



Guía para desarrolladores

Amazon Location Service



Amazon Location Service: Guía para desarrolladores

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

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

Bienvenido	1
¿Qué es Amazon Location Service?	1
Características principales	2
Servicios relacionados	3
Inicio rápido	5
Crear una aplicación web	5
Crear recursos de	6
Configurar autenticación	7
Creación de HTML	9
Agregar el mapa	12
Agregar búsqueda	16
Solicitud final	21
Sigüientes pasos	26
Crear una aplicación Android	26
Crear recursos de Amazon Location para su aplicación	27
Configurar autenticación	29
Crea la aplicación	32
Añada el mapa	32
Agregue una búsqueda	36
Añada el seguimiento	46
Sigüientes pasos	55
Crear una aplicación para iOS	55
Crear recursos de	56
Configurar autenticación	57
Crear la aplicación	60
Código inicial	61
Agrega un mapa	64
Añadir búsqueda	68
Añadir seguimiento	69
Sigüientes pasos	81
Conceptos de Amazon Location Service	82
Información general	83
Mapas	84
Estilos de mapa	85

Opiniones políticas	85
Capas personalizadas	86
Renderizado de mapas	86
Terminología de DNS	87
Búsqueda de lugares	89
Conceptos de geocodificación	90
Resultados de búsqueda	91
Múltiples resultados y relevancia	91
Resultados de direcciones	92
Almacenar los resultados de la geocodificación	94
Terminología de lugares	95
Rutas	96
Recursos para la calculadora de rutas	96
Cálculo de rutas	97
Planificación de rutas	98
Terminología de rutas	99
Geocercas y rastreadores	101
Geocercas	101
Rastreadores	103
Terminología de geocerca	107
Terminología de rastreadores	108
Casos de uso comunes	110
Participación de los usuarios y aplicaciones de geomarketing	111
Aplicaciones de seguimiento de activos	113
Aplicaciones de entrega	114
Proveedores de datos	116
Cobertura y características del proveedor de datos	116
Estilos de mapa	118
Más información	118
Esri	118
GrabMaps	127
HERE Technologies	132
Abrir datos	139
Características por proveedor de datos	149
Condiciones de uso y atribución de datos	155
Regiones y puntos de conexión	155

Regiones	155
puntos de conexión	157
Puntos de conexión de operación de la API	158
Service Quotas	160
Administración de las cuotas de Amazon Location Service	174
Desarrollo con Amazon Location	176
Escenarios y casos de uso	176
SDK y herramientas	178
SDK por lenguaje	178
MapLibre	182
Amazon Location SDK	187
API de localización de Amazon	211
Uso de Amazon Location con un AWS SDK	212
Actualizaciones de mensaje de error	212
Ejemplos de código	246
Sitio de ejemplo de Amazon Location	247
Tutorial: inicio rápido	248
Tutorial: enriquecimiento de bases de datos	249
Ejemplo: aplicación de exploración	249
Ejemplo: aplicar estilo a un mapa	250
Ejemplo: dibujar marcadores	250
Ejemplo: dibuja puntos agrupados	251
Ejemplo: dibuja un polígono	251
Ejemplo: cambiar el idioma del mapa	252
Blog: notificaciones de tiempo de entrega estimado	252
Ejemplo: actualizaciones de posición de transmisión	253
Ejemplo: aplicación móvil de geofencing and Tracking	254
Cómo utilizar Amazon Location	255
Requisitos previos de cuenta	256
Inscríbese en un Cuenta de AWS	256
Creación de un usuario con acceso administrativo	257
Conceder acceso a Amazon Location Service	258
Uso de mapas	260
Requisitos previos	261
Visualización de mapas	264
Dibujar en un mapa	320

Establece la extensión de un mapa	320
Administrar los recursos de mapas	322
Búsqueda de lugares	325
Requisitos previos	326
Geocodificación	330
Geocodificación inversa	337
Autocompletar	341
Uso de identificadores de lugares	348
Categorías y filtrado	350
Tutorial: Enriquecimiento de bases de datos	355
Administrar recursos de indexación de sitios	370
Calcular rutas	373
Requisitos previos	374
Calcule una ruta	378
Planificación de rutas	382
Posiciones no ubicadas en una carretera	388
Hora de salida	390
Modo de viaje	391
Administración de los recursos de ruta	393
Geocercas y rastreo	396
Paso 1: Agregar geocercas	398
Paso 2: Comience el rastreo	405
Paso 3: Vincular un rastreador a una colección de geocercas	420
Paso 4: Evalúe las posiciones de los dispositivos en comparación con las geocercas	421
Verifique las posiciones de los dispositivos	424
Reaccionar ante los eventos con EventBridge	426
Rastreo mediante AWS IoT y MQTT	432
Administrar los recursos de geovallas	441
Administrar los recursos del rastreador	448
Ejemplo de aplicación móvil de geofencing and Tracking	453
Etiquetado de los recursos	472
Restricciones	473
Concede permiso para etiquetar	474
Agregar una etiqueta a un recurso	475
Realice un seguimiento del costo por etiqueta	475
Controlar el acceso a los recursos de mediante etiquetas	476

Más información	477
Conceder acceso a Amazon Location	477
Usar claves de API	478
Uso de Amazon Cognito	484
Monitoreo de Amazon Location Service	495
Monitorización con CloudWatch	496
Uso CloudTrail con Amazon Location	501
Uso de AWS CloudFormation para crear recursos	505
Amazon Location y plantillas AWS CloudFormation	506
Obtener más información sobre AWS CloudFormation	506
Seguridad	508
Protección de datos	509
Privacidad de datos	510
Retención de datos	510
Cifrado de datos en reposo	511
Cifrado de datos en tránsito	524
Identity and Access Management	524
Público	524
Autenticación con identidades	525
Administración de acceso mediante políticas	529
Cómo funciona Amazon Location Service con IAM	532
Cómo funciona Amazon Location Service con usuarios no autenticados	540
Ejemplos de políticas basadas en identidades	541
Resolución de problemas	553
Respuesta frente a incidencias	556
Registro y supervisión	556
Validación de conformidad	557
Resiliencia	558
Seguridad de la infraestructura	559
Configuración y análisis de vulnerabilidades	559
Prevención del suplente confuso	559
Prácticas recomendadas de seguridad	560
Prácticas recomendadas de detección	560
Prácticas recomendadas preventivas	561
Prácticas recomendadas	562
Seguridad	562

Administración de recursos	563
Administración de costos y facturación	563
Quotas y uso	564
Historial de documentos	565
Glosario de AWS	576
.....	dlxxvii

Bienvenido a Amazon Location Service

Le damos la bienvenida a la Guía para desarrolladores de Amazon Location Service.

Los temas que encontrará a continuación pueden ayudarlo a entender la documentación dependiendo de lo que esté intentando hacer.

Obtenga información general de Amazon Location

- Obtenga más información sobre los [conceptos de Amazon Location](#).
- Profundice en la funcionalidad en el capítulo [Cómo utilizar Amazon Location Service](#).
- Consulte las aplicaciones de demostración en el [sitio de demostración de Amazon Location](#).
- Si ya tiene una Cuenta de AWS, puede usar la [consola de Amazon Location Service](#) para explorar la funcionalidad por su propia cuenta.

Use Amazon Location como desarrollador

- Cree su primera aplicación con [Inicio rápido](#).
- Descubra cómo funcionan las distintas características de Amazon Location Service en el capítulo [Cómo utilizar Amazon Location Service](#).
- Consulte los SDK y las herramientas disponibles en el capítulo [Desarrollo con Amazon Location](#).
- Consulte los [ejemplos de código y tutoriales](#) que puede usar en sus propias aplicaciones. También puede visitar la [página de ejemplos](#) del sitio de demostración de Amazon Location para encontrar ejemplos que podrá filtrar por función, idioma o plataforma.
- Obtenga información sobre las API de Amazon Location en la guía de [referencia de las API](#).

¿Qué es Amazon Location Service?

Amazon Location Service le permite agregar datos de ubicación y funciones a las aplicaciones, lo que incluye funciones como mapas, puntos de interés, geocodificación, enrutamiento, geovallas y rastreo. Amazon Location proporciona servicios basados en la ubicación (LBS) por medio de datos de alta calidad de proveedores globales de confianza como Esri, Grab y HERE. Con funciones asequibles de datos, rastreo, geolocalización y métricas integradas para el monitoreo del estado, puede crear aplicaciones sofisticadas habilitadas para la ubicación.

Con Amazon Location, mantendrá el control de los datos de su organización. Amazon Location anonimiza todas las consultas enviadas a los proveedores de datos al eliminar los metadatos de

los clientes y la información de la cuenta. Además, la información confidencial sobre la ubicación geográfica y el rastreo geográfico, como la ubicación de las instalaciones, los activos y el personal, estará siempre a salvo en su cuenta AWS. Así podrá proteger la información confidencial de terceros, la privacidad de los usuarios y reducir los riesgos de seguridad de su aplicación. Con Amazon Location, Amazon y terceros no tienen derecho a vender sus datos ni a utilizarlos con fines publicitarios.

Amazon Location está totalmente integrado con servicios como Amazon AWS CloudTrail CloudWatch EventBridge, Amazon e AWS Identity and Access Management (IAM). Amazon Location simplifica el flujo de trabajo de desarrollo mediante la integración de datos y agiliza el paso de las aplicaciones a la producción con funciones integradas de supervisión, seguridad y conformidad.

Para los aspectos destacados, los detalles del producto y los precios, consulte la página de servicio de [Amazon Location Service](#).

Características principales de Amazon Location

Amazon Location ofrece las siguientes características:

Mapas

Amazon Location Service Maps le permite visualizar la información de ubicación y es la base de muchas funcionalidades de los servicios basados en la ubicación. Amazon Location Service proporciona mosaicos de mapas de diferentes estilos procedentes de los proveedores de datos de ubicación global Esri, Grab y HERE, así como de mapas de Open Data.

Lugares

Amazon Location Service Places le permite integrar la funcionalidad de búsqueda en su aplicación, convertir direcciones en coordenadas geográficas de latitud y longitud (geocodificación) y convertir una coordenada en una dirección física (geocodificación inversa). Amazon Location Service obtiene datos geoespaciales de alta calidad de Esri, Grab y HERE para respaldar las funciones de Places.

Enrutamiento

Amazon Location Service Routes te permite buscar rutas y estimar el tiempo de viaje en función de la información sobre las up-to-date carreteras y el tráfico en tiempo real. Cree funciones

que permitan a su aplicación solicitar el tiempo, la distancia y las indicaciones de viaje entre dos ubicaciones. Calcule el tiempo y la distancia de una matriz de rutas para utilizarla en la planificación de rutas.

Geovalla

Amazon Location Service Geofences permite que su aplicación ofrezca la capacidad de detectar y actuar cuando un dispositivo entra o sale de un límite geográfico definido conocido como geovalla. Envía automáticamente un evento de entrada o salida a Amazon EventBridge cuando se detecte una brecha en la geocerca. Así podrá iniciar acciones posteriores, como enviar una notificación a un objetivo.

Rastreadores

Amazon Location Service Trackers le permite recuperar la ubicación actual y pasada de los dispositivos que han ejecutado su aplicación habilitada para rastreo. También puede vincular rastreadores con las geovallas de Amazon Location Service para comparar automáticamente las actualizaciones de ubicación de sus dispositivos con las geovallas. Los rastreadores pueden ayudarlo a reducir los costos al filtrar las actualizaciones de posición que no se han movido antes de almacenarlas o compararlas con las geovallas.

Cuando utilice rastreadores, la información confidencial sobre la ubicación de sus dispositivos rastreados no saldrá de su cuenta AWS. Así se protegerá la información confidencial de terceros, la privacidad de los usuarios y se reducirán los riesgos de seguridad.

Servicios que puede utilizar con Amazon Location

Utilice los siguientes servicios junto con Amazon Location Service.

Monitoreo y administración integrada

Amazon Location Service está integrado con Amazon CloudWatch y Amazon EventBridge para una supervisión y una gestión de datos eficientes: AWS CloudTrail

- Amazon CloudWatch: consulta las métricas sobre el uso y el estado del servicio, incluidas las solicitudes, la latencia, los errores y los registros. Para obtener más información, consulte [the section called “Monitorización con CloudWatch”](#).

- AWS CloudTrail: Registre y supervise las llamadas de la API, que incluyen las acciones realizadas por un usuario, un rol o un servicio de AWS. Para obtener más información, consulte [the section called “Uso CloudTrail con Amazon Location”](#).
- Amazon EventBridge: habilite una arquitectura de aplicaciones basada en eventos para que pueda usar AWS Lambda funciones para activar otras partes de su aplicación y flujos de trabajo. Para obtener más información, consulte [the section called “Reaccionar ante los eventos con EventBridge”](#).

Herramientas para desarrolladores

Amazon Location Service ofrece una variedad de herramientas para que los desarrolladores creen aplicaciones habilitadas para la ubicación. Estos incluyen los AWS SDK estándar, los SDK para dispositivos móviles y web y un código de muestra para combinarlos con bibliotecas de código abierto, como MapLibre. Utilice la [consola de Amazon Location Service](#) para obtener información sobre los recursos y empezar a utilizar una herramienta de aprendizaje visual e interactiva.

Inicio rápido con Amazon Location Service

La forma más eficaz de empezar a utilizar Amazon Location Service es utilizar la [consola de Amazon Location](#). Puede crear y gestionar sus recursos y probar la funcionalidad de Amazon Location a través de [la página Explorar](#).

Note

Para utilizar la consola de Amazon Location Service o seguir el resto de este tutorial, es necesario completar primero las instrucciones [Requisitos previos para utilizar Amazon Location Service](#), incluida la creación de una AWS cuenta y el acceso a Amazon Location.

Para empezar a aprender sobre las API de Amazon Location, utilice el siguiente tutorial para crear una aplicación sencilla que muestre un mapa interactivo y, también utilice la función de búsqueda. Hay tres versiones del tutorial: una muestra cómo crear una página web sencilla con Kotlin JavaScript, la segunda muestra lo mismo para una aplicación de Android que usa Kotlin y la tercera muestra lo mismo para una aplicación de iOS que usa Swift.

Temas

- [Crear una aplicación web](#)
- [Crear una aplicación Android](#)
- [Crear una aplicación para iOS](#)

Crear una aplicación web

En esta sección, creará una página web estática con un mapa y la posibilidad de buscar en una ubicación. Primero, creará sus recursos de Amazon Location y una clave de API para su aplicación.

Temas

- [Crear recursos de Amazon Location para su aplicación](#)
- [Configurar la autenticación para su aplicación](#)
- [Crear el código HTML para la aplicación](#)
- [agregar un mapa interactivo a la aplicación](#)
- [Agregar una búsqueda a su aplicación](#)

- [Ver la aplicación final](#)
- [Sigüientes pasos](#)

Crear recursos de Amazon Location para su aplicación

Si aún no los tiene, debe crear los recursos de Amazon Location que utilizará su aplicación. Aquí, cree un recurso de mapa para mostrar los mapas en su aplicación y un índice de ubicación para buscar ubicaciones en el mapa.

Para agregar recursos de ubicación a su aplicación


1. Elija el estilo de asignación que desee usar.
 - a. En la consola de Amazon Location, en la página de [Mapas](#), seleccione Crear mapa para obtener una vista previa de los estilos de mapa.
 - b. Agregue un Nombre y una Descripción para el nuevo recurso de mapa. Anote el nombre que usa para el recurso de mapa. Lo necesitará al crear el archivo de script más adelante en el tutorial.
 - c. Elija un mapa.

Note

Al elegir un estilo de mapa, también se elige el proveedor de datos de mapas que va a utilizar. Si su aplicación rastrea o enruta los activos que utiliza en su empresa, como vehículos de reparto o empleados, solo debe utilizar HERE como proveedor de geolocalización. Para obtener más información, consulte la sección 82 de los [Términos de servicio de AWS](#).

- d. Acepte los Términos y condiciones de Amazon Location y, a continuación, seleccione Crear mapa. Puede interactuar con el mapa que haya elegido: acercar, alejar o desplazarse en cualquier dirección.
 - e. Anote el nombre de recurso de Amazon (ARN) que se muestra para el nuevo recurso de mapa. Lo usará para crear la autenticación correcta más adelante en este tutorial.
2. Elija el índice de ubicación que desea usar.
 - a. En la consola de Amazon Location, en la página de [Índices de ubicación](#), seleccione Crear índice de ubicación.

- b. Agregue un Nombre y una Descripción para el nuevo recurso de índice de ubicación. Anote el nombre que usa para el recurso de índice de ubicación. Lo necesitará al crear el archivo de script más adelante en el tutorial.
- c. Elija un proveedor de datos.

 Note


En la mayoría de los casos, elija el proveedor de datos que coincida con el proveedor de mapas que ya eligió. Así, podrá garantizar que las búsquedas coincidan con los mapas.

Si su aplicación rastrea o enruta los activos que utiliza en su empresa, como vehículos de reparto o empleados, solo debe utilizar HERE como proveedor de geolocalización. Para obtener más información, consulte la sección 82 de los [Términos de servicio de AWS](#).

- d. Elija la opción de almacenamiento de datos. En este tutorial, los resultados no se almacenan, por lo que puede elegir No, solo de uso único.
- e. Acepte los Términos y condiciones de Amazon Location y, a continuación, seleccione Crear índice de ubicación.
- f. Anote el ARN que se muestra para su nuevo recurso de índice de ubicación. La usará para crear la autenticación correcta en la siguiente sección de este tutorial.

Configurar la autenticación para su aplicación

La aplicación que cree en este tutorial tiene un uso anónimo, lo que significa que sus usuarios no tienen que iniciar sesión en AWS para usar la aplicación. Sin embargo, de forma predeterminada, las API de Amazon Location Service requieren autenticación para su uso. Puede utilizar las claves de Amazon Cognito o de las API para proporcionar autenticación y autorización a los usuarios anónimos. En este tutorial, creará claves de las API para usarlas en la aplicación de muestra.

 Note

Para obtener más información sobre el uso de claves de API o Amazon Cognito con Amazon Location Service, consulte [Concesión de acceso a Amazon Location Service](#).

Para configurar la autenticación de su aplicación

1. Vaya a la [consola de Amazon Location](#) y seleccione Claves de API en el menú de la izquierda.
2. Elija Crear clave de la API.

Important

La clave de API que cree debe estar en la misma Cuenta de AWS de la región de AWS y en la misma región que los recursos de Amazon Location Service que creó en la sección anterior.

3. Rellene esta información en la página de Creación de la clave de API.
 - Nombre: un nombre para su clave de API, por ejemplo MyWebAppKey.
 - Recursos: elija el mapa y los recursos de índice de ubicación de Amazon Location que ha creado en la sección anterior. Puede agregar más de un recurso seleccionando agregar recurso. Esto permitirá que la clave de API se utilice con esos recursos.
 - Acciones: especifique las acciones que desea autorizar con esta clave de API. Debe seleccionar al menos geo: GetMap * y geo: SearchPlaceIndexForPosition para que el tutorial funcione como se espera.
 - Si lo desea, puede agregar una Descripción, una Fecha de caducidad o Etiquetas a su clave de API. También puede agregar una referencia (por ejemplo, *.example.com) para limitar el uso de la clave a un dominio concreto. Esto significará que el tutorial solo funcionará desde ese dominio.

Note

Se recomienda proteger el uso de la clave de API estableciendo una fecha de caducidad o una referencia, ambas a ser posible.

4. Seleccione Crear clave de API para crear la clave de API.
5. Elija Mostrar clave de API y copie el valor de la clave para usarla más adelante en el tutorial. Estará en el formulario `v1.public.a1b2c3d4...`

⚠ Important

Necesitará esta clave para escribir el código de su aplicación más adelante en este tutorial.

Crear el código HTML para la aplicación

En este tutorial, creará una página HTML estática que incluye un mapa y permite al usuario encontrar lo que hay en una ubicación del mapa. La aplicación constará de tres archivos: un archivo HTML y un archivo CSS para la página web, y un archivo JavaScript (.js) para el código que crea el mapa y responde a las interacciones del usuario y a los eventos del mapa.

En primer lugar, vamos a crear el marco HTML y CSS que se utilizará para la aplicación. Será una página sencilla con un elemento `<div>` que contenga el contenedor del mapa y un `<pre>` elemento que muestre las respuestas de JSON a sus consultas.

Para crear el código HTML para su aplicación de inicio rápido

1. Cree un nuevo archivo denominado `quickstart.html`.
2. Edite el archivo en el editor de texto o el entorno que prefiera. Agregue el siguiente código HTML al archivo.

```
<!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>
```

Este HTML incluye un marcador al archivo CSS que crearás en el siguiente paso, algunos elementos marcadores de posición para la aplicación y un texto explicativo.

Hay dos elementos marcadores de posición que utilizará más adelante en este tutorial. El primero es el elemento `<div id="map">`, que mantendrá el control del mapa. El segundo es el elemento `<pre id="response">`, que mostrará los resultados de la búsqueda en el mapa.

3. Guarde el archivo.

Ahora agregue el CSS para la página web. Esto establecerá el estilo del texto y los elementos marcadores de posición de la aplicación.

Para crear el CSS para su aplicación de inicio rápido

1. Cree un nuevo archivo llamado `main.css` en la misma carpeta que el archivo `quickstart.html` que ha creado en el procedimiento anterior.
2. Edite el archivo en el editor que desee usar. Agregue el siguiente texto al archivo.

```
* {
  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;
}

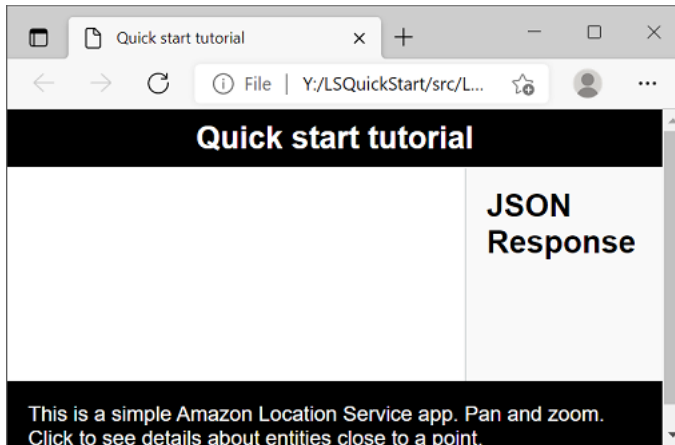
h2 {
  margin: 0;
}

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

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

Esto configura el mapa para que ocupe el espacio que no ha sido utilizado por ninguna otra aplicación, establece el área de nuestras respuestas para que ocupe el 30 % del ancho de la aplicación y, además, establece los colores y estilos para el título y el texto explicativo.

3. Guarde el archivo.
4. Ahora puede ver el archivo `quickstart.html` en un navegador para ver el diseño de la aplicación.



A continuación, agregará el control de mapa a la aplicación.

agregar un mapa interactivo a la aplicación

Ahora que tiene un marco y un marcador de posición `div`, puede agregar el control del mapa a su aplicación. Este tutorial utiliza [MapLibre GL JS](#) como control de mapa y obtiene datos de Amazon Location Service. También utilizarás el [JavaScript Ayudante de autenticación](#) para facilitar la firma de las llamadas a las API de Amazon Location con su clave de API.

Para agregar un mapa interactivo a tu aplicación

1. Abra el archivo `quickstart.html` que creó en la sección anterior.
2. Agregue referencias a las bibliotecas necesarias y al archivo de script que va a crear. Los cambios que debe realizar se muestran en **green**.

```
<!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>
```

Este procedimiento añade las siguientes dependencias a su aplicación:

- MapLibre VE A NOSOTROS. Esta biblioteca y hoja de estilos incluyen un control de mapa que muestra las teselas del mapa e incluye interactividad, como la panorámica y el zoom. El control también permite realizar extensiones, como dibujar sus propias características en el mapa.
- Cliente de Amazon Location. Esto proporciona las interfaces para la funcionalidad de Amazon Location necesaria para obtener datos de mapas y buscar lugares en el mapa. El cliente Amazon Location se basa en el AWS SDK de la JavaScript versión 3.
- Asistente de autenticación de Amazon Location. Esto proporciona funciones útiles para autenticar Amazon Location Service con claves de API o Amazon Cognito.

En este paso también se añade una referencia a `main.js`, que creará a continuación.

3. Guarde el archivo `quickstart.html`.

4. Cree un nuevo archivo llamado `main.js` en la misma carpeta que sus archivos HTML y CSS y ábralo para editarlo.
5. Agregue el siguiente script al archivo. El texto en *rojo* debe sustituirse por el valor de la clave de la API, el nombre del recurso del mapa y el nombre del recurso del lugar que creó anteriormente, así como por el identificador de la región (por ejemplo, `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();
```

Este código configura los recursos de Amazon Location, luego configura e inicializa un control de mapa MapLibre GL JS y lo coloca en tu `<div>` elemento con el id. `map`

Es importante entender la función `initializeMap()`. Crea un nuevo control de MapLibre mapa (llamado `mLglMap` localmente, pero llamado `map` en el resto del código) que se utiliza para renderizar el mapa en la aplicación.

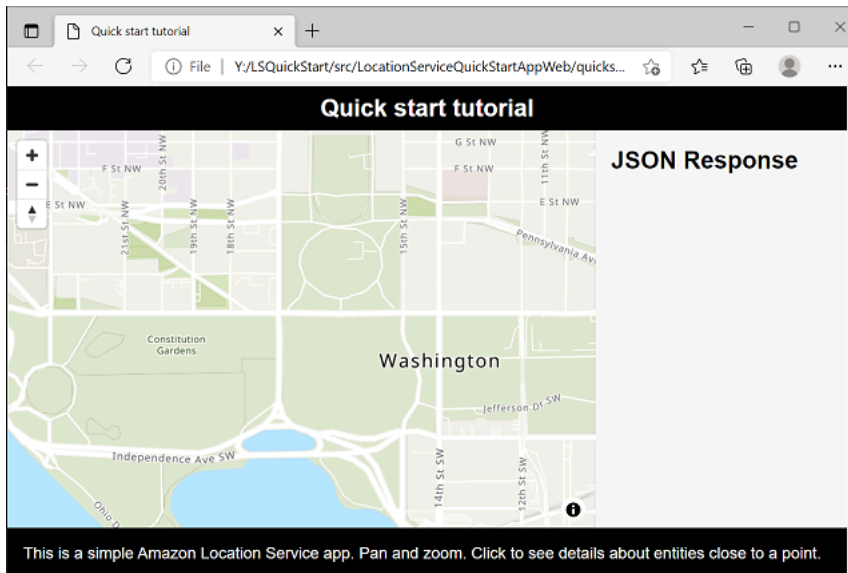
```
// Initialize the map
const mLglMap = new maplibregl.Map({
  container: "map", // HTML element ID of map element
  center: [-77.03674, 38.891602], // Initial map centerpoint
  zoom: 16, // Initial map zoom
  style: 'https://maps.geo.${region}.amazonaws.com/maps/v0/maps/${mapName}/style-
descriptor?key=${apiKey}',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

Al crear un nuevo control de MapLibre mapa, los parámetros que se transfieren indican el estado inicial del control de mapa. Aquí, se pueden establecer los siguientes parámetros.

- Contenedor HTML, que utiliza el elemento `div` del mapa en nuestro HTML.
 - El centro inicial del mapa hacia un punto de Washington, DC.
 - El nivel de zoom pasa a 16 (se amplía a un vecindario o a un nivel de bloque).
 - El estilo que se va a usar para el mapa, que proporciona MapLibre una URL que se puede usar para obtener los mosaicos del mapa y otra información para renderizarlo. Tenga en cuenta que esta URL incluye su clave de API para la autenticación.
6. Guarda JavaScript el archivo y ábrelo con un navegador. Ahora tiene un mapa en su página, donde puede usar las acciones de panorámica y zoom.

Note

Puedes usar esta aplicación para ver cómo se comporta el control del MapLibre mapa. Puede intentar usar `Ctrl` o `Shift` mientras arrastra para ver otras formas de interactuar con el mapa. Todas estas funciones son personalizables.



Su aplicación está casi completa. En la siguiente sección, se encargará de elegir una ubicación en el mapa y de mostrar la dirección de la ubicación elegida. También mostrará el JSON resultante en la página para ver los resultados completos.

Agregar una búsqueda a su aplicación

El último paso de la aplicación consiste en agregar la búsqueda en el mapa. En este caso, agregará una búsqueda de geocodificación inversa, en la que encontrará los elementos en una ubicación.

Note

Amazon Location Service también ofrece la posibilidad de buscar por nombre o dirección para encontrar la ubicación de los lugares en el mapa.

Para agregar la funcionalidad de búsqueda a su aplicación

1. Abra el archivo `main.js` que creó en la sección anterior.
2. Modifique la función `main`, como se muestra. Los cambios que debe realizar se muestran en **green**.

```
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.");
  }
});
```

```
    }  
  });  
}
```

Este código inicia la configuración del asistente de autenticación de Amazon Location para que utilice su clave de API.

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

A continuación, utiliza ese asistente de autenticación y la región que está utilizando para crear un nuevo cliente de Amazon Location.

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

Y por último, el código responde a la elección del usuario de un punto en el control del mapa. Lo hace capturando un evento MapLibre proporcionado para `click`.

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

El MapLibre `click` evento proporciona parámetros que incluyen la latitud y la longitud que el usuario eligió (`e.lngLat`). En el interior del evento `click`, el código crea el `searchPlaceIndexForPositionCommand` para buscar las entidades en la latitud y longitud determinadas.

```
// 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);
  ...
});
```

Aquí, `IndexName` es el nombre del recurso de índice de ubicación que creó anteriormente, la `Position` es la latitud y la longitud que se deben buscar, `Language` es el idioma preferido para los resultados y `MaxResults` le indica a Amazon Location que envíe solo un máximo de cinco resultados.

El código restante comprueba si hay algún error y, a continuación, muestra los resultados de la búsqueda en el `<pre>` elemento denominado `response` y muestra el resultado superior en un cuadro de alerta.

3. (Opcional) Si ahora guarda y abre el archivo `quickstart.html` en un navegador, al elegir una ubicación en el mapa se mostrará el nombre o la dirección del lugar que ha elegido.
4. El último paso de la aplicación consiste en utilizar la MapLibre funcionalidad para añadir un marcador en el lugar que el usuario haya seleccionado. Modifique la función `main` de la siguiente manera. Los cambios que debe realizar se muestran en **green**.

```
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",
    };

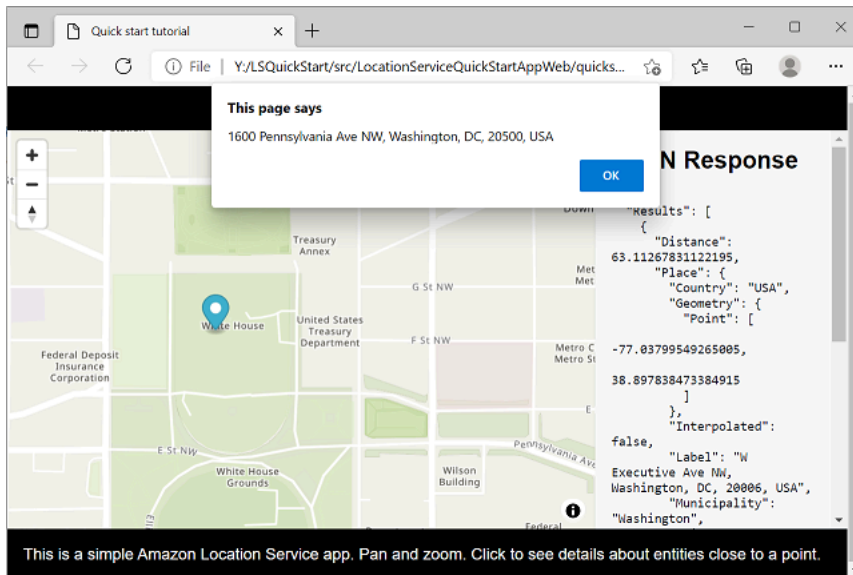
    // Set up command to search for results around clicked point
    const searchCommand = new
amazonLocationClient.SearchPlaceIndexForPositionCommand(params);

...

```

Este código declara un `marker` variable que se rellena cada vez que el usuario selecciona una ubicación y muestra dónde la seleccionó. El control del mapa representa automáticamente el marcador una vez que se agrega al mapa con `.addTo(map)`; . El código también comprueba si hay un marcador anterior y lo elimina, de modo que solo haya un marcador en la pantalla a la vez.

5. Guarde el archivo `main.js` y abra el archivo `quickstart.html` en un navegador. Puede desplazarse y hacer zoom en el mapa, como antes, pero ahora, si elige una ubicación, verá detalles sobre la ubicación que ha elegido.



La solicitud de inicio rápido está completa. Este tutorial le ha mostrado cómo crear una aplicación HTML estática que:

- Crea un mapa con el que los usuarios pueden interactuar.
- Maneja un evento de mapa (`click`).
- Llama a una API de Amazon Location Service específicamente para buscar en el mapa de una ubicación por medio de `searchPlaceIndexForPosition`.
- Utiliza el control del MapLibre mapa para añadir un marcador.

Ver la aplicación final

El código fuente final de esta aplicación se incluye en esta sección. También puedes encontrar el proyecto final [en GitHub](#).

[También puede encontrar una versión de la aplicación que utilice Amazon Cognito en lugar de las claves de API activadas. GitHub](#)

Overview

Seleccione cada pestaña para ver el código fuente final de los archivos en este tutorial de inicio rápido.

Los archivos son:

- `quickstart.html`: el marco para su aplicación, que incluye los soportes de elementos HTML para el mapa y los resultados de la búsqueda.
- `main.css`: la hoja de estilos de la aplicación.
- `main.js`: el script de la aplicación que autentica al usuario, crea el mapa y busca en un evento de `click`.

quickstart.html

El marco HTML para la aplicación de inicio rápido.

```
<!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

La hoja de estilos de la aplicación de inicio rápido.

```
* {
  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

El código de la aplicación de inicio rápido. El texto en *rojo* debe sustituirse por los nombres de los objetos de Amazon Location correspondientes.

```
// 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 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() {
```

```
// 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();
```

Siguientes pasos

Ha completado el tutorial de inicio rápido y debería entender, más o menos, cómo se utiliza Amazon Location Service para crear aplicaciones. Para sacar más provecho de Amazon Location, puede consultar los siguientes recursos:

- Profundice en los [conceptos de Amazon Location Service](#)
- Obtenga más información sobre [cómo utilizar las funciones y características de Amazon Location](#)
- Descubra cómo ampliar este ejemplo y crear aplicaciones más complejas consultando [ejemplos de código con Amazon Location](#)

Crear una aplicación Android

En esta sección, crearás una aplicación para Android con un mapa, la posibilidad de buscar en una ubicación y rastrear en primer plano. En primer lugar, creará sus recursos de Amazon Location, una identidad de Amazon Cognito y una clave de API para su aplicación.

Temas

- [Crear recursos de Amazon Location para su aplicación](#)
- [Configurar la autenticación para su aplicación](#)
- [Crear la aplicación base de Android](#)

- [agregar un mapa interactivo a la aplicación](#)
- [Agregar una búsqueda de geocodificación inversa a su aplicación](#)
- [Añadir el seguimiento a su solicitud](#)
- [Sigüientes pasos](#)

Crear recursos de Amazon Location para su aplicación

Si aún no los tiene, debe crear los recursos de Amazon Location que utilizará su aplicación. Aquí, crea un recurso de mapa para mostrar los mapas en su aplicación, un índice de lugares para buscar ubicaciones en el mapa y un rastreador para rastrear un objeto en el mapa.

Para agregar recursos de ubicación a su aplicación


1. Elija el estilo de asignación que desee usar.
 - a. En la consola de Amazon Location, en la página [Mapas](#), seleccione Crear mapa para obtener una vista previa de los estilos de mapa.
 - b. Agregue un Nombre y una Descripción para el nuevo recurso de mapa. Anote el nombre que usa para el recurso de mapa. Lo necesitará al crear el archivo de script más adelante en el tutorial.
 - c. Le recomendamos que elija el estilo de mapa HERE para su mapa.

Note

Al elegir un estilo de mapa, también se elige el proveedor de datos de mapas que va a utilizar. Si su aplicación rastrea o enruta los activos que utiliza en su empresa, como vehículos de reparto o empleados, solo debe utilizar HERE como proveedor de geolocalización. Para obtener más información, consulte la sección 82 de los [Términos de servicio de AWS](#).

- d. Acepte los Términos y condiciones de Amazon Location y, a continuación, seleccione Crear mapa. Puede interactuar con el mapa que haya elegido: acercar, alejar o desplazarse en cualquier dirección.
 - e. Anote el nombre de recurso de Amazon (ARN) que se muestra para el nuevo recurso de mapa. Lo usará para crear la autenticación correcta más adelante en este tutorial.
2. Elija el índice de ubicación que desea usar.

- a. En la consola de Amazon Location, en la página [Índices de ubicación](#), seleccione Crear índice de ubicación.
- b. Agregue un Nombre y una Descripción para el nuevo recurso de índice de ubicación. Anote el nombre que usa para el recurso de índice de ubicación. Lo necesitará al crear el archivo de script más adelante en el tutorial.
- c. Elija un proveedor de datos.

 Note

En la mayoría de los casos, elija el proveedor de datos que coincida con el proveedor de mapas que ya eligió. Así, podrá garantizar que las búsquedas coincidan con los mapas.

Si su aplicación rastrea o enruta los activos que utiliza en su empresa, como vehículos de reparto o empleados, solo debe utilizar HERE como proveedor de geolocalización. Para obtener más información, consulte la sección 82 de los [Términos de servicio de AWS](#).

- d. Elija la opción de almacenamiento de datos. En este tutorial, los resultados no se almacenan, por lo que puede elegir No, solo de uso único.
 - e. Acepte los Términos y condiciones de Amazon Location y, a continuación, seleccione Crear índice de ubicación.
 - f. Anote el ARN que se muestra para su nuevo recurso de índice de ubicación. La usará para crear la autenticación correcta en la siguiente sección de este tutorial.
3. Para crear un rastreador mediante la consola de Amazon Location.
- a. Abra la [consola de Amazon Location Service](#).
 - b. En el panel de navegación izquierdo, elija Rastreadores.
 - c. Seleccione Crear rastreador.
 - d. Rellene todos los campos obligatorios.
 - e. En Filtrado de posición, le recomendamos que utilice la configuración predeterminada: TimeBased.
 - f. Elija Crear rastreador para terminar.

Configurar la autenticación para su aplicación

La aplicación que cree en este tutorial tiene un uso anónimo, lo que significa que sus usuarios no tienen que iniciar sesión AWS para utilizar la aplicación. Sin embargo, las API de Amazon Location Service requieren autenticación para su uso. Puede usar claves de API o Amazon Cognito para proporcionar autenticación y autorización a los usuarios anónimos. En este tutorial, se utilizarán Amazon Cognito y las claves de API para autenticar la aplicación.

Note

Si desea más información sobre el uso de Amazon Cognito o de claves de API con Amazon Location Service, consulte [Concesión de acceso a Amazon Location Service](#).

Los siguientes tutoriales te muestran cómo configurar la autenticación para el mapa, el índice de lugares y el rastreador que creaste, así como configurar los permisos para Amazon Location.

Configuración de la autenticación

1. Ve a la [consola de Amazon Location](#) y selecciona las claves de API en el menú de la izquierda.
2. Haz clic en «Crear clave de API». Recuerde que la clave de API debe estar en la misma AWS cuenta y región que los recursos de Amazon Location Service creados anteriormente.
3. Rellena los detalles necesarios en la página «Crear clave de API»:
 - Nombre: proporciona un nombre para tu clave de API, por ejemplo. MyAppKey
 - Recursos: elige los recursos de Amazon Location Service Map e indexa los recursos creados anteriormente. Puede añadir varios recursos seleccionando «Añadir recurso». Esto permite que la clave de API se utilice con recursos específicos.
 - Acciones: especifique las acciones autorizadas para esta clave de API. Como mínimo, seleccione `geo:GetMap` y asegúrese de `geo:SearchPlaceIndexForPosition` que el tutorial funcione según lo previsto.
 - Opcionalmente, puede añadir una descripción, una fecha de caducidad, etiquetas o una referencia, por ejemplo, `https://www.example.com` para limitar el uso de la clave a un dominio específico y permitir que el tutorial funcione solo dentro de ese dominio.
4. Haz clic en Crear clave de API para generar la clave de API.
5. Seleccione Mostrar clave de API y copie el valor de la clave, por ejemplo, `v1.public.a1b2c3d4` para usarlo más adelante en el tutorial.

Cree una política de IAM para el seguimiento

1. Inicie sesión en la consola de IAM en <https://console.aws.amazon.com/iam/> con un usuario que tenga permisos de administrador.
2. En el panel de navegación, seleccione Políticas.
3. En el panel de contenido, elija Create policy (Crear política).
4. Elija la opción JSON y, a continuación, copie y pegue esta política de JSON en el cuadro de texto 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}"
      ]
    }
  ]
}
```

Este es un ejemplo de política de seguimiento. Para usar el ejemplo en su propia política, sustituya los Region TrackerName marcadores de posición y Account

Note

Si bien los grupos de identidades no autenticadas están pensados para ser expuestos en sitios de Internet no seguros, tenga en cuenta que se cambiarán por credenciales de AWS estándar y de tiempo limitado.

Es importante definir adecuadamente los roles de IAM asociados a los grupos de identidades no autenticadas. Para obtener más información sobre el uso y el alcance adecuado de las políticas en Amazon Cognito con Amazon Location Service, [consulte Concesión de acceso a Amazon Location Service](#).

5. En la página Revisar y crear, introduzca un nombre para el campo del nombre de la política. Revisa los permisos otorgados por tu política y, a continuación, selecciona Crear política para guardar tu trabajo.

La nueva política aparece en la lista de las políticas administradas y está lista para asociar.

Configura la autenticación para tu seguimiento

1. Configure la autenticación de la aplicación de mapas en la consola de [Amazon Cognito](#).
2. Abra la página de grupos de identidades.

Note

El grupo que cree debe estar en la misma AWS cuenta y AWS región que los recursos de Amazon Location Service que creó en la sección anterior.

3. Elija Crear grupo de identidades.
4. Empezando por el paso de configuración de la confianza del grupo de identidades. Para la autenticación del acceso de los usuarios, seleccione Acceso de invitado y pulse Siguiente.
5. En la página Configurar permisos, seleccione Usar un rol de IAM existente e introduzca el nombre del rol de IAM que creó en el paso anterior. Cuando esté listo, pulse Siguiente para pasar al siguiente paso.
6. En la página Configurar propiedades, proporciona un nombre para tu grupo de identidades. A continuación, pulse Siguiente.
7. En la página Revisar y crear, revise toda la información presente y, a continuación, pulse Crear grupo de identidades.

8. Abra la página Grupos de identidades y seleccione el grupo de identidades que acaba de crear. A continuación, copie o anote lo IdentityPoolId que utilizará más adelante en el script de su navegador.

Crear la aplicación base de Android

En este tutorial, crearás una aplicación para Android que incorpore un mapa y permita al usuario encontrar lo que hay en una ubicación del mapa.

En primer lugar, crea una aplicación Kotlin vacía con el nuevo asistente de proyectos de Android Studio.

Para crear una aplicación vacía () AndroidStudio

1. Empezar AndroidStudio. Abre el menú y selecciona Archivo, Nuevo o Nuevo proyecto.
2. En la pestaña Teléfono y tableta, seleccione Actividad vacía y, a continuación, seleccione Siguiente.
3. Elija un Nombre, un nombre de paquete y una ubicación de almacenamiento para su aplicación.
4. En la lista desplegable de Idioma, seleccione Kotlin.
5. Seleccione Finalizar para crear su aplicación en blanco.

agregar un mapa interactivo a la aplicación

Ahora que ha creado una aplicación básica, puede añadir el control del mapa a la aplicación. Este tutorial usa claves de API para administrar la vista del mapa. El control del mapa en sí mismo forma parte de la [biblioteca MapLibre nativa](#), con la clave de API y MapLibre, y los datos del mapa provienen de Amazon Location.

Para añadir un mapa a su aplicación, tendrá que realizar las siguientes acciones:

- Añada la MapLibre dependencia a su proyecto.
- Configura el código de visualización del mapa con compose.
- Escribe el código para mostrar el mapa.

Use el siguiente procedimiento para agregar el mapa a la aplicación:

1. Agregue la MapLibre dependencia a su proyecto

- a. En AndroidStudio, seleccione el menú Ver y elija Ventanas de herramientas, Proyecto. Esto abrirá la ventana de Proyecto, que le da acceso a todos los archivos del proyecto.
- b. En la ventana de Proyecto, abra Gradle y, a continuación, abra el `libs.versions.toml` archivo en la vista de árbol. Esto abrirá el archivo `libs.versions.toml` para editarlo. Ahora agrega la siguiente versión y los datos de las bibliotecas en el `libs.versions.toml` archivo.

```
[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. Una vez que termine de editar el `libs.versions.toml` archivo, AndroidStudio debe volver a sincronizar el proyecto. En la parte superior de la ventana de `libs.versions.toml` edición, AndroidStudio le pide que sincronice. Selecciona «Sincronizar ahora» para sincronizar tu proyecto antes de continuar.
- d. En la ventana del proyecto, abra Gradle Scripts en la vista de árbol y selecciona el `build.gradle` archivo para el módulo de la aplicación. Esto abrirá el archivo `build.gradle` para editarlo.
- e. En la parte inferior del archivo, en la sección de dependencias, agrega la siguiente dependencia.

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

- f. Cuando termines de agregar las dependencias de Gradle, Android Studio debe volver a sincronizar el proyecto. En la parte superior de la ventana de edición de `build.gradle`,

en Android Studio, selecciona Sincronizar ahora para sincronizar el proyecto antes de continuar.

2. Ahora configurarás el código de visualización del mapa con compose. Utilice los siguientes pasos:
 - a. Desde la ventana del proyecto, abre App, Java, el *nombre de tu paquete* en la vista de árbol y ve a la carpeta ui, dentro de la carpeta ui, crea un directorio de vistas.
 - b. Dentro del directorio de vistas, crea un MapLoadScreen.kt archivo.
 - c. Agregue el siguiente código al archivo MapLoadScreen.kt.

```
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. Escribe el código para mostrar el mapa.

a. Agregue el siguiente código al archivo MainActivity.kt.

```
// ...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)
        setContentView {
            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. Guarde el archivo `MainActivity.kt`. Ahora puede crear la aplicación. Para ejecutarlo, puede que tengas que configurar un dispositivo para emularlo o usar la aplicación en tu dispositivo. Android Studio Usa esta aplicación para ver cómo se comporta el control del mapa. Puedes desplazarte arrastrándolo por el mapa y pellizcándolo para ampliarlo.

En la siguiente sección, añadirá un marcador al mapa y mostrará la dirección de la ubicación en la que se encuentra el marcador a medida que mueve el mapa.

Agregar una búsqueda de geocodificación inversa a su aplicación

Ahora agregará la búsqueda de geocodificación inversa a su aplicación, donde encontrará los elementos en una ubicación. Para simplificar el uso de una aplicación de Android, buscaremos en el centro de la pantalla. Para encontrar una nueva ubicación, mueva el mapa a donde quiera buscar. Colocaremos un marcador en el centro del mapa para mostrar dónde estamos buscando.

Añadir una búsqueda de geocodificación inversa constará de dos partes.

- Añada un marcador en el centro de la pantalla para mostrar al usuario dónde estamos buscando.
- Añade un cuadro de texto para ver los resultados, busca lo que está en la ubicación del marcador y muéstralo en el cuadro de texto.

Para agregar un marcador a la aplicación

1. Guarda esta imagen en tu proyecto en la `app/res/drawable` carpeta como `red_marker.png` (también puedes acceder a la imagen desde) [GitHub](#). Como alternativa, puedes crear tu imagen. También puedes usar un archivo.png con transparencia para las partes que no quieras que se muestren.
2. Añada el siguiente código a su `MapLoadScreen` archivo.kt.


```
// ...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. Cree y ejecute su aplicación para obtener una vista previa de la funcionalidad.

Su aplicación ahora tiene un marcador en la pantalla. En este caso, se trata de una imagen estática que no se mueve. Se usa para mostrar el centro de la vista del mapa, que es donde buscaremos. En el siguiente procedimiento, añadiremos la búsqueda en esa ubicación.

Para agregar la búsqueda de geocodificación inversa en una ubicación a su aplicación

1. En la ventana del proyecto, abra Gradle para `libs.versions.toml` archivar en la vista de árbol. Esto abrirá el archivo `libs.versions.toml` para editarlo. Ahora agrega la siguiente versión y los datos de las bibliotecas en el `libs.versions.toml` archivo.

```
[versions]  
...  
okhttp = "4.12.0"  
  
[libraries]  
...  
com-squareup-okhttp3 = { group = "com.squareup.okhttp3", name = "okhttp",  
    version.ref = "okhttp" }  
  
[plugins]  
...
```

2. Una vez que termine de editar el `libs.versions.toml` archivo, AndroidStudio debe volver a sincronizar el proyecto. En la parte superior de la ventana de `libs.versions.toml` edición, AndroidStudio le pide que sincronice. Selecciona «Sincronizar ahora» para sincronizar tu proyecto antes de continuar.
3. En la ventana del proyecto, abra Gradle Scripts en la vista de árbol y selecciona el `build.gradle` archivo para el módulo de la aplicación. Esto abrirá el archivo `build.gradle` para editarlo.
4. En la parte inferior del archivo, en la sección de dependencias, agrega la siguiente dependencia.

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

```
}
```

5. Cuando termines de editar las dependencias de Gradle, AndroidStudio debes volver a sincronizar el proyecto. En la parte superior de la ventana de build.gradle edición, te AndroidStudio pide que sincronices. Seleccione esta opción SyncNow para sincronizar el proyecto antes de continuar.
6. Ahora, en la vista de árbol, agrega los datos al directorio de solicitudes y crea la clase de ReverseGeocodeRequest.kt datos. Añada el siguiente código a la clase.

```
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. Ahora, en la vista de árbol, agregue los datos al directorio de respuesta y cree la clase de ReverseGeocodeResponse.kt datos. Añada el siguiente código en su interior.

```
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. Ahora, desde la ventana del proyecto, abre App, Java, el *nombre de tu paquete* en la vista de árbol, y ve a la carpeta ui, dentro de la carpeta ui, crea el directorio ViewModel.

9. Dentro del directorio ViewModel, cree MainViewModel.kt el archivo.
10. Agregue el siguiente código al archivo MainViewModel.kt.

```
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
            }
        }
    }
})
}
}

```

- Si aún no está abierto, abra e archivo `MapLoadScreen.kt` como en el procedimiento anterior. Añada el código siguiente. Esto creará una vista de texto compuesto en la que verá los resultados de búsqueda con geocodificación inversa en la ubicación.

```

// ...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. En la aplicación, en java, en la carpeta con el nombre del paquete AndroidStudio, abre el MainActivity.kt archivo. Modifique el código como se muestra.

```

// ...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
                )
            }
        }
    }
}

```

Este código funciona con la vista de mapa. La posición de la cámara virtual define la vista del mapa MapLibre. Se puede pensar que mover el mapa es como mover esa cámara virtual.

- ViewModel tiene una variable de etiqueta: esta variable establece los datos en la vista de texto redactado.
- onMapReady tiene una variable de etiqueta: esta variable establece los datos en la vista de redacción de texto. ----SEP----: Esta función se actualiza para registrar dos nuevos eventos.
- El onCameraMove evento ocurre cada vez que el usuario mueve el mapa. En general, al mover el mapa, queremos ocultar la búsqueda hasta que el usuario termine de mover el mapa.
- El onCameraIdle evento se produce cuando el usuario hace una pausa para mover el mapa. Este evento llama a nuestra función de geocodificación inversa para buscar en el centro del mapa.
- reverseGeocode(latLng: LatLng, apiKey: String): Esta función, llamada en ese caso onCameraIdle, busca una ubicación en el centro del mapa y actualiza la etiqueta para mostrar los resultados. Utiliza el objetivo de la cámara, que define el centro del mapa (hacia donde mira la cámara).

13. Guarda tus archivos y crea y ejecuta tu aplicación para comprobar si funciona.

Tu aplicación de inicio rápido con función de búsqueda está completa.

Añadir el seguimiento a su solicitud

Para añadir el seguimiento a tu aplicación de muestra, sigue estos pasos:

1. Agrega las dependencias del SDK de seguimiento y autenticación a tu proyecto.
 2. Incluye entradas de permisos y servicios en tu `AndroidManifest` archivo.xml.
 3. Configura el código del botón de iniciar/detener el seguimiento con `compose`.
 4. Agrega código para crear un `LocationTracker` objeto e inicia y detiene el rastreo.
 5. Crea una ruta de prueba con el emulador de Android.
1. Agrega dependencias del SDK de seguimiento y autenticación a tu proyecto.
 - a. En la ventana del proyecto, abre Gradle y, a continuación, abre el `libs.versions.toml` archivo en la vista de árbol. Esto abrirá el archivo `libs.versions.toml` para editarlo. Ahora agrega la siguiente versión y los datos de las bibliotecas en el `libs.versions.toml` archivo.

```
[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. Una vez que termine de editar el `libs.versions.toml` archivo, AndroidStudio debe volver a sincronizar el proyecto. En la parte superior de la ventana de `libs.versions.toml` edición, AndroidStudio le pide que sincronice. Selecciona «Sincronizar ahora» para sincronizar tu proyecto antes de continuar.
- c. En la ventana del proyecto, abre Gradle Scripts en la vista de árbol y selecciona el `build.gradle` archivo para el módulo de la aplicación. Esto abrirá el archivo `build.gradle` para editarlo.

- d. En la parte inferior del archivo, en la sección de dependencias, agrega la siguiente dependencia.

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

- e. Cuando termines de editar las dependencias de Gradle, AndroidStudio debes volver a sincronizar el proyecto. En la parte superior de la ventana de edición de build.gradle, te pide que sincronices. AndroidStudio Selecciona esta opción SyncNow para sincronizar tu proyecto antes de continuar.

2. Incluye entradas de permisos y servicios en tu AndroidManifest archivo.xml.

- Para incluir las entradas de permisos y servicios correctas en el suyoAndroidManifest.xml file, actualice el archivo con el siguiente código:

```
<?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. Configura el código del botón de iniciar/detener el seguimiento con compose.
 - a. Agrega dos imágenes de Play y Pause en resolución debajo de un elemento de dibujo denominado `ic_pause` e `ic_play`. También puedes acceder a la imagen desde. [GitHub](#)
 - b. Si aún no está abierto, abra e archivo `MapLoadScreen.kt` como en el procedimiento anterior. Añada el código siguiente. Esto creará una vista de botones de composición en la que podremos hacer clic en él para iniciar y detener el seguimiento.

```
// ...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. Añada un código para crear un `LocationTracker` objeto e iniciar y detener el rastreo.

a. Añada el siguiente código dentro del `MainViewModel.kt` archivo.

```

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

```

b. Agregue el siguiente código al archivo `MainActivity.kt`.

```

// ...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)
        setContentView {
            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:
ListLocationEntry) {
        }

        override fun onUploaded(entries:
ListLocationEntry) {
        }
    })
}
})
}
}
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
            )
        }
    }
}
}
```

El código anterior muestra cómo crear un `LocationTracker` objeto con él `AuthHelper` y cómo iniciar y detener el rastreo con él `LocationTracker`.

- `authenticateUser()`: Este método crea `AuthHelper` `LocationTracker` objetos.
- `onStartStopTrackingClick`: Esta llamada se activa cuando el usuario hace clic en el botón de iniciar/detener el seguimiento, que iniciará o detendrá el seguimiento con el SDK de seguimiento.

5. Crea una ruta de prueba con el emulador de Android.

- Abre el emulador iniciando el AVD con Android Studio.
- Abre los controles ampliados haciendo clic en el icono Más (tres puntos) de la barra de herramientas del emulador.
- Abre Ubicación seleccionando Ubicación en la barra lateral.

- d. Cree una ruta con datos GPX o haga clic en el mapa y elija los datos de origen y destino.
- e. Inicie la simulación haciendo clic en PLAY ROUTE para empezar a simular la ruta GPS.
- f. Pruebe la aplicación ejecutándola y observando cómo gestiona la ruta simulada.

Esta es la demostración completa de la aplicación Android Quick Start.

Siguientes pasos

El código fuente de esta aplicación está disponible en [GitHub](#).

Para sacar más provecho de Amazon Location, puede consultar los siguientes recursos:

- Profundice en los [conceptos de Amazon Location Service](#)
- Obtenga más información sobre [cómo utilizar las funciones y características de Amazon Location](#)
- Descubra cómo ampliar este ejemplo y crear aplicaciones más complejas consultando [ejemplos de código con Amazon Location](#)

Crear una aplicación para iOS

En esta sección, creará una aplicación para iOS con la capacidad de buscar en una ubicación y rastrear en primer plano. En primer lugar, creará sus recursos de Amazon Location y una identidad de Amazon Cognito para su aplicación.

Temas

- [Crear recursos de Amazon Location para su aplicación](#)
- [Configurar la autenticación para su aplicación](#)
- [Creación de la aplicación iOS base](#)
- [Configurando el código inicial](#)
- [agregar un mapa interactivo a la aplicación](#)
- [Agregar una búsqueda a su aplicación](#)
- [Añadir el seguimiento a su aplicación](#)
- [Siguientes pasos](#)

Crear recursos de Amazon Location para su aplicación

Si aún no los tiene, debe crear los recursos de Amazon Location que utilizará su aplicación. Creará un recurso cartográfico para mostrar los mapas en su aplicación, un índice de lugares para buscar ubicaciones en el mapa y un rastreador para rastrear un objeto a lo largo del mapa.

Para agregar recursos de ubicación a su aplicación


1. Elija el estilo de asignación que desee usar.
 - a. En la consola de Amazon Location, en la página [Mapas](#), seleccione Crear mapa para obtener una vista previa de los estilos de mapa.
 - b. Agregue un Nombre y una Descripción para el nuevo recurso de mapa. Anote el nombre que usa para el recurso de mapa. Lo necesitará al crear el archivo de script más adelante en el tutorial.
 - c. Elija un mapa.

Note

Al elegir un estilo de mapa, también se elige el proveedor de datos de mapas que va a utilizar. Si su aplicación rastrea o enruta los activos que utiliza en su empresa, como vehículos de reparto o empleados, solo debe utilizar HERE como proveedor de geolocalización. Para obtener más información, consulte la sección 82 de los [Términos de servicio de AWS](#).

- d. Acepte los Términos y condiciones de Amazon Location y, a continuación, seleccione Crear mapa. Puede interactuar con el mapa que haya elegido: acercar, alejar o desplazarse en cualquier dirección.
 - e. Anote el nombre de recurso de Amazon (ARN) que se muestra para el nuevo recurso de mapa. Lo usará para crear la autenticación correcta más adelante en este tutorial.
2. Elija el índice de ubicación que desea usar.
 - a. En la consola de Amazon Location, en la página [Índices de ubicación](#), seleccione Crear índice de ubicación.
 - b. Agregue un Nombre y una Descripción para el nuevo recurso de índice de ubicación. Anote el nombre que usa para el recurso de índice de ubicación. Lo necesitará al crear el archivo de script más adelante en el tutorial.

- c. Elija un proveedor de datos.

 Note

En la mayoría de los casos, elija el proveedor de datos que coincida con el proveedor de mapas que ya eligió. Así, podrá garantizar que las búsquedas coincidan con los mapas.

Si su aplicación rastrea o enruta los activos que utiliza en su empresa, como vehículos de reparto o empleados, solo debe utilizar HERE como proveedor de geolocalización. Para obtener más información, consulte la sección 82 de los [Términos de servicio de AWS](#).

- d. Elija la opción de almacenamiento de datos. En este tutorial, los resultados no se almacenan, por lo que puede elegir No, solo de uso único.
 - e. Acepte los Términos y condiciones de Amazon Location y, a continuación, seleccione Crear índice de ubicación.
 - f. Anote el ARN que se muestra para su nuevo recurso de índice de ubicación. La usará para crear la autenticación correcta en la siguiente sección de este tutorial.
3. Para crear un rastreador mediante la consola de Amazon Location.
 - a. Abre la [consola de Amazon Location Service](#).
 - b. En el panel de navegación izquierdo, elija Rastreadores.
 - c. Seleccione Crear rastreador.
 - d. Rellene todos los campos obligatorios.
 - e. En Filtrado de posición, le recomendamos que utilice la configuración predeterminada: TimeBased.
 - f. Elija Crear rastreador para terminar.

Configurar la autenticación para su aplicación

La aplicación que cree en este tutorial tiene un uso anónimo, lo que significa que sus usuarios no tienen que iniciar sesión AWS para utilizar la aplicación. Sin embargo, las API de Amazon Location Service requieren autenticación para su uso. Utilizará Amazon Cognito para proporcionar autenticación y autorización a los usuarios anónimos. En este tutorial, se utilizará Amazon Cognito para autenticar la aplicación.

Note

Para obtener más información sobre el uso de Amazon Cognito con Amazon Location Service, consulte [Concesión de acceso a Amazon Location Service](#)


Los siguientes tutoriales te muestran cómo configurar la autenticación para el mapa, el índice de lugares y el rastreador que creaste, así como configurar los permisos para Amazon Location.

Creación de una política de IAM para el seguimiento

1. Inicie sesión en la consola de IAM en <https://console.aws.amazon.com/iam/> con un usuario que tenga permisos de administrador.
2. En el panel de navegación, seleccione Políticas.
3. En el panel de contenido, elija Create policy (Crear política).
4. Elija la opción JSON y, a continuación, copie y pegue esta política de JSON en el cuadro de texto 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}"
      ]
    }
  ]
}
```

Este es un ejemplo de política de seguimiento. Para usar el ejemplo en su propia política, sustituya los TrackerName marcadores de posición Region AccountIndexName, MapName y.

 Note

Si bien los grupos de identidades no autenticadas están pensados para ser expuestos en sitios de Internet no seguros, tenga en cuenta que se cambiarán por credenciales estándar y de duración limitada. AWS


Es importante definir adecuadamente los roles de IAM asociados a los grupos de identidades no autenticadas. Para obtener más información sobre el uso y el alcance adecuado de las políticas en Amazon Cognito con Amazon Location Service, [consulte Concesión de acceso a Amazon Location Service](#).

5. En la página Revisar y crear, introduzca un nombre para el campo del nombre de la política. Revisa los permisos otorgados por tu política y, a continuación, selecciona Crear política para guardar tu trabajo.

La nueva política aparece en la lista de las políticas administradas y está lista para asociar.

Configura la autenticación para tu seguimiento

1. Configure la autenticación de la aplicación de mapas en la consola de [Amazon Cognito](#).
2. Abra la página de grupos de identidades.

 Note

El grupo que cree debe estar en la misma AWS cuenta y AWS región que los recursos de Amazon Location Service que creó en la sección anterior.

3. Elija Crear grupo de identidades.
4. Empezando por el paso de configuración de la confianza del grupo de identidades. Para la autenticación del acceso de los usuarios, seleccione Acceso como invitado y pulse Siguiente.
5. En la página Configurar permisos, seleccione Usar un rol de IAM existente e introduzca el nombre del rol de IAM que creó en el paso anterior. Cuando esté listo, pulse Siguiente para pasar al siguiente paso.

6. En la página Configurar propiedades, proporciona un nombre para tu grupo de identidades. A continuación, pulse Siguiente.
7. En la página Revisar y crear, revise toda la información presente y, a continuación, pulse Crear grupo de identidades.
8. Abra la página Grupos de identidades y seleccione el grupo de identidades que acaba de crear. A continuación, copie o anote lo IdentityPoolId que utilizará más adelante en el script de su navegador.

Creación de la aplicación iOS base

En este tutorial, crearás una aplicación para iOS que incorpore un mapa y permita al usuario encontrar lo que hay en una ubicación del mapa.

Primero, creemos una aplicación Swift usando el asistente de proyectos de Xcode.

Para crear una aplicación vacía (Xcode)

1. Abre Xcode y, en el menú, selecciona Archivo, Nuevo o Nuevo proyecto.
2. En la pestaña iOS, selecciona Aplicación y, a continuación, selecciona Siguiente.
3. Introduce un nombre de producto, un identificador de organización y una entrada en el campo InterfazSwiftUI. Seleccione Siguiente para finalizar la selección.
4. Seleccione una ubicación donde guardará su proyecto y pulse el botón de creación para crear la aplicación vacía.

Una vez que haya creado la aplicación base, necesitará instalar los paquetes necesarios para la aplicación de muestra.

Instalar las dependencias necesarias

1. En Xcode, haz clic con el botón derecho en el proyecto y selecciona Añadir paquetes... . Se abrirá la ventana Paquetes, donde podrá añadir paquetes a su proyecto.
2. En la ventana Paquetes, añada los siguientes paquetes:
 - Para el paquete nativo de Maplibre, utilice esta URL: <https://github.com/maplibre/maplibre-gl-native-distribution>. Desde la URL, añada estos paquetes:maplibre-gl-native-distribution, yMapbox.

- Para el SDK de autenticación de Amazon Location para iOS, usa esta URL: <https://github.com/aws-geospatial/amazon-location-mobile-auth-sdk-ios>. Desde la URL, agrega estos paquetes:amazon-location-mobile-auth-sdk-ios, y. AmazonLocationiOSErrorAuthSDK
- Para el SDK de seguimiento de ubicaciones de Amazon para iOS, usa esta URL: <https://github.com/aws-geospatial/amazon-location-mobile-tracking-sdk-ios>. Desde la URL, agrega estos paquetes:amazon-location-mobile-tracking-sdk-ios, y. AmazonLocationiOSTrackingSDK

Configurando el código inicial

Habilita los permisos de ubicación en tu aplicación

1. Abre tu proyecto de Xcode.
2. Localiza el Info.plist archivo del proyecto.
3. Agrega las claves necesarias para los permisos de ubicación en función de los requisitos de tu aplicación. Estas son las claves:
 - `NSLocationWhenInUseUsageDescription`: Descripción de por qué tu aplicación necesita acceso a la ubicación cuando está en uso.
 - `NSLocationAlwaysAndWhenInUseUsageDescription`: Descripción de por qué tu aplicación necesita un acceso continuo a la ubicación.

Ahora tendrá que configurar los valores de los recursos en su aplicación. Añade un nuevo archivo con el nombre `Config.xcconfig` y rellena los valores que creaste anteriormente en la consola de Amazon.

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

1. En la sección del navegador del lado izquierdo, selecciona el proyecto.
2. En la sección de objetivos, selecciona tu aplicación y haz clic en la pestaña de información.
3. Añade propiedades de información con valores como los siguientes:

4. Añade el `Config.swift` archivo con el contenido siguiente, que leerá los valores de configuración del archivo de información del paquete.

```
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. Crea una nueva carpeta con el nombre `ViewModel` y añade un `TrackingViewModel.swift` archivo en su interior.

```
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
}
}
```

agregar un mapa interactivo a la aplicación

Ahora añadirá el control del mapa a su aplicación. Este tutorial usa MapLibre la AWS API para administrar la vista del mapa en la aplicación. El control del mapa en sí es parte de la biblioteca [iOS nativa de MapLibre GL](#).

1. Agregue el `MapView.swift` archivo a la carpeta Vistas con el siguiente código:

```
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
        }
    }
}

```



```
        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. Añada el `TrackingView.swift` archivo con el siguiente código:

```
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)
            }
        }
    }
}
```

Ahora puede crear la aplicación. Para ejecutarlo, puede que tengas que configurar un dispositivo para emularlo en Xcode o usar la aplicación en tu dispositivo. Usa esta aplicación para ver cómo se

comporta el control del mapa. Puedes desplazarte arrastrando el mapa y pellizcando para ampliarlo. Por tu cuenta, puedes cambiar el funcionamiento del control del mapa para personalizarlo según las necesidades de tu aplicación.

Agregar una búsqueda a su aplicación

Ahora agregará la búsqueda de geocodificación inversa a la aplicación, donde encontrará los elementos en una ubicación. Para simplificar el uso de una app para iOS, buscaremos en el centro de la pantalla. Para encontrar una nueva ubicación, mueva el mapa a donde quiera buscar. Colocaremos un marcador en el centro del mapa para mostrar dónde estamos buscando.

1. Agregue el siguiente código en el archivo `TrackingViewModel.swift` que está relacionado con la búsqueda de geocodificación inversa

```
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. Actualice el `TrackingView.swift` archivo con el siguiente código, que mostrará la dirección de la ubicación centrada de la vista de mapa

```
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)
                    }
                }
            }
        }
    }
}
```

Añadir el seguimiento a su aplicación

El último paso de la aplicación consiste en añadir la funcionalidad de seguimiento a la aplicación. En este caso, añadirás los puntos de seguimiento para iniciar, detener el seguimiento, recuperar y mostrar los puntos de seguimiento en tu aplicación.

1. Agrega el `TrackingBottomView.swift` archivo a tu proyecto. Tiene un botón que inicia y detiene el seguimiento de las ubicaciones de los usuarios y muestra los puntos de seguimiento en el mapa.

```
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. Actualice TrackingView.swift el archivo con el siguiente código

```

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)
                    TrackingBottomView(trackingViewModel: trackingViewModel)
                }
            }
            else {
                Text("Loading...")
            }
        }
        .onAppear() {
            if !trackingViewModel.identityPoolId.isEmpty {
                Task {
                    do {
                        try await trackingViewModel.authWithCognito(identityPoolId:
trackingViewModel.identityPoolId)
                    }
                    catch {
                        print(error)
                    }
                }
            }
        }
    }
}

```

3. Añada el siguiente código en el `TrackingViewModel.swift` archivo. Estas funciones son responsables de iniciar y detener el seguimiento. También mostrará una alerta de error si se deniega el permiso de ubicación del usuario.
4. Para implementar el seguimiento en primer plano, copia y pega el siguiente ejemplo de código:

```

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

`startTracking` solicitará el permiso de ubicación del usuario. La aplicación debe usar los permisos «Cuando esté en uso» o «Solo una vez». De lo contrario, la aplicación mostrará un error de permiso denegado.

Para obtener y mostrar las ubicaciones de rastreo, siga este procedimiento:

1. Para obtener las ubicaciones del dispositivo del usuario, debe proporcionar la fecha y la hora de inicio y finalización. Una sola llamada devuelve un máximo de 100 ubicaciones de seguimiento, pero si hay más de 100 ubicaciones de seguimiento, devolverá un valor `nextToken`. Deberás llamar a las siguientes llamadas de `getTrackerDevice` 'Ubicación' con `nextToken` para cargar más puntos de seguimiento para la hora de inicio y finalización determinada.

```

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. Ahora reemplaza el código del `MapView.swift` archivo por el siguiente código:

```

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]?)
}

```

Para localizar valores de cadena, utilice el siguiente procedimiento.

1. Cree y añada un archivo nuevo llamado `Localizable.xcstrings`.
2. Haga clic con el botón derecho en el `Localizable.xcstrings` archivo y ábralo como Código fuente.
3. Sustituya su contenido por lo siguiente:

```

{
    "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. Guarde los archivos y cree y ejecute la aplicación para obtener una vista previa de la funcionalidad.
5. Conceda el permiso de ubicación y pulse el botón de rastreo. La aplicación empezará a subir las ubicaciones de los usuarios y las subirá al rastreador de ubicaciones de Amazon. También mostrará los cambios de ubicación del usuario, los puntos de seguimiento y la dirección actual en el mapa.

Su solicitud de inicio rápido está completa. En este tutorial se muestra cómo crear una aplicación para iOS que:

- Crea un mapa con el que los usuarios pueden interactuar.
- Gestiona varios eventos del mapa asociados al cambio de la vista del mapa por parte del usuario.
- Llama a una API de Amazon Location Service, específicamente para buscar en el mapa una ubicación, mediante la `searchByPosition` API de Amazon Location.

Siguientes pasos

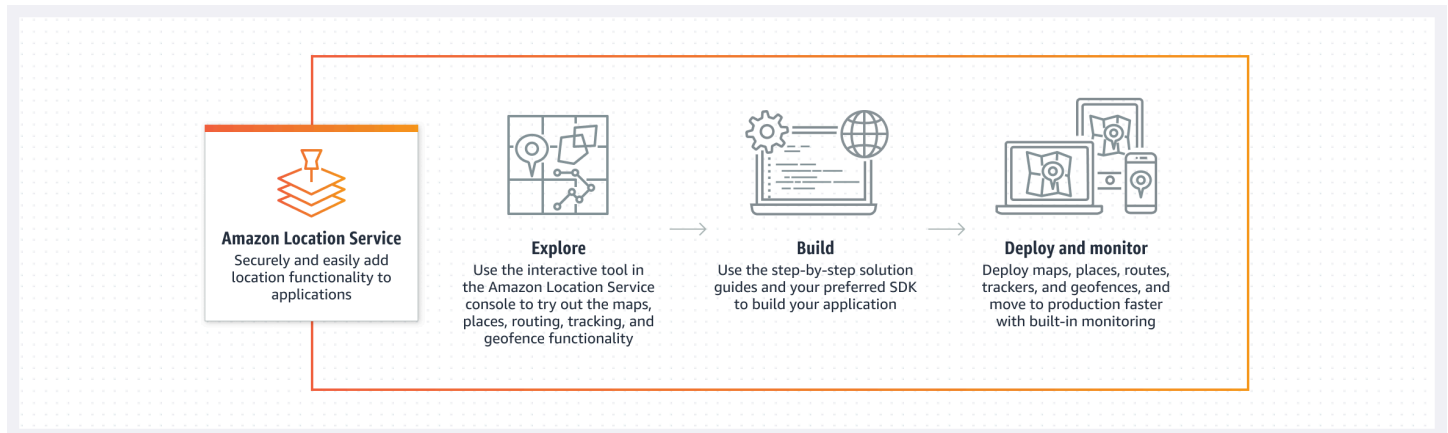
El código fuente de esta aplicación está disponible en [GitHub](#).

Para sacar más provecho de Amazon Location, puede consultar los siguientes recursos:

- Profundice en los [conceptos de Amazon Location Service](#)
- Obtenga más información sobre [cómo utilizar las funciones y características de Amazon Location](#)
- Descubra cómo ampliar este ejemplo y crear aplicaciones más complejas consultando [ejemplos de código con Amazon Location](#)

Conceptos de Amazon Location Service

Con Amazon Location Service, puede agregar datos de ubicación a su aplicación de forma segura. Explore algunas de las capacidades mediante la [herramienta visual e interactiva](#), disponible en la consola de Amazon Location. Con la herramienta de exploración, puede manipular un mapa predeterminado, buscar puntos de interés, dibujar geocercas alrededor de las áreas de interés y simular el envío de la ubicación de los dispositivos a un rastreador.



Cuando esté listo para construir, cree sus recursos y elija entre una variedad de estilos de mapas y proveedores de datos. A continuación, puede instalar el SDK que se adapte a su entorno de desarrollo y utilizar las API de ubicación de Amazon siguiendo las instrucciones de esta guía. Además, puede integrar la supervisión mediante Amazon CloudWatch y AWS CloudTrail.

Los temas de esta sección le proporcionan una visión general de los conceptos básicos de Amazon Location y lo preparan para empezar a trabajar con la ubicación en sus propias aplicaciones.

Temas

- [Información general de Amazon Location](#)
- [Mapas](#)
- [Búsqueda de lugares](#)
- [Rutas](#)
- [Geocercas y rastreadores](#)
- [Casos de uso habituales para el uso de Amazon Location Service](#)
- [¿Qué es un proveedor de datos?](#)
- [Regiones y puntos de conexión de Amazon](#)
- [Quotas de Amazon Location Service](#)

Información general de Amazon Location

Amazon Location Service proporciona acceso a la funcionalidad basada en la ubicación y a los proveedores de datos a través de recursos AWS . Amazon Location ofrece cinco tipos de AWS recursos, según el tipo de funcionalidad que necesites. Utilice los distintos recursos juntos para crear una aplicación completa basada en la ubicación. Puede crear uno o más de estos recursos mediante la consola de Amazon Location, las API de Amazon Location o los SDK.

Cada recurso define el [proveedor de datos](#) subyacente que se va a utilizar (cuando proceda) y proporciona acceso a las funciones relacionadas con su tipo.

Por ejemplo:

- [Mapas de Amazon Location Service](#) le permite elegir un mapa de un proveedor de mapas para usarlo en su aplicación móvil o web.
- [Amazon Location Service Places](#) le permite elegir un proveedor de datos para buscar puntos de interés, completar texto parcial, geocodificar y geocodificar de forma inversa.
- [Amazon Location Service Routes](#) te permite elegir un proveedor de datos, buscar rutas y estimar el tiempo de viaje en función de la información sobre las up-to-date carreteras y el tráfico en tiempo real.
- [Las geocercas de Amazon Location Service](#) le permiten definir áreas de interés como un límite virtual. A continuación, puede evaluar las ubicaciones comparándolas y recibir notificaciones de los eventos de entrada y salida.
- [Los rastreadores de Amazon Location Service](#) reciben actualizaciones de ubicación de tus dispositivos. Puede vincular los rastreadores a las colecciones de geocercas para que todas las actualizaciones de posición se evalúen automáticamente en función de sus geocercas.

Puede utilizar políticas IAM para administrar y autorizar el acceso a sus recursos de Amazon Location. También puede organizar sus recursos en grupos de recursos para gestionar y automatizar las tareas a medida que aumente el número de recursos. Para obtener más información sobre la administración de AWS recursos, consulte [¿Qué son los AWS Resource Groups?](#) En la Guía del usuario de AWS Resource Groups.

La ubicación se define mediante coordenadas de latitud y longitud que siguen el [Sistema Geodésico Mundial \(WGS 84\)](#), que se suele utilizar como sistema de referencia de coordenadas estándar para los servicios del Sistema de Posicionamiento Global (GPS).

En las siguientes secciones, se describe cómo funcionan los componentes de Amazon Location.

Mapas

El recurso de Mapas de Amazon Location Service le da acceso a los datos del mapa base subyacentes de un mapa. El recurso Mapas se utiliza con una biblioteca de representación de mapas para agregar un mapa interactivo a la aplicación. Puede agregar otras funciones al mapa, como marcadores (o marcadores), rutas y áreas poligonales, según sea necesario para su aplicación.

Note

Para obtener información sobre cómo utilizar los recursos de mapas en la práctica, consulte [Uso de Amazon Location Maps en su aplicación](#).

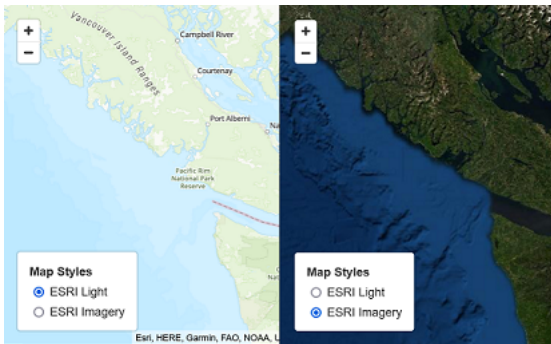
A continuación, se muestra información general sobre cómo crear y utilizar recursos de mapas:



1. Para crear un recurso de mapa en su AWS cuenta, debe seleccionar un estilo de mapa de un proveedor de datos.
2. A continuación, puede seleccionar e instalar el SDK que mejor se adapte a su entorno de desarrollo y a sus aplicaciones. Para obtener más información sobre las opciones disponibles, consulte el tema sobre el [acceso a Amazon Location](#).
3. Para mostrar un mapa en su aplicación, combine un recurso de mapa con una biblioteca de renderización, como Amplify o MapLibre Tangram. Para obtener más información, consulte [Uso de mapas](#) en esta guía.
4. A continuación, puede integrar la supervisión mediante servicios, como Amazon CloudWatch y AWS CloudTrail Amazon Location. Para obtener más información, consulte [Supervisión de Amazon Location Service con Amazon CloudWatch](#) y [Registro y monitoreo con AWS CloudTrail](#).

Estilos de mapa

Al crear un recurso de mapa, debe elegir un estilo de mapa para ese recurso. Los estilos de mapa definen el aspecto del mapa renderizado. Por ejemplo, la siguiente imagen muestra el mismo proveedor de datos con dos estilos diferentes de distintos recursos de mapas en Amazon Location. Un estilo es un estilo de carretera típico, basado en los datos vectoriales del mapa. El otro incluye datos ráster que muestran imágenes de satélite. El estilo puede cambiar al acercarse o alejarse del mapa, pero normalmente los estilos tienen un tema coherente. Es posible anular parte o toda la información de estilo antes de pasarla a la biblioteca de representación de mapas.



Opiniones políticas

Algunos estilos de mapas de Amazon Location Service admiten puntos de vista políticos adicionales.

Note

La vista política debe utilizarse de conformidad con las leyes aplicables, incluidas las leyes sobre la cartografía del país o la región en los que están disponibles los mapas, las imágenes y otros datos y el contenido de terceros a los que se accede a través de Amazon Location Service.

Los siguientes estilos de mapa admiten una vista política de la India (IND).

- [Estilos de mapa de Esri:](#)
 - Navegación Esri
 - Esri Light
 - Esri Street Map
 - Esri Dark Gray Canvas

- Esri Light Gray Canvas
- [Estilos de mapas de datos abiertos:](#)
 - Open Data Standard Light
 - Open Data Standard Dark
 - Open Data Visualization Light
 - Open Data Visualization Dark

En la consola de Amazon Location Service, puede filtrar los estilos que se muestran para mostrar solo los estilos que respaldan la vista política de la India.

Capas personalizadas

Una capa personalizada es una capa adicional que se puede habilitar para un estilo de mapa. Actualmente, solo el estilo de VectorEsriNavigation mapa admite la capa POI personalizada.

Al habilitar la capa POI personalizada, se agrega al mapa un conjunto más completo de lugares, como tiendas, servicios, restaurantes, atracciones y otros puntos de interés. De forma predeterminada, la capa personalizada es unset. Para obtener más información, consulta [MapConfiguration](#) la referencia de la API de ubicación.

Renderizado de mapas

Para renderizar un mapa en su aplicación, normalmente utilizará una biblioteca de renderizado de mapas. Existen varias opciones comunes que pueden utilizar las bibliotecas:

- MapLibre— MapLibre es una biblioteca de código abierto específica para renderizar mapas interactivos y es el método preferido para renderizar mapas de Amazon Location Service. MapLibre incluye la capacidad de renderizar datos ráster y vectoriales de una fuente de datos (como un recurso de mapas de Amazon Location). Puede ampliarlo MapLibre para dibujar sus propios datos en el mapa.
- Amplify: Amplify es un marco de código abierto para crear aplicaciones para la web, iOS, Android y más. Si su aplicación usa Amplify, puede ampliarla para incluir la funcionalidad de Amazon Location. Amplify incluye bibliotecas específicas para crear aplicaciones basadas en Amazon Location, incluido el renderizado de mapas. Amplify suele MapLibre renderizar el mapa, pero proporciona funciones adicionales específicas de Amazon Location Service para que su uso sea más eficiente, además de añadir funciones de búsqueda y de otro tipo.

- **Tangram:** Tangram es una biblioteca alternativa de código abierto que renderiza mapas interactivos, similar a. MapLibre

La biblioteca de renderizado de mapas extrae datos de Amazon Location Service en tiempo de ejecución y los procesa en función del recurso de mapa que seleccione. El recurso de mapa define el proveedor de datos y el estilo de mapa que se utilizarán.

La siguiente imagen muestra cómo se utiliza el recurso de mapa en Amazon Location Service junto con una biblioteca de renderizado de mapas para crear el mapa final.



1. Para crear un recurso de mapa en Amazon Location Service, utilice AWS Management Console o AWS CLI. Esto define el proveedor de datos y el estilo de mapa que desea utilizar.
2. La aplicación incluye una biblioteca de renderizado de mapas. Debe proporcionar a la biblioteca de representación de mapas el nombre del recurso de mapas que se va a utilizar. La biblioteca de representación de mapas extrae los datos y la información de estilo de ese recurso cartográfico de Amazon Location y renderiza el mapa en la pantalla.

Terminología de DNS

Recurso cartográfico

Le permite acceder a los datos del mapa de un proveedor seleccionado. Utilice el recurso de mapa para obtener mosaicos de mapa que contengan datos de mapa y un descriptor de estilo para especificar cómo se representan las características en un mapa.

Mapa base

Proporciona contexto geográfico a su mapa, que se almacena como capas de mosaicos vectoriales. Las capas de mosaicos incluyen contexto geográfico como nombres de calles, edificios y uso del suelo para referencia visual.

Vector

Los datos vectoriales son datos en forma de puntos, líneas y polígonos. Suele utilizarse para almacenar y mostrar carreteras, ubicaciones y áreas en un mapa. Una forma vectorial también se puede utilizar como iconos para los marcadores de un mapa.

Ráster

Los datos ráster son datos de imagen, compuestos por una cuadrícula, normalmente de colores. Suele utilizarse para almacenar y mostrar una representación de datos continuos en mapas, como mapas de terreno, imágenes de satélite o mapas térmicos. Las imágenes rasterizadas también se pueden utilizar como imágenes o iconos.

Estilos de mapa

Los datos vectoriales no incluyen de forma inherente información sobre cómo dibujar las capas de datos para crear el mapa final. Un estilo de mapa define el color y otra información de estilo de los datos para definir el aspecto que tendrán cuando se rendericen. Los recursos del mapa incluyen información de estilo para el mapa.

Amazon Location Service proporciona estilos que siguen las [especificaciones de estilo de Mapbox GL](#).

Mosaico vectorial

Formato de mosaico que almacena datos de mapas mediante formas vectoriales. Estos datos dan como resultado un mapa que se puede ajustar a la resolución de la pantalla y renderizar las entidades de forma selectiva de varias maneras, manteniendo un tamaño de archivo pequeño para un rendimiento óptimo.

Formato de archivo vectorial compatible: Mapbox Vector Tiles (MVT).

Archivo Glyph

Archivo binario que contiene caracteres Unicode codificados. Lo utiliza un renderizador de mapas para mostrar etiquetas.

Archivo Sprite

Archivo de imagen de gráfico de red portátil (PNG) que contiene imágenes ráster pequeñas, con descripciones de ubicación en un archivo JSON. Lo utiliza un renderizador de mapas para representar iconos o texturas en un mapa.

Búsqueda de lugares

Una función clave de Amazon Location Service es la capacidad de buscar la información de geolocalización. Amazon Location proporciona esta funcionalidad a través del recurso Place Index.

