AWS_INIT_HEALTH_STATUS`. Pour plus d'informations, consultez RegisterInstance dans la Référence d'API AWS Cloud Map .

Pour enregistrer une instance dans un service, procédez comme suit.

AWS Management Console

1. Connectez-vous à la AWS Cloud Map console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudmap/.
2. Dans le panneau de navigation, choisissez Namespaces (Espaces de noms).
3. Sur la page Namespaces (Espaces de noms), choisissez l'espace de noms qui contient le service à utiliser comme modèle pour enregistrer une instance de service.
4. Sur la *`namespace-name`* page Namespace :, choisissez le service que vous souhaitez utiliser.
5. Sur la *`service-name`* page Service :, choisissez Enregistrer une instance de service.
6. Sur la page Enregistrer une instance de service, choisissez un type d'instance. En fonction de la configuration de découverte des instances d'espace de noms, vous pouvez choisir de spécifier une adresse IP, un ID d'instance Amazon EC2 ou d'autres informations d'identification pour une ressource qui ne possède pas d'adresse IP.

> **Note**
>
> Vous pouvez choisir une instance EC2 uniquement dans les espaces de noms HTTP.

7. Pour l'ID de l'instance de service, fournissez un identifiant associé à l'instance de service.

> **Note**
>
> Si vous souhaitez mettre à jour une instance existante, fournissez l'identifiant associé à l'instance que vous souhaitez mettre à jour. Procédez ensuite aux étapes suivantes pour mettre à jour les valeurs et réenregistrer l'instance.

8. En fonction du type d'instance que vous avez choisi, effectuez les étapes suivantes.

> **Important**
>
> Vous ne pouvez pas utiliser le `AWS_` préfixe (sans distinction majuscules/minuscules) dans une clé lorsque vous spécifiez un attribut personnalisé.

Type d'instance	Étapes
Adresse IP	a. Sous Attributs standard, pour IPv4adresse, indiquez une IPv4 adresse, le cas échéant, à laquelle votre applicati on peut accéder à la ressource associée à cette instance de service. b. Pour IPv6 l'adresse, fournissez une adresse IPv6 IP, le cas échéant, à laquelle vos applicati ons peuvent accéder à

Type d’instance	Étapes
	la ressource associée à cette instance de service. c. Pour Port, spécifiez tout port que votre applicati on doit inclure pour accéder à la ressource associée à cette instance de service. Le port est requis lorsque le service inclut un enregistrement SRV ou un bilan de santé Amazon Route 53. d. (Facultatif) Sous Attributs personnalisés, spécifiez les paires clé-valeur que vous souhaitez associer à la ressource.
instance EC2	a. Pour l'ID d'instance EC2, sélectionnez l'ID de l'instance Amazon EC2 que vous souhaitez enregistrer en AWS Cloud Map tant qu'instan ce de service. b. (Facultatif) Sous Attributs personnalisés, spécifiez les paires clé-valeur que vous souhaitez associer à la ressource.

Type d’instance	Étapes	
Identifying information for another resource (Informations d'identification pour une autre ressource)	a. Sous Attributs standard, si la configuration du service inclut un enregistrement DNS CNAME, vous verrez un champ CNAME. Pour CNAME, spécifiez le nom de domaine que Route 53 doit renvoyer en réponse aux requêtes DNS (par exemple,`example.com`). b. Sous Attributs personnalisés, spécifiez toute information d'identification pour une ressource qui n'est pas une adresse IP ou un ID d'instance Amazon EC2 sous forme de paire clé-valeur. Par exemple, vous pouvez enregistrer une fonction Lambda en spécifiant une clé appelée `function` et en fournissant le nom de la fonction Lambda sous forme de valeur. Vous pouvez également spécifier une clé appelée `name` et fournir un nom que vous pouvez utiliser pour la	

Type d'instance	Étapes	
	découverte d'instances par programmation.	

9. Choisissez Register service instance (Enregistrer une instance de service).

AWS CLI

- Lorsque vous soumettez une `RegisterInstance` demande :
  - Pour chaque enregistrement DNS que vous définissez dans le service spécifié par`ServiceId`, un enregistrement est créé ou mis à jour dans la zone hébergée associée à l'espace de noms correspondant.
  - Si le service inclut`HealthCheckConfig`, un bilan de santé est créé en fonction des paramètres de la configuration du contrôle de santé.
  - Tous les bilans de santé sont associés à chacun des enregistrements nouveaux ou mis à jour.

  Enregistrez une instance de service avec la `register-instance` commande (remplacez les *red* valeurs par les vôtres).

```
aws servicediscovery register-instance \
    --service-id srv-xxxxxxxxx \
    --instance-id myservice-xx \
    --attributes=AWS_INSTANCE_IPV4=172.2.1.3,AWS_INSTANCE_PORT=808
```

AWS SDK for Python (Boto3)

1. Si ce n'est pas déjà `Boto3` fait, vous trouverez les instructions d'installation, de configuration et d'utilisation `Boto3` ici.
2. Importez `Boto3` et utilisez `servicediscovery` en tant que service.

```
import boto3
client = boto3.client('servicediscovery')
```

3. Lorsque vous soumettez une `RegisterInstance` demande :

- Pour chaque enregistrement DNS que vous définissez dans le service spécifié par`ServiceId`, un enregistrement est créé ou mis à jour dans la zone hébergée associée à l'espace de noms correspondant.
- Si le service inclut`HealthCheckConfig`, un bilan de santé est créé en fonction des paramètres de la configuration du contrôle de santé.
- Tous les bilans de santé sont associés à chacun des enregistrements nouveaux ou mis à jour.

Enregistrez une instance de service auprès de `register_instance()` (remplacez les *red* valeurs par les vôtres).

```
response = client.register_instance(
    Attributes={
        'AWS_INSTANCE_IPV4': '172.2.1.3',
        'AWS_INSTANCE_PORT': '808',
    },
    InstanceId='myservice-xx',
    ServiceId='srv-xxxxxxxxx',
)
# If you want to see the response
print(response)
```

Exemple de sortie de réponse

```
{
    'OperationId': '4yejorelbukcjzpnr6tlmrghsjwpngf4-k95yg2u7',
    'ResponseMetadata': {
        '...': '...',
    },
}
```

Lister les instances AWS Cloud Map de service

Pour afficher la liste des instances de service que vous avez enregistrées à l'aide d'un service, utilisez la procédure suivante.

AWS Management Console

1. Connectez-vous à la AWS Cloud Map console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudmap/.
2. Dans le panneau de navigation, choisissez Namespaces (Espaces de noms).
3. Choisissez le nom de l'espace de noms qui contient le service pour lequel vous souhaitez répertorier les instances de service.
4. Choisissez le nom du service que vous avez utilisé pour créer les instances de service. Vous verrez une liste d'instances sous Instances de service. Vous pouvez saisir l'ID de l'instance dans le champ de recherche pour répertorier une instance spécifique. Le champ Created by indique l'ID de l'instance Compte AWS qui a enregistré l'instance.

> **Note**
>
> Si l'espace de noms dans lequel l'instance est enregistrée est un espace de noms partagé, l' Compte AWS ID sous Created by n'est peut-être pas le même que l'ID de votre compte. Pour plus d'informations sur les espaces de noms partagés, consultezEspaces de AWS Cloud Map noms partagés.

AWS CLI

- Répertoriez les instances de service à l'aide de la `list-instances` commande (remplacez la *red* valeur par la vôtre).

```
aws servicediscovery list-instances --service-id srv-xxxxxxxxx
```

AWS SDK for Python (Boto3)

1. Si ce n'est pas déjà `Boto3` fait, vous trouverez les instructions d'installation, de configuration et d'utilisation `Boto3` ici.
2. Importez `Boto3` et utilisez `servicediscovery` en tant que service.

```
import boto3
client = boto3.client('servicediscovery')
```

3. Répertoriez les instances de service par `list_instances()` (remplacez la *red* valeur par la vôtre).

```
response = client.list_instances(
    ServiceId='srv-xxxxxxxxx',
)
# If you want to see the response
print(response)
```

Exemple de sortie de réponse

```
{
    'Instances': [
        {
            'Attributes': {
                'AWS_INSTANCE_IPV4': '172.2.1.3',
                'AWS_INSTANCE_PORT': '808',
            },
            'Id': 'i-xxxxxxxxxxxxxxxx',
        },
    ],
    'ResponseMetadata': {
        '...': '...',
    },
}
```

Mettre à jour une instance AWS Cloud Map de service

Vous pouvez mettre à jour les instances de service de deux façons, selon les valeurs que vous souhaitez mettre à jour :

- Mettre à jour des valeurs : si vous souhaitez mettre à jour l'une des valeurs que vous avez spécifiées pour une instance de service lorsque vous l'avez enregistrée, y compris les attributs personnalisés, vous devez réenregistrer l'instance de service et spécifier à nouveau toutes les valeurs. Suivez les étapes décrites ci-dessousEnregistrement d'une ressource en tant qu'instance AWS Cloud Map de service, en spécifiant l'ID d'instance de l'instance de service existante pour l'ID d'instance de service.

Vous pouvez également utiliser l'`RegisterInstance`API. Vous pouvez spécifier l'ID de l'instance et du service existants à l'aide des `ServiceId` paramètres `InstanceId` et spécifier à nouveau d'autres valeurs.

- Mettre à jour uniquement les attributs personnalisés : si vous souhaitez uniquement mettre à jour les attributs personnalisés d'une instance de service, vous n'avez pas besoin d'enregistrer l'instance à nouveau. Vous pouvez mettre à jour uniquement ces valeurs. Consultez Mise à jour des attributs personnalisés pour une instance de service.

Mise à jour des attributs personnalisés pour une instance de service

Pour mettre à jour uniquement les attributs personnalisés d'une instance de service

1. Connectez-vous à la AWS Cloud Map console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudmap/.
2. Dans le panneau de navigation, choisissez Namespaces (Espaces de noms).
3. Sur la page Namespaces (Espaces de noms), choisissez l'espace de noms qui contient le service que vous avez utilisé initialement pour enregistrer l'instance de service.
4. Sur la *`namespace-name`* page Namespace :, choisissez le service que vous avez utilisé pour enregistrer l'instance de service.
5. Sur la *`service-name`* page Service :, choisissez le nom de l'instance de service que vous souhaitez mettre à jour.
6. Dans la section Custom attributes (Attributs personnalisés) choisissez Edit (Modifier).
7. Sur la *`instance-name`* page Modifier une instance de service :, ajoutez, supprimez ou mettez à jour des attributs personnalisés. Vous pouvez mettre à jour les clés et les valeurs des attributs existants.
8. Choisissez Update service instance (Mettre à jour l'instance de service).

