



Panduan Developer

# Amazon Location Service



# Amazon Location Service: Panduan Developer

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan kekayaan masing-masing pemiliknya, yang mungkin atau mungkin tidak berafiliasi, terkait dengan, atau disponsori oleh Amazon.

---

# Table of Contents

Selamat datang .....	1
Apa itu Amazon Location Service? .....	1
Fitur utama .....	2
Layanan terkait .....	3
Quick start .....	5
Membuat aplikasi web .....	5
Buat sumber daya .....	6
Otentikasi pengaturan .....	7
Membuat HTML .....	9
Menambahkan peta .....	12
Menambahkan pencarian .....	16
Aplikasi akhir .....	21
Apa selanjutnya .....	26
Membuat aplikasi Android .....	26
Membuat sumber daya Lokasi Amazon untuk aplikasi Anda .....	26
Otentikasi pengaturan .....	28
Buat aplikasi .....	31
Tambahkan peta .....	32
Tambahkan pencarian .....	36
Tambahkan pelacakan .....	45
Apa selanjutnya .....	54
Membuat aplikasi iOS .....	55
Buat sumber daya .....	55
Otentikasi pengaturan .....	57
Membuat aplikasi .....	59
Kode awal .....	60
Tambahkan peta .....	63
Tambahkan pencarian .....	67
Tambahkan pelacakan .....	69
Apa selanjutnya .....	81
Konsep Lokasi Amazon .....	82
Gambaran Umum .....	83
Peta .....	84
Gaya peta .....	85

Pandangan politik .....	85
Lapisan Kustom .....	86
Rendering peta .....	87
Terminologi peta .....	88
Pencarian tempat .....	89
Konsep geocoding .....	91
Hasil pencarian .....	91
Berbagai hasil dan relevansi .....	92
Alamat hasil .....	92
Menyimpan hasil geocode .....	94
Terminologi tempat .....	95
Rute .....	96
Sumber daya kalkulator rute .....	96
Menghitung rute .....	97
Merencanakan rute .....	98
Terminologi rute .....	99
Geofences dan Pelacak .....	101
Geofences .....	101
Pelacak .....	103
Terminologi geofensi .....	107
Terminologi pelacak .....	108
Kasus penggunaan umum .....	110
Keterlibatan pengguna dan aplikasi geomarketing .....	111
Aplikasi pelacakan aset .....	113
Aplikasi pengiriman .....	114
Penyedia data .....	116
Cakupan dan fitur penyedia data .....	116
Gaya peta .....	118
Detail selengkapnya .....	118
Esri .....	118
GrabMaps .....	126
HERE Teknologi .....	132
Buka Data .....	139
Fitur oleh penyedia data .....	148
Ketentuan penggunaan dan atribusi data .....	153
Wilayah dan titik akhir .....	153

Wilayah .....	153
Titik akhir .....	155
Titik Akhir operasi API .....	156
Kuota layanan .....	158
Mengelola kuota layanan Lokasi Amazon Anda .....	173
Berkembang dengan Lokasi Amazon .....	175
Skenario dan kasus penggunaan .....	175
SDK dan alat .....	177
SDK menurut bahasa .....	177
MapLibre .....	181
SDK Lokasi Amazon .....	186
API Lokasi Amazon .....	210
Menggunakan Lokasi Amazon dengan AWS SDK .....	210
Pembaruan pesan kesalahan .....	211
Contoh kode .....	245
Situs Demo Lokasi Amazon .....	246
Tutorial: Mulai cepat .....	247
Tutorial: Pengayaan basis data .....	247
Contoh: Jelajahi aplikasi .....	248
Contoh: Gaya peta .....	249
Contoh: Gambar spidol .....	249
Contoh: Gambar poin berkerumun .....	250
Contoh: Menggambar poligon .....	250
Contoh: Ubah bahasa peta .....	251
Blog: Perkiraan pemberitahuan waktu pengiriman .....	252
Contoh: Pembaruan Posisi Stream .....	253
Contoh: Aplikasi seluler Geofencing dan Tracking .....	253
Cara menggunakan Lokasi Amazon .....	254
Prasyarat akun .....	255
Mendaftar untuk Akun AWS .....	255
Buat pengguna dengan akses administratif .....	256
Berikan akses ke Amazon Location Service .....	257
Menggunakan peta .....	259
Prasyarat .....	260
Menampilkan peta .....	263
Menggambar di peta .....	318

Pengaturan meluas untuk peta .....	319
Mengelola sumber daya peta .....	320
Pencarian tempat .....	324
Prasyarat .....	325
Geocoding .....	328
Membalikkan geocoding .....	336
Pelengkapan otomatis .....	340
Menggunakan ID tempat .....	346
Kategori dan penyaringan .....	348
Tutorial: Pengayaan basis data .....	353
Mengelola sumber daya indeks tempat .....	368
Menghitung rute .....	371
Prasyarat .....	372
Hitung rute .....	376
Perencanaan rute .....	380
Posisi tidak terletak di jalan .....	386
Waktu keberangkatan .....	388
Mode perjalanan .....	389
Mengelola sumber daya rute .....	391
Geofencing dan pelacakan .....	394
Langkah 1: Tambahkan geofences .....	396
Langkah 2: Mulai melacak .....	403
Langkah 3: Tautkan pelacak ke koleksi geofence .....	417
Langkah 4: Evaluasi posisi perangkat terhadap geofences .....	418
Verifikasi posisi perangkat .....	421
Bereaksi terhadap peristiwa dengan EventBridge .....	423
Melacak menggunakan AWS IoT dan MQTT .....	429
Mengelola sumber daya geofence .....	437
Mengelola sumber daya pelacak .....	445
Contoh aplikasi seluler Geofencing dan Tracking .....	449
Menandai Sumber Daya Anda .....	468
Pembatasan .....	469
Berikan izin untuk menandai .....	469
Penambahan sebuah tag ke sebuah sumber daya .....	470
Lacak biaya dengan tag .....	471
Mengontrol akses ke sumber daya menggunakan tanda .....	472

Pelajari selengkapnya .....	472
Berikan akses ke Lokasi Amazon .....	472
Menggunakan kunci API .....	473
Menggunakan Amazon Cognito .....	480
Memantau Amazon Location Service .....	491
Pemantauan CloudWatch dengan .....	491
Menggunakan CloudTrail dengan Lokasi Amazon .....	497
Menggunakan AWS CloudFormation untuk membuat sumber daya .....	501
Lokasi Amazon dan AWS CloudFormation template .....	501
Pelajari selengkapnya tentang AWS CloudFormation .....	502
Keamanan .....	503
Perlindungan data .....	504
Privasi data .....	505
Retensi data .....	505
Enkripsi data saat istirahat .....	505
Data dalam enkripsi transit .....	519
Identity and Access Management .....	519
Audiens .....	519
Mengautentikasi dengan identitas .....	520
Mengelola akses menggunakan kebijakan .....	524
Bagaimana Amazon Location Service bekerja dengan IAM .....	526
Cara Amazon Location Service bekerja dengan pengguna yang tidak diautentikasi .....	535
Contoh kebijakan berbasis identitas .....	535
Pemecahan Masalah .....	548
Respons insiden .....	550
Pembuatan Log dan Pemantauan .....	550
Validasi kepatuhan .....	551
Ketangguhan .....	553
Keamanan infrastruktur .....	553
Konfigurasi dan analisis kerentanan .....	554
Pencegahan Deputi Bingung .....	554
Praktik terbaik keamanan .....	554
Praktik terbaik Detektif .....	554
Praktik terbaik preventif .....	555
Praktik terbaik .....	556
Keamanan .....	556

---

Manajemen sumber daya .....	557
Manajemen penagihan dan biaya .....	558
Kuota dan penggunaan .....	558
Riwayat dokumen .....	559
Glosarium AWS .....	569
.....	dlxx



# Selamat datang di Amazon Location Service

Selamat datang di Panduan Pengembang Layanan Lokasi Amazon.

Topik berikut dapat membantu Anda memulai dokumentasi, berdasarkan apa yang Anda coba lakukan.

Dapatkan ikhtisar Lokasi Amazon

- Pelajari tentang [konsep di Lokasi Amazon](#).
- Selami lebih dalam fungsionalitas di bagian [Cara menggunakan Amazon Location Service](#) ini.
- Lihat aplikasi demo di [situs demo Lokasi Amazon](#).
- Jika sudah memiliki Akun AWS, Anda dapat menggunakan [konsol Amazon Location Service](#) untuk menjelajahi fungsionalitas secara langsung.

Gunakan Lokasi Amazon sebagai pengembang

- Bangun aplikasi pertama Anda dengan [Quick start](#).
- Pelajari cara kerja berbagai fitur Amazon Location Service di bagian [Cara menggunakan Amazon Location Service](#) ini.
- Lihat SDK dan alat yang tersedia untuk Anda di bagian ini [Berkembang dengan Lokasi Amazon](#).
- Lihat [contoh kode dan tutorial](#) yang dapat Anda gunakan di aplikasi Anda sendiri. Anda juga dapat mengunjungi [halaman sampel](#) situs demo Lokasi Amazon untuk menemukan sampel, dapat disaring berdasarkan fitur, bahasa, atau platform.
- Dapatkan informasi tentang API Lokasi Amazon di [panduan Referensi API](#).

## Apa itu Amazon Location Service?

Amazon Location Service memungkinkan Anda menambahkan data lokasi dan fungsionalitas ke aplikasi, yang mencakup kemampuan seperti peta, tempat menarik, geocoding, perutean, geofences, dan pelacakan. Amazon Location menyediakan layanan berbasis lokasi (LBS) menggunakan data berkualitas tinggi dari penyedia global dan tepercaya Esri, Grab, dan HERE. Dengan data yang terjangkau, kemampuan pelacakan dan geofencing, dan metrik bawaan untuk pemantauan kesehatan, Anda dapat membangun aplikasi canggih yang mendukung lokasi.

Dengan Lokasi Amazon, Anda mempertahankan kendali atas data organisasi Anda. Lokasi Amazon menganonimkan semua kueri yang dikirim ke penyedia data dengan menghapus metadata pelanggan dan informasi akun. Selain itu, informasi lokasi pelacakan dan geofencing yang sensitif,

seperti lokasi fasilitas, aset, dan personel, tidak meninggalkan AWS akun Anda sama sekali. Ini membantu Anda melindungi informasi sensitif dari pihak ketiga, melindungi privasi pengguna, dan mengurangi risiko keamanan aplikasi Anda. Dengan Amazon Location, Amazon dan pihak ketiga tidak memiliki hak untuk menjual data Anda atau menggunakannya untuk iklan.

Lokasi Amazon sepenuhnya terintegrasi dengan layanan seperti AWS CloudTrail, Amazon, Amazon CloudWatch EventBridge, dan AWS Identity and Access Management (IAM). Lokasi Amazon menyederhanakan alur kerja pengembangan Anda dengan integrasi data, dan melacak aplikasi dengan cepat ke produksi dengan fitur pemantauan, keamanan, dan kepatuhan bawaan.

Untuk sorotan, detail produk, dan harga, lihat halaman layanan untuk [Amazon Location Service](#).

## Fitur utama di Lokasi Amazon

Lokasi Amazon menyediakan fitur-fitur berikut:

### Peta

Amazon Location Service Maps memungkinkan Anda memvisualisasikan informasi lokasi dan merupakan dasar dari banyak kemampuan layanan berbasis lokasi. Amazon Location Service menyediakan map tile dengan gaya berbeda yang bersumber dari penyedia data lokasi global Esri, Grab, dan HERE, serta Open data maps.

### Tempat

Amazon Location Service Places memungkinkan Anda mengintegrasikan fungsionalitas penelusuran ke dalam aplikasi, mengonversi alamat menjadi koordinat geografis di lintang dan bujur (geocoding), dan mengonversi koordinat menjadi alamat jalan (reverse geocoding). Amazon Location Service menyediakan data geospasial berkualitas tinggi dari Esri, Grab, dan HERE untuk mendukung fungsi Places.

### Perutean

Amazon Location Service Routes memungkinkan Anda menemukan rute dan memperkirakan waktu perjalanan berdasarkan informasi up-to-date jalan raya dan lalu lintas langsung. Buat fitur yang memungkinkan aplikasi Anda meminta waktu, jarak, dan arah perjalanan antara dua lokasi mana pun. Hitung waktu dan jarak untuk matriks rute yang akan digunakan dalam perencanaan rute.

## Geofencing

Amazon Location Service Geofences memungkinkan Anda memberi aplikasi kemampuan untuk mendeteksi dan bertindak saat perangkat masuk atau keluar dari batas geografis yang ditentukan yang dikenal sebagai geofence. Secara otomatis mengirim acara masuk atau keluar ke Amazon EventBridge ketika pelanggaran geofence terdeteksi. Ini memungkinkan Anda memulai tindakan hilir seperti mengirim pemberitahuan ke target.

## Pelacak

Amazon Location Service Trackers memungkinkan Anda mengambil lokasi saat ini dan historis perangkat yang menjalankan aplikasi berkemampuan pelacakan. Anda juga dapat menautkan pelacak dengan geofences Amazon Location Service untuk mengevaluasi pembaruan lokasi dari perangkat Anda terhadap geofences Anda secara otomatis. Pelacak dapat membantu Anda mengurangi biaya dengan memfilter pembaruan posisi yang belum dipindahkan sebelum menyimpan atau mengevaluasinya terhadap geofences.

Saat Anda menggunakan pelacak, informasi lokasi sensitif pada perangkat yang dilacak tidak meninggalkan akun Anda AWS. Ini membantu melindungi informasi sensitif dari pihak ketiga, melindungi privasi pengguna, dan mengurangi risiko keamanan.

## Layanan yang dapat Anda gunakan dengan Lokasi Amazon

Gunakan layanan berikut bersama dengan Amazon Location Service.

### Pemantauan dan manajemen terpadu

Amazon Location Service terintegrasi dengan Amazon CloudWatch, AWS CloudTrail, dan Amazon EventBridge untuk pemantauan dan manajemen data yang efisien:

- Amazon CloudWatch — Lihat metrik tentang penggunaan dan kesehatan layanan, termasuk permintaan, latensi, kesalahan, dan log. Untuk informasi selengkapnya, lihat [the section called “Pemantauan CloudWatch dengan”](#).
- AWS CloudTrail— Log dan pantau panggilan API Anda, yang mencakup tindakan yang diambil oleh pengguna, peran, atau layanan AWS. Untuk informasi selengkapnya, lihat [the section called “Menggunakan CloudTrail dengan Lokasi Amazon”](#).
- Amazon EventBridge - Aktifkan arsitektur aplikasi berbasis peristiwa sehingga Anda dapat menggunakan AWS Lambda fungsi untuk mengaktifkan bagian lain dari aplikasi dan alur kerja

Anda. Untuk informasi selengkapnya, lihat [the section called “Bereaksi terhadap peristiwa dengan EventBridge”](#).

## Alat developer

Amazon Location Service menawarkan berbagai alat bagi pengembang untuk membangun aplikasi yang mendukung lokasi. Ini termasuk AWS SDK standar, SDK seluler dan web, dan kode sampel untuk menggabungkannya dengan pustaka sumber terbuka seperti MapLibre. Gunakan [konsol Amazon Location Service](#) untuk mempelajari sumber daya, dan untuk memulai dengan alat pembelajaran visual dan interaktif.

# Mulai cepat dengan Amazon Location Service

Cara paling efisien untuk memulai Amazon Location Service adalah dengan menggunakan [konsol Amazon Location](#). Anda dapat membuat dan mengelola sumber daya dan mencoba fungsionalitas Lokasi Amazon menggunakan [halaman Jelajahi](#).

## Note

Untuk menggunakan konsol Amazon Location Service, atau mengikuti sisa tutorial ini, mengharuskan Anda menyelesaikan terlebih dahulu [Prasyarat untuk menggunakan Amazon Location Service](#), termasuk membuat AWS akun, dan mengizinkan akses ke Lokasi Amazon.

Untuk mulai belajar tentang Amazon Location API, gunakan tutorial berikut untuk membuat aplikasi sederhana yang menampilkan peta interaktif dan menggunakan fungsionalitas pencarian. Ada tiga versi tutorial: satu menunjukkan cara membuat halaman web sederhana menggunakan JavaScript, yang kedua menunjukkan hal yang sama untuk aplikasi Android menggunakan Kotlin, dan yang ketiga menunjukkan hal yang sama untuk aplikasi iOS menggunakan Swift.

## Topik

- [Membuat aplikasi web](#)
- [Membuat aplikasi Android](#)
- [Membuat aplikasi iOS](#)

## Membuat aplikasi web

Di bagian ini, Anda akan membuat halaman web statis dengan peta dan kemampuan untuk mencari di lokasi. Pertama, Anda akan membuat sumber daya Lokasi Amazon dan membuat kunci API untuk aplikasi Anda.

## Topik

- [Membuat sumber daya Lokasi Amazon untuk aplikasi Anda](#)
- [Menyiapkan otentikasi untuk aplikasi Anda](#)
- [Membuat HTML untuk aplikasi Anda](#)
- [Menambahkan peta interaktif ke aplikasi Anda](#)

- [Menambahkan pencarian ke aplikasi Anda](#)
- [Melihat aplikasi akhir](#)
- [Apa selanjutnya](#)

## Membuat sumber daya Lokasi Amazon untuk aplikasi Anda

Jika Anda belum memilikinya, Anda harus membuat sumber daya Lokasi Amazon yang akan digunakan aplikasi Anda. Di sini, Anda membuat sumber daya peta untuk menampilkan peta di aplikasi Anda, dan indeks tempat untuk mencari lokasi di peta.

Untuk menambahkan sumber daya lokasi ke aplikasi Anda


1. Pilih gaya peta yang ingin Anda gunakan.
  - a. Di konsol Lokasi Amazon, pada halaman [Peta](#), pilih Buat peta untuk melihat pratinjau gaya peta.
  - b. Tambahkan Nama dan Deskripsi untuk sumber daya peta baru. Catat nama yang Anda gunakan untuk sumber daya peta. Anda akan membutuhkannya saat membuat file skrip Anda nanti di tutorial.
  - c. Pilih peta.

### Note

Memilih gaya peta juga memilih penyedia data peta mana yang akan Anda gunakan. Jika aplikasi Anda melacak atau merutekan aset yang Anda gunakan dalam bisnis Anda, seperti kendaraan pengiriman atau karyawan, Anda hanya dapat menggunakan HERE sebagai penyedia geolokasi Anda. Untuk informasi selengkapnya, lihat bagian 82 dari [ketentuan AWS layanan](#).

- d. Setuju dengan Syarat dan Ketentuan Lokasi Amazon, lalu pilih Buat peta. Anda dapat berinteraksi dengan peta yang Anda pilih: memperbesar, memperkecil, atau menggeser ke segala arah.
  - e. Catat Nama Sumber Daya Amazon (ARN) yang ditampilkan untuk sumber daya peta baru Anda. Anda akan menggunakannya untuk membuat otentikasi yang benar nanti dalam tutorial ini.
2. Pilih indeks tempat yang ingin Anda gunakan.

- a. Di konsol Lokasi Amazon di halaman [Indeks tempat](#), pilih Buat indeks tempat.
- b. Tambahkan Nama dan Deskripsi untuk sumber daya indeks tempat baru. Catat nama yang Anda gunakan untuk sumber daya indeks tempat. Anda akan membutuhkannya saat membuat file skrip Anda nanti di tutorial.
- c. Pilih penyedia data.

 Note


Dalam kebanyakan kasus, pilih penyedia data yang cocok dengan penyedia peta yang sudah Anda pilih. Ini membantu memastikan bahwa pencarian akan cocok dengan peta.

Jika aplikasi Anda melacak atau merutekan aset yang Anda gunakan dalam bisnis Anda, seperti kendaraan pengiriman atau karyawan, Anda hanya dapat menggunakan HERE sebagai penyedia geolokasi Anda. Untuk informasi selengkapnya, lihat bagian 82 dari [ketentuan AWS layanan](#).

- d. Pilih opsi Penyimpanan data. Untuk tutorial ini, hasilnya tidak disimpan, sehingga Anda dapat memilih Tidak, sekali pakai saja.
- e. Setuju dengan Syarat dan Ketentuan Lokasi Amazon, lalu pilih Buat indeks tempat.
- f. Catat ARN yang ditampilkan untuk sumber daya indeks tempat baru Anda. Anda akan menggunakannya untuk membuat otentikasi yang benar di bagian selanjutnya dari tutorial ini.

## Menyiapkan otentikasi untuk aplikasi Anda

Aplikasi yang Anda buat dalam tutorial ini memiliki penggunaan anonim, artinya pengguna Anda tidak diharuskan masuk AWS untuk menggunakan aplikasi. Namun, secara default, Amazon Location Service API memerlukan otentikasi untuk digunakan. Anda dapat menggunakan Amazon Cognito atau kunci API untuk memberikan otentikasi dan otorisasi bagi pengguna anonim. Dalam tutorial ini, Anda akan membuat kunci API untuk digunakan dalam aplikasi sampel.

 Note

Untuk informasi selengkapnya tentang menggunakan kunci API atau Amazon Cognito dengan Amazon Location Service, lihat. [Memberikan akses ke Amazon Location Service](#)

## Untuk menyiapkan otentikasi untuk aplikasi Anda

1. Buka [konsol Lokasi Amazon](#), dan pilih kunci API dari menu kiri.
2. Pilih Buat kunci API.

### Important

Kunci API yang Anda buat harus sama Akun AWS dan AWS Region sebagai resource Amazon Location Service yang Anda buat di bagian sebelumnya.

3. Salah satu halaman Create API key, isi informasi berikut.
  - Nama - Nama untuk kunci API Anda, seperti MyWebAppKey.
  - Sumber Daya — Pilih sumber daya indeks Peta Lokasi Amazon dan Tempat yang Anda buat di bagian sebelumnya. Anda dapat menambahkan lebih dari satu sumber daya dengan memilih Tambah sumber daya. Ini akan memungkinkan kunci API untuk digunakan dengan sumber daya tersebut.
  - Tindakan - Tentukan tindakan yang ingin Anda otorisasi dengan kunci API ini. Anda harus memilih setidaknya geo: GetMap \* dan geo: SearchPlaceIndexForPosition sehingga tutorial akan berfungsi seperti yang diharapkan.
  - Anda dapat menambahkan Deskripsi, Waktu kedaluwarsa, atau Tag secara opsional ke kunci API Anda. Anda juga dapat menambahkan referer (seperti \*.example.com), untuk membatasi kunci hanya digunakan dari domain tertentu. Ini berarti bahwa tutorial hanya akan bekerja dari domain itu.

### Note

Anda disarankan untuk melindungi penggunaan kunci API Anda dengan menetapkan waktu kedaluwarsa atau perujuk, jika tidak keduanya.

4. Pilih Buat kunci API untuk membuat kunci API.
5. Pilih Tampilkan kunci API, dan salin nilai kunci untuk digunakan nanti dalam tutorial. Itu akan dalam bentuk `v1.public.a1b2c3d4...`



**⚠ Important**

Anda akan memerlukan kunci ini saat menulis kode untuk aplikasi Anda nanti dalam tutorial ini.

## Membuat HTML untuk aplikasi Anda

Dalam tutorial ini, Anda akan membuat halaman HTML statis yang menyematkan peta, dan memungkinkan pengguna untuk menemukan apa yang ada di lokasi di peta. Aplikasi ini akan terdiri dari tiga file: file HTML dan file CSS untuk halaman web, dan file JavaScript (.js) untuk kode yang membuat peta dan merespons interaksi pengguna dan peristiwa peta.

Pertama, mari kita buat kerangka HTML dan CSS yang akan digunakan untuk aplikasi. Ini akan menjadi halaman sederhana dengan `<div>` elemen untuk menahan wadah peta dan `<pre>` elemen untuk menunjukkan respons JSON terhadap kueri Anda.

Untuk membuat HTML untuk aplikasi mulai cepat Anda

1. Buat file baru bernama `quickstart.html`.
2. Edit file di editor teks atau lingkungan pilihan Anda. Tambahkan HTML berikut ke file.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Quick start tutorial</title>

    <!-- Styles -->
    <link href="main.css" rel="stylesheet" />
  </head>

  <body>
    <header>
      <h1>Quick start tutorial</h1>
    </header>
    <main>
      <div id="map"></div>
      <aside>
        <h2>JSON Response</h2>
```

```
<pre id="response"></pre>
</aside>
</main>
<footer>This is a simple Amazon Location Service app. Pan and zoom. Click to
see details about entities close to a point.</footer>

</body>
</html>
```

HTML ini memiliki pointer ke file CSS yang akan Anda buat pada langkah berikutnya, beberapa elemen placeholder untuk aplikasi, dan beberapa teks penjelasan.

Ada dua elemen placeholder yang akan Anda gunakan nanti dalam tutorial ini. Yang pertama adalah `<div id="map">` elemen, yang akan memegang kontrol peta. Yang kedua adalah `<pre id="response">` elemen, yang akan menunjukkan hasil pencarian di peta.

### 3. Simpan file Anda.

Sekarang tambahkan CSS untuk halaman web. Ini akan mengatur gaya teks dan elemen placeholder untuk aplikasi.

Untuk membuat CSS untuk aplikasi mulai cepat Anda

1. Buat file baru bernama `main.css`, di folder yang sama dengan file `quickstart.html` yang dibuat dalam prosedur sebelumnya.
2. Edit file di editor apa pun yang ingin Anda gunakan. Tambahkan teks berikut ke file.

```
* {
  box-sizing: border-box;
  font-family: Arial, Helvetica, sans-serif;
}

body {
  margin: 0;
}

header {
  background: #000000;
  padding: 0.5rem;
}
```

```
h1 {
  margin: 0;
  text-align: center;
  font-size: 1.5rem;
  color: #ffffff;
}

main {
  display: flex;
  min-height: calc(100vh - 94px);
}

#map {
  flex: 1;
}

aside {
  overflow-y: auto;
  flex: 0 0 30%;
  max-height: calc(100vh - 94px);
  box-shadow: 0 1px 1px 0 #001c244d, 1px 1px 1px 0 #001c2426, -1px 1px 1px 0
#001c2426;
  background: #f9f9f9;
  padding: 1rem;
}

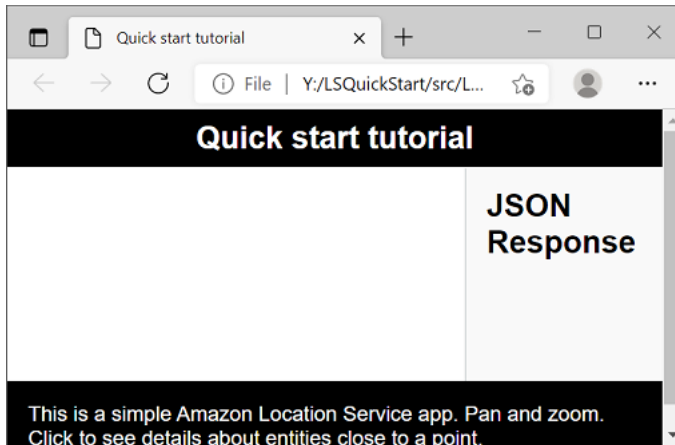
h2 {
  margin: 0;
}

pre {
  white-space: pre-wrap;
  font-family: monospace;
  color: #16191f;
}

footer {
  background: #000000;
  padding: 1rem;
  color: #ffffff;
}
```

Ini menetapkan peta untuk mengisi ruang yang tidak digunakan oleh hal lain, menetapkan area untuk tanggapan kami untuk mengambil 30% dari lebar aplikasi, dan menetapkan warna dan gaya untuk judul dan teks penjelasan.

3. Simpan file tersebut.
4. Anda sekarang dapat melihat `quickstart.html` file di browser untuk melihat tata letak aplikasi.



Selanjutnya, Anda akan menambahkan kontrol peta ke aplikasi.

## Menambahkan peta interaktif ke aplikasi Anda

Sekarang Anda memiliki kerangka kerja dan placeholder div, Anda dapat menambahkan kontrol peta ke aplikasi Anda. Tutorial ini menggunakan [MapLibre GL JS](#) sebagai kontrol peta, mendapatkan data dari Amazon Location Service. Anda juga akan menggunakan [JavaScript Pembantu otentikasi](#) untuk memfasilitasi penandatanganan panggilan ke Amazon Location API dengan kunci API Anda.

Untuk menambahkan peta interaktif ke aplikasi Anda

1. Buka `quickstart.html` file yang Anda buat di bagian sebelumnya.
2. Tambahkan referensi ke perpustakaan yang dibutuhkan, dan file skrip yang akan Anda buat. Perubahan yang perlu Anda lakukan ditampilkan **digreen**.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Quick start tutorial</title>
```

```
<!-- Styles -->
<link href="https://unpkg.com/maplibre-gl@3.x/dist/maplibre-gl.css"
rel="stylesheet" />
<link href="main.css" rel="stylesheet" />
</head>

<body>
  ...
  <footer>This is a simple Amazon Location Service app. Pan and zoom. Click to
see details about entities close to a point.</footer>

  <!-- JavaScript dependencies -->
  <script src="https://unpkg.com/maplibre-gl@3.x/dist/maplibre-gl.js"></script>
  <script src="https://unpkg.com/@aws/amazon-location-client@1.x/dist/
amazonLocationClient.js"></script>
  <script src="https://unpkg.com/@aws/amazon-location-utilities-auth-helper@1.x/
dist/amazonLocationAuthHelper.js"></script>

  <!-- JavaScript for the app -->
  <script src="main.js"></script>
</body>
</html>
```

Ini menambahkan dependensi berikut ke aplikasi Anda:

- MapLibre GL JS. Pustaka dan stylesheet ini menyertakan kontrol peta yang menampilkan ubin peta dan mencakup interaktivitas, seperti pan dan zoom. Kontrol juga memungkinkan ekstensi, seperti menggambar fitur Anda sendiri di peta.
- Klien Lokasi Amazon. Ini menyediakan antarmuka untuk fungsionalitas Lokasi Amazon yang diperlukan untuk mendapatkan data peta, dan untuk mencari tempat di peta. Klien Lokasi Amazon didasarkan pada AWS SDK untuk JavaScript v3.
- Pembantu Otentikasi Lokasi Amazon. Ini menyediakan fungsi bermanfaat untuk mengautentikasi Amazon Location Service dengan kunci API atau Amazon Cognito.

Langkah ini juga menambahkan referensimain.js, yang akan Anda buat selanjutnya.

3. Simpan file quickstart.html.
4. Buat file baru yang disebut main.js di folder yang sama dengan file HTML dan CSS Anda, dan buka untuk diedit.

5. Tambahkan skrip berikut ke file Anda. Teks berwarna *merah* harus diganti dengan nilai kunci API, nama sumber daya peta, dan nama sumber daya tempat yang Anda buat sebelumnya, serta pengenal wilayah untuk wilayah Anda (seperti `us-east-1`).

```
// Amazon Location Service resource names:
const mapName = "explore.map";
const placesName = "explore.place";
const region = "your_region";
const apiKey = "v1.public.a1b2c3d4...

// Initialize a map
async function initializeMap() {
  const mlglMap = new maplibregl.Map({
    container: "map", // HTML element ID of map element
    center: [-77.03674, 38.891602], // Initial map centerpoint
    zoom: 16, // Initial map zoom
    style: 'https://maps.geo.${region}.amazonaws.com/maps/v0/maps/${mapName}/style-
descriptor?key=${apiKey}', // Defines the appearance of the map and authenticates
using an API key
  });

  // Add navigation control to the top left of the map
  mlglMap.addControl(new maplibregl.NavigationControl(), "top-left");

  return mlglMap;
}

async function main() {
  // Initialize map and Amazon Location SDK client:
  const map = await initializeMap();
}

main();
```

Kode ini mengatur sumber daya Lokasi Amazon, lalu mengonfigurasi dan menginisialisasi kontrol peta MapLibre GL JS dan menempatkannya di `<div>` elemen Anda dengan id. `map`

`initializeMap()` Fungsi ini penting untuk dipahami. Ini menciptakan kontrol MapLibre peta baru (disebut `mlglMap` secara lokal, tetapi dipanggil `map` di sisa kode) yang digunakan untuk membuat peta dalam aplikasi Anda.

```
// Initialize the map
```

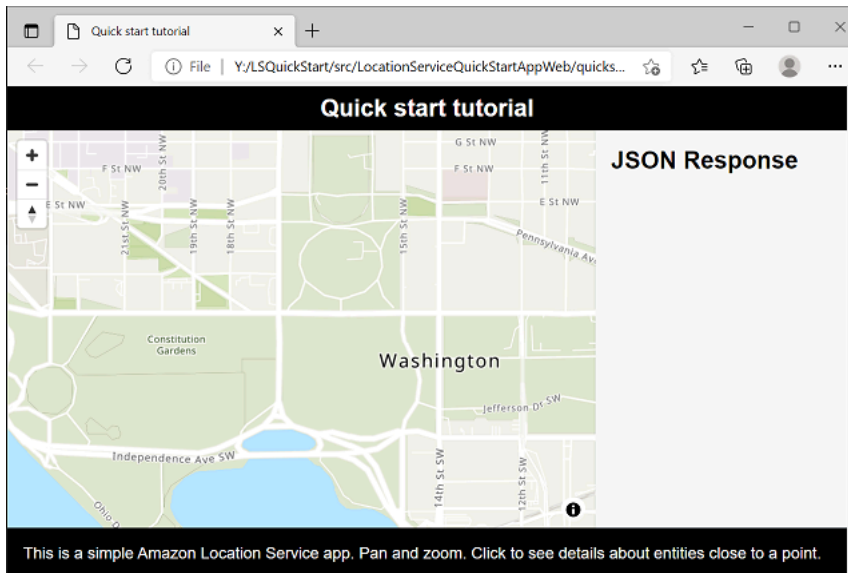
```
const mlg1Map = new maplibregl.Map({
  container: "map", // HTML element ID of map element
  center: [-77.03674, 38.891602], // Initial map centerpoint
  zoom: 16, // Initial map zoom
  style: 'https://maps.geo.${region}.amazonaws.com/maps/v0/maps/${mapName}/style-
descriptor?key=${apiKey}', // Defines the appearance of the map and authenticates
using an API key
});
```

Saat Anda membuat kontrol MapLibre peta baru, parameter yang Anda lewati menunjukkan status awal kontrol peta. Di sini, kami mengatur parameter berikut.

- Kontainer HTML, yang menggunakan elemen peta div dalam HTML kami.
  - Pusat awal peta ke suatu titik di Washington, DC.
  - Tingkat zoom ke 16 (diperbesar ke lingkungan atau tingkat blok).
  - Gaya yang digunakan untuk peta, MapLibre yang memberikan URL untuk digunakan untuk mendapatkan ubin peta dan informasi lainnya untuk membuat peta. Perhatikan bahwa URL ini menyertakan kunci API Anda untuk otentikasi.
6. Simpan JavaScript file Anda, dan buka dengan browser. Anda sekarang memiliki peta di halaman Anda, di mana Anda dapat menggunakan tindakan pan dan zoom.

#### Note

Anda dapat menggunakan aplikasi ini untuk melihat bagaimana kontrol MapLibre peta berperilaku. Anda dapat mencoba menggunakan Ctrl atau Shift saat menggunakan operasi menyeret, untuk melihat cara lain untuk berinteraksi dengan peta. Semua fungsi ini dapat disesuaikan.



Aplikasi Anda hampir selesai. Di bagian selanjutnya, Anda akan menangani pemilihan lokasi di peta, dan menunjukkan alamat lokasi yang dipilih. Anda juga akan menampilkan JSON yang dihasilkan di halaman, untuk melihat hasil lengkapnya.

## Menambahkan pencarian ke aplikasi Anda

Langkah terakhir untuk aplikasi Anda adalah menambahkan pencarian di peta. Dalam hal ini, Anda akan menambahkan pencarian geocoding terbalik, di mana Anda menemukan item di lokasi.

### Note

Amazon Location Service juga menyediakan kemampuan untuk mencari berdasarkan nama atau alamat untuk menemukan lokasi tempat di peta.

Untuk menambahkan fungsionalitas penelusuran ke aplikasi Anda

1. Buka `main.js` file yang Anda buat di bagian sebelumnya.
2. Ubah `main` fungsi, seperti yang ditunjukkan. Perubahan yang perlu Anda lakukan ditampilkan **digreen**.

```
async function main() {  
  // Create an authentication helper instance using an API key
```



```
const authHelper = await amazonLocationAuthHelper.withAPIKey(apiKey);

// Initialize map and Amazon Location SDK client:
const map = await initializeMap();

const client = new amazonLocationClient.LocationClient({
  region,
  ...authHelper.getLocationClientConfig(), // Provides configuration required to
make requests to Amazon Location
});

// On mouse click, display marker and get results:
map.on("click", async function (e) {
  // Set up parameters for search call
  let params = {
    IndexName: placesName,
    Position: [e.lngLat.lng, e.lngLat.lat],
    Language: "en",
    MaxResults: "5",
  };

  // Set up command to search for results around clicked point
  const searchCommand = new
amazonLocationClient.SearchPlaceIndexForPositionCommand(params);

  try {
    // Make request to search for results around clicked point
    const data = await client.send(searchCommand);

    // Write JSON response data to HTML
    document.querySelector("#response").textContent = JSON.stringify(data,
undefined, 2);

    // Display place label in an alert box
    alert(data.Results[0].Place.Label);
  } catch (error) {
    // Write JSON response error to HTML
    document.querySelector("#response").textContent = JSON.stringify(error,
undefined, 2);

    // Display error in an alert box
    alert("There was an error searching.");
  }
});
```

```
    }  
  });  
}
```

Kode ini dimulai dengan menyiapkan helper autentikasi Lokasi Amazon untuk menggunakan kunci API Anda.

```
const authHelper = await amazonLocationAuthHelper.withAPIKey(apiKey);
```

Kemudian ia menggunakan pembantu otentikasi itu, dan wilayah yang Anda gunakan untuk membuat klien Lokasi Amazon baru.

```
const client = new amazonLocationClient.LocationClient({  
  region,  
  ...authHelper.getLocationClientConfig(),  
});
```

Selanjutnya, kode merespons pengguna yang memilih tempat pada kontrol peta. Ini dilakukan dengan menangkap acara yang MapLibre disediakan untuk `click`.

```
map.on("click", async function(e) {  
  ...  
});
```

MapLibre `clickAcara` ini menyediakan parameter yang mencakup garis lintang dan bujur yang dipilih pengguna (`e.lngLat`). Dalam `click` acara tersebut, kode membuat `searchPlaceIndexForPositionCommand` untuk menemukan entitas pada garis lintang dan bujur yang diberikan.

```
// Set up parameters for search call  
let params = {  
  IndexName: placesName,  
  Position: [e.lngLat.lng, e.lngLat.lat],  
  Language: "en",  
  MaxResults: "5"  
};  
  
// Set up command to search for results around clicked point  
const searchCommand = new  
amazonLocationClient.SearchPlaceIndexForPositionCommand(params);
```

```
try {
  // Make request to search for results around clicked point
  const data = await client.send(searchCommand);
  ...
});
```

Di sini, `IndexName` adalah nama sumber daya Indeks Tempat yang Anda buat sebelumnya, `Position` adalah garis lintang dan bujur untuk dicari, `Language` adalah bahasa pilihan untuk hasil, dan memberi tahu Lokasi `MaxResults` Amazon untuk mengembalikan hanya maksimal lima hasil.

Kode yang tersisa memeriksa kesalahan, dan kemudian menampilkan hasil pencarian dalam `<pre>` elemen yang disebut `response`, dan menunjukkan hasil teratas dalam kotak peringatan.

3. (Opsional) Jika Anda menyimpan dan membuka `quickstart.html` file di browser sekarang, memilih lokasi di peta akan menunjukkan nama atau alamat tempat yang Anda pilih.
4. Langkah terakhir dalam aplikasi adalah menggunakan MapLibre fungsionalitas untuk menambahkan penanda di tempat yang dipilih pengguna. Ubah `main` fungsinya sebagai berikut. Perubahan yang perlu Anda lakukan ditampilkan **digreen**.

```
async function main() {
  // Create an authentication helper instance using an API key
  const authHelper = await amazonLocationAuthHelper.withAPIKey(apiKey);

  // Initialize map and Amazon Location SDK client
  const map = await initializeMap();
  const client = new amazonLocationClient.LocationClient({
    region,
    ...authHelper.getLocationClientConfig(), // Provides configuration required to
    make requests to Amazon Location
  });

  // Variable to hold marker that will be rendered on click
  let marker;

  // On mouse click, display marker and get results:
  map.on("click", async function (e) {
    // Remove any existing marker
    if (marker) {
      marker.remove();
    }
  });
}
```

```

// Render a marker on clicked point
marker = new maplibregl.Marker().setLngLat([e.lngLat.lng,
e.lngLat.lat]).addTo(map);

// Set up parameters for search call
let params = {
  IndexName: placesName,
  Position: [e.lngLat.lng, e.lngLat.lat],
  Language: "en",
  MaxResults: "5",
};

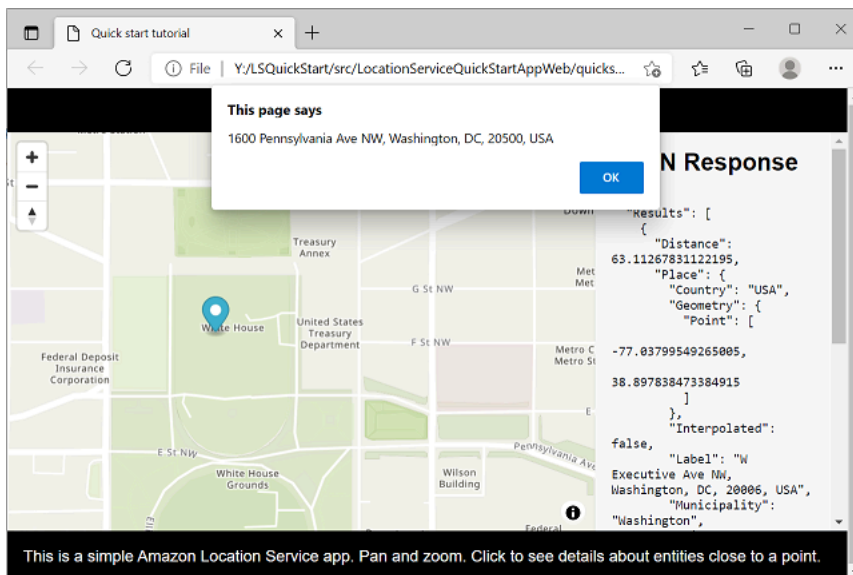
// Set up command to search for results around clicked point
const searchCommand = new
amazonLocationClient.SearchPlaceIndexForPositionCommand(params);

...

```

Kode ini mendeklarasikan `marker` variabel, yang diisi setiap kali pengguna memilih lokasi, menunjukkan di mana mereka memilih. Penanda secara otomatis dirender oleh kontrol peta, setelah ditambahkan ke peta dengan `.addTo(map)`; . Kode juga memeriksa penanda sebelumnya, dan menghapusnya, sehingga hanya ada 1 penanda di layar sekaligus.

5. Simpan `main.js` file, dan buka `quickstart.html` file di browser. Anda dapat menggeser dan memperbesar peta, seperti sebelumnya, tetapi sekarang jika Anda memilih lokasi, Anda akan melihat detail tentang lokasi yang Anda pilih.



Aplikasi mulai cepat Anda selesai. Tutorial ini telah menunjukkan cara membuat aplikasi HTML statis yang:

- Membuat peta yang dapat berinteraksi dengan pengguna.
- Menangani acara peta (`click`).
- Memanggil Amazon Location Service API, khususnya untuk mencari peta di lokasi, menggunakan `searchPlaceIndexForPosition`.
- Menggunakan kontrol MapLibre peta untuk menambahkan penanda.

## Melihat aplikasi akhir

Kode sumber akhir untuk aplikasi ini termasuk dalam bagian ini. Anda juga dapat menemukan proyek akhir [di GitHub](#).

[Anda juga dapat menemukan versi aplikasi yang menggunakan Amazon Cognito alih-alih kunci API. \[GitHub\]\(#\)](#)

### Overview

Pilih setiap tab untuk melihat kode sumber akhir dari file dalam tutorial mulai cepat ini.

File-file tersebut adalah:

- `quickstart.html` - kerangka kerja untuk aplikasi Anda, termasuk pemegang elemen HTML untuk peta dan hasil pencarian.
- `main.css` — stylesheet untuk aplikasi.
- `main.js` — skrip untuk aplikasi Anda yang mengautentikasi pengguna, membuat peta, dan mencari pada suatu `click` peristiwa.

### quickstart.html

Kerangka HTML untuk aplikasi mulai cepat.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Quick start tutorial</title>
```

```
<!-- Styles -->
<link href="https://unpkg.com/maplibre-gl@3.x/dist/maplibre-gl.css"
rel="stylesheet" />
<link href="main.css" rel="stylesheet" />
</head>

<body>
  ...
  <footer>This is a simple Amazon Location Service app. Pan and zoom. Click to see
details about entities close to a point.</footer>

  <!-- JavaScript dependencies -->
  <script src="https://unpkg.com/maplibre-gl@3.x/dist/maplibre-gl.js"></script>
  <script src="https://unpkg.com/@aws/amazon-location-client@1.x/dist/
amazonLocationClient.js"></script>
  <script src="https://unpkg.com/@aws/amazon-location-utilities-auth-helper@1.x/
dist/amazonLocationAuthHelper.js"></script>

  <!-- JavaScript for the app -->
  <script src="main.js"></script>
</body>
</html>
```

## main.css

Stylesheet untuk aplikasi mulai cepat.

```
* {
  box-sizing: border-box;
  font-family: Arial, Helvetica, sans-serif;
}

body {
  margin: 0;
}

header {
  background: #000000;
  padding: 0.5rem;
}

h1 {
  margin: 0;
  text-align: center;
```

```
    font-size: 1.5rem;
    color: #ffffff;
}

main {
  display: flex;
  min-height: calc(100vh - 94px);
}

#map {
  flex: 1;
}

aside {
  overflow-y: auto;
  flex: 0 0 30%;
  max-height: calc(100vh - 94px);
  box-shadow: 0 1px 1px 0 #001c244d, 1px 1px 1px 0 #001c2426, -1px 1px 1px 0 #001c2426;
  background: #f9f9f9;
  padding: 1rem;
}

h2 {
  margin: 0;
}

pre {
  white-space: pre-wrap;
  font-family: monospace;
  color: #16191f;
}

footer {
  background: #000000;
  padding: 1rem;
  color: #ffffff;
}
```

## main.js

Kode untuk aplikasi mulai cepat. Teks berwarna *merah* harus diganti dengan nama objek Lokasi Amazon yang sesuai.

```
// Amazon Location Service resource names:
const mapName = "explore.map";
const placesName = "explore.place";
const region = "your_region";
const apiKey = "v1.public.a1b2c3d4...

// Initialize a map
async function initializeMap() {
  // Initialize the map
  const mlg1Map = new maplibregl.Map({
    container: "map", // HTML element ID of map element
    center: [-77.03674, 38.891602], // Initial map centerpoint
    zoom: 16, // Initial map zoom
    style: `https://maps.geo.${region}.amazonaws.com/maps/v0/maps/${mapName}/style-
descriptor?key=${apiKey}`, // Defines the appearance of the map and authenticates
using an API key
  });

  // Add navigation control to the top left of the map
  mlg1Map.addControl(new maplibregl.NavigationControl(), "top-left");

  return mlg1Map;
}

async function main() {
  // Create an authentication helper instance using an API key
  const authHelper = await amazonLocationAuthHelper.withAPIKey(apiKey);

  // Initialize map and Amazon Location SDK client
  const map = await initializeMap();
  const client = new amazonLocationClient.LocationClient({
    region,
    ...authHelper.getLocationClientConfig(), // Provides configuration required to
make requests to Amazon Location
  });

  // Variable to hold marker that will be rendered on click
  let marker;

  // On mouse click, display marker and get results:
  map.on("click", async function (e) {
    // Remove any existing marker
    if (marker) {
```



```
        marker.remove();
    }

    // Render a marker on clicked point
    marker = new maplibregl.Marker().setLngLat([e.lngLat.lng,
e.lngLat.lat]).addTo(map);

    // Set up parameters for search call
    let params = {
        IndexName: placesName,
        Position: [e.lngLat.lng, e.lngLat.lat],
        Language: "en",
        MaxResults: "5",
    };

    // Set up command to search for results around clicked point
    const searchCommand = new
amazonLocationClient.SearchPlaceIndexForPositionCommand(params);

    try {
        // Make request to search for results around clicked point
        const data = await client.send(searchCommand);

        // Write JSON response data to HTML
        document.querySelector("#response").textContent = JSON.stringify(data,
undefined, 2);

        // Display place label in an alert box
        alert(data.Results[0].Place.Label);
    } catch (error) {
        // Write JSON response error to HTML
        document.querySelector("#response").textContent = JSON.stringify(error,
undefined, 2);

        // Display error in an alert box
        alert("There was an error searching.");
    }
});
}

main();
```

## Apa selanjutnya

Anda telah menyelesaikan tutorial mulai cepat, dan harus memiliki gagasan tentang bagaimana Amazon Location Service digunakan untuk membangun aplikasi. Untuk mendapatkan lebih banyak dari Lokasi Amazon, Anda dapat melihat sumber daya berikut:

- Selami lebih dalam [konsep Amazon Location Service](#)
- Dapatkan informasi selengkapnya tentang [cara menggunakan fitur dan fungsionalitas Lokasi Amazon](#)
- Lihat cara memperluas sampel ini dan membangun aplikasi yang lebih kompleks dengan melihat [contoh kode menggunakan Lokasi Amazon](#)

## Membuat aplikasi Android

Di bagian ini, Anda akan membuat aplikasi Android dengan peta, kemampuan untuk mencari di lokasi dan melacak di latar depan. Pertama, Anda akan membuat sumber daya Lokasi Amazon, identitas Amazon Cognito, dan kunci API untuk aplikasi Anda.

Topik

- [Membuat sumber daya Lokasi Amazon untuk aplikasi Anda](#)
- [Menyiapkan otentikasi untuk aplikasi Anda](#)
- [Membuat aplikasi Android dasar](#)
- [Menambahkan peta interaktif ke aplikasi Anda](#)
- [Menambahkan pencarian geocoding terbalik ke aplikasi Anda](#)
- [Menambahkan pelacakan ke aplikasi Anda](#)
- [Apa selanjutnya](#)

## Membuat sumber daya Lokasi Amazon untuk aplikasi Anda

Jika Anda belum memilikinya, Anda harus membuat sumber daya Lokasi Amazon yang akan digunakan aplikasi Anda. Di sini, Anda membuat sumber daya peta untuk menampilkan peta dalam aplikasi Anda, indeks tempat untuk mencari lokasi di peta, dan pelacak untuk melacak objek di seluruh peta.

## Untuk menambahkan sumber daya lokasi ke aplikasi Anda

1. Pilih gaya peta yang ingin Anda gunakan.
  - a. Di konsol Lokasi Amazon, pada halaman [Peta](#), pilih Buat peta untuk melihat pratinjau gaya peta.
  - b. Tambahkan Nama dan Deskripsi untuk sumber daya peta baru. Catat nama yang Anda gunakan untuk sumber daya peta. Anda akan membutuhkannya saat membuat file skrip Anda nanti di tutorial.
  - c. Kami sarankan Anda memilih stlye peta HERE untuk peta Anda.

### Note

Memilih gaya peta juga memilih penyedia data peta mana yang akan Anda gunakan. Jika aplikasi Anda melacak atau merutekan aset yang Anda gunakan dalam bisnis Anda, seperti kendaraan pengiriman atau karyawan, Anda hanya dapat menggunakan HERE sebagai penyedia geolokasi Anda. Untuk informasi selengkapnya, lihat bagian 82 dari [ketentuan AWS layanan](#).

- d. Setuju dengan Syarat dan Ketentuan Lokasi Amazon, lalu pilih Buat peta. Anda dapat berinteraksi dengan peta yang Anda pilih: memperbesar, memperkecil, atau menggeser ke segala arah.
  - e. Catat Nama Sumber Daya Amazon (ARN) yang ditampilkan untuk sumber daya peta baru Anda. Anda akan menggunakannya untuk membuat otentikasi yang benar nanti dalam tutorial ini.
2. Pilih indeks tempat yang ingin Anda gunakan.
  - a. Di konsol Lokasi Amazon di halaman [Indeks tempat](#), pilih Buat indeks tempat.
  - b. Tambahkan Nama dan Deskripsi untuk sumber daya indeks tempat baru. Catat nama yang Anda gunakan untuk sumber daya indeks tempat. Anda akan membutuhkannya saat membuat file skrip Anda nanti di tutorial.
  - c. Pilih penyedia data.

**Note**

Dalam kebanyakan kasus, pilih penyedia data yang cocok dengan penyedia peta yang sudah Anda pilih. Ini membantu memastikan bahwa pencarian akan cocok dengan peta.

Jika aplikasi Anda melacak atau merutekan aset yang Anda gunakan dalam bisnis Anda, seperti kendaraan pengiriman atau karyawan, Anda hanya dapat menggunakan HERE sebagai penyedia geolokasi Anda. Untuk informasi selengkapnya, lihat bagian 82 dari [ketentuan AWS layanan](#).

- d. Pilih opsi Penyimpanan data. Untuk tutorial ini, hasilnya tidak disimpan, sehingga Anda dapat memilih Tidak, sekali pakai saja.
  - e. Setuju dengan Syarat dan Ketentuan Lokasi Amazon, lalu pilih Buat indeks tempat.
  - f. Catat ARN yang ditampilkan untuk sumber daya indeks tempat baru Anda. Anda akan menggunakannya untuk membuat otentikasi yang benar di bagian selanjutnya dari tutorial ini.
3. Untuk membuat pelacak menggunakan konsol Lokasi Amazon.
- a. Buka [konsol Amazon Location Service](#).
  - b. Di panel navigasi kiri, pilih Pelacak.
  - c. Pilih Buat pelacak.
  - d. Isi semua bidang yang diperlukan.
  - e. Di bawah Pemfilteran posisi, kami menyarankan Anda menggunakan pengaturan default: TimeBased
  - f. Pilih Buat pelacak untuk menyelesaikan.

## Menyiapkan otentikasi untuk aplikasi Anda

Aplikasi yang Anda buat dalam tutorial ini memiliki penggunaan anonim, yang berarti bahwa pengguna Anda tidak diharuskan masuk AWS untuk menggunakan aplikasi. Namun, Amazon Location Service API memerlukan otentikasi untuk digunakan. Anda dapat menggunakan kunci API atau Amazon Cognito untuk memberikan otentikasi dan otorisasi bagi pengguna anonim. Tutorial ini akan menggunakan Amazon Cognito dan kunci API untuk mengautentikasi aplikasi Anda.

**Note**

Untuk informasi selengkapnya tentang menggunakan Amazon Cognito atau kunci API dengan Amazon Location Service, lihat. [Memberikan akses ke Amazon Location Service](#)

Tutorial berikut menunjukkan cara mengatur otentikasi untuk peta, indeks tempat, dan pelacak yang Anda buat serta menyiapkan izin untuk Lokasi Amazon.

### Mengatur otentikasi

1. Arahkan ke [konsol Lokasi Amazon](#) dan pilih kunci API dari menu sebelah kiri.
2. Klik 'Buat kunci API'. Ingat bahwa kunci API harus berada di AWS akun dan wilayah yang sama dengan resource Amazon Location Service yang dibuat sebelumnya.
3. Isi detail yang diperlukan di halaman 'Buat kunci API':
  - Nama: Berikan nama untuk kunci API Anda, seperti MyAppKey.
  - Resources: Pilih sumber daya indeks Amazon Location Service Map dan Place yang dibuat sebelumnya. Anda dapat menambahkan beberapa sumber daya dengan memilih 'Tambahkan Sumber Daya'. Ini memungkinkan kunci API untuk digunakan dengan sumber daya tertentu.
  - Tindakan: Tentukan tindakan resmi untuk kunci API ini. Minimal, pilih `geo:GetMap` dan `geo:SearchPlaceIndexForPosition` untuk memastikan fungsi tutorial sebagaimana dimaksud.
  - Opsional Anda dapat menambahkan Deskripsi, Waktu kedaluwarsa, Tag, atau perujuk misalnya `https://www.example.com` untuk membatasi penggunaan kunci ke domain tertentu, memungkinkan tutorial berfungsi hanya dalam domain itu.
4. Klik Buat Kunci API untuk menghasilkan kunci API.
5. Pilih Show API Key dan salin nilai kunci `v1.public.a1b2c3d4` misalnya untuk digunakan nanti dalam tutorial.

### Membuat kebijakan IAM untuk melacak

1. Masuk ke konsol IAM di <https://console.aws.amazon.com/iam/> dengan pengguna Anda yang memiliki izin administrator.
2. Di panel navigasi, pilih Kebijakan.

3. Di panel konten, pilih Buat kebijakan.
4. Pilih opsi JSON, lalu salin dan tempel kebijakan JSON ini ke dalam kotak teks JSON.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "geo:GetMapTile",
        "geo:GetMapStyleDescriptor",
        "geo:GetMapSprites",
        "geo:GetMapGlyphs",
        "geo:SearchPlaceIndexForPosition",
        "geo:GetDevicePositionHistory",
        "geo:BatchUpdateDevicePosition"
      ],
      "Resource": [
        "arn:aws:geo:{Region}:{Account}:map/{MapName}",
        "arn:aws:geo:{Region}:{Account}:place-index/{IndexName}",
        "arn:aws:geo:{Region}:{Account}:tracker/{TrackerName}"
      ]
    }
  ]
}
```

Ini adalah contoh kebijakan untuk Tracking. Untuk menggunakan contoh untuk kebijakan Anda sendiri, ganti, RegionAccount, dan TrackerNameplaceholder.

#### Note

Meskipun kumpulan identitas yang tidak diautentikasi dimaksudkan untuk eksposur di situs internet yang tidak aman, perhatikan bahwa kumpulan tersebut akan ditukar dengan kredensi AWS standar dan terbatas waktu.

Penting untuk mencakup peran IAM yang terkait dengan kumpulan identitas yang tidak diautentikasi dengan tepat. Untuk informasi selengkapnya tentang penggunaan dan kebijakan pelingkupan yang tepat di Amazon Cognito dengan Amazon Location Service, lihat Memberikan akses ke [Amazon](#) Location Service.

5. Pada halaman Tinjau dan Buat, berikan nama untuk bidang nama kebijakan. Tinjau izin yang diberikan oleh kebijakan Anda, lalu pilih Buat Kebijakan untuk menyimpan pekerjaan Anda.

Kebijakan baru muncul di daftar kebijakan terkelola dan siap dilampirkan.

Siapkan otentikasi untuk pelacakan Anda

1. Siapkan otentikasi untuk aplikasi peta Anda di konsol [Amazon Cognito](#).
2. Buka halaman Identity pool.

#### Note

Kumpulan yang Anda buat harus berada di AWS akun dan AWS Wilayah yang sama dengan sumber daya Amazon Location Service yang Anda buat di bagian sebelumnya.

3. Pilih Buat kumpulan Identitas.
4. Dimulai dengan langkah Configure identity pool trust. Untuk otentikasi akses pengguna, pilih Akses tamu, dan tekan berikutnya.
5. Pada halaman Konfigurasi izin pilih Gunakan peran IAM yang ada dan masukkan nama peran IAM yang Anda buat di langkah sebelumnya. Saat siap tekan di sebelah untuk melanjutkan ke langkah berikutnya.
6. Pada halaman Konfigurasi properti, berikan nama untuk kumpulan identitas Anda. Lalu tekan Next.
7. Pada halaman Tinjau dan buat, tinjau semua informasi yang ada lalu tekan Buat kumpulan identitas.
8. Buka halaman Identity pool, dan pilih kumpulan identitas yang baru saja Anda buat. Kemudian salin atau tuliskan IdentityPoolId yang akan Anda gunakan nanti di skrip browser Anda.

## Membuat aplikasi Android dasar

Dalam tutorial ini, Anda akan membuat aplikasi Android yang menyematkan peta dan memungkinkan pengguna untuk menemukan apa yang ada di lokasi di peta.

Pertama, buat aplikasi Kotlin kosong menggunakan wizard proyek baru Android Studio.

## Untuk membuat aplikasi kosong (AndroidStudio)

1. Mulai AndroidStudio. Buka menu, dan pilih File, New, New Project.
2. Dari tab Telepon dan Tablet, pilih Aktivitas Kosong, lalu pilih Berikutnya.
3. Pilih Nama, nama Package, dan Save location untuk aplikasi Anda.
4. Dalam daftar dropdown untuk Bahasa, pilih Kotlin.
5. Pilih Selesai untuk membuat aplikasi kosong Anda.

## Menambahkan peta interaktif ke aplikasi Anda

Sekarang setelah Anda membuat aplikasi dasar, Anda dapat menambahkan kontrol peta ke aplikasi Anda. Tutorial ini menggunakan kunci API untuk mengelola tampilan peta. Kontrol peta itu sendiri adalah bagian dari [pustaka MapLibre Asli](#), dengan kunci API dan MapLibre, dan data peta berasal dari Lokasi Amazon.

Untuk menambahkan peta ke aplikasi Anda, Anda harus melakukan tindakan berikut:

- Tambahkan MapLibre dependensi ke proyek Anda.
- Siapkan kode tampilan peta dengan compose.
- Tulis kode untuk menampilkan peta.

Gunakan prosedur berikut untuk menambahkan peta ke aplikasi Anda:

1. Tambahkan MapLibre dependensi ke proyek Anda
  - a. Di AndroidStudio, pilih menu View, dan pilih Tool Windows, Project. Ini akan membuka jendela Project, yang memberi Anda akses ke semua file dalam proyek Anda.
  - b. Di jendela Project, buka gradle lalu buka `libs.versions.toml` file di tampilan pohon. Ini akan membuka `libs.versions.toml` file untuk diedit. Sekarang tambahkan versi dan data pustaka di bawah ini dalam `libs.versions.toml` file.

```
[versions]
...
auth = "0.2.4"
tracking = "0.2.4"

[libraries]
```



```
...
auth = { group = "software.amazon.location", name = "auth", version.ref =
"auth" }
tracking = { module = "software.amazon.location:tracking", version.ref =
"tracking" }

[plugins]
...
```

- c. Setelah Anda selesai mengedit `libs.versions.toml` file, AndroidStudio harus menyinkronkan ulang proyek. Di bagian atas jendela `libs.versions.toml` pengeditan, AndroidStudio meminta Anda untuk menyinkronkan. Pilih 'Sinkronkan Sekarang' untuk menyinkronkan proyek Anda sebelum melanjutkan.
- d. Di jendela Project, buka Gradle Scripts dalam tampilan pohon dan pilih `build.gradle` file untuk modul aplikasi Anda. Ini akan membuka `build.gradle` file untuk diedit.
- e. Di bagian bawah file, di bagian dependensi, tambahkan dependensi berikut.

```
dependencies {
    ...
    implementation(libs.org.maplibre.gl)
}
```

- f. Setelah Anda selesai menambahkan dependensi Gradle, Android Studio harus menyinkronkan ulang project. Di bagian atas jendela pengeditan `build.gradle`, Android Studio, pilih Sinkronkan Sekarang untuk menyinkronkan project Anda sebelum melanjutkan.
2. Sekarang Anda akan mengatur kode tampilan peta dengan compose. Gunakan langkah-langkah berikut:
    - a. Dari jendela Project, buka App, Java, *nama paket Anda* di tampilan pohon, dan buka folder ui, di dalam folder ui buat direktori tampilan.
    - b. Direktori tampilan dalam membuat `MapLoadScreen.kt` file.
    - c. Tambahkan kode berikut ke `MapLoadScreen.kt` file Anda.

```
import androidx.compose.foundation.layout.Box
import androidx.compose.foundation.layout.fillMaxHeight
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.viewinterop.AndroidView
import org.maplibre.android.maps.OnMapReadyCallback
```

```
@Composable
fun MapLoadScreen(
    mapReadyCallback: OnMapReadyCallback,
) {
    Box(
        modifier = Modifier
            .fillMaxWidth()
            .fillMaxHeight(),
    ) {
        MapView(mapReadyCallback)
    }
}

@Composable
fun MapView(mapReadyCallback: OnMapReadyCallback) {
    AndroidView(
        factory = { context ->
            val mapView = org.maplibre.android.maps.MapView(context)
            mapView.onCreate(null)
            mapView.getMapAsync(mapReadyCallback)
            mapView
        },
    )
}
```

3. Tulis kode untuk menampilkan peta.
  - a. Tambahkan kode berikut ke MainActivity.kt file Anda.

```
// ...other imports
import org.maplibre.android.MapLibre
import org.maplibre.android.camera.CameraPosition
import org.maplibre.android.geometry.LatLng
import org.maplibre.android.maps.MapLibreMap
import org.maplibre.android.maps.OnMapReadyCallback
import org.maplibre.android.maps.Style

class MainActivity : ComponentActivity(), OnMapReadyCallback {
    private val region = "YOUR_AWS_REGION"
    private val mapName = "YOUR_AWS_MAP_NAME"
    private val apiKey = "YOUR_AWS_API_KEY"
    override fun onCreate(savedInstanceState: Bundle?) {
        MapLibre.getInstance(this)
```

```
super.onCreate(savedInstanceState)
setContent {
    TestMapAppTheme {
        Surface(
            modifier = Modifier.fillMaxSize(),
            color = MaterialTheme.colorScheme.background
        ) {
            MapLoadScreen(this)
        }
    }
}

override fun onMapReady(map: MapLibreMap) {
    map.setStyle(
        Style.Builder()
            .fromUri(
                "https://maps.geo.$region.amazonaws.com/maps/v0/maps/
$mapName/style-descriptor?key=$apiKey"
            ),
        ) {
        map.uiSettings.isAttributionEnabled = true
        map.uiSettings.isLogoEnabled = false
        map.uiSettings.attributionGravity = Gravity.BOTTOM or Gravity.END
        val initialPosition = LatLng(47.6160281982247,
-122.32642111977668)
        map.cameraPosition = CameraPosition.Builder()
            .target(initialPosition)
            .zoom(14.0)
            .build()
    }
}
}
```

- b. Simpan file `MainActivity.kt`. Anda sekarang dapat membangun aplikasi. Untuk menjalankannya, Anda mungkin harus menyiapkan perangkat untuk meniru perangkat Android Studio, atau menggunakan aplikasi di perangkat Anda. Gunakan aplikasi ini untuk melihat bagaimana kontrol peta berperilaku. Anda dapat menggeser dengan menyeretnya di peta dan mencubitnya untuk memperbesar.

Di bagian selanjutnya, Anda akan menambahkan penanda pada peta dan menunjukkan alamat lokasi di mana penanda berada saat Anda memindahkan peta.

## Menambahkan pencarian geocoding terbalik ke aplikasi Anda

Anda sekarang akan menambahkan pencarian geocoding terbalik ke aplikasi Anda, di mana Anda menemukan item di lokasi. Untuk menyederhanakan menggunakan aplikasi Android, kami akan mencari di tengah layar. Untuk menemukan lokasi baru, pindahkan peta ke tempat yang ingin Anda cari. Kami akan menempatkan penanda di tengah peta untuk menunjukkan di mana kami mencari.

Menambahkan pencarian geocoding terbalik akan terdiri dari dua bagian.

- Tambahkan penanda di tengah layar untuk menunjukkan kepada pengguna di mana kita mencari.
- Tambahkan kotak teks untuk hasil, lalu cari apa yang ada di lokasi penanda dan tampilkan di kotak teks.

Untuk menambahkan penanda ke aplikasi Anda

1. Simpan gambar ini ke proyek Anda di `app/res/drawable` folder sebagai `red_marker.png` (Anda juga dapat mengakses gambar dari [GitHub](#). Atau, Anda dapat membuat gambar Anda. Anda juga dapat menggunakan `file.png` dengan transparansi untuk bagian yang tidak ingin ditampilkan.
2. Tambahkan kode berikut ke `MapLoadScreenfile.kt` Anda.

```
// ...other imports
import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.size
import androidx.compose.ui.Alignment
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.unit.dp
import com.amazon.testmapapp.R

@Composable
fun MapLoadScreen(
    mapReadyCallback: OnMapReadyCallback,
) {
    Box(
        modifier = Modifier
            .fillMaxWidth()
            .fillMaxHeight(),
    ) {
        MapView(mapReadyCallback)
    }
}
```

```

        Box(
            modifier = Modifier
                .align(Alignment.Center),
        ) {
            Image(
                painter = painterResource(id = R.drawable.red_marker),
                contentDescription = "marker",
                modifier = Modifier
                    .size(40.dp)
                    .align(Alignment.Center),
            )
        }
    }
}

@Composable
fun MapView(mapReadyCallback: OnMapReadyCallback) {
    AndroidView(
        factory = { context ->
            val mapView = org.maplibre.android.maps.MapView(context)
            mapView.onCreate(null)
            mapView.getMapAsync(mapReadyCallback)
            mapView
        },
    )
}

```

### 3. Buat dan jalankan aplikasi Anda untuk melihat pratinjau fungsionalitas.

Aplikasi Anda sekarang memiliki penanda di layar. Dalam hal ini, ini adalah gambar statis yang tidak bergerak. Ini digunakan untuk menunjukkan pusat tampilan peta, di mana kita akan mencari. Dalam prosedur berikut, kami akan menambahkan pencarian di lokasi itu.

Untuk menambahkan penelusuran geocoding terbalik di suatu lokasi ke aplikasi Anda

1. Di jendela Project, buka gradle untuk `libs.versions.toml` berkas dalam tampilan pohon. Ini akan membuka `libs.versions.toml` file untuk diedit. Sekarang tambahkan versi dan data pustaka di bawah ini dalam `libs.versions.toml` file.

```

[versions]
...
okhttp = "4.12.0"

```

```
[libraries]
...
com-squareup-okhttp3 = { group = "com.squareup.okhttp3", name = "okhttp",
version.ref = "okhttp" }

[plugins]
...
```

2. Setelah Anda selesai mengedit `libs.versions.toml` file, AndroidStudio harus menyinkronkan ulang proyek. Di bagian atas jendela `libs.versions.toml` pengeditan, AndroidStudio meminta Anda untuk menyinkronkan. Pilih 'Sinkronkan Sekarang' untuk menyinkronkan proyek Anda sebelum melanjutkan.
3. Di jendela Project, buka Gradle Scripts dalam tampilan pohon dan pilih `build.gradle` file untuk modul aplikasi Anda. Ini akan membuka `build.gradle` file untuk diedit.
4. Di bagian bawah file, di bagian dependensi, tambahkan dependensi berikut.

```
dependencies {
    ...
    implementation(libs.com.squareup.okhttp3)
}
```

5. Setelah Anda selesai mengedit dependensi Gradle, AndroidStudio harus menyinkronkan ulang proyek. Di bagian atas jendela `build.gradle` pengeditan, AndroidStudio meminta Anda untuk menyinkronkan. Pilih SyncNow untuk menyinkronkan proyek Anda sebelum melanjutkan.
6. Sekarang di tampilan pohon tambahkan data, ke direktori permintaan, dan buat kelas `ReverseGeocodeRequest.kt` data. Tambahkan kode berikut ke kelas.

```
import com.google.gson.annotations.SerializedName

data class ReverseGeocodeRequest(
    @SerializedName("Language")
    val language: String,
    @SerializedName("MaxResults")
    val maxResults: Int,
    @SerializedName("Position")
    val position: List<Double>
)
```

7. Sekarang di tampilan pohon tambahkan data ke direktori respons, dan buat kelas `ReverseGeocodeResponse.kt` data. Tambahkan kode berikut di dalamnya.

```
import com.google.gson.annotations.SerializedName

data class ReverseGeocodeResponse(
    @SerializedName("Results")
    val results: List<Result>
)

data class Result(
    @SerializedName("Place")
    val place: Place
)

data class Place(
    @SerializedName("Label")
    val label: String
)
```

8. Sekarang, Dari jendela Project, buka App, Java, *nama paket Anda* di tampilan pohon, dan pergi ke folder ui, di dalam folder ui buat direktori ViewModel.
9. Di dalam direktori ViewModel buat `MainViewModel.kt` file.
10. Tambahkan kode berikut ke `MainViewModel.kt` file Anda.

```
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.setValue
import androidx.lifecycle.ViewModel
import com.amazon.testmapapp.data.request.ReverseGeocodeRequest
import com.amazon.testmapapp.data.response.ReverseGeocodeResponse
import com.google.gson.Gson
import java.io.IOException
import okhttp3.Call
import okhttp3.Callback
import okhttp3.MediaType.Companion.toMediaTypeOrNull
import okhttp3.OkHttpClient
import okhttp3.Request
import okhttp3.RequestBody.Companion.toRequestBody
import okhttp3.Response
import org.maplibre.android.geometry.LatLng
import org.maplibre.android.maps.MapLibreMap
```

```
class MainViewModel : ViewModel() {
    var label by mutableStateOf("")
    var isLabelAdded: Boolean by mutableStateOf(false)
    var client = OkHttpClient()
    var mapLibreMap: MapLibreMap? = null

    fun reverseGeocode(latLng: LatLng, apiKey: String) {
        val region = "YOUR_AWS_REGION"
        val indexName = "YOUR_AWS_PLACE_INDEX"
        val url =
            "https://places.geo.{$region}.amazonaws.com/places/v0/indexes/
            {$indexName}/search/position?key={$apiKey}"

        val requestBody = ReverseGeocodeRequest(
            language = "en",
            maxResults = 1,
            position = listOf(latLng.longitude, latLng.latitude)
        )
        val json = Gson().toJson(requestBody)

        val mediaType = "application/json".toMediaTypeOrNull()
        val request = Request.Builder()
            .url(url)
            .post(json.toRequestBody(mediaType))
            .build()

        client.newCall(request).enqueue(object : Callback {
            override fun onFailure(call: Call, e: IOException) {
                e.printStackTrace()
            }

            override fun onResponse(call: Call, response: Response) {
                if (response.isSuccessful) {
                    val jsonResponse = response.body?.string()

                    val reverseGeocodeResponse =
                        Gson().fromJson(jsonResponse,
ReverseGeocodeResponse::class.java)

                    val responseLabel =
reverseGeocodeResponse.results.firstOrNull()?.place?.label

                    if (responseLabel != null) {
```



```

        label = responseLabel
        isLabelAdded = true
    }
}
})
}
}
}

```

11. Jika belum terbuka, buka `MapLoadScreen.kt` file, seperti pada prosedur sebelumnya. Menambahkan kode berikut. Ini akan membuat tampilan teks tulis di mana Anda akan melihat hasil pencarian geocoding terbalik di lokasi.

```

// ...other imports
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Spacer
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.height
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.Text
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.platform.testTag
import androidx.compose.ui.semantics.contentDescription
import androidx.compose.ui.semantics.semantics
import androidx.compose.ui.unit.sp
import com.amazon.testmapapp.ui.viewModel.MainViewModel

@Composable
fun MapLoadScreen(
    mapReadyCallback: OnMapReadyCallback,
    mainViewModel: MainViewModel,
) {
    Box(
        modifier = Modifier
            .fillMaxWidth()
            .fillMaxHeight(),
    ) {
        MapView(mapReadyCallback)
        Box(
            modifier = Modifier

```

```
        .align(Alignment.Center),
    ) {
        Image(
            painter = painterResource(id = R.drawable.red_marker),
            contentDescription = "marker",
            modifier = Modifier
                .size(40.dp)
                .align(Alignment.Center),
        )
    }
    if (mainViewModel.isLabelAdded) {
        Column(
            modifier = Modifier.fillMaxSize(),
            verticalArrangement = Arrangement.Bottom
        ) {
            Box(
                modifier = Modifier
                    .fillMaxWidth()
                    .background(Color.White),
            ) {
                Text(
                    text = mainViewModel.label,
                    modifier = Modifier
                        .padding(16.dp)
                        .align(Alignment.Center)
                        .testTag("label")
                        .semantics {
                            contentDescription = "label"
                        },
                    fontSize = 14.sp,
                )
            }
            Spacer(modifier = Modifier.height(80.dp))
        }
    }
}

@Composable
fun MapView(mapReadyCallback: OnMapReadyCallback) {
    AndroidView(
        factory = { context ->
            val mapView = org.maplibre.android.maps.MapView(context)
            mapView.onCreate(null)
        }
    )
}
```

```

        mapView.getMapAsync(mapReadyCallback)
        mapView
    },
)
}

```

12. Di aplikasi, di bawah java, di folder nama paket di AndroidStudio, buka MainActivity.kt file. Ubah kode seperti yang ditunjukkan.

```

// ...other imports
import androidx.activity.viewModels
import com.amazon.testmapapp.ui.viewModel.MainViewModel

class MainActivity : ComponentActivity(), OnMapReadyCallback,
MapLibreMap.OnCameraMoveStartedListener, MapLibreMap.OnCameraIdleListener {

    private val mainViewModel: MainViewModel by viewModels()
    private val region = "YOUR_AWS_REGION"
    private val mapName = "YOUR_AWS_MAP_NAME"
    private val apiKey = "YOUR_AWS_API_KEY"
    override fun onCreate(savedInstanceState: Bundle?) {
        MapLibre.getInstance(this)
        super.onCreate(savedInstanceState)
        setContent {
            TestMapAppTheme {
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    MapLoadScreen(this, mainViewModel)
                }
            }
        }
    }

    override fun onMapReady(map: MapLibreMap) {
        map.setStyle(
            Style.Builder()
                .fromUri(
                    "https://maps.geo.$region.amazonaws.com/maps/v0/maps/$mapName/
style-descriptor?key=$apiKey"
                ),
        ) {

```

```
map.uiSettings.isAttributionEnabled = true
map.uiSettings.isLogoEnabled = false
map.uiSettings.attributionGravity = Gravity.BOTTOM or Gravity.END
val initialPosition = LatLng(47.6160281982247, -122.32642111977668)
map.cameraPosition = CameraPosition.Builder()
    .target(initialPosition)
    .zoom(14.0)
    .build()

map.addOnCameraMoveStartedListener(this)
map.addOnCameraIdleListener(this)
map.cameraPosition.target?.let { latLng ->
    mainViewModel.reverseGeocode(
        LatLng(
            latLng.latitude,
            latLng.longitude
        ), apiKey
    )
}
}

override fun onCameraMoveStarted(p0: Int) {
    mainViewModel.label = ""
    mainViewModel.isLabelAdded = false
}

override fun onCameraIdle() {
    if (!mainViewModel.isLabelAdded) {
        mainViewModel.mapLibreMap?.cameraPosition?.target?.let { latLng ->
            mainViewModel.reverseGeocode(
                LatLng(
                    latLng.latitude,
                    latLng.longitude
                ), apiKey
            )
        }
    }
}
}
```

Kode ini bekerja dengan tampilan peta. Posisi kamera virtual mendefinisikan tampilan peta di MapLibre. Memindahkan peta dapat dianggap sebagai menggerakkan kamera virtual itu.

- `ViewModel` memiliki variabel `label`: Variabel ini menetapkan data dalam tampilan teks penulisan.
  - `onMapReady`: Fungsi ini diperbarui untuk mendaftarkan dua acara baru.
  - `onCameraMovePeristiwa` ini terjadi setiap kali pengguna memindahkan peta. Secara umum, saat memindahkan peta, kami ingin menyembunyikan pencarian sampai pengguna selesai memindahkan peta.
  - `onCameraIdlePeristiwa` terjadi ketika pengguna berhenti memindahkan peta. Acara ini memanggil fungsi `geocode` terbalik kami untuk mencari di tengah peta.
  - `reverseGeocode(latLng: LatLng, apiKey: String)`: Fungsi ini, dipanggil dalam acara `onCameraIdle`, mencari di tengah peta untuk lokasi dan memperbarui label untuk menunjukkan hasilnya. Ini menggunakan target kamera, yang mendefinisikan pusat peta (di mana kamera melihat).
13. Simpan file Anda, dan buat serta jalankan aplikasi Anda untuk melihat apakah itu berfungsi.

Aplikasi mulai cepat Anda dengan fungsionalitas pencarian selesai.

## Menambahkan pelacakan ke aplikasi Anda

Untuk menambahkan pelacakan ke aplikasi sampel Anda, ikuti langkah-langkah berikut:

1. Tambahkan dependensi SDK pelacakan dan autentikasi ke project Anda.
  2. Sertakan izin dan entri layanan dalam `AndroidManifest` file.xmlmu.
  3. Siapkan kode tombol pelacakan start/stop dengan `compose`.
  4. Tambahkan kode untuk membuat `LocationTracker` objek dan mulai dan berhenti melacak.
  5. Buat rute pengujian dengan Android Emulator.
1. Tambahkan dependensi SDK pelacakan dan autentikasi ke project Anda.
    - a. Di jendela Project, buka `gradle` lalu buka `libs.versions.toml` file di tampilan pohon. Ini akan membuka `libs.versions.toml` file untuk diedit. Sekarang tambahkan versi dan data pustaka di bawah ini dalam `libs.versions.toml` file.

```
[versions]
...
auth = "0.0.1"
```

```
tracking = "0.0.1"

[libraries]
...
auth = { group = "software.amazon.location", name = "auth", version.ref =
"auth" }
tracking = { module = "software.amazon.location:tracking", version.ref =
"tracking" }

[plugins]
...
```

- b. Setelah Anda selesai mengedit `libs.versions.toml` file, AndroidStudio harus menyinkronkan ulang proyek. Di bagian atas jendela `libs.versions.toml` pengeditan, AndroidStudio meminta Anda untuk menyinkronkan. Pilih 'Sinkronkan Sekarang' untuk menyinkronkan proyek Anda sebelum melanjutkan.
- c. Di jendela Project, buka Gradle Scripts dalam tampilan pohon dan pilih `build.gradle` file untuk modul aplikasi Anda. Ini akan membuka `build.gradle` file untuk diedit.
- d. Di bagian bawah file, di bagian dependensi, tambahkan dependensi berikut.

```
dependencies {
    ...
    implementation(libs.auth)
    implementation(libs.tracking)
}
```

- e. Setelah Anda selesai mengedit dependensi Gradle, AndroidStudio harus menyinkronkan ulang proyek. Di bagian atas jendela pengeditan `build.gradle`, AndroidStudio meminta Anda untuk menyinkronkan. Pilih SyncNow untuk menyinkronkan proyek Anda sebelum melanjutkan.
2. Sertakan izin dan entri layanan dalam `AndroidManifest.xml` file.
    - Untuk menyertakan izin dan entri layanan yang benar dalam `AndroidManifest.xml` file, perbarui file dengan kode berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

```
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.INTERNET"/>
<application
    android:allowBackup="true"
    android:dataExtractionRules="@xml/data_extraction_rules"
    android:fullBackupContent="@xml/backup_rules"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.AndroidQuickStartApp"
    tools:targetApi="31">
    <activity
        android:name=".MainActivity"
        android:exported="true"
        android:label="@string/app_name"
        android:theme="@style/Theme.AndroidQuickStartApp">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>

</manifest>
```

3. Siapkan kode tombol pelacakan start/stop dengan compose.
  - a. Tambahkan dua gambar Play dan Pause in res di bawah drawable bernama ic\_pause dan ic\_play. Anda juga dapat mengakses gambar dari [GitHub](#).
  - b. Jika belum terbuka, buka MapLoadScreen.kt file, seperti pada prosedur sebelumnya. Menambahkan kode berikut. Ini akan membuat tampilan Tombol tulis di mana kita dapat mengkliknya untuk memulai dan menghentikan pelacakan.

```
// ...other imports
import androidx.compose.material3.Button
import androidx.compose.material3.ButtonDefaults

@Composable
fun MapLoadScreen(
    mapReadyCallback: OnMapReadyCallback,
    mainViewModel: MainViewModel,
```

```
onStartStopTrackingClick: () -> Unit
) {
    Box(
        modifier = Modifier
            .fillMaxWidth()
            .fillMaxHeight(),
    ) {
        MapView(mapReadyCallback)
        Box(
            modifier = Modifier
                .align(Alignment.Center),
        ) {
            Image(
                painter = painterResource(id = R.drawable.red_marker),
                contentDescription = "marker",
                modifier = Modifier
                    .size(40.dp)
                    .align(Alignment.Center),
            )
        }
        if (mainViewModel.isLabelAdded) {
            Column(
                modifier = Modifier.fillMaxSize(),
                verticalArrangement = Arrangement.Bottom
            ) {
                Box(
                    modifier = Modifier
                        .fillMaxWidth()
                        .background(Color.White),
                ) {
                    Text(
                        text = mainViewModel.label,
                        modifier = Modifier
                            .padding(16.dp)
                            .align(Alignment.Center)
                            .testTag("label")
                            .semantics {
                                contentDescription = "label"
                            },
                        fontSize = 14.sp,
                    )
                }
                Spacer(modifier = Modifier.height(80.dp))
            }
        }
    }
}
```



```
    }
    Column(
        modifier = Modifier
            .fillMaxSize()
            .padding(bottom = 16.dp),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Bottom,
    ) {
        Button(
            onClick = onStartStopTrackingClick,
            modifier = Modifier
                .padding(horizontal = 16.dp)
        ) {
            Text(
                text = if
                    (mainViewModel.isLocationTrackingForegroundActive) "Stop tracking" else "Start
                    tracking",
                color = Color.Black
            )
            Spacer(modifier = Modifier.size(ButtonDefaults.IconSpacing))
            Image(
                painter = painterResource(id = if
                    (mainViewModel.isLocationTrackingForegroundActive) R.drawable.ic_pause else
                    R.drawable.ic_play),
                contentDescription = if
                    (mainViewModel.isLocationTrackingForegroundActive) "stop_tracking" else
                    "start_tracking"
            )
        }
    }
}

@Composable
fun MapView(mapReadyCallback: OnMapReadyCallback) {
    AndroidView(
        factory = { context ->
            val mapView = org.maplibre.android.maps.MapView(context)
            mapView.onCreate(null)
            mapView.getMapAsync(mapReadyCallback)
            mapView
        },
    )
}
```

```
}
```

4. Tambahkan kode untuk membuat LocationTracker objek dan mulai dan berhenti melacak.

a. Tambahkan kode berikut di dalam MainViewModel.kt file.

```
...  
var isLocationTrackingForegroundActive: Boolean by mutableStateOf(false)  
var locationTracker: LocationTracker? = null
```

b. Tambahkan kode berikut ke MainActivity.kt file Anda.

```
// ...other imports  
import software.amazon.location.auth.AuthHelper  
import software.amazon.location.auth.LocationCredentialsProvider  
import software.amazon.location.tracking.LocationTracker  
import software.amazon.location.tracking.aws.LocationTrackingCallback  
import software.amazon.location.tracking.config.LocationTrackerConfig  
import software.amazon.location.tracking.database.LocationEntry  
import software.amazon.location.tracking.filters.DistanceLocationFilter  
import software.amazon.location.tracking.filters.TimeLocationFilter  
import software.amazon.location.tracking.util.TrackingSdkLogLevel  
  
class MainActivity : ComponentActivity(), OnMapReadyCallback,  
    MapLibreMap.OnCameraMoveStartedListener, MapLibreMap.OnCameraIdleListener {  
  
    private val mainViewModel: MainViewModel by viewModels()  
    private val poolId = "YOUR_AWS_IDENTITY_POOL_ID"  
    private val trackerName = "YOUR_AWS_TRACKER_NAME"  
    private val region = "YOUR_AWS_REGION"  
    private val mapName = "YOUR_AWS_MAP_NAME"  
    private val apiKey = "YOUR_AWS_API_KEY"  
    private val coroutineScope = MainScope()  
    private lateinit var locationCredentialsProvider:  
LocationCredentialsProvider  
    private lateinit var authHelper: AuthHelper  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        MapLibre.getInstance(this)  
        super.onCreate(savedInstanceState)  
        setContent {  
            TestMapAppTheme {  
                Surface(  

```

```

        modifier = Modifier.fillMaxSize(),
        color = MaterialTheme.colorScheme.background
    ) {
        MapLoadScreen(this, mainViewModel, onStartStopTrackingClick
= {
            if (mainViewModel.isLocationTrackingForegroundActive) {
                mainViewModel.isLocationTrackingForegroundActive =
false
                mainViewModel.locationTracker?.stop()
            } else {
                if (checkLocationPermission(this))
return@MapLoadScreen
                    mainViewModel.isLocationTrackingForegroundActive =
true
mainViewModel.locationTracker?.start(locationTrackingCallback = object :
                LocationTrackingCallback {
                    override fun
onLocationAvailabilityChanged(locationAvailable: Boolean) {
                        }

                    override fun onLocationReceived(location:
LocationEntry) {
                        }

                    override fun onUploadSkipped(entries:
LocationEntry) {
                        }

                    override fun onUploadStarted(entries:
List<LocationEntry>) {
                        }

                    override fun onUploaded(entries:
List<LocationEntry>) {
                        }
                })
            }
        })
    }
}
authenticateUser()

```

```
    }

    private fun authenticateUser() {
        coroutineScope.launch {
            authHelper = AuthHelper(applicationContext)
            locationCredentialsProvider =
authHelper.authenticateWithCognitoIdentityPool(
                poolId,
            )
            locationCredentialsProvider.let {
                val config = LocationTrackerConfig(
                    trackerName = trackerName,
                    logLevel = TrackingSdkLogLevel.DEBUG,
                    latency = 1000,
                    frequency = 5000,
                    waitForAccurateLocation = false,
                    minUpdateIntervalMillis = 5000,
                )
                mainViewModel.locationTracker = LocationTracker(
                    applicationContext,
                    it,
                    config,
                )

                mainViewModel.locationTracker?.enableFilter(TimeLocationFilter())

                mainViewModel.locationTracker?.enableFilter(DistanceLocationFilter())
            }
        }
    }

    private fun checkLocationPermission(context: Context) =
        ActivityCompat.checkSelfPermission(
            context,
            Manifest.permission.ACCESS_FINE_LOCATION,
        ) != PackageManager.PERMISSION_GRANTED &&
        ActivityCompat.checkSelfPermission(
            context,
            Manifest.permission.ACCESS_COARSE_LOCATION,
        ) != PackageManager.PERMISSION_GRANTED

    override fun onMapReady(map: MapLibreMap) {
        map.setStyle(
            Style.Builder()
        )
    }
}
```

```
        .fromUri(
            "https://maps.geo.$region.amazonaws.com/maps/v0/maps/
$mapName/style-descriptor?key=$apiKey"
        ),
    ) {
        mainViewModel.mapLibreMap = map
        map.uiSettings.isAttributionEnabled = true
        map.uiSettings.isLogoEnabled = false
        map.uiSettings.attributionGravity = Gravity.BOTTOM or Gravity.END
        val initialPosition = LatLng(47.6160281982247, -122.32642111977668)
        map.cameraPosition = CameraPosition.Builder()
            .target(initialPosition)
            .zoom(14.0)
            .build()

        map.addOnCameraMoveStartedListener(this)
        map.addOnCameraIdleListener(this)
        map.cameraPosition.target?.let { latLng ->
            mainViewModel.reverseGeocode(
                LatLng(
                    latLng.latitude,
                    latLng.longitude
                ), apiKey
            )
        }
    }
}

override fun onCameraMoveStarted(p0: Int) {
    mainViewModel.label = ""
    mainViewModel.isLabelAdded = false
}

override fun onCameraIdle() {
    if (!mainViewModel.isLabelAdded) {
        mainViewModel.mapLibreMap?.cameraPosition?.target?.let { latLng ->
            mainViewModel.reverseGeocode(
                LatLng(
                    latLng.latitude,
                    latLng.longitude
                ), apiKey
            )
        }
    }
}
```

```
}  
}
```

Kode di atas menunjukkan cara membuat `LocationTracker` objek dengan `AuthHelper` dan bagaimana memulai dan berhenti melacak dengan `LocationTracker`.

- `authenticateUser()`: Metode ini menciptakan `AuthHelper` dan `LocationTracker` objek.
- `onStartStopTrackingClick`: Panggilan balik ini dipicu saat pengguna mengklik tombol start/stop tracking, yang akan memulai/menghentikan pelacakan dengan `Tracking SDK`.

## 5. Buat rute pengujian dengan Android Emulator.

- Buka Emulator dengan meluncurkan AVD menggunakan Android Studio.
- Buka `Extended Controls` dengan mengklik ikon Lainnya (tiga titik) di toolbar emulator.
- Buka Lokasi dengan memilih Lokasi dari bilah sisi.
- Buat rute dengan data GPX atau dengan mengklik peta dan memilih sumber dan data tujuan.
- Mulai Simulasi dengan mengklik `PLAY ROUTE` untuk mulai mensimulasikan rute GPS.
- Uji Aplikasi dengan menjalankan aplikasi Anda dan mengamati bagaimana menangani rute simulasi.

Ini adalah demo lengkap dari aplikasi Android Quick Start.

## Apa selanjutnya

Kode sumber untuk aplikasi ini tersedia di [GitHub](#).

Untuk mendapatkan lebih banyak dari Lokasi Amazon, Anda dapat melihat sumber daya berikut:

- Selami lebih dalam [konsep Amazon Location Service](#)
- Dapatkan informasi selengkapnya tentang [cara menggunakan fitur dan fungsionalitas Lokasi Amazon](#)
- Lihat cara memperluas sampel ini dan membangun aplikasi yang lebih kompleks dengan melihat [contoh kode menggunakan Lokasi Amazon](#)

# Membuat aplikasi iOS

Di bagian ini, Anda akan membuat aplikasi iOS dengan kemampuan untuk mencari di lokasi dan melacak di latar depan. Pertama, Anda akan membuat sumber daya Lokasi Amazon Anda, dan identitas Amazon Cognito untuk aplikasi Anda.

## Topik

- [Membuat sumber daya Lokasi Amazon untuk aplikasi Anda](#)
- [Menyiapkan otentikasi untuk aplikasi Anda](#)
- [Membuat aplikasi iOS dasar](#)
- [Menyiapkan kode awal](#)
- [Menambahkan peta interaktif ke aplikasi Anda](#)
- [Menambahkan pencarian ke aplikasi Anda](#)
- [Menambahkan pelacakan ke aplikasi Anda](#)
- [Apa selanjutnya](#)

## Membuat sumber daya Lokasi Amazon untuk aplikasi Anda

Jika Anda belum memilikinya, Anda harus membuat sumber daya Lokasi Amazon yang akan digunakan aplikasi Anda. Anda akan membuat sumber daya peta untuk menampilkan peta dalam aplikasi Anda, indeks tempat untuk mencari lokasi di peta, dan pelacak untuk melacak objek di seluruh peta.

Untuk menambahkan sumber daya lokasi ke aplikasi Anda

1. Pilih gaya peta yang ingin Anda gunakan.
  - a. Di konsol Lokasi Amazon, pada halaman [Peta](#), pilih Buat peta untuk melihat pratinjau gaya peta.
  - b. Tambahkan Nama dan Deskripsi untuk sumber daya peta baru. Catat nama yang Anda gunakan untuk sumber daya peta. Anda akan membutuhkannya saat membuat file skrip Anda nanti di tutorial.
  - c. Pilih peta.

**Note**

Memilih gaya peta juga memilih penyedia data peta mana yang akan Anda gunakan. Jika aplikasi Anda melacak atau merutekan aset yang Anda gunakan dalam bisnis Anda, seperti kendaraan pengiriman atau karyawan, Anda hanya dapat menggunakan HERE sebagai penyedia geolokasi Anda. Untuk informasi selengkapnya, lihat bagian 82 dari [ketentuan AWS layanan](#).

- d. Setuju dengan Syarat dan Ketentuan Lokasi Amazon, lalu pilih Buat peta. Anda dapat berinteraksi dengan peta yang Anda pilih: memperbesar, memperkecil, atau menggeser ke segala arah.
  - e. Catat Nama Sumber Daya Amazon (ARN) yang ditampilkan untuk sumber daya peta baru Anda. Anda akan menggunakannya untuk membuat otentikasi yang benar nanti dalam tutorial ini.
2. Pilih indeks tempat yang ingin Anda gunakan.
- a. Di konsol Lokasi Amazon di halaman [Indeks tempat](#), pilih Buat indeks tempat.
  - b. Tambahkan Nama dan Deskripsi untuk sumber daya indeks tempat baru. Catat nama yang Anda gunakan untuk sumber daya indeks tempat. Anda akan membutuhkannya saat membuat file skrip Anda nanti di tutorial.
  - c. Pilih penyedia data.

**Note**

Dalam kebanyakan kasus, pilih penyedia data yang cocok dengan penyedia peta yang sudah Anda pilih. Ini membantu memastikan bahwa pencarian akan cocok dengan peta.

Jika aplikasi Anda melacak atau merutekan aset yang Anda gunakan dalam bisnis Anda, seperti kendaraan pengiriman atau karyawan, Anda hanya dapat menggunakan HERE sebagai penyedia geolokasi Anda. Untuk informasi selengkapnya, lihat bagian 82 dari [ketentuan AWS layanan](#).

- d. Pilih opsi Penyimpanan data. Untuk tutorial ini, hasilnya tidak disimpan, sehingga Anda dapat memilih Tidak, sekali pakai saja.
- e. Setuju dengan Syarat dan Ketentuan Lokasi Amazon, lalu pilih Buat indeks tempat.



- f. Catat ARN yang ditampilkan untuk sumber daya indeks tempat baru Anda. Anda akan menggunakannya untuk membuat otentikasi yang benar di bagian selanjutnya dari tutorial ini.
3. Untuk membuat pelacak menggunakan konsol Lokasi Amazon.
    - a. Buka [konsol Amazon Location Service](#).
    - b. Di panel navigasi kiri, pilih Pelacak.
    - c. Pilih Buat pelacak.
    - d. Isi semua bidang yang diperlukan.
    - e. Di bawah Pemfilteran posisi, kami menyarankan Anda menggunakan pengaturan default: TimeBased
    - f. Pilih Buat pelacak untuk menyelesaikan.

## Menyiapkan otentikasi untuk aplikasi Anda

Aplikasi yang Anda buat dalam tutorial ini memiliki penggunaan anonim, yang berarti bahwa pengguna Anda tidak diharuskan masuk AWS untuk menggunakan aplikasi. Namun, Amazon Location Service API memerlukan otentikasi untuk digunakan. Anda akan menggunakan Amazon Cognito untuk memberikan otentikasi dan otorisasi bagi pengguna anonim. Tutorial ini akan menggunakan Amazon Cognito untuk mengautentikasi aplikasi Anda.

### Note

Untuk informasi selengkapnya tentang penggunaan Amazon Cognito dengan Amazon Location Service, lihat [Memberikan akses ke Amazon Location Service](#)

Tutorial berikut menunjukkan cara mengatur otentikasi untuk peta, indeks tempat, dan pelacak yang Anda buat serta menyiapkan izin untuk Lokasi Amazon.

### Membuat kebijakan IAM untuk melacak

1. Masuk ke konsol IAM di <https://console.aws.amazon.com/iam/> dengan pengguna Anda yang memiliki izin administrator.
2. Di panel navigasi, pilih Kebijakan.
3. Di panel konten, pilih Buat kebijakan.

- Pilih opsi JSON, lalu salin dan tempel kebijakan JSON ini ke dalam kotak teks JSON.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "geo:GetMapTile",
        "geo:GetMapStyleDescriptor",
        "geo:GetMapSprites",
        "geo:GetMapGlyphs",
        "geo:SearchPlaceIndexForPosition",
        "geo:GetDevicePositionHistory",
        "geo:BatchUpdateDevicePosition"
      ],
      "Resource": [
        "arn:aws:geo:{Region}:{Account}:map/{MapName}",
        "arn:aws:geo:{Region}:{Account}:place-index/{IndexName}",
        "arn:aws:geo:{Region}:{Account}:tracker/{TrackerName}"
      ]
    }
  ]
}
```

Ini adalah contoh kebijakan untuk Tracking. Untuk menggunakan contoh untuk kebijakan Anda sendiri, ganti,Region, Account, IndexName, MapName dan TrackerNameplaceholder.

#### Note

Meskipun kumpulan identitas yang tidak diautentikasi dimaksudkan untuk eksposur di situs internet yang tidak aman, perhatikan bahwa mereka akan ditukar dengan kredensi standar dan terbatas waktu. AWS


Penting untuk mencakup peran IAM yang terkait dengan kumpulan identitas yang tidak diautentikasi dengan tepat. Untuk informasi selengkapnya tentang penggunaan dan kebijakan pelingkupan yang tepat di Amazon Cognito dengan Amazon Location Service, lihat Memberikan akses ke [Amazon](#) Location Service.

- Pada halaman Tinjau dan Buat, berikan nama untuk bidang nama kebijakan. Tinjau izin yang diberikan oleh kebijakan Anda, lalu pilih Buat Kebijakan untuk menyimpan pekerjaan Anda.

Kebijakan baru muncul di daftar kebijakan terkelola dan siap dilampirkan.

Siapkan otentikasi untuk pelacakan Anda

1. Siapkan otentikasi untuk aplikasi peta Anda di konsol [Amazon Cognito](#).
2. Buka halaman Identity pool.

 Note

Kumpulan yang Anda buat harus berada di AWS akun dan AWS Wilayah yang sama dengan sumber daya Amazon Location Service yang Anda buat di bagian sebelumnya.

3. Pilih Buat kumpulan Identitas.
4. Dimulai dengan langkah Configure identity pool trust. Untuk otentikasi akses pengguna, pilih Akses tamu, dan tekan berikutnya.
5. Pada halaman Konfigurasi izin pilih Gunakan peran IAM yang ada dan masukkan nama peran IAM yang Anda buat di langkah sebelumnya. Saat siap tekan di sebelah untuk melanjutkan ke langkah berikutnya.
6. Pada halaman Konfigurasi properti, berikan nama untuk kumpulan identitas Anda. Lalu tekan Next.
7. Pada halaman Tinjau dan buat, tinjau semua informasi yang ada lalu tekan Buat kumpulan identitas.
8. Buka halaman Identity pool, dan pilih kumpulan identitas yang baru saja Anda buat. Kemudian salin atau tuliskan IdentityPoolId yang akan Anda gunakan nanti di skrip browser Anda.

## Membuat aplikasi iOS dasar

Dalam tutorial ini, Anda akan membuat aplikasi iOS yang menyematkan peta, dan memungkinkan pengguna untuk menemukan apa yang ada di lokasi di peta.

Pertama, mari buat aplikasi Swift menggunakan wizard proyek Xcode.

Untuk membuat aplikasi kosong (Xcode)

1. Buka Xcode, dan dari menu, pilih File, New, New Project.
2. Dari tab iOS, pilih Aplikasi, lalu pilih Berikutnya.

3. Berikan Nama Produk, Pengidentifikasi Organisasi, dan di input SwiftUI bidang Antarmuka. Pilih Berikutnya untuk menyelesaikan seleksi.
4. Pilih lokasi di mana Anda akan menyimpan proyek Anda dan tekan tombol buat untuk membuat aplikasi kosong.

Setelah Anda membuat aplikasi dasar, Anda harus menginstal paket yang diperlukan untuk aplikasi sampel.

Menginstal dependensi yang diperlukan

1. Di Xcode, klik kanan pada proyek dan pilih Add Packages... . Ini akan membuka jendela Paket, di mana Anda dapat menambahkan paket ke proyek Anda.
2. Di jendela Paket, tambahkan paket berikut:
  - Untuk paket asli Maplibre, gunakan URL ini: <https://github.com/maplibre/.maplibre-gl-native-distribution> Dari URL, tambahkan paket-paket ini:maplibre-gl-native-distribution, danMapbox.
  - Untuk SDK iOS autentikasi Lokasi Amazon, gunakan URL ini: <https://github.com/aws-geospatial/amazon-location-mobile-auth> -sdk-ios. Dari URL, tambahkan paket-paket ini:amazon-location-mobile-auth-sdk-ios, danAmazonLocationiOSAuthSDK.
  - Untuk SDK iOS yang melacak Lokasi Amazon, gunakan URL ini: <https://github.com/aws-geospatial/amazon-location-mobile-tracking-sdk-ios>. Dari URL, tambahkan paket-paket ini:amazon-location-mobile-tracking-sdk-ios, danAmazonLocationIOSTrackingSDK.

## Menyiapkan kode awal

Mengaktifkan izin Lokasi di aplikasi Anda

1. Buka proyek Xcode Anda.
2. Temukan Info.plist file proyek.
3. Tambahkan kunci yang diperlukan untuk izin lokasi berdasarkan persyaratan aplikasi Anda. Berikut kuncinya:
  - `NSLocationWhenInUseUsageDescription`: Deskripsi mengapa aplikasi Anda memerlukan akses lokasi saat sedang digunakan.

- `CLLocationAlwaysAndWhenInUseUsageDescription`: Deskripsi mengapa aplikasi Anda membutuhkan akses lokasi berkelanjutan.

Sekarang Anda perlu mengonfigurasi nilai sumber daya di aplikasi Anda. Tambahkan file baru bernama `Config.xcconfig` dan isi nilai yang telah Anda buat sebelumnya di konsol Amazon.

```
REGION =  
INDEX_NAME =  
MAP_NAME =  
IDENTITY_POOL_ID =  
TRACKER_NAME =
```

1. Dari bagian navigator sisi kiri, pilih proyek.
2. Di bawah bagian target, pilih aplikasi Anda dan klik tab info.
3. Tambahkan properti info dengan nilai seperti di bawah ini:
4. Tambahkan `Config.swift` file dengan konten di bawah ini, yang akan membaca nilai konfigurasi dari file info Bundle.

```
import Foundation  
  
enum Config {  
    static let region = Bundle.main.object(forKey: "Region") as!  
    String  
    static let mapName = Bundle.main.object(forKey: "MapName") as!  
    String  
    static let indexName = Bundle.main.object(forKey: "IndexName")  
as! String  
    static let identityPoolId = Bundle.main.object(forKey:  
"IdentityPoolId") as! String  
    static let trackerName = Bundle.main.object(forKey:  
"TrackerName") as! String  
}
```

5. Buat folder baru dengan nama `ViewModel` dan tambahkan `TrackingViewModel.swift` file di dalamnya.

```
import SwiftUI  
import AmazonLocationiOSAuthSDK  
import MapLibre
```

```
final class TrackingViewModel : ObservableObject {
    @Published var trackingButtonText = NSLocalizedString("StartTrackingLabel",
comment: "")
    @Published var trackingButtonColor = Color.blue
    @Published var trackingButtonIcon = "play.circle"
    @Published var region : String
    @Published var mapName : String
    @Published var indexName : String
    @Published var identityPoolId : String
    @Published var trackerName : String
    @Published var showAlert = false
    @Published var alertTitle = ""
    @Published var alertMessage = ""
    @Published var centerLabel = ""

    var clientIntialised: Bool
    var client: LocationTracker!
    var authHelper: AuthHelper
    var credentialsProvider: LocationCredentialsProvider?
    var mlnMapView: MLNMapView?
    var mapViewDelegate: MapViewDelegate?
    var lastGetTrackingTime: Date?
    var trackingActive: Bool

    init(region: String, mapName: String, indexName: String, identityPoolId:
String, trackerName: String) {
        self.region = region
        self.mapName = mapName
        self.indexName = indexName
        self.identityPoolId = identityPoolId
        self.trackerName = trackerName
        self.authHelper = AuthHelper()
        self.trackingActive = false
        self.clientIntialised = false
    }

    func authWithCognito(identityPoolId: String?) {
        guard let identityPoolId =
identityPoolId?.trimmingCharacters(in: .whitespacesAndNewlines)
        else {
            alertTitle = NSLocalizedString("Error", comment: "")
            alertMessage = NSLocalizedString("NotAllFieldsAreConfigured", comment:
"")
        }
    }
}
```

```

        showAlert = true
        return
    }
    credentialsProvider =
    authHelper.authenticateWithCognitoUserPool(identityPoolId: identityPoolId)
    initializeClient()
}

func initializeClient() {
    client = LocationTracker(provider: credentialsProvider!, trackerName:
trackerName)
    clientIntialised = true
}
}

```

## Menambahkan peta interaktif ke aplikasi Anda

Anda sekarang akan menambahkan kontrol peta ke aplikasi Anda. Tutorial ini menggunakan MapLibre dan AWS API untuk mengelola tampilan peta dalam aplikasi. Kontrol peta itu sendiri adalah bagian dari pustaka [iOS Asli MapLibre GL](#).

1. Tambahkan `MapView.swift` file di bawah folder Views dengan kode berikut:

```

import SwiftUI
import MapLibre

struct MapView: UIViewRepresentable {
    var onMapViewAvailable: ((MLNMapView) -> Void)?
    var mlnMapView: MLNMapView?
    var trackingViewModel: TrackingViewModel

    func makeCoordinator() -> MapView.Coordinator {
        return Coordinator(self, trackingViewModel: trackingViewModel)
    }

    func makeUIView(context: Context) -> MLNMapView {
        let styleURL = URL(string: "https://maps.geo.
\\(trackingViewModel.region).amazonaws.com/maps/v0/maps/
\\(trackingViewModel.mapName)/style-descriptor")
        let mapView = MLNMapView(frame: .zero, styleURL: styleURL)
        mapView.autoresizingMask = [.flexibleWidth, .flexibleHeight]
        mapView.setZoomLevel(15, animated: true)
    }
}

```

```
mapView.showsUserLocation = true
mapView.userTrackingMode = .follow
context.coordinator.mlnMapView = mapView
mapView.delegate = context.coordinator

mapView.logoView.isHidden = true
context.coordinator.addCenterMarker()

onMapViewAvailable?(mapView)
trackingViewModel.mlnMapView = mapView
return mapView
}

func updateUIView(_ uiView: MLNMapView, context: Context) {
}

class Coordinator: NSObject, MLNMapViewDelegate, MapViewDelegate {
    var control: MapView
    var mlnMapView: MLNMapView?
    var trackingViewModel: TrackingViewModel
    var centerMarker: MLNPointAnnotation?

    public init(_ control: MapView, trackingViewModel: TrackingViewModel) {
        self.control = control
        self.trackingViewModel = trackingViewModel
        super.init()
        self.trackingViewModel.mapViewDelegate = self
    }

    func mapViewDidFinishRenderingMap(_ mapView: MLNMapView, fullyRendered:
Bool) {
        if(fullyRendered) {
            mapView.accessibilityIdentifier = "MapView"
            mapView.isAccessibilityElement = false
        }
    }

    func addCenterMarker() {
        guard let mlnMapView = mlnMapView else {
            return
        }

        let centerCoordinate = mlnMapView.centerCoordinate
        let marker = MLNPointAnnotation()
```



```

        marker.coordinate = centerCoordinate
        marker.accessibilityLabel = "CenterMarker"
        mlnMapView.addAnnotation(marker)
        centerMarker = marker

        trackingViewModel.reverseGeocodeCenter(centerCoordinate:
centerCoordinate, marker: marker)
    }

    func mapView(_ mapView: MLNMapView, regionDidChangeAnimated animated: Bool)
    {
        if let marker = centerMarker {
            DispatchQueue.main.asyncAfter(deadline: .now() + 1.0)
            {
                mapView.deselectAnnotation(marker, animated: false)
                marker.coordinate = mapView.centerCoordinate
                let centerCoordinate = mapView.centerCoordinate
                self.trackingViewModel.reverseGeocodeCenter(centerCoordinate:
centerCoordinate, marker: marker)
            }
        }
    }
}

```

2. Tambahkan `AWSSignatureV4Delegate` file di bawah `ViewModel` folder. File ini digunakan untuk menandatangani dengan semua permintaan `MapView` http untuk membuat peta:

```

import MapLibre
import AmazonLocationiOSAuthSDK

class AWSSignatureV4Delegate : NSObject, MLNOfflineStorageDelegate {
    private let awsSigner: AWSSigner

    init(credentialsProvider: LocationCredentialsProvider) {
        self.awsSigner = DENY LIST ERROR , serviceName: "geo")
        super.init()
    }

    func offlineStorage(_ storage: MLNOfflineStorage, urlForResourceOf kind:
MLNResourceKind, with url: URL) -> URL {
        if url.host?.contains("amazonaws.com") != true {
            return url
        }
    }
}

```

```
    }
    let signedURL = awsSigner.signURL(url: url, expires: .hours(1))

    return signedURL
  }
}
```

3. Tambahkan `UserLocationView.swift` file di bawah folder `Views`. Ini menambahkan tombol yang memusatkan peta ke lokasi pengguna

```
import SwiftUI

struct UserLocationView: View {
    @ObservedObject var trackingViewModel: TrackingViewModel
    var body: some View {
        Button(action: {
            trackingViewModel.locateMe()
        }) {
            Image(systemName: "scope")
                .resizable()
                .frame(width: 24, height: 24)
                .padding(5)
                .background(Color.white)
                .foregroundColor(.blue)
                .clipShape(RoundedRectangle(cornerRadius: 8))
                .shadow(color: Color.black.opacity(0.3), radius: 3, x: 0, y: 2)
        }
        .accessibility(identifier: "LocateMeButton")
        .padding(.trailing, 10)
        .padding(.bottom, 10)
        .frame(maxWidth: .infinity, alignment: .trailing)
    }
}
```

4. Tambahkan `TrackingView.swift` file dengan kode berikut:

```
import SwiftUI

struct TrackingView: View {
    @ObservedObject var trackingViewModel: TrackingViewModel
    var body: some View {
        ZStack(alignment: .bottom) {
            MapView(trackingViewModel: trackingViewModel)
        }
    }
}
```

```
        VStack {
            UserLocationView(trackingViewModel: trackingViewModel)
        }
    }
    .onAppear() {
        if !trackingViewModel.identityPoolId.isEmpty {
            trackingViewModel.authWithCognito(identityPoolId:
trackingViewModel.identityPoolId)
        }
    }
}
}
```

Anda sekarang dapat membangun aplikasi. Untuk menjalankannya, Anda mungkin harus menyiapkan perangkat untuk meniru di Xcode atau menggunakan aplikasi di perangkat Anda. Gunakan aplikasi ini untuk melihat bagaimana kontrol peta berperilaku. Anda dapat menggeser dengan menyeret peta dan mencubit untuk memperbesar. Sendiri, Anda dapat mengubah cara kerja kontrol peta untuk menyesuainya dengan kebutuhan aplikasi Anda.

## Menambahkan pencarian ke aplikasi Anda

Anda sekarang akan menambahkan pencarian geocoding terbalik ke aplikasi, di mana Anda menemukan item di lokasi. Untuk menyederhanakan penggunaan aplikasi iOS, kami akan mencari di tengah layar. Untuk menemukan lokasi baru, pindahkan peta ke tempat yang ingin Anda cari. Kami akan menempatkan penanda di tengah peta untuk menunjukkan di mana kami mencari.

1. Tambahkan kode berikut dalam file `TrackingViewModel.swift` yang terkait dengan pencarian geocoding terbalik

```
func reverseGeocodeCenter(centerCoordinate: CLLocationCoordinate2D, marker:
MLNPointAnnotation) {
    let position = [NSNumber(value: centerCoordinate.longitude), NSNumber(value:
centerCoordinate.latitude)]
    searchPositionAPI(position: position, marker: marker)
}

func searchPositionAPI(position: [Double], marker: MLNPointAnnotation) {
    if let amazonClient = authHelper.getLocationClient() {
        Task {
            let searchRequest = SearchPlaceIndexForPositionInput(indexName:
indexName, language: "en" , maxResults: 10, position: position)
```

```

        let searchResponse = try? await amazonClient.searchPosition(indexName:
indexName, input: searchRequest)
        DispatchQueue.main.async {
            self.centerLabel = searchResponse?.results?.first?.place?.label ??
"""
            self.mlnMapView?.selectAnnotation(marker, animated: true,
completionHandler: {})
        }
    }
}
}

```

2. Perbarui `TrackingView.swift` file dengan kode berikut yang akan menampilkan alamat lokasi terpusat mapview

```

import SwiftUI

struct TrackingView: View {
    @ObservedObject var trackingViewModel: TrackingViewModel
    var body: some View {
        ZStack(alignment: .bottom) {
            if trackingViewModel.mapSigningIntialised {
                MapView(trackingViewModel: trackingViewModel)
                VStack {
                    UserLocationView(trackingViewModel: trackingViewModel)
                    CenterAddressView(trackingViewModel: trackingViewModel)
                }
            }
            else {
                Text("Loading...")
            }
        }
        .onAppear() {
            if !trackingViewModel.identityPoolId.isEmpty {
                Task {
                    do {
                        try await trackingViewModel.authWithCognito(identityPoolId:
trackingViewModel.identityPoolId)
                    }
                    catch {
                        print(error)
                    }
                }
            }
        }
    }
}

```

```
    }  
  }  
}
```

## Menambahkan pelacakan ke aplikasi Anda

Langkah terakhir untuk aplikasi Anda adalah menambahkan fungsionalitas pelacakan ke aplikasi Anda. Dalam hal ini, Anda akan menambahkan titik pelacak mulai, berhenti melacak, mengambil, dan menampilkan titik pelacak di aplikasi Anda.

1. Tambahkan `TrackingBottomView.swift` file dalam proyek Anda. Yang memiliki tombol yang memulai dan berhenti melacak lokasi pengguna dan menunjukkan titik pelacakan di peta.