Note

Para obtener información acerca de cómo utilizar los recursos de índice de ubicación para realizar búsquedas en la práctica, consulte [Búsqueda de datos de ubicación y geolocalización mediante Amazon Location](#).

Puede usar las API de índice de sitios para buscar:

- Puntos de interés, como restaurantes y lugares de interés. Busque por nombre y ubicación opcional para buscar y reciba una lista de opciones ordenadas por relevancia.
- Una dirección física que recibe la latitud y la longitud de esa dirección. Esto se conoce como geocodificación.
- Posición de latitud y longitud que recibe la dirección postal asociada u otra información sobre la ubicación. Esto se conoce como geocodificación inversa.
- Una consulta de texto libre parcial o mal escrita, normalmente cuando un usuario la escribe. Esto se conoce como autocompletar, autosugerir o concordancia parcial.

El índice de ubicación incluye el proveedor de datos que se debe utilizar para la búsqueda.

Note

Los datos del mapa y otra información de geolocalización, incluidas las ubicaciones exactas, pueden variar de un proveedor de datos a otro. Como práctica recomendada, use el mismo proveedor de datos para el índice de ubicación, el mapa y otros recursos de Amazon

Location. Por ejemplo, si los lugares devueltos por su índice de ubicación no coinciden con la ubicación de los mismos lugares que proporciona su recurso cartográfico, puede colocar un marcador en lo que parezca ser un lugar incorrecto del mapa.

A continuación, se muestra cómo crear y utilizar recursos de índice de ubicación:



1. En primer lugar, debe crear un recurso de indexación de lugares en su AWS cuenta seleccionando un proveedor de datos.
2. A continuación, puede seleccionar e instalar el SDK que mejor se adapte a su entorno de desarrollo y a sus aplicaciones. Para obtener más información sobre las opciones disponibles, consulte el tema sobre el [acceso a Amazon Location](#).
3. Comience a usar las API de Amazon Location Places. Para obtener más información, consulte el tema sobre el uso de la [búsqueda de lugares](#).
4. A continuación, puede integrar la supervisión mediante servicios como Amazon CloudWatch y AWS CloudTrail. Para obtener más información, consulte [the section called “Monitorización con CloudWatch”](#) y [the section called “Uso CloudTrail con Amazon Location”](#).

Conceptos de geocodificación

Un índice de ubicación de Amazon Location proporciona una acción denominada [SearchPlaceIndexForText](#) que permite especificar el texto que se va a buscar. Por ejemplo, puede buscar:

- Lugares: una búsqueda de **Paris** podría arrojar la ubicación de la ciudad en Francia.
- Empresas: una búsqueda de **coffee shop** podría arrojar una lista de cafeterías, incluidos sus nombres y ubicaciones. También puede especificar una ubicación en la que buscar o un cuadro delimitador en el que buscar para que los resultados sean más relevantes. En este caso, si se proporciona una ubicación en el centro de Seattle (Washington), se mostrarían las cafeterías de esa zona.

- Direcciones: una búsqueda de **1600 Pennsylvania Ave, Washington D.C.** podría arrojar la ubicación de la Casa Blanca en los Estados Unidos (que se encuentra en esa dirección).

La búsqueda de texto de esta manera generalmente se denomina geocodificación, que implica buscar una ubicación geográfica para la dirección o el lugar.

Amazon Location Service también proporciona una acción de geocodificación inversa denominada [SearchPlaceIndexForPosition](#). Toma una ubicación geográfica y muestra la dirección, la empresa u otra información sobre lo que hay en esa ubicación.

Resultados de búsqueda

Cuando realiza una solicitud de búsqueda correcta en Amazon Location Service, se muestran uno o más resultados. Cada resultado incluye una etiqueta, que es el nombre o la descripción del resultado. Por ejemplo, una búsqueda de **coffee shop**, puede arrojar un resultado con la etiqueta **Hometown Cafe** que indique que se ha encontrado una cafetería llamada “Hometown Cafe”. El resultado de la búsqueda también suele incluir una dirección estructurada (con propiedades como el número de dirección, la unidad, la calle y el código postal). Según el proveedor de datos, también incluirá otros metadatos, como el país y la zona horaria.

Para realizar una búsqueda por nombre o categoría de una empresa (por ejemplo **coffee shop**), puede que desee mostrar todos los resultados obtenidos en un mapa. En el caso de una búsqueda de direcciones, es posible que desee utilizar el primer resultado automáticamente. Consulte el tema siguiente para obtener información sobre la relevancia.

Múltiples resultados y relevancia

Al buscar por texto, Amazon Location Service suele encontrar más de un resultado. Por ejemplo, una búsqueda de **Paris** puede mostrar la ciudad de Francia, pero también la ciudad de Texas. Los resultados se ordenan por relevancia, según lo determine el proveedor de datos.

Note

Todos los proveedores muestran los resultados en orden de relevancia. Si elige Esri o Grab como proveedor de datos, los resultados incluyen un valor de relevancia que puede usar para comprender la relevancia relativa entre los resultados de una sola solicitud.

Si especifica información adicional, como el nombre de un país o una ubicación en la que buscar, puede cambiar el orden de los resultados, reducir el número de resultados o incluso cambiar el conjunto de resultados mostrados. Por ejemplo, una búsqueda de **Paris** con una ubicación en Texas para buscar es más probable que muestre *Paris, Texas* como el primer resultado que *Paris, France*.

En una aplicación interactiva, puede usar la relevancia como ayuda para decidir si acepta el primer resultado o para pedirle al usuario que elimine la ambigüedad entre varios resultados devueltos. Si el primer resultado tiene una gran relevancia, puede que simplemente lo acepte como la respuesta correcta. Si hay varios resultados de alta relevancia o no hay ningún resultado de alta relevancia, puede que desee enumerar los resultados y dejar que el usuario seleccione el mejor resultado.

Resultados de direcciones

Puede buscar direcciones con Amazon Location Service con la misma acción

[SearchPlaceIndexForText](#). Cuanta más información proporcione, mayor será la probabilidad de que la dirección mostrada coincida con la indicada. Por ejemplo, es menos probable que **123 Main St** encuentre un resultado correcto que **123 Main St, Anytown, California, 90210**.

Las direcciones tienen varios atributos, como el número de calle, la calle, la ciudad, la región y el código postal, etc. Estos atributos se utilizan para buscar una dirección en el índice de sitios que coincida con tantos aspectos como sea posible. Cuantos más atributos se encuentren, más relevante se considerará la coincidencia y mayor será la probabilidad de que se muestre.

Note

La relevancia de los resultados de las direcciones se basa en el grado de coincidencia entre el resultado y la entrada. Puede ser el número de atributos que coinciden, pero también el grado de coincidencia de los resultados con la entrada. Por ejemplo, una entrada de **123 Main St** tendría mayor relevancia cuando *Main St* se encuentre en los datos que si fuera *Maine St* el único resultado. *Maine St* se seguirá mostrando, pero probablemente con un valor de relevancia inferior.

Los resultados de la búsqueda incluyen una etiqueta para la dirección completa (*123 Main St, Anytown, California, 90210*), pero también los atributos estructurados individuales de la dirección mostrada. Esto resulta útil porque se puede utilizar, por ejemplo, para rellenar los campos de direcciones de una base de datos o para examinar los resultados y buscar la ciudad, la región o el código postal de la ubicación encontrada.

Interpolación

Las direcciones incluidas en los datos del índice de sitios incluyen las coincidencias exactas de direcciones. Por ejemplo, supongamos que hay una calle 9th street y una manzana tiene 2 casas 220 y 240, como en la imagen siguiente.



El proveedor de datos crea los datos de geolocalización con esas dos direcciones conocidas. Puede buscar esas dos direcciones y las encontrará. Una vez que el proveedor de datos haya creado los datos del mapa, supongamos que se agrega una nueva casa entre las dos primeras direcciones. A esta nueva casa se le da la dirección 230. Si busca **230 S 9th St**, el proveedor de datos seguirá encontrando un resultado. En lugar de utilizar una dirección conocida, interpolará entre las direcciones ya conocidas y calculará la posición de la nueva dirección a partir de ellas. En este caso, podría suponer que 230 está a medio camino entre 220 y 240 (y en el mismo lado de la calle) y mostrar una ubicación aproximada en función de esa ubicación.

i Note

Los proveedores de datos actualizan periódicamente sus datos de geolocalización con nuevas direcciones. En este caso, **230 S 9th St** se agregaría a los datos del proveedor de datos, pero normalmente habrá un período en el que se cree una nueva dirección pero aún no se haya agregado a los datos.

En este caso, el proveedor de datos no puede determinar si la nueva dirección existe en el mundo, ya que aún no está en los datos, pero ofrece la mejor respuesta posible a partir de la información de la que dispone. Este resultado se denomina interpolado y el proveedor de datos puede mostrarlo en los resultados. Si `interpolated` es `false`, se trata de una dirección conocida. Si se muestra como `true`, es una dirección aproximada. Si no se muestra, significa que el proveedor de datos no proporcionó la información sobre si el resultado proviene de la interpolación.

Important

El proveedor de datos también puede mostrar resultados interpolados para direcciones que no existen en absoluto. Por ejemplo, en este caso, si introdujo **232 S 9th St**, el proveedor encontraría esta dirección inexistente y mostraría una ubicación cercana a 230, pero en el lado 240. Las direcciones interpoladas son útiles para llegar a la ubicación correcta, pero conviene tener en cuenta que no son direcciones conocidas.

Almacenar los resultados de la geocodificación

Al crear un recurso de índice de ubicación, debe especificar una opción de almacenamiento de datos (denominada `IntendedUse` en la API). Se puede configurar para que sean resultados de un solo uso o almacenados. Se trata de una pregunta sobre el uso que pretende dar a los resultados. Si va a almacenar los resultados (incluso con fines de almacenamiento en caché), debe elegir la opción de almacenamiento, no la de un solo uso.

Note

Si elige la opción almacenada (etiquetada como `Yes`, los resultados se almacenarán en la consola, o elegir `storage` en la API `CreatePlaceIndex`), Amazon Location Service no almacenará los resultados por usted. Esto indica que planea almacenar los resultados.

Cuando piense en cómo va a utilizar los resultados de sus consultas en Amazon Location Service, siempre debe tener en cuenta las [condiciones del servicio de AWS](#) aplicables.

Terminología de lugares

Recurso de índice de ubicación

Le permite elegir un origen de datos para admitir las consultas de búsqueda. Por ejemplo, puede buscar puntos de interés, direcciones o coordenadas. Cuando se envía una consulta de búsqueda a un recurso de índice de ubicación, se completa con el origen de datos configurada del recurso.

Geocodificación

La geocodificación es el proceso de tomar una entrada de texto, buscarla en el índice de ubicación y mostrar los resultados con la posición.

Geocodificación inversa

La geocodificación inversa es el proceso de tomar una posición y mostrar información sobre esa posición desde el índice de ubicación, como la dirección, la ciudad o el negocio de esa ubicación.

Relevancia

La relevancia es el grado de coincidencia entre un resultado y la entrada. No es una medida de la exactitud.

Interpolación

La interpolación es el proceso de encontrar direcciones desconocidas utilizando ubicaciones de direcciones conocidas como puntos de guía.

Códigos de país ISO 3166

Amazon Location Service Places utiliza [los códigos de país 3166 de la Organización Internacional de Normalización \(ISO\)](#) para referirse a países o regiones.

Para encontrar el código de un país o región específicos, utilice la plataforma de [navegación en línea de la ISO](#).

Rutas

En esta sección se brinda información general sobre los conceptos relacionados con las rutas mediante Amazon Location Service.

Note

Para obtener información sobre cómo utilizar los recursos de ruta en la práctica, consulte [Calcular rutas utilizando Amazon Location Service](#).

Recursos para la calculadora de rutas

Los recursos de la calculadora de rutas le permiten encontrar rutas y estimar el tiempo de viaje en función de la red de up-to-date carreteras y la información de tráfico en tiempo real del proveedor de datos que elija.

Puede usar las API de Routes para crear funciones que permitan a su aplicación solicitar el tiempo de viaje, la distancia y la geometría de la ruta entre dos ubicaciones. También puede usar la API de Routes para solicitar el tiempo y la distancia de viaje entre un conjunto de salidas y destinos en una sola solicitud para calcular una matriz.

A continuación, se muestra cómo crear y utilizar el recurso de calculadora de ruta:



1. En primer lugar, debe crear un recurso de calculadora de rutas en su AWS cuenta seleccionando un proveedor de datos.
2. A continuación, puede seleccionar e instalar el SDK que mejor se adapte a su entorno de desarrollo y a sus aplicaciones.
3. Comience a usar las API de Amazon Location Routes. Para obtener más información acerca de cómo utilizar las API de enrutamiento, consulte [Calcular rutas utilizando Amazon Location Service](#).

4. A continuación, puede integrar la supervisión mediante servicios como Amazon CloudWatch y AWS CloudTrail. Para obtener más información, consulte [Supervisión de Amazon Location Service con Amazon CloudWatch](#) y [Registro y monitoreo con AWS CloudTrail](#).

Cálculo de rutas

Un recurso de calculadora de rutas de Amazon Location proporciona una acción denominada `CalculateRoute` que puede utilizar para crear una ruta entre dos ubicaciones geográficas (la de salida y la de destino). La ruta calculada incluye la geometría para dibujar la ruta en un mapa, además del tiempo y la distancia totales de la ruta.

Uso de puntos de referencia

Al crear la solicitud de ruta, puede agregar puntos de referencia adicionales a la ruta. Se trata de puntos entre la salida y el destino que actúan como paradas a lo largo de la ruta. La ruta se calculará a través de cada uno de los puntos de referencia especificados. La ruta desde un punto de la solicitud hasta el siguiente se denomina Leg. Cada tramo incluye una distancia, un tiempo y la geometría de esa parte de la ruta.

Note

Los puntos de referencia se enrutan en el orden indicado en la solicitud. No se reordenan según la ruta más corta. Consulte la sección [Planificación de rutas](#) para obtener información sobre cómo encontrar la ruta más corta.

Puede incluir hasta 25 puntos de referencia en una sola solicitud para calcular una ruta.

Tráfico y hora de salida

Amazon Location Service tiene en cuenta el tráfico al calcular una ruta. El tráfico que considera se basa en la hora que especifique. Puede especificar que salga ahora o puede proporcionar una hora específica a la que desee salir, lo que afectará al resultado de la ruta al ajustarse al tráfico a la hora especificada.

Note

Puede calcular la hora de llegada utilizando la hora de salida y el tiempo de respuesta de la ruta, por ejemplo, para estimar la llegada de un conductor.

Si quiere que Amazon Location no tenga en cuenta el tráfico, no especifique la hora de salida ni la hora de salida ahora. Esto calculará una ruta que asume las mejores condiciones de tráfico para la ruta.

Opciones de modo de viaje

Puede configurar el modo de viaje al calcular una ruta con Amazon Location Service. El modo de viaje predeterminado es automóvil, pero también puede seleccionar camión o caminar.

Si especificas el modo automóvil o camión, también puede especificar opciones adicionales.

En el modo automóvil, puede especificar que desea evitar las autopistas de peaje o los transbordadores. Esto evitará los transbordadores y las autopistas de peaje, pero seguirá recorriéndolas si es la única forma de llegar al destino.

En el modo camión, también puede evitar los transbordadores y las autopistas de peaje, pero también puede especificar el tamaño y el peso del camión para evitar rutas en las que no pueda entrar el camión.

Planificación de rutas

Puede utilizar Amazon Location Service para crear entradas para su software de planificación y optimización de rutas. Puede crear resultados de ruta, incluidos el tiempo y la distancia de viaje, para las rutas entre un conjunto de posiciones de salida y un conjunto de posiciones de destino. Esto se denomina crear una matriz de rutas.

Note

Hay muchos escenarios diferentes que el software de planificación y optimización de rutas puede resolver. Por ejemplo, el software de planificación puede usar el conjunto de tiempos y distancias entre puntos para calcular la ruta más corta que se detiene en cada punto, lo que proporciona una ruta eficiente para un solo conductor. Como alternativa, el software de planificación se puede utilizar para dividir las paradas entre varios camiones, lo que mejora la eficiencia en toda la flota o para garantizar que cada cliente reciba la visita en el plazo que necesite. Amazon Location proporciona las funciones de enrutamiento de una manera eficiente para permitir que el software de planificación complete su tarea.

Por ejemplo, dadas las posiciones de salida A y B y las posiciones de destino X e Y, Amazon Location Service mostrará el tiempo y la distancia de viaje para las rutas de A a X, A a Y, B a X y B a Y.

Al igual que al calcular una sola ruta, puede calcular las rutas con diferentes modos de transporte, prevenciones y condiciones del tráfico. Por ejemplo, puede especificar que el vehículo es un camión de 35 metros de largo y la ruta calculada utilizará esas restricciones para determinar el tiempo y la distancia de viaje. No puede incluir puntos de referencia en un cálculo matricial de rutas.

El número de resultados mostrados (y de rutas calculadas) es el número de posiciones de salida multiplicado por el número de posiciones de destino. Se le cobrará por cada ruta calculada, no por cada solicitud al servicio, por lo que una matriz de rutas con 10 salidas y 10 destinos se facturará como 100 rutas.

Terminología de rutas

Recursos para la calculadora de rutas

Un AWS recurso que le permite estimar el tiempo y la distancia de viaje y trazar las rutas en un mapa con datos de tráfico y red de carreteras procedentes del proveedor de datos que haya elegido.

Con los recursos de la calculadora de rutas, puede calcular las rutas para diferentes modos de transporte, desvíos y condiciones del tráfico.

Ruta

Una ruta contiene los detalles que se utilizan cuando se recorre una ruta desde la posición de partida, las posiciones de los puntos de referencia y la posición de destino.

Algunos ejemplos de detalles de una ruta son:

- La distancia de una posición a otra.
- El tiempo que se tarda en viajar de una posición a la siguiente.
- La LineString geometría que representa la trayectoria de la ruta.

Para obtener más información sobre las rutas, consulta la [sintaxis de respuesta de la CalculateRoute operación](#) en la referencia de la API de rutas de Amazon Location Service.

Matriz de rutas

Una lista de rutas, desde un conjunto de posiciones de salida hasta un conjunto de posiciones de destino. Útil como entrada en el software de planificación u optimización de rutas.

Para obtener más información sobre el cálculo de una matriz de rutas, consulta la [sintaxis de la CalculateRouteMatrix operación](#) en la referencia de la API de rutas de Amazon Location Service.

Geometría **LineString**

Una ruta de Amazon Location consta de uno o más tramos (una ruta que va de un punto intermedio a otro dentro de la ruta general). La geometría de cada tramo es una polilínea representada como **LineString**. Una **LineString** es una matriz ordenada de posiciones que se puede usar para trazar una ruta en un mapa.

El siguiente es un ejemplo de **LineString** con tres puntos:

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

Puntos de referencia

Los puntos de referencia son posiciones intermedias que actúan como paradas a lo largo de una ruta entre la posición de salida y la posición de destino. El orden de escala de la ruta sigue el orden en el que se indican las posiciones de los puntos de referencia en la solicitud.

Tramo

Un solo tramo es el viaje de una posición a otra. Si las posiciones no están ubicadas en una carretera, se mueven a la carretera más cercana. El número de tramos de una ruta es uno menos que el número total de posiciones.

Una ruta sin puntos de referencia consta de un solo trayecto, desde la posición de salida hasta el destino. Una ruta con 1 punto de referencia consta de 2 tramos, desde la posición de salida hasta el punto de referencia y, después, desde el punto de referencia hasta el destino.

Paso

Un paso es una subsección de un tramo. Cada paso proporciona información resumida sobre ese paso del tramo.

Geocercas y rastreadores

En esta sección se proporciona una descripción general de los conceptos relacionados con el uso de geocercas y rastreadores de Amazon Location Service. Las geocercas son límites de polígonos que puede utilizar para recibir notificaciones cuando los dispositivos o las posiciones entren y salgan de las áreas. Los recursos del rastreador se utilizan para almacenar y actualizar las posiciones de los dispositivos a medida que se mueven.

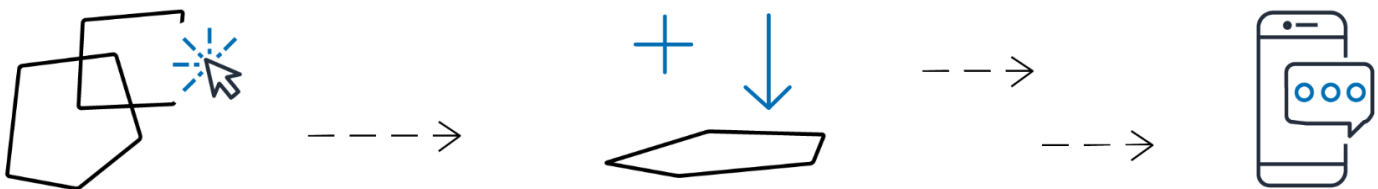
Note

Para obtener información sobre cómo utilizar geocercas y rastreadores en la práctica, consulte [Geocercar un área de interés mediante Amazon Location](#).

Geocercas

Los recursos de la colección de geocercas permiten almacenar y gestionar las geocercas, es decir, límites virtuales en un mapa. Puede evaluar las ubicaciones comparándolas con un recurso de recopilación de geocercas y recibir notificaciones cuando la actualización de la ubicación cruce el límite de cualquiera de las geocercas de la colección de geocercas.

A continuación, se muestra cómo crear y utilizar recursos de colección de geocercas:



1. Cree un recurso de recopilación de geovallas en su cuenta. AWS
2. Agregue geocercas a esa colección. Puede hacerlo mediante la herramienta de carga de geocercas de la consola de Amazon Location o mediante la API de geocercas de Amazon Location. Para obtener más información sobre las opciones disponibles, consulte [Cómo acceder a Amazon Location](#).

Las geocercas se pueden definir mediante un polígono o un círculo. Use un polígono para saber cuándo un dispositivo entra en un área específica. Use un círculo para saber cuándo un dispositivo se encuentra a una distancia (radio) determinada de un punto.

3. Puede empezar a evaluar las ubicaciones comparándolas con todas tus geocercas. Cuando una actualización de ubicación cruza los límites de una o más geovallas, el recurso de recopilación de geovallas emite uno de los siguientes tipos de eventos de geovallas en Amazon: EventBridge
 - ENTRADA: se genera un evento para cada geocerca en el que la actualización de ubicación cruza su límite al ingresar a este.
 - SALIDA: se genera un evento para cada geocerca en el que la actualización de ubicación cruza su límite al salir de este.

Para obtener más información, consulte [the section called “Reaccionar ante los eventos con EventBridge”](#). También puede integrar la supervisión mediante servicios como Amazon CloudWatch y AWS CloudTrail. Para obtener más información, consulte [the section called “Monitorización con CloudWatch”](#) y [the section called “Uso CloudTrail con Amazon Location”](#).

Por ejemplo, si está rastreando una flota de camiones y desea recibir una notificación cuando un camión entre en un área determinada de cualquiera de sus almacenes. Puede crear una cerca geográfica para el área alrededor de cada almacén. Luego, cuando los camiones le envíen ubicaciones actualizadas, puede usar Amazon Location Service para evaluar esas posiciones y comprobar si un camión ha entrado (o salido) de una de las áreas de la geocerca.

Note

Se le facturará en función del número de colecciones de geocercas con las que realice la evaluación. La cantidad de geocercas de cada colección no afecta a su factura. Dado que cada colección de geocercas puede contener hasta 50 000 geocercas, tal vez desee combinar sus geocercas en un menor número de colecciones, siempre que sea posible, para reducir el costo de las evaluaciones de las geocercas. Los eventos generados incluirán el identificador de la geocerca individual de la colección, así como el identificador de la colección.

Eventos de geocercas

Las ubicaciones de las posiciones que está monitoreando están referenciadas mediante un identificador llamado DeviceId (y las posiciones se denominan posiciones de dispositivos). Puede enviar una lista de las posiciones de los dispositivos para evaluarlas directamente al recurso de recolección de geocercas o puede usar un rastreador. Consulte la siguiente sección para obtener más información sobre los rastreadores.

Solo recibes eventos (a través de Amazon EventBridge) cuando un dispositivo entra o sale de una geocerca, no por cada cambio de posición. Esto significa que normalmente recibirá eventos y tendrá que responder a ellos con mucha menos frecuencia que en todas las actualizaciones de posición del dispositivo.

Note

Para la primera evaluación de la ubicación de un DeviceID específico, se supone que el dispositivo no estaba anteriormente en ninguna geocerca. Por lo tanto, la primera actualización generará un evento ENTER, si está dentro de una geocerca de la colección, y ningún evento si no lo está.

Para calcular si un dispositivo ha entrado o salido de una geocerca, Amazon Location Service debe mantener el estado de posición anterior del dispositivo. Este estado de posición se almacena durante 30 días. Después de 30 días sin una actualización para un dispositivo, una nueva actualización de ubicación se considerará la primera actualización de posición.

Rastreadores

Un rastreador almacena las actualizaciones de posición de un conjunto de dispositivos. El rastreador se puede utilizar para consultar la ubicación actual o el historial de ubicaciones de los dispositivos. Almacena las actualizaciones, pero reduce el espacio de almacenamiento y el ruido visual al filtrar las ubicaciones antes de almacenarlas.

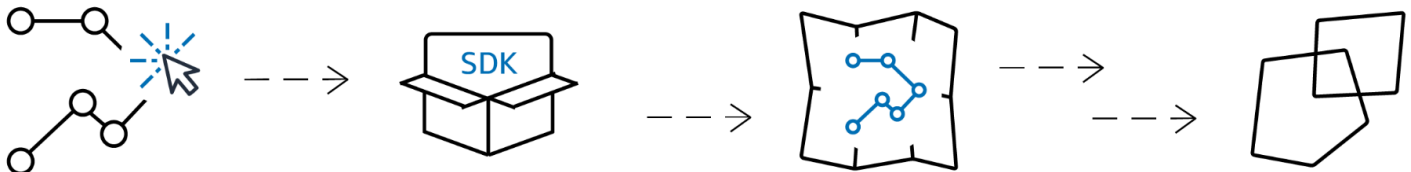
Cada actualización de posición almacenada en los recursos del rastreador puede incluir una medida de la precisión de la posición y hasta tres campos de metadatos sobre la posición o el dispositivo que desee almacenar. Los metadatos se almacenan como pares clave-valor y pueden almacenar información como la velocidad, la dirección, la presión de los neumáticos o la temperatura del motor.

Note

El almacenamiento del rastreador se cifra automáticamente con las claves AWS propias. Puede agregar otra capa de cifrado mediante las claves de KMS que administre, para garantizar que solo usted pueda acceder a sus datos. Para obtener más información, consulte [Cifrado de datos en reposo para Amazon Location Service](#).

El filtrado y el almacenamiento de la posición de los rastreadores son útiles por sí solos, pero los rastreadores son especialmente útiles cuando se combinan con geocercas. Puede vincular los rastreadores a uno o varios de sus recursos de recolección de geocercas, y las actualizaciones de posición se evalúan automáticamente comparándolas con las geocercas de esas colecciones. El uso adecuado del filtrado también puede reducir considerablemente los costos de las evaluaciones de las geocercas.

El siguiente diagrama muestra cómo crear y utilizar recursos de rastreadores:



1. En primer lugar, debe crear un recurso de seguimiento en su AWS cuenta.
2. A continuación, decida cómo vas a enviar las actualizaciones de ubicación a los recursos de su rastreador. Use los [SDK de AWS](#) para integrar las capacidades de rastreo en sus aplicaciones móviles. Como alternativa, puede utilizar MQTT siguiendo las step-by-step instrucciones del [seguimiento mediante MQTT](#).
3. Ahora puede usar su recurso de rastreo para registrar el historial de ubicaciones y visualizarlo en un mapa.
4. También puede vincular su recurso de rastreo a una o más colecciones de geocercas para que cada actualización de posición que se envíe a su recurso de rastreo se evalúe automáticamente comparándola con todas las geocercas de todas las colecciones de geocercas vinculadas. Puede vincular el recurso en la página de detalles de recursos del rastreador de la consola de ubicaciones de Amazon o mediante la API de Amazon Location Trackers.
5. A continuación, puede integrar la supervisión mediante servicios como Amazon CloudWatch y AWS CloudTrail. Para obtener más información, consulte [the section called “Monitorización con CloudWatch”](#) y [the section called “Uso CloudTrail con Amazon Location”](#).

Uso de rastreadores con geocercas

Los rastreadores proporcionan una funcionalidad adicional cuando se combinan con geocercas. Para evaluar automáticamente las ubicaciones de los rastreadores, asocie un rastreador a una colección de geocercas, ya sea a través de la consola de Amazon Location o de la API. Cada vez que el rastreador reciba una ubicación actualizada, esa ubicación se evaluará comparándola

con cada geocerca de la colección y se generarán EXIT los eventos correspondientes ENTER en Amazon. EventBridge También puede aplicar filtros al rastreador y, en función del filtrado, reducir los costos de las evaluaciones de la geocerca evaluando únicamente las actualizaciones de ubicación significativas.

Si asocia el rastreador a un conjunto de geocercas después de que ya haya recibido algunas actualizaciones de posición, la primera actualización de posición tras la asociación se considerará una actualización inicial para las evaluaciones de geocercas. Si se encuentra dentro de una geocerca, recibirá un evento ENTER. Si no está dentro de ninguna geocerca, no recibirá ningún evento EXIT, independientemente del estado anterior.

Filtrado de posición

Los rastreadores pueden filtrar automáticamente las posiciones que se les envían. Existen varios motivos por los que es posible que desee filtrar algunas de las actualizaciones de ubicación de su dispositivo. Si tiene un sistema que solo envía informes cada minuto más o menos, quizás quiera filtrar los dispositivos por tiempo, almacenando y evaluando las posiciones solo cada 30 segundos. Incluso si monitoriza con más frecuencia, es posible que desee filtrar las actualizaciones de posición para eliminar el ruido del hardware del GPS. Las ubicaciones de posición del GPS son intrínsecamente ruidosas. Su precisión no es perfecta al 100 %, por lo que incluso un dispositivo estacionario parece moverse ligeramente. A bajas velocidades, esta fluctuación provoca confusión visual y puede provocar entradas y salidas falsas si el dispositivo se encuentra cerca del borde de una geocerca.

El filtrado de posición funciona a medida que un rastreador recibe las actualizaciones de posición, lo que reduce el ruido visual en las rutas del dispositivo (fluctuación), reduce el número de eventos de entrada y salida falsos de la geocerca y ayuda a gestionar los costos al reducir el número de actualizaciones de posición almacenadas y de las evaluaciones de geocerca activadas.

Los rastreadores ofrecen tres opciones de filtrado de posición para ayudar a gestionar los costos y reducir las interrupciones en las actualizaciones de ubicación.

- Basado en la precisión: utilícelo con cualquier dispositivo que proporcione una medición de la precisión. La mayoría de los dispositivos GPS y móviles proporcionan esta información. La precisión de cada medición de posición se ve afectada por muchos factores ambientales, como la recepción de los satélites GPS, el paisaje y la proximidad de los dispositivos wifi y bluetooth. La mayoría de los dispositivos, incluidos la mayoría de los dispositivos móviles, pueden proporcionar una estimación de la precisión de la medición junto con la medición. Mediante el filtrado AccuracyBased, Amazon Location ignora las actualizaciones de ubicación si el dispositivo

se ha movido menos de la precisión medida. Por ejemplo, si dos actualizaciones consecutivas de un dispositivo tienen un rango de precisión de 5 m y 10 m, Amazon Location ignora la segunda actualización si el dispositivo se ha movido menos de 15 m. Amazon Location no evalúa las actualizaciones ignoradas comparándolas con las geocercas ni las almacena.

Si no se proporciona precisión, se trata como cero y la medición se considera perfectamente precisa, y no se aplicará ningún filtro a las actualizaciones.

Note

Puede utilizar un filtrado basado en la precisión para eliminar todos los filtros. Si selecciona un filtrado basado en la precisión, pero sobrescribe todos los datos de precisión a cero u omite la precisión por completo, Amazon Location no filtrará ninguna actualización.

En la mayoría de los casos, el filtrado basado en la precisión es una buena opción para filtrar las actualizaciones de posición, ya que proporciona un equilibrio entre la ubicación de seguimiento y, al mismo tiempo, filtra las actualizaciones innecesarias, lo que reduce los costos.

- **Basado en la distancia:** utilícelo cuando sus dispositivos no proporcionen una medición precisa, pero aun así desee aprovechar las ventajas del filtrado para reducir las fluctuaciones y gestionar los costos. El filtrado `DistanceBased` ignora las actualizaciones de ubicación en las que los dispositivos se han movido menos de 30 m (98,4 pies). Cuando utiliza el filtrado de posición `DistanceBased`, Amazon Location no evalúa estas actualizaciones ignoradas comparándolas con las geocercas ni las almacena.

La precisión de la mayoría de los dispositivos móviles, incluida la precisión media de los dispositivos iOS y Android, está dentro de los 15 m. En la mayoría de las aplicaciones, el filtrado `DistanceBased` puede reducir el efecto de las imprecisiones de ubicación al mostrar la trayectoria del dispositivo en un mapa, y el efecto rebote de varios eventos consecutivos de entrada y salida cuando los dispositivos se encuentran cerca del borde de una geocerca. También puede ayudar a reducir el coste de la aplicación, ya que realiza menos llamadas para evaluar las geocercas asociadas o recuperar la posición de los dispositivos.

El filtrado basado en la distancia es útil si quiere filtrar, pero su dispositivo no proporciona mediciones de precisión o si desea filtrar un mayor número de actualizaciones que con el filtrado basado en la precisión.

- **Basado en el tiempo:** (predeterminado) úselo cuando sus dispositivos envíen actualizaciones de posición con mucha frecuencia (más de una vez cada 30 segundos) y desee realizar evaluaciones de la geocerca prácticamente en tiempo real sin almacenar todas las actualizaciones. Durante el filtrado TimeBased, las actualizaciones de ubicación se evalúan con las recopilaciones de geocercas asociadas, pero no se almacenan todas las actualizaciones de ubicación. Si la frecuencia de actualización es superior a 30 segundos, solo se almacenará una actualización cada 30 segundos por cada ID de dispositivo único.

El filtrado basado en el tiempo resulta especialmente útil si desea almacenar menos posiciones, pero desea que cada actualización de posición se evalúe comparándola con los conjuntos de geocercas asociados.

Note

Tenga en cuenta los costos de su aplicación de rastreo a la hora de decidir el método de filtrado y la frecuencia de las actualizaciones de posición. Se le facturará por cada actualización de ubicación y una vez por la evaluación de la actualización de posición con respecto a cada colección de geocercas vinculadas. Por ejemplo, si utiliza un filtrado basado en el tiempo, si su rastreador está vinculado a dos colecciones de geocercas, cada actualización de posición contará como una solicitud de actualización de ubicación y dos evaluaciones de la colección de geocercas. Si informa de las actualizaciones de posición de sus dispositivos cada 5 segundos y utiliza un filtrado basado en el tiempo, se le facturarán 720 actualizaciones de ubicación y 1440 evaluaciones de geocercas por hora para cada dispositivo.

Terminología de geocerca

Recolección de geocerca

Contiene cero o más geocercas. Es capaz de monitorear las geocercas emitiendo eventos de entrada y salida, cuando se solicitan, para evaluar la posición de un dispositivo con respecto a sus geocercas.

Geocercas

Geometría poligonal o circular que define un límite virtual en un mapa.

Geometría poligonal

Una geocerca de Amazon Location es un límite virtual de un área geográfica y se representa como una geometría poligonal o como un círculo.

Un círculo es un punto rodeado de una distancia. Usa un círculo cuando desee recibir una notificación si un dispositivo se encuentra a una distancia determinada de una ubicación.

Un polígono es una matriz compuesta por uno o más anillos lineales. Utilice un polígono cuando desee definir un límite específico para las notificaciones del dispositivo. Un anillo lineal es una matriz de cuatro o más vértices, donde el primer y el último vértice son iguales para formar un límite cerrado. Cada vértice es un punto bidimensional de la forma `[longitud, latitud]`, donde las unidades de longitud y latitud son grados. Los vértices se deben enumerar en el sentido contrario a las agujas del reloj alrededor del polígono.

Note

Amazon Location Service no admite polígonos con más de un anillo. Esto incluye huecos, islas o multipolígonos. Amazon Location tampoco admite polígonos enrollados en el sentido de las agujas del reloj o que crucen el antimeridiano.

El siguiente es un ejemplo de un anillo externo lineal único:

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

Terminología de rastreadores

Recurso de seguimiento

AWS Recurso que recibe actualizaciones de ubicación de los dispositivos. El recurso de seguimiento proporciona soporte para consultas de ubicación, como la ubicación actual e histórica del dispositivo. Al vincular un recurso de rastreo a una colección de geocercas, se

evalúan automáticamente las actualizaciones de ubicación comparándolas con todas las geocercas de la colección de geocercas vinculada.

Datos de posición rastreados

Un recurso de seguimiento almacena información sobre sus dispositivos a lo largo del tiempo. La información incluye una serie de actualizaciones de posición, en las que cada actualización incluye la ubicación, la hora y los metadatos opcionales. Los metadatos pueden incluir la precisión de una posición y hasta tres pares de valores clave para ayudarte a rastrear la información clave sobre cada posición, como la velocidad, la dirección, la presión de los neumáticos, el combustible restante o la temperatura del motor del vehículo que estás rastreando. Los rastreadores mantienen el historial de ubicación del dispositivo durante 30 días.

Filtrado de posición

El filtrado de posiciones puede ayudarle a controlar los costos y mejorar la calidad de su aplicación de rastreo, ya que filtra las actualizaciones de posición que no proporcionan información valiosa antes de almacenarlas o evaluarlas comparándolas con las geocercas.

Puede elegir entre los filtros AccuracyBased, DistanceBased o TimeBased. De forma predeterminada, el filtrado de posición está establecido en TimeBased.

Puede configurar el filtrado de posición al crear o actualizar los recursos del rastreador.

Formato de marca de tiempo RFC 3339

Amazon Location Service Trackers utiliza el formato [RFC 3339](#), que sigue el formato [8601 de la Organización Internacional de Normalización \(ISO\) para](#) las fechas y las horas.

El formato es “YYYY-MM-DDThh:mm:ss.sssZ+00:00”:

- YYYY-MM-DD: representa el formato de fecha.
- T: indica que seguirán los valores de hora.
- hh:mm:ss.sss: representa la hora en formato de 24 horas.
- Z: indica que la zona horaria utilizada es UTC, que puede ir seguida de desviaciones con respecto a la zona horaria UTC.
- +00:00: indica de forma opcional las desviaciones de la zona horaria UTC. Por ejemplo, +01:00 indica UTC + 1 hora.

Ejemplo

Para el 2 de julio de 2020, a las 12:15:20 de la tarde, con un ajuste de 1 hora adicional a la zona horaria UTC.

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

Casos de uso habituales para el uso de Amazon Location Service

Amazon Location Service le permite crear una gama de aplicaciones, desde el seguimiento de activos hasta el marketing basado en la ubicación. A continuación se indican algunos casos de uso comunes:

Participación de los usuarios y geomarketing

Utilice los datos de ubicación para crear soluciones que mejoren la participación de los usuarios con el marketing dirigido a los clientes objetivo. Por ejemplo, Amazon Location puede activar un evento que genere una notificación cuando un cliente que pidió un café en su aplicación móvil esté cerca. Además, puede crear funciones de segmentación geográfica para que los minoristas puedan enviar códigos de descuento o folletos digitales a los clientes que se encuentren cerca de las tiendas objetivo.

Seguimiento de activos

Cree funciones de seguimiento de activos para ayudar a las empresas a comprender las ubicaciones actuales e históricas de sus productos, personal e infraestructura. Con las funciones de seguimiento de activos, puede crear una serie de soluciones que optimicen el personal remoto, protejan los envíos en ruta y maximicen la eficacia de los envíos.

Delivery

Integre las funciones de ubicación en las aplicaciones de entrega para almacenar, rastrear y coordinar el lugar de salida, los vehículos de entrega y su destino. Por ejemplo, una aplicación de entrega de comida con las funciones de ubicación de Amazon integradas tiene funciones de rastreo de ubicación y geolocalización que pueden notificar automáticamente a un restaurante cuando hay un repartidor cerca. Esto reduce el tiempo de espera y ayuda a mantener la calidad de la comida entregada.

En este tema, se proporciona una descripción general de la arquitectura y los pasos para las aplicaciones que puede crear con Amazon Location.

Temas

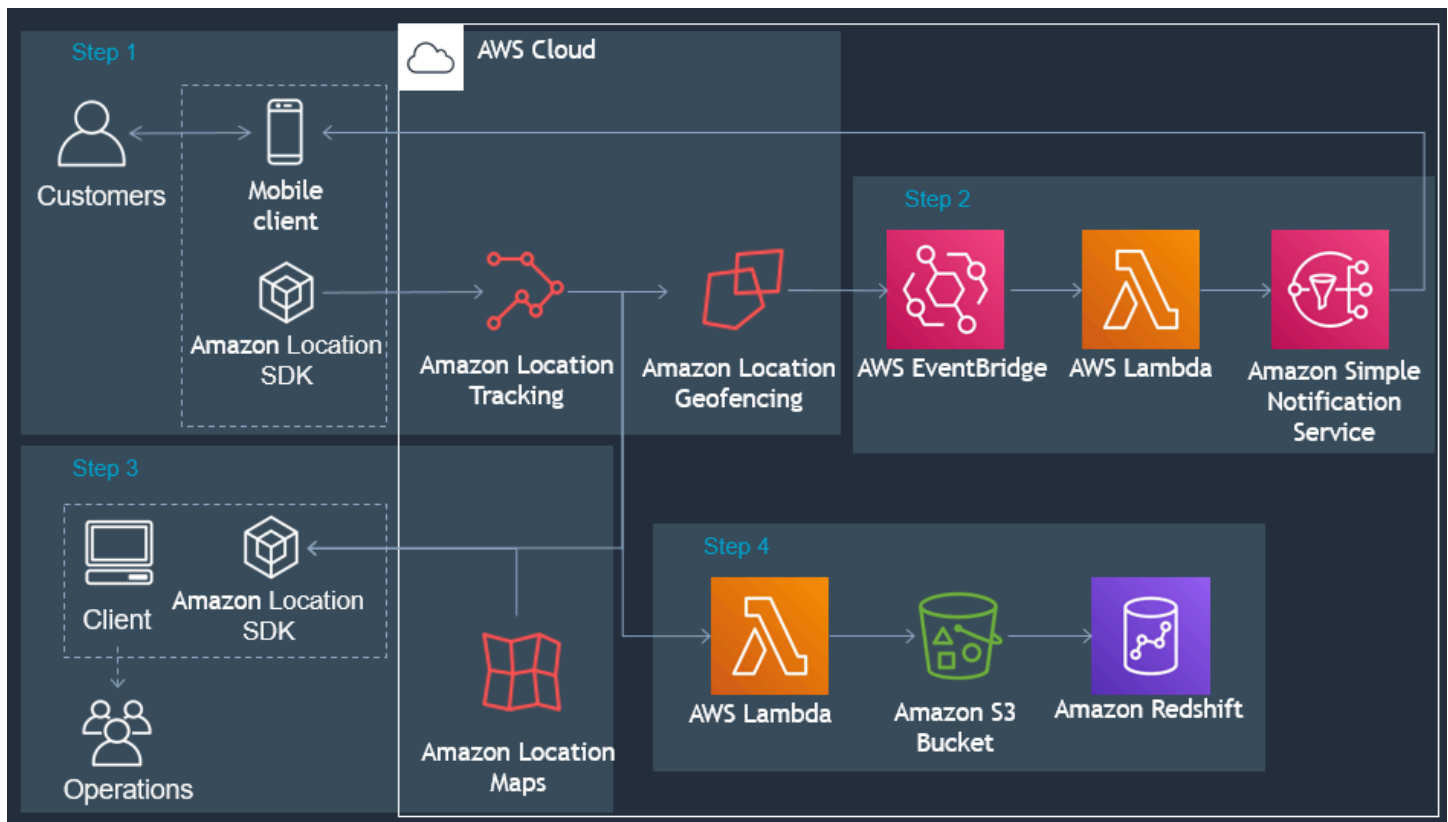
- [Participación de los usuarios y aplicaciones de geomarketing](#)
- [Aplicaciones de seguimiento de activos](#)
- [Aplicaciones de entrega](#)

Participación de los usuarios y aplicaciones de geomarketing

La siguiente es una ilustración de la arquitectura de una aplicación de geomarketing y participación de los usuarios que utiliza Amazon Location:

Con esta arquitectura, puede:

- Iniciar eventos en función de la proximidad de un objetivo para poder enviar ofertas a los clientes cercanos o atraer a quienes abandonaron su establecimiento recientemente (lo que se denomina geotargeting).
- Visualice las ubicaciones de los dispositivos de los clientes en un mapa para monitorear las tendencias a lo largo del tiempo.
- Almacene las ubicaciones de los dispositivos de los clientes para que pueda analizarlas a lo largo del tiempo.
- Analizar el historial de ubicaciones para identificar tendencias y oportunidades de optimización.



La siguiente es una descripción general de los pasos necesarios para crear una aplicación de geomarketing y participación de los usuarios:

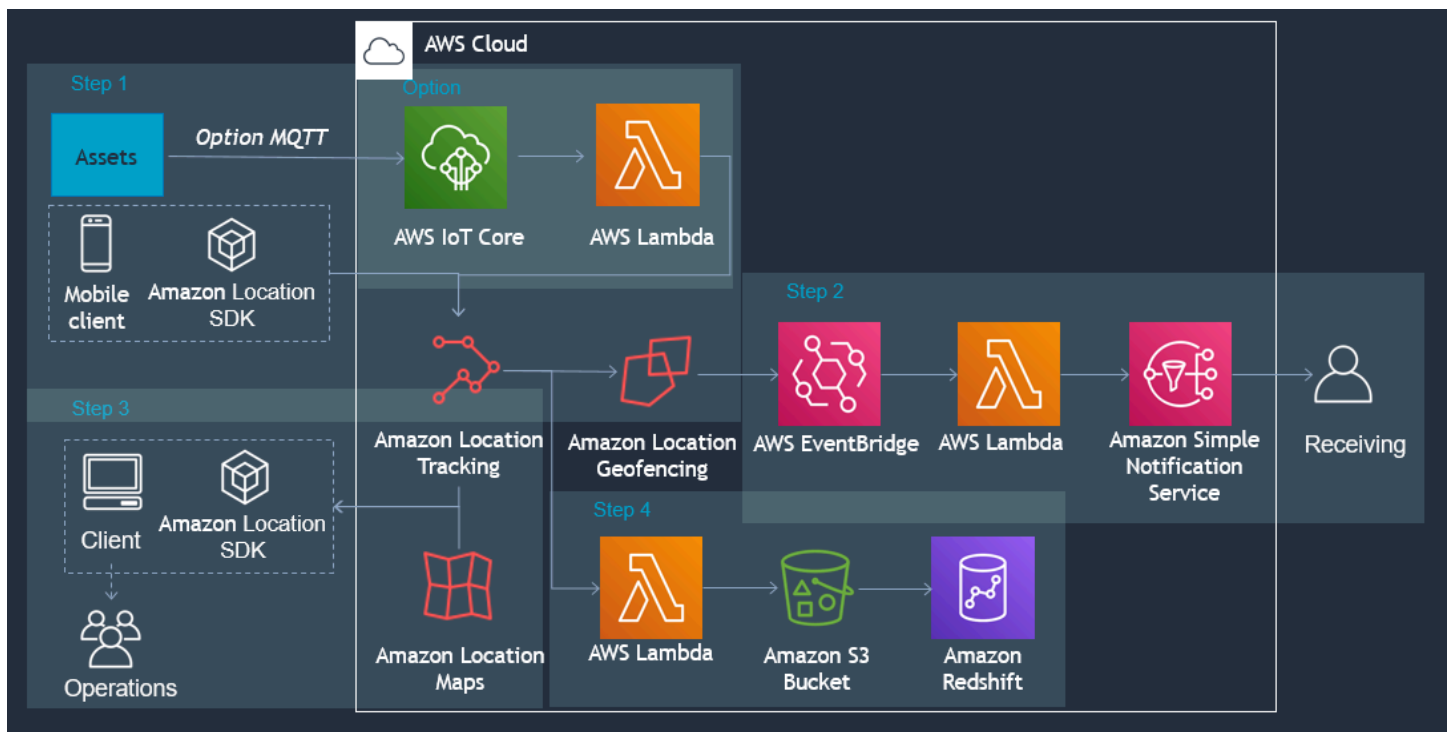
1. Cree sus geocercas en las colecciones de geocercas y vincule los rastreadores a ellas. Para obtener más información, consulte [the section called “Geocercas y rastreo”](#).
2. Configura Amazon EventBridge para que envíe una notificación a los clientes que entren o salgan de un área de interés geovallada. Para obtener más información, consulte [the section called “Reaccionar ante los eventos con EventBridge”](#).
3. Muestre las ubicaciones y geocercas de los clientes en un mapa. Para obtener más información, consulte [Uso de mapas](#).
4. Guarde los datos de ubicación en un almacenamiento a largo plazo para su posterior análisis.
5. Una vez que hayas creado tu aplicación, podrás utilizar Amazon CloudWatch y AWS CloudTrail gestionar tu aplicación. Para obtener más información, consulte [the section called “Monitorización con CloudWatch”](#) y [the section called “Uso CloudTrail con Amazon Location”](#).

Aplicaciones de seguimiento de activos

La siguiente es una ilustración de la arquitectura de una aplicación de geomarketing y participación de los usuarios que utiliza Amazon Location:

Con esta arquitectura, puede:

- Mostrar las ubicaciones de los activos en un mapa para ilustrar el panorama general. Por ejemplo, mostrar un mapa térmico con ubicaciones o eventos históricos para ayudar a un equipo de operaciones o planificación.
- Iniciar eventos en función de la proximidad de los activos para avisar al departamento de recepción a fin de prepararse para la llegada de un envío y reducir el tiempo de procesamiento.
- Almacenar las ubicaciones de los activos para iniciar acciones en sus aplicaciones de backend o para analizar los datos a lo largo del tiempo.
- Analizar el historial de ubicaciones para identificar tendencias y oportunidades de optimización.



A continuación, se proporciona una descripción general de los pasos necesarios para crear una aplicación de seguimiento de activos:

1. Cree sus geocercas en las colecciones de geocercas y vincule los rastreadores a ellas. Para obtener más información, consulte [the section called “Geocercas y rastreo”](#).

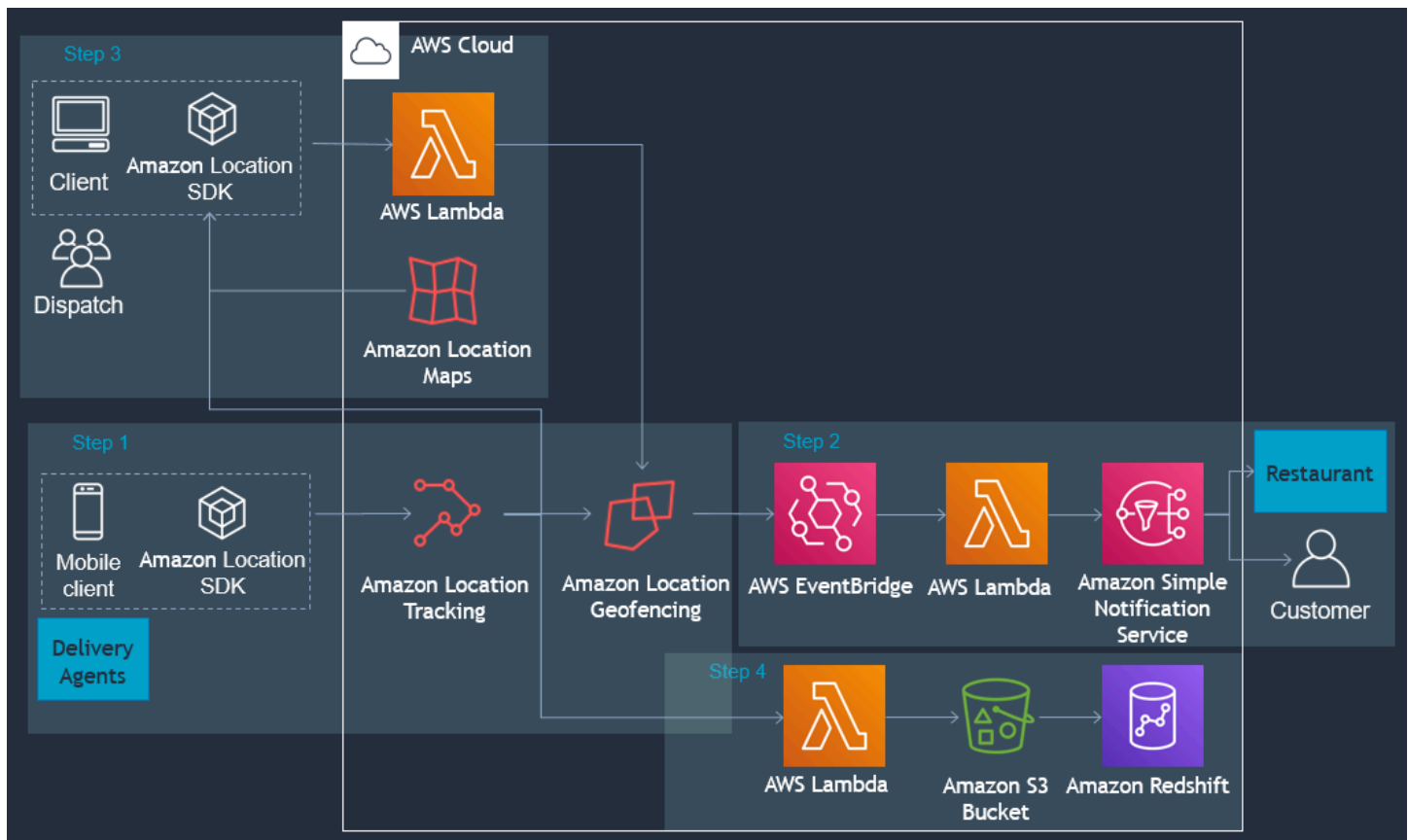
2. Configura Amazon EventBridge para enviar una notificación o iniciar un proceso. Para obtener más información, consulte [the section called “Reaccionar ante los eventos con EventBridge”](#).
3. Muestre los activos rastreados y las geocercas activas en un mapa. Para obtener más información, consulte [Uso de mapas](#).
4. Guarde los datos de ubicación en un almacenamiento a largo plazo para su posterior análisis.
5. Una vez que hayas creado tu aplicación, podrás utilizar Amazon CloudWatch y AWS CloudTrail gestionar tu aplicación. Para obtener más información, consulte [the section called “Monitorización con CloudWatch”](#) y [the section called “Uso CloudTrail con Amazon Location”](#).

Aplicaciones de entrega

La siguiente es una ilustración de la arquitectura de una aplicación de geomarketing y participación de los usuarios que utiliza Amazon Location:

Con esta arquitectura, puede:

- Iniciar eventos en función de la proximidad de los agentes de entrega para que las recogidas estén listas a tiempo y los clientes puedan recibir una notificación cuando llegue la entrega.
- Mostrar las ubicaciones de los conductores, así como las ubicaciones de recogida y entrega, prácticamente en tiempo real en un mapa para mostrar a los equipos de despacho un panorama general.
- Almacenar las ubicaciones de los agentes de reparto para que pueda actuar en función de ellas en su aplicación interna o analizarlas a lo largo del tiempo.
- Analizar el historial de ubicaciones para identificar tendencias y oportunidades de optimización.



A continuación, se proporciona una descripción general de los pasos necesarios para crear una aplicación de seguimiento de activos:

1. Cree sus colecciones de geocercas y vincule los dispositivos rastreados a la colección. Para obtener más información, consulte [the section called “Geocercas y rastreo”](#).
2. Cree una AWS Lambda función para añadir y eliminar automáticamente geovallas a medida que se van registrando sus pedidos.
3. Configura Amazon EventBridge para enviar notificaciones o iniciar un proceso. Para obtener más información, consulte [the section called “Reaccionar ante los eventos con EventBridge”](#).
4. Muestre los activos rastreados y las geocercas activas en un mapa. Para obtener más información, consulte [Uso de mapas](#).
5. Guarde los datos de ubicación en un almacenamiento a largo plazo para su posterior análisis.
6. Una vez que hayas creado tu aplicación, podrás utilizar Amazon CloudWatch y AWS CloudTrail gestionar tu aplicación. Para obtener más información, consulte [the section called “Monitorización con CloudWatch”](#) y [the section called “Uso CloudTrail con Amazon Location”](#).

¿Qué es un proveedor de datos?

Use Amazon Location Service para acceder a los recursos de geolocalización de varios proveedores de datos a través de su cuenta de AWS sin necesidad de contratos o integraciones de terceros. Esto puede ayudarlo a centrarse en crear su aplicación, sin tener que gestionar cuentas, credenciales, licencias y facturación de terceros.

Los siguientes servicios de Amazon Location utilizan proveedores de datos.

- Mapas: elija estilos de diferentes proveedores de mapas al [crear un recurso de mapa](#). Puede usar los recursos de mapas para crear un mapa interactivo para visualizar los datos.
- Ubicaciones: elija un proveedor de datos al [crear un recurso de índice de ubicación](#) para admitir consultas de geocodificación, geocodificación inversa y búsquedas.
- Rutas: elija un proveedor de datos que admita las consultas para los cálculos de rutas en diferentes geografías y aplicaciones al [crear un recurso de calculadora de rutas](#). Con el proveedor de datos que elija, Amazon Location Service le permite calcular las rutas en función de los datos de la red de up-to-date carreteras, los datos de tráfico en tiempo real, los cierres planificados y los patrones de tráfico históricos.

Cada proveedor recopila y selecciona sus datos utilizando diferentes medios. También pueden tener diferentes conocimientos en diferentes regiones del mundo. En esta sección, se proporcionan detalles sobre nuestros proveedores de datos. Puede seleccionar cualquier proveedor de datos según sus preferencias.

Cuando utilice los recursos de Amazon Location Service con proveedores de datos, asegúrese de leer los términos y condiciones del servicio. Para obtener más información, consulte las [condiciones del servicio de AWS](#). Consulte también la sección [the section called “Privacidad de datos”](#) para obtener más información sobre cómo Amazon Location protege su privacidad.

Cobertura y características del proveedor de datos

La siguiente tabla muestra la cobertura y las funciones de alto nivel para cada proveedor de datos.

Proveedores de datos	Cobertura geográfica	Cobertura de características	Región de AWS
Esri	Global	Mapas, lugares, rutas	Todas las regiones en las que Amazon

Proveedores de datos	Cobertura geográfica	Cobertura de características	Región de AWS
			Location está disponible.
Grab	Sudeste de Asia	Mapas, lugares, rutas	Asia Pacífico (Singapur), ap-southeast-1 , solamente.
AQUÍ	Global	Mapas, lugares, rutas	Todas las regiones en las que Amazon Location está disponible.
Abrir datos	Global	Mapas	Todas las regiones en las que Amazon Location está disponible.

Para obtener más información acerca de las características específicas de cada proveedor de datos, consulte [Características por proveedor de datos](#).

Cada proveedor de datos recopila y produce datos de diferentes maneras. Puede obtener más información sobre sus áreas de cobertura en los siguientes temas:

- [Cobertura: Esri](#)
- [Cobertura: Grab](#)
- [Cobertura: HERE](#)
- [Cobertura: Open Data](#)

Si tiene algún problema con los datos y desea informar de un error al proveedor de datos, consulte los siguientes temas:

- [Informe de errores a Esri](#)
- [Informe de errores para los datos GrabMaps](#)

- [Informe de errores a HERE](#)
- [Informar de errores y contribuir a los datos abiertos](#)

Estilos de mapa

Cada proveedor de datos proporciona un conjunto de estilos de mapa para representar los datos de mapa que proporcionan. Por ejemplo, un estilo puede incluir imágenes de satélite o estar optimizado para mostrar las carreteras para la navegación. Puede encontrar la lista y los ejemplos de los estilos de cada proveedor en los temas siguientes.

- [Estilos de mapa de Esri](#)
- [Estilos de mapas de Grab](#)
- [Estilos de mapas de HERE](#)
- [Estilos de mapas de datos abiertos](#)

Más información sobre cada proveedor de datos

En las siguientes secciones se proporciona información sobre cada proveedor.

- [Esri](#)
- [GrabMaps](#)
- [HERE Technologies](#)
- [Abrir datos](#)

Esri

Amazon Location Service utiliza los servicios de ubicación de Esri para ayudar a AWS los clientes a utilizar mapas, geocodificar y calcular rutas de forma eficaz. Los servicios de ubicación de Esri están diseñados con datos de ready-to-use ubicación fidedignos y de alta calidad, seleccionados por equipos expertos de cartógrafos, geógrafos y demógrafos.

Para obtener información adicional, consulte [Esri](#) en Proveedores de servicios Amazon Location Service.

Temas

- [Estilos de mapa de Esri](#)
- [Cobertura: Esri](#)
- [Condiciones de uso y atribución de datos: Esri](#)
- [Informe de errores a Esri](#)

Estilos de mapa de Esri

Amazon Location Service admite los siguientes estilos de mapa de Esri al [crear un recurso de mapa](#).

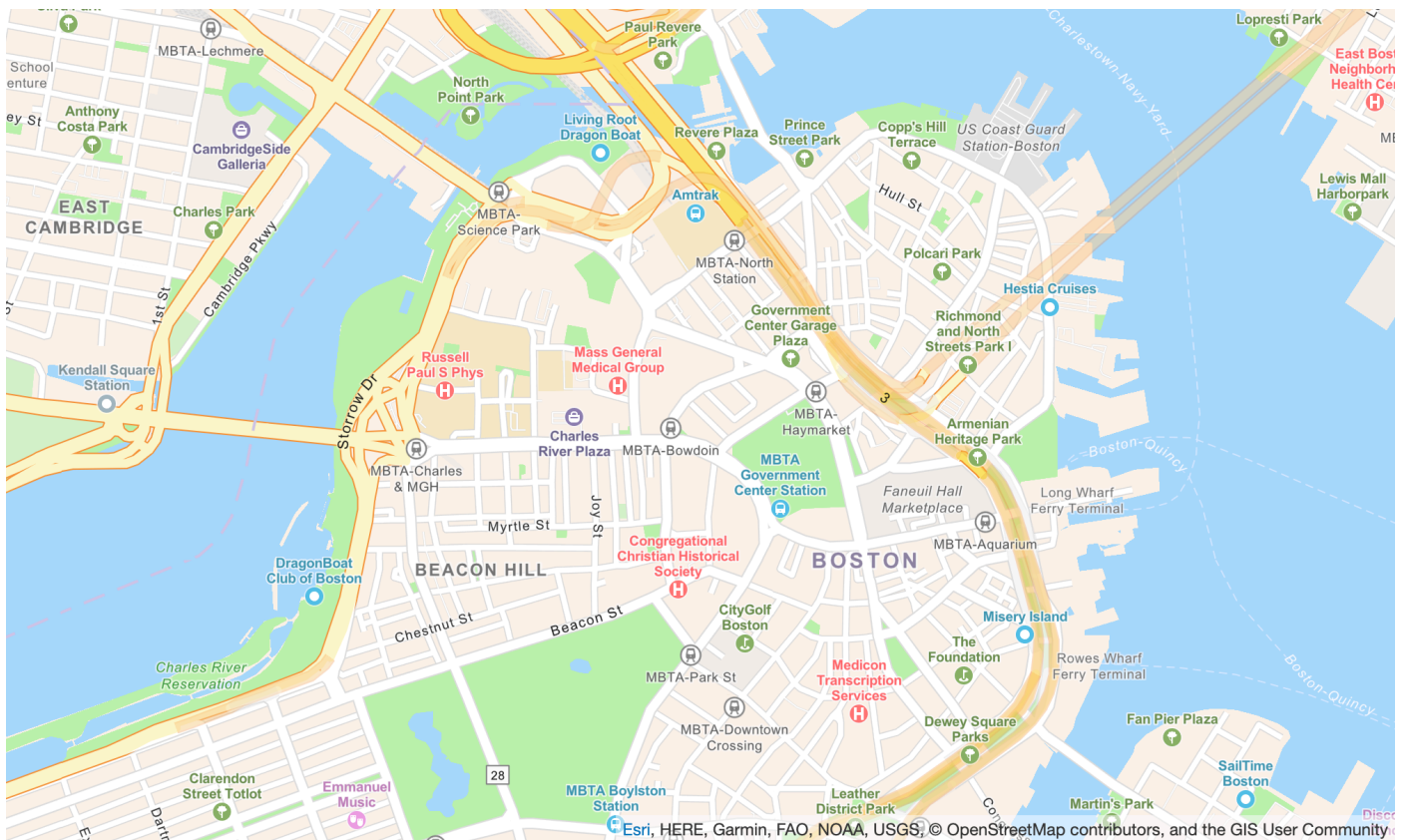
Note

No se admiten los estilos de mapa de Esri que no aparecen en esta sección.

Los estilos vectoriales de Esri admiten [Opiniones políticas](#) alternativos.

Esri Navigation

Navegación Esri




Nombre del estilo de mapa: `VectorEsriNavigation`

: el estilo de mapa World Navigation (navegación por el mundo) de Esri, que proporciona un mapa de base detallado del mundo simbolizado con un estilo de mapa de navegación personalizado, diseñado para utilizarse durante el día en dispositivos móviles.

El mapa incluye autopistas y carreteras principales, carreteras secundarias, ferrocarriles, fuentes de agua, ciudades, parques, monumentos, plantas de edificios y fronteras administrativas. La capa de mosaicos vectoriales de este mapa se creó con las mismas fuentes de datos utilizadas para el World Street Map y otros mapas base de Esri. Habilite la POI capa configurándola para aprovechar los datos de lugares adicionales. [CustomLayers](#)

Para obtener más información, consulte [Esri World Street](#) en el sitio web de Esri.

 Note

El mapa de `VectorEsriNavigation` que se muestra arriba tiene la capa POI habilitada.

Fonts

Amazon Location sirve fuentes que utilizan [GetMapGlyphs](#). Las siguientes son las pilas de fuentes disponibles para este mapa:

- Arial Italic
- Arial Regular
- Arial Bold
- Arial Unicode MS Bold
- Arial Unicode MS Regular

Esri Imagery

Esri Imagery

Nombre del estilo de mapa: `RasterEsriImagery`

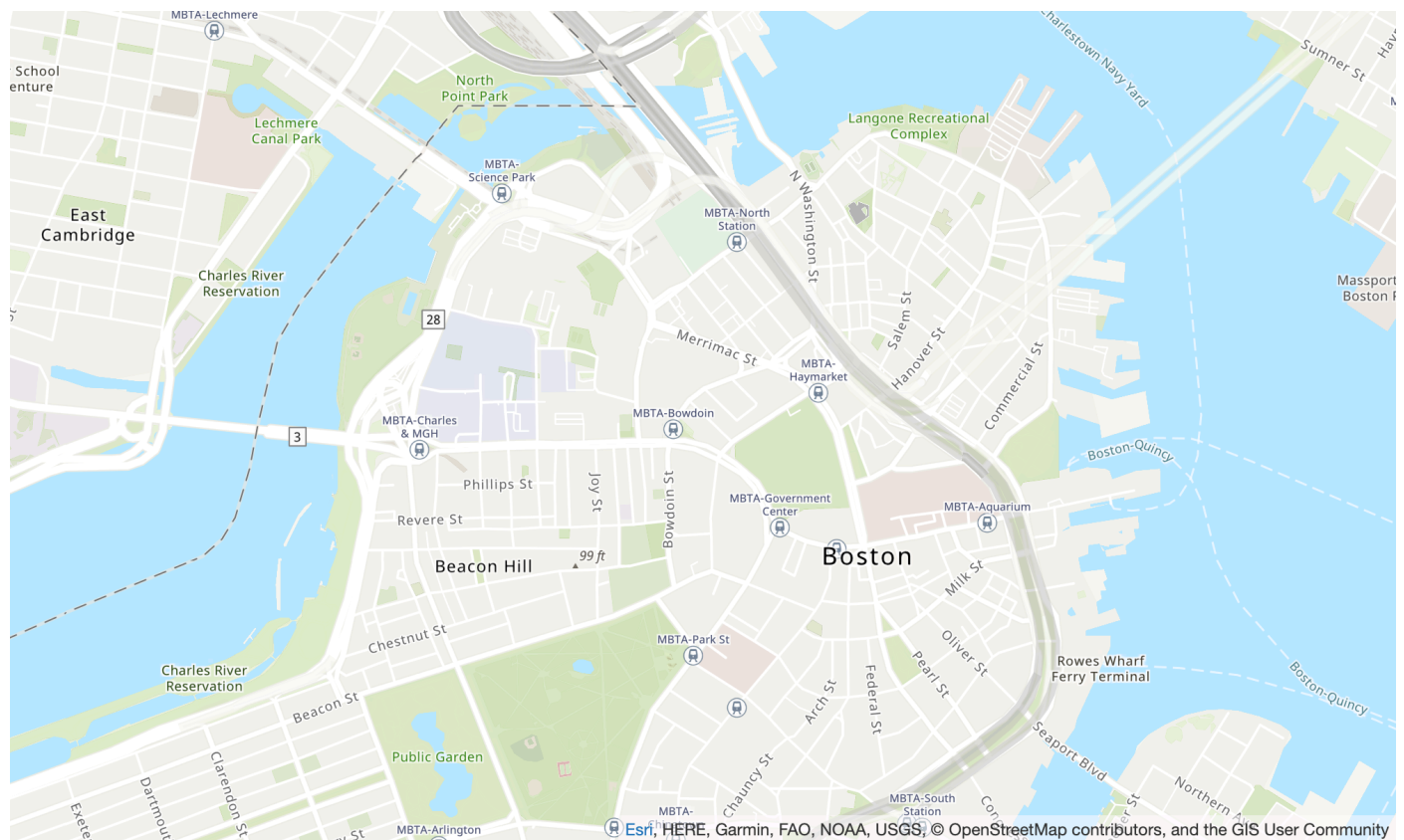
Un mapa de base de cuadrícula que proporciona imágenes satelitales y aéreas de un metro o más en muchas partes del mundo e imágenes satelitales de menor resolución en todo el mundo.

El mapa incluye 15 millones de imágenes a escala pequeña y media (de ~ 1:591 M a ~ 1:72 k) e imágenes SPOT de 2,5 millones (de ~ 1:288 k a ~ 1:72 k) del mundo. El mapa incluye imágenes de Maxar con una resolución de 0,5 m del territorio continental de los Estados Unidos y partes de Europa occidental. Este mapa incluye imágenes adicionales del submetro de Maxar en muchas partes del mundo. En otras partes del mundo, la comunidad de usuarios de SIG ha aportado imágenes en diferentes resoluciones. En determinadas comunidades, están disponibles imágenes de muy alta resolución (hasta 0,03 m) en una escala de aproximadamente 1:280.

Para obtener más información, consulte [Esri World Imagery](#) en el sitio Web de Esri.

Esri Light

Esri Light



Nombre del estilo de mapa: `VectorEsriTopographic`

Esto proporciona un mapa base detallado del mundo simbolizado con un estilo de mapa clásico de Esri. El mapa incluye autopistas y carreteras principales, carreteras secundarias, ferrocarriles, fuentes de agua, ciudades, parques, monumentos, plantas de edificios y fronteras administrativas.

Este mapa base se ha compilado a partir de diversas fuentes autorizadas de varios proveedores de datos, como el Servicio Geológico de los Estados Unidos (USGS), la Agencia de Protección Ambiental (EPA) de los Estados Unidos, el Servicio de Parques Nacionales (NPS), la Organización de las Naciones Unidas para la Alimentación y la Agricultura (FAO), el Departamento de Recursos Naturales de Canadá (NRCAN), HERE y Esri. Los datos de áreas seleccionadas provienen de OpenStreetMap los colaboradores. Además, los datos los proporciona la comunidad de SIG.

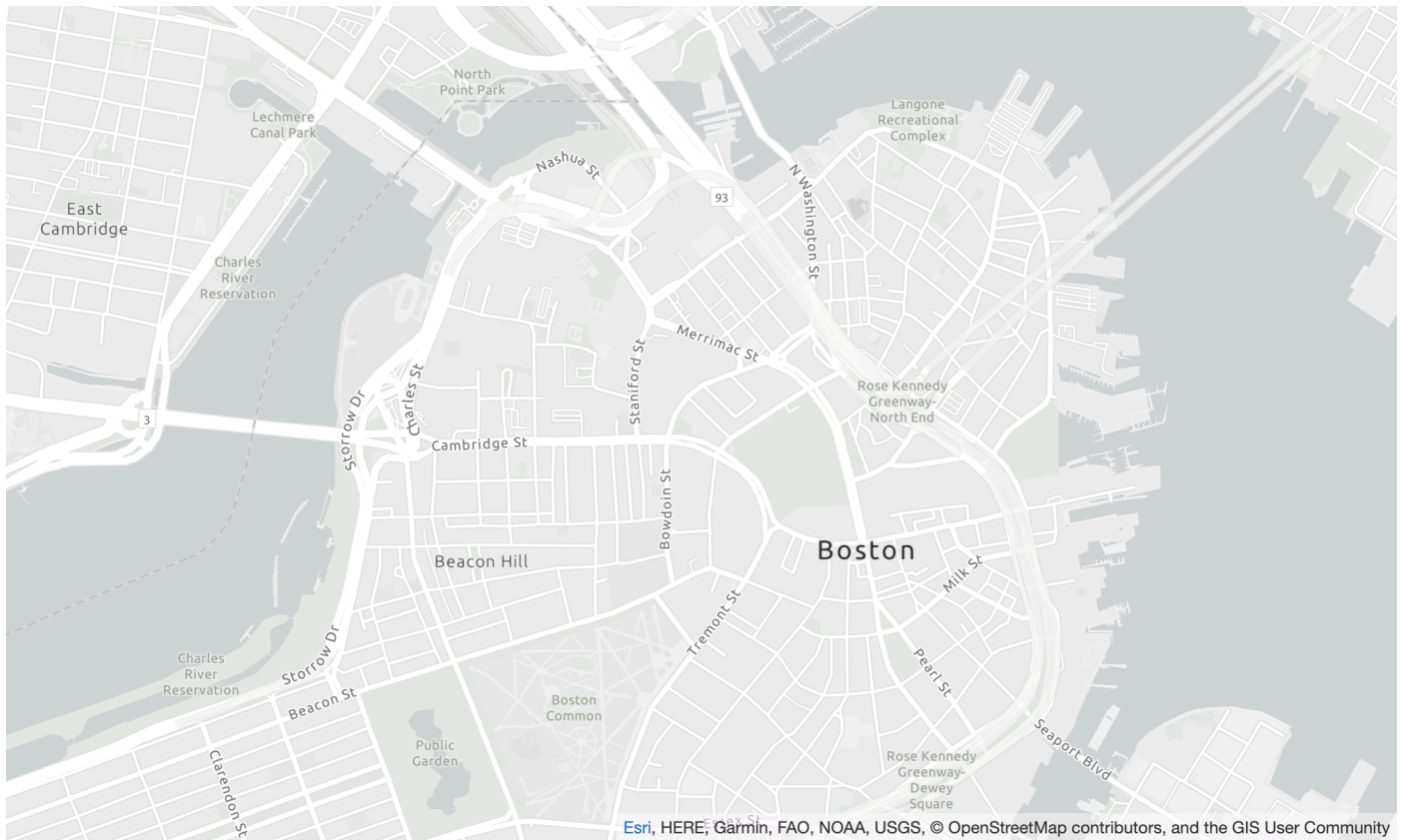
Fonts

Amazon Location sirve fuentes que utilizan [GetMapGlyphs](#). Las siguientes son las pilas de fuentes disponibles para este mapa:

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

Esri Light Gray Canvas

Esri Light Gray Canvas



Nombre del estilo de mapa: VectorEsriLightGrayCanvas

Un mapa de base vectorial con un fondo gris claro y neutro con colores, etiquetas y características mínimos, diseñado para llamar la atención sobre el contenido temático.

Esta capa de mosaicos vectoriales se ha creado con los mismos orígenes de datos que se utilizan para el Light Grey Canvas y otros mapas base de Esri. El mapa incluye autopistas y carreteras principales, carreteras secundarias, ferrocarriles, fuentes de agua, ciudades, parques, monumentos, plantas de edificios y fronteras administrativas.

Para obtener más información, consulte [Esri Dark Grey Canvas](#) en el sitio web de Esri.

Fonts

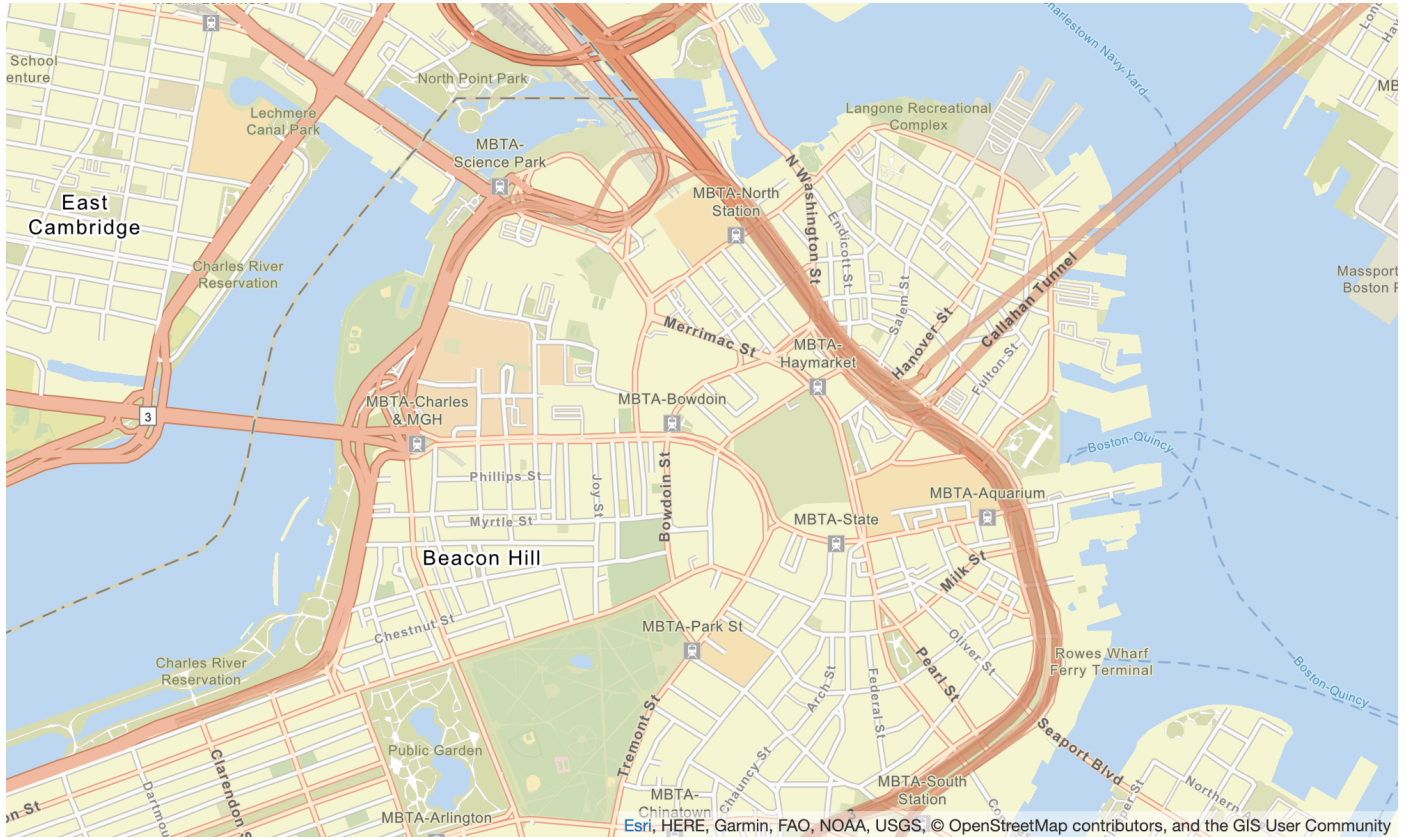
Amazon Location sirve fuentes que utilizan [GetMapGlyphs](#). Las siguientes son las pilas de fuentes disponibles para este mapa:

- Ubuntu Italic
- Ubuntu Regular
- Ubuntu Light

- Ubuntu Bold

Esri Street Map

Esri Street Map



Nombre del estilo de mapa: `VectorEsriStreets`

: el estilo de mapa World Navigation (navegación por el mundo) de Esri, que proporciona un mapa de base detallado del mundo simbolizado con un estilo de mapa de navegación personalizado, diseñado para utilizarse durante el día en dispositivos móviles.

El mapa incluye autopistas y carreteras principales, carreteras secundarias, ferrocarriles, fuentes de agua, ciudades, parques, monumentos, plantas de edificios y fronteras administrativas. También incluye un conjunto más enriquecido de lugares, como tiendas, servicios, restaurantes, atracciones y otros puntos de interés. La capa de mosaicos vectoriales de este mapa se creó con las mismas fuentes de datos utilizadas para el World Street Map y otros mapas base de Esri.

Para obtener más información, consulte [Esri World Street](#) en el sitio web de Esri.

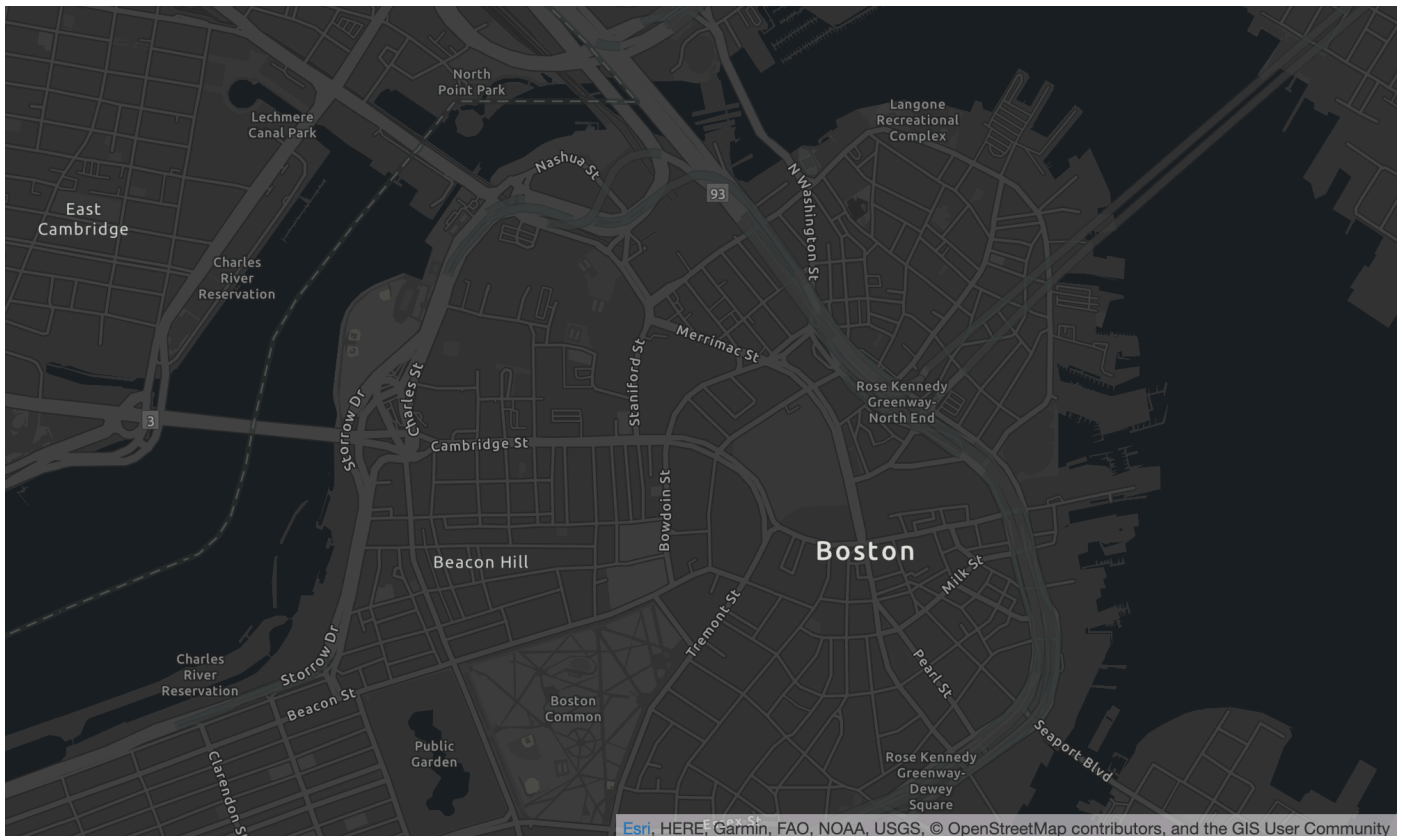
Fonts

Amazon Location sirve fuentes que utilizan [GetMapGlyphs](#). Las siguientes son las pilas de fuentes disponibles para este mapa:

- Arial Italic
- Arial Regular
- Arial Bold
- Arial Unicode MS Bold
- Arial Unicode MS Regular

Esri Dark Gray Canvas

Esri Dark Gray Canvas



Nombre del estilo de mapa: `VectorEsriDarkGrayCanvas`

Un mapa de base vectorial con un fondo gris oscuro y neutro con colores, etiquetas y características mínimos, diseñado para llamar la atención sobre el contenido temático.

El mapa incluye autopistas y carreteras principales, carreteras secundarias, ferrocarriles, fuentes de agua, ciudades, parques, monumentos, plantas de edificios y fronteras administrativas. Las

capas de mosaicos vectoriales de este mapa se crean con las mismas fuentes de datos utilizadas para el mapa ráster Dark Gray Canvas y otros mapas base de Esri.

Para obtener más información, consulte [Esri Dark Grey Canvas](#) en el sitio web de Esri.

Fonts

Amazon Location sirve fuentes que utilizan [GetMapGlyphs](#). Las siguientes son las pilas de fuentes disponibles para este mapa:

- Ubuntu Medium en cursiva
- Ubuntu Medium
- Ubuntu Italic
- Ubuntu Regular
- Ubuntu Bold

Cobertura: Esri

Puede usar Esri como proveedor de datos para admitir consultas de geocodificación, geocodificación inversa y búsquedas al [crear un recurso de índice de ubicación](#), o para admitir consultas para calcular una ruta cuando [crea un recurso de calculadora de rutas](#).

