aws

Panduan Developer

AWS Cloud Map

AWS Cloud Map: Panduan Developer

Table of Contents

Apa itu AWS Cloud Map?

AWS Cloud Map adalah solusi terkelola sepenuhnya yang dapat Anda gunakan untuk memetakan nama logis ke layanan backend dan sumber daya yang bergantung pada aplikasi Anda. Ini juga membantu aplikasi Anda menemukan sumber daya menggunakan salah satu AWS SDKs, panggilan RESTful API, atau kueri DNS. AWS Cloud Map hanya melayani sumber daya yang sehat, yang dapat berupa tabel Amazon DynamoDB (DynamoDB), Amazon Simple Queue Service (Amazon SQS) antrian, setiap layanan aplikasi tingkat tinggi yang dibuat menggunakan EC2 instans Amazon Elastic Compute Cloud (Amazon) atau tugas Amazon Elastic Container Service (Amazon ECS), dan lebih.

Komponen AWS Cloud Map

Namespace

Untuk memulai, pertama-tama Anda membuat AWS Cloud Map namespace yang berfungsi sebagai cara untuk mengelompokkan layanan untuk aplikasi. Namespace mengidentifikasi nama yang ingin Anda gunakan untuk menemukan sumber daya Anda dan juga menentukan cara Anda ingin menemukan sumber daya: menggunakan panggilan AWS Cloud Map DiscoverInstancesAPI, kueri DNS di VPC, atau kueri DNS publik. Dalam kebanyakan kasus, namespace berisi semua layanan untuk aplikasi, seperti aplikasi penagihan. Untuk informasi selengkapnya, lihat AWS Cloud Map ruang nama.

Layanan

Setelah membuat namespace, Anda membuat AWS Cloud Map layanan untuk setiap jenis sumber daya yang ingin Anda gunakan AWS Cloud Map untuk menemukan titik akhir. Misalnya, Anda dapat membuat layanan untuk server web dan server database.

Layanan adalah template yang AWS Cloud Map digunakan ketika aplikasi Anda menambahkan sumber daya lain, seperti server web lain. Jika Anda memilih untuk menemukan sumber daya menggunakan DNS ketika Anda membuat namespace, layanan berisi informasi tentang jenis catatan yang ingin Anda gunakan untuk menemukan web server. Layanan juga menunjukkan apakah Anda ingin memeriksa kesehatan sumber daya dan apakah Anda ingin menggunakan pemeriksaan kesehatan Amazon Route 53 atau pemeriksa kesehatan pihak ketiga. Untuk informasi selengkapnya, lihat AWS Cloud Map layanan.

Contoh layanan

Ketika aplikasi Anda menambahkan sumber daya, Anda dapat memanggil tindakan AWS Cloud Map RegisterInstanceAPI dalam kode, yang membuat instance AWS Cloud Map layanan dalam layanan. Instance layanan berisi informasi tentang bagaimana aplikasi Anda dapat menemukan sumber daya, baik menggunakan DNS atau menggunakan tindakan AWS Cloud Map DiscoverInstancesAPI.

Ketika aplikasi Anda perlu terhubung ke sumber daya, ia memanggil DiscoverInstancesatau menggunakan kueri DNS publik atau pribadi dengan menentukan namespace dan layanan yang terkait dengan sumber daya. AWS Cloud Map mengembalikan informasi tentang cara menemukan satu atau lebih sumber daya. Jika Anda menentukan pemeriksaan kesehatan saat membuat layanan, hanya AWS Cloud Map mengembalikan instans yang sehat. Untuk informasi selengkapnya, lihat AWS Cloud Map contoh layanan.

Mengakses AWS Cloud Map

Anda dapat mengakses dengan AWS Cloud Map cara berikut:

- AWS Management Console— Prosedur di seluruh panduan ini menjelaskan cara menggunakan AWS Management Console untuk melakukan tugas.
- AWS SDKsJika Anda menggunakan bahasa pemrograman yang AWS menyediakan SDK, Anda dapat menggunakan SDK untuk mengakses. AWS Cloud Map SDKs menyederhanakan otentikasi, mengintegrasikan dengan mudah dengan lingkungan pengembangan Anda, dan menyediakan akses ke AWS Cloud Map perintah. Untuk informasi lebih lanjut, lihat Alat untuk Amazon Web Services.
- AWS Command Line Interface— Untuk informasi selengkapnya, lihat Memulai dengan AWS CLI di Panduan AWS Command Line Interface Pengguna.
- AWS Tools for Windows PowerShell— Untuk informasi selengkapnya, lihat Memulai dengan AWS Tools for Windows PowerShell di Panduan Alat AWS untuk PowerShell Pengguna.
- AWS Cloud Map API — Jika Anda menggunakan bahasa pemrograman yang tidak tersedia untuk SDK, lihat Referensi AWS Cloud Map API untuk informasi tentang tindakan API dan tentang cara membuat permintaan API.

Note

IPv6 Dukungan Klien - Mulai 22 Juni 2023 di semua wilayah baru, perintah apa pun yang dikirim AWS Cloud Map dari `IPv6` klien dialihkan ke titik akhir dualstack () baru. `servicediscovery.<region>.api.aws` AWS Cloud Map `IPv6`-hanya jaringan yang dapat dijangkau untuk titik akhir legacy (**`servicediscovery.<region>.amazonaws.com`**) dan dualstack di wilayah berikut yang dirilis sebelum 22 Juni 2023:

- US East (Ohio) – us-east-2
- US East (N. Virginia) – us-east-1
- US West (N. California) – us-west-1
- US West (Oregon) – us-west-2
- Afrika (Cape Town) — af-south-1
- Asia Pacific (Hong Kong) – ap-east-1
- Asia Pasifik (Hyderabad) — ap-south-2
- Asia Pasifik (Jakarta) — ap-southeast-3
- Asia Pasifik (Melbourne) — ap-southeast-4
- Asia Pacific (Mumbai) – ap-south-1
- Asia Pacific (Osaka) – ap-northeast-3
- Asia Pacific (Seoul) – ap-northeast-2
- Asia Pacific (Singapore) – ap-southeast-1
- Asia Pacific (Sydney) – ap-southeast-2
- Asia Pacific (Tokyo) – ap-northeast-1
- Canada (Central) – ca-central-1
- Europe (Frankfurt) – eu-central-1
- Europe (Ireland) – eu-west-1
- Europe (London) – eu-west-2
- Eropa (Milan) — eu-south-1
- Europe (Paris) – eu-west-3
- Eropa (Spanyol) - eu-south-2
- Europe (Stockholm) – (eu-north-1)

- Eropa (Zürich) — eu-central-2
- Timur Tengah (Bahrain) — me-south-1
- Timur Tengah (UEA) — me-central-1
- Amerika Selatan (Sao Paulo) – sa-east-1
- AWS GovCloud (AS-Timur) — -1 us-gov-east
- AWS GovCloud (AS-Barat) — -1 us-gov-west

AWS Identity and Access Management

AWS Cloud Map terintegrasi dengan AWS Identity and Access Management (IAM), layanan yang dapat digunakan organisasi Anda untuk melakukan tindakan berikut:

- Buat pengguna dan grup di bawah AWS akun organisasi Anda
- Bagikan sumber daya AWS akun Anda di antara pengguna di akun dengan cara yang efisien
- Menetapkan kredensial keamanan unik untuk setiap pengguna
- Secara bertahap mengontrol akses pengguna ke layanan dan sumber daya

Misalnya, Anda dapat menggunakan IAM AWS Cloud Map untuk mengontrol pengguna mana di AWS akun Anda yang dapat membuat namespace baru atau mendaftarkan instance.

Untuk informasi umum tentang IAM, lihat sumber daya berikut ini:

- Identity and Access Management untuk AWS Cloud Map
- AWS Identity and Access Management
- Panduan Pengguna IAM

AWS Cloud Map Harga

AWS Cloud Map harga didasarkan pada sumber daya yang Anda daftarkan di registri layanan dan panggilan API yang Anda buat untuk menemukannya. Dengan tidak AWS Cloud Map ada pembayaran di muka, dan Anda hanya membayar untuk apa yang Anda gunakan.

Opsional, Anda dapat mengaktifkan penemuan berbasis DNS untuk sumber daya dengan alamat IP. Anda juga dapat mengaktifkan pemeriksaan kondisi untuk sumber daya Anda menggunakan

pemeriksaan kondisi Amazon Route 53, apakah Anda menemukan instans menggunakan panggilan API atau kueri DNS. Anda akan dikenakan biaya tambahan terkait dengan Route 53 DNS dan penggunaan pemeriksaan kondisi.

Untuk informasi lebih lanjut, lihat AWS Cloud Map Harga.

AWS Cloud Map dan Kepatuhan AWS Cloud

Untuk informasi tentang AWS Cloud Map kepatuhan terhadap berbagai peraturan kepatuhan keamanan dan standar audit, lihat halaman berikut:

- AWS Kepatuhan Cloud
- AWS Layanan dalam Lingkup oleh Program Kepatuhan

Memulai dengan AWS Cloud Map

Panduan berikut menunjukkan cara mengatur untuk menggunakan AWS Cloud Map dan melakukan tugas umum menggunakan AWS Cloud Map ruang nama.

Ikhtisar panduan	Pelajari selengkapnya
Mendaftar AWS dan bersiap untuk menggunak an AWS Cloud Map	Siapkan untuk digunakan AWS Cloud Map
Menggunakan kueri DNS dan panggilan API untuk menemukan layanan backend.	Pelajari cara menggunakan penemuan AWS Cloud Map layanan dengan kueri DNS dan panggilan API
Menggunakan kueri DNS dan panggilan API untuk menemukan layanan backend menggunakan. AWS CLI	Pelajari cara menggunakan penemuan AWS Cloud Map layanan dengan kueri DNS dan panggilan API menggunakan AWS CLI
Membuat contoh aplikasi dan menggunakan atribut kustom dalam kode untuk menemukan sumber daya.	Pelajari cara menggunakan penemuan AWS Cloud Map layanan dengan atribut khusus
Membuat contoh aplikasi dan menggunakan atribut kustom dalam kode untuk menemukan sumber daya menggunakan AWS CLI.	Pelajari cara menggunakan penemuan AWS Cloud Map layanan dengan atribut kustom menggunakan AWS CLI

Siapkan untuk digunakan AWS Cloud Map

Gambaran umum dan prosedur di bagian berikut dimaksudkan untuk membantu Anda memulai AWS dan mempersiapkan Anda untuk mulai menggunakan AWS Cloud Map.

Topik

- Mendaftar untuk AWS
- Akses API, AWS CLIAWS Tools for Windows PowerShell,, atau AWS SDKs
- Mengatur AWS Command Line Interface atau AWS Tools for Windows PowerShell
- Unduh AWS SDK

Mendaftar untuk AWS

Mendaftar untuk Akun AWS

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar untuk Akun AWS

1. Buka https://portal.aws.amazon.com/billing/pendaftaran.
2. Ikuti petunjuk online.

   Bagian dari prosedur pendaftaran melibatkan menerima panggilan telepon atau pesan teks dan memasukkan kode verifikasi pada keypad telepon.

   Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWSdibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya di akun. Sebagai praktik keamanan terbaik, tetapkan akses administratif ke pengguna, dan gunakan hanya pengguna root untuk melakukan tugas yang memerlukan akses pengguna root.

AWS mengirimi Anda email konfirmasi setelah proses pendaftaran selesai. Kapan saja, Anda dapat melihat aktivitas akun Anda saat ini dan mengelola akun Anda dengan masuk https://aws.amazon.comke/ dan memilih Akun Saya.

Buat pengguna dengan akses administratif

Setelah Anda mendaftar Akun AWS, amankan Pengguna root akun AWS, aktifkan AWS IAM Identity Center, dan buat pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Amankan Anda Pengguna root akun AWS

1. Masuk ke AWS Management Consolesebagai pemilik akun dengan memilih pengguna Root dan memasukkan alamat Akun AWS email Anda. Di laman berikutnya, masukkan kata sandi.

   Untuk bantuan masuk dengan menggunakan pengguna root, lihat Masuk sebagai pengguna root di AWS Sign-In Panduan Pengguna.
2. Mengaktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

   Untuk petunjuk, lihat Mengaktifkan perangkat MFA virtual untuk pengguna Akun AWS root (konsol) Anda di Panduan Pengguna IAM.

Buat pengguna dengan akses administratif

1. Aktifkan Pusat Identitas IAM.

   Untuk mendapatkan petunjuk, silakan lihat Mengaktifkan AWS IAM Identity Center di Panduan Pengguna AWS IAM Identity Center .

2. Di Pusat Identitas IAM, berikan akses administratif ke pengguna.

   Untuk tutorial tentang menggunakan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, lihat Mengkonfigurasi akses pengguna dengan default Direktori Pusat Identitas IAM di Panduan AWS IAM Identity Center Pengguna.

Masuk sebagai pengguna dengan akses administratif

- Untuk masuk dengan pengguna Pusat Identitas IAM, gunakan URL masuk yang dikirim ke alamat email saat Anda membuat pengguna Pusat Identitas IAM.

  Untuk bantuan masuk menggunakan pengguna Pusat Identitas IAM, lihat Masuk ke portal AWS akses di Panduan AWS Sign-In Pengguna.

Tetapkan akses ke pengguna tambahan

1. Di Pusat Identitas IAM, buat set izin yang mengikuti praktik terbaik menerapkan izin hak istimewa paling sedikit.

   Untuk petunjuknya, lihat Membuat set izin di Panduan AWS IAM Identity Center Pengguna.

2. Tetapkan pengguna ke grup, lalu tetapkan akses masuk tunggal ke grup.

   Untuk petunjuk, lihat Menambahkan grup di Panduan AWS IAM Identity Center Pengguna.

Akses API, AWS CLIAWS Tools for Windows PowerShell,, atau AWS SDKs

Untuk menggunakan API, AWS CLI, AWS Tools for Windows PowerShell, atau AWS SDKs, Anda harus membuat kunci akses. Kunci ini terdiri dari ID kunci akses dan kunci akses rahasia, yang digunakan untuk menandatangani permintaan terprogram yang Anda buat. AWS

Pengguna membutuhkan akses terprogram jika mereka ingin berinteraksi dengan AWS luar. AWS Management Console Cara untuk memberikan akses terprogram tergantung pada jenis pengguna yang mengakses AWS.

Untuk memberi pengguna akses programatis, pilih salah satu opsi berikut.

Pengguna mana yang membutuhkan akses programatis?	Untuk	Oleh
Identitas tenaga kerja (Pengguna yang dikelola di Pusat Identitas IAM)	Gunakan kredensi sementara untuk menandatangani permintaan terprogram ke AWS CLI,, AWS SDKs atau. AWS APIs	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk AWS CLI, lihat Mengkonfigurasi yang akan AWS CLI digunakan AWS IAM Identity Center dalam Panduan AWS Command Line Interface Pengguna. • Untuk AWS SDKs, alat, dan AWS APIs, lihat Autentikasi Pusat Identitas IAM di Panduan Referensi Alat AWS SDKs dan Alat.
IAM	Gunakan kredensi sementara untuk menandatangani permintaan terprogram ke AWS CLI,, AWS SDKs atau. AWS APIs	Mengikuti petunjuk dalam Menggunakan kredensil sementara dengan AWS sumber daya di Panduan Pengguna IAM.
IAM	(Tidak direkomendasikan) Gunakan kredensi jangka panjang untuk menandatangani permintaan terprogram ke AWS CLI,, AWS SDKs atau. AWS APIs	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk mengetahui AWS CLI, lihat Mengautentikasi

Pengguna mana yang membutuhkan akses programatis?	Untuk	Oleh
		menggunakan kredensil pengguna IAM di Panduan Pengguna.AWS Command Line Interface • Untuk AWS SDKs dan alat, lihat Mengautentikasi menggunakan kredensil jangka panjang di Panduan Referensi Alat AWS SDKs dan Alat. • Untuk AWS APIs, lihat Mengelola kunci akses untuk pengguna IAM di Panduan Pengguna IAM.

Mengatur AWS Command Line Interface atau AWS Tools for Windows PowerShell

The AWS Command Line Interface (AWS CLI) adalah alat terpadu untuk mengelola AWS layanan. Untuk informasi tentang cara menginstal dan mengkonfigurasi AWS CLI, lihat Menginstal atau memperbarui ke versi terbaru dari Panduan AWS Command Line Interface Pengguna. AWS CLI

Jika Anda memiliki pengalaman dengan Windows PowerShell, Anda mungkin lebih suka menggunakannya AWS Tools for Windows PowerShell. Untuk informasi selengkapnya, lihat Menyiapkan AWS Tools for Windows PowerShell di Alat AWS untuk PowerShell Panduan Pengguna.

Unduh AWS SDK

Jika Anda menggunakan bahasa pemrograman yang AWS menyediakan SDK, sebaiknya gunakan SDK, bukan API. AWS Cloud Map Menggunakan SDK memiliki beberapa manfaat. SDKs membuat otentikasi lebih sederhana, mengintegrasikan dengan mudah dengan lingkungan pengembangan Anda, dan menyediakan akses ke AWS Cloud Map perintah. Untuk informasi lebih lanjut, lihat Alat untuk Layanan Web Amazon.

Pelajari cara menggunakan penemuan AWS Cloud Map layanan dengan kueri DNS dan panggilan API

Tutorial berikut mensimulasikan arsitektur microservice dengan dua layanan backend. Layanan pertama akan ditemukan menggunakan kueri DNS. Layanan kedua akan ditemukan hanya menggunakan AWS Cloud Map API.

> **Note**
>
> Rincian sumber daya, seperti nama domain dan alamat IP, hanya untuk tujuan simulasi. Mereka tidak dapat diselesaikan melalui internet.

Untuk end-to-end AWS CLI versi tutorial ini, lihatPelajari cara menggunakan penemuan AWS Cloud Map layanan dengan kueri DNS dan panggilan API menggunakan AWS CLI.

Prasyarat

Prasyarat berikut harus dipenuhi untuk menyelesaikan tutorial dengan sukses.

- Sebelum memulai, selesaikan langkah-langkah di Siapkan untuk digunakan AWS Cloud Map.
- Jika Anda belum menginstal AWS Command Line Interface, ikuti langkah-langkah di Menginstal atau memperbarui versi terbaru AWS CLI untuk menginstalnya.

  Tutorial ini membutuhkan terminal baris perintah atau shell untuk menjalankan perintah. Di Linux dan macOS, gunakan shell dan manajer paket pilihan Anda.

  > **Note**
  >
  > Di Windows, beberapa perintah Bash CLI yang biasa Anda gunakan dengan Lambda (`zip`seperti) tidak didukung oleh terminal bawaan sistem operasi. Untuk mendapatkan versi terintegrasi Windows dari Ubuntu dan Bash, instal Windows Subsystem untuk Linux.

- Tutorial ini membutuhkan lingkungan lokal dengan perintah utilitas pencarian `dig` DNS.

Langkah 1: Buat AWS Cloud Map namespace

Pada langkah ini, Anda membuat AWS Cloud Map namespace publik. AWS Cloud Map membuat zona yang dihosting Route 53 atas nama Anda dengan nama yang sama ini. Ini memberi Anda kemampuan untuk menemukan instance layanan yang dibuat di namespace ini baik menggunakan catatan DNS publik atau dengan menggunakan panggilan API. AWS Cloud Map

1. Masuk ke AWS Management Console dan buka AWS Cloud Map konsol di https://console.aws.amazon.com/cloudmap/.
2. Pilih Buat namespace.
3. Untuk nama Namespace, tentukan. `cloudmap-tutorial.com`

   > **Note**
   >
   > Jika Anda akan menggunakan ini dalam produksi, Anda ingin memastikan bahwa Anda menentukan nama domain yang Anda miliki atau memiliki akses ke. Tetapi untuk tujuan tuturial ini, tidak perlu menjadi domain aktual yang sedang digunakan.

4. (Opsional) Untuk deskripsi Namespace, tentukan deskripsi untuk tujuan Anda menggunakan namespace.
5. Untuk penemuan Instance, pilih panggilan API dan kueri DNS publik.
6. Tinggalkan sisa nilai default dan pilih Buat namespace.

Langkah 2: Buat AWS Cloud Map layanan

Pada langkah ini, Anda membuat dua layanan. Layanan pertama akan dapat ditemukan menggunakan DNS publik dan panggilan API. Layanan kedua akan ditemukan hanya menggunakan panggilan API.

1. Masuk ke AWS Management Console dan buka AWS Cloud Map konsol di https://console.aws.amazon.com/cloudmap/.
2. Di panel navigasi kiri, pilih Namespaces untuk mencantumkan ruang nama yang telah Anda buat.
3. Dari daftar ruang nama, pilih **`cloudmap-tutorial.com`** namespace dan pilih Lihat detail.
4. Di bagian Layanan, pilih Buat layanan dan lakukan hal berikut untuk membuat layanan pertama.

a. Untuk nama Layanan, masukkan`public-service`. Nama layanan akan diterapkan ke catatan DNS yang AWS Cloud Map dibuat. Format yang digunakan adalah*<service-name>*.*<namespace-name>*.

b. Untuk Konfigurasi Penemuan Layanan, pilih API dan DNS.

c. Di bagian konfigurasi DNS, untuk kebijakan Routing, pilih Multivalue answer routing.

> **Note**
>
> Konsol akan menerjemahkan ini ke MULTIVALUE setelah dipilih. Untuk informasi selengkapnya tentang opsi perutean yang tersedia, lihat Memilih kebijakan perutean di Panduan Pengembang Route 53.

d. Tinggalkan sisa nilai default dan pilih Buat layanan yang akan mengembalikan Anda ke halaman detail namespace.

5. Di bagian Layanan, pilih Buat layanan dan lakukan hal berikut untuk membuat layanan kedua.

a. Untuk nama Layanan, masukkan`backend-service`.

b. Untuk Konfigurasi Penemuan Layanan, pilih API saja.

c. Tinggalkan sisa nilai default dan pilih Buat layanan.

Langkah 3: Daftarkan instance AWS Cloud Map layanan

Pada langkah ini, Anda membuat dua instance layanan, satu untuk setiap layanan di namespace kami.

1. Masuk ke AWS Management Console dan buka AWS Cloud Map konsol di https://console.aws.amazon.com/cloudmap/.
2. Dari daftar ruang nama, pilih namespace yang Anda buat di langkah 1 dan pilih Lihat detail.
3. Pada halaman detail namespace, dari daftar layanan, pilih `public-service` layanan dan pilih Lihat detail.
4. Di bagian Service instance, pilih Register service instance dan lakukan hal berikut untuk membuat instance layanan pertama.

a. Untuk ID contoh Layanan, tentukan`first`.

b. Untuk IPv4 alamat, tentukan`192.168.2.1`.

c. Tinggalkan sisa nilai default dan pilih Register service instance.