Annulation de l'enregistrement d'une instance de service AWS Cloud Map

Avant de pouvoir supprimer un service, vous devez annuler l'enregistrement de toutes les instances de service qui ont été enregistrées à l'aide de ce service.

Pour annuler l'enregistrement d'une instance de service, utilisez la procédure suivante.

AWS Management Console

1. Connectez-vous à la AWS Cloud Map console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudmap/.
2. Dans le panneau de navigation, choisissez Namespaces (Espaces de noms).
3. Choisissez l'option pour l'espace de noms qui contient l'instance de service dont vous souhaitez annuler l'enregistrement.
4. Sur la *namespace-name* page Namespace :, choisissez le service que vous avez utilisé pour enregistrer l'instance de service.
5. Sur la *service-name* page Service :, choisissez l'instance de service dont vous souhaitez annuler l'enregistrement.
6. Choisissez Deregister (Annuler l'enregistrement).
7. Confirmez que vous voulez annuler l'enregistrement de l'instance de service.

AWS CLI

- Désenregistrez une instance de service à l'aide de la `deregister-instance` commande (remplacez les *red* valeurs par les vôtres). Cette commande supprime les enregistrements DNS Amazon Route 53 et tous les contrôles de santé AWS Cloud Map créés pour l'instance spécifiée.

```
aws servicediscovery deregister-instance \
    --service-id srv-xxxxxxxxx \
    --instance-id myservice-53
```

AWS SDK for Python (Boto3)

1. Si ce n'est pas déjà `Boto3` fait, vous trouverez les instructions d'installation, de configuration et d'utilisation `Boto3` ici.
2. Importez `Boto3` et utilisez `servicediscovery` comme service.

```
import boto3
client = boto3.client('servicediscovery')
```

3. Désenregistrez une instance de service avec `deregister-instance()` (remplacez les *red* valeurs par les vôtres). Cette commande supprime les enregistrements DNS Amazon Route 53 et tous les contrôles de santé AWS Cloud Map créés pour l'instance spécifiée.

```
response = client.deregister_instance(
    InstanceId='myservice-53',
    ServiceId='srv-xxxxxxxxx',
)
# If you want to see the response
print(response)
```

Exemple de sortie de réponse

```
{
    'OperationId': '4yejorelbukcjzpnr6tlmrghsjwpngf4-k98rnaiq',
    'ResponseMetadata': {
        '...': '...',
    },
}
```

Sécurité dans AWS Cloud Map

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 cette notion par les termes sécurité du cloud et sécurité dans le cloud :

- 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 cadre des programmes de conformitéAWS. Pour en savoir plus sur les programmes de conformité qui s'appliquent à AWS Cloud Map, consultez la section AWS Services concernés par programme de conformité.
- Sécurité dans le cloud — Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris de la sensibilité de vos données, des exigences de votre entreprise, ainsi que de la législation et de la réglementation applicables.

La documentation suivante vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de son utilisation AWS Cloud Map. Les rubriques suivantes expliquent comment procéder à la configuration AWS Cloud Map pour atteindre 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 AWS Cloud Map ressources.

Rubriques

Identity and Access Management pour AWS Cloud Map

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. Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser AWS Cloud Map les ressources. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

Rubriques

Public ciblé

La façon dont vous utilisez Gestion des identités et des accès AWS (IAM) varie en fonction de votre rôle :

- Utilisateur du service : demandez des autorisations à votre administrateur si vous ne pouvez pas accéder aux fonctionnalités (voir Résolution des problèmes AWS Cloud Map d'identité et d'accès)
- Administrateur du service : déterminez l’accès des utilisateurs et soumettez les demandes d’autorisation (voir Comment AWS Cloud Map fonctionne avec IAM)
- Administrateur IAM : rédigez des politiques pour gérer l’accès (voir Exemples de politiques basées sur l'identité pour AWS Cloud Map)

Authentification par des identités

L'authentification est la façon dont vous vous connectez à AWS l'aide de vos informations d'identification. Vous devez être authentifié en tant qu'utilisateur IAM ou en assumant un rôle IAM. Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en tant qu'identité fédérée à l'aide d'informations d'identification provenant d'une source d'identité telle que AWS IAM Identity Center (IAM Identity Center), d'une

authentification unique ou d'informations d'identification. Google/Facebook Pour plus d'informations sur la connexion, consultez Connexion à votre Compte AWS dans le Guide de l'utilisateur Connexion à AWS .

Pour l'accès par programmation, AWS fournit un SDK et une CLI pour signer les demandes de manière cryptographique. Pour plus d'informations, consultez Signature AWS Version 4 pour les demandes d'API dans le Guide de l'utilisateur IAM.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une seule identité de connexion appelée utilisateur Compte AWS root qui dispose d'un accès complet à toutes Services AWS les ressources. Il est vivement déconseillé d'utiliser l'utilisateur racine pour vos tâches quotidiennes. Pour les tâches qui requièrent des informations d'identification de l'utilisateur racine, consultez Tâches qui requièrent les informations d'identification de l'utilisateur racine dans le Guide de l'utilisateur IAM.

Identité fédérée

Il est recommandé d'obliger les utilisateurs humains à utiliser la fédération avec un fournisseur d'identité pour accéder à Services AWS l'aide d'informations d'identification temporaires.

Une identité fédérée est un utilisateur provenant de l'annuaire de votre entreprise, de votre fournisseur d'identité Web ou Directory Service qui y accède à Services AWS l'aide d'informations d'identification provenant d'une source d'identité. Les identités fédérées assument des rôles qui fournissent des informations d'identification temporaires.

Pour une gestion des accès centralisée, nous vous recommandons d'utiliser AWS IAM Identity Center. Pour plus d'informations, consultez Qu'est-ce que IAM Identity Center ? dans le Guide de l'utilisateur AWS IAM Identity Center .

Utilisateurs et groupes IAM

Un utilisateur IAM est une identité qui dispose d'autorisations spécifiques pour une seule personne ou application. Nous vous recommandons d'utiliser ces informations d'identification temporaires au lieu des utilisateurs IAM avec des informations d'identification à long terme. Pour plus d'informations, voir Exiger des utilisateurs humains qu'ils utilisent la fédération avec un fournisseur d'identité pour accéder à AWS l'aide d'informations d'identification temporaires dans le guide de l'utilisateur IAM.

Les groupes IAM spécifient une collection d'utilisateurs IAM et permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Pour plus d'informations, consultez Cas d'utilisation pour les utilisateurs IAM dans le Guide de l'utilisateur IAM.

Rôles IAM

Un rôle IAM est une identité dotée d'autorisations spécifiques qui fournit des informations d'identification temporaires. Vous pouvez assumer un rôle en passant d'un rôle d'utilisateur à un rôle IAM (console) ou en appelant une opération d' AWS API AWS CLI ou d'API. Pour plus d'informations, consultez Méthodes pour endosser un rôle dans le Guide de l'utilisateur IAM.

Les rôles IAM sont utiles pour l'accès des utilisateurs fédérés, les autorisations temporaires des utilisateurs IAM, les accès intercompte, les accès entre services et les applications exécutées sur Amazon EC2. Pour plus d'informations, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.

Gestion de l'accès à l'aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique définit les autorisations lorsqu'elles sont associées à une identité ou à une ressource. AWS évalue ces politiques lorsqu'un directeur fait une demande. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations les documents de politique JSON, consultez Vue d'ensemble des politiques JSON dans le Guide de l'utilisateur IAM.

À l'aide de politiques, les administrateurs précisent qui a accès à quoi en définissant quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Un administrateur IAM crée des politiques IAM et les ajoute aux rôles, que les utilisateurs peuvent ensuite assumer. Les politiques IAM définissent les autorisations quelle que soit la méthode que vous utilisez pour exécuter l'opération.

Politiques basées sur l'identité

Les stratégies basées sur l'identité sont des documents de stratégie d'autorisations JSON que vous attachez à une identité (utilisateur, groupe ou rôle). Ces politiques contrôlent les actions que peuvent exécuter ces identités, 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.

Les politiques basées sur l'identité peuvent être des politiques intégrées (intégrées directement dans une seule identité) ou des politiques gérées (politiques autonomes associées à plusieurs identités).

Pour découvrir comment choisir entre des politiques gérées et en ligne, consultez Choix entre les politiques gérées et les politiques en ligne dans le Guide de l'utilisateur IAM.

Politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Les exemples incluent les politiques de confiance de rôle IAM et les stratégies de compartiment Amazon S3. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Vous devez spécifier un principal dans une politique basée sur les ressources.

Les politiques basées sur les ressources sont des politiques en ligne situées dans ce service. Vous ne pouvez pas utiliser les politiques AWS gérées par IAM dans une stratégie basée sur les ressources.

Autres types de politique

AWS prend en charge des types de politiques supplémentaires qui peuvent définir les autorisations maximales accordées par les types de politiques les plus courants :

- Limites d'autorisations : une limite des autorisations définit le nombre maximum d'autorisations qu'une politique basée sur l'identité peut accorder à une entité IAM. Pour plus d'informations, consultez Limites d'autorisations pour des entités IAM dans le Guide de l'utilisateur IAM.
- Politiques de contrôle des services (SCPs) — Spécifiez les autorisations maximales pour une organisation ou une unité organisationnelle dans AWS Organizations. Pour plus d'informations, consultez Politiques de contrôle de service dans le Guide de l'utilisateur AWS Organizations .
- Politiques de contrôle des ressources (RCPs) : définissez le maximum d'autorisations disponibles pour les ressources de vos comptes. Pour plus d'informations, voir Politiques de contrôle des ressources (RCPs) dans le guide de AWS Organizations l'utilisateur.
- Politiques de session : politiques avancées que vous passez en tant que paramètre lorsque vous créez par programmation une session temporaire pour un rôle ou un utilisateur fédéré. Pour plus d'informations, consultez Politiques de session dans le Guide de l'utilisateur IAM.