Esri ofrece distintos niveles de calidad de datos en distintas regiones del mundo. Para obtener información adicional sobre la cobertura en su región de interés, consulte:

- [Detalles de Esri sobre la cobertura de geocodificación](#)
- [Detalles de Esri sobre las redes de calles y la cobertura del tráfico](#)

Condiciones de uso y atribución de datos: Esri

Antes de usar los datos de Esri, asegúrese de cumplir con todos los requisitos legales aplicables, incluidos los términos de licencia aplicables a Esri y AWS.

Para obtener más información sobre los AWS requisitos, consulte [las condiciones de servicio de AWS](#).

Para obtener información sobre las directrices de atribución de Esri, consulte las [atribuciones de datos y las condiciones de uso de Esri](#).

Informe de errores a Esri

Si tiene algún problema con los datos y quiere informar a Esri de los errores y discrepancias, consulte el artículo de soporte técnico de Esri sobre [cómo proporcionar comentarios sobre](#) los mapas base y la geocodificación.

GrabMaps

Grab es la organización de reparto más grande del sudeste asiático, con millones de socios conductores y clientes. Su filial crea datos up-to-date cartográficos en esos países o regiones para su propio uso, entre otros. [GrabMaps](#) Amazon Location Service utiliza GrabMaps «servicios de ubicación» para ayudar a AWS los clientes a utilizar mapas, geocodificar y calcular rutas de forma eficaz. GrabMaps«Los servicios de ubicación están diseñados para proporcionar datos de ready-to-use ubicación fidedignos y de alta calidad, específicamente para los países del sudeste asiático.

Para obtener información sobre la capacidad adicional, consulte [GrabMaps](#) sobre los proveedores de datos de Amazon Location Service.

Important

Grab solo brinda mapas de países del sudeste asiático y solo está disponible en la región Asia-Pacífico (Singapur) (ap-southeast-1). Para obtener más información, consulte [Países, regiones y áreas cubiertas](#).

Temas

- [Estilos de mapas de Grab](#)
- [Cobertura: Grab](#)
- [Países, regiones y áreas cubiertas](#)
- [Condiciones de uso y atribución de datos: Grab](#)
- [Informe de errores para los datos GrabMaps](#)

Estilos de mapas de Grab

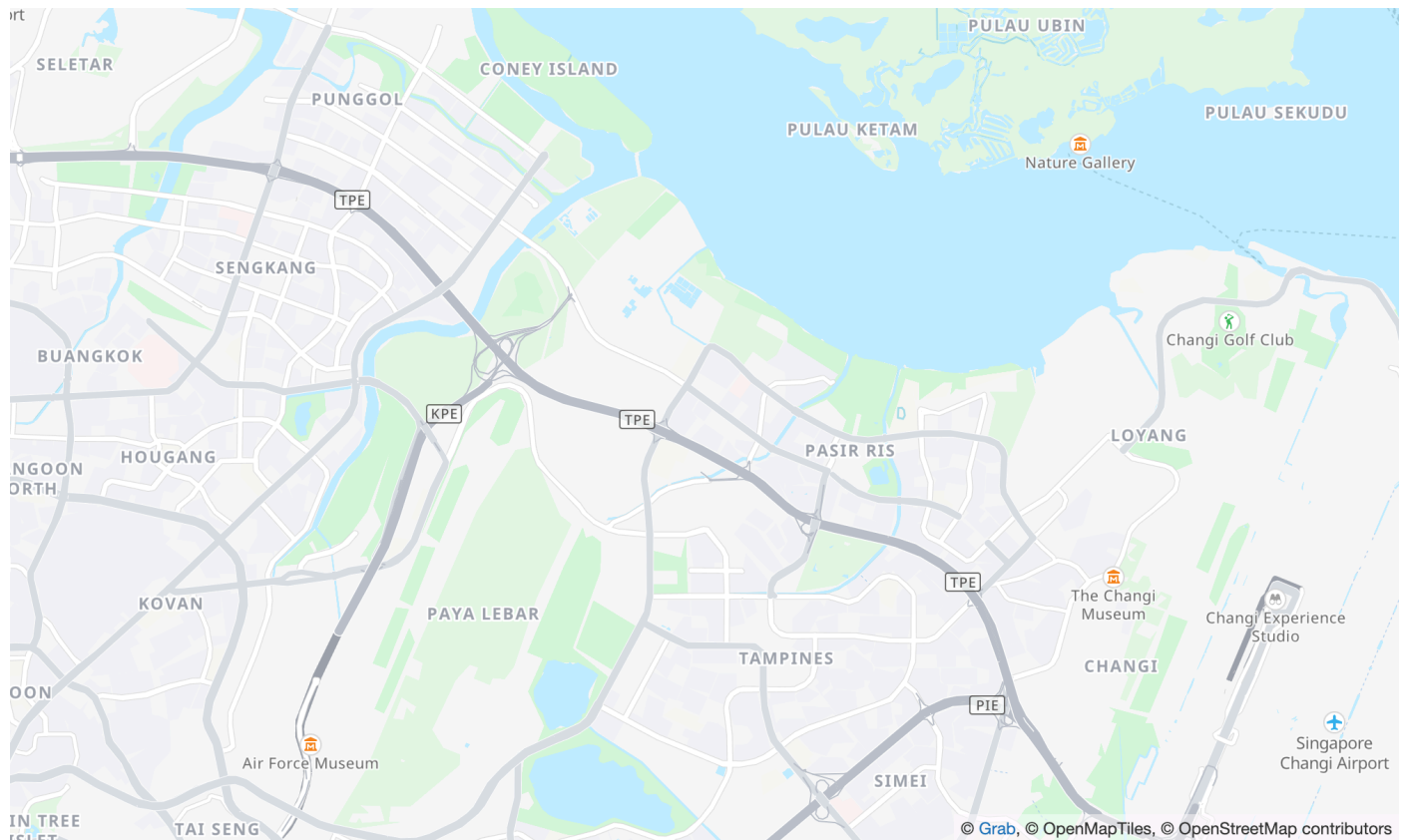
Amazon Location Service admite los siguientes estilos de mapa de Esri al [crear un recurso de mapa](#):

Note

Los estilos de mapa Grab que no aparecen en esta sección no son compatibles actualmente.

Grab Standard Light Map

Grab Standard Light Map



Nombre del estilo de mapa: `VectorGrabStandardLight`

: el estilo de mapa estándar de Grab brinda un mapa de base con colores detallados que indican los usos del suelo, los nombres de las áreas, las carreteras, los monumentos y los puntos de interés que cubren el sudeste asiático.

Fonts

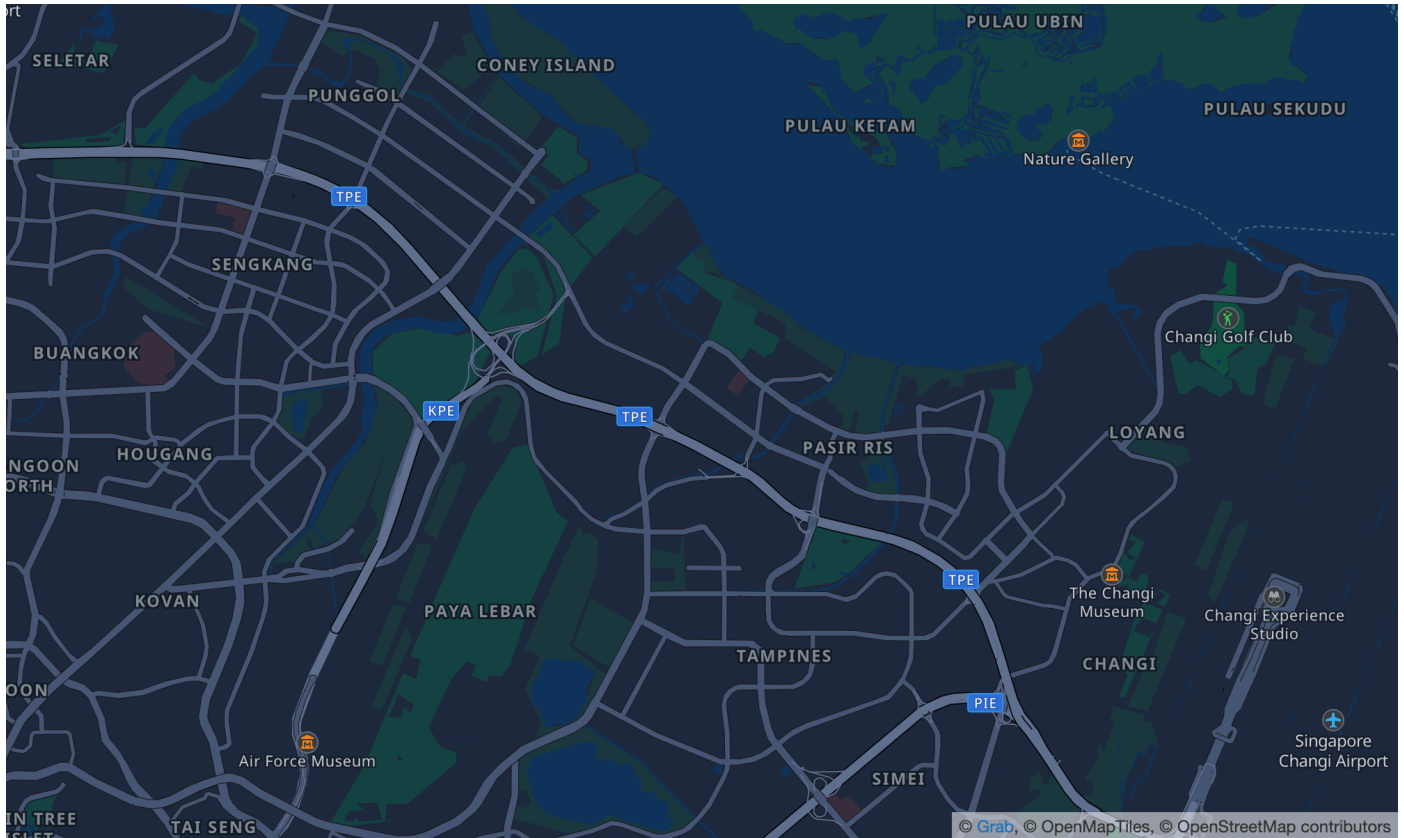
Amazon Location sirve fuentes que utilizan [GetMapGlyphs](#). Las siguientes son las pilas de fuentes disponibles para este mapa:

- Noto Sans Regular

- Noto Sans Medium
- Noto Sans Bold

Grab Standard Dark Map

Grab Standard Dark Map



Nombre del estilo de mapa: `VectorGrabStandardDark`

La variante oscura de Grab brinda un mapa de base con colores detallados que indican los usos del suelo, los nombres de las áreas, las carreteras, los monumentos y los puntos de interés que cubren el sudeste asiático.

Fonts

Amazon Location sirve fuentes que utilizan [GetMapGlyphs](#). Las siguientes son las pilas de fuentes disponibles para este mapa:

- Noto Sans Regular
- Noto Sans Medium

- Noto Sans Bold

Cobertura: Grab

Puede usar Grab como proveedor de datos para admitir consultas de geocodificación, geocodificación inversa y búsquedas al [crear un recurso de índice de ubicación](#), o para admitir consultas para calcular una ruta cuando [crea un recurso de calculadora de rutas](#).

Países, regiones y áreas cubiertas

Grab solo brinda mapas de países del sudeste asiático y solo está disponible en la región Asia-Pacífico (Singapur) (ap-southeast-1).

Grab proporciona datos detallados de los siguientes países o regiones:

- Malasia
- Filipinas
- Tailandia
- Singapur
- Vietnam
- Indonesia
- Myanmar
- Camboya

Note

Fuera de estas áreas, los recursos de Amazon Location Service creados con Grab como proveedor de datos no proporcionarán ningún resultado. Esto incluye los resultados de búsqueda o las rutas.

Los mapas de Grab se encuentran dentro de los siguientes límites:

- Sur: latitud -21.943045533438166
- Oeste: longitud 90.0
- Norte: latitud 31.952162238024968

- Este: longitud 146.25

Para los niveles de zoom del 1 al 4, Grab incluye cobertura global. Para los niveles de zoom 5 o inferiores, los mosaicos del mapa solo se proporcionan dentro de este cuadro delimitado.

Note

Fuera de este cuadro delimitado, los recursos de mapas de Amazon Location Service creados con Grab como proveedor de datos no mostrarán mosaicos de mapas. Para evitar errores 404 en su aplicación, puede limitar el mapa con un cuadro delimitador, tal y como se describe en [Establecer las extensiones de un mapa mediante MapLibre](#).

Modos de desplazamiento de Grab

En cuanto a las rutas, Grab ofrece rutas para automóviles y motocicletas para todos los países o regiones enumerados anteriormente.

Grab no admite rutas para camiones.

Para rutas en bicicleta y a pie, Grab es compatible con las siguientes ciudades:.

- Singapur
- Yakarta
- Manila
- Valle de Klang
- Bangkok
- Ciudad de Ho Chi Minh
- Hanói

Condiciones de uso y atribución de datos: Grab

Al utilizar los datos de Grab, debes cumplir con todos los requisitos legales aplicables, incluidos los términos de licencia aplicables a Grab y AWS.

Para obtener más información sobre los AWS requisitos, consulte [las condiciones de servicio de AWS](#).

Para obtener información sobre las «directrices de atribución GrabMaps», consulte la sección 9.23 de las [atribuciones de datos y las condiciones](#) de uso de Grab.

Informe de errores para los datos GrabMaps

Si tiene algún problema con los datos de origen y desea informar de errores o discrepancias GrabMaps, póngase en [contacto con el servicio de asistencia AWS técnica](#).

HERE Technologies

Amazon Location Service utiliza los servicios de ubicación de HERE Technologies para ayudar a AWS los clientes a utilizar mapas, geocodificar y calcular rutas de forma eficaz. Los datos de ubicación de HERE ofrecen una plataforma centrada en la ubicación que es abierta, segura y privada. Al seleccionar los datos de ubicación de HERE, está seleccionando datos precisos, actualizados y sólidos que se despliegan de forma nativa en la nube AWS .

Para obtener información adicional, consulte [HERE](#) en Proveedores de servicios Amazon Location Service.

Temas

- [Estilos de mapas de HERE](#)
- [Cobertura: HERE](#)
- [Condiciones de uso y atribución de datos: HERE](#)
- [Informe de errores a HERE](#)

Estilos de mapas de HERE

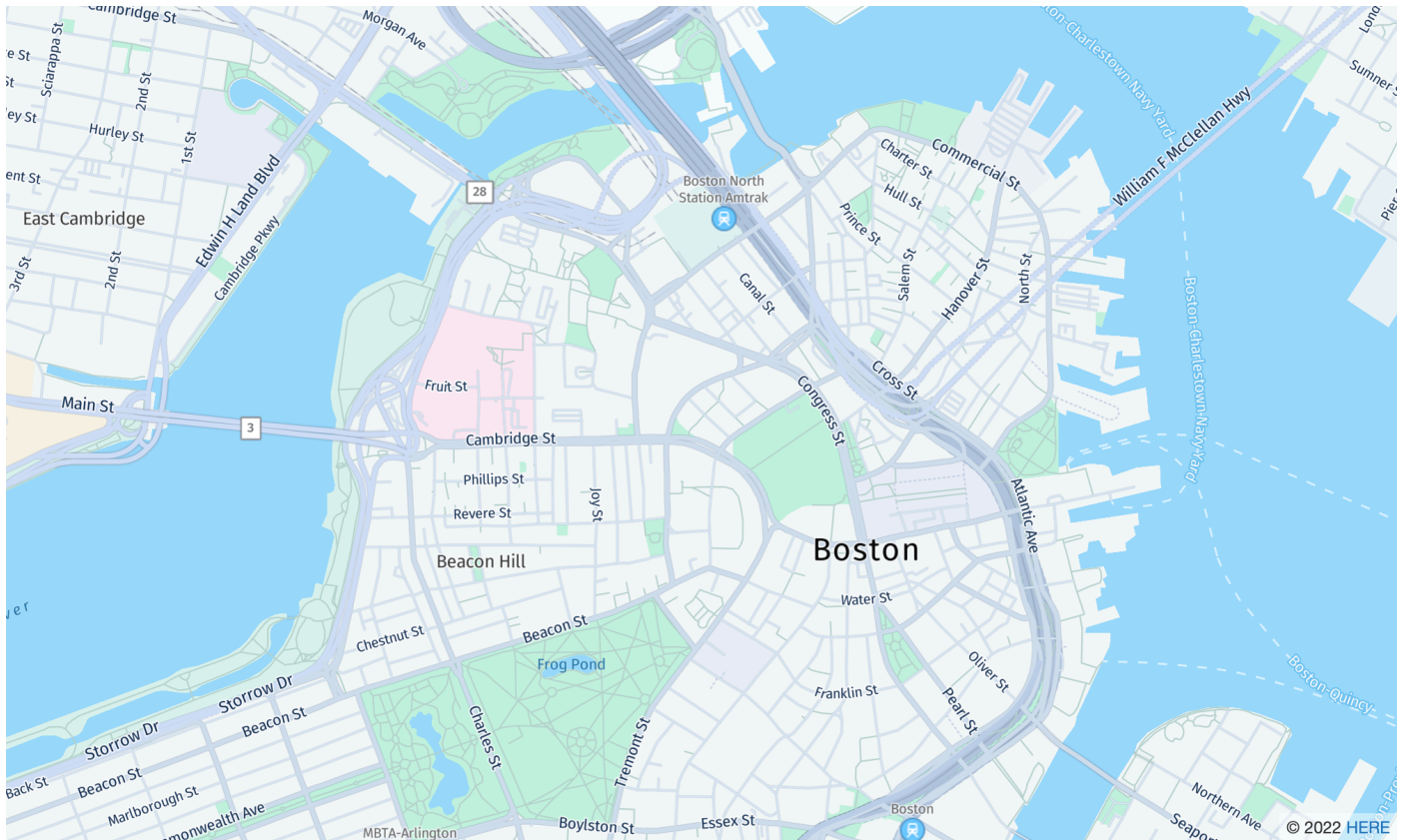
Amazon Location Service admite los siguientes estilos de mapa de HERE al [crear un recurso de mapa](#):

Note

Los estilos de mapa HERE que no aparecen en esta sección no son compatibles actualmente.

HERE Explore

HERE Explore



Nombre del estilo de mapa: VectorHereExplore

HERE Explore

Un mapa base detallado y neutral del mundo. El mapa incluye autopistas y carreteras principales, carreteras secundarias, ferrocarriles, fuentes de agua, ciudades, parques, monumentos, plantas de edificios y fronteras administrativas. Ahora también incluye un mapa de Japón completamente diseñado.

Fonts

Amazon Location sirve fuentes que utilizan [GetMapGlyphs](#). Las siguientes son las pilas de fuentes disponibles para este mapa:

- Fira GO Italic
- Fira GO Regular
- Fira GO Bold
- Noto Sans CJK JP Light
- Noto Sans CJK JP Regular

- Noto Sans CJK JP Bold

HERE Imagery

HERE Imagery



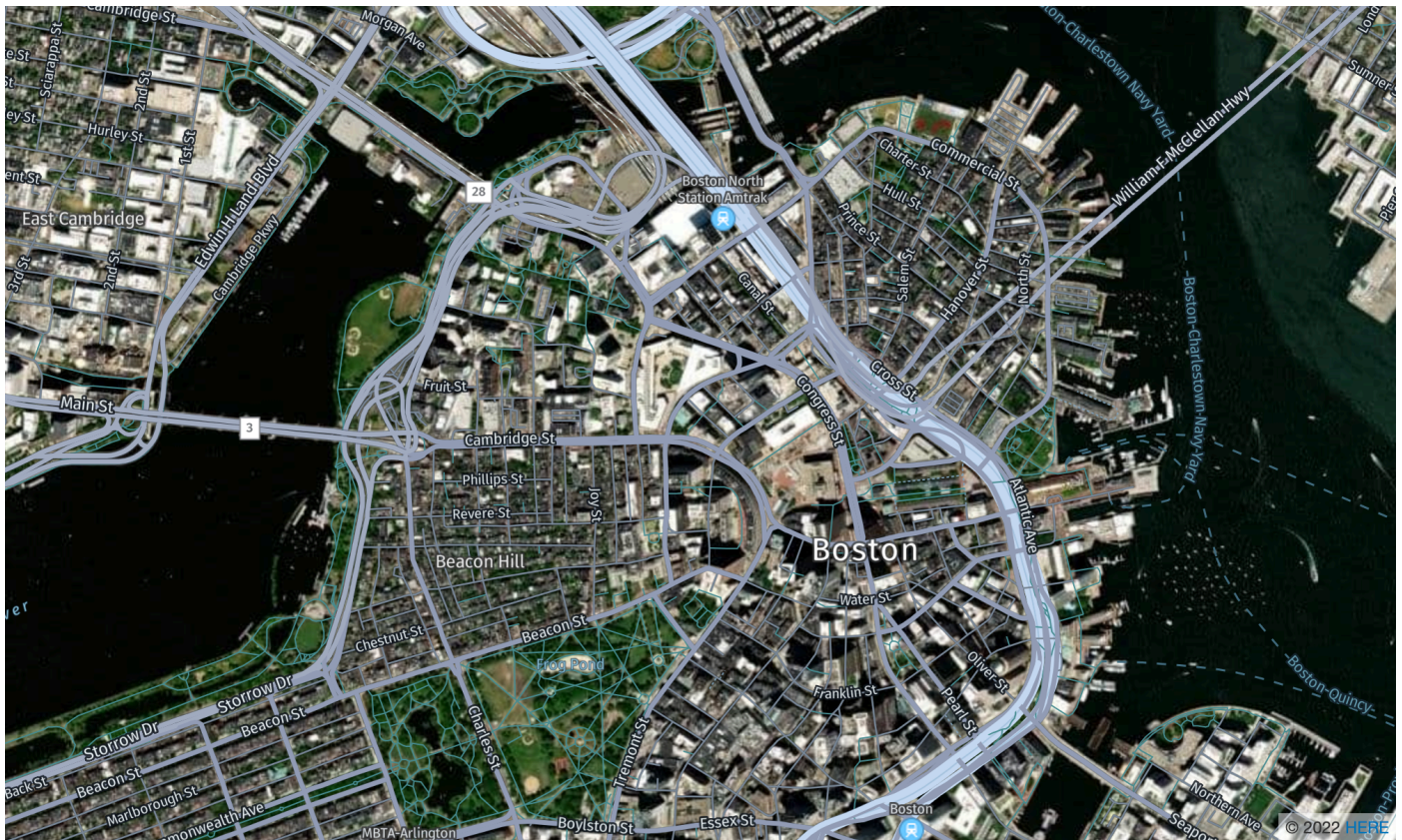
Nombre del estilo de mapa: RasterHereExploreSatellite

HERE Imagery

HERE Imagery proporciona imágenes satelitales de alta resolución con cobertura global.

HERE Hybrid

HERE Hybrid



Nombre del estilo de mapa: HybridHereExploreSatellite

HERE Hybrid

El estilo HERE Hybrid muestra la red de carreteras, los nombres de las calles y las etiquetas de las ciudades en imágenes de satélite. Este estilo superpone dos mosaicos del mapa: la imagen de satélite (mosaico rasterizado) en el fondo y la red de carreteras y las etiquetas (mosaico vectorial) en la parte superior. Este estilo recuperará automáticamente los mosaicos rasterizados y vectoriales necesarios para renderizar el mapa.

i Note

Los estilos híbridos utilizan mosaicos vectoriales y rasterizado para representar el mapa que se ve. Esto significa que se recuperan más mosaicos que cuando se utilizan únicamente de tipo vectorial o rasterizado. Los cargos incluirán todas los mosaicos recuperados.

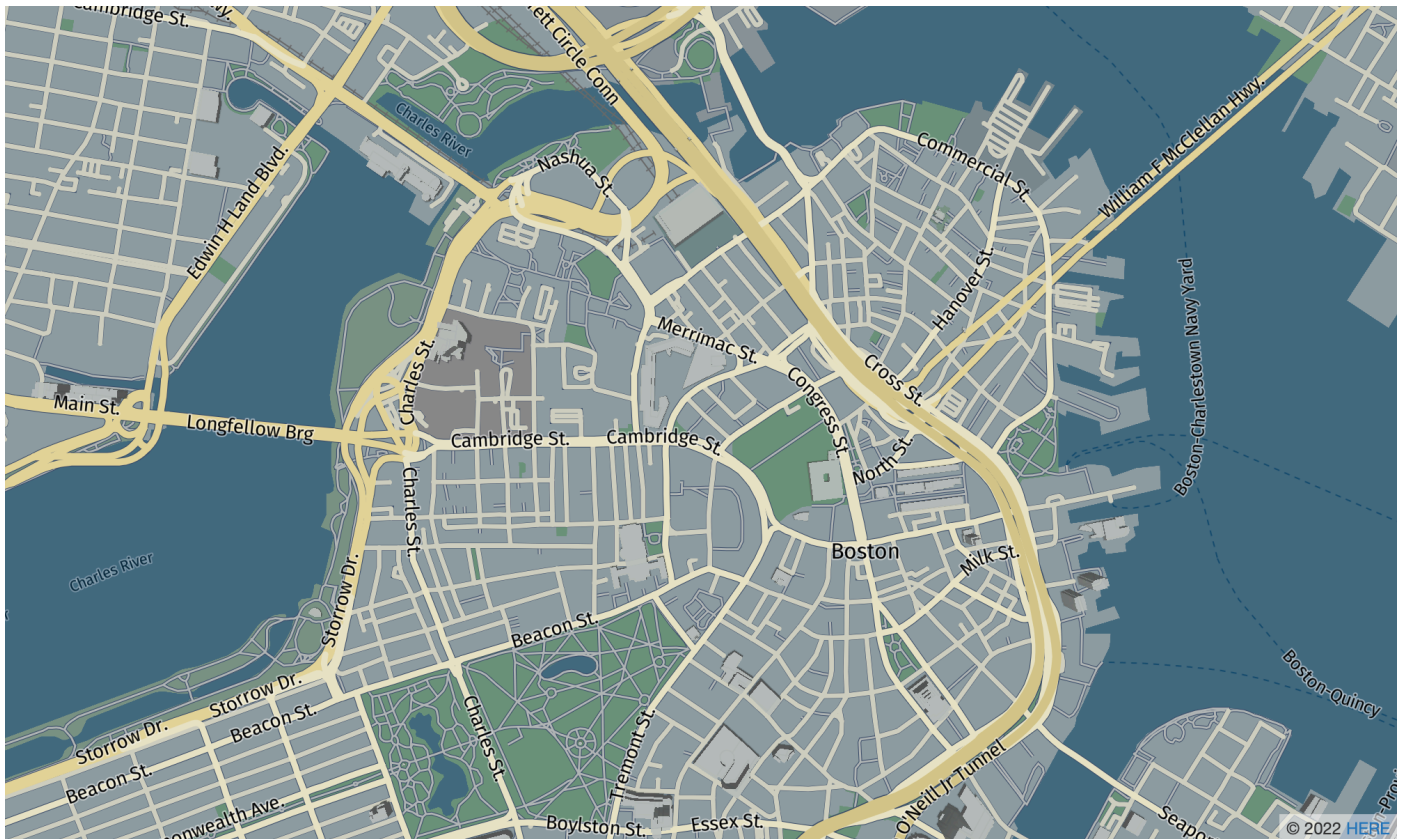
Fonts

Amazon Location sirve fuentes que utilizan [GetMapGlyphs](#). Las siguientes son las pilas de fuentes disponibles para este mapa:

- Fira GO Italic
- Fira GO Regular
- Fira GO Bold
- Noto Sans CJK JP Light
- Noto Sans CJK JP Regular
- Noto Sans CJK JP Bold

HERE Contrast (Berlin)

HERE Contrast (Berlin)



Nombre del estilo de mapa: VectorHereContrast

HERE Contrast (Berlin)

Un mapa básico detallado del mundo que combina renderizados 3D y 2D. El mapa de alto contraste incluye autopistas y carreteras principales, carreteras secundarias, ferrocarriles, fuentes de agua, ciudades, parques, monumentos, plantas de edificios y fronteras administrativas.

Fonts

Amazon Location sirve fuentes que utilizan [GetMapGlyphs](#). Las siguientes son las pilas de fuentes disponibles para este mapa:

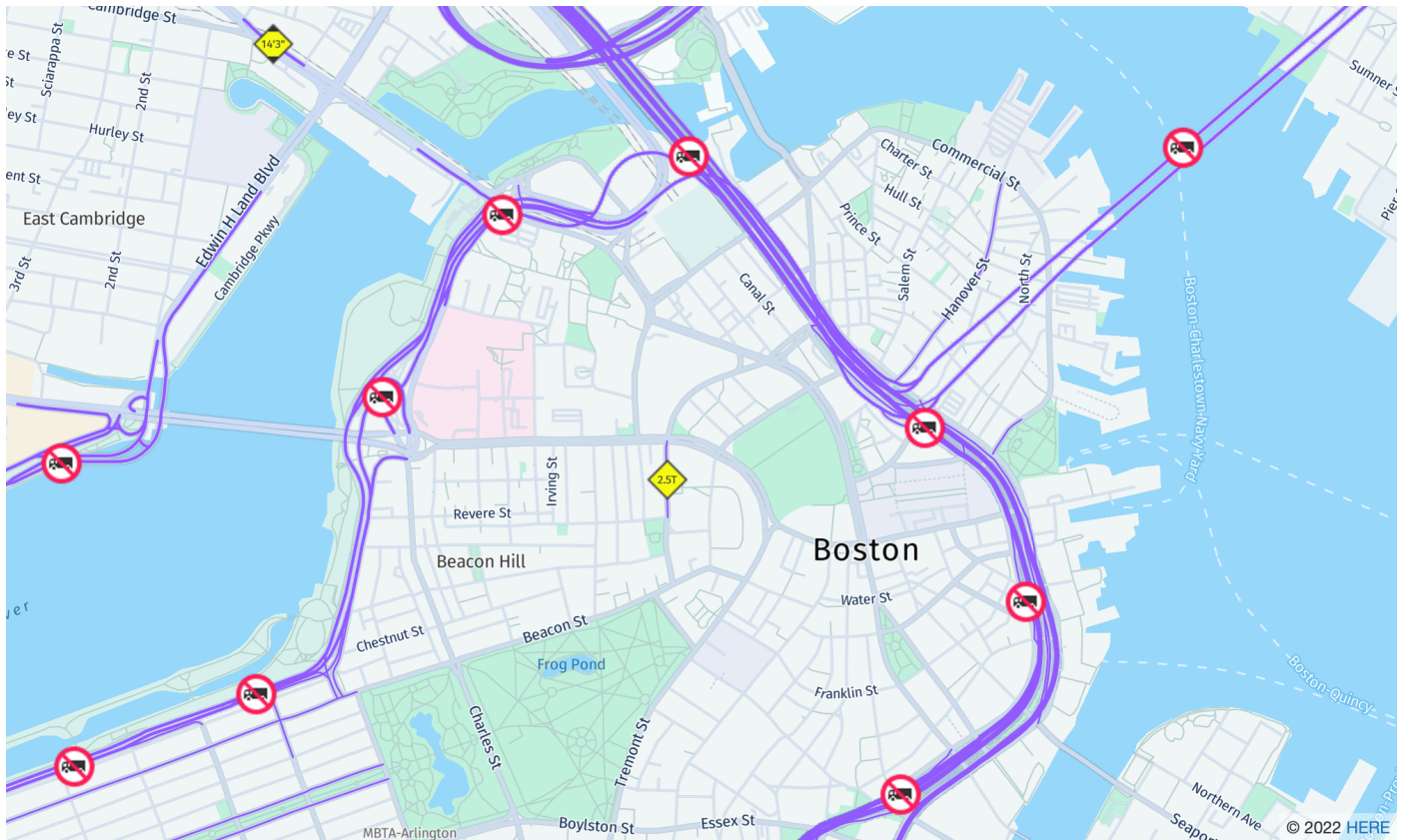
- Fira GO Regular
- Fira GO Bold

Note

Se cambió el nombre de este estilo de `VectorHereBerlin` (HERE Berlin maps). `VectorHereBerlin` está obsoleto, pero seguirá funcionando en las aplicaciones que lo utilizan.

HERE Explore Truck

HERE Explore Truck



Nombre del estilo de mapa: VectorHereExploreTruck

HERE Explore Truck

Un mapa base detallado y neutral del mundo. El mapa de calles se basa en el estilo HERE Explore y destaca las restricciones y atributos de la pista (como el ancho, la altura y si transporta materiales peligrosos) con símbolos e iconos para apoyar los casos prácticos dentro de las áreas de transporte y logística.

Fonts

Amazon Location sirve fuentes que utilizan [GetMapGlyphs](#). Las siguientes son las pilas de fuentes disponibles para este mapa:

- Fira GO Italic
- Fira GO Regular
- Fira GO Bold
- Noto Sans CJK JP Light
- Noto Sans CJK JP Regular

- Noto Sans CJK JP Bold

Para obtener información adicional sobre la calidad de los datos cartográficos en diferentes regiones del mundo, consulte la cobertura cartográfica de [HERE](#).

Cobertura: HERE

Puede usar HERE como proveedor de datos para admitir consultas de geocodificación, geocodificación inversa y búsquedas al [crear un recurso de índice de ubicación](#), o para admitir consultas para calcular una ruta cuando [crea un recurso de calculadora de rutas](#).

HERE proporciona diferentes niveles de calidad de datos en diferentes regiones del mundo. Para obtener información adicional sobre la cobertura en su región de interés, consulte:

- [HERE: cobertura de geocodificación](#)
- [HERE: Cobertura de rutas para automóviles](#)
- [HERE Cobertura de rutas para camiones](#)

Condiciones de uso y atribución de datos: HERE

Antes de utilizar los datos de HERE, asegúrese de cumplir con todos los requisitos legales aplicables, incluidos los términos de licencia aplicables a HERE y AWS. Debido a las limitaciones de la licencia, no puede utilizar HERE para almacenar los resultados de geocodificación de ubicaciones en Japón.

Para obtener información sobre los AWS requisitos, consulte [las condiciones de servicio de AWS](#).

Para obtener información adicional sobre las directrices de atribución de HERE, consulte la sección 2 de las [condiciones para proveedores de HERE Technologies aplicables a la ubicación y otro contenido](#).

Informe de errores a HERE

Para informar de errores y discrepancias en el mapa a HERE, vaya a <https://www.here.com/contact> y seleccione Informar de un error en el mapa.

Abrir datos

Amazon Location Service proporciona acceso a datos de mapas de código abierto a través del proveedor de datos abiertos. Open Data proporciona mapas base globales creados a partir de

la [distribución de mapas Daylight](#) de [OpenStreetMap OSM](#), [Natural Earth](#) y otras fuentes de datos abiertas. Los mapas proporcionados están diseñados para ser compatibles con diferentes aplicaciones y casos de uso, incluida la logística y la entrega, y la visualización de datos en entornos web y móviles. Con más de un millón de creadores de mapas, la comunidad de OSM actualiza cientos de miles de funciones por día. Amazon Location Service incorpora estas modificaciones con regularidad.

Para obtener información adicional, consulte [HERE](#) en Proveedores de servicios Amazon Location Service.

Temas

- [Estilos de mapas de datos abiertos](#)
- [Cobertura: Open Data](#)
- [Condiciones de uso y atribución de datos: Open Data](#)
- [Informar de errores y contribuir a los datos abiertos](#)

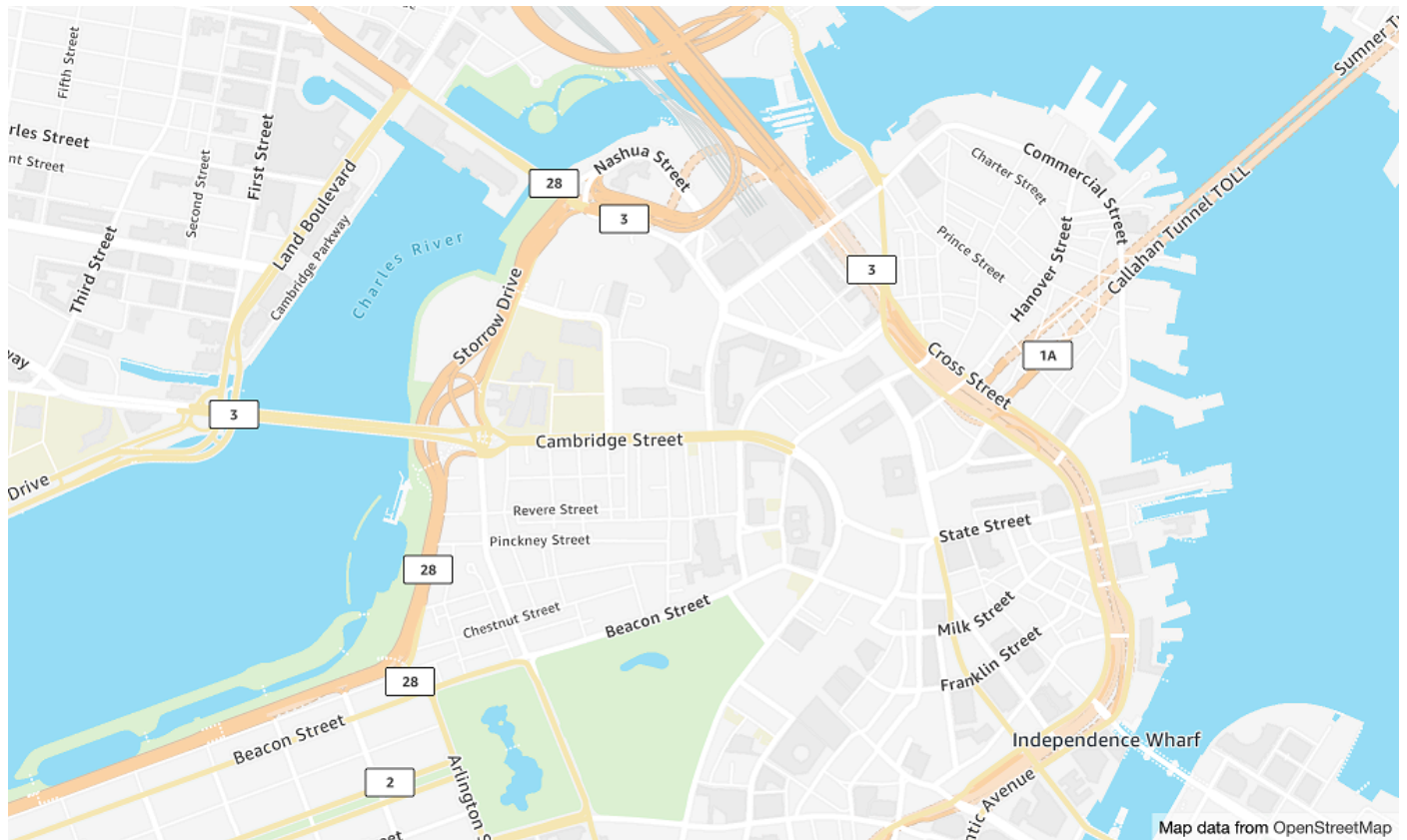
Estilos de mapas de datos abiertos

Amazon Location Service admite los siguientes estilos de mapa de HERE al [crear un recurso de mapa](#):

Los estilos de mapa de Open Data admiten estilos [Opiniones políticas](#) alternativos.

Open Data Standard Light

Open Data Standard Light



Nombre del estilo de mapa: VectorOpenDataStandardLight

Esto proporciona un mapa de base detallado del mundo en un estilo de mapa ligero, adecuado para utilizar en sitios web y aplicaciones móviles. El mapa incluye autopistas y carreteras principales, carreteras secundarias, ferrocarriles, fuentes de agua, ciudades, parques, monumentos, plantas de edificios y fronteras administrativas.


Este mapa base se basa en la [distribución de mapas diurnos](#) de OSM recopilada por colaboradores de (OSM). OpenStreetMap La comunidad de OSM incluye a más de 1,8 millones de colaboradores que actualizan más de 500.000 funciones a diario. Amazon Location Service incorpora estas modificaciones de forma periódica.

Fonts

Amazon Location sirve fuentes que utilizan [GetMapGlyphs](#). Las siguientes son las pilas de fuentes disponibles para este mapa:

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

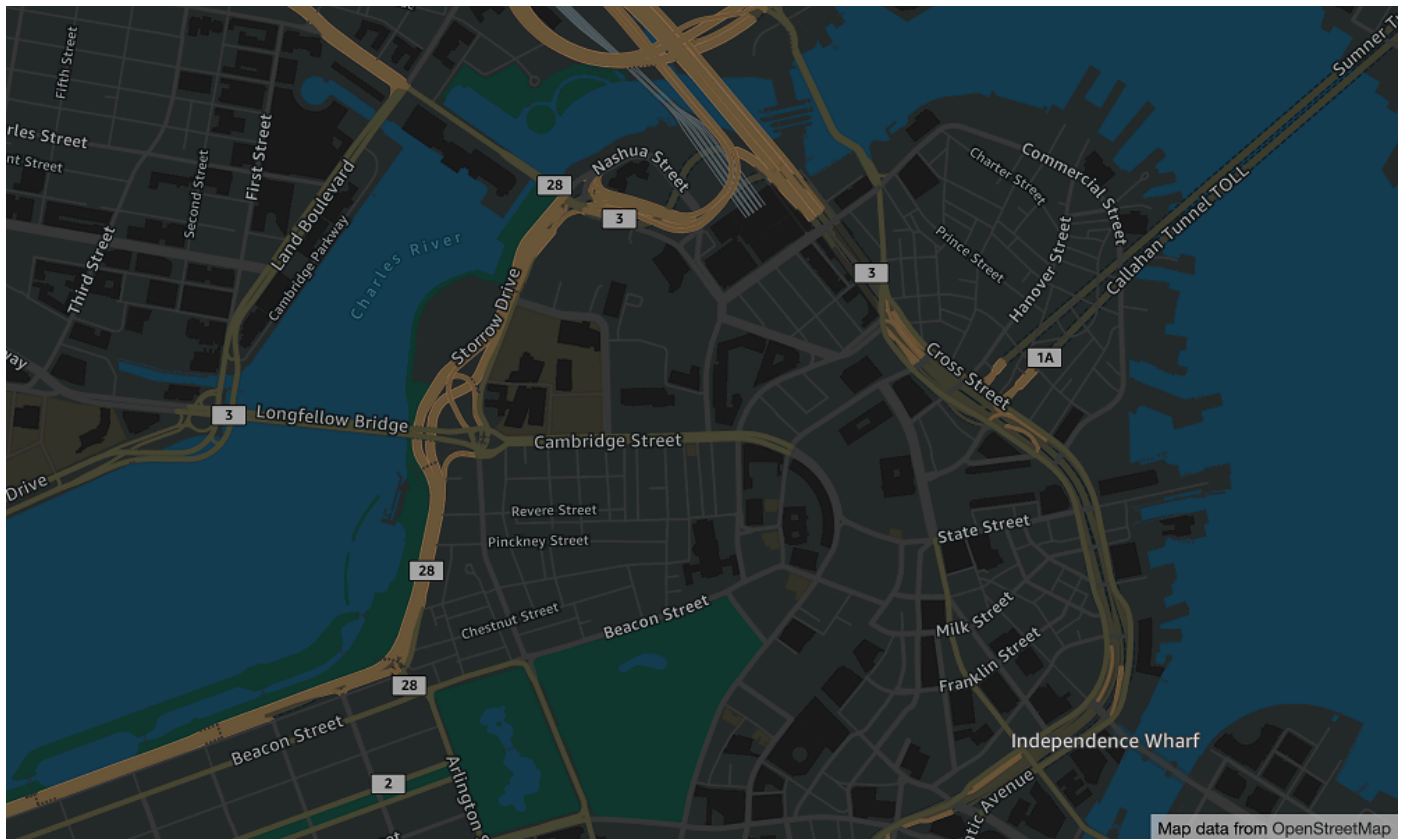
- Amazon Ember Medium, Noto Sans Medium
- Amazon Ember Regular Italic, Noto Sans Italic
- Amazon Ember Regular, Noto Sans Regular
- Amazon Ember Regular, Noto Sans Regular, Noto Sans Arabic Regular
- Amazon Ember Condensed RC Bold, Noto Sans Bold, Noto Sans Arabic Condensed Bold
- Amazon Ember Bold, Noto Sans Bold, Noto Sans Arabic Bold
- Amazon Ember Regular Italic, Noto Sans Italic, Noto Sans Arabic Regular
- Amazon Ember Condensed RC Regular, Noto Sans Regular, Noto Sans Arabic Condensed Regular
- Amazon Ember Medium, Noto Sans Medium, Noto Sans Arabic Medium

 Note

Las fuentes utilizadas por VectorOpenDataStandardLight son fuentes combinadas que utilizan Amazon Ember para la mayoría de los archivos Glyph, pero Noto Sans para glyphs no son compatibles con Amazon Ember.

Open Data Standard Dark

Open Data Standard Dark



Nombre del estilo de mapa: `VectorOpenDataStandardDark`

Se trata de un estilo de mapa de temática oscura que brinda un mapa de base detallado del mundo adecuado para utilizar en sitios web y aplicaciones móviles. El mapa incluye autopistas y carreteras principales, carreteras secundarias, ferrocarriles, fuentes de agua, ciudades, parques, monumentos, plantas de edificios y fronteras administrativas.


Este mapa base se basa en la [distribución de mapas diurnos](#) de OSM recopilada por colaboradores de (OSM). OpenStreetMap La comunidad de OSM incluye a más de 1,8 millones de colaboradores que actualizan más de 500.000 funciones a diario. Amazon Location Service incorpora estas modificaciones de forma periódica.

Fonts

Amazon Location sirve fuentes que utilizan [GetMapGlyphs](#). Las siguientes son las pilas de fuentes disponibles para este mapa:

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

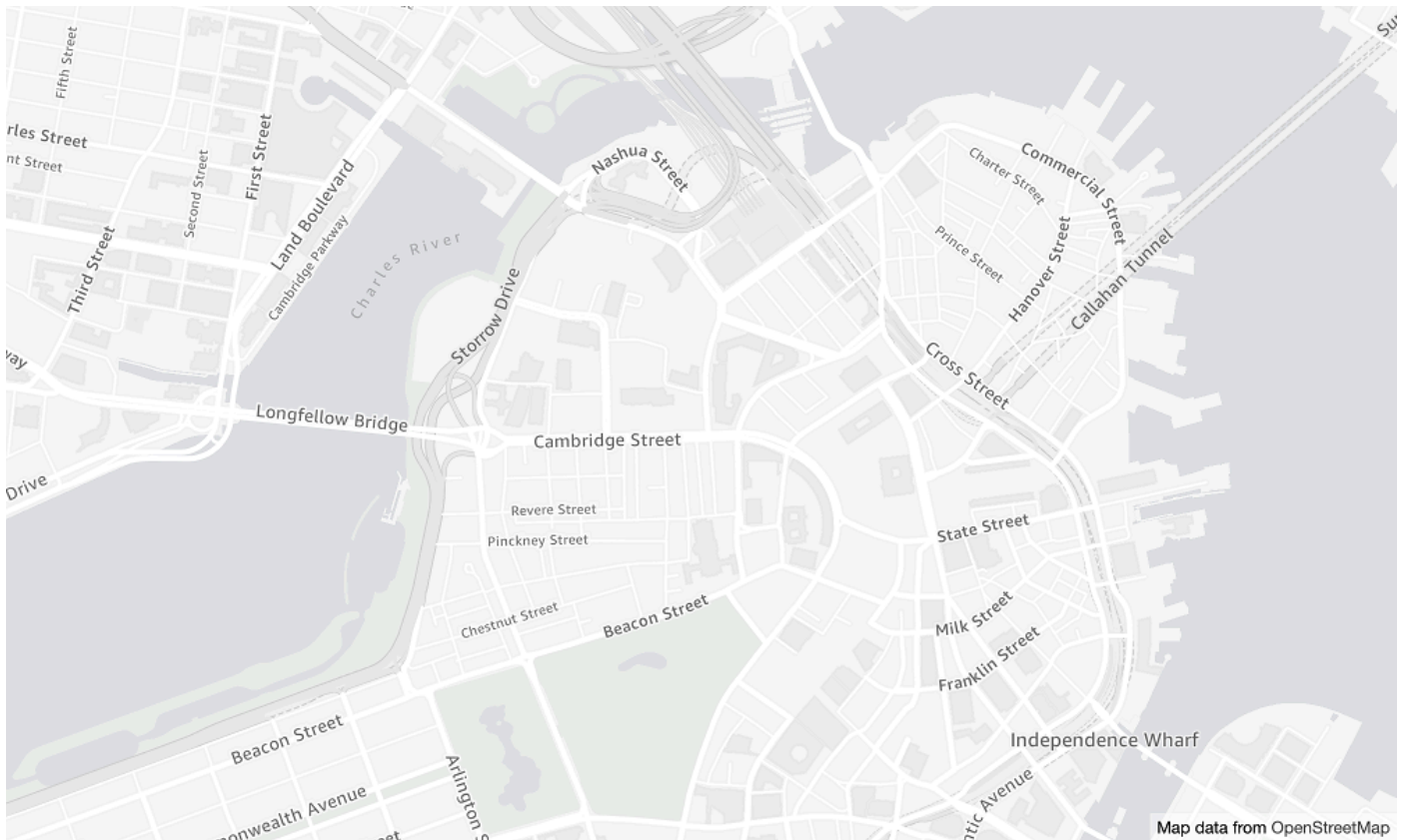
- Amazon Ember Medium, Noto Sans Medium
- Amazon Ember Regular Italic, Noto Sans Italic
- Amazon Ember Regular, Noto Sans Regular
- Amazon Ember Regular, Noto Sans Regular, Noto Sans Arabic Regular
- Amazon Ember Condensed RC Bold, Noto Sans Bold, Noto Sans Arabic Condensed Bold
- Amazon Ember Bold, Noto Sans Bold, Noto Sans Arabic Bold
- Amazon Ember Regular Italic, Noto Sans Italic, Noto Sans Arabic Regular
- Amazon Ember Condensed RC Regular, Noto Sans Regular, Noto Sans Arabic Condensed Regular
- Amazon Ember Medium, Noto Sans Medium, Noto Sans Arabic Medium

 Note

Las fuentes utilizadas por VectorOpenDataStandardDark son fuentes combinadas que utilizan Amazon Ember para la mayoría de los archivos Glyph, pero Noto Sans para glyphs no son compatibles con Amazon Ember.

Open Data Visualization Light

Open Data Visualization Light



Nombre del estilo de mapa: `VectorOpenDataVisualizationLight`

Se trata de un estilo de temática clara con colores apagados y menos elementos que ayudan a comprender los datos superpuestos.


Este mapa base se basa en la [distribución de mapas diurnos](#) de OSM recopilada por colaboradores de (OSM). OpenStreetMap La comunidad de OSM incluye a más de 1,8 millones de colaboradores que actualizan más de 500.000 funciones a diario. Amazon Location Service incorpora estas modificaciones de forma periódica.

Fonts

Amazon Location sirve fuentes que utilizan [GetMapGlyphs](#). Las siguientes son las pilas de fuentes disponibles para este mapa:

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

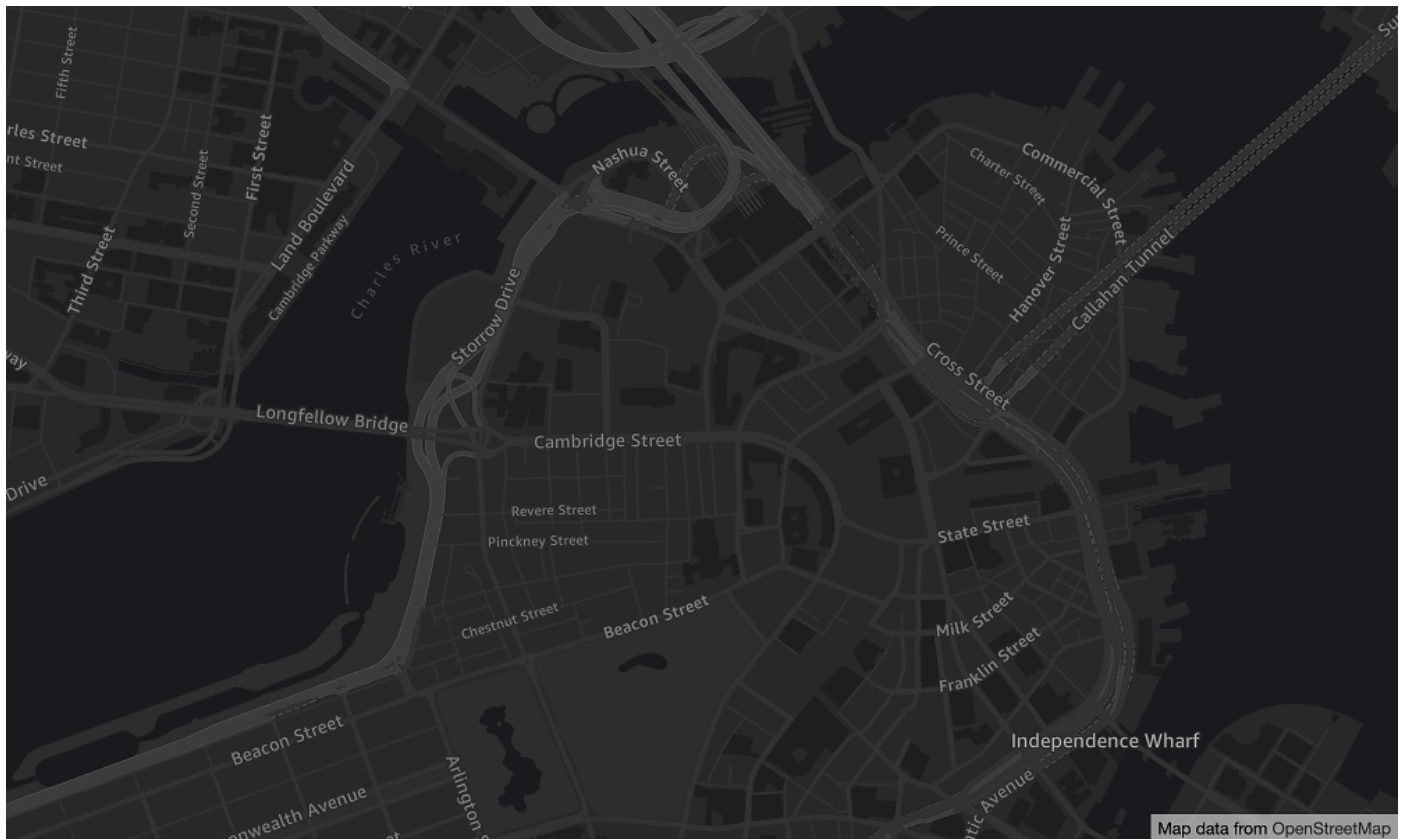
- Amazon Ember Regular Italic, Noto Sans Italic
- Amazon Ember Regular, Noto Sans Regular
- Amazon Ember Regular, Noto Sans Regular, Noto Sans Arabic Regular
- Amazon Ember Condensed RC Bold, Noto Sans Bold, Noto Sans Arabic Condensed Bold
- Amazon Ember Bold, Noto Sans Bold, Noto Sans Arabic Bold
- Amazon Ember Regular Italic, Noto Sans Italic, Noto Sans Arabic Regular
- Amazon Ember Condensed RC Regular, Noto Sans Regular, Noto Sans Arabic Condensed Regular
- Amazon Ember Medium, Noto Sans Medium, Noto Sans Arabic Medium

 Note

Las fuentes utilizadas por `VectorOpenDataVisualizationLight` son fuentes combinadas que utilizan Amazon Ember para la mayoría de los archivos Glyph, pero Noto Sans para glyphs no son compatibles con Amazon Ember.

Open Data Visualization Dark

Open Data Visualization Dark



Nombre del estilo de mapa: `VectorOpenDataVisualizationDark`

Se trata de un estilo de temática oscura con colores apagados y menos funciones que ayudan a comprender los datos superpuestos.

Este mapa base se basa en la [distribución de mapas diurnos](#) de OSM recopilada por colaboradores de (OSM). OpenStreetMap La comunidad de OSM incluye a más de 1,8 millones de colaboradores que actualizan más de 500.000 funciones a diario. Amazon Location Service incorpora estas modificaciones de forma periódica.

Fonts

Amazon Location sirve fuentes que utilizan [GetMapGlyphs](#). Las siguientes son las pilas de fuentes disponibles para este mapa:

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

- Amazon Ember Regular Italic, Noto Sans Italic
- Amazon Ember Regular, Noto Sans Regular
- Amazon Ember Regular, Noto Sans Regular, Noto Sans Arabic Regular
- Amazon Ember Condensed 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 Condensed RC Regular, Noto Sans Regular, Noto Sans Arabic Condensed Regular
- Amazon Ember Medium, Noto Sans Medium, Noto Sans Arabic Medium

Note

Las fuentes utilizadas por `VectorOpenDataVisualizationDark` son fuentes combinadas que utilizan Amazon Ember para la mayoría de los archivos Glyph, pero Noto Sans para glyphs no son compatibles con Amazon Ember.

Cobertura: Open Data

Open Data incluye mapas con cobertura global para renderizarlos con un [recurso de mapas de Amazon Location Service](#).

Note

Open Data se utiliza únicamente con los recursos de mapas de Amazon Location Service. No puede usar Open Data como proveedor de datos para admitir consultas de geocodificación, geocodificación inversa y búsquedas, ni para admitir consultas para calcular una ruta.

Condiciones de uso y atribución de datos: Open Data

Antes de usar Open Data, asegúrate de cumplir con todos los requisitos legales aplicables, incluidos los términos de licencia aplicables a Open Data y AWS.

Para obtener más información sobre los AWS requisitos, consulte [las condiciones de servicio de AWS](#).

Para obtener información sobre las directrices de atribución de datos abiertos, consulte las directrices sobre [derechos OpenStreetMap de autor y licencia](#) y OpenStreetMap atribución.

Informar de errores y contribuir a los datos abiertos

OpenStreetMap (OSM) y Natural Earth son proyectos de datos abiertos impulsados por la comunidad. Si tiene algún problema con los datos, puede denunciar los errores o aportar directamente soluciones o sugerencias.

- Para informar de un error u ofrecer una sugerencia en OSM, puede crear una nota en el mapa. Se trata de un comentario en el mapa que ayuda a los colaboradores a corregir el mapa. [Las notas se crean a través del sitio web. OpenStreetMap](#) Para obtener más información sobre las notas, consulte [Notas](#) en la OpenStreetMap wiki.
- Para obtener más información sobre cómo contribuir directamente a OpenStreetMap, incluida la adición de ubicaciones y la corrección de errores, consulte [Contribuir con datos de mapas](#) en la OpenStreetMap wiki.
- Para enviar una solicitud de corrección de los datos de Natural Earth, puede enviar un problema a través del sitio web de [Natural Earth](#).

Note

La corrección de los errores OpenStreetMap puede producirse rápidamente, sin embargo, las correcciones pueden tardar un tiempo en aparecer en la distribución cartográfica diurna de los datos de OSM que utiliza el proveedor de datos abiertos. El sitio web de [Daylight Map Distribution](#) proporciona más información sobre el proceso. Además, Amazon Location Service actualiza los datos de mapas utilizados en Amazon Location Service aproximadamente una vez al mes.

Características por proveedor de datos

En esta sección se describen las funciones disponibles en Amazon Location Service, clasificadas por proveedor de datos.

En la siguiente tabla, se muestra información general de alto nivel sobre las características.

Proveedores de datos	Cobertura geográfica	Cobertura de características	Región de AWS
Esri	Global	Mapas, lugares, rutas	Todas las regiones en las que Amazon Location está disponible.
Grab	Sudeste de Asia	Mapas, lugares, rutas	Asia Pacífico (Singapur), ap-southeast-1 , solamente.
AQUÍ	Global	Mapas, lugares, rutas	Todas las regiones en las que Amazon Location está disponible.
Abrir datos	Global	Mapas	Todas las regiones en las que Amazon Location está disponible.

En las siguientes pestañas se muestran los detalles de cada área de características.

Map Features

En la siguiente tabla, se muestran las características del mapa por proveedor de datos. Para obtener más información sobre conceptos de mapas, consulte [Mapas](#).

Proveedores de datos	Tipos de nodos compatibles	Niveles de zoom vectorial	Niveles de zoom rasterizado
Esri	Vector	0-15	0-23
	Ráster (imágenes)		

Proveedores de datos	Tipos de nodos compatibles	Niveles de zoom vectorial	Niveles de zoom rasterizado
	Para obtener más información, consulte Estilos de mapa de Esri .		
Grab	Vector (solo en el sudeste asiático) Para obtener más información, consulte Estilos de mapas de Grab .	0-14	Ninguno
AQUÍ	Vector Ráster (imágenes) Híbrido Para obtener más información, consulte Estilos de mapas de HERE .	1-17	0-19
Abrir datos	Vector Para obtener más información, consulte Estilos de mapas de datos abiertos .	0-15	Ninguno

Note

Los niveles de zoom representan la configuración máxima y mínima, tal como se define en las API de cada proveedor. Las distintas áreas del mapa pueden tener distintos máximos; por ejemplo, los mosaicos oceánicos pueden tener menos niveles de zoom detallados que las áreas de las principales ciudades.

MapLibre (y otros motores de representación de mapas) te permiten establecer niveles de zoom mínimos y máximos, y también respetan los niveles de zoom del proveedor de datos en un área, por lo que no tienes que escribir código para gestionar estas discrepancias.

Places and Search

En la siguiente tabla, se muestran las características del mapa por proveedor de datos. Para obtener más información sobre conceptos de lugares, consulte [Búsqueda de lugares](#).

Proveedores de datos	Geocodificación	Geocodificación inversa	Autocompletar	GetPlace
Esri	Todas las características, excepto: PlaceId	Todas las características, excepto: TimeZone PlaceId	Todas las características	Todas las características
Grab	Todas las características, excepto: unidad/tipo No se admiten categorías	Todas las características	Todas las características	Todas las características, excepto: Unidad/tipo SubMunicipality

Proveedores de datos	Geocodificación	Geocodificación inversa	Autocompletar	GetPlace
AQUÍ	Todas las características, excepto: Unidad: número Unidad/tipo relevancia Limitaciones adicionales del filtrado	Todas las características	Todas las características	Todas las características, excepto: Unidad: número Unidad/tipo SubMunicipality
Abrir datos	No admitido	No admitido	No admitido	Solo compatible con lo siguiente: SubMunicipality

Route features

En la siguiente tabla, se muestran las características de ruta por proveedor de datos. Para obtener más información sobre conceptos de ruta, consulte [Rutas](#). Para obtener descripciones más detalladas de las limitaciones de la matriz de rutas, consulte [Restricciones en las posiciones de salida y destino](#).

Proveedores de datos	Modo de viaje	Calcule una ruta	Matriz de rutas
Esri	Automóvil, camión, caminar	La salida y el destino deben estar a menos de 400 km uno del otro. El tiempo total	Hasta 10 posiciones de salida y de destino.

Proveedores de datos	Modo de viaje	Calcule una ruta	Matriz de rutas
		<p>de viaje no puede ser superior a 400 minutos.</p> <p>ArrivalTime no es compatible.</p>	<p>No se admite en Corea.</p> <p>Los pares de salida y destino deben estar a menos de 400 km uno del otro.</p>
Grab	<p>Automóvil, motocicleta.</p> <p>Caminar y andar en bicicleta en ciudades seleccionadas.</p>	Sin límites de distancia.	Hasta 350 posiciones de salida y de destino.
AQUÍ	Automóvil, camión, caminar	Sin límite de distancia. No se calcularán las rutas que se alejen más de 10 km del círculo alrededor de las posiciones de salida y destino.	<p>Hasta 350 posiciones de salida y de destino.</p> <p>Todas las posiciones de salida y destino deben estar situadas en un círculo de 180 km.</p> <p>Se admiten rutas más largas, con restricciones adicionales.</p>
Abrir datos	No admitido	No admitido	No admitido

Condiciones de uso y atribución de datos para los proveedores de datos

Antes de utilizar un proveedor de datos, asegúrese de cumplir con todos los requisitos legales aplicables, incluidos los términos de licencia aplicables al uso del proveedor.

Para obtener más información sobre los AWS requisitos, consulte [las condiciones de servicio de AWS](#).

Cuando utilice un proveedor de datos con sus recursos de Amazon Location para su solicitud o documentación, asegúrese de proporcionar las atribuciones de cada proveedor de datos que utilice.

Para obtener más información sobre el cumplimiento y la atribución de cada proveedor de datos, consulte los siguientes temas.

- Esri: [Condiciones de uso y atribución de datos: Esri](#)
- Grab: [Condiciones de uso y atribución de datos: Grab](#)
- HERE: [Condiciones de uso y atribución de datos: HERE](#)
- Datos abiertos: [Condiciones de uso y atribución de datos: Open Data](#)

Regiones y puntos de conexión de Amazon


La ubicación de Amazon está disponible en las siguientes AWS regiones:

Regiones

Nombre de la región	Región	Punto de conexión	Protocolo
Este de EE. UU. (Ohio)	us-east-2	geo.us-east-2.amazonaws.com	HTTPS
Este de EE. UU. (Norte de Virginia)	us-east-1	geo.us-east-1.amazonaws.com	HTTPS

Nombre de la región	Región	Punto de conexión	Protocolo
Oeste de EE. UU. (Oregón)	us-west-2	geo.us-west-2.amazonaws.com	HTTPS
Asia-Pacífico (Bombay)	ap-south-1	geo.ap-south-1.amazonaws.com	HTTPS
Asia-Pacífico (Singapur)	ap-southeast-1	geo.ap-southeast-1.amazonaws.com	HTTPS
Asia-Pacífico (Sídney)	ap-southeast-2	geo.ap-southeast-2.amazonaws.com	HTTPS
Asia-Pacífico (Tokio)	ap-northeast-1	geo.ap-northeast-1.amazonaws.com	HTTPS
Canadá (centro)	ca-central-1	geo.ca-central-1.amazonaws.com	HTTPS
Europa (Fráncfort)	eu-central-1	geo.eu-central-1.amazonaws.com	HTTPS
Europa (Irlanda)	eu-west-1	geo.eu-west-1.amazonaws.com	HTTPS
Europa (Londres)	eu-west-2	geo.eu-west-2.amazonaws.com	HTTPS

Nombre de la región	Región	Punto de conexión	Protocolo
Europa (Estocolmo)	eu-north-1	geo.eu-north-1.amazonaws.com	HTTPS
América del Sur (São Paulo)	sa-east-1	geo.sa-east-1.amazonaws.com	HTTPS
AWS GovCloud (EE. UU.-Oeste)	us-gov-west-1	geo.us-gov-west-1.amazonaws.com	HTTPS
		geo-fips.us-gov-west-1.amazonaws.com	HTTPS

 Note

Para obtener más información acerca de cómo utilizar los puntos de conexión de esta tabla, consulte la siguiente sección.

puntos de conexión

La sintaxis general de un punto de conexión regional de Amazon Location es la siguiente:

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

Dentro de esta sintaxis, Amazon Location utiliza los siguientes códigos de servicio:

Servicio	Código de servicio
Mapas de Amazon Location	Mapas
Amazon Location Places	lugares

Servicio	Código de servicio
Geocercas de Amazon Location	geocercado
Amazon Location Trackers	seguimiento
Rutas de Amazon Location	Rutas

Por ejemplo, el punto de conexión regional de Mapas de Amazon Location Este de EE. UU. (Norte de Virginia) sería: <https://maps.geo.us-east-1.amazonaws.com>.

Puntos de conexión de operación de la API

La sintaxis de un punto de conexión del plano de control de Amazon Location Service es la siguiente:

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

Las acciones del plano de control de Amazon Location Service son:

Servicio	Punto de conexión	Operación de la API
Mapas de Amazon Location	https://cp.maps.geo. <i>region</i> .amazonaws.com	CreateMap
		DeleteMap
		DescribeMap
		ListMaps
		UpdateMap
Amazon Location Places	https://cp.places.geo. <i>region</i> .amazonaws.com	CreatePlaceIndex
		DeletePlaceIndex
		DescribePlaceIndex
		ListPlaceIndexes
		UpdatePlaceIndex

Servicio	Punto de conexión	Operación de la API
Geocercas de Amazon Location	https://cp.geofencing.geo.region.amazonaws.com	CreateGeofenceCollection DeleteGeofenceCollection DescribeGeofenceCollection ListGeofenceCollections UpdateGeofenceCollection
Amazon Location Trackers	https://cp.trackimg.geo.region.amazonaws.com	CreateTracker DeleteTracker DescribeTracker UpdateTracker ListTrackers AssociateTrackerConsumer DisassociateTrackerConsumer ListTrackerConsumers
Rutas de Amazon Location	https://cp.routes.geo.region.amazonaws.com	CreateRouteCalculator DeleteRouteCalculator DescribeRouteCalculator ListRouteCalculators UpdateRouteCalculator

Servicio	Punto de conexión	Operación de la API
Metadatos de ubicación de Amazon	https://cp.metadatas.geo.region.amazonaws.com	CreateKey DeleteKey DescribeKey ListKeys UpdateKey

Quotas de Amazon Location Service

En este tema se proporciona un resumen de los límites de tarifas y las cuotas de Amazon Location Service.

Note

Si requiere una cuota más alta, Puede utilizar la consola de Service Quotas para [solicitar aumentos de cuota](#) para las cuotas ajustables. Al solicitar un aumento de cuota, selecciona la región en la que deseas aumentarlo, ya que la mayoría de las cuotas son específicas de la AWS región.

Service Quotas es la cantidad máxima de recursos que puede tener por AWS cuenta y AWS región. Amazon Location Service deniega las solicitudes adicionales que superen la service quota.

Límites de tarifas (cuotas que comienzan con una tasa de...) son el número máximo de solicitudes por segundo, con una velocidad de ráfaga del 80 por ciento del límite en cualquier parte del segundo, definida para cada operación de la API. Amazon Location Service limita las solicitudes que superan el límite de tarifas de la operación.

Nombre	Valor predeterminado	Ajuste	Descripción
Recursos clave de API por cuenta	Cada región admitida: 500	No	El número máximo de recursos clave de API

Nombre	Valor predeterminado	Ajuste	Descripción
			(activos o caducados) que puede tener por cuenta.
Recursos de recolección de geocercas por cuenta	Cada región admitida: 1500	Sí	El número máximo de recursos de recolección de geocercas que puede crear por cuenta.
Geocercas por colección de geocercas	Cada región admitida: 50 000	No	El número máximo de geocercas que puede crear por recolección de geocercas.
Mapear recursos por cuenta	Cada región admitida: 40	Sí	Cantidad máxima de recursos de mapa que puede crear por cuenta.
Coloca los recursos del índice por cuenta	Cada región admitida: 40	Sí	El número máximo de grupos de recursos de Place Index que puede crear en esta cuenta.
Tasa de solicitudes de Associate TrackerConsumer API	Cada región admitida: 10 por segundo	Sí	El número máximo de AssociateTrackerConsumer solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.

Nombre	Valor predeterminado	Ajuste	Descripción
Tasa de solicitudes a BatchDeleteDevicePositionHistory la API	Cada región admitida: 50 por segundo	Sí	El número máximo de BatchDeleteDevicePositionHistory solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a BatchDeleteGeofence la API	Cada región admitida: 50 por segundo	Sí	El número máximo de BatchDeleteGeofence solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a BatchEvaluateGeofences la API	Cada región admitida: 50 por segundo	Sí	El número máximo de BatchEvaluateGeofences solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a BatchGetDevicePosition la API	Cada región admitida: 50 por segundo	Sí	El número máximo de BatchGetDevicePosition solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a BatchPutGeofence la API	Cada región admitida: 50 por segundo	Sí	El número máximo de BatchPutGeofence solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.

Nombre	Valor predeterminado	Ajuste	Descripción
Tasa de solicitudes a BatchUpdateDevicePosition la API	Cada región admitida: 50 por segundo	Sí	El número máximo de BatchUpdateDevicePosition solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a CalculateRoute la API	Cada región admitida: 10 por segundo	Sí	El número máximo de CalculateRoute solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a CalculateRouteMatrix la API	Cada región admitida: 5 por segundo	Sí	El número máximo de CalculateRouteMatrix solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a CreateGeofenceCollection la API	Cada región admitida: 10 por segundo	Sí	El número máximo de CreateGeofenceCollection solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a CreateKey la API	Cada región admitida: 10 por segundo	Sí	El número máximo de CreateKey solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.

Nombre	Valor predeterminado	Ajuste	Descripción
Tasa de solicitudes a CreateMap la API	Cada región admitida: 10 por segundo	Sí	El número máximo de CreateMap solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a CreatePlaceIndex la API	Cada región admitida: 10 por segundo	Sí	El número máximo de CreatePlaceIndex solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a CreateRouteCalculator la API	Cada región admitida: 10 por segundo	Sí	El número máximo de CreateRouteCalculator solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a CreateTracker la API	Cada región admitida: 10 por segundo	Sí	El número máximo de CreateTracker solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a DeleteGeofenceCollection la API	Cada región admitida: 10 por segundo	Sí	El número máximo de DeleteGeofenceCollection solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.

Nombre	Valor predeterminado	Ajuste	Descripción
Tasa de solicitudes a DeleteKey la API	Cada región admitida: 10 por segundo	Sí	El número máximo de DeleteKey solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a DeleteMap la API	Cada región admitida: 10 por segundo	Sí	El número máximo de DeleteMap solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a DeletePlaceIndex la API	Cada región admitida: 10 por segundo	Sí	El número máximo de DeletePlaceIndex solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a DeleteRouteCalculator la API	Cada región admitida: 10 por segundo	Sí	El número máximo de DeleteRouteCalculator solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a DeleteTracker la API	Cada región admitida: 10 por segundo	Sí	El número máximo de DeleteTracker solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.

Nombre	Valor predeterminado	Ajuste	Descripción
Tasa de solicitudes a DescribeGeofenceCollection la API	Cada región admitida: 10 por segundo	<u>Sí</u>	El número máximo de DescribeGeofenceCollection solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a DescribeKey la API	Cada región admitida: 10 por segundo	<u>Sí</u>	El número máximo de DescribeKey solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a DescribeMap la API	Cada región admitida: 10 por segundo	<u>Sí</u>	El número máximo de DescribeMap solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a DescribePlaceIndex la API	Cada región admitida: 10 por segundo	<u>Sí</u>	El número máximo de DescribePlaceIndex solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a DescribeRouteCalculator la API	Cada región admitida: 10 por segundo	<u>Sí</u>	El número máximo de DescribeRouteCalculator solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.

Nombre	Valor predeterminado	Ajuste	Descripción
Tasa de solicitudes a DescribeTracker la API	Cada región admitida: 10 por segundo	Sí	El número máximo de DescribeTracker solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a DisassociateTrackerConsumer la API	Cada región admitida: 10 por segundo	Sí	El número máximo de DisassociateTrackerConsumer solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a ForecastGeofenceEvents la API	Cada región admitida: 50 por segundo	Sí	El número máximo de ForecastGeofenceEvents solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a GetDevicePosition la API	Cada región admitida: 50 por segundo	Sí	El número máximo de GetDevicePosition solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a GetDevicePositionHistory la API	Cada región admitida: 50 por segundo	Sí	El número máximo de GetDevicePositionHistory solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.

Nombre	Valor predeterminado	Ajuste	Descripción
Tasa de solicitudes a GetGeofence la API	Cada región admitida: 50 por segundo	Sí	El número máximo de GetGeofence solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a GetMapGlyphs la API	Cada región admitida: 50 por segundo	Sí	El número máximo de GetMapGlyphs solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a GetMapSprites la API	Cada región admitida: 50 por segundo	Sí	El número máximo de GetMapSprites solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a GetMapStyleDescriptor la API	Cada región admitida: 50 por segundo	Sí	El número máximo de GetMapStyleDescriptor solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a GetMapTile la API	Cada región admitida: 500 por segundo	Sí	El número máximo de GetMapTile solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.

Nombre	Valor predeterminado	Ajuste	Descripción
Tasa de solicitudes a GetPlace la API	Cada región admitida: 50 por segundo	<u>Sí</u>	El número máximo de GetPlace solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a ListDevicePositions la API	Cada región admitida: 50 por segundo	<u>Sí</u>	El número máximo de ListDevicePositions solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a ListGeofenceCollections la API	Cada región admitida: 10 por segundo	<u>Sí</u>	El número máximo de ListGeofenceCollections solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a ListGeofences la API	Cada región admitida: 50 por segundo	<u>Sí</u>	El número máximo de ListGeofences solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a ListKeys la API	Cada región admitida: 10 por segundo	<u>Sí</u>	El número máximo de ListKeys solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.

Nombre	Valor predeterminado	Ajuste	Descripción
Tasa de solicitudes a ListMaps la API	Cada región admitida: 10 por segundo	Sí	El número máximo de ListMaps solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a ListPlaceIndexes la API	Cada región admitida: 10 por segundo	Sí	El número máximo de ListPlaceIndexes solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a ListRouteCalculators la API	Cada región admitida: 10 por segundo	Sí	El número máximo de ListRouteCalculators solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a ListTagsForResource la API	Cada región admitida: 10 por segundo	Sí	El número máximo de ListTagsForResource solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a ListTrackerConsumers la API	Cada región admitida: 10 por segundo	Sí	El número máximo de ListTrackerConsumers solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.

Nombre	Valor predeterminado	Ajuste	Descripción
Tasa de solicitudes a ListTrackers la API	Cada región admitida: 10 por segundo	Sí	El número máximo de ListTrackers solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a PutGeofence la API	Cada región admitida: 50 por segundo	Sí	El número máximo de PutGeofence solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a SearchPlaceIndexForPosition la API	Cada región admitida: 50 por segundo	Sí	El número máximo de SearchPlaceIndexForPosition solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a SearchPlaceIndexForSuggestions la API	Cada región admitida: 50 por segundo	Sí	El número máximo de SearchPlaceIndexForSuggestions solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a SearchPlaceIndexForText la API	Cada región admitida: 50 por segundo	Sí	El número máximo de SearchPlaceIndexForText solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.

Nombre	Valor predeterminado	Ajuste	Descripción
Tasa de solicitudes a TagResource la API	Cada región admitida: 10 por segundo	Sí	El número máximo de TagResource solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a UntagResource la API	Cada región admitida: 10 por segundo	Sí	El número máximo de UntagResource solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a UpdateGeofenceCollection la API	Cada región admitida: 10 por segundo	Sí	El número máximo de UpdateGeofenceCollection solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a UpdateKey la API	Cada región admitida: 10 por segundo	Sí	El número máximo de UpdateKey solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a UpdateMap la API	Cada región admitida: 10 por segundo	Sí	El número máximo de UpdateMap solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.

Nombre	Valor predeterminado	Ajuste	Descripción
Tasa de solicitudes a UpdatePlaceIndex la API	Cada región admitida: 10 por segundo	Sí	El número máximo de UpdatePlaceIndex solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a UpdateRouteCalculator la API	Cada región admitida: 10 por segundo	Sí	El número máximo de UpdateRouteCalculator solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a UpdateTracker la API	Cada región admitida: 10 por segundo	Sí	El número máximo de UpdateTracker solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Tasa de solicitudes a VerifyDevicePosition la API	Cada región admitida: 50 por segundo	Sí	El número máximo de VerifyDevicePosition solicitudes que puedes realizar por segundo. Las solicitudes adicionales se limitan.
Recursos de la calculadora de rutas por cuenta	Cada región admitida: 40	Sí	El número máximo de recursos de Calculadora de ruta que puede crear por cuenta.

Nombre	Valor predeterminado	Ajuste	Descripción
Rastrea a los consumidores por rastreador	Cada región admitida: 5	No	El número máximo de colecciones de geocercas con el que se puede asociar el recurso rastreador.
Recursos de seguimiento por cuenta	Cada región admitida: 500	<u>Sí</u>	El número máximo de grupos de recursos de rastreador que puede crear en esta cuenta.

Note

Puede monitorear el uso según sus cuotas con Cloudwatch. Para obtener más información, consulte [Se usa CloudWatch para monitorear el uso comparándolo con las cuotas.](#)

Administración de las cuotas de Amazon Location Service

Amazon Location Service está integrado con Service Quotas, un AWS servicio que te permite ver y gestionar tus cuotas desde una ubicación central. Para obtener más información, consulte [¿Qué son las Service Quotas?](#) en la Guía del usuario de Service Quotas.

Con Service Quotas, resulta más sencillo buscar el valor de las Service Quotas de Amazon Location Service.

AWS Management Console

Ver las service quotas de Amazon Location Service mediante la consola

1. Abra la consola de Service Quotas en <https://console.aws.amazon.com/servicequotas/>.
2. En el panel de navegación, elija Servicios de AWS .
3. En la lista de servicios de AWS , busque y seleccione Amazon Location.

En la lista Service Quotas, puede ver el nombre de la Service Quota, el valor aplicado (si está disponible), la cuota predeterminada de AWS y si el valor de cuota es ajustable.

4. Para ver información adicional sobre una cuota de servicio, como, por ejemplo, la descripción, elija el nombre de cuota.
5. (Opcional) Para solicitar un aumento de cuota, seleccione la cuota que desea aumentar, seleccione Solicitar aumento de cuota, escriba o seleccione la información necesaria y seleccione Solicitar.

Para trabajar más con Service Quotas mediante la consola, consulte la [Guía del usuario de Service Quotas](#). Para solicitar un aumento de cuota, consulte [Solicitud de un aumento de cuota](#) en la Guía de usuario de Service Quotas.

AWS CLI

Ver las cuotas de Amazon Location Service mediante AWS CLI

Ejecute el siguiente comando para consultar las cuotas predeterminadas de Amazon Location.

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

Para trabajar más con las cuotas de servicio mediante el AWS CLI, consulte la [Referencia de AWS CLI comandos de Service Quotas](#). Para solicitar un aumento de cuota, consulte el [request-service-quota-increase](#) comando en la [Referencia de comandos de la AWS CLI](#).

Cómo empezar como desarrollador con Amazon Location Service

Puede usar Amazon Location Service para proporcionar funciones relacionadas con la geografía para aplicaciones en muchos sistemas y factores de forma diferentes, incluidos los servicios web de backend, las aplicaciones web y las aplicaciones móviles. Se proporcionan muchas herramientas para ayudarle a crear sus aplicaciones, incluidos los SDK, las librerías y el código de muestra.

En esta sección se proporcionan información y enlaces que le ayudarán a comenzar a utilizar Amazon Location. En concreto, en los siguientes temas se proporciona la información que puede resultarle más útil:

- [Escenarios y casos de uso](#): una lista de escenarios de desarrollo y cómo Amazon Location Service puede ayudarle a completarlos.
- [SDK y herramientas de Amazon Location](#): los kits de desarrollo de software (SDK) y librerías que le ayudarán a programar con Amazon Location.
- Referencia de las [API de Amazon Location Service: referencia](#) a las principales API de Amazon Location que se incluyen con el AWS SDK.
- [Ejemplos de código](#): en esta sección se proporcionan ejemplos que le ayudarán a empezar o a agregar funcionalidades a su aplicación actual.
- [Tutorial de inicio rápido](#): en este tutorial se muestra cómo crear su primera aplicación. Existen versiones del tutorial para crear una aplicación web o una aplicación móvil basada en Android.
- [Conceptos de Amazon Location Service](#): esta sección de esta guía describe los conceptos básicos de Amazon Location, incluidas las secciones sobre Mapas, Búsqueda de lugares, Rutas y Geovallas y Rastreadores.
- [Amplify](#): Amplify es una solución completa que encapsula gran parte de la funcionalidad necesaria para crear aplicaciones web y móviles utilizando Nube de AWS. Si ya utiliza Amplify o decide utilizar Amplify, esta aplicación cuenta con una librería geográfica integrada con Amazon Location Service que puede utilizar. Para empezar a utilizar Amplify Geo, consulte la documentación [aquí](#).

Escenarios y casos de uso

Amazon Location Service es un servicio que se ejecuta en Nube de AWS. Puede llamarlo desde sus propias instancias de Amazon EC2 en la nube, pero muchas aplicaciones de mapeo se ejecutan

en dispositivos o en una combinación de dispositivos y la nube. A continuación, se enumeran solo algunos escenarios típicos y cómo podría abordar su desarrollo.

- Una aplicación de backend que le ayuda a optimizar las rutas para los conductores de su flota.

Puede escribir una aplicación que se ejecute en [Amazon EC2](#) en la Nube de AWS que utilice Amazon Location Service para [calcular las matrices de rutas](#) como entrada para un optimizador de rutas para su flota. Utilice el [SDK de AWS](#) para realizar llamadas a Amazon Location.

- Una aplicación web que permita a sus clientes encontrar las ubicaciones de su empresa.

Puede crear un sitio web que se ejecute en instancias de Amazon EC2, incluida una aplicación basada en la ubicación. Utilice el [AWS SDK JavaScript para](#) desarrollar una aplicación web para buscar ubicaciones mediante la [búsqueda de lugares](#) y mostrar los resultados en un [mapa](#) mediante MapLibre. Utilice el SDK de Amazon Location para facilitar la programación con ubicación.

- Agregue funciones de ubicación a una aplicación de iOS o Android existente.

Puedes usar el AWS SDK para Swift (iOS) o [Kotlin](#) (Android) para realizar llamadas a Amazon Location y añadir funciones de [búsqueda de lugares](#) y [mapas](#) a tu aplicación. Se usa MapLibre para renderizar mapas. Hay [SDK de AWS](#) adicionales disponibles para otros lenguajes.

- Rastree los activos (dispositivos o vehículos) y reciba actualizaciones cuando se encuentren en las áreas de entrada o salida que usted defina.

Una aplicación para rastrear dispositivos consta de varias partes.

- Cada dispositivo que esté rastreando debe tener un recurso de [rastreador](#) creado para rastrearlo. Debe enviar actualizaciones de posición a Amazon Location Service, por ejemplo, mediante [MQTT](#).
- Cree [geovallas](#) para definir las áreas en las que desea que se registren los eventos de entrada y salida de sus activos.
- Puede usar [Amazon EC2](#) o [AWS Lambda](#) para responder a sus eventos cuando los activos entren o salgan de las áreas de geovalla.
- Puede ampliar esta opción para crear aplicaciones web o para dispositivos que le permitan rastrear y mostrar las ubicaciones de sus activos en los mapas.