5. Menggunakan breadcrumb di bagian atas halaman, pilih cloudmap-tutorial.com untuk menavigasi kembali ke halaman detail namespace.
6. Pada halaman detail namespace, dari daftar layanan, pilih layanan backend-service dan pilih Lihat detail.
7. Di bagian Service instance, pilih Register service instance dan lakukan hal berikut untuk membuat instance layanan kedua.

   a. Untuk ID contoh Layanan, tentukan `second` untuk menunjukkan bahwa ini adalah instance layanan kedua.

   b. Untuk jenis Instance, pilih Mengidentifikasi informasi untuk sumber daya lain.

   c. Untuk atribut Custom, tambahkan pasangan kunci-nilai dengan `service-name` sebagai kunci dan `backend` sebagai nilai.

   d. Pilih Daftarkan instans layanan.

Langkah 4: Temukan contoh AWS Cloud Map layanan

Sekarang setelah AWS Cloud Map namespace, layanan, dan instance layanan dibuat, Anda dapat memverifikasi semuanya berfungsi dengan menemukan instance. Gunakan `dig` perintah untuk memverifikasi pengaturan DNS publik dan AWS Cloud Map API untuk memverifikasi layanan backend. Untuk informasi selengkapnya tentang `dig` perintah, lihat dig - DNS lookup utility.

1. Masuk ke AWS Management Console dan buka konsol Route 53 di https://console.aws.amazon.com/route53/.
2. Pada navigasi di sebelah kiri, pilih Zona yang di-hosting.
3. Pilih zona yang dihosting cloudmap-tutorial.com. Ini menampilkan detail zona yang dihosting di panel terpisah. Perhatikan server Nama yang terkait dengan zona host Anda karena kami akan menggunakannya di langkah berikutnya.
4. Menggunakan perintah dig dan salah satu server nama Route 53 untuk zona host Anda, kueri catatan DNS untuk instance layanan Anda.

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

   Output `ANSWER SECTION` dalam harus menampilkan IPv4 alamat yang Anda kaitkan dengan `public-service` layanan Anda.

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

5. Menggunakan AWS CLI, kueri atribut untuk instance layanan kedua Anda.

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

Output menampilkan atribut yang Anda kaitkan dengan layanan sebagai pasangan kunci-nilai.

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

Langkah 5: Bersihkan sumber daya

Setelah Anda menyelesaikan tutorial, Anda dapat menghapus sumber daya. AWS Cloud Map mengharuskan Anda membersihkannya dalam urutan terbalik, instance layanan terlebih dahulu, lalu layanan, dan akhirnya namespace. AWS Cloud Map akan membersihkan sumber daya Route 53 atas nama Anda ketika Anda melalui langkah-langkah ini.

1. Masuk ke AWS Management Console dan buka AWS Cloud Map konsol di https://console.aws.amazon.com/cloudmap/.
2. Dari daftar ruang nama, pilih **`cloudmap-tutorial.com`** namespace dan pilih Lihat detail.
3. Pada halaman detail namespace, dari daftar layanan, pilih `public-service` layanan dan pilih Lihat detail.
4. Di bagian Service instance, pilih `first` instance dan pilih Deregister.

5. Menggunakan breadcrumb di bagian atas halaman, pilih cloudmap-tutorial.com untuk menavigasi kembali ke halaman detail namespace.
6. Pada halaman detail namespace, dari daftar layanan, pilih layanan layanan publik dan pilih Hapus.
7. Ulangi langkah 3-6 untuk. `backend-service`
8. Di navigasi kiri, pilih Namespaces.
9. Pilih **`cloudmap-tutorial.com`** namespace dan pilih Delete.

> **Note**
>
> Meskipun AWS Cloud Map membersihkan sumber daya Route 53 atas nama Anda, Anda dapat menavigasi ke konsol Route 53 untuk memverifikasi bahwa zona yang `cloudmap-tutorial.com` dihosting dihapus.

Pelajari cara menggunakan penemuan AWS Cloud Map layanan dengan kueri DNS dan panggilan API menggunakan AWS CLI

Tutorial ini menunjukkan bagaimana menggunakan penemuan AWS Cloud Map layanan menggunakan AWS Command Line Interface (CLI). Anda akan membuat arsitektur microservice dengan dua layanan backend — satu dapat ditemukan menggunakan kueri DNS dan satu lagi dapat ditemukan menggunakan API saja. AWS Cloud Map

Untuk tutorial yang menyertakan langkah-langkah AWS Cloud Map konsol, lihatPelajari cara menggunakan penemuan AWS Cloud Map layanan dengan kueri DNS dan panggilan API.

Prasyarat

Prasyarat berikut harus dipenuhi untuk menyelesaikan tutorial dengan sukses.

- Sebelum memulai, selesaikan langkah-langkah di Siapkan untuk digunakan AWS Cloud Map.
- Jika Anda belum menginstal AWS Command Line Interface, ikuti langkah-langkah di Menginstal atau memperbarui versi terbaru AWS CLI untuk menginstalnya.

  Tutorial ini membutuhkan terminal baris perintah atau shell untuk menjalankan perintah. Di Linux dan macOS, gunakan shell dan manajer paket pilihan Anda.

> **Note**
>
> Di Windows, beberapa perintah Bash CLI yang biasa Anda gunakan dengan Lambda (`zip`seperti) tidak didukung oleh terminal bawaan sistem operasi. Untuk mendapatkan versi terintegrasi Windows dari Ubuntu dan Bash, instal Windows Subsystem untuk Linux.

- Tutorial ini membutuhkan lingkungan lokal dengan perintah utilitas pencarian `dig` DNS.

Buat AWS Cloud Map namespace

Pertama, Anda akan membuat AWS Cloud Map namespace publik. AWS Cloud Map akan membuat zona yang dihosting Route 53 dengan nama yang sama, memungkinkan penemuan layanan melalui catatan DNS dan panggilan API.

1. Buat namespace DNS publik:

   ```
   aws servicediscovery create-public-dns-namespace \
       --name cloudmap-tutorial.com \
       --creator-request-id cloudmap-tutorial-request-1 \
       --region us-east-2
   ```

   Perintah mengembalikan ID operasi yang dapat Anda gunakan untuk memeriksa status pembuatan namespace:

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

2. Periksa status operasi untuk mengonfirmasi namespace berhasil dibuat:

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

3. Setelah operasi berhasil, dapatkan ID namespace:

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

```
    --output text
```

Perintah ini mengembalikan ID namespace, yang Anda perlukan untuk langkah-langkah selanjutnya:

```
ns-abcd1234xmplefgh
```

Buat AWS Cloud Map layanan

Sekarang, buat dua layanan dalam namespace Anda. Layanan pertama akan dapat ditemukan menggunakan panggilan DNS dan API, sedangkan yang kedua akan ditemukan hanya menggunakan panggilan API.

1. Buat layanan pertama dengan penemuan DNS diaktifkan:

```
aws servicediscovery create-service \
    --name public-service \
    --namespace-id ns-abcd1234xmplefgh \
    --dns-config "RoutingPolicy=MULTIVALUE,DnsRecords=[{Type=A,TTL=300}]" \
    --region us-east-2
```

Perintah mengembalikan rincian tentang layanan yang dibuat:

```
{
    "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. Buat layanan kedua dengan penemuan khusus API:

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

Perintah mengembalikan rincian tentang layanan yang dibuat:

```
{
    "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"
    }
}
```

Daftarkan instance AWS Cloud Map layanan

Selanjutnya, daftarkan instance layanan untuk setiap layanan Anda. Contoh ini mewakili sumber daya aktual yang akan ditemukan.

1. Daftarkan instance pertama dengan IPv4 alamat untuk penemuan DNS:

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

Perintah mengembalikan ID operasi:

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

2. Periksa status operasi untuk mengonfirmasi instans berhasil terdaftar:

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

3. Daftarkan instance kedua dengan atribut khusus untuk penemuan API:

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

Perintah mengembalikan ID operasi:

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

4. Periksa status operasi untuk mengonfirmasi instans berhasil terdaftar:

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

Temukan contoh AWS Cloud Map layanan

Sekarang setelah Anda membuat dan mendaftarkan instance layanan Anda, Anda dapat memverifikasi semuanya berfungsi dengan menemukannya menggunakan kueri DNS dan API. AWS Cloud Map

1. Pertama, dapatkan ID zona yang dihosting Route 53:

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

Ini mengembalikan ID zona yang dihosting:

```
/hostedzone/Z1234ABCDXMPLEFGH
```

2. Dapatkan server nama untuk zona host Anda:

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

Ini mengembalikan salah satu server nama:

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

3. Gunakan `dig` perintah untuk menanyakan catatan DNS untuk layanan publik Anda:

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

Output harus menampilkan IPv4 alamat yang Anda kaitkan dengan layanan Anda:

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

4. Gunakan AWS CLI untuk menemukan instance layanan backend:

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

Output menampilkan atribut yang Anda kaitkan dengan layanan:

```
{
```

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

Bersihkan sumber daya

Setelah Anda menyelesaikan tutorial, bersihkan sumber daya untuk menghindari biaya yang dikenakan. AWS Cloud Map mengharuskan Anda membersihkannya dalam urutan terbalik: instance layanan terlebih dahulu, lalu layanan, dan terakhir namespace.

1. Deregister instance layanan pertama:

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

2. Deregister instance layanan kedua:

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

3. Hapus layanan publik:

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

4. Hapus layanan backend:

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

5. Hapus namespace :

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

6. Verifikasi bahwa zona yang dihosting Route 53 telah dihapus:

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

Pelajari cara menggunakan penemuan AWS Cloud Map layanan dengan atribut khusus

Tutorial berikut menunjukkan bagaimana Anda dapat menggunakan penemuan AWS Cloud Map layanan dengan atribut khusus yang dapat ditemukan menggunakan API. AWS Cloud Map Tutorial memandu Anda melalui membuat dan menjalankan aplikasi klien menggunakan AWS CloudShell. Aplikasi menggunakan dua fungsi Lambda untuk menulis data ke tabel DynamoDB dan kemudian membaca dari tabel. Fungsi Lambda dan tabel DynamoDB terdaftar sebagai instance layanan. AWS Cloud Map Kode dalam aplikasi klien dan fungsi Lambda menggunakan atribut AWS Cloud Map khusus untuk menemukan sumber daya yang diperlukan untuk melakukan pekerjaan.

Untuk versi AWS CLI berbasis tutorial ini, lihatPelajari cara menggunakan penemuan AWS Cloud Map layanan dengan atribut kustom menggunakan AWS CLI.

> **Important**
>
> Anda akan membuat AWS sumber daya selama lokakarya yang akan dikenakan biaya di AWS akun Anda. Disarankan untuk membersihkan sumber daya segera setelah Anda menyelesaikan bengkel untuk meminimalkan biaya.

Prasyarat

Sebelum memulai, selesaikan langkah-langkah di Siapkan untuk digunakan AWS Cloud Map.

Langkah 1: Buat AWS Cloud Map namespace

Pada langkah ini, Anda membuat AWS Cloud Map namespace. Namespace adalah konstruksi yang digunakan untuk mengelompokkan layanan untuk aplikasi. Saat Anda membuat namespace, Anda menentukan bagaimana sumber daya akan ditemukan. Sumber daya yang dibuat di namespace yang dibuat pada langkah ini akan dapat ditemukan dengan panggilan AWS Cloud Map API menggunakan atribut khusus.

1. Masuk ke AWS Management Console dan buka AWS Cloud Map konsol di https://console.aws.amazon.com/cloudmap/.
2. Pilih Buat namespace.
3. Untuk nama Namespace, tentukan. `cloudmap-tutorial`
4. (Opsional) Untuk deskripsi Namespace, tentukan deskripsi untuk tujuan Anda menggunakan namespace.
5. Untuk penemuan Instance, pilih panggilan API.
6. Tinggalkan sisa nilai default dan pilih Buat namespace.

Langkah 2: Buat tabel DynamoDB

Pada langkah ini, Anda membuat tabel DynamoDB. Tabel ini digunakan untuk menyimpan dan mengambil data untuk aplikasi sampel yang akan Anda buat dalam langkah-langkah berikut.

Untuk informasi tentang cara membuat DynamoDB, lihat Langkah 1: Membuat tabel di DynamoDB dalam Panduan Pengembang DynamoDB dan gunakan tabel berikut untuk menentukan opsi apa yang akan ditentukan.

Opsi	Nilai	
Nama tabel	cloudmap	
Kunci partisi	id	

Simpan nilai default untuk sisa pengaturan dan buat tabel.

Langkah 3: Buat layanan AWS Cloud Map data dan daftarkan tabel DynamoDB sebagai contoh

Pada langkah ini, Anda membuat AWS Cloud Map layanan dan kemudian mendaftarkan tabel DynamoDB yang dibuat pada langkah terakhir sebagai instance layanan.

1. Buka AWS Cloud Map konsol di https://console.aws.amazon.com/cloudmap/
2. Dari daftar ruang nama, pilih **`cloudmap-tutorial`** namespace dan pilih Lihat detail.
3. Di bagian Layanan, pilih Buat layanan dan lakukan hal berikut.
   a. Untuk nama Layanan, masukkan`data-service`.
   b. Tinggalkan sisa nilai default dan pilih Buat layanan.
4. Di bagian Layanan, pilih `data-service` layanan dan pilih Lihat detail.
5. Di bagian Instans layanan, pilih Daftar instance layanan.
6. Pada halaman contoh layanan Register, lakukan hal berikut.
   a. Untuk jenis Instance, pilih Mengidentifikasi informasi untuk sumber daya lain.
   b. Untuk id contoh Layanan, tentukan`data-instance`.
   c. Di bagian Atribut khusus, tentukan pasangan kunci-nilai berikut: key =`tablename`, value =. `cloudmap`

Langkah 4: Buat peran AWS Lambda eksekusi

Pada langkah ini, Anda membuat peran IAM yang digunakan AWS Lambda fungsi pada langkah berikutnya. Anda dapat memberi nama peran IAM `cloudmap-tutorial-role` dan menghilangkan batas izin karena peran hanya digunakan untuk tutorial ini, dan Anda dapat menghapusnya setelahnya.

Untuk membuat peran layanan untuk Lambda (konsol IAM)

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi konsol IAM, pilih Peran, dan lalu pilih Buat peran.
3. Untuk jenis entitas Tepercaya, pilih Layanan AWS.
4. Untuk kasus Layanan atau penggunaan, pilih Lambda, lalu pilih kasus penggunaan Lambda.

5. Pilih Berikutnya.
6. Cari, lalu pilih kotak di samping, `PowerUserAccess` kebijakan, lalu pilih Berikutnya.
7. Pilih Berikutnya.
8. Untuk nama Peran, tentukan`cloudmap-tutorial-role`.
9. Tinjau peran lalu pilih Buat peran.

Langkah 5: Buat fungsi Lambda untuk menulis data

Pada langkah ini, Anda membuat fungsi Lambda yang ditulis dari awal yang menulis data ke tabel DynamoDB dengan menggunakan API untuk menanyakan layanan yang Anda buat. AWS Cloud Map AWS Cloud Map

Untuk informasi tentang membuat fungsi Lambda, lihat Membuat fungsi Lambda dengan konsol di Panduan AWS Lambda Pengembang dan gunakan tabel berikut untuk menentukan opsi apa yang akan ditentukan atau dipilih.

Opsi	Nilai
Nama fungsi	writefunction
Waktu Aktif	Python 3.12
Arsitektur	x86_64
Izin	Gunakan peran yang ada
Peran yang ada	cloudmap-tutorial-role

Setelah Anda membuat fungsi, perbarui kode contoh untuk mencerminkan kode Python berikut, dan kemudian menyebarkan fungsi. Perhatikan bahwa Anda menentukan atribut `datatable` kustom yang Anda kaitkan dengan instance AWS Cloud Map layanan yang Anda buat untuk tabel DynamoDB. Fungsi menghasilkan kunci yang merupakan angka acak antara 1 dan 100 dan mengaitkannya dengan nilai yang diteruskan ke fungsi ketika dipanggil.

```
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)
    }
```

Setelah menerapkan fungsi, untuk menghindari kesalahan batas waktu, perbarui batas waktu fungsi menjadi 5 detik. Untuk informasi selengkapnya, lihat Mengonfigurasi batas waktu fungsi Lambda di Panduan Pengembang.AWS Lambda

Langkah 6: Buat layanan AWS Cloud Map aplikasi dan daftarkan fungsi tulis Lambda sebagai instance

Pada langkah ini, Anda membuat AWS Cloud Map layanan dan kemudian mendaftarkan fungsi tulis Lambda sebagai instance layanan.

1. Buka AWS Cloud Map konsol di https://console.aws.amazon.com/cloudmap/
2. Di navigasi kiri, pilih Namespaces.
3. Dari daftar ruang nama, pilih **`cloudmap-tutorial`** namespace dan pilih Lihat detail.
4. Di bagian Layanan, pilih Buat layanan dan lakukan hal berikut.

    a. Untuk nama Layanan, masukkan`app-service`.

b. Tinggalkan sisa nilai default dan pilih Buat layanan.

5. Di bagian Layanan, pilih `app-service` layanan dan pilih Lihat detail.
6. Di bagian Instans layanan, pilih Daftar instance layanan.
7. Pada halaman contoh layanan Register, lakukan hal berikut.

   a. Untuk jenis Instance, pilih Mengidentifikasi informasi untuk sumber daya lain.

   b. Untuk id contoh Layanan, tentukan`write-instance`.

   c. Di bagian Atribut kustom, tentukan pasangan kunci-nilai berikut.

      - kunci =`action`, nilai = `write`
      - kunci =`functionname`, nilai = `writefunction`

Langkah 7: Buat fungsi Lambda untuk membaca data

Pada langkah ini, Anda membuat fungsi Lambda yang ditulis dari awal yang menulis data ke tabel DynamoDB yang Anda buat.

Untuk informasi tentang membuat fungsi Lambda, lihat Membuat fungsi Lambda dengan konsol di Panduan AWS Lambda Pengembang dan gunakan tabel berikut untuk menentukan opsi apa yang akan ditentukan atau dipilih.

Opsi	Nilai
Nama fungsi	fungsi baca
Waktu Aktif	Python 3.12
Arsitektur	x86_64
Izin	Gunakan peran yang ada
Peran yang ada	cloudmap-tutorial-role

Setelah Anda membuat fungsi, perbarui kode contoh untuk mencerminkan kode Python berikut, dan kemudian menyebarkan fungsi. Fungsi memindai tabel dan mengembalikan semua item.

```
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)
    }
```

Setelah menerapkan fungsi, untuk menghindari kesalahan batas waktu, perbarui batas waktu fungsi menjadi 5 detik. Untuk informasi selengkapnya, lihat Mengonfigurasi batas waktu fungsi Lambda di Panduan Pengembang.AWS Lambda

Langkah 8: Daftarkan fungsi baca Lambda sebagai instance layanan AWS Cloud Map

Pada langkah ini, Anda mendaftarkan fungsi baca Lambda sebagai instance layanan di `app-service` layanan yang sebelumnya Anda buat.

1. Buka AWS Cloud Map konsol di https://console.aws.amazon.com/cloudmap/
2. Di navigasi kiri, pilih Namespaces.
3. Dari daftar ruang nama, pilih **`cloudmap-tutorial`** namespace dan pilih Lihat detail.
4. Di bagian Layanan, pilih `app-service` layanan dan pilih Lihat detail.
5. Di bagian Instans layanan, pilih Daftar instance layanan.
6. Pada halaman contoh layanan Register, lakukan hal berikut.

    a. Untuk jenis Instance, pilih Mengidentifikasi informasi untuk sumber daya lain.

b. Untuk id contoh Layanan, tentukan`read-instance`.

c. Di bagian Atribut kustom, tentukan pasangan kunci-nilai berikut.

- kunci =`action`, nilai = `read`
- kunci =`functionname`, nilai = `readfunction`

Langkah 9: Buat dan jalankan klien baca dan tulis AWS CloudShell

Anda dapat membuat dan menjalankan aplikasi klien AWS CloudShell yang menggunakan kode untuk menemukan layanan yang Anda konfigurasikan AWS Cloud Map dan melakukan panggilan ke layanan ini.

1. Buka AWS CloudShell konsol di https://console.aws.amazon.com/cloudshell/
2. Gunakan perintah berikut untuk membuat file bernama`writefunction.py`.

```
vim writeclient.py
```

3. Dalam `writeclient.py` file, masuk ke mode insert dengan menekan `i` tombol. Kemudian, salin dan tempel kode berikut. Kode ini menemukan fungsi Lambda untuk menulis data dengan mencari `name=writeservice` atribut khusus dalam `app-service` layanan. Nama fungsi Lambda yang bertanggung jawab untuk menulis data ke tabel DynamoDB dikembalikan. Kemudian fungsi Lambda dipanggil, meneruskan payload sampel yang ditulis ke tabel sebagai nilai.

```
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. Tekan tombol escape, ketik `:wq`, dan tekan tombol enter untuk menyimpan file dan keluar.
5. Gunakan perintah berikut untuk menjalankan kode Python.

```
python3 writeclient.py
```

Outputnya harus berupa `200` respons, mirip dengan yang berikut ini.

```
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. Untuk memverifikasi penulisan berhasil pada langkah sebelumnya, buat klien baca.
   a. Gunakan perintah berikut untuk membuat file bernama`readfunction.py`.

```
vim readclient.py
```

   b. Dalam `readclient.py` file, tekan `i` tombol untuk masuk ke mode insert. Kemudian, salin dan tempel kode berikut. Kode ini memindai tabel dan akan mengembalikan nilai yang Anda tulis ke tabel pada langkah sebelumnya.