Plusieurs types de politique

Lorsque plusieurs types de politiques s'appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS déterminer s'il faut autoriser

une demande lorsque plusieurs types de politiques sont impliqués, consultez la section Logique d'évaluation des politiques dans le guide de l'utilisateur IAM.

Comment AWS Cloud Map fonctionne avec IAM

Avant d'utiliser IAM pour gérer l'accès à AWS Cloud Map, découvrez les fonctionnalités IAM disponibles. AWS Cloud Map

Fonctionnalité IAM	AWS Cloud Map soutien
Politiques basées sur l'identité	Oui
Politiques basées sur les ressources	Non
Actions de politique	Oui
Ressources de politique	Oui
Clés de condition de politique (spécifiques au service)	Oui
ACLs	Non
ABAC (étiquettes dans les politiques)	Oui
Informations d'identification temporaires	Oui
Transmission des sessions d'accès (FAS)	Oui
Rôles du service	Non
Rôles liés à un service	Oui

Pour obtenir une vue d'ensemble de la façon dont AWS Cloud Map les autres AWS services fonctionnent avec la plupart des fonctionnalités IAM, consultez la section AWS Services compatibles avec IAM dans le Guide de l'utilisateur IAM.

Politiques basées sur l'identité pour AWS Cloud Map

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 politiques basées sur l'identité pour AWS Cloud Map

Pour consulter des exemples de politiques AWS Cloud Map basées sur l'identité, consultez. Exemples de politiques basées sur l'identité pour AWS Cloud Map

Politiques basées sur les ressources au sein de AWS Cloud Map

Prend en charge les politiques basées sur les ressources : non

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Par exemple, les politiques de confiance de rôle IAM et les politiques de compartiment Amazon S3 sont des politiques basées sur les ressources. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l’accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d’actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Pour permettre un accès intercompte, vous pouvez spécifier un compte entier ou des entités IAM dans un autre compte en tant que principal dans une politique basée sur les ressources. Pour plus d’informations, consultez Accès intercompte aux ressources dans IAM dans le Guide de l’utilisateur IAM.

> **Note**
>
> Vous pouvez utiliser AWS Resource Access Manager (AWS RAM) pour partager un espace de AWS Cloud Map noms en toute sécurité. Une politique basée sur les ressources est appliquée à votre espace de noms par le service. AWS RAM Pour de plus amples informations, veuillez consulter Espaces de AWS Cloud Map noms partagés.

Actions politiques pour AWS Cloud Map

Prend en charge les actions de politique : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Action` d'une politique JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une politique. Intégration d'actions dans une politique afin d'accorder l'autorisation d'exécuter les opérations associées.

Pour consulter la liste des AWS Cloud Map actions, reportez-vous à la section Actions définies par AWS Cloud Map dans la référence d'autorisation de service.

Les actions de politique en AWS Cloud Map cours utilisent le préfixe suivant avant l'action :

```
servicediscovery
```

Pour indiquer plusieurs actions dans une seule déclaration, séparez-les par des virgules.

```
"Action": [
      "servicediscovery:action1",
      "servicediscovery:action2"
         ]
```

Pour consulter des exemples de politiques AWS Cloud Map basées sur l'identité, consultez. Exemples de politiques basées sur l'identité pour AWS Cloud Map

Ressources politiques pour AWS Cloud Map

Prend en charge les ressources de politique : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément de politique JSON `Resource` indique le ou les objets auxquels l'action s'applique. Il est recommandé de définir une ressource à l'aide de son Amazon Resource Name (ARN). Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

```
"Resource": "*"
```

Pour consulter la liste des types de AWS Cloud Map ressources et leurs caractéristiques ARNs, consultez la section Ressources définies par AWS Cloud Map dans la référence d'autorisation de service. Pour savoir grâce à quelles actions vous pouvez spécifier l'ARN de chaque ressource, consultez Actions définies par AWS Cloud Map.

Pour consulter des exemples de politiques AWS Cloud Map basées sur l'identité, consultez. Exemples de politiques basées sur l'identité pour AWS Cloud Map

Clés de conditions de politique pour AWS Cloud Map

Prend en charge les clés de condition de politique spécifiques au service : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Condition` indique à quel moment les instructions s'exécutent en fonction de critères définis. Vous pouvez créer des expressions conditionnelles qui utilisent des opérateurs de condition, tels que les signes égal ou inférieur à, pour faire correspondre la condition de la politique aux valeurs de la demande. Pour voir toutes les clés de condition AWS globales, voir les clés de contexte de condition AWS globales dans le guide de l'utilisateur IAM.

Pour consulter la liste des clés de AWS Cloud Map condition, reportez-vous à la section Clés de condition pour AWS Cloud Map la référence d'autorisation de service. Pour savoir avec quelles

actions et ressources vous pouvez utiliser une clé de condition, consultez la section Actions définies par AWS Cloud Map.

AWS Cloud Map prend en charge les clés de condition spécifiques au service suivantes que vous pouvez utiliser pour filtrer avec précision vos politiques IAM.

`servicediscovery:NamespaceArn`

Un filtre qui vous permet d'obtenir les objets en spécifiant l'Amazon Resource Name (ARN) de l'espace de noms connexe.

`servicediscovery:NamespaceName`

Filtre qui vous permet d'obtenir des objets en spécifiant le nom de l'espace de noms connexe.

`servicediscovery:ServiceArn`

Filtre qui vous permet d'obtenir des objets en spécifiant l'Amazon Resource Name (ARN) pour le service connexe.

`servicediscovery:ServiceName`

Filtre qui vous permet d'obtenir des objets en spécifiant le nom du service connexe.

`servicediscovery:ServiceCreatedByAccount`

Un filtre qui vous permet d'obtenir des objets en spécifiant l'ID du service Compte AWS qui a créé le service.

Pour consulter des exemples de politiques AWS Cloud Map basées sur l'identité, consultez. Exemples de politiques basées sur l'identité pour AWS Cloud Map

ACLs in AWS Cloud Map

Supports ACLs : Non

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLs sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

ABAC avec AWS Cloud Map

Prise en charge d’ABAC (balises dans les politiques) : Oui

Le contrôle d'accès par attributs (ABAC) est une stratégie d'autorisation qui définit les autorisations en fonction des attributs appelés balises. Vous pouvez associer des balises aux entités et aux AWS ressources IAM, puis concevoir des politiques ABAC pour autoriser les opérations lorsque la balise du principal correspond à la balise de la ressource.

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`.

Si un service prend en charge les trois clés de condition pour tous les types de ressources, alors la valeur pour ce service est Oui. Si un service prend en charge les trois clés de condition pour certains types de ressources uniquement, la valeur est Partielle.

Pour plus d'informations sur ABAC, consultez Définition d'autorisations avec l'autorisation 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 AWS Cloud Map

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.

Transférer les sessions d'accès pour AWS Cloud Map

Prend en charge les sessions d'accès direct (FAS) : oui

Les sessions d'accès direct (FAS) utilisent les autorisations du principal appelant et Service AWS, combinées Service AWS à la demande d'envoi de demandes aux services en aval. Pour plus de détails sur la politique relative à la transmission de demandes FAS, consultez la section Sessions de transmission d'accès.

Rôles de service pour AWS Cloud Map

Prend en charge les rôles de service : Non

Un rôle de service est un rôle IAM qu'un service endosse pour accomplir des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer un rôle de service à partir d'IAM. Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM.

> **Warning**
>
> La modification des autorisations d'un rôle de service peut altérer la fonctionnalité d' AWS Cloud Map . Modifiez les rôles de service uniquement lorsque AWS Cloud Map vous recevez des instructions à cet effet.

Rôles liés à un service pour AWS Cloud Map

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 la création ou la gestion des rôles liés à un service, consultez Services AWS qui fonctionnent avec IAM. Recherchez un service dans le tableau qui inclut un `Yes` dans la colonne Rôle lié à un service. Choisissez le lien Oui pour consulter la documentation du rôle lié à ce service.

Exemples de politiques basées sur l'identité pour AWS Cloud Map

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou modifier les ressources AWS Cloud Map . Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM.

Pour apprendre à créer une politique basée sur l'identité IAM à l'aide de ces exemples de documents de politique JSON, consultez Création de politiques IAM (console) dans le Guide de l'utilisateur IAM.

Pour plus de détails sur les actions et les types de ressources définis par AWS Cloud Map, y compris le format de ARNs pour chacun des types de ressources, voir Actions, ressources et clés de condition AWS Cloud Map dans la référence d'autorisation de service.

Rubriques

Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer AWS Cloud Map des ressources dans votre compte. Ces actions peuvent entraîner des frais pour votre Compte AWS. Lorsque vous créez ou modifiez des politiques basées sur l'identité, suivez ces instructions et recommandations :

- Commencez AWS par les politiques gérées et passez aux autorisations du moindre privilège : pour commencer à accorder des autorisations à vos utilisateurs et à vos charges de travail, utilisez les politiques AWS gérées qui accordent des autorisations pour de nombreux cas d'utilisation courants. Ils sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire davantage les autorisations en définissant des politiques gérées par les AWS clients spécifiques à vos cas d'utilisation. Pour plus d'informations, consultez politiques gérées par AWS ou politiques gérées par AWS pour les activités professionnelles dans le Guide de l'utilisateur IAM.
- Accordez les autorisations de moindre privilège : lorsque vous définissez des autorisations avec des politiques IAM, accordez uniquement les autorisations nécessaires à l'exécution d'une seule tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation d'IAM pour appliquer des autorisations, consultez politiques et autorisations dans IAM dans le Guide de l'utilisateur IAM.
- Utilisez des conditions dans les politiques IAM pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques afin de limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées par le biais d'un service spécifique Service

AWS, tel que CloudFormation. Pour plus d'informations, consultez Conditions pour éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.

- Utilisez l'Analyseur d'accès IAM pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : l'Analyseur d'accès IAM valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les bonnes pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez Validation de politiques avec IAM Access Analyzer dans le Guide de l'utilisateur IAM.
- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root, activez l'authentification MFA pour une sécurité accrue. Compte AWS Pour exiger la MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez Sécurisation de l'accès aux API avec MFA dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur les bonnes pratiques dans IAM, consultez Bonnes pratiques de sécurité dans IAM dans le Guide de l'utilisateur IAM.