```
import SwiftUI  
  
struct TrackingBottomView: View {  
    @ObservedObject var trackingViewModel: TrackingViewModel  
    var body: some View {  
        Button(action: {  
            Task {  
                if(trackingViewModel.trackingButtonText ==  
NSLocalizedString("StartTrackingLabel", comment: "")) {  
                    trackingViewModel.startTracking()  
                } else {  
                    trackingViewModel.stopTracking()  
                }  
            }  
        }) {  
            HStack {  
                Spacer()  
                Text("Tracking")  
                    .foregroundColor(trackingViewModel.trackingButtonColor)  
                    .background(.white)  
                    .cornerRadius(15.0)  
  
                Image(systemName: trackingViewModel.trackingButtonIcon)  
                    .resizable()  
                    .frame(width: 24, height: 24)  
                    .padding(5)  
                    .background(.white)  
            }  
        }  
    }  
}
```

```

        .foregroundColor(trackingViewModel.trackingButtonColor)
    }
}
.accessibility(identifier: "TrackingButton")
.background(.white)
.clipShape(RoundedRectangle(cornerRadius: 8))
.padding(.trailing, 10)
.padding(.bottom, 40)
.frame(width: 130, alignment: .trailing)
.shadow(color: Color.black.opacity(0.3), radius: 3, x: 0, y: 2)
}
}

```

## 2. Perbarui TrackingView.swift file dengan kode berikut

```

import SwiftUI

struct TrackingView: View {
    @ObservedObject var trackingViewModel: TrackingViewModel
    var body: some View {
        ZStack(alignment: .bottom) {
            if trackingViewModel.mapSigningInitialised {
                MapView(trackingViewModel: trackingViewModel)
                VStack {
                    UserLocationView(trackingViewModel: trackingViewModel)
                    CenterAddressView(trackingViewModel: trackingViewModel)
                    TrackingBottomView(trackingViewModel: trackingViewModel)
                }
            }
            else {
                Text("Loading...")
            }
        }
        .onAppear() {
            if !trackingViewModel.identityPoolId.isEmpty {
                Task {
                    do {
                        try await trackingViewModel.authWithCognito(identityPoolId:
trackingViewModel.identityPoolId)
                    }
                    catch {
                        print(error)
                    }
                }
            }
        }
    }
}

```

```

    }
  }
}
}
}

```

3. Tambahkan kode berikut dalam `TrackingViewModel.swift` file. Fungsi-fungsi ini bertanggung jawab untuk memulai dan menghentikan pelacakan. Ini juga akan menampilkan peringatan kesalahan jika izin lokasi pengguna ditolak.
4. Untuk menerapkan pelacakan latar depan salin tempel contoh kode berikut:

```

func showLocationDeniedRationale() {
    alertTitle = NSLocalizedString("locationManagerAlertTitle", comment: "")
    alertMessage = NSLocalizedString("locationManagerAlertText", comment: "")
    showAlert = true
}

// Required in info.plist: Privacy - Location When In Use Usage Description
func startTracking() {
    do {
        print("Tracking Started...")
        if(client == nil) {
            initializeClient()
        }
        try client.startTracking()
        DispatchQueue.main.async { [self] in
            self.trackingButtonText = NSLocalizedString("StopTrackingLabel",
comment: "")
            self.trackingButtonColor = .red
            self.trackingButtonIcon = "pause.circle"
            trackingActive = true
        }
    } catch TrackingLocationError.permissionDenied {
        showLocationDeniedRationale()
    } catch {
        print("error in tracking")
    }
}

func stopTracking() {
    print("Tracking Stopped...")
    client.stopTracking()
    trackingButtonText = NSLocalizedString("StartTrackingLabel", comment: "")
}

```

```
trackingButtonColor = .blue
trackingButtonIcon = "play.circle"
trackingActive = false
}
```

### Note

Mereka `startTracking` akan meminta izin lokasi pengguna. Aplikasi harus menggunakan izin `When In Use` atau `Only Once`. Jika tidak, aplikasi akan membuang kesalahan izin ditolak.

Untuk mendapatkan dan menampilkan lokasi pelacakan, ikuti prosedur ini:

1. Untuk mendapatkan lokasi dari perangkat pengguna, Anda perlu memberikan tanggal dan waktu mulai dan berakhir. Satu panggilan mengembalikan maksimum 100 lokasi pelacakan, tetapi jika ada lebih dari 100 lokasi pelacakan, itu akan mengembalikan nilai `nextToken`. Anda perlu memanggil panggilan `getTrackerDeviceLocation` berikutnya dengan `nextToken` untuk memuat lebih banyak titik pelacakan untuk waktu mulai dan berakhir yang diberikan.

```
func getTrackingPoints(nextToken: String? = nil) async throws {
    guard trackingActive else {
        return
    }
    // Initialize startTime to 24 hours ago from the current date and time.
    let startTime: Date = Date().addingTimeInterval(-86400)
    var endTime: Date = Date()
    if lastGetTrackingTime != nil {
        endTime = lastGetTrackingTime!
    }
    let result = try await client?.getTrackerDeviceLocation(nextToken:
nextToken, startTime: startTime, endTime: endTime)
    if let trackingData = result {

        lastGetTrackingTime = Date()
        let devicePositions = trackingData.devicePositions

        let positions = devicePositions!.sorted { (pos1:
LocationClientTypes.DevicePosition, pos2: LocationClientTypes.DevicePosition) ->
Bool in
            guard let date1 = pos1.sampleTime,
```



```

        let date2 = pos2.sampleTime else {
            return false
        }
        return date1 < date2
    }

    let trackingPoints = positions.compactMap { position ->
CLLocationCoordinate2D? in
        guard let latitude = position.position!.last, let longitude =
position.position!.first else {
            return nil
        }
        return CLLocationCoordinate2D(latitude: latitude, longitude:
longitude)
    }
    DispatchQueue.main.async {
        self.mapViewDelegate!.drawTrackingPoints( trackingPoints:
trackingPoints)
    }
    if let nextToken = trackingData.nextToken {
        try await getTrackingPoints(nextToken: nextToken)
    }
}
}
}

```

2. Sekarang ganti kode dalam `MapView.swift` file dengan kode berikut:

```

import SwiftUI
import MapLibre

struct MapView: UIViewRepresentable {
    var onMapViewAvailable: ((MLNMapView) -> Void)?
    var mlnMapView: MLNMapView?
    var trackingViewModel: TrackingViewModel

    func makeCoordinator() -> MapView.Coordinator {
        return Coordinator(self, trackingViewModel: trackingViewModel)
    }

    func makeUIView(context: Context) -> MLNMapView {
        let styleURL = URL(string: "https://maps.geo.
\\(trackingViewModel.region).amazonaws.com/maps/v0/maps/
\\(trackingViewModel.mapName)/style-descriptor")
        let mapView = MLNMapView(frame: .zero, styleURL: styleURL)
    }
}

```

```
mapView.autoresizingMask = [.flexibleWidth, .flexibleHeight]
mapView.setZoomLevel(15, animated: true)
mapView.showsUserLocation = true
mapView.userTrackingMode = .follow
context.coordinator.mlnMapView = mapView
mapView.delegate = context.coordinator

mapView.logoView.isHidden = true
context.coordinator.addCenterMarker()

onMapViewAvailable?(mapView)
trackingViewModel.mlnMapView = mapView
return mapView
}

func updateUIView(_ uiView: MLNMapView, context: Context) {
}

class Coordinator: NSObject, MLNMapViewDelegate, MapViewDelegate {
    var control: MapView
    var mlnMapView: MLNMapView?
    var trackingViewModel: TrackingViewModel
    var centerMarker: MLNPointAnnotation?

    public init(_ control: MapView, trackingViewModel: TrackingViewModel) {
        self.control = control
        self.trackingViewModel = trackingViewModel
        super.init()
        self.trackingViewModel.mapViewDelegate = self
    }

    func mapViewDidFinishRenderingMap(_ mapView: MLNMapView, fullyRendered:
Bool) {
        if(fullyRendered) {
            mapView.accessibilityIdentifier = "MapView"
            mapView.isAccessibilityElement = false
        }
    }

    func addCenterMarker() {
        guard let mlnMapView = mlnMapView else {
            return
        }
    }
}
```

```
        let centerCoordinate = mlnMapView.centerCoordinate
        let marker = MLNPointAnnotation()
        marker.coordinate = centerCoordinate
        marker.accessibilityLabel = "CenterMarker"
        mlnMapView.addAnnotation(marker)
        centerMarker = marker

        trackingViewModel.reverseGeocodeCenter(centerCoordinate:
centerCoordinate, marker: marker)
    }

    func mapView(_ mapView: MLNMapView, regionDidChangeAnimated animated: Bool)
    {
        if let marker = centerMarker {
            DispatchQueue.main.asyncAfter(deadline: .now() + 1.0) {
                mapView.deselectAnnotation(marker, animated: false)
                marker.coordinate = mapView.centerCoordinate
                let centerCoordinate = mapView.centerCoordinate
                self.trackingViewModel.reverseGeocodeCenter(centerCoordinate:
centerCoordinate, marker: marker)
            }
        }
    }

    func mapView(_ mapView: MLNMapView, viewFor annotation: MLNAnnotation) ->
MLNAnnotationView? {
        guard let pointAnnotation = annotation as? MLNPointAnnotation else {
            return nil
        }

        let reuseIdentifier: String
        var color: UIColor = .black
        if pointAnnotation.accessibilityLabel == "Tracking" {
            reuseIdentifier = "TrackingAnnotation"
            color = UIColor(red: 0.00784313725, green: 0.50588235294, blue:
0.58039215686, alpha: 1)
        } else if pointAnnotation.accessibilityLabel == "LocationChange" {
            reuseIdentifier = "LocationChange"
            color = .gray
        } else {
            reuseIdentifier = "DefaultAnnotationView"
        }
    }
```

```

        var annotationView =
mapView.dequeueReusableAnnotationView(withIdentifier: reuseIdentifier)

        if annotationView == nil {
            if reuseIdentifier != "DefaultAnnotationView" {
                annotationView = MLNAnnotationView(annotation: annotation,
reuseIdentifier: reuseIdentifier)
                //If point annotation is an uploaded Tracking point the radius
is 20 and color is blue, otherwise radius is 10 and color is gray
                let radius = pointAnnotation.accessibilityLabel == "Tracking" ?
20:10

                annotationView?.frame = CGRect(x: 0, y: 0, width: radius,
height: radius)

                annotationView?.backgroundColor = color
                annotationView?.layer.cornerRadius = 10

                if pointAnnotation.accessibilityLabel == "Tracking" {
                    annotationView?.layer.borderColor = UIColor.white.cgColor
                    annotationView?.layer.borderWidth = 2.0
                    annotationView?.layer.shadowColor = UIColor.black.cgColor
                    annotationView?.layer.shadowOffset = CGSize(width: 0,
height: 2)

                    annotationView?.layer.shadowRadius = 3
                    annotationView?.layer.shadowOpacity = 0.2
                    annotationView?.clipsToBounds = false
                }
            }
            else {
                return nil
            }
        }

        return annotationView
    }

    func mapView(_ mapView: MLNMapView, didUpdate userLocation:
MLNUserLocation?) {
        if (userLocation?.location) != nil {
            if trackingViewModel.trackingActive {
                let point = MLNPointAnnotation()
                point.coordinate = (userLocation?.location!.coordinate)!
                point.accessibilityLabel = "LocationChange"
                mapView.addAnnotation(point)
                Task {

```

```
        do {
            try await trackingViewModel.getTrackingPoints()
        }
        catch {
            print(error)
        }
    }
}

func checkIfTrackingAnnotationExists(on mapView: MLNMapView, at
coordinates: CLLocationCoordinate2D) -> Bool {
    let existingAnnotation = mapView.annotations?.first(where: { annotation
in
        guard let annotation = annotation as? MLNPointAnnotation else
{ return false }
        return annotation.coordinate.latitude == coordinates.latitude &&
        annotation.coordinate.longitude == coordinates.longitude &&
        annotation.accessibilityLabel == "Tracking" })
    return existingAnnotation != nil
}

public func drawTrackingPoints(trackingPoints: [CLLocationCoordinate2D]?) {
    guard let mapView = mlnMapView, let newTrackingPoints =
trackingPoints, !newTrackingPoints.isEmpty else {
        return
    }

    let uniqueCoordinates = newTrackingPoints.filter { coordinate in
        !checkIfTrackingAnnotationExists(on: mapView, at: coordinate)
    }

    let points = uniqueCoordinates.map { coordinate -> MLNPointAnnotation
in
        let point = MLNPointAnnotation()
        point.coordinate = coordinate
        point.accessibilityLabel = "Tracking"
        return point
    }
    mapView.addAnnotations(points)
}
}
```

```
protocol MapViewDelegate: AnyObject {
    func drawTrackingPoints(trackingPoints: [CLLocationCoordinate2D]?)
}
```

Untuk melokalisasi nilai string, gunakan prosedur berikut.

1. Buat dan tambahkan file baru bernama `Localizable.xcstrings`.
2. Klik kanan pada `Localizable.xcstrings` file dan buka sebagai Source Code.
3. Ganti kontennya dengan yang berikut:

```
{
  "sourceLanguage" : "en",
  "strings" : {
    "Cancel" : {
      "extractionState" : "manual",
      "localizations" : {
        "en" : {
          "stringUnit" : {
            "state" : "translated",
            "value" : "Cancel"
          }
        }
      }
    },
    "Error" : {
      "extractionState" : "manual",
      "localizations" : {
        "en" : {
          "stringUnit" : {
            "state" : "translated",
            "value" : "Error"
          }
        }
      }
    },
    "Loading..." : {
  },
  "locationManagerAlertText" : {
    "extractionState" : "manual",
```

```
"localizations" : {
  "en" : {
    "stringUnit" : {
      "state" : "translated",
      "value" : "Allow \\\\"Quick Start App\\" to use your location"
    }
  }
},
"locationManagerAlertTitle" : {
  "extractionState" : "manual",
  "localizations" : {
    "en" : {
      "stringUnit" : {
        "state" : "translated",
        "value" : "We need your location to detect your location in map"
      }
    }
  }
},
"NotAllFieldsAreConfigured" : {
  "extractionState" : "manual",
  "localizations" : {
    "en" : {
      "stringUnit" : {
        "state" : "translated",
        "value" : "Not all the fields are configured"
      }
    }
  }
},
"OK" : {
  "extractionState" : "manual",
  "localizations" : {
    "en" : {
      "stringUnit" : {
        "state" : "translated",
        "value" : "OK"
      }
    }
  }
},
"StartTrackingLabel" : {
  "localizations" : {
```

```
        "en" : {
            "stringUnit" : {
                "state" : "translated",
                "value" : "Start Tracking"
            }
        }
    },
    "StopTrackingLabel" : {
        "localizations" : {
            "en" : {
                "stringUnit" : {
                    "state" : "translated",
                    "value" : "Stop Tracking"
                }
            }
        }
    },
    "Tracking" : {
    }
},
"version" : "1.0"
}
```

4. Simpan file Anda, dan buat serta jalankan aplikasi Anda untuk melihat pratinjau fungsionalitas.
5. Izinkan izin lokasi dan ketuk tombol pelacakan. Aplikasi akan mulai mengunggah lokasi pengguna dan mengunggahnya ke pelacak Lokasi Amazon. Ini juga akan menampilkan perubahan lokasi pengguna, titik pelacakan, dan alamat saat ini di peta.

Aplikasi mulai cepat Anda selesai. Tutorial ini telah menunjukkan kepada Anda cara membuat aplikasi iOS yang:

- Membuat peta yang dapat berinteraksi dengan pengguna.
- Menangani beberapa peristiwa peta yang terkait dengan pengguna mengubah tampilan peta.
- Memanggil Amazon Location Service API, khususnya untuk mencari peta di lokasi, menggunakan Amazon Location's `searchByPosition` API.



## Apa selanjutnya

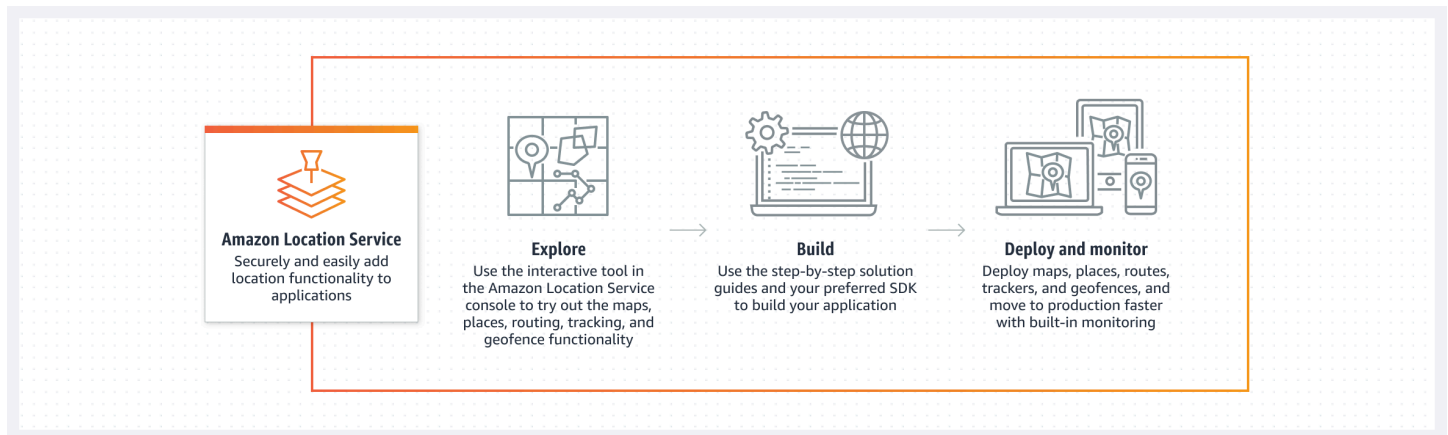
Kode sumber untuk aplikasi ini tersedia di [GitHub](#).

Untuk mendapatkan lebih banyak dari Lokasi Amazon, Anda dapat melihat sumber daya berikut:

- Selami lebih dalam [konsep Amazon Location Service](#)
- Dapatkan informasi selengkapnya tentang [cara menggunakan fitur dan fungsionalitas Lokasi Amazon](#)
- Lihat cara memperluas sampel ini dan membangun aplikasi yang lebih kompleks dengan melihat [contoh kode menggunakan Lokasi Amazon](#)

# Konsep Amazon Location Service

Dengan Amazon Location Service, Anda dapat menambahkan data lokasi ke aplikasi dengan aman. Jelajahi beberapa kemampuan dengan menggunakan [alat visual dan interaktif](#), yang tersedia di konsol Lokasi Amazon. Dengan menggunakan alat jelajahi, Anda dapat memanipulasi peta default, mencari tempat menarik, menggambar geofences di sekitar area yang diminati, dan mensimulasikan pengiriman lokasi perangkat ke pelacak.



Saat Anda siap membangun, buat sumber daya Anda dan pilih dari berbagai gaya peta dan penyedia data. Kemudian Anda dapat menginstal SDK yang cocok dengan lingkungan pengembangan Anda, dan menggunakan Amazon Location API menggunakan petunjuk dalam panduan ini. Selain itu, Anda dapat mengintegrasikan pemantauan dengan menggunakan Amazon CloudWatch dan AWS CloudTrail.

Topik di bagian ini memberi Anda gambaran umum tentang konsep inti Lokasi Amazon dan mempersiapkan Anda untuk mulai bekerja dengan lokasi di aplikasi Anda sendiri.

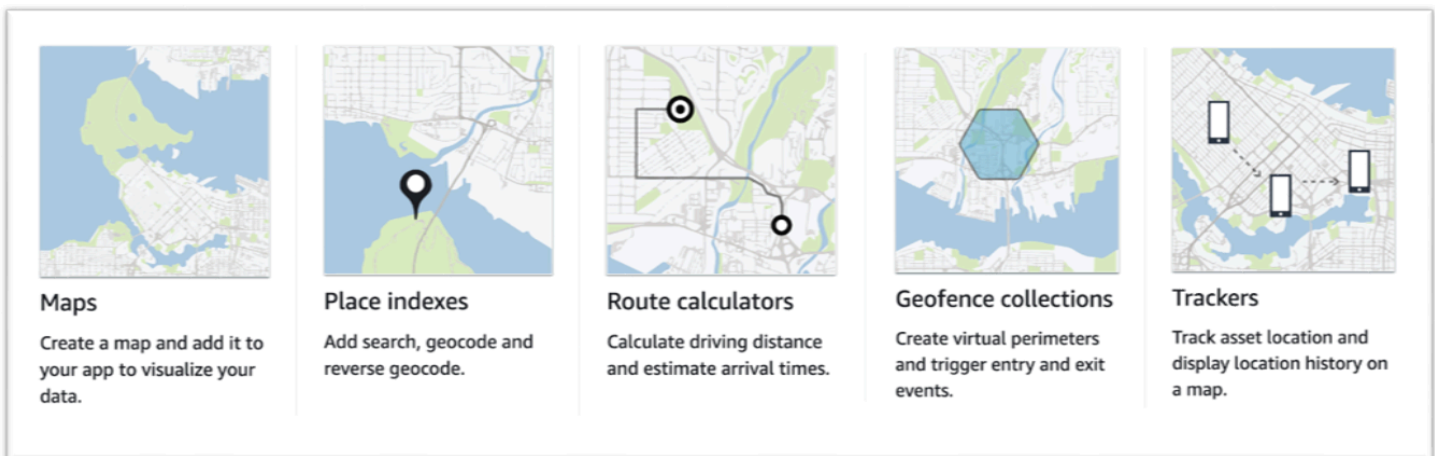
## Topik

- [Ikhtisar Lokasi Amazon](#)
- [Peta](#)
- [Pencarian tempat](#)
- [Rute](#)
- [Geofences dan Pelacak](#)
- [Kasus penggunaan umum untuk menggunakan Amazon Location Service](#)
- [Apa itu penyedia data?](#)

- [Wilayah Lokasi Amazon dan titik akhir](#)
- [Kuota Amazon Location Service](#)

## Ikhtisar Lokasi Amazon

Amazon Location Service menyediakan akses ke fungsionalitas berbasis lokasi dan penyedia data melalui AWS sumber daya. Lokasi Amazon menawarkan lima jenis AWS sumber daya, tergantung pada jenis fungsionalitas yang Anda butuhkan. Gunakan sumber daya yang berbeda bersama-sama untuk membuat aplikasi berbasis lokasi lengkap. Anda dapat membuat satu atau beberapa sumber daya ini menggunakan konsol Lokasi Amazon, API Lokasi Amazon, atau SDK.



Setiap sumber daya mendefinisikan [penyedia data](#) yang mendasari untuk digunakan (jika berlaku), dan memberikan akses ke fungsionalitas yang terkait dengan jenisnya.

Sebagai contoh:

- [Amazon Location Service Maps](#) memungkinkan Anda memilih peta dari penyedia peta untuk digunakan di aplikasi seluler atau web Anda.
- [Amazon Location Service Places](#) memungkinkan Anda memilih penyedia data untuk mencari tempat menarik, menyelesaikan sebagian teks, geocoding, dan reverse geocoding.
- [Amazon Location Service Routes](#) memungkinkan Anda memilih penyedia data dan menemukan rute serta memperkirakan waktu perjalanan berdasarkan informasi up-to-date jalan raya dan lalu lintas langsung.
- [Amazon Location Service Geofences](#) memungkinkan Anda mendefinisikan area minat sebagai batas virtual. Anda kemudian dapat mengevaluasi lokasi terhadap mereka dan mendapatkan pemberitahuan acara masuk dan keluar.

- [Amazon Location Service Trackers](#) menerima pembaruan lokasi dari perangkat Anda. Anda dapat menautkan pelacak ke koleksi geofence sehingga semua pembaruan posisi dievaluasi secara otomatis terhadap geofences Anda.

Anda dapat menggunakan kebijakan IAM untuk mengelola dan mengotorisasi akses ke sumber daya Lokasi Amazon Anda. Anda juga dapat mengatur sumber daya Anda ke dalam kelompok sumber daya untuk mengelola dan mengotomatiskan tugas seiring bertambahnya jumlah sumber daya Anda. Untuk informasi selengkapnya tentang mengelola AWS sumber daya, lihat [Apa itu AWS Resource Groups?](#) Dalam Panduan Pengguna AWS Resource Groups.

Lokasi didefinisikan dengan menggunakan koordinat lintang dan bujur yang mengikuti Sistem [Geodetik Dunia \(WGS 84\)](#), yang biasa digunakan sebagai sistem referensi koordinat standar untuk layanan Global Positioning System (GPS).

Bagian berikut menjelaskan cara kerja komponen Lokasi Amazon.

## Peta

Sumber daya Amazon Location Service Map memberi Anda akses ke data peta dasar yang mendasarinya. Anda menggunakan sumber daya Peta dengan pustaka rendering peta untuk menambahkan peta interaktif ke aplikasi Anda. Anda dapat menambahkan fungsionalitas lain ke peta Anda, seperti penanda (atau pin), rute, dan area poligon, sesuai kebutuhan untuk aplikasi Anda.

### Note

Untuk informasi tentang cara menggunakan sumber daya peta dalam praktik, lihat [Menggunakan Peta Lokasi Amazon di aplikasi Anda](#).

Berikut ini adalah ikhtisar tentang cara membuat dan menggunakan sumber daya peta:

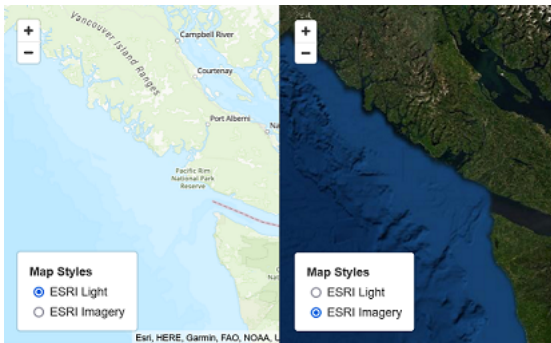


1. Anda membuat sumber daya peta di AWS akun Anda dengan memilih gaya peta dari penyedia data.

2. Anda kemudian dapat memilih dan menginstal SDK yang cocok dengan lingkungan pengembangan dan aplikasi Anda. Untuk informasi selengkapnya tentang opsi yang tersedia, lihat topik tentang [Mengakses Lokasi Amazon](#).
3. Untuk menampilkan peta di aplikasi Anda, gabungkan sumber daya peta dengan pustaka rendering, seperti Amplify MapLibre, atau Tangram. Untuk informasi selengkapnya, lihat [Menggunakan peta dalam panduan](#) ini.
4. Anda kemudian dapat mengintegrasikan pemantauan dengan menggunakan layanan, seperti Amazon CloudWatch dan AWS CloudTrail dengan Lokasi Amazon. Untuk informasi lebih lanjut lihat, [Memantau Amazon Location Service dengan Amazon CloudWatch](#) dan [Pencatatan dan pemantauan dengan AWS CloudTrail](#).

## Gaya peta

Saat Anda membuat sumber daya peta, Anda harus memilih gaya peta untuk sumber daya itu. Gaya peta menentukan tampilan peta yang dirender. Misalnya, gambar berikut menunjukkan penyedia data yang sama dengan dua gaya berbeda dari sumber peta yang berbeda di Lokasi Amazon. Salah satu gaya adalah gaya jalan yang khas, berdasarkan data vektor di peta. Yang lainnya termasuk data raster yang menunjukkan citra satelit. Gaya dapat berubah saat Anda memperbesar atau memperkecil peta, tetapi biasanya gaya memiliki tema yang konsisten. Dimungkinkan untuk mengganti bagian atau semua informasi gaya sebelum meneruskannya ke pustaka rendering peta.



## Pandangan politik

Gaya peta tertentu di Amazon Location Service mendukung pandangan politik tambahan.

**Note**

Pandangan politik harus digunakan sesuai dengan hukum yang berlaku, termasuk undang-undang tentang pemetaan negara atau wilayah tempat peta, gambar, dan data lainnya serta konten pihak ketiga yang Anda akses melalui Amazon Location Service tersedia.

Gaya peta berikut mendukung pandangan politik India (IND).

- [Gaya peta Esri](#):
  - Navigasi Esri
  - Cahaya Esri
  - Peta Jalan Esri
  - Esri Kanvas Abu-abu Gelap
  - Kanvas Abu-abu Muda Esri
- [Buka gaya peta Data](#):
  - Lampu Standar Data Terbuka
  - Buka Data Standar Gelap
  - Buka Lampu Visualisasi Data
  - Buka Visualisasi Data Gelap

Di konsol Amazon Location Service, Anda dapat memfilter gaya yang ditampilkan hanya untuk menampilkan gaya yang mendukung tampilan politik India.

## Lapisan Kustom

Lapisan kustom adalah lapisan tambahan yang dapat Anda aktifkan untuk gaya peta. Saat ini hanya gaya VectorEsriNavigation peta yang mendukung lapisan POI kustom.

Saat Anda mengaktifkan lapisan POI kustom, ia menambahkan serangkaian tempat yang lebih kaya, seperti toko, layanan, restoran, atraksi, dan tempat menarik lainnya ke peta Anda. Secara default, lapisan kustom adalah unset. Untuk informasi selengkapnya lihat, [MapConfiguration](#) di referensi API Lokasi.

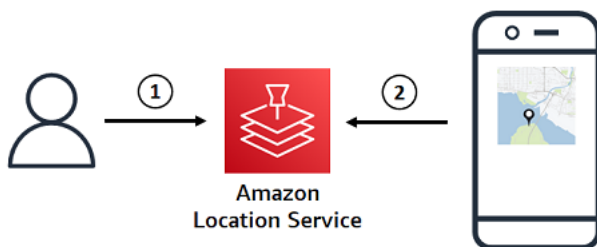
## Rendering peta

Untuk merender peta dalam aplikasi Anda, Anda biasanya akan menggunakan pustaka rendering peta. Ada beberapa opsi umum untuk perpustakaan untuk digunakan:

- **MapLibre**— MapLibre adalah pustaka open source khusus untuk merender peta interaktif, dan merupakan metode yang disukai untuk merender peta dari Amazon Location Service. MapLibre mencakup kemampuan untuk merender data raster dan vektor dari sumber data (seperti sumber daya peta Lokasi Amazon). Anda dapat memperluas MapLibre untuk menggambar data Anda sendiri di peta.
- **Amplify** — Amplify adalah kerangka kerja open source untuk membangun aplikasi untuk web, iOS, Android, dan lainnya. Jika aplikasi Anda menggunakan Amplify, Anda dapat memperluasnya untuk menyertakan fungsionalitas Lokasi Amazon. Amplify menyertakan library khusus untuk membuat aplikasi berbasis Amazon Location, termasuk merender peta. Amplify digunakan MapLibre untuk merender peta, tetapi menyediakan fungsionalitas tambahan yang khusus untuk Amazon Location Service agar lebih efisien digunakan, dan juga menambahkan penelusuran dan fungsionalitas lainnya.
- **Tangram** — Tangram adalah perpustakaan sumber terbuka alternatif yang membuat peta interaktif, mirip dengan. MapLibre

Pustaka rendering peta menarik data dari Amazon Location Service saat runtime, merender data peta berdasarkan sumber daya peta yang Anda pilih. Sumber daya peta mendefinisikan penyedia data dan gaya peta yang akan digunakan.

Gambar berikut menunjukkan bagaimana sumber daya peta digunakan di Amazon Location Service bersama dengan pustaka rendering peta untuk membuat peta akhir.



1. Anda membuat sumber daya peta di Amazon Location Service, menggunakan AWS Management Console atau AWS CLI. Ini mendefinisikan penyedia data dan gaya peta yang ingin Anda gunakan.

2. Aplikasi Anda menyertakan pustaka rendering peta. Anda memberi pustaka rendering peta nama sumber daya peta yang akan digunakan. Pustaka rendering peta menarik data dan informasi gaya untuk sumber daya peta tersebut dari Lokasi Amazon dan merender peta di layar.

## Terminologi peta

### Sumber daya peta

Memungkinkan Anda mengakses data peta dari penyedia yang dipilih. Gunakan sumber daya peta untuk mengambil ubin peta yang berisi data peta dan deskriptor gaya untuk menentukan bagaimana fitur ditampilkan pada peta.

### Peta dasar

Menyediakan konteks geografis ke peta Anda, yang disimpan sebagai lapisan ubin vektor. Lapisan ubin mencakup konteks geografis seperti nama jalan, bangunan, dan penggunaan lahan untuk referensi visual.

### vektor

Data vektor adalah data bentuk yang terdiri dari titik, garis, dan poligon. Ini sering digunakan untuk menyimpan dan menampilkan jalan, lokasi, dan area di peta. Bentuk vektor juga dapat digunakan sebagai ikon untuk penanda pada peta.

### Raster

Data raster adalah data gambar, terdiri dari kisi, biasanya dari warna. Ini sering digunakan untuk menyimpan dan menampilkan representasi data berkelanjutan pada peta, seperti medan, citra satelit, atau peta panas. Gambar raster juga dapat digunakan sebagai gambar atau ikon.

### Gaya Peta

Data vektor tidak secara inheren mencakup informasi tentang cara menggambar lapisan data untuk membuat peta akhir. Gaya peta mendefinisikan warna dan informasi gaya lainnya untuk data untuk menentukan bagaimana tampilannya saat dirender. Sumber daya peta mencakup informasi gaya untuk peta.

Amazon Location Service menyediakan gaya mengikuti spesifikasi gaya [Mapbox GL](#).



## Ubin vektor

Format ubin yang menyimpan data peta menggunakan bentuk vektor. Data ini menghasilkan peta yang dapat menyesuaikan dengan resolusi tampilan, dan secara selektif merender fitur dalam beberapa cara, sambil mempertahankan ukuran file yang kecil untuk kinerja optimal.

Format file vektor yang didukung: Mapbox Vector Tiles (MVT).

## File mesin terbang

File biner yang berisi karakter Unicode yang dikodekan. Digunakan oleh perender peta untuk menampilkan label.

## Berkas Sprite

File gambar Portable Network Graphic (PNG) yang berisi gambar raster kecil, dengan deskripsi lokasi dalam file JSON. Digunakan oleh perender peta untuk membuat ikon atau tekstur pada peta.

# Pencarian tempat

Fungsi utama Amazon Location Service adalah kemampuan untuk mencari informasi geolokasi. Amazon Location menyediakan fungsionalitas ini melalui sumber daya indeks Place.

### Note

Untuk informasi tentang cara menggunakan sumber daya indeks tempat untuk mencari dalam praktik, lihat [Mencari data tempat dan geolokasi menggunakan Amazon Location](#).

Anda dapat menggunakan API indeks tempat untuk mencari:

- Tempat menarik, seperti restoran dan landmark. Cari berdasarkan nama, dan lokasi opsional untuk mencari di sekitar, dan menerima daftar opsi yang diurutkan berdasarkan relevansi.
- Alamat jalan, menerima garis lintang dan bujur untuk alamat itu. Ini dikenal sebagai geocoding.
- Posisi lintang dan bujur, menerima alamat jalan terkait atau informasi lain tentang lokasi. Ini dikenal sebagai reverse geocoding.

- Kueri teks bentuk bebas sebagian atau salah eja, biasanya sebagai tipe pengguna. Ini dikenal sebagai pelengkapan otomatis, saran otomatis, atau pencocokan fuzzy.

Indeks tempat mencakup penyedia data mana yang akan digunakan untuk pencarian.

### Note

Data peta dan informasi geolokasi lainnya, termasuk lokasi yang tepat, dapat bervariasi antar penyedia data. Sebagai praktik terbaik, gunakan penyedia data yang sama untuk indeks tempat, peta, dan sumber daya Lokasi Amazon lainnya. Misalnya, jika tempat yang dikembalikan oleh indeks tempat Anda tidak cocok dengan lokasi tempat yang sama yang disediakan oleh sumber daya peta Anda, Anda dapat menempatkan penanda di lokasi yang tampaknya salah di peta.

Berikut ini menunjukkan cara membuat dan menggunakan sumber daya indeks tempat:



1. Pertama, Anda membuat sumber daya indeks tempat di AWS akun Anda dengan memilih penyedia data.
2. Anda kemudian dapat memilih dan menginstal SDK yang cocok dengan lingkungan pengembangan dan aplikasi Anda. Untuk informasi selengkapnya tentang opsi yang tersedia, lihat topik tentang [Mengakses Lokasi Amazon](#).
3. Mulai gunakan Amazon Location Places API. Untuk informasi selengkapnya, lihat topik tentang menggunakan [pencarian Tempat](#).
4. Anda kemudian dapat mengintegrasikan pemantauan menggunakan layanan seperti Amazon CloudWatch dan AWS CloudTrail. Untuk informasi lebih lanjut lihat, [the section called “Pemantauan CloudWatch dengan”](#) dan [the section called “Menggunakan CloudTrail dengan Lokasi Amazon”](#).

## Konsep geocoding

Indeks tempat Lokasi Amazon menyediakan tindakan yang disebut

[SearchPlaceIndexForText](#) yang memungkinkan Anda menentukan teks untuk dicari. Misalnya, Anda dapat mencari:

- Tempat — pencarian **Paris** bisa mengembalikan lokasi kota di Prancis.
- Bisnis — pencarian **coffee shop** dapat mengembalikan daftar kedai kopi, termasuk nama dan lokasi mereka. Anda juga dapat menentukan lokasi untuk mencari di sekitar atau kotak pembatas untuk mencari di dalam, untuk membuat hasil lebih relevan. Dalam hal ini, menyediakan lokasi di pusat kota Seattle, Washington, akan mengembalikan kedai kopi di daerah itu.
- Alamat — pencarian **1600 Pennsylvania Ave, Washington D.C.** dapat mengembalikan lokasi Gedung Putih di Amerika Serikat (yang ada di alamat itu).

Mencari teks dengan cara ini umumnya disebut sebagai geocoding, yang melibatkan menemukan lokasi geografis untuk alamat atau tempat.

Amazon Location Service juga menyediakan tindakan geocoding terbalik yang disebut [SearchPlaceIndexForPosition](#). Ini mengambil lokasi geografis dan mengembalikan alamat, bisnis, atau informasi lain tentang apa yang ada di lokasi itu.

## Hasil pencarian

Saat Anda membuat permintaan penelusuran yang berhasil di Amazon Location Service, satu atau beberapa hasil akan dikembalikan. Setiap hasil menyertakan label, yang merupakan nama atau deskripsi hasilnya. Misalnya, pencarian **coffee shop**, mungkin mengembalikan hasil dengan label `Hometown Cafe`, memberi tahu Anda bahwa sebuah kedai kopi bernama “Hometown Cafe” ditemukan. Hasil pencarian juga biasanya akan menyertakan alamat terstruktur (dengan properti seperti nomor alamat, unit, jalan, dan kode pos). Tergantung pada penyedia data, itu akan mencakup data meta lainnya, juga, seperti negara dan zona waktu.

Untuk pencarian pada nama atau kategori bisnis (seperti **coffee shop**), Anda mungkin ingin menampilkan semua hasil yang dikembalikan pada peta. Untuk pencarian alamat, Anda mungkin ingin menggunakan hasil pertama secara otomatis. Lihat topik berikutnya untuk informasi tentang relevansi.

## Berbagai hasil dan relevansi

Saat mencari berdasarkan teks, Amazon Location Service akan sering menemukan lebih dari satu hasil. Misalnya, pencarian **Paris** dapat mengembalikan kota di Prancis, tetapi juga kota di Texas. Hasilnya diurutkan berdasarkan relevansi, sebagaimana ditentukan oleh penyedia data.

### Note

Hasil dikembalikan dalam urutan relevansi dari semua penyedia. Jika Anda memilih Esri atau Grab sebagai penyedia data Anda, hasilnya mencakup nilai relevansi yang dapat Anda gunakan untuk memahami relevansi relatif antara hasil permintaan tunggal.

Menentukan informasi tambahan, seperti nama negara, atau lokasi untuk dicari, dapat mengubah urutan hasil, mengurangi jumlah hasil, atau bahkan mengubah kumpulan hasil yang dikembalikan. Misalnya, pencarian **Paris** dengan lokasi di Texas untuk mencari di sekitar lebih mungkin untuk kembali **Paris, Texas** sebagai hasil pertama daripada **Paris, France**.

Dalam aplikasi interaktif, Anda dapat menggunakan relevansi untuk membantu memutuskan apakah akan menerima hasil teratas, atau meminta pengguna untuk membedakan antara beberapa hasil yang dikembalikan. Jika hasil pertama memiliki relevansi yang tinggi, Anda mungkin menerimanya sebagai jawaban yang benar. Jika ada beberapa hasil relevansi tinggi, atau tidak ada hasil relevansi tinggi, Anda mungkin ingin membuat daftar hasil dan membiarkan pengguna memilih hasil terbaik.

## Alamat hasil

Anda dapat mencari alamat dengan Amazon Location Service menggunakan [SearchPlaceIndexForText](#) tindakan yang sama. Semakin banyak informasi yang Anda berikan, semakin besar kemungkinan alamat yang dikembalikan akan cocok dengan yang diberikan. Misalnya, kecil **123 Main St** kemungkinannya untuk menemukan hasil yang benar daripada **123 Main St, Anytown, California, 90210**.

Alamat memiliki beberapa atribut, seperti nomor jalan, jalan, kota, wilayah, dan kode pos, dll. Atribut tersebut digunakan untuk menemukan alamat di indeks tempat yang cocok dengan aspek sebanyak mungkin. Semakin banyak atribut yang ditemukan, semakin relevan kecocokan dipertimbangkan, dan semakin besar kemungkinannya akan dikembalikan.

**Note**

Relevansi untuk hasil alamat didasarkan pada seberapa dekat hasilnya cocok dengan input. Ini bisa berupa jumlah atribut yang cocok, tetapi juga seberapa dekat hasilnya cocok dengan input. Misalnya, input **123 Main St** akan memiliki relevansi yang lebih tinggi ketika **Main St** ditemukan dalam data, daripada jika **Maine St** adalah satu-satunya hasil. **Maine St** masih akan dikembalikan, tetapi kemungkinan dengan nilai relevansi yang lebih rendah.

Hasil pencarian menyertakan label untuk alamat lengkap (123 Main St, Anytown, California, 90210), tetapi juga atribut terstruktur individual dari alamat yang dikembalikan. Ini sangat membantu, karena Anda dapat menggunakannya, misalnya, untuk mengisi bidang alamat dalam database, atau untuk memeriksa hasilnya dan menemukan kota, wilayah, atau kode pos dari lokasi yang ditemukan.

**Interpolasi**

Alamat dalam data indeks tempat mencakup kecocokan alamat yang tepat. Misalnya, anggaplah ada jalan, 9th street dan satu blok memiliki 2 rumah, 220 dan 240, seperti pada gambar berikut.



Penyedia data membuat data geolokasi dengan dua alamat yang diketahui tersebut. Anda dapat mencari dua alamat tersebut, dan mereka ditemukan. Setelah penyedia data membuat data peta, misalkan rumah baru ditambahkan, di antara dua alamat pertama. Rumah baru ini diberi

alamatnya 230. Jika Anda mencari **230 S 9th St**, penyedia data masih akan menemukan hasilnya. Alih-alih menggunakan alamat yang diketahui, itu akan menginterpolasi antara alamat yang sudah diketahui, dan memperkirakan posisi alamat baru dari alamat tersebut. Dalam hal ini, mungkin diasumsikan bahwa 230 berada di tengah-tengah antara 220 dan 240 (dan di sisi jalan yang sama), dan mengembalikan perkiraan lokasi berdasarkan itu.

#### Note

Penyedia data secara berkala memperbarui data geolokasi mereka dengan alamat baru. Dalam hal ini, **230 S 9th St** akan ditambahkan ke data penyedia data, tetapi biasanya akan ada periode ketika alamat baru telah dibuat tetapi belum ditambahkan ke data.

Dalam hal ini, penyedia data tidak dapat mengetahui apakah alamat baru itu ada di dunia, karena belum ada dalam data, tetapi memberikan jawaban terbaik dari informasi yang dimilikinya. Hasil ini disebut interpolasi, dan dapat dikembalikan oleh penyedia data dalam hasil. Jika `interpolated` kembali `false`, itu adalah alamat yang diketahui. Jika kembali `true`, itu adalah alamat perkiraan. Jika tidak dikembalikan, maka penyedia data tidak memberikan informasi tentang apakah hasilnya berasal dari interpolasi.

#### Important

Penyedia data juga dapat mengembalikan hasil interpolasi untuk alamat yang tidak ada sama sekali. Misalnya dalam kasus ini, jika Anda masuk **232 S 9th St**, penyedia akan menemukan alamat yang tidak ada ini, dan mengembalikan lokasi yang mendekati 230, tetapi di sisi 240. Alamat yang diinterpolasi berguna untuk membawa Anda ke lokasi yang tepat, tetapi perlu diingat bahwa alamat tersebut bukan alamat yang diketahui.

## Menyimpan hasil geocode

Saat Anda membuat sumber daya indeks tempat, Anda harus menentukan opsi penyimpanan data (disebut `IntendedUse` dalam API). Itu dapat diatur menjadi penggunaan tunggal atau hasil yang disimpan. Ini menanyakan tentang tujuan penggunaan hasil Anda. Jika Anda akan menyimpan hasilnya (bahkan untuk tujuan caching), Anda harus memilih opsi penyimpanan, bukan opsi penggunaan tunggal.

**Note**

Saat Anda memilih opsi tersimpan (diberi label Ya, hasil akan disimpan di konsol, atau memilih `storage` di `CreatePlaceIndex` API), Amazon Location Service tidak menyimpan hasilnya untuk Anda. Ini adalah indikasi bahwa Anda berencana untuk menyimpan hasilnya.

Saat melihat bagaimana Anda akan menggunakan hasil kueri Anda ke Amazon Location Service, Anda harus selalu mengetahui [Ketentuan AWS Layanan](#) yang berlaku.

## Terminologi tempat

### Tempatkan sumber daya indeks

Memungkinkan Anda memilih sumber data untuk mendukung permintaan pencarian. Misalnya, Anda dapat mencari tempat menarik, alamat, atau koordinat. Ketika kueri penelusuran dikirim ke sumber daya indeks tempat, itu dipenuhi menggunakan sumber data sumber daya yang dikonfigurasi.

### Geocoding

Geocoding adalah proses mengambil input teks, mencarinya di indeks tempat, dan mengembalikan hasil dengan posisi.

### Geocoding terbalik

Reverse geocoding adalah proses mengambil posisi dan mengembalikan informasi tentang posisi itu dari dalam indeks tempat, seperti alamat, kota, atau bisnis di lokasi tersebut.

### Relevansi

Relevansi adalah seberapa dekat hasil cocok dengan input. Ini bukan ukuran kebenaran.

### Interpolasi

Interpolasi adalah proses menemukan alamat yang tidak diketahui dengan menggunakan lokasi alamat yang dikenal sebagai titik panduan.

## ISO 3166 kode negara

Amazon Location Service Places menggunakan kode negara [Organisasi Internasional untuk Standardisasi \(ISO\) 3166](#) untuk merujuk ke negara atau wilayah.

Untuk menemukan kode untuk negara atau wilayah tertentu, gunakan [Platform Penjelajahan Online ISO](#).

## Rute

Bagian ini memberikan gambaran umum tentang konsep seputar perutean menggunakan Amazon Location Service.

### Note

Untuk informasi tentang cara menggunakan sumber daya rute dalam praktik, lihat [Menghitung rute menggunakan Amazon Location Service](#).

## Sumber daya kalkulator rute

Sumber daya kalkulator rute memungkinkan Anda menemukan rute dan memperkirakan waktu perjalanan berdasarkan jaringan up-to-date jalan dan informasi lalu lintas langsung dari penyedia data pilihan Anda.

Anda dapat menggunakan Route API untuk membangun fitur yang memungkinkan aplikasi Anda meminta waktu perjalanan, jarak, dan geometri rute antara dua lokasi. Anda juga dapat menggunakan API Rute untuk meminta waktu dan jarak perjalanan antara satu set keberangkatan dan tujuan dalam satu permintaan untuk menghitung matriks.

Berikut ini menunjukkan cara membuat dan menggunakan sumber kalkulator rute:



1. Pertama, Anda membuat sumber kalkulator rute di AWS akun Anda dengan memilih penyedia data.



2. Anda kemudian dapat memilih dan menginstal SDK yang cocok dengan lingkungan pengembangan dan aplikasi Anda.
3. Mulai gunakan Amazon Location Routes API. Untuk informasi selengkapnya tentang cara menggunakan API perutean, lihat topik di [Menghitung rute menggunakan Amazon Location Service](#).
4. Anda kemudian dapat mengintegrasikan pemantauan menggunakan layanan seperti Amazon CloudWatch dan AWS CloudTrail. Untuk informasi lebih lanjut lihat, [Memantau Amazon Location Service dengan Amazon CloudWatch](#) dan [Pencatatan dan pemantauan dengan AWS CloudTrail](#).

## Menghitung rute

Sumber daya kalkulator rute Lokasi Amazon menyediakan tindakan `CalculateRoute` yang disebut yang dapat Anda gunakan untuk membuat rute antara dua lokasi geografis (keberangkatan dan tujuan). Rute yang dihitung mencakup geometri untuk menggambar rute pada peta, ditambah waktu dan jarak keseluruhan rute.

### Menggunakan titik arah

Saat Anda membuat permintaan rute, Anda dapat menambahkan titik arah tambahan ke rute. Ini adalah titik antara keberangkatan dan tujuan yang bertindak sebagai pemberhentian di sepanjang rute. Rute akan dihitung melalui masing-masing titik arah yang ditentukan. Rute dari satu titik dalam permintaan ke titik berikutnya disebut `aLeg`. Setiap kaki mencakup jarak, waktu, dan geometri untuk bagian rute itu.

#### Note

Titik arah dirutekan sesuai urutan yang diberikan dalam permintaan. Mereka tidak dipesan ulang untuk jalur terpendek. Lihat [Merencanakan rute](#) bagian untuk informasi tentang menemukan jalur terpendek.

Anda dapat menyertakan hingga 25 titik arah dalam satu permintaan untuk menghitung rute.

### Lalu lintas dan waktu keberangkatan

Amazon Location Service memperhitungkan lalu lintas saat menghitung rute. Lalu lintas yang dianggapnya didasarkan pada waktu yang Anda tentukan. Anda dapat menentukan untuk berangkat

sekarang, atau Anda dapat memberikan waktu tertentu yang ingin Anda tinggalkan, yang akan memengaruhi hasil rute dengan menyesuaikan lalu lintas pada waktu yang ditentukan.

#### Note

Anda dapat menghitung waktu kedatangan menggunakan waktu keberangkatan dan waktu respons rute, untuk memperkirakan kedatangan pengemudi, misalnya.

Jika Anda ingin Lokasi Amazon tidak memperhitungkan lalu lintas, maka jangan tentukan waktu keberangkatan dan jangan tentukan keberangkatan sekarang. Ini akan menghitung rute yang mengasumsikan kondisi lalu lintas terbaik untuk rute tersebut.

#### Opsi mode perjalanan

Anda dapat mengatur mode perjalanan saat menghitung rute menggunakan Amazon Location Service. Mode perjalanan default adalah mobil, tetapi Anda dapat secara bergantian memilih truk atau berjalan.

Jika Anda menentukan mode mobil atau truk, Anda juga dapat menentukan opsi tambahan.

Untuk mode mobil, Anda dapat menentukan bahwa Anda ingin menghindari jalan tol atau feri. Ini akan berusaha untuk menghindari feri dan jalan tol, tetapi masih akan rute di sepanjang mereka, jika mereka adalah satu-satunya cara untuk sampai ke tujuan.

Untuk mode truk, Anda juga dapat menghindari feri dan jalan tol, tetapi selain itu, Anda dapat menentukan ukuran dan berat truk, untuk menghindari rute yang tidak akan mengakomodasi truk.

## Merencanakan rute

Anda dapat menggunakan Amazon Location Service untuk membuat input ke perangkat lunak perencanaan dan pengoptimalan rute Anda. Anda dapat membuat hasil rute, termasuk waktu perjalanan dan jarak perjalanan, untuk rute antara serangkaian posisi keberangkatan dan serangkaian posisi tujuan. Ini disebut membuat matriks rute.

#### Note

Ada banyak skenario berbeda yang dapat diselesaikan oleh perangkat lunak perencanaan dan pengoptimalan rute. Misalnya, perangkat lunak perencanaan dapat menggunakan rangkaian waktu dan jarak antar titik untuk menghitung jalur terpendek yang berhenti di

setiap titik, menyediakan rute yang efisien untuk satu pengemudi. Atau, perangkat lunak perencanaan dapat digunakan untuk membagi pemberhentian antara beberapa truk, memberikan efisiensi di seluruh armada, atau untuk memastikan bahwa setiap pelanggan dikunjungi dalam jangka waktu yang mereka butuhkan. Lokasi Amazon menyediakan fungsi perutean dengan cara yang efisien untuk memungkinkan perangkat lunak perencanaan menyelesaikan tugasnya.

Misalnya, mengingat posisi keberangkatan A dan B, dan posisi tujuan X dan Y, Amazon Location Service akan mengembalikan waktu perjalanan dan jarak perjalanan untuk rute dari A ke X, A ke Y, B ke X, dan B ke Y.

Seperti halnya menghitung satu rute, Anda dapat menghitung rute dengan berbagai moda transportasi, penghindaran, dan kondisi lalu lintas. Misalnya, Anda dapat menentukan bahwa kendaraan tersebut adalah truk yang panjangnya 35 kaki, dan rute yang dihitung akan menggunakan batasan tersebut untuk menentukan waktu tempuh dan jarak tempuh. Anda tidak dapat menyertakan titik arah dalam perhitungan matriks rute.

Jumlah hasil yang dikembalikan (dan rute dihitung) adalah jumlah posisi keberangkatan dikalikan dengan jumlah posisi tujuan. Anda dikenakan biaya untuk setiap rute yang dihitung, bukan setiap permintaan ke layanan, sehingga matriks rute dengan 10 keberangkatan dan 10 tujuan akan ditagih sebagai 100 rute.

## Terminologi rute

### Sumber daya kalkulator rute

AWS Sumber daya yang memungkinkan Anda memperkirakan waktu perjalanan, jarak, dan rute plot pada peta dengan lalu lintas dan data jaringan jalan yang bersumber dari penyedia data pilihan Anda.

Dengan menggunakan sumber kalkulator rute, Anda menghitung rute untuk berbagai moda transportasi, jalan memutar, dan kondisi lalu lintas.

### Rute

Rute berisi rincian yang digunakan saat bepergian di sepanjang jalur dari posisi keberangkatan, posisi waypoint, dan posisi tujuan.

Contoh detail dalam rute meliputi:

- Jarak dari satu posisi ke posisi lain.
- Waktu yang dibutuhkan untuk melakukan perjalanan dari satu posisi ke posisi berikutnya.
- LineString Geometri yang mewakili jalur rute.

Untuk informasi selengkapnya tentang rute, lihat [sintaks respons untuk CalculateRoute operasi di referensi Amazon Location Service Routes API](#).

## Matriks rute

Daftar rute, dari satu set posisi keberangkatan ke satu set posisi tujuan. Berguna sebagai input ke dalam perencanaan rute atau perangkat lunak pengoptimalan.

Untuk informasi selengkapnya tentang menghitung matriks rute, lihat [sintaks untuk CalculateRouteMatrix operasi di referensi Amazon Location Service Routes API](#).

## LineStringgeometri

Rute Lokasi Amazon terdiri dari satu atau lebih kaki (rute dari satu titik arah ke titik lain dalam rute keseluruhan). Geometri setiap kaki adalah polyline yang direpresentasikan sebagai a. LineString A LineString adalah susunan posisi yang dapat digunakan untuk merencanakan rute pada peta.

Berikut ini adalah contoh dari a LineString dengan tiga poin:

```
[  
  [-122.7565, 49.0021],  
  [-122.3394, 47.6159],  
  [-122.1082, 45.8371]  
]
```

## Titik arah

Waypoint adalah posisi perantara yang bertindak sebagai pemberhentian di sepanjang rute antara posisi keberangkatan dan posisi tujuan. Perintah persinggahan pada rute mengikuti urutan yang Anda berikan posisi waypoint dalam permintaan.

## Kaki

Satu kaki adalah perjalanan dari satu posisi ke posisi lain. Jika posisi tidak terletak di jalan, mereka dipindahkan ke jalan terdekat. Jumlah kaki dalam suatu rute adalah satu kurang dari jumlah total posisi.

Rute tanpa titik arah terdiri dari satu kaki, dari posisi keberangkatan ke tujuan. Rute dengan 1 waypoint terdiri dari 2 kaki, dari posisi keberangkatan ke waypoint, dan kemudian dari waypoint ke tujuan.

## Langkah

Langkah adalah subbagian dari kaki. Setiap langkah memberikan informasi ringkasan untuk langkah di kaki itu.

## Geofences dan Pelacak

Bagian ini menyediakan dan ikhtisar konsep bekerja dengan geofences dan pelacak Amazon Location Service. Geofences adalah batas poligon yang dapat Anda gunakan untuk diberi tahu saat perangkat atau posisi bergerak masuk dan keluar dari area. Sumber daya pelacak digunakan untuk menyimpan dan memperbarui posisi untuk perangkat saat mereka bergerak.

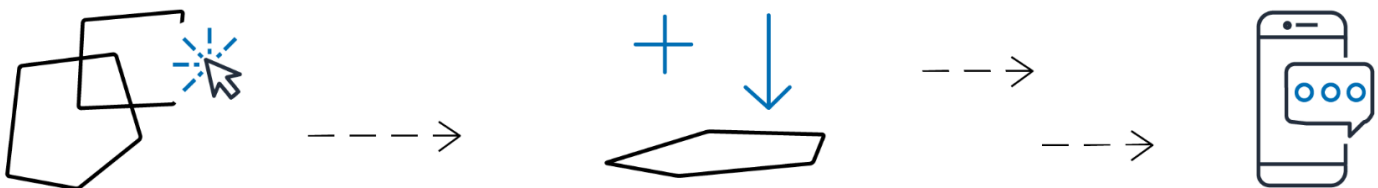
### Note

Untuk informasi tentang cara menggunakan geofences dan pelacak dalam praktik, lihat [Geofencing area yang diminati menggunakan Lokasi Amazon](#)

## Geofences

Sumber daya pengumpulan geofence memungkinkan Anda untuk menyimpan dan mengelola geofences—batas virtual pada peta. Anda dapat mengevaluasi lokasi terhadap sumber daya pengumpulan geofence dan mendapatkan pemberitahuan saat pembaruan lokasi melintasi batas geofences mana pun dalam koleksi geofence.

Berikut ini menunjukkan kepada Anda cara membuat dan menggunakan sumber daya pengumpulan geofence:



1. Buat sumber daya koleksi geofence di akun Anda AWS .

2. Tambahkan geofences ke koleksi itu. Anda dapat melakukannya dengan menggunakan alat unggah geofence di konsol Lokasi Amazon, atau dengan menggunakan Amazon Location Geofences API. Untuk informasi selengkapnya tentang opsi yang tersedia, lihat [Mengakses Lokasi Amazon](#).

Geofences dapat didefinisikan oleh poligon atau oleh lingkaran. Gunakan poligon untuk menemukan kapan perangkat memasuki area tertentu. Gunakan lingkaran untuk menemukan ketika perangkat berada dalam jarak tertentu (radius) dari suatu titik.

3. Anda dapat mulai mengevaluasi lokasi terhadap semua geofences Anda. Saat pembaruan lokasi melintasi batas satu atau beberapa geofences, sumber daya pengumpulan geofence Anda akan memancarkan salah satu jenis peristiwa geofence berikut di Amazon: EventBridge
  - ENTER - Satu acara dihasilkan untuk setiap geofence di mana pembaruan lokasi melintasi batasnya dengan memasukkannya.
  - EXIT - Satu peristiwa dihasilkan untuk setiap geofence di mana pembaruan lokasi melintasi batasnya dengan keluar dari itu.

Untuk informasi selengkapnya, lihat [the section called “Bereaksi terhadap peristiwa dengan EventBridge”](#). Anda juga dapat mengintegrasikan pemantauan menggunakan layanan seperti Amazon CloudWatch dan AWS CloudTrail. Untuk informasi lebih lanjut lihat, [the section called “Pemantauan CloudWatch dengan”](#) dan [the section called “Menggunakan CloudTrail dengan Lokasi Amazon”](#).

Misalnya, jika Anda melacak armada truk, dan Anda ingin mendapatkan pemberitahuan ketika sebuah truk datang dalam area tertentu dari salah satu gudang Anda. Anda dapat membuat geofence untuk area di sekitar setiap gudang. Kemudian, ketika truk mengirim Anda lokasi yang diperbarui, Anda dapat menggunakan Amazon Location Service untuk mengevaluasi posisi tersebut dan melihat apakah truk telah memasuki (atau keluar) salah satu area geofence.

#### Note

Anda ditagih oleh jumlah koleksi geofence yang Anda evaluasi. Tagihan Anda tidak terpengaruh oleh jumlah geofences di setiap koleksi. Karena setiap koleksi geofence dapat berisi hingga 50,000 geofences, Anda mungkin ingin menggabungkan geofences Anda menjadi koleksi yang lebih sedikit, jika memungkinkan, untuk mengurangi biaya evaluasi geofence Anda. Peristiwa yang dihasilkan akan mencakup ID geofence individu dalam koleksi, serta ID koleksi.

## Acara geofence

Lokasi untuk posisi yang Anda pantau direferensikan oleh ID yang disebut `DeviceId` (dan posisi disebut sebagai posisi perangkat). Anda dapat mengirim daftar posisi perangkat untuk mengevaluasi langsung ke sumber pengumpulan geofence, atau Anda dapat menggunakan pelacak. Lihat bagian selanjutnya untuk informasi lebih lanjut tentang pelacak.

Anda menerima peristiwa (melalui Amazon EventBridge) hanya ketika perangkat masuk atau keluar dari geofence, tidak untuk setiap perubahan posisi. Ini berarti bahwa Anda biasanya akan menerima acara dan harus meresponsnya jauh lebih jarang daripada setiap pembaruan posisi perangkat.

### Note

Untuk evaluasi lokasi pertama untuk `specificDeviceID`, diasumsikan bahwa perangkat sebelumnya tidak berada di geofences apa pun. Jadi pembaruan pertama akan menghasilkan ENTER acara, jika di dalam geofence dalam koleksi, dan tidak ada acara jika tidak.

Untuk menghitung apakah perangkat telah masuk atau keluar dari geofence, Amazon Location Service harus mempertahankan status posisi sebelumnya untuk perangkat tersebut. Status posisi ini disimpan selama 30 hari. Setelah 30 hari tanpa pembaruan untuk perangkat, pembaruan lokasi baru akan diperlakukan sebagai pembaruan posisi pertama.

## Pelacak

Pelacak menyimpan pembaruan posisi untuk koleksi perangkat. Pelacak dapat digunakan untuk menanyakan lokasi perangkat saat ini atau riwayat lokasi. Ini menyimpan pembaruan, tetapi mengurangi ruang penyimpanan dan kebisingan visual dengan memfilter lokasi sebelum menyimpannya.

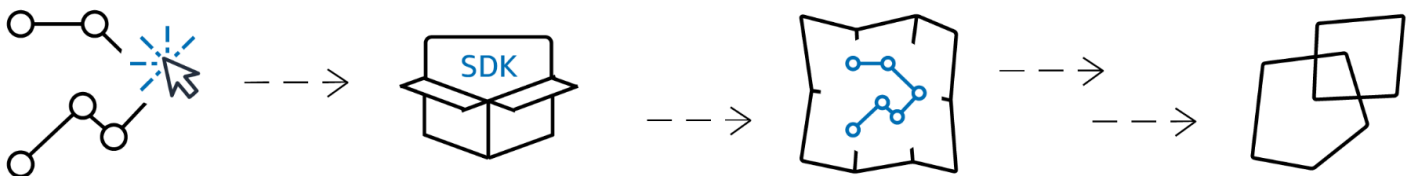
Setiap pembaruan posisi yang disimpan dalam sumber daya pelacak Anda dapat mencakup ukuran akurasi posisi dan hingga 3 bidang metadata tentang posisi atau perangkat yang ingin Anda simpan. Metadata disimpan sebagai pasangan nilai kunci, dan dapat menyimpan informasi seperti kecepatan, arah, tekanan ban, atau suhu mesin.

**Note**

Penyimpanan pelacak dienkripsi dengan kunci yang AWS dimiliki secara otomatis. Anda dapat menambahkan lapisan enkripsi lain menggunakan kunci KMS yang Anda kelola, untuk memastikan bahwa hanya Anda yang dapat mengakses data Anda. Untuk informasi selengkapnya, lihat [Enkripsi data saat istirahat untuk Amazon Location Service](#).

Pemfilteran dan penyimpanan posisi pelacak berguna sendiri, tetapi pelacak sangat berguna saat dipasangkan dengan geofences. Anda dapat menautkan pelacak ke satu atau beberapa sumber daya pengumpulan geofence Anda, dan pembaruan posisi dievaluasi secara otomatis terhadap geofences dalam koleksi tersebut. Penggunaan penyaringan yang tepat dapat sangat mengurangi biaya evaluasi geofence Anda juga.

Diagram berikut menunjukkan cara membuat dan menggunakan sumber daya pelacak:



1. Pertama, Anda membuat sumber daya pelacak di AWS akun Anda.
2. Selanjutnya, putuskan bagaimana Anda mengirim pembaruan lokasi ke sumber daya pelacak Anda. Gunakan [AWS SDK](#) untuk mengintegrasikan kemampuan pelacakan ke dalam aplikasi seluler Anda. [Sebagai alternatif, Anda dapat menggunakan MQTT dengan mengikuti step-by-step petunjuk dalam melacak menggunakan MQTT.](#)
3. Anda sekarang dapat menggunakan sumber daya pelacak Anda untuk merekam riwayat lokasi dan memvisualisasikannya di peta.
4. Anda juga dapat menautkan sumber daya pelacak Anda ke satu atau beberapa koleksi geofence sehingga setiap pembaruan posisi yang dikirim ke sumber daya pelacak Anda dievaluasi secara otomatis terhadap semua geofence di semua koleksi geofence yang ditautkan. Anda dapat menautkan sumber daya di halaman detail sumber daya pelacak di konsol Lokasi Amazon atau dengan menggunakan Amazon Location Trackers API.
5. Anda kemudian dapat mengintegrasikan pemantauan menggunakan layanan seperti Amazon CloudWatch dan AWS CloudTrail. Untuk informasi lebih lanjut lihat, [the section called “Pemantauan CloudWatch dengan”](#) dan [the section called “Menggunakan CloudTrail dengan Lokasi Amazon”](#).



## Menggunakan pelacak dengan geofences

Pelacak menyediakan fungsionalitas tambahan saat dipasangkan dengan geofences. Anda mengaitkan pelacak dengan koleksi geofence, baik melalui konsol Lokasi Amazon atau API, untuk mengevaluasi lokasi pelacak secara otomatis. Setiap kali pelacak menerima lokasi yang diperbarui, lokasi tersebut akan dievaluasi terhadap setiap geofence dalam koleksi, dan yang sesuai ENTER serta EXIT peristiwa dihasilkan di Amazon. EventBridge Anda juga dapat menerapkan pemfilteran ke pelacak, dan, tergantung pada pemfilterannya, Anda dapat mengurangi biaya untuk evaluasi geofence dengan hanya mengevaluasi pembaruan lokasi yang bermakna.

Jika Anda mengaitkan pelacak dengan koleksi geofence setelah menerima beberapa pembaruan posisi, pembaruan posisi pertama setelah asosiasi diperlakukan sebagai pembaruan awal untuk evaluasi geofence. Jika berada dalam geofence, Anda akan menerima acaraENTER. Jika tidak dalam geofences apa pun, Anda tidak akan menerima EXIT acara, terlepas dari keadaan sebelumnya.

### Penyaringan posisi

Pelacak dapat secara otomatis memfilter posisi yang dikirim kepada mereka. Ada beberapa alasan mengapa Anda mungkin ingin memfilter beberapa pembaruan lokasi perangkat Anda. Jika Anda memiliki sistem yang hanya mengirim laporan setiap menit atau lebih, Anda mungkin ingin memfilter perangkat berdasarkan waktu, menyimpan dan mengevaluasi posisi hanya setiap 30 detik. Bahkan jika Anda memantau lebih sering, Anda mungkin ingin memfilter pembaruan posisi untuk membersihkan kebisingan perangkat keras GPS. Lokasi posisi GPS secara inheren berisik. Akurasi tidak 100% sempurna, sehingga bahkan perangkat yang stasioner tampaknya bergerak sedikit. Pada kecepatan rendah, jitter ini menyebabkan kekacauan visual dan dapat menyebabkan peristiwa masuk dan keluar yang salah jika perangkat berada di dekat tepi geofence.

Pemfilteran posisi berfungsi saat pembaruan posisi diterima oleh pelacak, mengurangi gangguan visual di jalur perangkat Anda (jitter), mengurangi jumlah peristiwa masuk dan keluar geofence palsu, dan membantu mengelola biaya dengan mengurangi jumlah pembaruan posisi yang disimpan dan evaluasi geofence yang dipicu.

Pelacak menawarkan tiga opsi pemfilteran posisi untuk membantu mengelola biaya dan mengurangi jitter di pembaruan lokasi Anda.

- Berbasis akurasi - Gunakan dengan perangkat apa pun yang memberikan pengukuran akurasi. Sebagian besar GPS dan perangkat seluler menyediakan informasi ini. Keakuratan setiap pengukuran posisi dipengaruhi oleh banyak faktor lingkungan, termasuk penerimaan satelit GPS, lanskap, dan kedekatan perangkat wifi dan bluetooth. Sebagian besar perangkat, termasuk

sebagian besar perangkat seluler, dapat memberikan perkiraan keakuratan pengukuran bersama dengan pengukuran. Dengan AccuracyBased pemfilteran, Lokasi Amazon mengabaikan pembaruan lokasi jika perangkat bergerak kurang dari akurasi yang diukur. Misalnya, jika dua pembaruan berturut-turut dari perangkat memiliki rentang akurasi 5 m dan 10 m, Amazon Location mengabaikan pembaruan kedua jika perangkat telah bergerak kurang dari 15 m. Lokasi Amazon tidak mengevaluasi pembaruan yang diabaikan terhadap geofences, atau menyimpannya.

Ketika akurasi tidak disediakan, itu diperlakukan sebagai nol, dan pengukuran dianggap sangat akurat, dan tidak ada penyaringan yang akan diterapkan pada pembaruan.

#### Note

Anda dapat menggunakan pemfilteran berbasis akurasi untuk menghapus semua pemfilteran. Jika Anda memilih pemfilteran berbasis akurasi, tetapi mengganti semua data akurasi ke nol, atau menghilangkan akurasi sepenuhnya, maka Lokasi Amazon tidak akan memfilter pembaruan apa pun.

Dalam sebagian besar skenario, pemfilteran berbasis akurasi adalah pilihan yang baik untuk memfilter pembaruan posisi, memberikan keseimbangan lokasi pelacakan sambil memfilter pembaruan yang tidak diperlukan, sehingga mengurangi biaya.

- Berbasis jarak — Gunakan saat perangkat Anda tidak memberikan pengukuran akurasi, tetapi Anda tetap ingin memanfaatkan penyaringan untuk mengurangi jitter dan mengelola biaya. DistanceBased pemfilteran mengabaikan pembaruan lokasi di mana perangkat telah bergerak kurang dari 30 m (98,4 kaki). Saat Anda menggunakan pemfilteran DistanceBased posisi, Amazon Location tidak mengevaluasi pembaruan yang diabaikan ini terhadap geofences atau menyimpan pembaruan.

Keakuratan sebagian besar perangkat seluler, termasuk akurasi rata-rata perangkat iOS dan Android, berada dalam jarak 15 m. Di sebagian besar aplikasi, DistanceBased penyaringan dapat mengurangi efek ketidakakuratan lokasi saat menampilkan lintasan perangkat pada peta, dan efek pantulan dari beberapa peristiwa masuk dan keluar berturut-turut saat perangkat berada di dekat perbatasan geofence. Hal ini juga dapat membantu mengurangi biaya aplikasi Anda, dengan membuat lebih sedikit panggilan untuk mengevaluasi terhadap geofences terkait atau mengambil posisi perangkat.

Pemfilteran berbasis jarak berguna jika Anda ingin memfilter, tetapi perangkat Anda tidak memberikan pengukuran akurasi, atau Anda ingin memfilter lebih banyak pembaruan dibandingkan dengan berbasis akurasi.

- Berbasis waktu — (default) Gunakan saat perangkat Anda mengirim pembaruan posisi sangat sering (lebih dari sekali setiap 30 detik), dan Anda ingin mencapai evaluasi geofence mendekati waktu nyata tanpa menyimpan setiap pembaruan. Dalam TimeBased pemfilteran, setiap pembaruan lokasi dievaluasi terhadap koleksi geofence yang ditautkan, tetapi tidak setiap pembaruan lokasi disimpan. Jika frekuensi pembaruan Anda lebih sering dari 30 detik, hanya satu pembaruan per 30 detik yang disimpan untuk setiap ID perangkat unik.

Pemfilteran berbasis waktu sangat berguna ketika Anda ingin menyimpan lebih sedikit posisi, tetapi ingin setiap pembaruan posisi dievaluasi terhadap koleksi geofence terkait.

#### Note

Perhatikan biaya aplikasi pelacakan Anda saat memutuskan metode penyaringan Anda dan frekuensi pembaruan posisi. Anda ditagih untuk setiap pembaruan lokasi dan sekali untuk mengevaluasi pembaruan posisi terhadap setiap koleksi geofence yang ditautkan. Misalnya, saat menggunakan pemfilteran berbasis waktu, jika pelacak Anda ditautkan ke dua koleksi geofence, setiap pembaruan posisi akan dihitung sebagai satu permintaan pembaruan lokasi dan dua evaluasi pengumpulan geofence. Jika Anda melaporkan pembaruan posisi setiap 5 detik untuk perangkat Anda dan menggunakan pemfilteran berbasis waktu, Anda akan ditagih untuk 720 pembaruan lokasi dan 1.440 evaluasi geofensi per jam untuk setiap perangkat.

## Terminologi geofensi

### Koleksi Geofence

Berisi nol atau lebih geofences. Ia mampu memantau geofence dengan memancarkan peristiwa Masuk dan Keluar, ketika diminta, untuk mengevaluasi posisi perangkat terhadap geofences nya.

### Geofensi

Sebuah geometri poligon atau lingkaran yang mendefinisikan batas virtual pada peta.

## Geometri poligon

Geofence Lokasi Amazon adalah batas virtual untuk wilayah geografis dan direpresentasikan sebagai geometri poligon atau sebagai lingkaran.

Lingkaran adalah titik dengan jarak di sekitarnya. Gunakan lingkaran saat Anda ingin diberi tahu jika perangkat berada dalam jarak tertentu dari suatu lokasi.

Poligon adalah array yang terdiri dari 1 atau lebih cincin linier. Gunakan poligon saat Anda ingin menentukan batas tertentu untuk pemberitahuan perangkat. Cincin linier adalah susunan empat atau lebih simpul, di mana simpul pertama dan terakhir sama untuk membentuk batas tertutup. Setiap simpul adalah titik 2 dimensi dari bentuk *[bujur, lintang]*, di mana satuan bujur dan lintang adalah derajat. Simpul harus terdaftar dalam urutan berlawanan arah jarum jam di sekitar poligon.

### Note

Amazon Location Service tidak mendukung poligon dengan lebih dari satu cincin. Ini termasuk lubang, pulau atau multipoligon. Lokasi Amazon juga tidak mendukung poligon yang dililit searah jarum jam, atau yang melintasi antimeridian.

Berikut ini adalah contoh cincin eksternal linier tunggal:

```
[
  [
    [-5.716667, -15.933333],
    [-14.416667, -7.933333],
    [-12.316667, -37.066667],
    [-5.716667, -15.933333]
  ]
]
```

## Terminologi pelacak

### Sumber daya pelacak

AWS Sumber daya yang menerima pembaruan lokasi dari perangkat. Sumber daya pelacak menyediakan dukungan untuk kueri lokasi, seperti lokasi perangkat saat ini dan bersejarah.

Menautkan sumber daya pelacak ke koleksi geofence mengevaluasi pembaruan lokasi terhadap semua geofences dalam koleksi geofence tertaut secara otomatis.

### Data posisi dilacak

Sumber daya pelacak menyimpan informasi tentang perangkat Anda dari waktu ke waktu. Informasi tersebut mencakup serangkaian pembaruan posisi, di mana setiap pembaruan mencakup lokasi, waktu, dan metadata opsional. Metadata dapat mencakup akurasi posisi, dan hingga tiga pasangan nilai kunci untuk membantu Anda melacak informasi penting tentang setiap posisi, seperti kecepatan, arah, tekanan ban, sisa bahan bakar, atau suhu mesin kendaraan yang Anda lacak. Pelacak mempertahankan riwayat lokasi perangkat selama 30 hari.

### Penyaringan posisi

Pemfilteran posisi dapat membantu Anda mengontrol biaya dan meningkatkan kualitas aplikasi pelacakan Anda dengan memfilter pembaruan posisi yang tidak memberikan informasi berharga sebelum pembaruan disimpan atau dievaluasi terhadap geofences.

Anda dapat memilih `AccuracyBased`, `DistanceBased`, atau `TimeBased` memfilter. Secara default, penyaringan posisi diatur ke `TimeBased`.

Anda dapat mengonfigurasi pemfilteran posisi saat membuat atau memperbarui sumber daya pelacak.

### Format stempel waktu RFC 3339

Amazon Location Service Trackers menggunakan format [RFC 3339](#), yang mengikuti format [International Organization for Standardization \(ISO\) 8601](#) untuk tanggal dan waktu.

Formatnya adalah “yyyy-mm-ddthh: mm:ss.sssz+ 00:00”:

- YYYY-MM-DD— Merupakan format tanggal.
- T— Menunjukkan bahwa nilai waktu akan mengikuti.
- hh:mm:ss.sss— Merupakan waktu dalam format 24 jam.
- Z— Menunjukkan bahwa zona waktu yang digunakan adalah UTC, yang dapat diikuti dengan penyimpangan dari zona waktu UTC.
- +00:00— Secara opsional menunjukkan penyimpangan dari zona waktu UTC. Misalnya, +01:00 menunjukkan UTC + 1 jam.

### Contoh

Untuk 2 Juli 2020, pukul 12:15:20 sore, dengan penyesuaian tambahan 1 jam ke zona waktu UTC.

2020-07-02T12:15:20.000Z+01:00

## Kasus penggunaan umum untuk menggunakan Amazon Location Service

Amazon Location Service memungkinkan Anda membangun berbagai aplikasi, mulai dari pelacakan aset hingga pemasaran berbasis lokasi. Berikut ini adalah kasus penggunaan umum:

### Keterlibatan pengguna dan geomarketing

Gunakan data lokasi untuk membangun solusi yang meningkatkan keterlibatan pengguna dengan pemasaran untuk menargetkan pelanggan. Misalnya, Lokasi Amazon dapat memicu peristiwa yang meminta pemberitahuan ketika pelanggan yang memesan kopi di aplikasi seluler mereka berada di dekatnya. Selain itu, Anda dapat membangun fitur penargetan geografis sehingga pengecer dapat mengirim kode diskon atau selebaran digital kepada pelanggan yang berada di dekat toko target.

### Pelacakan aset

Buat fitur pelacakan aset untuk membantu bisnis memahami lokasi terkini dan historis produk, personel, dan infrastruktur mereka. Dengan fitur pelacakan aset, Anda dapat membangun sejumlah solusi yang mengoptimalkan kepegawaian jarak jauh, mengamankan pengiriman dalam perjalanan, dan memaksimalkan kemandirian pengiriman.

### Pengiriman

Integrasikan fitur lokasi ke dalam aplikasi pengiriman untuk menyimpan, melacak, dan mengoordinasikan lokasi keberangkatan, kendaraan pengiriman, dan tujuan mereka. Misalnya, aplikasi pengiriman makanan dengan fitur Lokasi Amazon bawaan memiliki kemampuan pelacakan lokasi dan geofencing yang dapat memberi tahu restoran secara otomatis saat pengemudi pengiriman berada di dekatnya. Ini mengurangi waktu tunggu dan membantu menjaga kualitas makanan yang dikirim.

Topik ini memberi Anda gambaran umum tentang arsitektur dan langkah-langkah untuk aplikasi yang dapat Anda buat dengan Amazon Location.

## Topik

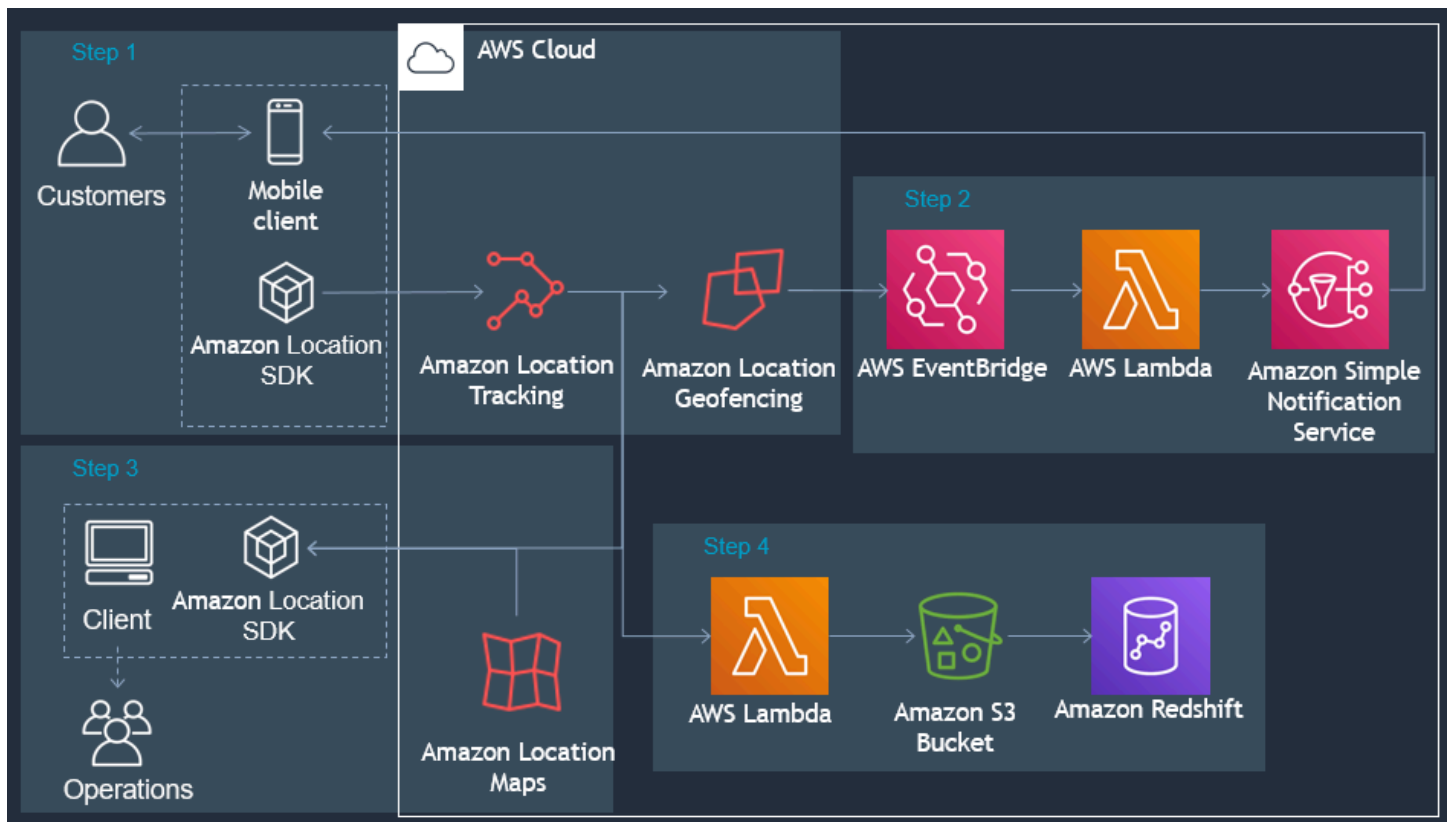
- [Keterlibatan pengguna dan aplikasi geomarketing](#)
- [Aplikasi pelacakan aset](#)
- [Aplikasi pengiriman](#)

## Keterlibatan pengguna dan aplikasi geomarketing

Berikut ini adalah ilustrasi dari keterlibatan pengguna dan arsitektur aplikasi geomarketing menggunakan Amazon Location:

Dengan arsitektur ini, Anda dapat:

- Memulai acara berdasarkan kedekatan target sehingga Anda dapat mengirim penawaran ke pelanggan terdekat atau melibatkan mereka yang baru saja meninggalkan perusahaan Anda (disebut penargetan geografis).
- Visualisasikan lokasi perangkat pelanggan di peta untuk memantau tren dari waktu ke waktu.
- Simpan lokasi perangkat pelanggan yang dapat Anda analisis dari waktu ke waktu.
- Menganalisis riwayat lokasi untuk mengidentifikasi tren dan peluang untuk optimasi.



Berikut ini adalah ikhtisar langkah-langkah yang diperlukan untuk membangun keterlibatan pengguna dan aplikasi geomarketing:

1. Buat geofences Anda di Koleksi Geofence dan tautkan Pelacak ke mereka. Untuk informasi selengkapnya, lihat [the section called “Geofencing dan pelacakan”](#).
2. Konfigurasi Amazon EventBridge untuk mengirim pemberitahuan kepada pelanggan yang masuk atau keluar dari area minat yang digeofensikan. Untuk informasi selengkapnya, lihat [the section called “Bereaksi terhadap peristiwa dengan EventBridge”](#).
3. Menampilkan lokasi pelanggan dan geofences pada peta. Untuk informasi selengkapnya, lihat [Menggunakan peta](#).
4. Simpan data lokasi ke penyimpanan jangka panjang untuk analisis lebih lanjut.
5. Setelah Anda membangun aplikasi Anda, Anda dapat menggunakan Amazon CloudWatch dan AWS CloudTrail mengelola aplikasi Anda. Untuk informasi selengkapnya, lihat [the section called “Pemantauan CloudWatch dengan”](#) dan [the section called “Menggunakan CloudTrail dengan Lokasi Amazon”](#).

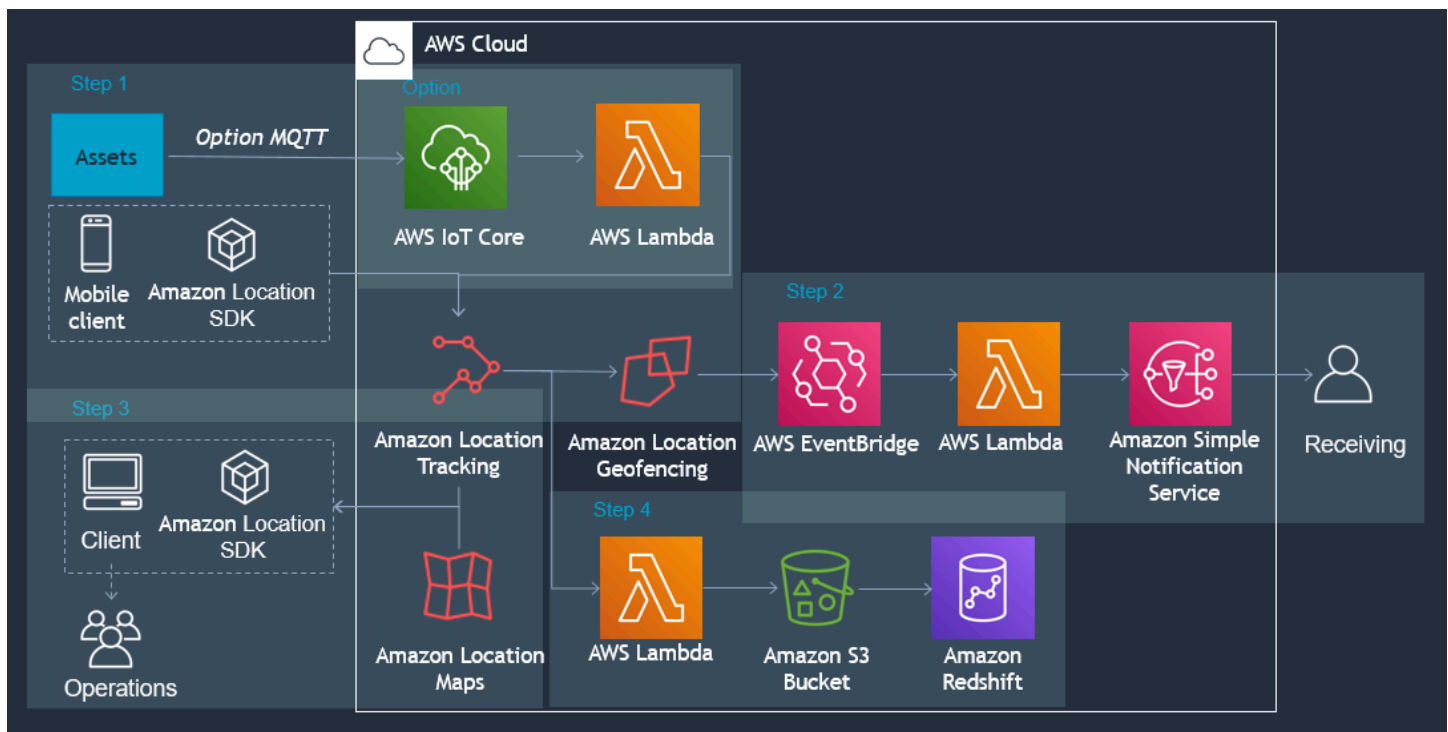


## Aplikasi pelacakan aset

Berikut ini adalah ilustrasi arsitektur aplikasi pelacakan aset menggunakan Lokasi Amazon:

Dengan arsitektur ini, Anda dapat:

- Tampilkan lokasi aset pada peta untuk mengilustrasikan gambaran besar. Misalnya, menampilkan peta panas menggunakan lokasi atau peristiwa bersejarah untuk membantu tim operasi atau perencanaan.
- Memulai acara berdasarkan kedekatan aset untuk memberikan pemberitahuan kepada departemen penerima untuk mempersiapkan kedatangan pengiriman dan mengurangi waktu pemrosesan.
- Simpan lokasi aset untuk memulai tindakan dalam aplikasi backend Anda atau untuk menganalisis data dari waktu ke waktu.
- Menganalisis riwayat lokasi untuk mengidentifikasi tren dan peluang untuk optimasi.



Berikut ini memberikan ikhtisar langkah-langkah yang diperlukan untuk membangun aplikasi pelacakan aset:

1. Buat geofences Anda di Koleksi Geofence dan tautkan Pelacak ke mereka. Untuk informasi selengkapnya, lihat [the section called “Geofencing dan pelacakan”](#).

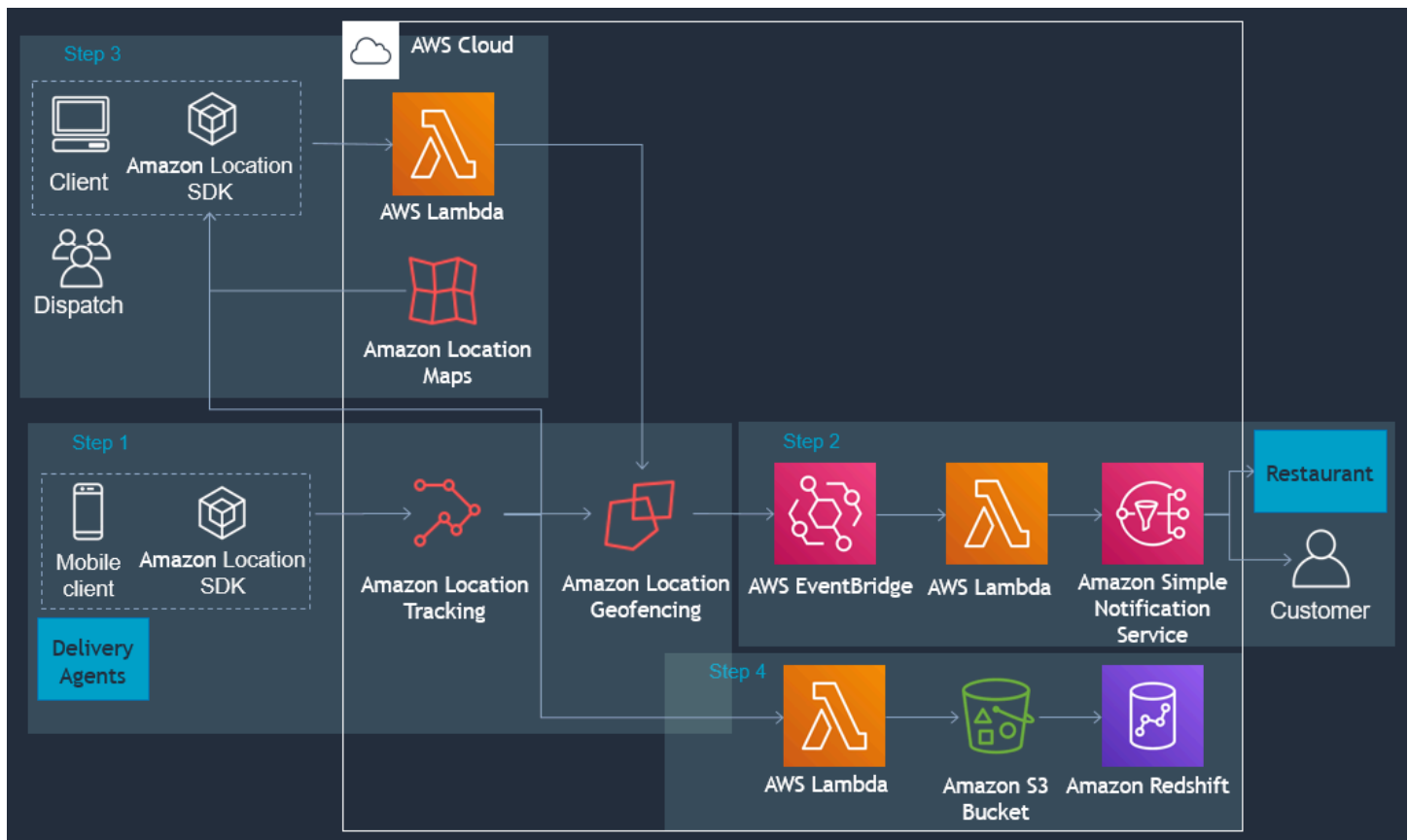
2. EventBridge Konfigurasi Amazon untuk mengirim pemberitahuan atau memulai proses. Untuk informasi selengkapnya, lihat [the section called “Bereaksi terhadap peristiwa dengan EventBridge”](#).
3. Tampilkan aset yang dilacak dan geofences aktif Anda di peta. Untuk informasi selengkapnya, lihat [Menggunakan peta](#).
4. Simpan data lokasi dalam penyimpanan jangka panjang untuk analisis lebih lanjut.
5. Setelah Anda membangun aplikasi Anda, Anda dapat menggunakan Amazon CloudWatch dan AWS CloudTrail mengelola aplikasi Anda. Untuk informasi selengkapnya, lihat [the section called “Pemantauan CloudWatch dengan”](#) dan [the section called “Menggunakan CloudTrail dengan Lokasi Amazon”](#).

## Aplikasi pengiriman

Berikut ini adalah ilustrasi arsitektur aplikasi pengiriman menggunakan Amazon Location.

Dengan arsitektur ini, Anda dapat:

- Memulai acara berdasarkan kedekatan agen pengiriman sehingga pickup siap tepat waktu dan pelanggan dapat diberitahu ketika pengiriman mereka tiba.
- Tampilkan lokasi pengemudi, serta lokasi penjemputan dan pengantaran dalam waktu dekat di peta untuk menunjukkan gambaran besar kepada tim pengiriman.
- Simpan lokasi agen pengiriman sehingga Anda dapat menindaklanjutinya di aplikasi backend Anda atau menganalisisnya dari waktu ke waktu.
- Menganalisis riwayat lokasi untuk mengidentifikasi tren dan peluang untuk optimasi.



Berikut ini adalah ikhtisar langkah-langkah yang diperlukan untuk membangun aplikasi pengiriman:

1. Buat koleksi geofence Anda dan tautkan perangkat yang dilacak ke koleksi. Untuk informasi lebih lanjut lihat [the section called “Geofencing dan pelacakan”](#).
2. Buat AWS Lambda fungsi untuk menambahkan dan menghapus geofences secara otomatis saat pesanan Anda dipesan.
3. Konfigurasi Amazon EventBridge untuk mengirim notifikasi atau memulai proses. Untuk informasi selengkapnya, lihat [the section called “Bereaksi terhadap peristiwa dengan EventBridge”](#).
4. Tampilkan aset yang dilacak dan geofences aktif di peta. Untuk informasi selengkapnya, lihat [Menggunakan peta](#).
5. Simpan data lokasi ke penyimpanan jangka panjang untuk analisis lebih lanjut.
6. Setelah Anda membangun aplikasi Anda, Anda dapat menggunakan Amazon CloudWatch dan AWS CloudTrail mengelola aplikasi Anda. Untuk informasi selengkapnya, lihat [the section called “Pemantauan CloudWatch dengan”](#) dan [the section called “Menggunakan CloudTrail dengan Lokasi Amazon”](#).

## Apa itu penyedia data?

Gunakan Amazon Location Service untuk mengakses sumber daya geolokasi dari beberapa penyedia data melalui AWS akun Anda tanpa memerlukan kontrak atau integrasi pihak ketiga. Ini dapat membantu Anda fokus membangun aplikasi Anda, tanpa harus mengelola akun, kredensial, lisensi, dan penagihan pihak ketiga.

Layanan Lokasi Amazon berikut menggunakan penyedia data.

- Peta — Pilih gaya dari penyedia peta yang berbeda saat Anda [membuat sumber daya peta](#). Anda dapat menggunakan sumber daya peta untuk membangun peta interaktif untuk memvisualisasikan data.
- Tempat — Pilih penyedia data saat Anda [membuat sumber daya indeks tempat](#) untuk mendukung kueri geocoding, reverse geocoding, dan pencarian.
- Rute — Pilih penyedia data untuk mendukung kueri perhitungan rute di berbagai geografi dan aplikasi saat Anda [membuat](#) sumber daya kalkulator rute. Dengan penyedia data pilihan Anda, Amazon Location Service memungkinkan Anda menghitung rute berdasarkan data jaringan up-to-date jalan, data lalu lintas langsung, penutupan terencana, dan pola lalu lintas historis.

Setiap penyedia mengumpulkan dan mengkurasi data mereka menggunakan cara yang berbeda. Mereka mungkin juga memiliki keahlian yang berbeda-beda di berbagai wilayah di dunia. Bagian ini memberikan rincian tentang penyedia data kami. Anda dapat memilih penyedia data berdasarkan preferensi Anda.

Pastikan Anda membaca syarat ketentuan saat menggunakan penyedia data Amazon Location Service. Untuk informasi selengkapnya, lihat [Ketentuan Layanan AWS](#). Lihat juga [the section called “Privasi data”](#) bagian untuk informasi selengkapnya tentang cara Lokasi Amazon melindungi privasi Anda.

## Cakupan dan fitur penyedia data

Tabel berikut menunjukkan cakupan dan fitur pada tingkat tinggi untuk setiap penyedia data.

Penyedia data	Cakupan geografis	Cakupan fitur	Wilayah AWS
Esri	Global	Peta, Tempat, Rute	<a href="#">Semua Wilayah</a> di mana Lokasi Amazon tersedia.
Grab	<a href="#">Asia Tenggara</a>	Peta, Tempat, Rute	Asia Pasifik (Singapura), ap-southeast-1, hanya.
KEMARI	Global	Peta, Tempat, Rute	<a href="#">Semua Wilayah</a> di mana Lokasi Amazon tersedia.
Buka Data	Global	Peta	<a href="#">Semua Wilayah</a> di mana Lokasi Amazon tersedia.

Untuk informasi selengkapnya tentang fitur spesifik masing-masing penyedia data, lihat [Fitur oleh penyedia data](#).

Setiap penyedia data mengumpulkan dan menghasilkan data dengan cara yang berbeda. Anda dapat mempelajari lebih lanjut tentang area cakupan mereka dalam topik berikut:

- [Cakupan: Esri](#)
- [Cakupan: Grab](#)
- [Cakupan: DI SINI](#)
- [Cakupan: Data Terbuka](#)

Jika Anda mengalami masalah dengan data dan ingin melaporkan kesalahan ke penyedia data, lihat topik berikut:

- [Kesalahan pelaporan ke Esri](#)
- [Pelaporan kesalahan untuk GrabMaps data](#)
- [Kesalahan melaporkan ke SINI](#)
- [Melaporkan kesalahan dan berkontribusi pada Data Terbuka](#)

## Gaya peta

Setiap penyedia data menyediakan satu set gaya peta untuk merender data peta yang mereka berikan. Misalnya gaya mungkin termasuk citra satelit, atau mungkin dioptimalkan untuk menunjukkan jalan untuk navigasi. Anda dapat menemukan daftar dan contoh gaya untuk setiap penyedia dalam topik berikut.

- [Gaya peta Esri](#)
- [Ambil gaya peta](#)
- [HERE gaya peta](#)
- [Buka gaya peta Data](#)

## Informasi lebih lanjut tentang setiap penyedia data

Tautan berikut memberikan informasi lebih lanjut tentang setiap penyedia data.

- [Esri](#)
- [GrabMaps](#)
- [HERE Teknologi](#)
- [Buka Data](#)

## Esri

Amazon Location Service menggunakan layanan lokasi Esri untuk membantu AWS pelanggan menggunakan peta, geocode, dan menghitung rute secara efektif. Layanan lokasi Esri dibangun dengan data berkualitas tinggi, otoritatif, dan ready-to-use lokasi, dikuratori oleh tim ahli kartografer, ahli geografi, dan demografi.

Untuk informasi kapabilitas tambahan, lihat [Esri](#) on Amazon Location Service penyedia data.

### Topik

- [Gaya peta Esri](#)
- [Cakupan: Esri](#)
- [Ketentuan penggunaan dan atribusi data: Esri](#)
- [Kesalahan pelaporan ke Esri](#)

## Gaya peta Esri

Amazon Location Service mendukung gaya peta Esri berikut saat [membuat sumber daya peta](#).

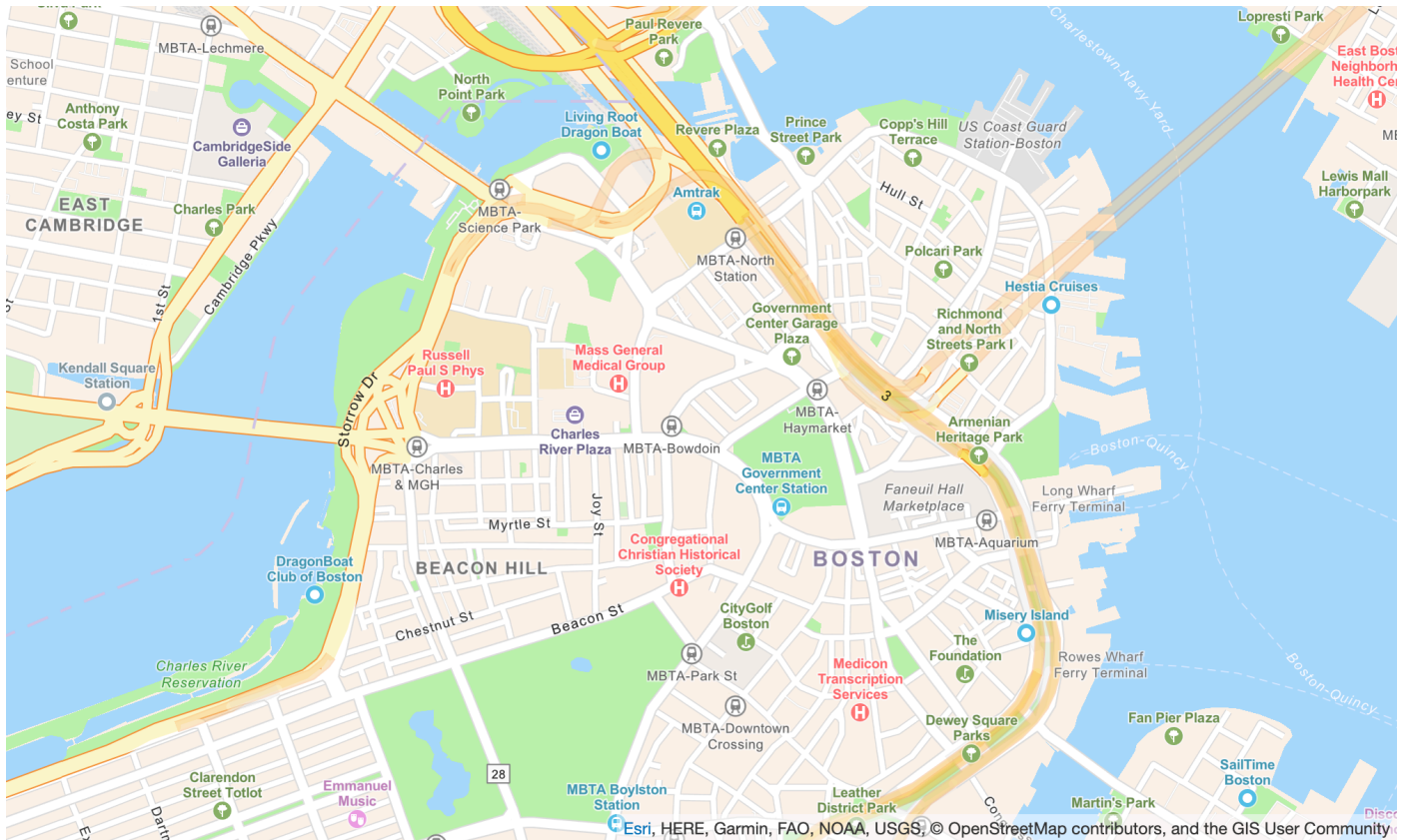
### Note

Gaya peta Esri yang tidak tercantum di bagian ini tidak didukung.

Gaya vektor Esri mendukung alternatif [Pandangan politik](#).

## Esri Navigation

### Navigasi Esri



Nama gaya peta: `VectorEsriNavigation`

Peta ini menyediakan peta dasar terperinci untuk dunia yang dilambangkan dengan gaya peta navigasi khusus yang dirancang untuk digunakan pada siang hari di perangkat seluler.

Peta jalan yang komprehensif ini mencakup jalan raya, jalan utama, jalan kecil, kereta api, fitur air, kota, taman, landmark, jejak bangunan, dan batas administratif. Lapisan ubin vektor

di peta ini dibangun menggunakan sumber data yang sama yang digunakan untuk Peta Jalan Dunia dan peta dasar Esri lainnya. Aktifkan POI layer dengan mengaturnya [CustomLayers](#) untuk memanfaatkan data tempat tambahan.

Untuk informasi lebih lanjut, lihat [Esri World Navigation](#) di situs web Esri.

#### Note

VectorEsriNavigationPeta yang digambarkan di atas memiliki POI lapisan yang diaktifkan.

## Font

Lokasi Amazon menyajikan font menggunakan [GetMapGlyphs](#). Berikut ini adalah tumpukan font yang tersedia untuk peta ini:

- Arial Italic
- Arial Reguler
- Arial Berani
- Arial Unicode MS Tebal
- Arial Unicode MS Reguler

## Esri Imagery

### Citra Esri

Nama gaya peta: `RasterEsriImagery`

Peta ini menyediakan citra satelit dan udara satu meter atau lebih baik di banyak bagian dunia dan citra satelit resolusi lebih rendah di seluruh dunia.

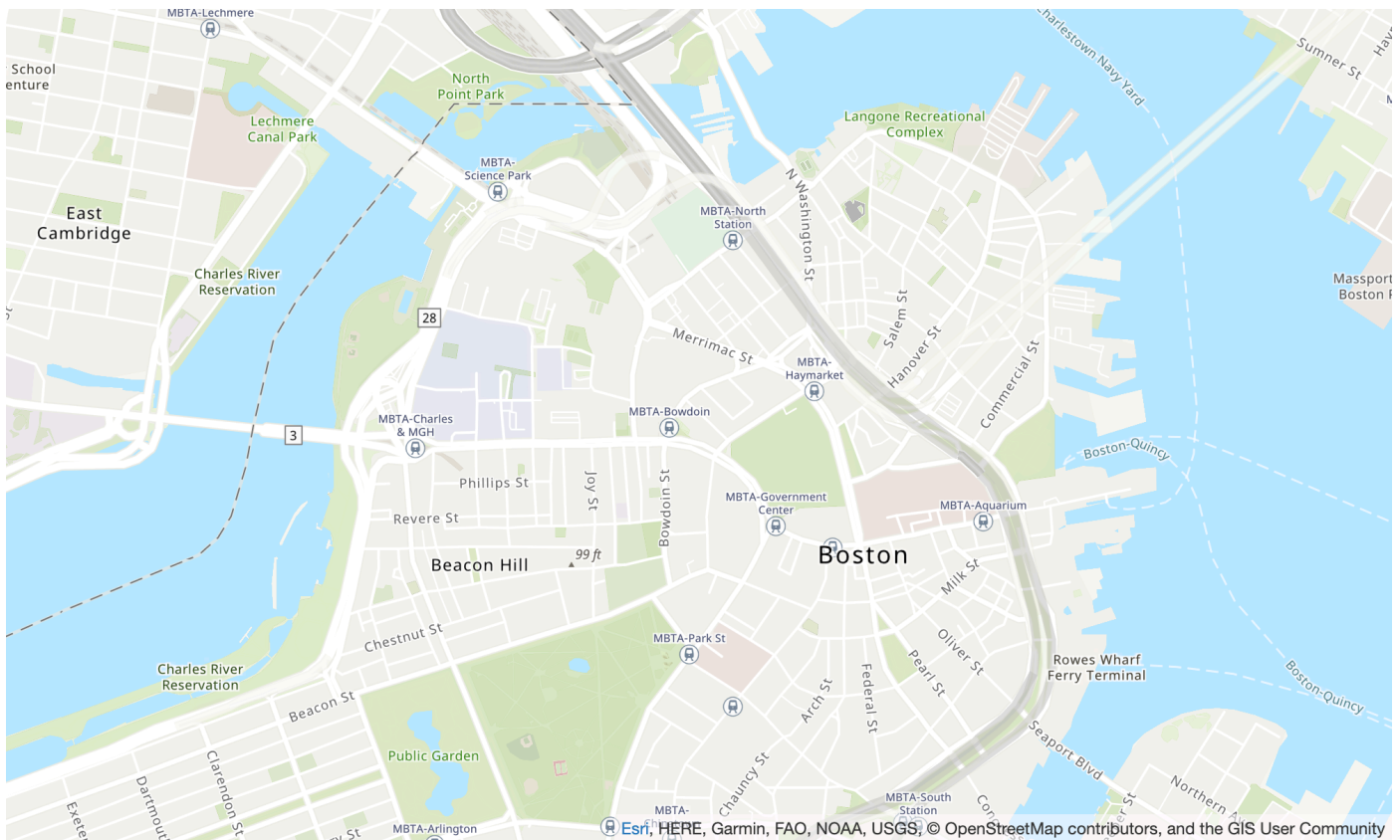
Peta ini mencakup citra 15m pada skala kecil dan menengah (~ 1:591 M hingga ~ 1:72 k) dan 2.5m SPOT Imagery (~ 1:288 k hingga ~ 1:72 k) untuk dunia. Peta ini menampilkan citra resolusi 0,5 m di benua Amerika Serikat dan sebagian Eropa Barat dari Maxar. Peta ini menampilkan citra submeter Maxar tambahan di banyak bagian dunia. Di belahan dunia lain, Komunitas Pengguna GIS telah menyumbangkan citra pada resolusi yang berbeda. Di komunitas tertentu, citra resolusi sangat tinggi (hingga 0,03m) tersedia hingga skala ~ 1:280.



Untuk informasi lebih lanjut, lihat [Esri World Imagery](#) di situs web Esri.

## Esri Light

### Cahaya Esri



Nama gaya peta: `VectorEsriTopographic`

Ini memberikan peta dasar terperinci untuk dunia yang dilambangkan dengan gaya peta Esri klasik. Ini termasuk jalan raya, jalan utama, jalan kecil, kereta api, fitur air, kota, taman, landmark, jejak bangunan, dan batas-batas administrasi.

Peta dasar ini disusun dari berbagai sumber otoritatif dari beberapa penyedia data, termasuk Survei Geologi AS (USGS), Badan Perlindungan Lingkungan AS (EPA), Layanan Taman Nasional AS (NPS), Organisasi Pangan dan Pertanian Perserikatan Bangsa-Bangsa (FAO), Departemen Sumber Daya Alam Kanada (NRCAN), HERE, dan Esri. Data untuk area tertentu bersumber dari OpenStreetMap kontributor. Selain itu, data disediakan oleh komunitas GIS.

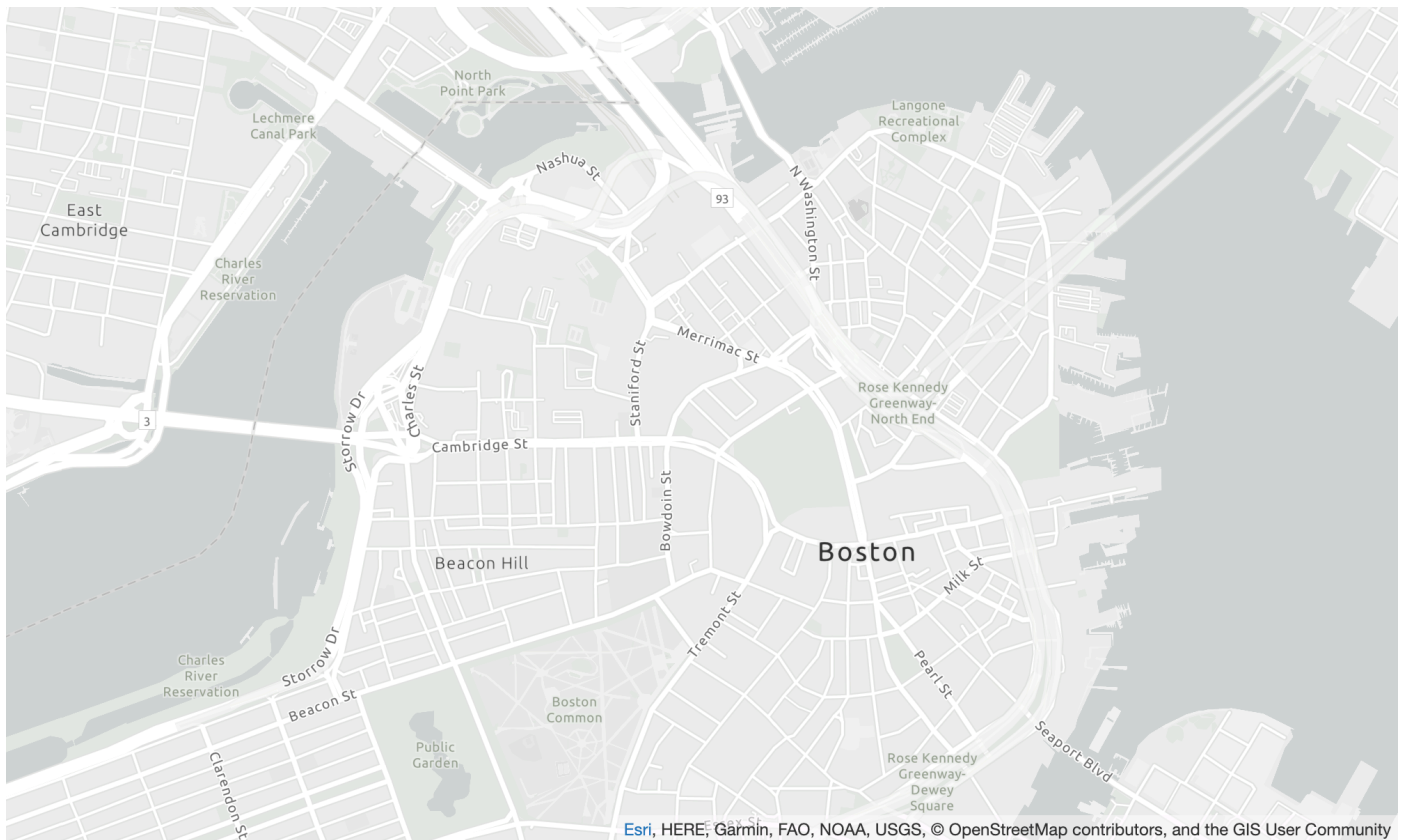
### Font

Lokasi Amazon menyajikan font menggunakan [GetMapGlyphs](#). Berikut ini adalah tumpukan font yang tersedia untuk peta ini:

- Noto Sans Italic
- Noto Sans Regular
- Noto Sans Bold
- Noto Serif Regular
- Roboto Condensed Light Italic

## Esri Light Gray Canvas

### Kanvas Abu-abu Muda Esri



### Nama gaya peta: VectorEsriLightGrayCanvas

Peta ini menyediakan peta dasar terperinci untuk dunia yang dilambangkan dengan gaya latar belakang netral abu-abu muda dengan warna, label, dan fitur minimal yang dirancang untuk menarik perhatian ke konten tematik Anda.

Lapisan ubin vektor ini dibangun menggunakan sumber data yang sama yang digunakan untuk Light Grey Canvas dan peta dasar Esri lainnya. Peta ini mencakup jalan raya, jalan utama, jalan kecil, kereta api, fitur air, kota, taman, landmark, jejak bangunan, dan batas-batas administratif.

Untuk informasi lebih lanjut, lihat [Esri Light Grey Canvas](#) di situs web Esri.

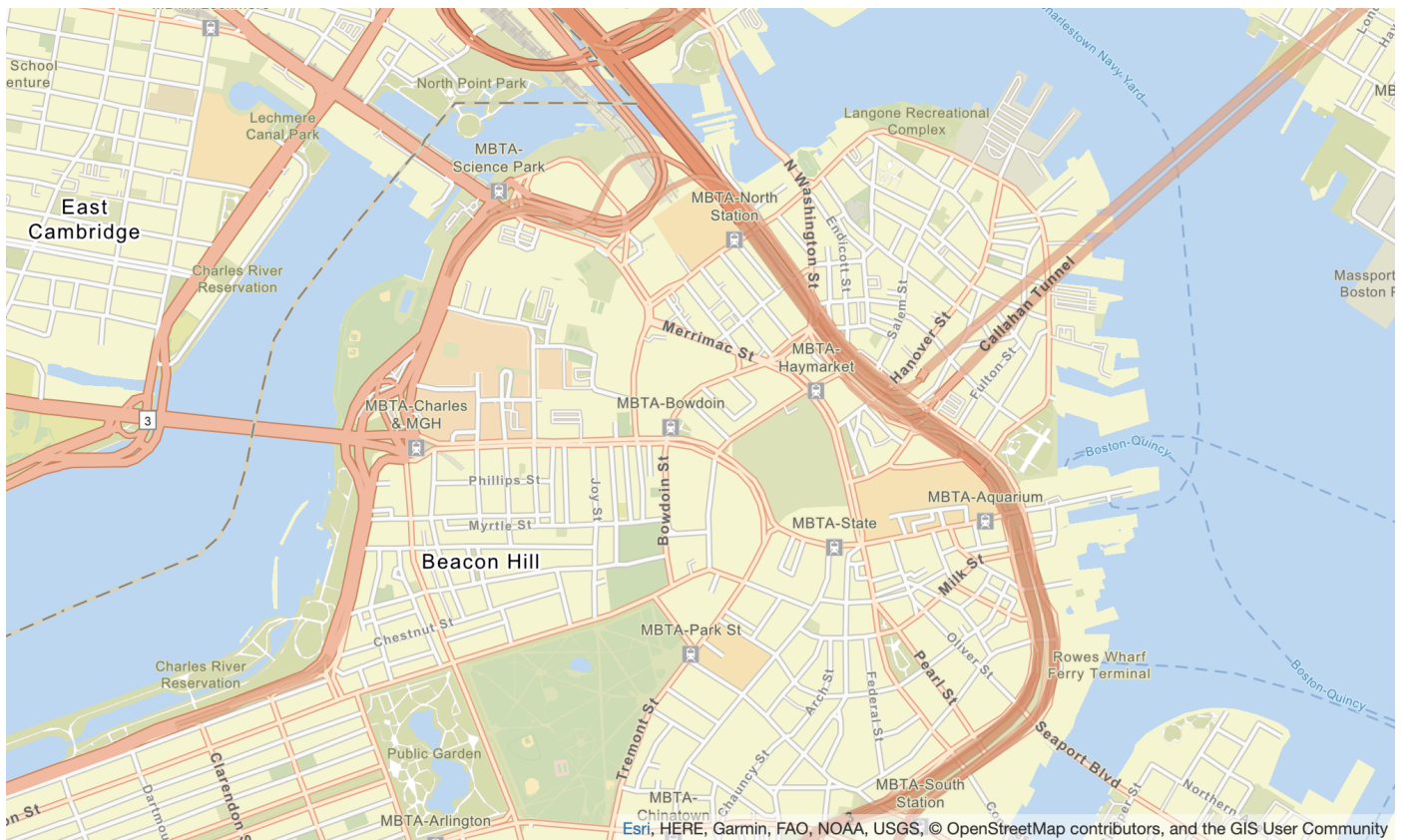
## Font

Lokasi Amazon menyajikan font menggunakan [GetMapGlyphs](#). Berikut ini adalah tumpukan font yang tersedia untuk peta ini:

- Ubuntu Miring
- Ubuntu Regular
- Cahaya Ubuntu
- Ubuntu Tebal

## Esri Street Map

### Peta Jalan Esri



Nama gaya peta: `VectorEsriStreets`

Peta ini menyediakan peta dasar terperinci untuk dunia yang dilambangkan dengan gaya peta navigasi khusus yang dirancang untuk digunakan pada siang hari di perangkat seluler.

Peta jalan yang komprehensif ini mencakup jalan raya, jalan utama, jalan kecil, kereta api, fitur air, kota, taman, landmark, jejak bangunan, dan batas administratif. Ini juga mencakup serangkaian tempat yang lebih kaya, seperti toko, layanan, restoran, atraksi, dan tempat menarik lainnya. Lapisan ubin vektor di peta ini dibangun menggunakan sumber data yang sama yang digunakan untuk Peta Jalan Dunia dan peta dasar Esri lainnya.

Untuk informasi lebih lanjut, lihat [Esri World Street](#) di situs web Esri.

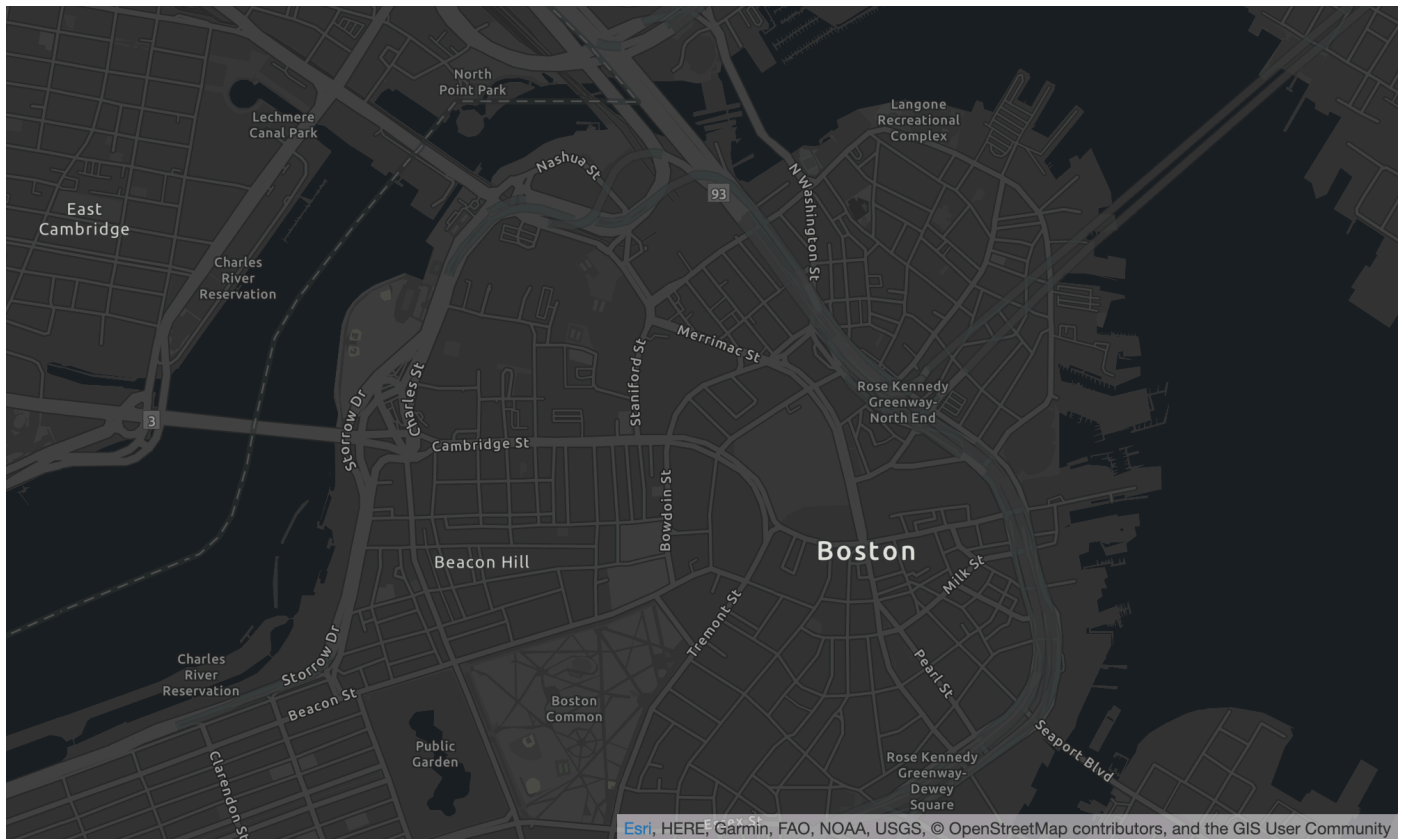
## Font

Lokasi Amazon menyajikan font menggunakan [GetMapGlyphs](#). Berikut ini adalah tumpukan font yang tersedia untuk peta ini:

- Arial Italic
- Arial Regular
- Arial Berani
- Arial Unicode MS Tebal
- Arial Unicode MS Regular

## Esri Dark Gray Canvas

Esri Canvas Abu-abu Gelap



Nama gaya peta: `VectorEsriDarkGrayCanvas`

Peta ini menyediakan peta dasar vektor terperinci untuk dunia yang dilambangkan dengan gaya latar belakang netral abu-abu gelap dengan warna, label, dan fitur minimal yang dirancang untuk menarik perhatian ke konten tematik Anda.

Peta ini mencakup jalan raya, jalan utama, jalan kecil, kereta api, fitur air, kota, taman, landmark, jejak bangunan, dan batas administrasi. Lapisan ubin vektor di peta ini dibangun menggunakan sumber data yang sama yang digunakan untuk peta raster Kanvas Abu-abu Gelap dan peta dasar Esri lainnya.

Untuk informasi lebih lanjut, lihat [Esri Dark Grey Canvas](#) di situs web Esri.

## Font

Lokasi Amazon menyajikan font menggunakan [GetMapGlyphs](#). Berikut ini adalah tumpukan font yang tersedia untuk peta ini:

- Ubuntu Medium Italic
- Ubuntu Medium

- Ubuntu Miring
- Ubuntu Reguler
- Ubuntu Tebal

## Cakupan: Esri

[Anda dapat menggunakan Esri sebagai penyedia data untuk mendukung kueri untuk geocoding, geocoding terbalik, dan pencarian saat Anda membuat sumber daya indeks tempat, atau untuk mendukung kueri untuk menghitung rute saat Anda membuat sumber daya kalkulator rute.](#)

Esri menyediakan berbagai tingkat kualitas data di berbagai wilayah di dunia. Untuk informasi tambahan tentang cakupan di wilayah yang Anda minati, lihat:

- [Detail Esri tentang cakupan geocoding](#)
- [Detail Esri tentang jaringan jalan dan cakupan lalu lintas](#)

## Ketentuan penggunaan dan atribusi data: Esri

Sebelum Anda menggunakan data Esri, pastikan Anda dapat mematuhi semua persyaratan hukum yang berlaku, termasuk persyaratan lisensi yang berlaku untuk Esri dan AWS.

Untuk informasi selengkapnya tentang AWS persyaratan, lihat [Ketentuan Layanan AWS](#).

Untuk informasi tentang pedoman atribusi Esri, lihat [Atribusi Data dan Ketentuan Penggunaan Esri](#).

## Kesalahan pelaporan ke Esri

Jika Anda mengalami masalah dengan data dan ingin melaporkan kesalahan dan perbedaan ke Esri, ikuti artikel dukungan teknis Esri untuk [Cara: Memberikan umpan balik tentang peta](#) dasar dan geocoding.

## GrabMaps

Grab adalah organisasi pengiriman terbesar di Asia Tenggara, dengan jutaan mitra pengemudi dan pelanggan. Anak perusahaan mereka, [GrabMaps](#), membuat data up-to-date pemetaan di negara/wilayah tersebut untuk digunakan sendiri, dan lainnya. Amazon Location Service menggunakan GrabMaps 'layanan lokasi untuk membantu AWS pelanggan menggunakan peta, geocode, dan

menghitung rute secara efektif. GrabMapsLayanan lokasi dibangun untuk menyediakan data berkualitas tinggi, otoritatif, dan ready-to-use lokasi, khusus untuk negara-negara Asia Tenggara.

Untuk informasi tentang kemampuan tambahan, lihat [GrabMaps](#) di penyedia data Amazon Location Service.

#### Important

Grab menyediakan peta hanya untuk Asia Tenggara, dan hanya tersedia di Wilayah Asia Pasifik (Singapura) (ap-southeast-1). Untuk informasi selengkapnya, lihat [Negara/wilayah dan area yang dicakup](#).

#### Topik

- [Ambil gaya peta](#)
- [Cakupan: Grab](#)
- [Negara/wilayah dan area yang dicakup](#)
- [Ketentuan penggunaan dan atribusi data: Grab](#)
- [Pelaporan kesalahan untuk GrabMaps data](#)

#### Ambil gaya peta

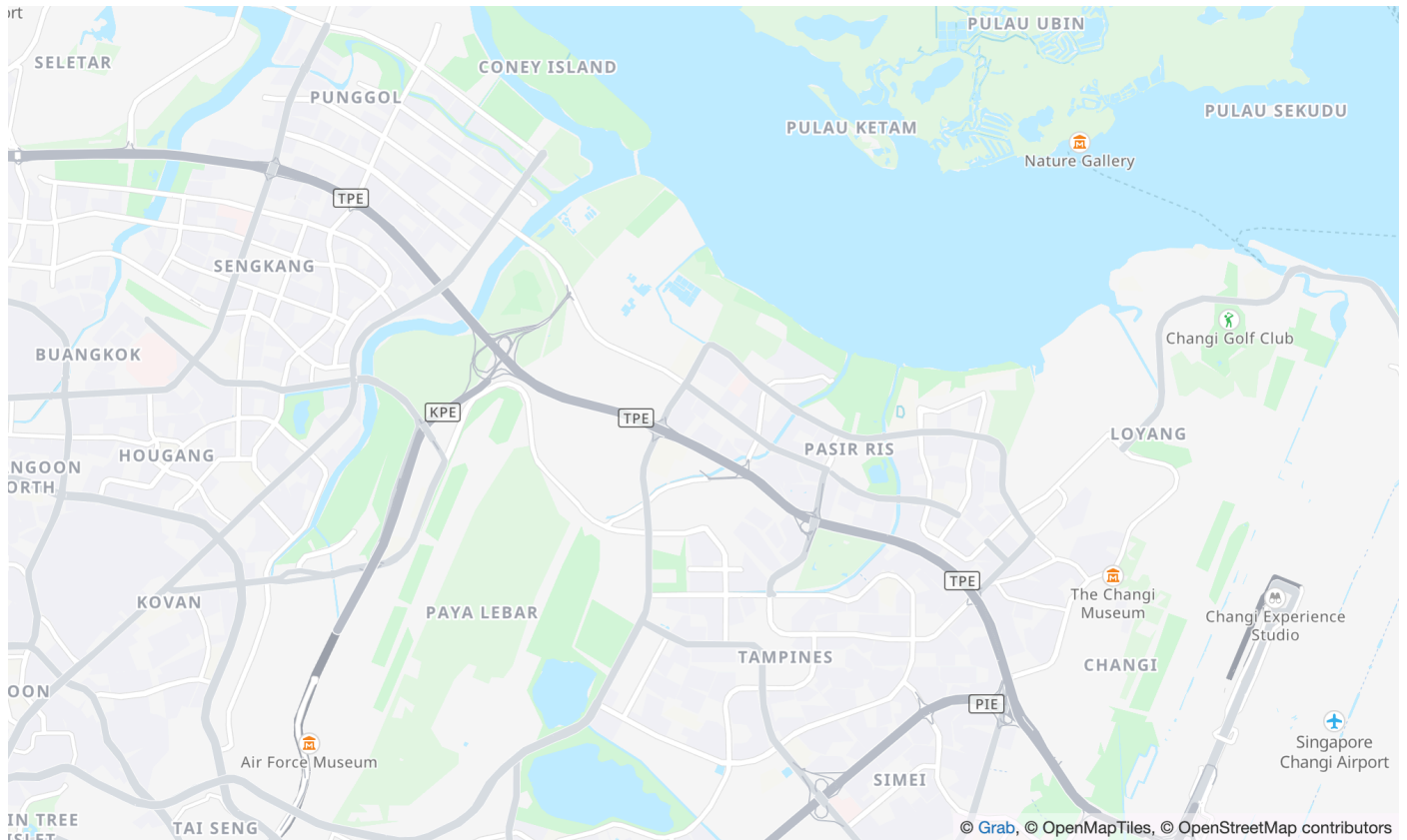
Amazon Location Service mendukung gaya peta Grab berikut saat [membuat sumber daya peta](#):

#### Note

Gaya peta ambil yang tidak tercantum di bagian ini saat ini tidak didukung.

#### Grab Standard Light Map

Peta Cahaya Standar Grab



Nama gaya peta: `VectorGrabStandardLight`

Peta dasar standar Grab dengan pewarnaan penggunaan lahan yang terperinci, nama area, jalan, landmark, dan tempat menarik yang mencakup Asia Tenggara.

Font

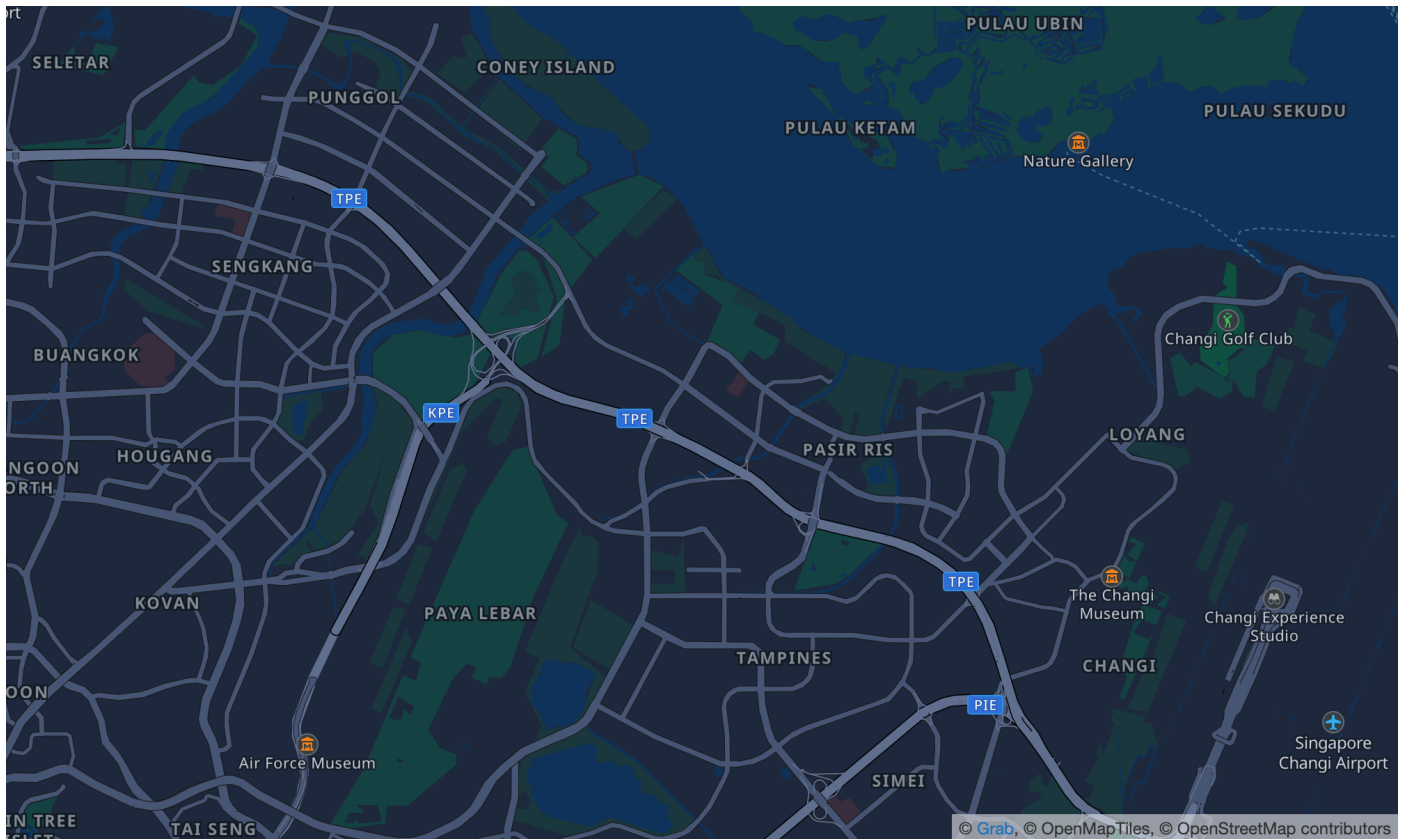
Lokasi Amazon menyajikan font menggunakan [GetMapGlyphs](#). Berikut ini adalah tumpukan font yang tersedia untuk peta ini:

- Noto Sans Regular
- Noto Sans Medium
- Noto Sans Bold

Grab Standard Dark Map

Peta Gelap Standar Grab





Nama gaya peta: `VectorGrabStandardDark`

Variasi gelap dari basemap standar Grab, dengan pewarnaan penggunaan lahan yang mendetail, nama area, jalan, landmark, dan tempat menarik yang mencakup Asia Tenggara.

Font

Lokasi Amazon menyajikan font menggunakan [GetMapGlyphs](#). Berikut ini adalah tumpukan font yang tersedia untuk peta ini:

- Noto Sans Regular
- Noto Sans Medium
- Noto Sans Bold

Cakupan: Grab

Anda dapat menggunakan Grab sebagai penyedia data untuk mendukung kueri geocoding, reverse geocoding, dan pencarian saat Anda [membuat sumber daya indeks tempat](#), atau untuk mendukung kueri untuk menghitung rute saat Anda [membuat](#) sumber daya kalkulator rute.

## Negara/wilayah dan area yang dicakup

Grab menyediakan peta hanya untuk Asia Tenggara, dan hanya tersedia di Asia Pasifik (Singapura) Region (ap-southeast-1).

Grab menyediakan data terperinci untuk negara/wilayah berikut:

- Malaysia
- Filipina
- Thailand
- Singapura
- Vietnam
- Indonesia
- Myanmar
- Kamboja

### Note

Di luar area tersebut, sumber daya Amazon Location Service yang dibuat dengan Grab sebagai penyedia data tidak akan memberikan hasil apa pun. Ini termasuk hasil pencarian atau rute.

Peta-peta dari Grab berada dalam batas-batas berikut:

- Selatan — Lintang -21.943045533438166
- Barat — Bujur 90.0
- Utara — Latitude 31.952162238024968
- Timur — Bujur 146.25

Untuk tingkat zoom 1—4, Grab menyertakan cakupan global. Untuk tingkat zoom 5 dan di bawahnya, ubin peta hanya disediakan di dalam kotak terbatas ini.

**Note**

Di luar kotak terbatas ini, sumber daya peta Amazon Location Service yang dibuat dengan Grab sebagai penyedia data tidak akan mengembalikan ubin peta. Untuk menghindari melihat 404 kesalahan dalam aplikasi Anda, Anda dapat membatasi peta dengan kotak pembatas, seperti yang dijelaskan dalam [Mengatur luasan untuk peta menggunakan MapLibre](#)

Dapatkan mode perjalanan perutean

Untuk routing, Grab menyediakan routing mobil dan motor untuk semua negara/wilayah yang terdaftar sebelumnya.

Grab tidak mendukung routing truk.

Untuk rute sepeda dan jalan kaki, Grab mendukung kota-kota berikut:

- Singapura
- Jakarta
- Manila
- Lembah Klang
- Bangkok
- Kota Ho Chi Minh
- Hanoi

**Ketentuan penggunaan dan atribusi data: Grab**

Saat menggunakan data Grab, Anda harus mematuhi semua persyaratan hukum yang berlaku, termasuk persyaratan lisensi yang berlaku untuk Grab dan AWS.

Untuk informasi selengkapnya tentang AWS persyaratan, lihat [Ketentuan Layanan AWS](#).

Untuk informasi tentang GrabMaps 'pedoman atribusi, lihat Bagian 9.23 [Atribusi Data dan Ketentuan Penggunaan](#) Grab.

## Pelaporan kesalahan untuk GrabMaps data

Jika Anda mengalami masalah dengan data dari GrabMaps, dan ingin melaporkan kesalahan atau perbedaan, [hubungi dukungan AWS teknis](#).

## HERE Teknologi

Amazon Location Service menggunakan layanan lokasi HERE Technologies untuk membantu AWS pelanggan menggunakan peta, geocode, dan menghitung rute secara efektif. Data lokasi HERE menawarkan platform lokasi-sentris yang terbuka, aman, dan pribadi. Dengan memilih data lokasi HERE, Anda memilih data yang akurat, segar, dan kuat yang diterapkan secara native di AWS Cloud.

Untuk informasi kapabilitas tambahan, lihat [DI SINI](#) di Amazon Location Service penyedia data.

### Topik

- [HERE gaya peta](#)
- [Cakupan: DI SINI](#)
- [Ketentuan penggunaan dan atribusi data: DI SINI](#)
- [Kesalahan melaporkan ke SINI](#)

## HERE gaya peta

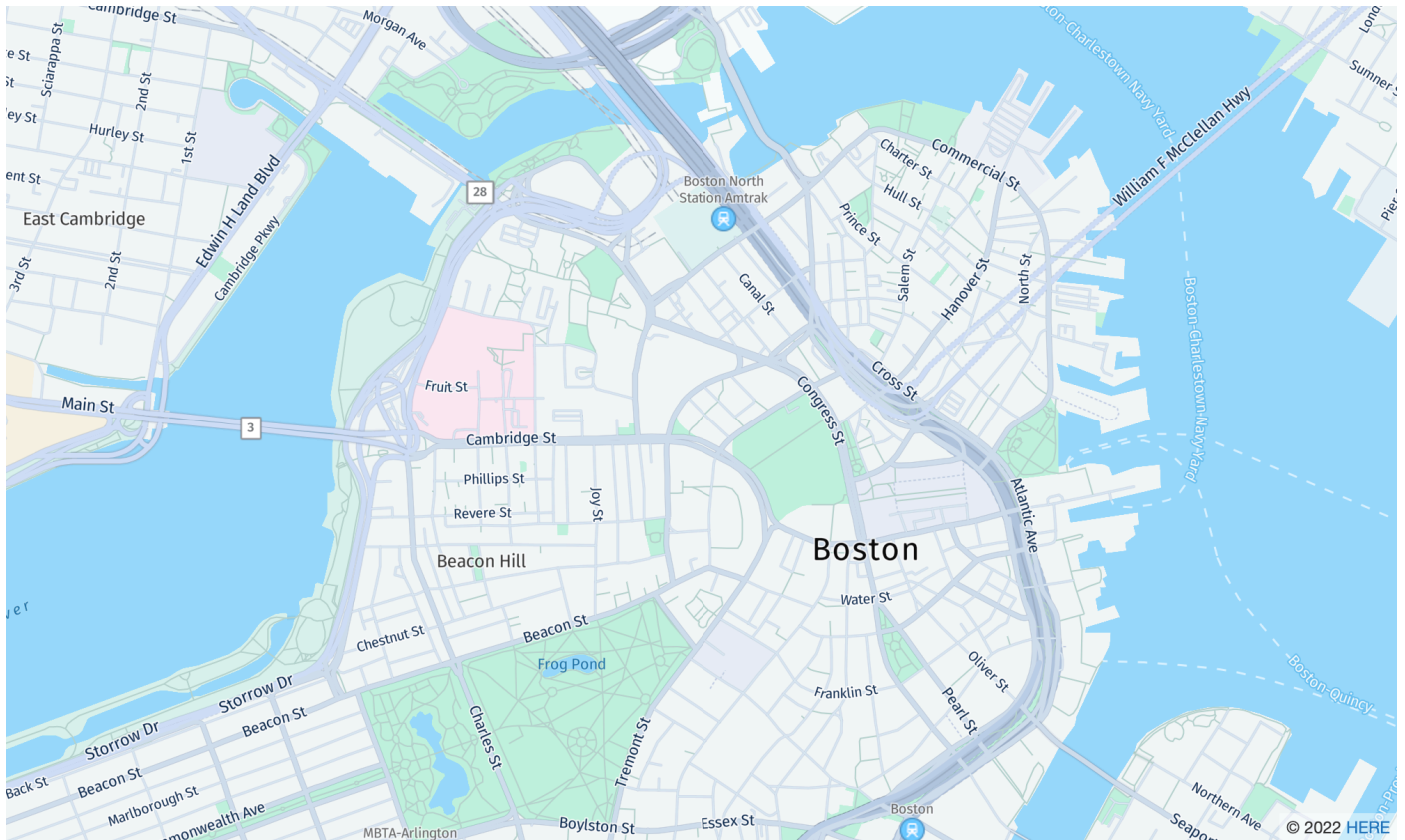
Amazon Location Service mendukung gaya peta HERE berikut saat [membuat sumber daya peta](#):

### Note

HERE gaya peta yang tidak tercantum dalam bagian ini saat ini tidak didukung.

## HERE Explore

DISINI Jelajahi



Nama gaya peta: VectorHereExplore

DISINI Jelajahi

Peta dasar dunia yang terperinci dan netral. Peta jalan mencakup jalan raya, jalan utama, jalan kecil, kereta api, fitur air, kota, taman, landmark, jejak bangunan, dan batas-batas administratif. Termasuk peta Jepang yang dirancang sepenuhnya.

Font

Lokasi Amazon menyajikan font menggunakan [GetMapGlyphs](#). Berikut ini adalah tumpukan font yang tersedia untuk peta ini:

- Fira GO Miring
- Fira GO Reguler
- Fira GO Bold
- Lampu Noto Sans CJK JP
- Noto Sans CJK JP Reguler
- Noto Sans CJK JP Berani

## HERE Imagery

### DI SINI Citra



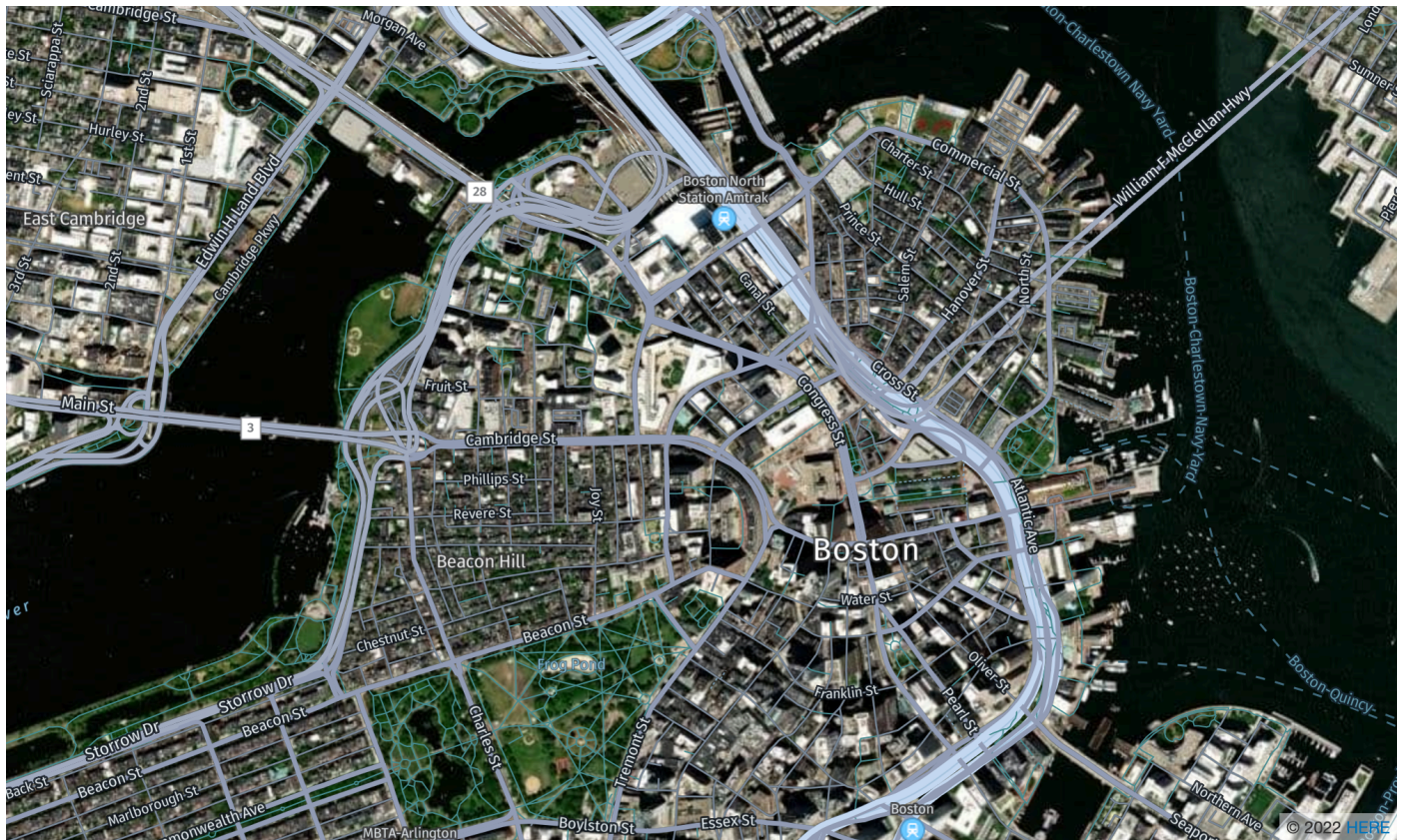
Nama gaya peta: `RasterHereExploreSatellite`

### DI SINI Citra

HERE Imagery menyediakan citra satelit resolusi tinggi dengan cakupan global.

### HERE Hybrid

### DI SINI Hibrida



Nama gaya peta: HybridHereExploreSatellite

DI SINI Hibrida

HERE Hybrid style menampilkan jaringan jalan, nama jalan, dan label kota melalui citra satelit. Gaya ini melapisi dua ubin peta: citra satelit (ubin raster) di latar belakang dan jaringan jalan dan label (ubin vektor) di atas. Gaya ini akan secara otomatis mengambil ubin raster dan vektor yang diperlukan untuk merender peta.

#### Note

Gaya hybrid menggunakan ubin vektor dan raster saat merender peta yang Anda lihat. Ini berarti bahwa lebih banyak ubin diambil daripada saat menggunakan ubin vektor atau raster saja. Biaya Anda akan mencakup semua ubin yang diambil.

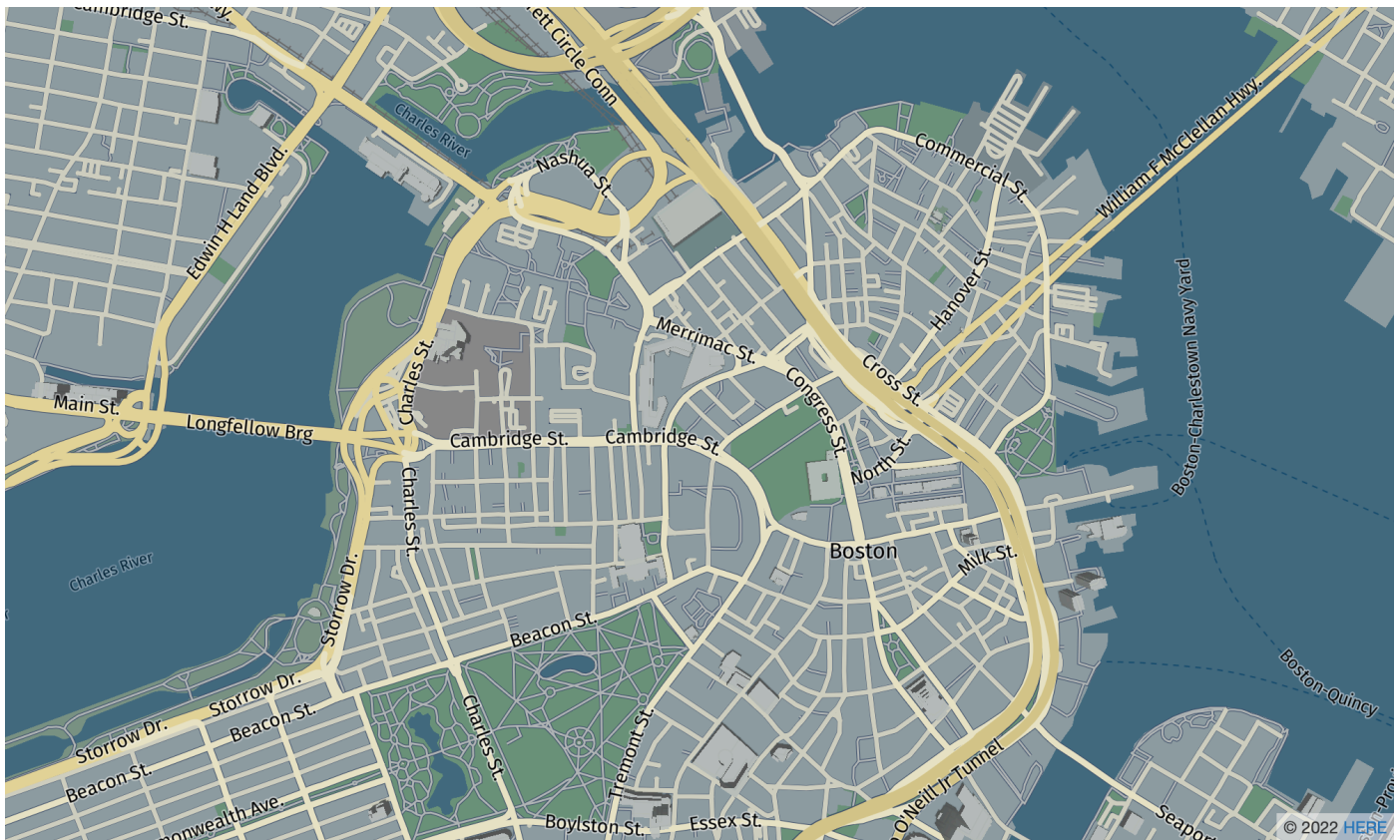
Font

Lokasi Amazon menyajikan font menggunakan [GetMapGlyphs](#). Berikut ini adalah tumpukan font yang tersedia untuk peta ini:

- Fira GO Miring
- Fira GO Regular
- Fira GO Bold
- Lampu Noto Sans CJK JP
- Noto Sans CJK JP Regular
- Noto Sans CJK JP Berani

## HERE Contrast (Berlin)

## HERE Kontras (Berlin)



Nama gaya peta: VectorHereContrast

## HERE Kontras (Berlin)

Peta dasar rinci dunia yang memadukan rendering 3D dan 2D. Peta jalan kontras tinggi mencakup jalan raya, jalan utama, jalan kecil, kereta api, fitur air, kota, taman, landmark, jejak bangunan, dan batas administratif.



## Font

Lokasi Amazon menyajikan font menggunakan [GetMapGlyphs](#). Berikut ini adalah tumpukan font yang tersedia untuk peta ini:

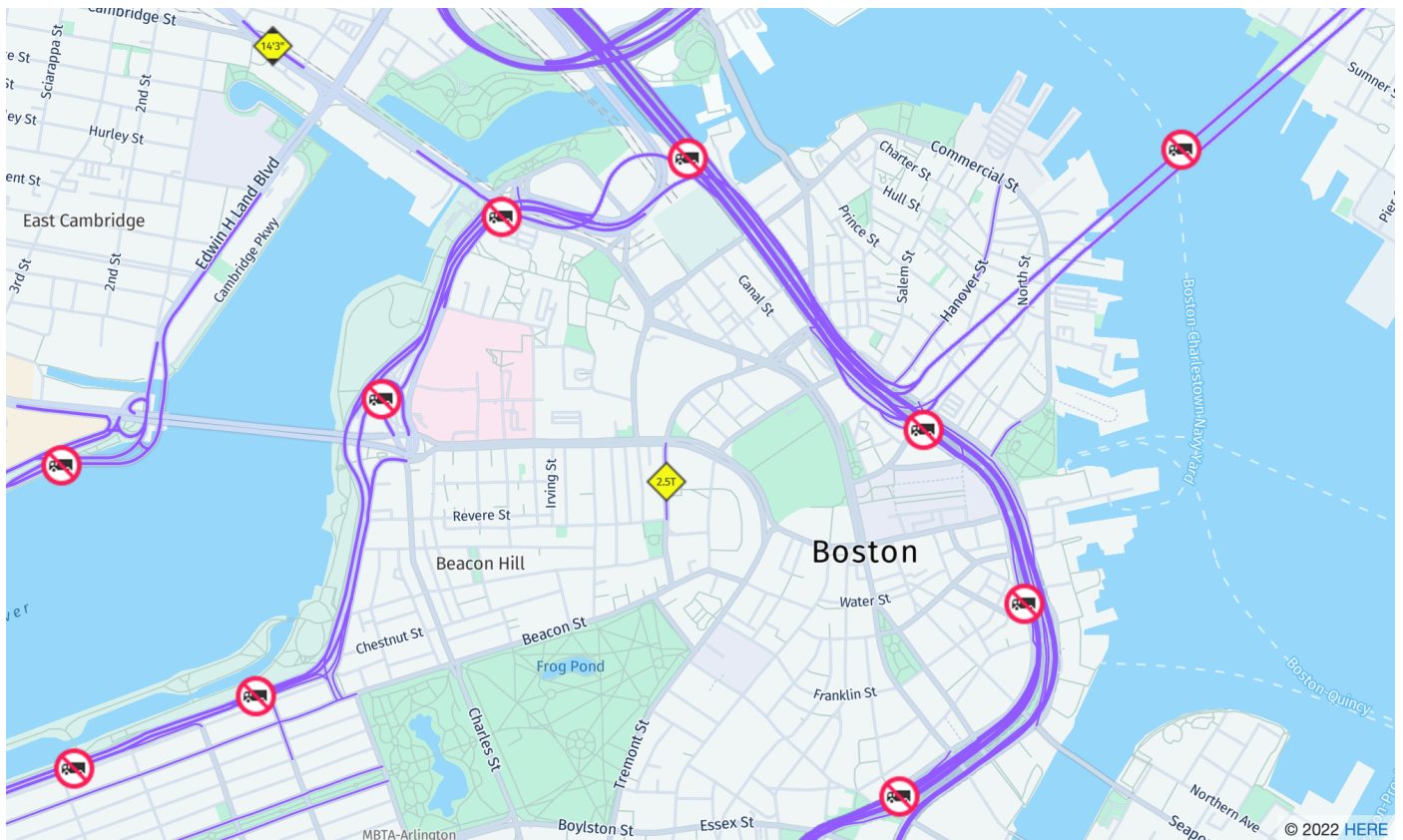
- Fira GO Regular
- Fira GO Bold

### Note

Gaya ini diganti namanya dari `VectorHereBerlin` (HERE peta Berlin). `VectorHereBerlin` tidak digunakan lagi, tetapi akan terus bekerja di aplikasi yang menggunakannya.

## HERE Explore Truck

### DI SINI Jelajahi Truk



Nama gaya peta: `VectorHereExploreTruck`

## DI SINI Jelajahi Truk

Peta dasar dunia yang terperinci dan netral. Peta jalan dibangun di atas gaya HERE Explore, dan menyoroti batasan dan atribut trek (termasuk lebar, tinggi, dan HAZMAT) dengan simbol dan ikon, untuk mendukung kasus penggunaan dalam transportasi dan logistik.

### Font

Lokasi Amazon menyajikan font menggunakan [GetMapGlyphs](#). Berikut ini adalah tumpukan font yang tersedia untuk peta ini:

- Fira GO Miring
- Fira GO Reguler
- Fira GO Bold
- Lampu Noto Sans CJK JP
- Noto Sans CJK JP Reguler
- Noto Sans CJK JP Berani

Untuk informasi tambahan tentang kualitas data peta di berbagai wilayah di dunia, lihat [DISINI cakupan peta](#).

## Cakupan: DI SINI

Anda dapat menggunakan HERE sebagai penyedia data untuk mendukung kueri geocoding, reverse geocoding, dan pencarian saat Anda [membuat sumber daya indeks tempat](#), atau untuk mendukung kueri untuk menghitung rute saat Anda [membuat](#) sumber daya kalkulator rute.

HERE menyediakan berbagai tingkat kualitas data di berbagai wilayah di dunia. Untuk informasi tambahan tentang cakupan di wilayah yang Anda minati, lihat berikut ini:

- [HERE cakupan geocoding](#)
- [DISINI cakupan perutean mobil](#)
- [DISINI cakupan perutean truk](#)

## Ketentuan penggunaan dan atribusi data: DI SINI

Sebelum Anda menggunakan data HERE, pastikan Anda dapat mematuhi semua persyaratan hukum yang berlaku, termasuk ketentuan lisensi yang berlaku untuk HERE dan AWS. Karena keterbatasan

lisensi, Anda tidak boleh menggunakan HERE untuk menyimpan hasil geocoding untuk lokasi di Jepang.

Untuk informasi tentang AWS persyaratan, lihat [Ketentuan Layanan AWS](#).

Untuk informasi tambahan tentang pedoman atribusi HERE, lihat Bagian 2 dari [Ketentuan Pemasok HERE Technologies yang Berlaku untuk Lokasi dan Konten Lainnya](#).

## Kesalahan melaporkan ke SINI

Untuk melaporkan kesalahan peta dan perbedaan ke SINI, buka <https://www.here.com/contact> dan pilih Laporkan kesalahan peta.

## Buka Data

Amazon Location Service menyediakan akses ke data peta open source melalui penyedia Open Data. Open Data menyediakan peta dasar global yang dibangun dari [distribusi peta Daylight](#) [OpenStreetMap \(OSM\)](#), [Natural Earth](#), dan sumber data terbuka lainnya. Peta yang disediakan dirancang untuk mendukung berbagai aplikasi dan kasus penggunaan, termasuk logistik dan pengiriman, dan visualisasi data di lingkungan web dan seluler. Dengan lebih dari satu juta pembuat peta, komunitas OSM memperbarui ratusan ribu fitur per hari. Amazon Location Service secara teratur menggabungkan pengeditan ini.

Untuk informasi kapabilitas tambahan, lihat [Open Data](#) on Amazon Location Service penyedia data.

### Topik

- [Buka gaya peta Data](#)
- [Cakupan: Data Terbuka](#)
- [Ketentuan penggunaan dan atribusi data: Data Terbuka](#)
- [Melaporkan kesalahan dan berkontribusi pada Data Terbuka](#)

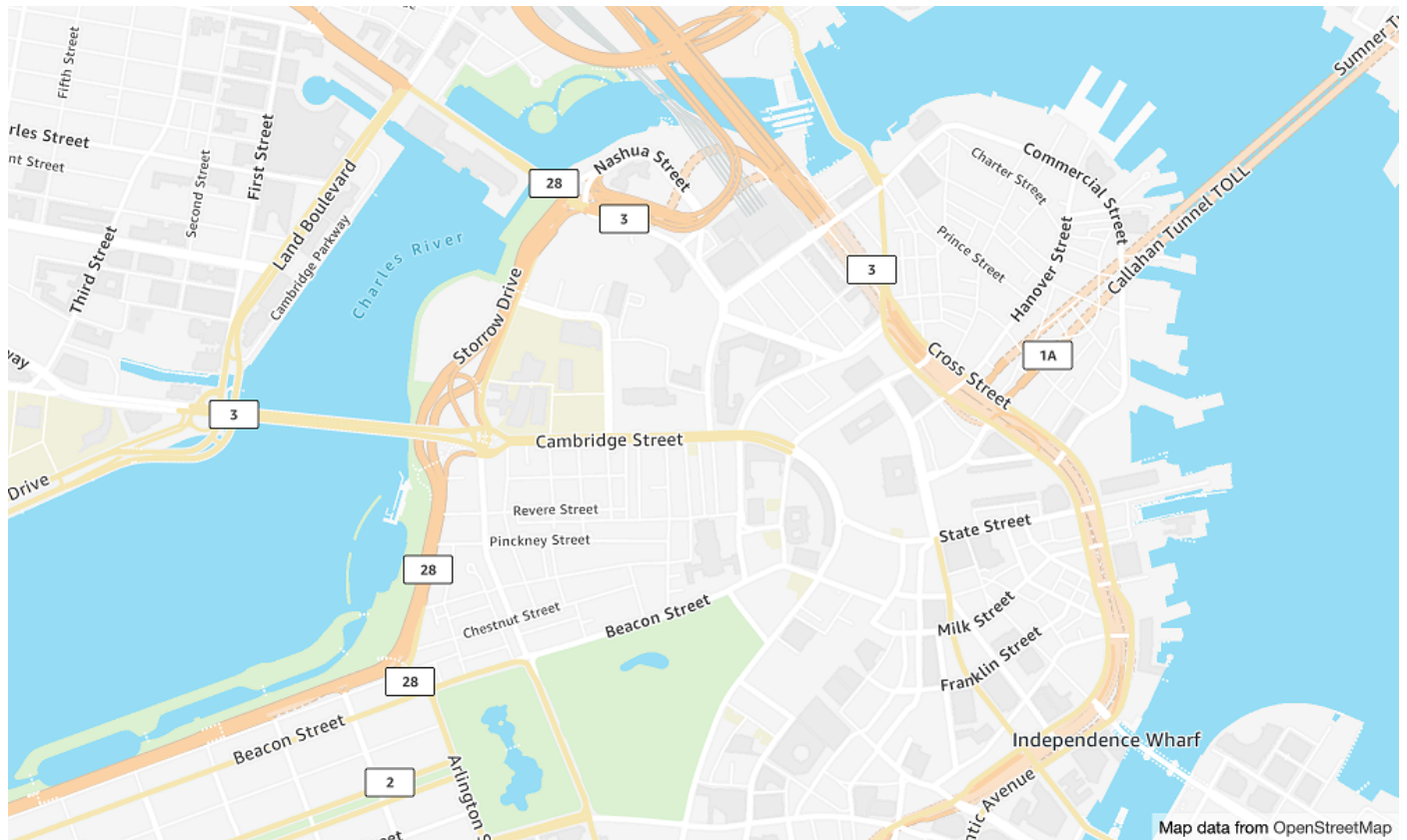
## Buka gaya peta Data

Amazon Location Service mendukung gaya peta berikut saat [membuat sumber daya peta](#):

Buka gaya peta Data mendukung alternatif [Pandangan politik](#).

### Open Data Standard Light

#### Lampu Standar Data Terbuka



Nama gaya peta: VectorOpenDataStandardLight

Ini menyediakan peta dasar terperinci untuk dunia dalam gaya peta ringan, cocok untuk penggunaan situs web dan aplikasi seluler. Ini termasuk jalan raya, jalan utama, jalan kecil, kereta api, fitur air, kota, taman, landmark, jejak bangunan, dan batas-batas administrasi.


Peta dasar ini didasarkan pada [distribusi peta OSM Daylight](#) yang dikompilasi dari kontributor OpenStreetMap (OSM). Komunitas OSM mencakup lebih dari 1,8 juta kontributor yang memperbarui lebih dari 500.000 fitur setiap hari. Amazon Location Service menggabungkan suntingan ini secara teratur.

## Font

Lokasi Amazon menyajikan font menggunakan [GetMapGlyphs](#). Berikut ini adalah tumpukan font yang tersedia untuk peta ini:

- Amazon Ember Bold, Noto Sans Bold
- Amazon Ember Kental RC Bold, Noto Sans Bold
- Amazon Ember Kental RC Regular, Noto Sans Regular

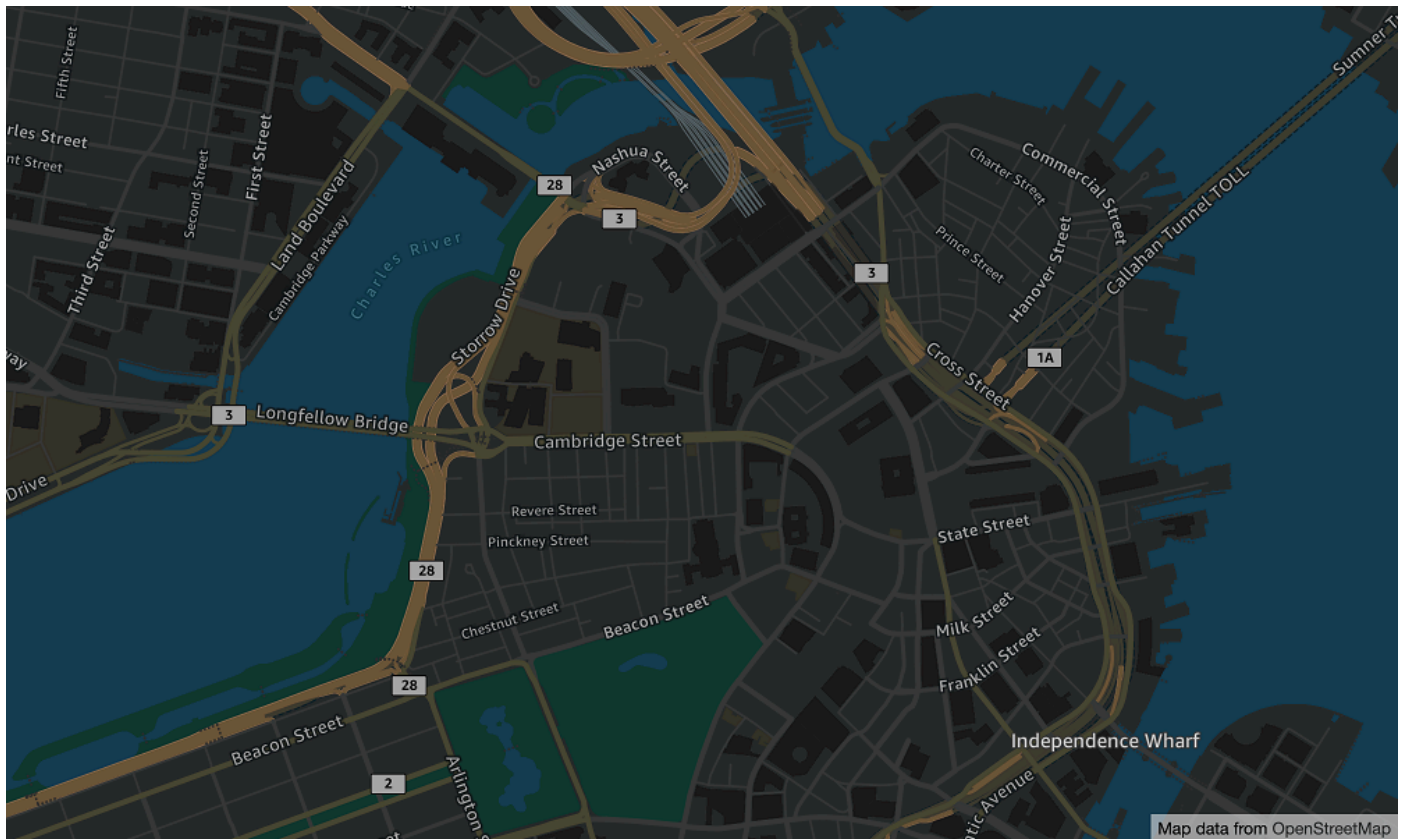
- Amazon Ember Medium, Noto Sans Medium
- Amazon Ember Regular Italic, Noto Sans Italic
- Amazon Ember Regular, Noto Tanpa Regular
- Amazon Ember Regular, Noto Sans Regular, Noto Sans Arab Regular
- Amazon Ember Kental RC Bold, Noto Sans Bold, Noto Sans Arabic Condensed Bold
- Amazon Ember Bold, Noto Sans Bold, Noto Sans Arabic Bold
- Amazon Ember Regular Italic, Noto Sans Italic, Noto Sans Arabic Regular
- Amazon Ember Kental RC Regular, Noto Sans Regular, Noto Sans Arab Terkondensasi Regular
- Amazon Ember Medium, Noto Sans Medium, Noto Sans Media Arab

 Note

Font yang digunakan oleh VectorOpenDataStandardLight adalah font gabungan yang digunakan Amazon Ember untuk sebagian besar mesin terbang tetapi Noto Sans untuk mesin terbang yang tidak didukung oleh Amazon Ember

Open Data Standard Dark

Buka Data Standar Gelap



Nama gaya peta: VectorOpenDataStandardDark

Ini adalah gaya peta bertema gelap yang menyediakan peta dasar terperinci untuk dunia, cocok untuk penggunaan situs web dan aplikasi seluler. Ini termasuk jalan raya, jalan utama, jalan kecil, kereta api, fitur air, kota, taman, landmark, jejak bangunan, dan batas-batas administrasi.


Peta dasar ini didasarkan pada [distribusi peta OSM Daylight](#) yang dikompilasi dari kontributor OpenStreetMap (OSM). Komunitas OSM mencakup lebih dari 1,8 juta kontributor yang memperbarui lebih dari 500.000 fitur setiap hari. Amazon Location Service menggabungkan suntingan ini secara teratur.

## Font

Lokasi Amazon menyajikan font menggunakan [GetMapGlyphs](#). Berikut ini adalah tumpukan font yang tersedia untuk peta ini:

- Amazon Ember Bold, Noto Sans Bold
- Amazon Ember Kental RC Bold, Noto Sans Bold
- Amazon Ember Kental RC Regular, Noto Sans Regular

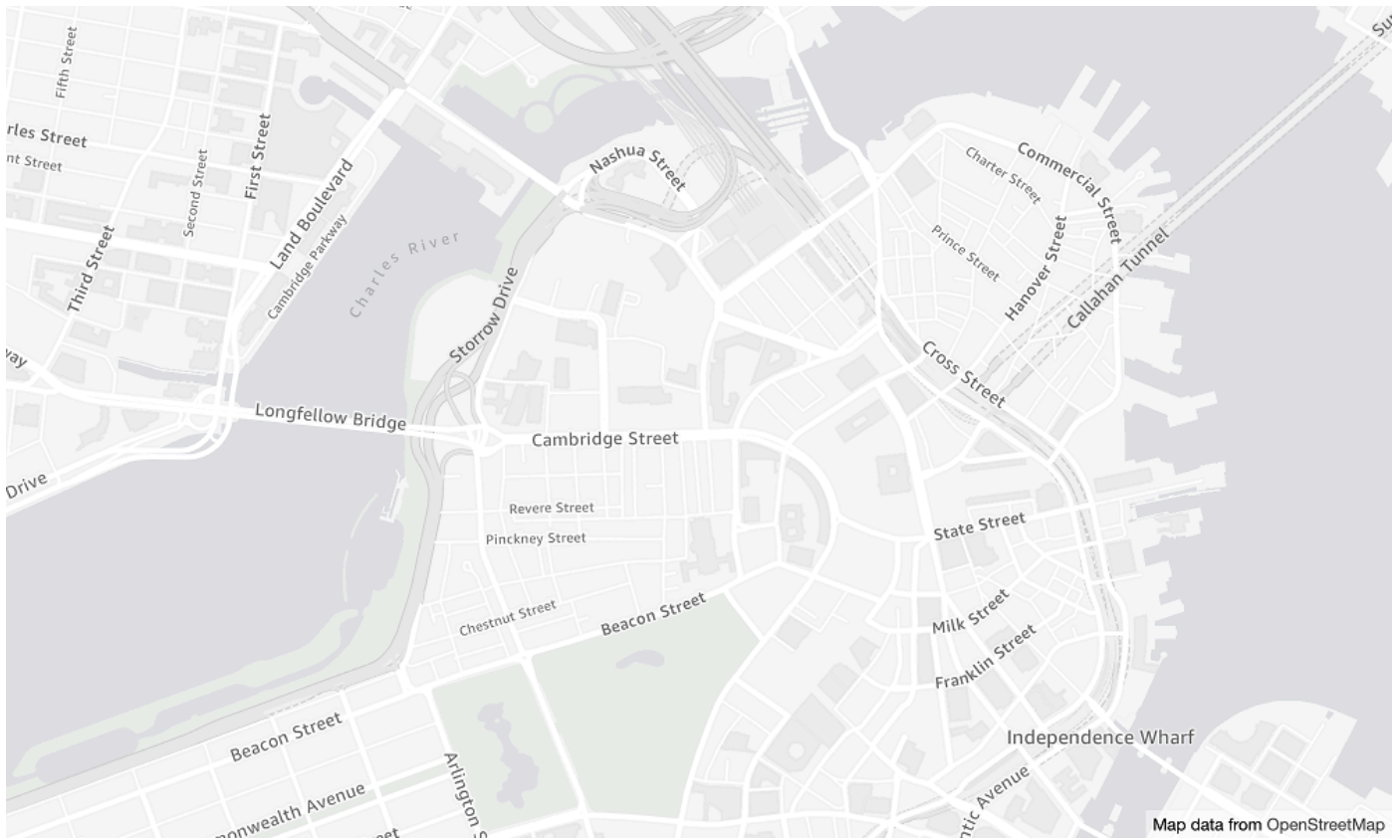
- Amazon Ember Medium, Noto Sans Medium
- Amazon Ember Regular Italic, Noto Sans Italic
- Amazon Ember Regular, Noto Tanpa Regular
- Amazon Ember Regular, Noto Sans Regular, Noto Sans Arab Regular
- Amazon Ember Kental RC Bold, Noto Sans Bold, Noto Sans Arabic Condensed Bold
- Amazon Ember Bold, Noto Sans Bold, Noto Sans Arabic Bold
- Amazon Ember Regular Italic, Noto Sans Italic, Noto Sans Arabic Regular
- Amazon Ember Kental RC Regular, Noto Sans Regular, Noto Sans Arab Terkondensasi Regular
- Amazon Ember Medium, Noto Sans Medium, Noto Sans Media Arab

 Note

Font yang digunakan oleh VectorOpenDataStandardDark adalah font gabungan yang digunakan Amazon Ember untuk sebagian besar mesin terbang tetapi Noto Sans untuk mesin terbang yang tidak didukung oleh Amazon Ember

## Open Data Visualization Light

### Buka Lampu Visualisasi Data



Nama gaya peta: `VectorOpenDataVisualizationLight`

Ini adalah gaya bertema cahaya dengan warna yang diredam dan lebih sedikit fitur yang membantu dalam memahami data overlay.

Peta dasar ini didasarkan pada [distribusi peta OSM Daylight](#) yang dikompilasi dari kontributor OpenStreetMap (OSM). Komunitas OSM mencakup lebih dari 1,8 juta kontributor yang memperbarui lebih dari 500.000 fitur setiap hari. Amazon Location Service menggabungkan suntingan ini secara teratur.

Font

Lokasi Amazon menyajikan font menggunakan [GetMapGlyphs](#). Berikut ini adalah tumpukan font yang tersedia untuk peta ini:

- Amazon Ember Bold, Noto Sans Bold
- Amazon Ember Kental RC Bold, Noto Sans Bold
- Amazon Ember Kental RC Reguler, Noto Sans Reguler
- Amazon Ember Medium, Noto Sans Medium



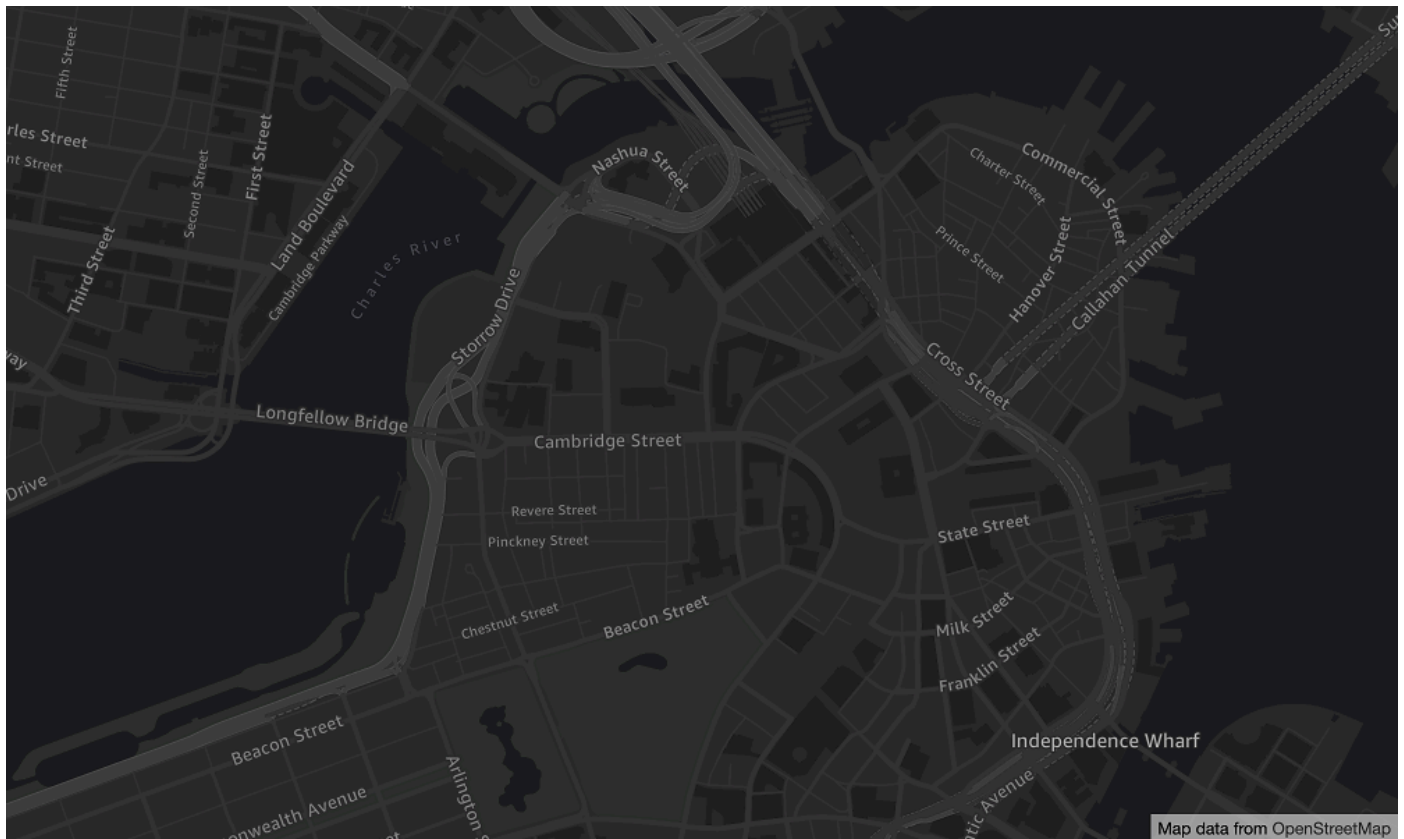
- Amazon Ember Reguler Italic, Noto Sans Italic
- Amazon Ember Reguler, Noto Tanpa Reguler
- Amazon Ember Reguler, Noto Sans Reguler, Noto Sans Arab Reguler
- Amazon Ember Kental RC Bold, Noto Sans Bold, Noto Sans Arabic Condensed Bold
- Amazon Ember Bold, Noto Sans Bold, Noto Sans Arabic Bold
- Amazon Ember Reguler Italic, Noto Sans Italic, Noto Sans Arabic Reguler
- Amazon Ember Kental RC Reguler, Noto Sans Reguler, Noto Sans Arab Terkondensasi Reguler
- Amazon Ember Medium, Noto Sans Medium, Noto Sans Media Arab

 Note

Font yang digunakan oleh `VectorOpenDataVisualizationLight` adalah font gabungan yang digunakan Amazon Ember untuk sebagian besar mesin terbang tetapi Noto Sans untuk mesin terbang yang tidak didukung oleh Amazon Ember

## Open Data Visualization Dark

Buka Visualisasi Data Gelap



Nama gaya peta: `VectorOpenDataVisualizationDark`

Ini adalah gaya bertema gelap dengan warna yang diredam dan lebih sedikit fitur yang membantu dalam memahami data overlay.

Peta dasar ini didasarkan pada [distribusi peta OSM Daylight](#) yang dikompilasi dari kontributor OpenStreetMap (OSM). Komunitas OSM mencakup lebih dari 1,8 juta kontributor yang memperbarui lebih dari 500.000 fitur setiap hari. Amazon Location Service menggabungkan suntingan ini secara teratur.

Font

Lokasi Amazon menyajikan font menggunakan [GetMapGlyphs](#). Berikut ini adalah tumpukan font yang tersedia untuk peta ini:

- Amazon Ember Bold, Noto Sans Bold
- Amazon Ember Kental RC Bold, Noto Sans Bold
- Amazon Ember Kental RC Regular, Noto Sans Regular
- Amazon Ember Medium, Noto Sans Medium

- Amazon Ember Regular Italic, Noto Sans Italic
- Amazon Ember Regular, Noto Tanpa Regular
- Amazon Ember Regular, Noto Sans Regular, Noto Sans Arab Regular
- Amazon Ember Kental RC Bold, Noto Sans Bold, Noto Sans Arabic Condensed Bold
- Amazon Ember Bold, Noto Sans Bold, Noto Sans Arabic Bold
- Amazon Ember Regular Italic, Noto Sans Italic, Noto Sans Arabic Regular
- Amazon Ember Kental RC Regular, Noto Sans Regular, Noto Sans Arab Terkondensasi Regular
- Amazon Ember Medium, Noto Sans Medium, Noto Sans Media Arab

#### Note

Font yang digunakan oleh VectorOpenDataVisualizationDark adalah font gabungan yang digunakan Amazon Ember untuk sebagian besar mesin terbang tetapi Noto Sans untuk mesin terbang yang tidak didukung oleh Amazon Ember

## Cakupan: Data Terbuka

Open Data mencakup peta dengan cakupan global untuk rendering dengan [sumber daya peta Amazon Location Service](#).

#### Note

Open Data hanya untuk digunakan dengan sumber daya peta Amazon Location Service. Anda tidak dapat menggunakan Open Data sebagai penyedia data untuk mendukung kueri geocoding, reverse geocoding, dan pencarian, atau untuk mendukung kueri untuk menghitung rute.

## Ketentuan penggunaan dan atribusi data: Data Terbuka

Sebelum Anda menggunakan Data Terbuka, pastikan Anda dapat mematuhi semua persyaratan hukum yang berlaku, termasuk persyaratan lisensi yang berlaku untuk Data Terbuka dan AWS.

Untuk informasi selengkapnya tentang AWS persyaratan, lihat [Ketentuan Layanan AWS](#).

Untuk informasi tentang pedoman atribusi Data Terbuka, lihat OpenStreetMap [Hak Cipta dan Lisensi serta Pedoman OpenStreetMap Lisensi/Atribusi](#).

## Melaporkan kesalahan dan berkontribusi pada Data Terbuka

OpenStreetMap (OSM) dan Natural Earth adalah proyek data terbuka berbasis komunitas. Jika Anda mengalami masalah dengan data, Anda dapat melaporkan kesalahan atau langsung berkontribusi perbaikan atau saran.

- Untuk melaporkan kesalahan atau menawarkan saran di OSM, Anda dapat membuat catatan di peta. Ini adalah komentar di peta yang membantu kontributor dalam membuat perbaikan pada peta. Anda membuat catatan melalui situs [OpenStreetMap web](#). Untuk informasi selengkapnya tentang catatan, lihat [Catatan](#) di OpenStreetMap wiki.
- Untuk informasi selengkapnya tentang berkontribusi langsung OpenStreetMap, termasuk menambahkan lokasi dan memperbaiki kesalahan, lihat [Menyumbang data peta](#) di OpenStreetMap wiki.
- Untuk mengirimkan permintaan koreksi data di Natural Earth, Anda dapat mengirimkan masalah melalui [situs web Natural Earth](#).

### Note

Memperbaiki kesalahan OpenStreetMap dapat terjadi dengan cepat, namun, diperlukan waktu untuk koreksi muncul dalam distribusi peta. Siang hari dari data OSM yang digunakan oleh penyedia Open Data. Situs web [Daylight Map Distribution](#) memberikan informasi lebih lanjut tentang prosesnya. Selain itu, Amazon Location Service memperbarui data peta yang digunakan di Amazon Location Service kira-kira setiap bulan.

## Fitur oleh penyedia data

Bagian ini menjelaskan fitur yang tersedia di Amazon Location Service, yang dikategorikan berdasarkan penyedia data.

Tabel berikut memberikan ikhtisar fitur tingkat tinggi.

Penyedia data	Cakupan geografis	Cakupan fitur	Wilayah AWS
Esri	Global	Peta, Tempat, Rute	<a href="#">Semua Wilayah</a> di mana Lokasi Amazon tersedia.
Grab	<a href="#">Asia Tenggara</a>	Peta, Tempat, Rute	Asia Pasifik (Singapura), ap-southeast-1, hanya.
KEMARI	Global	Peta, Tempat, Rute	<a href="#">Semua Wilayah</a> di mana Lokasi Amazon tersedia.
Buka Data	Global	Peta	<a href="#">Semua Wilayah</a> di mana Lokasi Amazon tersedia.

Tab berikut menunjukkan detail dalam setiap area fitur.

## Map Features

Tabel berikut menunjukkan fitur peta oleh penyedia data. Untuk informasi lebih lanjut tentang konsep peta, lihat [Peta](#).

Penyedia data	Jenis peta yang didukung	Tingkat zoom vektor	Tingkat zoom raster
Esri	vektor  Raster (citra)  Untuk informasi selengkapnya, lihat <a href="#">Gaya peta Esri</a> .	0-15	0-23
Grab	vektor	0-14	none

Penyedia data	Jenis peta yang didukung	Tingkat zoom vektor	Tingkat zoom raster
	(Hanya <a href="#">Asia Tenggara</a> )  Untuk informasi selengkapnya, lihat <a href="#">Ambil gaya peta</a> .		
KEMARI	vektor  Raster (citra)  Hibrida  Untuk informasi selengkapnya, lihat <a href="#">HERE gaya peta</a> .	1-17	0-19
Buka Data	vektor  Untuk informasi selengkapnya, lihat <a href="#">Buka gaya peta Data</a> .	0-15	none

#### Note

Tingkat zoom mewakili pengaturan maksimum dan minimum, seperti yang didefinisikan dalam API masing-masing penyedia. Area peta yang berbeda mungkin memiliki maksimum yang berbeda; misalnya, ubin laut mungkin memiliki tingkat zoom detail yang lebih sedikit daripada area di kota-kota besar.

MapLibre (dan mesin rendering peta lainnya) memungkinkan Anda untuk mengatur tingkat zoom minimum dan maksimum, dan juga akan menghormati tingkat zoom penyedia data di suatu area, sehingga Anda tidak perlu menulis kode untuk menangani perbedaan ini.

## Places and Search

Tabel berikut menunjukkan fitur tempat dan pencarian oleh penyedia data. Untuk informasi lebih lanjut tentang konsep tempat, lihat [Pencarian tempat](#).

Penyedia data	Geocoding	Geocoding Terbalik	Pelengkapan otomatis	GetPlace
Esri	Semua fitur, kecuali:  PlaceId	Semua fitur, kecuali:  TimeZone  PlaceId	Semua fitur	Semua fitur
Grab	Semua fitur, kecuali:  tipe unit  Kategori tidak didukung	Semua fitur	Semua fitur	Semua fitur, kecuali:  tipe unit  SubMunicipality
KEMARI	Semua fitur, kecuali:  nomor unit  tipe unit  relevansi  <a href="#">Batasan tambahan pada penyaringan</a>	Semua fitur	Semua fitur	Semua fitur, kecuali:  nomor unit  tipe unit  SubMunicipality
Buka Data	Tidak didukung	Tidak didukung	Tidak didukung	Hanya mendukung:  SubMunicipality

## Route features

Tabel berikut menunjukkan fitur rute oleh penyedia data. Untuk informasi selengkapnya tentang konsep rute, lihat [Rute](#). Untuk deskripsi lebih rinci tentang batasan matriks rute, lihat [Pembatasan posisi keberangkatan dan tujuan](#).

Penyedia data	Mode perjalanan	Hitung rute	Matriks rute
Esri	Mobil, Truk, Berjalan	Keberangkatan dan tujuan harus berjarak 400 km satu sama lain. Total waktu perjalanan tidak boleh lebih dari 400 menit.  ArrivalTime tidak didukung.	Hingga 10 posisi keberangkatan dan tujuan.  Tidak didukung di Korea.  Semua pasangan keberangkatan dan tujuan harus berada dalam jarak 400 km satu sama lain.
Grab	Mobil, sepeda motor.  Berjalan dan Sepeda di <a href="#">kota-kota tertentu</a> .	Tidak ada batas jarak.	Hingga 350 posisi keberangkatan dan tujuan.
KEMARI	Mobil, Truk, Berjalan	Tidak ada batas jarak. Rute yang menempuh jarak lebih dari 10 km di luar lingkaran di sekitar posisi keberangkatan dan tujuan tidak akan dihitung.	Hingga 350 posisi keberangkatan dan tujuan.  Semua posisi keberangkatan dan tujuan harus jatuh dengan lingkaran 180 km.  Rute yang lebih panjang didukung,



Penyedia data	Mode perjalanan	Hitung rute	Matriks rute dengan <a href="#">batasan tambahan</a> .
Buka Data	Tidak didukung	Tidak didukung	Tidak didukung

## Ketentuan penggunaan dan atribusi data untuk penyedia data

Sebelum Anda menggunakan penyedia data, pastikan Anda dapat mematuhi semua persyaratan hukum yang berlaku, termasuk persyaratan lisensi yang berlaku untuk penggunaan penyedia.

Untuk informasi selengkapnya tentang AWS persyaratan, lihat [Ketentuan Layanan AWS](#).

Saat menggunakan penyedia data dengan sumber daya Lokasi Amazon untuk aplikasi atau dokumentasi Anda, pastikan untuk memberikan atribusi untuk setiap penyedia data yang Anda gunakan.

Untuk informasi selengkapnya tentang kepatuhan dan atribusi untuk setiap penyedia data, lihat topik berikut.

- Esri — [Ketentuan penggunaan dan atribusi data: Esri](#)
- Ambil — [Ketentuan penggunaan dan atribusi data: Grab](#)
- DI SINI - [Ketentuan penggunaan dan atribusi data: DI SINI](#)
- Buka data — [Ketentuan penggunaan dan atribusi data: Data Terbuka](#)

## Wilayah Lokasi Amazon dan titik akhir

Lokasi Amazon tersedia di AWS Wilayah berikut:

### Wilayah

Nama Wilayah	Wilayah	Titik Akhir	Protokol
AS Timur (Ohio)	us-east-2	geo.us-east-2.amazonaws.com	HTTPS

Nama Wilayah	Wilayah	Titik Akhir	Protokol
AS Timur (Virginia Utara)	us-east-1	geo.us-east-1.amazonaws.com	HTTPS
AS Barat (Oregon)	us-west-2	geo.us-west-2.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	geo.ap-south-1.amazonaws.com	HTTPS
Asia Pasifik (Singapura)	ap-southeast-1	geo.ap-southeast-1.amazonaws.com	HTTPS
Asia Pasifik (Sydney)	ap-southeast-2	geo.ap-southeast-2.amazonaws.com	HTTPS
Asia Pasifik (Tokyo)	ap-northeast-1	geo.ap-northeast-1.amazonaws.com	HTTPS
Kanada (Pusat)	ca-central-1	geo.ca-central-1.amazonaws.com	HTTPS
Eropa (Frankfurt)	eu-central-1	geo.eu-central-1.amazonaws.com	HTTPS
Eropa (Irlandia)	eu-west-1	geo.eu-west-1.amazonaws.com	HTTPS
Eropa (London)	eu-west-2	geo.eu-west-2.amazonaws.com	HTTPS

Nama Wilayah	Wilayah	Titik Akhir	Protokol
Eropa (Stockholm)	eu-north-1	geo.eu-north-1.amazonaws.com	HTTPS
Amerika Selatan (Sao Paulo)	sa-east-1	geo.sa-east-1.amazonaws.com	HTTPS
AWS GovCloud (AS-Barat)	us-gov-west-1	geo.us-gov-west-1.amazonaws.com	HTTPS
		geo-fips.us-gov-west-1.amazonaws.com	HTTPS

### Note

Untuk informasi selengkapnya tentang cara menggunakan titik akhir dalam tabel ini, lihat bagian berikut.

## Titik akhir

Sintaks umum untuk titik akhir Lokasi Wilayahal Amazon adalah sebagai berikut:

```
protocol://service-code.geo.region-code.amazonaws.com
```

Dalam sintaks ini, Amazon Location menggunakan kode layanan berikut:

Layanan	Kode layanan
Peta Amazon Location	peta
Tempat Amazon Location	tempat

Layanan	Kode layanan
Pagar Virtual Amazon Location	pagar virtual
Pelacak Amazon Location	pelacakan
Rute Amazon Location	rute

Misalnya, titik akhir regional untuk Amazon Location Maps for US East (Virginia N.) adalah: [https://maps .geo. us-east-1 .amazonaws.com](https://maps.geo.us-east-1.amazonaws.com).

## Titik Akhir operasi API

Sintaks untuk titik akhir bidang kontrol Amazon Location Service adalah sebagai berikut:

```
protocol://cp.service-code.geo.region-code.amazonaws.com
```

Tindakan bidang kontrol untuk Amazon Location Service adalah:

Layanan	Titik Akhir	Operasi API
Peta Amazon Location	<a href="https://cp.maps.geo.wilayah.amazonaws.com">https://cp.maps.geo.wilayah .amazonaws.com</a>	<a href="#">CreateMap</a> <a href="#">DeleteMap</a> <a href="#">DescribeMap</a> <a href="#">ListMaps</a> <a href="#">UpdateMap</a>
Tempat Amazon Location	<a href="https://cp.places.geo.wilayah.amazonaws.com">https://cp.places.geo.wilayah .amazonaws.com</a>	<a href="#">CreatePlaceIndex</a> <a href="#">DeletePlaceIndex</a> <a href="#">DescribePlaceIndex</a> <a href="#">ListPlaceIndexes</a> <a href="#">UpdatePlaceIndex</a>

Layanan	Titik Akhir	Operasi API
Pagar Virtual Amazon Location	<a href="https://cp.geofencing.geo.wilayah.amazonaws.com">https://cp.geofencing.geo.wilayah .amazonaws.com</a>	<a href="#">CreateGeofenceCollection</a> <a href="#">DeleteGeofenceCollection</a> <a href="#">DescribeGeofenceCollection</a> <a href="#">ListGeofenceCollections</a> <a href="#">UpdateGeofenceCollection</a>
Pelacak Amazon Location	<a href="https://cp.tracking.geo.wilayah .amazonaws.com">https://cp.tracking.geo.wilayah .amazonaws.com</a>	<a href="#">CreateTracker</a> <a href="#">DeleteTracker</a> <a href="#">DescribeTracker</a> <a href="#">UpdateTracker</a> <a href="#">ListTrackers</a> <a href="#">AssociateTrackerConsumer</a> <a href="#">DisassociateTrackerConsumer</a> <a href="#">ListTrackerConsumers</a>
Rute Amazon Location	<a href="https://cp.routes.geo.wilayah .amazonaws.com">https://cp.routes.geo.wilayah .amazonaws.com</a>	<a href="#">CreateRouteCalculator</a> <a href="#">DeleteRouteCalculator</a> <a href="#">DescribeRouteCalculator</a> <a href="#">ListRouteCalculators</a> <a href="#">UpdateRouteCalculator</a>

Layanan	Titik Akhir	Operasi API
Metadata Lokasi Amazon	<a href="https://cp.metadata.geo.wilayah.amazonaws.com">https://cp.metadata.geo.wilayah .amazonaws.com</a>	<a href="#">CreateKey</a> <a href="#">DeleteKey</a> <a href="#">DescribeKey</a> <a href="#">ListKeys</a> <a href="#">UpdateKey</a>

## Kuota Amazon Location Service

Topik ini memberikan ringkasan batas tarif dan kuota untuk Amazon Location Service.

### Note

Jika Anda memerlukan kuota yang lebih tinggi, Anda dapat menggunakan konsol Service Quotas [untuk meminta peningkatan kuota untuk kuota](#) yang dapat disesuaikan. Saat meminta kenaikan kuota, pilih Wilayah tempat Anda memerlukan peningkatan kuota karena sebagian besar kuota khusus untuk Wilayah. AWS

Service Quotas adalah jumlah maksimum sumber daya yang dapat Anda miliki per AWS akun dan AWS Wilayah. Amazon Location Service menolak permintaan tambahan yang melebihi kuota layanan.

Batas nilai (kuota yang dimulai dengan Tingkat... ) adalah jumlah maksimum permintaan per detik, dengan tingkat burst 80 persen dari batas dalam bagian mana pun dari detik, yang ditentukan untuk setiap operasi API. Amazon Location Service membatasi permintaan yang melebihi batas tarif operasi.

Nama	Default	Dapat disesuaikan	Deskripsi
Sumber daya API Key per akun	Setiap Wilayah yang didukung: 500	Tidak	Jumlah maksimum sumber daya kunci API (aktif atau kedaluwarsa) yang dapat Anda miliki per akun.
Sumber daya Koleksi Geofence per akun	Setiap Wilayah yang didukung: 1.500	<a href="#">Ya</a>	Jumlah maksimum sumber daya Geofence Collection yang dapat Anda buat per akun.
Geofences untuk Koleksi Geofence	Setiap Wilayah yang didukung: 50.000	Tidak	Jumlah maksimum Geofences yang dapat Anda buat per Geofence Collection.
Peta sumber daya per akun	Setiap Wilayah yang didukung: 40	<a href="#">Ya</a>	Jumlah maksimum sumber daya Peta yang dapat Anda buat per akun.
Tempatkan sumber daya Indeks per akun	Setiap Wilayah yang didukung: 40	<a href="#">Ya</a>	Jumlah maksimum sumber daya Indeks Tempat yang dapat Anda buat per akun.
Tingkat permintaan AssociateTrackerConsumer API	Setiap Wilayah yang didukung: 10 per detik	<a href="#">Ya</a>	Jumlah maksimum AssociateTrackerConsumer permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.

Nama	Default	Dapat disesu an	Deskripsi
Tingkat permintaan BatchDeleteDevicePositionHistory API	Setiap Wilayah yang didukung: 50 per detik	<a href="#">Ya</a>	Jumlah maksimum BatchDeleteDevicePositionHistory permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan BatchDeleteGeofence API	Setiap Wilayah yang didukung: 50 per detik	<a href="#">Ya</a>	Jumlah maksimum BatchDeleteGeofence permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan BatchEvaluateGeofences API	Setiap Wilayah yang didukung: 50 per detik	<a href="#">Ya</a>	Jumlah maksimum BatchEvaluateGeofences permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan BatchGetDevicePosition API	Setiap Wilayah yang didukung: 50 per detik	<a href="#">Ya</a>	Jumlah maksimum BatchGetDevicePosition permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.



Nama	Default	Dapat disesu an	Deskripsi
Tingkat permintaan BatchPutGeofence API	Setiap Wilayah yang didukung: 50 per detik	<a href="#"><u>Ya</u></a>	Jumlah maksimum BatchPutGeofence permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan BatchUpdateDevicePosition API	Setiap Wilayah yang didukung: 50 per detik	<a href="#"><u>Ya</u></a>	Jumlah maksimum BatchUpdateDevicePosition permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan CalculateRoute API	Setiap Wilayah yang didukung: 10 per detik	<a href="#"><u>Ya</u></a>	Jumlah maksimum CalculateRoute permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan CalculateRouteMatrix API	Setiap Wilayah yang didukung: 5 per detik	<a href="#"><u>Ya</u></a>	Jumlah maksimum CalculateRouteMatrix permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan CreateGeofenceCollection API	Setiap Wilayah yang didukung: 10 per detik	<a href="#"><u>Ya</u></a>	Jumlah maksimum CreateGeofenceCollection permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.

Nama	Default	Dapat disesu an	Deskripsi
Tingkat permintaan CreateKey API	Setiap Wilayah yang didukung: 10 per detik	<a href="#">Ya</a>	Jumlah maksimum CreateKey permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan CreateMap API	Setiap Wilayah yang didukung: 10 per detik	<a href="#">Ya</a>	Jumlah maksimum CreateMap permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan CreatePlaceIndex API	Setiap Wilayah yang didukung: 10 per detik	<a href="#">Ya</a>	Jumlah maksimum CreatePlaceIndex permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan CreateRouteCalculator API	Setiap Wilayah yang didukung: 10 per detik	<a href="#">Ya</a>	Jumlah maksimum CreateRouteCalculator permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan CreateTracker API	Setiap Wilayah yang didukung: 10 per detik	<a href="#">Ya</a>	Jumlah maksimum CreateTracker permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.

Nama	Default	Dapat disesu an	Deskripsi
Tingkat permintaan DeleteGeofenceCollection API	Setiap Wilayah yang didukung: 10 per detik	<a href="#">Ya</a>	Jumlah maksimum DeleteGeofenceCollection permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan DeleteKey API	Setiap Wilayah yang didukung: 10 per detik	<a href="#">Ya</a>	Jumlah maksimum DeleteKey permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan DeleteMap API	Setiap Wilayah yang didukung: 10 per detik	<a href="#">Ya</a>	Jumlah maksimum DeleteMap permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan DeletePlaceIndex API	Setiap Wilayah yang didukung: 10 per detik	<a href="#">Ya</a>	Jumlah maksimum DeletePlaceIndex permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan DeleteRouteCalculator API	Setiap Wilayah yang didukung: 10 per detik	<a href="#">Ya</a>	Jumlah maksimum DeleteRouteCalculator permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.

Nama	Default	Dapat disesu an	Deskripsi
Tingkat permintaan DeleteTracker API	Setiap Wilayah yang didukung: 10 per detik	<a href="#">Ya</a>	Jumlah maksimum DeleteTracker permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan DescribeGeofenceCollection API	Setiap Wilayah yang didukung: 10 per detik	<a href="#">Ya</a>	Jumlah maksimum DescribeGeofenceCollection permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan DescribeKey API	Setiap Wilayah yang didukung: 10 per detik	<a href="#">Ya</a>	Jumlah maksimum DescribeKey permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan DescribeMap API	Setiap Wilayah yang didukung: 10 per detik	<a href="#">Ya</a>	Jumlah maksimum DescribeMap permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan DescribePlaceIndex API	Setiap Wilayah yang didukung: 10 per detik	<a href="#">Ya</a>	Jumlah maksimum DescribePlaceIndex permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.

Nama	Default	Dapat disesu an	Deskripsi
Tingkat permintaan DescribeRouteCalculator API	Setiap Wilayah yang didukung: 10 per detik	<a href="#">Ya</a>	Jumlah maksimum DescribeRouteCalculator permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan DescribeTracker API	Setiap Wilayah yang didukung: 10 per detik	<a href="#">Ya</a>	Jumlah maksimum DescribeTracker permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan DisassociateTrackerConsumer API	Setiap Wilayah yang didukung: 10 per detik	<a href="#">Ya</a>	Jumlah maksimum DisassociateTrackerConsumer permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan ForecastGeofenceEvents API	Setiap Wilayah yang didukung: 50 per detik	<a href="#">Ya</a>	Jumlah maksimum ForecastGeofenceEvents permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.

Nama	Default	Dapat disesu an	Deskripsi
Tingkat permintaan GetDevicePosition API	Setiap Wilayah yang didukung: 50 per detik	<a href="#">Ya</a>	Jumlah maksimum GetDevicePosition permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan GetDevicePositionHistory API	Setiap Wilayah yang didukung: 50 per detik	<a href="#">Ya</a>	Jumlah maksimum GetDevicePositionHistory permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan GetGeofence API	Setiap Wilayah yang didukung: 50 per detik	<a href="#">Ya</a>	Jumlah maksimum GetGeofence permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan GetMapGlyphs API	Setiap Wilayah yang didukung: 50 per detik	<a href="#">Ya</a>	Jumlah maksimum GetMapGlyphs permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan GetMapSprites API	Setiap Wilayah yang didukung: 50 per detik	<a href="#">Ya</a>	Jumlah maksimum GetMapSprites permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.

Nama	Default	Dapat disesuaikan	Deskripsi
Tingkat permintaan GetMapStyleDescriptor API	Setiap Wilayah yang didukung: 50 per detik	<a href="#">Ya</a>	Jumlah maksimum GetMapStyleDescriptor permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan GetMapTile API	Setiap Wilayah yang didukung: 500 per detik	<a href="#">Ya</a>	Jumlah maksimum GetMapTile permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan GetPlace API	Setiap Wilayah yang didukung: 50 per detik	<a href="#">Ya</a>	Jumlah maksimum GetPlace permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan ListDevicePositions API	Setiap Wilayah yang didukung: 50 per detik	<a href="#">Ya</a>	Jumlah maksimum ListDevicePositions permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan ListGeofenceCollections API	Setiap Wilayah yang didukung: 10 per detik	<a href="#">Ya</a>	Jumlah maksimum ListGeofenceCollections permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.

Nama	Default	Dapat disesu an	Deskripsi
Tingkat permintaan ListGeofences API	Setiap Wilayah yang didukung: 50 per detik	<a href="#"><u>Ya</u></a>	Jumlah maksimum ListGeofences permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan ListKeys API	Setiap Wilayah yang didukung: 10 per detik	<a href="#"><u>Ya</u></a>	Jumlah maksimum ListKeys permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan ListMaps API	Setiap Wilayah yang didukung: 10 per detik	<a href="#"><u>Ya</u></a>	Jumlah maksimum ListMaps permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan ListPlaceIndexes API	Setiap Wilayah yang didukung: 10 per detik	<a href="#"><u>Ya</u></a>	Jumlah maksimum ListPlaceIndexes permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan ListRouteCalculators API	Setiap Wilayah yang didukung: 10 per detik	<a href="#"><u>Ya</u></a>	Jumlah maksimum ListRouteCalculators permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.




Nama	Default	Dapat disesu an	Deskripsi
Tingkat permintaan ListTagsForResource API	Setiap Wilayah yang didukung: 10 per detik	<a href="#">Ya</a>	Jumlah maksimum ListTagsForResource permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan ListTrackerConsumer API	Setiap Wilayah yang didukung: 10 per detik	<a href="#">Ya</a>	Jumlah maksimum ListTrackerConsumer permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan ListTrackers API	Setiap Wilayah yang didukung: 10 per detik	<a href="#">Ya</a>	Jumlah maksimum ListTrackers permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan PutGeofence API	Setiap Wilayah yang didukung: 50 per detik	<a href="#">Ya</a>	Jumlah maksimum PutGeofence permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan SearchPlaceIndexForPosition API	Setiap Wilayah yang didukung: 50 per detik	<a href="#">Ya</a>	Jumlah maksimum SearchPlaceIndexForPosition permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.

Nama	Default	Dapat disesu an	Deskripsi
Tingkat permintaan SearchPlaceIndexForSuggestions API	Setiap Wilayah yang didukung: 50 per detik	<a href="#"><u>Ya</u></a>	Jumlah maksimum SearchPlaceIndexForSuggestions permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan SearchPlaceIndexForText API	Setiap Wilayah yang didukung: 50 per detik	<a href="#"><u>Ya</u></a>	Jumlah maksimum SearchPlaceIndexForText permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan TagResource API	Setiap Wilayah yang didukung: 10 per detik	<a href="#"><u>Ya</u></a>	Jumlah maksimum TagResource permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan UntagResource API	Setiap Wilayah yang didukung: 10 per detik	<a href="#"><u>Ya</u></a>	Jumlah maksimum UntagResource permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan UpdateGeofenceCollection API	Setiap Wilayah yang didukung: 10 per detik	<a href="#"><u>Ya</u></a>	Jumlah maksimum UpdateGeofenceCollection permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.

Nama	Default	Dapat disesu an	Deskripsi
Tingkat permintaan UpdateKey API	Setiap Wilayah yang didukung: 10 per detik	<a href="#">Ya</a>	Jumlah maksimum UpdateKey permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan UpdateMap API	Setiap Wilayah yang didukung: 10 per detik	<a href="#">Ya</a>	Jumlah maksimum UpdateMap permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan UpdatePlaceIndex API	Setiap Wilayah yang didukung: 10 per detik	<a href="#">Ya</a>	Jumlah maksimum UpdatePlaceIndex permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan UpdateRouteCalculator API	Setiap Wilayah yang didukung: 10 per detik	<a href="#">Ya</a>	Jumlah maksimum UpdateRouteCalculator permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Tingkat permintaan UpdateTracker API	Setiap Wilayah yang didukung: 10 per detik	<a href="#">Ya</a>	Jumlah maksimum UpdateTracker permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.

Nama	Default	Dapat disesuaikan	Deskripsi
Tingkat permintaan VerifyDevicePosition API	Setiap Wilayah yang didukung: 50 per detik	<a href="#">Ya</a>	Jumlah maksimum VerifyDevicePosition permintaan yang dapat Anda buat per detik. Permintaan tambahan diperlambat.
Sumber daya Kalkulator Rute per akun	Setiap Wilayah yang didukung: 40	<a href="#">Ya</a>	Jumlah maksimum sumber daya Kalkulator Rute yang dapat Anda buat per akun.
Pelacak konsumen per pelacak	Setiap Wilayah yang didukung: 5	Tidak	Jumlah maksimum Koleksi Geofence yang dapat dikaitkan dengan sumber daya Tracker.
Sumber daya pelacak per akun	Setiap Wilayah yang didukung: 500	<a href="#">Ya</a>	Jumlah maksimum sumber daya Pelacak yang dapat Anda buat per akun.

 Note

Anda dapat memantau penggunaan Anda terhadap kuota Anda dengan Cloudwatch. Untuk informasi selengkapnya, lihat [Menggunakan CloudWatch untuk memantau penggunaan terhadap kuota](#).

## Mengelola kuota layanan Lokasi Amazon Anda

Amazon Location Service terintegrasi dengan Service Quotas, AWS layanan yang memungkinkan Anda melihat dan mengelola kuota dari lokasi pusat. Untuk informasi selengkapnya, lihat [Apa itu Service Quotas?](#) di Panduan Pengguna Service Quotas.

Service Quotas memudahkan untuk mencari nilai kuota layanan Lokasi Amazon Anda.

### AWS Management Console

Untuk melihat kuota layanan Lokasi Amazon menggunakan konsol

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

Dalam daftar service quotas, Anda dapat melihat nama service quotas, nilai terapan (jika tersedia), kuota default AWS , dan apakah nilai kuota dapat disesuaikan.

4. Untuk melihat informasi tambahan tentang service quotas, seperti deskripsi, pilih nama kuota.
5. (Opsional) Untuk meminta peningkatan kuota, pilih kuota yang ingin Anda tingkatkan, pilih Request quota increase (Meminta kenaikan kuota), masukkan atau pilih informasi yang diperlukan, dan pilih Request (Permintaan).

Untuk bekerja lebih lanjut dengan kuota layanan menggunakan konsol lihat [Panduan Pengguna Service Quotas](#). Untuk meminta kenaikan kuota, lihat [Meminta kenaikan kuota](#) dalam Panduan Pengguna Service Quotas.

### AWS CLI

Untuk melihat kuota layanan Lokasi Amazon menggunakan AWS CLI

Jalankan perintah berikut untuk melihat kuota Lokasi Amazon default.

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

Untuk bekerja lebih banyak 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](#).

# Memulai sebagai pengembang menggunakan Amazon Location Service

Anda dapat menggunakan Amazon Location Service untuk menyediakan fungsionalitas terkait geografis untuk aplikasi di berbagai bentuk dan sistem, termasuk layanan web backend, aplikasi web, dan aplikasi seluler. Ada banyak alat yang disediakan untuk membantu Anda membangun aplikasi, termasuk SDK, pustaka, dan kode sampel.

Bagian ini menyediakan informasi dan tautan untuk membantu Anda memulai dengan Lokasi Amazon. Secara khusus, topik berikut memberikan informasi yang paling membantu Anda:

- [Skenario dan kasus penggunaan](#)— Daftar skenario pengembangan dan bagaimana Amazon Location Service dapat membantu Anda menyelesaikannya.
- [SDK dan alat Lokasi Amazon](#) — Kit pengembangan perangkat lunak (SDK) dan pustaka yang akan membantu Anda saat memprogram dengan Lokasi Amazon.
- [Referensi API Amazon Location Service](#) — Referensi ke API Lokasi Amazon inti yang dikirimkan bersama AWS SDK.
- [Contoh kode](#) - Bagian ini menyediakan contoh yang akan membantu Anda memulai atau menambahkan fungsionalitas ke aplikasi yang ada.
- [Tutorial mulai cepat](#) - Tutorial ini menunjukkan cara membuat aplikasi pertama Anda. Ada versi tutorial untuk membuat aplikasi web atau aplikasi seluler berbasis Android.
- [Konsep Amazon Location Service](#)— Bagian panduan ini menjelaskan konsep dasar Lokasi Amazon, termasuk bagian tentang Peta, pencarian Tempat, Rute, dan Geofences dan Pelacak.
- [Amplify](#) - Amplify adalah solusi lengkap yang merangkum banyak fungsi yang diperlukan untuk membuat aplikasi web dan seluler menggunakan aplikasi. AWS Cloud Jika Anda sudah menggunakan Amplify, atau memilih untuk menggunakan Amplify, ia memiliki pustaka geo menggunakan Amazon Location Service bawaan yang dapat Anda gunakan. [Untuk memulai Amplify Geo, lihat dokumentasinya di sini.](#)

## Skenario dan kasus penggunaan

Amazon Location Service adalah layanan yang berjalan di AWS Cloud. Anda dapat menyebutnya dari instans Amazon EC2 Anda sendiri di cloud, tetapi banyak aplikasi pemetaan akan berjalan di

perangkat, atau kombinasi perangkat dan cloud. Berikut ini mencantumkan beberapa skenario umum dan bagaimana Anda bisa mendekati pengembangannya.

- Aplikasi backend yang membantu Anda mengoptimalkan rute untuk pengemudi di armada Anda.

Anda dapat menulis aplikasi yang berjalan di [Amazon EC2](#) di AWS Cloud yang menggunakan Amazon Location Service untuk [menghitung matriks rute](#) sebagai input ke pengoptimal rute untuk armada Anda. Gunakan [AWS SDK](#) untuk melakukan panggilan ke Lokasi Amazon.

- Aplikasi web yang memungkinkan pelanggan Anda menemukan lokasi bisnis Anda.

Anda dapat membuat situs web yang berjalan di instans Amazon EC2, termasuk aplikasi berbasis lokasi. Gunakan [AWS SDK JavaScript untuk](#) mengembangkan aplikasi web untuk mencari lokasi menggunakan [pencarian tempat](#), dan menampilkan hasil pada [peta](#) menggunakan MapLibre. Gunakan Amazon Location SDK untuk mempermudah pemrograman dengan lokasi.

- Tambahkan fitur lokasi ke aplikasi iOS atau Android yang ada.

Anda dapat menggunakan AWS SDK untuk Swift (iOS) atau [Kotlin](#) (Android) untuk melakukan panggilan ke Lokasi Amazon guna menambahkan fungsi [pencarian tempat](#) dan [peta](#) ke aplikasi Anda. Gunakan MapLibre untuk membuat peta. Ada [AWS SDK](#) tambahan yang tersedia untuk bahasa lain.

- Lacak aset (perangkat atau kendaraan), dan dapatkan pembaruan saat mereka masuk atau keluar area yang Anda tentukan.

Aplikasi untuk melacak perangkat terdiri dari beberapa bagian.

- Setiap perangkat yang Anda lacak harus memiliki sumber daya [pelacak](#) yang dibuat untuk melacaknya. Itu harus mengirim pembaruan posisi ke Amazon Location Service, misalnya, dengan menggunakan [MQTT](#).
- Buat [geofences](#) untuk menentukan area yang ingin Anda dapatkan masuk dan keluar acara untuk aset Anda.
- Anda dapat menggunakan [Amazon EC2](#) atau [AWS Lambda](#) untuk menanggapi peristiwa Anda saat aset masuk atau keluar dari area geofence.
- Anda dapat memperluas ini untuk membuat aplikasi web atau perangkat untuk melacak dan menampilkan lokasi aset Anda di peta.

Bagian berikut memberikan rincian tentang alat dan pustaka yang tersedia untuk digunakan dengan setiap aspek Amazon Location Service.



# SDK dan alat untuk menggunakan Amazon Location Service

Ada beberapa alat yang akan membantu Anda menggunakan Amazon Location Service.

- **AWS SDK** — Perangkat pengembangan AWS perangkat lunak (SDK) tersedia dalam banyak bahasa pemrograman populer, menyediakan API, contoh kode, dan dokumentasi yang membuatnya lebih mudah untuk membangun aplikasi dalam bahasa pilihan Anda. AWS SDK mencakup API dan fungsionalitas Lokasi Amazon inti, termasuk akses ke Peta, pencarian Tempat, Rute, Geofence, dan Pelacak. Untuk mempelajari lebih lanjut tentang SDK yang tersedia untuk digunakan dengan Amazon Location Service untuk berbagai aplikasi dan bahasa, lihat [SDK menurut bahasa](#).
- **MapLibre**— Amazon Location Service merekomendasikan merender peta menggunakan mesin [MapLibre](#) rendering. MapLibre adalah mesin untuk menampilkan peta di web atau aplikasi seluler. MapLibre juga memiliki model plugin, dan mendukung antarmuka pengguna untuk mencari dan rute dalam beberapa bahasa dan platform. Untuk mempelajari lebih lanjut tentang penggunaan MapLibre dan fungsionalitas yang disediakan, lihat [MapLibre](#).
- **Amazon Location SDK** — Amazon Location SDK adalah kumpulan pustaka open source yang memudahkan pengembangan aplikasi dengan Amazon Location Service. Pustaka menyediakan fungsionalitas untuk mendukung otentikasi aplikasi seluler dan web, pelacakan lokasi untuk aplikasi seluler, konversi antara tipe data Lokasi Amazon dan [GeoJSON](#), serta paket yang dihosting dari klien Lokasi Amazon untuk SDK v3. AWS Untuk mempelajari lebih lanjut tentang Amazon Location SDK, lihat [SDK Lokasi Amazon](#).

## SDK menurut bahasa

Tabel berikut memberikan informasi tentang AWS SDK dan MapLibre versi untuk bahasa dan kerangka kerja, berdasarkan jenis aplikasi: aplikasi web, seluler, atau backend.

### Versi SDK

Kami menyarankan Anda menggunakan versi terbaru AWS SDK, dan SDK lainnya, yang Anda gunakan dalam proyek Anda, dan untuk menjaga SDK tetap up to date. AWS SDK memberi Anda fitur dan fungsionalitas terbaru, dan juga pembaruan keamanan. Untuk menemukan versi AWS SDK terbaru JavaScript, misalnya, lihat topik [penginstalan browser](#) di AWS SDK untuk JavaScript dokumentasi.

## Web frontend

AWS SDK dan MapLibre versi berikut tersedia untuk pengembangan aplikasi frontend web.

Bahasa/Kerangka	AWS SDK	Kerangka Rendering
Sepenuhnya didukung		
JavaScript	<a href="https://aws.amazon.com/sdk-for-javascript/">https://aws.amazon.com/sdk-for-javascript/</a>	<a href="https://maplibre.org/projects/maplibre-gl-js/">https://maplibre.org/projects/maplibre-gl-js/</a>
ReactJS	<a href="https://aws.amazon.com/sdk-for-javascript/">https://aws.amazon.com/sdk-for-javascript/</a>	<a href="https://github.com/maplibre/maplibre-react-native">https://github.com/maplibre/maplibre-react-native</a>
TypeScript	<a href="https://aws.amazon.com/sdk-for-javascript/">https://aws.amazon.com/sdk-for-javascript/</a>	<a href="https://maplibre.org/projects/maplibre-gl-js/">https://maplibre.org/projects/maplibre-gl-js/</a>
Sebagian didukung		
Bergetar	<a href="https://docs.amplify.aws/start/q/integration/flutter/">https://docs.amplify.aws/start/q/integration/flutter/</a>  Flutter belum sepenuhnya didukung oleh AWS, tetapi dukungan terbatas ditawarkan melalui Amplify.	<a href="https://github.com/maplibre/flutter-maplibre-gl">https://github.com/maplibre/flutter-maplibre-gl</a>  Perpustakaan MapLibre Flutter dianggap eksperimental.
Node.js	<a href="https://aws.amazon.com/sdk-for-javascript/">https://aws.amazon.com/sdk-for-javascript/</a>	Tidak ada MapLibre dukungan untuk Node.js.
PHP	<a href="https://aws.amazon.com/sdk-for-php/">https://aws.amazon.com/sdk-for-php/</a>	Tidak ada MapLibre dukungan untuk PHP.

## Mobile frontend

AWS SDK dan MapLibre versi berikut tersedia untuk pengembangan aplikasi frontend seluler.

Bahasa/Kerangka	AWS SDK	Kerangka Rendering
Sepenuhnya didukung		
Java	<a href="https://aws.amazon.com/sdk-for-java/">https://aws.amazon.com/sdk-for-java/</a>	<a href="https://maplibre.org/projects/maplibre-native/">https://maplibre.org/projects/maplibre-native/</a>
Kotlin	<a href="https://aws.amazon.com/sdk-for-kotlin/">https://aws.amazon.com/sdk-for-kotlin/</a>  <a href="#">SDK Otentikasi Seluler Layanan Lokasi Amazon untuk Android: https://github.com/aws-geospatial/-sdk-android amazon-location-mobile-auth</a>  Amazon Location Service Mobile Tracking SDK for Android: <a href="#">amazon-location-mobile-tracking</a> <a href="https://github.com/aws-geospatial/-sdk-android">https://github.com/aws-geospatial/-sdk-android</a>	Membutuhkan binding kustom, seperti MapLibre berbasis Java.
ObjectiveC	<a href="https://github.com/aws-amplify/aws-sdk-ios">https://github.com/aws-amplify/aws-sdk-ios</a>	<a href="https://maplibre.org/projects/maplibre-native/">https://maplibre.org/projects/maplibre-native/</a>
ReactNative	<a href="https://aws.amazon.com/sdk-for-javascript/">https://aws.amazon.com/sdk-for-javascript/</a>	<a href="https://github.com/maplibre/maplibre-react-native">https://github.com/maplibre/maplibre-react-native</a>
Swift	<a href="https://aws.amazon.com/sdk-for-swift/">https://aws.amazon.com/sdk-for-swift/</a>  <a href="#">SDK Otentikasi Seluler Layanan Lokasi Amazon untuk iOS: https://github.com/aws-geospatial/-sdk-ios amazon-location-mobile-auth</a>	<a href="https://maplibre.org/projects/maplibre-native/">https://maplibre.org/projects/maplibre-native/</a>

Bahasa/Kerangka	AWS SDK	Kerangka Rendering
	Amazon Location Service Mobile Tracking SDK for iOS: <a href="https://github.com/aws-geospatial/-sdk-ios">amazon-location-mobile-tracking</a> <a href="https://github.com/aws-geospatial/-sdk-ios">https://github.com/aws-geospatial/ -sdk-ios</a>	
Sebagian didukung		
Bergetar	<a href="https://docs.amplify.aws/start/q/integration/flutter/">https://docs.amplify.aws/start/q/integration/flutter/</a>  Flutter belum sepenuhnya didukung oleh AWS, tetapi dukungan terbatas ditawarkan melalui Amplify.	<a href="https://github.com/maplibre/flutter-maplibre-gl">https://github.com/maplibre/flutter-maplibre-gl</a>  Perpustakaan MapLibre Flutter dianggap eksperimental.

## Backend application

AWS SDK berikut tersedia untuk pengembangan aplikasi backend. MapLibre tidak tercantum di sini, karena rendering peta biasanya tidak diperlukan untuk aplikasi backend.

Bahasa	AWS SDK
.NET	<a href="https://aws.amazon.com/sdk-for-net/">https://aws.amazon.com/sdk-for-net/</a>
C++	<a href="https://aws.amazon.com/sdk-for-cpp/">https://aws.amazon.com/sdk-for-cpp/</a>
Go	<a href="https://aws.amazon.com/sdk-for-go/">https://aws.amazon.com/sdk-for-go/</a>
Java	<a href="https://aws.amazon.com/sdk-for-java/">https://aws.amazon.com/sdk-for-java/</a>
JavaScript	<a href="https://aws.amazon.com/sdk-for-javascript/">https://aws.amazon.com/sdk-for-javascript/</a>
Node.js	<a href="https://aws.amazon.com/sdk-for-javascript/">https://aws.amazon.com/sdk-for-javascript/</a>
TypeScript	<a href="https://aws.amazon.com/sdk-for-javascript/">https://aws.amazon.com/sdk-for-javascript/</a>

Bahasa	AWS SDK
Kotlin	<a href="https://aws.amazon.com/sdk-for-kotlin/">https://aws.amazon.com/sdk-for-kotlin/</a>
PHP	<a href="https://aws.amazon.com/sdk-for-php/">https://aws.amazon.com/sdk-for-php/</a>
Python	<a href="https://aws.amazon.com/sdk-for-python/">https://aws.amazon.com/sdk-for-python/</a>
Ruby	<a href="https://aws.amazon.com/sdk-for-ruby/">https://aws.amazon.com/sdk-for-ruby/</a>
Rust	<a href="https://aws.amazon.com/sdk-for-rust/">https://aws.amazon.com/sdk-for-rust/</a>  AWS SDK untuk Rust ada di pratinjau pengembang.

## Menggunakan MapLibre alat dan pustaka dengan Lokasi Amazon

Salah satu alat penting untuk membuat aplikasi interaktif dengan Lokasi Amazon adalah MapLibre. [MapLibre](#) terutama mesin rendering untuk menampilkan peta di web atau aplikasi seluler. Namun, itu juga mencakup dukungan untuk plug-in, dan menyediakan fungsionalitas untuk bekerja dengan aspek lain dari Lokasi Amazon. Berikut ini menjelaskan alat yang dapat Anda gunakan, berdasarkan area lokasi yang ingin Anda gunakan.

### Note

Untuk menggunakan aspek apa pun dari Lokasi Amazon, instal [AWS SDK untuk bahasa yang ingin Anda gunakan](#).

### • Peta

Untuk menampilkan peta dalam aplikasi Anda, Anda memerlukan mesin rendering peta yang akan menggunakan data yang disediakan oleh Amazon Location, dan menggambar ke layar. Mesin rendering peta juga menyediakan fungsionalitas untuk menggeser dan memperbesar peta, atau untuk menambahkan penanda atau pin tekan dan anotasi lainnya ke peta.

Amazon Location Service merekomendasikan merender peta menggunakan mesin [MapLibre](#) rendering. MapLibre GL JS adalah mesin untuk menampilkan peta JavaScript, sementara MapLibre Native menyediakan peta untuk iOS atau Android.

MapLibre juga memiliki ekosistem plug-in untuk memperluas fungsionalitas inti. Untuk informasi lebih lanjut, kunjungi <https://maplibre.org/maplibre-gl-js-docs/plugins/>.

- Pencarian tempat

Untuk membuat antarmuka pengguna pencarian lebih sederhana, Anda dapat menggunakan [MapLibre geocoder](#) untuk web (aplikasi Android dapat menggunakan [plug-in Android Places](#)).

Gunakan [perpustakaan geocoder Lokasi Amazon untuk Maplibre](#) untuk menyederhanakan proses menggunakan Lokasi Amazon dengan di Aplikasi. `amazon-location-for-maplibre-gl-geocoder` JavaScript

- Rute

Untuk menampilkan rute di peta, gunakan [MapLibrepetunjuk arah](#).

- Geofences dan Pelacak

MapLibre tidak memiliki rendering atau alat khusus untuk geofences dan pelacakan, tetapi Anda dapat menggunakan fungsionalitas rendering dan [plug-in](#) untuk menampilkan geofences dan perangkat yang dilacak di peta.

Perangkat yang dilacak dapat menggunakan [MQTT](#) atau mengirim pembaruan secara manual ke Amazon Location Service. Peristiwa geofence dapat ditanggapi dengan menggunakan [AWS Lambda](#)

Banyak pustaka open source tersedia untuk menyediakan fungsionalitas tambahan untuk Amazon Location Service, misalnya [Turf](#) yang menyediakan fungsionalitas analisis spasial.

Banyak pustaka menggunakan data berformat [GeoJSON](#) standar terbuka. Amazon Location Service menyediakan library untuk mendukung penggunaan GeoJSON dalam aplikasi JavaScript . Untuk informasi lebih lanjut, lihat bagian selanjutnya, [SDK dan pustaka Lokasi Amazon](#).

## Plugin MapLibre Geocoder Lokasi Amazon

Plugin MapLibre geocoder Lokasi Amazon dirancang untuk memudahkan Anda memasukkan fungsionalitas Lokasi Amazon ke dalam JavaScript aplikasi Anda, saat bekerja dengan rendering peta dan geocoding menggunakan perpustakaan. [maplibre-gl-geocoder](#)

## Penginstalan

Anda dapat menginstal plugin MapLibre geocoder Lokasi Amazon dari NPM untuk penggunaan dengan modul, dengan perintah ini:

```
npm install @aws/amazon-location-for-maplibre-gl-geocoder
```

Anda dapat mengimpor ke file HTML untuk digunakan langsung di browser, dengan skrip:

```
<script src="https://www.unpkg.com/@aws/amazon-location-for-maplibre-gl-geocoder@1">/script<
```

## Penggunaan dengan Modul

Kode ini menyiapkan JavaScript peta Maplibre GL dengan kemampuan geocoding Lokasi Amazon. Ini menggunakan otentikasi melalui Amazon Cognito Identity Pool untuk mengakses sumber daya Lokasi Amazon. Peta dirender dengan gaya dan koordinat pusat yang ditentukan, dan memungkinkan untuk mencari tempat di peta.