```
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. Tekan tombol escape, ketik `:wq`, dan tekan tombol enter untuk menyimpan file dan keluar.

d. Gunakan perintah berikut untuk menjalankan kode Python.

```
python3 readclient.py
```

Outputnya akan terlihat mirip dengan berikut ini, mencantumkan nilai yang ditulis ke tabel dengan menjalankan `writefunction.py` dan kunci acak yang dihasilkan dalam fungsi tulis 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}}"}'
```

Langkah 10: Bersihkan sumber daya

Setelah Anda menyelesaikan tutorial, hapus sumber daya untuk menghindari biaya tambahan. AWS Cloud Map mengharuskan Anda membersihkannya dalam urutan terbalik, instance layanan terlebih dahulu, lalu layanan, dan akhirnya namespace. Langkah-langkah berikut memandu Anda melalui pembersihan AWS Cloud Map sumber daya yang digunakan dalam tutorial.

Untuk menghapus sumber AWS Cloud Map daya

1. Masuk ke AWS Management Console dan buka AWS Cloud Map konsol di https://console.aws.amazon.com/cloudmap/.
2. Dari daftar ruang nama, pilih **`cloudmap-tutorial`** namespace dan pilih Lihat detail.
3. Pada halaman detail namespace, dari daftar layanan, pilih `data-service` layanan dan pilih Lihat detail.
4. Di bagian Service instance, pilih `data-instance` instance dan pilih Deregister.
5. Menggunakan breadcrumb di bagian atas halaman, pilih cloudmap-tutorial.com untuk menavigasi kembali ke halaman detail namespace.
6. Pada halaman detail namespace, dari daftar layanan, pilih layanan layanan data dan pilih Hapus.

7. Ulangi langkah 3-6 untuk `app-service` layanan dan instance `write-instance` dan `read-instance` layanan.
8. Di navigasi kiri, pilih Namespaces.
9. Pilih **`cloudmap-tutorial`** namespace dan pilih Delete.

Tabel berikut mencantumkan prosedur yang dapat Anda ikuti untuk menghapus sumber daya lain yang digunakan dalam tutorial.

Sumber Daya	Langkah-langkah	
Tabel DynamoDB	Langkah 6: (Opsional) Hapus tabel DynamoDB Anda untuk membersihkan sumber daya di Panduan Pengembang Amazon DynamoDB	
Fungsi Lambda dan peran eksekusi IAM terkait	Bersihkan di Panduan AWS Lambda Pengembang	

Pelajari cara menggunakan penemuan AWS Cloud Map layanan dengan atribut kustom menggunakan AWS CLI

Tutorial ini menunjukkan bagaimana Anda dapat menggunakan penemuan AWS Cloud Map layanan dengan atribut khusus. Anda akan membuat aplikasi microservices yang digunakan AWS Cloud Map untuk menemukan sumber daya secara dinamis menggunakan atribut kustom. Aplikasi ini terdiri dari dua fungsi Lambda yang menulis data ke dan membaca dari tabel DynamoDB, dengan semua sumber daya terdaftar di. AWS Cloud Map

Untuk AWS Management Console versi tutorialnya, lihatPelajari cara menggunakan penemuan AWS Cloud Map layanan dengan atribut khusus.

Prasyarat

Sebelum Anda memulai tutorial ini, selesaikan langkah-langkahnyaSiapkan untuk digunakan AWS Cloud Map.

Buat AWS Cloud Map namespace

Namespace adalah konstruksi yang digunakan untuk mengelompokkan layanan untuk aplikasi. Pada langkah ini, Anda akan membuat namespace yang memungkinkan sumber daya dapat ditemukan melalui panggilan API. AWS Cloud Map

1. Jalankan perintah berikut untuk membuat namespace HTTP:

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

   Perintah mengembalikan ID operasi. Anda dapat memeriksa status operasi dengan perintah berikut:

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

2. Setelah namespace dibuat, Anda dapat mengambil ID-nya untuk digunakan dalam perintah berikutnya:

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

3. Simpan ID namespace dalam variabel untuk digunakan nanti:

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

Membuat tabel DynamoDB

Selanjutnya, buat tabel DynamoDB yang akan menyimpan data untuk aplikasi Anda:

1. Jalankan perintah berikut untuk membuat tabel:

   ```
   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. Tunggu tabel menjadi aktif sebelum melanjutkan:

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

   Perintah ini menunggu sampai tabel sepenuhnya dibuat dan siap digunakan.

Buat layanan AWS Cloud Map data dan daftarkan tabel DynamoDB

Sekarang, buat layanan di namespace Anda untuk mewakili sumber daya penyimpanan data:

1. Jalankan perintah berikut untuk membuat AWS Cloud Map layanan untuk sumber daya penyimpanan data:

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

2. Dapatkan ID layanan untuk layanan data:

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

3. Mendaftarkan tabel DynamoDB sebagai contoh layanan dengan atribut kustom yang menentukan nama tabel:

```
aws servicediscovery register-instance \
  --service-id $DATA_SERVICE_ID \
  --instance-id data-instance \
  --attributes tablename=cloudmap
```

   Atribut kustom `tablename=cloudmap` memungkinkan layanan lain untuk menemukan nama tabel DynamoDB secara dinamis.

Buat peran IAM untuk fungsi Lambda

Buat peran IAM yang akan digunakan fungsi Lambda untuk AWS mengakses sumber daya:

1. Buat dokumen kebijakan kepercayaan untuk peran IAM:

```
cat > lambda-trust-policy.json << EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
```

2. Jalankan perintah berikut untuk membuat peran IAM menggunakan kebijakan kepercayaan:

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

3. Buat file untuk kebijakan IAM khusus dengan izin hak istimewa paling sedikit:

```
cat > cloudmap-policy.json << EOF
{
  "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"
    }
  ]
}
EOF
```

4. Buat dan lampirkan kebijakan ke peran 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
```

Buat fungsi Lambda untuk menulis data

Untuk membuat fungsi Lambda yang menulis data ke tabel DynamoDB, ikuti langkah-langkah berikut:

1. Buat file Python untuk fungsi tulis:

```
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
```

Fungsi ini digunakan AWS Cloud Map untuk menemukan nama tabel DynamoDB dari atribut kustom, lalu menulis data ke tabel.

2. Package dan deploy fungsi 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. Perbarui batas waktu fungsi untuk menghindari kesalahan batas waktu:

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

Buat layanan AWS Cloud Map aplikasi dan daftarkan fungsi tulis Lambda

Untuk membuat layanan lain di namespace Anda untuk mewakili fungsi aplikasi, ikuti langkah-langkah berikut:

1. Buat layanan untuk fungsi aplikasi:

```
aws servicediscovery create-service \
  --name app-service \
```

```
  --namespace-id $NAMESPACE_ID \
  --creator-request-id app-service-request
```

2. Dapatkan ID layanan untuk layanan aplikasi:

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

3. Daftarkan fungsi tulis Lambda sebagai instance layanan dengan atribut khusus:

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

   Atribut kustom `action=write` dan `functionname=writefunction` memungkinkan klien untuk menemukan fungsi ini berdasarkan tujuannya.

Buat fungsi Lambda untuk membaca data

Untuk membuat fungsi Lambda yang membaca data dari tabel DynamoDB, ikuti langkah-langkah berikut:

1. Buat file Python untuk fungsi baca:

```
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
```

Fungsi ini juga digunakan AWS Cloud Map untuk menemukan nama tabel DynamoDB, lalu membaca data dari tabel. Ini termasuk penanganan kesalahan dan komentar pagination.

2. Package dan deploy fungsi 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. Perbarui batas waktu fungsi:

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

Daftarkan fungsi baca Lambda sebagai instance layanan

Untuk mendaftarkan fungsi baca Lambda sebagai instance layanan lain di layanan aplikasi, ikuti langkah ini:

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

Atribut kustom `action=read` dan `functionname=readfunction` memungkinkan klien untuk menemukan fungsi ini berdasarkan tujuannya.

Membuat dan menjalankan aplikasi klien

Untuk membuat aplikasi klien Python yang digunakan AWS Cloud Map untuk menemukan dan memanggil fungsi tulis, ikuti langkah-langkah berikut:

1. Buat file Python untuk aplikasi klien tulis:

```
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
```

Klien ini menggunakan `QueryParameters` opsi untuk menemukan instance layanan dengan `action=write` atribut.

2. Buat file Python untuk aplikasi klien baca:

```
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. Jalankan klien tulis untuk menambahkan data ke tabel DynamoDB:

```
python3 writeclient.py
```

Output harus menunjukkan respons yang berhasil dengan kode status HTTP 200.

4. Jalankan klien baca untuk mengambil data dari tabel DynamoDB:

```
python3 readclient.py
```

Output harus menunjukkan data yang ditulis ke tabel, termasuk ID yang dihasilkan secara acak dan nilai “Ini adalah data uji”.

Pembersihan sumber daya

Setelah selesai dengan tutorial, bersihkan sumber daya untuk menghindari biaya tambahan.

1. Pertama, jalankan perintah berikut untuk membatalkan pendaftaran instance layanan:

```
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. Jalankan perintah berikut untuk menghapus layanan:

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

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

3. Jalankan perintah berikut untuk menghapus namespace:

```
aws servicediscovery delete-namespace \
```

```
  --id $NAMESPACE_ID
```

4. Jalankan perintah berikut untuk menghapus fungsi Lambda:

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

5. Jalankan perintah berikut untuk menghapus peran dan kebijakan 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. Jalankan perintah berikut untuk menghapus tabel DynamoDB:

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

7. Jalankan perintah berikut untuk membersihkan file sementara:

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

AWS Cloud Map ruang nama

Namespace adalah entitas logis AWS Cloud Map yang digunakan untuk mengelompokkan layanan aplikasi di bawah nama umum dan tingkat kemampuan ditemukan. Saat Anda membuat namespace, Anda menentukan yang berikut:

- Nama yang Anda ingin aplikasi Anda gunakan untuk menemukan instance.
- Metode di mana contoh layanan yang Anda daftarkan AWS Cloud Map dapat ditemukan. Anda dapat memutuskan apakah sumber daya Anda perlu ditemukan secara publik melalui internet, secara pribadi di virtual private cloud (VPC) tertentu, atau hanya dengan panggilan API.

Berikut ini adalah konsep umum tentang ruang nama.

- Ruang nama khusus untuk tempat Wilayah AWS mereka dibuat. Untuk digunakan AWS Cloud Map di beberapa wilayah, Anda harus membuat ruang nama di setiap wilayah.
- Jika Anda membuat namespace untuk memungkinkan misalnya penemuan oleh kueri DNS di VPC, AWS Cloud Map secara otomatis membuat zona yang dihosting Route 53 pribadi. Zona yang dihosting ini dapat dikaitkan dengan beberapa VPCs. Untuk informasi selengkapnya, lihat Mengaitkan VPCWith HostedZone di Referensi API Amazon Route 53.

Topik

Membuat AWS Cloud Map namespace untuk mengelompokkan layanan aplikasi

Anda dapat membuat namespace untuk mengelompokkan layanan untuk aplikasi Anda dengan nama ramah yang memungkinkan penemuan sumber daya aplikasi melalui panggilan API atau kueri DNS.

Opsi penemuan instance

Tabel berikut merangkum berbagai opsi penemuan instance AWS Cloud Map dan jenis namespace yang sesuai yang dapat Anda buat, tergantung pada layanan dan penyiapan aplikasi Anda.

Jenis namespace	Metode penemuan instance	Cara kerjanya	Informasi tambahan
HTTP	Panggilan API	Sumber daya dalam aplikasi Anda dapat menemukan sumber daya lain hanya dengan memanggil `DiscoverInstances` API.	• DiscoverInstances • CreateHttpNamespace
DNS privat	Panggilan API dan kueri DNS di VPC	Sumber daya dalam aplikasi Anda dapat menemukan sumber daya lain dengan memanggil `DiscoverInstances` API, dan dengan menanyakan server nama di zona host Route 53 pribadi yang dibuat secara otomatis. AWS Cloud Map Zona host yang dibuat oleh AWS Cloud Map memiliki nama yang sama dengan namespace dan berisi catatan DNS yang memiliki nama dalam	• DiscoverInstances • CreatePrivateDnsNamespace

Jenis namespace	Metode penemuan instance	Cara kerjanya	Informasi tambahan
		format. *service-name namespace-name*. **Note** Penyelesai Route 53 menyelesaikan kueri DNS yang berasal dari VPC menggunakan catatan di zona yang dihosting privat. Jika zona yang di-hosting privat tidak termasuk catatan yang cocok dengan nama domain dalam kueri DNS, Route 53 menanggapi kueri dengan `NXDOMAIN` (domain yang tidak ada).	

Jenis namespace	Metode penemuan instance	Cara kerjanya	Informasi tambahan
DNS Publik	Panggilan API dan kueri DNS publik	Sumber daya dalam aplikasi Anda dapat menemukan sumber daya lain dengan memanggil `DiscoverInstances` API dan dengan menanyakan server nama di zona host Route 53 publik yang dibuat secara otomatis. AWS Cloud Map Zona yang dihosting publik memiliki nama yang sama dengan namespace dan berisi catatan DNS yang memiliki nama dalam format. *`service-name namespace-name`*. **Note** Nama namespace dalam hal ini harus berupa nama domain yang	• DiscoverInstances • CreatePublicDnsNamespace

Jenis namespace	Metode penemuan instance	Cara kerjanya	Informasi tambahan
		telah Anda daftarkan.	

Prosedur

Anda dapat mengikuti langkah-langkah ini untuk membuat namespace menggunakan AWS CLI, AWS Management Console, atau SDK untuk Python.

AWS Management Console

1. Masuk ke AWS Management Console dan buka AWS Cloud Map konsol di https://console.aws.amazon.com/cloudmap/.
2. Pilih Buat namespace.
3. Untuk nama Namespace, masukkan nama yang akan digunakan untuk menemukan instance.

   > **Note**
   >
   > - Ruang nama yang dikonfigurasi untuk kueri DNS publik harus diakhiri dengan domain tingkat atas. Misalnya, `.com`.
   > - Anda dapat menentukan nama domain yang diinternasionalisasi (IDN) jika Anda mengubah namanya menjadi Punycode terlebih dahulu. Untuk informasi tentang pengubah online, lakukan pencarian internet di "punycode converter".
   >
   >   Anda juga dapat mengubah nama domain yang diinternasionalisasi menjadi Punycode saat Anda membuat namespace secara terprogram. Misalnya, jika Anda menggunakan Java, Anda dapat mengkonversi nilai Unicode ke Punycode dengan menggunakan `toASCII` metode perpustakaan java.net.IDN.

4. (Opsional) Untuk deskripsi Namespace, masukkan informasi tentang namespace yang akan terlihat di halaman Namespaces dan di bawah informasi Namespace. Anda dapat menggunakan informasi ini untuk mengidentifikasi namespace dengan mudah.
5. Untuk penemuan Instance, Anda dapat memilih antara panggilan API, panggilan API, dan kueri DNS VPCs, serta panggilan API dan kueri DNS publik untuk membuat ruang nama

HTTP, DNS pribadi, atau DNS publik. Untuk informasi selengkapnya, lihat Opsi penemuan instance.

Berdasarkan pilihan Anda, ikuti langkah-langkah ini.

- Jika Anda memilih panggilan API dan kueri DNS di VPCs, untuk VPC, pilih virtual private cloud (VPC) yang ingin Anda kaitkan dengan namespace.
- Jika Anda memilih panggilan API dan kueri DNS dalam VPCs atau panggilan API dan kueri DNS publik, untuk TTL, tentukan nilai numerik dalam hitungan detik. Nilai time to live (TTL) menentukan berapa lama DNS menyelesaikan informasi cache untuk catatan DNS start of authority (SOA) dari zona host Route 53 yang dibuat dengan namespace Anda. Untuk informasi selengkapnya tentang TTL, lihat TTL (detik) di Panduan Pengembang Amazon Route 53.

6. (Opsional) Di bawah Tag, pilih Tambahkan tag lalu tentukan kunci dan nilai untuk menandai namespace Anda. Anda dapat menentukan satu atau lebih tag untuk ditambahkan ke namespace Anda. Tag memungkinkan Anda untuk mengkategorikan AWS sumber daya Anda sehingga Anda dapat lebih mudah mengelolanya. Untuk informasi selengkapnya, lihat Menandai sumber daya Anda AWS Cloud Map.
7. Pilih Buat namespace. Anda dapat melihat status operasi dengan menggunakan ListOperations. Untuk informasi selengkapnya, lihat ListOperationsdi Referensi AWS Cloud Map API

AWS CLI

- Buat namespace dengan perintah untuk jenis penemuan instance yang Anda inginkan (ganti *red* nilainya dengan milik Anda sendiri).
  - Buat namespace HTTP menggunakan. `create-http-namespace` Contoh layanan yang terdaftar menggunakan namespace HTTP dapat ditemukan menggunakan `DiscoverInstances` permintaan, tetapi tidak dapat ditemukan menggunakan DNS.

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

  - Buat namespace pribadi berdasarkan DNS dan hanya terlihat di dalam VPC Amazon tertentu menggunakan. `create-private-dns-namespace` Anda dapat menemukan instance yang terdaftar dengan namespace DNS pribadi dengan menggunakan permintaan atau menggunakan DNS `DiscoverInstances`

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

- Buat namespace publik berdasarkan DNS yang terlihat di internet menggunakan. `create-public-dns-namespace` Anda dapat menemukan instans yang didaftarkan dengan namespace DNS publik dengan menggunakan permintaan `DiscoverInstances` atau menggunakan DNS.

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

AWS SDK for Python (Boto3)

1. Jika Anda belum `Boto3` menginstal, Anda dapat menemukan petunjuk untuk menginstal, mengkonfigurasi, dan menggunakan `Boto3` di sini.
2. Impor `Boto3` dan gunakan `servicediscovery` sebagai layanan Anda.

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

3. Buat namespace dengan perintah untuk tipe penemuan instance yang Anda inginkan (ganti *red* nilainya dengan milik Anda sendiri):

   - Buat namespace HTTP menggunakan. `create_http_namespace()` Contoh layanan yang terdaftar menggunakan namespace HTTP dapat ditemukan menggunakan`discover_instances()`, tetapi tidak dapat ditemukan menggunakan DNS.

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

   - Buat namespace pribadi berdasarkan DNS dan hanya terlihat di dalam VPC Amazon tertentu menggunakan. `create_private_dns_namespace()` Anda dapat menemukan instance yang terdaftar dengan namespace DNS pribadi dengan menggunakan salah satu atau menggunakan DNS `discover_instances()`

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

- Buat namespace publik berdasarkan DNS yang terlihat di internet menggunakan. `create_public_dns_namespace()` Anda dapat menemukan instance yang terdaftar dengan namespace DNS publik dengan menggunakan salah satu atau `discover_instances()` menggunakan DNS.

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

- Contoh keluaran respons

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

Langkah selanjutnya

Setelah membuat namespace, Anda dapat membuat layanan di namespace untuk mengelompokkan sumber daya aplikasi yang secara kolektif melayani tujuan tertentu dalam aplikasi Anda. Layanan bertindak sebagai templat untuk mendaftarkan sumber daya aplikasi sebagai instance. Untuk informasi selengkapnya tentang membuat AWS Cloud Map layanan, lihatMembuat AWS Cloud Map layanan untuk komponen aplikasi.

Daftar ruang AWS Cloud Map nama

Setelah membuat ruang nama, Anda dapat melihat daftar ruang nama yang telah Anda buat dengan mengikuti langkah-langkah ini.

AWS Management Console

1. Masuk ke AWS Management Console dan buka AWS Cloud Map konsol di https://console.aws.amazon.com/cloudmap/.
2. Di panel navigasi, pilih Namespaces untuk melihat daftar ruang nama. Anda dapat memesan ruang nama berdasarkan nama, deskripsi, mode penemuan instance, atau ID namespace. Anda juga dapat memasukkan nama namespace atau ID ke dalam kolom pencarian untuk mencari dan melihat namespace tertentu.

AWS CLI

- Buat daftar ruang nama dengan perintah. `list-namespaces`

```
aws servicediscovery list-namespaces
```

AWS SDK for Python (Boto3)

1. Jika Anda belum `Boto3` menginstal, Anda dapat menemukan petunjuk untuk menginstal, mengkonfigurasi, dan menggunakan `Boto3` di sini.
2. Impor `Boto3` dan gunakan `servicediscovery` sebagai layanan Anda.

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

3. Daftar ruang nama dengan. `list_namespaces()`

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

Contoh keluaran respons