Utilisation de la AWS Cloud Map console

Pour accéder à la AWS Cloud Map console, vous devez disposer d'un ensemble minimal d'autorisations. Ces autorisations doivent vous permettre de répertorier et d'afficher les détails AWS Cloud Map des ressources de votre Compte AWS. Si vous créez une politique basée sur l'identité qui est plus restrictive que l'ensemble minimum d'autorisations requis, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs ou rôles) tributaires de cette politique.

Il n'est pas nécessaire d'accorder des autorisations de console minimales aux utilisateurs qui appellent uniquement l'API AWS CLI ou l' AWS API. Autorisez plutôt l'accès à uniquement aux actions qui correspondent à l'opération d'API qu'ils tentent d'effectuer.

Pour garantir que les utilisateurs et les rôles peuvent toujours utiliser la AWS Cloud Map console, associez également la politique AWS Cloud Map `ConsoleAccess` ou la politique `ReadOnly` AWS gérée aux entités. Pour plus d'informations, consultez Ajout d'autorisations à un utilisateur dans le Guide de l'utilisateur IAM.

AWS Cloud Map exemple d'accès à la console

Pour accorder un accès complet à la AWS Cloud Map console, vous devez accorder les autorisations conformément à la politique d'autorisation suivante :

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "servicediscovery:*",
            "route53:GetHostedZone",
            "route53:ListHostedZonesByName",
            "route53:CreateHostedZone",
            "route53:DeleteHostedZone",
            "route53:ChangeResourceRecordSets",
            "route53:CreateHealthCheck",
            "route53:GetHealthCheck",
            "route53:DeleteHealthCheck",
            "route53:UpdateHealthCheck",
            "ec2:DescribeInstances",
            "ec2:DescribeVpcs",
            "ec2:DescribeRegions"
         ],
         "Resource":"*"
      }
   ]
}
```

Voici pourquoi les autorisations sont obligatoires :

`servicediscovery:*`

Permet d'effectuer toutes les AWS Cloud Map actions.

`route53:CreateHostedZone`, **`route53:GetHostedZone`**, **`route53:ListHostedZonesByName`**, **`route53:DeleteHostedZone`**

Permet de AWS Cloud Map gérer les zones hébergées lorsque vous créez et supprimez des espaces de noms DNS publics et privés.

`route53:CreateHealthCheck`, **`route53:GetHealthCheck`**, **`route53:DeleteHealthCheck`**, **`route53:UpdateHealthCheck`**

Nous pouvons AWS Cloud Map gérer les bilans de santé en incluant les bilans d'état d'Amazon Route 53 lorsque vous créez un service.

`ec2:DescribeVpcs` et **`ec2:DescribeRegions`**

Laissez AWS Cloud Map gérer les zones hébergées privées.

Permettre AWS Cloud Map aux utilisateurs de consulter leurs propres autorisations

Cet exemple montre comment créer une politique qui permet aux utilisateurs IAM d'afficher les politiques en ligne et gérées attachées à leur identité d'utilisateur. Cette politique inclut les autorisations permettant d'effectuer cette action sur la console ou par programmation à l'aide de l'API AWS CLI or AWS .

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
```

```
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Autoriser l'accès en lecture à toutes les AWS Cloud Map ressources

La stratégie d'autorisations suivante accorde à l'utilisateur un accès en lecture seule à toutes les ressources AWS Cloud Map :

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "servicediscovery:Get*",
            "servicediscovery:List*",
            "servicediscovery:DiscoverInstances"
         ],
         "Resource":"*"
      }
   ]
}
```

AWS Cloud Map exemple d'instance de service

L'exemple suivant montre une politique d'autorisation qui accorde à un utilisateur l'autorisation d'enregistrer, de désenregistrer et de découvrir des instances de service. Le `Sid`, ou ID de l'instruction, est facultatif :

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Sid" : "AllowInstancePermissions",
         "Effect": "Allow",
         "Action": [
            "servicediscovery:RegisterInstance",
            "servicediscovery:DeregisterInstance",
            "servicediscovery:DiscoverInstances",
            "servicediscovery:Get*",
            "servicediscovery:List*",
            "route53:GetHostedZone",
            "route53:ListHostedZonesByName",
            "route53:ChangeResourceRecordSets",
            "route53:CreateHealthCheck",
            "route53:GetHealthCheck",
            "route53:DeleteHealthCheck",
            "route53:UpdateHealthCheck",
            "ec2:DescribeInstances"
         ],
         "Resource": "*"
      }
   ]
}
```

La stratégie accorde des autorisations sur les actions qui sont requises pour enregistrer et gérer des instances de service. L'autorisation Route 53 est requise si vous utilisez des espaces de noms DNS publics ou privés, car elle AWS Cloud Map crée, met à jour et supprime les enregistrements Route 53 et vérifie l'état de santé lorsque vous enregistrez et désenregistrez des instances. Le caractère générique (*) `Resource` donne accès à toutes les AWS Cloud Map instances, aux enregistrements Route 53 et aux bilans de santé détenus par le AWS compte courant.

Créer un exemple AWS Cloud Map de service

Lorsque vous ajoutez une politique d'autorisation pour permettre à une identité IAM de créer un AWS Cloud Map service, vous devez spécifier l'Amazon Resource Name (ARN) de l' AWS Cloud Map espace de noms et du service dans le champ de ressource. L'ARN inclut la région, l'ID de compte et

l'ID d'espace de noms. Comme vous ne savez pas encore quel est l'identifiant du service, nous vous recommandons d'utiliser un caractère générique. Voici un exemple d'extrait de politique.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "servicediscovery:CreateService"
            ],
            "Resource": [
                "arn:aws:servicediscovery:us-east-1:111122223333:namespace/ns-
p32123EXAMPLE",
                "arn:aws:servicediscovery:us-east-1:111122223333:service/*"
            ]
        }
    ]
}
```

Exemple de création d' AWS Cloud Map espaces de noms

La politique d'autorisation suivante permet aux utilisateurs de créer tous les types d' AWS Cloud Map espaces de noms :

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "servicediscovery:CreateHttpNamespace",
            "servicediscovery:CreatePrivateDnsNamespace",
            "servicediscovery:CreatePublicDnsNamespace",
            "route53:CreateHostedZone",
            "route53:GetHostedZone",
```