En la siguiente sección se proporcionan detalles sobre las herramientas y librerías disponibles para su uso en cada aspecto de Amazon Location Service.

SDK y herramientas para usar Amazon Location Service

Existen varias herramientas que le ayudarán a utilizar Amazon Location Service.

- **AWS SDK:** los kits de desarrollo de AWS software (SDK) están disponibles en muchos lenguajes de programación populares y proporcionan una API, ejemplos de código y documentación que facilitan la creación de aplicaciones en el idioma de su preferencia. Los AWS SDK incluyen las principales API y funciones de Amazon Location, incluido el acceso a Maps, Places search, Routes, Geofence y Trackers. Para obtener más información sobre los SDK disponibles para su uso con Amazon Location Service para diferentes aplicaciones y lenguajes, consulte [SDK por lenguaje](#).
- **MapLibre**— Amazon Location Service recomienda renderizar los mapas con el motor de [MapLibre](#) renderizado. MapLibre es un motor para mostrar mapas en aplicaciones web o móviles. MapLibre también tiene un modelo de complemento y es compatible con la interfaz de usuario para la búsqueda y las rutas en algunos idiomas y plataformas. Para obtener más información sobre su uso MapLibre y las funciones que proporciona, consulte [MapLibre](#).
- **Amazon Location SDK:** Amazon Location SDK es un conjunto de librerías de código abierto que facilitan el desarrollo de aplicaciones con Amazon Location Service. Las bibliotecas proporcionan funciones para admitir la autenticación de aplicaciones móviles y web, el seguimiento de la ubicación para aplicaciones móviles, la conversión entre los tipos de datos de Amazon Location y [GeoJSON](#), así como un paquete alojado del cliente Amazon Location para el AWS SDK v3. Para obtener más información sobre Amazon Location SDK, consulte [Amazon Location SDK](#).

SDK por lenguaje

En las siguientes tablas se proporciona información sobre AWS los SDK y MapLibre las versiones para lenguajes y marcos, por tipo de aplicación: aplicación web, móvil o de fondo.

Versiones del SDK

Te recomendamos que utilices la versión más reciente del AWS SDK y cualquier otro SDK que utilices en tus proyectos y que mantengas los SDK actualizados. El AWS SDK te proporciona las funciones y funciones más recientes, así como actualizaciones de seguridad. Para encontrar la versión más reciente del AWS SDK JavaScript, por ejemplo, consulte el tema de [instalación del navegador](#) en el AWS SDK para ver la JavaScript documentación.

Web frontend

Los siguientes AWS SDK y MapLibre versiones están disponibles para el desarrollo de aplicaciones de interfaz web.

Lenguaje / marco	SDK de AWS	Marco de renderización
Totalmente compatible		
JavaScript	https://aws.amazon.com//sdk-for-javascript	https://maplibre.org/projects/maplibre-gl-js/
ReactJS	https://aws.amazon.com/sdk-for-javascript/	https://github.com/maplibre/maplibre-react-native
TypeScript	https://aws.amazon.com/sdk-for-javascript/	https://maplibre.org/projects/maplibre-gl-js/
Compatible parcialmente		
Flutter	https://docs.amplify.aws/start/q/integration/flutter/ Flutter aún no es totalmente compatible AWS, pero Amplify ofrece un soporte limitado.	https://github.com/maplibre/flutter-maplibre-gl La biblioteca MapLibre Flutter se considera experimental.
Node.js	https://aws.amazon.com//sdk-for-javascript	No hay MapLibre soporte para Node.js.
PHP	https://aws.amazon.com/sdk-for-php/	No hay MapLibre soporte para PHP.

Mobile frontend

Los siguientes AWS SDK y MapLibre versiones están disponibles para el desarrollo de aplicaciones frontend móviles.

Lenguaje / marco	AWS SDK	Marco de renderización
Totalmente compatible		
Java	https://aws.amazon.com/sdk-for-java/	https://maplibre.org/projects/maplibre-native/
Kotlin	<p>https://aws.amazon.com/sdk-for-kotlin/</p> <p>SDK de autenticación móvil de Amazon Location Service para Android: https://github.com/aws-geospatial/amazon-location-mobile-auth-sdk-android</p> <p>SDK de seguimiento móvil de Amazon Location Service para Android: https://github.com/aws-geospatial/amazon-location-mobile-tracking-sdk-android</p>	<p>https://maplibre.org/projects/maplibre-native/</p> <p>Requiere enlaces personalizados, ya que está basado en Java. MapLibre</p>
ObjectiveC	https://github.com/aws-amplify/aws-sdk-ios	https://maplibre.org/projects/maplibre-native/
ReactNative	https://aws.amazon.com/sdk-for-javascript/	https://github.com/maplibre/maplibre-react-native
Swift	<p>https://aws.amazon.com/sdk-for-swift/</p> <p>SDK de autenticación móvil de Amazon Location Service para iOS: https://github.com/aws-geospatial/amazon-location-mobile-auth-sdk-ios</p>	https://maplibre.org/projects/maplibre-native/

Lenguaje / marco	AWS SDK	Marco de renderización
	SDK de seguimiento móvil de Amazon Location Service para iOS: https://github.com/aws-geospatial/amazon-location-mobile-tracking-sdk-ios	
Compatible parcialmente		
Flutter	https://docs.amplify.aws/start/q/integration/flutter/ Flutter aún no es totalmente compatible con AWS, pero Amplify ofrece un soporte limitado.	https://github.com/maplibre/flutter-maplibre-gl La biblioteca MapLibre Flutter se considera experimental.

Backend application

Los siguientes AWS SDK están disponibles para el desarrollo de aplicaciones de back-end. MapLibre no aparece aquí porque la representación de mapas no suele ser necesaria para las aplicaciones de backend.

Idioma	SDK de AWS
.NET	https://aws.amazon.com/sdk-for-net
C++	https://aws.amazon.com/sdk-for-cpp/
Go	https://aws.amazon.com/sdk-for-go/
Java	https://aws.amazon.com/sdk-for-java/
JavaScript	https://aws.amazon.com/sdk-for-javascript/
Node.js	https://aws.amazon.com/sdk-for-javascript/

Idioma	SDK de AWS
TypeScript	https://aws.amazon.com/sdk-for-javascript/
Kotlin	https://aws.amazon.com/sdk-for-kotlin/
PHP	https://aws.amazon.com/sdk-for-php/
Python	https://aws.amazon.com/sdk-for-python/
Ruby	https://aws.amazon.com/sdk-for-ruby/
Rust	https://aws.amazon.com/sdk-for-rust/ El AWS SDK de Rust está en versión preliminar para desarrolladores.

Uso de MapLibre herramientas y bibliotecas con Amazon Location

Una de las herramientas importantes para crear aplicaciones interactivas con Amazon Location es MapLibre. [MapLibre](#) es principalmente un motor de renderizado para mostrar mapas en una aplicación web o móvil. Sin embargo, también incluye soporte para complementos y proporciona funciones para trabajar con otros aspectos de Amazon Location. A continuación se describen las herramientas que puede utilizar en función del área de ubicación con la que desee trabajar.

Note

Para utilizar cualquier aspecto de Amazon Location, instale el [SDK de AWS para el lenguaje que desee utilizar](#).

- Mapas

Para mostrar mapas en su aplicación, necesita un motor de representación de mapas que utilice los datos proporcionados por Amazon Location y los dibuje en la pantalla. Los motores de renderización de mapas también ofrecen la funcionalidad de desplazar y ampliar el mapa, o de agregar marcadores o chinchetas y otras anotaciones al mapa.

Amazon Location Service recomienda renderizar los mapas con el motor de [MapLibre](#) renderizado. MapLibre GL JS es un motor para mostrar mapas JavaScript, mientras que MapLibre Native proporciona mapas para iOS o Android.

MapLibre también tiene un ecosistema de complementos para ampliar la funcionalidad principal. Para obtener más información, visite <https://maplibre.org/maplibre-gl-js-docs/plugins/>.

- Búsqueda de lugares

[Para simplificar la creación de una interfaz de usuario de búsqueda, puede usar el MapLibre geocodificador para la web \(las aplicaciones de Android pueden usar el complemento Android Places\).](#)

Utilice la [biblioteca de geocodificadores Amazon Location for Maplibre](#) para simplificar el proceso de uso de Amazon Location with `amazon-location-for-maplibre-gl-geocoder` in Applications. JavaScript

- Rutas

[Para mostrar las rutas en el mapa, utilice las indicaciones. MapLibre](#)

- Geovallas y rastreadores

MapLibre no tiene ninguna representación ni herramientas específicas para las geovallas y el rastreo, pero puede usar la función de renderización y los [complementos](#) para mostrar las geocercas y los dispositivos rastreados en el mapa.

Los dispositivos que se están rastreando pueden usar [MQTT](#) o enviar actualizaciones manualmente a Amazon Location Service. Se puede responder a los eventos de Geofence utilizando [AWS Lambda](#).

Hay muchas librerías de código abierto disponibles para proporcionar funciones adicionales a Amazon Location Service, por ejemplo, [Turf](#), que proporciona funciones de análisis espacial.

Muchas librerías utilizan los datos con formato de estándar abierto [GeoJSON](#). Amazon Location Service proporciona una biblioteca para admitir el uso de GeoJSON en JavaScript las aplicaciones. Para obtener más información, consulte la siguiente sección, [Amazon Location SDK y librerías](#).

Complemento Amazon Location MapLibre Geocoder

El complemento Amazon Location MapLibre geocoder está diseñado para facilitar la incorporación de la funcionalidad Amazon Location en sus JavaScript aplicaciones cuando trabaja con la representación de mapas y la geocodificación mediante la biblioteca. [maplibre-gl-geocoder](#)

Instalación

Puede instalar el complemento Amazon Location MapLibre Geocoder desde NPM para usarlo con módulos, con este comando:

```
npm install @aws/amazon-location-for-maplibre-gl-geocoder
```

Puede importarlo a un archivo HTML para usarlo directamente en el navegador, con un script:

```
<script src="https://www.unpkg.com/@aws/amazon-location-for-maplibre-gl-geocoder@1"/>/script<
```

Uso con el módulo

Este código configura un JavaScript mapa GL de Maplibre con capacidades de geocodificación de Amazon Location. Utiliza la autenticación mediante Amazon Cognito Identity Pool para acceder a los recursos de Amazon Location. El mapa se representa con un estilo y unas coordenadas centrales específicos, y permite buscar lugares en el mapa.

```
// 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());
```

Uso con un navegador

En este ejemplo, se utiliza el Amazon Location Client para realizar una solicitud que se autentica mediante Amazon Cognito.

Note

Algunos de estos ejemplos utilizan Amazon Location Client. El Amazon Location Client se basa en el [AWS SDK de la JavaScript versión 3](#) y permite realizar llamadas a Amazon Location mediante un script al que se hace referencia en un archivo HTML.

Incluye lo siguiente en un archivo 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>
```

Incluya lo siguiente en un JavaScript archivo:

```
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());
```

A continuación se enumeran las funciones y los comandos que se utilizan en el complemento Amazon Location MapLibre Geocoder:

- **buildAmazonLocationMaplibreGeocoder**

Esta clase crea una instancia de la `AmazonLocationMaplibreGeocoder` que es el punto de entrada a las demás llamadas:

```
const amazonLocationMaplibreGeocoder = buildAmazonLocationMaplibreGeocoder(client,
  placesIndex, {enableAll: true});
```

• **getPlacesGeocoder**

Devuelve un objeto iControl listo para usar que se puede añadir directamente a un mapa.

```
const geocoder = getPlacesGeocoder();

// Initialize map
let map = await initializeMap();

// Add the geocoder to the map.
map.addControl(geocoder);
```

Amazon Location SDK y librerías

Amazon Location SDK es un conjunto de librerías de código abierto que proporcionan funciones útiles para desarrollar aplicaciones de Amazon Location. Se incluye la siguiente funcionalidad:

- Cliente Amazon Location: los objetos Amazon Location del AWS SDK v3 vienen agrupados y empaquetados para facilitar su uso en el desarrollo web.
- Autenticación: la utilidad de autenticación simplifica la autenticación (mediante Amazon Cognito o claves de API) al crear una página web o una [JavaScript](#) aplicación [iOS](#) o [Android](#) para Amazon Location Service.
- Seguimiento: los SDK de seguimiento móvil están disponibles para [iOS](#) y [Android](#). Este SDK facilita que las aplicaciones móviles interactúen con Amazon Location Trackers.
- Funciones GeoJSON de Amazon Location: las [utilidades de conversión de GeoJSON facilitan la conversión](#) entre los datos con [formato GeoJSON estándar del sector y los formatos](#) de la API de ubicación de Amazon.

Temas

- [Cómo empezar a utilizar Amazon Location SDK](#)
- [Cliente de Amazon Location](#)
- [JavaScript Ayudante de autenticación](#)
- [Ayudantes de conversión de GeoJSON](#)

- [SDK de autenticación móvil de Android](#)
- [SDK de autenticación móvil para iOS](#)
- [SDK de seguimiento para dispositivos móviles Android](#)
- [SDK de seguimiento móvil para iOS](#)

Cómo empezar a utilizar Amazon Location SDK

Amazon Location SDK es un conjunto de funciones que pueden simplificar el uso de Amazon Location Service en una aplicación. Puede instalar e importar estas funciones a su aplicación. JavaScript En las siguientes secciones se describe el cliente de Amazon Location y las librerías auxiliares de autenticación y GeoJSON.

Cliente de Amazon Location

Con AWS el SDK v3, el SDK está separado por servicio. Puede instalar solo las partes que necesite. Por ejemplo, para instalar el cliente de Amazon Location y el proveedor de credenciales de Amazon Cognito, utilice los siguientes comandos.

```
npm install @aws-sdk/client-location
npm install @aws-sdk/credential-providers
```

Para facilitar el uso de Amazon Location Service en las aplicaciones de interfaz JavaScript web, AWS proporciona un paquete hospedado de la biblioteca de ubicaciones de Amazon y el proveedor de credenciales. Para usar el cliente agrupado, agréguelo a su HTML en una etiqueta de script, de la siguiente manera:

```
<script src="https://unpkg.com/@aws/amazon-location-client@1.x/dist/
amazonLocationClient.js"></script>
```

Note

El paquete se mantiene actualizado y es compatible con versiones anteriores para facilitar su uso. El uso de esta etiqueta de script o la instalación de NPM siempre obtendrá la última versión.

JavaScript Ayudante de autenticación

El asistente de JavaScript autenticación de Amazon Location facilita la autenticación al realizar llamadas a la API de Amazon Location desde tu JavaScript aplicación. Este asistente de autenticación le ayuda específicamente a utilizar [Amazon Cognito](#) o [las claves de API](#) como método de autenticación. Se trata de una biblioteca de código abierto que está disponible en <https://github.com/aws-geospatial/GitHub-amazon-location-utilities-auth-helper-js>.

Note

La compatibilidad con Amazon Cognito en el asistente de autenticación no admite la función de identidades federadas de Amazon Cognito.

Instalación

Puedes usar las bibliotecas con una instalación local, si usas un sistema de compilación como webpack, o incluir JavaScript paquetes prediseñados con etiquetas en tu código html. `<script>`

- Utilice el siguiente comando para instalar la librería utilizando NPM:

```
npm install @aws/amazon-location-utilities-auth-helper
```

- Use el siguiente comando en su archivo HTML para cargar el script:

```
<script src="https://unpkg.com/@aws/amazon-location-utilities-auth-helper@1.x/dist/amazonLocationAuthHelper.js"></script>
```

Importación

Para usar una función específica en tu JavaScript aplicación, debes importarla. El siguiente código se utiliza para importar la función `withIdentityPoolId` a la aplicación.

```
import { withIdentityPoolId } from '@aws/amazon-location-utilities-auth-helper';
```

Funciones de autenticación

Los ayudantes de autenticación de Amazon Location incluyen las siguientes funciones que devuelven un objeto `AuthHelper`:

- `async withIdentityPoolId(identityPoolId: string): AuthHelper`— Esta función devuelve un `AuthHelper` objeto, inicializado para funcionar con Amazon Cognito
- `async withAPIKey(API_KEY: string): AuthHelper`— Esta función devuelve un `AuthHelper` objeto, inicializado para que funcione con claves de API.

El objeto `AuthHelper` proporciona las siguientes funciones:

- `AuthHelper.getMapAuthenticationOptions()`— Esta función del `AuthHelper` objeto devuelve un JavaScript objeto con el `transformRequest` que se puede utilizar con las opciones del mapa en MapLibre JS. Solo se proporciona cuando se inicializa con un grupo de identidades.
- `AuthHelper.getLocationClientConfig()`— Esta función del `AuthHelper` objeto devuelve un JavaScript objeto con el `credentials` que se puede utilizar para inicializar un `LocationClient`.
- `AuthHelper.getCredentials()`— Esta función del `AuthHelper` objeto devuelve las credenciales internas de Amazon Cognito. Solo se proporciona cuando se inicializa con un grupo de identidades.

Ejemplo: inicialización de un objeto de MapLibre mapa con Amazon Cognito mediante un `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
});
```

Ejemplo: inicializar un objeto de MapLibre mapa con una clave de API (no `AuthHelper` es necesario en este caso)

```
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',
});
```

Ejemplo: inicialice el cliente de ubicación desde el AWS SDK para JS con Amazon Cognito y AuthHelper

En este ejemplo, se usa el AWS SDK para JavaScript la versión 3.

```
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
}));
```

Ejemplo: inicialice el cliente de ubicación desde el AWS SDK para JS mediante una clave de API y AuthHelper

En este ejemplo, se usa el AWS SDK para la JavaScript versión 3.

```
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
}));
```

Ayudantes de conversión de GeoJSON

[Los ayudantes de conversión de GeoJSON de Amazon Location proporcionan herramientas para convertir los tipos de datos de Amazon Location Service al formato estándar del sector GeoJSON y viceversa.](#) GeoJSON se utiliza, por ejemplo, MapLibre para representar datos geográficos en el mapa. Se trata de una biblioteca de código abierto que está disponible en GitHub <https://github.com/aws-geospatial/amazon-location-utilities-datatypes-js>.

Instalación

Puedes usar las bibliotecas con una instalación local, como un paquete web, o incluir JavaScript paquetes prediseñados con `<script>` etiquetas en tu código html.

- Utilice el siguiente comando para instalar la librería usando NPM.

```
npm install @aws/amazon-location-utilities-datatypes
```

- Use el siguiente comando en su archivo HTML para cargar el script:

```
<script src="https://unpkg.com/@aws/amazon-location-utilities-datatypes@1.x/dist/amazonLocationDataConverter.js"></script>
```

Importación

Para usar una función específica en tu JavaScript aplicación, debes importarla. El siguiente código se utiliza para importar la función `placeToFeatureCollection` a la aplicación.

```
import { placeToFeatureCollection } from '@aws/amazon-location-utilities-datatypes';
```

Funciones de conversión de GeoJSON

Los ayudantes de conversión de GeoJSON de Amazon Location incluyen las siguientes funciones:

- `placeToFeatureCollection(place: GetPlaceResponse | searchPlaceIndexForPositionResponse | searchPlaceIndexForTextResponse, keepNull: boolean): FeatureCollection`— Esta función convierte las respuestas de las funciones de búsqueda de lugares en un GeoJSON `FeatureCollection` con 1 o más características de punto.
- `devicePositionToFeatureCollection(devicePositions: GetDevicePositionResponse | BatchGetDevicePositionResponse | GetDevicePositionHistoryResponse | ListDevicePositionsResponse, keepNull: boolean)`— Esta función convierte las respuestas de las funciones de posición del dispositivo rastreador en un GeoJSON `FeatureCollection` con 1 o más funciones de punto.
- `routeToFeatureCollection(legs: CalculateRouteResponse): FeatureCollection`— Esta función convierte las respuestas de la función de cálculo de ruta en un GeoJSON `FeatureCollection` con una sola `MultiStringLine` función. Cada tramo de la ruta se representa mediante una `LineString` entrada en `MultiStringLine`.
- `geofenceToFeatureCollection(geofences: GetGeofenceResponse | PutGeofenceRequest | BatchPutGeofenceRequest | ListGeofencesResponse): FeatureCollection`— Esta función convierte la solicitud o respuesta de las funciones de geocerca en un `FeatureCollection` GeoJSON con características poligonales. Puede convertir las geocercas tanto en la respuesta como en la solicitud, lo que le permite mostrar las geocercas en un mapa antes de cargarlas con `PutGeofence` o `BatchPutGeofence`.

Esta función convertirá una geovalla circular en una característica con un polígono aproximado, pero también tendrá propiedades de «centro» y «radio» para recrear la geovalla circular, si es necesario (consulte la siguiente función).

- `featureCollectionToGeofences(featureCollection: FeatureCollection): BatchPutGeofenceRequestEntry[]`— Esta función convierte un GeoJSON `FeatureCollection` con características poligonales en una matriz de `BatchPutGeofenceRequestEntry` objetos, por lo que el resultado se puede utilizar para crear una solicitud a `BatchPutGeofence`.

Si una de las entidades `FeatureCollection` tiene propiedades de «centro» y «radio», se convertirá en una entrada de solicitud de geocerca circular, sin tener en cuenta la geometría del polígono.

Ejemplo: convierte los resultados de la búsqueda en una capa de puntos en MapLibre

En este ejemplo se usa el AWS SDK para la JavaScript versión 3.

```
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 de autenticación móvil de Android

Estas utilidades te ayudan a autenticarte cuando realizas llamadas a la API de Amazon Location Service desde tus aplicaciones de Android. Esto ayuda específicamente cuando se utilizan [Amazon Cognito](#) o [claves de API como método](#) de autenticación.

El SDK de autenticación móvil Android está disponible en github: [Amazon Location Service Mobile Authentication SDK for Android](#). Además, tanto el SDK de autenticación móvil como el AWS SDK están disponibles en el [repositorio de AWS Maven](#).

Instalación

Para usar el SDK de autenticación móvil, agrega las siguientes instrucciones de importación a tu `build.gradle` archivo en Android Studio.

```
implementation("software.amazon.location:auth:0.0.1")
```

```
implementation("com.amazonaws:aws-android-sdk-location:2.72.0")
```

Funciones de autenticación

El SDK auxiliar de autenticación tiene las siguientes funciones:

- `authHelper.authenticateWithApiKey("My-Amazon-Location-API-Key")`: `LocationCredentialsProvider`: Esta función devuelve una clave de API `LocationCredentialsProvider` inicializada para que funcione con ella.
- `authHelper.authenticateWithCognitoIdentityPool("My-Cognito-Identity-Pool-Id")`: `LocationCredentialsProvider`: Esta función devuelve un conjunto de identidades `LocationCredentialsProvider` inicializado para que funcione con un grupo de identidades de Amazon Cognito.

Uso

Para usar el SDK en su código, importe las siguientes clases:

```
import com.amazonaws.services.geo.AmazonLocationClient
import software.amazon.location.auth.AuthHelper
import software.amazon.location.auth.LocationCredentialsProvider
```

Tienes dos opciones a la hora de crear las instancias de ayuda de autenticación y de proveedor de clientes de ubicación. Puede crear una instancia con las [claves de la API de ubicación](#) de [Amazon o Amazon Cognito](#).

- Para crear una instancia auxiliar de autenticación mediante una clave de API de Amazon Location, declara la clase auxiliar de la siguiente manera:

```
var authHelper = AuthHelper(applicationContext)
var locationCredentialsProvider : LocationCredentialsProvider =
    authHelper.authenticateWithApiKey("My-Amazon-Location-API-Key")
```

- Para crear una instancia auxiliar de autenticación con Amazon Cognito, declare la clase auxiliar de la siguiente manera:

```
var authHelper = AuthHelper(applicationContext)
var locationCredentialsProvider : LocationCredentialsProvider =
    authHelper.authenticateWithCognitoIdentityPool("My-Cognito-Identity-Pool-Id")
```

Puedes crear una instancia de cliente de Amazon Location con el proveedor de credenciales de ubicación y realizar llamadas al servicio Amazon Location. En el siguiente ejemplo, se buscan lugares cercanos a una latitud y longitud especificadas.

```
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 de autenticación móvil para iOS

Estas utilidades te ayudan a autenticarte cuando realizas llamadas a la API de Amazon Location Service desde tus aplicaciones iOS. Esto ayuda específicamente cuando se utilizan [Amazon Cognito](#) o [claves de API como método](#) de autenticación.

El SDK de autenticación móvil para iOS está disponible en github: [Amazon Location Service Mobile Authentication SDK for iOS](#).

Instalación

Instala el SDK en un proyecto de Xcode:

1. Ve a Archivo y selecciona Add Package Dependencies en tu proyecto de XCode.
2. Escribe la URL del paquete: <https://github.com/aws-geospatial/amazon-location-mobile-auth-sdk-ios/> en la barra de búsqueda y pulsa la tecla Enter.
3. Seleccione el amazon-location-mobile-auth-sdk-ios paquete y pulse Add Package.
4. Seleccione el producto AmazonLocationiOSAuthSDK empaquetado y pulse Add Package.

Funciones de autenticación

El SDK auxiliar de autenticación tiene las siguientes funciones:

- `authHelper.authenticateWithApiKey("My-Amazon-Location-API-Key"): LocationCredentialsProvider`: Esta función devuelve una clave de API `LocationCredentialsProvider` inicializada para que funcione con ella.

- `authHelper.authenticateWithCognitoIdentityPool("My-Cognito-Identity-Pool-Id")`: `LocationCredentialsProvider`: Esta función devuelve un conjunto de identidades `LocationCredentialsProvider` inicializado para que funcione con un grupo de identidades de Amazon Cognito.

Uso

Para usar el SDK de autenticación móvil, añade las siguientes instrucciones a su actividad:

```
import AmazonLocationiOSAuthSDK
import AWSLocationXCF
```

Tienes dos opciones a la hora de crear las instancias de ayuda de autenticación y de proveedor de cliente de ubicación. Puede crear una instancia con las [claves de la API de ubicación](#) de [Amazon o Amazon Cognito](#).

- Para crear una instancia auxiliar de autenticación mediante una clave de API de Amazon Location, declara la clase auxiliar de la siguiente manera:

```
let authHelper = AuthHelper()
let locationCredentialsProvider = authHelper.authenticateWithAPIKey(apiKey: "My-
Amazon-Location-API-Key", region: "account-region")
```

- Para crear una instancia auxiliar de autenticación con Amazon Cognito, declare la clase auxiliar de la siguiente manera:

```
let authHelper = AuthHelper()
let locationCredentialsProvider =
  authHelper.authenticateWithCognitoUserPool(identityPoolId: "My-Amazon-Location-API-
Key", region: "account-region")
```

Puedes crear una instancia de cliente de Amazon Location con el proveedor de credenciales de ubicación y realizar llamadas al servicio Amazon Location. En el siguiente ejemplo, se buscan lugares cercanos a una latitud y longitud especificadas.

```
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 de seguimiento para dispositivos móviles Android

El SDK de seguimiento móvil de Amazon Location proporciona utilidades que ayudan a autenticar fácilmente, capturar las posiciones de los dispositivos y enviar actualizaciones de posición a los rastreadores de ubicación de Amazon. El SDK admite el filtrado local de las actualizaciones de ubicación con intervalos de actualización configurables. Esto reduce los costes de datos y optimiza la conectividad intermitente de las aplicaciones de Android.

El SDK de rastreo para Android está disponible en GitHub: [Amazon Location Mobile Tracking SDK for Android](#). Además, tanto el SDK de autenticación móvil como el AWS SDK están disponibles en el [repositorio de AWS Maven](#). El SDK de seguimiento de Android está diseñado para funcionar con el AWS SDK general.

En esta sección se tratan los siguientes temas del SDK de seguimiento móvil de Amazon Location para Android:

Temas

- [Instalación](#)
- [Uso](#)
- [Filtros](#)
- [Funciones de seguimiento del SDK para dispositivos móviles de Android](#)
- [Ejemplos](#)

Instalación

Para instalar el SDK, agrega las siguientes líneas a la sección de dependencias del archivo build.gradle en 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")
```

Uso

Este procedimiento te muestra cómo usar el SDK para autenticar y crear el objeto: `LocationTracker`

Note

Este procedimiento supone que ha importado la biblioteca mencionada en la [Instalación](#) sección.

1. Importe las siguientes clases en su código:

```
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. A continuación, cree una `AuthHelper`, ya que el `LocationCredentialsProvider` parámetro es obligatorio para crear un `LocationTracker` objeto:

```
// Create an authentication helper using credentials from Cognito
val authHelper = AuthHelper(applicationContext)
val locationCredentialsProvider : LocationCredentialsProvider =
    authHelper.authenticateWithCognitoIdentityPool("My-Cognito-Identity-Pool-Id")
```

3. Ahora, usa la `LocationCredentialsProvider` tecla y `LocationTrackerConfig` para crear un `LocationTracker` objeto:

```
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,  
)
```

Filtros

El SDK de seguimiento móvil de Amazon Location para Android tiene tres filtros de ubicación integrados.

- **TimeLocationFilter**: filtra la ubicación actual que se va a cargar en función de un intervalo de tiempo definido.
- **DistanceLocationFilter**: filtra las actualizaciones de ubicación en función de un umbral de distancia especificado.
- **AccuracyLocationFilter**: filtra las actualizaciones de ubicación comparando la distancia recorrida desde la última actualización con la precisión de la ubicación actual.

En este ejemplo se añaden filtros en el `LocationTracker` momento de la creación:

```
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,  
)
```

Este ejemplo habilita y deshabilita el filtro en tiempo de ejecución con `LocationTracker`:

```
// To enable the filter  
locationTracker?.enableFilter(TimeLocationFilter())
```

```
// To disable the filter
locationTracker?.disableFilter(TimeLocationFilter())
```

Funciones de seguimiento del SDK para dispositivos móviles de Android

El SDK de seguimiento móvil de Amazon Location para Android incluye las siguientes funciones:

- Clase: `LocationTracker`

```
constructor(context: Context, locationCredentialsProvider:
LocationCredentialsProvider, trackerName: String), o bien
constructor(context: Context, locationCredentialsProvider:
LocationCredentialsProvider, clientConfig: LocationTrackerConfig)
```

Se trata de una función inicializadora para crear un `LocationTracker` objeto. Requiere instancias de `trackerName` y `LocationCredentialsProvider`, opcionalmente, una instancia de `LocationTrackingConfig`. Si no se proporciona la configuración, se inicializará con los valores predeterminados.

- Clase: `LocationTracker`

```
start(locationTrackingCallback: LocationTrackingCallback)
```

Inicia el proceso de acceder a la ubicación del usuario y enviarla a un rastreador de ubicaciones de Amazon.

- Clase: `LocationTracker`

```
isTrackingInForeground()
```

Comprueba si el seguimiento de la ubicación está actualmente en curso.

- Clase: `LocationTracker`

```
stop()
```

Detiene el proceso de seguimiento de la ubicación del usuario.

- Clase: `LocationTracker`

```
startTracking()
```

Inicia el proceso de acceder a la ubicación del usuario y enviarla al AWS rastreador.

- Clase: `LocationTracker`

```
startBackground(mode: BackgroundTrackingMode, serviceCallback: ServiceCallback)
```

Inicia el proceso de acceder a la ubicación del usuario y enviarla al AWS rastreador mientras la aplicación está en segundo plano. `BackgroundTrackingMode` tiene las siguientes opciones:

- `ACTIVE_TRACKING`: Esta opción realiza un seguimiento activo de las actualizaciones de ubicación de un usuario.
- `BATTERY_SAVER_TRACKING`: esta opción rastrea las actualizaciones de ubicación del usuario cada 15 minutos.
- Clase: `LocationTracker`

```
stopBackgroundService()
```

Detiene el proceso de acceder a la ubicación del usuario y enviarla al AWS rastreador mientras la aplicación está en segundo plano.

- Clase: `LocationTracker`

```
getTrackerDeviceLocation()
```

Recupera la ubicación del dispositivo de Amazon Location Services.

- Clase: `LocationTracker`

```
getDeviceLocation(locationTrackingCallback: LocationTrackingCallback?)
```

Recupera la ubicación actual del dispositivo del cliente del proveedor de ubicaciones fusionadas y la carga en el rastreador de ubicaciones de Amazon.

- Clase: `LocationTracker`

```
uploadLocationUpdates(locationTrackingCallback: LocationTrackingCallback?)
```

Carga la ubicación del dispositivo a Amazon Location Services después de filtrarla en función de los filtros de ubicación configurados.

- Clase: `LocationTracker`

```
enableFilter(filter: LocationFilter)
```

Activa un filtro de ubicación concreto.

- Clase: `LocationTracker`

```
checkFilterIsExistsAndUpdateValue(filter: LocationFilter)
```

Deshabilita un filtro de ubicación en particular.

- Clase: `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())
```

Esto inicializa el `LocationTrackerConfig` con valores de parámetros definidos por el usuario. Si no se proporciona un valor de parámetro, se establecerá en un valor predeterminado.

- Clase: `LocationFilter`

```
shouldUpload(currentLocation: LocationEntry, previousLocation:
LocationEntry?): Boolean
```

`LocationFilter` es un protocolo que los usuarios pueden implementar para la implementación de sus filtros personalizados. Debe implementar la `shouldUpload` función para comparar la ubicación anterior y actual y devolver si se debe cargar la ubicación actual.

Ejemplos

El siguiente ejemplo de código muestra la funcionalidad del SDK de rastreo móvil.

En este ejemplo, se utiliza `LocationTracker` para iniciar y detener el seguimiento en segundo plano:

```
// 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 de seguimiento móvil para iOS

El SDK de seguimiento móvil de Amazon Location proporciona utilidades que ayudan a autenticar fácilmente, capturar las posiciones de los dispositivos y enviar actualizaciones de posición a los rastreadores de ubicación de Amazon. El SDK admite el filtrado local de las actualizaciones de ubicación con intervalos de actualización configurables. Esto reduce los costes de datos y optimiza la conectividad intermitente de las aplicaciones iOS.

El SDK de rastreo para iOS está disponible en GitHub: [Amazon Location Mobile Tracking SDK for iOS](#).

En esta sección se tratan los siguientes temas del SDK de seguimiento móvil de Amazon Location para iOS:

Temas

- [Instalación](#)
- [Uso](#)
- [Filtros](#)
- [Funciones de seguimiento del SDK para iOS Mobile](#)
- [Ejemplos](#)

Instalación

Usa el siguiente procedimiento para instalar el SDK de rastreo móvil para iOS:

1. En tu proyecto de Xcode, ve a Archivo y selecciona Add Package Dependencies.
2. Escribe la siguiente URL: <https://github.com/aws-geospatial/amazon-location-mobile-tracking-sdk-ios/> en la barra de búsqueda y pulsa la tecla Enter.

3. Seleccione el `amazon-location-mobile-tracking-sdk-ios` paquete y haga clic en Add Package.
4. Seleccione el producto `AmazonLocationiOSTrackingSDK` del paquete y haga clic en Add Package.

Uso

El siguiente procedimiento muestra cómo crear un ayudante de autenticación con las credenciales de Cognito.

1. Tras instalar la biblioteca, debe añadir una o ambas descripciones al archivo `info.plist`:

```
Privacy - Location When In Use Usage Description
Privacy - Location Always and When In Use Usage Description
```

2. A continuación, importa lo siguiente `AuthHelper` en tu clase:

```
import AmazonLocationIOSAuthSDK
import AmazonLocationiOSTrackingSDK
```

3. A continuación, creará un `AuthHelper` objeto y lo utilizará con el AWS SDK mediante la creación de un ayudante de autenticación con las credenciales de 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)
```

Filtros

El SDK de seguimiento móvil de Amazon Location para iOS tiene tres filtros de ubicación integrados.

- `TimeLocationFilter`: filtra la ubicación actual que se va a cargar en función de un intervalo de tiempo definido.
- `DistanceLocationFilter`: filtra las actualizaciones de ubicación en función de un umbral de distancia especificado.
- `AccuracyLocationFilter`: filtra las actualizaciones de ubicación comparando la distancia recorrida desde la última actualización con la precisión de la ubicación actual.

En este ejemplo se añaden filtros en el `LocationTracker` momento de la creación:

```
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,  
)
```

Este ejemplo habilita y deshabilita el filtro en tiempo de ejecución con `LocationTracker`:

```
// To enable the filter  
locationTracker?.enableFilter(TimeLocationFilter())  
  
// To disable the filter  
locationTracker?.disableFilter(TimeLocationFilter())
```

Funciones de seguimiento del SDK para iOS Mobile

El SDK de seguimiento móvil de Amazon Location para iOS incluye las siguientes funciones:

- Clase: `LocationTracker`

```
init(provider: LocationCredentialsProvider, trackerName: String, config: LocationTrackerConfig? = nil)
```

Se trata de una función inicializadora para crear un `LocationTracker` objeto. Requiere instancias de `trackerName` y `LocationCredentialsProvider`, opcionalmente, una instancia de `LocationTrackingConfig`. Si no se proporciona la configuración, se inicializará con los valores predeterminados.

- Clase: `LocationTracker`

```
setTrackerConfig(config: LocationTrackerConfig)
```

Esto establece que la configuración del rastreador surta efecto en cualquier momento después de la inicialización del rastreador de ubicación.

- Clase: `LocationTracker`

```
getTrackerConfig()
```

Esto obtiene la configuración de seguimiento de ubicación para usarla o modificarla en tu aplicación.

Devoluciones: `LocationTrackerConfig`

- Clase: `LocationTracker`

```
getDeviceId()
```

Obtiene el identificador del dispositivo generado por el rastreador de ubicación.

Devoluciones: `String?`

- Clase: `LocationTracker`

```
startTracking()
```

Inicia el proceso de acceder a la ubicación del usuario y enviarla al AWS rastreador.

- Clase: `LocationTracker`

```
resumeTracking()
```

Reanuda el proceso de acceder a la ubicación del usuario y enviarla al AWS rastreador.

- Clase: `LocationTracker`

```
stopTracking()
```

Detiene el proceso de seguimiento de la ubicación del usuario.

- Clase: `LocationTracker`

```
startBackgroundTracking(mode: BackgroundTrackingMode)
```

Inicia el proceso de acceder a la ubicación del usuario y enviarla al AWS rastreador mientras la aplicación está en segundo plano. `BackgroundTrackingMode` tiene las siguientes opciones:

- `Active`: Esta opción no detiene automáticamente las actualizaciones de ubicación.
 - `BatterySaving`: Esta opción detiene automáticamente las actualizaciones de ubicación.
 - `None`: En general, esta opción desactiva las actualizaciones de ubicación en segundo plano.
- Clase: `LocationTracker`

```
resumeBackgroundTracking(mode: BackgroundTrackingMode)
```

Reanuda el proceso de acceder a la ubicación del usuario y enviarla al AWS rastreador mientras la aplicación está en segundo plano.

- Clase: `LocationTracker`

```
stopBackgroundTracking()
```

Detiene el proceso de acceder a la ubicación del usuario y enviarla al AWS rastreador mientras la aplicación está en segundo plano.

- Clase: `LocationTracker`

```
getTrackerDeviceLocation(nextToken: String?, startTime: Date? = nil,  
endTime: Date? = nil, completion: @escaping (Result<GetLocationResponse,  
Error>)
```

Recupera las ubicaciones de seguimiento cargadas para el dispositivo del usuario entre la fecha y la hora de inicio y finalización.

Devoluciones: Void

- Clase: LocationTrackerConfig

```
init()
```

Esto inicializa el LocationTrackerConfig con los valores predeterminados.

- Clase: 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)
```

Esto inicializa el LocationTrackerConfig con valores de parámetros definidos por el usuario. Si no se proporciona un valor de parámetro, se establecerá en un valor predeterminado.

- Clase: LocationFilter

```
shouldUpload(currentLocation: LocationEntity, previousLocation: LocationEntity?, trackerConfig: LocationTrackerConfig)
```

LocationFilterEs un protocolo que los usuarios pueden implementar para la implementación de sus filtros personalizados. Un usuario tendría que implementar una shouldUpload función para comparar la ubicación anterior y actual y regresar si se debe cargar la ubicación actual.

Ejemplos

En esta sección se detallan ejemplos del uso del SDK de seguimiento móvil de Amazon Location para iOS.

Note

Asegúrese de que los permisos necesarios estén establecidos en el `info.plist` archivo. Son los mismos permisos que se enumeran en la [Uso](#) sección.

El siguiente ejemplo muestra la funcionalidad para rastrear la ubicación del dispositivo y recuperar las ubicaciones rastreadas:

[Privacy - Location When In Use Usage Description](#)
[Privacy - Location Always and When In Use Usage Description](#)

Comience a rastrear la ubicación:

```
do {
    try locationTracker.startTracking()
    }
catch TrackingLocationError.permissionDenied {
    // Handle permissionDenied by showing the alert message or opening the app
    settings
    }
}
```

Reanude el seguimiento de la ubicación:

```
do {
    try locationTracker.resumeTracking()
    }
catch TrackingLocationError.permissionDenied {
    // Handle permissionDenied by showing the alert message or opening the app settings
    }
}
```

Deja de rastrear la ubicación:

```
locationTracker.stopTracking()
```

Iniciar el seguimiento en segundo plano:

```
do {

    locationTracker.startBackgroundTracking(mode: .Active) // .Active, .BatterySaving, .None
    }
catch TrackingLocationError.permissionDenied {
    // Handle permissionDenied by showing the alert message or opening the app settings
    }
}
```

Reanude el seguimiento en segundo plano:

```
do {
```

```
locationTracker.resumeBackgroundTracking(mode: .Active)
}
catch TrackingLocationError.permissionDenied {
    // Handle permissionDenied by showing the alert message or opening the app settings
}
```

Para detener el seguimiento en segundo plano:

```
locationTracker.stopBackgroundTracking()
```

Recupera las ubicaciones rastreadas del dispositivo desde el rastreador:

```
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 de localización de Amazon

Amazon Location Service proporciona operaciones de API para acceder mediante programación a la funcionalidad de ubicación. Esto incluye las API para mapas, lugares, rutas, rastreadores, geovallas y el etiquetado de sus recursos. Para obtener información sobre las acciones de API disponibles, consulte la [referencia de la API de Amazon Location Service](#).

Puede encontrar ejemplos en el capítulo [Ejemplos de código](#) de esta guía.

Uso de Amazon Location con un AWS SDK

AWS Los kits de desarrollo de software (SDK) están disponibles para muchos lenguajes de programación populares. Cada SDK proporciona una API, ejemplos de código y documentación que facilitan a los desarrolladores la creación de AWS aplicaciones en su idioma preferido.

Para obtener más información sobre los SDK disponibles para su uso con Amazon Location Service por idioma, consulte [SDK por lenguaje](#) en esta guía.

Versiones del SDK

Te recomendamos que utilices la versión más reciente del AWS SDK y cualquier otro SDK que utilices en tus proyectos, y que mantengas los SDK actualizados. El AWS SDK te proporciona las funciones y funciones más recientes, así como actualizaciones de seguridad. Para encontrar la versión más reciente del AWS SDK JavaScript, por ejemplo, consulte el tema de [instalación del navegador](#) en el AWS SDK para ver la JavaScript documentación.

Actualizaciones del mensaje de error de la API de ubicación de Amazon

A partir del 1 de agosto de 2023, el equipo de Amazon Location cambiará los mensajes de error de la API tal y como se describe en las siguientes tablas. Los códigos de error no se cambiarán. Si sus aplicaciones dependen de cadenas de mensajes de error exactas, debe actualizarlas con las nuevas cadenas. Para obtener ayuda con preguntas o problemas, póngase en contacto con AWS Support.

Temas

- [Lugares](#)
- [Mapas](#)
- [Rastreadores](#)
- [Rutas](#)
- [Metadatos](#)
- [Geovallas](#)

Lugares

Lugares

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
500	InternalServerErrorException	Internal Server Exception	Internal server error. Try again later.
404	ResourceNotFoundException	resource <PlaceIndexName> not found, reason: <Reason> Resource '<PlaceIndexName>' not found placeIdx<PlaceIndexName> not found, reason: <Reason> no place index with name '%s' found	Place index not found: <PlaceIndexName>.
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.

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
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.
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.

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
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	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 https://wikipedia.org/wiki/IETF_language_tag .
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.

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
400	ValidationException	Grab is only supported in the ap-southeast-1 region	'DataSource' Grab must only be used in following regions: ap-southeast-1.
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 https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/	Place index resources have exceeded the quota per account per region. For more information, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/ .
409	ConflictException	Resource already exists	Place index already exists: <PlaceIndexName>.

Mapas

Mapas

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
500	InternalServerErrorException	Internal Server Exception	Internal server error. Try again later.

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
		unable to find style template Error fetching style 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.

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
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>.
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 https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapGlyphs.html .

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
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 https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapGlyphs.html .
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 https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapSprites.html .

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
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 https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapSprites.html .
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 https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapSprites.html .
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'.

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
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.
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.

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
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>).
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 $0..2^{\text{Zoom}} - 1$ ($0..<MaxTileCoordinate>$)	y-coordinate part of 'Y' must be an integer in the range $0..2^{\text{Zoom}} - 1$ ($0..<MaxTileCoordinate>$).
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.

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
400	ValidationException	Expired token	'NextToken' must not be expired.
400	ValidationException	Unsupported Map Style: <Style>	<Style> is not a supported map style. For more information about list of supported map styles, see https://docs.aws.amazon.com/location/latest/APIReference/API_MapConfiguration.html .
402	ServiceQuotaExceededException	Map resources per account exceeded quota limits. For more info, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/	Map resources have exceeded the quota per account per region. For more information, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/ .
409	ConflictException	Resource already exists	Map already exists: <MapName>.

Rastreadores

Rastreadores

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
500	InternalServerErrorException	Internal Server Exception internal server error unable to retrieve point from the storage unable to verify tracker Error processing List request	Internal server error. Try again later.
404	ResourceNotFoundException	tracker not found: <TrackerName> Tracker with name <TrackerName> was not found	Tracker not found: <TrackerName>.
404	ResourceNotFoundException	association not found: TrackerName <TrackerName>; and ConsumerArn <ConsumerArn >	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	'ConsumerArn' must refer to a resource

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
		in the same region as the tracker it is associated to	in the same region as the tracker it is associated with.
400	ValidationException	'ConsumerArn' must refer to a resource in the same AWS account as the tracker it is 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

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.
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 https://docs.aws.amazon.com/kms/latest/developerguide/key-state.html .

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
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.
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 https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/ .

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
402	ServiceQuotaExceededException	Trackers per account exceeded quota limits. For more info, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/	Tracking resources have exceeded the quota per account per region. For more information, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/ .
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>.

Rutas

Rutas

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
500	InternalServerErrorException	Internal Server Exception	Internal server error. Try again later.

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
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.
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.

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
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
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.

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
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.
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.

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
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]'.
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.

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
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.
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.

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
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>.
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>.

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
400	ValidationException	No route found	No route found. For more information, see https://developer.amazon.com/documentation/routing-api/dev_guide/topics/notice.html .
400	ValidationException	No route found	No route found.
402	ServiceQuotaExceededException	Route calculators per account exceeded quota limits. For more info, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/	Route calculator resources have exceeded the quota per account per region. For more information, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/ .
409	ConflictException	Resource already exists	Route calculator already exists: <RouteCalculatorName>.

Metadatos

Metadatos

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
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>.
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	'ExpireTime' must not be in the past.

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
		be in the past when creating a key	
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.
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.

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
409	ConflictException	API key named <APIKeyName> already exists	Api key already exists: <APIKeyName>.

Geovallas

Geovallas

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
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>	Internal server error. Try again later.

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
404	ResourceNotFoundException	collection not found: <GeofenceCollectionName> <GeofenceCollectionName> geofence collection not found Resource not found error no geofence with given name found	Geofence Collection not found: <GeofenceCollectionName>.
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 https://docs.aws.amazon.com/kms/latest/developerguide/create-keys.html#create-symmetric-cmk .
400	ValidationException	Invalid token	'NextToken' must be a valid token.

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
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'.
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 https://docs.aws.amazon.com/kms/latest/developerguide/key-state.html .
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.

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
400	ValidationException	access denied	Symmetric Customer Master Key (CMK) must allow Amazon Location to create grants to its KMS key.
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>.

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
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'.
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.

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
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 https://www.rfc-editor.org/rfc/rfc7946.html#appendix-A.3 .
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.

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
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>.
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.

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
400	ValidationException	arn: invalid prefix	'SourceArn' must be a valid ARN. For more information, see https://docs.aws.amazon.com/general/latest/gr/AWS-arns-and-namespaces.html .
400	ValidationException	arn: not enough sections	'SourceArn' must be a valid ARN. For more information, see https://docs.aws.amazon.com/general/latest/gr/AWS-arns-and-namespaces.html .
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 https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/	Geofence collection resources have exceeded the quota per account per region. For more information, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/ .

Código de error	Excepción	Mensaje de error anterior	Nuevo mensaje de error
409	ConflictException	collection already exists: <Geofence CollectionName>	Geofence Collection already exists: <GeofenceCollectionName>.
409	ConflictException	Resource conflict error	Geofence already exists: <Geofence Name>.

Ejemplos de código y tutoriales para trabajar con Amazon Location Service

En este tema, se muestra una lista de ejemplos de código, tutoriales y entradas de blog que lo ayudarán a obtener más información sobre Amazon Location Service. Cada ejemplo de código incluye una descripción de su funcionamiento.

Puedes encontrar más ejemplos en la página [AWS Geospacial, en la GitHub página](#) de [AWS ejemplos GitHub de Amazon Location](#) y en el [sitio del AWS blog](#).

Note

Es bueno entender la diferencia entre la página AWS geoespacial y la GitHub página de AWS ejemplos GitHub .

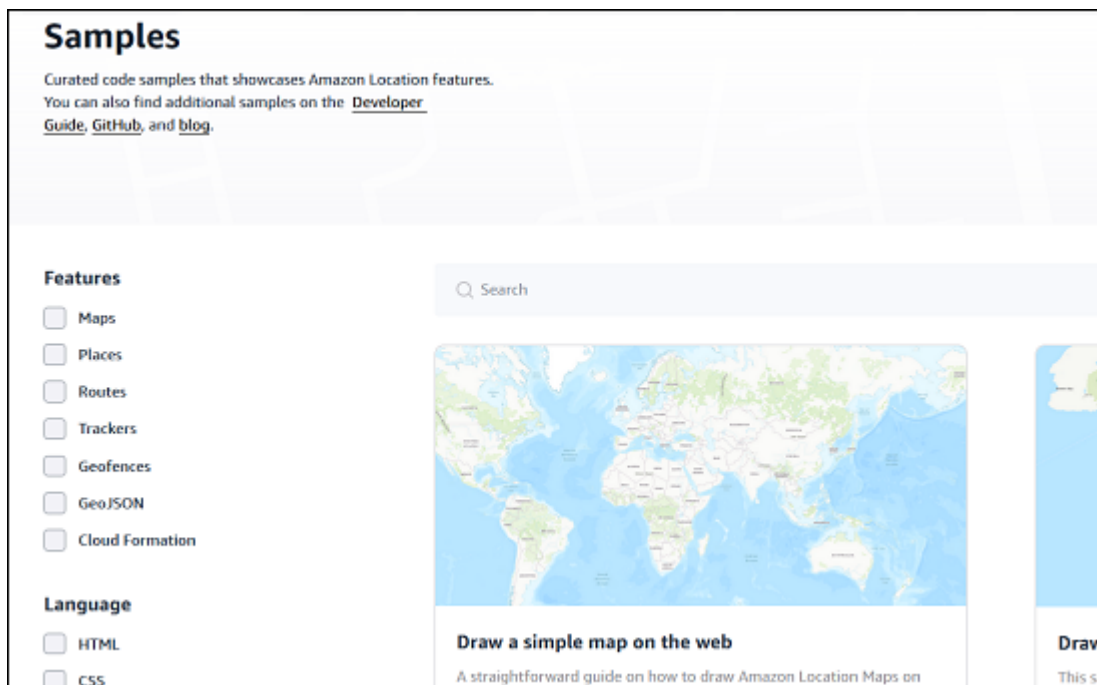
- Geoespacial GitHub: la [GitHub página AWS Geoespacial](#) incluye ejemplos creados y mantenidos por el equipo de Amazon Location Service.
- Muestras GitHub: la [GitHub página de AWS muestras de Amazon Location](#) incluye muestras que se crearon para Amazon Location, pero que pueden o no mantenerse activamente.

El tutorial de [inicio rápido](#) es un buen punto de partida antes de usar otros ejemplos, ya que enseña cómo completar los requisitos previos que son útiles en la mayoría de los ejemplos.

Temas

- [Sitio de ejemplo de Amazon Location](#)
- [Tutorial: inicio rápido](#)
- [Tutorial: enriquecimiento de bases de datos](#)
- [Ejemplo: aplicación de exploración](#)
- [Ejemplo: aplicar estilo a un mapa](#)
- [Ejemplo: dibujar marcadores](#)
- [Ejemplo: dibuja puntos agrupados](#)
- [Ejemplo: dibuja un polígono](#)
- [Ejemplo: cambiar el idioma del mapa](#)
- [Blog: notificaciones de tiempo de entrega estimado](#)
- [Ejemplo: actualizaciones de posición de transmisión](#)
- [Ejemplo: aplicación móvil de geofencing and Tracking](#)

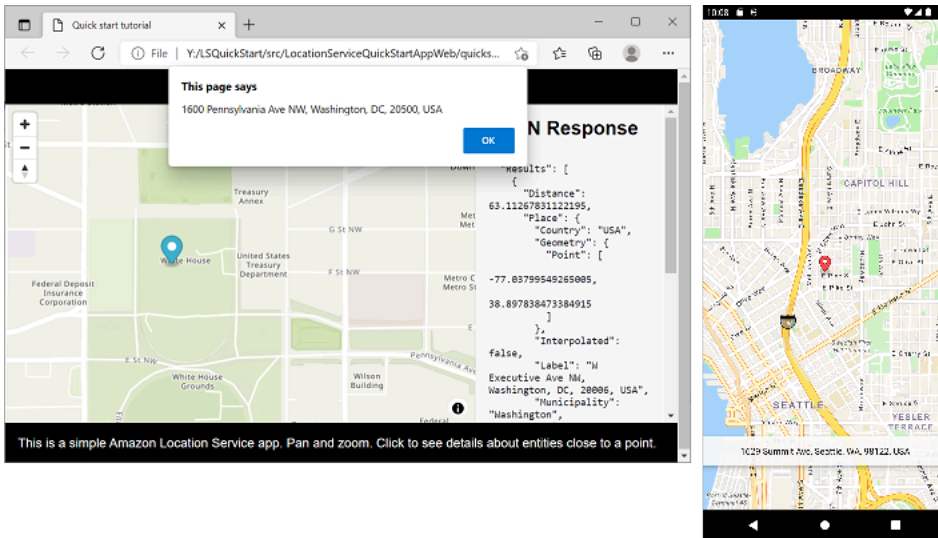
Sitio de ejemplo de Amazon Location



Puede ver ejemplos con el código fuente de Amazon Location Service en acción en el [sitio de demostraciones de Amazon Location](#). Este sitio incluye una [demostración web alojada](#) y también una aplicación de demostración para [Android](#).

También puede encontrar una amplia gama de ejemplos, que se pueden filtrar por características, idioma y plataforma en la página de [ejemplos](#) del sitio.

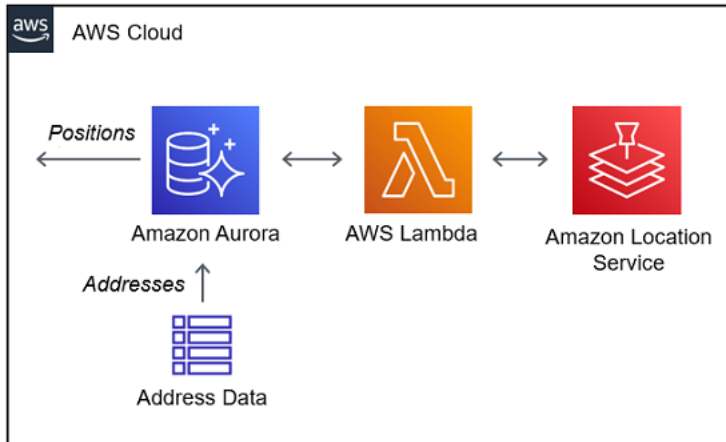
Tutorial: inicio rápido



Hay tutoriales de inicio rápido disponibles para dispositivos web, iOS y Android. Para cada plataforma, el tutorial te muestra cómo añadir un mapa interactivo a una aplicación y cómo realizar llamadas a las API de Amazon Location Service desde tu aplicación. El tutorial está disponible en JavaScript en una página web estática, en Kotlin para una aplicación de teléfono Android o en Swift para una aplicación de iOS.

- JavaScript para un enlace a la documentación de una página web estática: [Crear una aplicación web](#)
- Enlace a la documentación de Kotlin para una aplicación de Android: [Inicio rápido con Amazon Location Service](#)
- Enlace a la documentación de Swift para una aplicación iOS: [Crear una aplicación para iOS](#)

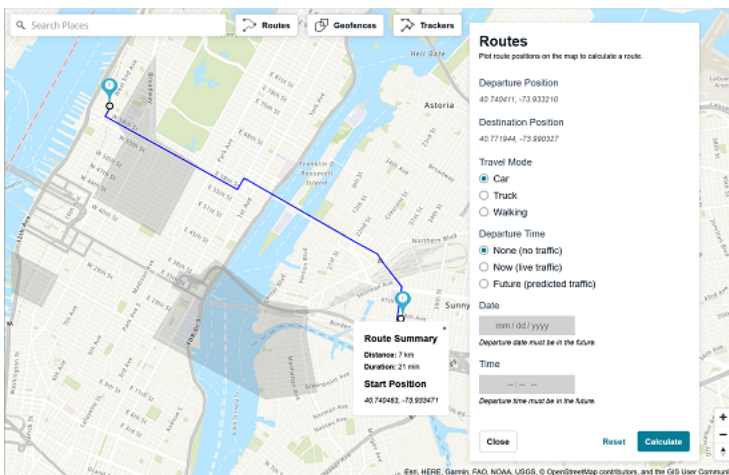
Tutorial: enriquecimiento de bases de datos



En este tutorial se muestra cómo utilizar Amazon Location Service, llamado from, AWS Lambda para normalizar las direcciones y añadir latitud y longitud a los registros de una base de datos de Amazon Aurora. Utiliza Amazon Aurora y AWS Lambda.

Enlace a la documentación: [Funciones definidas por el usuario de Amazon Aurora PostgreSQL para Amazon Location Service](#)

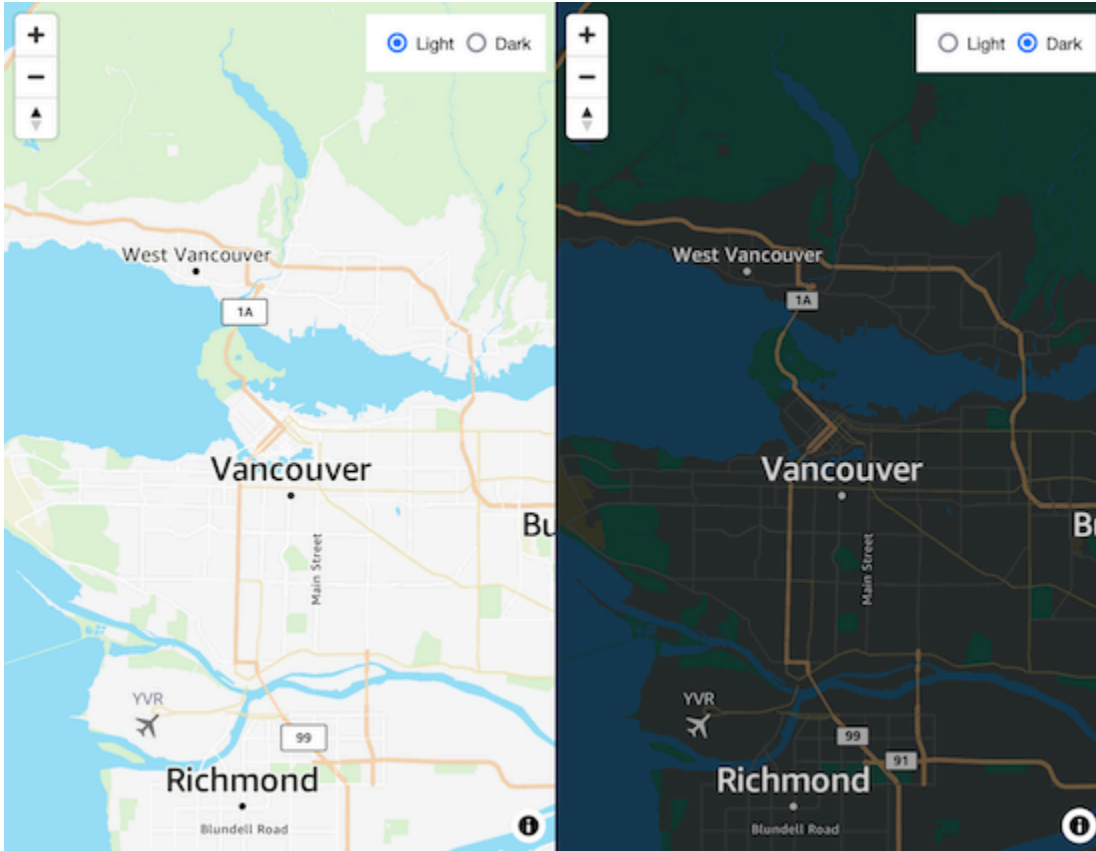
Ejemplo: aplicación de exploración



Una de las mejores formas de obtener información sobre las funciones de Amazon Location Service es utilizar la [función de exploración](#) de la consola de Amazon Location. Este ejemplo completo de aplicación web imita las funciones de mapas, lugares, rutas, geovallas y rastreadores de la consola para mostrarle cómo recrear estas funciones en su propia aplicación. Utiliza Amplify, React y JavaScript

GitHub Enlace de muestras: [Explore la aplicación de muestra](#)

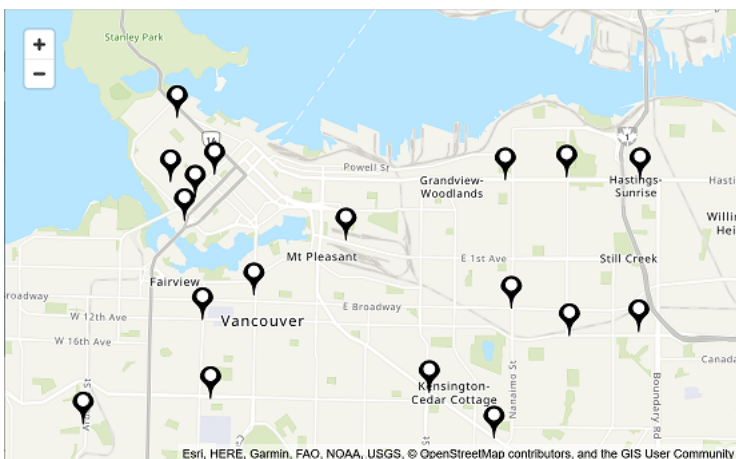
Ejemplo: aplicar estilo a un mapa



Este ejemplo de código muestra cómo cambiar entre un mapa satelital y un mapa de carreteras vectorial, utilizando MapLibre in JavaScript. Uses MapLibre, el asistente de autenticación de Amazon Location y JavaScript.

GitHub Enlace geoespacial: [mapa interactivo con cambio de estilo](#)

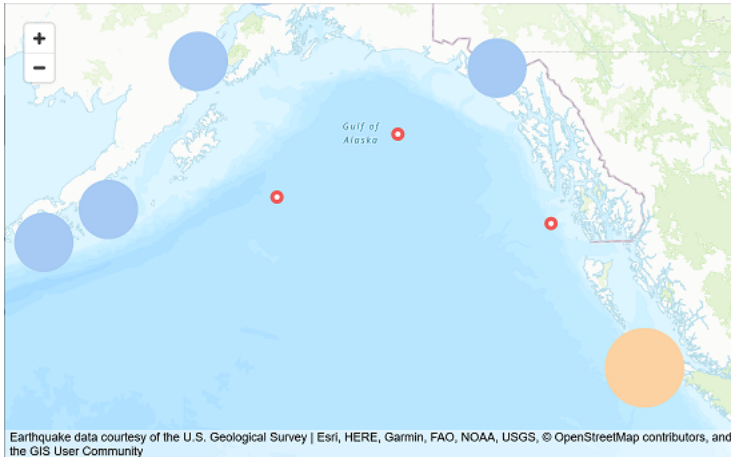
Ejemplo: dibujar marcadores



Este ejemplo de código muestra las ubicaciones de Amazon Locker en Vancouver, Columbia Británica (Canadá). Muestra cómo dibujar marcadores en las ubicaciones de los puntos. Utiliza Node.js MapLibre, React, el asistente de autenticación de Amazon Location y JavaScript.

GitHub Enlace geoespacial: [mapa interactivo con marcadores](#) en los puntos

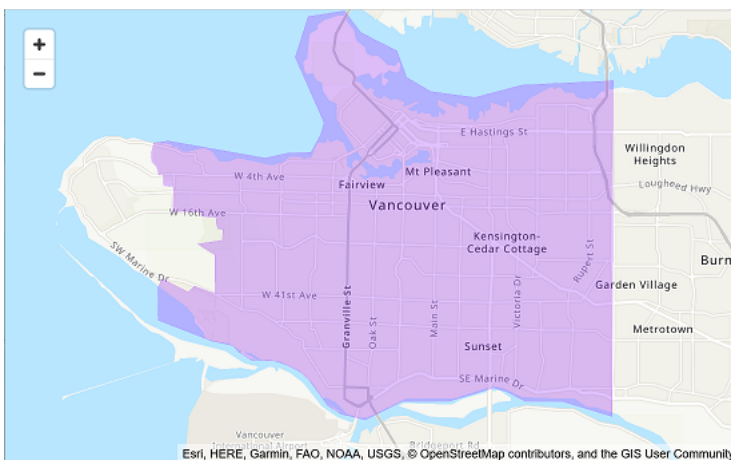
Ejemplo: dibuja puntos agrupados



Este ejemplo de código utiliza datos sísmicos del USGS para mostrar cómo dibujar puntos que se agrupan cuando están muy juntos en el mapa. Usos MapLibre, Node.js, React, Amplify y JavaScript.

GitHub Enlace de ejemplos: [mapa interactivo con grupos de puntos](#)

Ejemplo: dibuja un polígono



Este ejemplo de código muestra cómo dibujar un polígono en el mapa. Utiliza Node.js MapLibre, React, el asistente de autenticación de Amazon Location y JavaScript.

GitHub Enlace geoespacial: [mapa interactivo](#) con polígonos

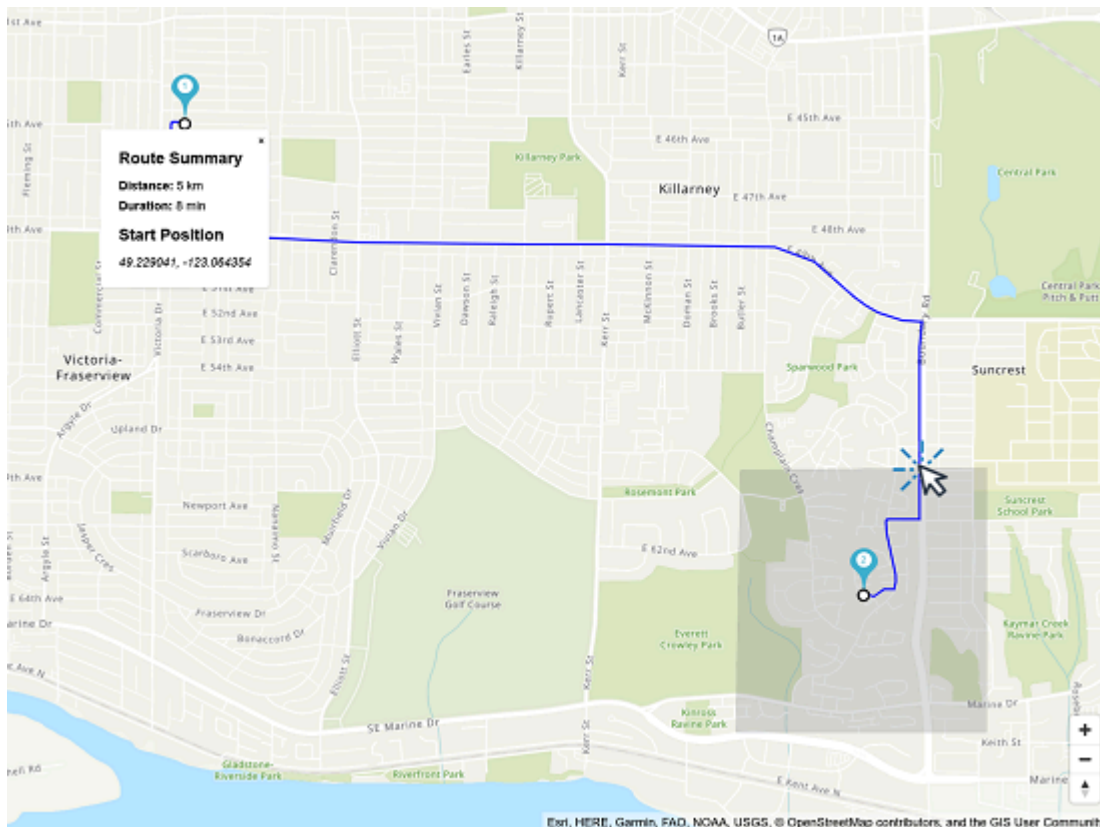
Ejemplo: cambiar el idioma del mapa



Este ejemplo de código muestra cómo puede cambiar el idioma de visualización de los mapas en Amazon Location. Utiliza Amplify, React y MapLibre

GitHub Enlace de muestra: Ejemplo de [cambio de idioma de mapa](#)

Blog: notificaciones de tiempo de entrega estimado



Esta entrada de blog muestra diferentes formas de notificar a los clientes los tiempos de entrega estimados. Explica el uso de rutas para mostrar el tiempo de conducción estimado para, a continuación, utilizar rastreadores y geovallas para notificar cuando un conductor se acerca al cliente. Utiliza Amplify, React, Amazon y Amazon EventBridge Simple Notification Service (Amazon SNS).

Enlace al blog: [hora estimada de llegada y notificaciones de proximidad](#)

Ejemplo: actualizaciones de posición de transmisión



Aplicación Kinesis Stream To Tracker: en este ejemplo se muestra cómo utilizar Kinesis Data Stream para publicar actualizaciones de rastreadores con Amazon Location Service. El ejemplo es una aplicación lambda desplegable escrita en Python que se puede integrar con un Kinesis Data Stream para consumir los eventos de Kinesis y actualizar por lotes las posiciones de los dispositivos.

Enlace al repositorio: [Amazon Location Amazon Kinesis Data Streams Stream To Tracker App](#)

Para obtener más información sobre el rastreo y las geocercas, consulte la documentación sobre [geocercas y rastreadores](#). [Los desarrolladores pueden implementar la aplicación siguiendo la documentación del Serverless Application Repository de AWS o directamente desde la consola Lambda.](#)

Aplicación de ejemplo de transmisión de posición del dispositivo: este ejemplo de código muestra cómo transmitir los datos de posición del dispositivo a una transmisión de datos de Kinesis Data Stream y cómo funcionan las notificaciones de geocerca. Esta aplicación depende de la aplicación de ejemplo Kinesis Stream to Tracker, indicada anteriormente, que se ejecute para actualizar las posiciones de los rastreadores transmitidos en Amazon Location Service.

Enlace al repositorio: [aplicación de muestra Amazon Location Device Position Streaming](#)

Ejemplo: aplicación móvil de geofencing and Tracking

Este ejemplo de aplicación muestra cómo interactúan un rastreador y una geocerca mediante una combinación de funciones de Lambda y AWS IoT Amazon Location. Hay tutoriales disponibles para iOS y Android.

Enlace al tutorial: [Ejemplo de aplicación móvil Geofence and Tracker](#)

Cómo utilizar Amazon Location Service

Puede utilizar las funciones de Amazon Location Service para completar tareas geográficas y relacionadas con la ubicación. A continuación, puede combinar estas tareas para abordar casos de uso más complejos, como el geomarketing, la entrega y el seguimiento de activos.

Cuando esté listo para incorporar funciones de ubicación a su aplicación, utilice los siguientes métodos para utilizar la funcionalidad de Amazon Location Service, en función de sus objetivos e inclinaciones:

- Herramientas de exploración: si quiere experimentar con los recursos de Amazon Location, las siguientes herramientas son la forma más rápida de acceder a las API y probarlas:
 - La [consola de Amazon Location](#) ofrece una variedad de herramientas de acceso rápido. Puede crear y administrar sus recursos y probar las API desde la página [Explorar](#). La consola también es útil para crear recursos (normalmente una tarea que se realiza una sola vez) como preparación para usar cualquiera de los otros métodos que se describen más adelante.
 - La [AWSinterfaz de línea de comandos](#) (CLI) le permite crear recursos y acceder a las API de ubicación de Amazon mediante un terminal. Maneja la autenticación AWS CLI cuando la configura con sus credenciales.
 - Puede ver [ejemplos de código y tutoriales](#) que muestran cómo realizar tareas con las API de Amazon Location Service. Esto incluye [un ejemplo](#) que imita gran parte de la funcionalidad de la página Explorar de la consola.
- SDK de plataforma: si no está visualizando los datos en un mapa, puede usar cualquiera de las [AWSherramientas estándar](#) como base AWS.
 - Están disponibles los siguientes SDK: C++, Go, Java, .NET JavaScript, Node.js, PHP, Python y Ruby.
- SDK y bibliotecas de frontend: si desea utilizar Amazon Location para crear una aplicación en una plataforma móvil o visualizar datos en un mapa en cualquier plataforma, dispone de las siguientes opciones:
 - Las AWS Amplify bibliotecas integran Amazon Location en [iOS](#), [Android](#) y aplicaciones [JavaScriptweb](#).
 - Las MapLibre bibliotecas te permiten renderizar mapas del lado del cliente en [iOS](#), [Android](#) y aplicaciones [JavaScriptweb](#) mediante Amazon Location.

- Las bibliotecas Tangram ES permiten renderizar mapas 2D y 3D a partir de datos vectoriales mediante OpenGL ES en aplicaciones web para [iOS](#) y [Android](#). También existe Tangram para [JavaScript](#) aplicaciones web.
- Envío de solicitudes HTTPS directas: si trabaja con un lenguaje de programación para el que no hay un SDK disponible, o si quiere tener más control sobre cómo enviar una solicitud a AWS, puede acceder a Amazon Location enviando solicitudes HTTPS directas autenticadas mediante el proceso de firma de Signature Version 4. Para obtener más información sobre el [Proceso de firma Signature Version 4](#), consulte Referencia general de AWS.

En este capítulo se describen muchas de las tareas que son comunes a las aplicaciones que utilizan datos de ubicación. La sección de [casos de uso comunes](#) describe cómo combinarlos con otros AWS servicios para lograr casos de uso más complejos.

Temas

- [Requisitos previos para utilizar Amazon Location Service](#)
- [Uso de Amazon Location Maps en su aplicación](#)
- [Búsqueda de datos de ubicación y geolocalización mediante Amazon Location](#)
- [Calcular rutas utilizando Amazon Location Service](#)
- [Geocercar un área de interés mediante Amazon Location](#)
- [Etiquetado de los recursos de Amazon Location Service](#)
- [Concesión de acceso a Amazon Location Service](#)
- [Monitoreo de Amazon Location Service](#)
- [Creación de recursos de Amazon Location Service con AWS CloudFormation](#)

Requisitos previos para utilizar Amazon Location Service

En esta sección, se describen los pasos para utilizar Amazon Location Service. Debe disponer de una Cuenta de AWS de Amazon Location y haber configurado el acceso a ella para los usuarios que quieran utilizarla.

Inscríbese en un Cuenta de AWS

Si no tiene una Cuenta de AWS, complete los siguientes pasos para crearlo.

Para suscribirte a una Cuenta de AWS

1. Abra <https://portal.aws.amazon.com/billing/signup>.
2. Siga las instrucciones que se le indiquen.

Parte del procedimiento de registro consiste en recibir una llamada telefónica e indicar un código de verificación en el teclado del teléfono.

Cuando te registras en un Cuenta de AWS, Usuario raíz de la cuenta de AWS se crea un. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar [tareas que requieren acceso de usuario raíz](#).

AWS te envía un correo electrónico de confirmación una vez finalizado el proceso de registro. Puede ver la actividad de la cuenta y administrar la cuenta en cualquier momento entrando en <https://aws.amazon.com/> y seleccionando Mi cuenta.

Creación de un usuario con acceso administrativo

Después de crear un usuario administrativo Cuenta de AWS, asegúrelo Usuario raíz de la cuenta de AWS AWS IAM Identity Center, habilite y cree un usuario administrativo para no usar el usuario root en las tareas diarias.

Proteja su Usuario raíz de la cuenta de AWS

1. Inicie sesión [AWS Management Console](#) como propietario de la cuenta seleccionando el usuario root e introduciendo su dirección de Cuenta de AWS correo electrónico. En la siguiente página, escriba su contraseña.

Para obtener ayuda para iniciar sesión con el usuario raíz, consulte [Signing in as the root user](#) en la Guía del usuario de AWS Sign-In .

2. Active la autenticación multifactor (MFA) para el usuario raíz.

Para obtener instrucciones, consulte [Habilitar un dispositivo MFA virtual para el usuario Cuenta de AWS raíz \(consola\)](#) en la Guía del usuario de IAM.

Creación de un usuario con acceso administrativo

1. Activar IAM Identity Center.

Consulte las instrucciones en [Activar AWS IAM Identity Center](#) en la Guía del usuario de AWS IAM Identity Center .

2. En IAM Identity Center, conceda acceso administrativo a un usuario.

Para ver un tutorial sobre su uso Directorio de IAM Identity Center como fuente de identidad, consulte [Configurar el acceso de los usuarios con la configuración predeterminada Directorio de IAM Identity Center en la](#) Guía del AWS IAM Identity Center usuario.

Iniciar sesión como usuario con acceso de administrador

- Para iniciar sesión con el usuario de IAM Identity Center, utilice la URL de inicio de sesión que se envió a la dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

Para obtener ayuda para iniciar sesión con un usuario del Centro de identidades de IAM, consulte [Iniciar sesión en el portal de AWS acceso](#) en la Guía del AWS Sign-In usuario.

Concesión de acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos de privilegios mínimos.

Para conocer las instrucciones, consulte [Create a permission set](#) en la Guía del usuario de AWS IAM Identity Center .

2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

Para conocer las instrucciones, consulte [Add groups](#) en la Guía del usuario de AWS IAM Identity Center .

Conceder acceso a Amazon Location Service

De forma predeterminada, los usuarios que no son administradores no tienen permisos. Antes de que puedan acceder a Amazon Location, debe conceder el permiso adjuntando una política de IAM con permisos específicos. Asegúrese de seguir el principio de privilegios mínimos al conceder acceso a los recursos.

Note

Para obtener información sobre cómo dar acceso a los usuarios no autenticados a la funcionalidad de Amazon Location Service (por ejemplo, en una aplicación basada en la web), consulte [Concesión de acceso a Amazon Location Service](#).

El siguiente ejemplo de política otorga a un usuario permiso para acceder a todas las operaciones de Amazon Location. Para obtener más ejemplos, consulte [Ejemplos de políticas basadas en identidad para Amazon Location Service](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "geo:*"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}
```

Para dar acceso, agregue permisos a los usuarios, grupos o roles:

- Usuarios y grupos en AWS IAM Identity Center:

Cree un conjunto de permisos. Siga las instrucciones de [Creación de un conjunto de permisos](#) en la Guía del usuario de AWS IAM Identity Center .

- Usuarios administrados en IAM a través de un proveedor de identidades:

Cree un rol para la federación de identidades. Siga las instrucciones descritas en [Creación de un rol para un proveedor de identidad de terceros \(federación\)](#) en la Guía del usuario de IAM.

- Usuarios de IAM:

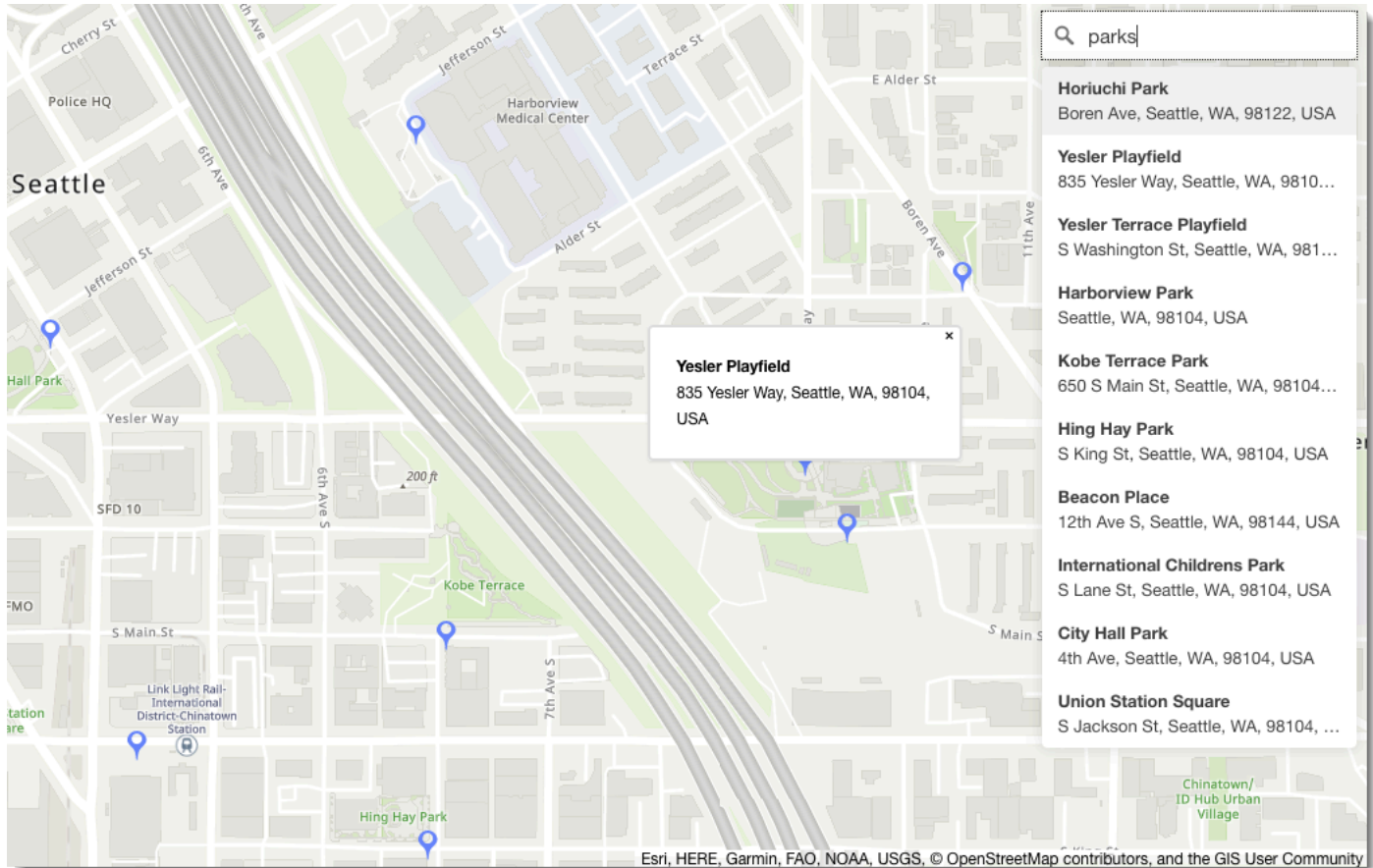
- Cree un rol que el usuario pueda aceptar. Siga las instrucciones descritas en [Creación de un rol para un usuario de IAM](#) en la Guía del usuario de IAM.

- (No recomendado) Adjunte una política directamente a un usuario o añada un usuario a un grupo de usuarios. Siga las instrucciones descritas en [Adición de permisos a un usuario \(consola\)](#) de la Guía del usuario de IAM.

Al crear aplicaciones que utilizan Amazon Location Service, es posible que necesite que algunos usuarios tengan acceso no autenticado. Para ver estos casos de uso, consulte [Habilitar el acceso no autenticado mediante Amazon Cognito](#).

Uso de Amazon Location Maps en su aplicación

Los mapas de ubicación de Amazon son rentables e interactivos. Puede sustituir un mapa existente en su aplicación para ahorrar dinero o agregar uno nuevo para mostrar visualmente los datos basados en la ubicación, como la ubicación de su tienda.



Amazon Location Service le permite elegir un proveedor de datos para las operaciones de mapas mediante la creación y configuración de un recurso de mapa. El recurso de mapa configura el proveedor de datos y el estilo que se utiliza para representar el mapa.

Después de crear el recurso, puede enviar las solicitudes directamente mediante el SDK AWS o mediante una biblioteca creada específicamente para representar los mapas de su entorno.

Note

Para obtener una descripción general de los conceptos de mapas, consulte [Mapas](#).

Temas

- [Requisitos previos](#)
- [Mostrar un mapa en su aplicación](#)
- [Dibujar características de datos en un mapa](#)
- [Establecer las extensiones de un mapa mediante MapLibre](#)
- [Administrar sus recursos de mapas](#)

Requisitos previos

Antes de mostrar un mapa en su aplicación, siga los pasos previos:

Temas

- [Crear un recurso de mapa](#)
- [Autenticación de sus solicitudes](#)

Crear un recurso de mapa

Para usar un mapa en su aplicación, debe tener un recurso de mapa que especifique el estilo del mapa y el proveedor de datos que se utilizarán en los mapas.

Note

Si su aplicación rastrea o enruta los activos que utiliza en su empresa, como vehículos de reparto o empleados, no debe utilizar Esri como proveedor de geolocalización. Consulte la sección 82 de los [Términos de servicio de AWS](#) para obtener más detalles.

Puede crear un recurso de mapa utilizando la consola de Amazon Location ServiceAWS CLI, el o las API de ubicación de Amazon.

Console

Para crear un recurso de mapa mediante la consola de Amazon Location Service

1. En la consola de Amazon Location, en la página de [Mapas](#), seleccione Crear mapa para obtener una vista previa de los estilos de mapa.
2. Agregue un nombre y una descripción para el nuevo recurso de mapa.
3. Elija un estilo de mapa.

Note

Si su aplicación rastrea o enruta los activos que utiliza en su empresa, como vehículos de reparto o empleados, no debe utilizar Esri como proveedor de geolocalización. Consulte la sección 82 de los [Términos de servicio de AWS](#) para obtener más detalles.

4. Elija uno de [Opiniones políticas](#) que desee utilizar.
5. Acepte los Términos y condiciones de Amazon Location y, a continuación, seleccione Crear mapa. Puede interactuar con el mapa que haya elegido: acercar, alejar o desplazarte en cualquier dirección.
6. Para que sus usuarios puedan cambiar de estilo (por ejemplo, para que puedan cambiar entre imágenes de satélite y estilos vectoriales), debe crear un recurso de mapa para cada estilo.

Puede eliminar los recursos con estilos de mapa que no desee usar en la [página de inicio de Mapas](#) de la consola.

API

Para crear un recurso de mapa utilizando las API de Amazon Location

Use la operación [CreateMap](#) de las API de Amazon Location.

El siguiente ejemplo es una solicitud de API para crear un recurso de mapa llamado *ExampleMap* mediante el estilo de *VectorEsriStreets* mapa.