```
{
```

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

Menghapus namespace AWS Cloud Map

Setelah selesai menggunakan namespace, Anda dapat menghapusnya. Saat menghapus namespace, Anda tidak lagi dapat menggunakannya untuk mendaftar atau menemukan instans layanan.

> **Note**
>
> Saat Anda membuat namespace, jika Anda menentukan bahwa Anda ingin menemukan instance layanan menggunakan kueri DNS publik atau kueri DNS VPCs, AWS Cloud Map buat zona host publik atau pribadi Amazon Route 53. Saat Anda menghapus namespace, AWS Cloud Map menghapus zona host yang sesuai.

Sebelum menghapus namespace, Anda harus membatalkan pendaftaran semua instance layanan dan kemudian menghapus semua layanan yang dibuat di namespace. Untuk informasi selengkapnya, silakan lihat Membatalkan pendaftaran instance layanan AWS Cloud Map dan Menghapus layanan AWS Cloud Map.

Setelah membatalkan pendaftaran instance dan menghapus layanan yang dibuat di namespace, ikuti langkah-langkah berikut untuk menghapus namespace.

AWS Management Console

1. Masuk ke AWS Management Console dan buka AWS Cloud Map konsol di https://console.aws.amazon.com/cloudmap/.
2. Di panel navigasi, pilih Namespace.
3. Pilih namespace yang ingin Anda hapus, lalu pilih Hapus.

4. Konfirmasikan bahwa Anda ingin menghapus layanan dengan memilih Hapus lagi.

AWS CLI

- Hapus namespace dengan `delete-namespace` perintah (ganti *red* nilainya dengan milik Anda). Jika namespace masih berisi satu atau beberapa layanan, permintaan gagal.

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

AWS SDK for Python (Boto3)

1. Jika Anda belum `Boto3` menginstal, Anda dapat menemukan petunjuk untuk menginstal, mengkonfigurasi, dan menggunakan `Boto3` di sini.
2. Impor `Boto3` dan gunakan `servicediscovery` sebagai layanan Anda.

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

3. Hapus namespace dengan `delete_namespace()` (ganti *red* nilainya dengan milik Anda). Jika namespace masih berisi satu atau beberapa layanan, permintaan gagal.

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

Contoh keluaran respons

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

AWS Cloud Map layanan

AWS Cloud Map Layanan adalah template untuk mendaftarkan instance layanan yang terdiri dari nama layanan dan konfigurasi DNS, jika berlaku, untuk layanan. Anda juga dapat mengatur pemeriksaan kesehatan untuk menentukan status kesehatan contoh dalam layanan dan menyaring sumber daya yang tidak sehat. Layanan dapat mewakili komponen aplikasi Anda. Misalnya, Anda dapat membuat layanan untuk sumber daya yang menangani pembayaran pada aplikasi Anda dan lainnya untuk sumber daya yang mengelola pengguna.

Layanan memungkinkan Anda menemukan sumber daya untuk aplikasi dengan mendapatkan kembali satu atau lebih titik akhir yang dapat digunakan untuk terhubung ke sumber daya. Lokasi sumber daya dilakukan menggunakan kueri DNS atau tindakan AWS Cloud Map `DiscoverInstances`API, tergantung pada cara Anda mengonfigurasi namespace. Anda dapat menggunakan AWS Cloud Map konsol untuk membuat cakupan penemuan instance di tingkat layanan.

Anda juga dapat menentukan metadata kustom sebagai atttributes di tingkat layanan menggunakan API. `UpdateServiceAttributes` Anda dapat mengatur atribut layanan untuk menghindari duplikasi atribut di seluruh instance dan memodifikasi atribut ini tanpa perlu membuat perubahan apa pun pada atribut instance. Informasi yang dapat Anda tentukan sebagai atribut di tingkat layanan termasuk, tetapi tidak terbatas pada, berikut ini:

- Bobot titik akhir untuk memindahkan lalu lintas selama penerapan progresif.
- Preferensi layanan seperti batas waktu API dan kebijakan coba ulang yang disarankan.

Untuk informasi selengkapnya, lihat UpdateServiceAttributesdi referensi AWS Cloud Map API.

Topik berikut menjelaskan pemeriksaan kesehatan dan konfigurasi DNS untuk layanan dan menyertakan instruksi untuk membuat, mencantumkan, memperbarui, dan menghapus layanan.

Topik

- Menghapus layanan AWS Cloud Map

AWS Cloud Map konfigurasi pemeriksaan kesehatan layanan

Pemeriksaan Kesehatan membantu menentukan apakah contoh layanan sehat atau tidak. Jika Anda tidak mengonfigurasi pemeriksaan kesehatan selama pembuatan layanan, lalu lintas akan diarahkan ke instance layanan terlepas dari status kesehatan instans. Saat Anda mengonfigurasi pemeriksaan kesehatan, AWS Cloud Map mengembalikan sumber daya yang sehat secara default. Anda dapat menggunakan `HealthStatus` parameter `DiscoverInstances` API untuk memfilter sumber daya berdasarkan status kesehatan dan mendapatkan daftar sumber daya yang tidak sehat. Anda juga dapat menggunakan `GetInstancesHealthStatus` API untuk mengambil status kesehatan dari instance layanan tertentu.

Anda dapat mengonfigurasi pemeriksaan kesehatan Route 53 atau pemeriksaan kesehatan pihak ketiga khusus saat membuat AWS Cloud Map layanan.

Pemeriksaan kondisi Route 53

Jika Anda menentukan setelan untuk pemeriksaan kesehatan Amazon Route 53, AWS Cloud Map buat pemeriksaan kesehatan Route 53 setiap kali Anda mendaftarkan instance dan menghapus pemeriksaan kesehatan saat Anda membatalkan pendaftaran instans.

Untuk ruang nama DNS publik, AWS Cloud Map mengaitkan pemeriksaan kesehatan dengan catatan Route 53 yang AWS Cloud Map dibuat saat Anda mendaftarkan instance.Jika Anda menentukan keduanya `A` dan jenis `AAAA` rekaman dalam konfigurasi DNS layanan, AWS Cloud Map buat pemeriksaan kesehatan yang menggunakan IPv4 alamat untuk memeriksa kesehatan sumber daya. Jika titik akhir yang ditentukan oleh IPv4 alamat tidak sehat, Route 53 menganggap kedua `A` dan `AAAA` catatan tidak sehat. Jika Anda menentukan jenis `CNAME` rekaman dalam konfigurasi DNS layanan, Anda tidak dapat mengonfigurasi pemeriksaan kesehatan Route 53.

Untuk ruang nama yang Anda gunakan panggilan API untuk menemukan instans untuk, AWS Cloud Map membuat pemeriksaan kesehatan Route 53. Namun, tidak ada catatan DNS untuk AWS Cloud Map mengaitkan pemeriksaan kesehatan dengan. Untuk menentukan apakah pemeriksaan kesehatan sehat, Anda dapat mengonfigurasi pemantauan menggunakan konsol Route 53 atau menggunakan Amazon CloudWatch. Untuk informasi selengkapnya tentang menggunakan konsol Route 53, lihat Dapatkan pemberitahuan ketika Pemeriksaan Kondisi gagal dalam Panduan Pengembang Amazon Route 53. Untuk informasi selengkapnya tentang penggunaan CloudWatch, lihat PutMetricAlarmdi Referensi Amazon CloudWatch API.

> **Note**
>
> - Anda tidak dapat mengonfigurasi pemeriksaan kesehatan Amazon Route 53 untuk layanan yang dibuat di namespace DNS pribadi.
> - Pemeriksa kesehatan Route 53 di setiap pemeriksaan kesehatan Wilayah AWS mengirimkan permintaan pemeriksaan kesehatan ke titik akhir setiap 30 detik. Rata-rata, titik akhir Anda menerima permintaan pemeriksaan kondisi setiap dua detik. Namun, pemeriksa kondisi tidak berkoordinasi satu sama lain. Oleh karena itu, terkadang Anda mungkin melihat beberapa permintaan dalam satu detik yang diikuti oleh beberapa detik tanpa pemeriksaan kondisi sama sekali. Untuk daftar wilayah pemeriksaan kesehatan, lihat Wilayah.

Untuk informasi tentang biaya untuk pemeriksaan kesehatan Route 53, lihat Route 53 Harga.

Pemeriksaan kesehatan khusus

Jika Anda mengonfigurasi AWS Cloud Map untuk menggunakan pemeriksaan kesehatan khusus saat mendaftarkan instans, Anda harus menggunakan pemeriksa kesehatan pihak ketiga untuk mengevaluasi kesehatan sumber daya Anda. Pemeriksaan kesehatan kustom berguna dalam keadaan berikut:

- Anda tidak dapat menggunakan pemeriksaan kesehatan Route 53 karena sumber daya tidak tersedia melalui internet. Misalnya, anggaplah bahwa Anda memiliki instans yang terletak di Amazon VPC. Anda dapat menggunakan pemeriksaan kesehatan kustom untuk contoh ini. Namun, agar pemeriksaan kesehatan bekerja, pemeriksa kesehatan Anda juga harus berada di VPC yang sama dengan instans Anda.
- Anda ingin menggunakan pemeriksa kesehatan pihak ketiga terlepas dari mana sumber daya Anda berada.

Saat Anda menggunakan pemeriksaan kesehatan khusus, AWS Cloud Map tidak memeriksa kesehatan sumber daya yang diberikan secara langsung. Sebagai gantinya, pemeriksa kesehatan pihak ketiga memeriksa kesehatan sumber daya dan mengembalikan status ke aplikasi Anda. Aplikasi Anda kemudian harus mengirimkan `UpdateInstanceCustomHealthStatus` permintaan yang menyampaikan status ini ke AWS Cloud Map. Jika status awal yang diteruskan adalah`UNHEALTHY`, dan jika tidak ada yang lain `UpdateInstanceCustomHealthStatus` dalam

30 detik yang menyampaikan statusHEALTHY, sumber daya dipastikan tidak sehat. AWS Cloud Map berhenti merutekan lalu lintas ke sumber daya itu.

AWS Cloud Map konfigurasi layanan DNS

Saat Anda membuat layanan di namespace yang mendukung penemuan instance oleh kueri DNS, AWS Cloud Map buat catatan DNS Route 53. Anda harus menentukan kebijakan perutean Route 53 dan jenis catatan DNS yang akan berlaku untuk semua catatan DNS Route 53 yang dibuat. AWS Cloud Map

Kebijakan perutean

Kebijakan perutean menentukan cara Route 53 merespons kueri DNS yang digunakan untuk penemuan instance layanan. Kebijakan routing yang didukung dan bagaimana kaitannya AWS Cloud Map adalah sebagai berikut.

Perutean tertimbang

Route 53 mengembalikan nilai yang berlaku dari satu instance AWS Cloud Map layanan yang dipilih secara acak dari antara instance yang Anda daftarkan menggunakan layanan yang sama AWS Cloud Map . Semua catatan memiliki bobot yang sama, sehingga Anda tidak dapat merutekan lebih atau kurang lalu lintas ke setiap instans.

Sebagai contoh, misalkan layanan termasuk konfigurasi untuk satu catatan A dan pemeriksaan kesehatan, dan Anda menggunakan layanan untuk mendaftarkan 10 instans. Route 53 menanggapi permintaan DNS dengan alamat IP untuk satu instans yang dipilih secara acak dari antara instans yang sehat. Jika tidak ada instans yang sehat, Route 53 menanggapi kueri DNS seolah-olah semua instans sehat.

Jika Anda tidak menentukan pemeriksaan kesehatan untuk layanan, Route 53 mengasumsikan bahwa semua instans sehat dan mengembalikan nilai yang berlaku untuk satu instans yang dipilih secara acak.

Untuk informasi lebih lanjut, lihat perutean Tertimbang dalam Panduan Pengembang Amazon Route 53.

Rute jawaban multinilai

Jika Anda menentukan pemeriksaan kesehatan untuk layanan dan hasil pemeriksaan kesehatan sehat, Route 53 mengembalikan nilai yang berlaku hingga delapan instans.

Misalnya, anggaplah bahwa layanan tersebut mencakup konfigurasi untuk catatan A dan pemeriksaan kesehatan. Anda menggunakan layanan untuk mendaftar 10 instans. Route 53 menanggapi permintaan DNS dengan alamat IP untuk hanya maksimal delapan instans sehat. Jika kurang dari delapan instans sehat, Route 53 menanggapi setiap permintaan DNS dengan alamat IP untuk semua instans sehat.

Jika Anda tidak menentukan pemeriksaan kesehatan untuk layanan, Route 53 mengasumsikan bahwa semua instans sehat dan mengembalikan nilai hingga delapan instans.

Untuk informasi selengkapnya, lihat Merutekan jawaban multinilai di Panduan Pengembang Amazon Route 53.

Jenis catatan

Jenis catatan DNS Route 53 menentukan jenis nilai yang dikembalikan Route 53 sebagai respons terhadap kueri DNS yang digunakan untuk penemuan instance layanan. Berbagai jenis rekaman DNS yang dapat Anda tentukan, dan nilai terkait yang dikembalikan oleh Route 53 sebagai respons terhadap kueri adalah sebagai berikut.

A

Jika Anda menentukan jenis ini, Route 53 mengembalikan alamat IP sumber daya dalam IPv4 format, seperti 192.0.2.44.

AAAA

Jika Anda menentukan jenis ini, Route 53 mengembalikan alamat IP sumber daya dalam IPv6 format, seperti 2001:0 db 8:85 a 3:0000:0000:abcd: 0001:2345.

CNAME

Jika Anda menentukan jenis ini, Route 53 mengembalikan nama domain sumber daya (seperti www.example.com).

> **Note**
>
> - Untuk mengonfigurasi data DNS CNAME, Anda harus menentukan kebijakan perutean tertimbang.
> - Saat mengonfigurasi data DNS CNAME, Anda tidak dapat mengonfigurasi pemeriksaan kesehatan Route 53.

SRV

Jika Anda menentukan jenis ini, Route 53 mengembalikan nilai untuk SRV catatan. Nilai untuk catatan SRV menggunakan nilai-nilai berikut:

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

Pertimbangkan hal berikut:

- Nilai dari `priority` dan `weight` keduanya diatur ke 1 dan tidak dapat diubah.
- Untuk`port`, AWS Cloud Map gunakan nilai yang Anda tentukan untuk Port (AWS_INSTANCE_PORT) saat Anda mendaftarkan instance.
- Nilai `service-hostname` adalah rangkaian nilai berikut:
  - Nilai yang Anda tentukan untuk ID instance Service (InstanceId) saat Anda mendaftarkan instance
  - Nama layanan
  - Nama namespace

  Misalnya, Anda menentukan tes sebagai ID instance saat Anda mendaftarkan instance. Nama layanan ini adalah backend dan nama namespace adalah contoh.com. AWS Cloud Map menugaskan nilai berikut untuk `service-hostname` atribut dalam SRV rekaman:

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

> **Note**
>
> Jika Anda menentukan nilai IPv4 alamat, IPv6 alamat, atau keduanya saat Anda mendaftarkan instance, AWS Cloud Map secara otomatis membuat catatan A dan/atau AAAA yang memiliki nama yang sama dengan nilai `service-hostname` dalam catatan SRV.

Anda dapat menentukan jenis catatan dalam kombinasi berikut:

- A
- AAAA
- A dan AAAA
- CNAME

- SRV

Jika Anda menentukan jenis catatan A dan AAAA, Anda dapat menentukan alamat IPv4 IP, alamat IPv6 IP, atau keduanya saat Anda mendaftarkan instance.

Membuat AWS Cloud Map layanan untuk komponen aplikasi

Setelah membuat namespace, Anda dapat membuat layanan untuk mewakili berbagai komponen aplikasi Anda yang melayani tujuan tertentu. Misalnya, Anda dapat membuat layanan untuk sumber daya di aplikasi Anda yang memproses pembayaran.

> **Note**
>
> Anda tidak dapat membuat beberapa layanan yang dapat diakses oleh kueri DNS dengan nama yang hanya berbeda menurut kasus (seperti CONTOH dan contoh). Mencoba melakukannya akan menghasilkan layanan ini memiliki nama DNS yang sama. Jika Anda menggunakan namespace yang hanya dapat diakses oleh panggilan API, maka Anda dapat membuat layanan yang dengan nama yang berbeda hanya berdasarkan huruf.

Ikuti langkah-langkah ini untuk membuat layanan menggunakan AWS Management Console, AWS CLI, dan SDK untuk Python.

AWS Management Console

1. Masuk ke AWS Management Console dan buka AWS Cloud Map konsol di https://console.aws.amazon.com/cloudmap/.
2. Di panel navigasi, pilih Namespace.
3. Pada Namespace halaman, pilih namespace yang Anda ingin menambahkan layanan.
4. Pada *namespace-name* halaman Namespace: pilih Buat layanan.
5. Untuk nama Layanan, masukkan nama yang menjelaskan contoh yang Anda daftarkan saat menggunakan layanan ini. Nilai digunakan untuk menemukan instance AWS Cloud Map layanan baik dalam panggilan API atau dalam kueri DNS.

> **Note**
>
> Jika Anda AWS Cloud Map ingin membuat catatan SRV saat mendaftarkan instance dan Anda menggunakan sistem yang memerlukan format SRV tertentu (seperti HAProxy), tentukan yang berikut untuk nama Layanan:
>
> - Mulai nama dengan garis bawah (_), misalnya _exampleservice.
> - Akhiri nama dengan *._protocol*, misalnya. _tcp.
>
> Saat Anda mendaftarkan instance, AWS Cloud Map membuat catatan SRV dan menetapkan nama dengan menggabungkan nama layanan dan nama namespace, misalnya:
> _exampleservice. _tcp.example.com

6. (Opsional) Untuk deskripsi Layanan, masukkan deskripsi untuk layanan. Deskripsi yang Anda masukkan di sini muncul di halaman Layanan dan pada halaman detail untuk setiap layanan.
7. Jika namespace mendukung kueri DNS, di bawah konfigurasi Penemuan layanan, Anda dapat mengonfigurasi kemampuan penemuan di tingkat layanan. Pilih antara mengizinkan panggilan API dan kueri DNS atau hanya panggilan API untuk penemuan instance dalam layanan ini.

> **Note**
>
> Jika Anda memilih panggilan API, tidak AWS Cloud Map akan membuat catatan SRV saat Anda mendaftarkan instance.

Jika Anda memilih API dan DNS, ikuti langkah-langkah ini untuk mengonfigurasi catatan DNS. Anda dapat menambah atau menghapus catatan DNS.

1. Untuk kebijakan Perutean, pilih kebijakan perutean Amazon Route 53 untuk catatan DNS yang AWS Cloud Map dibuat saat Anda mendaftarkan instance. Anda dapat memilih antara routing tertimbang dan routing jawaban Multivalue. Untuk informasi selengkapnya, lihat Kebijakan perutean.

> **Note**
>
> Anda tidak dapat menggunakan konsol untuk mengonfigurasi AWS Cloud Map untuk membuat catatan alias Route 53 saat mendaftarkan instance. Jika Anda ingin membuat catatan alias AWS Cloud Map untuk penyeimbang beban Elastic Load Balancing saat Anda mendaftarkan instance secara terprogram, pilih Perutean tertimbang untuk kebijakan Perutean.

2. Untuk jenis Rekam, pilih jenis catatan DNS yang menentukan apa yang dikembalikan Route 53 sebagai respons terhadap kueri DNS. AWS Cloud Map Untuk informasi selengkapnya, lihat Jenis catatan.
3. Untuk TTL, tentukan nilai numerik untuk menentukan nilai time to live (TTL), dalam hitungan detik, di tingkat layanan. Nilai dari TTL menentukan berapa lama DNS penyelesai cache informasi untuk catatan ini sebelum penyelesai meneruskan permintaan DNS lain untuk Amazon Route 53 untuk mendapatkan pengaturan yang diperbarui.

8. Di bawah Konfigurasi pemeriksaan Kesehatan, untuk opsi pemeriksaan Kesehatan, pilih jenis pemeriksaan kesehatan yang berlaku untuk instance layanan. Anda dapat memilih untuk tidak mengonfigurasi pemeriksaan kesehatan apa pun, atau Anda dapat memilih antara pemeriksaan kesehatan Route 53 atau pemeriksaan kesehatan eksternal untuk instans Anda. Untuk informasi selengkapnya, lihat AWS Cloud Map konfigurasi pemeriksaan kesehatan layanan.

> **Note**
>
> Pemeriksaan kesehatan Route 53 hanya dapat dikonfigurasi untuk layanan di ruang nama DNS publik.

Jika Anda memilih pemeriksaan kesehatan Rute 53, berikan informasi berikut.

1. Untuk ambang kegagalan, berikan angka antara 1 dan 10 yang menentukan jumlah pemeriksaan kesehatan Rute 53 berturut-turut yang harus dilewati atau gagal jika status kesehatannya berubah.
2. Untuk protokol pemeriksaan Kesehatan, pilih metode yang akan digunakan Route 53 untuk memeriksa kesehatan instance layanan.

3. Jika Anda memilih protokol pemeriksaan kesehatan HTTP atau HTTPS, untuk jalur pemeriksaan Kesehatan, berikan jalur yang Anda inginkan untuk diminta Amazon Route 53 saat melakukan pemeriksaan kesehatan. Jalur dapat berupa nilai apapun seperti file `/docs/route53-health-check.html`. Ketika sumber daya sehat, nilai yang dikembalikan adalah kode status HTTP 2xx atau 3xx format. Anda juga dapat menyertakan parameter rangkaian kueri, misalnya, `/welcome.html?language=jp&login=y`. AWS Cloud Map Konsol tersebut secara otomatis menambahkan garis miring (/) karakter.

Untuk informasi lebih lanjut tentang pemeriksaan kesehatan Route 53, lihat Bagaimana Amazon Route 53 Menentukan Apakah Pemeriksaan Kesehatan Sehat di Panduan Pengembang Amazon Route 53.

9. (Opsional) Di bawah Tag, pilih Tambahkan tag lalu tentukan kunci dan nilai untuk menandai namespace Anda. Anda dapat menentukan satu atau lebih tag untuk ditambahkan ke namespace Anda. Tag memungkinkan Anda untuk mengkategorikan AWS sumber daya Anda sehingga Anda dapat lebih mudah mengelolanya. Untuk informasi selengkapnya, lihat Menandai sumber daya Anda AWS Cloud Map.
10. Pilih Buat layanan.

AWS CLI

- Buat layanan dengan `create-service` perintah. Ganti *red* nilai dengan nilai Anda sendiri.

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

Output:

```
{
        "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)

Jika Anda belum Boto3 menginstal, Anda dapat menemukan petunjuk untuk menginstal, mengkonfigurasi, dan menggunakan Boto3 di sini.

1. Impor Boto3 dan gunakan `servicediscovery` sebagai layanan Anda.

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

2. Buat layanan dengan`create_service()`. Ganti *red* nilai dengan nilai Anda sendiri. Untuk informasi selengkapnya, lihat create_service.

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

Contoh keluaran respons

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

Langkah selanjutnya

Setelah membuat layanan, Anda dapat mendaftarkan sumber daya aplikasi Anda sebagai instance layanan yang berisi informasi tentang bagaimana aplikasi Anda dapat menemukan sumber daya. Untuk informasi selengkapnya tentang mendaftarkan instance AWS Cloud Map layanan, lihatMendaftarkan sumber daya sebagai instance AWS Cloud Map layanan.

Anda juga dapat menentukan metadata kustom seperti bobot titik akhir, batas waktu API, dan kebijakan coba lagi sebagai atribut layanan setelah membuat layanan. Untuk informasi selengkapnya, lihat ServiceAttributesdan `UpdateServiceAttributes`di Referensi AWS Cloud Map API.

Memperbarui AWS Cloud Map layanan

Bergantung pada konfigurasi layanan, Anda dapat memperbarui tagnya, ambang kegagalan pemeriksaan kesehatan Route 53, dan time to live (TTL) untuk penyelesai DNS. Untuk memperbarui layanan, lakukan prosedur berikut.

AWS Management Console

1. Masuk ke AWS Management Console dan buka AWS Cloud Map konsol di https://console.aws.amazon.com/cloudmap/.
2. Di panel navigasi, pilih Namespace.
3. Pada halaman Namespaces, pilih namespace tempat layanan dibuat.
4. Pada *namespace-name* halaman Namespace:, pilih layanan yang ingin Anda edit dan pilih Lihat detail.
5. Pada *service-name* halaman Layanan: pilih Edit.

   > **Note**
   >
   > Anda tidak dapat menggunakan alur kerja tombol Edit untuk mengedit nilai layanan yang hanya mengizinkan panggilan API untuk penemuan instance. Namun, Anda dapat menambahkan atau menghapus tag pada *service-name* halaman Layanan:.

6. Pada halaman Edit layanan, di bawah Deskripsi layanan, Anda dapat memperbarui deskripsi yang ditetapkan sebelumnya untuk layanan atau menambahkan deskripsi baru. Anda juga dapat menambahkan tag dan memperbarui TTL untuk penyelesai DNS.
7. Di bawah konfigurasi DNS, untuk TTL, Anda dapat menentukan periode waktu yang diperbarui, dalam hitungan detik, yang menentukan berapa lama DNS resolver informasi cache untuk catatan ini sebelum resolver meneruskan kueri DNS lain ke Amazon Route 53 untuk mendapatkan pengaturan yang diperbarui.
8. Jika Anda telah menyiapkan pemeriksaan kesehatan Route 53, untuk ambang kegagalan, Anda dapat menentukan nomor baru antara 1 dan 10 yang menentukan jumlah pemeriksaan kesehatan Route 53 berturut-turut yang harus dilewati atau gagal oleh instans layanan agar status kesehatannya berubah.
9. Pilih Perbarui layanan.