```
            "route53:ListHostedZonesByName",
            "ec2:DescribeVpcs",
            "ec2:DescribeRegions"
         ],
         "Resource":"*"
      }
   ]
}
```

AWS politiques gérées pour AWS Cloud Map

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 politique 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 : AWSCloud MapDiscoverInstanceAccess

Vous pouvez attacher `AWSCloudMapDiscoverInstanceAccess` à vos entités IAM. Permet d'accéder à l'API AWS Cloud Map Discovery.

Pour voir les autorisations de cette stratégie, consultez AWSCloudMapDiscoverInstanceAccess dans le AWS Guide de référence des stratégies gérées par.

AWS politique gérée : AWSCloud MapReadOnlyAccess

Vous pouvez attacher `AWSCloudMapReadOnlyAccess` à vos entités IAM. Accorde un accès en lecture seule à toutes les AWS Cloud Map actions.

Pour voir les autorisations de cette stratégie, consultez AWSCloudMapReadOnlyAccess dans le AWS Guide de référence des stratégies gérées par.

AWS politique gérée : AWSCloud MapRegisterInstanceAccess

Vous pouvez attacher `AWSCloudMapRegisterInstanceAccess` à vos entités IAM. Accorde un accès en lecture seule aux espaces de noms et aux services et autorise l'enregistrement et le désenregistrement des instances de service.

Pour voir les autorisations de cette stratégie, consultez AWSCloudMapRegisterInstanceAccess dans le AWS Guide de référence des stratégies gérées par.

AWS politique gérée : AWSCloud MapFullAccess

Vous pouvez attacher `AWSCloudMapFullAccess` à vos entités IAM. Fournit un accès complet à toutes les AWS Cloud Map actions

Pour voir les autorisations de cette stratégie, consultez AWSCloudMapFullAccess dans le AWS Guide de référence des stratégies gérées par.

AWS Cloud Map mises à jour des politiques AWS gérées

Consultez les détails des mises à jour des politiques AWS gérées AWS Cloud Map depuis que ce service a commencé à suivre ces modifications. Pour recevoir des alertes automatiques concernant les modifications, abonnez-vous au flux RSS sur la page d'historique du AWS Cloud Map document.

Modifier	Description	Date
AWSCloudMapDiscoverInstanceAccess, AWSCloudMapRegisterInstanceAccess, AWSCloudMapReadOnlyAccess— Mises à jour des politiques existantes.	AWS Cloud Map a mis à jour ces politiques pour permettre l'accès aux nouvelles opérations de AWS Cloud Map `DiscoverInstanceRevision` l'API.	15 août 2023

AWS Cloud Map Référence des autorisations d'API

Lorsque vous configurez le contrôle d'accès et que vous rédigez une politique d'autorisation que vous pouvez associer à une identité IAM (politiques basées sur l'identité), vous pouvez utiliser la liste suivante comme référence. La liste inclut chaque action d' AWS Cloud Map API et les actions auxquelles vous devez accorder des autorisations d'accès. Vous spécifiez les actions dans le `Action` champ de la politique. Pour plus de détails sur la valeur de la ressource que vous devez spécifier dans le `Resource` champ ou dans la politique IAM, voir Actions, ressources et clés de condition AWS Cloud Map dans la référence d'autorisation de service.

Vous pouvez utiliser des clés de condition AWS Cloud Map spécifiques dans vos politiques IAM pour certaines opérations. Pour plus d'informations, consultez la section Clés de condition pour AWS Cloud Map la référence d'autorisation de service.

Pour spécifier une action, utilisez le préfixe `servicediscovery` suivi du nom de l'action d'API (par exemple, `servicediscovery:CreatePublicDnsNamespace` ou `route53:CreateHostedZone`).

Autorisations requises pour les AWS Cloud Map actions

CreateHttpNamespace

Autorisations requises (action d'API) :

- `servicediscovery:CreateHttpNamespace`

CreatePrivateDnsNamespace

Autorisations requises (action d'API) :

- `servicediscovery:CreatePrivateDnsNamespace`
- `route53:CreateHostedZone`
- `route53:GetHostedZone`
- `route53:ListHostedZonesByName`
- `ec2:DescribeVpcs`
- `ec2:DescribeRegions`

CreatePublicDnsNamespace

Autorisations requises (action d'API) :

- `servicediscovery:CreatePublicDnsNamespace`

- `route53:CreateHostedZone`
- `route53:GetHostedZone`
- `route53:ListHostedZonesByName`

CreateService

Autorisations requises (Action d'API) : `servicediscovery:CreateService`

DeleteNamespace

Autorisations requises (action d'API) :

- `servicediscovery:DeleteNamespace`

DeleteService

Autorisations requises (Action d'API) : `servicediscovery:DeleteService`

DeleteServiceAttributes

Autorisations requises (Action d'API) : `servicediscovery:DeleteServiceAttributes`

DeregisterInstance

Autorisations requises (action d'API) :

- `servicediscovery:DeregisterInstance`
- `route53:GetHealthCheck`
- `route53:DeleteHealthCheck`
- `route53:UpdateHealthCheck`

DiscoverInstances

Autorisations requises (Action d'API) : `servicediscovery:DiscoverInstances`

GetInstance

Autorisations requises (Action d'API) : `servicediscovery:GetInstance`

GetInstancesHealthStatus

Autorisations requises (Action d'API) : `servicediscovery:GetInstancesHealthStatus`

GetNamespace

Autorisations requises (Action d'API) : `servicediscovery:GetNamespace`

GetOperation

Autorisations requises (Action d'API) : `servicediscovery:GetOperation`

GetService

Autorisations requises (Action d'API) : `servicediscovery:GetService`

GetServiceAttributes

Autorisations requises (Action d'API) : `servicediscovery:GetServiceAttributes`

ListInstances

Autorisations requises (Action d'API) : `servicediscovery:ListInstances`

ListNamespaces

Autorisations requises (Action d'API) : `servicediscovery:ListNamespaces`

ListOperations

Autorisations requises (Action d'API) : `servicediscovery:ListOperations`

ListServices

Autorisations requises (Action d'API) : `servicediscovery:ListServices`

ListTagsForResource

Autorisations requises (Action d'API) : `servicediscovery:ListTagsForResource`

RegisterInstance

Autorisations requises (action d'API) :

- `servicediscovery:RegisterInstance`
- `route53:GetHealthCheck`
- `route53:CreateHealthCheck`
- `route53:UpdateHealthCheck`
- `ec2:DescribeInstances`

TagResource

Autorisations requises (Action d'API) : `servicediscovery:TagResource`

UntagResource

Autorisations requises (Action d'API) : `servicediscovery:UntagResource`

UpdateHttpNamespace

Autorisations requises (Action d'API) : `servicediscovery:UpdateHttpNamespace`

UpdateInstanceCustomHealthStatus

Autorisations requises (Action d'API) : `servicediscovery:UpdateInstanceCustomHealthStatus`

UpdatePrivateDnsNamespace

Autorisations requises (action d'API) :

- `servicediscovery:UpdatePrivateDnsNamespace`
- `route53:ChangeResourceRecordSets`

UpdatePublicDnsNamespace

Autorisations requises (action d'API) :

- `servicediscovery:UpdatePublicDnsNamespace`
- `route53:ChangeResourceRecordSets`

UpdateService

Autorisations requises (action d'API) :

- `servicediscovery:UpdateService`
- `route53:GetHealthCheck`
- `route53:CreateHealthCheck`
- `route53:DeleteHealthCheck`
- `route53:UpdateHealthCheck`

UpdateServiceAttributes

Autorisations requises (Action d'API) : `servicediscovery:UpdateServiceAttributes`

Résolution des problèmes AWS Cloud Map d'identité et d'accès

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec AWS Cloud Map IAM.

Rubriques

Je ne suis pas autorisé à effectuer une action dans AWS Cloud Map

Si vous recevez une erreur qui indique que vous n'êtes pas autorisé à effectuer une action, vos politiques doivent être mises à jour afin de vous permettre d'effectuer l'action.

L'exemple d'erreur suivant se produit quand l'utilisateur IAM `mateojackson` tente d'utiliser la console pour afficher des informations détaillées sur une ressource *`my-example-widget`* fictive, mais ne dispose pas des autorisations `servicediscovery:`*`GetWidget`* fictives.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 servicediscovery:GetWidget on resource: my-example-widget
```

Dans ce cas, la politique qui s'applique à l'utilisateur `mateojackson` doit être mise à jour pour autoriser l'accès à la ressource *`my-example-widget`* à l'aide de l'action `servicediscovery:`*`GetWidget`*.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez une erreur selon laquelle vous n'êtes pas autorisé à exécuter `iam:PassRole` l'action, vos stratégies doivent être mises à jour afin de vous permettre de transmettre un rôle à AWS Cloud Map.

Certains vous Services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, vous devez disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé `marymajor` essaie d'utiliser la console pour exécuter une action dans AWS Cloud Map. Toutefois, l'action nécessite que le service

ait des autorisations accordées par une fonction de service. Mary n'est pas autorisée à transmettre le rôle au service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dans ce cas, les politiques de Mary doivent être mises à jour pour lui permettre d'exécuter l'action `iam:PassRole`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je souhaite permettre à des personnes extérieures Compte AWS à moi d'accéder à mes AWS Cloud Map ressources

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez spécifier qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques basées sur les ressources ou les listes de contrôle d'accès (ACLs), vous pouvez utiliser ces politiques pour autoriser les utilisateurs à accéder à vos ressources.

Pour plus d'informations, consultez les éléments suivants :

- Pour savoir si ces fonctionnalités sont prises AWS Cloud Map en charge, consultezComment AWS Cloud Map fonctionne avec IAM.
- Pour savoir comment fournir l'accès à vos ressources sur celles Comptes AWS que vous possédez, consultez la section Fournir l'accès à un utilisateur IAM dans un autre utilisateur Compte AWS que vous possédez dans le Guide de l'utilisateur IAM.
- Pour savoir comment fournir l'accès à vos ressources à des tiers Comptes AWS, consultez la section Fournir un accès à des ressources Comptes AWS détenues par des tiers dans le guide de l'utilisateur IAM.
- Pour savoir comment fournir un accès par le biais de la fédération d'identité, consultez Fournir un accès à des utilisateurs authentifiés en externe (fédération d'identité) dans le Guide de l'utilisateur IAM.
- Pour en savoir plus sur la différence entre l'utilisation des rôles et des politiques basées sur les ressources pour l'accès intercompte, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.

Validation de conformité pour AWS Cloud Map

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.

Résilience dans AWS Cloud Map

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 de disponibilité à l'autre sans interruption. Les zones de disponibilité sont plus hautement disponibles, tolérantes aux pannes et évolutives que les infrastructures traditionnelles à un ou plusieurs centres de données.

AWS Cloud Map est avant tout un service mondial. Cependant, vous pouvez AWS Cloud Map créer des bilans de santé Route 53 qui vérifient l'état des ressources dans des régions spécifiques, tels que les instances Amazon EC2 et les équilibreurs de charge Elastic Load Balancing.

Pour plus d'informations sur AWS les régions et les zones de disponibilité, consultez la section Infrastructure AWS mondiale.

Sécurité de l'infrastructure dans AWS Cloud Map

En tant que service géré, AWS Cloud Map il 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 AWS Cloud Map 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.

Vous pouvez améliorer le niveau de sécurité de votre VPC en le configurant de manière AWS Cloud Map à utiliser un point de terminaison VPC d'interface. Pour de plus amples informations, veuillez consulter Accès AWS Cloud Map via un point de terminaison d'interface (AWS PrivateLink).

Accès AWS Cloud Map via un point de terminaison d'interface (AWS PrivateLink)

Vous pouvez l'utiliser AWS PrivateLink pour créer une connexion privée entre votre VPC et. AWS Cloud Map Vous pouvez y accéder AWS Cloud Map comme s'il 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 y accéder. AWS Cloud Map

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 le demandeur qui servent de point d'entrée pour le trafic destiné à AWS Cloud Map.

Pour plus d’informations, consultez Accès aux Services AWS via AWS PrivateLink dans le Guide AWS PrivateLink .

Considérations relatives à AWS Cloud Map

Avant de configurer un point de terminaison d'interface pour AWS Cloud Map, consultez les considérations du AWS PrivateLink guide.

Si votre Amazon VPC ne possède pas de passerelle Internet et que vos tâches utilisent le pilote de journal pour envoyer des informations de `awslogs` journal à CloudWatch Logs, vous devez créer

un point de terminaison VPC d'interface pour les journaux. CloudWatch Pour plus d'informations, consultez la section Utilisation CloudWatch des journaux avec les points de terminaison VPC d'interface dans le guide de l'utilisateur Amazon CloudWatch Logs.

Les points de terminaison VPC ne prennent pas en charge AWS les demandes interrégionales. Veillez à créer votre point de terminaison dans la même région que celle dans laquelle vous souhaitez envoyer vos appels d'API à AWS Cloud Map.

Les points de terminaison d'un VPC prennent uniquement en charge le DNS fourni par Amazon via Amazon Route 53. Si vous souhaitez utiliser votre propre DNS, vous pouvez utiliser le transfert DNS conditionnel. Pour plus d'informations, consultez la section Ensembles d'options DHCP dans le guide de l'utilisateur Amazon VPC.

Le groupe de sécurité attaché au point de terminaison du VPC doit autoriser les connexions entrantes sur le port 443 depuis le sous-réseau privé de l'Amazon VPC.

Créez un point de terminaison d'interface pour AWS Cloud Map

Vous pouvez créer un point de terminaison d'interface pour AWS Cloud Map utiliser la console Amazon VPC ou le AWS Command Line Interface ()AWS CLI. Pour plus d'informations, consultez Création d'un point de terminaison d'interface dans le Guide AWS PrivateLink .