```
// Import MapLibre GL JS
import maplibregl from "maplibre-gl";
// Import from the AWS JavaScript SDK V3
import { LocationClient } from "@aws-sdk/client-location";
// Import the utility functions
import { withIdentityPoolId } from "@aws/amazon-location-utilities-auth-helper";
// Import the AmazonLocationWithMaplibreGeocoder
import { buildAmazonLocationMaplibreGeocoder, AmazonLocationMaplibreGeocoder } from
"@aws/amazon-location-for-maplibre-gl-geocoder"

const identityPoolId = "Identity Pool ID";
const mapName = "Map Name";
const region = "Region"; // region containing the Amazon Location resource
const placeIndex = "PlaceIndexName" // Name of your places resource in your AWS
Account.

// Create an authentication helper instance using credentials from Amazon Cognito
const authHelper = await withIdentityPoolId("Identity Pool ID");

const client = new LocationClient({
  region: "Region", // Region containing Amazon Location resources
  ...authHelper.getLocationClientConfig(), // Configures the client to use
  credentials obtained via Amazon Cognito
```

```
});

// Render the map
const map = new maplibregl.Map({
  container: "map",
  center: [-123.115898, 49.295868],
  zoom: 10,
  style: `https://maps.geo.${region}.amazonaws.com/maps/v0/maps/${mapName}/style-
descriptor`,
  ...authHelper.getMapAuthenticationOptions(),
});

// Gets an instance of the AmazonLocationMaplibreGeocoder Object.
const amazonLocationMaplibreGeocoder = buildAmazonLocationMaplibreGeocoder(client,
placeIndex, {enableAll: true});

// Now we can add the Geocoder to the map.
map.addControl(amazonLocationMaplibreGeocoder.getPlacesGeocoder());
```

## Penggunaan dengan browser

Contoh ini menggunakan Klien Lokasi Amazon untuk membuat permintaan yang mengautentikasi menggunakan Amazon Cognito.

### Note

Beberapa contoh ini menggunakan Amazon Location Client. Klien Lokasi Amazon didasarkan pada [AWS SDK untuk JavaScript V3 dan memungkinkan untuk](#) melakukan panggilan ke Lokasi Amazon melalui skrip yang direferensikan dalam file HTML.

Sertakan yang berikut ini dalam file HTML:

```
< Import the Amazon Location With Maplibre Geocoder>
<script src="https://www.unpkg.com/@aws/amazon-location-with-maplibre-geocoder@1"></
script>
<Import the Amazon Location Client>
<script src="https://www.unpkg.com/@aws/amazon-location-client@1"></script>
<!Import the utility library>
<script src="https://www.unpkg.com/@aws/amazon-location-utilities-auth-helper@1"></
script>
```



Sertakan yang berikut ini dalam JavaScript file:

```
const identityPoolId = "Identity Pool ID";
const mapName = "Map Name";
const region = "Region"; // region containing Amazon Location resource

// Create an authentication helper instance using credentials from Amazon Cognito
const authHelper = await
  amazonLocationAuthHelper.withIdentityPoolId(identityPoolId);

// Render the map
const map = new maplibregl.Map({
  container: "map",
  center: [-123.115898, 49.295868],
  zoom: 10,
  style: `https://maps.geo.${region}.amazonaws.com/maps/v0/maps/${mapName}/style-
descriptor`,
  ...authHelper.getMapAuthenticationOptions(),
});

// Initialize the AmazonLocationMaplibreGeocoder object
const amazonLocationMaplibreGeocoderObject =
  amazonLocationMaplibreGeocoder.buildAmazonLocationMaplibreGeocoder(client,
  placesName, {enableAll: true});

// Use the AmazonLocationWithMaplibreGeocoder object to add a geocoder to the map.
map.addControl(amazonLocationMaplibreGeocoderObject.getPlacesGeocoder());
```

Di bawah ini adalah fungsi, dan perintah yang digunakan dalam plugin MapLibre geocoder Lokasi Amazon:

- **buildAmazonLocationMaplibreGeocoder**

Kelas ini menciptakan sebuah instance dari AmazonLocationMaplibreGeocoder yang merupakan titik masuk ke yang lain semua panggilan lainnya:

```
const amazonLocationMaplibreGeocoder = buildAmazonLocationMaplibreGeocoder(client,
  placesIndex, {enableAll: true});
```

- **getPlacesGeocoder**

Mengembalikan objek iControl siap pakai yang dapat ditambahkan langsung ke peta.

```
const geocoder = getPlacesGeocoder();

// Initialize map
let map = await initializeMap();

// Add the geocoder to the map.
map.addControl(geocoder);
```

## SDK dan pustaka Lokasi Amazon

Amazon Location SDK adalah sekumpulan pustaka open source yang menyediakan fungsionalitas yang berguna untuk mengembangkan aplikasi Lokasi Amazon. Fungsionalitas berikut disertakan:

- Klien Lokasi Amazon — Objek Lokasi Amazon di AWS SDK v3 dibundel dan dikemas untuk kemudahan penggunaan dalam pengembangan web.
- Otentikasi — Utilitas otentikasi menyederhanakan otentikasi (menggunakan Amazon Cognito atau kunci API) saat membuat halaman web, aplikasi [JavaScriptiOS](#), atau Android untuk [Amazon Location Service](#).
- Pelacakan — SDK pelacakan seluler tersedia untuk [iOS](#) dan [Android](#). SDK ini memudahkan aplikasi seluler untuk berinteraksi dengan Pelacak Lokasi Amazon.
- Fungsi GeoJSON Lokasi Amazon - Utilitas [konversi GeoJSON memudahkan konversi antara data berformat GeoJSON](#) standar industri, dan format [API](#) Lokasi Amazon.

### Topik

- [Cara mulai menggunakan Amazon Location SDK](#)
- [Klien Lokasi Amazon](#)
- [JavaScript Pembantu otentikasi](#)
- [Pembantu konversi GeoJSON](#)
- [SDK Otentikasi Seluler Android](#)
- [SDK Otentikasi Seluler iOS](#)
- [SDK Pelacakan Seluler Android](#)
- [SDK Pelacakan Seluler iOS](#)

## Cara mulai menggunakan Amazon Location SDK

Amazon Location SDK adalah serangkaian fungsi yang dapat membuat penggunaan Amazon Location Service dalam aplikasi menjadi lebih sederhana. Anda dapat menginstal dan mengimpor fungsi-fungsi ini ke dalam JavaScript aplikasi Anda. Bagian berikut menjelaskan klien Lokasi Amazon, dan pustaka pembantu otentikasi dan GeoJSON.

### Klien Lokasi Amazon

Dengan AWS SDK v3, SDK dipisahkan oleh layanan. Anda dapat menginstal hanya bagian-bagian yang Anda butuhkan. Misalnya, untuk menginstal klien Lokasi Amazon dan penyedia kredensial untuk Amazon Cognito, gunakan perintah berikut.

```
npm install @aws-sdk/client-location
npm install @aws-sdk/credential-providers
```

Untuk memfasilitasi penggunaan Amazon Location Service dalam aplikasi JavaScript web frontend, AWS sediakan bundel yang dihosting dari library Amazon Location dan penyedia kredensialnya. Untuk menggunakan klien yang dibundel, tambahkan ke HTML Anda dalam tag skrip, sebagai berikut:

```
<script src="https://unpkg.com/@aws/amazon-location-client@1.x/dist/amazonLocationClient.js"></script>
```

#### Note

Paket ini tetap up to date dan kompatibel ke belakang untuk kemudahan penggunaan. Menggunakan tag skrip ini atau pemasangan NPM akan selalu mendapatkan versi terbaru.

### JavaScript Pembantu otentikasi

Pembantu JavaScript autentikasi Lokasi Amazon membuatnya lebih mudah untuk mengautentikasi saat melakukan panggilan Amazon Location API dari aplikasi Anda. JavaScript Pembantu otentikasi ini secara khusus membantu Anda saat menggunakan [Amazon Cognito](#) atau [kunci API](#) sebagai metode otentikasi Anda. Ini adalah pustaka open source yang tersedia di GitHub, di sini: <https://github.com/aws-geospatial/amazon-location-utilities-auth-helper-js>.

**Note**

Dukungan Amazon Cognito di pembantu otentikasi tidak mendukung fitur identitas federasi Amazon Cognito.

## Instalasi

Anda dapat menggunakan pustaka dengan instalasi lokal, jika Anda menggunakan sistem build seperti webpack, atau dengan menyertakan JavaScript bundel bawaan dengan `<script>` tag di html Anda.

- Gunakan perintah berikut untuk menginstal perpustakaan, menggunakan NPM:

```
npm install @aws/amazon-location-utilities-auth-helper
```

- Gunakan perintah berikut dalam file HTML Anda untuk memuat skrip:

```
<script src="https://unpkg.com/@aws/amazon-location-utilities-auth-helper@1.x/dist/amazonLocationAuthHelper.js"></script>
```

## Impor

Untuk menggunakan fungsi tertentu dalam JavaScript aplikasi Anda, Anda harus mengimpor fungsi itu. Kode berikut digunakan untuk mengimpor fungsi `withIdentityPoolId` ke dalam aplikasi Anda.

```
import { withIdentityPoolId } from '@aws/amazon-location-utilities-auth-helper';
```

## Fungsi otentikasi

Pembantu autentikasi Lokasi Amazon menyertakan fungsi berikut yang mengembalikan objek `AuthHelper`:

- `async withIdentityPoolId( identityPoolId: string): AuthHelper`— Fungsi ini mengembalikan `AuthHelper` objek, diinisialisasi untuk bekerja dengan Amazon Cognito
- `async withAPIKey( API_KEY: string): AuthHelper`— Fungsi ini mengembalikan `AuthHelper` objek, diinisialisasi untuk bekerja dengan Kunci API.

AuthHelperObjek menyediakan fungsi-fungsi berikut:

- `AuthHelper.getMapAuthenticationOptions()`— Fungsi objek ini mengembalikan AuthHelper JavaScript objek dengan `transformRequest` yang dapat digunakan dengan opsi peta di MapLibre JS. Hanya disediakan ketika diinisialisasi dengan kumpulan identitas.
- `AuthHelper.getLocationClientConfig()`— Fungsi objek ini mengembalikan AuthHelper JavaScript objek dengan `credentials` yang dapat digunakan untuk menginisialisasi a `LocationClient`.
- `AuthHelper.getCredentials()`— Fungsi AuthHelper objek ini mengembalikan kredensi internal dari Amazon Cognito. Hanya disediakan ketika diinisialisasi dengan kumpulan identitas.

Contoh: Menginisialisasi objek MapLibre peta dengan Amazon Cognito, menggunakan AuthHelper

```
import { withIdentityPoolId } from '@aws/amazon-location-utilities-auth-helper';

const authHelper = await withIdentityPoolId("identity-pool-id"); // use Cognito pool id
for credentials

const map = new maplibregl.Map({
  container: "map", // HTML element ID of map element
  center: [-123.1187, 49.2819], // initial map center point
  zoom: 16, // initial map zoom
  style: https://maps.geo.region.amazonaws.com/maps/v0/maps/mapName/style-
descriptor', // Defines the appearance of the map
  ...authHelper.getMapAuthenticationOptions(), // Provides credential options
  required for requests to Amazon Location
});
```

Contoh: Menginisialisasi objek MapLibre peta dengan kunci API (tidak **AuthHelper** diperlukan dalam kasus ini)

```
const map = new maplibregl.Map({
  container: "map", // HTML element ID of map element
  center: [-123.1187, 49.2819], // initial map center point
  zoom: 16, // initial map zoom
  style: https://maps.geo.region.amazonaws.com/maps/v0/maps/${mapName}/style-
descriptor?key=api-key-id',
});
```

## Contoh: Inisialisasi klien Lokasi dari AWS SDK untuk JS, menggunakan Amazon Cognito dan AuthHelper

Contoh ini menggunakan AWS SDK untuk JavaScript v3.

```
import { withIdentityPoolId } from '@aws/amazon-location-utilities-auth-helper';

const authHelper = await withIdentityPoolId("identity-pool-id"); // use Cognito pool id
for credentials

//initialize the Location client:
const client = new LocationClient({
  region: "region",
  ...authHelper.getLocationClientConfig() // sets up the Location client to use the
  Cognito pool defined above
});

//call a search function with the location client:
const result = await client.send(new SearchPlaceIndexForPositionCommand({
  IndexName: "place-index", // Place index resource to use
  Position: [-123.1187, 49.2819], // position to search near
  MaxResults: 10 // number of results to return
}));
```

## Contoh: Inisialisasi klien Lokasi dari AWS SDK untuk JS, menggunakan kunci API dan AuthHelper

Contoh ini menggunakan AWS SDK untuk JavaScript v3.

```
import { withAPIKey } from '@aws/amazon-location-utilities-auth-helper';

const authHelper = await withAPIKey("api-key-id"); // use API Key id for credentials

//initialize the Location client:
const client = new LocationClient({
  region: "region",
  ...authHelper.getLocationClientConfig() // sets up the Location client to use the
  API Key defined above
});

//call a search function with the location client:
const result = await client.send(new SearchPlaceIndexForPositionCommand({
  IndexName: "place-index", // Place index resource to use
  Position: [-123.1187, 49.2819], // position to search near
```

```
    MaxResults: 10 // number of results to return
  });
```

## Pembantu konversi GeoJSON

[Pembantu konversi Amazon Location GeoJSON menyediakan alat untuk mengonversi tipe data Amazon Location Service ke dan dari format GeoJSON standar industri.](#) GeoJSON digunakan, misalnya, MapLibre dengan untuk membuat data geografis pada peta. Ini adalah pustaka open source yang tersedia di GitHub, di sini: <https://github.com/aws-geospatial/amazon-location-utilities-datatypes-js>.

### Instalasi

Anda dapat menggunakan pustaka dengan instalasi lokal, seperti webpack, atau dengan menyertakan JavaScript bundel pra-bangun dengan `<script>` tag di html Anda.

- Gunakan perintah berikut untuk menginstal perpustakaan, menggunakan NPM.

```
npm install @aws/amazon-location-utilities-datatypes
```

- Gunakan perintah berikut dalam file HTML Anda untuk memuat skrip:

```
<script src="https://unpkg.com/@aws/amazon-location-utilities-datatypes@1.x/dist/amazonLocationDataConverter.js"></script>
```

### Impor

Untuk menggunakan fungsi tertentu dalam JavaScript aplikasi Anda, Anda harus mengimpor fungsi itu. Kode berikut digunakan untuk mengimpor fungsi `placeToFeatureCollection` ke dalam aplikasi Anda.

```
import { placeToFeatureCollection } from '@aws/amazon-location-utilities-datatypes';
```

### Fungsi konversi GeoJSON

Pembantu konversi GeoJSON Lokasi Amazon menyertakan fungsi-fungsi berikut:

- `placeToFeatureCollection(place: GetPlaceResponse | searchPlaceIndexForPositionResponse | searchPlaceIndexForTextResponse,`

`keepNull: boolean)`: `Feature`— Fungsi ini mengubah respons dari fungsi pencarian tempat ke `FeatureCollection` GeoJSON dengan 1 atau lebih fitur `Point`.

- `devicePositionToFeatureCollection(devicePositions: GetDevicePositionResponse | BatchGetDevicePositionResponse | GetDevicePositionHistoryResponse | ListDevicePositionsResponse, keepNull: boolean)`— Fungsi ini mengubah respons dari fungsi posisi perangkat pelacak ke `GeoJSON FeatureCollection` dengan 1 atau lebih fitur `Point`.
- `routeToFeatureCollection(legs: CalculateRouteResponse): FeatureCollection`— Fungsi ini mengubah respons dari fungsi menghitung rute ke `FeatureCollection` GeoJSON dengan satu fitur. `MultiStringLine` Setiap kaki rute diwakili oleh `LineString` entri di `MultiStringLine`.
- `geofenceToFeatureCollection(geofences: GetGeofenceResponse | PutGeofenceRequest | BatchPutGeofenceRequest | ListGeofencesResponse): FeatureCollection`— Fungsi ini mengubah permintaan fungsi geofence atau respons ke `GeoJSON` dengan fitur `Polygon`. `FeatureCollection` Ini dapat mengonversi geofences baik dalam respons maupun permintaan, memungkinkan Anda untuk menampilkan geofences di peta sebelum mengunggahnya dengan atau. `PutGeofence` `BatchPutGeofence`

Fungsi ini akan mengubah geofensi lingkaran menjadi fitur dengan poligon yang diperkirakan, tetapi juga akan memiliki properti “pusat” dan “radius” untuk membuat ulang geofensi lingkaran, jika perlu (lihat fungsi berikutnya).

- `featureCollectionToGeofences(featureCollection: FeatureCollection): BatchPutGeofenceRequestEntry[]`— Fungsi ini mengubah `FeatureCollection` GeoJSON dengan fitur `Polygon` ke array `BatchPutGeofenceRequestEntry` objek, sehingga hasilnya dapat digunakan untuk membuat permintaan ke. `BatchPutGeofence`

Jika Fitur di `FeatureCollection` memiliki properti “pusat” dan “radius”, itu akan diubah menjadi entri permintaan geofence lingkaran, mengabaikan geometri poligon.

Contoh: Mengkonversi hasil pencarian ke lapisan titik di MapLibre

Contoh ini menggunakan AWS SDK untuk JavaScript v3.

```
import { placeToFeatureCollection } from '@aws/amazon-location-utility-datatypes';
...

```



```
let map; // map here is an initialized MapLibre instance

const client = new LocationClient(config);
const input = { your_input };
const command = new searchPlaceIndexForTextCommand(input);
const response = await client.send(command);

// calling utility function to convert the response to GeoJSON
const featureCollection = placeToFeatureCollection(response);
map.addSource("search-result", featureCollection);
map.addLayer({
  id: "search-result",
  type: "circle",
  source: "search-result",
  paint: {
    "circle-radius": 6,
    "circle-color": "#B42222",
  },
});
```

## SDK Otentikasi Seluler Android

Utilitas ini membantu Anda mengautentikasi saat melakukan panggilan Amazon Location Service API dari aplikasi Android Anda. Ini secara khusus membantu saat menggunakan [Amazon Cognito](#) atau [kunci API sebagai metode](#) otentikasi.

SDK otentikasi seluler Android tersedia di github: Amazon [Location Service Mobile Authentication SDK for Android](#). [Selain itu, baik SDK otentikasi seluler dan SDK tersedia di AWS repositori Maven.AWS](#)

### Penginstalan

Untuk menggunakan SDK autentikasi seluler, tambahkan pernyataan impor berikut ke `build.gradle` file Anda di Android Studio.

```
implementation("software.amazon.location:auth:0.0.1")
implementation("com.amazonaws:aws-android-sdk-location:2.72.0")
```

### Fungsi Otentikasi

SDK pembantu otentikasi memiliki fungsi berikut:

- `authHelper.authenticateWithApiKey("My-Amazon-Location-API-Key")`: `LocationCredentialsProvider`: Fungsi ini mengembalikan `LocationCredentialsProvider` inisialisasi untuk bekerja dengan Kunci API.
- `authHelper.authenticateWithCognitoIdentityPool("My-Cognito-Identity-Pool-Id")`: `LocationCredentialsProvider`: Fungsi ini mengembalikan `LocationCredentialsProvider` inisialisasi untuk bekerja dengan kumpulan identitas Amazon Cognito.

## Penggunaan

Untuk menggunakan SDK dalam kode Anda, impor kelas berikut:

```
import com.amazonaws.services.geo.AmazonLocationClient
import software.amazon.location.auth.AuthHelper
import software.amazon.location.auth.LocationCredentialsProvider
```

Anda memiliki dua opsi saat membuat instance pembantu otentikasi dan penyedia klien lokasi. Anda dapat membuat instance menggunakan [kunci API Lokasi Amazon](#) atau [Amazon Cognito](#).

- Untuk membuat instance pembantu otentikasi menggunakan Amazon Location API Key, deklarasikan kelas helper sebagai berikut:

```
var authHelper = AuthHelper(applicationContext)
var locationCredentialsProvider : LocationCredentialsProvider =
    authHelper.authenticateWithApiKey("My-Amazon-Location-API-Key")
```

- Untuk membuat instance pembantu otentikasi menggunakan Amazon Cognito, deklarasikan kelas helper sebagai berikut:

```
var authHelper = AuthHelper(applicationContext)
var locationCredentialsProvider : LocationCredentialsProvider =
    authHelper.authenticateWithCognitoIdentityPool("My-Cognito-Identity-Pool-Id")
```

Anda dapat membuat instance klien Lokasi Amazon menggunakan penyedia kredensi lokasi dan melakukan panggilan ke layanan Lokasi Amazon. Contoh berikut mencari tempat di dekat garis lintang dan bujur tertentu.

```
var locationClient =
    authHelper.getLocationClient(locationCredentialsProvider.getCredentialsProvider())
var searchPlaceIndexForPositionRequest =
    SearchPlaceIndexForPositionRequest().withIndexName("My-Place-Index-
    Name").withPosition(arrayListOf(30.405423, -97.718833))
var nearbyPlaces =
    locationClient.searchPlaceIndexForPosition(searchPlaceIndexForPositionRequest)
```

## SDK Otentikasi Seluler iOS

Utilitas ini membantu Anda mengautentikasi saat melakukan panggilan Amazon Location Service API dari aplikasi iOS Anda. Ini secara khusus membantu saat menggunakan [Amazon Cognito](#) atau [kunci API sebagai metode](#) otentikasi.

SDK otentikasi seluler iOS tersedia di github: Amazon [Location Service Mobile Authentication SDK for iOS](#).

### Penginstalan

Instal SDK dalam proyek Xcode:

1. Buka File, lalu pilih Add Package Dependencies di proyek XCode Anda.
2. Ketik URL paket: <https://github.com/aws-geospatial/amazon-location-mobile-auth-sdk-ios/> ke bilah pencarian dan tekan tombol enter.
3. Pilih amazon-location-mobile-auth-sdk-ios paket dan tekan Add Package.
4. Pilih produk AmazonLocationiOSAuthSDK paket dan tekan Add Package.

### Fungsi Otentikasi

SDK pembantu otentikasi memiliki fungsi berikut:

- `authHelper.authenticateWithApiKey("My-Amazon-Location-API-Key")`: `LocationCredentialsProvider`: Fungsi ini mengembalikan `LocationCredentialsProvider` inisialisasi untuk bekerja dengan Kunci API.
- `authHelper.authenticateWithCognitoIdentityPool("My-Cognito-Identity-Pool-Id")`: `LocationCredentialsProvider`: Fungsi ini mengembalikan `LocationCredentialsProvider` inisialisasi untuk bekerja dengan kumpulan identitas Amazon Cognito.

## Penggunaan

Untuk menggunakan SDK autentikasi seluler, tambahkan pernyataan berikut ke aktivitas Anda:

```
import AmazonLocationiOSAuthSDK
import AWSLocationXCF
```

Anda memiliki dua opsi saat membuat instance pembantu otentikasi dan penyedia klien lokasi. Anda dapat membuat instance menggunakan [kunci API Lokasi Amazon](#) atau [Amazon Cognito](#).

- Untuk membuat instance pembantu otentikasi menggunakan Amazon Location API Key, deklarasikan kelas helper sebagai berikut:

```
let authHelper = AuthHelper()
let locationCredentialsProvider = authHelper.authenticateWithAPIKey(apiKey: "My-
Amazon-Location-API-Key", region: "account-region")
```

- Untuk membuat instance pembantu otentikasi menggunakan Amazon Cognito, deklarasikan kelas helper sebagai berikut:

```
let authHelper = AuthHelper()
let locationCredentialsProvider =
  authHelper.authenticateWithCognitoUserPool(identityPoolId: "My-Amazon-Location-API-
Key", region: "account-region")
```

Anda dapat membuat instance klien Lokasi Amazon menggunakan penyedia kredensi lokasi dan melakukan panggilan ke layanan Lokasi Amazon. Contoh berikut mencari tempat di dekat garis lintang dan bujur tertentu.

```
let locationClient = AWSLocation.default()
let searchPlaceIndexForPositionRequest =
  AWSLocationSearchPlaceIndexForPositionRequest()!
searchPlaceIndexForPositionRequest.indexName = "My-Place-Index-Name"
searchPlaceIndexForPositionRequest.position = [30.405423, -97.718833]
let nearbyPlaces = locationClient.searchPlaceIndex(forPosition:
  searchPlaceIndexForPositionRequest)
```

## SDK Pelacakan Seluler Android

SDK pelacakan seluler Lokasi Amazon menyediakan utilitas yang membantu dengan mudah mengautentikasi, menangkap posisi perangkat, dan mengirim pembaruan posisi ke Pelacak Lokasi Amazon. SDK mendukung pemfilteran lokal pembaruan lokasi dengan interval pembaruan yang dapat dikonfigurasi. Ini mengurangi biaya data dan mengoptimalkan konektivitas intermiten untuk aplikasi Android Anda.

SDK pelacakan Android tersedia di GitHub: [Amazon Location Mobile Tracking SDK for Android](#). Selain itu, baik SDK otentikasi seluler dan SDK tersedia di AWS repositori Maven.AWS SDK pelacakan Android dirancang untuk bekerja dengan AWS SDK umum.

Bagian ini mencakup topik-topik berikut untuk Amazon Location mobile tracking Android SDK:

### Topik

- [Penginstalan](#)
- [Penggunaan](#)
- [Filter](#)
- [Fungsi pelacakan SDK Seluler Android](#)
- [Contoh](#)

### Penginstalan

Untuk menginstal SDK, tambahkan baris berikut ke bagian dependensi file build.gradle Anda di Android Studio:

```
implementation("software.amazon.location:tracking:0.0.1")
implementation("software.amazon.location:auth:0.0.1")
implementation("com.amazonaws:aws-android-sdk-location:2.72.0")
```

### Penggunaan

Prosedur ini menunjukkan cara menggunakan SDK untuk mengautentikasi dan membuat objek: `LocationTracker`

**Note**

Prosedur ini mengasumsikan Anda telah mengimpor perpustakaan yang disebutkan di [Penginstalan](#) bagian.

1. Impor kelas berikut dalam kode Anda:

```
import software.amazon.location.tracking.LocationTracker
import software.amazon.location.tracking.config.LocationTrackerConfig
import software.amazon.location.tracking.util.TrackingSdkLogLevel
import com.amazonaws.services.geo.AmazonLocationClient
import software.amazon.location.auth.AuthHelper
import software.amazon.location.auth.LocationCredentialsProvider
```

2. Selanjutnya buatAuthHelper, karena LocationCredentialsProvider parameter diperlukan untuk membuat LocationTracker objek:

```
// Create an authentication helper using credentials from Cognito
val authHelper = AuthHelper(applicationContext)
val locationCredentialsProvider : LocationCredentialsProvider =
    authHelper.authenticateWithCognitoIdentityPool("My-Cognito-Identity-Pool-Id")
```

3. Sekarang, gunakan LocationCredentialsProvider dan LocationTrackerConfig untuk membuat LocationTracker objek:

```
val config = LocationTrackerConfig(
    trackerName = "MY-TRACKER-NAME",
    logLevel = TrackingSdkLogLevel.DEBUG,
    accuracy = Priority.PRIORITY_HIGH_ACCURACY,
    latency = 1000,
    frequency = 5000,
    waitForAccurateLocation = false,
    minUpdateIntervalMillis = 5000,
)
locationTracker = LocationTracker(
    applicationContext,
    locationCredentialsProvider,
    config,
)
```

## Filter

Android SDK pelacakan seluler Amazon Location memiliki tiga filter lokasi bawaan.

- **TimeLocationFilter**: Memfilter lokasi saat ini yang akan diunggah berdasarkan interval waktu yang ditentukan.
- **DistanceLocationFilter**: Memfilter pembaruan lokasi berdasarkan ambang jarak yang ditentukan.
- **AccuracyLocationFilter**: Memfilter pembaruan lokasi dengan membandingkan jarak yang dipindahkan sejak pembaruan terakhir dengan akurasi lokasi saat ini.

Contoh ini menambahkan filter `LocationTracker` pada saat pembuatan:

```
val config = LocationTrackerConfig(
    trackerName = "MY-TRACKER-NAME",
    logLevel = TrackingSdkLogLevel.DEBUG,
    accuracy = Priority.PRIORITY_HIGH_ACCURACY,
    latency = 1000,
    frequency = 5000,
    waitForAccurateLocation = false,
    minUpdateIntervalMillis = 5000,
    locationFilters = mutableListOf(TimeLocationFilter(), DistanceLocationFilter(),
    AccuracyLocationFilter())
)
locationTracker = LocationTracker(
    applicationContext,
    locationCredentialsProvider,
    config,
)
```

Contoh ini mengaktifkan dan menonaktifkan filter saat runtime dengan: `LocationTracker`

```
// To enable the filter
locationTracker?.enableFilter(TimeLocationFilter())

// To disable the filter
locationTracker?.disableFilter(TimeLocationFilter())
```

## Fungsi pelacakan SDK Seluler Android

SDK for Android pelacakan seluler Lokasi Amazon mencakup fungsi-fungsi berikut:

- Kelas: `LocationTracker`

```
constructor(context: Context,locationCredentialsProvider:
LocationCredentialsProvider,trackerName: String), atau
constructor(context: Context,locationCredentialsProvider:
LocationCredentialsProvider,clientConfig: LocationTrackerConfig)
```

Ini adalah fungsi penginisialisasi untuk membuat objek. `LocationTracker` Ini membutuhkan contoh `LocationCredentialsProvider`, `trackerName` dan secara opsional sebuah instance dari `LocationTrackingConfig` Jika konfigurasi tidak disediakan, konfigurasi akan diinisialisasi dengan nilai default.

- Kelas: `LocationTracker`

```
start(locationTrackingCallback: LocationTrackingCallback)
```

Memulai proses mengakses lokasi pengguna dan mengirimkannya ke pelacak Lokasi Amazon.

- Kelas: `LocationTracker`

```
isTrackingInForeground()
```

Memeriksa apakah pelacakan lokasi sedang berlangsung.

- Kelas: `LocationTracker`

```
stop()
```

Menghentikan proses melacak lokasi pengguna.

- Kelas: `LocationTracker`

```
startTracking()
```

Memulai proses mengakses lokasi pengguna dan mengirimkannya ke AWS pelacak.

- Kelas: `LocationTracker`

```
startBackground(mode: BackgroundTrackingMode, serviceCallback:
ServiceCallback)
```

Memulai proses mengakses lokasi pengguna dan mengirimkannya ke AWS pelacak saat aplikasi berada di latar belakang. `BackgroundTrackingMode` memiliki opsi berikut:

- ~~ACTIVE\_TRACKING~~: Opsi ini secara aktif melacak pembaruan lokasi pengguna.



- `BATTERY_SAVER_TRACKING`: Opsi ini melacak pembaruan lokasi pengguna setiap 15 menit.
- Kelas: `LocationTracker`

```
stopBackgroundService()
```

Menghentikan proses mengakses lokasi pengguna dan mengirimkannya ke AWS pelacak saat aplikasi berada di latar belakang.

- Kelas: `LocationTracker`

```
getTrackerDeviceLocation()
```

Mengambil lokasi perangkat dari layanan Lokasi Amazon.

- Kelas: `LocationTracker`

```
getDeviceLocation(locationTrackingCallback: LocationTrackingCallback?)
```

Mengambil lokasi perangkat saat ini dari klien penyedia lokasi gabungan dan mengunggahnya ke pelacak Lokasi Amazon.

- Kelas: `LocationTracker`

```
uploadLocationUpdates(locationTrackingCallback: LocationTrackingCallback?)
```

Mengunggah lokasi perangkat ke layanan Lokasi Amazon setelah memfilter berdasarkan filter lokasi yang dikonfigurasi.

- Kelas: `LocationTracker`

```
enableFilter(filter: LocationFilter)
```

Mengaktifkan filter lokasi tertentu.

- Kelas: `LocationTracker`

```
checkFilterIsExistsAndUpdateValue(filter: LocationFilter)
```

Nonaktifkan filter lokasi tertentu.

- Kelas: `LocationTrackerConfig`

```
LocationTrackerConfig( // Required var trackerName: String, // Optional var locationFilters: MutableList =
```

```
mutableListOf( TimeLocationFilter(), DistanceLocationFilter(), ), var
logLevel: TrackingSdkLogLevel = TrackingSdkLogLevel.DEBUG, var accuracy:
Int = Priority.PRIORITY_HIGH_ACCURACY, var latency: Long = 1000, var
frequency: Long = 1500, var waitForAccurateLocation: Boolean = false, var
minUpdateIntervalMillis: Long = 1000, var persistentNotificationConfig:
NotificationConfig = NotificationConfig())
```

Ini menginisialisasi `LocationTrackerConfig` dengan nilai parameter yang ditentukan pengguna. Jika nilai parameter tidak disediakan, itu akan diatur ke nilai default.

- Kelas: `LocationFilter`

```
shouldUpload(currentLocation: LocationEntry, previousLocation:
LocationEntry?): Boolean
```

`LocationFilter` ini adalah protokol yang dapat diterapkan pengguna untuk implementasi filter khusus mereka. Anda perlu menerapkan `shouldUpload` fungsi untuk membandingkan lokasi sebelumnya dan saat ini dan kembali jika lokasi saat ini harus diunggah.

## Contoh

Contoh kode berikut menunjukkan fungsionalitas SDK pelacakan seluler.

Contoh ini menggunakan `LocationTracker` untuk memulai dan menghentikan pelacakan di latar belakang:

```
// For starting the location tracking
locationTracker?.startBackground(
BackgroundTrackingMode.ACTIVE_TRACKING,
object : ServiceCallback {
    override fun serviceStopped() {
        if (selectedTrackingMode == BackgroundTrackingMode.ACTIVE_TRACKING) {
            isLocationTrackingBackgroundActive = false
        } else {
            isLocationTrackingBatteryOptimizeActive = false
        }
    }
},
)

// For stopping the location tracking
```

```
locationTracker?.stopBackgroundService()
```

## SDK Pelacakan Seluler iOS

SDK pelacakan seluler Lokasi Amazon menyediakan utilitas yang membantu dengan mudah mengautentikasi, menangkap posisi perangkat, dan mengirim pembaruan posisi ke Pelacak Lokasi Amazon. SDK mendukung pemfilteran lokal pembaruan lokasi dengan interval pembaruan yang dapat dikonfigurasi. Ini mengurangi biaya data dan mengoptimalkan konektivitas intermiten untuk aplikasi iOS Anda.

SDK pelacakan iOS tersedia di GitHub: [Amazon Location Mobile Tracking SDK for iOS](#).

Bagian ini mencakup topik-topik berikut untuk Amazon Location mobile tracking iOS SDK:

### Topik

- [Penginstalan](#)
- [Penggunaan](#)
- [Filter](#)
- [Fungsi pelacakan SDK Seluler iOS](#)
- [Contoh](#)

### Penginstalan

Gunakan prosedur berikut untuk menginstal SDK pelacakan seluler untuk iOS:

1. Dalam proyek Xcode Anda, buka File dan pilih Add Package Dependencies.
2. Ketik URL berikut: <https://github.com/aws-geospatial/amazon-location-mobile-tracking-sdk-ios/> ke bilah pencarian dan tekan tombol enter.
3. Pilih amazon-location-mobile-tracking-sdk-ios paket dan klik Add Package.
4. Pilih produk AmazonLocationiOSTrackingSDK paket dan klik Add Package.

### Penggunaan

Prosedur berikut menunjukkan cara membuat pembantu otentikasi menggunakan kredensial dari Cognito.

1. Setelah menginstal perpustakaan, Anda perlu menambahkan satu atau kedua deskripsi ke dalam `info.plist` file Anda:

Privacy - Location When In Use Usage Description  
 Privacy - Location Always and When In Use Usage Description

2. Selanjutnya, impor AuthHelper di kelas Anda:

```
import AmazonLocationiOSAuthSDKimport AmazonLocationIOSTrackingSDK
```

3. Kemudian Anda akan membuat AuthHelper objek dan menggunakannya dengan AWS SDK, dengan membuat pembantu otentikasi menggunakan kredensial dari Amazon Cognito.

```
let authHelper = AuthHelper()
let locationCredentialsProvider =
  authHelper.authenticateWithCognitoUserPool(identityPoolId: "My-Cognito-Identity-
  Pool-Id", region: "My-region") //example: us-east-1
let locationTracker = LocationTracker(provider: locationCredentialsProvider,
  trackerName: "My-tracker-name")

// Optionally you can set ClientConfig with your own values in either initialize or
  in a separate function
// let trackerConfig = LocationTrackerConfig(locationFilters:
  [TimeLocationFilter(), DistanceLocationFilter()],
  trackingDistanceInterval: 30,
  trackingTimeInterval: 30,
  logLevel: .debug)

// locationTracker = LocationTracker(provider: credentialsProvider, trackerName:
  "My-tracker-name",config: trackerConfig)
// locationTracker.setConfig(config: trackerConfig)
```

## Filter

SDK iOS pelacakan seluler Lokasi Amazon memiliki tiga filter lokasi bawaan.

- **TimeLocationFilter:** Memfilter lokasi saat ini yang akan diunggah berdasarkan interval waktu yang ditentukan.
- **DistanceLocationFilter:** Memfilter pembaruan lokasi berdasarkan ambang jarak yang ditentukan.

- **AccuracyLocationFilter**: Memfilter pembaruan lokasi dengan membandingkan jarak yang dipindahkan sejak pembaruan terakhir dengan akurasi lokasi saat ini.

Contoh ini menambahkan filter `LocationTracker` pada saat pembuatan:

```
val config = LocationTrackerConfig(  
    trackerName = "MY-TRACKER-NAME",  
    logLevel = TrackingSdkLogLevel.DEBUG,  
    accuracy = Priority.PRIORITY_HIGH_ACCURACY,  
    latency = 1000,  
    frequency = 5000,  
    waitForAccurateLocation = false,  
    minUpdateIntervalMillis = 5000,  
    locationFilters = mutableListOf(TimeLocationFilter(), DistanceLocationFilter(),  
    AccuracyLocationFilter())  
)  
  
locationTracker = LocationTracker(  
    applicationContext,  
    locationCredentialsProvider,  
    config,  
)
```

Contoh ini mengaktifkan dan menonaktifkan filter saat runtime dengan: `LocationTracker`

```
// To enable the filter  
locationTracker?.enableFilter(TimeLocationFilter())  
  
// To disable the filter  
locationTracker?.disableFilter(TimeLocationFilter())
```

## Fungsi pelacakan SDK Seluler iOS

SDK pelacakan seluler Lokasi Amazon untuk iOS mencakup fungsi-fungsi berikut:

- Kelas: `LocationTracker`

```
init(provider: LocationCredentialsProvider, trackerName: String, config:  
LocationTrackerConfig? = nil)
```

Ini adalah fungsi penginisialisasi untuk membuat objek. `LocationTracker` Ini membutuhkan contoh `LocationCredentialsProvider`, `trackerName` dan secara opsional sebuah instance dari `LocationTrackingConfig` Jika konfigurasi tidak disediakan, konfigurasi akan diinisialisasi dengan nilai default.

- Kelas: `LocationTracker`

```
setTrackerConfig(config: LocationTrackerConfig)
```

Ini menetapkan konfigurasi Tracker untuk berlaku kapan saja setelah inisialisasi pelacak lokasi

- Kelas: `LocationTracker`

```
getTrackerConfig()
```

Ini membuat konfigurasi pelacakan lokasi untuk digunakan atau dimodifikasi di aplikasi Anda.

Pengembalian: `LocationTrackerConfig`

- Kelas: `LocationTracker`

```
getDeviceId()
```

Mendapat Id perangkat yang dihasilkan pelacak lokasi.

Pengembalian: `String?`

- Kelas: `LocationTracker`

```
startTracking()
```

Memulai proses mengakses lokasi pengguna dan mengirimkannya ke AWS pelacak.

- Kelas: `LocationTracker`

```
resumeTracking()
```

Melanjutkan proses mengakses lokasi pengguna dan mengirimkannya ke pelacak. AWS

- Kelas: `LocationTracker`

```
stopTracking()
```

Menghentikan proses melacak lokasi pengguna.

- Kelas: `LocationTracker`

```
startBackgroundTracking(mode: BackgroundTrackingMode)
```

Memulai proses mengakses lokasi pengguna dan mengirimkannya ke AWS pelacak saat aplikasi berada di latar belakang. BackgroundTrackingMode memiliki opsi berikut:

- **Active**: Opsi ini tidak secara otomatis menjeda pembaruan lokasi.
- **BatterySaving**: Opsi ini secara otomatis menghentikan pembaruan lokasi
- **None**: Opsi ini secara keseluruhan menonaktifkan pembaruan lokasi latar belakang
- **Kelas: LocationTracker**

```
resumeBackgroundTracking(mode: BackgroundTrackingMode)
```

Melanjutkan proses mengakses lokasi pengguna dan mengirimkannya ke AWS pelacak saat aplikasi berada di latar belakang.

- **Kelas: LocationTracker**

```
stopBackgroundTracking()
```

Menghentikan proses mengakses lokasi pengguna dan mengirimkannya ke AWS pelacak saat aplikasi berada di latar belakang.

- **Kelas: LocationTracker**

```
getTrackerDeviceLocation(nextToken: String?, startTime: Date? = nil,  
endTime: Date? = nil, completion: @escaping (Result<GetLocationResponse,  
Error>)
```

Mengambil lokasi pelacakan yang diunggah untuk perangkat pengguna antara tanggal dan waktu mulai dan berakhir.

Pengembalian: Void

- **Kelas: LocationTrackerConfig**

```
init()
```

Ini menginisialisasi LocationTrackerConfig dengan nilai default.

- **Kelas: LocationTrackerConfig**

```
init(locationFilters: [LocationFilter]? = nil, trackingDistanceInterval:  
Double? = nil, trackingTimeInterval: Double? = nil,
```

```
trackingAccuracyLevel: Double? = nil, uploadFrequency: Double? = nil,  
desiredAccuracy: CLLocationAccuracy? = nil, activityType: CLActivityType?  
= nil, logLevel: LogLevel? = nil)
```

Ini menginisialisasi `LocationTrackerConfig` dengan nilai parameter yang ditentukan pengguna. Jika nilai parameter tidak disediakan itu akan diatur ke nilai default.

- Kelas: `LocationFilter`

```
shouldUpload(currentLocation: LocationEntity, previousLocation:  
LocationEntity?, trackerConfig: LocationTrackerConfig)
```

`LocationFilter` ini adalah protokol yang dapat diterapkan pengguna untuk implementasi filter khusus mereka. Seorang pengguna perlu mengimplementasikan `shouldUpload` fungsi untuk membandingkan lokasi sebelumnya dan saat ini dan kembali jika lokasi saat ini harus diunggah.

## Contoh

Bagian ini merinci contoh penggunaan SDK for iOS Pelacakan Seluler Lokasi Amazon.

### Note

Pastikan bahwa izin yang diperlukan diatur dalam `info.plist` file. Ini adalah izin yang sama yang tercantum di [Penggunaan](#) bagian ini.

Contoh berikut menunjukkan fungsionalitas untuk melacak lokasi perangkat dan mengambil lokasi yang dilacak:

```
Privacy - Location When In Use Usage Description  
Privacy - Location Always and When In Use Usage Description
```

Mulai melacak lokasi:

```
do {  
    try locationTracker.startTracking()  
}  
catch TrackingLocationError.permissionDenied {  
    // Handle permissionDenied by showing the alert message or opening the app  
    settings
```



```
}
```

Lanjutkan melacak lokasi:

```
do {  
    try locationTracker.resumeTracking()  
    }  
catch TrackingLocationError.permissionDenied {  
    // Handle permissionDenied by showing the alert message or opening the app settings  
    }  
}
```

Berhenti melacak lokasi:

```
locationTracker.stopTracking()
```

Mulai pelacakan latar belakang:

```
do {  
  
    locationTracker.startBackgroundTracking(mode: .Active) // .Active, .BatterySaving, .None  
    }  
catch TrackingLocationError.permissionDenied {  
    // Handle permissionDenied by showing the alert message or opening the app settings  
    }  
}
```

Lanjutkan pelacakan latar belakang:

```
do {  
    locationTracker.resumeBackgroundTracking(mode: .Active)  
    }  
catch TrackingLocationError.permissionDenied {  
    // Handle permissionDenied by showing the alert message or opening the app settings  
    }  
}
```

Untuk menghentikan pelacakan latar belakang:

```
locationTracker.stopBackgroundTracking()
```

Ambil lokasi perangkat yang dilacak dari pelacak:

```
func getTrackingPoints(nextToken: String? = nil) {
    let startTime: Date = Date().addingTimeInterval(-86400) // Yesterday's day date and
        time
    let endTime: Date = Date()
    locationTracker.getTrackerDeviceLocation(nextToken: nextToken, startTime: startTime,
        endTime: endTime, completion: { [weak self] result in
        switch result {
        case .success(let response):

            let positions = response.devicePositions
            // You can draw positions on map or use it further as per your requirement

            // If nextToken is available, recursively call to get more data
            if let nextToken = response.nextToken {
                self?.getTrackingPoints(nextToken: nextToken)
            }
        case .failure(let error):
            print(error)
        }
    })
}
```

## API Lokasi Amazon

Amazon Location Service menyediakan operasi API untuk mengakses fungsionalitas lokasi secara terprogram. Ini termasuk API untuk Peta, Tempat, Rute, Pelacak, Geofences, dan menandai sumber daya Anda. Untuk informasi tentang tindakan API yang tersedia, lihat [referensi Amazon Location Service API](#).

Anda dapat menemukan sampel di [Contoh kode](#) bagian panduan ini.

## Menggunakan Lokasi Amazon dengan AWS SDK

AWS kit pengembangan perangkat lunak (SDK) tersedia untuk banyak bahasa pemrograman populer. Setiap SDK menyediakan API, contoh kode, dan dokumentasi yang memudahkan pengembang untuk membangun AWS aplikasi dalam bahasa pilihan mereka.

Untuk informasi selengkapnya tentang SDK yang tersedia untuk digunakan dengan Amazon Location Service menurut bahasa, lihat [SDK menurut bahasa](#) di panduan ini.

### Versi SDK

Sebaiknya gunakan versi SDK terbaru, dan AWS SDK lainnya, yang Anda gunakan dalam proyek, dan untuk memperbarui SDK. AWS SDK memberi Anda fitur dan fungsionalitas terbaru, dan juga pembaruan keamanan. Untuk menemukan versi AWS SDK terbaru JavaScript, misalnya, lihat topik [penginstalan browser](#) di AWS SDK untuk JavaScript dokumentasi.

## Pembaruan pesan kesalahan API Lokasi Amazon

Mulai 1 Agustus 2023, tim Lokasi Amazon mengubah pesan kesalahan API seperti yang dijelaskan dalam tabel berikut. Kode kesalahan tidak akan diubah. Jika aplikasi Anda bergantung pada string pesan kesalahan yang tepat, Anda harus memperbarui aplikasi Anda dengan string baru. Untuk bantuan dengan pertanyaan atau masalah, hubungi AWS Support.

### Topik

- [Tempat](#)
- [Peta](#)
- [Pelacak](#)
- [Rute](#)
- [Metadata](#)
- [Geofences](#)

### Tempat

#### Tempat

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
500	InternalServerErrorException	Internal Server Exception	Internal server error. Try again later.
404	ResourceNotFoundException	resource <PlaceIndexName> not found, reason: <Reason>	Place index not found: <PlaceIndexName>.

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
		Resource '<PlaceIndexName>' not found  placeIdx<PlaceIndexName> not found, reason: <Reason>  no place index with name '%s' found	
404	ResourceNotFoundException	place not found	Place not found: <PlaceId>.
400	ValidationException	PlaceIndex <PlaceIndexName> cannot be used for SearchPlaceIndexForSuggestions because it has IntendedUse <IntendedUse>	A place index with 'IntendedUse' set to Storage does not support 'SearchPlaceIndexForSuggestions' operation.
400	ValidationException	only one of 'BiasPosition' or 'FilterBBox' may be set	Only one of 'BiasPosition' or 'FilterBBox' may be set.
400	ValidationException	BiasPosition must have exactly 2 entries	'BiasPosition' must have exactly 2 entries.
400	ValidationException	BiasPosition[0] must be between -180 and 180	'BiasPosition[0]' must be between -180 and 180.
400	ValidationException	BiasPosition[1] must be between -90 and 90	'BiasPosition[1]' must be between -90 and 90.

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
400	ValidationException	FilterBBox must have exactly 4 entries	'FilterBBox' must have exactly 4 entries.
400	ValidationException	FilterBBox[0] must be between -180 and 180	'FilterBBox[0]' must be between -180 and 180.
400	ValidationException	FilterBBox[1] must be between -90 and 90	'FilterBBox[1]' must be between -90 and 90.
400	ValidationException	FilterBBox[2] must be between -180 and 180	'FilterBBox[2]' must be between -180 and 180.
400	ValidationException	FilterBBox[3] must be between -90 and 90	'FilterBBox[3]' must be between -90 and 90.
400	ValidationException	FilterBBox must have more southwesterly point before more northeasterly point	'FilterBBox' must have more southwesterly position before more northeasterly position.
400	ValidationException	Position must have exactly 2 entries	'Position' must have exactly 2 entries.
400	ValidationException	Position[0] must be between -180 and 180	'Position[0]' must be between -180 and 180.
400	ValidationException	Position[1] must be between -90 and 90	'Position[1]' must be between -90 and 90.

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
400	ValidationException	Language is not a valid BCP 47 language tag	'Language' must comply with the BCP 47 Language Tag standard, but was set to <GivenValue>. For more information, see <a href="https://wikipedia.org/wiki/IETF_language_tag">https://wikipedia.org/wiki/IETF_language_tag</a> .
400	ValidationException	'placeID' is invalid	'PlaceId' must be a valid ID.
400	ValidationException	no customer account ID parameter found	'RequesterAccountID' is a required field.
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.
400	ValidationException	unsupported price plan '<PricingPlan>'	'PricingPlan' must be set to RequestBasedUsage.
400	ValidationException	'DataSource' must be one of: Here, Esri	'DataSource' must be one of Esri, Grab, Here.
400	ValidationException	Grab is only supported in the ap-southeast-1 region	'DataSource' Grab must only be used in following regions: ap-southeast-1.

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
400	ValidationException	'IntendedUse' and 'PricingPlan' must both be provided to update either property	'IntendedUse' and 'PricingPlan' must both be provided to update either attribute .
402	ServiceQuotaExceededException	Place resources per account exceeded quota limits. For more info, see <a href="https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/">https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/</a>	Place index resources have exceeded the quota per account per region. For more information, see <a href="https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/">https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/</a> .
409	ConflictException	Resource already exists	Place index already exists: <PlaceIndexName>.

## Peta

### Peta

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
500	InternalServerError	Internal Server Exception  unable to find style template  Error fetching style	Internal server error. Try again later.

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
		was not able to serialize the map style file	
404	ResourceNotFoundException	Map not found	Map not found: <MapName>.
404	ResourceNotFoundException	Sprites are not supported for this resource	Sprite not found: <SpriteName>.
400	ValidationException	Resource name should be set	'MapName' is a required field.
400	ValidationException	Must provide a valid number for start and end of Range	Font Unicode range start and end numbers must both be provided.
400	ValidationException	Start of range is an invalid number: <StartValue>	Start of font Unicode range must be a valid number.
400	ValidationException	End of range is an invalid number: <StartValue>	End of font Unicode range must be a valid number.
400	ValidationException	End of range must be exactly 255 higher from start of range, difference found: <Difference>	The difference between the start and end of the font Unicode range must be exactly 255. Difference found: <Difference>.



Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
400	ValidationException	Start of range must be a multiple of 256, found <StartValue>	Start of font Unicode range must be a multiple of 256, but was set to: <StartValue>.
400	ValidationException	Request font is empty	'FontStack' is a required field.
400	ValidationException	Request font is not valid for the datasource <DataSource>	<FontStack> is not a supported font stack for data source <DataSource>. For more information about the list of supported font stacks, see <a href="https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapGlyphs.html">https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapGlyphs.html</a> .
400	ValidationException	Request font is not valid	<FontStack> is not a supported font stack for data source <DataSource>. For more information about the list of supported font stacks, see <a href="https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapGlyphs.html">https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapGlyphs.html</a> .

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
400	ValidationException	DataSource is invalid: <DataSource>	'DataSource' must be one of Esri, Grab, Here.
400	ValidationException	Request filename is empty	'FileName' is a required field.
400	ValidationException	Request filename is not valid	<SpriteFile> is not a supported sprite file name. For more information about the list of supported sprite file names, see <a href="https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapSprites.html">https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapSprites.html</a> .
400	ValidationException	Filename is invalid: <FileName>	<SpriteFile> is not a supported sprite file name. For more information about the list of supported sprite file names, see <a href="https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapSprites.html">https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapSprites.html</a> .

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
400	ValidationException	Filename is an invalid content type: <FileName>	<SpriteFile> is not a supported sprite file name. For more information about the list of supported sprite file names, see <a href="https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapSprites.html">https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapSprites.html</a> .
400	ValidationException	Filename is invalid: <FileName>	'Filename' must not be empty.
400	ValidationException	y-coordinate part of 'Y' must be a valid integer	y- coordinate part of 'Y' must be an integer.
400	ValidationException	tile resolution part of 'Y' must be a valid integer followed by 'x'	Tile resolution part of 'Y' must be an integer followed by 'X'.
400	ValidationException	file type extension part of 'Y' must not be empty if a '.' is present	File type extension part of 'Y' must not be empty if a '.' is present.
400	ValidationException	'Z' must be a valid integer	'Z' must be an integer.
400	ValidationException	'X' must be a valid integer	'X' must be an integer.

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
400	ValidationException	'Z' must not be less than minimum zoom of style '<Style>' (<Minimum Value>)	'Z' must not be less than minimum zoom of style <Style> (<MinimumValue>).
400	ValidationException	'Z' must not be greater than maximum zoom of style '<Style>' (<Maximum Value>)	'Z' must not be greater than maximum zoom of style Style (<MaximumValue>).
400	ValidationException	'Z' value not supported	'Z' must be between 0 and 63.
400	ValidationException	tile resolution part of 'Y' must be omitted because '<Style>' is a vector style	Tile resolution part of 'Y' must be omitted for style <Style>.
400	ValidationException	tile resolution part of 'Y' must be at least 1	Tile resolution part of 'Y' must be at least 1.
400	ValidationException	tile resolution part of 'Y' must not be greater than max resolution of style '<Style>' (<Maximum Resolution>)	Tile resolution part of 'Y' must not be greater than maximum resolution of style <Style> (max <MaxResolution>).

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
400	ValidationException	file type extension part of 'Y' must be one of <SupportedFileFormats> (or may be omitted) for style '<Style>'	File type extension part of 'Y' must be one of <SupportedFileFormats> (or may be omitted) for style <Style>.
400	ValidationException	file type extension part of 'Y' must be omitted for style '<Style>'	File type extension part of 'Y' must be omitted for style <Style>.
400	ValidationException	y-coordinate part of 'Y' must be an integer in the range 0..2^Zoom -1 (0..<MaxTileCoordinate>)	y-coordinate part of 'Y' must be an integer in the range 0..2^Zoom -1 (0..<MaxTileCoordinate>).
400	ValidationException	'DataSource' must be one of: Here, Esri	'DataSource' must be one of Esri, Grab, Here.
400	ValidationException	unsupported price plan '<PricingPlan>'	'PricingPlan' must be set to RequestBasedUsage.
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
400	ValidationException	Unsupported Map Style: <Style>	<Style> is not a supported map style. For more information about list of supported map styles, see <a href="https://docs.aws.amazon.com/location/latest/APIReference/API_MapConfiguration.html">https://docs.aws.amazon.com/location/latest/APIReference/API_MapConfiguration.html</a> .
402	ServiceQuotaExceededException	Map resources per account exceeded quota limits. For more info, see <a href="https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/">https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/</a>	Map resources have exceeded the quota per account per region. For more information, see <a href="https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/">https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/</a> .
409	ConflictException	Resource already exists	Map already exists: <MapName>.

## Pelacak

### Pelacak

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
500	InternalServerErrorException	Internal Server Exception	Internal server error. Try again later.

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
		<p>internal server error</p> <p>unable to retrieve point from the storage</p> <p>unable to verify tracker</p> <p>Error processing List request</p>	
404	ResourceNotFoundException	<p>tracker not found: &lt;TrackerName&gt;</p> <p>Tracker with name &lt;TrackerName&gt; was not found</p>	Tracker not found: <TrackerName>.
404	ResourceNotFoundException	<p>association not found: TrackerName &lt;TrackerName&gt;; and ConsumerArn &lt;ConsumerArn &gt;</p>	Association between tracker <TrackerName> and consumer <ConsumerArn> is not found.
400	ValidationException	'ConsumerArn' must refer to a geofence collection resource	'ConsumerArn' must refer to a geofence collection resource.
400	ValidationException	'ConsumerArn' must refer to a resource in the same region as the tracker it is associated to	'ConsumerArn' must refer to a resource in the same region as the tracker it is associated with.

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
400	ValidationException	'ConsumerArn' must refer to a resource in the same AWS account as the tracker is it associated to	'ConsumerArn' must refer to a resource in the same AWS account as the tracker it is associated with.
400	ValidationException	'DataSource' must be one of: Here, Esri	'DataSource' must be one of Esri, Grab, Here.
400	ValidationException	Nothing to update.	At least one of the following fields must be set: 'Description', 'PositionFiltering'
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.
400	ValidationException	request.TrackerName not found on request	'TrackerName ' is a required field.
400	ValidationException	no deviceId parameter found	'DeviceId' is a required field.
400	ValidationException	unsupported price plan '<PricingPlan>'	'PricingPlan' must be set to RequestBasedUsage
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.



Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
400	ValidationException	provided start time is incorrect, should follow the format YYYY-MM-DDThh:mm:ss.sssZ“	'StartTimeInclusive' must follow the format YYYY-MM-DDThh:mm:ss.sssZ.
400	ValidationException	provided end time is incorrect, should follow the format YYYY-MM-DDThh:mm:ss.sssZ	'EndTimeExclusive' must follow the format YYYY-MM-DDThh:mm:ss.sssZ.
400	ValidationException	end time must be after start time	'EndTimeExclusive' must be after 'StartTimeInclusive'.
400	ValidationException	invalid key state	KMS key must be a symmetric Customer Master Key (CMK). Invalid state found. For more information about how key state affects the use of a KMS key, see <a href="https://docs.aws.amazon.com/kms/latest/developerguide/key-state.html">https://docs.aws.amazon.com/kms/latest/developerguide/key-state.html</a> .
400	ValidationException	key not found	Invalid KMS key. '<KmsKeyId>' <KmsKeyIdValue> not found.

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
400	ValidationException	key is disabled	Symmetric Customer Master Key (CMK) must be enabled.
400	ValidationException	access denied	Symmetric Customer Master Key (CMK) must allow Amazon Location to create grants to its KMS key.
402	ServiceQuotaExceededException	Tracker <TrackerName> may not have more than <Max> consumer associations	Tracker resource may not have more than <Max> consumer associations. For more information, see <a href="https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/">https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/</a> .
402	ServiceQuotaExceededException	Trackers per account exceeded quota limits. For more info, see <a href="https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/">https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/</a>	Tracking resources have exceeded the quota per account per region. For more information, see <a href="https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/">https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/</a> .

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
409	ConflictException	association already exists: TrackerName <TrackerName>; and ConsumerArn <ConsumerArn>	An association already exists between tracker <TrackerName> and consumer <ConsumerArn>.
409	ConflictException	Tracker already exists: <TrackerName>	Tracker already exists: <TrackerName>.

## Rute

### Rute

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
500	InternalServerErrorException	Internal Server Exception	Internal server error. Try again later.
404	ResourceNotFoundException	Resource not found	Route calculator not found: <RouteCalculatorName>.
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.
400	ValidationException	'DataSource' must be one of: Here, Esri, Grab	'DataSource' must be one of Esri, Grab, Here.

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
400	ValidationException	<PricingPlan> pricing plan is not supported	'PricingPlan' must be set to RequestBasedUsage
400	ValidationException	unsupported price plan '<PricingPlan>'	'PricingPlan' must be set to RequestBasedUsage
400	ValidationException	Grab is only supported in the ap-southeast-1 region	'DataSource' <DataSourceName> must only be used in following regions: ap-southeast-1.
400	ValidationException	PricingPlan must be 'RequestBasedUsage'	'PricingPlan' must be set to RequestBasedUsage.
400	ValidationException	'DeparturePositions[0][0]' must be between -180 and 180	'DeparturePositions[0][0]' must be between -180 and 180.
400	ValidationException	'DeparturePositions[0][1]' must be between -90 and 90	'DeparturePositions[0][1]' must be between -90 and 90.
400	ValidationException	'DestinationPositions[0][0]' must be between -180 and 180	'DestinationPositions[0][0]' must be between -180 and 180.
400	ValidationException	'DestinationPositions[0][1]' must be between -90 and 90.	'DestinationPositions[0][1]' must be between -90 and 90

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
400	ValidationException	'DepartNow' may not be true if 'DepartureTime' is set	Only one of 'DepartNow' or 'DepartureTime' may be set.
400	ValidationException	'<TravelModeOption>' may not be set when 'TravelMode' has value <TravelModeOption>	'<TravelModeOption>' must not be set when 'TravelMode' has value <TravelModeOption>.
400	ValidationException	'CarModeOptions' may not be set when 'TravelMode' has value Walking	'CarModeOptions' must not be set when 'TravelMode' has value Walking.
400	ValidationException	'TruckModeOptions' may not be set when 'TravelMode' has value Walking	'TruckModeOptions' must not be set when 'TravelMode' has value Walking.
400	ValidationException	'TruckModeOptions' may not be set when 'TravelMode' has value Car	'TruckModeOptions' must not be set when 'TravelMode' has value Car.
400	ValidationException	'CarModeOptions' may not be set when 'TravelMode' has value Truck	'CarModeOptions' must not be set when 'TravelMode' has value Truck.

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
400	ValidationException	At least one of [Height, Length, Width] must be set in 'TruckModeOptions.Dimensions'	At least one of the following attribute must be set in TruckModeOptions.Dimensions: Height, Length, Width.
400	ValidationException	At least one of [Total] must be set in 'TruckModeOptions.Weight'	At least one of the following attribute must be set in TruckModeOptions.Weight: Total.
400	ValidationException	'DeparturePositions' count must be 10 or less with DataSource set to Esri	'DeparturePositions' must have length at most 10 for 'DataSource' Esri.
400	ValidationException	'DestinationPositions' count must be 10 or less with DataSource set to Esri	'DestinationPositions' must have length at most 10 for 'DataSource' Esri.
400	ValidationException	'DeparturePositions[0]' is more than 40km away from 'DestinationPositions[0]'	'DeparturePositions[0]' must not be more than 40 km away from 'DestinationPositions[0]'.
400	ValidationException	'DeparturePositions[0]' is more than 400km away from 'DestinationPositions[0]'	'DeparturePositions[0]' must not be more than 400 km away from 'DestinationPositions[0]'.

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
400	ValidationException	DeparturePositions [0] is contained within an unsupported region. Korea is not supported for CalculateRouteMatrix with the provider Esri.	DeparturePositions [0] is located in Korea, which is not supported when using CalculateRouteMatrix with data provider Esri.
400	ValidationException	'<HereTruckDimension>' must be between <Min> and <Max> <Unit>	'HereTruckDimension' must be between <Min> and <Max> <Unit>.
400	ValidationException	'WaypointPositions[0][0]' must be between -180 and 180	'WaypointPositions[0][0]' must be between -180 and 180.
400	ValidationException	'WaypointPositions[0][1]' must be between -90 and 90	'WaypointPositions[0][1]' must be between -90 and 90.
400	ValidationException	'WaypointPositions[1][0]' must be between -180 and 180	'WaypointPositions[1][0]' must be between -180 and 180.
400	ValidationException	'WaypointPositions[1][1]' must be between -90 and 90	'WaypointPositions[1][1]' must be between -90 and 90.
400	ValidationException	No road segment could be matched for one or more coordinates within a radius (1km)	One or more provided positions are more than 1 km from the nearest road segment.

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
400	ValidationException	Some positions in the request are unreachable	Some positions in the request are unreachable.
400	ValidationException	Total distance between all waypoints must be not be greater than 40km for DataSource Esri when using TravelMode Walking	Total distance between all route positions must not be greater than 40 km for 'DataSource' Esri and 'TravelMode' Walking.
400	ValidationException	Total distance between all waypoints must be not be greater than 400km for DataSource Esri	Total distance between all route positions must not be greater than 400 km for 'DataSource' Esri.
400	ValidationException	Following positions in the request are unreachable: <UnreachablePositions>	The following positions are unreachable: <UnreachablePositions>.
400	ValidationException	'DepartureTime' contains a badly-formatted timestamp	'DepartureTime' must follow the format YYYY-MM-DDThh:mm:ss.sssZ.
400	ValidationException	'TravelMode' <TravelMode> is not supported by <DataProvider>	'TravelMode' <TravelMode> not supported by data provider <DataProvider>.



Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
400	ValidationException	'DeparturePositions' must be set	'DeparturePositions' must not be empty.
400	ValidationException	'DestinationPositions' must be set	'DestinationPositions' must not be empty.
400	ValidationException	Some inputs in the request are invalid	Some inputs in the request are invalid.
400	ValidationException	No route found between position <FirstPosition> and position <SecondPosition>	No route found between position <FirstPosition> and position <SecondPosition>.
400	ValidationException	No route found	No route found. For more information, see <a href="https://developer.amazon.com/documentation/routing-api/dev_guide/topics/no_tice.html">https://developer.amazon.com/documentation/routing-api/dev_guide/topics/no_tice.html</a> .
400	ValidationException	No route found	No route found.

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
402	ServiceQuotaExceededException	Route calculators per account exceeded quota limits. For more info, see <a href="https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/">https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/</a>	Route calculator resources have exceeded the quota per account per region. For more information, see <a href="https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/">https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/</a> .
409	ConflictException	Resource already exists	Route calculator already exists: <RouteCalculatorName>.

## Metadata

### Metadata

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
500	InternalServerErrorException	Internal Server Error Error processing List request	Internal server error. Try again later.
404	ResourceNotFoundException	APIKey not found	Api key not found: <APIKeyName>.
404	ResourceNotFoundException	APIKeyID not found	ApiKeyId not found: <APIKeyID>.

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
400	ValidationException	Either ExpireTime or NoExpiry must be provided	At least one of the following fields must be set: 'ExpireTime', 'NoExpiry'.
400	ValidationException	NoExpiry cannot be set to false if no ExpireTime is provided	'ExpireTime' must be set when 'NoExpiry' has value false.
400	ValidationException	ExpireTime cannot be set if NoExpiry is true	'ExpireTime' must not be set when 'NoExpiry' has value true.
400	ValidationException	Expire time '<ExpireTimeValue>' is not a valid time format	'ExpireTime' must follow the format YYYY-MM-DDThh:mm:ss.sssZ.
400	ValidationException	Expire time '<ExpireTimeValue>' cannot be in the past when creating a key	'ExpireTime' must not be in the past.
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
400	ValidationException	The API Key %s has been recently used and the requested update may impact current usage. Specify ForceUpdate=true to update the API Key configuration.	This update may cause some users to lose API access. Because this API Key has been used in the last 7 days, you must set 'ForceUpdate' to true to confirm this change.
400	ValidationException	Expire time '<ExpireTimeValue>' must not be more than 1 minute in the past	'ExpireTime' must not be more than 1 minute in the past.
400	ValidationException	Description, ExpireTime, NoExpiry and Restrictions can't all be empty	At least one of the following fields must be set: 'Description', 'ExpireTime', 'NoExpiry', 'Restrictions'.
400	ValidationException	API Key expired	'ApiKeyId' must not be expired.
409	ConflictException	API key named <APIKeyName> already exists	Api key already exists: <APIKeyName>.

## Geofences

### Geofences

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
500	InternalServerErrorException	<p>internal server error</p> <p>Internal server error</p> <p>Unsupported geofence geometry encountered</p> <p>geometry marshal error</p> <p>geometry load error</p> <p>unable to get geofence collection</p> <p>unable to delete geofences</p> <p>unable to retrieve geofence</p> <p>Error processing List request</p>	<p>Internal server error.</p> <p>Try again later.</p>
404	ResourceNotFoundException	<p>collection not found: &lt;GeofenceCollectionName&gt;</p> <p>&lt;GeofenceCollectionName&gt; geofence collection not found</p>	<p>Geofence Collection not found: &lt;GeofenceCollectionName&gt;.</p>

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
		Resource not found error  no geofence with given name found	
400	ValidationException	unsupported price plan '<PricingPlan>'	'PricingPlan' must be set to RequestBasedUsage.
400	ValidationException	KMS key must be a symmetric CMK. Invalid usage type: <UsageType>	KMS key must be a symmetric Customer Master Key (CMK). Invalid usage type <UsageType>. For how to create a symmetric CMK, refer to <a href="https://docs.aws.amazon.com/kms/latest/developerguide/create-keys.html#create-symmetric-cmk">https://docs.aws.amazon.com/kms/latest/developerguide/create-keys.html#create-symmetric-cmk</a> .
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.
400	ValidationException	PricingPlanDataSource cannot be updated without updating PricingPlan	'PricingPlan' must be provided to update 'PricingPlanDataSource'.

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
400	ValidationException	nothing to update	At least one of the following fields must be set: 'Description'
400	ValidationException	invalid key state	KMS key must be a symmetric Customer Master Key (CMK). Invalid state <InvalidState>. For more information about how key state affects the use of a KMS key, see <a href="https://docs.aws.amazon.com/kms/latest/developerguide/key-state.html">https://docs.aws.amazon.com/kms/latest/developerguide/key-state.html</a> .
400	ValidationException	key not found	Invalid KMS key. '<KmsKeyId>' <KmsKeyIdValue> not found.
400	ValidationException	key is disabled	Symmetric Customer Master Key (CMK) must be enabled.
400	ValidationException	access denied	Symmetric Customer Master Key (CMK) must allow Amazon Location to create grants to its KMS key.

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
400	ValidationException	duplicate geofence ID in batch	'GeofenceId' <DuplicatedGeofenceId> is duplicated in batch.
400	ValidationException	missing GeofenceId	'GeofenceId' must not be empty.
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.
400	ValidationException	Position[0] must be between -180 and 180	'Position[0]' must be between -180 and 180.
400	ValidationException	Position[1] must be between -90 and 90	'Position[1]' must be between -90 and 90.
400	ValidationException	radius must be less than or equal to 1000km	'Geometry.Circle.Radius' must be less than or equal to 1000km.
400	ValidationException	no geofence with given name found	Geofence not found: <CollectionName>.
400	ValidationException	Geometry must contain either a Circle or Polygon, not both	Only one of 'Circle' or 'Polygon' may be set within 'Geometry'.
400	ValidationException	Geometry must contain a Polygon or a Circle	One of 'Polygon' or 'Circle' must be set within 'Geometry'.



Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
400	ValidationException	radius must be greater than 0m	'Geometry.Circle.Radius' must be greater than 0m.
400	ValidationException	empty polygon	'Geometry.Polygon' must not be empty.
400	ValidationException	empty polygon ring	'Geometry.Polygon' must not be empty.
400	ValidationException	circle can not cross antimeridian	'Geometry.Circle' must not cross antimeridian. Cut it in two such that neither part's representation crosses the antimeridian.
400	ValidationException	polygon can not cross antimeridian	'Geometry.Polygon' must not cross antimeridian. Cut it in two such that neither part's representation crosses the antimeridian.
400	ValidationException	polygon can not have interior rings (holes), remove holes	'Geometry.Polygon' must not have interior rings (holes). For more information about interior rings see <a href="https://www.rfc-editor.org/rfc/rfc7946.html#appendix-A.3">https://www.rfc-editor.org/rfc/rfc7946.html#appendix-A.3</a> .

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
400	ValidationException	polygon ring is not closed	'Geometry.Polygon' contains an open ring. Close the ring by ensuring the first and last positions are equal.
400	ValidationException	polygon ring has more than 1000 vertices	'Geometry.Polygon' must not have more than 1000 vertices.
400	ValidationException	polygon ring has fewer than 4 positions	Number of vertices in 'Geometry.Polygon' must be greater or equal to 4.
400	ValidationException	invalid center	'Geometry.Circle.Center' must be a valid position (longitude/latitude pair).
400	ValidationException	radius must be greater than 0m	'Geometry.Circle.Radius' must be greater than 0 m.
400	ValidationException	longitude range should be between -180 and 180 degrees	Longitude must be between -180 and 180 degrees, but was set to <Provided Longitude>.

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
400	ValidationException	latitude range should be between -90 and 90 degrees	Latitude must be between -90 and 90 degrees, but was set to <Provided Longitude>.
400	ValidationException	polygon exterior ring is expected to be counter clockwise	'Geometry.Polygon' must be oriented counter-clockwise.
400	ValidationException	polygon interior ring should be clockwise oriented	'Geometry.Polygon' must be oriented clockwise.
400	ValidationException	radius must be less than or equal to 1000km	'Geometry.Circle.Radius' must be less than or equal to 1000 km.
400	ValidationException	timestamp.Parse() error	'SampleTime' must follow the format YYYY-MM-DDThh:mm:ss.sssZ.
400	ValidationException	invalid input	'SourceArn' must refer to a tracker resource.
400	ValidationException	arn: invalid prefix	'SourceArn' must be a valid ARN. For more information, see <a href="https://docs.aws.amazon.com/general/latest/gr/AWS-arns-and-namespaces.html">https://docs.aws.amazon.com/general/latest/gr/AWS-arns-and-namespaces.html</a> .

Kode kesalahan	Pengecualian	Pesan kesalahan lama	Pesan kesalahan baru
400	ValidationException	arn: not enough sections	'SourceArn' must be a valid ARN. For more information, see <a href="https://docs.aws.amazon.com/general/latest/gr/AWS-arns-and-namespaces.html">https://docs.aws.amazon.com/general/latest/gr/AWS-arns-and-namespaces.html</a> .
400	ValidationException	invalid resource part	'SourceArn' must refer to a tracker resource.
402	ServiceQuotaExceededException	Geofence collections per account exceeded quota limits. For more info, see <a href="https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/">https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/</a>	Geofence collection resources have exceeded the quota per account per region. For more information, see <a href="https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/">https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/</a> .
409	ConflictException	collection already exists: <Geofence CollectionName>	Geofence Collection already exists: <GeofenceCollectionName>.
409	ConflictException	Resource conflict error	Geofence already exists: <Geofence Name>.

# Contoh kode dan tutorial untuk bekerja dengan Amazon Location Service

Topik ini menampilkan daftar contoh kode, tutorial, dan posting blog untuk membantu Anda mempelajari Amazon Location Service. Setiap contoh kode menyertakan deskripsi tentang cara kerjanya.

Anda dapat menemukan sampel tambahan di [GitHub halaman AWS Geospasial](#), [GitHub halaman AWS sampel untuk Lokasi Amazon](#), dan di [situs AWS blog](#).

## Note

Adalah baik untuk memahami perbedaan antara halaman AWS Geospasial dan GitHub halaman AWS sampel.

- Geospasial GitHub - [GitHub Halaman AWS Geospasial](#) mencakup sampel yang dibuat dan dikelola oleh tim Amazon Location Service.
- Sampel GitHub — [GitHub Halaman AWS sampel untuk Lokasi Amazon](#) mencakup sampel yang dibuat untuk Lokasi Amazon, tetapi mungkin atau mungkin tidak dipelihara secara aktif.

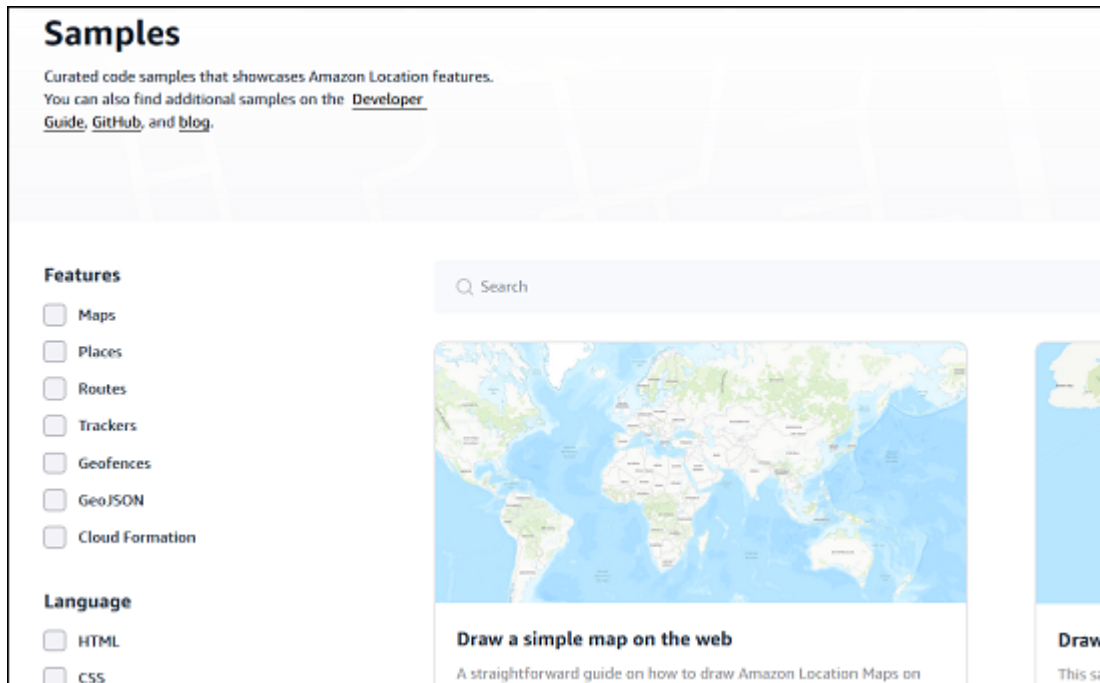
Tutorial [mulai cepat](#) adalah tempat yang baik untuk memulai sebelum menggunakan sampel lain, karena ini menunjukkan cara menyelesaikan prasyarat yang berguna untuk sebagian besar sampel.

## Topik

- [Situs Demo Lokasi Amazon](#)
- [Tutorial: Mulai cepat](#)
- [Tutorial: Pengayaan basis data](#)
- [Contoh: Jelajahi aplikasi](#)
- [Contoh: Gaya peta](#)
- [Contoh: Gambar spidol](#)
- [Contoh: Gambar poin berkerumun](#)
- [Contoh: Menggambar poligon](#)
- [Contoh: Ubah bahasa peta](#)

- [Blog: Perkiraan pemberitahuan waktu pengiriman](#)
- [Contoh: Pembaruan Posisi Stream](#)
- [Contoh: Aplikasi seluler Geofencing dan Tracking](#)

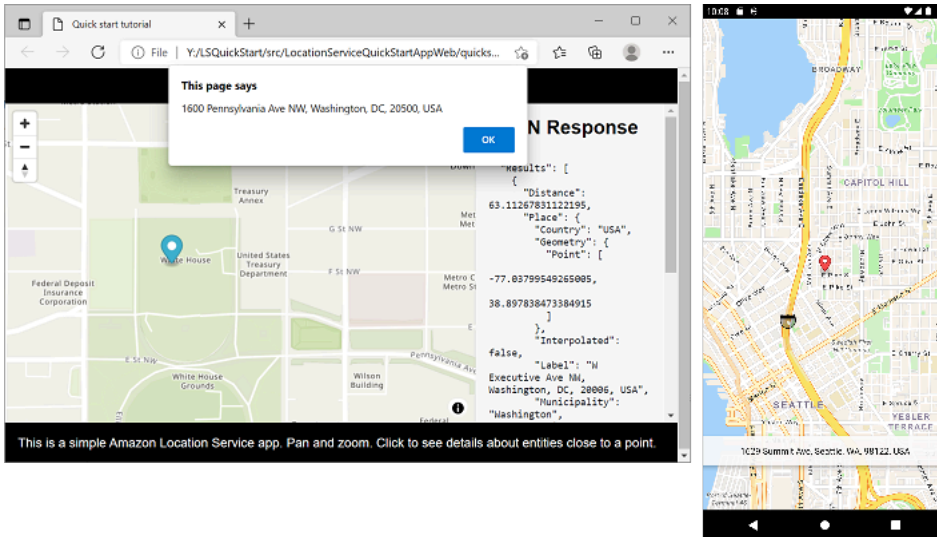
## Situs Demo Lokasi Amazon



Anda dapat melihat demo dengan kode sumber Amazon Location Service beraksi di [situs Amazon Location Demo](#). Situs ini mencakup [demo web yang dihosting](#), dan juga aplikasi demo untuk [Android](#).

Anda juga dapat menemukan beragam sampel, dapat disaring berdasarkan fitur, bahasa, dan platform di halaman [Sampel](#) situs.

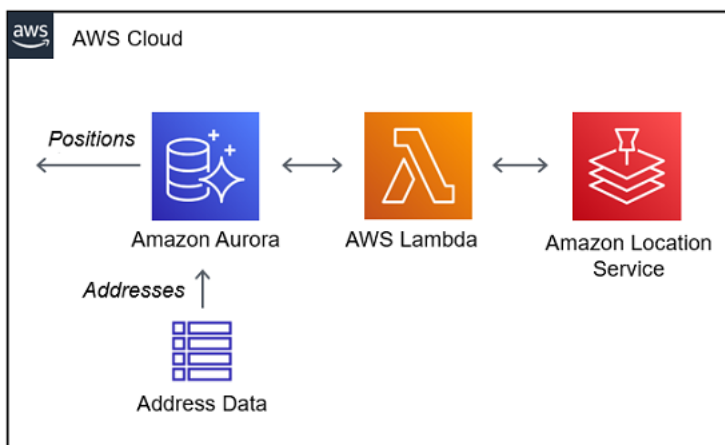
## Tutorial: Mulai cepat



Ada tutorial mulai cepat yang tersedia untuk perangkat web, iOS, dan Android. Untuk setiap platform, tutorial menunjukkan cara menambahkan peta interaktif ke aplikasi, dan cara melakukan panggilan ke Amazon Location Service API dari aplikasi Anda. Tutorial ini tersedia JavaScript di halaman web statis, Kotlin untuk aplikasi ponsel Android, atau Swift untuk aplikasi iOS.

- JavaScript untuk tautan dokumentasi halaman web statis: [Membuat aplikasi web](#)
- Kotlin untuk tautan dokumentasi aplikasi Android: [Mulai cepat dengan Amazon Location Service](#)
- Swift untuk tautan dokumentasi aplikasi iOS: [Membuat aplikasi iOS](#)

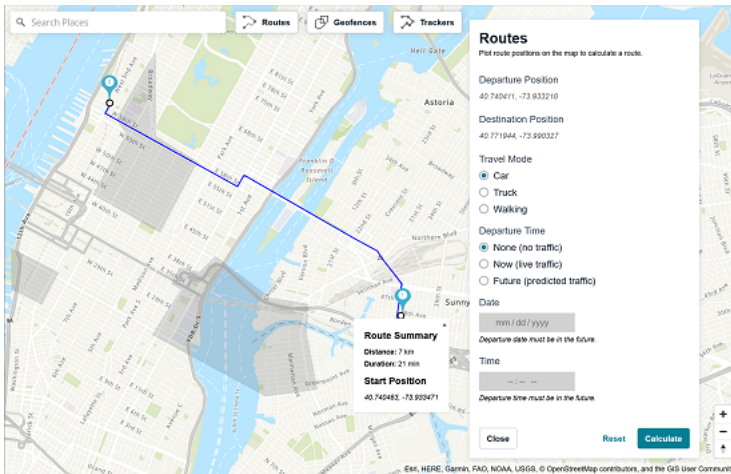
## Tutorial: Pengayaan basis data



Tutorial ini menunjukkan cara menggunakan Amazon Location Service, dipanggil dari AWS Lambda untuk menormalkan alamat dan menambahkan garis lintang dan bujur ke catatan dalam database Amazon Aurora. Menggunakan Amazon Aurora dan AWS Lambda

Tautan dokumentasi: [Amazon Aurora PostgreSQL fungsi yang ditentukan pengguna untuk Amazon Location Service](#)

## Contoh: Jelajahi aplikasi

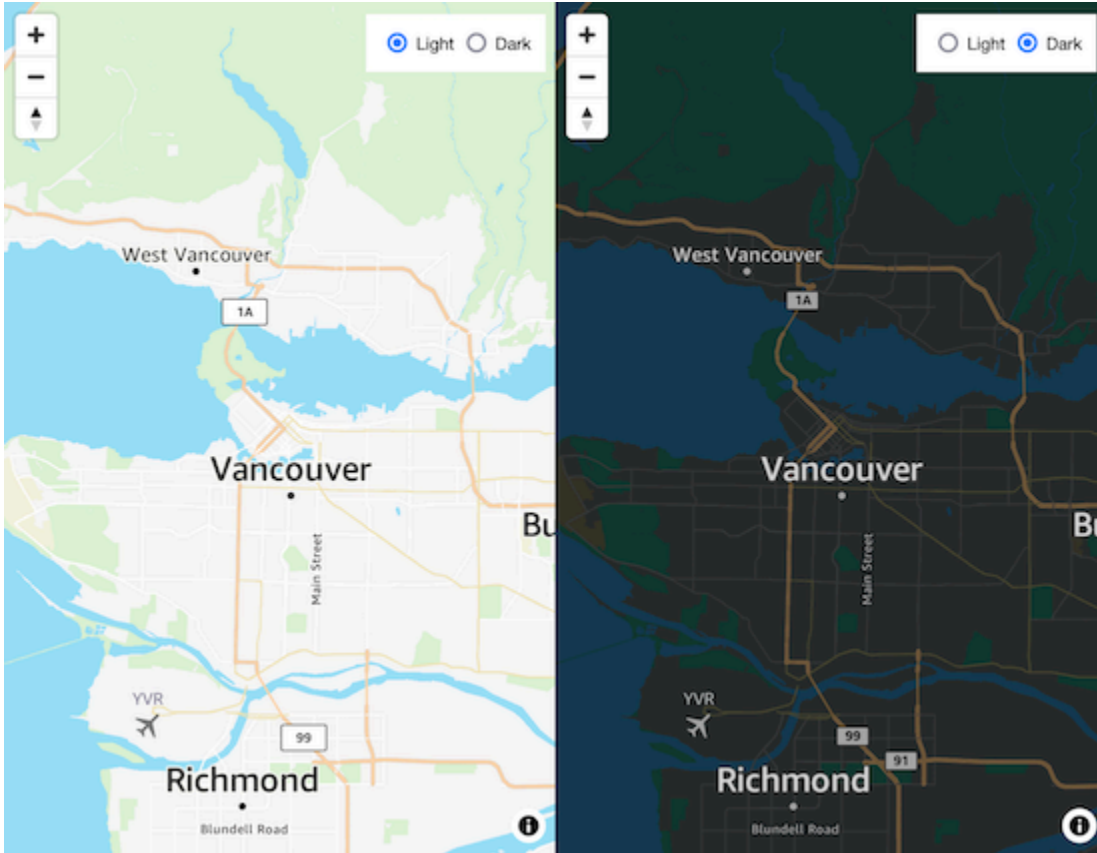


Salah satu cara terbaik untuk mempelajari fungsionalitas Amazon Location Service adalah dengan menggunakan [fungsionalitas Jelajahi](#) dalam konsol Lokasi Amazon. Contoh aplikasi web lengkap ini meniru fungsionalitas peta, tempat, rute, geofences, dan pelacak dari konsol untuk menunjukkan cara membuat ulang fitur ini di aplikasi Anda sendiri. Menggunakan Amplify, React, dan JavaScript

GitHub Tautan sampel: [Jelajahi aplikasi sampel](#)



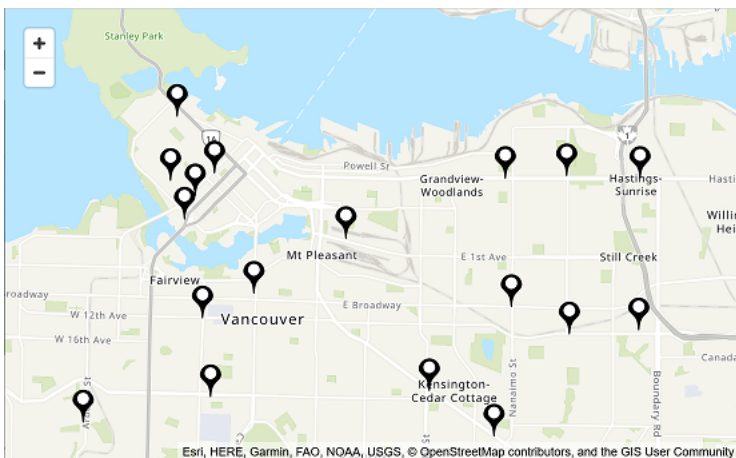
## Contoh: Gaya peta



Contoh kode ini menunjukkan cara beralih antara peta satelit dan peta jalan vektor, menggunakan MapLibre in JavaScript. Penggunaan MapLibre, pembantu otentikasi Lokasi Amazon, dan JavaScript

GitHub Tautan geospasial: [Peta interaktif dengan peralihan gaya](#)

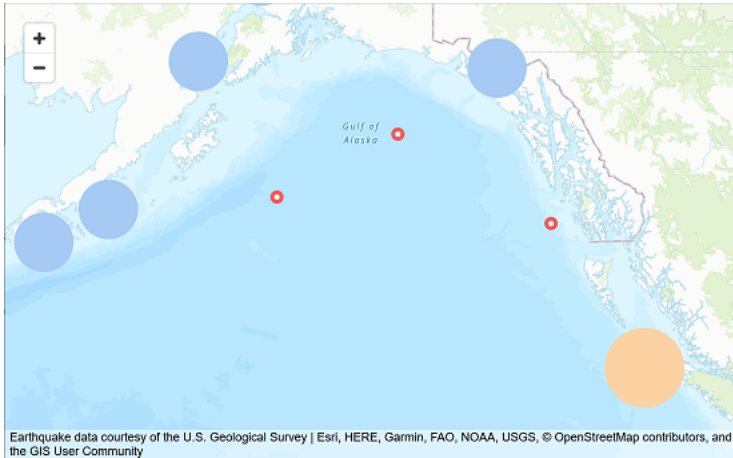
## Contoh: Gambar spidol



Contoh kode ini menunjukkan lokasi Amazon Locker di Vancouver, BC, Kanada. Ini menunjukkan cara menggambar spidol di lokasi titik. Menggunakan MapLibre, Node.js, React, pembantu otentikasi Lokasi Amazon dan. JavaScript

GitHub Tautan geospasial: [Peta interaktif dengan penanda](#) di titik-titik

## Contoh: Gambar poin berkerumun



Menggunakan data gempa USGS, contoh kode ini menunjukkan cara menggambar titik-titik yang mengelompok bersama ketika mereka berdekatan di peta. Menggunakan MapLibre, Node.js, React, Amplify, dan. JavaScript

GitHub Tautan sampel: [Peta interaktif dengan kelompok titik](#)

## Contoh: Menggambar poligon



Contoh kode ini menunjukkan cara menggambar poligon di peta. Menggunakan MapLibre, Node.js, React, pembantu otentikasi Lokasi Amazon, dan. JavaScript

GitHub Tautan geospasial: [Peta interaktif dengan poligon](#)

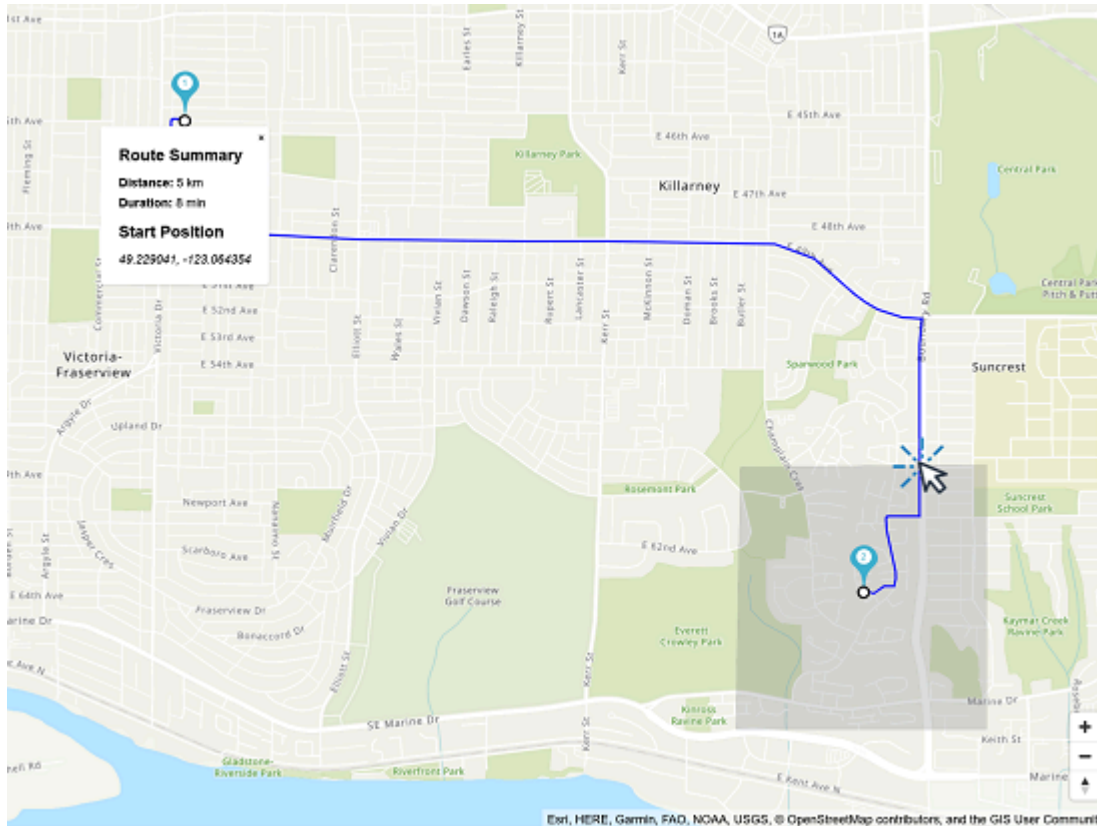
## Contoh: Ubah bahasa peta



Contoh kode ini menunjukkan bagaimana Anda dapat mengubah bahasa tampilan peta di Lokasi Amazon. Menggunakan Amplify, React, dan MapLibre

GitHub Tautan sampel: [Ubah Contoh Bahasa Peta](#)

## Blog: Perkiraan pemberitahuan waktu pengiriman



Posting blog ini menunjukkan berbagai cara untuk memberi tahu pelanggan dengan perkiraan waktu pengiriman. Ini menjelaskan penggunaan rute untuk menunjukkan perkiraan waktu mengemudi, dan kemudian menggunakan pelacak dan geofences untuk memberi tahu ketika pengemudi mendekati pelanggan. Menggunakan Amplify, React, Amazon EventBridge, dan Amazon Simple Notification Service (Amazon SNS).

Tautan blog: [Perkiraan Waktu Kedatangan dan Pemberitahuan Kedekatan](#)

## Contoh: Pembaruan Posisi Stream



Kinesis Stream To Tracker App: Contoh ini menunjukkan cara menggunakan Kinesis Data Stream untuk memposting pembaruan pelacak dengan Amazon Location Service. Sampel adalah aplikasi lambda deployable yang ditulis dengan python yang dapat diintegrasikan dengan Kinesis Data Stream untuk mengkonsumsi peristiwa Kinesis dan posisi perangkat pembaruan batch.

Tautan repositori: [Amazon Kinesis Data Streams Aliran Lokasi Amazon Ke Aplikasi Pelacak](#)

Untuk informasi selengkapnya tentang tracking dan geofences, lihat dokumentasi [Geofences](#) and [Trackers](#). [Pengembang dapat menerapkan aplikasi dengan mengikuti dokumentasi Serverless Application Repository AWS](#), atau secara terpisah dari konsol Lambda.

Aplikasi Contoh Streaming Posisi Perangkat: Contoh kode ini menunjukkan cara mengalirkan data posisi perangkat ke Aliran Data Kinesis dan cara kerja notifikasi geofence. Aplikasi ini bergantung pada Kinesis Stream to Tracker Sample App, yang tercantum di atas, untuk berjalan agar posisi pelacak streaming diperbarui di Amazon Location Service.

Tautan repositori: Aplikasi Contoh [Streaming Posisi Perangkat Lokasi Amazon](#)

## Contoh: Aplikasi seluler Geofencing dan Tracking

Contoh aplikasi ini menunjukkan bagaimana pelacak dan geofence berinteraksi menggunakan kombinasi fitur Lambda, AWS IoT dan Amazon Location. Ada tutorial yang tersedia untuk iOS dan Android.

Tautan tutorial: [Contoh aplikasi seluler Geofence dan Tracker](#)

# Cara menggunakan Amazon Location Service

Anda dapat menggunakan kemampuan Amazon Location Service untuk menyelesaikan tugas terkait geografis dan lokasi. Anda kemudian dapat menggabungkan tugas-tugas ini untuk mengatasi kasus penggunaan yang lebih kompleks seperti geomarketing, pengiriman, dan pelacakan aset.

Saat Anda siap untuk membangun fitur lokasi ke dalam aplikasi Anda, gunakan metode berikut untuk menggunakan fungsionalitas Amazon Location Service, tergantung pada sasaran dan kecenderungan Anda:

- **Alat eksplorasi** - Jika Anda ingin bereksperimen dengan sumber daya Lokasi Amazon, alat berikut adalah cara tercepat untuk mengakses dan mencoba API:
  - [Konsol Lokasi Amazon](#) menyediakan berbagai alat akses cepat. Anda dapat membuat dan mengelola sumber daya dan mencoba API menggunakan [halaman Jelajahi](#). Konsol ini juga berguna untuk membuat sumber daya (biasanya tugas satu kali) dalam persiapan untuk menggunakan salah satu metode lain yang dijelaskan nanti.
  - [AWSCommand Line Interface](#) (CLI) memungkinkan Anda membuat sumber daya dan mengakses Amazon Location API menggunakan terminal. AWS CLIMenangani otentikasi saat Anda mengonfigurasinya dengan kredensi Anda.
  - Anda dapat melihat [contoh kode dan tutorial](#) yang menunjukkan cara melakukan tugas menggunakan Amazon Location Service API. Ini termasuk [contoh](#) yang meniru sebagian besar fungsionalitas halaman Jelajahi di konsol.
- **Platform SDK** — Jika Anda tidak memvisualisasikan data pada peta, Anda dapat menggunakan salah satu [alat AWS standar](#) untuk membangun. AWS
  - SDK berikut tersedia: C ++, Go, Java,, .NET JavaScript, Node.js, PHP, Python, dan Ruby.
- **Frontend SDK dan library** - Jika Anda ingin menggunakan Amazon Location untuk membangun aplikasi di platform seluler atau memvisualisasikan data pada peta di platform apa pun, Anda memiliki opsi berikut:
  - AWS AmplifyPustaka mengintegrasikan Lokasi Amazon dalam aplikasi [iOS](#), [Android](#), dan [JavaScriptweb](#).
  - MapLibre Library memungkinkan Anda merender peta sisi klien ke [iOS](#), [Android](#), dan aplikasi [JavaScriptweb](#) menggunakan Amazon Location.
  - [Pustaka Tangram ES memungkinkan Anda merender peta 2D dan 3D dari data vektor menggunakan OpenGL ES dalam aplikasi web iOS dan Android](#). Ada juga Tangram untuk aplikasi [JavaScriptweb](#).

- Mengirim permintaan HTTPS langsung - Jika Anda bekerja dengan bahasa pemrograman yang tidak tersedia SDK, atau jika Anda ingin lebih mengontrol cara Anda mengirim permintaan AWS, Anda dapat mengakses Lokasi Amazon dengan mengirimkan permintaan HTTPS langsung yang diautentikasi oleh proses penandatanganan Versi Tanda Tangan 4. Untuk informasi selengkapnya tentang [proses penandatanganan Signature Version 4](#), lihat Referensi Umum AWS.

Bab ini menjelaskan banyak tugas yang umum untuk aplikasi menggunakan data lokasi. Bagian [kasus penggunaan umum](#) menjelaskan cara menggabungkannya dengan AWS layanan lain untuk mencapai kasus penggunaan yang lebih kompleks.

## Topik

- [Prasyarat untuk menggunakan Amazon Location Service](#)
- [Menggunakan Peta Lokasi Amazon di aplikasi Anda](#)
- [Mencari data tempat dan geolokasi menggunakan Amazon Location](#)
- [Menghitung rute menggunakan Amazon Location Service](#)
- [Geofencing area yang diminati menggunakan Lokasi Amazon](#)
- [Menandai sumber daya Amazon Location Service](#)
- [Memberikan akses ke Amazon Location Service](#)
- [Memantau Amazon Location Service](#)
- [Membuat sumber daya Amazon Location Service dengan AWS CloudFormation](#)

## Prasyarat untuk menggunakan Amazon Location Service

Bagian ini menjelaskan apa yang perlu Anda lakukan untuk menggunakan Amazon Location Service. Anda harus memiliki Akun AWS dan telah mengatur akses ke Lokasi Amazon untuk pengguna yang ingin menggunakannya.

## 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/signup>.
2. Ikuti petunjuk online.

Bagian dari prosedur pendaftaran melibatkan tindakan menerima panggilan telepon dan memasukkan kode verifikasi di keypad telepon.

Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWS dibuat. 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 mengirim Anda email konfirmasi setelah proses pendaftaran selesai. Anda dapat melihat aktivitas akun Anda saat ini dan mengelola akun Anda dengan mengunjungi <https://aws.amazon.com/> 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 Console](#) sebagai 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.

## Berikan akses ke Amazon Location Service

Pengguna non-admin Anda tidak memiliki izin secara default. Sebelum mereka dapat mengakses Lokasi Amazon, Anda harus memberikan izin dengan melampirkan kebijakan IAM dengan izin tertentu. Pastikan untuk mengikuti prinsip hak istimewa paling sedikit saat memberikan akses ke sumber daya.

### Note

Untuk informasi tentang memberikan pengguna yang tidak diautentikasi akses ke fungsionalitas Amazon Location Service (misalnya, dalam aplikasi berbasis web), lihat [Memberikan akses ke Amazon Location Service](#)

Kebijakan contoh berikut memberikan izin pengguna untuk mengakses semua operasi Lokasi Amazon. Untuk contoh lainnya, lihat [Contoh kebijakan berbasis identitas untuk Amazon Location Service](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "geo:*"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}
```

Untuk memberikan akses, menambahkan izin ke pengguna, grup, atau peran Anda:

- Pengguna dan grup di AWS IAM Identity Center:

Buat rangkaian izin. Ikuti instruksi di [Buat rangkaian izin](#) di Panduan Pengguna AWS IAM Identity Center .

- Pengguna yang dikelola di IAM melalui penyedia identitas:

Buat peran untuk federasi identitas. Ikuti instruksi dalam [Membuat peran untuk penyedia identitas pihak ketiga \(federasi\)](#) di Panduan Pengguna IAM.

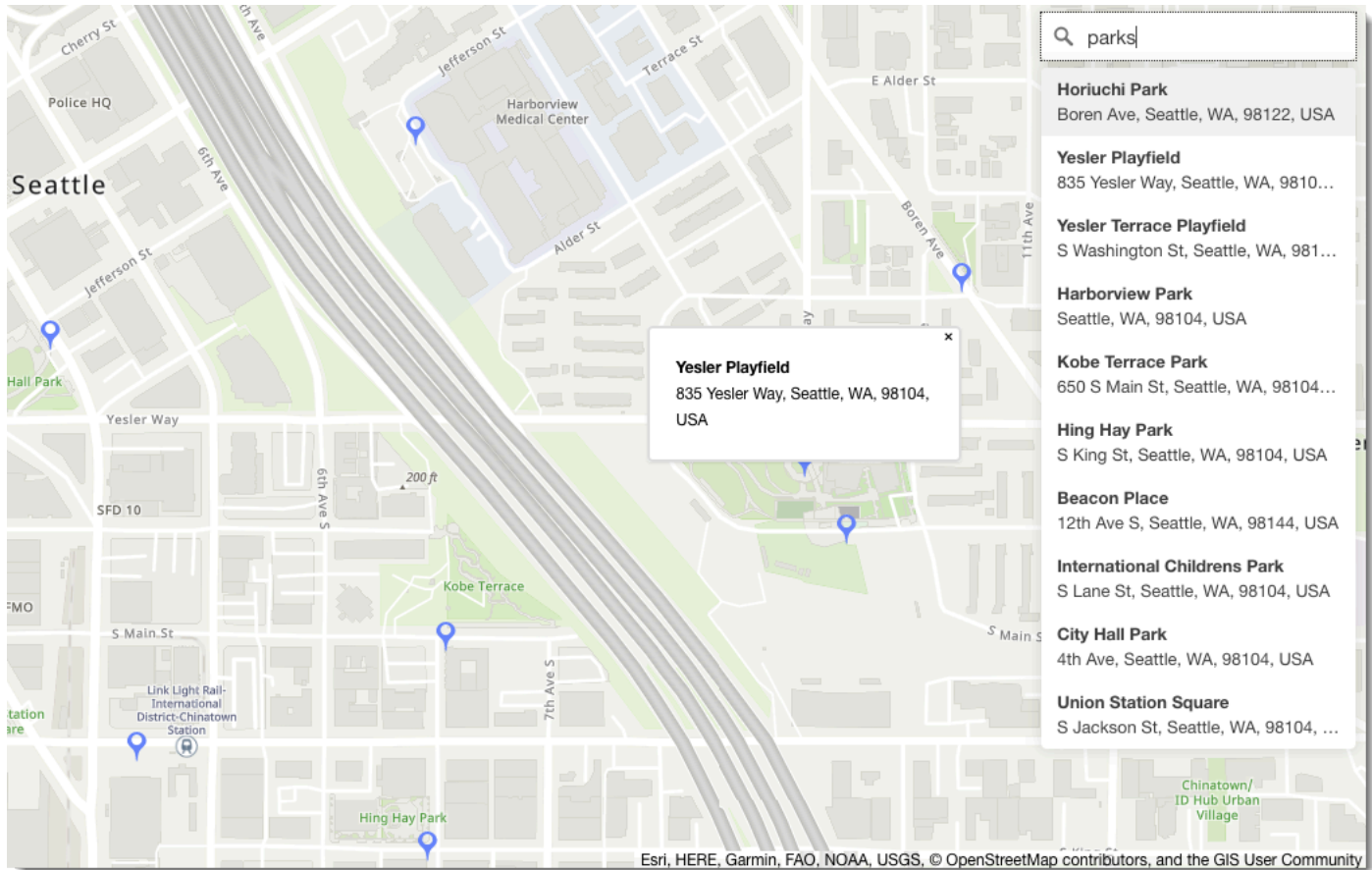
- Pengguna IAM:

- Buat peran yang dapat diambil pengguna Anda. Ikuti instruksi dalam [Membuat peran untuk pengguna IAM](#) dalam Panduan Pengguna IAM.
- (Tidak disarankan) Pasang kebijakan langsung ke pengguna atau tambahkan pengguna ke grup pengguna. Ikuti instruksi dalam [Menambahkan izin ke pengguna \(konsol\)](#) dalam Panduan Pengguna IAM.

Saat membuat aplikasi yang menggunakan Amazon Location Service, Anda mungkin memerlukan beberapa pengguna untuk memiliki akses yang tidak diautentikasi. Untuk kasus penggunaan ini, lihat [Mengaktifkan akses tidak diautentikasi menggunakan Amazon Cognito](#).

## Menggunakan Peta Lokasi Amazon di aplikasi Anda

Peta Lokasi Amazon hemat biaya dan interaktif. Anda dapat mengganti peta yang ada di aplikasi Anda untuk menghemat uang, atau menambahkan yang baru untuk menampilkan data berbasis lokasi secara visual, seperti lokasi toko Anda.



Amazon Location Service memungkinkan Anda memilih penyedia data untuk operasi peta dengan membuat dan mengonfigurasi sumber daya peta. Sumber daya peta mengonfigurasi penyedia data dan gaya yang digunakan untuk merender peta.

Setelah membuat sumber daya, Anda dapat mengirim permintaan menggunakan AWS SDK secara langsung, atau dengan menggunakan pustaka yang dibuat khusus untuk merender peta di lingkungan Anda.

### Note

Untuk ikhtisar konsep peta, lihat [Peta](#).

## Topik

- [Prasyarat](#)
- [Menampilkan peta di aplikasi Anda](#)
- [Menggambar fitur data pada peta](#)
- [Mengatur luasan untuk peta menggunakan MapLibre](#)
- [Mengelola sumber daya peta Anda](#)

## Prasyarat

Sebelum Anda menampilkan peta dalam aplikasi Anda ikuti langkah-langkah prasyarat:

### Topik

- [Buat sumber daya peta](#)
- [Mengautentikasi permintaan Anda](#)

## Buat sumber daya peta

Untuk menggunakan peta dalam aplikasi Anda, Anda harus memiliki sumber daya peta, yang menentukan gaya peta dan penyedia data yang akan digunakan di peta Anda.