```
POST /maps/v0/maps HTTP/1.1
Content-type: application/json

{
  "Configuration": {
    "Style": "VectorEsriStreets"
  },
  "MapName": "ExampleMap"
}
```

Note

Si su aplicación rastrea o enruta los activos que utiliza en su empresa, como vehículos de reparto o empleados, no debe utilizar Esri como proveedor de geolocalización. Consulte la sección 82 de los [Términos de servicio de AWS](#) para obtener más detalles.

AWS CLI

Para crear un recurso de mapa mediante comandos AWS CLI

Utilice el comando [create-map](#).

En el siguiente ejemplo, se crea un recurso de mapa denominado *ExampleMap* usar *VectorEsriStreets* como estilo de mapa.

```
aws location \
  create-map \
  --configuration Style="VectorEsriStreets" \
  --map-name "ExampleMap"
```

Note

Si su aplicación rastrea o enruta los activos que utiliza en su empresa, como vehículos de reparto o empleados, no debe utilizar Esri como proveedor de geolocalización. Consulte la sección 82 de los [Términos de servicio de AWS](#) para obtener más detalles.

Autenticación de sus solicitudes

Una vez que haya creado un recurso de mapa y esté listo para empezar a incorporar funciones de ubicación a su aplicación, tendrá que elegir cómo autenticará sus solicitudes.

Note

La mayoría de las aplicaciones front-end de mapas requieren acceso no autenticado a los mapas u otras características de Amazon Location Service. Dependiendo de su aplicación, es posible que desee utilizar AWS Signature v4 para autenticar las solicitudes, o puede utilizar las claves de la API de Amazon Cognito o Amazon Location para un uso no autenticado. Para obtener más información sobre estas dos opciones, consulte [Concesión de acceso a Amazon Location Service](#).

Mostrar un mapa en su aplicación

En esta sección se proporcionan tutoriales sobre cómo utilizar las herramientas de representación de mapas para mostrar un mapa en su aplicación móvil o web al utilizar las API de ubicación de Amazon. Como se mencionó en el [Cómo utilizar Amazon Location Service](#) tema, puede elegir entre varias bibliotecas para renderizar mapas con Amazon Location, incluidas Amplify y MapLibre Tangram.

Realice una de las siguientes acciones para mostrar un mapa en su aplicación:

- La forma más directa de mostrar un mapa en sus aplicaciones web y móviles es utilizarlas. MapLibre Puedes seguir los [MapLibre tutoriales](#) o incluso el [tutorial de inicio rápido](#) para aprender a usarlo MapLibre.
- Si usted esAWS Amplify desarrollador, quizás quiera usar el SDK de Amplify Geo. Para obtener más información, siga el [Tutorial de Amplify](#).
- Si ya es usuario de Tangram y quiere seguir utilizándolo para renderizar su mapa mientras se mudas a Amazon Location Service, siga el [Tutorial de Tangram](#).

Temas

- [Uso de la MapLibre biblioteca con Amazon Location Service](#)
- [Uso de la biblioteca Amplify con Amazon Location Service](#)
- [Uso de Tangram con Amazon Location Service](#)

Uso de la MapLibre biblioteca con Amazon Location Service

En los siguientes tutoriales, se explica cómo utilizar la MapLibre biblioteca con Amazon Location.

Temas

- [Uso de MapLibre GL JS con Amazon Location Service](#)
- [Uso del SDK MapLibre nativo para Android con Amazon Location Service](#)
- [Uso del SDK MapLibre nativo para iOS con Amazon Location Service](#)

Uso de MapLibre GL JS con Amazon Location Service

Utilice [MapLibre GL JS](#) para incrustar mapas del lado del cliente en las aplicaciones web.

MapLibre GL JS es una JavaScript biblioteca de código abierto compatible con los estilos y mosaicos que proporciona la API de mapas de Amazon Location Service. Puedes integrar MapLibre GL JS en un HTML básico o en una JavaScript aplicación para incorporar mapas personalizables y adaptables del lado del cliente.

Este tutorial describe cómo integrar MapLibre GL JS con Amazon Location dentro de un HTML y una JavaScript aplicación básicos. Las mismas bibliotecas y técnicas que se presentan en este tutorial también se aplican a los marcos de trabajo, como [React](#) y [Angular](#).

La aplicación de muestra de este tutorial está disponible como parte del repositorio de muestras de Amazon Location Service en [GitHub](#).

Creación de la aplicación: andamiaje

Este tutorial crea una aplicación web que se utiliza JavaScript para crear un mapa en una página HTML.

Comience por crear una página HTML (`index.html`) que incluya el contenedor del mapa:

- Introduzca un elemento `div` con un `id` de `map` para aplicar las dimensiones del mapa a la vista del mapa. Las dimensiones se heredan de la ventana de visualización.

```
<html>
  <head>
    <style>
      body {
        margin: 0;
```

```
    }

    #map {
      height: 100vh; /* 100% of viewport height */
    }
  </style>
</head>
<body>
  <!-- map container -->
  <div id="map" />
</body>
</html>
```

Creación de la aplicación: adición de dependencias

Agregue las siguientes dependencias a su aplicación:

- MapLibre GL JS (v3.x) y su CSS asociado.
- Amazon Location [JavaScript Ayudante de autenticación](#).

```
<!-- 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>
```

Esto crea una página vacía con el contenedor del mapa.

Creación de la aplicación: configuración

Para configurar la aplicación mediante: JavaScript

1. Introduzca los nombres e identificadores de sus recursos.

```
// Cognito Identity Pool ID
const identityPoolId = "us-east-1:54f2ba88-9390-498d-aaa5-0d97fb7ca3bd";
```

```
// Amazon Location Service Map name
const mapName = "ExampleMap";
```

2. Cree una instancia de un proveedor de credenciales mediante el grupo de identidades no autenticadas que creó en [Uso de mapas: paso 2, configurar la autenticación](#). Lo colocaremos en una función llamada `initializeMap`, que también contendrá otro código de inicialización del mapa, que se agregará en el siguiente paso

```
// 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
}
```

Creación de la aplicación: inicialización del mapa

Para que el mapa se muestre después de cargar la página, debe inicializarlo. Puede ajustar la ubicación inicial del mapa, agregar controles adicionales y superponer datos.

```
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

Debe proporcionar una marca denominativa o una atribución de texto para cada proveedor de datos que utilice, ya sea en su solicitud o en su documentación. Las cadenas de atribución se incluyen en la respuesta del descriptor de estilo, debajo de las `sources.esri.attribution` `clavessources.here.attribution`, `ysources.grabmaptiles.attribution`. MapLibre GL JS proporcionará la atribución automáticamente. Cuando utilice los recursos de Amazon Location con [proveedores de datos](#), asegúrese de leer los [términos y condiciones del servicio](#).

Ejecución de la aplicación

Puede ejecutar esta aplicación de ejemplo utilizándola en un servidor web local o abriéndola en un navegador.

Para usar un servidor web local, puede usar `npx`, ya que se instala como parte de Node.js. Puede usar `npx serve` desde el mismo directorio que `index.html`. Esto activa la aplicación en `localhost:5000`.

Note

Si la política que ha creado para su rol no autenticado de Amazon Cognito incluye una condición `referer`, es posible que no pueda realizar pruebas con direcciones URL `localhost:.` En este caso, puede realizar la prueba con un servidor web que proporcione una URL incluida en su política.

Después de completar el tutorial, la aplicación final será similar al siguiente ejemplo.

```
<!-- index.html -->
<html>
  <head>
    <link href="https://unpkg.com/maplibre-gl@3.x/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@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>
```

Al ejecutar esta aplicación, se muestra un mapa a pantalla completa con el estilo de mapa que haya elegido. Este ejemplo está disponible en el repositorio de muestras de Amazon Location Service en [GitHub](#).

Uso del SDK MapLibre nativo para Android con Amazon Location Service

Usa el SDK [MapLibrenativo](#) para integrar mapas interactivos en tus aplicaciones de Android.

El SDK MapLibre nativo para Android es una biblioteca basada en [Mapbox Native](#) y es compatible con los estilos y mosaicos que proporciona la API de mapas de Amazon Location Service. Puedes integrar MapLibre Native SDK for Android para integrar vistas de mapas interactivos con mapas vectoriales escalables y personalizables en tus aplicaciones de Android.

En este tutorial se describe cómo integrar el SDK MapLibre nativo para Android con Amazon Location. La aplicación de muestra de este tutorial está disponible como parte del repositorio de muestras de Amazon Location Service en [GitHub](#).

Creación de la aplicación: inicialización

Para inicializar la aplicación:

1. Cree un nuevo proyecto de Android Studio a partir de la plantilla Empty Activity.
2. Asegúrese de que Kotlin esté seleccionado como idioma del proyecto.
3. Seleccione un SDK mínimo de API 14: Android 4.0 (Ice Cream Sandwich) o posterior.
4. Abra la estructura del proyecto y, a continuación, vaya a Archivo > Estructura del proyecto... para elegir la sección Dependencias.
5. Con <Todos los módulos> seleccionado, pulse el botón + para agregar una nueva dependencia de biblioteca.
6. Agregue la versión 2.20.0 o posterior del SDK para Android de AWS. Por ejemplo:
`com.amazonaws:aws-android-sdk-core:2.20.0`
7. Agrega la versión 9.4.0 o posterior de MapLibre Native SDK for Android. Por ejemplo:
`org.maplibre.gl:android-sdk:9.4.0`
8. En el nivel de proyecto de tu archivo build.gradle, agrega el siguiente repositorio de maven para acceder a los paquetes de Android: MapLibre

```
allprojects {  
    repositories {
```

```

        // Retain your existing repositories
        google()
        jcenter()

        // Declare the repositories for MapLibre
        mavenCentral()
    }
}

```

Creación de la aplicación: configuración

Para configurar la aplicación con los recursos y la región AWS:

```

<?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>

```

Creación de la aplicación: diseño de actividades

Editar `app/src/main/res/layout/activity_main.xml`:

- Agregue un `MapView`, que renderiza el mapa. Esto también establecerá el punto central inicial del mapa.
- Agrega un `TextView`, que muestra la atribución.

```

<?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

Debe proporcionar una marca denominativa o una atribución de texto para cada proveedor de datos que utilice, ya sea en su solicitud o en su documentación. Los string de atribución se incluyen en la respuesta del descriptor de estilo, debajo de las claves `sources.esri.attribution`, `sources.here.attribution`, y `source.grabmaptiles.attribution`. Cuando utilice los recursos de Amazon Location con [proveedores de datos](#), asegúrese de leer los [términos y condiciones del servicio](#).

Creación de la aplicación: solicite la transformación

Cree una clase denominada `SigV4Interceptor` para interceptar las solicitudes de AWS y fírmelas con [Signature Version 4](#). Se registrará en el cliente HTTP utilizado para obtener los recursos del mapa cuando se cree la actividad principal.

```
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()
    }
}
}
```

Creación de la aplicación: Actividad principal

La actividad principal se encarga de inicializar las vistas que se mostrarán a los usuarios. Esto implica:

- Creación de una instancia de Amazon Cognito. `CredentialsProvider`
- Registro del interceptor de firmas de la versión 4.
- Configurar el mapa apuntándolo a un descriptor de estilo de mapa y mostrando la atribución adecuada.

`MainActivity` también es responsable de reenviar los eventos del ciclo de vida a la vista del mapa, lo que le permite conservar la ventana gráfica activa entre las invocaciones.

```
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()  
}  
}
```

Al ejecutar esta aplicación, se muestra un mapa a pantalla completa con el estilo que elija. Este ejemplo está disponible como parte del repositorio de muestras de Amazon Location Service en [GitHub](#).

Uso del SDK MapLibre nativo para iOS con Amazon Location Service

Usa [MapLibreNative SDK for iOS para](#) incrustar mapas del lado del cliente en las aplicaciones iOS.

El SDK MapLibre nativo para iOS es una biblioteca basada en [Mapbox GL Native](#) y es compatible con los estilos y mosaicos que proporciona la API de mapas de Amazon Location Service. Puede integrar el SDK MapLibre nativo para iOS para integrar vistas de mapas interactivas con mapas vectoriales escalables y personalizables en sus aplicaciones iOS.

En este tutorial se describe cómo integrar el SDK MapLibre nativo para iOS con Amazon Location. La aplicación de muestra de este tutorial está disponible como parte del repositorio de muestras de Amazon Location Service en [GitHub](#).

Creación de la aplicación: inicialización

Para inicializar la aplicación:

1. Crear un nuevo proyecto Xcode a partir de la plantilla App.
2. Seleccione SwiftUI para su interfaz.
3. Seleccione la aplicación SwiftUI para su ciclo de vida.
4. Seleccione Swift como su idioma.

Añadir MapLibre dependencias mediante Swift Packages

Para agregar una dependencia de paquete a su proyecto de Xcode:

1. Vaya a Archivo > Paquetes Swift > Agregar dependencia de un paquete.
2. Introduzca la URL del repositorio: **<https://github.com/maplibre/maplibre-gl-native-distribution>**

Note

Para obtener más información sobre Paquetes Swift, consulte [Cómo agregar dependencias de paquetes a su aplicación](#) en Apple.com

3. En tu terminal, instala CocoaPods:

```
sudo gem install cocoapods
```

4. Navega hasta el directorio de proyectos de tu aplicación e inicializa el Podfile con el administrador de CocoaPods paquetes:

```
pod init
```

5. Abra el Podfile para agregarlo AWSCore como dependencia:

```
platform :ios, '12.0'  
  
target 'Amazon Location Service Demo' do  
  use_frameworks!  
  
  pod 'AWSCore'  
end
```

6. Descarga e instalación de dependencias:

```
pod install --repo-update
```

7. Abre el espacio de trabajo de Xcode que creó: CocoaPods

```
xed .
```

Creación de la aplicación: configuración

Agregue las siguientes claves y valores a Info.plist para configurar la aplicación:

Clave	Valor
AWSRegion	us-east-1
IdentityPoolId	us-east-1:54f2ba88-9390-498d-aaa5-0d97fb7ca3bd
MapName	ExampleMap

Creación de la aplicación: diseño ContentView

Para renderizar el mapa, edite ContentView.swift:

- Agregue un `MapView`, que renderice el mapa.
- Agregue un `TextField`, que muestre la atribución.

Esto también establece el punto central inicial del mapa.

```
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

Debe proporcionar una marca denominativa o una atribución de texto para cada proveedor de datos que utilice, ya sea en su solicitud o en su documentación. Los string de atribución se incluyen en la respuesta del descriptor de estilo, debajo de las claves `sources.esri.attribution`, `sources.here.attribution`, y

`source.grabmaptiles.attribution`. Cuando utilice los recursos de Amazon Location con [proveedores de datos](#), asegúrese de leer los [términos y condiciones del servicio](#).

Creación de la aplicación: solicite la transformación

Cree un nuevo archivo Swift denominado `AWSSignatureV4Delegate.swift` que contiene la siguiente definición de clase para interceptar las solicitudes de AWS y fírmelas con [Signature Version 4](#). Se asignará una instancia de esta clase como delegado de almacenamiento fuera de línea, que también se encargará de reescribir las URL en la vista de mapa.

```
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
}
```

```
}
```

Creación de la aplicación: vista de mapa

La vista de mapa se encarga de inicializar una instancia `AWSSignatureV4Delegate` y configurar la instancia subyacente `MGLMapView`, que recupera los recursos y renderiza el mapa. También gestiona la propagación de las cadenas de atribución desde la fuente del descriptor de estilo hasta la `ContentView`.

Cree un nuevo archivo Swift denominado `MapView.swift` que contenga la struct siguiente definición:

```
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
    }
}
```

Al ejecutar esta aplicación, se muestra un mapa a pantalla completa con el estilo que elija. Este ejemplo está disponible como parte del repositorio de muestras de Amazon Location Service en [GitHub](#).

Uso de la biblioteca Amplify con Amazon Location Service

En los siguientes tutoriales, se explica cómo utilizar AWS Amplify con Amazon Location. Amplify usa MapLibre GL JS para renderizar mapas en su aplicación JavaScript basada.

Amplify es un conjunto de bibliotecas de clientes de código abierto que proporcionan interfaces para diferentes categorías de servicios, incluida Amplify Geo, que funciona con Amazon Location Service. [Obtenga más información sobre la JavaScript biblioteca AWS Amplify Geo](#).

Note

En este tutorial se supone que ya ha seguido los pasos de [Uso de mapas: para agregar un mapa a la aplicación](#).

Creación de la aplicación: andamiaje

En este tutorial se crea una aplicación web que JavaScript se utiliza para crear un mapa en una página HTML.

Comience por crear una página HTML (`index.html`) que incluya el contenedor del mapa:

- Introduzca un elemento `div` con un `id` de `map` para aplicar las dimensiones del mapa a la vista del mapa. Las dimensiones se heredan de la ventana de visualización.

```
<html>
  <head>
    <style>
      body { margin: 0; }
      #map { height: 100vh; } /* 100% of viewport height */
    </style>
  </head>
```

```
<body>
  <!-- map container -->
  <div id="map" />
</body>
</html>
```

Creación de la aplicación: adición de dependencias

Agregue las siguientes dependencias a su aplicación:

- AWS Amplify mapotecas y geotecas.
- AWS Amplify biblioteca principal.
- AWS Amplify biblioteca de autenticación.
- AWS Amplify hoja de estilos.

```
<!-- 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+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>
<script>
  // application-specific code
```

```
</script>
```

Esto crea una página vacía con el contenedor del mapa.

Creación de la aplicación: configuración

Para configurar la aplicación mediante JavaScript:

1. Introduzca los identificadores del grupo de identidades no autenticadas que creó en [Uso de mapas: paso 2, Configurar la autenticación](#).

```
// 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. Configure AWS Amplify para usar los recursos que ha creado, incluidos el grupo de identidades y el recurso de mapa (que se muestran aquí con el nombre predeterminado de `explore.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,
    },
  }
})
```

```
});
```

Creación de la aplicación: inicialización del mapa

Para que el mapa se muestre después de cargar la página, debe inicializarlo. Puede ajustar la ubicación inicial del mapa, agregar controles adicionales y superponer datos.

```
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

Debe proporcionar una marca denominativa o una atribución de texto para cada proveedor de datos que utilice, ya sea en su solicitud o en su documentación. Los string de atribución se incluyen en la respuesta del descriptor de estilo, debajo de las claves `sources.esri.attribution`, `sources.here.attribution`, y `sources.grabmaptiles.attribution`. Amplify proporcionará automáticamente la atribución. Cuando utilice los recursos de Amazon Location con [proveedores de datos](#), asegúrese de leer los [términos y condiciones del servicio](#).

Ejecución de la aplicación

Puede ejecutar esta aplicación de ejemplo utilizándola en un servidor web local o abriéndola en un navegador.

Para usar un servidor web local, puede usar npx, instalado como parte de Node.js, o cualquier otro servidor web de su elección. Para usar npx, escriba `npx serve` desde el mismo directorio que `index.html`. Esto activa la aplicación en `localhost:5000`.

Note

Si la política que ha creado para su rol no autenticado de Amazon Cognito incluye una condición `referer`, es posible que no pueda realizar pruebas con direcciones URL `localhost:.` En este caso, puede realizar la prueba con un servidor web que proporcione una URL incluida en su política.

Después de completar el tutorial, la aplicación final será similar al siguiente ejemplo.

```
<html>
  <head>
    <!-- CSS dependencies -->
    <link href="https://cdn.amplify.aws/packages/maplibre-
gl/1.15.2/maplibre-gl.css" rel="stylesheet" integrity="sha384-
DrPVD9GufrxGb7kWwRv0CywPTmfvbK0Z5i5pN7urmIThew0zXKTME+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-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,
        }
      );
    }
  </script>

```

```
    }
  );

  map.addControl(new maplibregl.NavigationControl(), "top-left");
}

initializeMap();
</script>
</body>
</html>
```

Al ejecutar esta aplicación, se muestra un mapa a pantalla completa con el estilo de mapa que haya elegido. Este ejemplo también se describe en la pestaña Insertar mapa de cualquier página de recursos de mapas de la [consola de Amazon Location Service](#).

Tras completar este tutorial, consulte el tema [Mostrar un mapa](#) de la AWS Amplify documentación para obtener más información, incluida la forma de mostrar los marcadores en el mapa.

Uso de Tangram con Amazon Location Service

Esta sección proporciona los siguientes tutoriales sobre cómo integrar Tangram con Amazon Location.

Important

Los estilos de Tangram de los siguientes tutoriales solo son compatibles con los recursos de mapas de ubicación de Amazon configurados con el estilo `VectorHereContrast`.

A continuación, se muestra un ejemplo de un comando de AWS CLI para crear un nuevo recurso de mapa denominado *TangramExampleMap* mediante el *VectorHereContrast* estilo:

```
aws --region us-east-1 \  
  location \  
  create-map \  
  --map-name "TangramExampleMap" \  
  --configuration "Style=VectorHereContrast"
```

Note

La facturación viene determinada por el uso. Puede incurrir en cargos por el uso de otros servicios de AWS. Para obtener más información, consulte los [precios de Amazon Location Service](#).

Temas

- [Uso de Tangram con Amazon Location Service](#)
- [Uso de Tangram ES para Android con Amazon Location Service](#)
- [Uso de Tangram ES para iOS con Amazon Location Service](#)

Uso de Tangram con Amazon Location Service

[Tangram](#) es un motor de mapeo flexible, diseñado para la representación en tiempo real de mapas 2D y 3D a partir de mosaicos vectoriales. Se puede usar con los estilos diseñados por Mapzen y los mosaicos HERE proporcionados por la API de mapas de Amazon Location Service. Esta guía describe cómo integrar Tangram con Amazon Location en una JavaScript aplicación HTML/ básica, aunque las mismas bibliotecas y técnicas también se aplican cuando se utilizan marcos como React y Angular.

Tangram se basa en [Leaflet](#), una biblioteca de código abierto JavaScript para mapas interactivos aptos para dispositivos móviles. Esto significa que muchos complementos y controles compatibles con Leaflet también funcionan con Tangram.

Los estilos de Tangram diseñados para funcionar con el [esquema de Tilezen](#) son en gran medida compatibles con Amazon Location cuando se utilizan mapas de HERE. Entre ellos se incluyen:

- [Bubble Wrap](#): un estilo de orientación con todas las funciones e íconos útiles para los puntos de interés
- [Cinnabar](#): un diseño clásico, ideal para aplicaciones generales de cartografía
- [Refill](#): un estilo de mapa minimalista diseñado para superposiciones de visualización de datos, inspirado en el popular estilo de Toner de Stamen Design
- [Tron](#): una exploración de las transformaciones de escala en el lenguaje visual de TRON
- [Walkabout](#): un estilo centrado en actividades al aire libre que es perfecto para practicar senderismo o salir a pasear

[Esta guía describe cómo integrar Tangram con Amazon Location en una JavaScript aplicación HTML/básica utilizando el estilo Tangram llamado Bubble Wrap.](#) Este ejemplo está disponible como parte del repositorio de muestras de Amazon Location Service en [GitHub](#).

Si bien otros estilos de Tangram se combinan mejor con mosaicos ráster, que codifican la información del terreno, Amazon Location aún no admite esta función.

Important

Los estilos Tangram del siguiente tutorial sólo son compatibles con los recursos de mapa de Amazon Location configurados con el estilo `VectorHereContrast`.

Creación de la aplicación: andamiaje

La aplicación es una página HTML con la JavaScript que crear el mapa en su aplicación web. Cree una página HTML (`index.html`) y cree el contenedor del mapa:

- Introduzca un elemento `div` con un `id` de mapa para aplicar las dimensiones del mapa a la vista del mapa.
- Las dimensiones se heredan de la ventana de visualización.

```
<html>
  <head>
    <style>
      body {
        margin: 0;
      }

      #map {
        height: 100vh; /* 100% of viewport height */
      }
    </style>
  </head>
  <body>
    <!-- map container -->
    <div id="map" />
  </body>
</html>
```

Creación de la aplicación: adición de dependencias

Agregue la siguiente dependencia:

- Leaflet y su CSS asociado.
- Tangram.
- AWS SDK para 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>
```

Esto crea una página vacía con los requisitos previos necesarios. El siguiente paso le guiará a través de la escritura del JavaScript código de su aplicación.

Creación de la aplicación: configuración

Para configurar su aplicación con sus recursos y credenciales:

1. Introduzca los nombres e identificadores de sus recursos.

```
// 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. Cree una instancia de un proveedor de credenciales mediante el grupo de identidades no autenticadas que creó en [Uso de mapas: paso 2, configurar la autenticación](#). Como se utilizan

credenciales ajenas al flujo de trabajo normal del SDK de AWS, las sesiones caducan al cabo de una hora.

```
// 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. Si bien Tangram le permite anular las URL utilizadas para recuperar los mosaicos, no incluye la posibilidad de interceptar las solicitudes para firmarlas.

Para solucionar este problema, anule `sources.mapzen.url` para que apunte a Amazon Location con un nombre de host sintético `amazon.location`, que será gestionado por un [empleo del servicio](#). El siguiente es un ejemplo de configuración de escena que usa [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,
    },
  },
};
```

Creación de la aplicación: solicite la transformación

Para registrar e inicializar el service worker, cree una función `registerServiceWorker` a la que se pueda llamar antes de inicializar el mapa. Esto registra el JavaScript código proporcionado en un archivo independiente `sw.js` denominado `service worker controllerindex.html`.

Las credenciales se cargan desde Amazon Cognito y se transmiten al empleado del servicio junto con la región para proporcionar la información necesaria para firmar las solicitudes de mosaicos con la [versión 4 de Signature](#).

```
/**
 * 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");
}
}
```

La implementación de Service Worker en `sw.js` escucha los message eventos para detectar los cambios en la configuración de las credenciales y la región. También actúa como un servidor proxy al `fetch` escuchar los eventos. `fetch` los eventos que se `amazon.location` dirijan al nombre de host sintético se reescribirán para que se dirijan a la API Amazon Location correspondiente y se firmarán con 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));
});
```

Para renovar automáticamente las credenciales y enviárselas al empleado del servicio antes de que caduquen, utilice la siguiente función: `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());
}
```

Creación de la aplicación: inicialización del mapa

Para que el mapa se muestre después de cargar la página, debe inicializarlo. Tiene la opción de ajustar la ubicación inicial del mapa, agregar controles adicionales y superponer datos.

Note

Debe proporcionar una marca denominativa o una atribución de texto para cada proveedor de datos que utilice, ya sea en su solicitud o en su documentación. Los string de atribución se incluyen en la respuesta del descriptor de estilo, debajo de las claves `sources.esri.attribution`, `sources.here.attribution`, y `source.grabmaptiles.attribution`.

Como Tangram no solicita estos recursos y solo es compatible con los mapas de HERE, use "© 2020 HERE". Cuando utilice los recursos de Amazon Location con [proveedores de datos](#), asegúrese de leer los [términos y condiciones del servicio](#).

```
/**
 * 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();
```

Ejecución de la aplicación

Para ejecutar este ejemplo, puede:

- Utilizar un host que admita HTTPS,
- Utilizar un servidor web local para cumplir con las restricciones de seguridad de los trabajadores del servicio.

Para usar un servidor web local, puede usar `npx`, porque está instalado como parte de Node.js. Puede usar `npx serve` desde el mismo directorio que `index.html` y `sw.js`. Esto sirve a la aplicación en localhost:5000.

El archivo es el siguiente: `index.html`

```
<!-- index.html -->
<html>
  <head>
    <link
      rel="stylesheet"
      href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css"
      integrity="sha512-xodZBNTC5n17Xt2atTPuE1HxjVMSvLVW9ocqUKLsCC5CXdbqCmb1Ash0MAS6/
keqq/sMZM19scR4PsZChSR7A=="
      crossorigin=""
    />
    <style>
      body {
        margin: 0;
      }
    </style>
  </head>
  <body>
    <div id="map">
      <img alt="Map showing a location in the Pacific Northwest." data-bbox="100 100 900 900"/>
    </div>
  </body>
</html>
```



```
#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>
```

El archivo es el siguiente: `sw.js`

```
// 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));
});
```

```
});
```

Este ejemplo está disponible como parte del repositorio de muestras de Amazon Location Service en [GitHub](#).

Uso de Tangram ES para Android con Amazon Location Service

[Tangram ES](#) es una biblioteca de C++ para renderizar mapas 2D y 3D a partir de datos vectoriales utilizando OpenGL ES. Es la contraparte nativa de [Tangram](#).

Los estilos de Tangram diseñados para funcionar con el [esquema de Tilezen](#) son en gran medida compatibles con Amazon Location cuando se utilizan mapas de HERE. Entre ellos se incluyen:

- [Bubble Wrap](#): un estilo de orientación con todas las funciones e íconos útiles para los puntos de interés.
- [Cinnabar](#): un diseño clásico, ideal para aplicaciones generales de cartografía.
- [Refill](#): un estilo de mapa minimalista diseñado para superposiciones de visualización de datos, inspirado en el popular estilo de Toner de Stamen Design.
- [Tron](#): una exploración de las transformaciones de escala en el lenguaje visual de TRON.
- [Walkabout](#): un estilo centrado en actividades al aire libre que es perfecto para practicar senderismo o salir a pasear.

Esta guía describe cómo integrar Tangram ES para Android con Amazon Location utilizando el estilo Tangram llamado Cinnabar. Este ejemplo está disponible como parte del repositorio de muestras de Amazon Location Service en [GitHub](#).

Mientras que otros estilos de Tangram se acompañan mejor de mosaicos rasterizados, que codifican la información del terreno, esta característica aún no es compatible con Amazon Location.


Important

Los estilos Tangram del siguiente tutorial sólo son compatibles con los recursos de mapa de Amazon Location configurados con el estilo `VectorHereContrast`.

Creación de la aplicación: inicialización

Para inicializar la aplicación:

1. Cree un nuevo proyecto de Android Studio a partir de la plantilla Empty Activity.
2. Asegúrese de que Kotlin esté seleccionado como idioma del proyecto.
3. Seleccione un SDK mínimo de API 16: Android 4.1 (Jelly Bean) o posterior.
4. Abra la estructura del proyecto para seleccionar Archivo, Estructura del proyecto... y elija la sección Dependencias.
5. Con <Todos los módulos> seleccionado, pulse el botón + para agregar una nueva dependencia de biblioteca.
6. Agregue la versión 2.19.1 o posterior del SDK para Android de AWS. Por ejemplo:
`com.amazonaws:aws-android-sdk-core:2.19.1`
7. Agregue Tangram, versión 0.13.0 o posterior. Por ejemplo:
`com.mapzen.tangram:tangram:0.13.0`.

 Note

Si busca Tangram: `com.mapzen.tangram:tangram:0.13.0` se generará un mensaje diciendo que “no se encuentra”, pero si selecciona Aceptar, podrá agregarlo.

Creación de la aplicación: configuración

Para configurar la aplicación con los recursos y la región AWS:

1. Cree `app/src/main/res/values/configuration.xml`.
2. Introduzca los nombres e identificadores de sus recursos, así como AWS la Región en la que fueron creados:

```
<?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>
```

Creación de la aplicación: diseño de actividades

Editar `app/src/main/res/layout/activity_main.xml`:

- Agregue un `MapView`, que renderiza el mapa. Esto también establecerá el punto central inicial del mapa.
- Agrega un `TextView`, que muestra la atribución.

Esto también establecerá el punto central inicial del mapa.

Note

Debe proporcionar una marca denominativa o una atribución de texto para cada proveedor de datos que utilice, ya sea en su solicitud o en su documentación. Los string de atribución se incluyen en la respuesta del descriptor de estilo, debajo de las claves `sources.esri.attribution`, `sources.here.attribution`, y `source.grabmaptiles.attribution`.

Como Tangram no solicita estos recursos y solo es compatible con los mapas de HERE, use "© 2020 HERE". Cuando utilice los recursos de Amazon Location con [proveedores de datos](#), asegúrese de leer los [términos y condiciones del servicio](#).

```
<?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>
```

Creación de la aplicación: solicite la transformación

Cree una clase denominada `SigV4Interceptor` para interceptar AWS las solicitudes y fírmelas con [Signature Version 4](#). Se registrará en el cliente HTTP utilizado para obtener los recursos del mapa cuando se cree la actividad principal.

```
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()
    }
}
```

Creación de la aplicación: Actividad principal

La actividad principal se encarga de inicializar las vistas que se mostrarán a los usuarios. Esto implica:

- Creación de una instancia de Amazon Cognito. `CredentialsProvider`
- Registro del interceptor de firmas de la versión 4.

- Configurar el mapa apuntando a un estilo de mapa, anulando las URL de los mosaicos y mostrando la atribución adecuada.

MainActivity también es responsable de reenviar los eventos del ciclo de vida a la vista del mapa.

```
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()
    }
}
```

Al ejecutar esta aplicación, se muestra un mapa a pantalla completa con el estilo que elija. Este ejemplo está disponible como parte del repositorio de muestras de Amazon Location Service en [GitHub](#).

Uso de Tangram ES para iOS con Amazon Location Service

[Tangram ES](#) es una biblioteca de C++ para renderizar mapas 2D y 3D a partir de datos vectoriales utilizando OpenGL ES. Es la contraparte nativa de [Tangram](#).

Los estilos de Tangram diseñados para funcionar con el [esquema de Tilezen](#) son en gran medida compatibles con Amazon Location cuando se utilizan mapas de HERE. Entre ellos se incluyen:

- [Bubble Wrap](#): un estilo de orientación con todas las funciones e íconos útiles para los puntos de interés
- [Cinnabar](#): un diseño clásico, ideal para aplicaciones generales de cartografía
- [Refill](#): un estilo de mapa minimalista diseñado para superposiciones de visualización de datos, inspirado en el popular estilo de Toner de Stamen Design
- [Tron](#): una exploración de las transformaciones de escala en el lenguaje visual de TRON
- [Walkabout](#): un estilo centrado en actividades al aire libre que es perfecto para practicar senderismo o salir a pasear

Esta guía describe cómo integrar Tangram ES para iOS con Amazon Location utilizando el estilo Tangram llamado Cinnabar. Este ejemplo está disponible como parte del repositorio de muestras de Amazon Location Service en [GitHub](#).

Mientras que otros estilos de Tangram se acompañan mejor de mosaicos rasterizados, que codifican la información del terreno, esta característica aún no es compatible con Amazon Location.

⚠ Important

Los estilos Tangram del siguiente tutorial sólo son compatibles con los recursos de mapa de Amazon Location configurados con el estilo `VectorHereContrast`.

Creación de la aplicación: inicialización

Para inicializar la aplicación:

1. Crear un nuevo proyecto Xcode a partir de la plantilla App.
2. Seleccione SwiftUI para su interfaz.
3. Seleccione la aplicación SwiftUI para su ciclo de vida.
4. Seleccione Swift como su idioma.

Creación de la aplicación: agregar dependencias

Para añadir dependencias, puede utilizar un administrador de dependencias, como [CocoaPods](#):

1. En tu terminal, instala CocoaPods:

```
sudo gem install cocoapods
```

2. Navega hasta el directorio de proyectos de tu aplicación e inicializa el Podfile con el administrador de CocoaPods paquetes:

```
pod init
```

3. Abra el Podfile para agregar AWSCore y Tangram-es como dependencias:

```
platform :ios, '12.0'

target 'Amazon Location Service Demo' do
  use_frameworks!

  pod 'AWSCore'
```

```
pod 'Tangram-es'  
end
```

4. Descarga e instalación de dependencias:

```
pod install --repo-update
```

5. Abre el espacio de trabajo de Xcode que creó: CocoaPods

```
xed .
```

Creación de la aplicación: configuración

Agregue las siguientes claves y valores a Info.plist para configurar la aplicación y deshabilitar la telemetría:

Clave	Valor
AWSRegion	us-east-1
IdentityPoolId	us-east-1:54f2ba88-9390-498d-aaa5-0d97fb7ca3bd
MapName	ExampleMap
SceneURL	https://www.nextzen.org/carto/cinnabar-style/9/cinnabar-style.zip

Creación de la aplicación: diseño ContentView

Para renderizar el mapa, edite ContentView.swift:

- Agregue un MapView, que renderice el mapa.
- Agregue un TextField, que muestre la atribución.

Esto también establece el punto central inicial del mapa.

Note

Debe proporcionar una marca denominativa o una atribución de texto para cada proveedor de datos que utilice, ya sea en su solicitud o en su documentación. Los string de atribución se incluyen en la respuesta del descriptor de estilo, debajo de las claves `sources.esri.attribution`, `sources.here.attribution`, y `source.grabmaptiles.attribution`. Cuando utilice los recursos de Amazon Location con [proveedores de datos](#), asegúrese de leer los [términos y condiciones del servicio](#).

```
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()
    }
}
```


Creación de la aplicación: solicite la transformación

Cree un nuevo archivo Swift denominado `AWSSignatureV4URLHandler.swift` que contenga la siguiente definición de clase para interceptar las solicitudes de AWS y firmarlas con [Signature Version 4](#). Esto se registrará como un controlador de URL en Tangram MapView.

```
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)
}
}
```

Creación de la aplicación: vista de mapa

La vista de mapa se encarga de inicializar una instancia `AWSSignatureV4Delegate` y configurar la instancia subyacente `MGLMapView`, que recupera los recursos y renderiza el mapa. También gestiona la propagación de las cadenas de atribución desde la fuente del descriptor de estilo hasta la `ContentView`.

Cree un nuevo archivo Swift denominado `MapView.swift` que contenga la struct siguiente definición:

```
import AWSCore
import TangramMap
import SwiftUI

struct MapView: UIViewRepresentable {
    private let mapView: TGMMapView
```

```
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) {
}
}
```

Al ejecutar esta aplicación, se muestra un mapa a pantalla completa con el estilo que elija. Este ejemplo está disponible como parte del repositorio de muestras de Amazon Location Service en [GitHub](#).

Dibujar características de datos en un mapa

Una vez que tenga una aplicación que renderice un mapa, utilizando Amplify MapLibre o Tangram para renderizar el mapa, el siguiente paso natural es dibujar entidades en la parte superior del mapa. Por ejemplo, le conviene representar las ubicaciones de sus clientes como marcadores en el mapa.

En general, puedes usar las [funciones de búsqueda de lugares](#) para buscar ubicaciones a partir de tus datos y, a continuación, MapLibre usar la funcionalidad de Amplify o Tangram para renderizar las ubicaciones.

Para ver ejemplos de renderización de distintos tipos de objetos en un mapa, consulta los siguientes MapLibre ejemplos:

- [Ejemplo: dibujar marcadores](#)
- [Ejemplo: dibuja puntos agrupados](#)
- [Ejemplo: dibuja un polígono](#)

Para ver más ejemplos y tutoriales, consulte [Ejemplos de código y tutoriales para trabajar con Amazon Location Service](#).

Establecer las extensiones de un mapa mediante MapLibre

Hay ocasiones en las que no desea que sus usuarios puedan desplazarse panorámicamente o hacer zoom por todo el mundo. Si utilizas el control MapLibre de mapa, puedes limitar la extensión o los límites del control del mapa con la `maxBounds` opción y restringir el zoom con las opciones `minZoom` `maxZoom`

El siguiente ejemplo de código muestra cómo inicializar el control del mapa para restringir la panorámica a un límite específico (en este caso, la extensión del origen de datos de Grab).

Note

Estos ejemplos están incluidos en JavaScript el contexto del tutorial y funcionan en él. [Crear una aplicación web](#)

```
// Set bounds to Grab data provider region
var bounds = [
    [90.0, -21.943045533438166], // Southwest coordinates
    [146.25, 31.952162238024968] // Northeast coordinates
];

var mlglMap = new maplibregl.Map(
    {
        container: 'map',
        style: mapName,
        maxBounds: bounds // Sets bounds as max
        transformRequest,
    }
);
```

Del mismo modo, puede establecer un nivel de zoom mínimo y máximo para el mapa. Los valores de ambos pueden estar entre 0 y 24, aunque los valores predeterminados son 0 para el zoom mínimo y 22 para el máximo (es posible que los proveedores de datos no proporcionen datos en todos los niveles de zoom). La mayoría de las bibliotecas de mapas lo gestionan automáticamente). El siguiente ejemplo inicializa las `maxZoom` opciones `minZoom` y del control de MapLibre mapa.

```
// Set the minimum and maximum zoom levels
var mlglMap = new maplibregl.Map(
    {
        container: 'map',
        style: mapName,
        maxZoom: 12,
        minZoom: 5,
        transformRequest,
    }
);
```

Tip

El control de MapLibre mapa también permite configurar estas opciones en tiempo de ejecución, en lugar de durante la inicialización, con funciones `get...` y `set...`. Por ejemplo, utilice `getMaxBounds` y `setMaxBounds` para cambiar los límites del mapa en tiempo de ejecución.

Administrar sus recursos de mapas

Puede administrar sus recursos de mapas mediante la consola de Amazon Location, AWS CLI o las API de Amazon Location.

Enumerar los recursos de mapas

Puede ver una lista de sus recursos de mapas mediante la consola de Amazon Location, AWS CLI o las API de Amazon Location.

Console

Para ver una lista de los recursos de mapas existentes mediante la consola de Amazon Location

1. Abra la consola de Amazon Location en <https://console.aws.amazon.com/location/>.
2. Seleccione Mapas en el panel de navegación izquierdo.
3. Consulte una lista de sus recursos de mapas en Mis mapas.

API

Utilice la operación [ListMaps](#) desde las API de Amazon Location Maps.

El siguiente ejemplo es una solicitud de API para obtener una lista de los recursos de mapas de la cuenta de AWS.

```
POST /maps/v0/list-maps
```

A continuación se muestra un ejemplo de respuesta para [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

Utilice el comando [list-map](#).

El siguiente ejemplo es un AWS CLI para obtener una lista de los recursos de mapas de la cuenta de AWS.

```
aws location list-maps
```

Obtener detalles de los recursos de mapas

Puede obtener detalles sobre cualquier recurso de mapas de su cuenta de AWS mediante la consola de Amazon Location, AWS CLI o las API de Amazon Location.

Console

Para ver los detalles de un recurso de mapas mediante la consola de Amazon Location

1. Abra la consola de Amazon Location en <https://console.aws.amazon.com/location/>.
2. Seleccione Mapas en el panel de navegación izquierdo.
3. En Mis mapas, seleccione el enlace con el nombre del recurso de mapas de destino.

API

Utilice la operación [DescribeMap](#) desde las API de Amazon Location Maps.

El siguiente ejemplo es una solicitud de API para obtener los detalles del recurso del mapa *ExampleMap*.

```
GET /maps/v0/maps/ExampleMap
```

A continuación, se muestra un ejemplo de respuesta para [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

Utilice el comando [describe-map](#).

El siguiente ejemplo es AWS CLI para obtener los detalles de los recursos del mapa *ExampleMap*.

```
aws location describe-map \
  --map-name "ExampleMap"
```

Eliminar un recurso de mapa

Puede eliminar un recurso de mapas de su cuenta de AWS mediante la consola de Amazon Location, AWS CLI o las API de Amazon Location.

Warning

Esta operación elimina el recurso de forma permanente.

Console

Para eliminar un recurso de mapa existente mediante la consola de Amazon Location

1. Abra la consola de Amazon Location en <https://console.aws.amazon.com/location/>.
2. Seleccione Mapas en el panel de navegación izquierdo.
3. En la lista Mis mapas, seleccione el mapa de destino de la lista.

4. Elija Eliminar mapa.

API

Utilice la operación [DeleteMap](#) desde las API de Amazon Location Maps.

El siguiente ejemplo es una solicitud de API para eliminar el recurso del mapa *ExampleMap*.

```
DELETE /maps/v0/maps/ExampleMap
```

A continuación se muestra un ejemplo de una respuesta correcta para [DeleteMap](#):

```
HTTP/1.1 200
```

CLI

Utilice el comando [delete-map](#).

El siguiente ejemplo es un AWS CLI comando para eliminar el recurso de mapa *ExampleMap*.

```
aws location delete-map \  
  --map-name "ExampleMap"
```

Búsqueda de datos de ubicación y geolocalización mediante Amazon Location

Amazon Location incluye la posibilidad de buscar los datos de geolocalización o ubicación del proveedor que elija. Hay varios tipos de búsqueda disponibles.

- **Geocodificación:** la geocodificación es el proceso de búsqueda de direcciones, regiones, nombres comerciales u otros puntos de interés en función de la entrada de texto. Devuelve los detalles y la ubicación (en latitud y longitud) de los resultados encontrados.
- **Geocodificación inversa:** la geocodificación inversa le permite encontrar lugares cerca de una ubicación determinada.
- **Autocompletar:** el autocompletado es el proceso de hacer sugerencias automáticas a medida que el usuario escribe una consulta. Por ejemplo, si escriben **Par**, una sugerencia podría ser **Paris, France**.

Amazon Location le permite elegir un proveedor de datos para las operaciones de búsqueda de lugares mediante la creación y configuración de un recurso de índice de ubicación.

Una vez que hayas creado tu recurso, puedes enviar solicitudes mediante el AWS SDK para tu idioma preferido, Amplify, o los puntos finales de la API REST. Puede usar los datos de la respuesta para marcar ubicaciones en un mapa, enriquecer los datos de posición y convertir las posiciones en texto legible para las personas.

Note

Para obtener una descripción general de los conceptos de búsqueda de lugares, consulte [Búsqueda de lugares](#).

Temas

- [Requisitos previos](#)
- [Geocodificación](#)
- [Geocodificación inversa](#)
- [Autocompletar](#)
- [Uso de identificadores de lugares](#)
- [Coloca categorías y filtra los resultados](#)
- [Funciones definidas por el usuario de Amazon Aurora PostgreSQL para Amazon Location Service](#)
- [Administrar sus recursos de indexación de sitios](#)

Requisitos previos

Antes de empezar con la geocodificación, la geocodificación inversa o la búsqueda de lugares, siga estos pasos previos:

Temas

- [Crear un recurso de índice de ubicación](#)
- [Autenticación de sus solicitudes](#)

Crear un recurso de índice de ubicación

Comience por crear un recurso de índice de sitios en su cuenta. AWS

Al crear un recurso de índice de ubicación, puede elegir entre los proveedores de datos disponibles para admitir las consultas de geocodificación, geocodificación inversa y búsquedas:

1. Esri: para obtener más información sobre la cobertura de Esri en la región de su interés, consulte la [cobertura de geocodificación de Esri](#) en la documentación de Esri.
2. Tecnologías HERE: para obtener más información sobre la cobertura de HERE en la región que le interese, consulte la [cobertura de geocodificación de HERE](#) en la documentación de HERE.
3. Grab: Grab proporciona datos solo en el sudeste asiático. Para obtener más información sobre la cobertura de Grab, consulte [Países, regiones y áreas cubiertas](#) en esta guía.

Puede hacerlo mediante la consola de Amazon Location Service AWS CLI, la o las API de ubicación de Amazon.

Console

Para crear un recurso de índice de ubicación mediante la consola de Amazon Location Service

1. Abra la consola de Amazon Location Service en <https://console.aws.amazon.com/location/>.
2. En el panel de navegación izquierdo, elija Índices de ubicación.
3. Elija Crear índice de ubicación.
4. Complete las casillas siguientes:
 - Nombre: introduzca un nombre para el recurso de índice de ubicación. Por ejemplo, *ExamplePlaceIndex*. 100 caracteres como máximo. Entre las entradas válidas se incluyen caracteres alfanuméricos, guiones, puntos y guiones bajos.
 - Descripción : escriba una descripción opcional.
5. En Proveedores de datos, elija un [proveedor de datos](#) disponible para usarlo con su recurso de índice de ubicación.

Note

Si su aplicación rastrea o enruta los activos que utiliza en su empresa, como vehículos de reparto o empleados, no debe utilizar Esri como proveedor de geolocalización. Consulte la sección 82 de los [Términos de servicio deAWS](#) para obtener más detalles.

6. En Opciones de almacenamiento de datos, especifique si desea almacenar los resultados de búsqueda de su recurso de índice de ubicación.
7. (Opcional) En Etiquetas, escriba una Clave y un Valor de etiqueta. Esto añade una etiqueta a su nuevo recurso de índice de ubicación. Para obtener más información, consulte [Etiquetado de los recursos](#).
8. Elija Crear índice de ubicación.

API

Para crear un recurso de índice de ubicación mediante las API de Amazon Location

Use la operación de [CreatePlaceIndex](#) de las API de Amazon Location Places.

El siguiente ejemplo es una solicitud de API para crear un recurso de índice de sitios llamado *ExamplePlaceIndex* mediante el proveedor de datos *Esri*.

```
POST /places/v0/indexes
Content-type: application/json

{
  "DataSource": "Esri",
  "DataSourceConfiguration": {
    "IntendedUse": "SingleUse"
  },
  "Description": "string",
  "IndexName": "ExamplePlaceIndex",
  "Tags": {
    "Tag1" : "Value1"
  }
}
```

AWS CLI

Para crear un recurso de índice de ubicación mediante comandos de AWS CLI

Utilice el comando [create-place-index](#).

En el siguiente ejemplo, se crea un recurso de índice de sitios denominado *ExamplePlaceIndex* con *Esri* como proveedor de datos.

```
aws location \
```

```
create-place-index \  
--data-source "Esri" \  
--description "Example place index" \  
--index-name "ExamplePlaceIndex" \  
--tags Tag1=Value1
```

Note

La facturación depende del uso. Puede incurrir en cargos por el uso de otros servicios de AWS . Para obtener más información, consulte los [precios de Amazon Location Service](#).

Autenticación de sus solicitudes

Una vez que haya creado un recurso de índice de ubicación y esté listo para empezar a incorporar funciones de ubicación a su aplicación, elija cómo autenticaría sus solicitudes:

- Para explorar las formas en las que puede acceder a los servicios, consulte [Cómo acceder a Amazon Location Service](#).
- Si tiene un sitio web con usuarios anónimos, puede utilizar claves de API o Amazon Cognito.

Ejemplo

En el siguiente ejemplo, se muestra el uso de una clave de API para la autorización, el [JavaScriptSDK v3 de AWS](#) y la ubicación [JavaScript Ayudante de autenticación](#) de Amazon.

```
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);
```

Geocodificación

La geocodificación es un proceso que convierte texto como una dirección, una región, el nombre de una empresa o un punto de interés, en un conjunto de coordenadas geográficas. Puede usar los recursos del índice de ubicación para enviar solicitudes de geocodificación e incorporar los datos recuperados de la geocodificación para mostrarlos en un mapa para su aplicación web o móvil.

En esta sección, se explica cómo enviar una solicitud de geocodificación sencilla y cómo enviar solicitudes de geocodificación con especificaciones opcionales.

Geocodificación

Puede enviar una simple solicitud de geocodificación mediante la operación [SearchPlaceIndexForText](#) para convertir una dirección en un conjunto de coordenadas. Una solicitud simple contiene el siguiente parámetro obligatorio:

- **Text**: una dirección, un nombre, una ciudad o una región para convertirlos en un conjunto de coordenadas. Por ejemplo, la cadena Any Town.

Para especificar un número máximo de resultados por página, utilice el siguiente parámetro opcional:

- **MaxResults**: limita el número máximo de resultados devueltos en la respuesta a la consulta.

Puedes usar las API de ubicación de Amazon AWS CLI o las API.

API

El siguiente ejemplo es una [SearchPlaceIndexForText](#) solicitud para buscar en el recurso de índice de *ExamplePlaceIndex* sitios una dirección, nombre, ciudad o región denominada *Any Town*.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json

{
  "Text": "Any Town",
  "MaxResults": 10
}
```

AWS CLI

El siguiente ejemplo es un [search-place-index-for-text](#) comando para buscar en el recurso de índice de lugares una dirección, nombre, ciudad o región denominada *Any Town*. *ExamplePlaceIndex*

```
aws location \
  search-place-index-for-text \
    --index-name ExamplePlaceIndex \
    --text "Any Town" \
    --max-results 10
```

Geocodificar cerca de una posición

Al geocodificar, puede geocodificar cerca de una posición determinada con el siguiente parámetro opcional:

- **BiasPosition**: la posición cercana a la que desea buscar. Esto reduce la búsqueda al buscar los resultados más cercanos a la posición dada. Definido como [longitude, latitude]

El siguiente ejemplo es una solicitud de [SearchPlaceIndexForText](#) para buscar en el recurso de índice de ubicación una dirección, nombre, ciudad o región denominada Cualquier ciudad cercana a la posición [-123.4567, 45.6789].

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json
```

```
{
  "Text": "Any Town",
  "BiasPosition": [-123.4567, 45.6789]
}
```

Geocodificar dentro de un cuadro delimitador

Puede geocodificar dentro de un cuadro delimitador para limitar los resultados a las coordenadas dentro de un límite determinado mediante el siguiente parámetro opcional:

- **FilterBoundingBox**: un cuadro delimitador que se especifica para filtrar los resultados a las coordenadas que se encuentran dentro de los límites del cuadro. Definido como [LongitudeSW, LatitudeSW, LongitudeNE, LatitudeNE]

Note

Una solicitud no puede contener los parámetros de **FilterBoundingBox** y **BiasPosition** a la vez. Si se especifican ambos parámetros en la solicitud, se produce un error de **ValidationException**.

El siguiente ejemplo es una solicitud de [SearchPlaceIndexForText](#) para buscar dentro de un cuadro delimitador una dirección, nombre, ciudad o región denominada *Cualquier ciudad*. El cuadro delimitador es el siguiente:

- La longitud de la esquina suroeste es *-124.1450*.
- La latitud de la esquina suroeste es *41.7045*.
- La longitud de la esquina noreste es *124.1387*.
- La latitud de la esquina noreste es *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
]
```

Geocodificar dentro de un país

Puede geocodificar en uno o más países que especifique mediante el siguiente parámetro opcional:

- **FilterCountries:** el país o la región en los que desea geocodificar. Puede definir hasta 100 países en una solicitud utilizando un código de país de tres letras según el estándar [ISO 3166](#). Por ejemplo, utilice AUS para Australia.

El siguiente ejemplo es una solicitud de [SearchPlaceIndexForText](#) para la búsqueda de una dirección, nombre, ciudad o región denominada *Cualquier ciudad* en Alemania o Francia.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json

{
  "Text": "Any Town",
  "FilterCountries": ["DEU","FRA"]
}
```

Filtrar por categoría

Puede filtrar las categorías que se muestran en su solicitud de geocodificación mediante el siguiente parámetro opcional:

- **FilterCategories:** las categorías de resultados que desea que se muestren en la consulta. Puede especificar hasta 5 categorías en una única solicitud. Puede encontrar la lista de categorías de Amazon Location Service en la sección [Categorías](#). Por ejemplo, puede utilizar Hotel en la consulta para que se especifiquen solo los hoteles que regresan.

El siguiente ejemplo es una solicitud de [SearchPlaceIndexForText](#) para buscar una cafetería llamada *Hometown Coffee* en los Estados Unidos.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
```

```
Content-type: application/json

{
  "Text": "Hometown Coffee",
  "FilterCategories": ["Coffee Shop"],
  "FilterCountries": ["USA"]
}
```

Para obtener más información sobre el filtrado de categorías, consulte [Coloca categorías y filtra los resultados](#)

Geocodifique en un idioma preferido

Puede establecer una preferencia de idioma para los resultados de la búsqueda mediante el parámetro opcional Language. Por ejemplo, es **100 Main St, Anytown, USA** posible que se devuelva una búsqueda de **100 Main St, Any Town, USA** por defecto. Sin embargo, si selecciona **fr** como Language, es posible que los resultados aparezcan como **100 Rue Principale, Any Town, États-Unis** en su lugar.

- Language: un código de idioma que se utilizará para representar los resultados de la consulta. El valor debe ser un código de idioma [BCP 47](#) válido. Por ejemplo, en para inglés.

Note

Si no se especifica Language o el idioma especificado no es compatible con un resultado, se utilizará el idioma predeterminado del socio para ese resultado.

El siguiente ejemplo es una solicitud de SearchPlaceIndexforText para buscar un lugar llamado **Any Town** con el idioma preferido especificado como el de.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json
{
  "Text": "Any Town",
  "Language": "de"
}
```

Ejemplo de respuesta

Example

El siguiente es un ejemplo de respuesta cuando llamas a la operación de [SearchPlaceIndexForText](#) desde las API de Amazon Location Places. Los resultados incluyen las [ubicaciones](#) relevantes y el [resumen](#) de la solicitud. Se muestran dos respuestas basadas en la selección de Esri o HERE como socio.

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": "AQAAAIADsn2T3KdrRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-
o3nqdlJZAdgcT2oWi1w9pS4wXX0k301vsK1GsPyHjV4EJxsu289i3hVO_BUPgP7SFoWAI8BW2v7LvAjQ5NfUPy7a1v9a
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ",
    "Distance":
    523.4619749905755
  }
]
}

```

Geocodificación inversa

La geocodificación inversa es un proceso que convierte un conjunto de coordenadas en texto significativo, como una dirección, una región, el nombre de una empresa o un punto de interés. Puede utilizar los recursos de índice de ubicación para enviar solicitudes de geocodificación inversa e incorporar los datos recuperados de la geocodificación inversa para mostrarlos en un mapa para su aplicación web o móvil.

En esta sección, se explica cómo enviar una solicitud sencilla de geocodificación inversa.

Geocodificación inversa

Puede enviar una solicitud sencilla para geocodificar de forma inversa un conjunto de coordenadas y convertirlas en una dirección significativa, un punto de interés o una ubicación general sin una dirección mediante la operación de [SearchPlaceIndexForPosition](#). Una solicitud simple contiene el siguiente parámetro obligatorio:

- **Position**: un conjunto de coordenadas que desea convertir en una dirección, un punto de interés o una ubicación general. Definido mediante el formato de `[longitude, latitude]`.

Para especificar un número máximo de resultados por página, añada el siguiente parámetro opcional:

- **MaxResults**: limita el número máximo de resultados devueltos en la respuesta a la consulta.

Si desea especificar un idioma preferido para los resultados de la consulta, utilice el siguiente parámetro opcional:

- **Language**: un código de idioma que se utilizará para representar los resultados. El valor debe ser un código de idioma [BCP 47](#) válido. Por ejemplo, en para inglés.

Note

Si no se especifica **Language** o el idioma especificado no es compatible con un resultado, se utilizará el idioma predeterminado del socio para ese resultado.

Puedes usar las API de ubicación de Amazon AWS CLI o las API.

API

El siguiente ejemplo es una [SearchPlaceIndexForPosition](#) solicitud para buscar en el recurso del índice de sitios una dirección significativa *ExamplePlaceIndex*, un punto de interés o una ubicación general cercana a la posición `[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

El siguiente ejemplo es un [search-place-index-for-position](#) comando para buscar en el recurso de indexación de lugares una dirección significativa `ExamplePlaceIndex`, un punto de interés o una ubicación general cercana a la posición [122.3394, 47.6159].

```
aws location \
  search-place-index-for-position \
    --index-name ExamplePlaceIndex \
    --position -122.3394 47.6159 \
    --max-results 5 \
    --language de
```

Ejemplo de respuesta

Example

El siguiente es un ejemplo de respuesta al llamar a la operación [SearchPlaceIndexForPosition](#) desde las API de Amazon Location Places. Los resultados muestran las [ubicaciones](#) relevantes y el [resumen](#) de la solicitud. Se muestran dos respuestas basadas en la selección de Esri o Here como socio.

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": "AQAAAIAADsn2T3KdrRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-
o3nqd1JZAdgcT2oWi1w9pS4wXX0k301vsK1GsPyHjV4EJxsu289i3hVO_BUPgP7SFoWai8BW2v7LvAjQ5NfUPy7a1v9a
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ",
      "Distance": 1.868586125090601
    }
  ]
}

```

Autocompletar

Autocomplete proporciona comentarios reactivos a los usuarios finales a medida que escriben su consulta de búsqueda. Proporciona sugerencias para direcciones y puntos de interés basados en texto de formato libre parcial o mal escrito. Puede utilizar los recursos de índice de ubicación para solicitar sugerencias de autocompletar y mostrar las sugerencias resultantes en su aplicación.

Amazon Location no admite el almacenamiento de sugerencias que se completan automáticamente. Se devuelve un error si el índice de ubicación utilizado para una llamada de autocompletar está configurada para su uso con los códigos geográficos almacenados. Cree y configure varios índices de ubicación para utilizar los códigos geográficos almacenados y solicitar sugerencias.

En esta sección, se describe cómo enviar una solicitud de autocompletar. Comienza con la solicitud más básica y, a continuación, muestra los parámetros opcionales que puede usar para aumentar la relevancia de los resultados de búsqueda que se completan automáticamente.

Uso de la función de autocompletar

Puede enviar una simple solicitud de sugerencias de autocompletar mediante la operación de [SearchPlaceIndexForSuggestions](#). La forma más simple de la solicitud tiene un único parámetro obligatorio, la consulta Text:

- Text: el texto parcial de formato libre que se utilizará para generar sugerencias de ubicaciones. Por ejemplo, la cadena `eiffel tow`.

Para limitar el número de resultados regresados, añada el parámetro opcional `MaxResults`:

- `MaxResults`: limita el número de resultados regresados en la respuesta de la consulta.

Puede utilizar las API de Amazon Location o las de AWS CLI.

API

El siguiente ejemplo es una [SearchPlaceIndexForSuggestions](#) solicitud para buscar en el recurso de indexación de lugares hasta 5 sugerencias basadas en el topónimo parcial `kamp`. `ExamplePlaceIndex`

```
POST /places/v0/indexes/ExamplePlaceIndex/search/suggestions
Content-type: application/json

{
  "Text": "kamp",
  "MaxResults": 5
}
```

AWS CLI

El siguiente ejemplo es un [search-place-index-for-suggestions](#) comando para buscar en el recurso de índice de lugares hasta **5** sugerencias basadas en el nombre parcial del lugar *kamp*. *ExamplePlaceIndex*

```
aws location \
    search-place-index-for-suggestions \
    --index-name ExamplePlaceIndex \
    --text kamp \
    --max-results 5
```

La llamada a SearchPlaceIndexForSuggestions da como resultado una lista de ubicaciones con un nombre y un identificador para cada uno. Puede usar esos resultados para presentar sugerencias sobre lo que el usuario podría estar buscando mientras escribe, por ejemplo, proporcionando una lista desplegable de opciones debajo de un cuadro de texto. Como ejemplo, estos son los resultados de las sugerencias basadas en un usuario que ha escrito *kamp*.

```
{
  "Summary": {
    "Text": "kamp",
    "MaxResults": 5,
    "DataSource": "Esri"
  },
  "Results": [
    {
      "Text": "Kampuchea",
      "PlaceId": "AQAAAIAADsn2T3KdrRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-
o3nqd1JZAdgcT2oWi1w9pS4wXX0k301vsK1GsPyHjV4EJxsu289i3hVO_BUPgP7SFoWAI8BW2v7LvAjQ5NfUPy7a1v9ajT3
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ"
    },
    {
      "Text": "Kampoul, Kabul, AFG",
      "PlaceId":
"AQAAAIAAAA1mx1_-9ffzXD07rBgo9fh6E01Pd1YKvuT5rz2qBDxqBkhTlgkei0PR2s5sa3YBLxUqQI8bhYmsYcu9R-
DkX3L9QSi3CB5LhNPu160iSFJo6H8S1CrX03QsJALhrr9mdbg0R4R4YDywkHkeBlbnbn7g5C5LI_wYx873WeQZuilwtsGm8j
UeXcb_bg"
    },
    {
      "Text": "Kampala, UGA",
```

```

    "PlaceId":
      "AQAAAIAAzZfZt3qMrUKG0byhP6MM0pqy2L8SULLVWT7a3ertLBRS6Q5n7I4s9D7E0nRHADAJ7mL7kvX1Q8HD-
      mpuiATXNJ1Ix4_V_1B15zHe8j1YKMWvXbgb08cMpgR2fqYqZMR1x-
      dfB0080oqujKZldvPIDK1kNe3GwcaqvMWWPMeaGd203brFynubAe-MmFF-Gjz-WBMfUy9og6MV7bkk6NGCA"
    },
    {
      "Text": "Kampar, Riau, IDN",
      "PlaceId": "AQAAAIAAvbXXx-
      sr0i111tH0kPdao0GF7WQ_KaZ444SEnevycp6Gtf_2JWgPfCE5bIQCYwya1uZQpX2a8YJoFm2K7Co14fLu7IK0yYOLhZx4k
    },
    {
      "Text": "Kampung Pasir Gudang Baru, Johor, MYS",
      "PlaceId":
        "AQAAAIAA4HLQHdjUDcaaXLE9wtNIT1cjQYLgkBNMoG2eNN0AaQ8PJoWabLRXmmPUaAj8MAD6vT0i6zqaun5Mixyj7vnYX
    }
  ]
}

```

En la siguiente sección, se explica cómo usar el PlaceID de estos resultados.

Uso de los resultados con autocompletar

La llamada a `SearchPlaceIndexForSuggestions` da como resultado una lista de ubicaciones con un nombre y un identificador para cada uno. Puede usar esos resultados para presentar sugerencias sobre lo que el usuario podría estar buscando mientras escribe, por ejemplo, proporcionando una lista desplegable de opciones debajo de un cuadro de texto. Cuando el usuario elige uno de los resultados, puede llamar a la [GetPlace](#) operación con el identificador que haya seleccionado para obtener los detalles de ese lugar, como la ubicación, la dirección u otros detalles.

Note

PlaceId es válido solo si lo siguiente coincide en la solicitud de búsqueda original y en la llamada a `GetPlace`.

- ¿Cliente Cuenta de AWS
- Región de AWS
- Proveedor de datos especificado en el recurso de índice de ubicación

Normalmente, se utiliza `GetPlace` con las API de Amazon Location. El siguiente ejemplo es una solicitud de [GetPlace](#) para buscar una de las sugerencias de la sección anterior. Este ejemplo se basa en el nombre parcial de lugar *kamp*.

```
POST /places/v0/indexes/ExamplePlaceIndex/
places/AQAAAIAADsn2T3KdrRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-
o3nqdlJZAdgcT2oWi1w9pS4wXX0k301vsKLGsPyHjV4EJxsu289i3hV0_BUPgP7SFoWai8BW2v7LvAjQ5NfUPy7a1v9ajT3
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ
```

Completar automáticamente cerca de una posición

Cuando busque sugerencias de lugares que se completen automáticamente mediante el uso de [SearchPlaceIndexForSuggestions](#), puede obtener sugerencias más relevantes a nivel local añadiendo el siguiente parámetro opcional:

- `BiasPosition`: la posición cercana a la que desea buscar. Definido como `[longitude, latitude]`.

En el siguiente ejemplo, se utiliza una [SearchPlaceIndexForSuggestions](#) solicitud para buscar sugerencias de sitios que `ExamplePlaceIndex` coincidan con el campo de consulta parcial cercano a la posición `[32.5827, 0.3169]`.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/suggestions
Content-type: application/json


{
  "Text": "kamp",
  "BiasPosition": [32.5827,0.3169]
}
```

Las sugerencias devueltas para el mismo `Text` pueden ser diferentes si se elige una `BiasPosition` diferente, como `[-96.7977, 32.7776]`.

Completar automáticamente dentro de un recuadro delimitador

Puede limitar la búsqueda de autocompletar para recibir solo sugerencias de lugares que se encuentren dentro de un límite determinado añadiendo el siguiente parámetro opcional:

- **FilterBBox**: un cuadro delimitador que se especifica para filtrar los resultados a las coordenadas que se encuentran dentro de los límites del cuadro. Definido como [LongitudeSW, LatitudeSW, LongitudeNE, LatitudeNE]

 Note

Una solicitud no puede contener los parámetros de **FilterBBox** y **BiasPosition** a la vez. Si se especifican ambos parámetros en la solicitud, se produce un error de **ValidationException**.

En el siguiente ejemplo, se utiliza una [SearchPlaceIndexForSuggestions](#) solicitud para buscar en el recurso `ExamplePlaceIndex` de indexación de lugares sugerencias de lugares que coincidan con el intervalo de consulta parcial y que estén contenidas en el cuadro delimitador, donde:

- La longitud de la esquina suroeste del cuadro delimitador es *32.5020*.
- La latitud de la esquina suroeste del cuadro delimitador es *0.2678*.
- La longitud de la esquina noreste del recuadro delimitador es *32.6129*.
- La latitud de la esquina noreste del cuadro delimitador es *0.3502*.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/suggestions
```

```
Content-type: application/json
```

```
{
  "Text": "kamp",
  "FilterBBox": [
    32.5020, 0.2678,
    32.6129, 0.3502
  ]
}
```

Las sugerencias devueltas para el mismo `Text` serán diferentes si se elige una `FilterBBox` diferente, como [*-97.9651, 32.0640, -95.1196, 34.0436*].

Completar automáticamente dentro de un país

Puede limitar la búsqueda de autocompletar para recibir solo sugerencias de lugares ubicados dentro de un país o un conjunto de países determinados añadiendo el siguiente parámetro opcional:

- `FilterCountries`: los países en los que quieres buscar sugerencias de lugares. Puede especificar hasta 100 países en una solicitud utilizando un código de país de tres letras [ISO 3166](#). Por ejemplo, utilice AUS para Australia.

En el siguiente ejemplo, se utiliza una [SearchPlaceIndexForSuggestions](#) solicitud para buscar en el recurso `ExamplePlaceIndex` de índice de lugares sugerencias de lugares que coincidan con la consulta parcial `kamp` y que estén incluidas en Uganda, Kenia o Tanzania:

```
POST /places/v0/indexes/ExamplePlaceIndex/search/suggestions
Content-type: application/json

{
  "Text": "kamp",
  "FilterCountries": ["UGA", "KEN", "TZA"]
}
```

Las sugerencias devueltas para el mismo Text son diferentes si se elige una lista de `FilterCountries`, como [«USA»].

Ejemplo de respuesta

A continuación se muestra un ejemplo de respuesta de las opciones de autocompletar sugeridas para la operación de [SearchPlaceIndexForSuggestions](#), utilizando el texto de kamp.

```
{
  "Summary": {
    "Text": "kamp",
    "MaxResults": 5,
    "DataSource": "Esri"
  },
  "Results": [
    {
      "Text": "Kampuchea",
```

```

    "PlaceId": "AQAAAIAADsn2T3KdRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-
o3nqdlJZAdgcT2oWi1w9pS4wXX0k301vsK1GsPyHjV4EJxsu289i3hV0_BUPgP7SFoWAI8BW2v7LvAjQ5NfUPy7a1v9ajT3
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ"
  },
  {
    "Text": "Kampoul, Kabul, AFG",
    "PlaceId":
"AQAAAIAAAA1mx1_-9ffzXD07rBgo9fh6E01Pd1YKvuT5rz2qBDxqBkhTlGkei0PR2s5sa3YBLxUqQI8bhymYcu9R-
DkX3L9QSi3CB5LhNPu160iSFJo6H8S1CrX03QsJALhrr9mdbg0R4R4YDywkHkeB1nbn7g5C5LI_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
  }
]
}

```

Uso de identificadores de lugares

La búsqueda de lugares devuelve una lista de resultados. La mayoría de los resultados incluyen un `PlaceId` para ese resultado. Puede usar un `PlaceId` en una operación de [GetPlace](#) para devolver la información sobre ese lugar (incluidos el nombre, la dirección, la ubicación u otros detalles).

Note

Si [SearchPlaceIndexForSuggestions](#) se utiliza, se devolverán `PlaceId` los resultados de cualquier índice de sitios creado con cualquier fuente de datos. Utilice

[SearchPlaceIndexForTexto](#) [SearchPlaceIndexForPosition](#) devolverá un valor `PlaceId` únicamente si la fuente de datos utilizada está AQUÍ.

Cada `PlaceId` define de forma única el lugar al que hace referencia, pero un solo lugar puede tener más de un `PlaceId` a lo largo del tiempo y en función del contexto. Las siguientes reglas describen la singularidad y la longevidad de un `PlaceId`.

- Los datos `PlaceId` devueltos en las llamadas que realices son específicos para tu Cuenta de AWS, para la AWS región y para el proveedor de datos de tu `PlaceIndex` recurso. `GetPlace` solo encontrará resultados cuando estos tres atributos coincidan con la llamada original que creó la `PlaceId`.
- El `PlaceId` de un lugar cambiará cuando cambien los datos sobre ese lugar. Por ejemplo, cuando la empresa a la que hace referencia cambia de ubicación o cambia de nombre.
- Los datos `PlaceId` devueltos por una llamada de búsqueda repetida pueden cambiar cuando el servicio de backend realiza una actualización. Se seguirá encontrando el anterior `PlaceId`, pero es posible que las nuevas llamadas a la búsqueda devuelvan un identificador diferente.

El `PlaceId` es una cadena. No hay un límite específico para la longitud de un `PlaceId`. A continuación, se muestra un ejemplo de un `PlaceId` válido.

```
AQAAAAIAADsn2T3KdrRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-  
o3nqd1JZA dgcT2oWi1w9pS4wXX0k301vsK1GsPyHjV4EJxsu289i3hV0_BUPgP7SFoWAi8BW2v7LvAjQ5NfUPy7a1v9ajT3  
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ
```

Si llamas `GetPlace` con un `PlaceId` por un lugar cuyos datos han cambiado (por ejemplo, un local empresarial que ha dejado de funcionar), se generará un error `404 ResourceNotFound`. Si se llama `GetPlace` con un valor `PlaceId` que no es válido o con uno fuera de contexto (por ejemplo, desde otro) Cuenta de AWS, se devolverá un `400 ValidationException` error.

Si bien puedes usar `PlaceID` en solicitudes posteriores, `PlaceID` no pretende ser un identificador permanente y el ID puede cambiar entre llamadas consecutivas a la API. Consulta el siguiente comportamiento de `PlaceID` para cada proveedor de datos:

- Esri: Los ID de lugar cambiarán cada trimestre como mínimo. El período de tiempo habitual para estos cambios sería marzo, junio, septiembre y diciembre. Los identificadores de lugares también pueden cambiar entre el cambio trimestral típico, pero eso será mucho menos frecuente.

- **AQUÍ:** Le recomendamos que almacene los datos en caché durante no más de una semana para mantenerlos actualizados. Puede suponer que se publicarán menos del 1% de los turnos de identificación en comparación con el lanzamiento, es decir, aproximadamente 1 o 2 veces por semana.
- **Grab:** Los ID de lugar pueden caducar o dejar de ser válidos en las siguientes situaciones.
 - **Operaciones de datos:** Grab Map Ops puede eliminar el POI de la base de datos de puntos de interés de Grab basándose en datos reales, como estar cerrado en el mundo real, ser detectado como un POI duplicado o tener información incorrecta. Grab sincronizará los datos con el entorno de Waypoint semanalmente.
 - **POI interpolado:** el POI interpolado es un POI temporal que se genera en tiempo real al atender una solicitud y se marcará como derivado en el campo de la respuesta. `place.result_type` La información de los POI interpolados se conservará durante al menos 30 días, lo que significa que, en un plazo de 30 días, podrá obtener los detalles del POI mediante el ID de lugar de la API Place Details. Transcurridos 30 días, es posible que los POI interpolados (tanto el ID como los detalles) caduquen y no se pueda acceder a ellos desde la API de Places Details.

Coloca categorías y filtra los resultados

Los lugares están categorizados. Si busca una empresa, la empresa podría ser un `Restaurant`, por ejemplo, una. Incluso los resultados de una búsqueda de una dirección se pueden clasificar en función de si coincide con una dirección, una calle o una intersección.

En términos generales, Amazon Location Service clasifica los lugares en Tipos de lugares. Los puntos de interés se clasifican además en Tipos de puntos de interés.

Note

No todos los resultados tendrán categorías.

Puede usar las categorías para filtrar las búsquedas de geocodificación.

Filtrado de resultados

Cuando utilice las `SearchPlaceIndexForText`, podrá filtrar los resultados devueltos por las categorías que desee utilizar. Por ejemplo:

- Si quiere buscar un lugar llamado «café local» y recibir solo resultados clasificados como cafeterías, puede hacerlo llamando a las `SearchPlaceIndexForText` e incluyendo la categoría punto de interés, `Coffee Shop` en el parámetro de `FilterCategories`.
- Si busca «123 Main St, Anytown, WA, 98123, EE. UU.», puede filtrar los resultados solo por direcciones para que no coincidan, por ejemplo, con el código postal. Filtre solo las direcciones incluyendo el Tipo de lugar y el `AddressType` en parámetro de `FilterCategories`.

Note

No todos los proveedores de datos admiten el filtrado o lo admiten de la misma manera. Para obtener más información, consulte [Limitaciones de filtrado por proveedor de datos](#).

En la siguiente sección, se enumeran las categorías por las que puede filtrar.

Categorías

En las siguientes listas se muestran las categorías que Amazon Location Service utiliza para clasificar y filtrar. Estas categorías se utilizan en todos los idiomas, independientemente de si el parámetro de idioma está configurado en un idioma diferente.

Note

Amazon Location Service asigna categorías de proveedores de datos a este conjunto de categorías. Si un proveedor de datos incluye un lugar en una categoría que no forma parte de la lista de categorías de Amazon Location Service, la categoría de proveedor se incluirá en los resultados como categoría complementaria.

Tipos de lugares: estos tipos se utilizan para indicar el tipo de coincidencia que se utilizó para encontrar el resultado.

- `AddressType`: aparece cuando el resultado coincide con una dirección.
- `StreetType`: aparece cuando el resultado coincide con una calle.
- `IntersectionType`: aparece cuando el resultado coincide con la intersección de dos calles.
- `PointOfInterestType`: aparece cuando el resultado coincide con un punto de interés, como un negocio o un lugar cívico.

- **CountryType**: aparece cuando el resultado coincide con un país o una región importante.
- **RegionType**: aparece cuando el resultado coincide con una región de un país, como un estado o una provincia.
- **SubRegionType**: aparece cuando el resultado coincide con una subregión de un país, como un condado o un área metropolitana.
- **MunicipalityType**: aparece cuando el resultado coincide con una ciudad o un pueblo.
- **NeighborhoodType**: aparece cuando el resultado coincide con un barrio o área dentro de una ciudad.
- **PostalCodeType**: aparece cuando el resultado coincide con un código postal.

Categorías de puntos de interés: estas categorías se utilizan para indicar el tipo de empresa o ubicación para obtener resultados sobre puntos de interés.

- 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

Limitaciones de filtrado por proveedor de datos

No todos los proveedores tienen la misma funcionalidad de filtrado. En la siguiente tabla, se describen las diferencias.

Proveedor	API con soporte para filtros	Categorías compatibles con el filtrado	Valores de retorno:
Esri	<code>SearchPlaceIndexForText</code> , <code>SearchPlaceIndexForSuggestions</code>	Filtrar por Tipos de lugares y Categorías de puntos de interés.	Las categorías se devuelven por <code>SearchPlaceIndexForText</code> , <code>SearchPlaceIndexForPosition</code> y <code>GetPlace</code>
Aquí	<code>SearchPlaceIndexForText</code> , <code>SearchPlaceIndexForSuggestions</code>	Filtrar solo por Tipos de lugar.	Las categorías se devuelven mediante <code>SearchPlaceIndexForText</code> y <code>SearchPlaceIndexFor</code>

Proveedor	API con soporte para filtros	Categorías compatibles con el filtrado	Valores de retorno:
			rSuggestions, SearchPlaceIndexFo rPosition y GetPlace
Grab	no admitido	no admitido	no admitido
Abrir datos	n/d (no se admite la búsqueda de lugares)	n/a	n/a

Funciones definidas por el usuario de Amazon Aurora PostgreSQL para Amazon Location Service

Puede utilizar Amazon Location Service para trabajar con las coordenadas y direcciones almacenadas en tablas de bases de datos para limpiar y enriquecer sus datos geospaciales.


Por ejemplo:

- Puede utilizar la geocodificación para convertir las direcciones en coordenadas a fin de normalizar y rellenar los huecos en los datos de las direcciones almacenadas en una tabla de base de datos.
- Puede geocodificar las direcciones para obtener sus posiciones y utilizar las coordenadas con funciones espaciales de la base de datos, como una función que muestre las filas de un área específica.
- Puede utilizar datos enriquecidos para generar informes automatizados, como generar un informe automatizado que ilustre todos los dispositivos de un área determinada, o un informe automatizado para machine learning que ilustre las áreas con tasas de error más altas al enviar actualizaciones de ubicación.

En este tutorial se muestra cómo formatear y enriquecer las direcciones almacenadas en una tabla de base de datos de Amazon Aurora PostgreSQL mediante Amazon Location Service.

- Amazon Aurora PostgreSQL: un motor de base de datos relacional completamente administrado, compatible con MySQL y PostgreSQL, que genera hasta cinco veces el rendimiento de MySQL

y hasta tres veces el rendimiento de PostgreSQL sin cambiar la mayoría de sus aplicaciones existentes. Para obtener más información, consulte la sección [¿Qué es Amazon Aurora?](#) en la Guía del usuario de Amazon Aurora.

 Important

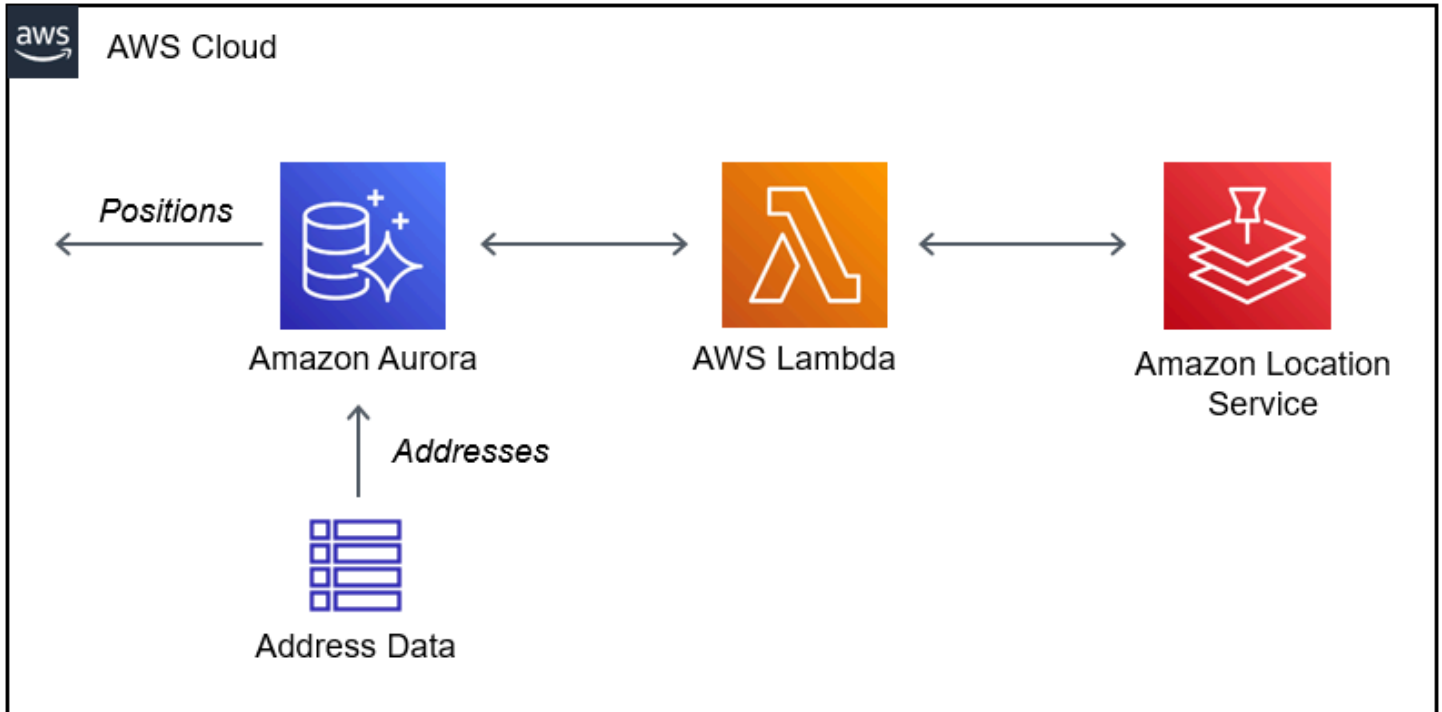
La aplicación resultante de este tutorial utiliza un índice de ubicación que almacena resultados de geocodificación. Para obtener información sobre los cargos aplicables por almacenar resultados de geocodificación, consulte los [precios de Amazon Location Service](#).

El código de muestra está disponible en el repositorio de muestras de Amazon Location Service [GitHub](#), que incluye [una AWS CloudFormation plantilla](#).

Temas

- [Información general](#)
- [Requisitos previos](#)
- [Inicio rápido](#)
- [Crear un recurso de índice de ubicación](#)
- [Crear una función de AWS Lambda para geocodificación](#)
- [Conceder a Amazon Aurora PostgreSQL acceso a AWS Lambda](#)
- [Invocar la función de AWS Lambda](#)
- [Enriquecer una base de datos que contiene datos de direcciones](#)
- [Siguiendo pasos](#)

Información general



La arquitectura incluye las siguientes integraciones:

- Esta solución utiliza un recurso de índice de ubicación de Amazon Location para admitir las consultas de geocodificación que utilizan la operación `SearchPlaceIndexForText`.
- AWS Lambda utiliza un Lambda de Python que geocodifica las direcciones cuando una política de IAM da permiso para permitir que AWS Lambda llame a la operación de geocodificación de Amazon Location, `SearchPlaceIndexForText`.
- Conceda permiso a Amazon Aurora PostgreSQL para invocar la función de Lambda de geocodificación mediante una función SQL definida por el usuario.

Requisitos previos

Antes de empezar, necesita los siguientes requisitos previos:

- Un clúster de Amazon Aurora PostgreSQL. Para obtener más información sobre [Crear un clúster de base de datos de Amazon Aurora](#) en la Guía del usuario de Amazon Aurora.

Note

Si su clúster de Amazon Aurora no está disponible de forma pública, también debe configurar Amazon Aurora para conectarse a AWS Lambda en una nube privada virtual (VPC) de su cuenta de AWS. Para obtener más información, consulte [Conceder a Amazon Aurora PostgreSQL acceso a AWS Lambda](#).

- Una herramienta de desarrollo de SQL para conectarse al clúster de Amazon Aurora PostgreSQL.

Inicio rápido

Como alternativa a seguir los pasos de este tutorial, puede lanzar una pila rápida para implementar una función de AWS Lambda que admita la operación [SearchPlaceIndexForText](#) de Amazon Location. Esto configura automáticamente su cuenta de AWS para permitir que Amazon Aurora llame a AWS Lambda.

Una vez que haya configurado su cuenta de AWS, tendrá que:

- agregar la función de Lambda a Amazon Aurora. Consulte agregar el rol de IAM a un clúster de base de datos de Amazon Aurora en [Conceder a Amazon Aurora PostgreSQL acceso a AWS Lambda](#).
- Cargar la función definida por el usuario en su base de datos. Consulte [Invocar la función de AWS Lambda](#).

A yellow button with a blue play icon and the text "Launch Stack".

Crear un recurso de índice de ubicación

Comience creando un recurso de índice de ubicación para admitir las consultas de geocodificación.

1. Abra la consola de Amazon Location Service en <https://console.aws.amazon.com/location/>.
2. En el panel de navegación izquierdo, elija Índices de lugares.
3. Complete las casillas siguientes:

- Nombre: introduzca un nombre para el recurso de índice de ubicación. Por ejemplo, *AuroraPlaceIndex*. 100 caracteres como máximo. Entre las entradas válidas se incluyen caracteres alfanuméricos, guiones, puntos y guiones bajos.
 - Descripción: escriba una descripción opcional. Por ejemplo, *Índice de lugares para Amazon Aurora*.
4. En Proveedores de datos, elija un [proveedor de datos](#) disponible para usarlo con su recurso de índice de ubicación. Si no tiene ninguna preferencia, le recomendamos que comience con *Esri*.
 5. En Opciones de almacenamiento de datos, especifique Sí, se almacenarán los resultados. Esto indica que tiene intención de guardar los resultados de geocodificación en una base de datos.
 6. (Opcional) En Etiquetas, escriba una Clave y un Valor de etiqueta. Esto añade una etiqueta a su nuevo recurso de índice de ubicación. Para obtener más información, consulte [Etiquetado de los recursos](#).
 7. Elija Crear índice de ubicación.

Crear una función de AWS Lambda para geocodificación

Para crear una conexión entre Amazon Aurora PostgreSQL y Amazon Location Service, necesita una función de AWS Lambda que gestione las solicitudes del motor de base de datos. Esta función traduce el evento de la función definida por el usuario de Lambda y llama a la operación `SearchPlaceIndexForText` de Amazon Location.

Puede crear la función mediante la consola de AWS Lambda, AWS Command Line Interface o las API de AWS Lambda.

Para crear una función definida por el usuario de Lambda con la consola

1. Abra la consola de AWS Lambda en <https://console.aws.amazon.com/lambda/>.
2. En el panel de navegación izquierdo, elija Funciones.
3. Elija Crear función y asegúrese de que esté seleccionada la opción Autor desde cero.
4. Complete las casillas siguientes:
 - Nombre de función: introduzca un nombre único para su función. Las entradas válidas incluyen caracteres alfanuméricos, guiones y guiones bajos sin espacios. Por ejemplo, *AuroraGeocoder*.
 - Tiempo de ejecución: elija *Python 3.8*.

5. Elija Crear función.
6. Elija la pestaña Código para abrir el editor.
7. Sobrescriba el código del marcador de posición en `lambda_function.py` con lo siguiente:

```
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. Si has asignado a tu índice de sitios un nombre diferente al de *AuroraPlaceIndex*, crea una variable de entorno con un nombre `PLACE_INDEX_NAME` para asignar el nombre del recurso a:

- En la pestaña Configuración, elija Variables de entorno.
 - Seleccione Editar y, a continuación, seleccione agregar variable de entorno.
 - Para Clave: escriba PLACE_INDEX_NAME.
 - Para Valor: escriba el nombre de su recurso de índice de ubicación.
9. Seleccione Implementar para almacenar la función actualizada.
 10. En el menú desplegable Prueba, elija Configurar evento de prueba.
 11. Elija Crear nuevo evento de prueba.
 12. Introduzca el siguiente evento de prueba:

```
{
  "text": "Baker Beach",
  "biasPosition": [-122.483, 37.790],
  "filterCountries": ["USA"]
}
```

13. Elija Prueba para probar la función de Lambda.
14. Elija la pestaña Configuración.
15. En Configuración general: seleccione Permisos.
16. En Rol de ejecución: elija el Nombre del rol con el hipervínculo para conceder permisos de Amazon Location Service a su función de Lambda.
17. En la pestaña Permisos: seleccione el menú desplegable Agregar permisos y, a continuación, Crear política insertada.
18. Seleccione la pestaña JSON.
19. Agregue la política de IAM siguiente:
 - La siguiente política otorga permiso para enviar SearchPlaceIndexForText al recurso del índice de sitios *AuroraPlaceIndex*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "geo:SearchPlaceIndexForText",
      "Resource": "arn:aws:geo:<Region>:<AccountId>:place-index/AuroraPlaceIndex"
    }
  ]
}
```

```
]
}
```

20. Elija Revisar política.
21. Escriba un nombre para la política. Por ejemplo, *AuroraPlaceIndexReadOnly*.
22. Elija Crear política.

Conceder a Amazon Aurora PostgreSQL acceso a AWS Lambda

Antes de que Amazon Aurora PostgreSQL pueda invocar una función de AWS Lambda, debe conceder permiso de acceso.

Si su clúster de Amazon Aurora PostgreSQL no es de acceso público, primero tendrá que crear un punto de conexión de VPC para AWS Lambda a fin de que Amazon Aurora pueda llamar a su función de Lambda.

Crear un punto de conexión de VPC para AWS Lambda

Note

Este paso solo es obligatorio si no se puede acceder a su clúster de Amazon Aurora PostgreSQL de forma pública.

1. Abra la [Amazon Virtual Private Cloud Console](#).
2. En el panel de navegación izquierdo, elija Puntos de conexión.
3. Seleccione Crear punto de conexión.
4. En el filtro Nombre de servicio, introduzca «lambda» y, a continuación, seleccione `com.amazonaws.<region>.lambda`.
5. Elija la VPC que contiene el clúster de Aurora.
6. Seleccione una subred para cada zona de disponibilidad.
7. En el filtro Grupo de seguridad, introduzca «predeterminado» o el nombre del grupo de seguridad al que pertenece su clúster de Aurora y, a continuación, elija el grupo de seguridad.
8. Seleccione Crear punto de conexión.

Crear una política de IAM para permitir que se invoque su función de AWS Lambda

1. Abra la [consola de IAM](#).
2. En el panel de navegación izquierdo, amplíe Administración de acceso para elegir Políticas.
3. Elija Crear política.
4. En la pestaña JSON, introduzca la política siguiente:
 - Se muestra un ejemplo de una política de IAM que concede a Amazon Aurora PostgreSQL permiso para invocar la función `AuroraGeocoder` de AWS Lambda.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "lambda:InvokeFunction",
      "Resource": [
        "arn:aws:lambda:<Region>:<AccountId>:function:AuroraGeocoder"
      ]
    }
  ]
}
```

5. Elija Siguiente: Etiquetas para agregar etiquetas opcionales.
6. Elija Siguiente: Revisar.
7. Revise su política e introduzca los siguientes detalles para la política:
 - Nombre: utilice caracteres alfanuméricos y '+=, @-_' . 128 caracteres como máximo. Por ejemplo, *AuroraGeocoderInvoke*.
 - Descripción: escriba una descripción opcional. Utilice caracteres alfanuméricos y '+=, @-_' . 1000 caracteres como máximo.
8. Elija Crear política. Anote el ARN de esta política, que utiliza para asociar la política a un rol de IAM.