Créez un point de terminaison d'interface pour AWS Cloud Map utiliser les noms de service suivants :

> **Note**
>
> `DiscoverInstances`L'API ne sera pas disponible sur ces deux points de terminaison.

```
com.amazonaws.region.servicediscovery
```

```
com.amazonaws.region.servicediscovery-fips
```

Créez un point de terminaison d'interface pour que le plan de AWS Cloud Map données accède à l'`DiscoverInstances`API en utilisant les noms de service suivants :

```
com.amazonaws.region.data-servicediscovery
```

```
com.amazonaws.region.data-servicediscovery-fips
```

> **Note**
>
> Vous devez désactiver l'injection de préfixe d'hôte lorsque vous appelez `DiscoverInstances` avec les noms DNS VPCE régionaux ou zonaux pour les points de terminaison du plan de données. Le AWS CLI et AWS SDKs ajoutez différents préfixes d'hôte au point de terminaison du service lorsque vous appelez chaque opération d'API, ce qui produit des URL non valides lorsque vous spécifiez un point de terminaison VPC.

Si vous activez le DNS privé pour le point de terminaison de l'interface, vous pouvez envoyer des demandes d'API à AWS Cloud Map l'aide de son nom DNS régional par défaut. Par exemple, `servicediscovery.us-east-1.amazonaws.com`.

La AWS PrivateLink connexion VPCE est prise en charge dans toutes les régions où elle AWS Cloud Map est prise en charge ; toutefois, le client doit vérifier quelles zones de disponibilité prennent en charge le VPCE avant de définir un point de terminaison. Pour savoir quelles zones de disponibilité sont prises en charge avec les points de terminaison VPC d'interface dans une région, utilisez la describe-vpc-endpoint-services commande ou utilisez le. AWS Management Console Par exemple, les commandes suivantes renvoient les zones de disponibilité dans lesquelles vous pouvez déployer des points de terminaison VPC d' AWS Cloud Map interface dans la région USA Est (Ohio) :

```
aws --region us-east-2 ec2 describe-vpc-endpoint-services --query 'ServiceDetails[?
ServiceName==`com.amazonaws.us-east-2.servicediscovery`].AvailabilityZones[]'
```

Surveillance AWS Cloud Map

La surveillance est essentielle pour assurer la fiabilité, la disponibilité et les performances de vos solutions AWS . 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. Toutefois, avant de commencer la surveillance, vous devez créer un plan de surveillance qui contient les réponses aux questions suivantes :

- Quels sont les objectifs de la surveillance ?
- Quelles sont les ressources à surveiller ?
- À quelle fréquence les ressources doivent-elles être surveillées ?
- Quels outils de surveillance utiliser ?
- Qui exécute les tâches de supervision ?
- Qui doit être informé en cas de problème ?

Rubriques

Enregistrez les appels AWS Cloud Map d'API en utilisant AWS CloudTrail

AWS Cloud Map est intégré à AWS CloudTrailun service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un Service AWS. CloudTrail capture tous les appels d'API AWS Cloud Map sous forme d'événements. Les appels capturés incluent des appels provenant de la AWS Cloud Map console et des appels de code vers les opérations de l' AWS Cloud Map API. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été faite AWS Cloud Map, 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 de vos 90 Compte AWS derniers jours, créez un magasin de données sur les événements de Trail ou CloudTrailLake.

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.

CloudTrail Stockages de données sur les événements du lac

CloudTrail Lake vous permet d'exécuter des requêtes SQL sur vos événements. CloudTrail Lake convertit les événements existants au format JSON basé sur les lignes au format Apache

ORC. ORC est un format de stockage en colonnes qui est optimisé pour une récupération rapide des données. Les événements sont agrégés dans des magasins de données d'événement. Ceux-ci constituent des collections immuables d'événements basées sur des critères que vous sélectionnez en appliquant des sélecteurs d'événements avancés. Les sélecteurs que vous appliquez à un magasin de données d'événement contrôlent les événements qui persistent et que vous pouvez interroger. Pour plus d'informations sur CloudTrail Lake, consultez la section Travailler avec AWS CloudTrail Lake dans le guide de AWS CloudTrail l'utilisateur.

CloudTrail Les stockages et requêtes de données sur les événements de Lake entraînent des coûts. Lorsque vous créez un magasin de données d'événement, vous choisissez l'option de tarification que vous voulez utiliser pour le magasin de données d'événement. L'option de tarification détermine le coût d'ingestion et de stockage des événements, ainsi que les périodes de conservation par défaut et maximale pour le magasin de données d'événement. Pour plus d'informations sur la CloudTrail tarification, consultez la section AWS CloudTrail Tarification.

AWS Cloud Map événements de données dans CloudTrail

Les événements de données fournissent des informations sur les opérations de ressource effectuées sur ou dans une ressource (par exemple, la découverte d'une instance enregistrée dans un espace de noms). Ils sont également connus sous le nom opérations de plans de données. Les événements de données sont souvent des activités dont le volume est élevé. Par défaut, CloudTrail n'enregistre pas les événements liés aux données. L'historique des CloudTrail événements n'enregistre pas les événements liés aux données.

Des frais supplémentaires s'appliquent pour les événements de données. Pour plus d'informations sur la CloudTrail tarification, consultez la section AWS CloudTrail Tarification.

Vous pouvez enregistrer les événements de données pour les types de AWS Cloud Map ressources à l'aide de la CloudTrail console ou AWS CLI des opérations de CloudTrail l'API. Pour plus d'informations sur la façon de journaliser les événements de données, consultez Journalisation des événements de données avec la AWS Management Console et Journalisation des événements de données avec l' AWS Command Line Interface dans le Guide de l'utilisateur AWS CloudTrail .

Le tableau suivant répertorie les types de AWS Cloud Map ressources pour lesquels vous pouvez enregistrer des événements de données. La colonne Type d'événement de données (console) indique la valeur à choisir dans la liste des types d'événements de données de la CloudTrail console. La colonne de valeur resources.type indique la **`resources.type`** valeur que vous devez spécifier lors de la configuration de sélecteurs d'événements avancés à l'aide du ou. AWS CLI CloudTrail APIs

La CloudTrail colonne Données APIs enregistrées indique les appels d'API enregistrés CloudTrail pour le type de ressource.

Type d'événement de données (console)	valeur resources.type	Données APIs enregistrées sur CloudTrail
AwsApiCall	`AWS::ServiceDiscovery::Namespace`	• DiscoverInstances • DiscoverInstancesRevision
AwsApiCall	`AWS::ServiceDiscovery::Service`	• DiscoverInstances • DiscoverInstancesRevision • GetServiceAttributes

Vous pouvez configurer des sélecteurs d'événements avancés pour filtrer les champs `eventName`, `readOnly` et `resources.ARN` afin de ne journaliser que les événements importants pour vous. Pour plus d'informations sur ces champs, consultez AdvancedFieldSelector dans la Référence d'API AWS CloudTrail .

L'exemple suivant montre comment configurer des sélecteurs d'événements avancés pour consigner tous les événements liés aux AWS Cloud Map données.

```
"AdvancedEventSelectors":
[
    {
        "Name": "Log all AWS Cloud Map data events",
        "FieldSelectors": [
            { "Field": "eventCategory", "Equals": ["Data"] },
            { "Field": "resources.type", "Equals":
 ["AWS::ServiceDiscovery::Namespace"] }
        ]
    }
]
```

AWS Cloud Map événements de gestion 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.

AWS Cloud Map enregistre toutes les opérations AWS Cloud Map du plan de contrôle en tant qu'événements de gestion. Pour obtenir la liste des opérations du plan de AWS Cloud Map contrôle auxquelles il est AWS Cloud Map possible de se connecter CloudTrail, consultez la référence de l'AWS Cloud Map API.

AWS Cloud Map exemples d'événements

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 événement CloudTrail de gestion illustrant l'`CreateHTTPNamespace`opération.


```
{
    "eventVersion": "1.09",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE:alejandro_rosalez",
        "arn": "arn:aws:sts::111122223333:assumed-role/users/alejandro_rosalez",
        "accountId": "111122223333",
        "accessKeyId": "AIDACKCEVSQ6C2EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROA123456789EXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/readonly-role",
                "accountId": "111122223333",
                "userName": "alejandro_rosalez"
            },
            "attributes": {
                "creationDate": "2024-03-19T16:15:37Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2024-03-19T19:23:13Z",
    "eventSource": "servicediscovery.amazonaws.com",
    "eventName": "CreateHttpNamespace",
    "awsRegion": "eu-west-3",
    "sourceIPAddress": "192.0.2.0",
```

```
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/537.36",
    "requestParameters": {
        "name": "example-namespace",
        "creatorRequestId": "eda8b524-ca14-4f68-a176-dc4dfd165c26",
        "tags": []
    },
    "responseElements": {
        "operationId": "7xm4i7ghhkaalma666nrg6itf2eylcbp-gwipo38o"
    },
    "requestID": "641274d0-dbbe-4e64-9b53-685769a086c7",
    "eventID": "4a1ab076-ef1b-4bcf-aa95-cec5fb64f2bd",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_128_GCM_SHA256",
        "clientProvidedHostHeader": "servicediscovery.eu-west-3.amazonaws.com"
    },
    "sessionCredentialFromConsole": "true"
}
```