### Note

Jika aplikasi Anda melacak atau merutekan aset yang Anda gunakan dalam bisnis Anda, seperti kendaraan pengiriman atau karyawan, Anda tidak boleh menggunakan Esri sebagai penyedia geolokasi Anda. Lihat bagian 82 dari [ketentuan AWS layanan](#) untuk detail selengkapnya.

Anda dapat membuat sumber daya peta menggunakan konsol Amazon Location Service, Amazon Location API/AWS CLI, atau Amazon Location.

### Console

Untuk membuat sumber daya peta menggunakan konsol Amazon Location Service

1. Di konsol Lokasi Amazon, pada halaman [Maps](#), pilih Buat peta untuk melihat pratinjau gaya peta.

2. Tambahkan nama dan deskripsi untuk sumber daya peta baru.
3. Pilih gaya peta.

**Note**

Jika aplikasi Anda melacak atau merutekan aset yang Anda gunakan dalam bisnis Anda, seperti kendaraan pengiriman atau karyawan, Anda tidak boleh menggunakan Esri sebagai penyedia geolokasi Anda. Lihat bagian 82 dari [ketentuan AWS layanan](#) untuk detail selengkapnya.

4. Pilih dari yang [Pandangan politik](#) akan digunakan.
5. Setuju dengan Syarat dan Ketentuan Lokasi Amazon, lalu pilih Buat peta. Anda dapat berinteraksi dengan peta yang Anda pilih: memperbesar, memperkecil, atau menggeser ke segala arah.
6. Untuk memungkinkan pengguna beralih gaya (misalnya, untuk memungkinkan mereka beralih antara citra satelit dan gaya vektor), Anda harus membuat sumber daya peta untuk setiap gaya.

Anda dapat menghapus sumber daya dengan gaya peta yang tidak ingin Anda gunakan di [halaman beranda Maps](#) di konsol.

## API

Untuk membuat sumber daya peta menggunakan Amazon Location API

Gunakan [CreateMap](#) operasi dari Amazon Location API.

Contoh berikut adalah permintaan API untuk membuat sumber daya peta yang disebut *ExampleMap* menggunakan gaya *VectorEsriStreets* peta.

```
POST /maps/v0/maps HTTP/1.1
Content-type: application/json

{
  "Configuration": {
    "Style": "VectorEsriStreets"
  },
  "MapName": "ExampleMap"
}
```

```
}
```

### Note

Jika aplikasi Anda melacak atau merutekan aset yang Anda gunakan dalam bisnis Anda, seperti kendaraan pengiriman atau karyawan, Anda tidak boleh menggunakan Esri sebagai penyedia geolokasi Anda. Lihat bagian 82 dari [ketentuan AWS layanan](#) untuk detail selengkapnya.

## AWS CLI

Untuk membuat sumber daya peta menggunakan AWS CLI perintah

Gunakan perintah [create-map](#).

Contoh berikut menciptakan sumber daya peta yang disebut *ExampleMap* menggunakan *VectorEsriStreets* sebagai gaya peta.

```
aws location \  
  create-map \  
    --configuration Style="VectorEsriStreets" \  
    --map-name "ExampleMap"
```

### Note

Jika aplikasi Anda melacak atau merutekan aset yang Anda gunakan dalam bisnis Anda, seperti kendaraan pengiriman atau karyawan, Anda tidak boleh menggunakan Esri sebagai penyedia geolokasi Anda. Lihat bagian 82 dari [ketentuan AWS layanan](#) untuk detail selengkapnya.

## Mengautentikasi permintaan Anda

Setelah Anda membuat sumber daya peta dan Anda siap untuk mulai membangun fitur lokasi ke dalam aplikasi Anda, Anda harus memilih bagaimana Anda akan mengautentikasi permintaan Anda.

**Note**

Sebagian besar aplikasi front end maps memerlukan akses tidak terautentikasi ke peta atau fitur lain dari Amazon Location Service. Bergantung pada aplikasi Anda, Anda mungkin ingin menggunakan AWS Signature v4 untuk mengautentikasi permintaan, atau Anda dapat menggunakan kunci Amazon Cognito atau Amazon Location API untuk penggunaan yang tidak diautentikasi. Untuk mempelajari lebih lanjut tentang semua opsi ini, lihat [Memberikan akses ke Amazon Location Service](#).

## Menampilkan peta di aplikasi Anda

Bagian ini menyediakan tutorial tentang cara menggunakan alat rendering peta untuk menampilkan peta di aplikasi seluler atau web Anda saat menggunakan API Lokasi Amazon. Seperti yang disebutkan dalam [Cara menggunakan Amazon Location Service](#) topik, Anda memiliki pilihan pustaka untuk digunakan saat merender peta dengan Lokasi Amazon, termasuk Amplify,, dan Tangram. MapLibre

Lakukan salah satu hal berikut untuk menampilkan peta di aplikasi Anda:

- Cara paling langsung untuk menampilkan peta di web dan aplikasi front end seluler Anda adalah dengan menggunakannya MapLibre. Anda dapat mengikuti [MapLibre tutorial](#) atau bahkan [tutorial Mulai cepat](#) untuk mempelajari cara menggunakan MapLibre.
- Jika Anda adalah AWS Amplify pengembang yang sudah ada, Anda mungkin ingin menggunakan Amplify Geo SDK. Untuk mempelajari lebih lanjut, ikuti tutorial [Amplify](#).
- Jika Anda adalah pengguna Tangram yang sudah ada, dan ingin terus menggunakannya untuk merender peta Anda, saat pindah ke Amazon Location Service, ikuti tutorial [Tangram](#).

### Topik

- [Menggunakan MapLibre library dengan Amazon Location Service](#)
- [Menggunakan library Amplify dengan Amazon Location Service](#)
- [Menggunakan Tangram dengan Amazon Location Service](#)

## Menggunakan MapLibre library dengan Amazon Location Service

Tutorial berikut memandu Anda menggunakan MapLibre Perpustakaan dengan Lokasi Amazon.

## Topik

- [Menggunakan MapLibre GL JS dengan Amazon Location Service](#)
- [Menggunakan MapLibre Native SDK for Android dengan Amazon Location Service](#)
- [Menggunakan MapLibre Native SDK for iOS dengan Amazon Location Service](#)

## Menggunakan MapLibre GL JS dengan Amazon Location Service

Gunakan [MapLibre GL JS](#) untuk menyematkan peta sisi klien ke dalam aplikasi web.

MapLibre GL JS adalah JavaScript pustaka sumber terbuka yang kompatibel dengan gaya dan ubin yang disediakan oleh Amazon Location Service Maps API. Anda dapat mengintegrasikan MapLibre GL JS dalam HTML dasar atau JavaScript aplikasi untuk menyematkan peta sisi klien yang dapat disesuaikan dan responsif.

Tutorial ini menjelaskan cara mengintegrasikan MapLibre GL JS dengan Amazon Location dalam HTML dasar dan JavaScript aplikasi. [Pustaka dan teknik yang sama yang disajikan dalam tutorial ini juga berlaku untuk kerangka kerja, seperti React dan Angular.](#)

Aplikasi sampel untuk tutorial ini tersedia sebagai bagian dari repositori sampel Amazon Location Service di [GitHub](#)

### Membangun aplikasi: Scaffolding

Tutorial ini membuat aplikasi web yang digunakan JavaScript untuk membangun peta pada halaman HTML.

Mulailah dengan membuat halaman HTML (`index.html`) yang menyertakan wadah peta:

- Masukkan `div` elemen dengan `id` of `map` untuk menerapkan dimensi peta ke tampilan peta. Dimensi diwarisi dari `viewport`.

```
<html>
  <head>
    <style>
      body {
        margin: 0;
      }
    </style>
  </head>
</html>
```



```
#map {
  height: 100vh; /* 100% of viewport height */
}
</style>
</head>
<body>
  <!-- map container -->
  <div id="map" />
</body>
</html>
```

Membangun aplikasi: Menambahkan dependensi

Tambahkan dependensi berikut ke aplikasi Anda:

- MapLibre GL JS (v3.x), dan CSS terkait.
- Lokasi Amazon [JavaScript Pembantu otentikasi](#).

```
<!-- CSS dependencies -->
<link
  href="https://unpkg.com/maplibre-gl@3.x/dist/maplibre-gl.css"
  rel="stylesheet"
/>
<!-- JavaScript dependencies -->
<script src="https://unpkg.com/maplibre-gl@3.x/dist/maplibre-gl.js"></script>
<script src="https://unpkg.com/@aws/amazon-location-authentication-helper.js"></script>
<script>
  // application-specific code
</script>
```

Ini membuat halaman kosong dengan wadah peta.

Membangun aplikasi: Konfigurasi

Untuk mengkonfigurasi aplikasi Anda menggunakan JavaScript:

1. Masukkan nama dan pengidentifikasi sumber daya Anda.

```
// Cognito Identity Pool ID
const identityPoolId = "us-east-1:54f2ba88-9390-498d-aaa5-0d97fb7ca3bd";
// Amazon Location Service Map name
```

```
const mapName = "ExampleMap";
```

2. Buat instance penyedia kredensi menggunakan kumpulan identitas tidak diautentikasi yang Anda buat di [Menggunakan peta - Langkah 2](#), Siapkan otentikasi. Kami akan menempatkan ini dalam fungsi yang disebut `initializeMap`, yang juga akan berisi kode inisialisasi peta lainnya, ditambahkan pada langkah berikutnya

```
// extract the Region from the Identity Pool ID; this will be used for both Amazon
  Cognito and Amazon Location
AWS.config.region = identityPoolId.split(":")[0];

async function initializeMap() {
  // Create an authentication helper instance using credentials from Cognito
  const authHelper = await
  amazonLocationAuthHelper.withIdentityPoolId(identityPoolId);

  // ... more here, later
}
```

## Membangun aplikasi: Inisialisasi peta

Agar peta ditampilkan setelah halaman dimuat, Anda harus menginisialisasi peta. Anda dapat menyesuaikan lokasi peta awal, menambahkan kontrol tambahan, dan data hamparan.

```
async function initializeMap() {
  // Create an authentication helper instance using credentials from Cognito
  const authHelper = await amazonLocationAuthHelper.withIdentityPoolId(identityPoolId);

  // Initialize the map
  const map = new maplibregl.Map({
    container: "map",
    center: [-123.1187, 49.2819], // initial map centerpoint
    zoom: 10, // initial map zoom
    style: 'https://maps.geo.${region}.amazonaws.com/maps/v0/maps/${mapName}/style-
descriptor',
    ...authHelper.getMapAuthenticationOptions(), // authentication, using cognito
  });

  map.addControl(new maplibregl.NavigationControl(), "top-left");
}
```

```
initializeMap();
```

### Note

Anda harus memberikan tanda kata atau atribusi teks untuk setiap penyedia data yang Anda gunakan, baik pada aplikasi atau dokumentasi Anda. String atribusi disertakan dalam respons deskriptor gaya di `bawahsources.esri.attribution,sources.here.attribution`, dan kunci `sources.grabmaptiles.attribution` MapLibre GL JS akan secara otomatis memberikan atribusi. Saat menggunakan sumber daya Lokasi Amazon dengan [penyedia data](#), pastikan untuk membaca [syarat dan ketentuan layanan](#).

## Menjalankan aplikasi

Anda dapat menjalankan contoh aplikasi ini dengan menggunakannya di server web lokal, atau membukanya di browser.

Untuk menggunakan server web lokal, Anda dapat menggunakan `npx`, karena itu diinstal sebagai bagian dari Node.js. Anda dapat menggunakan `npx serve` dari dalam direktori yang sama dengan `index.html`. Ini melayani aplikasi pada `localhost:5000`.

### Note

Jika kebijakan yang Anda buat untuk peran Amazon Cognito yang tidak diautentikasi menyertakan `referrer` suatu kondisi, Anda mungkin diblokir dari pengujian dengan URL `localhost`: Dalam hal ini, Anda dapat menguji dengan server web yang menyediakan URL yang ada dalam kebijakan Anda.

Setelah menyelesaikan tutorial, aplikasi akhir terlihat seperti contoh berikut.

```
<!-- index.html -->
<html>
  <head>
    <link href="https://unpkg.com/maplibre-gl@3.x/dist/maplibre-gl.css"
rel="stylesheet" />
    <style>
      body {
        margin: 0;
      }
    </style>
  </head>
  <body>
    <div id="map">
      <img alt="Map showing a location on a map." data-bbox="100 720 900 900"/>
    </div>
  </body>
</html>
```

```
    #map {
      height: 100vh;
    }
  </style>
</head>

<body>
  <!-- map container -->
  <div id="map" />
  <!-- JavaScript dependencies -->
  <script src="https://unpkg.com/maplibre-gl@3.x/dist/maplibre-gl.js"></script>
  <script src="https://unpkg.com/@aws/amazon-location-authentication-helper.js"></
script>
  <script>
    // configuration
    const identityPoolId = "us-east-1:54f2ba88-9390-498d-aaa5-0d97fb7ca3bd"; //
Cognito Identity Pool ID
    const mapName = "ExampleMap"; // Amazon Location Service Map Name

    // extract the region from the Identity Pool ID
    const region = identityPoolId.split(":")[0];

    async function initializeMap() {
      // Create an authentication helper instance using credentials from Cognito
      const authHelper = await
amazonLocationAuthHelper.withIdentityPoolId(identityPoolId);

      // Initialize the map
      const map = new maplibregl.Map({
        container: "map",
        center: [-123.115898, 49.295868],
        zoom: 10,
        style: `https://maps.geo.${region}.amazonaws.com/maps/v0/maps/${mapName}/
style-descriptor`,
        ...authHelper.getMapAuthenticationOptions(),
      });
      map.addControl(new maplibregl.NavigationControl(), "top-left");
    }

    initializeMap();
  </script>
</body>
</html>
```

Menjalankan aplikasi ini menampilkan peta layar penuh menggunakan gaya peta yang Anda pilih. Sampel ini tersedia di repositori sampel Amazon Location Service di [GitHub](#)

## Menggunakan MapLibre Native SDK for Android dengan Amazon Location Service

Gunakan [MapLibreNative](#) SDK untuk menyematkan peta interaktif ke dalam aplikasi Android Anda.

MapLibre Native SDK for Android adalah library berbasis [Mapbox](#) Native, dan kompatibel dengan style dan tile yang disediakan oleh Amazon Location Service Maps API. Anda dapat mengintegrasikan MapLibre Native SDK for Android untuk menyematkan tampilan peta interaktif dengan peta vektor yang dapat diskalakan dan dapat disesuaikan di aplikasi Android Anda.

Tutorial ini menjelaskan cara mengintegrasikan MapLibre Native SDK for Android dengan Amazon Location. Aplikasi sampel untuk tutorial ini tersedia sebagai bagian dari repositori sampel Amazon Location Service di [GitHub](#)

## Membangun aplikasi: Inisialisasi

Untuk menginisialisasi aplikasi Anda:

1. Buat project Android Studio baru dari template Empty Activity.
2. Pastikan bahwa Kotlin dipilih untuk bahasa proyek.
3. Pilih SDK Minimum API 14: Android 4.0 (Ice Cream Sandwich) atau yang lebih baru.
4. Buka Project Structure, lalu pergi ke File> Project Structure... untuk memilih bagian Dependensi.
5. Dengan <All Modules>dipilih, lalu pilih tombol + untuk menambahkan Ketergantungan Perpustakaan baru.
6. Tambahkan AWS Android SDK versi 2.20.0 atau yang lebih baru. Sebagai contoh:  
`com.amazonaws:aws-android-sdk-core:2.20.0`
7. Tambahkan MapLibre Native SDK for Android versi 9.4.0 atau yang lebih baru. Sebagai contoh:  
`org.maplibre.gl:android-sdk:9.4.0`
8. Pada level project file build.gradle Anda, tambahkan repositori maven berikut untuk mengakses paket untuk Android: MapLibre

```
allprojects {
    repositories {
        // Retain your existing repositories
        google()
        jcenter()
    }
}
```

```
        // Declare the repositories for MapLibre
        mavenCentral()
    }
}
```

## Membangun aplikasi: Konfigurasi

Untuk mengonfigurasi aplikasi Anda dengan sumber daya dan AWS Wilayah Anda:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="identityPoolId">us-east-1:54f2ba88-9390-498d-aaa5-0d97fb7ca3bd</string>
    <string name="mapName">ExampleMap</string>
    <string name="awsRegion">us-east-1</string>
</resources>
```

## Membangun aplikasi: Tata letak aktivitas

Suntingapp/src/main/res/layout/activity\_main.xml:

- Tambahkan aMapView, yang membuat peta. Ini juga akan mengatur titik pusat awal peta.
- Tambahkan aTextView, yang menampilkan atribusi.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <com.mapbox.mapboxsdk.maps.MapView
        android:id="@+id/mapView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:mapbox_cameraTargetLat="49.2819"
        app:mapbox_cameraTargetLng="-123.1187"
        app:mapbox_cameraZoom="12"
        app:mapbox_uiAttribution="false"
```

```
        app:mapbox_uiLogo="false" />

    <TextView
        android:id="@+id/attributionView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#80808080"
        android:padding="5sp"
        android:textColor="@android:color/black"
        android:textSize="10sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        tools:ignore="SmallSp" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

### Note

Anda harus memberikan tanda kata atau atribusi teks untuk setiap penyedia data yang Anda gunakan, baik pada aplikasi atau dokumentasi Anda. String atribusi disertakan dalam respons deskriptor gaya di bawah `sources.esri.attribution`, `sources.here.attribution`, dan kunci `source.grabmaptiles.attribution`. Saat menggunakan sumber daya Lokasi Amazon dengan [penyedia data](#), pastikan untuk membaca [syarat dan ketentuan layanan](#).

Membangun aplikasi: Minta transformasi

Buat kelas bernama `SigV4Interceptor` untuk mencegah permintaan AWS dan menandatangani menggunakan [Signature Version 4](#). Ini akan terdaftar dengan klien HTTP yang digunakan untuk mengambil sumber daya peta saat Aktivitas Utama dibuat.

```
package aws.location.demo.okhttp

import com.amazonaws.DefaultRequest
import com.amazonaws.auth.AWS4Signer
import com.amazonaws.auth.AWSCredentialsProvider
import com.amazonaws.http.HttpMethodName
import com.amazonaws.util.IOUtils
import okhttp3.HttpUrl
import okhttp3.Interceptor
import okhttp3.Request
```

```

import okhttp3.Response
import okio.Buffer
import java.io.ByteArrayInputStream
import java.net.URI

class SigV4Interceptor(
    private val credentialsProvider: AWSCredentialsProvider,
    private val serviceName: String
) : Interceptor {
    override fun intercept(chain: Interceptor.Chain): Response {
        val originalRequest = chain.request()

        if (originalRequest.url().host().contains("amazonaws.com")) {
            val signer = if (originalRequest.url().encodedPath().contains("@")) {
                // the presence of "@" indicates that it doesn't need to be double URL-
                encoded
                AWS4Signer(false)
            } else {
                AWS4Signer()
            }

            val awsRequest = toAWSRequest(originalRequest, serviceName)
            signer.setServiceName(serviceName)
            signer.sign(awsRequest, credentialsProvider.credentials)

            return chain.proceed(toSignedOkHttpRequest(awsRequest, originalRequest))
        }

        return chain.proceed(originalRequest)
    }

    companion object {
        fun toAWSRequest(request: Request, serviceName: String): DefaultRequest<Any> {
            // clone the request (AWS-style) so that it can be populated with
            credentials
            val dr = DefaultRequest<Any>(serviceName)

            // copy request info
            dr.httpMethod = HttpMethodName.valueOf(request.method())
            with(request.url()) {
                dr.resourcePath = uri().path
                dr.endpoint = URI.create("${scheme()}://${host()}")

                // copy parameters

```



```
        for (p in queryParameterNames()) {
            if (p != "") {
                dr.addParameter(p, queryParameter(p))
            }
        }

// copy headers
for (h in request.headers().names()) {
    dr.addHeader(h, request.header(h))
}

// copy the request body
val bodyBytes = request.body()?.let { body ->
    val buffer = Buffer()
    body.writeTo(buffer)
    IOUtils.toByteArray(buffer.inputStream())
}

dr.content = ByteArrayInputStream(bodyBytes ?: ByteArray(0))

return dr
}

fun toSignedOkHttpRequest(
    awsRequest: DefaultRequest<Any>,
    originalRequest: Request
): Request {
    // copy signed request back into an OkHttp Request
    val builder = Request.Builder()

    // copy headers from the signed request
    for ((k, v) in awsRequest.headers) {
        builder.addHeader(k, v)
    }

    // start building an HttpUrl
    val urlBuilder = HttpUrl.Builder()
        .host(awsRequest.endpoint.host)
        .scheme(awsRequest.endpoint.scheme)
        .encodedPath(awsRequest.resourcePath)

    // copy parameters from the signed request
    for ((k, v) in awsRequest.parameters) {
```

```
        urlBuilder.addQueryParameter(k, v)
    }

    return builder.url(urlBuilder.build())
        .method(originalRequest.method(), originalRequest.body())
        .build()
    }
}
}
```

## Membangun aplikasi: Kegiatan utama

Aktivitas Utama bertanggung jawab untuk menginisialisasi tampilan yang akan ditampilkan kepada pengguna. Ini melibatkan:

- Membuat Instantiasi Amazon Cognito. `CredentialsProvider`
- Mendaftarkan pencegat Signature Version 4.
- Mengkonfigurasi peta dengan mengarahkannya ke deskriptor gaya peta, dan menampilkan atribusi yang sesuai.

`MainActivity` juga bertanggung jawab untuk meneruskan peristiwa siklus hidup ke tampilan peta, memungkinkannya mempertahankan viewport aktif di antara pemanggilan.

```
package aws.location.demo.maplibre

import android.os.Bundle
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import aws.location.demo.okhttp.SigV4Interceptor
import com.amazonaws.auth.CognitoCachingCredentialsProvider
import com.amazonaws.regions.Regions
import com.mapbox.mapboxsdk.Mapbox
import com.mapbox.mapboxsdk.maps.MapView
import com.mapbox.mapboxsdk.maps.Style
import com.mapbox.mapboxsdk.module.http.HttpRequestUtil
import okhttp3.OkHttpClient

private const val SERVICE_NAME = "geo"

class MainActivity : AppCompatActivity() {
    private var mapView: MapView? = null
```

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)

    // configuration
    val identityPoolId = getString(R.string.identityPoolId)
    val region = getString(R.string.awsRegion)
    val mapName = getString(R.string.mapName)

    // Credential initialization
    val credentialProvider = CognitoCachingCredentialsProvider(
        applicationContext,
        identityPoolId,
        Regions.fromName(identityPoolId.split(":").first())
    )

    // initialize MapLibre
    Mapbox.getInstance(this, null)
    HttpRequestUtil.setOkHttpClient(
        OkHttpClient.Builder()
            .addInterceptor(SigV4Interceptor(credentialProvider, SERVICE_NAME))
            .build()
    )

    // initialize the view
    setContentView(R.layout.activity_main)

    // initialize the map view
    mapView = findViewById(R.id.mapView)
    mapView?.onCreate(savedInstanceState)
    mapView?.getMapAsync { map ->
        map.setStyle(
            Style.Builder()
                .fromUri("https://maps.geo.{$region}.amazonaws.com/maps/v0/maps/
${mapName}/style-descriptor")
        ) { style ->
            findViewById<TextView>(R.id.attributionView).text =
style.sources.first()?.attribution
        }
    }
}

override fun onStart() {
    super.onStart()
}
```

```
        mapView?.onStart()
    }

    override fun onResume() {
        super.onResume()
        mapView?.onResume()
    }

    override fun onPause() {
        super.onPause()
        mapView?.onPause()
    }

    override fun onStop() {
        super.onStop()
        mapView?.onStop()
    }

    override fun onSaveInstanceState(outState: Bundle) {
        super.onSaveInstanceState(outState)
        mapView?.onSaveInstanceState(outState)
    }

    override fun onLowMemory() {
        super.onLowMemory()
        mapView?.onLowMemory()
    }

    override fun onDestroy() {
        super.onDestroy()
        mapView?.onDestroy()
    }
}
```

Menjalankan aplikasi ini menampilkan peta layar penuh dengan gaya yang Anda pilih. Sampel ini tersedia sebagai bagian dari repositori sampel Amazon Location Service di [GitHub](#)

### Menggunakan MapLibre Native SDK for iOS dengan Amazon Location Service

Gunakan [MapLibreNative SDK for iOS untuk](#) menyematkan peta sisi klien ke dalam aplikasi iOS.

MapLibre Native SDK for iOS adalah pustaka [berdasarkan Mapbox GL](#) Native, dan kompatibel dengan gaya dan ubin yang disediakan oleh Amazon Location Service Maps API. Anda dapat

mengintegrasikan MapLibre Native SDK for iOS untuk menyematkan tampilan peta interaktif dengan peta vektor yang dapat diskalakan dan dapat disesuaikan ke dalam aplikasi iOS Anda.

Tutorial ini menjelaskan cara mengintegrasikan MapLibre Native SDK for iOS dengan Amazon Location. Aplikasi sampel untuk tutorial ini tersedia sebagai bagian dari repositori sampel Amazon Location Service di [GitHub](#)

Membangun aplikasi: Inisialisasi


Untuk menginisialisasi aplikasi Anda:

1. Buat proyek Xcode baru dari template App.
2. Pilih SwiftUI untuk antarmukanya.
3. Pilih aplikasi SwiftUI untuk Siklus Hidupnya.
4. Pilih Swift untuk bahasanya.

Menambahkan MapLibre dependensi menggunakan Paket Swift

Untuk menambahkan ketergantungan paket ke proyek Xcode Anda:

1. Arahkan ke File > Paket Swift > Add Package Dependency.
2. Masukkan URL repositori: **<https://github.com/maplibre/maplibre-gl-native-distribution>**

 Note

Untuk informasi selengkapnya tentang Paket Swift, lihat [Menambahkan Dependensi Paket ke Aplikasi Anda](#) di Apple.com

3. Di terminal Anda, instal CocoaPods:

```
sudo gem install cocoapods
```

4. Arahkan ke direktori proyek aplikasi Anda dan inisialisasi Podfile dengan manajer CocoaPods paket:

```
pod init
```

5. Buka Podfile untuk ditambahkan AWSCore sebagai dependensi:

```
platform :ios, '12.0'

target 'Amazon Location Service Demo' do
  use_frameworks!

  pod 'AWSCore'
end
```

## 6. Unduh dan instal dependensi:

```
pod install --repo-update
```

## 7. Buka ruang kerja Xcode yang membuat CocoaPods :

```
xed .
```

## Membangun aplikasi: Konfigurasi

Tambahkan kunci dan nilai berikut ke Info.plist untuk mengkonfigurasi aplikasi:

Kunci	Nilai
AWSRegion	us-east-1
IdentityPoolId	us-timur- 1:54 f2ba88-9390-498d-aaa5-0d97f b7ca3bd
MapName	ExampleMap

## Membangun aplikasi: ContentView tata letak

Untuk merender peta, edit `ContentView.swift`:

- Tambahkan `MapView` yang membuat peta.
- Tambahkan `TextField` yang menampilkan atribusi.

Ini juga menetapkan titik pusat awal peta.

```
import SwiftUI

struct ContentView: View {
    @State private var attribution = ""

    var body: some View {
        MapView(attribution: $attribution)
            .centerCoordinate(.init(latitude: 49.2819, longitude: -123.1187))
            .zoomLevel(12)
            .edgesIgnoringSafeArea(.all)
            .overlay(
                TextField("", text: $attribution)
                    .disabled(true)
                    .font(.system(size: 12, weight: .light, design: .default))
                    .foregroundColor(.black)
                    .background(Color.init(Color.RGBColorSpace.sRGB, white: 0.5,
opacity: 0.5))
                    .cornerRadius(1),
                alignment: .bottomTrailing)
            }
    }

    struct ContentView_Previews: PreviewProvider {
        static var previews: some View {
            ContentView()
        }
    }
}
```

### Note

Anda harus memberikan tanda kata atau atribusi teks untuk setiap penyedia data yang Anda gunakan, baik pada aplikasi atau dokumentasi Anda. String atribusi disertakan dalam respons deskriptor gaya di bawah `sources.esri.attribution`, `sources.here.attribution`, dan kunci `source.grabmaptiles.attribution`. Saat menggunakan sumber daya Lokasi Amazon dengan [penyedia data](#), pastikan untuk membaca [syarat dan ketentuan layanan](#).

## Membangun aplikasi: Minta transformasi

Buat file Swift baru bernama `AWSSignatureV4Delegate.swift` berisi definisi kelas berikut untuk mencegah permintaan AWS dan menandatangani menggunakan [Signature Version 4](#). Sebuah instance dari kelas ini akan ditetapkan sebagai delegasi penyimpanan offline, yang juga bertanggung jawab untuk menulis ulang URL, dalam tampilan peta.

```
import AWSCore
import Mapbox

class AWSSignatureV4Delegate : NSObject, MGLOfflineStorageDelegate {
    private let region: AWSRegionType
    private let identityPoolId: String
    private let credentialsProvider: AWSCredentialsProvider

    init(region: AWSRegionType, identityPoolId: String) {
        self.region = region
        self.identityPoolId = identityPoolId
        self.credentialsProvider = AWSCognitoCredentialsProvider(regionType: region,
identityPoolId: identityPoolId)
        super.init()
    }

    class func doubleEncode(path: String) -> String? {
        return path.addingPercentEncoding(withAllowedCharacters: .urlPathAllowed)?
            .addingPercentEncoding(withAllowedCharacters: .urlPathAllowed)
    }

    func offlineStorage(_ storage: MGLOfflineStorage, urlForResourceOf kind:
MGLResourceKind, with url: URL) -> URL {
        if url.host?.contains("amazonaws.com") != true {
            // not an AWS URL
            return url
        }

        // URL-encode spaces, etc.
        let keyPath = String(url.path.dropFirst())
        guard let percentEncodedKeyPath =
keyPath.addingPercentEncoding(withAllowedCharacters: .urlPathAllowed) else {
            print("Invalid characters in path '\(keyPath)'; unsafe to sign")
            return url
        }
    }
}
```



```

let endpoint = AWSEndpoint(region: region, serviceName: "geo", url: url)
let requestHeaders: [String: String] = ["host": endpoint!.hostName]

// sign the URL
let task = AWSSignatureV4Signer
    .generateQueryStringForSignatureV4(
        withCredentialProvider: credentialsProvider,
        httpMethod: .GET,
        expireDuration: 60,
        endpoint: endpoint!,
        // workaround for https://github.com/aws-amplify/aws-sdk-ios/
issues/3215
        keyPath: AWSSignatureV4Delegate.doubleEncode(path:
percentEncodedKeyPath),
        requestHeaders: requestHeaders,
        requestParameters: .none,
        signBody: true)
task.waitUntilFinished()

if let error = task.error as NSError? {
    print("Error occurred: \(error)")
}

if let result = task.result {
    var urlComponents = URLComponents(url: (result as URL),
resolvingAgainstBaseURL: false)!
    // re-use the original path; workaround for https://github.com/aws-amplify/
aws-sdk-ios/issues/3215
    urlComponents.path = url.path

    // have Mapbox GL fetch the signed URL
    return (urlComponents.url)!
}

// fall back to an unsigned URL
return url
}
}

```

## Membangun aplikasi: Tampilan peta

Tampilan Peta bertanggung jawab untuk menginisialisasi instance `AWSSignatureV4Delegate` dan mengonfigurasi yang mendasarinya `MGLMapView`, yang mengambil sumber daya dan merender

peta. Ini juga menangani penyebaran string atribusi dari sumber deskriptor gaya kembali ke.

## ContentView

Buat file Swift baru bernama `MapView.swift` berisi struct definisi berikut:

```
import SwiftUI
import AWSCore
import Mapbox

struct MapView: UIViewRepresentable {
    @Binding var attribution: String

    private var mapView: MGLMapView
    private var signingDelegate: MGLOfflineStorageDelegate

    init(attribution: Binding<String>) {
        let regionName = Bundle.main.object(forKey: "AWSRegion") as!
String
        let identityPoolId = Bundle.main.object(forKey: "IdentityPoolId")
as! String
        let mapName = Bundle.main.object(forKey: "MapName") as! String

        let region = (regionName as NSString).aws_regionTypeValue()

        // MGLOfflineStorage doesn't take ownership, so this needs to be a member here
        signingDelegate = AWSSignatureV4Delegate(region: region, identityPoolId:
identityPoolId)

        // register a delegate that will handle SigV4 signing
        MGLOfflineStorage.shared.delegate = signingDelegate

        mapView = MGLMapView(
            frame: .zero,
            styleURL: URL(string: "https://maps.geo.\(regionName).amazonaws.com/maps/
v0/maps/\(mapName)/style-descriptor"))

        _attribution = attribution
    }

    func makeCoordinator() -> Coordinator {
        Coordinator($attribution)
    }

    class Coordinator: NSObject, MGLMapViewDelegate {
```

```
var attribution: Binding<String>

init(_ attribution: Binding<String>) {
    self.attribution = attribution
}

func mapView(_ mapView: MGLMapView, didFinishLoading style: MGLStyle) {
    let source = style.sources.first as? MGLVectorTileSource
    let attribution = source?.attributionInfos.first
    self.attribution.wrappedValue = attribution?.title.string ?? ""
}

// MARK: - UIViewRepresentable protocol

func makeUIView(context: UIViewRepresentableContext<MapView>) -> MGLMapView {
    mapView.delegate = context.coordinator

    mapView.logoView.isHidden = true
    mapView.attributionButton.isHidden = true
    return mapView
}

func updateUIView(_ uiView: MGLMapView, context:
UIViewRepresentableContext<MapView>) {
}

// MARK: - MGLMapView proxy

func centerCoordinate(_ centerCoordinate: CLLocationCoordinate2D) -> MapView {
    mapView.centerCoordinate = centerCoordinate
    return self
}

func zoomLevel(_ zoomLevel: Double) -> MapView {
    mapView.zoomLevel = zoomLevel
    return self
}
}
```

Menjalankan aplikasi ini menampilkan peta layar penuh dengan gaya yang Anda pilih. Sampel ini tersedia sebagai bagian dari repositori sampel Amazon Location Service di [GitHub](#)

## Menggunakan library Amplify dengan Amazon Location Service

Tutorial berikut memandu Anda menggunakan AWS Amplify dengan Amazon Location. Amplify menggunakan MapLibre GL JS untuk merender peta di aplikasi berbasis Anda JavaScript.

Amplify adalah sekumpulan pustaka klien open-source yang menyediakan antarmuka ke berbagai kategori layanan, termasuk Amplify Geo, yang didukung oleh Amazon Location Service. [Pelajari lebih lanjut tentang JavaScript perpustakaan AWS Amplify Geo.](#)

### Note

Tutorial ini mengasumsikan bahwa Anda telah mengikuti langkah-langkah dalam [Menggunakan peta - Untuk menambahkan peta ke aplikasi Anda.](#)

### Membangun aplikasi: Scaffolding

Tutorial ini membuat aplikasi web yang digunakan JavaScript untuk membangun peta pada halaman HTML.

Mulailah dengan membuat halaman HTML (`index.html`) yang menyertakan wadah peta:

- Masukkan `div` elemen dengan `id of map` untuk menerapkan dimensi peta ke tampilan peta. Dimensi diwarisi dari `viewport`.

```
<html>
  <head>
    <style>
      body { margin: 0; }
      #map { height: 100vh; } /* 100% of viewport height */
    </style>
  </head>

  <body>
    <!-- map container -->
    <div id="map" />
  </body>
</html>
```

## Membangun aplikasi: Menambahkan dependensi

Tambahkan dependensi berikut ke aplikasi Anda:

- AWS Amplify peta dan perpustakaan geo.
- AWS Amplify perpustakaan inti.
- AWS Amplify perpustakaan autentikasi.
- AWS Amplify stylesheet.

```
<!-- CSS dependencies -->
  <link href="https://cdn.amplify.aws/packages/maplibre-
gl/1.15.2/maplibre-gl.css" rel="stylesheet" integrity="sha384-
DrPVD9GufrixGb7kWwRv0CywpXTmfvbK0Z5i5pN7urmIThew0zXKTME+gutUgtpeD"
  crossorigin="anonymous" referrerpolicy="no-referrer"></link>

<!-- JavaScript dependencies -->
  <script src="https://cdn.amplify.aws/packages/maplibre-gl/1.15.2/maplibre-gl.js"
  integrity="sha384-rwYfkmA0pciZS2bDuwZ/Xa/Gog6jXem8D/whm3wnsZSVFemDDlprcUXHnDDUcrNU"
  crossorigin="anonymous" referrerpolicy="no-referrer"></script>
  <script src="https://cdn.amplify.aws/packages/core/4.3.0/aws-amplify-core.min.js"
  integrity="sha384-70h+5w0l7XGyYvSqbkKi2Q7SA5K640V5nyW2/LEbevDQEV1HMJqJLA1A00z2hu8fJ"
  crossorigin="anonymous" referrerpolicy="no-referrer"></script>
  <script src="https://cdn.amplify.aws/packages/auth/4.3.8/aws-amplify-auth.min.js"
  integrity="sha384-jfkXCEfYyVmDXyKlGWNwv54xRaZgk14m7sjeb2jLVBtUXCD2p+WU8YZ2mPZ9Xbdw"
  crossorigin="anonymous" referrerpolicy="no-referrer"></script>
  <script src="https://cdn.amplify.aws/packages/geo/1.1.0/aws-amplify-geo.min.js"
  integrity="sha384-TFMTyWuCbipXTzv0gzJbV8TPUupG1rA1AVrznAhCSpXTIdGw82bGd8RTk5rr3nP"
  crossorigin="anonymous" referrerpolicy="no-referrer"></script>
  <script src="https://cdn.amplify.aws/packages/maplibre-gl-js-
amplify/1.1.0/maplibre-gl-js-amplify.umd.min.js" integrity="sha384-7/
RxWonKW1nM9zCKiwU9x6bkQTjldosg0D1vZYm0Zj+K/vUSnA3s0Mh1RRWAtHPi" crossorigin="anonymous"
  referrerpolicy="no-referrer"></script>
<script>
  // application-specific code
</script>
```

Ini membuat halaman kosong dengan wadah peta.

## Membangun aplikasi: Konfigurasi

Untuk mengkonfigurasi aplikasi Anda menggunakan JavaScript:

1. Masukkan pengidentifikasi kumpulan identitas yang tidak diautentikasi yang Anda buat di [Menggunakan peta - Langkah 2, Siapkan](#) otentikasi.

```
// Cognito Identity Pool ID
const identityPoolId = "region:identityPoolID"; // for example: us-
east-1:123example-1234-5678
// extract the Region from the Identity Pool ID
const region = identityPoolId.split(":")[0];
```

2. Konfigurasi AWS Amplify untuk menggunakan sumber daya yang telah Anda buat, termasuk kumpulan identitas dan sumber daya Peta (ditampilkan di sini dengan nama `defaultexplore.map`).

```
// Configure Amplify
const { Amplify } = aws_amplify_core;
const { createMap } = AmplifyMapLibre;

Amplify.configure({
  Auth: {
    identityPoolId,
    region,
  },
  geo: {
    AmazonLocationService: {
      maps: {
        items: {
          "explore.map": {
            style: "Default style"
          },
        },
        default: "explore.map",
      },
      region,
    },
  }
});
```

## Membangun aplikasi: Inisialisasi peta

Agar peta ditampilkan setelah halaman dimuat, Anda harus menginisialisasi peta. Anda dapat menyesuaikan lokasi peta awal, menambahkan kontrol tambahan, dan data hamparan.

```
async function initializeMap() {
  const map = await createMap(
    {
      container: "map",
      center: [-123.1187, 49.2819],
      zoom: 10,
      hash: true,
    }
  );

  map.addControl(new maplibregl.NavigationControl(), "top-left");
}

initializeMap();
```

### Note

Anda harus memberikan tanda kata atau atribusi teks untuk setiap penyedia data yang Anda gunakan, baik pada aplikasi atau dokumentasi Anda. String atribusi disertakan dalam respons deskriptor gaya di `bawahsources.esri.attribution`, `sources.here.attribution`, dan kunci `sources.grabmaptiles.attribution`. Amplify akan secara otomatis memberikan atribusi. Saat menggunakan sumber daya Lokasi Amazon dengan [penyedia data](#), pastikan untuk membaca [syarat dan ketentuan layanan](#).

## Menjalankan aplikasi

Anda dapat menjalankan contoh aplikasi ini dengan menggunakannya di server web lokal, atau membukanya di browser.

Untuk menggunakan server web lokal, Anda dapat menggunakan `npx`, diinstal sebagai bagian dari Node.js, atau server web lain pilihan Anda. Untuk menggunakan `npx`, ketik `npx serve` dari dalam direktori yang sama dengan `index.html`. Ini melayani aplikasi pada `localhost:5000`.

### Note

Jika kebijakan yang Anda buat untuk peran Amazon Cognito yang tidak diautentikasi menyertakan `referer` suatu kondisi, Anda mungkin diblokir dari pengujian dengan URL.

localhost: Dalam hal ini, Anda dapat menguji dengan server web yang menyediakan URL yang ada dalam kebijakan Anda.

Setelah menyelesaikan tutorial, aplikasi akhir terlihat seperti contoh berikut.

```
<html>
  <head>
    <!-- CSS dependencies -->
    <link href="https://cdn.amplify.aws/packages/maplibre-
gl/1.15.2/maplibre-gl.css" rel="stylesheet" integrity="sha384-
DrPVD9GufrxGb7kWwRv0CywpXTmfvbK0Z5i5pN7urmIThew0zXKTME+gutUgtpeD"
crossorigin="anonymous" referrerpolicy="no-referrer"></link>

    <!-- JavaScript dependencies -->
    <script src="https://cdn.amplify.aws/packages/maplibre-gl/1.15.2/maplibre-gl.js"
integrity="sha384-rwYfkmA0pciZS2bDuwZ/Xa/Gog6jXem8D/whm3wnsZSVFemDDlprcUXHnDDUcrNU"
crossorigin="anonymous" referrerpolicy="no-referrer"></script>
    <script src="https://cdn.amplify.aws/packages/core/4.3.0/aws-amplify-core.min.js"
integrity="sha384-70h+5w0l7XGyYvSqBki2Q7SA5K640V5nyW2/LEbevDQEV1HMJqJLA1A00z2hu8fJ"
crossorigin="anonymous" referrerpolicy="no-referrer"></script>
    <script src="https://cdn.amplify.aws/packages/auth/4.3.8/aws-amplify-auth.min.js"
integrity="sha384-jfkXCEfYyVmDXyKlgWNwv54xRaZgk14m7sjeb2jLVBtUXCD2p+WU8YZ2mPZ9Xbdw"
crossorigin="anonymous" referrerpolicy="no-referrer"></script>
    <script src="https://cdn.amplify.aws/packages/geo/1.1.0/aws-amplify-geo.min.js"
integrity="sha384-TFMTyWuCbiptXTzv0gzJbV8TPUupG1rA1AVrznAhCSpXTIdGw82bGd8RTk5rr3nP"
crossorigin="anonymous" referrerpolicy="no-referrer"></script>
    <script src="https://cdn.amplify.aws/packages/maplibre-gl-js-
amplify/1.1.0/maplibre-gl-js-amplify.umd.min.js" integrity="sha384-7/
RxWonKW1nM9zCKiwU9x6bkQTjldosg0D1vZYm0Zj+K/vUSnA3s0Mh1RRWAtHPi" crossorigin="anonymous"
referrerpolicy="no-referrer"></script>

    <style>
      body { margin: 0; }
      #map { height: 100vh; }
    </style>
  </head>

  <body>
    <div id="map" />
    <script type="module">
      // Cognito Identity Pool ID
```



```
const identityPoolId = "region:identityPoolId"; // for example: us-
east-1:123example-1234-5678
// extract the Region from the Identity Pool ID
const region = identityPoolId.split(":")[0];

// Configure Amplify
const { Amplify } = aws_amplify_core;
const { createMap } = AmplifyMapLibre;

Amplify.configure({
  Auth: {
    identityPoolId,
    region,
  },
  geo: {
    AmazonLocationService: {
      maps: {
        items: {
          "explore.map": {
            style: "Default style"
          },
        },
        default: "explore.map",
      },
      region,
    },
  }
});

async function initializeMap() {
  const map = await createMap(
    {
      container: "map",
      center: [-123.1187, 49.2819],
      zoom: 10,
      hash: true,
    }
  );

  map.addControl(new maplibregl.NavigationControl(), "top-left");
}

initializeMap();
</script>
```

```
</body>
</html>
```

Menjalankan aplikasi ini menampilkan peta layar penuh menggunakan gaya peta yang Anda pilih. Contoh ini juga dijelaskan pada tab Sematkan peta pada halaman sumber daya Peta apa pun di [konsol Amazon Location Service](#).

Setelah Anda menyelesaikan tutorial ini, buka [Tampilkan topik peta](#) dalam AWS Amplify dokumentasi untuk mempelajari lebih lanjut, termasuk cara menampilkan penanda di peta.

## Menggunakan Tangram dengan Amazon Location Service

Bagian ini memberikan tutorial berikut tentang cara mengintegrasikan Tangram dengan Lokasi Amazon.

### Important

Gaya Tangram dalam tutorial berikut hanya kompatibel dengan sumber daya peta Lokasi Amazon yang dikonfigurasi dengan VectorHereContrast gaya.

Berikut ini adalah contoh perintah AWS CLI untuk membuat sumber daya peta baru yang disebut *TangramExampleMap* menggunakan gaya: *VectorHereContrast*

```
aws --region us-east-1 \
  location \
  create-map \
  --map-name "TangramExampleMap" \
  --configuration "Style=VectorHereContrast"
```

### Note

Penagihan ditentukan oleh penggunaan Anda. Anda mungkin dikenakan biaya untuk penggunaan AWS layanan lain. Untuk informasi selengkapnya, lihat [harga Amazon Location Service](#).

## Topik

- [Menggunakan Tangram dengan Amazon Location Service](#)
- [Menggunakan Tangram ES untuk Android dengan Amazon Location Service](#)
- [Menggunakan Tangram ES untuk iOS dengan Amazon Location Service](#)

## Menggunakan Tangram dengan Amazon Location Service

[Tangram](#) adalah mesin pemetaan fleksibel, dirancang untuk rendering real-time peta 2D dan 3D dari ubin vektor. Ini dapat digunakan dengan gaya yang dirancang Mapzen dan ubin HERE yang disediakan oleh Amazon Location Service Maps API. Panduan ini menjelaskan cara mengintegrasikan Tangram dengan Amazon Location dalam JavaScript HTML/aplikasi dasar, meskipun pustaka dan teknik yang sama juga berlaku saat menggunakan kerangka kerja seperti React dan Angular.

Tangram dibangun di atas [Leaflet](#), JavaScript perpustakaan sumber terbuka untuk peta interaktif mobile-friendly. Ini berarti bahwa banyak plugin dan kontrol yang kompatibel dengan Leaflet juga bekerja dengan Tangram.

Gaya Tangram yang dibuat untuk bekerja dengan [skema Tilezen sebagian](#) besar kompatibel dengan Lokasi Amazon saat menggunakan peta dari SINI. Ini termasuk:

- [Bubble Wrap](#) - Gaya wayfinding berfitur lengkap dengan ikon bermanfaat untuk tempat menarik
- [Cinnabar](#) — Tampilan klasik dan pilihan untuk aplikasi pemetaan umum
- [Isi Ulang](#) — Gaya peta minimalis yang dirancang untuk hamparan visualisasi data, terinspirasi oleh gaya Toner mani oleh Stamen Design
- [Tron](#) — Eksplorasi transformasi skala dalam bahasa visual TRON
- [Walkabout](#) — Gaya yang berfokus pada luar ruangan yang sempurna untuk hiking atau keluar dan berkeliling

[Panduan ini menjelaskan cara mengintegrasikan Tangram dengan Lokasi Amazon dalam JavaScript HTML/aplikasi dasar menggunakan Gaya Tangram yang disebut Bubble Wrap.](#) Sampel ini tersedia sebagai bagian dari repositori sampel Amazon Location Service di [GitHub](#)

Sementara gaya Tangram lainnya paling baik disertai dengan ubin raster, yang menyandikan informasi medan, fitur ini belum didukung oleh Amazon Location.

**⚠ Important**

Gaya Tangram dalam tutorial berikut hanya kompatibel dengan sumber daya peta Lokasi Amazon yang dikonfigurasi dengan `VectorHereContrast` gaya.

### Membangun aplikasi: Scaffolding

Aplikasi ini adalah halaman HTML dengan JavaScript untuk membangun peta pada aplikasi web Anda. Buat halaman HTML (`index.html`) dan buat wadah peta:

- Masukkan `div` elemen `id` dengan peta untuk menerapkan dimensi peta ke tampilan peta.
- Dimensi diwarisi dari `viewport`.

```
<html>
  <head>
    <style>
      body {
        margin: 0;
      }

      #map {
        height: 100vh; /* 100% of viewport height */
      }
    </style>
  </head>
  <body>
    <!-- map container -->
    <div id="map" />
  </body>
</html>
```

### Membangun aplikasi: Menambahkan dependensi

Tambahkan dependensi berikut:

- Leaflet dan CSS yang terkait.
- Tangram.
- AWS SDK untuk JavaScript.

```
<!-- CSS dependencies -->
<link
  rel="stylesheet"
  href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css"
  integrity="sha512-xodZBNTC5n17Xt2atTPuE1HxjVMSvLVW9ocqUKLsCC5CXdbqCmblAsh0MAS6/keqq/
  sMzMZ19scR4PsZChSR7A=="
  crossorigin=""
/>
<!-- JavaScript dependencies -->
<script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"></script>
<script src="https://unpkg.com/tangram"></script>
<script src="https://sdk.amazonaws.com/js/aws-sdk-2.784.0.min.js"></script>
<script>
  // application-specific code
</script>
```

Ini membuat halaman kosong dengan prasyarat yang diperlukan. Langkah selanjutnya memandu Anda melalui penulisan JavaScript kode untuk aplikasi Anda.

## Membangun aplikasi: Konfigurasi

Untuk mengonfigurasi aplikasi Anda dengan sumber daya dan kredensial Anda:

1. Masukkan nama dan pengidentifikasi sumber daya Anda.

```
// Cognito Identity Pool ID
const identityPoolId = "us-east-1:54f2ba88-9390-498d-aaa5-0d97fb7ca3bd";
// Amazon Location Service map name; must be HERE-backed
const mapName = "TangramExampleMap";
```

2. Buat instance penyedia kredensi menggunakan kumpulan identitas tidak diautentikasi yang Anda buat di [Menggunakan peta - Langkah 2](#), Siapkan otentikasi. Karena ini menggunakan kredensial di luar alur kerja AWS SDK normal, sesi kedaluwarsa setelah satu jam.

```
// extract the region from the Identity Pool ID; this will be used for both Amazon
  Cognito and Amazon Location
AWS.config.region = identityPoolId.split(":", 1)[0];

// instantiate a Cognito-backed credential provider
const credentials = new AWS.CognitoIdentityCredentials({
  IdentityPoolId: identityPoolId,
});
```

3. Meskipun Tangram memungkinkan Anda untuk mengganti URL yang digunakan untuk mengambil ubin, itu tidak termasuk kemampuan untuk mencegat permintaan sehingga mereka dapat ditandatangani.

Untuk mengatasinya, ganti `sources.mapzen.url` untuk menunjuk ke Lokasi Amazon menggunakan nama host `syntheticamazon.location`, yang akan ditangani oleh pekerja [layanan](#). Berikut ini adalah contoh konfigurasi adegan menggunakan [Bubble Wrap](#):

```
const scene = {
  import: [
    // Bubble Wrap style
    "https://www.nextzen.org/carto/bubble-wrap-style/10/bubble-wrap-style.zip",
    "https://www.nextzen.org/carto/bubble-wrap-style/10/themes/label-7.zip",
    "https://www.nextzen.org/carto/bubble-wrap-style/10/themes/bubble-wrap-road-shields-usa.zip",
    "https://www.nextzen.org/carto/bubble-wrap-style/10/themes/bubble-wrap-road-shields-international.zip",
  ],
  // override values beneath the `sources` key in the style above
  sources: {
    mapzen: {
      // point at Amazon Location using a synthetic URL, which will be handled by the service
      // worker
      url: `https://amazon.location/${mapName}/{z}/{x}/{y}`,
    },
    // effectively disable raster tiles containing encoded normals
    normals: {
      max_zoom: 0,
    },
    "normals-elevation": {
      max_zoom: 0,
    },
  },
};
```

### Membangun aplikasi: Minta transformasi

Untuk mendaftarkan dan menginisialisasi service worker, buat `registerServiceWorker` fungsi yang akan dipanggil sebelum peta diinisialisasi. Ini mendaftarkan JavaScript kode yang disediakan dalam file terpisah yang disebut `sw.js` sebagai pengendali `index.html` pekerja layanan.

Kredensial dimuat dari Amazon Cognito dan diteruskan ke service worker di samping Wilayah untuk memberikan informasi guna menandatangani permintaan ubin [dengan](#) Signature Version 4.

```
/**
 * Register a service worker that will rewrite and sign requests using Signature
 * Version 4.
 */
async function registerServiceWorker() {
  if ("serviceWorker" in navigator) {
    try {
      const reg = await navigator.serviceWorker.register("./sw.js");

      // refresh credentials from Amazon Cognito
      await credentials.refreshPromise();

      await reg.active.ready;

      if (navigator.serviceWorker.controller == null) {
        // trigger a navigate event to active the controller for this page
        window.location.reload();
      }

      // pass credentials to the service worker
      reg.active.postMessage({
        credentials: {
          accessKeyId: credentials.accessKeyId,
          secretAccessKey: credentials.secretAccessKey,
          sessionToken: credentials.sessionToken,
        },
        region: AWS.config.region,
      });
    } catch (error) {
      console.error("Service worker registration failed:", error);
    }
  } else {
    console.warn("Service worker support is required for this example");
  }
}
```

Implementasi Service Worker dalam `sw.js` mendengarkan message peristiwa untuk mengambil kredensi dan perubahan konfigurasi Wilayah. Ini juga bertindak sebagai server proxy dengan mendengarkan fetch acara. `fetchperistiwa` yang menargetkan nama host `amazon.location`

sintetis akan ditulis ulang untuk menargetkan API Lokasi Amazon yang sesuai dan ditandatangani menggunakan Amplify Core. Signer

```
// sw.js
self.importScripts(
  "https://unpkg.com/@aws-amplify/core@3.7.0/dist/aws-amplify-core.min.js"
);

const { Signer } = aws_amplify_core;

let credentials;
let region;

self.addEventListener("install", (event) => {
  // install immediately
  event.waitUntil(self.skipWaiting());
});

self.addEventListener("activate", (event) => {
  // control clients ASAP
  event.waitUntil(self.clients.claim());
});

self.addEventListener("message", (event) => {
  const {
    data: { credentials: newCredentials, region: newRegion },
  } = event;

  if (newCredentials != null) {
    credentials = newCredentials;
  }

  if (newRegion != null) {
    region = newRegion;
  }
});

async function signedFetch(request) {
  const url = new URL(request.url);
  const path = url.pathname.slice(1).split("/");

  // update URL to point to Amazon Location
  url.pathname = `/maps/v0/maps/${path[0]}/tiles/${path.slice(1).join("/")}`;
```



```
url.host = `maps.geo.${region}.amazonaws.com`;
// strip params (Tangram generates an empty api_key param)
url.search = "";

const signed = Signer.signUrl(url.toString(), {
  access_key: credentials.accessKeyId,
  secret_key: credentials.secretAccessKey,
  session_token: credentials.sessionToken,
});

return fetch(signed);
}

self.addEventListener("fetch", (event) => {
  const { request } = event;

  // match the synthetic hostname we're telling Tangram to use
  if (request.url.includes("amazon.location")) {
    return event.respondWith(signedFetch(request));
  }

  // fetch normally
  return event.respondWith(fetch(request));
});
```

Untuk memperbarui kredensial secara otomatis dan mengirimkannya ke service worker sebelum kedaluwarsa, gunakan fungsi berikut di dalam: `index.html`

```
async function refreshCredentials() {
  await credentials.refreshPromise();

  if ("serviceWorker" in navigator) {
    const controller = navigator.serviceWorker.controller;

    controller.postMessage({
      credentials: {
        accessKeyId: credentials.accessKeyId,
        secretAccessKey: credentials.secretAccessKey,
        sessionToken: credentials.sessionToken,
      },
    });
  } else {
    console.warn("Service worker support is required for this example.");
  }
}
```

```
}  
  
// schedule the next credential refresh when they're about to expire  
setTimeout(refreshCredentials, credentials.expireTime - new Date());  
}
```

## Membangun aplikasi: Inisialisasi peta

Agar peta ditampilkan setelah halaman dimuat, Anda harus menginisialisasi peta. Anda memiliki opsi untuk menyesuaikan lokasi peta awal, menambahkan kontrol tambahan, dan data hamparan.

### Note

Anda harus memberikan tanda kata atau atribusi teks untuk setiap penyedia data yang Anda gunakan, baik pada aplikasi atau dokumentasi Anda. String atribusi disertakan dalam respons deskriptor gaya di `bawahsources.esri.attribution`, `sources.here.attribution`, dan kunci `source.grabmaptiles.attribution`

Karena Tangram tidak meminta sumber daya ini, dan hanya kompatibel dengan peta dari SINI, gunakan “© 2020 DI SINI”. Saat menggunakan sumber daya Lokasi Amazon dengan [penyedia data](#), pastikan untuk membaca [syarat dan ketentuan layanan](#).

```
/**  
 * Initialize a map.  
 */  
async function initializeMap() {  
  // register the service worker to handle requests to https://amazon.location  
  await registerServiceWorker();  
  
  // Initialize the map  
  const map = L.map("map").setView([49.2819, -123.1187], 10);  
  Tangram.leafletLayer({  
    scene,  
  }).addTo(map);  
  map.attributionControl.setPrefix("");  
  map.attributionControl.addAttribution("© 2020 HERE");  
}  
  
initializeMap();
```

## Menjalankan aplikasi

Untuk menjalankan sampel ini, Anda dapat:

- Gunakan host yang mendukung HTTPS,
- Gunakan server web lokal untuk mematuhi batasan keamanan pekerja layanan.

Untuk menggunakan server web lokal, Anda dapat menggunakan npx, karena itu diinstal sebagai bagian dari Node.js. Anda dapat menggunakan npx `serve` dari dalam direktori yang sama dengan `index.html` dan `sw.js`. Ini melayani aplikasi di [localhost: 5000](http://localhost:5000).

Berikut ini adalah `index.html` filenya:

```
<!-- index.html -->
<html>
  <head>
    <link
      rel="stylesheet"
      href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css"
      integrity="sha512-xodZBNTC5n17Xt2atTPuE1HxjVMSvLVW9ocqUKLsCC5CXdbqCmblAsh0MAS6/
keqq/sMzMZ19scR4PsZChSR7A=="
      crossorigin=""
    />
    <style>
      body {
        margin: 0;

        #map {
          height: 100vh;
        }
      }
    </style>
  </head>

  <body>
    <div id="map" />
    <script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"></script>
    <script src="https://unpkg.com/tangram"></script>
    <script src="https://sdk.amazonaws.com/js/aws-sdk-2.784.0.min.js"></script>
    <script>
      // configuration
      // Cognito Identity Pool ID
```

```
const identityPoolId = "<Identity Pool ID>";
// Amazon Location Service Map name; must be HERE-backed
const mapName = "<Map name>";

AWS.config.region = identityPoolId.split(":")[0];

// instantiate a credential provider
credentials = new AWS.CognitoIdentityCredentials({
  IdentityPoolId: identityPoolId,
});

const scene = {
  import: [
    // Bubble Wrap style
    "https://www.nextzen.org/carto/bubble-wrap-style/10/bubble-wrap-style.zip",
    "https://www.nextzen.org/carto/bubble-wrap-style/10/themes/label-7.zip",
    "https://www.nextzen.org/carto/bubble-wrap-style/10/themes/bubble-wrap-road-
shields-usa.zip",
    "https://www.nextzen.org/carto/bubble-wrap-style/10/themes/bubble-wrap-road-
shields-international.zip",
  ],
  // override values beneath the `sources` key in the style above
  sources: {
    mapzen: {
      // point at Amazon Location using a synthetic URL, which will be handled by
the service
      // worker
      url: `https://amazon.location/${mapName}/{z}/{x}/{y}`,
    },
    // effectively disable raster tiles containing encoded normals
    normals: {
      max_zoom: 0,
    },
    "normals-elevation": {
      max_zoom: 0,
    },
  },
};

/**
 * Register a service worker that will rewrite and sign requests using Signature
Version 4.
 */
async function registerServiceWorker() {
```

```
if ("serviceWorker" in navigator) {
  try {
    const reg = await navigator.serviceWorker.register("./sw.js");

    // refresh credentials from Amazon Cognito
    await credentials.refreshPromise();

    await reg.active.ready;

    if (navigator.serviceWorker.controller == null) {
      // trigger a navigate event to active the controller for this page
      window.location.reload();
    }

    // pass credentials to the service worker
    reg.active.postMessage({
      credentials: {
        accessKeyId: credentials.accessKeyId,
        secretAccessKey: credentials.secretAccessKey,
        sessionToken: credentials.sessionToken,
      },
      region: AWS.config.region,
    });
  } catch (error) {
    console.error("Service worker registration failed:", error);
  }
  else {
    console.warn("Service Worker support is required for this example");
  }
}

/**
 * Initialize a map.
 */
async function initializeMap() {
  // register the service worker to handle requests to https://amazon.location
  await registerServiceWorker();

  // Initialize the map
  const map = L.map("map").setView([49.2819, -123.1187], 10);
  Tangram.leafletLayer({
    scene,
  }).addTo(map);
  map.attributionControl.setPrefix("");
}
```

```
    map.attributionControl.addAttribution("© 2020 HERE");
  }

  initializeMap();
</script>
</body>
</html>
```

Berikut ini adalah sw.js filenya:

```
// sw.js
self.importScripts(
  "https://unpkg.com/@aws-amplify/core@3.7.0/dist/aws-amplify-core.min.js"
);

const { Signer } = aws_amplify_core;

let credentials;
let region;

self.addEventListener("install", (event) => {
  // install immediately
  event.waitUntil(self.skipWaiting());
});

self.addEventListener("activate", (event) => {
  // control clients ASAP
  event.waitUntil(self.clients.claim());
});

self.addEventListener("message", (event) => {
  const {
    data: { credentials: newCredentials, region: newRegion },
  } = event;

  if (newCredentials !== null) {
    credentials = newCredentials;
  }

  if (newRegion !== null) {
    region = newRegion;
  }
});
```

```
async function signedFetch(request) {
  const url = new URL(request.url);
  const path = url.pathname.slice(1).split("/");

  // update URL to point to Amazon Location
  url.pathname = `/maps/v0/maps/${path[0]}/tiles/${path.slice(1).join("/")}`;
  url.host = `maps.geo.${region}.amazonaws.com`;
  // strip params (Tangram generates an empty api_key param)
  url.search = "";

  const signed = Signer.signUrl(url.toString(), {
    access_key: credentials.accessKeyId,
    secret_key: credentials.secretAccessKey,
    session_token: credentials.sessionToken,
  });

  return fetch(signed);
}

self.addEventListener("fetch", (event) => {
  const { request } = event;

  // match the synthetic hostname we're telling Tangram to use
  if (request.url.includes("amazon.location")) {
    return event.respondWith(signedFetch(request));
  }

  // fetch normally
  return event.respondWith(fetch(request));
});
```

Sampel ini tersedia sebagai bagian dari repositori sampel Amazon Location Service di [GitHub](#)

Menggunakan Tangram ES untuk Android dengan Amazon Location Service

[Tangram ES](#) adalah pustaka C++ untuk merender peta 2D dan 3D dari data vektor menggunakan OpenGL ES. Ini adalah mitra asli [Tangram](#).

Gaya Tangram yang dibuat untuk bekerja dengan [skema Tilezen sebagian](#) besar kompatibel dengan Lokasi Amazon saat menggunakan peta dari SINI. Ini termasuk:

- [Bubble Wrap](#) - Gaya pencarian jalan berfitur lengkap dengan ikon bermanfaat untuk tempat menarik.
- [Cinnabar](#) — Tampilan klasik dan pilihan untuk aplikasi pemetaan umum.
- [Isi Ulang](#) — Gaya peta minimalis yang dirancang untuk hamparan visualisasi data, terinspirasi oleh gaya Toner mani oleh Stamen Design.
- [Tron](#) — Eksplorasi transformasi skala dalam bahasa visual TRON.
- [Walkabout](#) — Gaya yang berfokus pada luar ruangan yang sempurna untuk hiking atau keluar dan berkeliling.

Panduan ini menjelaskan cara mengintegrasikan Tangram ES untuk Android dengan Lokasi Amazon menggunakan gaya Tangram yang disebut Cinnabar. Sampel ini tersedia sebagai bagian dari repositori sampel Amazon Location Service di [GitHub](#)

Sementara gaya Tangram lainnya paling baik disertai dengan ubin raster, yang menyandikan informasi medan, fitur ini belum didukung oleh Amazon Location.

#### Important

Gaya Tangram dalam tutorial berikut hanya kompatibel dengan sumber daya peta Lokasi Amazon yang dikonfigurasi dengan `VectorHereContrast` gaya.

## Membangun aplikasi: Inisialisasi

Untuk menginisialisasi aplikasi Anda:

1. Buat project Android Studio baru dari template Empty Activity.
2. Pastikan bahwa Kotlin dipilih untuk bahasa proyek.
3. Pilih SDK Minimum API 16: Android 4.1 (Jelly Bean) atau yang lebih baru.
4. Buka Struktur Proyek untuk memilih File, Struktur Proyek... , dan pilih bagian Dependensi.
5. Dengan <All Modules>dipilih, pilih tombol + untuk menambahkan Ketergantungan Perpustakaan baru.
6. Tambahkan AWS Android SDK versi 2.19.1 atau yang lebih baru. Sebagai contoh:  
`com.amazonaws:aws-android-sdk-core:2.19.1`
7. Tambahkan Tangram versi 0.13.0 atau yang lebih baru. Misalnya:  
`com.mapzen.tangram:tangram:0.13.0`



**Note**

Mencari Tangram: `com.mapzen.tangram:tangram:0.13.0` akan menghasilkan pesan bahwa itu "tidak ditemukan", tetapi memilih OK akan memungkinkannya untuk ditambahkan.

## Membangun aplikasi: Konfigurasi

Untuk mengonfigurasi aplikasi Anda dengan sumber daya dan AWS Wilayah Anda:

1. Buat `app/src/main/res/values/configuration.xml`.
2. Masukkan nama dan pengidentifikasi sumber daya Anda, dan juga AWS Wilayah tempat mereka dibuat:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="identityPoolId">us-east-1:54f2ba88-9390-498d-aaa5-0d97fb7ca3bd</string>
  <string name="mapName">TangramExampleMap</string>
  <string name="awsRegion">us-east-1</string>
  <string name="sceneUrl">https://www.nextzen.org/carto/cinnabar-style/9/cinnabar-style.zip</string>
  <string name="attribution">© 2020 HERE</string>
</resources>
```

## Membangun aplikasi: Tata letak aktivitas

Sunting `app/src/main/res/layout/activity_main.xml`:

- Tambahkan `aMapView`, yang membuat peta. Ini juga akan mengatur titik pusat awal peta.
- Tambahkan `aTextView`, yang menampilkan atribusi.

Ini juga akan mengatur titik pusat awal peta.

**Note**

Anda harus memberikan tanda kata atau atribusi teks untuk setiap penyedia data yang Anda gunakan, baik pada aplikasi atau dokumentasi Anda. String atribusi disertakan dalam respons deskriptor gaya di `bawahsources.esri.attribution,sources.here.attribution`, dan kunci `source.grabmaptiles.attribution`

Karena Tangram tidak meminta sumber daya ini, dan hanya kompatibel dengan peta dari SINI, gunakan "© 2020 DI SINI". Saat menggunakan sumber daya Lokasi Amazon dengan [penyedia data](#), pastikan untuk membaca [syarat dan ketentuan layanan](#).

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <com.mapzen.tangram.MapView
        android:id="@+id/map"
        android:layout_height="match_parent"
        android:layout_width="match_parent" />

    <TextView
        android:id="@+id/attributionView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#80808080"
        android:padding="5sp"
        android:textColor="@android:color/black"
        android:textSize="10sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        tools:ignore="SmallSp" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

## Membangun aplikasi: Minta transformasi

Buat kelas bernama `SigV4Interceptor` untuk mencegah AWS permintaan dan menandatangani menggunakan [Signature Version 4](#). Ini akan terdaftar dengan klien HTTP yang digunakan untuk mengambil sumber daya peta saat Aktivitas Utama dibuat.

```
package aws.location.demo.okhttp

import com.amazonaws.DefaultRequest
import com.amazonaws.auth.AWS4Signer
import com.amazonaws.auth.AWSCredentialsProvider
import com.amazonaws.http.HttpMethodName
import com.amazonaws.util.IOUtils
import okhttp3.HttpUrl
import okhttp3.Interceptor
import okhttp3.Request
import okhttp3.Response
import okio.Buffer
import java.io.ByteArrayInputStream
import java.net.URI

class SigV4Interceptor(
    private val credentialsProvider: AWSCredentialsProvider,
    private val serviceName: String
) : Interceptor {
    override fun intercept(chain: Interceptor.Chain): Response {
        val originalRequest = chain.request()

        if (originalRequest.url().host().contains("amazonaws.com")) {
            val signer = if (originalRequest.url().encodedPath().contains("@")) {
                // the presence of "@" indicates that it doesn't need to be double URL-
                encoded
                AWS4Signer(false)
            } else {
                AWS4Signer()
            }

            val awsRequest = toAWSRequest(originalRequest, serviceName)
            signer.setServiceName(serviceName)
            signer.sign(awsRequest, credentialsProvider.credentials)

            return chain.proceed(toSignedOkHttpRequest(awsRequest, originalRequest))
        }
    }
}
```

```
        return chain.proceed(originalRequest)
    }

    companion object {
        fun toAWSRequest(request: Request, serviceName: String): DefaultRequest<Any> {
            // clone the request (AWS-style) so that it can be populated with
credentials
            val dr = DefaultRequest<Any>(serviceName)

            // copy request info
            dr.httpMethod = HttpMethodName.valueOf(request.method())
            with(request.url()) {
                dr.resourcePath = uri().path
                dr.endpoint = URI.create("${scheme()}://${host()}")

                // copy parameters
                for (p in queryParameterNames()) {
                    if (p != "") {
                        dr.addParameter(p, queryParameter(p))
                    }
                }
            }

            // copy headers
            for (h in request.headers().names()) {
                dr.addHeader(h, request.header(h))
            }

            // copy the request body
            val bodyBytes = request.body()?.let { body ->
                val buffer = Buffer()
                body.writeTo(buffer)
                IOUtils.toByteArray(buffer.inputStream())
            }

            dr.content = ByteArrayInputStream(bodyBytes ?: ByteArray(0))

            return dr
        }

        fun toSignedOkHttpRequest(
            awsRequest: DefaultRequest<Any>,
            originalRequest: Request
```

```
    ): Request {
        // copy signed request back into an OkHttp Request
        val builder = Request.Builder()

        // copy headers from the signed request
        for ((k, v) in awsRequest.headers) {
            builder.addHeader(k, v)
        }

        // start building an HttpUrl
        val urlBuilder = HttpUrl.Builder()
            .host(awsRequest.endpoint.host)
            .scheme(awsRequest.endpoint.scheme)
            .encodedPath(awsRequest.resourcePath)

        // copy parameters from the signed request
        for ((k, v) in awsRequest.parameters) {
            urlBuilder.addQueryParameter(k, v)
        }

        return builder.url(urlBuilder.build())
            .method(originalRequest.method(), originalRequest.body())
            .build()
    }
}
```

## Membangun aplikasi: Kegiatan utama

Aktivitas Utama bertanggung jawab untuk menginisialisasi tampilan yang akan ditampilkan kepada pengguna. Ini melibatkan:

- Membuat Instantiasi Amazon Cognito. `CredentialsProvider`
- Mendaftarkan pencegat Signature Version 4.
- Mengkonfigurasi peta dengan mengarahkannya ke gaya peta, mengganti URL ubin, dan menampilkan atribusi yang sesuai.

`MainActivity` juga bertanggung jawab untuk meneruskan peristiwa siklus hidup ke tampilan peta.

```
package aws.location.demo.tangram
```

```
import android.os.Bundle
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import aws.location.demo.okhttp.SigV4Interceptor
import com.amazonaws.auth.CognitoCachingCredentialsProvider
import com.amazonaws.regions.Regions
import com.mapzen.tangram.*
import com.mapzen.tangram.networking.DefaultHttpHandler
import com.mapzen.tangram.networking.HttpHandler

private const val SERVICE_NAME = "geo"

class MainActivity : AppCompatActivity(), MapView.MapReadyCallback {
    private var mapView: MapView? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        setContentView(R.layout.activity_main)

        mapView = findViewById(R.id.map)

        mapView?.getMapAsync(this, getHttpHandler())
        findViewById<TextView>(R.id.attributionView).text =
        getString(R.string.attribution)
    }

    override fun onMapReady(mapController: MapController?) {
        val sceneUpdates = arrayListOf(
            SceneUpdate(
                "sources.mapzen.url",
                "https://maps.geo.${getString(R.string.awsRegion)}.amazonaws.com/maps/
v0/maps/${
                    getString(
                        R.string.mapName
                    )
                }/tiles/{z}/{x}/{y}"
            )
        )

        mapController?.let { map ->
            map.updateCameraPosition(
                CameraUpdateFactory.newLngLatZoom(
                    LngLat(-123.1187, 49.2819),
```

```
        12F
    )
    )
    map.loadSceneFileAsync(
        getString(R.string.sceneUrl),
        sceneUpdates
    )
}

private fun getHttpHandler(): HttpHandler {
    val builder = DefaultHttpHandler.getClientBuilder()

    val credentialsProvider = CognitoCachingCredentialsProvider(
        applicationContext,
        getString(R.string.identityPoolId),
        Regions.US_EAST_1
    )

    return DefaultHttpHandler(
        builder.addInterceptor(
            SigV4Interceptor(
                credentialsProvider,
                SERVICE_NAME
            )
        )
    )
}

override fun onResume() {
    super.onResume()
    mapView?.onResume()
}

override fun onPause() {
    super.onPause()
    mapView?.onPause()
}

override fun onLowMemory() {
    super.onLowMemory()
    mapView?.onLowMemory()
}
```

```
override fun onDestroy() {  
    super.onDestroy()  
    mapView?.onDestroy()  
}  
}
```

Menjalankan aplikasi ini menampilkan peta layar penuh dengan gaya yang Anda pilih. Sampel ini tersedia sebagai bagian dari repositori sampel Amazon Location Service di [GitHub](#)

Menggunakan Tangram ES untuk iOS dengan Amazon Location Service

[Tangram ES](#) adalah pustaka C++ untuk merender peta 2D dan 3D dari data vektor menggunakan OpenGL ES. Ini adalah mitra asli [Tangram](#).

Gaya Tangram yang dibuat untuk bekerja dengan [skema Tilezen sebagian](#) besar kompatibel dengan Lokasi Amazon saat menggunakan peta dari SINI. Ini termasuk:

- [Bubble Wrap](#) - Gaya wayfinding berfitur lengkap dengan ikon bermanfaat untuk tempat menarik
- [Cinnabar](#) — Tampilan klasik dan pilihan untuk aplikasi pemetaan umum
- [Isi Ulang](#) — Gaya peta minimalis yang dirancang untuk hamparan visualisasi data, terinspirasi oleh gaya Toner mani oleh Stamen Design
- [Tron](#) — Eksplorasi transformasi skala dalam bahasa visual TRON
- [Walkabout](#) — Gaya yang berfokus pada luar ruangan yang sempurna untuk hiking atau keluar dan berkeliling

Panduan ini menjelaskan cara mengintegrasikan Tangram ES untuk iOS dengan Lokasi Amazon menggunakan gaya Tangram yang disebut Cinnabar. Sampel ini tersedia sebagai bagian dari repositori sampel Amazon Location Service di [GitHub](#)

Sementara gaya Tangram lainnya paling baik disertai dengan ubin raster, yang menyandikan informasi medan, fitur ini belum didukung oleh Amazon Location.

#### Important

Gaya Tangram dalam tutorial berikut hanya kompatibel dengan sumber daya peta Lokasi Amazon yang dikonfigurasi dengan VectorHereContrast gaya.



## Membangun aplikasi: Inisialisasi

Untuk menginisialisasi aplikasi:

1. Buat proyek Xcode baru dari template App.
2. Pilih SwiftUI untuk antarmukanya.
3. Pilih aplikasi SwiftUI untuk Siklus Hidupnya.
4. Pilih Swift untuk bahasanya.

## Membangun aplikasi: Tambahkan dependensi

Untuk menambahkan dependensi, Anda dapat menggunakan manajer dependensi, seperti:

### [CocoaPods](#)

1. Di terminal Anda, instal CocoaPods:

```
sudo gem install cocoapods
```

2. Arahkan ke direktori proyek aplikasi Anda dan inisialisasi Podfile dengan manajer CocoaPods paket:

```
pod init
```

3. Buka Podfile untuk menambahkan AWSCore dan Tangram-es sebagai dependensi:

```
platform :ios, '12.0'  
  
target 'Amazon Location Service Demo' do  
  use_frameworks!  
  
  pod 'AWSCore'  
  pod 'Tangram-es'  
end
```

4. Unduh dan instal dependensi:

```
pod install --repo-update
```

5. Buka ruang kerja Xcode yang membuat CocoaPods :

```
xed .
```

## Membangun aplikasi: Konfigurasi

Tambahkan kunci dan nilai berikut ke Info.plist untuk mengkonfigurasi aplikasi dan menonaktifkan telemetri:

Kunci	Nilai
AWSRegion	us-east-1
IdentityPoolId	us-timur- 1:54 f2ba88-9390-498d-aaa5-0d97f b7ca3bd
MapName	ExampleMap
SceneUrl	https://www.nextzen.org/carto/cinnabar-style/9/cinnabar-style.zip

## Membangun aplikasi: ContentView tata letak

Untuk merender peta, edit `ContentView.swift`:

- Tambahkan `MapView` yang membuat peta.
- Tambahkan `TextField` yang menampilkan atribusi.

Ini juga menetapkan titik pusat awal peta.

### Note

Anda harus memberikan tanda kata atau atribusi teks untuk setiap penyedia data yang Anda gunakan, baik pada aplikasi atau dokumentasi Anda. String atribusi disertakan dalam respons deskriptor gaya di `bawahsources.esri.attribution,sources.here.attribution`, dan kunci `source.grabmaptiles.attribution` Saat menggunakan sumber daya Lokasi Amazon dengan [penyedia data](#), pastikan untuk membaca [syarat dan ketentuan layanan](#).

```

import SwiftUI
import TangramMap

struct ContentView: View {
    var body: some View {
        MapView()
            .cameraPosition(TGCameraPosition(
                center: CLLocationCoordinate2DMake(49.2819, -123.1187),
                zoom: 10,
                bearing: 0,
                pitch: 0))
            .edgesIgnoringSafeArea(.all)
            .overlay(
                Text("© 2020 HERE")
                    .disabled(true)
                    .font(.system(size: 12, weight: .light, design: .default))
                    .foregroundColor(.black)
                    .background(Color.init(Color.RGBColorSpace.sRGB, white: 0.5,
opacity: 0.5))
                    .cornerRadius(1),
                alignment: .bottomTrailing)
    }
}

struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}

```

## Membangun aplikasi: Minta transformasi

Buat file Swift baru bernama `AWSSignatureV4URLHandler.swift` berisi definisi kelas berikut untuk mencegah AWS permintaan dan menandatangani menggunakan [Signature Version 4](#). Ini akan didaftarkan sebagai penangan URL dalam MapView Tangram.

```

import AWSCore
import TangramMap

class AWSSignatureV4URLHandler: TGDefaultURLHandler {
    private let region: AWSRegionType
    private let identityPoolId: String
    private let credentialsProvider: AWSCredentialsProvider

```

```
init(region: AWSRegionType, identityPoolId: String) {
    self.region = region
    self.identityPoolId = identityPoolId
    self.credentialsProvider = AWSCognitoCredentialsProvider(regionType: region,
identityPoolId: identityPoolId)
    super.init()
}

override func downloadRequestAsync(_ url: URL, completionHandler: @escaping
TGDownloadCompletionHandler) -> UInt {
    if url.host?.contains("amazonaws.com") != true {
        // not an AWS URL
        return super.downloadRequestAsync(url, completionHandler:
completionHandler)
    }

    // URL-encode spaces, etc.
    let keyPath = String(url.path.dropFirst())
    guard let keyPathSafe =
keyPath.addingPercentEncoding(withAllowedCharacters: .urlPathAllowed) else {
        print("Invalid characters in path '\(keyPath)'; unsafe to sign")
        return super.downloadRequestAsync(url, completionHandler:
completionHandler)
    }

    // sign the URL
    let endpoint = AWSEndpoint(region: region, serviceName: "geo", url: url)
    let requestHeaders: [String: String] = ["host": endpoint!.hostName]
    let task = AWSSignatureV4Signer
        .generateQueryStringForSignatureV4(
        withCredentialProvider: credentialsProvider,
        httpMethod: .GET,
        expireDuration: 60,
        endpoint: endpoint!,
        keyPath: keyPathSafe,
        requestHeaders: requestHeaders,
        requestParameters: .none,
        signBody: true)
    task.waitUntilFinished()

    if let error = task.error as NSError? {
        print("Error occurred: \(error)")
    }
}
```

```

        if let result = task.result {
            // have Tangram fetch the signed URL
            return super.downloadRequestAsync(result as URL, completionHandler:
completionHandler)
        }

        // fall back to an unsigned URL
        return super.downloadRequestAsync(url, completionHandler: completionHandler)
    }
}

```

## Membangun aplikasi: Tampilan peta

Tampilan peta bertanggung jawab untuk menginisialisasi instance `AWSSignatureV4Delegate` dan mengonfigurasi yang mendasarinya `MGLMapView`, yang mengambil sumber daya dan merender peta. Ini juga menangani penyebaran string atribusi dari sumber deskriptor gaya kembali ke `ContentView`

Buat file Swift baru bernama `MapView.swift` berisi struct definisi berikut:

```

import AWSCore
import TangramMap
import SwiftUI

struct MapView: UIViewRepresentable {
    private let mapView: TGMapView

    init() {
        let regionName = Bundle.main.object(forKey: "AWSRegion") as!
String
        let identityPoolId = Bundle.main.object(forKey: "IdentityPoolId")
as! String
        let mapName = Bundle.main.object(forKey: "MapName") as! String
        let sceneURL = URL(string: Bundle.main.object(forKey: "SceneURL")
as! String)!

        let region = (regionName as NSString).aws_regionTypeValue()

        // rewrite tile URLs to point at AWS resources
        let sceneUpdates = [
            TGSceneUpdate(path: "sources.mapzen.url",

```

```
        value: "https://maps.geo.\(regionName).amazonaws.com/maps/v0/
maps/\(mapName)/tiles/{z}/{x}/{y}")]

    // instantiate a TGURLHandler that will sign AWS requests
    let urlHandler = AWSSignatureV4URLHandler(region: region, identityPoolId:
identityPoolId)

    // instantiate the map view and attach the URL handler
    mapView = TGMapView(frame: .zero, urlHandler: urlHandler)

    // load the map style and apply scene updates (properties modified at runtime)
    mapView.loadScene(from: sceneURL, with: sceneUpdates)
}

func cameraPosition(_ cameraPosition: TGCameraPosition) -> MapView {
    mapView.cameraPosition = cameraPosition

    return self
}

// MARK: - UIViewRepresentable protocol

func makeUIView(context: Context) -> TGMapView {
    return mapView
}

func updateUIView(_ uiView: TGMapView, context: Context) {
}
}
```

Menjalankan aplikasi ini menampilkan peta layar penuh dengan gaya yang Anda pilih. Sampel ini tersedia sebagai bagian dari repositori sampel Amazon Location Service di [GitHub](#)

## Menggambar fitur data pada peta

Setelah Anda memiliki aplikasi yang membuat peta, menggunakan Amplify MapLibre,, atau Tangram untuk merender peta, langkah alami berikutnya adalah menggambar fitur di atas peta. Misalnya, Anda mungkin ingin menjadikan lokasi pelanggan Anda sebagai penanda di peta.

Secara umum, Anda dapat menggunakan [fungsi pencarian Tempat](#) untuk menemukan lokasi dari data Anda, lalu menggunakan fungsionalitas Amplify, MapLibre, atau Tangram untuk merender lokasi.

Untuk melihat sampel rendering berbagai jenis objek di peta, lihat MapLibre contoh berikut:

- [Contoh: Gambar spidol](#)
- [Contoh: Gambar poin berkerumun](#)
- [Contoh: Menggambar poligon](#)

Untuk sampel dan tutorial lainnya, lihat [Contoh kode dan tutorial untuk bekerja dengan Amazon Location Service](#).

## Mengatur luasan untuk peta menggunakan MapLibre

Ada kalanya Anda tidak ingin pengguna Anda dapat menggeser atau memperbesar seluruh dunia. Jika Anda menggunakan MapLibre kontrol peta, Anda dapat membatasi luasan, atau batas, kontrol peta dengan `maxBounds` opsi, dan membatasi zoom dengan `minZoom` dan opsi. `maxZoom`

Contoh kode berikut menunjukkan cara menginisialisasi kontrol peta untuk membatasi panning ke batas tertentu (dalam hal ini, luasan sumber data Grab).

### Note

Sampel ini masuk JavaScript, dan bekerja dalam konteks [Membuat aplikasi web](#) tutorial.

```
// Set bounds to Grab data provider region
var bounds = [
  [90.0, -21.943045533438166], // Southwest coordinates
  [146.25, 31.952162238024968] // Northeast coordinates
];

var mglMap = new maplibregl.Map(
  {
    container: 'map',
    style: mapName,
    maxBounds: bounds // Sets bounds as max
    transformRequest,
  }
);
```

Demikian pula, Anda dapat mengatur tingkat zoom minimum dan maksimum untuk peta. Nilai untuk keduanya bisa antara 0 dan 24, meskipun defaultnya adalah 0 untuk zoom minimum dan 22 untuk

maksimum (penyedia data mungkin tidak menyediakan data di semua tingkat zoom. Sebagian besar pustaka peta menangani ini secara otomatis). Contoh berikut menginisialisasi `minZoom` dan `maxZoom` opsi pada kontrol MapLibre Peta.

```
// Set the minimum and maximum zoom levels
var mglMap = new maplibregl.Map(
  {
    container: 'map',
    style: mapName,
    maxZoom: 12,
    minZoom: 5,
    transformRequest,
  }
);
```

### Tip

Kontrol MapLibre Peta juga memungkinkan pengaturan opsi ini saat runtime, bukan selama inisialisasi, dengan `get...` dan `set...` fungsi. Misalnya, gunakan `getMaxBounds` dan `setMaxBounds` untuk mengubah batas peta saat runtime.

## Mengelola sumber daya peta Anda

Anda dapat mengelola sumber daya peta menggunakan konsol Lokasi Amazon, API Lokasi AmazonAWS CLI, atau Amazon Location.

### Daftar sumber daya peta

Anda dapat melihat daftar sumber daya peta menggunakan konsol Lokasi Amazon, API Lokasi AmazonAWS CLI, atau Amazon Location.

#### Console

Untuk melihat daftar sumber daya peta yang ada menggunakan konsol Lokasi Amazon

1. Buka konsol Lokasi Amazon di <https://console.aws.amazon.com/location/>.
2. Pilih Maps dari panel navigasi kiri.
3. Lihat daftar sumber daya peta Anda di bawah Peta saya.



## API

Gunakan [ListMaps](#) operasi dari Amazon Location Maps API.

Contoh berikut adalah permintaan API untuk mendapatkan daftar sumber daya peta di AWS akun.

```
POST /maps/v0/list-maps
```

Berikut ini adalah contoh respons untuk [ListMaps](#):

```
{
  "Entries": [
    {
      "CreateTime": 2020-10-30T01:38:36Z,
      "DataSource": "Esri",
      "Description": "string",
      "MapName": "ExampleMap",
      "UpdateTime": 2020-10-30T01:38:36Z
    }
  ],
  "NextToken": "1234-5678-9012"
}
```

## CLI

Gunakan perintah [list-map](#).

Contoh berikut adalah AWS CLI untuk mendapatkan daftar sumber daya peta di AWS akun.

```
aws location list-maps
```

## Dapatkan detail sumber daya peta

Anda bisa mendapatkan detail tentang sumber daya peta apa pun di akun AWS Anda menggunakan konsol Lokasi Amazon, API Lokasi Amazon AWS CLI, atau Amazon Location.

### Console

Untuk melihat detail sumber daya peta menggunakan konsol Lokasi Amazon

1. Buka konsol Lokasi Amazon di <https://console.aws.amazon.com/location/>.

2. Pilih Maps dari panel navigasi kiri.
3. Di bawah Peta saya, pilih tautan nama sumber daya peta target.

## API

Gunakan [DescribeMap](#) operasi dari Amazon Location Maps API.

Contoh berikut adalah permintaan API untuk mendapatkan detail sumber daya peta *ExampleMap*.

```
GET /maps/v0/maps/ExampleMap
```

Berikut ini adalah contoh respons untuk [DescribeMap](#):

```
{
  "Configuration": {
    "Style": "VectorEsriNavigation"
  },
  "CreateTime": 2020-10-30T01:38:36Z,
  "DataSource": "Esri",
  "Description": "string",
  "MapArn": "arn:aws:geo:us-west-2:123456789012:maps/ExampleMap",
  "MapName": "ExampleMap",
  "Tags": {
    "Tag1" : "Value1"
  },
  "UpdateTime": 2020-10-30T01:40:36Z
}
```

## CLI

Gunakan perintah [describe-map](#).

Contoh berikut adalah AWS CLI untuk mendapatkan detail sumber daya peta untuk *ExampleMap*.

```
aws location describe-map \  
  --map-name "ExampleMap"
```

## Hapus sumber daya peta

Anda dapat menghapus sumber daya peta dari AWS akun menggunakan konsol Lokasi Amazon, API Lokasi AmazonAWS CLI, atau Amazon Location.

### Warning

Operasi ini menghapus sumber daya secara permanen.

## Console

Untuk menghapus sumber daya peta yang ada menggunakan konsol Lokasi Amazon

1. Buka konsol Lokasi Amazon di <https://console.aws.amazon.com/location/>.
2. Pilih Maps dari panel navigasi kiri.
3. Di bawah Daftar peta saya, pilih peta target dari daftar.
4. Pilih Hapus peta.

## API

Gunakan [DeleteMap](#) operasi dari Amazon Location Maps API.

Contoh berikut adalah permintaan API untuk menghapus sumber daya peta *ExampleMap*.

```
DELETE /maps/v0/maps/ExampleMap
```

Berikut ini adalah contoh respon sukses untuk [DeleteMap](#):

```
HTTP/1.1 200
```

## CLI

Gunakan perintah [delete-map](#).

Contoh berikut adalah AWS CLI perintah untuk menghapus sumber daya peta *ExampleMap*.

```
aws location delete-map \  
  --map-name "ExampleMap"
```

# Mencari data tempat dan geolokasi menggunakan Amazon Location

Lokasi Amazon mencakup kemampuan untuk mencari geolokasi, atau tempat, data penyedia yang Anda pilih. Ada beberapa jenis pencarian yang tersedia.

- **Geocoding** adalah proses mencari alamat, wilayah, nama bisnis, atau tempat menarik lainnya, berdasarkan input teks. Ini mengembalikan detail dan lokasi (dalam garis lintang dan bujur) dari hasil yang ditemukan.
- **Reverse geocoding** — Reverse geocoding memungkinkan Anda menemukan tempat di dekat lokasi tertentu.
- **Pelengkapan Otomatis** — Pelengkapan Otomatis adalah proses membuat saran otomatis saat pengguna mengetik dalam kueri. Misalnya, jika mereka mengetik **Par** satu saran mungkin **Paris, France**.

Lokasi Amazon memungkinkan Anda memilih penyedia data untuk operasi penelusuran tempat dengan membuat dan mengonfigurasi sumber daya indeks tempat.

Setelah membuat sumber daya, Anda dapat mengirim permintaan menggunakan AWS SDK untuk bahasa pilihan Anda, Amplify, atau titik akhir REST API. Anda dapat menggunakan data dari respons untuk menandai lokasi di peta, memperkaya data posisi, dan mengubah posisi menjadi teks yang dapat dibaca manusia.

## Note

Untuk ikhtisar tentang mencari konsep tempat, lihat [Pencarian tempat](#).

## Topik

- [Prasyarat](#)
- [Geocoding](#)
- [Membalikkan geocoding](#)
- [Pelengkapan otomatis](#)
- [Menggunakan ID tempat](#)
- [Tempatkan kategori dan hasil penyaringan](#)

- [Amazon Aurora PostgreSQL fungsi yang ditentukan pengguna untuk Amazon Location Service](#)
- [Mengelola sumber daya indeks tempat Anda](#)

## Prasyarat

Sebelum Anda memulai geocoding, membalikkan geocoding atau mencari tempat, ikuti langkah-langkah prasyarat:

### Topik

- [Membuat sumber daya indeks tempat](#)
- [Mengautentikasi permintaan Anda](#)

## Membuat sumber daya indeks tempat

Mulailah dengan membuat sumber daya indeks tempat di AWS akun Anda.

Saat membuat sumber daya indeks tempat, Anda dapat memilih dari penyedia data yang tersedia untuk mendukung kueri geocoding, geocoding terbalik, dan penelusuran:

1. Esri — Untuk informasi lebih lanjut tentang cakupan Esri di wilayah minat Anda, lihat [cakupan geocoding Esri](#) dalam dokumentasi Esri.
2. HERE Technologies - Untuk informasi lebih lanjut tentang cakupan HERE di wilayah yang Anda minati, lihat [cakupan geocoding HERE](#) dalam dokumentasi HERE.
3. Grab — Grab menyediakan data hanya untuk Asia Tenggara. Untuk informasi selengkapnya tentang cakupan Grab, lihat [Negara/wilayah dan area yang dicakup](#) di panduan ini.


Anda dapat melakukannya menggunakan konsol Amazon Location Service, the AWS CLI, atau Amazon Location API.

### Console

Untuk membuat sumber daya indeks tempat menggunakan konsol Amazon Location Service

1. Buka konsol Amazon Location Service di <https://console.aws.amazon.com/location/>.
2. Di panel navigasi kiri, pilih Tempatkan indeks.
3. Pilih Buat indeks tempat.

4. Isi kotak-kotak berikut:
  - Nama — Masukkan nama untuk sumber daya indeks tempat. Misalnya, *ExamplePlaceIndex*. Maksimal 100 karakter. Entri yang valid termasuk karakter alfanumerik, tanda hubung, titik, dan garis bawah.
  - Deskripsi - Masukkan deskripsi opsional.
5. Di bawah Penyedia data, pilih [penyedia data](#) yang tersedia untuk digunakan dengan sumber daya indeks tempat Anda.

 Note

Jika aplikasi Anda melacak atau merutekan aset yang Anda gunakan dalam bisnis Anda, seperti kendaraan pengiriman atau karyawan, Anda tidak boleh menggunakan Esri sebagai penyedia geolokasi Anda. Lihat bagian 82 dari [ketentuanAWS layanan](#) untuk detail selengkapnya.

6. Di bawah Opsi penyimpanan data, tentukan apakah Anda ingin menyimpan hasil penelusuran dari sumber daya indeks tempat Anda.
7. (Opsional) Di bawah Tag, masukkan tag Kunci dan Nilai. Ini menambahkan tag sumber daya indeks tempat baru Anda. Untuk informasi selengkapnya, lihat [Menandai sumber daya Anda](#).
8. Pilih Buat indeks tempat.

## API

Untuk membuat sumber daya indeks tempat menggunakan Amazon Location API

Gunakan [CreatePlaceIndex](#) operasi dari Amazon Location Places API.

Contoh berikut adalah permintaan API untuk membuat sumber daya indeks tempat yang disebut *ExamplePlaceIndex* menggunakan penyedia data *Esri*.

```
POST /places/v0/indexes
Content-type: application/json

{
  "DataSource": "Esri",
  "DataSourceConfiguration": {
    "IntendedUse": "SingleUse"
```

```
    },  
    "Description": "string",  
    "IndexName": "ExamplePlaceIndex",  
    "Tags": {  
        "Tag1" : "Value1"  
    }  
}
```

## AWS CLI

Untuk membuat sumber daya indeks tempat menggunakan AWS CLI perintah

Gunakan perintah [create-place-index](#).

Contoh berikut menciptakan sumber daya indeks tempat yang disebut *ExamplePlaceIndex* menggunakan *Esri* sebagai penyedia data.

```
aws location \  
  create-place-index \  
  --data-source "Esri" \  
  --description "Example place index" \  
  --index-name "ExamplePlaceIndex" \  
  --tags Tag1=Value1
```

### Note

Penagihan tergantung pada penggunaan Anda. Anda mungkin dikenakan biaya untuk penggunaan AWS layanan lain. Untuk informasi selengkapnya, lihat [harga Amazon Location Service](#).

## Mengautentikasi permintaan Anda

Setelah Anda membuat sumber daya indeks tempat dan Anda siap untuk mulai membangun fitur lokasi ke dalam aplikasi Anda, pilih bagaimana Anda akan mengautentikasi permintaan Anda:

- Untuk mempelajari cara mengakses layanan, lihat [Mengakses Amazon Location Service](#).
- Jika Anda memiliki situs web dengan pengguna anonim, Anda mungkin ingin menggunakan Kunci API atau Amazon Cognito.

## Contoh

Contoh berikut menunjukkan penggunaan kunci API untuk otorisasi, menggunakan [AWS JavaScript SDK v3](#), dan Lokasi Amazon. [JavaScript Pembantu otentikasi](#)

```
import { LocationClient, SearchPlaceIndexForTextCommand } from "@aws-sdk/client-location";
import { withAPIKey } from "@aws/amazon-location-utilities-auth-helper";

const apiKey = "v1.public.your-api-key-value"; // API key

// Create an authentication helper instance using an API key
const authHelper = await withAPIKey(apiKey);

const client = new LocationClient({
  region: "<region>", // region containing Cognito pool
  ...authHelper.getLocationClientConfig(), // Provides configuration required to make requests to Amazon Location
});

const input = {
  IndexName: "ExamplePlaceIndex",
  Text: "Anyplace",
  BiasPosition: [-123.4567, 45.6789]
};

const command = new SearchPlaceIndexForTextCommand(input);

const response = await client.send(command);
```

## Geocoding

Geocoding adalah proses yang mengubah teks, seperti alamat, wilayah, nama bisnis, atau tempat menarik, menjadi satu set koordinat geografis. Anda dapat menggunakan sumber daya indeks tempat untuk mengirimkan permintaan geocoding dan menggabungkan data yang diambil dari geocoding untuk menampilkan data pada peta untuk web atau aplikasi seluler Anda.

Bagian ini memandu Anda melalui cara mengirim permintaan geocoding sederhana, dan cara mengirim permintaan geocoding dengan spesifikasi opsional.



## Geocoding

Anda dapat mengirimkan permintaan sederhana ke geocode menggunakan [SearchPlaceIndexForText](#) operasi untuk mengonversi alamat ke satu set koordinat. Permintaan sederhana berisi parameter yang diperlukan berikut:

- **Text**— Alamat, nama, kota, atau wilayah untuk dikonversi ke satu set koordinat. Misalnya, `stringAny Town`.

Untuk menentukan jumlah maksimum hasil per halaman, gunakan parameter opsional berikut:

- **MaxResults**— Membatasi jumlah maksimum hasil yang dikembalikan dalam respons kueri.

Anda dapat menggunakan AWS CLI atau Amazon Location API.

### API

Contoh berikut adalah [SearchPlaceIndexForText](#) permintaan untuk mencari sumber daya indeks tempat, *ExamplePlaceIndex*, untuk alamat, nama, kota atau wilayah yang disebut *Kota Mana-mana*.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json

{
  "Text": "Any Town",
  "MaxResults": 10
}
```

### AWS CLI

Contoh berikut adalah [search-place-index-for-text](#) perintah untuk mencari sumber daya indeks tempat, *ExamplePlaceIndex*, untuk alamat, nama, kota atau wilayah yang disebut *Kota Mana-mana*.

```
aws location \
  search-place-index-for-text \
    --index-name ExamplePlaceIndex \
    --text "Any Town" \
    --max-results 10
```

## Geocode dekat posisi

Saat geocoding, Anda dapat melakukan geocode di dekat posisi tertentu dengan parameter opsional berikut:

- **BiasPosition**— Posisi yang ingin Anda cari di dekatnya. Ini mempersempit pencarian Anda dengan mencari hasil yang paling dekat dengan posisi yang diberikan. Didefinisikan sebagai [longitude, latitude]

*Contoh berikut adalah [SearchPlaceIndexForText](#) permintaan untuk mencari sumber daya indeks tempat untuk alamat, nama, kota atau wilayah yang disebut Kota Mana-mana di dekat posisi [-123.4567, 45.6789].*

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json

{
  "Text": "Any Town",
  "BiasPosition": [-123.4567,45.6789]
}
```

## Geocode dalam kotak pembatas

Anda dapat melakukan geocode dalam kotak pembatas untuk mempersempit hasil Anda ke koordinat dalam batas tertentu menggunakan parameter opsional berikut:

- **FilterBBox**— Kotak pembatas yang Anda tentukan untuk memfilter hasil Anda ke koordinat dalam batas-batas kotak. Didefinisikan sebagai [LongitudeSW, LatitudeSW, LongitudeNE, LatitudeNE]

### Note

Permintaan tidak dapat berisi parameter `FilterBBox` dan `BiasPosition` parameter. Menentukan kedua parameter dalam permintaan mengembalikan `ValidationException` kesalahan.

Contoh berikut adalah [SearchPlaceIndexForText](#) permintaan untuk mencari di dalam kotak pembatas untuk alamat, nama, kota atau wilayah yang disebut Kota Apa *Pun*. Kotak pembatas mengikuti itu:

- *Garis bujur sudut barat daya adalah -124.1450.*
- *Garis lintang sudut barat daya adalah 41.7045.*
- *Garis bujur sudut timur laut adalah -124.1387.*
- *Garis lintang sudut timur laut adalah 41.7096.*

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json
```

```
{
  "Text": "Any Town",
  "FilterBBox": [
    -124.1450,41.7045,
    -124.1387,41.7096
  ]
}
```

## Geocode dalam suatu negara

Anda dapat melakukan geocode dalam satu atau beberapa negara yang Anda tentukan dengan menggunakan parameter opsional berikut:

- **FilterCountries**— Negara atau wilayah yang ingin Anda geocode di dalamnya. Anda dapat menentukan hingga 100 negara dalam satu permintaan menggunakan kode negara tiga huruf [ISO 3166](#). Misalnya, gunakan AUS untuk Australia.

Contoh berikut adalah [SearchPlaceIndexForText](#) permintaan untuk mencari alamat, nama, kota atau wilayah yang disebut *Kota Mana-mana* di Jerman dan Prancis.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json
```

```
{
  "Text": "Any Town",
  "FilterCountries": ["DEU","FRA"]
}
```

```
}
```

## Pemfilteran berdasarkan kategori

Anda dapat memfilter kategori yang dikembalikan dalam permintaan geocode Anda dengan menggunakan parameter opsional berikut:

- **FilterCategories**— Kategori hasil yang ingin Anda kembalikan dalam kueri Anda. Anda dapat menentukan hingga 5 kategori dalam satu permintaan. Anda dapat menemukan daftar kategori Amazon Location Service di bagian [Kategori](#). Misalnya, Anda dapat menentukan `Hotel` untuk menentukan hanya hotel yang kembali dalam kueri Anda.

Contoh berikut adalah [SearchPlaceIndexForText](#) permintaan untuk mencari kedai kopi yang disebut *Hometown Coffé* di Amerika Serikat.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json

{
  "Text": "Hometown Coffee",
  "FilterCategories": ["Coffee Shop"],
  "FilterCountries": ["USA"]
}
```

Untuk detail selengkapnya tentang pemfilteran pada kategori, lihat [Tempatkan kategori dan hasil penyaringan](#)

## Geocode dalam bahasa pilihan

Anda dapat mengatur preferensi bahasa untuk hasil pencarian Anda dengan menggunakan `Language` parameter opsional. Misalnya, pencarian **100 Main St, Anytown, USA** dapat kembali secara `100 Main St, Any Town, USA` default. Tetapi jika Anda memilih `fr` sebagai `Language`, maka hasilnya mungkin kembali `100 Rue Principale, Any Town, États-Unis` sebagai gantinya.

- **Language**— Kode bahasa yang digunakan untuk merender hasil kueri Anda. Nilai harus berupa kode bahasa [BCP 47](#) yang valid. Misalnya, `en` untuk bahasa Inggris.

**Note**

Jika tidak Language ditentukan, atau bahasa yang ditentukan tidak didukung untuk hasil, bahasa default mitra untuk hasil tersebut akan digunakan.

Contoh berikut adalah `SearchPlaceIndexforText` permintaan untuk mencari tempat yang dipanggil **Any Town** dengan bahasa pilihan yang ditentukan sebagaide.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json
{
  "Text": "Any Town",
  "Language": "de"
}
```

## Contoh tanggapan

### Example

Berikut ini adalah contoh respons saat Anda memanggil [SearchPlaceIndexForText](#) operasi dari Amazon Location Places API. Hasilnya termasuk [tempat](#) yang relevan dan [ringkasan](#) permintaan. Dua tanggapan ditampilkan, berdasarkan memilih Esri atau HERE sebagai mitra.

### Example request

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json

{
  "Text": "Amazon",
  "MaxResults": 1,
  "FilterCountries": ["USA"],
  "BiasPosition": [-112.10, 46.32]
}
```

### Example response (Esri)

```
{
  "Results": [
    {
```

```
    "Place": {
      "Country": "USA",
      "Geometry": {
        "Point": [
          -112.10667999999998,
          46.319090000000074
        ]
      },
      "Interpolated": false,
      "Label": "Amazon, MT, USA",
      "Municipality": "Amazon",
      "Region": "Montana",
      "SubRegion": "Jefferson County"
    },
    "Distance": 523.4619749879726,
    "Relevance": 1
  }
],
"Summary": {
  "BiasPosition": [
    -112.1,
    46.32
  ],
  "DataSource": "Esri",
  "FilterCountries": [
    "USA"
  ],
  "MaxResults": 1,
  "ResultBBox": [
    -112.10667999999998,
    46.319090000000074,
    -112.10667999999998,
    46.319090000000074
  ],
  "Text": "Amazon"
}
}
```

### Example response (HERE)

```
{
  "Summary": {
    "Text": "Amazon",
```

```

    "BiasPosition": [
      -112.1,
      46.32
    ],
    "FilterCountries": [
      "USA"
    ],
    "MaxResults": 1,
    "ResultBBox": [
      -112.10668,
      46.31909,
      -112.10668,
      46.31909
    ],
    "DataSource": "Here"
  },
  "Results": [
    {
      "Place": {
        "Label": "Amazon, Jefferson City, MT, United States",
        "Geometry": {
          "Point": [
            -112.10668,
            46.31909
          ]
        },
        "Neighborhood": "Amazon",
        "Municipality": "Jefferson City",
        "SubRegion": "Jefferson",
        "Region": "Montana",
        "Country": "USA",
        "Interpolated": false,
        "TimeZone": {
          "Name": "America/Denver",
          "Offset": -25200
        }
      },
      "PlaceId": "AQAAAIAADsn2T3KdrRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-
o3nqdlJJZAdgcT2oWi1w9pS4wXX0k301vsK1GsPyHjV4EJxsu289i3hV0_BUPgP7SFoWAI8BW2v7LvAjQ5NfUPy7a1v9a
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ",
      "Distance":
523.4619749905755
    }
  ]

```

```
}
```

## Membalikkan geocoding

Reverse geocoding adalah proses yang mengubah satu set koordinat menjadi teks yang bermakna, seperti alamat, wilayah, nama bisnis, atau tempat menarik. Anda dapat menggunakan sumber daya indeks tempat untuk mengirimkan permintaan geocoding terbalik dan menggabungkan data yang diambil dari geocoding terbalik untuk menampilkan data pada peta untuk web atau aplikasi seluler Anda.

Bagian ini memandu Anda melalui cara mengirim permintaan geocoding terbalik sederhana.

### Membalikkan geocoding

Anda dapat mengirimkan permintaan sederhana untuk membalikkan geocode satu set koordinat dan mengubahnya menjadi alamat yang bermakna, tempat menarik atau lokasi umum tanpa alamat menggunakan operasi. [SearchPlaceIndexForPosition](#) Permintaan sederhana berisi parameter yang diperlukan berikut:

- **Position**— Satu set koordinat yang ingin Anda konversi ke alamat, tempat menarik, atau lokasi umum. Didefinisikan menggunakan format [`longitude`, `latitude`].

Untuk menentukan jumlah maksimum hasil per halaman, tambahkan parameter opsional berikut:

- **MaxResults**— Membatasi jumlah maksimum hasil yang dikembalikan dalam respons kueri.

Jika Anda ingin menentukan bahasa pilihan untuk hasil kueri Anda, gunakan parameter opsional berikut:

- **Language**- Kode bahasa yang akan digunakan untuk merender hasil. Nilai harus berupa kode bahasa [BCP 47](#) yang valid. Misalnya, `en` untuk bahasa Inggris.

#### Note

Jika tidak `Language` ditentukan, atau bahasa yang ditentukan tidak didukung untuk hasil, bahasa default mitra untuk hasil tersebut akan digunakan.



Anda dapat menggunakan AWS CLI atau Amazon Location API.

## API

Contoh berikut adalah [SearchPlaceIndexForPosition](#) permintaan untuk mencari sumber daya indeks tempat,, untuk alamat yang bermakna `ExamplePlaceIndex`, tempat menarik atau lokasi umum di dekat posisi `[122.3394, 47.6159]`.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/position
Content-type: application/json

{
  "Position": [-122.3394,47.6159],
  "MaxResults": 5,
  "Language": "de"
}
```

## AWS CLI

Contoh berikut adalah [search-place-index-for-position](#) perintah untuk mencari sumber daya indeks tempat,, untuk alamat yang bermakna `ExamplePlaceIndex`, tempat menarik atau lokasi umum di dekat posisi `[122.3394, 47.6159]`.

```
aws location \
  search-place-index-for-position \
    --index-name ExamplePlaceIndex \
    --position -122.3394 47.6159 \
    --max-results 5 \
    --language de
```

## Contoh tanggapan

### Example

Berikut ini adalah contoh respons saat memanggil [SearchPlaceIndexForPosition](#) operasi dari Amazon Location Places API. Hasil mengembalikan [tempat](#) yang relevan dan [ringkasan](#) permintaan. Dua tanggapan ditampilkan, berdasarkan memilih Esri atau Di sini sebagai mitra.

## Example request

```
POST /places/v0/indexes/ExamplePlaceIndex/search/position
Content-type: application/json

{
  "Position": [-122.3394,47.6159],
  "MaxResults": 1
}
```

## Example response (Esri)

```
{
  "Results": [
    {
      "Place": {
        "AddressNumber": "2111",
        "Country": "USA",
        "Geometry": {
          "Point": [
            -122.33937999999995,
            47.615910000000004
          ]
        },
        "Interpolated": false,
        "Label": "The Spheres, 2111 7th Ave, Seattle, WA, 98121, USA",
        "Municipality": "Seattle",
        "Neighborhood": "Belltown",
        "PostalCode": "98121",
        "Region": "Washington",
        "SubRegion": "King County"
      },
      "Distance": 1.8685861313438727
    }
  ],
  "Summary": {
    "DataSource": "Esri",
    "MaxResults": 1,
    "Position": [
      -122.3394,
      47.6159
    ]
  }
}
```

```
}

```

### Example response (HERE)

```
{
  "Summary": {
    "Position": [
      -122.3394,
      47.6159
    ],
    "MaxResults": 1,
    "DataSource": "Here"
  },
  "Results": [
    {
      "Place": {
        "Label": "2111 7th Ave, Seattle, WA 98121-5114, United States",
        "Geometry": {
          "Point": [
            -122.33938,
            47.61591
          ]
        },
        "AddressNumber": "2111",
        "Street": "7th Ave",
        "Neighborhood": "Belltown",
        "Municipality": "Seattle",
        "SubRegion": "King",
        "Region": "Washington",
        "Country": "USA",
        "PostalCode": "98121-5114",
        "Interpolated": false,
        "TimeZone": {
          "Name": "America/Los_Angeles",
          "Offset": -28800
        }
      },
      "PlaceId": "AQAAAIADsn2T3KdirWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-
o3nqd1JZAdgcT2oWi1w9pS4wXX0k301vsK1GsPyHjV4EJxsu289i3hVO_BUPgP7SFoWAI8BW2v7LvAjQ5NfUPy7a1v9a
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ",
      "Distance": 1.868586125090601
    }
  ]
}
```

```
}
```

## Pelengkapan otomatis

Autocomplete memberikan umpan balik responsif kepada pengguna akhir saat mereka mengetik permintaan pencarian mereka. Ini memberikan saran untuk alamat dan tempat menarik berdasarkan teks bentuk bebas sebagian atau salah eja. Anda dapat menggunakan sumber daya indeks tempat untuk meminta saran pelengkapan otomatis, dan menampilkan saran yang dihasilkan dalam aplikasi Anda.

Lokasi Amazon tidak mendukung penyimpanan saran pelengkapan otomatis. Kesalahan dikembalikan jika indeks tempat yang digunakan untuk panggilan pelengkapan otomatis dikonfigurasi untuk digunakan dengan geocode yang disimpan. Untuk menggunakan geocode tersimpan dan kueri untuk saran, buat dan konfigurasi beberapa indeks tempat.

Bagian ini menjelaskan cara mengirim permintaan pelengkapan otomatis. Ini dimulai dengan bentuk permintaan yang paling dasar, dan kemudian menunjukkan parameter opsional yang dapat Anda gunakan untuk meningkatkan relevansi hasil pencarian pelengkapan otomatis.

### Menggunakan pelengkapan otomatis

Anda dapat mengirimkan permintaan sederhana untuk saran pelengkapan otomatis dengan menggunakan [SearchPlaceIndexForSuggestions](#) operasi. Bentuk permintaan yang paling sederhana memiliki satu parameter yang diperlukan, `kueriText`:

- `Text`— Teks parsial bentuk bebas yang digunakan untuk menghasilkan saran tempat. Misalnya, `stringeiffel tow`.

Untuk membatasi jumlah hasil yang dikembalikan, tambahkan `MaxResults` parameter opsional:

- `MaxResults`— Membatasi jumlah hasil yang dikembalikan dalam respons kueri.

Anda dapat menggunakan API Lokasi Amazon atau AWS CLI.

## API

Contoh berikut adalah [SearchPlaceIndexForSuggestions](#) permintaan untuk mencari sumber daya indeks tempat *ExamplePlaceIndex*, hingga 5 saran berdasarkan nama tempat sebagian *kamp*.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/suggestions
Content-type: application/json

{
  "Text": "kamp",
  "MaxResults": 5
}
```

## AWS CLI

Contoh berikut adalah [search-place-index-for-suggestions](#) perintah untuk mencari sumber daya indeks tempat *ExamplePlaceIndex*, hingga 5 saran berdasarkan nama tempat sebagian *kamp*.

```
aws location \
  search-place-index-for-suggestions \
  --index-name ExamplePlaceIndex \
  --text kamp \
  --max-results 5
```

Panggilan untuk `SearchPlaceIndexForSuggestions` menghasilkan daftar tempat dengan nama dan ID untuk masing-masing. Anda dapat menggunakan hasil tersebut untuk menyajikan saran tentang apa yang mungkin dicari pengguna, saat mereka mengetik, seperti memberikan daftar dropdown pilihan di bawah kotak teks. Misalnya, berikut adalah hasil untuk saran, berdasarkan pengguna mengetik *kamp*.

```
{
  "Summary": {
    "Text": "kamp",
    "MaxResults": 5,
    "DataSource": "Esri"
  },
  "Results": [
    {
```

```

        "Text": "Kampuchea",
        "PlaceId": "AQAAAAIAADsn2T3KdrWearXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-
o3nqd1JZAdgcT2oWi1w9pS4wXX0k301vsK1GsPyHjV4EJxsu289i3hV0_BUPgP7SFoWai8BW2v7LvAjQ5NfUPy7a1v9ajT3
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ"
    },
    {
        "Text": "Kampoul, Kabul, AFG",
        "PlaceId":
"AQAAAAIAAA1mx1_-9ffzXD07rBgo9fh6E01Pd1YKvuT5rz2qBDxqBkhTlgkei0PR2s5sa3YBLxUqQI8bhymyYcu9R-
DkX3L9QSi3CB5LhNPu160iSFJo6H8S1CrX03QsJALhrr9mdbg0R4R4YDywkHkeBlbn7g5C5LI_wYx873WeQZuilwtsGm8j
UeXcb_bg"
    },
    {
        "Text": "Kampala, UGA",
        "PlaceId":
"AQAAAAIAAzZfZt3qMrUKG0byhP6MM0pqy2L8SULLVWT7a3ertLBRS6Q5n7I4s9D7E0nRHADaj7mL7kvX1Q8HD-
mpuiATXNJ1Ix4_V_1B15zHe8j1YKMWvXbgb08cMpgR2fqYqZMR1x-
dfB0080oqujKZldvPIDK1kNe3GwcaqvMWWPMeaGd203brFynubAe-MmFF-Gjz-WBMfUy9og6MV7bkk6NGCA"
    },
    {
        "Text": "Kampar, Riau, IDN",
        "PlaceId": "AQAAAAIAAvbXXx-
sr0i111tH0kPdao0GF7WQ_KaZ444SEnevycp6Gtf_2JWgPfCE5bIQCYwya1uZQpX2a8YJoFm2K7Co14fLu7IK0yYOLhZx4k
    },
    {
        "Text": "Kampung Pasir Gudang Baru, Johor, MYS",
        "PlaceId":
"AQAAAAIAA4HLQHdjUDcaaXLE9wtNIT1cjQYLgkbnMoG2eNN0AaQ8PJoWabLRXmmPUaAj8MAD6vT0i6zqaun5Mixyj7vnYX
    }
}
}

```

Bagian selanjutnya menjelaskan cara menggunakan PlaceID dari hasil ini.