Crear un rol de IAM para conceder permiso a Amazon Relational Database Service (Amazon RDS)

Al crear un rol de IAM, Amazon Aurora PostgreSQL puede asumir el rol en su nombre para acceder a su función de Lambda. Para obtener más información, vea [Crear un rol para delegar permisos a un usuario de IAM](#) en Guía del usuario de IAM.

El siguiente ejemplo es un AWS CLI comando que crea un rol llamado

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"
    }
  ]
}'
```

Adjuntar su política de IAM al rol de IAM

Cuando tenga un rol de IAM, asocie la política de IAM que creó.

El siguiente ejemplo es un AWS CLI comando que asocia la política *AuroraGeocoderInvoke* al rol *AuroraGeocoderInvokeRole*.

```
aws iam attach-role-policy --policy-arn AuroraGeocoderInvoke --role-
name AuroraGeocoderInvokeRole
```

Agregar el rol de IAM a un clúster de base de datos de Amazon Aurora

El siguiente ejemplo es un AWS CLI comando para añadir una función de IAM a un Amazon Aurora PostgreSQL clúster de base de datos denominado *MyAuroraCluster*

```
aws rds add-role-to-db-cluster \
--db-cluster-identifier MyAuroraCluster \
--feature-name Lambda \
--role-arn AuroraGeocoderInvokeRole \
--region your-region
```

Invocar la función de AWS Lambda

Tras conceder permiso a Amazon Aurora PostgreSQL para invocar la función de Lambda de geocodificación, puede crear una función de Amazon Aurora PostgreSQL definida por el usuario para invocar la función de AWS Lambda de geocodificación. Para obtener más información, consulte

[Invocación de una función de AWS Lambda desde un clúster de bases de datos de Amazon Aurora PostgreSQL](#) en la Guía del usuario de Amazon Aurora.

Instalar las extensiones necesarias de PostgreSQL

Para instalar las extensiones `aws_lambda` y `aws_commons` de PostgreSQL necesarias, consulte [Información general sobre el uso de una función de Lambda](#) en la Guía del usuario de Amazon Aurora.

```
CREATE EXTENSION IF NOT EXISTS aws_lambda CASCADE;
```

Instalar las extensiones necesarias de PostGIS

PostGIS es una extensión de PostgreSQL para almacenar y administrar información espacial. Para obtener más información, consulte [Trabajo con la extensión de PostGIS](#) en la Guía del usuario de Amazon Relational Database Service.

```
CREATE EXTENSION IF NOT EXISTS postgis;
```

Crear una función de SQL definida por el usuario que invoque la función de Lambda

En un editor de SQL, cree una nueva función definida por el usuario `f_SearchPlaceIndexForText` para invocarla: *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;
$function$;
```

Llamar a la función de SQL para geocodificar desde Aurora

Al ejecutar la instrucción SQL, se invoca la *AuroraGeocoder* función Lambda, que toma los registros de direcciones de la tabla de la base de datos de la base de datos y Amazon Aurora PostgreSQL los geocodifica mediante un recurso de índice de lugares.

Note

Amazon Aurora PostgreSQL invoca la función de Lambda para cada llamada a la función de SQL definida por el usuario.

Si está geocodificando 50 filas, Amazon Aurora PostgreSQL invoca la función de Lambda 50 veces. Una invocación para cada fila.

La siguiente función `f_SearchPlaceIndexForText` SQL realiza solicitudes a la [SearchPlaceIndexForText](#) API de Amazon Location a través de la función *AuroraGeocoder* Lambda. La función devuelve una columna de geom que es una geometría de PostGIS, que `ST_AsText(geom)` convierte en texto.

```
SELECT *, ST_AsText(geom)
FROM f_SearchPlaceIndexForText('Vancouver, BC');
```

De forma predeterminada, el retorno contendrá una fila. Para solicitar filas adicionales, hasta el límite de `MaxResults`, ejecute la siguiente instrucción de SQL proporcionando un `BiasPosition` y limitándolo a los resultados en Canadá.

```
SELECT *
FROM f_SearchPlaceIndexForText('Mount Pleasant', ST_MakePoint(-123.113, 49.260), null,
'{"CAN"}', 5);
```

Para filtrar los resultados mediante un cuadro delimitador, pase un [Box2D](#) como `filter_bbox`:

- [FilterBBox](#): filtra los resultados devolviendo los lugares dentro de un cuadro delimitador. Se trata de un parámetro opcional.

```
SELECT *
FROM f_SearchPlaceIndexForText('Mount Pleasant', null, 'BOX(-139.06 48.30, -114.03
60.00)::box2d, '{"CAN"}', 5);
```

Para obtener más información sobre los tipos y funciones de PostGIS, consulte la [Referencia de PostGIS](#).

Enriquecer una base de datos que contiene datos de direcciones

Puede crear una dirección formateada y, al mismo tiempo, normalizarla y geocodificarla mediante la operación `SearchPlaceIndexForText` de Amazon Location dada una tabla de base de datos con los siguientes datos divididos en las siguientes columnas:

- `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;
```

En el siguiente ejemplo, se ilustra una fila de tabla de datos resultante:

Siguientes pasos

El código de muestra está disponible en el repositorio de muestras de Amazon Location Service [GitHub](#), que incluye [una AWS CloudFormation plantilla](#).

Administrar sus recursos de indexación de sitios

Puede administrar sus recursos de indexación de lugares mediante la consola de Amazon Location, el AWS CLI, o las API de ubicación de Amazon.

Enumere sus recursos de indexación de sitios

Puede ver su lista de recursos de índice de ubicación utilizando la consola de Amazon Location, la AWS CLI, o las API de ubicación de Amazon:

Console

Para ver una lista de los recursos de índice de ubicación mediante la consola de ubicación de Amazon

1. Abra la consola de Amazon Location en <https://console.aws.amazon.com/location/>.
2. Elija Índices de sitios en el panel de navegación izquierdo.
3. Consulta una lista de tus recursos de indexación de sitios en los índices de My place.

API

Use la operación [ListPlaceIndexes](#) de las API de Amazon Location Places.

El siguiente ejemplo es una solicitud de API para obtener una lista de los recursos de indexación de sitios de la cuenta AWS.

```
POST /places/v0/list-indexes
```

A continuación, se muestra un ejemplo de respuesta para [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

Utilice el comando [list-place-indexes](#).

El siguiente ejemplo es AWS CLI para obtener una lista de los recursos de indexación de sitios de la cuenta AWS.

```
aws location list-place-indexes
```

Obtenga los detalles de los recursos del indexación de lugares

Puede obtener detalles sobre cualquier recurso de índice de sitios de su cuenta AWS utilizando la consola de Amazon Location, la AWS CLI, o las API de ubicación de Amazon:

Console

Para ver una lista de los recursos de índice de sitios mediante la consola de Amazon Location

1. Abra la consola de Amazon Location en <https://console.aws.amazon.com/location/>.
2. Elija Índices de sitios en el panel de navegación izquierdo.
3. En los índices de Mis sitios, seleccione el enlace con el nombre del recurso de índice de sitios de destino.

API

Use la operación [DescribePlaceIndex](#) de las API de Amazon Location Places.

El siguiente ejemplo es una solicitud de API para obtener los detalles de los recursos del índice de lugares *ExamplePlaceIndex*.

```
GET /places/v0/indexes/ExamplePlaceIndex
```

A continuación, se muestra un ejemplo de respuesta para [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

Utilice el comando [describe-place-index](#).

El siguiente ejemplo es AWS CLI para obtener los detalles de los recursos del índice de sitios *ExamplePlaceIndex*.

```
aws location describe-place-index \
  --index-name "ExamplePlaceIndex"
```

Eliminar un recurso de índice de sitio

Puede eliminar un recurso de índice de lugar de su cuenta de AWS mediante la consola de ubicación de Amazon, el AWS CLI, o las API de ubicación de Amazon:

Console

Para eliminar un recurso de índice de ubicación mediante la consola de Amazon Location

Warning

Esta operación elimina el recurso de forma permanente.

1. Abra la consola de Amazon Location en <https://console.aws.amazon.com/location/>.
2. Elija Índices de sitios en el panel de navegación izquierdo.
3. Bajo índice de Mi sitio, seleccione el recurso de índice de lugar de destino.
4. Elija Eliminar índice de sitios.

API

Use la operación [DeletePlaceIndex](#) de las API de Amazon Location Places.

El siguiente ejemplo es una solicitud de API para eliminar el recurso del índice de lugares *ExamplePlaceIndex*.

```
DELETE /places/v0/indexes/ExamplePlaceIndex
```

A continuación se muestra un ejemplo de una respuesta correcta para [DeletePlaceIndex](#):

```
HTTP/1.1 200
```

CLI

Utilice el comando [delete-place-index](#).

El siguiente ejemplo es un AWS CLI comando para eliminar el recurso de índice de sitios *ExamplePlaceIndex*.

```
aws location delete-place-index \  
  --index-name "ExamplePlaceIndex"
```

Calcular rutas utilizando Amazon Location Service

Amazon Location le permite seleccionar un proveedor de datos para calcular una ruta mediante la creación y configuración de un recurso de calculadora de rutas.

Puede utilizar el recurso de la calculadora de rutas para [calcular una ruta](#) con parámetros específicos mediante el SDK AWS o los puntos de conexión de la API de REST. Utilice este recurso de calculadora de rutas para calcular las rutas entre un origen, un destino y hasta 23 puntos de referencia para diferentes modos de transporte, evitaciones y condiciones del tráfico.

También puede utilizar el recurso de la calculadora de rutas para crear entradas para sus algoritmos o productos de planificación de rutas mediante el [cálculo de una matriz de rutas](#). Calcule el tiempo de viaje y la distancia de viaje entre un conjunto de posiciones de salida y un conjunto de posiciones de destino. El software de planificación de rutas puede utilizar esos datos de tiempo y distancia para optimizar una ruta o un conjunto de rutas; por ejemplo, si está planificando varias rutas de entrega y desea encontrar la mejor ruta y hora para cada parada. Puede calcular una matriz de rutas para diferentes modos de transporte, puntos a evitar y condiciones de tráfico.

Note

Para obtener una descripción general de los conceptos de rutas, consulte [Rutas](#).

Temas

- [Requisitos previos](#)
- [Calcule una ruta](#)
- [Planificación de rutas con una matriz de rutas](#)
- [Posiciones no ubicadas en una carretera](#)
- [Hora de salida](#)
- [Modo de viaje](#)
- [Administrar los recursos de la calculadora de rutas](#)

Requisitos previos

Antes de empezar a calcular las rutas, siga los siguientes pasos previos:

Temas

- [Creación de un recurso de la calculadora de rutas](#)
- [Autenticación de sus solicitudes](#)

Creación de un recurso de la calculadora de rutas

Para poder calcular una ruta, cree un recurso de la calculadora de rutas en su cuenta AWS.

Al crear un recurso de calculadora de rutas, puede elegir entre los proveedores de datos disponibles:

1. Esri: para obtener información adicional sobre la cobertura de Esri en la región de su interés, consulte [Información de Esri sobre la cobertura de redes de calle y tráfico](#).
2. HERE Technologies: para obtener información adicional sobre la cobertura de HERE en la región de su interés, consulte [Cobertura de enrutamiento de automóviles de HERE](#) y [Cobertura de enrutamiento de camiones de HERE](#).
3. Grab: para obtener más información sobre la cobertura de Grab, consulte [Países, regiones y áreas cubiertas](#).

Note

Si su aplicación rastrea o enruta los activos que utiliza en su empresa, como vehículos de reparto o empleados, no debe utilizar Esri como proveedor de geolocalización. Consulte la sección 82 de los [Términos de servicio de AWS](#) para obtener más detalles.

Para ello, puede utilizar la consola de Amazon Location Service, la de AWS CLI o las API de Amazon Location.

Console

Para crear un recurso de calculadora de rutas mediante la consola de ubicación de Amazon

1. Abra la consola de Amazon Location en <https://console.aws.amazon.com/location/>.
2. En el panel de navegación izquierdo, elija Calculadoras de rutas.
3. Elija Crear calculadora de rutas.
4. Complete las casillas siguientes:
 - Nombre: introduzca un nombre para el recurso de calculadora de rutas. Por ejemplo, *ExampleCalculator*. 100 caracteres como máximo. Entre las entradas válidas se incluyen caracteres alfanuméricos, guiones, puntos y guiones bajos.
 - Descripción : escriba una descripción opcional.
5. En el caso de los proveedores de datos, elija un [proveedor de datos](#) para usarlo como calculadora de rutas.
6. (Opcional) En Etiquetas, escriba una Clave y un Valor de etiqueta. Esto añade una etiqueta a su nuevo recurso de calculadora de rutas. Para obtener más información, consulte [Etiquetado de los recursos](#).

7. Elija Crear calculadora de rutas.

API

Para crear un recurso de calculadora de rutas mediante las API de Amazon Location

Use la operación [CreateRouteCalculator](#) de las API de Amazon Location Places.

El siguiente ejemplo es una solicitud de API para crear un recurso de calculadora de rutas denominado *ExampleCalculator* mediante el proveedor de datos *Esri*.

```
POST /routes/v0/calculators
Content-type: application/json

{
  "CalculatorName": "ExampleCalculator",
  "DataSource": "Esri",
  "Description": "string",
  "Tags": {
    "Tag1" : "Value1"
  }
}
```

AWS CLI

Para crear un recurso de la calculadora de rutas utilizando comandos AWS CLI

Utilice el comando `create-route-calculator`.

En el siguiente ejemplo, se crea un recurso de calculadora de rutas denominado *ExampleCalculator* usar *Esri* como proveedor de datos.

```
aws location \
  create-route-calculator \
  --calculator-name "ExampleCalculator" \
  --data-source "Esri" \
  --tags Tag1=Value1
```

Note

La facturación depende del uso. Puede incurrir en cargos por el uso de otros servicios de AWS. Para obtener más información, consulte los [precios de Amazon Location Service](#).

Autenticación de sus solicitudes

Una vez que haya creado un recurso de calculadora de rutas y esté listo para empezar a incorporar funciones de localización en su aplicación, elija cómo autenticaría sus solicitudes:

- Para explorar las formas en las que puede acceder a los servicios, consulte [Cómo acceder a Amazon Location Service](#).
- Si tiene un sitio web con usuarios anónimos, puede utilizar claves de API o Amazon Cognito.

Ejemplo

En el siguiente ejemplo, se muestra el uso de una clave de API para la autorización, el [JavaScriptSDK v3 de AWS](#) y la ubicación [JavaScript Ayudante de autenticación](#) de Amazon.

```
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);
```

Calcule una ruta

Puede usar Amazon Location Service para calcular las rutas entre un origen y un destino, con hasta 23 puntos de referencia a lo largo de la ruta, para diferentes modos de transporte, puntos a evitar y condiciones del tráfico.

Note

Primero debe crear un recurso de calculadora de rutas y configurar la autenticación para sus solicitudes a Amazon Location. Para obtener más información, consulte [Requisitos previos](#).

Empiece a calcular las rutas

Envíe una solicitud sencilla mediante la operación [CalculateRoute](#). Una solicitud simple contiene los siguientes campos obligatorios:

- `DeparturePosition`: la posición inicial desde la que se calcula la ruta. Definida como `[longitude, latitude]`
- `DestinationPosition`: la posición final en la que se calculará la ruta. Definida como `[longitude, latitude]`.

Note

Si especifica una posición de salida o destino que no se encuentra en una carretera, Amazon Location [mueve la posición a la carretera más cercana](#).

Si lo desea, puede [especificar puntos](#) de referencia, una [hora de salida](#) y un [modo de viaje](#) en su solicitud.

Puede usar AWS CLI o las API de Amazon Location.

API

El siguiente ejemplo es una `CalculateRoute` solicitud que utiliza el recurso `ExampleCalculator` de la calculadora de rutas. La solicitud especifica el cálculo de una ruta desde una posición de salida `[-122.7565, 49.0021]` hasta una posición de destino `[-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

El siguiente ejemplo es un `calculate-route` comando que utiliza el recurso de calculadora de rutas `ExampleCalculator`. La solicitud especifica el cálculo de una ruta desde una posición de salida `[-122.7565, 49.0021]` hasta una posición de destino `[-122.3394, 47.6159]`.

```
aws location \
  calculate-route \
    --calculator-name ExampleCalculator \
    --departure-position -122.7565 49.0021 \
    --destination-position -122.3394 47.6159
```

De forma predeterminada, la respuesta muestra `Distance` en kilómetros. Puede cambiar la unidad de medida a millas mediante el siguiente parámetro opcional:

- `DistanceUnit`: especifica el sistema de unidades que se utilizará para los resultados de distancia.

Example

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route
Content-type: application/json
{
  "DeparturePosition": [-122.7565,49.0021],
  "DestinationPosition": [-122.3394, 47.6159],
  "DistanceUnit": "Miles"
}
```

```
}
```

Establecer puntos de referencia

Al calcular una ruta, puede especificar hasta 23 puntos de escala intermedios entre la posición de salida y la posición de destino utilizando las posiciones de los puntos de referencia.

- **WaypointPositions**: especifica una lista ordenada de posiciones intermedias para incluirlas a lo largo de una ruta entre la posición de salida y la posición de destino.

Note

Si especifica una posición de punto de referencia que no se encuentra en una carretera, Amazon Location mueve la posición a la carretera más cercana.

Example

La siguiente solicitud [CalculateRoute](#) calcula una ruta con 2 puntos de referencia:

- La posición de salida es [-122.7565, 49.0021] y la posición de destino es [-122.3394, 47.6159].
- Para el parámetro de solicitud **WaypointPositions**:
 - La primera posición de parada es [-122.1884, 48.0936].
 - La segunda posición de parada es [-122.3493, 47.6205].
- Para incluir la geometría de la cadena de líneas del tramo entre estos dos puntos de referencia, defina el siguiente parámetro opcional en **true**:
 - **IncludeLegGeometry**: incluye la geometría de cada ruta entre un par de posiciones de la respuesta.

```
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
}
```

Ejemplo de respuesta

El siguiente es un ejemplo de solicitud con la respuesta correspondiente cuando se llama a la operación [CalculateRoute](#) desde la API Amazon Location Routes con el valor `IncludeLegGeometry` en `true`, que incluye la geometría de cadena de líneas de cada ruta entre un par de posiciones de la respuesta.

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,
  "RouteBBox": [
    -122.7565,49.0021,
    -122.3394,47.6159
  ]
}
}
```

Planificación de rutas con una matriz de rutas

Puede utilizar Amazon Location Service para crear entradas para su software de planificación y optimización de rutas. Puede crear resultados de ruta, incluidos el tiempo y la distancia de viaje, para las rutas entre un conjunto de posiciones de salida y un conjunto de posiciones de destino.

Por ejemplo, dadas las posiciones de salida A y B y las posiciones de destino X e Y, Amazon Location Service mostrará el tiempo y la distancia de viaje para las rutas de A a X, A a Y, B a X y B a Y.

Puede calcular las rutas con diferentes modos de transporte, evasiones y condiciones del tráfico. Por ejemplo, puede especificar que el vehículo es un camión de 15 metros de largo y la ruta calculada utilizará esas restricciones para determinar el tiempo y la distancia de viaje.

El número de resultados mostrados (y de rutas calculadas) es el número de posiciones de salida multiplicado por el número de posiciones de destino. Se le cobrará por cada ruta calculada, no por cada solicitud al servicio, por lo que una matriz de rutas con 10 salidas y 10 destinos se facturará como 100 rutas.

Calcular una matriz de rutas

Puede calcular una matriz de rutas entre un conjunto de posiciones de salida y un conjunto de posiciones de destino. Los resultados de la ruta incluirán el tiempo de viaje y la distancia de viaje.

Requisito previo

- Primero debe crear un recurso de calculadora de rutas y configurar la autenticación para sus solicitudes a Amazon Location. Para obtener más información, consulte [Requisitos previos](#).

Envíe una solicitud mediante la operación [CalculateRouteMatrix](#). Una solicitud simple contiene los siguientes campos obligatorios:

- **DeparturePositions**: el conjunto de posiciones de partida de las cuales calcular las rutas. Definido como una matriz de [longitude, latitude]
- **DestinationPositions**: el conjunto de posiciones de destino de las cuales calcular las rutas. Definido como una matriz de [longitude, latitude].

Note

Si especifica una posición de salida o destino que no se encuentra en una carretera, Amazon Location [mueve la posición a la carretera más cercana](#).

Opcionalmente, puede especificar una [hora de salida](#) y un [modo de viaje](#) en su solicitud.

Puede usar AWS CLI o las API de Amazon Location.

API

El siguiente ejemplo es una `CalculateRouteMatrix` solicitud que utiliza el recurso de calculadora de rutas *ExampleCalculator*. La solicitud especifica el cálculo de la matriz de rutas desde las posiciones de partida `[-122.7565, 49.0021]` y `[-122.2014, 47.6101]` hasta las posiciones de destino `[-122.3394, 47.6159]` and `[-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

El siguiente ejemplo es un `calculate-route-matrix` comando que utiliza el recurso de calculadora de rutas *ExampleCalculator*. La solicitud especifica el cálculo de la matriz de rutas desde las posiciones de partida `[-122.7565, 49.0021]` y `[-122.2014, 47.6101]` hasta las posiciones de destino `[-122.3394, 47.6159]` and `[-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]]"
```

De forma predeterminada, la respuesta muestra `Distance` en kilómetros. Puede cambiar la unidad de medida a millas mediante el siguiente parámetro opcional:

- `DistanceUnit`: especifica el sistema de unidades que se utilizará para los resultados de distancia.

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"  
}
```

Restricciones en las posiciones de salida y destino

Al calcular una matriz de rutas, existen restricciones en las posiciones de salida y destino. Estas restricciones varían según el proveedor utilizado por el recurso RouteCalculator.

Limitación	Esri	Grab	AQUÍ
Número de posiciones	Hasta 10 posiciones de salida y 10 posiciones de destino.	Hasta 350 posiciones de salida y 350 posiciones de destino.	Hasta 350 posiciones de salida y 350 posiciones de destino. Para rutas más largas, se aplican restricciones adicionales. Consulte la sección .
Distancia entre posiciones	Cualquier par de posiciones de salida y destino deben estar a menos de 400 km una de la otra (40 km para las rutas de senderismo).		Todas las posiciones de salida y destino deben estar dentro de un círculo de 180 km de diámetro. Para rutas más largas, se aplican restricciones adicionales. Consulte la sección .
Longitud de la ruta	Las rutas no se completarán si el tiempo total de viaje de la ruta es superior a 400 minutos.		No se calcularán las rutas que se desvíen más de 10 km fuera de un círculo alrededor de los puntos de salida y destino. Para rutas más largas, se aplican

Limitación	Esri	Grab	AQUÍ
			restricciones adicionales. Consulte la sección .
Regiones	En Corea no se admite el cálculo de una matriz de rutas.	Disponible en el sudeste asiático. Para obtener una lista de los países o regiones compatibles y más información, consulte Países, regiones y áreas cubiertas .	Sin restricciones adicionales.

Planificación de rutas más larga

Calcular una matriz de resultados de rutas es útil para una planificación eficiente de la ruta, pero el cálculo puede llevar algún tiempo. Todos los proveedores de datos de Amazon Location Service limitan el número de rutas o la distancia de las rutas que se pueden calcular. Por ejemplo, HERE permite crear rutas entre 350 posiciones de salida y destino, pero esas posiciones deben estar dentro de un círculo de 180 km. ¿Qué hacer si quiere planificar rutas más largas?

Puede calcular una matriz de rutas con longitudes ilimitadas para un número menor de rutas utilizando `RouteCalculator` con HERE como proveedor de datos. Esto no cambia la forma en que llama a la API [CalculateRouteMatrix](#), Amazon Location simplemente permite rutas más largas si cumple con los requisitos.

Los requisitos para calcular rutas de mayor longitud son:

- `RouteCalculator` debe utilizar el proveedor de datos HERE.
- El número de posiciones de salida no debe ser superior a 15.
- El número total de rutas a calcular no debe ser superior a 100.
- No se permiten rutas de larga distancia para camiones que eviten peajes cuando las rutas superen los 1000 km. Esta combinación es más lenta de calcular y puede provocar que se agote el tiempo de espera de la llamada. Puede calcular estas rutas de forma individual con la [CalculateRoute](#) operación.

Si su llamada no cumple estos requisitos (por ejemplo, solicita 150 cálculos de rutas en una sola llamada), `CalculateRouteMatrix` volverá a permitir solo las reglas de ruta más cortas. A continuación, podrá calcular las rutas, siempre que las posiciones estén dentro de un círculo de 180 km.

Al calcular rutas más largas, tenga en cuenta lo siguiente:

- Las rutas más largas pueden tardar más en calcularse, incluso más que el tiempo máximo de las API de ubicación de Amazon. Si tiene tiempos de espera frecuentes con rutas específicas, puede probar con un número menor de rutas en cada destino de llamada a `CalculateRouteMatrix`.
- Si añade más puntos de destino o de salida a su solicitud `CalculateRouteMatrix`, la operación puede cambiar al modo más restringido y puede obtener un error para una ruta que se puede calcular sin problemas cuando hay menos rutas que crear. En este caso, reduzca el número de posiciones de destino o salida y realice varias solicitudes para obtener el conjunto completo de cálculos de rutas que necesita.

Ejemplo de respuesta

El siguiente es un ejemplo de solicitud con la respuesta correspondiente al llamar a la operación [CalculateRouteMatrix](#) desde la API Amazon Location Routes.

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"
  }
}

```

Posiciones no ubicadas en una carretera

Al utilizar `CalculateRoute` o `CalculateRouteMatrix` especificar una posición de salida, destino o punto de referencia que no esté ubicada en una carretera, Amazon Location mueve la posición a una carretera cercana.

La siguiente solicitud [CalculateRoute](#) especifica una posición de salida y una posición de destino que no se encuentran en una carretera:

```

POST /routes/v0/calculators/ExampleCalculator/calculate/route
Content-type: application/json
{
  "DeparturePosition": [-123.128014, 49.298472],

```



```
"DestinationPosition": [-123.134701, 49.294315]
}
```

La respuesta resultante muestra una posición que se ajusta a una carretera cercana:

```
{
  "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"
}
}

```

Hora de salida

De forma predeterminada, cuando llama `CalculateRoute` o `CalculateRouteMatrix` si no especifica una hora de salida en la solicitud, las rutas calculadas reflejan las condiciones óptimas del tráfico.

Puede establecer una hora de salida específica para utilizar las condiciones de tráfico predictivas y en tiempo real del proveedor de datos que elija mediante una de las siguientes opciones:

- **DepartNow**: cuando se establece en *true*, utiliza las condiciones del tráfico en tiempo real para calcular la ruta de viaje más rápida.
- **DepartureTime**: cuando se proporciona, utiliza condiciones de tráfico predictivas y conocidas para el tiempo solicitado. Definido en el siguiente [formato](#): `YYYY-MM-DDThh:mm:ss.sssZ`.

Example

La siguiente solicitud [CalculateRoute](#) establece la hora de salida para el 2 de julio de 2024 a las 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,
}
```

Modo de viaje

Puede configurar un modo de viaje cuando utilice `CalculateRoute` o `CalculateRouteMatrix`. El modo de viaje afecta a la velocidad de viaje y a la compatibilidad con las carreteras. Si bien el modo de viaje predeterminado es el coche, puede especificar qué modo de viaje va a utilizar mientras recorre una ruta con el siguiente parámetro opcional:

- **TravelMode**: especifica el modo de transporte al calcular una ruta, como: *BicycleCar*, *Motorcycle*, *Truck*, o *Walking*.

Limitaciones

- Si especifica el modo de viaje `Walking` y su proveedor de datos es Esri, el inicio y el destino deben estar a menos de 40 km.
- `Bicycle` o solo `Motorcycle` están disponibles cuando se utiliza Grab como proveedor de datos.
- Grab ofrece solo `Bicycle` y `Walking` rutas en determinadas ciudades. Para obtener más información, consulte [Países, regiones y áreas cubiertas](#).
- `Truck` no está disponible cuando se utiliza Grab como proveedor de datos.

Preferencias adicionales

Si especifica una `TravelMode` de `Car`, puede especificar preferencias de ruta adicionales con el siguiente parámetro opcional:

- `CarModeOptions`: especifica las preferencias de ruta cuando se viaja en coche, como `AvoidFerries` o `AvoidTolls`.

Si especifica una `TravelMode` de `Truck`, puede especificar preferencias de ruta adicionales con el siguiente parámetro opcional:

- `TruckModeOptions`: especifica las preferencias de ruta cuando se viaja en un camión, por ejemplo `AvoidFerries` o `AvoidTolls`, además de especificar las rutas que pueden adaptarse a las direcciones `TruckDimensions` y `TruckWeight`.

Example

La siguiente solicitud [CalculateRoute](#) especifica `Truck` como el modo de viaje. Las restricciones de ruta adicionales incluyen: evitar las rutas que utilizan transbordadores y evitar las carreteras en las que no se puedan acomodar las dimensiones y el peso del camión.

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

Administrar los recursos de la calculadora de rutas

Puede gestionar los recursos de su calculadora de rutas desde la consola de Amazon Location, la AWS CLI o las API de Amazon Location.

Enumere sus recursos de calculadora de rutas

Puede ver su lista de calculadora de rutas desde la consola de Amazon Location, la AWS CLI o las API de Amazon Location:

Console

Para ver una lista de calculadoras de rutas mediante la consola de Amazon Location

1. Abra la consola de Amazon Location en <https://console.aws.amazon.com/location/>.
2. En el panel de navegación de la izquierda, elija Calculadoras de rutas.
3. Consulte los detalles de la calculadora de rutas en Mis calculadoras de rutas.

API

Siga las instrucciones de [ListRouteCalculators](#) para las API de Amazon Location Routes.

El siguiente ejemplo es una solicitud de la API para obtener una lista de las calculadoras de rutas de la cuenta AWS.

```
POST /routes/v0/list-calculators
```

A continuación, se muestra un ejemplo de respuesta para [ListRouteCalculators](#):

```
{
```

```
"Entries": [  
  {  
    "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

Utilice el comando `list-route-calculators`.

El siguiente ejemplo de AWS CLI sirve para obtener una lista de las calculadoras de rutas de la cuenta AWS.

```
aws location list-route-calculators
```

Obtenga los detalles de la calculadora de rutas

Puede obtener detalles sobre cualquier recurso de su calculadora de rutas de su cuenta AWS desde la consola de ubicaciones de Amazon, la AWS CLI o las API de Amazon Location:

Console

Para ver los detalles de una calculadora de rutas desde la consola de Amazon Location

1. Abra la consola de Amazon Location en <https://console.aws.amazon.com/location/>.
2. En el panel de navegación de la izquierda, elija Calculadoras de rutas.
3. En Mis calculadoras de rutas, seleccione el enlace con el nombre de la calculadora de rutas de destino.

API

Siga las instrucciones de [DescribeRouteCalculator](#) para las API de Amazon Location Routes.

El siguiente ejemplo es una solicitud de API para obtener los detalles de la calculadora de rutas *ExampleCalculator*.

```
GET /routes/v0/calculators/ExampleCalculator
```

A continuación, se muestra un ejemplo de respuesta para [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

Utilice el comando `describe-route-calculator`.

El siguiente ejemplo es AWS CLI para obtener los detalles de la calculadora de rutas *ExampleCalculator*.

```
aws location describe-route-calculator \
  --calculator-name "ExampleCalculator"
```

Eliminar una calculadora de ruteo

Puede eliminar una calculadora de rutas de su cuenta AWS desde la consola de Amazon Location, la AWS CLI o las API de Amazon Location:

Console

Para eliminar una calculadora de rutas mediante la consola de Amazon Location

⚠ Warning

Esta operación elimina el recurso de forma permanente.

1. Abra la consola de Amazon Location en <https://console.aws.amazon.com/location/>.
2. En el panel de navegación de la izquierda, elija Calculadoras de rutas.
3. En Mis calculadoras de rutas, seleccione la calculadora de rutas de destino.
4. Seleccione Eliminar calculadora de rutas.

API

Siga las instrucciones de [DeleteRouteCalculator](#) para las API de Amazon Location Routes.

El siguiente ejemplo es una solicitud de API para eliminar la colección de geocercas.

ExampleCalculator

```
DELETE /routes/v0/calculators/ExampleCalculator
```

A continuación, se muestra un ejemplo de respuesta para [DeleteRouteCalculator](#):

```
HTTP/1.1 200
```

CLI

Utilice el comando `delete-route-calculator`.

El siguiente ejemplo es un AWS CLI comando para eliminar la colección de geocercas.


ExampleCalculator

```
aws location delete-route-calculator \  
    --calculator-name "ExampleCalculator"
```

Geocercar un área de interés mediante Amazon Location

Una aplicación de geocercado evalúa la posición de un dispositivo rastreado en relación con las áreas de interés previamente registradas. Esto permite tomar medidas en función de las

actualizaciones de posición. Por ejemplo, puede iniciar un evento que genere una notificación cuando un cliente que pidió café en su aplicación móvil esté cerca de una tienda.

 Note

Para obtener una descripción general de los conceptos de geocercado y rastreadores, consulte [Geocercas y rastreadores](#).

En esta sección de la guía se proporcionan step-by-step instrucciones para crear una aplicación de geoperimetraje con Amazon Location Service.

Información general sobre los pasos

1. Agregue geocercas alrededor de las áreas de interés y guárdelas en un recurso de colección de geocercas.
2. Comience a rastrear sus dispositivos objetivo y almacene el historial de ubicaciones del dispositivo en un recurso de rastreador.
3. Vincule su recurso de rastreador con su recurso de colección de geocercas para que las actualizaciones de ubicación del dispositivo se evalúen automáticamente comparándolas con todas sus geocercas.
4. Si no quiere utilizar los rastreadores de ubicaciones de Amazon para conservar el historial de ubicaciones de sus dispositivos, puede comparar las posiciones de los dispositivos directamente con sus recursos de colección de geocercas.

Una vez implementada la solución de geocercado, el recurso de colección de geocercas emite los siguientes eventos:

- ENTER: un dispositivo rastreado entra en una geocerca dentro de una colección de geocercas.
- EXIT: un dispositivo rastreado sale de una geocerca dentro de una colección de geocercas.

Puedes usar Amazon EventBridge para reaccionar ante los eventos enrutándolos a otro lugar.

Como alternativa al envío de actualizaciones a través de las API de Amazon Location Service desde cada dispositivo, puede usar MQTT para enviar actualizaciones de dispositivos.

En los temas siguientes se describen detalladamente estos pasos y estas alternativas.

Temas

- [Agregar geocercas](#)
- [Comenzar el rastreo](#)
- [Vincular un rastreador a una colección de geocercas](#)
- [Evalúe las posiciones de los dispositivos comparándolas con las geocercas](#)
- [Verifica las posiciones de los dispositivos](#)
- [Cómo reaccionar a los eventos de Amazon Location Service con Amazon EventBridge](#)
- [Rastreo mediante AWS IoT y MQTT con Amazon Location Service](#)
- [Administrar sus recursos de colección de geovallas](#)
- [Administrar los recursos de su rastreador](#)
- [Ejemplo de aplicación móvil de geofencing and Tracking](#)

Agregar geocercas

Las geocercas contienen puntos y vértices que forman un límite cerrado que define un área de interés. Las colecciones de geocercas almacenan y administran una o varias geocercas.

[Las colecciones de geocercas de Amazon Location almacenan las geocercas definidas mediante un formato de datos geoespaciales estándar denominado GeoJSON \(RFC 7946\)](#). Puede utilizar herramientas, como [geojson.io](#), sin coste alguno para dibujar sus geocercas gráficamente y guardar el archivo GeoJSON de salida.

Note

Amazon Location no admite polígonos con agujeros, multipolígonos, polígonos en el sentido de las agujas del reloj ni geocercas que crucen el antimeridiano.

Crear una colección de geocercas

Cree una colección de geovallas para almacenar y gestionar geocercas mediante la consola de ubicación de Amazon, las AWS CLI o las API de ubicación de Amazon.

Console

Para crear una colección de geocercas mediante la consola de Amazon Location

1. Abra la consola de Amazon Location Service en <https://console.aws.amazon.com/location/>.
2. En el panel de navegación izquierdo, elija Colecciones de geocercas.
3. Seleccione Crear colección de geocercas.
4. Complete las casillas siguientes:
 - Nombre: escriba un nombre único. Por ejemplo, *ExampleGeofenceCollection* 100 caracteres como máximo. Entre las entradas válidas se incluyen caracteres alfanuméricos, guiones, puntos y guiones bajos.
 - Descripción: introduzca una descripción opcional para diferenciar sus recursos.
5. En el EventBridge caso de la regla con CloudWatch como objetivo, puede crear una EventBridge regla opcional para empezar [a reaccionar ante los eventos de la geocerca](#). Esto permite a Amazon Location publicar eventos en Amazon CloudWatch Logs.
6. (Opcional) En Etiquetas, escriba una clave y un valor de etiqueta. Esto añade una etiqueta a su nueva colección de geocercas. Para obtener más información, consulte [Etiquetado de los recursos de Amazon Location Service](#).
7. (Opcional) En Cifrado de claves gestionadas por el cliente, puede optar por agregar una clave gestionada por el cliente. Esto añade una clave simétrica gestionada por el cliente que usted crea, posee y administra mediante el cifrado que AWS posee de forma predeterminada. Para obtener más información, consulte [Cifrado de datos en reposo](#).
8. Seleccione Crear colección de geocercas.

API

Para crear una colección de geocercas mediante las API de Amazon Location

Utilice la operación [CreateGeofenceCollection](#) desde las API de Amazon Location Geofences.

En el siguiente ejemplo, se utiliza una solicitud de API para crear una colección de geovallas llamada *ExampleGeofenceCollection*. La colección de geocercas está asociada a una [AWS KMS clave administrada por el cliente para](#) cifrar los datos del cliente.

```
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

Para crear una colección de geocercas mediante comandos AWS CLI

Utilice el comando [create-geofence-collection](#).

En el siguiente ejemplo se utiliza una AWS CLI para crear una colección de geovallas llamada *ExampleGeofenceCollection*. La colección de geovallas está asociada a una [AWS KMS clave administrada por el cliente para](#) cifrar los datos del cliente.

```
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

La facturación depende del uso. Puede incurrir en cargos por el uso de otros servicios de AWS . Para obtener más información, consulte los [precios de Amazon Location Service](#).

Dibujar geocercas

Ahora que ha creado su colección de geocercas, puede definir sus geocercas. Las geocercas se definen como un polígono o un círculo. Para dibujar una geocerca poligonal, puede utilizar una herramienta de edición de GeoJSON, como [geojson.io](#).

Para crear una geocerca como un círculo, debe definir el punto central del círculo y el radio. Por ejemplo, si desea crear una geocerca para recibir notificaciones cada vez que un dispositivo se encuentre a menos de 50 metros de una ubicación específica, debe utilizar la latitud y la longitud de esa ubicación y especificar el radio en 50 metros.

Con las API de Amazon Location Service, también puede agregar metadatos a su geocerca en forma de pares clave-valor. Pueden resultar útiles para almacenar información sobre la geocerca, como su tipo, u otra información específica de su aplicación. Puede utilizar estos metadatos al [Cómo reaccionar a los eventos de Amazon Location Service con Amazon EventBridge](#).

Agregar geocercas poligonales

En esta sección se describe la creación de geocercas poligonales

Dibujar geocercas con una herramienta de GeoJSON

Ahora que ha creado su colección de geocercas, puede definir las mediante una herramienta de edición de GeoJSON, como [geojson.io](#).

Para crear un archivo de GeoJSON

1. Abra una herramienta de edición de GeoJSON. Por ejemplo, [geojson.io](#).
2. Elija el icono Dibujar un polígono y dibuje su área de interés.
3. Seleccione Guardar y, a continuación, seleccione GeoJSON en el menú desplegable.

Coloque geocercas de GeoJSON en una colección de geocercas

Puede utilizar el archivo GeoJSON resultante para cargar sus geocercas mediante la consola de Amazon Location Service, las API de ubicación de Amazon AWS CLI o las API de ubicación de Amazon:

Console

Para agregar una geocerca a una colección de geocercas mediante la consola de Amazon Location Service

1. Abra la consola de Amazon Location Service en <https://console.aws.amazon.com/location/>.
2. En el panel de navegación izquierdo, elija Colecciones de geocercas.
3. En la lista de Colecciones de geocercas, seleccione el enlace con el nombre de la colección de geocercas de destino.
4. En Geocercas, elija Crear geocercas.
5. En la ventana Agregar geocercas, arrastre y suelte su GeoJSON en la ventana.
6. Seleccione Agregar geocercas.

API

Para agregar geocercas mediante las API de Amazon Location

Utilice la operación [PutGeofence](#) desde las API de Amazon Location Geofences.

En el siguiente ejemplo, se utiliza una solicitud de API para añadir una geocerca con el ID GEOFENCE-EXAMPLE1 a una colección *de geovallas* llamada.

ExampleGeofenceCollection También especifica una única propiedad de metadatos de geocerca con la clave Type y el valor 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]
      ]
    ]
  }
}
```

Como alternativa, puede agregar más de una geocerca mediante la operación

[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

Para agregar una geocerca a una colección de geocercas mediante comandos AWS CLI

Utilice el comando [put-geofence](#).

En el siguiente ejemplo, se usa una AWS CLI para agregar una geocerca a una colección de geocercas llamada *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"
}

```

agregar geocercas circulares

En esta sección se describe la creación de geocercas circulares. Debe conocer la latitud y la longitud del punto que desea que sea el centro del círculo y el radio en metros del círculo. Puede crear geocercas circulares con las API de Amazon Location o con la AWS CLI.

API

Para agregar geocercas circulares mediante las API de Amazon Location

Utilice la operación [PutGeofence](#) desde las API de Amazon Location Geofences.

En el siguiente ejemplo, se utiliza una solicitud de API para añadir una geocerca con el ID GEOFENCE-EXAMPLE2 a una colección de *geovallas* llamada: *ExampleGeofenceCollection*

```
PUT /geofencing/v0/collections/ExampleGeofenceCollection/geofence/GEOFENCE-EXAMPLE2
Content-type: application/json

{
  "Geometry": {
    "Circle": {
      "Center": [-5.716667, -15.933333],
      "Radius": 50
    }
  }
}
```

AWS CLI

Para añadir una geocerca circular a una colección de geocercas mediante comandos AWS CLI

Utilice el comando [put-geofence](#).

En el siguiente ejemplo, se usa una AWS CLI para agregar una geocerca a una colección de geocercas llamada. *ExampleGeofenceCollection*

```
$ aws location \
  put-geofence \
    --collection-name ExampleGeofenceCollection \
    --geofence-id ExampleGeofenceCircle \
    --geometry 'Circle={Center=[-5.716667, -15.933333], Radius=50}'
```

Note

También puede incluir JSON para geometrías complejas en su propio archivo, como en el siguiente ejemplo.


```
$ aws location \
  put-geofence \
    --collection-name ExampleGeofenceCollection \
    --geofence-id ExampleGeofenceCircle \
    --geometry file:circle.json
```

En el ejemplo, el archivo circle.json incluye JSON para la geometría circular.

```
{
  "Circle": {
    "Center": [-74.006975, 40.717127],
    "Radius": 287.7897969218057
  }
}
```

Comenzar el rastreo

Esta sección le guía a través de la creación de una aplicación de rastreo que captura las ubicaciones de los dispositivos.

Crear un rastreador

Cree un recurso de rastreador para almacenar y procesar las actualizaciones de posición de sus dispositivos. Puedes usar la consola de Amazon Location Service AWS CLI, las o las API de ubicación de Amazon.

Cada actualización de posición almacenada en los recursos de su rastreador puede incluir una medida de la precisión de la posición y hasta tres campos de metadatos sobre la posición o el dispositivo que desee almacenar. Los metadatos se almacenan como pares clave-valor y pueden almacenar información como la velocidad, la dirección, la presión de los neumáticos o la temperatura del motor.

Los rastreadores filtran las actualizaciones de posición a medida que se reciben. Esto reduce el ruido visual en las rutas de los dispositivos (lo que se denomina fluctuación) y reduce el número de eventos de entrada y salida falsos de la geocerca. Esto también ayuda a administrar los costos al reducir la cantidad de evaluaciones de geocercas iniciadas.

Los rastreadores ofrecen tres opciones de filtrado de posición para ayudar a gestionar los costos y reducir las interrupciones en las actualizaciones de ubicación.

- **Basado en la precisión:** utilícelo con cualquier dispositivo que proporcione una medición de la precisión. La mayoría de los dispositivos móviles proporcionan esta información. La precisión de cada medición de posición se ve afectada por muchos factores ambientales, como la recepción de los satélites GPS, el paisaje y la proximidad de dispositivos Wi-Fi y Bluetooth. La mayoría de los dispositivos, incluidos la mayoría de los dispositivos móviles, pueden proporcionar una estimación de la precisión de la medición junto con la medición. Mediante el filtrado `AccuracyBased`, Amazon Location ignora las actualizaciones de ubicación si el dispositivo se ha movido menos de la precisión medida. Por ejemplo, si dos actualizaciones consecutivas de un dispositivo tienen un rango de precisión de 5 m y 10 m, Amazon Location ignora la segunda actualización si el dispositivo se ha movido menos de 15 m. Amazon Location no evalúa las actualizaciones ignoradas comparándolas con las geocercas ni las almacena.

Si no se proporciona la precisión, se trata como cero y la medición se considera perfectamente precisa.

Note

También puede utilizar un filtrado basado en la precisión para eliminar todos los filtros. Si selecciona un filtrado basado en la precisión, pero sobrescribe todos los datos de precisión a cero u omite la precisión por completo, Amazon Location no filtrará ninguna actualización.

- **Basado en la distancia:** utilícelo cuando sus dispositivos no proporcionen una medición precisa, pero aun así desee aprovechar las ventajas del filtrado para reducir las fluctuaciones y gestionar los costos. El filtrado `DistanceBased` ignora las actualizaciones de ubicación en las que los dispositivos se han movido menos de 30 m (98,4 pies). Cuando utiliza el filtrado de posición `DistanceBased`, Amazon Location no evalúa estas actualizaciones ignoradas comparándolas con las geocercas ni las almacena.

La precisión de la mayoría de los dispositivos móviles, incluida la precisión media de los dispositivos iOS y Android, está dentro de los 15 m. En la mayoría de las aplicaciones, el filtrado `DistanceBased` puede reducir el efecto de las imprecisiones de ubicación al mostrar la trayectoria del dispositivo en un mapa, y el efecto rebote de varios eventos consecutivos de entrada y salida cuando los dispositivos se encuentran cerca del borde de una geocerca. También

puede ayudar a reducir el coste de la aplicación, ya que realiza menos llamadas para evaluar las geocercas asociadas o recuperar la posición de los dispositivos.

- Basado en el tiempo: (predeterminado) úselo cuando sus dispositivos envíen actualizaciones de posición con mucha frecuencia (más de una vez cada 30 segundos) y desee realizar evaluaciones de la geocerca prácticamente en tiempo real sin almacenar todas las actualizaciones. Durante el filtrado TimeBased, las actualizaciones de ubicación se evalúan con las recopilaciones de geocercas asociadas, pero no se almacenan todas las actualizaciones de ubicación. Si la frecuencia de actualización es superior a 30 segundos, solo se almacenará una actualización cada 30 segundos por cada ID de dispositivo único.

Note

Tenga en cuenta los costos de su aplicación de rastreo a la hora de decidir el método de filtrado y la frecuencia de las actualizaciones de posición. Se le facturará por cada actualización de ubicación y una vez por la evaluación de la actualización de posición con respecto a cada colección de geocercas vinculadas. Por ejemplo, si utiliza un filtrado basado en el tiempo, si su rastreador está vinculado a dos colecciones de geocercas, cada actualización de posición contará como una solicitud de actualización de ubicación y dos evaluaciones de la colección de geocercas. Si informa de las actualizaciones de posición de sus dispositivos cada 5 segundos y utiliza un filtrado basado en el tiempo, se le facturarán 720 actualizaciones de ubicación y 1440 evaluaciones de geocercas por hora para cada dispositivo.

La cantidad de geocercas de cada colección no afecta a su factura. Dado que cada colección de geocercas puede contener hasta 50 000 geocercas, tal vez desee combinar sus geocercas en un menor número de colecciones, siempre que sea posible, para reducir el costo de las evaluaciones de las geocercas.

De forma predeterminada, recibirá EventBridge los eventos cada vez que un dispositivo rastreado entre o salga de una geocerca vinculada. Para obtener más información, consulte [Vincular un rastreador a una colección de geocercas](#).

Puede activar los eventos para todas las actualizaciones de posición filtradas de un recurso de rastreo. Para obtener más información, consulte [Habilitar los eventos de actualización para un rastreador](#).

Note

Si deseas cifrar tus datos con tu propia clave gestionada por el AWS KMS cliente, la función de consultas a polígonos delimitadores estará desactivada de forma predeterminada.

Esto se debe a que, al utilizar esta función de consultas de polígonos delimitadores, una representación de las posiciones de los dispositivos no se cifrará con la clave gestionada.

AWS KMS Sin embargo, la posición exacta del dispositivo se sigue cifrando con la clave gestionada.

Puede optar por activar la característica de consultas de polígonos delimitadores configurando el parámetro `KmsKeyEnableGeospatialQueries` en «verdadero» al crear o actualizar un rastreador.

Console

Para crear un rastreador mediante la consola de Amazon Location

1. Abra la consola de Amazon Location Service en <https://console.aws.amazon.com/location/>.
2. En el panel de navegación izquierdo, elija Rastreadores.
3. Seleccione Crear rastreador.
4. Cumplimente los siguientes campos:
 - Nombre: escriba un nombre único. Por ejemplo, *ExampleTracker* 100 caracteres como máximo. Entre las entradas válidas se incluyen caracteres alfanuméricos, guiones, puntos y guiones bajos.
 - Descripción: escriba una descripción opcional.
5. En Filtrado de posición, elija la opción que mejor se adapte a la forma en que piensa utilizar el recurso de rastreo. Si no configura el Filtrado de posición, el ajuste predeterminado es TimeBased. Para obtener más información, consulte [Rastreadores](#) en esta guía y [PositionFiltering](#) en la referencia de la API de Amazon Location Service Trackers.
6. (Opcional) En Etiquetas, escriba una clave y un valor de etiqueta. Esto añade una etiqueta a su nueva colección de geocercas. Para obtener más información, consulte [Etiquetado de los recursos](#).
7. (Opcional) En Cifrado de claves gestionadas por el cliente, puede optar por agregar una clave gestionada por el cliente. Esto agrega una clave simétrica administrada por el cliente

que usted crea, posee y administra mediante el cifrado de AWS propiedad predeterminada. Para obtener más información, consulte [Cifrado de datos en reposo](#).

- (Opcional) En `KmsKeyEnableGeospatialQueries`, puede optar por habilitar las consultas geoespaciales. Esto le permite utilizar la característica Consultas de polígonos delimitadores y, al mismo tiempo, cifrar sus datos con una clave de AWS KMS gestionada por el cliente.

Note

Cuando utiliza la característica Consultas de polígonos delimitadores, no se cifra una representación de las posiciones de sus dispositivos con su clave gestionada de AWS KMS . Sin embargo, la posición exacta del dispositivo se sigue cifrando con la clave gestionada.

- (Opcional) En EventBridge la configuración, puede optar por habilitar los EventBridge eventos para las actualizaciones de posición filtradas. Esto enviará un evento cada vez que una actualización de posición de un dispositivo de este rastreador cumpla con la evaluación del filtrado de posición.
- Seleccione Crear rastreador.

API

Para crear un rastreador mediante las API de Amazon Location

Use la operación [CreateTracker](#) de las API de Amazon Location Trackers.

En el siguiente ejemplo, se utiliza una solicitud de API para crear un rastreador llamado *ExampleTracker*. El recurso de seguimiento está asociado a una [AWS KMS clave administrada por el cliente para cifrar los datos del cliente](#) y no [permite la entrada de actualizaciones de posición](#). 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"
}
}
```

Crear un rastreador con la opción **KmsKeyEnableGeospatialQueries** activada

El siguiente ejemplo tiene el parámetro `KmsKeyEnableGeospatialQueries` establecido en «verdadero». Esto le permite utilizar la función de consultas de polígonos delimitadores y, al mismo tiempo, cifrar los datos con una clave gestionada por el cliente. AWS KMS

Para obtener información sobre el uso de la característica Consultas de polígonos delimitadores, consulte [???](#)

 Note

Al utilizar la función de consultas de polígonos delimitadores, no se cifra una representación de las posiciones de los dispositivos con la clave gestionada. AWS KMS Sin embargo, la posición exacta del dispositivo se sigue cifrando con la clave gestionada.

```
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

Para crear un rastreador mediante comandos de AWS CLI

Utilice el comando [create-tracker](#).

En el siguiente ejemplo, se utiliza AWS CLI para crear un rastreador llamado. *ExampleTracker*. El recurso de seguimiento está asociado a una [AWS KMS clave gestionada por el cliente para cifrar los datos del cliente](#) y no [permite la entrada de actualizaciones de posición](#). 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"
```

Crear un rastreador con la opción **KmsKeyEnableGeospatialQueries** activada

El siguiente ejemplo tiene el parámetro `KmsKeyEnableGeospatialQueries` establecido en «verdadero». Esto le permite utilizar la función de consultas de polígonos delimitadores y, al mismo tiempo, cifrar los datos con una clave gestionada por el cliente. AWS KMS

Para obtener información sobre el uso de la característica Consultas de polígonos delimitadores, consulte [???](#)

Note

Al utilizar la función de consultas de polígonos delimitadores, no se cifra una representación de las posiciones de los dispositivos con la clave gestionada. AWS KMS Sin embargo, la posición exacta del dispositivo se sigue cifrando con la clave gestionada.

```
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

La facturación depende del uso. Puede incurrir en cargos por el uso de otros servicios de AWS . Para obtener más información, consulte los [precios de Amazon Location Service](#).

Puede editar la descripción, el filtrado de posición y la EventBridge configuración una vez creado el rastreador seleccionando Editar rastreador.

Autenticación de sus solicitudes

Una vez que haya creado un recurso de rastreo y esté listo para empezar a evaluar las posiciones de los dispositivos comparándolas con las geocercas, elija cómo autenticaría sus solicitudes:

- Para explorar las formas en las que puede acceder a los servicios, consulte [Cómo acceder a Amazon Location Service](#).
- Si desea publicar las posiciones de los dispositivos con solicitudes no autenticadas, puede utilizar Amazon Cognito.

Ejemplo

En el siguiente ejemplo, se muestra el uso de un grupo de identidades de Amazon Cognito para la autorización, el [JavaScript SDK v3 de AWS](#) y la ubicación de Amazon. [JavaScript Ayudante de autenticación](#)

```
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);
```

Actualizar el rastreador con la posición del dispositivo

Para rastrear sus dispositivos, puede publicar actualizaciones de la posición del dispositivo en su rastreador. Más adelante, puede recuperar estas posiciones de los dispositivos o el historial de posiciones del dispositivo desde su recurso de rastreador.

Cada actualización de posición debe incluir el ID del dispositivo, una marca de tiempo y una posición. Si lo desea, puede incluir otros metadatos, como la precisión y hasta 3 pares clave-valor para su propio uso.

Si su rastreador está asociado a una o más colecciones de geocercas, las actualizaciones se evaluarán en función de esas geocercas (siguiendo las reglas de filtrado que especificó para el rastreador). Si un dispositivo entra en un área geovallada (al moverse desde el interior del área hacia el exterior o viceversa), recibirá los eventos en su interior. EventBridge Estos eventos ENTER o EXIT incluyen los detalles de la actualización de la posición, como el ID del dispositivo, la marca de tiempo y cualquier metadato asociado.

Note

Para obtener más información sobre cómo filtrar posiciones, consulte [Crear un rastreador](#). Para obtener más información sobre los eventos de geocercas, consulte [Cómo reaccionar a los eventos de Amazon Location Service con Amazon EventBridge](#).

Use uno de estos métodos para enviar las actualizaciones del dispositivo:

- [Envía actualizaciones de MQTT](#) a un recurso de AWS IoT Core y vincúlalo a tu recurso de seguimiento.
- Envíe actualizaciones de ubicación mediante la API de Amazon Location Trackers, mediante la AWS CLI o las API de Amazon Location. Puede usar los [SDK de AWS](#) para llamar a las API desde su aplicación iOS o Android.

API

Para enviar una actualización de posición mediante las API de Amazon Location

Use la operación [BatchUpdateDevicePosition](#) de las API de Amazon Location Trackers.

En el siguiente ejemplo, se utiliza una solicitud de API para publicar una actualización de la posición del dispositivo en un rastreador *ExampleTracker*. *ExampleDevice*

```
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

Para enviar una actualización de posición mediante comandos de AWS CLI

Utilice el comando [batch-update-device-position](#).

En el siguiente ejemplo, se utiliza an AWS CLI para publicar una actualización de la posición del dispositivo para *ExampleDevice-1* y *ExampleDevice-2* en un rastreador *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"}}]'
```

Obtener el historial de ubicaciones de un dispositivo a partir de un rastreador

Su recurso de rastreador de ubicaciones de Amazon conserva el historial de ubicaciones de todos tus dispositivos rastreados durante un período de 30 días. Puedes recuperar el historial de ubicaciones del dispositivo, incluidos todos los metadatos asociados, desde su recurso de rastreador. Los siguientes ejemplos utilizan las AWS CLI, o las API de ubicación de Amazon.

API

Para obtener el historial de ubicaciones del dispositivo desde un rastreador mediante las API de Amazon Location

Use la operación [GetDevicePositionHistory](#) de las API de Amazon Location Trackers.

En el siguiente ejemplo, se utiliza una solicitud de URI de API para obtener el historial de ubicaciones del dispositivo *ExampleDevice* desde un rastreador llamado que *ExampleTracker* comienza desde 19:05:07 (inclusive) y termina en 19:20:07 (exclusivo) el 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

Para obtener el historial de ubicaciones del dispositivo desde un rastreador mediante AWS CLI comandos

Utilice el comando [get-device-position-history](#).

En el siguiente ejemplo, se utiliza un AWS CLI para obtener el historial de ubicaciones del dispositivo *ExampleDevice* desde un rastreador llamado que *ExampleTracker* comienza desde 19:05:07 (inclusive) y termina en 19:20:07 (exclusivo) en 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"
```

Enumerar las posiciones de sus dispositivos

Puedes ver una lista de las posiciones de los dispositivos de un rastreador mediante la AWS CLI API o las API de Amazon Location con la ListDevicePositions API. Cuando llamas a la ListDevicePositions API, se muestra una lista de las últimas posiciones de todos los dispositivos asociados a un rastreador determinado. De forma predeterminada, esta API devuelve 100 de las últimas posiciones de los dispositivos por página de resultados de un rastreador determinado. Para devolver solo los dispositivos de una región específica, use el parámetro FilterGeometry para crear una consulta de polígonos delimitadores. De esta forma, cuando llames ListDevicePositions, solo se devolverán los dispositivos que se encuentren dentro del polígono.

Note

Si deseas cifrar tus datos con tu propia clave gestionada por el AWS KMS cliente, la función de consultas a polígonos delimitadores estará desactivada de forma predeterminada. Esto se debe a que, al utilizar esta función, una representación de las posiciones de sus dispositivos no se cifrará con la clave gestionada. AWS KMS Sin embargo, la posición exacta del dispositivo sigue cifrada con su clave gestionada. Puede optar por activar la característica Consultas de polígonos delimitadores. Esto se hace configurando el parámetro KmsKeyEnableGeospatialQueries en «verdadero» al crear o actualizar un rastreador.

API

Use la operación [ListDevicePositions](#) de las API de Amazon Location Trackers.

El siguiente ejemplo es una solicitud de API para obtener una lista de las posiciones de los dispositivos en un área poligonal mediante el parámetro opcional [FilterGeometry](#). El ejemplo devuelve 3 ubicaciones de dispositivos presentes en el área definida por la matriz Polygon.

```
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"
}
```

A continuación se muestra un ejemplo de respuesta para [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

Utilice el comando [list-trackers](#).

El siguiente ejemplo sirve AWS CLI para obtener una lista de dispositivos en un área poligonal.

```
aws location list-device-positions TODO: add arguments add props for filter geo
```

Vincular un rastreador a una colección de geocercas

Ahora que tiene un conjunto de geocercas y un rastreador, puede vincularlos entre sí para que las actualizaciones de ubicación se evalúen automáticamente comparándolas con todas sus geocercas. Si no quiere evaluar todas las actualizaciones de ubicación o, si no va a almacenar algunas de sus ubicaciones en un recurso de rastreo, puede [comparar las posiciones de los dispositivos con respecto a las geocercas](#) cuando lo desee.

Cuando las posiciones de los dispositivos se comparan con las geocercas, se generan eventos. Puede establecer una acción para estos eventos. Para obtener más información sobre las acciones que puede configurar para los eventos de geocerca, consulte [Reaccionar ante los eventos de Amazon Location Service con Amazon EventBridge](#).

Un evento de Amazon Location incluye los atributos de la actualización de posición del dispositivo que lo genera y algunos atributos de la geocerca por la que se entra o se sale. Para obtener más información sobre los datos incluidos en un evento de geocercas, consulte [Ejemplos de EventBridge eventos de Amazon para Amazon Location Service](#).

Los siguientes ejemplos vinculan un recurso de seguimiento a una colección de geovallas mediante la consola, las API de ubicación de Amazon o las AWS CLI API de ubicación de Amazon.

Console

Para vincular un recurso de rastreo a una colección de geocercas mediante la consola de Amazon Location Service

1. Abra la consola de Amazon Location Service en <https://console.aws.amazon.com/location/>.
2. En el panel de navegación izquierdo, elija Rastreadores.
3. En Rastreadores de dispositivos, seleccione el enlace con el nombre del rastreador objetivo.
4. En Colecciones de geocercas vinculadas, elija Vincular colección de geocercas.
5. En la ventana Colección de geocercas vinculadas, seleccione una colección de geocercas en el menú desplegable.
6. Elija Vincular.

Tras vincular el recurso de rastreador, se le asignará el estado Activo.

API

Para vincular un recurso de rastreo a una colección de geocercas mediante las API de Amazon Location

Use la operación [AssociateTrackerConsumer](#) de las API de Amazon Location Trackers.

En el siguiente ejemplo, se utiliza una solicitud de API que se *ExampleTracker* asocia a una colección de geovallas mediante [su nombre de recurso de Amazon](#) (ARN).

```
POST /tracking/v0/trackers/ExampleTracker/consumers
Content-type: application/json

{
  "ConsumerArn": "arn:aws:geo:us-west-2:123456789012:geofence-
collection/ExampleGeofenceCollection"
}
```

AWS CLI

Para vincular un recurso de rastreo a una colección de geovallas mediante comandos AWS CLI

Utilice el comando [associate-tracker-consumer](#).

En el siguiente ejemplo, se utiliza un AWS CLI para crear una colección de geovallas llamada. *ExampleGeofenceCollection*

```
aws location \
  associate-tracker-consumer \
    --consumer-arn "arn:aws:geo:us-west-2:123456789012:geofence-
collection/ExampleGeofenceCollection" \
    --tracker-name "ExampleTracker"
```

Evalúe las posiciones de los dispositivos comparándolas con las geocercas

Hay dos formas de evaluar las posiciones frente a las geocercas para generar eventos de geocercas:

- Puede vincular los Rastreadores y las Colecciones de geocercas. Para obtener más información, consulte la sección: [Vincular un rastreador a una colección de geocercas](#).
- Puede realizar una solicitud directa al recurso de recopilación de geocercas para evaluar una o más posiciones mediante la API. [BatchEvaluateGeofences](#)

Además, puede pronosticar los eventos de geocerca entrantes para un dispositivo que entre, salga o permanezca inactivo dentro de una geocerca. Utilice la API para pronosticar eventos [ForecastGeofenceEvents](#).

Si también quiere rastrear el historial de ubicaciones de su dispositivo o mostrar las ubicaciones en un mapa, vincule el rastreador a una colección de geocercas. Como alternativa, es posible que no desee evaluar todas las actualizaciones de ubicación o que no tenga intención de almacenar los datos de ubicación en un recurso de rastreo. En cualquiera de estos casos, puede hacer una solicitud directa a la colección de geocercas y evaluar las posiciones de uno o más dispositivos comparándolas con sus geocercas.

La evaluación de las posiciones de los dispositivos con respecto a las geocercas genera eventos. Puede reaccionar ante estos eventos y enviarlos a otros AWS servicios. Para obtener más información sobre las acciones que puede realizar al recibir eventos de geocerca, consulte [Reaccionar ante los eventos de Amazon Location Service con Amazon EventBridge](#).

Un evento de Amazon Location incluye los atributos de la actualización de posición del dispositivo que lo genera, como la hora, la posición, la precisión y los metadatos clave-valor, además de algunos atributos de la geocerca por la que se entra o se sale. Para obtener más información sobre los datos incluidos en un evento de geocercas, consulte [Ejemplos de EventBridge eventos de Amazon para Amazon Location Service](#).

Los siguientes ejemplos utilizan las AWS CLI, o las API de ubicación de Amazon.

API

Para evaluar las posiciones de los dispositivos frente a la posición de las geocercas mediante las API de Amazon Location

Utilice la operación [BatchEvaluateGeofences](#) desde las API de Amazon Location Geofences.

En el siguiente ejemplo, se utiliza una solicitud de API para evaluar la posición del dispositivo *ExampleDevice* en una colección de geovallas asociada. *ExampleGeofenceCollection*. Sustituya estos valores por sus propios identificadores de geocercas y dispositivos.

```
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

Para evaluar las posiciones de los dispositivos con respecto a la posición de las geocercas mediante comandos AWS CLI

Utilice el comando [batch-evaluate-geofences](#).

En el siguiente ejemplo, se utiliza una AWS CLI para evaluar la posición de un conjunto de geocercas asociado con *ExampleDevice* respecto a él. *ExampleGeofenceCollection* Sustituya estos valores por sus propios identificadores de geocercas y dispositivos.

```
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"}}]'
```

La evaluación de las posiciones de los dispositivos con respecto a las geocercas genera eventos. Tradicionalmente, se puede reaccionar ante los eventos utilizando [Amazon EventBridge](#), pero este proceso solo le permite reaccionar ante los eventos después de que hayan ocurrido. Si necesita anticipar cuándo un dispositivo entra o sale de una geocerca, por ejemplo, si un dispositivo cruza una frontera y, en consecuencia, estará sujeto a una normativa diferente, puede utilizar la [ForecastGeofenceEvents](#) API para predecir eventos geográficos futuros.

La [ForecastGeofenceEvents](#) API utiliza criterios como la proximidad time-to-breach, la velocidad y la posición del dispositivo para predecir eventos. La API devolverá un `ForecastedBreachTime`, que indica el tiempo estimado en que se producirá el evento de geocerca.

En el siguiente ejemplo, se utilizan las API de ubicación de Amazon.

API

Para pronosticar eventos de geocercas mediante las API de ubicación de Amazon

Utilice la operación [ForecastGeofenceEvents](#) desde las API de Amazon Location Geofences.

En el siguiente ejemplo, se utiliza una solicitud de API para pronosticar los eventos de geofence de un pariente con respecto a un *ExampleDevice*. *ExampleGeofence* Sustituya estos valores por sus propios identificadores de geocercas y dispositivos.

```
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
}
```

Verifica las posiciones de los dispositivos

Para comprobar la integridad de la posición de un dispositivo, utilice la [VerifyDevicePosition](#) API. Esta API devuelve información sobre la integridad de la posición del dispositivo, evaluando propiedades como la señal del móvil del dispositivo, el punto de acceso Wi-Fi, la dirección Ipv4 y si se está utilizando un proxy.

Requisitos previos

Antes de poder utilizar las API enumeradas para la verificación de dispositivos, asegúrate de cumplir los siguientes requisitos:

- Ha creado un rastreador para el dispositivo o los dispositivos que desea comprobar. Para obtener más información, consulte [Comenzar el rastreo](#).

En el siguiente ejemplo, se muestra una solicitud para la [VerifyDevicePosition](#) API de Amazon Location.

API

Para verificar las posiciones de los dispositivos mediante las API de ubicación de Amazon

Usa la [VerifyDevicePosition](#) operación de las API de seguimiento de ubicaciones de Amazon.

El siguiente ejemplo muestra una solicitud de API para evaluar la integridad de la posición de un dispositivo. Sustituya estos valores por sus propios identificadores de dispositivo.

```
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

El SDK de integridad de ubicaciones proporciona funciones mejoradas relacionadas con la verificación de dispositivos y está disponible para su uso previa solicitud. Para acceder al SDK, póngase en contacto con [Sales Support](#).

Cómo reaccionar a los eventos de Amazon Location Service con Amazon EventBridge

Amazon EventBridge es un bus de eventos sin servidor que conecta las aplicaciones de manera eficiente mediante datos de AWS servicios como Amazon Location. EventBridge recibe eventos

de Amazon Location y dirige esos datos a objetivos como AWS Lambda. Puede configurar reglas de direccionamiento para determinar adónde enviar sus datos a fin de crear arquitecturas de aplicaciones que reaccionen en tiempo real.

De forma predeterminada, solo se envían los EXIT eventos geocercados (ENTER y los eventos cuando los dispositivos entran o salen de las áreas geocercadas). EventBridge También puede activar todos los eventos de actualización de posición filtrados para un recurso de seguimiento. Para obtener más información, consulte [Habilitar los eventos de actualización para un rastreador](#).

Para obtener más información, consulta [los eventos y patrones de eventos](#) en la Guía del EventBridge usuario de Amazon.

Temas

- [Habilitar los eventos de actualización para un rastreador](#)
- [Crea reglas de eventos para Amazon Location](#)
- [Ejemplos de EventBridge eventos de Amazon para Amazon Location Service](#)

Habilitar los eventos de actualización para un rastreador

De forma predeterminada, Amazon Location solo ENTER envía eventos EXIT geoperimetales a. EventBridge Puedes activar todos los UPDATE eventos de posición filtrados a los que se pueda enviar un rastreador. EventBridge Puede hacerlo al [crear](#) o [actualizar](#) un rastreador.

Por ejemplo, para actualizar un rastreador existente mediante el AWS CLI, puede utilizar el siguiente comando (utilice el nombre del recurso del rastreador en lugar de *MyTracker*).

```
aws location update-tracker --tracker-name MyTracker --event-bridge-enabled
```

Para desactivar los eventos de posición de un rastreador, debe usar la API o la consola de Amazon Location Service.

Crea reglas de eventos para Amazon Location

Puedes crear [hasta 300 reglas por bus de eventos](#) EventBridge para configurar las acciones que se tomen en respuesta a un evento de Amazon Location.

Por ejemplo, puede crear una regla para los eventos de geovalla en la que se envíe una notificación automática cuando se detecte un teléfono dentro del límite de una geovalla.

Para crear una regla para los eventos de Amazon Location

Con los siguientes valores, [crea una EventBridge regla](#) basada en los eventos de Amazon Location:

- En Tipo de regla, elija Regla con un patrón de evento.
- Agregue el siguiente patrón en el cuadro Patrón de evento:

```
{
  "source": ["aws.geo"],
  "detail-type": ["Location Geofence Event"]
}
```

Para crear una regla para las actualizaciones de posición del rastreador, puede utilizar este modelo:

```
{
  "source": ["aws.geo"],
  "detail-type": ["Location Device Position Event"]
}
```

Si lo desea, puede especificar solo los eventos de ENTER o EXIT añadiendo la etiqueta de `detail` (si su regla es para las actualizaciones de posición del rastreador, solo hay un `EventType`, así que, no es necesario filtrarlo):

```
{
  "source": ["aws.geo"],
  "detail-type": ["Location Geofence Event"],
  "detail": {
    "EventType": ["ENTER"]
  }
}
```

Si lo desea, también puede filtrar por propiedades de la posición o la geovalla:

```
{
  "source": ["aws.geo"],
  "detail-type": ["Location Geofence Event"],
  "detail": {
    "EventType": ["ENTER"],
    "GeofenceProperties": {
```



```
    "Type": "LoadingDock"
  },
  "PositionProperties": {
    "VehicleType": "Truck"
  }
}
```

- En Seleccionar objetivos, elija la acción objetivo que se realizará cuando se reciba un evento de Amazon Location Service.

Por ejemplo, utilice un tema de Amazon Simple Notification Service (SNS) para enviar un correo electrónico o un mensaje de texto cuando se produce un evento. Primero debe crear un tema de Amazon SNS mediante la consola de Amazon SNS. Para obtener más información, consulte [Uso de Amazon SNS para notificaciones de usuario](#).

Warning

Se recomienda confirmar que la regla de los eventos se ha aplicado correctamente o es posible que la acción automatizada no comience tal y como se esperaba. Para verificar la regla de eventos, inicie las condiciones para la regla de eventos. Por ejemplo, simule un dispositivo que entra en un área geovallada.

También puede capturar todos los eventos desde Amazon Location excluyendo la sección de `detail-type`. Por ejemplo:

```
{
  "source": [
    "aws.geo"
  ]
}
```

Note

El mismo evento puede entregarse más de una vez. Puede usar el identificador del evento para eliminar las duplicaciones de los eventos que reciba.

Ejemplos de EventBridge eventos de Amazon para Amazon Location Service

A continuación, se muestra un ejemplo de un evento de entrada de una geovalla iniciado por una llamada de: `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"
    }
  }
}
```

A continuación, se muestra un ejemplo de un evento de salida de una geovalla que se inicia con una llamada de: `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"
    }
  }
}

```

A continuación, se muestra un ejemplo de un evento de actualización de posición que se inicia con una llamada: `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"
  }
}
}
```

Rastreo mediante AWS IoT y MQTT con Amazon Location Service

[MQTT](#) es un protocolo de mensajería ligero y ampliamente adoptado diseñado para dispositivos restringidos. AWS IoT Core admite conexiones de dispositivos que utilizan el protocolo MQTT y el protocolo MQTT over WebSocket Secure (WSS).

[AWS IoT Core](#) conecta dispositivos a AWS y le permite enviar y recibir mensajes entre ellos. El motor de reglas AWS IoT Core almacena las consultas sobre los temas de los mensajes de sus dispositivos y le permite definir acciones para enviar mensajes a otros servicios de AWS, como Amazon Location Service. A los dispositivos que conocen su ubicación como coordenadas se les puede reenviar su ubicación a Amazon Location a través del motor de reglas.

Note

Los dispositivos pueden conocer su propia posición, por ejemplo, a través del GPS incorporado. AWS IoT también es compatible con el rastreo de la ubicación de dispositivos de terceros. Para obtener más información, consulte [AWS IoT Ubicación de dispositivos de núcleo](#) de la AWS IoT Guía para desarrolladores de núcleo.

En el siguiente tutorial se describe el rastreo mediante reglas de AWS IoT Core. También puede enviar la información del dispositivo a su propia función de AWS Lambda si necesita procesarla antes de enviarla a Amazon Location. Para obtener más información sobre el uso de Lambda para procesar las ubicaciones de sus dispositivos, consulte [Utilizar AWS Lambda con MQTT](#).

Temas

- [Requisito previo](#)
- [Crear una regla de AWS IoT Core](#)
- [Pruebe su regla de AWS IoT Core en la consola](#)
- [Utilizar AWS Lambda con MQTT](#)

Requisito previo

Antes de comenzar el rastreo, debe completar los siguientes requisitos previos:

- [Crear un recurso de rastreo](#) al que enviar los datos de ubicación del dispositivo.
- [Crear un rol de IAM](#) para conceder a AWS IoT Core acceso a su rastreador.

Cuando siga estos pasos, use la siguiente política para dar acceso a su rastreador:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "WriteDevicePosition",
      "Effect": "Allow",
      "Action": "geo:BatchUpdateDevicePosition",
      "Resource": "arn:aws:geo:*:*:tracker/*"
    }
  ]
}
```

```
}
```

Crear una regla de AWS IoT Core

A continuación, cree una regla de AWS IoT Core para reenviar la telemetría posicional de sus dispositivos a Amazon Location Service. Para obtener más información sobre la creación de reglas, consulte los siguientes temas en la AWS IoT Core Guía para desarrolladores:

- [Crear una regla de AWS IoT](#) para obtener información sobre cómo crear una nueva regla.
- [Acción de ubicación](#) para obtener información específica sobre la creación de una regla para publicar en Amazon Location

Pruebe su regla de AWS IoT Core en la consola

Si actualmente ningún dispositivo publica telemetría que incluya la ubicación, puede probar su regla con la consola de AWS IoT Core. La consola tiene un cliente de prueba en el que puede publicar un mensaje de muestra para comprobar los resultados de la solución.

1. [Inicie sesión en la AWS IoT Core consola en https://console.aws.amazon.com/iot/](https://console.aws.amazon.com/iot/).
2. En el menú de navegación de la izquierda, expanda Pruebas y elija Cliente de pruebas de MQTT.
3. En Publicar en un tema, defina el Nombre del tema como *iot/topic* (o el nombre del tema que haya configurado en su regla de AWS IoT Core, si es diferente) y proporcione lo siguiente para la Carga de mensajes. Sustituya la marca de tiempo *1604940328* por una marca de tiempo válida de los últimos 30 días (los rastreadores de Amazon Location Service ignoran las marcas de tiempo de más de 30 días).

```
{
  "payload": {
    "deviceid": "thing123",
    "timestamp": 1604940328,
    "location": { "lat": 49.2819, "long": -123.1187 },
    "accuracy": { "Horizontal": 20.5 },
    "positionProperties": { "field1": "value1", "field2": "value2" }
  }
}
```

4. Seleccione Publicar en un tema para enviar el mensaje de prueba.

5. Para validar que Amazon Location Service recibió el mensaje, use el siguiente comando AWS CLI. Si lo modificó durante la configuración, sustituya el nombre del rastreador por el que utilizó.