L'exemple suivant montre un événement de CloudTrail données qui illustre l'`DiscoverInstances`opération.

```
{
            "eventVersion": "1.09",
            "userIdentity": {
                "type": "AssumedRole",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE:alejandro_rosalez",
                "arn": "arn:aws:sts::111122223333:assumed-role/role/Admin",
                "accountId": "111122223333",
                "accessKeyId": "AIDACKCEVSQ6C2EXAMPLE",
                "sessionContext": {
                    "sessionIssuer": {
                        "type": "Role",
                        "principalId": "AROA123456789EXAMPLE",
                        "arn": "arn:aws:iam::"111122223333":role/Admin",
                        "accountId": "111122223333",
                        "userName": "Admin"
```

```
                    },
                    "attributes": {
                        "creationDate": "2024-03-19T16:15:37Z",
                        "mfaAuthenticated": "false"
                    }
                }
            },
            "eventTime": "2024-03-19T21:19:12Z",
            "eventSource": "servicediscovery.amazonaws.com",
            "eventName": "DiscoverInstances",
            "awsRegion": "eu-west-3",
            "sourceIPAddress": "13.38.34.79",
            "userAgent": "Boto3/1.20.34 md/Botocore#1.34.60 ua/2.0 os/linux#6.5.0-1014-
aws md/arch#x86_64 lang/python#3.10.12 md/pyimpl#CPython cfg/retry-mode#legacy
 Botocore/1.34.60",
            "requestParameters": {
                "namespaceName": "example-namespace",
                "serviceName": "example-service",
                "queryParameters": {"example-key": "example-value"}
            },
            "responseElements": null,
            "requestID": "e5ee36f1-edb0-4814-a4ba-2e8c97621c79",
            "eventID": "503cedb6-9906-4ee5-83e0-a64dde27bab0",
            "readOnly": true,
            "resources": [
                {
                    "accountId": "111122223333",
                    "type": "AWS::ServiceDiscovery::Namespace",
                    "ARN": "arn:aws:servicediscovery:eu-west-3:111122223333:namespace/
ns-vh4nbmhEXAMPLE"
                },
                {
                    "accountId": "111122223333",
                    "type": "AWS::ServiceDiscovery::Service",
                    "ARN": "arn:aws:servicediscovery:eu-west-3:111122223333:service/
srv-h46op6ylEXAMPLE"
                }
            ],
            "eventType": "AwsApiCall",
            "managementEvent": false,
            "recipientAccountId": "111122223333",
            "eventCategory": "Data",
            "tlsDetails": {
                "tlsVersion": "TLSv1.3",
```

```
                "cipherSuite": "TLS_AES_128_GCM_SHA256",
                "clientProvidedHostHeader": "data-servicediscovery.eu-
west-3.amazonaws.com"
            },
            "sessionCredentialFromConsole": "true"
        }
```

Pour plus d'informations sur le contenu des CloudTrail enregistrements, voir le contenu des CloudTrail enregistrements dans le Guide de AWS CloudTrail l'utilisateur.

Marquer vos ressources AWS Cloud Map

Une étiquette est une étiquette que vous attribuez à une AWS ressource. Chaque balise est constituée d'une clé et d'une valeur facultative que vous définissez.

Les balises vous permettent de classer vos AWS ressources en fonction, par exemple, de leur objectif, de leur propriétaire ou de leur environnement. Lorsque vous avez de nombreuses ressources de même type, vous pouvez rapidement identifier une ressource spécifique en fonction des balises que vous lui avez attribuées. Par exemple, vous pouvez définir un ensemble de balises pour vos AWS Cloud Map services afin de suivre le propriétaire et le niveau de pile de chaque service. Nous vous recommandons de concevoir un ensemble cohérent de clés de balise pour chaque type de ressource.

Les balises ne sont pas automatiquement affectées à vos ressources. Une fois que vous avez ajouté une balise, vous pouvez modifier les clés et valeurs de balise ou supprimer les balises d'une ressource à tout moment. Si vous supprimez une ressource, ses balises sont également supprimées.

Les balises n'ont aucune signification sémantique AWS Cloud Map et sont interprétées strictement comme des chaînes de caractères. Vous pouvez définir la valeur d'une balise sur une chaîne vide, mais vous ne pouvez pas définir la valeur d'une balise sur null. Si vous ajoutez une balise ayant la même clé qu'une balise existante sur cette ressource, la nouvelle valeur remplace l'ancienne valeur.

Vous pouvez travailler avec des balises à l'aide de l'API AWS Management Console AWS CLI, du et de l' AWS Cloud Map API.

Si vous utilisez Gestion des identités et des accès AWS (IAM), vous pouvez contrôler quels utilisateurs de votre AWS compte sont autorisés à créer, modifier ou supprimer des tags.

Comment les ressources sont étiquetées

Vous pouvez étiqueter des AWS Cloud Map espaces de noms et des services nouveaux ou existants.

Si vous utilisez la AWS Cloud Map console, vous pouvez appliquer des balises aux nouvelles ressources lors de leur création ou aux ressources existantes à tout moment à l'aide de l'onglet Tags de la page de ressources correspondante.

Si vous utilisez l' AWS Cloud Map API, le ou un AWS SDK AWS CLI, vous pouvez appliquer des balises aux nouvelles ressources à l'aide du `tags` paramètre de l'action d'API correspondante ou

aux ressources existantes à l'aide de l'action d'TagResourceAPI. Pour de plus amples informations, veuillez consulter TagResource.

En outre, certaines actions de création de ressources vous permettent de spécifier des balises pour une ressource lors de la création de cette dernière. Si des balises ne peuvent pas être appliquées au cours de la création de ressources, le processus de création de ressources échoue. Cela garantit que les ressources que vous vouliez baliser lors de la création sont créées avec des balises spécifiées ou ne sont pas créées du tout. Si vous balisez des ressources au moment de la création, vous n'avez pas besoin d'exécuter de scripts de balisage personnalisés après la création des ressources.

Le tableau suivant décrit les AWS Cloud Map ressources qui peuvent être balisées et les ressources qui peuvent être balisées lors de leur création.

Support de balisage pour les ressources AWS Cloud Map

Ressource	Prend en charge les étiquettes	Prend en charge la propagation des étiquettes	Supporte le balisage lors de la création (AWS Cloud Map API AWS CLI, AWS SDK)
AWS Cloud Map espaces de noms	Oui	Non Les balises d'espace de noms ne se propagent à aucune autre ressource associée à l'espace de noms.	Oui
AWS Cloud Map services	Oui	Non Les balises de service ne se propagent à aucune autre ressource associée au service.	Oui

Restrictions

Les restrictions de base suivantes s’appliquent aux balises :

- Nombre maximum de balises pour chaque ressource : 50

- Pour chaque ressource, chaque clé de balise doit être unique, et chaque clé de balise peut avoir une seule valeur.
- Longueur de clé maximale : 128 caractères Unicode en UTF-8
- Longueur de valeur maximale : 256 caractères Unicode en UTF-8
- Si votre schéma de balisage est utilisé pour plusieurs AWS services et ressources, n'oubliez pas que d'autres services peuvent être soumis à des restrictions quant aux caractères autorisés. Les caractères généralement autorisés sont les lettres, les chiffres et les espaces représentables en UTF-8, ainsi que les caractères suivants : + - = . _ : / @.
- Les clés et valeurs de balise sont sensibles à la casse.
- N'utilisez pas `aws:AWS:`, ni aucune combinaison majuscules ou minuscules, comme un préfixe pour les clés ou les valeurs, car il est réservé à l'usage. AWS Vous ne pouvez pas modifier ni supprimer des clés ou valeurs d'étiquette ayant ce préfixe. Les balises comportant ce préfixe ne sont pas prises en compte dans votre tags-per-resource limite.

Mise à jour des balises pour les AWS Cloud Map ressources

Utilisez les AWS CLI commandes ou opérations d' AWS Cloud Map API suivantes pour ajouter, mettre à jour, répertorier et supprimer les balises de vos ressources.

Support de balisage pour les ressources AWS Cloud Map

Tâche	Action d'API	AWS CLI	AWS Tools for Windows PowerShell
Ajouter ou remplacer une ou plusieurs étiquettes.	TagResource	tag-resource	Ajouter une SDResource étiquette
Supprimer une ou plusieurs étiquettes.	UntagResource	untag-resource	Supprimer- SDResource Tag
Répertorie les balises d'une ressource.	ListTagsForResource	list-tags-for-resource	Obtenir le SDResource tag

Les exemples suivants montrent comment ajouter ou supprimer les étiquettes d'une ressource à l'aide de l' AWS CLI.

Exemple 1 : Baliser une ressource existante

La commande suivante permet de baliser une ressource existante.

```
aws servicediscovery tag-resource --resource-arn resource_ARN --tags team=devs
```

Exemple 2 : Supprimer la balise d'une ressource existante

La commande suivante permet de supprimer une balise d'une ressource existante.

```
aws servicediscovery untag-resource --resource-arn resource_ARN --tag-keys tag_key
```

Exemple 3 : Afficher la liste des balises d'une ressource

La commande suivante permet de répertorier l'ensemble des étiquettes associées à une ressource existante.

```
aws servicediscovery list-tags-for-resource --resource-arn resource_ARN
```

Certaines actions de création de ressources vous permettent de spécifier des étiquettes lorsque vous créez la ressource. Les actions suivantes prennent en charge l'identification lors de la création.

Tâche	Action d'API	AWS CLI	AWS Tools for Windows PowerShell
Crée un espace de noms HTTP	CreateHttpNamespace	create-http-namespace	Nouveau- SDHttp Namespace
Créer un espace de noms privé basé sur DNS	CreatePrivateDnsNamespace	create-private-dns-namespace	Nouveau- SDPrivate DnsNamespace
Créer un espace de noms public basé sur DNS	CreatePublicDnsNamespace	create-public-dns-namespace	Nouveau- SDPublic DnsNamespace

Tâche	Action d'API	AWS CLI	AWS Tools for Windows PowerShell
Créer un service	CreateService	create-service	Nouveau- SDService

AWS Cloud Map quotas de service

AWS Cloud Map les ressources sont soumises aux quotas de service suivants au niveau du compte. Chaque quota répertorié s'applique à chaque AWS région dans laquelle vous créez AWS Cloud Map des ressources.

Nom	Par défaut	Ajusta	Description
Attributs personnalisés par instance	Chaque Région prise en charge : 30	Non	Le nombre maximum d'attributs personnalisés que vous pouvez spécifier lorsque vous enregistrez une instance.
DiscoverInstances taux de rafale des opérations par compte	Chaque Région prise en charge : 2 000	Oui	Le taux de rafale maximal pour appeler une DiscoverInstances opération à partir d'un seul compte.
DiscoverInstances opération par compte (taux stable)	Chaque Région prise en charge : 1 000	Oui	Le débit constant maximal pour effectuer des appels DiscoverI nstances à partir d'un seul compte.
DiscoverInstancesRevision taux d'opération par compte	Chaque région prise en charge : 3 000	Oui	Débit maximal pour appeler une DiscoverI nstancesRevision opération à partir d'un seul compte.
Instances par espace de noms	Chaque Région prise en charge : 2 000	Oui	Le nombre maximum d'instances de service que vous pouvez enregistrer à l'aide du même espace de noms.

Nom	Par défaut	Ajusta	Description
Instances par service	Chaque Région prise en charge : 1 000	Non	Le nombre maximum d'instances que vous pouvez enregistrer dans une région à l'aide du même service.
Espaces de noms par région	Chaque Région prise en charge : 50	Oui	Le nombre maximum d'espaces de noms que vous pouvez créer par région.