AWS CLI

- Perbarui layanan dengan `update-service` perintah (ganti *red* nilainya dengan milik Anda sendiri).

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

Output:

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

AWS SDK for Python (Boto3)

1. Jika Anda belum Boto3 menginstal, Anda dapat menemukan petunjuk untuk menginstal, mengkonfigurasi, dan menggunakan Boto3 di sini.
2. Impor Boto3 dan gunakan `servicediscovery` sebagai layanan Anda.

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

3. Perbarui layanan dengan `update_service()` (ganti *red* nilainya dengan milik Anda sendiri).

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

```
    }
)
```

Contoh keluaran respons

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

AWS Cloud Map Layanan daftar di namespace

Untuk melihat daftar layanan yang Anda buat di namespace, lakukan prosedur berikut.

AWS Management Console

1. Masuk ke AWS Management Console dan buka AWS Cloud Map konsol di https://console.aws.amazon.com/cloudmap/.
2. Di panel navigasi, pilih Namespace.
3. Pilih nama namespace yang berisi layanan yang ingin Anda daftarkan. Anda dapat melihat daftar semua layanan di bawah Layanan dan memasukkan nama layanan atau ID di bidang pencarian untuk menemukan layanan tertentu.

AWS CLI

- Daftar layanan dengan `list-services` perintah. Perintah berikut mencantumkan semua layanan dalam namespace menggunakan ID namespace sebagai filter. Ganti *red* nilainya dengan nilai Anda sendiri.

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

AWS SDK for Python (Boto3)

1. Jika Anda belum `Boto3` menginstal, Anda dapat menemukan petunjuk untuk menginstal, mengkonfigurasi, dan menggunakan `Boto3` di sini.
2. Impor `Boto3` dan gunakan `servicediscovery` sebagai layanan Anda.

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

3. Daftar layanan dengan`list_services()`.

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

Contoh keluaran respons

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

Menghapus layanan AWS Cloud Map

Sebelum dapat menghapus layanan, Anda harus membatalkan pendaftaran semua instans layanan yang terdaftar menggunakan layanan. Untuk informasi selengkapnya, lihat Membatalkan pendaftaran instance layanan AWS Cloud Map.

Setelah membatalkan pendaftaran semua instance yang terdaftar menggunakan layanan, lakukan prosedur berikut untuk menghapus layanan.

AWS Management Console

1. Masuk ke AWS Management Console dan buka AWS Cloud Map konsol di https://console.aws.amazon.com/cloudmap/.
2. Di panel navigasi, pilih Namespace.
3. Pilih opsi untuk namespace yang berisi layanan yang ingin Anda hapus.
4. Pada *namespace-name* halaman Namespace:, pilih opsi untuk layanan yang ingin Anda hapus.
5. Pilih Hapus.
6. Mengonfirmasi bahwa Anda ingin menghapus layanan.

AWS CLI

- Hapus layanan dengan `delete-service` perintah (ganti *red* nilainya dengan milik Anda sendiri).

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

AWS SDK for Python (Boto3)

1. Jika Anda belum `Boto3` menginstal, Anda dapat menemukan petunjuk untuk menginstal, mengkonfigurasi, dan menggunakan `Boto3` di sini.
2. Impor `Boto3` dan gunakan `servicediscovery` sebagai layanan Anda.

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

3. Hapus layanan dengan `delete_service()` (ganti *red* nilainya dengan milik Anda).

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

Contoh keluaran respons

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

AWS Cloud Map contoh layanan

Sebuah contoh layanan berisi informasi tentang bagaimana untuk menemukan sumber daya, seperti server web, untuk aplikasi. Setelah mendaftarkan instance, Anda menemukannya dengan menggunakan kueri DNS atau tindakan API. AWS Cloud Map DiscoverInstances Sumber daya yang dapat Anda daftarkan termasuk, tetapi tidak terbatas pada, hal-hal berikut:

- EC2 Contoh Amazon
- Tabel Amazon DynamoDB
- Bucket Amazon S3
- Antrean Amazon Simple Queue Service (Amazon SQS)
- APIs diterapkan di atas Amazon API Gateway

Anda dapat menentukan nilai atribut untuk instance layanan, dan klien dapat menggunakan atribut ini untuk memfilter sumber daya yang AWS Cloud Map dikembalikan. Misalnya, aplikasi dapat meminta sumber daya dalam tahap deployment tertentu, seperti BETA atau PROD. Anda juga dapat menggunakan atribut untuk pembuatan versi.

Prosedur berikut menjelaskan bagaimana Anda dapat mendaftarkan sumber daya dalam aplikasi sebagai instance layanan, melihat daftar instance terdaftar dalam layanan, mengedit parameter instans tertentu, dan membatalkan pendaftaran instance.

Topik

Mendaftarkan sumber daya sebagai instance AWS Cloud Map layanan

Anda dapat mendaftarkan sumber daya aplikasi Anda sebagai instance dalam AWS Cloud Map layanan. Misalnya, anggap Anda telah membuat layanan yang dipanggil `users` untuk semua sumber

daya aplikasi yang mengelola data pengguna. Anda kemudian dapat mendaftarkan tabel DynamoDB yang digunakan untuk menyimpan data pengguna sebagai contoh dalam layanan ini.

> **Note**
>
> Fitur-fitur berikut tidak tersedia di AWS Cloud Map konsol:
>
> - Ketika Anda mendaftar instans layanan menggunakan konsol, Anda tidak dapat membuat catatan alias yang merutekan lalu lintas ke Elastic Load Balancing (ELB). Ketika Anda mendaftar instans, Anda harus menyertakan `AWS_ALIAS_DNS_NAME` atribut. Untuk informasi selengkapnya, lihat RegisterInstance di dalam Referensi API AWS Cloud Map .
> - Jika Anda mendaftarkan instans menggunakan layanan yang menyertakan pemeriksaan kondisi kustom, Anda tidak dapat menentukan status awal untuk pemeriksaan kondisi kustom. Secara default, status awal pemeriksaan kondisi kustom adalah Sehat. Jika Anda ingin status kondisi awal menjadi Tidak sehat, mendaftar instans pemrograman dan termasuk `AWS_INIT_HEALTH_STATUS` atribut. Untuk informasi selengkapnya, lihat RegisterInstance di dalam Referensi API AWS Cloud Map .

Untuk mendaftarkan instance dalam layanan, ikuti langkah-langkah ini.

AWS Management Console

1. Masuk ke AWS Management Console dan buka AWS Cloud Map konsol di https://console.aws.amazon.com/cloudmap/.
2. Di panel navigasi, pilih Namespace.
3. Pada Namespace halaman, pilih namespace yang berisi layanan yang ingin Anda gunakan sebagai templat untuk mendaftar instans layanan.
4. Pada *`namespace-name`* halaman Namespace:, pilih layanan yang ingin Anda gunakan.
5. Pada *`service-name`* halaman Layanan:, pilih Daftar instance layanan.
6. Pada halaman contoh layanan Register, pilih jenis Instance. Bergantung pada konfigurasi penemuan instance namespace, Anda dapat memilih untuk menentukan alamat IP, ID EC2 instans Amazon, atau informasi pengenal lainnya untuk sumber daya yang tidak memiliki alamat IP.

> **Note**
>
> Anda dapat memilih EC2 instance hanya di ruang nama HTTP.

7. Untuk ID contoh Layanan, berikan pengenal yang terkait dengan instance layanan.

> **Note**
>
> Jika Anda ingin memperbarui instance yang ada, berikan pengenal yang terkait dengan instance yang ingin Anda perbarui. Kemudian, gunakan langkah selanjutnya untuk memperbarui nilai dan mendaftarkan ulang instance.

8. Berdasarkan pilihan jenis Instance Anda, lakukan langkah-langkah berikut.

> **Important**
>
> Anda tidak dapat menggunakan `AWS_` awalan (tidak peka huruf besar/kecil) dalam kunci saat menentukan atribut kustom.

Jenis instans	Langkah-langkah	
Alamat IP	a. Di bawah atribut Standar, untuk IPv4alamat, berikan IPv4 alamat, jika ada, tempat aplikasi Anda dapat mengakses sumber daya yang terkait dengan instance layanan ini. b. Untuk IPv6 alamat, berikan alamat IPv6 IP, jika ada, di mana aplikasi Anda dapat mengakses sumber daya yang terkait	

Jenis instans	Langkah-langkah	
	dengan instance layanan ini. c. Untuk Port, tentukan port apa pun yang harus disertakan aplikasi Anda untuk mengakses sumber daya yang terkait dengan instance layanan ini. Port diperlukan ketika layanan menyertak an catatan SRV atau pemeriksaan kesehatan Amazon Route 53. d. (Opsional) Di bawah atribut Kustom, tentukan pasangan kunci-nilai yang ingin Anda kaitkan dengan sumber daya.	
EC2 contoh	a. EC2 Misalnya ID, pilih ID EC2 instance Amazon yang ingin Anda daftarkan sebagai instance AWS Cloud Map layanan. b. (Opsional) Di bawah atribut Kustom, tentukan pasangan kunci-nilai yang ingin Anda kaitkan dengan sumber daya.	

Jenis instans	Langkah-langkah	
Mengidentifikasi informasi untuk sumber daya lain	a. Di bawah atribut Standar, jika konfigurasi layanan menyertakan catatan DNS CNAME, Anda akan melihat bidang CNAME. Untuk CNAME, tentukan nama domain yang ingin Anda kembalikan Route 53 sebagai respons terhadap kueri DNS (misalnya,). `example.com` b. Di bawah Atribut khusus, tentukan informasi pengenal apa pun untuk sumber daya yang bukan alamat IP atau ID EC2 instans Amazon sebagai pasangan nilai kunci. Misalnya, Anda dapat mendaftarkan fungsi Lambda dengan menentukan kunci yang dipanggil `function` dan memberikan nama fungsi Lambda sebagai nilai. Anda juga dapat menentukan kunci yang dipanggil `name` dan memberikan nama yang dapat Anda gunakan untuk penemuan instance terprogram.	

9. Pilih Daftarkan instans layanan.

AWS CLI

- Saat Anda mengirimkan `RegisterInstance` permintaan:
  - Untuk setiap catatan DNS yang Anda tentukan dalam layanan yang ditentukan oleh`ServiceId`, catatan dibuat atau diperbarui di zona yang dihosting yang terkait dengan namespace yang sesuai.
  - Jika layanan termasuk`HealthCheckConfig`, pemeriksaan kesehatan dibuat berdasarkan pengaturan dalam konfigurasi pemeriksaan kesehatan.
  - Setiap pemeriksaan kesehatan dikaitkan dengan masing-masing catatan baru atau yang diperbarui.

  Daftarkan instance layanan dengan `register-instance` perintah (ganti *red* nilainya dengan milik Anda sendiri).

  ```
  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. Jika Anda belum `Boto3` menginstal, Anda dapat menemukan petunjuk untuk menginstal, mengkonfigurasi, dan menggunakan `Boto3` di sini.
2. Impor `Boto3` dan gunakan `servicediscovery` sebagai layanan Anda.

   ```
   import boto3
   client = boto3.client('servicediscovery')
   ```
3. Saat Anda mengirimkan `RegisterInstance` permintaan:
   - Untuk setiap catatan DNS yang Anda tentukan dalam layanan yang ditentukan oleh`ServiceId`, catatan dibuat atau diperbarui di zona yang dihosting yang terkait dengan namespace yang sesuai.
   - Jika layanan termasuk`HealthCheckConfig`, pemeriksaan kesehatan dibuat berdasarkan pengaturan dalam konfigurasi pemeriksaan kesehatan.

- Setiap pemeriksaan kesehatan dikaitkan dengan masing-masing catatan baru atau yang diperbarui.

Daftarkan instance layanan dengan `register_instance()` (ganti *red* nilainya dengan milik Anda sendiri).

```
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)
```

Contoh keluaran respons

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

Daftar contoh AWS Cloud Map layanan

Untuk melihat daftar instans layanan yang Anda terdaftar menggunakan layanan, lakukan prosedur berikut.

AWS Management Console

1. Masuk ke AWS Management Console dan buka AWS Cloud Map konsol di https://console.aws.amazon.com/cloudmap/.
2. Di panel navigasi, pilih Namespace.
3. Pilih nama namespace yang berisi layanan yang ingin Anda daftarkan instans layanan.

4. Pilih nama layanan yang digunakan untuk membuat instans layanan. Anda akan melihat daftar instance di bawah instance Layanan. Anda dapat memasukkan ID instance di bidang pencarian untuk mencantumkan instance tertentu.

AWS CLI

- Daftar instance layanan dengan `list-instances` perintah (ganti *red* nilainya dengan milik Anda sendiri).

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

AWS SDK for Python (Boto3)

1. Jika Anda belum `Boto3` menginstal, Anda dapat menemukan petunjuk untuk menginstal, mengkonfigurasi, dan menggunakan `Boto3` di sini.
2. Impor `Boto3` dan gunakan `servicediscovery` sebagai layanan Anda.

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

3. Daftar instance layanan dengan `list_instances()` (ganti *red* nilainya dengan milik Anda sendiri).

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

Contoh keluaran respons

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

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

Memperbarui instance AWS Cloud Map layanan

Anda dapat memperbarui instans layanan dalam dua cara, tergantung pada nilai yang ingin Anda perbarui:

- Memperbarui nilai apa pun: Jika Anda ingin memperbarui nilai apa pun yang Anda tentukan untuk instance layanan saat Anda mendaftarkannya, termasuk atribut khusus, Anda perlu mendaftarkan ulang instance layanan dan menentukan ulang semua nilai. Ikuti langkah-langkahnyaMendaftarkan sumber daya sebagai instance AWS Cloud Map layanan, tentukan ID instance dari instance layanan yang ada untuk ID instance Layanan.

  Atau, Anda dapat menggunakan `RegisterInstance`API. Anda dapat menentukan ID dari instance dan layanan yang ada menggunakan `ServiceId` parameter `InstanceId` dan dan menentukan ulang nilai lainnya.
- Perbarui hanya atribut kustom: Jika Anda ingin memperbarui hanya atribut kustom untuk instans layanan, Anda tidak perlu mendaftarkan ulang instans. Anda dapat memperbarui hanya nilai-nilai tersebut. Lihat Memperbarui atribut kustom untuk instance layanan.

Memperbarui atribut kustom untuk instance layanan

Untuk memperbarui hanya atribut kustom untuk instans layanan

1. Masuk ke AWS Management Console dan buka AWS Cloud Map konsol di https://console.aws.amazon.com/cloudmap/.
2. Di panel navigasi, pilih Namespace.
3. Pada Namespace halaman, pilih namespace yang berisi layanan yang Anda awalnya digunakan untuk mendaftar instans layanan.
4. Pada *`namespace-name`* halaman Namespace:, pilih layanan yang Anda gunakan untuk mendaftarkan instance layanan.

5. Pada *service-name* halaman Service:, pilih nama instance layanan yang ingin Anda perbarui.
6. Di atribut kustom bagian, pilih Mengedit.
7. Pada contoh layanan Edit: ***instance-name*** halaman, tambahkan, hapus, atau perbarui atribut kustom. Anda dapat memperbarui kedua kunci dan nilai-nilai untuk atribut yang ada.
8. Pilih Perbarui instans layanan.

Membatalkan pendaftaran instance layanan AWS Cloud Map

Sebelum dapat menghapus layanan, Anda harus membatalkan pendaftaran semua instans layanan yang terdaftar menggunakan layanan.

Untuk membatalkan pendaftaran instans layanan, lakukan prosedur berikut.

AWS Management Console

1. Masuk ke AWS Management Console dan buka AWS Cloud Map konsol di https://console.aws.amazon.com/cloudmap/.
2. Di panel navigasi, pilih Namespace.
3. Pilih opsi untuk namespace yang berisi contoh layanan yang ingin Anda batalkan pendaftarannya.
4. Pada *namespace-name* halaman Namespace:, pilih layanan yang Anda gunakan untuk mendaftarkan instance layanan.
5. Pada *service-name* halaman Service:, pilih instance layanan yang ingin Anda deregister.
6. Pilih Batalkan pendaftaran.
7. Pastikan bahwa Anda ingin membatalkan pendaftaran instans layanan.

AWS CLI

- Deregister instance layanan dengan `deregister-instance` perintah (ganti *red* nilai dengan milik Anda sendiri). Perintah ini menghapus catatan DNS Amazon Route 53 dan pemeriksaan kesehatan apa pun yang AWS Cloud Map dibuat untuk instance yang ditentukan.

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