## Menggunakan hasil pelengkapan otomatis

Panggilan untuk `SearchPlaceIndexForSuggestions` menghasilkan daftar tempat dengan nama dan ID untuk masing-masing. Anda dapat menggunakan hasil tersebut untuk menyajikan saran tentang apa yang mungkin dicari pengguna, saat mereka mengetik, seperti memberikan daftar dropdown pilihan di bawah kotak teks. Ketika pengguna memilih salah satu hasil, Anda kemudian dapat memanggil [GetPlace](#) operasi dengan ID pilihan mereka untuk mengembalikan detail tempat itu, termasuk lokasi, alamat, atau detail lainnya.

**Note**

A PlaceId hanya berlaku jika semua hal berikut sama dalam permintaan pencarian asli, dan panggilan keGetPlace.

- Pelanggan Akun AWS
- Wilayah AWS
- Penyedia data ditentukan dalam sumber daya indeks tempat

Biasanya, Anda menggunakan GetPlace dengan Amazon Location API. Contoh berikut adalah [GetPlace](#) permintaan untuk menemukan salah satu saran dari bagian sebelumnya. Contoh ini didasarkan pada nama tempat sebagian *kamp*.

```
POST /places/v0/indexes/ExamplePlaceIndex/  
places/AQAAAIAADsn2T3KdrRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-  
o3nqdlJZAdgcT2oWi1w9pS4wXX0k301vsKlGsPyHjV4EJxsu289i3hV0_BUPgP7SFoWAI8BW2v7LvAjQ5NfUPy7a1v9ajT3-  
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ
```

## Pelengkapan otomatis di dekat posisi

Saat Anda mencari saran tempat pelengkapan otomatis dengan menggunakan [SearchPlaceIndexForSuggestions](#), Anda bisa mendapatkan lebih banyak saran yang relevan secara lokal dengan menambahkan parameter opsional berikut:

- BiasPosition— Posisi yang ingin Anda cari di dekatnya. Didefinisikan sebagai [longitude, latitude].

*Contoh berikut menggunakan [SearchPlaceIndexForSuggestions](#) permintaan untuk mencari sumber daya ExamplePlaceIndex indeks tempat untuk saran tempat yang cocok dengan kueri sebagian kamp dekat posisi [32.5827, 0.3169].*

```
POST /places/v0/indexes/ExamplePlaceIndex/search/suggestions  
Content-type: application/json
```

```
{  
  "Text": "kamp",  
  "BiasPosition": [32.5827,0.3169]
```

```
}
```

*Saran yang dikembalikan untuk hal yang sama Text bisa berbeda jika yang berbeda BiasPosition dipilih, seperti [-96.7977, 32.7776].*

## Pelengkapan otomatis dalam kotak pembatas

Anda dapat mempersempit pencarian pelengkapan otomatis untuk hanya menerima saran untuk tempat-tempat yang berada dalam batas tertentu dengan menambahkan parameter opsional berikut:

- `FilterBoundingBox`— Kotak pembatas yang Anda tentukan untuk memfilter hasil Anda ke koordinat dalam batas-batas kotak. Didefinisikan sebagai [`LongitudeSW`, `LatitudeSW`, `LongitudeNE`, `LatitudeNE`]

### Note

Permintaan tidak dapat berisi parameter `FilterBoundingBox` dan `BiasPosition` parameter. Menentukan kedua parameter dalam permintaan mengembalikan `ValidationException` kesalahan.

Contoh berikut menggunakan [SearchPlaceIndexForSuggestions](#) permintaan untuk mencari sumber daya indeks tempat `ExamplePlaceIndex` untuk saran tempat yang cocok dengan kueri sebagian `kamp`, dan yang terkandung dalam kotak pembatas di mana:

- *Garis bujur sudut barat daya kotak pembatas adalah 32.5020.*
- *Garis lintang sudut barat daya kotak pembatas adalah 0,2678.*
- *Garis bujur sudut timur laut kotak pembatas adalah 32.6129.*
- *Garis lintang sudut timur laut kotak pembatas adalah 0,3502.*

```
POST /places/v0/indexes/ExamplePlaceIndex/search/suggestions
Content-type: application/json
```

```
{
  "Text": "kamp",
  "FilterBoundingBox": [
    32.5020, 0.2678,
    32.6129, 0.3502
```



```
    ]
  }
```

*Saran yang dikembalikan untuk hal yang sama berbeda jika yang berbeda Text dipilih, seperti [-97.9651, 32.0640, -95.1196, 34.0436]. FilterBBBox*

## Pelengkapan otomatis dalam suatu negara

Anda dapat mempersempit pencarian pelengkapan otomatis untuk hanya menerima saran untuk tempat-tempat yang berada di dalam negara tertentu, atau kumpulan negara, dengan menambahkan parameter opsional berikut:

- **FilterCountries**— Negara tempat Anda ingin mencari saran tempat di dalamnya. Anda dapat menentukan hingga 100 negara dalam satu permintaan menggunakan kode negara tiga huruf [ISO 3166](#). Misalnya, gunakan AUS untuk Australia.

Contoh berikut menggunakan [SearchPlaceIndexForSuggestions](#) permintaan untuk mencari sumber daya indeks tempat *ExamplePlaceIndex* untuk saran tempat yang cocok dengan kueri sebagian *kamp* dan yang terkandung dalam Uganda, Kenya, atau Tanzania:

```
POST /places/v0/indexes/ExamplePlaceIndex/search/suggestions
Content-type: application/json

{
  "Text": "kamp",
  "FilterCountries": ["UGA", "KEN", "TZA"]
}
```

Saran yang dikembalikan untuk hal Text yang sama berbeda jika FilterCountries daftar yang berbeda dipilih, seperti ["USA"].

## Contoh tanggapan

*Berikut ini adalah contoh respons pelengkapan otomatis yang disarankan untuk [SearchPlaceIndexForSuggestions](#) operasi, menggunakan kamp teks.*

```
{
  "Summary": {
    "Text": "kamp",
```

```

    "MaxResults": 5,
    "DataSource": "Esri"
  },
  "Results": [
    {
      "Text": "Kampuchea",
      "PlaceId": "AQAAAIADsn2T3KdRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-
o3nqdlJZAdgcT2oWi1w9pS4wXX0k301vsK1GsPyHjV4EJxsu289i3hV0_BUPgP7SFoWAI8BW2v7LvAjQ5NfUPy7a1v9ajT3
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ"
    },
    {
      "Text": "Kampoul, Kabul, AFG",
      "PlaceId":
"AQAAAIAAA1mx1_-9ffzXD07rBgo9fh6E01Pd1YKvuT5rz2qBDxqBkhTlgkei0PR2s5sa3YBLxUqQI8bhymyYcu9R-
DkX3L9QSi3CB5LhNPu160iSFJo6H8S1CrX03QsJALhrr9mdbg0R4R4YDywkHkeBlnbn7g5C5LI_wYx873WeQZuilwtsGm8j
UeXcb_bg"
    },
    {
      "Text": "Kampala, UGA",
      "PlaceId":
"AQAAAIAAzZfZt3qMrUkG0byhP6MM0pqy2L8SUL1VWT7a3ertLBRS6Q5n7I4s9D7E0nRHADaj7mL7kvX1Q8HD-
mpuiATXNJ1Ix4_V_1B15zHe8jLYKMWvXbgb08cMpgR2fqYqZMR1x-
dfB0080oqujKZ1dvPIDK1kNe3GwcaqvMWWPMeaGd203brFynubAe-MmFF-Gjz-WBMfUy9og6MV7bkk6NGCA"
    },
    {
      "Text": "Kampar, Riau, IDN",
      "PlaceId": "AQAAAIAAvbXXx-
sr0i111tH0kPdao0GF7WQ_KaZ444SEnevycp6Gtf_2JWgPfCE5bIQCYwya1uZQpX2a8YJoFm2K7Col4fLu7IK0yY0LhZx4k
",
    },
    {
      "Text": "Kampung Pasir Gudang Baru, Johor, MYS",
      "PlaceId":
"AQAAAIAA4HLQHdjUDcaaXLE9wtNIT1cjQYLgkbnMoG2eNN0AaQ8PJowabLRXmmPUaAj8MAD6vT0i6zqaun5Mixyj7vnYX
"
    }
  ]
}

```

## Menggunakan ID tempat

Mencari tempat mengembalikan daftar hasil. Sebagian besar hasil termasuk PlaceId untuk hasil itu. Anda dapat menggunakan a PlaceId dalam [GetPlace](#) operasi untuk mengembalikan informasi tentang tempat itu (termasuk nama, alamat, lokasi, atau detail lainnya).

**Note**

Menggunakan [SearchPlaceIndexForSuggestions](#) akan mengembalikan PlaceId hasil untuk indeks tempat mana pun yang dibuat dengan sumber data apa pun. Menggunakan [SearchPlaceIndexForText](#) atau [SearchPlaceIndexForPosition](#) akan mengembalikan PlaceId hanya jika sumber data yang digunakan adalah DI SINI.

Masing-masing PlaceId secara unik mendefinisikan tempat yang dirujuknya, tetapi satu tempat dapat memiliki lebih dari satu dari waktu PlaceId ke waktu, dan berdasarkan konteksnya. Aturan berikut menggambarkan keunikan dan umur panjang a. PlaceId

- Panggilan yang PlaceId dikembalikan yang Anda lakukan khusus untuk Anda Akun AWS, ke AWS Wilayah, dan ke penyedia data di PlaceIndex sumber daya Anda. GetPlace akan menemukan hasil hanya jika ketiga atribut ini cocok dengan panggilan asli yang membuat PlaceId.
- PlaceId Untuk suatu tempat akan berubah ketika data tentang tempat itu berubah. Misalnya, ketika bisnis yang dirujuk pindah lokasi atau mengubah nama.
- PlaceId Pengembalian dari panggilan pencarian berulang dapat berubah ketika layanan backend membuat pembaruan. Yang lebih lama PlaceId akan terus ditemukan, tetapi panggilan baru untuk mencari dapat mengembalikan ID yang berbeda.

PlaceId itu adalah string. Tidak ada batasan khusus untuk panjang aPlaceId. Berikut ini adalah contoh yang validPlaceId.

```
AQAAAAIAADsn2T3KdRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-
o3nqd1JZAdgcT2oWi1w9pS4wXX0k301vsK1GsPyHjV4EJxsu289i3hV0_BUPgP7SFoWai8BW2v7LvAjQ5NfUPy7a1v9ajT3
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ
```

Memanggil GetPlace dengan tempat yang datanya telah berubah (misalnya, lokasi bisnis yang telah keluar dari bisnis), akan menghasilkan ResourceNotFound kesalahan. PlaceId 404 Memanggil GetPlace dengan PlaceId yang tidak valid, atau salah satu di luar konteks, seperti dari yang lain Akun AWS, akan mengembalikan ValidationException kesalahan400,.

Meskipun Anda dapat menggunakan placeID dalam permintaan berikutnya, placeID tidak dimaksudkan sebagai pengenal permanen dan ID dapat berubah di antara panggilan API berturut-turut. Silakan lihat perilaku placeID berikut untuk setiap penyedia data:

- Esri: ID Place akan berubah setiap kuartal minimal. Periode waktu khas untuk perubahan ini adalah Maret, Juni, September, dan Desember. ID tempat juga dapat berubah antara perubahan triwulanan tipikal tetapi itu akan jauh lebih jarang.
- DI SINI: Kami menyarankan Anda menyimpan data tidak lebih dari seminggu untuk menjaga data Anda tetap segar. Anda dapat berasumsi bahwa kurang dari 1% pergeseran ID akan dirilis selama rilis yang kira-kira 1 - 2 kali per minggu.
- Grab: ID Place dapat kedaluwarsa atau menjadi tidak valid dalam situasi berikut.
  - Operasi data: POI dapat dihapus dari database POI Grab oleh Grab Map Ops berdasarkan kebenaran dasar, seperti ditutup di dunia nyata, terdeteksi sebagai POI duplikat, atau memiliki informasi yang salah. Grab akan menyinkronkan data ke lingkungan Waypoint setiap minggu.
  - POI interpolasi: POI interpolasi adalah POI sementara yang dihasilkan secara real time saat melayani permintaan, dan akan ditandai sebagai turunan di lapangan dalam respons. `place.result_type` Informasi POI yang diinterpolasi akan disimpan setidaknya selama 30 hari, yang berarti bahwa dalam 30 hari, Anda dapat memperoleh detail POI berdasarkan Place ID dari Place Details API. Setelah 30 hari, POI yang diinterpolasi (baik ID Tempat maupun detail) dapat kedaluwarsa dan tidak dapat diakses dari API Detail Tempat.

## Tempatkan kategori dan hasil penyaringan

Tempat dikategorikan. Jika Anda mencari bisnis, bisnis mungkin `Restaurant`, misalnya. Bahkan hasil pencarian alamat dapat dikategorikan berdasarkan apakah itu cocok dengan alamat, jalan, atau persimpangan.

Secara umum, Amazon Location Service mengkategorikan tempat ke dalam jenis Tempat. Tempat menarik selanjutnya dikategorikan ke dalam jenis Point of interest.

### Note

Tidak semua hasil akan memiliki kategori.

Anda dapat menggunakan kategori untuk memfilter pencarian geocoding Anda.

## Hasil penyaringan

Saat Anda menggunakan `SearchPlaceIndexForText`, Anda dapat memfilter hasil yang dikembalikan oleh kategori yang ingin Anda gunakan. Sebagai contoh:

- Jika Anda ingin mencari tempat yang disebut “Kopi Kampung Halaman”, dan hanya mengembalikan hasil yang dikategorikan sebagai kedai kopi, Anda dapat melakukannya dengan menelepon `SearchPlaceIndexForText` dan memasukkan kategori `Point of interest, Coffee Shop` dalam parameter. `FilterCategories`
- Saat mencari “123 Main St, Anytown, WA, 98123, USA”, Anda dapat memfilter hasil hanya ke alamat, sehingga Anda tidak mendapatkan kecocokan, misalnya, kode pos. Filter hanya ke alamat dengan menyertakan tipe `Place, AddressType` dalam `FilterCategories` parameter.

#### Note

Tidak semua penyedia data mendukung penyaringan, atau mendukungnya dengan cara yang sama. Untuk informasi selengkapnya, lihat [Pembatasan penyaringan oleh penyedia data](#).

Bagian selanjutnya mencantumkan kategori yang dapat Anda filter.

## Kategori

Daftar berikut menunjukkan kategori yang digunakan Amazon Location Service untuk mengkategorikan dan memfilter. Kategori ini digunakan dalam semua bahasa, terlepas dari parameter bahasa diatur ke bahasa yang berbeda.

#### Note

Amazon Location Service memetakan kategori penyedia data ke kumpulan kategori ini. Jika penyedia data menempatkan tempat ke dalam kategori yang bukan bagian dari daftar kategori Amazon Location Service, kategori penyedia akan dimasukkan dalam hasil sebagai kategori tambahan.

Jenis tempat — Jenis ini digunakan untuk menunjukkan jenis kecocokan yang digunakan untuk menemukan hasilnya.

- `AddressType`— Dikembalikan ketika hasilnya dicocokkan dengan alamat.
- `StreetType`— Kembali ketika hasilnya dicocokkan dengan jalan.
- `IntersectionType`— Kembali ketika hasilnya dicocokkan dengan persimpangan dua jalan.

- `PointOfInterestType`— Dikembalikan ketika hasilnya cocok dengan tempat menarik, seperti bisnis, atau lokasi sipil.
- `CountryType`— Dikembalikan ketika hasilnya dicocokkan dengan negara atau wilayah utama.
- `RegionType`— Dikembalikan ketika hasilnya dicocokkan dengan wilayah dalam suatu negara, seperti negara bagian atau provinsi.
- `SubRegionType`— Dikembalikan ketika hasilnya dicocokkan dengan subwilayah dalam suatu negara, seperti daerah atau wilayah metropolitan.
- `MunicipalityType`— Kembali ketika hasilnya dicocokkan dengan kota atau kota.
- `NeighborhoodType`— Dikembalikan ketika hasilnya dicocokkan dengan lingkungan atau area di dalam kota.
- `PostalCodeType`— Dikembalikan ketika hasilnya dicocokkan dengan kode pos.

Kategori tempat menarik — Kategori ini digunakan untuk menunjukkan jenis bisnis atau lokasi untuk hasil tempat menarik.

- `Airport`
- `Amusement Park`
- `Aquarium`
- `Art Gallery`
- `ATM`
- `Bakery`
- `Bank`
- `Bar`
- `Beauty Salon`
- `Bus Station`
- `Car Dealer`
- `Car Rental`
- `Car Repair`
- `Car Wash`
- `Cemetery`
- `Cinema`
- `City Hall`

- Clothing Store
- Coffee Shop
- Consumer Electronics Store
- Convenience Store
- Court House
- Dentist
- Embassy
- Fire Station
- Fitness Center
- Gas Station
- Government Office
- Grocery
- Higher Education
- Hospital
- Hotel
- Laundry
- Library
- Liquor Store
- Lodging
- Market
- Medical Clinic
- Motel
- Museum
- Nightlife
- Nursing Home
- Park
- Parking
- Pet Store
- Pharmacy
- Plumbing

- Police Station
- Post Office
- Religious Place
- Restaurant
- School
- Shopping Mall
- Sports Center
- Storage
- Taxi Stand
- Tourist Attraction
- Train Station
- Veterinary Care
- Zoo

## Pembatasan penyaringan oleh penyedia data

Tidak semua penyedia memiliki fungsi penyaringan yang sama. Tabel berikut menjelaskan perbedaannya.

Penyedia	API dengan dukungan filter	Kategori yang didukung untuk pemfilteran	Nilai yang dikembalikan
Esri	<code>SearchPlaceIndexForText</code> , <code>SearchPlaceIndexForSuggestions</code>	Filter berdasarkan jenis Tempat dan kategori Tempat Menarik.	Kategori dikembalikan oleh <code>SearchPlaceIndexForText</code> , <code>SearchPlaceIndexForPosition</code> , dan <code>GetPlace</code>
Di sini	<code>SearchPlaceIndexForText</code> , <code>SearchPlace</code>	Filter berdasarkan jenis Place saja.	Kategori dikembalikan oleh <code>SearchPlaceIndexForText</code>



Penyedia	API dengan dukungan filter	Kategori yang didukung untuk pemfilteran	Nilai yang dikembalikan
	ceIndexForSuggestions		danSearchPlaceIndexForSuggestions ,SearchPlaceIndexForPosition , dan GetPlace
Ambil	Tidak didukung	Tidak didukung	Tidak didukung
Buka Data	n/a (mencari tempat tidak didukung)	tidak berlaku	T/A

## Amazon Aurora PostgreSQL fungsi yang ditentukan pengguna untuk Amazon Location Service

Anda dapat menggunakan Amazon Location Service untuk bekerja dengan koordinat dan alamat yang disimpan dalam tabel database untuk membersihkan dan memperkaya data geospasial Anda.

Sebagai contoh:

- Anda dapat menggunakan geocoding untuk mengonversi alamat menjadi koordinat untuk menormalkan dan mengisi celah dalam data untuk alamat yang disimpan dalam tabel database.
- Anda dapat geocode alamat untuk mendapatkan posisi mereka dan menggunakan koordinat dengan fungsi spasial database, seperti fungsi yang menunjukkan baris di area tertentu.
- Anda dapat menggunakan data yang diperkaya untuk menghasilkan pelaporan otomatis, seperti membuat laporan otomatis yang menggambarkan semua perangkat di area tertentu, atau laporan otomatis untuk pembelajaran mesin yang menggambarkan area dengan tingkat kegagalan yang lebih tinggi saat mengirim pembaruan lokasi.

Tutorial ini menunjukkan cara memformat dan memperkaya alamat yang disimpan dalam tabel Amazon Aurora PostgreSQL database menggunakan Amazon Location Service.

- Amazon Aurora PostgreSQL— Mesin database relasional yang dikelola sepenuhnya, kompatibel dengan MySQL dan PostgreSQL, yang menghasilkan hingga lima kali throughput MySQL dan hingga tiga kali throughput PostgreSQL tanpa mengubah sebagian besar aplikasi Anda yang ada. Untuk informasi lebih lanjut, lihat [Apa itu Amazon Aurora?](#) di Panduan Pengguna Amazon Aurora.

**⚠ Important**

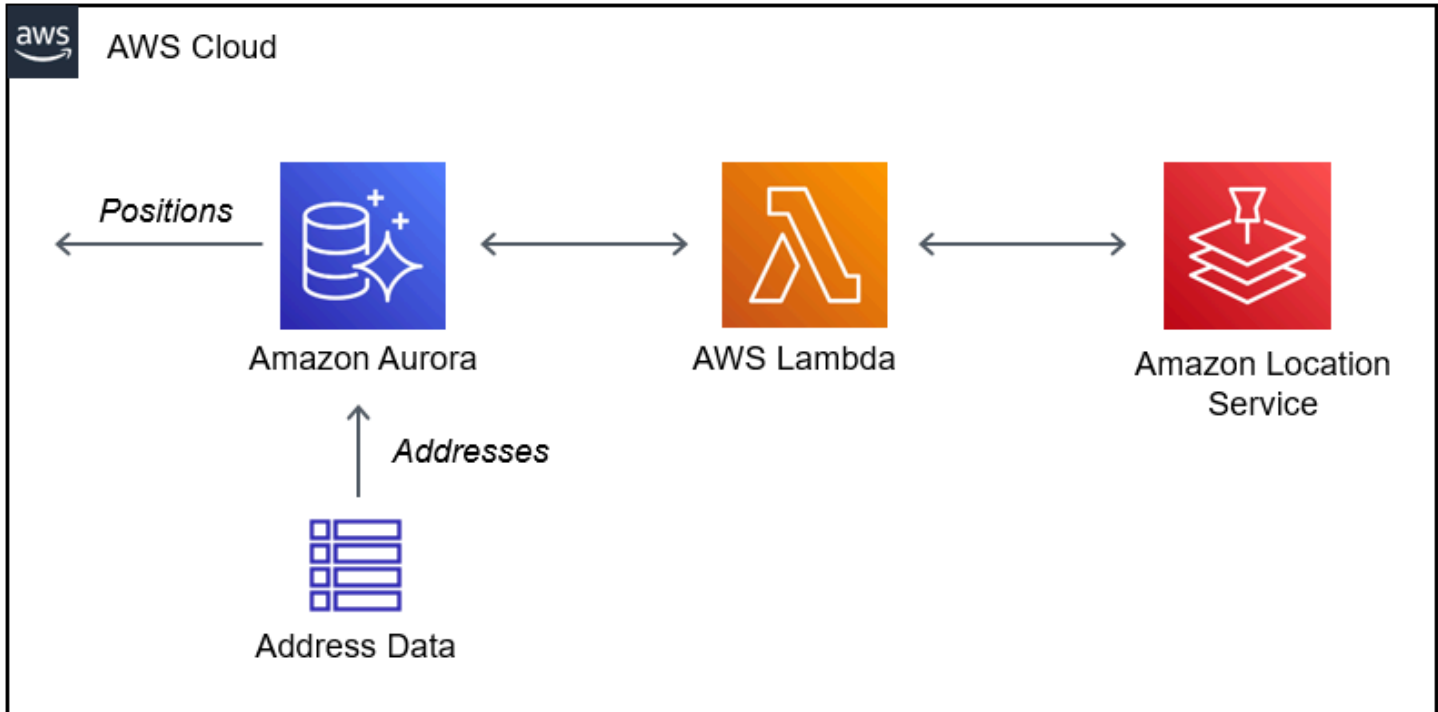
Aplikasi yang dihasilkan dalam tutorial ini menggunakan indeks tempat yang menyimpan hasil geocoding. Untuk informasi tentang biaya yang berlaku untuk menyimpan hasil geocoding, lihat [harga Amazon Location Service](#).

Kode sampel tersedia di repositori sampel Amazon Location Service [GitHub](#), [yang AWS CloudFormation](#) menyertakan templat.

#### Topik

- [Ikhtisar](#)
- [Prasyarat](#)
- [Quick start](#)
- [Buat sumber daya indeks tempat](#)
- [Buat AWS Lambda fungsi untuk geocoding](#)
- [Berikan Amazon Aurora PostgreSQL akses ke AWS Lambda](#)
- [Memanggil fungsi AWS Lambda](#)
- [Memperkaya database yang berisi data alamat](#)
- [Langkah selanjutnya](#)

## Ikhtisar



Arsitektur melibatkan integrasi berikut:

- Solusi ini menggunakan sumber daya indeks tempat Lokasi Amazon untuk mendukung kueri geocoding menggunakan operasi. `SearchPlaceIndexForText`
- AWS Lambda menggunakan Lambda Python yang ditangani geocode saat kebijakan IAM memberikan izin untuk mengizinkan AWS Lambda memanggil operasi geocoding Lokasi Amazon, `SearchPlaceIndexForText`
- Berikan izin Amazon Aurora PostgreSQL untuk menjalankan fungsi Lambda geocoding menggunakan fungsi yang ditentukan pengguna SQL.

## Prasyarat

Sebelum Anda mulai, Anda memerlukan prasyarat berikut:

- Sebuah Amazon Aurora PostgreSQL cluster. Untuk informasi selengkapnya tentang [Membuat klaster DB Amazon Aurora](#), lihat Panduan Pengguna Amazon Aurora.

**Note**

Jika klaster Amazon Aurora Anda tidak tersedia untuk umum, Anda juga harus mengonfigurasi Amazon Aurora untuk terhubung di cloud pribadi virtual (VPC) AWS Lambda di akun Anda. AWS Untuk informasi selengkapnya, lihat [Berikan Amazon Aurora PostgreSQL akses ke AWS Lambda](#).

- Alat pengembang SQL untuk terhubung ke Amazon Aurora PostgreSQL cluster.

## Quick start

Sebagai alternatif untuk melalui langkah-langkah dalam tutorial ini, Anda dapat meluncurkan tumpukan cepat untuk menyebarkan AWS Lambda fungsi yang mendukung operasi [SearchPlaceIndexForText](#) Lokasi Amazon. Ini secara otomatis mengonfigurasi AWS akun Anda untuk memungkinkan Amazon Aurora menelepon. AWS Lambda

Setelah Anda mengkonfigurasi AWS akun Anda, Anda perlu:

- Tambahkan fitur Lambda ke Amazon Aurora. Lihat Menambahkan peran IAM ke cluster Amazon Aurora DB di [Berikan Amazon Aurora PostgreSQL akses ke AWS Lambda](#)
- Muat fungsi yang ditentukan pengguna ke dalam database Anda. Lihat [Memanggil fungsi AWS Lambda](#).

A yellow button with a black play icon and the text "Launch Stack".

## Buat sumber daya indeks tempat

Mulailah dengan membuat sumber daya indeks tempat untuk mendukung kueri geocoding.

1. Buka konsol Amazon Location Service di <https://console.aws.amazon.com/location/>.
2. Di panel navigasi kiri, pilih Tempatkan indeks.
3. Isi kotak-kotak berikut:
  - Nama — Masukkan nama untuk sumber daya indeks tempat. Misalnya, *AuroraPlaceIndex*. Maksimal 100 karakter. Entri yang valid termasuk karakter alfanumerik, tanda hubung, titik, dan garis bawah.

- Deskripsi - Masukkan deskripsi opsional. Misalnya, *Tempatkan indeks untuk Amazon Aurora*.
4. Di bawah Penyedia data, pilih [penyedia data](#) yang tersedia untuk digunakan dengan sumber daya indeks tempat Anda. Jika Anda tidak memiliki preferensi, kami sarankan memulai dengan *Esri*.
  5. Di bawah Opsi penyimpanan data, tentukan Ya, hasil akan disimpan. Ini menunjukkan bahwa Anda berniat untuk menyimpan hasil geocoding dalam database.
  6. (Opsional) Di bawah Tag, masukkan tag Kunci dan Nilai. Ini menambahkan tag sumber daya indeks tempat baru Anda. Untuk informasi selengkapnya, lihat [Menandai sumber daya Anda](#).
  7. Pilih Buat indeks tempat.

## Buat AWS Lambda fungsi untuk geocoding

Untuk membuat koneksi antara Amazon Aurora PostgreSQL dan Amazon Location Service, Anda memerlukan AWS Lambda fungsi untuk menangani permintaan dari mesin database. Fungsi ini menerjemahkan peristiwa fungsi yang ditentukan pengguna Lambda dan memanggil operasi Lokasi Amazon. `SearchPlaceIndexForText`

Anda dapat membuat fungsi menggunakan AWS Lambda konsol, theAWS Command Line Interface, atau AWS Lambda API.

Untuk membuat fungsi yang ditentukan pengguna Lambda menggunakan konsol

1. Buka konsol AWS Lambda di <https://console.aws.amazon.com/lambda/>.
2. Dari navigasi kiri, pilih Fungsi.
3. Pilih Create Function, dan pastikan bahwa Author dari awal dipilih.
4. Isi kotak-kotak berikut:
  - Nama fungsi - Masukkan nama unik untuk fungsi Anda. Entri yang valid termasuk karakter alfanumerik, tanda hubung, dan garis bawah tanpa spasi. Misalnya, *AuroraGeocoder*.
  - Runtime - Pilih *Python 3.8*.
5. Pilih Buat fungsi.
6. Pilih tab Kode untuk membuka editor.
7. Timpa kode placeholder `lambda_function.py` dengan yang berikut ini:

```
from os import environ

import boto3
from botocore.config import Config

# load the place index name from the environment, falling back to a default
PLACE_INDEX_NAME = environ.get("PLACE_INDEX_NAME", "AuroraPlaceIndex")

location = boto3.client("location", config=Config(user_agent="Amazon Aurora
  PostgreSQL"))

"""
This Lambda function receives a payload from Amazon Aurora and translates it to
an Amazon Location `SearchPlaceIndex` call and returns the results as-is, to be
post-processed by a PL/pgSQL function.
"""
def lambda_handler(event, context):
    kwargs = {}

    if event.get("biasPosition") is not None:
        kwargs["BiasPosition"] = event["biasPosition"]

    if event.get("filterBBox") is not None:
        kwargs["FilterBBox"] = event["filterBBox"]

    if event.get("filterCountries") is not None:
        kwargs["FilterCountries"] = event["filterCountries"]

    if event.get("maxResults") is not None:
        kwargs["MaxResults"] = event["maxResults"]

    return location.search_place_index_for_text(
        IndexName=PLACE_INDEX_NAME,
        Text=event["text"],
        **kwargs)["Results"]
```

8. Jika Anda telah menamai indeks tempat Anda selain *AuroraPlaceIndex*, buat variabel lingkungan bernama `PLACE_INDEX_NAME` untuk menetapkan nama sumber daya ke:
- Dari tab Konfigurasi, pilih Variabel Lingkungan.
  - Pilih Edit, lalu pilih Tambahkan variabel lingkungan.
  - Untuk Kunci: Enter `PLACE_INDEX_NAME`.

- Untuk Nilai: Masukkan nama sumber daya indeks tempat Anda.
9. Pilih Deploy untuk menyimpan fungsi yang diperbarui.
  10. Dari menu drop-down Test, pilih Configure test Event.
  11. Pilih Buat peristiwa pengujian baru.
  12. Masukkan acara pengujian berikut:

```
{
  "text": "Baker Beach",
  "biasPosition": [-122.483, 37.790],
  "filterCountries": ["USA"]
}
```

13. Pilih Uji untuk menguji fungsi Lambda.
14. Pilih tab Konfigurasi.
15. Di bawah Konfigurasi umum: Pilih Izin.
16. Di bawah Peran eksekusi: Pilih nama Peran yang ditautkan hiper untuk memberikan izin Amazon Location Service ke fungsi Lambda Anda.
17. Di bawah tab Izin: Pilih drop-down Tambahkan izin, lalu pilih Buat kebijakan sebaris.
18. Pilih tab JSON.
19. Tambahkan kebijakan IAM berikut:
  - Kebijakan berikut memberikan izin untuk mengirim SearchPlaceIndexForText ke sumber daya indeks tempat *AuroraPlaceIndex*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "geo:SearchPlaceIndexForText",
      "Resource": "arn:aws:geo:<Region>:<AccountId>:place-index/AuroraPlaceIndex"
    }
  ]
}
```

20. Pilih Tinjau kebijakan.
21. Masukkan nama kebijakan. Misalnya, *AuroraPlaceIndexReadOnly*.

## 22. Pilih Buat kebijakan.

### Berikan Amazon Aurora PostgreSQL akses ke AWS Lambda

Sebelum Amazon Aurora PostgreSQL dapat menjalankan AWS Lambda fungsi, Anda harus memberikan izin akses.

Jika Amazon Aurora PostgreSQL klaster Anda tidak dapat diakses publik, Anda harus terlebih dahulu membuat titik akhir VPC AWS Lambda agar Amazon Aurora memanggil fungsi Lambda Anda.

#### Buat Endpoint VPC untuk AWS Lambda

##### Note

Langkah ini hanya diperlukan jika Amazon Aurora PostgreSQL klaster Anda tidak dapat diakses publik.

1. Buka [Amazon Virtual Private Cloud Console](#).
2. Di navigasi kiri, pilih Endpoints.
3. Pilih Buat titik akhir.
4. Di filter Nama Layanan, masukkan “lambda”, lalu pilih `com.amazonaws.<region>.lambda`.
5. Pilih VPC yang berisi cluster Aurora Anda.
6. Pilih subnet untuk setiap zona ketersediaan.
7. Di filter grup Keamanan, masukkan “default” atau nama grup keamanan cluster Aurora Anda adalah anggota, lalu pilih grup keamanan.
8. Pilih Buat titik akhir.

#### Buat kebijakan IAM untuk memberikan izin untuk menjalankan fungsi Anda AWS Lambda

1. Buka [konsol IAM](#).
2. Di navigasi kiri, perluas Manajemen Akses untuk memilih Kebijakan.
3. Pilih Buat kebijakan.
4. Pada tab JSON, masukkan kebijakan berikut:



- Berikut ini adalah contoh kebijakan IAM yang memberikan Amazon Aurora PostgreSQL izin untuk menjalankan fungsi. AuroraGeocoder AWS Lambda

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "lambda:InvokeFunction",
      "Resource": [
        "arn:aws:lambda:<Region>:<AccountId>:function:AuroraGeocoder"
      ]
    }
  ]
}
```

5. Pilih Berikutnya: Tag untuk menambahkan tag opsional.
6. Pilih Berikutnya: Peninjauan.
7. Tinjau kebijakan Anda dan masukkan detail berikut untuk kebijakan ini:
  - Nama - Gunakan karakter alfanumerik dan '+=, .@-\_' . Maksimal 128 karakter. Misalnya, *AuroraGeocoderInvoke*.
  - Deskripsi - Masukkan deskripsi opsional. Gunakan karakter alfanumerik dan '+=, .@-\_' . Maksimal 1000 karakter.
8. Pilih Buat kebijakan. Perhatikan ARN untuk kebijakan ini, yang Anda gunakan untuk melampirkan kebijakan ke peran IAM.

Buat peran IAM untuk memberikan izin ke Amazon Relational Database Service (Amazon RDS)

Dengan membuat peran IAM, Amazon Aurora PostgreSQL dapat mengambil peran atas nama Anda untuk mengakses fungsi Lambda Anda. Untuk informasi selengkapnya, lihat [Membuat peran untuk mendelegasikan izin ke pengguna IAM](#) dalam Panduan Pengguna IAM.

Contoh berikut adalah AWS CLI perintah yang menciptakan peran bernama

*AuroraGeocoderInvokeRole*:

```
aws iam create-role --role-name rds-lambda-role --assume-role-policy-document '{
  "Version": "2012-10-17",
```

```

    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
          "Service": "rds.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
      }
    ]
  }'

```

Lampirkan kebijakan IAM Anda ke peran IAM

Bila Anda memiliki peran IAM, lampirkan kebijakan IAM yang telah Anda buat.

Contoh berikut adalah AWS CLI perintah yang melampirkan kebijakan *AuroraGeocoderInvoke* ke peran *AuroraGeocoderInvokeRole*.

```

aws iam attach-role-policy --policy-arn AuroraGeocoderInvoke --role-
name AuroraGeocoderInvokeRole

```

Tambahkan peran IAM ke cluster Amazon Aurora DB

Contoh berikut adalah AWS CLI perintah untuk menambahkan peran IAM ke cluster Amazon Aurora PostgreSQL DB bernama *MyAuroraCluster*.

```

aws rds add-role-to-db-cluster \
--db-cluster-identifier MyAuroraCluster \
--feature-name Lambda \
--role-arn AuroraGeocoderInvokeRole \
--region your-region

```

## Memanggil fungsi AWS Lambda

Setelah Anda memberikan izin Amazon Aurora PostgreSQL untuk menjalankan fungsi Lambda geocoding Anda, Anda dapat membuat fungsi yang Amazon Aurora PostgreSQL ditentukan pengguna untuk menjalankan fungsi geocoding. AWS Lambda Untuk informasi selengkapnya, lihat [Memanggil AWS Lambda fungsi dari kluster Amazon Aurora PostgreSQL DB](#) di Panduan Pengguna Amazon Aurora.

Instal ekstensi PostgreSQL yang diperlukan

Untuk menginstal ekstensi dan `aws_lambda` ekstensi PostgreSQL yang diperlukan, [lihat `aws\_lambda` Ringkasan penggunaan fungsi Lambda di Panduan Pengguna Amazon Aurora](#).

```
CREATE EXTENSION IF NOT EXISTS aws_lambda CASCADE;
```

Instal ekstensi PostGIS yang diperlukan

PostGIS adalah ekstensi dari PostgreSQL untuk menyimpan dan mengelola informasi spasial. Untuk informasi selengkapnya, lihat [Bekerja dengan ekstensi PostGIS di Panduan Pengguna Layanan Amazon Relational Database Service](#).

```
CREATE EXTENSION IF NOT EXISTS postgis;
```

Buat fungsi yang ditentukan pengguna SQL yang memanggil fungsi Lambda

Dalam editor SQL, buat fungsi baru yang ditentukan pengguna `f_SearchPlaceIndexForText` untuk memanggil fungsi: *AuroraGeocoder*

```
CREATE OR REPLACE FUNCTION f_SearchPlaceIndexForText(
  text text,
  bias_position geometry(Point, 4326) DEFAULT NULL,
  filter_bbox box2d DEFAULT NULL,
  filter_countries text[] DEFAULT NULL,
  max_results int DEFAULT 1
)
RETURNS TABLE (
  label text,
  address_number text,
  street text,
  municipality text,
  postal_code text,
  sub_region text,
  region text,
  country text,
  geom geometry(Point, 4326)
)
LANGUAGE plpgsql
IMMUTABLE
AS $function$
begin
  RETURN QUERY
  WITH results AS (
```

```

SELECT json_array_elements(payload) rsp
FROM aws_lambda.invoke(
  aws_commons.create_lambda_function_arn('AuroraGeocoder'),
  json_build_object(
    'text', text,
    'biasPosition',
    CASE WHEN bias_position IS NOT NULL THEN
      array_to_json(ARRAY[ST_X(bias_position), ST_Y(bias_position)])
    END,
    'filterBBox',
    CASE WHEN filter_bbox IS NOT NULL THEN
      array_to_json(ARRAY[ST_XMin(filter_bbox), ST_YMin(filter_bbox),
ST_XMax(filter_bbox), ST_YMax(filter_bbox)])
    END,
    'filterCountries', filter_countries,
    'maxResults', max_results
  )
)
)
)
SELECT
  rsp->'Place'->'Label' AS label,
  rsp->'Place'->'AddressNumber' AS address_number,
  rsp->'Place'->'Street' AS street,
  rsp->'Place'->'Municipality' AS municipality,
  rsp->'Place'->'PostalCode' AS postal_code,
  rsp->'Place'->'SubRegion' AS sub_region,
  rsp->'Place'->'Region' AS region,
  rsp->'Place'->'Country' AS country,
  ST_GeomFromGeoJSON(
    json_build_object(
      'type', 'Point',
      'coordinates', rsp->'Place'->'Geometry'->'Point'
    )
  ) geom
FROM results;
end;
$function$;

```

## Panggil fungsi SQL ke geocode dari Aurora

Menjalankan pernyataan SQL memanggil fungsi Lambda *AuroraGeocoder*, yang mengambil catatan alamat dari tabel database dalam Amazon Aurora PostgreSQL database dan geocode mereka menggunakan sumber daya indeks tempat.

**Note**

Amazon Aurora PostgreSQL memanggil fungsi Lambda untuk setiap panggilan ke fungsi yang ditentukan pengguna SQL.

Jika Anda melakukan geocoding 50 baris, Amazon Aurora PostgreSQL memanggil fungsi Lambda 50 kali. Satu doa untuk setiap baris.

Fungsi `f_SearchPlaceIndexForText` SQL berikut membuat permintaan ke Amazon Location [SearchPlaceIndexForText](#) API melalui fungsi `AuroraGeocoder` Lambda. Fungsi mengembalikan geom kolom yang merupakan geometri PostGIS, `ST_AsText(geom)` yang mengkonversi ke teks.

```
SELECT *, ST_AsText(geom)
FROM f_SearchPlaceIndexForText('Vancouver, BC');
```

Secara default, pengembalian akan berisi satu baris. Untuk meminta baris tambahan, hingga `MaxResults` batas, jalankan pernyataan SQL berikut sambil memberikan `BiasPosition` dan membatasi hasil di Kanada.

```
SELECT *
FROM f_SearchPlaceIndexForText('Mount Pleasant', ST_MakePoint(-123.113, 49.260), null,
'{"CAN"}', 5);
```

Untuk memfilter hasil menggunakan kotak pembatas, lalu berikan [Box2D](#) `asfilter_bbox`:

- [FilterBBox](#)— Memfilter hasil dengan mengembalikan tempat di dalam kotak pembatas. Ini adalah parameter opsional.

```
SELECT *
FROM f_SearchPlaceIndexForText('Mount Pleasant', null, 'BOX(-139.06 48.30, -114.03
60.00)>:::box2d, '{"CAN"}', 5);
```

[Untuk informasi selengkapnya tentang jenis dan fungsi PostGIS, lihat Referensi PostGIS.](#)

## Memperkaya database yang berisi data alamat

Anda dapat membuat alamat yang diformat dan secara bersamaan menormalkan dan geocode menggunakan operasi Lokasi Amazon yang `SearchPlaceIndexForText` diberikan tabel database dengan data berikut dipecah menjadi kolom berikut:

- `id`
- `address`
- `city`
- `state`
- `zip`

```
WITH source_data AS (
  SELECT
    id,
    address || ', ' || city || ', ' || state || ', ' || zip AS formatted_address
  FROM addresses
),
geocoded_data AS (
  SELECT
    *,
    (f_SearchPlaceIndexForText(formatted_address)).*
  FROM source_data
)
SELECT
  id,
  formatted_address,
  label normalized_address,
  ST_Y(geom) latitude,
  ST_X(geom) longitude
FROM geocoded_data
-- limit the number of rows that will be geocoded; remove this to geocode the entire
table
LIMIT 1;
```

Contoh berikut menggambarkan satu baris datatable yang dihasilkan:

id	formatted_address	normalized_address
latitude	longitude	

```

-----+-----+-----
+-----+-----
42 | 123 Anytown Ave N, Seattle, WA | 123 Anytown Ave N, Seattle, WA, 12345, USA |
47.6223000127926 | -122.336745971039
(1 row)

```

Perbarui tabel database dan isi kolom

Contoh berikut memperbarui tabel dan mengisi kolom dengan hasil SearchPlaceIndexForText query:

```

WITH source_data AS (
  -- select rows that have not been geocoded and created a formatted address for each
  SELECT
    id,
    address || ', ' || city || ', ' || state || ', ' || zip AS formatted_address
  FROM addresses
  WHERE label IS NULL
  -- limit the number of rows that will be geocoded; remove this to geocode the entire
  table
  LIMIT 1
),
geocoded_data AS (
  -- geocode each row and keep it linked to the source's ID
  SELECT
    id,
    (f_SearchPlaceIndexForText(formatted_address)).*
  FROM source_data
)
UPDATE addresses
-- populate columns
SET
  normalized_address = geocoded_data.label,
  latitude = ST_Y(geocoded_data.geom),
  longitude = ST_X(geocoded_data.geom)
FROM geocoded_data
-- ensure that rows match
WHERE addresses.id = geocoded_data.id;

```

## Langkah selanjutnya

Kode sampel tersedia di repositori sampel Amazon Location Service [GitHub](#), [yang AWS CloudFormation](#) menyertakan templat.

## Mengelola sumber daya indeks tempat Anda

Anda dapat mengelola sumber daya indeks tempat menggunakan konsol Lokasi Amazon, API Lokasi AmazonAWS CLI, atau Amazon Location.

### Buat daftar sumber daya indeks tempat Anda

Anda dapat melihat daftar sumber daya indeks tempat menggunakan konsol Lokasi Amazon, API Lokasi AmazonAWS CLI, atau Amazon Location:

#### Console

Untuk melihat daftar sumber daya indeks tempat menggunakan konsol Lokasi Amazon

1. Buka konsol Lokasi Amazon di <https://console.aws.amazon.com/location/>.
2. Pilih Tempatkan indeks dari panel navigasi kiri.
3. Lihat daftar sumber daya indeks tempat Anda di bawah indeks tempat saya.

#### API

Gunakan [ListPlaceIndexes](#) operasi dari Amazon Location Places API.

Contoh berikut adalah permintaan API untuk mendapatkan daftar sumber daya indeks tempat di AWS akun.

```
POST /places/v0/list-indexes
```

Berikut ini adalah contoh respon untuk [ListPlaceIndexes](#):

```
{
  "Entries": [
    {
      "CreateTime": 2020-10-30T01:38:36Z,
      "DataSource": "Esri",
      "Description": "string",
      "IndexName": "ExamplePlaceIndex",
      "UpdateTime": 2020-10-30T01:40:36Z
    }
  ],
  "NextToken": "1234-5678-9012"
```



```
}
```

## CLI

Gunakan perintah [list-place-indexes](#).

Contoh berikut adalah AWS CLI untuk mendapatkan daftar sumber daya indeks tempat di AWS akun.

```
aws location list-place-indexes
```

## Dapatkan detail sumber daya indeks tempat

Anda bisa mendapatkan detail tentang sumber daya indeks tempat apa pun di AWS akun Anda menggunakan konsol Lokasi Amazon, API Lokasi AmazonAWS CLI, atau Amazon Location:

### Console

Untuk melihat detail sumber daya indeks tempat menggunakan konsol Lokasi Amazon

1. Buka konsol Lokasi Amazon di <https://console.aws.amazon.com/location/>.
2. Pilih Tempatkan indeks dari panel navigasi kiri.
3. Di bawah Indeks tempat saya, pilih tautan nama sumber daya indeks tempat target.

### API

Gunakan [DescribePlaceIndex](#) operasi dari Amazon Location Place API.

Contoh berikut adalah permintaan API untuk mendapatkan rincian sumber daya indeks tempat untuk *ExamplePlaceIndex*.

```
GET /places/v0/indexes/ExamplePlaceIndex
```

Berikut ini adalah contoh respon untuk [DescribePlaceIndex](#):

```
{
  "CreateTime": 2020-10-30T01:38:36Z,
  "DataSource": "Esri",
  "DataSourceConfiguration": {
```

```
    "IntendedUse": "SingleUse"
  },
  "Description": "string",
  "IndexArn": "arn:aws:geo:us-west-2:123456789012:place-indexes/ExamplePlaceIndex",
  "IndexName": "ExamplePlaceIndex",
  "Tags": {
    "string" : "string"
  },
  "UpdateTime": 2020-10-30T01:40:36Z
}
```

## CLI

Gunakan perintah [describe-place-index](#).

Contoh berikut adalah AWS CLI untuk mendapatkan rincian sumber daya indeks tempat untuk *ExamplePlaceIndex*.

```
aws location describe-place-index \
  --index-name "ExamplePlaceIndex"
```

## Hapus sumber daya indeks tempat

Anda dapat menghapus sumber daya indeks tempat dari akun AWS menggunakan konsol Lokasi Amazon, API Lokasi AmazonAWS CLI, atau Amazon Location:

### Console

Untuk menghapus sumber daya indeks tempat menggunakan konsol Lokasi Amazon

#### Warning

Operasi ini menghapus sumber daya secara permanen.

1. Buka konsol Lokasi Amazon di <https://console.aws.amazon.com/location/>.
2. Pilih Tempatkan indeks dari panel navigasi kiri.
3. Di bawah Indeks tempat saya, pilih sumber daya indeks tempat target.
4. Pilih Hapus indeks tempat.

## API

Gunakan [DeletePlaceIndex](#) operasi dari Amazon Location Places API.

Contoh berikut adalah permintaan API untuk menghapus sumber daya indeks tempat *ExamplePlaceIndex*.

```
DELETE /places/v0/indexes/ExamplePlaceIndex
```

Berikut ini adalah contoh respon sukses untuk [DeletePlaceIndex](#):

```
HTTP/1.1 200
```

## CLI

Gunakan perintah [delete-place-index](#).

Contoh berikut adalah AWS CLI perintah untuk menghapus sumber daya indeks tempat *ExamplePlaceIndex*.

```
aws location delete-place-index \  
  --index-name "ExamplePlaceIndex"
```

## Menghitung rute menggunakan Amazon Location Service

Lokasi Amazon memungkinkan Anda memilih penyedia data untuk menghitung rute dengan membuat dan mengonfigurasi sumber daya kalkulator rute.

Anda dapat menggunakan sumber daya kalkulator rute untuk [menghitung rute yang](#) diberikan parameter tertentu menggunakan AWS SDK, atau titik akhir REST API. Gunakan sumber kalkulator rute ini untuk menghitung rute antara asal, tujuan, dan hingga 23 titik arah untuk berbagai moda transportasi, penghindaran, dan kondisi lalu lintas.

Anda juga dapat menggunakan sumber daya kalkulator rute untuk membuat input untuk algoritme perencanaan rute atau produk Anda dengan [menghitung matriks rute](#). Hitung waktu perjalanan dan jarak tempuh antara satu set posisi keberangkatan dan satu set posisi tujuan. Perangkat lunak perencanaan rute dapat menggunakan data waktu dan jarak tersebut untuk mengoptimalkan rute atau serangkaian rute; misalnya, jika Anda merencanakan beberapa rute pengiriman, dan ingin

menemukan rute dan waktu terbaik untuk setiap pemberhentian. Anda dapat menghitung matriks rute untuk berbagai moda transportasi, penghindaran, dan kondisi lalu lintas.

### Note

Untuk ikhtisar konsep routing, lihat [Rute](#).

## Topik

- [Prasyarat](#)
- [Hitung rute](#)
- [Perencanaan rute dengan matriks rute](#)
- [Posisi tidak terletak di jalan](#)
- [Waktu keberangkatan](#)
- [Mode perjalanan](#)
- [Mengelola sumber daya kalkulator rute Anda](#)

## Prasyarat

Sebelum Anda mulai menghitung rute, ikuti langkah-langkah prasyarat:

### Topik

- [Buat sumber daya kalkulator rute](#)
- [Mengautentikasi permintaan Anda](#)

## Buat sumber daya kalkulator rute

Sebelum Anda dapat menghitung rute, buat sumber kalkulator rute di AWS akun Anda.

Saat membuat sumber kalkulator rute, Anda dapat memilih dari penyedia data yang tersedia:

1. Esri — Untuk informasi lebih lanjut tentang cakupan Esri di wilayah minat Anda, lihat [detail Esri di jaringan jalan dan](#) cakupan lalu lintas.
2. HERE Technologies — Untuk informasi lebih lanjut tentang cakupan HERE di wilayah yang Anda minati, lihat cakupan [perutean mobil DI SINI dan cakupan perutean truk HERE](#).

3. Grab — Untuk informasi selengkapnya tentang cakupan Grab, lihat [Negara/wilayah dan area yang dicakup](#).

#### Note

Jika aplikasi Anda melacak atau merutekan aset yang Anda gunakan dalam bisnis Anda, seperti kendaraan pengiriman atau karyawan, Anda tidak boleh menggunakan Esri sebagai penyedia geolokasi Anda. Lihat bagian 82 dari [ketentuan AWS layanan](#) untuk detail selengkapnya.

Anda dapat melakukannya menggunakan konsol Amazon Location Service, theAWS CLI, atau Amazon Location API.

#### Console

Untuk membuat sumber daya kalkulator rute menggunakan konsol Lokasi Amazon

1. Buka konsol Lokasi Amazon di <https://console.aws.amazon.com/location/>.
2. Di panel navigasi kiri, pilih Kalkulator rute.
3. Pilih Buat kalkulator rute.
4. Isi kotak-kotak berikut:
  - Nama — Masukkan nama untuk sumber kalkulator rute. Misalnya, *ExampleCalculator*. Maksimal 100 karakter. Entri yang valid termasuk karakter alfanumerik, tanda hubung, titik, dan garis bawah.
  - Deskripsi - Masukkan deskripsi opsional.
5. Untuk penyedia data, pilih [penyedia data](#) untuk digunakan sebagai kalkulator rute.
6. (Opsional) Di bawah Tag, masukkan tag Kunci dan Nilai. Ini menambahkan tag sumber kalkulator rute baru Anda. Untuk informasi selengkapnya, lihat [Menandai sumber daya Anda](#).
7. Pilih Buat kalkulator rute.

#### API

Untuk membuat sumber daya kalkulator rute menggunakan Amazon Location API

Gunakan [CreateRouteCalculator](#) operasi dari Amazon Location Places API.

Contoh berikut adalah permintaan API untuk membuat sumber daya kalkulator rute yang disebut *ExampleCalculator* menggunakan penyedia data *Esri*.

```
POST /routes/v0/calculators
Content-type: application/json

{
  "CalculatorName": "ExampleCalculator",
  "DataSource": "Esri",
  "Description": "string",
  "Tags": {
    "Tag1" : "Value1"
  }
}
```

## AWS CLI

Untuk membuat sumber daya kalkulator rute menggunakan AWS CLI perintah

Gunakan perintah `create-route-calculator`.

Contoh berikut membuat sumber kalkulator rute yang disebut *ExampleCalculator* menggunakan *Esri* sebagai penyedia data.

```
aws location \
  create-route-calculator \
  --calculator-name "ExampleCalculator" \
  --data-source "Esri" \
  --tags Tag1=Value1
```

### Note

Penagihan tergantung pada penggunaan Anda. Anda mungkin dikenakan biaya untuk penggunaan AWS layanan lain. Untuk informasi selengkapnya, lihat [harga Amazon Location Service](#).

## Mengautentikasi permintaan Anda

Setelah Anda membuat sumber daya kalkulator rute dan Anda siap untuk mulai membangun fitur lokasi ke dalam aplikasi Anda, pilih bagaimana Anda akan mengautentikasi permintaan Anda:

- Untuk mempelajari cara mengakses layanan, lihat [Mengakses Amazon Location Service](#).
- Jika Anda memiliki situs web dengan pengguna anonim, Anda mungkin ingin menggunakan Kunci API atau Amazon Cognito.

### Contoh

Contoh berikut menunjukkan penggunaan kunci API untuk otorisasi, menggunakan [AWS JavaScript SDK v3](#), dan Lokasi Amazon. [JavaScript Pembantu otentikasi](#)

```
import { LocationClient, CalculateRouteCommand } from "@aws-sdk/client-location";
import { withAPIKey } from "@aws/amazon-location-utilities-auth-helper";

const apiKey = "v1.public.your-api-key-value"; // API key

// Create an authentication helper instance using an API key
const authHelper = await withAPIKey(apiKey);

const client = new LocationClient({
  region: "<region>", // region containing Cognito pool
  ...authHelper.getLocationClientConfig(), // Provides configuration required to make
  requests to Amazon Location
});

const input = {
  CalculatorName: "ExampleCalculator",
  DeparturePosition: [-123.4567, 45.6789],
  DestinationPosition: [-123.123, 45.234],
};

const command = new CalculateRouteCommand(input);

const response = await client.send(command);
```

## Hitung rute

Anda dapat menggunakan Amazon Location Service untuk menghitung rute antara asal dan tujuan, dengan hingga 23 titik arah di sepanjang rute, untuk berbagai moda transportasi, penghindaran, dan kondisi lalu lintas.

### Note

Anda harus terlebih dahulu membuat sumber daya kalkulator rute dan menyiapkan otentikasi untuk permintaan Anda ke Lokasi Amazon. Untuk informasi selengkapnya, lihat [Prasyarat](#).

## Mulai menghitung rute

Kirim permintaan sederhana dengan menggunakan [CalculateRoute](#) operasi. Permintaan sederhana berisi bidang wajib berikut:

- `DeparturePosition`— Posisi awal untuk menghitung rute dari. Didefinisikan sebagai `[longitude, latitude]`
- `DestinationPosition`— Posisi akhir untuk menghitung rute. Didefinisikan sebagai `[longitude, latitude]`.

### Note

Jika Anda menentukan posisi keberangkatan atau tujuan yang tidak terletak di jalan, Lokasi Amazon [memindahkan posisi ke jalan terdekat](#).

Anda dapat secara opsional menentukan [titik arah](#), [waktu keberangkatan](#), dan [mode perjalanan dalam permintaan](#) Anda.

Anda dapat menggunakan AWS CLI atau Amazon Location API.

## API

Contoh berikut adalah `CalculateRoute` permintaan menggunakan sumber kalkulator rute *ExampleCalculator*. *Permintaan menentukan penghitungan rute dari posisi keberangkatan `[-122.7565, 49.0021]` ke posisi tujuan `[-122.3394, 47.6159]`.*



```
POST /routes/v0/calculators/ExampleCalculator/calculate/route
Content-type: application/json
{
  "DeparturePosition": [-122.7565, 49.0021],
  "DestinationPosition": [-122.3394, 47.6159]
}
```

## AWS CLI

Contoh berikut adalah `calculate-route` perintah menggunakan sumber kalkulator rute *ExampleCalculator*. *Permintaan menentukan penghitungan rute dari posisi keberangkatan [-122.7565, 49.0021] ke posisi tujuan [-122.3394, 47.6159].*

```
aws location \
  calculate-route \
    --calculator-name ExampleCalculator \
    --departure-position -122.7565 49.0021 \
    --destination-position -122.3394 47.6159
```

Secara default, respons kembali `Distance` dalam kilometer. Anda dapat mengubah satuan pengukuran menjadi mil menggunakan parameter opsional berikut:

- `DistanceUnit`- Menentukan sistem unit yang akan digunakan untuk hasil jarak.


## Example

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route
Content-type: application/json
{
  "DeparturePosition": [-122.7565, 49.0021],
  "DestinationPosition": [-122.3394, 47.6159],
  "DistanceUnit": "Miles"
}
```

## Mengatur titik arah

Saat menghitung rute, Anda dapat menentukan hingga 23 titik persinggahan antara posisi keberangkatan dan posisi tujuan menggunakan posisi titik arah.

- **WaypointPositions**— Menentukan daftar urutan posisi menengah untuk memasukkan sepanjang rute antara posisi keberangkatan dan posisi tujuan.

 Note

Jika Anda menentukan posisi waypoint yang tidak terletak di jalan, Amazon Location memindahkan posisi ke jalan terdekat.

## Example

[CalculateRoute](#)Permintaan berikut menghitung rute dengan 2 titik arah:

- Posisi keberangkatan adalah [-122.7565, 49.0021], dan posisi tujuan adalah [-122.3394, 47.6159].
- Untuk parameter permintaan **WaypointPositions**:
  - Posisi stop over pertama adalah [-122.1884, 48.0936].
  - Posisi stop over kedua adalah [-122.3493, 47.6205].
- *Untuk menyertakan geometri linestring kaki antara dua titik arah ini, atur parameter opsional berikut ke true:*
  - **IncludeLegGeometry**— Termasuk geometri setiap jalur antara sepasang posisi dalam respons.

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route
Content-type: application/json
{
  "DeparturePosition": [-122.7565,49.0021],
  "DestinationPosition": [-122.3394, 47.6159],
  "WaypointPositions":[
    [-122.1884,48.0936],
    [-122.3493,47.6205]
  ],
  "IncludeLegGeometry": true
}
```

## Contoh tanggapan

Berikut ini adalah contoh permintaan dengan respons yang sesuai saat memanggil [CalculateRoute](#) operasi dari Amazon Location Routes API dengan `IncludeLegGeometry` set ke `true`, yang mencakup geometri linestring dari setiap jalur antara sepasang posisi dalam respons.

### Example request

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route
Content-type: application/json
{
  "DeparturePosition": [-122.7565,49.0021],
  "DestinationPosition": [-122.3394, 47.6159],
  "IncludeLegGeometry": true
}
```

### Example response

```
{
  "Legs": [
    {
      "Distance": 178.5,
      "DurationSeconds": 6480,
      "EndPosition": [-122.3394,47.6159],
      "Geometry": {
        "LineString": [
          [-122.7565,49.0021],
          [-122.3394,47.6159]
        ]
      },
      "StartPosition": [-122.7565,49.0021],
      "Steps": [
        {
          "Distance": 178.5,
          "DurationSeconds": 6480,
          "EndPosition": [-122.3394,47.6159],
          "GeometryOffset": 0,
          "StartPosition": [-122.7565,49.0021]
        }
      ]
    }
  ],
  "Summary": {
```

```
"DataSource": "Esri",
"Distance": 178.5,
"DistanceUnit": "Kilometers",
"DurationSeconds": 6480,
"RouteBoundingBox": [
  -122.7565,49.0021,
  -122.3394,47.6159
]
}
```

## Perencanaan rute dengan matriks rute

Anda dapat menggunakan Amazon Location Service untuk membuat input ke perangkat lunak perencanaan dan pengoptimalan rute Anda. Anda dapat membuat hasil rute, termasuk waktu perjalanan dan jarak perjalanan, untuk rute antara serangkaian posisi keberangkatan dan serangkaian posisi tujuan.

Misalnya, mengingat posisi keberangkatan A dan B, dan posisi tujuan X dan Y, Amazon Location Service akan mengembalikan waktu perjalanan dan jarak perjalanan untuk rute dari A ke X, A ke Y, B ke X, dan B ke Y.

Anda dapat menghitung rute dengan berbagai moda transportasi, penghindaran, dan kondisi lalu lintas. Misalnya, Anda dapat menentukan bahwa kendaraan tersebut adalah truk yang panjangnya 35 kaki, dan rute yang dihitung akan menggunakan batasan tersebut untuk menentukan waktu tempuh dan jarak tempuh.

Jumlah hasil yang dikembalikan (dan rute dihitung) adalah jumlah posisi keberangkatan dikalikan dengan jumlah posisi tujuan. Anda dikenakan biaya untuk setiap rute yang dihitung, bukan setiap permintaan ke layanan, sehingga matriks rute dengan 10 keberangkatan dan 10 tujuan akan ditagih sebagai 100 rute.

## Menghitung matriks rute

Anda dapat menghitung matriks rute antara satu set posisi keberangkatan dan serangkaian posisi tujuan. Hasil rute akan mencakup waktu tempuh dan jarak tempuh.

## Prasyarat

- Anda harus terlebih dahulu membuat sumber daya kalkulator rute dan menyiapkan otentikasi untuk permintaan Anda ke Lokasi Amazon. Untuk informasi selengkapnya, lihat [Prasyarat](#).

Kirim permintaan dengan menggunakan [CalculateRouteMatrix](#) operasi. Permintaan minimal berisi bidang wajib berikut:

- `DeparturePositions`— Kumpulan posisi awal untuk menghitung rute. Didefinisikan sebagai array `[longitude, latitude]`
- `DestinationPositions`— Kumpulan posisi akhir untuk menghitung rute. Didefinisikan sebagai array dari `[longitude, latitude]`.

#### Note

Jika Anda menentukan posisi keberangkatan atau tujuan yang tidak terletak di jalan, Lokasi Amazon [memindahkan posisi ke jalan terdekat](#).

Anda dapat secara opsional menentukan [waktu keberangkatan](#), dan [mode perjalanan](#) dalam permintaan Anda.

Anda dapat menggunakan AWS CLI atau Amazon Location API.

## API

Contoh berikut adalah `CalculateRouteMatrix` permintaan menggunakan sumber kalkulator rute *ExampleCalculator*. *Permintaan menentukan penghitungan matriks rute dari posisi keberangkatan [-122.7565, 49.0021] dan [-122.2014, 47.6101] ke posisi tujuan [-122.3394, 47.6159] dan [-122.4813, 48.7511].*

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route-matrix
Content-type: application/json
{
  "DeparturePositions": [
    [-122.7565, 49.0021],
    [-122.2014, 47.6101]
  ],
  "DestinationPositions": [
    [-122.3394, 47.6159],
    [-122.4813, 48.7511]
  ]
}
```

```
]
}
```

## AWS CLI

Contoh berikut adalah `calculate-route-matrix` perintah menggunakan sumber kalkulator rute *ExampleCalculator*. *Permintaan menentukan penghitungan matriks rute dari posisi keberangkatan [-122.7565, 49.0021] dan [-122.2014, 47.6101] ke posisi tujuan [-122.3394, 47.6159] dan [-122.4813, 48.7511].*

```
aws location \
  calculate-route-matrix \
    --calculator-name ExampleCalculator \
    --departure-positions "[[-122.7565,49.0021],[-122.2014,47.6101]]" \
    --destination-positions "[[-122.3394,47.6159],[-122.4813,48.7511]]"
```

Secara default, respons kembali Distance dalam kilometer. Anda dapat mengubah satuan pengukuran menjadi mil menggunakan parameter opsional berikut:

- `DistanceUnit`- Menentukan sistem unit yang akan digunakan untuk hasil jarak.

## Example

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route-matrix
Content-type: application/json
{
  "DeparturePositions": [
    [-122.7565,49.0021],
    [-122.2014,47.6101]
  ],
  "DestinationPositions": [
    [-122.3394, 47.6159],
    [-122.4813,48.7511]
  ],
  "DistanceUnit": "Miles"
}
```

## Pembatasan posisi keberangkatan dan tujuan

Saat menghitung matriks rute, ada batasan pada posisi keberangkatan dan tujuan. Pembatasan ini bervariasi tergantung pada penyedia yang digunakan oleh RouteCalculator sumber daya.

Batasan	Esri	Ambil	KEMARI
Jumlah posisi	Hingga 10 posisi keberangkatan dan 10 posisi tujuan.	Hingga 350 posisi keberangkatan dan 350 posisi tujuan.	Hingga 350 posisi keberangkatan dan 350 posisi tujuan.  Untuk rute yang lebih panjang, pembatasan tambahan berlaku. Lihat <a href="#">bagian</a> .
Jarak antar posisi	Setiap pasangan posisi keberangkatan dan tujuan harus berada dalam jarak 400 km satu sama lain (40 km untuk rute berjalan kaki).		Semua posisi keberangkatan dan tujuan harus berada dalam lingkaran berdiameter 180 km.  Untuk rute yang lebih panjang, pembatasan tambahan berlaku. Lihat <a href="#">bagian</a> .
Panjang rute	Rute tidak akan selesai jika total waktu perjalanan untuk rute tersebut lebih dari 400 menit.		Rute yang menyimpang lebih dari 10 km di luar lingkaran di sekitar titik keberangkatan dan tujuan tidak akan dihitung.  Untuk rute yang lebih panjang, pembatasan

Batasan	Esri	Ambil	KEMARI
			n tambahan berlaku. Lihat <a href="#">bagian</a> .
Wilayah	Menghitung matriks rute tidak didukung di Korea.	Tersedia di Asia Tenggara. Untuk daftar negara/wilayah yang didukung dan informasi selengkapnya, lihat. <a href="#">Negara/wilayah dan area yang dicakup</a>	Tidak ada batasan tambahan.

## Perencanaan rute yang lebih panjang

Menghitung matriks hasil rute berguna untuk perencanaan rute yang efisien, tetapi perhitungannya bisa memakan waktu. Semua penyedia data Amazon Location Service membatasi jumlah rute atau jarak rute yang dapat dihitung. Misalnya, HERE memungkinkan pembuatan rute antara 350 posisi keberangkatan dan tujuan, tetapi posisi tersebut harus berada dalam lingkaran 180 km. Bagaimana jika Anda ingin merencanakan dengan rute yang lebih panjang?

Anda dapat menghitung matriks rute dengan panjang tidak terbatas untuk jumlah rute yang lebih kecil, menggunakan, RouteCalculator dengan HERE sebagai penyedia data. Ini tidak mengubah cara Anda memanggil [CalculateRouteMatrix](#) API, Amazon Location hanya memungkinkan rute yang lebih panjang ketika Anda memenuhi persyaratan.

Persyaratan untuk perhitungan rute panjang yang lebih panjang adalah:

- RouteCalculatorHarus menggunakan penyedia data HERE.
- Jumlah posisi keberangkatan tidak boleh lebih dari 15.
- Jumlah total rute untuk menghitung tidak boleh lebih dari 100.
- Perutean jarak jauh tidak diperbolehkan untuk rute truk dengan penghindaran tol ketika rute lebih dari 1.000 km. Kombinasi ini lebih lambat untuk dihitung, dan dapat menyebabkan panggilan habis. Anda dapat menghitung rute ini secara individual dengan [CalculateRoute](#) operasi.



Jika panggilan Anda tidak memenuhi persyaratan ini (misalnya, Anda meminta 150 perhitungan rute dalam satu panggilan), maka `CalculateRouteMatrix` akan kembali hanya mengizinkan aturan rute yang lebih pendek. Anda kemudian dapat menghitung rute, selama posisi berada dalam lingkaran 180 km.

Saat menghitung rute yang lebih panjang, ingatlah poin-poin ini:

- Rute yang lebih panjang dapat memakan waktu lebih lama untuk dihitung, bahkan lebih lama dari waktu maksimum untuk Amazon Location API. Jika Anda sering mendapatkan batas waktu dengan rute tertentu, Anda dapat mencoba sejumlah kecil rute di setiap panggilan. `CalculateRouteMatrix`
- Jika Anda menambahkan lebih banyak posisi tujuan atau keberangkatan ke `CalculateRouteMatrix` permintaan Anda, operasi dapat beralih ke mode yang lebih terbatas, dan Anda bisa mendapatkan kesalahan untuk rute yang dapat dihitung tanpa masalah ketika ada lebih sedikit rute untuk dibuat. Dalam hal ini, kurangi jumlah posisi tujuan atau keberangkatan, dan buat beberapa permintaan untuk mendapatkan set lengkap perhitungan rute yang Anda butuhkan.

## Contoh tanggapan

Berikut ini adalah contoh permintaan dengan respons yang sesuai saat memanggil [CalculateRouteMatrix](#) operasi dari Amazon Location Routes API.

### Example request

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route-matrix
Content-type: application/json
{
  "DeparturePositions": [
    [-122.7565,49.0021],
    [-122.2014,47.6101]
  ],
  "DestinationPositions": [
    [-122.3394, 47.6159],
    [-122.4813,48.7511]
  ]
}
```

### Example response

```
{
```

```

"RouteMatrix": [
  [
    {
      "Distance": 178.764,
      "DurationSeconds": 7565
    },
    {
      "Distance": 39.795,
      "DurationSeconds": 1955
    }
  ],
  [
    {
      "Distance": 15.31,
      "DurationSeconds": 1217
    },
    {
      "Distance": 142.506,
      "DurationSeconds": 6279
    }
  ]
],
"Summary": {
  "DataSource": "Here",
  "RouteCount": 4,
  "ErrorCount": 0,
  "DistanceUnit": "Kilometers"
}
}

```

## Posisi tidak terletak di jalan

Saat menggunakan `CalculateRoute` atau `CalculateRouteMatrix`, jika Anda menentukan posisi keberangkatan, tujuan, atau titik jalan yang tidak terletak di jalan Amazon Lokasi memindahkan posisi ke jalan terdekat.

[CalculateRoute](#) Permintaan berikut menentukan posisi keberangkatan dan posisi tujuan yang tidak terletak di jalan:

```

POST /routes/v0/calculators/ExampleCalculator/calculate/route
Content-type: application/json
{

```

```
"DeparturePosition": [-123.128014, 49.298472],  
"DestinationPosition": [-123.134701, 49.294315]  
}
```

Respons yang dihasilkan mengembalikan posisi yang tersentak ke jalan terdekat:

```
{  
  "Legs": [  
    {  
      "StartPosition": [-123.12815, 49.29717],  
      "EndPosition": [-123.13375, 49.2926],  
      "Distance": 4.223,  
      "DurationSeconds": 697,  
      "Steps": [  
        {  
          "StartPosition": [-123.12815, 49.29717 ],  
          "EndPosition": [-123.12806, 49.29707 ],  
          "Distance": 0.013,  
          "DurationSeconds": 8  
        },  
        {  
          "StartPosition": [-123.12806, 49.29707 ],  
          "EndPosition": [-123.1288, 49.29659 ],  
          "Distance": 0.082,  
          "DurationSeconds": 36  
        },  
        {  
          "StartPosition": [-123.1288, 49.29659 ],  
          "EndPosition": [-123.12021, 49.29853 ],  
          "Distance": 0.742,  
          "DurationSeconds": 128  
        },  
        {  
          "StartPosition": [-123.12021, 49.29853 ],  
          "EndPosition": [-123.1201, 49.29959 ],  
          "Distance": 0.131,  
          "DurationSeconds": 26  
        },  
        {  
          "StartPosition": [-123.1201, 49.29959 ],  
          "EndPosition": [-123.13562, 49.30681 ],  
          "Distance": 1.47,  
          "DurationSeconds": 238  
        }  
      ]  
    }  
  ]  
}
```

```
    },
    {
      "StartPosition": [ -123.13562, 49.30681 ],
      "EndPosition": [ -123.13693, 49.30615 ],
      "Distance": 0.121,
      "DurationSeconds": 28
    },
    {
      "StartPosition": [ -123.13693, 49.30615 ],
      "EndPosition": [ -123.13598, 49.29755 ],
      "Distance": 0.97,
      "DurationSeconds": 156
    },
    {
      "StartPosition": [ -123.13598, 49.29755 ],
      "EndPosition": [ -123.13688, 49.29717 ],
      "Distance": 0.085,
      "DurationSeconds": 15
    },
    {
      "StartPosition": [ -123.13688, 49.29717 ],
      "EndPosition": [ -123.13375, 49.2926 ],
      "Distance": 0.609,
      "DurationSeconds": 62
    }
  ]
}
],
"Summary": {
  "RouteBBox": [ -123.13693, 49.2926, -123.1201, 49.30681 ],
  "DataSource": "Here",
  "Distance": 4.223,
  "DurationSeconds": 697,
  "DistanceUnit": "Kilometers"
}
}
```

## Waktu keberangkatan

Secara default, saat Anda menelepon `CalculateRoute` atau `CalculateRouteMatrix`, jika Anda tidak memberikan waktu keberangkatan dalam permintaan, rute yang dihitung mencerminkan kondisi lalu lintas yang optimal.

Anda dapat mengatur waktu keberangkatan tertentu untuk menggunakan kondisi lalu lintas langsung dan prediktif dari penyedia data pilihan Anda, dengan menggunakan salah satu opsi berikut:

- **DepartNow**— Ketika diatur ke *true*, ia menggunakan kondisi lalu lintas langsung untuk menghitung jalur perjalanan tercepat.
- **DepartureTime**— Ketika disediakan, ia menggunakan kondisi lalu lintas prediktif dan diketahui untuk waktu yang diminta. Didefinisikan dalam [format](#) berikut: `YYYY-MM-DDThh:mm:ss.sssZ`.

## Example

[CalculateRoute](#) Permintaan berikut menetapkan waktu keberangkatan ke 2 Juli 2024, pukul 12:15:20 UTC.

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route
Content-type: application/json
{
  "DeparturePosition": [-122.7565,49.0021],
  "DestinationPosition": [-122.3394, 47.6159],
  "WaypointPositions":[
    [-122.1884,48.0936],
    [-122.3493,47.6205]
  ]
  "IncludeLegGeometry": true,
  "DepartureTime": 2024-07-02T12:15:20.000Z,
}
```

## Mode perjalanan

Anda dapat mengatur mode perjalanan saat menggunakan `CalculateRoute` atau `CalculateRouteMatrix`. Mode perjalanan mempengaruhi kecepatan perjalanan dan kompatibilitas jalan. Meskipun mode default perjalanan adalah dengan mobil, Anda dapat menentukan mode perjalanan yang Anda gunakan saat bepergian di sepanjang rute dengan parameter opsional berikut:

- **TravelMode**— Menentukan moda transportasi saat menghitung rute, seperti: *Bicycle*, *Car*, *MotorcycleTruck*, atau *Walking*.

## Batasan

- Jika Anda menentukan Walking mode perjalanan dan penyedia data Anda adalah Esri, awal dan tujuan harus berada dalam jarak 40km.
- Bicycle atau hanya Motorcycle tersedia saat menggunakan Grab sebagai penyedia data.
- Grab hanya menyediakan Bicycle dan Walking rute di kota-kota tertentu. Untuk informasi selengkapnya, lihat [Negara/wilayah dan area yang dicakup](#).
- Truck tidak tersedia saat menggunakan Grab sebagai penyedia data.

## Preferensi tambahan

Jika Anda menentukan TravelMode dari *Car*, Anda dapat menentukan preferensi rute tambahan dengan parameter opsional berikut:

- CarModeOptions— Menentukan preferensi rute saat bepergian di dalam mobil, seperti *AvoidFerries* atau *AvoidTolls*.

Jika Anda menentukan TravelMode dari *Truck*, Anda dapat menentukan preferensi rute tambahan dengan parameter opsional berikut:

- TruckModeOptions— Menentukan preferensi rute saat bepergian di truk, seperti *AvoidFerries* atau *AvoidTolls*, selain menentukan rute yang dapat mengakomodasi dan *TruckDimensions*. *TruckWeight*

## Example

[CalculateRoute](#) Permintaan berikut menentukan *Truck* sebagai modus perjalanan. Pembatasan rute tambahan meliputi: menghindari rute yang menggunakan feri dan menghindari jalan yang tidak dapat mengakomodasi dimensi dan berat truk.

```
{
  "DeparturePosition": [-122.7565,49.0021],
  "DestinationPosition": [-122.3394, 47.6159],
  "DepartNow": true,
  "TravelMode": "Truck",
  "TruckModeOptions": {
    "AvoidFerries": true,
    "AvoidTolls": false,
    "Dimensions": {
      "Height": 4.5,
```

```
    "Length": 15.5,  
    "Unit": "Meters",  
    "Width": 4.5  
  },  
  "Weight": {  
    "Total": 7500,  
    "Unit": "Pounds"  
  }  
}  
}
```

## Mengelola sumber daya kalkulator rute Anda

Anda dapat mengelola sumber daya kalkulator rute menggunakan konsol Lokasi Amazon, API Lokasi AmazonAWS CLI, atau Amazon Location.

### Buat daftar sumber daya kalkulator rute Anda

Anda dapat melihat daftar kalkulator rute menggunakan konsol Lokasi Amazon, API Lokasi AmazonAWS CLI, atau Amazon:

#### Console

Untuk melihat daftar kalkulator rute menggunakan konsol Lokasi Amazon

1. Buka konsol Lokasi Amazon di <https://console.aws.amazon.com/location/>.
2. Pilih Kalkulator rute dari panel navigasi kiri.
3. Lihat detail kalkulator rute di bawah Kalkulator rute saya.

#### API

Gunakan [ListRouteCalculators](#) operasi dari Amazon Location Routes API.

Contoh berikut adalah permintaan API untuk mendapatkan daftar kalkulator rute di AWS akun.

```
POST /routes/v0/list-calculators
```

Berikut ini adalah contoh respons untuk [ListRouteCalculators](#):

```
{  
  "Entries": [  
    {  
      "Name": "Example Route",  
      "Type": "Point-to-Point",  
      "Status": "Active",  
      "Length": 15.5,  
      "Width": 4.5,  
      "Weight": {  
        "Total": 7500,  
        "Unit": "Pounds"  
      }  
    }  
  ]  
}
```

```
{
  "CalculatorName": "ExampleCalculator",
  "CreateTime": 2020-09-30T22:59:34.142Z,
  "DataSource": "Esri",
  "Description": "string",
  "UpdateTime": 2020-09-30T23:59:34.142Z
},
"NextToken": "1234-5678-9012"
}
```

## CLI

Gunakan perintah `list-route-calculators`.

Contoh berikut adalah AWS CLI untuk mendapatkan daftar kalkulator rute di AWS akun.

```
aws location list-route-calculators
```

## Dapatkan detail kalkulator rute

Anda bisa mendapatkan detail tentang sumber kalkulator rute apa pun di AWS akun Anda menggunakan konsol Lokasi Amazon, API Lokasi AmazonAWS CLI, atau Amazon Location:

### Console

Untuk melihat detail kalkulator rute menggunakan konsol Lokasi Amazon

1. Buka konsol Lokasi Amazon di <https://console.aws.amazon.com/location/>.
2. Pilih Kalkulator rute dari panel navigasi kiri.
3. Di bawah Kalkulator rute saya, pilih tautan nama kalkulator rute target.

### API

Gunakan [DescribeRouteCalculator](#) operasi dari Amazon Location Routes API.

Contoh berikut adalah permintaan API untuk mendapatkan detail kalkulator rute *ExampleCalculator*.

```
GET /routes/v0/calculators/ExampleCalculator
```



Berikut ini adalah contoh respons untuk [DescribeRouteCalculator](#):

```
{
  "CalculatorArn": "arn:aws:geo:us-west-2:123456789012:route-
calculator/ExampleCalculator",
  "CalculatorName": "ExampleCalculator",
  "CreateTime": "2020-09-30T22:59:34.142Z",
  "DataSource": "Esri",
  "Description": "string",
  "Tags": {
    "Tag1" : "Value1"
  },
  "UpdateTime": "2020-09-30T23:59:34.142Z"
}
```

## CLI

Gunakan perintah `describe-route-calculator`.

Contoh berikut adalah AWS CLI untuk mendapatkan rincian kalkulator rute untuk *ExampleCalculator*.

```
aws location describe-route-calculator \
  --calculator-name "ExampleCalculator"
```

## Hapus kalkulator rute

Anda dapat menghapus kalkulator rute dari AWS akun menggunakan konsol Lokasi Amazon, API Lokasi Amazon, AWS CLI, atau Amazon Location:

### Console

Untuk menghapus kalkulator rute menggunakan konsol Lokasi Amazon

#### Warning

Operasi ini menghapus sumber daya secara permanen.

1. Buka konsol Lokasi Amazon di <https://console.aws.amazon.com/location/>.

2. Pilih Kalkulator rute dari panel navigasi kiri.
3. Di bawah Kalkulator rute saya, pilih kalkulator rute target.
4. Pilih Hapus kalkulator rute.

## API

Gunakan [DeleteRouteCalculator](#) operasi dari Amazon Location Routes API.

Contoh berikut adalah permintaan API untuk menghapus koleksi geofence.

*ExampleCalculator*

```
DELETE /routes/v0/calculators/ExampleCalculator
```

Berikut ini adalah contoh respons untuk [DeleteRouteCalculator](#):

```
HTTP/1.1 200
```

## CLI

Gunakan perintah `delete-route-calculator`.

Contoh berikut adalah AWS CLI perintah untuk menghapus koleksi geofence.

*ExampleCalculator*

```
aws location delete-route-calculator \  
  --calculator-name "ExampleCalculator"
```

## Geofencing area yang diminati menggunakan Lokasi Amazon

Aplikasi geofencing mengevaluasi posisi perangkat yang dilacak relatif terhadap bidang minat yang terdaftar sebelumnya. Ini memungkinkan tindakan diambil berdasarkan pembaruan posisi. Misalnya, Anda dapat memulai acara yang meminta pemberitahuan ketika pelanggan yang memesan kopi di aplikasi seluler mereka berada di dekat toko.

### Note

Untuk ikhtisar konsep geofencing dan pelacak, lihat [Geofences dan Pelacak](#)

Bagian panduan ini memberikan step-by-step petunjuk untuk membuat aplikasi geofencing menggunakan Amazon Location Service.

### Ikhtisar langkah-langkah

1. Tambahkan geofences di sekitar area yang diminati dan simpan di sumber daya koleksi geofence.
2. Mulai lacak perangkat target Anda dan simpan riwayat lokasi perangkat di sumber daya pelacak.
3. Tautkan sumber daya pelacak Anda ke sumber daya pengumpulan geofence Anda sehingga pembaruan lokasi perangkat dievaluasi secara otomatis terhadap semua geofences Anda.
4. Anda dapat mengevaluasi posisi perangkat secara langsung terhadap sumber daya pengumpulan geofence jika Anda tidak ingin menggunakan Pelacak Lokasi Amazon untuk menyimpan riwayat lokasi perangkat Anda.

Setelah Anda menerapkan solusi geofencing Anda, sumber daya pengumpulan geofence Anda memancarkan peristiwa berikut:

- ENTER— Perangkat yang dilacak memasuki geofence dalam koleksi geofence.
- EXIT— Perangkat yang dilacak keluar dari geofence dalam koleksi geofence.

Anda dapat menggunakan Amazon EventBridge untuk bereaksi terhadap peristiwa dengan merutekan mereka di tempat lain.

Sebagai alternatif untuk mengirim pembaruan melalui Amazon Location Service API dari setiap perangkat, Anda dapat menggunakan MQTT untuk mengirim pembaruan perangkat.

Topik berikut menjelaskan langkah-langkah dan alternatif ini secara rinci.

### Topik

- [Tambahkan geofences](#)
- [Mulai melacak](#)
- [Tautkan pelacak ke koleksi geofence](#)
- [Evaluasi posisi perangkat terhadap geofences](#)
- [Verifikasi posisi perangkat](#)
- [Bereaksi terhadap peristiwa Amazon Location Service dengan Amazon EventBridge](#)
- [Melacak penggunaan AWS IoT dan MQTT dengan Amazon Location Service](#)

- [Mengelola sumber daya pengumpulan geofence Anda](#)
- [Mengelola sumber daya pelacak Anda](#)
- [Contoh aplikasi seluler Geofencing dan Tracking](#)

## Tambahkan geofences

Geofences berisi titik dan simpul yang membentuk batas tertutup, yang mendefinisikan area yang menarik. Koleksi Geofence menyimpan dan mengelola satu atau beberapa geofences.

[Koleksi geofence Lokasi Amazon menyimpan geofences yang ditentukan dengan menggunakan format data geospasial standar yang disebut GeoJSON \(RFC 7946\)](#). Anda dapat menggunakan alat, seperti [geojson.io](#), tanpa biaya untuk menggambar geofences Anda secara grafis dan menyimpan file GeoJSON output.

### Note

Lokasi Amazon tidak mendukung poligon dengan lubang, multipoligon, poligon searah jarum jam, dan geofences yang melintasi antimeridian.

## Buat koleksi geofence

Buat koleksi geofence untuk menyimpan dan mengelola geofences menggunakan konsol Lokasi Amazon, API Lokasi Amazon, AWS CLI atau Amazon Location.

### Console

Untuk membuat koleksi geofence menggunakan konsol Lokasi Amazon

1. Buka konsol Amazon Location Service di <https://console.aws.amazon.com/location/>.
2. Di panel navigasi kiri, pilih koleksi Geofence.
3. Pilih Buat koleksi geofence.
4. Isi kotak-kotak berikut:
  - Nama — Masukkan nama unik. Misalnya, *ExampleGeofenceCollection*. Maksimal 100 karakter. Entri yang valid termasuk karakter alfanumerik, tanda hubung, titik, dan garis bawah.
  - Deskripsi — Masukkan deskripsi opsional untuk membedakan sumber daya Anda.

5. Di bawah EventBridge aturan dengan CloudWatch sebagai target, Anda dapat membuat EventBridge aturan opsional untuk mulai [bereaksi terhadap peristiwa geofence](#). Ini memungkinkan Lokasi Amazon untuk mempublikasikan peristiwa ke Amazon CloudWatch Logs.
6. (Opsional) Di bawah Tag, masukkan tag Kunci dan Nilai. Ini menambahkan tag koleksi geofence baru Anda. Untuk informasi selengkapnya, lihat [Menandai sumber daya Amazon Location Service](#).
7. (Opsional) Di bawah enkripsi kunci terkelola Pelanggan, Anda dapat memilih untuk Menambahkan kunci terkelola pelanggan. Ini menambahkan kunci terkelola pelanggan simetris yang Anda buat, miliki, dan kelola melalui enkripsi yang AWS dimiliki default. Untuk informasi selengkapnya, lihat [Mengkripsi data saat istirahat](#).
8. Pilih Buat koleksi geofence.

## API

Untuk membuat koleksi geofence menggunakan Amazon Location API

Gunakan [CreateGeofenceCollection](#) operasi dari Amazon Location Geofences API.

Contoh berikut menggunakan permintaan API untuk membuat koleksi geofence yang disebut *ExampleGeofenceCollection* Pengumpulan geofence dikaitkan dengan [AWS KMS kunci yang dikelola pelanggan untuk mengenkripsi data](#) pelanggan.

```
POST /geofencing/v0/collections
Content-type: application/json

{
  "CollectionName": "ExampleGeofenceCollection",
  "Description": "Geofence collection 1 for shopping center",
  "KmsKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
  "Tags": {
    "Tag1" : "Value1"
  }
}
```

## AWS CLI

Untuk membuat koleksi geofence menggunakan perintah AWS CLI

Gunakan perintah [create-geofence-collection](#).

Contoh berikut menggunakan AWS CLI untuk membuat koleksi geofence yang disebut.

*ExampleGeofenceCollection* Pengumpulan geofence dikaitkan dengan [AWS KMS kunci yang dikelola pelanggan untuk mengenkripsi data](#) pelanggan.

```
aws location \  
  create-geofence-collection \  
  --collection-name "ExampleGeofenceCollection" \  
  --description "Shopping center geofence collection" \  
  --kms-key-id "1234abcd-12ab-34cd-56ef-1234567890ab" \  
  --tags Tag1=Value1
```

### Note

Penagihan tergantung pada penggunaan Anda. Anda mungkin dikenakan biaya untuk penggunaan AWS layanan lain. Untuk informasi selengkapnya, lihat [harga Amazon Location Service](#).

## Menggambar geofences

Sekarang setelah Anda membuat koleksi geofence Anda, Anda dapat menentukan geofences Anda. Geofences didefinisikan baik sebagai poligon atau sebagai lingkaran. [Untuk menggambar geofence poligon Anda dapat menggunakan alat pengeditan GeoJSON, seperti geojson.io.](#)

Untuk membuat geofence sebagai lingkaran, Anda harus menentukan titik tengah lingkaran, dan jari-jarinya. Misalnya, jika Anda ingin membuat geofence untuk diberi tahu setiap kali perangkat berada dalam jarak 50 meter dari lokasi tertentu, Anda akan menggunakan garis lintang dan bujur lokasi itu dan menentukan radius sebagai 50 meter.

Menggunakan Amazon Location Service API, Anda juga dapat menambahkan metadata ke geofence Anda, dalam bentuk pasangan nilai kunci. Ini dapat berguna untuk menyimpan informasi tentang geofence, seperti jenisnya, atau informasi lain yang spesifik untuk aplikasi Anda. Anda dapat menggunakan metadata ini kapan. [Bereaksi terhadap peristiwa Amazon Location Service dengan Amazon EventBridge](#)

## Menambahkan geofences poligon

Bagian ini menjelaskan pembuatan geofences poligon

## Menggambar geofences menggunakan alat GeoJSON

[Sekarang setelah Anda membuat koleksi geofence, Anda dapat menentukan geofences Anda dengan menggunakan alat pengeditan GeoJSON, seperti \[geojson.io\]\(https://geojson.io\).](#)

Untuk membuat file GeoJSON

1. Buka alat pengeditan GeoJSON. Misalnya, [geojson.io](https://geojson.io).
2. Pilih ikon Draw a polygon dan gambar area minat Anda.
3. Pilih Simpan, lalu pilih GeoJSON dari menu tarik-turun.

Letakkan geofences GeoJSON dalam koleksi geofence

Anda dapat menggunakan file GeoJSON yang dihasilkan untuk mengunggah geofences menggunakan konsol AWS CLI Amazon Location Service,, atau Amazon Location API:

### Console

Untuk menambahkan geofence ke koleksi geofence menggunakan konsol Amazon Location Service

1. Buka konsol Amazon Location Service di <https://console.aws.amazon.com/location/>.
2. Di panel navigasi kiri, pilih koleksi Geofence.
3. Dari daftar koleksi Geofence, pilih tautan nama untuk koleksi geofence target.
4. Di bawah Geofences, pilih Buat geofences.
5. Di jendela Add geofences, seret, dan jatuhkan GeoJSON Anda ke jendela.
6. Pilih Tambahkan geofences.

### API

Untuk menambahkan geofences menggunakan Amazon Location API

Gunakan [PutGeofence](#) operasi dari Amazon Location Geofences API.

Contoh berikut menggunakan permintaan API untuk menambahkan geofence yang diberikan ID *GEOFENCE-EXAMPLE1 ke koleksi geofence* yang disebut *ExampleGeofenceCollection* Ini juga menentukan properti metadata geofence tunggal dengan kunci dan nilai. Type `loadingArea`

```
PUT /geofencing/v0/collections/ExampleGeofenceCollection/geofence/GEOFENCE-EXAMPLE1
Content-type: application/json

{
  "GeofenceProperties": {
    "Type" : "loadingArea"
  },
  "Geometry": {
    "Polygon": [
      [
        [-5.716667, -15.933333],
        [-14.416667, -7.933333],
        [-12.316667, -37.066667],
        [-5.716667, -15.933333]
      ]
    ]
  }
}
```

Atau, Anda dapat menambahkan lebih dari satu geofence menggunakan operasi.

### [BatchPutGeofence](#)

```
POST /geofencing/v0/collections/ExampleGeofenceCollection/put-geofences
Content-type: application/json

{
  "Entries": [
    {
      "GeofenceProperties": {
        "Type" : "loadingArea"
      },
      "GeofenceId": "GEOFENCE-EXAMPLE1",
      "Geometry": {
        "Polygon": [
          [
            [-5.716667, -15.933333],
            [-14.416667, -7.933333],
            [-12.316667, -37.066667],
            [-5.716667, -15.933333]
          ]
        ]
      }
    }
  ]
}
```



```
]
}
```

## AWS CLI

Untuk menambahkan geofence ke koleksi geofence menggunakan perintah AWS CLI

Gunakan perintah [put-geofence](#).

Contoh berikut menggunakan AWS CLI untuk menambahkan geofence ke koleksi geofence yang disebut. *ExampleGeofenceCollection*

```
$ aws location \
  put-geofence \
    --collection-name ExampleGeofenceCollection \
    --geofence-id ExampleGeofenceTriangle \
    --geofence-properties '{"Type": "loadingArea"}' \
    --geometry 'Polygon=[[[[-5.716667, -15.933333],[ -14.416667, -7.933333],
[-12.316667, -37.066667],[ -5.716667, -15.933333]]]]'
{
  "CreateTime": "2020-11-11T00:16:14.487000+00:00",
  "GeofenceId": "ExampleGeofenceTriangle",
  "UpdateTime": "2020-11-11T00:19:59.894000+00:00"
}
```

## Menambahkan geofences melingkar

Bagian ini menjelaskan pembuatan geofences melingkar. Anda harus mengetahui garis lintang dan bujur titik yang Anda inginkan menjadi pusat lingkaran, dan jari-jari dalam meter lingkaran. Anda dapat membuat geofences melingkar dengan Amazon Location API atau AWS CLI

### API

Untuk menambahkan geofences melingkar menggunakan Amazon Location API

Gunakan [PutGeofence](#) operasi dari Amazon Location Geofences API.

Contoh berikut menggunakan permintaan API untuk menambahkan geofence yang diberikan ID *GEOFENCE-EXAMPLE2* ke *koleksi geofence* yang disebut: *ExampleGeofenceCollection*

```

PUT /geofencing/v0/collections/ExampleGeofenceCollection/geofence/GEOFENCE-EXAMPLE2
Content-type: application/json

{
  "Geometry": {
    "Circle": {
      "Center": [-5.716667, -15.933333],
      "Radius": 50
    }
  }
}

```

## AWS CLI

Untuk menambahkan geofence melingkar ke koleksi geofence menggunakan perintah AWS CLI

Gunakan perintah [put-geofence](#).

Contoh berikut menggunakan AWS CLI untuk menambahkan geofence ke koleksi geofence yang disebut. *ExampleGeofenceCollection*

```

$ aws location \
  put-geofence \
    --collection-name ExampleGeofenceCollection \
    --geofence-id ExampleGeofenceCircle \
    --geometry 'Circle={Center=[-5.716667, -15.933333], Radius=50}'

```

### Note

Anda juga dapat menempatkan JSON untuk geometri kompleks ke dalam file sendiri seperti pada contoh berikut.

```

$ aws location \
  put-geofence \
    --collection-name ExampleGeofenceCollection \
    --geofence-id ExampleGeofenceCircle \
    --geometry file:circle.json

```

Dalam contoh, file circle.json menyertakan JSON untuk geometri lingkaran.

```
{
```

```
"Circle": {  
  "Center": [-74.006975, 40.717127],  
  "Radius": 287.7897969218057  
}
```

## Mulai melacak

Bagian ini memandu Anda melalui pembuatan aplikasi pelacakan yang menangkap lokasi perangkat.

### Buat pelacak

Buat sumber daya pelacak untuk menyimpan dan memproses pembaruan posisi dari perangkat Anda. Anda dapat menggunakan konsol Amazon Location Service, Amazon Location API AWS CLI, atau Amazon Location.

Setiap pembaruan posisi yang disimpan dalam sumber daya pelacak Anda dapat mencakup ukuran akurasi posisi, dan hingga tiga bidang metadata tentang posisi atau perangkat yang ingin Anda simpan. Metadata disimpan sebagai pasangan nilai kunci, dan dapat menyimpan informasi seperti kecepatan, arah, tekanan ban, atau suhu mesin.

Pelacak memfilter pembaruan posisi saat diterima. Ini mengurangi noise visual di jalur perangkat Anda (disebut jitter), dan mengurangi jumlah peristiwa masuk dan keluar geofence palsu. Ini juga membantu mengelola biaya dengan mengurangi jumlah evaluasi geofence yang dimulai.

Pelacak menawarkan tiga opsi pemfilteran posisi untuk membantu mengelola biaya dan mengurangi jitter di pembaruan lokasi Anda.

- **Berbasis akurasi** - Gunakan dengan perangkat apa pun yang memberikan pengukuran akurasi. Sebagian besar perangkat seluler memberikan informasi ini. Keakuratan setiap pengukuran posisi dipengaruhi oleh banyak faktor lingkungan, termasuk penerimaan satelit GPS, lanskap, dan kedekatan perangkat Wi-Fi dan Bluetooth. Sebagian besar perangkat, termasuk sebagian besar perangkat seluler, dapat memberikan perkiraan keakuratan pengukuran bersama dengan pengukuran. Dengan AccuracyBased pemfilteran, Lokasi Amazon mengabaikan pembaruan lokasi jika perangkat bergerak kurang dari akurasi yang diukur. Misalnya, jika dua pembaruan berturut-turut dari perangkat memiliki rentang akurasi 5 m dan 10 m, Amazon Location mengabaikan pembaruan kedua jika perangkat telah bergerak kurang dari 15 m. Lokasi Amazon tidak mengevaluasi pembaruan yang diabaikan terhadap geofences, atau menyimpannya.

Ketika akurasi tidak disediakan, itu diperlakukan sebagai nol, dan pengukuran dianggap sangat akurat.

#### Note

Anda juga dapat menggunakan pemfilteran berbasis akurasi untuk menghapus semua pemfilteran. Jika Anda memilih pemfilteran berbasis akurasi, tetapi mengganti semua data akurasi ke nol, atau menghilangkan akurasi sepenuhnya, maka Lokasi Amazon tidak akan memfilter pembaruan apa pun.

- **Berbasis jarak** — Gunakan saat perangkat Anda tidak memberikan pengukuran akurasi, tetapi Anda tetap ingin memanfaatkan penyaringan untuk mengurangi jitter dan mengelola biaya. `DistanceBased` pemfilteran mengabaikan pembaruan lokasi di mana perangkat telah bergerak kurang dari 30 m (98,4 kaki). Saat Anda menggunakan pemfilteran `DistanceBased` posisi, Amazon Location tidak mengevaluasi pembaruan yang diabaikan ini terhadap geofences atau menyimpan pembaruan.

Keakuratan sebagian besar perangkat seluler, termasuk akurasi rata-rata perangkat iOS dan Android, berada dalam jarak 15 m. Di sebagian besar aplikasi, `DistanceBased` penyaringan dapat mengurangi efek ketidakakuratan lokasi saat menampilkan lintasan perangkat pada peta, dan efek pantulan dari beberapa peristiwa masuk dan keluar berturut-turut saat perangkat berada di dekat perbatasan geofence. Hal ini juga dapat membantu mengurangi biaya aplikasi Anda, dengan membuat lebih sedikit panggilan untuk mengevaluasi terhadap geofences terkait atau mengambil posisi perangkat.

- **Berbasis waktu** — (default) Gunakan saat perangkat Anda mengirim pembaruan posisi sangat sering (lebih dari sekali setiap 30 detik), dan Anda ingin mencapai evaluasi geofence mendekati waktu nyata tanpa menyimpan setiap pembaruan. Dalam `TimeBased` pemfilteran, setiap pembaruan lokasi dievaluasi terhadap koleksi geofence yang ditautkan, tetapi tidak setiap pembaruan lokasi disimpan. Jika frekuensi pembaruan Anda lebih sering dari 30 detik, hanya satu pembaruan per 30 detik yang disimpan untuk setiap ID perangkat unik.

#### Note

Perhatikan biaya aplikasi pelacakan Anda saat memutuskan metode penyaringan Anda dan frekuensi pembaruan posisi. Anda ditagih untuk setiap pembaruan lokasi dan sekali untuk mengevaluasi pembaruan posisi terhadap setiap koleksi geofence yang ditautkan. Misalnya,

saat menggunakan pemfilteran berbasis waktu, jika pelacak Anda ditautkan ke dua koleksi geofence, setiap pembaruan posisi akan dihitung sebagai satu permintaan pembaruan lokasi dan dua evaluasi pengumpulan geofence. Jika Anda melaporkan pembaruan posisi setiap 5 detik untuk perangkat Anda dan menggunakan pemfilteran berbasis waktu, Anda akan ditagih untuk 720 pembaruan lokasi dan 1.440 evaluasi geofensi per jam untuk setiap perangkat. Tagihan Anda tidak terpengaruh oleh jumlah geofences di setiap koleksi. Karena setiap koleksi geofence dapat berisi hingga 50,000 geofences, Anda mungkin ingin menggabungkan geofences Anda menjadi koleksi yang lebih sedikit, jika memungkinkan, untuk mengurangi biaya evaluasi geofence Anda.

Secara default, Anda akan mendapatkan EventBridge acara setiap kali perangkat yang dilacak masuk atau keluar dari geofence yang ditautkan. Untuk informasi selengkapnya, lihat [Tautkan pelacak ke koleksi geofence](#).

Anda dapat mengaktifkan acara untuk semua pembaruan posisi yang difilter untuk sumber daya pelacak. Untuk informasi selengkapnya, lihat [Aktifkan peristiwa pembaruan untuk pelacak](#).

#### Note


Jika Anda ingin mengenkripsi data menggunakan kunci yang dikelola AWS KMS pelanggan Anda sendiri, maka fitur Bounding Polygon Queries akan dinonaktifkan secara default. Ini karena dengan menggunakan fitur Bounding Polygon Queries ini, representasi posisi perangkat Anda tidak akan dienkripsi menggunakan kunci terkelola Anda. AWS KMS Namun, posisi perangkat yang tepat masih dienkripsi menggunakan kunci terkelola Anda. Anda dapat memilih untuk ikut serta ke fitur Bounding Polygon Query dengan menyetel `KmsKeyEnableGeospatialQueries` parameter ke `true` saat membuat atau memperbarui Tracker.

## Console

Untuk membuat pelacak menggunakan konsol Lokasi Amazon

1. Buka konsol Amazon Location Service di <https://console.aws.amazon.com/location/>.
2. Di panel navigasi kiri, pilih Pelacak.
3. Pilih Buat pelacak.
4. Isi kolom berikut:

- Nama — Masukkan nama unik. Misalnya, *ExampleTracker*. Maksimal 100 karakter. Entri yang valid termasuk karakter alfanumerik, tanda hubung, titik, dan garis bawah.
  - Deskripsi - Masukkan deskripsi opsional.
5. Di bawah Pemfilteran posisi, pilih opsi yang paling sesuai dengan cara Anda ingin menggunakan sumber daya pelacak Anda. Jika Anda tidak mengatur Pemfilteran posisi, pengaturan defaultnya adalah `TimeBased`. Untuk informasi selengkapnya, lihat [Pelacak](#) di panduan ini, dan [PositionFiltering](#) di Referensi API Amazon Location Service Trackers.
  6. (Opsional) Di bawah Tag, masukkan tag Kunci dan Nilai. Ini menambahkan tag koleksi geofence baru Anda. Untuk informasi selengkapnya, lihat [Menandai sumber daya Anda](#).
  7. (Opsional) Di bawah enkripsi kunci terkelola Pelanggan, Anda dapat memilih untuk Menambahkan kunci terkelola pelanggan. Ini menambahkan kunci terkelola pelanggan simetris yang Anda buat, miliki, dan kelola melalui enkripsi yang AWS dimiliki default. Untuk informasi selengkapnya, lihat [Mengenkripsi data saat istirahat](#).
  8. (Opsional) Di bawah `KmsKeyEnableGeospatialQueries`, Anda dapat memilih untuk mengaktifkan Kueri Geospasial. Ini memungkinkan Anda menggunakan fitur Bounding Polygon Queries, sambil mengenkripsi data Anda menggunakan kunci terkelola AWS KMS pelanggan.

 Note

Saat Anda menggunakan fitur Bounding Polygon Query, representasi posisi perangkat Anda tidak akan dienkripsi menggunakan kunci terkelola. AWS KMS Namun, posisi perangkat yang tepat masih dienkripsi menggunakan kunci terkelola Anda.

9. (Opsional) Di bawah EventBridge konfigurasi, Anda dapat memilih untuk mengaktifkan EventBridge acara untuk pembaruan posisi yang difilter. Ini akan mengirimkan acara setiap kali pembaruan posisi untuk perangkat di pelacak ini memenuhi evaluasi penyaringan posisi.
10. Pilih Buat pelacak.

## API

Untuk membuat pelacak dengan menggunakan Amazon Location API

Gunakan [CreateTracker](#) operasi dari Amazon Location Trackers API.

Contoh berikut menggunakan permintaan API untuk membuat pelacak yang disebut *ExampleTracker*. Sumber daya pelacak dikaitkan dengan [AWS KMS kunci yang dikelola pelanggan untuk mengenkripsi data pelanggan](#), dan tidak [mengaktifkan pembaruan posisi](#).

### EventBridge

```
POST /tracking/v0/trackers
Content-type: application/json

{
  "TrackerName": "ExampleTracker",
  "Description": "string",
  "KmsKeyEnableGeospatialQueries": false,
  "EventBridgeEnabled": false,
  "KmsKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
  "PositionFiltering": "AccuracyBased",
  "Tags": {
    "string" : "string"
  }
}
```

### Buat pelacak dengan diaktifkan **KmsKeyEnableGeospatialQueries**

Contoh berikut memiliki parameter `KmsKeyEnableGeospatialQueries` diatur ke `true`. Ini memungkinkan Anda menggunakan fitur Bounding Polygon Queries, sambil mengenkripsi data menggunakan kunci yang dikelola pelanggan. AWS KMS

Untuk informasi tentang penggunaan fitur Bounding Polygon Query, lihat [???](#)

#### Note

Saat Anda menggunakan fitur Bounding Polygon Query, representasi posisi perangkat Anda tidak akan dienkripsi menggunakan kunci terkelola. AWS KMS Namun, posisi perangkat yang tepat masih dienkripsi menggunakan kunci terkelola Anda.

```
POST /tracking/v0/trackers
Content-type: application/json

{
```

```
"TrackerName": "ExampleTracker",
"Description": "string",
"KmsKeyEnableGeospatialQueries": true,
"EventBridgeEnabled": false,
"KmsKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
"PositionFiltering": "AccuracyBased",
"Tags": {
  "string" : "string"
}
}
```

## AWS CLI

Untuk membuat pelacak menggunakan perintah AWS CLI

Gunakan perintah [create-tracker](#).

Contoh berikut menggunakan AWS CLI untuk membuat pelacak yang disebut *ExampleTracker*. Sumber daya pelacak dikaitkan dengan [AWS KMS kunci yang dikelola pelanggan untuk mengenkripsi data pelanggan](#), dan tidak [mengaktifkan pembaruan posisi](#). EventBridge

```
aws location \
  create-tracker \
  --tracker-name "ExampleTracker" \
  --position-filtering "AccuracyBased" \
  --event-bridge-enabled false \
  --kms-key-enable-geospatial-queries false \
  --kms-key-id "1234abcd-12ab-34cd-56ef-1234567890ab"
```

Buat pelacak dengan diaktifkan **KmsKeyEnableGeospatialQueries**

Contoh berikut memiliki parameter `KmsKeyEnableGeospatialQueries` diatur ke `true`. Ini memungkinkan Anda menggunakan fitur Bounding Polygon Queries, sambil mengenkripsi data menggunakan kunci yang dikelola pelanggan. AWS KMS

Untuk informasi tentang penggunaan fitur Bounding Polygon Query, lihat [???](#)

### Note

Saat Anda menggunakan fitur Bounding Polygon Query, representasi posisi perangkat Anda tidak akan dienkripsi menggunakan kunci dikelola. AWS KMS Namun, posisi perangkat yang tepat masih dienkripsi menggunakan kunci dikelola Anda.



```
aws location \  
  create-tracker \  
  --tracker-name "ExampleTracker" \  
  --position-filtering "AccuracyBased" \  
  --event-bridge-enabled false \  
  --kms-key-enable-geospatial-queries true \  
  --kms-key-id "1234abcd-12ab-34cd-56ef-1234567890ab"
```

### Note

Penagihan tergantung pada penggunaan Anda. Anda mungkin dikenakan biaya untuk penggunaan AWS layanan lain. Untuk informasi selengkapnya, lihat [harga Amazon Location Service](#).

Anda dapat mengedit Deskripsi, Pemfilteran posisi, dan EventBridge konfigurasi setelah pelacak dibuat dengan memilih Edit tracker.

## Mengautentikasi permintaan Anda

Setelah Anda membuat sumber daya pelacak dan Anda siap untuk mulai mengevaluasi posisi perangkat terhadap geofences, pilih bagaimana Anda akan mengautentikasi permintaan Anda:

- Untuk mempelajari cara mengakses layanan, lihat [Mengakses Amazon Location Service](#).
- Jika Anda ingin mempublikasikan posisi perangkat dengan permintaan yang tidak diautentikasi, Anda mungkin ingin menggunakan Amazon Cognito.

### Contoh

Contoh berikut menunjukkan penggunaan kumpulan identitas Amazon Cognito untuk otorisasi, menggunakan [AWS JavaScript SDK v3](#), dan Lokasi Amazon. [JavaScript Pembantu otentikasi](#)

```
import { LocationClient, BatchUpdateDevicePositionCommand } from "@aws-sdk/client-location";  
import { withIdentityPoolId } from "@aws/amazon-location-utilities-auth-helper";  
  
// Unauthenticated identity pool you created  
const identityPoolId = "us-east-1:1234abcd-5678-9012-abcd-sample-id";
```

```
// Create an authentication helper instance using credentials from Cognito
const authHelper = await withIdentityPoolId(identityPoolId);

const client = new LocationClient({
  region: "us-east-1", // The region containing both the identity pool and tracker
  resource
  ...authHelper.getLocationClientConfig(), // Provides configuration required to make
  requests to Amazon Location
});

const input = {
  TrackerName: "ExampleTracker",
  Updates: [
    {
      DeviceId: "ExampleDevice-1",
      Position: [-123.4567, 45.6789],
      SampleTime: new Date("2020-10-02T19:09:07.327Z"),
    },
    {
      DeviceId: "ExampleDevice-2",
      Position: [-123.123, 45.123],
      SampleTime: new Date("2020-10-02T19:10:32Z"),
    },
  ],
};

const command = new BatchUpdateDevicePositionCommand(input);

// Send device position updates
const response = await client.send(command);
```

## Perbarui pelacak Anda dengan posisi perangkat

Untuk melacak perangkat Anda, Anda dapat memposting pembaruan posisi perangkat ke pelacak Anda. Anda kemudian dapat mengambil posisi perangkat ini atau riwayat posisi perangkat dari sumber daya pelacak Anda.

Setiap pembaruan posisi harus menyertakan ID perangkat, stempel waktu, dan posisi. Anda dapat secara opsional menyertakan metadata lain, termasuk akurasi dan hingga 3 pasangan nilai kunci untuk Anda gunakan sendiri.

Jika pelacak Anda ditautkan ke satu atau beberapa koleksi geofence, pembaruan akan dievaluasi terhadap geofences tersebut (mengikuti aturan pemfilteran yang Anda tentukan untuk pelacak). Jika perangkat melanggar area geofenced (dengan berpindah dari dalam area ke luar, atau sebaliknya), Anda akan menerima acara di. EventBridge Ini ENTER atau EXIT peristiwa mencakup detail pembaruan posisi, termasuk ID perangkat, stempel waktu, dan metadata terkait apa pun.

### Note

Untuk informasi selengkapnya tentang pemfilteran posisi, lihat [Buat pelacak](#).

Untuk informasi selengkapnya tentang peristiwa geofence, lihat. [Bereaksi terhadap peristiwa Amazon Location Service dengan Amazon EventBridge](#)

Gunakan salah satu metode ini untuk mengirim pembaruan perangkat:

- [Kirim pembaruan MQTT](#) ke sumber daya AWS IoT Core dan tautkan ke sumber daya pelacak Anda.
- Kirim pembaruan lokasi menggunakan Amazon Location Trackers API, dengan menggunakan AWS CLI, atau Amazon Location API. Anda dapat menggunakan [AWS SDK](#) untuk memanggil API dari aplikasi iOS atau Android Anda.

## API

Untuk mengirim pembaruan posisi menggunakan Amazon Location API

Gunakan [BatchUpdateDevicePosition](#) operasi dari Amazon Location Trackers API.

Contoh berikut menggunakan permintaan API untuk memposting pembaruan posisi perangkat *ExampleDevice* ke pelacak *ExampleTracker*.