* Lorsque vous créez un espace de noms, nous créons automatiquement une zone hébergée Amazon Route 53. Cette zone hébergée est prise en compte dans le quota du nombre de zones hébergées que vous pouvez créer avec un AWS compte. Pour plus d'informations, consultez la section Quotas sur les zones hébergées dans le guide du développeur Amazon Route 53.

** L'augmentation du nombre d'instances pour les espaces de noms DNS AWS Cloud Map nécessite une augmentation de la limite Route 53 du nombre d'enregistrements par zone hébergée, ce qui entraîne des frais supplémentaires.

Gestion de vos quotas AWS Cloud Map de service

AWS Cloud Map est intégré à Service Quotas, un AWS service qui vous permet de consulter et de gérer vos quotas à partir d'un emplacement central. Pour plus d'informations, veuillez consulter la rubrique Qu'est-ce que Service Quotas ? dans le Guide de l'utilisateur Service Quotas.

Service Quotas permet de rechercher facilement la valeur de vos quotas de AWS Cloud Map service.

AWS Management Console

Pour consulter les quotas AWS Cloud Map de service à l'aide du AWS Management Console

1. Ouvrez la console Service Quotas sur https://console.aws.amazon.com/servicequotas/.
2. Dans le panneau de navigation, choisissez services AWS .
3. Dans la liste des services AWS , recherchez et sélectionnez AWS Cloud Map.

4. Dans la liste des quotas de service pour AWS Cloud Map, vous pouvez voir le nom du quota de service, la valeur appliquée (si elle est disponible), le quota AWS par défaut et si la valeur du quota est ajustable.

   Pour afficher des informations supplémentaires sur un quota de service, telles que la description, choisissez le nom du quota pour afficher les détails du quota.

5. (Facultatif) Pour demander une augmentation de quota, sélectionnez le quota que vous souhaitez augmenter et choisissez Demander une augmentation au niveau du compte.

Pour travailler davantage avec les quotas de service à l'aide du AWS Management Console guide de l'utilisateur sur les quotas de service.

AWS CLI

Pour consulter les quotas AWS Cloud Map de service à l'aide du AWS CLI

Exécutez la commande suivante pour afficher les AWS Cloud Map quotas par défaut.

```
aws service-quotas list-aws-default-service-quotas \
    --query 'Quotas[*].
{Adjustable:Adjustable,Name:QuotaName,Value:Value,Code:QuotaCode}' \
    --service-code AWSCloudMap \
    --output table
```

Exécutez la commande suivante pour afficher les AWS Cloud Map quotas que vous avez appliqués.

```
aws service-quotas list-service-quotas \
    --service-code AWSCloudMap
```

Pour plus d'informations sur l'utilisation des quotas de service à l'aide du AWS CLI, consultez le Guide de référence des AWS CLI commandes Service Quotas. Pour demander une augmentation de quota, consultez la commande `request-service-quota-increase` dans la référence des commandes AWS CLI.

Gérer la limitation des demandes d' AWS Cloud Map DiscoverInstances API

AWS Cloud Map limite les demandes DiscoverInstancesd'API pour chaque AWS compte par région. Le throttling contribue à améliorer les performances du service et à garantir une utilisation équitable pour tous les AWS Cloud Map clients. La régulation garantit que les appels à l' AWS Cloud Map DiscoverInstancesAPI ne dépassent pas les quotas de demandes d'DiscoverInstancesAPI maximaux autorisés. DiscoverInstances Les appels d'API provenant de l'une des sources suivantes sont soumis aux quotas de demandes :

- Une application tierce
- Un outil de ligne de commande
- La AWS Cloud Map console

Si vous dépassez le quota de limitation de l'API, le code `RequestLimitExceeded` d'erreur s'affiche. Pour de plus amples informations, veuillez consulter the section called “Limitation du débit de demande”.

Comment l'étranglement est appliqué

AWS Cloud Map utilise l'algorithme Token Bucket pour implémenter la régulation des API. Avec cet algorithme, votre compte dispose d'un compartiment contenant un nombre spécifique de jetons. Le nombre de jetons dans le compartiment représente votre quota de limitation à chaque seconde. Il existe un compartiment pour une seule région, qui s'applique à tous les points de terminaison de la région.

Limitation du débit de demande

Le throttling limite le nombre de demandes d'DiscoverInstancesAPI que vous pouvez effectuer. Chaque demande supprime un jeton du bucket. Par exemple, la taille du bucket pour l'opération d'DiscoverInstancesAPI est de 2 000 jetons, vous pouvez donc effectuer jusqu'à 2 000 DiscoverInstancesdemandes en une seconde. Si vous dépassez 2 000 demandes en une seconde, vous êtes limité et les demandes restantes au cours de cette seconde échouent.

Les seaux se rechargent automatiquement à un débit défini. Si le compartiment n'est pas à pleine capacité, un nombre défini de jetons est ajouté chaque seconde jusqu'à ce que le compartiment atteigne sa capacité maximale. Si le compartiment est plein à l'arrivée des jetons de recharge, ces

jetons sont jetés. La taille du bucket pour le fonctionnement de l'DiscoverInstancesAPI est de 2 000 jetons et le taux de recharge est de 1 000 jetons par seconde. Si vous effectuez 2 000 demandes d'DiscoverInstancesAPI par seconde, le bucket est immédiatement réduit à zéro (0) jeton. Le seau est ensuite rempli de 1 000 jetons par seconde jusqu'à ce qu'il atteigne sa capacité maximale de 2 000 jetons.

Vous pouvez utiliser des jetons au fur et à mesure qu'ils sont ajoutés au bucket. Il n'est pas nécessaire d'attendre que le compartiment atteigne sa capacité maximale avant de faire des demandes d'API. Si vous épuisez le compartiment en effectuant 2 000 demandes d'DiscoverInstancesAPI en une seconde, vous pouvez toujours effectuer jusqu'à 1 000 demandes d'DiscoverInstancesAPI par seconde aussi longtemps que nécessaire. Cela signifie que vous pouvez immédiatement utiliser les jetons de recharge lorsqu'ils sont ajoutés à votre bucket. Le bucket ne commence à se recharger à sa capacité maximale que lorsque vous faites moins de demandes d'API par seconde que le taux de recharge.

Nouvelles tentatives ou traitement par lots

Si une demande d'API échoue, il se peut que votre application doive réessayer la demande. Pour réduire le nombre de demandes d'API, utilisez un intervalle de sommeil approprié entre les demandes successives. Pour obtenir de meilleurs résultats, utilisez un intervalle de veille croissant ou variable.

Calcul de l’intervalle de veille

Lorsque vous devez interroger ou relancer une demande d’API, nous vous recommandons d’utiliser un algorithme d’interruption exponentielle pour calculer l’intervalle de sommeil entre les appels d’API. En utilisant des temps d'attente de plus en plus longs entre les tentatives pour des réponses d'erreur consécutives, vous pouvez réduire le nombre de demandes ayant échoué. Pour plus d'informations et des exemples d'implémentation de cet algorithme, voir Retry Behavior dans le guide de référence AWS SDKs and Tools.

Ajustement des quotas de limitation des API

Vous pouvez demander une augmentation des quotas de limitation des API pour votre AWS compte. Pour demander un ajustement de quota, contactezAWS Support Center.

Historique du document pour AWS Cloud Map

Le tableau suivant décrit les principales mises à jour et les nouvelles fonctionnalités du Guide du AWS Cloud Map développeur. Nous mettons aussi la documentation à jour régulièrement pour prendre en compte les commentaires qui nous sont envoyés.

Modification	Description	Date
AWS Cloud Map partage d'espaces de noms entre comptes	Vous pouvez désormais partager des espaces de noms avec d'autres organisations Comptes AWS ou au sein d'une organisation en AWS Organizations utilisant AWS Resource Access Manager (AWS RAM) pour simplifier la découverte et le registre des services entre comptes.	14 août 2025
AWS Cloud Map attributs de service	Vous pouvez désormais spécifier des attributs au niveau du service afin d'éviter de dupliquer les attributs entre les instances enregistrées auprès d'un service. Vous pouvez utiliser ces attributs pour le routage complexe du trafic, la définition des valeurs de délai d'expiration et de nouvelle tentative, ainsi que pour la coordination entre les services et les intégrations externes.	13 décembre 2024

Tutoriels ajoutés	Deux didacticiels présentant les cas d'utilisation courants de l' AWS Cloud Map ajout.	27 mars 2024
CloudTrail documentation d'intégration mise à jour	La documentation décrivant l' AWS Cloud Map intégration avec CloudTrail pour enregistrer l'activité de l'API a été mise à jour.	20 mars 2024
Mises à jour des politiques gérées	`AWSCloudMapDiscoverInstance Access` `AWSCloudMapRegisterInstance Access` , et les `AWSCloudMapReadOnlyAccess` politiques ont été mises à jour.	20 septembre 2023
Cloud Map et AWS PrivateLink	Vous pouvez désormais utiliser an AWS PrivateLink pour créer une connexion privée entre votre VPC et. AWS Cloud Map	15 septembre 2023
Mise à jour de la politique gérée	`AWSCloudMapDiscoverInstanceAccess` la politique a été mise à jour.	15 août 2023
AWS SDK pour Python	Ajout d'exemples de ligne de commande en Python.	13 septembre 2022
IPv6 soutien	Les points de terminaison d'API sont désormais disponibles `IPv6` uniquement dans les réseaux.	28 janvier 2022

Découverte d'instances de service	AWS Cloud Map ajout de la prise en charge de la création de services dans un espace de noms qui prend en charge les requêtes DNS détectables uniquement à l'aide de l'opération d'DiscoverInstancesAPI et non à l'aide de requêtes DNS.	24 mars 2021
Étiquette des ressources	AWS Cloud Map ajout de la prise en charge de l'ajout de balises de métadonnées à vos espaces de noms et services à l'aide du AWS Management Console.	8 février 2021
Étiquette des ressources	AWS Cloud Map ajout de la prise en charge de l'ajout de balises de métadonnées à vos espaces de noms et à vos services à l'aide du AWS CLI et APIs.	22 juin 2020
Publication initiale	Il s'agit de la première version du Guide du AWS Cloud Map développeur.	28 novembre 2018

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.