```
    --instance-id myservice-53
```

AWS SDK for Python (Boto3)

1. Jika Anda belum `Boto3` menginstal, Anda dapat menemukan petunjuk untuk menginstal, mengkonfigurasi, dan menggunakan `Boto3` di sini.
2. Impor `Boto3` dan gunakan `servicediscovery` sebagai layanan Anda.

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

3. Deregister instance layanan dengan `deregister-instance()` (ganti *red* nilai dengan milik Anda sendiri). Perintah ini menghapus catatan DNS Amazon Route 53 dan pemeriksaan kesehatan apa pun yang AWS Cloud Map dibuat untuk instance yang ditentukan.

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

Contoh keluaran respons

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

Keamanan di AWS Cloud Map

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. Model tanggung jawab bersama menggambarkan hal ini sebagai keamanan dari cloud dan keamanan di cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara berkala menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari Program kepatuhan AWS. Untuk mempelajari tentang program kepatuhan yang berlaku AWS Cloud Map, lihat AWS Layanan dalam Lingkup berdasarkan Program Kepatuhan.
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, yang mencakup sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi berikut membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan AWS Cloud Map. Topik berikut menunjukkan cara mengonfigurasi AWS Cloud Map untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga belajar cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan AWS Cloud Map sumber daya Anda.

Topik

Identity and Access Management untuk AWS Cloud Map

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya. AWS Cloud Map IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan. AWS Cloud Map

Pengguna layanan — Jika Anda menggunakan AWS Cloud Map layanan untuk melakukan pekerjaan Anda, maka administrator Anda memberi Anda kredensi dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak AWS Cloud Map fitur untuk melakukan pekerjaan Anda, Anda mungkin memerlukan izin tambahan. Memahami cara mengelola akses dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di AWS Cloud Map, lihat Memecahkan masalah AWS Cloud Map identitas dan akses.

Administrator layanan — Jika Anda bertanggung jawab atas AWS Cloud Map sumber daya di perusahaan Anda, Anda mungkin memiliki akses penuh ke AWS Cloud Map. Tugas Anda adalah menentukan AWS Cloud Map fitur dan sumber daya mana yang harus diakses pengguna layanan Anda. Kemudian, Anda harus mengirimkan permintaan kepada administrator IAM untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep dasar IAM. Untuk mempelajari lebih lanjut tentang bagaimana perusahaan Anda dapat menggunakan IAM AWS Cloud Map, lihatBagaimana AWS Cloud Map bekerja dengan IAM.

Administrator IAM – Jika Anda adalah administrator IAM, Anda mungkin ingin belajar dengan lebih detail tentang cara Anda menulis kebijakan untuk mengelola akses ke AWS Cloud Map. Untuk melihat contoh kebijakan AWS Cloud Map berbasis identitas yang dapat Anda gunakan di IAM, lihat. Contoh kebijakan berbasis identitas untuk AWS Cloud Map

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensi identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk AWS sebagai identitas federasi dengan menggunakan kredensil yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (IAM Identity Center), autentikasi masuk tunggal perusahaan Anda, dan kredensi Google atau Facebook Anda adalah contoh identitas federasi. Saat Anda masuk sebagai identitas terfederasi, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan peran IAM. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal AWS akses. Untuk informasi selengkapnya tentang masuk AWS, lihat Cara masuk ke Panduan AWS Sign-In Pengguna Anda Akun AWS.

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensil Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Guna mengetahui informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, lihat AWS Signature Version 4 untuk permintaan API dalam Panduan Pengguna IAM.

Apa pun metode autentikasi yang digunakan, Anda mungkin diminta untuk menyediakan informasi keamanan tambahan. Misalnya, AWS merekomendasikan agar Anda menggunakan otentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari selengkapnya, lihat Autentikasi multi-faktor dalam Panduan Pengguna AWS IAM Identity Center dan Autentikasi multi-faktor AWS di IAM dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Identitas ini disebut pengguna Akun AWS root dan diakses dengan masuk dengan alamat email dan kata sandi yang Anda gunakan

untuk membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar lengkap tugas yang mengharuskan Anda masuk sebagai pengguna root, lihat Tugas yang memerlukan kredensial pengguna root dalam Panduan Pengguna IAM.

Identitas gabungan

Sebagai praktik terbaik, mewajibkan pengguna manusia, termasuk pengguna yang memerlukan akses administrator, untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS dengan menggunakan kredensi sementara.

Identitas federasi adalah pengguna dari direktori pengguna perusahaan Anda, penyedia identitas web, direktori Pusat Identitas AWS Directory Service, atau pengguna mana pun yang mengakses Layanan AWS dengan menggunakan kredensil yang disediakan melalui sumber identitas. Ketika identitas federasi mengakses Akun AWS, mereka mengambil peran, dan peran memberikan kredensi sementara.

Untuk manajemen akses terpusat, kami sarankan Anda menggunakan AWS IAM Identity Center. Anda dapat membuat pengguna dan grup di Pusat Identitas IAM, atau Anda dapat menghubungkan dan menyinkronkan ke sekumpulan pengguna dan grup di sumber identitas Anda sendiri untuk digunakan di semua aplikasi Akun AWS dan aplikasi Anda. Untuk informasi tentang Pusat Identitas IAM, lihat Apakah itu Pusat Identitas IAM? dalam Panduan Pengguna AWS IAM Identity Center .

Pengguna dan grup IAM

Pengguna IAM adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, kami merekomendasikan untuk mengandalkan kredensial sementara, bukan membuat pengguna IAM yang memiliki kredensial jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan tertentu yang memerlukan kredensial jangka panjang dengan pengguna IAM, kami merekomendasikan Anda merotasi kunci akses. Untuk informasi selengkapnya, lihat Merotasi kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensial jangka panjang dalam Panduan Pengguna IAM.

Grup IAM adalah identitas yang menentukan sekumpulan pengguna IAM. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin bagi beberapa pengguna sekaligus. Grup mempermudah manajemen izin untuk sejumlah besar pengguna sekaligus. Misalnya, Anda dapat meminta kelompok untuk menyebutkan IAMAdmins dan memberikan izin kepada grup tersebut untuk mengelola sumber daya IAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk mempelajari selengkapnya, lihat Kasus penggunaan untuk pengguna IAM dalam Panduan Pengguna IAM.

Peran IAM

Peran IAM adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus. Peran ini mirip dengan pengguna IAM, tetapi tidak terkait dengan orang tertentu. Untuk mengambil peran IAM sementara AWS Management Console, Anda dapat beralih dari pengguna ke peran IAM (konsol). Anda dapat mengambil peran dengan memanggil operasi AWS CLI atau AWS API atau dengan menggunakan URL kustom. Untuk informasi selengkapnya tentang cara menggunakan peran, lihat Metode untuk mengambil peran dalam Panduan Pengguna IAM.

Peran IAM dengan kredensial sementara berguna dalam situasi berikut:

- Akses pengguna terfederasi – Untuk menetapkan izin ke identitas terfederasi, Anda membuat peran dan menentukan izin untuk peran tersebut. Ketika identitas terfederasi mengautentikasi, identitas tersebut terhubung dengan peran dan diberi izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat Buat peran untuk penyedia identitas pihak ketiga dalam Panduan Pengguna IAM. Jika menggunakan Pusat Identitas IAM, Anda harus mengonfigurasi set izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah identitas tersebut diautentikasi, Pusat Identitas IAM akan mengorelasikan set izin ke peran dalam IAM. Untuk informasi tentang set izin, lihat Set izin dalam Panduan Pengguna AWS IAM Identity Center .
- Izin pengguna IAM sementara – Pengguna atau peran IAM dapat mengambil peran IAM guna mendapatkan berbagai izin secara sementara untuk tugas tertentu.
- Akses lintas akun – Anda dapat menggunakan peran IAM untuk mengizinkan seseorang (prinsipal tepercaya) di akun lain untuk mengakses sumber daya di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, dengan beberapa Layanan AWS, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan peran sebagai proxy). Untuk mempelajari perbedaan antara peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Akses sumber daya lintas akun di IAM dalam Panduan Pengguna IAM.
- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Misalnya, saat Anda melakukan panggilan dalam suatu layanan, biasanya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Sebuah layanan mungkin

melakukannya menggunakan izin prinsipal yang memanggil, menggunakan peran layanan, atau peran terkait layanan.

- Sesi akses teruskan (FAS) — Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat Sesi akses maju.
- Peran layanan – Peran layanan adalah peran IAM yang dijalankan oleh layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Buat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.
- Peran terkait layanan — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke peran layanan. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

- Aplikasi yang berjalan di Amazon EC2 — Anda dapat menggunakan peran IAM untuk mengelola kredensi sementara untuk aplikasi yang berjalan pada EC2 instance dan membuat AWS CLI atau AWS permintaan API. Ini lebih baik untuk menyimpan kunci akses dalam EC2 instance. Untuk menetapkan AWS peran ke EC2 instance dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instans yang dilampirkan ke instance. Profil instance berisi peran dan memungkinkan program yang berjalan pada EC2 instance untuk mendapatkan kredensi sementara. Untuk informasi selengkapnya, lihat Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan di EC2 instans Amazon di Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan menentukan apakah permintaan

diizinkan atau ditolak. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan isi dokumen kebijakan JSON, lihat Gambaran umum kebijakan JSON dalam Panduan Pengguna IAM.

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk melakukan operasinya. Misalnya, anggaplah Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan tersebut bisa mendapatkan informasi peran dari AWS Management Console, API AWS CLI, atau AWS API.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan yang dikelola. Kebijakan inline disematkan langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran dalam. Akun AWS Kebijakan AWS terkelola mencakup kebijakan terkelola dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan yang dikelola atau kebijakan inline, lihat Pilih antara kebijakan yang dikelola dan kebijakan inline dalam Panduan Pengguna IAM.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya,

administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan prinsipal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola dari IAM dalam kebijakan berbasis sumber daya.

Daftar kontrol akses (ACLs)

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung. ACLs Untuk mempelajari selengkapnya ACLs, lihat Ringkasan daftar kontrol akses (ACL) di Panduan Pengembang Layanan Penyimpanan Sederhana Amazon.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang kurang umum. Jenis-jenis kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda oleh jenis kebijakan yang lebih umum.

- Batasan izin – Batasan izin adalah fitur lanjutan tempat Anda mengatur izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas ke entitas IAM (pengguna IAM atau peran IAM). Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batasan izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang `Principal` tidak dibatasi oleh batasan izin. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya tentang batasan izin, lihat Batasan izin untuk entitas IAM dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCPs) — SCPs adalah kebijakan JSON yang menentukan izin maksimum untuk organisasi atau unit organisasi (OU) di. AWS Organizations AWS Organizations adalah layanan untuk mengelompokkan dan mengelola secara terpusat beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur dalam suatu organisasi, maka Anda dapat menerapkan kebijakan kontrol layanan (SCPs) ke salah satu atau semua akun Anda. SCP membatasi izin untuk entitas di akun anggota, termasuk masing-masing. Pengguna root akun AWS

Untuk informasi selengkapnya tentang Organizations dan SCPs, lihat Kebijakan kontrol layanan di Panduan AWS Organizations Pengguna.

- Kebijakan kontrol sumber daya (RCPs) — RCPs adalah kebijakan JSON yang dapat Anda gunakan untuk menetapkan izin maksimum yang tersedia untuk sumber daya di akun Anda tanpa memperbarui kebijakan IAM yang dilampirkan ke setiap sumber daya yang Anda miliki. RCP membatasi izin untuk sumber daya di akun anggota dan dapat memengaruhi izin efektif untuk identitas, termasuk Pengguna root akun AWS, terlepas dari apakah itu milik organisasi Anda. Untuk informasi selengkapnya tentang Organizations dan RCPs, termasuk daftar dukungan Layanan AWS tersebut RCPs, lihat Kebijakan kontrol sumber daya (RCPs) di Panduan AWS Organizations Pengguna.
- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter ketika Anda membuat sesi sementara secara programatis untuk peran atau pengguna terfederasi. Izin sesi yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya, lihat Kebijakan sesi dalam Panduan Pengguna IAM.

Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat Logika evaluasi kebijakan di Panduan Pengguna IAM.

Bagaimana AWS Cloud Map bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses AWS Cloud Map, pelajari fitur IAM yang tersedia untuk digunakan. AWS Cloud Map

Fitur IAM	AWS Cloud Map dukungan
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Tidak
Tindakan kebijakan	Ya

Fitur IAM	AWS Cloud Map dukungan
Sumber daya kebijakan	Ya
kunci-kunci persyaratan kebijakan (spesifik layanan)	Ya
ACLs	Tidak
ABAC (tanda dalam kebijakan)	Ya
Kredensial sementara	Ya
Sesi akses teruskan (FAS)	Ya
Peran layanan	Tidak
Peran terkait layanan	Tidak

Untuk mendapatkan tampilan tingkat tinggi tentang cara AWS Cloud Map dan AWS layanan lain bekerja dengan sebagian besar fitur IAM, lihat AWS layanan yang bekerja dengan IAM di Panduan Pengguna IAM.

Kebijakan berbasis identitas untuk AWS Cloud Map

Mendukung kebijakan berbasis identitas: Ya

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta kondisi yang menjadi dasar dikabulkan atau ditolaknya tindakan tersebut. Anda tidak dapat menentukan secara spesifik prinsipal dalam sebuah kebijakan berbasis identitas karena prinsipal berlaku bagi pengguna atau peran yang melekat kepadanya. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, lihat Referensi elemen kebijakan JSON IAM dalam Panduan Pengguna IAM.

Contoh kebijakan berbasis identitas untuk AWS Cloud Map

Untuk melihat contoh kebijakan AWS Cloud Map berbasis identitas, lihat. Contoh kebijakan berbasis identitas untuk AWS Cloud Map

Kebijakan berbasis sumber daya dalam AWS Cloud Map

Mendukung kebijakan berbasis sumber daya: Tidak

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan prinsipal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan secara spesifik seluruh akun atau entitas IAM di akun lain sebagai prinsipal dalam kebijakan berbasis sumber daya. Menambahkan prinsipal akun silang ke kebijakan berbasis sumber daya hanya setengah dari membangun hubungan kepercayaan. Ketika prinsipal dan sumber daya berbeda Akun AWS, administrator IAM di akun tepercaya juga harus memberikan izin entitas utama (pengguna atau peran) untuk mengakses sumber daya. Mereka memberikan izin dengan melampirkan kebijakan berbasis identitas kepada entitas. Namun, jika kebijakan berbasis sumber daya memberikan akses ke principal dalam akun yang sama, tidak diperlukan kebijakan berbasis identitas tambahan. Untuk informasi selengkapnya, lihat Akses sumber daya lintas akun di IAM dalam Panduan Pengguna IAM.

Tindakan kebijakan untuk AWS Cloud Map

Mendukung tindakan kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan kebijakan biasanya memiliki

nama yang sama dengan operasi AWS API terkait. Ada beberapa pengecualian, misalnya tindakan hanya izin yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Sertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Untuk melihat daftar AWS Cloud Map tindakan, lihat Tindakan yang ditentukan oleh AWS Cloud Map dalam Referensi Otorisasi Layanan.

Tindakan kebijakan AWS Cloud Map menggunakan awalan berikut sebelum tindakan:

```
servicediscovery
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan tersebut dengan koma.

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

Untuk melihat contoh kebijakan AWS Cloud Map berbasis identitas, lihat. Contoh kebijakan berbasis identitas untuk AWS Cloud Map

Sumber daya kebijakan untuk AWS Cloud Map

Mendukung sumber daya kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Anda dapat melakukan ini untuk tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, misalnya operasi pencantuman, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

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

Untuk melihat daftar jenis sumber daya dan jenis AWS Cloud Map sumber daya ARNs, lihat Sumber daya yang ditentukan oleh AWS Cloud Map dalam Referensi Otorisasi Layanan. Untuk mempelajari tindakan yang dapat menentukan ARN setiap sumber daya, lihat Tindakan yang ditentukan oleh AWS Cloud Map.

Untuk melihat contoh kebijakan AWS Cloud Map berbasis identitas, lihat. Contoh kebijakan berbasis identitas untuk AWS Cloud Map

Kunci kondisi kebijakan untuk AWS Cloud Map

Mendukung kunci kondisi kebijakan khusus layanan: Yes

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Condition` (atau blok `Condition`) akan memungkinkan Anda menentukan kondisi yang menjadi dasar suatu pernyataan berlaku. Elemen `Condition` bersifat opsional. Anda dapat membuat ekspresi bersyarat yang menggunakan operator kondisi, misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta.

Jika Anda menentukan beberapa elemen `Condition` dalam sebuah pernyataan, atau beberapa kunci dalam elemen `Condition` tunggal, maka AWS akan mengevaluasinya menggunakan operasi `AND` logis. Jika Anda menentukan beberapa nilai untuk satu kunci kondisi, AWS mengevaluasi kondisi menggunakan `OR` operasi logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Sebagai contoh, Anda dapat memberikan izin kepada pengguna IAM untuk mengakses sumber daya hanya jika izin tersebut mempunyai tanda yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, lihat Elemen kebijakan IAM: variabel dan tanda dalam Panduan Pengguna IAM.

AWS mendukung kunci kondisi global dan kunci kondisi khusus layanan. Untuk melihat semua kunci kondisi AWS global, lihat kunci konteks kondisi AWS global di Panduan Pengguna IAM.

Untuk melihat daftar kunci AWS Cloud Map kondisi, lihat Kunci kondisi untuk AWS Cloud Map dalam Referensi Otorisasi Layanan. Untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi, lihat Tindakan yang ditentukan oleh AWS Cloud Map.

AWS Cloud Map mendukung kunci kondisi khusus layanan berikut yang dapat Anda gunakan untuk menyediakan pemfilteran halus untuk kebijakan IAM Anda.

`servicediscovery:NamespaceArn`

Sebuah filter yang memungkinkan Anda mendapatkan objek dengan menentukan Amazon Resource Name (ARN) untuk namespace terkait.

`servicediscovery:NamespaceName`

Sebuah filter yang memungkinkan Anda mendapatkan objek dengan menentukan nama namespace terkait.

`servicediscovery:ServiceArn`

Sebuah filter yang memungkinkan Anda mendapatkan objek dengan menentukan Amazon Resource Name (ARN) untuk layanan terkait.

`servicediscovery:ServiceName`

Sebuah filter yang memungkinkan Anda mendapatkan objek dengan menentukan nama layanan terkait.

Untuk melihat contoh kebijakan AWS Cloud Map berbasis identitas, lihat. Contoh kebijakan berbasis identitas untuk AWS Cloud Map

ACLs di AWS Cloud Map

Mendukung ACLs: Tidak

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan JSON.

ABAC dengan AWS Cloud Map

Mendukung ABAC (tanda dalam kebijakan): Ya

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut. Dalam AWS, atribut ini disebut tag. Anda dapat melampirkan tag ke entitas IAM (pengguna

atau peran) dan ke banyak AWS sumber daya. Penandaan ke entitas dan sumber daya adalah langkah pertama dari ABAC. Kemudian rancanglah kebijakan ABAC untuk mengizinkan operasi ketika tanda milik prinsipal cocok dengan tanda yang ada di sumber daya yang ingin diakses.

ABAC sangat berguna di lingkungan yang berkembang dengan cepat dan berguna di situasi saat manajemen kebijakan menjadi rumit.

Untuk mengendalikan akses berdasarkan tanda, berikan informasi tentang tanda di elemen kondisi dari kebijakan menggunakan kunci kondisi `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci kondisi untuk setiap jenis sumber daya, nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci kondisi untuk hanya beberapa jenis sumber daya, nilainya adalah Parsial.

Untuk informasi selengkapnya tentang ABAC, lihat Tentukan izin dengan otorisasi ABAC dalam Panduan Pengguna IAM. Untuk melihat tutorial yang menguraikan langkah-langkah pengaturan ABAC, lihat Menggunakan kontrol akses berbasis atribut (ABAC) dalam Panduan Pengguna IAM.

Menggunakan kredensi sementara dengan AWS Cloud Map

Mendukung kredensial sementara: Ya

Beberapa Layanan AWS tidak berfungsi saat Anda masuk menggunakan kredensi sementara. Untuk informasi tambahan, termasuk yang Layanan AWS bekerja dengan kredensi sementara, lihat Layanan AWS yang bekerja dengan IAM di Panduan Pengguna IAM.

Anda menggunakan kredensi sementara jika Anda masuk AWS Management Console menggunakan metode apa pun kecuali nama pengguna dan kata sandi. Misalnya, ketika Anda mengakses AWS menggunakan tautan masuk tunggal (SSO) perusahaan Anda, proses tersebut secara otomatis membuat kredensil sementara. Anda juga akan secara otomatis membuat kredensial sementara ketika Anda masuk ke konsol sebagai seorang pengguna lalu beralih peran. Untuk informasi selengkapnya tentang peralihan peran, lihat Beralih dari pengguna ke peran IAM (konsol) dalam Panduan Pengguna IAM.

Anda dapat membuat kredenal sementara secara manual menggunakan API AWS CLI atau AWS . Anda kemudian dapat menggunakan kredensi sementara tersebut untuk mengakses. AWS AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensi sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat Kredensial keamanan sementara di IAM.

Teruskan sesi akses untuk AWS Cloud Map

Mendukung sesi akses maju (FAS): Ya

Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat Sesi akses maju.

Peran layanan untuk AWS Cloud Map

Mendukung peran layanan: Tidak

Peran layanan adalah peran IAM yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Buat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.

> **Warning**
>
> Mengubah izin untuk peran layanan dapat merusak fungsionalitas AWS Cloud Map . Edit peran layanan hanya jika AWS Cloud Map memberikan panduan untuk melakukannya.

Peran terkait layanan untuk AWS Cloud Map

Mendukung peran terkait layanan: Tidak

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

Untuk detail tentang pembuatan atau manajemen peran terkait layanan, lihat Layanan AWS yang berfungsi dengan IAM. Cari layanan dalam tabel yang memiliki `Yes` di kolom Peran terkait layanan. Pilih tautan Ya untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Contoh kebijakan berbasis identitas untuk AWS Cloud Map

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau mengubah sumber daya AWS Cloud Map . Mereka juga tidak dapat melakukan tugas dengan menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), atau AWS API. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM dengan menggunakan contoh dokumen kebijakan JSON ini, lihat Membuat kebijakan IAM (konsol) di Panduan Pengguna IAM.

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh AWS Cloud Map, termasuk format ARNs untuk setiap jenis sumber daya, lihat Kunci tindakan, sumber daya, dan kondisi AWS Cloud Map dalam Referensi Otorisasi Layanan.

Topik

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus AWS Cloud Map sumber daya di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan

yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat Kebijakan yang dikelola AWS atau Kebijakan yang dikelola AWS untuk fungsi tugas dalam Panduan Pengguna IAM.

- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk mengajukan izin, lihat Kebijakan dan izin dalam IAM dalam Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat Elemen kebijakan JSON IAM: Kondisi dalam Panduan Pengguna IAM.
- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat Validasi kebijakan dengan IAM Access Analyzer dalam Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat Amankan akses API dengan MFA dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat Praktik terbaik keamanan di IAM dalam Panduan Pengguna IAM.

Menggunakan konsol AWS Cloud Map

Untuk mengakses AWS Cloud Map konsol, Anda harus memiliki set izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang AWS Cloud Map sumber

daya di Anda Akun AWS. Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna atau peran) dengan kebijakan tersebut.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai gantinya, izinkan akses hanya ke tindakan yang sesuai dengan operasi API yang coba mereka lakukan.

Untuk memastikan bahwa pengguna dan peran masih dapat menggunakan AWS Cloud Map konsol, lampirkan juga kebijakan AWS Cloud Map *ConsoleAccess* atau *ReadOnly* AWS terkelola ke entitas. Untuk informasi selengkapnya, lihat Menambah izin untuk pengguna dalam Panduan Pengguna IAM.

AWS Cloud Map contoh akses konsol

Untuk memberikan akses penuh ke AWS Cloud Map konsol, Anda memberikan izin dalam kebijakan izin berikut:

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":"*"
```

```
      }
   ]
}
```

Inilah mengapa izin diperlukan:

`servicediscovery:*`

Memungkinkan Anda melakukan semua AWS Cloud Map tindakan.

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

Memungkinkan AWS Cloud Map mengelola zona yang dihosting saat Anda membuat dan menghapus ruang nama DNS publik dan pribadi.

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

Memungkinkan AWS Cloud Map mengelola pemeriksaan kesehatan saat Anda menyertakan pemeriksaan kesehatan Amazon Route 53 saat Anda membuat layanan.

`ec2:DescribeVpcs` dan **`ec2:DescribeRegions`**

Biarkan AWS Cloud Map mengelola zona yang dihosting pribadi.

Memungkinkan AWS Cloud Map pengguna untuk melihat izin mereka sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan pengguna IAM melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau menggunakan API atau secara terprogram. AWS CLI 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": "*"
        }
    ]
}
```

Izinkan akses baca ke semua AWS Cloud Map sumber daya

Kebijakan izin berikut memberi akses hanya-baca pengguna ke semua AWS Cloud Map Sumber Daya:

JSON

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

```
      }
   ]
}
```

AWS Cloud Map contoh layanan

Contoh berikut menunjukkan kebijakan izin yang memberikan izin kepada pengguna untuk mendaftar, membatalkan pendaftaran, dan menemukan instance layanan. `Sid`, atau ID pernyataan, adalah opsional:

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": "*"
      }
   ]
}
```

Kebijakan memberikan izin untuk tindakan yang diperlukan untuk daftar dan mengelola instans layanan. Izin Route 53 diperlukan jika Anda menggunakan ruang nama DNS publik atau pribadi

karena AWS Cloud Map membuat, memperbarui, dan menghapus catatan Route 53 dan pemeriksaan kesehatan saat Anda mendaftar dan membatalkan pendaftaran instance. Karakter wildcard (*) dalam `Resource` memberikan akses ke semua AWS Cloud Map instance, dan catatan Route 53 serta pemeriksaan kesehatan yang dimiliki oleh akun saat ini. AWS

Buat contoh AWS Cloud Map layanan

Saat menambahkan kebijakan izin untuk mengizinkan identitas IAM membuat AWS Cloud Map layanan, Anda harus menentukan Nama Sumber Daya Amazon (ARN) dari AWS Cloud Map namespace dan layanan di bidang sumber daya. ARN mencakup Region, ID akun, dan ID namespace. Karena Anda belum tahu apa ID layanan layanan tersebut, kami sarankan menggunakan wildcard. Berikut ini adalah contoh cuplikan kebijakan.

JSON

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

Buat AWS Cloud Map contoh ruang nama

Kebijakan izin berikut memungkinkan pengguna membuat semua jenis ruang AWS Cloud Map nama:

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 kebijakan terkelola untuk AWS Cloud Map

Kebijakan AWS terkelola adalah kebijakan mandiri yang dibuat dan dikelola oleh AWS. AWS Kebijakan terkelola dirancang untuk memberikan izin bagi banyak kasus penggunaan umum sehingga Anda dapat mulai menetapkan izin kepada pengguna, grup, dan peran.

Perlu diingat bahwa kebijakan AWS terkelola mungkin tidak memberikan izin hak istimewa paling sedikit untuk kasus penggunaan spesifik Anda karena tersedia untuk digunakan semua pelanggan. AWS Kami menyarankan Anda untuk mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola pelanggan yang khusus untuk kasus penggunaan Anda.

Anda tidak dapat mengubah izin yang ditentukan dalam kebijakan AWS terkelola. Jika AWS memperbarui izin yang ditentukan dalam kebijakan AWS terkelola, pemutakhiran akan memengaruhi semua identitas utama (pengguna, grup, dan peran) yang dilampirkan kebijakan tersebut. AWS kemungkinan besar akan memperbarui kebijakan AWS terkelola saat baru Layanan AWS diluncurkan atau operasi API baru tersedia untuk layanan yang ada.

Untuk informasi selengkapnya, lihat Kebijakan terkelola AWS dalam Panduan Pengguna IAM.

AWS kebijakan terkelola: AWSCloud MapDiscoverInstanceAccess

Anda dapat melampirkan `AWSCloudMapDiscoverInstanceAccess` ke entitas IAM Anda. Menyediakan akses ke AWS Cloud Map Discovery API.

Untuk melihat izin kebijakan ini, lihat AWSCloudMapDiscoverInstanceAccessdi Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AWSCloud MapReadOnlyAccess

Anda dapat melampirkan `AWSCloudMapReadOnlyAccess` ke entitas IAM Anda. Memberikan akses hanya-baca ke semua tindakan. AWS Cloud Map

Untuk melihat izin kebijakan ini, lihat AWSCloudMapReadOnlyAccessdi Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AWSCloud MapRegisterInstanceAccess

Anda dapat melampirkan `AWSCloudMapRegisterInstanceAccess` ke entitas IAM Anda. Memberikan akses hanya-baca ke ruang nama dan layanan serta memberikan izin untuk mendaftar dan membatalkan pendaftaran instance layanan.

Untuk melihat izin kebijakan ini, lihat AWSCloudMapRegisterInstanceAccessdi Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AWSCloud MapFullAccess

Anda dapat melampirkan `AWSCloudMapFullAccess` ke entitas IAM Anda. Menyediakan akses penuh ke semua AWS Cloud Map tindakan

Untuk melihat izin kebijakan ini, lihat AWSCloudMapFullAccessdi Referensi Kebijakan AWS Terkelola.

AWS Cloud Map pembaruan kebijakan AWS terkelola

Lihat detail tentang pembaruan kebijakan AWS terkelola AWS Cloud Map sejak layanan ini mulai melacak perubahan ini. Untuk peringatan otomatis tentang perubahan, berlangganan umpan RSS di halaman riwayat AWS Cloud Map dokumen.

Perubahan	Deskripsi	Tanggal
AWSCloudMapDiscoverInstanceAccess, AWSCloudMapRegisterInstanceAccess, AWSCloudMapReadOnlyAccess— Pembaruan kebijakan yang ada.	AWS Cloud Map memperbarui kebijakan ini untuk menyediakan akses ke operasi AWS Cloud Map `DiscoverInstanceRevision` API baru.	15 Agustus 2023

AWS Cloud Map Referensi izin API

Saat menyiapkan kontrol akses dan menulis kebijakan izin yang dapat dilampirkan ke identitas IAM (kebijakan berbasis identitas), Anda dapat menggunakan daftar berikut sebagai referensi. Daftar ini mencakup setiap tindakan AWS Cloud Map API dan tindakan yang harus Anda berikan akses izin. Anda menentukan tindakan di `Action` bidang untuk kebijakan tersebut. Untuk detail tentang nilai sumber daya yang harus Anda tentukan di `Resource` bidang atau kebijakan IAM, lihat Kunci tindakan, sumber daya, dan kondisi AWS Cloud Map di Referensi Otorisasi Layanan.

Anda dapat menggunakan AWS Cloud Map kunci kondisi khusus dalam kebijakan IAM Anda untuk beberapa operasi. Untuk informasi selengkapnya, lihat Kunci kondisi untuk AWS Cloud Map Referensi Otorisasi Layanan.

Untuk menentukan tindakan, gunakan `servicediscovery` prefiks diikuti dengan nama tindakan API, misalnya, `servicediscovery:CreatePublicDnsNamespace` dan `route53:CreateHostedZone`.

Izin yang diperlukan untuk tindakan AWS Cloud Map

CreateHttpNamespace

Izin yang diperlukan (tindakan API):

- `servicediscovery:CreateHttpNamespace`

CreatePrivateDnsNamespace

Izin yang diperlukan (tindakan API):

- `servicediscovery:CreatePrivateDnsNamespace`
- `route53:CreateHostedZone`

- `route53:GetHostedZone`
- `route53:ListHostedZonesByName`
- `ec2:DescribeVpcs`
- `ec2:DescribeRegions`

CreatePublicDnsNamespace

Izin yang diperlukan (tindakan API):

- `servicediscovery:CreatePublicDnsNamespace`
- `route53:CreateHostedZone`
- `route53:GetHostedZone`
- `route53:ListHostedZonesByName`

CreateService

Izin yang Diperlukan (Tindakan API): `servicediscovery:CreateService`

DeleteNamespace

Izin yang diperlukan (tindakan API):

- `servicediscovery:DeleteNamespace`

DeleteService

Izin yang Diperlukan (Tindakan API): `servicediscovery:DeleteService`

DeleteServiceAttributes

Izin yang Diperlukan (Tindakan API): `servicediscovery:DeleteServiceAttributes`

DeregisterInstance

Izin yang diperlukan (tindakan API):

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

DiscoverInstances

Izin yang Diperlukan (Tindakan API): `servicediscovery:DiscoverInstances`

GetInstance

Izin yang Diperlukan (Tindakan API): `servicediscovery:GetInstance`

GetInstancesHealthStatus

Izin yang Diperlukan (Tindakan API): `servicediscovery:GetInstancesHealthStatus`

GetNamespace

Izin yang Diperlukan (Tindakan API): `servicediscovery:GetNamespace`

GetOperation

Izin yang Diperlukan (Tindakan API): `servicediscovery:GetOperation`

GetService

Izin yang Diperlukan (Tindakan API): `servicediscovery:GetService`

GetServiceAttributes

Izin yang Diperlukan (Tindakan API): `servicediscovery:GetServiceAttributes`

ListInstances

Izin yang Diperlukan (Tindakan API): `servicediscovery:ListInstances`

ListNamespaces

Izin yang Diperlukan (Tindakan API): `servicediscovery:ListNamespaces`

ListOperations

Izin yang Diperlukan (Tindakan API): `servicediscovery:ListOperations`

ListServices

Izin yang Diperlukan (Tindakan API): `servicediscovery:ListServices`

ListTagsForResource

Izin yang Diperlukan (Tindakan API): `servicediscovery:ListTagsForResource`

RegisterInstance

Izin yang diperlukan (tindakan API):

- `servicediscovery:RegisterInstance`

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

TagResource

Izin yang Diperlukan (Tindakan API): `servicediscovery:TagResource`

UntagResource

Izin yang Diperlukan (Tindakan API): `servicediscovery:UntagResource`

UpdateHttpNamespace

Izin yang Diperlukan (Tindakan API): `servicediscovery:UpdateHttpNamespace`

UpdateInstanceCustomHealthStatus

Izin yang Diperlukan (Tindakan API): `servicediscovery:UpdateInstanceCustomHealthStatus`

UpdatePrivateDnsNamespace

Izin yang diperlukan (tindakan API):

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

UpdatePublicDnsNamespace

Izin yang diperlukan (tindakan API):

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

UpdateService

Izin yang diperlukan (tindakan API):

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

UpdateServiceAttributes

Izin yang Diperlukan (Tindakan API): `servicediscovery:UpdateServiceAttributes`

Memecahkan masalah AWS Cloud Map identitas dan akses

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan AWS Cloud Map dan IAM.

Topik

- Saya tidak berwenang untuk melakukan tindakan di AWS Cloud Map
- Saya tidak berwenang untuk melakukan iam: PassRole
- Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses AWS Cloud Map sumber daya saya

Saya tidak berwenang untuk melakukan tindakan di AWS Cloud Map

Jika Anda menerima pesan kesalahan bahwa Anda tidak memiliki otorisasi untuk melakukan tindakan, kebijakan Anda harus diperbarui agar Anda dapat melakukan tindakan tersebut.

Contoh kesalahan berikut terjadi ketika pengguna IAM `mateojackson` mencoba menggunakan konsol untuk melihat detail tentang suatu sumber daya *`my-example-widget`* rekaan, tetapi tidak memiliki izin `servicediscovery:`*`GetWidget`* rekaan.

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

Dalam hal ini, kebijakan untuk pengguna `mateojackson` harus diperbarui untuk mengizinkan akses ke sumber daya *`my-example-widget`* dengan menggunakan tindakan `servicediscovery:`*`GetWidget`*.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya tidak berwenang untuk melakukan iam: PassRole

Jika Anda menerima kesalahan yang tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran AWS Cloud Map.

Beberapa Layanan AWS memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi ketika pengguna IAM bernama `marymajor` mencoba menggunakan konsol tersebut untuk melakukan tindakan di AWS Cloud Map. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses AWS Cloud Map sumber daya saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACLs), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- Untuk mempelajari apakah AWS Cloud Map mendukung fitur-fitur ini, lihatBagaimana AWS Cloud Map bekerja dengan IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat Menyediakan akses ke pengguna IAM di pengguna lain Akun AWS yang Anda miliki di Panduan Pengguna IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat Menyediakan akses ke pengguna terautentikasi eksternal (federasi identitas) dalam Panduan Pengguna IAM.

- Untuk mempelajari perbedaan antara menggunakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Akses sumber daya lintas akun di IAM di Panduan Pengguna IAM.

Validasi kepatuhan untuk AWS Cloud Map

Untuk mempelajari apakah an Layanan AWS berada dalam lingkup program kepatuhan tertentu, lihat Layanan AWS di Lingkup oleh Program Kepatuhan Layanan AWS dan pilih program kepatuhan yang Anda minati. Untuk informasi umum, lihat Program AWS Kepatuhan Program AWS .

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat Mengunduh Laporan di AWS Artifact .

Tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- Kepatuhan dan Tata Kelola Keamanan – Panduan implementasi solusi ini membahas pertimbangan arsitektur serta memberikan langkah-langkah untuk menerapkan fitur keamanan dan kepatuhan.
- Referensi Layanan yang Memenuhi Syarat HIPAA — Daftar layanan yang memenuhi syarat HIPAA. Tidak semua memenuhi Layanan AWS syarat HIPAA.
- AWS Sumber Daya AWS — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- AWS Panduan Kepatuhan Pelanggan - Memahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan Layanan AWS dan memetakan panduan untuk kontrol keamanan di berbagai kerangka kerja (termasuk Institut Standar dan Teknologi Nasional (NIST), Dewan Standar Keamanan Industri Kartu Pembayaran (PCI), dan Organisasi Internasional untuk Standardisasi (ISO)).
- Mengevaluasi Sumber Daya dengan Aturan dalam Panduan AWS Config Pengembang — AWS Config Layanan menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- AWS Security Hub— Ini Layanan AWS memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS. Security Hub menggunakan kontrol keamanan untuk sumber daya AWS Anda serta untuk memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat Referensi kontrol Security Hub.

- Amazon GuardDuty — Ini Layanan AWS mendeteksi potensi ancaman terhadap beban kerja Akun AWS, kontainer, dan data Anda dengan memantau lingkungan Anda untuk aktivitas yang mencurigakan dan berbahaya. GuardDuty dapat membantu Anda mengatasi berbagai persyaratan kepatuhan, seperti PCI DSS, dengan memenuhi persyaratan deteksi intrusi yang diamanatkan oleh kerangka kerja kepatuhan tertentu.
- AWS Audit ManagerIni Layanan AWS membantu Anda terus mengaudit AWS penggunaan Anda untuk menyederhanakan cara Anda mengelola risiko dan kepatuhan terhadap peraturan dan standar industri.

Ketahanan di AWS Cloud Map

Infrastruktur AWS global dibangun di sekitar AWS Wilayah dan Zona Ketersediaan. AWS Wilayah menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis melakukan failover di antara Zona Ketersediaan tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur biasa yang terdiri dari satu atau beberapa pusat data.

AWS Cloud Map Pada dasarnya adalah layanan global. Namun, Anda dapat menggunakannya AWS Cloud Map untuk membuat pemeriksaan kesehatan Route 53 yang memeriksa kesehatan sumber daya di Wilayah tertentu, seperti EC2 instans Amazon dan penyeimbang beban Elastic Load Balancing.

Untuk informasi selengkapnya tentang AWS Wilayah dan Availability Zone, lihat Infrastruktur AWS Global.

Keamanan infrastruktur di AWS Cloud Map

Sebagai layanan terkelola, AWS Cloud Map dilindungi oleh keamanan jaringan AWS global. Untuk informasi tentang layanan AWS keamanan dan cara AWS melindungi infrastruktur, lihat Keamanan AWS Cloud. Untuk mendesain AWS lingkungan Anda menggunakan praktik terbaik untuk keamanan infrastruktur, lihat Perlindungan Infrastruktur dalam Kerangka Kerja yang AWS Diarsiteksikan dengan Baik Pilar Keamanan.

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses AWS Cloud Map melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Pengangkutan (TLS). Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Sandi cocok dengan sistem kerahasiaan maju sempurna (perfect forward secrecy, PFS) seperti DHE (Ephemeral Diffie-Hellman) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Selain itu, permintaan harus ditandatangani menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan prinsipal IAM. Atau Anda dapat menggunakan AWS Security Token Service (AWS STS) untuk menghasilkan kredensial keamanan sementara untuk menandatangani permintaan.

Anda dapat meningkatkan postur keamanan VPC Anda dengan mengonfigurasi AWS Cloud Map untuk menggunakan titik akhir VPC antarmuka. Untuk informasi selengkapnya, lihat Akses AWS Cloud Map menggunakan endpoint antarmuka ()AWS PrivateLink.

Akses AWS Cloud Map menggunakan endpoint antarmuka ()AWS PrivateLink

Anda dapat menggunakan AWS PrivateLink untuk membuat koneksi pribadi antara VPC Anda dan. AWS Cloud Map Anda dapat mengakses AWS Cloud Map seolah-olah itu ada di VPC Anda, tanpa menggunakan gateway internet, perangkat NAT, koneksi VPN, atau koneksi. AWS Direct Connect Instans di VPC Anda tidak memerlukan alamat IP publik untuk mengakses. AWS Cloud Map

Anda membuat koneksi pribadi ini dengan membuat titik akhir antarmuka, yang didukung oleh AWS PrivateLink. Kami membuat antarmuka jaringan endpoint di setiap subnet yang Anda aktifkan untuk titik akhir antarmuka. Ini adalah antarmuka jaringan yang dikelola pemohon yang berfungsi sebagai titik masuk untuk lalu lintas yang ditakdirkan. AWS Cloud Map

Untuk informasi selengkapnya, lihat Mengakses Layanan AWS melalui AWS PrivateLink di Panduan AWS PrivateLink .

Pertimbangan untuk AWS Cloud Map

Sebelum Anda menyiapkan titik akhir antarmuka AWS Cloud Map, tinjau Pertimbangan dalam Panduan.AWS PrivateLink

Jika VPC Amazon Anda tidak memiliki gateway internet dan tugas Anda menggunakan driver `awslogs` log untuk mengirim informasi log ke Log, Anda harus membuat antarmuka VPC endpoint untuk CloudWatch Log. CloudWatch Untuk informasi selengkapnya, lihat Menggunakan CloudWatch Log dengan Titik Akhir VPC Antarmuka di Panduan Pengguna Amazon CloudWatch Logs.

Titik akhir VPC tidak mendukung AWS permintaan lintas wilayah. Pastikan bahwa Anda membuat titik akhir Anda di Wilayah yang sama tempat Anda berencana untuk mengeluarkan panggilan API ke AWS Cloud Map.

Titik akhir VPC hanya mendukung DNS yang disediakan Amazon melalui Amazon Route 53. Jika Anda ingin menggunakan DNS Anda sendiri, Anda dapat menggunakan penerusan DNS bersyarat. Untuk informasi selengkapnya, lihat Set Opsi DHCP di Panduan Pengguna Amazon VPC.

Grup keamanan yang terpasang pada titik akhir VPC harus mengizinkan koneksi masuk pada port 443 dari subnet pribadi VPC Amazon.

Buat titik akhir antarmuka untuk AWS Cloud Map

Anda dapat membuat titik akhir antarmuka untuk AWS Cloud Map menggunakan konsol VPC Amazon atau () AWS Command Line Interface .AWS CLI Untuk informasi selengkapnya, lihat Membuat titik akhir antarmuka di AWS PrivateLink Panduan.

Buat titik akhir antarmuka untuk AWS Cloud Map menggunakan nama layanan berikut:

> **Note**
>
> `DiscoverInstances`API tidak akan tersedia di dua titik akhir ini.

```
com.amazonaws.region.servicediscovery
```

```
com.amazonaws.region.servicediscovery-fips
```

Buat titik akhir antarmuka untuk bidang AWS Cloud Map data untuk mengakses `DiscoverInstances` API menggunakan nama layanan berikut:

```
com.amazonaws.region.data-servicediscovery
```

```
com.amazonaws.region.data-servicediscovery-fips
```

> **Note**
>
> Anda harus menonaktifkan injeksi awalan host saat menelepon `DiscoverInstances` dengan nama DNS VPCE regional atau zona untuk titik akhir bidang data. AWS CLI

Dan menambahkan titik akhir layanan dengan berbagai AWS SDKs awalan host saat Anda memanggil setiap operasi API, yang menghasilkan URL yang tidak valid saat Anda menentukan titik akhir VPC.

Jika Anda mengaktifkan DNS pribadi untuk titik akhir antarmuka, Anda dapat membuat permintaan API untuk AWS Cloud Map menggunakan nama DNS Regional default. Misalnya, `servicediscovery.us-east-1.amazonaws.com`.

AWS PrivateLink Koneksi VPCE didukung di Wilayah mana pun yang AWS Cloud Map didukung; namun, pelanggan perlu memeriksa Availability Zones mana yang mendukung VPCE sebelum menentukan titik akhir. Untuk mengetahui Zona Ketersediaan mana yang didukung dengan titik akhir VPC antarmuka di Wilayah, gunakan describe-vpc-endpoint-services perintah atau gunakan. AWS Management Console Misalnya, perintah berikut mengembalikan zona ketersediaan tempat Anda dapat menerapkan titik akhir VPC AWS Cloud Map antarmuka di Wilayah AS Timur (Ohio):

```
aws --region us-east-2 ec2 describe-vpc-endpoint-services --query 'ServiceDetails[?
ServiceName==`com.amazonaws.us-east-2.servicediscovery`].AvailabilityZones[]'
```

Pemantauan AWS Cloud Map

Pemantauan adalah bagian penting dalam menjaga keandalan, ketersediaan, dan performa solusi AWS Anda. Anda harus mengumpulkan data pemantauan dari semua bagian AWS solusi Anda sehingga Anda dapat lebih mudah men-debug kegagalan multi-titik jika terjadi. Namun sebelum mulai memantau; Anda harus membuat rencana pemantauan yang mencakup jawaban atas pertanyaan berikut:

- Apa sasaran pemantauan Anda?
- Sumber daya apa yang akan Anda pantau?
- Seberapa sering Anda akan memantau sumber daya ini?
- Alat pemantauan apa yang akan Anda gunakan?
- Siapa yang akan melakukan tugas pemantauan?
- Siapa yang harus diberi tahu saat terjadi kesalahan?

Topik

Log panggilan AWS Cloud Map API menggunakan AWS CloudTrail

AWS Cloud Map terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau Layanan AWS. CloudTrail menangkap semua panggilan API untuk AWS Cloud Map sebagai peristiwa. Panggilan yang diambil termasuk panggilan dari AWS Cloud Map konsol dan panggilan kode ke operasi AWS Cloud Map API. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat AWS Cloud Map, alamat IP dari mana permintaan dibuat, kapan dibuat, dan detail tambahan.

Setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan berikut hal ini:

- Baik permintaan tersebut dibuat dengan kredensial pengguna root atau pengguna.
- Apakah permintaan dibuat atas nama pengguna Pusat Identitas IAM.
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna gabungan.

- Apakah permintaan tersebut dibuat oleh Layanan AWS lain.

CloudTrail aktif di Anda Akun AWS ketika Anda membuat akun dan Anda secara otomatis memiliki akses ke riwayat CloudTrail Acara. Riwayat CloudTrail Acara menyediakan catatan yang dapat dilihat, dapat dicari, dapat diunduh, dan tidak dapat diubah dari 90 hari terakhir dari peristiwa manajemen yang direkam dalam file. Wilayah AWS Untuk informasi selengkapnya, lihat Bekerja dengan riwayat CloudTrail Acara di Panduan AWS CloudTrail Pengguna. Tidak ada CloudTrail biaya untuk melihat riwayat Acara.

Untuk catatan acara yang sedang berlangsung dalam 90 hari Akun AWS terakhir Anda, buat jejak atau penyimpanan data acara CloudTrailDanau.

CloudTrail jalan setapak

Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Semua jalur yang dibuat menggunakan AWS Management Console Multi-region. Anda dapat membuat jalur Single-region atau Multi-region dengan menggunakan. AWS CLI Membuat jejak Multi-wilayah disarankan karena Anda menangkap aktivitas Wilayah AWS di semua akun Anda. Jika Anda membuat jejak wilayah Tunggal, Anda hanya dapat melihat peristiwa yang dicatat di jejak. Wilayah AWS Untuk informasi selengkapnya tentang jejak, lihat Membuat jejak untuk Anda Akun AWS dan Membuat jejak untuk organisasi di Panduan AWS CloudTrail Pengguna.

Anda dapat mengirimkan satu salinan acara manajemen yang sedang berlangsung ke bucket Amazon S3 Anda tanpa biaya CloudTrail dengan membuat jejak, namun, ada biaya penyimpanan Amazon S3. Untuk informasi selengkapnya tentang CloudTrail harga, lihat AWS CloudTrail Harga. Untuk informasi tentang harga Amazon S3, lihat Harga Amazon S3.

CloudTrail Menyimpan data acara danau

CloudTrail Lake memungkinkan Anda menjalankan kueri berbasis SQL pada acara Anda. CloudTrail Lake mengonversi peristiwa yang ada dalam format JSON berbasis baris ke format Apache ORC. ORC adalah format penyimpanan kolumnar yang dioptimalkan untuk pengambilan data dengan cepat. Peristiwa digabungkan ke dalam penyimpanan data peristiwa, yang merupakan kumpulan peristiwa yang tidak dapat diubah berdasarkan kriteria yang Anda pilih dengan menerapkan pemilih acara tingkat lanjut. Penyeleksi yang Anda terapkan ke penyimpanan data acara mengontrol peristiwa mana yang bertahan dan tersedia untuk Anda kueri. Untuk informasi lebih lanjut tentang CloudTrail Danau, lihat Bekerja dengan AWS CloudTrail Danau di Panduan AWS CloudTrail Pengguna.

CloudTrail Penyimpanan data acara danau dan kueri menimbulkan biaya. Saat Anda membuat penyimpanan data acara, Anda memilih opsi harga yang ingin Anda gunakan untuk penyimpanan data acara. Opsi penetapan harga menentukan biaya untuk menelan dan menyimpan peristiwa, dan periode retensi default dan maksimum untuk penyimpanan data acara. Untuk informasi selengkapnya tentang CloudTrail harga, lihat AWS CloudTrail Harga.

AWS Cloud Map peristiwa data di CloudTrail

Peristiwa data memberikan informasi tentang operasi sumber daya yang dilakukan pada atau di sumber daya (misalnya, menemukan instance terdaftar di namespace). Ini juga dikenal sebagai operasi bidang data. Peristiwa data seringkali merupakan aktivitas volume tinggi. Secara default, CloudTrail tidak mencatat peristiwa data. Riwayat CloudTrail peristiwa tidak merekam peristiwa data.

Biaya tambahan berlaku untuk peristiwa data. Untuk informasi selengkapnya tentang CloudTrail harga, lihat AWS CloudTrail Harga.

Anda dapat mencatat peristiwa data untuk jenis AWS Cloud Map sumber daya menggunakan CloudTrail konsol AWS CLI, atau operasi CloudTrail API. Untuk informasi selengkapnya tentang cara mencatat peristiwa data, lihat Mencatat peristiwa data dengan AWS Management Console dan Mencatat peristiwa data dengan AWS Command Line Interface di Panduan AWS CloudTrail Pengguna.

Tabel berikut mencantumkan jenis AWS Cloud Map sumber daya yang dapat Anda log peristiwa data. Kolom tipe peristiwa data (konsol) menunjukkan nilai yang akan dipilih dari daftar tipe peristiwa Data di CloudTrail konsol. Kolom nilai resources.type menunjukkan **`resources.type`** nilai, yang akan Anda tentukan saat mengonfigurasi penyeleksi acara lanjutan menggunakan or. AWS CLI CloudTrail APIs CloudTrailKolom Data yang APIs dicatat ke menampilkan panggilan API yang dicatat CloudTrail untuk jenis sumber daya.

Jenis peristiwa data (konsol)	nilai resources.type	Data APIs masuk CloudTrail
AwsApiCall	`AWS::ServiceDiscovery::Namespace`	• DiscoverInstances • DiscoverInstancesRevision
AwsApiCall	`AWS::ServiceDiscovery::Service`	• DiscoverInstances • DiscoverInstancesRevision

Anda dapat mengonfigurasi pemilih acara lanjutan untuk memfilter pada `eventName`,`readOnly`, dan `resources.ARN` bidang untuk mencatat hanya peristiwa yang penting bagi Anda. Untuk informasi selengkapnya tentang bidang ini, lihat AdvancedFieldSelector di Referensi API AWS CloudTrail .

Contoh berikut menunjukkan cara mengkonfigurasi pemilih acara lanjutan untuk mencatat semua peristiwa AWS Cloud Map data.