```
POST /tracking/v0/trackers/ExampleTracker/positions
Content-type: application/json
{
  "Updates": [
    {
      "DeviceId": "1",
      "Position": [
        -123.12245146162303, 49.27521118043802
      ],
      "SampleTime": "2022-10-24T19:09:07.327Z",
```

```
"PositionProperties": {
  "name" : "device1"
},
"Accuracy": {
  "Horizontal": 10
}
},

{
  "DeviceId": "2",
  "Position": [
-123.1230104928471, 49.27752402723152
  ],
  "SampleTime": "2022-10-02T19:09:07.327Z"
},
{
  "DeviceId": "3",
  "Position": [
-123.12325592118916, 49.27340530543111
  ],
  "SampleTime": "2022-10-02T19:09:07.327Z"
},
{
  "DeviceId": "4",
  "Position": [
-123.11958813096311, 49.27774641063121
  ],
  "SampleTime": "2022-10-02T19:09:07.327Z"
},
{
  "DeviceId": "5",
  "Position": [
-123.1277418058896, 49.2765989015285
  ],
  "SampleTime": "2022-10-02T19:09:07.327Z"
},
{
  "DeviceId": "6",
  "Position": [
-123.11964267059481, 49.274188155916534
  ],
  "SampleTime": "2022-10-02T19:09:07.327Z"
}
]
```

```
}
```

## AWS CLI

Untuk mengirim pembaruan posisi menggunakan AWS CLI perintah

Gunakan perintah [batch-update-device-position](#).

Contoh berikut menggunakan AWS CLI untuk memposting pembaruan posisi perangkat untuk *ExampleDevice-1* dan *ExampleDevice-2* ke pelacak *ExampleTracker*.

```
aws location batch-update-device-position \  
--tracker-name ExampleTracker \  
--updates '[{"DeviceId":"ExampleDevice-1","Position":  
[-123.123,47.123],"SampleTime":"2021-11-30T21:47:25.149Z"},  
{"DeviceId":"ExampleDevice-2","Position":  
[-123.123,47.123],"SampleTime":"2021-11-30T21:47:25.149Z","Accuracy":  
{"Horizontal":10.30},"PositionProperties":{"field1":"value1","field2":"value2"}}]'
```

## Dapatkan riwayat lokasi perangkat dari pelacak

Sumber daya pelacak Lokasi Amazon Anda menyimpan riwayat lokasi semua perangkat yang dilacak selama 30 hari. Anda dapat mengambil riwayat lokasi perangkat, termasuk semua metadata terkait, dari sumber daya pelacak Anda. Contoh berikut menggunakan AWS CLI, atau Amazon Location API.

### API

Untuk mendapatkan riwayat lokasi perangkat dari pelacak menggunakan Amazon Location API

Gunakan [GetDevicePositionHistory](#) operasi dari Amazon Location Trackers API.

Contoh berikut menggunakan permintaan URI API untuk mendapatkan riwayat lokasi perangkat dari pelacak yang disebut *ExampleTracker* mulai *ExampleDevice* dari 19:05:07 (inklusif) dan berakhir pada 19:20:07 (eksklusif) aktif. 2020-10-02

```
POST /tracking/v0/trackers/ExampleTracker/devices/ExampleDevice/list-positions  
Content-type: application/json  
{  
  "StartTimeInclusive": "2020-10-02T19:05:07.327Z",  
  "EndTimeExclusive": "2020-10-02T19:20:07.327Z"  
}
```

## AWS CLI

Untuk mendapatkan riwayat lokasi perangkat dari pelacak menggunakan perintah AWS CLI

Gunakan perintah [get-device-position-history](#).

Contoh berikut menggunakan AWS CLI untuk mendapatkan riwayat lokasi perangkat dari pelacak yang disebut *ExampleTracker* mulai *ExampleDevice* dari 19:05:07 (inklusif) dan berakhir pada 19:20:07 (eksklusif) aktif. 2020-10-02

```
aws location \  
  get-device-position-history \  
    --device-id "ExampleDevice" \  
    --start-time-inclusive "2020-10-02T19:05:07.327Z" \  
    --end-time-exclusive "2020-10-02T19:20:07.327Z" \  
    --tracker-name "ExampleTracker"
```

## Buat daftar posisi perangkat Anda

Anda dapat melihat daftar posisi perangkat untuk pelacak menggunakan AWS CLI, atau Amazon Location API, dengan ListDevicePositions API. Saat Anda memanggil ListDevicePositions API, daftar posisi terbaru untuk semua perangkat yang terkait dengan pelacak tertentu akan ditampilkan. Secara default API ini mengembalikan 100 posisi perangkat terbaru per halaman hasil untuk pelacak tertentu. Untuk hanya mengembalikan perangkat dalam wilayah tertentu, gunakan FilterGeometry parameter untuk membuat Kueri Poligon Bounding. Dengan cara ini saat Anda menelepon ListDevicePositions, hanya perangkat di dalam poligon yang akan dikembalikan.

### Note

Jika Anda ingin mengenkripsi data menggunakan kunci yang dikelola AWS KMS pelanggan Anda sendiri, maka fitur Bounding Polygon Queries akan dinonaktifkan secara default. Ini karena dengan menggunakan fitur ini, representasi posisi perangkat Anda tidak akan dienkripsi menggunakan kunci AWS KMS terkelola Anda. Posisi perangkat yang tepat, bagaimanapun; masih dienkripsi menggunakan kunci terkelola Anda. Anda dapat memilih untuk ikut serta dalam fitur Bounding Polygon Queries. Ini dilakukan dengan mengatur KmsKeyEnableGeospatialQueries parameter ke true saat membuat atau memperbarui Tracker.

## API

Gunakan [ListDevicePositions](#) operasi dari Amazon Location Trackers API.

Contoh berikut adalah permintaan API untuk mendapatkan daftar posisi perangkat di area poligonal, menggunakan parameter opsional. [FilterGeometry](#) Contoh mengembalikan 3 lokasi perangkat hadir di area yang ditentukan oleh Polygon array.

```
POST /tracking/v0/trackers/TrackerName/list-positions HTTP/1.1
Content-type: application/json
```

```
{
  "FilterGeometry": {
    "Polygon": [
      [
        [
          -123.12003339442259,
          49.27425121147397
        ],
        [
          -123.1176984148229,
          49.277063620879744
        ],
        [
          -123.12389509145294,
          49.277954183760926
        ],
        [
          -123.12755921328647,
          49.27554025235713
        ],
        [
          -123.12330236586217,
          49.27211836076236
        ],
        [
          -123.12003339442259,
          49.27425121147397
        ]
      ]
    ]
  },
  "MaxResults": 3,
```

```
"NextToken": "1234-5678-9012"  
}
```

Berikut ini adalah contoh respons untuk [ListDevicePositions](#):

```
{  
  "Entries": [  
    {  
      "DeviceId": "1",  
      "SampleTime": "2022-10-24T19:09:07.327Z",  
      "Position": [  
        -123.12245146162303,  
        49.27521118043802  
      ],  
      "Accuracy": {  
        "Horizontal": 10  
      },  
      "PositionProperties": {  
        "name": "device1"  
      }  
    },  
    {  
      "DeviceId": "3",  
      "SampleTime": "2022-10-02T19:09:07.327Z",  
      "Position": [  
        -123.12325592118916,  
        49.27340530543111  
      ]  
    },  
    {  
      "DeviceId": "2",  
      "SampleTime": "2022-10-02T19:09:07.327Z",  
      "Position": [  
        -123.1230104928471,  
        49.27752402723152  
      ]  
    }  
  ],  
  "NextToken": "1234-5678-9012"  
}
```



## CLI

Gunakan perintah [list-trackers](#).

Contoh berikut adalah AWS CLI untuk mendapatkan daftar perangkat di area poligonal.

```
aws location list-device-positions TODO: add arguments add props for filter geo
```

## Tautkan pelacak ke koleksi geofence

Sekarang setelah Anda memiliki koleksi geofence dan pelacak, Anda dapat menautkannya bersama-sama sehingga pembaruan lokasi secara otomatis dievaluasi terhadap semua geofences Anda. Jika Anda tidak ingin mengevaluasi semua pembaruan lokasi, atau sebagai alternatif, jika Anda tidak menyimpan beberapa lokasi di sumber daya pelacak, Anda dapat [mengevaluasi posisi perangkat terhadap geofences](#) sesuai permintaan.

Ketika posisi perangkat dievaluasi terhadap geofences, peristiwa dihasilkan. Anda dapat mengatur tindakan untuk acara ini. Untuk informasi selengkapnya tentang tindakan yang dapat Anda setel untuk peristiwa geofence, lihat [Bereaksi terhadap peristiwa Amazon Location Service dengan Amazon EventBridge](#)

Peristiwa Lokasi Amazon menyertakan atribut pembaruan posisi perangkat yang menghasilkannya dan beberapa atribut geofence yang dimasukkan atau keluar. Untuk informasi selengkapnya tentang data yang disertakan dalam peristiwa geofence, lihat [Contoh EventBridge acara Amazon untuk Amazon Location Service](#)

Contoh berikut menautkan sumber daya pelacak ke koleksi geofence menggunakan konsol, API Lokasi Amazon AWS CLI, atau Amazon.

### Console

Untuk menautkan sumber daya pelacak ke koleksi geofence menggunakan konsol Amazon Location Service

1. Buka konsol Amazon Location Service di <https://console.aws.amazon.com/location/>.
2. Di panel navigasi kiri, pilih Pelacak.
3. Di bawah Pelacak perangkat, pilih tautan nama pelacak target.
4. Di bawah Koleksi Geofence Tertaut, pilih Link Geofence Collection.

5. Di jendela Linked Geofence Collection, pilih koleksi geofence dari menu tarik-turun.
6. Pilih Tautkan.

Setelah Anda menautkan sumber daya pelacak, itu akan diberi status Aktif.

## API

Untuk menautkan sumber daya pelacak ke koleksi geofence menggunakan Amazon Location API

Gunakan [AssociateTrackerConsumer](#) operasi dari Amazon Location Trackers API.

Contoh berikut menggunakan permintaan API yang terkait *ExampleTracker* dengan koleksi geofence menggunakan [Amazon Resource Name \(ARN\)](#).

```
POST /tracking/v0/trackers/ExampleTracker/consumers
Content-type: application/json

{
  "ConsumerArn": "arn:aws:geo:us-west-2:123456789012:geofence-
collection/ExampleGeofenceCollection"
}
```

## AWS CLI

Untuk menautkan sumber daya pelacak ke koleksi geofence menggunakan perintah AWS CLI

Gunakan perintah [associate-tracker-consumer](#).

Contoh berikut menggunakan AWS CLI untuk membuat koleksi geofence yang disebut *ExampleGeofenceCollection*

```
aws location \
  associate-tracker-consumer \
    --consumer-arn "arn:aws:geo:us-west-2:123456789012:geofence-
collection/ExampleGeofenceCollection" \
    --tracker-name "ExampleTracker"
```

## Evaluasi posisi perangkat terhadap geofences

Ada dua cara untuk mengevaluasi posisi terhadap geofences untuk menghasilkan peristiwa geofence:

- Anda dapat menautkan Pelacak dan Koleksi Geofence. Untuk informasi lebih lanjut, lihat bagian: [Tautkan pelacak ke koleksi geofence](#).
- Anda dapat membuat permintaan langsung ke sumber daya pengumpulan geofence untuk mengevaluasi satu atau beberapa posisi, menggunakan API. [BatchEvaluateGeofences](#)

Selain itu, Anda dapat memperkirakan peristiwa geofence yang masuk untuk perangkat yang masuk, keluar, atau tetap menganggur dalam geofence. Gunakan [ForecastGeofenceEvents](#) API untuk meramalkan peristiwa.

Jika Anda juga ingin melacak riwayat lokasi perangkat atau menampilkan lokasi di peta, tautkan pelacak dengan koleksi geofence. Atau, Anda mungkin tidak ingin mengevaluasi semua pembaruan lokasi, atau Anda tidak bermaksud menyimpan data lokasi di sumber daya pelacak. Jika salah satu dari ini terjadi, Anda dapat membuat permintaan langsung ke koleksi geofence dan mengevaluasi satu atau lebih posisi perangkat terhadap geofences nya.

Mengevaluasi posisi perangkat terhadap geofences menghasilkan peristiwa. Anda dapat bereaksi terhadap peristiwa ini dan mengarahkannya ke AWS layanan lain. Untuk informasi selengkapnya tentang tindakan yang dapat Anda lakukan saat menerima peristiwa geofence, lihat [Bereaksi terhadap peristiwa Amazon Location Service dengan Amazon EventBridge](#)

Peristiwa Lokasi Amazon menyertakan atribut pembaruan posisi perangkat yang menghasilkannya, termasuk metadata waktu, posisi, akurasi, dan nilai kunci, serta beberapa atribut geofence yang dimasukkan atau keluar. Untuk informasi selengkapnya tentang data yang disertakan dalam peristiwa geofence, lihat. [Contoh EventBridge acara Amazon untuk Amazon Location Service](#)

Contoh berikut menggunakan AWS CLI, atau Amazon Location API.

## API

Untuk mengevaluasi posisi perangkat terhadap posisi geofences menggunakan Amazon Location API

Gunakan [BatchEvaluateGeofences](#) operasi dari Amazon Location Geofences API.

Contoh berikut menggunakan permintaan API untuk mengevaluasi posisi perangkat *ExampleDevice* ke koleksi geofence terkait. *ExampleGeofenceCollection* Ganti nilai-nilai ini dengan geofence dan ID perangkat Anda sendiri.

```
POST /geofencing/v0/collections/ExampleGeofenceCollection/positions HTTP/1.1
```

```
Content-type: application/json

{
  "DevicePositionUpdates": [
    {
      "DeviceId": "ExampleDevice",
      "Position": [-123.123, 47.123],
      "SampleTime": "2021-11-30T21:47:25.149Z",
      "Accuracy": {
        "Horizontal": 10.30
      },
      "PositionProperties": {
        "field1": "value1",
        "field2": "value2"
      }
    }
  ]
}
```

## AWS CLI

Untuk mengevaluasi posisi perangkat terhadap posisi geofences menggunakan perintah AWS CLI

Gunakan perintah [batch-evaluate-geofences](#).

Contoh berikut menggunakan AWS CLI untuk mengevaluasi posisi *ExampleDevice* terhadap koleksi geofence terkait. *ExampleGeofenceCollection* Ganti nilai-nilai ini dengan geofence dan ID perangkat Anda sendiri.

```
aws location \
  batch-evaluate-geofences \
    --collection-name ExampleGeofenceCollection \
    --device-position-updates '[{"DeviceId":"ExampleDevice","Position":
[-123.123,47.123],"SampleTime":"2021-11-30T21:47:25.149Z","Accuracy":
{"Horizontal":10.30},"PositionProperties":{"field1":"value1","field2":"value2"}}]'
```

Mengevaluasi posisi perangkat terhadap geofences menghasilkan peristiwa. Secara tradisional Anda dapat bereaksi terhadap peristiwa dengan menggunakan [Amazon EventBridge](#), tetapi proses ini hanya memungkinkan Anda bereaksi terhadap peristiwa setelah itu terjadi. Jika Anda perlu mengantisipasi kapan perangkat masuk atau keluar dari geofence, misalnya jika perangkat melintasi perbatasan dan akan tunduk pada peraturan yang berbeda sebagai konsekuensinya, maka Anda

dapat menggunakan API [ForecastGeofenceEvents](#) untuk memprediksi peristiwa geofence masa depan.

[ForecastGeofenceEvents](#) API menggunakan kriteria seperti perangkat time-to-breach, kedekatan, kecepatan, dan posisi untuk memprediksi peristiwa. Di sana API akan mengembalikan a `ForecastedBreachTime`, yang menandakan perkiraan waktu peristiwa geofence akan terjadi.

Contoh berikut menggunakan Amazon Location API.

## API

Untuk memperkirakan peristiwa geofence menggunakan Amazon Location API

Gunakan [ForecastGeofenceEvents](#) operasi dari Amazon Location Geofences API.

Contoh berikut menggunakan permintaan API untuk memperkirakan peristiwa geofence untuk *ExampleDevice* relatif terhadap *ExampleGeofence*. Ganti nilai-nilai ini dengan geofence dan ID perangkat Anda sendiri.

```
POST /geofencing/v0/collections/CollectionName/forecast-geofence-events HTTP/1.1
Content-type: application/json

{
  "DeviceState": {
    "Position": [ number ],
    "Speed": number
  },
  "DistanceUnit": "string",
  "MaxResults": number,
  "NextToken": "string",
  "SpeedUnit": "string",
  "TimeHorizonMinutes": number
}
```

## Verifikasi posisi perangkat

Untuk memeriksa integritas posisi perangkat, gunakan [VerifyDevicePosition](#) API. API ini menampilkan informasi tentang integritas posisi perangkat, dengan mengevaluasi properti seperti sinyal sel perangkat, titik akses Wi-Fi, alamat Ipv4, dan jika proxy sedang digunakan.

## Prasyarat

Sebelum dapat menggunakan API yang terdaftar untuk verifikasi perangkat, pastikan Anda memiliki prasyarat berikut:

- Anda telah membuat pelacak untuk perangkat atau perangkat yang ingin Anda periksa. Untuk informasi selengkapnya, lihat [Mulai melacak](#).

Contoh berikut menunjukkan permintaan untuk Amazon Location [VerifyDevicePosition](#) API.

### API

Untuk memverifikasi posisi perangkat menggunakan Amazon Location API

Gunakan [VerifyDevicePosition](#) operasi dari Amazon Location Tracking API.

Contoh berikut menunjukkan permintaan API untuk mengevaluasi integritas posisi perangkat. Ganti nilai-nilai ini dengan ID perangkat Anda sendiri.

```
POST /tracking/v0/trackers/TrackerName/positions/verify HTTP/1.1
Content-type: application/json

{
  "DeviceState": {
    "Accuracy": {
      "Horizontal": number
    },
    "CellSignals": {
      "LteCellDetails": [
        {
          "CellId": number,
          "LocalId": {
            "Earfcn": number,
            "Pci": number
          },
          "Mcc": number,
          "Mnc": number,
          "NetworkMeasurements": [
            {
              "CellId": number,
              "Earfcn": number,
              "Pci": number,
              "Rsrp": number,
```

```
        "Rsrq": number
      }
    ],
    "NrCapable": boolean,
    "Rsrp": number,
    "Rsrq": number,
    "Tac": number,
    "TimingAdvance": number
  }
]
},
"DeviceId": "ExampleDevice",
"Ipv4Address": "string",
"Position": [ number ],
"SampleTime": "string",
"WiFiAccessPoints": [
  {
    "MacAddress": "string",
    "Rss": number
  }
]
},
"DistanceUnit": "string"
}
```

### Note

Location Integrity SDK menyediakan fitur yang disempurnakan terkait dengan verifikasi perangkat, dan tersedia untuk digunakan berdasarkan permintaan. Untuk mendapatkan akses ke SDK, hubungi [Sales Support](#).

## Bereaksi terhadap peristiwa Amazon Location Service dengan Amazon EventBridge

Amazon EventBridge adalah bus acara tanpa server yang secara efisien menghubungkan aplikasi bersama-sama menggunakan data dari AWS layanan seperti Amazon Location. EventBridge menerima peristiwa dari Lokasi Amazon dan merutekan data tersebut ke target seperti AWS Lambda. Anda dapat mengatur aturan perutean untuk menentukan ke mana harus mengirim data Anda untuk membangun arsitektur aplikasi yang bereaksi secara real time.

Hanya peristiwa geofence (ENTER dan EXIT peristiwa, saat perangkat memasuki atau meninggalkan area geofensi) yang dikirim secara default. EventBridge Anda juga dapat mengaktifkan semua peristiwa pembaruan posisi yang difilter untuk sumber daya pelacak. Untuk informasi selengkapnya, lihat [Aktifkan peristiwa pembaruan untuk pelacak](#).

Untuk informasi selengkapnya, lihat [Pola Peristiwa dan Peristiwa](#) di Panduan EventBridge Pengguna Amazon.

## Topik

- [Aktifkan peristiwa pembaruan untuk pelacak](#)
- [Membuat aturan acara untuk Lokasi Amazon](#)
- [Contoh EventBridge acara Amazon untuk Amazon Location Service](#)

## Aktifkan peristiwa pembaruan untuk pelacak

Secara default, Lokasi Amazon hanya mengirim ENTER dan peristiwa EXIT geofence ke EventBridge Anda dapat mengaktifkan semua UPDATE peristiwa posisi yang difilter untuk pelacak yang akan dikirim EventBridge. Anda dapat melakukan ini ketika Anda [membuat](#) atau [memperbarui](#) pelacak.

Misalnya, untuk memperbarui pelacak yang ada menggunakan AWS CLI, Anda dapat menggunakan perintah berikut (gunakan nama sumber daya pelacak Anda sebagai pengganti). *MyTracker*

```
aws location update-tracker --tracker-name MyTracker --event-bridge-enabled
```

Untuk menonaktifkan peristiwa posisi untuk pelacak, Anda harus menggunakan API atau konsol Amazon Location Service.

## Membuat aturan acara untuk Lokasi Amazon

Anda dapat membuat [hingga 300 aturan per bus acara](#) EventBridge untuk mengonfigurasi tindakan yang diambil sebagai respons terhadap peristiwa Lokasi Amazon.

Misalnya, Anda dapat membuat aturan untuk peristiwa geofence di mana pemberitahuan push akan dikirim saat ponsel terdeteksi dalam batas geografis.

Untuk membuat aturan untuk acara Lokasi Amazon

Dengan menggunakan nilai berikut, [buat EventBridge aturan](#) berdasarkan peristiwa Lokasi Amazon:



- Untuk Tipe aturan, pilih Aturan dengan pola peristiwa.
- Dalam kotak pola acara, tambahkan pola berikut:

```
{
  "source": ["aws.geo"],
  "detail-type": ["Location Geofence Event"]
}
```

Untuk membuat aturan untuk pembaruan posisi pelacak, Anda dapat menggunakan pola berikut:

```
{
  "source": ["aws.geo"],
  "detail-type": ["Location Device Position Event"]
}
```

Anda dapat secara opsional menentukan hanya ENTER atau EXIT peristiwa dengan menambahkan detail tag (jika aturan Anda adalah untuk pembaruan posisi pelacak, hanya ada satu EventType, jadi tidak perlu memfilternya):

```
{
  "source": ["aws.geo"],
  "detail-type": ["Location Geofence Event"],
  "detail": {
    "EventType": ["ENTER"]
  }
}
```

Anda juga dapat secara opsional memfilter properti posisi atau geofence:

```
{
  "source": ["aws.geo"],
  "detail-type": ["Location Geofence Event"],
  "detail": {
    "EventType": ["ENTER"],
    "GeofenceProperties": {
      "Type": "LoadingDock"
    },
    "PositionProperties": {
      "VehicleType": "Truck"
    }
  }
}
```

```
}  
}
```

- Untuk target Pilih, pilih tindakan target yang akan diambil saat peristiwa diterima dari Amazon Location Service.

Misalnya, Anda dapat menggunakan topik Amazon Simple Notification Service (SNS) untuk mengirim email atau pesan teks ketika peristiwa terjadi. Anda harus terlebih dahulu membuat topik Amazon SNS menggunakan konsol Amazon SNS. Untuk informasi selengkapnya, lihat [Menggunakan Amazon SNS untuk pemberitahuan pengguna](#).

#### Warning

Praktik terbaik untuk mengonfirmasi bahwa aturan acara berhasil diterapkan atau tindakan otomatis Anda mungkin tidak dimulai seperti yang diharapkan. Untuk memverifikasi aturan acara Anda, mulai kondisi untuk aturan acara. Misalnya, simulasikan perangkat yang memasuki area geofenced.

Anda juga dapat menangkap semua acara dari Lokasi Amazon, hanya dengan mengecualikan detail-type bagian tersebut. Sebagai contoh:

```
{  
  "source": [  
    "aws.geo"  
  ]  
}
```

#### Note

Acara yang sama dapat disampaikan lebih dari satu kali. Anda dapat menggunakan id acara untuk menghapus duplikat peristiwa yang Anda terima.

## Contoh EventBridge acara Amazon untuk Amazon Location Service

Berikut ini adalah contoh acara untuk memasukkan geofence yang diprakarsai dengan menelepon. `BatchUpdateDevicePosition`

```

{
  "version": "0",
  "id": "aa11aa22-33a-4a4a-aaa5-example",
  "detail-type": "Location Geofence Event",
  "source": "aws.geo",
  "account": "636103698109",
  "time": "2020-11-10T23:43:37Z",
  "region": "eu-west-1",
  "resources": [
    "arn:aws:geo:eu-west-1:0123456789101:geofence-collection/GeofenceEvents-GeofenceCollection_EXAMPLE",
    "arn:aws:geo:eu-west-1:0123456789101:tracker/Tracker_EXAMPLE"
  ],
  "detail": {
    "EventType": "ENTER",
    "GeofenceId": "polygon_14",
    "DeviceId": "Device1-EXAMPLE",
    "SampleTime": "2020-11-10T23:43:37.531Z",
    "Position": [
      -123.12390073297821,
      49.23433613216247
    ],
    "Accuracy": {
      "Horizontal": 15.3
    },
    "GeofenceProperties": {
      "ExampleKey1": "ExampleField1",
      "ExampleKey2": "ExampleField2"
    },
    "PositionProperties": {
      "ExampleKey1": "ExampleField1",
      "ExampleKey2": "ExampleField2"
    }
  }
}

```

Berikut ini adalah contoh acara untuk keluar dari geofence yang diprakarsai dengan menelepon.

### BatchUpdateDevicePosition

```

{
  "version": "0",
  "id": "aa11aa22-33a-4a4a-aaa5-example",
  "detail-type": "Location Geofence Event",

```

```

"source": "aws.geo",
"account": "123456789012",
"time": "2020-11-10T23:41:44Z",
"region": "eu-west-1",
"resources": [
  "arn:aws:geo:eu-west-1:0123456789101:geofence-collection/GeofenceEvents-
GeofenceCollection_EXAMPLE",
  "arn:aws:geo:eu-west-1:0123456789101:tracker/Tracker_EXAMPLE"
],
"detail": {
  "EventType": "EXIT",
  "GeofenceId": "polygon_10",
  "DeviceId": "Device1-EXAMPLE",
  "SampleTime": "2020-11-10T23:41:43.826Z",
  "Position": [
    -123.08569321875426,
    49.23766166742559
  ],
  "Accuracy": {
    "Horizontal": 15.3
  },
  "GeofenceProperties": {
    "ExampleKey1": "ExampleField1",
    "ExampleKey2": "ExampleField2"
  },
  "PositionProperties": {
    "ExampleKey1": "ExampleField1",
    "ExampleKey2": "ExampleField2"
  }
}
}
}

```

Berikut ini adalah contoh peristiwa untuk update posisi, dimulai dengan memanggil `BatchUpdateDevicePosition`.

```

{
  "version": "0",
  "id": "aa11aa22-33a-4a4a-aaa5-example",
  "detail-type": "Location Device Position Event",
  "source": "aws.geo",
  "account": "123456789012",
  "time": "2020-11-10T23:41:44Z",
  "region": "eu-west-1",

```

```
"resources": [
  "arn:aws:geo:eu-west-1:0123456789101:tracker/Tracker_EXAMPLE"
],
"detail": {
  "EventType": "UPDATE",
  "TrackerName": "tracker_2",
  "DeviceId": "Device1-EXAMPLE",
  "SampleTime": "2020-11-10T23:41:43.826Z",
  "ReceivedTime": "2020-11-10T23:41:39.235Z",
  "Position": [
    -123.08569321875426,
    49.23766166742559
  ],
  "Accuracy": {
    "Horizontal": 15.3
  },
  "PositionProperties": {
    "ExampleKey1": "ExampleField1",
    "ExampleKey2": "ExampleField2"
  }
}
}
```

## Melacak penggunaan AWS IoT dan MQTT dengan Amazon Location Service

[MQTT](#) adalah protokol pesan ringan dan diadopsi secara luas yang dirancang untuk perangkat terbatas. AWS IoT Core mendukung koneksi perangkat yang menggunakan protokol MQTT dan protokol MQTT over WebSocket Secure (WSS).

[AWS IoT Core](#) menghubungkan perangkat ke AWS dan memungkinkan Anda untuk mengirim dan menerima pesan di antara mereka. Mesin AWS IoT Core aturan menyimpan kueri tentang topik pesan perangkat Anda dan memungkinkan Anda menentukan tindakan untuk mengirim pesan ke AWS layanan lain, seperti Amazon Location Service. Perangkat yang mengetahui lokasinya sebagai koordinat dapat meneruskan lokasinya ke Lokasi Amazon melalui mesin aturan.

**Note**

Perangkat mungkin mengetahui posisinya sendiri, misalnya melalui GPS bawaan. AWS IoT juga memiliki dukungan untuk pelacakan lokasi perangkat pihak ketiga. Untuk informasi selengkapnya, lihat [Lokasi Perangkat AWS IoT Inti](#) di Panduan Pengembang AWS IoT Inti.

Panduan berikut menjelaskan pelacakan menggunakan AWS IoT Core aturan. Anda juga dapat mengirim informasi perangkat ke AWS Lambda fungsi Anda sendiri, jika Anda perlu memprosesnya sebelum mengirim ke Lokasi Amazon. Untuk detail selengkapnya tentang penggunaan Lambda untuk memproses lokasi perangkat, lihat. [Menggunakan AWS Lambda dengan MQTT](#)

**Topik**

- [Prasyarat](#)
- [Buat AWS IoT Core aturan](#)
- [Uji AWS IoT Core aturan Anda di konsol](#)
- [Menggunakan AWS Lambda dengan MQTT](#)

**Prasyarat**

Sebelum Anda dapat mulai melacak, Anda harus menyelesaikan prasyarat berikut:

- [Buat sumber daya pelacak](#) yang akan Anda kirim data lokasi perangkat.
- [Buat peran IAM](#) untuk memberikan AWS IoT Core akses ke pelacak Anda.

Saat mengikuti langkah-langkah tersebut, gunakan kebijakan berikut untuk memberikan akses ke pelacak Anda:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "WriteDevicePosition",
      "Effect": "Allow",
      "Action": "geo:BatchUpdateDevicePosition",
      "Resource": "arn:aws:geo:*:*:tracker/*"
    }
  ]
}
```

```
}
```

## Buat AWS IoT Core aturan

Selanjutnya, buat AWS IoT Core aturan untuk meneruskan telemetri posisi perangkat Anda ke Amazon Location Service. Untuk informasi selengkapnya tentang membuat aturan, lihat topik berikut di Panduan AWS IoT Core Pengembang:

- [Membuat AWS IoT aturan](#) untuk informasi tentang membuat aturan baru.
- [Tindakan lokasi](#) untuk informasi khusus untuk membuat aturan untuk penerbitan ke Lokasi Amazon

## Uji AWS IoT Core aturan Anda di konsol

Jika saat ini tidak ada perangkat yang menerbitkan telemetri yang menyertakan lokasi, Anda dapat menguji aturan menggunakan konsol. AWS IoT Core Konsol memiliki klien pengujian tempat Anda dapat mempublikasikan pesan sampel untuk memverifikasi hasil solusi.

1. Masuk ke AWS IoT Core konsol di <https://console.aws.amazon.com/iot/>.
2. Di navigasi kiri, perluas Uji, dan pilih klien uji MQTT.
3. Di bawah Publikasikan ke topik, setel nama Topik ke ***iot/topik*** (atau nama topik yang Anda atur dalam AWS IoT Core aturan, jika berbeda), dan berikan yang berikut ini untuk muatan Pesan. Ganti stempel waktu ***1604940328*** dengan stempel waktu yang valid dalam 30 hari terakhir (stempel waktu yang lebih lama dari 30 hari diabaikan oleh pelacak Amazon Location Service).

```
{
  "payload": {
    "deviceid": "thing123",
    "timestamp": 1604940328,
    "location": { "lat": 49.2819, "long": -123.1187 },
    "accuracy": { "Horizontal": 20.5 },
    "positionProperties": { "field1": "value1", "field2": "value2" }
  }
}
```

4. Pilih Publikasikan ke topik untuk mengirim pesan pengujian.

5. Untuk memvalidasi bahwa pesan diterima oleh Amazon Location Service, gunakan AWS CLI perintah berikut. Jika Anda memodifikasinya selama pengaturan, ganti nama pelacak dengan yang Anda gunakan.

```
aws location batch-get-device-position --tracker-name MyTracker --device-ids  
thing123
```

## Menggunakan AWS Lambda dengan MQTT

Meskipun penggunaan AWS Lambda tidak lagi diperlukan saat mengirim data lokasi perangkat ke Lokasi Amazon untuk dilacak, Anda mungkin masih ingin menggunakan Lambda dalam beberapa kasus. Misalnya, jika Anda ingin memproses sendiri data lokasi perangkat Anda, sebelum mengirimkannya ke Lokasi Amazon. Topik berikut menjelaskan cara menggunakan Lambda untuk memproses pesan sebelum mengirimnya ke pelacak Anda. Untuk informasi lebih lanjut tentang pola ini, lihat [arsitektur referensi](#).

### Topik

- [Prasyarat](#)
- [Buat fungsi Lambda](#)
- [Buat AWS IoT Core aturan](#)
- [Uji AWS IoT Core aturan Anda di konsol](#)

### Prasyarat

Sebelum Anda dapat mulai melacak, Anda harus [membuat sumber daya pelacak](#). Untuk membuat sumber daya pelacak, Anda dapat menggunakan konsol Lokasi Amazon, API Lokasi AmazonAWS CLI, atau Amazon Location.

Contoh berikut menggunakan konsol Amazon Location Service untuk membuat resource tracker:

1. Buka konsol Amazon Location Service di <https://console.aws.amazon.com/location/>.
2. Di panel navigasi kiri, pilih Pelacak.
3. Pilih Buat pelacak.
4. Isi kotak-kotak berikut:



- Nama — Masukkan nama unik yang memiliki maksimal 100 karakter. Entri yang valid termasuk karakter alfanumerik, tanda hubung, dan garis bawah. Misalnya, *MyTracker*.
- Deskripsi - Masukkan deskripsi opsional. Misalnya, *Tracker untuk menyimpan posisi AWS IoT Core perangkat*.
- Pemfilteran posisi - Pilih pemfilteran yang ingin Anda gunakan untuk pembaruan posisi. Misalnya, pemfilteran berbasis akurasi.

## 5. Pilih Buat pelacak.

### Buat fungsi Lambda

Untuk membuat koneksi antara AWS IoT Core dan Amazon Location Service, Anda memerlukan AWS Lambda fungsi untuk memproses pesan yang diteruskan oleh. AWS IoT Core Fungsi ini akan mengekstrak data posisi apa pun, memformatnya untuk Amazon Location Service, dan mengirimkannya melalui Amazon Location Tracker API. Anda dapat membuat fungsi ini melalui AWS Lambda konsol, atau Anda dapat menggunakan AWS Command Line Interface (AWS CLI) atau AWS Lambda API.

Untuk membuat fungsi Lambda yang menerbitkan pembaruan posisi ke Lokasi Amazon menggunakan konsol:

1. Buka konsol AWS Lambda di <https://console.aws.amazon.com/lambda/>.
2. Dari navigasi kiri, pilih Fungsi.
3. Pilih Create Function, dan pastikan bahwa Author dari awal dipilih.
4. Isi kotak-kotak berikut:
  - Nama fungsi - Masukkan nama unik untuk fungsi Anda. Entri yang valid termasuk karakter alfanumerik, tanda hubung, dan garis bawah tanpa spasi. Misalnya, *MyLambda*.
  - Runtime - Pilih *Python 3.8*.
5. Pilih Buat fungsi.
6. Pilih tab Kode untuk membuka editor.
7. [Timpa kode placeholder `lambda\_function.py` dengan yang berikut ini, ganti nilai yang ditetapkan `TRACKER\_NAME` dengan nama pelacak yang Anda buat sebagai prasyarat.](#)

```
from datetime import datetime
import json
```

```
import os

import boto3

# Update this to match the name of your Tracker resource
TRACKER_NAME = "MyTracker"

"""
This Lambda function receives a payload from AWS IoT Core and publishes device
updates to
Amazon Location Service via the BatchUpdateDevicePosition API.

Parameter 'event' is the payload delivered from AWS IoT Core.

In this sample, we assume that the payload has a single top-level key 'payload' and
a nested key
'location' with keys 'lat' and 'long'. We also assume that the name of the device
is nested in
the payload as 'deviceid'. Finally, the timestamp of the payload is present as
'timestamp'. For
example:

>>> event
{ 'payload': { 'deviceid': 'thing123', 'timestamp': 1604940328,
  'location': { 'lat': 49.2819, 'long': -123.1187 },
  'accuracy': {'Horizontal': 20.5 },
  'positionProperties': {'field1':'value1','field2':'value2'} }
}

If your data doesn't match this schema, you can either use the AWS IoT Core rules
engine to
format the data before delivering it to this Lambda function, or you can modify the
code below to
match it.
"""
def lambda_handler(event, context):
    update = {
        "DeviceId": event["payload"]["deviceid"],
        "SampleTime": datetime.fromtimestamp(event["payload"]
["timestamp"]).strftime("%Y-%m-%dT%H:%M:%SZ"),
        "Position": [
            event["payload"]["location"]["long"],
            event["payload"]["location"]["lat"]
        ]
    }
```

```

    }
    if "accuracy" in event["payload"]:
        update["Accuracy"] = event["payload"]['accuracy']
    if "positionProperties" in event["payload"]:
        update["PositionProperties"] = event["payload"]['positionProperties']

    client = boto3.client("location")
    response = client.batch_update_device_position(TrackerName=TRACKER_NAME,
    Updates=[update])

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

```

8. Pilih Deploy untuk menyimpan fungsi yang diperbarui.
9. Pilih tab Konfigurasi.
10. Di bagian Izin, pilih nama Peran hyperlink untuk memberikan izin Amazon Location Service ke fungsi Lambda Anda.
11. Dari halaman Ringkasan peran Anda, pilih Tambahkan izin, lalu dari daftar tarik-turun, pilih Buat kebijakan sebaris.
12. Pilih tab JSON, dan timpa kebijakan dengan dokumen berikut. Ini memungkinkan fungsi Lambda Anda memperbarui posisi perangkat yang dikelola oleh semua sumber daya pelacak di semua Wilayah.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "WriteDevicePosition",
      "Effect": "Allow",
      "Action": "geo:BatchUpdateDevicePosition",
      "Resource": "arn:aws:geo:*:*:tracker/*"
    }
  ]
}


```

13. Pilih Tinjau kebijakan.
14. Masukkan nama kebijakan. Misalnya, *AmazonLocationTrackerWriteOnly*.
15. Pilih Buat kebijakan.

Anda dapat memodifikasi kode fungsi ini, jika perlu, untuk beradaptasi dengan skema pesan perangkat Anda sendiri.

Buat AWS IoT Core aturan

Selanjutnya, buat AWS IoT Core aturan untuk meneruskan telemetri posisi perangkat Anda ke AWS Lambda fungsi untuk transformasi dan publikasi ke Amazon Location Service. Aturan contoh yang diberikan mengasumsikan bahwa setiap transformasi muatan perangkat yang diperlukan ditangani oleh fungsi Lambda Anda. Anda dapat membuat aturan ini melalui AWS IoT Core konsol, AWS Command Line Interface (AWS CLI), atau AWS IoT Core API.

 Note

Saat AWS IoT konsol menangani izin yang diperlukan AWS IoT Core untuk mengizinkan menjalankan fungsi Lambda, jika Anda membuat aturan dari AWS CLI atau SDK, Anda [harus mengonfigurasi kebijakan untuk memberikan](#) izin. AWS IoT

Untuk membuat AWS IoT Core menggunakan konsol

1. Masuk ke AWS IoT Core konsol di <https://console.aws.amazon.com/iot/>.
2. Di navigasi kiri, perluas Act, dan pilih Rules.
3. Pilih Buat aturan untuk memulai panduan aturan baru.
4. Masukkan nama dan deskripsi untuk aturan Anda.
5. Untuk pernyataan kueri Aturan, perbarui FROM atribut untuk merujuk ke topik di mana setidaknya satu perangkat menerbitkan telemetri yang menyertakan lokasi. Jika Anda menguji solusinya, tidak diperlukan modifikasi.

```
SELECT * FROM 'iot/topic'
```

6. Di bawah Setel satu atau beberapa tindakan, pilih Tambah tindakan.
7. Pilih Kirim pesan ke fungsi lambda.
8. Pilih Konfigurasi tindakan.
9. Temukan dan pilih fungsi Lambda Anda dari daftar.
10. Pilih Tambahkan tindakan.
11. Pilih Buat aturan.

## Uji AWS IoT Core aturan Anda di konsol

Jika saat ini tidak ada perangkat yang menerbitkan telemetri yang menyertakan lokasi, Anda dapat menguji aturan dan solusi ini menggunakan konsol. AWS IoT Core Konsol memiliki klien pengujian tempat Anda dapat mempublikasikan pesan sampel untuk memverifikasi hasil solusi.

1. Masuk ke AWS IoT Core konsol di <https://console.aws.amazon.com/iot/>.
2. Di navigasi kiri, perluas Uji, dan pilih klien uji MQTT.
3. Di bawah Publikasikan ke topik, setel nama Topik ke ***iot/topik*** (atau nama topik yang Anda atur dalam AWS IoT Core aturan, jika berbeda), dan berikan yang berikut ini untuk muatan Pesan. Ganti stempel waktu ***1604940328*** dengan stempel waktu yang valid dalam 30 hari terakhir (stempel waktu yang lebih dari 30 hari diabaikan).

```
{
  "payload": {
    "deviceid": "thing123",
    "timestamp": 1604940328,
    "location": { "lat": 49.2819, "long": -123.1187 },
    "accuracy": { "Horizontal": 20.5 },
    "positionProperties": { "field1": "value1", "field2": "value2" }
  }
}
```

4. Pilih Publikasikan ke topik untuk mengirim pesan pengujian.
5. Untuk memvalidasi bahwa pesan diterima oleh Amazon Location Service, gunakan AWS CLI perintah berikut. Jika Anda memodifikasinya selama penyiapan, ganti nama pelacak dan id perangkat dengan yang Anda gunakan.

```
aws location batch-get-device-position --tracker-name MyTracker --device-ids
thing123
```

## Mengelola sumber daya pengumpulan geofence Anda

Kelola koleksi geofence Anda menggunakan konsol Lokasi Amazon, API Lokasi AmazonAWS CLI, atau Amazon Location.

## Buat daftar sumber daya koleksi geofence Anda

Anda dapat melihat daftar koleksi geofence menggunakan konsol Lokasi Amazon, API Lokasi AmazonAWS CLI, atau Amazon Location:

### Console

Untuk melihat daftar koleksi geofence menggunakan konsol Lokasi Amazon

1. Buka konsol Lokasi Amazon di <https://console.aws.amazon.com/location/>.
2. Pilih koleksi Geofence dari panel navigasi kiri.
3. Lihat daftar koleksi geofence Anda di bawah Koleksi geofence saya.

### API

Gunakan [ListGeofenceCollections](#) operasi dari Amazon Location Geofences API.

Contoh berikut adalah permintaan API untuk mendapatkan daftar koleksi geofence di akun. AWS

```
POST /geofencing/v0/list-collections
```

Berikut ini adalah contoh respons untuk `ListGeofenceCollections`:

```
{
  "Entries": [
    {
      "CollectionName": "ExampleCollection",
      "CreateTime": 2020-09-30T22:59:34.142Z,
      "Description": "string",
      "UpdateTime": 2020-09-30T23:59:34.142Z
    },
    "NextToken": "1234-5678-9012"
  ]
}
```

### CLI

Gunakan perintah [list-geofence-collections](#).

Contoh berikut adalah AWS CLI untuk mendapatkan daftar koleksi geofence di akun. AWS

```
aws location list-geofence-collections
```

## Dapatkan detail koleksi geofence

Anda bisa mendapatkan detail tentang sumber daya pengumpulan geofence apa pun di AWS akun menggunakan konsol Lokasi Amazon, API Lokasi AmazonAWS CLI, atau Amazon Location:

### Console

Untuk melihat detail koleksi geofence menggunakan konsol Lokasi Amazon

1. Buka konsol Lokasi Amazon di <https://console.aws.amazon.com/location/>.
2. Pilih koleksi Geofence dari panel navigasi kiri.
3. Di bawah Koleksi geofence saya, pilih tautan nama koleksi geofence target.

### API

Gunakan [DescribeGeofenceCollection](#) operasi dari Amazon Location Geofences API.

Contoh berikut adalah permintaan API untuk mendapatkan detail koleksi geofence.

*ExampleCollection*

```
GET /geofencing/v0/collections/ExampleCollection
```

Berikut ini adalah contoh respons untuk `DescribeGeofenceCollection`:

```
{
  "CollectionArn": "arn:aws:geo:us-west-2:123456789012:geofence-collection/
GeofenceCollection",
  "CollectionName": "ExampleCollection",
  "CreateTime": 2020-09-30T22:59:34.142Z,
  "Description": "string",
  "KmsKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
  "Tags": {
    "Tag1" : "Value1"
  },
  "UpdateTime": 2020-09-30T23:59:34.142Z
}
```

## CLI

Gunakan perintah [describe-geofence-collection](#).

Contoh berikut adalah AWS CLI untuk mendapatkan detail koleksi geofence untuk.

*ExampleCollection*


```
aws location describe-geofence-collection \  
  --collection-name "ExampleCollection"
```

## Hapus koleksi geofence

Anda dapat menghapus koleksi geofence dari AWS akun menggunakan konsol Lokasi Amazon, API Lokasi AmazonAWS CLI, atau Amazon Location.

### Console

Untuk menghapus koleksi geofence menggunakan konsol Lokasi Amazon

 Warning

Operasi ini menghapus sumber daya secara permanen.

1. Buka konsol Lokasi Amazon di <https://console.aws.amazon.com/location/>.
2. Pilih koleksi Geofence dari panel navigasi kiri.
3. Di bawah Koleksi geofence saya, pilih koleksi geofence target.
4. Pilih Hapus koleksi geofence.

### API

Gunakan [DeleteGeofenceCollection](#) operasi dari Amazon Location API.

Contoh berikut adalah permintaan API untuk menghapus koleksi geofence.

*ExampleCollection*

```
DELETE /geofencing/v0/collections/ExampleCollection
```



Berikut ini adalah contoh respons untuk `DeleteGeofenceCollection`:

```
HTTP/1.1 200
```

## CLI

Gunakan perintah [delete-geofence-collection](#).

Contoh berikut adalah AWS CLI perintah untuk menghapus koleksi geofence.

*ExampleCollection*

```
aws location delete-geofence-collection \  
  --collection-name "ExampleCollection"
```

## Daftar geofences yang disimpan

Anda dapat mencantumkan geofences yang disimpan dalam koleksi geofence tertentu menggunakan konsol Lokasi Amazon, API Lokasi Amazon, AWS CLI atau Amazon Location.

### Console

Untuk melihat daftar geofences menggunakan konsol Lokasi Amazon

1. Buka konsol Lokasi Amazon di <https://console.aws.amazon.com/location/>.
2. Pilih koleksi Geofence dari panel navigasi kiri.
3. Di bawah Koleksi geofence saya, pilih tautan nama koleksi geofence target.
4. Lihat geofences dalam koleksi geofence di bawah Geofences

## API

Gunakan [ListGeofences](#) operasi dari Amazon Location Geofences API.

Contoh berikut adalah permintaan API untuk mendapatkan daftar geofences yang disimpan dalam koleksi geofence. *ExampleCollection*

```
POST /geofencing/v0/collections/ExampleCollection/list-geofences
```

Berikut ini adalah contoh respons untuk `ListGeofences`:

```
{
  "Entries": [
    {
      "CreateTime": 2020-09-30T22:59:34.142Z,
      "GeofenceId": "geofence-1",
      "Geometry": {
        "Polygon": [
          [-5.716667, -15.933333,
            [-14.416667, -7.933333],
            [-12.316667, -37.066667],
            [-5.716667, -15.933333]
          ]
        },
      "Status": "ACTIVE",
      "UpdateTime": 2020-09-30T23:59:34.142Z
    }
  ],
  "NextToken": "1234-5678-9012"
}
```

## CLI

Gunakan perintah [list-geofences](#).

Contoh berikut adalah AWS CLI untuk mendapatkan daftar geofences yang disimpan dalam koleksi geofence. *ExampleCollection*

```
aws location list-geofences \
  --collection-name "ExampleCollection"
```

## Dapatkan detail geofence

Anda bisa mendapatkan detail geofence tertentu, seperti waktu pembuatan, waktu pembaruan, geometri, dan status, dari koleksi geofence menggunakan konsol Lokasi Amazon, AWS CLI atau API Lokasi Amazon.

## Console

Untuk melihat status geofence menggunakan konsol Lokasi Amazon

1. Buka konsol Lokasi Amazon di <https://console.aws.amazon.com/location/>.

2. Pilih koleksi Geofence dari panel navigasi kiri.
3. Di bawah Koleksi geofence saya, pilih tautan nama koleksi geofence target.
4. Di bawah Geofences, Anda akan dapat melihat status geofences Anda.

## API

Gunakan [GetGeofence](#) operasi dari Amazon Location Geofences API.

Contoh berikut adalah permintaan API untuk mendapatkan detail geofence dari koleksi geofence.

*ExampleCollection*

```
GET /geofencing/v0/collections/ExampleCollection/geofences/ExampleGeofence1
```

Berikut ini adalah contoh respons untuk `GetGeofence`:

```
{
  "CreateTime": 2020-09-30T22:59:34.142Z,
  "GeofenceId": "ExampleGeofence1",
  "Geometry": {
    "Polygon": [
      [-1,-1],
      [1,-1],
      [0,1],
      [-1,-1]
    ]
  },
  "Status": "ACTIVE",
  "UpdateTime": 2020-09-30T23:59:34.142Z
}
```

## CLI

Gunakan perintah [get-geofence](#).

Contoh berikut adalah AWS CLI untuk mendapatkan detail koleksi geofence untuk.

*ExampleCollection*

```
aws location get-geofence \
  --collection-name "ExampleCollection" \
  --geofence-id "ExampleGeofence1"
```

## Hapus geofences

Anda dapat menghapus geofences dari koleksi geofence menggunakan konsol Lokasi Amazon, API Lokasi Amazon, AWS CLI atau Amazon Location.

### Console

Untuk menghapus geofence menggunakan konsol Lokasi Amazon

#### Warning

Operasi ini menghapus sumber daya secara permanen.

1. Buka konsol Lokasi Amazon di <https://console.aws.amazon.com/location/>.
2. Pilih koleksi Geofence dari panel navigasi kiri.
3. Di bawah Koleksi geofence saya, pilih tautan nama koleksi geofence target.
4. Di bawah Geofences, pilih geofence target.
5. Pilih Hapus geofence.

### API

Gunakan [BatchDeleteGeofence](#) operasi dari Amazon Location Geofences API.

Contoh berikut adalah permintaan API untuk menghapus geofences dari koleksi geofence.

*ExampleCollection*

```
POST /geofencing/v0/collections/ExampleCollection/delete-geofences
Content-type: application/json

{
  "GeofenceIds": [ "ExampleGeofence11" ]
}
```

Berikut ini adalah contoh respon sukses untuk [BatchDeleteGeofence](#).

```
HTTP/1.1 200
```

## CLI

Gunakan perintah [batch-delete-geofence](#).

Contoh berikut adalah AWS CLI perintah untuk menghapus geofences dari koleksi geofence.

*ExampleCollection*

```
aws location batch-delete-geofence \  
  --collection-name "ExampleCollection" \  
  --geofence-ids "ExampleGeofence11"
```

## Mengelola sumber daya pelacak Anda

Anda dapat mengelola pelacak menggunakan konsol Lokasi Amazon, API Lokasi AmazonAWS CLI, atau Amazon Location.

### Buat daftar pelacak Anda

Anda dapat melihat daftar pelacak menggunakan konsol Lokasi Amazon, API Lokasi AmazonAWS CLI, atau Amazon:

#### Console

Untuk melihat daftar pelacak yang ada menggunakan konsol Lokasi Amazon

1. Buka konsol Lokasi Amazon di <https://console.aws.amazon.com/location/>.
2. Pilih Pelacak dari navigasi kiri.
3. Lihat daftar sumber daya pelacak Anda di bawah Pelacak saya.

#### API

Gunakan [ListTrackers](#) operasi dari Amazon Location Trackers API.

Contoh berikut adalah permintaan API untuk mendapatkan daftar pelacak di AWS akun Anda.

```
POST /tracking/v0/list-trackers
```

Berikut ini adalah contoh respons untuk [ListTrackers](#):

```
{
```

```
"Entries": [  
  {  
    "CreateTime": 2020-10-02T19:09:07.327Z,  
    "Description": "string",  
    "TrackerName": "ExampleTracker",  
    "UpdateTime": 2020-10-02T19:10:07.327Z  
  }  
],  
"NextToken": "1234-5678-9012"  
}
```

## CLI

Gunakan perintah [list-trackers](#).

Contoh berikut adalah AWS CLI untuk mendapatkan daftar pelacak di AWS akun Anda.

```
aws location list-trackers
```

## Memutuskan sambungan pelacak dari koleksi geofence

Anda dapat memutuskan sambungan pelacak dari koleksi geofence menggunakan konsol Lokasi Amazon, API Lokasi Amazon, AWS CLI atau Amazon Location:

### Console

Untuk memisahkan pelacak dari koleksi geofence terkait menggunakan konsol Lokasi Amazon

1. Buka konsol Lokasi Amazon di <https://console.aws.amazon.com/location/>.
2. Pilih Pelacak dari panel navigasi kiri.
3. Di bawah Pelacak saya, pilih tautan nama pelacak target.
4. Di bawah Koleksi Geofence Tertaut, pilih koleksi geofence dengan status Tertaut.
5. Pilih Putuskan Tautan.

### API

Gunakan [DisassociateTrackerConsumer](#) operasi dari Amazon Location Trackers API.

Contoh berikut adalah permintaan API untuk memisahkan pelacak dari koleksi geofence terkait.

```
DELETE /tracking/v0/trackers/ExampleTracker/consumers/arn:aws:geo:us-west-2:123456789012:geofence-collection/ExampleCollection
```

Berikut ini adalah contoh respons untuk [DisassociateTrackerConsumer](#):

```
HTTP/1.1 200
```

## CLI

Gunakan perintah [disassociate-tracker-consumer](#).

Contoh berikut adalah AWS CLI perintah untuk memisahkan pelacak dari koleksi geofence terkait.

```
aws location disassociate-tracker-consumer \  
  --consumer-arn "arn:aws:geo:us-west-2:123456789012:geofence-collection/  
ExampleCollection" \  
  --tracker-name "ExampleTracker"
```

## Dapatkan detail pelacak

Anda bisa mendapatkan detail tentang pelacak apa pun di AWS akun Anda dengan menggunakan konsol Lokasi Amazon, API Lokasi Amazon AWS CLI, atau Amazon Location.

### Console

Untuk melihat detail pelacak dengan menggunakan konsol Lokasi Amazon

1. Buka konsol Lokasi Amazon di <https://console.aws.amazon.com/location/>.
2. Pilih Pelacak dari navigasi kiri.
3. Di bawah Pelacak saya, pilih tautan nama pelacak target.
4. Lihat detail pelacak di bawah Informasi.

### API

Gunakan [DescribeTracker](#) operasi dari Amazon Location Tracker API.

Contoh berikut adalah permintaan API untuk mendapatkan detail pelacak. *ExampleTracker*

```
GET /tracking/v0/trackers/ExampleTracker
```

Berikut ini adalah contoh respons untuk [DescribeTracker](#):

```
{
  "CreateTime": 2020-10-02T19:09:07.327Z,
  "Description": "string",
  "EventBridgeEnabled": false,
  "KmsKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
  "PositionFiltering": "TimeBased",
  "Tags": {
    "Tag1" : "Value1"
  },
  "TrackerArn": "arn:aws:geo:us-west-2:123456789012:tracker/ExampleTracker",
  "TrackerName": "ExampleTracker",
  "UpdateTime": 2020-10-02T19:10:07.327Z
}
```

## CLI

Gunakan perintah [describe-tracker](#).

Contoh berikut adalah AWS CLI perintah untuk mendapatkan detail pelacak untuk *ExampleTracker*.

```
aws location describe-tracker \
  --tracker-name "ExampleTracker"
```

## Hapus pelacak

Anda dapat menghapus pelacak dari AWS akun menggunakan konsol Lokasi Amazon, API Lokasi Amazon, AWS CLI, atau Amazon Location:

### Console

Untuk menghapus sumber daya peta yang ada menggunakan konsol Lokasi Amazon

#### Warning

Operasi ini menghapus sumber daya secara permanen. Jika sumber daya pelacak sedang digunakan, Anda mungkin mengalami kesalahan. Pastikan bahwa sumber daya target bukan dependensi untuk aplikasi Anda.



1. Buka konsol Lokasi Amazon di <https://console.aws.amazon.com/location/>.
2. Pilih Pelacak dari panel navigasi kiri.
3. Di bawah Pelacak saya, pilih pelacak target.
4. Pilih Hapus pelacak.

## API

Gunakan [DeleteTracker](#) operasi dari Amazon Location Tracker API.

Contoh berikut adalah permintaan API untuk menghapus pelacak *ExampleTracker*.

```
DELETE /tracking/v0/trackers/ExampleTracker
```

Berikut ini adalah contoh respons untuk [DeleteTracker](#):

```
HTTP/1.1 200
```

## CLI

Gunakan perintah [delete-tracker](#).

Contoh berikut adalah AWS CLI perintah untuk menghapus pelacak *ExampleTracker*.

```
aws location delete-tracker \  
  --tracker-name "ExampleTracker"
```

## Contoh aplikasi seluler Geofencing dan Tracking

Topik ini mencakup tutorial yang dirancang untuk mendemonstrasikan fitur utama menggunakan geofences dan pelacak Lokasi Amazon dalam aplikasi seluler. Aplikasi menunjukkan bagaimana pelacak dan geofence berinteraksi menggunakan kombinasi fitur Lambda, AWS IoT dan Amazon Location. Ada dua tutorial yang tersedia.

- [Contoh aplikasi pelacakan dan geofencing untuk Android, dan Anda dapat mengkloning file proyek dari GitHub: https://github.com/aws-geospatial/tree/main/.amazon-location-samples-android-tracking-with-geofence-notifications](https://github.com/aws-geospatial/tree/main/.amazon-location-samples-android-tracking-with-geofence-notifications)

- [Contoh aplikasi pelacakan dan geofencing untuk iOS, dan Anda dapat mengkloning file proyek dari GitHub: https://github.com/aws-geospatial/ /tree/main/.amazon-location-samples-ios-tracking-with-geofence-notifications](https://github.com/aws-geospatial/.amazon-location-samples-ios-tracking-with-geofence-notifications)

## Contoh aplikasi pelacakan dan geofence untuk Android

Topik ini mencakup tutorial Android yang dirancang untuk mendemonstrasikan fitur utama menggunakan geofences dan pelacak Lokasi Amazon dalam aplikasi seluler. Aplikasi menunjukkan bagaimana pelacak dan geofence berinteraksi menggunakan kombinasi fitur Lambda, AWS IoT dan Amazon Location.

### Topik

- [Membuat sumber daya Lokasi Amazon untuk aplikasi Anda](#)
- [Buat Koleksi Geofence](#)
- [Tautkan pelacak ke koleksi geofence](#)
- [Menggunakan AWS Lambda dengan MQTT](#)
- [Siapkan contoh kode aplikasi](#)
- [Menggunakan aplikasi sampel](#)

### Membuat sumber daya Lokasi Amazon untuk aplikasi Anda

Untuk memulai, Anda perlu membuat sumber daya Lokasi Amazon yang diperlukan. Sumber daya ini akan sangat penting untuk fungsionalitas aplikasi dan mengeksekusi cuplikan kode yang disediakan.

#### Note

Jika Anda belum membuat AWS akun, ikuti petunjuk di panduan pengguna [pengelolaan AWS akun](#).

Untuk memulai, Anda perlu membuat Id Kolam Identitas Amazon Cognito, gunakan prosedur berikut:

1. Buka [konsol Amazon Cognito](#) dan pilih Identity pool dari menu sebelah kiri, lalu pilih Create Identity pool.
2. Pastikan Akses Tamu dicentang, dan tekan Next untuk continue.

3. Selanjutnya buat peran IAM baru atau Gunakan peran IAM yang ada.
4. Masukkan nama kumpulan Identity, dan pastikan Identity Pool memiliki akses ke (geo) sumber daya Lokasi Amazon untuk peta dan pelacak yang akan Anda buat di prosedur selanjutnya.

Selanjutnya Anda perlu membuat dan menata peta di konsol Lokasi AWS Amazon, gunakan prosedur berikut:

1. Arahkan ke [bagian Maps](#) di konsol Lokasi Amazon dan pilih Buat Peta.
2. Berikan peta baru Nama dan Deskripsi. Rekam nama yang Anda tetapkan, seperti yang digunakan nanti dalam tutorial.
3. Saat memilih gaya peta, pertimbangkan penyedia data peta. Lihat bagian 82 dari [ketentuan AWS layanan](#) untuk detail selengkapnya.
4. Terima [Syarat dan Ketentuan Lokasi Amazon](#), lalu pilih Buat Peta, untuk menyelesaikan proses pembuatan peta.

Selanjutnya Anda perlu membuat pelacak di konsol Lokasi Amazon, gunakan prosedur berikut:

1. Buka [bagian Maps](#) di konsol Lokasi Amazon.
2. Pilih Buat pelacak.
3. Isi kolom yang diperlukan. Catat Nama pelacak karena akan dibiaskan di seluruh tutoiral ini.
4. Di bawah bidang Pemfilteran posisi, pilih opsi yang paling sesuai dengan cara Anda ingin menggunakan sumber daya pelacak Anda. Jika Anda tidak mengatur Pemfilteran posisi, pengaturan defaultnya adalah TimeBased. Untuk informasi selengkapnya, lihat [Pelacak](#), dan [PositionFiltering](#) di Referensi API Lokasi Amazon.
5. Pilih Buat pelacak untuk menyelesaikan pembuatan pelacak.

## Buat Koleksi Geofence

Sekarang Anda akan membuat koleksi geofence. Anda dapat menggunakan konsol, API atau CLI. Prosedur berikut memandu Anda melalui setiap opsi.

- Buat koleksi geofence menggunakan konsol Lokasi Amazon:
  1. Buka bagian [Koleksi Geofence](#) di konsol Lokasi Amazon.
  2. Pilih Buat koleksi geofence.

3. Berikan nama dan deskripsi untuk koleksi.
  4. Di bawah EventBridge aturan dengan Amazon CloudWatch sebagai target, Anda dapat membuat EventBridge aturan opsional untuk mulai bereaksi terhadap peristiwa geofence. Hal ini memungkinkan Amazon Location untuk mempublikasikan acara ke Amazon CloudWatch Logs.
  5. Tekan Buat koleksi geofence untuk menyelesaikan pembuatan koleksi.
- Membuat koleksi geofence menggunakan Amazon Location API:

Gunakan [CreateGeofenceCollection](#) operasi dari Amazon Location Geofences API. Contoh berikut menggunakan permintaan API untuk membuat koleksi geofence yang disebut.

***GEOCOLLECTION\_NAME***

```
POST /geofencing/v0/collections
Content-type: application/json

{
  "CollectionName": "GEOCOLLECTION_NAME",
  "Description": "Geofence collection 1 for shopping center",
  "Tags": {
    "Tag1" : "Value1"
  }
}
```

- Buat koleksi geofence menggunakan perintah CLI AWS :

Gunakan perintah `create-geofence-collection`. Contoh berikut menggunakan AWS CLI untuk membuat koleksi geofence yang disebut. *GEOCOLLECTION\_NAME* Untuk informasi selengkapnya tentang penggunaan AWS CLI, lihat Dokumentasi [Antarmuka Baris AWS Perintah](#).

```
aws location \
  create-geofence-collection \
  --collection-name "ExampleGeofenceCollection" \
  --description "Shopping center geofence collection" \
  --tags Tag1=Value1
```

## Tautkan pelacak ke koleksi geofence

Untuk menautkan pelacak ke koleksi geofence, Anda dapat menggunakan konsol, API, atau CLI. Prosedur berikut memandu Anda melalui setiap opsi.

Menautkan sumber daya pelacak ke koleksi geofence menggunakan konsol Amazon Location Service:

1. Buka konsol Lokasi Amazon.
2. Di panel navigasi kiri, pilih Pelacak.
3. Di bawah Pelacak Perangkat, pilih tautan nama pelacak target.
4. Di bawah Koleksi Geofence Tertaut, pilih Link Geofence Collection.
5. Di jendela Linked Geofence Collection, pilih koleksi geofence dari menu tarik-turun.
6. Pilih Tautkan.
7. Setelah Anda menautkan sumber daya pelacak, itu akan diberi status Aktif.

Menautkan resource pelacak ke koleksi geofence menggunakan Amazon Location API:

Gunakan `AssociateTrackerConsumer` operasi dari Amazon Location Trackers API. Contoh berikut menggunakan permintaan API yang mengaitkan `ExampleTracker` dengan koleksi geofence menggunakan Amazon Resource Name (ARN).

```
POST /tracking/v0/trackers/ExampleTracker/consumers
Content-type: application/json
{
  "ConsumerArn": "arn:aws:geo:us-west-2:123456789012:geofence-
collection/GOECOLLECTION_NAME"
}
```

Tautkan sumber daya pelacak ke koleksi geofence menggunakan perintah CLI AWS :

Gunakan perintah `associate-tracker-consumer` . Contoh berikut menggunakan AWS CLI untuk membuat koleksi geofence yang disebut. *GOECOLLECTION\_NAME*

```
aws location \
associate-tracker-consumer \
  --consumer-arn "arn:aws:geo:us-west-2:123456789012:geofence-
collection/GOECOLLECTION_NAME" \
  --tracker-name "ExampleTracker"
```

## Menggunakan AWS Lambda dengan MQTT

Untuk membuat koneksi antara AWS IoT dan Lokasi Amazon, Anda memerlukan fungsi Lambda untuk memproses pesan yang diteruskan oleh peristiwa. EventBridge CloudWatch Fungsi ini akan mengekstrak data posisi apa pun, memformatnya untuk Lokasi Amazon, dan mengirimkannya melalui Amazon Location Tracker API.

Prosedur berikut menunjukkan cara membuat fungsi ini melalui konsol Lambda:

1. Buka [konsol](#) .
2. Dari navigasi kiri, pilih Fungsi.
3. Kemudian pilih Create Function, dan pastikan bahwa opsi Author from scratch dipilih.
4. berikan nama Fungsi, dan untuk opsi Runtime, pilih Node.js 16.x.
5. Pilih Buat fungsi.
6. Buka tab Kode untuk mengakses editor.
7. Timpa kode placeholder dalam `index.js` file dengan yang berikut:

```
const AWS = require('aws-sdk')
const iot = new AWS.Iot();
exports.handler = function(event) {
  console.log("event===>>>", JSON.stringify(event));
  var param = {
    endpointType: "iot:Data-ATS"
  };
  iot.describeEndpoint(param, function(err, data) {
    if (err) {
      console.log("error===>>>", err, err.stack); // an error occurred
    } else {
      var endp = data['endpointAddress'];
      const iotdata = new AWS.IotData({endpoint: endp});
      const trackerEvent = event["detail"]["EventType"];
      const src = event["source"];
      const time = event["time"];
      const gfId = event["detail"]["GeofenceId"];
      const resources = event["resources"][0];
      const splitResources = resources.split(".");
      const geofenceCollection = splitResources[splitResources.length -
1];

      const coordinates = event["detail"]["Position"];
```

```

const deviceId = event["detail"]["DeviceId"];
console.log("deviceId===>>>", deviceId);
const msg = {
  "trackerEventType" : trackerEvent,
  "source" : src,
  "eventTime" : time,
  "geofenceId" : gfId,
  "coordinates": coordinates,
  "geofenceCollection": geofenceCollection
};
const params = {
  topic: `${deviceId}/tracker`,
  payload: JSON.stringify(msg),
  qos: 0
};
iotdata.publish(params, function(err, data) {
  if (err) {
    console.log("error===>>>", err, err.stack); // an error
    occurred
  } else {
    console.log("Lambda triggered===>>>", trackerEvent); //
    successful response
  }
});
});
}

```

8. Tekan tombol Deploy untuk menyimpan fungsi yang diperbarui.
9. Selanjutnya buka tab Konfigurasi.
10. Di bagian Pemicu, tekan tombol Tambah Pemicu.
11. Pilih EventBridge (CloudWatch Acara) di bidang Sumber.
12. Pilih opsi Aturan yang Ada.
13. Masukkan nama aturan, misalnya AmazonLocationMonitor-GEOFENCECOLLECTION\_NAME.
14. Tekan tombol Tambah.
15. Ini juga akan melampirkan pernyataan kebijakan berbasis Sumber Daya di tab izin

Sekarang Anda akan mengatur MQTT Test Client menggunakan AWS IoT, gunakan prosedur berikut:

1. Buka <https://console.aws.amazon.com/iot/>.
2. Di panel navigasi kiri, pilih klien pengujian MQTT.
3. Anda akan melihat bagian berjudul MQTT test client di mana Anda dapat mengonfigurasi koneksi MQTT Anda.
4. Setelah mengkonfigurasi pengaturan yang diperlukan, klik tombol Connect untuk membuat koneksi ke broker MQTT menggunakan parameter yang disediakan.
5. Catat titik akhir, seperti yang digunakan nanti di tutorial.

Setelah terhubung ke klien pengujian, Anda dapat berlangganan topik MQTT atau mempublikasikan pesan ke topik menggunakan bidang masukan masing-masing yang disediakan di antarmuka klien pengujian MQTT. Selanjutnya Anda akan membuat AWS IoT kebijakan.

6. Di menu sebelah kiri, di bawah opsi Kelola perluas Keamanan dan klik Kebijakan.
7. Klik tombol Create Policy.
8. Masukkan nama kebijakan.
9. Pada Dokumen Kebijakan pilih tab JSON.
10. Salin tempel kebijakan yang ditunjukkan di bawah ini, tetapi pastikan untuk memperbarui semua elemen dengan Anda *REGION* dan *ACCOUNT\_ID*:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "iot:Connect",
        "iot:Publish",
        "iot:Subscribe",
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:REGION:ACCOUNT_ID:client/${cognito-identity.amazonaws.com:sub}",
        "arn:aws:iot:REGION:ACCOUNT_ID:topic/${cognito-identity.amazonaws.com:sub}",
        "arn:aws:iot:REGION:ACCOUNT_ID:topicfilter/${cognito-identity.amazonaws.com:sub}/*",
        "arn:aws:iot:REGION:ACCOUNT_ID:topic/${cognito-identity.amazonaws.com:sub}/tracker"
      ]
    }
  ]
}
```



```

    ],
    "Effect": "Allow"
  }
]
}

```

## 11. Pilih tombol Create untuk menyelesaikan.

Setelah menyelesaikan prosedur sebelumnya, Anda sekarang akan memperbarui izin untuk peran tamu sebagai berikut:

1. Arahkan ke Amazon Cognito dan buka kumpulan identitas Anda. Kemudian, lanjutkan ke akses pengguna dan pilih peran tamu.
2. Klik kebijakan izin untuk mengaktifkan pengeditan.

```

{
  'Version': '2012-10-17',
  'Statement': [
    {
      'Action': [
        'geo:GetMap*',
        'geo:BatchUpdateDevicePosition',
        'geo:BatchEvaluateGeofences',
        'iot:Subscribe',
        'iot:Publish',
        'iot:Connect',
        'iot:Receive',
        'iot:AttachPrincipalPolicy',
        'iot:AttachPolicy',
        'iot:DetachPrincipalPolicy',
        'iot:DetachPolicy'
      ],
      'Resource': [
        'arn:aws:geo:us-east-1:{USER_ID}:map/{MAP_NAME}',
        'arn:aws:geo:us-east-1:{USER_ID}:tracker/{TRACKER_NAME}',
        'arn:aws:geo:us-east-1:{USER_ID}:geofence-collection/
{GEOFENCE_COLLECTION_NAME}',
        'arn:aws:iot:us-east-1:{USER_ID}:client/${cognito-
identity.amazonaws.com:sub}',
        'arn:aws:iot:us-east-1:{USER_ID}:topic/${cognito-
identity.amazonaws.com:sub}',

```

```

        'arn:aws:iot:us-east-1:{USER_ID}:topicfilter/${cognito-
identity.amazonaws.com:sub}/*',
        'arn:aws:iot:us-east-1:{USER_ID}:topic/${cognito-
identity.amazonaws.com:sub}/tracker'
    ],
    'Effect': 'Allow'
},
{
    'Condition': {
        'StringEquals': {
            'cognito-identity.amazonaws.com:sub': '${cognito-
identity.amazonaws.com:sub}'
        }
    },
    'Action': [
        'iot:AttachPolicy',
        'iot:DetachPolicy',
        'iot:AttachPrincipalPolicy',
        'iot:DetachPrincipalPolicy'
    ],
    'Resource': [
        '*'
    ],
    'Effect': 'Allow'
}
]
}

```

3. Dengan perubahan kebijakan di atas, semua AWS sumber daya yang diperlukan sekarang dikonfigurasi dengan tepat untuk aplikasi.

Siapkan contoh kode aplikasi

1. Kloning repositori ini: <https://github.com/aws-geospatial/amazon-location-samples-android/tree/main/tracking-with-geofence-notifications> ke mesin lokal Anda.
2. Buka AmazonSampleSDKApp proyek di Android Studio.
3. Buat dan jalankan aplikasi di perangkat Android atau emulator Anda.

Menggunakan aplikasi sampel

Untuk menggunakan sampel ikuti prosedur ini:

- **Buat `custom.properties`:**

Untuk mengonfigurasi `custom.properties` file Anda, ikuti langkah-langkah berikut:

1. Buka editor teks atau IDE pilihan Anda.
2. Buat file baru.
3. Simpan file dengan nama `custom.properties`.
4. Perbarui `custom.properties` dengan contoh kode berikut, dan ganti, `MQTT_END_POINT`, `POLICY_NAME`, `GEOFENCE_COLLECTION_NAME`, dan `TOPIC_TRACKER` dengan nama resouce Anda:

```
MQTT_END_POINT=YOUR_END_POINT.us-east-1.amazonaws.com
POLICY_NAME=YOUR_POLICY
GEOFENCE_COLLECTION_NAME=YOUR_GEOFENCE
TOPIC_TRACKER=YOUR_TRACKER
```

5. Bersihkan dan Bangun kembali proyek. Setelah ini, Anda dapat menjalankan proyek.

- **Masuk:**

Untuk masuk ke aplikasi, ikuti langkah-langkah di bawah ini:

1. Tekan tombol Masuk.
2. Berikan Identity Pool Id, nama Tracker, dan nama Map.
3. Tekan Masuk lagi untuk menyelesaikan.

- **Kelola Filter:**

Buka layar konfigurasi, dan lakukan hal berikut:

1. Aktifkan atau nonaktifkan filter menggunakan UI sakelar.
2. Perbarui filter Waktu dan Jarak bila diperlukan.

- **Operasi Pelacakan:**

Buka layar pelacakan dan lakukan hal berikut:

- Anda dapat memulai dan menghentikan pelacakan di latar depan, latar belakang, atau dalam mode penghemat baterai dengan menekan tombol masing-masing.

## Contoh aplikasi pelacakan dan geofencing untuk iOS

Topik ini mencakup tutorial iOS yang dirancang untuk mendemonstrasikan fitur utama menggunakan geofences dan pelacak Lokasi Amazon dalam aplikasi seluler. Aplikasi menunjukkan bagaimana pelacak dan geofence berinteraksi menggunakan kombinasi fitur Lambda, AWS IoT dan Amazon Location.

### Topik

- [Membuat sumber daya Lokasi Amazon untuk aplikasi Anda](#)
- [Buat Koleksi Geofence](#)
- [Tautkan pelacak ke koleksi geofence](#)
- [Menggunakan AWS Lambda dengan MQTT](#)
- [Menyiapkan contoh kode aplikasi](#)
- [Menggunakan aplikasi sampel](#)

### Membuat sumber daya Lokasi Amazon untuk aplikasi Anda

Untuk memulai, Anda perlu membuat sumber daya Lokasi Amazon yang diperlukan. Sumber daya ini akan sangat penting untuk fungsionalitas aplikasi dan mengeksekusi cuplikan kode yang disediakan.

#### Note

Jika Anda belum membuat AWS akun, ikuti petunjuk di panduan pengguna [pengelolaan AWS akun](#).

Untuk memulai, Anda perlu membuat Id Kolam Identitas Amazon Cognito, gunakan prosedur berikut:

1. Buka [konsol Amazon Cognito](#) dan pilih Identity pool dari menu sebelah kiri, lalu pilih Create Identity pool.
2. Pastikan Akses Tamu dicentang, dan tekan Next untuk continue.
3. Selanjutnya buat peran IAM baru atau Gunakan peran IAM yang ada.
4. Masukkan nama kumpulan Identity, dan pastikan Identity Pool memiliki akses ke (geo) sumber daya Lokasi Amazon untuk peta dan pelacak yang akan Anda buat di prosedur selanjutnya.

Selanjutnya Anda perlu membuat dan menata peta di konsol Lokasi AWS Amazon, gunakan prosedur berikut:

1. Arahkan ke [bagian Maps](#) di konsol Lokasi Amazon dan pilih Buat Peta.
2. Berikan peta baru Nama dan Deskripsi. Rekam nama yang Anda tetapkan, seperti yang digunakan nanti dalam tutorial.
3. Saat memilih gaya peta, pertimbangkan penyedia data peta. Lihat bagian 82 dari [ketentuan AWS layanan](#) untuk detail selengkapnya.
4. Terima [Syarat dan Ketentuan Lokasi Amazon](#), lalu pilih Buat Peta, untuk menyelesaikan proses pembuatan peta.

Selanjutnya Anda perlu membuat pelacak di konsol Lokasi Amazon, gunakan prosedur berikut:

1. Buka [bagian Maps](#) di konsol Lokasi Amazon.
2. Pilih Buat pelacak.
3. Isi kolom yang diperlukan. Catat Nama pelacak karena akan dibiaskan di seluruh tutoiral ini.
4. Di bawah bidang Pemfilteran posisi, pilih opsi yang paling sesuai dengan cara Anda ingin menggunakan sumber daya pelacak Anda. Jika Anda tidak mengatur Pemfilteran posisi, pengaturan defaultnya adalah TimeBased. Untuk informasi selengkapnya, lihat [Mulai melacak](#), dan [PositionFiltering](#) di Referensi API Lokasi Amazon.
5. Pilih Buat pelacak untuk menyelesaikan pembuatan pelacak.

## Buat Koleksi Geofence

Sekarang Anda akan membuat koleksi geofence. Anda dapat menggunakan konsol, API atau CLI. Prosedur berikut memandu Anda melalui setiap opsi.

- Buat koleksi geofence menggunakan konsol Lokasi Amazon:
  1. Buka bagian [Koleksi Geofence](#) di konsol Lokasi Amazon.
  2. Pilih Buat koleksi geofence.
  3. Berikan nama dan deskripsi untuk koleksi.
  4. Di bawah EventBridge aturan dengan Amazon CloudWatch sebagai target, Anda dapat membuat EventBridge aturan opsional untuk mulai bereaksi terhadap peristiwa geofence. Hal ini memungkinkan Amazon Location untuk mempublikasikan acara ke Amazon CloudWatch Logs.

5. Tekan Create geofence collection untuk menyelesaikan pembuatan koleksi.
- Membuat koleksi geofence menggunakan Amazon Location API:

Gunakan [CreateGeofenceCollection](#) operasi dari Amazon Location Geofences API. Contoh berikut menggunakan permintaan API untuk membuat koleksi geofence yang disebut.

***GEOCOLLECTION\_NAME***

```
POST /geofencing/v0/collections
Content-type: application/json

{
  "CollectionName": "GEOCOLLECTION_NAME",
  "Description": "Geofence collection 1 for shopping center",
  "Tags": {
    "Tag1" : "Value1"
  }
}
```

- Buat koleksi geofence menggunakan perintah CLI AWS :

Gunakan perintah `create-geofence-collection`. Contoh berikut menggunakan AWS CLI untuk membuat koleksi geofence yang disebut. ***GEOCOLLECTION\_NAME*** Untuk informasi selengkapnya tentang penggunaan AWS CLI, lihat Dokumentasi [Antarmuka Baris AWS Perintah](#).

```
aws location \
  create-geofence-collection \
  --collection-name "ExampleGeofenceCollection" \
  --description "Shopping center geofence collection" \
  --tags Tag1=Value1
```

## Tautkan pelacak ke koleksi geofence

Untuk menautkan pelacak ke koleksi geofence, Anda dapat menggunakan konsol, API, atau CLI. Prosedur berikut memandu Anda melalui setiap opsi.

Menautkan sumber daya pelacak ke koleksi geofence menggunakan konsol Amazon Location Service:

1. Buka konsol Lokasi Amazon.
2. Di panel navigasi kiri, pilih Pelacak.

3. Di bawah Pelacak Perangkat, pilih tautan nama pelacak target.
4. Di bawah Koleksi Geofence Tertaut, pilih Link Geofence Collection.
5. Di jendela Linked Geofence Collection, pilih koleksi geofence dari menu tarik-turun.
6. Pilih Tautkan.
7. Setelah Anda menautkan sumber daya pelacak, itu akan diberi status Aktif.

Menautkan resource pelacak ke koleksi geofence menggunakan Amazon Location API:

Gunakan `AssociateTrackerConsumer` operasi dari Amazon Location Trackers API. Contoh berikut menggunakan permintaan API yang terkait `ExampleTracker` dengan koleksi geofence menggunakan Amazon Resource Name (ARN).

```
POST /tracking/v0/trackers/ExampleTracker/consumers
Content-type: application/json
{
  "ConsumerArn": "arn:aws:geo:us-west-2:123456789012:geofence-
collection/GEOCOLLECTION_NAME"
}
```

Llink sumber daya pelacak ke koleksi geofence menggunakan perintah AWS CLI:

Gunakan perintah `associate-tracker-consumer` . Contoh berikut menggunakan AWS CLI untuk membuat koleksi geofence yang disebut. *GEOCOLLECTION\_NAME*

```
aws location \
  associate-tracker-consumer \
    --consumer-arn "arn:aws:geo:us-west-2:123456789012:geofence-
collection/GEOCOLLECTION_NAME" \
    --tracker-name "ExampleTracker"
```

## Menggunakan AWS Lambda dengan MQTT

Untuk membuat koneksi antara AWS IoT dan Lokasi Amazon, Anda memerlukan fungsi Lambda untuk memproses pesan yang diteruskan oleh peristiwa. EventBridge CloudWatch Fungsi ini akan mengekstrak data posisi apa pun, memformatnya untuk Lokasi Amazon, dan mengirimkannya melalui Amazon Location Tracker API.

Prosedur berikut menunjukkan cara membuat fungsi ini melalui konsol Lambda:

1. Buka [konsol](#) .
2. Dari navigasi kiri, pilih Fungsi.
3. Kemudian pilih Create Function, dan pastikan bahwa opsi Author from scratch dipilih.
4. berikan nama Function, dan untuk opsi Runtime, pilih Node.js 16.x.
5. Pilih Buat fungsi.
6. Buka tab Kode untuk mengakses editor.
7. Timpa kode placeholder dalam `index.js` file dengan yang berikut:

```
const AWS = require('aws-sdk')
const iot = new AWS.Iot();
exports.handler = function(event) {
  console.log("event===>>>", JSON.stringify(event));
  var param = {
    endpointType: "iot:Data-ATS"
  };
  iot.describeEndpoint(param, function(err, data) {
    if (err) {
      console.log("error===>>>", err, err.stack); // an error occurred
    } else {
      var endp = data['endpointAddress'];
      const iotdata = new AWS.IotData({endpoint: endp});
      const trackerEvent = event["detail"]["EventType"];
      const src = event["source"];
      const time = event["time"];
      const gfId = event["detail"]["GeofenceId"];
      const resources = event["resources"][0];
      const splitResources = resources.split(".");
      const geofenceCollection = splitResources[splitResources.length -
1];

      const coordinates = event["detail"]["Position"];

      const deviceId = event["detail"]["DeviceId"];
      console.log("deviceId===>>>", deviceId);
      const msg = {
        "trackerEventType" : trackerEvent,
        "source" : src,
        "eventTime" : time,
        "geofenceId" : gfId,
        "coordinates": coordinates,
        "geofenceCollection": geofenceCollection
```



```
    };
    const params = {
      topic: `${deviceId}/tracker`,
      payload: JSON.stringify(msg),
      qos: 0
    };
    iotdata.publish(params, function(err, data) {
      if (err) {
        console.log("error====>>>", err, err.stack); // an error
        occurred
      } else {
        console.log("Ladmbda triggered====>>>", trackerEvent); //
        successful response
      }
    });
  }
});
}
```

8. Tekan tombol Deploy untuk menyimpan fungsi yang diperbarui.
9. Selanjutnya buka tab Konfigurasi.
10. Di bagian Pemicu, tekan tombol Add Trigger.
11. Pilih EventBridge (CloudWatch Acara) di bidang Sumber.
12. Pilih opsi Aturan yang Ada.
13. Masukkan nama aturan, misalnya AmazonLocationMonitor-GEOFENCECOLLECTION\_NAME.
14. Tekan tombol Tambah.
15. Ini juga akan melampirkan pernyataan kebijakan berbasis Sumber Daya di tab izin

Sekarang Anda akan mengatur AWS IoT MQTT Test Client, gunakan prosedur berikut:

1. Buka <https://console.aws.amazon.com/iot/>.
2. Di panel navigasi kiri, pilih klien pengujian MQTT.
3. Anda akan melihat bagian berjudul MQTT test client di mana Anda dapat mengonfigurasi koneksi MQTT Anda.
4. Setelah mengkonfigurasi pengaturan yang diperlukan, klik tombol Connect untuk membuat koneksi ke broker MQTT menggunakan parameter yang disediakan.
5. Catat titik akhir, seperti yang digunakan nanti di tutorial.

Setelah terhubung ke klien pengujian, Anda dapat berlangganan topik MQTT atau mempublikasikan pesan ke topik menggunakan bidang masukan masing-masing yang disediakan di antarmuka klien pengujian MQTT. Selanjutnya Anda akan membuat AWS IoT kebijakan.

6. Di menu sebelah kiri, di bawah opsi Kelola perluas Keamanan dan klik Kebijakan.
7. Klik tombol Buat Kebijakan.
8. Masukkan nama kebijakan.
9. Pada Dokumen Kebijakan pilih tab JSON.
10. Salin tempel kebijakan yang ditunjukkan di bawah ini, tetapi pastikan untuk memperbarui semua elemen dengan Anda *REGION* dan *ACCOUNT\_ID*:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "iot:Connect",
        "iot:Publish",
        "iot:Subscribe",
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:REGION:ACCOUNT_ID:client/${cognito-identity.amazonaws.com:sub}",
        "arn:aws:iot:REGION:ACCOUNT_ID:topic/${cognito-identity.amazonaws.com:sub}",
        "arn:aws:iot:REGION:ACCOUNT_ID:topicfilter/${cognito-identity.amazonaws.com:sub}/*",
        "arn:aws:iot:REGION:ACCOUNT_ID:topic/${cognito-identity.amazonaws.com:sub}/tracker"
      ],
      "Effect": "Allow"
    }
  ]
}
```

11. Pilih tombol Create untuk menyelesaikan.

## Menyiapkan contoh kode aplikasi

Untuk mengatur kode sampel, Anda harus menginstal alat-alat berikut:

- Git
- XCode 15.3 atau Lebih Baru
- iOS Simulator 16 atau lebih baru

Gunakan prosedur ini untuk menyiapkan contoh kode aplikasi:

1. [Kloning repositori git dari URL ini: https://github.com/aws-geospatial/ /tree/main/ amazon-location-samples-ios tracking-with-geofence-notifications](https://github.com/aws-geospatial/tree/main/amazon-location-samples-ios-tracking-with-geofence-notifications)
2. Buka file `AWSLocationSampleApp.xcodeproj` proyek.
3. Tunggu hingga proses resolusi paket selesai.
4. Pada menu navigasi proyek ganti nama `ConfigTemplate.xcconfig` menjadi `Config.xcconfig` dan isi nilai-nilai berikut:

```
IDENTITY_POOL_ID = `YOUR_IDENTITY_POOL_ID`  
MAP_NAME = `YOUR_MAP_NAME`  
TRACKER_NAME = `YOUR_TRACKER_NAME`  
WEBSOCKET_URL = `YOUR_MQTT_TEST_CLIENT_ENDPOINT`  
GEOFENCE_ARN = `YOUR_GEOFENCE_COLLECTION_NAME`
```

## Menggunakan aplikasi sampel

Setelah menyiapkan kode sampel, Anda sekarang dapat menjalankan aplikasi di simulator iOS atau perangkat fisik.

1. Membangun dan menjalankan aplikasi.
2. Aplikasi akan meminta Anda untuk izin lokasi dan pemberitahuan. Anda harus mengizinkan mereka.
3. Tekan tombol Konfigurasi Cognito.
4. Simpan konfigurasi.
5. Anda sekarang dapat melihat opsi Filter untuk waktu, jarak, dan akurasi. Gunakan sesuai kebutuhan Anda.
6. Buka tab Tracking di aplikasi dan Anda akan melihat peta dan tombol Start Tracking.

7. Jika Anda telah menginstal aplikasi di simulator, Anda mungkin ingin mensimulasikan perubahan lokasi. Ini dapat dilakukan di Fitur di bawah opsi menu Lokasi. Misalnya pilih Fitur, lalu Lokasi, lalu Freeway Drive.
8. Tekan tombol Mulai Pelacakan. Anda akan melihat titik pelacakan di peta.
9. Aplikasi ini juga melacak lokasi di latar belakang. Jadi, ketika Anda memindahkan aplikasi di latar belakang, aplikasi akan meminta izin Anda untuk melanjutkan pelacakan dalam mode latar belakang.
10. Anda dapat menghentikan pelacakan dengan mengetuk tombol Stop Tracking.

## Menandai sumber daya Amazon Location Service

Gunakan penandaan sumber daya di Lokasi Amazon untuk membuat tag untuk mengkategorikan sumber daya Anda berdasarkan tujuan, pemilik, lingkungan, atau kriteria. Menandai sumber daya membantu Anda mengelola, mengidentifikasi, mengatur, mencari, dan memfilter sumber daya Anda.

Misalnya, dengan AWS Resource Groups, Anda dapat membuat grup AWS sumber daya berdasarkan satu atau beberapa tag atau bagian tag. Anda juga dapat membuat grup berdasarkan kemunculannya di tumpukan AWS CloudFormation. Dengan menggunakan Resource Groups dan Tag Editor, Anda dapat menggabungkan dan melihat data untuk aplikasi yang terdiri dari beberapa layanan, sumber daya, dan Wilayah di satu tempat. Untuk informasi selengkapnya tentang [Strategi Penandaan Umum](#), lihat AWS General Reference.

Setiap tag adalah label yang terdiri dari kunci dan nilai yang Anda tentukan:

- Tag key — Label umum yang mengkategorikan nilai tag. Misalnya, `CostCenter`.
- Nilai tag - Deskripsi opsional untuk kategori kunci tag. Misalnya, `MobileAssetTrackingResourcesProd`.

Topik ini membantu Anda memulai penandaan dengan meninjau batasan penandaan. Ini juga menunjukkan cara membuat tag dan menggunakan tag untuk melacak AWS biaya Anda untuk setiap tag aktif dengan menggunakan laporan alokasi biaya.

Topik

- [Pembatasan penandaan](#)
- [Berikan izin untuk menandai sumber daya](#)

- [Menambahkan tag ke sumber daya Amazon Location Service](#)
- [Lacak biaya sumber daya berdasarkan tag](#)
- [Kontrol akses ke sumber daya Amazon Location Service menggunakan tag](#)
- [Pelajari selengkapnya](#)

## Pembatasan penandaan

Batasan dasar berikut berlaku untuk tanda:

- Tag maksimum per sumber daya - 50
- Untuk setiap sumber daya, setiap kunci tanda harus unik, dan setiap kunci tanda hanya dapat memiliki satu nilai.

### Note

Jika Anda menambahkan tanda baru yang memiliki kunci tanda yang sama dengan tanda yang sudah ada, tanda yang baru akan menimpa tanda yang sudah ada.

- Panjang kunci maksimum – 128 karakter Unicode dalam UTF-8
- Panjang nilai maksimum – 256 karakter Unicode dalam UTF-8
- Karakter yang diperbolehkan pada layanan adalah huruf, angka, dan spasi yang dapat mewakili dalam UTF-8, serta karakter berikut: + - = . \_ : / @.
- Kunci dan nilai tag peka huruf besar dan kecil.
- Prefiks `aws :` disimpan untuk penggunaan AWS. Jika tanda memiliki kunci tanda dengan prefiks ini, Anda tidak dapat mengedit atau menghapus kunci atau nilai tanda tersebut. Tag dengan `aws :` awalan tidak dihitung terhadap tag Anda per batas sumber daya.

## Berikan izin untuk menandai sumber daya

Anda dapat menggunakan kebijakan IAM untuk mengontrol akses ke sumber daya Lokasi Amazon dan memberikan izin untuk menandai sumber daya saat pembuatan. Selain memberikan izin untuk membuat sumber daya, kebijakan dapat menyertakan `Action` izin untuk mengizinkan operasi penandaan:

- `geo:TagResource`— Memungkinkan pengguna untuk menetapkan satu atau beberapa tag ke sumber daya Lokasi Amazon tertentu.

- `geo:UntagResource`— Memungkinkan pengguna untuk menghapus satu atau beberapa tag dari sumber daya Lokasi Amazon tertentu.
- `geo:ListTagsForResource`— Memungkinkan pengguna untuk mencantumkan semua tag yang ditetapkan ke sumber daya Lokasi Amazon.

Berikut ini adalah contoh kebijakan untuk memungkinkan pengguna membuat koleksi geofence dan tag resource:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowTaggingForGeofenceCollectionOnCreation",
      "Effect": "Allow",
      "Action": [
        "geo:CreateGeofenceCollection",
        "geo:TagResource"
      ],
      "Resource": "arn:aws:geo:region:accountID:geofence-collection/*"
    }
  ]
}
```

## Menambahkan tag ke sumber daya Amazon Location Service

Anda dapat menambahkan tag saat membuat sumber daya menggunakan konsol Lokasi Amazon, API Lokasi AmazonAWS CLI, atau Amazon:

- [Buat sumber daya peta](#)
- [Buat sumber daya indeks tempat](#)
- [Buat sumber daya kalkulator rute](#)
- [Buat koleksi geofence](#)
- [Buat sumber daya pelacak](#)

Untuk menandai sumber daya yang ada, edit atau hapus tag

1. Buka konsol Lokasi Amazon di <https://console.aws.amazon.com/location/>.
2. Di panel navigasi kiri, pilih sumber daya yang ingin Anda tag. Misalnya, Maps.

3. Pilih sumber daya dari daftar.
4. Pilih Kelola tag untuk menambah, mengedit, atau menghapus tag Anda.

## Lacak biaya sumber daya berdasarkan tag

Anda dapat menggunakan tag untuk alokasi biaya untuk melacak AWS biaya Anda secara rinci. Setelah Anda mengaktifkan tag alokasi biaya, AWS gunakan tag alokasi biaya untuk mengatur penagihan sumber daya pada laporan alokasi biaya Anda. Ini membantu Anda mengkategorikan dan melacak biaya penggunaan Anda.

Ada dua jenis tag alokasi biaya yang dapat Anda aktifkan:

- [AWS-generated](#) — Tag ini dihasilkan oleh AWS. AWStag menggunakan `aws :` awalan, misalnya, `aws : createdBy`.
- [Ditentukan pengguna](#) - Ini adalah tag khusus yang Anda buat. Tag yang ditentukan pengguna menggunakan `user :` awalan, misalnya, `user : CostCenter`

Anda harus mengaktifkan setiap jenis tag satu per satu. Setelah tag diaktifkan, Anda dapat [mengaktifkan AWS Cost Explorer](#) atau melihat laporan alokasi biaya bulanan Anda.

### AWS-generated tags

Untuk mengaktifkan tag yang dihasilkan AWS

1. Buka konsol Manajemen Penagihan dan Biaya di <https://console.aws.amazon.com/billing/>.
2. Di panel navigasi kiri, pilih Tag Alokasi Biaya.
3. Di bawah tab Tag Alokasi Biaya yang AWS Dihasilkan, pilih tombol tag yang ingin Anda aktifkan.
4. Pilih Aktifkan.

### User-defined tags

Untuk mengaktifkan tag yang ditentukan pengguna

1. Buka konsol Manajemen Penagihan dan Biaya di <https://console.aws.amazon.com/billing/>.
2. Di panel navigasi kiri, pilih Tag Alokasi Biaya.

3. Di bawah tab Tag Alokasi Biaya yang Ditentukan Pengguna, pilih kunci tag yang ingin Anda aktifkan.
4. Pilih Aktifkan.

Setelah Anda mengaktifkan tag, AWS buat [Laporan Alokasi Biaya bulanan](#) untuk penggunaan dan biaya sumber daya Anda. Laporan alokasi biaya ini mencakup semua AWS biaya Anda untuk setiap periode penagihan, termasuk sumber daya yang ditandai dan tidak ditandai. Untuk informasi selengkapnya, lihat [Menggunakan Tag Alokasi Biaya](#) dalam Panduan Pengguna AWS Billing and Cost Management.

## Kontrol akses ke sumber daya Amazon Location Service menggunakan tag

AWS Identity and Access Management (IAM) mendukung kondisi berbasis tag, yang memungkinkan Anda mengelola otorisasi sumber daya berdasarkan kunci dan nilai tag tertentu. Misalnya, kebijakan peran IAM dapat mencakup kondisi untuk membatasi akses ke lingkungan tertentu, seperti pengembangan, pengujian, atau produksi, berdasarkan tag.

Untuk informasi selengkapnya, lihat topik tentang [kontrol akses sumber daya berdasarkan tag](#).

## Pelajari selengkapnya

Untuk informasi lebih lanjut tentang:

- Menandai praktik terbaik, lihat [Menandai sumber daya AWS di Referensi Umum AWS](#).
- Menggunakan tag untuk mengontrol akses ke AWS sumber daya, lihat [Mengontrol akses ke AWS sumber daya menggunakan tag](#) di Panduan AWS Identity and Access Management Pengguna.

## Memberikan akses ke Amazon Location Service

Untuk menggunakan Amazon Location Service, pengguna harus diberikan akses ke sumber daya dan API yang membentuk Amazon Location. Ada tiga strategi yang dapat Anda gunakan untuk memberikan akses ke sumber daya Anda.

- Gunakan IAM — Untuk memberikan akses ke pengguna yang diautentikasi dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM), buat kebijakan IAM yang memungkinkan akses ke sumber daya yang Anda inginkan. Untuk informasi selengkapnya tentang IAM dan Lokasi Amazon, lihat [Identity and Access Management untuk Amazon Location Service](#).



- Gunakan kunci API — Untuk memberikan akses ke pengguna yang tidak diautentikasi, Anda dapat membuat Kunci API yang memberikan akses hanya-baca ke sumber daya Amazon Location Service Anda. Ini berguna dalam kasus di mana Anda tidak ingin mengautentikasi setiap pengguna. Misalnya, aplikasi web. Untuk informasi selengkapnya tentang kunci API, lihat [Mengizinkan akses tamu yang tidak diautentikasi ke aplikasi Anda menggunakan kunci API](#).
- Gunakan Amazon Cognito — Alternatif untuk kunci API adalah dengan menggunakan Amazon Cognito untuk memberikan akses anonim. Amazon Cognito memungkinkan Anda membuat otorisasi yang lebih kaya dengan kebijakan untuk menentukan apa yang dapat dilakukan oleh pengguna yang tidak diautentikasi. Untuk informasi selengkapnya tentang menggunakan Amazon Cognito, lihat [Mengizinkan akses tamu yang tidak diautentikasi ke aplikasi Anda menggunakan Amazon Cognito](#)

#### Note

Anda juga dapat menggunakan Amazon Cognito untuk menggunakan proses autentikasi Anda sendiri atau untuk menggabungkan beberapa metode otentikasi, menggunakan Identitas Federasi Amazon Cognito. Untuk informasi selengkapnya, lihat [Memulai Identitas Federasi di Panduan](#) Pengembang Amazon Cognito.

#### Topik

- [Mengizinkan akses tamu yang tidak diautentikasi ke aplikasi Anda menggunakan kunci API](#)
- [Mengizinkan akses tamu yang tidak diautentikasi ke aplikasi Anda menggunakan Amazon Cognito](#)

## Mengizinkan akses tamu yang tidak diautentikasi ke aplikasi Anda menggunakan kunci API

Saat memanggil Amazon Location Service API di aplikasi, Anda biasanya membuat panggilan ini sebagai pengguna yang diautentikasi yang diberi wewenang untuk melakukan panggilan API. Namun, ada beberapa kasus di mana Anda tidak ingin mengautentikasi setiap pengguna aplikasi Anda. Misalnya, Anda mungkin ingin aplikasi web yang menunjukkan lokasi bisnis Anda tersedia bagi siapa saja yang menggunakan situs web, apakah mereka masuk atau tidak. Dalam hal ini, salah satu alternatifnya adalah menggunakan kunci API untuk melakukan panggilan API.

Kunci API adalah nilai kunci yang dikaitkan dengan sumber daya Amazon Location Service tertentu di Anda Akun AWS, dan tindakan spesifik yang dapat Anda lakukan pada sumber daya tersebut.

Anda dapat menggunakan kunci API dalam aplikasi Anda untuk membuat panggilan yang tidak diautentikasi ke Amazon Location API untuk sumber daya tersebut. Misalnya, jika Anda mengaitkan kunci API dengan sumber daya peta MyMap, dan GetMap\* tindakannya, maka aplikasi yang menggunakan kunci API tersebut akan dapat melihat peta yang dibuat dengan sumber daya tersebut, dan akun Anda akan dikenakan biaya sebagai penggunaan lain dari akun Anda. Kunci API yang sama tidak akan memberikan izin untuk mengubah atau memperbarui sumber daya peta—hanya menggunakan sumber daya yang diizinkan.

### Note

Kunci API hanya tersedia untuk digunakan dengan peta, tempat, dan sumber daya rute, dan Anda tidak dapat memodifikasi atau membuat sumber daya tersebut. Jika aplikasi Anda memerlukan akses ke sumber daya atau tindakan lain untuk pengguna yang tidak diautentikasi, Anda dapat menggunakan Amazon Cognito untuk menyediakan akses bersama dengan, atau bukan, kunci API. Untuk informasi selengkapnya, lihat [Mengizinkan akses tamu yang tidak diautentikasi ke aplikasi Anda menggunakan Amazon Cognito](#).

Kunci API menyertakan nilai teks biasa yang memberikan akses ke satu atau beberapa sumber daya di Anda. Akun AWSJika seseorang menyalin kunci API Anda, mereka dapat mengakses sumber daya yang sama. Untuk menghindari hal ini, Anda dapat menentukan domain tempat kunci API dapat digunakan saat Anda membuat kunci. Domain ini disebut referer. Jika diperlukan, Anda juga dapat membuat Kunci API jangka pendek dengan menetapkan waktu kedaluwarsa untuk Kunci API Anda.

### Topik

- [Kunci API dibandingkan dengan Amazon Cognito](#)
- [Membuat kunci API](#)
- [Menggunakan kunci API untuk memanggil Amazon Location API](#)
- [Menggunakan kunci API untuk merender peta](#)
- [Mengelola masa pakai kunci API](#)

## Kunci API dibandingkan dengan Amazon Cognito

Kunci API dan Amazon Cognito digunakan dengan cara yang sama untuk skenario serupa, jadi mengapa Anda menggunakan satu di atas yang lain? Daftar berikut menyoroti beberapa perbedaan antara keduanya.

- Kunci API hanya tersedia untuk peta, tempat, dan sumber daya rute, dan hanya untuk tindakan tertentu. Amazon Cognito dapat digunakan untuk mengautentikasi akses ke sebagian besar Amazon Location Service API.
- Performa permintaan peta dengan kunci API biasanya lebih cepat daripada skenario serupa dengan Amazon Cognito. Otentikasi yang lebih sederhana berarti lebih sedikit perjalanan pulang pergi ke layanan dan permintaan cache saat mendapatkan ubin peta yang sama lagi dalam periode waktu singkat.
- Dengan Amazon Cognito, Anda dapat menggunakan proses otentikasi Anda sendiri atau menggabungkan beberapa metode otentikasi, menggunakan Identitas Federasi Amazon Cognito. Untuk informasi selengkapnya, lihat [Memulai Identitas Federasi di Panduan Pengembang Amazon Cognito](#).

## Membuat kunci API

Anda dapat membuat kunci API, dan mengaitkannya dengan satu atau beberapa sumber daya di Akun AWS.

Anda dapat membuat kunci API menggunakan konsol Amazon Location Service, Amazon Location API AWS CLI, atau Amazon Location.

### Console

Untuk membuat kunci API menggunakan konsol Amazon Location Service

1. Di [konsol Lokasi Amazon](#), pilih kunci API dari menu sebelah kiri.
2. Pada halaman kunci API, pilih Create API key.
3. Pada halaman Create API key, isi informasi berikut:
  - Nama - Nama untuk kunci API Anda, seperti MyWebAppKey.
  - Deskripsi — Deskripsi opsional untuk kunci API Anda.
  - Sumber Daya — Pilih sumber daya Lokasi Amazon untuk memberikan akses dengan kunci API ini dari dropdown. Anda dapat menambahkan lebih dari satu sumber daya dengan memilih Tambah sumber daya.
  - Tindakan - Tentukan tindakan yang ingin Anda otorisasi dengan kunci API ini. Anda harus memilih setidaknya satu tindakan untuk mencocokkan setiap jenis sumber daya yang telah Anda pilih. Misalnya, jika Anda memilih sumber daya tempat, Anda harus memilih setidaknya salah satu pilihan di bawah Tindakan Tempat.

- Waktu kedaluwarsa - Secara opsional, tambahkan tanggal kedaluwarsa dan waktu untuk kunci API Anda. Untuk informasi selengkapnya, lihat [Mengelola masa pakai kunci API](#).
  - Referer — Secara opsional, tambahkan satu atau beberapa domain tempat Anda dapat menggunakan kunci API. Misalnya, jika kunci API adalah untuk mengizinkan aplikasi berjalan di situs `webexample.com`, maka Anda dapat menempatkan `*.example.com/` sebagai perujuk yang diizinkan.
  - Tag - Secara opsional, tambahkan tag ke kunci API.
4. Pilih Buat kunci API untuk membuat kunci API.
  5. Pada halaman detail untuk kunci API, Anda dapat melihat informasi tentang kunci API yang telah Anda buat. Pilih Tampilkan kunci API untuk melihat nilai kunci yang Anda gunakan saat memanggil Amazon Location API. Nilai kunci akan memiliki format `v1.public.a1b2c3d4...` Untuk informasi selengkapnya tentang penggunaan kunci API untuk merender peta, lihat [Menggunakan kunci API untuk merender peta](#).

## API

Untuk membuat kunci API menggunakan Amazon Location API

Gunakan [CreateKey](#) operasi dari Amazon Location API.

Contoh berikut adalah permintaan API untuk membuat kunci API yang dipanggil tanpa tanggal *ExampleKey* kedaluwarsa, dan akses ke sumber daya peta tunggal.

```
POST /metadata/v0/keys HTTP/1.1
Content-type: application/json

{
  "KeyName": "ExampleKey"
  "Restrictions": {
    "AllowActions": [
      "geo:GetMap*"
    ],
    "AllowResources": [
      "arn:aws:geo:region:map/mapname"
    ]
  },
  "NoExpiry": true
}
```

Responsnya mencakup nilai kunci API yang akan digunakan saat mengakses sumber daya di aplikasi Anda. Nilai kunci akan memiliki format `v1.public.a1b2c3d4...`. Untuk mempelajari selengkapnya tentang menggunakan kunci API untuk merender peta, lihat [Menggunakan kunci API untuk merender peta](#).

Anda juga dapat menggunakan [DescribeKey](#) API untuk menemukan nilai kunci untuk kunci di lain waktu.

## AWS CLI

Untuk membuat kunci API menggunakan AWS CLI perintah

Gunakan perintah [create-key](#).

Contoh berikut membuat kunci API yang dipanggil tanpa tanggal *ExampleKey* kedaluwarsa, dan akses ke sumber daya peta tunggal.

```
aws location \
  create-key \
  --key-name ExampleKey \
  --restrictions '{"AllowActions":["geo:GetMap*"],"AllowResources":
["arn:aws:geo:region:map/mapname"]}' \
  --no-expiry
```

Responsnya mencakup nilai kunci API yang akan digunakan saat mengakses sumber daya di aplikasi Anda. Nilai kunci akan memiliki format `v1.public.a1b2c3d4...`. Untuk mempelajari selengkapnya tentang menggunakan kunci API untuk merender peta, lihat [Menggunakan kunci API untuk merender peta](#). `create-key` Responsnya terlihat seperti berikut ini.

```
{
  "Key": "v1.public.a1b2c3d4...",
  "KeyArn": "arn:aws:geo:region:accountId:api-key/ExampleKey",
  "KeyName": "ExampleKey",
  "CreateTime": "2023-02-06T22:33:15.693Z"
}
```

Anda juga dapat menggunakan `describe-key` untuk menemukan nilai kunci di lain waktu. Contoh berikut menunjukkan cara memanggil `describe-key` kunci API bernama *ExampleKey*.

```
aws location describe-key \
  --key-name ExampleKey
```

## Menggunakan kunci API untuk memanggil Amazon Location API

Setelah membuat kunci API, Anda dapat menggunakan nilai kunci untuk melakukan panggilan ke Amazon Location API di aplikasi Anda.

API yang mendukung kunci API memiliki parameter tambahan yang mengambil nilai kunci API. Misalnya, jika Anda memanggil GetPlace API, Anda dapat mengisi parameter [kunci](#), sebagai berikut

```
GET /places/v0/indexes/IndexName/places/PlaceId?key=KeyVaLue
```

Jika Anda mengisi nilai ini, Anda tidak perlu mengautentikasi panggilan API dengan AWS Sig v4 seperti biasa.

Untuk JavaScript pengembang, Anda dapat menggunakan Lokasi Amazon [JavaScript Pembantu otentikasi](#) untuk membantu mengautentikasi operasi API dengan kunci API.

Untuk pengembang seluler, Anda dapat menggunakan SDK otentikasi seluler Lokasi Amazon berikut:

- [Amazon Location Service Mobile Authentication SDK for iOS](#)
- [Amazon Location Service Mobile Authentication SDK for Android](#)

Untuk AWS CLI pengguna, ketika Anda menggunakan `--key` parameter, Anda juga harus menggunakan `--no-sign-request` parameter, untuk menghindari penandatanganan dengan Sig v4.

### Note

Jika Anda menyertakan tanda tangan key dan dan AWS Sig v4 dalam panggilan ke Amazon Location Service, hanya kunci API yang digunakan.

## Menggunakan kunci API untuk merender peta

Anda dapat menggunakan nilai kunci API untuk merender peta dalam aplikasi Anda menggunakan MapLibre. Ini sedikit berbeda dengan menggunakan kunci API di API Lokasi Amazon lainnya yang Anda panggil secara langsung, karena MapLibre membuat panggilan tersebut untuk Anda.

Contoh kode berikut menunjukkan penggunaan kunci API untuk merender peta di halaman web sederhana dengan menggunakan kontrol peta MapLibre GL JS. Agar kode ini berfungsi dengan baik,

ganti *v1.public.your-api-key-value*, *us-east-1*, dan *ExampleMap*string dengan nilai yang cocok dengan Anda. Akun AWS

```
<!-- index.html -->
<html>
  <head>
    <link href="https://unpkg.com/maplibre-gl@1.14.0/dist/maplibre-gl.css"
rel="stylesheet" />
    <style>
      body { margin: 0; }
      #map { height: 100vh; }
    </style>
  </head>
  <body>
    <!-- Map container -->
    <div id="map" />
    <!-- JavaScript dependencies -->
    <script src="https://unpkg.com/maplibre-gl@1.14.0/dist/maplibre-gl.js"></script>
    <script>
      const apiKey = "v1.public.your-api-key-value"; // API key
      const region = "us-east-1"; // Region
      const mapName = "ExampleMap"; // Map name
      // URL for style descriptor
      const styleUrl = `https://maps.geo.${region}.amazonaws.com/maps/v0/maps/
${mapName}/style-descriptor?key=${apiKey}`;
      // Initialize the map
      const map = new maplibregl.Map({
        container: "map",
        style: styleUrl,
        center: [-123.1187, 49.2819],
        zoom: 11,
      });
      map.addControl(new maplibregl.NavigationControl(), "top-left");
    </script>
  </body>
</html>
```

## Mengelola masa pakai kunci API

Anda dapat membuat kunci API yang berfungsi tanpa batas waktu. Namun, jika Anda ingin membuat kunci API sementara, memutar kunci API secara teratur, atau mencabut kunci API yang ada, Anda dapat menggunakan kedaluwarsa kunci API.