```
aws location batch-get-device-position --tracker-name MyTracker --device-ids  
thing123
```

Utilizar AWS Lambda con MQTT

Si bien el uso de AWS Lambda ya no es obligatorio para enviar datos de ubicación del dispositivo a Amazon Location para el rastreo, es posible que desee utilizar Lambda en algunos casos. Por ejemplo, si desea procesar usted mismo los datos de ubicación de su dispositivo antes de enviarlos a Amazon Location. En los temas siguientes se describe cómo utilizar Lambda para procesar los mensajes antes de enviarlos al rastreador. Para obtener más información sobre este patrón, consulte la [arquitectura de referencia](#).

Temas

- [Requisito previo](#)
- [Crear una función de Lambda](#)
- [Crear una regla de AWS IoT Core](#)
- [Pruebe su regla de AWS IoT Core en la consola](#)

Requisito previo

Antes de empezar a realizar el rastreo, debe [crear un recurso de rastreo](#). Para crear un recurso de rastreo, puede usar la consola de Amazon Location, AWS CLI o las API de Amazon Location.

En el siguiente ejemplo, se utiliza la consola de Amazon Location Service para crear el recurso de rastreo:

1. Abra la consola de Amazon Location Service en <https://console.aws.amazon.com/location/>.
2. En el panel de navegación izquierdo, elija Rastreadores.
3. Seleccione Crear rastreador.
4. Complete las casillas siguientes:

- Nombre: introduzca un nombre único que tenga un máximo de 100 caracteres. Entre las entradas válidas se incluyen caracteres alfanuméricos, guiones y guiones bajos. Por ejemplo, *MyTracker*.
- Descripción: escriba una descripción opcional. Por ejemplo, *Rastreador para almacenar las posiciones del dispositivo de AWS IoT Core*.
- Filtrado de posición: seleccione el filtro que desee usar para las actualizaciones de posición. Por ejemplo, Filtrado basado en la precisión.

5. Seleccione Crear rastreador.

Crear una función de Lambda

Para crear una conexión entre AWS IoT Core y Amazon Location Service, necesita una función de AWS Lambda que procese los mensajes reenviados por AWS IoT Core. Esta función extraerá todos los datos posicionales, los formateará para Amazon Location Service y los enviará a través de la API de Amazon Location Tracker. Puede crear esta función a través de la consola de AWS Lambda o puede usar AWS Command Line Interface (AWS CLI) o las API de AWS Lambda.

Para crear una función de Lambda que publique actualizaciones de posición en Amazon Location mediante la consola:

1. Abra la consola de AWS Lambda en <https://console.aws.amazon.com/lambda/>.
2. En el panel de navegación izquierdo, elija Funciones.
3. Elija Crear función y asegúrese de que esté seleccionada la opción Autor desde cero.
4. Complete las casillas siguientes:
 - Nombre de función: introduzca un nombre único para su función. Las entradas válidas incluyen caracteres alfanuméricos, guiones y guiones bajos sin espacios. Por ejemplo, *MyLambda*.
 - Tiempo de ejecución: elija *Python 3.8*.
5. Elija Crear función.
6. Elija la pestaña Código para abrir el editor.
7. Sobrescriba el código del marcador de posición `lambda_function.py` con lo siguiente y sustituya el valor asignado a `TRACKER_NAME` por el nombre del rastreador que creó como [requisito previo](#).


```
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. Seleccione Implementar para almacenar la función actualizada.
9. Elija la pestaña Configuración.
10. En la sección Permisos, elija el Nombre del rol con el hipervínculo para conceder permisos de Amazon Location Service a su función de Lambda.
11. En la página Resumen de su rol, seleccione Agregar permisos y, a continuación, en la lista desplegable, seleccione Crear política integrada.
12. Elija la pestaña JSON y sobrescriba la política con el siguiente documento. Esto permite que la función de Lambda actualice las posiciones de los dispositivos gestionadas por todos los recursos de rastreo en todas las regiones.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "WriteDevicePosition",
      "Effect": "Allow",
      "Action": "geo:BatchUpdateDevicePosition",
      "Resource": "arn:aws:geo:*:*:tracker/*"
    }
  ]
}

```

13. Elija Revisar política.

14. Escriba un nombre para la política. Por ejemplo, *AmazonLocationTrackerWriteOnly*.
15. Elija Crear política.

Puede modificar el código de esta función, según sea necesario, para adaptarlo al esquema de mensajes de su propio dispositivo.

Crear una regla de AWS IoT Core

A continuación, cree una regla de AWS IoT Core para reenviar la telemetría posicional de sus dispositivos a la función AWS Lambda para su transformación y publicación en Amazon Location Service. La regla de ejemplo proporcionada asume que cualquier transformación necesaria de las cargas útiles del dispositivo es gestionada por la función de Lambda. Puede crear esta regla a través de la consola de AWS IoT Core, la AWS Command Line Interface (AWS CLI) o las API de AWS IoT Core.

Note

Si bien la consola AWS IoT gestiona los permisos necesarios para permitir que AWS IoT Core invoque la función de Lambda, si va a crear la regla a partir de AWS CLI o SDK, debe [configurar una política para conceder permiso a AWS IoT](#).

Para crear un AWS IoT Core con la consola

1. Inicie sesión en la AWS IoT Core consola en <https://console.aws.amazon.com/iot/>.
2. En el panel de navegación izquierdo, expanda Actuar y, a continuación, elija Reglas.
3. Elija Crear una regla para iniciar el asistente de nueva regla.
4. Escriba un nombre y la descripción de la regla.
5. Para la Declaración de consulta de la regla, actualice el atributo FROM para que haga referencia a un tema en el que al menos un dispositivo publique telemetría que incluya la ubicación. Si está probando la solución, no es necesario modificarla.

```
SELECT * FROM 'iot/topic'
```

6. En Definir una o varias acciones, elija Agregar acción.
7. Seleccione Enviar un mensaje a una función de Lambda.
8. Elija Configurar acción.

9. Busque y seleccione su función de Lambda de la lista.
10. Seleccione Agregar acción.
11. Elija Crear regla.

Pruebe su regla de AWS IoT Core en la consola

Si actualmente ningún dispositivo publica telemetría que incluya la ubicación, puede probar su regla y esta solución mediante la consola de AWS IoT Core. La consola tiene un cliente de prueba en el que puede publicar un mensaje de muestra para comprobar los resultados de la solución.

1. Inicia sesión en la AWS IoT Core consola en <https://console.aws.amazon.com/iot/>.
2. En el menú de navegación de la izquierda, expanda Pruebas y elija Cliente de pruebas de MQTT.
3. En Publicar en un tema, defina el Nombre del tema como *iot/topic* (o el nombre del tema que haya configurado en su regla de AWS IoT Core, si es diferente) y proporcione lo siguiente para la Carga de mensajes. Sustituya la marca de tiempo *1604940328* por una marca de tiempo válida de los últimos 30 días (se omiten las marcas de tiempo de más de 30 días).

```
{
  "payload": {
    "deviceid": "thing123",
    "timestamp": 1604940328,
    "location": { "lat": 49.2819, "long": -123.1187 },
    "accuracy": { "Horizontal": 20.5 },
    "positionProperties": { "field1": "value1", "field2": "value2" }
  }
}
```

4. Seleccione Publicar en un tema para enviar el mensaje de prueba.
5. Para validar que Amazon Location Service recibió el mensaje, use el siguiente comando AWS CLI. Si los modificó durante la configuración, sustituya el nombre del rastreador y el identificador del dispositivo por los que utilizó.

```
aws location batch-get-device-position --tracker-name MyTracker --device-ids
thing123
```

Administrar sus recursos de colección de geovallas

Gestione sus colecciones de geovallas mediante la consola de Amazon Location, AWS CLI o las API de Amazon Location.

Enumerar sus recursos de colección de geovallas

Puede ver su lista de colecciones de geovallas mediante la consola de Amazon Location, AWS CLI o las API de Amazon Location:

Console

Para ver una lista de colecciones de geovallas mediante la consola de Amazon Location

1. Abra la consola de Amazon Location en <https://console.aws.amazon.com/location/>.
2. En el panel de navegación izquierdo, elija Colecciones de geovallas.
3. Consulte una lista de sus colecciones de geovallas en Mis colecciones de geovallas.

API

Utilice la operación [ListGeofenceCollections](#) desde las API de Amazon Location Geofences.

El siguiente ejemplo es una solicitud de API para obtener una lista de las colecciones de geovallas de la cuenta de AWS.

```
POST /geofencing/v0/list-collections
```

A continuación se muestra un ejemplo de respuesta para 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

Utilice el comando [list-geofence-collections](#).

El siguiente ejemplo es un AWS CLI para obtener una lista de las colecciones de geovallas de la cuenta de AWS.

```
aws location list-geofence-collections
```

Obtenga los detalles de la colección de geovallas

Puede obtener información sobre cualquier recurso de colección de geovallas de su cuenta de AWS mediante la consola de Amazon Location, AWS CLI o la API de Amazon Location:

Console

Para ver los detalles de una colección de geovallas mediante la consola de Amazon Location

1. Abra la consola de Amazon Location en <https://console.aws.amazon.com/location/>.
2. En el panel de navegación izquierdo, elija Colecciones de geovallas.
3. En Mis colecciones de geovallas, seleccione el enlace con el nombre de la colección de geovallas de destino.

API

Utilice la operación [DescribeGeofenceCollection](#) desde las API de Amazon Location Geofences.

El siguiente ejemplo es una solicitud de API para obtener los detalles de la recopilación de geocercas. *ExampleCollection*

```
GET /geofencing/v0/collections/ExampleCollection
```

A continuación, se muestra un ejemplo de respuesta para `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

Utilice el comando [describe-geofence-collection](#).

El siguiente ejemplo es AWS CLI para obtener los detalles de la colección de geovallas.

ExampleCollection

```
aws location describe-geofence-collection \
  --collection-name "ExampleCollection"
```

Eliminar una colección de geovallas

Puede eliminar una colección de geovallas de su cuenta de AWS mediante la consola de Amazon Location, AWS CLI o las API de Amazon Location.

Console

Para eliminar una colección de geovallas mediante la consola de Amazon Location

Warning

Esta operación elimina el recurso de forma permanente.

1. Abra la consola de Amazon Location en <https://console.aws.amazon.com/location/>.
2. En el panel de navegación izquierdo, elija Colecciones de geovallas.
3. En Mi colección de geovallas, seleccione la colección de geovallas de destino.

4. Seleccione Eliminar colección de geovallas.

API

Use la operación [DeleteGeofenceCollection](#) de las API de Amazon Location.

El siguiente ejemplo es una solicitud de API para eliminar la colección de geovallas.

ExampleCollection

```
DELETE /geofencing/v0/collections/ExampleCollection
```

A continuación, se muestra un ejemplo de respuesta para DeleteGeofenceCollection:

```
HTTP/1.1 200
```

CLI

Utilice el comando [delete-geofence-collection](#).

El siguiente ejemplo es un AWS CLI comando para eliminar la colección de geovallas.

ExampleCollection

```
aws location delete-geofence-collection \  
  --collection-name "ExampleCollection"
```

Enumerar las geovallas almacenadas

Puede enumerar las geovallas almacenadas en una colección de geovallas específica mediante la consola de Amazon Location, AWS CLI o las API de Amazon Location.

Console

Para ver una lista de geovallas mediante la consola de Amazon Location

1. Abra la consola de Amazon Location en <https://console.aws.amazon.com/location/>.
2. En el panel de navegación izquierdo, elija Colecciones de geovallas.
3. En Mi colección de geovallas, seleccione el enlace con el nombre de la colección de geovallas de destino.

4. Consulte las geovallas de la colección de geovallas en Geovallas

API

Utilice la operación [ListGeofences](#) desde las API de Amazon Location Geofences.

El siguiente ejemplo es una solicitud de API para obtener una lista de geocercas almacenadas en la colección de geocercas. *ExampleCollection*

```
POST /geofencing/v0/collections/ExampleCollection/list-geofences
```

A continuación, se muestra un ejemplo de respuesta para 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

Utilice el comando [list-geofences](#).

El siguiente ejemplo sirve AWS CLI para obtener una lista de geocercas almacenadas en la colección de geocercas. *ExampleCollection*

```
aws location list-geofences \
```

```
--collection-name "ExampleCollection"
```

Obtener detalles de geovallas

Puede obtener los detalles de una geovalla específica, como la hora de creación, la hora de actualización, la geometría y el estado, de una colección de geovallas mediante la consola de Amazon Location, AWS CLI o las API de Amazon Location.

Console

Para ver el estado de una geovalla mediante la consola de Amazon Location

1. Abra la consola de Amazon Location en <https://console.aws.amazon.com/location/>.
2. En el panel de navegación izquierdo, elija Colecciones de geovallas.
3. En Mi colección de geovallas, seleccione el enlace con el nombre de la colección de geovallas de destino.
4. En Geovallas, podrá ver el estado de sus geovallas.

API

Utilice la operación [GetGeofence](#) desde las API de Amazon Location Geofences.

El siguiente ejemplo es una solicitud de API para obtener los detalles de una geocerca de una colección de geocercas. *ExampleCollection*

```
GET /geofencing/v0/collections/ExampleCollection/geofences/ExampleGeofence1
```

A continuación, se muestra un ejemplo de respuesta para 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

Utilice el comando [get-geofence](#).

El siguiente ejemplo es AWS CLI para obtener los detalles de la colección de geocercas.

ExampleCollection

```
aws location get-geofence \  
  --collection-name "ExampleCollection" \  
  --geofence-id "ExampleGeofence1"
```

Eliminar geovallas

Puede eliminar las geovallas de una colección de geovallas mediante la consola de Amazon Location, AWS CLI o las API de Amazon Location.

Console

Para eliminar una geovalla mediante la consola de Amazon Location

Warning

Esta operación elimina el recurso de forma permanente.

1. Abra la consola de Amazon Location en <https://console.aws.amazon.com/location/>.
2. En el panel de navegación izquierdo, elija Colecciones de geovallas.
3. En Mi colección de geovallas, seleccione el enlace con el nombre de la colección de geovallas de destino.
4. En Geovallas, seleccione la geovalla de destino.
5. Seleccione Eliminar geovalla.

API

Utilice la operación [BatchDeleteGeofence](#) desde las API de Amazon Location Geofences.

El siguiente ejemplo es una solicitud de API para eliminar geocercas de la colección de geocercas. *ExampleCollection*

```
POST /geofencing/v0/collections/ExampleCollection/delete-geofences
Content-type: application/json

{
  "GeofenceIds": [ "ExampleGeofence11" ]
}
```

A continuación se muestra un ejemplo de una respuesta correcta para [BatchDeleteGeofence](#).

```
HTTP/1.1 200
```

CLI

Utilice el comando [batch-delete-geofence](#).

El siguiente ejemplo es un AWS CLI comando para eliminar las geocercas de la colección de geocercas. *ExampleCollection*

```
aws location batch-delete-geofence \
  --collection-name "ExampleCollection" \
  --geofence-ids "ExampleGeofence11"
```

Administrar los recursos de su rastreador

Puede gestionar sus rastreadores mediante la consola de Amazon Location, las AWS CLI o las API de Amazon Location.

Enumerar sus rastreadores

Puede ver su lista de rastreadores mediante la consola de ubicaciones de Amazon, las AWS CLI o las API de Amazon Location:

Console

Para ver una lista de los rastreadores existentes mediante la consola de Amazon Location

1. Abra la consola de Amazon Location en <https://console.aws.amazon.com/location/>.
2. En la navegación izquierda, elija Rastreadores.
3. Consulte una lista de los recursos de su rastreador desde Mis rastreadores.

API

Use la operación [ListTrackers](#) de las API de Amazon Location Trackers.

El siguiente ejemplo es una solicitud de API para obtener una lista de los rastreadores de su cuenta de AWS.

```
POST /tracking/v0/list-trackers
```

A continuación, se muestra un ejemplo de respuesta para [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

Utilice el comando [list-trackers](#).

El siguiente ejemplo es un AWS CLI para obtener una lista de los rastreadores de su cuenta de AWS.

```
aws location list-trackers
```

Desconectar un rastreador de una colección de geovallas

Puede desconectar un rastreador de una colección de geovallas mediante la consola de Amazon Location, las AWS CLI o las API de Amazon Location:

Console

Para desasociar un rastreador de una colección de geovallas asociada mediante la consola de Amazon Location

1. Abra la consola de Amazon Location en <https://console.aws.amazon.com/location/>.
2. Desde el panel de navegación izquierdo, elija Rastreadores.
3. En Mis rastreadores, seleccione el enlace con el nombre del rastreador objetivo.
4. En Colecciones de geovallas vinculadas, seleccione una colección de geovallas con el estado vinculado.
5. Elija Desvincular.

API

Use la operación [DisassociateTrackerConsumer](#) de las API de Amazon Location Trackers.

El siguiente ejemplo es una solicitud de API para desasociar un rastreador de una colección de geovallas asociada.

```
DELETE /tracking/v0/trackers/ExampleTracker/consumers/arn:aws:geo:us-west-2:123456789012:geofence-collection/ExampleCollection
```

A continuación, se muestra un ejemplo de respuesta para [DisassociateTrackerConsumer](#):

```
HTTP/1.1 200
```

CLI

Utilice el comando [disassociate-tracker-consumer](#).

El siguiente ejemplo es un comando AWS CLI para desasociar un rastreador de una colección de geovallas asociada.

```
aws location disassociate-tracker-consumer \
```

```
--consumer-arn "arn:aws:geo:us-west-2:123456789012:geofence-collection/
ExampleCollection" \
--tracker-name "ExampleTracker"
```

Obtenga información acerca de su rastreador

Puede obtener información sobre cualquier rastreador de su cuenta de AWS desde la consola de Amazon Location, las AWS CLI o las API de Amazon Location.

Console

Para ver los detalles del rastreador mediante la consola de Amazon Location

1. Abra la consola de Amazon Location en <https://console.aws.amazon.com/location/>.
2. En la navegación izquierda, elija Rastreadores.
3. En Mis rastreadores, seleccione el enlace con el nombre del rastreador objetivo.
4. Consulte los detalles del rastreador en Información.

API

Use la operación de [DescribeTracker](#) desde las API de Amazon Location Tracker.

El siguiente ejemplo es una solicitud de API para obtener los detalles del rastreador *ExampleTracker*.

```
GET /tracking/v0 trackers/ExampleTracker
```

A continuación, se muestra un ejemplo de respuesta para [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

Utilice el comando [describe-tracker](#).

El siguiente ejemplo es un AWS CLI comando para obtener los detalles del rastreador *ExampleTracker*.

```
aws location describe-tracker \  
  --tracker-name "ExampleTracker"
```

Eliminar un rastreador

Puede eliminar un rastreador de su cuenta AWS desde la consola de Amazon Location, las AWS CLI o las API de Amazon Location:

Console

Para eliminar un recurso de mapa existente mediante la consola de Amazon Location

Warning

Esta operación elimina el recurso de forma permanente. Si el recurso del rastreador está en uso, es posible que se produzca un error. Asegúrese de que el recurso de destino no sea una dependencia de sus aplicaciones.

1. Abra la consola de Amazon Location en <https://console.aws.amazon.com/location/>.
2. Desde el panel de navegación izquierdo, elija Rastreadores.
3. En Mis rastreadores, seleccione el rastreador de destino.
4. Seleccione Eliminar rastreador.

API

Use la operación de [DeleteTracker](#) desde las API de Amazon Location Tracker.

El siguiente ejemplo es una solicitud de API para eliminar el rastreador *ExampleTracker*.

```
DELETE /tracking/v0/trackers/ExampleTracker
```

A continuación, se muestra un ejemplo de respuesta para [DeleteTracker](#):

```
HTTP/1.1 200
```

CLI

Utilice el comando [delete-tracker](#).

El siguiente ejemplo es un AWS CLI comando para eliminar el rastreador *ExampleTracker*.

```
aws location delete-tracker \  
  --tracker-name "ExampleTracker"
```

Ejemplo de aplicación móvil de geofencing and Tracking

En este tema se incluyen tutoriales diseñados para demostrar las características clave del uso de las geovallas y los rastreadores de Amazon Location en una aplicación móvil. Las aplicaciones demuestran cómo interactúan un rastreador y una geocerca mediante una combinación de funciones de Lambda y AWS IoT Amazon Location. Hay dos tutoriales disponibles.

- [Muestra una aplicación de rastreo y geofencing para Android](#). Puedes clonar los archivos del proyecto desde GitHub: <https://github.com/aws-geospatial/amazon-location-samples-android/tree/main/tracking-with-geofence-notifications>
- [Ejemplo de aplicación de rastreo y geofencing para iOS](#), y puede clonar los archivos del proyecto desde GitHub: <https://github.com/aws-geospatial/amazon-location-samples-ios/tree/main/tracking-with-geofence-notifications>

Aplicación de seguimiento de muestras y geovallas para Android

Este tema cubre el tutorial de Android diseñado para demostrar las características clave del uso de las geovallas y rastreadores de Amazon Location en una aplicación móvil. Las aplicaciones muestran cómo interactúan un rastreador y un geofence mediante una combinación de funciones de Lambda y AWS IoT Amazon Location.

Temas

- [Crea recursos de Amazon Location para tu aplicación](#)
- [Crea una colección de Geofence](#)
- [Vincular un rastreador a una colección de geovallas](#)
- [Uso de AWS Lambda con MQTT](#)
- [Configure el código de la aplicación de muestra](#)
- [Uso de la aplicación de muestra](#)

Crea recursos de Amazon Location para tu aplicación

Para empezar, tendrás que crear los recursos de Amazon Location necesarios. Estos recursos serán esenciales para el funcionamiento de la aplicación y para ejecutar los fragmentos de código proporcionados.

Note

Si no has creado una AWS cuenta, sigue las instrucciones de la guía del usuario de [administración de AWS cuentas](#).

Para empezar, tendrá que crear un ID de grupo de identidades de Amazon Cognito, mediante el siguiente procedimiento:

1. Abra la [consola de Amazon Cognito](#) y seleccione Grupos de identidades en el menú de la izquierda y, a continuación, seleccione Crear grupo de identidades.
2. Asegúrese de que Guest Access esté marcada y pulse Siguiente para continuar.
3. A continuación, cree una nueva función de IAM o utilice una función de IAM existente.
4. Introduce un nombre de grupo de identidades y asegúrate de que Identity Pool tiene acceso a (geo) los recursos de Amazon Location para el mapa y el rastreador que vas a crear en el siguiente procedimiento.

A continuación, debe crear y aplicar estilo a un mapa en la consola de AWS Amazon Location, siguiendo el siguiente procedimiento:

1. Ve a la [sección Mapas](#) de la consola Amazon Location y selecciona Create Map.

2. Asigna un nombre y una descripción al nuevo mapa. Registre el nombre que le asigne, tal y como se utilizará más adelante en el tutorial.
3. Al elegir un estilo de mapa, tenga en cuenta el proveedor de datos del mapa. Consulte la sección 82 de las [condiciones del AWS servicio](#) para obtener más información.
4. Acepta los [términos y condiciones de Amazon Location y](#), a continuación, selecciona Crear mapa para finalizar el proceso de creación del mapa.

A continuación, debe crear un rastreador en la consola de Amazon Location, mediante el siguiente procedimiento:

1. Abre la [sección Mapas](#) en la consola de Amazon Location.
2. Seleccione Crear rastreador.
3. Rellena los campos obligatorios. Anota el nombre del rastreador, ya que aparecerá como referencia a él en este tutorial.
4. En el campo de filtrado de posición, elige la opción que mejor se adapte a la forma en que piensas utilizar el recurso de tu rastreador. Si no configuras el filtrado de posición, el ajuste predeterminado es `TimeBased`. Para obtener más información, consulta [Rastreadores y PositionFiltering](#) en la referencia de la API de ubicación de Amazon.
5. Seleccione Crear rastreador para terminar de crear el rastreador.

Crea una colección de Geofence

Ahora crearás una colección de geofence. Puede usar la consola, la API o la CLI. Los siguientes procedimientos le guiarán por cada opción.

- Crea una colección de geovallas con la consola de Amazon Location:
 1. Abre la sección [Colecciones de Geofence](#) de la consola de Amazon Location.
 2. Seleccione Crear colección de geovallas.
 3. Escriba un nombre y una descripción para la colección.
 4. Según la EventBridge regla con Amazon CloudWatch como objetivo, puede crear una EventBridge regla opcional para empezar a reaccionar ante los eventos de geofence. Esto permite a Amazon Location publicar eventos en Amazon CloudWatch Logs.
 5. Pulse el botón Crear colección geofence para terminar de crear la colección.
- Crea una colección de geovallas mediante la API de Amazon Location:

Usa la [CreateGeofenceCollection](#) operación desde las API de Amazon Location Geofences. En el siguiente ejemplo, se utiliza una solicitud de API para crear una colección de geocercas llamada. *GEOCOLLECTION_NAME*

```
POST /geofencing/v0/collections
Content-type: application/json

{
  "CollectionName": "GEOCOLLECTION_NAME",
  "Description": "Geofence collection 1 for shopping center",
  "Tags": {
    "Tag1" : "Value1"
  }
}
```

- Cree una colección de geocercas mediante los comandos de la AWS CLI:

Utilice el comando `create-geofence-collection`. El siguiente ejemplo utiliza una AWS CLI para crear una colección de geocercas llamada. *GEOCOLLECTION_NAME* Para obtener más información sobre el uso de la AWS CLI, consulte la [documentación de la interfaz de línea de AWS comandos](#).

```
aws location \
  create-geofence-collection \
  --collection-name "ExampleGeofenceCollection" \
  --description "Shopping center geofence collection" \
  --tags Tag1=Value1
```

Vincular un rastreador a una colección de geovallas

Para vincular un rastreador a una colección de geovallas, puede usar la consola, la API o la CLI. Los siguientes procedimientos le guiarán por cada opción.

Vincule un recurso de rastreo a una colección de geovallas mediante la consola de Amazon Location Service:

1. Abre la consola de Amazon Location.
2. En el panel de navegación izquierdo, elija Rastreadores.
3. En Device Trackers, selecciona el enlace con el nombre del rastreador objetivo.

4. En Colecciones de geovallas vinculadas, elija Vincular colección de geovallas.
5. En la ventana Colección de geovallas vinculadas, seleccione una colección de geovallas en el menú desplegable.
6. Elija Vincular.
7. Tras vincular el recurso de rastreador, se le asignará el estado Activo.

Vincula un recurso de rastreo a una colección de geovallas mediante las API de ubicación de Amazon:

Use la operación `AssociateTrackerConsumer` de las API de Amazon Location Trackers. En el siguiente ejemplo, se utiliza una solicitud de API que asocia y a una colección de geovallas mediante su nombre de recurso de Amazon (ARN). `ExampleTracker`

```
POST /tracking/v0/trackers/ExampleTracker/consumers
Content-type: application/json
{
  "ConsumerArn": "arn:aws:geo:us-west-2:123456789012:geofence-
collection/GOECOLLECTION_NAME"
}
```

Vincule un recurso de rastreo a una colección de geovallas mediante los comandos de la AWS CLI:

Utilice el comando `associate-tracker-consumer` . El siguiente ejemplo utiliza una AWS CLI para crear una colección de geocercas llamada. *GOECOLLECTION_NAME*

```
aws location \
associate-tracker-consumer \
  --consumer-arn "arn:aws:geo:us-west-2:123456789012:geofence-
collection/GOECOLLECTION_NAME" \
  --tracker-name "ExampleTracker"
```

Uso de AWS Lambda con MQTT

Para crear una conexión entre una ubicación de Amazon AWS IoT y Amazon, necesitas una función Lambda para procesar los mensajes reenviados por EventBridge CloudWatch eventos. Esta función

extraerá todos los datos posicionales, los formateará para Amazon Location y los enviará a través de la API Amazon Location Tracker.

El siguiente procedimiento muestra cómo crear esta función a través de la consola Lambda:

1. Abra la [consola de](#).
2. En el panel de navegación izquierdo, elija Funciones.
3. A continuación, elija Crear función y asegúrese de que esté seleccionada la opción Autor desde cero.
4. proporcione un nombre de función y, para la opción Runtime, elija Node.js 16.x.
5. Elija Crear función.
6. Abra la pestaña Código para acceder al editor.
7. Sobrescribe el código marcador de posición del `index.js` archivo con lo siguiente:

```
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. Pulse el botón Desplegar para guardar la función actualizada.
9. A continuación, abra la pestaña Configuración.
10. En la sección Activadores, pulse el botón Añadir disparador.
11. Seleccione EventBridge (CloudWatch Eventos) en el campo Fuente.
12. Seleccione la opción Reglas existentes.
13. Introduzca el nombre de la regla, por ejemplo AmazonLocationMonitor-GEOFENCECOLLECTION_NAME.
14. Pulse el botón Añadir.
15. Esto también adjuntará las declaraciones de política basadas en los recursos en la pestaña de permisos

Ahora configurará el cliente de pruebas de MQTT mediante el AWS IoT siguiente procedimiento:

1. Abra el <https://console.aws.amazon.com/iot/>.

2. En el panel de navegación izquierdo, seleccione el cliente de prueba MQTT.
3. Verá una sección titulada Cliente de prueba MQTT en la que puede configurar su conexión MQTT.
4. Después de configurar los ajustes necesarios, haga clic en el botón Connect para establecer una conexión con el corredor MQTT utilizando los parámetros proporcionados.
5. Registre el punto final, tal como se utiliza más adelante en el tutorial.

Una vez conectado al cliente de prueba, puede suscribirse a los temas de MQTT o publicar mensajes sobre los temas utilizando los campos de entrada correspondientes que se proporcionan en la interfaz del cliente de prueba de MQTT. A continuación, creará una AWS IoT política.

6. En el menú de la izquierda, en Administrar, expanda la opción Seguridad y haga clic en Políticas.
7. Haga clic en el botón Crear política.
8. Escriba un nombre para la política.
9. En el documento de política, seleccione la pestaña JSON.
10. Copia y pega la política que se muestra a continuación, pero asegúrate de actualizar todos los elementos con tu **REGION** y **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. Selecciona el botón Crear para terminar.

Tras completar el procedimiento anterior, ahora actualizará los permisos del rol de invitado de la siguiente manera:

1. Navegue hasta Amazon Cognito y abra su grupo de identidades. A continuación, vaya al acceso de usuario y seleccione el rol de invitado.
2. Haga clic en las políticas de permisos para habilitar la edición.

```

{
  '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. Con los cambios de política anteriores, todos los AWS recursos necesarios ahora están configurados adecuadamente para la aplicación.

Configure el código de la aplicación de muestra

1. Clona este repositorio: <https://github.com/aws-geospatial/amazon-location-samples-android/tree/main/tracking-with-geofence-notifications> en tu máquina local.
2. Abre el AmazonSampleSDKApp proyecto en Android Studio.
3. Crea y ejecuta la aplicación en tu emulador o dispositivo Android.

Uso de la aplicación de muestra

Para usar el ejemplo, siga estos procedimientos:

- Cree un **custom.properties**:

Para configurar el `custom.properties` archivo, sigue estos pasos:

1. Abre tu editor de texto o IDE preferido.
2. Cree un archivo nuevo.
3. Guarde el archivo con el nombre `custom.properties`.
4. Actualízalo `custom.properties` con el siguiente ejemplo de código y reemplaza `MQTT_END_POINT` `POLICY_NAME` `GEOFENCE_COLLECTION_NAME`, y por los `TOPIC_TRACKER` nombres de tus recursos:

```
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. Limpie y reconstruya el proyecto. Después de esto, puede ejecutar el proyecto.

- Iniciar sesión:

Para iniciar sesión en la aplicación, sigue los pasos que se indican a continuación:

1. Presiona el botón de inicio de sesión.
2. Proporcione un identificador de grupo de identidades, un nombre de rastreador y un nombre de mapa.
3. Vuelva a pulsar Iniciar sesión para terminar.

- Administrar filtros:

Abra la pantalla de configuración y realice lo siguiente:

1. Activa o desactiva los filtros mediante la interfaz de usuario del conmutador.
2. Actualiza los filtros de tiempo y distancia cuando sea necesario.

- Operaciones de seguimiento:

Abra la pantalla de rastreo y realice lo siguiente:

- Puede iniciar y detener el rastreo en primer plano, en segundo plano o en modo de ahorro de batería pulsando los botones correspondientes.

Aplicación de seguimiento de muestras y geofencing para iOS

Este tema cubre el tutorial de iOS diseñado para demostrar las características clave del uso de las geovallas y rastreadores de Amazon Location en una aplicación móvil. Las aplicaciones muestran cómo interactúan un rastreador y un geofence mediante una combinación de funciones de Lambda y AWS IoT Amazon Location.

Temas

- [Crea recursos de Amazon Location para tu aplicación](#)
- [Crea una colección de Geofence](#)
- [Vincular un rastreador a una colección de geovallas](#)
- [Uso de AWS Lambda con MQTT](#)
- [Configurar un ejemplo de código de aplicación](#)
- [Uso de la aplicación de ejemplo](#)

Crea recursos de Amazon Location para tu aplicación

Para empezar, tendrás que crear los recursos de Amazon Location necesarios. Estos recursos serán esenciales para el funcionamiento de la aplicación y para ejecutar los fragmentos de código proporcionados.

Note

Si no has creado una AWS cuenta, sigue las instrucciones de la guía del usuario de [administración de AWS cuentas](#).

Para empezar, tendrá que crear un ID de grupo de identidades de Amazon Cognito, mediante el siguiente procedimiento:

1. Abra la [consola de Amazon Cognito](#) y seleccione Grupos de identidades en el menú de la izquierda y, a continuación, seleccione Crear grupo de identidades.
2. Asegúrese de que Guest Access esté marcada y pulse Siguiente para continuar.
3. A continuación, cree una nueva función de IAM o utilice una función de IAM existente.

4. Introduzca un nombre de grupo de identidades y asegúrese de que Identity Pool tiene acceso a (geo) los recursos de Amazon Location para el mapa y el rastreador que va a crear en el siguiente procedimiento.

A continuación, debe crear y aplicar estilo a un mapa en la consola de AWS Amazon Location, siguiendo el siguiente procedimiento:

1. Ve a la [sección Mapas](#) de la consola Amazon Location y selecciona Create Map.
2. Asigna un nombre y una descripción al nuevo mapa. Registre el nombre que le asigne, tal como se utilizará más adelante en el tutorial.
3. Al elegir un estilo de mapa, tenga en cuenta el proveedor de datos del mapa. Consulte la sección 82 de las [condiciones del AWS servicio](#) para obtener más información.
4. Acepta los [términos y condiciones de Amazon Location y](#), a continuación, selecciona Crear mapa para finalizar el proceso de creación del mapa.

A continuación, debe crear un rastreador en la consola de Amazon Location, mediante el siguiente procedimiento:

1. Abre la [sección Mapas](#) en la consola de Amazon Location.
2. Seleccione Crear rastreador.
3. Rellena los campos obligatorios. Anota el nombre del rastreador, ya que aparecerá como referencia a él en este tutorial.
4. En el campo de filtrado de posición, elige la opción que mejor se adapte a la forma en que piensas utilizar el recurso de tu rastreador. Si no configuras el filtrado de posición, el ajuste predeterminado es TimeBased. Para obtener más información, consulta [Empezar a rastrear y PositionFiltering](#) en la referencia de la API de ubicación de Amazon.
5. Seleccione Crear rastreador para terminar de crear el rastreador.

Crea una colección de Geofence

Ahora crearás una colección de geofence. Puede usar la consola, la API o la CLI. Los siguientes procedimientos le guiarán por cada opción.

- Crea una colección de geovallas con la consola de Amazon Location:
 1. Abre la sección [Colecciones de Geofence](#) de la consola de Amazon Location.

2. Seleccione Crear colección de geovallas.
 3. Escriba un nombre y una descripción para la colección.
 4. Según la EventBridge regla con Amazon CloudWatch como objetivo, puede crear una EventBridge regla opcional para empezar a reaccionar ante los eventos de geofence. Esto permite a Amazon Location publicar eventos en Amazon CloudWatch Logs.
 5. Pulse el botón Crear colección geofence para terminar de crear la colección.
- Crea una colección de geovallas mediante la API de Amazon Location:

Usa la [CreateGeofenceCollection](#) operación desde las API de Amazon Location Geofences. En el siguiente ejemplo, se utiliza una solicitud de API para crear una colección de geocercas llamada.

GEOCOLLECTION_NAME

```
POST /geofencing/v0/collections
Content-type: application/json

{
  "CollectionName": "GEOCOLLECTION_NAME",
  "Description": "Geofence collection 1 for shopping center",
  "Tags": {
    "Tag1" : "Value1"
  }
}
```

- Cree una colección de geocercas mediante los comandos de la AWS CLI:

Utilice el comando `create-geofence-collection`. El siguiente ejemplo utiliza una AWS CLI para crear una colección de geocercas llamada *GEOCOLLECTION_NAME*. Para obtener más información sobre el uso de la AWS CLI, consulte la [documentación de la interfaz de línea de AWS comandos](#).

```
aws location \
  create-geofence-collection \
  --collection-name "ExampleGeofenceCollection" \
  --description "Shopping center geofence collection" \
  --tags Tag1=Value1
```

Vincular un rastreador a una colección de geovallas

Para vincular un rastreador a una colección de geovallas, puede usar la consola, la API o la CLI. Los siguientes procedimientos le guiarán por cada opción.

Vincule un recurso de rastreo a una colección de geovallas mediante la consola de Amazon Location Service:

1. Abra la consola de Amazon Location.
2. En el panel de navegación izquierdo, elija Rastreadores.
3. En Device Trackers, selecciona el enlace con el nombre del rastreador objetivo.
4. En Colecciones de geovallas vinculadas, elija Vincular colección de geovallas.
5. En la ventana Colección de geovallas vinculadas, seleccione una colección de geovallas en el menú desplegable.
6. Elija Vincular.
7. Tras vincular el recurso de rastreador, se le asignará el estado Activo.

Vincula un recurso de rastreo a una colección de geovallas mediante las API de ubicación de Amazon:

Use la operación `AssociateTrackerConsumer` de las API de Amazon Location Trackers. En el siguiente ejemplo, se utiliza una solicitud de API que se `ExampleTracker` asocia a una colección de geovallas mediante su nombre de recurso de Amazon (ARN).

```
POST /tracking/v0/trackers/ExampleTracker/consumers
Content-type: application/json
{
  "ConsumerArn": "arn:aws:geo:us-west-2:123456789012:geofence-
collection/GEOCOLLECTION_NAME"
}
```

Vincule un recurso de seguimiento a una colección de geovallas mediante los comandos de la AWS CLI:

Utilice el comando `associate-tracker-consumer` . En el siguiente ejemplo, se utiliza una CLI de AWS para crear una colección de geocercas llamada. *GEOCOLLECTION_NAME*

```
aws location \
  associate-tracker-consumer \
```

```
--consumer-arn "arn:aws:geo:us-west-2:123456789012:geofence-  
collection/GEOCOLLECTION_NAME" \  
--tracker-name "ExampleTracker"
```

Uso de AWS Lambda con MQTT

Para crear una conexión entre AWS IoT una ubicación de Amazon, necesitas una función Lambda para procesar los mensajes reenviados por EventBridge CloudWatch eventos. Esta función extraerá todos los datos posicionales, los formateará para Amazon Location y los enviará a través de la API Amazon Location Tracker.

El siguiente procedimiento muestra cómo crear esta función a través de la consola Lambda:

1. Abra la [consola de](#) .
2. En el panel de navegación izquierdo, elija Funciones.
3. A continuación, elija Crear función y asegúrese de que esté seleccionada la opción Autor desde cero.
4. proporcione un nombre de función y, para la opción Runtime, elija Node.js 16.x.
5. Elija Crear función.
6. Abra la pestaña Código para acceder al editor.
7. Sobrescribe el código marcador de posición del `index.js` archivo con lo siguiente:

```
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. Pulse el botón Implementar para guardar la función actualizada.
9. A continuación, abra la pestaña Configuración.
10. En la sección Activadores, pulse el botón Añadir disparador.
11. Seleccione EventBridge (CloudWatch Eventos) en el campo Fuente.
12. Seleccione la opción Reglas existentes.
13. Introduzca el nombre de la regla, por ejemplo AmazonLocationMonitor-GEOFENCECOLLECTION_NAME.

14. Pulse el botón Añadir.
15. Esto también adjuntará las declaraciones de política basadas en los recursos en la pestaña de permisos

Ahora va a configurar el cliente de pruebas de AWS IoT MQTT; utilice el siguiente procedimiento:

1. Abra el <https://console.aws.amazon.com/iot/>.
2. En el panel de navegación izquierdo, seleccione el cliente de prueba MQTT.
3. Verá una sección titulada Cliente de prueba MQTT en la que puede configurar su conexión MQTT.
4. Después de configurar los ajustes necesarios, haga clic en el botón Connect para establecer una conexión con el corredor MQTT utilizando los parámetros proporcionados.
5. Registre el punto final, tal como se utiliza más adelante en el tutorial.

Una vez conectado al cliente de prueba, puede suscribirse a los temas de MQTT o publicar mensajes sobre los temas utilizando los campos de entrada correspondientes que se proporcionan en la interfaz del cliente de prueba de MQTT. A continuación, creará una AWS IoT política.

6. En el menú de la izquierda, en Administrar, expanda la opción Seguridad y haga clic en Políticas.
7. Haga clic en el botón Crear política.
8. Escriba un nombre para la política.
9. En el documento de política, seleccione la pestaña JSON.
10. Copia y pega la política que se muestra a continuación, pero asegúrate de actualizar todos los elementos con tu *REGION* mano *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. Selecciona el botón Crear para terminar.

Configurar un ejemplo de código de aplicación

Para configurar el código de ejemplo, debes tener instaladas las siguientes herramientas:

- Git
- XCode 15.3 o posterior
- Simulador de iOS 16 o posterior

Utilice este procedimiento para configurar el código de la aplicación de ejemplo:

1. Clona el repositorio de git desde esta URL: <https://github.com/aws-geospatial/amazon-location-samples-ios/tree/main/tracking-with-geofence-notifications>.
2. Abra el archivo de proyecto `AWSLocationSampleApp.xcodeproj`.
3. Espera a que finalice el proceso de resolución del paquete.
4. En el menú de navegación del proyecto, cambie el nombre `ConfigTemplate.xcconfig` a los siguientes valores `Config.xcconfig` y rellénelos:

```

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`

```

Uso de la aplicación de ejemplo

Tras configurar el código de muestra, ahora puedes ejecutar la aplicación en un simulador de iOS o en un dispositivo físico.

1. Crea y ejecuta la aplicación.
2. La aplicación te pedirá permisos de ubicación y notificación. Tienes que permitirlos.
3. Pulse el botón de configuración de Cognito.
4. Guarde la configuración.
5. Ahora puede ver las opciones de filtro para ver el tiempo, la distancia y la precisión. Úselos según sus necesidades.
6. Vaya a la pestaña de seguimiento de la aplicación y verá el mapa y el botón Iniciar seguimiento.
7. Si ha instalado la aplicación en un simulador, es posible que desee simular los cambios de ubicación. Esto se puede hacer en Características, en la opción del menú Ubicación. Por ejemplo, selecciona Características, Ubicación y, por último, Freeway Drive.
8. Pulsa el botón Iniciar seguimiento. Deberías ver los puntos de rastreo en el mapa.
9. La aplicación también rastrea las ubicaciones en segundo plano. Por lo tanto, cuando muevas la aplicación en segundo plano, te pedirá permiso para continuar rastreando en segundo plano.
10. Puede detener el seguimiento pulsando el botón Detener seguimiento.

Etiquetado de los recursos de Amazon Location Service

Utilice el etiquetado de recursos en Amazon Location para crear etiquetas con las que clasificar los recursos según su finalidad, propietario, entorno o criterio. Etiquetar sus recursos le ayuda a administrar, identificar, organizar, buscar y filtrar sus recursos.

Por ejemplo, con AWS Resource Groups, puede crear grupos de recursos de AWS basados en una o más etiquetas o partes de etiquetas. También puede crear grupos en función de su aparición en una pila de AWS CloudFormation. Con los grupos de recursos y el Tag Editor, puede consolidar y consultar datos de aplicaciones que consten de varios servicios, recursos y regiones en un solo lugar. Para obtener más información sobre [las estrategias de etiquetado habituales](#), consulte la referencia general de AWS.

Cada etiqueta consta de una clave y un valor definidos:

- **Clave de etiqueta:** etiqueta general que clasifica los valores de las etiquetas. Por ejemplo, `CostCenter`.
- **Valor de etiqueta:** descripción opcional de la categoría de clave de la etiqueta. Por ejemplo, `MobileAssetTrackingResourcesProd`.

Este tema puede ser de ayuda para familiarizarse con el etiquetado al revisar las restricciones de etiquetado. También le muestra cómo crear etiquetas y utilizarlas para realizar un seguimiento del costo de AWS de cada etiqueta activa mediante informes de asignación de costos.

Temas

- [Restricciones de etiquetado](#)
- [Conceder permiso para etiquetar recursos](#)
- [Agregue una etiqueta a un recurso de Amazon Location Service](#)
- [Realice un seguimiento del costo de los recursos por etiqueta](#)
- [Controlar el acceso a los recursos de Amazon Location Service mediante etiquetas](#)
- [Más información](#)

Restricciones de etiquetado

Se aplican las siguientes restricciones básicas a las etiquetas:

- Máximo de etiquetas por recurso: 50
- Para cada recurso, cada clave de etiqueta debe ser única y solo puede tener un valor.

Note

Si añade una nueva etiqueta con la misma clave que una etiqueta existente, la nueva etiqueta sobrescribirá a la antigua.

- Longitud máxima de la clave: 128 caracteres Unicode en UTF-8
- Longitud máxima del valor: 256 caracteres Unicode en UTF-8
- Los caracteres permitidos en los servicios son: letras, números y espacios representables en UTF-8, además de los siguientes caracteres: `+ - = . _ : / @`.
- Las claves y los valores de las etiquetas distinguen entre mayúsculas y minúsculas.

- El prefijo `aws:` se reserva para uso de AWS. Si la etiqueta tiene una clave de etiqueta con este prefijo, no puede editar ni eliminar la clave o el valor de la etiqueta. Las etiquetas que tengan el prefijo `aws:` no cuentan para el límite de etiquetas por recurso.

Conceder permiso para etiquetar recursos

Puede usar las políticas de IAM para controlar el acceso a los recursos de su Amazon Location y conceder permiso para etiquetar un recurso al crearlo. Además de conceder permisos para crear recursos, la política puede incluir permisos de `Action` para permitir las operaciones de etiquetado:

- `geo:TagResource`: Permite a un usuario asignar una o varias etiquetas a un recurso de Amazon Location específico.
- `geo:UntagResource`: Permite a un usuario asignar una o varias etiquetas a un recurso de Amazon Location específico.
- `geo:ListTagsForResource`: Permite a un usuario enumerar todas las etiquetas asignadas a un recurso Amazon location.

El siguiente es un ejemplo de política que permite a un usuario crear una colección de geovallas y etiquetar los recursos:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowTaggingForGeofenceCollectionOnCreation",
      "Effect": "Allow",
      "Action": [
        "geo:CreateGeofenceCollection",
        "geo:TagResource"
      ],
      "Resource": "arn:aws:geo:region:accountID:geofence-collection/*"
    }
  ]
}
```

Agregue una etiqueta a un recurso de Amazon Location Service

Puede agregar etiquetas al crear sus recursos mediante la consola de Amazon Location, la AWS CLI o las API de Amazon Location:

- [Crear un recurso de mapa](#)
- [Crear un recurso de índice de ubicación](#)
- [Crear un recurso de la calculadora de rutas](#)
- [Crear una colección de geovallas](#)
- [Crear un recurso de rastreo](#)

Para etiquetar recursos existentes, editar o eliminar etiquetas

1. Abra la consola de Amazon Location en <https://console.aws.amazon.com/location/>.
2. En el panel de navegación izquierdo, elija el recurso que desea etiquetar. Por ejemplo, Mapas.
3. Elija un recurso de la lista.
4. Seleccione Administrar etiquetas para agregar, editar o eliminar las etiquetas.

Realice un seguimiento del costo de los recursos por etiqueta

Puede usar etiquetas para la asignación de costos para realizar un seguimiento detallado de sus costos de AWS. Después de activar las etiquetas de asignación de costos, AWS las utiliza para organizar la facturación de los recursos en el informe de asignación de costos. Esto le ayudará a categorizar y realizar un seguimiento de sus costos de uso.

Hay dos tipos de etiquetas de asignación de costos que puede activar:

- [Generadas en AWS](#): estas etiquetas las genera AWS. Las etiquetas de AWS usan el prefijo de `aws:`, por ejemplo, `aws:createdBy`.
- [Definidas por el usuario](#): son etiquetas personalizadas que usted crea. Las etiquetas definidas por el usuario utilizan el prefijo `user:`, por ejemplo, `user:CostCenter`.

Debe activar cada tipo de etiqueta de forma individual. Una vez activadas las etiquetas, puede [activar AWS Cost Explorer](#) o ver su informe mensual de asignación de costos.

AWS-generated tags

Para activar las etiquetas generadas por AWS

1. Abra la consola Billing and Cost Management en <https://console.aws.amazon.com/billing/>.
2. En el panel de navegación izquierdo, seleccione Etiquetas de asignación de costos.
3. En la pestaña Etiquetas de asignación de costos generadas en AWS, seleccione las claves de etiqueta que desea activar.
4. Seleccione Activar.

User-defined tags

Para activar las etiquetas definidas por el usuario

1. Abra la consola Billing and Cost Management en <https://console.aws.amazon.com/billing/>.
2. En el panel de navegación izquierdo, seleccione Etiquetas de asignación de costos.
3. En la pestaña Etiquetas de asignación de costos definidas por el usuario, seleccione las claves de etiquetas que desee activar.
4. Seleccione Activar.

Tras activar las etiquetas, AWS genera un [Informe mensual de asignación de costos](#) para el uso y el costo de los recursos. Este informe de asignación de costos incluye todos los costos de AWS de cada período de facturación, incluidos los recursos etiquetados y no etiquetados. Para obtener más información, consulte [Uso de etiquetas de asignación de costos](#) en la Guía del usuario de AWS Billing and Cost Management.

Controlar el acceso a los recursos de Amazon Location Service mediante etiquetas

AWS Identity and Access Management Las políticas (IAM) admiten condiciones basadas en etiquetas, lo que le permite administrar la autorización de los recursos en función de etiquetas, claves y valores específicos. Por ejemplo, una política de roles de IAM puede incluir condiciones para limitar el acceso a entornos específicos (como desarrollo, prueba o producción) en función de sus etiquetas.

Para obtener más información, consulte el tema sobre el [control del acceso a los recursos en función de las etiquetas](#).

Más información

Para obtener más información acerca de:

- Prácticas recomendadas para el etiquetado, consulte [Etiquetado de recursos de AWS](#) en la Referencia general de AWS.
- Uso de etiquetas para controlar el acceso a los recursos de AWS, consulte [Control de acceso a los recursos de AWS con etiquetas](#) en la AWS Identity and Access Management Guía del usuario de .

Concesión de acceso a Amazon Location Service

Para utilizar Amazon Location Service, el usuario debe tener acceso a los recursos y las API que componen Amazon Location. Hay tres estrategias que puede utilizar para conceder acceso a sus recursos.

- Utilizar la IAM: para conceder el acceso a los usuarios autenticados con AWS IAM Identity Center o AWS Identity and Access Management (IAM), cree una política de IAM que permita el acceso a los recursos que desee. Para obtener más información sobre IAM y Amazon Location, consulte [Identity and Access Management para Amazon Location Service](#).
- Utilizar las claves de API: para conceder acceso a usuarios no autenticados, puede crear claves de API que den acceso de solo lectura a sus recursos de Amazon Location Service. Esto resulta útil en el caso de que no desee autenticar a todos los usuarios. Por ejemplo, una aplicación web. Para obtener más información acerca de las claves API, consulte [Permitir el acceso de invitados no autenticados a su aplicación mediante claves de API](#).
- Utilizar Amazon Cognito: puede utilizar Amazon Cognito para conceder acceso anónimo como alternativa a las claves de API. Amazon Cognito le permite crear una autorización más completa con una política para definir lo que pueden hacer los usuarios no autenticados. Para obtener más información sobre cómo usar Amazon Cognito, consulte [Cómo permitir el acceso a sus aplicaciones como invitado no autenticado con Amazon Cognito](#).

Note

También puede usar Amazon Cognito para usar su propio proceso de autenticación o para combinar varios métodos de autenticación mediante Amazon Cognito Federated Identities. Para obtener más información, consulte [Introducción a las identidades federadas](#) en la Guía para desarrolladores de Amazon Cognito.

Temas

- [Permitir el acceso de invitados no autenticados a su aplicación mediante claves de API](#)
- [Cómo permitir el acceso a sus aplicaciones como invitado no autenticado con Amazon Cognito](#)

Permitir el acceso de invitados no autenticados a su aplicación mediante claves de API

Cuando llama a las API de Amazon Location Service en sus aplicaciones, normalmente realiza esta llamada como un usuario autenticado que está autorizado a realizar las llamadas a las API. No obstante, hay algunos casos en los que no desea autenticar a todos los usuarios de su aplicación. Por ejemplo, es posible que desee que cualquier persona que utilice el sitio web tenga acceso a una aplicación web que muestre la ubicación de su empresa, tanto si ha iniciado sesión como si no. En este caso, una alternativa es utilizar las claves de la API para realizar las llamadas a la API.

Las claves de API son un valor clave que se asocia a recursos específicos de Amazon Location Service y a acciones específicas que puedes realizar en esos recursos. Cuenta de AWS puede usar una clave de API en su aplicación para realizar llamadas no autenticadas a las API de Amazon Location para esos recursos. Por ejemplo, si asocia una clave de API al recurso de mapa Mi mapa y a las acciones GetMap*, la aplicación que utilice esa clave de API podrá ver los mapas creados con ese recurso y se cobrará a su cuenta como cualquier otro uso de su cuenta. Esa misma clave de API no otorgaría permisos para cambiar o actualizar el recurso del mapa, solo se permite usar el recurso.

Note

Las claves de API solo están disponibles para su uso con los recursos de mapa, lugares y rutas, y no se pueden modificar ni crear esos recursos. Si su aplicación necesita acceso a otros recursos o acciones para los usuarios no autenticados, puede usar Amazon Cognito para proporcionar acceso junto con las claves de API o en lugar de ellas. Para obtener más información, consulte [Cómo permitir el acceso a sus aplicaciones como invitado no autenticado con Amazon Cognito](#).

Las claves de API incluyen un valor de texto simple que da acceso a uno o más recursos de su cuenta Cuenta de AWS. Si alguien copia su clave de API, podrá acceder a esos mismos recursos. Para evitarlo, puede especificar los dominios en los que se puede usar la clave de API al crear la clave. Estos dominios se denominan referentes. Si es necesario, también puede crear claves de API a corto plazo estableciendo los tiempos de caducidad de sus claves de API.

Temas

- [Claves de API comparadas con Amazon Cognito](#)
- [Crear claves de API](#)
- [Usar una clave de API para llamar a una API de Amazon Location](#)
- [Usar una clave de API para renderizar un mapa](#)
- [Gestionar la vida útil de claves de API](#)

Claves de API comparadas con Amazon Cognito

Las claves de API y Amazon Cognito se utilizan de forma similar en situaciones similares, así que ¿por qué utilizaría una en lugar de la otra? En la siguiente lista, se enumeran algunas de las diferencias entre ambas.

- Las claves de API solo están disponibles para los recursos de mapas, lugares y rutas, y solo para determinadas acciones. Amazon Cognito se puede utilizar para autenticar el acceso a la mayoría de las API de Amazon Location Service.
- El rendimiento de las solicitudes de mapas con claves de API suele ser más rápido que en escenarios similares con Amazon Cognito. Una autenticación más sencilla significa menos viajes de ida y vuelta al servicio y solicitudes almacenadas en caché cuando se vuelve a obtener el mismo mosaico de mapa en un período de tiempo corto.
- Con Amazon Cognito, puede utilizar su propio proceso de autenticación o combinar varios métodos de autenticación mediante Amazon Cognito Federated Identities. Para obtener más información, consulte [Introducción a las identidades federadas](#) en la Guía para desarrolladores de Amazon Cognito.

Crear claves de API

Puede crear una clave de API y asociarla a uno o más recursos de su Cuenta de AWS.

Puedes crear una clave de API mediante la consola de Amazon Location Service AWS CLI, las o las API de ubicación de Amazon.

Console

Para crear una clave de API mediante la consola de Amazon Location Service

1. En la [consola de Amazon Location](#), seleccione las claves de API en el menú de la izquierda.

2. En la página Claves de API, elija Crear clave de API.
3. En la página Crear clave de API, rellene la siguiente información:
 - Nombre: un nombre para su clave de API, por ejemplo MyWebAppKey.
 - Descripción: una descripción opcional para su clave de API.
 - Recursos: elija los recursos de Amazon Location a los que desea acceder con esta clave de API en el menú desplegable. Puede agregar más de un recurso seleccionando agregar recurso.
 - Acciones: especifique las acciones que desea autorizar con esta clave de API. Debe seleccionar al menos una acción para que coincida con cada tipo de recurso que haya seleccionado. Por ejemplo, si ha seleccionado un recurso de ubicación, debe seleccionar al menos una de las opciones de la sección Acciones de ubicación.
 - Hora de caducidad: si lo desea, añada una fecha y una hora de caducidad para su clave de API. Para obtener más información, consulte [Gestionar la vida útil de claves de API](#).
 - Referentes: si lo desea, añada uno o más dominios en los que pueda usar la clave de API. Por ejemplo, si la clave de la API es para permitir que una aplicación se ejecute en el sitio `webexample.com`, puede poner `*.example.com/` como referente permitido.
 - Etiquetas: si lo desea, añada etiquetas a la clave de API.
4. Seleccione Crear clave de API para crear la clave de API.
5. En la página de detalles de la clave de API, puede ver información sobre la clave de API que ha creado. Seleccione Mostrar clave de API para ver el valor de clave que utiliza al llamar a las API de Amazon Location. El valor de la clave tendrá el formato `v1.public.a1b2c3d4...`. Para obtener más información sobre el uso de la clave de API para la representación de mapas, consulte [Usar una clave de API para renderizar un mapa](#).

API

Para crear una clave de API mediante las API de Amazon Location

Use la operación de [CreateKey](#) de las API de Amazon Location.

El siguiente ejemplo es una solicitud de API para crear una clave de API llamada sin fecha de caducidad y *ExampleKey* con acceso a un único recurso de mapa.

```
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
}
```

La respuesta incluye el valor de la clave de API que se debe utilizar al acceder a los recursos de las aplicaciones. El valor de la clave tendrá el formato `v1.public.a1b2c3d4...`. Para obtener más información acerca del uso de la clave de API de para renderizar mapas, consulte [Usar una clave de API para renderizar un mapa](#).

También puede usar la [DescribeKey](#) API para buscar el valor de una clave más tarde.

AWS CLI

Para crear una clave de API con los comandos de la AWS CLI

Utilice el comando [create-key](#).

En el siguiente ejemplo, se crea una clave de API llamada sin fecha de caducidad y *ExampleKey* con acceso a un único recurso de mapa.

```
aws location \
  create-key \
  --key-name ExampleKey \
  --restrictions '{"AllowActions":["geo:GetMap*"],"AllowResources":
["arn:aws:geo:region:map/mapname"]}' \
  --no-expiry
```

La respuesta incluye el valor de la clave de API que se debe utilizar al acceder a los recursos de las aplicaciones. El valor de la clave tendrá el formato `v1.public.a1b2c3d4...`. Para obtener más información acerca del uso de la clave de API de para renderizar mapas, consulte [Usar una clave de API para renderizar un mapa](#). La respuesta a `create-key` tiene este aspecto.

```
{
```

```
"Key": "v1.public.a1b2c3d4...",
"KeyArn": "arn:aws:geo:region:accountId:api-key/ExampleKey",
"KeyName": "ExampleKey",
"CreateTime": "2023-02-06T22:33:15.693Z"
}
```

También puede utilizar la `describe-key` para buscar el valor de la clave más tarde. En el siguiente ejemplo, se muestra cómo `describe-key` invocar una clave de API denominada *ExampleKey*.

```
aws location describe-key \
  --key-name ExampleKey
```

Usar una clave de API para llamar a una API de Amazon Location

Tras crear una clave de API, puede usar el valor de la clave para realizar llamadas a las API de Amazon Location de su aplicación.

Las API que admiten las claves de API tienen un parámetro adicional que toma el valor de la clave de API. Por ejemplo, si llama a la `GetPlace` API, puede rellenar el parámetro [clave](#) de la siguiente manera

```
GET /places/v0/indexes/IndexName/places/PlaceId?key=KeyVaLue
```

Si rellenas este valor, no necesitas autenticar la llamada a la API con AWS Sig v4 como lo harías normalmente.

Para JavaScript los desarrolladores, puedes usar Amazon Location como ayuda [JavaScript Ayudante de autenticación](#) para autenticar las operaciones de la API con las claves de la API.

Para los desarrolladores de dispositivos móviles, puedes usar los siguientes SDK de autenticación móvil de Amazon Location:

- [SDK de autenticación móvil de Amazon Location Service para iOS](#)
- [SDK de autenticación móvil de Amazon Location Service para Android](#)

Para AWS CLI los usuarios, al usar el `--key` parámetro, también deben usarlo para evitar firmar con Sig v4. `--no-sign-request`

Note

Si incluyes una firma Sig v4 key y una firma AWS Sig v4 en una llamada a Amazon Location Service, solo se utilizará la clave de API.

Usar una clave de API para renderizar un mapa

Puedes usar el valor de la clave de la API para representar un mapa en tu aplicación utilizando MapLibre. Esto es un poco diferente al uso de las claves de API en otras API de Amazon Location a las que llamas directamente, ya MapLibre que hace esas llamadas por ti.

En el siguiente código de ejemplo, se muestra el uso de la clave de API para representar un mapa en una página web sencilla mediante el control de mapas MapLibre GL JS. Para que este código funcione correctamente, sustituya la *v1.public.your-api-key-value*, *us-east-1* y *ExampleMap* cadenas con valores que coincidan con su. Cuenta de 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>
```

Gestionar la vida útil de claves de API

Puede crear claves de API que funcionen indefinidamente. Sin embargo, si quiere crear una clave de API temporal, rotar las claves de API de forma regular o revocar una clave de API existente, puede utilizar la caducidad de la clave de API.

Al crear una nueva clave de API o actualizar una existente, puede establecer el tiempo de caducidad de esa clave de API.

- La clave de API se desactiva automáticamente cuando alcanza su hora de vencimiento. Las claves inactivas ya no se pueden usar para realizar solicitudes de mapas.
- Puede eliminar una clave de API 90 días después de desactivarla.
- Si tiene una clave inactiva que aún no ha eliminado, puede restaurarla actualizando la fecha de caducidad a una fecha futura.
- Para crear una clave permanente, puede eliminar la fecha de caducidad.
- Si intenta desactivar una clave de API que se haya utilizado en los últimos 7 días, se le pedirá que confirme que desea realizar el cambio. Si utilizas la API de Amazon Location Service, o la AWS CLI, recibirás un error, a menos que establezcas el `ForceUpdate` parámetro en `true`.

Cómo permitir el acceso a sus aplicaciones como invitado no autenticado con Amazon Cognito

Puede utilizar la autenticación de Amazon Cognito como alternativa al uso directo AWS Identity and Access Management (IAM) tanto con los SDK de frontend como con las solicitudes HTTPS directas.

Es posible que desee utilizar esta forma de autenticación por los siguientes motivos:

- **Usuarios no autenticados:** si tiene un sitio web con usuarios anónimos, puede utilizar los grupos de identidades de Amazon Cognito. Para obtener más información, consulte la sección en [the section called “Uso de Amazon Cognito”](#).
- **Su propia autenticación:** si desea utilizar su propio proceso de autenticación o combinar varios métodos de autenticación, puede utilizar Amazon Cognito Federated Identities. Para obtener más información, consulte [Introducción a las identidades federadas](#) en la Guía para desarrolladores de Amazon Cognito.

Amazon Cognito ofrece autenticación, autorización y administración de usuarios para aplicaciones móviles y web. Puede usar los grupos de identidades no autenticadas de Amazon Cognito con Amazon Location como una forma de que las aplicaciones recuperen credenciales temporales y restringidas. AWS

Para obtener más información, consulte [Introducción a los grupos de usuarios](#) en la Guía para desarrolladores de Amazon Cognito.

Note

Para los desarrolladores de dispositivos móviles, Amazon Location proporciona SDK de autenticación móvil para iOS y Android. Consulte los siguientes repositorios de github para obtener más información:

- [SDK de autenticación móvil de Amazon Location Service para iOS](#)
- [SDK de autenticación móvil de Amazon Location Service para Android](#)

Creación de un grupo de identidades en Amazon Cognito

Puede crear grupos de identidades de Amazon Cognito para permitir el acceso de invitados no autenticados a su aplicación a través de la consola de Amazon Cognito, las API de Amazon Cognito o AWS CLI las API de Amazon Cognito.

Important

El grupo que cree debe estar en la misma AWS región Cuenta de AWS y en la misma región que los recursos de Amazon Location Service que esté utilizando.

Puede utilizar las políticas de IAM asociadas a los roles de identidad no autenticados con las siguientes acciones:

- `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`

La inclusión de otras acciones de Amazon Location no tendrá ningún efecto y las identidades no autenticadas no podrán llamarlas.

Example

Para crear un grupo de identidades con la consola de Amazon Cognito

1. Vaya a la [consola de Amazon Cognito](#).
2. Seleccione Administrar grupos de identidades.
3. Seleccione Crear un nuevo grupo de identidades y, a continuación, introduzca un nombre para su grupo de identidades.
4. En Identidades sin autenticar, elija Habilitar el acceso a identidades sin autenticar.
5. Elija Crear grupo.
6. Elija los roles de IAM que quiera usar con su grupo de identidades.

7. Amplíe Ver detalles.
8. En Identidades no autenticadas, introduzca un nombre de rol.
9. Amplíe la sección Ver documento de política y, a continuación, seleccione Editar para agregar la política.
10. Agregue su política para dar acceso a sus recursos.

Los siguientes son ejemplos de políticas para mapas, lugares, rastreadores y rutas. Para usar los ejemplos en su propia política, reemplace los marcadores de *región* e *ID de cuenta*:

Maps policy example

La siguiente política concede acceso de solo lectura a un recurso de mapa denominado.

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"
    }
  ]
}
```

Si agrega una [condición de IAM](#) que coincida con `aws:referrer`, podrá limitar el acceso del navegador a sus recursos a una lista de direcciones URL o prefijos de direcciones URL. El siguiente ejemplo permite el acceso a un recurso de mapa cuyo nombre `RasterEsriImagery` proviene únicamente del sitio web `example.com`:

Warning

Si bien `aws:referrer` puede limitar el acceso, no es un mecanismo de seguridad. Es peligroso incluir un valor de encabezado de referencia conocido públicamente.

Las partes no autorizadas podrían utilizar navegadores personalizados o modificados para proporcionar cualquier valor `aws:referer` que eligieran. Por lo tanto, no `aws:referer` debe utilizarse para impedir que personas no autorizadas realicen solicitudes directas AWS. Se ofrece únicamente para que los clientes puedan proteger su contenido digital, como el contenido almacenado en Amazon S3, para evitar las referencias en sitios de terceros no autorizados. Para obtener más información, consulte [AWS:referer](#).

```
{
  "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/*"
          ]
        }
      }
    }
  ]
}
```

Si [utiliza Tangram](#) para mostrar un mapa, no utilizará los descriptores de estilo, glifos ni sprites que devuelve la API de Maps. En su lugar, se configura apuntando a un archivo.zip que contiene las reglas de estilo y los recursos necesarios. La siguiente política otorga acceso de solo lectura a un recurso de mapa con el nombre *ExampleMap* de la `GetMapTile` operación.

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

```

    "Effect": "Allow",
    "Action": [
      "geo:GetMapTile"
    ],
    "Resource": "arn:aws:geo:region:accountID:map/ExampleMap"
  }
]
}

```

Places policy example

La siguiente política concede acceso de solo lectura a un recurso de indexación de lugares denominado *ExamplePlaceIndex* para buscar lugares mediante texto o posiciones.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PlacesReadOnly",
      "Effect": "Allow",
      "Action": [
        "geo:SearchPlaceIndex*",
        "geo:GetPlace"
      ],
      "Resource": "arn:aws:geo:region:accountID:place-index/ExamplePlaceIndex"
    }
  ]
}

```

Si agrega una [condición de IAM](#) que coincida con `aws:referrer`, podrá limitar el acceso del navegador a sus recursos a una lista de direcciones URL o prefijos de direcciones URL. El siguiente ejemplo deniega el acceso a un recurso de índice de lugares cuyo nombre *ExamplePlaceIndex* proviene de todos los sitios web de referencia, excepto. `example.com`

Warning

Si bien `aws:referrer` puede limitar el acceso, no es un mecanismo de seguridad. Es peligroso incluir un valor de encabezado de referencia conocido públicamente. Las partes no autorizadas podrían utilizar navegadores personalizados o modificados para proporcionar cualquier valor `aws:referrer` que eligieran. Por lo tanto, no

`aws:referer` debe utilizarse para impedir que personas no autorizadas realicen AWS solicitudes directas. Se ofrece únicamente para que los clientes puedan proteger su contenido digital, como el contenido almacenado en Amazon S3, para evitar las referencias en sitios de terceros no autorizados. Para obtener más información, consulte [AWS:referer](#).

```
{
  "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

La siguiente política otorga acceso a un recurso de rastreo denominado *ExampleTracker* para actualizar las posiciones de los dispositivos.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "UpdateDevicePosition",
      "Effect": "Allow",
      "Action": [
        "geo:BatchUpdateDevicePosition"
      ]
    }
  ]
}
```

```

    "Resource": "arn:aws:geo:region:accountID:tracker/ExampleTracker"
  }
]
}

```

Si agrega una [condición de IAM](#) que coincida con `aws:referer`, podrá limitar el acceso del navegador a sus recursos a una lista de direcciones URL o prefijos de direcciones URL. El siguiente ejemplo deniega el acceso a un recurso de seguimiento cuyo nombre *ExampleTracker* proviene de todos los sitios web de referencia, excepto `example.com`.

Warning

Si bien `aws:referer` puede limitar el acceso, no es un mecanismo de seguridad. Es peligroso incluir un valor de encabezado de referencia conocido públicamente. Las partes no autorizadas podrían utilizar navegadores personalizados o modificados para proporcionar cualquier valor `aws:referer` que eligieran. Por lo tanto, no `aws:referer` debe utilizarse para impedir que terceros no autorizados realicen AWS solicitudes directas. Se ofrece únicamente para que los clientes puedan proteger su contenido digital, como el contenido almacenado en Amazon S3, para evitar las referencias en sitios de terceros no autorizados. Para obtener más información, consulte [AWS:referer](#).

```

{
  "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

La siguiente política otorga acceso a un recurso de calculadora de rutas denominado *ExampleCalculator* para calcular una ruta.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RoutesReadOnly",
      "Effect": "Allow",
      "Action": [
        "geo:CalculateRoute"
      ],
      "Resource": "arn:aws:geo:region:accountID:route-
calculator/ExampleCalculator"
    }
  ]
}
```

Si agrega una [condición de IAM](#) que coincida con `aws:referrer`, podrá limitar el acceso del navegador a sus recursos a una lista de direcciones URL o prefijos de direcciones URL. El siguiente ejemplo deniega el acceso a una calculadora de rutas con el nombre *ExampleCalculator* de todos los sitios web de referencia, excepto `example.com`.

Warning

Si bien `aws:referrer` puede limitar el acceso, no es un mecanismo de seguridad. Es peligroso incluir un valor de encabezado de referencia conocido públicamente. Las partes no autorizadas podrían utilizar navegadores personalizados o modificados para proporcionar cualquier valor `aws:referrer` que eligieran. Por lo tanto, no `aws:referrer` debe utilizarse para impedir que personas no autorizadas realicen AWS solicitudes directas. Se ofrece únicamente para que los clientes puedan proteger su contenido digital, como el contenido almacenado en Amazon S3, para evitar las referencias en sitios de terceros no autorizados. Para obtener más información, consulte [AWS:referrer](#).


```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "geo:*",
      "Resource": "arn:aws:geo:us-west-2:111122223333:route-
calculator/ExampleCalculator",
      "Condition": {
        "StringLike": {
          "aws:referrer": [
            "https://example.com/*",
            "https://www.example.com/*"
          ]
        }
      }
    }
  ]
}
```

Note

Si bien los grupos de identidades no autenticadas están pensados para ser expuestos en sitios de Internet no seguros, tenga en cuenta que se cambiarán por credenciales estándar y de duración limitada. AWS

Es importante definir adecuadamente los roles de IAM asociadas a los grupos de identidades no autenticadas.

11. Elija Permitir para crear sus grupos de identidades.

El grupo de identidades resultante sigue la sintaxis `<region>:<GUID>`.

Por ejemplo:

```
us-east-1:1sample4-5678-90ef-aaaa-1234abcd56ef
```

Para ver más ejemplos de políticas específicas de Amazon Location, consulte [the section called “Ejemplos de políticas basadas en identidades”](#).

Uso de los grupos de identidades de Amazon Cognito en JavaScript

En el siguiente ejemplo, se intercambia el grupo de identidades no autenticadas que ha creado por credenciales que, a continuación, se utilizan para obtener el descriptor de estilo del recurso de mapa.

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

Las credenciales recuperadas de identidades no autenticadas son válidas durante una hora.

A continuación, se muestra un ejemplo de una función que renueva automáticamente las credenciales antes de que caduquen.

```
async function refreshCredentials() {
  await credentials.refreshPromise();
  // schedule the next credential refresh when they're about to expire
  setTimeout(refreshCredentials, credentials.expireTime - new Date());
}
```

Para simplificar este trabajo, puede usar la [JavaScript Ayudante de autenticación](#) de Amazon Location. Esto sustituye tanto a la obtención de las credenciales como a la actualización de las mismas. En este ejemplo, se usa el SDK de la versión 3 AWS . JavaScript

```
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));
```

Siguientes pasos

- Para modificar roles, vaya a la [consola de IAM](#).
- Para gestionar sus grupos de identidades, vaya a la consola de [Amazon Cognito](#).

Monitoreo de Amazon Location Service

Al utilizar Amazon Location Service, puede monitorizar su uso y sus recursos a lo largo del tiempo mediante:

- Amazon CloudWatch. Supervisa sus recursos de Amazon Location Service y proporciona métricas con estadísticas prácticamente en tiempo real.
- AWS CloudTrail. Proporciona un seguimiento de eventos de todas las llamadas a las API de Amazon Location Service.

En esta sección se proporciona información relativa al uso de estos servicios.

Temas

- [Supervisión de Amazon Location Service con Amazon CloudWatch](#)
- [Registro y monitoreo con AWS CloudTrail](#)

Supervisión de Amazon Location Service con Amazon CloudWatch

Amazon CloudWatch monitorea tus AWS recursos y las aplicaciones en las que ejecutas prácticamente AWS en tiempo real. Puedes monitorear los recursos de Amazon Location utilizando CloudWatch, que recopila datos sin procesar y procesa las métricas para convertirlas en estadísticas significativas casi en tiempo real. Puedes ver información histórica de hasta 15 meses o buscar métricas para consultarlas en la CloudWatch consola de Amazon y obtener más información sobre tus recursos de ubicaciones de Amazon. Puede establecer alarmas definiendo umbrales y enviar notificaciones o realizar acciones cuando se alcancen dichos umbrales.

Para obtener más información, consulta la [Guía del CloudWatch usuario de Amazon](#)

Temas

- [Métricas de Amazon Location Service exportadas a Amazon CloudWatch](#)
- [Ver las métricas de Amazon Location Service](#)
- [Crea CloudWatch alarmas para las métricas de Amazon Location Service](#)
- [Se usa CloudWatch para monitorear el uso comparándolo con las cuotas](#)
- [CloudWatch ejemplos de métricas para Amazon Location Service](#)

Métricas de Amazon Location Service exportadas a Amazon CloudWatch

Las métricas son puntos de datos ordenados por tiempo que se exportan a CloudWatch. Una dimensión es un par nombre/valor que identifica la métrica. Para obtener más información, consulta [Uso de CloudWatch métricas](#) y [CloudWatch dimensiones](#) en la Guía del CloudWatch usuario de Amazon.

Las siguientes son las métricas a las que Amazon Location Service exporta CloudWatch en el espacio de `AWS/Location` nombres.

Métrica	Descripción
CallCount	El número de llamadas realizadas a un punto de conexión de la API determinado.

Métrica	Descripción
	<p>Dimensiones válidas: nombres de API de Amazon Location Service</p> <p>Estadísticas válidas: suma</p> <p>Unidades: recuento</p>
ErrorCount	<p>El número de respuestas de error a las llamadas realizadas a un punto de conexión de la API determinado.</p> <p>Dimensiones válidas: nombres de API de Amazon Location Service</p> <p>Estadísticas válidas: suma</p> <p>Unidades: recuento</p>
SuccessCount	<p>El número de llamadas realizadas correctamente a un punto de conexión de la API determinado.</p> <p>Dimensiones válidas: nombres de API de Amazon Location Service</p> <p>Estadísticas válidas: suma</p> <p>Unidades: recuento</p>
CallLatency	<p>El tiempo que tarda la operación en procesar y devolver una respuesta cuando se realiza una llamada a un punto de conexión de la API determinado.</p> <p>Dimensiones válidas: nombres de API de Amazon Location Service</p> <p>Estadísticas válidas: Promedio</p> <p>Unidades: milisegundos</p>

Ver las métricas de Amazon Location Service

Puedes ver las métricas de Amazon Location Service en la Amazon CloudWatch consola o mediante la Amazon CloudWatch API.

Para ver las métricas mediante la consola CloudWatch

Example

1. Abra la CloudWatch consola en <https://console.aws.amazon.com/cloudwatch/>.
2. En el panel de navegación, seleccione Métricas.
3. En la pestaña Todas las métricas, seleccione el espacio de nombres de Amazon Location.
4. Seleccione el tipo de métrica que desea ver.
5. Seleccione una métrica para agregarla al gráfico.

Para obtener más información, consulta [Ver métricas disponibles](#) en la Guía del CloudWatch usuario de Amazon.

Crea CloudWatch alarmas para las métricas de Amazon Location Service

Puedes usarlo CloudWatch para configurar alarmas en tus métricas de Amazon Location Service. Por ejemplo, puedes crear una alarma CloudWatch para enviar un correo electrónico cada vez que se produzca un aumento en el recuento de errores.

Los siguientes temas ofrecen información general de alto nivel sobre cómo definir alarmas mediante CloudWatch. Para obtener instrucciones detalladas, consulta [Uso de alarmas](#) en la Guía del CloudWatch usuario de Amazon.

Para configurar las alarmas mediante la CloudWatch consola

Example

1. Abra la CloudWatch consola en <https://console.aws.amazon.com/cloudwatch/>.
2. En el panel de navegación, elija Alarmas.
3. Elija Crear Alarma.
4. Elija Select Metric (Seleccionar métrica).
5. En la pestaña Todas las métricas, seleccione el espacio de nombres de Amazon Location.
6. Seleccione una categoría métrica.
7. Busque la fila con la métrica para la que desee crear una alarma y, a continuación, active la casilla de verificación situada junto a esta fila.
8. Elija Seleccionar métrica.
9. En Métrica, rellene los valores.

10Especifique las condiciones de la alarma.

11Elija Siguiente.


12Si desea enviar una notificación cuando se cumplan las condiciones de alarma:

- En Activación del estado de alarma, seleccione el estado de alarma para solicitar el envío de una notificación.
- En Seleccione un tema de SNS, elija Crear nuevo tema para crear un nuevo tema de Amazon Simple Notification Service (Amazon SNS). Introduzca el nombre del tema y el correo electrónico al que desea enviar la notificación.
- En Enviar una notificación, introduzca direcciones de correo electrónico adicionales a las que enviar la notificación.
- Seleccione Agregar notificación. Esta lista se guarda y aparece en el campo para futuras alarmas.

13.Cuando haya terminado, elija Siguiente.

14Ingrese un nombre y una descripción para la alarma y, a continuación, elija Siguiente.

15.Confirme los detalles de la alarma y, a continuación, seleccione Siguiente.

 Note

Al crear un nuevo tema de Amazon SNS, debe verificar la dirección de correo electrónico antes de poder enviar una notificación. Si el correo electrónico no está verificado, no se recibirá la notificación cuando se inicie una alarma debido a un cambio de estado.

Para obtener más información sobre cómo configurar las alarmas mediante la CloudWatch consola, consulta [Crear una alarma que envíe correo electrónico](#) en la Guía del CloudWatch usuario de Amazon.

Se usa CloudWatch para monitorear el uso comparándolo con las cuotas

Puedes crear CloudWatch alarmas de Amazon para que te notifiquen cuando la utilización de una cuota determinada supere un umbral configurable. Esto le permite reconocer cuándo está cerca de sus límites de cuota y adaptar su utilización para evitar sobrecostos o solicitar un aumento de la cuota, si es necesario. Para obtener información sobre cómo CloudWatch monitorizar las cuotas, consulta [Visualización de tus cuotas de servicio y configuración de alarmas](#) en la Guía del CloudWatch usuario de Amazon.

CloudWatch ejemplos de métricas para Amazon Location Service

Puedes usar la [GetMetricData](#) API para recuperar las métricas de Amazon Location.

- Por ejemplo, puedes controlar `CallCount` y establecer una alarma para cuando se produzca un descenso en el número.

Supervisar las métricas de `CallCount` para `SendDeviceLocation` puede ayudarte a obtener una perspectiva sobre los activos rastreados. Si las cifras de `CallCount` disminuyen, significa que los activos rastreados, como una flota de camiones, han dejado de enviarse a sus ubicaciones actuales. Configurar una alarma para ello puede avisarle que se ha producido un problema.

- Como otro ejemplo, puede monitorear `ErrorCount` y configurar una alarma para cuando se produzca un aumento en el número.

Los rastreadores deben estar asociados a las colecciones de geovallas para que las ubicaciones de los dispositivos se puedan evaluar comparándolas con las geovallas. Si tiene una flota de dispositivos que requiere actualizaciones de ubicación continuas, ver `CallCount` de `BatchEvaluateGeofence` o `BatchPutDevicePosition` descender a cero indica que las actualizaciones han dejado de fluir.

El siguiente es un ejemplo de resultado [GetMetricData](#) con las métricas para `CallCount` y `ErrorCount` para la creación de recursos de mapas.

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

Registro y monitoreo con AWS CloudTrail

AWS CloudTrail es un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un AWS servicio. CloudTrail registra todas las llamadas a la API como eventos. Puede utilizar Amazon Location Service CloudTrail para supervisar las llamadas a la API, que incluyen las llamadas desde la consola de Amazon Location Service y las llamadas del AWS SDK a las operaciones de la API de Amazon Location Service.

Al crear una ruta, puede habilitar la entrega continua de CloudTrail eventos a un bucket de S3, incluidos los eventos de Amazon Location Service. Si no configura una ruta, podrá ver los eventos más recientes en la CloudTrail consola, en el historial de eventos. Con la información recopilada por CloudTrail, puedes determinar la solicitud que se realizó a Amazon Location Service, la dirección IP desde la que se realizó la solicitud, quién la hizo, cuándo se realizó y detalles adicionales.

Para obtener más información al respecto CloudTrail, consulte la [Guía AWS CloudTrail del usuario](#).

Temas

- [Información sobre Amazon Location Service en CloudTrail](#)
- [Descripción de las entradas del archivo de registro de Amazon Location Service](#)

Información sobre Amazon Location Service en CloudTrail

CloudTrail está activado en tu AWS cuenta al crearla. Cuando se produce una actividad en Amazon Location Service, esa actividad se registra en un CloudTrail evento junto con otros eventos de AWS servicio en el historial de eventos. Puede ver, buscar y descargar los últimos eventos de la cuenta de AWS. Para obtener más información, consulte [Visualización de eventos con el historial de CloudTrail eventos](#).

Para mantener un registro continuo de los eventos de la cuenta de AWS, incluidos los eventos de Amazon Location Service, cree un registro de seguimiento. Un rastro permite CloudTrail entregar los archivos de registro a un bucket de S3. De manera predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las regiones de AWS. El registro de seguimiento registra los eventos de todas las regiones de la partición de AWS y envía los archivos de registro al bucket de S3 especificado. Además, puede configurar otros AWS servicios para analizar más a fondo los datos de eventos recopilados en los CloudTrail registros y actuar en función de ellos.

Para más información, consulte los siguientes temas:

- [Introducción a la creación de registros de seguimiento](#)
- [CloudTrail Integraciones y servicios compatibles](#)
- [Configuración de las notificaciones de Amazon SNS para CloudTrail](#)
- [Recibir archivos de CloudTrail registro de varias regiones](#) y [recibir archivos de CloudTrail registro de varias cuentas](#)

Todas las acciones de Amazon Location Service se registran CloudTrail y se documentan en las [referencias de la API de Amazon Location Service](#). Por ejemplo, las llamadas a `UpdateTracker` y `DescribeTracker` las acciones generan entradas en los archivos de CloudTrail registro. `CreateTracker`

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad le ayudará a determinar si la solicitud se realizó:

- Con credenciales de usuario root o AWS Identity and Access Management (usuario de IAM).
- Con credenciales de seguridad temporales de un rol o de un usuario federado.
- Desde otro servicio de AWS.

Para obtener más información, consulte el [Elemento `userIdentity` de CloudTrail](#).

Descripción de las entradas del archivo de registro de Amazon Location Service

Un rastro es una configuración que permite la entrega de eventos como archivos de registro a un bucket de S3 que especifique o a Amazon CloudWatch Logs. Para obtener más información, consulte [Trabajar con archivos de CloudTrail registro](#) en la Guía del AWS CloudTrail usuario.

CloudTrail los archivos de registro contienen una o más entradas de registro. Un evento representa una única solicitud de cualquier origen e incluye información sobre la operación solicitada, la fecha y la hora de la operación, los parámetros de la solicitud, etcétera.

Note

CloudTrail Los archivos de registro no son un seguimiento ordenado de las llamadas a la API pública, por lo que no aparecen en ningún orden específico. Para determinar el orden de las operaciones, utilice [eventTime](#).

En el siguiente ejemplo, se muestra una entrada de CloudTrail registro que demuestra la `CreateTracker` operación, lo que crea un recurso de seguimiento.

```
{
  "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",
}

```

A continuación se muestra una entrada de registro de la operación `DescribeTracker`, que devuelve los detalles de un recurso de seguimiento.