```
"AdvancedEventSelectors":
[
    {
        "Name": "Log all AWS Cloud Map data events",
        "FieldSelectors": [
            { "Field": "eventCategory", "Equals": ["Data"] },
            { "Field": "resources.type", "Equals":
 ["AWS::ServiceDiscovery::Namespace"] }
        ]
    }
]
```

AWS Cloud Map acara manajemen di CloudTrail

Acara manajemen memberikan informasi tentang operasi manajemen yang dilakukan pada sumber daya di Anda Akun AWS. Ini juga dikenal sebagai operasi pesawat kontrol. Secara default, CloudTrail mencatat peristiwa manajemen.

AWS Cloud Map mencatat semua operasi pesawat AWS Cloud Map kontrol sebagai peristiwa manajemen. Untuk daftar operasi bidang AWS Cloud Map kontrol yang AWS Cloud Map masuk ke log CloudTrail, lihat Referensi AWS Cloud Map API.

AWS Cloud Map contoh acara

Peristiwa mewakili permintaan tunggal dari sumber manapun dan mencakup informasi tentang operasi API yang diminta, tanggal dan waktu operasi, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, sehingga peristiwa tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan acara CloudTrail manajemen yang menunjukkan `CreateHTTPNamespace` operasi.

```
{
    "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"
}
```

Contoh berikut menunjukkan peristiwa CloudTrail data yang menunjukkan `DiscoverInstances` operasi.

```
{
            "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"
        }
```


Untuk informasi tentang konten CloudTrail rekaman, lihat konten CloudTrail rekaman di Panduan AWS CloudTrail Pengguna.

Menandai sumber daya Anda AWS Cloud Map

Tag adalah label yang Anda tetapkan ke AWS sumber daya. Setiap tanda terdiri dari kunci dan nilai opsional, yang keduanya Anda tentukan.

Tag memungkinkan Anda untuk mengkategorikan AWS sumber daya berdasarkan, misalnya, tujuan, pemilik, atau lingkungan. Saat Anda memiliki banyak sumber daya dengan jenis yang sama, Anda dapat dengan segera mengidentifikasi sumber daya yang spesifik berdasarkan tanda yang telah Anda tetapkan pada sumber daya. Misalnya, Anda dapat menentukan satu set tag untuk AWS Cloud Map layanan Anda untuk membantu Anda melacak setiap pemilik layanan dan tingkat tumpukan. Kami menyarankan agar Anda merancang serangkaian kunci tanda yang konsisten untuk setiap jenis sumber daya.

Selain itu, tanda tidak dapat menetapkan secara otomatis ke sumber daya Anda. Setelah Anda menambahkan sebuah tanda, Anda dapat mengedit kunci serta nilai tanda atau menghilangkan tanda dari sumber daya kapanpun yang Anda mau. Jika Anda menghapus sebuah sumber daya, tanda apapun untuk sumber daya tersebut juga dihapus.

Tag tidak memiliki arti semantik AWS Cloud Map dan ditafsirkan secara ketat sebagai serangkaian karakter. Anda dapat mengatur nilai tanda ke string kosong, tetapi tidak dapat mengatur nilai tanda ke null. Jika Anda menambahkan tanda yang memiliki kunci yang sama dengan tanda yang telah ada pada sumber daya tersebut, nilai yang baru akan menimpa nilai yang lama.

Anda dapat bekerja dengan tag menggunakan AWS Management Console, the AWS CLI, dan AWS Cloud Map API.

Jika Anda menggunakan AWS Identity and Access Management (IAM), Anda dapat mengontrol pengguna mana di AWS akun Anda yang memiliki izin untuk membuat, mengedit, atau menghapus tag.

Bagaimana sumber daya ditandai

Anda dapat menandai AWS Cloud Map ruang nama dan layanan baru atau yang sudah ada.

Jika menggunakan AWS Cloud Map konsol, Anda dapat menerapkan tag ke sumber daya baru saat dibuat atau ke sumber daya yang ada kapan saja menggunakan tab Tag di halaman sumber daya yang relevan.

Jika Anda menggunakan AWS Cloud Map API, SDK AWS CLI, atau AWS SDK, Anda dapat menerapkan tag ke sumber daya baru menggunakan `tags` parameter pada tindakan API yang relevan atau ke sumber daya yang ada menggunakan tindakan TagResourceAPI. Untuk informasi selengkapnya, lihat TagResource.

Selain itu, beberapa tindakan pembuatan sumber daya memungkinkan Anda menentukan tanda untuk sumber daya saat sumber daya diciptakan. Jika tanda tidak dapat diterapkan selama pembuatan sumber daya, proses pembuatan sumber daya akan gagal. Hal ini memastikan bahwa sumber daya yang ingin Anda tandai pada saat pembuatan dapat dibuat dengan tanda yang ditentukan atau justru tidak dibuat sama sekali. Jika Anda menandai sumber daya pada saat pembuatan, Anda tidak perlu menjalankan skrip penandaan khusus setelah pembuatan sumber daya.

Tabel berikut menjelaskan AWS Cloud Map sumber daya yang dapat ditandai, dan sumber daya yang dapat ditandai pada pembuatan.

Menandai dukungan untuk sumber daya AWS Cloud Map

Sumber daya	Mendukung tanda	Penyebaran tanda Support	Mendukung penandaan pada pembuatan (AWS Cloud Map API, AWS CLI, AWS SDK)
AWS Cloud Map ruang nama	Ya	Tidak. Tag namespace tidak menyebarkan ke sumber daya lain yang terkait dengan namespace.	Ya
AWS Cloud Map layanan	Ya	Tidak. Tag layanan tidak menyebar ke sumber daya lain yang terkait dengan layanan.	Ya

Pembatasan

Batasan dasar berikut berlaku untuk tag:

- Jumlah maksimum tanda per sumber daya – 50
- Untuk setiap sumber daya, setiap kunci tanda harus unik, dan setiap kunci tanda hanya dapat memiliki satu nilai.
- Panjang kunci maksimum – 128 karakter Unicode dalam UTF-8
- Panjang nilai maksimum – 256 karakter Unicode dalam UTF-8
- Jika skema penandaan Anda digunakan di beberapa AWS layanan dan sumber daya, ingatlah bahwa layanan lain mungkin memiliki batasan pada karakter yang diizinkan. Karakter yang diperbolehkan adalah: huruf, spasi, dan angka yang dapat mewakili dalam UTF-8, serta karakter berikut: + - = . _ : / @.
- Kunci dan nilai tanda sensitif huruf besar dan kecil.
- Jangan gunakan`aws:`,`AWS:`, atau kombinasi huruf besar atau kecil seperti awalan untuk kunci atau nilai, karena dicadangkan untuk digunakan. AWS Anda tidak dapat menyunting atau menghapus kunci atau nilai tag dengan awalan ini. Tag dengan awalan ini tidak dihitung terhadap tags-per-resource batas Anda.

Memperbarui tag untuk AWS Cloud Map sumber daya

Gunakan AWS CLI perintah atau operasi AWS Cloud Map API berikut untuk menambahkan, memperbarui, membuat daftar, dan menghapus tag untuk sumber daya Anda.

Menandai dukungan untuk sumber daya AWS Cloud Map

Tugas	Tindakan API	AWS CLI	AWS Tools for Windows PowerShell
Tambahkan atau timpa satu atau beberapa tanda.	TagResource	tag-sumber daya	Tambah- SDResource Tag
Hapus satu atau beberapa tanda.	UntagResource	untag-sumber daya	Hapus- SDResource Tag

Tugas	Tindakan API	AWS CLI	AWS Tools for Windows PowerShell
Membuat daftar tanda untuk sumber daya	ListTagsForResource	list-tags-for-resource	Dapatkan- SDResource Tag

Contoh-contoh berikut menunjukkan cara menambahkan atau menghilangkan tanda sumber daya menggunakan AWS CLI.

Contoh 1: Menandai sumber daya yang sudah ada

Perintah berikut ini menandai sumber daya yang sudah ada.

```
aws servicediscovery tag-resource --resource-arn resource_ARN --tags team=devs
```

Contoh 2: Menghapus tanda sumber daya yang sudah ada

Perintah berikut ini menghapus tanda dari sumber daya yang sudah ada.

```
aws servicediscovery untag-resource --resource-arn resource_ARN --tag-keys tag_key
```

Contoh 3: Cantumkan tanda untuk sumber daya

Perintah berikut akan mencantumkan tanda terkait dengan sumber daya yang sudah ada.

```
aws servicediscovery list-tags-for-resource --resource-arn resource_ARN
```

Beberapa tindakan pembuatan sumber daya memungkinkan Anda untuk menentukan tanda saat membuat sumber daya. Tindakan berikut mendukung penandaan saat pembuatan.

Tugas	Tindakan API	AWS CLI	AWS Tools for Windows PowerShell
Buat namespace HTTP	CreateHttpNamespace	create-http-namespace	Baru- SDHttp Namespace

Tugas	Tindakan API	AWS CLI	AWS Tools for Windows PowerShell
Membuat namespace pribadi berdasarkan DNS	CreatePrivateDnsNamespace	create-private-dns-namespace	Baru- SDPrivateDnsNamespace
Membuat namespace publik berdasarkan DNS	CreatePublicDnsNamespace	create-public-dns-namespace	Baru- SDPublicDnsNamespace
Membuat layanan	CreateService	membuat-layanan	Baru- SDService

AWS Cloud Map kuota layanan

AWS Cloud Map sumber daya tunduk pada kuota layanan tingkat akun berikut. Setiap kuota yang tercantum berlaku untuk setiap AWS Wilayah tempat Anda membuat AWS Cloud Map sumber daya.

Nama	Default	Dapat disesuaikan	Deskripsi
Atribut kustom per instans	Setiap Wilayah yang didukung: 30	Tidak	Jumlah maksimum atribut kustom yang dapat Anda tentukan saat Anda mendaftarkan instance.
DiscoverInstances operasi per laju burst akun	Setiap Wilayah yang didukung: 2.000	Ya	Kecepatan burst maksimum untuk memanggil DiscoverInstances operasi dari satu akun.
DiscoverInstances operasi per akun tingkat stabil	Setiap Wilayah yang didukung: 1.000	Ya	Tingkat stabil maksimum untuk memanggil DiscoverInstances operasi dari satu akun.
DiscoverInstancesRevision operasi per tingkat akun	Setiap Wilayah yang didukung: 3.000	Ya	Tingkat maksimum untuk memanggil DiscoverInstancesRevision operasi dari satu akun.
Instans per namespace	Setiap Wilayah yang didukung: 2.000	Ya	Jumlah maksimum instance layanan yang dapat Anda daftarkan menggunakan namespace yang sama.

Nama	Default	Dapat disesuaikan	Deskripsi
Instans per layanan	Setiap Wilayah yang didukung: 1.000	Tidak	Jumlah maksimum instans yang dapat Anda daftarkan di Wilayah menggunakan layanan yang sama.
Namespace per Wilayah	Setiap Wilayah yang didukung: 50	Ya	Jumlah maksimum ruang nama yang dapat Anda buat per Wilayah.

* Bila Anda membuat namespace, kita secara otomatis membuat zona yang di-hosting Amazon Route 53. Zona yang dihosting ini dihitung terhadap kuota jumlah zona yang dihosting yang dapat Anda buat dengan akun AWS . Untuk informasi selengkapnya, lihat kuota pada zona yang di-hosting dalam Panduan Developer Amazon Route 53.

** Meningkatkan instans untuk ruang nama DNS AWS Cloud Map memerlukan peningkatan catatan per batas zona Route 53 yang dihosting, yang menimbulkan biaya tambahan.

Mengelola kuota AWS Cloud Map layanan Anda

AWS Cloud Map telah terintegrasi dengan Service Quotas, sebuah AWS layanan yang memungkinkan Anda untuk melihat dan mengelola kuota Anda dari lokasi pusat. Untuk informasi selengkapnya, lihat Apa itu Service Quotas? dalam Panduan Pengguna Service Quotas.

Service Quotas memudahkan untuk mencari nilai kuota AWS Cloud Map layanan Anda.

AWS Management Console

Untuk melihat kuota AWS Cloud Map layanan menggunakan AWS Management Console

1. Buka konsol Service Quotas di https://console.aws.amazon.com/servicequotas/.
2. Di panel navigasi, pilih Layanan AWS .
3. Dari daftar Layanan AWS , cari dan pilih AWS Cloud Map.

4. Dalam daftar kuota layanan untuk AWS Cloud Map, Anda dapat melihat nama kuota layanan, nilai yang diterapkan (jika tersedia), kuota AWS default, dan apakah nilai kuota dapat disesuaikan.

   Untuk melihat informasi tambahan tentang kuota layanan, seperti deskripsi, pilih nama kuota untuk memunculkan detail kuota.
5. (Opsional) Untuk meminta kenaikan kuota, pilih kuota yang ingin Anda tingkatkan dan pilih Permintaan peningkatan di tingkat akun.

Untuk bekerja lebih banyak dengan kuota layanan menggunakan AWS Management Console lihat Panduan Pengguna Service Quotas.

AWS CLI

Untuk melihat kuota AWS Cloud Map layanan menggunakan AWS CLI

Jalankan perintah berikut untuk melihat AWS Cloud Map kuota default.

```
aws service-quotas list-aws-default-service-quotas \
    --query 'Quotas[*].
{Adjustable:Adjustable,Name:QuotaName,Value:Value,Code:QuotaCode}' \
    --service-code AWSCloudMap \
    --output table
```

Jalankan perintah berikut untuk melihat AWS Cloud Map kuota yang diterapkan.

```
aws service-quotas list-service-quotas \
    --service-code AWSCloudMap
```

Untuk informasi selengkapnya tentang bekerja dengan kuota layanan menggunakan AWS CLI, lihat Referensi Perintah Service AWS CLI Quotas. Untuk meminta peningkatan kuota, lihat perintah `request-service-quota-increase` dalam Referensi Perintah AWS CLI.

Menangani AWS Cloud Map DiscoverInstances pembatasan permintaan API

AWS Cloud Map membatasi permintaan DiscoverInstancesAPI untuk setiap AWS akun berdasarkan per wilayah. Throttling membantu meningkatkan kinerja layanan dan membantu memberikan

penggunaan yang adil untuk semua AWS Cloud Map pelanggan. Throttling memastikan bahwa panggilan ke AWS Cloud Map DiscoverInstancesAPI tidak melebihi kuota permintaan DiscoverInstancesAPI maksimum yang diizinkan. DiscoverInstances Panggilan API yang berasal dari salah satu sumber berikut tunduk pada kuota permintaan:

- Aplikasi pihak ketiga
- Alat baris perintah
- AWS Cloud Map Konsol

Jika melebihi kuota throttling API, Anda mendapatkan kode kesalahan `RequestLimitExceeded`. Untuk informasi lebih lanjut, lihat the section called “Pembatasan Laju Permintaan”.

Bagaimana throttling diterapkan

AWS Cloud Map menggunakan algoritma token bucket untuk mengimplementasikan pelambatan API. Dengan algoritme ini, akun Anda memiliki bucket yang memegang sejumlah tertentu token. Jumlah token dalam bucket mewakili kuota throttling Anda pada detik tertentu. Ada satu bucket untuk Wilayah tunggal, dan itu berlaku untuk semua titik akhir di Wilayah.

Pembatasan Laju Permintaan

Throttling membatasi jumlah permintaan DiscoverInstancesAPI yang dapat Anda buat. Setiap permintaan menghapus satu token dari bucket. Misalnya, ukuran bucket untuk operasi DiscoverInstancesAPI adalah 2.000 token, sehingga Anda dapat membuat hingga 2.000 DiscoverInstancespermintaan dalam satu detik. Jika Anda melebihi 2.000 permintaan dalam satu detik, Anda throttled dan permintaan yang tersisa dalam detik itu gagal.

Bucket secara otomatis diisi ulang pada tingkat yang ditetapkan. Jika bucket tidak pada kapasitasnya, sejumlah token ditambahkan kembali setiap detik sampai bucket mencapai kapasitas. Jika bucket pada kapasitas saat token isi ulang tiba, maka token ini dibuang. Ukuran bucket untuk operasi DiscoverInstancesAPI adalah 2.000 token, dan tingkat isi ulang adalah 1.000 token setiap detik. Jika Anda membuat 2.000 permintaan DiscoverInstancesAPI dalam satu detik, bucket segera dikurangi menjadi nol (0) token. Bucket tersebut kemudian diisi ulang hingga 1.000 token setiap detik hingga mencapai kapasitas maksimum 2.000 token.

Anda dapat menggunakan token karena mereka ditambahkan ke bucket. Anda tidak perlu menunggu bucket berada pada kapasitas maksimum sebelum membuat permintaan API. Jika Anda menghabiskan bucket dengan membuat 2.000 permintaan DiscoverInstancesAPI dalam satu detik,

Anda masih dapat membuat hingga 1.000 permintaan DiscoverInstancesAPI setiap detik setelah itu selama yang Anda butuhkan. Ini berarti Anda dapat segera menggunakan token isi ulang saat ditambahkan ke bucket Anda. Bucket hanya mulai diisi ulang ke kapasitas maksimum ketika Anda membuat permintaan API lebih sedikit setiap detik dari tingkat isi ulang.

Pemrosesan coba ulang atau batch

Jika permintaan API gagal, aplikasi Anda mungkin perlu mencoba lagi permintaan. Untuk meredam jumlah permintaan API, gunakan interval tidur yang sesuai antara permintaan berturut-turut. Untuk hasil terbaik, gunakan interval tidur yang meningkat atau variabel.

Menghitung interval tidur

Ketika Anda harus melakukan polling atau mencoba lagi permintaan API, sebaiknya gunakan algoritme backoff eksponensial untuk menghitung interval tidur antara panggilan API. Dengan menggunakan semakin lama waktu tunggu antara mencoba untuk respons kesalahan berturut-turut, Anda dapat mengurangi jumlah permintaan gagal. Untuk informasi selengkapnya dan contoh implementasi algoritme ini, lihat Coba Ulang Perilaku di Panduan Referensi Alat AWS SDKs dan Alat.

Menyesuaikan kuota throttling API

Anda dapat meminta peningkatan kuota pembatasan API untuk akun Anda. AWS Untuk meminta penyesuaian kuota, hubungi AWS Dukungan Pusat.

Riwayat dokumen untuk AWS Cloud Map

Tabel berikut menjelaskan pembaruan utama dan fitur baru untuk Panduan AWS Cloud Map Pengembang. Kami juga rutin memperbarui dokumentasi untuk menjawab umpan balik yang Anda kirimkan kepada kami.

Perubahan	Deskripsi	Tanggal
AWS Cloud Map atribut layanan	Anda sekarang dapat menentukan atribut di tingkat layanan untuk menghindari duplikasi atribut di seluruh instance yang terdaftar ke layanan. Anda dapat menggunakan atribut ini untuk perutean lalu lintas yang kompleks, menetapkan nilai batas waktu dan coba lagi, dan untuk koordinasi antara layanan dan integrasi eksternal.	Desember 13, 2024
Tutorial ditambahkan	Dua tutorial yang menunjukkan kasus penggunaan umum untuk menggunakan AWS Cloud Map ditambahkan.	Maret 27, 2024
CloudTrail dokumentasi integrasi diperbarui	Dokumentasi yang menjelaskan AWS Cloud Map integrasi CloudTrail dengan aktivitas API log telah diperbarui.	Maret 20, 2024
Pembaruan kebijakan terkelola	`AWSCloudMapDiscoverInstance Access` ,`AWSCloudMapRegisterInstance Access` , dan `AWSCloudM`	20 September 2023

	`apReadOnlyAccess` kebijakan diperbarui.	
Cloud Map dan AWS PrivateLink	Anda sekarang dapat menggunakan sebuah AWS PrivateLink untuk membuat koneksi pribadi antara VPC Anda dan. AWS Cloud Map	15 September 2023
Pembaruan kebijakan terkelola	`AWSCloudMapDiscoverInstance Access` kebijakan telah diperbarui.	15 Agustus 2023
AWS SDK untuk Python	Ditambahkan contoh baris perintah Python.	13 September 2022
IPv6 dukungan	Titik akhir API sekarang tersedia di jaringan `IPv6` - only.	28 Januari 2022
Penemuan contoh layanan	AWS Cloud Map menambahkan dukungan untuk membuat layanan di namespace yang mendukung kueri DNS yang hanya dapat ditemukan menggunakan operasi DiscoverInstancesAPI dan tidak menggunakan kueri DNS.	24 Maret 2021
Penandaan sumber daya	AWS Cloud Map menambahkan dukungan untuk menambahkan tag metadata ke ruang nama dan layanan Anda menggunakan. AWS Management Console	8 Februari 2021

Penandaan sumber daya	AWS Cloud Map menambahkan dukungan untuk menambahkan tag metadata ke ruang nama dan layanan Anda menggunakan dan. AWS CLI APIs	22 Juni 2020
Rilis Awal	Ini merupakan rilis pertama AWS Cloud Map Panduan Developer.	28 November, 2018

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.