Saat membuat kunci API baru, atau memperbarui kunci yang sudah ada, Anda dapat mengatur waktu kedaluwarsa untuk kunci API tersebut.

- Ketika kunci API mencapai waktu kedaluwarsa, kunci secara otomatis dinonaktifkan. Kunci tidak aktif tidak lagi dapat digunakan untuk membuat permintaan peta.
- Anda dapat menghapus kunci API 90 hari setelah menonaktifkannya.
- Jika Anda memiliki kunci tidak aktif yang belum dihapus, Anda dapat memulihkannya dengan memperbarui waktu kedaluwarsa ke masa depan.
- Untuk membuat kunci permanen, Anda dapat menghapus waktu kedaluwarsa.
- Jika Anda mencoba menonaktifkan kunci API yang telah digunakan dalam 7 hari terakhir, Anda akan diminta untuk mengonfirmasi bahwa Anda ingin melakukan perubahan. Jika Anda menggunakan Amazon Location Service API, atau AWS CLI, Anda akan menerima error, kecuali Anda menetapkan `ForceUpdate` parameter ke `true`.

## Mengizinkan akses tamu yang tidak diautentikasi ke aplikasi Anda menggunakan Amazon Cognito

Anda dapat menggunakan otentikasi Amazon Cognito sebagai alternatif untuk langsung menggunakan AWS Identity and Access Management (IAM) dengan SDK frontend dan permintaan HTTPS langsung.

Anda mungkin ingin menggunakan bentuk otentikasi ini karena alasan berikut:

- Pengguna yang tidak diautentikasi — Jika Anda memiliki situs web dengan pengguna anonim, Anda dapat menggunakan kumpulan identitas Amazon Cognito. Untuk informasi lebih lanjut, lihat bagian [dithe section called “Menggunakan Amazon Cognito”](#).
- Otentikasi Anda sendiri — Jika Anda ingin menggunakan proses otentikasi Anda sendiri, atau menggabungkan beberapa metode otentikasi, Anda dapat menggunakan Identitas Federasi Amazon Cognito. Untuk informasi selengkapnya, lihat [Memulai Identitas Federasi di Panduan Pengembang Amazon Cognito](#).

Amazon Cognito menyediakan otentikasi, otorisasi, dan manajemen pengguna untuk aplikasi web dan seluler. Anda dapat menggunakan kumpulan identitas Amazon Cognito yang tidak diautentikasi dengan Lokasi Amazon sebagai cara bagi aplikasi untuk mengambil kredensial sementara yang tercakup ke bawah. AWS



Untuk informasi selengkapnya, lihat [Memulai Kumpulan Pengguna](#) di Panduan Pengembang Amazon Cognito.

#### Note

Untuk pengembang seluler, Amazon Location menyediakan SDK otentikasi seluler untuk iOS dan Android, lihat repositori github berikut untuk informasi selengkapnya:

- [Amazon Location Service Mobile Authentication SDK for iOS](#)
- [Amazon Location Service Mobile Authentication SDK for Android](#)

## Buat kumpulan identitas Amazon Cognito

Anda dapat membuat kumpulan identitas Amazon Cognito untuk mengizinkan akses tamu yang tidak diautentikasi ke aplikasi Anda melalui konsol Amazon Cognito, API, atau AWS CLI Amazon Cognito.

#### Important

Kumpulan yang Anda buat harus sama Akun AWS dan AWS Wilayah sebagai sumber daya Amazon Location Service yang Anda gunakan.

Anda dapat menggunakan kebijakan IAM yang terkait dengan peran identitas yang tidak diautentikasi dengan tindakan berikut:

- `geo:GetMap*`
- `geo:SearchPlaceIndex*`
- `geo:GetPlace`
- `geo:CalculateRoute*`
- `geo:GetGeofence`
- `geo:ListGeofences`
- `geo:PutGeofence`
- `geo:BatchDeleteGeofence`
- `geo:BatchPutGeofence`
- `geo:BatchEvaluateGeofences`

- `geo:GetDevicePosition*`
- `geo:ListDevicePositions`
- `geo:BatchDeleteDevicePositionHistory`
- `geo:BatchGetDevicePosition`
- `geo:BatchUpdateDevicePosition`

Menyertakan tindakan Lokasi Amazon lainnya tidak akan berpengaruh, dan identitas yang tidak diautentikasi tidak akan dapat memanggilnya.

## Example

Untuk membuat kumpulan identitas menggunakan konsol Amazon Cognito

1. Masuk ke [Konsol Amazon Cognito](#).
2. Pilih Kelola Kolam Identitas.
3. Pilih Buat kumpulan identitas baru, lalu masukkan nama untuk kumpulan identitas Anda.
4. Dari bagian Identitas tidak diautentikasi yang dapat dilipat, pilih Aktifkan akses ke identitas yang tidak diautentikasi.
5. Pilih Buat kolam.
6. Pilih peran IAM mana yang ingin Anda gunakan dengan kumpulan identitas Anda.
7. Perluas Lihat Detail.
8. Di bawah Identitas tidak diautentikasi, masukkan nama peran.
9. Perluas bagian Lihat Dokumen Kebijakan, lalu pilih Edit untuk menambahkan kebijakan Anda.
10. Tambahkan kebijakan Anda untuk memberikan akses ke sumber daya Anda.

Berikut ini adalah contoh kebijakan untuk Peta, Tempat, Pelacak, dan Rute. Untuk menggunakan contoh kebijakan Anda sendiri, ganti placeholder *region* dan *AccountID*:

## Maps policy example

Kebijakan berikut memberikan akses hanya-baca ke sumber daya peta bernama.

### *ExampleMap*

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

```

{
  "Sid": "MapsReadOnly",
  "Effect": "Allow",
  "Action": [
    "geo:GetMapStyleDescriptor",
    "geo:GetMapGlyphs",
    "geo:GetMapSprites",
    "geo:GetMapTile"
  ],
  "Resource": "arn:aws:geo:region:accountID:map/ExampleMap"
}
]
}

```

Menambahkan [kondisi IAM](#) yang cocok `aws:referrer` memungkinkan Anda membatasi akses browser ke sumber daya Anda ke daftar URL atau awalan URL. Contoh berikut memungkinkan akses ke sumber daya peta yang diberi nama hanya `RasterEsriImagery` dari situs `webexample.com`:

#### Warning

Meskipun `aws:referrer` dapat membatasi akses, itu bukan mekanisme keamanan. Menyertakan nilai header perujuk yang diketahui publik bukanlah sesuatu yang aman. Pihak yang tidak berwenang dapat menggunakan browser yang diubah atau disesuaikan untuk menyediakan nilai `aws:referrer` yang mereka pilih. Akibatnya, tidak `aws:referrer` boleh digunakan untuk mencegah pihak yang tidak berwenang membuat AWS permintaan langsung. Ini ditawarkan untuk memungkinkan pelanggan melindungi konten digital mereka, seperti konten yang disimpan di Amazon S3, agar tidak dirujuk pada pihak ketiga yang tidak berwenang. [Untuk informasi lebih lanjut, lihat:referrer.AWS](#)

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "geo:GetMap*",
      "Resource": "arn:aws:geo:us-west-2:111122223333:map/
RasterEsriImagery",

```

```

        "Condition": {
            "StringLike": {
                "aws:referer": [
                    "https://example.com/*",
                    "https://www.example.com/*"
                ]
            }
        }
    ]
}

```

Jika Anda [menggunakan Tangram](#) untuk menampilkan peta, peta tidak menggunakan deskriptor gaya, mesin terbang, atau sprite yang dikembalikan oleh Maps API. Sebaliknya, itu dikonfigurasi dengan menunjuk ke file.zip yang berisi aturan gaya dan aset yang diperlukan. Kebijakan berikut memberikan akses hanya-baca ke sumber daya peta yang diberi nama *ExampleMap* untuk operasi. `GetMapTile`

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "MapsReadOnly",
      "Effect": "Allow",
      "Action": [
        "geo:GetMapTile"
      ],
      "Resource": "arn:aws:geo:region:accountID:map/ExampleMap"
    }
  ]
}

```

### Places policy example

Kebijakan berikut memberikan akses hanya-baca ke sumber daya indeks tempat yang diberi nama *ExamplePlaceIndex* untuk mencari tempat berdasarkan teks atau posisi.

```

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

```

```

    "Sid": "PlacesReadOnly",
    "Effect": "Allow",
    "Action": [
        "geo:SearchPlaceIndex*",
        "geo:GetPlace"
    ],
    "Resource": "arn:aws:geo:region:accountID:place-index/ExamplePlaceIndex"
  }
]
}

```

Menambahkan [kondisi IAM](#) yang cocok `aws:referer` memungkinkan Anda membatasi akses browser ke sumber daya Anda ke daftar URL atau awalan URL. Contoh berikut menolak akses ke sumber daya indeks tempat yang dinamai *ExamplePlaceIndex* dari semua situs web yang merujuk, kecuali `example.com`.

#### Warning

Meskipun `aws:referer` dapat membatasi akses, itu bukan mekanisme keamanan. Menyertakan nilai header perujuk yang diketahui publik bukanlah sesuatu yang aman. Pihak yang tidak berwenang dapat menggunakan browser yang diubah atau disesuaikan untuk menyediakan nilai `aws:referer` yang mereka pilih. Akibatnya, tidak `aws:referer` boleh digunakan untuk mencegah pihak yang tidak berwenang membuat AWS permintaan langsung. Ini ditawarkan untuk memungkinkan pelanggan melindungi konten digital mereka, seperti konten yang disimpan di Amazon S3, agar tidak dirujuk pada pihak ketiga yang tidak berwenang. [Untuk informasi lebih lanjut, lihat:referer.AWS](#)

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "geo:*",
      "Resource": "arn:aws:geo:us-west-2:111122223333:place-
index/ExamplePlaceIndex",
      "Condition": {
        "StringLike": {
          "aws:referer": [

```

```

        "https://example.com/*",
        "https://www.example.com/*"
    ]
  }
}
]
}

```

## Trackers policy example

Kebijakan berikut memberikan akses ke sumber daya pelacak yang diberi nama *ExampleTracker* untuk memperbarui posisi perangkat.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "UpdateDevicePosition",
      "Effect": "Allow",
      "Action": [
        "geo:BatchUpdateDevicePosition"
      ],
      "Resource": "arn:aws:geo:region:accountID:tracker/ExampleTracker"
    }
  ]
}

```

Menambahkan [kondisi IAM](#) yang cocok `aws:referer` memungkinkan Anda membatasi akses browser ke sumber daya Anda ke daftar URL atau awalan URL. Contoh berikut menolak akses ke sumber daya pelacak yang dinamai *ExampleTracker* dari semua situs web yang merujuk, kecuali `example.com`

### Warning

Meskipun `aws:referer` dapat membatasi akses, itu bukan mekanisme keamanan. Menyertakan nilai header perujuk yang diketahui publik bukanlah sesuatu yang aman. Pihak yang tidak berwenang dapat menggunakan browser yang diubah atau disesuaikan untuk menyediakan nilai `aws:referer` yang mereka pilih. Akibatnya, tidak `aws:referer` boleh digunakan untuk mencegah pihak yang tidak berwenang

membuat AWS permintaan langsung. Ini ditawarkan untuk memungkinkan pelanggan melindungi konten digital mereka, seperti konten yang disimpan di Amazon S3, agar tidak dirujuk pada pihak ketiga yang tidak berwenang. [Untuk informasi lebih lanjut, lihat:referer.AWS](#)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "geo:GetDevice*",
      "Resource": "arn:aws:geo:us-west-2:111122223333:tracker/ExampleTracker",
      "Condition": {
        "StringLike": {
          "aws:referer": [
            "https://example.com/*",
            "https://www.example.com/*"
          ]
        }
      }
    }
  ]
}
```

### Routes policy example

Kebijakan berikut memberikan akses ke sumber kalkulator rute yang diberi nama *ExampleCalculator* untuk menghitung rute.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RoutesReadOnly",
      "Effect": "Allow",
      "Action": [
        "geo:CalculateRoute"
      ],

```

```

    "Resource": "arn:aws:geo:region:accountID:route-
calculator/ExampleCalculator"
  }
]
}

```

Menambahkan [kondisi IAM](#) yang cocok `aws:referer` memungkinkan Anda membatasi akses browser ke sumber daya Anda ke daftar URL atau awalan URL. Contoh berikut menolak akses ke kalkulator rute yang dinamai *ExampleCalculator* dari semua situs web yang merujuk, kecuali `example.com`.

#### Warning

Meskipun `aws:referer` dapat membatasi akses, itu bukan mekanisme keamanan. Menyertakan nilai header perujuk yang diketahui publik bukanlah sesuatu yang aman. Pihak yang tidak berwenang dapat menggunakan browser yang diubah atau disesuaikan untuk menyediakan nilai `aws:referer` yang mereka pilih. Akibatnya, tidak `aws:referer` boleh digunakan untuk mencegah pihak yang tidak berwenang membuat AWS permintaan langsung. Ini ditawarkan untuk memungkinkan pelanggan melindungi konten digital mereka, seperti konten yang disimpan di Amazon S3, agar tidak dirujuk pada pihak ketiga yang tidak berwenang. [Untuk informasi lebih lanjut, lihat:referer.AWS](#)

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "geo:*",
      "Resource": "arn:aws:geo:us-west-2:111122223333:route-
calculator/ExampleCalculator",
      "Condition": {
        "StringLike": {
          "aws:referer": [
            "https://example.com/*",
            "https://www.example.com/*"
          ]
        }
      }
    }
  ]
}

```



```

    }
  ]
}

```

### Note

Meskipun kumpulan identitas yang tidak diautentikasi dimaksudkan untuk eksposur di situs internet yang tidak aman, perhatikan bahwa mereka akan ditukar dengan kredensi standar dan terbatas waktu. AWS

Penting untuk mencakup peran IAM yang terkait dengan kumpulan identitas yang tidak diautentikasi dengan tepat.

11. Pilih Izinkan untuk membuat kumpulan identitas Anda.

Kumpulan identitas yang dihasilkan mengikuti sintaks `<region>:<GUID>`.

Sebagai contoh:

```
us-east-1:1sample4-5678-90ef-aaaa-1234abcd56ef
```

Untuk contoh kebijakan lainnya khusus untuk Lokasi Amazon, lihat [the section called "Contoh kebijakan berbasis identitas"](#).

## Menggunakan kumpulan identitas Amazon Cognito di JavaScript

Contoh berikut menukar kumpulan identitas tidak diautentikasi yang telah Anda buat untuk kredensi yang kemudian digunakan untuk mengambil deskriptor gaya untuk sumber daya peta Anda.

### *ExampleMap*

```

const AWS = require("aws-sdk");

const credentials = new AWS.CognitoIdentityCredentials({
  IdentityPoolId: "<identity pool ID>" // for example, us-east-1:1sample4-5678-90ef-
  aaaa-1234abcd56ef
});

const client = new AWS.Location({
  credentials,

```

```
    region: AWS.config.region || "<region>"
  });

console.log(await client.getMapStyleDescriptor("ExampleMap").promise());
```

### Note

Kredensial yang diambil dari identitas yang tidak diautentikasi berlaku selama satu jam.

Berikut ini adalah contoh fungsi yang secara otomatis memperbarui kredensial sebelum kedaluwarsa.

```
async function refreshCredentials() {
  await credentials.refreshPromise();
  // schedule the next credential refresh when they're about to expire
  setTimeout(refreshCredentials, credentials.expireTime - new Date());
}
```

Untuk menyederhanakan pekerjaan ini, Anda dapat menggunakan Lokasi [JavaScript Pembantu otentikasi](#) Amazon. Ini menggantikan keduanya mendapatkan kredensial, dan menyegarkannya. Contoh ini menggunakan AWS SDK untuk JavaScript v3.

```
import { LocationClient, GetMapStyleDescriptorCommand } from "@aws-sdk/client-location";
import { withIdentityPoolId } from "@aws/amazon-location-utilities-auth-helper";

const identityPoolId = "<identity pool ID>"; // for example, us-east-1:1sample4-5678-90ef-aaaa-1234abcd56ef

// Create an authentication helper instance using credentials from Cognito
const authHelper = await withIdentityPoolId(identityPoolId);

const client = new LocationClient({
  region: "<region>", // The region containing both the identity pool and tracker resource
  ...authHelper.getLocationClientConfig(), // Provides configuration required to make requests to Amazon Location
});

const input = {
```

```
MapName: "ExampleMap",
};

const command = new GetMapStyleDescriptorCommand(input);

console.log(await client.send(command));
```

## Langkah selanjutnya

- Untuk memodifikasi peran Anda, buka [konsol IAM](#).
- Untuk mengelola kumpulan identitas Anda, buka konsol [Amazon Cognito](#).

## Memantau Amazon Location Service

Saat menggunakan Amazon Location Service, Anda dapat memantau penggunaan dan sumber daya dari waktu ke waktu dengan menggunakan:

- Amazon CloudWatch. Memantau sumber daya Amazon Location Service Anda, dan menyediakan metrik dengan statistik dalam waktu hampir nyata.
- AWS CloudTrail. Menyediakan pelacakan peristiwa semua panggilan ke Amazon Location Service API.

Bagian ini memberikan informasi tentang penggunaan layanan ini.

### Topik

- [Memantau Amazon Location Service dengan Amazon CloudWatch](#)
- [Pencatatan dan pemantauan dengan AWS CloudTrail](#)

## Memantau Amazon Location Service dengan Amazon CloudWatch

Amazon CloudWatch memantau AWS sumber daya Anda dan aplikasi yang Anda jalankan hampir AWS secara real time. Anda dapat memantau sumber daya Lokasi Amazon menggunakan CloudWatch, yang mengumpulkan data mentah dan memproses metrik menjadi statistik yang bermakna dalam waktu hampir nyata. Anda dapat melihat informasi historis hingga 15 bulan, atau menelusuri metrik untuk dilihat di CloudWatch konsol Amazon untuk mendapatkan perspektif lebih lanjut tentang sumber daya Lokasi Amazon Anda. Anda juga dapat mengatur alarm dengan

menentukan ambang batas, dan mengirim pemberitahuan atau mengambil tindakan ketika ambang batas tersebut terpenuhi.

Untuk informasi selengkapnya, lihat [Panduan CloudWatch Pengguna Amazon](#)

## Topik

- [Metrik Amazon Location Service diekspor ke Amazon CloudWatch](#)
- [Lihat metrik Amazon Location Service](#)
- [Membuat CloudWatch alarm untuk metrik Amazon Location Service](#)
- [Menggunakan CloudWatch untuk memantau penggunaan terhadap kuota](#)
- [CloudWatch contoh metrik untuk Amazon Location Service](#)

## Metrik Amazon Location Service diekspor ke Amazon CloudWatch

Metrik adalah titik data yang diurutkan waktu yang diekspor ke. CloudWatch Dimensi adalah pasangan nama/nilai yang mengidentifikasi metrik. Untuk informasi selengkapnya, lihat [Menggunakan CloudWatch metrik](#) dan [CloudWatch](#) dimensi di Panduan CloudWatch Pengguna Amazon.

Berikut ini adalah metrik yang diekspor Amazon Location Service CloudWatch di namespace. `AWS/Location`

Metrik	Deskripsi
<code>CallCount</code>	<p>Jumlah panggilan yang dilakukan ke titik akhir API tertentu.</p> <p>Dimensi yang Valid: Nama API Amazon Location Service</p> <p>Statistik Valid: Sum</p> <p>Unit: Hitungan</p>
<code>ErrorCount</code>	<p>Jumlah respons kesalahan dari panggilan yang dilakukan ke titik akhir API tertentu.</p> <p>Dimensi yang Valid: Nama API Amazon Location Service</p> <p>Statistik Valid: Sum</p>

Metrik	Deskripsi
	Unit: Hitungan
SuccessCount	Jumlah panggilan yang berhasil dilakukan ke titik akhir API tertentu.  Dimensi yang Valid: Nama API Amazon Location Service  Statistik Valid: Sum  Unit: Hitungan
CallLatency	Jumlah waktu yang diperlukan operasi untuk memproses dan mengembalikan respons saat panggilan dilakukan ke titik akhir API tertentu.  Dimensi yang Valid: Nama API Amazon Location Service  Statistik yang Valid: Average  Satuan: Milidetik

## Lihat metrik Amazon Location Service

Anda dapat melihat metrik untuk Amazon Location Service di Amazon CloudWatch konsol atau dengan menggunakan Amazon CloudWatch API.

Untuk melihat metrik menggunakan konsol CloudWatch

### Example

1. Buka CloudWatch konsol di <https://console.aws.amazon.com/cloudwatch/>.
2. Pada panel navigasi, silakan pilih Metrik.
3. Pada tab Semua metrik, pilih namespace Lokasi Amazon.
4. Pilih jenis metrik untuk dilihat.
5. Pilih metrik dan tambahkan ke bagan.

Untuk informasi selengkapnya, lihat [Melihat Metrik yang Tersedia](#) di Panduan CloudWatch Pengguna Amazon.

## Membuat CloudWatch alarm untuk metrik Amazon Location Service

Anda dapat menggunakan CloudWatch untuk menyetel alarm pada metrik Amazon Location Service. Misalnya, Anda dapat membuat alarm CloudWatch untuk mengirim email setiap kali terjadi lonjakan jumlah kesalahan.

Topik berikut memberi Anda gambaran tingkat tinggi tentang cara menyetel alarm menggunakan CloudWatch. Untuk petunjuk terperinci, lihat [Menggunakan Alarm](#) di Panduan CloudWatch Pengguna Amazon.

Untuk mengatur alarm menggunakan konsol CloudWatch

### Example

1. Buka CloudWatch konsol di <https://console.aws.amazon.com/cloudwatch/>.
2. Di panel navigasi, pilih Alarm.
3. Pilih Buat Alarm.
4. Pilih Pilih Metrik.
5. Pada tab Semua metrik, pilih namespace Lokasi Amazon.
6. Pilih kategori metrik.
7. Temukan baris dengan metrik yang ingin Anda buat alarm, lalu pilih kotak centang di sebelah baris ini.
8. Pilih Pilih Metrik.
9. Di bawah Metric, isi nilainya.
10. Tentukan Kondisi alarm.
11. Pilih Berikutnya.
12. Jika Anda ingin mengirim pemberitahuan ketika kondisi alarm terpenuhi:
  - Di bawah Pemicu status alarm, pilih status alarm untuk meminta pemberitahuan dikirim.
  - Di bawah Pilih topik SNS, pilih Buat topik baru untuk membuat topik baru Amazon Simple Notification Service (Amazon SNS). Masukkan nama topik dan email untuk mengirim notifikasi.
  - Di bawah Kirim pemberitahuan untuk memasukkan alamat email tambahan untuk mengirim pemberitahuan ke.
  - Pilih Tambahkan notifikasi. Daftar ini disimpan dan muncul dalam bidang isian untuk alarm selanjutnya.

13. Setelah selesai, pilih Selanjutnya.

14. Masukkan nama dan deskripsi untuk alarm, lalu pilih Berikutnya.

15. Konfirmasikan detail alarm, lalu pilih Berikutnya.

#### Note

Saat membuat topik Amazon SNS baru, Anda harus memverifikasi alamat email sebelum pemberitahuan dapat dikirim. Jika email tidak diverifikasi, pemberitahuan tidak akan diterima ketika alarm dimulai oleh perubahan status.

Untuk informasi selengkapnya tentang cara menyetel alarm menggunakan CloudWatch konsol, lihat [Membuat Alarm yang Mengirim Email](#) di Panduan CloudWatch Pengguna Amazon.

## Menggunakan CloudWatch untuk memantau penggunaan terhadap kuota

Anda dapat membuat CloudWatch alarm Amazon untuk memberi tahu Anda ketika penggunaan kuota yang diberikan melebihi ambang batas yang dapat dikonfigurasi. Ini memungkinkan Anda mengenali kapan Anda mendekati batas kuota Anda, dan menyesuaikan pemanfaatan Anda untuk menghindari pembengkakan biaya, atau meminta kenaikan kuota, jika diperlukan. Untuk informasi tentang cara menggunakan CloudWatch untuk memantau kuota, lihat [Memvisualisasikan kuota layanan Anda dan menyetel alarm di Panduan Pengguna](#) Amazon. CloudWatch

## CloudWatch contoh metrik untuk Amazon Location Service

Anda dapat menggunakan [GetMetricData](#) API untuk mengambil metrik untuk Lokasi Amazon.

- Misalnya, Anda dapat memantau `CallCount` dan mengatur alarm saat terjadi penurunan angka.

Memantau `CallCount` metrik `SendDeviceLocation` dapat membantu memberi Anda perspektif tentang aset yang dilacak. Jika `CallCount` turun, itu berarti aset yang dilacak, seperti armada truk, telah berhenti mengirim lokasi mereka saat ini. Menyetel alarm untuk ini dapat membantu memberi tahu Anda bahwa masalah telah terjadi.

- Untuk contoh lain, Anda dapat memantau `ErrorCount` dan mengatur alarm ketika lonjakan jumlah terjadi.

Pelacak harus dikaitkan dengan koleksi geofence agar lokasi perangkat dapat dievaluasi terhadap geofences. Jika Anda memiliki armada perangkat yang memerlukan pembaruan

lokasi berkelanjutan, melihat `CallCount` for `BatchEvaluateGeofence` atau `BatchPutDevicePosition` drop ke nol menunjukkan bahwa pembaruan tidak lagi mengalir.

Berikut ini adalah contoh output untuk [GetMetricData](#) dengan metrik untuk `CallCount` dan `ErrorCount` untuk membuat sumber daya peta.

```
{
  "StartTime": 1518867432,
  "EndTime": 1518868032,
  "MetricDataQueries": [
    {
      "Id": "m1",
      "MetricStat": {
        "Metric": {
          "Namespace": "AWS/Location",
          "MetricName": "CallCount",
          "Dimensions": [
            {
              "Name": "SendDeviceLocation",
              "Value": "100"
            }
          ]
        },
        "Period": 300,
        "Stat": "SampleCount",
        "Unit": "Count"
      }
    },
    {
      "Id": "m2",
      "MetricStat": {
        "Metric": {
          "Namespace": "AWS/Location",
          "MetricName": "ErrorCount",
          "Dimensions": [
            {
              "Name": "AssociateTrackerConsumer",
              "Value": "0"
            }
          ]
        },
        "Period": 1,
```



```
        "Stat": "SampleCount",
        "Unit": "Count"
    }
}
]
```

## Pencatatan dan pemantauan dengan AWS CloudTrail

AWS CloudTrail adalah layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan. CloudTrail merekam semua panggilan API sebagai peristiwa. Anda dapat menggunakan Amazon Location Service with CloudTrail untuk memantau panggilan API Anda, yang mencakup panggilan dari konsol Amazon Location Service dan panggilan AWS SDK ke operasi Amazon Location Service API.

Saat membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail peristiwa secara terus menerus ke bucket S3, termasuk peristiwa untuk Amazon Location Service. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat ke Amazon Location Service, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk informasi selengkapnya CloudTrail, lihat [Panduan AWS CloudTrail Pengguna](#).

### Topik

- [Informasi Layanan Lokasi Amazon di CloudTrail](#)
- [Memahami Entri File Log Layanan Lokasi Amazon](#)

## Informasi Layanan Lokasi Amazon di CloudTrail

CloudTrail diaktifkan di AWS akun Anda saat Anda membuat akun. Ketika aktivitas terjadi di Amazon Location Service, aktivitas tersebut direkam dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh peristiwa terbaru di akun AWS. Untuk informasi selengkapnya, lihat [Melihat Acara dengan Riwayat CloudTrail Acara](#).

Untuk catatan peristiwa yang sedang berlangsung di AWS akun Anda, termasuk acara untuk Amazon Location Service, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket

S3. Secara bawaan, ketika Anda membuat jejak di konsol tersebut, jejak diterapkan ke semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di partisi AWS dan mengirimkan file log ke bucket S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log.

Untuk informasi selengkapnya, lihat berikut:

- [Gambaran umum untuk Membuat Jejak](#)
- [CloudTrail Layanan dan Integrasi yang Didukung](#)
- [Mengkonfigurasi Notifikasi Amazon SNS untuk CloudTrail](#)
- [Menerima File CloudTrail Log dari Beberapa Wilayah](#) dan [Menerima File CloudTrail Log dari Beberapa Akun](#)

Semua tindakan Amazon Location Service dicatat oleh CloudTrail dan didokumentasikan dalam [referensi Amazon Location Service API](#). Misalnya, panggilan `createTracker`, `updateTracker` dan `describeTracker` tindakan menghasilkan entri dalam file CloudTrail log.

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

- Dengan kredensi pengguna root atau AWS Identity and Access Management (IAM).
- Dengan kredensi keamanan sementara untuk peran atau pengguna federasi.
- Dengan AWS layanan lain.

Untuk informasi lain, lihat [Elemen userIdentity CloudTrail](#) .

## Memahami Entri File Log Layanan Lokasi Amazon

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket S3 yang Anda tentukan, atau ke Amazon CloudWatch Logs. Untuk informasi selengkapnya, lihat [Bekerja dengan file CloudTrail log](#) di Panduan AWS CloudTrail Pengguna.

CloudTrail file log berisi satu atau lebih entri log. Acara mewakili satu permintaan dari sumber apa pun dan mencakup informasi tentang operasi yang diminta, tanggal dan waktu operasi, parameter permintaan, dan sebagainya.

**Note**

CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, jadi file tersebut tidak muncul dalam urutan tertentu. Untuk menentukan urutan operasi, gunakan [eventTime](#).

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan CreateTracker operasi, yang menciptakan sumber daya pelacak.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "123456789012",
    "arn": "arn:aws:geo:us-east-1:123456789012:tracker/ExampleTracker",
    "accountId": "123456789012",
    "accessKeyId": "123456789012",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "123456789012",
        "arn": "arn:aws:geo:us-east-1:123456789012:tracker/ExampleTracker",
        "accountId": "123456789012",
        "userName": "exampleUser",
      },
      "webIdFederationData": {},
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2020-10-22T16:36:07Z"
      }
    }
  },
  "eventTime": "2020-10-22T17:43:30Z",
  "eventSource": "geo.amazonaws.com",
  "eventName": "CreateTracker",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0/24-TEST-NET-1",
  "userAgent": "aws-internal/3 aws-sdk-java/1.11.864
Linux/4.14.193-110.317.amzn2.x86_64 OpenJDK_64-Bit_Server_VM/11.0.8+10-LTS java/11.0.8
kotlin/1.3.72 vendor/Amazon.com_Inc. exec-env/AWS_Lambda_java11",
  "requestParameters": {
```

```

    "TrackerName": "ExampleTracker",
    "Description": "Resource description"
  },
  "responseElements": {
    "TrackerName": "ExampleTracker",
    "Description": "Resource description"
    "TrackerArn": "arn:partition:service:region:account-id:resource-id",
    "CreateTime": "2020-10-22T17:43:30.521Z"
  },
  "requestID": "557ec619-0674-429d-8e2c-eba0d3f34413",
  "eventID": "3192bc9c-3d3d-4976-bbef-ac590fa34f2c",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012",
}

```

Berikut ini menunjukkan entri log untuk DescribeTracker operasi, yang mengembalikan rincian sumber daya pelacak.

```

{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "123456789012",
    "arn": "arn:partition:service:region:account-id:resource-id",
    "accountId": "123456789012",
    "accessKeyId": "123456789012",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "123456789012",
        "arn": "arn:partition:service:region:account-id:resource-id",
        "accountId": "123456789012",
        "userName": "exampleUser",
      },
      "webIdFederationData": {},
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2020-10-22T16:36:07Z"
      }
    }
  },
  "eventTime": "2020-10-22T17:43:33Z",
}

```

```
"eventSource": "geo.amazonaws.com",
"eventName": "DescribeTracker",
"awsRegion": "us-east-1",
"sourceIPAddress": "192.0.2.0/24-TEST-NET-1",
"userAgent": "aws-internal/3 aws-sdk-java/1.11.864
Linux/4.14.193-110.317.amzn2.x86_64 OpenJDK_64-Bit_Server_VM/11.0.8+10-LTS java/11.0.8
kotlin/1.3.72 vendor/Amazon.com_Inc. exec-env/AWS_Lambda_java11",
"requestParameters": {
  "TrackerName": "ExampleTracker"
},
"responseElements": null,
"requestID": "997d5f93-cfef-429a-bbed-daab417ceab4",
"eventID": "d9e0eebe-173c-477d-b0c9-d1d8292da103",
"readOnly": true,
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012",
}
```

## Membuat sumber daya Amazon Location Service dengan AWS CloudFormation

Amazon Location Service terintegrasi dengan AWS CloudFormation, layanan yang membantu Anda memodelkan dan menyiapkan AWS sumber daya sehingga Anda dapat menghabiskan lebih sedikit waktu untuk membuat dan mengelola sumber daya dan infrastruktur Anda. Anda membuat templat yang menjelaskan semua AWS sumber daya yang Anda inginkan (seperti sumber daya Lokasi Amazon), serta menyediakan serta AWS CloudFormation mengonfigurasi sumber daya tersebut untuk Anda.

Bila Anda menggunakan AWS CloudFormation, Anda dapat menggunakan kembali template Anda untuk mengatur sumber daya Lokasi Amazon Anda secara konsisten dan berulang kali. Jelaskan sumber daya Anda sekali, lalu sediakan sumber daya yang sama berulang-ulang di beberapa akun dan Wilayah AWS.

### Lokasi Amazon dan AWS CloudFormation template

Untuk menyediakan dan mengonfigurasi sumber daya untuk Lokasi Amazon dan layanan terkait, Anda harus memahami [AWS CloudFormation templat](#). Templat adalah file teks dengan format JSON atau YAML. Templat ini menjelaskan sumber daya yang ingin Anda sediakan di tumpukan AWS CloudFormation. Jika tidak terbiasa dengan JSON atau YAML, Anda dapat menggunakan Desainer

AWS CloudFormation untuk membantu Anda memulai dengan templat AWS CloudFormation. Lihat informasi selengkapnya di [Apa yang dimaksud dengan Desainer AWS CloudFormation?](#) dalam Panduan Pengguna AWS CloudFormation.

Amazon Location mendukung pembuatan jenis sumber daya berikut di AWS CloudFormation:

- [AWS::Location::Map](#)
- [AWS::Location::PlaceIndex](#)
- [AWS::Location::RouteCalculator](#)
- [AWS::Location::Tracker](#)
- [AWS::Location::TrackerConsumer](#)
- [AWS::Location::GeofenceCollection](#)

Untuk informasi selengkapnya, termasuk contoh templat JSON dan YAMAL untuk sumber daya Lokasi Amazon, lihat [referensi jenis sumber daya Amazon Location Service](#) di AWS CloudFormation Panduan Pengguna.

## Pelajari selengkapnya tentang AWS CloudFormation

Untuk mempelajari selengkapnya tentang AWS CloudFormation, lihat sumber daya berikut:

- [AWS CloudFormation](#)
- [Panduan Pengguna AWS CloudFormation](#)
- [Referensi API AWS CloudFormation](#)
- [Panduan Pengguna Antarmuka Baris Perintah AWS CloudFormation](#)

# Keamanan di Amazon Location Service

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](#) menjelaskan hal ini sebagai keamanan cloud dan keamanan dalam 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 teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari [Program AWS Kepatuhan Program AWS Kepatuhan](#) . Untuk mempelajari tentang program kepatuhan yang berlaku untuk Amazon Location Service, lihat [AWS Services in Scope by Compliance Program AWS](#) .
- 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 ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan Lokasi Amazon. Topik berikut menunjukkan cara mengonfigurasi Lokasi Amazon untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan sumber daya Lokasi Amazon Anda.

## Topik

- [Perlindungan data di Amazon Location Service](#)
- [Identity and Access Management untuk Amazon Location Service](#)
- [Respon Insiden di Amazon Location Service](#)
- [Validasi kepatuhan untuk Amazon Location Service](#)
- [Ketahanan di Amazon Location Service](#)
- [Keamanan infrastruktur di Amazon Location Service](#)
- [Analisis konfigurasi dan kerentanan di Lokasi Amazon](#)
- [Pencegahan confused deputy lintas layanan](#)
- [Praktik terbaik keamanan untuk Amazon Location Service](#)

- [Praktik terbaik untuk Amazon Location Service](#)

## Perlindungan data di Amazon Location Service

[Model tanggung jawab AWS bersama model tanggung](#) berlaku untuk perlindungan data di Amazon Location Service. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Lihat informasi yang lebih lengkap tentang privasi data dalam [Pertanyaan Umum Privasi Data](#). Lihat informasi tentang perlindungan data di Eropa di pos blog [Model Tanggung Jawab Bersama dan GDPR AWS](#) di Blog Keamanan AWS .

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensial dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan sumber daya. AWS Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-2 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi yang lebih lengkap tentang titik akhir FIPS yang tersedia di [Standar Pemrosesan Informasi Federal \(FIPS\) 140-2](#).

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan Lokasi Amazon atau lainnya Layanan AWS menggunakan konsol, API AWS CLI, atau AWS SDK. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan



atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

## Privasi data

Dengan Amazon Location Service, Anda mempertahankan kendali atas data organisasi Anda. Lokasi Amazon menganonimkan semua kueri yang dikirim ke penyedia data dengan menghapus metadata pelanggan dan informasi akun.

Lokasi Amazon tidak menggunakan penyedia data untuk melacak dan melakukan geofencing. Ini berarti data sensitif Anda tetap ada di AWS akun Anda. Ini membantu melindungi informasi lokasi yang sensitif, seperti fasilitas, aset, dan lokasi personel, dari pihak ketiga, melindungi privasi pengguna, dan mengurangi risiko keamanan aplikasi Anda.

Untuk informasi tambahan, lihat [FAQ Privasi AWS Data](#).

## Penyimpanan data di Lokasi Amazon

Karakteristik berikut berhubungan dengan cara Amazon Location mengumpulkan dan menyimpan data untuk layanan:

- Amazon Location Service Trackers — Saat Anda menggunakan API Pelacak untuk melacak lokasi entitas, koordinatnya dapat disimpan. Lokasi perangkat disimpan selama 30 hari sebelum dihapus oleh layanan.
- Amazon Location Service Geofences — Saat Anda menggunakan Geofences API untuk menentukan area minat, layanan akan menyimpan geometri yang Anda berikan. Mereka harus dihapus secara eksplisit.

### Note

Menghapus AWS akun Anda menghapus semua sumber daya di dalamnya. Untuk informasi tambahan, lihat [FAQ Privasi AWS Data](#).

## Enkripsi data saat istirahat untuk Amazon Location Service

Amazon Location Service menyediakan enkripsi secara default untuk melindungi data pelanggan sensitif saat istirahat menggunakan kunci enkripsi yang AWS dimiliki.

- AWS kunci yang dimiliki — Lokasi Amazon menggunakan kunci ini secara default untuk secara otomatis mengenkripsi data yang dapat diidentifikasi secara pribadi. Anda tidak dapat melihat, mengelola, atau menggunakan kunci yang AWS dimiliki, atau mengaudit penggunaannya. Namun, Anda tidak perlu mengambil tindakan apa pun atau mengubah program apa pun untuk melindungi kunci yang mengenkripsi data Anda. Untuk informasi selengkapnya, lihat [kunci yang AWS dimiliki](#) di Panduan AWS Key Management Service Pengembang.

Enkripsi data saat istirahat secara default membantu mengurangi overhead operasional dan kompleksitas yang terlibat dalam melindungi data sensitif. Pada saat yang sama, ini memungkinkan Anda untuk membangun aplikasi aman yang memenuhi kepatuhan enkripsi yang ketat dan persyaratan peraturan.

Meskipun Anda tidak dapat menonaktifkan lapisan enkripsi ini atau memilih jenis enkripsi alternatif, Anda dapat menambahkan lapisan enkripsi kedua di atas kunci enkripsi AWS milik yang ada dengan memilih kunci yang dikelola pelanggan saat Anda membuat sumber daya pelacak dan pengumpulan geofence:

- Kunci terkelola pelanggan — Lokasi Amazon mendukung penggunaan kunci terkelola pelanggan simetris yang Anda buat, miliki, dan kelola untuk menambahkan enkripsi lapisan kedua di atas enkripsi AWS milik yang ada. Karena Anda memiliki kontrol penuh atas lapisan enkripsi ini, Anda dapat melakukan tugas-tugas seperti:
  - Menetapkan dan memelihara kebijakan utama
  - Menetapkan dan memelihara kebijakan dan hibah IAM
  - Mengaktifkan dan menonaktifkan kebijakan utama
  - Memutar bahan kriptografi kunci
  - Menambahkan tanda
  - Membuat alias kunci
  - Kunci penjadwalan untuk penghapusan

Untuk informasi selengkapnya, lihat [kunci terkelola pelanggan](#) di Panduan AWS Key Management Service Pengembang.

Tabel berikut merangkum bagaimana Lokasi Amazon mengenkripsi data yang dapat diidentifikasi secara pribadi.

Tipe data	AWS enkripsi kunci yang dimiliki	Enkripsi kunci yang dikelola pelanggan (Opsional)
<p>Position</p> <p>Geometri titik yang berisi <a href="#">detail posisi perangkat</a>.</p>	Diaktifkan	Diaktifkan
<p>PositionProperties</p> <p>Satu set pasangan kunci-nilai yang <a href="#">terkait dengan pembaruan posisi</a>.</p>	Diaktifkan	Diaktifkan
<p>GeofenceGeometry</p> <p><a href="#">Geometri geofensi</a> poligon yang mewakili area geofenced</p>	Diaktifkan	Diaktifkan
<p>DeviceId</p> <p>Pengenal perangkat ditentukan saat <a href="#">mengunggah pembaruan posisi perangkat</a> ke sumber daya pelacak.</p>	Diaktifkan	Tidak didukung
<p>GeofenceId</p> <p>Pengidentifikasi yang ditentukan saat <a href="#">menyimpan geometri geofence</a>, atau <a href="#">sekumpulan geofences dalam koleksi geofence</a> tertentu.</p>	Diaktifkan	Tidak didukung

**Note**

Lokasi Amazon secara otomatis mengaktifkan enkripsi saat istirahat menggunakan kunci yang AWS dimiliki untuk melindungi data yang dapat diidentifikasi secara pribadi tanpa biaya. Namun, AWS KMS biaya berlaku untuk menggunakan kunci yang dikelola pelanggan. Untuk informasi selengkapnya tentang harga, lihat [AWS Key Management Service harga](#).

Untuk informasi lebih lanjut tentang AWS KMS, lihat [Apa itu AWS Key Management Service?](#)

## Bagaimana Amazon Location Service menggunakan hibah di AWS KMS

Lokasi Amazon memerlukan [hibah](#) untuk menggunakan kunci yang dikelola pelanggan Anda.

Saat Anda membuat [sumber daya pelacak](#) atau [koleksi geofence](#) yang dienkripsi dengan kunci yang dikelola pelanggan, Amazon Location membuat hibah atas nama Anda dengan mengirimkan permintaan ke [CreateGrant](#) AWS KMS Hibah AWS KMS digunakan untuk memberikan Akses Lokasi Amazon ke kunci KMS di akun pelanggan.

Lokasi Amazon memerlukan hibah untuk menggunakan kunci terkelola pelanggan Anda untuk operasi internal berikut:

- Kirim [DescribeKey](#) permintaan AWS KMS untuk memverifikasi bahwa ID kunci KMS yang dikelola pelanggan simetris yang dimasukkan saat membuat pelacak atau koleksi geofence valid.
- Kirim [GenerateDataKeyWithoutPlaintext](#) permintaan AWS KMS untuk menghasilkan kunci data yang dienkripsi oleh kunci terkelola pelanggan Anda.
- Kirim permintaan [Dekripsi](#) ke AWS KMS untuk mendekripsi kunci data terenkripsi sehingga mereka dapat digunakan untuk mengenkripsi data Anda.

Anda dapat mencabut akses ke hibah, atau menghapus akses layanan ke kunci yang dikelola pelanggan kapan saja. Jika Anda melakukannya, Lokasi Amazon tidak akan dapat mengakses data apa pun yang dienkripsi oleh kunci yang dikelola pelanggan, yang memengaruhi operasi yang bergantung pada data tersebut. Misalnya, jika Anda mencoba [mendapatkan posisi perangkat](#) dari pelacak terenkripsi yang tidak dapat diakses Lokasi Amazon, maka operasi akan mengembalikan kesalahan. `AccessDeniedException`

## Buat kunci terkelola pelanggan

Anda dapat membuat kunci terkelola pelanggan simetris dengan menggunakan AWS Management Console, atau AWS KMS API.

Untuk membuat kunci terkelola pelanggan simetris

Ikuti langkah-langkah untuk [Membuat kunci terkelola pelanggan simetris](#) di Panduan AWS Key Management Service Pengembang.

### Kebijakan utama

Kebijakan utama mengontrol akses ke kunci yang dikelola pelanggan Anda. Setiap kunci yang dikelola pelanggan harus memiliki persis satu kebijakan utama, yang berisi pernyataan yang menentukan siapa yang dapat menggunakan kunci dan bagaimana mereka dapat menggunakannya. Saat membuat kunci terkelola pelanggan, Anda dapat menentukan kebijakan kunci. Untuk informasi selengkapnya, lihat [Mengelola akses ke kunci yang dikelola pelanggan](#) di Panduan AWS Key Management Service Pengembang.

Untuk menggunakan kunci terkelola pelanggan dengan sumber daya Lokasi Amazon, operasi API berikut harus diizinkan dalam kebijakan kunci:

- [kms:CreateGrant](#)— Menambahkan hibah ke kunci yang dikelola pelanggan. Memberikan akses kontrol ke kunci KMS tertentu, yang memungkinkan akses untuk [memberikan operasi yang diperlukan Lokasi Amazon](#). Untuk informasi selengkapnya tentang [Menggunakan Hibah](#), lihat Panduan AWS Key Management Service Pengembang.

Hal ini memungkinkan Amazon Location untuk melakukan hal berikut:

- Panggilan `GenerateDataKeyWithoutPlainText` untuk menghasilkan kunci data terenkripsi dan menyimpannya, karena kunci data tidak segera digunakan untuk mengenkripsi.
- Panggilan `Decrypt` untuk menggunakan kunci data terenkripsi yang disimpan untuk mengakses data terenkripsi.
- Siapkan kepala sekolah yang pensiun untuk memungkinkan layanan. `RetireGrant`
- [kms:DescribeKey](#)— Memberikan detail kunci yang dikelola pelanggan untuk memungkinkan Lokasi Amazon memvalidasi kunci.

Berikut ini adalah contoh pernyataan kebijakan yang dapat Anda tambahkan untuk Lokasi Amazon:

```
"Statement" : [
```

```
{
  "Sid" : "Allow access to principals authorized to use Amazon Location",
  "Effect" : "Allow",
  "Principal" : {
    "AWS" : "*"
  },
  "Action" : [
    "kms:DescribeKey",
    "kms:CreateGrant"
  ],
  "Resource" : "*",
  "Condition" : {
    "StringEquals" : {
      "kms:ViaService" : "geo.region.amazonaws.com",
      "kms:CallerAccount" : "111122223333"
    }
  },
},
{
  "Sid": "Allow access for key administrators",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:root"
  },
  "Action" : [
    "kms:*"
  ],
  "Resource": "arn:aws:kms:region:111122223333:key/key_ID"
},
{
  "Sid" : "Allow read-only access to key metadata to the account",
  "Effect" : "Allow",
  "Principal" : {
    "AWS" : "arn:aws:iam::111122223333:root"
  },
  "Action" : [
    "kms:Describe*",
    "kms:Get*",
    "kms:List*",
    "kms:RevokeGrant"
  ],
  "Resource" : "*"
}
]
```

Untuk informasi selengkapnya tentang [menentukan izin dalam kebijakan](#), lihat Panduan AWS Key Management Service Pengembang.

Untuk informasi selengkapnya tentang [akses kunci pemecahan](#) masalah, lihat Panduan AWS Key Management Service Pengembang.

## Menentukan kunci terkelola pelanggan untuk Lokasi Amazon

Anda dapat menentukan kunci yang dikelola pelanggan sebagai enkripsi lapisan kedua untuk sumber daya berikut:

- [Sumber daya pelacak](#)
- [Koleksi Geofence](#)

Saat membuat sumber daya, Anda dapat menentukan kunci data dengan memasukkan ID KMS, yang digunakan Lokasi Amazon untuk mengenkripsi data pribadi yang dapat diidentifikasi yang disimpan oleh sumber daya.

- ID KMS — [Pengidentifikasi kunci](#) untuk kunci yang dikelola AWS KMS pelanggan. Masukkan ID kunci, ARN kunci, nama alias, atau ARN alias.

## Konteks enkripsi Amazon Location Service

[Konteks enkripsi](#) adalah kumpulan opsional pasangan kunci-nilai yang berisi informasi kontekstual tambahan tentang data.

AWS KMS menggunakan konteks enkripsi sebagai [data otentikasi tambahan](#) untuk mendukung enkripsi yang [diautentikasi](#). Bila Anda menyertakan konteks enkripsi dalam permintaan untuk mengenkripsi data, AWS KMS mengikat konteks enkripsi ke data terenkripsi. Untuk mendekripsi data, Anda menyertakan konteks enkripsi yang sama dalam permintaan.

### Konteks enkripsi Amazon Location Service

Lokasi Amazon menggunakan konteks enkripsi yang sama di semua operasi AWS KMS kriptografi, di mana kuncinya adalah `aws:geo:arn` dan nilainya adalah sumber daya [Amazon Resource Name](#) (ARN).

### Example

```
"encryptionContext": {
```

```
"aws:geo:arn": "arn:aws:geo:us-west-2:111122223333:geofence-collection/SAMPLE-GeofenceCollection"
}
```

## Menggunakan konteks enkripsi untuk pemantauan

Saat Anda menggunakan kunci terkelola pelanggan simetris untuk mengenkripsi pelacak atau koleksi geofence, Anda juga dapat menggunakan konteks enkripsi dalam catatan audit dan log untuk mengidentifikasi bagaimana kunci yang dikelola pelanggan digunakan. Konteks enkripsi juga muncul di [log yang dihasilkan oleh AWS CloudTrail atau Amazon CloudWatch Logs](#).

## Menggunakan konteks enkripsi untuk mengontrol akses ke kunci terkelola pelanggan Anda

Anda dapat menggunakan konteks enkripsi dalam kebijakan utama dan kebijakan IAM conditions untuk mengontrol akses ke kunci terkelola pelanggan simetris Anda. Anda juga dapat menggunakan kendala konteks enkripsi dalam hibah.

Lokasi Amazon menggunakan batasan konteks enkripsi dalam hibah untuk mengontrol akses ke kunci yang dikelola pelanggan di akun atau wilayah Anda. Batasan hibah mengharuskan operasi yang diizinkan oleh hibah menggunakan konteks enkripsi yang ditentukan.

## Example

Berikut ini adalah contoh pernyataan kebijakan kunci untuk memberikan akses ke kunci yang dikelola pelanggan untuk konteks enkripsi tertentu. Kondisi dalam pernyataan kebijakan ini mengharuskan hibah memiliki batasan konteks enkripsi yang menentukan konteks enkripsi.

```
{
  "Sid": "Enable DescribeKey",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleReadOnlyRole"
  },
  "Action": "kms:DescribeKey",
  "Resource": "*"
},
{
  "Sid": "Enable CreateGrant",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleReadOnlyRole"
  },
  "Action": "kms:CreateGrant",
```



```

    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "kms:EncryptionContext:aws:geo:arn": "arn:aws:geo:us-
west-2:111122223333:tracker/SAMPLE-Tracker"
      }
    }
  }
}

```

## Memantau kunci enkripsi Anda untuk Amazon Location Service

Saat menggunakan kunci terkelola AWS KMS pelanggan dengan sumber daya Amazon Location Service, Anda dapat menggunakan [AWS CloudTrail](#) atau [Amazon CloudWatch Logs](#) untuk melacak permintaan yang dikirimkan oleh Amazon Location AWS KMS.

Contoh berikut adalah AWS CloudTrail peristiwa untuk `CreateGrant`, `GenerateDataKeyWithoutPlainTextDecrypt`, dan `DescribeKey` untuk memantau operasi KMS yang dipanggil oleh Lokasi Amazon untuk mengakses data yang dienkripsi oleh kunci terkelola pelanggan Anda:

### CreateGrant

Saat Anda menggunakan kunci yang dikelola AWS KMS pelanggan untuk mengenkripsi sumber daya pelacak atau pengumpulan geofence, Amazon Location mengirimkan `CreateGrant` permintaan atas nama Anda untuk mengakses kunci KMS di akun Anda. AWS Hibah yang dibuat Lokasi Amazon khusus untuk sumber daya yang terkait dengan kunci yang dikelola AWS KMS pelanggan. Selain itu, Lokasi Amazon menggunakan `RetireGrant` operasi untuk menghapus hibah saat Anda menghapus sumber daya.

Contoh peristiwa berikut mencatat `CreateGrant` operasi:

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",

```

```

        "principalId": "AROAIIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "userName": "Admin"
    },
    "webIdFederationData": {},
    "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2021-04-22T17:02:00Z"
    }
},
"invokedBy": "geo.amazonaws.com"
},
"eventTime": "2021-04-22T17:07:02Z",
"eventSource": "kms.amazonaws.com",
"eventName": "CreateGrant",
"awsRegion": "us-west-2",
"sourceIPAddress": "172.12.34.56",
"userAgent": "ExampleDesktop/1.0 (V1; OS)",
"requestParameters": {
    "retiringPrincipal": "geo.region.amazonaws.com",
    "operations": [
        "GenerateDataKeyWithoutPlaintext",
        "Decrypt",
        "DescribeKey"
    ],
    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
    "granteePrincipal": "geo.region.amazonaws.com"
},
"responseElements": {
    "grantId":
"0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE"
},
"requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"readOnly": false,
"resources": [
    {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
}

```

```

    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333"
  }

```

## GenerateDataKeyWithoutPlainText

Saat Anda mengaktifkan kunci terkelola AWS KMS pelanggan untuk sumber daya koleksi pelacak atau geofence, Amazon Location akan membuat kunci tabel unik. Ini mengirimkan `GenerateDataKeyWithoutPlainText` permintaan ke AWS KMS yang menentukan kunci yang dikelola AWS KMS pelanggan untuk sumber daya.

Contoh peristiwa berikut mencatat `GenerateDataKeyWithoutPlainText` operasi:

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AWSService",
    "invokedBy": "geo.amazonaws.com"
  },
  "eventTime": "2021-04-22T17:07:02Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GenerateDataKeyWithoutPlaintext",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "172.12.34.56",
  "userAgent": "ExampleDesktop/1.0 (V1; OS)",
  "requestParameters": {
    "encryptionContext": {
      "aws:geo:arn": "arn:aws:geo:us-west-2:111122223333:geofence-collection/SAMPLE-GeofenceCollection"
    },
    "keySpec": "AES_256",
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
  },
  "responseElements": null,
  "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "readOnly": true,
  "resources": [
    {

```

```

        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "111122223333",
"sharedEventID": "57f5dbee-16da-413e-979f-2c4c6663475e"
}

```

## Decrypt

Saat Anda mengakses pelacak terenkripsi atau koleksi geofence, Amazon Location memanggil Decrypt operasi untuk menggunakan kunci data terenkripsi yang disimpan untuk mengakses data terenkripsi.

Contoh peristiwa berikut mencatat Decrypt operasi:

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AWSService",
    "invokedBy": "geo.amazonaws.com"
  },
  "eventTime": "2021-04-22T17:10:51Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "Decrypt",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "172.12.34.56",
  "userAgent": "ExampleDesktop/1.0 (V1; OS)",
  "requestParameters": {
    "encryptionContext": {
      "aws:geo:arn": "arn:aws:geo:us-west-2:111122223333:geofence-collection/
SAMPLE-GeofenceCollection"
    },
    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
    "encryptionAlgorithm": "SYMMETRIC_DEFAULT"
  },
  "responseElements": null,

```

```

"requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"readOnly": true,
"resources": [
  {
    "accountId": "111122223333",
    "type": "AWS::KMS::Key",
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "111122223333",
"sharedEventID": "dc129381-1d94-49bd-b522-f56a3482d088"
}

```

## DescribeKey

Lokasi Amazon menggunakan DescribeKey operasi untuk memverifikasi apakah kunci terkelola AWS KMS pelanggan yang terkait dengan pelacak atau koleksi geofence Anda ada di akun dan wilayah.

Contoh peristiwa berikut mencatat DescribeKey operasi:

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "userName": "Admin"
      },
      "webIdFederationData": {},

```

```
    "attributes": {
      "mfaAuthenticated": "false",
      "creationDate": "2021-04-22T17:02:00Z"
    },
    "invokedBy": "geo.amazonaws.com"
  },
  "eventTime": "2021-04-22T17:07:02Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "DescribeKey",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "172.12.34.56",
  "userAgent": "ExampleDesktop/1.0 (V1; OS)",
  "requestParameters": {
    "keyId": "00dd0db0-0000-0000-ac00-b0c000SAMPLE"
  },
  "responseElements": null,
  "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "readOnly": true,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "eventCategory": "Management",
  "recipientAccountId": "111122223333"
}
```

## Pelajari selengkapnya

Sumber daya berikut memberikan informasi lebih lanjut tentang enkripsi data saat istirahat.

- Untuk informasi selengkapnya tentang [konsep AWS Key Management Service dasar](#), lihat Panduan AWS Key Management Service Pengembang.
- Untuk informasi selengkapnya tentang [praktik terbaik Keamanan AWS Key Management Service](#), lihat Panduan AWS Key Management Service Pengembang.

## Enkripsi data dalam transit untuk Amazon Location Service

Amazon Location melindungi data dalam perjalanan, saat melakukan perjalanan ke dan dari layanan, dengan secara otomatis mengenkripsi semua data antar-jaringan menggunakan protokol enkripsi Transport Layer Security (TLS) 1.2. Permintaan HTTPS langsung yang dikirim ke Amazon Location Service API ditandatangani dengan menggunakan [Algoritma AWS Signature Version 4](#) untuk membuat koneksi yang aman.

## Identity and Access Management untuk Amazon Location Service

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 diotorisasi (memiliki izin) untuk menggunakan sumber daya Lokasi Amazon. IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

### Topik

- [Audiens](#)
- [Mengautentikasi dengan identitas](#)
- [Mengelola akses menggunakan kebijakan](#)
- [Bagaimana Amazon Location Service bekerja dengan IAM](#)
- [Cara Amazon Location Service bekerja dengan pengguna yang tidak diautentikasi](#)
- [Contoh kebijakan berbasis identitas untuk Amazon Location Service](#)
- [Memecahkan masalah identitas dan akses Amazon Location Service](#)

## Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan di Lokasi Amazon.

Pengguna layanan — Jika Anda menggunakan layanan Lokasi Amazon untuk melakukan pekerjaan Anda, administrator Anda memberi Anda kredensi dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak fitur Lokasi Amazon untuk melakukan pekerjaan Anda, Anda mungkin memerlukan izin tambahan. Memahami cara akses dikelola dapat membantu Anda meminta izin

yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di Lokasi Amazon, lihat [Memecahkan masalah identitas dan akses Amazon Location Service](#).

Administrator layanan - Jika Anda bertanggung jawab atas sumber daya Lokasi Amazon di perusahaan Anda, Anda mungkin memiliki akses penuh ke Lokasi Amazon. Tugas Anda adalah menentukan fitur dan sumber daya Lokasi Amazon 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 Basic IAM. Untuk mempelajari selengkapnya tentang bagaimana perusahaan Anda dapat menggunakan IAM dengan Lokasi Amazon, lihat [Bagaimana Amazon Location Service bekerja dengan IAM](#).

Administrator IAM - Jika Anda administrator IAM, Anda mungkin ingin mempelajari detail tentang cara menulis kebijakan untuk mengelola akses ke Lokasi Amazon. Untuk melihat contoh kebijakan berbasis identitas Lokasi Amazon yang dapat Anda gunakan di IAM, lihat [Contoh kebijakan berbasis identitas untuk Amazon Location Service](#)

## Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensial 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 kredensial yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (IAM Identity Center), autentikasi masuk tunggal perusahaan Anda, dan kredensial 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 kredensial Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Untuk informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, lihat [Menandatangani permintaan AWS API](#) di 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 [Menggunakan autentikasi multi-faktor \(MFA\) dalam AWS](#) 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 kredensial 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 kredensial yang disediakan melalui sumber identitas. Ketika identitas federasi mengakses Akun AWS, mereka mengambil peran, dan peran memberikan kredensial 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 memiliki grup yang bernama IAMAdmins dan memberikan izin ke 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 [Kapan harus membuat pengguna IAM \(bukan peran\)](#) 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. Anda dapat mengambil peran IAM untuk sementara AWS Management Console dengan [beralih peran](#). 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 [Menggunakan peran IAM](#) 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 [Membuat 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 di Panduan Pengguna IAM](#).
- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Sebagai contoh, ketika Anda memanggil 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 [Membuat 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 Layanan AWS. Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Akun AWS Anda 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 kredensial sementara untuk aplikasi yang berjalan pada instans EC2 dan membuat atau permintaan API. AWS CLI AWS Cara ini lebih dianjurkan daripada menyimpan kunci akses dalam instans EC2. Untuk menetapkan AWS peran ke instans EC2 dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instans berisi peran dan memungkinkan program yang berjalan di instans EC2 mendapatkan kredensial

sementara. Untuk informasi selengkapnya, lihat [Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan dalam instans Amazon EC2](#) dalam Panduan Pengguna IAM.

Untuk mempelajari apakah kita harus menggunakan peran IAM atau pengguna IAM, lihat [Kapan harus membuat peran IAM \(bukan pengguna\)](#) dalam 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 [Membuat kebijakan IAM](#) 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 dilampirkan 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 [Memilih 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 (ACL)

Daftar kontrol akses (ACL) mengendalikan prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun kebijakan tersebut tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung ACL. Untuk mempelajari ACL selengkapnya, lihat [Gambaran umum daftar kontrol akses \(ACL\)](#) dalam Panduan Developer Amazon Simple Storage Service.

## 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 (SCP) — SCP 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 di organisasi, Anda dapat menerapkan kebijakan kontrol layanan (SCP) 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 Organisasi dan SCP, lihat [Cara kerja SCP](#) dalam Panduan Pengguna AWS Organizations.
- 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 Amazon Location Service bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke Lokasi Amazon, pelajari fitur IAM apa yang tersedia untuk digunakan dengan Lokasi Amazon.

Fitur IAM yang dapat Anda gunakan dengan Amazon Location Service

Fitur IAM	Dukungan Lokasi Amazon
<a href="#">Kebijakan berbasis identitas</a>	Ya
<a href="#">Kebijakan berbasis sumber daya</a>	Tidak

Fitur IAM	Dukungan Lokasi Amazon
<a href="#">Tindakan kebijakan</a>	Ya
<a href="#">Sumber daya kebijakan</a>	Ya
<a href="#">kunci-kunci persyaratan kebijakan (spesifik layanan)</a>	Ya
<a href="#">ACL</a>	Tidak
<a href="#">ABAC (tanda dalam kebijakan)</a>	Ya
<a href="#">Kredensial sementara</a>	Ya
<a href="#">Izin prinsipal</a>	Tidak
<a href="#">Peran layanan</a>	Tidak
<a href="#">Peran terkait layanan</a>	Tidak

Untuk mendapatkan tampilan tingkat tinggi tentang cara kerja Lokasi Amazon dan AWS layanan lainnya dengan sebagian besar fitur IAM, lihat [AWS layanan yang bekerja dengan IAM di Panduan Pengguna IAM](#).

## Kebijakan berbasis identitas untuk Lokasi Amazon

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 [Membuat kebijakan IAM](#) 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 Lokasi Amazon

Untuk melihat contoh kebijakan berbasis identitas Lokasi Amazon, lihat. [Contoh kebijakan berbasis identitas untuk Amazon Location Service](#)

## Kebijakan berbasis sumber daya dalam Lokasi Amazon

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 prinsipal dalam akun yang sama, tidak diperlukan kebijakan berbasis identitas tambahan. Untuk informasi selengkapnya, lihat [Akses sumber daya lintas akun di IAM](#) di Panduan Pengguna IAM.

## Tindakan kebijakan untuk Lokasi Amazon

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.

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

Untuk melihat daftar tindakan Lokasi Amazon, lihat [Tindakan yang Ditentukan oleh Amazon Location Service](#) di Referensi Otorisasi Layanan.

Tindakan kebijakan di Lokasi Amazon menggunakan awalan berikut sebelum tindakan:

```
geo
```

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

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

Anda juga dapat menentukan beberapa tindakan menggunakan wildcard (\*). Sebagai contoh, untuk menentukan semua tindakan yang dimulai dengan kata `Get`, sertakan tindakan berikut:

```
"Action": "geo:Get*"
```

Untuk melihat contoh kebijakan berbasis identitas Lokasi Amazon, lihat [Contoh kebijakan berbasis identitas untuk Amazon Location Service](#)

## Sumber daya kebijakan untuk Lokasi Amazon

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 Lokasi Amazon dan ARNnya, lihat Sumber [Daya yang Ditentukan oleh Amazon Location Service](#) di Referensi Otorisasi Layanan. Untuk mempelajari tindakan mana yang dapat Anda tentukan ARN dari setiap sumber daya, lihat [Tindakan yang Ditentukan oleh Amazon Location Service](#).

Untuk melihat contoh kebijakan berbasis identitas Lokasi Amazon, lihat. [Contoh kebijakan berbasis identitas untuk Amazon Location Service](#)

## Kunci kondisi kebijakan untuk Lokasi Amazon

Mendukung kunci kondisi kebijakan khusus layanan	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 `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 tag yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, lihat [Elemen kebijakan IAM: variabel dan tag](#) 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 kondisi Lokasi Amazon, lihat [Kunci Kondisi untuk Amazon Location Service](#) di Referensi Otorisasi Layanan. Untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi, lihat [Tindakan yang Ditentukan oleh Amazon Location Service](#).

Lokasi Amazon mendukung kunci kondisi untuk memungkinkan Anda mengizinkan atau menolak akses ke geofences atau perangkat tertentu dalam pernyataan kebijakan Anda. Kunci kondisi berikut tersedia:

- `geo:GeofenceIds` untuk digunakan dengan tindakan `Geofence`. Tipenya adalah `ArrayOfString`.
- `geo:DeviceIds` untuk digunakan dengan tindakan `Tracker`. Tipenya adalah `ArrayOfString`.


Tindakan berikut dapat digunakan `geo:GeofenceIds` dalam kebijakan IAM Anda:

- `BatchDeleteGeofences`
- `BatchPutGeofences`
- `GetGeofence`
- `PutGeofence`

Tindakan berikut dapat digunakan `geo:DeviceIds` dalam kebijakan IAM Anda:

- `BatchDeleteDevicePositionHistory`

- `BatchGetDevicePosition`
- `BatchUpdateDevicePosition`
- `GetDevicePosition`
- `GetDevicePositionHistory`

 Note

Anda tidak dapat menggunakan tombol kondisi ini dengan `BatchEvaluateGeofences`, `ListGeofences`, atau `ListDevicePosition` tindakan.

Untuk melihat contoh kebijakan berbasis identitas Lokasi Amazon, lihat. [Contoh kebijakan berbasis identitas untuk Amazon Location Service](#)

## ACL di Lokasi Amazon

Mendukung ACL

Tidak

Daftar kontrol akses (ACL) mengendalikan pengguna utama mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun kebijakan tersebut tidak menggunakan format dokumen kebijakan JSON.

## ABAC dengan Lokasi Amazon

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 tag milik prinsipal cocok dengan tag 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 tag, berikan informasi tentang tag 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 [Apa itu 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.

Untuk informasi selengkapnya tentang menandai sumber daya Lokasi Amazon, lihat [Menandai sumber daya Amazon Location Service](#).

Untuk melihat contoh kebijakan berbasis identitas untuk membatasi akses ke sumber daya berdasarkan tag pada sumber daya tersebut, lihat [Kontrol akses sumber daya berdasarkan tag](#).

## Menggunakan kredensial sementara dengan Lokasi Amazon

Mendukung penggunaan kredensial sementara    Ya

Beberapa Layanan AWS tidak berfungsi saat Anda masuk menggunakan kredensial 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 kredensial 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 [Peralihan peran \(konsol\)](#) dalam Panduan Pengguna IAM.

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

## Izin utama lintas layanan untuk Lokasi Amazon

Mendukung sesi akses maju (FAS)	Tidak
---------------------------------	-------

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 Lokasi Amazon

Mendukung peran layanan	Tidak
-------------------------	-------

Peran layanan adalah sebuah [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 [Membuat sebuah peran untuk mendelegasikan izin ke Layanan AWS](#) dalam Panduan pengguna IAM.

### Warning

Mengubah izin untuk peran layanan dapat merusak fungsionalitas Lokasi Amazon. Edit peran layanan hanya jika Lokasi Amazon memberikan panduan untuk melakukannya.

## Peran terkait layanan untuk Lokasi Amazon

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.

## Cara Amazon Location Service bekerja dengan pengguna yang tidak diautentikasi

Banyak skenario untuk menggunakan Amazon Location Service, termasuk menampilkan peta di web atau di aplikasi seluler, mengharuskan akses ke pengguna yang belum masuk dengan IAM. Untuk skenario yang tidak diautentikasi ini, Anda memiliki dua opsi.

- Gunakan kunci API — Untuk memberikan akses ke pengguna yang tidak diautentikasi, Anda dapat membuat Kunci API yang memberikan akses hanya-baca ke sumber daya Amazon Location Service Anda. Ini berguna dalam kasus di mana Anda tidak ingin mengautentikasi setiap pengguna. Misalnya, aplikasi web. Untuk informasi selengkapnya tentang kunci API, lihat [Mengizinkan akses tamu yang tidak diautentikasi ke aplikasi Anda menggunakan kunci API](#).
- Gunakan Amazon Cognito — Alternatif untuk kunci API adalah dengan menggunakan Amazon Cognito untuk memberikan akses anonim. Amazon Cognito memungkinkan Anda membuat otorisasi yang lebih kaya dengan kebijakan IAM untuk menentukan apa yang dapat dilakukan oleh pengguna yang tidak diautentikasi. Untuk informasi selengkapnya tentang menggunakan Amazon Cognito, lihat [Mengizinkan akses tamu yang tidak diautentikasi ke aplikasi Anda menggunakan Amazon Cognito](#)

Untuk gambaran umum tentang menyediakan akses ke pengguna yang tidak diautentikasi, lihat [Memberikan akses ke Amazon Location Service](#)

## Contoh kebijakan berbasis identitas untuk Amazon Location Service

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi sumber daya Lokasi Amazon. 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 akan dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM menggunakan contoh dokumen kebijakan JSON ini, lihat [Membuat kebijakan IAM](#) dalam Panduan Pengguna IAM.

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh Lokasi Amazon, termasuk format ARN untuk setiap jenis sumber daya, lihat [Tindakan, Sumber Daya, dan Kunci Kondisi untuk Amazon Location Service](#) di Referensi Otorisasi Layanan.

## Topik

- [Praktik terbaik kebijakan](#)
- [Menggunakan konsol Lokasi Amazon](#)
- [Mengizinkan pengguna melihat izin mereka sendiri](#)
- [Menggunakan sumber daya Amazon Location Service dalam kebijakan](#)
- [Izin untuk memperbarui posisi perangkat](#)
- [Kebijakan hanya-baca untuk sumber daya pelacak](#)
- [Kebijakan untuk membuat geofences](#)
- [Kebijakan hanya-baca untuk geofences](#)
- [Izin untuk merender sumber daya peta](#)
- [Izin untuk memungkinkan operasi pencarian](#)
- [Kebijakan hanya-baca untuk kalkulator rute](#)
- [Kontrol akses sumber daya berdasarkan tombol kondisi](#)
- [Kontrol akses sumber daya berdasarkan tag](#)

## Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya Lokasi Amazon 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 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 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 [Mengonfigurasi akses API yang dilindungi MFA](#) dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat [Praktik terbaik keamanan dalam IAM](#) dalam Panduan Pengguna IAM.

## Menggunakan konsol Lokasi Amazon

Untuk mengakses konsol Amazon Location Service, Anda harus memiliki set izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang sumber daya Lokasi Amazon di 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 dapat menggunakan konsol Lokasi Amazon, lampirkan kebijakan berikut ke entitas. Untuk informasi selengkapnya, lihat [Menambah izin untuk pengguna](#) dalam Panduan Pengguna IAM.

Kebijakan berikut memberikan akses ke konsol Amazon Location Service, agar dapat membuat, menghapus, mencantumkan, dan melihat detail tentang sumber daya Lokasi Amazon di AWS akun Anda.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GeoPowerUser",
      "Effect": "Allow",
      "Action": [
        "geo:*"
      ],
      "Resource": "*"
    }
  ]
}
```

Atau, Anda dapat memberikan izin hanya-baca untuk memfasilitasi akses hanya-baca. Dengan izin hanya-baca, pesan kesalahan akan muncul jika pengguna mencoba menulis tindakan seperti membuat atau menghapus sumber daya. Sebagai contoh, lihat [the section called “Kebijakan hanya-baca untuk pelacak”](#)

## Mengizinkan pengguna 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:ListGroupForUser",
        "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": "*"
    }
  ]
}

```

## Menggunakan sumber daya Amazon Location Service dalam kebijakan

Amazon Location Service menggunakan awalan berikut untuk sumber daya:

Awalan sumber daya Lokasi Amazon

Sumber Daya	Awalan sumber daya
Sumber daya peta	map

Sumber Daya	Awalan sumber daya
Tempatkan sumber daya	place-index
Sumber daya rute	route-calculator
Melacak sumber daya	tracker
Sumber daya Koleksi Geofence	geofence-collection

Gunakan sintaks ARN berikut:

```
arn:Partition:geo:Region:Account:ResourcePrefix/ResourceName
```

Untuk informasi selengkapnya tentang format ARN, lihat [Nama Sumber Daya Amazon \(ARN\) dan Ruang Nama AWS Layanan](#).

Contoh

- Gunakan ARN berikut untuk mengizinkan akses ke sumber daya peta tertentu.

```
"Resource": "arn:aws:geo:us-west-2:account-id:map/map-resource-name"
```

- Untuk menentukan akses ke semua map sumber daya milik akun tertentu, gunakan wildcard (\*):

```
"Resource": "arn:aws:geo:us-west-2:account-id:map/*"
```

- Beberapa tindakan Lokasi Amazon, seperti untuk membuat sumber daya, tidak dapat dilakukan pada sumber daya tertentu. Dalam kasus tersebut, Anda harus menggunakan wildcard (\*).

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

Untuk melihat daftar jenis sumber daya Lokasi Amazon dan ARNnya, lihat Sumber [Daya yang Ditentukan oleh Amazon Location Service](#) di Referensi Otorisasi Layanan. Untuk mempelajari tindakan mana yang dapat Anda tentukan ARN dari setiap sumber daya, lihat [Tindakan yang Ditentukan oleh Amazon Location Service](#).

## Izin untuk memperbarui posisi perangkat

Untuk memperbarui posisi perangkat untuk beberapa pelacak, Anda harus memberi pengguna akses ke satu atau beberapa sumber daya pelacak Anda. Anda juga ingin mengizinkan pengguna memperbarui sejumlah posisi perangkat.

*Dalam contoh ini, selain memberikan akses ke sumber daya Tracker1 dan Tracker2, kebijakan berikut memberikan izin untuk menggunakan `geo:BatchUpdateDevicePosition` tindakan terhadap sumber daya Tracker1 dan Tracker2.*

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "UpdateDevicePositions",
      "Effect": "Allow",
      "Action": [
        "geo:BatchUpdateDevicePosition"
      ],
      "Resource": [
        "arn:aws:geo:us-west-2:account-id:tracker/Tracker1",
        "arn:aws:geo:us-west-2:account-id:tracker/Tracker2"
      ]
    }
  ]
}
```

Jika Anda ingin membatasi pengguna agar hanya dapat memperbarui posisi perangkat untuk perangkat tertentu, Anda dapat menambahkan kunci kondisi untuk id perangkat tersebut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "UpdateDevicePositions",
      "Effect": "Allow",
      "Action": [
        "geo:BatchUpdateDevicePosition"
      ],
      "Resource": [
        "arn:aws:geo:us-west-2:account-id:tracker/Tracker1",

```

```

    "arn:aws:geo:us-west-2:account-id:tracker/Tracker2"
  ],
  "Condition": {
    "ForAllValues:StringLike": {
      "geo:DeviceIds": [
        "deviceId"
      ]
    }
  }
}
]
}

```

## Kebijakan hanya-baca untuk sumber daya pelacak

Untuk membuat kebijakan hanya-baca untuk semua sumber daya pelacak di AWS akun Anda, Anda harus memberikan akses ke semua sumber daya pelacak. Anda juga ingin memberi pengguna akses ke tindakan yang memungkinkan mereka mendapatkan posisi perangkat untuk beberapa perangkat, mendapatkan posisi perangkat dari satu perangkat, dan mendapatkan riwayat posisi.

Dalam contoh ini, kebijakan berikut memberikan izin untuk tindakan berikut:

- `geo:BatchGetDevicePosition` untuk mengambil posisi beberapa perangkat.
- `geo:GetDevicePosition` untuk mengambil posisi satu perangkat.
- `geo:GetDevicePositionHistory` untuk mengambil riwayat posisi perangkat.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GetDevicePositions",
      "Effect": "Allow",
      "Action": [
        "geo:BatchGetDevicePosition",
        "geo:GetDevicePosition",
        "geo:GetDevicePositionHistory"
      ],
      "Resource": "arn:aws:geo:us-west-2:account-id:tracker/*"
    }
  ]
}

```

```
}
```

## Kebijakan untuk membuat geofences

Untuk membuat kebijakan agar pengguna dapat membuat geofences, Anda harus memberikan akses ke tindakan tertentu yang memungkinkan pengguna membuat satu atau beberapa geofences pada koleksi geofence.

Kebijakan di bawah ini memberikan izin untuk tindakan berikut pada *Koleksi*:

- `geo:BatchPutGeofence` untuk membuat beberapa geofences.
- `geo:PutGeofence` untuk membuat geofence tunggal.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CreateGeofences",
      "Effect": "Allow",
      "Action": [
        "geo:BatchPutGeofence",
        "geo:PutGeofence"
      ],
      "Resource": "arn:aws:geo:us-west-2:account-id:geofence-collection/Collection"
    }
  ]
}
```

## Kebijakan hanya-baca untuk geofences

Untuk membuat kebijakan hanya-baca untuk geofences yang disimpan dalam koleksi geofence di AWS akun Anda, Anda harus memberikan akses ke tindakan yang dibaca dari koleksi geofence yang menyimpan geofences.

Kebijakan di bawah ini memberikan izin untuk tindakan berikut pada *Koleksi*:

- `geo:ListGeofences` untuk membuat daftar geofences dalam koleksi geofence yang ditentukan.
- `geo:GetGeofence` untuk mengambil geofence dari koleksi geofence.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GetGeofences",
      "Effect": "Allow",
      "Action": [
        "geo:ListGeofences",
        "geo:GetGeofence"
      ],
      "Resource": "arn:aws:geo:us-west-2:account-id:geofence-collection/Collection"
    }
  ]
}
```

## Izin untuk merender sumber daya peta

Untuk memberikan izin yang cukup untuk merender peta, Anda harus memberikan akses ke ubin peta, sprite, mesin terbang, dan deskriptor gaya:

- `geo:GetMapTile` mengambil ubin peta yang digunakan untuk secara selektif merender fitur pada peta.
- `geo:GetMapSprites` mengambil lembar sprite PNG dan dokumen JSON yang sesuai yang menjelaskan offset di dalamnya.
- `geo:GetMapGlyphs` mengambil glyph yang digunakan untuk menampilkan teks.
- `geo:GetMapStyleDescriptor` mengambil deskriptor gaya peta, yang berisi aturan rendering.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GetTiles",
      "Effect": "Allow",
      "Action": [
        "geo:GetMapTile",
        "geo:GetMapSprites",
        "geo:GetMapGlyphs",
        "geo:GetMapStyleDescriptor"
      ],
      "Resource": "arn:aws:geo:us-west-2:account-id:map/Map"
    }
  ]
}
```



```
    }  
  ]  
}
```

## Izin untuk memungkinkan operasi pencarian

Untuk membuat kebijakan untuk mengizinkan operasi penelusuran, Anda harus terlebih dahulu memberikan akses ke sumber daya indeks tempat di AWS akun Anda. Anda juga ingin memberikan akses ke tindakan yang memungkinkan pengguna mencari menggunakan teks dengan geocoding dan mencari menggunakan posisi dengan membalikkan geocoding.

Dalam contoh ini, selain memberikan akses ke *PlaceIndex*, kebijakan berikut juga memberikan izin untuk tindakan berikut:

- `geo:SearchPlaceIndexForPosition` memungkinkan Anda untuk mencari tempat, atau tempat menarik di dekat posisi tertentu.
- `geo:SearchPlaceIndexForText` memungkinkan Anda untuk mencari alamat, nama, kota atau wilayah menggunakan teks bentuk bebas.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "Search",  
      "Effect": "Allow",  
      "Action": [  
        "geo:SearchPlaceIndexForPosition",  
        "geo:SearchPlaceIndexForText"  
      ],  
      "Resource": "arn:aws:geo:us-west-2:account-id:place-index/PlaceIndex"  
    }  
  ]  
}
```

## Kebijakan hanya-baca untuk kalkulator rute

Anda dapat membuat kebijakan hanya-baca untuk memungkinkan pengguna mengakses sumber daya kalkulator rute untuk menghitung rute.

Dalam contoh ini, selain memberikan akses ke *ExampleCalculator*, kebijakan berikut memberikan izin untuk operasi berikut:

- `geo:CalculateRoute` menghitung rute yang diberikan posisi keberangkatan, posisi tujuan, dan daftar posisi waypoint.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RoutesReadOnly",
      "Effect": "Allow",
      "Action": [
        "geo:CalculateRoute"
      ],
      "Resource": "arn:aws:geo:us-west-2:accountID:route-calculator/ExampleCalculator"
    }
  ]
}
```

## Kontrol akses sumber daya berdasarkan tombol kondisi

Saat Anda membuat kebijakan IAM untuk memberikan akses untuk menggunakan geofences atau posisi perangkat, Anda dapat menggunakan [operator Kondisi](#) untuk kontrol yang lebih tepat atas geofences atau perangkat mana yang dapat diakses pengguna. Anda dapat melakukannya dengan menyertakan id geofence atau id perangkat di `Condition` elemen kebijakan Anda.

Contoh kebijakan berikut menunjukkan cara membuat kebijakan yang memungkinkan pengguna memperbarui posisi perangkat untuk perangkat tertentu.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "UpdateDevicePositions",
      "Effect": "Allow",
      "Action": [
        "geo:BatchUpdateDevicePosition"
      ],
      "Resource": [
        "arn:aws:geo:us-west-2:account-id:tracker/Tracker"
      ]
    }
  ]
}
```

```
    ],
    "Condition":{
      "ForAllValues:StringLike":{
        "geo:DeviceIds":[
          "deviceId"
        ]
      }
    }
  }
]
```

## Kontrol akses sumber daya berdasarkan tag

Saat Anda membuat kebijakan IAM untuk memberikan akses untuk menggunakan sumber daya Lokasi Amazon, Anda dapat menggunakan [kontrol akses berbasis atribut untuk kontrol](#) yang lebih baik atas sumber daya yang dapat dimodifikasi, digunakan, atau dihapus oleh pengguna. Anda dapat melakukannya dengan menyertakan informasi tag dalam Condition elemen kebijakan Anda untuk mengontrol akses berdasarkan [tag](#) sumber daya Anda.

Contoh kebijakan berikut menunjukkan cara membuat kebijakan yang memungkinkan pengguna membuat geofences. *Ini memberikan izin untuk tindakan berikut untuk membuat satu atau beberapa geofences pada koleksi geofence yang disebut Koleksi:*

- geo:BatchPutGeofence untuk membuat beberapa geofences.
- geo:PutGeofence untuk membuat geofence tunggal.

Namun, kebijakan ini menggunakan Condition elemen untuk memberikan izin hanya jika tag *KoleksiOwner*, memiliki nilai nama pengguna pengguna tersebut.

- Misalnya, jika pengguna bernama richard-roe mencoba melihat *Koleksi* Lokasi Amazon, *Koleksi* harus diberi tag Owner=richard-roe atau owner=richard-roe. Jika tidak, pengguna ditolak aksesnya.

### Note

Kunci tanda syarat Owner sama dengan kedua Owner dan owner karena nama kunci syarat tidak terpengaruh huruf besar/kecil. Untuk informasi lebih lanjut, lihat [Elemen Kebijakan IAM JSON: Persyaratan](#) dalam Panduan Pengguna IAM.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CreateGeofencesIfOwner",
      "Effect": "Allow",
      "Action": [
        "geo:BatchPutGeofence",
        "geo:PutGeofence"
      ],
      "Resource": "arn:aws:geo:us-west-2:account-id:geofence-collection/Collection",
      "Condition": {
        "StringEquals": {"geo:ResourceTag/Owner": "${aws:username}"}
      }
    }
  ]
}
```

Untuk tutorial tentang [cara menentukan izin untuk mengakses sumber daya AWS berdasarkan tag](#), lihat Panduan AWS Identity and Access Management Pengguna.

## Memecahkan masalah identitas dan akses Amazon Location Service

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan Lokasi Amazon dan IAM.

### Topik

- [Saya tidak berwenang untuk melakukan tindakan di Lokasi Amazon](#)
- [Saya tidak berwenang untuk melakukan iam: PassRole](#)
- [Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses sumber daya Lokasi Amazon saya](#)

### Saya tidak berwenang untuk melakukan tindakan di Lokasi Amazon

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 `geo:GetWidget` rekaan.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:  
geo: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 *geo: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 bahwa Anda tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran ke Lokasi Amazon.

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 untuk melakukan tindakan di Lokasi Amazon. 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 sumber daya Lokasi Amazon 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 (ACL), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- Untuk mengetahui apakah Lokasi Amazon mendukung fitur-fitur ini, lihat [Bagaimana Amazon Location Service 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](#).

## Respon Insiden di Amazon Location Service

Keamanan adalah prioritas tertinggi di AWS. Sebagai bagian dari [model tanggung jawab bersama AWS](#) Cloud, AWS mengelola pusat data dan arsitektur jaringan yang memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan. Sebagai AWS pelanggan, Anda berbagi tanggung jawab untuk menjaga keamanan di cloud. Ini berarti Anda mengontrol keamanan yang Anda pilih untuk diterapkan dari AWS alat dan fitur yang dapat Anda akses.

Dengan menetapkan garis dasar keamanan yang memenuhi tujuan aplikasi Anda yang berjalan di cloud, Anda dapat mendeteksi penyimpangan yang dapat Anda tanggap. Karena respons insiden keamanan dapat menjadi topik yang kompleks, kami mendorong Anda untuk meninjau sumber daya berikut sehingga Anda lebih dapat memahami dampak respons insiden (IR) dan pilihan Anda terhadap tujuan perusahaan Anda: [Panduan Respons Insiden Keamanan AWS, whitepaper Praktik Terbaik AWS Security, dan AWS Cloud Adoption Framework \(AWS CAF\)](#).

## Logging dan Monitoring di Amazon Location Service

Penebangan dan pemantauan adalah bagian penting dari respons insiden. Ini memungkinkan Anda menetapkan garis dasar keamanan untuk mendeteksi penyimpangan yang dapat Anda selidiki dan tanggap. Dengan menerapkan pencatatan dan pemantauan untuk Amazon Location Service, Anda dapat mempertahankan keandalan, ketersediaan, dan kinerja untuk proyek dan sumber daya Anda.

AWS menyediakan beberapa alat yang dapat membantu Anda mencatat dan mengumpulkan data untuk respons insiden:

### AWS CloudTrail

Amazon Location Service terintegrasi dengan AWS CloudTrail, yang merupakan layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan. Ini termasuk tindakan dari konsol Amazon Location Service, dan panggilan terprogram ke operasi Amazon Location API. Catatan tindakan ini disebut peristiwa. Untuk informasi selengkapnya, lihat [Logging dan monitoring Amazon Location Service dengan AWS CloudTrail](#).

### Amazon CloudWatch

Anda dapat menggunakan Amazon CloudWatch untuk mengumpulkan dan menganalisis metrik yang terkait dengan akun Amazon Location Service Anda. Anda dapat mengaktifkan CloudWatch alarm untuk memberi tahu Anda jika metrik memenuhi kondisi tertentu, dan telah mencapai ambang batas yang ditentukan. Saat Anda membuat alarm, CloudWatch kirimkan pemberitahuan ke Layanan Pemberitahuan Sederhana Amazon yang Anda tentukan. Untuk informasi selengkapnya, lihat [Monitoring Amazon Location Service dengan Amazon CloudWatch](#).

### AWS Health Dasbor

Dengan menggunakan [AWS Health Dasbor](#), Anda dapat memverifikasi status layanan Amazon Location Service. Anda juga dapat memantau dan melihat data historis tentang peristiwa atau masalah apa pun yang mungkin memengaruhi AWS lingkungan Anda. Untuk informasi selengkapnya, silakan lihat [Panduan Pengguna AWS Health](#).


## Validasi kepatuhan untuk Amazon Location Service

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:

- [Panduan Memulai Cepat Keamanan dan Kepatuhan — Panduan](#) penerapan ini membahas pertimbangan arsitektur dan memberikan langkah-langkah untuk menerapkan lingkungan dasar AWS yang berfokus pada keamanan dan kepatuhan.
- [Arsitektur untuk Keamanan dan Kepatuhan HIPAA di Amazon Web Services](#) — Whitepaper ini menjelaskan bagaimana perusahaan dapat menggunakan AWS untuk membuat aplikasi yang memenuhi syarat HIPAA.

 Note

Tidak semua memenuhi Layanan AWS syarat HIPAA. Untuk informasi selengkapnya, lihat [Referensi Layanan yang Memenuhi 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 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 Manager](#)Ini Layanan AWS membantu Anda terus mengaudit AWS penggunaan Anda untuk menyederhanakan cara Anda mengelola risiko dan kepatuhan terhadap peraturan dan standar industri.



## Ketahanan di Amazon Location Service

Infrastruktur AWS global dibangun di sekitar Wilayah AWS dan Availability Zones. Wilayah AWS 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 fail over di antara zona tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau multi tradisional.

Untuk informasi selengkapnya tentang Wilayah AWS dan Availability Zone, lihat [Infrastruktur AWS Global](#).

Selain infrastruktur AWS global, Amazon Location menawarkan beberapa fitur untuk membantu mendukung ketahanan data dan kebutuhan pencadangan Anda.

## Keamanan infrastruktur di Amazon Location Service

Sebagai layanan terkelola, Amazon Location Service 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 Lokasi Amazon 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.

## Analisis konfigurasi dan kerentanan di Lokasi Amazon

Konfigurasi dan kontrol TI adalah tanggung jawab bersama antara AWS dan Anda, pelanggan kami. Untuk informasi selengkapnya, lihat [model tanggung jawab AWS bersama](#).

## Pencegahan confused deputy lintas layanan

Masalah deputy yang bingung adalah masalah keamanan di mana entitas yang tidak memiliki izin untuk melakukan tindakan dapat memaksa entitas yang lebih istimewa untuk melakukan tindakan. Pada tahun AWS, peniruan lintas layanan dapat mengakibatkan masalah wakil yang membingungkan. Peniruan identitas lintas layanan dapat terjadi ketika satu layanan (layanan yang dipanggil) memanggil layanan lain (layanan yang dipanggil). Layanan pemanggilan dapat dimanipulasi menggunakan izinnya untuk bertindak pada sumber daya pelanggan lain dengan cara yang seharusnya tidak dilakukannya kecuali bila memiliki izin untuk mengakses. Untuk mencegah hal ini, AWS menyediakan alat yang membantu Anda melindungi data untuk semua layanan dengan pengguna utama layanan yang telah diberi akses ke sumber daya di akun Anda.

Amazon Location Service tidak bertindak sebagai layanan panggilan atas nama Anda ke AWS layanan lain, jadi Anda tidak perlu menambahkan perlindungan ini dalam kasus ini. Untuk mempelajari lebih lanjut tentang wakil yang [bingung](#), lihat [Masalah wakil](#) yang bingung di Panduan AWS Identity and Access Management Pengguna.

## Praktik terbaik keamanan untuk Amazon Location Service

Amazon Location Service menyediakan sejumlah fitur keamanan yang perlu dipertimbangkan saat Anda mengembangkan dan menerapkan kebijakan keamanan Anda sendiri. Praktik terbaik berikut adalah pedoman umum dan tidak mewakili solusi keamanan yang lengkap. Karena praktik terbaik ini mungkin tidak sesuai atau tidak memadai untuk lingkungan Anda, perlakukan itu sebagai pertimbangan yang bermanfaat, bukan sebagai resep.

## Praktik terbaik keamanan Detektif untuk Amazon Location Service

Praktik terbaik berikut untuk Amazon Location Service dapat membantu mendeteksi insiden keamanan:

### Menerapkan alat pemantauan AWS

Pemantauan sangat penting untuk respons insiden dan menjaga keandalan dan keamanan sumber daya Amazon Location Service dan solusi Anda. Anda dapat menerapkan alat

pemantauan dari beberapa alat dan layanan yang tersedia melalui AWS untuk memantau sumber daya Anda dan layanan AWS Anda lainnya.

Misalnya, Amazon CloudWatch memungkinkan Anda memantau metrik untuk Amazon Location Service dan memungkinkan Anda mengatur alarm untuk memberi tahu Anda jika metrik memenuhi kondisi tertentu yang telah Anda tetapkan dan telah mencapai ambang batas yang telah Anda tetapkan. Saat membuat alarm, Anda dapat mengatur CloudWatch untuk mengirim pemberitahuan ke peringatan menggunakan Amazon Simple Notification Service. Untuk informasi selengkapnya, lihat [the section called “Pembuatan Log dan Pemantauan”](#).

### Aktifkan alat pencatatan AWS

Logging menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau layanan AWS di Amazon Location Service. Anda dapat menerapkan alat logging seperti AWS CloudTrail mengumpulkan data tentang tindakan untuk mendeteksi aktivitas API yang tidak biasa.

Saat Anda membuat jejak, Anda dapat mengonfigurasi CloudTrail untuk mencatat peristiwa. Peristiwa adalah catatan operasi sumber daya yang dilakukan pada atau di dalam sumber daya seperti permintaan yang dibuat ke Lokasi Amazon, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, saat permintaan dibuat, bersama dengan data tambahan. Untuk informasi selengkapnya, lihat [Mencatat Peristiwa Data untuk Jejak](#) di Panduan AWS CloudTrail Pengguna.

## Praktik terbaik keamanan preventif untuk Amazon Location Service

Praktik terbaik berikut untuk Amazon Location Service dapat membantu mencegah insiden keamanan:

### Gunakan koneksi aman

Selalu gunakan koneksi terenkripsi, seperti yang dimulai dengan `https://` untuk menjaga keamanan informasi sensitif saat transit.

### Menerapkan akses hak istimewa paling sedikit ke sumber daya

Saat Anda membuat kebijakan khusus ke sumber daya Lokasi Amazon, berikan hanya izin yang diperlukan untuk melakukan tugas. Disarankan untuk memulai dengan set izin minimum dan memberikan izin tambahan sesuai kebutuhan. Menerapkan akses hak istimewa paling sedikit sangat penting untuk mengurangi risiko dan dampak yang dapat dihasilkan dari kesalahan atau serangan berbahaya. Untuk informasi selengkapnya, lihat [the section called “Identity and Access Management”](#).

## Gunakan ID unik secara global sebagai ID perangkat

Gunakan konvensi berikut untuk ID perangkat.

- ID perangkat harus unik.
- ID perangkat tidak boleh dirahasiakan, karena dapat digunakan sebagai kunci asing ke sistem lain.
- ID perangkat tidak boleh berisi informasi identitas pribadi (PII), seperti ID perangkat telepon atau alamat email.
- ID perangkat tidak boleh diprediksi. Pengidentifikasi buram seperti UUID direkomendasikan.

Jangan sertakan PII dalam properti posisi perangkat

Saat mengirim pembaruan perangkat (misalnya, menggunakan [DevicePositionUpdate](#)), jangan sertakan informasi identifikasi pribadi (PII) seperti nomor telepon atau alamat email di `PositionProperties`

## Praktik terbaik untuk Amazon Location Service

Topik ini memberikan praktik terbaik untuk membantu Anda menggunakan Amazon Location Service. Meskipun praktik terbaik ini dapat membantu Anda memanfaatkan sepenuhnya Amazon Location Service, praktik tersebut tidak mewakili solusi lengkap. Anda harus mengikuti hanya rekomendasi yang berlaku untuk lingkungan Anda.

Topik

- [Keamanan](#)
- [Manajemen sumber daya](#)
- [Manajemen penagihan dan biaya](#)
- [Kuota dan penggunaan](#)

### Keamanan

Untuk membantu mengelola atau bahkan menghindari risiko keamanan, pertimbangkan praktik terbaik berikut:

- Gunakan federasi identitas dan peran IAM untuk mengelola, mengontrol, atau membatasi akses ke sumber daya Lokasi Amazon Anda. Untuk informasi selengkapnya, lihat [Praktik Terbaik IAM](#) dalam Panduan Pengguna IAM.

- Ikuti Prinsip Hak Istimewa Terkecil untuk hanya memberikan akses minimum yang diperlukan ke sumber daya Amazon Location Service Anda. Untuk informasi selengkapnya, lihat [the section called “Mengelola akses menggunakan kebijakan”](#).
- Untuk sumber daya Amazon Location Service yang digunakan dalam aplikasi web, batasi akses menggunakan kondisi `aws:referer` IAM, membatasi penggunaan oleh situs selain yang termasuk dalam daftar izin.
- Gunakan alat pemantauan dan pencatatan untuk melacak akses dan penggunaan sumber daya. Untuk informasi selengkapnya, lihat [the section called “Pembuatan Log dan Pemantauan”](#) dan [Mencatat Peristiwa Data untuk Jejak](#) di Panduan AWS CloudTrail Pengguna.
- Gunakan koneksi aman, seperti yang dimulai dengan `https://` untuk menambah keamanan dan melindungi pengguna terhadap serangan saat data sedang ditransmisikan antara server dan browser.

Untuk informasi lebih lanjut tentang praktik terbaik keamanan detektif dan preventif, lihat topik di [the section called “Praktik terbaik keamanan”](#).

## Manajemen sumber daya

Untuk membantu mengelola sumber daya lokasi Anda secara efektif di Amazon Location Service, pertimbangkan praktik terbaik berikut:

- Gunakan titik akhir regional yang merupakan pusat basis pengguna yang Anda harapkan untuk meningkatkan pengalaman mereka. Untuk informasi tentang titik akhir wilayah, lihat [Wilayah Lokasi Amazon dan titik akhir](#).
- Untuk sumber daya yang menggunakan penyedia data, seperti sumber daya peta dan sumber daya indeks tempat, pastikan untuk mengikuti ketentuan perjanjian penggunaan penyedia data tertentu. Untuk informasi selengkapnya, lihat [Penyedia data](#).
- Minimalkan pembuatan sumber daya dengan memiliki satu sumber daya untuk setiap konfigurasi peta, indeks tempat, atau rute. Dalam suatu wilayah, Anda biasanya hanya memerlukan satu sumber daya per penyedia data atau gaya peta. Sebagian besar aplikasi menggunakan sumber daya yang ada, dan tidak membuat sumber daya pada waktu berjalan.
- Saat menggunakan sumber daya yang berbeda dalam satu aplikasi, seperti sumber daya peta dan kalkulator rute, gunakan penyedia data yang sama di setiap sumber daya untuk memastikan bahwa data cocok. Misalnya, geometri rute yang Anda buat dengan kalkulator rute Anda sejajar dengan jalan-jalan di peta yang digambar menggunakan sumber daya peta.

## Manajemen penagihan dan biaya

Untuk membantu mengelola biaya dan penagihan Anda, pertimbangkan praktik terbaik berikut:

- Gunakan alat pemantauan, seperti Amazon CloudWatch, untuk melacak penggunaan sumber daya Anda. Anda dapat mengatur peringatan yang memberi tahu Anda saat penggunaan akan melebihi batas yang ditentukan. Untuk informasi selengkapnya, lihat [Membuat Alarm Penagihan untuk Memantau Perkiraan Biaya AWS Anda](#) di Panduan CloudWatch Pengguna Amazon.

## Kuota dan penggunaan

Anda Akun AWS menyertakan kuota yang menetapkan batas default jumlah penggunaan Anda. Anda dapat mengatur alarm untuk mengingatkan Anda ketika penggunaan Anda mendekati batas Anda, dan Anda dapat meminta kenaikan ke kuota, saat Anda membutuhkannya. Untuk informasi tentang cara bekerja dengan kuota, lihat topik berikut.

- [Kuota Amazon Location Service](#)
- [Menggunakan CloudWatch untuk memantau penggunaan terhadap kuota](#)
- [Memvisualisasikan kuota layanan Anda dan mengatur alarm di Panduan Pengguna Amazon CloudWatch](#) .

Anda dapat membuat alarm untuk memberi Anda peringatan awal ketika Anda hampir melampaui batas Anda. Sebaiknya setel alarm untuk setiap kuota di setiap Wilayah AWS tempat Anda menggunakan Lokasi Amazon. Misalnya, Anda dapat memantau penggunaan SearchPlaceIndexForText operasi Anda, dan membuat alarm ketika Anda melebihi 80 persen dari kuota Anda saat ini.

Ketika Anda mendapatkan peringatan alarm tentang kuota Anda, Anda harus memutuskan apa yang harus dilakukan. Anda mungkin menggunakan sumber daya tambahan karena basis pelanggan Anda telah tumbuh. Dalam hal ini, Anda mungkin ingin meminta peningkatan kuota Anda, seperti peningkatan kuota 50 persen untuk panggilan API di Wilayah tersebut. Atau, mungkin ada kesalahan dalam layanan Anda yang menyebabkan Anda melakukan panggilan tambahan yang tidak perlu ke Lokasi Amazon. Dalam hal ini Anda ingin memecahkan masalah dalam layanan Anda.

## Riwayat dokumen

Tabel berikut menjelaskan dokumentasi untuk Amazon Location Service. Untuk pemberitahuan tentang pembaruan, Anda dapat berlangganan umpan RSS.

Perubahan	Deskripsi	Tanggal
<a href="#">Amazon Location Service merilis SDK baru JavaScript</a>	Untuk mempermudah pengembangan aplikasi Lokasi Amazon dengan front end web, Amazon Location menambahkan SDK open source baru yang mendukung AWS SDK untuk JavaScript v3, menyederhanakan otentikasi dan menggunakan GeoJSON. Untuk informasi selengkapnya, lihat <a href="#">SDK Lokasi Amazon</a> .	6 Juli 2023
<a href="#">Amazon Location Service merilis kunci API ke ketersediaan umum</a>	Amazon Location menambahkan dukungan untuk tempat dan rute dan mengumumkan ketersediaan umum fitur kunci API. Untuk informasi selengkapnya, lihat <a href="#">Menggunakan kunci API</a> .	6 Juli 2023
<a href="#">Amazon Location Service menambahkan EventBridge acara Amazon untuk pembaruan posisi</a>	Amazon Location menambahkan dukungan untuk mengirim peristiwa pembaruan posisi pelacak ke EventBridge. Untuk informasi selengkapnya, termasuk cara mengaktifkan peristiwa untuk pelacak, lihat	6 Juli 2023

---

<a href="#">Bereaksi terhadap peristiwa dengan EventBridge</a>		
<a href="#">Lokasi Amazon menambahkan metadata ke geofences</a>	Menggunakan Amazon Location API, Anda sekarang dapat menambahkan properti metadata ke geofences Anda. Ini disimpan dengan geofence Anda, dan disertakan dalam acara yang terkait dengan geofence di Amazon. EventBridge Untuk informasi selengkapnya, lihat <a href="#">Menggambar geofences</a> dan <a href="#">Membuat aturan acara</a> .	15 Juni 2023
<a href="#">Lokasi Amazon menambahkan kategori untuk tempat</a>	Lokasi Amazon menambahkan kategori di hasil penelusuran tempat, dan memfilter hasil berdasarkan kategori. Untuk informasi selengkapnya, lihat <a href="#">Kategori dan pemfilteran</a> .	15 Juni 2023
<a href="#">Lokasi Amazon memperkenalkan tampilan politik</a>	Lokasi Amazon menambahkan tampilan politik ke gaya peta tertentu. Untuk informasi lebih lanjut, lihat <a href="#">Pandangan politik</a> .	23 Mei 2023
<a href="#">Amazon Location memperkenalkan demo baru dan situs sampel</a>	Amazon Location mengumumkan situs web baru yang memberi Anda akses ke demo dan sampel Lokasi Amazon. Untuk informasi selengkapnya, lihat <a href="#">situs demo Lokasi Amazon</a> .	3 Mei 2023



<a href="#">Lokasi Amazon memperkenalkan rute yang lebih panjang CalculateRouteMatrix</a>	Amazon Location sekarang memungkinkan rute panjang tak terbatas untuk rute matriks rute yang dibuat dengan penyedia data HERE. Untuk informasi selengkapnya, lihat <a href="#">Perencanaan rute yang lebih panjang</a> .	24 April 2023
<a href="#">Dokumentasi Lokasi Amazon menambahkan perbedaan fitur oleh penyedia data</a>	Dokumentasi Lokasi Amazon telah diperbarui dengan informasi mengenai perbedaan antara setiap penyedia data di Peta, pencarian Tempat, dan perutean. Untuk informasi selengkapnya, lihat <a href="#">Fitur menurut penyedia data</a> .	30 Maret 2023
<a href="#">Amazon Location Open Data memetakan ketersediaan umum</a>	Ketersediaan umum penyedia dan gaya data Amazon Location Service, berdasarkan peta OpenStreetMap Daylight. Untuk informasi selengkapnya, lihat <a href="#">Buka Data</a> .	7 Maret 2023
<a href="#">Lokasi Amazon menambahkan metode otorisasi baru dalam pratinjau</a>	Amazon Location Service menambahkan kunci API sebagai metode otorisasi baru untuk pengguna anonim, dalam mode pratinjau. Untuk informasi selengkapnya, lihat <a href="#">Mengizinkan akses tamu yang tidak diautentikasi ke aplikasi Anda menggunakan kunci API</a> .	23 Februari 2023

<a href="#">Dokumentasi Lokasi Amazon diperbarui dengan praktik terbaik IAM terbaru</a>	Dokumentasi Amazon Location Service telah diperbarui untuk memenuhi praktik AWS Identity and Access Management terbaik terbaru. Untuk informasi selengkapnya, lihat <a href="#">Keamanan di Amazon Location Service</a> .	26 Januari 2023
<a href="#">Amazon Location Service Ditambahkan GrabMaps Sebagai Penyedia Data di Asia Tenggara</a>	Amazon Location memperkenalkan GrabMaps sebagai penyedia data di Asia Tenggara. Untuk informasi lebih lanjut, lihat <a href="#">GrabMaps</a> .	10 Januari 2023
<a href="#">Amazon Location Service Membuka peta Data dalam pratinjau</a>	Penyedia dan gaya data Lokasi Amazon baru ditambahkan dalam pratinjau publik, berdasarkan peta OpenStreetMap Daylight. Untuk informasi selengkapnya, lihat <a href="#">Buka Data (Pratinjau)</a> .	Desember 15, 2022
<a href="#">Gaya citra satelit HERE baru</a>	Dua gaya peta baru telah ditambahkan untuk peta menggunakan HERE sebagai penyedia data, HERE Satellite Imagery dan HERE Hybrid map styles. Untuk informasi lebih lanjut, lihat <a href="#">DI SINI gaya peta</a> .	25 Oktober 2022
<a href="#">Unit dalam alamat</a>	Amazon Location Service sekarang mendukung unit dalam alamat, misalnya “123 Main St, Apartment 3B, Anytown, USA”.	September 20, 2022

---

<a href="#">Dapatkan tempat berdasarkan ID</a>	Amazon Location Service sekarang mencakup dukungan untuk menemukan lokasi yang tepat yang disarankan oleh SearchPlaceIndexForSuggestions operasi menggunakan GetPlace operasi. Lihat <a href="#">Menggunakan pelengkapan otomatis</a> .	September 20, 2022
<a href="#">Kunci kondisi tambahan untuk kebijakan IAM</a>	Amazon Location Service sekarang mendukung kunci kondisi tambahan yang memungkinkan Anda menyetel akses untuk geofences atau perangkat tertentu dalam kebijakan IAM. Lihat <a href="#">tombol Kondisi</a> .	23 Agustus 2022
<a href="#">Lingkaran geofences</a>	Amazon Location Service sekarang mendukung geofences yang didefinisikan sebagai lingkaran dengan titik pusat dan radius, untuk mendapatkan peristiwa ketika perangkat berada dalam jarak tertentu dari suatu lokasi. Lihat <a href="#">Menambahkan geofences melingkar</a> .	Agustus 11, 2022
<a href="#">Referensi API gabungan</a>	Amazon Location Service sekarang memiliki panduan referensi API tunggal, bukan panduan terpisah untuk setiap sublayanan. Untuk informasi selengkapnya tentang API, lihat <a href="#">Amazon Location API</a> .	Juli 7, 2022

---

<a href="#">Integrasi Service Quotas</a>	Amazon Location sekarang <a href="#">terintegrasi dengan Service Quotas</a> yang memungkinkan Anda untuk melihat dan mengelola kuota Anda melalui AWS Management Console atau menggunakan AWS CLI	6 Juli 2022
<a href="#">Bab dokumentasi konsep diperbarui</a>	<a href="#">Bab konsep Lokasi Amazon</a> telah diperbarui dengan informasi lebih lanjut untuk pengguna Lokasi Amazon.	22 April 2022
<a href="#">Tutorial mulai cepat Android baru</a>	<a href="#">Tutorial mulai cepat</a> baru untuk pengembangan Android menggunakan Kotlin telah ditambahkan untuk membuat pengembang aktif dan berjalan dengan cepat.	15 April 2022
<a href="#">Gaya Peta HERE baru</a>	Dua gaya peta baru telah ditambahkan untuk peta menggunakan HERE sebagai penyedia data. Untuk informasi lebih lanjut, lihat <a href="#">DI SINI gaya peta</a> .	15 Maret 2022
<a href="#">Merestrukturisasi dokumentasi dengan menambahkan contoh kode dan tutorial</a>	Panduan pengembang ini telah direstrukturisasi untuk menemukan topik dengan lebih mudah, termasuk bagian <a href="#">awal Cepat</a> dan <a href="#">contoh Kode</a> baru.	25 Februari 2022

[Pemfilteran posisi berbasis akurasi untuk pelacak](#)

Anda sekarang dapat menggunakan filter berbasis akurasi saat [membuat sumber daya pelacak](#).

Desember 7, 2021

[Pelengkapan otomatis untuk indeks tempat](#)

Anda sekarang dapat menggunakan [pelengkapan otomatis](#) saat mencari indeks tempat.

Desember 6, 2021

[Tutorial Amplify baru untuk menggunakan peta](#)

Tutorial baru tersedia yang menunjukkan cara menggunakan AWS Amplify untuk menampilkan peta dalam aplikasi web. Tutorial ini tersedia di [Menggunakan perpustakaan Amplify dengan Amazon Location Service](#).

24 November 2021

[Tempatkan ekstensi kueri](#)

Amazon Location Service sekarang mendukung pengaturan bahasa pilihan untuk hasil saat geocoding, atau reverse geocoding, dan menambahkan zona waktu dan informasi lainnya ke hasil. Untuk informasi selengkap nya tentang geocoding dan reverse geocoding, lihat Geocoding, [reverse geocoding](#), dan search.

November 16, 2021

[Penyaringan posisi pelacak](#)

Amazon Location Service menambahkan fitur pemfilteran posisi baru ke pelacak yang dapat membantu Anda mengontrol biaya. Fitur ini menyaring beberapa pembaruan posisi pada perangkat sebelum pembaruan disimpan atau dievaluasi terhadap geofences. Untuk informasi selengkapnya tentang pemfilteran posisi, lihat [Pelacak](#).

5 Oktober 2021

[Perbarui operasi](#)

Operasi berikut telah ditambahkan ke Referensi API Amazon Location Service: [UpdateMap](#), [UpdatePlaceIndex](#), [UpdateRouteCalculator](#), [UpdateGeofenceCollection](#), dan [UpdateTracker](#).

19 Juli 2021

[Pembaruan tutorial: fungsi yang Amazon Aurora PostgreSQL ditentukan pengguna](#)

Tutorial baru telah ditambahkan tentang cara menggunakan [fungsi yang Amazon Aurora PostgreSQL ditentukan pengguna dengan Lokasi Amazon](#) untuk memvalidasi, membersihkan, dan memperkaya data geospasial.

19 Juli 2021

<a href="#">CloudFormation Sumber daya AWS</a>	Amazon Location sekarang mendukung <a href="#">pembuatan jenis sumber daya berikut di CloudFormation sumber daya AWS</a> : AWS::Location::Map AWS::Location::PlaceIndex ,AWS::Location::RouteCalculator ,AWS::Location::Tracker ,AWS::Location::TrackerConsumer , danAWS::Location::GeofenceCollection .	7 Juni 2021
<a href="#">Sumber daya penandaan</a>	Anda sekarang dapat <a href="#">menambahkan tag ke sumber daya Lokasi Amazon</a> untuk membantu mengelola, mengidentifikasi, mengatur, mencari, dan memfilter sumber daya Anda.	1 Juni 2021
<a href="#">Ketersediaan umum</a>	Rilis ketersediaan umum dokumentasi pengembang Amazon Location Service: <a href="#">Wilayah dan titik akhir serta kuota layanan diperbarui</a> .	1 Juni 2021
<a href="#">Citra Esri</a>	Lokasi Amazon sekarang mendukung penggunaan gaya peta Esri: <a href="#">Esri Imagery</a> . Untuk informasi lebih lanjut, lihat <a href="#">Esri World Imagery</a> di situs web Esri.	1 Juni 2021

---

<a href="#">Menghitung rute</a>	Anda sekarang dapat <a href="#">menggunakan kalkulator rute Lokasi Amazon</a> untuk menghitung rute dan memperkirakan waktu perjalanan berdasarkan jaringan up-to-date jalan dan informasi lalu lintas langsung dari penyedia data pilihan Anda.	1 Juni 2021
<a href="#">AWS KMSenkripsi kunci terkelola pelanggan untuk data saat istirahat</a>	Amazon Location sekarang mendukung penggunaan kunci terkelola pelanggan simetris yang Anda buat, miliki, dan kelola untuk <a href="#">menambahkan lapisan enkripsi kedua di atas enkripsi AWS milik yang ada</a> .	1 Juni 2021
<a href="#">Rilis pratinjau publik</a>	Rilis awal dokumentasi pratinjau publik.	16 Desember 2020
<a href="#">Pembaruan tutorial: Menampilkan peta</a>	Tutorial untuk menampilkan peta menggunakan MapLibre untuk Android dan iOS telah diperbarui untuk menggunakan SDK MapLibre asli.	17 Maret 2020



# Glosarium AWS

Lihat terminologi AWS terbaru di [AWS glosarium](#) dalam Referensi Glosarium AWS.

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