```

{
    "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",
}

```

Creación de recursos de Amazon Location Service con AWS CloudFormation

Amazon Location Service está integrado a AWS CloudFormation, un servicio que ayuda a modelar y configurar sus recursos de AWS para que pueda dedicar menos tiempo a crear y administrar recursos e infraestructuras. Usted crea una plantilla que describa todos los recursos de AWS que

desea (tales como recursos Amazon Location) y AWS CloudFormation aprovisiona y configura estos recursos por usted.

Cuando utiliza AWS CloudFormation, puede volver a utilizar la plantilla para configurar los recursos de Amazon Location de forma coherente y repetida. Solo tiene que describir los recursos una vez y luego aprovisionar los mismos recursos una y otra vez en varias cuentas y regiones de AWS.

Amazon Location y plantillas AWS CloudFormation

Para aprovisionar y configurar los recursos de Amazon Location y los servicios relacionados, debe entender las [plantillas de AWS CloudFormation](#). Las plantillas son archivos de texto con formato JSON o YAML. Estas plantillas describen los recursos que desea aprovisionar en sus pilas de AWS CloudFormation. Si no conoce bien JSON o YAML, puede utilizar Designer de AWS CloudFormation para comenzar a utilizar las plantillas de AWS CloudFormation. Para obtener más información, consulte [¿Qué es Designer de AWS CloudFormation?](#) en la Guía del usuario de AWS CloudFormation.

Amazon Location admite la creación de los siguientes tipos de recursos en AWS CloudFormation:

- [AWS::Location::Map](#)
- [AWS::Location::PlaceIndex](#)
- [AWS::Location::RouteCalculator](#)
- [AWS::Location::Tracker](#)
- [AWS::Location::TrackerConsumer](#)
- [AWS::Location::GeofenceCollection](#)

Para obtener más información, incluidos ejemplos de plantillas JSON y YAML para los recursos de Amazon Location Service, consulte la [referencia del tipo de recursos de Amazon Location Service](#) en la Guía del usuario de AWS CloudFormation.

Obtener más información sobre AWS CloudFormation

Para conocer más información acerca de AWS CloudFormation, consulte los siguientes recursos:

- [AWS CloudFormation](#)
- [Guía del usuario de AWS CloudFormation](#)
- [Referencia de la API de AWS CloudFormation](#)

- [Guía del usuario de la interfaz de la línea de comandos de AWS CloudFormation](#)

Seguridad en Amazon Location Service

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de los centros de datos y las arquitecturas de red diseñados para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre AWS usted y usted. El [modelo de responsabilidad compartida](#) la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la Nube de AWS. AWS también le proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los [AWS programas](#) de de . Para obtener más información sobre los programas de conformidad que se aplican a Amazon Location Service, consulta [AWS Servicios incluidos en el ámbito del programa de conformidad AWS](#) .
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. Usted también es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos aplicables.

Esta documentación le permite comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza Amazon Location. En los siguientes temas, se mostrará cómo configurar Amazon Location para satisfacer sus objetivos de seguridad y conformidad. También aprenderás a usar otros AWS servicios que te ayudan a monitorear y proteger tus recursos de Amazon Location.

Temas

- [Protección de los datos en Amazon Location Service](#)
- [Identity and Access Management para Amazon Location Service](#)
- [Respuesta a incidentes en Amazon Location Service](#)
- [Validación de conformidad para Amazon Location Service](#)
- [Resiliencia de Amazon Location Service](#)
- [Seguridad de la infraestructura en Amazon Location Service](#)
- [Configuración y análisis de vulnerabilidades en Amazon Location](#)
- [Prevención de la sustitución confusa entre servicios](#)
- [Prácticas recomendadas de seguridad para Amazon Location Service](#)

- [Prácticas recomendadas para Amazon Location Service](#)

Protección de los datos en Amazon Location Service

El [modelo de](#) se aplica a protección de datos en Amazon Location Service. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global en la que se ejecutan todos los Nube de AWS. Usted es responsable de mantener el control sobre el contenido alojado en esta infraestructura. Usted también es responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulte las [Preguntas frecuentes sobre la privacidad de datos](#). Para obtener información sobre la protección de datos en Europa, consulte la publicación de blog sobre el [Modelo de responsabilidad compartida de AWS y GDPR](#) en el Blog de seguridad de AWS .

Con fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utilice la autenticación multifactor (MFA) en cada cuenta.
- Utilice SSL/TLS para comunicarse con los recursos. AWS Se recomienda el uso de TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad de los usuarios con. AWS CloudTrail
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados Servicios de AWS.
- Utilice servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita módulos criptográficos validados por FIPS 140-2 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulte [Estándar de procesamiento de la información federal \(FIPS\) 140-2](#).

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como, por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabajas con Amazon Location u otro Servicios de AWS dispositivo mediante la consola, la API o AWS los SDK. AWS CLI Cualquier dato que

ingrese en etiquetas o campos de formato libre utilizados para nombres se puede emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya información de credenciales en la URL a fin de validar la solicitud para ese servidor.

Privacidad de datos

Con Amazon Location Service, mantiene el control de los datos de su organización. Amazon Location anonimiza todas las consultas enviadas a los proveedores de datos al eliminar los metadatos de los clientes y la información de la cuenta.

Amazon Location no utiliza proveedores de datos para el seguimiento y la geolocalización. Esto significa que sus datos confidenciales permanecen en su AWS cuenta. Esto ayuda a proteger la información de ubicación confidencial, como la ubicación de las instalaciones, los activos y el personal, de terceros, a proteger la privacidad de los usuarios y a reducir el riesgo de seguridad de la aplicación.

Para obtener información adicional, consulte las [preguntas frecuentes sobre la privacidad de los datos de AWS](#).

Retención de datos en la ubicación de Amazon

Las siguientes características se refieren a la forma en que Amazon Location recopila y almacena los datos para el servicio:

- **Rastreadores de Amazon Location Service:** cuando utiliza las API de Rastreadores para rastrear la ubicación de las entidades, se pueden almacenar sus coordenadas. Las ubicaciones de los dispositivos se almacenan durante 30 días antes de que el servicio las elimine.
- **Geocercas de Amazon Location Service:** cuando utiliza las API de geocercas para definir áreas de interés, el servicio almacena las geometrías que ha proporcionado. Deben eliminarse de forma explícita.

Note

Al eliminar tu AWS cuenta, eliminas todos los recursos que contiene. Para obtener información adicional, consulte las [preguntas frecuentes sobre la privacidad de los datos de AWS](#).

Cifrado de datos en reposo para Amazon Location Service

Amazon Location Service proporciona cifrado de forma predeterminada para proteger los datos confidenciales de los clientes en reposo mediante claves de cifrado AWS propias.

- **AWS claves propias:** Amazon Location utiliza estas claves de forma predeterminada para cifrar automáticamente los datos de identificación personal. No puedes ver, administrar ni usar las claves AWS propias, ni auditar su uso. Sin embargo, no tiene que realizar ninguna acción ni cambiar ningún programa para proteger las claves que cifran sus datos. Para obtener más información, consulte las [claves propiedad de AWS](#) en la Guía para desarrolladores de AWS Key Management Service .

El cifrado de los datos en reposo de forma predeterminada ayuda a reducir la sobrecarga operativa y la complejidad que implica la protección de los datos confidenciales. Al mismo tiempo, le permite crear aplicaciones seguras que cumplen con los estrictos requisitos normativos y de conformidad con el cifrado.

Si bien no puedes deshabilitar esta capa de cifrado ni seleccionar un tipo de cifrado alternativo, puedes añadir una segunda capa de cifrado sobre las claves de cifrado que ya poseas AWS si eliges una clave gestionada por el cliente al crear tus recursos de recopilación de rastreadores y cercas geográficas:

- **Claves administradas por el cliente:** Amazon Location admite el uso de una clave simétrica administrada por el cliente que usted crea, posee y administra para agregar una segunda capa de cifrado sobre el cifrado propio existente AWS . Como usted tiene el control total de este cifrado, puede realizar dichas tareas como:
 - Establecer y mantener políticas de claves
 - Establecer y mantener concesiones y políticas de IAM
 - Habilitar y deshabilitar políticas de claves
 - Rotar el material criptográfico
 - Agregar etiquetas.
 - Crear alias de clave
 - Programar la eliminación de claves

Para obtener más información, consulte las [claves administradas por el cliente](#) en la Guía para desarrolladores de AWS Key Management Service .

En la siguiente tabla se resume cómo Amazon Location cifra los datos de identificación personal.

Tipo de datos	AWS cifrado de clave propia	Cifrado de claves administradas por el cliente (opcional)
<p><code>Position</code></p> <p>Una geometría de puntos que contiene los detalles de la posición del dispositivo.</p>	Habilitado	Habilitado
<p><code>PositionProperties</code></p> <p>Un conjunto de pares clave-valor asociados con la actualización de posición.</p>	Habilitado	Habilitado
<p><code>GeofenceGeometry</code></p> <p>Geometría de una geocerca poligonal que representa el área geocercada.</p>	Habilitado	Habilitado
<p><code>DeviceId</code></p> <p>El identificador del dispositivo especificado al cargar una actualización de la posición del dispositivo en un recurso de rastreo.</p>	Habilitado	No compatible
<p><code>GeofenceId</code></p> <p>Un identificador que se especifica al almacenar una geometría de geocerca o un lote de geocercas en una colección de geocercas determinada.</p>	Habilitado	No compatible

Note

Amazon Location permite automáticamente el cifrado en reposo mediante claves AWS propias para proteger los datos de identificación personal sin coste alguno.

Sin embargo, se aplican AWS KMS cargos por el uso de una clave administrada por el cliente. Para obtener más información acerca de los precios, consulte [Precios de AWS Key Management Service](#).

Para obtener más información AWS KMS, consulte [¿Qué es AWS Key Management Service?](#)

Cómo utiliza Amazon Location Service las subvenciones en AWS KMS

Amazon Location requiere una [concesión](#) para utilizar su clave administrada por el cliente.

Cuando creas un [recurso de rastreo](#) o una [colección de geovallas](#) cifrada con una clave gestionada por el cliente, Amazon Location crea una subvención en tu nombre enviando una [CreateGrants](#) solicitud a. AWS KMS Las concesiones in se AWS KMS utilizan para dar acceso a Amazon Location a una clave de KMS en la cuenta de un cliente.

Amazon Location necesita la concesión para utilizar la clave administrada por el cliente para las siguientes operaciones internas:

- Envíe [DescribeKeys](#) solicitudes AWS KMS a para comprobar que el identificador de clave de KMS simétrico gestionado por el cliente introducido al crear un rastreador o una colección de geovallas es válido.
- Envíe [GenerateDataKeyWithoutPlaintext](#) solicitudes AWS KMS para generar claves de datos cifradas por su clave administrada por el cliente.
- Envíe solicitudes de [descifrado](#) AWS KMS a para descifrar las claves de datos cifrados para que puedan usarse para cifrar sus datos.

Puede revocar el acceso a la concesión o eliminar el acceso del servicio a la clave administrada por el cliente en cualquier momento. Si lo hace, Amazon Location no podrá acceder a ninguno de los datos cifrados por la clave administrada por el cliente, lo que afectará a las operaciones que dependen de esos datos. Por ejemplo, si intentas [obtener las posiciones de los dispositivos](#) a partir de un rastreador cifrado al que Amazon Location no puede acceder, la operación mostrará un `AccessDeniedException` error.

Crear una clave administrada por el cliente

Puede crear una clave simétrica gestionada por el cliente mediante las API o las API AWS Management Console. AWS KMS

Para crear una clave simétrica administrada por el cliente

Siga los pasos para [crear una clave simétrica gestionada por el cliente](#) que se indican en la Guía para desarrolladores de AWS Key Management Service .

Política de claves

Las políticas de clave controlan el acceso a la clave administrada por el cliente. Cada clave administrada por el cliente debe tener exactamente una política de clave, que contiene instrucciones que determinan quién puede usar la clave y cómo puede utilizarla. Cuando crea la clave administrada por el cliente, puede especificar una política de clave. Para obtener más información, consulte [Administración del acceso a las claves administradas por el cliente](#) en la Guía para desarrolladores de AWS Key Management Service .

Para utilizar la clave administrada por el cliente con sus recursos de Amazon Location, se deben permitir las siguientes operaciones de API en la política de claves:

- [kms:CreateGrant](#): añade una concesión a una clave administrada por el cliente. Otorga el acceso de control a una clave KMS específica, que permite acceder a las [operaciones de concesión](#) que requieren acceso verificado. Para obtener más información sobre el [uso de concesiones](#), consulte la Guía para desarrolladores de AWS Key Management Service .

Esto permite a Amazon Location realizar las siguientes tareas:

- Llamar a `GenerateDataKeyWithoutPlainText` para generar una clave de datos cifrada y almacenarla, ya que la clave de datos no se utiliza inmediatamente para cifrar.
- Llamar a `Decrypt` para usar la clave de datos cifrados almacenada para acceder a los datos cifrados.
- Configurar una entidad principal que se retire para permitir que el servicio `RetireGrant`.
- [kms:DescribeKey](#): proporciona los detalles de la clave administrada por el cliente para permitir que el servicio de Amazon Location valide la clave.

Los siguientes son ejemplos de declaraciones de política que puede agregar para Amazon Location:

```
"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" : "*"
}
]

```

Para obtener más información sobre [cómo especificar permisos en una política](#), consulte la Guía para desarrolladores de AWS Key Management Service .

Para obtener información sobre la [solución de problemas de acceso a las claves](#), consulte la Guía para desarrolladores de AWS Key Management Service .

Especificar una clave administrada por el cliente para Amazon Location

Puede especificar una clave administrada por el cliente como cifrado de segunda capa para los siguientes recursos:

- [Recurso de seguimiento](#)
- [Recolección de geocerca](#)

Al crear un recurso, puede especificar la clave de datos introduciendo un ID de KMS, que Amazon Location utiliza para cifrar los datos personales identificables almacenados en el recurso.

- ID de KMS: [identificador clave](#) de una clave gestionada por el AWS KMS cliente. Introduzca el ID de la clave, el ARN de la clave, el nombre de alias o el ARN del alias.

Contexto de cifrado de Amazon Location Service

Un [contexto de cifrado](#) es un conjunto opcional de pares clave-valor que pueden contener información contextual adicional sobre los datos.

AWS KMS utiliza el contexto de cifrado como [datos autenticados adicionales](#) para respaldar el cifrado [autenticado](#). Al incluir un contexto de cifrado en una solicitud de cifrado de datos, AWS KMS vincula el contexto de cifrado a los datos cifrados. Para descifrar los datos, debe incluir el mismo contexto de cifrado en la solicitud.

Contexto de cifrado de Amazon Location Service

Amazon Location utiliza el mismo contexto de cifrado en todas las operaciones AWS KMS criptográficas, donde la clave es `aws:geo:arn` y el valor es el [nombre de recurso de Amazon \(ARN\) del recurso](#).

Example

```
"encryptionContext": {
```



```
"aws:geo:arn": "arn:aws:geo:us-west-2:111122223333:geofence-collection/SAMPLE-GeofenceCollection"
}
```

Uso del contexto de cifrado para la supervisión

Si utiliza una clave simétrica gestionada por el cliente para cifrar su rastreador o su colección de geocercas, también puede utilizar el contexto de cifrado en los registros y registros de auditoría para identificar cómo se utiliza la clave gestionada por el cliente. El contexto de cifrado también aparece en [los registros generados por AWS CloudTrail Amazon CloudWatch Logs](#).

Utilizar el contexto de cifrado para controlar el acceso a la clave administrada por el cliente

Puede utilizar el contexto de cifrado en las políticas de claves y las políticas de IAM como `conditions` para controlar el acceso a la clave simétrica administrada por el cliente. Puede usar también una restricción de contexto de cifrado en una concesión.

Amazon Location utiliza el contexto de cifrado para restringir las concesiones que permiten el acceso a la clave administrada por el cliente o a su cuenta y región. La restricción de concesión requiere que las operaciones que permite la concesión utilicen el contexto de cifrado especificado.

Example

Los siguientes son ejemplos de declaraciones de política de claves para conceder acceso a una clave administrada por el cliente para un contexto de cifrado específico. La condición de esta declaración de política exige que las concesiones tengan una restricción de contexto de cifrado que especifique el contexto de cifrado.

```
{
  "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"
      }
    }
  }
}

```

Supervisión de las claves de cifrado para Amazon Location Service

Cuando utilizas una clave gestionada por el AWS KMS cliente con tus recursos de Amazon Location Service, puedes utilizar [AWS CloudTrailAmazon CloudWatch Logs](#) para realizar un seguimiento de las solicitudes a las que Amazon Location envía AWS KMS.

Los siguientes ejemplos son AWS CloudTrail eventos para `CreateGrant`, `GenerateDataKeyWithoutPlainTextDecrypt`, y `DescribeKey` para monitorear las operaciones de KMS llamadas por Amazon Location para acceder a los datos cifrados por la clave administrada por el cliente:

CreateGrant

Cuando utilizas una clave gestionada por el AWS KMS cliente para cifrar tu rastreador o los recursos de recolección de geovallas, Amazon Location envía una `CreateGrant` solicitud en tu nombre para acceder a la clave KMS de tu cuenta. AWS La concesión que Amazon Location crea es específica para el recurso asociado a la clave administrada por el cliente de AWS KMS . Además, Amazon Location utiliza la operación `RetireGrant` para eliminar una concesión cuando se elimina un recurso.

El siguiente evento de ejemplo registra la operación de `CreateGrant`:

```

{
  "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": "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

Cuando habilitas una clave gestionada por el AWS KMS cliente para tu rastreador o recurso de recolección de geovallas, Amazon Location crea una clave de tabla única. Envía una `GenerateDataKeyWithoutPlainText` solicitud a la AWS KMS que se especifica la clave gestionada por el AWS KMS cliente para el recurso.

El siguiente evento de ejemplo registra la operación de `GenerateDataKeyWithoutPlainText`:

```

{
  "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": "57f5dbec-16da-413e-979f-2c4c6663475e"
  }
}

```

Decrypt

Cuando accedes a un rastreador cifrado o a una colección de geocercas, Amazon Location solicita a la operación de Decrypt que utilice la clave de datos cifrados almacenada para acceder a los datos cifrados.

El siguiente evento de ejemplo registra la operación de Decrypt:

```

{
  "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

Amazon Location utiliza la operación de `DescribeKey` para verificar si la clave gestionada por el cliente de AWS KMS asociada a su rastreador o colección de geocercas existe en la cuenta y la región.

El siguiente evento de ejemplo registra la operación de `DescribeKey`:

```

{
  "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"
}
```

Más información

Los siguientes recursos proporcionan más información sobre cifrado de datos en reposo.

- Para obtener más información acerca de los [conceptos básicos de AWS Key Management Service](#), consulte la Guía para desarrolladores de AWS Key Management Service .
- Para obtener más información sobre [las prácticas recomendadas de seguridad AWS Key Management Service](#), consulte la [Guía para AWS Key Management Service](#) desarrolladores.

Cifrado de datos en tránsito para Amazon Location Service

Amazon Location protege los datos en tránsito, a medida que viajan hacia y desde el servicio, cifrando automáticamente todos los datos entre redes mediante el protocolo de cifrado seguridad de la capa de transporte (TLS) 1.2. Las solicitudes HTTPS directas enviadas a las API de Amazon Location Service se firman mediante el [algoritmo AWS Signature versión 4](#) para establecer una conexión segura.

Identity and Access Management para Amazon Location Service

AWS Identity and Access Management (IAM) es una herramienta Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a AWS los recursos. Los administradores de IAM controlan quién se puede autenticar (iniciar sesión) y autorizar (tener permisos) para utilizar los recursos de Amazon Location. La IAM es una Servicio de AWS herramienta que puede utilizar sin coste adicional.

Temas

- [Público](#)
- [Autenticación con identidades](#)
- [Administración de acceso mediante políticas](#)
- [Cómo funciona Amazon Location Service con IAM](#)
- [Cómo funciona Amazon Location Service con usuarios no autenticados](#)
- [Ejemplos de políticas basadas en identidad para Amazon Location Service](#)
- [Solución de problemas de identidad y acceso de Amazon Location Service](#)

Público

La forma de usar AWS Identity and Access Management (IAM) varía según el trabajo que realices en Amazon Location.

Usuario de servicio: si utiliza el servicio Amazon Location Service para realizar el trabajo, el administrador proporciona las credenciales y los permisos que necesita. A medida que utilice más características de Amazon Location para realizar el trabajo, es posible que necesite permisos adicionales. Entender cómo se administra el acceso puede ayudarlo a solicitar los permisos correctos al administrador. Si no puede acceder a una característica de Amazon Location, consulte [Solución de problemas de identidad y acceso de Amazon Location Service](#).

Administrador de servicio: si está a cargo de los recursos de Amazon Location de su empresa, probablemente tenga acceso completo a Amazon Location. Su trabajo consiste en determinar a qué características y recursos de Amazon Location deben acceder sus usuarios del servicio. Luego, debe enviar solicitudes a su administrador de IAM para cambiar los permisos de los usuarios de su servicio. Revise la información de esta página para conocer los conceptos básicos de IAM. Para obtener más información acerca de cómo la empresa puede utilizar IAM con Amazon Location, consulte [Cómo funciona Amazon Location Service con IAM](#).

Administrador de IAM: si es un administrador de IAM, es posible que desee obtener información sobre cómo escribir políticas para administrar el acceso a Amazon Location. Para consultar ejemplos de políticas de Amazon Location basadas en identidades que puede utilizar en IAM, consulte [Ejemplos de políticas basadas en identidad para Amazon Location Service](#).

Autenticación con identidades

La autenticación es la forma en que inicias sesión para AWS usar tus credenciales de identidad. Debe estar autenticado (con quien haya iniciado sesión AWS) como usuario de IAM o asumiendo una función de IAM. Usuario raíz de la cuenta de AWS

Puede iniciar sesión AWS como una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios (Centro de identidades de IAM), la autenticación de inicio de sesión único de su empresa y sus credenciales de Google o Facebook son ejemplos de identidades federadas. Al iniciar sesión como una identidad federada, su administrador habrá configurado previamente la federación de identidades mediante roles de IAM. Cuando accedes AWS mediante la federación, estás asumiendo un rol de forma indirecta.

Según el tipo de usuario que sea, puede iniciar sesión en el portal AWS Management Console o en el de AWS acceso. Para obtener más información sobre cómo iniciar sesión AWS, consulte [Cómo iniciar sesión Cuenta de AWS en su](#) Guía del AWS Sign-In usuario.

Si accede AWS mediante programación, AWS proporciona un kit de desarrollo de software (SDK) y una interfaz de línea de comandos (CLI) para firmar criptográficamente sus solicitudes con sus

credenciales. Si no utilizas AWS herramientas, debes firmar las solicitudes tú mismo. Para obtener más información sobre cómo usar el método recomendado para firmar las solicitudes usted mismo, consulte [Firmar las solicitudes de la AWS API](#) en la Guía del usuario de IAM.

Independientemente del método de autenticación que use, es posible que deba proporcionar información de seguridad adicional. Por ejemplo, le AWS recomienda que utilice la autenticación multifactor (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte [Autenticación multifactor](#) en la Guía del usuario de AWS IAM Identity Center y [Uso de la autenticación multifactor \(MFA\) en AWS](#) en la Guía del usuario de IAM.

Cuenta de AWS usuario root

Al crear una Cuenta de AWS, comienza con una identidad de inicio de sesión que tiene acceso completo a todos Servicios de AWS los recursos de la cuenta. Esta identidad se denomina usuario Cuenta de AWS raíz y se accede a ella iniciando sesión con la dirección de correo electrónico y la contraseña que utilizaste para crear la cuenta. Recomendamos encarecidamente que no utilice el usuario raíz para sus tareas diarias. Proteja las credenciales del usuario raíz y utilícelas solo para las tareas que solo el usuario raíz pueda realizar. Para ver la lista completa de las tareas que requieren que inicie sesión como usuario raíz, consulte [Tareas que requieren credenciales de usuario raíz](#) en la Guía del usuario de IAM.

Identidad federada

Como práctica recomendada, exija a los usuarios humanos, incluidos los que requieren acceso de administrador, que utilicen la federación con un proveedor de identidades para acceder Servicios de AWS mediante credenciales temporales.

Una identidad federada es un usuario del directorio de usuarios de su empresa, un proveedor de identidades web AWS Directory Service, el directorio del Centro de Identidad o cualquier usuario al que acceda Servicios de AWS mediante las credenciales proporcionadas a través de una fuente de identidad. Cuando las identidades federadas acceden Cuentas de AWS, asumen funciones y las funciones proporcionan credenciales temporales.

Para una administración de acceso centralizada, le recomendamos que utilice AWS IAM Identity Center. Puede crear usuarios y grupos en el Centro de identidades de IAM o puede conectarse y sincronizarse con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en todas sus Cuentas de AWS aplicaciones. Para obtener más información, consulte [¿Qué es el Centro de identidades de IAM?](#) en la Guía del usuario de AWS IAM Identity Center .

Usuarios y grupos de IAM

Un [usuario de IAM](#) es una identidad propia Cuenta de AWS que tiene permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos emplear credenciales temporales, en lugar de crear usuarios de IAM que tengan credenciales de larga duración como contraseñas y claves de acceso. No obstante, si tiene casos de uso específicos que requieran credenciales de larga duración con usuarios de IAM, recomendamos rotar las claves de acceso. Para más información, consulte [Rotar las claves de acceso periódicamente para casos de uso que requieran credenciales de larga duración](#) en la Guía del usuario de IAM.

Un [grupo de IAM](#) es una identidad que especifica un conjunto de usuarios de IAM. No puede iniciar sesión como grupo. Puede usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos de grandes conjuntos de usuarios. Por ejemplo, podría tener un grupo cuyo nombre fuese IAMAdmins y conceder permisos a dicho grupo para administrar los recursos de IAM.

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios tienen credenciales permanentes a largo plazo y los roles proporcionan credenciales temporales. Para más información, consulte [Cuándo crear un usuario de IAM \(en lugar de un rol\)](#) en la Guía del usuario de IAM.

Roles de IAM

Un [rol de IAM](#) es una identidad dentro de usted Cuenta de AWS que tiene permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una determinada persona. Puede asumir temporalmente una función de IAM en el AWS Management Console [cambiando](#) de función. Puede asumir un rol llamando a una operación de AWS API AWS CLI o utilizando una URL personalizada. Para más información sobre los métodos para el uso de roles, consulte [Uso de roles de IAM](#) en la Guía del usuario de IAM.

Los roles de IAM con credenciales temporales son útiles en las siguientes situaciones:

- **Acceso de usuario federado:** para asignar permisos a una identidad federada, puede crear un rol y definir sus permisos. Cuando se autentica una identidad federada, se asocia la identidad al rol y se le conceden los permisos define el rol. Para obtener información acerca de roles para federación, consulte [Creación de un rol para un proveedor de identidades de terceros](#) en la Guía del usuario de IAM. Si utiliza IAM Identity Center, debe configurar un conjunto de permisos. IAM Identity Center correlaciona el conjunto de permisos con un rol en IAM para controlar a qué

pueden acceder las identidades después de autenticarse. Para obtener información acerca de los conjuntos de permisos, consulte [Conjuntos de permisos](#) en la Guía del usuario de AWS IAM Identity Center .

- **Permisos de usuario de IAM temporales:** un usuario de IAM puede asumir un rol de IAM para recibir temporalmente permisos distintos que le permitan realizar una tarea concreta.
- **Acceso entre cuentas:** puede utilizar un rol de IAM para permitir que alguien (una entidad principal de confianza) de otra cuenta acceda a los recursos de la cuenta. Los roles son la forma principal de conceder acceso entre cuentas. Sin embargo, con algunas Servicios de AWS, puedes adjuntar una política directamente a un recurso (en lugar de usar un rol como proxy). Para obtener información acerca de la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte [Acceso a recursos entre cuentas en IAM](#) en la Guía del usuario de IAM.
- **Acceso entre servicios:** algunos Servicios de AWS utilizan funciones en otros Servicios de AWS. Por ejemplo, cuando realiza una llamada en un servicio, es común que ese servicio ejecute aplicaciones en Amazon EC2 o almacene objetos en Amazon S3. Es posible que un servicio haga esto usando los permisos de la entidad principal, usando un rol de servicio o usando un rol vinculado al servicio.
- **Sesiones de acceso directo (FAS):** cuando utilizas un usuario o un rol de IAM para realizar acciones en ellas AWS, se te considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. El FAS utiliza los permisos del principal que llama Servicio de AWS y los solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte [Reenviar sesiones de acceso](#).
- **Rol de servicio:** un rol de servicio es un [rol de IAM](#) que adopta un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte [Creación de un rol para delegar permisos a un Servicio de AWS](#) en la Guía del usuario de IAM.
- **Función vinculada al servicio:** una función vinculada a un servicio es un tipo de función de servicio que está vinculada a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

- Aplicaciones que se ejecutan en Amazon EC2: puede usar un rol de IAM para administrar las credenciales temporales de las aplicaciones que se ejecutan en una instancia EC2 y realizan AWS CLI solicitudes a la API. Es preferible hacerlo de este modo a almacenar claves de acceso en la instancia de EC2. Para asignar una AWS función a una instancia EC2 y ponerla a disposición de todas sus aplicaciones, debe crear un perfil de instancia adjunto a la instancia. Un perfil de instancia contiene el rol y permite a los programas que se ejecutan en la instancia de EC2 obtener credenciales temporales. Para más información, consulte [Uso de un rol de IAM para conceder permisos a aplicaciones que se ejecutan en instancias Amazon EC2](#) en la Guía del usuario de IAM.

Para obtener información sobre el uso de los roles de IAM, consulte [Cuándo crear un rol de IAM \(en lugar de un usuario\)](#) en la Guía del usuario de IAM.

Administración de acceso mediante políticas

El acceso se controla AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política es un objeto AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando un director (usuario, usuario raíz o sesión de rol) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. La mayoría de las políticas se almacenan AWS como documentos JSON. Para obtener más información sobre la estructura y el contenido de los documentos de política JSON, consulte [Información general de políticas JSON](#) en la Guía del usuario de IAM.

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos.

Las políticas de IAM definen permisos para una acción independientemente del método que se utilice para realizar la operación. Por ejemplo, suponga que dispone de una política que permite la acción `iam:GetRole`. Un usuario con esa política puede obtener información sobre el rol de la API AWS Management Console AWS CLI, la o la AWS API.

Políticas basadas en identidades

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte [Creación de políticas de IAM](#) en la Guía del usuario de IAM.

Las políticas basadas en identidades pueden clasificarse además como políticas insertadas o políticas administradas. Las políticas insertadas se integran directamente en un único usuario, grupo o rol. Las políticas administradas son políticas independientes que puede adjuntar a varios usuarios, grupos y roles de su Cuenta de AWS empresa. Las políticas administradas incluyen políticas AWS administradas y políticas administradas por el cliente. Para más información sobre cómo elegir una política administrada o una política insertada, consulte [Elegir entre políticas administradas y políticas insertadas](#) en la Guía del usuario de IAM.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe [especificar una entidad principal](#) en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o Servicios de AWS

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puedes usar políticas AWS gestionadas de IAM en una política basada en recursos.

Listas de control de acceso (ACL)

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3 y Amazon VPC son ejemplos de servicios que admiten las ACL. AWS WAF Para obtener más información sobre las ACL, consulte [Información general de Lista de control de acceso \(ACL\)](#) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite tipos de políticas adicionales y menos comunes. Estos tipos de políticas pueden establecer el máximo de permisos que los tipos de políticas más frecuentes le conceden.

- **Límites de permisos:** un límite de permisos es una característica avanzada que le permite establecer los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM (usuario o rol de IAM). Puede establecer un límite de permisos para una entidad. Los permisos resultantes son la intersección de las políticas basadas en la identidad de la entidad y los límites de permisos. Las políticas basadas en recursos que especifiquen el usuario o rol en el campo `Principal` no estarán restringidas por el límite de permisos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información sobre los límites de los permisos, consulte [Límites de permisos para las entidades de IAM](#) en la Guía del usuario de IAM.
- **Políticas de control de servicios (SCP):** las SCP son políticas de JSON que especifican los permisos máximos para una organización o unidad organizativa (OU). AWS Organizations es un servicio para agrupar y gestionar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilita todas las características en una organización, entonces podrá aplicar políticas de control de servicio (SCP) a una o a todas sus cuentas. El SCP limita los permisos de las entidades en las cuentas de los miembros, incluidas las de cada una. Usuario raíz de la cuenta de AWS Para obtener más información acerca de Organizations y las SCP, consulte [Funcionamiento de las SCP](#) en la Guía del usuario de AWS Organizations .
- **Políticas de sesión:** las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidades del rol y las políticas de la sesión. Los permisos también pueden proceder de una política en función de recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para más información, consulte [Políticas de sesión](#) en la Guía del usuario de IAM.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo AWS determinar si se debe permitir una solicitud cuando se trata de varios tipos de políticas, consulte la [lógica de evaluación de políticas](#) en la Guía del usuario de IAM.

Cómo funciona Amazon Location Service con IAM

Antes de utilizar IAM para administrar el acceso a Amazon Location, obtenga información sobre qué características de IAM se encuentran disponibles con Amazon Location.

Características de IAM que puede utilizar con Amazon Location Service

Característica de IAM	Amazon Location Support
Políticas basadas en identidades	Sí
Políticas basadas en recursos	No
Acciones de políticas	Sí
Recursos de políticas	Sí
Claves de condición de política (específicas del servicio)	Sí
ACL	No
ABAC (etiquetas en políticas)	Sí
Credenciales temporales	Sí
Permisos de entidades principales	No
Roles de servicio	No
Roles vinculados al servicio	No

Para obtener una visión general de cómo funcionan Amazon Location y otros AWS servicios con la mayoría de las funciones de IAM, consulta [AWS los servicios que funcionan con IAM](#) en la Guía del usuario de IAM.

Políticas de Amazon Location basadas en identidades

Compatibilidad con las políticas basadas en identidad Sí

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte [Creación de políticas de IAM](#) en la Guía del usuario de IAM.

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. No es posible especificar la entidad principal en una política basada en identidad porque se aplica al usuario o rol al que está adjunto. Para más información sobre los elementos que puede utilizar en una política de JSON, consulte [Referencia de los elementos de las políticas de JSON de IAM](#) en la Guía del usuario de IAM.

Ejemplos de políticas basadas en identidad para Amazon Location

Para ver ejemplos de políticas basadas en identidades de Amazon Location, consulte [Ejemplos de políticas basadas en identidad para Amazon Location Service](#).

Políticas basadas en recursos de Amazon Location

Compatibilidad con las políticas basadas en recursos No

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe [especificar una entidad principal](#) en una política en función de recursos. Los directores pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Para habilitar el acceso entre cuentas, puede especificar toda una cuenta o entidades de IAM de otra cuenta como la entidad principal de una política en función de recursos. Añadir a una política en función de recursos una entidad principal entre cuentas es solo una parte del establecimiento de una relación de confianza. Cuando el principal y el recurso son diferentes Cuentas de AWS, el administrador de IAM de la cuenta de confianza también debe conceder a la entidad principal (usuario o rol) permiso para acceder al recurso. Para conceder el permiso, adjunte la entidad a una política basada en identidad. Sin embargo, si la política en función de recursos concede el acceso a una entidad principal de la misma cuenta, no es necesaria una política basada en identidad adicional. Para obtener más información, consulte [Acceso a recursos entre cuentas en IAM](#) en la Guía del usuario de IAM.

Acciones de políticas de Amazon Location

Admite acciones de política	Sí
-----------------------------	----

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Las acciones políticas suelen tener el mismo nombre que la operación de AWS API asociada. Hay algunas excepciones, como acciones de solo permiso que no tienen una operación de API coincidente. También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Para ver una lista de las acciones de Amazon VPC, consulte [Acciones definidas por Amazon Location Service](#) en la Referencia de autorizaciones de servicio.

Las acciones de políticas de Amazon Location utilizan el siguiente prefijo antes de la acción:

```
geo
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

```
"Action": [  
    "geo:action1",
```

```
"geo:action2"  
]
```

Puede utilizar caracteres comodín (*) para especificar varias acciones . Por ejemplo, para especificar todas las acciones que comiencen con la palabra Get, incluya la siguiente acción:

```
"Action": "geo:Get*"
```

Para ver ejemplos de políticas basadas en identidades de Amazon Location, consulte [Ejemplos de políticas basadas en identidad para Amazon Location Service](#).

Recursos de políticas para Amazon Location

Admite recursos de políticas	Sí
------------------------------	----

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso utilizando el [Nombre de recurso de Amazon \(ARN\)](#). Puede hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

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

Para ver una lista de tipos de recursos de Amazon Location y sus ARN, consulte [Tipos de recurso definidos por Amazon Location Service](#) en la Referencia de autorizaciones de servicio. Para obtener información acerca de las acciones con las que puede especificar el ARN de cada recurso, consulte [Acciones definidas por Amazon Location Service](#).

Para ver ejemplos de políticas basadas en identidades de Amazon Location, consulte [Ejemplos de políticas basadas en identidad para Amazon Location Service](#).

Claves de condición de políticas de Amazon Location

Admite claves de condición de políticas específicas del servicio	Sí
--	----

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puede crear expresiones condicionales que utilicen [operadores de condición](#), tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación AND lógica. Si especifica varios valores para una única clave de condición, AWS evalúa la condición mediante una OR operación lógica. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puede utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puede conceder un permiso de usuario de IAM para acceder a un recurso solo si está etiquetado con su nombre de usuario de IAM. Para más información, consulte [Elementos de la política de IAM: variables y etiquetas](#) en la Guía del usuario de IAM.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición AWS globales, consulte las claves de [contexto de condición AWS globales en la Guía](#) del usuario de IAM.

Para obtener una lista de las claves de condición de Amazon Location, consulte [Claves de condición para Amazon Location Service](#) en la Referencia de autorizaciones de servicio. Para obtener más información acerca de las acciones y los recursos con los que puede utilizar una clave de condición, consulte las [acciones definidas por Amazon Location Service](#).

Amazon Location admite claves de condición que te permiten permitir o denegar el acceso a geocercas o dispositivos específicos en tus declaraciones de política. Las siguientes claves de condición están disponibles:

- `geo:GeofenceIds` para su uso con acciones de geocerca. El tipo es `ArrayOfString`.
- `geo:DeviceIds` para su uso con las acciones de Rastreador. El tipo es `ArrayOfString`.

Puede utilizar las siguientes acciones con `geo:GeofenceIds` en su política de IAM:

- `BatchDeleteGeofences`
- `BatchPutGeofences`
- `GetGeofence`
- `PutGeofence`

Puede utilizar las siguientes acciones con `geo:DeviceIds` en su política de IAM:

- `BatchDeleteDevicePositionHistory`
- `BatchGetDevicePosition`
- `BatchUpdateDevicePosition`
- `GetDevicePosition`
- `GetDevicePositionHistory`

Note

No puede usar estas claves de condición con las acciones `BatchEvaluateGeofences`, `ListGeofences`, o `ListDevicePosition`.

Para ver ejemplos de políticas basadas en identidades de Amazon Location, consulte [Ejemplos de políticas basadas en identidad para Amazon Location Service](#).

ACL en Amazon Location

Admite las ACL

No

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

ABAC con Amazon

Admite ABAC (etiquetas en las políticas)	Sí
--	----

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define permisos en función de atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar etiquetas a las entidades de IAM (usuarios o roles) y a muchos AWS recursos. El etiquetado de entidades y recursos es el primer paso de ABAC. A continuación, designa las políticas de ABAC para permitir operaciones cuando la etiqueta de la entidad principal coincida con la etiqueta del recurso al que se intenta acceder.

ABAC es útil en entornos que crecen con rapidez y ayuda en situaciones en las que la administración de las políticas resulta engorrosa.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el [elemento de condición](#) de una política utilizando las claves de condición `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más información sobre ABAC, consulte [¿Qué es ABAC?](#) en la Guía del usuario de IAM. Para ver un tutorial con los pasos para configurar ABAC, consulte [Uso del control de acceso basado en atributos \(ABAC\)](#) en la Guía del usuario de IAM.

Para obtener más información acerca del etiquetado de recursos de Amazon Location, consulte [Etiquetado de los recursos de Amazon Location Service](#).

Para consultar un ejemplo de política basada en la identidad para limitar el acceso a un recurso en función de las etiquetas de ese recurso, consulte [Controlar el acceso a los recursos en función de las etiquetas](#).

Uso de credenciales temporales con Amazon Location

Compatible con el uso de credenciales temporales	Sí
--	----

Algunos Servicios de AWS no funcionan cuando inicias sesión con credenciales temporales. Para obtener información adicional, incluidas las que Servicios de AWS funcionan con credenciales temporales, consulta [Cómo Servicios de AWS funcionan con IAM](#) en la Guía del usuario de IAM.

Utiliza credenciales temporales si inicia sesión en ellas AWS Management Console mediante cualquier método excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accedes AWS mediante el enlace de inicio de sesión único (SSO) de tu empresa, ese proceso crea automáticamente credenciales temporales. También crea credenciales temporales de forma automática cuando inicia sesión en la consola como usuario y luego cambia de rol. Para más información sobre el cambio de roles, consulte [Cambio a un rol \(consola\)](#) en la Guía del usuario de IAM.

Puedes crear credenciales temporales manualmente mediante la AWS CLI API o. AWS A continuación, puede utilizar esas credenciales temporales para acceder AWS. AWS recomienda generar credenciales temporales de forma dinámica en lugar de utilizar claves de acceso a largo plazo. Para más información, consulte [Credenciales de seguridad temporales en IAM](#).

Permisos de entidades principales entre servicios de Amazon Location

Admite sesiones de acceso directo (FAS)	No
---	----

Cuando utilizas un usuario o un rol de IAM para realizar acciones en AWSél, se te considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos del principal que llama y los que solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Servicio de AWS Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte [Reenviar sesiones de acceso](#).

Roles de servicio para Amazon Location

Compatible con roles de servicio	No
----------------------------------	----

Un rol de servicio es un [rol de IAM](#) que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener

más información, consulte [Creación de un rol para delegar permisos a un Servicio de AWS](#) en la Guía del usuario de IAM.

Warning

Cambiar los permisos de un rol de servicio podría interrumpir la funcionalidad de Amazon Location. Edite los roles de servicio solo cuando Amazon Location proporcione orientación para hacerlo.

Roles vinculados a servicios para Amazon Location

Compatible con roles vinculados al servicio	No
---	----

Un rol vinculado a un servicio es un tipo de rol de servicio que está vinculado a un Servicio de AWS. El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en su Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

Para más información sobre cómo crear o administrar roles vinculados a servicios, consulte [Servicios de AWS que funcionan con IAM](#). Busque un servicio en la tabla que incluya Yes en la columna Rol vinculado a un servicio. Seleccione el vínculo Sí para ver la documentación acerca del rol vinculado a servicios para ese servicio.

Cómo funciona Amazon Location Service con usuarios no autenticados

Muchos escenarios para usar Amazon Location Service, incluida la visualización de mapas en la web o en una aplicación móvil, requieren permitir el acceso a los usuarios que no han iniciado sesión con IAM. Para estos escenarios no autenticados, tiene dos opciones.

- **Utilizar claves de API:** para conceder acceso a usuarios no autenticados, puede crear claves de API que den acceso de solo lectura a sus recursos de Amazon Location Service. Esto resulta útil en el caso de que no desee autenticar a todos los usuarios. Por ejemplo, una aplicación web. Para obtener más información acerca de las claves API, consulte [Permitir el acceso de invitados no autenticados a su aplicación mediante claves de API](#).
- **Utilizar Amazon Cognito:** puede utilizar Amazon Cognito para conceder acceso anónimo como alternativa a las claves de API. Amazon Cognito le permite crear una autorización más completa

con política de IAM para definir lo que pueden hacer los usuarios no autenticados. Para obtener más información sobre cómo usar Amazon Cognito, consulte [Cómo permitir el acceso a sus aplicaciones como invitado no autenticado con Amazon Cognito](#).

Para obtener información general sobre cómo proporcionar acceso a usuarios no autenticados, consulte [Concesión de acceso a Amazon Location Service](#).

Ejemplos de políticas basadas en identidad para Amazon Location Service

De forma predeterminada, los usuarios y roles no tienen permiso para crear ni modificar los recursos de Amazon Location. Tampoco pueden realizar tareas mediante la AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS la API. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos.

Para obtener información acerca de cómo crear una política basada en identidades de IAM mediante el uso de estos documentos de políticas JSON de ejemplo, consulte [Creación de políticas de IAM](#) en la Guía del usuario de IAM.

Para obtener más información sobre las acciones y los tipos de recursos definidos por Amazon Location, incluido el formato de los ARN para cada uno de los tipos de recursos, consulte [Acciones, recursos y claves de condición de Amazon Location Service](#) en la Referencia de autorizaciones de servicio.

Temas

- [Prácticas recomendadas sobre las políticas](#)
- [Uso de la consola de Amazon Location](#)
- [Cómo permitir a los usuarios consultar sus propios permisos](#)
- [Uso de los recursos de Amazon Location Service en la política](#)
- [Permisos para actualizar las posiciones de los dispositivos](#)
- [Política de solo lectura para los recursos del rastreador](#)
- [Política de creación de geocercas](#)
- [Política de solo lectura para geocercas](#)
- [Permisos para renderizar un recurso de mapa](#)

- [Permisos para permitir las operaciones de búsqueda](#)
- [Política de solo lectura para calculadoras de rutas](#)
- [Controle el acceso a los recursos en función de las claves de condición](#)
- [Controlar el acceso a los recursos en función de las etiquetas](#)

Prácticas recomendadas sobre las políticas

Las políticas basadas en identidades determinan si alguien puede crear, eliminar o acceder a los recursos de Amazon Location de la cuenta. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos para muchos casos de uso comunes. Están disponibles en su Cuenta de AWS. Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Con el fin de obtener más información, consulte las [políticas administradas por AWS](#) o las [políticas administradas por AWS para funciones de trabajo](#) en la Guía de usuario de IAM.
- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulte [Políticas y permisos en IAM](#) en la Guía del usuario de IAM.
- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo AWS CloudFormation. Para obtener más información, consulte [Elementos de la política de JSON de IAM: Condición](#) en la Guía del usuario de IAM.
- Utilice el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para

más información, consulte [Política de validación de Analizador de acceso de IAM](#) en la Guía de usuario de IAM.

- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para solicitar la MFA cuando se invocan las operaciones de la API, agregue las condiciones de la MFA a sus políticas. Para más información, consulte [Configuración del acceso a una API protegido por MFA](#) en la Guía de usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte las [Prácticas recomendadas de seguridad en IAM](#) en la Guía del usuario de IAM.

Uso de la consola de Amazon Location

Para acceder a la consola de Amazon Location Service, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirte enumerar y ver detalles sobre los recursos de Amazon Location de tu cuenta Cuenta de AWS. Si crea una política basada en identidades que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para las entidades (usuarios o roles) que tengan esa política.

No es necesario que concedas permisos mínimos de consola a los usuarios que solo realicen llamadas a la API AWS CLI o a la AWS API. En su lugar, permite acceso únicamente a las acciones que coincidan con la operación de API que intentan realizar.

Para asegurarse de que los usuarios y los roles puedan seguir utilizando la consola de Amazon Location, asocie también a las siguiente política administrada a las entidades. Para más información, consulte [Adición de permisos a un usuario](#) en la Guía del usuario de IAM:

La siguiente política da acceso a la consola de Amazon Location Service para poder crear, eliminar, enumerar y ver detalles sobre los recursos de Amazon Location en su cuenta de AWS .

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GeoPowerUser",
      "Effect": "Allow",
      "Action": [
        "geo:*"
      ],
    }
  ],
}
```

```

    "Resource": "*"
  }
]
}

```

Como alternativa, puede conceder permisos de solo lectura para facilitar el acceso de solo lectura. Con los permisos de solo lectura, aparecerá un mensaje de error si el usuario intenta realizar acciones de escritura, como crear o eliminar recursos. Para ver un ejemplo, consulte [the section called "Política de solo lectura para rastreadores"](#)

Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas administradas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la API AWS CLI o AWS .

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "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": "*"
}
]
}

```

Uso de los recursos de Amazon Location Service en la política

Amazon Location Service utiliza los siguientes prefijos para los recursos:

Prefijo de recurso de Amazon Location

Recurso	Prefijo de recurso
Recursos de mapas	map
Recursos de sitios	place-index
Recursos de ruta	route-calculator
Recursos de rastreo	tracker
Recursos de la colección de geocerca	geofence-collection

Utilice la siguiente sintaxis ARN:

```
arn:Partition:geo:Region:Account:ResourcePrefix/ResourceName
```

Para obtener más información sobre el formato de los ARN, consulte Nombres de [recursos de Amazon \(ARN\) y espacios de nombres de AWS servicio](#).

Ejemplos

- Utilice el siguiente ARN para permitir el acceso a un recurso de mapa específico.

```
"Resource": "arn:aws:geo:us-west-2:account-id:map/map-resource-name"
```

- Para especificar el acceso a todos los recursos de map que pertenecen a una cuenta específica, utilice el carácter comodín (*):

```
"Resource": "arn:aws:geo:us-west-2:account-id:map/*"
```

- Algunas acciones de Amazon Location, como las que se utilizan para crear recursos, no se pueden llevar a cabo en un recurso específico. En dichos casos, debe utilizar el carácter comodín (*).

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

Para ver una lista de tipos de recursos de Amazon Location y sus ARN, consulte [Tipos de recurso definidos por Amazon Location Service](#) en la Referencia de autorizaciones de servicio. Para obtener información acerca de las acciones con las que puede especificar el ARN de cada recurso, consulte [Acciones definidas por Amazon Location Service](#).

Permisos para actualizar las posiciones de los dispositivos

Para actualizar las posiciones de los dispositivos de varios rastreadores, querrá conceder a un usuario acceso a uno o más de sus recursos de seguimiento. También querrá permitir que el usuario actualice un lote de posiciones de dispositivos.

En este ejemplo, además de conceder acceso a los recursos de *Tracker1* y *Tracker2*, la siguiente política otorga permiso para utilizar la acción `geo:BatchUpdateDevicePosition` contra los recursos de *Tracker1* y *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"
      ]
    }
  ]
}
```

Si quiere limitar al usuario a que solo pueda actualizar las posiciones de los dispositivos de un dispositivo específico, puede agregar una clave de condición para el identificador de ese dispositivo.

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

Política de solo lectura para los recursos del rastreador

Para crear una política de solo lectura para todos los recursos de rastreadores de tu AWS cuenta, tendrás que conceder el acceso a todos los recursos de rastreadores. También querrá conceder a los usuarios acceso a las acciones que les permitan obtener la posición del dispositivo en varios dispositivos, obtener la posición del dispositivo desde un único dispositivo y obtener el historial de posiciones.

En este ejemplo, la siguiente política concede permisos para las siguientes acciones:

- `geo:BatchGetDevicePosition` para recuperar la posición de varios dispositivos.
- `geo:GetDevicePosition` para recuperar la posición de un solo dispositivo.
- `geo:GetDevicePositionHistory` para recuperar el historial de posición de un dispositivo.

```
{
  "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/*"
    }
  ]
}
```

Política de creación de geocercas

Para crear una política que permita a un usuario crear geocercas, tendrá que conceder acceso a acciones específicas que permitan a los usuarios crear una o más geocercas en una colección de geocercas.

La siguiente política permite realizar las siguientes acciones en la *recopilación*:

- `geo:BatchPutGeofence` para crear varias geocercas.
- `geo:PutGeofence` para crear una geocerca única.

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



```
}
```

Política de solo lectura para geocercas

Para crear una política de solo lectura para las geocercas almacenadas en una colección de geocercas de su cuenta de AWS , tendrá que conceder acceso a las acciones que lean datos de la colección de geocercas que almacena las geocercas.

La siguiente política permite realizar las siguientes acciones en la *recopilación*:

- `geo:ListGeofences` para enumerar las geocercas de la colección de geocercas especificada.
- `geo:GetGeofence` para recuperar una geocerca de la colección de geocercas.

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

Permisos para renderizar un recurso de mapa

Para conceder los permisos necesarios para renderizar los mapas, tendrás que conceder acceso a los mosaicos, los sprites, los glifos y el descriptor de estilo del mapa:

- `geo:GetMapTile` recupera los mosaicos del mapa que se utilizan para renderizar de forma selectiva las entidades de un mapa.
- `geo:GetMapSprites` recupera la hoja de sprites en PNG y el documento JSON correspondiente en el que se describen los desfases que contiene.
- `geo:GetMapGlyphs` recupera los archivos Glyph utilizados para mostrar el texto.

- `geo:GetMapStyleDescriptor` recupera el descriptor de estilo del mapa, que contiene las reglas de renderizado.

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

Permisos para permitir las operaciones de búsqueda

Para crear una política que permita las operaciones de búsqueda, primero tendrás que conceder acceso al recurso de indexación de sitios de tu AWS cuenta. También querrá conceder acceso a las acciones que permitan al usuario buscar con texto mediante geocodificación y buscar con una posición mediante geocodificación inversa.

En este ejemplo, además de conceder acceso a *PlaceIndex*, la siguiente política también concede permisos para las siguientes acciones:

- `geo:SearchPlaceIndexForPosition` permite buscar lugares o puntos de interés cercanos a una posición determinada.
- `geo:SearchPlaceIndexForText` permite buscar una dirección, un nombre, una ciudad o una región mediante texto de formato libre.

```
{
  "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"
}
]
}

```

Política de solo lectura para calculadoras de rutas

Puede crear una política de solo lectura para permitir que un usuario acceda a un recurso de calculadora de rutas para calcular una ruta.

En este ejemplo, además de conceder acceso a *ExampleCalculator*, la siguiente política concede permiso a la siguiente operación:

- `geo:CalculateRoute` calcula una ruta con una posición de salida, una posición de destino y una lista de posiciones de puntos de referencia.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RoutesReadOnly",
      "Effect": "Allow",
      "Action": [
        "geo:CalculateRoute"
      ],
      "Resource": "arn:aws:geo:us-west-2:accountID:route-calculator/ExampleCalculator"
    }
  ]
}

```

Controle el acceso a los recursos en función de las claves de condición

Al crear una política de IAM para conceder acceso al uso de geocercas o posiciones de dispositivos, puede utilizar [operadores de condición](#) para controlar con mayor precisión a qué geocercas o

dispositivos puede acceder un usuario. Para ello, incluye el identificador de la geocerca o el identificador del dispositivo en el elemento `Condition` de la política.

En el siguiente ejemplo se muestra cómo puede crear una política que permita a un usuario actualizar las posiciones de un dispositivo específico.

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

Controlar el acceso a los recursos en función de las etiquetas

Cuando crea una política de IAM para permitir el uso de sus recursos de Amazon Location, puede usar el control de [acceso basado en atributos para controlar](#) mejor los recursos que un usuario puede modificar, usar o eliminar. Para ello, incluya información sobre las etiquetas en el elemento `Condition` de su política para controlar el acceso en función de las [etiquetas](#) de sus recursos.

En el siguiente ejemplo de política se muestra cómo puede crear una política que permita a un usuario crear geocercas. Esto otorga el permiso a las siguientes acciones para crear una o más geocercas en una colección de geocercas denominada *Colección*:

- `geo:BatchPutGeofence` para crear varias geocercas.
- `geo:PutGeofence` para crear una geocerca única.

Sin embargo, esta política utiliza el elemento `Condition` para conceder el permiso solo si la etiqueta `Colectión`, `Owner`, tiene el valor del nombre de usuario de ese usuario.

- Por ejemplo, si un usuario llamado `richard-roe` intenta ver una `coleccion` de ubicaciones de Amazon, la `coleccion` debe estar etiquetada como `Owner=richard-roe` o `owner=richard-roe`. De lo contrario, se deniega el acceso al usuario.

Note

La clave de la etiqueta de condición `Owner` coincide con los nombres de las claves de condición `Owner` y `owner` porque no distinguen entre mayúsculas y minúsculas. Para obtener más información, consulte [Elementos de la política de JSON de IAM: Condición](#) en la Guía del usuario de 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}"}
      }
    }
  ]
}
```

Para ver un tutorial sobre [cómo definir los permisos de acceso a los recursos de AWS en función de las etiquetas](#), consulte la Guía del usuario de AWS Identity and Access Management .

Solución de problemas de identidad y acceso de Amazon Location Service

Utilice la siguiente información para diagnosticar y solucionar los problemas habituales que pueden surgir cuando se trabaja con Amazon Location e IAM.

Temas

- [No tengo autorización para realizar una acción en Amazon Location](#)
- [No estoy autorizado a realizar tareas como: PassRole](#)
- [Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis recursos de Amazon Location](#)

No tengo autorización para realizar una acción en Amazon Location

Si recibe un error que indica que no tiene autorización para realizar una acción, las políticas se deben actualizar para permitirle realizar la acción.

En el siguiente ejemplo, el error se produce cuando el usuario de IAM `mateojackson` intenta utilizar la consola para consultar los detalles acerca de un recurso ficticio `my-example-widget`, pero no tiene los permisos ficticios `geo:GetWidget`.

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

En este caso, la política del usuario `mateojackson` debe actualizarse para permitir el acceso al recurso `my-example-widget` mediante la acción `geo:GetWidget`.

Si necesita ayuda, póngase en contacto con su AWS administrador. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

No estoy autorizado a realizar tareas como: PassRole

Si recibe un error que indica que no tiene autorización para llevar a cabo la acción `iam:PassRole`, las políticas se deben actualizar para permitirle pasar un rol a Amazon Location.

Algunos Servicios de AWS permiten transferir una función existente a ese servicio en lugar de crear una nueva función de servicio o una función vinculada a un servicio. Para ello, debe tener permisos para transferir el rol al servicio.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en Amazon Location. Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

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

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con su administrador. AWS El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis recursos de Amazon Location

Puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Puede especificar una persona de confianza para que asuma el rol. En el caso de los servicios que admitan las políticas basadas en recursos o las listas de control de acceso (ACL), puede utilizar dichas políticas para conceder a las personas acceso a sus recursos.

Para más información, consulte lo siguiente:

- Para saber si Amazon Location admite estas características, consulte [Cómo funciona Amazon Location Service con IAM](#).
- Para obtener información sobre cómo proporcionar acceso a los recursos de su Cuentas de AWS propiedad, consulte [Proporcionar acceso a un usuario de IAM en otro de su propiedad Cuenta de AWS en](#) la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta [Cómo proporcionar acceso a recursos que Cuentas de AWS son propiedad de terceros](#) en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulte [Proporcionar acceso a usuarios autenticados externamente \(identidad federada\)](#) en la Guía del usuario de IAM.
- Para obtener información acerca del uso de roles y políticas basadas en recursos para el acceso entre cuentas, consulte [Acceso a recursos entre cuentas en IAM](#) en la Guía del usuario de IAM.

Respuesta a incidentes en Amazon Location Service

La seguridad de AWS es nuestra mayor prioridad. Como parte del [modelo de responsabilidad compartida AWS](#) en la nube, AWS gestiona un centro de datos y una arquitectura de red que cumple con los requisitos de las organizaciones más sensibles a la seguridad. Como AWS cliente, usted comparte la responsabilidad de mantener la seguridad en la nube. Esto significa que usted controla la seguridad que decide implementar desde las AWS herramientas y funciones a las que tiene acceso.

Al establecer una base de seguridad que cumpla con los objetivos de las aplicaciones que se ejecutan en la nube, puede detectar las desviaciones a las que puede responder. Dado que la respuesta a los incidentes de seguridad puede ser un tema complejo, le recomendamos que consulte los siguientes recursos para comprender mejor el impacto que la respuesta a incidentes (IR) y sus elecciones tienen en sus objetivos corporativos: la [Guía de respuesta a incidentes de seguridad de AWS](#), el documento técnico sobre [las mejores prácticas de seguridad de AWS](#) y el [Marco de adopción de la nube de AWS \(AWS CAF\)](#).

Registro y monitoreo en Amazon Location Service

El registro y la supervisión son una parte importante de la respuesta a los incidentes. Le permite establecer una base de seguridad para detectar las desviaciones que puede investigar y a las que puede responder. Mediante la implementación de registros y monitoreo para Amazon Location Service, podrá mantener la fiabilidad, la disponibilidad y el rendimiento de sus proyectos y recursos.

AWS proporciona varias herramientas que pueden ayudarle a registrar y recopilar datos para responder a los incidentes:

AWS CloudTrail

Amazon Location Service se integra con AWS CloudTrail, que es un servicio que proporciona un registro de las acciones realizadas por un usuario, función o AWS servicio. Entre estas se incluyen las acciones realizadas desde la consola de Amazon Location Service y las llamadas mediante programación a las operaciones de la API de Amazon Location. Estos registros de acción se denominan eventos. Para obtener más información, consulta [Cómo registrar y monitorizar Amazon Location Service con AWS CloudTrail](#).

Amazon CloudWatch

Puedes usar Amazon CloudWatch para recopilar y analizar las métricas relacionadas con tu cuenta de Amazon Location Service. Puedes activar CloudWatch las alarmas para que te

notifiquen si una métrica cumple determinadas condiciones y ha alcanzado un umbral específico. Cuando creas una alarma, CloudWatch envía una notificación a un Amazon Simple Notification Service que tú definas. Para obtener más información, consulta la sección [Supervisión de Amazon Location Service con Amazon CloudWatch](#).

AWS Health Cuadros de mando

Mediante los [paneles](#) de control de AWS Health , puede verificar el estado del servicio Amazon Location Service. También puede supervisar y ver los datos históricos sobre cualquier evento o problema que pueda afectar a su AWS entorno. Para obtener más información, consulte la [Guía del usuario de AWS Health](#).

Validación de conformidad para Amazon Location Service

Para saber si uno Servicio de AWS está dentro del ámbito de aplicación de programas de cumplimiento específicos, consulte [Servicios de AWS Alcance por programa de cumplimiento](#) [Servicios de AWS](#) de cumplimiento y elija el programa de cumplimiento que le interese. Para obtener información general, consulte Programas de [AWS cumplimiento > Programas AWS](#) .

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte [Descarga de informes en AWS Artifact](#) .

Su responsabilidad de cumplimiento al Servicios de AWS utilizarlos viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. AWS proporciona los siguientes recursos para ayudar con el cumplimiento:

- [Guías de inicio rápido sobre seguridad y cumplimiento](#): estas guías de implementación analizan las consideraciones arquitectónicas y proporcionan los pasos para implementar entornos básicos centrados en AWS la seguridad y el cumplimiento.
- Diseño de [arquitectura para garantizar la seguridad y el cumplimiento de la HIPAA en Amazon Web Services](#): este documento técnico describe cómo las empresas pueden crear aplicaciones aptas para AWS la HIPAA.

Note

No Servicios de AWS todas cumplen los requisitos de la HIPAA. Para más información, consulte la [Referencia de servicios compatibles con HIPAA](#).

- [AWS Recursos de](#) de cumplimiento: esta colección de libros de trabajo y guías puede aplicarse a su industria y ubicación.
- [AWS Guías de cumplimiento para clientes](#): comprenda el modelo de responsabilidad compartida desde el punto de vista del cumplimiento. Las guías resumen las mejores prácticas para garantizar la seguridad Servicios de AWS y orientan los controles de seguridad en varios marcos (incluidos el Instituto Nacional de Estándares y Tecnología (NIST), el Consejo de Normas de Seguridad del Sector de Tarjetas de Pago (PCI) y la Organización Internacional de Normalización (ISO)).
- [Evaluación de los recursos con reglas](#) en la guía para AWS Config desarrolladores: el AWS Config servicio evalúa en qué medida las configuraciones de los recursos cumplen con las prácticas internas, las directrices del sector y las normas.
- [AWS Security Hub](#)— Este Servicio de AWS proporciona una visión completa del estado de su seguridad interior AWS. Security Hub utiliza controles de seguridad para evaluar sus recursos de AWS y comprobar su cumplimiento con los estándares y las prácticas recomendadas del sector de la seguridad. Para obtener una lista de los servicios y controles compatibles, consulte la [Referencia de controles de Security Hub](#).
- [Amazon GuardDuty](#): Servicio de AWS detecta posibles amenazas para sus cargas de trabajo Cuentas de AWS, contenedores y datos mediante la supervisión de su entorno para detectar actividades sospechosas y maliciosas. GuardDuty puede ayudarlo a cumplir con varios requisitos de conformidad, como el PCI DSS, al cumplir con los requisitos de detección de intrusiones exigidos por ciertos marcos de cumplimiento.
- [AWS Audit Manager](#)— Esto le Servicio de AWS ayuda a auditar continuamente su AWS uso para simplificar la gestión del riesgo y el cumplimiento de las normativas y los estándares del sector.

Resiliencia de Amazon Location Service

La infraestructura AWS global se basa en zonas Regiones de AWS de disponibilidad. Regiones de AWS proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que están conectadas mediante redes de baja latencia, alto rendimiento y alta redundancia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de uno o varios centros de datos.

[Para obtener más información sobre las zonas de disponibilidad Regiones de AWS y las zonas de disponibilidad, consulte Infraestructura global.AWS](#)

Además de la infraestructura AWS global, Amazon Location ofrece varias funciones que ayudan a respaldar sus necesidades de respaldo y resiliencia de datos.

Seguridad de la infraestructura en Amazon Location Service

Como servicio gestionado, Amazon Location Service está protegido por la seguridad de la red AWS global. Para obtener información sobre los servicios AWS de seguridad y cómo se protege la infraestructura, consulte [Seguridad AWS en la nube](#). Para diseñar su AWS entorno utilizando las mejores prácticas de seguridad de la infraestructura, consulte [Protección de infraestructuras en un marco](#) de buena AWS arquitectura basado en el pilar de la seguridad.

Utilizas las llamadas a la API AWS publicadas para acceder a Amazon Location a través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Conjuntos de cifrado con confidencialidad directa total (PFS) como DHE (Ephemeral Diffie-Hellman) o ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad de seguridad de IAM principal. También puede utilizar [AWS Security Token Service](#) (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Configuración y análisis de vulnerabilidades en Amazon Location

La configuración y los controles de TI son una responsabilidad compartida entre usted AWS y usted, nuestro cliente. Para obtener más información, consulte el [modelo de responsabilidad AWS compartida](#).

Prevención de la sustitución confusa entre servicios

El problema de la sustitución confusa es un problema de seguridad en el que una entidad que no tiene permiso para realizar una acción puede obligar a una entidad con más privilegios a realizar la acción. En AWS, la suplantación de identidad entre servicios puede provocar el confuso problema de un diputado. La suplantación entre servicios puede producirse cuando un servicio (el servicio que lleva a cabo las llamadas) llama a otro servicio (el servicio al que se llama). El servicio que lleva a

cabo las llamadas se puede manipular para utilizar sus permisos a fin de actuar en función de los recursos de otro cliente de una manera en la que no debe tener permiso para acceder. Para evitarlo, AWS proporciona herramientas que lo ayudan a proteger sus datos para todos los servicios con entidades principales de servicio a las que se les ha dado acceso a los recursos de su cuenta.

Amazon Location Service no actúa como un servicio de llamadas en tu nombre a otros AWS servicios, por lo que no necesitas añadir estas protecciones en este caso. Para obtener más información sobre el suplente confuso, consulte [El problema del suplente confuso](#) en la Guía del usuario AWS Identity and Access Management .

Prácticas recomendadas de seguridad para Amazon Location Service

Amazon Location Service proporciona una serie de características de seguridad que debe tener en cuenta a la hora de desarrollar e implementar sus propias políticas de seguridad. Las siguientes prácticas recomendadas son directrices generales y no constituyen una solución de seguridad completa. Puesto que es posible que estas prácticas recomendadas no sean adecuadas o suficientes para el entorno, considérelas como consideraciones útiles en lugar de como normas.

Prácticas recomendadas de detección de seguridad para Amazon Location Service

Las siguientes prácticas recomendadas para Amazon Location Service le pueden ser de utilidad para detectar incidentes de seguridad:

Implementación de las herramientas de monitoreo de AWS

La supervisión es fundamental para la respuesta a los incidentes y mantiene la fiabilidad y la seguridad de los recursos de Amazon Location Service y sus soluciones. Puede implementar herramientas de monitoreo de entre las diversas herramientas y servicios disponibles en AWS para monitorear sus recursos y otros servicios de AWS.

Por ejemplo, Amazon te CloudWatch permite monitorizar las métricas de Amazon Location Service y configurar alarmas para que te notifiquen si una métrica cumple determinadas condiciones que has establecido y ha alcanzado un umbral que has definido. Cuando creas una alarma, puedes configurar CloudWatch el envío de una notificación de alerta mediante Amazon Simple Notification Service. Para obtener más información, consulte [the section called “Registro y supervisión”](#).

Habilite las herramientas de registro de AWS

Logging proporciona un registro de las acciones que realiza un usuario, un rol o un servicio de Amazon Location Service. Puede implementar herramientas de registro, por ejemplo, AWS CloudTrail para recopilar datos sobre las acciones a fin de detectar actividades inusuales en la API.

Al crear un rastro, puede configurarlo CloudTrail para que registre eventos. Los eventos son registros de operaciones de recursos realizadas en un recurso o dentro de él, como la solicitud realizada a Amazon Location, la dirección IP desde la que se realizó la solicitud, cuándo se realizó la solicitud, junto con datos adicionales. Para obtener más información, consulte [Registro de eventos de datos para senderos](#) en la Guía del AWS CloudTrail usuario.

Prácticas recomendadas de seguridad preventiva para Amazon Location Service

Las siguientes prácticas recomendadas para Amazon Location Service le pueden ser de utilidad para prevenir incidentes de seguridad:

Use conexiones seguras

Utilice siempre conexiones cifradas, como las que se utilizan al principio de `https://` para proteger la información confidencial en tránsito.

Implementación del acceso a los recursos con privilegios mínimos

Al crear políticas personalizadas para recursos de Amazon Location, conceda solo los permisos necesarios para llevar a cabo una tarea. Es más seguro comenzar con un conjunto mínimo de permisos y conceder permisos adicionales según sea necesario. La implementación del acceso con privilegios mínimos es esencial a la hora de reducir los riesgos de seguridad y el impacto que podrían causar los errores o los intentos malintencionados. Para obtener más información, consulte [the section called “Identity and Access Management”](#).

Utilice identificadores únicos a nivel mundial como identificadores de dispositivo

Use las siguientes convenciones para los ID de los dispositivos.

- Los ID de dispositivo deben ser únicos.
- Los ID de los dispositivos no deben ser secretos, ya que se pueden usar como claves externas para otros sistemas.

- Los ID de los dispositivos no deben contener información de identificación personal (PII), como los identificadores de los dispositivos telefónicos o las direcciones de correo electrónico.
- Los ID de los dispositivos no deben ser predecibles. Se recomiendan los identificadores opacos, como los UUID.

No incluya la información de identificación personal en las propiedades de posición del dispositivo

Al enviar actualizaciones de dispositivos (por ejemplo, usarlas [DevicePositionUpdate](#)), no incluya información de identificación personal (PII), como el número de teléfono o la dirección de correo electrónico, en el `PositionProperties`

Prácticas recomendadas para Amazon Location Service

En este tema, se proporcionan las prácticas recomendadas para ayudarle a utilizar Amazon Location Service. Si bien estas prácticas recomendadas pueden ayudarte a aprovechar al máximo las ventajas de Amazon Location Service, no representan una solución completa. Debe seguir únicamente las recomendaciones aplicables a su entorno.

Temas

- [Seguridad](#)
- [Administración de recursos](#)
- [Administración de costos y facturación](#)
- [Quotas y uso](#)

Seguridad

Para ayudar a administrar o incluso evitar los riesgos de seguridad, considere las siguientes prácticas recomendadas:

- Utilice las funciones de federación de identidad y de IAM para gestionar, controlar o limitar el acceso a los recursos de sus ubicaciones de Amazon. Para obtener más información, consulte [Prácticas recomendadas de IAM](#) en la Guía del usuario de IAM.
- Siga el principio del privilegio mínimo para conceder solo el acceso mínimo requerido a tus recursos de Amazon Location Service. Para obtener más información, consulte [the section called "Administración de acceso mediante políticas"](#).

- En el caso de los recursos de Amazon Location Service utilizados en aplicaciones web, restrinja el acceso mediante una condición de `aws:referer:iam` que limite el uso a sitios distintos de los incluidos en la lista de permitidos.
- Use herramientas de monitoreo y registro para rastrear el acceso y el uso de los recursos. Para obtener más información, consulte el apartado [Registro the section called “Registro y supervisión” de eventos de datos para senderos](#) en la Guía del AWS CloudTrail usuario.
- Utilice conexiones seguras, como las que inician con `https://` para agregar seguridad y proteger a los usuarios contra los ataques mientras se transmiten datos entre el servidor y el navegador.

Para obtener más información acerca de las prácticas recomendadas de seguridad preventiva y de detección, consulte el tema sobre las [the section called “Prácticas recomendadas de seguridad”](#).

Administración de recursos

Para gestionar de forma eficaz los recursos de ubicación en Amazon Location Service, considere las siguientes prácticas recomendadas:

- Utilice puntos de enlace regionales que sean fundamentales para la base de usuarios prevista para mejorar su experiencia. Para obtener más información acerca de los puntos de conexión por región, consulte [Regiones y puntos de conexión de Amazon](#).
- En el caso de los recursos que utilizan proveedores de datos, como los recursos de mapas y los recursos de indexación de sitios, asegúrese de cumplir las condiciones de uso del proveedor de datos específico. Para obtener más información, consulte [Proveedores de datos](#).
- Minimice la creación de recursos al disponer de un recurso para cada configuración de mapa, índice de ubicación o rutas. Dentro de una región, normalmente solo se necesita un recurso por proveedor de datos o estilo de mapa. La mayoría de las aplicaciones utilizan los recursos existentes y no crean recursos en tiempo de ejecución.
- Cuando utilice distintos recursos en una sola aplicación, como un recurso de mapa y una calculadora de rutas, utilice el mismo proveedor de datos en cada recurso para garantizar que los datos coincidan. Por ejemplo, que la geometría de una ruta que cree con la calculadora de rutas se alinee con las calles del mapa dibujadas con el recurso de mapa.

Administración de costos y facturación

Para ayudarle a administrar los costos y la facturación, considere las siguientes prácticas recomendadas:

- Usa herramientas de monitoreo, como Amazon CloudWatch, para realizar un seguimiento del uso de los recursos. Puede configurar alertas que le notifiquen cuando el uso esté a punto de superar los límites especificados. Para obtener más información, consulte [Creación de una alarma de facturación para supervisar los cargos estimados de AWS](#) en la Guía del CloudWatch usuario de Amazon.

Quotas y uso

Cuenta de AWS Incluye cuotas que establecen un límite predeterminado para la cantidad de uso. Puede configurar alarmas para que le avisen cuando su consumo se acerque al límite, y puede solicitar un aumento de la cuota cuando lo necesite. Para obtener más información sobre cómo trabajar con cuotas, consulte los siguientes temas.

- [Quotas de Amazon Location Service](#)
- [Se usa CloudWatch para monitorear el uso comparándolo con las cuotas](#)
- [Visualiza tus cuotas de servicio y configura las alarmas](#) en la Guía del CloudWatch usuario de Amazon.

Puede crear alarmas para avisarle con antelación cuando esté a punto de superar sus límites. Te recomendamos configurar alarmas para cada cuota en cada Región de AWS lugar en el que utilices Amazon Location. Por ejemplo, puede supervisar el uso que hace de la operación `SearchPlaceIndexForText` y crear una alarma cuando supere el 80 por ciento de su cuota actual.

Cuando reciba una advertencia de alarma sobre su cuota, deberá decidir qué hacer. Es posible que esté utilizando recursos adicionales porque su base de clientes ha crecido. En ese caso, es posible que desee solicitar un aumento de su cuota, por ejemplo, un aumento del 50 por ciento en la cuota para una llamada a la API en esa región. O bien, puede que haya un error en su servicio que provoque que realices más llamadas innecesarias a Amazon Location. En ese caso, querrá resolver el problema de su servicio.

Historial del documento

En la siguiente tabla se describen las versiones de la documentación de Amazon Location Service. Para recibir notificaciones sobre las actualizaciones, puede suscribirse a una fuente RSS.

Cambio	Descripción	Fecha
Amazon Location Service lanza un nuevo SDK para JavaScript	Para facilitar el desarrollo de aplicaciones de Amazon Location con interfaces web, Amazon Location añade un nuevo SDK de código abierto compatible con el SDK de AWS para la JavaScript versión 3, lo que simplifica la autenticación y utiliza GeoJSON. Para obtener más información, consulte Amazon Location SDK .	6 de julio de 2023
Amazon Location Service publica las claves de API para su disponibilidad general	Amazon Location añade compatibilidad con el lugar y la ruta, y anuncia la disponibilidad general de la función de claves de API. Para obtener más información, consulte Uso de claves de API .	6 de julio de 2023
Amazon Location Service añade Amazon EventBridge Events para actualizaciones de posición	Amazon Location añade soporte para enviar eventos de actualización de posición del rastreador a EventBridge. Para obtener más información, incluido cómo habilitar los eventos para un rastreador,	6 de julio de 2023

consulte <u>Reaccionar ante los eventos con EventBridge</u>.		
Amazon Location añade metadatos a las geovallas	Con la API de Amazon Location, ahora puede agregar propiedades de metadatos a sus geovallas. Se almacenan con tu geocerca y se incluyen en los eventos relacionados con la geocerca en Amazon. EventBridge Para obtener más información, consulte Dibujar geovallas y Crear reglas de eventos .	15 de junio de 2023
Amazon Location añade categorías para los lugares	Amazon Location añade categorías en los resultados de búsqueda locales y filtra los resultados por categoría. Para obtener más información, consulte Categorías y filtrado .	15 de junio de 2023
Amazon Location introduce puntos de vista políticos	Amazon Location añade puntos de vista políticos a determinados estilos de mapas. Para obtener más información, consulte Puntos de vista políticos .	23 de mayo de 2023
Amazon Location introduce un nuevo sitio de demostraciones y muestras	Amazon Location anuncia un nuevo sitio web que le da acceso a demostraciones y muestras de Amazon Location. Para obtener más información, consulte el Sitio de demostración de Amazon Location .	3 de mayo de 2023

[Amazon Location introduce rutas más largas en CalculateRouteMatrix](#)

Amazon Location ahora permite rutas de longitud ilimitada para las rutas matriciales creadas con el proveedor de datos HERE. Para obtener más información, consulte [Planificación de rutas más largas](#).

24 de abril de 2023

[La documentación de Amazon Location añade diferencias de características según el proveedor de datos](#)

La documentación de Amazon Location se ha actualizado con información sobre las diferencias entre cada proveedor de datos en Mapas, Búsqueda de lugares y Rutas. Para obtener más información, consulte [Características por proveedor de datos](#).

30 de marzo de 2023

[Disponibilidad general de los mapas de Open Data de Amazon Location](#)

Disponibilidad general del proveedor y el estilo de datos de Amazon Location Service, según OpenStreetMap los mapas Daylight. Para obtener más información, consulte [Open Data](#).

7 de marzo de 2023

[Amazon Location añade un nuevo método de autorización en la vista previa](#)

Amazon Location Service añade claves de API como un nuevo método de autorización para usuarios anónimos, en modo de vista previa. Para obtener más información, consulte [Permitir el acceso de invitados no autenticados a su aplicación mediante claves de API](#).

23 de febrero de 2023

[Documentación de Amazon Location actualizada con las mejores prácticas de IAM más recientes](#)

La documentación de Amazon Location Service se ha actualizado para cumplir con las mejores prácticas de AWS Identity and Access Management más recientes. Para obtener más información, consulte [Seguridad en Amazon Location Service](#).

26 de enero de 2023

[Amazon Location Service se suma GrabMaps como proveedor de datos en el sudeste asiático](#)

Amazon Location GrabMaps se presenta como proveedor de datos en el sudeste asiático. Para obtener más información, consulte [GrabMaps](#).

10 de enero de 2023

[Vista previa de los mapas de Open Data de Amazon Location Service](#)

El nuevo proveedor y estilo de datos de ubicación de Amazon se agregaron en la vista previa pública, basados en OpenStreetMap los mapas Daylight. Para obtener más información, consulte [Open Data \(Versión preliminar\)](#).

15 de diciembre de 2022

[Nuevos estilos de imágenes satelitales de HERE](#)

Se han agregado dos nuevos estilos de mapa para los mapas que utilizan HERE como proveedor de datos: las imágenes satelitales de HERE y los estilos de mapa híbridos de HERE. Para obtener más información, consulte [Estilos de mapa de HERE](#).

25 de octubre de 2022

Unidades en direcciones	Amazon Location Service ahora admite unidades en direcciones, por ejemplo, «123 Main St, Apartment 3B, Anytown, EE. UU.».	20 de septiembre de 2022
Encontrar lugares por ID	Amazon Location Service ahora incluye asistencia para encontrar la ubicación exacta sugerida por la operación SearchPlaceIndexForSuggestions mediante la operación GetPlace. Consulte Uso de la función autocompletar .	20 de septiembre de 2022
Claves de condición adicionales para la política de IAM	Amazon Location Service ahora admite claves de condición adicionales que permiten configurar el acceso a geovallas o dispositivos específicos en la política de IAM. Consulte Claves de condición .	23 de agosto de 2022
Geovallas circulares	Amazon Location Service ahora admite geovallas definidas como un círculo con un punto central y un radio, para obtener eventos cuando los dispositivos se encuentran a una distancia determinada de una ubicación. Consulte agregar geovallas circulares .	11 de agosto de 2022

Referencia de API combinada	Amazon Location Service ahora tiene una única guía de referencia de API, en lugar de guías independientes para cada subservicio. Para obtener más información sobre las API, consulte API de Amazon Location .	7 de julio de 2022
Integración de Service Quotas	Amazon Location ahora está integrado con Service Quotas , lo que le permite ver y administrar sus Quotas a través de AWS Management Console o utilizando AWS CLI.	6 de julio de 2022
Capítulo de documentación de conceptos actualizado	El capítulo sobre conceptos de Amazon Location se ha actualizado con más información para los usuarios de Amazon Location.	22 de abril de 2022
Nuevo tutorial de inicio rápido para Android	Se ha agregado un nuevo tutorial de inicio rápido para el desarrollo de Android con Kotlin para que los desarrolladores comiencen a trabajar rápidamente.	15 de abril de 2022
Nuevos estilos de mapas de HERE	Se han agregado dos nuevos estilos de mapa para los mapas que utilizan HERE como proveedor de datos. Para obtener más información, consulte Estilos de mapa de HERE .	15 de marzo de 2022

[Reestructuración de la documentación con tutoriales y ejemplos de código adicionales](#)

Esta guía para desarrolladores se ha reestructurado para encontrar temas más fácilmente, e incluye nuevos capítulos de [Inicio rápido](#) y [Ejemplos de código](#).

25 de febrero de 2022

[Filtrado de posición basado en la precisión para rastreadores](#)

Ahora puede usar filtros basados en la precisión al [crear un recurso de rastreador](#).

7 de diciembre de 2021

[Completar automáticamente los índices de lugares](#)

Ahora puede utilizar la función de [autocompletar](#) al buscar índices de lugares.

6 de diciembre de 2021

[Nuevo tutorial de Amplify para usar mapas](#)

Hay disponible un nuevo tutorial que demuestra cómo usar AWS Amplify para mostrar mapas en una aplicación web. El tutorial está disponible en [Uso de la librería Amplify con Amazon Location Service](#).

24 de noviembre de 2021

[Extensiones de consulta de lugares](#)

Amazon Location Service ahora permite establecer un idioma preferido para los resultados al realizar la geocodificación o la geocodificación inversa, y añade la zona horaria y otra información a los resultados. Para obtener más información sobre la geocodificación y la geocodificación inversa, consulte [Geocodificación, geocodificación inversa y búsqueda](#).

16 de noviembre de 2021

[Filtro de posición del rastreador](#)

Amazon Location Service añade una nueva función de filtrado de posición a los rastreadores que puede ayudarle a controlar los costos. Esta función filtra algunas actualizaciones de posición en los dispositivos antes de que se almacenen o se evalúen comparándolas con las geovallas. Para obtener más información sobre cómo filtrar posiciones, consulte [Rastreadores](#).

5 de octubre de 2021

<u>Operaciones actualizadas</u>	Se han añadido las siguientes operaciones a las referencias de la API de Amazon Location Service: <u>UpdateMapUpdatePlaceIndex</u> , <u>UpdateRouteCalculator</u> , <u>UpdateGeofenceCollection</u> , y <u>UpdateTracker</u> .	19 de julio de 2021
<u>Actualización de tutorial: funciones definidas por el usuario de Amazon Aurora PostgreSQL</u>	Se ha añadido un nuevo tutorial sobre cómo utilizar las <u>funciones definidas por el usuario de Amazon Aurora PostgreSQL con Amazon Location</u> para validar, limpiar y enriquecer los datos geoespaciales.	19 de julio de 2021
<u>CloudFormation Recursos de AWS</u>	Amazon Location ahora admite <u>la creación de los siguientes tipos de CloudFormation recursos en los recursos de AWS:</u> AWS::Location::Map AWS::Location::PlaceIndex AWS::Location::RouteCalculator AWS::Location::Tracker AWS::Location::TrackerConsumer yAWS::Location::GeofenceCollection .	7 de junio de 2021

Etiquetado de recursos	Ahora puede agregar etiquetas a sus recursos de Amazon Location para ayudarle a gestionar, identificar, organizar, buscar y filtrar sus recursos.	1 de junio de 2021
Disponibilidad general	Disponibilidad general de la documentación para desarrolladores de Amazon Location Service: actualización de Región y puntos de conexión y Service Quotas .	1 de junio de 2021
Esri Imagery	Amazon Location ahora admite el uso del estilo de mapa de Esri: Esri Imagery . Para obtener más información, consulte Esri World Imagery en el sitio web de Esri.	1 de junio de 2021
Calcular rutas	Ahora puedes usar las calculadoras de rutas de Amazon Location para calcular rutas y estimar el tiempo de viaje en función de la red de up-to-date carreteras y la información de tráfico en tiempo real del proveedor de datos que elijas.	1 de junio de 2021

<u>Cifrado mediante clave gestionado por el cliente de AWS KMS para los datos en reposo</u>	Amazon Location ahora admite el uso de una clave simétrica gestionada por el cliente que usted crea, posee y gestiona para <u>agregar una segunda capa de cifrado sobre el cifrado propiedad de AWS existente</u> .	1 de junio de 2021
<u>Versión de vista previa pública</u>	Versión inicial de la documentación para la versión de vista previa pública.	16 de diciembre de 2020
<u>Actualización de tutorial: visualización de mapas</u>	Los tutoriales para mostrar mapas en Android e iOS se han actualizado para usar el SDK MapLibre nativo. MapLibre	17 de marzo de 2020

Glosario de AWS

Para ver la terminología más reciente de AWS, consulte el [Glosario de AWS](#) en la Referencia de Glosario de AWS.

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la versión original de inglés, prevalecerá la versión en inglés.